

# **PHÂN TÍCH, THIẾT KẾ HỆ THỐNG THÔNG TIN**

## **BÀI 04. THIẾT KẾ DỮ LIỆU**

Phù Khắc Anh

# TÓM TẮT NỘI DUNG

- I. Thiết kế dữ liệu cấp cao
- II. Thiết kế dữ liệu cấp thấp
- III. Thiết kế mã
- IV. Thiết kế dữ liệu vật lý
- V. Bài tập

# I. Thiết kế dữ liệu cấp cao

- Giúp sinh viên nhận xác định được cách thiết kế dữ liệu cấp cao, độc lập với mô hình cài đặt và dùng chung cho nhiều mô hình dữ liệu, chuyển đổi tổng quát hóa và chọn lựa khóa chính

## II. Thiết kế dữ liệu cấp thấp

- Giúp sinh viên nhận định được cách chuyển đổi kết quả của thiết kế luận lý cấp cao sang một mô hình dữ liệu nhất định



### III. Thiết kế mã

- Xác định cách thiết kế mã cho dữ liệu

### IV. Thiết kế dữ liệu vật lý

- Chuyển các mô tả dữ liệu luận lý sang các đặc tả kỹ thuật nhằm lưu trữ và truy xuất dữ liệu





# V. BÀI TẬP

# I. Thiết kế dữ liệu

- Chuyển đổi tổng quát hóa
- Lựa chọn tập con

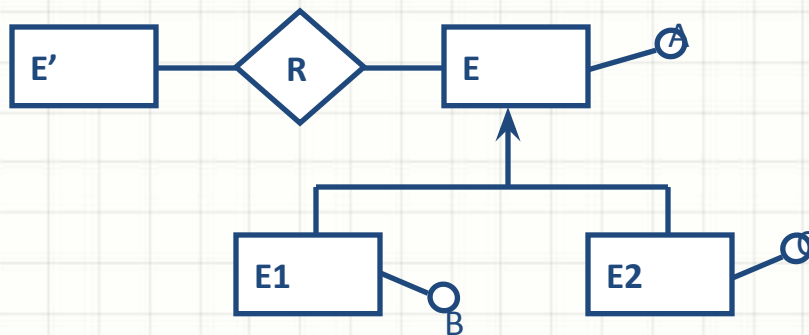
# Chuyển đổi tổng quát hóa & tập con

- Cần thiết khi
  - Mô hình cài đặt không hỗ trợ
    - Mô hình quan hệ, mạng, phân cấp...
- Xem xét
  - Các đặc trưng kế thừa của thực-thể-chuyên-biệt từ thực-thể-tổng-quát
    - Thuộc tính, định danh, mối kết hợp
  - Mối kết hợp tổng quát hóa



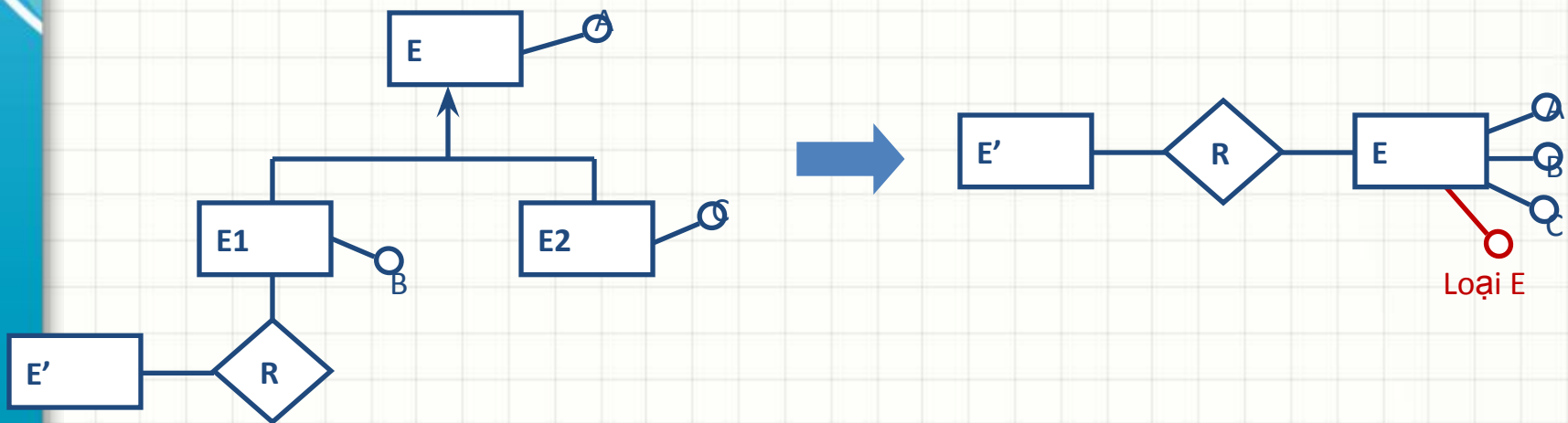
# Chuyển đổi tổng quát hóa & tập con

- Phương án chọn lựa
  - Dùng thực thể tổng quát
  - Dùng thực thể chuyên biệt
  - Dùng mối kết hợp

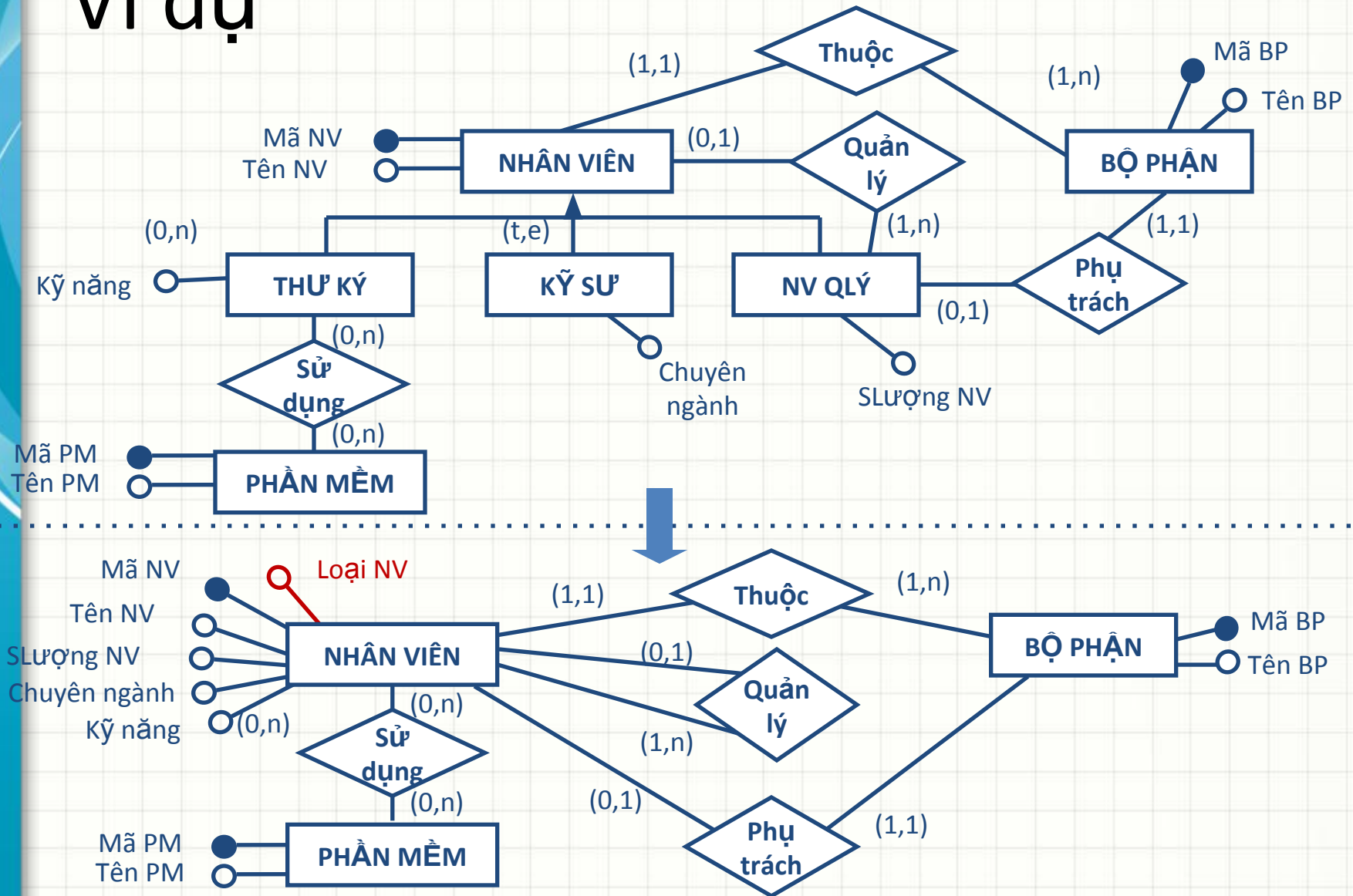


# Dùng thực thể tổng quát

- Các đặc trưng của thực-thể-chuyên-biệt sẽ được chuyển sang thực-thể-tổng-quát
- Thêm thuộc tính LoạiE
  - $MGT(\text{LoạiE}) = \{E, E1, E2, E1E2\}$



# Ví dụ

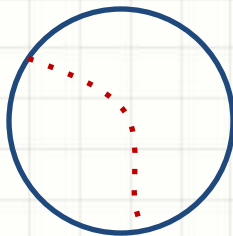
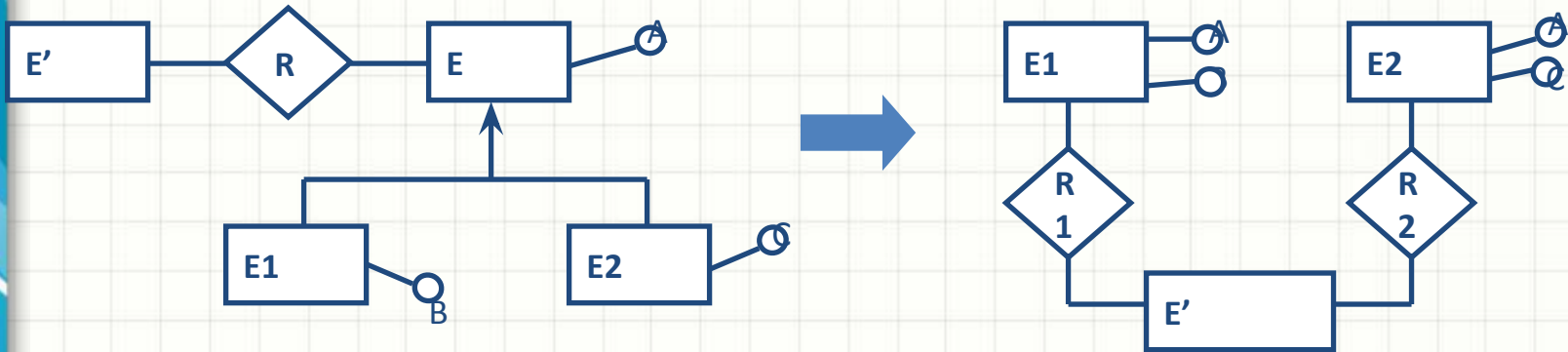


# Dùng thực thể tổng quát (tt)

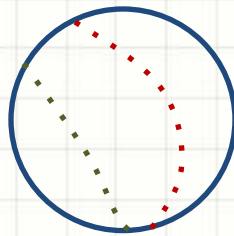
Ưu điểm	Khuyết điểm
Giải pháp đơn giản nhất, không phát sinh thêm các mối kết hợp	Có thể phát sinh ra 1 số lớn các giá trị rỗng cho các thuộc tính mà chỉ dùng cho một loại thực thể tập con mà thôi
Áp dụng cho tất cả các cấu trúc tổng quát hóa như toàn bộ (t) và bán phần (p), chồng lấp (o) và riêng biệt (e)	<p>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</p> <p>Phát sinh thêm một số RBTV cần phải kiểm tra</p>

# Dùng thực thể chuyên biệt

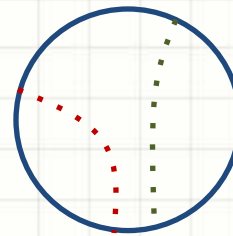
- Chuyển đổi tất cả đặc trưng của thực-thể-tổng-quát xuống các thực-thể-chuyên-biệt



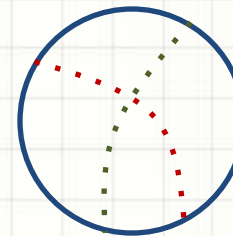
(t,e)



(t,e)

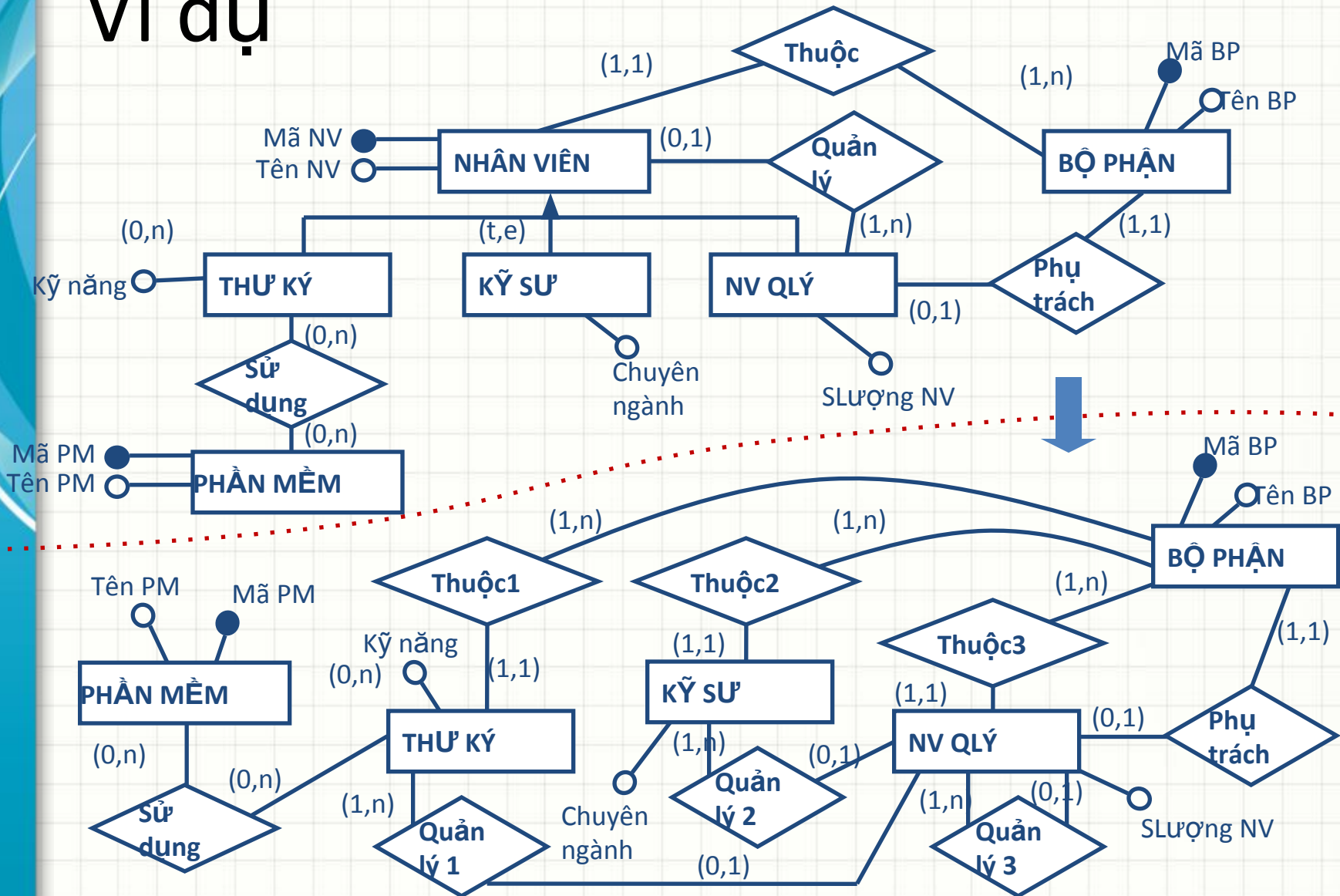


(t,e)



(t,e)

# Ví dụ

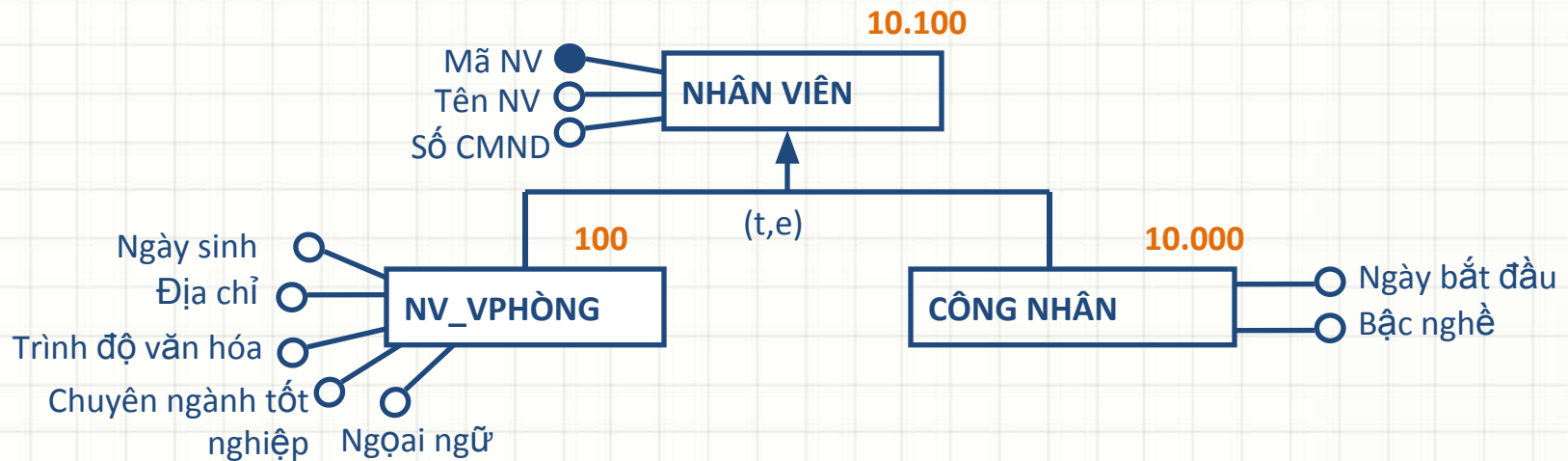




# Dùng thực thể chuyên biệt (tt)

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	<p>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</p> <p>Nếu thuộc tính của thực thể tổng quát là đáng kể thì sự lặp lại trong lược đồ đáng xem xét lại</p> <p>Các tác vụ trước đây thao tác lên thực thể tổng quát nay phải thao tác lên tất cả các thực thể tập con</p>

# Lựa chọn chuyên biệt | tổng quát



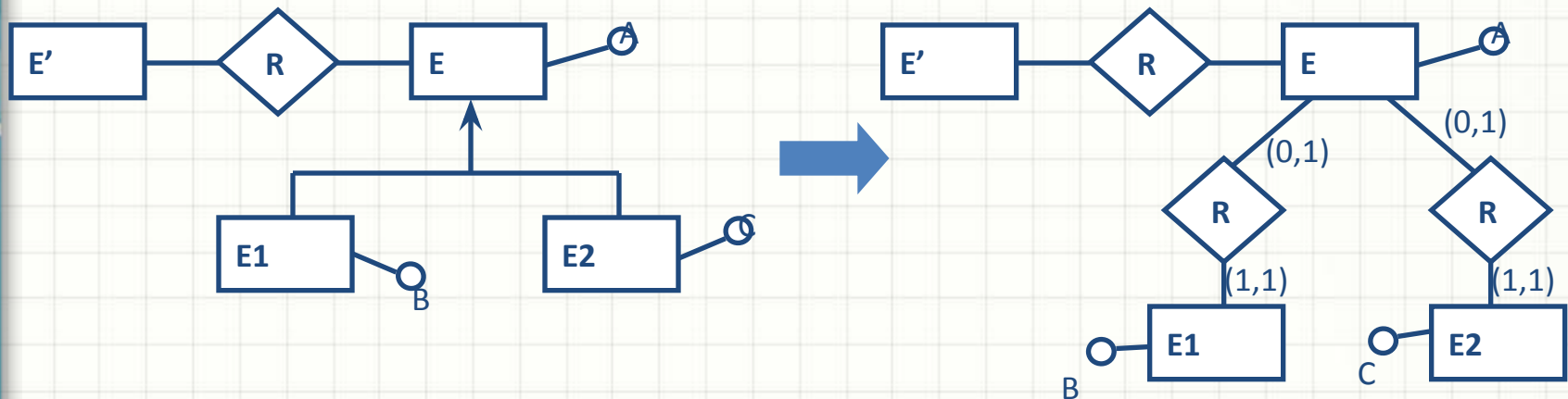
Xử lý	Tần suất	Chọn lựa tối ưu
(o1) Tính lương sản phẩm cho công nhân phân xưởng	2/ tháng	Tách (gộp)
(o2) Tính lương cho nhân viên văn phòng	1/tháng	Tách
(o3) Tìm kiếm thông tin về công nhân	1000/ngày	Tách (gộp)
(o4) Tổng hợp danh sách chung của toàn bộ nhân viên	5/tháng	Gộp
(o5) Truy xuất thông tin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

# Lựa chọn chuyên biệt | tổng quát

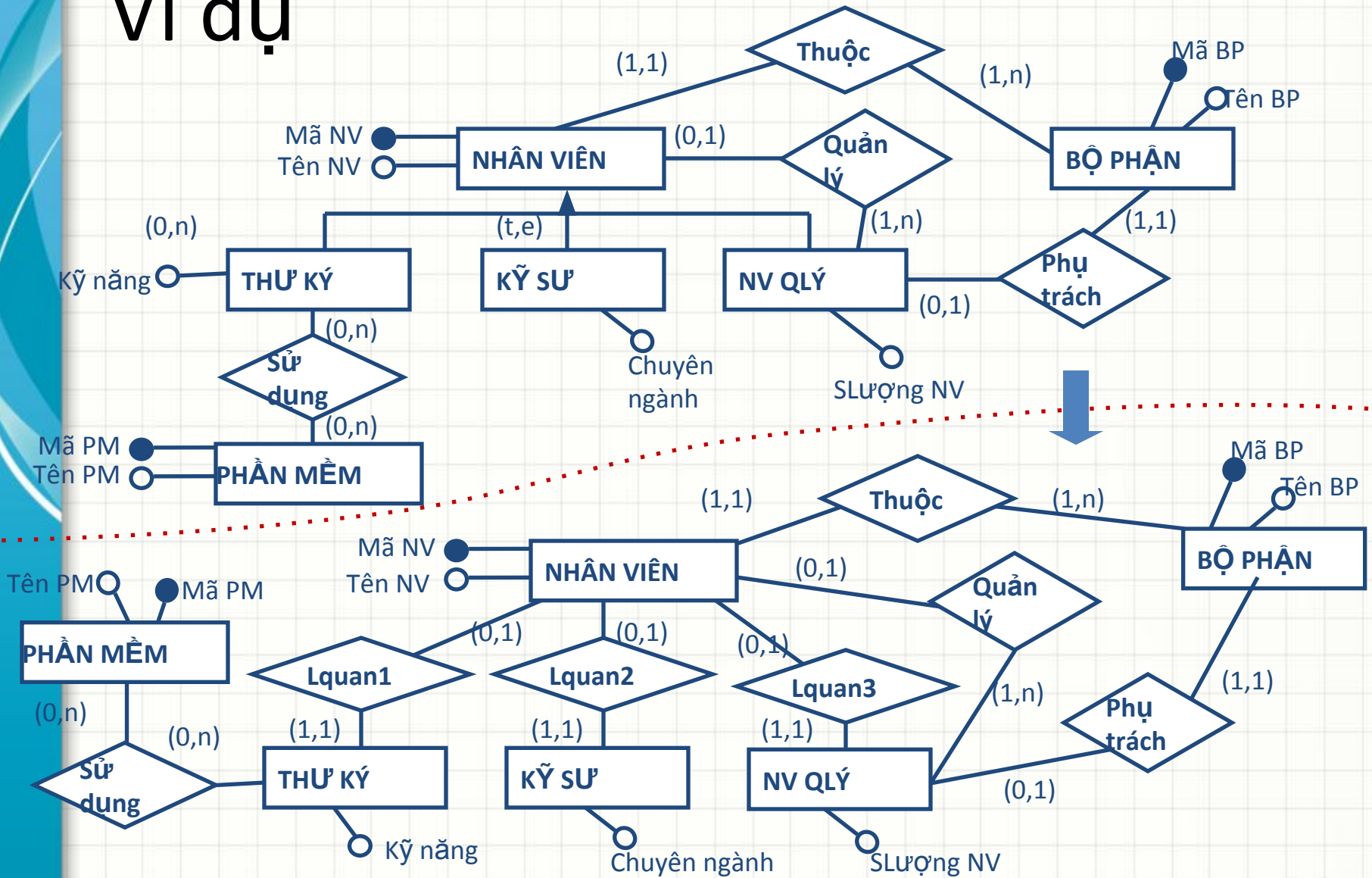
- Phụ thuộc
  - Dung lượng
    - Gộp: dung lượng lớn
    - Tách: dung lượng tối ưu
  - Xử lý
    - Ưu tiên các xử lý có tần suất cao
    - Ví dụ
      - Nếu ưu tiên o4 : gộp
      - Nếu ưu tiên o2, o3: tách

# Dùng mối kết hợp

- Bỏ tổng quát, chuyên biệt
- Tạo mối kết hợp từ thực-thể-tổng-quát đến thực-thể-chuyên-biệt



# Ví dụ



# Dùng mối kết hợp (tt)

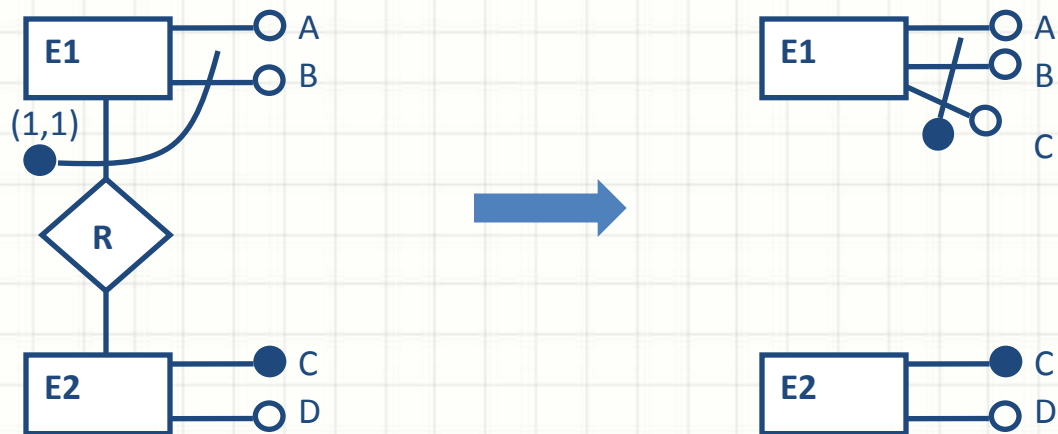
Ưu điểm	Khuyết điểm
Có thể mô hình tất cả các loại cấu trúc tổng quát hóa toàn bộ / bán phần và chồng chéo / riêng biệt	Lược đồ kết quả khá phức tạp. Ví dụ như thêm một thể hiện cho một thực thể tập con phải thêm mới một thể hiện cho quan hệ và một thể hiện cho thực thể tổng quát hóa
Rất yên chuyên khi thay đổi yêu cầu của ứng dụng	Phải chấp nhận sự dư thừa khi biểu diễn mối liên kết IS-A thành mối kết hợp



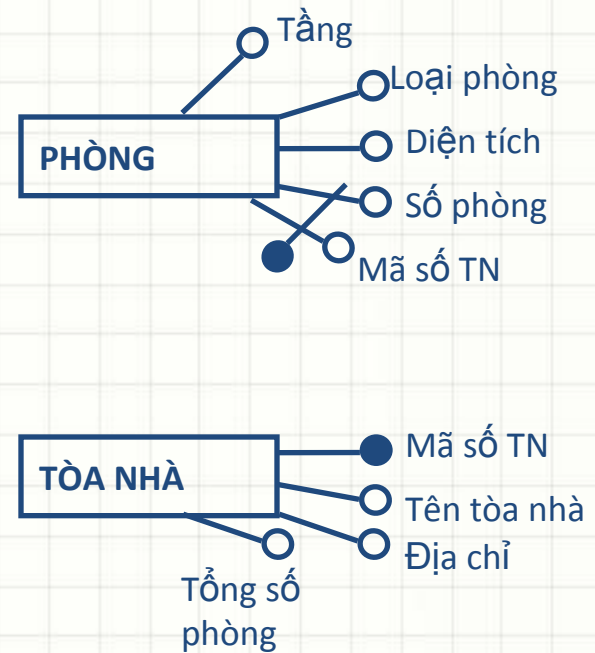
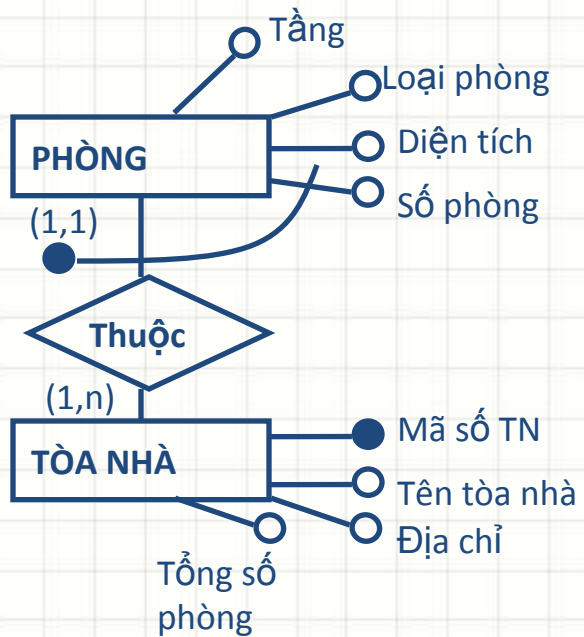
## II. Thiết kế luận lý dữ liệu cấp thấp

- Chuẩn bị
  - Loại bỏ định danh ngoài
  - Loại bỏ thuộc tính đa trị và thuộc tính kết hợp
- Chuyển đổi mô hình thực thể kết hợp sang mô hình quan hệ

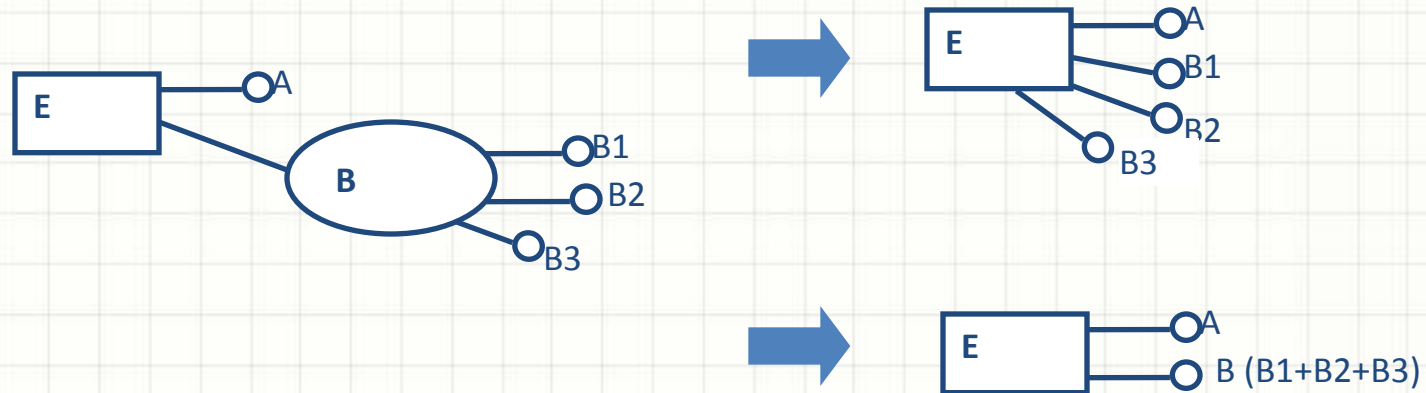
# Loại bỏ định danh bên ngoài



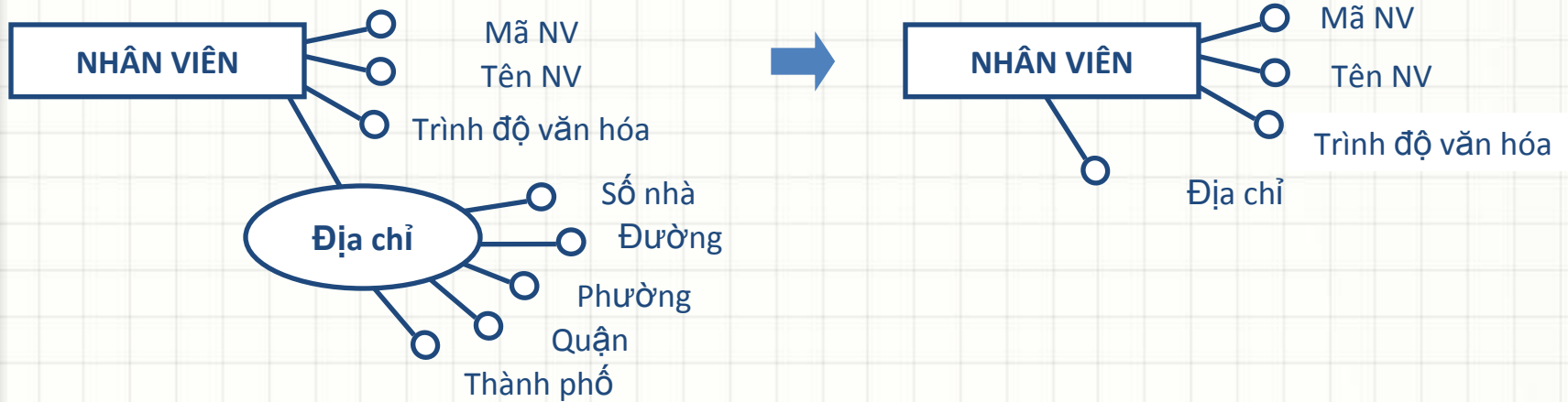
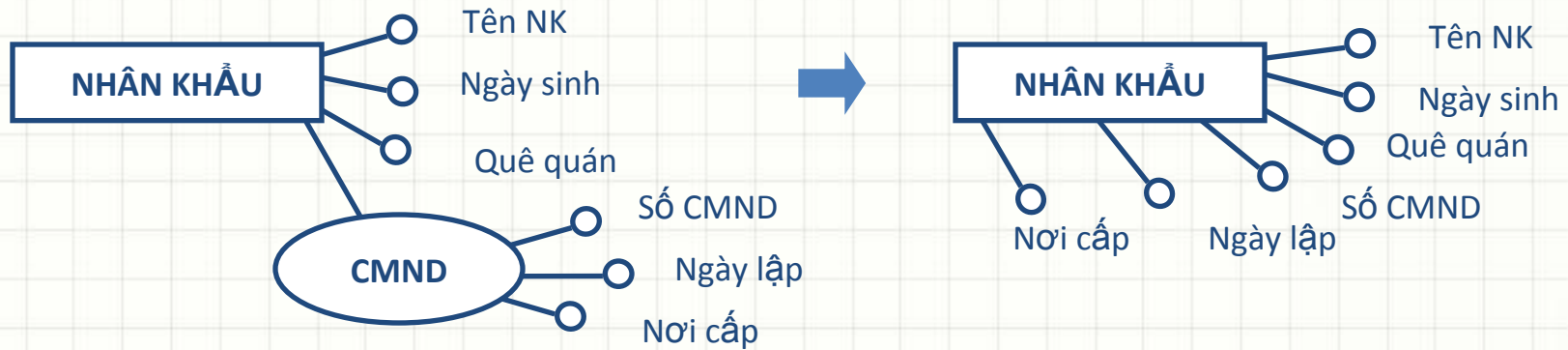
# Ví dụ



# Loại bỏ thuộc tính kết hợp

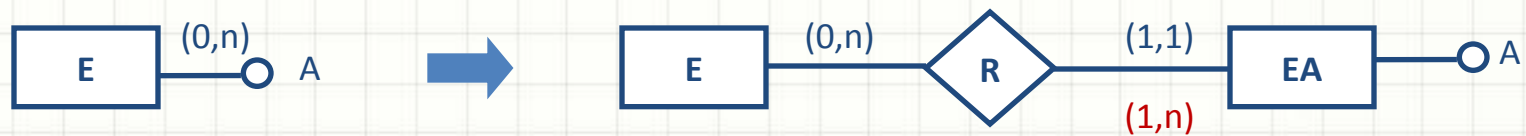


# Ví dụ



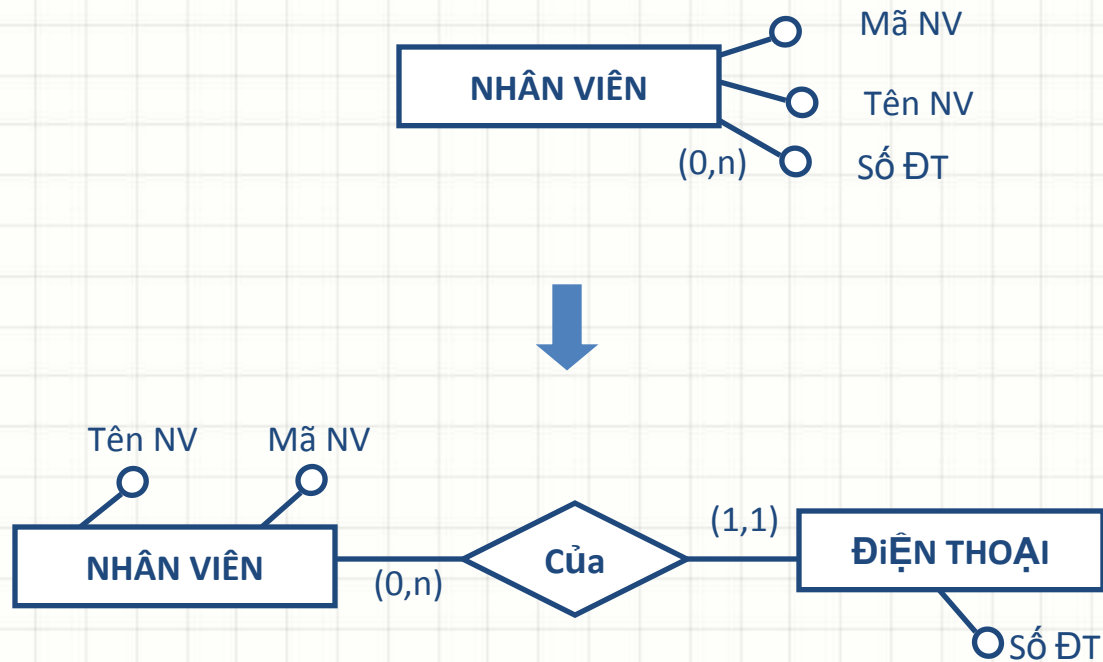
# Loại bỏ thuộc tính đa trị

- Thuộc tính đa trị ở thực thể



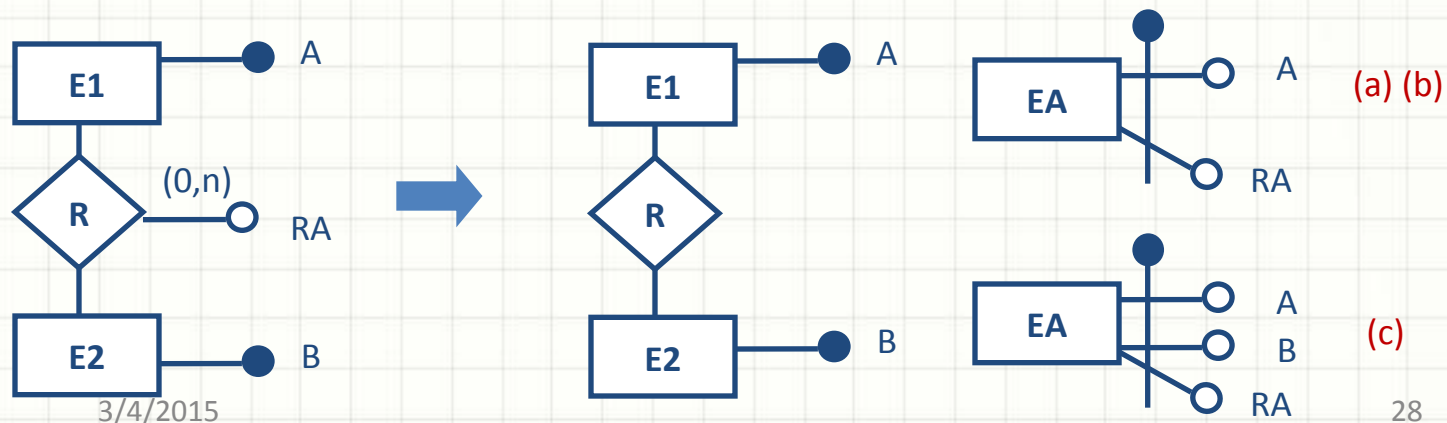


# Ví dụ

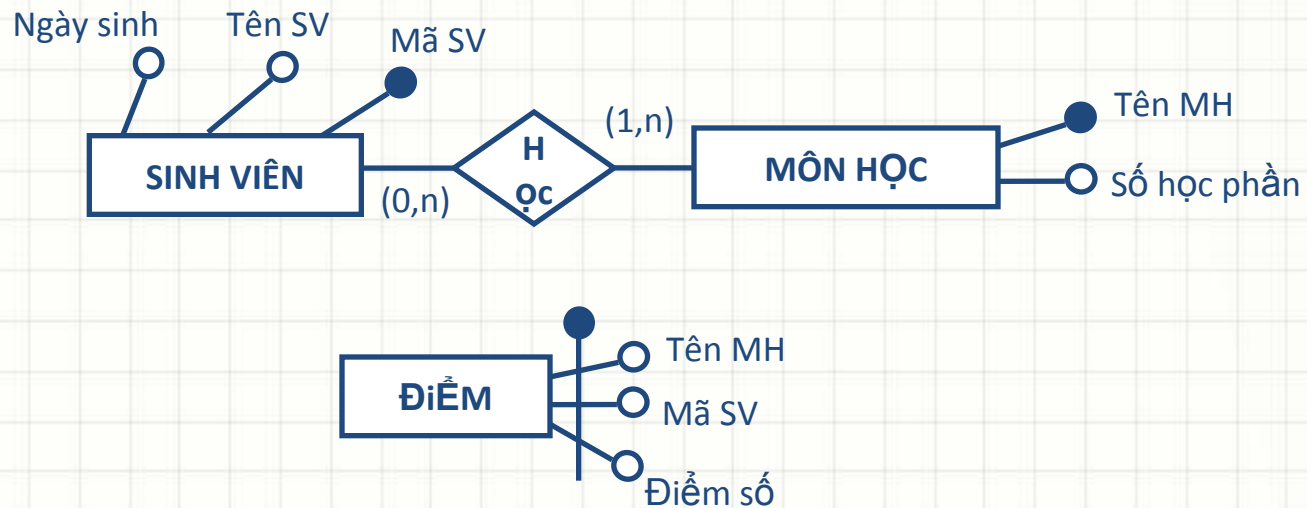
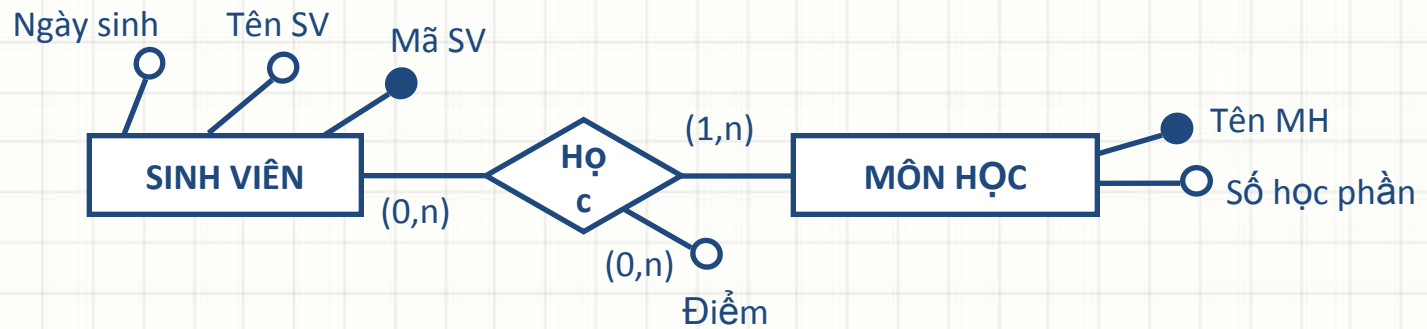


# Loại bỏ thuộc tính đa trị (tt)

- Thuộc tính đa trị ở mỗi kết hợp
  - (a) R là mỗi kết hợp 1-1
    - EA sẽ có định danh của E1 hoặc E2 kết hợp với RA
  - (b) R là mỗi kết hợp 1-N
    - EA sẽ có định danh của E1 kết hợp với RA
  - (c) R là mỗi kết hợp N-N
    - EA sẽ có định danh của E1 và E2 kết hợp với RA



# Ví dụ

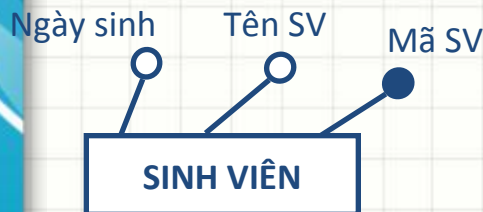


# Chuyển mô hình ER $\rightarrow$ Quan hệ

- Chuyển đổi thực thể
- Chuyển đổi mối kết hợp

# Chuyển đổi thực thể

- 1 thực thể → 1 quan hệ
- Thuộc tính → thuộc tính
- Định danh → khóa chính



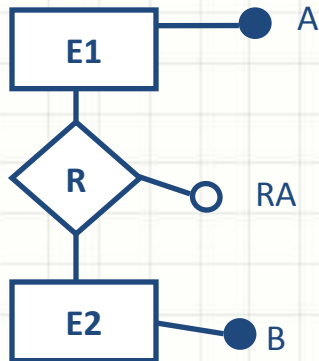
**SINH\_VIÊN**(MA\_SV, TEN\_SV, NGÀY\_SINH)



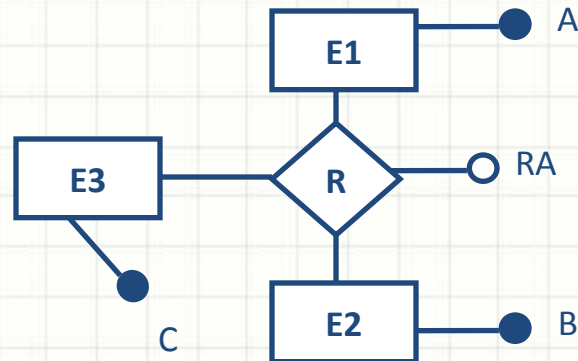
**MÔN\_HỌC**(TÊN\_MH,  
SỐ\_HỌC\_PHẦN)

# Chuyển đổi mối kết hợp

- Quy tắc chung



$R(\underline{A}, B, RA)$

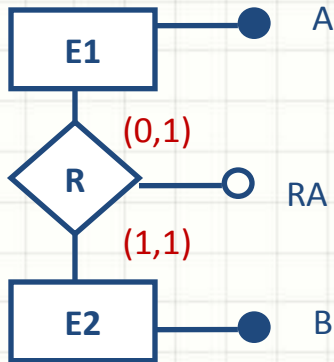


$R(\underline{A}, B, C, RA)$

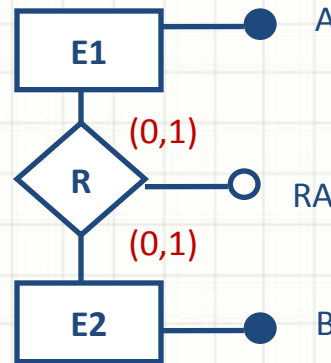


# Chuyển đổi mối kết hợp (tt)

- Mối kết hợp 1-1



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

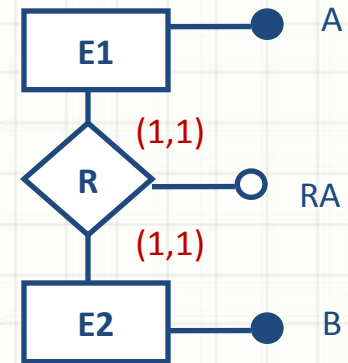


$E1(\underline{A}, \dots)$   
 $E2(\underline{B}, \dots, A) : A$  là  
 khóa

Hoặc

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

$E2(\underline{B}, \dots)$

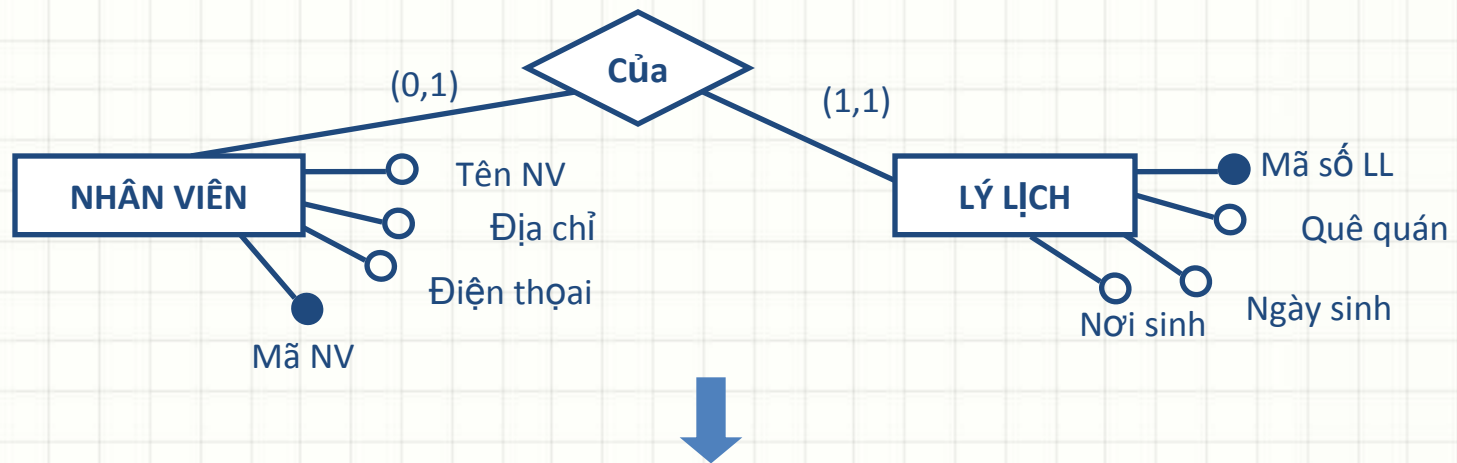


$E1E2(\underline{A}, B, \dots) : A$  là khóa  
 $B$  là thuộc tính

Hoặc

$E1E2(A, \underline{B}, \dots) : B$  là khóa  
 $A$  là thuộc tính

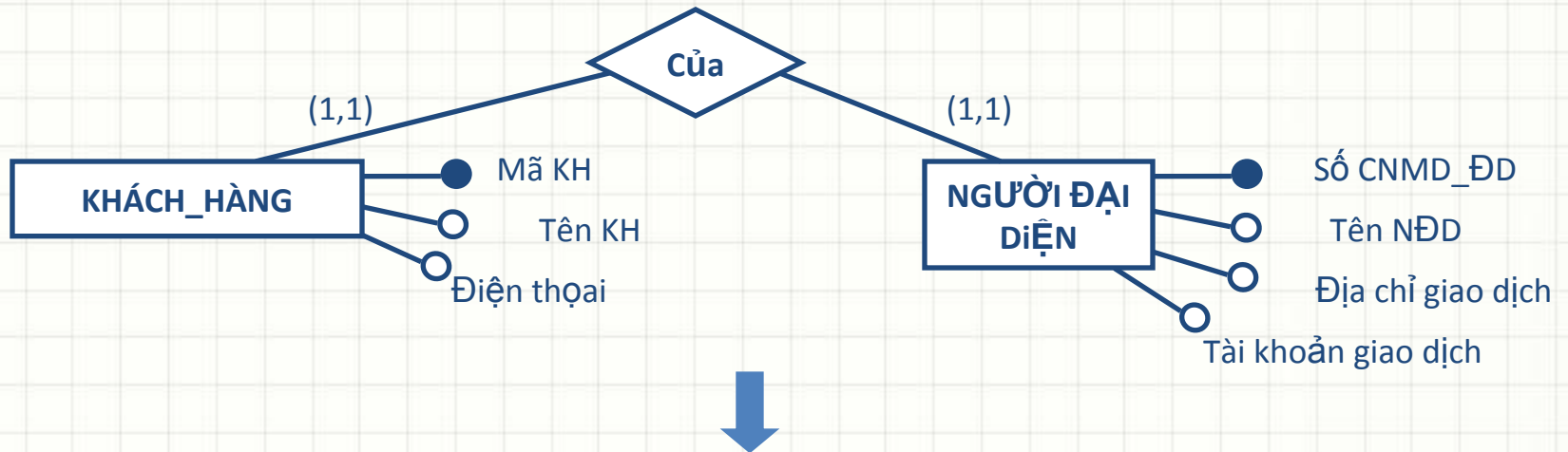
# Ví dụ



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

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

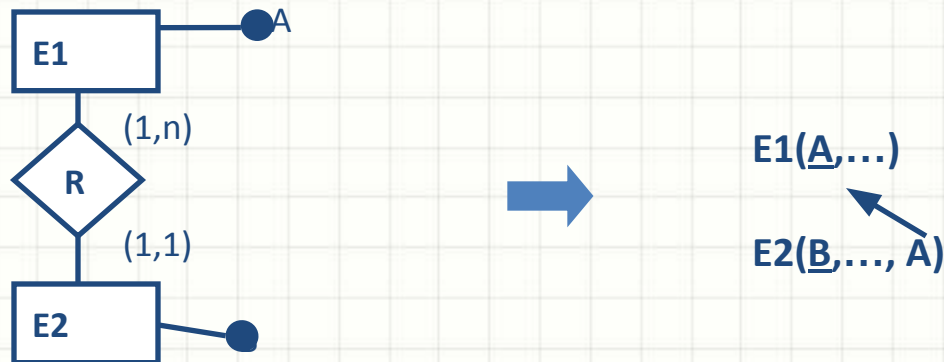
# Ví dụ (tt)



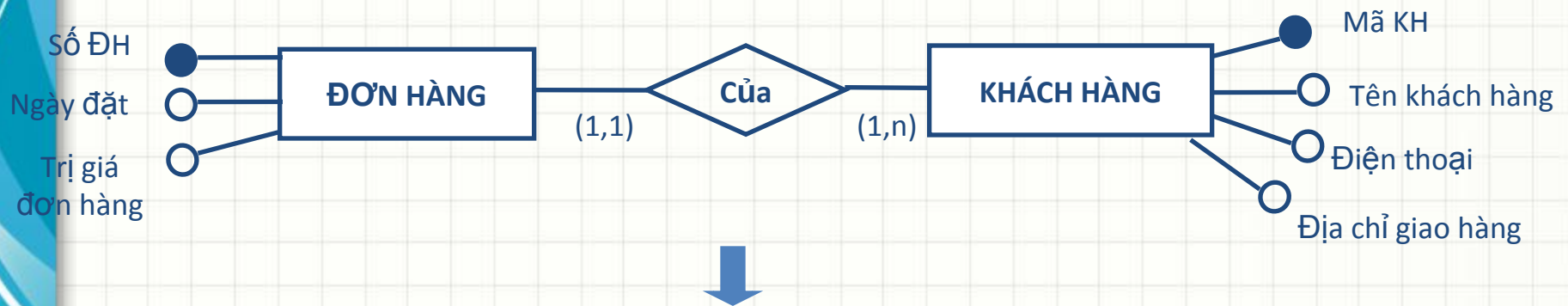
**KH\_NĐDIỆN (MÃ\_KH, TÊN\_KH, ĐIỆN\_THOẠI, SỐ\_CMND\_ĐD, TÊN\_NĐD, ĐỊA\_CHỈ\_GIAO\_DỊCH, TÀI KHOẢN\_GIAO\_DỊCH)**

# Chuyển đổi mối kết hợp (tt)

- Mối kết hợp 1-N



# Ví dụ

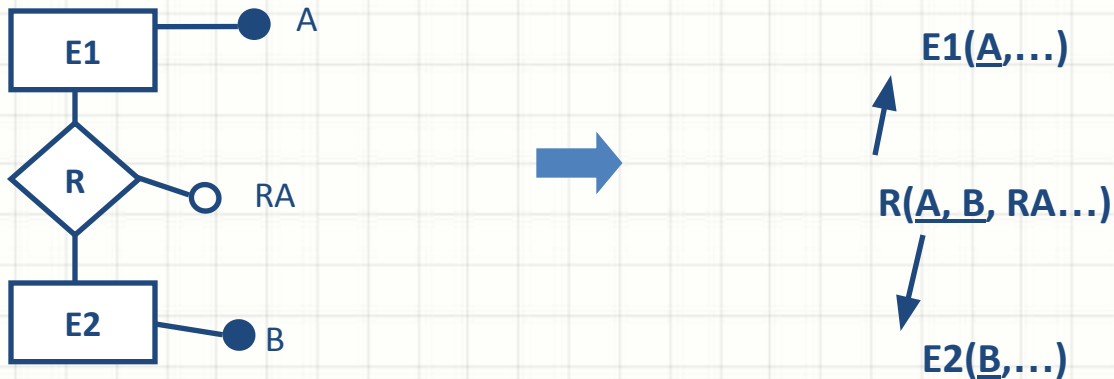


**KHÁCH HÀNG**(MA\_KH, TEN\_KH, ĐIỆN THOẠI,  
ĐỊA CHỈ\_GH)

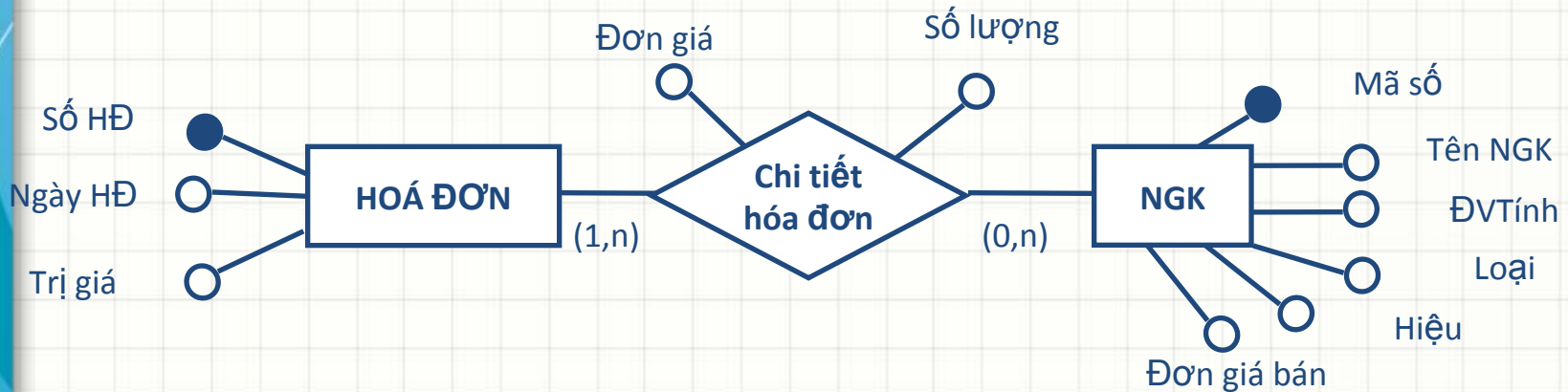
**ĐƠN HÀNG**(SO\_ĐH, NGÀY\_ĐẶT, TRỊ\_GIÁ\_ĐH, MA\_KH)

# Chuyển đổi mối kết hợp (tt)

- Mối kết hợp N-N



# Ví dụ



HÓA ĐƠN(SỐ\_HD, NGÀY\_HD, TRỊ GIÁ)

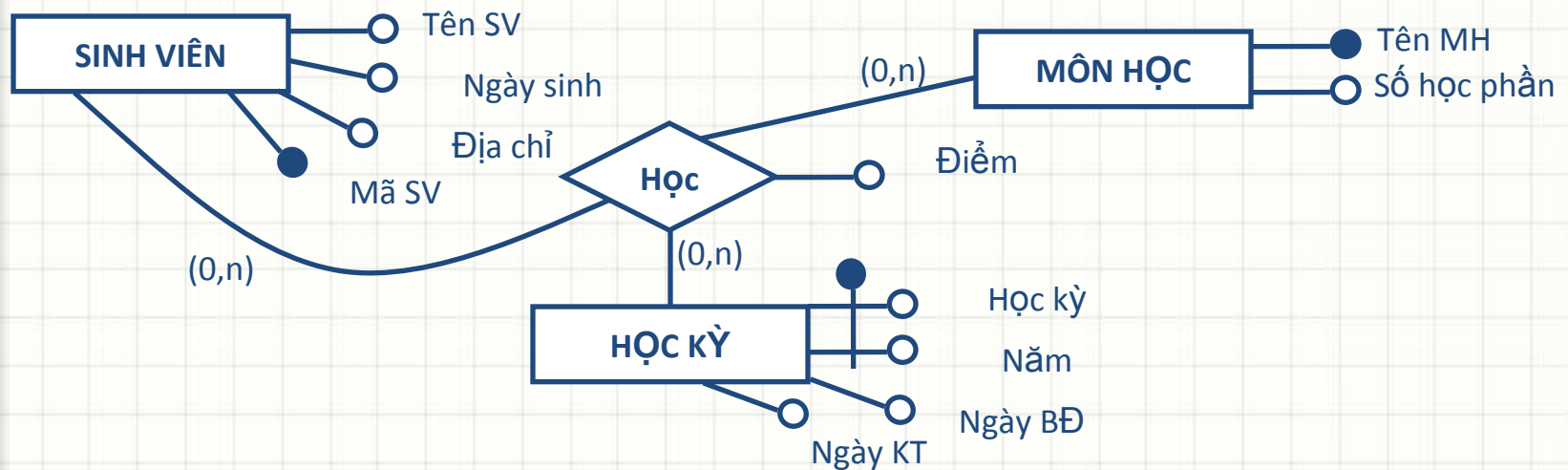
CHITIẾT\_HD(SỐ\_HD, MÃ SỐ, SỐ LƯỢNG, ĐƠN GIÁ)

NGK(MÃ SỐ, TÊN\_NGK, ĐVTÍNH, LOẠI, HIỆU, ĐƠN GIÁ\_BÁN)



# Chuyển đổi mối kết hợp (tt)

- Mối kết hợp đa phân



**SINH\_VIÊN**(MÃ\_SV, TÊN\_SV, NGÀY\_SINH, ĐỊA\_CHỈ)

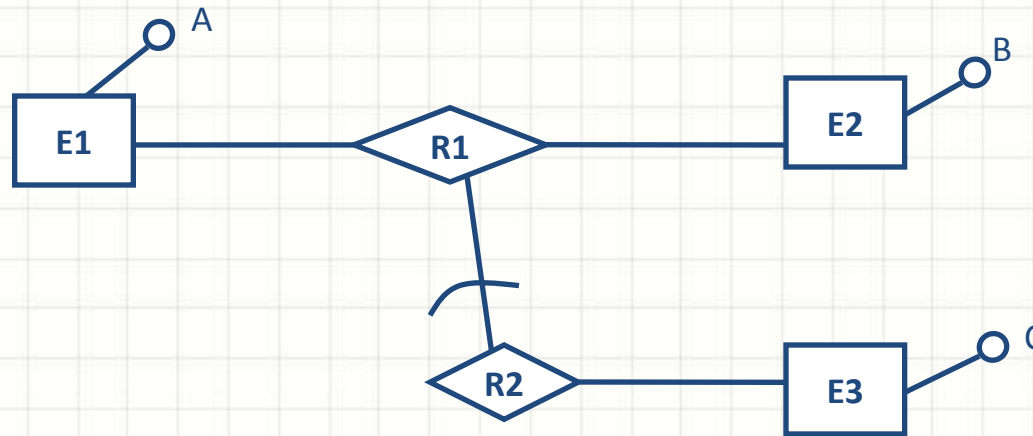
**MÔN\_HỌC**(TÊN\_MH, SỐ\_HỌC\_PHẦN)

**HỌC\_KỲ**(HỌC\_KỲ, NĂM, NGÀY\_BĐ, NGÀY\_KT)

**HỌC**(MÃ\_SV, TÊN\_MH, HỌC\_KỲ, NĂM, ĐIỂM)

# Chuyển đổi mỗi kết hợp (tt)

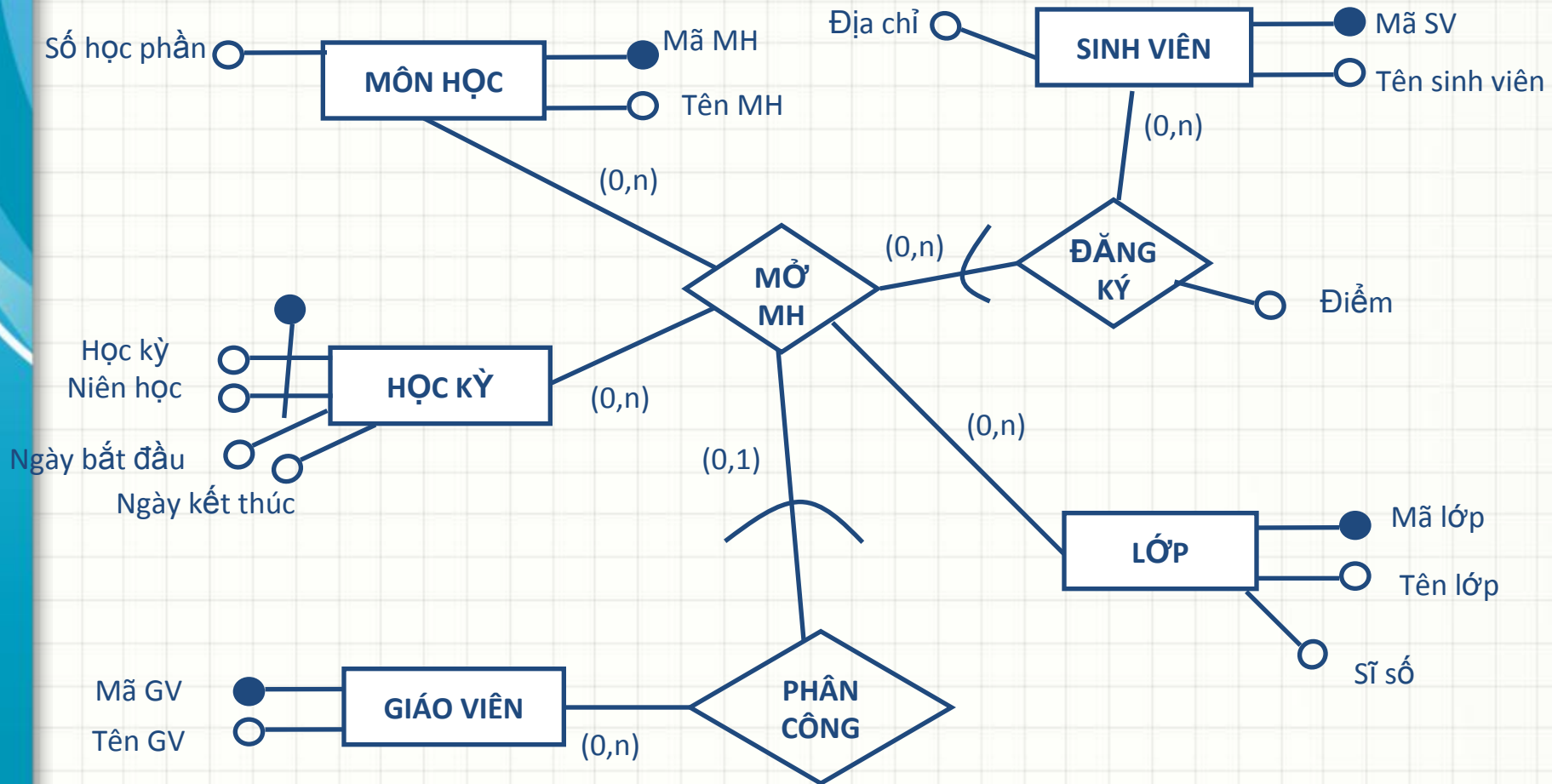
- Mỗi kết hợp mở rộng



$R1(\underline{A}, \underline{B}, \dots)$

$R2(\underline{A}, \underline{B}, \underline{C}, \dots)$

# Ví dụ



# Ví dụ (tt)

MÔN\_HỌC(MA\_MH, TÊN\_MH, SỐ\_HP)

HỌC\_KỲ(HOC\_KỲ, NIÊN\_HOC, NGÀY\_BĐ, NGÀY\_KT)

GIÁO\_VIÊN(MÃ\_GV, TÊN\_GV)

SINH\_VIÊN(MÃ\_SV, TÊN\_SV)

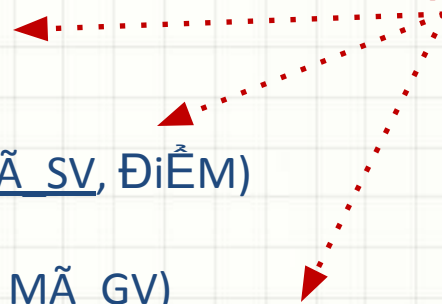
LỚP(MÃ\_LỚP, TÊN\_LỚP, SĨ\_SỐ)

MỞ\_MH(MÃ\_MH, MÃ\_LỚP, HOC\_KỲ, NIÊN\_HOC)

ĐĂNG\_KÝ(MÃ\_MH, MÃ\_LỚP, HOC\_KỲ, NIÊN\_HOC, MÃ\_SV, ĐIỂM)

PHÂN\_CÔNG(MÃ\_MH, MÃ\_LỚP, HOC\_KỲ, NIÊN\_HOC, MÃ\_GV)

Khóa chính quá  
phức tạp



# III. Thiết kế mã

- Tìm các hình thức mã hóa
  - Mã quận, mã bưu điện
  - Mã xe
  - Mã sinh viên
  - ...
- Dễ dàng cho việc
  - Quản lý và xử lý thông tin trong máy tính
  - Nhận dạng và phân loại dữ liệu
- Phải được thiết kế trên cơ sở uyển chuyển và có khả năng mở rộng

# Thiết kế mã (tt)

- Ngữ nghĩa của mã hóa
  - Duy nhất
    - Mã phải duy nhất để nhận dạng đối tượng dữ liệu
  - Phân loại
    - Phân loại dữ liệu thành các nhóm khác nhau, dựa trên mã phân biệt được nhóm
  - Sắp xếp
    - Mã hóa thể hiện thứ tự của dữ liệu
  - Kiểm tra
    - Kiểm soát dữ liệu có được nhập đúng hay không

# Thiết kế mã (tt)

- Một số đặc điểm
  - Vùng mã hóa: phạm vi mà mã được sử dụng
    - Có liên kết với các hệ thống bên ngoài, mã nên là chuẩn công nghiệp chung
    - Trong cùng 1 đơn vị, mã có phạm vi toàn công ty
  - Chu kỳ sử dụng
    - Ước lượng 1 lượng mã cần đủ trong chu kỳ
  - Ngữ nghĩa
    - Có thể hiểu mã bởi các thành phần liên quan



# Ví dụ vùng mã hóa

- Mã môn học được thiết kế khác nhau trong cùng 1 trường

TH201 - Thiết kế Cơ Sở Dữ Liệu

Khoa CNTT

CNTT201 - Thiết kế Cơ Sở Dữ Liệu

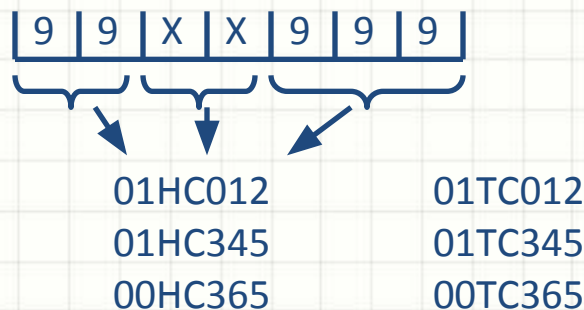
Phòng Đào Tạo

Gây ra 1 số hoạt động không cần thiết như : chuyển đổi mã liên quan đến kết quả học tập trên môn học, ...

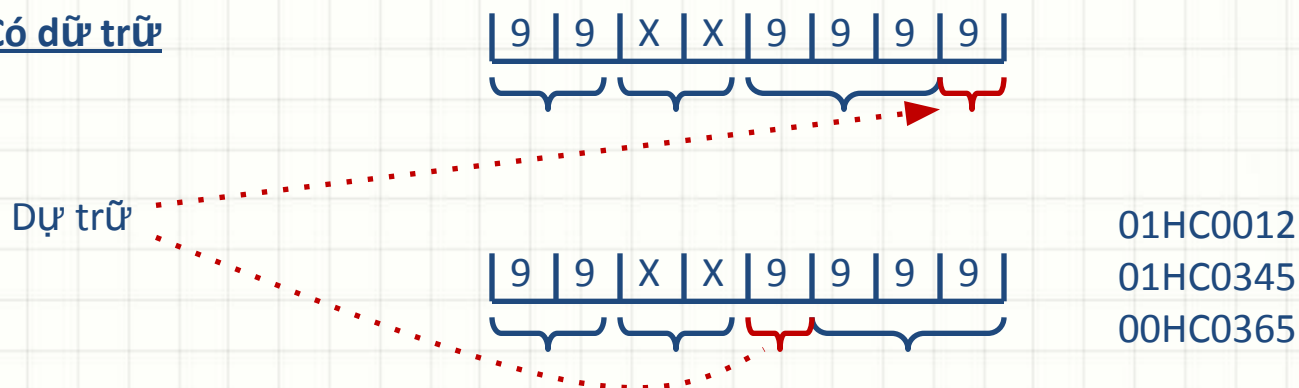
# Ví dụ chu kỳ sử dụng

- Mã số sinh viên

Không dữ trữ

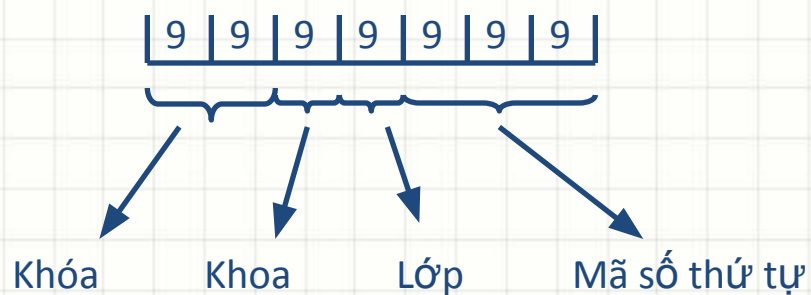


Có dữ trữ



# Ví dụ ngữ nghĩa

- Mã số sinh viên



0011037

0011456

0111230

# Một số 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 được quy 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

# Một số loại mã (tt)

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, ....	<u>Mã hóa phòng ban trong một công ty:</u> 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	Mỗi ký số của mã sẽ gán cho một ngữ nghĩa	<u>Mã SV của trường ĐHKHTN TPHCM</u> <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											

# Một số loại mã (tt)

Tên mã	Mô tả	Ví dụ
Mã gợi nhớ	Chữ viết tắt hoặc biểu tượng của đối tượng được dùng để mã hóa	<u>Mã hóa lớp học của trường đại học như sau:</u> 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	

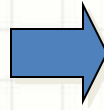
# Thiết kế dữ liệu vật lý

- 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

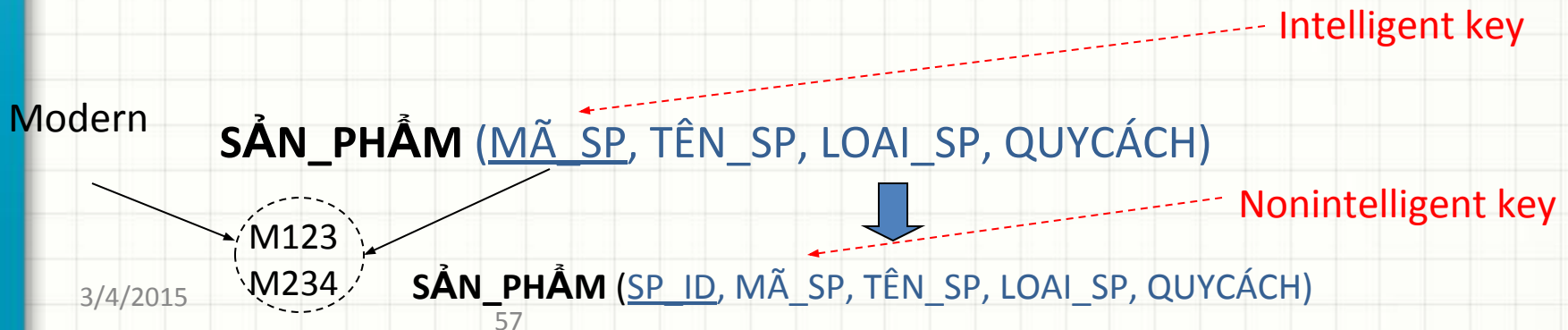
- Thiết kế field
- Phân chia dữ liệu (partition)
- Gộp dữ liệu (demormalization)
- Thiết kế file dữ liệu vật lý

# Thiết kế dữ liệu vật lý

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

# Thiết kế dữ liệu vật lý

- Thiết kế field – Chọn lựa khoá chính:
  - Việc chọn lựa khóa chính trong mô hình luận lý đôi khi không thuận tiện trong cập nhật
  - 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
  - Ví dụ:



# Thiết kế dữ liệu vật lý

- Thiết kế field bao gồm:
  - Ví dụ: **SANPHAM**

SP_ID	Mã_SP	Tên_SP	Loại_SP	Quy cách
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

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

## 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

# 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:

Khoá chính luận lý (intelligent key)

ĐĂNG\_KÝ(MÃ\_MH, MÃ\_LỚP, HỌC\_KỲ, NIÊN\_HOC, MÃ\_SV, ĐIỂM)

Khoá chính mới (nonintelligent key)

ĐĂNG\_KÝ(DKY\_ID, MÃ\_MH, MÃ\_LỚP, HỌC\_KỲ, NIÊN\_HOC, MÃ\_SV, ĐIỂM)

Tổ hợp các thuộc tính này được cài đặt là một khoá



# 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)

– Ví dụ:

**HOA\_DON**

**KL: ~10.000.000/năm**

Số_HD	Ngày_HD	Diễn_giải	Trị giá
Hd00001	1/1/04	Xxxxxxx	1.000.000
Hd00002	2/1/04	Yyyyyyy	2.000.000
....			
Hd15000	1/1/05	Zzxzxxzx	1.400.000
Hd15001	2/1/05	Qqqqqqqq	2.100.000
...			
Hd30000	2/1/06	Asasasas	12.000.000
Hd30001	2/1/06	Dsdsdsds	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

**HOA\_DON**

KL: ~10.000.000/năm

Số_HD	Ngày_HD	Diễn_giải	Trị giá
Hd00001	1/1/04	Xxxxxxx	1.000.000
Hd00002	2/1/04	Yyyyyyy	2.000.000
....			
Hd15000	1/1/05	Zxzxzzxzx	1.400.000
Hd15001	2/1/05	Qqqqqqqq	2.100.000
...			
Hd30000	2/1/06	Asasasas	12.000.000
Hd30001	2/1/06	Dsdsdsds	1.000.000

**HD004**

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

**HD005**

Số_HD	Ngày_HD	Diễn_giải	Trị giá
Hd15000	1/1/05	Zxzxzzxzx	1.400.000
Hd15001	2/1/05	Qqqqqqqq	2.100.000

**HD006**

Số_HD	Ngày_HD	Diễn_giải	Trị giá
Hd30000	2/1/06	Asasasas	12.000.000
Hd30001	2/1/06	Dsdsdsds	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 ký 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)

Cấu trúc truy cập thường xuyên



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

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 lặp 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\_HBÔNG(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

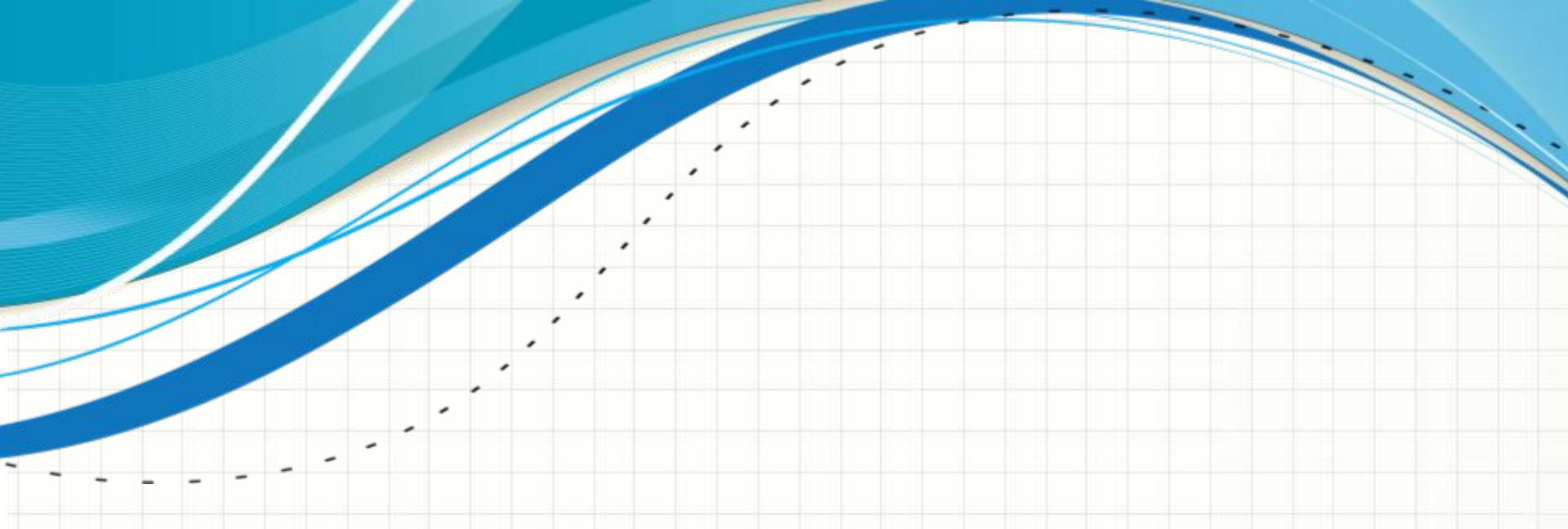
Truy vấn thường xuyên:

- Q1 (Mã\_SV, TÊN\_SV, TÊN\_CN)

Trùng lặp thông tin

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





**CÂU HỎI ?**