

PHÁT TRIỂN ỨNG DỤNG CSDL 1

# Chương 3. THIẾT KẾ DỮ LIỆU

GVLT: Hoàng Anh Tú

# Nội dung trình bày

- **Thiết kế luận lý dữ liệu**
  - **Thiết kế luận lý cấp cao**
  - **Thiết kế luận lý cấp thấp**
  - **Thiết kế mã**
  - **Thiết kế vật lý dữ liệu**

# Thiết kế luận lý dữ liệu

## Thiết kế luận lý cấp cao:

- Độc lập với mô hình cài đặt
- Dùng chung cho nhiều loại mô hình dữ liệu

## Thiết kế luận lý cấp thấp:

- Chuyển đổi lược đồ kết quả của bước 1 sang một mô hình dữ liệu nhất định (mô hình quan hệ,...)

# Thiết kế luận lý dữ liệu cấp cao

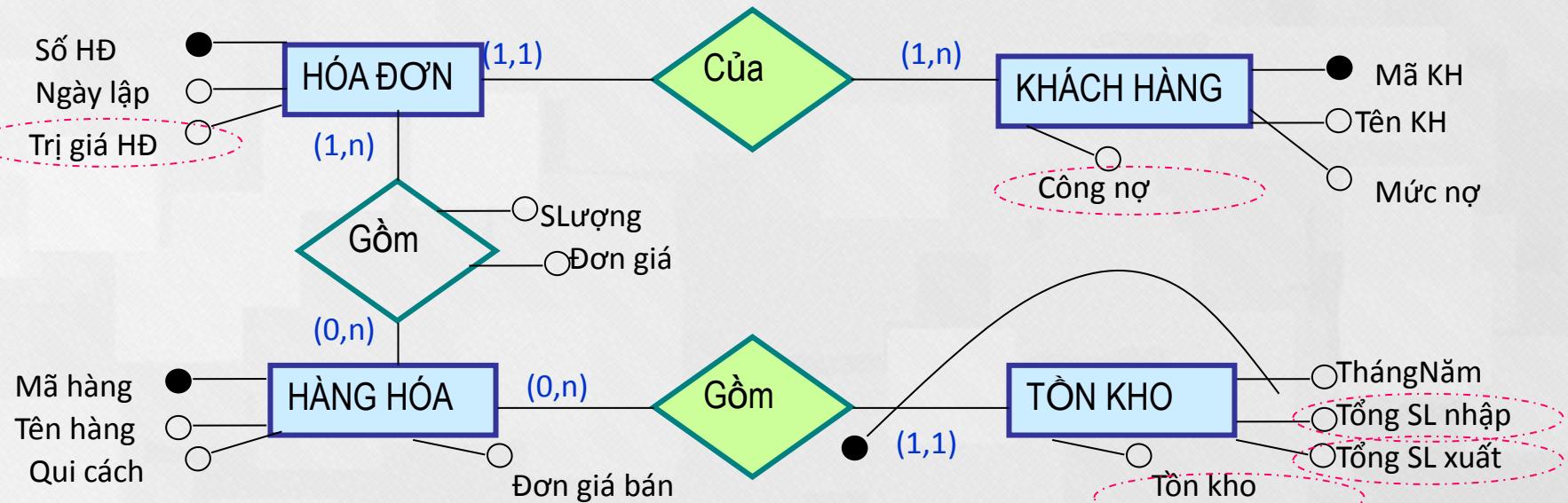
- ➊ Quyết định về dữ liệu suy diễn
- ➋ Chuyển đổi tổng quát hóa và tập con
- ➌ Chọn lựa khóa chính

# Thiết kế luận lý dữ liệu cấp cao

## Quyết định về dữ liệu suy diễn

Dữ liệu suy diễn: là thuộc tính mà giá trị của nó có thể tính toán số học từ những thuộc tính khác

Ví dụ:



# Thiết kế luận lý dữ liệu cấp cao

## Quyết định về dữ liệu suy diễn

### Ưu điểm:

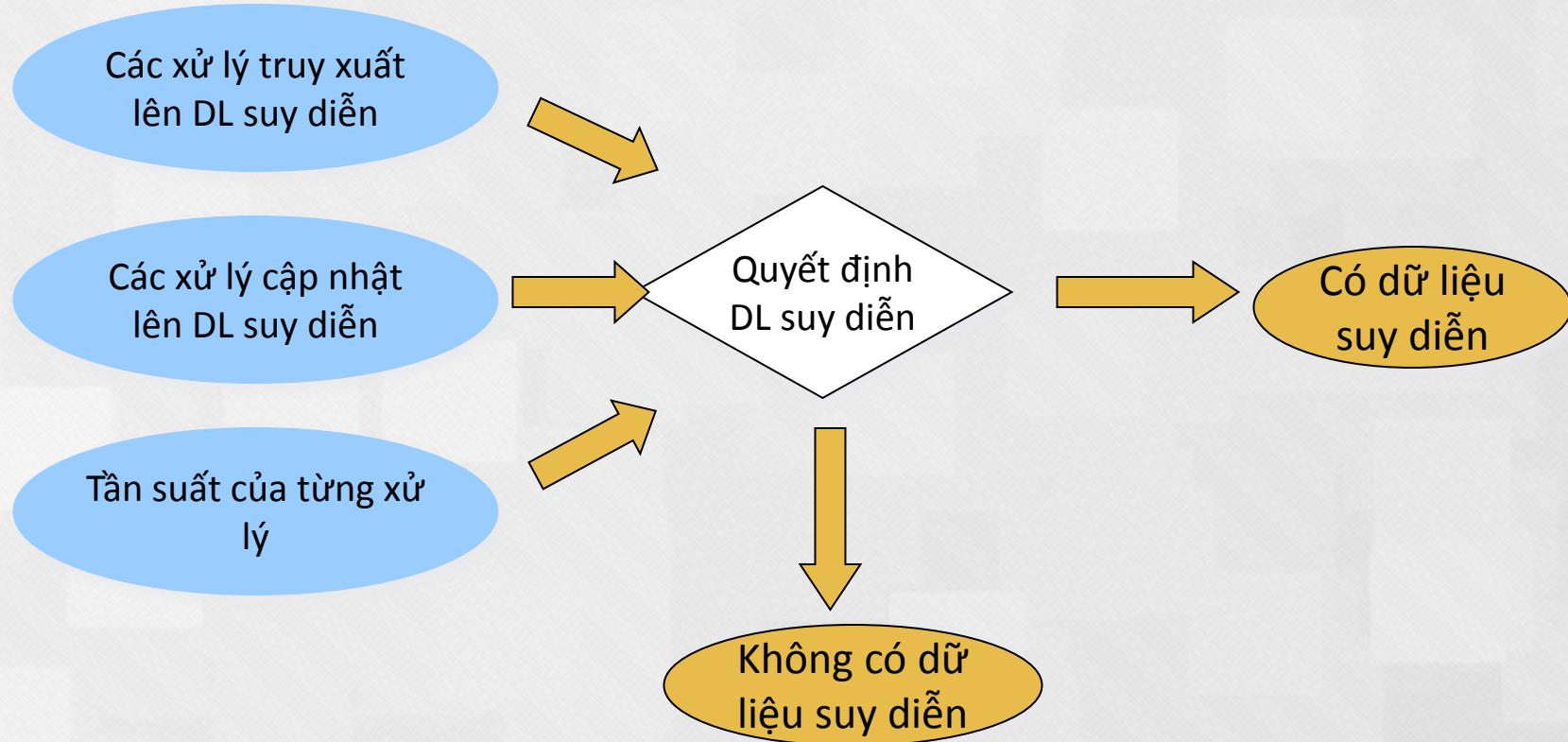
- Tăng tốc độ truy vấn do không phải tính toán lại giá trị các thuộc tính này tại thời điểm thực hiện truy vấn

### Khuyết điểm:

- Cập nhật: phải kiểm tra được tính nhất quán với các dữ liệu liên quan (giảm tốc độ cập nhật)
- Tăng dung lượng lưu trữ (do phải lưu trữ thêm các dữ liệu suy diễn)
- Phải lập trình để đảm bảo ràng buộc về dữ liệu suy diễn

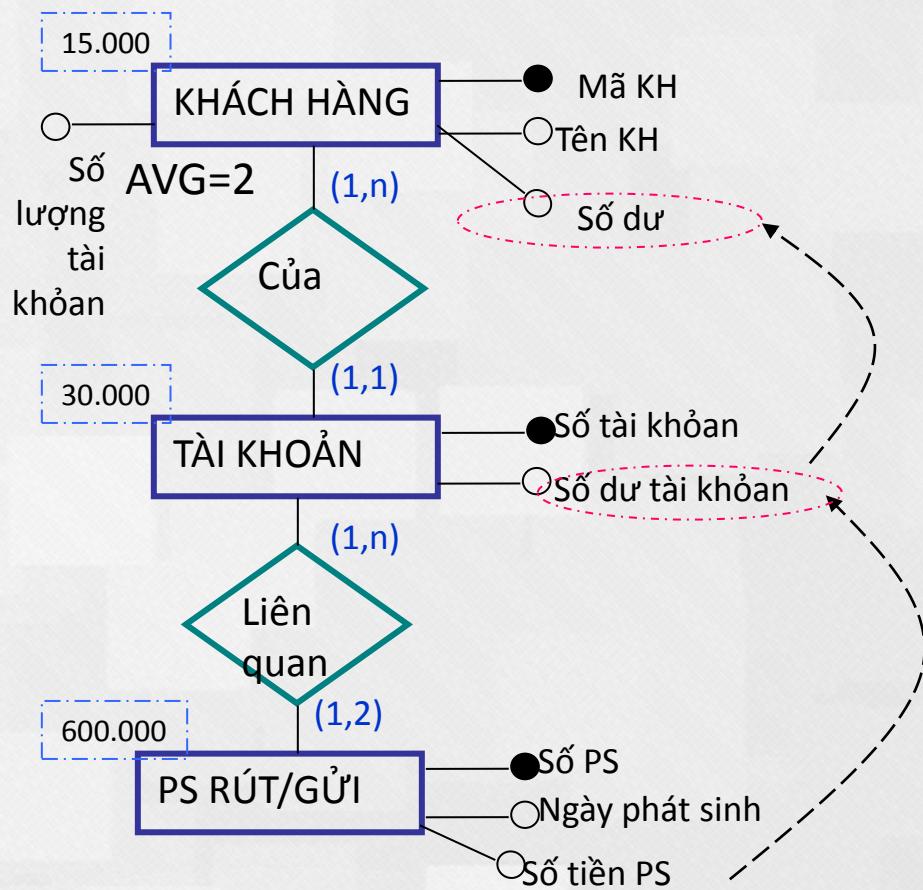
# Thiết kế luận lý dữ liệu cấp cao

## Quyết định về dữ liệu suy diễn



# Thiết kế luận lý dữ liệu cấp cao

## Quyết định về dữ liệu suy diễn – ví dụ:



Khái niệm	Loại	Khối lượng
Khách hàng	Thực thể	15.000
Tài khoản	Thực thể	30.000
Phát sinh	Thực thể	600.000
Của	Kết hợp	30.000
Liên quan	Kết hợp	800.000

# Thiết kế luận lý dữ liệu cấp cao

## Quyết định dữ liệu suy diễn – Ví dụ

Tác vụ	Diễn giải	Tần suất / ngày
o1	Mở tài khoản	100
o2	Đọc số dư tài khoản	3000
o3	Rút tiền	2000
o4	Gửi tiền	1000

Câu hỏi: Có nên sử dụng dữ liệu suy diễn **Số dư** cho thực thể **Khách Hàng** hay không ?

# Thiết kế luận lý dữ liệu cấp cao

- Quyết định về dữ liệu suy diễn – ví dụ:

- Các xử lý liên quan – có dữ liệu suy diễn (A)

Tên tác vụ	Khái niệm	Loại	Đọc Ghi	Tần suất (/ngày) / Bản số t.bình
0 <sub>1</sub> : Mở tài khoản	Tài khoản	Thực thể	Ghi	100
	Khách hàng	Thực thể	Ghi	100
	Của	Kết hợp	Ghi	100
0 <sub>2</sub> : Đọc cân số khách hàng	Khách hàng	Thực thể	Đọc	3000
0 <sub>4</sub> : Rút tiền	Tài khoản	Thực thể	Đọc	2000
	Khách hàng	Thực thể	Ghi	2000
			Đọc	2000
			Ghi	2000
0 <sub>5</sub> : Gởi tiền	Tài khoản	Thực thể	Đọc	1000
	Khách hàng	Thực thể	Ghi	1000
			Đọc	1000
			Ghi	1000

# Thiết kế luận lý dữ liệu cấp cao

- Quyết định về dữ liệu suy diễn – ví dụ:

- Các xử lý liên quan – không có dữ liệu suy diễn (B)

Tên tác vụ	Khái niệm	Loại	Đọc Ghi	Tần suất (/ngày) / Bản số t.bình
$O_1$ : Mở tài khoản	Tài khoản Khách hàng Của	Thực thể Thực thể Kết hợp	Ghi Ghi Ghi	100 100 100
$O_2$ : Đọc cân số khách hàng	Khách hàng <b>Tài khoản</b> Của	Thực thể <b>Thực thể</b> <b>Kết hợp</b>	Đọc <b>Đọc</b> <b>Đọc</b>	3000 <b>3000 x 2 = 6000</b> <b>3000 x 2 = 6000</b>
$O_4$ : Rút tiền	Tài khoản	Thực thể	Đọc Ghi	2000 2000
$O_5$ : Gởi tiền	Tài khoản	Thực thể	Đọc Ghi	1000 1000

(A) 3000 Đ + 3000 G ~ (B) 12000 Đ

# Thiết kế luận lý dữ liệu cấp cao

## Quyết định về dữ liệu suy diễn

- Nếu  $(A) \gg (B)$   $\rightarrow$  chọn không có thuộc tính suy diễn (số dư)  
 $\rightarrow$  do phải trả chi phí có dữ liệu suy diễn (A) nhiều hơn  
không có (B)
- Nếu  $(A) \ll (B)$   $\rightarrow$  chọn có thuộc tính suy diễn (số dư)
- Ngoài ra: có những nhu cầu phát sinh định tính không thể  
quyết định bằng phương pháp định lượng

**(A) 3000 Đ + 3000 G ~ (B) 12000 Đ**

# Thiết kế luận lý dữ liệu cấp cao

## Chuyển đổi tổng quát hóa và tập con

### Cần thiết khi: mô hình cài đặt không hỗ trợ cho vấn đề về tổng quát hóa, thừa kế

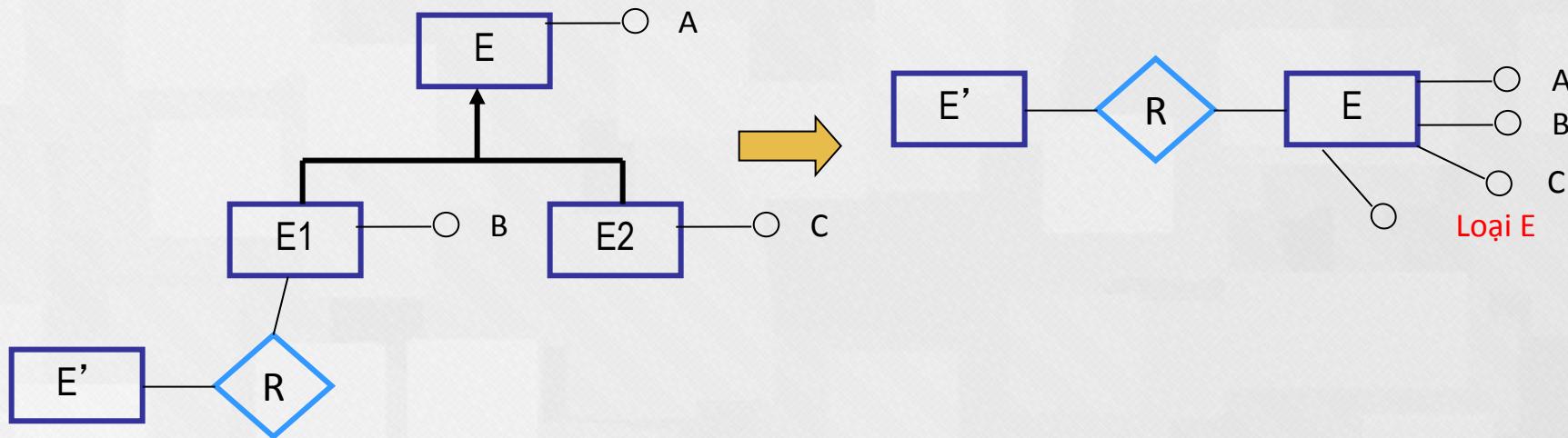
- Ví dụ: mô hình quan hệ, mô hình mạng, mô hình phân cấp,...

### Các việc cần xem xét:

- Các đặc trưng kế thừa (thuộc tính, định danh, mối kết hợp) của thực thể chuyên biệt từ thực thể tổng quát.
- Mối kết hợp tổng quát hóa (Is A).
- Có 3 phương án.

# Thiết kế luận lý dữ liệu cấp cao

## Dùng thực thể tổng quát



### Cách thực hiện:

- Các đặc trưng của các thực thể chuyên biệt sẽ được chuyển qua thực thể tổng quát
- Thêm vào một thuộc tính phân loại (*hoặc một thực thể*) **loại E**
- Loại bỏ các thực thể chuyên biệt

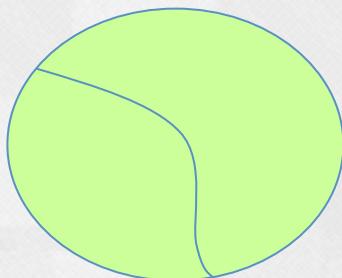
# Thiết kế luận lý dữ liệu cấp cao

## Dùng thực thể tổng quát

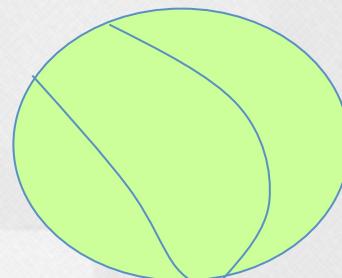
- MGT(LoạiE)  $\cong \{E, E1, E2, E1E2\} \cong \{0,1,2,3\}$

- Biểu diễn sự tương quan:

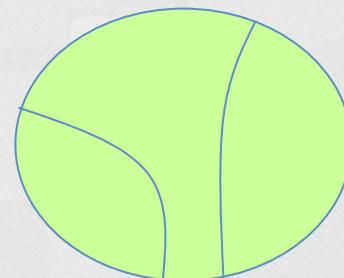
- (t,e):  $\rightarrow$  Ràng buộc MGT(LoạiE)  $\cong \{E1, E2\}$
- (t,o):  $\rightarrow$  Ràng buộc MGT(LoạiE)  $\cong \{E1, E2, E1E2\}$
- (p,e):  $\rightarrow$  Ràng buộc MGT(LoạiE)  $\cong \{E, E1, E2\}$
- (p,o):  $\rightarrow$  Ràng buộc MGT(LoạiE)  $\cong \{E, E1, E2, E1E2\}$



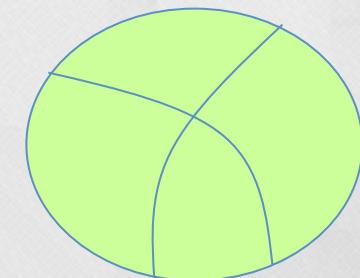
(t,e)



(t,o)



(p,e)

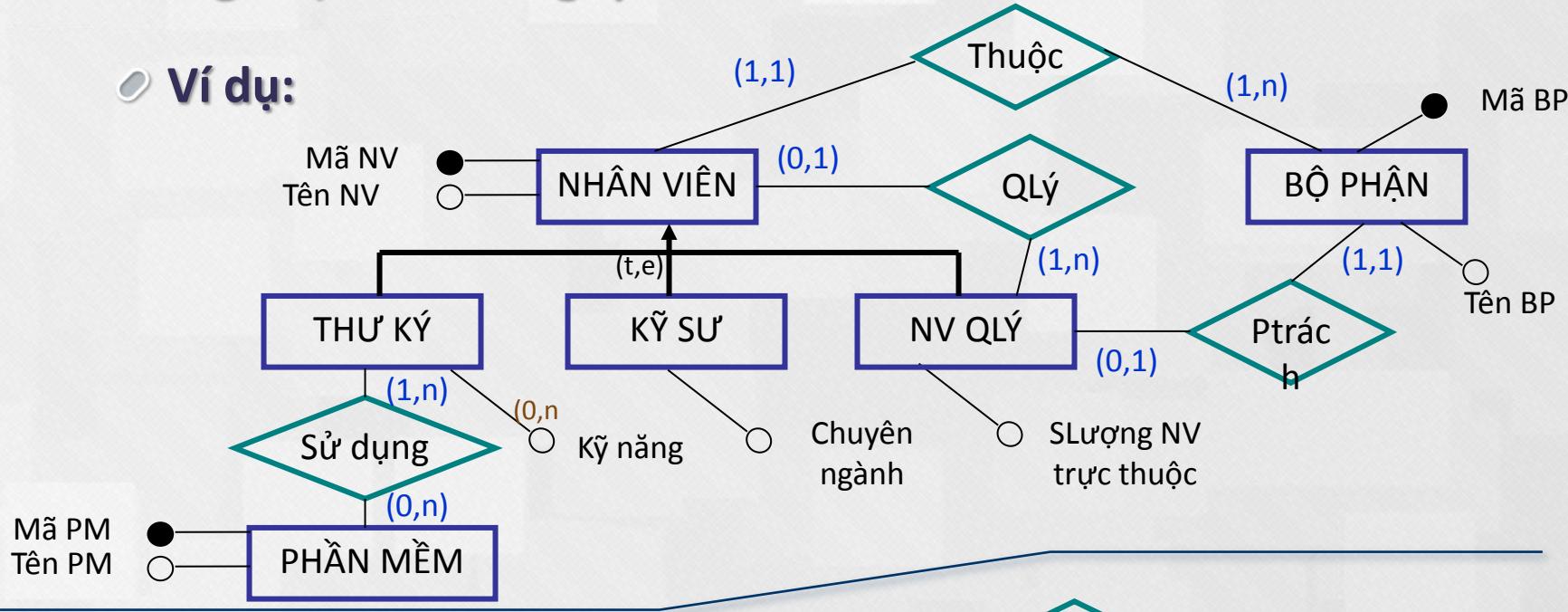


(p,o)

# Thiết kế luận lý dữ liệu cấp cao

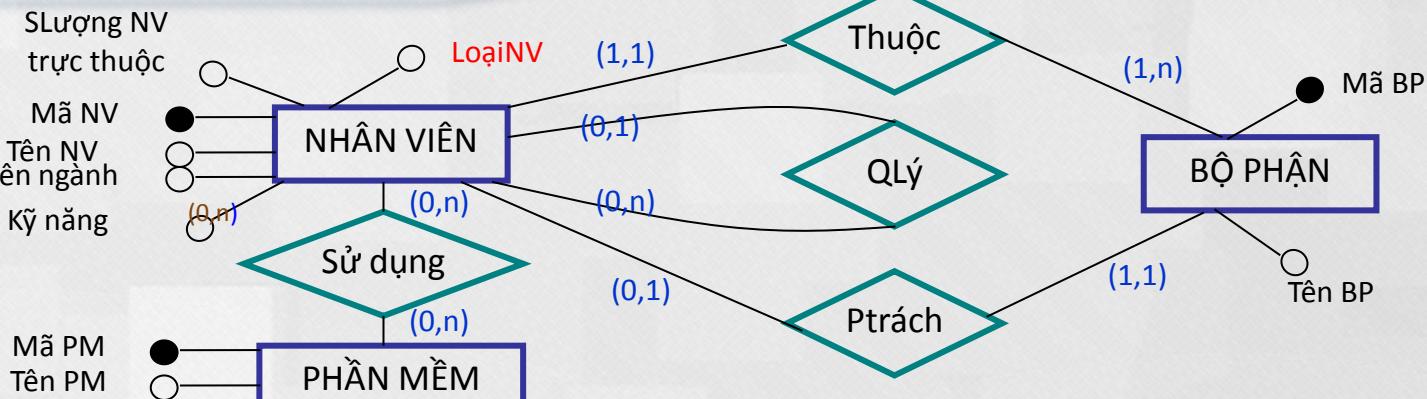
## Dùng thực thể tổng quát

### Ví dụ:



### Ràng buộc:

- LoạiNV ∈ {TK, KS, QL}



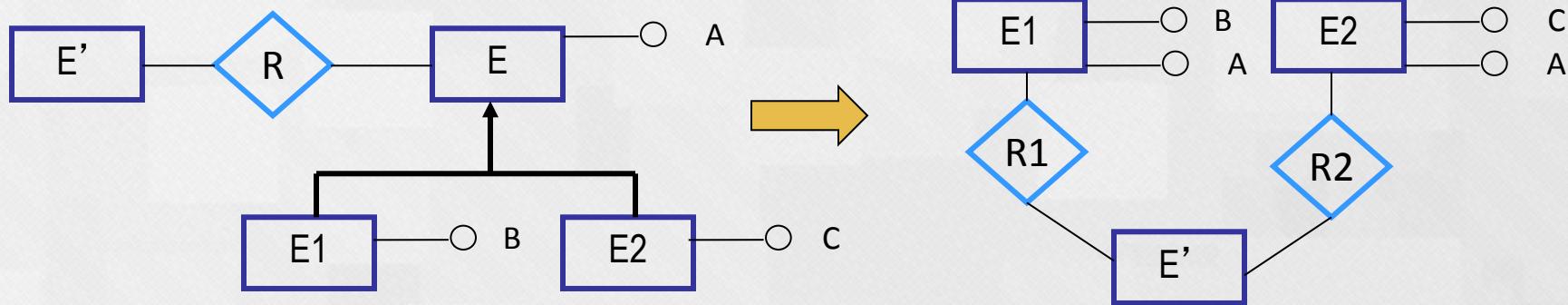
# Thiết kế luận lý dữ liệu cấp cao

## Dùng thực thể tổng quát

Ưu điểm	Khuyết điểm
Giải pháp đơn giản nhất, không phát sinh thêm các mối kết hợp	Có thể phát sinh ra một số lớn các giá trị rỗng cho các thuộc tính mà chỉ dùng cho một loại thực thể tập con mà thôi.
Áp dụng cho tất cả các cấu trúc tổng quát hóa như toàn bộ (t) và bán phần (p), chồng lắp (o) và riêng biệt (e)	Tất cả các tác vụ muốn truy cập đến một thực thể tập con phải truy cập toàn bộ tất cả các thực thể tập con Phát sinh thêm một số RBTV cần phải kiểm tra

# Thiết kế luận lý dữ liệu cấp cao

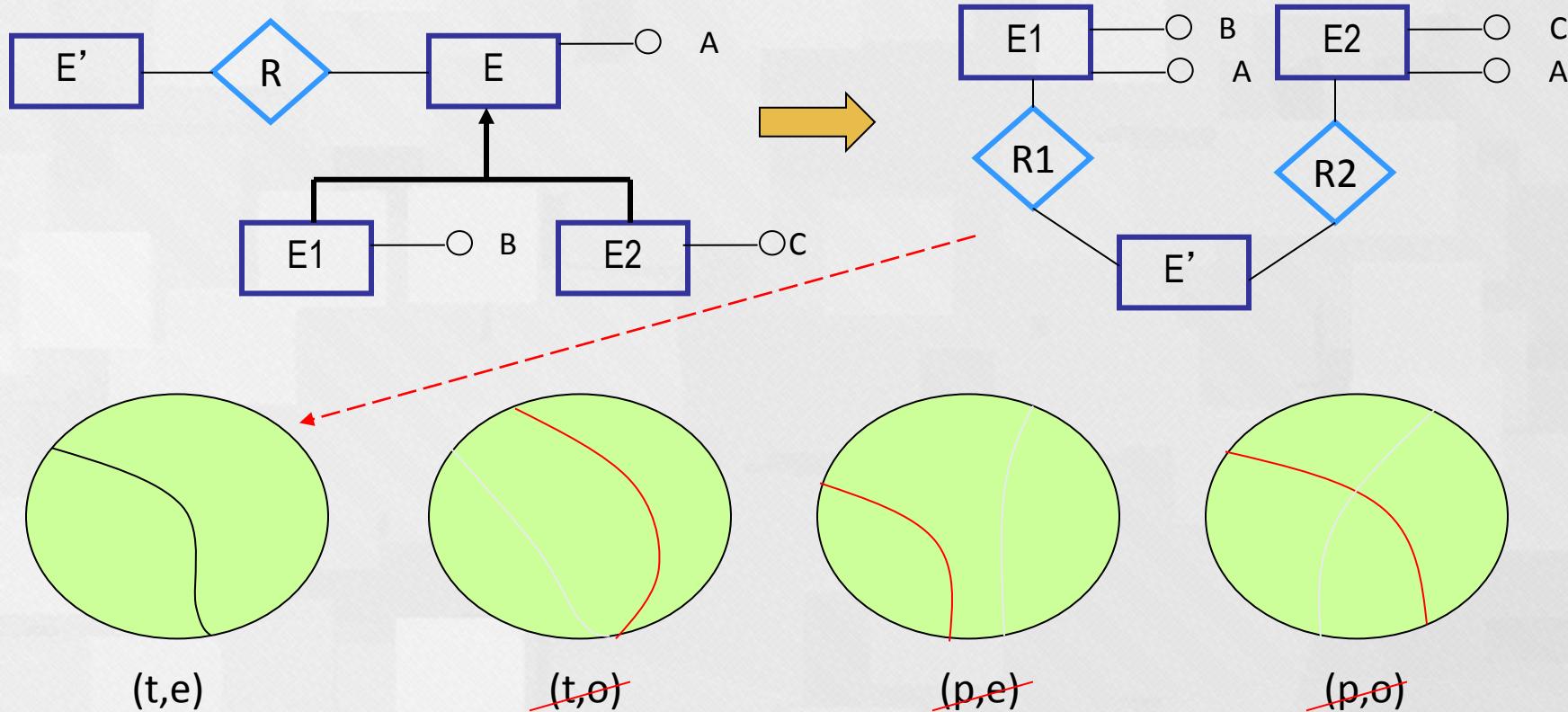
## Dùng các thực thể chuyên biệt



- *Chuyển đổi tất cả đặc trưng (thuộc tính, mối kết hợp, định danh) của thực thể tổng quát xuống lần lượt các thực thể chuyên biệt*
- *Hủy bỏ thực thể tổng quát*

# Thiết kế luận lý dữ liệu cấp cao

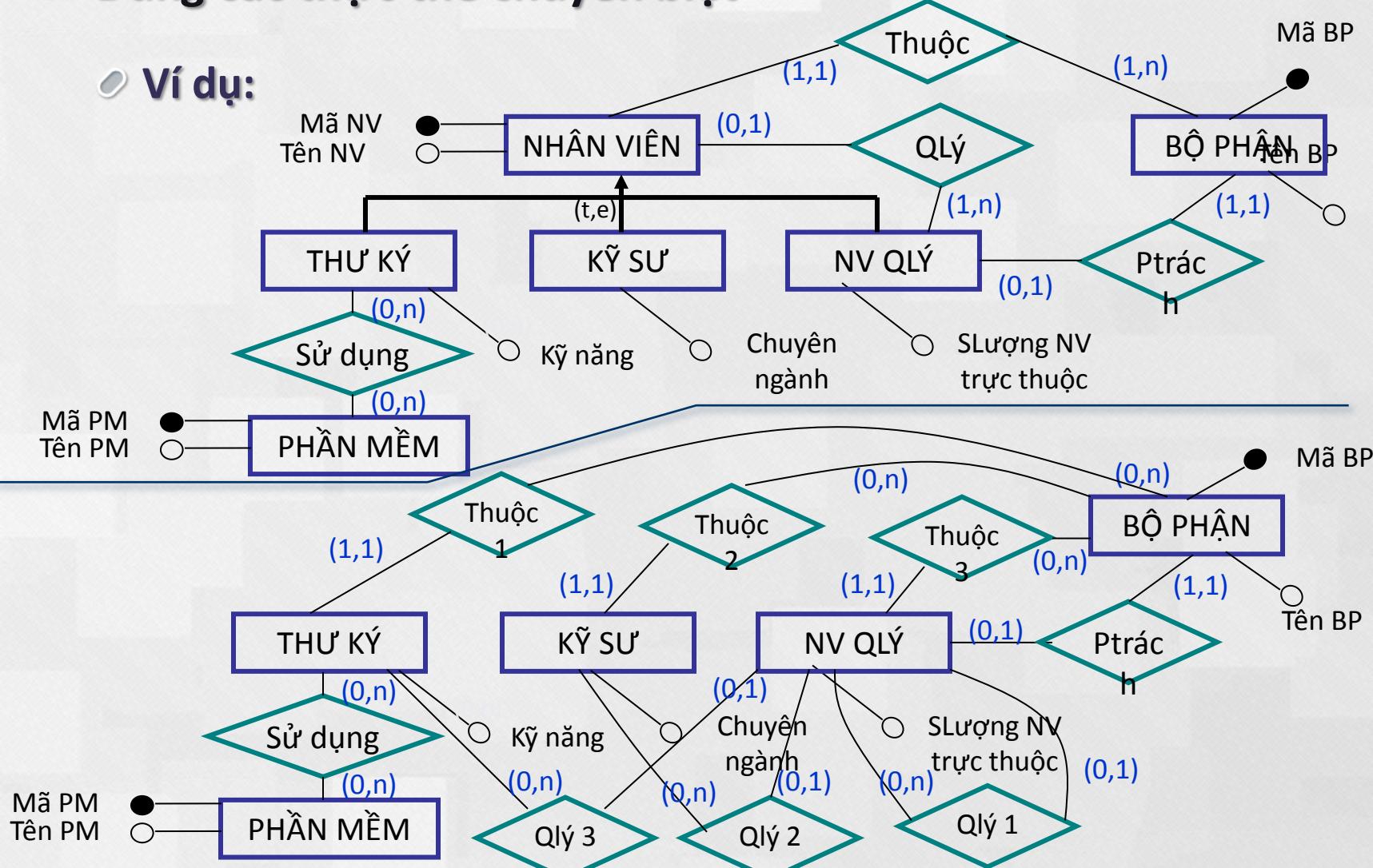
- Dùng các thực thể chuyên biệt
- Ảnh hưởng sự tương quan



# Thiết kế luận lý dữ liệu cấp cao

## Dùng các thực thể chuyên biệt

### Ví dụ:



# Thiết kế luận lý dữ liệu cấp cao

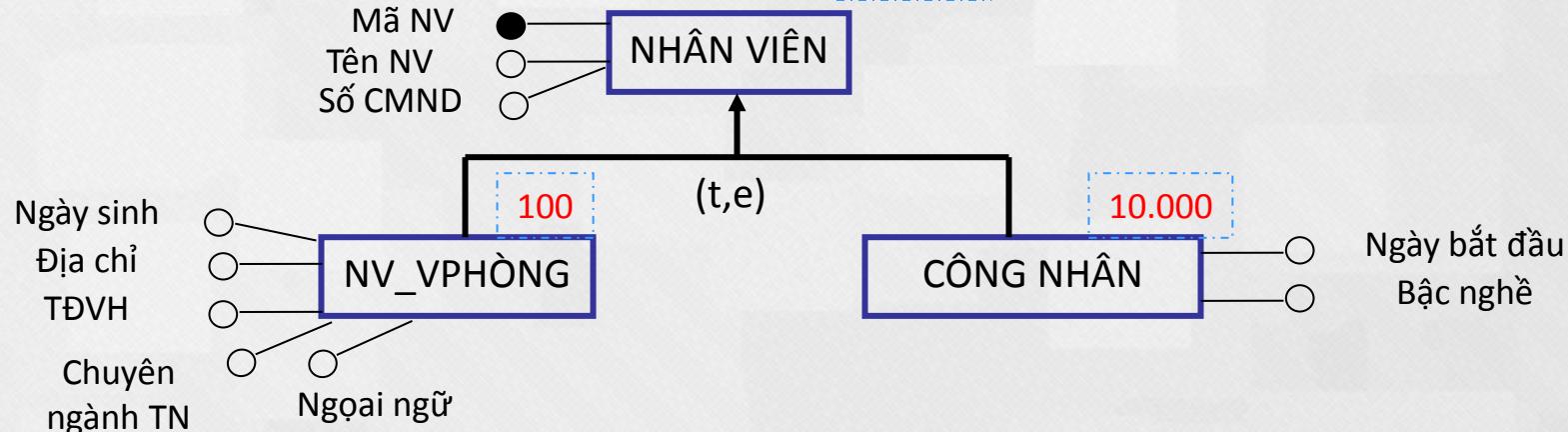
## ○ Dùng các thực thể chuyên biệt

### ○ Nhận xét:

Ưu điểm	Khuyết điểm
Thuộc tính riêng của thực thể tập con chỉ biểu diễn riêng cho loại thực thể tập con đó	Cách chọn lựa này không áp dụng được cho cấu trúc tổng quát hóa loại chồng chéo (o) và bán phần (p), chỉ dùng được cho toàn phần (t) và riêng biệt (e)
Các tác vụ liên quan đến một loại thực thể tập con chỉ truy cập đến loại thực thể tập con đó	Quan niệm các thực thể tập con trước đây cùng là chuyên biệt hóa của một thực thể tổng quát không còn nữa. Quan niệm này có thể là chủ yếu liên quan đến một số xử lý. Ví dụ thư ký, kỹ sư và quản lý là nhân viên.
	Nếu thuộc tính của thực thể tổng quát là đáng kể thì sự lập lại trong lược đồ đáng xem xét lại
	Các tác vụ trước đây thao tác lên thực thể tổng quát nay phải thao tác lên tất cả các thực thể tập con

# Thiết kế luận lý dữ liệu cấp cao

## Chọn lựa giữa 2 phương án :

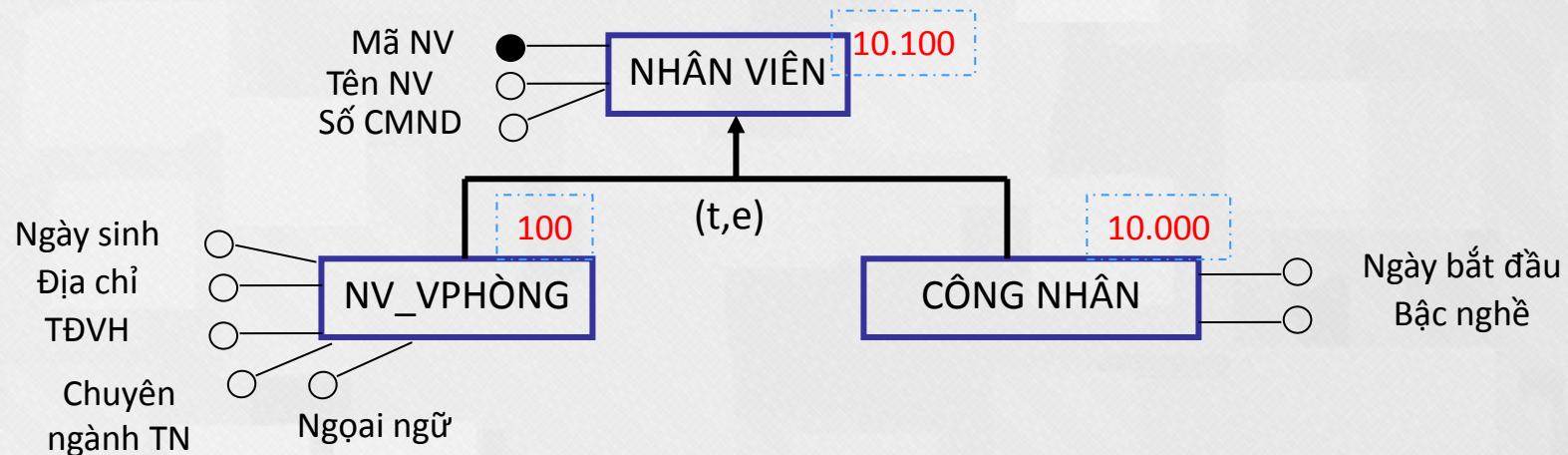


## Xét các xử lý truy xuất liên quan

Xử lý	Tần suất	Chọn lựa tối ưu
(o1) Tính lương sản phẩm cho công nhân phân xưởng	2/ tháng	Tách (gộp)
(o2) Tính lương cho nhân viên văn phòng	1/tháng	Tách
(o3) Tìm kiếm thông tin về công nhân	1000/ngày	Tách (gộp)
(o4) Tổng hợp danh sách chung của toàn bộ nhân viên	5/tháng	Gộp
(o5) Truy xuất thông tin nhân viên văn phòng	20/tháng	Tách

# Thiết kế luận lý dữ liệu cấp cao

## Chọn lựa giữa 2 phương án :



**Nhận xét:** việc chọn lựa phụ thuộc vào:

-**Dung lượng:**

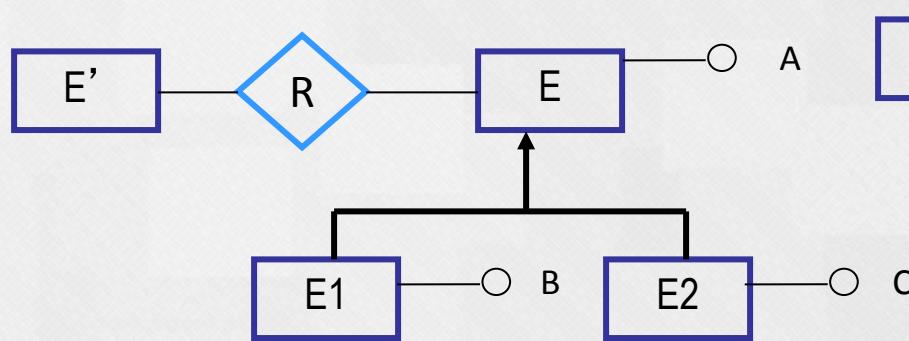
- Gộp: dung lượng lớn (bởi vì phải để trống 10.000 giá trị các thuộc tính NV\_VPHÒNG của các thể hiện CÔNG NHÂN) → truy xuất chậm
- Tách: dung lượng tối ưu

-**Xử lý:** xác định các xử lý ưu tiên (các xử lý có tần xuất cao,...)

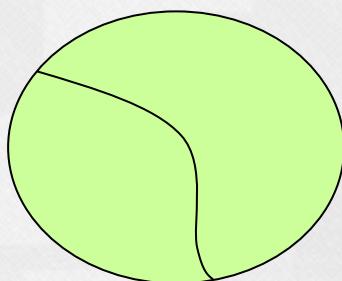
-Ví dụ: nếu ưu tiên cho (o4) → Gộp; (o2), (o3),... → Tách

# Thiết kế luận lý dữ liệu cấp cao

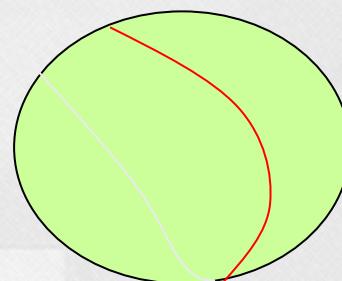
## Dùng mối kết hợp:



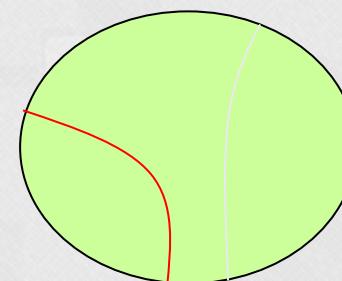
- *Hủy bỏ tổng quát – chuyên biệt*
- *Tạo mối kết hợp từ thực thể tổng quát đến các thực thể chuyên biệt*



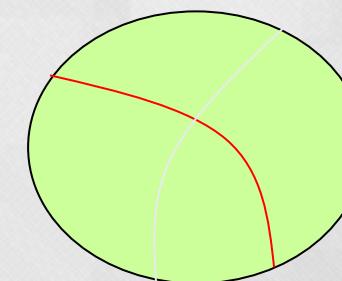
$(t,e)$



$(t,o)$



$(p,e)$

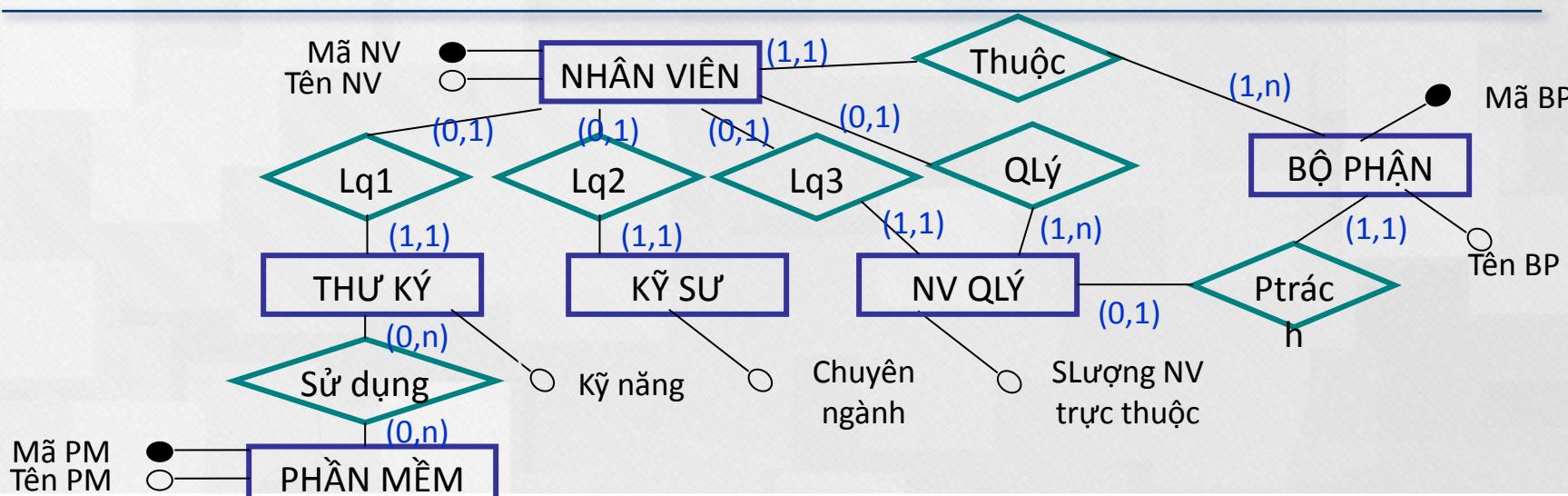
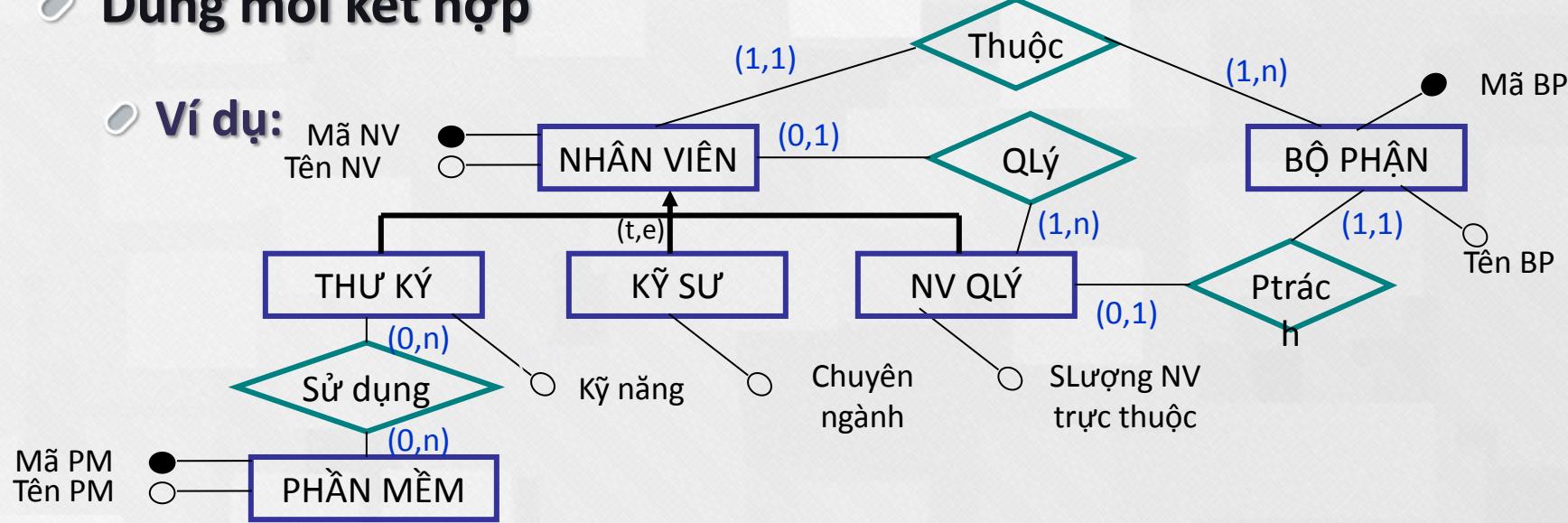


$(p,o)$

# Thiết kế luận lý dữ liệu cấp cao

## Dùng mối kết hợp

### Ví dụ:



# Thiết kế luận lý dữ liệu cấp cao

- ➊ Dùng mối kết hợp

- ➋ Nhận xét

Ưu điểm	Khuyết điểm
Có thể mô hình tất cả các loại cấu trúc tổng quát hóa toàn bộ / bán phần và chồng chéo / riêng biệt	Lược đồ kết quả khá phức tạp. Ví dụ như thêm một thẻ hiện cho một thực thể tập con phải thêm mới một thẻ hiện cho quan hệ và một thẻ hiện cho thực thể tổng quát hóa
Rất uyển chuyển khi thay đổi yêu cầu của ứng dụng	

# Thiết kế luận lý dữ liệu cấp thấp

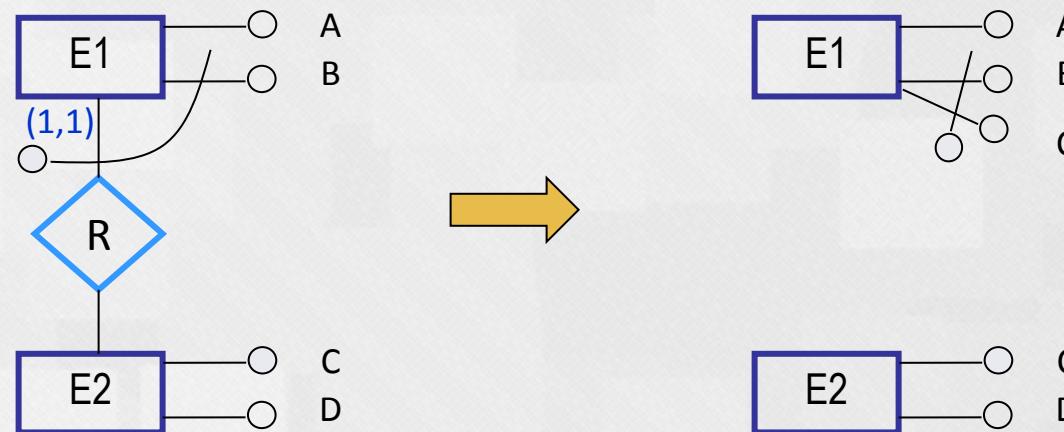
- ⦿ **Bước 1:** Chuẩn bị chuyển đổi

- ⦿ Loại bỏ định danh bên ngoài
- ⦿ Loại bỏ thuộc tính đa trị và thuộc tính kết hợp

- ⦿ **Bước 2:** Chuyển đổi từ mô hình thực thể - kết hợp sang mô hình quan hệ

# Thiết kế luận lý dữ liệu cấp thấp

## Loại bỏ định danh bên ngoài



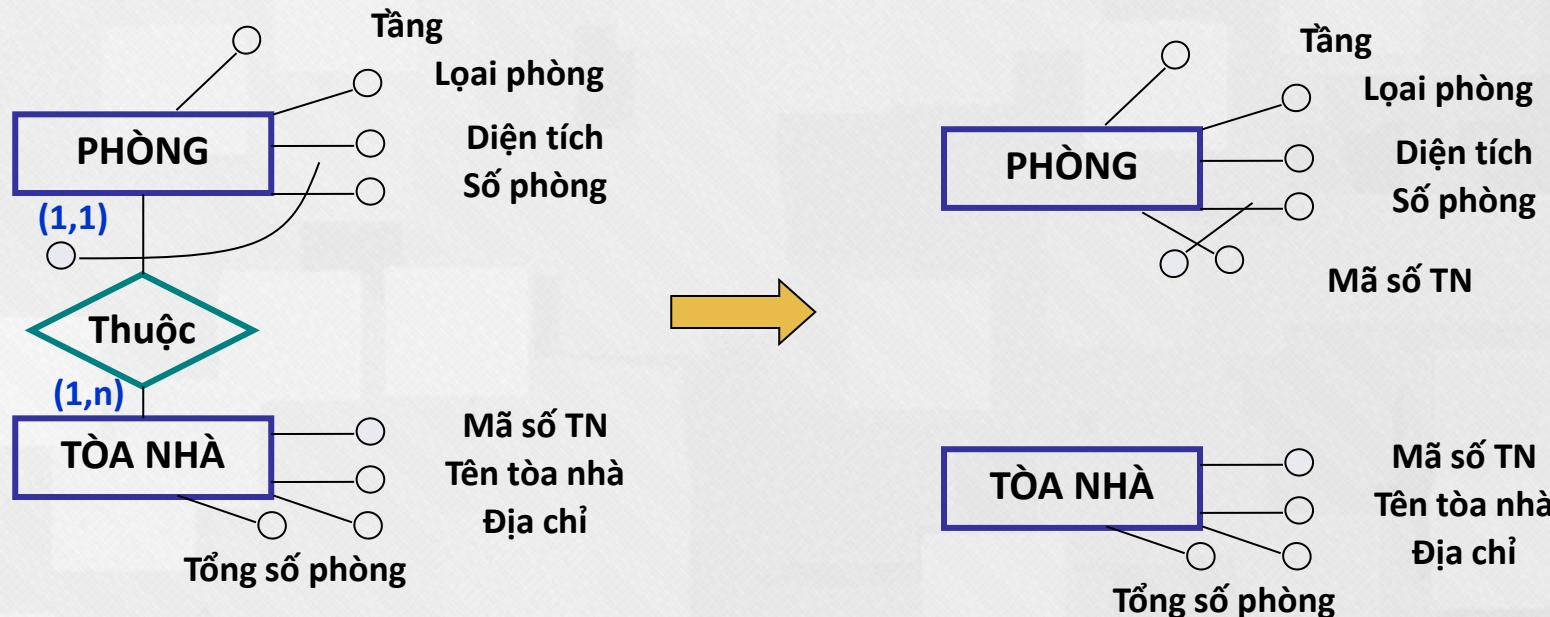
### Cách làm:

- Lấy định danh của các thực thể thuộc phần định danh của thực thể đó kết hợp với thuộc tính định danh làm định danh của thực thể
- Loại bỏ mối kết hợp

# Thiết kế luận lý dữ liệu cấp thấp

- Loại bỏ định danh bên ngoài

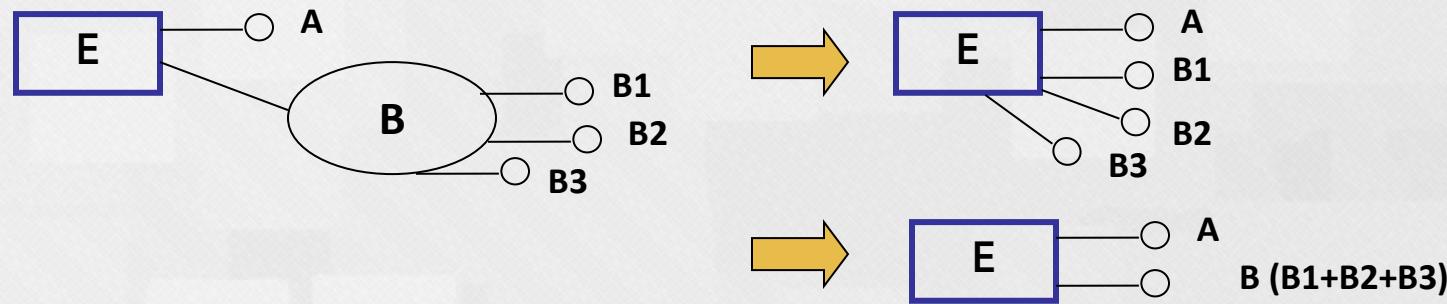
- Ví dụ:



# Thiết kế luận lý dữ liệu cấp thấp

- Loại bỏ thuộc tính đa trị và thuộc tính kết hợp

- Chuyển đổi thuộc tính kết hợp



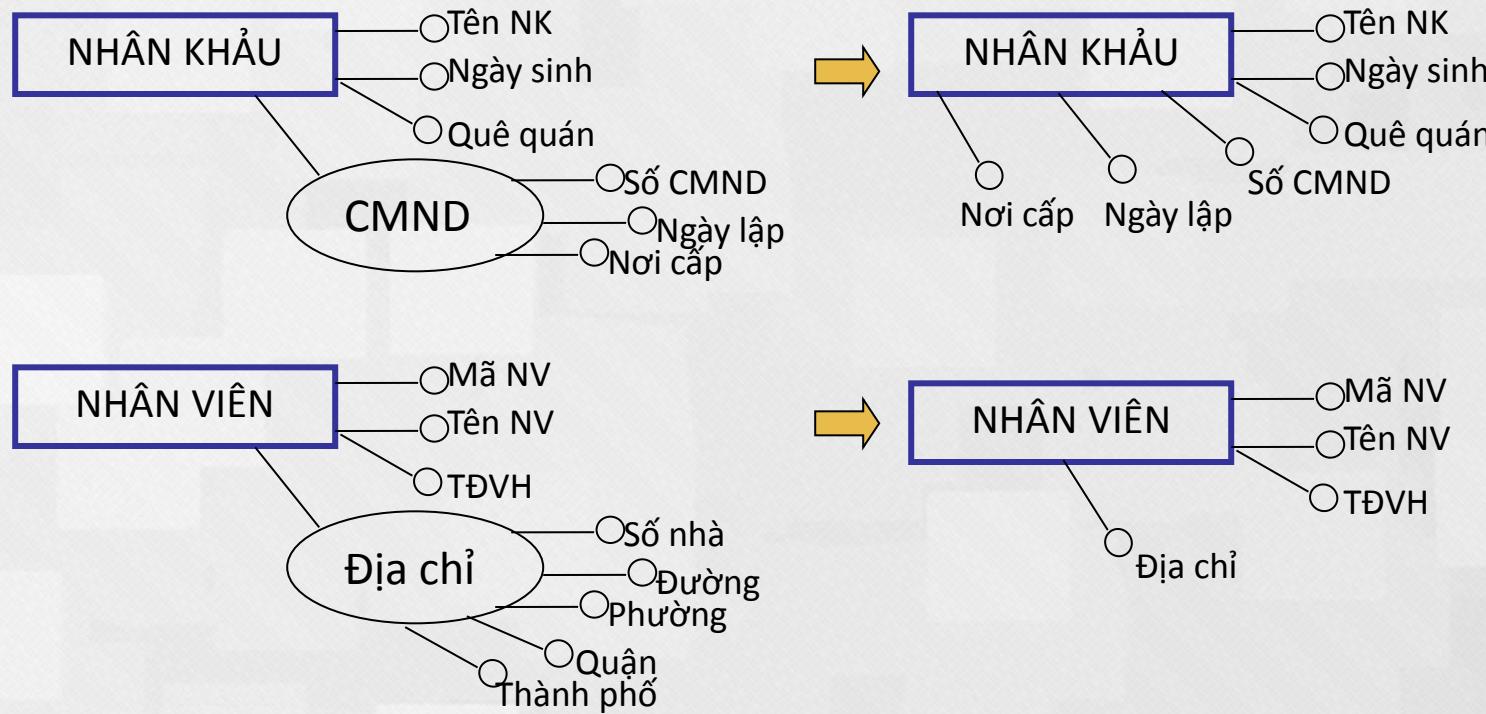
- Thuộc tính thành phần được truy cập nhiều → chuyển các thuộc tính thành phần thành thuộc tính đơn
- Thuộc tính kết hợp được truy cập thường xuyên → chuyển thành một thuộc tính đơn

# Thiết kế luận lý dữ liệu cấp thấp

- Loại bỏ thuộc tính đa trị và thuộc tính kết hợp

- Chuyển đổi thuộc tính kết hợp

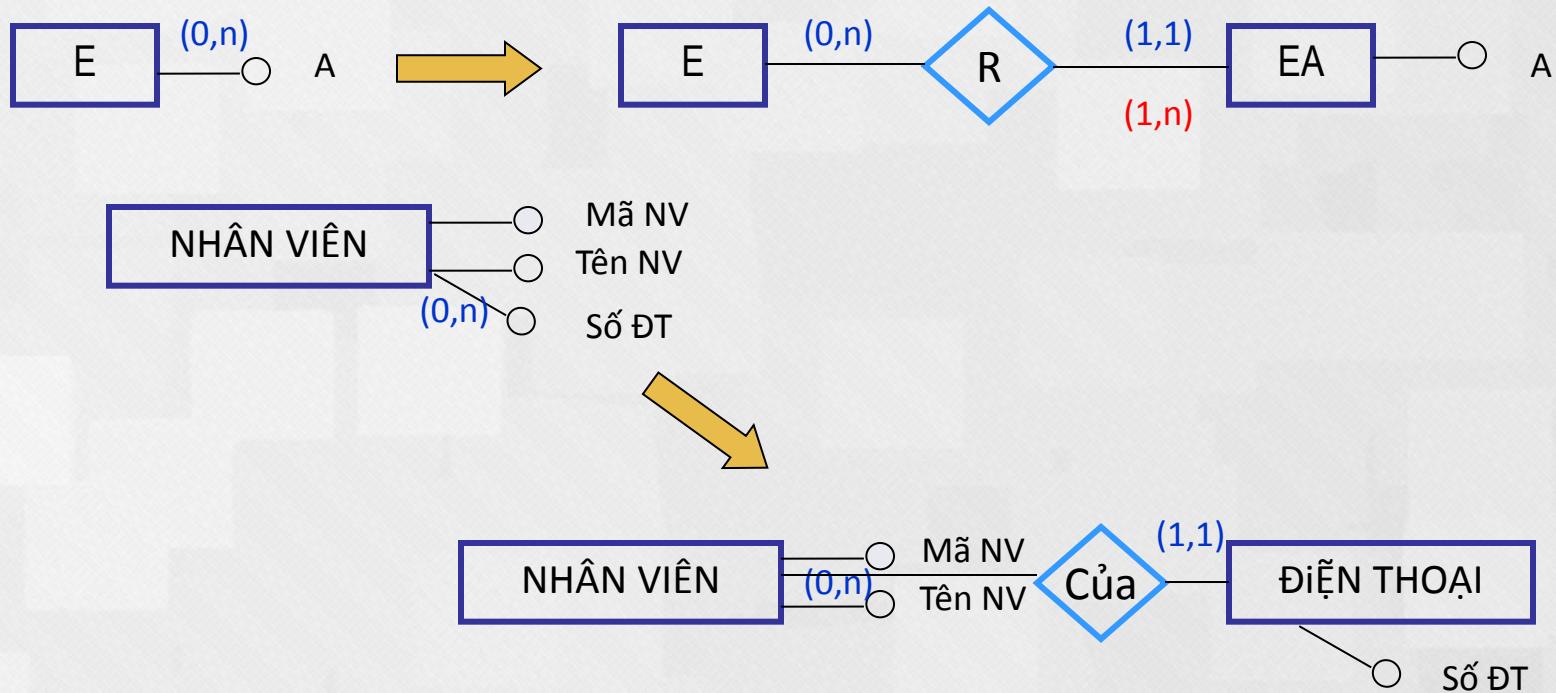
- Ví dụ:



# Thiết kế luận lý dữ liệu cấp thấp

- Loại bỏ thuộc tính đa trị và thuộc tính kết hợp

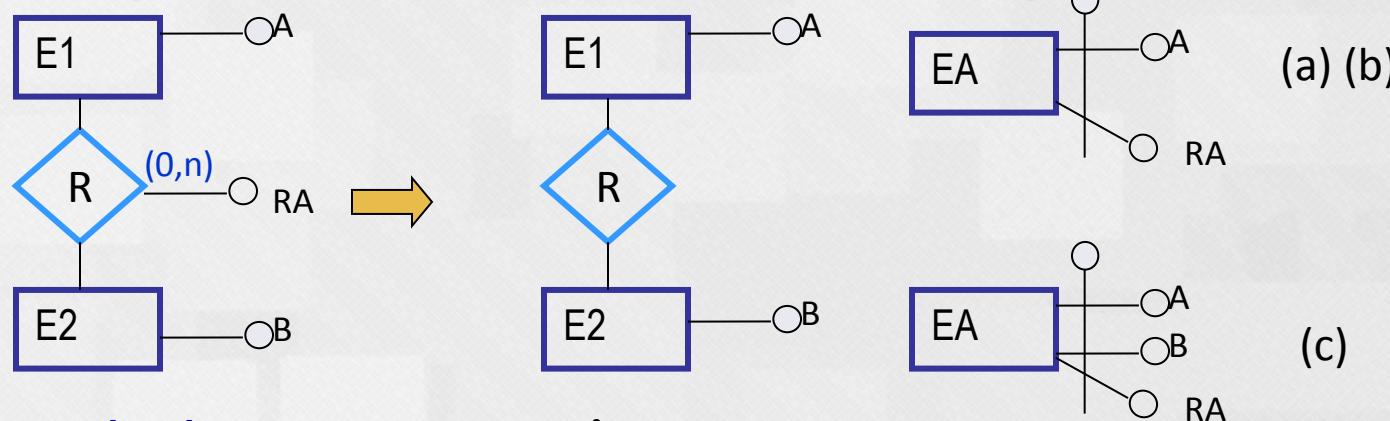
- Chuyển đổi thuộc tính đa trị - thực thể



# Thiết kế luận lý dữ liệu cấp thấp

- Loại bỏ thuộc tính đa trị và thuộc tính kết hợp

- Chuyển đổi thuộc tính đa trị - mối kết hợp



(a) **R là mối kết hợp 1-1**: thực thể EA sẽ có định danh là thuộc tính định danh của E1 hoặc của E2 kết hợp với thuộc tính RA

(b) **R là mối kết hợp 1 – N**: thực thể EA sẽ có định danh là thuộc tính định danh của E1 kết hợp với thuộc tính RA

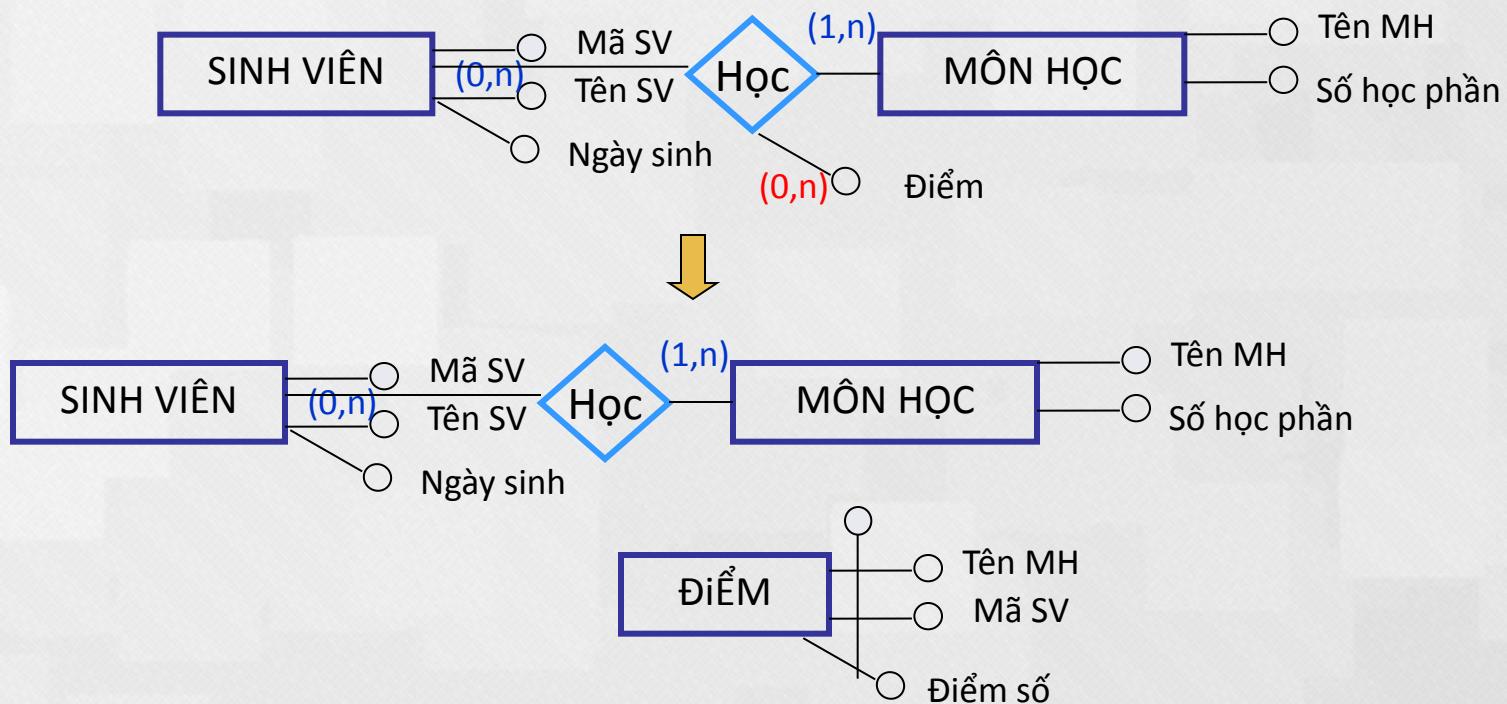
(c) **R là mối kết hợp N – N**: thực thể EA sẽ có định danh là thuộc tính định danh của E1, E2 kết hợp với thuộc tính RA

# Thiết kế luận lý dữ liệu cấp thấp

- Loại bỏ thuộc tính đa trị và thuộc tính kết hợp

- Chuyển đổi thuộc tính đa trị - mối kết hợp

Ví dụ:



# Thiết kế luận lý dữ liệu cấp thấp

## Chuyển đổi từ mô hình TTKH → mô hình Quan hệ

### Chuyển đổi thực thể

- 1 thực thể → 1 lược đồ quan hệ
- Thuộc tính → thuộc tính
- Định danh → khóa (khóa chính)

### Ví dụ :

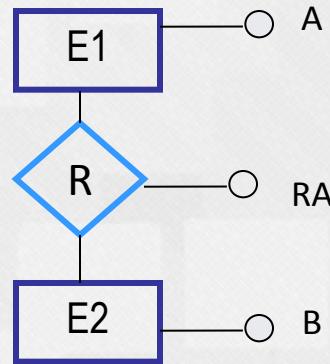


# Thiết kế luận lý dữ liệu cấp thấp

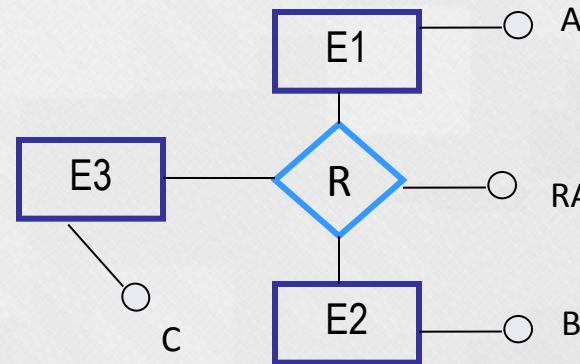
- Chuyển đổi từ mô hình TTKH → mô hình Quan hệ

- Chuyển đổi mối kết hợp

- Nguyên tắc chung



R(A, B, RA)



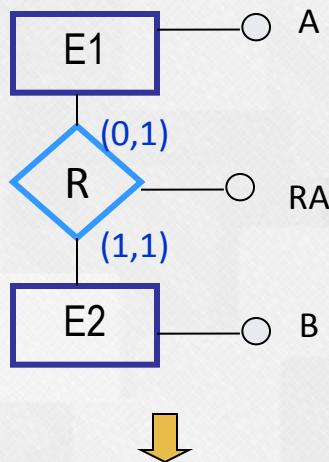
R(A, B, C, RA)

# Thiết kế luận lý dữ liệu cấp thấp

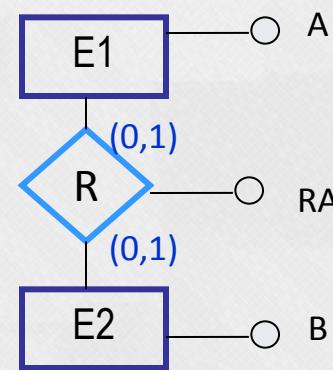
- Chuyển mô hình TTKH → mô hình Quan hệ

- Chuyển đổi mối kết hợp nhị phân

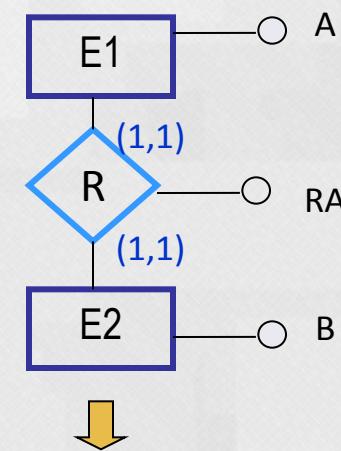
- Mối kết hợp 1-1



$E1(\underline{A}, \dots)$   
 $E2(\underline{B}, \dots, A)$  : A là khóa



$E1(\underline{A}, \dots)$   
 $E2(\underline{B}, \dots, A)$ : A là khóa  
*Hoặc*  
 $E1(\underline{A}, \dots, B)$ : B là khóa  
 $E2(\underline{B}, \dots)$



$E1E2(\underline{A}, B, \dots)$  : B là khóa  
*Hoặc*  
 $E1E2(A, \underline{B}, \dots)$  : A là khóa

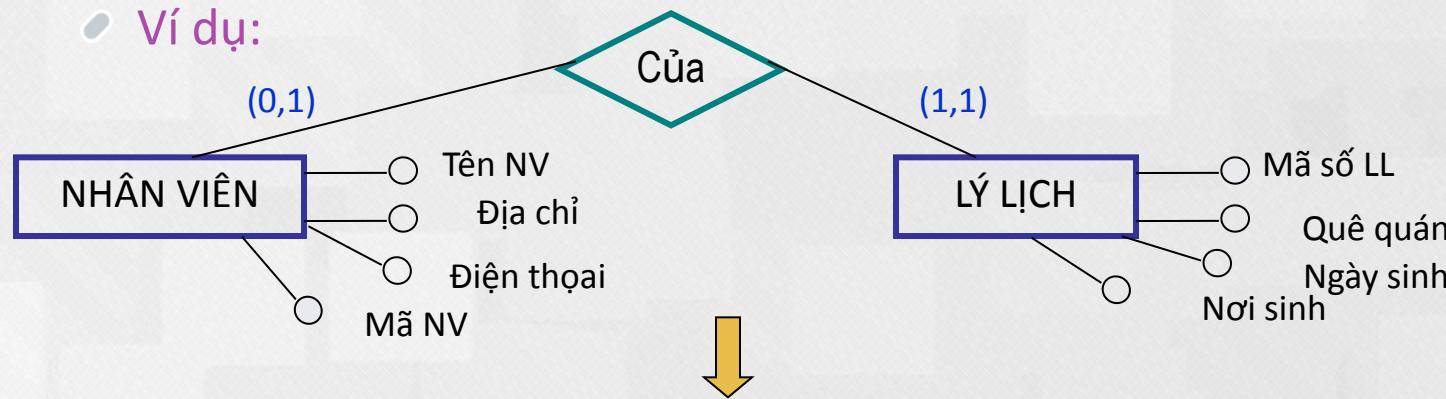
# Thiết kế luận lý dữ liệu cấp thấp

- Chuyển mô hình TTKH → mô hình Quan hệ

- Chuyển đổi mối kết hợp nhị phân

- Mối kết hợp 1-1

- Ví dụ:



-NHÂN\_VIÊN(MA\_NV, TÊN\_NV, ĐỊA\_CHỈ, ĐIỆN\_THOẠI)

-LÝ\_LỊCH(MÃ\_SỐ\_LL, QUÊ\_QUÁN, NGÀY\_SINH, NƠI\_SINH, MA\_NV)

-MA\_NV là một khóa trong quan hệ LÝ\_LỊCH

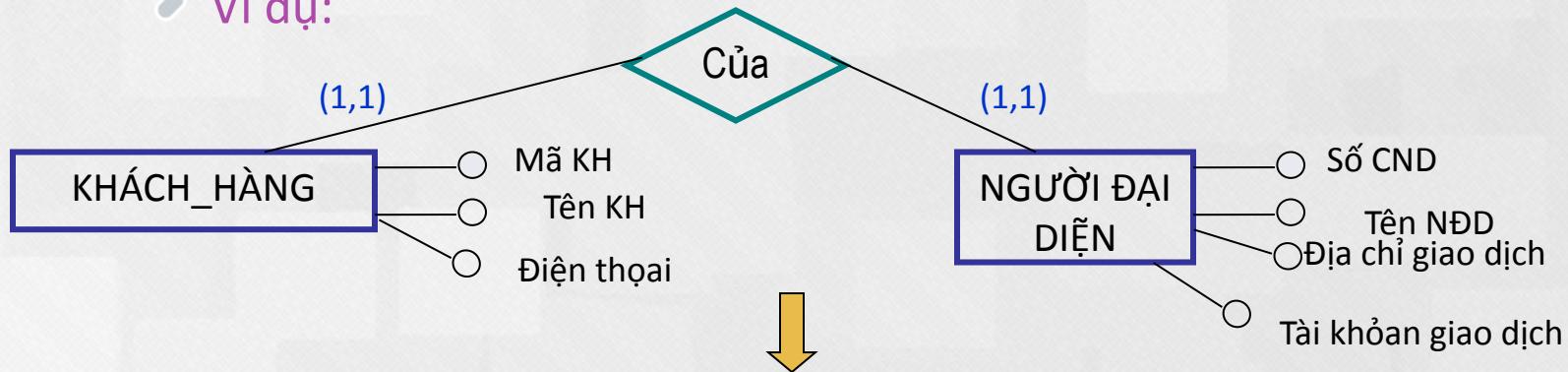
# Thiết kế luận lý dữ liệu cấp thấp

- Chuyển mô hình TTKH → mô hình Quan hệ

- Chuyển đổi mối kết hợp nhị phân

- Mối kết hợp 1-1

- Ví dụ:



**KH\_NĐDIỆN (MA\_KH, TÊN\_KH, ĐIỆN\_THOẠI, SO\_CMND\_ĐD, TÊN\_NĐD, DIA\_CHỈ\_GIAO\_DỊCH, TÀI KHỎAN\_GIAO\_DỊCH)**

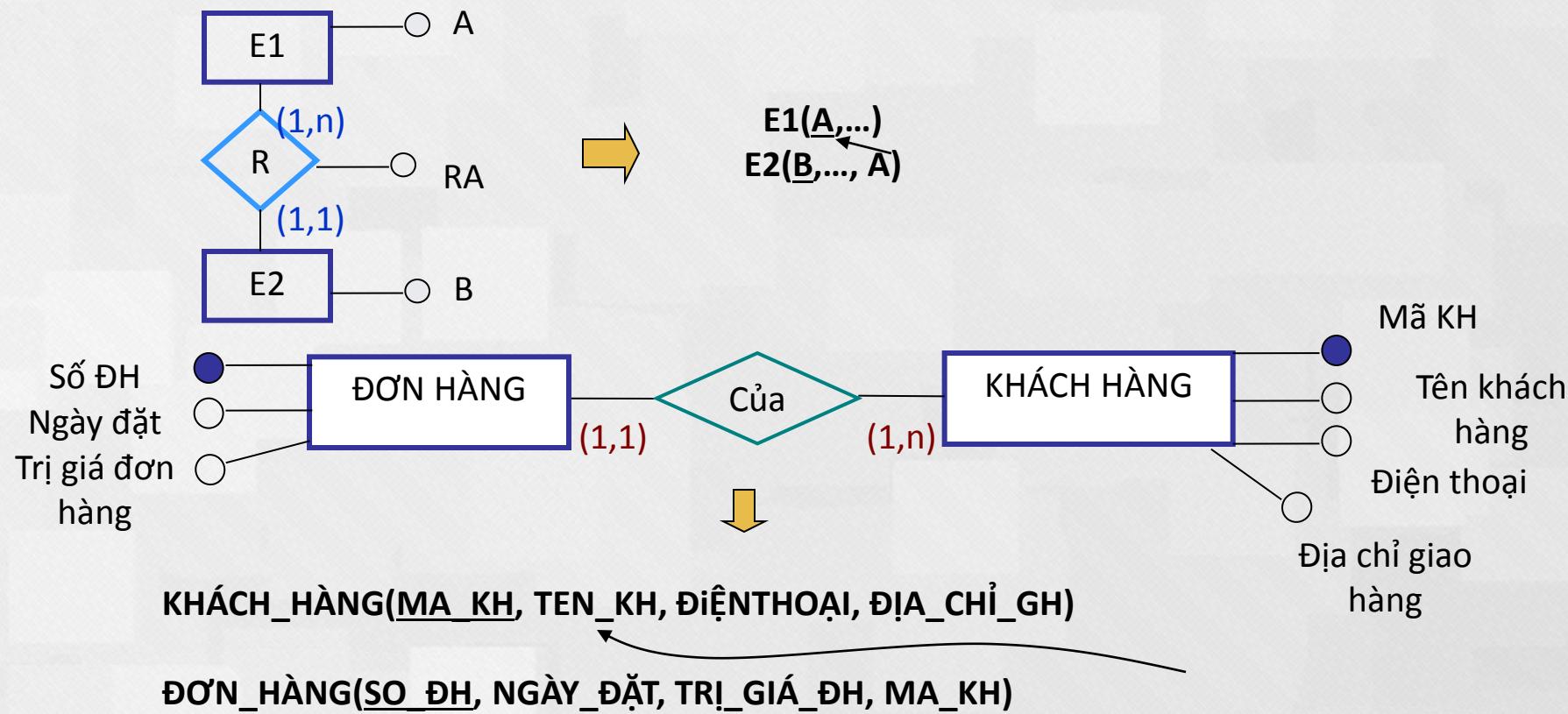
*SO\_CMND\_ĐD là một khóa trong quan hệ KHÁCH\_HÀNG*

# Thiết kế luận lý dữ liệu cấp thấp

- Chuyển mô hình TTKH → mô hình Quan hệ

- Chuyển đổi mối kết hợp nhị phân

- Mối kết hợp 1-N

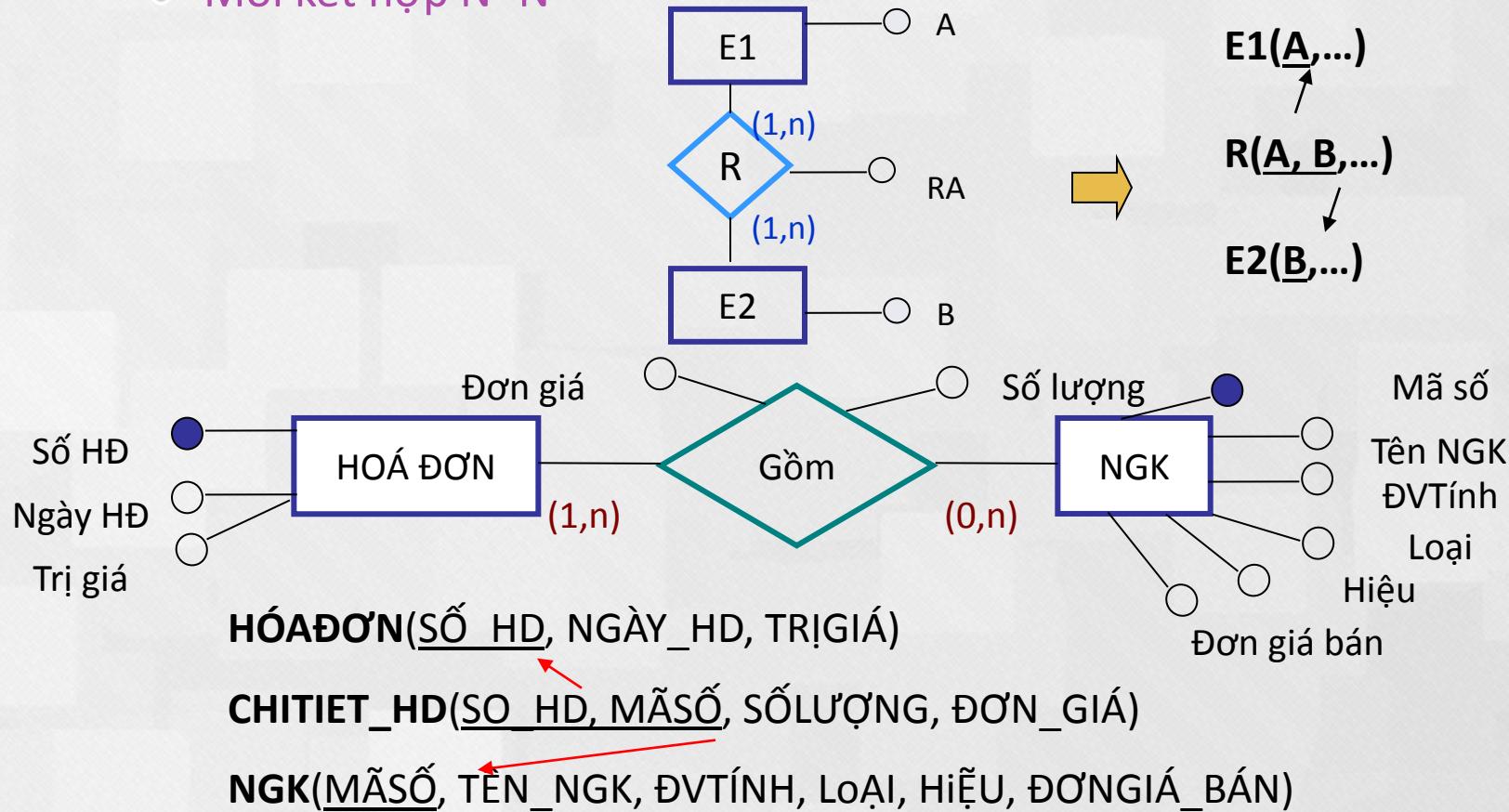


# Thiết kế luận lý dữ liệu cấp thấp

- Chuyển mô hình TTKH → mô hình Quan hệ

- Chuyển đổi mối kết hợp nhị phân

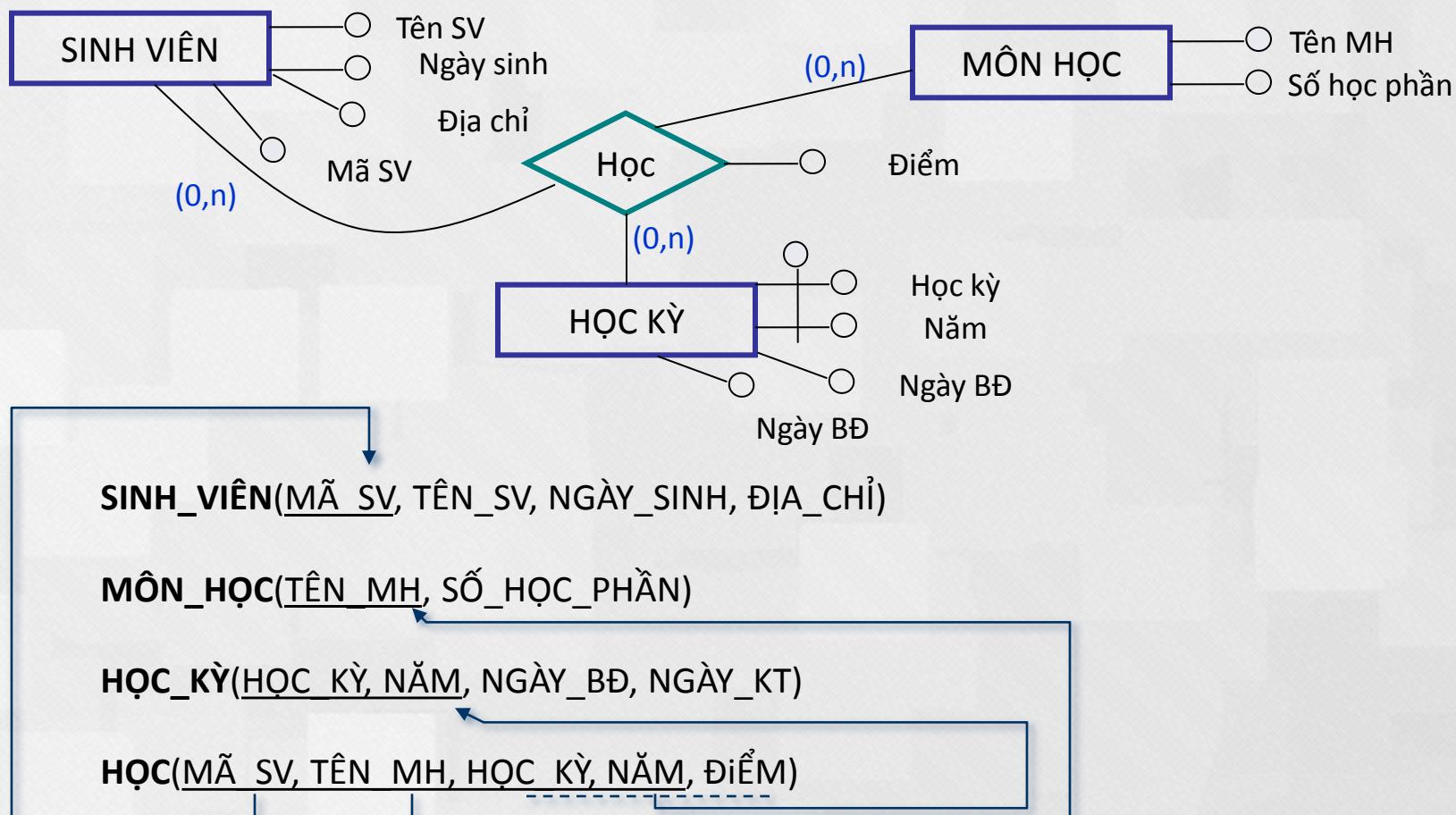
- Mối kết hợp N -N



# Thiết kế luận lý dữ liệu cấp thấp

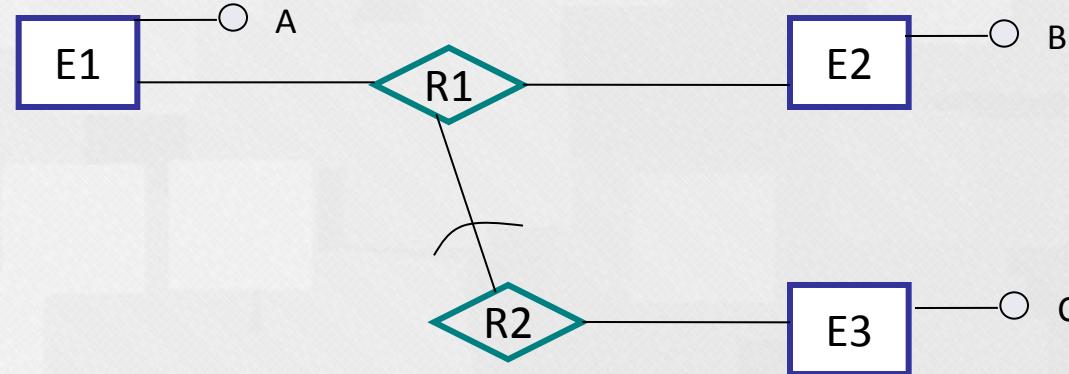
- Chuyển mô hình TTKH → mô hình Quan hệ

- Chuyển đổi mối kết hợp đa phân



# Thiết kế luận lý dữ liệu cấp thấp

- Chuyển đổi từ mô hình TTKH → mô hình Quan hệ
- Chuyển đổi mối kết hợp mở rộng

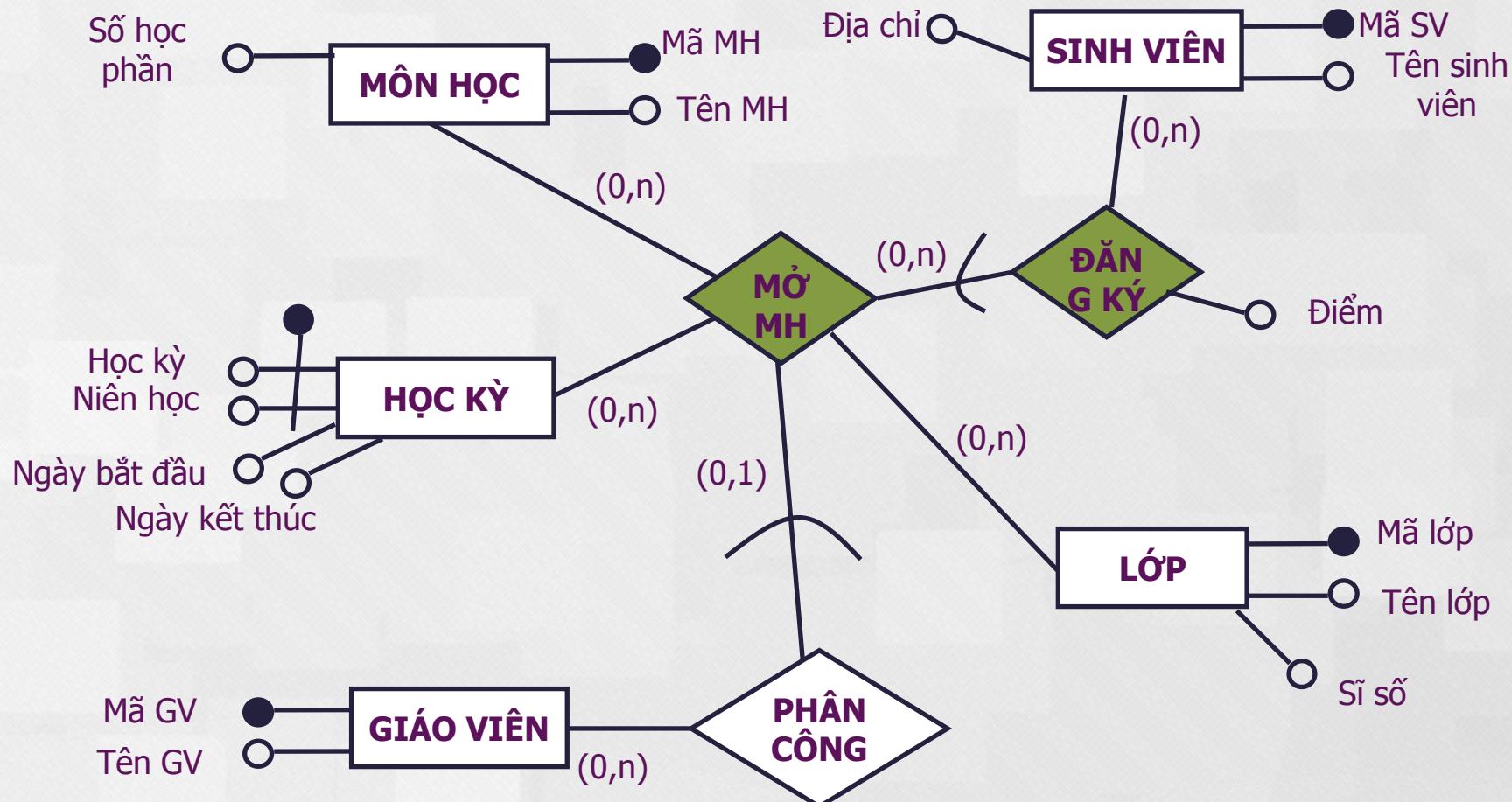


R1(A,B,...)  
↑  
R2(A,B,C,...)

# Thiết kế luận lý dữ liệu cấp thấp

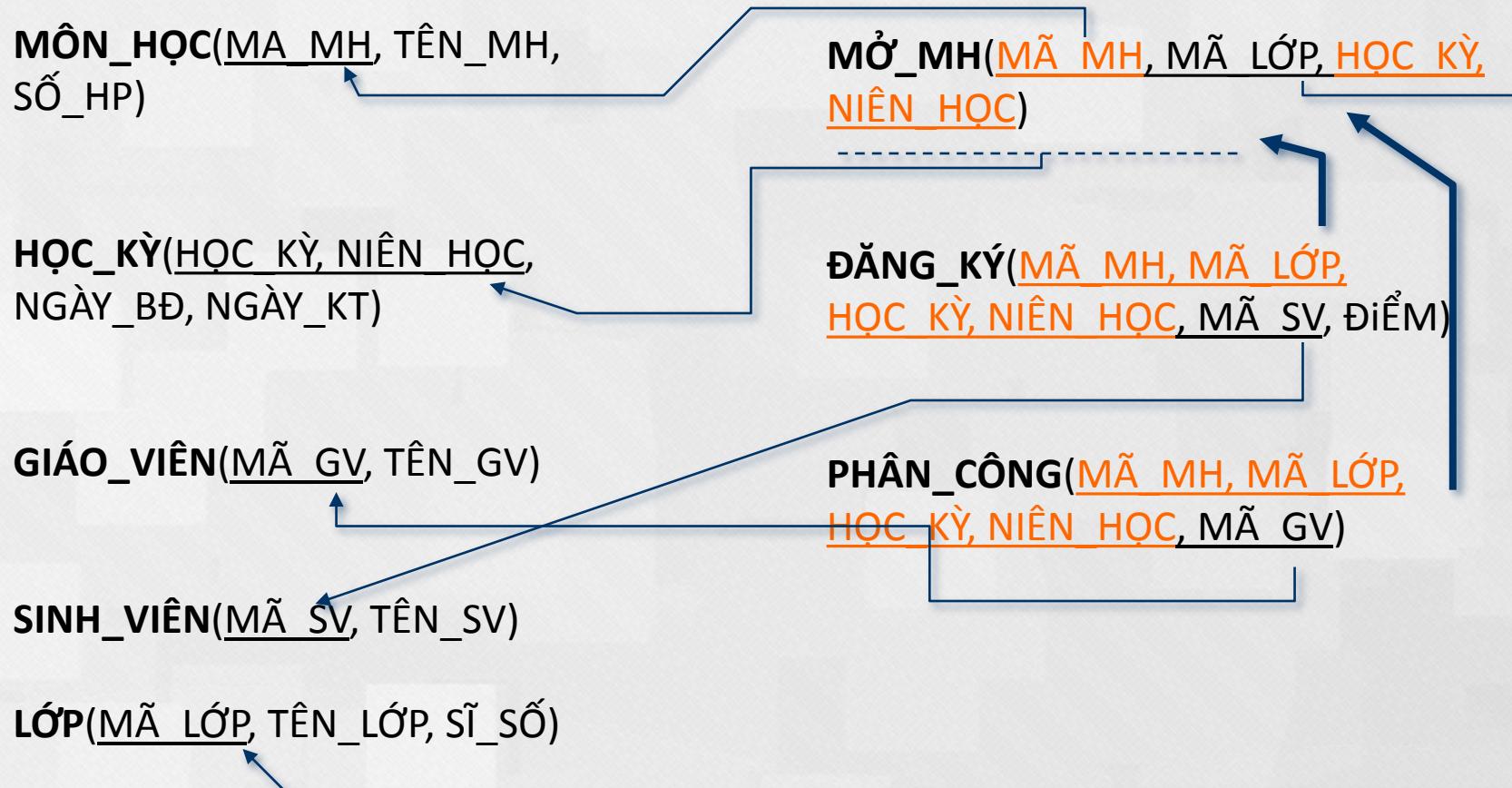
- Chuyển mô hình TTKH → mô hình Quan hệ

- Chuyển đổi mối kết hợp mở rộng – ví dụ:



# Thiết kế luận lý dữ liệu cấp thấp

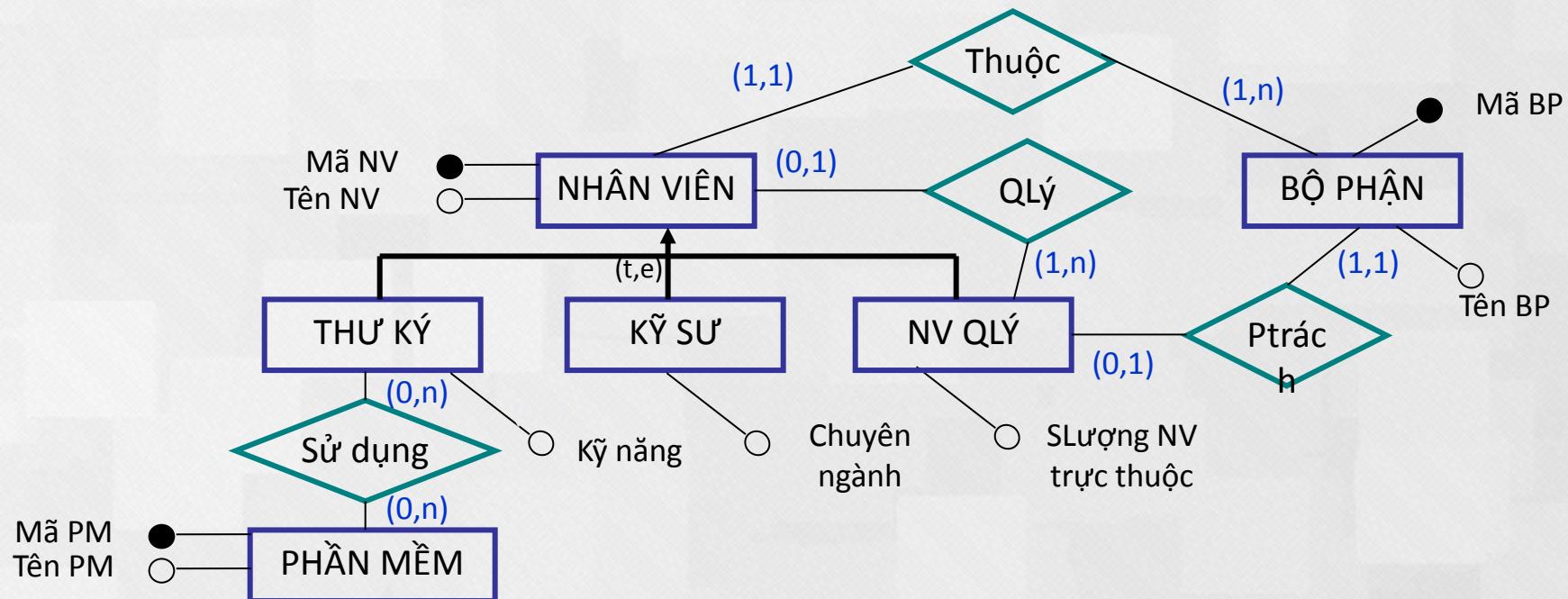
- Chuyển mô hình TTKH → mô hình Quan hệ
- Chuyển đổi mối kết hợp mở rộng – ví dụ:



# Thiết kế luận lý dữ liệu cấp thấp

- Chuyển mô hình TTKH → mô hình Quan hệ

- Ví dụ : SV tự thực hiện



# Nội dung trình bày

- **Thiết kế luận lý dữ liệu**
  - **Thiết kế luận lý cấp cao**
  - **Thiết kế luận lý cấp thấp**
- **Thiết kế mã**
- **Thiết kế vật lý dữ liệu**

## Thiết kế mã

- ⦿ Tìm các hình thức mã hóa như là: mã quận, mã bưu điện, mã xe, mã số sản phẩm, NVL, mã sv,...
- ⦿ Dễ dàng cho việc quản lý và xử lý thông tin trong máy tính

# Thiết kế mã

## ⦿ Mục tiêu:

- ⦿ **Dễ dàng hơn cho việc nhận dạng và phân loại dữ liệu**
- ⦿ **Mã hóa phải được thiết kế trên cơ sở uyển chuyển và có khả năng mở rộng**

## ⦿ Ngữ nghĩa của mã hóa: dựa trên 4 loại

- ⦿ **Duy nhất:** mã phải là duy nhất để nhận dạng đối tượng dữ liệu
  - ⦿ Ví dụ: các khách hàng có cùng tên có thể phân biệt thông qua mã khách hàng
- ⦿ **Phân loại:** phân loại đối tượng dữ liệu thành những nhóm khác nhau và dựa vào mã hóa, có thể nhận ra được nhóm của dữ liệu
  - ⦿ Ví dụ: mã hóa để phân loại chủng loại hàng hóa

# Thiết kế mã

- **Sắp xếp: mã hóa để thể hiện thứ tự của dữ liệu**
- **Kiểm tra: kiểm soát dữ liệu mã có được nhập đúng hay không**
- **Các đặc điểm cần xem xét trong thiết kế mã**
- **Vùng mã hóa: phạm vi mà mã được sử dụng**
  - Khi liên kết với các hệ thống bên ngoài, mã nên được dùng như là một chuẩn công nghiệp chung
  - Khi được dùng trong cùng đơn vị, mã nên có phạm vi toàn công ty

# Thiết kế mã

- Ví dụ: mã môn học được thiết kế khác nhau trong cùng một trường

TH201 Thiết kế Cơ Sở Dữ Liệu

Khoa CNTT

CNTT201 Thiết kế Cơ Sở Dữ Liệu

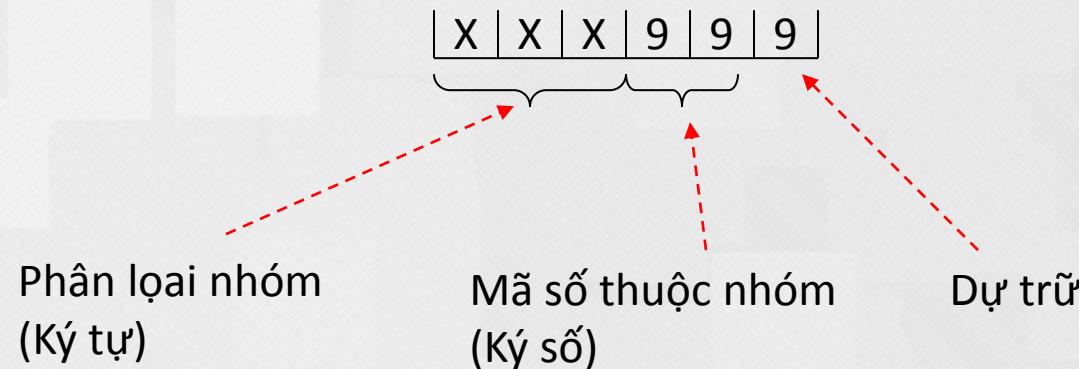
PĐTạo

Cùng một môn học nhưng cách đặt mã khác nhau giữa Khoa CNTT và PĐTạo → tạo ra các hoạt động không cần thiết như là: chuyển đổi mã liên quan đến các kết quả học tập trên môn học,....

# Thiết kế mã

## Chu kỳ sử dụng: ước lượng chu kỳ sử dụng mã và số lượng gia tăng trong tương lai

- Yêu cầu cần một lượng mã đủ trong chu kỳ
- thiết kế các mã dãy trự cho việc mở rộng và đặt ở cuối mã
- Ví dụ:

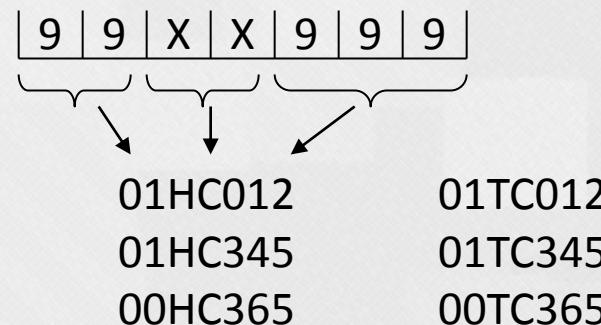


# Thiết kế mã

- Chu kỳ sử dụng:

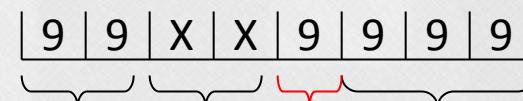
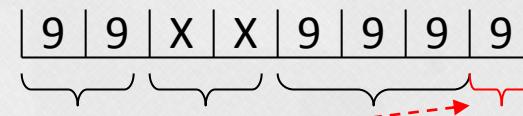
- Ví dụ: mã số sinh viên

## Không dữ trữ



## Có dữ trữ

Dự trữ



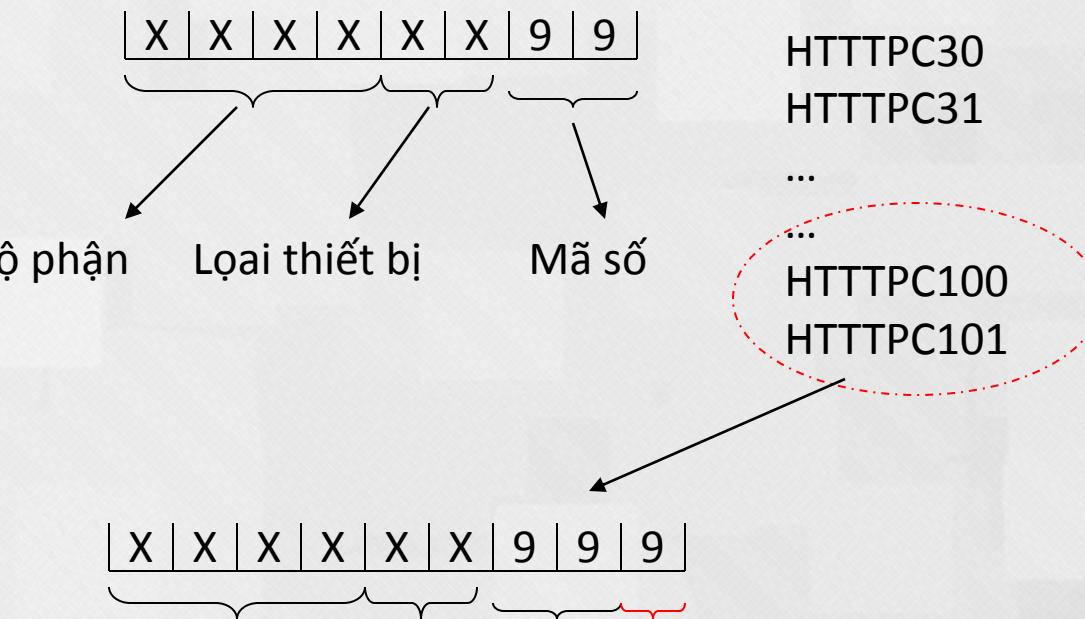
01HC0012  
01HC0345  
00HC0365

# Thiết kế mã

## Chu kỳ sử dụng:

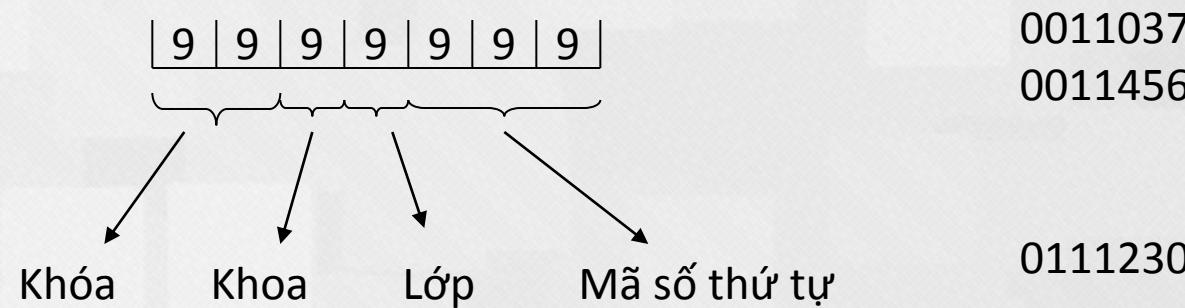
- Ví dụ: mã số thiết bị máy tính các bộ phận của trường

Không dữ trữ



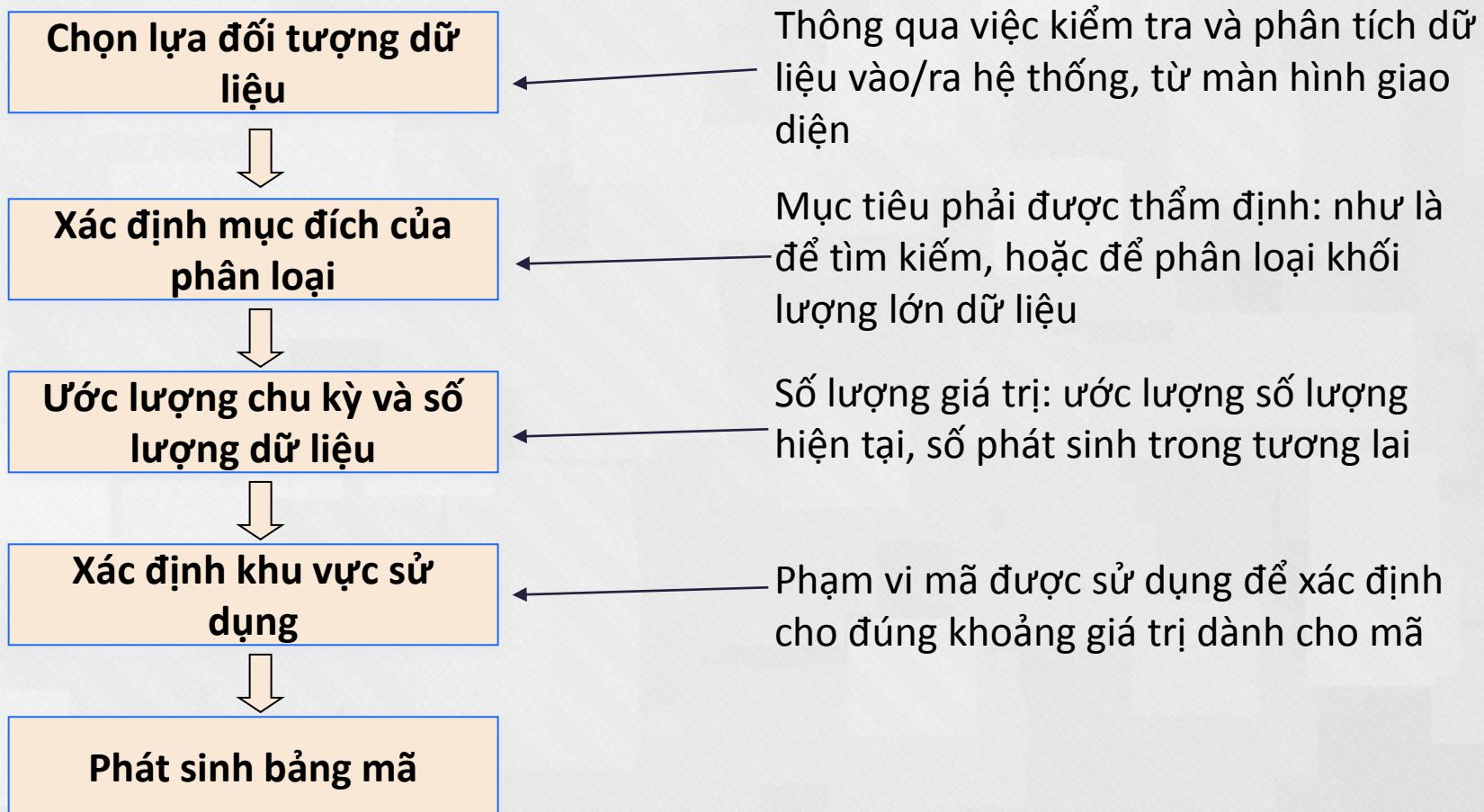
# Thiết kế mã

- Ngữ nghĩa: mã có thể hiểu được bởi các thành viên liên quan
- Ví dụ: mã số sinh viên



# Thiết kế mã

## Các bước trong thiết kế mã



# Thiết kế mã

## Mô tả các loại mã

Tên mã	Mô tả	Ví dụ
Mã tuần tự	Một số được gán một cách tuần tự	<u>Mã số xe của tỉnh:</u> 50 Thành phố HCM 60 Đồng Nai 62 Long An 63 Tiền Giang ...
Mã khối	Một số được gán cho mỗi khối. Rồi một số sẽ được gán tuần tự trong khối đó → thuận tiện cho phân loại dữ liệu	<u>Mã khách hàng của từng chi nhánh</u> <u>được qui định như sau:</u> 0001 khách hàng tại trung tâm 1000 khách hàng chi nhánh A 2000 khách hàng chi nhánh B 3000 khách hàng chi nhánh C ...

# Thiết kế mã

## Mô tả các loại mã

Tên mã	Mô tả	Ví dụ												
Mã thập phân	Các đối tượng sẽ được mã hóa từ 0 đến 9, rồi đến lượt thành viên của mỗi đối tượng này cũng sẽ được mã hóa từ 0 đến 9, ....	Mã hóa phòng ban trong một công ty: 00 Ban giám đốc 1 Bộ phận kinh doanh 1 Nhóm 1 2 Nhóm 2 2 Bộ phận phát triển 10 Chi nhánh A 20 Chi nhánh B 30 Chi nhánh C												
Mã theo ký số (phân loại nhóm)	Mỗi ký số của mã sẽ gán cho một ngữ nghĩa	Mã SV của trường ĐHKHTN TPHCM <table style="margin-left: auto; margin-right: auto;"> <tr> <td>①</td><td>②</td><td>③</td><td>④</td> </tr> <tr> <td>01</td><td>1</td><td>1</td><td>100</td> </tr> <tr> <td>01</td><td>2</td><td>1</td><td>120</td> </tr> </table> <p>① Năm ② Khoa ③ Lớp ④ số thứ tự</p>	①	②	③	④	01	1	1	100	01	2	1	120
①	②	③	④											
01	1	1	100											
01	2	1	120											

# Thiết kế mã

## Mô tả các loại mã

Tên mã	Mô tả	Ví dụ
Mã gợi nhớ	Chữ viết tắt hoặc biểu tượng của đối tượng được dùng để mã hóa	Mã hóa lớp học của trường đại học như sau: 00TC Lớp tại chức khóa 2000 01TC Lớp tại chức khóa 2001 00HC Lớp hoàn chỉnh khóa 2000 01HC Lớp hoàn chỉnh khóa 2001
Mã kiểm tra	Dùng một ký số kiểm tra thêm vào sau mỗi mã số	

# Chương 7 Thiết kế dữ liệu

- **Thiết kế luận lý dữ liệu**
  - **Thiết kế luận lý cấp cao**
  - **Thiết kế luận lý cấp thấp**
- **Thiết kế mã**
- **Thiết kế vật lý dữ liệu**

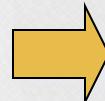
# Thiết kế dữ liệu vật lý

## Mục đích:

- **Chuyển các mô tả dữ liệu luận lý sang các đặc tả kỹ thuật nhằm lưu trữ và truy xuất dữ liệu**
- **Tạo một thiết kế cho việc lưu trữ dữ liệu nhằm cung cấp một hiệu năng phù hợp và đảm bảo tính toàn vẹn, an toàn và khả năng phục hồi của CSDL**

# Thiết kế dữ liệu vật lý

- Mô hình quan hệ
- Khối lượng dữ liệu
- Định nghĩa của thuộc tính
- Yêu cầu về thời gian trả lời
- Nhu cầu về an toàn dữ liệu
- Nhu cầu về backup/resotore
- Hệ quản trị CSDL sử dụng



- Kiểu dữ liệu thuộc tính
- Mô tả mẫu tin vật lý
- Tổ chức file
- Kiến trúc chỉ mục và CSDL
- Tối ưu hóa truy vấn

# Thiết kế dữ liệu vật lý

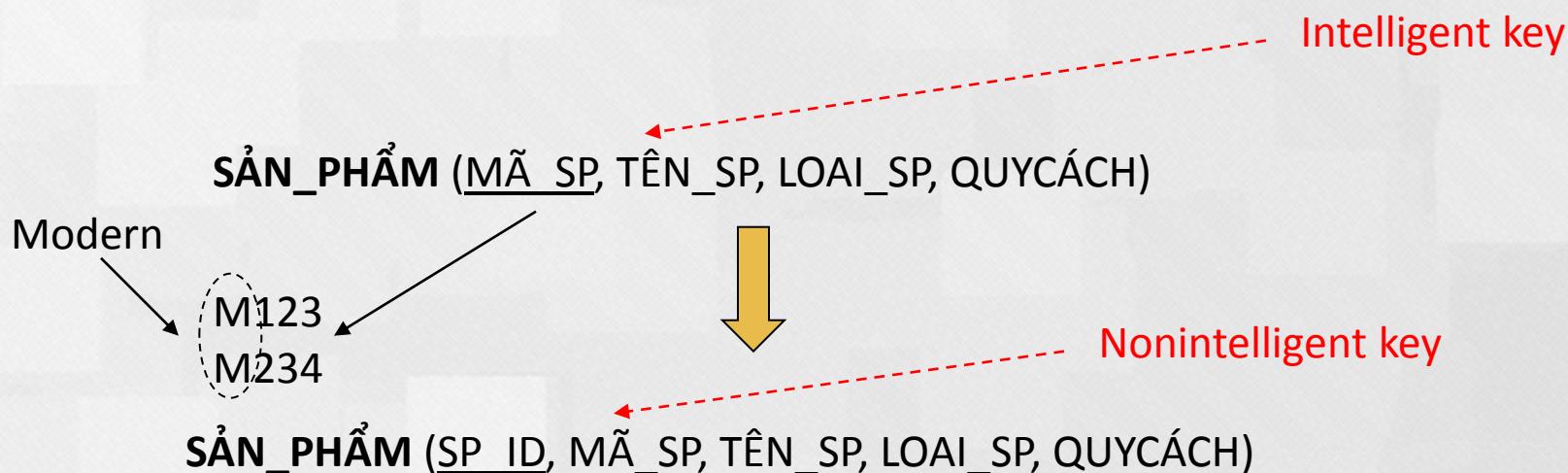
- **Thiết kế field**
- **Phân chia dữ liệu (partition)**
- **Gộp dữ liệu (demormalization)**
- **Thiết kế file dữ liệu vật lý**
- **Tổ chức file chỉ mục**
- **Clustering file**

# Thiết kế dữ liệu vật lý

- ➊ Thiết kế field: đơn vị dữ liệu nhỏ nhất
- ➋ Thiết kế field bao gồm:
  - ➌ Chọn kiểu dữ liệu: phải thỏa các tiêu chuẩn sau:
    - ➍ Tối thiểu không gian lưu trữ
    - ➎ Hiển thị tất cả tình huống giá trị
    - ➏ Cải tiến việc toàn vẹn dữ liệu
    - ➐ Hỗ trợ cho tất cả thao tác dữ liệu

# Thiết kế dữ liệu vật lý

- ➊ Thiết kế field – Chọn lựa khóa chính:
- ➋ Việc chọn lựa khóa chính trong mô hình luận lý đôi khi không thuận tiện trong cập nhật
  - **Giải pháp:** Thêm một thuộc tính khóa không có ý nghĩa thực tế (nonintelligent key) → hỗ trợ cho các phép toán cập nhật tốt hơn thuộc tính khóa trước đó
- ➌ Ví dụ:



# Thiết kế dữ liệu vật lý

- Thiết kế field bao gồm:

- Ví dụ:

**SANPHAM**

SP_ID	Mã_SP	Tên_SP	Loại_SP	Quycách
1	M123	Sản phẩm X	M	12x4
2	M234	Sản phẩm Y	M	15x3
3	S012	Sản phẩm M	S	12x2
4	L121	Sản phẩm N	L	18x6

**HÓAĐƠN**

HD_ID	Số_HD	Ngày_HD	SP_ID	SLượng	Đơn giá
1	001/HD	1/1/2004	1	10	200
2	002/HD	1/1/2004	1	5	200
3	003/HD	2/1/2004	2	120	120
4	004/HD	3/1/2004	4	200	700

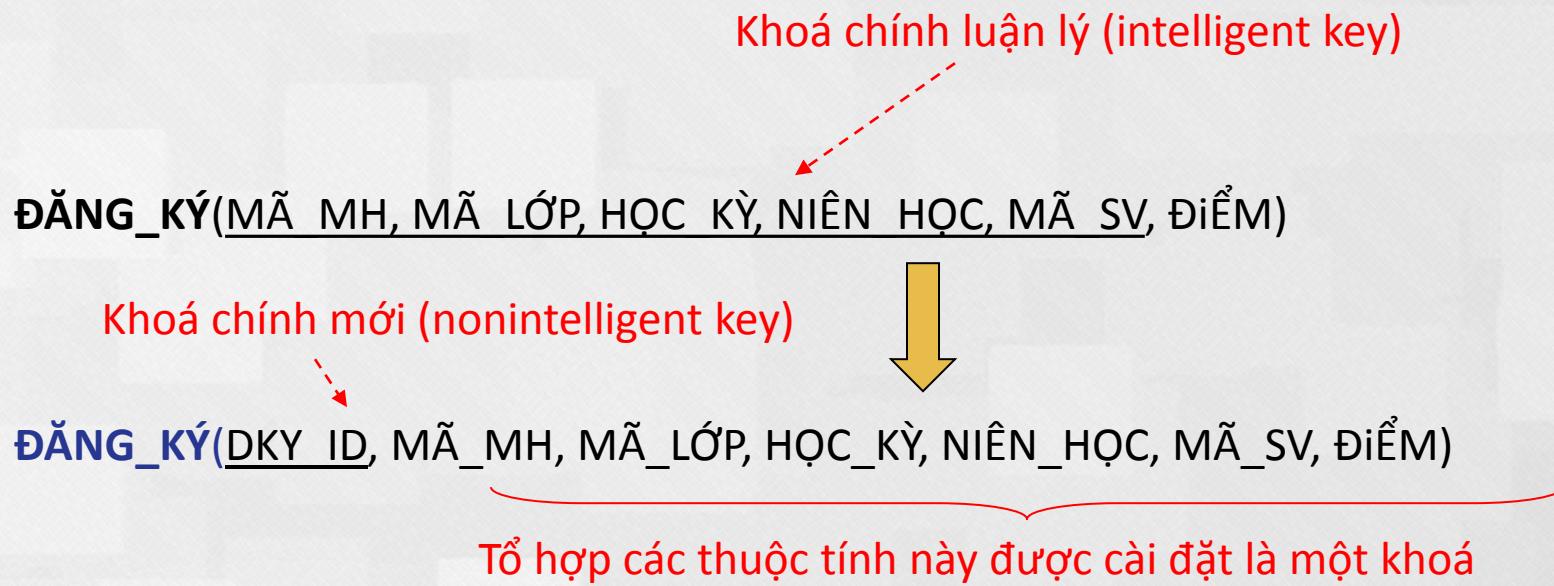
MÃ\_SP của table SANPHAM có thể bị thay đổi mà không ảnh hưởng đến table HÓAĐƠN có khóa ngoại tham chiếu đến SANPHAM

# Thiết kế dữ liệu vật lý

- Thiết kế field – Chọn lựa khoá chính:

- Khoá chính phức tạp sẽ làm giảm tốc độ truy cập CSDL trong máy tính → chọn lựa khoá chính thay thế (nonintelligent key) đơn giản và hiệu quả hơn

- Ví dụ:



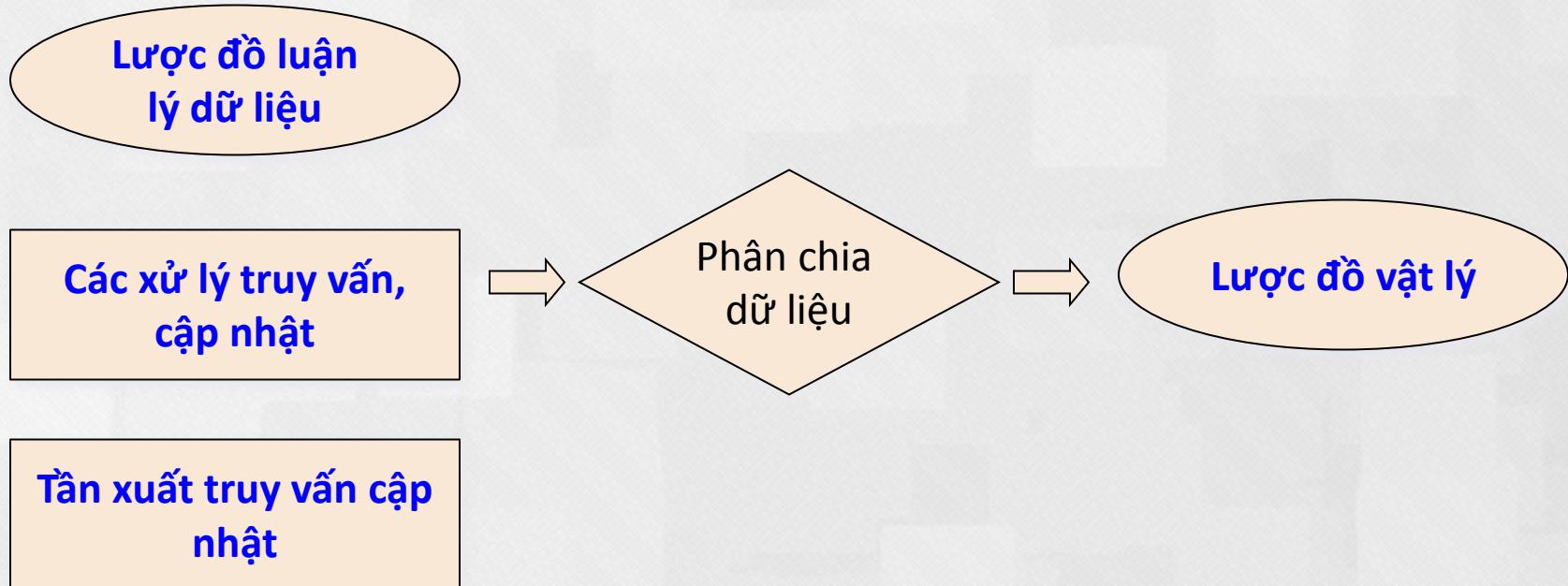
# Thiết kế dữ liệu vật lý

## Phân chia dữ liệu (partition)

- Phân chia theo chiều ngang (horizontal partition): phân chia các dòng trong một table thành nhiều table khác nhau
  - Tình huống áp dụng: khi nhiều người dùng khác nhau cần truy cập các dòng dữ liệu khác nhau
  - Ưu điểm:
    - Tối ưu hóa tốc độ truy cập dữ liệu
  - Nhược điểm
    - Phức tạp khi phải truy cập toàn bộ dữ liệu

# Thiết kế dữ liệu vật lý

## Phân chia dữ liệu (partition)



# Thiết kế dữ liệu vật lý

## Phân chia dữ liệu (partition)

### Ví dụ:

HOA\_DON

KL: ~10.000.000/năm

Số_HD	Ngày_HD	Diễn_giải	Trị giá
Hd00001	1/1/04	Xxxxxx	1.000.000
Hd00002	2/1/04	Yyyyyy	2.000.000
....			
Hd15000	1/1/05	Zxxxxxxxxx	1.400.000
Hd15001	2/1/05	Qqqqqqqqq	2.100.000
...			
Hd30000	2/1/06	Asasasa	12.000.000
Hd30001	2/1/06	Dsdssds	1.000.000

### Các xử lý truy cập dữ liệu

Mã số	Tên xử lý	Tần suất
O1	Tìm hóa đơn	100/ngày
O2	Tính doanh thu tháng	1/tháng
O3	Tính doanh thu theo khách hàng	100/tháng
O4	Tổng hợp doanh số năm	1/năm
O5	Lập biểu đồ so sánh doanh số theo các năm	1/năm

# Thiết kế dữ liệu vật lý

## Phân chia dữ liệu (partition)

### Ví dụ:

Tách table HOA\_DON theo năm

Các phép toán o1, o2, o4 hiệu quả hơn

**HD004**

Số_HD	Ngày_HD	Diễn_giải	Trị giá
Hd00001	1/1/04	Xxxxxx	1.000.000
Hd00002	2/1/04	Yyyyyyy	2.000.000

**HOA\_DON**

KL: ~10.000.000/năm

Số_HD	Ngày_HD	Diễn_giải	Trị giá
Hd00001	1/1/04	Xxxxxx	1.000.000
Hd00002	2/1/04	Yyyyyyy	2.000.000
....			
Hd15000	1/1/05	Zxxxxxxxxx	1.400.000
Hd15001	2/1/05	Qqqqqqqqqq	2.100.000
...			
Hd30000	2/1/06	Asasasas	12.000.000
Hd30001	2/1/06	Dsdssdsds	1.000.000

**HD005**

Số_HD	Ngày_HD	Diễn_giải	Trị giá
Hd15000	1/1/05	Zxxxxxxxxx	1.400.000
Hd15001	2/1/05	Qqqqqqqqqq	2.100.000

**HD006**

Số_HD	Ngày_HD	Diễn_giải	Trị giá
Hd30000	2/1/06	Asasasas	12.000.000
Hd30001	2/1/06	Dsdssdsds	1.000.000

# Thiết kế dữ liệu vật lý

## Phân chia dữ liệu (partition)

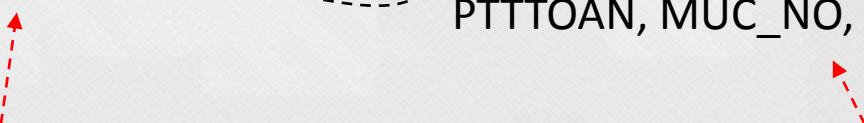
### Phân chia theo chiều dọc (vertical partition):

- Phân chia một cấu trúc luận lý thành những cấu trúc lưu trữ vật lý khác nhau
- Ví dụ:

**KHÁCH\_HANG(MÃ\_KH, TÊN\_KH, DCHI, DT, TAIKHOAN, MASOTHUE, PTTTOAN, MUC\_NO, CONG\_NO)**

**KH1(MA\_KH, TEN\_KH, DCHI)**

**KH2(DT, TAIKHOAN, MASOTHUE, PTTTOAN, MUC\_NO, CONG\_NO)**



Cấu trúc truy cập thường xuyên

Cấu trúc truy cập không thường xuyên

# Thiết kế dữ liệu vật lý

## ⦿ Gộp dữ liệu (denormalization)

### ⦿ Mục tiêu:

- ⦿ Tối ưu hóa truy vấn dữ liệu

### ⦿ Hạn chế:

- ⦿ Phát sinh trùng lắp dữ liệu
- ⦿ Kiểm soát tính nhất quán dữ liệu

# Thiết kế dữ liệu vật lý

- ❖ Gộp dữ liệu (denormalization)

- ❖ Gộp 2 quan hệ liên kết 1-1

**SINH\_VIÊN(MÃ\_SV, TEN\_SV, CHUYEN\_NGANH)**

**HỒSƠ\_HBỐNG(MÃ\_HS, NGÀY\_HS, KHẢ\_NĂNG, MA\_SV)**

Xử lý	Dữ liệu liên quan
O1	MÃ_SV, TEN_SV, CHUYEN_NGANH, NGÀY_SINH, KHẢ_NĂNG
O2	TEN_SV, NGÀY_HS, KHẢ_NĂNG



**SINH\_VIÊN(MÃ\_SV, TEN\_SV, CHUYEN\_NGANH, NGÀY\_HS, KHẢ\_NĂNG)**

# Thiết kế dữ liệu vật lý

- Gộp dữ liệu (denormalization)

- Gộp 2 quan hệ liên kết 1-N

Mã_SV	Tên_SV	...	MA_CN
991100	N.V.A	...	HTTT
991101	N.T.B	...	CNPM
991112	P.V.C	...	CNPM
991120	T.T.N	...	HTTT
991200	T.V.M	...	CNPM

Mã_CN	Tên_CN
HTTT	Hệ thống thông tin
CNPM	Công nghệ phần mềm
MMT	Mạng máy tính
CNTT	Công nghệ tri thức

Truy vấn thường xuyên:

- Q1 (Mã\_SV, TÊN\_SV, TÊN\_CN)

# Thiết kế dữ liệu vật lý

## Gộp dữ liệu (denormalization)

### Gộp 2 quan hệ liên kết 1-N

Mã_SV	Tên_SV	...	MA_CN	Tên_CN
991100	N.V.A	...	HTTT	Hệ thống thông tin
991101	N.T.B	...	CNPM	Công nghệ phần mềm
991112	P.V.C	...	CNPM	Công nghệ phần mềm
991120	T.T.N	...	HTTT	Hệ thống thông tin
991200	T.V.M	...	CNPM	Công nghệ phần mềm

Truy vấn thường xuyên:

- Q1 (Mã\_SV, TÊN\_SV, TÊN\_CN)

Trùng lắp thông tin

Cấu trúc gộp trên sẽ tối ưu hơn cho truy vấn Q1, nhưng sẽ dẫn đến trùng lắp thông tin

# Thiết kế dữ liệu vật lý

## • Tổ chức file chỉ mục (index)

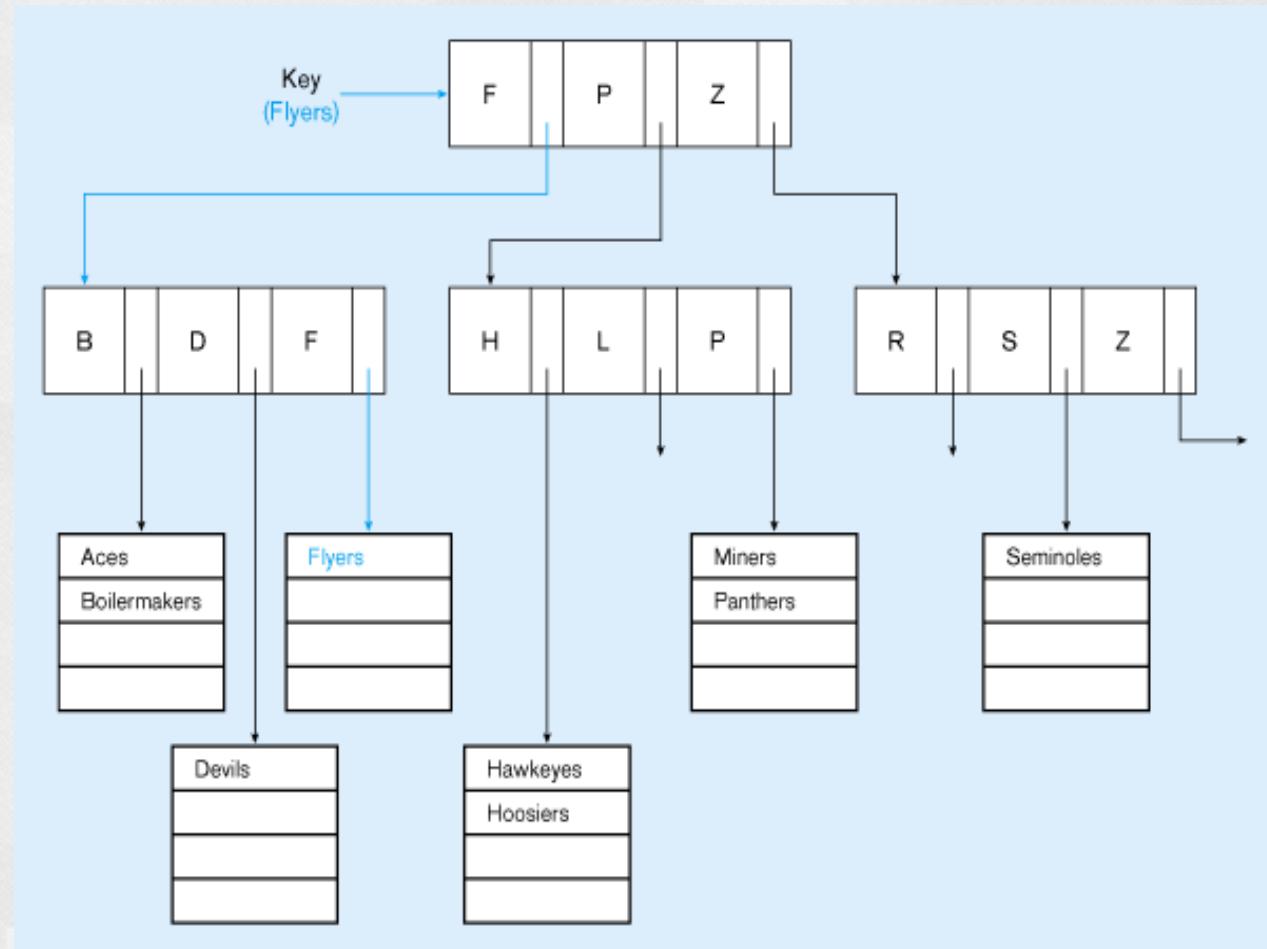
### • Các yếu tố nên sử dụng chỉ mục

- Cấu trúc table lớn
- Đánh chỉ mục khóa chính mỗi table
- Đánh chỉ mục các column tìm kiếm (xuất hiện thường xuyên trong mệnh đề WHERE)
- Các column trong ORDER BY, GROUP BY
- Sử dụng nhiều chỉ mục cho các CSDL ít biến đổi; hạn chế sử dụng trong các CSDL hay biến đổi → bởi vì các thay đổi (thêm, xóa, sửa) sẽ bị chậm đi do phải cập nhật lại chỉ mục

# Thiết kế dữ liệu vật lý

- Các file chỉ mục (index)

- B\_Tree



Độ phức tạp tỉ lệ với chiều cao của B-Tree

# Thiết kế dữ liệu vật lý

- Các file chỉ mục (index)

- Join index

Customer

RowID	Cust#	CustName	City	State
10001	C2027	Hadley	Dayton	Ohio
10002	C1026	Baines	Columbus	Ohio
10003	C0042	Ruskin	Columbus	Ohio
10004	C3861	Davies	Toledo	Ohio
...				

Store

RowID	Store#	City	Size	Manager
20001	S4266	Dayton	K2	E2166
20002	S2654	Columbus	K3	E0245
20003	S3789	Dayton	K4	E3330
20004	S1941	Toledo	K1	E0874
...				

Join Index

CustRowID	StoreRowID	Common Value*
10001	20001	Dayton
10001	20003	Dayton
10002	20002	Columbus
10003	20002	Columbus
10004	20004	Toledo
...		

# Thiết kế dữ liệu vật lý

- Các file chỉ mục (index)

- Join index

Order

RowID	Order#	Order Date	Cust#(FK)
30001	O5532	10/01/2001	C3861
30002	O3478	10/01/2001	C1062
30003	O8734	10/02/2001	C1062
30004	O9845	10/02/2001	C2027
...			

Customer

RowID	Cust#(PK)	CustName	City	State
10001	C2027	Hadley	Dayton	Ohio
10002	C1026	Baines	Columbus	Ohio
10003	C0042	Ruskin	Columbus	Ohio
10004	C3861	Davies	Toledo	Ohio
...				

Join Index

CustRowID	OrderRowID	Cust#
10001	30004	C2027
10002	30002	C1062
10002	30003	C1062
10004	30001	C3861
...		