

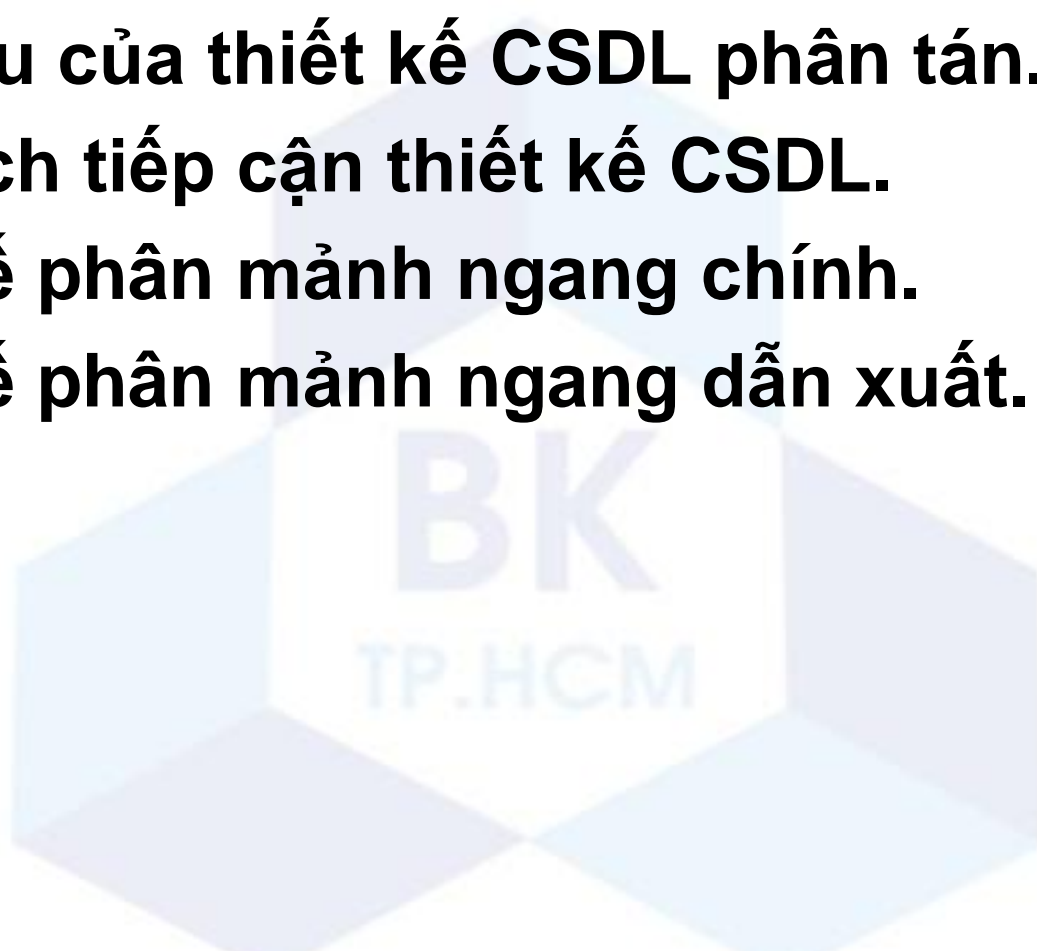
Chương 5

Thiết kế CSDL phân tán



Nội dung

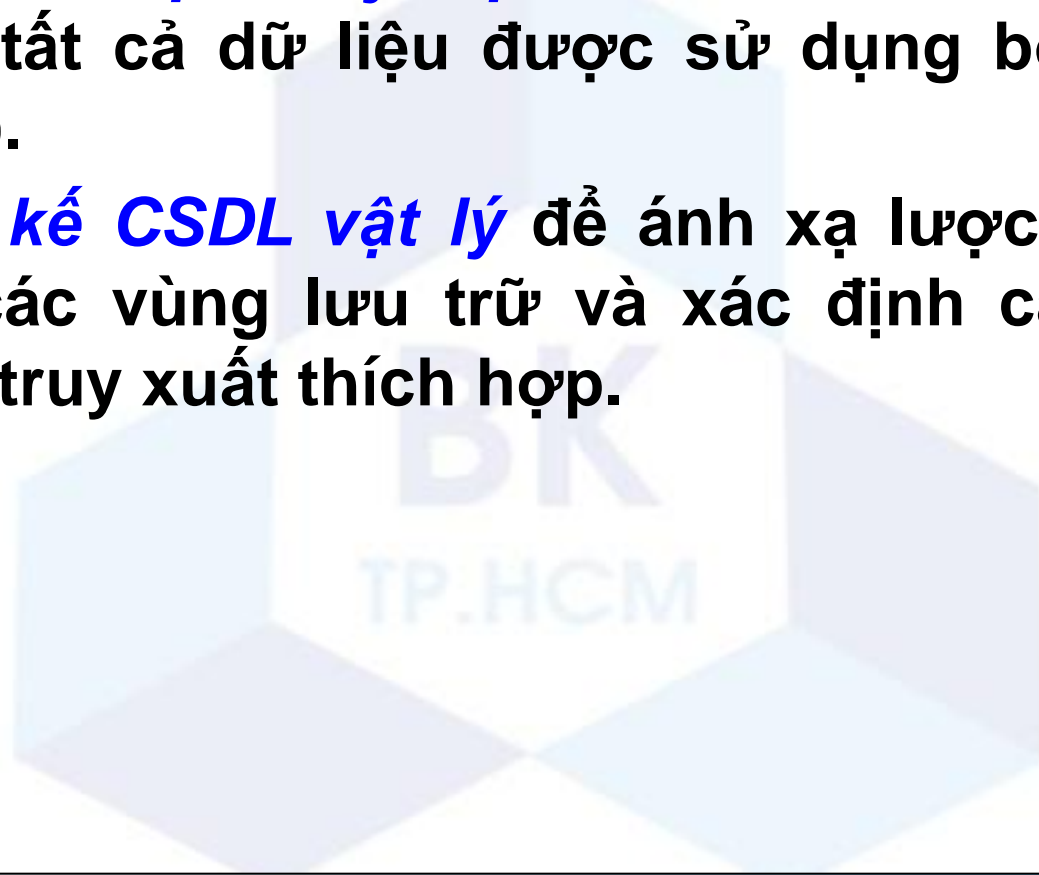
- ❖ Các bước thiết kế CSDL.
- ❖ Mục tiêu của thiết kế CSDL phân tán.
- ❖ Các cách tiếp cận thiết kế CSDL.
- ❖ Thiết kế phân mảnh ngang chính.
- ❖ Thiết kế phân mảnh ngang dẫn xuất.



Các bước thiết kế cơ sở dữ liệu

❖ Thiết kế CSDL tập trung

- ▶ **Thiết kế lược đồ ý niệm** để mô tả CSDL được tích hợp (tất cả dữ liệu được sử dụng bởi các ứng dụng).
- ▶ **Thiết kế CSDL vật lý** để ánh xạ lược đồ ý niệm vào các vùng lưu trữ và xác định các phương pháp truy xuất thích hợp.



Các bước thiết kế cơ sở dữ liệu

❖ Thiết kế CSDL phân tán

- ▶ **Thiết kế lược đồ toàn cục** giống với thiết kế lược đồ ý niệm của CSDL tập trung.
- ▶ **Thiết kế phân mảnh** để xác định các quan hệ toàn cục được phân chia thành các mảnh ngang, mảnh dọc hoặc phân mảnh hỗn hợp như thế nào.
- ▶ **Thiết kế định vị mảnh** để xác định các mảnh được ánh xạ vào các hình ảnh vật lý như thế nào và xác định việc nhân bản các mảnh.
- ▶ **Thiết kế CSDL vật lý cục bộ** giống với thiết kế CSDL vật lý của CSDL tập trung.

Các bước thiết kế cơ sở dữ liệu

- ❖ Thiết kế các chương trình ứng dụng được thực hiện sau thiết kế các lược đồ, nhưng các hiểu biết về các yêu cầu của ứng dụng ảnh hưởng đến thiết kế các lược đồ để hỗ trợ ứng dụng một cách hiệu quả.
- ❖ Trong thiết kế CSDL phân tán, cần phải hiểu biết thật chính xác về các yêu cầu của ứng dụng, nhất là đối với các ứng dụng **quan trọng hơn** (được thực hiện thường xuyên hoặc được chạy hiệu quả).

Các bước thiết kế cơ sở dữ liệu

- ❖ Trong các yêu cầu của ứng dụng, cần quan tâm đến:
 - ▶ **Nơi chạy ứng dụng**, còn được gọi là **nơi gốc** của ứng dụng.
 - ▶ **Tần suất chạy ứng dụng** (số lần chạy trong một đơn vị thời gian) tại mỗi nơi.
 - ▶ **Số lượng, loại và sự phân tán của các truy xuất** trong mỗi ứng dụng đến mỗi đối tượng dữ liệu cần thiết.

Mục tiêu của thiết kế phân tán dữ liệu

❖ Tính cục bộ xử lý (*processing locality*)

- ▶ Phân tán dữ liệu để làm cực đại hóa tính cục bộ xử lý là đặt dữ liệu càng gần các ứng dụng sử dụng các dữ liệu này càng tốt. Đây là một trong các mục tiêu chủ yếu của CSDL phân tán.
- ▶ Một quan hệ không là một *đơn vị phân tán* thích hợp vì:
 1. Khung nhìn thường là tập con của các quan hệ. Tính cục bộ xử lý của ứng dụng không được xác định trên các quan hệ nhưng trên các tập con của các quan hệ này. Các tập con của các quan hệ là các đơn vị phân tán.

Mục tiêu của thiết kế phân tán dữ liệu

❖ Tính cục bộ xử lý

2. Nếu các ứng dụng có các khung nhìn được định nghĩa trên một quan hệ được đặt tại các nơi, thì có hai cách để xem quan hệ là một đơn vị phân tán:
 - Quan hệ không được nhân bản và chỉ được lưu trữ tại một nơi: một số lượng lớn không cần thiết các truy xuất dữ liệu từ xa.
 - Quan hệ được nhân bản tại tất cả hoặc một số nơi có chạy các ứng dụng: có sự nhân bản không cần thiết và gây ra các vấn đề không mong muốn trong việc cập nhật.

Mục tiêu của thiết kế phân tán dữ liệu

❖ Tính cục bộ xử lý

- ▶ Tính cục bộ xử lý dựa vào các *tham chiếu cục bộ* và các *tham chiếu từ xa*. Các tham chiếu này chỉ phụ thuộc vào sự phân tán dữ liệu.
- ▶ *Tính cục bộ hoàn toàn (complete locality)*: các ứng dụng có thể được thực hiện hoàn toàn tại nơi gốc của chúng.
 - Giảm bớt các truy xuất từ xa.
 - Dễ kiểm soát việc thực hiện ứng dụng.

Mục tiêu của thiết kế phân tán dữ liệu

❖ Tính sẵn sàng và độ tin cậy của dữ liệu

- ▶ **Tính sẵn sàng** (*availability*) và **độ tin cậy** (*reliability*) là ưu điểm của các hệ thống phân tán.
- ▶ Mức độ sẵn sàng cao đối với các ứng dụng chỉ đọc khi lưu trữ nhiều bản nhân của cùng một thông tin.
- ▶ Độ tin cậy đạt được bằng cách lưu trữ nhiều bản nhân của cùng một thông tin.
 - Có thể phục hồi khi máy tính bị ngừng hoặc có hư hỏng vật lý của một trong các bản nhân bằng cách sử dụng các bản nhân khác.

Mục tiêu của thiết kế phân tán dữ liệu

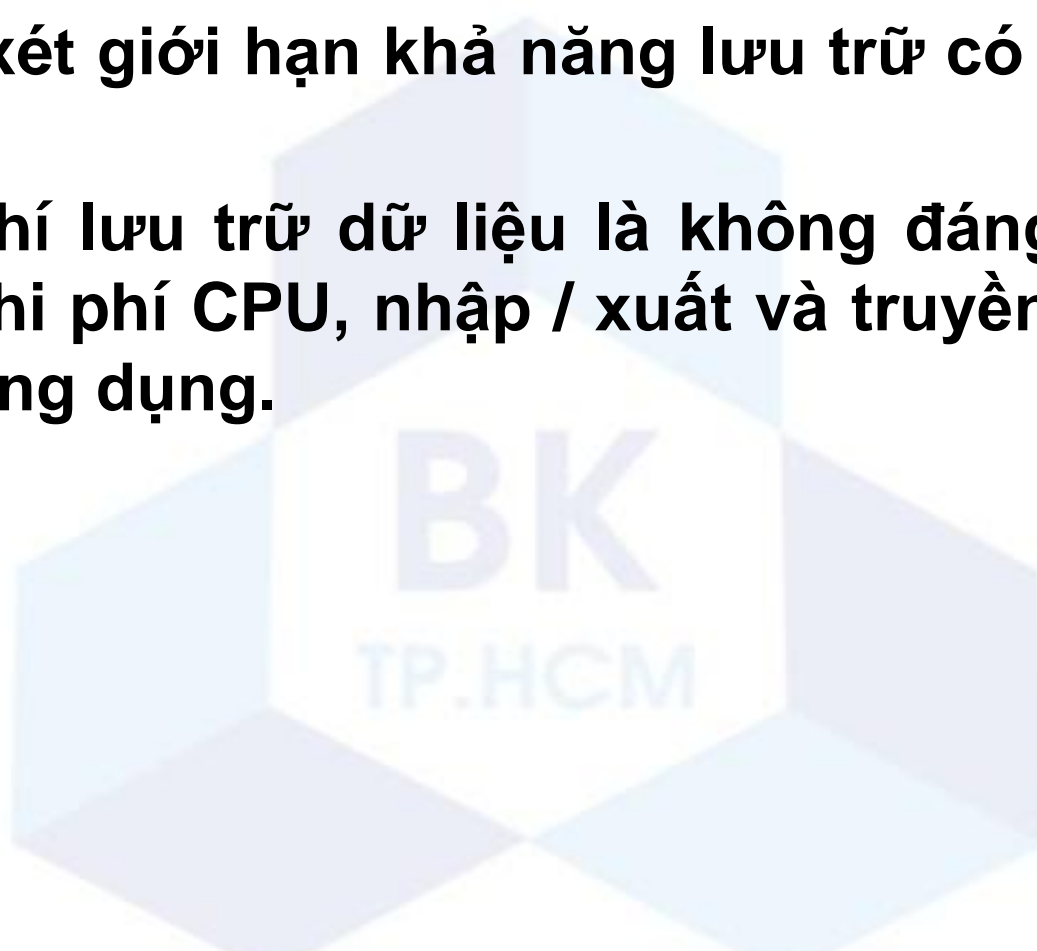
❖ Điều phối tải làm việc

- ▶ Điều phối tải làm việc để tận dụng ưu điểm của các nguồn lực khác nhau hoặc tính năng của các máy tính tại mỗi nơi và cực đại hóa mức độ thực hiện song song các ứng dụng.
- ▶ Điều phối tải làm việc có thể ảnh hưởng ngược lại với tính cục bộ xử lý.
 - Cần phải cân nhắc giữa tính cục bộ xử lý và điều phối tải làm việc trong thiết kế phân tán dữ liệu.
- ▶ Phân mảnh các quan hệ có thể thực hiện đồng thời một truy vấn bằng cách chia truy vấn này thành các truy vấn con để thực hiện trên các mảnh (*tính đồng thời nội truy vấn*).
- ▶ Sự phân mảnh làm tăng mức độ đồng thời và thông lượng của hệ thống.

Mục tiêu của thiết kế phân tán dữ liệu

❖ Chi phí lưu trữ và khả năng lưu trữ có sẵn

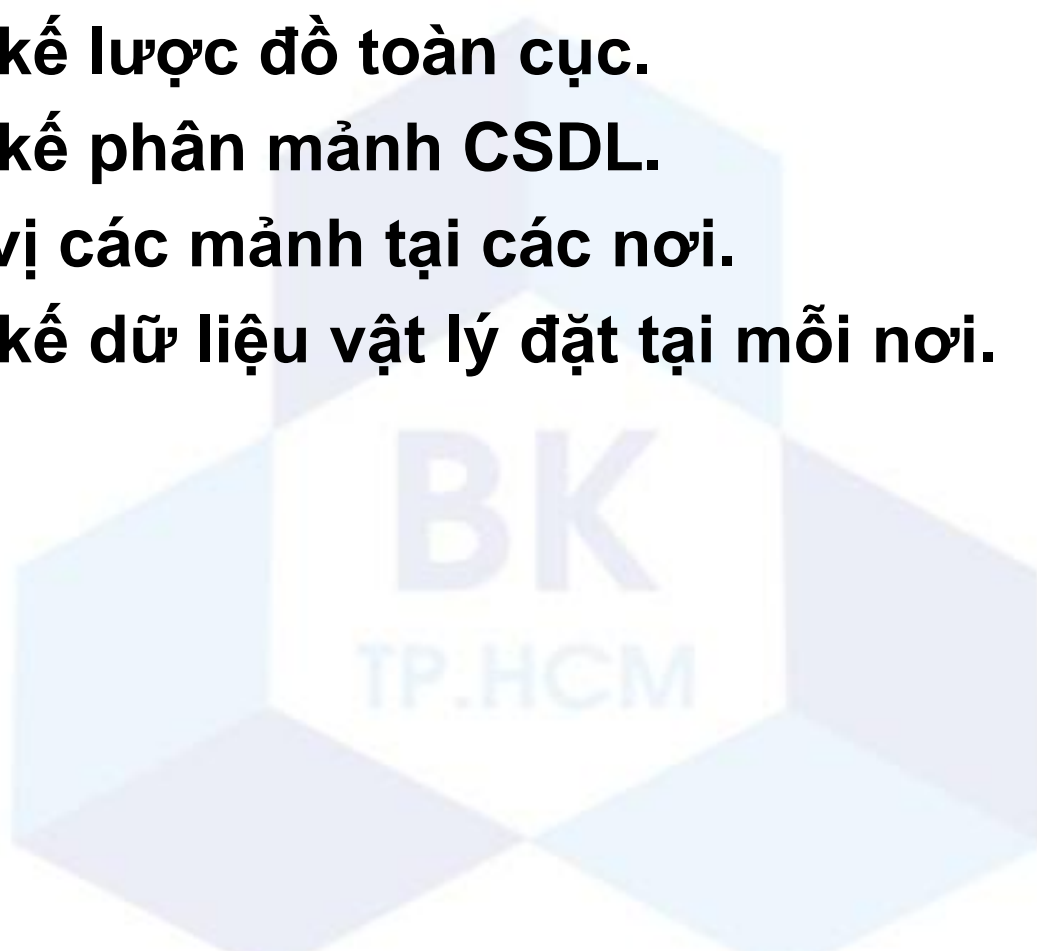
- ▶ Phải xét giới hạn khả năng lưu trữ có sẵn tại mỗi nơi.
- ▶ Chi phí lưu trữ dữ liệu là không đáng kể so với các chi phí CPU, nhập / xuất và truyền thông của các ứng dụng.

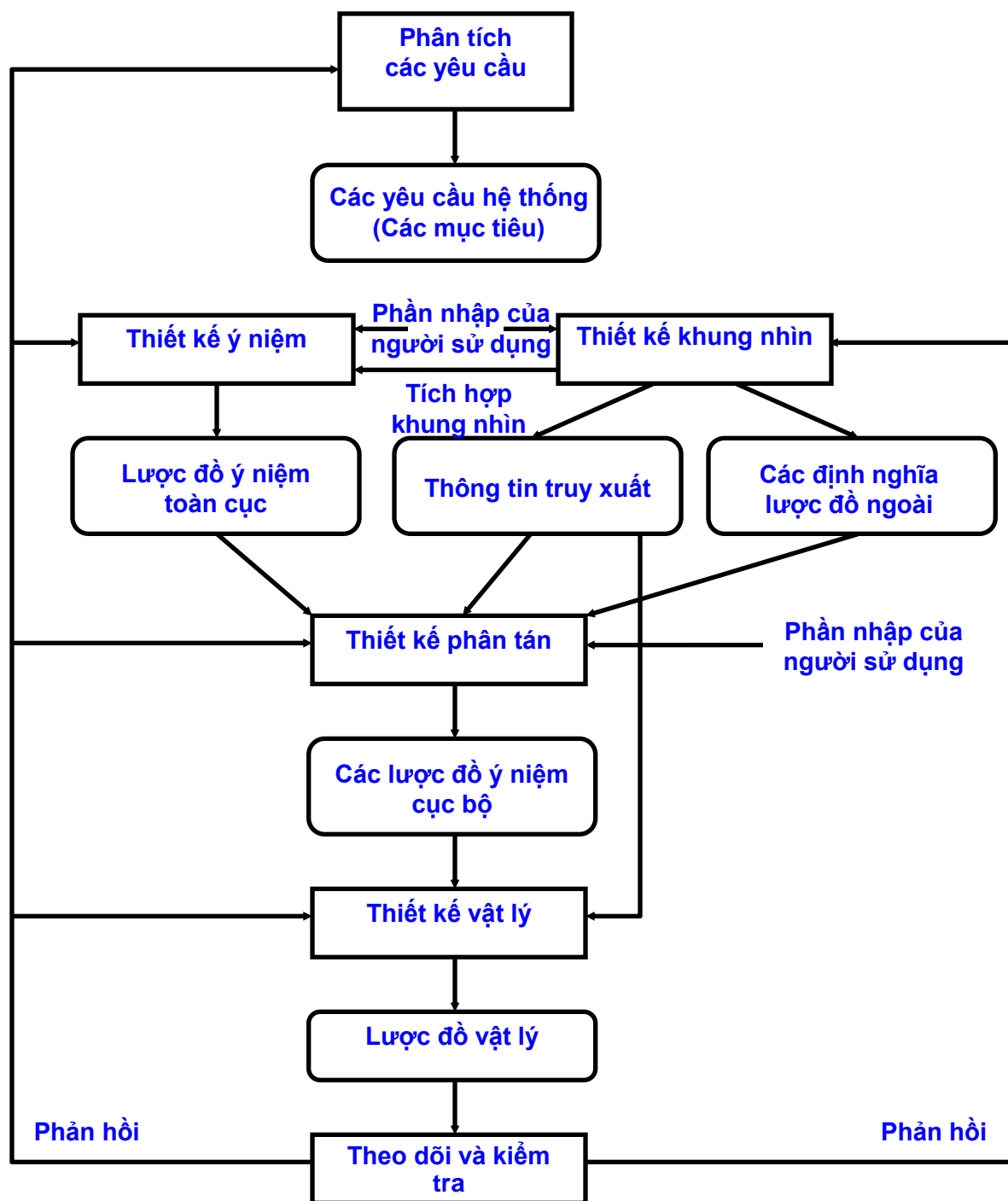


Cách tiếp cận từ trên xuống

❖ 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.





Hình 5.1. Sơ đồ thiết kế CSDL.

Cách tiếp cận từ dưới lên

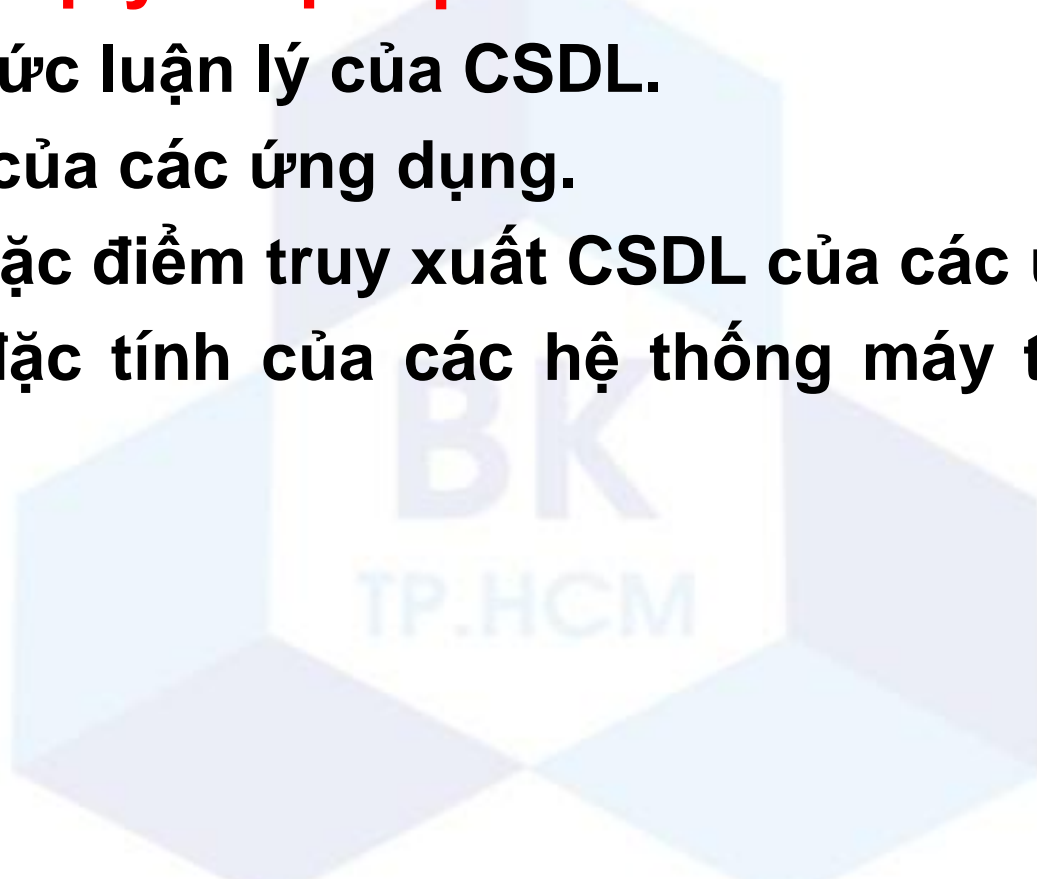
❖ Thiết kế từ dưới lên (*bottom-up design*)

- ▶ 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.

Các yêu cầu thông tin

❖ Các yếu tố trong thiết kế tối ưu ảnh hưởng đến các quyết định phân tán.

- ▶ Tổ chức luận lý của CSDL.
- ▶ Vị trí của các ứng dụng.
- ▶ Các đặc điểm truy xuất CSDL của các ứng dụng.
- ▶ Các đặc tính của các hệ thống máy tính tại mỗi nơi.



Các yêu cầu thông tin

❖ Các loại thông tin để thiết kế phân tán

- ▶ Thông tin về CSDL
- ▶ Thông tin về ứng dụng
- ▶ Thông tin về mạng truyền thông
- ▶ Thông tin về hệ thống máy tính



Thiết kế phân mảnh dữ liệu

❖ Mục đích của thiết kế phân mảnh

- ▶ Xác định các mảnh không giao nhau – là các *đơn vị phân tán* hoặc *đơn vị định vị luận lý* (*logical unit of allocation*).

❖ Thiết kế các mảnh

- ▶ Gom nhóm các bộ (phân mảnh ngang) hoặc các thuộc tính (phân mảnh dọc) có *cùng các đặc tính* theo quan điểm định vị của chúng.
- ▶ Mỗi nhóm tạo thành một mảnh.

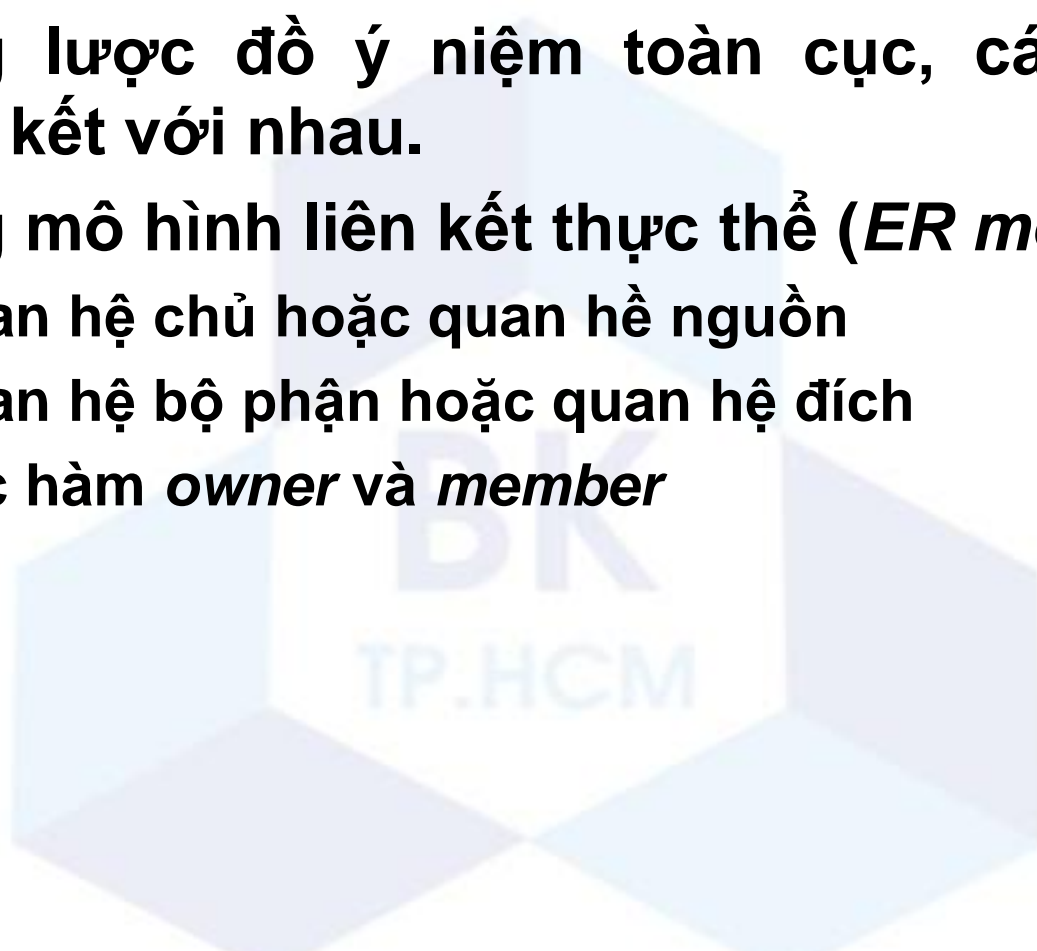
Thiết kế phân mảnh ngang

- ❖ Mỗi mảnh là một tập hợp con gồm các bộ của quan hệ.
- ❖ Phân mảnh ngang chính là phân chia một quan hệ dựa vào các vị từ định tính được định nghĩa trên quan hệ này.
- ❖ Phân mảnh ngang dẫn xuất là phân chia một quan hệ dựa vào các vị từ định tính được định nghĩa trên một quan hệ khác.

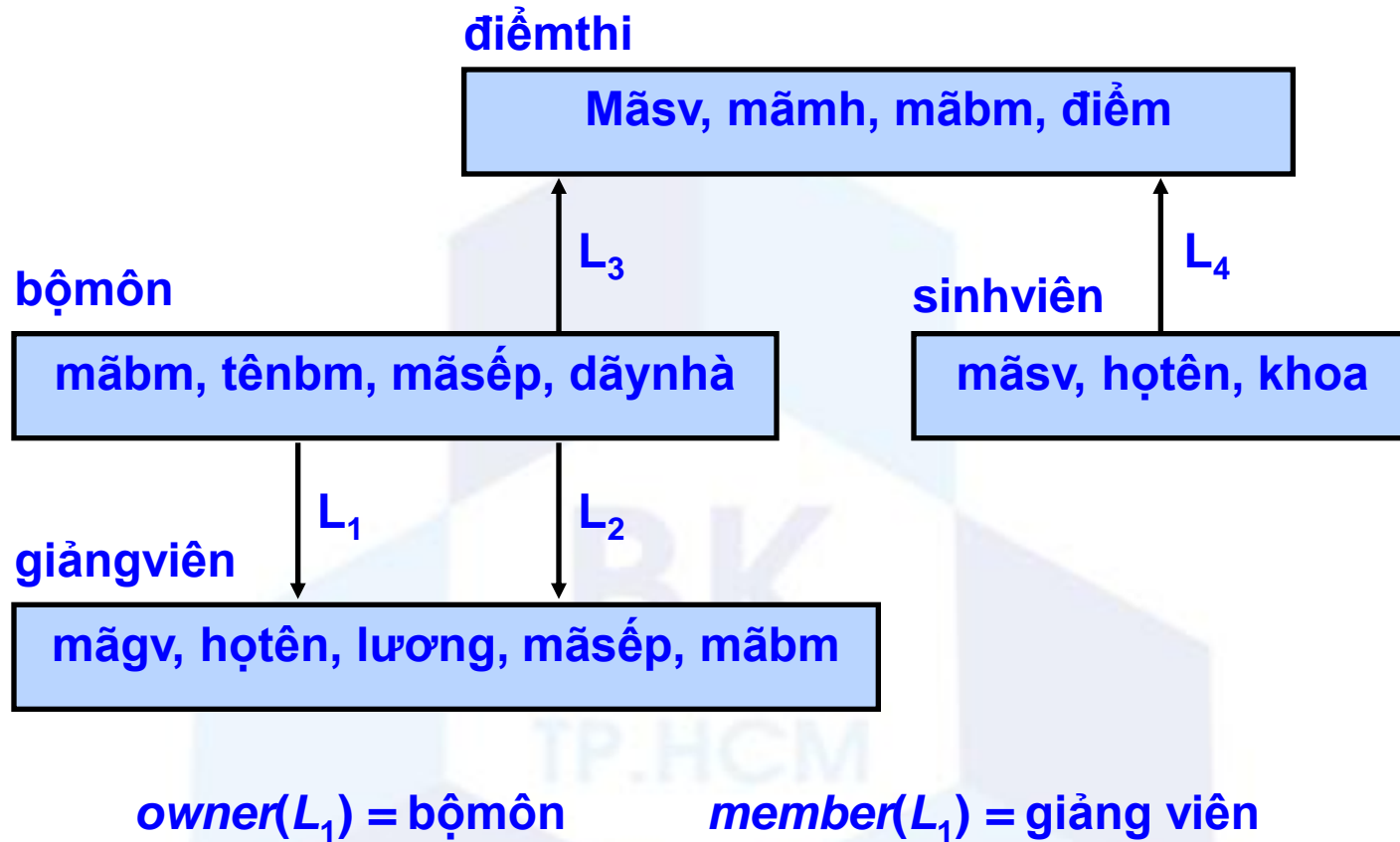
Thiết kế phân mảnh ngang

❖ Thông tin về CSDL

- ▶ Trong lược đồ ý niệm toàn cục, các quan hệ được kết với nhau.
- ▶ Trong mô hình liên kết thực thể (*ER model*):
 - Quan hệ chủ hoặc quan hệ nguồn
 - Quan hệ bộ phận hoặc quan hệ đích
 - Các hàm *owner* và *member*



Thiết kế phân mảnh ngang

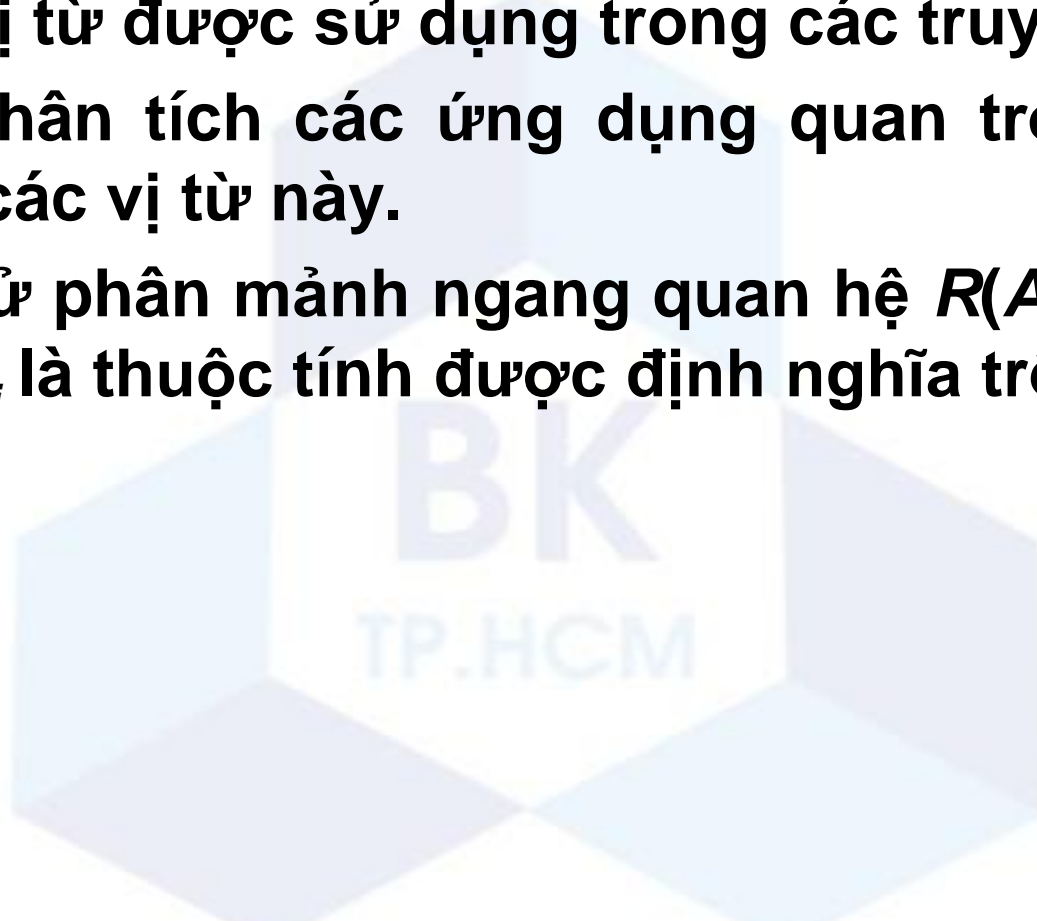


Hình 5.2. Biểu diễn các mối liên kết giữa các quan hệ dùng các đường liên kết .

Thiết kế phân mảnh ngang

❖ Thông tin về ứng dụng

- ▶ Các vị từ được sử dụng trong các truy vấn.
- ▶ Chỉ phân tích các ứng dụng quan trọng để xác định các vị từ này.
- ▶ Giả sử phân mảnh ngang quan hệ $R(A_1, A_2, \dots, A_n)$, với A_i là thuộc tính được định nghĩa trên miền D_i .



Thiết kế phân mảnh ngang

❖ Thông tin về ứng dụng

- ▶ **Vị từ đơn giản** (*simple predicate*) p_j được định nghĩa trên R có dạng:

$A_i \theta value$

θ là một trong các phép so sánh $=, \neq, <, \leq, >, \geq$

$value$ được chọn từ miền trị của A_i ($value \in D_i$)

- ▶ Ký hiệu P_r là tập các vị từ đơn giản được định nghĩa trên quan hệ R . Các phần tử của P_r được ký hiệu là p_j .

Thiết kế phân mảnh ngang

❖ Thông tin về ứng dụng

- ▶ **Vị từ giao tối thiểu** (*minterm predicate*) m_j đối với tập các vị từ đơn giản $P_r = \{p_1, p_2, \dots, p_m\}$ là một tổ hợp giao của tất cả các vị từ xuất hiện trong P_r (ở dạng thông thường hoặc ở dạng phủ định) sao cho m_j không bị mâu thuẫn.

$$m_j = \bigwedge p_i^*, 1 \leq i \leq m$$

với $p_i^* = p_i$ hoặc $p_i^* = \neg p_i$ và $m_j \neq false$

- ▶ Gọi tập các vị từ giao tối thiểu là:

$$M = \{m_1, m_2, \dots, m_z\}$$

Thiết kế phân mảnh ngang

❖ Thông tin về ứng dụng

- ▶ **Độ chọn giao tối thiểu** (*minterm selectivity*) là số bộ của quan hệ sẽ được truy xuất bởi một truy vấn tương ứng với một vị từ giao tối thiểu cho trước.
 - $sel(m_j)$ là độ chọn của vị từ giao tối thiểu m_j .
- ▶ **Tần số truy xuất** (*access frequency*) là tần số của các ứng dụng truy xuất dữ liệu.
 - Gọi $Q = \{q_1, q_2, \dots, q_q\}$ là tập các truy vấn.
 - $acc(q_i)$ là tần số truy xuất của truy vấn q_i trong một khoảng thời gian cho trước.
 - $acc(m_j)$ là tần số truy xuất của vị từ giao tối thiểu m_j .

Thiết kế phân mảnh ngang chính

- ❖ Mảnh ngang chính được xác định bằng phép chọn trên quan hệ toàn cục.

$$R_i = \sigma_{F_i}(R); \quad 1 \leq i \leq n$$

- ▶ F_i là điều kiện chọn của mảnh R_i
- ▶ Nếu F_i ở dạng chuẩn giao thì nó là một vị từ giao tối thiểu m_i
- ❖ Tính đúng đắn của phân mảnh ngang chính: mỗi bộ của quan hệ toàn cục được đưa vào trong một và chỉ một mảnh.

Thiết kế phân mảnh ngang chính

- ❖ Xác định phân mảnh ngang chính của một quan hệ toàn cục là xác định một tập các **vị từ chọn** (*selection predicate*) đầy đủ và tách biệt.
- ❖ Các bộ thuộc cùng một mảnh phải được tham chiếu giống nhau trong tất cả các ứng dụng.
- ❖ **Mảnh ngang** (*horizontal fragment*) hoặc **mảnh giao tối thiểu** (*minterm fragment*) R_i của quan hệ R bao gồm tất cả các bộ của R thỏa mãn vị từ giao tối thiểu m_i .

Thiết kế phân mảnh ngang chính

- ❖ Với tập các vị từ giao tối thiểu M , số lượng mảnh ngang bằng số lượng vị từ giao tối thiểu.
- ❖ **Các bước thiết kế phân mảnh ngang**
 - ▶ **Bước 1:** Tìm tập các vị từ chọn $P_{r'}$ là đầy đủ và tối thiểu.
 - ▶ **Bước 2:** Tìm tập các vị từ giao tối thiểu có thể được định nghĩa trên các vị từ của $P_{r'}$

Thiết kế phân mảnh ngang chính

- ❖ Một vị từ đơn giản p_i được gọi là **thích hợp** (*relevant*) đối với một tập P_r các vị từ đơn giản, nếu tồn tại ít nhất hai vị từ giao tối thiểu m_i và m_j của P_r mà các biểu thức của chúng chỉ khác nhau ở p_i (tức là m_i chứa p_i và m_j chứa $\neg p_i$) và tồn tại ít nhất một ứng dụng tham chiếu khác nhau đến hai mảnh f_i và f_j (tương ứng với m_i và m_j).

Thiết kế phân mảnh ngang chính

- ❖ Một tập các vị từ đơn giản P_r được gọi là **đầy đủ** (*complete*) nếu và chỉ nếu bất kỳ hai bộ nào thuộc bất kỳ mảnh giao tối thiểu nào được định nghĩa theo P_r thì bất kỳ ứng dụng nào đều tham chiếu đến hai bộ này với cùng một xác suất.
- ❖ Một tập các vị từ đơn giản P_r được gọi là **tối thiểu** (*minimal*) nếu tất cả các vị từ của nó là các vị từ thích hợp.
- ❖ Cho $P_r = \{p_1, p_2, \dots, p_m\}$ là một tập các vị từ đơn giản. Để cho P_r biểu diễn phân mảnh đúng đắn và hiệu quả thì P_r phải **đầy đủ** và **tối thiểu**.

Thiết kế phân mảnh ngang chính

❖ Ví dụ

- ▶ Một số ứng dụng quan trọng yêu cầu thông tin về các nhân viên là thành viên của các dự án.
- ▶ Một số ứng dụng quan trọng yêu cầu dữ liệu của các nhân viên là người lập trình (*job* = 'P'), có thể chạy tại bất kỳ nơi nào và tham chiếu đến tất cả người lập trình với cùng xác suất.
- ▶ Giả sử chỉ có hai phòng có *deptnum* là 1 và 2, do đó *deptnum* = 1 tương đương với *deptnum* # 2 và ngược lại.

Thiết kế phân mảnh ngang chính

❖ Ví dụ

- ▶ $P_1 = \{\text{deptnum} = 1\}$ là không đầy đủ vì các ứng dụng tham chiếu đến các bộ của những người lập trình ở trong mỗi mảnh được tạo ra bởi P_1 với xác suất lớn hơn.
- ▶ $P_2 = \{\text{deptnum} = 1; \text{job} = \text{'P'}\}$ là đầy đủ và tối thiểu.
- ▶ $P_3 = \{\text{deptnum} = 1; \text{job} = \text{'P'}, \text{sal} > 35000\}$ là đầy đủ nhưng không tối thiểu vì $\text{sal} > 35000$ không là vị từ thích hợp.

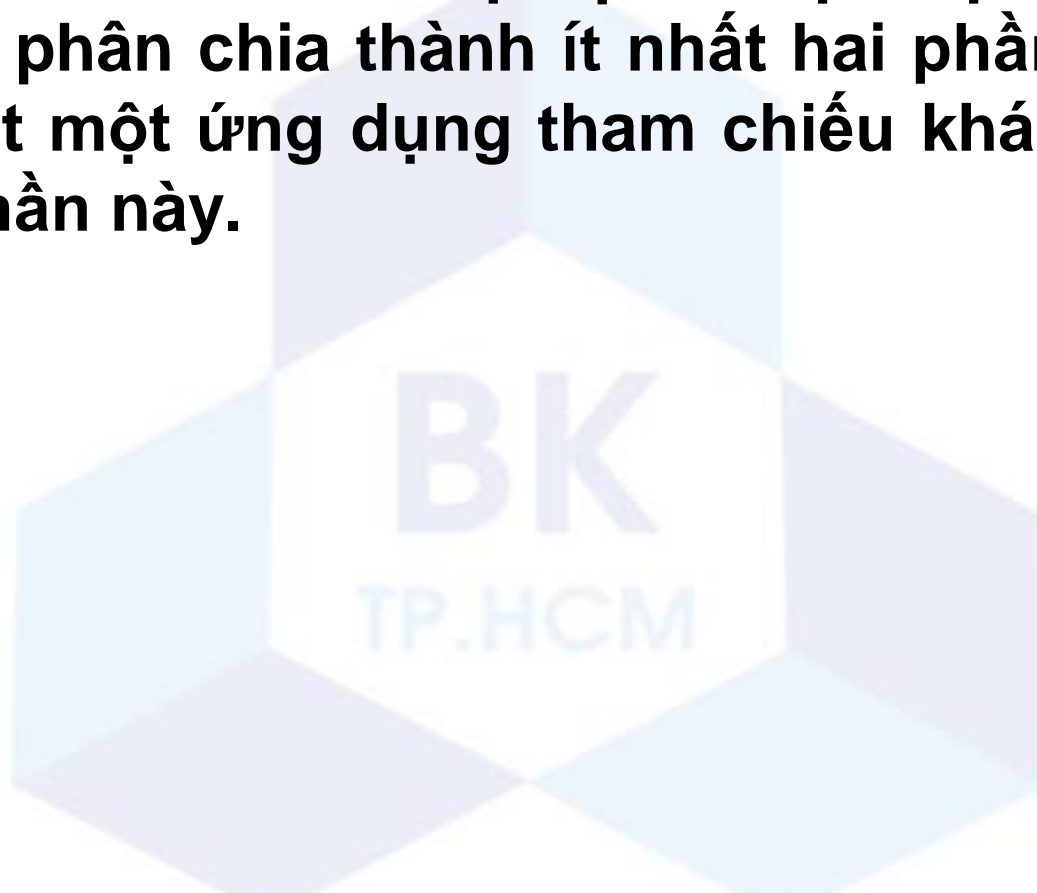
Thiết kế phân mảnh ngang chính

❖ **Giải thuật COM_MIN** tạo ra tập các vị từ $P_{r'}$ là đầy đủ và tối thiểu.

- ▶ **Bắt đầu:** Xét p_i chia các bộ của R thành hai phần và tồn tại ít nhất một ứng dụng tham chiếu khác nhau đến hai phần này. Cho $P_{r'} = p_i$.
- ▶ **Phương pháp:** Xét p_j chia ít nhất một mảnh f_k của $P_{r'}$ thành hai phần và tồn tại ít nhất một ứng dụng tham chiếu khác nhau đến hai phần này. Cho $P_{r'} \leftarrow P_{r'} \cup p_j$. Loại bỏ các vị từ không thích hợp ra khỏi $P_{r'}$. Lặp lại bước này cho đến khi tập các mảnh giao tối thiểu của $P_{r'}$ là đầy đủ.

Thiết kế phân mảnh ngang chính

- **Quy tắc 1:** là quy tắc cơ bản về tính đầy đủ và tính tối thiểu mà một quan hệ hoặc một mảnh được phân chia thành ít nhất hai phần và tồn tại ít nhất một ứng dụng tham chiếu khác nhau đến hai phần này.



Nhập: R : quan hệ; P_r : tập các vị từ đơn giản

Xuất: $P_{r'}$: tập các vị từ đơn giản là đầy đủ và tối thiểu
declare

F : tập các mảnh giao tối thiểu;

begin

Tìm $p_i \in P_r$ sao cho p_i phân chia R theo quy tắc 1;

$P_{r'} \leftarrow p_i$;

$P_r \leftarrow P_r - p_i$;

$F \leftarrow f_i$; $\{f_i \text{ là mảnh giao tối thiểu theo } p_i\}$

repeat

begin

$p_j \in P_r$ sao cho p_j chia mảnh f_k của $P_{r'}$ theo quy tắc 1;

$P_{r'} \leftarrow P_{r'} \cup p_j$;

$P_r \leftarrow P_r - p_j$;

$F \leftarrow F \cup f_j$;

while $\exists p_k \in P_{r'}$ là một vị từ không thích hợp do

begin

$P_{r'} \leftarrow P_{r'} - p_k$;

$F \leftarrow F - f_k$

end

end

until $P_{r'}$ là đầy đủ

end. {COM_MIN}

Thiết kế phân mảnh ngang chính

❖ Ví dụ

- ▶ Xét $sal > 35000$ là thích hợp:

$$P_1 = \{sal > 35000\}$$

- ▶ Xét $deptnum = 1$ là thích hợp và được đưa vào tập hợp trên tạo ra:

$$P_2 = \{sal > 35000, deptnum = 1\}$$

- ▶ Xét $job = 'P'$ là thích hợp và tập cuối cùng

$$P_3 = \{sal > 35000, deptnum = 1, job = 'P'\}$$

- ▶ Phát hiện $sal > 35000$ là không thích hợp đối với P_1 , tập cuối cùng

$$P_4 = \{deptnum = 1, job = 'P'\}$$

là đầy đủ và tối thiểu.

Thiết kế phân mảnh ngang chính

❖ Tìm tập các vị từ giao tối thiểu

- ▶ Xác định tập đầy đủ các vị từ có thể rất tốn kém vì tập này có thể rất lớn (lũy thừa của số lượng vị từ đơn giản).
- ▶ Tìm tập đầy đủ các vị từ phải được thực hiện **hợp lý** bằng cách:
 - Tập trung vào các ứng dụng khá quan trọng.
 - Không phân biệt các mảnh có các đặc điểm rất giống nhau.

Thiết kế phân mảnh ngang chính

❖ Tìm tập các vị từ giao tối thiểu

- ▶ Trong tập các vị từ đầy đủ, loại bỏ các vị từ vô nghĩa mà chúng mâu thuẫn với tập các phép suy diễn / để tìm ra các vị từ giao tối thiểu.

- ▶ Ví dụ:

Cho $P_{r'} = \{p_1, p_2\}$ với

$p_1: att = value_1$

$p_2: att = value_2$

và miền trị của att là $\{value_1, value_2\}$

Thiết kế phân mảnh ngang chính

❖ Tìm tập các vị từ giao tối thiểu

Tập I gồm hai phép suy diễn là:

$$i_1: (att = value_1) \Rightarrow \neg (att = value_2)$$

$$i_2: \neg (att = value_1) \Rightarrow (att = value_2)$$

Tập M gồm bốn vị từ được định nghĩa theo P_r :

$$m_1: (att = value_1) \wedge (att = value_2)$$

$$m_2: (att = value_1) \wedge \neg (att = value_2)$$

$$m_3: \neg (att = value_1) \wedge (att = value_2)$$

$$m_4: \neg (att = value_1) \wedge \neg (att = value_2)$$

m_1 và m_4 mâu thuẫn với tập I , loại chúng ra khỏi tập M .

m_2 và m_3 có thể được rút gọn:

$$m_2: att = value_1$$

$$m_3: att = value_2$$

Thiết kế phân mảnh ngang chính

❖ **Giải thuật PHORIZONTAL** để phân mảnh ngang chính một quan hệ.

Nhập: R_i : quan hệ; P_{r_i} : tập các vị từ đơn giản.

Xuất: M_i : tập các mảnh giao tối thiểu.

begin

$P_{r'_i} \leftarrow \text{COM_MIN}(R_i, R_{r_i});$

Xác định M_i là tập các vị từ giao tối thiểu;

Xác định I là tập các phép suy diễn giữa $p_i \in P_{r'_i}$;

for each $m_i \in M_i$ do

if m_i mâu thuẫn với I then

$M_i \leftarrow M_i - m_i$

end. {PHORIZONTAL}

Thiết kế phân mảnh ngang dẫn xuất

- ❖ Phân mảnh ngang dẫn xuất được định nghĩa trên các quan hệ bộ phận của đường liên kết theo phép chọn trên quan hệ chủ của đường liên kết này.
- ❖ Đường liên kết giữa quan hệ chủ và quan hệ bộ phận được định nghĩa là một phép kết bằng.
- ❖ Một phép kết bằng có thể được thực hiện bằng các phép nửa kết.

Thiết kế phân mảnh ngang dẫn xuất

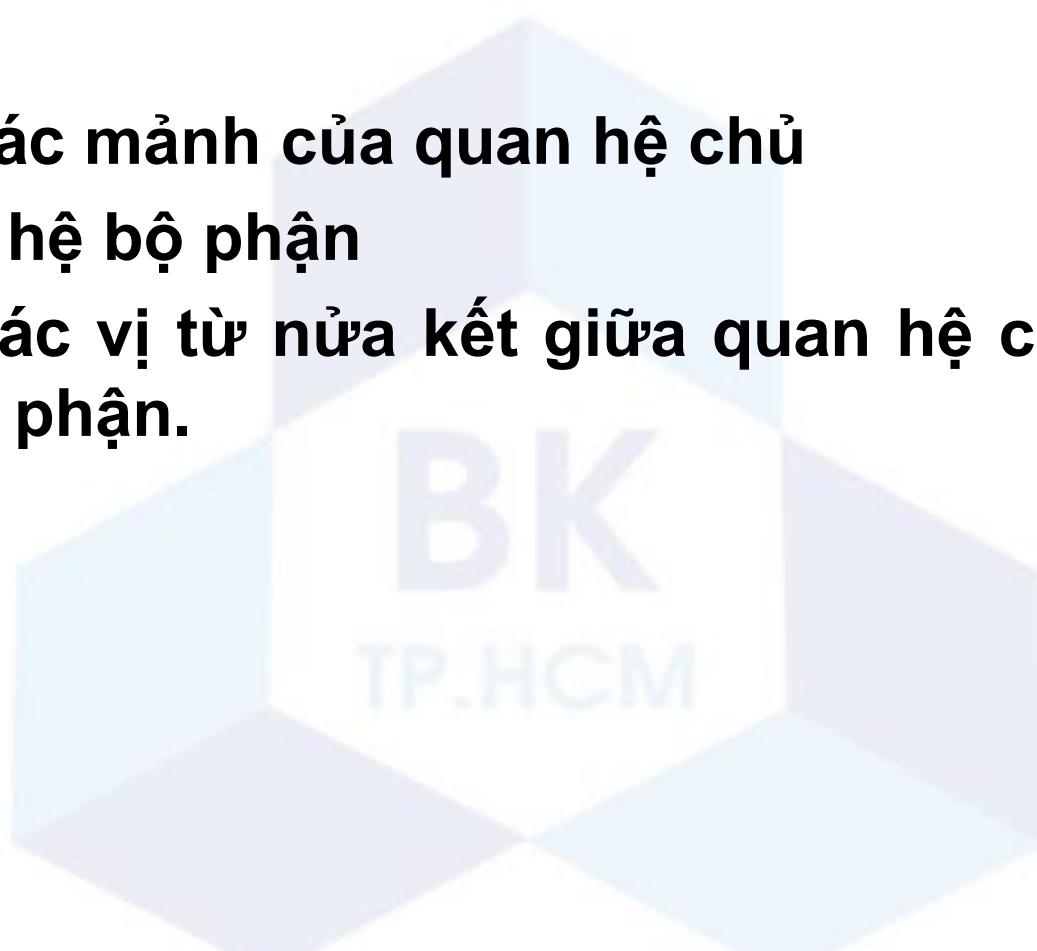
- ❖ Xét đường liên kết L với $owner(L) = S$ và $member(L) = R$, các mảnh ngang dẫn xuất của R được định nghĩa như sau:

$$R_i = R \triangleright_{<_F} S_i, \quad 1 \leq i \leq n$$

- ▶ n là số lượng lớn nhất các mảnh được định nghĩa trên R .
- ▶ $S_i = \sigma_{F_i}(S)$ với F_i là công thức dùng để định nghĩa mảnh ngang chính S_i
- ▶ F là điều kiện nửa kết.

Thiết kế phân mảnh ngang dẫn xuất

- ❖ Để thực hiện phân mảnh ngang dẫn xuất, cần có:
 - ▶ Tập các mảnh của quan hệ chủ
 - ▶ Quan hệ bộ phận
 - ▶ Tập các vị từ nửa kết giữa quan hệ chủ và quan hệ bộ phận.



Thiết kế phân mảnh ngang dẫn xuất

- ❖ **Phép kết phân tán** (*distributed join*) là một phép kết giữa các quan hệ được phân mảnh ngang.

$$R \bowtie_F S = (\cup_i R_i) \bowtie_F (\cup_j S_j)$$

$$R \bowtie_F S = \cup_{ij} (R_i \bowtie_F S_j)$$

- ▶ Có thể suy diễn để xác định một số phép kết từng phần $R_i \bowtie_F S_j = \emptyset$.
- ❖ Phép kết phân tán được biểu diễn bằng **đồ thị kết** (*join graph*).

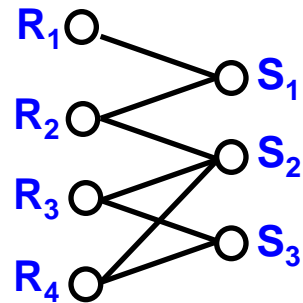
Thiết kế phân mảnh ngang dẫn xuất

- ❖ Đồ thị kết của phép kết phân tán $R \triangleright \triangleleft_F S$ là một đồ thị $G = \langle N, E \rangle$.
 - ▶ Các nút N biểu diễn các mảnh của R và S .
 - ▶ Các cạnh vô hướng giữa các nút biểu diễn các phép kết giữa các mảnh mà kết quả khác rỗng.
 - ▶ Để đơn giản, không đưa vào trong N các mảnh của R hoặc của S mà chúng kết với tất cả các mảnh của quan hệ khác đều cho ra kết quả rỗng.

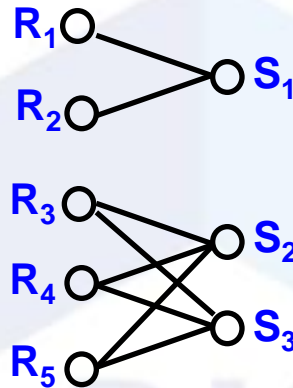
Thiết kế phân mảnh ngang dẫn xuất

- ❖ Đồ thị kết được gọi là **hoàn toàn** (*total*) nếu nó chứa tất cả các cạnh có thể có giữa các mảnh của R và S .
- ❖ Đồ thị kết được gọi là **suy giảm** (*reduced*) nếu không có một số cạnh giữa các mảnh của R và S .
 - ▶ Đồ thị kết suy giảm được gọi là **phân hoạch** (*partitioned*) nếu nó bao gồm hai hoặc nhiều đồ thị con và không có các cạnh giữa chúng.
 - ▶ Đồ thị kết suy giảm được gọi là **đơn giản** (*simple*) nếu nó là phân hoạch và mỗi đồ thị con có đúng một cạnh.

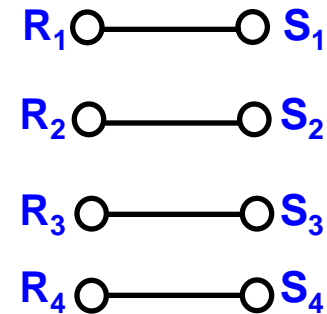
Thiết kế phân mảnh ngang dẫn xuất



(a) Đồ thị kết



(b) Đồ thị kết phân hoạch



(c) Đồ thị kết đơn giản

Hình 5.3. Các loại đồ thị kết.

Thiết kế phân mảnh ngang dẫn xuất

- ❖ Có thể có nhiều đường liên kết đến một quan hệ R và có nhiều cách phân mảnh ngang dẫn xuất cho R dựa trên hai tiêu chuẩn:
 - ▶ Sự phân mảnh có các đặc điểm kết tốt hơn.
 - ▶ Sự phân mảnh được sử dụng trong nhiều ứng dụng hơn.

