

---

## Chương 3

# MÔ HÌNH DỮ LIỆU QUAN HỆ

---

**GIẢNG VIÊN: PHẠM THỊ XUÂN HIỀN**

---

# NỘI DUNG

## 1. Giới thiệu

2. Các khái niệm của mô hình quan hệ
3. Ràng buộc toàn vẹn
4. Các đặc trưng của quan hệ
5. Chuyển lược đồ ER sang thiết kế quan hệ

# Mô hình dữ liệu quan hệ (Relational Data Model)

- Do tiến sĩ E.F.Codd đưa ra “A Relation Model for Large Shared Data Banks Communications of ACM” (06/1970)
- Mô hình này bao gồm:
  - Các khái niệm liên quan đến cấu trúc dữ liệu dưới dạng dòng và cột như quan hệ, bộ, thuộc tính, khoá chính, khoá ngoại ...
  - Được xây dựng trên cơ sở của phép toán tập hợp và phép toán quan hệ
  - Ràng buộc toàn vẹn quan hệ
- Các HQTCSDDL được xây dựng theo lý thuyết mô hình quan hệ
- Là cơ sở của các HQT CSDL thương mại  
IBM, Oracle, SQL Server, Microsoft Access

# NỘI DUNG

1. Giới thiệu
- 2. Các khái niệm của mô hình quan hệ**
3. Ràng buộc toàn vẹn
4. Các đặc trưng của quan hệ
5. Chuyển lược đồ ER sang thiết kế quan hệ

# Các khái niệm của mô hình quan hệ

- Quan hệ (Relation)
- Thuộc tính (Attribute)
- Bộ (Tuple)
- Miền giá trị (Domain)
- Lược đồ (Schema)

# Quan hệ (Relation)

- **Một quan hệ là một bảng (table) 2 chiều**
  - **Mỗi dòng (trừ dòng đầu tiên)** gồm các giá trị thể hiện một thực thể hoặc mối kết hợp trong thực tế
  - **Mỗi tiêu đề cột** cho biết ý nghĩa của từng giá trị trên một dòng (tất cả dữ liệu trên cùng 1 cột đều có cùng kiểu dữ liệu)

Vd: Quan hệ **NhanVien**, mỗi dòng gồm các giá trị thể hiện của **NhanVien**

thuộc tính

MaNV	HoTen	NamSinh	Phai	Luong	PhG
NV01	Nguyễn Trung Tiến	1993	Nam	5 000 000	5
NV02	Trần Thị Yến	1987	Nữ	5 500 000	4

1 dòng là 1 nhân viên

# Quan hệ (Relation) (tt)

- Theo thuật ngữ của mô hình quan hệ:
  - Mỗi **bảng** gọi là một **quan hệ** (relation)
  - Mỗi **dòng** dữ liệu gọi là một **bộ** (tuple)
  - Mỗi **tiêu đề cột** gọi là một **thuộc tính** (attribute)
  - **Tập hợp các giá trị** mà một thuộc tính có thể nhận lấy, gọi là **miền giá trị (domain)** của thuộc tính.

MaNV	HoTen	NamSinh	Phai	Luong	PhG
NV01	Nguyễn Trung Tiến	1993	Nam	5 000 000	5
NV02	Trần Thị Yến	1987	Nữ	5 500 000	4

**2 Bộ, 6 Thuộc tính**

# Miền giá trị của thuộc tính (domain)

- Một miền giá trị  $D$  là một tập hợp các giá trị nguyên tố (atomic value) (Giá trị nguyên tố là **giá trị không thể chia nhỏ**)
- **Để mô tả một miền giá trị  $D$ :** người ta thường dùng kiểu dữ liệu và định dạng của các giá trị trong  $D$ 
  - Kiểu dữ liệu cơ bản: chuỗi ký tự, số (số nguyên, số thực), date
  - Kiểu dữ liệu phức tạp: set, list, array, record => **Không chấp nhận**

## Ví dụ:

- **MaSV**: tập hợp chuỗi 7 ký tự
- **Diem**: số thực thuộc  $[0;10]$  và chính xác đến 0.5



# Lược đồ

## 1. Lược đồ quan hệ (relation schema)

- Tên của quan hệ
- Tên của tập thuộc tính

$$R(A_1, A_2, A_3, \dots, A_n)$$

Tên quan hệ

Tập thuộc tính

**NhanVien (MaNV, HoTen, NamSinh, Dchi, Phai, Luong, PhG)**

Ví dụ: lược đồ quan hệ NhanVien có 7 thuộc tính

<u>MaNV</u>	HoTen	NamSinh	Phai	Luong	PhG
NV01	Nguyễn Trung Tiến	1993	Nam	5 000 000	5
NV02	Trần Thị Yến	1987	Nữ	5 500 000	4

Mỗi nhân viên có một MaNV duy nhất dùng để nhận diện HoTen, NamSinh, Phai, Luong, PhG

# Lược đồ

**2. Lược đồ CSDL (database schema):** Gồm nhiều lược đồ quan hệ

**NhanVien** (MaNV, HoTen, NamSinh, Dchi, Phai, Luong, PhG)

**PhongBan** (MaPhG, TenPhG, TruongPhong, NgayNhanChuc)

**DiaDiem\_PhG** (MaPhG, DiaDiem)

**ThanNhan** (Ma\_Nvien, TenTN, Phai, NgSinh, QuanHe)

**DeAn** (TenDA, MaDA, Ddiem\_DA, Phong)

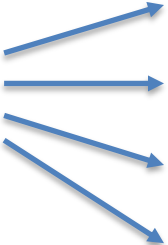
Lược  
đồ  
CSDL

# Thể hiện của quan hệ (relation instance)

- Là tập hợp các bộ giá trị của quan hệ tại một thời điểm nhất định
- Ký hiệu: thể hiện của quan hệ  $Q$  là  $T_Q$
- Ví dụ:  $T_{\text{MHoc}}$  là thể hiện của quan hệ Mhoc tại thời điểm hiện tại gồm có các bộ như sau:

**Mhoc**(MaMH: string, TenMH: string, TinChi: integer, Khoa: string)

Tuples  
(records, rows)



Mhoc	MaMH	TenMH	TinChi	Khoa
	COSC1310	Nhập môn tin học	4	CNTT
	COSC3320	Cấu trúc dữ liệu	4	CNTT
	MATH2410	Toán rời rạc	3	TOAN
	COSC3380	Cơ sở dữ liệu	3	CNTT

# Bậc của quan hệ

- **Bậc (Degree):** của lược đồ quan hệ là số lượng thuộc tính trong lược đồ

Số thuộc tính	Bậc của quan hệ
1	Quan hệ nhất phân (unary relation)
2	Quan hệ nhị phân (binary relation)
3	Quan hệ tam phân (ternary relation)
n	Quan hệ n phân (ternary relation)

- **Lượng số (cardinality):** số lượng các bộ của quan hệ. Lượng số sẽ thay đổi khi thêm hoặc xóa các tuple (hàng)

# Quy ước các ký tự

- **Lược đồ quan hệ R, bậc n**
  - **R** ( **$A_1, A_2, A_3, \dots, A_n$** ) hoặc **R** ( **$A_1: D_1, A_2: D_2, A_3: D_3, \dots, A_n: D_n$** )
  - R là tên lược đồ quan hệ
- **Quan hệ (thể hiện quan hệ)**
  - r, q, s, p
- **Bộ**
  - t, u, v
- **Miền giá trị của thuộc tính A**
  - Dom(A)
- **Giá trị của thuộc tính A trong bộ thứ i**
  - t.A hay t[A]

# Ví dụ về mô hình quan hệ

## SINHVIEN

<u>MASV</u>	TEN	MALOP
TCTH01	Sơn	TCTHA
TCTH02	Bảo	TCTHB
TCTH03	Trang	TCTHA

## MONHOC (COURSE)

<u>MAMH</u>	TENMH	TINCHI	KHOA
THVP	Nhập môn lập trình	4	CNTT
CSDL	Cơ sở dữ liệu	4	CNTT
CTDL	Toán rời rạc	3	TOAN

## LOPHOC

<u>MALOP</u>	TENLOP	SISO
TCTHA	TCTH32A	80
TCTHB	TCTH32B	65
TCTHC	TCTH32C	82

## KQUA (TRAINING)

<u>MASV</u>	<u>MAMH</u>	DIEM
TCTH01	THVP	8
TCTH01	CSDL	6
TCTH01	CTDL	7
TCTH02	THVP	9
TCTH02	CSDL	8
TCTH03	THVP	10

# NỘI DUNG

1. Giới thiệu
2. Các khái niệm của mô hình quan hệ
- 3. Ràng buộc toàn vẹn**
4. Các đặc trưng của quan hệ
5. Chuyển lược đồ ER sang thiết kế quan hệ

# Các ràng buộc toàn vẹn

- **RBTV (Integrity Constraint):** là những quy tắc, điều kiện, ràng buộc cần được thoả mãn mọi thể hiện của CSDL quan hệ.
- **RBTV** được mô tả khi định nghĩa được lược đồ quan hệ
- **RBTV** được kiểm tra khi các quan hệ có thay đổi
- Các kiểu ràng buộc chính:
  - ✓ Ràng buộc miền giá trị (Domain Constraints)
  - ✓ Ràng buộc khoá (Key Constraints)
  - ✓ Ràng buộc thực thể (Entity Integrity Constraints)
  - ✓ Ràng buộc toàn vẹn tham chiếu (Referential Integrity Constraints)
  - ✓ Ràng buộc do người dùng định nghĩa (User defined integrity)



# Bảo toàn thực thể (Entity Integrity)

- **Bảo toàn thực thể** dùng để bảo đảm **tính duy nhất của mỗi hàng trong bảng**.
  - Để bảo toàn thực thể thì mỗi quan hệ có 1 khoá chính và giá trị của khoá chính phải luôn hợp lệ
  - Trong một quan hệ cơ sở, mọi thuộc tính khoá không được có giá trị rỗng (NULL)
- Được thể hiện trong SQL server thông qua các ràng buộc sau:
  - Ràng buộc khoá chính (Primary Key)
  - Ràng buộc duy nhất (Unique)
  - Chỉ mục (Index)
  - Thuộc tính Identity

# Ràng buộc khoá (Key Constraints)

- **Ràng buộc khoá:** một quan hệ được định nghĩa như là một tập hợp các bộ (tuples). Tất cả các bộ trong một quan hệ phải là duy nhất
- **Khoá (keys):** một thuộc tính hoặc một tập các thuộc tính dùng để xác định một dòng trong một quan hệ. Khoá được chia làm 3 loại:
  - Siêu khoá (Super Key)
  - Khoá (Key)
  - Khoá chính (Primary Key)

# SIÊU KHOÁ (SUPER KEY)

- Các bộ trong quan hệ phải khác nhau từng đôi một
- Siêu khoá (SK)
  - Gọi SK là **tập con khác rỗng** các thuộc tính của R
  - SK là siêu khoá khi:

$$\forall r, \forall t_1, t_2 \in r, t_1 \neq t_2 \text{ thì } t_1[S] \neq t_2[S]$$

- SK là **tập các thuộc tính** dùng để xác định tính **duy nhất của mỗi bộ** trong quan hệ
- Mọi lược đồ quan hệ có **tối thiểu một siêu khoá**

# Ví dụ SIÊU KHOÁ (SUPER KEY)

HoTen	NamSinh	DiaChi	Phai	Luong	PhG
Phạm Thường	1994	Quận Bình Tân	Nữ	400 000	5
Nguyễn Hằng	1998	Quận Tân Phú	Nữ	250 000	4
Nguyễn Tâm	1994	Quận 8	Nam	1 000 000	4
Lê Thu	1999	Quận 8	Nữ	NULL	5



Chọn siêu khoá

HoTen	NamSinh	Luong	PhG
Phạm Thường	1994	400 000	5
Nguyễn Hằng	1998	250 000	4
Nguyễn Tâm	1994	1 000 000	4
Lê Thu	1999	NULL	5

(Thường, Phạm, 1994, 400 000, 5)



# Ví dụ SIÊU KHOÁ (SUPER KEY)

HoNV	NamSinh	DiaChi	Phai	Luong	PhG
Phạm Thương	1994	Quận Bình Tân	Nữ	400 000	5
Nguyễn Hằng	1998	Quận Tân Phú	Nữ	250 000	4
Lê Thu	1994	Quận 8	Nam	1 000 000	4
Lê Thu	1999	Quận 8	Nữ	NULL	4



Chọn siêu khoá

HoTen	DiaChi	Luong	PhG
Phạm Thương	Quận Bình Tân	400 000	5
Nguyễn Hằng	Quận Tân Phú	250 000	4
Lê Thu	Quận 8	1 000 000	4
Lê Thu	Quận 8	NULL	4

← (Thu, Quận 8, 1 000 000, 4)

← (Thu, Quận 8, NULL, 4)

Không  
biết Thu  
nào?



# Ví dụ SIÊU KHOÁ (SUPER KEY)

Quan hệ **LopHoc(MaLop, TenLop, SoSV, MaKhoa, MaGV<sub>VCN</sub>)**

- **Siêu khoá :**

$K1 = \{MaLop\}$

$K3 = \{MaGV_{VCN}\}$

$K2 = \{MaLop, TenLop\}$

$K4 = \{MaLop, SoSV\}$

$K5 = \{MaLop, MaGV_{VCN}\}$

$K6 = \{MaLop, TenLop, MaKhoa\}$

$K7 = \{MaLop, SoSV, MaKhoa\}$

$K8 = \{MaLop, TenLop, SoSV, MaKhoa\} \rightarrow$  Khó<sup>2</sup>a hiển nhiên

# KHOÁ DỰ TUYỂN (Candidate Key)

- **Khoá dự tuyển** là một siêu khoá  $K$  mà không có một tập con thực sự bất kỳ  $K' \subset K$  lại là một siêu khoá
  - **Tính duy nhất (uniqueness)**: các giá trị của  $K$  trong các bộ của  $r$  là duy nhất
  - **Tính tối giản (irreducibility hay minimality)** không có tập con thực sự  $K' \subset K$  lại có tính duy nhất.

Ví dụ:

Quan hệ **LopHoc (MaLop, TenLop, SoSV, MaKhoa, MaGVN)**

❖ Siêu khoá :

$K1 = \{\text{MaLop}\}$

$K3 = \{\text{MaGVN}\}$

$K2 = \{\text{MaLop}, \text{MaGVN}\}$

$K4 = \{\text{MaLop}, \text{TenLop}\}$

Xác định khoá dự tuyển?

# KHOÁ DỰ TUYỂN (Candidate Key)

- **Khoá dự tuyển** cần thoả mãn 2 tính chất sau:
  - Xác định duy nhất
  - Không dư thừa: khi xoá đi bất kỳ thuộc tính nào của khoá đều phá huỷ tính xác định duy nhất của khoá

Ví dụ trên:

- Xét  $K1 = \{Malop\}$  là siêu khóa vì chỉ có một thuộc tính đảm bảo tính duy nhất và tối giản nên  **$K1$  là khoá dự tuyển**
- Xét  $K2 = \{Malop, MaGVCN\}$  là siêu khoá và  **$K2$  không là khoá dự tuyển** vì tồn tại một  $K' = Malop \subset K2$  mà Malop lại là một siêu khóa.

Tương tự xét  $K3, K4???$



# KHOÁ (KEY)

- Gọi  $K$  là một tập con khác rỗng của các thuộc tính  $R$
- $K$  là khoá nếu thoả **2 điều kiện**
  1.  $K$  là một siêu khoá của  $R$
  2.  $\forall K' \subset K, K' \neq K, K'$  không phải là siêu khoá của  $R$

## Nhận xét:

- Giá trị của khoá dùng để nhận biết một bộ trong quan hệ
- Khoá là một đặc trưng của lược đồ quan hệ, không phụ thuộc vào thể hiện quan hệ
- Khoá được xây dựng dựa vào ý nghĩa của một số thuộc tính trong quan hệ
- Lược đồ quan hệ có thể có nhiều khoá

# KHOÁ (KEY) (tt)

Siêu khoá K = [TenNV, DiaChi, Luong, PhG]

TenNV	DiaChi	Luong	PhG
Thương	Quận Bình Tân	400 000	5
Hằng	Quận Tân Phú	250 000	4
Thu	Quận 8	1 000 000	4

## Khoá con K'

### TH1: K' là 1 thuộc tính

$K' = \text{TenNV}$

$K' = \text{DiaChi}$

$K' = \text{Luong}$

$K' = \text{PhG}$

### TH2: K' là 2 thuộc tính

$K' = \text{TenNV}, \text{DiaChi}$

$K' = \text{TenNV}, \text{Luong}$

$K' = \text{TenNV}, \text{PhG}$

$K' = \text{DiaChi}, \text{Luong}$

$K' = \text{DiaChi}, \text{PhG}$

$K' = \text{Luong}, \text{PhG}$

### TH3: K' là 3 thuộc tính

$K' = \text{TenNV}, \text{DiaChi}, \text{Luong}$

$K' = \text{TenNV}, \text{DiaChi}, \text{PhG}$

$K' = \text{DiaChi}, \text{Luong}, \text{PhG}$

# KHOÁ CHÍNH (Primary Key)

## Ví dụ:

NhanVien(MaNV, HoTen, NgSinh, DiaChi, Phai, Luong, PhG)

----

Có 2 cách chọn khoá chính:

- (1) MaNV
- (2) HoTen, NgSinh



NhanVien(MaNV, HoTen, NgSinh, DiaChi, Phai, Luong, PhG)

## Khi cài đặt quan hệ thành bảng

- ☐ Chọn 1 khoá làm cơ sở để nhận biết các bộ

Khoá có **ít thuộc tính hơn**

- ☐ Khoá được chọn gọi là khoá chính (PK)

**PK  $\neq$  NULL**

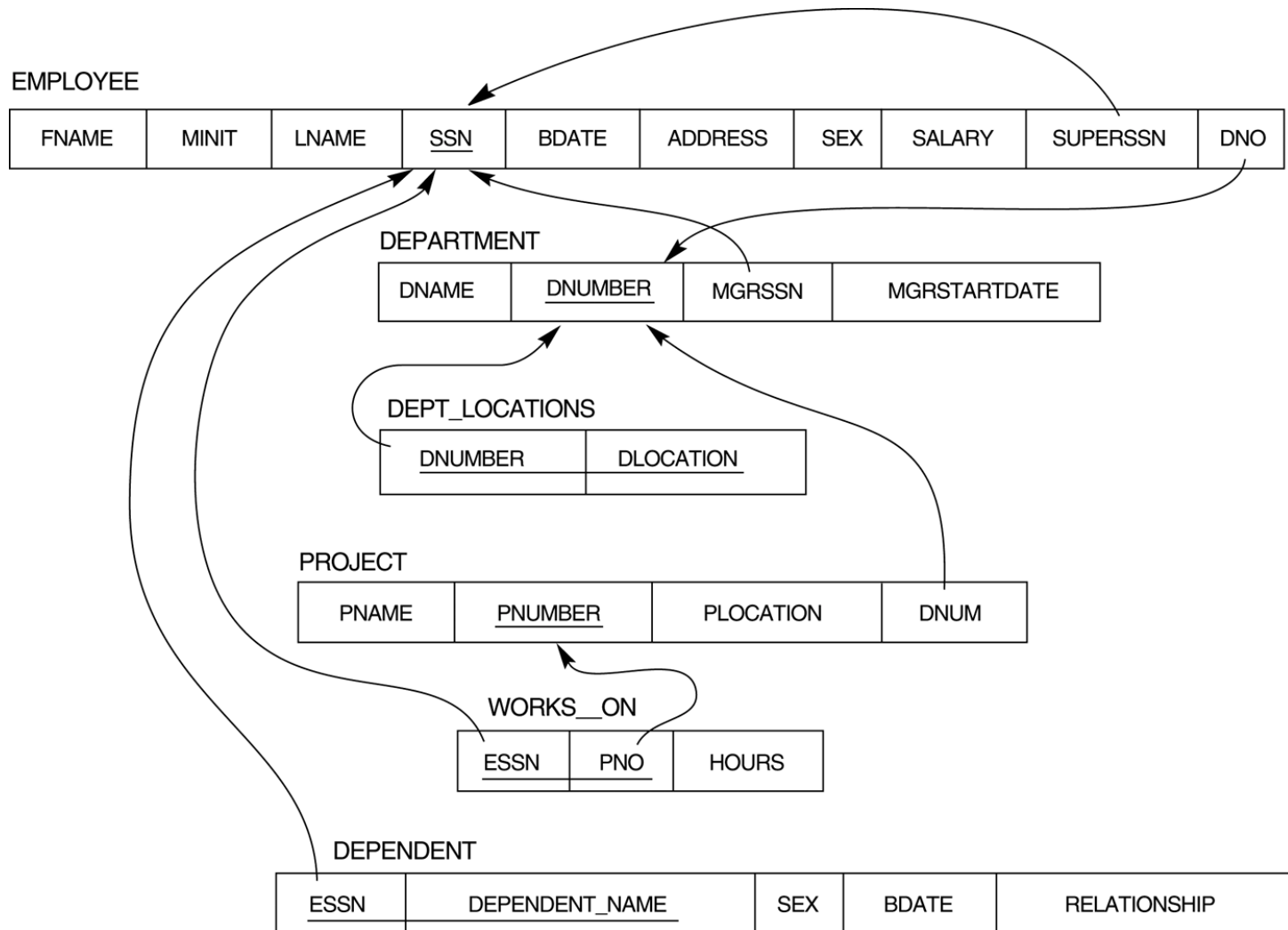
# KHOÁ NGOẠI (Foreign Key)

**Khoá ngoại (FK):** tập các trường hoặc thuộc tính trong một quan hệ được sử dụng để “tham chiếu” đến một bộ trong quan hệ khác

SINHVIEN					
HoTen	NamSinh	DiaChi	Phai	Luong	PhG
Phạm Thương	1994	Quận Bình Tân	Nữ	400 000	5
Nguyễn Hằng	1998	Quận Tân Phú	Nữ	250 000	4
Nguyễn Tâm	1994	Quận 8	Nam	1 000 000	4
Lê Thu	1999	Quận 8	Nữ	NULL	5

PHONGBAN	
MaPhG	TenPhG
4	CNTT
5	Nghiên cứu

# KHOÁ NGOẠI (Foreign Key)



# KHOÁ NGOẠI (Foreign Key) (tt)

## Nhận xét:

- Trong một lược đồ quan hệ, một thuộc tính vừa có thể tham gia vào khoá chính, vừa tham gia vào khoá ngoại
- Khoá ngoại có thể tham chiếu đến khoá chính trên cùng 1 lược đồ quan hệ
- Có thể có nhiều khoá ngoại tham chiếu đến cùng một khoá chính
- Ràng buộc tham chiếu = ràng buộc khoá ngoại

NHANVIEN(MaNV, TenNV, HoNV, NgSinh, DiaChi, Phai, Luong, PhG)

THANNHAN(MaNV, TenThanNhan, NgSinh, Phai)

# (1) BÀI TẬP XÁC ĐỊNH KHOÁ

Xét các lược đồ quan hệ sau:

- Employee(Emp\_ID, Name, Dept\_Name, Salary)
- Training(Emp\_ID, CourseID, Date\_Completed)
- Department(Dept\_Name, Location, Fax)
- Course(CourseID, CourseName, ...)

Xác định khóa chính, khóa ngoại ??

- Employee(Emp\_ID, Name, Dept\_Name, Salary)
- Training(Emp\_ID, CourseID, Date\_Completed)
- Department(Dept\_Name, Location, Fax)
- Course(CourseID, CourseName, ...)

## (2) BÀI TẬP XÁC ĐỊNH KHOÁ

Xác định các khóa của một lược đồ CSDL như sau:

- KHOA (MaKhoa, TenKhoa, NgayThanhLap)
- LOPHOC (MaLop, TenLop, NienKhoa, SoHocvien, MaKhoa)
- MONHOC (MaMon, TenMon, SoTC)
- HOCVIEN (MaHV, HoHV, TenHV, NgaySinh, QueQuan, MaLop)
- GIAOVIEN (MaGV, HoGV, TenGV, NgaySinh, HocVi, ChuyenNganh)
- KQUATHI (MaHV, MaMon, LanThi, NgayThi, DiemThi, GhiChu)
- DAY (MaGV, MaLop, MaMon)



## (2) BÀI TẬP XÁC ĐỊNH KHOÁ

- KHOA (MaKhoa, TenKhoa, NgayThanhLap)
- LOPHOC (MaLop, TenLop, NienKhoa, SoHocvien, MaKhoa)
- MONHOC (MaMon, TenMon, SoTC)
- HOCVIEN (MaHV, HoHV, TenHV, NgaySinh, QueQuan, MaLop)
- GIAOVIEN (MaGV, HoGV, TenGV, NgaySinh, HocVi, ChuyenNganh)
- KQUATHI (MaHV, MaMon, LanThi, NgayThi, DiemThi, GhiChu)
- DAY (MaGV, MaLop, MaMon)

# Toàn vẹn tham chiếu (referential integrity)

- Toàn vẹn tham chiếu dùng để xác định mối quan hệ giữa các bảng khi thêm, sửa hay xóa các hàng trong bảng.
- **Mục đích:** duy trì tính nhất quán (consistency) giữa các bộ của 2 quan hệ.
- Trong SQL server, thì toàn vẹn tham chiếu dựa vào mối quan hệ giữa khóa ngoại và khóa chính của các bảng.
  - **Bảng chứa khóa chính:** được gọi là bảng chính (primary table)
  - **Bảng chứa khóa ngoại:** được gọi là bảng quan hệ (related table)
- Nó bảo đảm giá trị khóa phải thống nhất (consistent) giữa các bảng. Không thể tham chiếu đến 1 giá trị không tồn tại và nếu giá trị khóa thay đổi thì tất cả tham chiếu đến giá trị khóa này cũng phải đồng loạt thay đổi theo

# Toàn vẹn tham chiếu (referential integrity)

- **Mục đích:** duy trì tính nhất quán (consistency) giữa các bộ của 2 quan hệ
- **Chú ý:** “ Nếu tồn tại 1 khóa ngoại trong 1 quan hệ, thì mỗi giá trị khóa ngoại phải tương ứng với 1 giá trị khóa chính trong một quan hệ khác (hoặc trong cùng quan hệ này) hoặc giá trị khóa ngoại phải là giá trị rỗng”

**MONHOC**

<u>MAMH</u>	TENMH	TINCHI	KHOA
THVP	Nhập môn TH	4	CNTT
CSDL	Cấu trúc dữ liệu	4	CNTT
CTDL	Toán rời rạc	3	TOAN

1

**KQUA**

MASV	MAMH	DIEM
TCTH01	THVP	8
TCTH01	CSDL	6
TCTH01	CTDL	7
TCTH02	THVP	9
TCTH02	CSDL	8
TCTH03	THVP	10

N

# Bảo toàn miền (Domain Integrity)

- Bảo toàn miền để bảo đảm tính hợp lệ của dữ liệu trong một thuộc tính.
- Việc xác định miền giá trị của các thuộc tính bao gồm một số các yêu cầu sau: *Tên thuộc tính, Kiểu dữ liệu, Độ dài dữ liệu, khuôn dạng của dữ liệu, các giá trị giới hạn cho phép, ý nghĩa, có duy nhất hay không, có cho phép giá trị rỗng hay không.*
- “Tất cả giá trị xuất hiện trong một thuộc tính của một quan hệ phải cùng một miền trị”
- Ví dụ:
  - Trong quan hệ KQUATHI, DiemThi là một số nguyên nằm trong khoảng từ 0 đến 10.
  - Trong quan hệ KQUATHI, LanThi là 1 hoặc 2.

# Bảo toàn miền (Domain Integrity)

- Được thể hiện trong SQL server thông qua các ràng buộc sau:
  - Kiểu dữ liệu (Data types)
  - Ràng buộc CHECK
  - Định nghĩa RULE, DEFAULT, NOT NULL.
- Giá trị NULL dùng để chỉ:
  - Giá trị chưa biết của thuộc tính
  - Giá trị đã biết nhưng đang bị thiếu
  - Giá trị null khác số 0 hay chuỗi rỗng “”

# Tính toàn vẹn do người dùng xác định (User-defined integrity)

- Tính toàn vẹn do người dùng xác định cho phép xác định các quy tắc nghiệp vụ của tổ chức mà không thuộc vào các loại toàn vẹn khác.
- Mức độ hỗ trợ cho các toàn vẹn do người dùng xác định phụ thuộc vào mỗi DBMS.
- Được thể hiện trong SQL server thông qua:
  - Các ràng buộc mức bảng
  - Stored procedure
  - Trigger
- Ví dụ: “ Mỗi văn phòng chi nhánh chỉ có tối đa 20 nhân viên”

# Quan hệ có cấu trúc tốt (well-structured relation)

- Là một quan hệ có dư thừa dữ liệu là tối thiểu và cho phép người sử dụng thêm, xóa hay sửa đổi các bộ của quan hệ mà không bị sai hay mâu thuẫn dữ liệu (data inconsistency)

# Bất thường dữ liệu (Data anomaly)

- Bất thường là 1 lỗi sai hay sự không nhất quán xảy ra khi dư thừa dữ liệu
- Ba loại bất thường:
  - Bất thường khi thêm vào
  - Bất thường khi xóa bỏ
  - Bất thường khi sửa đổi



# Ví dụ Bất thường dữ liệu (Data anomaly)

Xét quan hệ:

Employee(Emp\_ID, Name, Dept\_Name, Salary, Course, Date\_Completed)

- Khi thêm 1 nhân viên mới mà nhân viên này không tham gia bất kỳ khóa học nào
- Khi 1 nhân viên tham gia nhiều khóa học, xuất hiện nhiều lần trong quan hệ, nếu nhân viên này được tăng lương
- Nếu 1 nhân viên chỉ tham gia 1 khóa học và khóa học này chỉ có duy nhất 1 nhân viên, nếu nhân viên này bị xóa

→ Employee có phải là quan hệ cấu trúc tốt???

→ Sửa lại

Employee(Emp\_ID, Name, Dept\_Name, Salary, CourseID)

Course (CourseID, Date\_Completed)

---

# NỘI DUNG

1. Giới thiệu
2. Các khái niệm của mô hình quan hệ
3. Ràng buộc toàn vẹn
- 4. Các đặc trưng của quan hệ**
5. Chuyển lược đồ ER sang thiết kế quan hệ

# KHOÁ NGOẠI (FK)

- Thứ tự các bộ trong quan hệ là **không quan trọng**
- Thứ tự giữa các giá trị trong một bộ là **quan trọng**

**NHANVIEN**

TenNV	HoV	NamSinh	DiaChi	Phai	Luong	PhG
Thương	Phạm	1994	Quận Bình Tân	Nữ	400 000	5

Bộ <Thương, Phạm, 1994, Quận Bình Tân, Nữ, 400 000, 5>

≠

Bộ <Thương, Phạm, 1994, Quận Bình Tân, 400 000, Nữ, 5>

- Mỗi giá trị trong một bộ hoặc là một **giá trị nguyên tố** hoặc là **giá trị rỗng**
- Không có bộ nào trùng nhau


# NỘI DUNG

1. Giới thiệu
2. Các khái niệm của mô hình quan hệ
3. Ràng buộc toàn vẹn
4. Các đặc trưng của quan hệ
- 5. Chuyển lược đồ E/R sang thiết kế quan hệ**

**Các quy tắc chuyển đổi**

# Chuyển đổi từ ERD sang mô hình quan hệ

- Mục tiêu là tạo ra bản thiết kế **CSDL quan hệ** với đầu vào là mô hình ER
- Ánh xạ khái niệm của 2 mô hình

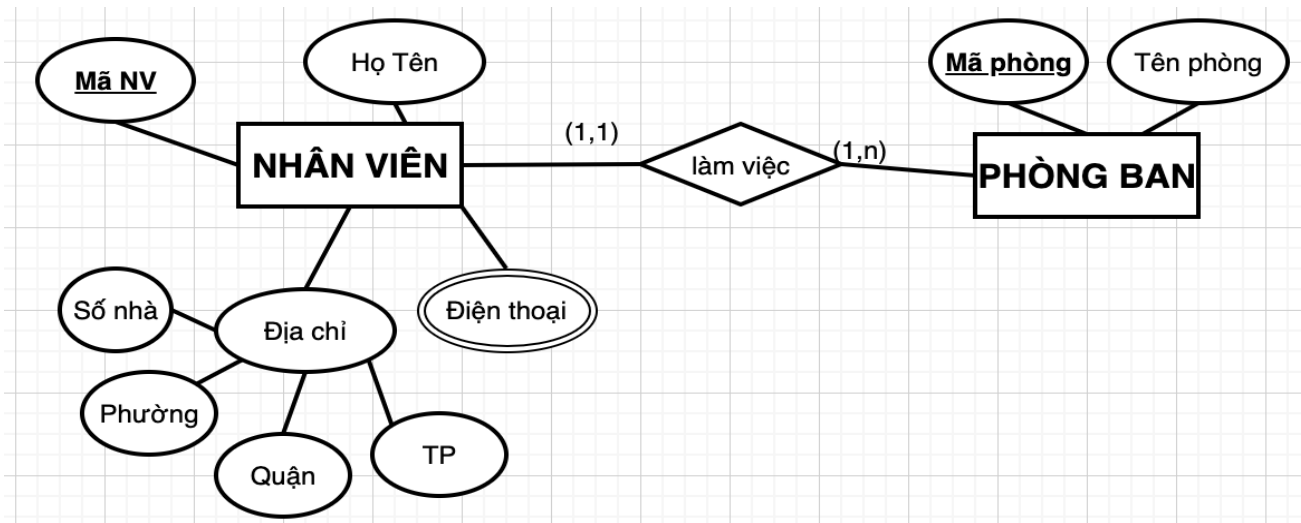
MÔ HÌNH ER		MÔ HÌNH QUAN HỆ
Tập thực thể		Quan hệ
Thực thể		Bộ
Thuộc tính của tập thực thể		Thuộc tính của quan hệ
Khoá chính của tập thực thể		Khoá chính của quan hệ
<b>Mối quan hệ giữa các tập thực thể</b>		<b>Khoá ngoại</b> của quan hệ

# Các bước chuyển từ ERD sang mô hình quan hệ

- (1) Chuyển đổi tập thực thể thông thường
- (2) Chuyển đổi tập thực thể yếu
- (3) Chuyển đổi mối liên kết 2 ngôi:
  - Mối liên kết n-n: sử dụng quan hệ kết hợp
  - Mối liên kết 1-n: sử dụng khoá ngoại hoặc quan hệ kết hợp
  - Mối liên kết 1-1: sử dụng khoá ngoại hoặc hợp nhất các thực thể hoặc dùng quan hệ kết hợp
- (4) Chuyển đổi tập thực thể kết hợp
- (5) Chuyển đổi mối liên kết 1 ngôi
- (6) Chuyển đổi mối liên kết 3 ngôi
- (7) Chuyển đổi kế thừa

# 1. Chuyển tập thực thể

- Với mỗi tập thực thể (**trừ tập thực thể yếu**) tạo một quan hệ tương ứng:
  - Tên quan hệ là tên của tập thực thể
  - (các) thuộc tính của quan hệ là (các) **thuộc tính đơn** của tập thực thể (**ngoại trừ thuộc tính kết hợp và đa trị**)



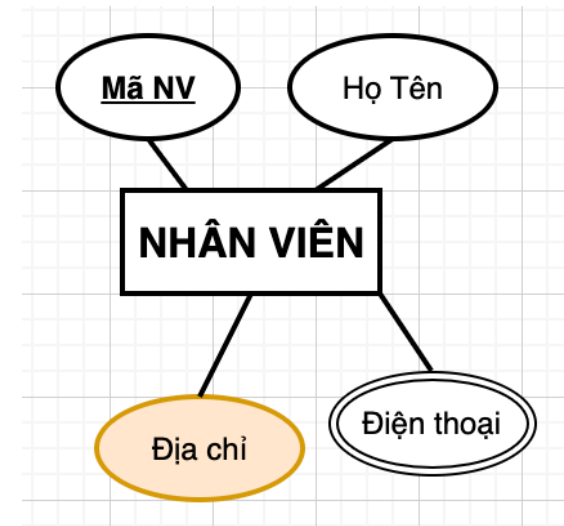
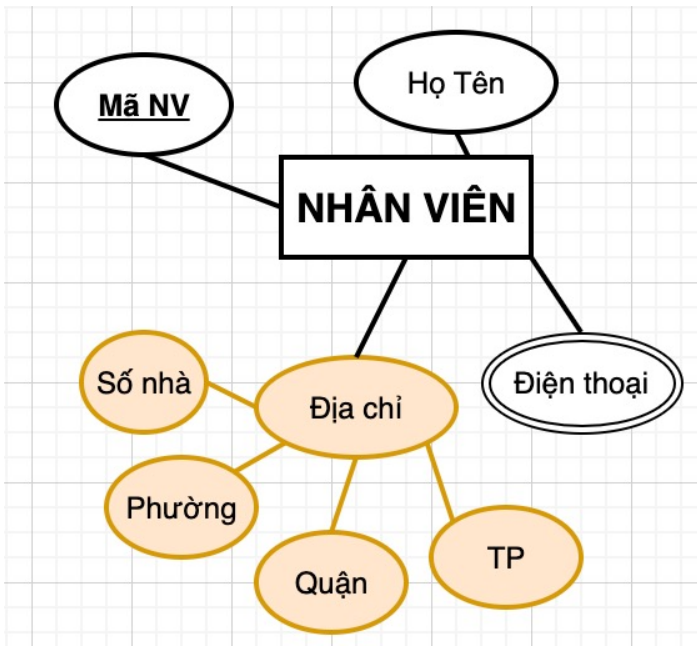
Chuyển tập thực thể thành quan hệ tương ứng:

- NHANVIEN(MaNV, HoTen)
- PHONGBAN(MaPhong, TenPhong)

# 1. Chuyển tập thực thể - Thuộc tính kết hợp

Tuỳ thực tế có thể chia làm 2 trường hợp

(1) – Một thuộc tính kết hợp được chuyển thành **một thuộc tính đơn bình thường**



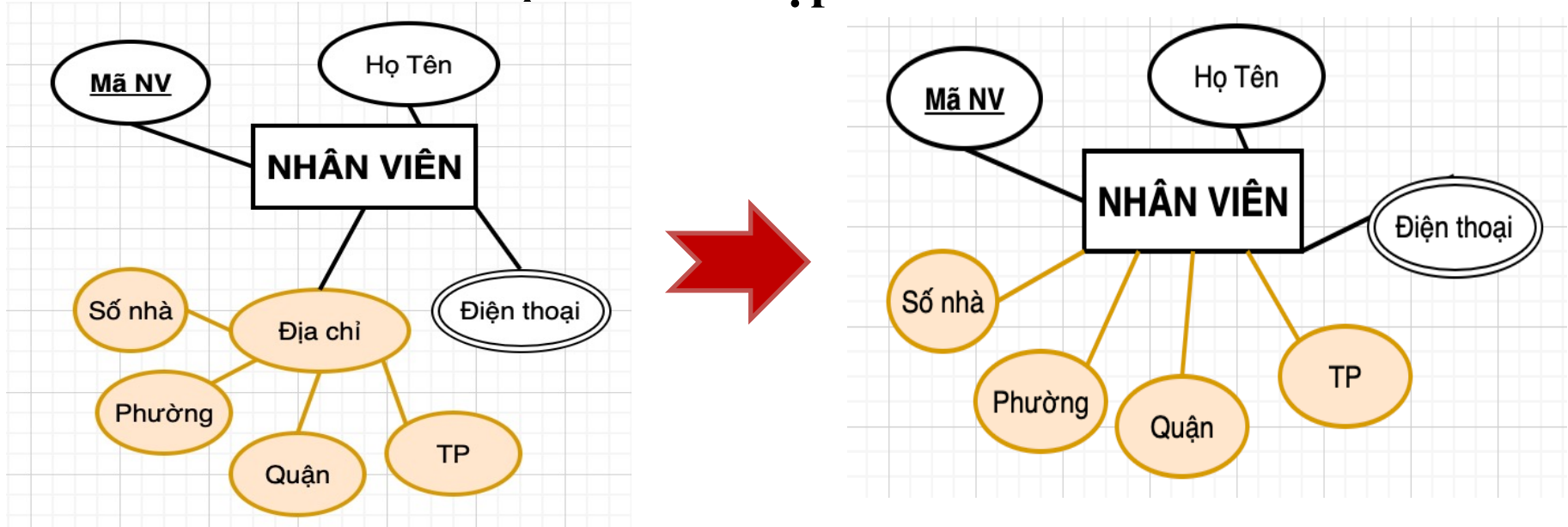
NHANVIEN(MaNV, HoTen , **DiaChi**)

MaNV	HoTen	DiaChi
NV01	Nguyễn Thành Chí Tâm	12 Nguyễn Văn Bảo, P.4, Q. Gò Vấp, TP. HCM



# 1. Chuyển tập thực thể - Thuộc tính kết hợp

(2) – Một thuộc tính kết hợp được chuyển thành một tập hợp các thuộc đơn cấu thành nên thuộc tính kết hợp đó

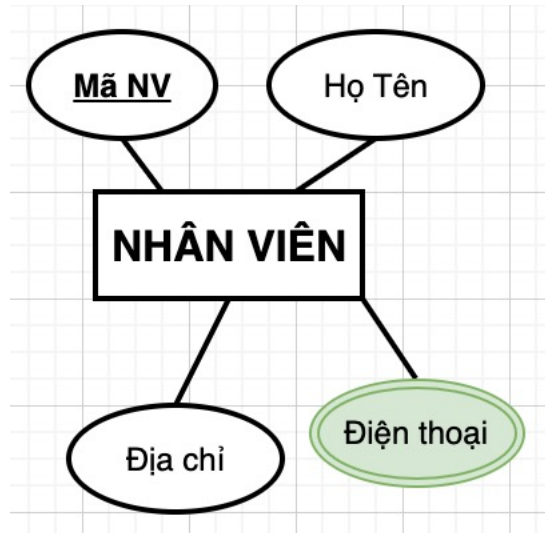


NHANVIEN(MaNV, HoTen, SoNha, Phuong, Quan, TP)

MaNV	HoTen	SoNha	Phuong	Quan	TP
NV01	Nguyễn Thành Chí Tâm	12 Nguyễn Văn Bảo	P.4	Q. Gò Vấp	TP. HCM

# 1. Chuyển tập thực thể - Thuộc tính đa trị

Với mỗi thuộc tính đa trị, tạo một quan hệ mới có các thuộc tính gồm **thuộc tính khoá chính của quan hệ và thuộc tính đa trị** đó



DIENTHOAI (MaNV, SoDT)

Hoặc

DIENTHOAI (MaNV, SoDT)

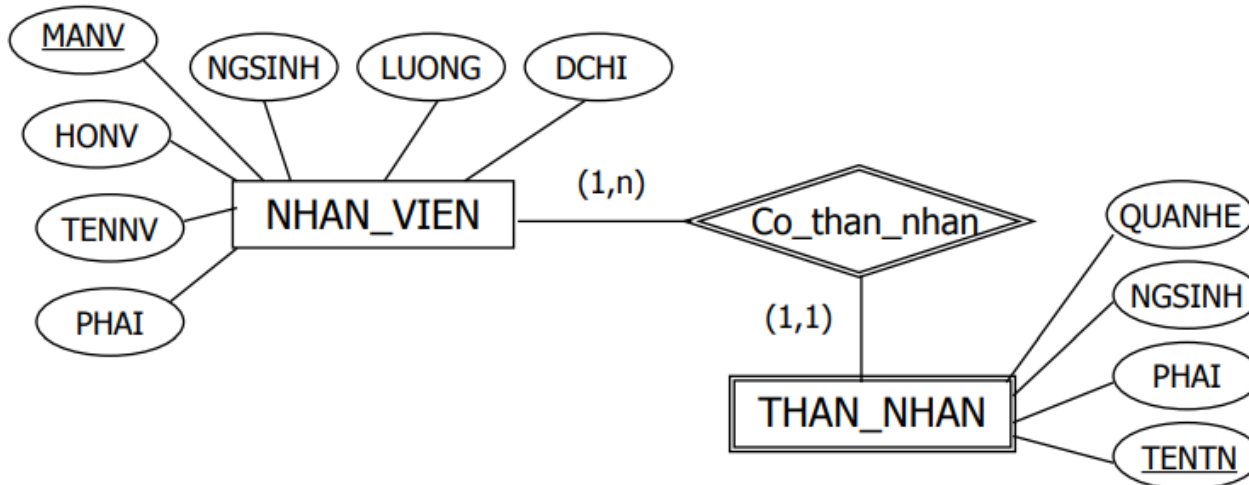
NHANVIEN(MaNV, HoTen, DiaChi)

## 2. Chuyển tập thực thể yếu

Với tập thực thể yếu, tạo một quan hệ tương ứng có:

- Tên trùng với tên tập thực thể yếu
- Thêm vào thuộc tính khoá của quan hệ liên quan

NhanVien (MaNV, HoNV, TenNV, Phai, NgSinh, Luong, Dchi)



THANNHAN (MaNV, TenTN, Phai, QuanHe, NgSinh)

### 3. Chuyển mối liên kết 2 ngôi (1-n)

Có hai cách tiếp cận: (1) sử dụng khoá ngoại, (2) tạo quan hệ kết hợp

#### (1) sử dụng khoá ngoại

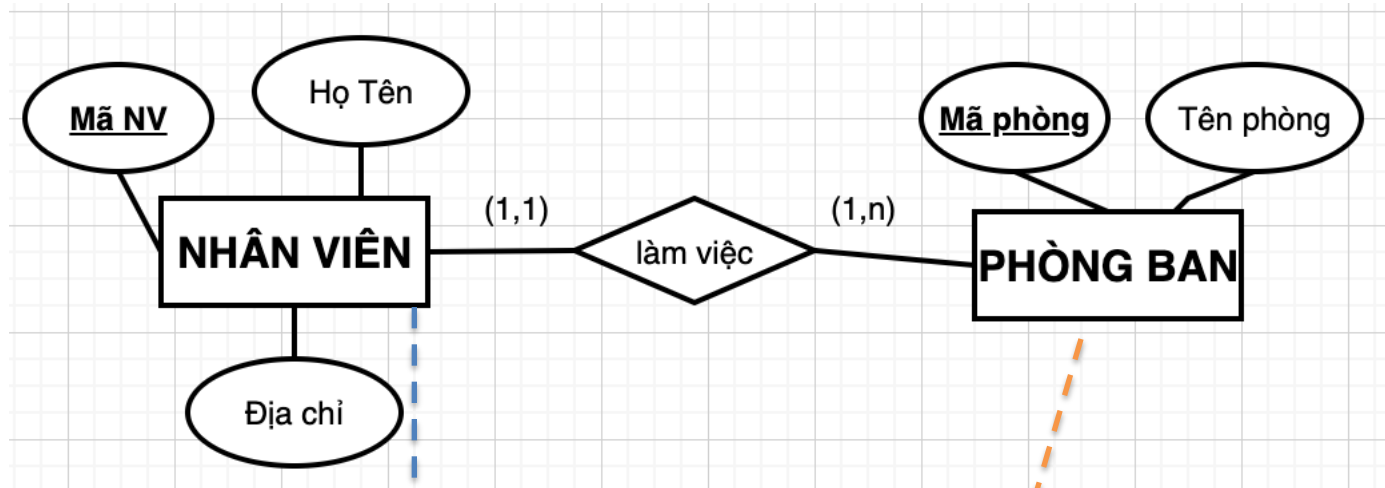
Thêm khoá chính của quan hệ phía 1 vào quan hệ phía n (trở thành khoá ngoại của quan hệ phía n)

#### (2) Tạo quan hệ kết hợp (ít dùng)

- Quan hệ kết hợp sẽ bao gồm thuộc tính khoá của 2 quan hệ
- Khoá chính của quan hệ kết hợp là **thuộc tính khoá của quan hệ phía n**

### 3. Ví dụ chuyển mỗi liên kết 2 ngôi (1-n)

Thiết kế CSDL dựa trên mô hình ER sau:

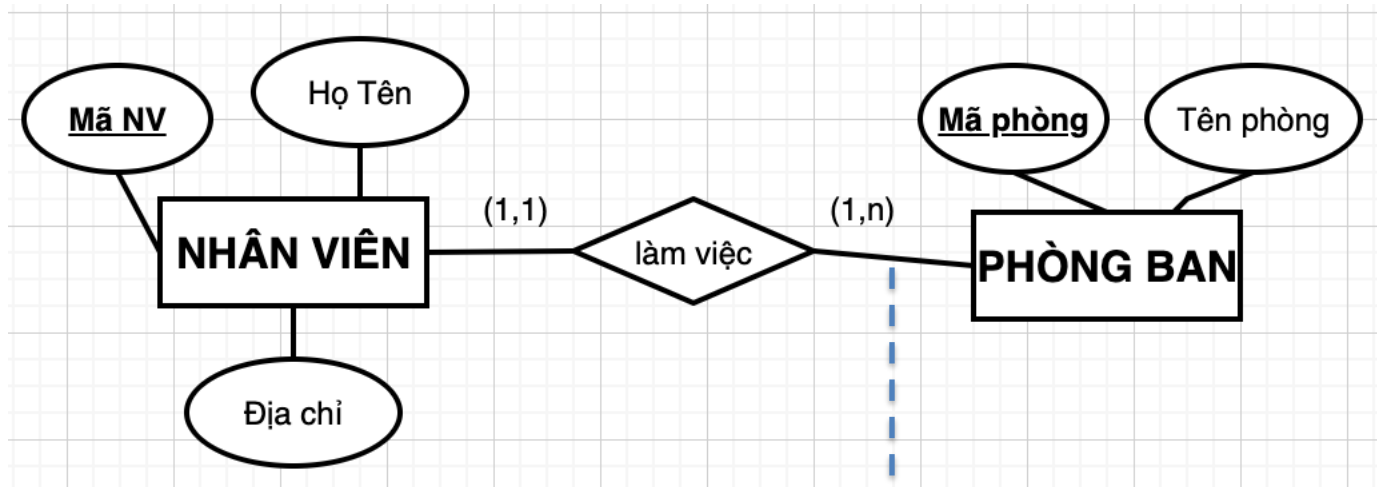


**Bước 1:** chuyển tập thực thể thành quan hệ tương ứng

PHONGBAN( MaPhong, TenPhong)

NHANVIEN(MaNV, HoTen, DiaChi)

### 3. Ví dụ chuyển mối liên kết 2 ngôi (1-n)



**Bước 2:** (cách 1) chuyển mối quan hệ bằng cách thêm khoá ngoại

PHONGBAN( MaPhong, TenPhong)

NHANVIEN(MaNV, HoTen, DiaChi, MaPhong)

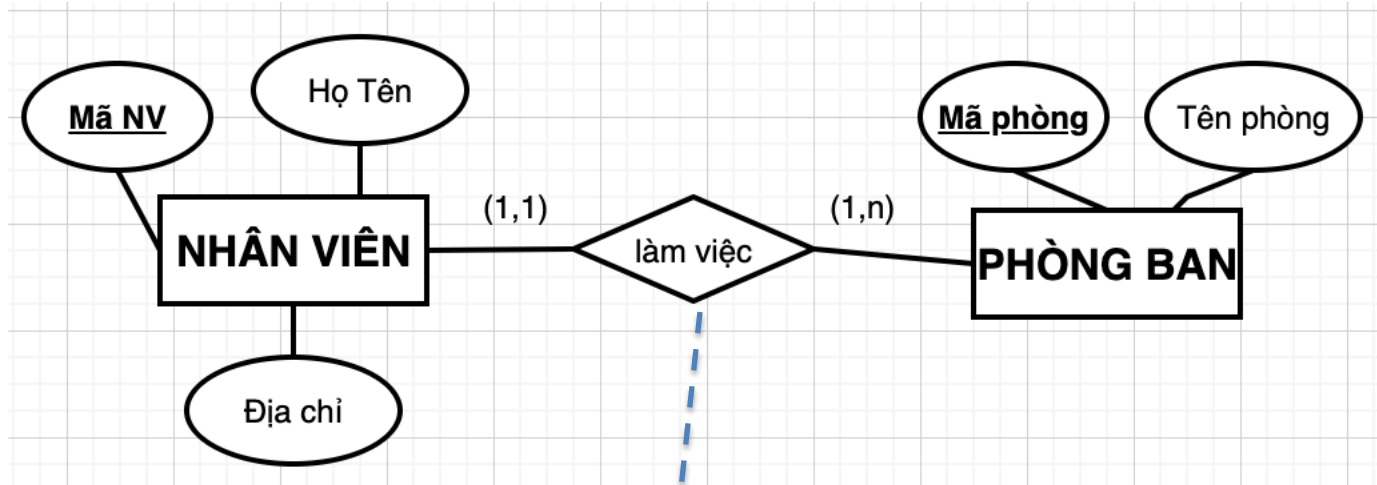
### 3. Ví dụ chuyển mỗi liên kết 2 ngôi (1-n)

Lược đồ CSDL quan hệ (cách 1)

PHONGBAN( MaPhong, TenPhong)

NHANVIEN(MaNV, HoTen, DiaChi, MaPhong)

### 3. Ví dụ Chuyển mỗi liên kết 2 ngôi (1-n)



**Bước 2: (cách 2 – ít dùng)** sử dụng quan hệ kết hợp

PHONGBAN( MaPhong, TenPhong)

NHANVIEN(MaNV, HoTen, DiaChi)

NHANVIEN\_PHONGBAN (MaNV, MaPhong)

FK1 = {MANV}

FK2 = {MaPhong}



### 3. Ví dụ chuyển mỗi liên kết 2 ngôi (1-n)

Lược đồ CSDL quan hệ (cách 2)

PHONGBAN( MaPhong, TenPhong)

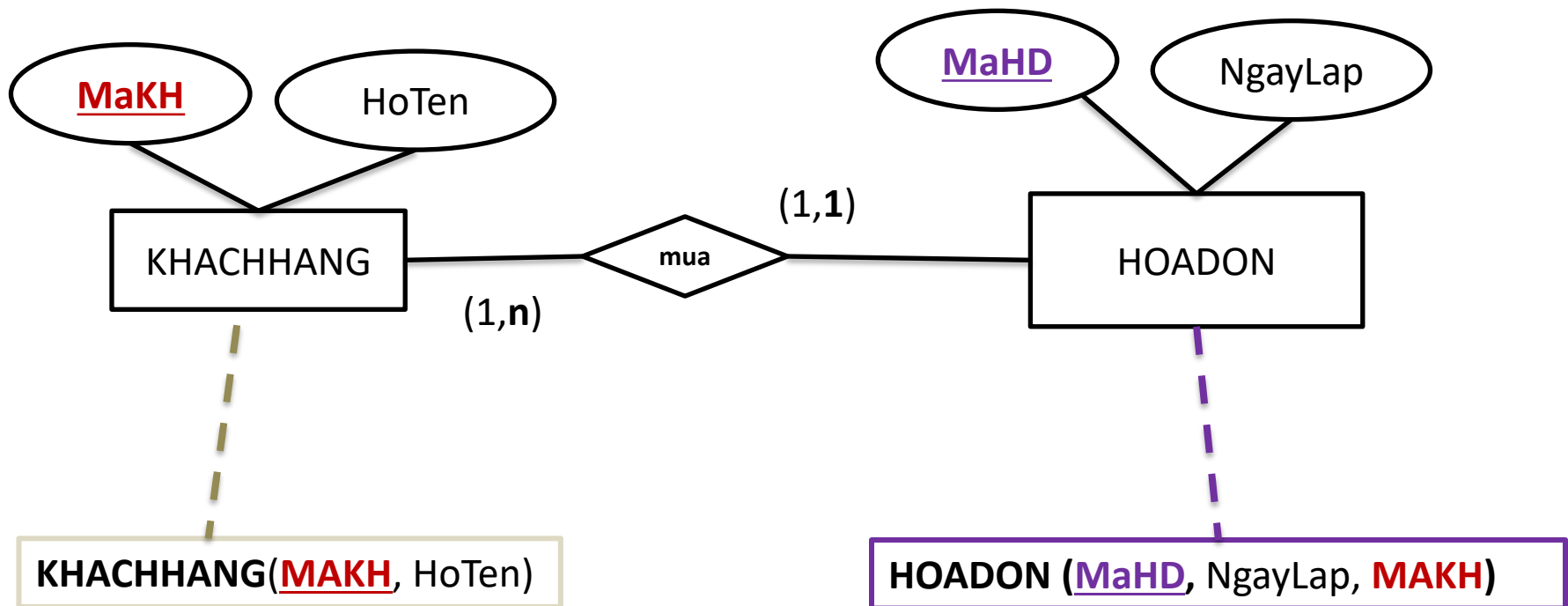
NHANVIEN(MaNV, HoTen, DiaChi)

NHANVIEN\_PHONGBAN (MaNV, MaPhong)

FK1 = {MANV}

FK2 = {MaPhong}

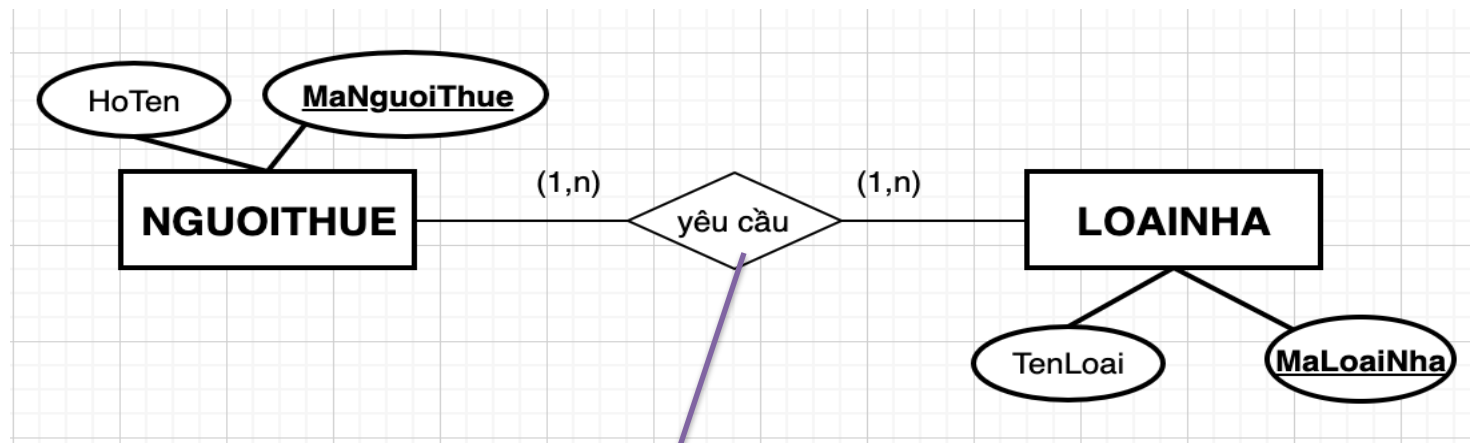
### 3. Ví dụ chuyển mối liên kết 2 ngôi (1-n)



### 3. Chuyển mối liên kết 2 ngôi (n-n)

#### Tạo một quan hệ mới có:

- Tên quan hệ là tên của tập mỗi kết hợp
- Thuộc tính gồm những **thuộc tính khoá** của các tập thực thể liên quan, và **thuộc tính riêng** của mỗi quan hệ (nếu có)

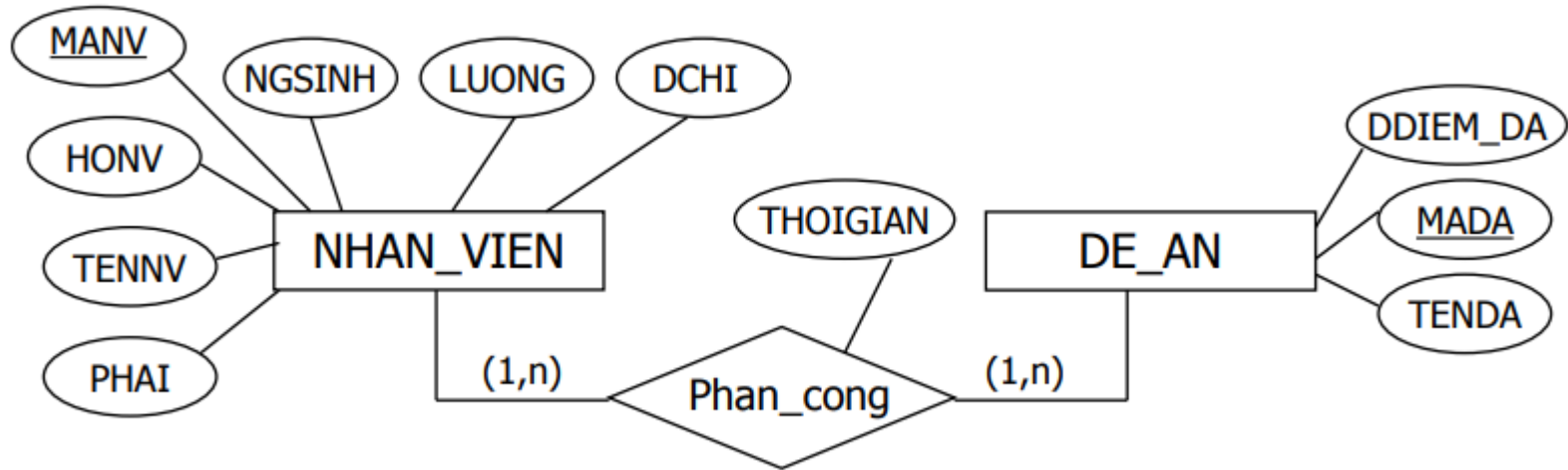


NGUOITHUE (MaNguoiThue, HoTen)

LOAINHA(MaLoaiNha, TenLoai)

NGUOITHUE\_ LOAINHA(MaLoaiNha, MaNguoiThue)

### 3. Chuyển mỗi liên kết 2 ngôi (n-n)



#### Lược đồ CSDL quan hệ

NhanVien (MaNV, HoNV, TenNV, Phai, NgSinh, Luong, Dchi)

DeAn (MaDA, TenDA, Ddiem\_DA)

PhanCong(MaNV, MaDA, ThoiGian)

### 3. Chuyển mỗi liên kết 2 ngôi (1-1)

Có 3 cách tiếp cận: (1) sử dụng khoá ngoại, (2) hợp nhất thành 1 quan hệ, (3) tạo quan hệ kết hợp

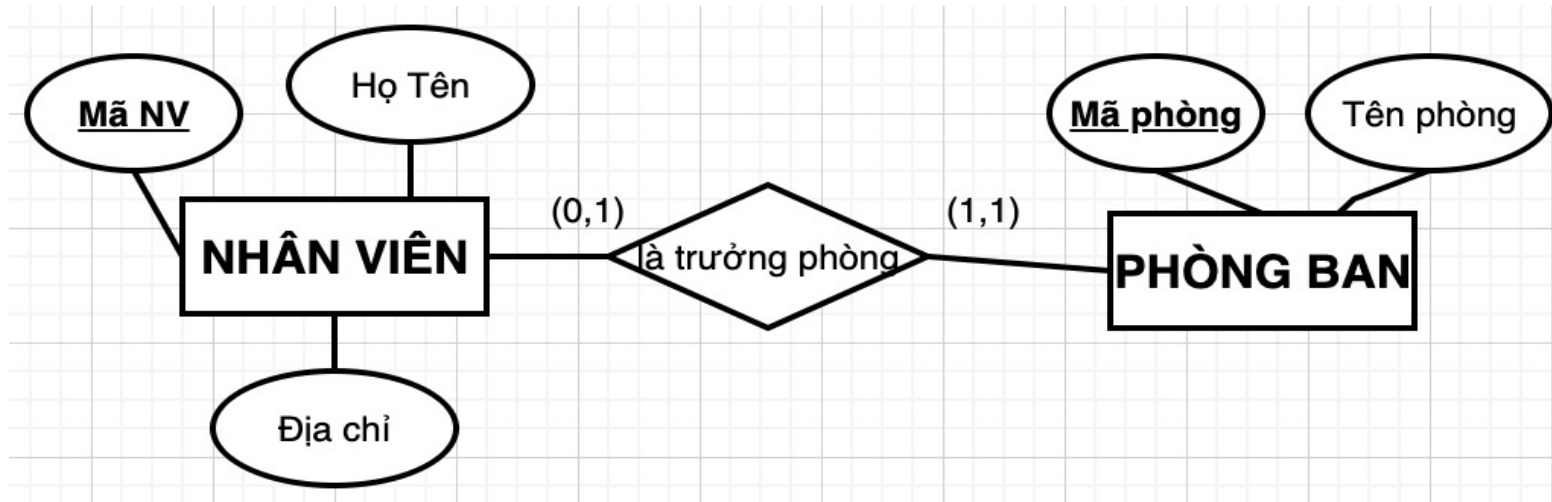
----

#### **(3) Tạo quan hệ kết hợp**

- Chuyển mỗi quan hệ 1:1 giữa R và T thành một quan hệ kết hợp S
- Quan hệ kết hợp S có
  - Các thuộc tính khoá của R và T (sẽ có 2 khoá ngoại của S)
  - Một khoá ngoại của S là Primary Key, khoá ngoại có ràng buộc unique (duy nhất)

### 3. Ví dụ chuyển mối liên kết 2 ngôi (1-1)

#### (1) Sử dụng khoá ngoại



Chọn cách nào???

Model 1

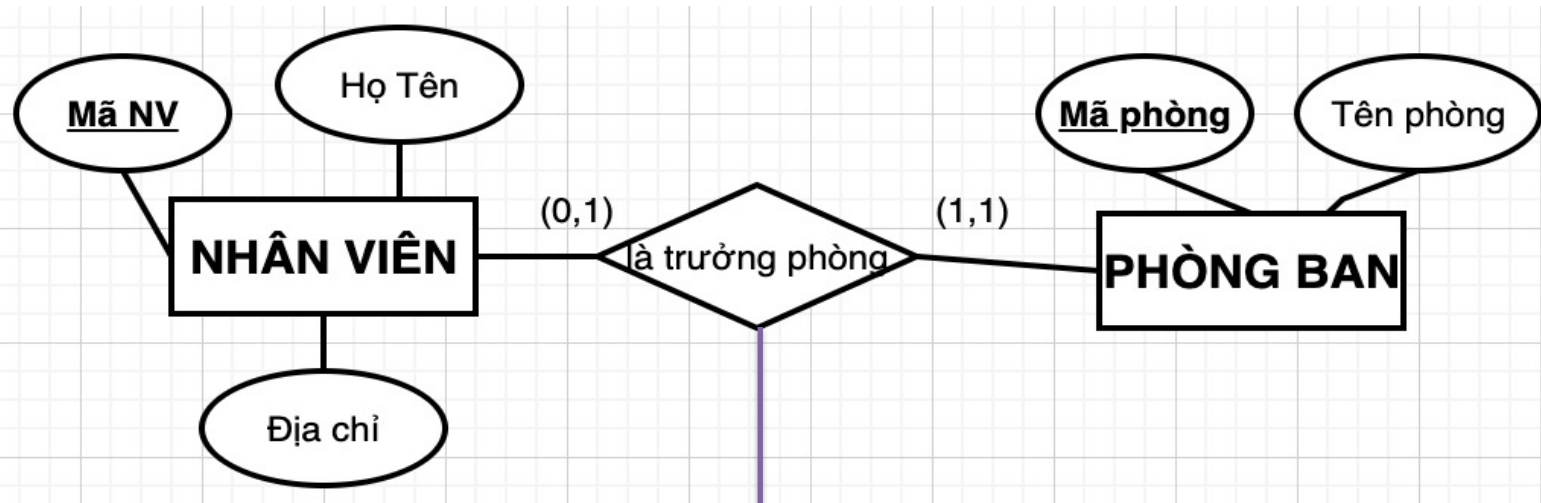
NHANVIEN (**MaNV**, HoTen)  
PHONGBAN (**MaPhong**, TenPhong, MaTruongPhong)

Model 2

NHANVIEN (**MaNV**, HoTen, MaTruongPhong)  
PHONGBAN (**MaPhong**, TenPhong)

### 3. Ví dụ chuyển mỗi liên kết 2 ngôi (1-1)

#### (2) Sử dụng quan hệ kết hợp (ít dùng)



NHANVIEN (MaNV, HoTen)

PHONGBAN (MaPhong, TenPhong)

TRUONGPHONG\_PHONGBAN (MaNV, MaPhong)

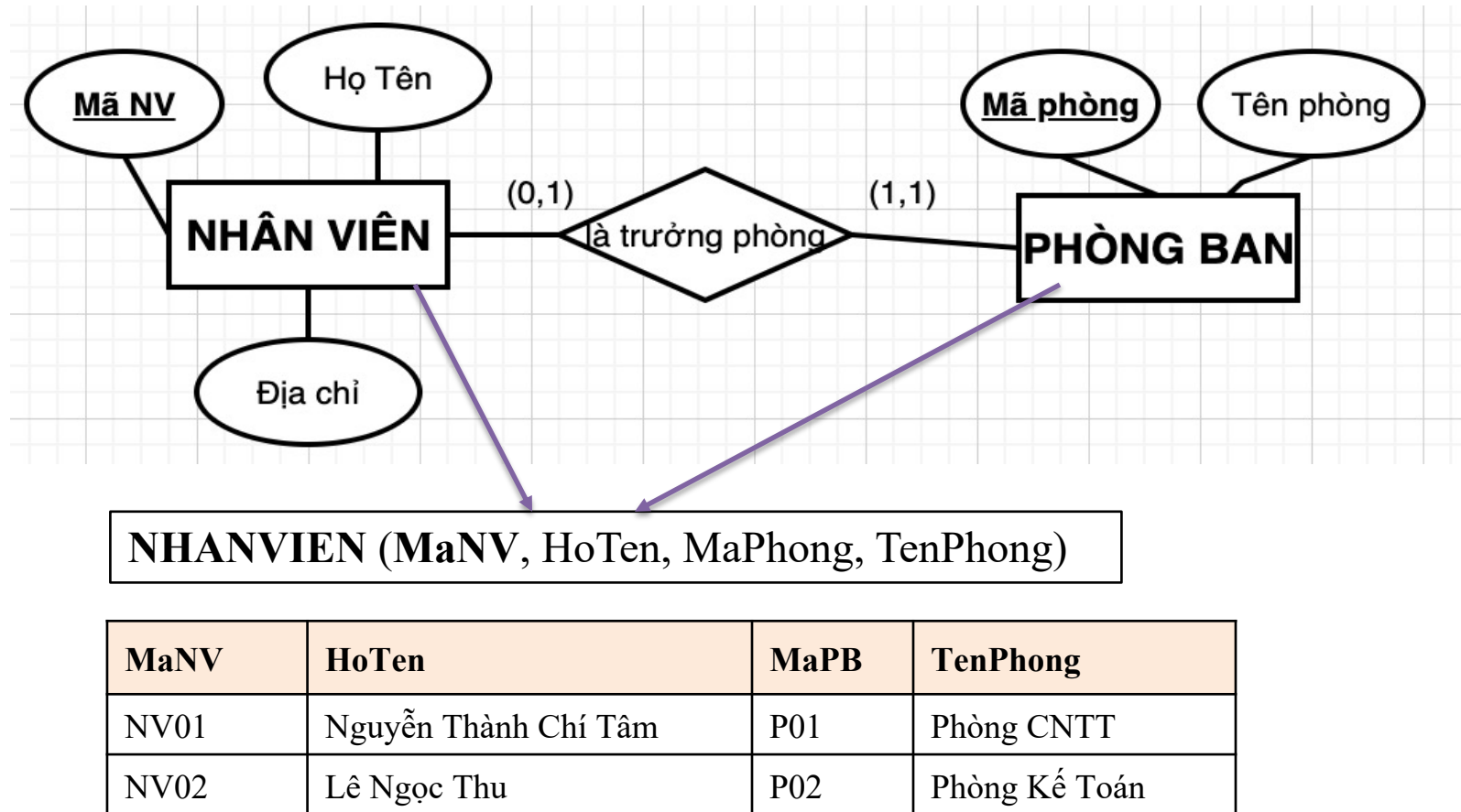
PK = {MaPhong}

Unique key = {MaNV}

MaPB	MaNV
P01	NV01
P02	NV02
P03	NV04

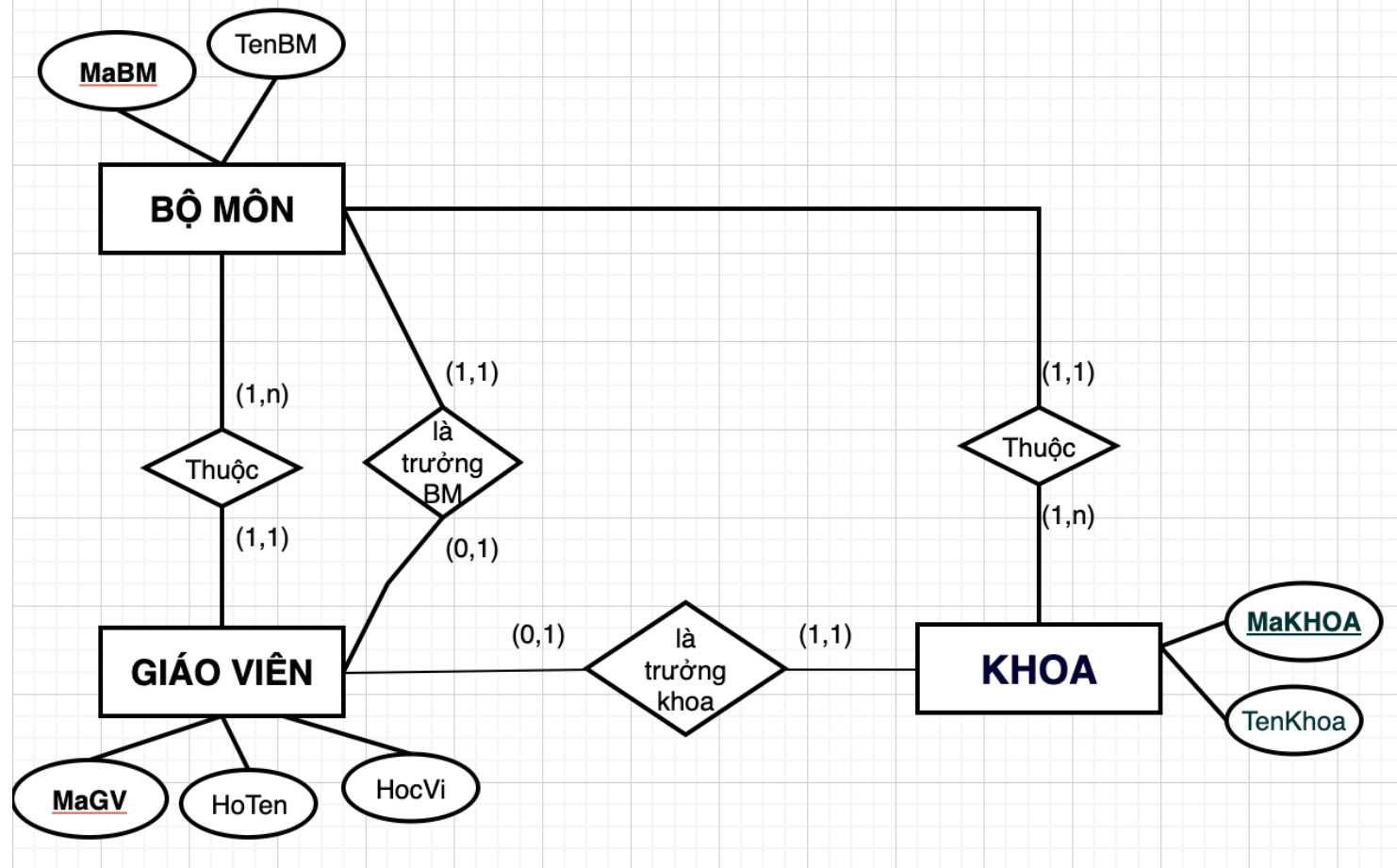
### 3. Ví dụ chuyển mỗi liên kết 2 ngôi (1-1)

#### (3) Hợp nhất 2 quan hệ thành 1 quan hệ





# Ví dụ



GIAOVIEN (MaGV, HoTen, HocVi, MaBM )

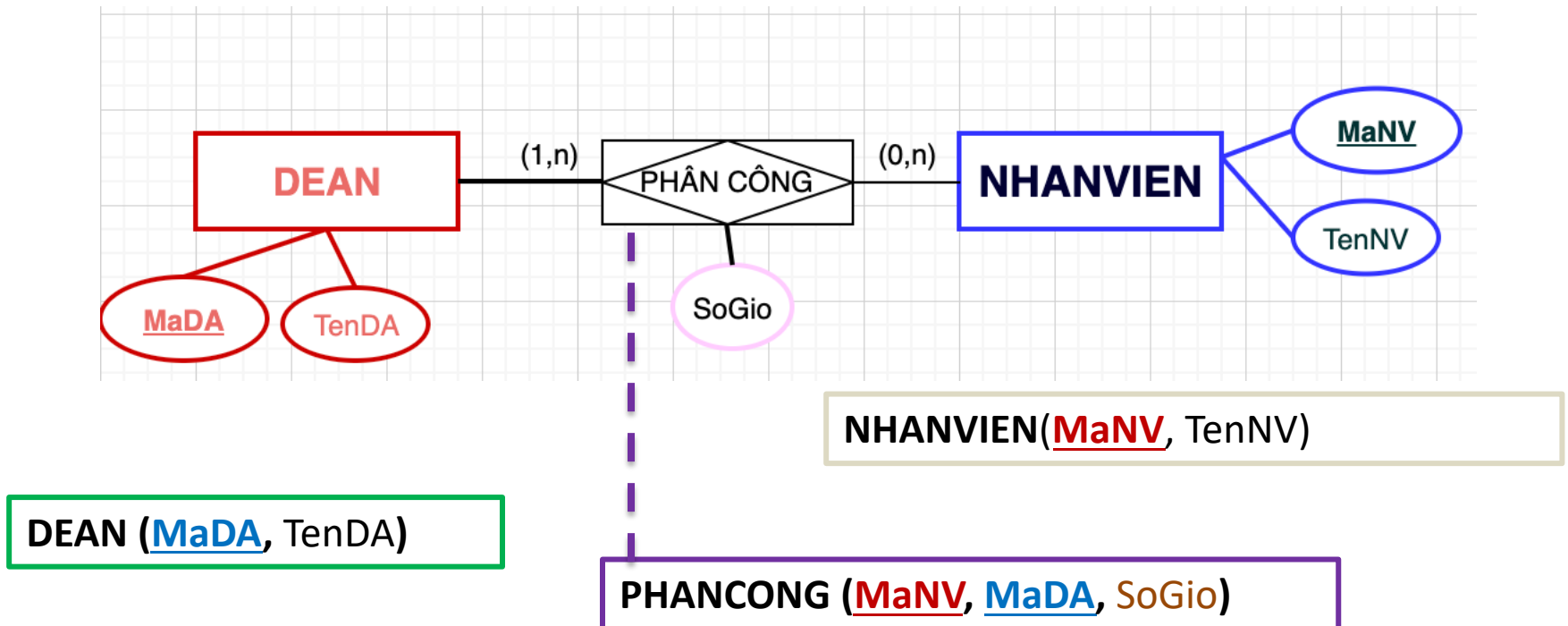
BOMON(MaBM, TenBM, MaTruongBM, MaKhoa)

KHOA (MaKhoa, TenKhoa, MaTruongKhoa)

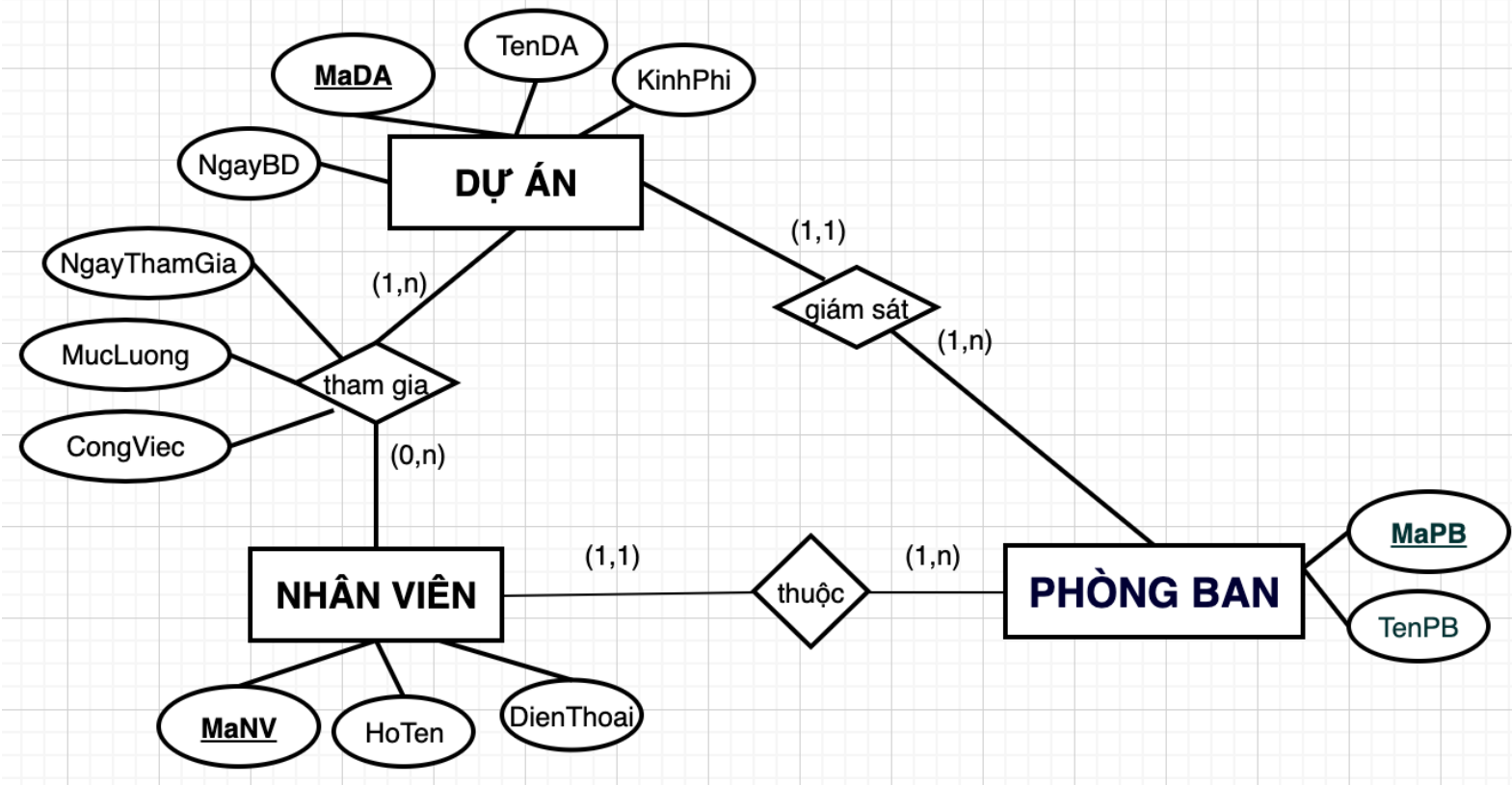
## 4. Chuyển tập thực thể kết hợp

- Một quan hệ mới được tạo ra để diễn tả cho kiểu thực thể kết hợp
  - Nếu kiểu **thực thể kết hợp không có thuộc tính** xác định riêng, khoá chính của quan hệ mới bao gồm khoá chính của 2 quan hệ tương ứng với 2 kiểu thực thể có liên quan
- Nếu kiểu **thực thể kết hợp có thuộc tính xác định riêng** thì thuộc tính này được dùng làm khoá chính cho quan hệ mới.

## 4. Ví dụ chuyển tập thực thể kết hợp



## 4. Ví dụ chuyển tập thực thể kết hợp



NHANVIEN (MaNV, HoTen, DienThoai, MaPB)

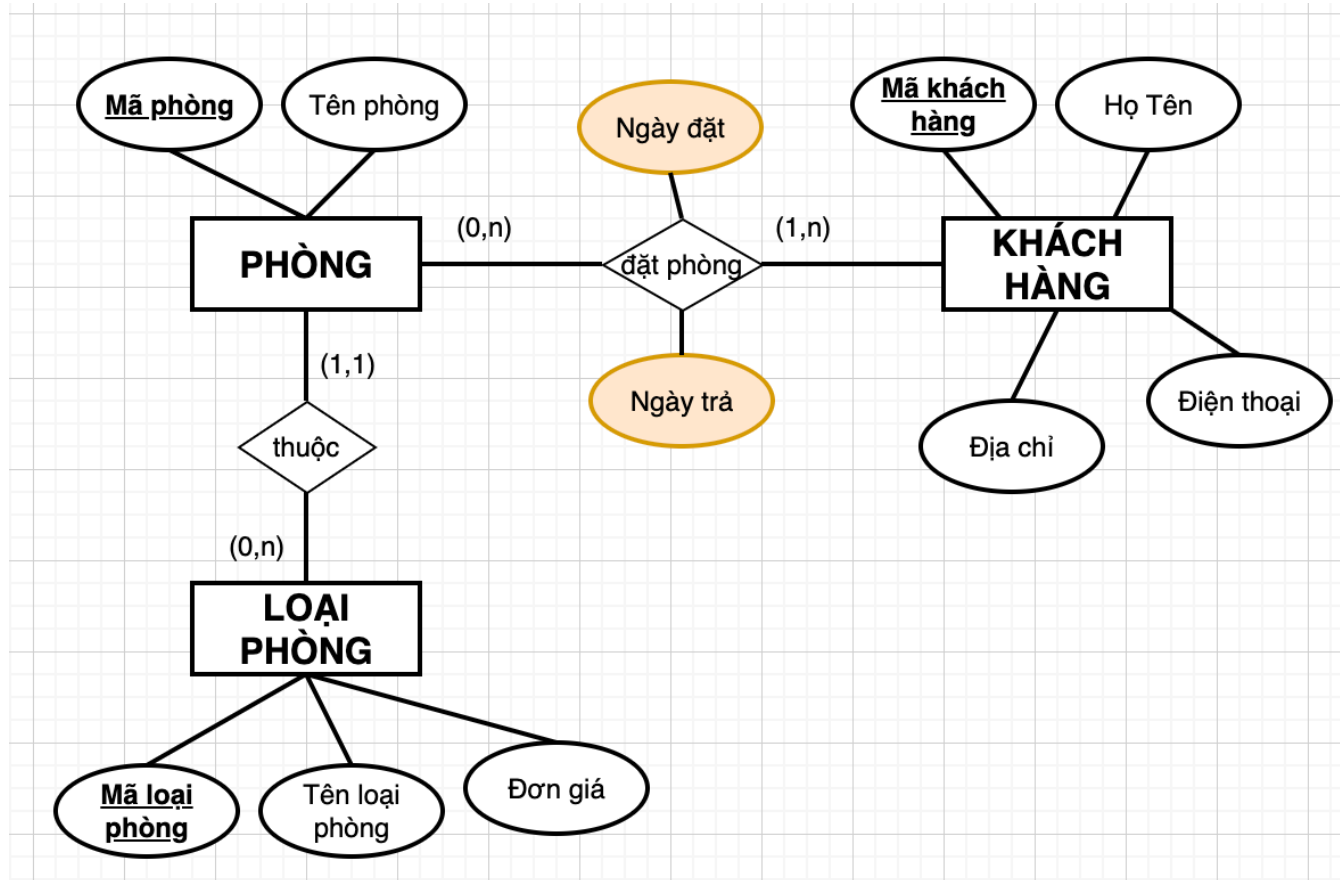
DUAN(MaDA, TenDA, KinhPhi, NgayBD, MaPB)

PHONGBAN(MaPB, TenPB)

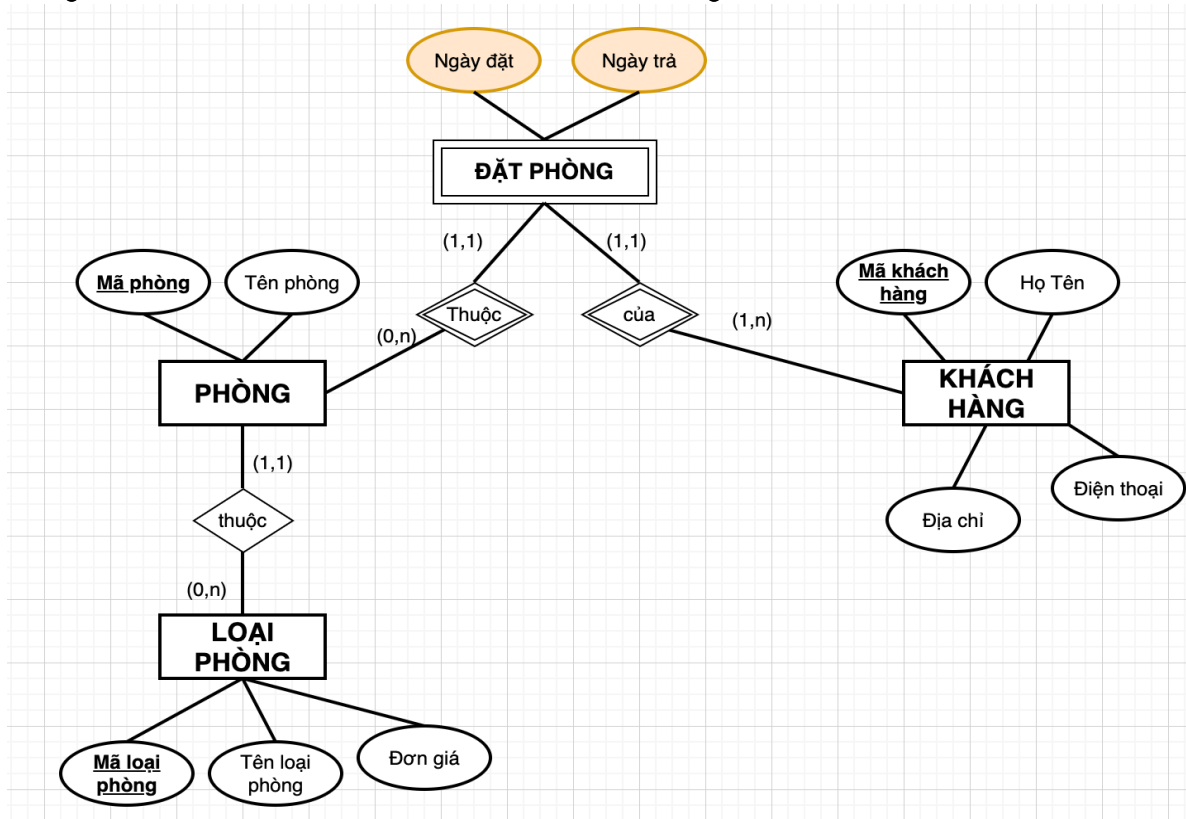
NHANVIEN\_DUAN(MaNV, MaDA, NgayThamGia, MucLuong, CongViec)

## 4. Ví dụ “QUẢN LÝ ĐẶT PHÒNG”

Mỗi liên kết “**đặt phòng**” chuyển thành tập thực thể kết hợp



## 4. Ví dụ “QUẢN LÝ ĐẶT PHÒNG”



KhachHang(**MaKH**, HoTen, DiaChi, DienThoai)

LoaiPhong(**MaLoaiPhong**, TenLoaiPhong, DonGia)

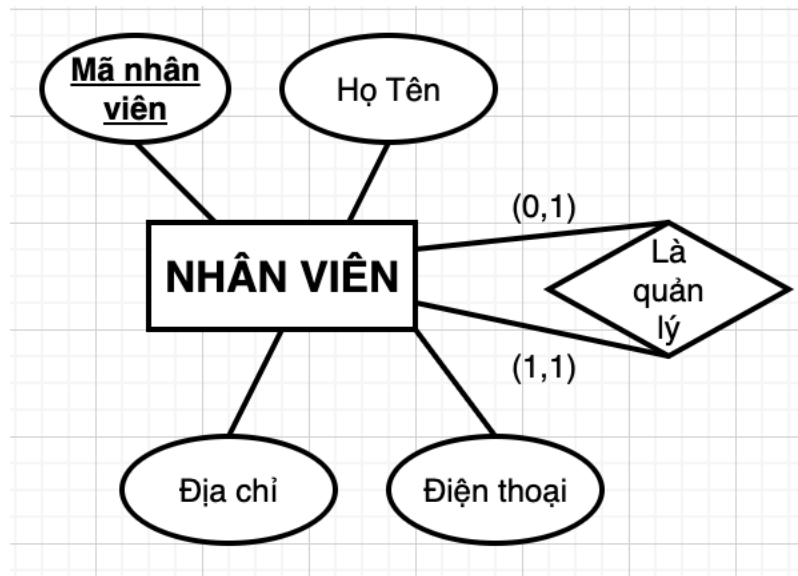
Phong(**MaPhong**, Tenphong, **MaLoaiPhong**)

DatPhong(**MaKH**, **MaPhong**, NgayDat, NgayTra)

## 5. Chuyển mỗi liên kết 1 ngôi

- Với liên kết 1 ngôi 1- n: thêm 1 khoá ngoại vào quan hệ của mỗi liên kết này để tham chiếu đến các giá trị của khoá chính của quan hệ (khoá ngoại và khoá chính phải cùng miền trị)

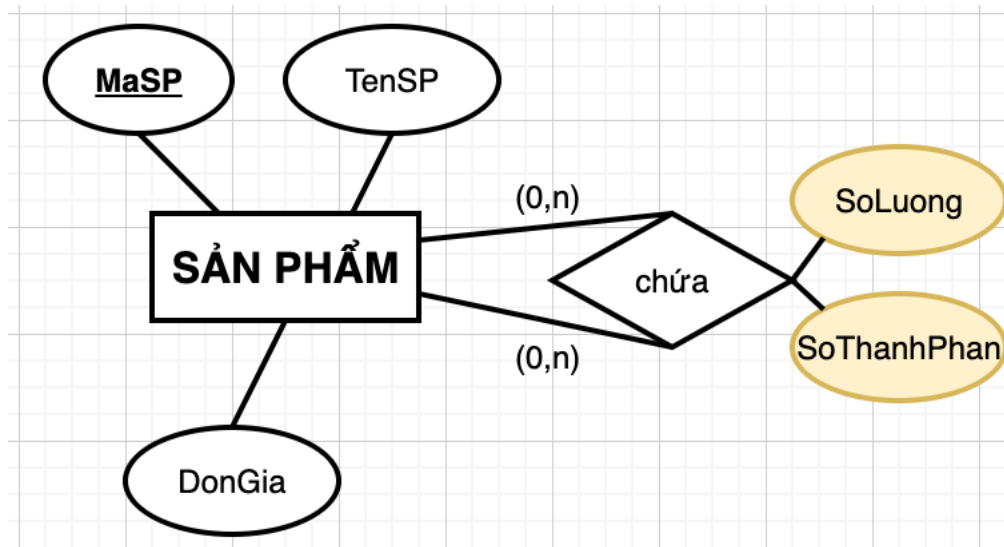
→ Được gọi là **khoá ngoại đệ quy**



NHANVIEN(MaNhanVien, HoTen, DiaChi, DienThoai, MaNQL)

## 5. Chuyển mỗi liên kết 1 ngôi

- Với liên kết 1 ngôi **n - n**: mỗi liên kết được chuyển thành 2 quan hệ
  - Một quan hệ diễn tả kiểu thực thể
  - Một quan hệ diễn tả chính mỗi liên kết với khoá chính bao gồm 2 thuộc tính lấy từ khoá chính của quan hệ tương ứng



SANPHAM(MaSP, TenSP, DonGia)

CHUA(MaSP, SoLuong, SoThanhPhan)



## 6. Chuyển mỗi liên kết 3 ngôi

- Mỗi liên kết 3 ngôi và các kiểu thực thể có liên quan được chuyển thành 4 quan hệ: **3 quan hệ cho 3 kiểu thực thể** liên quan và **1 quan hệ cho mỗi liên kết**
- Bốn trường hợp:
  - 1: 1: 1
  - 1: 1: n
  - 1: n: n
  - n: n: 1

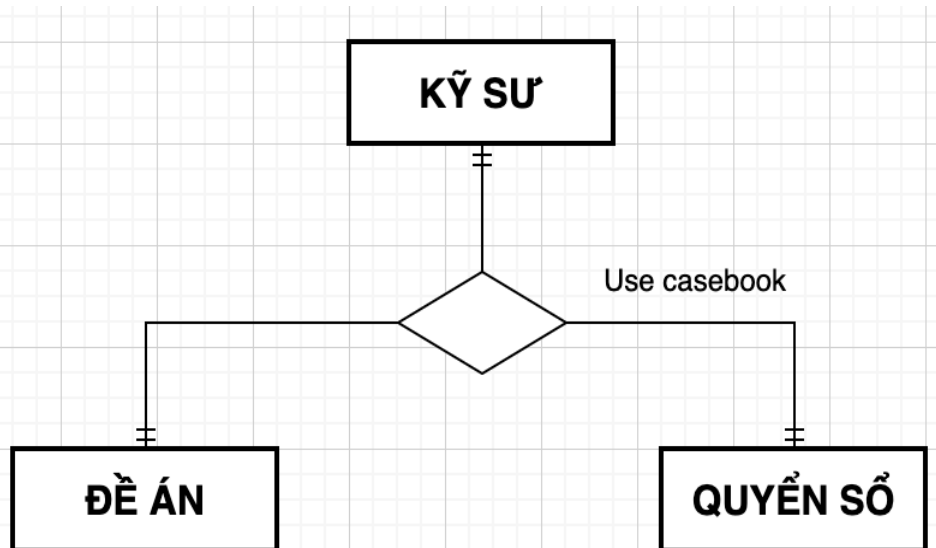
## 6. Chuyển mỗi liên kết 3 ngôi

### ▪ Trường hợp 1:1:1

“Mỗi kỹ sư dùng 1 casebook khác nhau cho mỗi project mà họ tham gia. Các kỹ sư không dùng chung casebook khi làm cùng một project”

---

3 tập thực thể: **KySu**, **Quyenso**, **DeAn**



KySu (**MaKS**, TenKS)

DeAn (**MaDA**, TenDA)

Quyenso (**MaQS**, TenQS)

SuDung\_Quyenso(**MaKS**, **MaDA**, MaQS)

## 6. Chuyển mối liên kết 3 ngôi

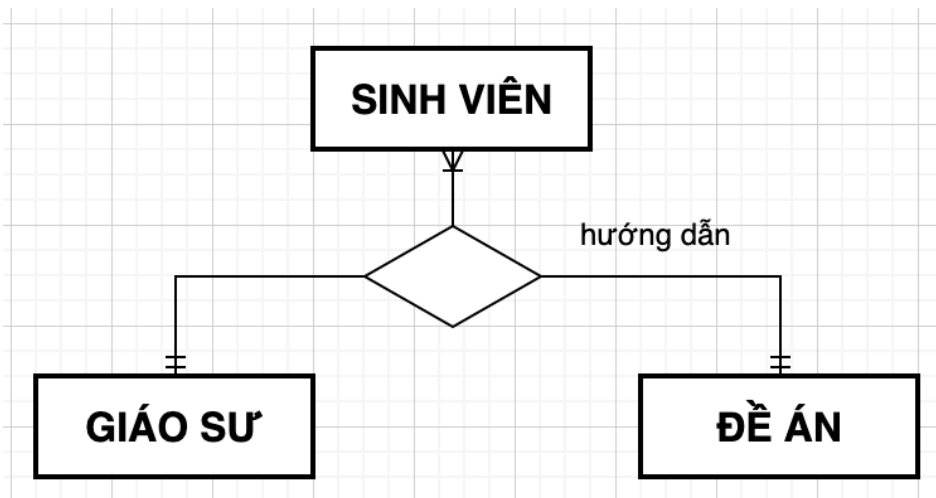
### ▪ Trường hợp 1:1:n

“Sinh viên làm đề tài theo sự hướng dẫn của giáo sư. Không giáo sư nào hướng dẫn cùng 1 sinh viên làm nhiều hơn 1 đề tài. Không sinh viên nào làm 1 đề tài với nhiều hơn 1 giáo sư”

---

3 tập thực thể: **GiaoSu, SinhVien, DeAn**

---



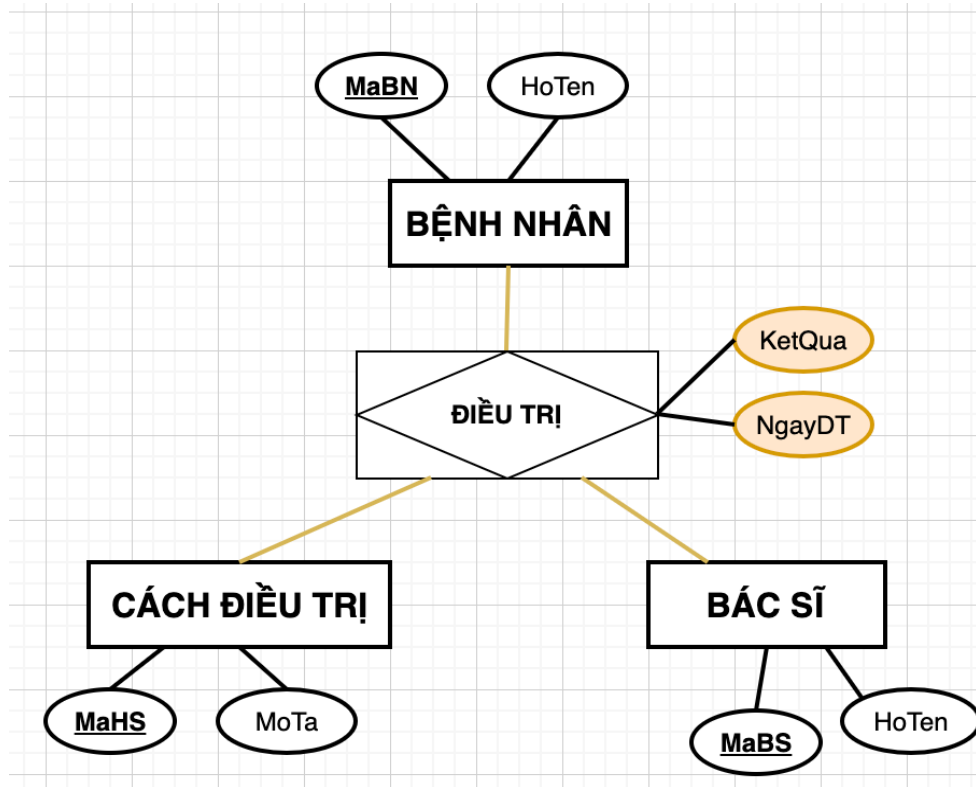
GiaoSu (**MaGS**, TenGS)

DeAn (**MaDA**, TenDA)

SinhVien(**MaSV**, TenSV)

HuongDan(**MaGS**, **MaSV**, MaDA)

## 6. Chuyển mỗi liên kết 3 ngôi



BenhNhan (MaBN, HoTen)  
BacSi (MaBS, HoTen)  
CachDieuTri (MaHS, MoTa)  
DieuTri (MaBN, MaHS, MaBS, KetQua, NgayDT)

# 7. Chuyển cấu trúc kế thừa

Có 3 cách:

- (1) Tạo một quan hệ cho mỗi tập thực thể: **gồm 1 superset và các subset**. Khoá của mỗi quan hệ là giống nhau, và là khoá K của superset
- (2) Mỗi subset tạo thành 1 quan hệ: **gồm các thuộc tính của subset và tất cả các thuộc tính của superset**. Khoá của mỗi quan hệ là khoá K của superset
- (3) Tạo 1 quan hệ đơn với 1 thuộc tính phân biệt và nhiều thuộc tính phân biệt

# 7. Ví dụ Chuyển cấu trúc kế thừa

## Cách 1

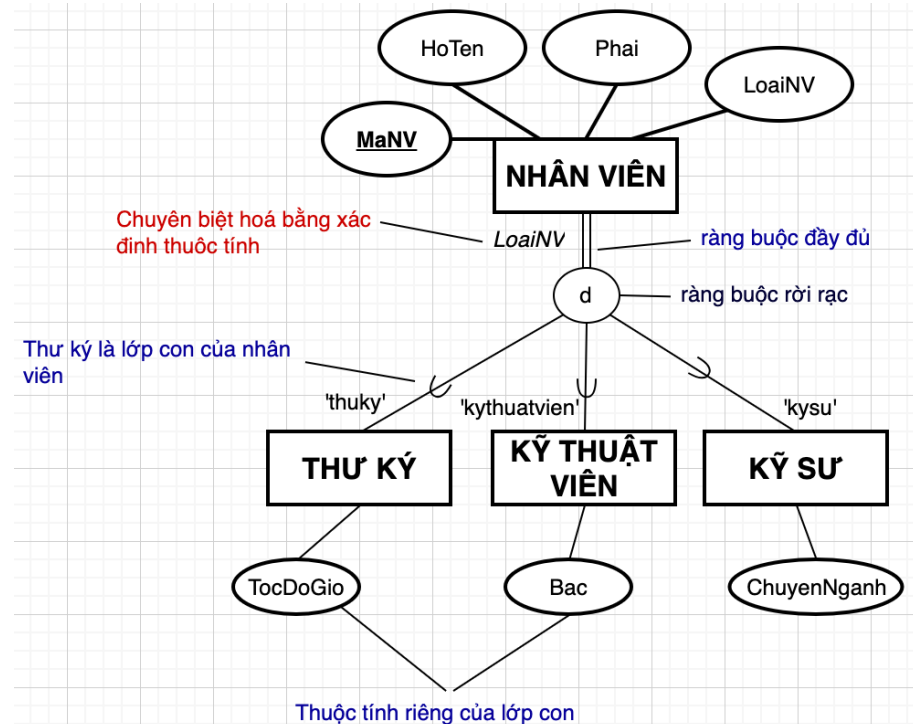
NhanVien (MaNV, HoTen, Phai, **LoaiNV**)  
ThuKy(MaNV, ToDoGio)  
KyThuatVien(MaNV, Bac)  
KySu(MaNV, ChuyenNganh)

## Cách 2

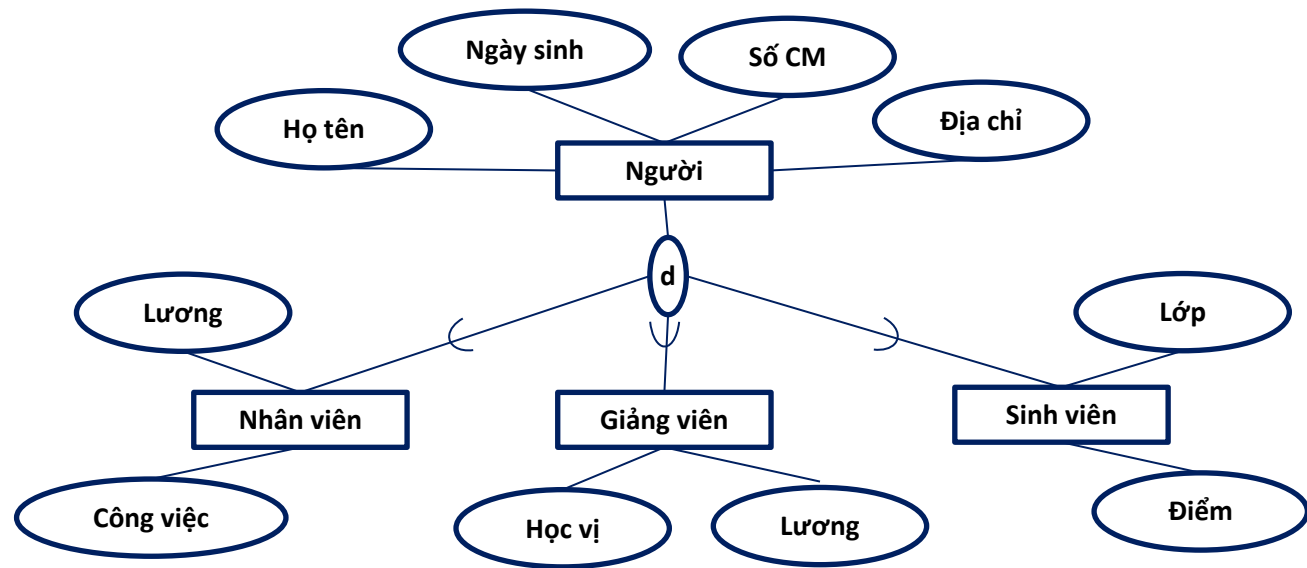
ThuKy(MaNV, HoTen, Phai, ToDoGio)  
KyThuatVien(MaNV, HoTen, Phai, Bac)  
KySu(MaNV, HoTen, Phai, ChuyenNganh)

## Cách 3

NhanVien(MaNV, HoTen, Phai, **LoaiNV**, ToDoGio, Bac, ChuyenNganh)



# 7. Ví dụ chuyển cấu trúc kế thừa



## Cách 1

Người(SoCM, HoTen, NgaySinh, DiaChi, *LoaiNV*)

NhanVien (SoCM, Luong, CongViec)

GiangVien (SoCM, HocVi, Luong)

SinhVien (SoCM, Lop, Diem)

## Cách 2

NhanVien (SoCM, HoTen, NgaySinh, DiaChi, Luong, CongViec)

GiangVien (SoCM, HoTen, NgaySinh, DiaChi, HocVi, Luong)

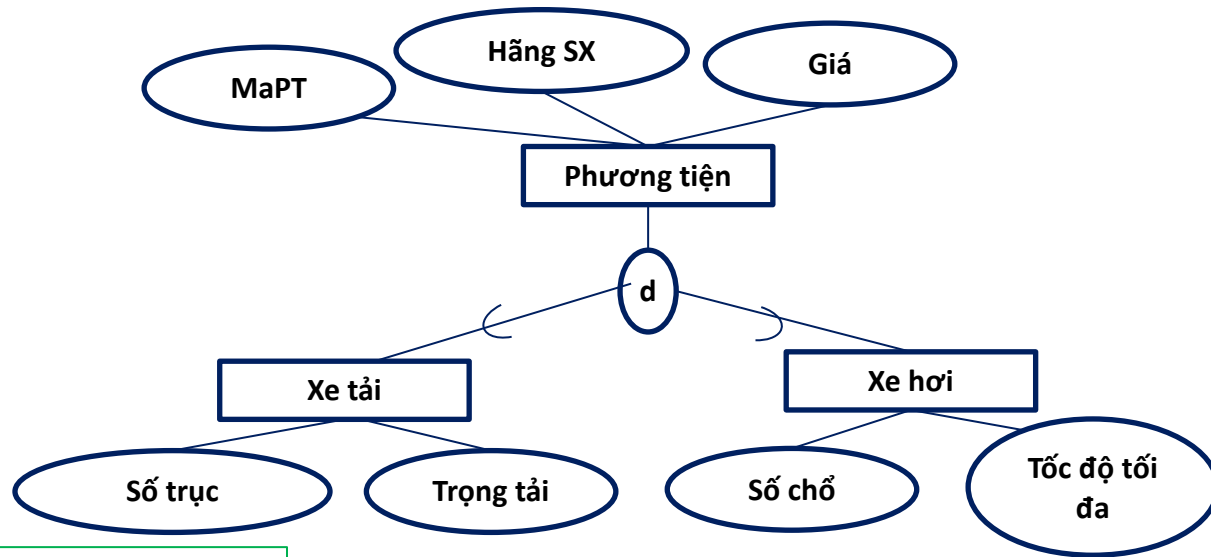
SinhVien (SoCM, HoTen, NgaySinh, DiaChi, Lop, Diem)

## Cách 3: chọn 1 thuộc tính phân biệt

NhanVien(SoCM, HoTen, NgaySinh, DiaChi, *LoaiNV*, Luong, CongViec , HocVi, Luong , Lop, Diem)

# 7. Ví dụ chuyển cấu trúc kế thừa

Ví dụ: Quan hệ của thực thể **Phương tiện**



## Cách 1

PhuongTien (MaPT, HangSX, Gia, **LoaiXe**)

XeTai(MaPT, SoTruc, TrongTai)

XeHoi(MaPT, SoCho, TocDoToiDa)

## Cách 2

XeTai(MaPT, HangSX, Gia, SoTruc, TrongTai)

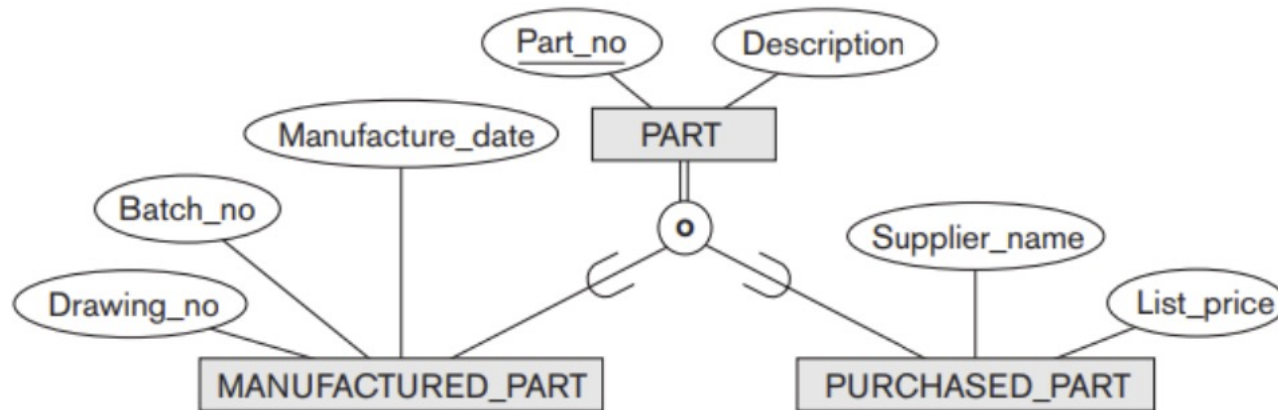
XeHoi(MaPT, HangSX, Gia, SoCho, TocDoToiDa)

## Cách 3

PhuongTien (MaPT, HangSX, Gia, **LoaiXe**, SoTruc, TrongTai, SoCho, TocDoToiDa)



# 7. Ví dụ chuyển cấu trúc kế thừa



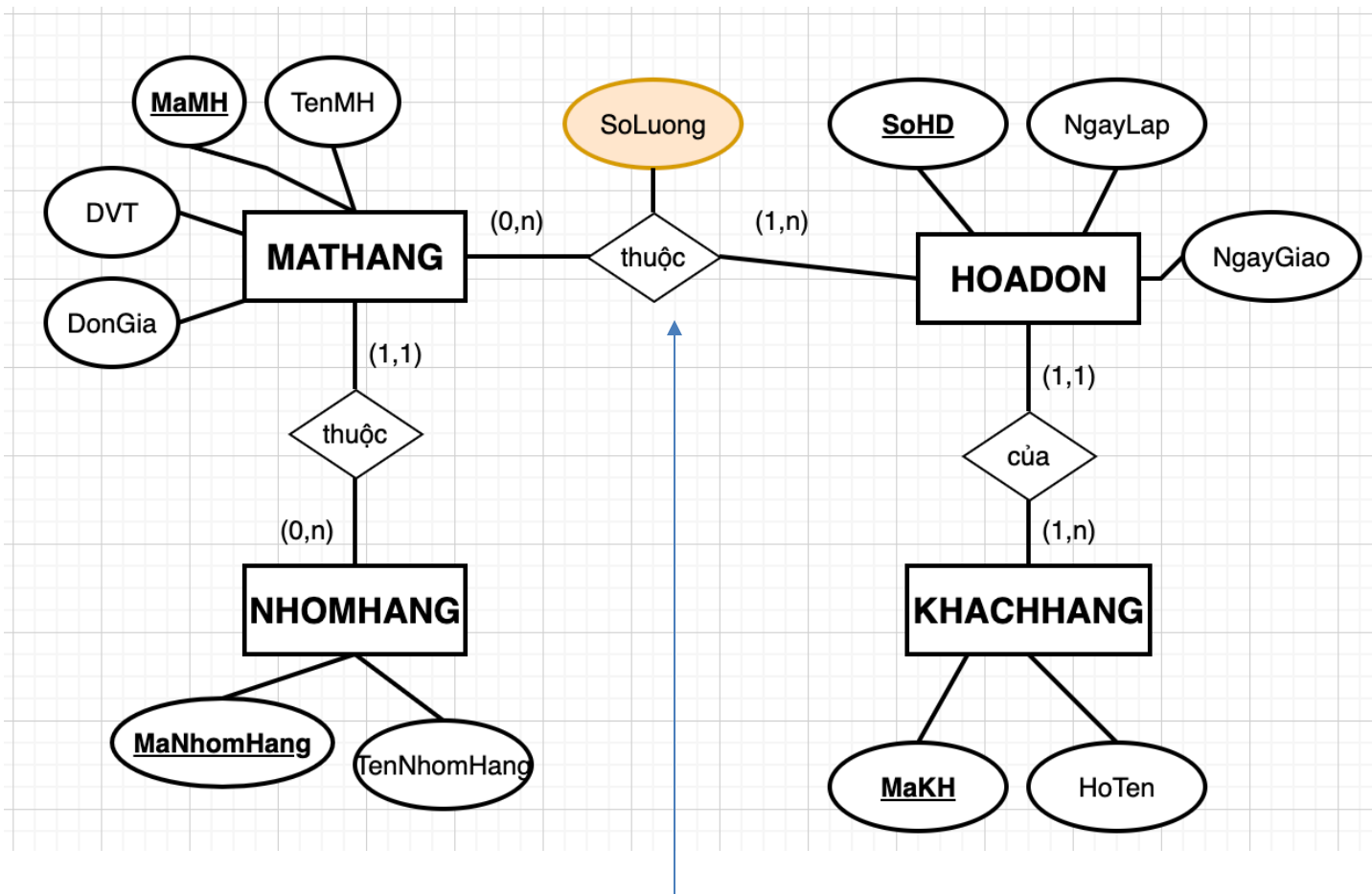
## Cách 3: chọn nhiều thuộc tính

PART (Part\_no, Description, **Mflag**, Drawing\_no, Manufacture\_date, Batch\_no, **Pflag**, Supplier\_name, List\_price)

Sử dụng 2 thuộc tính type (giúp nhận diện một thực thể thuộc về subset nào)

# BÀI TẬP 1

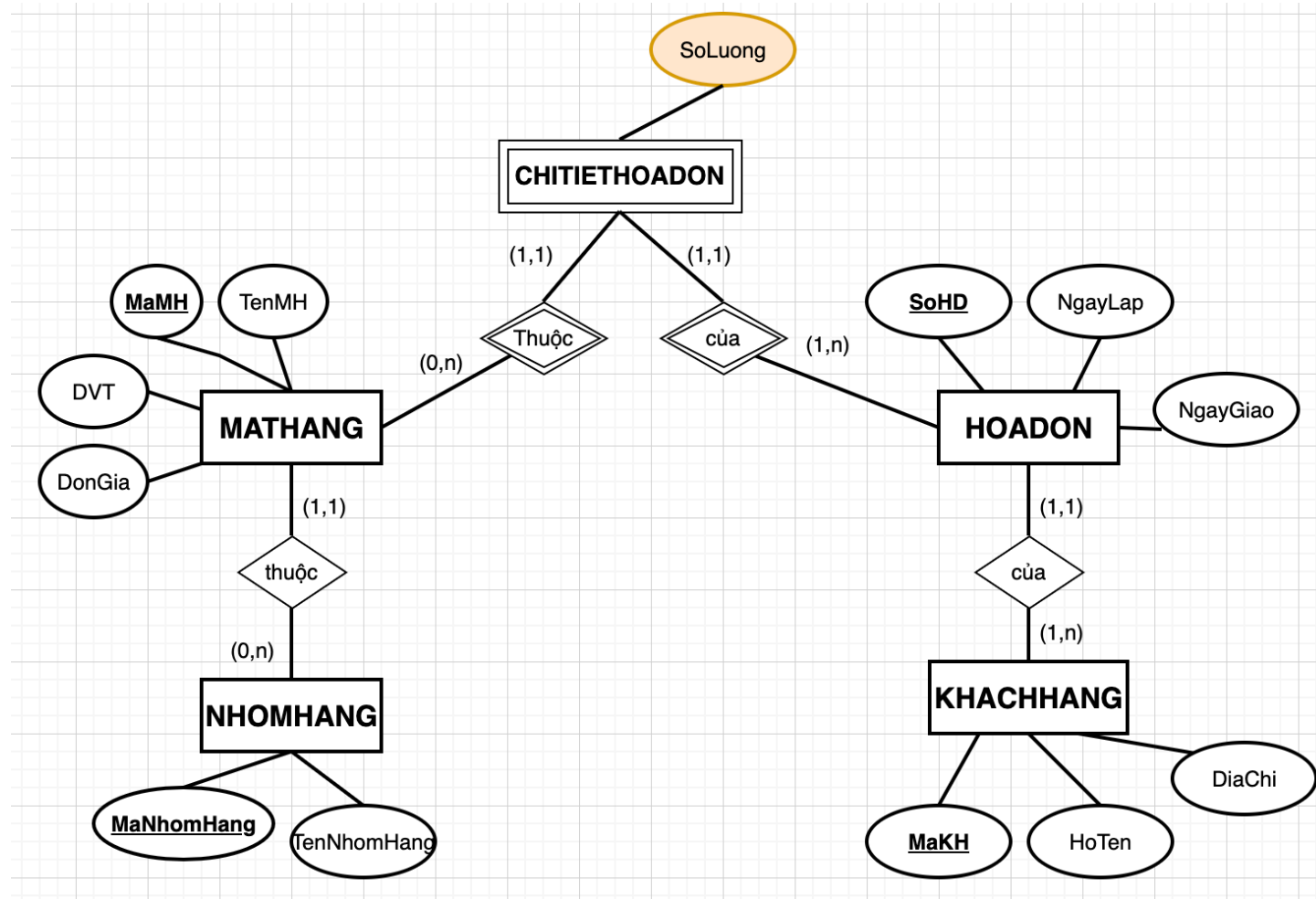
Dựa trên mô hình ER như hình sau, hãy đưa ra thiết kế CSDL quan hệ?



Chuyển thành tập thực thể kết hợp

# BÀI TẬP 1

Dựa trên mô hình ER như hình sau, hãy đưa ra thiết kế CSDL quan hệ?



# ĐÁP ÁN BÀI TẬP 1

## Kết quả: Lược đồ CSDL quan hệ

NhomHang (MaNhomHang, TenNhomHang)

MatHang (MaMH, TenMH, DVT, DonGia, MaNhomHang)

FK = {MaNhomHang} references NhomHang(MaNhomHang)

KhachHang(MaKH, HoTen, DiaChi)

HoaDon (SoHD, NgayLap, NgayGiao, MaKH)

FK = {MaKH} references KhachHang(MaKH)

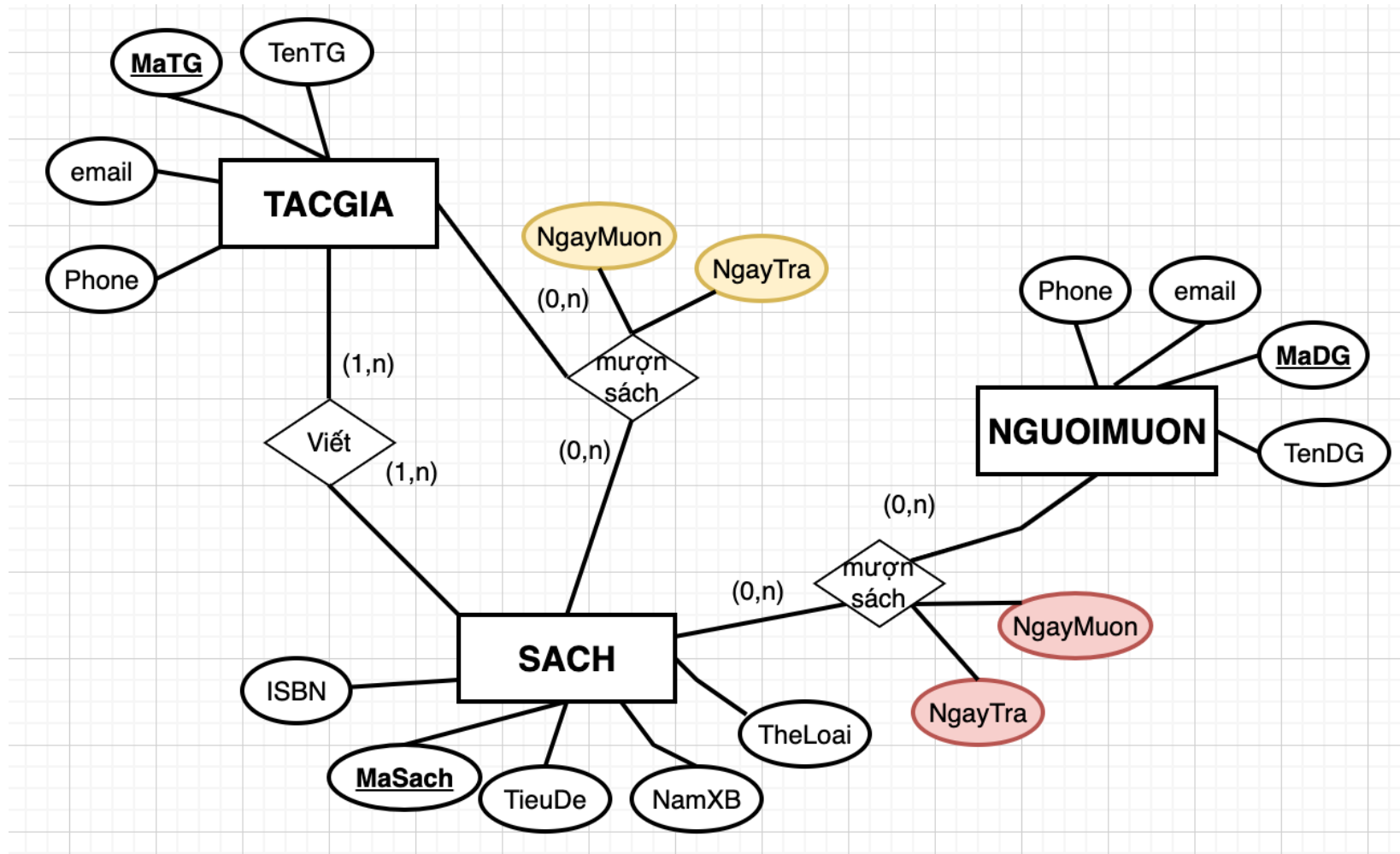
ChiTietHoaDon (MaMH, SoHD, SoLuong)

FK1 = {MaMH} references MatHang(MaMH)

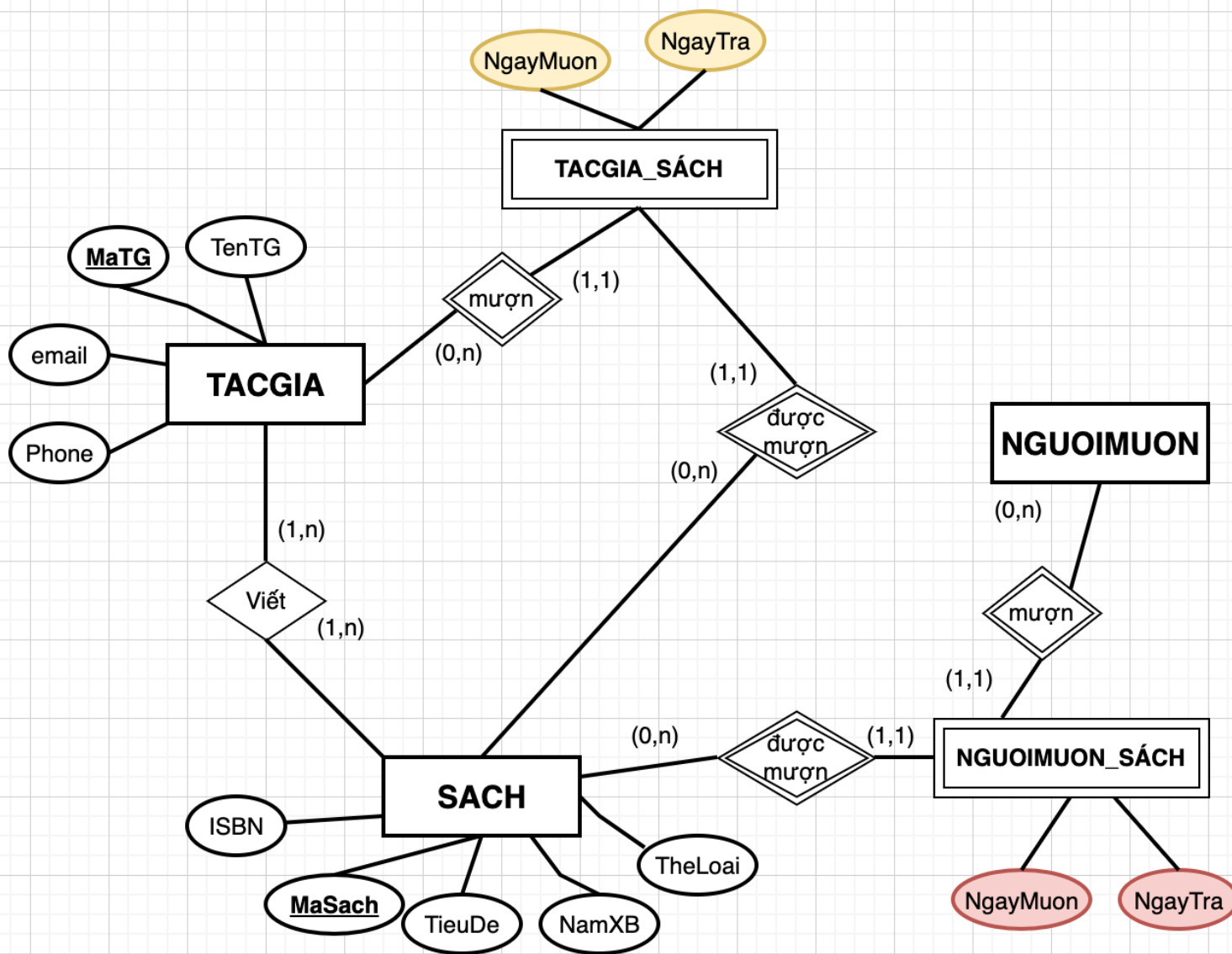
FK2 = {SoHD} references HoaDon(SoHD)

# BÀI TẬP 2

Dựa trên mô hình ER như hình sau, hãy đưa ra thiết kế CSDL quan hệ?



# BÀI TẬP 2



# ĐÁP ÁN BÀI TẬP 2

**Kết quả: Lược đồ CSDL quan hệ**

TacGia (**MaTG**, TenTG, Email, Phone)

Sach (**MaSach**, ISBN, TieuDe, NamXB, TheLoai)

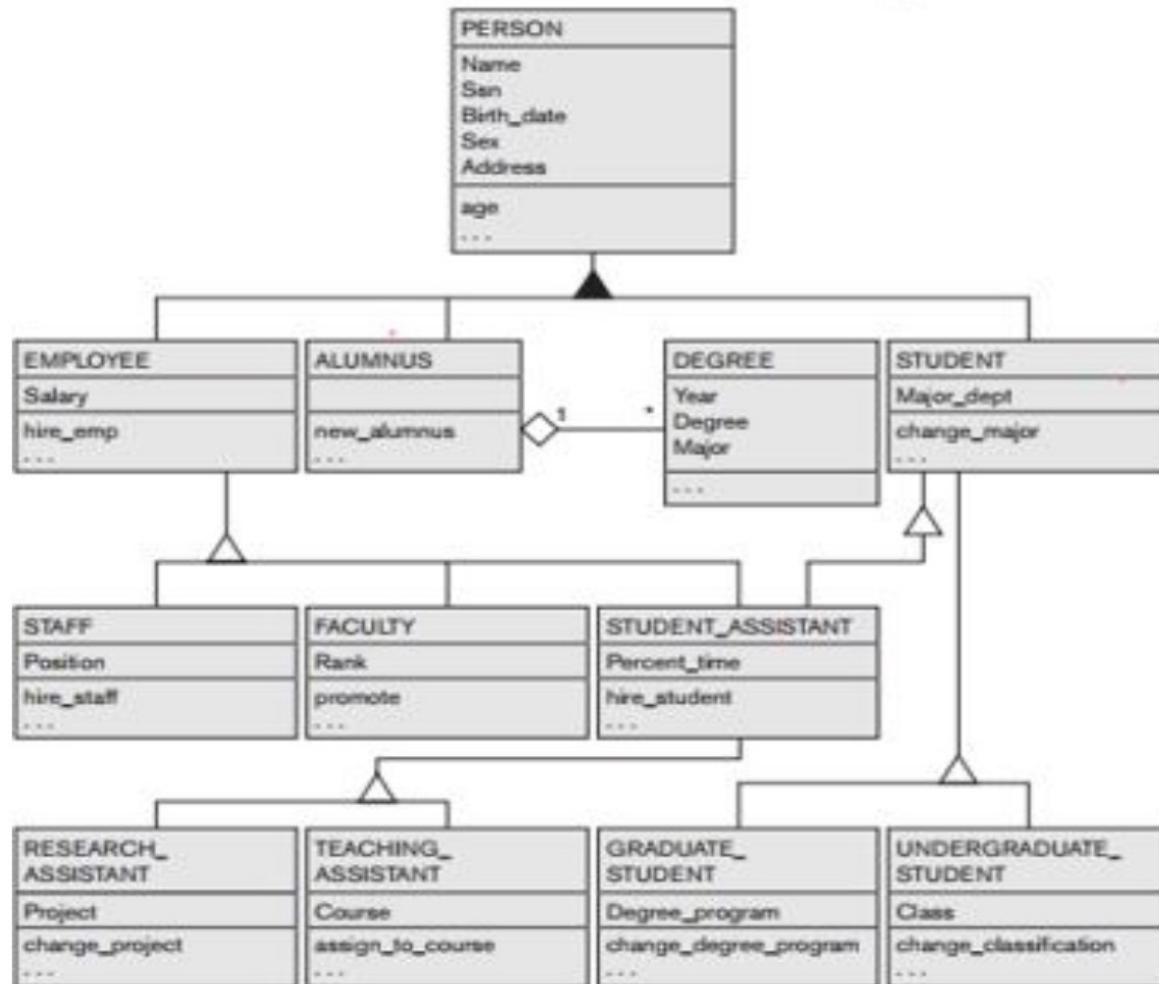
NguoiMuon (**MaDG**, TenDG, Email, Phone)

TacGia\_Sach (**MaTG**, **MaSach**, NgayMuon, NgayTra) # tác giả mượn sách

NguoiMuon\_Sach (**MaDG**, **MaSach**, NgayMuon, NgayTra) # độc giả mượn sách

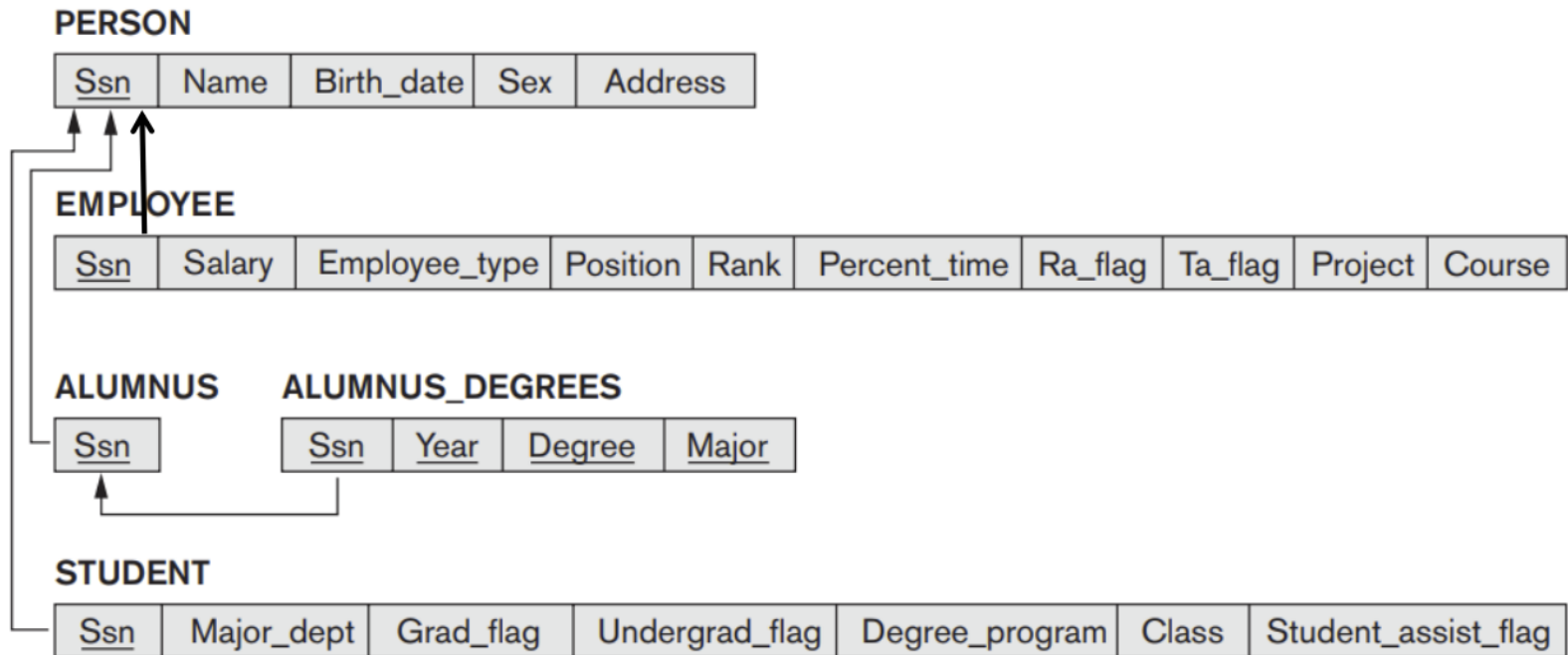
TG\_S (**MaTG**, **MaSach**) # tác giả viết nhiều cuốn sách

# BÀI TẬP 3

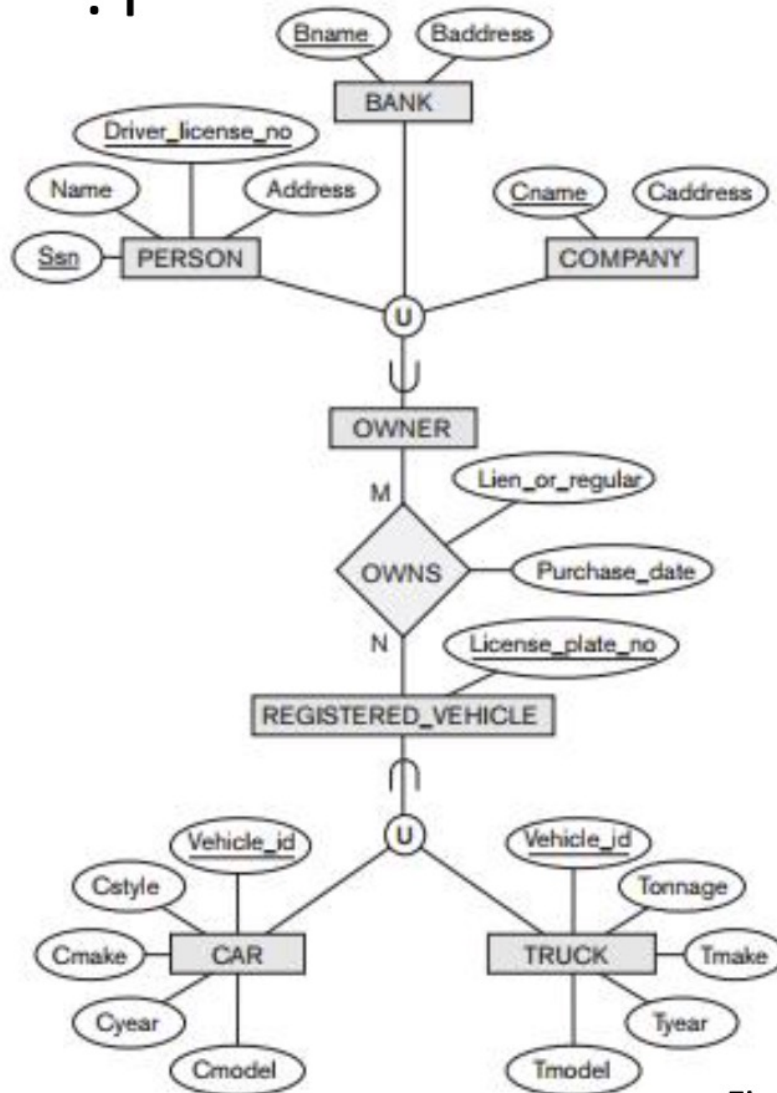




# ĐÁP ÁN BÀI TẬP 3



# Bài tập 3



## PERSON

<u>Ssn</u>	Driver_license_no	Name	Address	Owner_id
------------	-------------------	------	---------	----------

## BANK

<u>Bname</u>	Baddress	Owner_id
--------------	----------	----------

## COMPANY

<u>Cname</u>	Caddress	Owner_id
--------------	----------	----------

## OWNER

<u>Owner_id</u>
-----------------

## REGISTERED\_VEHICLE

<u>Vehicle_id</u>	License_plate_number
-------------------	----------------------

## CAR

<u>Vehicle_id</u>	Cstyle	Cmake	Cmodel	Cyear
-------------------	--------	-------	--------	-------

## TRUCK

<u>Vehicle_id</u>	Tmake	Tmodel	Tyear	Ttonnage
-------------------	-------	--------	-------	----------

## OWNS

<u>Owner_id</u>	<u>Vehicle_id</u>	Purchase_date	Lien_or_regular
-----------------	-------------------	---------------	-----------------

Figure 8.8, page 259

