Chương 4

MÔ HÌNH QUAN HỆ - RELATIONAL MODEL

Nội dung

- Các định nghĩa cơ bản của mô hình dữ liệu quan hệ
  - Quan hệ
  - Lược đồ quan hệ
- Quan hệ toán học
  - Lược đồ quan hệ và quan hệ
  - Khóa quan hệ
- Ràng buộc toàn vẹn
- Chuyển mô hình thực thể kết hợp sang mô hình quan hệ

Mô hình dữ liệu quan hệ (Relational data model)

- Mô hình Dữ liệu Quan hệ (*Relational Data Model*) dựa trên khái niệm quan hệ. Được đề xuất bởi E.F. Codd năm 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, khóa chính, khóa ngoại, ...
  - Được xây dựng trên cơ sở của phép toán tập hợp và phép toán quan hệ.
  - Ràng buộc toàn vẹn quan hệ.
- Các HQTCSDLQH được xây dựng dựa theo lý thuyết mô hình quan hệ

Mô hình dữ liệu quan hệ (Relational data model)

- **Quan hệ (Relation)**: là một bảng giá trị gồm hai thành phần: lược đồ quan hệ (*relation schema)* và thể hiện của quan hệ (*relation instance)*

Thuộc tính (Attribute)
Trường (Field)

MaNV	HoTenNV	Tuoi
NV01	Nguyễn Trung Tiến	25
NV02	Trần Thị Yến	35
NV03	Nguyễn Văn An	40

Bộ (Tuple)
Mẩu tin (Record)

Mô hình dữ liệu quan hệ (Relational data model)

- ***Lược đồ quan hệ*** (*relation schema):* xác định tên của quan hệ, tên và kiểu dữ liệu của thuộc tính của quan hệ. Ký hiệu: R(A1, A2, …, An).
  - A1, A2, …, An: danh sách các thuộc tính.
  - R là tên của quan hệ

  NHANVIEN(MaNV, HoTenNV, Tuoi)

MaNV	HoTenNV	Tuoi
NV01	Nguyễn Trung Tiến	25
NV02	Trần Thị Yến	35
NV03	Nguyễn Văn An	40

Mô hình dữ liệu quan hệ (Relational data model)

- ***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: $T_{Student}$ là thể hiện của quan hệ STUDENT tại thời điểm hiện tại gồm có các bộ như sau:

Students(sid: string, name: string, login: string, age: integer , gpa: real)

FIELDS (ATTRIBUTES, COLUMNS)

Field names

TUPLES (RECORDS, ROWS)

sid	*name*	*login*	*age*	*gpa*
50000	Dave	dave@cs	19	3.3
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

Lược đồ quan hệ (relation schema)

- Cho $A_1, A_2, .., A_n$ là tên các thuộc tính thuộc các miền $D_1, D_2,.., D_n$
- Lược đồ quan hệ R với 1 tập thuộc tính $\{A_1, A_2, .., A_n\}$

  $R(A_1, A_2, .., A_n)$
- Gọi $U=\{A_1, A_2,.., A_n\}$ là tập các thuộc tính của lược đồ quan hệ R
- Mỗi lược đồ quan hệ luôn kèm một tân từ để diễn tả ý nghĩa của nó.
- Biểu diễn 1 lược đồ quan hệ:

  TÊN_QUAN_HỆ(thuộc tính 1, thuộc tính 2,...)

  Ví dụ : lược đồ quan hệ NHANVIEN có 4 thuộc tính

  NHANVIEN(MANV, Ten, MaPb, Luong)

  Tân từ: Mỗi nhân viên có một MaNV duy nhất dùng để nhận diện TenNV, MaPB và Lương.

Miền giá trị của thuộc tính (Domain)

- Domain : là tập giá trị hợp lệ của một thuộc tính trong một quan hệ.
  - Lược đồ quan hệ R(A1, A2, …, An)
  - D: miền giá trị của Ai ký hiệu dom(Ai)

- Ví dụ:
  - Thuộc tính Address của quan hệ STUDENT có domain là tập hợp các chuỗi.
  - Thuộc tính điểm có domain là các số thực trong phạm vi từ 0 đến 10.

Bậc của Quan hệ

- **Bậc (degree):** là số lượng các thuộc tính của quan hệ.

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 hay xoá các tuple (hàng).

Các tính chất của quan hệ (Relation)

Các tính chất của một quan hệ

- Giá trị đưa vào cột là đơn nhất
- Các giá trị trong cùng một cột phải thuộc cùng một miền giá trị (cùng kiểu).
- Thứ tự dòng cột tuỳ ý.
- **Không được phép có 2 dòng dữ liệu trùng nhau**

MaNV	HoTenNV	Tuoi	
NV01	Nguyễn Trung Tiến	25	t1
NV02	Trần Thị Yến	35	t2

Trực quan về Quan hệ

- Ví dụ : Để quản lý điểm sv, nếu ta lưu trữ theo bảng sau thì không đúng với mô hình quan hệ vì

Thuộc tính

Bộ

MASV	**TEN**	MALOP	**MAMH**	TENMH	ĐIEM
TCTH01	Sơn	TCTHA	THVP	Tin Học Văn Phòng	6.0
TCTH01	Sơn	TCTHA	CSDL	Cơ Sở Dữ Liệu	5.0
TCTH02	Lan	TCTHA	CSDL	Cơ Sở Dữ Liệu	3.0

- Ta không thể lưu trữ một SV mới khi SV này chưa có điểm thi.
- Khi cần sửa đổi ten thì phải sửa tất cả các dòng có liên quan đến SV này.
- Khi có nhu cầu xóa điểm thi một SV kéo theo khả năng xóa luôn họ tên Sv đó.
- Để lưu trữ đúng mô hình quan hệ, ta có thể lưu trữ thành các bảng như sau

Ví dụ về mô hình quan hệ

SVIEN

MASV	TEN	MALOP
TCTH01	Sơn	TCTHA
TCTH02	Bảo	TCTHB
TCTH03	Trang	TCTHA

LOP

MALOP	TENLOP	SISO
TCTHA	TCTH32A	80
TCTHB	TCTH32B	65
TCTHC	TCTH32C	82

MONHOC (COURSE)

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

KQUA (TRANNING)

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

Các ràng buộc toàn vẹn (Integrity constraints)

- Ràng buộc là những quy tắc được áp đặt lên trên dữ liệu đảm bảo *tính tin cậy* và *độ chính xác* của dữ liệu. Các luật toàn vẹn được thiết kế để giữ cho dữ liệu phù hợp và đúng đắn.
- 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 toàn vẹn - Referential integrity
  - 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 khóa chính và giá trị của khóa chính phải luôn hợp lệ.
  - "Trong một quan hệ cơ sở, mọi thuộc tính khóa chính không được có giá trị rỗng (null)"
- Được thể hiện trong SQL server thông qua các ràng buộc sau:
  - Ràng buộc khóa chính (primary key)
  - Ràng buộc duy nhất (unique)
  - Chỉ mục (index)
  - Thuộc tính identity

Ràng buộc khóa (Key Constraints)

- **Ràng buộc khóa (Key Constraints)**: 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.
- **Khóa (key):** Một thuộc tính hoặc một tập các thuộc tính dùng để xác định một dòng trong một quan hệ. Khóa được chia thành 3 loại:
  - Siêu khóa (Super key)
  - Khóa dự tuyển (Candidate key)
  - Khóa chính (Primary key)

Ràng buộc khóa (Key Constraints)

- Siêu khoá (superkey) là một thuộc tính hoặc một tập các thuộc tính dùng để xác định duy nhất một bộ của quan hệ
  - Một siêu khoá có thể chứa thêm các thuộc tính không cần thiết để xác định duy nhất một bộ
  - Siêu khoá hiển nhiên là tập tất cả các thuộc tính của quan hệ

Cho hai bộ bất kỳ t1 và t2 trong r(R) thì ta có $t1[SK] \neq t2[SK]$.

Khoá quan hệ (Relational Keys)

- Ví dụ: Với quan hệ LOPHOC (MaLop, TenLop, SoSV, MaKhoa, MaGVCN)
- Siêu khoá :
  - K1 = {Malop}
  - K2={MaLop, TenLop}
  - K3 = {MaGVCN}
  - K4={MaLop, SoSV}
  - K5= {MaLop, MAGVCN}
  - K6={Malop,TenLop, MaKhoa}
  - K7= {MaLop, SoSV, MaKhoa}
  - K8= {Malop, TenLop, SoSV, MaKhoa} →Khóa hiển nhiên
- Bài tập: Xét 1 lược đồ quan hệ

  STUDENT (StudentID, FirstName, LastName, BirthDate, Major, SSN, ClassID)
  - Xác định các siêu khóa có thể có của R???

Ràng buộc khóa (Key Constraints)

- Khoá dự tuyển (candidate key): là một siêu khoá K mà không có một tập con thực sự bất kỳ K' ⊂ 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' ⊂ K lại có tính duy nhất.
- Ví dụ: Với quan hệ LOPHOC (MaLop, TenLop, SoSV, MaKhoa, MaGVCN)
- Siêu khoá :
  - K1 = {Malop}
  - K2={MaLop, MAGVCN}
  - K3 = {MaGVCN}
  - K4= {MaLop, TenLop}

K1 là siêu khóa vì chỉ có một thuộc tính đảm bảo tính duy nhất và tối giản nên K1 là khóa dự tuyển

Xét K2={Malop, MaGVCN}

K2 không là khóa dự tuyển vì tồn tại một K'=Malop ⊂ K2 mà Malop lại là một siêu khóa.

Ràng buộc khóa (Key Constraints)

- Khóa dự tuyển cần thỏa mãn 2 tính chất sau:
  - Xác định duy nhất.
  - Không dư thừa: Khi xóa đi bất kỳ một thuộc tính nào của khóa đều phá hủy tính xác định duy nhất của khóa.
- Ví dụ: tìm khóa dự tuyển của quan hệ NHANVIEN(Manv, HoTen, CMND, BSXe, Phai, NgaySinh, NgayVaoLam)

Ràng buộc khóa (Key Constraints)

- **Khóa chính (primary key):** là một khóa dự tuyển được chọn để xác định duy nhất một bộ của quan hệ. Khóa chính có thể chỉ có 1 thuộc tính hay khóa phức hợp.
- Ví dụ: Ví dụ: Với quan hệ NHANVIEN (Manv, TenNV, PHAI, CMND, MATHEBHYT)
- Khoá dự tuyển là:
  - K1 = {Manv} K2={CMND} K3={MATHEBHYT}

  →Khóa chính là MaNV
- Ví dụ 2: R={SSN, Name, BDate, Address, Salary}

  SSN →Name, BDate, Address, Salary

  (Nguồn) → (Đích)
- Ta thấy, từ SSN ta có thể suy ra toàn bộ các thuộc tính ứng. Vậy SSN được gọi là khóa chính.

Các loại khoá quan hệ (Relational Keys)

Một số gợi ý khi chọn khóa:

- Khóa không nên là tập hợp của quá nhiều thuộc tính. Trong trường hợp khóa có nhiều thuộc tính, có thể thêm một thuộc tính "nhân tạo" thay chúng làm khóa chính cho quan hệ.
- Nếu khóa chính được cấu thành từ một số thuộc tính, thì các thành phần nên tránh sử dụng thuộc tính có giá trị thay đổi theo thời gian: như tên địa danh, phân loại
- Ví dụ: khóa chính của quan hệ SINHVIEN???

SINHVIEN(Masv, Tensv, Ngaysinh, Phai, Malop)

Các loại khoá quan hệ (Relational Keys)

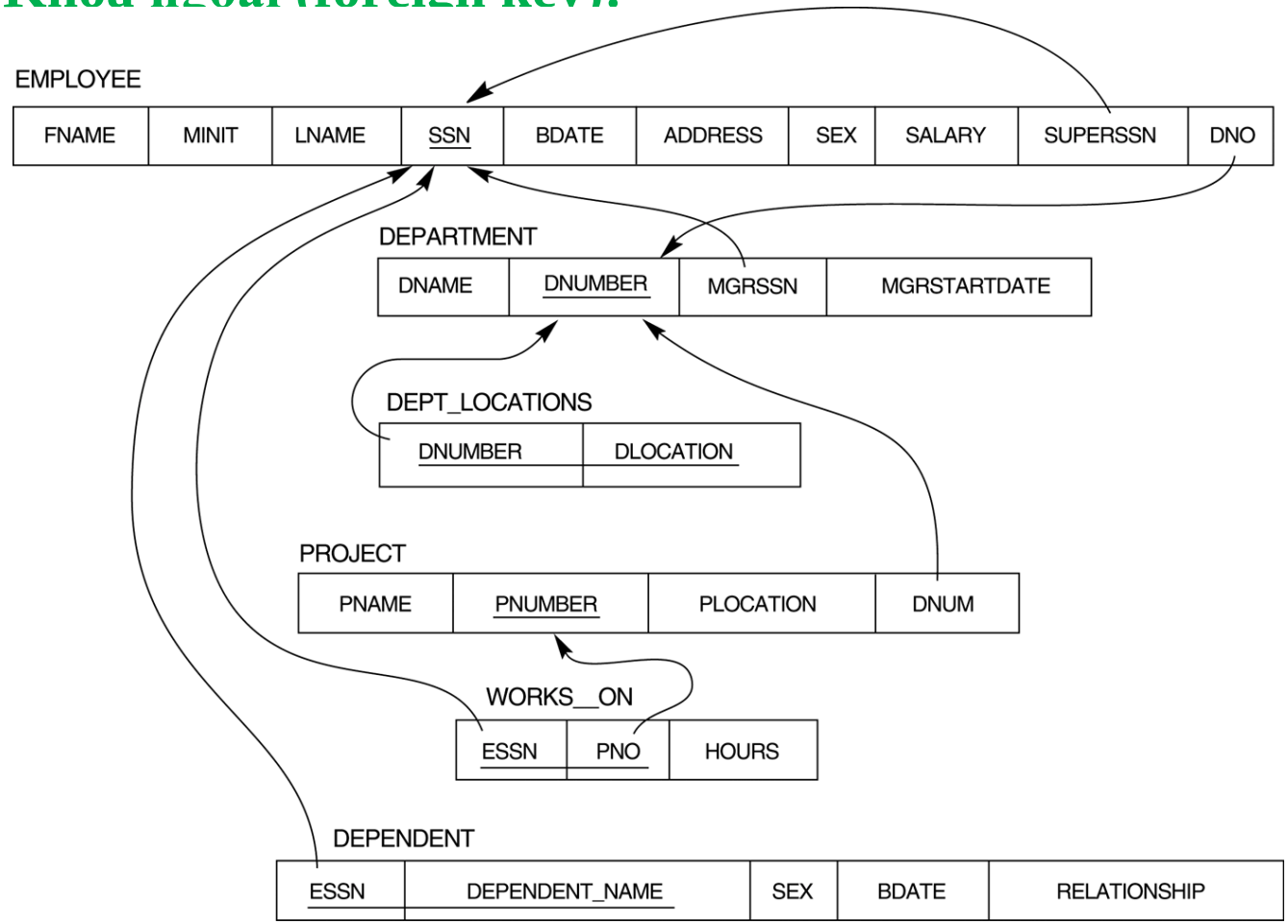
- **Khóa ngoại (foreign key):** tập các trường (field) hoặc thuộc tính trong một quan hệ được sử dụng để "tham chiếu" đến một bộ trong quan hệ khác.

TENPHG	MAPHG
Nghien cuu	**5**
Dieu hanh	4
Quan ly	1

TENNV	HONV	NGSINH	DCHI	PHAI	LUONG	PHONG
Tung	Nguyen	12/08/1955	638 NVC Q5	Nam	40000	**5**
Hang	Bui	07/19/1968	332 NTH Q1	Nu	25000	4
Nhu	Le	06/20/1951	291 HVH QPN	Nu	43000	4
Hung	Nguyen	09/15/1962	Ba Ria VT	Nam	38000	5

Các loại khoá quan hệ (Relational Keys)

- **Khóa ngoại (foreign key):**



Các loại khoá quan hệ (Relational Keys)

- 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,…)

Mô hình dữ liệu quan hệ (Relational data model)

Bài tập: 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)

Mô hình dữ liệu quan hệ (Relational data model)

Bài tập: 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)

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

- Tính toàn vẹn tham chiếu được thể hiện thông qua 3 quy tắc sau:
  - Không thể thêm các bản ghi vào bảng quan hệ nếu không có bản ghi tương ứng trong bảng chính.
  - Không thể thay đổi giá trị trong bảng chính nếu làm cho các bản ghi tương ứng trong bảng quan hệ bị mất tham chiếu
  - Không thể xóa các bản ghi trong bảng chính nếu nó được tham chiếu bởi 1 số bản ghi trong bảng quan hệ

SVIEN

MASV	TEN	MALOP
TCTH01	Sơn	TCTHA
TCTH02	Bảo	TCTHB
TCTH03	Trang	TCTHA

LOP

MALOP	TENLOP	SISO
TCTHA	TCTH32A	80
TCTHB	TCTH32B	65
TCTHC	TCTH32C	82

n 1

Toàn vẹn tham chiếu (Reference 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"

Ví dụ về mô hình quan hệ

SVIEN

MASV	TEN	MALOP
TCTH01	Sơn	TCTHA
TCTH02	Bảo	TCTHB
TCTH03	Trang	TCTHA

LOP

MALOP	TENLOP	SISO
TCTHA	TCTH32A	80
TCTHB	TCTH32B	65
TCTHC	TCTH32C	82

N

1

KQUA

MONHOC

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

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

1

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 về bất thường dữ liệu

- 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)

Chuyển từ mô hình ER sang mô hình Quan hệ

Sự tương ứng với mô hình ER

- Mỗi quan hệ (bảng) tương ứng với một kiểu thực thể hoặc với một kiểu mối liên kết nhiều - nhiều.
- Mỗi hàng tương ứng với một thể hiện thực thể hoặc với một thể hiện mối liên kết nhiều - nhiều.
- Mỗi cột tương ứng với một thuộc tính.
- Từ *quan hệ* (*relation*) trong cơ sở dữ liệu quan hệ không có cùng nghĩa với từ *mối quan hệ* (*relationship*) trong mô hình *ER*

Chuyển đổi từ ERD thành các quan hệ

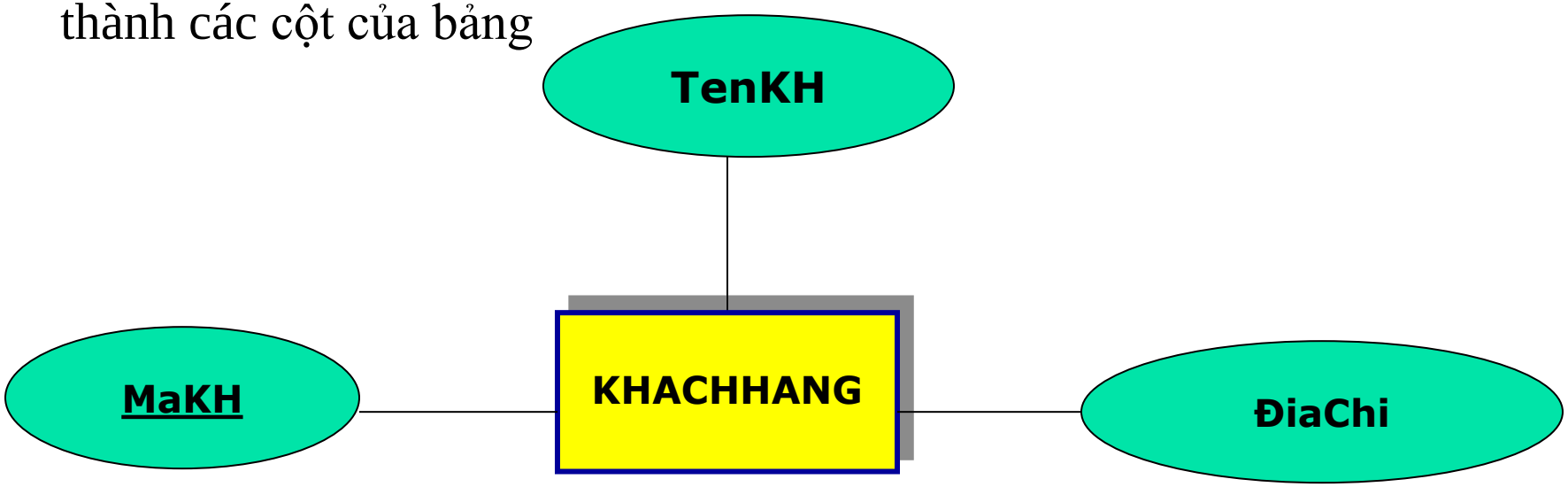
Các bước chuyển từ ERD sang mô hình quan hệ

- Bước 1: chuyển đổi kiểu thực thể thông thường (regular entity)
- Bước 2: Chuyển đổi kiểu thực thể yếu (weak entity)
- Bước 3: Chuyển đổi mối liên kết 2 ngôi
- Bước 4: Chuyển đổi kiểu thực thể kết hợp
- Bước 5: chuyển đổi mối liên kết 1 ngôi
- Bước 6: chuyển đổi mối liên kết 3 ngôi
- Bước 7: chuyển đổi mối liên kết siêu kiểu / kiểu con

Chuyển đổi từ ERD thành các quan hệ

Bước 1: chuyển đổi kiểu thực thể thông thường (regular entity)

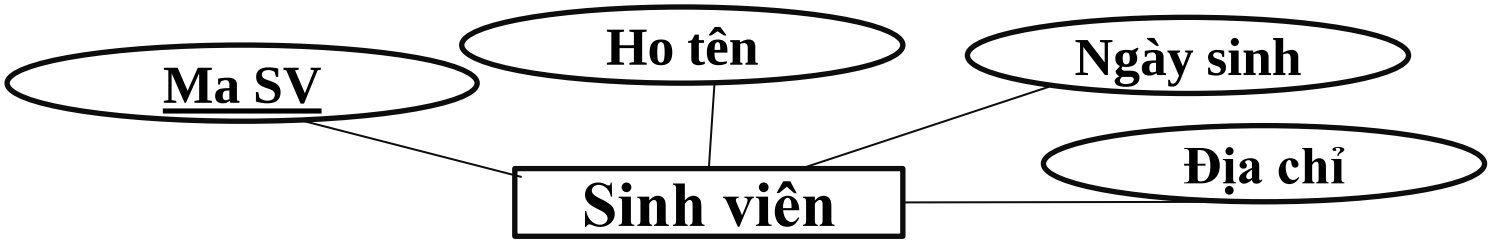
- Mỗi kiểu thực thể thông thường (không phải kiểu thực thể yếu) được chuyển thành 1 quan hệ (bảng). Khoá chính của kiểu thực thể trở thành khoá chính của bảng. Các thuộc tính của kiểu thực thể trở thành các cột của bảng



KHACHHANG(MaKH, TenKH, ĐiaChi)

Chuyển đổi từ ERD thành các quan hệ

Bước 1: chuyển đổi kiểu thực thể thông thường (regular entity)



- Chuyển thành **quan hệ Sinhvien**

  Sinhvien (MaSV, Hoten, Ngaysinh, Diachi)

- Hoặc biểu diễn dưới dạng:

MaSV	Hoten	Ngaysinh	Diachi

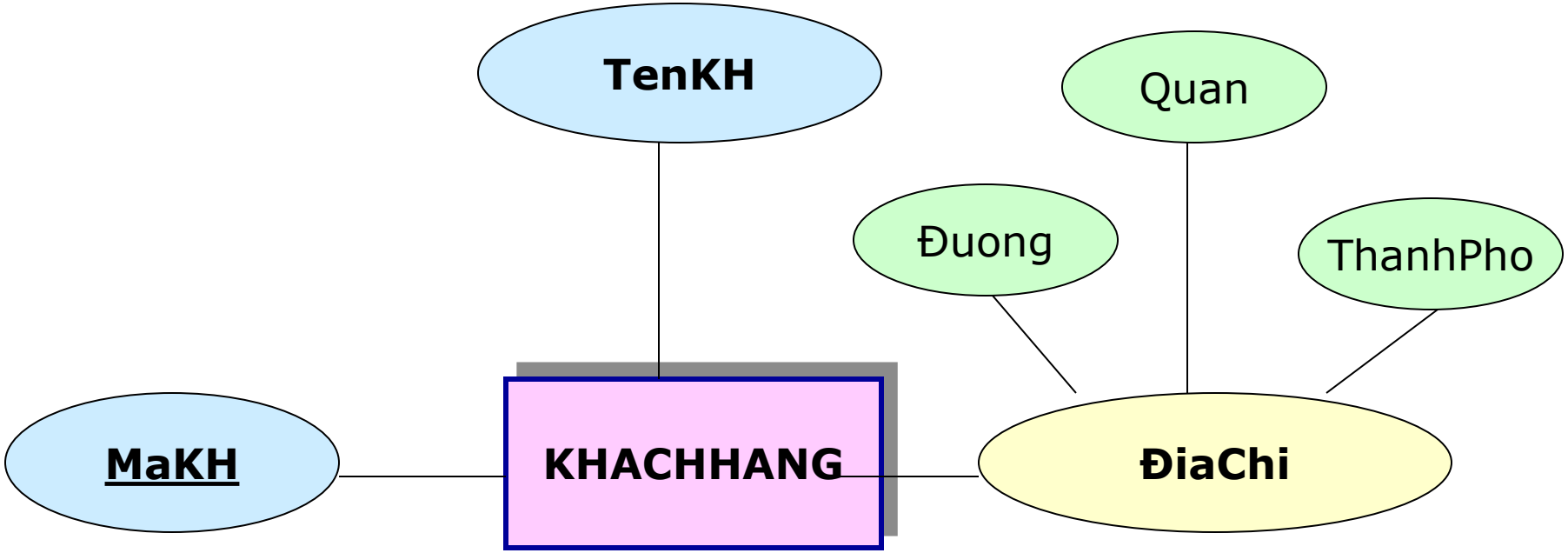
Chuyển đổi từ ERD thành các quan hệ

Bước 1: chuyển đổi kiểu thực thể thông thường (regular entity)

- **Với thuộc tính phức hợp:** các thành phần sẽ được chuyển thành các thuộc tính trong quan hệ
- **Với thuộc tính đa trị:** thực thể sẽ được chuyển thành 2 quan hệ
  - Một quan hệ chứa mọi thuộc tính trừ thuộc tính đa trị
  - Một quan hệ có 2 thuộc tính: 1 khoá và 1 thuộc tính đa trị, khóa chính là 2 thuộc tính đó

Chuyển đổi từ ERD thành các quan hệ

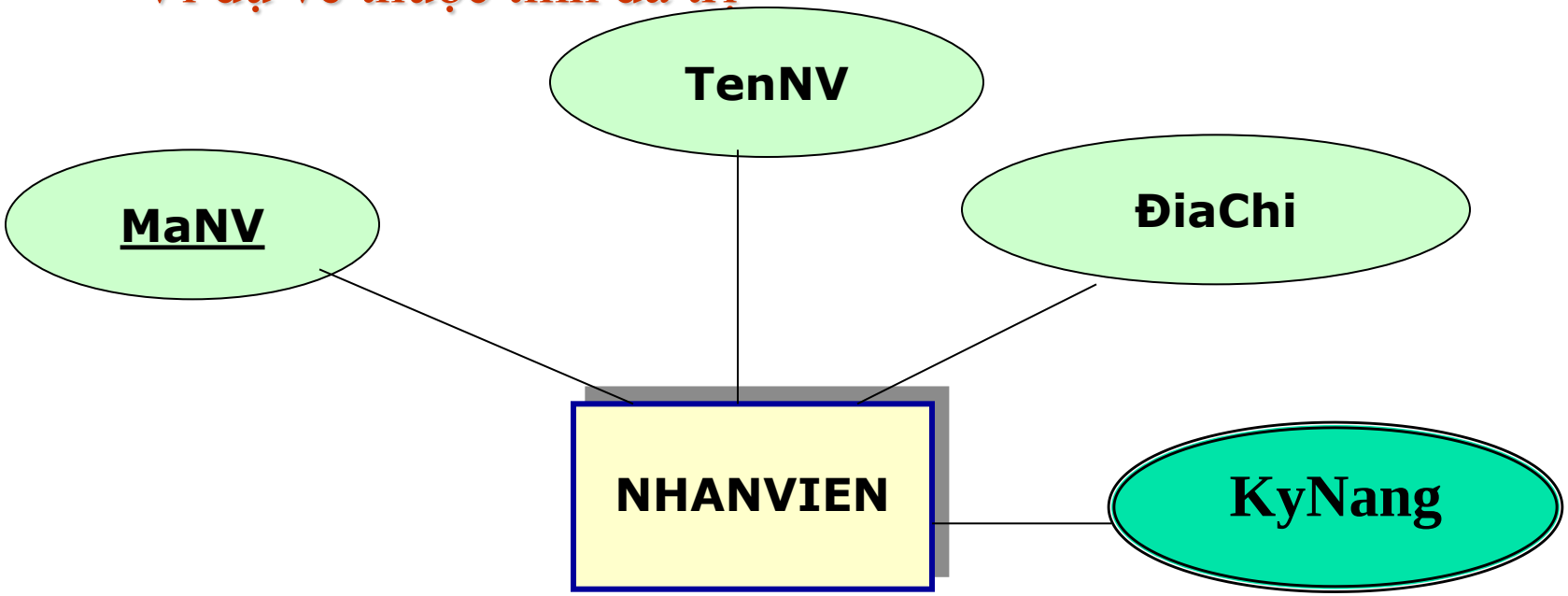
Ví dụ về thuộc tính phức hợp



KHACHHANG(<u>MaKH</u>, TenKH, Đuong, Quan, ThanhPho)

Chuyển đổi từ ERD thành các quan hệ

Ví dụ về thuộc tính đa trị

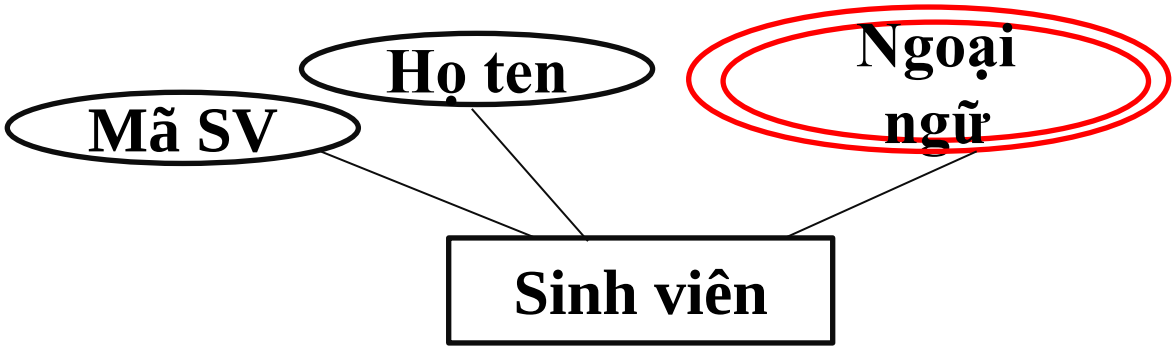


NHANVIEN(MaNV, TenNV, ĐiaChi)

NHANVIEN_KYNANG(MaNV, KyNang)

Chuyển đổi từ ERD thành các quan hệ

Bài tập về thuộc tính đa trị



Chuyển sang mô hình Quan hệ

Sinhvien (**MaSV,** Hoten)

NgoaiNguSV (**Ngoaingu, MaSV**)

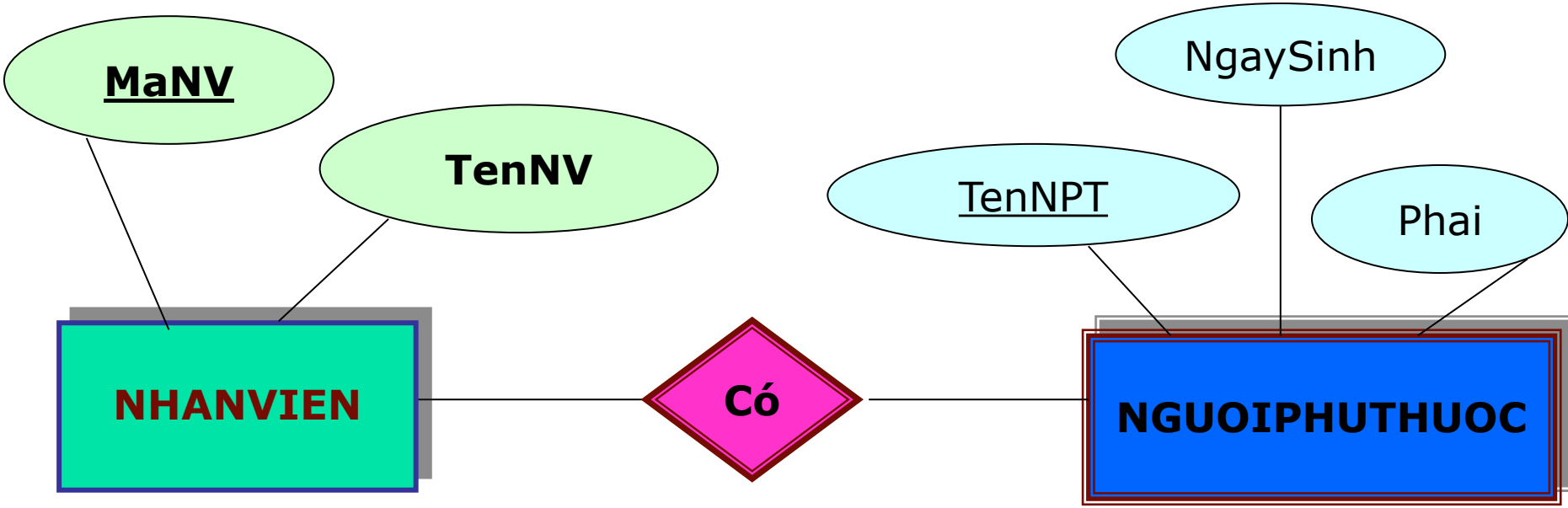
Chuyển đổi từ ERD thành các quan hệ

Bước 2: Chuyển đổi kiểu thực thể yếu (weak entity)

- Mỗi thực thể yếu chuyển thành 1 quan hệ. Thuộc tính xác định của kiểu thực thể chủ được đưa vào như là 1 thuộc tính của quan hệ.
  - **Khóa chính cuả quan hệ = thuộc tính xác định của thực thể chủ + thuộc tính xác định của thực thể yếu**

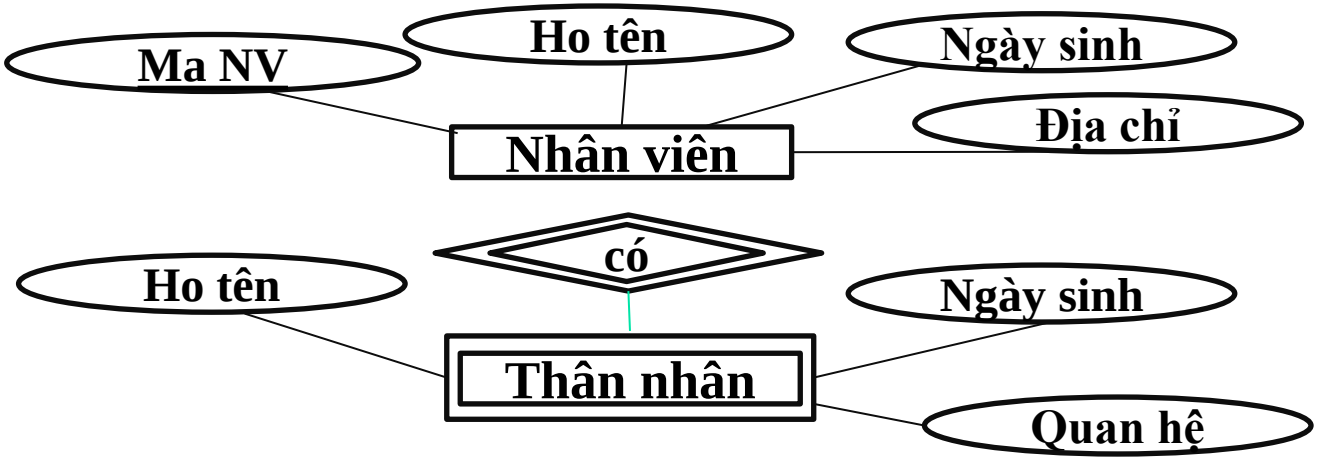
Chuyển đổi từ ERD thành các quan hệ

Ví dụ về thực thể yếu



NHANVIEN(MaNV, TenNV)
NGUOIPHUTHUOC(TenNPT, MaNV, NgaySinh, Phai)

Chuyển đổi từ ERD thành các quan hệ



Nhanvien (**MaNV**, Hoten, Ngaysinh, Diachi)

Thannhan (**MaNV, Hoten,** Ngaysinh, quanhe)

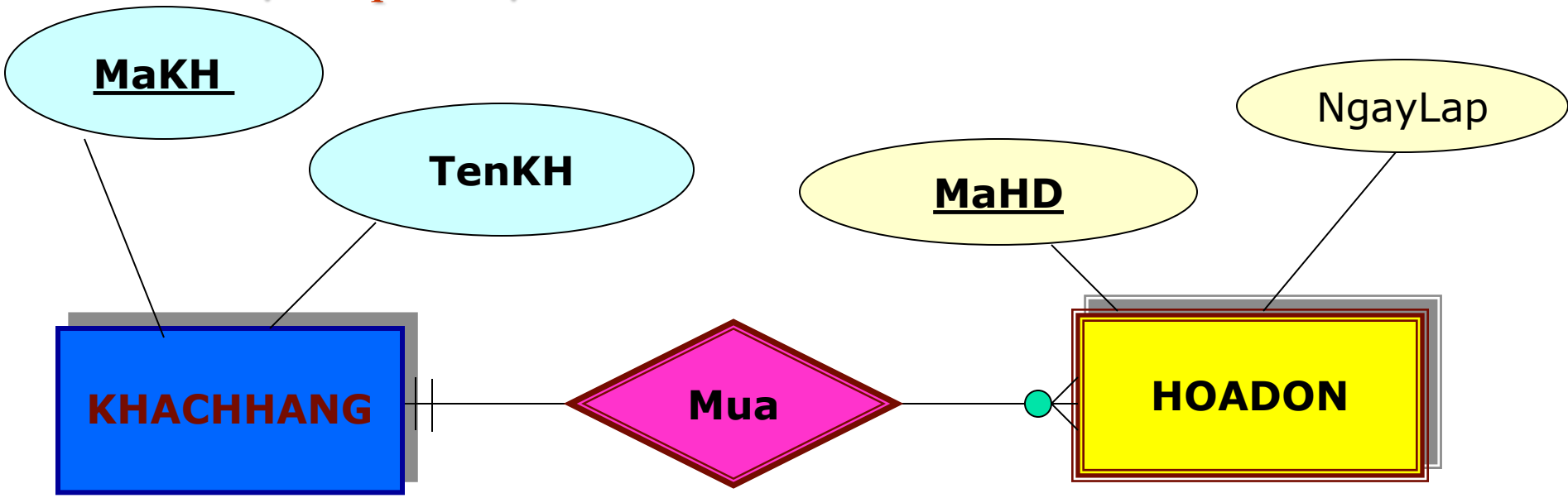
Chuyển đổi từ ERD thành các quan hệ

Bước 3: Chuyển đổi mối liên kết 2 ngôi

- Với mối liên kết 1-M: thuộc tính xác định của kiểu thực thể phía 1 được đưa vào làm khoá ngoại của kiểu thực thể phía M.

Chuyển đổi từ ERD thành các quan hệ

Ví dụ về quan hệ 1-M



KHACHHANG(MaKH, TenKH)
HOADON(MaHD, NgayLap, MaKH)

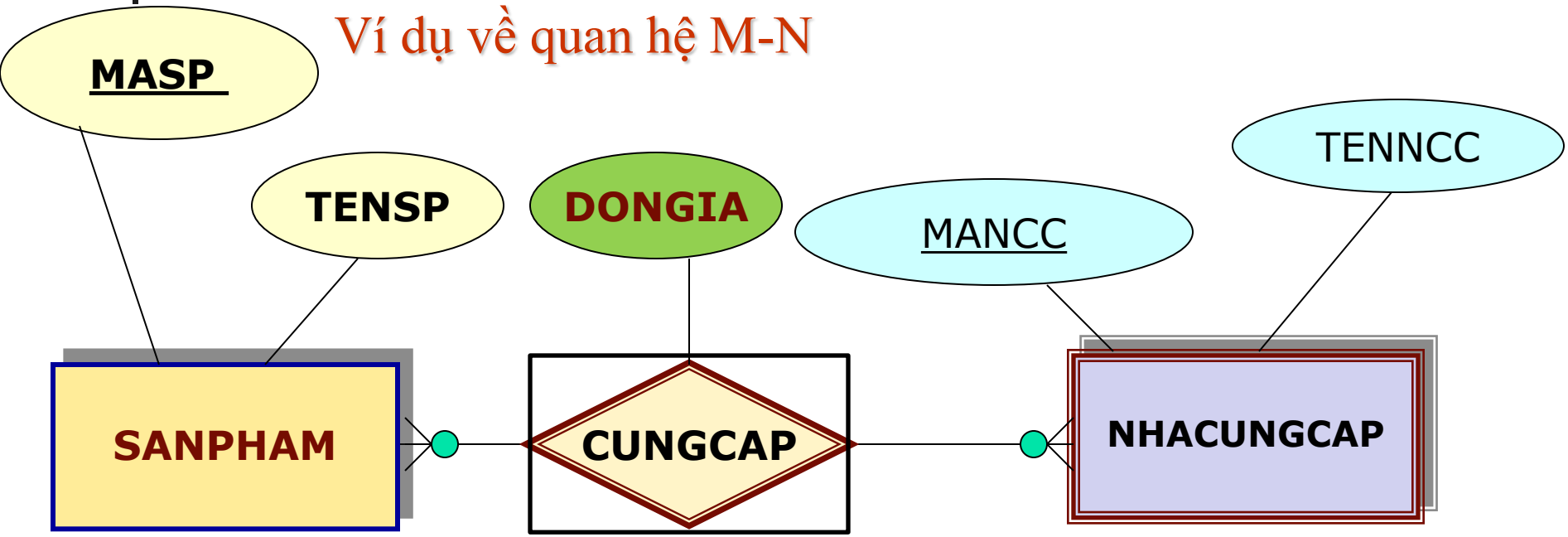
Chuyển đổi từ ERD thành các quan hệ

Bước 3: Chuyển đổi mối liên kết 2 ngôi

- Với mối liên kết M-N: một quan hệ mới được tạo ra. Khoá chính của quan hệ mới bao gồm khóa chính của 2 quan hệ tương ứng với kiểu thực thể tham gia vào mối liên kết.

Chuyển đổi từ ERD thành các quan hệ

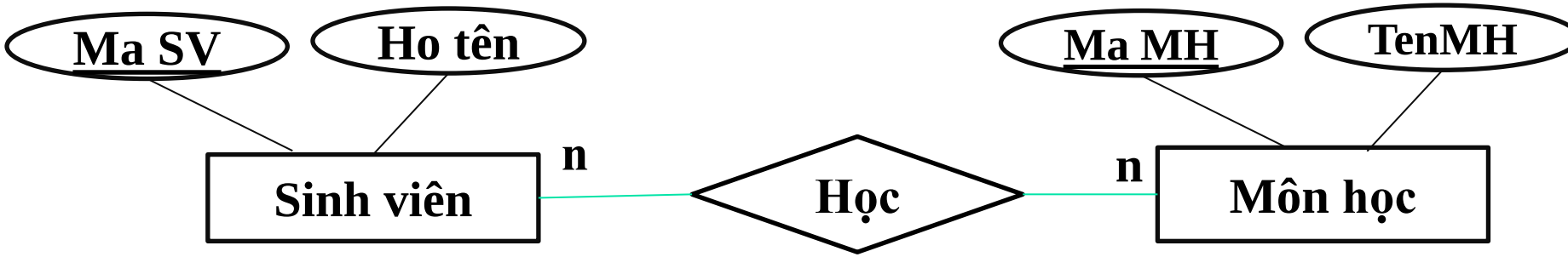
Ví dụ về quan hệ M-N



SANPHAM(MASP, TENSP)
CUNGCAP(MASP, MANCC, DONGIA)
NHACUNGCAP(MANCC, TENNCC)

Chuyển đổi từ ERD thành các quan hệ

Bài tập: quan hệ **Học** giữa thực thể **Sinh viên** và thực thể **Môn học**



Chuyển sang mô hình **Quan hệ**

Sinhvien(**MaSV**, Hoten)

Ketqua(**MaSV, MaMH,** Điem) *// thực thể kết hợp*

Monhoc(**MaMH**, TenMH)

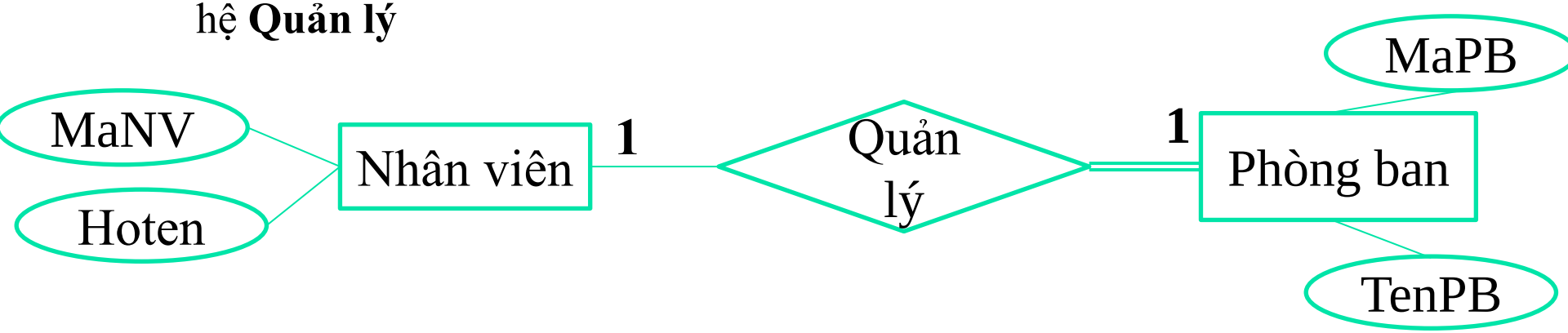
Chuyển đổi từ ERD thành các quan hệ

Bước 3: Chuyển đổi mối liên kết 2 ngôi

- Với mối liên kết 1-1: là 1 trường hợp đặc biệt của 1-M. Khoá chính của 1 trong 2 quan hệ sẽ được đưa vào làm khoá ngoại của quan hệ kia.
- Nếu 1 trong 2 đầu mối liên kết là nhiệm ý, thì nên đưa khoá chính của kiểu thực thể bắt buộc vào làm khoá ngoại của quan hệ tương ứng với kiểu thực thể nhiệm ý

Chuyển đổi từ ERD thành các quan hệ

Ví dụ về quan hệ 1-1: quan hệ **Quản lý** giữa thực thể **Nhân viên** và **Phòng ban.** **Vì** thực thể **Phòng ban** tham gia toàn phần vào mối quan hệ **Quản lý**



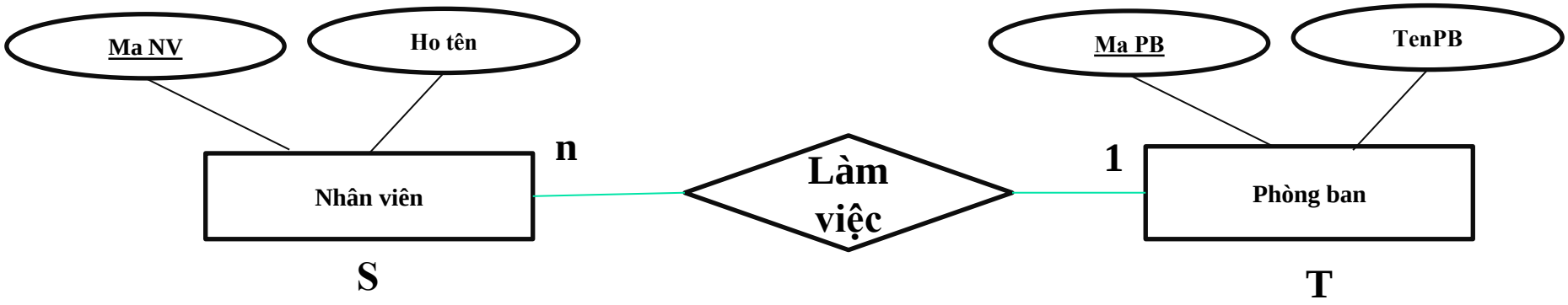
Chuyển sang mô hình **Quan hệ** có dạng**:**

Nhanvien(**MaNV**, Hoten)

PhongBan(**MaPB**, TenPB, MaNV)

Chuyển đổi từ ERD thành các quan hệ

Ví dụ về quan hệ 1-1: Quan hệ **Làm việc** giữa 2 loại thực thể **Nhân viên** và **Phòng ban**

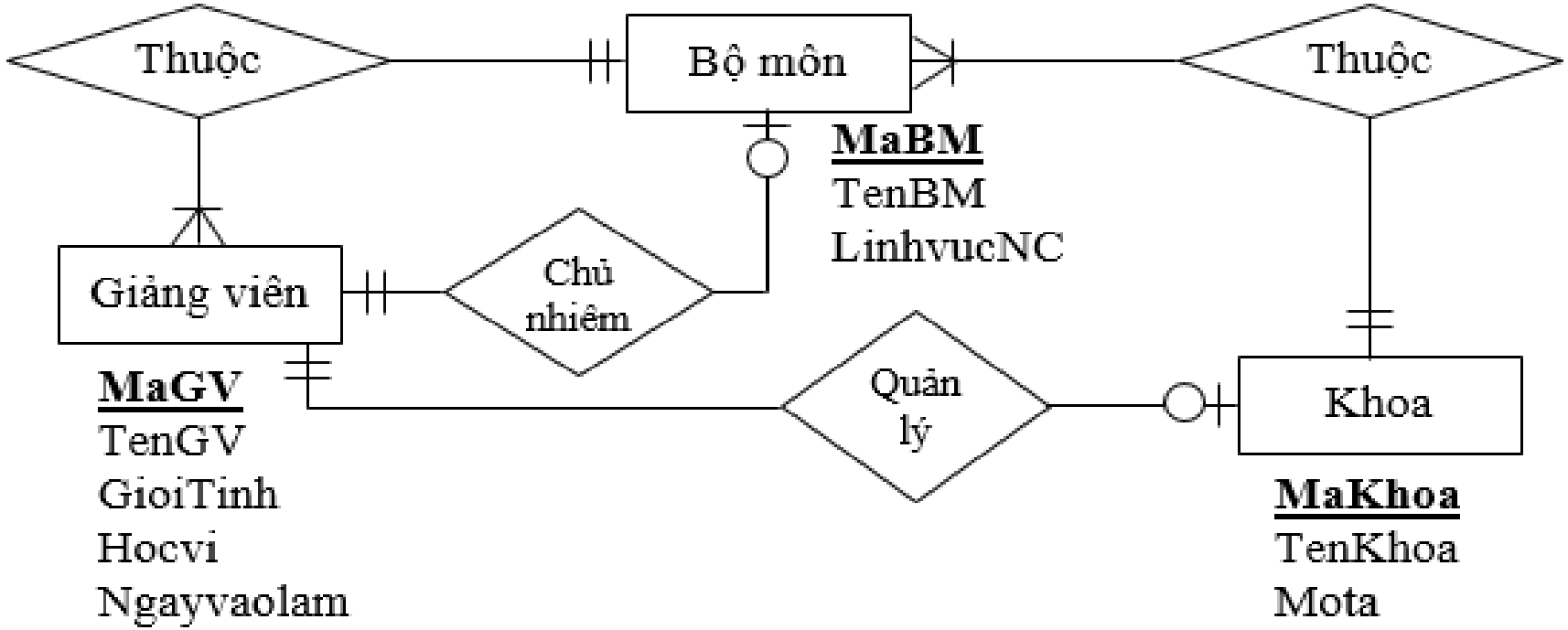


Chuyển sang mô hình **Quan hệ** có dạng**:**

Phongban(**MaPB**, TenPB)

Nhanvien(**MaNV**, Hoten, **MaPB**)

Chuyển đổi từ ERD thành các quan hệ



KHOA(MAKHOA, TENKHOA, MOTA, MAGV)
BOMON(MABM, TENBM, LINHVUCNC, MAGV, MAKHOA)
GIANGVIEN(MAGV, TENGV,…,EMAIL, MABM)

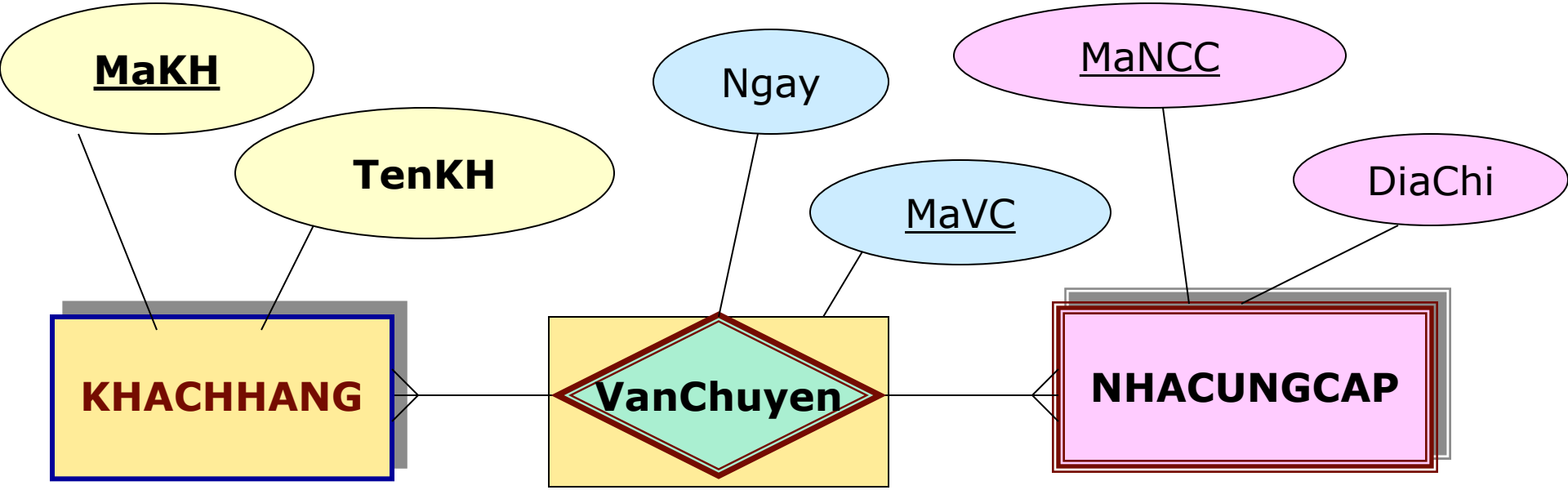
Chuyển đổi từ ERD thành các quan hệ

Bước 4: Chuyển đổi kiểu 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.

Chuyển đổi từ ERD thành các quan hệ

Ví dụ về kiểu thực thể kết hợp

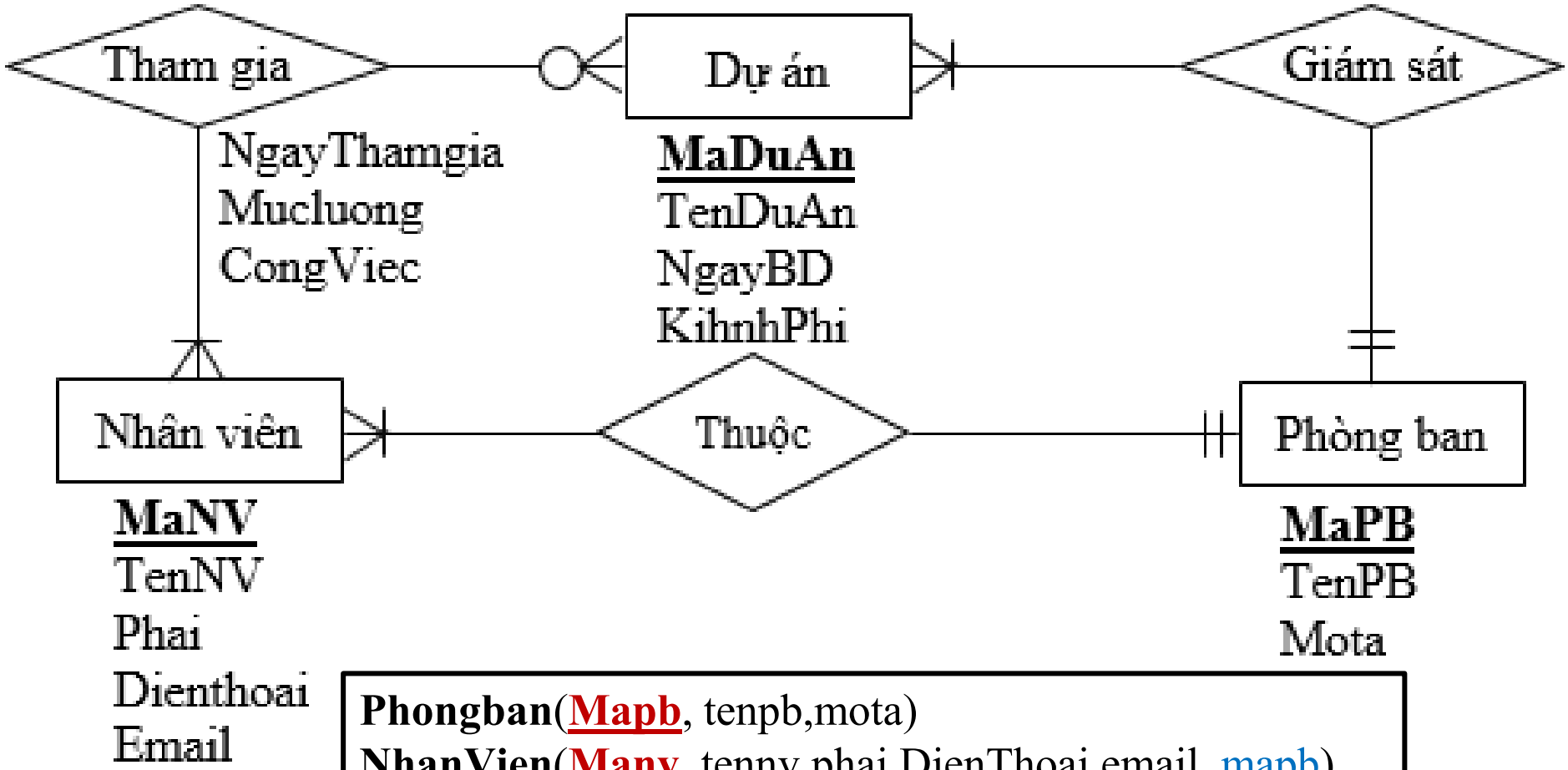


KHACHHANG(MaKH, TenKH)

VANCHUYEN(MaVC, MaKH, MaNCC, Ngay

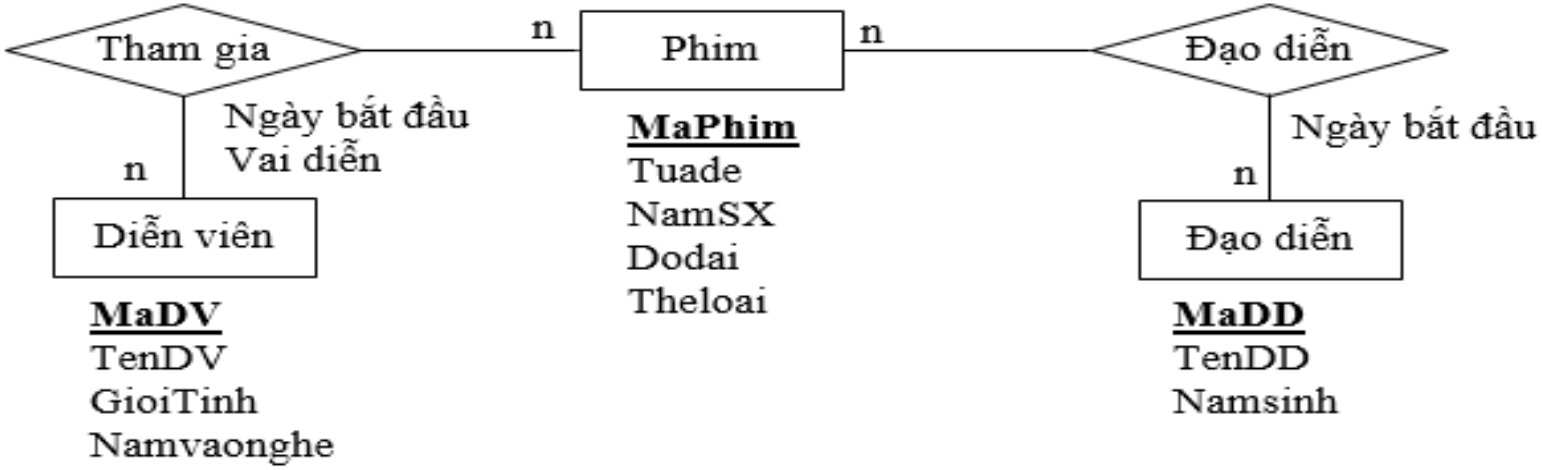
NHACUNGCAP(MaNCC, DiaChi)

Chuyển đổi từ ERD thành các quan hệ



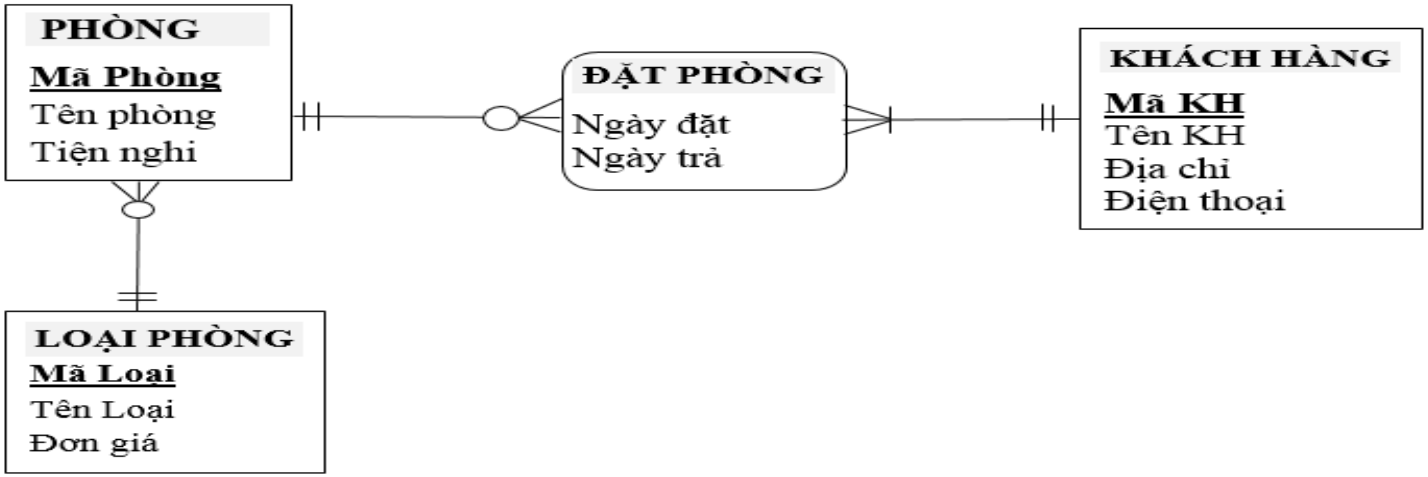
Phongban(**Mapb**, tenpb,mota)
NhanVien(**Manv**, tennv,phai,DienThoai,email, mapb)
Duan(**Mada**, TenDa, NgayBD, KinhPhi,MaPB)
Thamgia(**Manv, Mada**, NgayTG,MucLuong,CongViec)

Chuyển đổi từ ERD thành các quan hệ



DAODIEN(MADD, TENDD, NAMSINH)
PHIM (MAPHIM, TUADE,…,THELOAI)
DAODIENPHIM(MADD, MAPHIM, NGAYBD)
DIENVIEN(MADV, TENDV,…)
THAMGIA(MADV, MAPHIM, NGAYBD, VAIDIEN)

Chuyển đổi từ ERD thành các quan hệ



KhachHang(**MaKH**, tenKH, điachi, điienthoai)
LoaiPhong(**MaLoai**, tennloai, Dongia)
Phong(**MaPhong**, Tenphong, TienNghi, MaLoai)
DatPhong(**MaKH, MaPhong**, NgayDat, NgayTra)

Chuyển đổi từ ERD thành các quan hệ

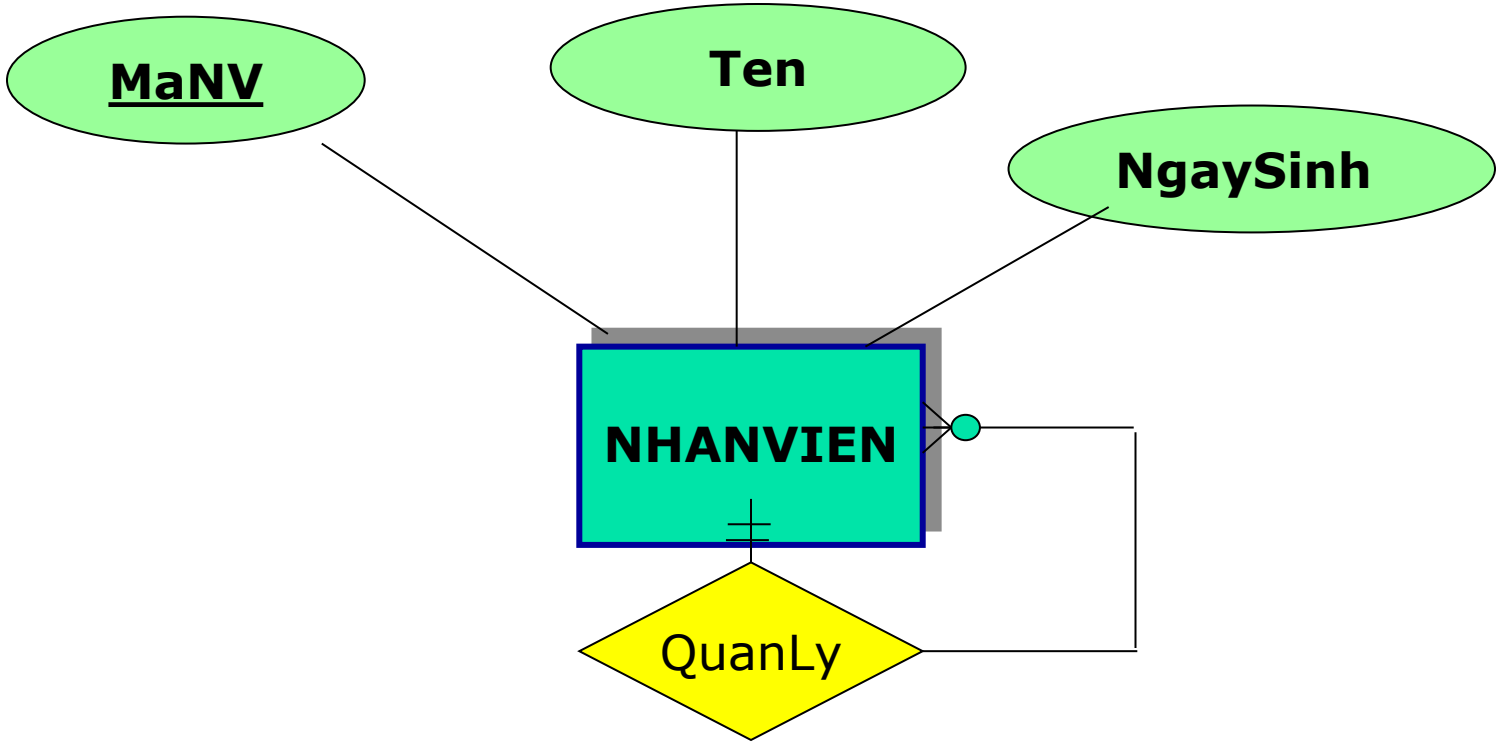
Bước 5: chuyển đổi mối liên kết 1 ngôi

- Với liên kết 1 ngôi 1-M: Thêm 1 khoá ngoại vào quan hệ của mối liên kết này để tham chiếu đến các trị của khoá chính của quan hệ (khoá ngoại và khoá chính phải cùng miền trị).

➔ Được gọi là khoá ngoại đệ quy

Chuyển đổi từ ERD thành các quan hệ

Ví dụ



NHANVIEN(MaNV, Ten, NgaySinh, MaNQL)

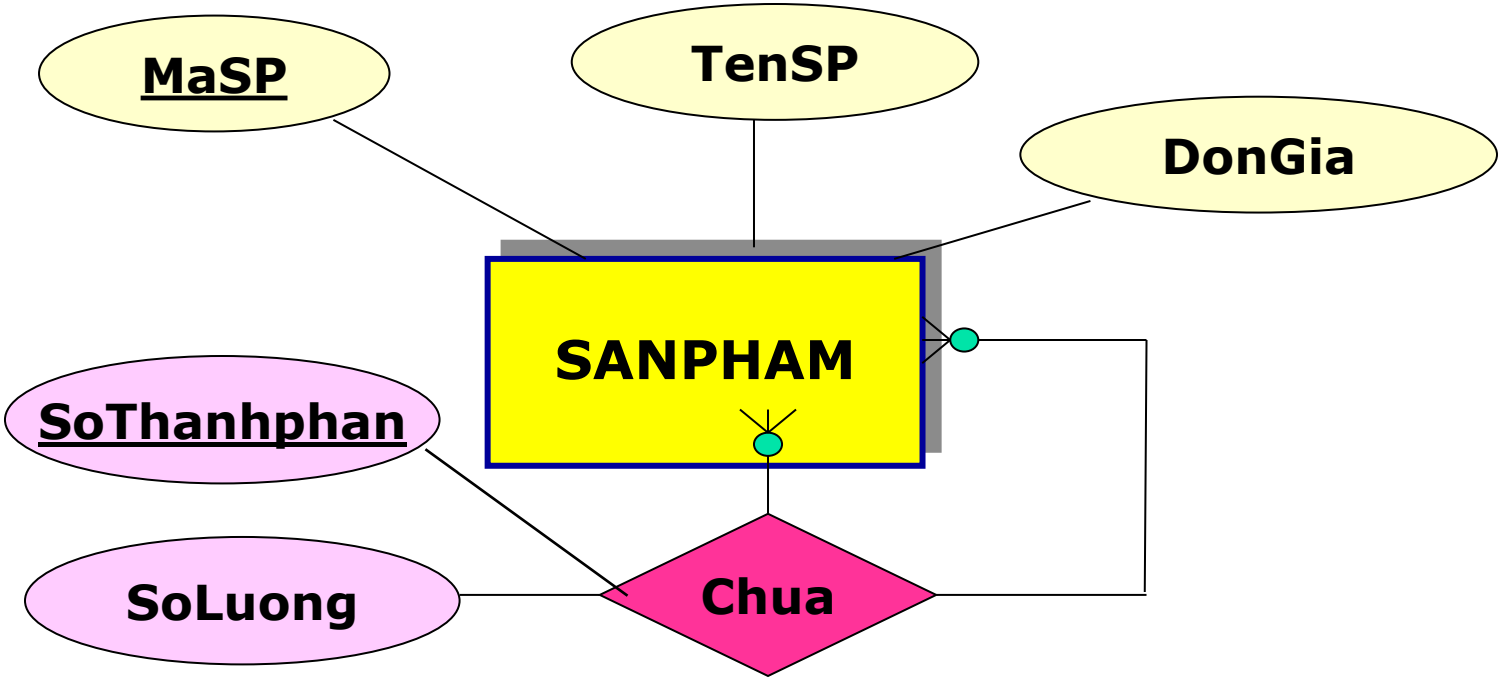
Chuyển đổi từ ERD thành các quan hệ

Bước 5: chuyển đổi mối liên kết 1 ngôi

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

Chuyển đổi từ ERD thành các quan hệ

Ví dụ mối liên kết 1 ngôi M-N



SANPHAM(MaSP, TenSP, DonGia)
CHUA(MaSP, SoThanhPhan, Soluong)

Chuyển đổi từ ERD thành các quan hệ

Bước 6: chuyển đổi mối liên kết 3 ngôi

- Mối 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 mối liên kết
- Bốn trường hợp :
  - 1:1:1
  - 1:1:N
  - 1:N:M
  - N:M:L

Chuyển đổi từ ERD thành các quan hệ

Bước 6: chuyển đổi mối liên kết 3 ngôi

- Trường hợp 1:1:1:

"Mỗi kỹ sư dùng 1 case book khác nhau cho mỗi project mà họ tham gia. Các kỹ sư không dùng chung case book khi làm cùng 1 project"

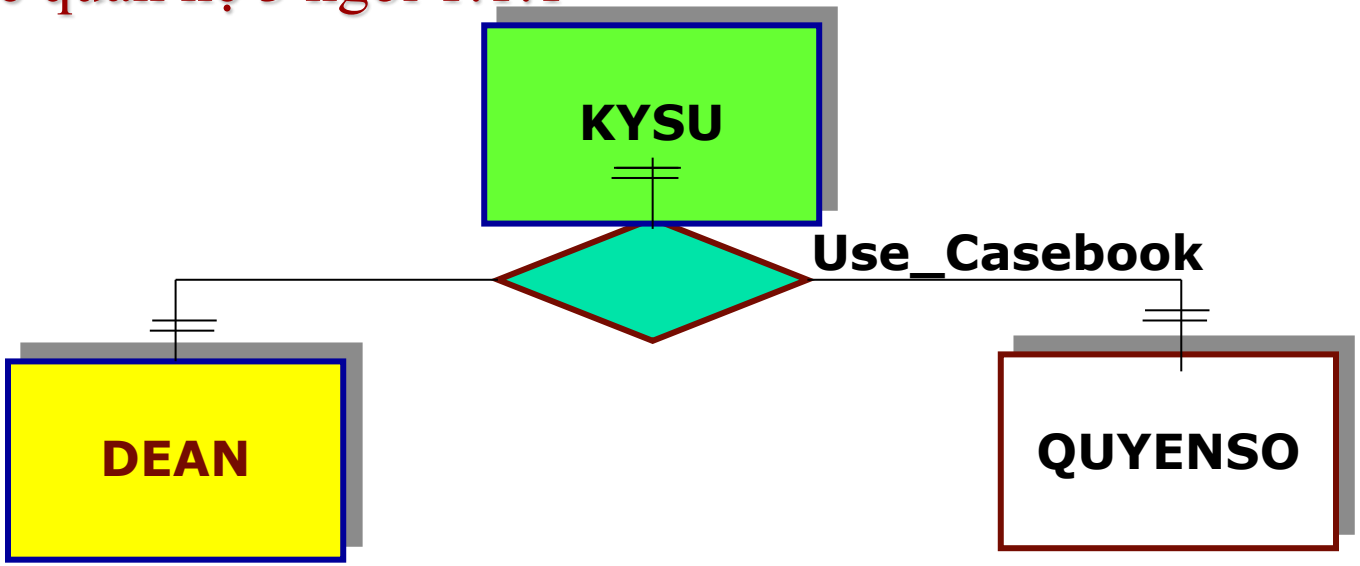
➔ 3 kiểu thực thể: Engineer, Project, CaseBook

➔ Các phụ thuộc hàm:

Eng_No, Proj_No → Book_No
Book_No, Proj_No → Eng_No
Eng_No, Book_No → Proj_No

Chuyển đổi từ ERD thành các quan hệ

Ví dụ về quan hệ 3 ngôi 1:1:1



KYSU(<u>MaKS</u>,…)
DEAN(<u>MaDA</u>,…)
QUYENSO(<u>MaQS</u>,…)
SUDUNG_QUYENSO (<u>MaKS, MaDA</u>, <u>MaQS</u>,…)

68

Chuyển đổi từ ERD thành các quan hệ

Ví dụ về quan hệ 3 ngôi 1:1:1

- Ba khoá candidate:
  - Eng_No, Proj_No
  - Book_No, Proj_No
  - Eng_No, Book_No

➔ Có thể chọn bất kỳ khoá candidate nào làm khoá chính

Chuyển đổi từ ERD thành các quan hệ

Bước 6: chuyển đổi mối liên kết 3 ngôi

- Trường hợp 1:1:N:

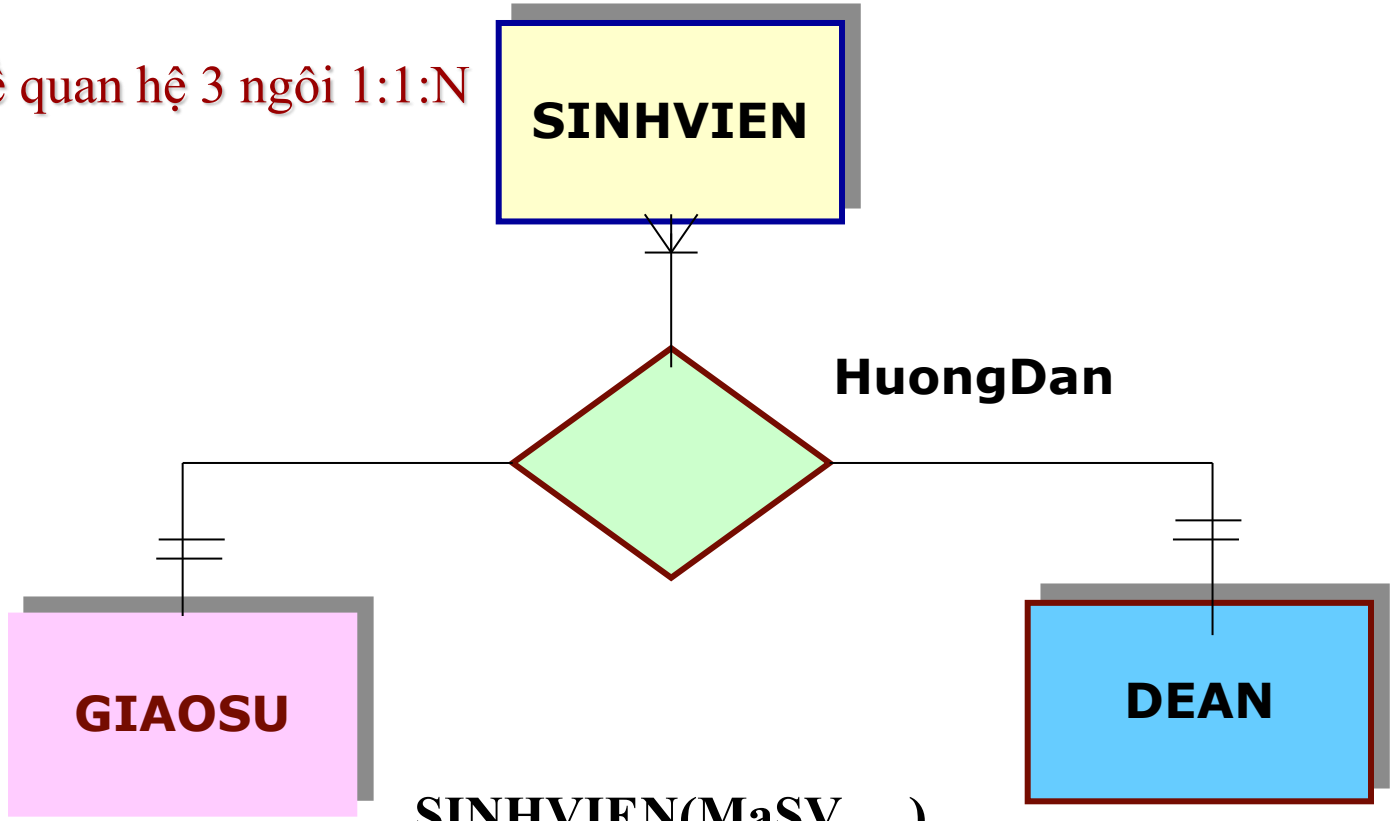
"Sinh viên làm đề tài theo sự hướng dẫn của giáo sư. Không giáo sư nào hướng dẫn cùng 1 sinh viên làm nhiều hơn 1 đề tài. Không sinh viên nào làm 1 đề tài với nhiều hơn 1 giáo sư"

- 3 kiểu thực thể: Teacher, Student, Project
- Các phụ thuộc hàm:
  - Stud_No, Tea_No → Proj_No
  - Stud_No, Proj_No → Tea_No

→ 2 khoá candidate, có thể chọn 1 trong 2 khoá làm khoá chính

Chuyển đổi từ ERD thành các quan hệ

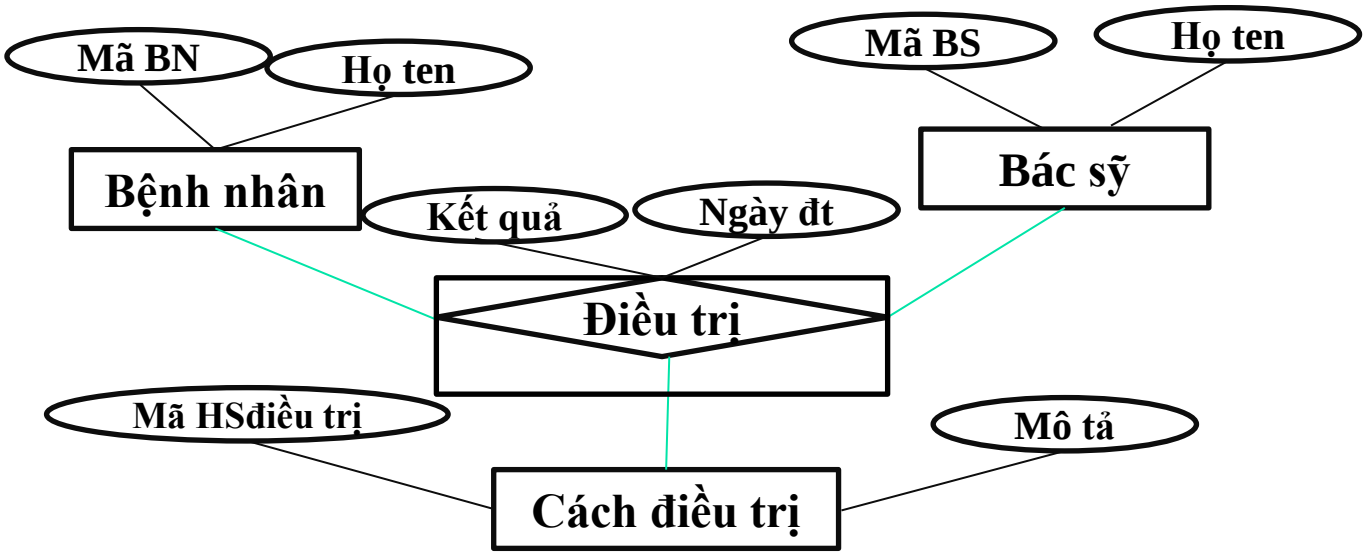
Ví dụ về quan hệ 3 ngôi 1:1:N



SINHVIEN(MaSV,…)
DEAN(MaDA,…)
GIAOSU(TMaGS,…)
HUONGDAN(MaGS, MaSV, MaDA,…)

Chuyển mối quan hệ N-ary

- Ví dụ: Mối quan hệ Điều trị là quan hệ N-ary



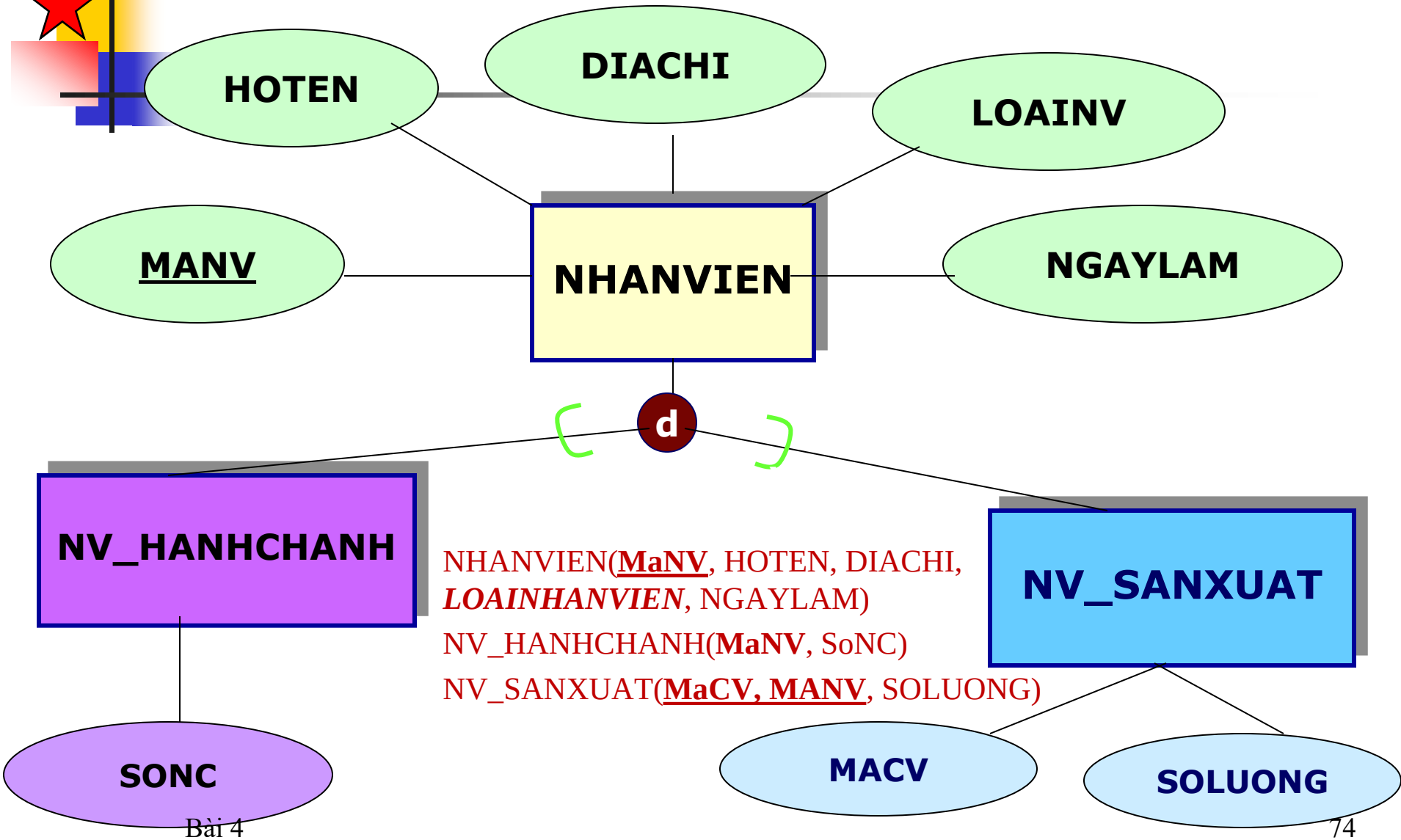
BenhNhan (**MaBN,** Hoten)
Bacsy (**MaBS**, Hoten)
Cachdieutri (**MaDtri,** Mota)
BS_Dtri_BN (**MaBN, MaBS, MaDtri**, Ngay, Ketqua)

Chuyển đổi từ ERD thành các quan hệ

Bước 7: chuyển đổi mối liên kết siêu kiểu/ kiểu con

- Trình tự chuyển đổi: **Tùy chọn 8A**
  - Tạo 1 quan hệ cho siêu kiểu và 1 quan hệ cho mỗi kiểu con
  - Đưa vào quan hệ cho siêu kiểu những thuộc tính chung
  - Đưa vào mỗi quan hệ cho mỗi kiểu con khoá chính của siêu kiểu cùng với thuộc tính riêng của từng kiểu con
  - Thêm thuộc tính phân biệt kiểu con vào siêu kiểu

Chuyển đổi từ ERD thành các quan hệ



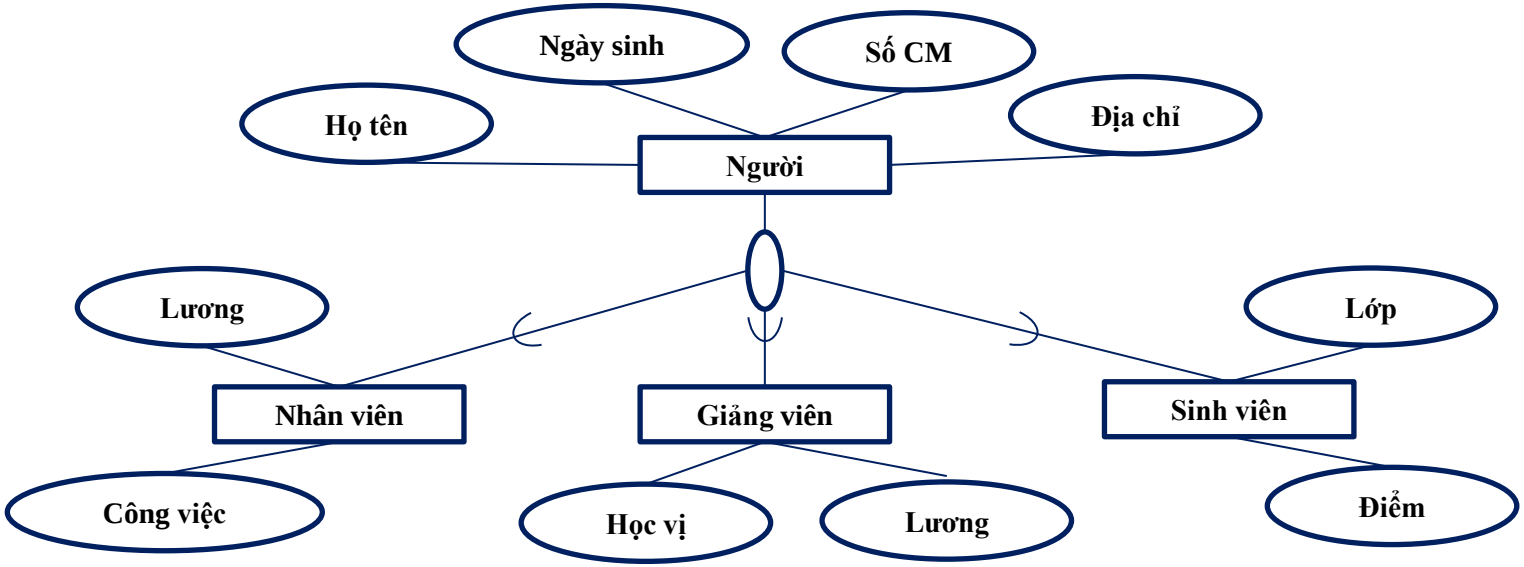
NHANVIEN(**MaNV**, HOTEN, DIACHI, ***LOAINHANVIEN***, NGAYLAM)

NV_HANHCHANH(**MaNV**, SoNC)

NV_SANXUAT(**MaCV, MANV**, SOLUONG)

Chuyển mối quan hệ lớp cha-con

- Ví dụ: mô hình ER của thực thể Người



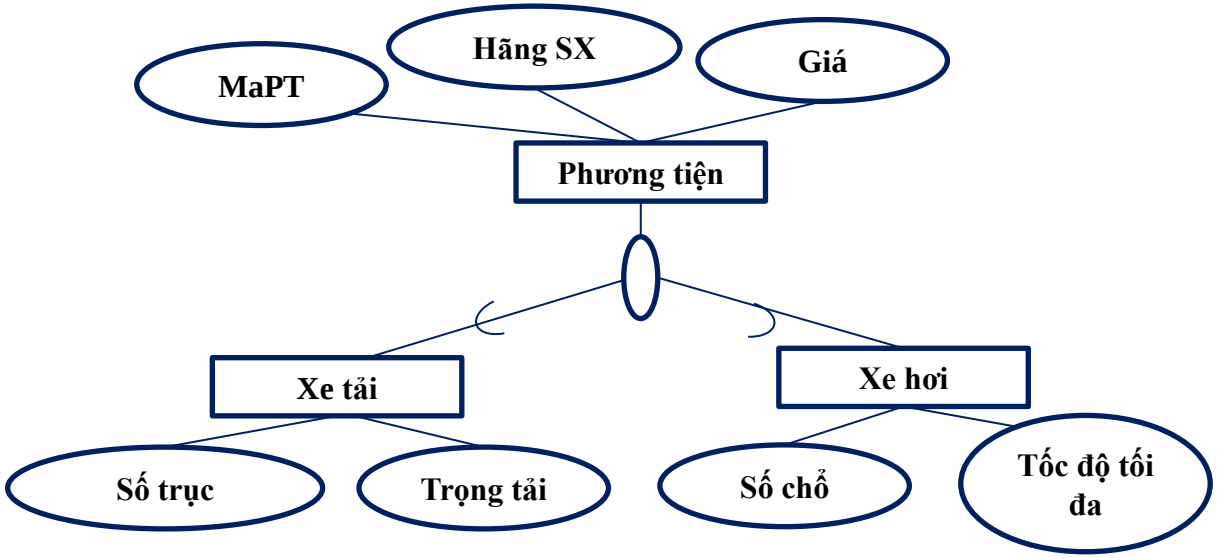
Nguoi (**SoCM**, Hoten, Ngaysinh, diachi,Loai)
Nhanvien(**SoCM**, congviec, luong)
Giangvien(**SoCM**, Hocvi, luong)
Sinhvien(**SoCM**, Lop, diem)

Chuyển mối quan hệ lớp cha-con

- **Tùy chọn 8B:** gộp nhiều quan hệ lớp con thành một quan hệ.
  - Tạo một quan hệ Li cho mỗi subclass Si, với 1<i<m
    - Thuộc tính của Li: $Attr(L_i)$={thuộc tính của S_i} U {k,a_1…,a_n}.
    - Khóa của Li: $PK(L_i)$ = k.
  - Tùy chọn này chỉ thích hợp với chuyên biệt hóa mà lớp con là **total** (mọi thực thể trong lớp cha phải thuộc ít nhất một trong các lớp con).

Chuyển mối quan hệ lớp cha-con

- Ví dụ: quan hệ của thực thể **Phương tiện**



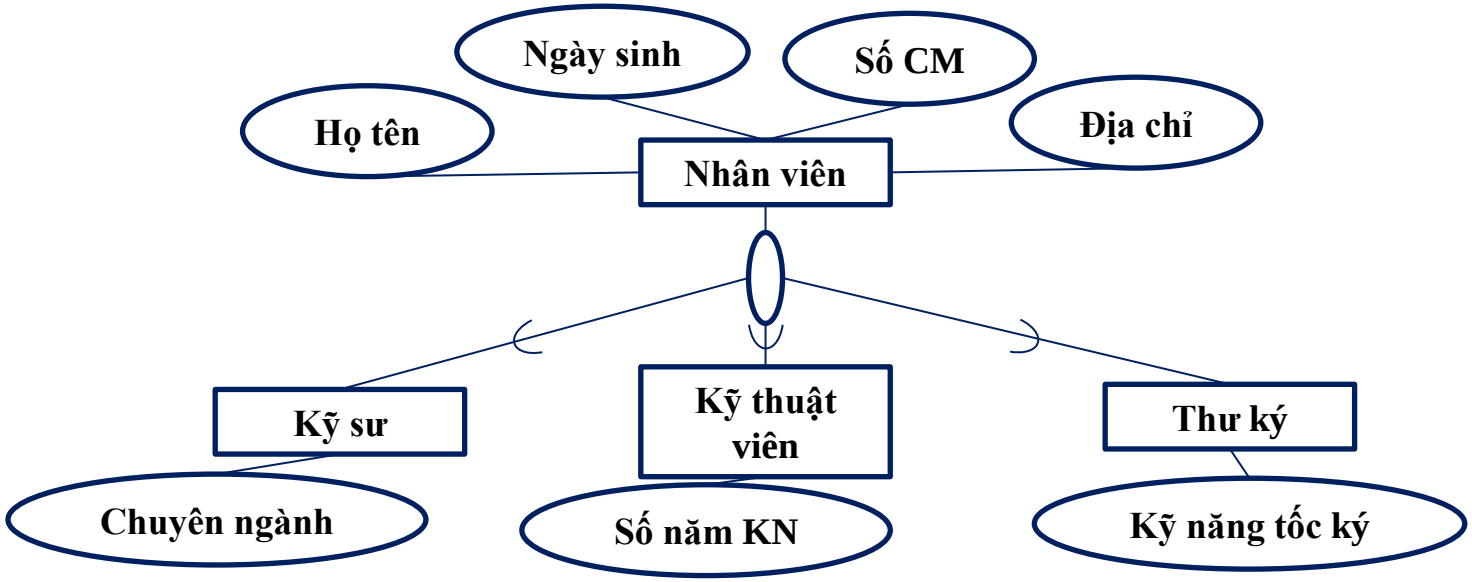
Xetai (**MaPT**, Hangsx, Gia, Trongtai, Sotruc)
Xehoi (**MaPT**, Hangsx, Gia, Socho, Tocdotoida)

Chuyển mối quan hệ lớp cha-con

- **Tùy chọn 8C:** Một quan hệ đơn với một thuộc tính phân phân biệt.
  - Tạo một quan hệ đơn L với:
    - Thuộc tính của L: Attrs(L) = $\{k,a_1,\ldots a_n\} \cup$ {thuộc tính của S_1} $\cup \ldots \cup$ {thuộc tính của S_m} $\cup$ {t}
    - Khóa của L: PK(L) = k.
  - Thuộc tính t gọi là thuộc tính phân biệt, dùng để xác định một bộ trong quan hệ thuộc lớp con nào

Chuyển mối quan hệ lớp cha-con

- Ví dụ: mô hình ER của thực thể **Nhanvien**



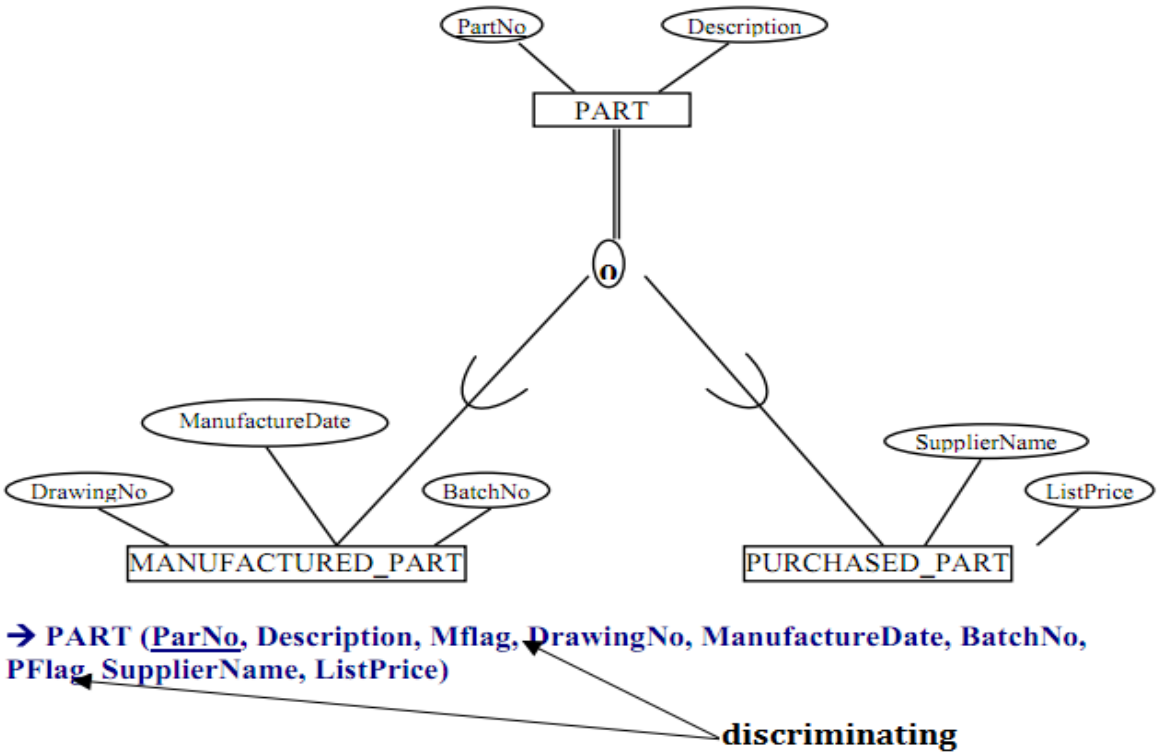
Nhanvien (**SoCM**, Hoten, Ngaysinh, Diachi, Chuyennganh, SonamKN, Ky nangtocky, **LoaiCV**)

Chuyển mối quan hệ lớp cha-con

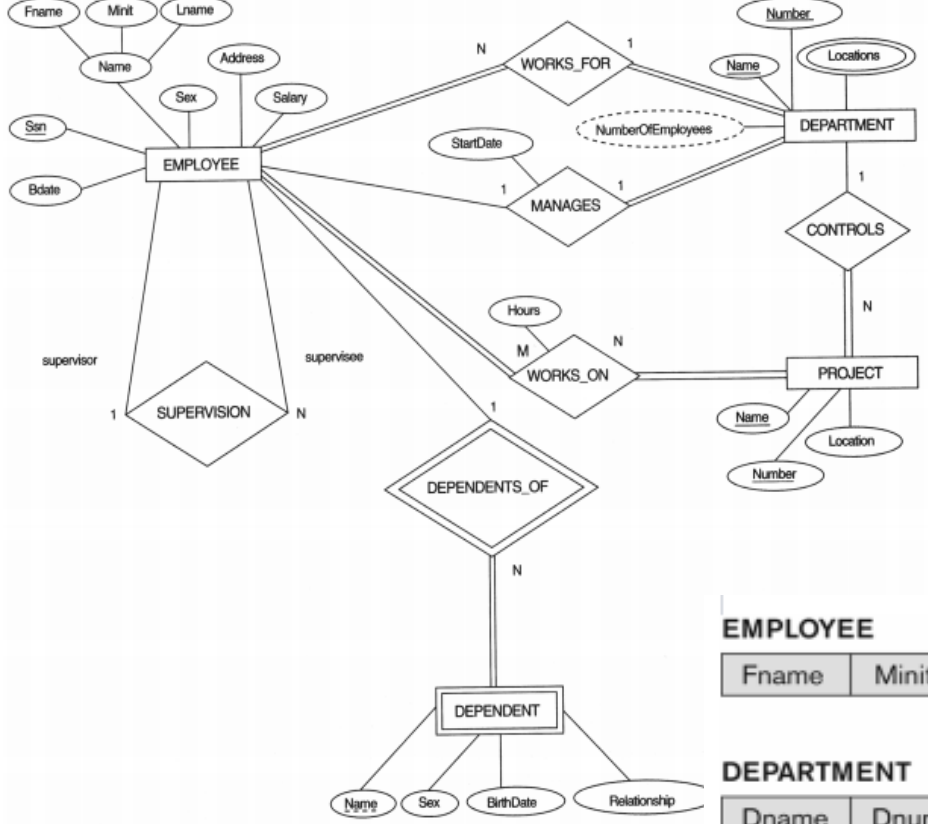
- **Tùy chọn 8D:** một quan hệ đơn với nhiều loại thuộc tính.
  - Tạo một lược đồ quan hệ đơn L với:
    - Thuộc tính của L: Attrs(L) = $\{k,a_1,\ldots a_n\} \cup$ {thuộc tính của S_1} $\cup\ldots\cup$ {thuộc tính của S_m} U $\{t_1, t_2,\ldots,t_m\}$
    - Khóa của L: PK(L) = k.
  - Mỗi t_i, $1<i<m$, là một thuộc tính kiểu Boolean để xác định một bộ thuộc lớp con Si nào.

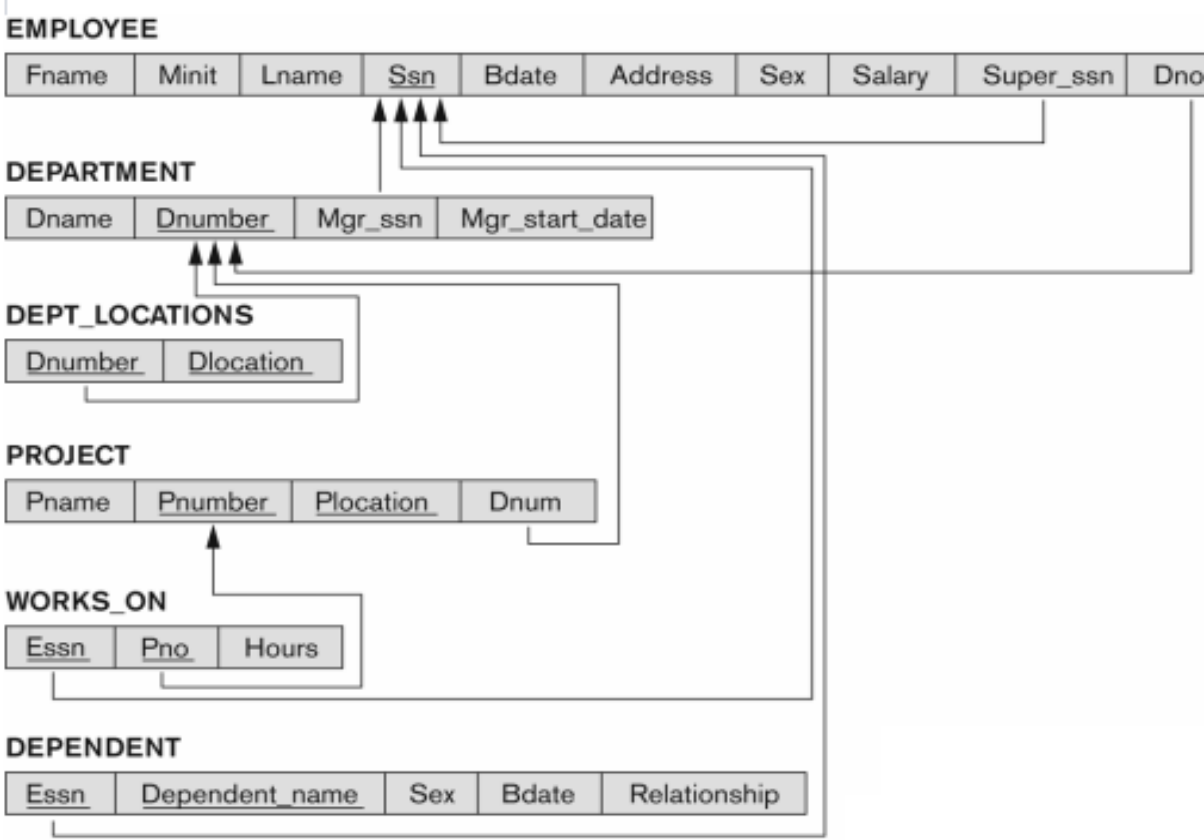
Chuyển mối quan hệ lớp cha-con

- Ví dụ:



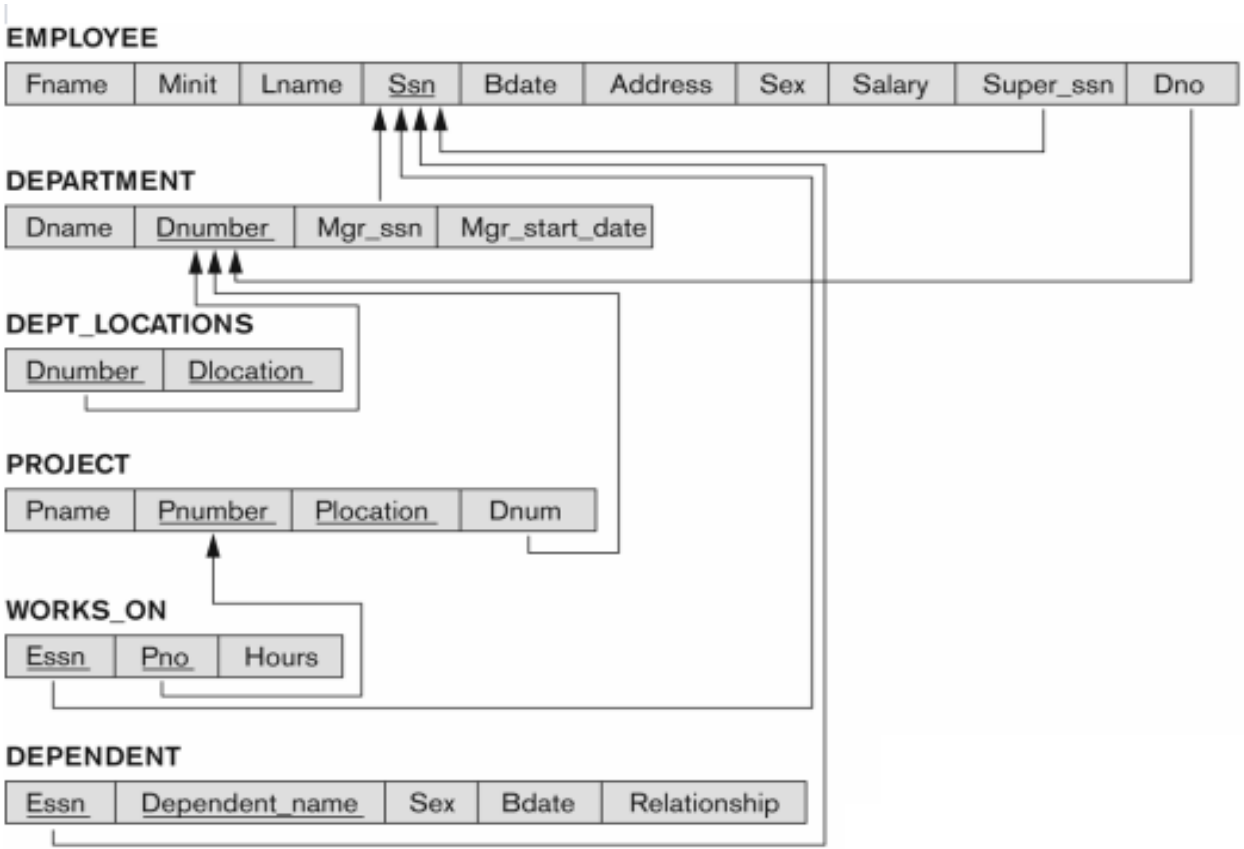
ER của CSDL


Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
WORKS_FOR
N
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
MANAGES
1
1
CONTROLS
1
N
Hours
M
N
WORKS_ON
PROJECT
Name
Location
Number
supervisor
supervisee
SUPERVISION
1
N
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship


EMPLOYEE
Fname
Minit
Lname
Ssn
Bdate
Address
Sex
Salary
Super_ssn
Dno
DEPARTMENT
Dname
Dnumber
Mgr_ssn
Mgr_start_date
DEPT_LOCATIONS
Dnumber
Dlocation
PROJECT
Pname
Pnumber
Plocation
Dnum
WORKS_ON
Essn
Pno
Hours
DEPENDENT
Essn
Dependent_name
Sex
Bdate
Relationship

Ví dụ: Mô hình ER của CSDL Company

- Kết quả chuyển từ mô hình ER của COMPANY sang mô hình quan hệ.



Bài tập

- Chuyển các sơ đồ ERD trong bài 3 sang lược đồ quan hệ

Bài tập

- Dựa vào mẫu hoá đơn bán hàng hãy thiết kế mô hình dữ liệu quan hệ

PHIẾU MƯỢN SÁCH

Số phiếu: Ngày mượn: .....

Tên sinh viên: .....

Mã sinh viên: .....

Mã lớp: .....

STT	**Mã sách**	Tên sách	Nhà xuất bản	Ghi chú
1				
2				
3				

Bài tập

- Dựa vào mẫu hoá đơn bán hàng hãy thiết kế mô hình dữ liệu quan hệ
- **Trong đó: Số hoá đơn** xác định được **ngày tạo lập; Mã khách hàng** xác định được **tên khách hàng, địa chỉ; Mã hàng** xác định được **tên hàng hoá, đơn vị tính, đơn giá** và **số lượng**

HOÁ ĐƠN

Số hoá đơn: 120131 Ngày tạo lập: 01/01/2009

Tên khách hàng: Nguyễn Văn B

Địa chỉ: 222 đường Hùng Vương, Q.5, TP.HCM Mã khách hàng: 011019977

Tổng tiền: 19 000 000 VNĐ

STT	**Mã hàng**	Tên hàng hoá	Đơn vị tính	Đơn giá (VNĐ)	Số lượng
1	H1010	Laptop HP DV4 T504	bộ	15000000	1
2	S1040	Máy in Canon LX319	bộ	2000000	2

Bài tập

- Thiết kế mô hình dữ liệu quan hệ dựa vào mẫu phiếu mượn sách trong thư viện.
- **Trong đó: Số phiếu** xác định được **ngày mượn; Mã sinh viên** xác định được **tên sinh viên, mã lớp; Mã sách** xác định được **tên sách, nhà xuất bản, ghi chú**

PHIẾU MƯỢN SÁCH

Số phiếu: Ngày mượn: .....

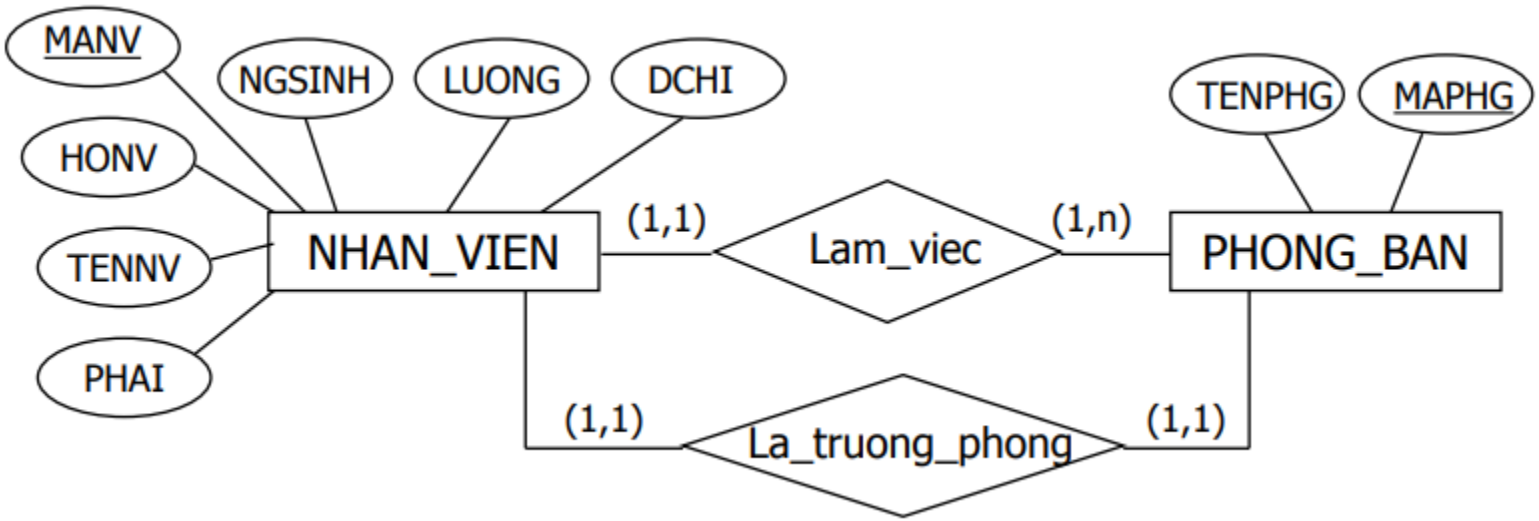
Tên sinh viên: .....

Mã sinh viên: .....

Mã lớp: .....

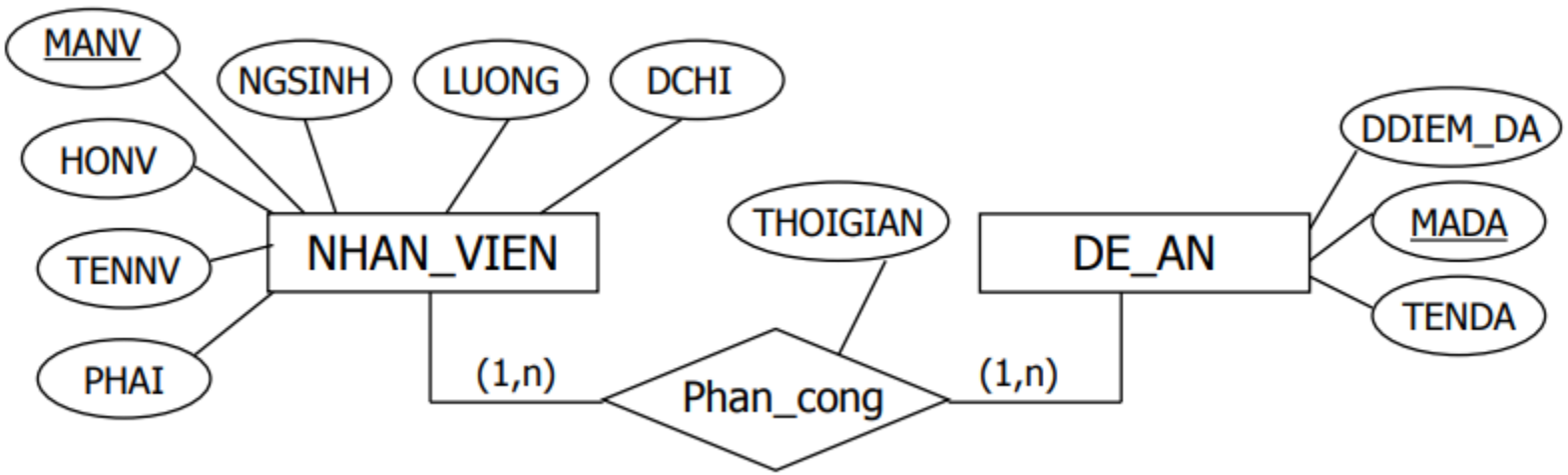
STT	**Mã sách**	Tên sách	Nhà xuất bản	Ghi chú
1				
2				
3				

Bài tập chuyển mô hình ERD sang mô hình quan hệ



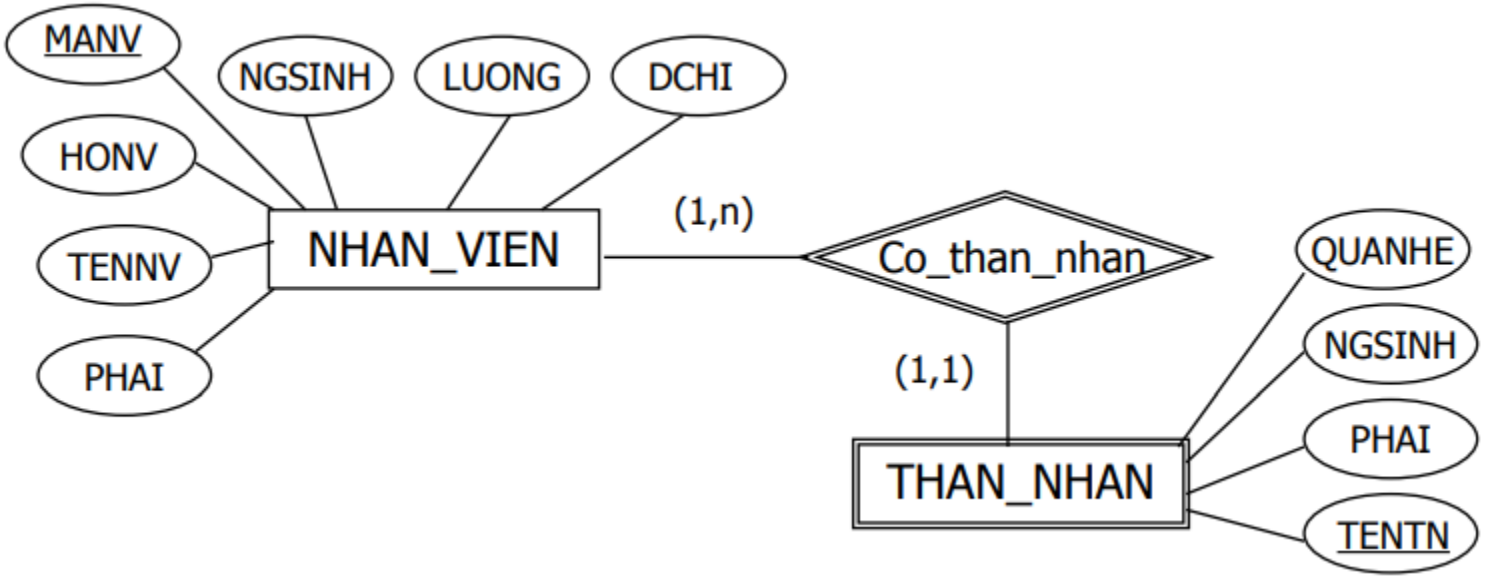
- NHAN_VIEN(**MANV**, TENNV, HONV, NGSINH, DCHI, PHAI, LUONG, **MAPHG**)
- PHONG_BAN(TENPHG, **MAPHG, MANV_TP**)

Bài tập chuyển mô hình ERD sang mô hình quan hệ



- NHAN_VIEN(**MANV**, TENNV, HONV, NGSINH, DCHI, PHAI, LUONG)
- DE_AN(TENDA, **MADA,** DDIEM_DA)
- PHAN_CONG(MANV, MADA, THOIGIAN)

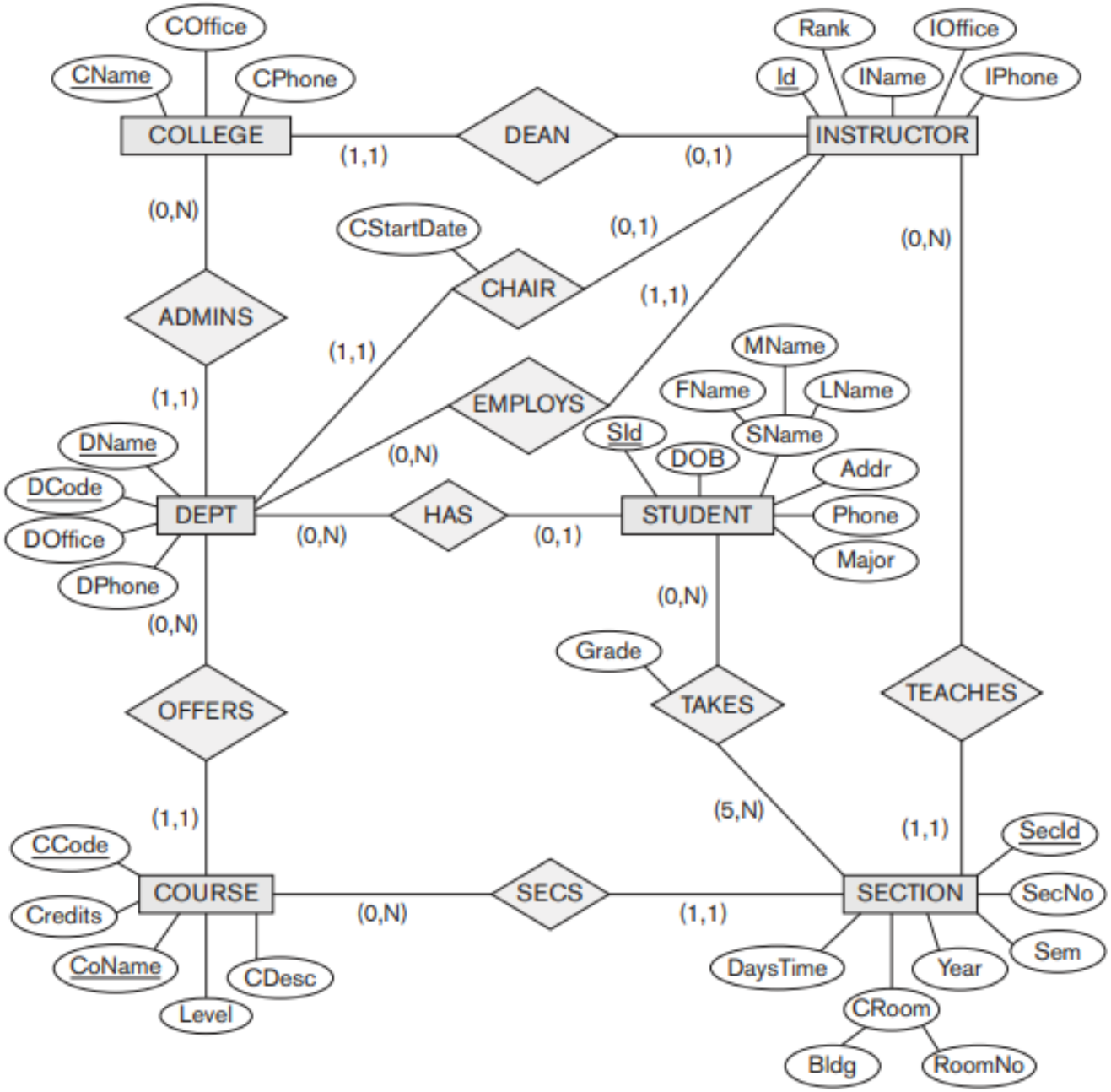
Bài tập chuyển mô hình ERD sang mô hình quan hệ



- NHAN_VIEN(**<u>MANV</u>**, TENNV, HONV, NGSINH, DCHI, PHAI, LUONG)
- THAN_NHAN(**<u>MANV, TENTN</u>**, PHAI, NGSINH, QUANHE)

- Hệ thống được sử dụng để quản lý việc mượn sách trong một thư viện. Các tài liệu cho độc giả mượn có các thuộc tính là mã tài liệu (khóa), tên tài liệu (tựa đề). Tài liệu gồm 2 loại: sách và báo_tạp chí.
- Mỗi tựa đề sách cần được biết do tác giả nào viết. Thông tin về tác giả gồm mã tác giả (khóa), tên tác giả, năm sinh. Một tác giả viết nhiều sách, một sách có thể đồng tác giả.
- Mỗi tựa đề sách có nhiều lần xuất bản (tái bản). Thông tin về một lần xuất bản gồm có: lần xuất bản, năm xuất bản, khổ giấy, số trang, nhà xuất bản, giá, có hoặc không kèm đĩa CD. Lần xuất bản được đánh số 1, 2, 3, ... cho mỗi tựa đề sách, do đó có sự trùng nhau giữa các tựa đề sách khác nhau.
- Mỗi lần xuất bản một tựa đề sách, Thư viện nhập vào nhiều cuốn sách. Mỗi cuốn sách này được quản lý riêng dựa vào số thứ tự được đánh số từ 1, 2, 3, ...trong số các cuốn sách cùng tựa đề và cùng một lần xuất bản. Khi cho độc giả mượn, thông tin ghi trong thẻ độc giả phải xác định chính xác cuốn nào. Thông tin về mỗi cuốn sách này còn có thêm tình trạng để lưu tình trạng hiện tại của sách (tốt, rách, mất trang,...).

- Khác với việc cho mượn sách, việc cho mượn báo_tạp chí không cần chỉ chính xác tờ nào trong số các tờ cùng tựa đề và cùng một lần xuất bản. Tuy nhiên trong số này (cùng tựa đề và cùng một lần xuất bản), độc giả mỗi lần chỉ có thể mượn 1 tờ.
- Mỗi tựa đề báo_tạp chí cần các thông tin: năm bắt đầu phát hành, định kỳ (hàng ngày, hàng tuần hay hàng tháng), nhà xuất bản; đối với mỗi kỳ xuất bản cần biết số lượng tờ thư viện nhập về, số lượng tờ còn lại trong thư viện hiện tại (thuộc tính này được tính từ số tờ thư viện nhập về trừ đi số tờ đang có độc giả mượn).
- Thông tin về độc giả gồm số thẻ độc giả (khóa), ngày cấp thẻ, tên, nghề nghiệp, phái. Mỗi lần độc giả có thể mượn nhiều sách cũng như báo_tạp chí, thông tin cần lưu là ngày mượn và ngày trả cho từng tài liệu mượn.


COffice
CName
CPhone
COLLEGE
DEAN
(1,1)
(0,1)
Rank
IOffice
Id
IName
IPhone
INSTRUCTOR
(0,N)
CStartDate
(0,1)
(0,N)
CHAIR
(1,1)
ADMINS
(1,1)
MName
(1,1)
EMPLOYS
FName
LName
SId
SName
DOB
DName
(0,N)
Addr
DCode
DEPT
HAS
STUDENT
Phone
DOffice
(0,N)
(0,1)
Major
DPhone
(0,N)
(0,N)
Grade
OFFERS
TAKES
TEACHES
(1,1)
(5,N)
(1,1)
SecId
CCode
COURSE
SECS
SECTION
SecNo
Credits
(0,N)
(1,1)
Sem
CoName
CDesc
DaysTime
Year
Level
CRoom
Bldg
RoomNo

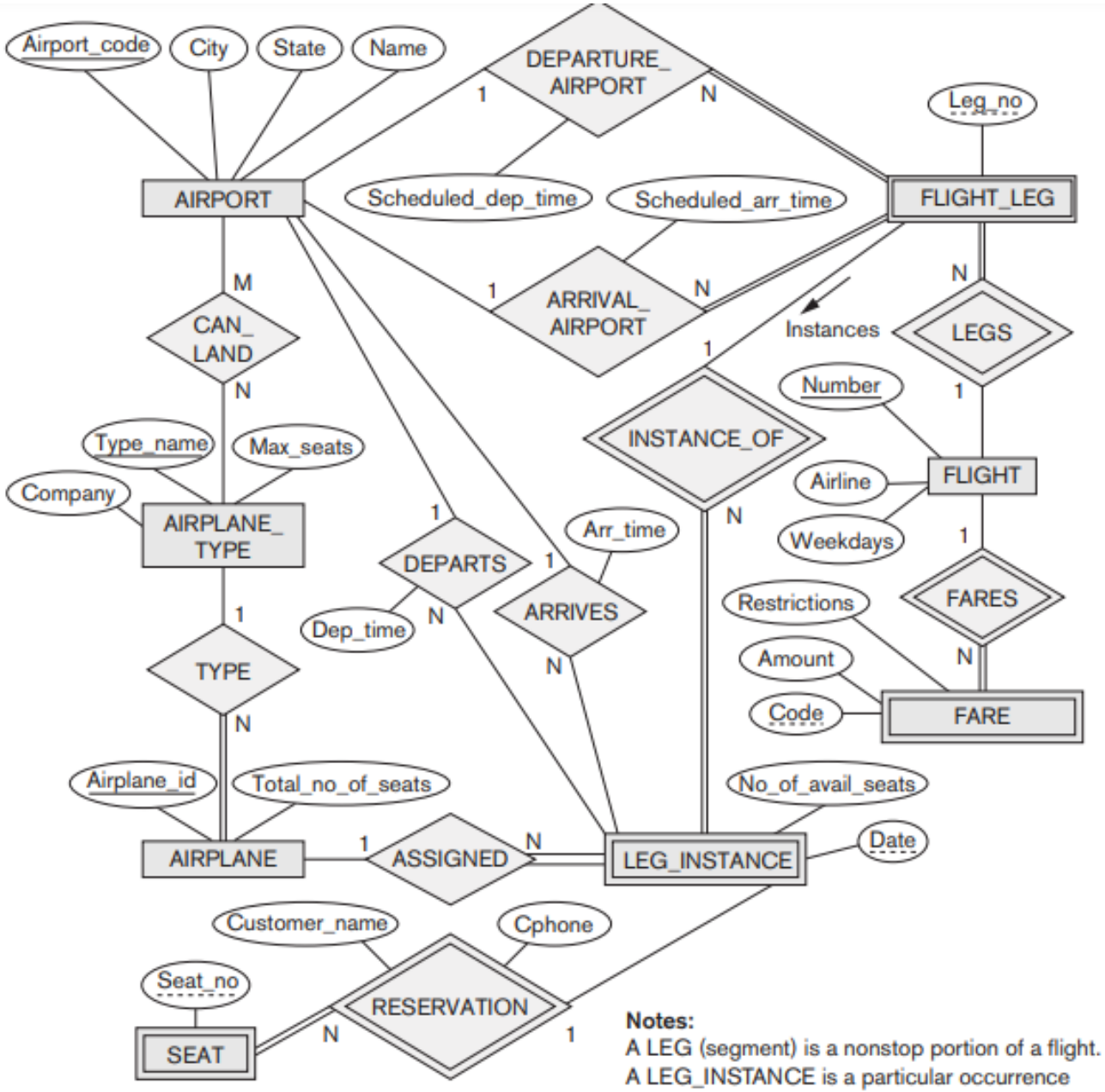
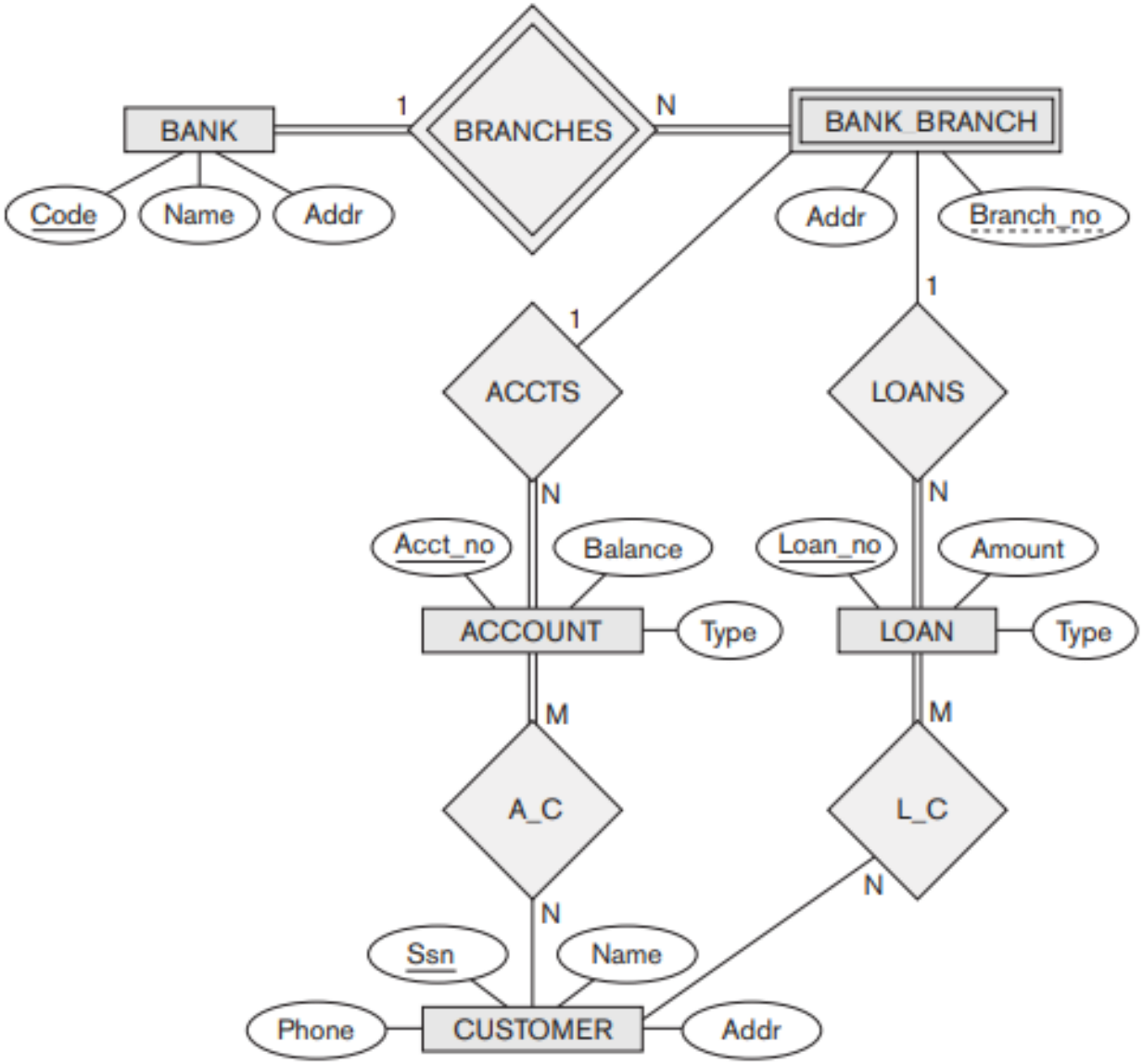

Airport_code
City
State
Name
DEPARTURE_ AIRPORT
1
N
Leg_no
AIRPORT
Scheduled_dep_time
Scheduled_arr_time
FLIGHT_LEG
M
1
ARRIVAL_ AIRPORT
N
N
CAN_ LAND
Instances
LEGS
N
1
Number
INSTANCE_OF
1
Type_name
Max_seats
Company
AIRPLANE_ TYPE
Airline
FLIGHT
N
Weekdays
1
1
Arr_time
DEPARTS
1
Restrictions
FARES
1
Dep_time
N
ARRIVES
Amount
N
TYPE
N
Code
FARE
N
Airplane_id
Total_no_of_seats
No_of_avail_seats
AIRPLANE
1
ASSIGNED
N
LEG_INSTANCE
Date
Customer_name
Cphone
Seat_no
RESERVATION
SEAT
N
1
Notes:
A LEG (segment) is a nonstop portion of a flight.
A LEG_INSTANCE is a particular occurrence of a LEG on a particular date.

Figure 3.21

Figure 3.22
An ER diagram for a BANK database schema.



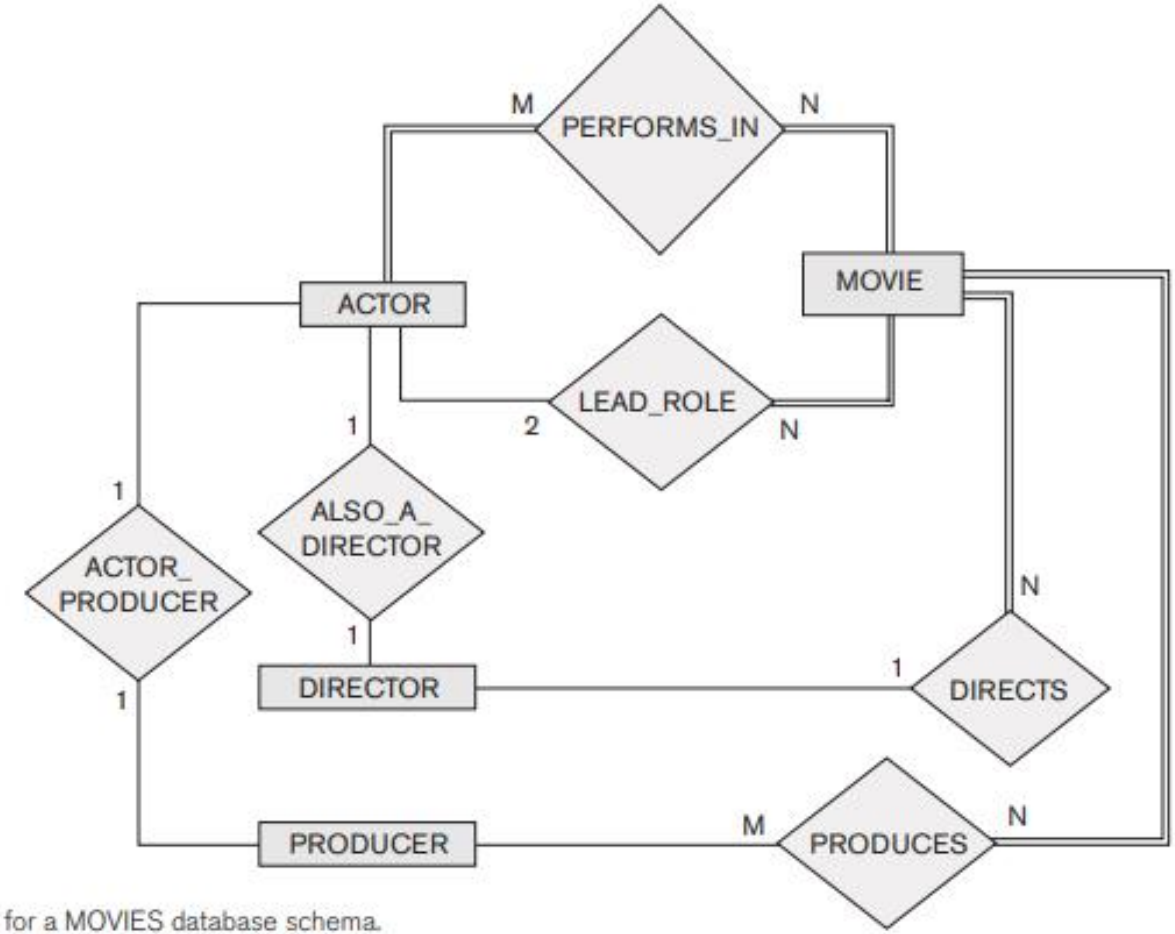


Figure 3.25
An ER diagram for a MOVIES database schema.

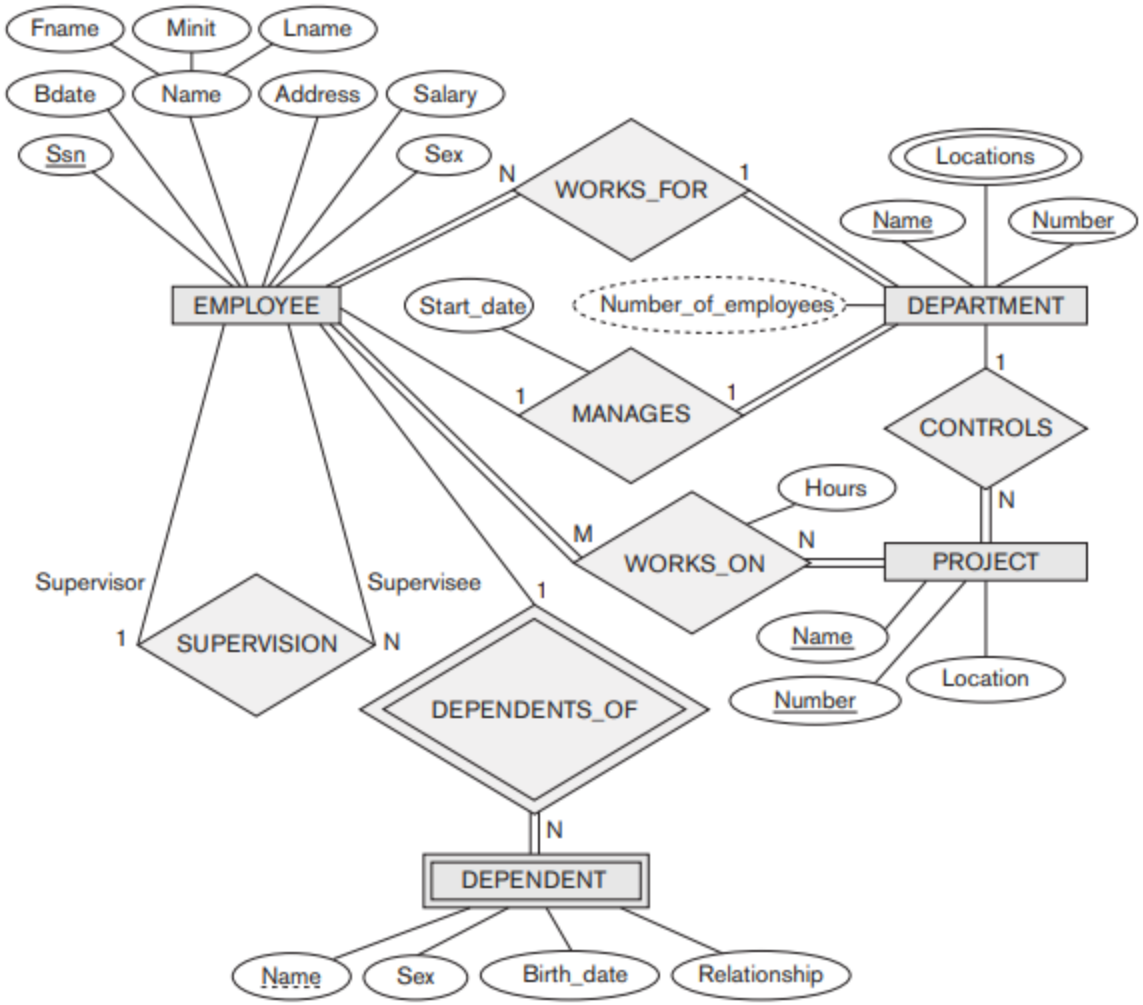


Figure 3.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 3.14.

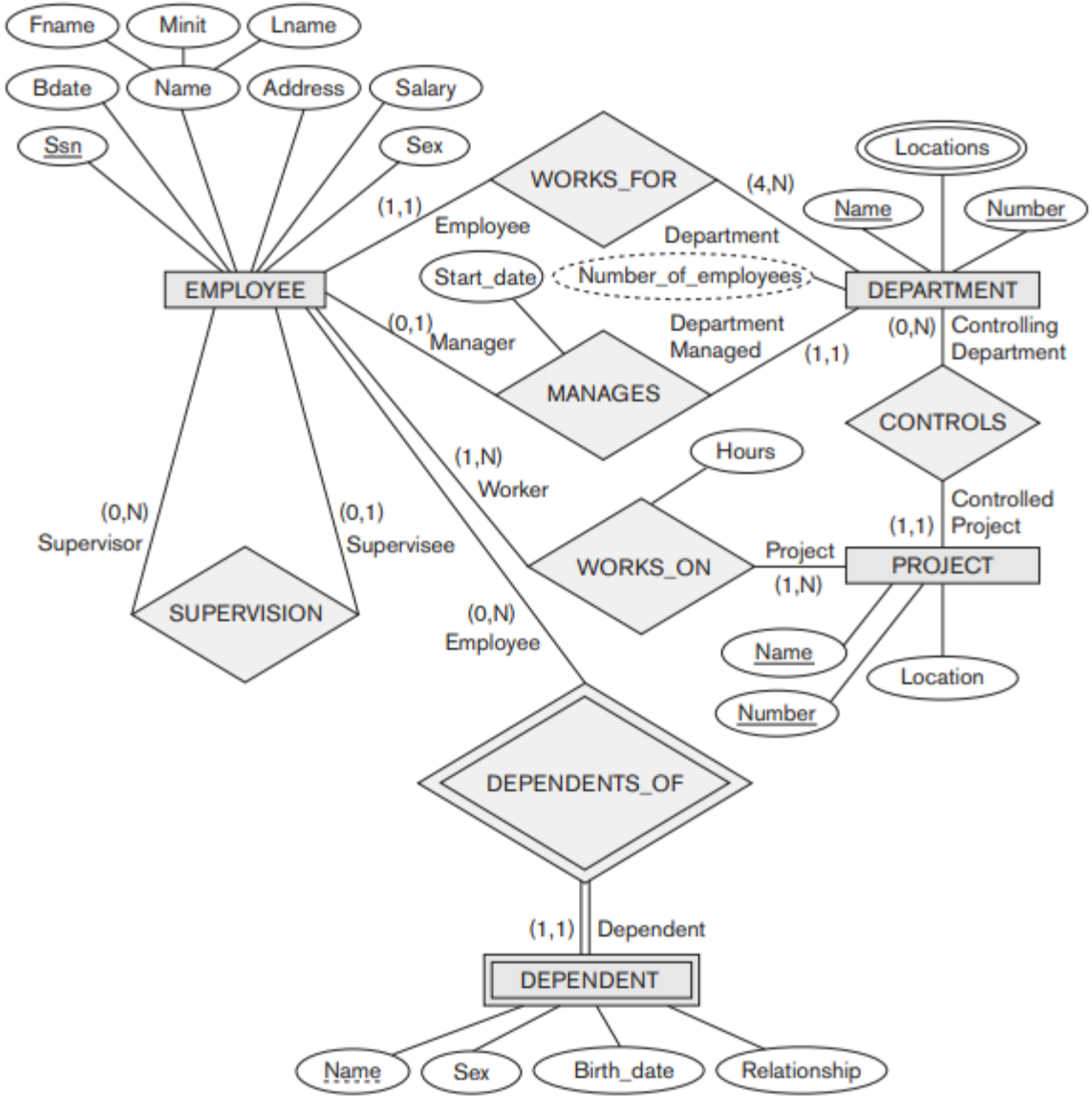


Figure 3.15
ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.

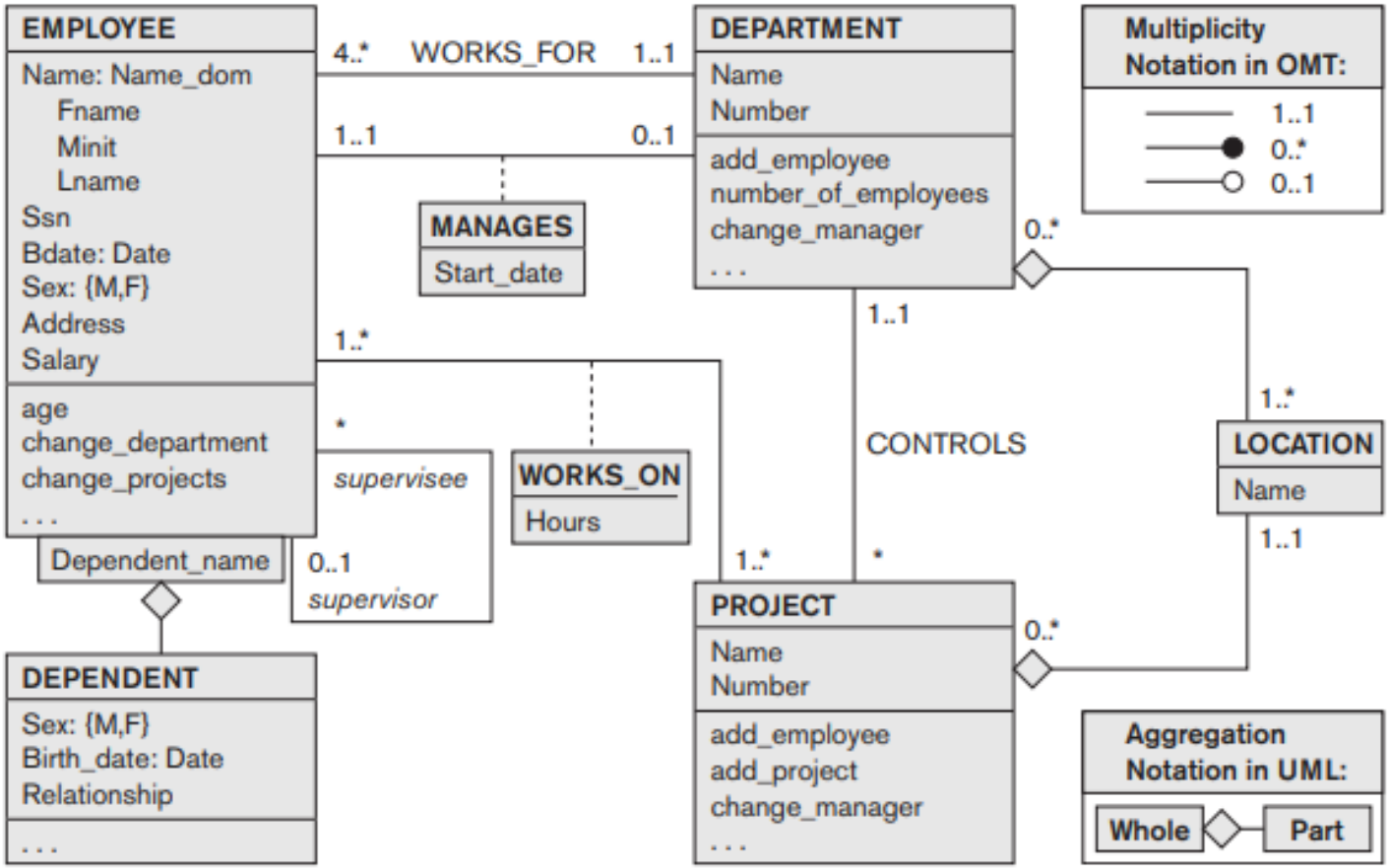
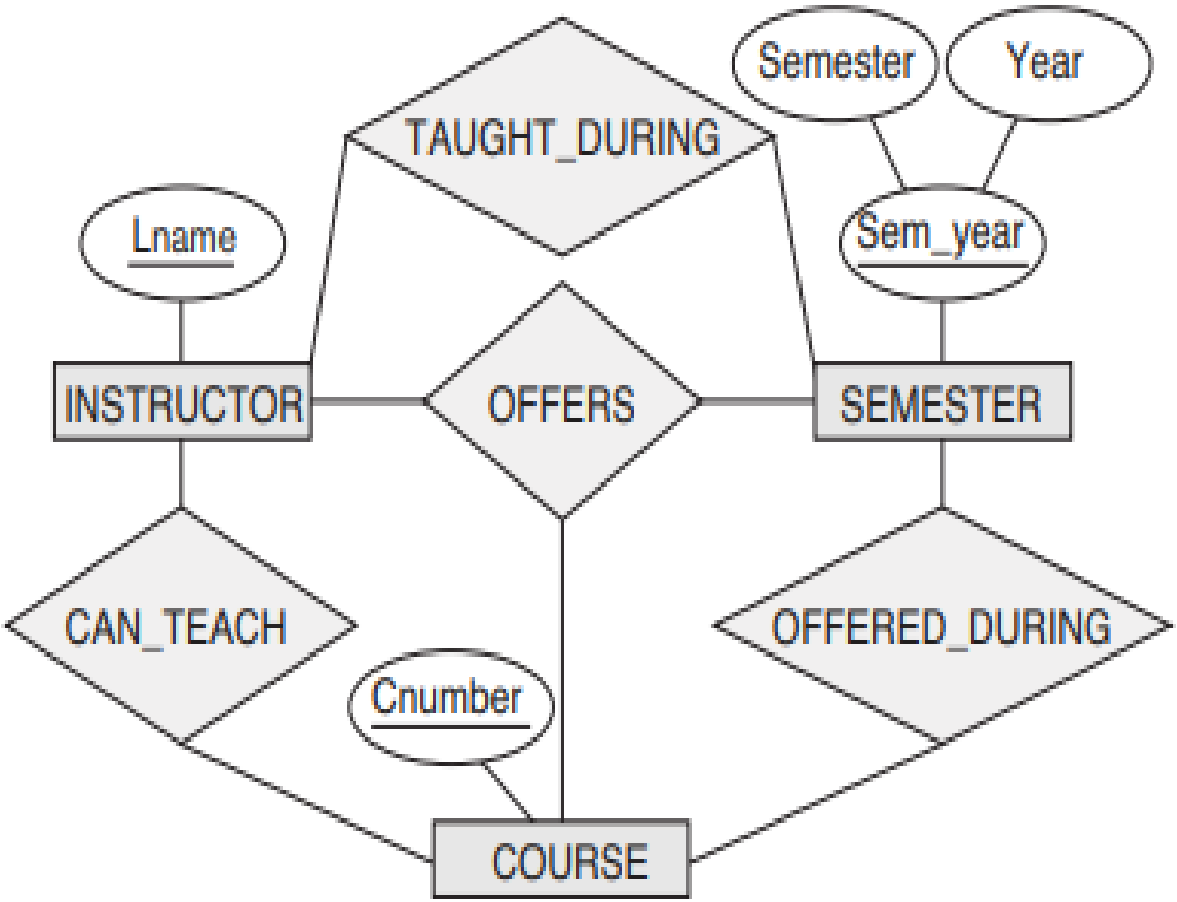


Figure 3.16
The COMPANY conceptual schema in UML class diagram notation.

Figure 3.18
Another example of ternary versus binary relationship types.



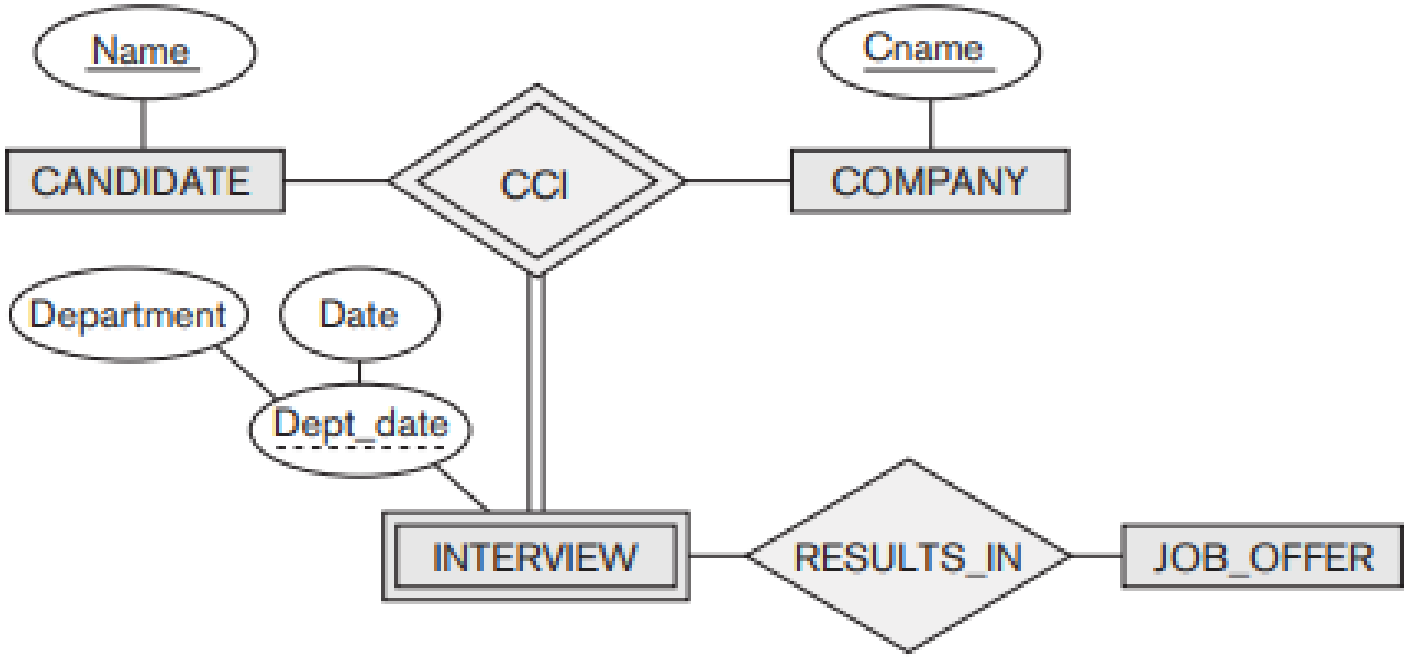


Figure 3.19
A weak entity type INTERVIEW with a ternary identifying relationship type.

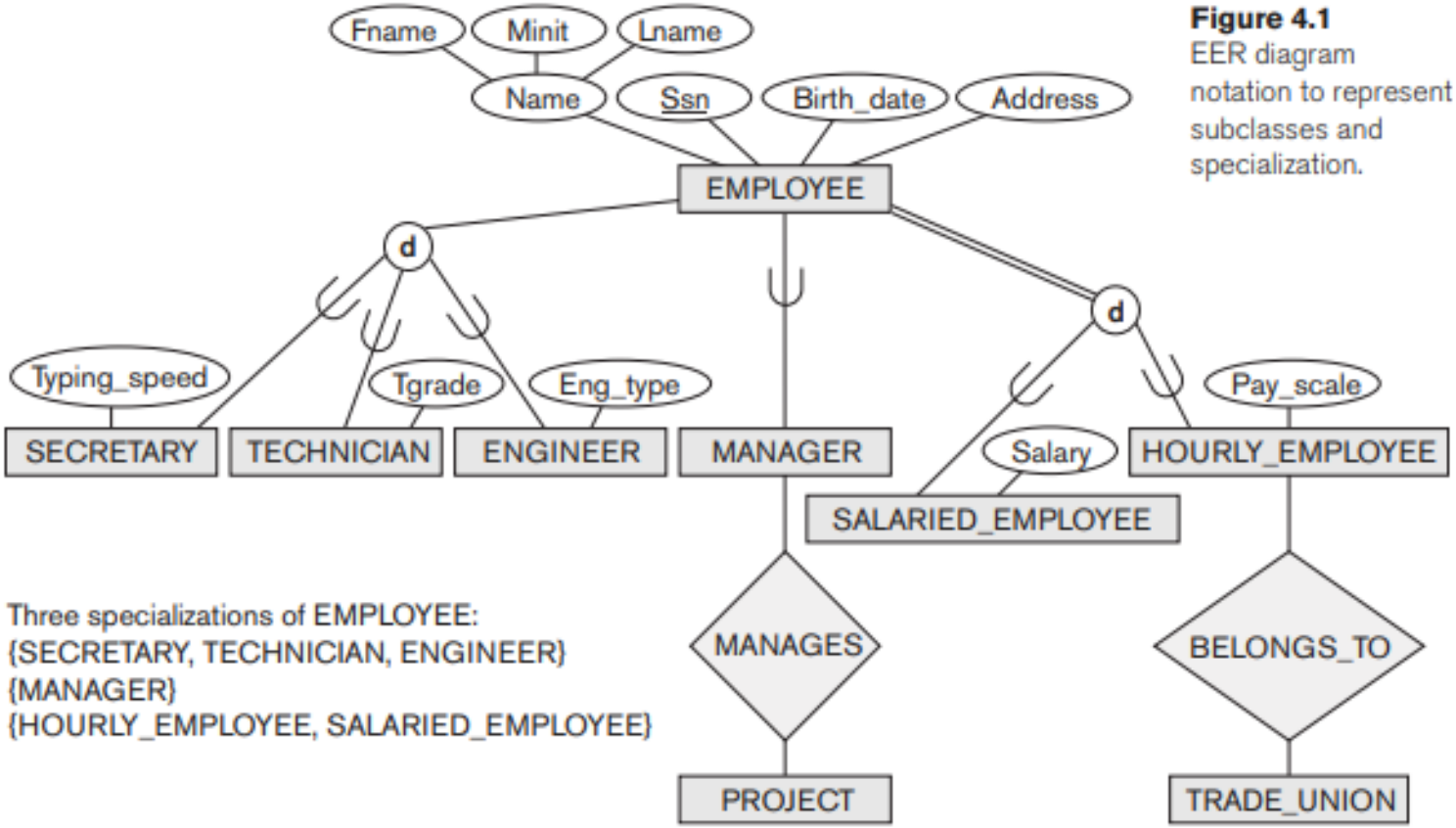


Figure 4.1
EER diagram notation to represent subclasses and specialization.

[3]A class/subclass relationship is often called an **IS-A** (or **IS-AN**) **relationship** because of the way we refer to the concept. We say a SECRETARY *is an* EMPLOYEE, a TECHNICIAN *is an* EMPLOYEE, and so on.

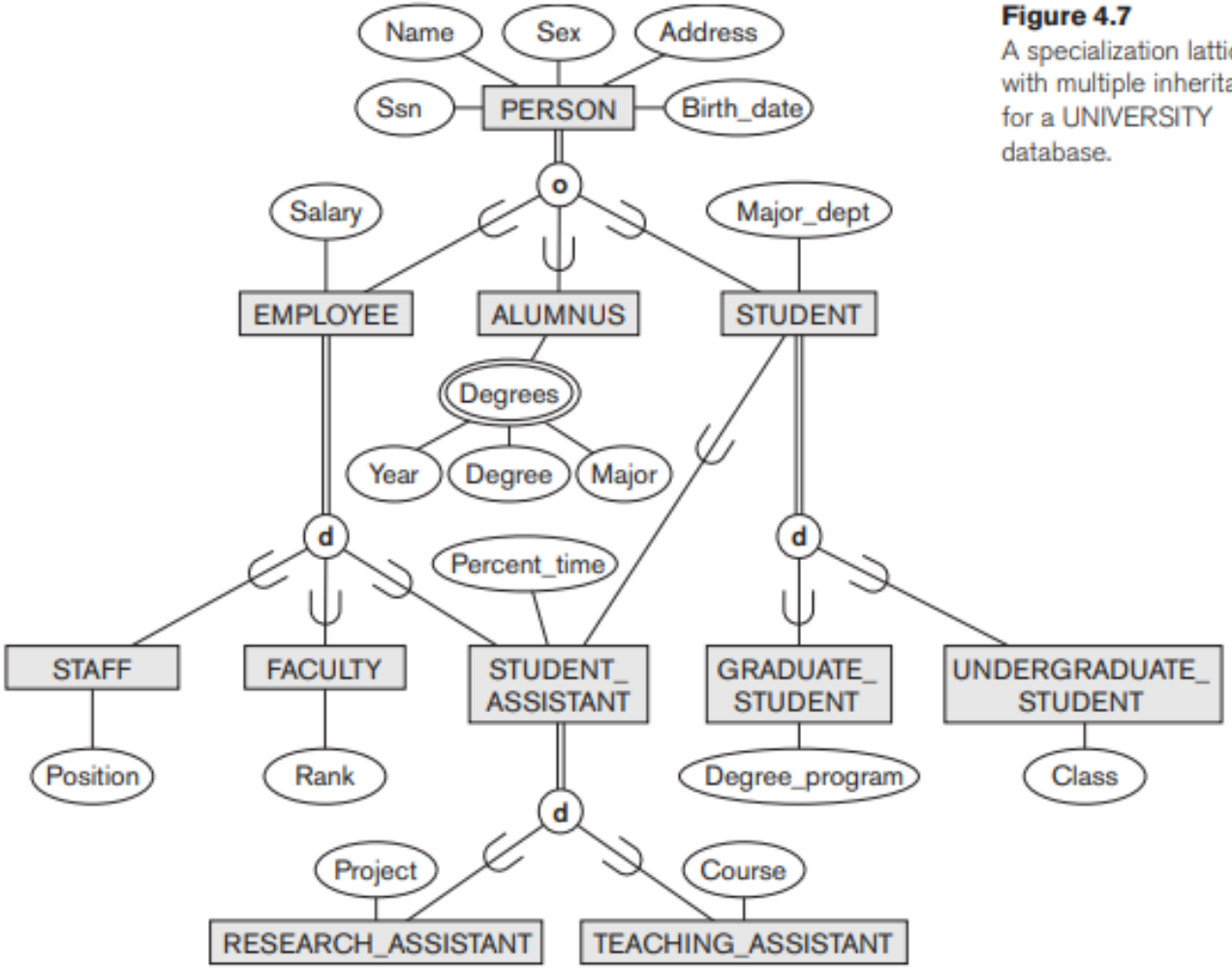


Figure 4.7
A specialization lattice with multiple inheritance for a UNIVERSITY database.

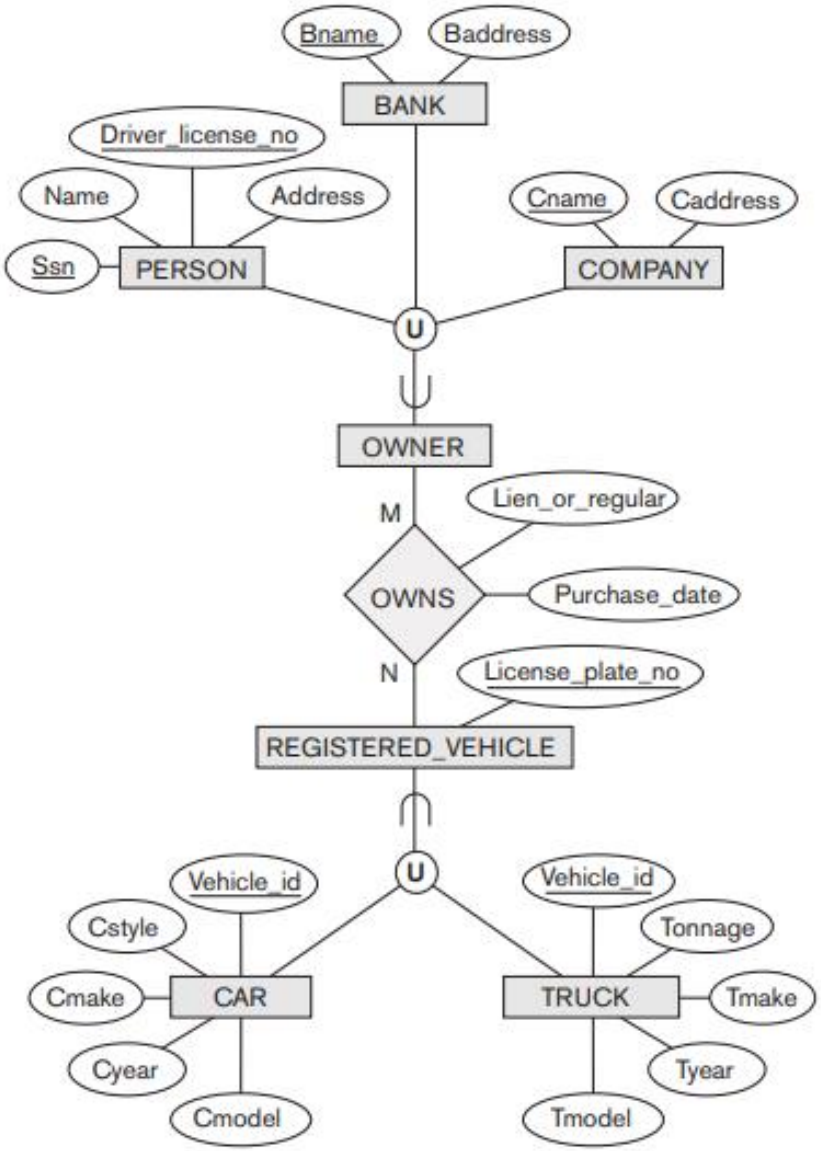


Figure 4.8
Two categories (union types): OWNER and REGISTERED_VEHICLE.

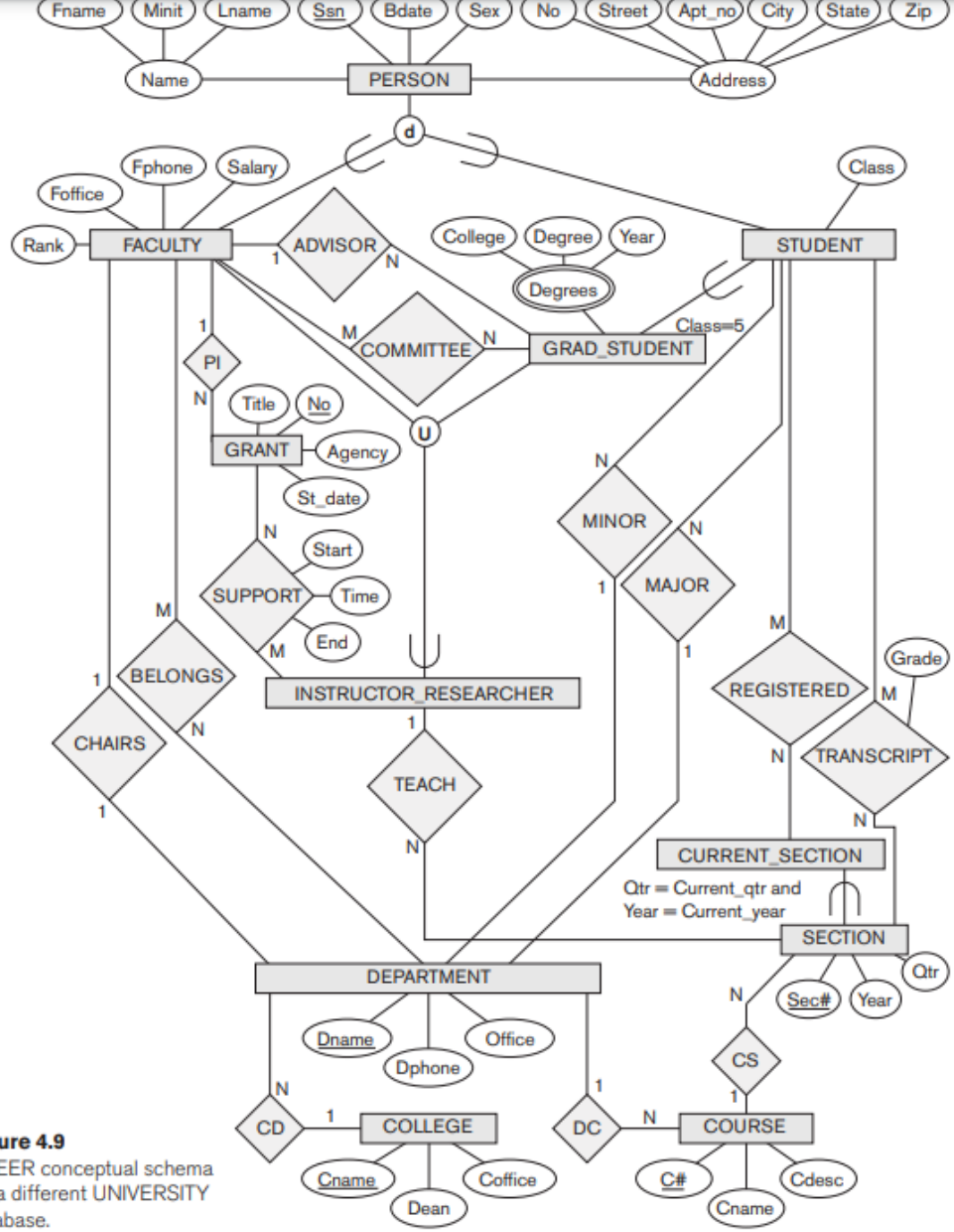


Figure 4.9
An EER conceptual schema for a different UNIVERSITY database.