
PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN

Nội dung

Phần I: Tổng quan

Chương 1 – Tổng quan về HTTT

Chương 2 – Mô hình và phương pháp mô hình hóa HTTT

Phần II: Phân tích

Chương 3 – Xác định yêu cầu và chọn lựa phương án

Chương 4 – Mô hình hóa dữ liệu

Chương 5 – Mô hình hóa xử lý

Phần III: Thiết kế

Chương 6 – Thiết kế dữ liệu

Chương 7 – Thiết kế hệ thống

Chương 8 – Thiết kế giao diện

Chương 6 – Thiết kế dữ liệu

1. Thiết kế luận lý dữ liệu

1.1. Thiết kế luận lý cấp cao

1.2. Thiết kế luận lý cấp thấp

2. Thiết kế mã

3. Thiết kế vật lý dữ liệu

1. Thiết kế luận lý dữ liệu

1.1. Thiết kế luận lý cấp cao:

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

1.2. Thiết kế luận lý cấp thấp:

Chuyển đổi lược đồ kết quả của bước 1 sang một mô hình dữ liệu nhất định.

1.1. Thiết kế luận lý dữ liệu cấp cao

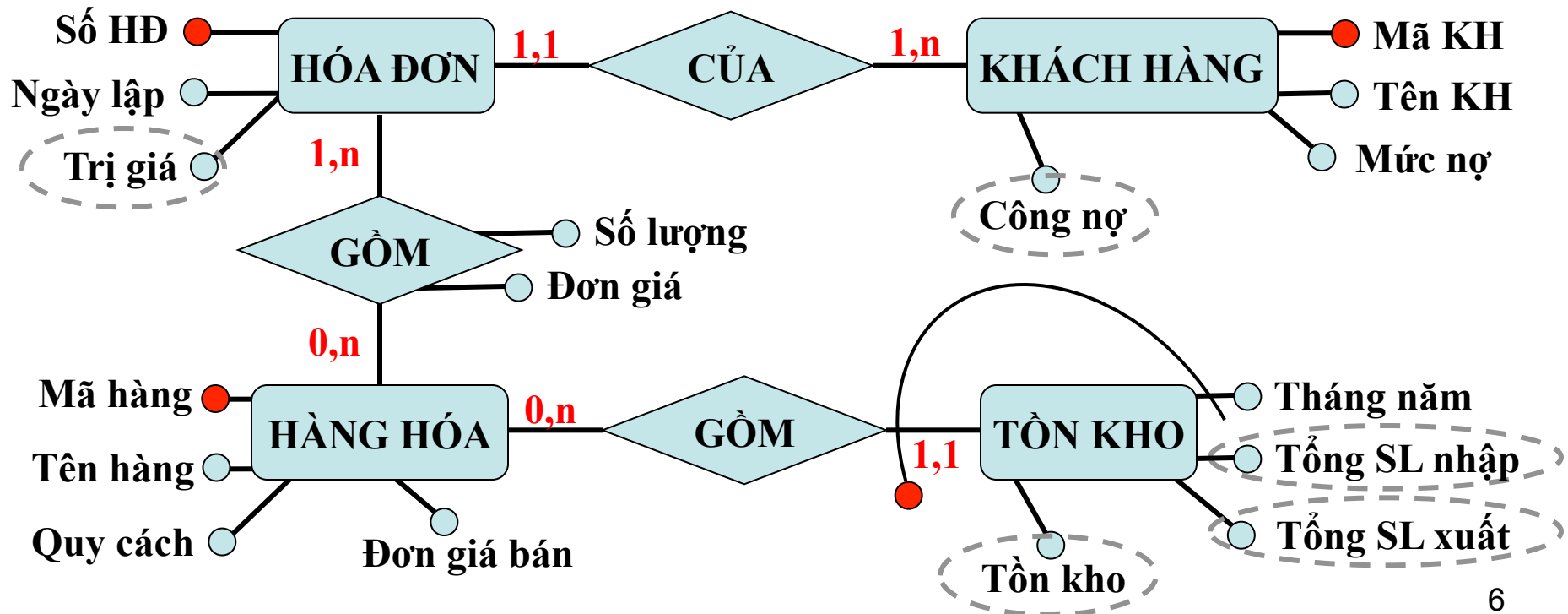
1. Quyết định về dữ liệu suy diễn
2. Chuyển đổi tổng quát hóa

1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

1. Quyết định về dữ liệu suy diễn:

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

Ví dụ:



1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

1. Quyết định về dữ liệu suy diễn (tt)

Ưu điểm:

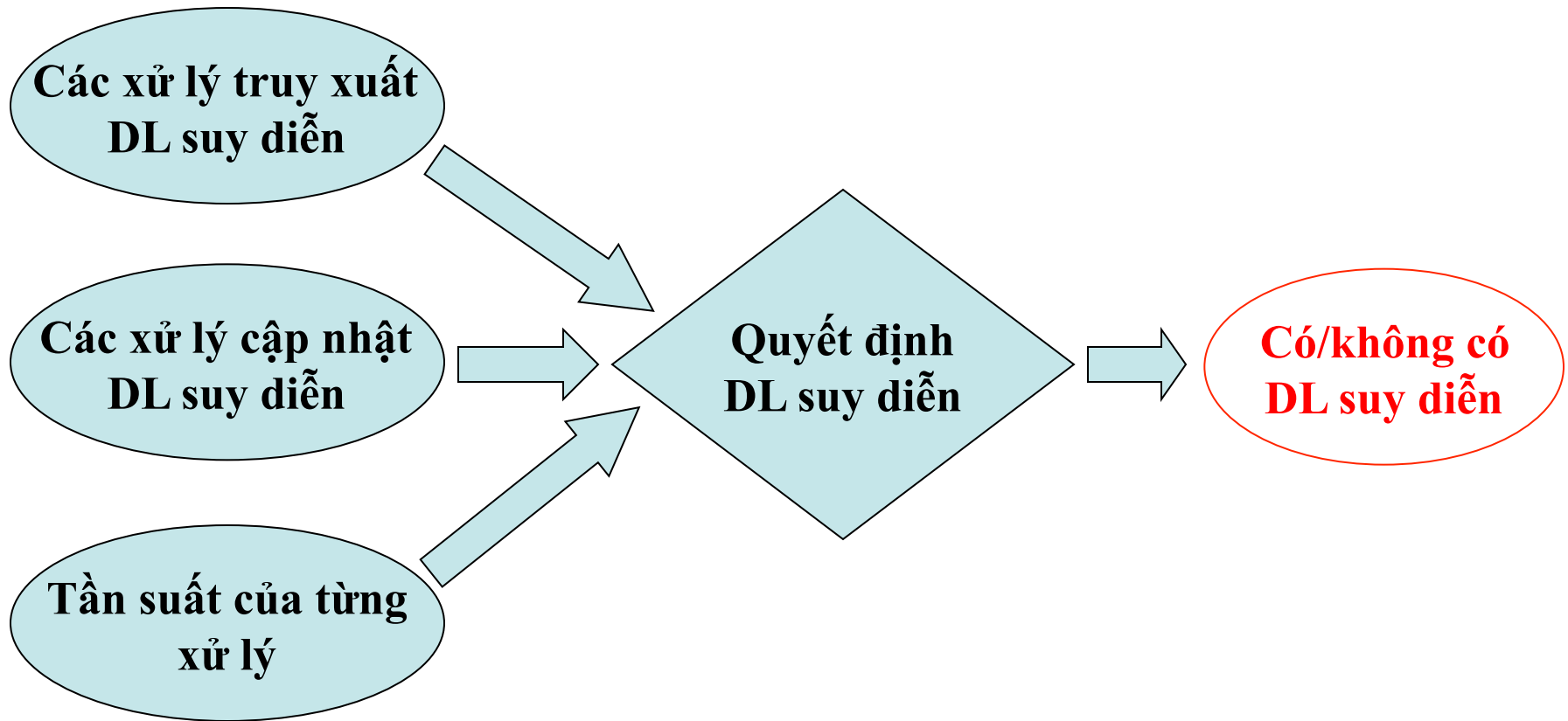
Tăng tốc độ truy xuất do không phải tính toán lại giá trị các thuộc tính này tại thời điểm thực hiện truy vấn.

Khuyết điểm:

- Tăng dung lượng lưu trữ;
- Chi phí tính toán và cập nhật (khi có thay đổi liên quan).

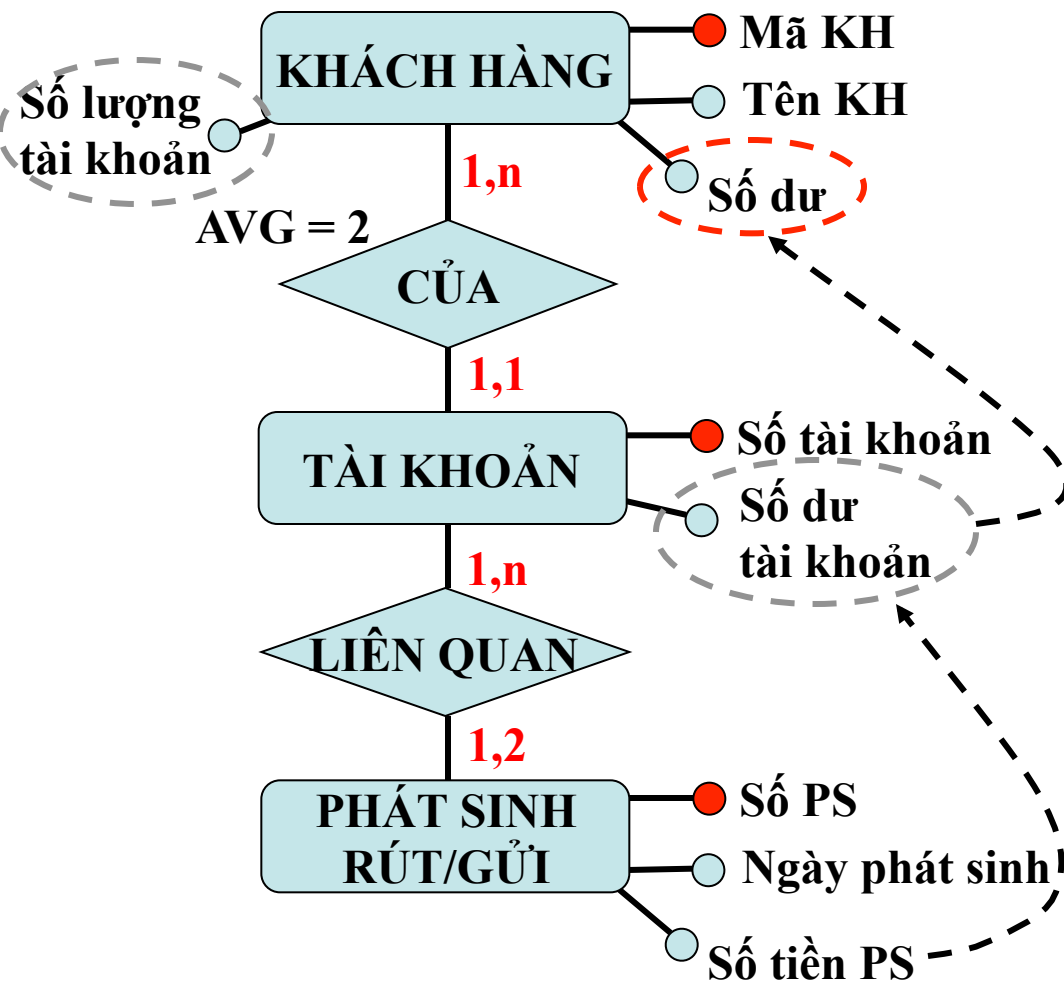
1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

1. Quyết định về dữ liệu suy diễn (tt)



1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

1. 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	Mối KH	30.000
LIÊN QUAN	Mối KH	800.000

1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

- Trường hợp có dữ liệu suy diễn “Số dư” (a)

Tên tác vụ	Khái niệm	Loại	Độc Ghi	Tần suất (/ngày)/ Bản số t.bình
0 ₁ : Mở tài khoản	TÀI KHOẢN	Thực thể	Ghi	100
	KHÁCH HÀNG	Thực thể	Ghi	100
	CỦA	Mối KH	Ghi	100
0 ₂ : Đọc cân số khách hàng	KHÁCH HÀNG	Thực thể	Độc	3000
0 ₄ : Rút tiền	TÀI KHOẢN	Thực thể	Độc	2000
			Ghi	2000
	KHÁCH HÀNG	Thực thể	Độc	2000
			Ghi	2000
0 ₅ : Gởi tiền	TÀI KHOẢN	Thực thể	Độc	1000
			Ghi	1000
	KHÁCH HÀNG	Thực thể	Độc	1000
			Ghi	1000

1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

- Trường hợp không có dữ liệu suy diễn **“Số dư”** (b)

Tên tác vụ	Khái niệm	Loại	Độc Ghi	Tần suất (/ngày)/ Bản số t.bình
0 ₁ : Mở tài khoản	TÀI KHOẢN	Thực thể	Ghi	100
	KHÁCH HÀNG	Thực thể	Ghi	100
	CỦA	Mỗi KH	Ghi	100
0 ₂ : Đọc cân số khách hàng	KHÁCH HÀNG	Thực thể	Độc	3000
	TÀI KHOẢN	Thực thể	Độc	3000 x 2 = 6000
	CỦA	Mỗi KH	Độc	3000 x 2 = 6000
0 ₄ : Rút tiền	TÀI KHOẢN	Thực thể	Độc	2000
			Ghi	2000
0 ₅ : Gởi tiền	TÀI KHOẢN	Thực thể	Độc	1000
			Ghi	1000

(a): 3000 Đ + 3000 G

(b): 12000 Đ

1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

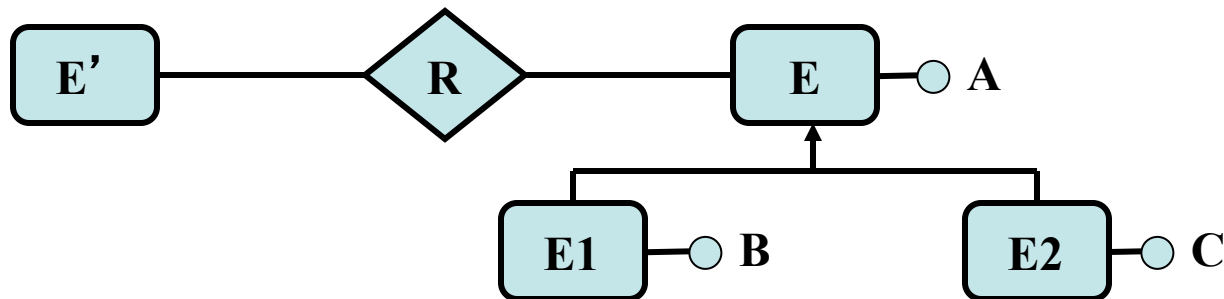
1. Quyết định về dữ liệu suy diễn (tt)

- Nếu $(a) \gg (b) \rightarrow$ chọn không có thuộc tính “Số dư”.
- Nếu $(a) \ll (b) \rightarrow$ chọn có thuộc tính “Số dư”.
- Tuy nhiên, kết quả định lượng chỉ là một trong các cơ sở để quyết định có/không có dữ liệu suy diễn.

1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

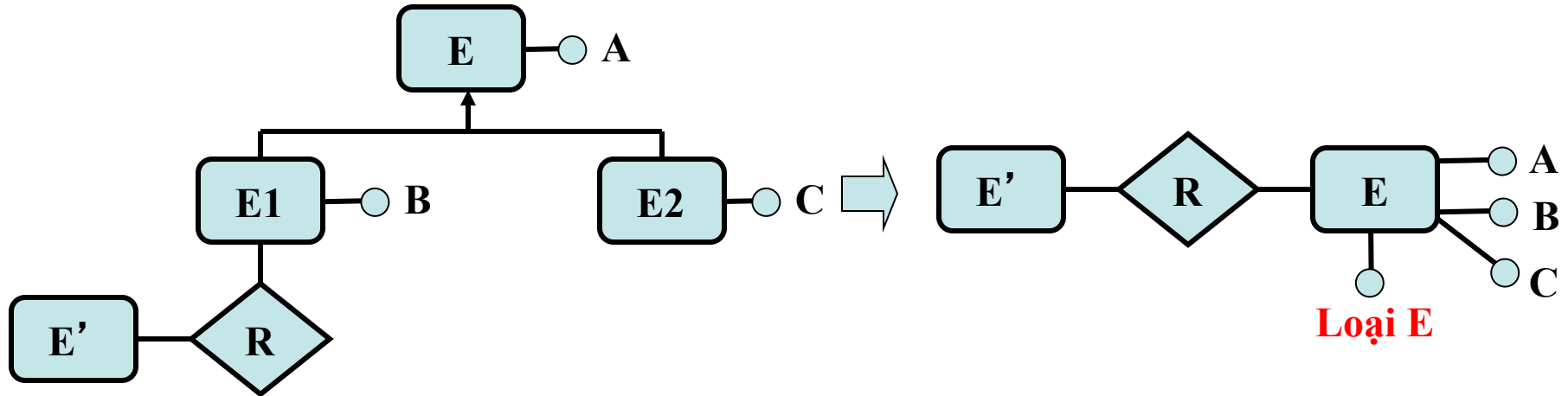
2. Chuyển đổi tổng quát hóa:

- Cần thiết khi mô hình cài đặt không hỗ trợ tổng quát hóa (mô hình quan hệ, mô hình mạng, ...).
- Có 3 phương án chọn lựa:
 - Dùng thực thể tổng quát (E)
 - Dùng thực thể chuyên biệt (E1,E2)
 - Dùng mối kết hợp (R)



1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

2. Chuyển đổi tổng quát hóa - Dùng thực thể tổng quát:



- Đặc điểm:

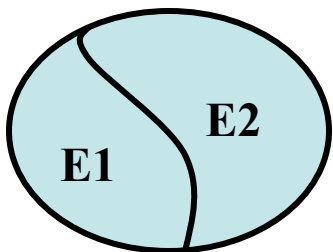
- Thuộc tính và mối kết hợp của các thực thể chuyên biệt sẽ trở thành thuộc tính và mối kết hợp của thực thể tổng quát.
- Loại bỏ các thực thể chuyên biệt.
- Thêm vào thực thể tổng quát một thuộc tính phân loại.

1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

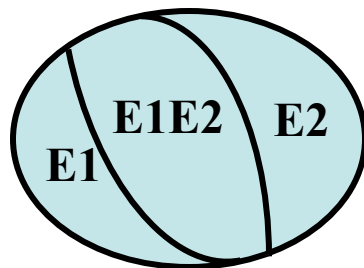
2. Chuyển đổi tổng quát hóa – Dùng thực thể tổng quát (tt)

- Đặc điểm (tt)

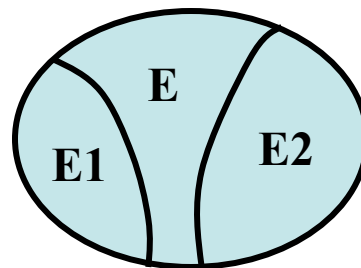
- $MGT(\text{Loại } E) \cong \{E, E1, E2, E1E2\} \cong \{0, 1, 2, 3\}$
- Biểu diễn sự tương quan:
 - $(t, e): \rightarrow$ Ràng buộc $MGT(\text{Loại } E) \cong \{E1, E2\}$
 - $(t, o): \rightarrow$ Ràng buộc $MGT(\text{Loại } E) \cong \{E1, E2, E1E2\}$
 - $(p, e): \rightarrow$ Ràng buộc $MGT(\text{Loại } E) \cong \{E, E1, E2\}$
 - $(p, o): \rightarrow$ Ràng buộc $MGT(\text{Loại } E) \cong \{E, E1, E2, E1E2\}$



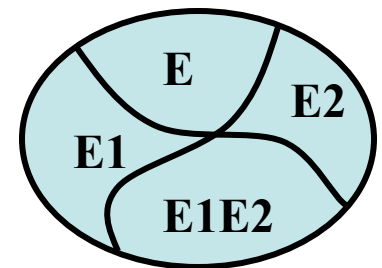
(t,e)



(t,o)



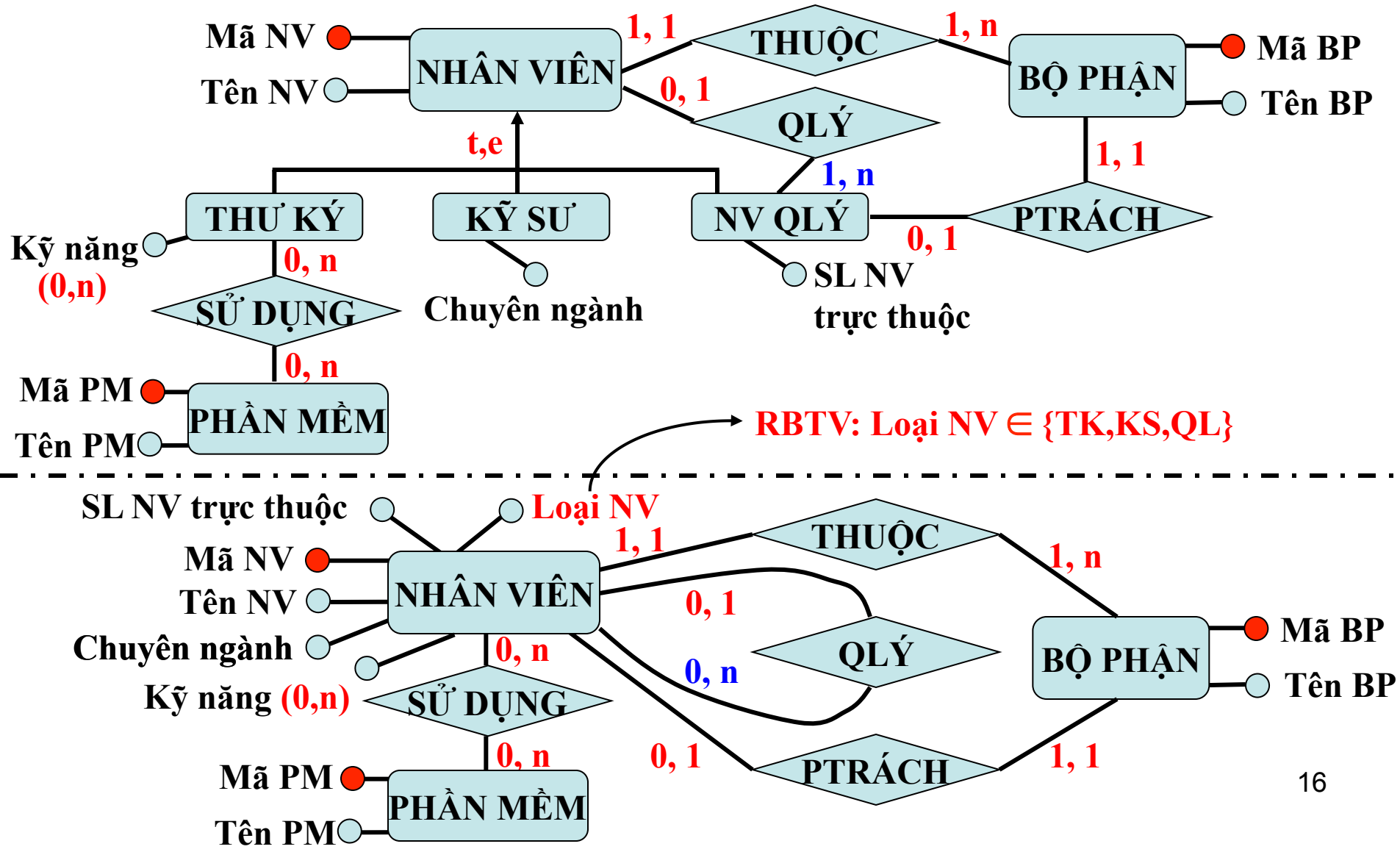
(p,e)



(p,o)

1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

2. Chuyển đổi tổng quát hóa – Dùng thực thể tổng quát – Ví dụ:



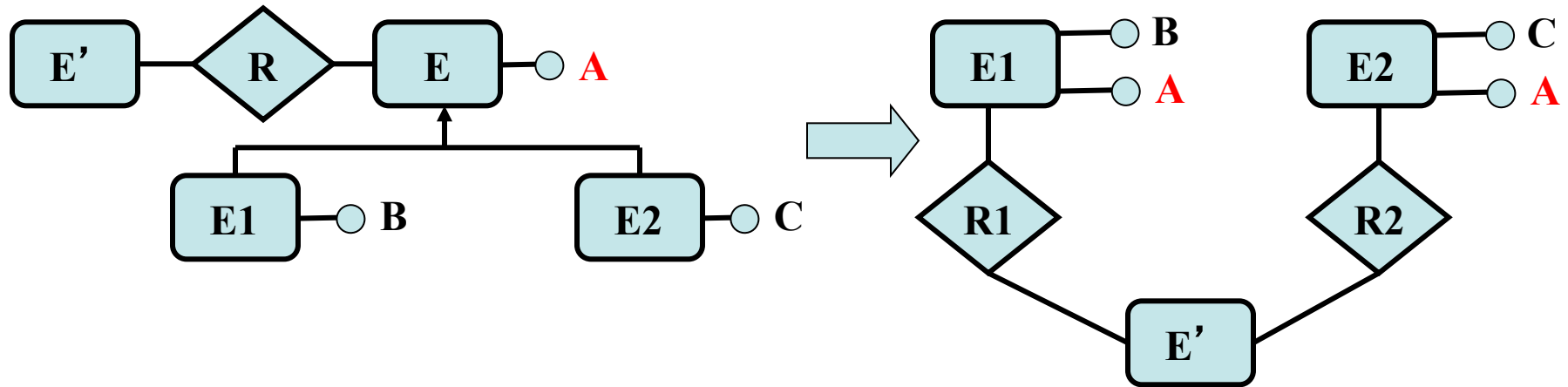
1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

2. Chuyển đổi tổng quát hóa - Dùng thực thể tổng quát (tt)

Ưu điểm	Khuyết điểm
Không phát sinh thêm các mối kết hợp.	Phát sinh một số giá trị rỗng tại các thuộc tính mà chỉ dùng cho một loại thực thể chuyên biệt mà thôi.
Áp dụng cho tất cả cấu trúc tổng quát hóa (t,p,e,o).	Các tác vụ muốn truy cập đến một loại thực thể chuyên biệt phải truy cập toàn bộ thực thể tổng quát.
	Phát sinh thêm một số RBTV cần phải kiểm tra.

1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

2. Chuyển đổi tổng quát hóa - Dùng thực thể chuyên biệt:



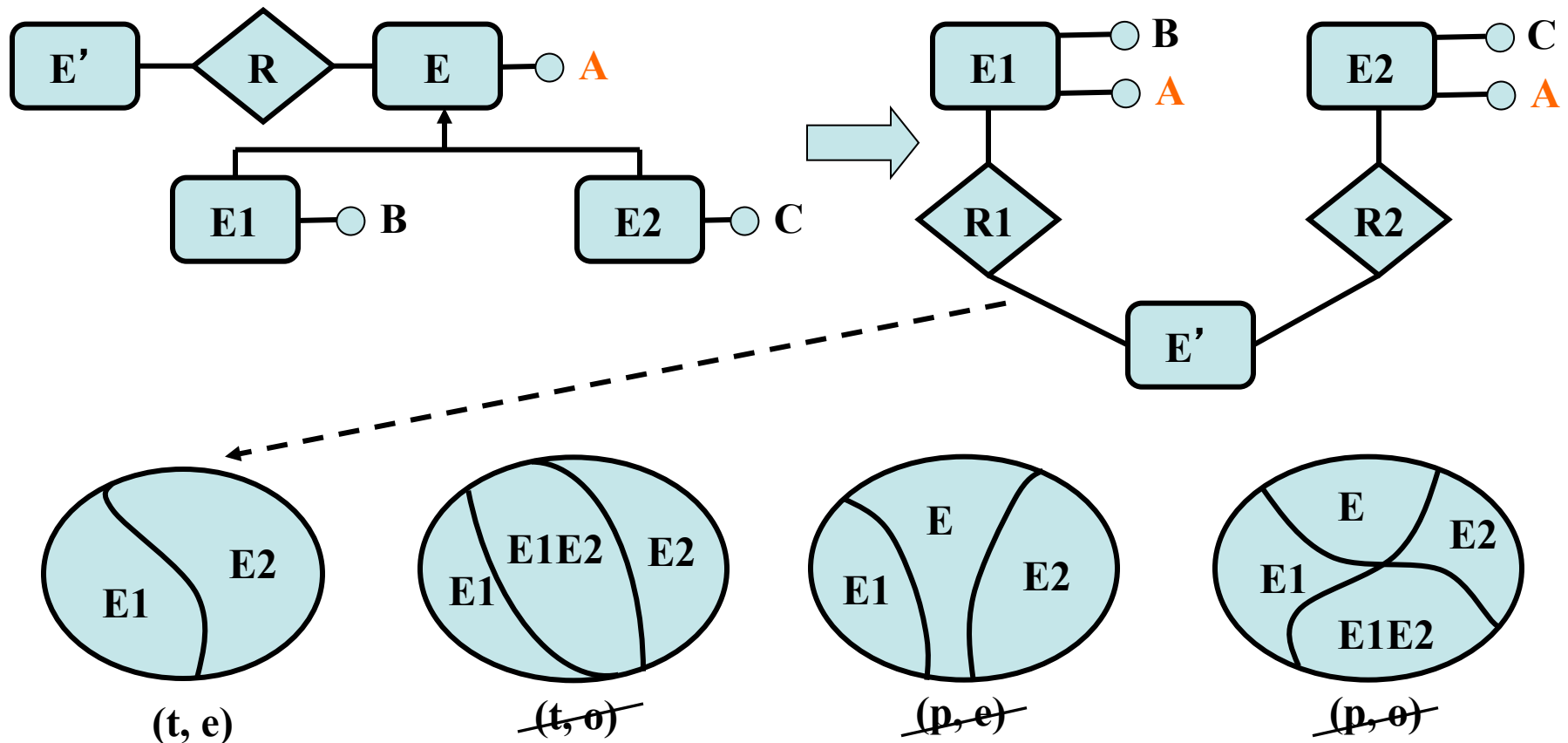
- Đặc điểm:

- Thuộc tính và mối kết hợp của thực thể tổng quát sẽ được chuyển xuống thành thuộc tính và mối kết hợp của **tất cả** thực thể chuyên biệt.
- Loại bỏ thực thể tổng quát.

1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

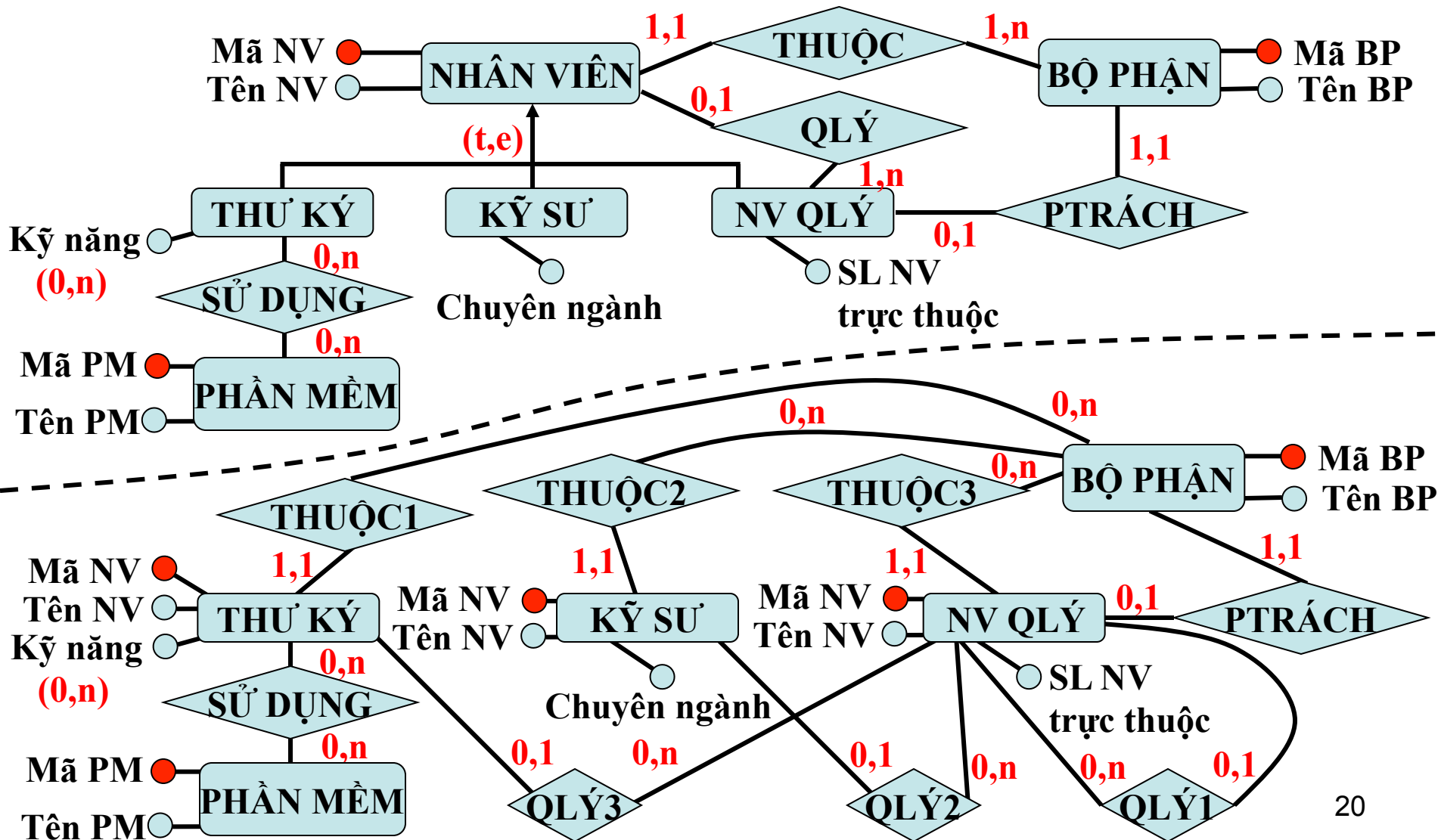
2. Chuyển đổi tổng quát hóa - Dùng thực thể chuyên biệt (tt)

- Ảnh hưởng sự tương quan:



1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

2. Chuyển đổi tổng quát hóa - Dùng thực thể chuyên biệt - Ví dụ:



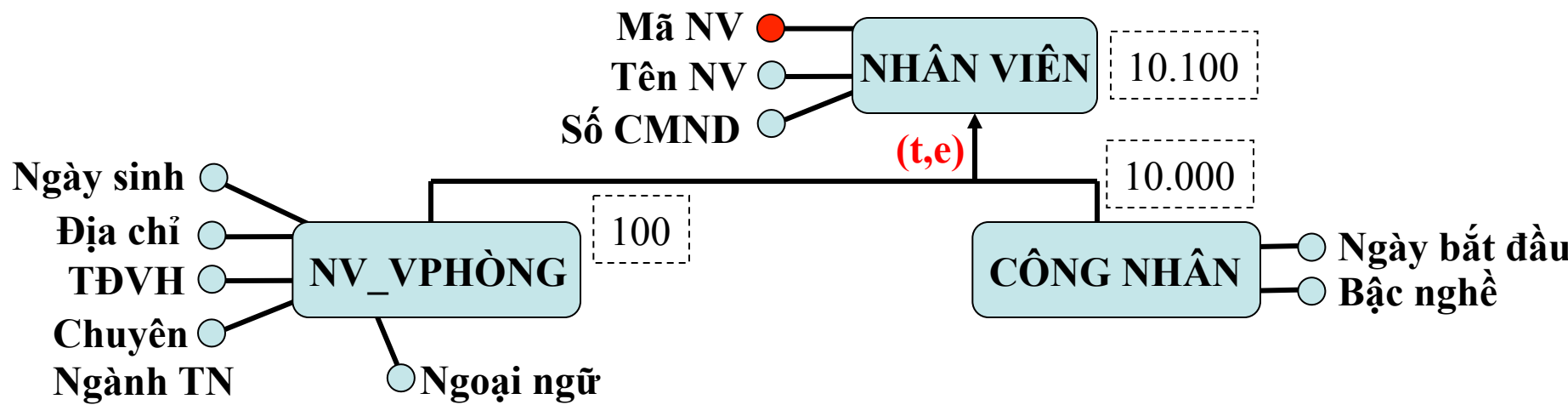
1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

2. Chuyển đổi tổng quát hóa - Dùng thực thể chuyên biệt (tt)

Ưu điểm	Khuyết điểm
Thuộc tính riêng của thực thể chuyên biệt chỉ biểu diễn cho loại thực thể chuyên biệt đó.	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 chuyên biệt chỉ truy xuất đến loại thực thể chuyên biệt đó.	Nếu số lượng thuộc tính của thực thể tổng quát là đáng kể thì sự lặp lại các thuộc tính này trong lược đồ kết quả là vấn đề cần phải xem xét lại.
	Các tác vụ thao tác lên thực thể tổng quát giờ phải thao tác lên tất cả thực thể chuyên biệt.

1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

2. Chuyển đổi tổng quát hóa - Chọn lựa giữa việc dùng thực thể tổng quát hay thực thể chuyên biệt:



Các xử lý liên quan		
Xử lý	Tần suất	Chọn lựa tối ưu
(o1) Tính lương sản phẩm cho công nhân phân xưởng	2/tháng	Tách
(o2) Tính lương cho nhân viên văn phòng	1/tháng	Tách
(o3) Tìm kiếm thông tin về công nhân	1000/ngày	Tách
(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

1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

2. Chuyển đổi tổng quát hóa - Chọn lựa giữa việc dùng thực thể tổng quát hay thực thể chuyên biệt (tt)

Việc lựa chọn phụ thuộc vào:

- **Dung lượng:**

- + Gộp: dung lượng lớn \rightarrow Truy xuất chậm
- + Tách: dung lượng tối ưu.

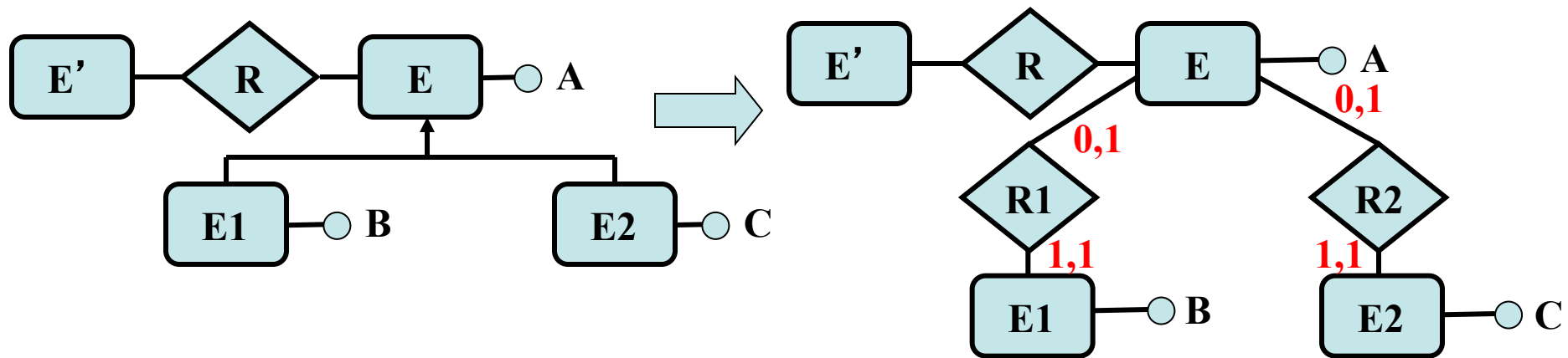
- **Xử lý:** xác định các xử lý ưu tiên (các xử lý có tần suất cao...) \rightarrow Tách/Gộp

Ví dụ: *ưu tiên (o4) \rightarrow Gộp*

ưu tiên (o2), (o3) \rightarrow Tách

1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

2. Chuyển đổi tổng quát hóa – Dùng mối kết hợp:

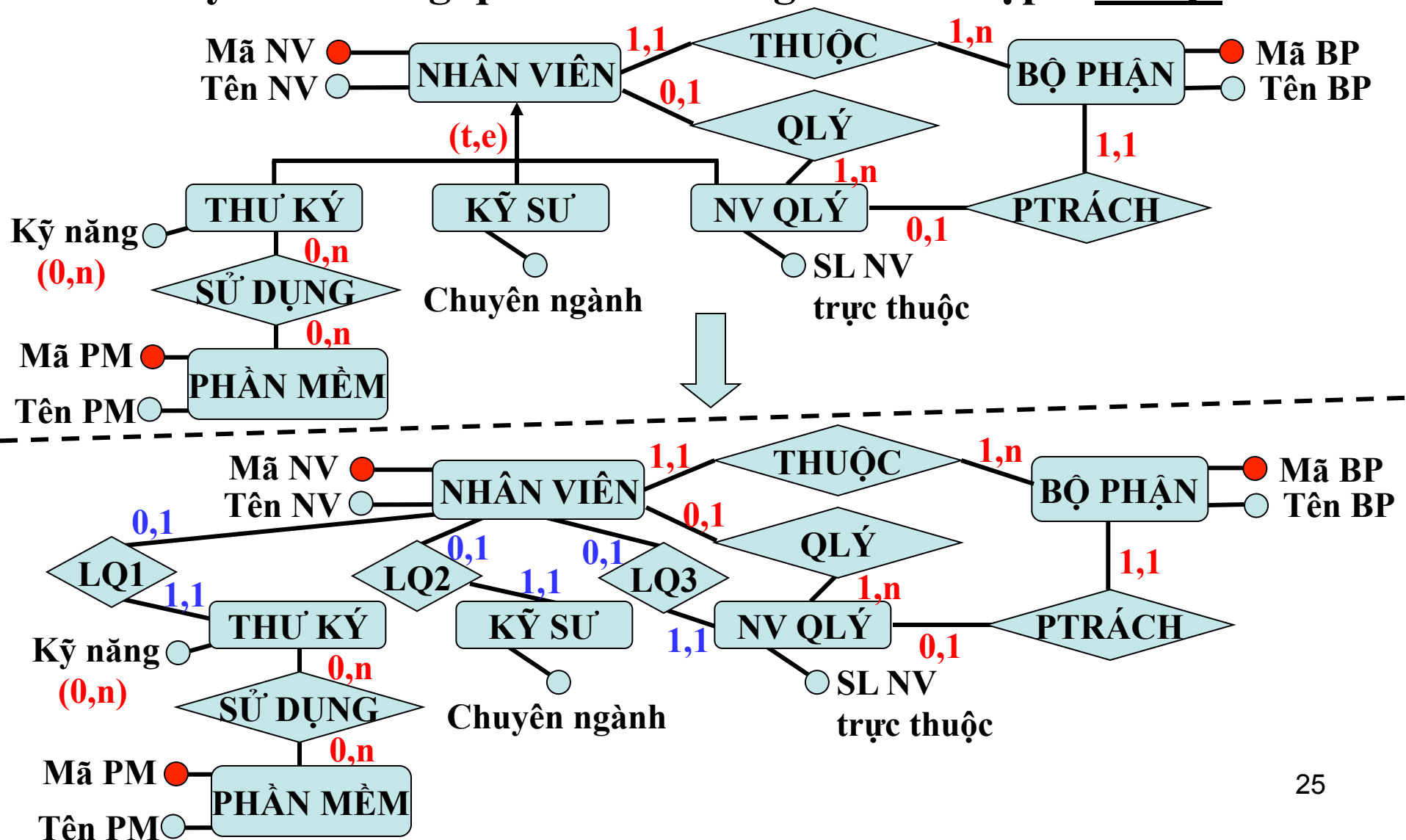


- **Đặc điểm:**

- Loại bỏ mối quan hệ tổng quát hóa.
- Tạo mối kết hợp từ thực thể tổng quát đến các thực thể chuyên biệt.
- Áp dụng cho tất cả cấu trúc tổng quát hóa (t,p,e,o).

1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

2. Chuyển đổi tổng quát hóa – Dùng mỗi kết hợp – Ví dụ:



1.1. Thiết kế luận lý dữ liệu cấp cao (tt)

2. Chuyển đổi tổng quát hóa – Dùng mối kết hợp (tt)

Ưu điểm	Khuyết điểm
Áp dụng cho tất cả cấu trúc tổng quát hóa (t,p,e,o).	Lược đồ kết quả khá phức tạp.
Rất uyển chuyển khi thay đổi yêu cầu ứng dụng.	Phải chấp nhận sự dư thừa khi biểu diễn mối quan hệ tổng quát hóa thành các mối kết hợp.

1.2. Thiết kế luận lý dữ liệu cấp thấp

1. Chuẩn bị chuyển đổi:

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

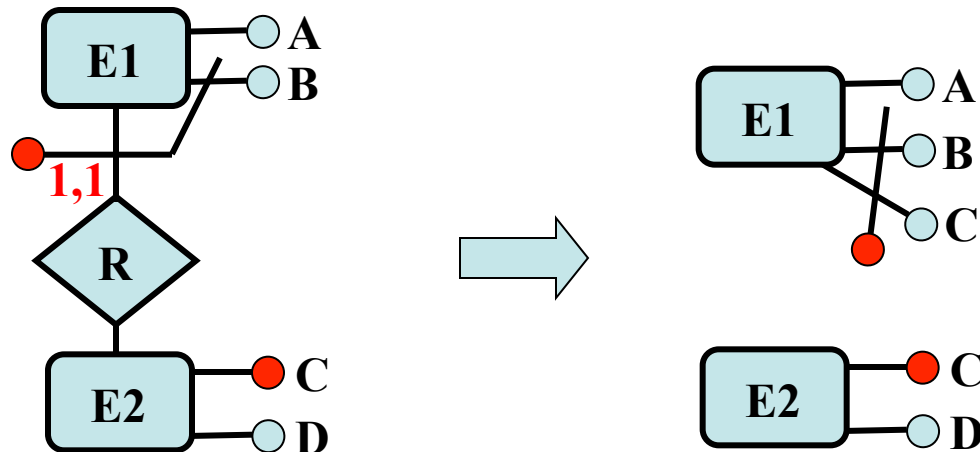
2. Chuyển đổi từ mô hình thực thể kết hợp sang mô hình quan hệ:

- Chuyển đổi thực thể
- Chuyển đổi mối kết hợp

1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

1. Chuẩn bị chuyển đổi:

- Loại bỏ định danh bên ngoài:



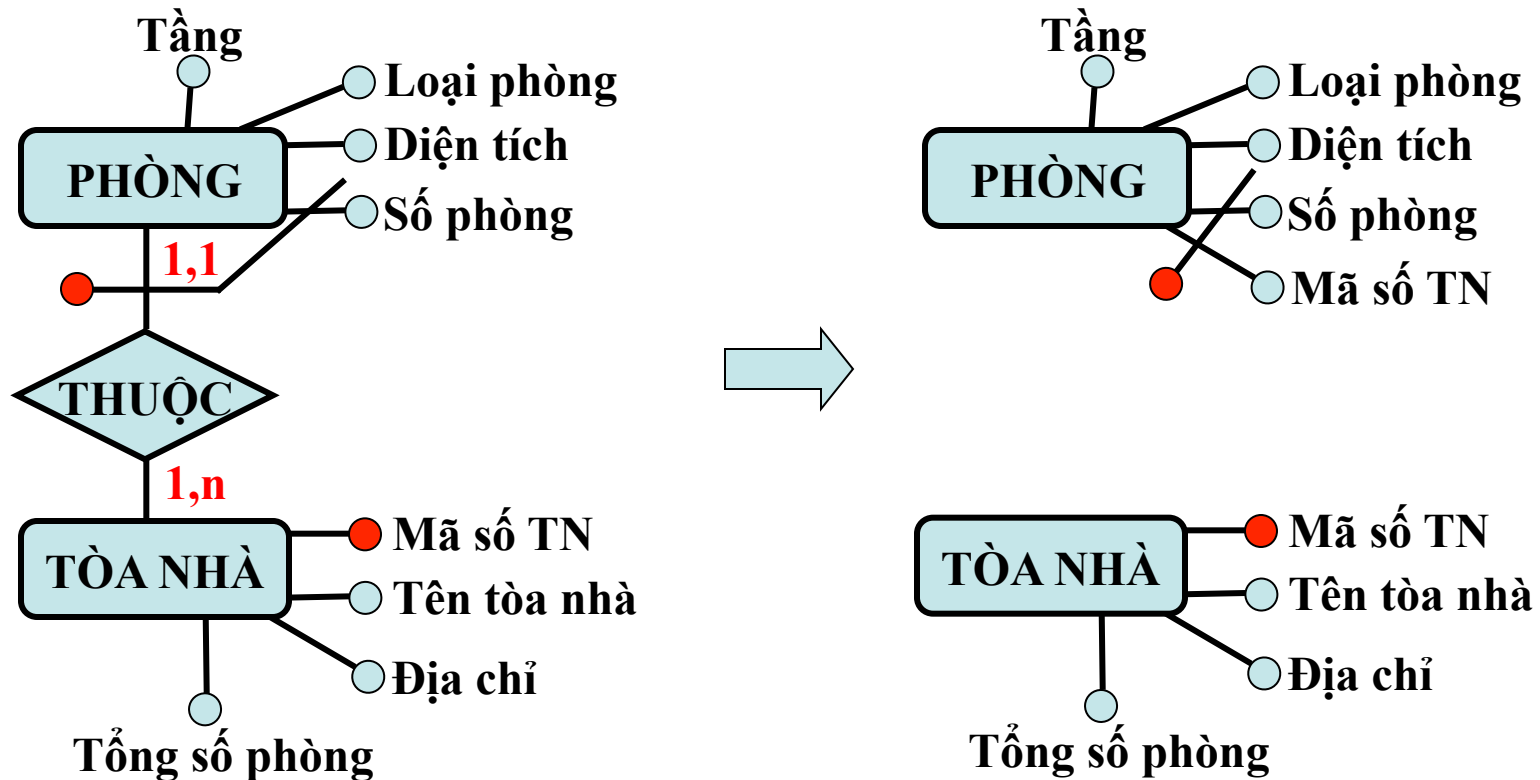
Cách làm:

- Định danh mới = định danh của các thực thể tham gia vào mỗi kết hợp + thuộc tính tham gia làm định danh ban đầu.
- Loại bỏ mỗi kết hợp.

1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

1. Chuẩn bị chuyển đổi (tt)

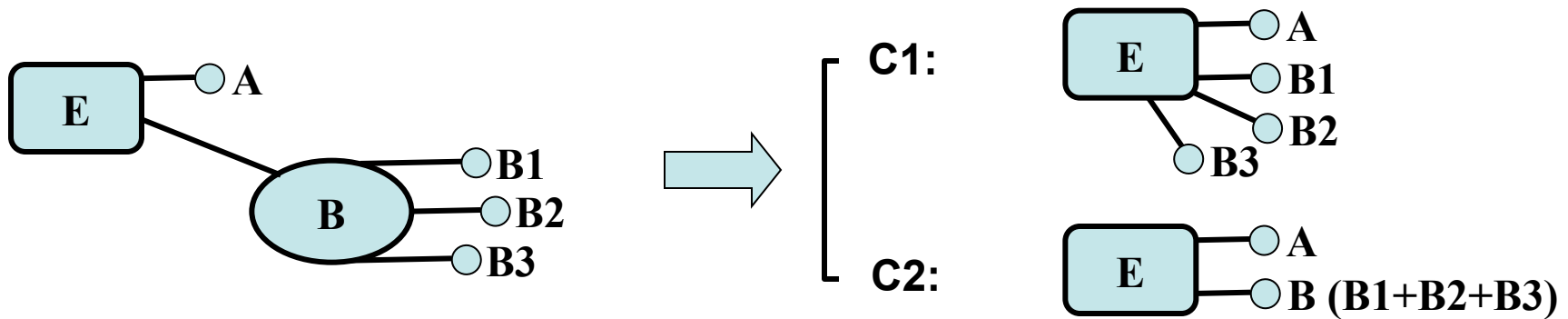
- Loại bỏ định danh bên ngoài - Ví dụ:



1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

1. Chuẩn bị chuyển đổi (tt)

- Loại bỏ thuộc tính kết hợp:



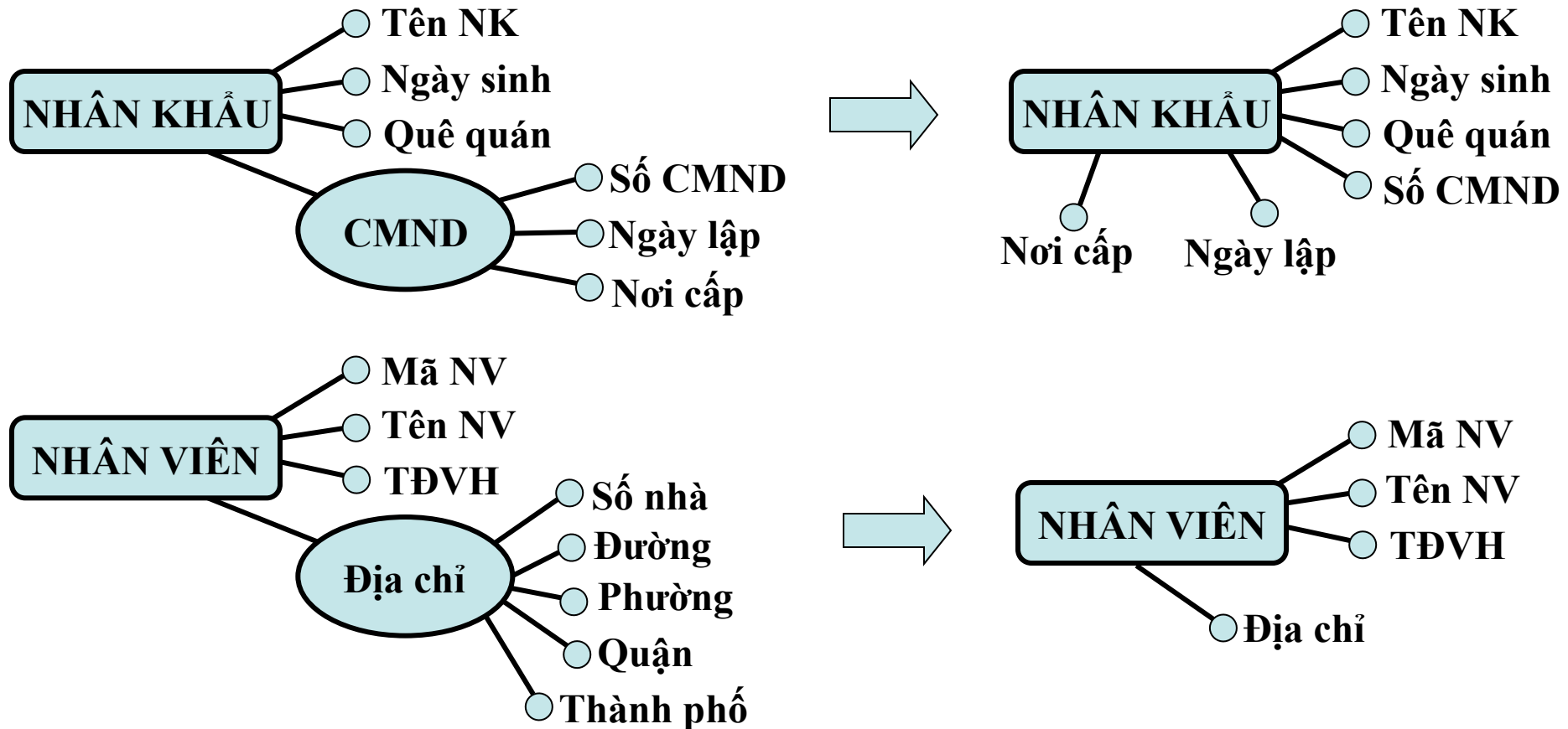
Cách làm:

- **C1:** Chuyển các thuộc tính thành phần thành các thuộc tính đơn khi các thuộc tính thành phần được truy cập nhiều.
- **C2:** Chuyển thuộc tính kết hợp thành một thuộc tính đơn khi không có nhu cầu truy xuất các thuộc tính thành phần.

1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

1. Chuẩn bị chuyển đổi (tt)

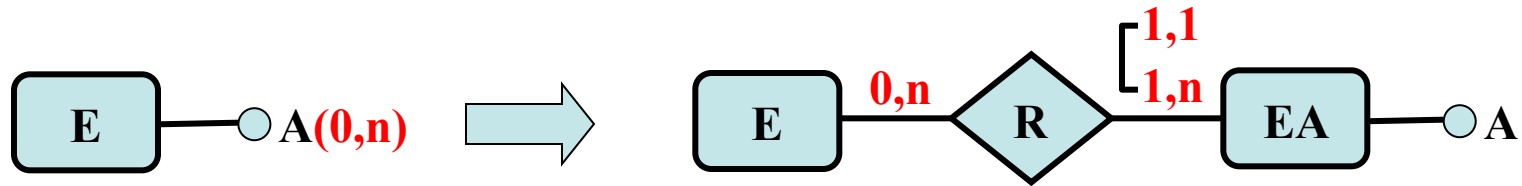
- Loại bỏ thuộc tính kết hợp – Ví dụ:



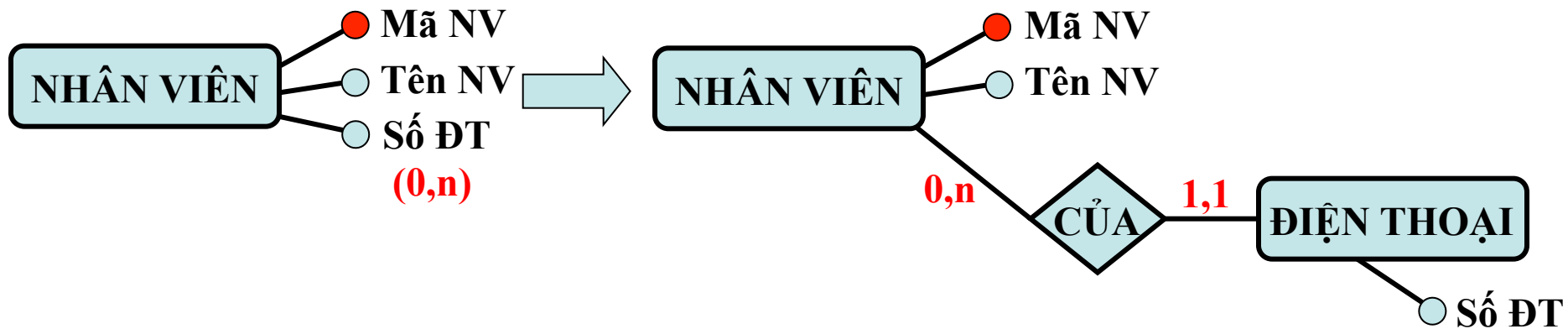
1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

1. Chuẩn bị chuyển đổi (tt)

- Loại bỏ thuộc tính đa trị của thực thể:



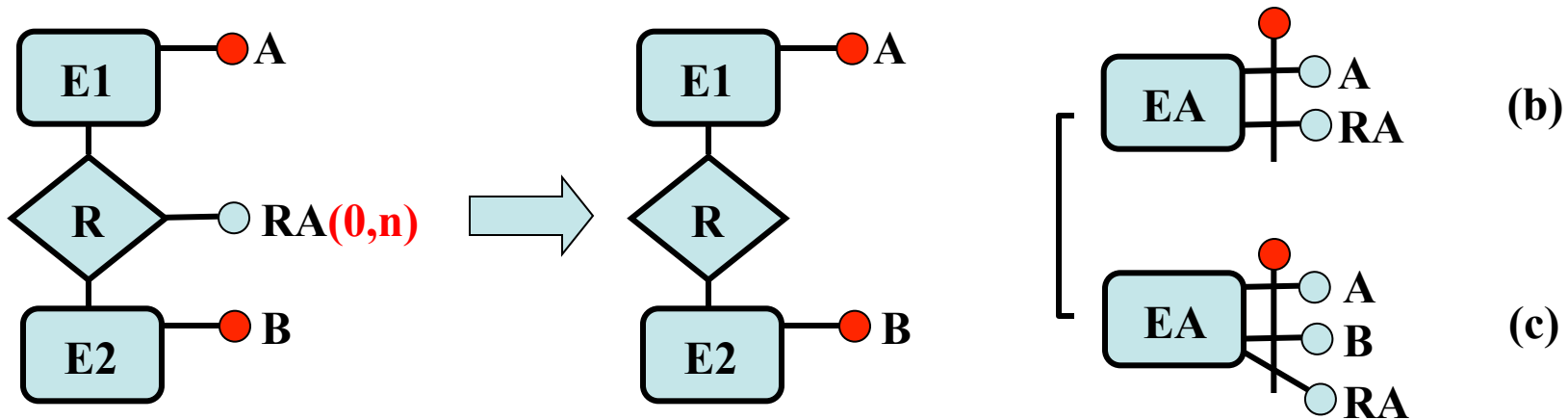
Ví dụ:



1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

1. Chuẩn bị chuyển đổi (tt)

- Loại bỏ thuộc tính đa trị của mỗi kết hợp:

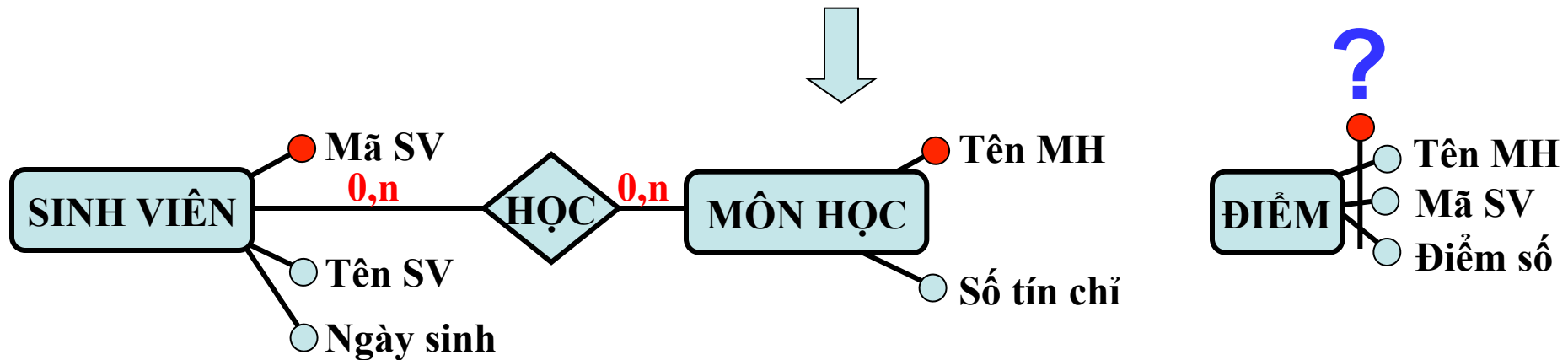
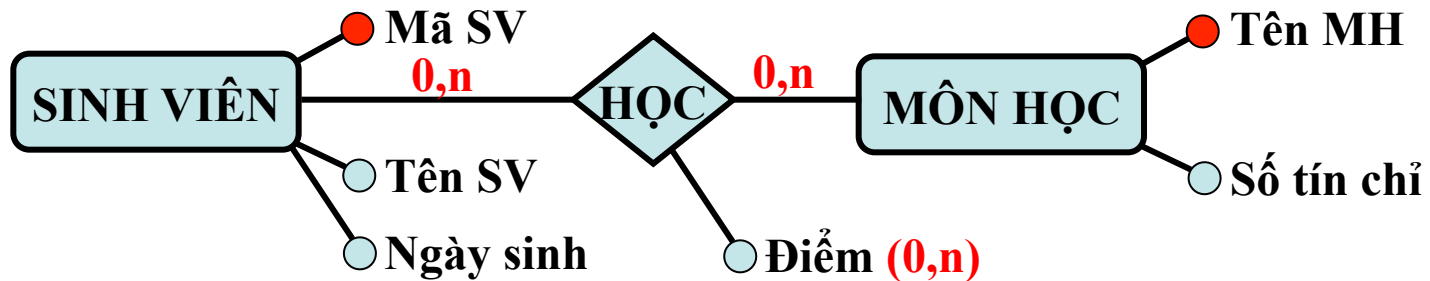


- (a) R là mối kết hợp 1-1: thực thể EA sẽ có định danh là định danh của **E1 hoặc E2** kết hợp với thuộc tính RA
- (b) R là mối kết hợp 1-n: thực thể EA sẽ có định danh là định danh của **E1** kết hợp với thuộc tính RA
- (c) R là mối kết hợp n-n: thực thể EA sẽ có định danh là định danh của **E1 và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

1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

1. Chuẩn bị chuyển đổi (tt)

- Loại bỏ thuộc tính đa trị của mỗi kết hợp – Ví dụ:

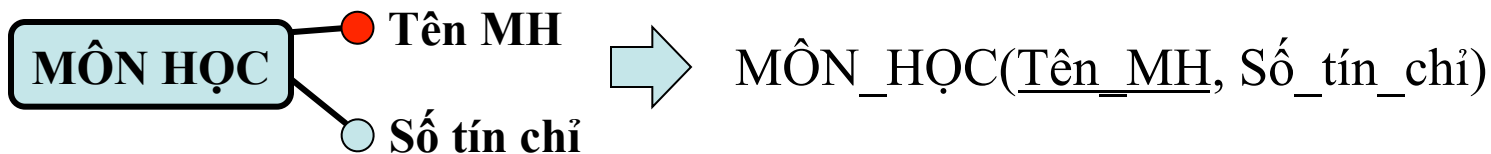


1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

2. Chuyển đổi từ mô hình TTKH → mô hình Quan hệ:

- Chuyển đổi thực thể:
 - Thực thể → lược đồ quan hệ
 - Thuộc tính → thuộc tính
 - Định danh → khóa chính

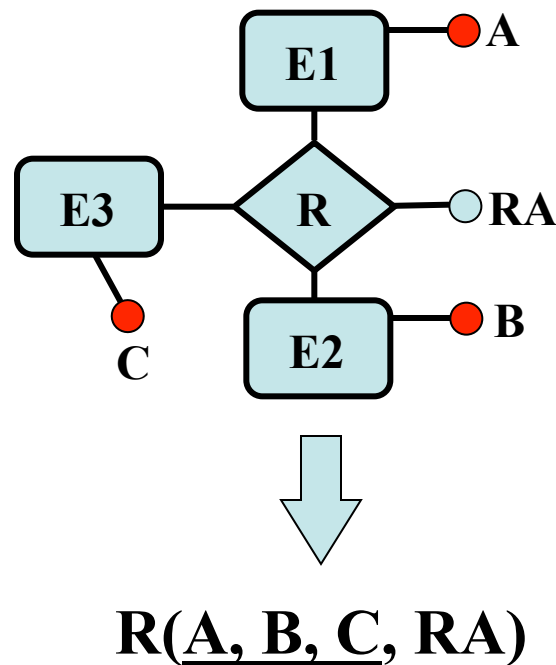
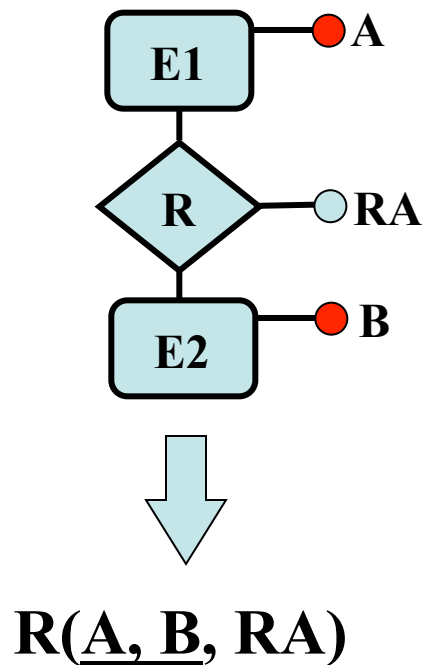
Ví dụ:



1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

2. Chuyển đổi từ mô hình TTKH → mô hình Quan hệ (tt)

- Chuyển đổi mỗi kết hợp:
 - Nguyên tắc chung:

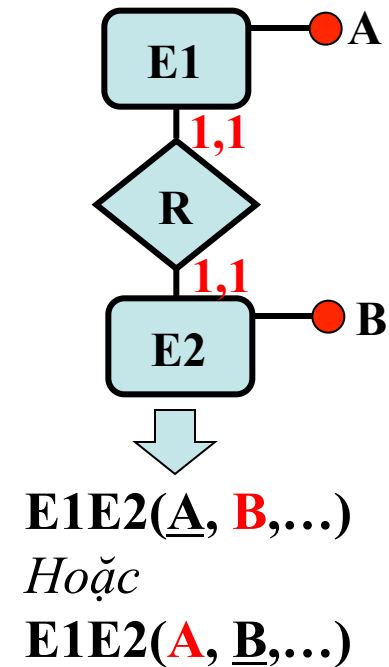
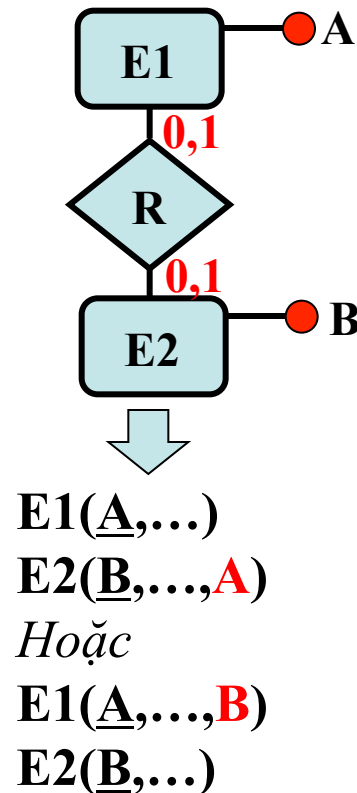
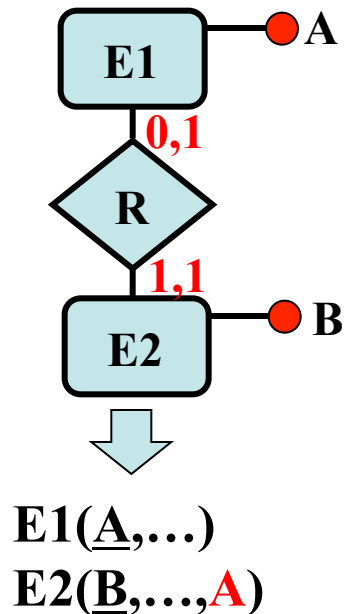


1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

2. Chuyển đổi từ mô hình TTKH → mô hình Quan hệ (tt)

- Chuyển đổi mỗi kết hợp (tt)

- Chuyển đổi mỗi kết hợp nhị phân - *Mỗi kết hợp 1-1*

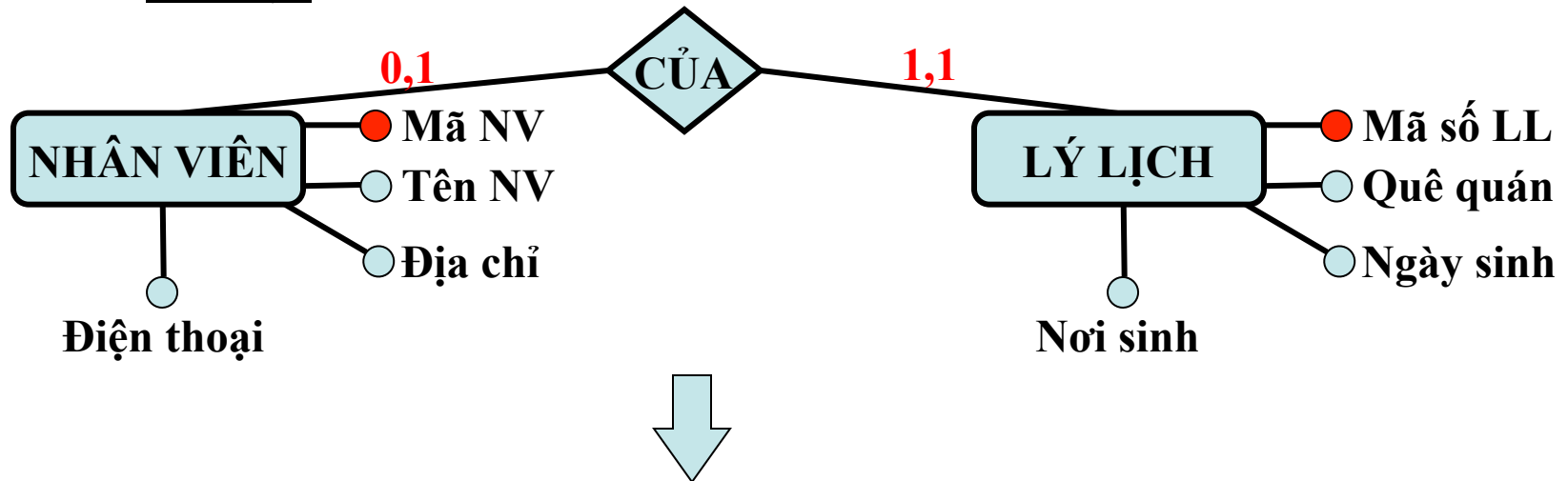


1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

2. Chuyển đổi từ mô hình TTKH → mô hình Quan hệ (tt)

- Chuyển đổi mỗi kết hợp (tt)
 - Chuyển đổi mỗi kết hợp nhị phân - *Mỗi kết hợp 1-1* (tt)

Ví dụ:



NHÂN_VIÊN(Mã_NV, Tên_NV, Địa_chỉ, Điện_thoại)

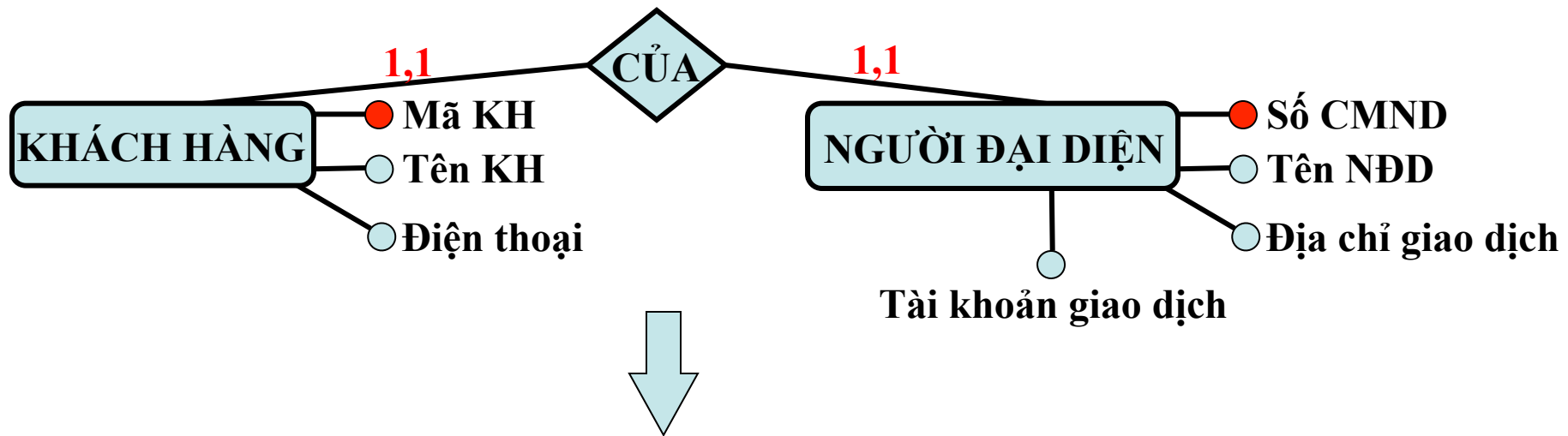
LÝ_LỊCH(Mã_số_LL, Quê_quán, Ngày_sinh, Nơi_sinh, **Mã_NV**)

1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

2. Chuyển đổi từ mô hình TTKH → mô hình Quan hệ (tt)

- Chuyển đổi mỗi kết hợp (tt)
 - Chuyển đổi mỗi kết hợp nhị phân - *Mỗi kết hợp 1-1 (tt)*

Ví dụ (tt)

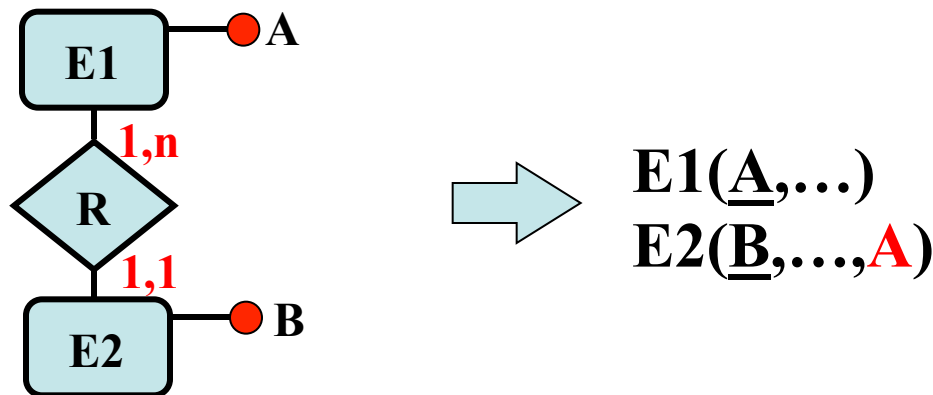


KH_NDD(Mã_KH, Tên_KH, Điện_thoại, Số_CMND, Tên_NDD, Địa_chỉ_giao_dịch, Tài_khoản_giao_dịch)

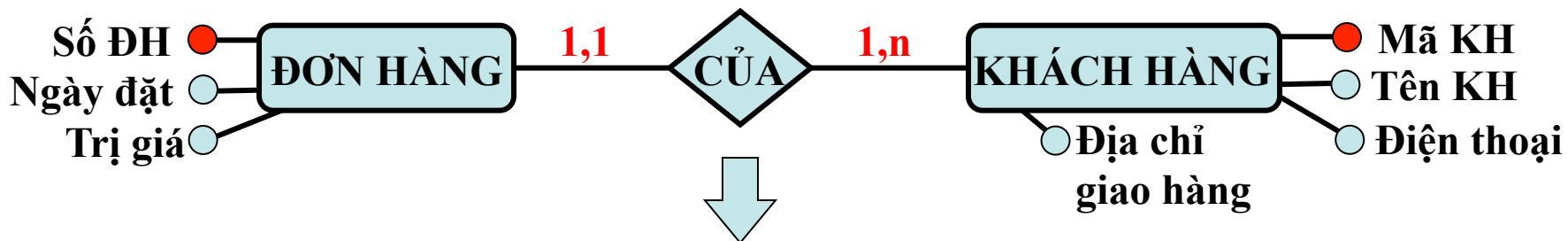
1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

2. Chuyển đổi từ mô hình TTKH → mô hình Quan hệ (tt)

- Chuyển đổi mỗi kết hợp (tt)
 - Chuyển đổi mỗi kết hợp nhị phân - *Mỗi kết hợp 1-n*



Ví dụ:



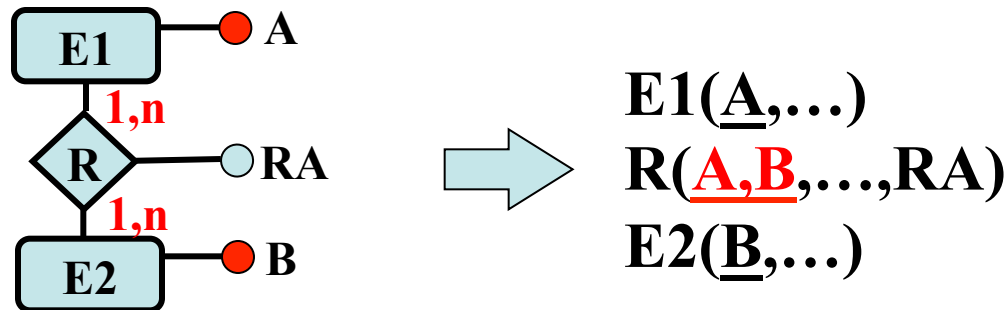
KHÁCH_HÀNG(Mã_KH, Tên_KH, Điện_thoại, Địa_chỉ_giao_hàng)

ĐƠN_HÀNG(Số_ĐH, Ngày_đặt, Trị_giá, Mã_KH)

1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

2. Chuyển đổi từ mô hình TTKH → mô hình Quan hệ (tt)

- Chuyển đổi mỗi kết hợp (tt)
 - Chuyển đổi mỗi kết hợp nhị phân - *Mỗi kết hợp n-n*



Ví dụ:



HÓA_ĐƠN(Số_HĐ, Ngày_lập, Trị_giá)

CHI_TIẾT_HÓA_ĐƠN(Số_HĐ, Mã_số, Số_lượng, Đơn_giá)

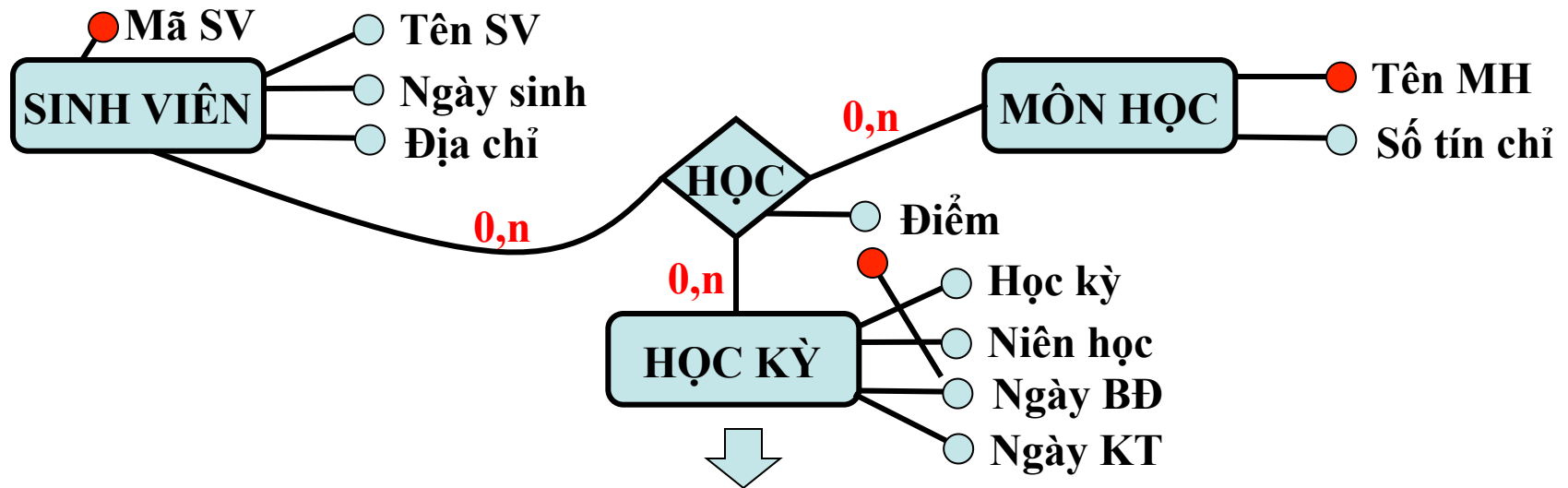
NGK(Mã_số, Tên_NGK, ĐVT, Loại, Hiệu, Đơn_giá_bán)

1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

2. Chuyển đổi từ mô hình TTKH → mô hình Quan hệ (tt)

- Chuyển đổi mỗi kết hợp (tt)

- Chuyển đổi mỗi kết hợp đa phân – Ví dụ:



SINH_VIÊN(Mã_SV, Tên_SV, Ngày_sinh, Địa_chỉ)

MÔN_HỌC(TÊN_MH, Số_tín_chỉ)

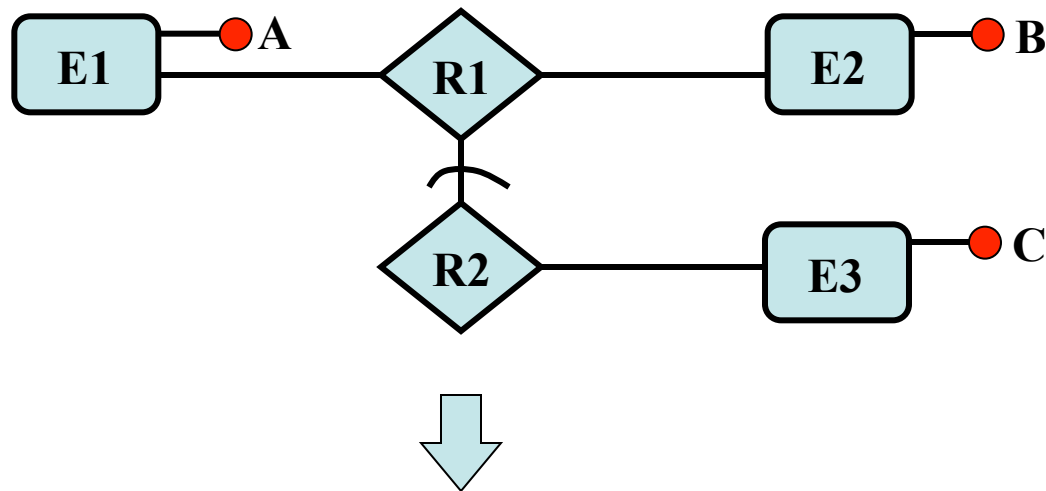
HỌC_KỲ(Học_kỳ, Niên_học, Ngày_BĐ, Ngày_KT)

HỌC(Mã_SV, Tên_MH, Học_kỳ, Niên_học, Điểm)

1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

2. Chuyển đổi từ mô hình TTKH → mô hình Quan hệ (tt)

- Chuyển đổi mỗi kết hợp (tt)
 - Chuyển đổi mỗi kết hợp mở rộng:



R1(A, B, ...)

R2(A, B, C, ...)

E1(A, ...)

E2(B, ...)

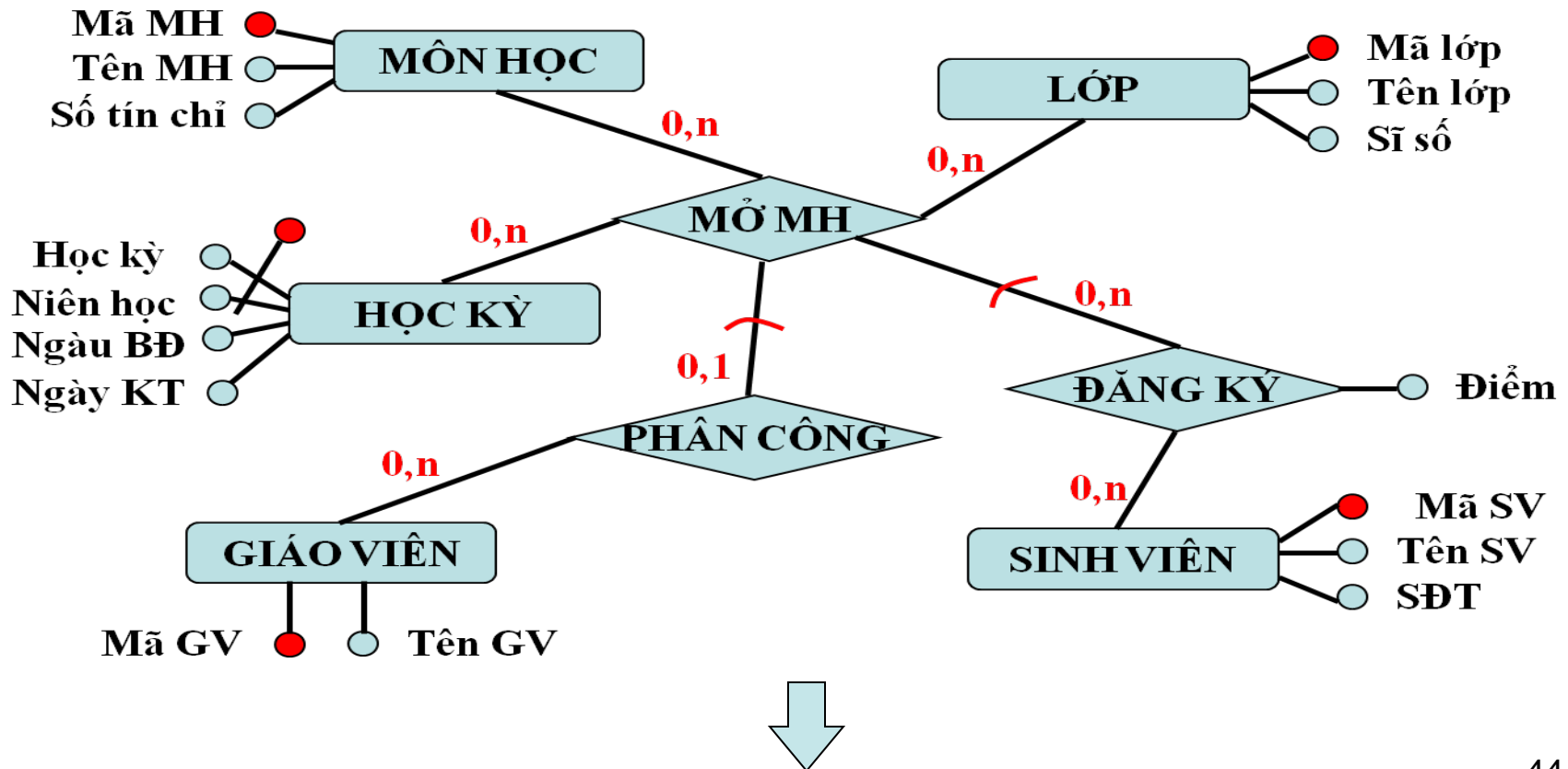
E3(C, ...)

1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

2. Chuyển đổi từ mô hình TTKH → mô hình Quan hệ (tt)

- Chuyển đổi mỗi kết hợp (tt)

- Chuyển đổi mỗi kết hợp mở rộng – Ví dụ:



1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

2. Chuyển đổi từ mô hình TTKH → mô hình Quan hệ (tt)

- Chuyển đổi mỗi kết hợp (tt)
 - Chuyển đổi mỗi kết hợp mở rộng – Ví dụ (tt)

Thực thể:

MÔN_HỌC(Mã_MH, Tên_MH, Số_tín_chỉ)

HỌC_KỲ(Học_kỳ, Niên_học, Ngày_BĐ, Ngày_KT)

GIÁO_VIÊN(Mã_GV, Tên_GV)

SINH_VIÊN(Mã_SV, Tên_SV, SĐT)

LỚP(Mã_lớp, Tên_lớp, Sĩ_số)

Mối kết hợp:

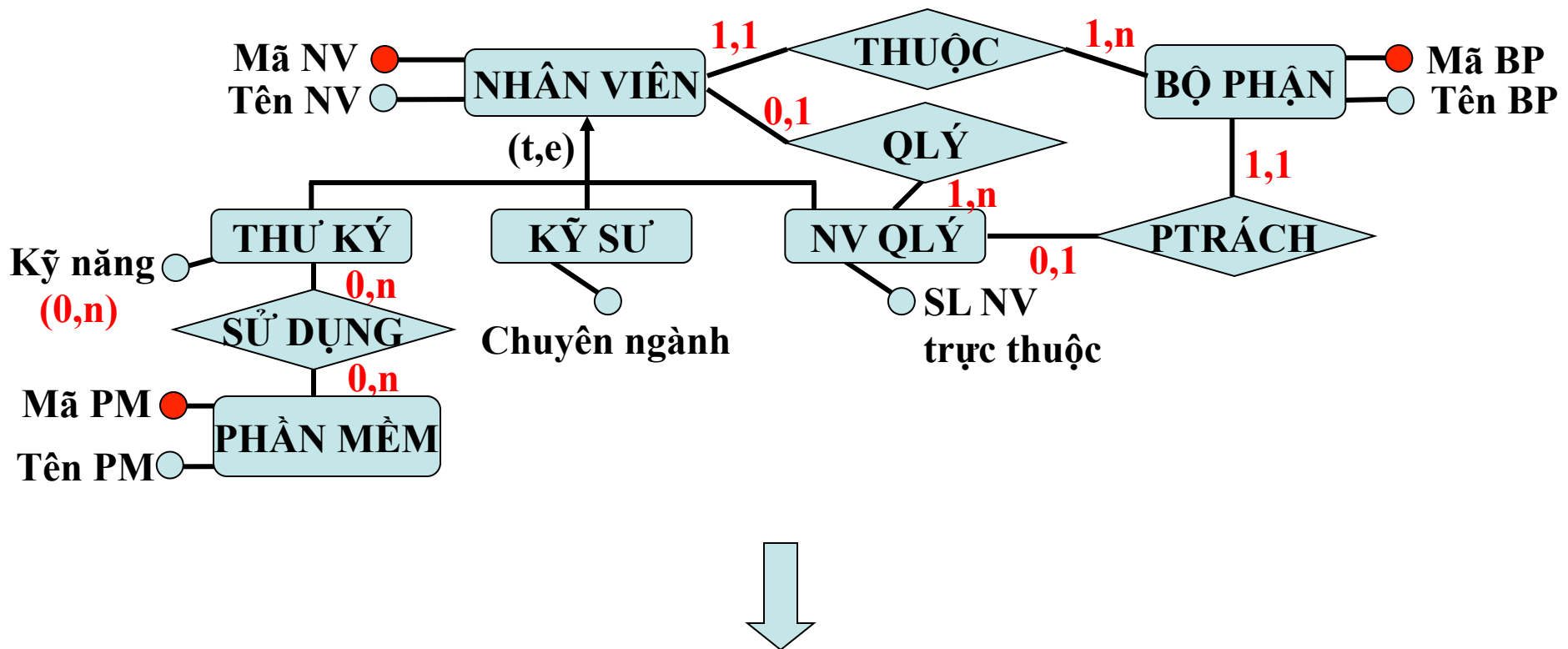
MỞ_MH(Mã_MH, Mã_lớp, Học_kỳ, Niên_học)

ĐĂNG_KÝ(Mã_MH, Mã_lớp, Học_kỳ, Niên_học, Mã_SV, Điểm)

PHÂN_CÔNG(Mã_MH, Mã_lớp, Học_kỳ, Niên_học, Mã_GV)

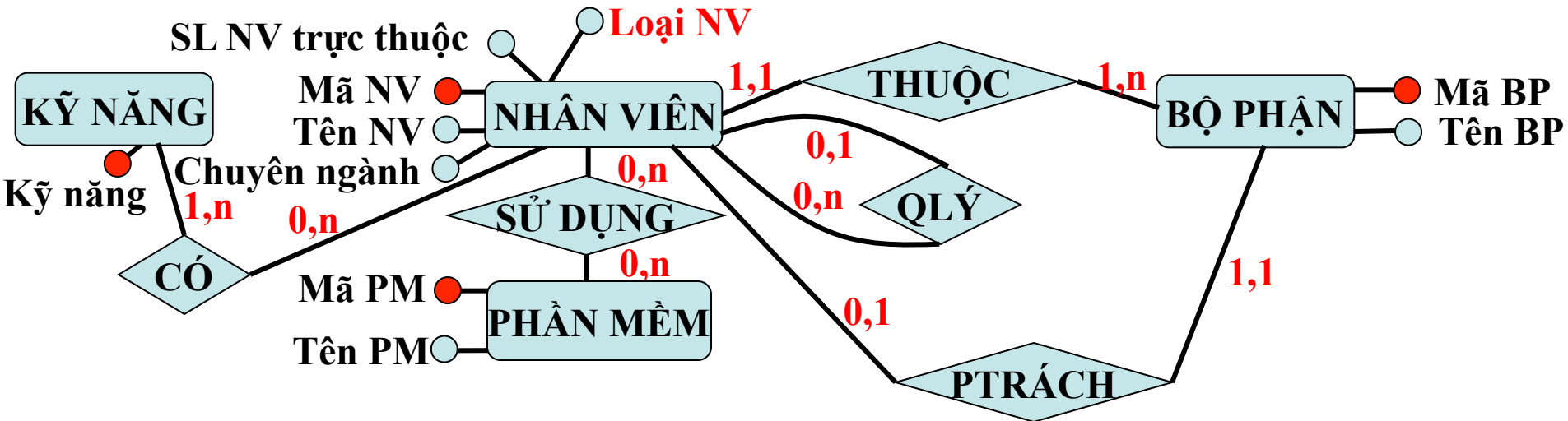
1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

Ví dụ tổng hợp:



1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

Ví dụ tổng hợp – Chuẩn bị chuyển đổi:



1.2. Thiết kế luận lý dữ liệu cấp thấp (tt)

Ví dụ tổng hợp – Chuyển đổi từ MH TTKH -> MH QH:

Thực thể:

NHÂN_VIÊN(Mã_NV, Tên_NV, Chuyên_ngành,
SL_NV_trực_thuộc, Loại_NV, Mã_BP, Mã_NVQL)

//NVQL trực tiếp của nhân viên

BỘ_PHẬN(Mã_BP, Tên_BP, Mã_NVQL) //NVQL cả bộ phận

PHẦN_MỀM(Mã_PM, Tên_PM)

KỸ_NĂNG(Kỹ_năng)

Mối kết hợp:

SỬ_DỤNG(Mã_NV, Mã_PM)

CÓ(Kỹ_năng, Mã_NV)

2. Thiết kế mã

- Vận dụng các kiểu mã hóa (liên tiếp, theo lát, phân đoạn, phân cấp, diễn nghĩa) để thiết kế mã.
- **Mục tiêu của thiết kế mã:**
 - Dễ dàng hơn cho việc nhận dạng và phân loại dữ liệu;
 - Thể hiện thứ tự để sắp xếp và tìm kiếm;
 - Tối ưu hóa về kích thước;
 - Có khả năng mở rộng.

2. Thiết kế mã (tt)

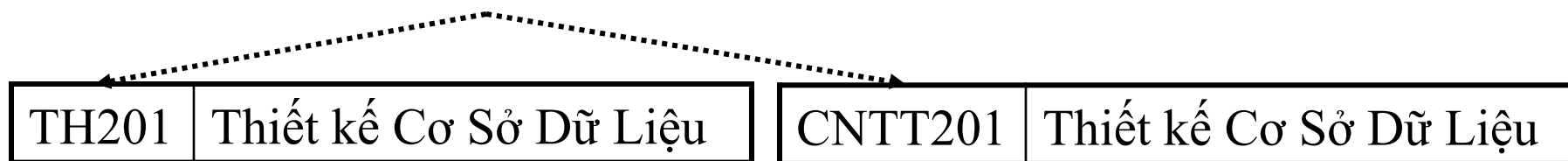
- **Ngữ nghĩa của mã hóa:**
 - **Duy nhất:** mã phải là duy nhất để nhận dạng đối tượng dữ liệu.
 - **Phân loại:** phân loại đối tượng dữ liệu thành những nhóm khác nhau và dựa vào mã có thể nhận ra được nhóm của dữ liệu.
 - **Sắp xếp:** mã hóa để thể hiện thứ tự của dữ liệu.
 - **Kiểm tra:** kiểm tra tính hợp lệ của mã được nhập.

2. Thiết kế mã (tt)

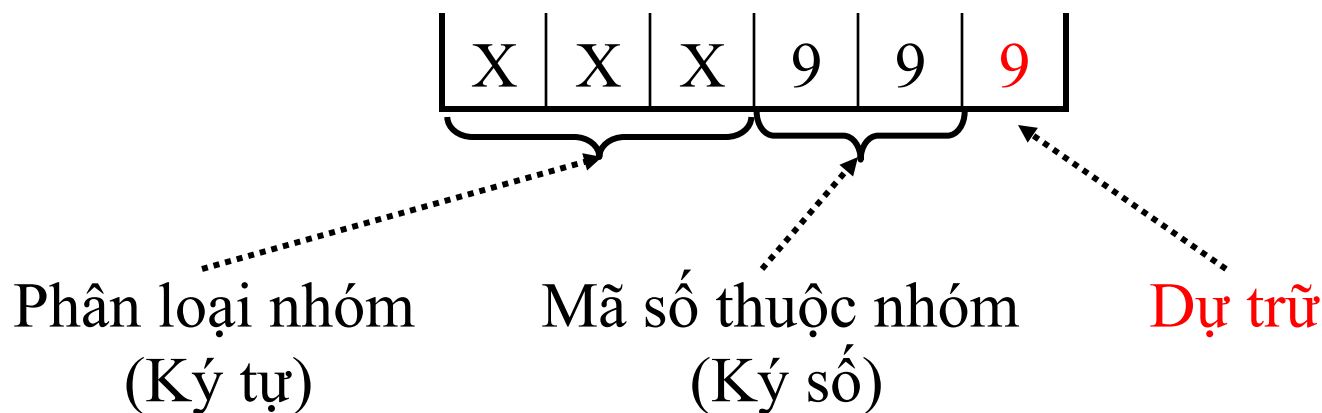
- **Các điểm cần xem xét khi thiết kế mã:**
 - **Vùng mã hóa:** phạm vi mà mã được sử dụng.
 - Khi liên kết với các hệ thống bên ngoài, mã nên được dùng như một chuẩn công nghiệp chung.
 - Khi được dùng trong cùng đơn vị, mã nên có phạm vi trong toàn công ty.
 - **Số lượng sử dụng:** ước lượng số lượng cần sử dụng hiện tại và trong tương lai -> thiết kế thêm phần dự trữ đặt ở cuối mã.
 - **Ngữ nghĩa:** mã có thể hiểu được bởi các thành viên liên quan.

2. Thiết kế mã (tt)

- Ví dụ vi phạm vùng mã hóa: mã môn học được thiết kế khác nhau trong cùng một trường.



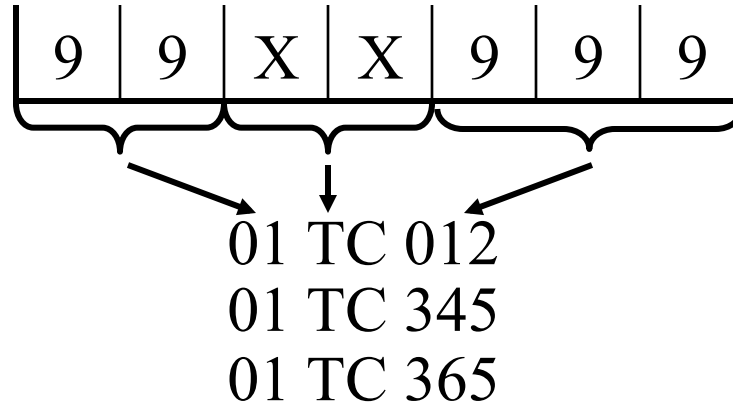
- Ví dụ thiết kế thêm phần dự trữ đặt ở cuối mã:



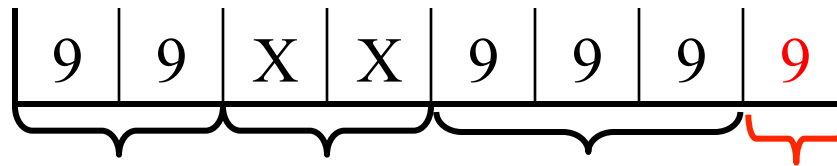
2. Thiết kế mã (tt)

Ví dụ: mã số sinh viên trong trường đại học

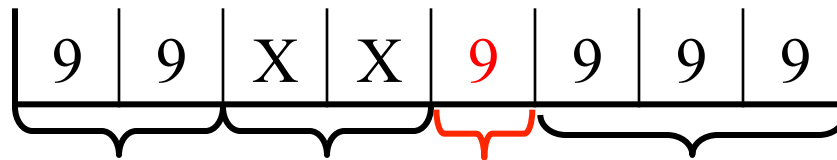
Không dự trữ:



Có dự trữ:



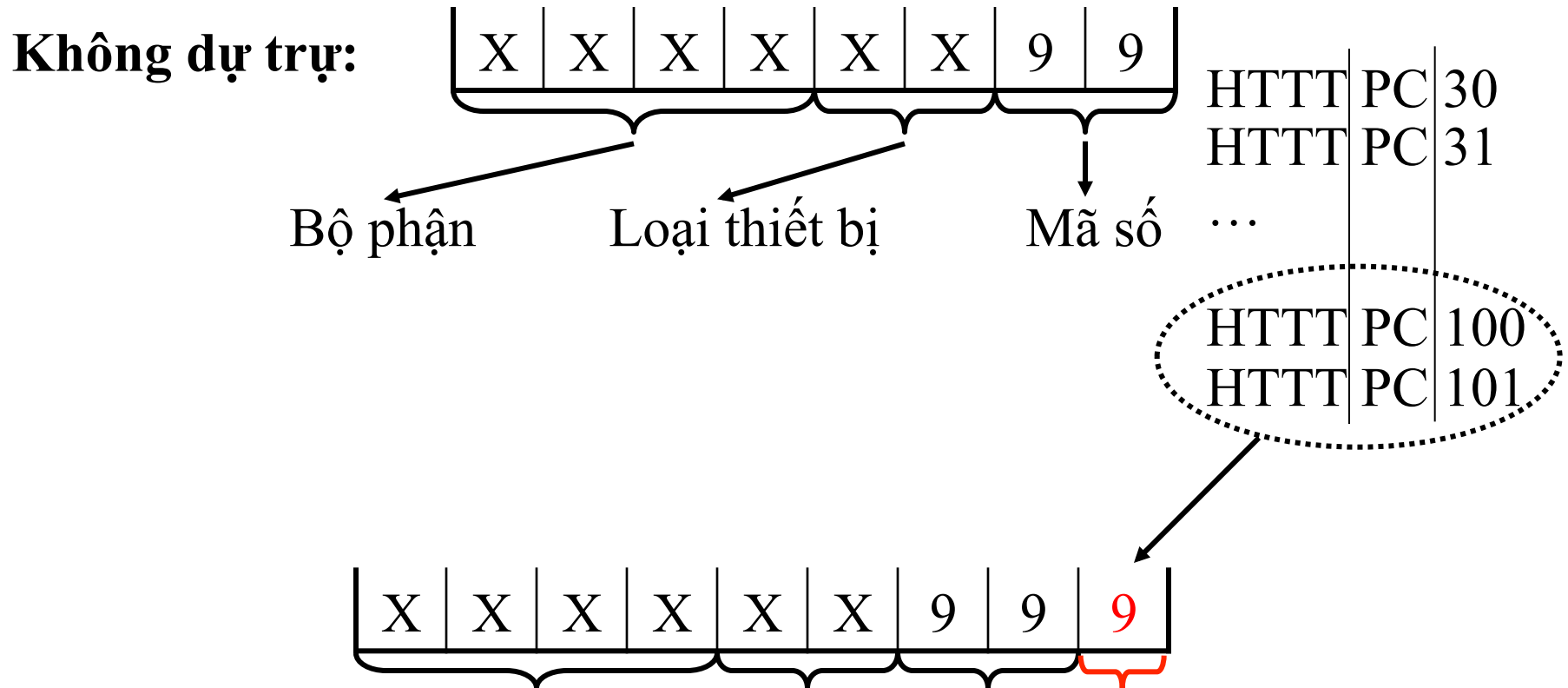
01 TC 0120
01 TC 3450
01 TC 3650



01 TC 0012
01 TC 0345
01 TC 0365

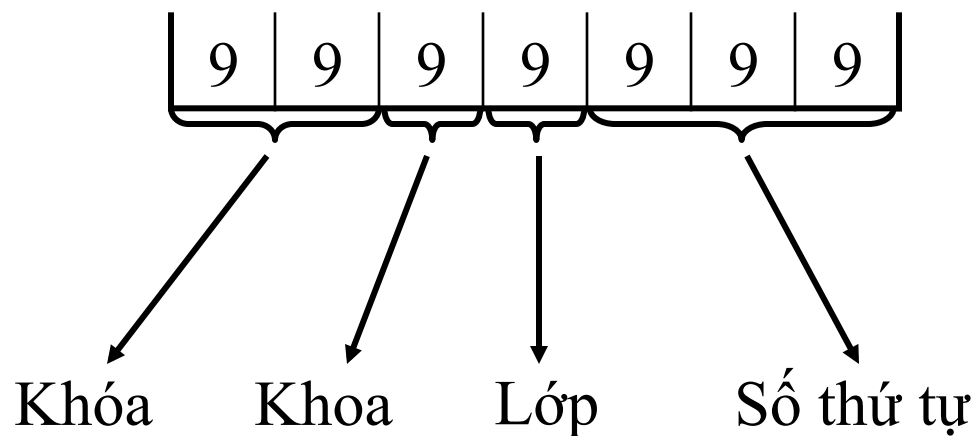
2. Thiết kế mã (tt)

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



2. Thiết kế mã (tt)

- Ví dụ đưa yếu tố ngữ nghĩa vào trong thiết kế mã:



00	1	1	037
00	1	1	456
00	1	1	230

2. Thiết kế mã (tt)

• Ví dụ tài liệu mã hóa:

Tên mã	Mã_BP
Mục đích	Định danh và phân loại tổ chức
Định dạng	<div><div>9</div><div>9</div><div>9</div><div>9</div></div>
Cách gán mã	<div><div>Ban GĐ^Đốc10</div><div><div>BP KDoanh1</div><div>BP P²Triển2</div><div>Các BP Khác3</div></div><div><div><div>Nhóm KD 11</div><div>Nhóm KD 22</div><div>Nhóm KD 33</div></div><div><div>Nhóm PT 11</div><div>Nhóm PT 22</div><div>Nhóm PT 33</div></div><div><div>Nhân sự1</div><div>Kế toán2</div><div>Sản xuất3</div><div>Kế hoạch4</div><div>Thư ký5</div></div><div><div>Chi nhánh 120</div><div>Chi nhánh 230</div></div></div></div>
Ghi chú	Sắp tới khi có chi nhánh/bộ phận phòng ban mới sẽ được gán một số tiếp theo.

2. Thiết kế mã (tt)

- Mô tả các loại mã:

Tên mã	Mô tả	Ví dụ
Mã tuần tự	<i>Dùng kiểu mã hóa liên tiếp.</i>	Mã số xe của tỉnh: 50 : Thành phố HCM 60 : Đồng Nai 62 : Long An 63 : Tiền Giang ...
Mã khối	<i>Dùng kiểu mã hóa theo lát:</i> Dùng từng lát cho từng nhóm đối tượng, trong mỗi lát dùng kiểu mã hóa liên tiếp. → Hỗ trợ cho việc phân loại dữ liệu.	Mã khách hàng của từng chi nhánh được quy định như sau: 1->999: KH tại trung tâm 1000->1999: KH ở chi nhánh A 2000->2999: KH ở chi nhánh B 3000->3999: KH ở chi nhánh C ...

2. Thiết kế mã (tt)

- Mô tả các loại mã (tt)

Tên mã	Mô tả	Ví dụ												
Mã thập phân	<p><i>Dùng kiểu mã hóa phân cấp.</i></p> <p>Các đối tượng sẽ được mã hóa từ 0 đến 9 rồi đến lượt thành viên của mỗi đối tượng này cũng được mã hóa từ 0 đến 9, ...</p>	Mã phòng ban của 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	<p><i>Dùng kiểu mã hóa phân đoạn.</i></p> <p>Mỗi ký số của mã sẽ được gán một ngữ nghĩa.</p>	Mã sinh viên của một trường đại học: <table><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> ① Năm ② Khoa ③ Lớp ④ Số thứ tự	①	②	③	④	01	1	1	100	01	2	1	120
①	②	③	④											
01	1	1	100											
01	2	1	120											

2. Thiết kế mã (tt)

- Mô tả các loại mã (tt)

Tên mã	Mô tả	Ví dụ
Mã gọi nhớ	<i>Dùng kiểu mã hóa điển nghĩa.</i> Dùng chữ viết tắt hoặc biểu tượng của đối tượng để mã hóa.	Mã lớp học của một trường đại học: 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 (số) ký số thêm vào trước/sau mã. Mã bao gồm giá trị của mã và ký số kiểm tra.	Mã số thuế...

3. Thiết kế vật lý dữ liệu

Mục đích:

- Chuyển các mô tả luận lý dữ liệu 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.

- 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/restore
- Hệ quản trị CSDL sử dụng



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

3. Thiết kế vật lý dữ liệu (tt)

- **Thiết kế field**
- **Phân chia dữ liệu (partition)**
- **Gộp dữ liệu (denormalization)**
- **Thiết kế file dữ liệu vật lý**
- **Tổ chức file chỉ mục**
- **Gom cụm file (clustering)**

3. Thiết kế vật lý dữ liệu – Thiết kế field

- **Field:** đơn vị dữ liệu nhỏ nhất.
- **Thiết kế field bao gồm:**
 - a) Chọn kiểu dữ liệu:* phải thỏa các tiêu chuẩn sau:
 - Tối thiểu không gian lưu trữ;
 - Hiển thị tất cả tình huống giá trị;
 - Đảm bảo việc toàn vẹn dữ liệu;
 - Xem xét việc hỗ trợ các hình thức xử lý trên dữ liệu.

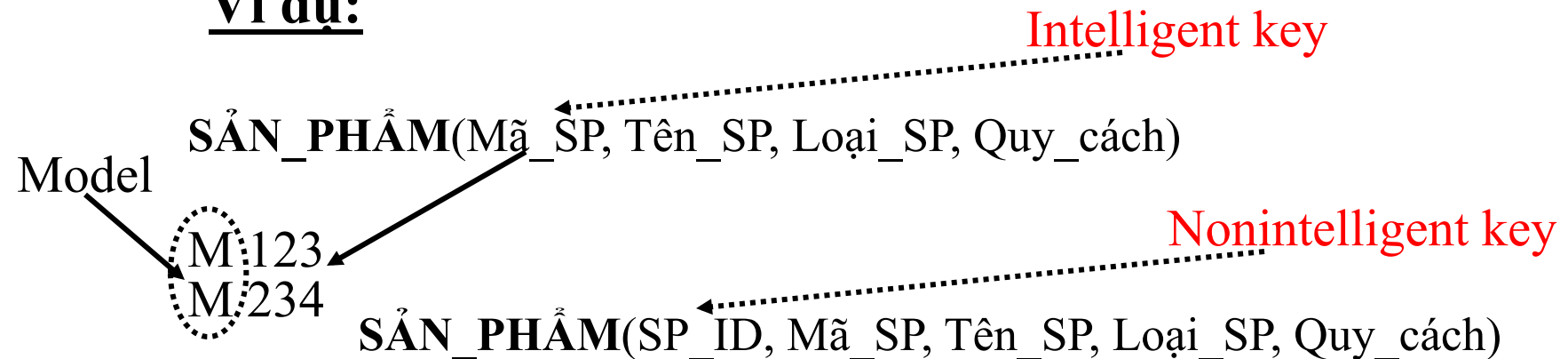
3. Thiết kế vật lý dữ liệu – Thiết kế field (tt)

b) Chọn lựa khóa chính:

- Việc chọn lựa khóa chính trong mô hình luận lý đôi khi không thuận tiện cho việc cập nhật.

→ Thêm một thuộc tính trừu tượng làm khóa chính (*Nonintelligent key*). Thuộc tính này không có trong thế giới thực, chỉ có trong phần mềm.

Ví dụ:



3. Thiết kế vật lý dữ liệu – Thiết kế field (tt)

Ví dụ (tt)

SẢN PHẨM

SP_ID	Mã_SP	Tên_SP	Loại_SP	Quy_cách
1	M123	A	M	12x4
2	M234	B	M	15x3
3	S012	C	S	12x2
4	L121	D	L	18x6

Mã_SP trong table SẢN PHẨM có thể được thay đổi mà không ảnh hưởng gì đến table HÓA ĐƠN có khóa ngoại là SP_ID tham chiếu đến table SẢN PHẨM

HÓA ĐƠN

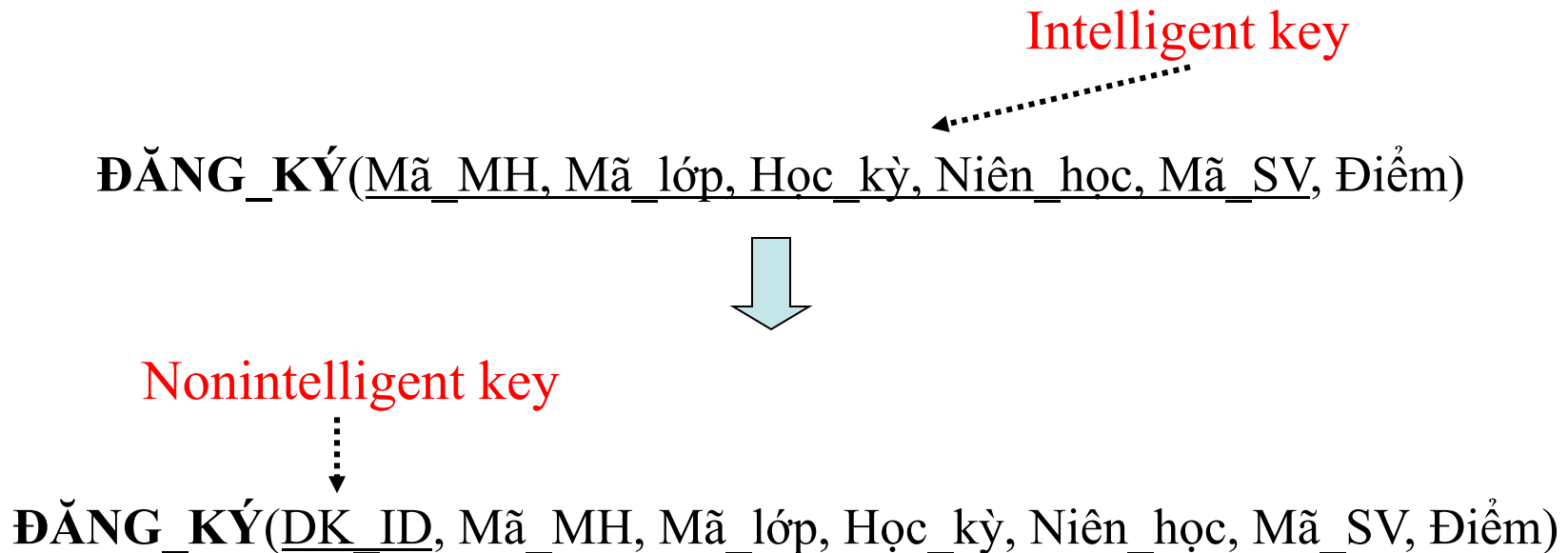
HD_ID	Số_HD	Ngày_lập	SP_ID	Số_lượng	Đơn_giá
1	001/HD	1/1/2004	1	10	200
2	002/HD	1/1/2004	1	5	200
3	003/HD	2/1/2004	2	120	120
4	004/HD	3/1/2004	3	200	700

3. Thiết kế vật lý dữ liệu – Thiết kế field (tt)

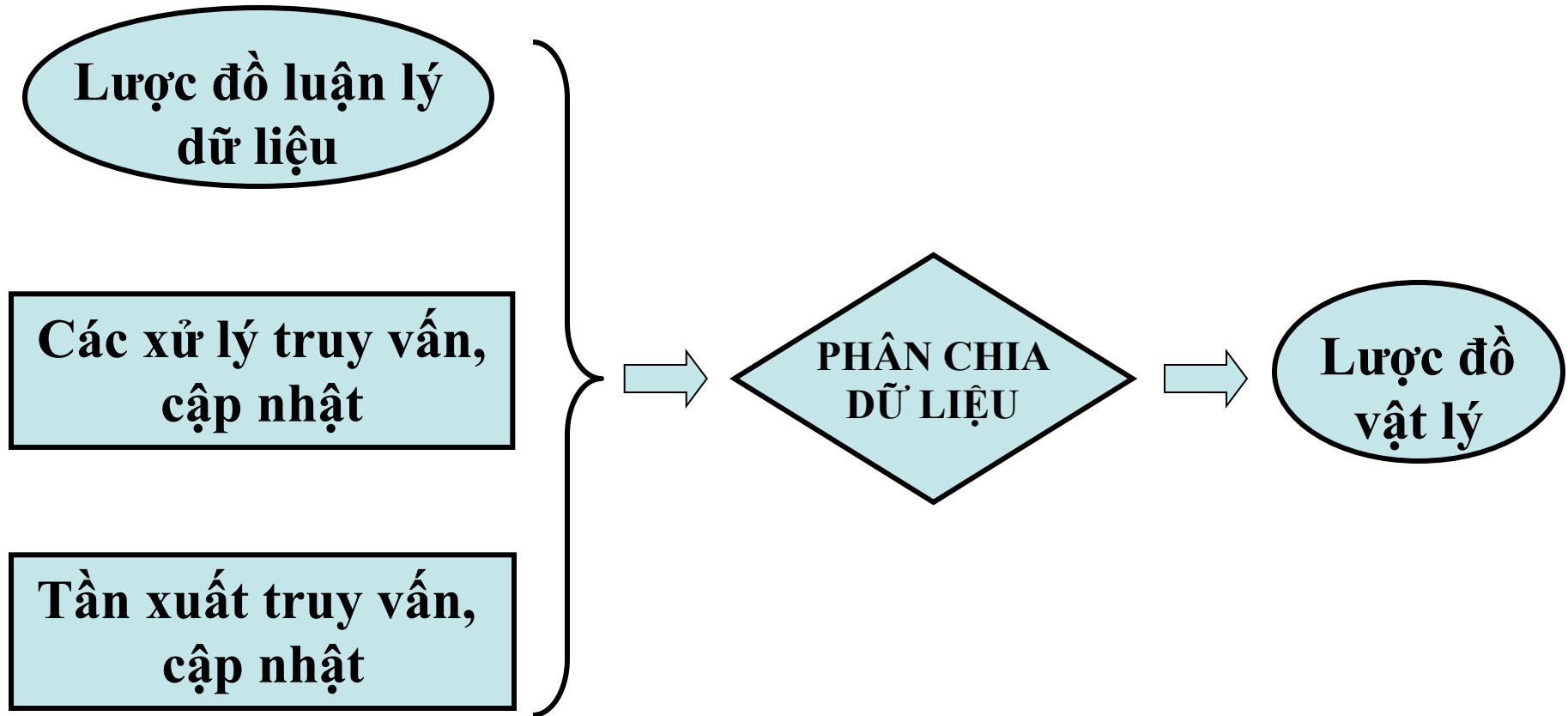
b) Chọn lựa khóa chính (tt)

- Khóa chính phức tạp sẽ làm giảm tốc độ truy cập CSDL.
- Sử dụng *Nonintelligent key* đơn giản và hiệu quả hơn.

Ví dụ:



3. Thiết kế vật lý dữ liệu – Phân chia dữ liệu



3. Thiết kế vật lý dữ liệu – Phân chia dữ liệu (tt)

a) Phân chia theo chiều ngang (horizontal partition)

- **Cách làm:** Phân chia các dòng dữ liệu 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.

3. Thiết kế vật lý dữ liệu – Phân chia dữ liệu (tt)

a) Phân chia theo chiều ngang (horizontal partition) - Ví dụ:

HÓA ĐƠN

KL:~10.000.000/năm

Số_HD	Ngày_lập	Diễn_giải	Trị_giá
Hd00001	1/1/2004	Xxxxxxxx	1.000.000
Hd00002	2/1/2004	Yxxxxxxx	2.000.000
...			
Hd15000	1/1/2005	Zzzzzzzzz	1.400.000
Hd15001	2/1/2005	Tttttttttt	2.100.000
...			
Hd30000	1/1/2006	Uuuuuuuu	12.000.000
Hd30001	2/1/2006	Vvvvvvvv	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

3. Thiết kế vật lý dữ liệu – Phân chia dữ liệu (tt)

a) Phân chia theo chiều ngang (horizontal partition) - Ví dụ (tt)

Tách table HÓA_ĐƠN theo năm, các xử lý O1, O2, O4 sẽ hiệu quả hơn.

HÓA_ĐƠN			
KL:~10.000.000/năm			
Số_HD	Ngày_lập	Diễn_giải	Trị_giá
Hd00001	1/1/2004	Xxxxxxxx	1.000.000
Hd00002	2/1/2004	Yxxxxxxx	2.000.000
...			
Hd15000	1/1/2005	Zzzzzzzzz	1.400.000
Hd15001	2/1/2005	Tttttttttt	2.100.000
...			
Hd30000	1/1/2006	Uuuuuuuu	12.000.000
Hd30001	2/1/2006	Vvvvvvvv	1.000.000

HD004

Số_HD	Ngày_HD	Diễn_giải	Trị_giá
Hd00001	1/1/2004	Xxxxxxxx	1.000.000
Hd00002	2/1/2004	Yxxxxxxx	2.000.000

HD005

Số_HD	Ngày_HD	Diễn_giải	Trị_giá
Hd15000	1/1/2005	Zzzzzzzzz	1.400.000
Hd15001	2/1/2005	Tttttttttt	2.100.000

HD006

Số_HD	Ngày_HD	Diễn_giải	Trị_giá
Hd30000	1/1/2006	Uuuuuuuu	12.000.000
Hd30001	2/1/2006	Vvvvvvvv	1.000.000

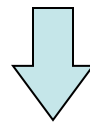
3. Thiết kế vật lý dữ liệu – Phân chia dữ liệu (tt)

b) Phân chia theo chiều dọc (vertical partition)

Cách làm: Phân chia một cấu trúc luận lý thành những cấu trúc lưu trữ vật lý khác nhau.

Ví dụ:

KHÁCH HÀNG(Mã_KH, Tên_KH, Địa_chỉ, SDT, Số_tài_khoản,
Mã_số_thuế, PTTT, Mức_nợ, Công_nợ)



KH1(Mã_KH, Tên_KH, Địa_chỉ)

Cấu trúc truy cập thường xuyên

KH2(Mã_KH, SDT, Số_tài_khoản,
Mã_số_thuế, PTTT, Mức_nợ, Công_nợ)

Cấu trúc truy cập không thường xuyên

3. Thiết kế vật lý dữ liệu – Gộp dữ liệu

- **Mục tiêu:** tối ưu hóa truy vấn dữ liệu.
- **Hạn chế:**
 - Phát sinh trùng lặp dữ liệu.
 - Chi phí kiểm soát tính nhất quán dữ liệu.

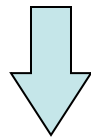
3. Thiết kế vật lý dữ liệu – Gộp dữ liệu (tt)

- Gộp 2 quan hệ liên kết 1-1:

SINH_VIÊN(Mã_SV, Tên_SV, Chuyên_ngành)

HỒ_SƠ_HỌC_BỔNG(Mã_HS, Ngày_lập, Khả_năng, Mã_SV)

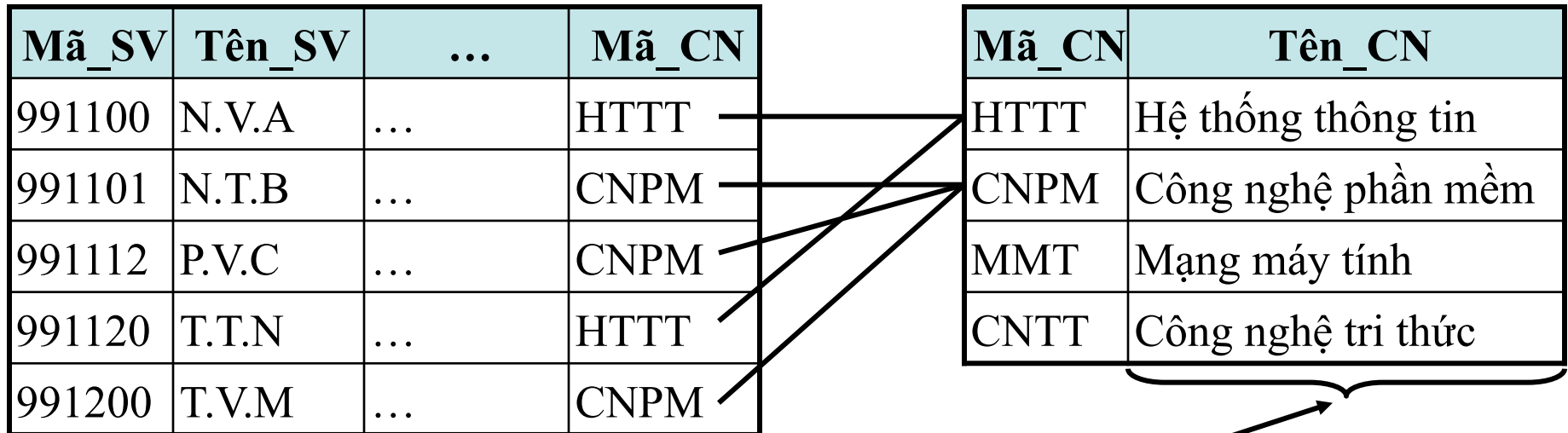
Xử lý	Dữ liệu liên quan
O1	Mã_SV, Tên_SV, Chuyên_ngành, Khả_năng
O2	Tên_SV, Khả_năng, Ngày_lập



SINH_VIÊN(Mã_SV, Tên_SV, Chuyên_ngành, Ngày_lập, Khả_năng)

3. Thiết kế vật lý dữ liệu – Gộp dữ liệu (tt)

- Gộp 2 quan hệ liên kết 1-n:



Truy vấn thường xuyên:

- Q1 (Mã_SV, Tên_SV, Tên_CN)

3. Thiết kế vật lý dữ liệu – Gộp dữ liệu (tt)

- Gộp 2 quan hệ liên kết 1-n (tt)

Mã_SV	Tên_SV	...	Mã_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

Trùng lặp thông tin

Cấu trúc gộp sẽ tối ưu cho Q1 nhưng dẫn đến trùng lặp thông tin

3. Thiết kế vật lý dữ liệu – Tổ chức file chỉ mục

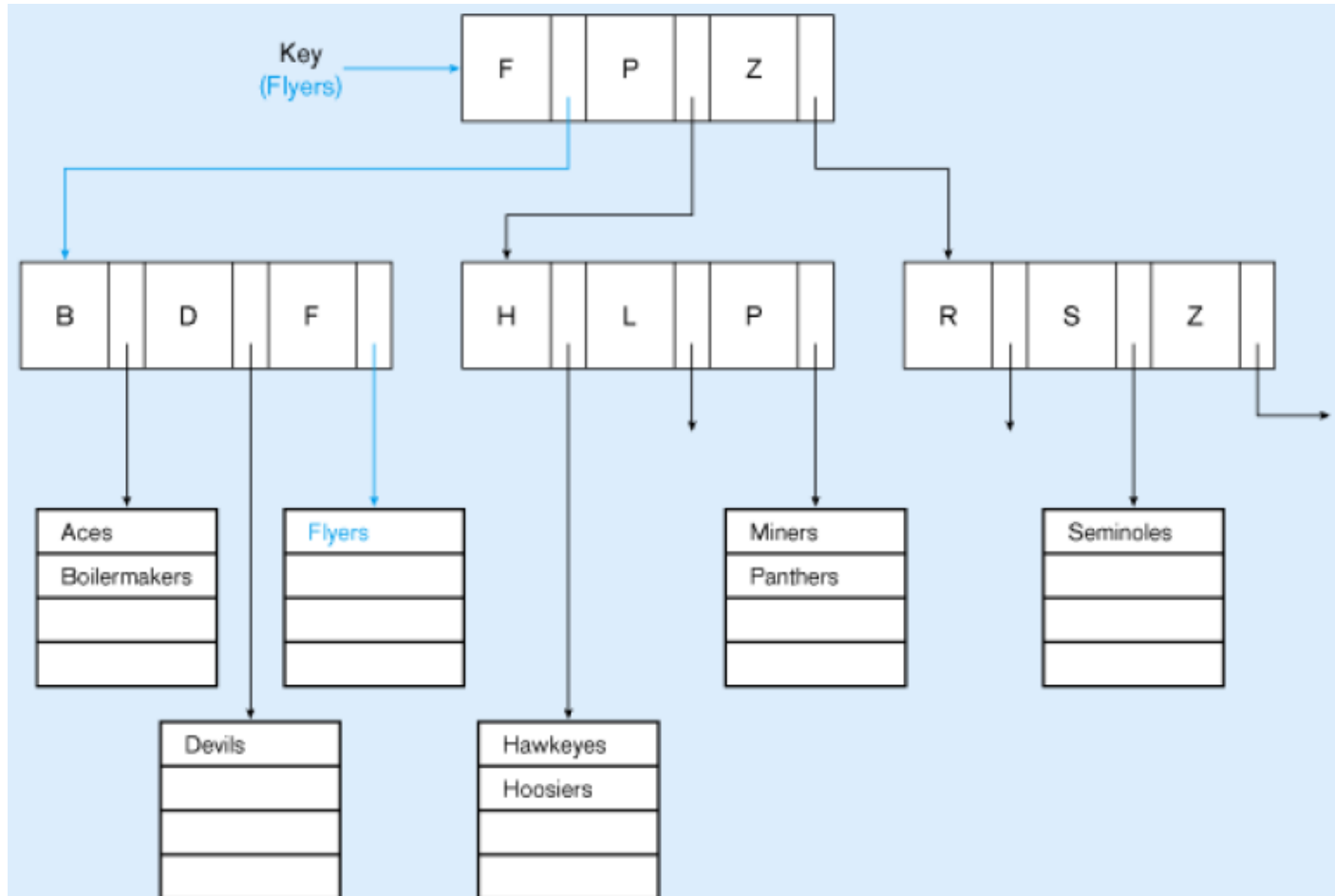
Các trường hợp nên sử dụng chỉ mục:

- Cấu trúc table lớn;
- Đánh chỉ mục khóa chính mỗi table;
- Đánh chỉ mục các column tìm kiếm, các column trong mệnh đề ORDER BY, GROUP BY.

Lưu ý: hạn chế sử dụng chỉ mục trong các CSDL hay biến đổi vì các thay đổi (thêm, xóa, sửa) sẽ bị chậm đi do phải cập nhật lại chỉ mục.

3. Thiết kế vật lý dữ liệu – Tổ chức file chỉ mục (tt)

B-Tree



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

3. Thiết kế vật lý dữ liệu – Tổ chức file chỉ mục (tt)

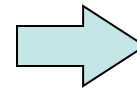
Join index

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
...		

3. Thiết kế vật lý dữ liệu – Tổ chức file chỉ mục (tt)

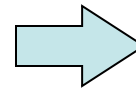
Join index

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
...		