

## MỤC LỤC

1.KHÁI NIỆM CSDL PHÂN TÁN: .....	3
2.KIẾN TRÚC CỦA MỘT HỆ CSDL PHÂN TÁN: .....	4
3.HỆ QUẢN TRỊ CSDL PHÂN TÁN (HQTCSDLPT) VÀ PHÂN LOẠI: .....	6
3.1.CÁC LOẠI CSDL PHÂN TÁN: .....	6
4.KỸ THUẬT PHÂN ĐOẠN, SAO BẢN VÀ ĐỊNH VỊ DỮ LIỆU: .....	8
A. CÁC ĐIỀU KIỆN RÀNG BUỘC TRONG THIẾT KẾ PHÂN ĐOẠN: .....	8
B. CÁC PHƯƠNG PHÁP PHÂN ĐOẠN: .....	9
C.SAO LẬP DỮ LIỆU: .....	13
D.ĐỊNH VỊ DỮ LIỆU: .....	14
5. TÍNH TRONG SUỐT CỦA HQTCSDLPT: .....	14
A.TRONG SUỐT PHÂN ĐOẠN (FRAGMENTATION TRANSPARENCY): .....	14
B.TÍNH TRONG SUỐT VỀ VỊ TRÍ (LOCATION TRANSPARENCY): .....	16
C. TRONG SUỐT ÁNH XẠ ĐỊA PHƯƠNG (LOCAL MAPPING TRANSPARENCY): .....	17
D. TRONG SUỐT NHÂN BẢN (REPLICATION TRANSPARENCY): .....	18
6. THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN: .....	18
A. CÁC BƯỚC THIẾT KẾ CSDL: .....	18
B. ĐỐI TƯỢNG THIẾT KẾ CỦA CƠ SỞ DỮ LIỆU PHÂN TÁN: .....	19
C. CÁC HƯỚNG TIẾP CẬN THIẾT KẾ CSDL: .....	20
D. THIẾT KẾ PHÂN ĐOẠN CƠ SỞ DỮ LIỆU: .....	23
7. XỬ LÝ TRUY VẤN TRONG CSDL PHÂN TÁN: .....	30
8.CƠ SỞ DỮ LIỆU PHÂN TÁN TRONG ORACLE .....	44
8.1.CÁC KHÁI NIỆM: .....	44
8.2.TỔNG QUAN CSDLPT TRONG ORACLE: .....	45
8.2.1.HỆ THỐNG CSDLPT ĐỒNG NHẤT: .....	45
8.2.2.HỆ THỐNG CSDLPT KHÔNG ĐỒNG NHẤT: .....	46
8.2.3.DATABASE LINKS(LIÊN KẾT CSDL): .....	49
8.2.4.CÁC TÊN CSDL TOÀN CỤC TRONG CÁC LIÊN KẾT CSDL: .....	51
8.3.GIẢI QUYẾT MẪU THUẦN TRONG MÔI TRƯỜNG PHÂN TÁN: .....	53

8.3.1.ĐẶT VẤN ĐỀ:.....	53
8.3.2.CÁC TÌNH HUỐNG:.....	53
8.3.3.CÁC PHƯƠNG PHÁP GIẢI QUYẾT:.....	54
8.3.4.PHÁT TRIỂN CHIẾN LƯỢC GIẢI QUYẾT MÂU THUẬN: .....	56
<b>8.4.CƠ CHẾ QUẢN LÝ GIAO TÁC:.....</b>	<b>57</b>
<b>8.5.SƠ ĐỒ CÁC TRẠM:.....</b>	<b>59</b>
8.5.1.SƠ ĐỒ TRẠM ĐIỀU HỢP:.....	59
8.5.2.SƠ ĐỒ TRẠNG THÁI TRẠM THÀNH PHẦN: .....	61
8.5.3.SƠ ĐỒ TRẠNG THÁI ĐIỂM THỎA THUẬN:.....	61
<b>8.6.KHẢ NĂNG PHỤC HỒI TRONG GIAO TÁC PHÂN TÁN: .....</b>	<b>62</b>
<b>8.7.MỘT SỐ VÍ DỤ LIÊN QUAN ĐẾN CƠ CHẾ THỎA THUẬN CỦA GIAO TÁC PHÂN TÁN: .....</b>	<b>63</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>65</b>

## 1.KHÁI NIỆM CSDL PHÂN TÁN:

*-Cơ sở dữ liệu phân tán:* Một tuyển tập dữ liệu có quan hệ logic với nhau, được phân bố trên các máy tính của một mạng máy tính.

*-Hệ quản trị CSDL phân tán:* Hệ thống phần mềm cho phép quản lý CSDL phân tán và đảm bảo tính trong suốt về sự phân tán đối với người dùng.

*-Ứng dụng cục bộ:* được yêu cầu và thực hiện trên máy tính ở một nút trong hệ CSDL phân tán và chỉ liên quan đến CSDL tại nút đó.

*-Ứng dụng toàn cục:* yêu cầu truy nhập dữ liệu ở nhiều nút thông qua hệ thống truyền thông.

*Ví dụ về hệ CSDL phân tán:* ATM và Google phân tán theo cách tự nhận biết, một yêu cầu gần server nào thì server đó xử lý. ATM phân tán rộng khắp, Google ở đâu cũng có. Tùy theo người lập trình và cách xử lý mà CSDL được tiến hành phát tán cho hợp lý.

*Ưu điểm của CSDL phân tán:*

- Phù hợp với cấu trúc của tổ chức.
- Nâng cao khả năng chia sẻ và tính tự trị địa phương.
- Nâng cao tính sẵn sàng.
- Nâng cao tính tin cậy.
- Nâng cao hiệu năng.
- Dễ mở rộng.

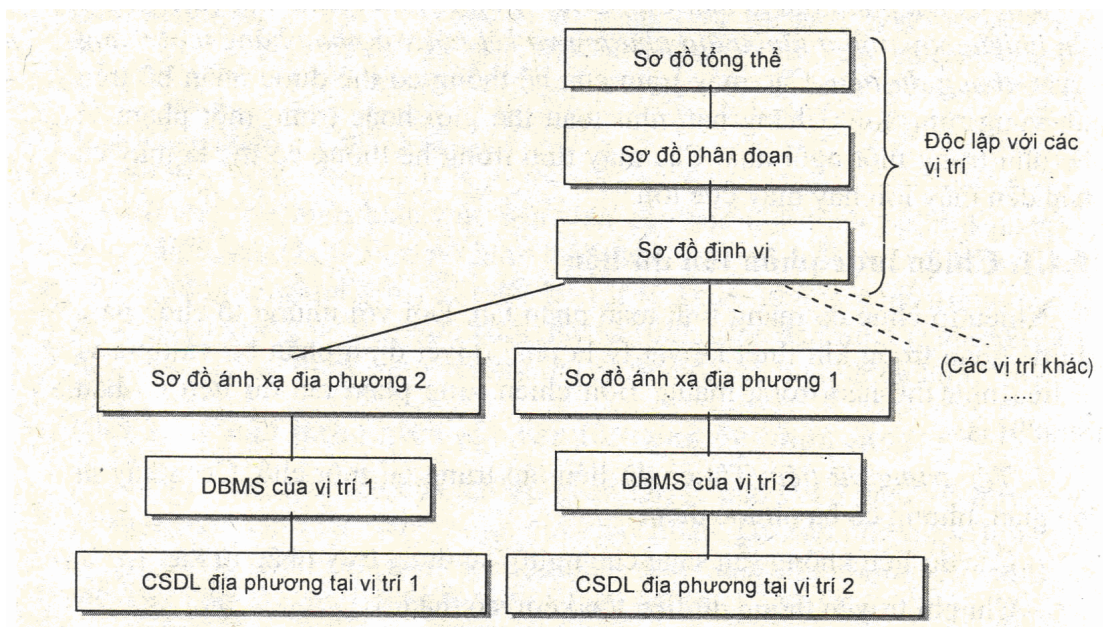
*Nhược điểm của CSDL phân tán:*

- Thiết kế CSDL phức tạp hơn.

- Khó điều khiển tính nhất quán dữ liệu.
- Khó phát hiện và khử lỗi.
- Giá thành cao.
- Thiếu chuẩn mực.
- Thiếu kinh nghiệm.
- Vấn đề về bảo mật.

## 2.KIẾN TRÚC CỦA MỘT HỆ CSDL PHÂN TÁN:

Do sự đa dạng, không có kiến trúc nào được công nhận tương đương với kiến trúc 3 mức ANSI/SPARC. Sơ đồ dưới đây cho ta một kiến trúc cơ bản (hình 1) để tổ chức cho một CSDL phân tán.



Hình 1. Kiến trúc cơ bản của CSDL phân tán

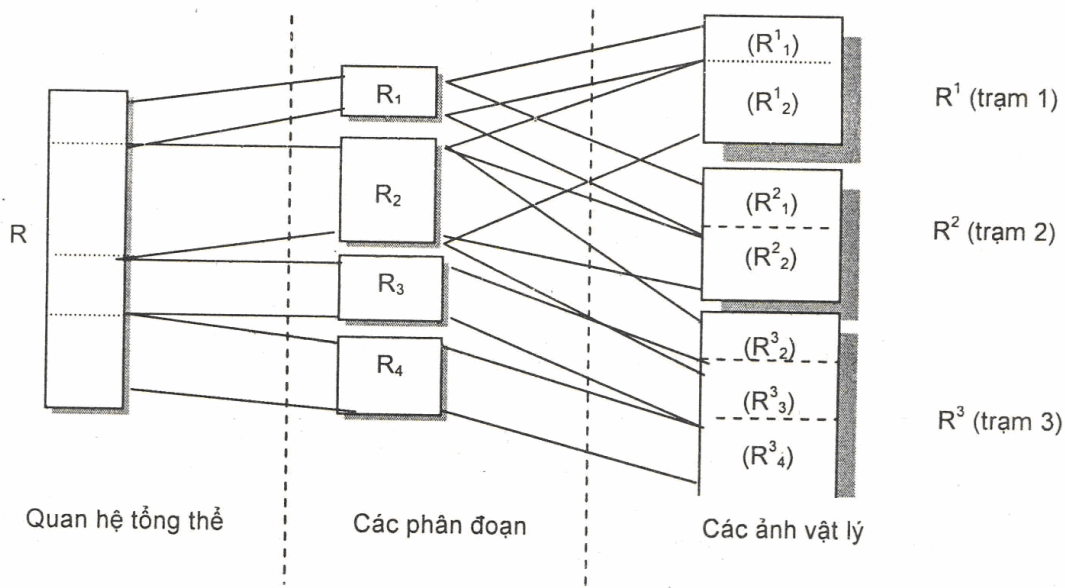
-**Sơ đồ tổng thể:** Sơ đồ này xác định tất cả các dữ liệu sẽ được lưu trữ trong CSDL phân tán. Sơ đồ tổng thể có thể được định nghĩa một cách chính xác theo cách như trong CSDL không phân tán. Ở đây sẽ sử dụng mô hình quan hệ để hình thành nên sơ đồ này. Sử dụng mô hình này, sơ đồ tổng thể bao gồm định nghĩa của một tập các quan hệ tổng thể.

-**Sơ đồ phân đoạn:** Mỗi quan hệ tổng thể có thể chia thành một vài phần nhỏ hơn không giao nhau được gọi là đoạn (*fragments*). Có nhiều cách khác nhau để thực hiện việc phân chia này. Sơ đồ tổng thể mô tả các ánh xạ giữa các quan hệ tổng thể và các đoạn được định nghĩa trong sơ đồ phân đoạn. Ánh xạ này là một- nhiều. Có thể có nhiều đoạn liên kết tới một quan hệ tổng thể, nhưng mỗi đoạn chỉ liên kết tới nhiều nhất là một quan hệ tổng thể. Các đoạn được chỉ ra bằng tên của quan hệ tổng thể cùng với tên của *chỉ mục đoạn*.

-**Sơ đồ định vị:** Các đoạn là các phần logic của một quan hệ tổng thể được định vị trên một hoặc nhiều vị trí vật lý trên mạng. Sơ đồ định vị xác định đoạn nào ở các trạm nào. Lưu ý rằng, kiểu ánh xạ được định nghĩa trong sơ đồ định vị quyết định CSDL phân tán là dư thừa hay không. Tất cả các đoạn liên kết với cùng một quan hệ tổng thể R và được định vị tại cùng một trạm j cấu thành ảnh vật lý của quan hệ tổng thể R tại trạm j. Bởi vậy, có thể ánh xạ một-một giữa một ảnh vật lý và một cặp (quan hệ tổng thể, trạm). Các ảnh vật lý có thể được chỉ ra bằng tên của một quan hệ tổng thể và một *chỉ mục trạm*.

Ví dụ: ký hiệu  $R_i$  chỉ tới đoạn thứ i của quan hệ tổng thể R (hình 2).

Ký hiệu  $R^j$  là ảnh vật lý của quan hệ tổng thể R tại trạm j (hình 2).



Hình 2. Các đoạn và các ảnh vật lý của một quan hệ tổng thể

-*Sơ đồ ánh xạ địa phương*: Ánh xạ các ảnh vật lý tới các đối tượng được các hệ quản trị CSDL địa phương thao tác tại các trạm. Ánh xạ này phụ thuộc vào các hệ quản trị CSDL địa phương. Do vậy, trong một hệ thống không đồng nhất, phải có các kiểu ánh xạ địa phương khác nhau tại các trạm khác nhau.

### **3.HỆ QUẢN TRỊ CSDL PHÂN TÁN (HQTCSDLPT) VÀ PHÂN LOẠI:**

Hệ quản trị CSDL phân tán cung cấp công cụ như tạo lập và quản lý CSDL phân tán. HQTCSDLPT có chức năng hỗ trợ việc tạo và bảo trì CSDL phân tán, chúng có các thành phần tương tự như một hệ quản trị CSDL tập trung và các thành phần hỗ trợ trong việc chuyển tải dữ liệu đến các trạm và ngược lại.

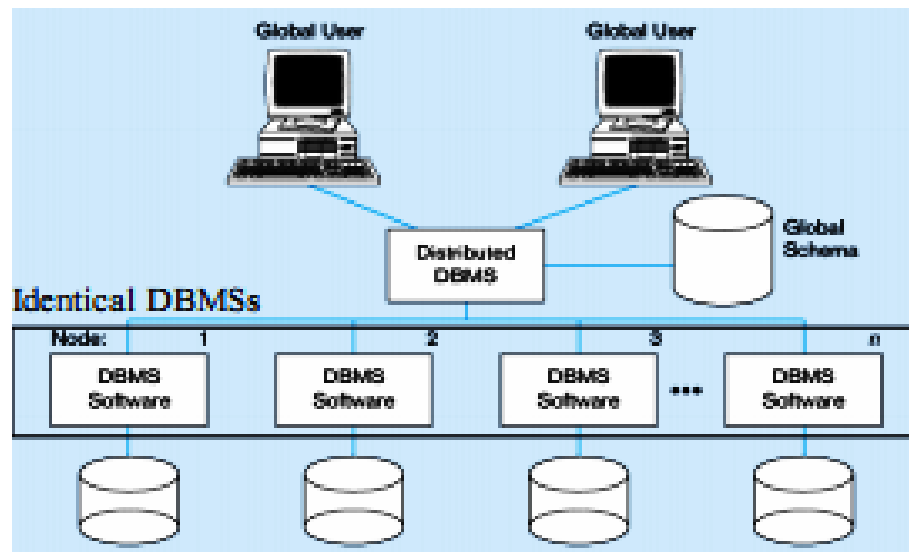
Các thành phần sau đây đòi hỏi một HQTCSDLPT thương mại phải có:

- Quản trị dữ liệu (*database management*): DB
- Truyền thông dữ liệu (*data communication*): DC
- Từ điển dữ liệu (*data dictionary*): DD dùng để mô tả thông tin về sự phân tán của dữ liệu trên mạng.
- Cơ sở dữ liệu phân tán (*distributed database*): DDB

#### **3.1.CÁC LOẠI CSDL PHÂN TÁN:**

-*CSDL phân tán thuần nhất*:

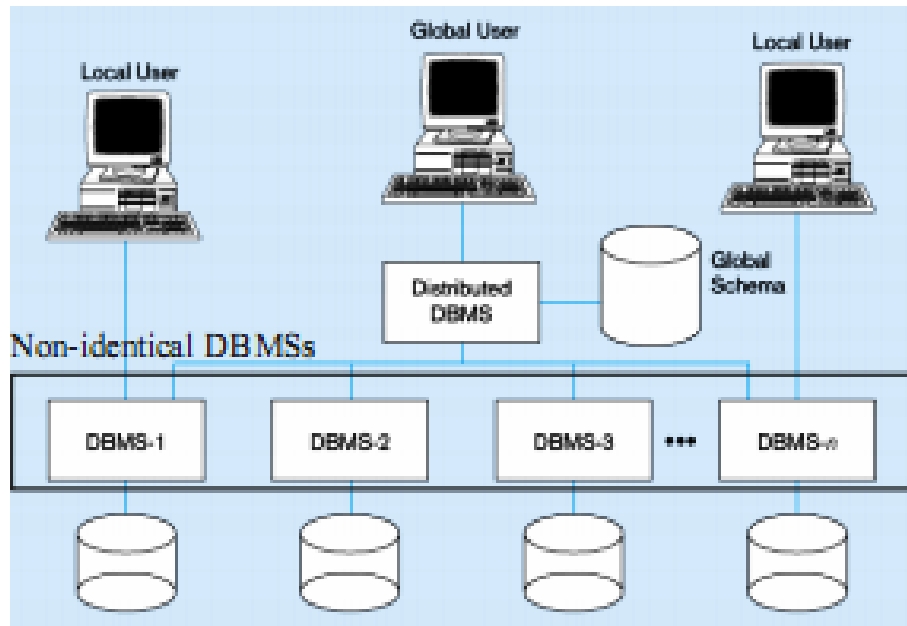
- +Tất cả các nút cùng sử dụng một loại hệ quản trị CSDL.
- +Thuận lợi cho việc tăng trưởng và cho phép nâng cao hiệu năng.
- +Có được bằng cách chia một CSDL thành một tập CSDL cục bộ.



Hình 3. CSDL phân tán thuần nhất

-CSDL phân tán hỗn tạp:

- +Các nút có thể thực hiện trên các hệ quản trị CSDL khác nhau.
- +Xảy ra khi các nút đã cài đặt CSDL riêng.
- +Có được bằng cách tích hợp các CSDL cục bộ đã có.



Hình 4. CSDL phân tán hỗn tạp

## 4.KĨ THUẬT PHÂN ĐOẠN, SAO BẢN VÀ ĐỊNH VỊ DỮ LIỆU:

Phân hoạch cơ sở dữ liệu thành các đoạn (*fragments*): sự phân đoạn cho phép phân chia một đối tượng đơn lẻ thành hai hay nhiều mảnh. Thông tin phân đoạn được lưu trữ trong catalog dữ liệu phân tán. Phần mềm xử lý giao tác sẽ truy nhập thông tin ở đây để xử lý các yêu cầu của người dùng.

Việc chia quan hệ tổng thể thành các đoạn có thể thực hiện bằng cách áp dụng các kiểu phân đoạn sau:

- Phân đoạn ngang.
- Phân đoạn dọc.
- Phân đoạn hỗn hợp: là sự kết hợp giữa phân đoạn ngang và phân đoạn dọc.

### A. CÁC ĐIỀU KIỆN RÀNG BUỘC TRONG THIẾT KẾ PHÂN ĐOẠN:

Một phương pháp thiết kế phân đoạn đúng đắn phải thỏa mãn ba ràng buộc sau:



-*Tính đầy đủ*: Toàn bộ dữ liệu của quan hệ tổng thể phải được ánh xạ vào các đoạn quan hệ và ngược lại. Điều này có nghĩa là, không tồn tại một mục dữ liệu nào thuộc vào quan hệ tổng thể mà không thuộc vào bất kỳ một đoạn nào.

-*Xây dựng lại*: Quan hệ tổng thể có thể được xây dựng lại từ các đoạn mà nó đã tách ra. Điều kiện này là hiển nhiên, bởi vì trong thực tế chỉ có các đoạn được lưu trữ trong CSDL phân tán, và quan hệ tổng thể phải được xây dựng lại thông qua các đoạn khi cần thiết.

-*Tính rời nhau*: các đoạn được tách ra từ quan hệ tổng thể phải là rời nhau. Vì vậy, việc tạo các bản sao phải rõ ràng với các đoạn được chia. Tuy nhiên, điều kiện này chỉ áp dụng chính vào việc phân đoạn ngang, trong khi việc phân đoạn dọc nhiều khi vẫn được phép vi phạm điều kiện này.

## B. CÁC PHƯƠNG PHÁP PHÂN ĐOẠN:

### Phân đoạn ngang:

*Phân đoạn ngang (horizontal fragmentation partitioning)* là tách quan hệ tổng thể R thành các tập con  $R_1, R_2, \dots, R_n$ . Mỗi tập con chứa một số  $n$ \_bộ của R, điều này rất hữu ích trong CSDL phân tán, nơi mỗi tập con bao gồm các dữ liệu có các thuộc tính địa lý chung. Mỗi  $n$ \_bộ thuộc vào một trong các thành phần để có thể khôi phục được quan hệ tổng thể R khi cần thiết. Việc khôi phục quan hệ R được thực hiện bằng phép hợp các quan hệ:

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

Ta có thể sử dụng các phép toán chọn lọc dựa trên quan hệ tổng thể để tách quan hệ tổng thể thành các đoạn.

Ví dụ, giả sử có quan hệ sau:

SINH VIÊN( mã, họ\_tên, ngày\_sinh, tỉnh)

Ta có thể phân thành hai phân đoạn ngang như sau:

SINH VIÊN 1 =  $SL_{\text{tỉnh} = \text{"Hải Phòng"}}$  SINH VIÊN

SINH VIÊN 2 =  $SL_{\text{tỉnh} = \text{"Hà Nội"}}$  SINH VIÊN

Ta có các tập:

SINHVIENT

001	Văn Hưng	21/3/78	Hà Nội
002	Thế Anh	22/4/79	Hải Phòng
003	Thị Hoa	10/2/78	Hải Phòng
004	Văn Bảo	13/5/79	Hà Nội
005	Ngọc Hùng	19/7/79	Hà Nội

**SINHVIEN1**

002	Thế Anh	22/4/79	Hải Phòng
003	Thị Hoa	10/2/78	Hải Phòng

**SINHVIEN2**

001	Văn Hưng	21/3/78	Hà Nội
004	Văn Bảo	13/5/79	Hà Nội
005	Ngọc Hùng	19/7/79	Hà Nội

Điều kiện xây dựng lại được đảm bảo:

$$\text{SINHVIEN} = \text{SINHVIEN1} \cup \text{SINHVIEN2}$$

Dễ dàng thấy rằng, các phân đoạn trên thỏa mãn điều kiện tách rời nhau và đầy đủ. Trong thực tế, nhiều khi cần có các đoạn mà n\_bộ của nó thuộc vào nhiều đoạn khác nhau.

Ví dụ, có quan hệ tổng thể sau:

GIÁO VIÊN (mã\_số, họ\_tên, giới\_tính, quê\_quán)

Do mỗi giáo viên thuộc vào một trường, các trường khác nhau có cách quản lý nhân viên của mình khác nhau, nên việc phân chia các nhân viên thuộc các trường khác nhau vào các quan hệ khác nhau là điều cần thiết. Tuy nhiên, trường không phải là một thuộc tính của quan hệ GIÁO VIÊN, nó là thuộc tính của quan hệ:

NHÂN VIÊN 1 = SELECT<sub>trường = "Đông Đa"</sub> NHÂN VIÊN

NHÂN VIÊN 2 = SELECT<sub>trường = "GIẢNG VỐ"</sub> NHÂN VIÊN

Bởi vậy, các phân đoạn của quan hệ GIÁO VIÊN có thể được định nghĩa như sau:

GIÁO VIÊN 1 = GIÁO VIÊN SJ<sub>mã\_số = mã\_số</sub> NHÂN VIÊN 1

GIÁO VIÊN 2 = GIÁO VIÊN SJ<sub>mã\_số = mã\_số</sub> NHÂN VIÊN 2

Ở trên đã sử dụng phép toán nối giữa các quan hệ GIÁO VIÊN và NHÂN VIÊN 1, NHÂN VIÊN 2.

Ta có thể mô tả điều kiện tham chiếu đầy đủ của hai đoạn trên là:

q<sub>1</sub>: GIÁO VIÊN.mã\_số = NHÂN VIÊN.mã\_số AND

NHÂN VIÊN.trường = “Đồng Đa”

q<sub>2</sub>: GIÁO VIÊN.mã\_số = NHÂN VIÊN.mã\_số AND

NHÂN VIÊN.trường = “Giảng Võ”

Điều kiện xây dựng lại của quan hệ tổng thể GIÁO VIÊN có thể thực hiện được thông qua phép toán hợp như đã nêu ra ở quan hệ sinh viên ở trên.

### **Phân đoạn dọc:**

*Phân đoạn dọc (vertical fragmentation partitioning)* của một quan hệ tổng thể là việc chia các thuộc tính của nó thành các nhóm. Điều này là hữu ích trong các CSDL phân tán, khi mà mỗi nhóm các thuộc tính có thể chứa dữ liệu có chung những thuộc tính địa lý.

Việc phân đoạn là đúng đắn nếu như mỗi thuộc tính của quan hệ tổng thể được ánh xạ thành thuộc tính của ít nhất một đoạn con, và phải đảm bảo điều kiện là quan hệ tổng thể có thể được xây dựng lại từ các phân đoạn mà nó chia ra bằng các phép kết nối.

Phân rã theo chiều dọc quan hệ tổng thể R thành các quan hệ R<sub>i</sub>:

R<sub>i</sub> = Chiếu<sub>thuộc tính R<sub>i</sub></sub> (R)

Quan hệ ban đầu được khôi phục nhờ các phép kết nối tự nhiên:

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

Việc đưa các khóa của quan hệ tổng thể vào trong mỗi đoạn được tách ra là cách dễ nhất để có thể xây dựng lại quan hệ tổng thể bằng các phép toán kết nối.

Ví dụ, cho quan hệ tổng thể:

NHÂN VIÊN (*mã\_NV, tên\_NV, lương, thuế\_TN, phòng, bộ\_phận*)

Một phân đoạn dọc của quan hệ trên có thể được định nghĩa như sau:

NHÂN VIÊN 1 = PROJECTION<sub>mã\_NV, tên\_NV, phòng, bộ\_phận</sub> (NHÂN VIÊN)

NHÂN VIÊN 2 = PROJECTION<sub>mã\_NV, lương, thuế\_TN</sub>(NHÂN VIÊN)

Việc xây dựng lại quan hệ tổng thể NHÂN VIÊN có thể được thể hiện như sau:

NHÂN VIÊN = NHÂN VIÊN 1 JOIN<sub>mã\_NV = mã\_NV</sub> (NHÂN VIÊN 2)

Quan hệ tổng thể NHÂN VIÊN được xây dựng lại nhờ việc kết nối hai phân đoạn NHÂN VIÊN 1 và NHÂN VIÊN 2 dựa vào khóa chính mã\_NV.

Trong thực tế, các phân đoạn dọc nhiều khi chứa cùng một số các thuộc tính (không phải thuộc tính khóa) như nhau trong các phân đoạn khác nhau, ta có thể loại bỏ các thuộc tính này khi xây dựng lại quan hệ tổng thể.

Ví dụ, với quan hệ tổng thể NHÂN VIÊN ở trên, ta có thể phân thành hai phân đoạn dọc sau:

NHÂN VIÊN 1 = PROJECTION<sub>mã\_NV, tên\_NV, phòng, bộ\_phận</sub> (NHÂN VIÊN)

NHÂN VIÊN 2 = PROJECTION<sub>mã\_NV, tên\_NV, lương, thuế\_TN</sub> (NHÂN VIÊN)

Thuộc tính tên\_NV được lặp lại trong cả hai phân đoạn. Việc xây dựng lại quan hệ tổng thể NHÂN VIÊN được thể hiện như sau:

NHÂN VIÊN = NHÂN VIÊN 1 JOIN<sub>mã\_NV = mã\_NV</sub>

OJECTION<sub>mã\_NV, lương, thuế\_TN</sub> (NHÂN VIÊN 2)

## Phân đoạn hỗn hợp

Ví dụ: Cho quan hệ tổng thể sau:

NHÂN VIÊN (mã\_NV, *tên\_NV, lương, thuế\_TN, phòng, bộ\_phận*)

Sau đây là một phân đoạn hỗn hợp (*hybrid fragmentation partitioning*), thu được bằng áp dụng các phân đoạn ngang dựa trên các phân đoạn dọc ở ví dụ trước.

NHÂN VIÊN 1 = SL<sub>mã\_NV <=10</sub> PJ<sub>mã\_NV, tên\_NV, phòng, bộ\_phận</sub> (NHÂN VIÊN)

NHÂN VIÊN 2 = SL<sub>mã\_NV <=20</sub> PJ<sub>mã\_NV, tên\_NV, phòng, bộ\_phận</sub> (NHÂN VIÊN)

NHÂN VIÊN 3 = SL<sub>mã\_NV > 20</sub> PJ<sub>mã\_NV, tên\_NV, phòng, bộ\_phận</sub> (NHÂN VIÊN)

$NHÂN\ VIÊN\ 4 = PJ_{mã\_NV, tên\_NV, lương, thuế\_NV} (NHÂN\ VIÊN)$

Chú ý: các ký hiệu ở đây là :

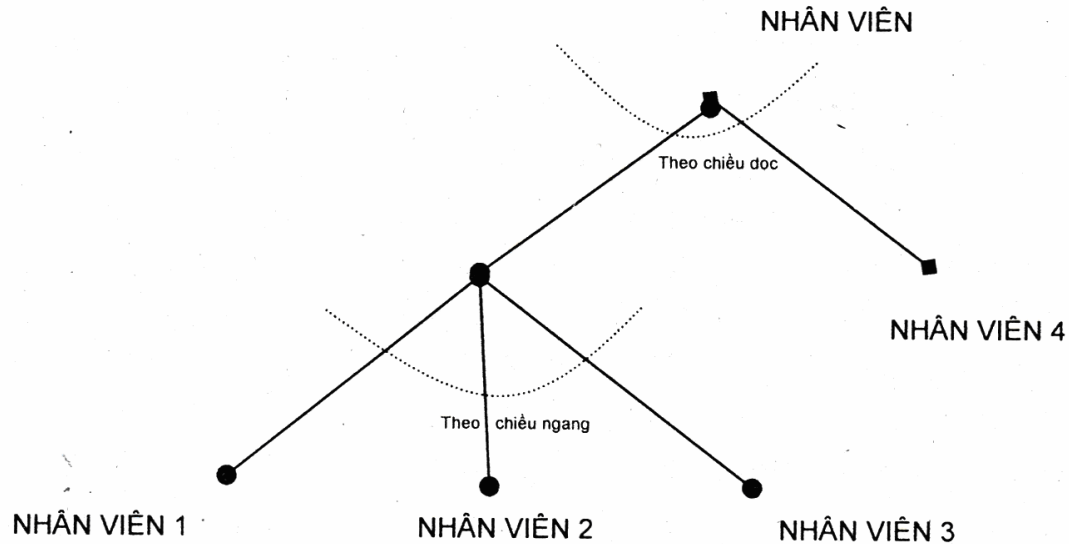
SL = SELECT;                      PJ = PROJECTION

JN = JOIN;                          UN = UNION

Việc xây dựng lại quan hệ tổng thể có thể được mô tả như sau:

$EMP = UN(EMP_1, EMP_2, EMP_3) JN_{EmpNum = EmpNum}$   
 $(PJ_{EmpNum, Name, Sal, Tax}(EMP_4))$

Phân đoạn hỗn hợp có thể được trình diễn qua cây phân đoạn (hình 5). Trong cây phân đoạn, gốc của cây tương ứng với quan hệ tổng thể, các mức tương ứng với các phân đoạn, và các nút ở giữa tương ứng với các kết quả trung gian trong quá trình phân đoạn. Tập hợp các nút con của một nút trình diễn sự phân đoạn của nút này bằng một thao tác phân đoạn (có thể là ngang hoặc dọc).



Hình 5. Cây phân đoạn của quan hệ NHÂN VIÊN.

### C.SAO LẬP DỮ LIỆU:

CSDL được sao thành nhiều bản sao từng phần, hay đầy đủ và được đặt ở hai hay nhiều vị trí trên mạng. Nếu bản sao của CSDL được lưu trữ tại mọi trạm, ta có

trường hợp sao *lập đầy đủ*. Phương thức này làm cực đại việc truy nhập tới dữ liệu ở mọi địa phương. Tuy nhiên, phương thức này nảy sinh nhiều vấn đề khi cập nhật (khi có thay đổi dữ liệu ở một cơ sở thì cần được xử lý lại và đồng bộ hóa dữ liệu cho tất cả các vị trí khác). Một kỹ thuật mới hơn cho phép tạo các bản sao không đầy đủ phù hợp với yêu cầu dữ liệu mỗi trạm lưu trữ và một bản sao đầy đủ ở máy dịch vụ. Sau mỗi thời gian, các bản sao được làm đồng bộ với bản chính ở máy dịch vụ bằng một công cụ phần mềm nào đó. Phần mềm *Briefcase* là một công cụ của Microsoft cho phép thực hiện dịch vụ này trên CDSL Access.

## **D. ĐỊNH VỊ DỮ LIỆU:**

Định vị dữ liệu 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 định vị (*location transparency*) người lập trình sẽ biết được vị trí của các đoạn trên các trạm.

## **5. TÍNH TRONG SUỐT CỦA HQTCSDLPT:**

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:

- *Trong suốt phân đoạn (fragmentation transparency)*
- *Trong suốt về vị trí (location transparency)*
- *Trong suốt ánh xạ địa phương (local mapping transparency)*
- *Trong suốt nhân bản (replication transparency)*
- *Không trong suốt (no transparency)*

### **A. TRONG SUỐT PHÂN ĐOẠN (FRAGMENTATION TRANSPARENCY):**

Trong suốt phân đoạn: là cấp độ cao nhất của mức độ trong suốt, người sử dụng hoặc chương trình ứng dụng chỉ làm việc trên các quan hệ của cơ sở dữ liệu.

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)

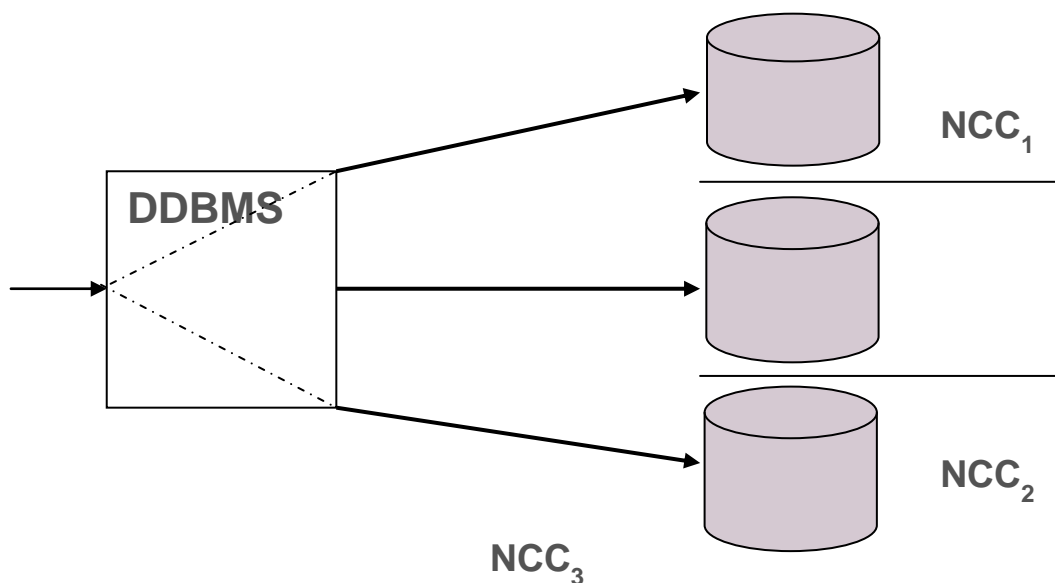
NCC2 (Id, Tên, Tuổi)

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 không.

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



## 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.

*Ví dụ:* Với quan hệ tổng thể R và các phân đoạn như đã nói ở trên nhưng giả sử rằng DDBMS 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**

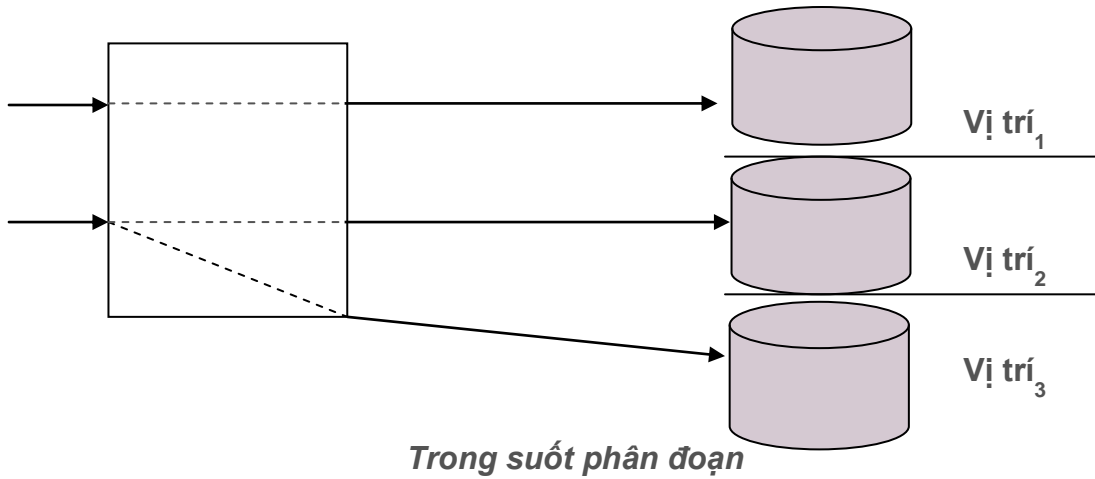
**SELECT \***

**FROM NCC2**

**WHERE Id="Id1"**

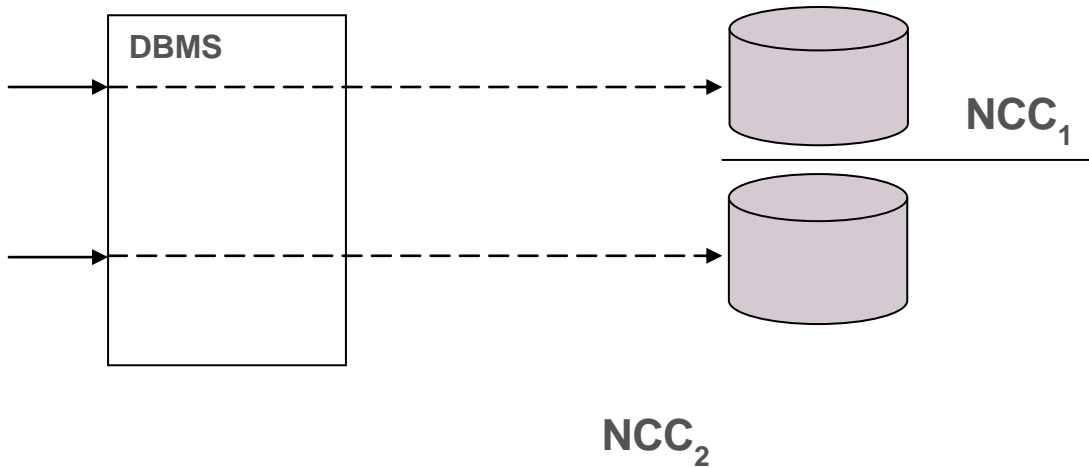
- Đầu tiên hệ thống sẽ thực hiện tìm kiếm ở phân đoạn NCC<sub>1</sub> và nếu DBMS trả về biến điều khiển #FOUND thì một câu lệnh truy vấn tương tự được thực hiện trên phân đoạn NCC<sub>2</sub>,...
- Ở đây quan hệ NCC<sub>2</sub> được sao làm hai bản trên hai vị trí<sub>2</sub> và vị trí<sub>3</sub>, ta chỉ cần tìm thông tin trên quan hệ NCC<sub>2</sub> mà không cần quan tâm nó ở vị trí nào.





### 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.



#### D. TRONG SUỐT NHÂN BẢN (REPLICATION TRANSPARENCY):

Mức trong suốt bản sao liên quan chặt chẽ tới mức trong suốt định vị. Mức trong suốt bản sao có nghĩa là người sử dụng không biết bản sao của đoạn đặt ở vị trí nào. Mức trong suốt bản sao tương đương mức trong suốt định vị. Tuy nhiên, trong những trường hợp thực tế người sử dụng không có mức trong suốt định vị nhưng lại có mức trong suốt bản sao.

### 6. THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN:

#### A. CÁC BƯỚC THIẾT KẾ CSDL:

##### Thiết kế CSDL tập trung

- *Thiết kế sơ đồ khái niệm*: mô tả cơ sở dữ liệu đã hợp nhất (mọi dữ liệu được sử dụng bởi ứng dụng cơ sở dữ liệu).

- *Thiết kế cơ sở dữ liệu vật lý*: tham chiếu từ lược đồ khái niệm tới vùng lưu trữ và xác định các cách thức truy cập khác nhau.

##### Thiết kế CSDL phân tán

- *Thiết kế lược đồ toàn cục*

- *Thiết kế phân đoạn*: xác định cách thức phân chia những quan hệ toàn bộ thành những đoạn dữ liệu theo chiều dọc, chiều ngang và kiểu hỗn hợp.

- *Thiết kế cấp phát đoạn dữ liệu*: xác định cách thức đoạn dữ liệu tham khảo đến ảnh vật lý nào và cũng xác định các bản sao của đoạn dữ liệu.

- *Thiết kế CSDL vật lý cục bộ*

Công việc thiết kế chương trình ứng dụng được thực hiện sau khi thiết kế sơ đồ và có kiến thức về yêu cầu của chương trình ứng dụng. Thiết kế sơ đồ để có khả năng cung cấp hiệu quả các chương trình ứng dụng. Vì vậy trong thiết kế cơ sở dữ liệu phân tán, hiểu biết rõ ràng và đầy đủ về yêu cầu của chương trình ứng dụng là cần thiết đối với chương trình ứng dụng quan trọng. Những công việc thiết kế được thực hiện thường xuyên để công việc thực hiện của thiết kế đúng đắn. Những yêu cầu trong chương trình ứng dụng:

- Vị trí nơi chương trình ứng dụng được đưa ra (cũng gọi là vị trí cơ sở của chương trình ứng dụng).

- Tính thường xuyên hoạt động của chương trình ứng dụng: số lần yêu cầu của chương trình ứng dụng trong một khoảng thời gian. Trường hợp thông thường chương trình ứng dụng có thể được đưa ra ở nhiều vị trí khác nhau vì vậy phải biết tần suất hoạt động của chương trình ứng dụng tại mỗi vị trí.

- Số lượng, kiểu và phân tán thống kê các lần truy cập đối với mỗi đối tượng dữ liệu được yêu cầu bởi các chương trình ứng dụng.

## **B. ĐỐI TƯỢNG THIẾT KẾ CỦA CƠ SỞ DỮ LIỆU PHÂN TÁN:**

Trong thiết kế phân tán dữ liệu, những đối tượng sau đây được quan tâm:

*Tiến trình địa phương*: phân tán dữ liệu để cực đại hoá tiến trình địa phương hay tăng thời gian bộ xử lý trung tâm cho tiến trình địa phương tương ứng với nguyên tắc là đơn giản hoá công việc: đặt dữ liệu gần chương trình ứng dụng thường xuyên sử dụng dữ liệu đó. Thực hiện được tiến trình ở địa phương là mục đích chính của cơ sở dữ liệu phân tán.

Thiết kế phân tán dữ liệu để cực đại hoá tiến trình ở địa phương có thể thực hiện qua việc thêm vào một số tham chiếu địa phương và tham chiếu từ xa tương ứng với mỗi phân đoạn.

Một cách mở rộng tầm quan trọng của việc tối ưu hoá là đưa vào một số mục tiêu khi chương trình ứng dụng có tính địa phương hoàn toàn. Thuật ngữ này để chỉ rõ những chương trình ứng dụng này có thể hoàn toàn thực hiện ở vị trí cơ sở. Tiềm lợi chính của tính địa phương không chỉ hoàn toàn là việc giảm công việc truy cập từ xa mà bên cạnh đó cũng làm tăng tính đơn giản trong điều khiển thực hiện chương trình ứng dụng.

*Tính sẵn có và dễ hiểu của dữ liệu phân tán:* Cấp độ cao của tính sẵn có đối với chương trình ứng dụng có thuộc tính chỉ đọc đạt được qua việc lưu trữ các bản sao của cùng một thông tin. Hệ thống phải chuyển sang một bản sao khác khi một bản sao nào đó truy cập ở điều kiện không bình thường hay bản sao đó không có sẵn.

Tính dễ hiểu cũng đạt được khi lưu trữ nhiều bản sao của cùng một thông tin khi cho phép hồi phục từ những hỏng hóc hay từ những phá hủy về mặt vật lý của cùng một bản sao bằng cách dùng một bản sao khác (khi vẫn tồn tại các bản sao khác). Hỏng hóc trong máy tính thường là những sự kiện xảy ra mà không còn khả năng hồi phục lại. Như vậy việc đặt nhiều bản sao ở trên những vị trí địa lý rải rác khắp nơi là giải pháp hợp lý.

*Phân chia khối lượng công việc:* Phân tán công việc cho những vị trí là đặc điểm quan trọng của hệ thống máy tính phân tán. Việc phân chia công việc cũng nhằm mục đích đạt được tiềm lợi về khả năng hoặc tiện ích máy tính ở mỗi vị trí trên mạng và cũng để tăng cấp độ thực hiện song song của chương trình ứng dụng. Khi phân chia khối lượng công việc có thể ảnh hưởng xấu đến tiến trình xử lý địa phương và cần thiết cân nhắc đến lợi hại trong thiết kế dữ liệu phân tán.

*Giá cả thiết bị lưu trữ và tính sẵn có:* Phân tán dữ liệu có thể phản ánh giá cả và tính sẵn có của thiết bị lưu trữ ở các vị trí khác nhau. Tuy nhiên có thể có những điểm đặc biệt trong mạng để lưu trữ dữ liệu hoặc có những điểm làm việc không cung cấp kho dữ liệu. Giá cả của thiết bị lưu trữ không thể thích hợp khi so sánh với CPU - đơn vị xử lý trung tâm, thiết bị vào ra và giá cả truyền thông của chương trình ứng dụng, do đó giới hạn tính sẵn có của thiết bị lưu trữ ở mỗi vị trí phải được cân nhắc.

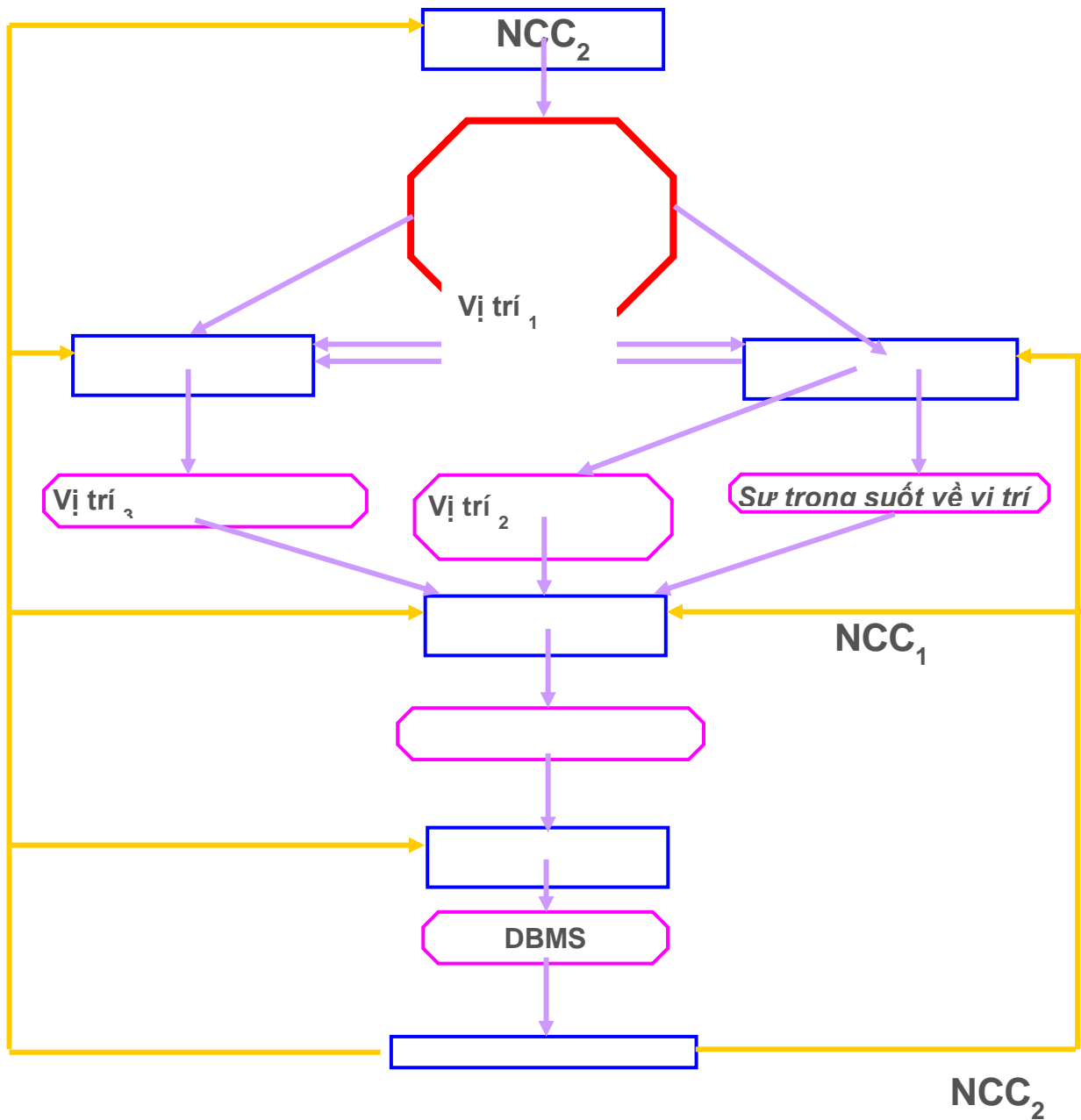
## **C. CÁC HƯỚNG TIẾP CẬN THIẾT KẾ CSDL:**

Có hai phương pháp thiết kế là hướng thiết kế Top-down và Bottom-up.

### **Thiết kế từ trên xuống – Top-down design:**

- Thiết kế lược đồ toàn cục

- Thiết kế phân mảnh CSDL
- Định vị các mảnh tại các nơi
- Thiết kế dữ liệu vật lý đặt tại mỗi nơi



### Thiết kế từ dưới lên – Bottom-up design:

- Chọn một mô hình CSDL chung để mô tả lược đồ toàn cục của CSDL
- Chuyển đổi mỗi lược đồ cục bộ thành mô hình dữ liệu chung

- Tích hợp các lược đồ cục bộ thành một lược đồ toàn cục chung

## **D. THIẾT KẾ PHÂN ĐOẠN CƠ SỞ DỮ LIỆU:**

Hầu hết các hệ cơ sở dữ liệu phân tán được tạo mới cho nên chúng ta chỉ đề cập đến thiết kế theo hướng Top down. Thiết kế phân đoạn là công việc đầu tiên phải giải quyết trong thiết kế Top-down cơ sở dữ liệu phân tán. Mục đích của việc phân đoạn để nhận ra những đoạn không trùng nhau (đoạn như vậy được gọi là đơn vị cấp phát logic).

Rõ ràng, các bộ hoặc các thuộc tính của quan hệ không thể được xem như một đơn vị cấp phát vì sẽ làm cho việc cấp phát trở lên phức tạp hơn. Thiết kế phân đoạn bao gồm công việc nhóm các bộ trong trường hợp phân đoạn ngang hay nhóm các thuộc tính trong trường hợp phân đoạn dọc có cùng đặc tính theo quan điểm cấp phát. Mỗi nhóm các bộ hoặc thuộc tính có cùng đặc tính hay tính chất để thành lập một đoạn. ý cơ bản là nếu bất cứ hai đơn vị cấp phát nào của cùng một đoạn thì có cùng đặc tính theo quan điểm cấp phát. Mỗi cách thức dùng để cấp phát dữ liệu sẽ chọn lựa các đoạn này với nhau. Vì vậy các đoạn hình thành nhờ các phương pháp này tạo ra các đơn vị cấp phát khác nhau.

Điểm chú ý ở trong công việc phân đoạn là vị trí cơ sở của mỗi chương trình ứng dụng phải thích hợp để nhận biết đặc tính địa phương đối với công việc xác định vị trí thích hợp của đoạn. Vì vậy cần quan niệm rằng chương trình ứng dụng thực hiện ở nhiều vị trí khác nhau có vai trò như các chương trình ứng dụng khác nhau thậm chí ngay cả khi chúng thực hiện cùng chức năng.

Tiếp theo là mô tả công việc phân đoạn ngang, phân đoạn dọc và phân đoạn hỗn hợp (chéo độc lập).

### **• Phân đoạn ngang:**

*Phân đoạn cơ sở:* phân đoạn ngang cơ sở được tìm ra khi sử dụng phép chọn quan hệ. Tính đúng đắn của phân đoạn ngang cơ sở đòi hỏi mỗi bộ của quan hệ được chọn vào một và chỉ một đoạn. Vì vậy việc xác định phân đoạn ngang cơ sở của một quan hệ yêu cầu xác định một tập các vị từ chọn ra không liên kết hoặc hoàn toàn liên kết. Yêu cầu đối với mỗi đoạn là các thành phần của nó phải tham chiếu đồng đều bởi chương trình ứng dụng.

Giả sử  $R$  là một quan hệ cần thiết để tạo ra phân đoạn ngang cơ sở. Phải xác định:

1. Vị từ đơn giản là vị từ có dạng.

Thuộc\_tính = giá\_trị

2. Vị từ minterm  $y$  đối với một tập  $P$  của những vị từ đơn giản là mối liên hệ giữa mọi vị từ xuất hiện trong  $P$ , có thể kiểu liên kết này là khẳng định hay phủ định nhưng không mâu thuẫn nhau. Vì vậy:

$$y = \bigwedge_{p_i \in p} p_i^*$$

với  $p_i^* = p$ ,  $p_i^* = \text{NOT } p_i$  và  $y \neq \text{false}$ .

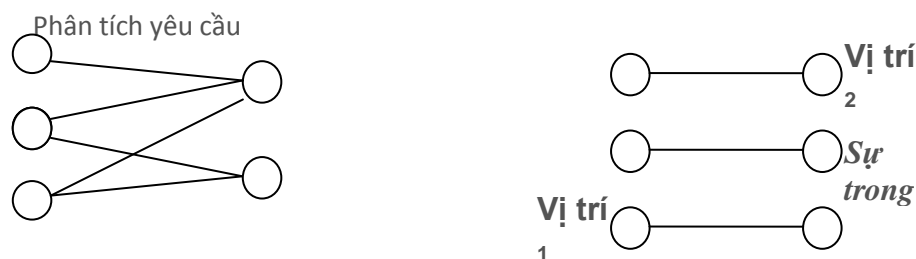
3. Đoạn là một tập các bộ mà vị từ minterm nắm giữ được.

4. Một vị từ đơn  $p_i$  thích hợp với mỗi quan hệ trong tập  $P$  mà biểu thức của các vị từ chỉ khác trong vị từ  $p_i$  (biểu thức xuất hiện trong dạng khẳng định trong một số trường hợp và phủ định ở một số trường hợp khác). Vì vậy các đoạn tương đương được tham chiếu theo nhiều cách bởi ít nhất một chương trình ứng dụng.

- Phân đoạn ngang suy diễn:

Mối liên kết phân tán là liên kết giữa các quan hệ được phân đoạn ngang. Khi chương trình ứng dụng yêu cầu liên kết giữa hai quan hệ  $R$  và  $S$  thì mọi bộ của  $R$  và  $S$  cần được so sánh với nhau. Vì vậy theo nguyên tắc, cần phải so sánh mọi đoạn  $R_i$  của  $R$  với mọi đoạn  $S_j$  của  $S$ . Tuy nhiên một số trường hợp có thể suy diễn một số liên kết bộ phận  $R_i \text{ JOIN } S_j$  là rỗng bên trong. Đối với việc phân tán dữ liệu, trường hợp này xảy ra khi giá trị của thuộc tính liên kết ở đoạn  $R_i$  và  $S_j$  không liên kết với nhau.

Mối liên kết phân tán thể hiện hiệu quả khi sử dụng mô hình liên kết. Mô hình liên kết  $G$  của liên kết phân tán  $R \text{ JOIN } S$  là mô hình  $(N, E)$  khi nút  $N$  thể hiện đoạn của  $R$  và  $S$  và biên không trực tiếp giữa hai nút biểu hiện liên kết giữa hai đoạn không rỗng bên trong. Để đơn giản hóa, không chứa trong  $N$  các đoạn của  $R$  hay  $S$ . Một số ví dụ về mô hình liên kết:



Có thể nói rằng mỗi mô hình liên kết là toàn bộ khi mô hình này gồm mọi biên có thể giữa các đoạn của  $R$  và  $S$ . Liên kết giảm khi không có số biên giữa các đoạn của  $R$  và các đoạn của  $S$ . Hai kiểu giảm liên kết đặc biệt thích hợp là:



-Giảm mô hình liên kết bộ phận nếu mô hình được tạo thành từ hai hay nhiều mô hình con không có biên giữa chúng.

-Giảm mô hình liên kết đơn giản nếu nó là bộ phận và mỗi mô hình liên kết con có một biên.

Xem xét cách phân đoạn tìm được trong mỗi liên hệ này: quan hệ  $R$  có các đoạn là  $R_i$  tìm được từ cách phân đoạn của  $S$  qua liên kết phụ:

$$R_i = R \text{ SJ}_F S_j$$

- **Phân đoạn dọc:**

Xác định phân đoạn dọc của quan hệ  $R$  là chia nhóm các thuộc tính thành tập các thuộc tính để các chương trình ứng dụng tham chiếu đến.

Tuy nhiên phải phân biệt vấn đề chia phần theo chiều dọc và theo cách này các tập không liên kết với nhau từ những vấn đề chia nhóm theo chiều ngang. Điều kiện đúng đối với việc chia theo chiều ngang là đòi hỏi mỗi thuộc tính của quan hệ  $R$  phải ít nhất thuộc về một tập mà mỗi tập gồm có khoá của  $R$  hoặc một tập các minh chứng.

Mục đích của việc phân đoạn dọc là nhận ra các phân đoạn  $R_i$ , như vậy nhiều chương trình ứng dụng có thể được thực hiện thao tác trên dữ liệu mà chỉ cần sử dụng một đoạn.

Công việc xác định cách phân đoạn cho quan hệ  $R$  không dễ dàng vì số lượng những phần có khả năng phát triển có thể kết hợp với số lượng các thuộc tính của quan hệ  $R$  và số lượng các bó có khả năng lớn thêm. Vì vậy thể hiện của quan hệ lớn, theo hướng tiếp cận heuristic cần thiết để xác định các phần hay các nhóm thuộc tính để phân chia hợp lý. Có hai cách phân chia các thuộc tính:

-Hướng chia từ trên xuống: quan hệ được liên tục chia thành các đoạn (không chọn lọc).

-Hướng nhóm từ dưới lên: các thuộc tính liên kết để tạo thành các đoạn (có chọn lọc).

Cả hai hướng có thể được phân loại riêng rẽ như dãy heuristic, các đoạn này được sử dụng để chuyển sang đánh dấu mỗi công việc lặp lại khả năng lựa chọn tốt nhất. Trong cả hai trường hợp sử dụng công thức báo hiệu khả năng chia hay hợp

tốt nhất. Một số kiểu lùi có thể được dùng để thử chuyển thuộc tính từ tập thuộc tính này sang tập thuộc tính còn lại để chia tiếp.

Nhóm các thuộc tính theo chiều dọc đưa ra bản lập một số thuộc tính trong các đoạn qua việc đánh dấu mỗi khả năng chọn tốt nhất những công việc lặp lại. Bản lập lại có ảnh hưởng khác nhau đến chương trình ứng dụng có thuộc tính chỉ đọc và cập nhật. Chương trình ứng dụng có thuộc tính chỉ đọc có lợi thế của bản lập lại vì các chương trình này làm công việc giống như tham chiếu đến dữ liệu ở địa phương. Đối với chương trình ứng dụng có thuộc tính chỉ cập nhật, tạo bản lập lại không thích hợp khi các chương trình này cập nhật mọi bản sao để bảo đảm tính đúng đắn.

### • **Phân đoạn hỗn hợp:**

Cách đơn giản nhất để phân đoạn hỗn hợp gồm có :

- áp dụng phân đoạn ngang cho các đoạn phân chi theo chiều dọc.
- áp dụng phân đoạn dọc cho các đoạn phân chi theo chiều ngang.

Mặc dù các công việc này có thể lặp lại, tạo ra cây phân đoạn phức tạp nào đó, có nghĩa là có hai hay nhiều cấp phân đoạn sinh ra trong thực tế. Hai vấn đề trên cho phép cả hai cách phân đoạn được xem xét mỗi quan hệ và vì vậy không đạt được tiện lợi cần thiết. Cách phân đoạn lần thứ hai có thể áp dụng cho đoạn con từ cách phân đoạn thứ nhất.

### **e. Cấp phát cho các đoạn:**

Cách dễ nhất thực hiện công việc cấp phát file là xem mỗi đoạn như một file riêng rẽ. Tuy nhiên cách này không thích hợp do ba lý do:

- Các đoạn không mô hình hóa thích hợp như các file riêng rẽ vì các đoạn không có cấu trúc như file dẫn đến khó tác động đến đoạn.
- Số đoạn nhiều hơn quan hệ cơ sở như vậy nhiều mô hình phân tích không tính toán được giải pháp cho vấn đề này.
- Mô hình hóa tác động chương trình ứng dụng với hệ thống file rất đơn giản trong khi chương trình ứng dụng ở cơ sở dữ liệu phân tán có thể tạo cách sử dụng dữ liệu dễ dàng.

Một số vấn đề này hiện chưa được giải quyết thỏa đáng, ví dụ như vấn đề thứ 3 đặc biệt khó vì đòi hỏi phải tối ưu chương trình ứng dụng, tái tạo lại quan hệ và nhiều tính toán phức tạp.

## 1. Các chuẩn thông thường của công việc cấp phát cho các đoạn:

Trong các công việc cấp phát cho các đoạn, quan trọng phân biệt được thiết kế cấp phát cho các đoạn dư thừa hay không dư thừa. Cách dễ nhất là hướng “phù hợp nhất”: tiêu chuẩn vị trí kết hợp với khả năng cấp phát cho các đoạn. Hướng này không quan tâm đến ảnh hưởng qua lại của việc đặt một đoạn ở vị trí những đoạn liên quan cũng đặt ở vị trí đó.

Bản lặp lại các đoạn làm phức tạp công việc thiết kế hơn vì:

- Cấp độ những bản sao của mỗi đoạn thích hợp với vấn đề có thể thay đổi thiết kế.

- Mô hình hóa chương trình ứng dụng có thuộc tính chỉ đọc bị làm phức tạp bởi thực tế chương trình ứng dụng có thể chọn một trong số vài vị trí khác nhau để truy cập đến các đoạn.

Để xác định cấp phát đoạn dư thừa thì phải dùng hai cách thức sau:

- Xác định nhóm mọi vị trí có lợi ích dụng cấp phát đoạn và cấp phát bản sao của đoạn cao hơn chi phí và cấp phát các bản sao của đoạn cho các vị trí thành phần của nhóm này. Cách này có nghĩa là lựa chọn các vị trí có lợi nhất.

- Đầu tiên xác định giải pháp của bài toán cấp phát không sao lại các đoạn và sau đó tiếp tục sao lại các bản sao bắt đầu từ nơi có tính chất lợi ích nhất. Tiến trình này được kết thúc khi bản sao không có lợi.

Cả hai phương pháp có một số nhược điểm. Trong phương pháp mọi vị trí có lợi nhất đánh giá chi phí và lợi ích cơ bản cho việc cấp phát các đoạn riêng rẽ hơn trường hợp không dư thừa vì không quan tâm đến tác động qua lại khi cấp phát những bản sao khác nhau của cùng một đoạn. Phương pháp thêm bản lặp lại là hướng tiếp cận theo heuristic theo cách này có thể đưa vào nguyên nhân tăng mức độ dư thừa ít hơn phương pháp tốt nhất. Cả hai tính sẵn có và tính dư thừa của hệ thống tăng nếu có hai bản sao của mỗi đoạn nhưng những bản sao sau này ít tăng theo tỉ lệ.

## 2. Đánh giá mức độ quan trọng về giá trị và lợi ích của công việc cấp phát đoạn:

Công thức đơn giản để đánh giá giá trị và lợi nhuận của công việc cấp phát đoạn cho quan hệ R.

$$n_{kj} = \Gamma k_j + \upsilon k_i$$

Với:  $i$  là đoạn index và  $j$  là vị trí index.

$f_{kj}$  tần xuất sử dụng của chương trình ứng dụng  $k$  tại vị trí  $j$ .

$\Gamma_{kj}$  là số lượng tham chiếu có tính chất hồi phục của chương trình ứng dụng  $k$  sang đoạn  $j$ .

$u_{kj}$  số lượng tham chiếu có thuộc tính cập nhật của chương trình ứng dụng  $k$  sang đoạn  $i$ ;

### **Đối với phân đoạn ngang:**

-Sử dụng hướng phân đoạn phù hợp nhất đối với cấp phát không lặp lại: đặt đoạn  $R$  ở vị trí có số lượng chương trình ứng dụng tham chiếu đến đoạn  $R$  lớn nhất. Đánh giá số lượng tham chiếu của các chương trình ứng dụng ở địa phương tới đoạn  $R_i$  ở vị trí  $j$  được tính theo công thức tổng các tham chiếu hồi phục với tần xuất sử dụng:

$$B_{ij} = \sum_k f_{kj} \cdot \Gamma_{kj}$$

$B_{ij}$  số lượng tham chiếu của các chương trình ứng dụng.

Đoạn  $R_j$  được đặt tại vị trí  $j^*$  với  $B_{ji^*}$  cực đại ( $j^*$  là vị trí lựa chọn) .

-Sử dụng phương pháp đặt đoạn ở mọi vị trí có lợi nhất đối với công việc cấp phát các bản sao: đặt đoạn  $R_j$  ở các vị trí  $j$  có chi phí tham chiếu, hồi phục của chương trình lớn hơn chi phí tham chiếu cập nhật đến đoạn  $R_i$  từ chương trình ở các vị trí khác. Biểu thức đánh giá tính như sau:

$$B_{ij} = \sum_k f_{kj} \cdot \Gamma_{ki} - C \cdot \sum_k \sum_{j' \neq j} f_{kj'} \cdot u_{ki}$$

Với  $C$  là hằng số, hằng số này là tỷ lệ giữa chi phí loại truy cập để cập nhật và loại chi phí để hồi phục . Truy cập mang tính cập nhật đắt hơn khi yêu cầu số lượng lớn thông báo điều khiển và thực hiện nhiều công việc của chương trình ứng dụng ở địa phương ( vì vậy  $C \geq 1$ ).

Đoạn  $R_i$  được cấp phát ở mọi vị trí  $j^*$  với biểu thức đánh giá  $B_{ij^*}$  với giá trị tuyệt đối. Bản sao của đoạn  $R_i$  được cấp phát ở vị trí với biểu thức đánh giá  $B_{ij}$  lớn nhất.

### **3.Sử dụng phương pháp thêm bản $R_i$ đối với cách phân đoạn lặp lại.**

Chỉ có thể đo lợi ích đặt bản sao của đoạn  $R_i$  trong giới hạn khả năng tăng và khả năng sẵn có của hệ thống. Khi bắt đầu công việc cấp phát, lợi ích này không tăng tỷ lệ với độ dư thừa của đoạn  $R_i$ . Giả sử  $d_i$  tương đương với cấp độ dư thừa của đoạn

$R_i$  và giả sử  $F_j$  tương đương với lợi ích với việc có đoạn  $R_j$  đầy lấp lại đủ ở mỗi vị trí. Hàm  $\beta(d_i)$  đo lợi ích này:

$$\beta(d_i) = (1 - 2^{1-d_i})F_i$$

Chú ý  $\beta(1) = 0$ ,  $\beta(2) = F_i/2$ ,  $\beta(3) = 3F_i$ . Sau đó tính toán lợi ích của việc rút ra bản sao mới của  $R_i$  ở vị trí  $j$  qua việc thay đổi công thức ở trường hợp 2 như sau:

$$B_{ij} = \sum_k f_{kj} \cdot \Gamma_{ki} - C \cdot \sum_k \sum_{j' \neq j} f_{kj'} u_{ki} + \beta(d_i)$$

### **Đánh giá số lượng tham chiếu đến phân đoạn theo chiều ngang:**

Để đo lợi ích của các phần phân đoạn ngang của đoạn  $R_i$  đặt ở vị trí thành hai đoạn  $R_s$  và  $R_t$  đặt ở vị trí  $r$  và vị trí  $t$ . Xem xét các tập sau đây:

- Có hai nhóm chương trình ứng dụng  $A_s$  và  $A_t$  sử dụng thuộc tính chỉ từ  $R_s$  và  $R_t$  đoạn của chương trình ứng dụng xử lý ở vị trí  $s$  và  $t$  và đưa ra ở vị trí địa phương  $s$  và  $t$ , mối liên quan giữa các chương trình ứng dụng này là đều tham chiếu từ xa.

- Có một tập  $A_t$  chương trình ứng dụng tại vị trí  $r$  và chỉ sử dụng các thuộc tính của  $R_s$  hoặc  $R_t$ . Các chương trình ứng dụng này cần tạo ra một tham chiếu địa phương từ xa.

- Có một tập  $A_2$  chương trình ứng dụng tại vị trí  $r$  và sử dụng cả các thuộc tính của  $R_s$  và  $R_t$ . Các chương trình ứng dụng này cần tạo ra hai tham chiếu địa phương từ xa.

- Có một tập  $A_3$  chương trình ứng dụng ở các vị trí khác nhau  $r$ ,  $s$  và  $t$  tham chiếu đến cả hai đoạn  $R_s$  và  $R_t$ . Các chương trình ứng dụng này cần tạo một tham chiếu địa phương từ xa

Đo lợi ích này theo công thức sau:

$$B_{tst} = \sum_{k \in A_s} f_{ks} n_{ki} + \sum_{k \in A_t} f_{kt} n_{ki} - \sum_{k \in A_1} f_{kr} n_{ki} + 2 \sum_{k \in A_2} f_{kr} n_{ki} + \sum_{k \in A_3} \sum_{k \notin r, s, t} f_{kj} n_{ki}$$

Để đơn giản hoá, công thức này đếm số lượng các lần truy cập. Phân biệt truy cập hồi phục và cập nhật đưa vào tài khoản các giá trị khác nhau, các giá trị này không hiệu quả khi dùng  $(r_{ki} + C \cdot u_{ki})$  thay vì  $n_{ki}$ .

Công thức này có thể sử dụng trong thuật toán chia nhỏ để xác định có chia  $R_i$  ở vị trí  $i$  vào đoạn  $R_s$  ở vị trí  $s$  và đoạn  $R_t$  ở vị trí  $t$  thích hợp qua việc thử mọi khả năng kết hợp của vị trí  $s$  và  $t$  chú ý trường hợp khi  $r = s$  hay  $r = t$ .

Nhóm theo các thuộc tính chiều dọc:

Đo lợi ích của việc nhóm các thuộc tính của đoạn  $R_{ij}$  theo chiều dọc ở vị trí  $r$  thành hai đoạn đặt ở vị trí  $s$  và  $t$  với thuộc tính lặp là  $l$ . Việc nhóm đòi hỏi các nhóm chương trình ứng dụng tương đương đưa ra từ các phần theo chiều ngang:

+ $A_s$  gồm các chương trình ứng dụng đặt ở vị trí  $s$  vì các chương trình này:

-Đọc các thuộc tính của đoạn  $R_s$  hoặc

-Cập nhật các thuộc tính của  $R_s$  không lặp lại ở thuộc tính lặp  $l$  cũng như đối với  $A_s$ .

+ $A_s$  gồm các chương trình ứng dụng có thuộc tính chỉ đọc một cách hình thức đặt vào vị trí  $R$  một thao tác truy cập vào thuộc tính lặp  $l$  thì thao tác này truy cập đến cả hai đoạn  $R_s$  và  $R_t$ .

+ $A_s$  gồm các chương trình ứng dụng ở các vị trí khác  $r, s$  hay  $t$  cập nhật vào thuộc tính  $l$  thì phải truy cập đến cả hai đoạn  $R_s$  và  $R_t$ .

## 7. XỬ LÝ TRUY VẤN TRONG CSDL PHÂN TÁN:

- **Biểu thức đại số quan hệ**

Biến đổi truy vấn SQL thành các biểu thức đại số quan hệ.

Một biểu thức đại số quan hệ (expression of relational algebra): chuỗi các phép toán (sequence of operations).

Hai biểu thức có cùng ngữ nghĩa có thể mô tả hai chuỗi phép toán khác nhau.

$\Pi$  name, deptnum

$\sigma$  deptnum = 15 (emp)

$\sigma$  deptnum = 15  $\Pi$  name, deptnum (emp)

- **Cây toán tử của truy vấn**

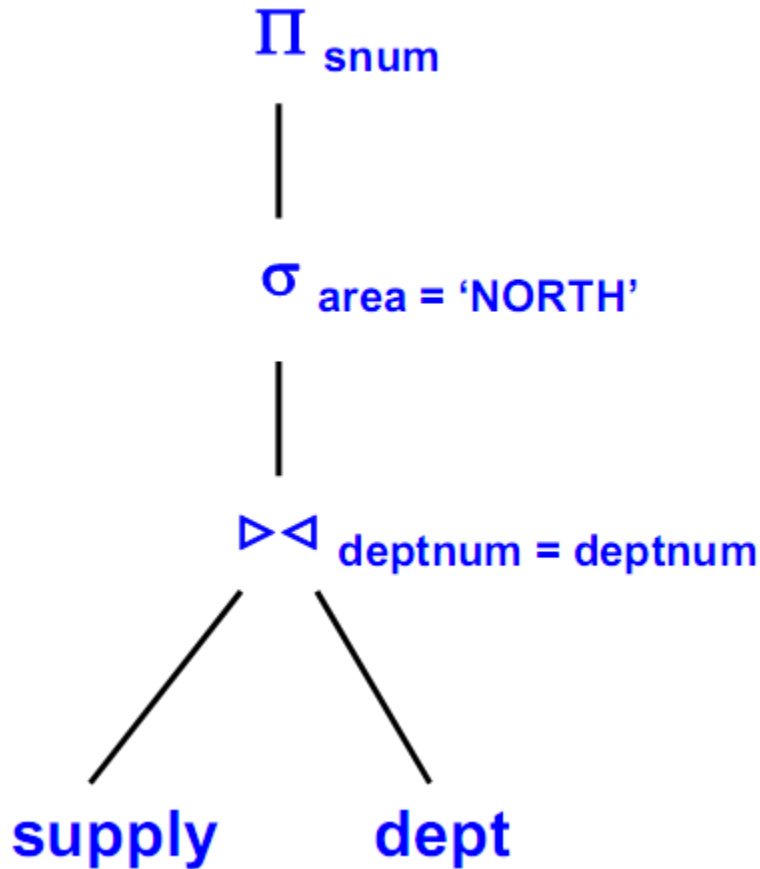
Một truy vấn được biểu diễn bằng cây toán tử (operator tree).

™ Ví dụ

Truy vấn  $Q_1$  – Hãy cho biết mã của các nhà

cung cấp có đơn hàng cung cấp ở phía Bắc.

Q<sub>1</sub>:  $\Pi_{snum} \sigma_{area = 'NORTH'}(supply \bowtie_{deptnum = deptnum} dept)$



- **Các phép biến đổi tương đương**

Hai quan hệ R1 và R2 là tương đương nếu các bộ của chúng biểu diễn cùng ánh xạ từ các tên thuộc tính vào các giá trị, ngay cả khi thứ tự của các thuộc tính là khác nhau.

Hai biểu thức đại số quan hệ E1 và E2 là tương đương, ký hiệu là  $E1 \leftrightarrow E2$  hoặc  $E1 \equiv E2$  nếu thay thế cùng các quan hệ cho các tên giống nhau trong hai biểu thức, thì chúng có các kết quả tương đương.

### **Các tính chất**

Tính giao hoán (commutativity) của các phép toán một ngôi:

$U1 \ U2 \ R \leftrightarrow U2 \ U1 \ R$

Tính giao hoán của các toán hạng của các phép toán hai ngôi:

$$R \ B \ S \leftrightarrow S \ B \ R$$

Tính kết hợp (associativity) của các phép toán hai ngôi:

$$R \ B \ (S \ B \ T) \leftrightarrow (R \ B \ S) \ B \ T$$

Tính lũy đẳng (idempotence) của các phép toán một ngôi:

$$U \ R \leftrightarrow U \ 1 \ U \ 2 \ R$$

trong đó  $U, U_1, U_2$  thuộc cùng loại phép toán.

Tính phân phối (distributivity) của các phép toán một ngôi đối với các phép toán hai ngôi:

$$U \ (R \ B \ S) \rightarrow U(R) \ B \ U(S)$$

Tính rút thừa số (factorization) của các phép toán một ngôi:

$$U(R) \ B \ U(S) \rightarrow U(R \ B \ S)$$

## • Tiêu chuẩn 1 và 2

Mục đích: giảm kích thước của các toán hạng của các phép toán hai ngôi trước khi thực hiện chúng.

<sup>TM</sup>Tiêu chuẩn 1 - Sử dụng tính lũy đẳng của phép chọn và phép chiếu để tạo ra các phép chọn và các phép chiếu thích hợp đối với mỗi quan hệ toán hạng.

<sup>TM</sup>Tiêu chuẩn 2 - Đẩy các phép chọn và các phép chiếu xuống phía dưới cây nếu có thể được.

## • Đồ thị toán tử và biểu thức con chung

<sup>TM</sup>Biểu thức con chung (common subexpression) là biểu thức xuất hiện nhiều lần trong truy vấn.

<sup>TM</sup>Tiết kiệm thời gian thực hiện của truy vấn.

<sup>TM</sup>Biến đổi cây toán tử thành một đồ thị toán tử.

<sup>TM</sup>Ví dụ

Truy vấn  $Q_2$  – Hãy cho biết các tên của các nhân viên làm việc trong phòng ban có mã người quản lý là 373 nhưng tiền lương của họ không lớn hơn \$35.000.



Q<sub>2</sub>:  $\Pi_{emp.name} ((emp \bowtie_{deptnum = deptnum} \sigma_{mgrnum = 373} dept) - (\sigma_{sal > 35000} emp \bowtie_{deptnum = deptnum} \sigma_{mgrnum = 373} dept))$

Biểu thức con chung

$emp \bowtie_{deptnum = deptnum} \sigma_{mgrnum = 373} dept$

<sup>TM</sup>Các phép biến đổi tương đương (liên quan đến một quan hệ R) để đơn giản hóa cây toán tử.

- **Biểu thức chuẩn tắc**

<sup>TM</sup>Biểu thức chuẩn tắc (canonical expression) của một biểu thức đại số quan hệ trên lược đồ toàn cục có được bằng cách thay thế mỗi tên quan hệ toàn cục xuất hiện trong nó bởi biểu thức đại số quan hệ tái tạo các quan hệ toàn cục từ các mảnh.

<sup>TM</sup>Sử dụng tính phân phối của phép chọn và phép chiếu đối với phép hợp và phép kết để phân phối việc xử lý đến các mảnh.

- **Đại số quan hệ định tính**

<sup>TM</sup>Quan hệ định tính (qualified relation) là một quan hệ được mở rộng bởi một vị từ định tính.

<sup>TM</sup>Ký hiệu một quan hệ định tính là một cặp  $[R: q_R]$ , trong đó R là một quan hệ được gọi là thân (body) của quan hệ định tính và  $q_R$  là một vị từ được gọi là vị từ định tính của quan hệ định tính.

- **Đại số quan hệ định tính**

Quy tắc 1

$\sigma_F [R: q_R] \Rightarrow [\sigma_F R: F \text{ AND } q_R]$

Quy tắc 2

$\Pi_A [R: q_R] \Rightarrow [\Pi_A R: q_R]$

Quy tắc 3

$[R: q_R] \times [S: q_S] \Rightarrow [R \times S: q_R \text{ AND } q_S]$

Quy tắc 4

$[R: q_R] - [S: q_S] \Rightarrow [R - S: q_R]$

Quy tắc 5

$$[R : q_R] \cup [S : q_S] \Rightarrow [R \cup S : q_R \text{ OR } q_S]$$

Quy tắc 6

$$[R : q_R] \blacktriangleright \blacktriangleleft_F [S : q_S] \Rightarrow [R \blacktriangleright \blacktriangleleft_F S : q_R \text{ AND } q_S \text{ AND } F]$$

Quy tắc 7

$$[R : q_R] \blacktriangleright <_F [S : q_S] \Rightarrow [R \blacktriangleright <_+ S : q_R \text{ AND } q_S \text{ AND } F]$$

<sup>TM</sup>Hai quan hệ định tính là tương đương nếu các thân của chúng là các quan hệ tương đương và các vị từ định tính của chúng biểu diễn cùng hàm chân trị (nghĩa là, nếu áp dụng cả hai vị từ định tính cho cùng một bộ thì chúng có cùng một giá trị chân trị).

<sup>TM</sup>Sử dụng các vị từ định tính để loại bỏ các mảnh không dùng để tạo ra kết quả của truy vấn.

<sup>TM</sup>Các phép biến đổi tương đương (liên quan đến quan hệ rỗng) để đơn giản hóa cây toán tử.

### • Tiêu chuẩn 3 và 4

Mục đích: đơn giản các quan hệ được phân mảnh ngang và các phép kết giữa các quan hệ được phân mảnh ngang.

Tiêu chuẩn 3 - Đẩy các phép chọn xuống phía các nút lá của cây, và sau đó thực hiện chúng bằng cách dùng đại số quan hệ định tính. Thay thế kết quả của phép chọn bởi quan hệ rỗng nếu vị từ định tính của kết quả bị mâu thuẫn.

Tiêu chuẩn 4 - Sử dụng đại số quan hệ định tính để định trị vị từ định tính của các toán hạng của các phép kết. Thay thế cây con, bao gồm phép kết và các toán hạng của nó, bởi quan hệ rỗng nếu vị từ định tính của kết quả của phép kết bị mâu thuẫn.

### • Đơn giản hóa các quan hệ được phân mảnh ngang

<sup>TM</sup>Ví dụ

Xét truy vấn Q3 trên quan hệ dept được phân mảnh ngang:

Q3:  $\sigma_{deptnum = 1} dept$

Đơn giản hóa các phép kết giữa các quan hệ được phân mảnh ngang

Giải pháp 1:

$$R \bowtie_{\mathcal{F}} S = (U_i R_i) \bowtie_{\mathcal{F}} (U_j S_j)$$

Giải pháp 2: phép kết phân tán (distributed join).

$$R \bowtie_{\mathcal{F}} S = U_{ij}(R_i \bowtie_{\mathcal{F}} S_j)$$

<sup>TM</sup>Đánh giá:

Chọn giải pháp 1 nếu có nhiều cặp mảnh được kết với nhau.

Chọn giải pháp 2 nếu có một số cặp mảnh được kết với nhau.

<sup>TM</sup>Đồ thị kết (join graph).

## • Tiêu chuẩn 5

<sup>TM</sup>Mục đích: biến đổi một truy vấn không có các phép kết phân tán thành một truy vấn có phép kết phân tán.

Tiêu chuẩn 5 - Để phân phối các phép kết xuất hiện trong một truy vấn toàn cục, các phép hợp (biểu diễn việc tập hợp các mảnh) phải được đẩy lên phía trên các phép kết muốn phân phối.

<sup>TM</sup>Ví dụ

Truy vấn Q<sub>4</sub> - Hãy cho biết tên (name) của tất

cả các nhà cung cấp có đơn hàng cung cấp:

Q<sub>4</sub>:  $\Pi_{\text{name}}(\text{supply} \bowtie \text{supplier})$

## • Sử dụng phép suy diễn cho các phép đơn giản hóa

<sup>TM</sup>Mâu thuẫn giữa các điều kiện chọn của các truy vấn và các vị từ định tính của các mảnh.

<sup>TM</sup>Bộ chứng minh định lý (theorem prover).

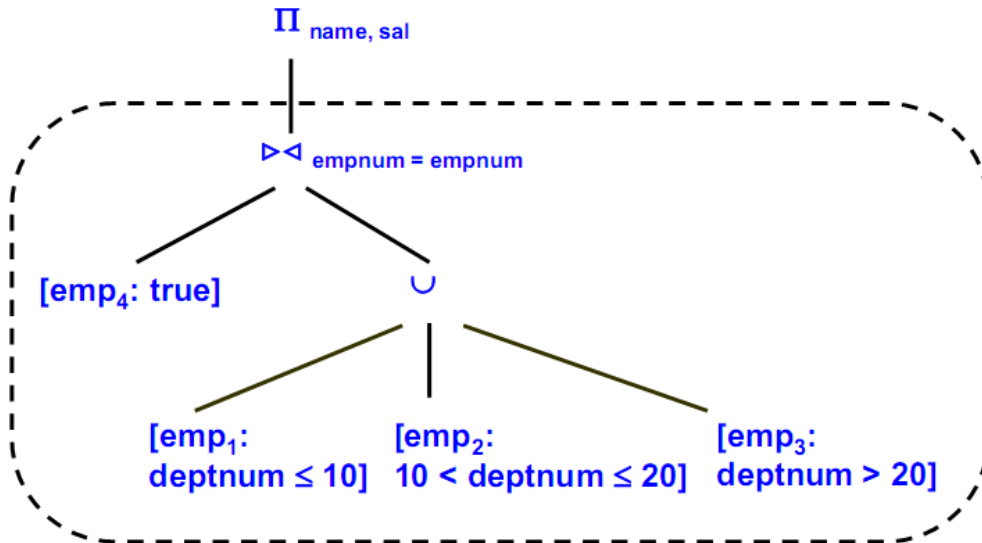
## • Đơn giản hoá các quan hệ được phân mảnh dọc

<sup>TM</sup>Mục đích: xác định một tập con bao gồm các mảnh đủ để trả lời truy vấn, sau đó loại bỏ tất cả các mảnh khác từ biểu thức truy vấn và các phép kết được dùng trong phép đổi ngược của lược đồ phân mảnh để tái tạo các quan hệ toàn cục.

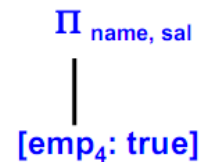
<sup>TM</sup>Ví dụ

Truy vấn Q<sub>5</sub> – Hãy cho biết tên và tiền lương  
của các nhân viên:

Q<sub>5</sub>:  $\Pi_{\text{name, sal}} \text{ emp}$



Hình 5.8. (a) Dạng chuẩn tắc của truy vấn Q<sub>5</sub>



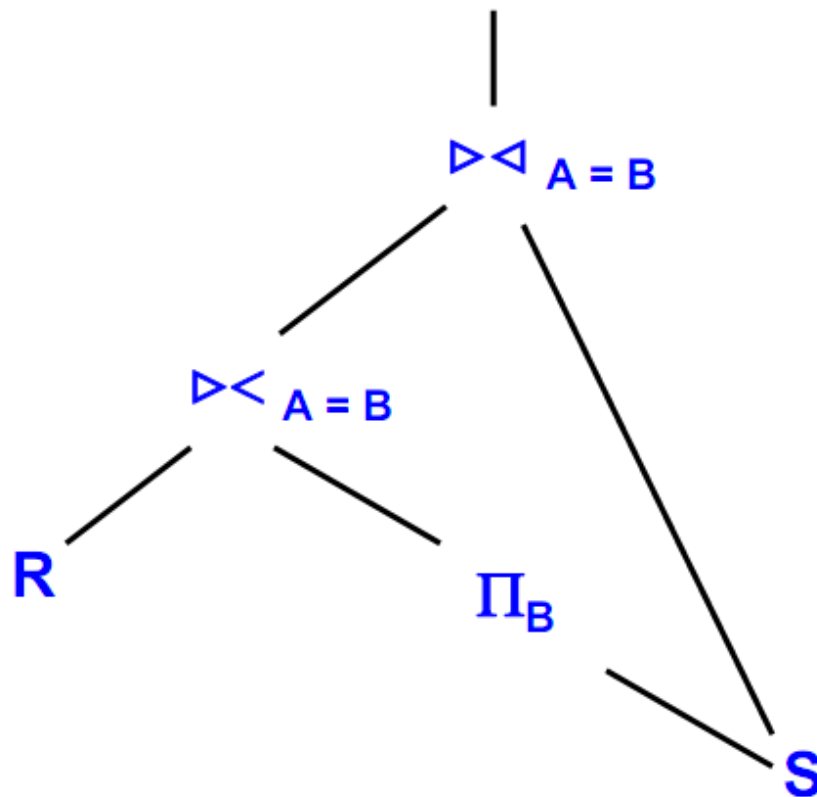
## • Chương trình nửa kết

<sup>TM</sup>Một phép kết có thể được thực hiện bởi một chương trình nửa kết (semi-join program) trong đó có các phép nửa kết.

<sup>TM</sup>Ví dụ

Xét phép kết bằng (equi-join)  $R \bowtie_{A=B} S$ , trong đó A và B là các thuộc tính (hoặc tập các thuộc tính) của R và S, chương trình nửa kết ứng với phép kết này là:

$$S \bowtie_{A=B} (R \ltimes_{A=B} \Pi_{S,B})$$



## • Phép gom nhóm

<sup>TM</sup>Phép gom nhóm

$\Psi_{G, AF} R$

$G$  – các thuộc tính dùng để xác định việc gom nhóm của  $R$ , được gọi là tập thuộc tính gom nhóm.  $G$  tương ứng với mệnh đề GROUP BY.

$AF$  – các hàm kết hợp được định trị trên mỗi nhóm.  $AF$  tương ứng với các hàm kết hợp cần được tính toán.

Có thể không có  $G$  hoặc  $AF$  .

$\Psi_{G, AF} R$  là một quan hệ có:

Lược đồ quan hệ được tạo ra bởi các thuộc tính của  $G$  và các hàm kết hợp của  $AF$ .

Nhiều bộ mà mỗi bộ là một nhóm trong  $R$ . Các thuộc tính của  $G$  lấy giá trị của nhóm. Các thuộc tính của  $AF$  lấy giá trị của các hàm kết hợp được định trị trên nhóm.

<sup>TM</sup>Ví dụ

Q6: `select AVG(quan)`

`from supply`

`where pnum = 'P1';`

$\Psi_{AVG(quan) \sigma pnum = 'P1'}$  supply

Q7: `select snum, pnum, SUM(quan)`

`from supply`

`group by snum, pnum;`

$\Psi_{snum, pnum, SUM(quan)}$  supply

Q8: `select snum, pnum, SUM(quan)`

`from supply`

`group by snum, pnum`

`having SUM(quan) > 300;`

$\sigma_{SUM(quan) > 300} \Psi_{snum, pnum, SUM(quan)}$  supply

## • Tính chất của phép gom nhóm

<sup>TM</sup>Tính phân phối của phép gom nhóm đối với phép hợp:

$\Psi_{G,AF} (R1 \cup R2) \rightarrow (\Psi_{G,AF} R1) \cup (\Psi_{G,AF} R2)$

Điều kiện cần và đủ: mỗi nhóm  $G_i$  hoặc được chứa hoặc không được giao nhau với mọi toán hạng  $R_j$ .

$$\forall i, j : (G_i \subseteq R_j) \text{ hoặc } (G_i \cap R_j = \emptyset)$$

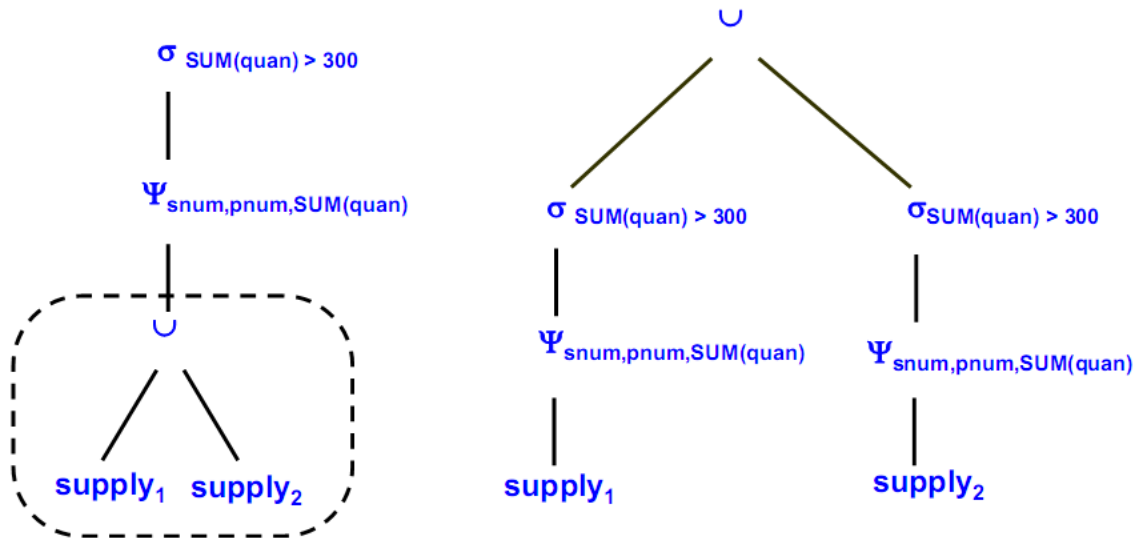
Mỗi nhóm phải được chứa hoàn toàn trong một mảnh.

Thực hiện phép gom nhóm trên các toán hạng của phép hợp và sau đó hợp các kết quả này.

## • Tiêu chuẩn 6

<sup>TM</sup>Mục đích: tập hợp các kết quả (nhỏ) của các phép gom nhóm thay vì tập hợp các quan hệ toàn cục (lớn).

™Tiêu chuẩn 6 - Để phân tán việc gom nhóm và định trị hàm kết hợp xuất hiện trong một truy vấn toàn cục, các phép hợp (biểu diễn việc tập hợp các mảnh) phải được đẩy lên phía trên phép gom nhóm tương ứng.



(a) Dạng chuẩn tắc của truy vấn Q<sub>8</sub>

(b) Bản phân tán của truy vấn Q<sub>8</sub>

## • Tính chất của hàm kết hợp

Hàm tìm giá trị nhỏ nhất

$$\text{MIN}(S) = \text{MIN}(\text{MIN}(S_1), \text{MIN}(S_2), \dots, \text{MIN}(S_n))$$

Hàm tìm giá trị lớn nhất

$$\text{MAX}(S) = \text{MAX}(\text{MAX}(S_1), \text{MAX}(S_2), \dots, \text{MAX}(S_n))$$

Hàm đếm

$$\text{COUNT}(S) = \text{SUM}(\text{COUNT}(S_1), \text{COUNT}(S_2), \dots, \text{COUNT}(S_n))$$

Hàm tính giá trị tổng cộng

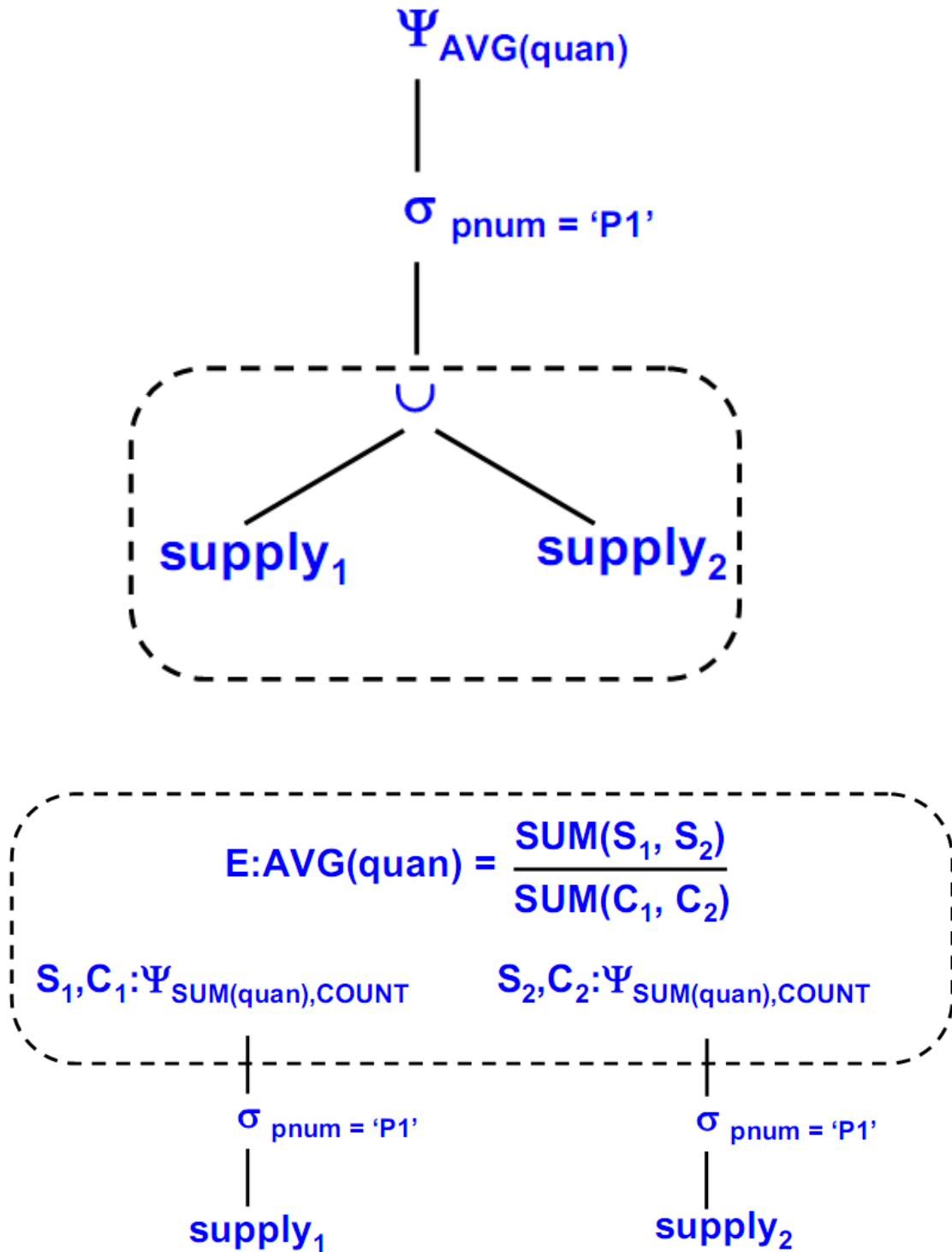
$$\text{SUM}(S) = \text{SUM}(\text{SUM}(S_1), \text{SUM}(S_2), \dots, \text{SUM}(S_n))$$

Hàm tính giá trị trung bình

$$\text{SUM}(\text{SUM}(S_1), \text{SUM}(S_2), \dots, \text{SUM}(S_n))$$

$$\text{AVG}(S) = \text{-----}$$

$SUM(COUNT(S_1), COUNT(S_2), \dots, COUNT(S_n))$



- Truy vấn có tham số



<sup>TM</sup> Truy vấn có tham số (parametric query) là truy vấn mà trong đó các công thức trong các điều kiện chọn của truy vấn bao gồm các tham số mà các giá trị của chúng chưa được biết khi biên dịch truy vấn.

<sup>TM</sup> Truy vấn có tham số cho phép thực hiện truy vấn nhiều lần với nhiều giá trị khác nhau của các tham số; ở mỗi lần thực hiện sẽ trả về kết quả khác nhau.

- **Đơn giản hóa truy vấn có tham số**

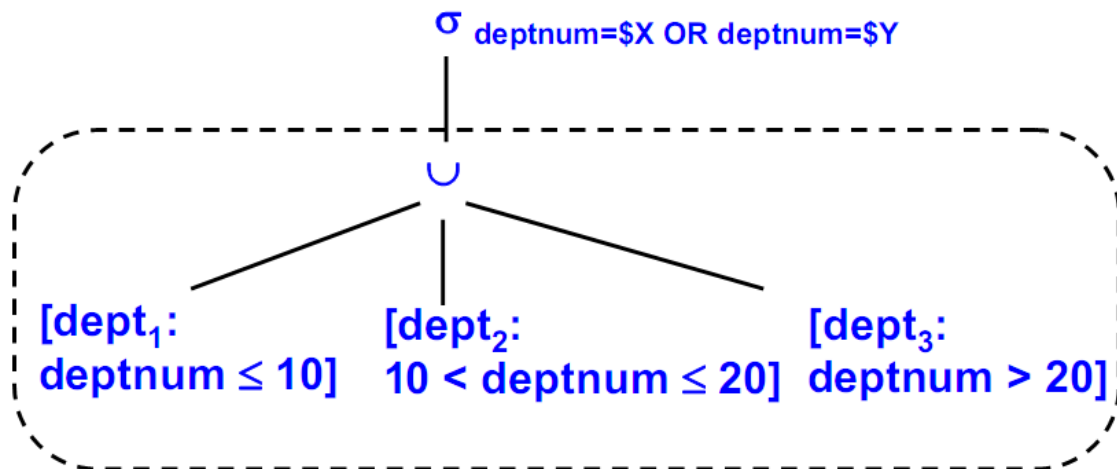
<sup>TM</sup> Ví dụ

Xét truy vấn Q<sub>9</sub> -Chọn các bộ của quan hệ toàn cục dept có các mã phòng ban cho trước. Phép chọn trên deptnum có tham số:

Q<sub>9</sub>:  $\sigma_{deptnum = \$X \text{ OR } deptnum = \$Y} dept$

Ở thời gian biên dịch: không biết các mảnh nào của quan hệ toàn cục dept sẽ được sử dụng.

Ở thời gian chạy: các giá trị thực sự được gán cho các tham số \$X và \$Y và xác định được các mảnh nào có liên quan đến truy vấn.

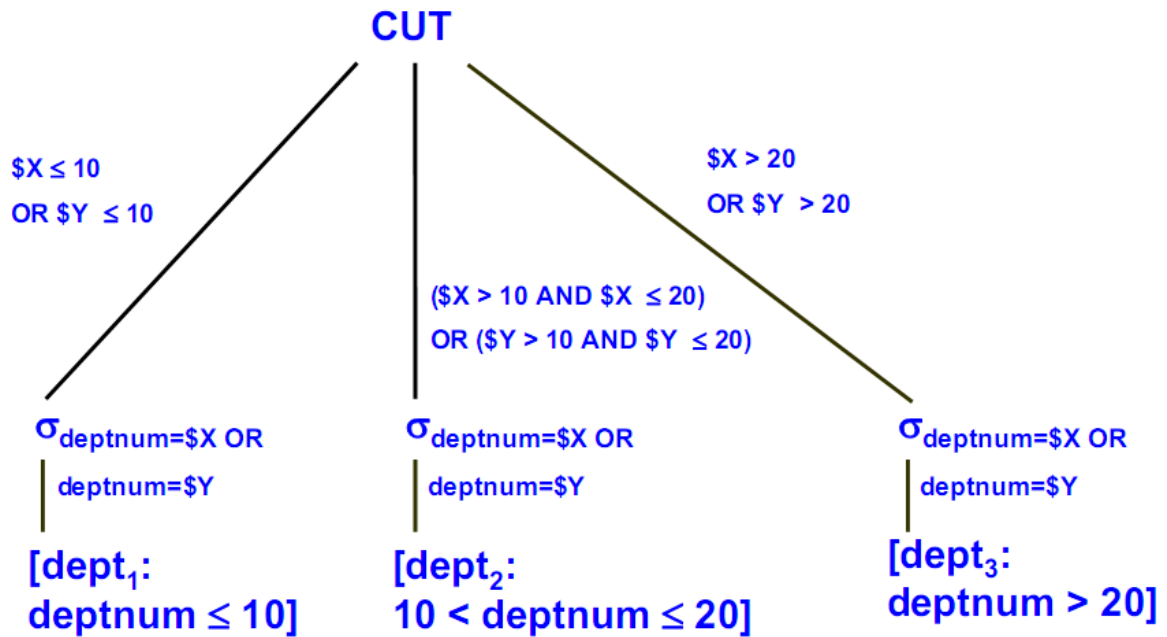


<sup>TM</sup> Đơn giản hóa truy vấn có tham số: áp dụng đại số quan hệ định tính để xác định các vị từ định tính của các biểu thức con làm mâu thuẫn với nhau.

<sup>TM</sup> Biểu diễn phép đơn giản hóa ở thời gian chạy:

Thay thế các phép hợp bởi một phép toán mới n-ngôi, được gọi là CUT.

Phép toán CUT thực hiện phép hợp của chỉ một số toán hạng của nó.



- **Sử dụng vùng nhớ tạm khi thực hiện nhiều lần truy vấn có tham số**

<sup>TM</sup> Giảm chi phí thực hiện: sử dụng các quan hệ tạm thời ở nơi gốc của truy vấn.

<sup>TM</sup> Ví dụ

Xét truy vấn  $Q_{10}$  - Hãy cho biết tên của các nhân viên đang làm việc ở phòng ban 12 mà có mã sếp là  $\$X$  (tham số của truy vấn):

$Q_{10}: \Pi_{name} \sigma_{mgrnum = \$X \text{ AND } deptnum = 12} emp$

$\Pi_{\text{name}}$   
 $\downarrow$   
 $\sigma_{\text{mgrnum} = \$X}$   
 $\downarrow$   
 $\Pi_{\text{name, mgrnum}}$   
 $\downarrow$   
 $\sigma_{\text{deptnum} = 12}$   
 $\downarrow$   
 $\text{emp}_2: 10 < \text{deptnum} \leq 20$

$\Pi_{\text{name}}$   
 $\downarrow$   
 $\sigma_{\text{mgrnum} = \$X}$   
 $\downarrow$   
 $T$   
 $\downarrow$   

$T \equiv \Pi_{\text{name, mgrnum}}$   
 $\downarrow$   
 $\sigma_{\text{deptnum} = 12}$   
 $\downarrow$   
 $\text{emp}_2: 10 < \text{deptnum} \leq 20$

## 8. CƠ SỞ DỮ LIỆU PHÂN TÁN TRONG ORACLE

### 8.1. CÁC KHÁI NIỆM:

- 1 Liên kết cơ sở dữ liệu: là 1 kết nối giữa 2 trạm máy chủ cơ sở dữ liệu (ở mức vật lý) và nó cho phép máy khách truy cập tới chúng giống như truy cập 1 cơ sở dữ liệu (ở mức logic)
- Truy vấn từ xa: là truy vấn mà lựa chọn thông tin từ 1 hay nhiều bảng trên cùng 1 trạm từ xa
- Cập nhập từ xa: là 1 cập nhập mà việc bổ sung dữ liệu trên 1 hay nhiều bảng trên cùng 1 trạm từ xa
- Truy vấn phân tán: là truy vấn lấy thông tin từ 2 hay nhiều trạm
- Cập nhập phân tán: là cập nhập bổ sung từ 2 hay nhiều trạm
- Giao tác phân tán: là bất kỳ giao tác nào bao gồm 1 hay nhiều phát biểu tác động trên 2 hay nhiều trạm phân biệt trong hệ phân tán
- Bản sao: nhằm đảm bảo mỗi lần cập nhập, các bản sao của đơn vị dữ liệu A sẽ cập nhập đồng thời trên các trạm khác nhau, Oracle dùng 1 phương thức Snapshot
- Snapshot: là 1 bản sao đầy đủ của 1 bảng hay tập các bảng mà phản ánh trạng thái gần đây của bản chủ. Snapshot được định nghĩa bởi phân tán tham chiếu đến 1 hay nhiều bảng, có 2 loại Snapshot:
  - Snapshot đơn giản: xây dựng trên 1 bảng duy nhất
  - Snapshot phức tạp: xây dựng trên nhiều bảng

Oracle lại chia Snapshot dưới 1 góc độ khác

- Snapshot chỉ đọc: không thể ghi
- Snapshot có thể cập nhập: có thể ghi

Việc làm mới dữ liệu phụ thuộc vào việc khai báo chu kỳ trong khi tạo Snapshot

- Điều khiển tính đồng hành: Oracle né tránh hiện tượng deadlock bằng 2 giải pháp:
  - Deadlock cục bộ: phương pháp chu trình đồ thị
  - Deadlock toàn cục: phương pháp quá hạn

## **8.2.TỔNG QUAN CSDLPT TRONG ORACLE:**

### **Kiến trúc CSDLPT trong Oracle**

1 hệ thống CSDLPT cho phép các ứng dụng truy cập dữ liệu từ CSDL cục bộ hay từ các CSDL từ xa.trong 1 hệ thống CSDLPT đồng nhất, mỗi CSDL là 1 CSDL Oracle.trong hệ thống CSDLPT không đồng nhất,có ít nhất 1 CSDL không phải CSDL Oracle.CSDLPT sử dụng 1 kiến trúc client/server để xử lý các yêu cầu thông tin

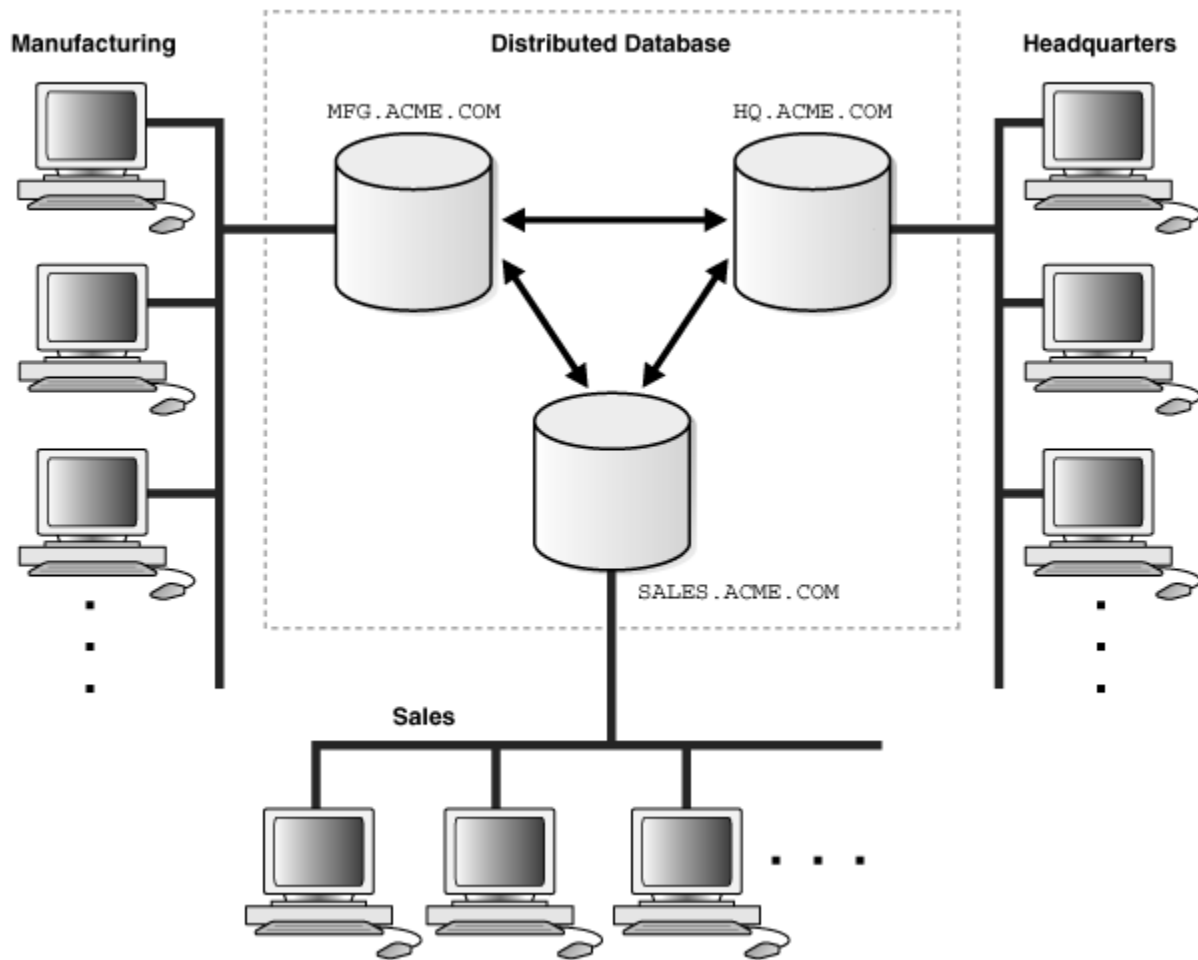
#### **8.2.1.HỆ THỐNG CSDLPT ĐỒNG NHẤT:**

1 hệ thống CSDLPT đồng nhất là 1 mạng kết nối của 2 hay nhiều CSDL Oracle được lưu trữ trên 1 hay nhiều máy.hình vẽ dưới minh họa 1 hệ thống phân tán kết nối với 3 CSDL:hq,mfg,sales.1 ứng dụng có thể truy cập hay sửa đồng thời dữ liệu trong 1 vài CSDL.ví dụ: có 1 truy vấn đơn từ 1 máy khách của nhà sản xuất trên CSDL cục bộ mfg có thể lấy dữ liệu được kết nối từ bảng products trên CSDL cục bộ và bảng dept trên CSDL từ xa hq

Ví dụ nếu ta được kết nối truy cập tới CSDL mfg nhưng lại muốn truy cập dữ liệu trên CSDL hq, vì vậy phải tạo ra 1 sự đồng nhất trên mfg cho bảng dept từ xa.nên nó cho phép ta thực hiện truy vấn sau:

```
SELECT * FROM dept;
```

Trong truy vấn này, hệ thống PT cho ta hình thức truy cập dữ liệu riêng.người dùng trên mfg không cần phải biết dữ liệu họ truy cập nằm ở CSDL từ xa nào



### 8.2.2.HỆ THỐNG CSDLPT KHÔNG ĐỒNG NHẤT:

Như đã nói ở trên, trong HT này có ít nhất 1 CSDL không phải là CSDL Oracle. trong ứng dụng, HT CSDLPT không đồng nhất được hiểu như là 1 CSDL độc lập, cục bộ hay Oracle. máy chủ CSDL Oracle cục bộ ẩn đi tính phân tán hay tính không đồng bộ của dữ liệu

Máy chủ CSDL Oracle truy cập tới HT CSDL không phải Oracle bằng cách sử dụng dịch vụ không đồng bộ Oracle trong liên kết với 1 agent (tác nhân). nếu ta truy cập dữ liệu của CSDL không phải của Oracle thì phải sử dụng cổng có tính trong suốt của Oracle, sau đó agent là 1 ứng dụng cụ thể. ví dụ, nếu ta gộp 1 CSDL quan hệ vào trong 1 HT CSDLPT Oracle, sau đó ta cần thu được 1 cổng vào trong suốt được định rõ quan hệ vì vậy CSDL Oracle trong hệ thống có thể giao tiếp với nó

Như vậy ta có thể sử dụng kết nối chung để truy cập dữ liệu của CSDL không phải Oracle miễn là nó được lưu trữ trong 1 HTCSDB có hỗ trợ giao thức ODBC hay OLE DB

### **Dịch vụ không đồng nhất(HS: Heterogeneous Services):**

Là 1 bộ phận được tích hợp sẵn trong phạm vi máy chủ CSDL Oracle. HS cung cấp kiến trúc và cơ chế quản lý chung cho các sản phẩm công cơ sở dữ liệu Oracle và các thiết bị truy cập khác không đồng nhất

### **Các tác nhân cổng vào trong suốt(Transparent Gateway Agents):**

Với mỗi HTCSDB không phải Oracle mà ta truy cập, HS có thể phải sử dụng 1 tác nhân cổng vào trong suốt để kết hợp với HTCSDB không phải Oracle đã được định rõ. tác nhân dùng để xác định rõ HTCSDB không phải Oracle, vì vậy với mỗi dạng HT cần tới 1 tác nhân khác nhau

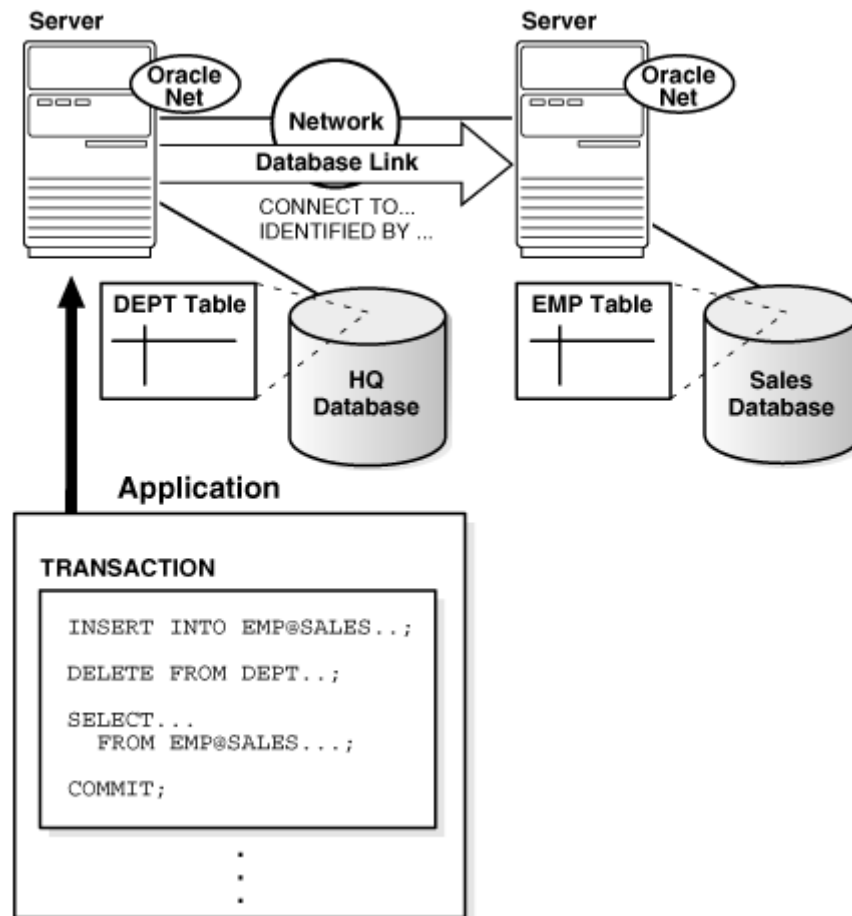
Tác nhân xử lý truy vấn SQL và các yêu cầu giao dịch ở HTCSDB không phải Oracle thay mặt cho máy chủ CSDL Oracle

### **Kết nối chung (Generic Connectivity):**

Cho phép ta kết nối nơi lưu giữ dữ liệu của csdl bằng cách sử dụng 1 tác nhân HS ODBC hay 1 tác nhân HS OLE DB. bất cứ nguồn dữ liệu nào tương thích với ODBC hay OLE DB có thể được truy cập bằng cách sử dụng tác nhân kết nối chung

### **Kiến trúc CSDL Client/Server:**

1 máy chủ CSDL là 1 phần mềm Oracle dùng để quản lý 1 CSDL, và 1 máy khách là 1 ứng dụng yêu cầu lấy thông tin từ 1 máy chủ. mỗi máy tính trong 1 mạng là 1 nút có thể lưu trữ 1 hay nhiều CSDL. mỗi nút trong HTCSDBPT có thể hoạt động như 1 máy khách khi nó thu thập các phản hồi từ 1 dữ liệu từ xa (ví dụ này, thông tin đầu tiên trong mỗi giao dịch được thu thập từ bảng emp trong CSDL sales)



1 máy khách có thể kết nối trực tiếp hoặc gián tiếp vào máy chủ CSDL. 1 kết nối trực tiếp xảy ra khi 1 máy khách kết nối tới 1 máy chủ và truy cập thông tin từ 1 CSDL chứa trong máy chủ đó. ví dụ: nếu ta kết nối tới CSDL hq và truy cập bảng dept ở trên CSDL này, ta có thể truy cập các bước như sau:

```
SELECT * FROM dept;
```

Câu truy vấn này là truy vấn trực tiếp bởi vì ta không phải truy cập đến máy chủ từ xa

Ngược lại 1 kết nối không trực tiếp xảy ra khi máy khách kết nối tới 1 máy chủ nhưng lại lấy thông tin từ 1 CSDL ở trong 1 máy chủ khác. ví dụ: nếu ta kết nối với CSDL hq nhưng lại truy cập bảng emp ở trên CSDL từ xa emp, ta có thể thực hiện như sau:

```
SELECT * FROM emp@sales;
```

Câu truy vấn này là không trực tiếp



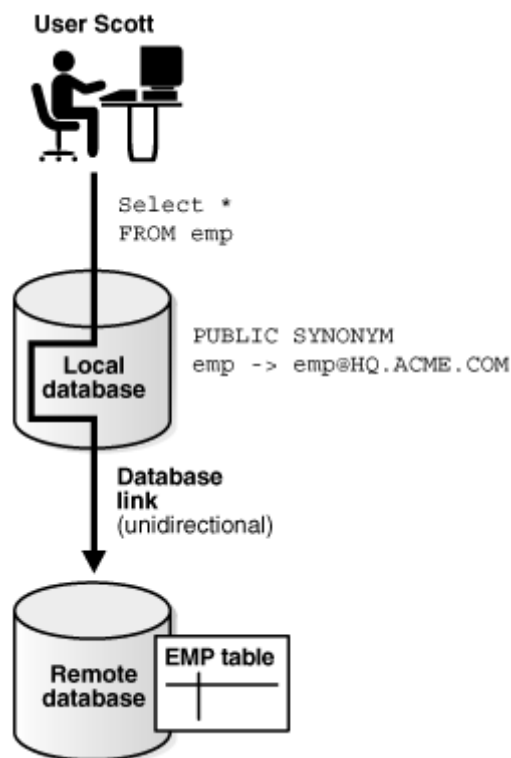
### 8.2.3.DATABASE LINKS(LIÊN KẾT CSDL):

1 liên kết CSDL là 1 điểm trở tới liên kết dùng để xác định đường kết nối từ 1 máy chủ CSDL Oracle tới 1 máy chủ CSDL khác.điểm trở liên kết này được định nghĩa là 1 thực thể nằm trong thư mục bảng dữ liệu.để truy cập tới liên kết, ta phải được kết nối tới CSDL cục bộ-nơi chứa thực thể thư mục dữ liệu

1 kết nối liên kết CSDL là đường kết nối nằm trong hướng mà 1 máy khách kết nối với 1 CSDL cục bộ A,có thể sử dụng 1 liên kết được lưu trữ trong CSDL A để truy cập tới thông tin trong CSDL từ xa B, nhưng những người sử dụng kết nối với CSDL B không thể sử dụng cùng 1 liên kết để truy cập tới dữ liệu nằm trong CSDL A.nếu những người dùng cục bộ ở CSDL B muốn truy cập dữ liệu trên CSDL A, thì họ phải xác định 1 liên kết kết nối khác, liên kết này được lưu giữ trong thư mục DL của CSDL B

1 kết nối liên kết CSDL cho phép người dùng cục bộ truy cập tới dữ liệu trên 1 CSDL từ xa.để kết nối này xảy ra, thì mỗi CSDL trong HTPT phải có 1 tên CSDL toàn cục duy nhất trong vùng mạng kết nối.tên CSDL toàn cục xác định duy nhất 1 máy chủ CSDL trong HTPT

Hình minh họa truy cập 1 bảng emp trên CSDL từ xa với tên toàn cục là hq.acme.com



Các liên kết CSDL có thể ở dạng private hay public. nếu chúng ở dạng private, thì chỉ người sử dụng tạo liên kết đó mới được truy cập. nếu chúng ở dạng public, thì tất cả người sử dụng CSDL đều được truy cập

1 điểm khác biệt cơ bản chủ yếu giữ các liên kết CSDL chính là cách kết nối tới CSDL từ xa được tạo ra. những người sử dụng truy cập tới 1 CSDL từ xa thông qua các dạng liên kết sau:

Type of Link	Description
<b>Connected user link</b>	những người sử dụng kết nối tới chính họ, có nghĩa là họ phải có 1 tài khoản trên CSDL từ xa với cùng tên và mật khẩu giống như tài khoản của họ trên CSDL cục bộ
<b>Fixed user link</b>	những người dùng kết nối sử dụng tên và mật khẩu được tham chiếu trong liên kết. ví dụ: , nếu jane sử dụng 1 liên kết cố định kết nối với CSDL hq với tên và mật khẩu là scott/tiger, thì cô ấy được kết nối thông qua tên scott, jane có tất cả các quyền trong hq như scott đã được phân quyền trong CSDL
<b>Current user link</b>	1 người dùng kết nối giống như là người dùng toàn cục. 1 người dùng cục bộ kết nối như là 1 người dùng toàn cục trong phạm vi của 1 thủ tục lưu trữ, và không lưu trữ được mật khẩu của người dùng toàn cục trong 1 định nghĩa liên kết. ví dụ, jane có thể truy cập 1 thủ tục mà scott đã viết, truy cập tài khoản của scott và sơ đồ của scott trong CSDL hq. liên kết người dùng hiện tại là 1 dạng bảo mật nâng cao của Oracle

Câu lệnh tạo ra liên kết CSDL : CREATE DATABASE LINK

### Các liên kết CSDL chia sẻ là gì?

1 liên kết CSDL chia sẻ là 1 liên kết giữa tiến trình xử lý máy chủ cục bộ và CSDL từ xa. liên kết có thể chia sẻ bởi vì nhiều tiến trình xử lý có thể sử dụng đồng thời cùng 1 liên kết

Khi 1 CSDL cục bộ được kết nối tới 1 CSDL từ xa thông qua 1 liên kết CSDL, thì CSDL có thể chạy ở 1 trong 2 chế độ dedicated hay shared. bảng sau sẽ mô tả rõ điều này:

Local Database Mode	Remote Database Mode
Dedicated	Dedicated
Dedicated	Shared server
Shared server	Dedicated
Shared server	Shared server

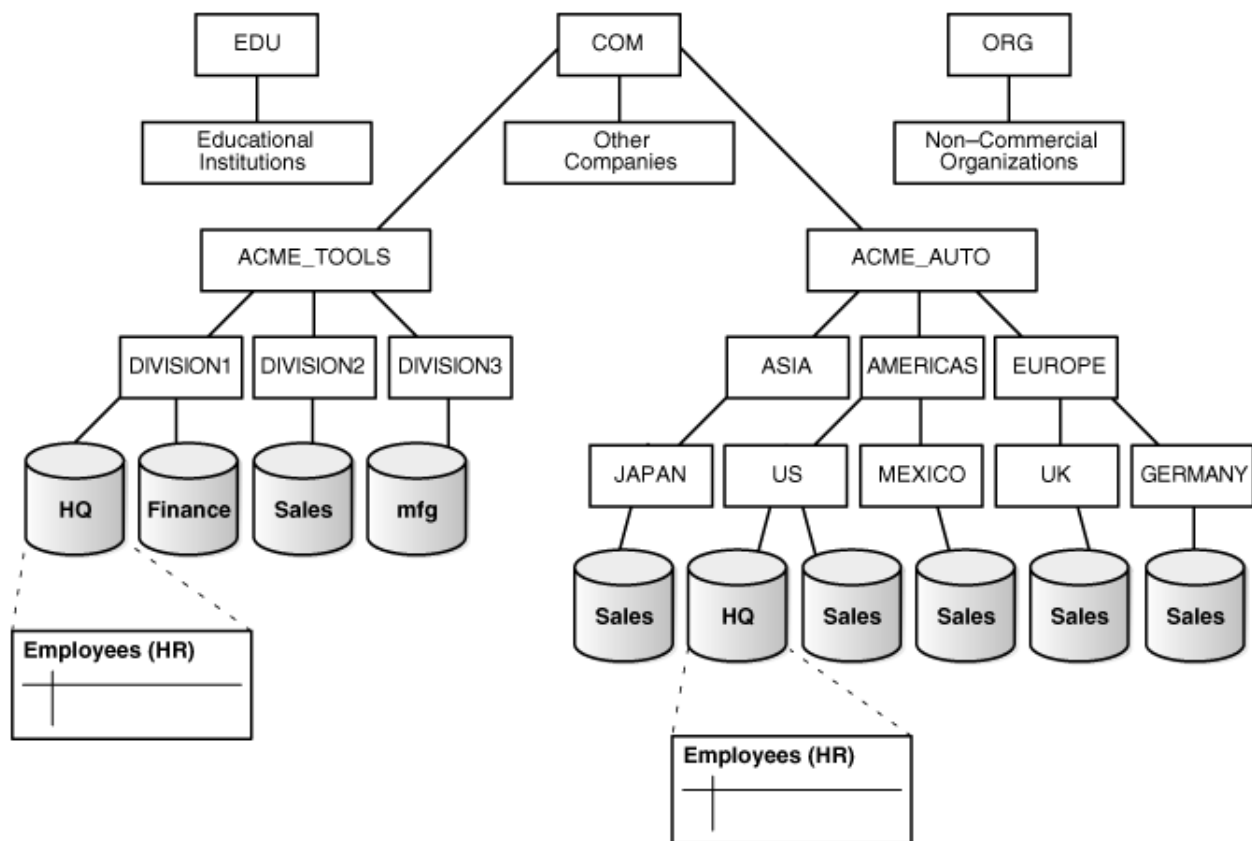
1 liên kết CSDL chia sẻ có thể tồn tại bất kỳ trong 4 loại cấu hình trên. các liên kết chia sẻ khác với liên kết chuẩn của CSDL theo các cách sau:

- Những người dùng khác nhau truy cập cùng 1 đối tượng mô hình thông qua 1 liên kết CSDL có thể chia sẻ 1 kết nối mạng
- Khi 1 người dùng cần cho phép 1 kết nối tới 1 máy chủ từ xa từ 1 tiến trình xử lý riêng, tiến trình xử lý này có thể sử dụng lại kết nối vừa được cho phép. sử dụng lại kết nối xảy ra nếu kết nối được cho phép trên cùng 1 tiến trình xử lý máy chủ với cùng 1 liên kết CSDL, có thể trong 1 phiên làm việc khác. còn trong liên kết CSDL không chia sẻ, 1 kết nối là không được chia sẻ thông qua nhiều phiên làm việc
- Khi ta sử dụng 1 liên kết chia sẻ trong 1 cấu hình máy chủ chia sẻ, 1 kết nối mạng được phép trực tiếp vượt ra ngoài tiến trình xử lý máy chủ trong máy chủ cục bộ. với liên kết CSDL không chia sẻ, kết nối này được cho phép thông qua bộ điều phối cục bộ

#### 8.2.4. CÁC TÊN CSDL TOÀN CỤC TRONG CÁC LIÊN KẾT CSDL:

Mỗi CSDL trong CSDL phân tán là duy nhất được xác định bởi tên CSDL toàn cục của nó. tên này có dạng tiền tố là tên vùng mạng được xác định bởi tham số khởi tạo DB\_DOMAIN trong việc tạo CSDL + với tên của CSDL được định nghĩa qua các tham số khởi tạo DB\_NAME

Sơ đồ phân cấp CSDL



Tên của 1 CSDL được định dạng bắt đầu từ lá của cây và theo các đường dẫn tới gốc.ví dụ, CSDL mfg nằm trong nhánh division3 của acme\_tools của vùng mạng com.tên CSDL toàn cục cho mfg được tạo ra là :

- mfg.division3.acme\_tools.com

trong khi 1 vài CSDL có thể được chia sẻ 1 tên riêng, mỗi CSDL phải có 1 tên CSDL toàn cục.ví dụ tên miền mạng us.americas.acme\_auto.com và uk.europe.acme\_auto.com mỗi cái đều chứa 1 CSDL sales.tên CSDL toàn cục được phân biệt CSDL sales trong americas với CSDL sales trong europe như sau:

- sales.us.americas.acme\_auto.com
- sales.uk.europe.acme\_auto.com

tên cho các liên kết CSDL

1 liên kết CSDL đặc trưng có cùng tên giống như tên CSDL toàn cục của CSDL từ xa ma nó tham chiếu tới.ví dụ, nếu tên CSDL toàn cục của 1 CSDL là sales.us.oracle.com thì liên kết CSDL cũng được gọi là sales.us.oracle.com

Khi ta cài đặt khởi tạo tham số GLOBAL\_NAMES là true, CSDL đảm bảo rằng tên của liên kết CSDL là giống với tên toàn cục CSDL của CSDL từ xa. ví dụ, nếu tên CSDL toàn cục của hq là hq.acme.com và GLOBAL\_NAMES là true, thì tên liên kết phải là hq.acme.com. chú ý rằng CSDL kiểm tra xem 1 phần miền của tên CSDL toàn cục được lưu trữ trong thư mục dữ liệu, nếu ta cài đặt khởi tạo tham số GLOBAL\_NAMES là false thì ta không phải cần sử dụng cách đặt tên toàn cục nữa. ta có thể đặt tên liên kết CSDL tùy ý. ví dụ, ta có thể đặt tên liên kết CSDL là hq.acme.com as foo

Sau khi ta bật chế độ đặt tên toàn cục, liên kết CSDL về bản chất là trong suốt với người dùng CSDLPT bởi vì tên của liên kết CSDL là giống với tên toàn cục của CSDL. ví dụ, câu lệnh tạo ra 1 liên kết CSDL trong CSDL toàn cục tới CSDL từ xa sales:

```
CREATE PUBLIC DATABASE LINK sales.division3.acme.com USING 'sales1';
```

## 8.3. GIẢI QUYẾT MÃU THUẦN TRONG MÔI TRƯỜNG PHÂN TÁN:

### 8.3.1. ĐẶT VẤN ĐỀ:

Do tồn tại nhiều bản sao, DL tại mỗi thời điểm trên các bản sao có thể không đồng nhất do lỗi mạng hay do đặc thù của các Snapshot chưa đến chu kỳ làm mới

Trước tình huống như thế Oracle chọn giải pháp nào? Đây là 1 vấn đề liên quan đến khóa phân tán, tuy nhiên do cách đọc và ghi là tương đối độc lập, snapshot né tránh phương thức: cấm ghi khi đọc và cấm đọc khi ghi, nhưng lại đối đầu với khó khăn mới là DL không đồng nhất

### 8.3.2. CÁC TÌNH HUỐNG:

Có 3 mẫu thuần:

- Mẫu thuần cập nhập
- Mẫu thuần vi phạm duy nhất
- Mẫu thuần khi xóa

**MÃU THUẦN CẬP NHẬP:** nảy sinh khi có sự khác nhau giữa giá trị mới và hiện hành

Ví dụ:

Nhankhau tại trạm A

MSNK                      HONNHAN  
Q1P101                    T  
Q1P102                    F  
Update Nhankhau  
Set HONNHAN = F  
Where MSNK = 'Q1P101';  
Kết quả:

Q1P101                    T (cũ)  
Q1P101                    F (mới)

MSNK                      HONNHAN  
Q1P101                    T  
Q1P102                    F  
Update Nhankhau  
Set HONNHAN = F  
Where MSNK = 'Q1P101';

Q1P101                    T  
Q1P101                    F (hiện hành)

So sánh giá trị cũ và hiện hành có mâu thuẫn

**MÂU THUẤN VI PHẠM DUY NHẤT:** nảy sinh khi có sự vi phạm ràng buộc duy nhất trong lệnh thêm hay cập nhập

Ví dụ: lấy lại ví dụ trên, trạm A thêm nhân khẩu mã 'Q1P103' và trạm B cũng thêm nhân khẩu mới trùng mã. lúc làm mới DL cho các bản sao, mâu thuẫn nảy sinh

Mâu thuẫn khi xóa: xảy ra khi trạm A đã xóa 1 dòng xác định trên NHANKHAU

Ví dụ: 'Q1P101' và trạm B vẫn tiếp tục cập nhập nhân khẩu này

### 8.3.3. CÁC PHƯƠNG PHÁP GIẢI QUYẾT:

Oracle đưa ra 1 giải pháp sau:

Ta qui ước: cầm lấy giá trị từ trạm nguồn cập nhập cho trạm đích

#### 8.3.3.1. PHƯƠNG PHÁP TỐI THIỂU VÀ TỐI ĐA:

Khi có mâu thuẫn DL xảy ra nó so sánh giá trị mới từ trạm nguồn với giá trị hiện hành của trạm đích. nếu giá trị mới của 1 cột chỉ định nhỏ hơn giá trị hiện hành, giá trị trạm gốc được dùng cho trạm đích. nếu giá trị hiện hành nhỏ hơn giá trị hiện hành được chấp nhận

Phương pháp tối đa cũng tương tự như trên nghĩa là: nếu giá trị mới từ trạm nguồn lớn hơn giá trị hiện hành tại trạm đích thì giá trị mới sẽ được chấp nhận tại trạm đích

### **8.3.3.2.PHƯƠNG PHÁP THỜI NHẢY SỚM NHẤT VÀ TRỄ NHẤT:**

Cần chọn cột kiểu DL: ngày

Khi có sự thay đổi nào đó trên cột đã chọn lựa, trình ứng dụng sẽ ghi lại ngày giờ hệ thống. nếu có sự mẫu thuẫn DL xảy ra, nó sẽ so sánh thời gian ghi trên các cột. với phương pháp thời nhảy sớm nhất nó chọn giá trị tại trạm có giá trị trên cột gần hiện tại nhất. tương tự cho kiểu thời nhảy trễ nhất

### **8.3.3.3.PHƯƠNG PHÁP THÊM VÀO VÀ GIÁ TRỊ TRUNG BÌNH:**

Cần chọn cột có kiểu DL: số

Khi có sự khác nhau giữa giá trị mới và giá trị hiện hành trên 2 trạm, phương pháp thêm vào chọn giá trị theo công thức:

Trị hiện hành = trị hiện hành + (trị mới – trị cũ)

Trong lúc phương pháp giá trị trung bình chọn công thức:

Trị hiện hành = (trị hiện hành + trị mới) / 2

### **8.3.3.4.PHƯƠNG PHÁP TRẠM ƯU TIÊN VÀ NHÓM ƯU TIÊN:**

Phương pháp này gán các giá trị ưu tiên theo 2 kiểu:

Tên trạm: dùng 1 bảng lưu thông tin mức độ ưu tiên các trạm, nếu có mâu thuẫn trên các trạm giá trị trên trạm có mức ưu tiên cao sẽ được chọn

Tên nhóm: mỗi nhóm cột trên 2 bảng sẽ có độ ưu tiên khác nhau. khi có mâu thuẫn nó sẽ đối so sánh giá trị theo nhóm cột và dựa vào độ ưu tiên để chọn DL trên trạm nào là đúng nhất

### **8.3.3.5.PHƯƠNG PHÁP HỦY BỎ VI PHẠM TÍNH DUY NHẤT:**

Các làm đơn giản như sau: khi đối so sánh DL từ trạm nguồn và trạm đích nếu phát hiện vi phạm tính duy nhất nó sẽ xóa các dòng mới phát sinh làm vi phạm ràng buộc

### **8.3.3.6.PHƯƠNG PHÁP THÊM TÊN TRẠM:**

Một cách giải quyết khác cho trường hợp xảy ra tính mâu thuẫn DL khi vi phạm tính duy nhất là thêm vào tên trạm ở phần đuôi.vì thế cột phát sinh lỗi có kiểu DL: chuỗi

#### **8.3.3.7.PHƯƠNG PHÁP CHÉP CHỒNG VÀ HỦY BỎ:**

Chép chồng: khi có mâu thuẫn nó thay thế giá trị hiện hành tại trạm đích bằng cách chép chồng các giá trị tại trạm nguồn

Ngược lại phương pháp hủy bỏ sẽ lấy giá trị từ trạm đích và hủy bỏ giá trị mới từ trạm nguồn

#### **8.3.4.PHÁT TRIỂN CHIẾN LƯỢC GIẢI QUYẾT MÂU THUẤN:**

Dựa trên nguyên lý ‘phòng bệnh còn hơn chữa bệnh’

Khi thiết kế cơ sở dữ liệu trong môi trường phân tán phải chú ý:

- Các ứng dụng phải được modun hóa
- DL phải định nghĩa trong sáng
- DL phải chuẩn hóa để giảm ước sự phụ thuộc lẫn nhau

Cần xem xét:

- Mô hình chỉ có 1 modun cập nhập còn các modun khác chỉ đọc
- Phân hóa DL cho các vị trí sở hữu đúng chức năng

Ngoài ra:

- Dùng sequence phát sinh số thứ tự tránh khả năng vi phạm tính duy nhất
- Phân quyền sở hữu DL tường minh
- Không lạm dụng việc xóa, vì giải quyết rất khó khăn liên quan đến việc ghi lại lịch sử của 1 dòng

Cuối cùng:

Sau khi đã làm tất cả các việc có thể để “phòng bệnh”, cần chọn lựa phương pháp giải quyết mâu thuẫn cho từng trường hợp cụ thể.không có lời giải nào là tổng quát vì phụ thuộc vào trình ứng dụng cụ thể, công việc chức năng cụ thể đối so sánh với các đặc điểm của các phương pháp, chúng ta sẽ tìm ra đáp số

Ví dụ:



Có 1 bảng A, trên cột ghi thông số của thị trường chứng khoán, do đặc trưng riêng tình huống này nếu có mâu thuẫn DL nên chọn phương pháp thời nhần với thời gian sớm nhất

## 8.4.CƠ CHẾ QUẢN LÝ GIAO TÁC:

Để đảm bảo tính an toàn dữ liệu toàn cục, Oracle tự động điều khiển và giám sát việc thỏa thuận hay rollback 1 giao tác

Việc xử lý việc cập nhập phân tán gồm 2 pha:

- Chuẩn bị pha
- Thỏa thuận pha

Tuy vận dụng cơ chế 2 pha của lý thuyết cơ sở dữ liệu phân tán phần trước nhưng Oracle lại hình thành thêm 1 trạm có vai trò mới.trạm thỏa thuận, phần sau sẽ cho ta thấy các điểm giống và khác nhau giữa Oracle và lý thuyết chung

Trước khi đi vào chi tiết, chúng ta lược qua các khái niệm liên quan đến các phần sau:

Vai trò của 1 trạm được xác định bởi:

- Giao tác là cục bộ hay từ xa
- Trong thỏa thuận của trạm đó
- Tất cả dữ liệu yêu cầu là có sẵn tại đó hay các trạm khác cần để được tham chiếu

Máy khách: trạm xem là máy khách khi nó tham chiếu thông tin từ CSDL trạm khác

Máy chủ: trạm hỗ trợ CSDL gọi là CSDL máy chủ

Trạm điều hợp cục bộ: 1 trạm mà phải tham chiếu dữ liệu trên trạm khác để hoàn thành 1 phần giao tác của nó.nó có trách nhiệm giám sát các giao tác trong các trạm mà nó thông tin trực tiếp

Trạm điều hợp toàn cục trạm nơi giao tác bắt nguồn.trạm này trở thành gốc trong phần “cây của 1 giao tác phân tán”, nó thực hiện các công việc:

- Tất cả các phát biểu SQL, lời gọi thủ tục từ xa được gửi do toàn cục trạm điều hợp
- Ra chỉ thị chuẩn bị các trạm
- Nếu chuẩn bị tốt, nó chỉ thị trạm thỏa thuận để tạo việc thỏa thuận toàn cục

- Nếu có 1 thông báo hủy bỏ, nó chỉ thị các trạm tạo ra rollback

Điểm thỏa thuận: công việc của nó nhằm tạo ra thỏa thuận hay rollback lúc nhận chỉ thị từ trạm điều hợp toàn cục. trạm được chọn thường là trạm lưu giữ hầu hết DL quan trọng. cần phân biệt trạm thỏa thuận với các đặc điểm sau:

- Nó không vào trạng thái chuẩn bị. vì nó lưu giữ DL quan trọng nên không bao giờ nghi ngờ nó ngay cả khi có lỗi xảy ra
- Kết quả ở giao tác phân tán ở trạm này cho biết các trạm khác là thỏa thuận hay rollback

1 giao tác phân tán được coi là thỏa thuận nếu tất cả các trạm đã chuẩn bị và giao tác đã thỏa thuận tại trạm thỏa thuận. tương tự, 1 giao tác được xem là không thỏa thuận nếu nó không thỏa thuận tại trạm này

Trong thỏa thuận: điểm này được chọn từ các trạm tham gia vào giao tác. nó được chỉ định khi trạm điều hợp toàn cục xác định các trạm được chuẩn bị để chuẩn bị thỏa thuận hay rollback. các trạm tham chiếu trực tiếp, trạm điều hợp cục bộ có trọng cao nhất sẽ được lựa chọn

Chú ý:

- Trạm chỉ đọc không thể chỉ định là trạm thỏa thuận
- Nếu có nhiều điểm cùng giá trị trọng thì Oracle chỉ định

a. pha chuẩn bị: trạm khởi tạo (còn gọi là trạm điều hợp) hỏi các trạm để chuẩn bị. khi 1 trạm thành phần được hỏi, có thể trả lời bằng 3 trạng thái:

- Chuẩn bị: dữ liệu trên trạm đã được bổ sung bởi phát biểu phân tán
- Chỉ đọc: dữ liệu không thể ghi vì thế chuẩn bị là không cần
- Hủy bỏ: chuẩn bị không được

b. pha thỏa thuận: trước khi pha này xảy ra, các trạm tham gia giao tác phân tán đảm bảo đầy đủ tài nguyên để thỏa thuận. pha thỏa thuận gồm 2 bước:

- Trạm điều hợp gửi thông điệp tới các trạm thông báo thỏa thuận
- Tại mỗi trạm, Oracle thỏa thuận giao tác phân tán có thể đóng 1 hay nhiều vai trò:
  - Máy khách
  - CSDL máy chủ
  - Trạm điều hợp toàn cục
  - Trạm điều hợp cục bộ
  - Trạm thỏa thuận

Trọng thỏa thuận của mỗi trạm được thông báo với trạm điều hợp khi khởi tạo việc kết nối được hình thành. trạm điều hợp ghi lại các thông tin này để thỏa thuận được chọn lựa trong khi chuẩn bị/ thỏa thuận

Việc xác định điểm thỏa thuận từ các trạm bảo đảm rằng hầu hết các máy chủ quan trọng không có lỗi trong khi xảy ra chuẩn bị/ thỏa thuận

Trọng thỏa thuận liên quan số lượng những va chạm có thể dẫn đến từ các giao tác nghi ngờ. ví dụ máy chủ mainframe có trọng lớn hơn máy chủ mini và đến lượt mini có trọng lớn hơn máy chủ PC

Số giao tác phân tán trên mỗi trạm: là số giao tác đồng thời có thể tham gia trên mỗi trạm. người quản trị CSDL, có thể gia tăng số này trong tham số DISTRIBUTED-TRANS

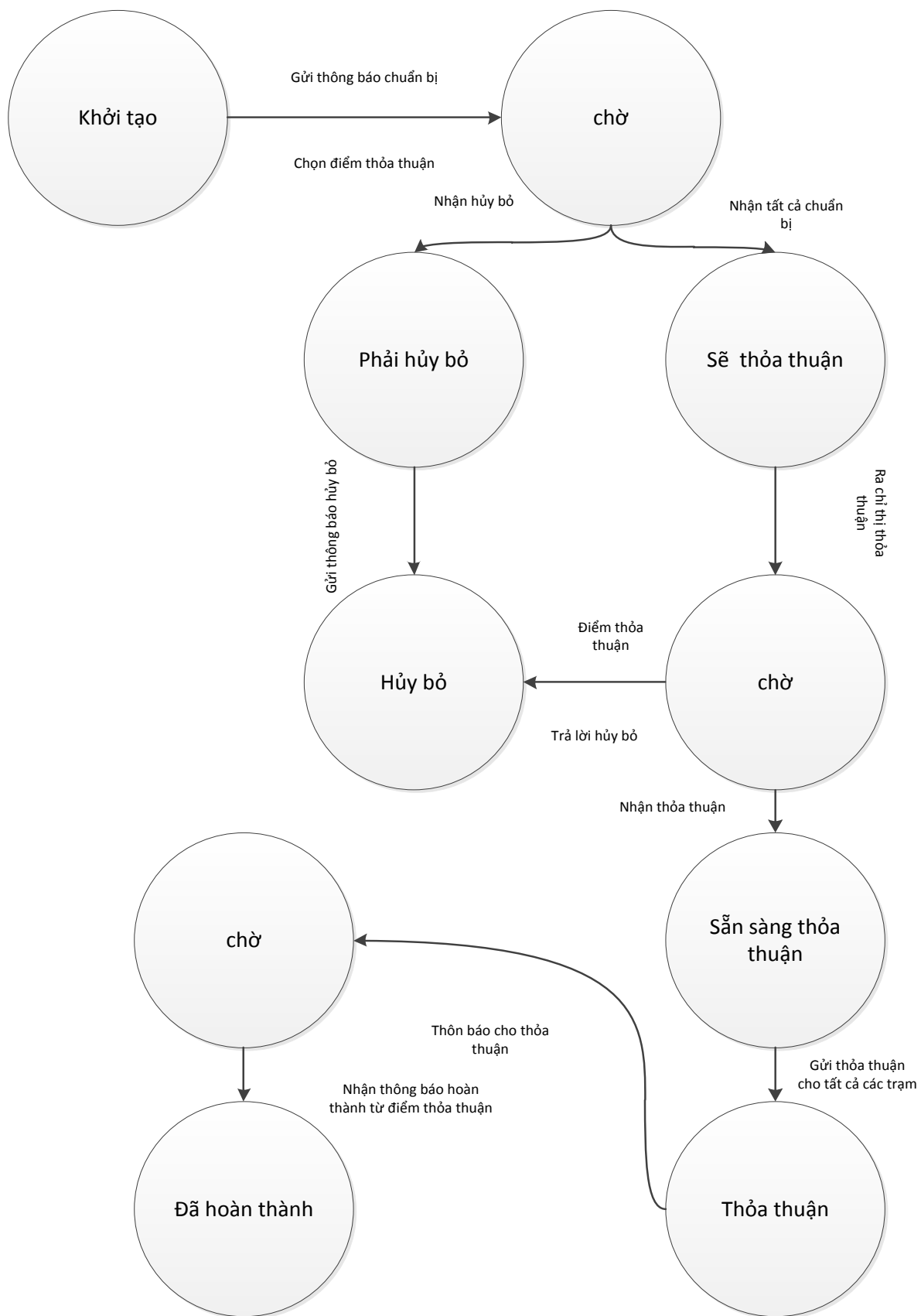
Các tình huống trong quá trình phân tán:

Khi người dùng phát câu lệnh SQL, Oracle cố gắng khóa nguồn tài nguyên và cố gắng thực hiện tốt phát biểu. nếu dữ liệu đòi hỏi hiện tại bị giữ bởi các phát biểu chưa thỏa thuận và vẫn bị khóa vượt quá thời hạn cho phép, phát biểu sẽ được rollback

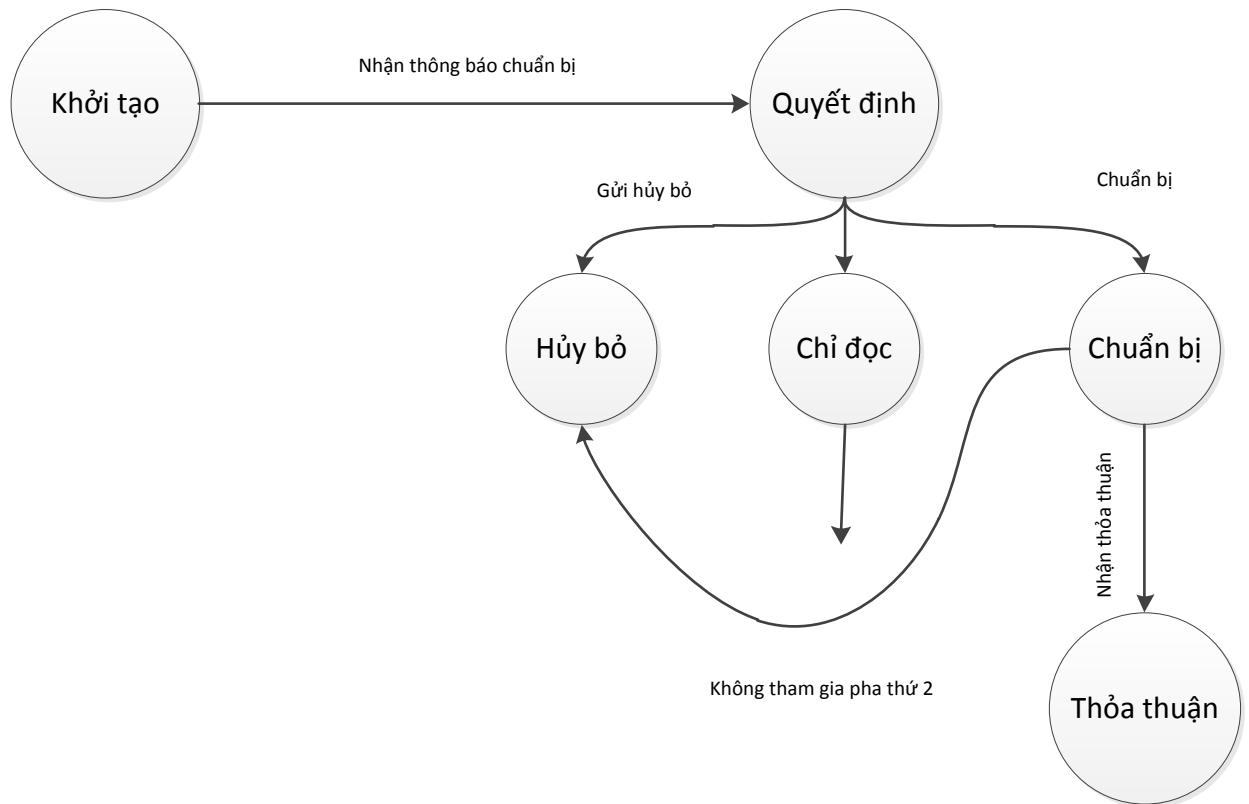
Khoảng thời gian trong tình huống trên có thể điều chỉnh trong tham số DISTRIBUTED-TIMEOUT. 1 khả năng khác: câu phát biểu SQL yêu cầu khóa dữ liệu cục bộ nhưng nó bị khóa bởi 1 giao tác phân tán nào đó ở trạng thái ngờ vực, 1 lỗi cũng xảy ra cùng với chỉ số ID giao tác gây cản trở đó. nếu sau đó câu phát biểu vẫn không thể thực hiện, người quản trị mạng cần có biện pháp giải quyết

## **8.5.SƠ ĐỒ CÁC TRẠM:**

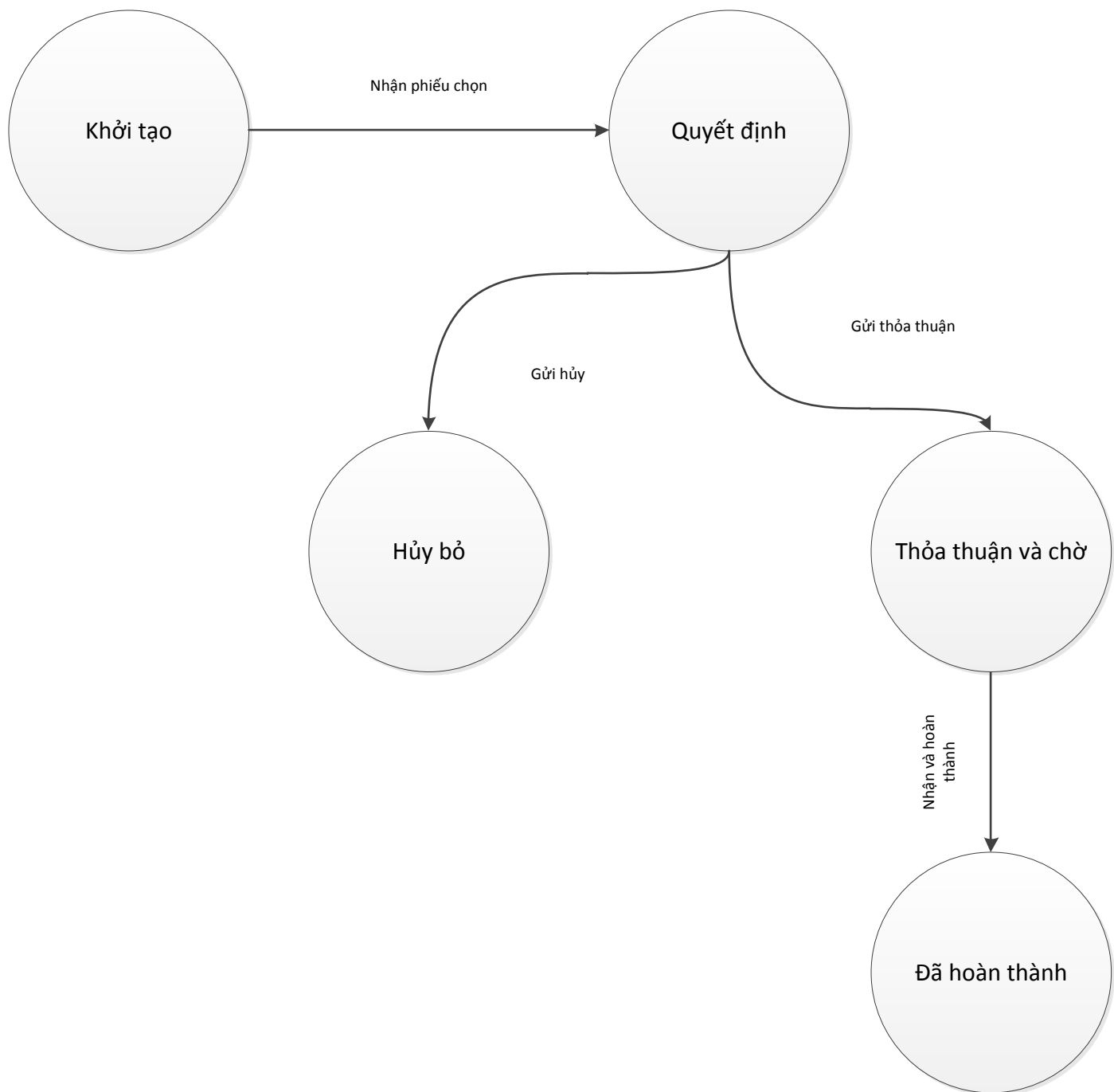
### **8.5.1.SƠ ĐỒ TRẠM ĐIỀU HỢP:**



### 8.5.2.SƠ ĐỒ TRẠNG THÁI TRẠM THÀNH PHẦN:



### 8.5.3.SƠ ĐỒ TRẠNG THÁI ĐIỂM THỎA THUẬN:



## 8.6.KHẢ NĂNG PHỤC HỒI TRONG GIAO TÁC PHÂN TÁN:

Oracle dùng cơ chế RECO để xử lý, khi có lỗi kết nối mạng xảy ra làm ngắt quãng cơ chế 2 pha thỏa thuận thì RECO tự động khôi phục các giao tác cục bộ ngừng việc.nó có thể dùng 1 liên kết tồn tại hay thiết lập 1 kết nối tới những trạm liên quan đến giao

tác bị lỗi.khi kết nối được thiết lập, RECO tự động giải quyết các giao tác nghi ngờ.các dòng tương ứng của bất kỳ giao tác nghi ngờ nào tự động di chuyển tới bảng của CSDL bị treo

## **8.7.MỘT SỐ VÍ DỤ LIÊN QUAN ĐẾN CƠ CHẾ THỎA THUẬN CỦA GIAO TÁC PHÂN TÁN:**

Giả sử có 2 CSDL tách rời SALES và WAREHOUSE, kết nối thông qua CSDL phân tán Oracle.1 người bán hàng xâm nhập vào CSDL sales tới các mẫu tin liên quan và cập nhật CSDL WAREHOUSE, các bước thực hiện sẽ là:

1. ứng dụng phát ra câu SQL:

giả sử:

INSERT INTO ORDERS

UPDATE INVENTORY@WAREHOUSE

INSERT INTO ORDERS

THỎA THUẬN

Chú ý:

- giao tác được khởi tạo tại CSDL sales, vì thế nó là trạm điều hợp toàn cục của giao tác phân tán
- phát biểu INSERT vào sales và UPDATE WAREHOUSE nên cả 2 là DB máy chủ.hơn nữa, sales tại UPDATE WAREHOUSE, nó đóng vai trò máy khách
- trong khi thực hiện SQL, giao tác vẫn chưa thỏa thuận cho đến khi kết thúc thỏa thuận 2 pha

2. ứng dụng phát lệnh thỏa thuận:

nếu kết thúc bằng lệnh ROLLBACK thì trạm điều hợp ra lệnh cho các trạm liên quan để ROLLBACK giao tác và giao tác hoàn tất.nếu phát biểu THỎA THUẬN cơ chế 2 pha bắt đầu với pha chuẩn bị

3. trạm điều hợp xác định điểm thỏa thuận:

giả sử sales được chọn, về nguyên tác trạm nào được chọn đã bàn ở phần trước

#### 4. trạm điều hợp gửi thông điệp chuẩn bị:

sau khi điểm thỏa thuận đã có, trạm điều hợp gửi thông điệp chuẩn bị cho tất cả các trạm trực tiếp tham chiếu, loại bỏ điểm thỏa thuận. trong trường hợp này, chỉ có WAREHOUSE có thông điệp

trạm nhận thông điệp cố gắng để chuẩn bị, nếu nó thấy có thể, nó sẽ ghi các thông tin trong redo log, trạm đã chuẩn bị thành công

- nếu trạm điều hợp nhận bất kỳ thông điệp nào với giá trị hủy bỏ, nó gửi tất cả các trạm ROLLBACK và cơ chế kết thúc
- nếu tất cả các trạm gửi phiếu chuẩn bị, trạm điều hợp sẽ hỏi điểm thỏa thuận để thỏa thuận giao tác

#### 5. điểm thỏa thuận:

sales máy chủ nhận thư báo chuẩn bị của WAREHOUSE, nó hỏi thỏa thuận. ví dụ này chính nó thỏa thuận giao tác cục bộ và ghi lại biến cố vào redo log cục bộ

#### 6. điểm thỏa thuận thông tin cho trạm điều hợp:

về việc thỏa thuận đã xong. điểm thỏa thuận phải nhớ nó đã thỏa thuận cho đến khi trạm điều hợp toàn cục xác nhận rằng giao tác đã thỏa thuận trên tất cả các trạm liên quan đến giao tác phân tán. trạm điều hợp sẽ thông báo tới các trạm trực tiếp thỏa thuận và các trạm tiến hành thỏa thuận đồng thời với các trạm điều hợp và ghi lại trong redo log

#### 7. sau khi tất cả các trạm liên quan và trạm điều hợp đã thỏa thuận giao tác:

trạm điều hợp toàn cục thông báo cho điểm thỏa thuận. điểm thỏa thuận đang chờ thông báo sẽ xóa trạng thái thông tin về giao tác và thông báo lại trạm điều hợp là nó đã hoàn thành, nghĩa là nó quên giao tác vừa rồi. điều này là đúng vì tất cả các trạm đã thỏa thuận thành công. trạm điều hợp nhận thông báo từ điểm thỏa thuận, trạm điều hợp toàn cục kết thúc giao tác bằng 1 hành động “quên” giao tác vừa rồi



## TÀI LIỆU THAM KHẢO

- ✓ Giáo trình phân tích thiết kế các hệ thống thông tin - Nguyễn Văn Vy
- ✓ Slide bài giảng môn cơ sở dữ liệu nâng cao - Nguyễn Thị Thanh Huyền
- ✓ Tài liệu hướng dẫn Oracle trong trang <http://docs.oracle.com/>
- ✓ Các bản slide và giáo trình khác trên mạng internet