Cơ sở dữ liệu phân tán

Đại cương Khoa học máy tính (Trường Đại học Gia Định)

Cơ sở dữ liệu phân tán
Distributed Database

1. Giới thiệu về CSDL phân tán

1.1 Định nghĩa CSDL phân tán

CSDL phân tán là một tập hợp các CSDL cùng hợp tác hoạt động, được lưu trữ trên các máy tính khác nhau (gọi là các trạm/sites) được kết nối với nhau bởi một mạng truyền thông và được quản lý bởi một hệ quản trị CSDL phân tán.

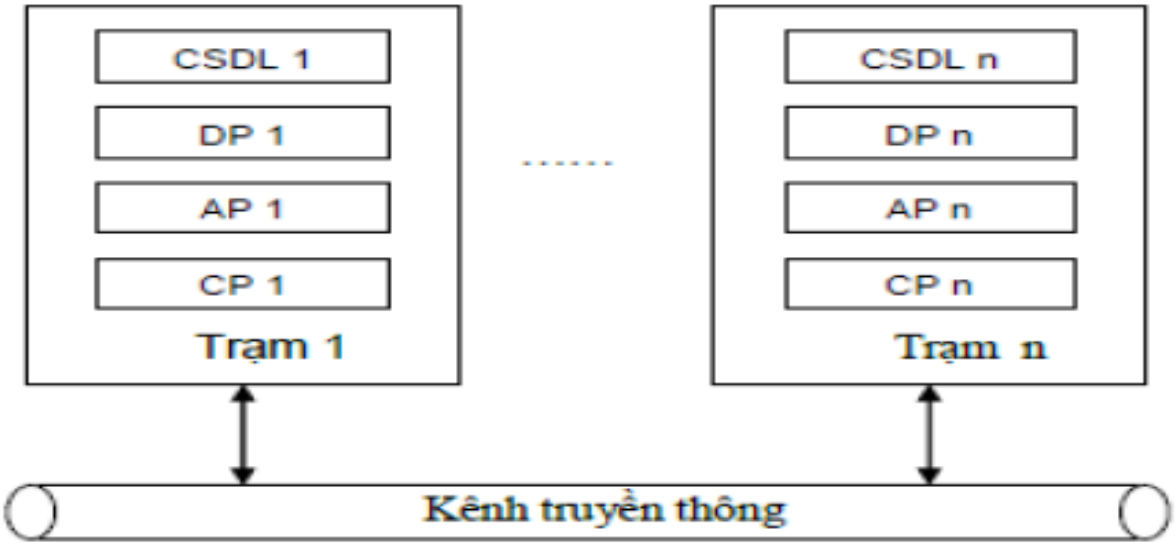
1.2 Các thành phần CSDL phân tán

Về mặt vật lý: Yếu tố chính để phân biệt một cơ sở dữ liệu phân tán với một cơ sở dữ liệu tập trung là:

- **Cơ sở dữ liệu phân tán phải có nhiều máy tính gọi là các trạm (sites/nodes).**
- **Các trạm phải được kết nối bởi một kênh truyền thông nào đó để truyền dữ liệu (trao đổi dữ liệu) bao gồm các giao thức truyền thông.**

Về mặt logic: Phần mềm quản trị CSDL phân tán gồm 3 module chính sau:

- **- Xử lý dữ liệu (DP-Database Processing)**
- **- Xử lý ứng dụng (AP-Application Processing)**
- **- Xử lý truyền thông (CP-Communication Processing)**



1.3 Tính thuần nhất và không thuần nhất

Tính *thuần nhất* của một CSDL phân tán thể hiện ở các tính chất cơ bản:

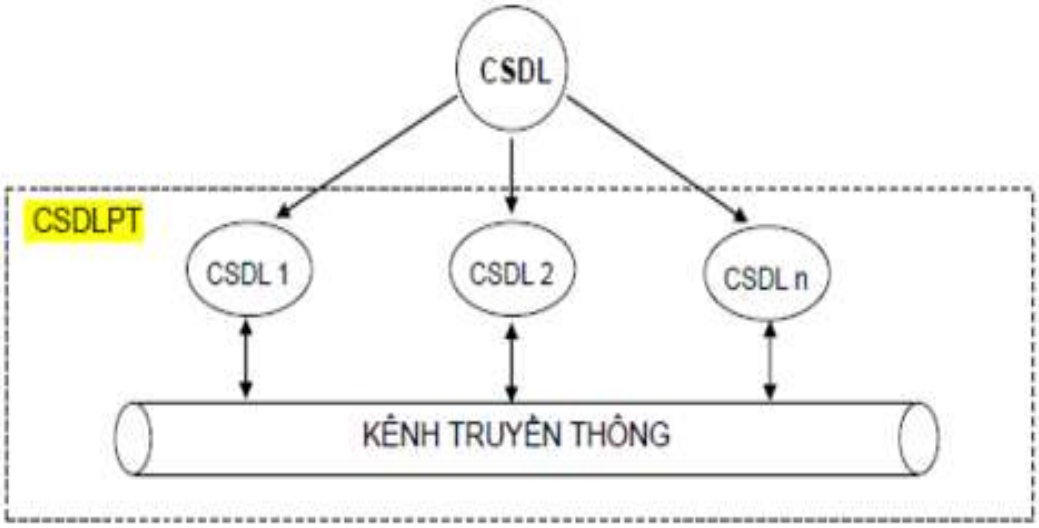
(i) Các CSDL cục bộ đều có cùng mô hình CSDL và có cùng cấu trúc dữ liệu.

(ii) Các CSDL cục bộ được quản trị bởi cùng một hệ quản trị CSDL.

1.3.1. CSDL phân tán thuần nhất (Homogeneous DDB)

- Đảm bảo "Tính *thuần nhất* của một CSDL phân tán"

- Thường là kết quả của một cách thiết kế "từ trên xuống"

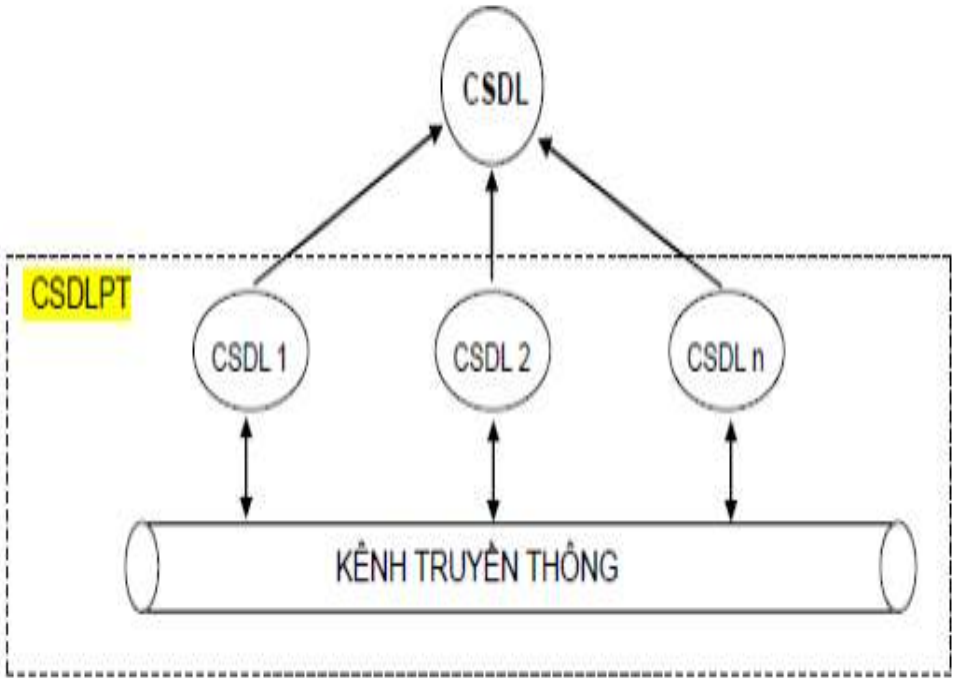


Thiết kế CSDL phân tán thuần nhất

1.3.2. CSDL phân tán không thuần nhất (Hetorogeneous DDB)

- Tính *không thuần nhất* của một CSDL phân tán thể hiện ở việc nó không thỏa một trong (hoặc cả hai) tính chất (i) và (ii)

- Thường là kết quả của một cách thiết kế "từ dưới lên"



1.4 Các mô hình phân tán dữ liệu

Cơ sở dữ liệu từ xa (Remote Database) là một CSDL đặt trên một máy tính khác với máy tính của người dùng và được truy cập nhờ vào các lệnh truyền thông được xác định bởi người dùng.

1.5 Hệ QT CSDL phân tán

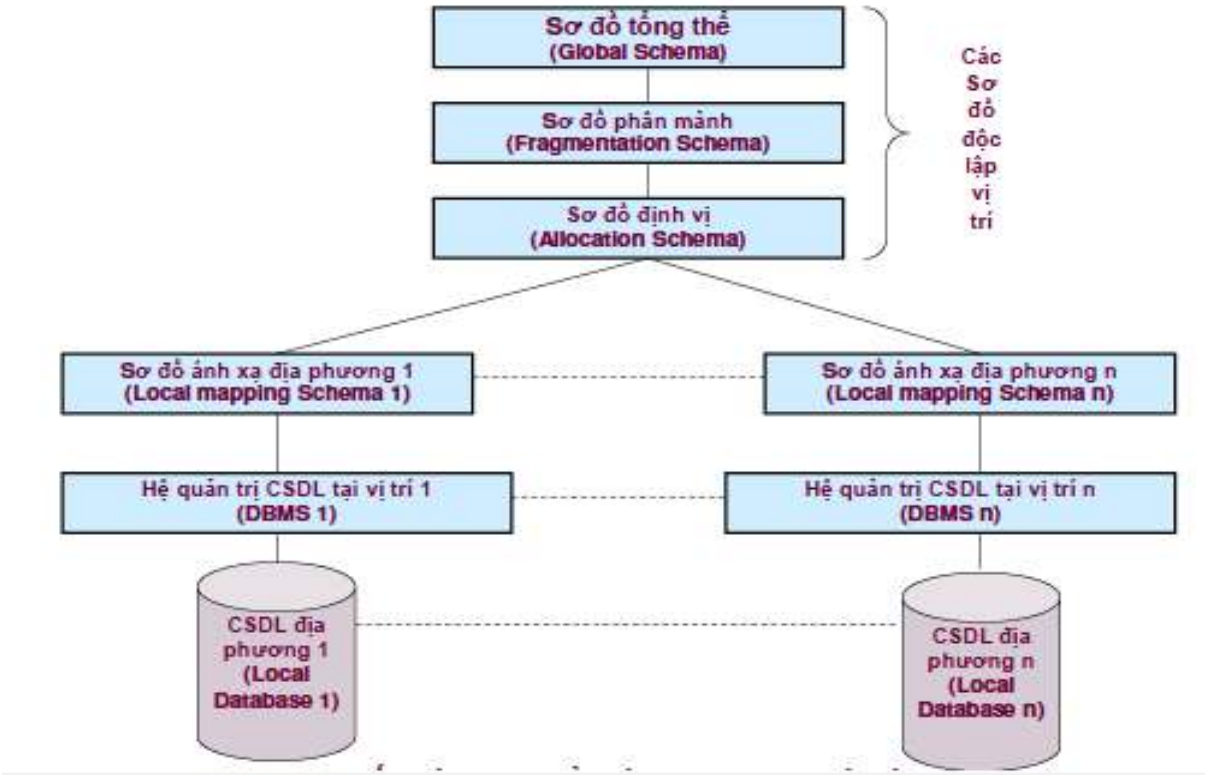
Hệ quản trị CSDL phân tán (Distributed Database Management System-DBMS) được định nghĩa là một hệ thống phần mềm cho phép quản lý các hệ CSDL (tạo lập và điều khiển các truy nhập cho các hệ CSDL phân tán) và làm cho việc phân tán trở nên trong suốt với người sử dụng.

Các thành phần Hệ QT CSDL phân tán:

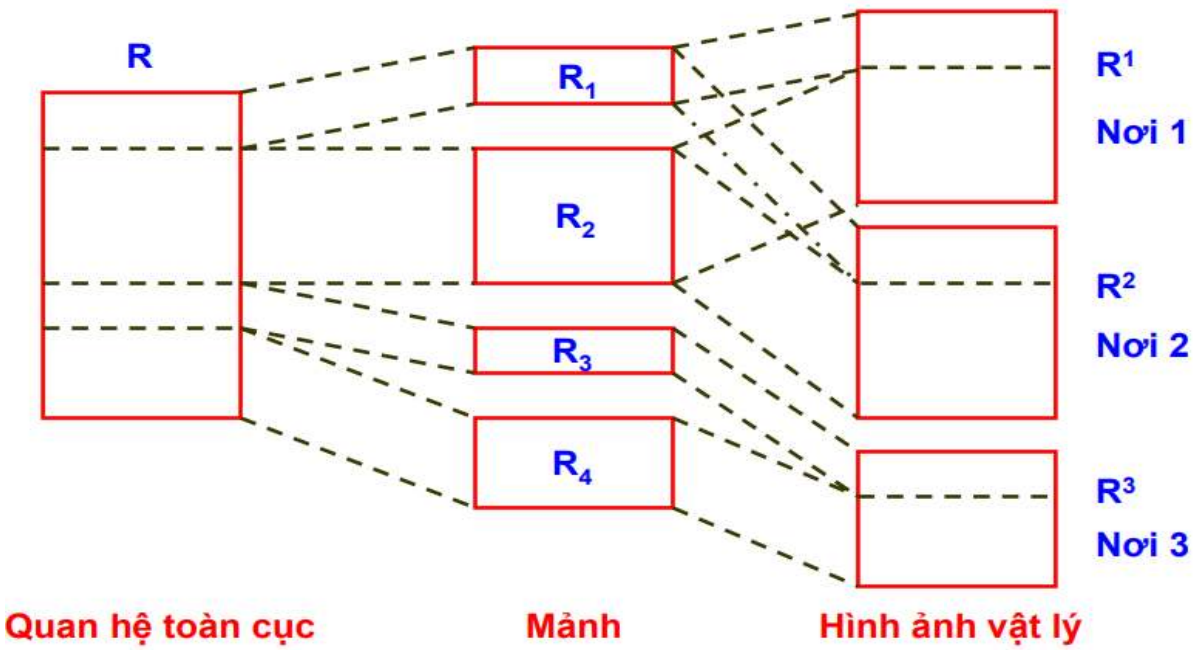
- **Quản trị dữ liệu (Database management): DB**
- **Truyền thông dữ liệu (Data Communication): DC**
- **Từ điển dữ liệu (Data Dictionary): DD dùng để mô tả thông tin về sự phân tán của dữ liệu trên mạng.**
- **Cơ sở dữ liệu phân tán (Distributed Database): DDB**

2 Kiến trúc CSDL phân tán

2.1. Kiến trúc cơ bản của CSDL phân tán

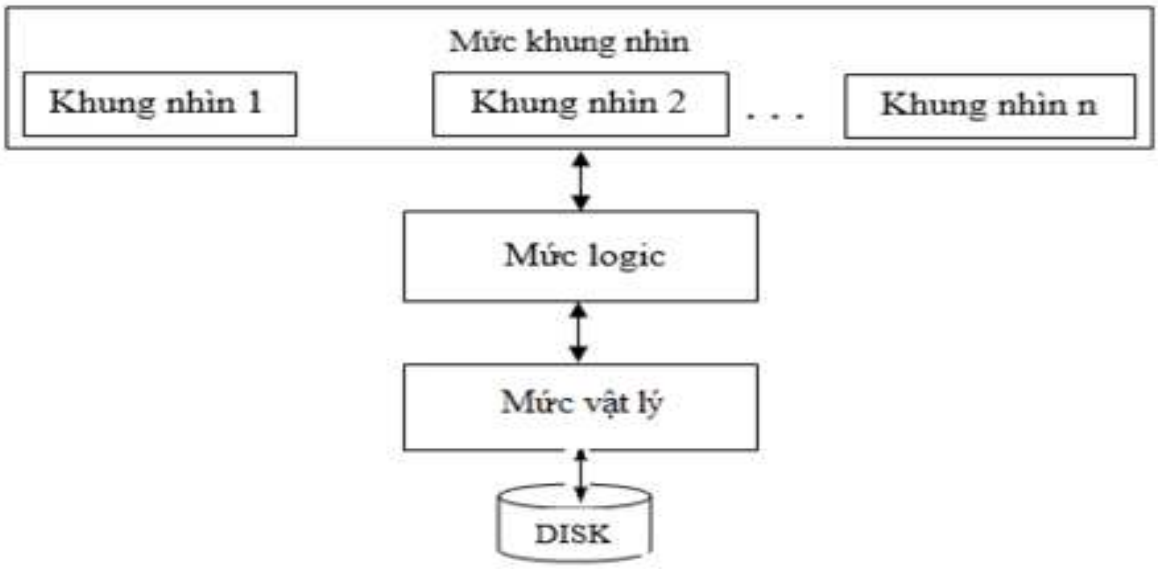


- **Sơ đồ tổng thể: Định nghĩa tất cả các dữ liệu sẽ được lưu trữ trong CSDL phân tán. Trong mô hình quan hệ, sơ đồ tổng thể bao gồm định nghĩa của các tập quan hệ tổng thể.**
- **Sơ đồ phân đoạn: Mỗi quan hệ tổng thể có thể chia thành một vài phần không gối lên nhau được gọi là đoạn (fragments). Có nhiều cách khác nhau để thực hiện việc phân chia này. Ánh xạ (một - nhiều) giữa sơ đồ tổng thể và các đoạn được định nghĩa trong sơ đồ phân đoạn.**
- **Sơ đồ định vị: Các đoạn là các phần logic của quan hệ tổng thể được định vị vật lý trên một hoặc nhiều vị trí trên mạng. Sơ đồ định vị định nghĩa đoạn nào định vị tại các vị trí 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.**
- **Sơ đồ ánh xạ địa phương: ánh xạ các ảnh vật lý và các đối tượng được lưu trữ tại một trạm (tất cả các đoạn của một quan hệ tổng thể trên cùng một vị trí tạo ra một ảnh vật lý)**



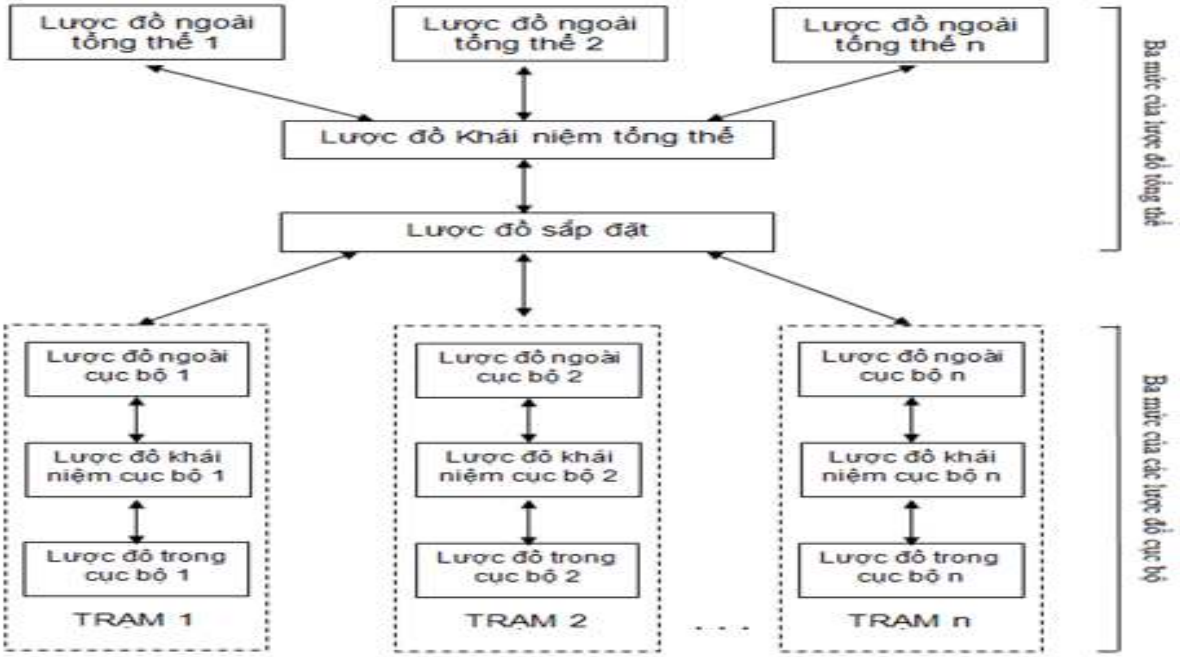
2.2. Kiến trúc của một Hệ CSDL phân tán

- ❑ **Kiến trúc thường dung của 1 CSDL tập trung**



Với CSDL tập trung thì kiến trúc cỉa nó sẽ 3 mức :

- ***Mức khung nhìn:*** **Gồm các lược đồ ngoài, mỗi lược đồ ngoài mô tả chỉ một phần của CSDL, thích hợp với một nhóm người sử dụng nhất định. Mỗi người sử dụng có thể không quan tâm đến toàn bộ CSDL mà chỉ cần một phần thông tin nào đó.**
- ***Mức logic:*** **Gồm các lược đồ khái niệm, các lược đồ này mô tả những dữ liệu nào được lưu trữ trong CSDL, mối quan hệ giữa chúng. Chẳng hạn lược đồ khái niệm chỉ quan tâm đến các quan hệ được lưu trữ trong CSDL và các mối liên hệ giữa chúng chứ không quan tâm đến cách thức lưu trữ các quan hệ này.**
- ***Mức vật lý:*** **Gồm các lược đồ trong, các lược đồ này mô tả dữ liệu được lưu trữ như thế nào trong bộ nhớ thứ cấp. Mức này phản ánh cấu trúc dữ liệu, cách tổ chức tệp, các kỹ thuật nén dữ liệu, cấp phát vùng nhớ…**
- ***2.2.1. Kiến trúc lược đồ của CSDL phân tán***



Kiến trúc lược đồ của CSDL phân tán là sự mở rộng kiến trúc 3 mức cho CSDL tập trung.

Với kiến trúc lược đồ này, tính độc lập dữ liệu dễ dàng được đảm bảo. Mọi thông tin liên quan đến các lược đồ và các phép biến đổi giữa các lược đồ (các ánh xạ) được lưu giữ trong từ điển dữ liệu.

Trước hết ta đưa ra một số khái niệm liên quan đến các lược đồ CSDL phân tán, trong đó các mức của lược đồ mô tả CSDLPT mà độc lập với mọi CSDL cục bộ thì được gọi là mức tổng thể, các mức lược đồ mô tả một CSDL cục bộ được là mức cục bộ.

Ba mức của lược đồ tổng thể:

- **Lược đồ khái niệm tổng thể: Định nghĩa tất cả các quan hệ có trong CSDL phân tán và cung cấp tính độc lập dữ liệu đối với môi trường phân tán.**
- **Các lược đồ ngoài tổng thể: Mỗi lược đồ ngoài tổng thể xác định một khung nhìn của một lớp người dùng nhất định. Các quy tắc định nghĩa khung nhìn được dùng để biến đổi dữ liệu ở mức lược đồ ngoài tổng thể thành dữ liệu ở mức lược đồ khái niệm tổng thể. Nhờ các quy tắc này mà tính độc lập logic của dữ liệu được đảm bảo.**
- **Lược đồ sắp đặt: Xác định chính xác cách thức và vị trí các quan hệ được sắp đặt trên các trạm khác nhau trong mạng. Nó chứa tất cả các thông tin liên quan đến sự định vị, sự phân đoạn và nhân bản dữ liệu, cùng với các quy tắc biến đổi dữ liệu ở mức lược đồ khái niệm tổng thể thành dữ liệu cục bộ có phân đoạn và nhân bản.**

Ba mức của các lược đồ cục bộ: **Cấu trúc lược đồ của một CSDL phân tán gồm 3 mức của lược đồ tổng thể và 3 mức của lược đồ cục bộ, được mô tả trong hình 1.9. - Các lược đồ trong cục bộ: Là mức vật lý của các lược đồ CSDL cục bộ, ý nghĩa và hoạt động giống như các lược đồ trong của CSDL tập trung.**

3 Kỹ thuật phân mảnh, cấp phát và sao lưu dữ liệu trong thiết kế CSDL phân tán

3.1 Các chiến lược phân tán dữ liệu

- **Có 4 chiến lược phân tán dữ liệu cơ bản:**
  - **Tập trung dữ liệu**
  - **Chia nhỏ dữ liệu**
  - **Sao lặp dữ liệu**

Phương thức lai

3.1.1 Tập trung dữ liệu:

Tất cả các dữ liệu được tập trung một chỗ.

Nhược điểm:

- **Dữ liệu không sẵn sàng cho truy nhập từ xa**
- **Chi phí truyền thông lớn, thường làm cực đại việc truy nhập dữ liệu tới nơi tập trung.**
- **Toàn bộ hệ thống ngừng khi cơ sở dữ liệu bị sự cố**

3.1.2 Chia nhỏ dữ liệu:

- **Cơ sở dữ liệu được chia thành các phần nhỏ liên kết nhau (không trùng lặp).**
- **Mỗi phần dữ liệu được đưa đến các trạm một cách thích hợp để sử dụng.**

3.1.3 Sao lặp dữ liệu:

- **CSDL được nhân thành nhiều bản từng phần hoặc đầy đủ và được đặt ở nhiều trạm trên mạng.**
- **Nếu bản sao của CSDL được lưu giữ tại mọi trạm của hệ thống ta có trường hợp sao lặp đầy đủ.**
- **Hiện nay có nhiều kỹ thuật mới cho phép tạo bản sao không đầy đủ phù hợp với yêu cầu dữ liệu ở mỗi trạm và một bản đầy đủ được qiuản lý ở server.**
- **Sau một khoảng thời gian nhất định các bản sao được làm đồng bộ với bản chính bằng một ứng dụng nào đó.**

3.1.4 Phương thức lai:

- **Cơ sở dữ liệu được phân thành nhiều phần: quan trọng và không quan trọng.**
- **Phần ít quan trọng được lưu giữ một nơi**
- **Phần quan trọng được lưu trữ ở nhiều nơi khác.**

	Nhân bản hoàn toàn	Nhân bản một phần	Phân hoạch
Xử lý vấn tin	Dễ	Cùng mức độ khó	
Quản lý thư mục	Dễ hoặc không tồn tại	Cùng mức độ khó	
Điều khiển đồng thời	Vừa phải	Khó	Dễ
Độ tin cậy	Rất cao	Cao	Thấp
Tính thực tế	Có thể áp dụng	Thực tế	Có thể áp dụng

So sánh các lựa chọn nhân bản

3.2. Phân mãnh dữ liệu

Phân mảnh quan hệ là gì?

Việc chia một quan hệ thành nhiều quan hệ nhỏ hơn được gọi là phân mảnh quan hệ.

3.2.1 Các lý do phân mảnh

Khung nhìn hoặc đơn vị truy xuất của các ứng dụng

Việc 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.

Việc phân mảnh các quan hệ sẽ cho phép thực hiện song song một câu vấn tin bằng cách chia nó ra thành một tập các câu vấn tin con hoạt tác trên các mảnh.

Nếu các ứng dụng có các khung nhìn được định nghĩa trên một quan hệ cho trước nằm tại những vị trí khác thì có hai cách chọn lựa đơn vị phân tán:

+ hoặc là toàn bộ quan hệ

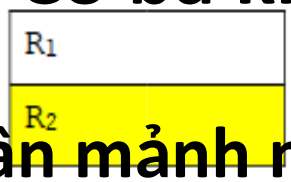
+ hoặc quan hệ được lưu ở một vị trí có chạy ứng dụng.

Khuyết điểm của việc phân mảnh:

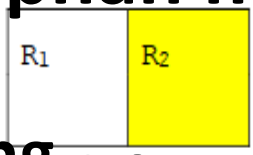
- **Nếu một khung nhìn đòi hỏi thông tin ở nhiều mảnh thì việc truy xuất dữ liệu để nối lại sẽ có chi phí cao.**
- **Liên quan đến tính toàn vẹn dữ liệu**

3.2.2. Các kiểu phân mảnh

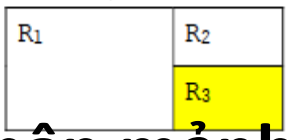
Có ba kiểu phân mảnh:



Phân mảnh ngang



Phân mảnh



Phân mảnh hỗn hợp

3.2.3. Các quy tắc phân mảnh

Có 3 quy tắc

- **Tính đầy đủ (completeness)**
- **Tính tái thiết được (reconstruction):**
- **Tính tách biệt (disjointness)**

Tính đầy đủ (completeness): Nếu một quan hệ **R** được phân rã thành các mảnh **R_1, R_2, ..., R_k** thì mỗi mục dữ liệu có trong **R** phải có trong ít nhất một mãnh **R_i** nào đó.

Tính tái thiết được (reconstruction): Nếu một quan hệ **R** được phân rã thành các mảnh **R_1, R_2,... , R_n** thì cần định nghĩa một phép toán quan hệ sao cho:

$$R = \nabla R_i , i \in [1,n]$$

Tính tách biệt (disjointness): Nếu một quan hệ **R** được phân rã thành các mảnh **R_1, R_2,... ,R_n** và mục dữ liệu **d_i** nằm trong mảnh **R_j** thì nó sẽ **không nằm** trong một mảnh **R_k** khác ($k \neq j$)

" i ¹ k và i , k Î [1,n]: R_i Ç R_k = Æ.

3.3. Các phương pháp phân mảnh

3.3.1. Phân mảnh ngang

Phân mảnh ngang một quan hệ tổng thể n-bộ R là tách R thành các quan hệ con n-bộ R_1, R_2, ..., R_k sao cho quan hệ R có thể được khôi phục lại từ các quan hệ con này bằng phép hợp:

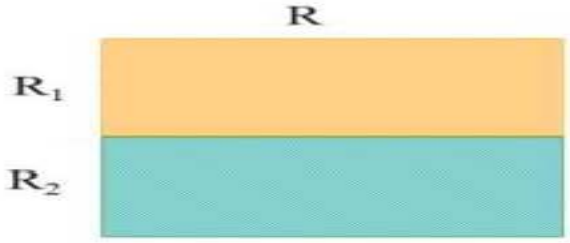
$$R = R_1 È R_2 È ... È R_k$$

Các loại phân mảnh ngang

- **Phân mảnh ngang nguyên thủy** (Primary Horizontal Fragmentation)
- **Phân mảnh ngang dẫn xuất** (Derived Horizontal Fragmentation)

3.3.1.1. Phân mảnh ngang nguyên thủy

Phân mảnh ngang nguyên thủy (Primary Horizontal Fragmentation) là sự phân chia các bộ của một quan hệ toàn cục thành các tập hợp con dựa vào các thuộc tính của quan hệ này, mỗi tập hợp con được gọi là một mảnh ngang (horizontal fragment).



Cách xác định phân mảnh ngang nguyên thủy:

$R_i = s_{F_i}(R)$, $1 \le i \le z$.

trong đó: F_i là điều kiện chọn để có được mảnh R_i

R là quan hệ toàn cục

Vị từ: là điều kiên trong phép chọn - ***vị từ định tính***

Ví dụ 1: Phân mảnh ngang nguyên thủy quan hệ toàn cục **DA** dựa vào **VT** (**vị trí)** của các dự án

DA

Bộ	MADA	TENDA	NS	VT
u_1	D1	Xây dựng phần mềm quản lý lương	20000	Nam Định
u_2	D2	Thiết kế trang Web bán hàng	12000	Hà Nội
u_3	D3	Nâng cấp hệ thống mạng	28000	Hà Nội

Phân mảnh ngang nguyên thủy quan hệ toàn cục **DA** dựa vào **vị trí** của các dự án được xác định như sau:

$DA_1 = s_{VT = \text{“Nam Định”}}(DA)$

$DA_2 = s_{VT = \text{“Hà Nội”}}(DA)$

Do đó, các vị từ định tính là:

q1: VT = “Nam Định”

q2: VT = “Hà Nội”

Ví dụ 1: Kết quả của các mảnh như sau:

Điều kiện đầy đủ

$DA_1 \subseteq DA$

$DA_2 \subseteq DA$

Điều kiện tái tạo

$DA = DA_1 \cup DA_2$

Điều kiện tách biệt

$DA1 \cap DA2 = \varnothing$

DA_1

MADA	TENDA	NS	VT
D1	Xây dựng phần mềm quản lý lương	20000	Nam Định

DA_2

MADA	TENDA	NS	VT
D2	Thiết kế trang Web bán hàng	12000	Hà Nội
D3	Nâng cấp hệ thống mạng	28000	Hà Nội

➔ Đảm bảo các điều kiện phân mảnh

Các bước thiết kế

Bước 1. Tìm tập các vị từ - chú ý các vị từ phải thỏa mãn tính đầy đủ và cực tiểu .

➔ Dùng thuật toán **COM_MIN**

Thuật toán **COM_MIN**: Cho phép tìm tập các vị từ đầy đủ và cực tiểu Pr' từ Pr. Chúng ta tạm quy ước:

Quy tắc 1: Q*uy tắc cơ bản về tính đầy đủ và cực tiểu , nó khẳng định rằng một quan hệ hoặc một mảnh được phân hoạch "thành ít nhất hai phần và chúng được truy xuất khác nhau bởi ít nhất một ứng dụng*"

Thuật toán COM-MIN

Input : R: quan hệ; Pr: tậpcác vị từ đơn giản;
Output: Pr': tập các vị từ cực tiểu và đầy đủ;
Declare
F: tập các mảnh hội sơ cấp;
Begin
Pr'=ϕ; F = ϕ;
For each vị từ p $\in$ Pr if p phân hoạch R theo Quy tắc 1 then
Begin
Pr': = Pr'$\cup$ p;
Pr: = Pr – p;
F: = F $\cup$ p; {f_i là mảnh hội sơ cấp theo p_i }
End; {Chúng ta đã chuyển các vị từ có phân mảnh R vào Pr'}
Repeat
For each p$\in$ Pr
if p phân hoạch một mảnh f_k của Pr' theo quy tắc 1 then
Begin
Pr': = Pr'$\cup$ p;
Pr: = Pr – p;
F: = F $\cup$ p;
End;
Until Pr' đầy đủ {Không còn p nào phân mảnh f_k của Pr'}
For each p $\in$ Pr', if $\exists$p' mà p<=>p' then
Begin
Pr':= Pr'-p;
F:= F - f;
End;
End. {COM_MIN}

- Thuật toán bắt dầu bằng cách tìm một vị từ có liên đới và phân hoạch quan hệ đã cho.

- Vòng lặp **Repeat-until** thêm các vị từ có phân hoạch các mảnh vào tập này, bảo đảm tính đầy đủ của Pr'.
- Đoạn cuối kiểm tra tính cực tiểu của Pr'. Vì thế cuối cùng ta có tập Pr' là cực tiểu và đầy đủ.

Bước 2. Suy dẫn ra tập các vị từ hội sơ cấp có thể được định nghĩa trên các vị từ trong tập Pr'

Bước 3. Loại bỏ một số mảnh vô nghĩa. Điều này được thực hiện bằng cách xác định những vị từ mâu thuẫn với tập các phép kéo theo

Thuật toán phân mảnh ngang nguyên thủy

Thuật toán PHORIZONTAL

Input: R: quan hệ; Pr: tập các vị từ đơn giản;

Output: M: tập các vị từ hội sơ cấp;

Begin

Pr':= COM_MIN(R, Pr);

Xác định tập M các vị từ hội sơ cấp;

Xác định tập I các phép kéo theo giữa các $p_i \in Pr'$;

For each $m_i \in M$ **do**

Begin

IF m_i mâu thuẫn với I **then**

M:= M-m_i

End;

End. {PHORIZONTAL}

3.3.1.2. Phân mảnh ngang dẫn xuất

Phân mảnh ngang dẫn xuất (Derived Horizontal Fragmentation): là sự phân chia các bộ của một quan hệ toàn cục thành các tập hợp con (được gọi là các mảnh ngang) dựa vào phân mảnh ngang của một quan hệ khác (được gọi là quan hệ chủ).

$R_i = R |>< S_i$, $1 \le i \le w$

Trong đó:

- w là số lượng các mảnh được định nghĩa trên R,

- $S_i = \sigma_{Fi}(S)$ với F_i là công thức định nghĩa mảnh ngang nguyên thuỷ S_i

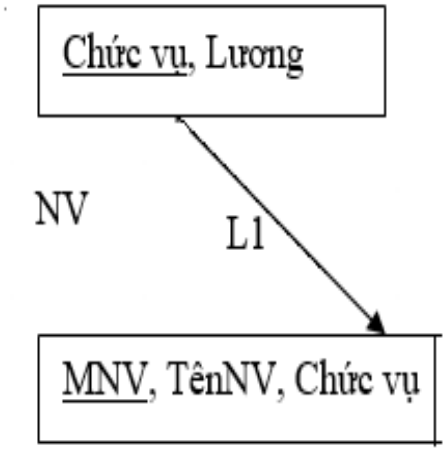
CHITRA

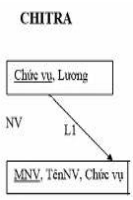
NHANVIEN

MNV	TênNV	Chức vụ
E1	J.Doe	Kỹ sư điện
E2	M.Smith	Phân tích
E2	M.Smith	Phân tích
E3	A.Lee	Kỹ sư cơ khí
E3	A.Lee	Kỹ sư cơ khí
E4	J.Miller	Programmer
E5	B.Casey	Phân tích hệ thống
E6	L.Chu	Kỹ sư điện
E7	R.david	Kỹ sư cơ khí
E8	J.Jones	Phân tích hệ thống

CHITRA

Chức vụ	Lương
Kỹ sư điện	40000
Phân tích hệ thống	34000
Kỹ sư cơ khí	27000
Lập trình	24000





Hai mảnh **Nhân viên$_1$** và **Nhân viên$_2$** được định nghĩa như sau:

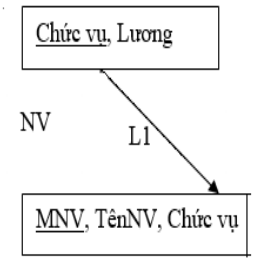
NV_1=NHANVIEN |>< $CHITRA_1$

NV_2=NHANVIEN |>< $CHITRA_2$

Trong đó $CT_1=\sigma_{Lương \le 30000}(CHITRA)$

$CT_2=\sigma_{Lương>30000}(CHITRA)$

CHITRA



Kết quả phân mảnh ngang dẫn xuất của quan hệ **NHANVIEN** như sau:

NV_1

MNV	TênNV	Chức vụ
E3	A.Lee	Kỹ sư cơ khí
E4	J.Miller	Lập trình viên
E7	R.David	Kỹ sư cơ khí

NV_2

MNV	TênNV	Chức vụ
E1	J.Doe	Kỹ sư điện
E2	M.Smith	Phân tích
E5	B.Casey	Phân tích hệ thống
E6	L.Chu	Kỹ sư điện
E8	J.Jones	Phân tích hệ thống

$CT_1=\sigma_{Lương \le 30000}(CHITRA)$

CT_1

Chức vụ	Lương
Kỹ sư cơ khí	27000
Lập trình	24000

$CT_2=\sigma_{Lương>30000}(CHITRA)$

CT_2

Chức vụ	Lương
Kỹ sư điện	40000
Phân tích hệ thống	34000

Chú ý

Muốn thực hiện **phân mảnh ngang dẫn xuất**, chúng ta **cần ba nguyên liệu** (input):

1. Tập các phân hoạch của quan hệ chủ nhân (Thí dụ: CT1, CT2).
2. Quan hệ thành viên
3. Tập các vị từ nối nửa giữa chủ nhân và thành viên (Chẳng hạn CT.Chucvu = NV.Chucvu).

Quyết định chọn cách phân mảnh nào cần dựa trên hai tiêu chuẩn sau:

1. Phân mảnh có đặc tính nối tốt hơn
2. Phân mảnh được sử dụng trong nhiều ứng dụng hơn.

3.3.2. Phân mảnh dọc

Phân mảnh dọc: một quan hệ tổng thể n-bộ **R** là tách **R** thành các quan hệ con $\mathbf{R_1, R_2, ..., R_k}$ sao cho quan hệ R có thể được khôi phục lại từ các quan hệ con này bằng phép nối:

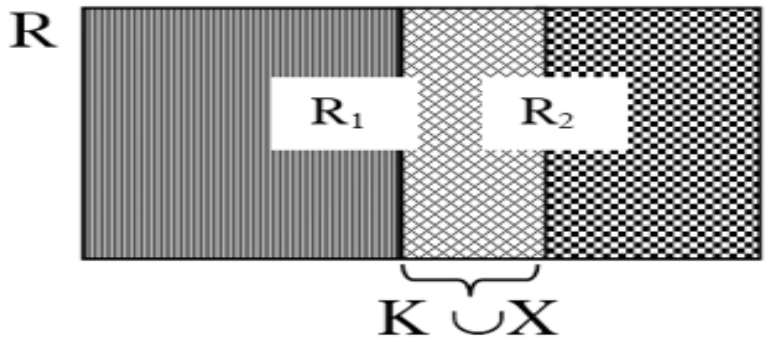
$\mathbf{R = R_1 \bowtie R_2 \bowtie ..., \bowtie R_k}$

Cách xác định một mảnh: Dùng phép chiếu (∏)

Các loại phân mảnh dọc

Phân mảnh dọc gom tụ dư thừa:

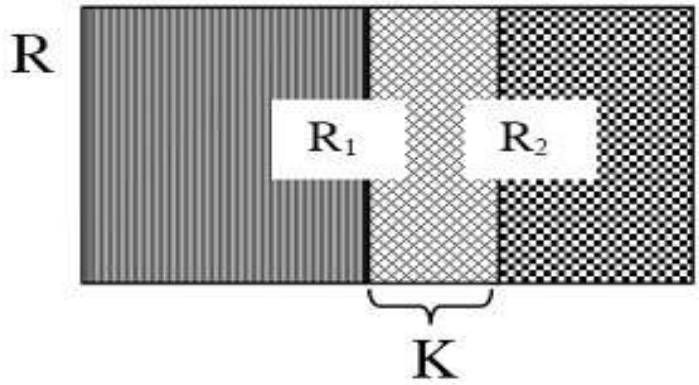
$\mathbf{Attr(R_1) Ç Attr(R_2) Ç … Ç Attr(R_n) = K È X}$



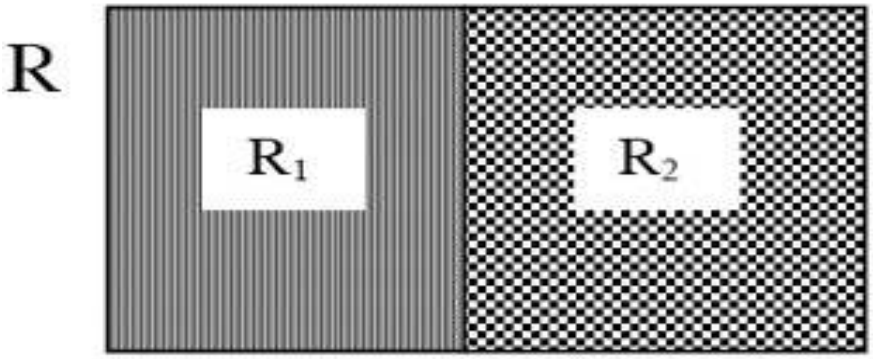
$\mathbf{R = R_1 >\!\!< _{R1.K = R2.K} \Pi _{Attr(R2)-X}(R_2)}$

Phân mảnh dọc gom tụ không dư thừa:

$\mathbf{Attr(R_1) Ç Attr(R_2) Ç … Ç Attr(R_n) = K}$



Phân mảnh dọc tách biệt:
Attr(R_1) Ç Attr(R_2) Ç … Ç Attr(R_n) = ⊘



Ví dụ:

DUAN

MADA	TENDA	NGANSACH	ĐIACHI
D1	CSDL	150000	Huế
D2	CÀI ĐẶT	135000	Hà nội
D3	BẢO TRÌ	250000	Hà nội
D4	PHÁT TRIỂN	310000	HCMC

xét phép toán đại số quan hệ:

$DUAN_1 = \prod_{\$1,\$3,\$4}(DUAN)$. $DUAN_2 = \prod_{\$1,\$2}(DUAN)$
$DUAN_1$

MADA	NGANSACH	ĐIACHI
D1	150000	Huế
D2	135000	Hà nôi
D3	250000	Hà nội
D4	310000	HCMC

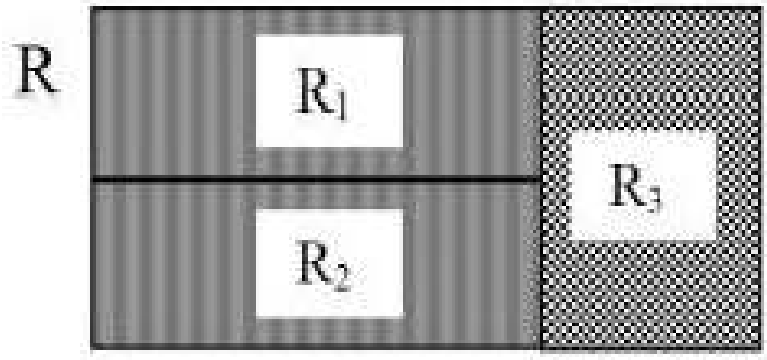
$DUAN_2$

MADA	TENDA
D1	CSDL
D2	CÀI ĐẶT
D3	BẢO TRÌ
D4	PHÁT TRIỂN

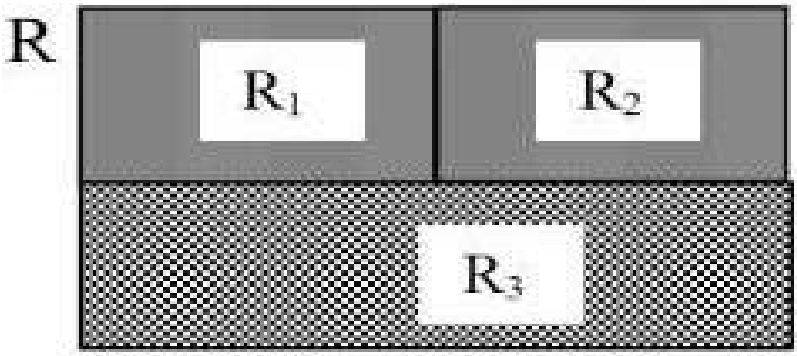
$DUAN_1 \subseteq DUAN$
$DUAN_2 \subseteq DUAN$
$DUAN = DUAN_1 \bowtie DUAN_4$

3.3.3. Phân mảnh hỗn hợp

Phân mảnh hỗn hợp: là cách áp dụng các phép phân mảnh ngang, dọc một cách đệ quy sao cho sao cho mỗi lần thực hiện phân mảnh phải thoả mãn các điều kiện đúng đắn.
Phân mảnh dọc trước, ngang sau:



Phân mảnh ngang trước, dọc sau:



Cách xác định một mảnh: dùng phép chiếu $\prod$, chọn σ

Điều kiện đúng đắn để phân mảnh hỗn hợp:
Điều kiện đầy đủ: Luôn được thỏa mãn
Điều kiện tái tạo: Áp dụng các quy tắc theo thứ tự ngược lại
Điều kiện tách biệt: Tùy từng trường hợp
Ví dụ:

DUAN

MADA	TENDA	NGANSACH	ĐIACHI
D1	CSDL	150000	Huế
D2	CÀI ĐẶT	135000	Hà nội
D3	BẢO TRÌ	250000	Hà nội
D4	PHÁT TRIỂN	310000	HCMC

xét phép toán đại số quan hệ:
$DUAN_3 = \sigma$ NGANSACH ≤ 200000 $\prod_{\$1,\$3}$(DUAN)

MADA	NGANSACH	ĐIACHI
D1	150000	Huế
D2	135000	Hà nội

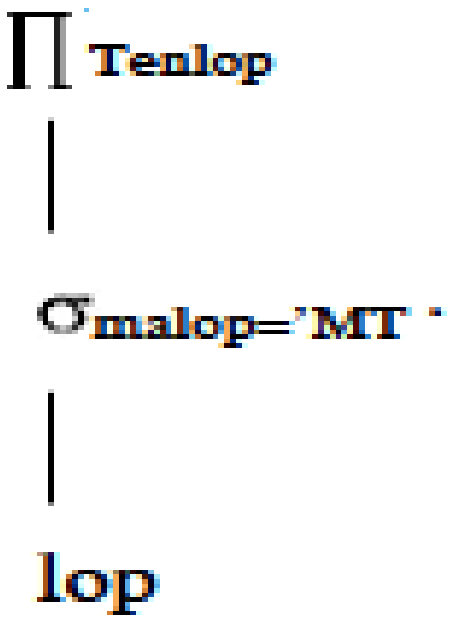
$DUAN_4 = \sigma_{NGANSACH > 200000}$ $\prod_{\$1,\$2}$(DUAN)

MADA	TENDA
D3	BẢO TRÌ
D4	PHÁT TRIỂN

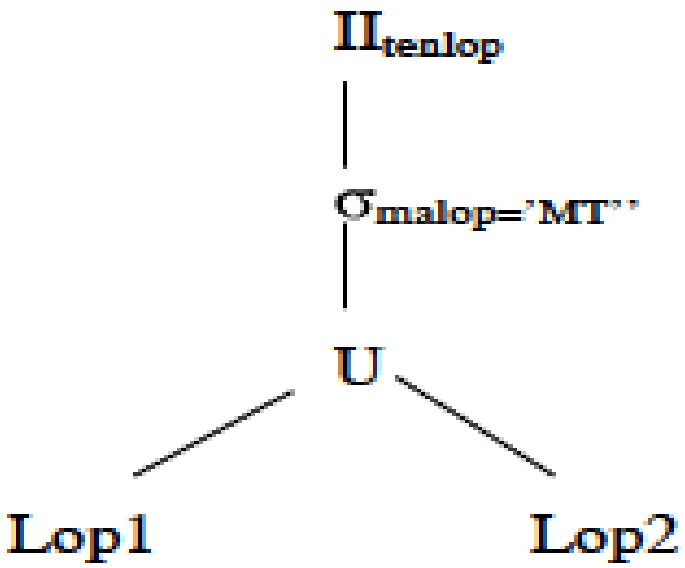
Chú ý
Quyết định chọn cách phân mảnh nào cần dựa trên hai tiêu chuẩn sau:
1. Phân mảnh có đặc tính nối tốt hơn
2. Phân mảnh được sử dụng trong nhiều ứng dụng hơn.

4, **Xử lý và tối ưu truy vấn trong CSDL phân tán**
4. Tối ưu hóa truy vấn trong CSDL phân tán
Tối ưu hoá trong cơ sở dữ liệu phân tán nghĩa là *giảm* ***chi phí bộ nhớ*** *trung gian, giảm* ***thời gian truy vấn*** *cũng như giảm* ***thời gian truyền dữ*** *liệu* trong các truy vấn phân tán.
Cho một CSDL QLSV với các nội dung sau
Sinhvien (masv, hoten, tuoi, malop)
Lop(malop, tenlop, malt, tenkhoa)
Monhoc(mamh, tenmh)
Hoc(masv, mamh, Diem)
4.1 Truy vấn; Biểu thức chuẩn tắc của truy vấn
4.1.1 Truy vấn

Truy vấn (query) là một biểu thức được biểu diễn bằng một ngôn ngữ thích hợp và dùng để xác định một phần dữ liệu được chứa trong cơ sở dữ liệu.
Một truy vấn có thể được biểu diễn bởi một **cây toán tử**. Một cây toán tử operator tree của một truy vấn, còn được gọi là **cây truy vấn** (query tree) hoặc **cây đại số quan hệ** (relational algebra tree)
Cây đại số quan hệ (relational algebra tree: là một cây mà một nút lá là một quan hệ trong cơ sở dữ liệu, và một nút khác lá (nút trung gian hoặc nút gốc) là một quan hệ trung gian được tạo ra bởi một phép toán đại số quan hệ.
Chuỗi các phép toán đại số quan hệ được thực hiện từ các nút lá đến nút gốc để tạo ra kết quả truy vấn.
Ví dụ: Xét truy vấn cho biết **tên lớp** của **lớp** có **mã lớp** là **MT** . Truy vấn này, có thể được biểu diễn bởi một biểu thức đại số quan hệ như sau :
$\prod_{\text{Tenlop}}(\sigma_{\text{malop='MT'}}(\text{lop}))$



4.1.2. Biểu thức chuẩn tắc của truy vấn
Biểu thức chuẩn tắc: của một biểu thức đại số quan hệ trên lược đồ toàn cục là một biểu thức có được bằng cách thay thế mỗi tên quan hệ toàn cục xuất hiện trong biểu thức bởi biểu thức tái lập của quan hệ toàn cục này
Ví dụ : Giả sử chúng ta có hai khoa tên là CNTT và VT. Quan hệ **lop** được **phân mảnh ngang** dựa vào **tenkhoa** thành hai mảnh **lop1** và **lop2**
Lop1 = $\sigma_{\text{tenkhoa='CNTT'}}$(lop)
Lop2 = $\sigma_{\text{tenkhoa='VT'}}$(lop)
Biểu thức tái lập của quan hệ toàn cục **lop** la
Lop = lop1 U lop2
Biểu thức chuẩn tắc của biểu thức truy vấn là :
$\Pi_{\text{tenlop}}(\sigma_{\text{malop='MT'}}(\text{lop1 U lop2}))$
Thay thế quan hệ toàn cục **lop** trong cây toán tử bởi biểu thức tái lập ở trên, chúng ta được cây toán tử như sau:



4.2. Tối ưu hóa truy vấn trong CSDL tập trung
Tối ưu hoá trong cơ sở dữ liệu tập trung gồm các bước sau:
Bước 1- Kiểm tra ngữ pháp
Bước 2- Kiểm tra sự hợp lệ
Bước 3- Dịch truy vấn
Bước 4- Tối ưu hóa biểu thức đại số quan hệ
Bước 5- Chọn lựa chiến lược truy xuất
Bước 6-Tạo sinh mã
4.3. Tối ưu hóa truy vấn trong CSDL phân tán
Tối ưu hoá truy vấn trong cơ sở dữ liệu phân tán bao gồm một số bước đầu của tối ưu hóa truy vấn trong cơ sở dữ liệu tập trung và một số bước tối ưu hóa có liên quan đến sự phân tán dữ liệu.
Gồm các bước sau:
Bước 1: Phân rã truy vấn
Bước 2: Định vị dữ liệu
Bước 3: Tối ưu hóa truy vấn toàn cục
Bước 4: Tối ưu hóa truy vấn cục bộ
4.3.1. Bước 1 - Phân rã truy vấn
Để biến đổi một truy vấn viết bằng ngôn ngữ cấp cao, chẳng hạn SQL, thành một biểu thức đại số quan hệ tương đương:
- Theo nghĩa chúng cho ra cùng một kết quả và
- Hiệu quả (theo nghĩa loại bỏ các phép toán đại số quan hệ không cần thiết, giảm vùng nhớ trung gian).
Tối ưu hóa truy vấn trên lược đồ toàn cục bao gồm 4 bước sau:
4.3.1.1 Phân tích truy vấn
DBMS **kiểm tra ngữ pháp của truy vấn**, kiểm tra sự tồn tại của các đối tượng dữ liệu (tên cột, tên bảng, vv…) của truy vấn trong cơ sở dữ liệu, phát hiện các phép toán trong truy vấn bị sai về kiểu dữ liệu, **điều kiện** của mệnh đề **WHERE** có thể bị sai về ngữ nghĩa.
Phân tích điều kiện của mệnh đề **WHERE** để phát hiện truy vấn bị sai. Có hai loại sai:
- Sai về kiểu dữ liệu (type incorrect)
- Sai về ngữ nghĩa (semantically incorrect)
Truy vấn bị sai về kiểu dữ liệu

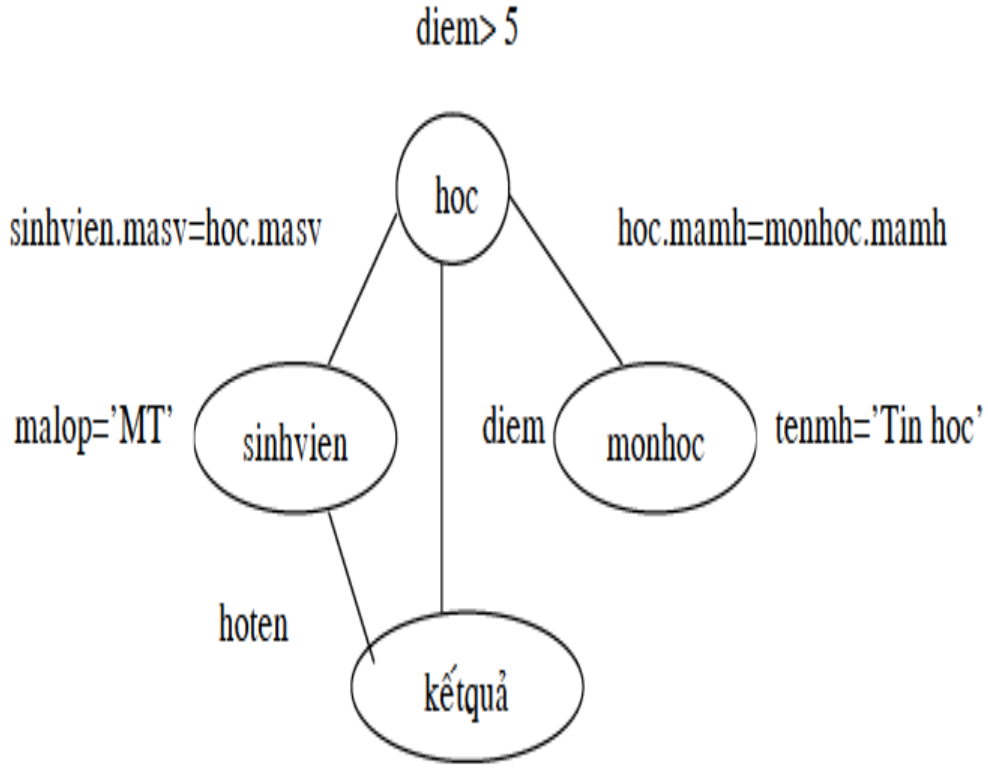
Một truy vấn bị sai về kiểu dữ liệu nếu các thuộc tính của nó hoặc các tên quan hệ không được định nghĩa trong lược đồ qoàn cục, hoặc nếu các phép toán được áp dụng cho các thộc tính bị sai về kiểu dữ liệu.
Để giải quyết cho vấn đề này, trong **lược đồ toàn cục** chúng ta **phải mô tả kiểu dữ liệu** của các **thuộc tính** của các quan hệ.
Ví dụ: **Truy vấn bị sai về kiểu dữ liệu**
Xét truy vấn sau:

```
SELECT mssv, hoten     FROM sinhvien
WHERE masv=123
```

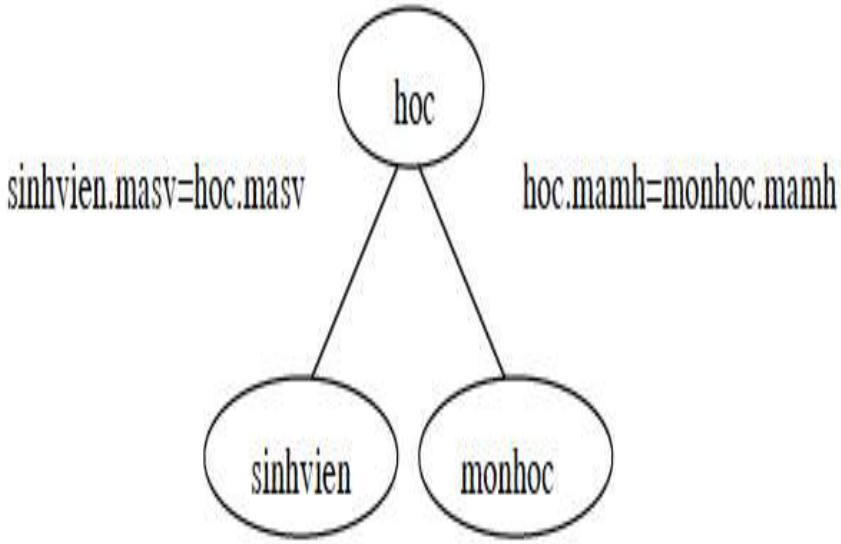
Truy vấn trên có 2 lỗi sai:
(1) mssv không tồn tại trong quan hệ sinhvien, và
(2) masv thuộc kiểu chuỗi không thể so sánh với hằng số 123
Truy vấn bị sai về ngữ nghĩa
Một truy vấn bị sai về ngữ nghĩa nếu nó có chứa các thành phần không tham gia vào quá trình tạo ra kết quả của truy vấn.
Để phát hiện một **truy vấn bị sai về ngữ nghĩa**, chúng ta dùng một **đồ thị truy vấn** (query graph) hoặc **đồ thị kết nối quan hệ** (relation connection graph) cho các truy vấn có chứa các **phép chọn, phép chiếu và phép kết**.
Ví dụ: **Truy vấn bị sai về ngữ nghĩa**
Xét truy vấn sau:

```
SELECT hoten, diem
FROM sinhvien, hoc, monhoc
WHERE sinhvien.masv=hoc.masv
        AND hoc.mamh=monhoc.mamh
        AND malop='MT'
        AND diem > 5
        AND tenmh = 'Tin hoc';
```

Ví dụ: **Truy vấn bị sai về ngữ nghĩa**



Đồ thị truy vấn



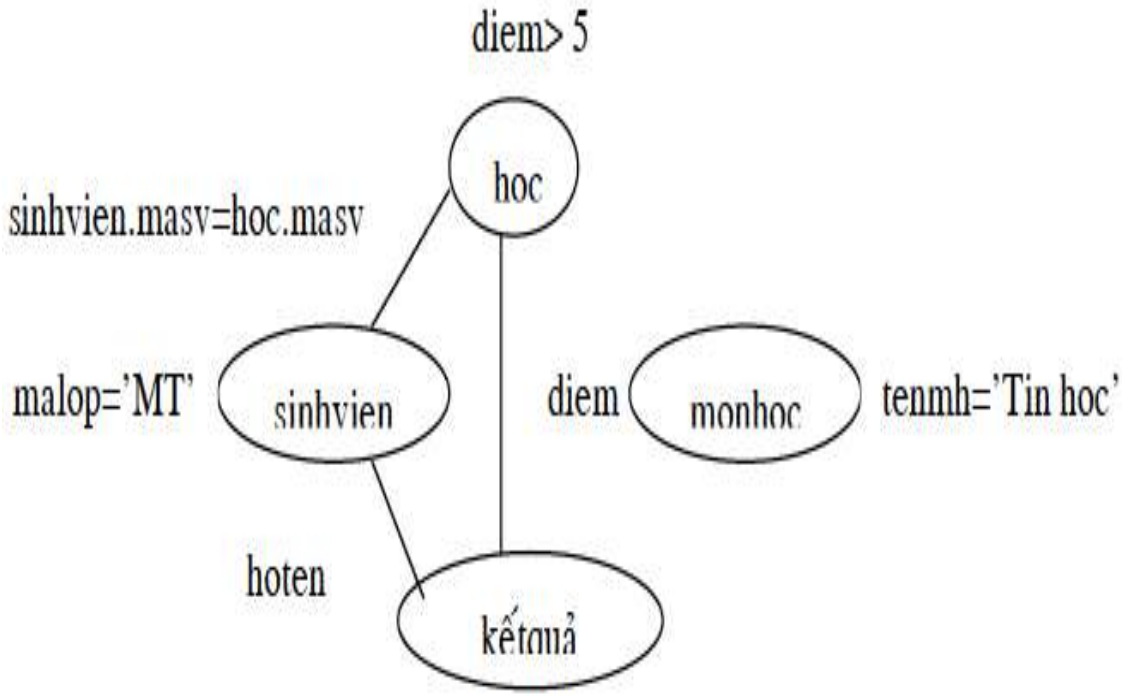
Đồ thị kết nối

- Một truy vấn bị sai về ngữ nghĩa nếu đồ thị truy vấn của nó là không liên thông.
- Đồ thị không liên thông là một đồ thị bao gồm nhiều thành phần liên thông, mỗi thành phần liên thông là một đồ thị con riêng biệt, hai thành phần liên thông không dược nối với nhau thông qua các cạnh.
- Trong trường hợp này, một truy vấn được xem là đúng đắn bằng cách chỉ giữ lại thành phần có liên quan đến quan hệ kết quả và loại bỏ các thành phần còn lại.

Ví dụ: **Truy vấn bị sai về ngữ nghĩa**
Xét truy vấn sau:

```
SELECT hoten, diem
FROM sinhvien, hoc, monhoc
WHERE sinhvien.masv=hoc.masv
        AND malop=’MT’
        AND diem > 5
        AND tenmh = ‘Tin hoc’;
```

Ví dụ: **Truy vấn bị sai về ngữ nghĩa**



Đồ thị truy vấn

Xử lý Truy vấn bị sai về ngữ nghĩa khi đồ thị truy vấn không liên thông. Có các giải pháp cho vấn đề này là:

(1) Hủy bỏ truy vấn này.

(2) Hủy bỏ các bảng không cần thiết trong mệnh đề From và các điều kiện có liên quan đến các bảng này trong mệnh đề WHERE

(3) Bổ sung điều kiện kết sao cho đồ thị truy vấn được liên thông. Một đồ thị truy vấn có thể không bị sai ngữ nghĩa nếu đồ thị này là một dồ thị đơn (có nhiều nhất một cạnh nối giữa hai đỉnh), liên thông và số cạnh bằng số đỉnh trừ 1.

4.3.1.2 Chuẩn hóa điều kiện của mệnh đề WHERE

Điều kiện ghi trong mệnh đề WHERE là một biểu thức luận lý có thể bao gồm các phép toán luận lý (not, and, or) được viết dưới một dạng bất kỳ. Ký hiệu các phép toán luận lý: not (-), and (^), or (v).

Bước này nhằm mục đích chuẩn hóa điều kiện của mệnh đề Where về một trong hai dạng chuẩn:

Dạng chuẩn giao (conjunctive normal form)

$(P_{11} \text{ v } P_{12} \text{ v} \ldots \text{v } P_{1n}) \text{ ^} \ldots \text{^ } (P_{m1} \text{ v } P_{m2} \text{ v} \ldots \text{v } P_{mn})$

Dạng chuẩn hợp (disjunctive normal form)

$(P_{11} \text{ ^ } P_{12} \text{ ^} \ldots \text{^ } P_{1n}) \text{ v} \ldots \text{v } (P_{m1} \text{ ^ } P_{m2} \text{ ^} \ldots \text{^ } P_{mn})$

Trong đó $\mathbf{P_{ij}}$ là một **biến luận lý** (có giá trị là true hoặc false) hoặc là một **vị từ** đơn giản dạng a **R** b

Để **biến đổi** điều kiện của mệnh đề **WHERE** về **một trong hai** dạng **chuẩn giao** hoặc dạng **chuẩn hợp**, chúng ta **sử dụng** các phép biến đổi **tương đương** của các phép toán luận lý.

Ký hiệu ≡ là sự tương đương.

Các phép biến đổi **tương đương:**

(1) P1 ^ P2 ≡ P2 ^ P1

(2) P1 ∨ P2 ≡ P2 v P1

(3) P1 ^ (P2 ^ P3) ≡ (P1 ^ P2) ^ P3

(4) P1 v (P2 v P3) ≡ (P1 v P2) v P3

(5) P1 ^ (P2 v P3) ≡ (P1 ^ P2) v (P1 ^ P3)

(6) P1 v (P2 ^ P3) ≡ (P1 v P2) ^ (P1 v P3)
(7) ¬(P1 ^ P2) ≡ ¬ P1 v ¬ P2
(8) ¬(P1 v P2) ≡ ¬ P1 ^ ¬ P2
(9) ¬(¬P) ≡ P

Ví dụ: Xét truy vấn :
SELECT malop
FROM sinhvien
WHERE (NOT (malop='MT1')
AND (malop='MT1' OR malop='MT2')
AND NOT (malop='MT2')) OR hoten='Nam'
Điều kiện q của mệnh đề WHERE là:
(NOT (malop='MT1') AND (malop='MT1' OR malop='MT2')
AND NOT (malop='MT2')) OR hoten='Nam'

Đặt:
P1 là malop='MT1'
P2 là malop='MT2'
P3 là hoten='Nam'
Điều kiện q sẽ là: (¬P1 ^ (P1 v P2) ^ ¬ P2) v P3
Áp dụng các phép biến đổi (3), (5) để đưa điều kiện q về dạng chuẩn hợp:
(((¬P1 ^ P1) v (¬P1 ^ P2)) ^ ¬ P2) v P3
(¬P1 ^ P2 ^ ¬P2) v P3 = P3
4.3.1.3 Đơn giản hóa điều kiện của mệnh đề WHERE
Bước này sử dụng **các phép biến đổi tương đương** của các phép toán luận lý (not, and, or) để **rút gọn điều kiện** của mệnh đề **WHERE.**
Các phép biến đổi tương đương gồm có:
(10) P ^ P ≡ P
(11) P v P ≡ P
(12) P ^ true ≡ P
(13) P v false ≡ P
(14) P ^ false ≡ false
(15) P v true ≡ true
(16) P ^ ¬P ≡ false
(17) P v ¬P ≡ true
(18) P1 ^ (P1 v P2) ≡ P1
(19) P1 v (P1 ^ P2) ≡ P1

Ví dụ: Xét truy vấn :
SELECT malop
FROM sinhvien
WHERE (NOT (malop='MT1')
AND (malop='MT1' OR malop='MT2')
AND NOT (malop='MT2')) OR hoten='Nam'
Điều kiện q của mệnh đề WHERE là:
(NOT (malop='MT1') AND (malop='MT1' OR malop='MT2')
AND NOT (malop='MT2')) OR hoten='Nam'
Đặt: P1 là malop='MT1'; P2 là malop='MT2'; P3 là hoten='Nam'
Điều kiện q ở dạng chuẩn hợp là:
(¬P1 ^ P1^ P2) v (¬P1 ^ P2 ^ ¬ P2) v P3

Áp dụng phép biến đổi (16) ta được: **(false ^ ¬ P2)v (¬P1^ false) v P3**
Áp dụng phép biến đổi (14) ta được: **false v false v P3**
Áp dụng phép biến đổi (15) ta được **q cuối cùng là P3** tức là hoten='Nam'.

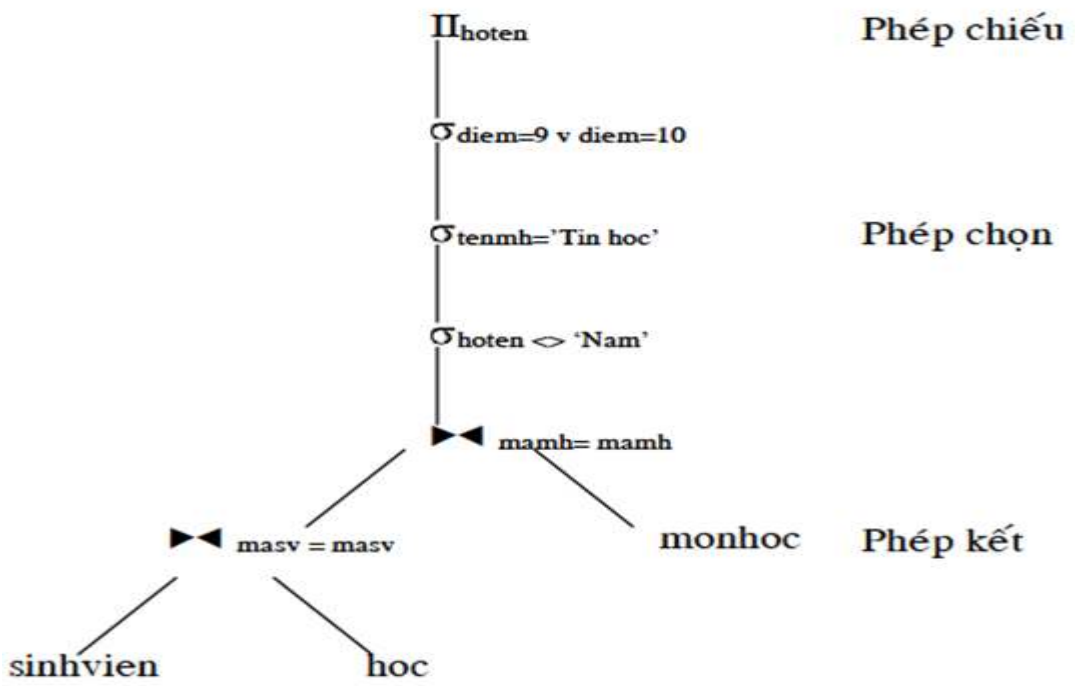
```
SELECT malop
FROM sinhvien
WHERE (NOT (malop='MT1')
        AND (malop='MT1' OR malop='MT2')
        AND NOT (malop='MT2')) OR hoten='Nam'
```

```
SELECT malop
FROM sinhvien
WHERE hoten='Nam';
```

4.3.1.4 Biến đổi truy vấn thành biểu thức đại số quan hệ hiệu quả
Bước này sử dụng các phép biến đổi tương đương của các phép toán đại số quan hệ nhằm để loại bỏ các phép toán đại số quan hệ không cần thiết và giảm vùng nhớ trung gian được sử dụng trong quá trình thực hiện các phép toán đại số quan hệ cần thiết cho truy vấn.
Bước này bao gồm hai bước sau đây:
– Biến đổi truy vấn thành một biểu thức đại số quan hệ, biểu diễn biểu thức đại số quan hệ này bằng một cây toán tử.
– Đơn giản hóa cây toán tử để có được một biểu thức đại số quan hệ hiệu qua
– Biểu diễn truy vấn bằng cây toán tử gồm các bước:
(1) Các nút lá được tạo lập từ các quan hệ ghi trong mệnh đề From
(2) Nút gốc được tạo lập bằng phép chiếu trên các thuộc tính ghi trong mệnh đề SELECT.
(3) Điều kiện ghi trong mệnh đề WHERE được biến đổi thành một chuỗi thích hợp các phép toán đại số quan hệ (phép chọn, phép kết, phép hợp…) đi từ các nút lá đến nút gốc. Chuỗi các phép toán này có thể được cho trực tiếp bởi thứ tự của các vị từ đơn giản và các phép toán luận lý.
Ví dụ: Xét truy vấn cho biết họ tên của các sinh viên không phải là 'Nam' học môn học 'Tin hoc' đạt điểm 9 hoặc 10.

```
SELECT hoten
FROM sinhvien, hoc, monhoc
WHERE sinhvien.masv= hoc.masv
        AND hoc.mamh= monhoc.mamh
        AND hoten<> 'Nam'
        AND tenmh= 'Tin hoc'
        AND (diem= 9 OR diem = 10)
```

Biểu diễn truy vấn bằng cây toán tử.



Biểu thức đại số quan hệ tương ứng là:

Π_{hoten} ($\sigma_{(diem=9\ v\ diem=10)\ \wedge\ tenmh='Tin\ hoc'\wedge hoten<>'Nam'}$
((sinhvien $\bowtie_{masv=masv}$ hoc) $\bowtie_{mamh=mamh}$ monhoc))

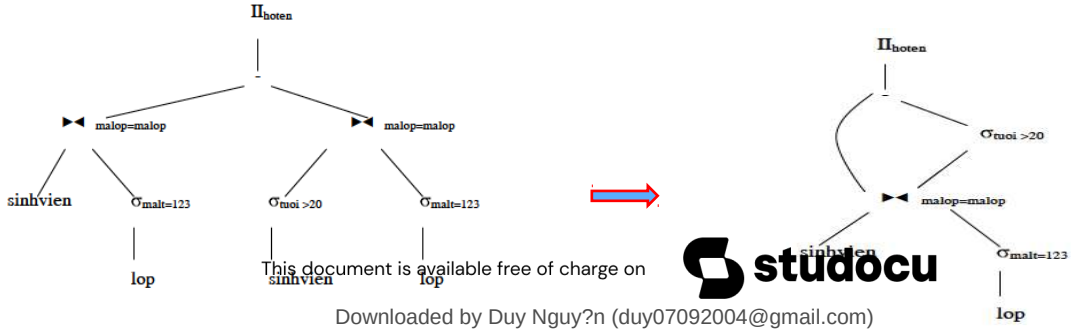
Đơn giản hoá cây toán tử nhằm mục đích để đạt hiệu quả (loại bỏ các phép toán dư thừa trên các quan hệ, giảm vùng nhớ trung gian, giảm thời gian xử lý truy vấn) bằng cách sử dụng các phép biến đổi tương đương của các phép toán đại số quan hệ.

Các phép biến đổi tương đương sau đây để đơn giản hóa một cây toán tử

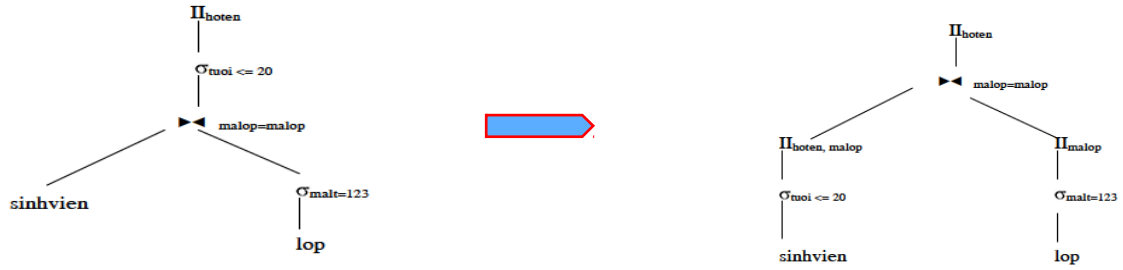
(1) $R \bowtie R \equiv R$
(2) $R \cup R \equiv R$
(3) $R - R \equiv \emptyset$
(4) $R \bowtie \sigma_F(R) \equiv \sigma_F R$
(5) $R \cup \sigma_F(R) \equiv R$
(6) $R - \sigma_F(R) \equiv \sigma_{\neg F}(R)$
(7) $\sigma_{F1}(R) \bowtie \sigma_{F2}(R) \equiv \sigma_{F1 \wedge F2}(R)$
(8) $\sigma_{F1}(R) \cup \sigma_{F2}(R) \equiv \sigma_{F1 \vee F2}(R)$
(9) $\sigma_{F1}(R) - \sigma_{F2}(R) \equiv \sigma_{F1 \wedge \neg F2}(R)$
(10) $R \cap R \equiv R$
(11) $R \cap \sigma_F(R) \equiv F\ R$
(12) $\sigma_{F1}(R) \cap \sigma_{F2}(R) \equiv \sigma_{F1 \wedge F2}(R)$
(13) $\sigma_F(R) - R \equiv \emptyset$

Xét truy vấn cho biết các họ tên của các sinh viên thuộc lớp có mã lớp trưởng là 123 và các sinh viên này có tuổi không lớn hơn 20 tuổi. Một biểu thức đại số quan hệ cho truy vấn này là:

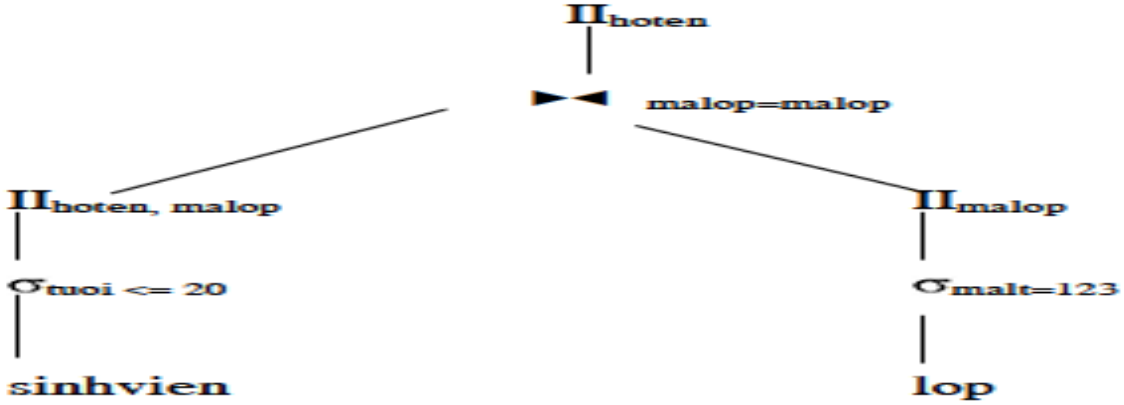
$\prod_{hoten}$ ((sinhvien$\bowtie_{malop=malop}$ $\sigma_{malt=123}$ (lop)) –
($\sigma_{tuoi>20}$(sinhvien) $\bowtie_{malop=malop}$ $\sigma_{malt=123}$ (lop))



Áp dụng phép biến đổi tương đương (6) với R là biểu thức:
sinhvien$\bowtie_{malop=malop}$ $\sigma_{malt=123}$ (lop)



Và biểu thức đại số quan hệ sau khi đã đơn giản hoá là:
$\prod_{hoten}(\prod_{hoten,malop}(\sigma_{tuoi<=20}(sinhvien))\bowtie_{malop=malop}\prod_{malop}(\sigma_{malt=123}(lop))$



Các tiêu chuẩn khi đơn giản hóa một biểu thức đại số quan hệ
Tiêu chuẩn 1. Dùng tính idempotence (tương đương) của phép chọn và phép chiếu để tạo ra các phép chọn và phép chiếu thích hợp cho mỗi quan hệ toán hạng.
Tiêu chuẩn 2. Thực hiện các phép chọn và các phép chiếu càng sớm càng tốt, tức là đẩy các phép chọn và các phép chiếu xuống phía dưới cây càng xa càng tốt.
Tiêu chuẩn 3. Khi các phép chọn được thực hiện sau một phép tích thì kết hợp các phép toán này để tạo thành một phép kết.
Tiêu chuẩn 4. Kết hợp chuỗi các phép toán một ngôi liên tiếp nhau áp dụng cho một quan hệ toán hạng. Một chuỗi các phép chọn liên tiếp nhau (hoặc một chuỗi các phép liên kết liên tiếp nhau) có thể được kết hợp thành một phép chọn (hoặc một phép kết).
Tiêu chuẩn 5. Khi phát hiện các biểu thức con chung trong biểu thức truy vấn, áp dụng các phép biến đổi tương đương để đơn giản hoá biểu thức truy vấn.

4.3.2. Bước 2 – Định vị dữ liệu
Bước định vị dữ liệu (Data Localization) còn được gọi là bước tối ưu hóa truy vấn trên lược đồ phân mảnh nhằm:

- Loại bỏ các phép toán đại số quan hệ không cần thiết trên các mảnh;
- Giảm vùng nhớ trung gian.

Tối ưu hóa truy vấn trên lược đồ phân mảnh bao gồm 2 bước sau:

Bước 2.1. Biến đổi biểu thức đại số quan hệ trên lược đồ toàn cục

Bước 2.2. Đơn giản hoá biểu thức đại số quan hệ trên lược đồ phân mảnh

4.3.2.1. Biến đổi biểu thức đại số quan hệ trên lược đồ toàn cục

Bước này sẽ **biến đổi biểu thức đại số quan hệ** trên **lược đồ toàn cục** (chứa các quan hệ toàn cuc) **thành biểu thức đại số quan hệ trên lược đồ phân mảnh** (chứa các mảnh của quan hệ toàn cục) **bằng** cách **thay thế mỗi quan hệ toàn cục trong cây toán tử** bởi biểu thức tái lập của nó.

Biểu thức tái lập của một **quan hệ toàn cục** là **một biểu thức đại số quan hệ** bao gồm **các mảnh** của quan hệ này mà biểu thức này cho **phép tạo lại quan hệ toàn cục** này. **Biểu thức tái lập** cũng **được biểu diễn** bằng **một cây toán tử**.

4.3.2.2. Đơn giản hoá biểu thức đại số quan hệ trên lược đồ phân mảnh

Đơn giản hoá biểu thức đại số quan hệ trên lược đồ phân mảnh để có được một biểu thức hiệu quả (loại **bỏ** các **phép toán không cần thiết**, **giảm vùng nhớ trung gian**) bằng cách **sử dụng** các **phép biến đổi tương đương của đại số quan hệ** và của **đại số quan hệ được tuyển chọn**.

Các phép biến đổi tương đương (áp dụng cho các quan hệ và các quan hệ được tuyển chọn) gồm có:

(1) $\sigma_F(\varnothing) \equiv \varnothing$

(2) $\prod x (\varnothing) \equiv \varnothing$

(3) $R \times \varnothing \equiv \varnothing$

(4) $R \cup \varnothing \equiv R$

(5) $R \cap \varnothing \equiv \varnothing$

(6) $R - \varnothing \equiv R$

(7) $\varnothing - R \equiv \varnothing$

(8) $R \bowtie \varnothing \equiv \varnothing$

(9) $R \ltimes \varnothing \equiv \varnothing$

(10) $\varnothing \ltimes R \equiv \varnothing$

Đơn giản hoá một biểu thức đại số quan hệ trên lược đồ phân mảnh được thực hiện dựa trên các tiêu chuẩn sau:

Tiêu chuẩn 6: Di chuyển các phép chọn xuống các nút lá của cây, và sau đó áp dụng chúng bằng cách dùng đại số quan hệ được tuyển chọn; thay thế các kết quả chọn lựa bởi quan hệ rỗng nếu điều kiện chọn của kết quả bị mâu thuẫn.

Tiêu chuẩn 7: Để phân phối các phép kết xuất hiện trong một truy vấn toàn cục, các phép hợp (biểu diễn tập hợp của các phân mảnh) phải được di chuyển lên phía trên các phép kết mà chúng ta muốn phân phối để loại bỏ các phép kết không cần thiết.

Tiêu chuẩn 8: Dùng đại số quan hệ được tuyển chọn để định trị điều kiện chọn của các toán hạng của các phép kết; thay thế cây con, bao gồm phép kết và các toán hạng của nó, bằng quan hệ rỗng nếu điều kiện chọn của kết quả của phép kêt bị mâu thuẫn.

4.3.3. Bước 3 – Tối ưu hóa truy vấn toàn cục

Bước tối ưu hoá truy vấn toàn cục nhằm để tìm ra một chiến lược thực hiện truy vấn sao cho chiến lược này gần tối ưu (theo nghĩa giảm thời gian thực hiện truy vấn trên dữ liệu được phân tán, giảm vùng nhớ trung gian).

Một chiến lược được đặc trưng bởi thứ tự thực hiện các phép toán đại số quan hệ và các tác vụ truyền thông cơ bản (gởi/nhận) dùng để truyền dữ liệu giữa các vị trí.

Bằng các hoán đổi thứ tự của các phép toán trong biểu thức truy vấn phân mảnh, ta có thể có được nhiều truy vấn tương đương.

Tối ưu hóa truy vấn toàn cục là tìm ra một thứ tự thực hiện các phép toán trong biểu thưc truy vấn sao cho ít tốn thời gian nhất. Đặc biệt khâu tốn kém thời gian trong cơ sở dữ liệu phân tán là khâu truyền dữ liệu do tốc độ và băng thông giới hạn.
Trong trường hợp nhân bản thì còn phải tính xem nhân bản nào được sử dụng nhằm giảm chi phí truyền thông.
Một khía cạnh quan trọng của tối ưu hoá truy vấn là thứ tự thực hiện các phép kết phân tán. Nhờ tính giao hoán của các phép kết, chúng ta có thể làm giảm chi phí thực hiện các phép kết này. Một kỹ thuật cơ bản để tối ưu hoá một chuỗi các phép kết phân tán là sử dụng phép nửa kết nhằm làm giảm chi phí truyền thông giữa các vị trí và tăng tính xử lý cục bộ tại các vị trí.
Ví dụ: Giả sử ta có sự phân tán dữ liệu sau:

Mảnh **sinhvien1** đặt tại **vị trí 1** và
Mảnh **lop1** đặt tại **vị trí 2**

Chúng ta cần thực hiện phép kết phân tán sau: **Sinhvien1 ⋈ lop1**
Bằng cách áp dụng phép nửa kết biểu thức trên tương đương với:
Lop1 ⋈ (sinhvien1 ⋉ ∏ malop(lop1))
Ví dụ: Bằng cách áp dụng phép nửa kết biểu thức trên tương đương với: **Lop1 ⋈ (sinhvien1 ⋉ ∏ malop(lop1))**
Do đó ta có một chiến lược thực hiện cho phép kết phân tán này với các tác vụ truyền thông sau:
1) Thực hiện $T_1 = \prod_{malop}(lop1)$ cục bộ tại vị trí 2.
2) Truyền T_1 từ vị trí 2 qua vị trí 1.
3) Thực hiện T_2 = sinhvien1 ⋉ T_1 cục bộ tại vị trí 1.
4) Truyền T_2 từ vị trí 1 qua vị trí 2.
5) Thực hiện T_3 = lop1 ⋉ T_2 cục bộ tại vị trí 2.
6) Truyền T_3 từ vị trí 2 qua vị trí của ứng dụng cần thực hiện của phép kết này
4.3.4. Bước 4 – Tối ưu hóa truy vấn cục bộ
Tối ưu hoá truy vấn cục bộ nhằm để thực hiện các truy vấn con được phân tán tại mỗi vị trí, gọi là truy vấn cục bộ có chứa các mảnh, sau đó được tối ưu hoá trên lược đồ cục bộ tại mỗi vị trí. Tối ưu hoá truy vấn cục bộ sử dụng các thuật toán tối ưu hoá truy vấn của cơ sở dữ liệu tập trung.
Mô hình chi phí phân tán
Tổng thời gian (hay Tổng chi phí): Tổng các thành phần chi phí
Thời gian đáp ứng: Thời gian tính từ khi khởi hoạt cho đến khi hoàn thành câu truy vấn
Mạng diện rộng WAN
- Khởi tạo và truyền thông điệp với chi phí cao
- Xử lý cục bộ có chi phí thấp
- Tỷ lệ truyền và thời gian xuất nhập có chi phí = 20:1

Mạng cục bộ LAN
- Phải xét cả chi phí cục bộ lẫn chi phí truyền
- Tỷ lệ = 1:1.6

CÁC THUẬT TOÁN TỐI ƯU TRUY VẤN PHÂN TÁN
Ba thuật toán cơ bản đại diện cho nhiều lớp thuật toán khác nhau là: Ingres phân tán

- System R*
- SDD-1
- IngresIngres dùng thời điểm tối ưu hoá động còn các hệ kia dùng thời điểm tĩnh.
- Hàm chi phí của SDD-1 và R* là hạ thấp tối đa tổng thời gian, còn Ingres phân tán là giảm tổ hợp thời gian đáp ứng và tổng thời gian.
- Cấu hình mạng được giả thiết là mạng diện rộng điểm - điểm theo SDD-1.
- Các thuật toán Ingres và R* có thể thực thi trên cả mạng LAN và WAN.
- SDD-1 sử dụng nối nửa như một kỹ thuật tối ưu hoá.

- Ingres có thể xử lý các mảnh.
- phân tán; System R*; SDD-1

SO SÁNH CÁC THUẬT TOÁN TỐI ƯU HOÁ

Thuật toán	Thời điểm tối ưu	Hàm mục tiêu	Hệ số tối ưu hoá	Topo mạng	Nối nửa	Số liệu thống kê	Phân mảnh
Dist. INGRES	Động	Thời gian đáp ứng hoặc tổng chi phí	Kích thước TB, chi phí xử lý	Tổng quát hoặc phát tán	Không	1	Ngang
R*	Tĩnh	Tổng chi phí	Lượng TB, kích thước TB, IO, CPU	Tổng quát hoặc cục bộ	Không	1, 2	Không
SDD-1	Tĩnh	Tổng chi phí	Kích thước TB	Tổng quát	Có	1, 3, 4, 5	Không

1. Lực lượng của quan hệ; 2. Số giá trị duy nhất của mỗi thuộc tính; 3. Hệ số tuyển chọn nối; 4. Kích thước của nối trong mỗi thuộc tính nối; 5. Kích thước thuộc tính và kích thước bộ

KẾT LUẬN

- Mục tiêu của việc xử lý truy vấn phân tán là hạ thấp tối đa hàm chi phí
- Nguyên liệu quan trọng của bài toán tối ưu truy vấn là số liệu thống kê CSDL và các công thức dùng để đánh giá kích thước các kết quả trung gian.
- Phép toán quan trọng nhất trong xử lý truy vấn phân tán là phép toán nối.
- Việc sử dụng thuật toán nào là còn tuỳ theo từng điều kiện cụ thể:
  - Với mạng diện rộng WAN, nên sử dụng thuật toán SDD-1.
  - Với mạng cục bộ LAN, có thể dùng thuật toán D-Ingres hoặc R* do không sử dụng các nối nửa, trong đó
    - D-Ingres tối ưu động thích hợp với phân mảnh ngang
    - R* tối ưu tĩnh thích hợp với các truy vấn được dùng thường xuyên.