



Chương 2: Thiết kế Cơ sở dữ liệu bằng cách sử dụng mô hình ER

Giới thiệu

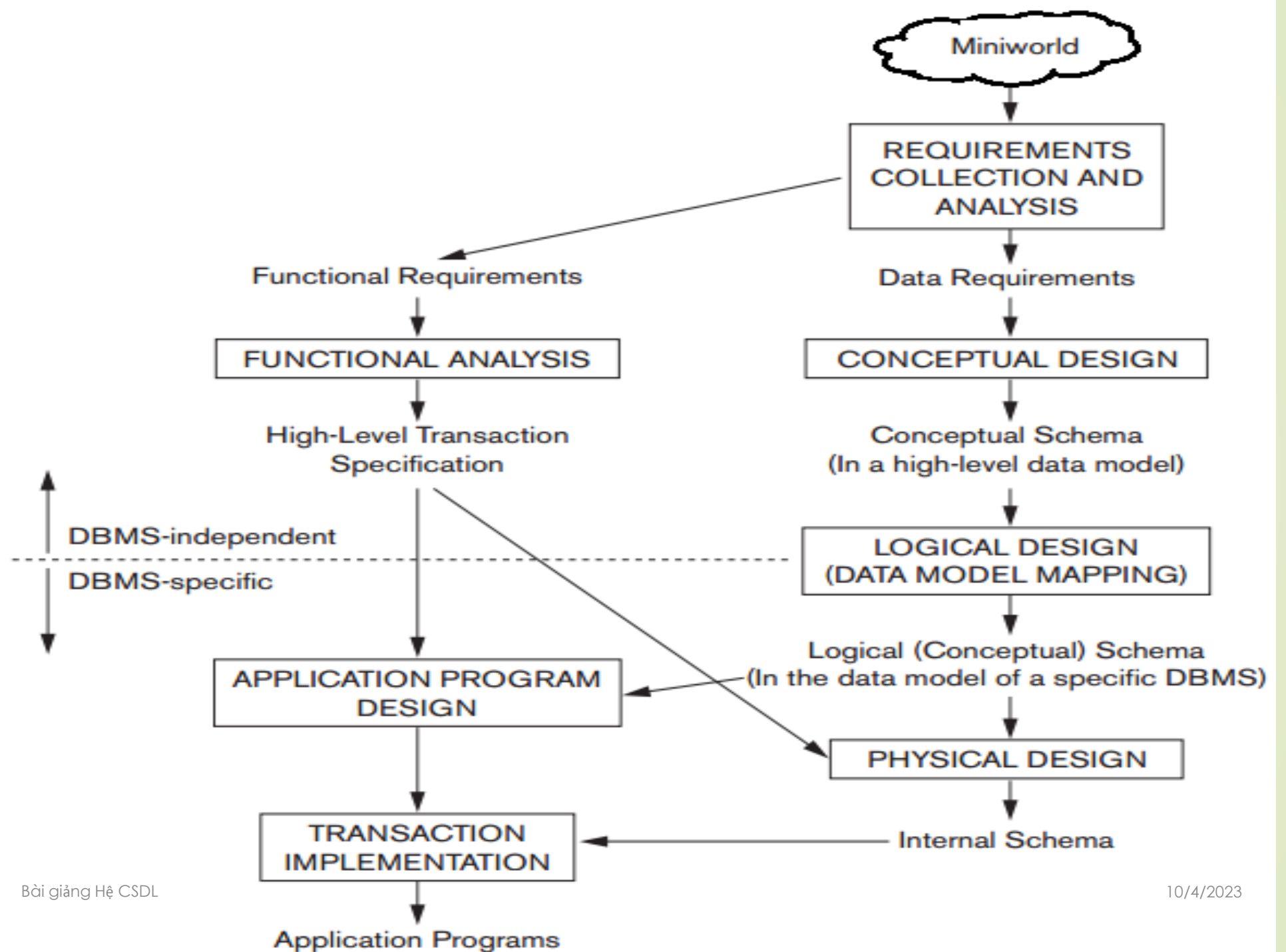
- Mô hình hóa cơ sở dữ liệu bằng cách sử dụng kỹ thuật đồ họa để biểu diễn các **đối tượng** và **quan hệ giữa các đối tượng** trong thế giới thực.
- Mô hình thực thể quan hệ (Entity Relationship - ER) dựa trên thế giới thực trong đó bao gồm một tập các đối tượng gọi là thực thể và mối quan hệ giữa các thực thể thực thể.

Các bước thiết kế một CSDL

- **Bước 1:** Phân tích các yêu cầu của người dùng
- **Bước 2:** Tạo lược đồ khái niệm CSDL bằng cách sử dụng mô hình dữ liệu khái niệm mức cao (high-level conceptual data model).
- **Lược đồ khái niệm là sự mô tả:**
 - Những yêu cầu của người dùng
 - Chi tiết về các loại thực thể.
 - Mối quan hệ giữa các loại thực thể
 - Các ràng buộc.

Các bước thiết kế một CSDL

- **Bước 3 -Thiết kế logic:** bằng cách sử dụng mô hình cơ sở dữ liệu quan hệ hoặc mô hình cơ sở dữ liệu hướng đối tượng.
 - Kết quả là một lược đồ cơ sở dữ liệu
 - Hiện thực cơ sở dữ liệu trong một hệ quản trị cơ sở dữ liệu.
- **Bước 4 -Thiết kế vật lý**
 - Xác định cấu trúc lưu trữ, các cách truy cập dữ liệu.
 - Các chương trình ứng dụng được thiết kế và hiện thực



Ví dụ CSDL Company

➤ Yêu cầu bài toán:

- Công ty (Company) được tổ chức thành các bộ phận (DEPARTMENTS).
- Mỗi bộ phận có những thông tin: Tên, Số, và một nhân viên là người quản lý bộ phận đó (*MANAGER*), trong cơ sở dữ liệu phải lưu thông tin ngày mà nhân viên đó bắt đầu quản lý bộ phận.
- Mỗi bộ phận thực hiện một số dự án (PROJECTs). Mỗi dự án có Tên dự án, Mã số dự án và vị trí của dự án.

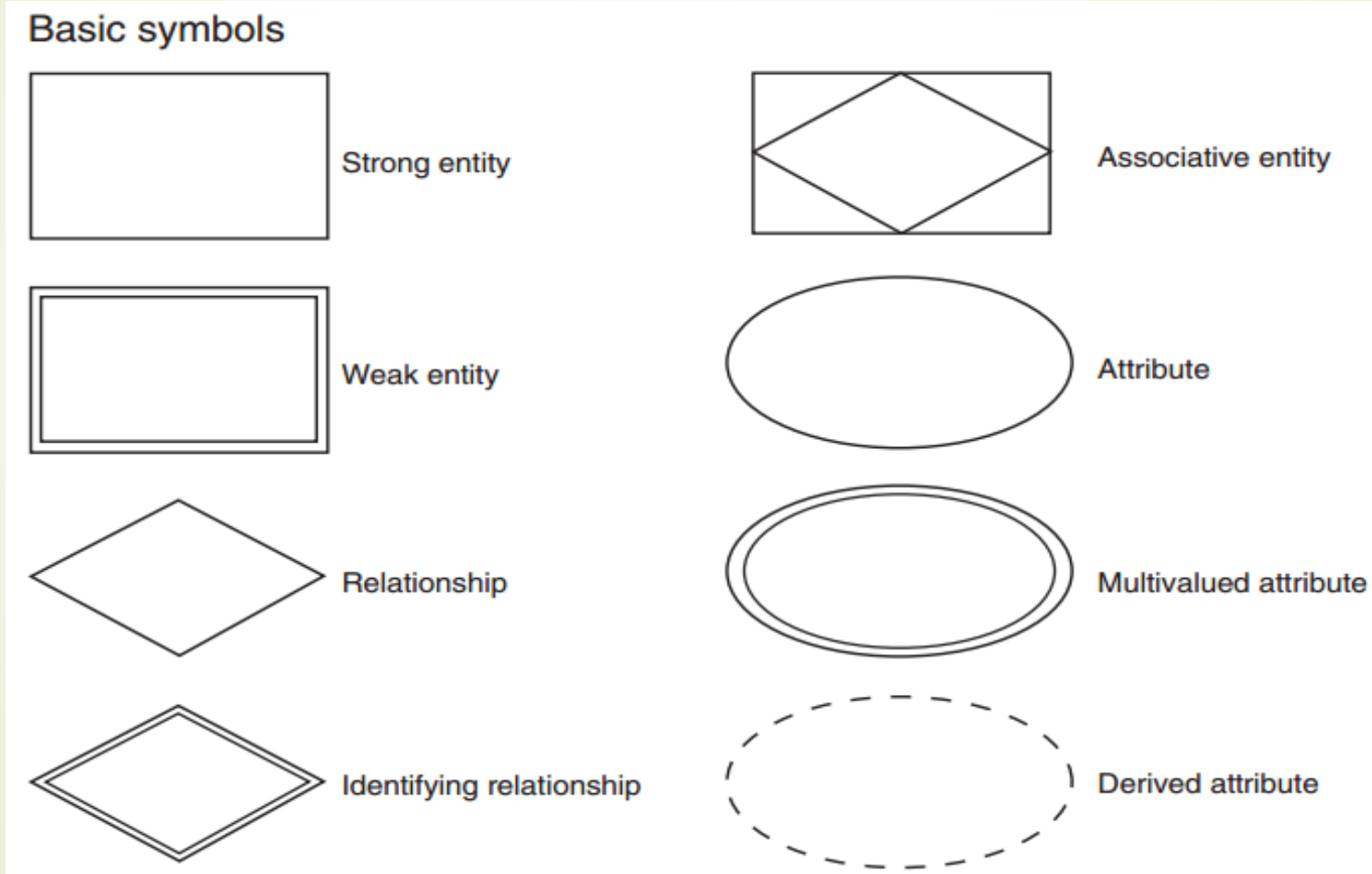
Ví dụ CSDL Company

- Mỗi nhân viên (EMPLOYEE) có Mã nhân viên, địa chỉ, phái, lương, ngày sinh
- Mỗi nhân viên làm việc cho một Phòng ban nhưng có thể tham gia nhiều dự án, trong cơ sở dữ liệu cần lưu thông tin số giờ của mỗi nhân viên làm việc trên mỗi dự án và thông tin về người giám sát của mỗi nhân viên
- Mỗi nhân viên có số người phụ thuộc (DEPENDENTS). Mỗi người phụ thuộc có những thông tin Tên, Phái, Ngày sinh và mối quan hệ với nhân viên

Khái niệm về mô hình ER

- **Mô hình ER** (*Entity–Relationship*) là một công cụ đồ họa để mô hình hóa dữ liệu được mô tả trong yêu cầu của bài toán.
- Mô hình ER gồm 3 thành phần chính
 - Thực thể
 - Thuộc tính
 - Mối quan hệ

Các ký hiệu trong mô hình ER



Thực thể và loại thực thể

- **Thực thể** là thành phần chính trong mô hình ER, bao gồm các đối tượng tồn tại độc lập với các đối tượng khác
Ví dụ: con người, sự vật ...
- **Loại thực thể (Entity type)**: là tập hợp các thực thể có cùng thuộc tính. Mỗi loại thực thể trong CSDL được mô tả bằng một tên và danh sách các thuộc tính.

Thực thể và loại thực thể

➤ Ví dụ:

- *Sinh Viên* là một loại thực thể (**entity type**)
- Sinh viên có Mã SV là 555-55-5555 là một thể hiện của thực thể (**entity instance**)

Thực thể và loại thực thể

➤ Biểu diễn thực thể trong mô hình ER

➤ Ký hiệu: hình chữ nhật

➤ Mỗi thực thể được gán một tên (danh từ)

Ví dụ: thực thể SINHVIEN

SINHVIEN

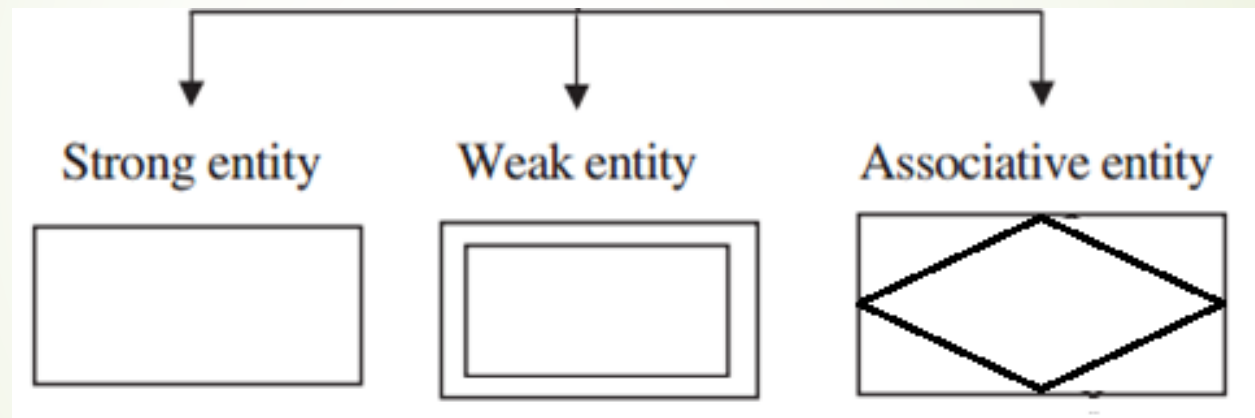
MONHOC

Thực thể và loại thực thể

- **Phân loại:** Có 3 loại thực thể
 - **Thực thể mạnh (*Strong entity*):** thực thể có thuộc tính khóa, không phụ thuộc vào thực thể khác
 - **Thực thể yếu (*Weak entity*):** thực thể tồn tại phụ thuộc vào thực thể khác, thực thể yếu không có thuộc tính khóa
 - **Thực thể kết hợp (*Associative entity*):** là loại thực thể được tạo ra từ sự kết hợp của các thực thể khác, chứa các thuộc tính tạo mối quan hệ giữa các thực thể

Thực thể và loại thực thể

➤ Ký hiệu các loại thực thể trong mô hình ER



Thuộc tính của thực thể

- Thuộc tính dùng để mô tả thực thể

Ví dụ: thực thể NHÂN VIÊN có thể được mô tả bởi tên, tuổi, địa chỉ, lương, và công việc.

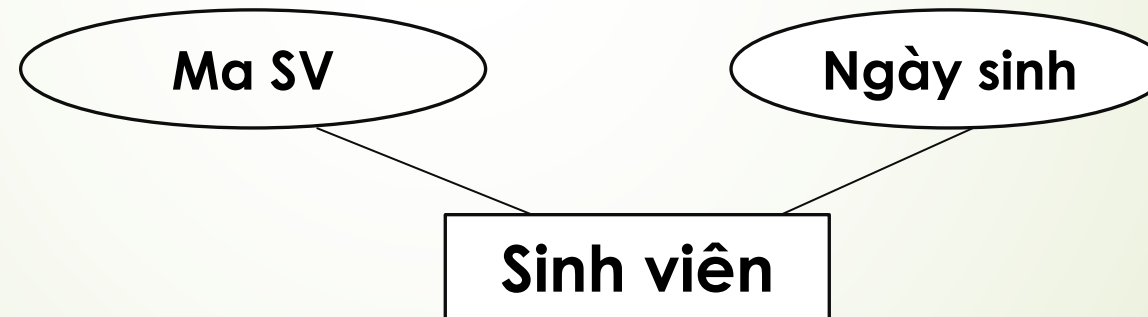
- Một thực thể xác định, thì mỗi thuộc tính có giá trị cụ thể
- Mỗi thuộc tính có một Domain: tập các giá trị được phép cho mỗi thuộc tính
- SINHVIEN(MASV,HOTEN,NGAYSINH)

Phân loại thuộc tính của thực thể

- **Thuộc tính đơn (simple or atomic attributes):** thuộc tính không thể chia nhỏ

Ví dụ: Họ SV, Ten SV

- **Biểu diễn trong mô hình ER**

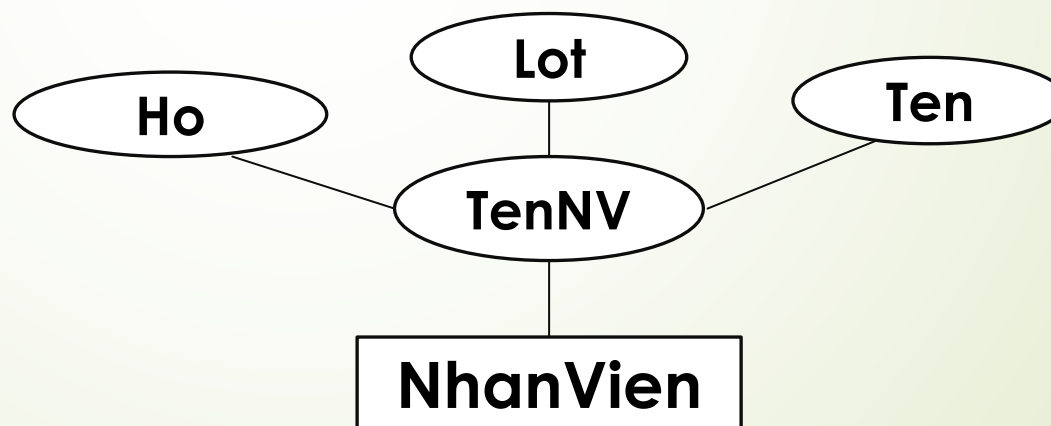


Phân loại thuộc tính của thực thể

- **Thuộc tính kết hợp (Composite attributes):** thuộc tính có thể chia nhỏ thành nhiều thuộc tính

Ví dụ: Họ tên SV, có thể chia thành hai thuộc tính Họ SV và Tên SV

- **Biểu diễn trong mô hình ER**

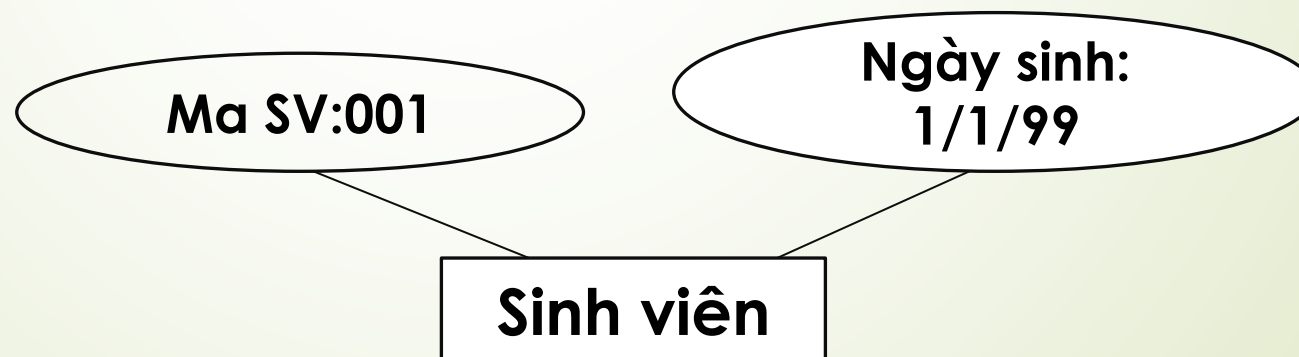


Phân loại thuộc tính của thực thể

- **Thuộc tính đơn trị (Single-Valued):** thuộc tính chỉ có một giá trị cho một thực thể riêng biệt.

Ví dụ: Mã SV, Ngày Sinh chỉ có một giá trị cho một sinh viên cụ thể.

- Biểu diễn trong mô hình ER



Phân loại thuộc tính của thực thể

- **Thuộc tính đa trị (Multivalued Attributes):** thuộc tính có nhiều giá trị cho một thực thể riêng biệt.

Ví dụ: một sinh viên có nhiều kỹ năng, thuộc tính kỹ năng là đa trị.

- **Biểu diễn trong mô hình ER**



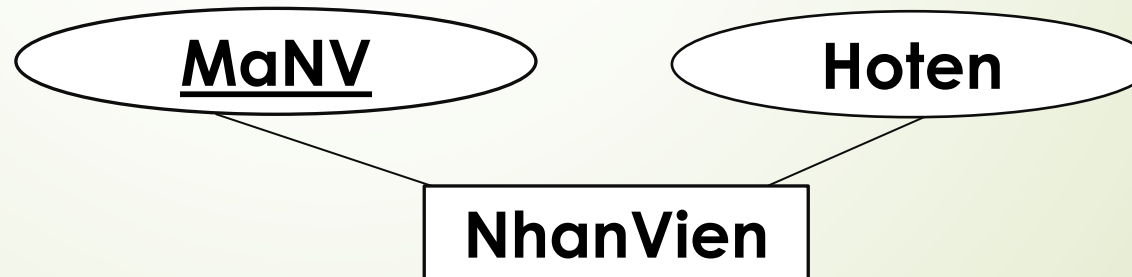
Phân loại thuộc tính của thực thể

- **Thuộc tính lưu trữ (Stored):** thuộc tính không được suy ra từ thuộc tính khác
Ví dụ: Ngày sinh
- **Thuộc tính suy diễn (Derived Attributes):** thuộc tính được suy diễn từ thuộc tính khác
Ví dụ: Tuổi được suy ra từ ngày sinh.
- **Biểu diễn trong mô hình ER**



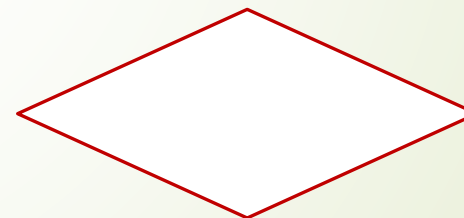
Phân loại thuộc tính của thực thể

- Thuộc tính **khóa** của một loại thực thể là thuộc tính dùng để xác định một thực thể riêng biệt trong tập thực thể.
- Thực thể ***không có thuộc tính khóa*** gọi là ***thực thể yếu***
- Biểu diễn trong mô hình ER



Mối quan hệ - Relationships

- **Mối quan hệ (Relationships)** là sự kết hợp của nhiều loại thực thể.
- Một mối quan hệ được xác định bằng một **động từ**
- **Loại mối quan hệ (Relationship type)** R là tập hợp các mối quan hệ giữa các thực thể riêng biệt trong các loại thực thể.
- Ký hiệu trong mô hình ER:



Mối quan hệ - Relationships

- Ví dụ: Sinh viên học các Môn học
 - Thực thể: Sinh viên và thực thể Môn Học có mối quan hệ học



Bậc của mỗi quan hệ

- **Bậc của mỗi quan hệ** là số loại thực thể (Entity type) tham gia vào mỗi quan hệ

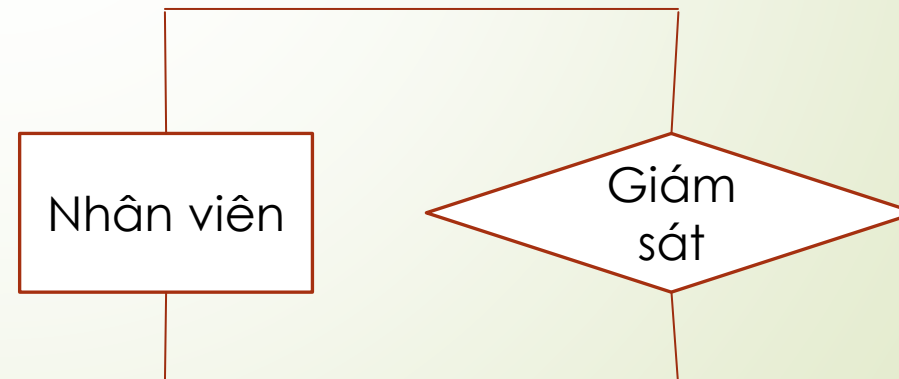
Ví dụ: quan hệ giữa 2 thực thể Sinh viên và Môn học là quan hệ bậc 2 (quan hệ nhị phân)



Bậc của mối quan hệ

- Gọi E_1, E_2, \dots, E_n là n tập thực thể, R là mối quan hệ giữa các tập thực thể. Bậc của R :
 - **Quan hệ một ngôi (unary)**: là sự kết hợp giữa 2 thực thể riêng biệt của cùng một loại thực thể ($R \in E_1 \times E_1$), còn gọi là quan hệ đệ quy.

Ví dụ: quan hệ Giám sát của thực thể Nhân viên



Bậc của mối quan hệ

- **Quan hệ nhị phân (Binary):** là sự kết hợp giữa hai thực thể riêng biệt của hai loại thực thể khác nhau ($R \in E_1 \times E_2$).

Ví dụ:

Quan hệ Tham gia giữa 2 thực thể **Nhân viên** và **Dự án**

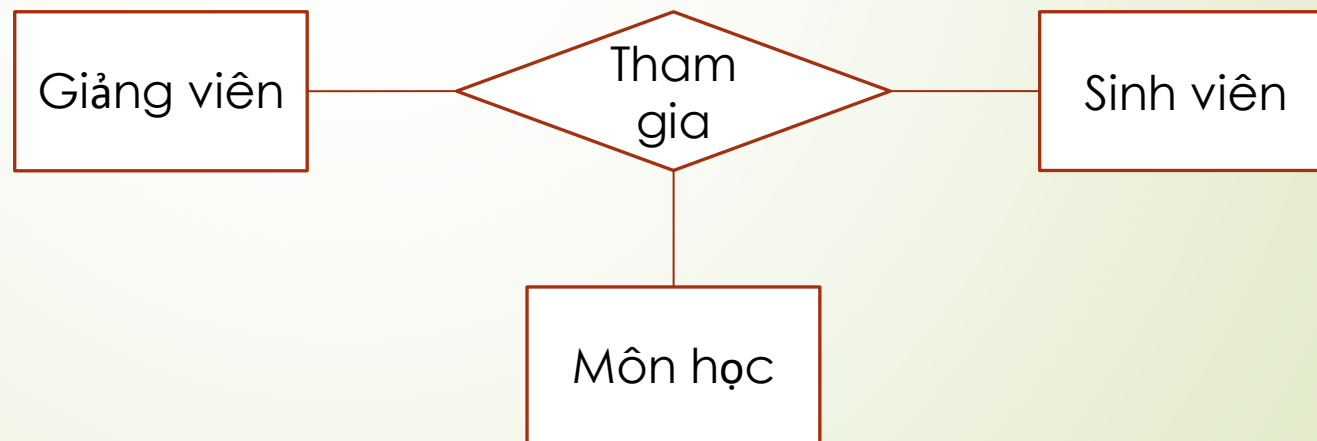


Bậc của mối quan hệ

- **Quan hệ bậc >2 (Ternary)** là sự kết hợp của >2 thực thể riêng biệt của các loại thực thể khác nhau ($R \in E_1 \times E_2 \times E_3$).

Ví dụ:

Quan hệ Dạy giữa các thực thể Giảng viên, Sinh viên và Môn học



Các ràng buộc trên mối quan hệ

- Có hai ràng buộc chính trên mối quan hệ nhị phân:
 - Ràng buộc bằng số (Cardinality)
 - Ràng buộc tham gia (Participation)

Các ràng buộc trên mối quan hệ

➡ **Cardinality**: xác định số tối thiểu / tối đa các thể hiện của một loại thực thể có thể tham gia vào mối quan hệ .

➡ Ví dụ:

➡ Quan hệ **Làm việc** giữa thực thể **Phòng ban** và thực thể **Nhân Viên** có **cardinality** là **1:N**.

Nghĩa là:

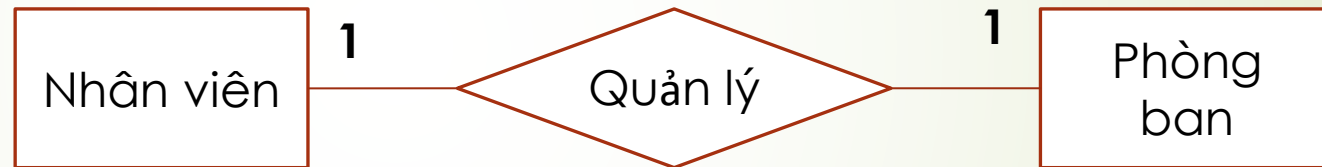
➡ Một **Phòng ban** có quan hệ với nhiều **Nhân Viên**

➡ Ngược lại, một **Nhân Viên** chỉ quan hệ với một **Phòng ban**

Các ràng buộc trên mối quan hệ

- **Quan hệ 1:1** tối đa một thể hiện của loại thực thể B kết hợp với một thể hiện của loại thực thể A và ngược lại.

- Ví dụ:



- Ý nghĩa:

- Một **Nhân viên** quản lý một **Phòng ban**, ngược lại, một **Phòng ban** được quản lý bởi một **Nhân viên**

Các ràng buộc trên mối quan hệ

- **Quan hệ 1:N** nhiều thực thể của loại thực thể B kết hợp với một thực thể của loại thực thể A, một thực thể của loại thực thể A kết hợp với một thực thể của loại thực thể B

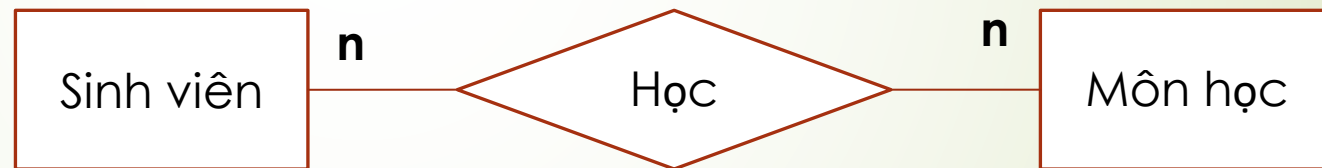
- Ví dụ:



- Ý nghĩa: Một **Nhân viên** làm việc tại một **Phòng ban**, ngược lại, một **Phòng ban** có nhiều **Nhân viên** làm việc

Các ràng buộc trên mối quan hệ


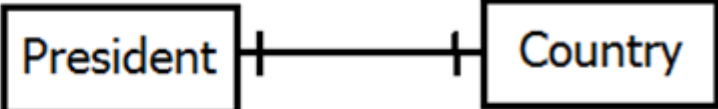

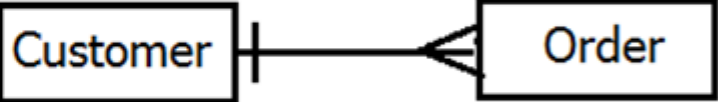

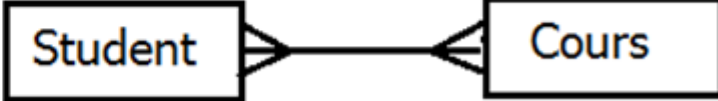
- **Quan hệ M:N** nhiều thực thể của loại thực thể A quan hệ với nhiều thực thể của loại thực thể B và ngược lại
- Ví dụ: quan hệ Học giữa 2 thực thể Sinh viên và Môn học



- Ý nghĩa:
 - Một **Sinh viên** học nhiều **Môn học**, ngược lại, một **Môn học** được học bởi nhiều **Sinh viên**

Các ràng buộc trên mối quan hệ

- Ký hiệu trong mô hình ER: có thể dùng bảng số hoặc ký hiệu:

Loại mối quan hệ	Ký hiệu	Ví dụ
1:1		
1:N		
N:M		

Các ràng buộc trên mối quan hệ

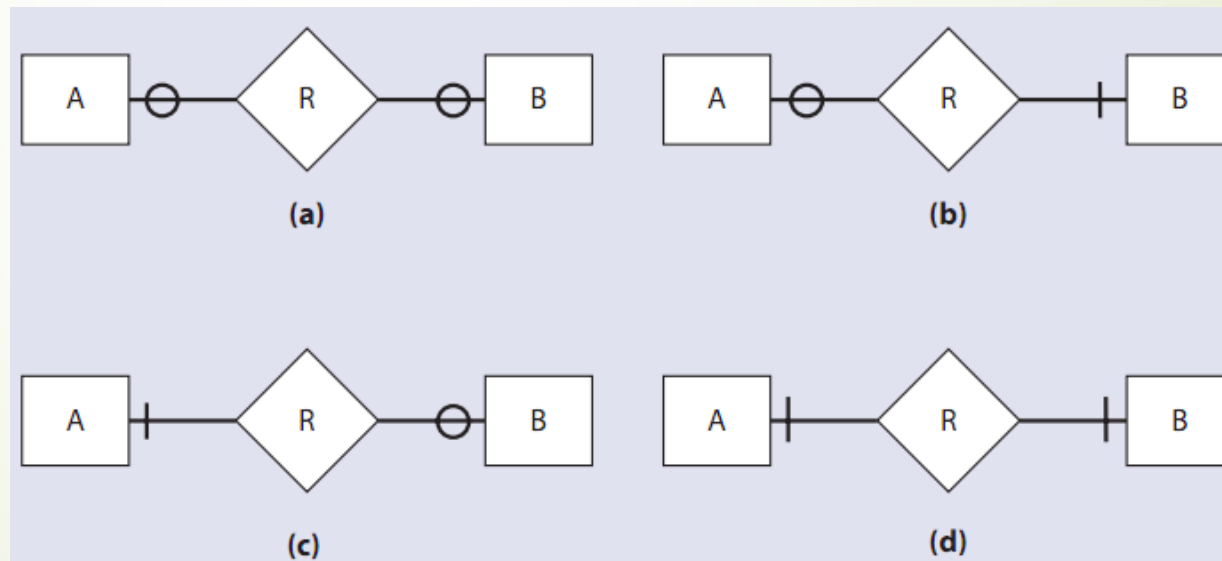
- **Ràng buộc tham gia** (Participation Constraints):
xác định số tối thiểu của thể hiện của mối quan hệ mà mỗi thực thể có thể tham gia, còn gọi là *minimum cardinality constraint*
- **Có hai loại**
 - Tham gia từng phần – Partial participation
 - Tham gia toàn phần - Total participation

Các ràng buộc trên mối quan hệ

- Trong mô hình E-R, số tối thiểu có 2 giá trị **0 hoặc 1**.
 - Nếu số tối thiểu là **0** thì loại thực thể tham gia tùy chọn vào mối quan hệ (**optional participant**)
 - Nếu số tối thiểu là 1 thì loại thực thể tham gia bắt buộc vào mối quan hệ (**mandatory participant**)

Các ràng buộc trên mối quan hệ

- Ký hiệu trong mô hình ER:
 - Tham gia tùy chọn được ký hiệu là “O”
 - Tham gia bắt buộc ký hiệu “|”
- Ví dụ:



Các ràng buộc trên mối quan hệ

- **Tham gia từng phần – Partial:** chỉ có một số thực thể trong loại thực thể tham gia vào mối quan hệ.
 - Ví dụ: một số thực thể **Nhân viên** của Loại thực thể **Nhân viên** không tham gia vào mối quan hệ **Quản lý**.

Các ràng buộc trên mối quan hệ

- **Tham gia toàn phần - Total participation:** tất cả các thực thể trong loại thực thể đều tham gia vào mối quan hệ.
 - Ví dụ: tất cả thực thể của Loại thực thể **Nhân viên** đều tham gia vào mối quan hệ **Làm việc**
- **Thực thể yếu** tham gia toàn phần vào mối quan hệ với thực thể chủ.

MÔ HÌNH THỰC THỂ KẾT HỢP MỞ RỘNG (Enhanced Entity Relationship Model - EER)

Khái niệm về mô hình EER

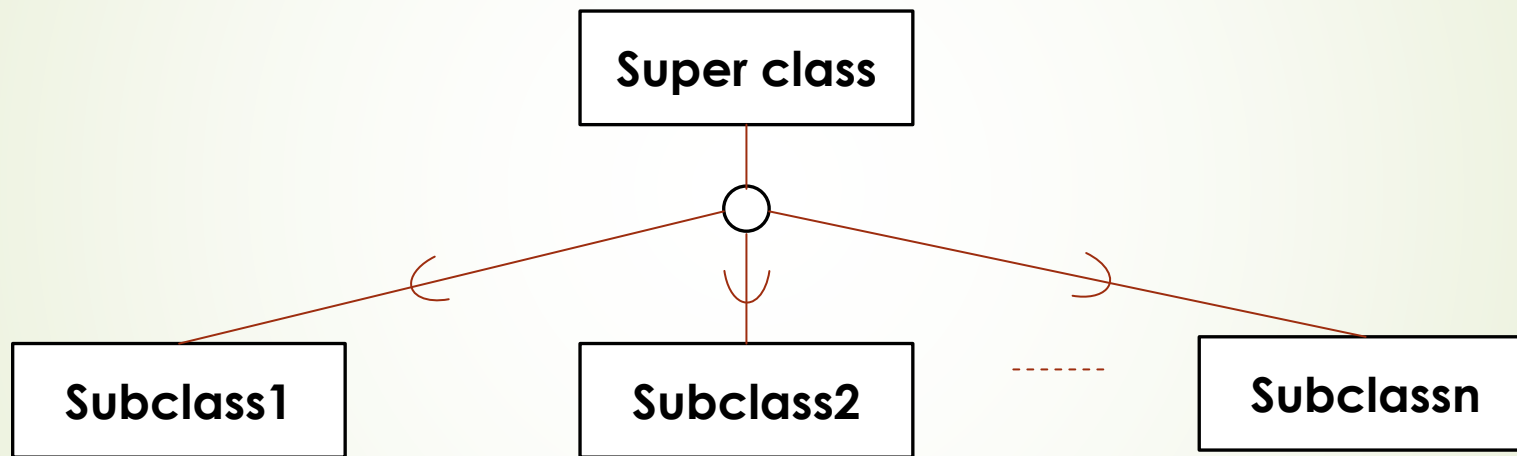
- Mô hình EER bao gồm tất cả khái niệm của mô hình ER cơ bản và thêm các khái niệm sau:
 - Lớp cha/con (Subclass/super class).
 - Chuyên biệt hóa (Specialization)/Tổng quát hóa (generalization)
- Được sử dụng để mô hình hóa các ứng dụng phức tạp.

Subclass - Superclass

- **Superclass** là một loại thực thể mà nó có một hoặc nhiều nhóm con riêng biệt với những thuộc tính duy nhất.
 - Superclass chỉ chứa những thuộc tính chung của tất cả các nhóm con.
 - Các nhóm con với những thuộc tính duy nhất được gọi là subclasses.
- **Subclass** là một loại thực thể mà nó có thể chia sẻ những thuộc tính chung hoặc các mối quan hệ riêng biệt từ những subclass khác.

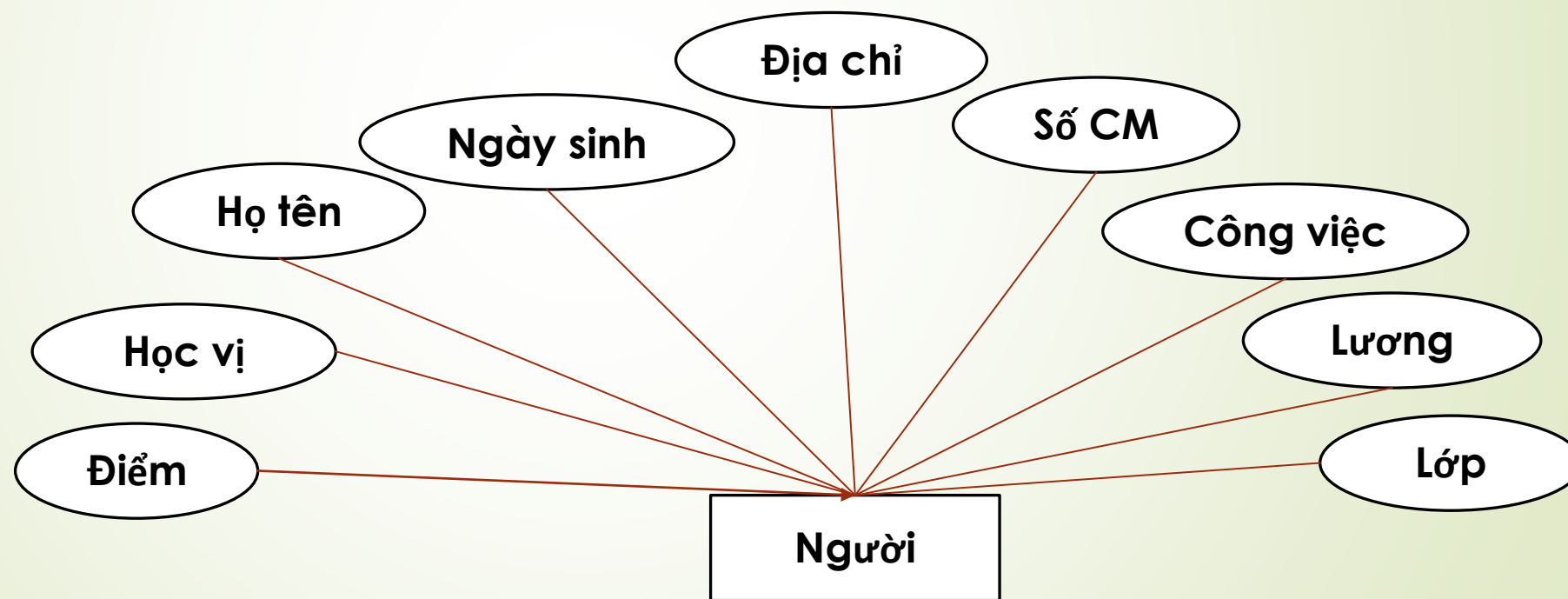
Subclass - Superclass

➡ Ký hiệu trong mô hình ER



Subclass - Superclass

- ▶ Ví dụ: thực thể **Người** là tập các thực thể với các thuộc tính như sau:

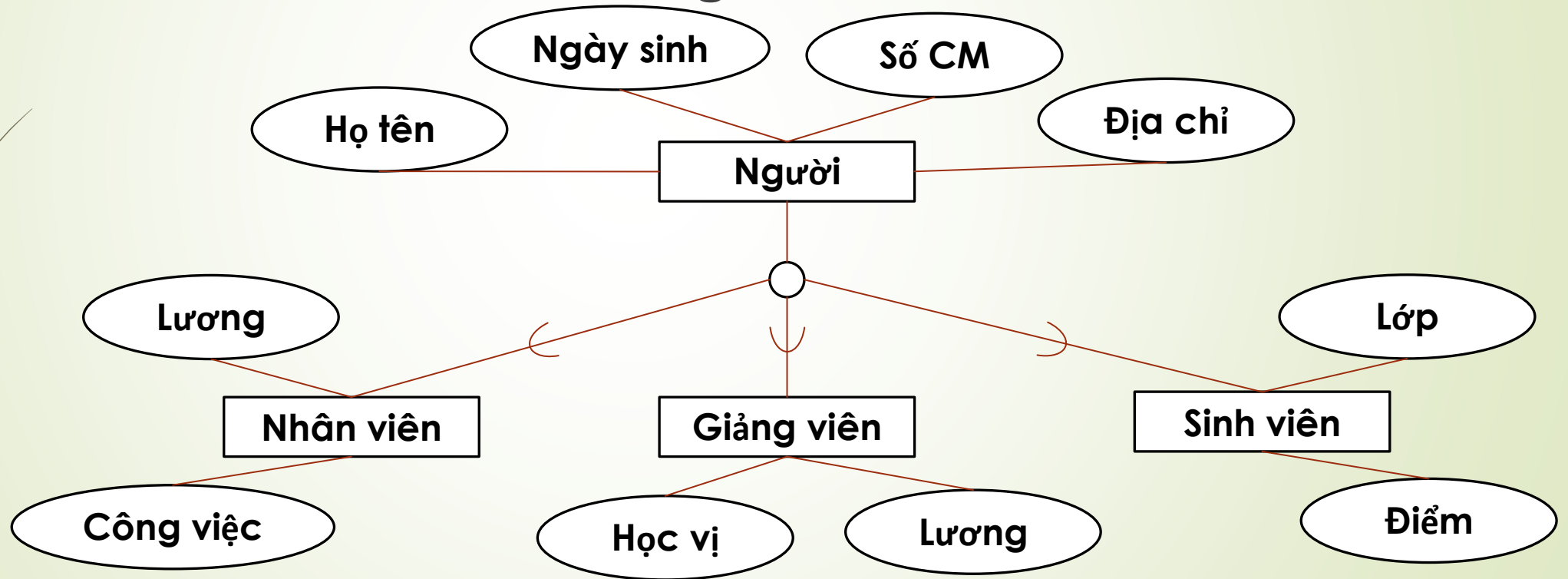


Subclass - Superclass

- **Cách biểu diễn trên có các vấn đề:**
 - Dư thừa dữ liệu
 - Cơ sở dữ liệu không thể hiện tính hiệu quả.
- **Giải pháp:**
 - Dùng mô hình lớp Superclass /Subclass

Subclass - Superclass

- Ví dụ: thực thể Người được tách thành các thực thể con: **Nhân viên**, **Giảng viên**, **Sinh viên**



Subclass - Superclass

➤ Thuộc tính kế thừa

- Những thuộc tính mà các thực thể của lớp con (subclass) kế thừa từ các thuộc tính của lớp cha (superclass).
- Thực thể của lớp con (subclass) kế thừa tất cả thuộc tính của lớp cha, đồng thời cũng có những thuộc tính và mối quan hệ riêng của nó.

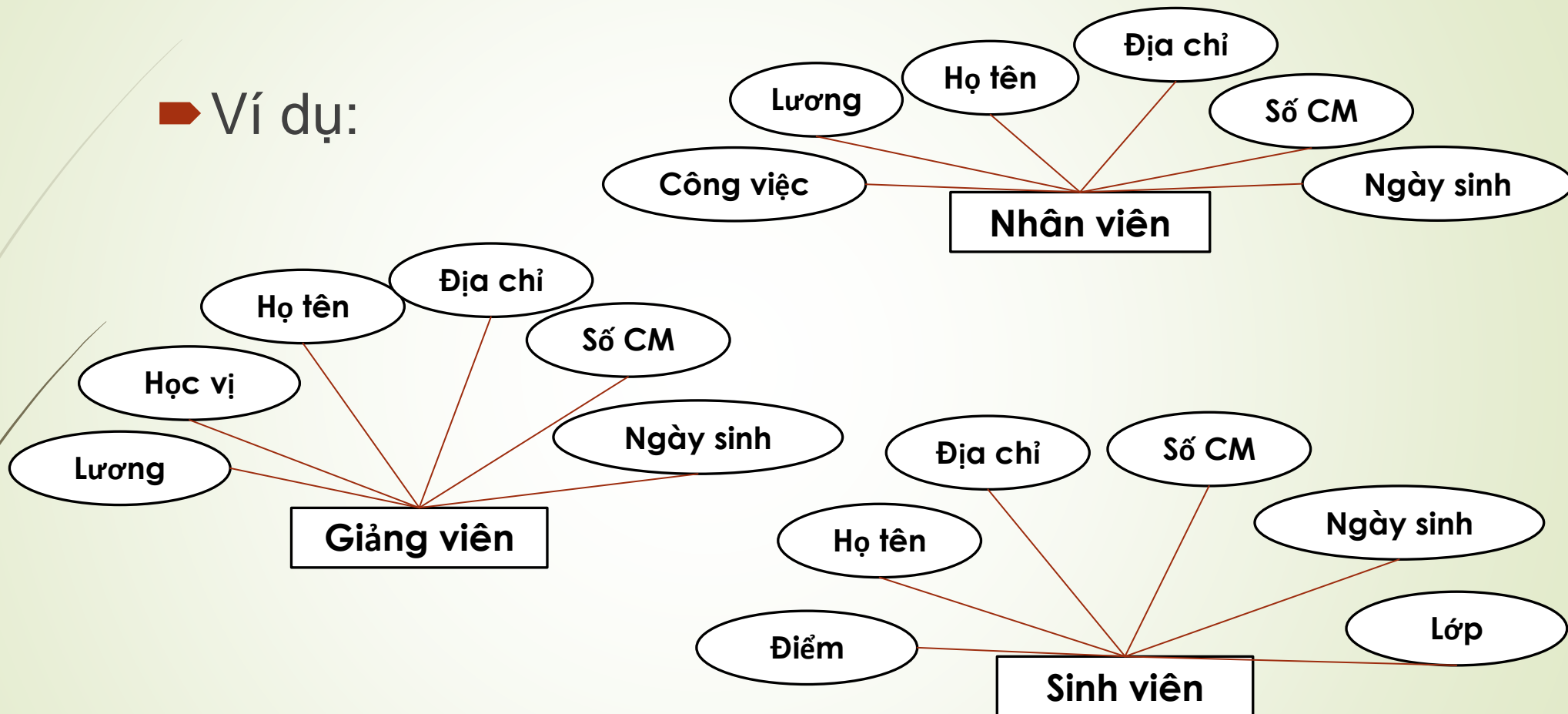
Tiến trình tổng quát hóa và chuyên biệt hóa

➡ Tổng quát hóa (Generalization)

- ➡ Xác định loại thực thể tổng quát từ tập các loại thực thể chuyên biệt bằng cách xác định những thuộc tính chung của nó.
- ➡ Tổng quát hóa là cách tiếp cận từ dưới lên bắt đầu từ loại thực thể của lớp con (subclass) đến loại thực thể của lớp cha (superclass)

Tiến trình tổng quát hóa và chuyên biệt hóa

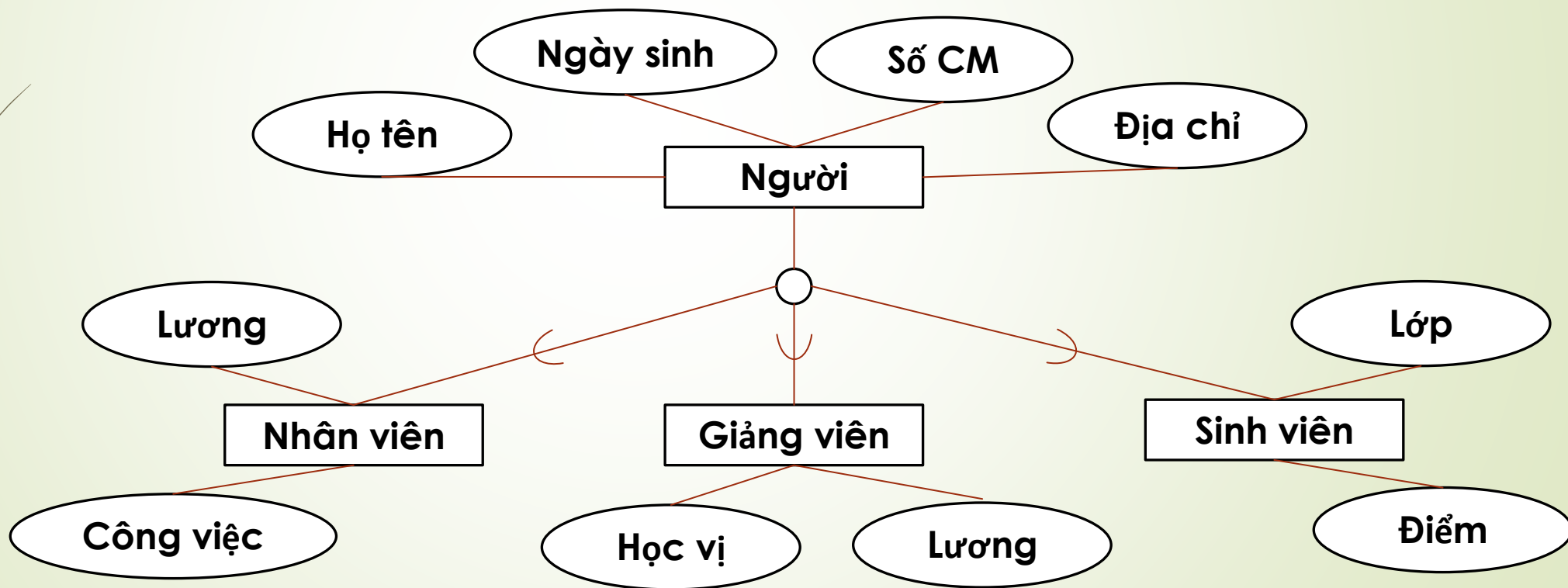
➡ Ví dụ:



Các thực thể Nhân viên, giảng viên và sinh viên trước khi tổng quát hóa

Tiến trình tổng quát hóa và chuyên biệt hóa

- Ví dụ: sau khi tổng quát hóa: những thuộc tính chung được đặt trong lớp cha.



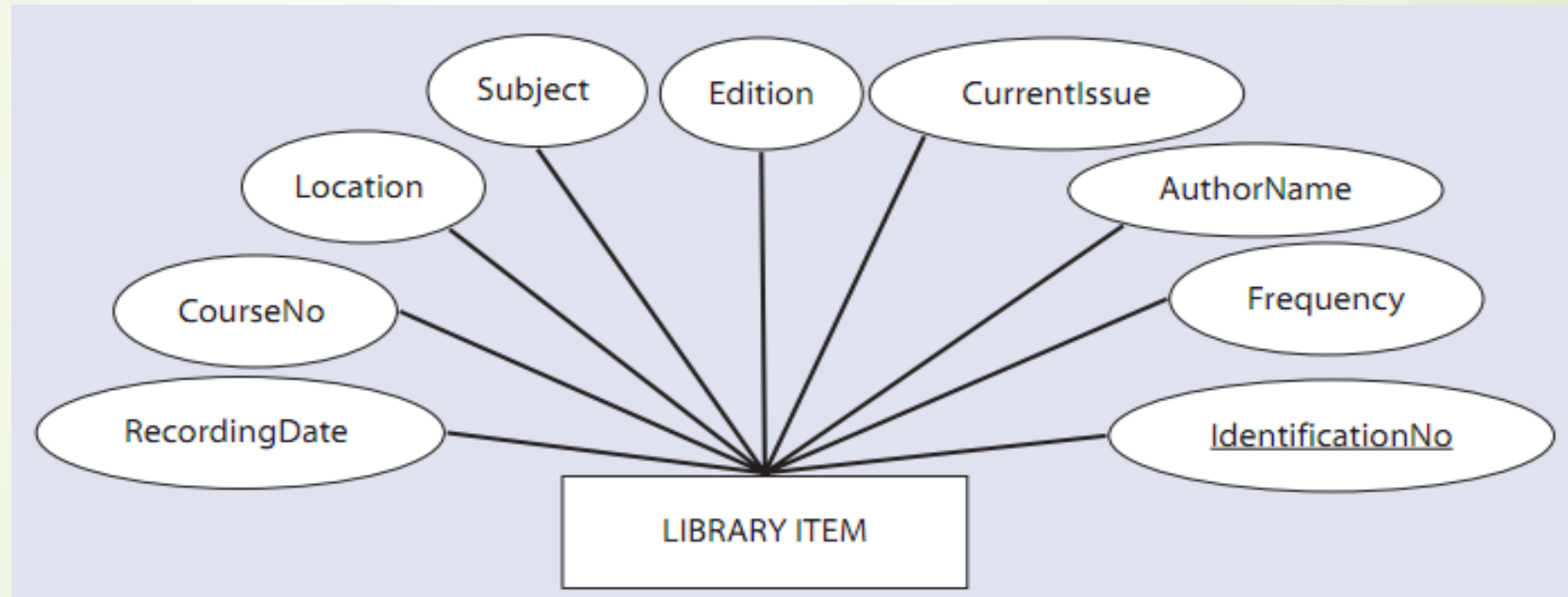
Tiến trình tổng quát hóa và chuyên biệt hóa

➤ **Chuyên biệt hóa (Specialization)**

- Xác định một hoặc nhiều lớp con của một lớp cha bằng cách xác định những thuộc tính riêng biệt của nó.
- Chuyên biệt hóa là cách tiếp cận từ trên xuống, bắt đầu từ loại thực thể tổng quát (superclass) xác định những lớp con (subclasses) dựa trên những thuộc tính riêng hoặc mối quan hệ cụ thể của lớp con

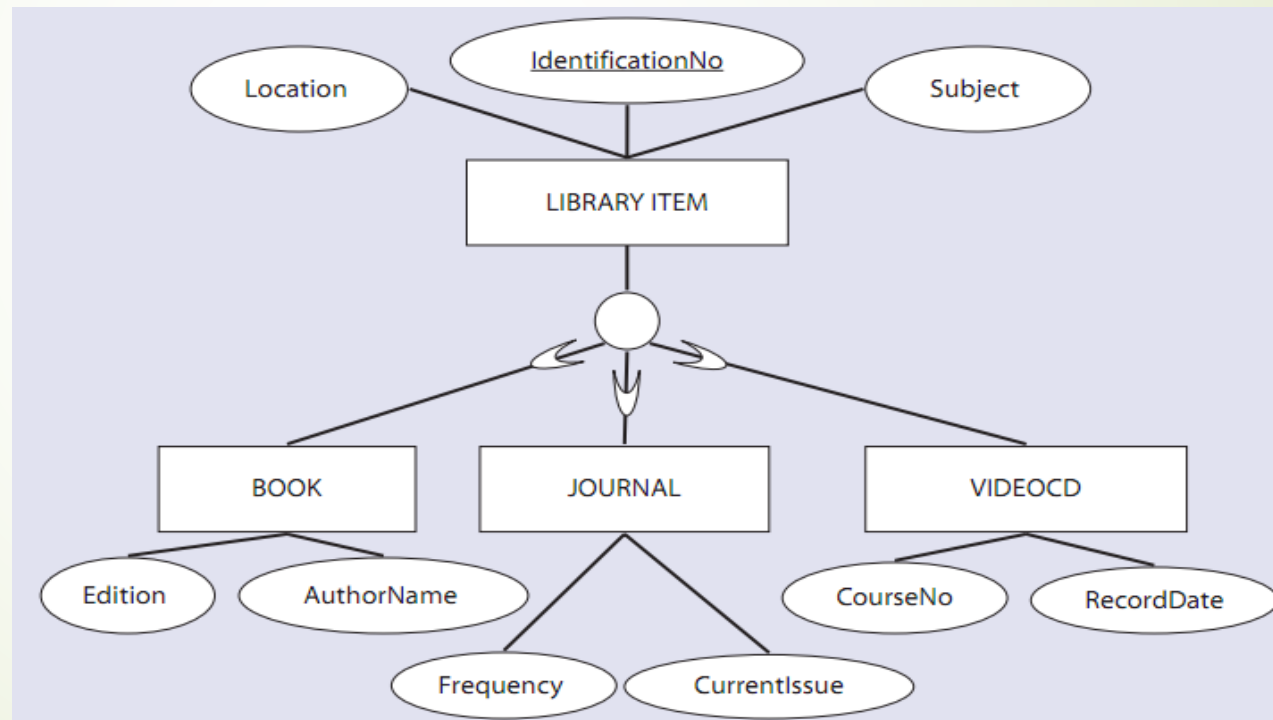
Tiến trình tổng quát hóa và chuyên biệt hóa

► Ví dụ: loại thực thể LIBRARY ITEM



Tiến trình tổng quát hóa và chuyên biệt hóa

- Ví dụ: Sau khi chuyên biệt hóa, lớp LIBRARY ITEM được tách thành các lớp con **Book**, **Journal**, **VideoCd**

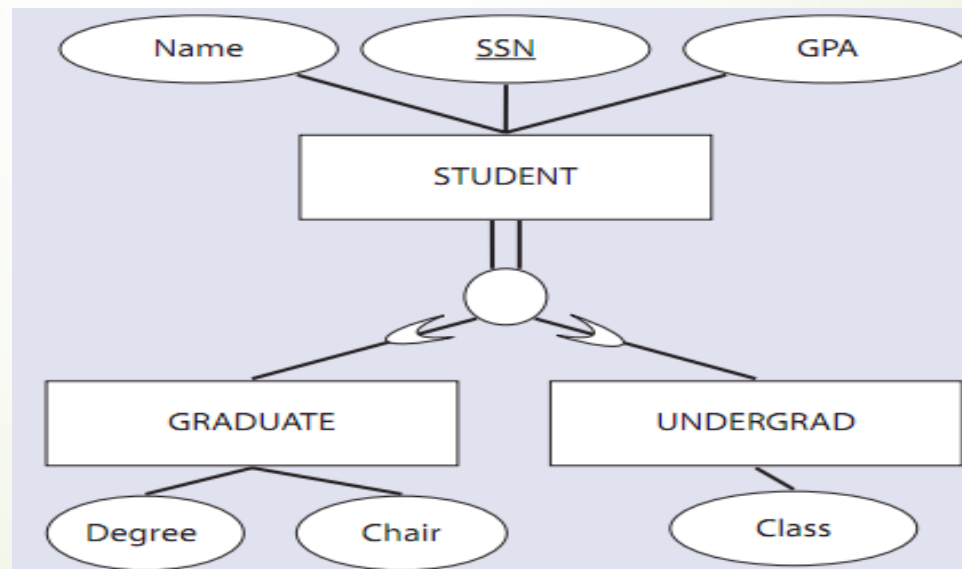


Ràng buộc tham gia

- **Ràng buộc tham gia** (*Participation Constraints*): mỗi thể hiện của một lớp cha (Superclass) phải tham gia như là một thể hiện của một lớp con (Subclass).
- Sự tham gia của một thể hiện của lớp cha có thể là bắt buộc hoặc tùy chọn trong một hoặc nhiều lớp con.
 - Ràng buộc bắt buộc là loại tham gia toàn phần
 - Ràng buộc tùy chọn là tham gia từng phần.

Ràng buộc tham gia

- **Quy tắc tham gia toàn phần (Total Participation Rule):** Trong tham gia toàn phần, thành viên là bắt buộc, mỗi thể hiện của một lớp cha phải là một thể hiện của ít nhất một lớp con.

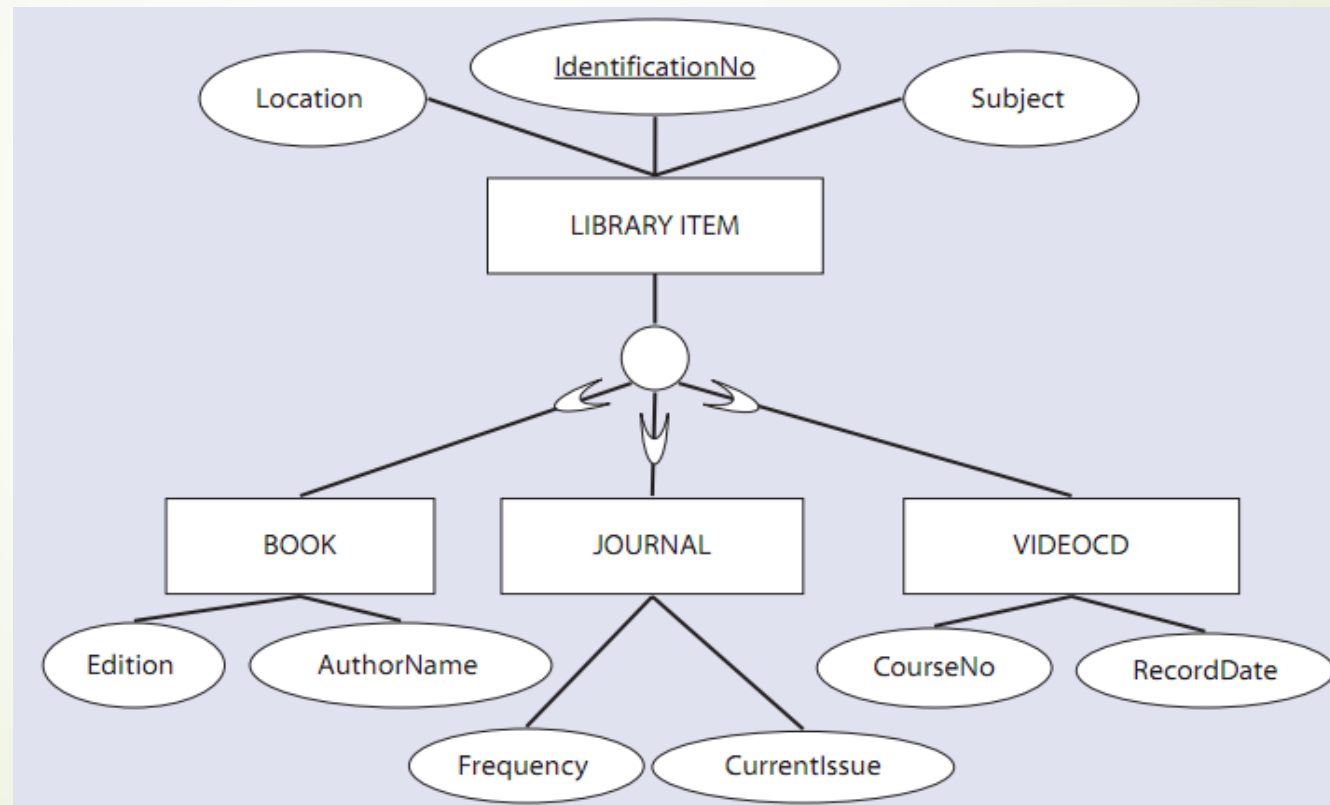


Ràng buộc tham gia

- **Quy tắc ràng buộc tham gia từng phần (*Partial Participation Rule*):** thành viên là tùy chọn trong tham gia từng phần. Một thể hiện của lớp cha có thể không là thành viên của bất kỳ lớp con nào
- Ví dụ:
 - Một thể hiện lớp cha **Library Item** có thể là thành viên của **Book, Video CD, Journals**, nhưng nó không phải là bắt buộc đối với một thể hiện thuộc bất kỳ của các lớp con.
 - Nếu Newspaper là một thể hiện của một lớp cha, nó không thuộc một một trong các lớp con.

Ràng buộc tham gia

➤ Ví dụ:

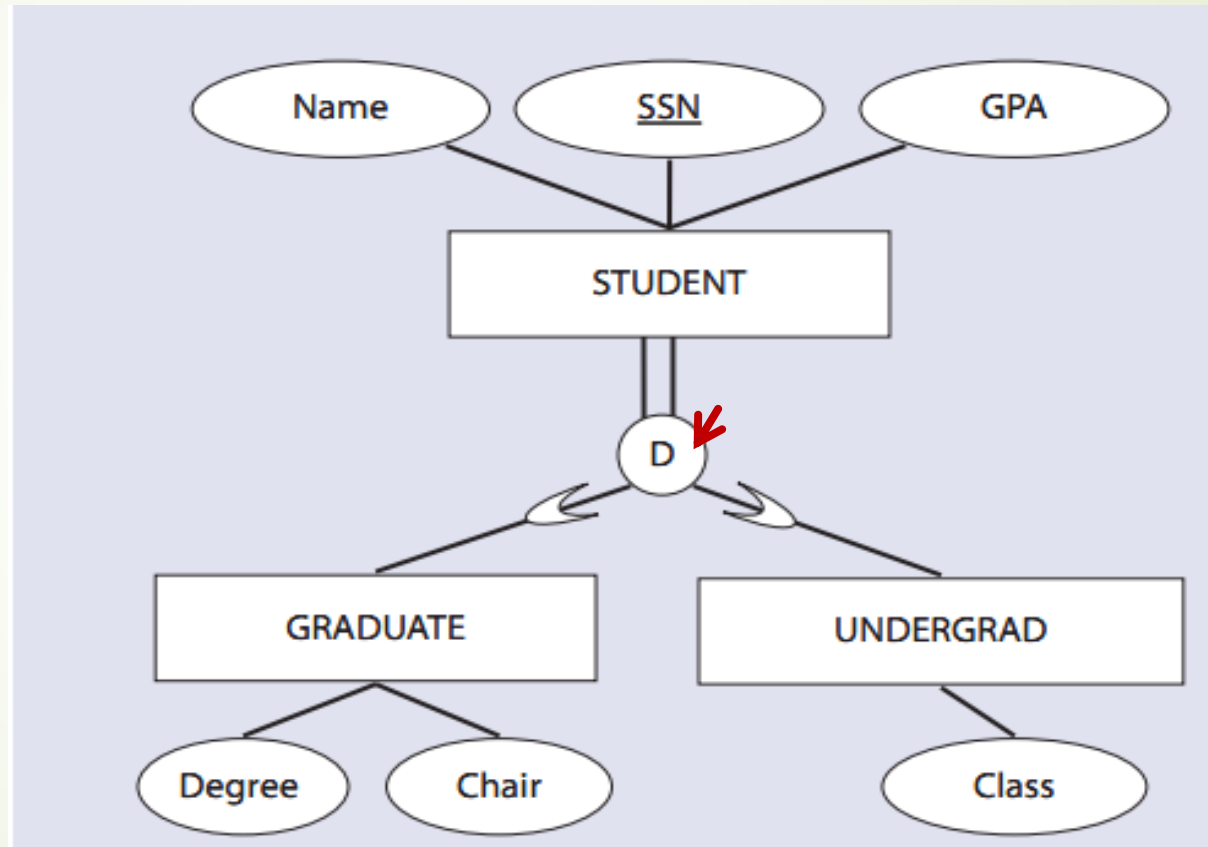


Ràng buộc riêng biệt (Disjoint constraints)

- **Disjoint constraints:** xác định một thể hiện của lớp cha có thể là thành viên của một hoặc nhiều lớp con.
- ***The disjoint rule:*** Nếu một thể hiện của lớp cha là thành viên của bất kỳ một lớp con thì nó không thể là thành viên của bất lớp con khác.
- Trong mô hình EER, ràng buộc Disjoint được ký hiệu là D.

Ràng buộc riêng biệt (Disjoint constraints)

➤ Ví dụ:

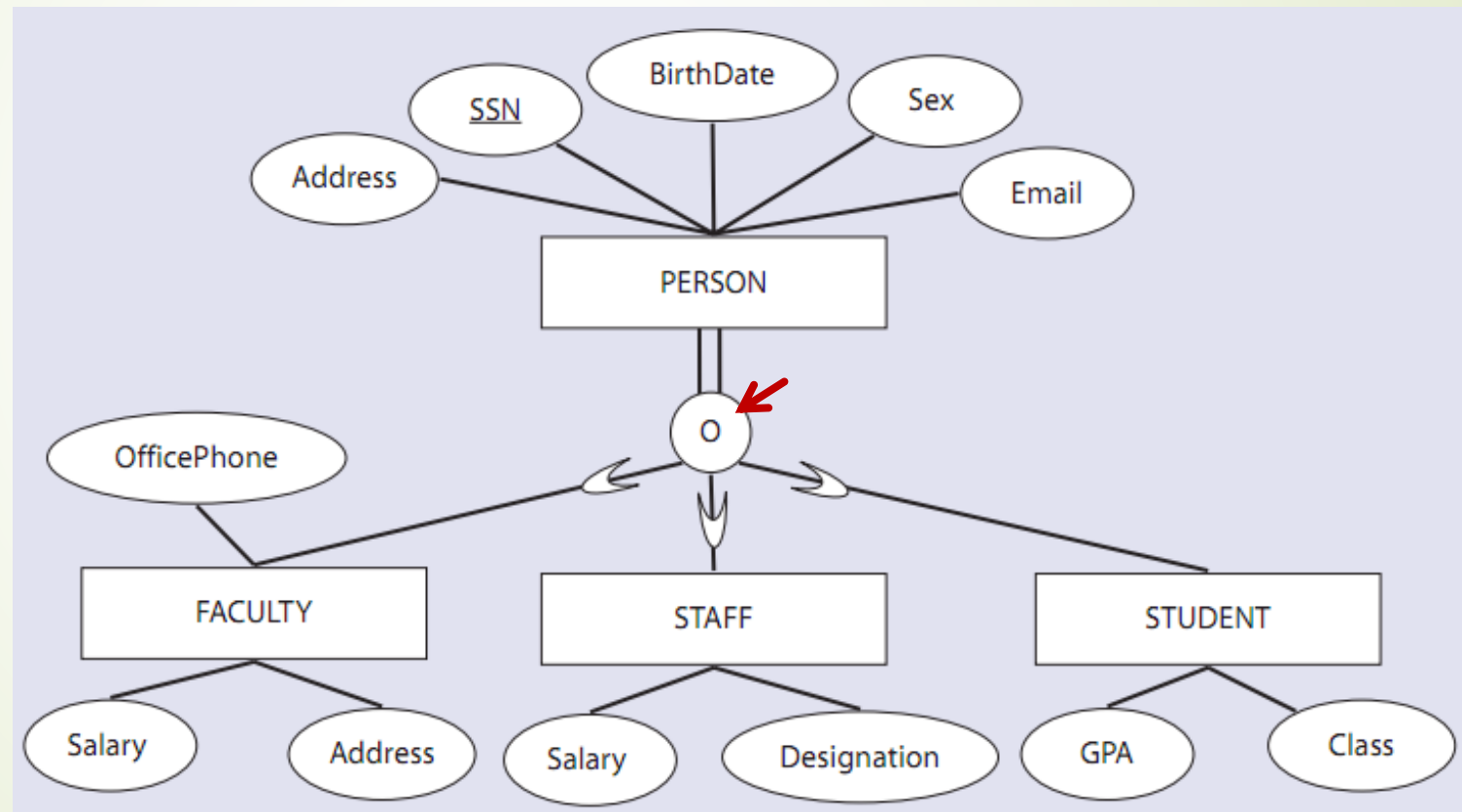


Ràng buộc riêng biệt (Disjoint constraints)

- **Quy tắc chồng lấp (*The overlap rule*):** nếu một thể hiện của một lớp cha là một thành viên của lớp con bất kỳ, thì nó có thể là một thành viên của nhiều hơn một lớp con.
- Trong mỗi quan hệ superclass/subclass, overlap constraint được ký hiệu là O

Ràng buộc riêng biệt (Disjoint constraints)

► Ví dụ:



Union Types Using Categories

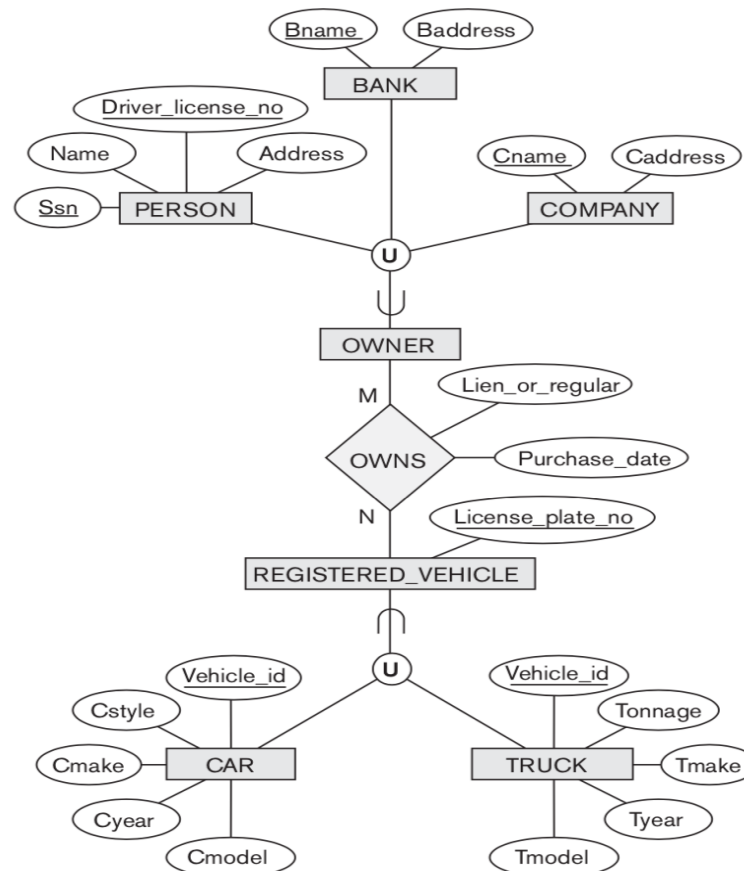


Figure 8.8
Two categories (union types): OWNER and REGISTERED_VEHICLE.