

# BÀI 2: THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

## Mục tiêu:

---

- Vấn đề thiết kế
- Các chiến lược thiết kế (trên xuống, dưới lên)
- Phân mảnh dữ liệu (ngang, đứng)
- Cấp phát và nhân bản các phân mảnh

- Khái niệm:

Ra các quyết định về việc bố trí dữ liệu và chương trình ở các vị trí khác nhau của một mạng máy tính và có thể bao gồm việc thiết kết mạng

Trong Hệ quản trị CSDL phân tán, việc bố trí các ứng dụng liên quan đến

- Bố trí các phần mềm DBMS
- Bố trí các ứng dụng chạy trên CSDL

- Cách tiếp cận từ trên xuống (top-down)
  - Thiết kế hệ thống từ đầu
  - Các hệ thống đồng nhất
- Cách tiếp cận từ dưới lên
  - Các CSDL đã tồn tại ở một số vị trí
  - Các CSDL cần được kết nối để giải quyết các nhiệm vụ chung

# Phương pháp thiết kế từ trên xuống (Top-Down) (1)

- Thiết kế từ tổng thể đến riêng biệt
  - Phân rã một hệ thống lớn thành các hệ thống con
- =>Phân tích các yêu cầu nhằm định nghĩa môi trường hệ thống
- =>Thu thập các yêu cầu về dữ liệu và nhu cầu xử lý của các trạm có sử dụng CSDL.

(2)

- **Thiết kế lược đồ tổng quan**

- **Thiết kế view:** xây dựng khung nhìn dữ liệu cho người sử dụng ở các trạm.

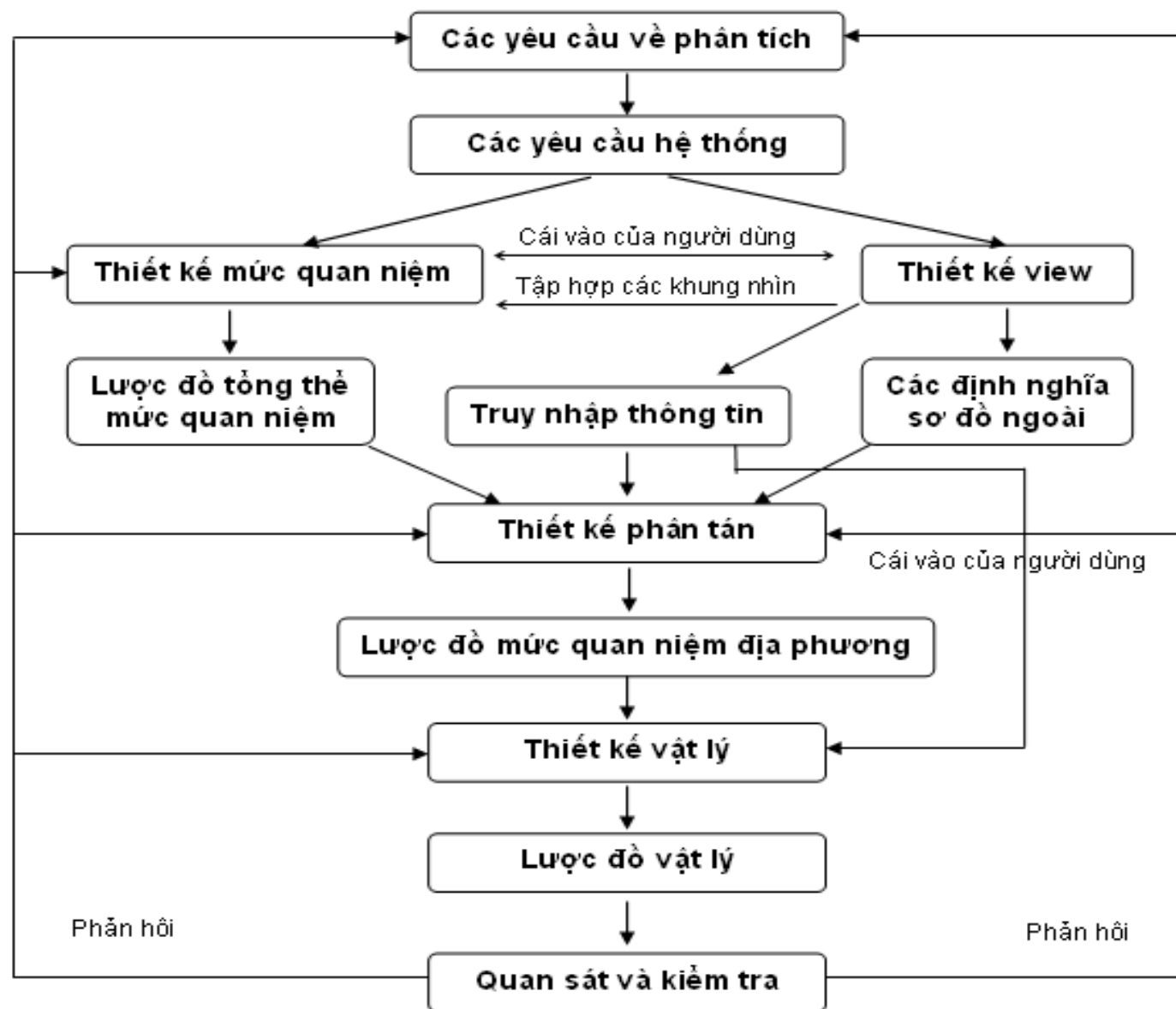
- **Thiết kế mức quan niệm:** là một tiến trình kiểm tra và xác định rõ hai nhóm quan hệ: phân tích thực thể và phân tích chức năng.

- + Phân tích thực thể: xác định các tập thực thể, các thuộc tính và các mối quan hệ giữa chúng.

- + Phân tích chức năng: xác định các chức năng của hệ thống và đưa ra các chức năng cơ sở.

(3)

- **Thiết kế phân phân tán:** Bằng cách tách các quan hệ thành các quan hệ nhỏ ( mảnh), đặt tại các vị trí (site), bao gồm hai phần:
  - + Thiết kế phân đoạn (**Fragmentation**)
  - + Thiết kế định vị và nhân bản (**Allocation and Replication**)
- **Thiết kế lược đồ quan niệm địa phương:** tạo ra các lược đồ mức quan niệm tại các địa phương
- **Thiết kế vật lý:** thực hiện ánh xạ lược đồ mức quan niệm tại các địa phương ra các đơn vị lưu trữ vật lý
- **Quan sát và kiểm tra:** kiểm tra các giai đoạn của quá trình thiết kế cơ sở dữ liệu



Sơ đồ thiết kế CSDL phân tán theo mô hình trên xuống

# Phương pháp thiết kế từ dưới lên (Bottom-Up)

- **Nhận xét**

- + Phương pháp thiết kế trên xuống thực sự có hiệu quả khi xây dựng một hệ thống mới.

- + Trong thực tế, một số CSDL đã tồn tại trước, được tổ chức trong môi trường tập trung và CSDL phân tán được phát triển bằng cách liên kết chúng lại thành một CSDL mới thống nhất (Các DBMS cục bộ khác nhau đã được sử dụng)

- **Cách thiết kế**

1. Chọn một mô hình dữ liệu chung để mô tả lược đồ tổng thể
2. Chuyển mỗi lược đồ cục bộ theo mô hình dữ liệu chung đã chọn
3. Tích hợp các lược đồ cục bộ vào lược đồ tổng thể



# Thiết kế phân tán

- Thiết kế phân mảnh
- Thiết kế định vị

## *Phân mảnh dữ liệu*

---

Thiết kế PT gồm: Thiết kế phân  
đoạn(**Fragmentation**) và Thiết kế định vị và nhân  
bản (**Allocation and Replication**)

---

- ❑ Phân mảnh là quá trình chia một quan hệ toàn cục thành nhiều mảnh có mối quan hệ logic với nhau

# *Tại sao phải phân mảnh?*

---

## *Như vậy*

- ❑ Việc phân mảnh một quan hệ thành nhiều mảnh, mỗi mảnh được xử lý như một đơn vị dữ liệu, cho phép thực hiện nhiều giao dịch đồng thời, làm tăng lưu lượng hoạt động của hệ thống.
- ❑ 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 cách mảnh.

*Tối ưu thời gian, thông lượng, chi phí khi thực hiện truy vấn thông tin  
làm tăng hiệu suất của hệ thống*

## *Các kiểu phân mảnh*

---

- ❑ Một quan hệ thường được biểu diễn dưới dạng bảng. Phân mảnh một bảng thành nhiều bảng con:
- ❑ Phân mảnh dọc: chia Các quan hệ theo chiều dọc. Nghĩa là thiết lập một tập các quan hệ mới từ 1 quan hệ gốc, mỗi quan hệ mới chỉ có một số thuộc tính từ quan hệ gốc. Thực chất đây là phép chiếu trên tập con các thuộc tính của quan hệ.
- ❑ Phân mảnh ngang: Quan hệ được chia theo chiều ngang. Thực chất là phép chọn quan hệ. Chọn những bộ của quan hệ thỏa mãn một biểu thức điều kiện cho trước.
- ❑ Phân mảnh hỗn hợp.

## *Các quy tắc phân mảnh ( Làm thế nào để kiểm tra tính đúng đắn của việc phân mảnh)*

*Các quy tắc đảm bảo cho cơ sở dữ liệu khi phân mảnh sẽ giảm thiểu tổn thất thông tin, mất thông tin hay ít tổn thất thông tin khi thực hiện các truy vấn dữ liệu phân tán. Đảm bảo tính không thay đổi về ngữ nghĩa, toàn vẹn dữ liệu, độc lập dữ liệu.*

- Tính đầy đủ*
- Tính phục hồi*
- Tính tách biệt*

# *Các quy tắc phân mảnh: Tính đầy đủ*

---

□ R được phân rã thành các mảnh  $R_1, R_2 \dots R_k$

□ *Phân mảnh ngang:*

$$\forall r \in R(\Omega) \Rightarrow \exists i \in [1..k] \text{ sao cho } r \in R_i(\Omega)$$

Mục dữ liệu là các n\_bộ

~~□ *Phân mảnh dọc:*~~

$$\forall A \in \Omega \Rightarrow \exists i \in [1..k] \text{ sao cho } A \in \Omega_i$$

Mục dữ liệu là các thuộc tính

□ Quy tắc này đảm bảo cho các mục dữ liệu trong R được ánh xạ hoàn toàn vào các mảnh và không bị mất.

□ Phân rã không tổn thất thông tin

## Các quy tắc phân mảnh: Tính phục hồi(tái cấu trúc)

- Nếu R được phân rã thành các mảnh  $R_1, R_2, \dots, R_k$ .

Khi đó:

*Phân mảnh ngang:*  $R = R_1 \cup R_2 \cup \dots \cup R_k,$

---

Nghĩa là quan hệ toàn cục phục hồi lại bằng cách hợp các quan hệ mảnh con.

*Phân mảnh dọc:*

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

- Quan hệ toàn cục bằng kết nối tự nhiên các quan hệ mảnh

- Tính phục hồi đảm bảo quan hệ toàn cục phân rã không tổn thất thông tin.

## *Các quy tắc phân mảnh: Tính tách biệt*

---

- Nếu R được phân rã thành các mảnh  $R_1, R_2, \dots, R_k$  .  
Khi đó:

*Phân mảnh ngang:*  $\forall i \neq j \in [1..k] \Rightarrow R_i \cap R_j = \phi$

~~Nghĩa là mỗi một n\_bộ của quan hệ toàn cục  
được chứa duy nhất trong một quan hệ con.~~

*Phân mảnh dọc:*

$\forall i \neq j \in [1..k] \Rightarrow \Omega_i \cap \Omega_j = \{\text{các thuộc tính khóa}\}$

Các thuộc tính của quan hệ con chỉ chung nhau  
thuộc tính khóa

- Quy tắc này đảm bảo các mảnh phân rã rời nhau, ko  
bị dư thừa dữ liệu.



# Phân mảnh ngang

PROJ<sub>1</sub> : Các dự án có ngân sách nhỏ hơn \$200,000

PROJ<sub>2</sub> : Các dự án có ngân sách lớn hơn hoặc bằng \$200,000

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Development	350000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

PROJ<sub>1</sub>

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Development	350000	New York

PROJ<sub>2</sub>

PNO	PNAME	BUDGET	LOC
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

PROJ<sub>1</sub>: Thông tin về ngân sách các dự án

PROJ<sub>2</sub>: Thông tin về tên và vị trí của các dự án

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Development	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

PROJ<sub>1</sub>

PNO	BUDGET
P1	150000
P2	135000
P3	250000
P4	310000
P5	500000

PNO	PNAME	LOC
P1	Instrumentation	Montreal
P2	Database Development	New York
P3	CAD/CAM	New York
P4	Maintenance	Paris
P5	CAD/CAM	Boston

# Cơ sở dữ liệu của một công ty máy tính

## NHANVIEN (E)

MANV	TENNV	CHUCVU
A1	Nam	Phân tích HT
A2	Trung	Lập trình viên
A3	Đông	Phân tích HT
A4	Bắc	Phân tích HT
A5	Tây	Lập trình viên
A6	Hùng	Kỹ sư điện
A7	Dũng	Phân tích HT
A8	Chiến	Thiết kế DL

## HOSO(G)

MANV	MADA	NHIEMVU	THOIGIAN
A1	D1	Quản lý	12
A2	D1	Phân tích	34
A2	D2	Phân tích	6
A3	D3	Kỹ thuật	12
A3	D4	Lập trình	10
A4	D2	Quản lý	6
A5	D2	Quản lý	20
A6	D4	Kỹ thuật	36
A7	D3	Quản lý	48
A8	D3	Lập trình	15

## DUAN (J)

MADA	TENDA	NGANSACH
D1	CSDL	20000
D2	CÀI ĐẶT	12000
D3	BẢO TRÌ	28000
D4	PHÁT TRIỂN	25000

## TLUONG (S)

CHUCVU	LUONG
Kỹ sư điện	1000
Phân tích HT	2500
Lập trình viên	3000
Thiết kế DL	4000

# Ví dụ về phân mảnh dữ liệu

*Ví dụ về phân mảnh ngang:* xét các phép toán đại số quan hệ sau:

DUAN 1 =  $\sigma_{\text{NGANSACH} \leq 20000}$  (DUAN)

DUAN 2 =  $\sigma_{\text{NGANSACH} > 20000}$  (DUAN)

DUAN 1

MADA	TENDA	NGANSACH
D1	CSDL	20000
D2	CÀI ĐẶT	12000

DUAN 2

MADA	TENDA	NGANSACH
D3	BẢO TRÌ	28000
D4	PHÁT TRIỂN	25000

Để thấy, các mảnh thỏa mãn *tính tái thiết được* và *tính đầy đủ*

$\text{DUAN 1} \subseteq \text{DUAN}$  ;       $\text{DUAN 2} \subseteq \text{DUAN}$  ;       $\text{DUAN} = \text{DUAN 1} \cup \text{DUAN 2}$

# Ví dụ về phân mảnh dữ liệu

*Ví dụ về phân mảnh đọc:* xét các phép toán đại số quan hệ sau:

$$\text{DUAN 3} = \Pi_{\$1, \$3} \text{DUAN} ; \quad \text{DUAN 4} = \Pi_{\$1, \$4} \text{DUAN}$$

DUAN 3

MADA	NGANSACH
D1	20000
D2	12000
D3	28000
D4	25000

DUAN 4

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

Dễ thấy, các mảnh thỏa mãn *tính tái thiết được* và *tính đầy đủ*

$$\text{DUAN 3} \subseteq \text{DUAN} ; \quad \text{DUAN 4} \subseteq \text{DUAN} ; \quad \text{DUAN} = \text{DUAN 3} \bowtie \text{DUAN 4}$$

*Ví dụ về phân mảnh hỗn hợp:* xét các phép toán đại số quan hệ sau:

$$\text{DUAN 5} = \sigma_{\text{NGANSACH} \leq 20000} (\Pi_{\$1, \$3} \text{DUAN})$$

$$\text{DUAN 6} = \sigma_{\text{NGANSACH} > 20000} (\Pi_{\$1, \$3} \text{DUAN})$$

DUAN 5

TENDA	NGANSACH
CSDL	20000
CÀI ĐẶT	12000

DUAN 6



TENDA	NGANSACH
CSDL	28000
CÀI ĐẶT	25000

- Không nhân bản

- mỗi mảnh nằm ở một vị trí

- Có nhân bản

- Nhân bản toàn phần: Mỗi mảnh ở mọi vị trí
- Nhân bản bán phần: Mỗi mảnh ở 1 số vị trí

## Luật cấp phát:

$$\frac{\text{read - only queries}}{\text{update queries}} \geq 1$$

Nếu

sẽ có lợi khi nhân

bản,

Trường hợp khác, nhân bản sẽ gây nhiều vấn

# Các yêu cầu thông tin cho cấp phát

- Bốn thể loại khác nhau:

- Thông tin CSDL

- Thông tin ứng dụng

- Thông tin mạng truyền thông

- Thông tin hệ thống máy tính