

HỆ THỐNG THÔNG TIN QUẢN LÝ

Cấu trúc dữ liệu theo mô hình thực thể - liên kết ER

Mô hình thực thể - liên kết (Entity Relationship Model – ER Model)

- ⇒ Mô hình dữ liệu thực thể kết hợp (E-R - entity-relationship data model) do Peter Pin_Shan Chen đề xuất năm 1976, nhìn thế giới thực như là một tập các đối tượng cẩn bản được gọi là các *thực thể*, và *các mối quan hệ* ở giữa các đối tượng này.
- ⇒ Mô hình ER được dùng để xây dựng mô hình dữ liệu ý niệm (Conceptual data modeling) nhằm biểu diễn cấu trúc và các ràng buộc của CSDL.
- ⇒ Mô hình ER như một công cụ để trao đổi ý tưởng giữa nhà thiết kế và người dùng cuối trong giai đoạn phân tích. Nó độc lập với DBMS và quá trình thi công database.

Mô hình thực thể - liên kết (Entity Relationship Model – ER Model)

⇒ Mục đích của mô hình E – R:

- Làm thống nhất quan điểm về dữ liệu của những người tham gia hệ thống gọi là **quy tắc nghiệp vụ** (business rule) : Người quản lý, người dùng cuối, người thiết kế hệ thống
- Xác định các xử lý về dữ liệu cũng như các ràng buộc (constraint) trên các dữ liệu.
- Giúp đỡ việc thể hiện cơ sở dữ liệu về mặt cấu trúc: Sử dụng thực thể và các mối liên kết giữa các thực thể. Biểu diễn mô hình quan hệ thực thể bằng một sơ đồ.

Quá trình thiết kế mô hình dữ liệu ý niệm

- ⇒ **Bước 1:** Nhận dạng các kiểu thực thể
- ⇒ **Bước 2:** Nhận dạng các kiểu liên kết giữa các thực thể
- ⇒ **Bước 3:** Nhận dạng các thuộc tính của các kiểu thực thể và các mối liên kết
- ⇒ **Bước 4:** Nhận dạng thuộc tính xác định cho mỗi kiểu thực thể
- ⇒ **Bước 5:** Nhận dạng các cấu trúc siêu kiểu/ kiểu con
- ⇒ **Bước 6:** Vẽ sơ đồ ER

Sơ đồ thực thể - liên kết

- ⇒ Mô hình ER được diễn tả bằng sơ đồ liên kết thực thể (entity relationship diagram - ERD)
- ⇒ Ba phần tử cơ bản:
 - Kiểu thực thể (entity Type)
 - Quan hệ (Relationship)
 - Các thuộc tính (Attribute)

Thực thể - Entity

- ⇒ Một *thực thể* (an entity) là một “sự vật” hoặc “đối tượng” mà nó tồn tại và có thể phân biệt được với các đối tượng khác. Ví dụ như một nhân viên trong một tổ chức là một thực thể.
- ⇒ Thực thể có thể là
 - Một người như nhân viên, sinh viên,..
 - Một nơi chốn như thành phố, đất nước,..
 - Một sự kiện như mua hàng, trả lương,..
 - Một khái niệm như môn học, tài khoản,...

Kiểu thực thể - Entity Type

⇒ Một *kiểu thực thể hay tập thực thể* (an entity set) là một tập hợp các thực thể có cùng những tính chất hoặc thuộc tính.

- Ví dụ như tập hợp tất cả những người mà họ là nhân viên của một tổ chức là một tập thực thể *nhanvien*.

⇒ Mỗi tập thực thể được đặt một tên gọi, thông thường là danh từ số ít và viết chữ in hoa.

- Ví dụ như KHACHHANG, HOADON,...

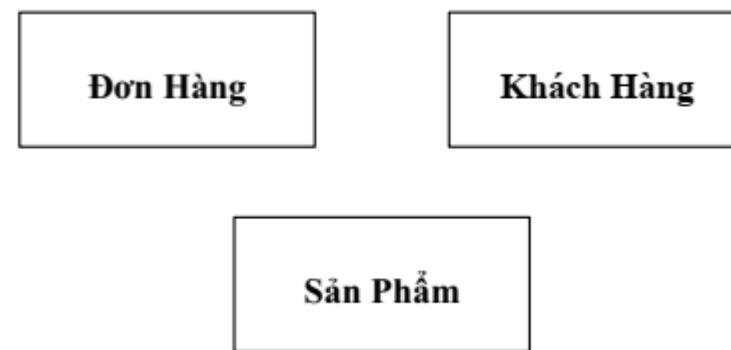
⇒ Biểu diễn: bằng hình chữ nhật.

KHACHHANG

SANPHAM

Kiểu thực thể - Entity Type

- ⇒ **Thể hiện (instance)** của một kiểu thực thể là một trường hợp cụ thể của kiểu thực thể đó.
- ⇒ Ví dụ: kiểu thực thể KhachHang có các điển hình là Lan và Minh. Mỗi KhachHang đều có mã khách khác nhau, và có thể thực hiện các dịch vụ như đặt hàng, thanh toán tiền



Cách đặt tên và ký hiệu

- ⇒ Mỗi kiểu thực thể phải có một tên gọi, nên là danh từ số ít và viết chữ hoa.
- ⇒ Ký hiệu của các kiểu thực thể

EMPLOYEE

Thực thể mạnh

DEPENDENT

Thực thể yếu

Các kiểu thực thể

- ⇒ **Kiểu thực thể mạnh** (strong entity type): tồn tại độc lập với những kiểu thực thể khác và có khóa chính
- ⇒ **Kiểu thực thể yếu** (weak entity type): tồn tại phụ thuộc vào kiểu thực thể khác, có thể không có đủ các thuộc tính để cấu thành một khóa chính

EMPLOYEE

DEPENDENT

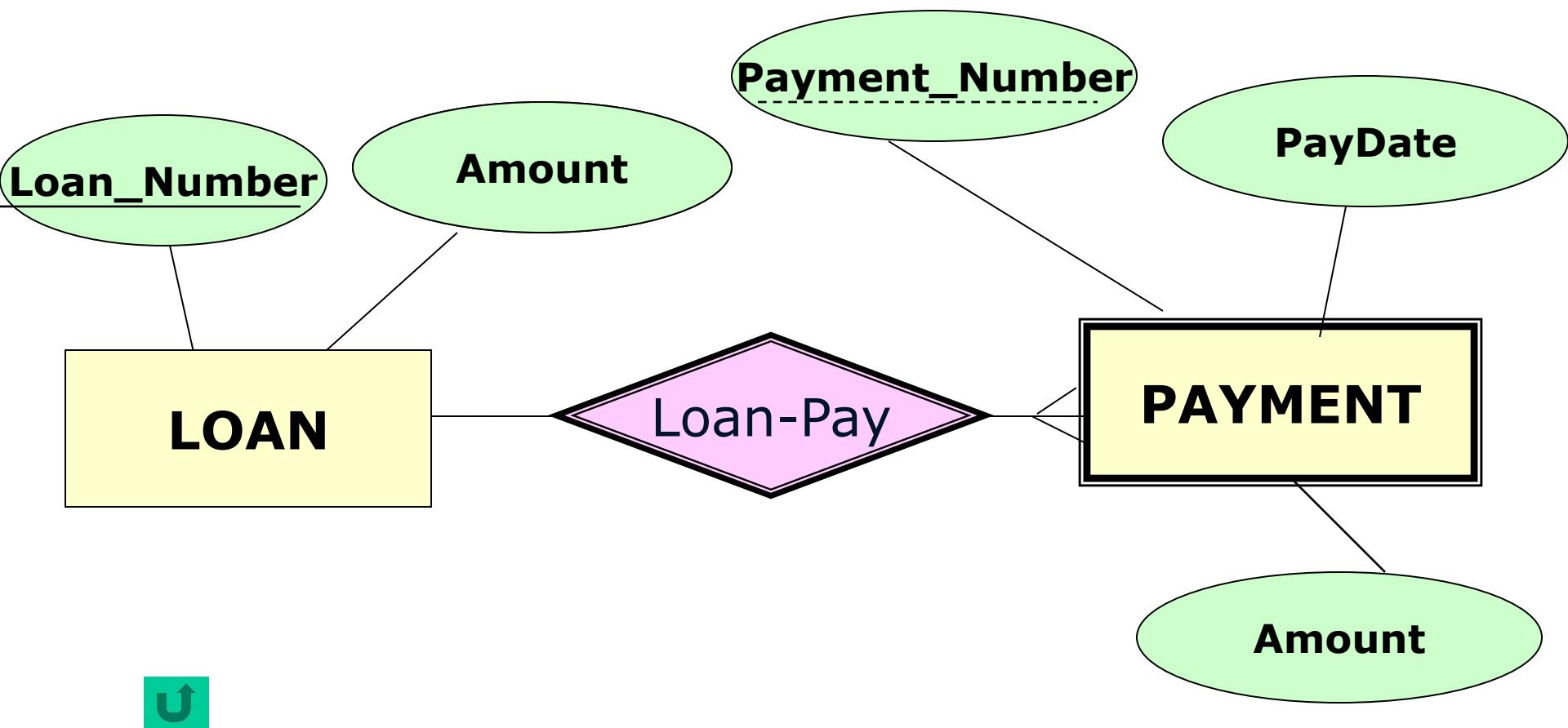
Thực thể mạnh

Thực thể yếu

- ⇒ Ví dụ:

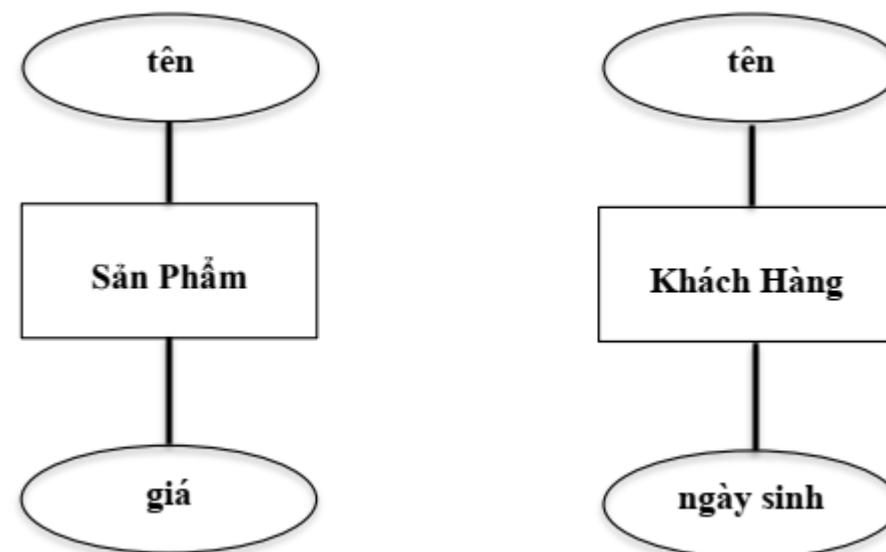
- LOAN (Mượn) là kiểu thực thể mạnh.
- PAYMENT (Trả) là kiểu thực thể yếu, lệ thuộc vào LOAN.
- Xác định thực thể mạnh và yếu cho 2 thực thể BenhNhan và ThanNhan

Ví dụ thực thể mạnh/yếu



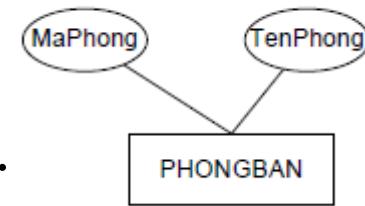
Thuộc tính - attribute

Một *thuộc tính* là một tính chất của thực thể, một đặc tính xác định phân biệt mà tất cả các thể hiện của thực thể có thể có. Ví dụ, thuộc tính Tên của thực thể **Khách Hàng**, thuộc tính Giá của thực thể **Sản Phẩm**



Thuộc tính - attribute

- ⇒ Mỗi tập thực thể có 1 số thuộc tính.
- ⇒ Thuộc tính là các đặc trưng (*properties*) được sử dụng để biểu diễn thực thể hay 1 mối liên kết.
- ⇒ Được biểu diễn bằng *hình OVAL*.
- ⇒ Các loại thuộc tính
 - Thuộc tính bắt buộc và thuộc tính tùy chọn.
 - Thuộc tính đơn và thuộc tính phức hợp.
 - Thuộc tính đơn trị và thuộc tính đa trị.
 - Thuộc tính chúa và thuộc tính dẫn xuất.
 - Thuộc tính khóa và thuộc tính không khóa.
 - Thuộc tính rỗng



Các kiểu thuộc tính

⇒ Thuộc tính đơn (simple attribute): là thuộc tính không thể phân nhỏ được.

Ví dụ: Color, Weight, HorsePower, Masv, CMND, Phai

⇒ Thuộc tính phức hợp (composite attribute): là thuộc tính được kết hợp của một số thành phần hay có thể chia thành các thuộc tính khác

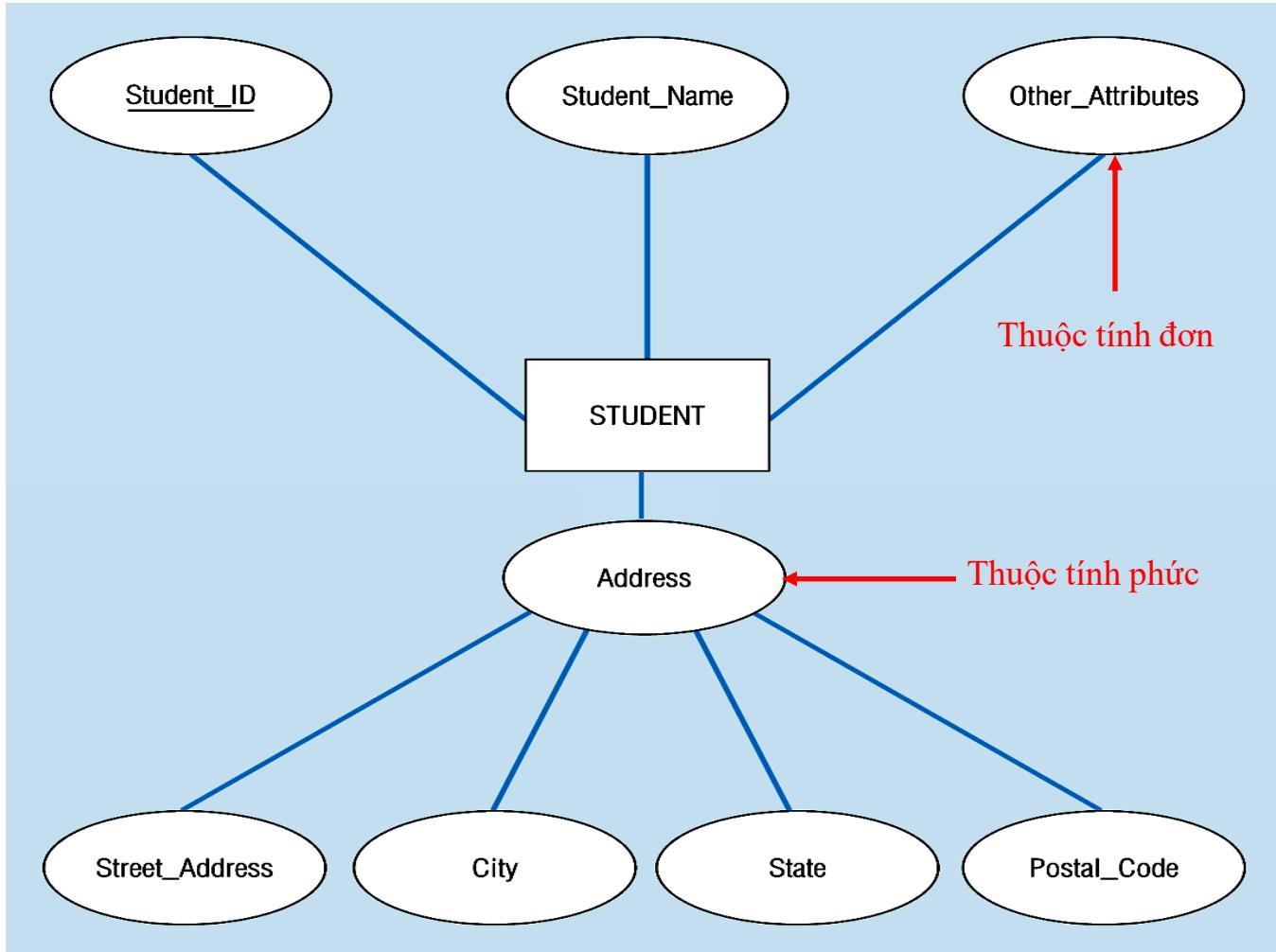
Ví dụ: Thuộc tính Address bao gồm các thành phần Street, District, City

▪ Thuộc tính chứa (stored attribute): là thuộc tính mà giá trị của nó không được suy dẫn từ các thuộc tính khác.

⇒ Thuộc tính dẫn xuất (derived attribute): là thuộc tính mà trị của nó có thể tính ra được từ các thuộc tính khác

Ví dụ: Year_Employed là thuộc tính dẫn xuất từ thuộc tính Date_Employed

Các kiểu thuộc tính

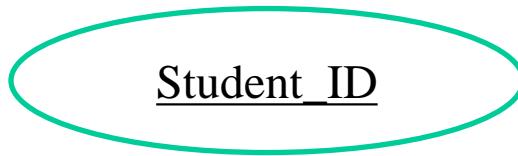


Các kiểu thuộc tính (tt)

- ⇒ **Thuộc tính đơn trị** (single valued attribute): có 1 giá trị duy nhất tại một thời điểm.
- ⇒ **Thuộc tính đa trị** (multivalued attribute): có nhiều giá trị tại một thời điểm.

Ví dụ: Thực thể COURSE có thuộc tính Teacher đa trị, một môn học có thể được dạy bởi nhiều hơn 1 thầy cô.

- ⇒ **Thuộc tính rỗng**. Một giá rỗng (null value) được dùng đến khi một thực thể không có giá trị đối với một thuộc tính. Ví dụ một nhân viên nào đó không có người trong gia đình thì giá trị của thuộc tính *tên người trong gia đình* đối với nhân viên đó phải là rỗng.

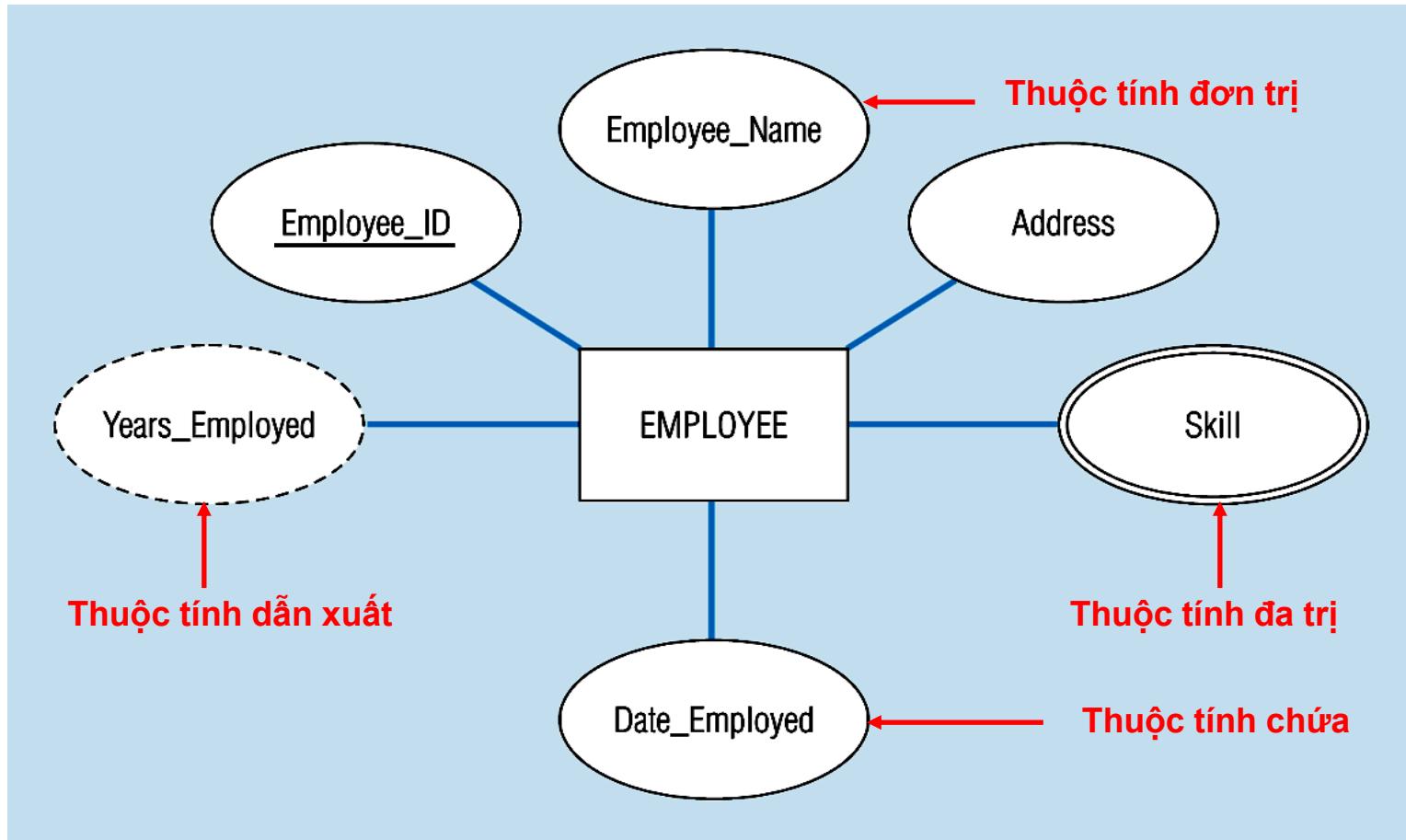


Thuộc tính xác định



Thuộc tính đa trị

Các kiểu thuộc tính (tt)



Các kiểu thuộc tính

⇒ Khóa / thuộc tính xác định (*key / identifier*)

- *Khóa* là một thuộc tính hoặc tổ hợp các thuộc tính dùng để xác định duy nhất một thể hiện của một kiểu thực thể.

⇒ Thuộc tính khóa và thuộc tính không khóa

- *Thuộc tính khóa (key attribute)* là thuộc tính ở trong khóa.
 - Thuộc tính khóa *được gạch dưới*.
- *Thuộc tính không khóa (non-key attribute)* là thuộc tính không ở trong khóa.
 - Thuộc tính không khóa còn được gọi là *thuộc tính mô tả (descriptor)*.



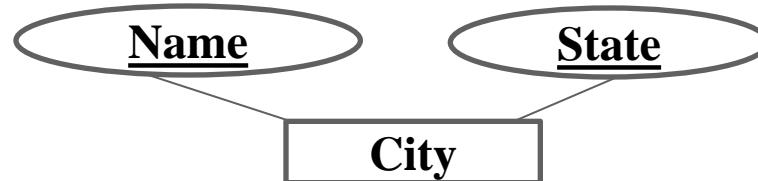
Các kiểu thuộc tính

⇒ Khóa đơn và khóa phức hợp

- *Khóa đơn (simple key)* là khóa chỉ có một thuộc tính.
- *Khóa phức hợp (composite key)* là khóa có nhiều hơn một thuộc tính.

⇒ Khóa dự tuyển (*candidate key*)

- *Khóa dự tuyển* là khóa của một tập thực thể.
- Một tập thực thể có ít nhất một khóa dự tuyển.

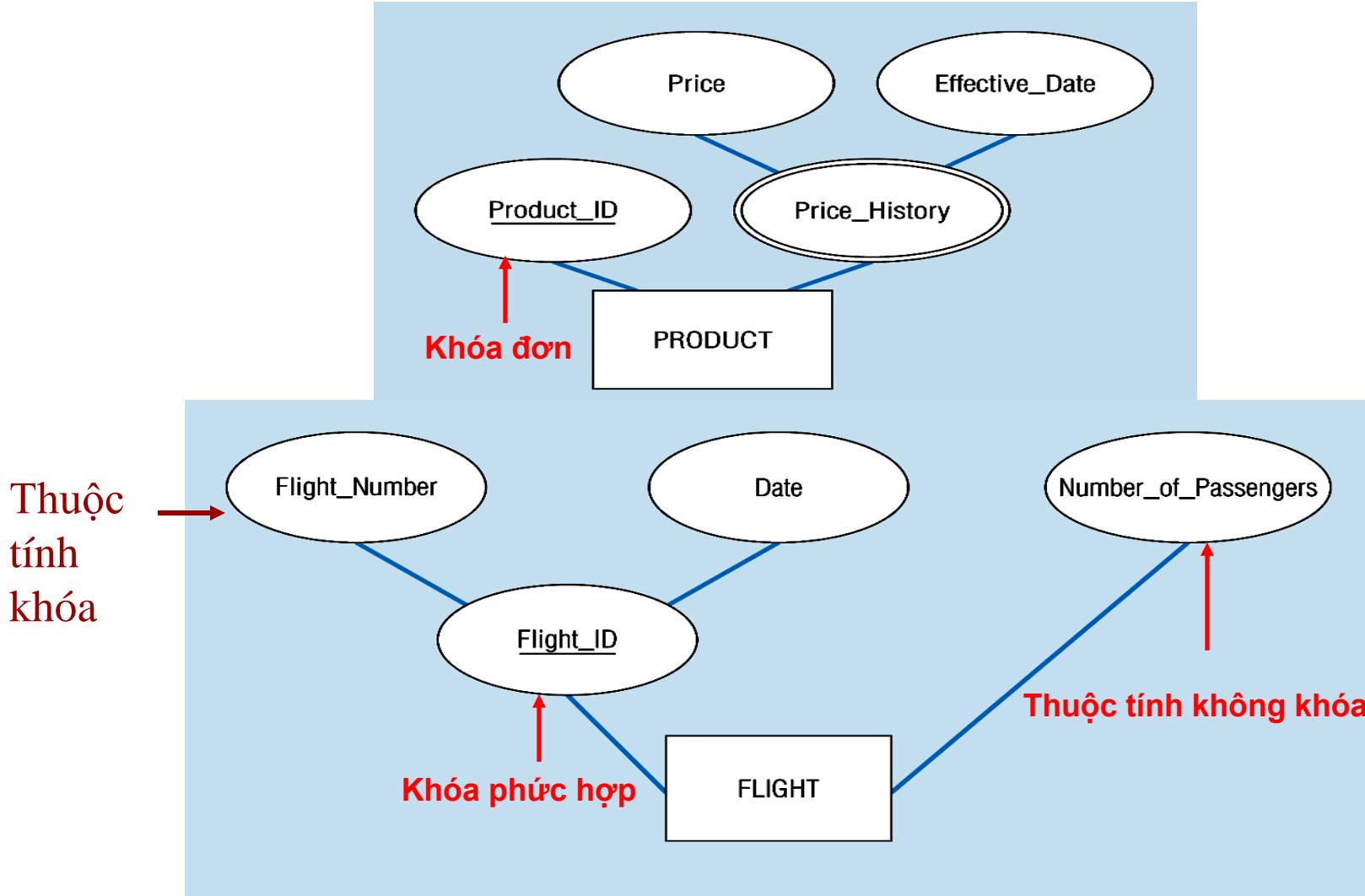


Các kiểu thuộc tính

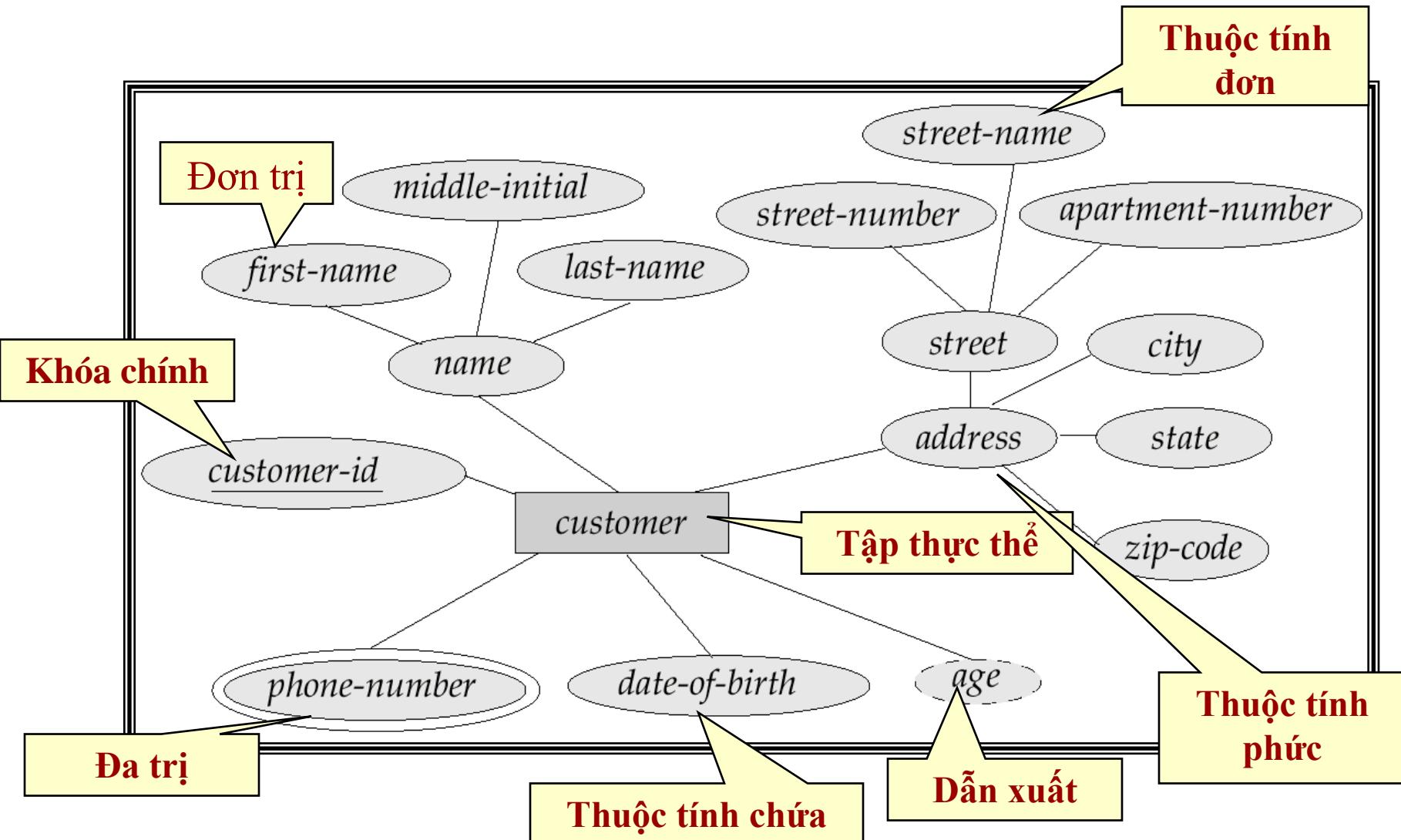
⇒ Khóa chính (*primary key*)

- *Khóa chính* là một khóa tiêu biểu trong các khóa dự tuyển của một kiểu thực thể.
- Một kiểu thực thể chỉ có một khóa chính.
- Khóa chính dùng để liên kết giữa các thực thể.

Các kiểu thuộc tính



Các kiểu thuộc tính



Bài tập

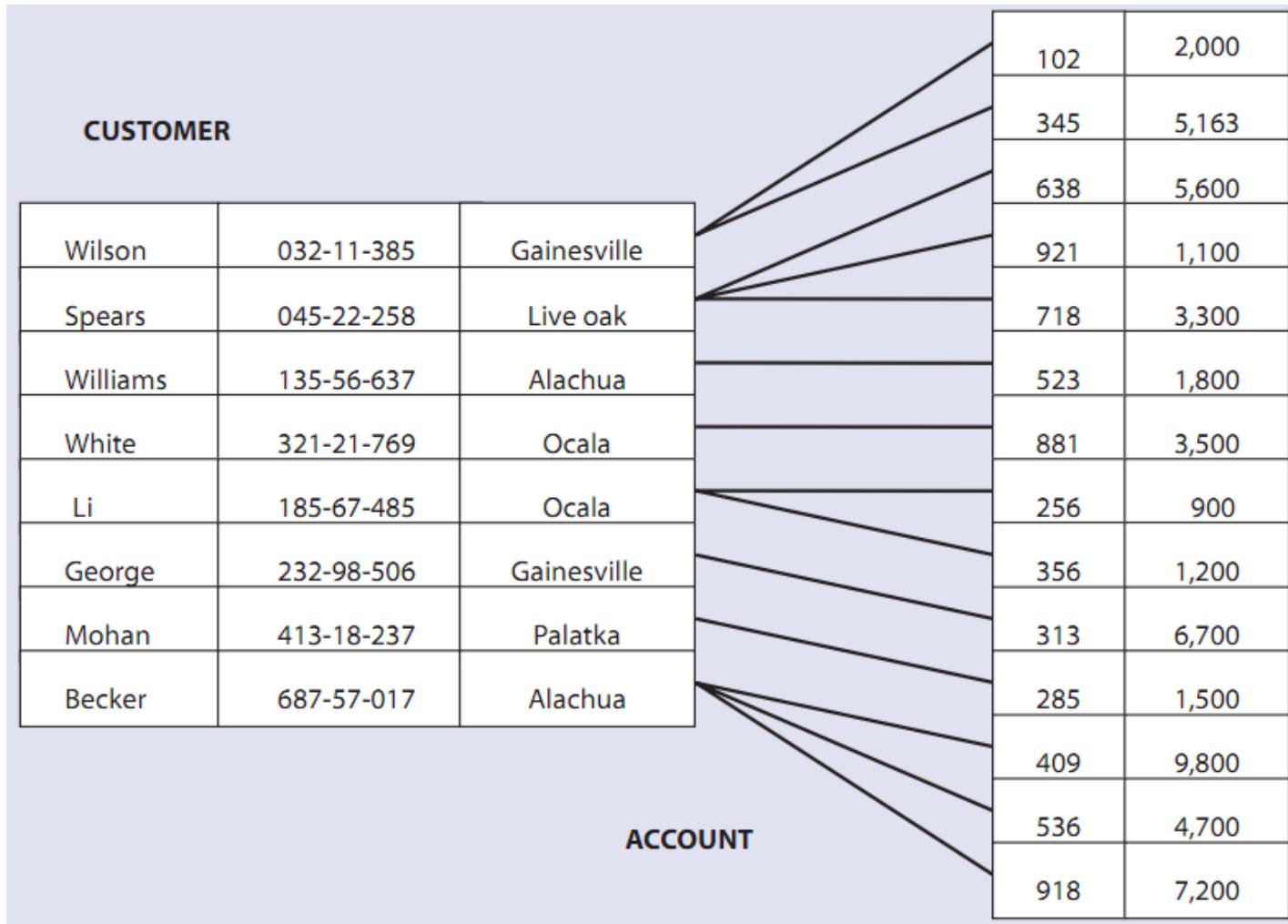
Hệ thống quản lý nhân viên

- Để quản lý việc phân công các nhân viên tham gia vào xây dựng các công trình. Công ty xây dựng X tổ chức quản lý như sau:
- Cùng lúc công ty có thể tham gia xây dựng nhiều công trình, mỗi công trình có một mã số công trình duy nhất (MACT), mỗi mã số công trình xác định các thông tin như: Tên gọi công trình (TENCT), địa điểm(ĐIADIEM), ngày công trình được cấp giấy phép xây dựng (NGAYCAPGP), ngày khởi công (NGAYKC), ngày hoàn thành (NGAYHT)
- Mỗi nhân viên của công ty ABC có một mã số nhân viên(MANV) duy nhất, một mã số nhân viên xác định các thông tin như: Họ tên (HOTEN), ngày sinh(NGAYSINH), phái (PHAI), địa chỉ (ĐIACHI), phòng ban, ...
- Công ty phân công các nhân viên tham gia vào các công trình, mỗi công trình có thể được phân cho nhiều nhân viên và mỗi nhân viên cùng lúc cũng có thể tham gia vào nhiều công trình. Với mỗi công trình một nhân viên có một số lượng ngày công (SLNGAYCONG) đã tham gia vào công trình đó.
- Công ty có nhiều phòng ban(Phòng kế toán, phòng kinh doanh, phòng kỹ thuật, phòng tổ chức, phòng chuyên môn, Phòng phục vụ,...). Mỗi phòng ban có một mã số phòng ban(MAPB) duy nhất, một phòng ban ứng với một tên phòng ban(TENPB)
- Xác định thực thể và thuộc tính mỗi thực thể

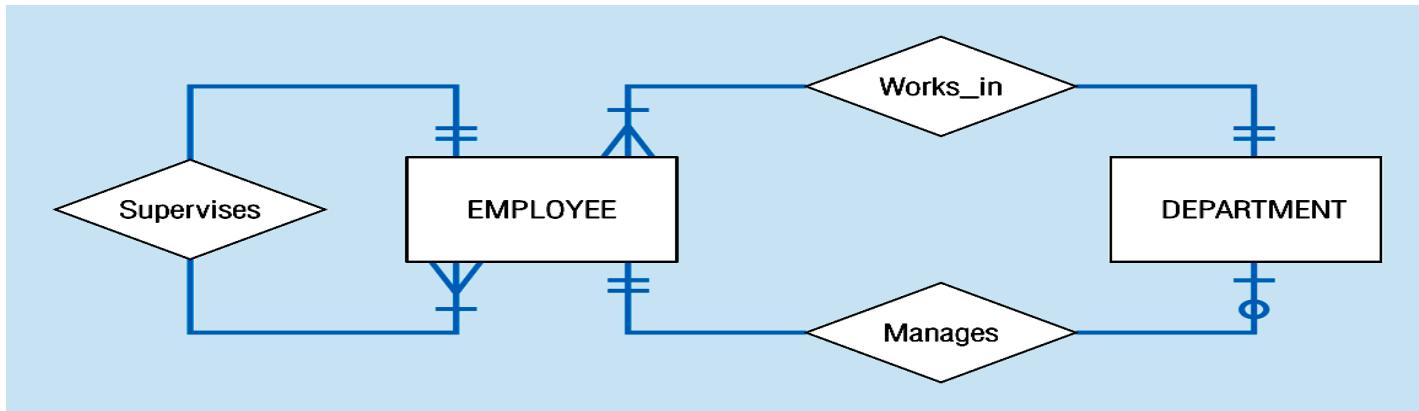
Mối liên kết - Relationship

- ⇒ **Mối liên kết (relationship)** diễn tả sự kết hợp giữa một hay nhiều kiểu thực thể với nhau, là sự kết hợp biểu diễn sự tương tác giữa các điển hình (instance) của một hay nhiều kiểu thực thể (entity type)
- ⇒ **Kiểu liên kết (relationship type)** là một sự kết hợp có ý nghĩa giữa các kiểu thực thể. Được biểu diễn bằng hình thoi.
- ⇒ **Một điển hình liên kết (relationship instance)** là một sự kết hợp giữa các điển hình thực thể nơi mà mỗi điển hình liên kết bao gồm chính xác 1 thực thể từ mỗi kiểu thực thể tham gia vào.

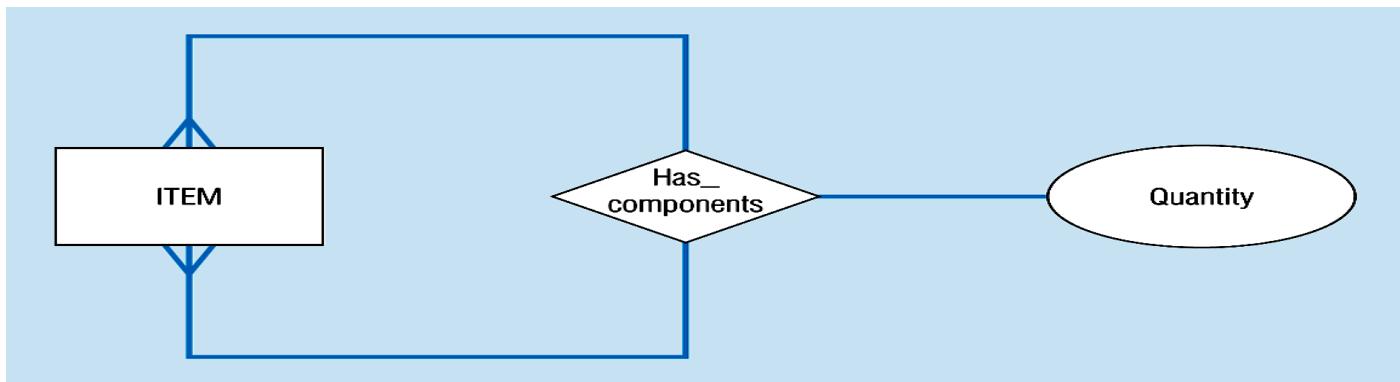
Mối liên kết - Relationship



Mối liên kết - Relationship

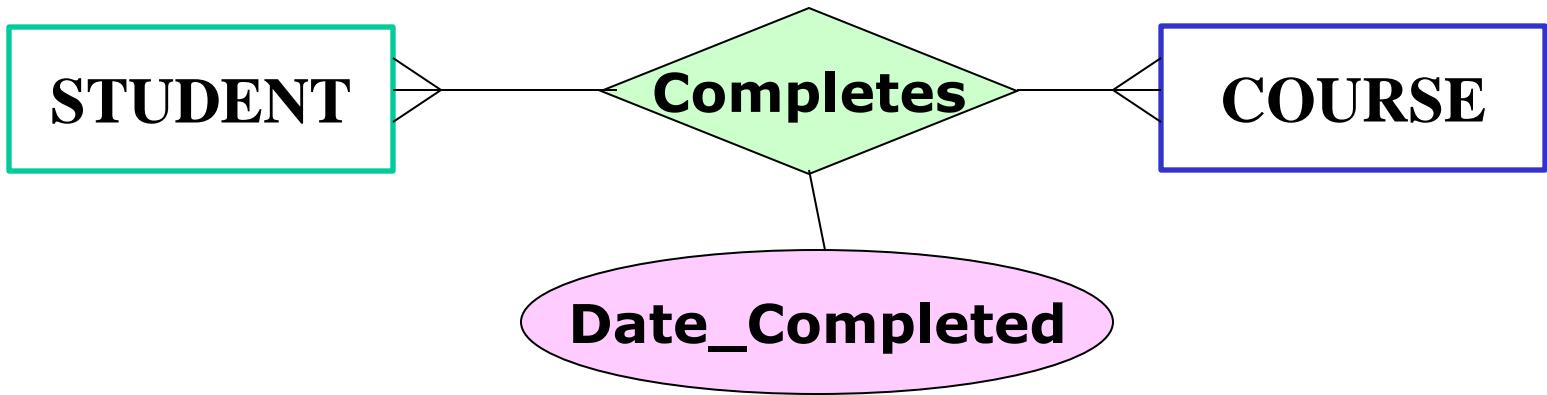


Hai thực thể có nhiều mối liên kết



Mối liên kết có thuộc tính

Ví dụ



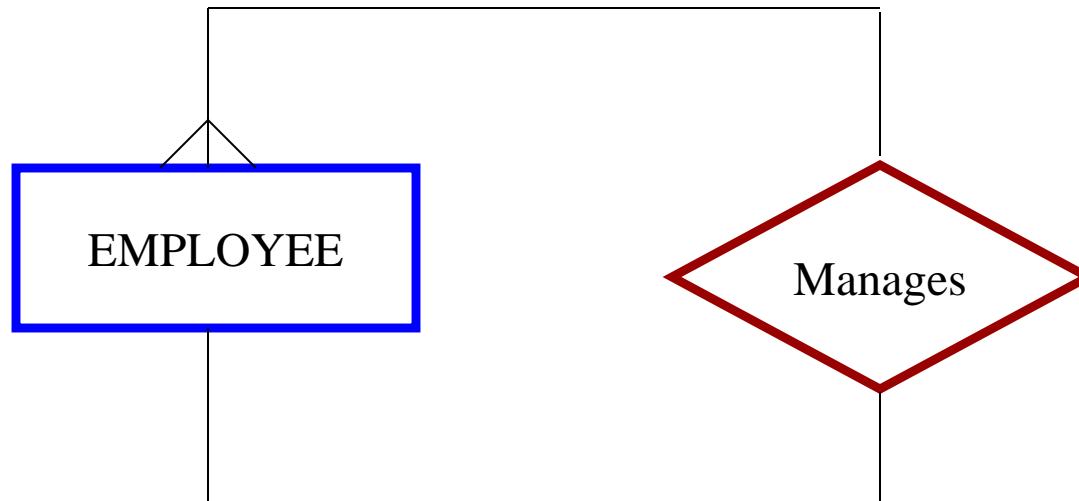
- Thuộc tính Date _ Completed nên đặt ở đâu trong lược đồ trên?
→ Là 1 thuộc tính của mối liên kết **Completed** (thích hợp hơn là thuộc tính của 2 thực thể STUDENT và COURSE)

Bậc và các kiểu liên kết

- ⇒ Độ của mối liên kết (*degree / arity of relationship*): là số lượng kiểu thực thể tham gia vào mối liên kết
- ⇒ Các kiểu liên kết
 - Liên kết 1 ngôi (*unary relationship*)
 - Liên kết 2 ngôi (*binary relationship*)
 - Liên kết 3 ngôi (*ternary relationship*): 3 kiểu thực thể đồng thời tham gia vào mối liên kết.

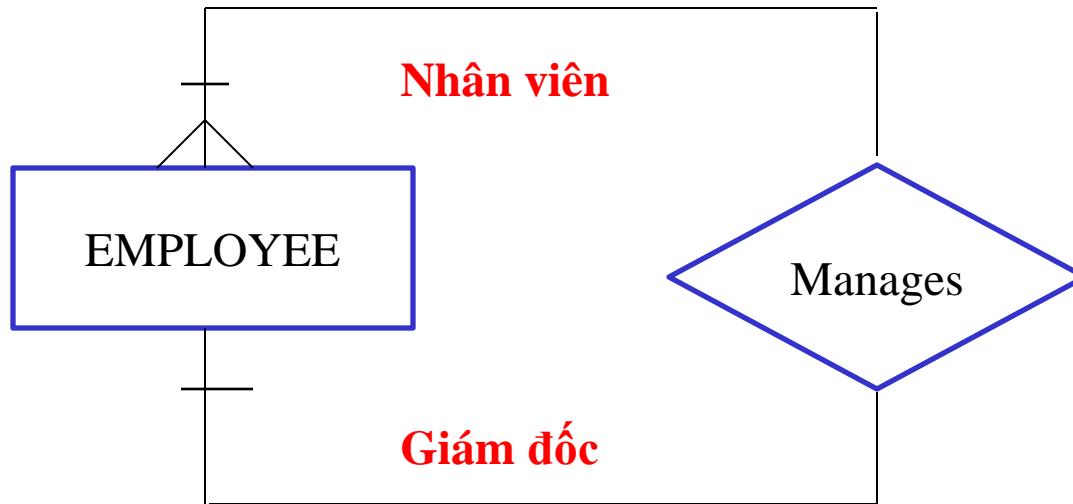
Liên kết một ngôi - Unary relationship

- ⇒ Là mối quan hệ giữa cùng 1 kiểu thực thể.
- ⇒ Còn gọi là mối liên kết đệ quy (recursive relationship)
- ⇒ Ví dụ:

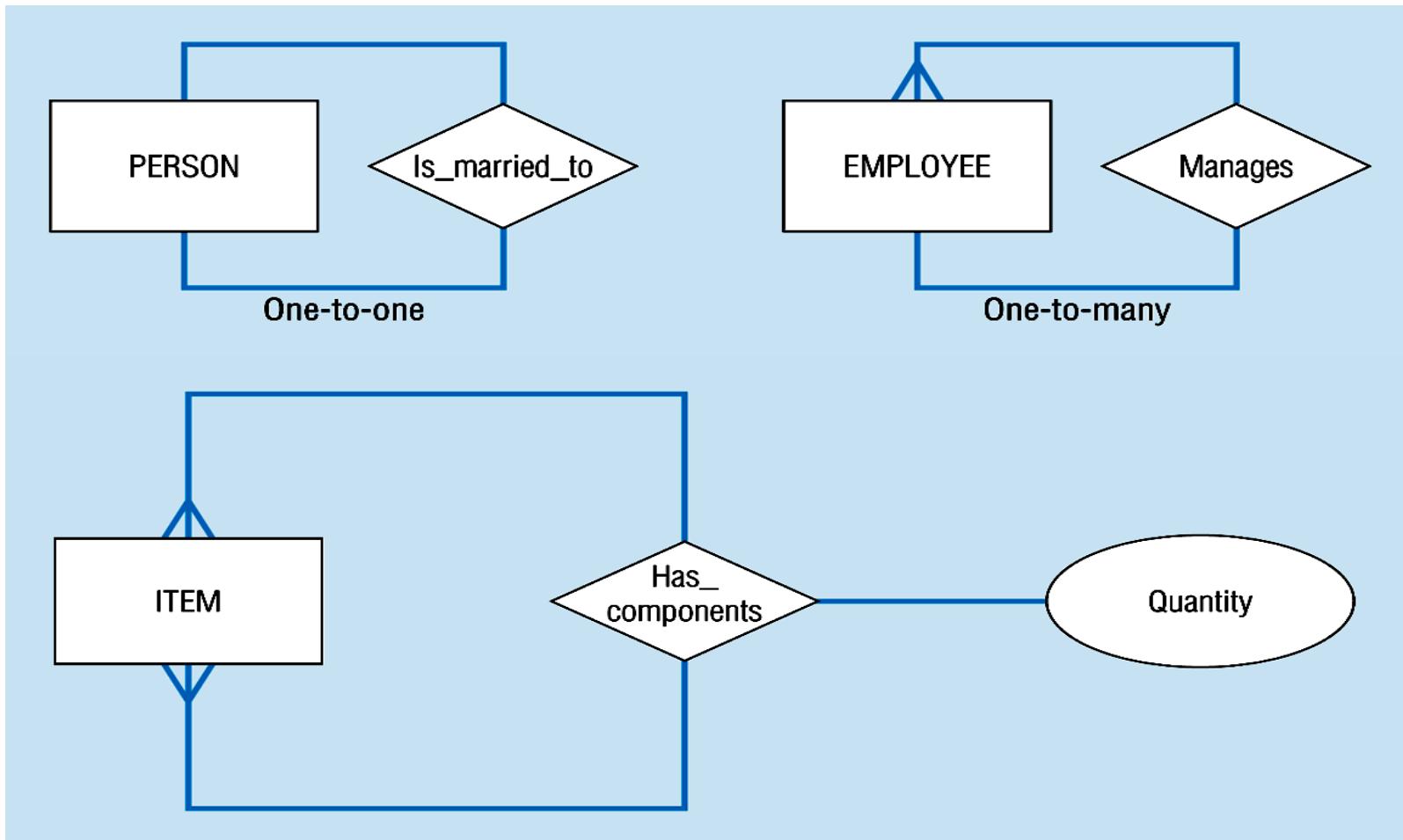


Liên kết một ngôi - Unary relationship

- ⇒ Đôi khi một thực thể xuất hiện nhiều hơn 1 lần trong mối quan hệ.
- ⇒ Để phân biệt, nên tạo role (nhãn) trên các cạnh nối giữa mối quan hệ và thực thể.



Liên kết một ngôi - Unary relationship



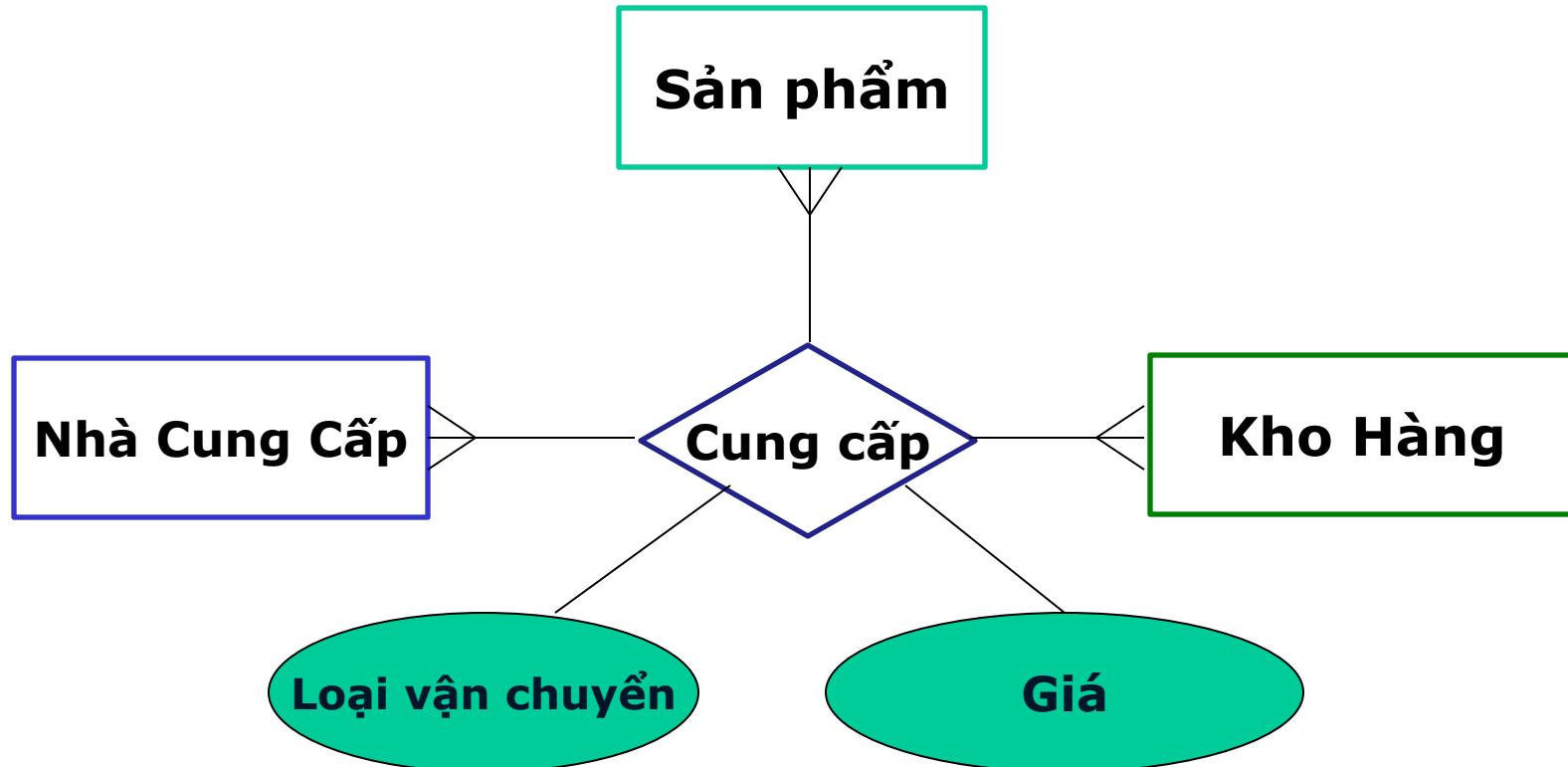
Liên kết hai ngôi - Binary relationship

- ⇒ Là mối liên kết giữa hai kiểu thực thể



Liên kết ba ngôi - Ternary relationship

⇒ Là mối liên kết giữa 3 kiểu thực thể

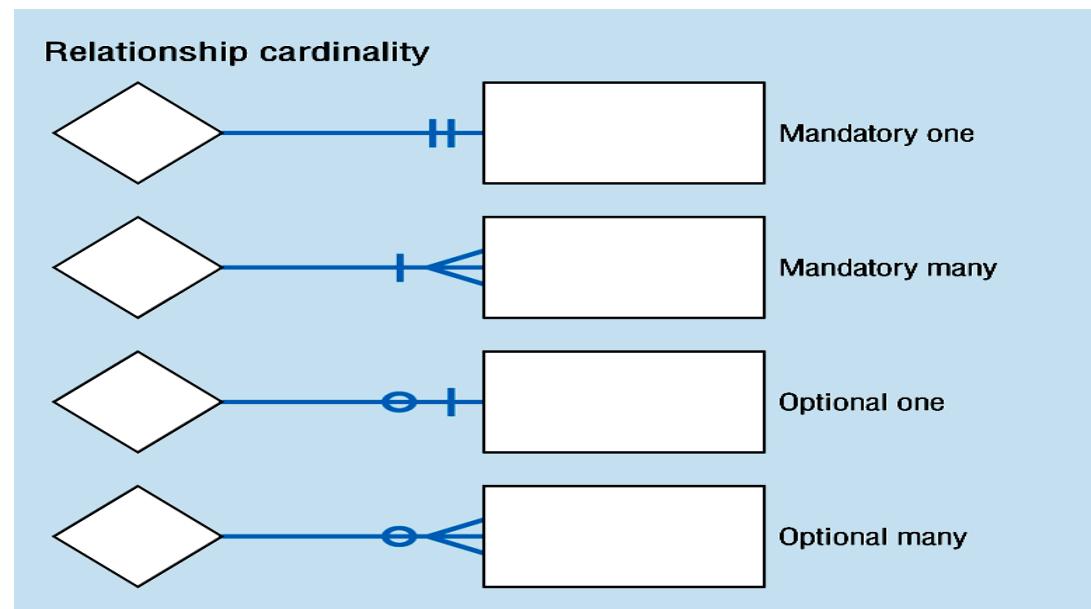


Lượng số của mối liên kết - Cardinality

- ⇒ Lượng số là số thể hiện của kiểu thực thể B mà có thể liên kết với mỗi thể hiện của kiểu thực thể A
- ⇒ Lượng số tối thiểu (minimum cardinality): bằng 0 hoặc 1, là số lần tối thiểu mà một thể hiện bất kỳ của một tập thực thể tham gia vào các thể hiện của mối kết hợp.
- ⇒ Lượng số tối đa (maximum cardinality): bằng 1 hoặc n, là số lần tối đa mà một thể hiện bất kỳ của một tập thực thể tham gia vào các thể hiện của mối kết hợp.

Lượng số của mối liên kết - Cardinality

- ⇒ Nếu lượng số tối thiểu là 0, kiểu thực thể B được gọi là nhiệm ý. Ký hiệu là O
- ⇒ Nếu lượng số tối thiểu và tối đa đều là 1 thì lượng số này được gọi là bắt buộc (mandatory). Ký hiệu là ||
- ⇒ Ba dạng liên kết:
 - Liên kết 1-1
 - Liên kết 1-n
 - Liên kết n-n



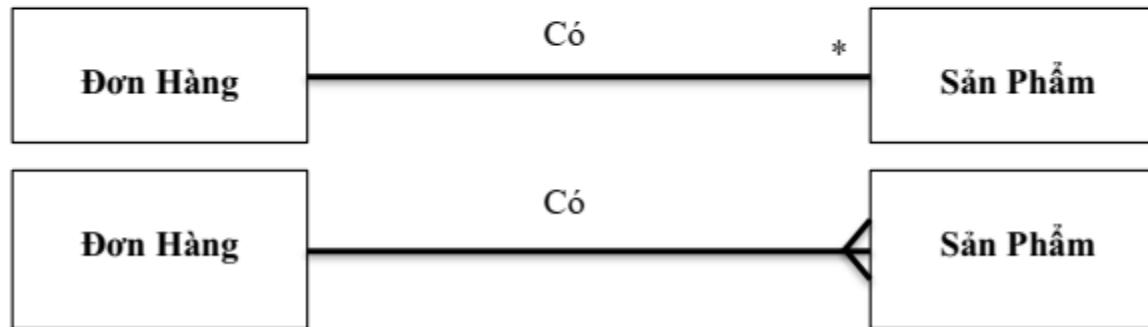
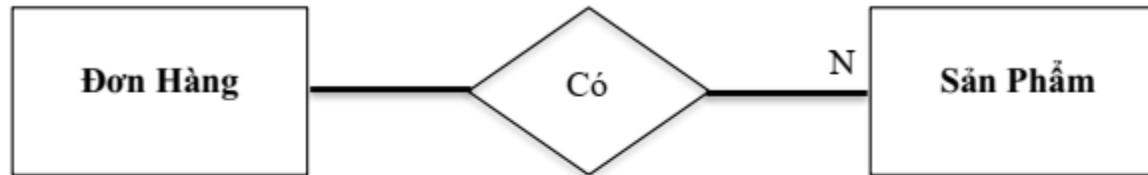
Nhom 1 stop on 09/2/2023

Lượng số của mối liên kết - Cardinality

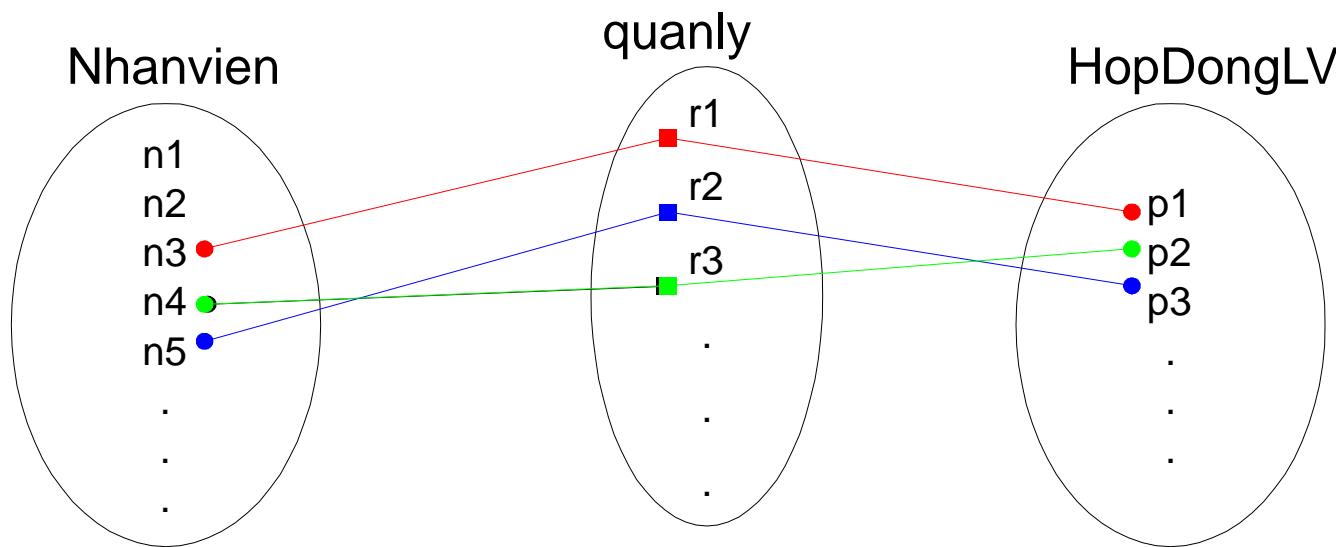
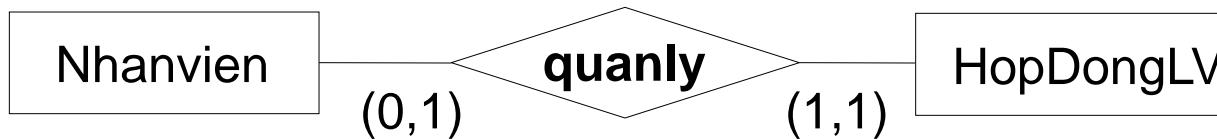
⇒ Các loại lượng số

- **một - một (one-to-one)**: một thể hiện của kiểu thực thể a liên kết với một thể hiện của kiểu thực thể b và ngược lại
- **một - nhiều (one-to-many)**: một thể hiện của kiểu thực thể a liên kết với nhiều thể hiện của kiểu thực thể b ; ngược lại một thể hiện của kiểu thực thể b chỉ liên kết với một thể hiện của kiểu thực thể a .
- **nhiều - nhiều (many-to-many)**: một thể hiện của kiểu thực thể a liên kết với nhiều thể hiện của kiểu thực thể b ; ngược lại một thể hiện của kiểu thực thể b liên kết với nhiều thể hiện của kiểu thực thể a .

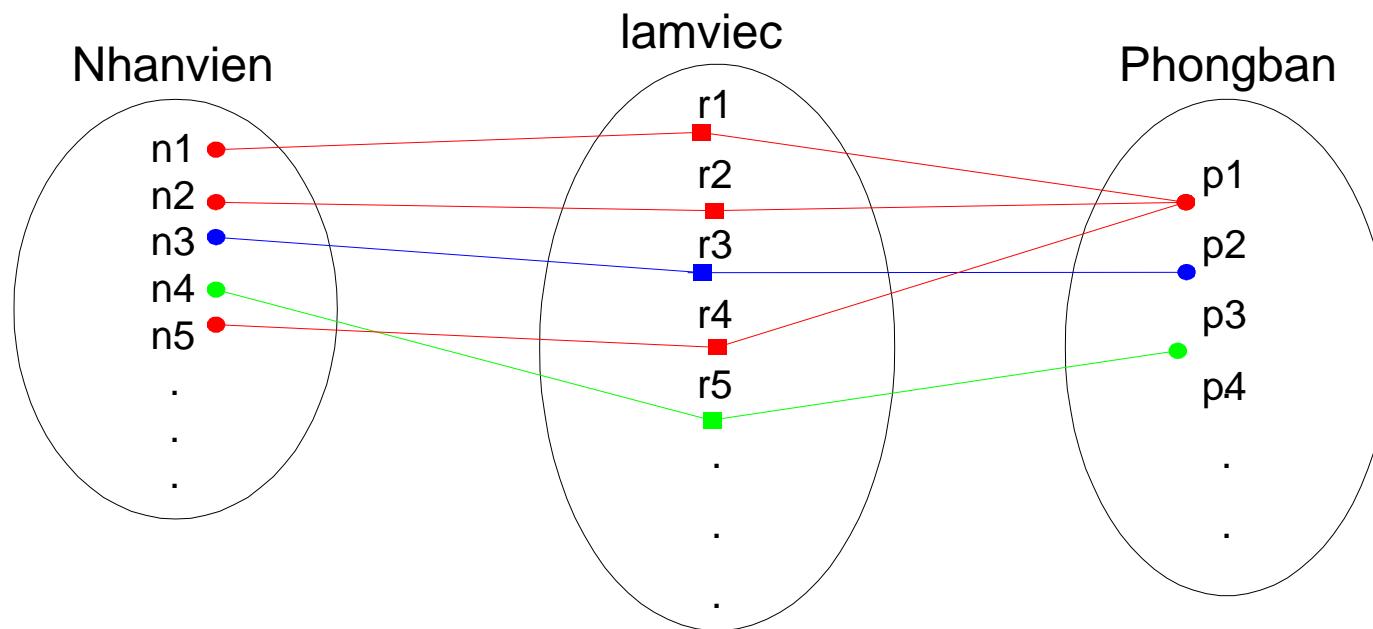
Quan hệ một-nhiều giữa các thực thể



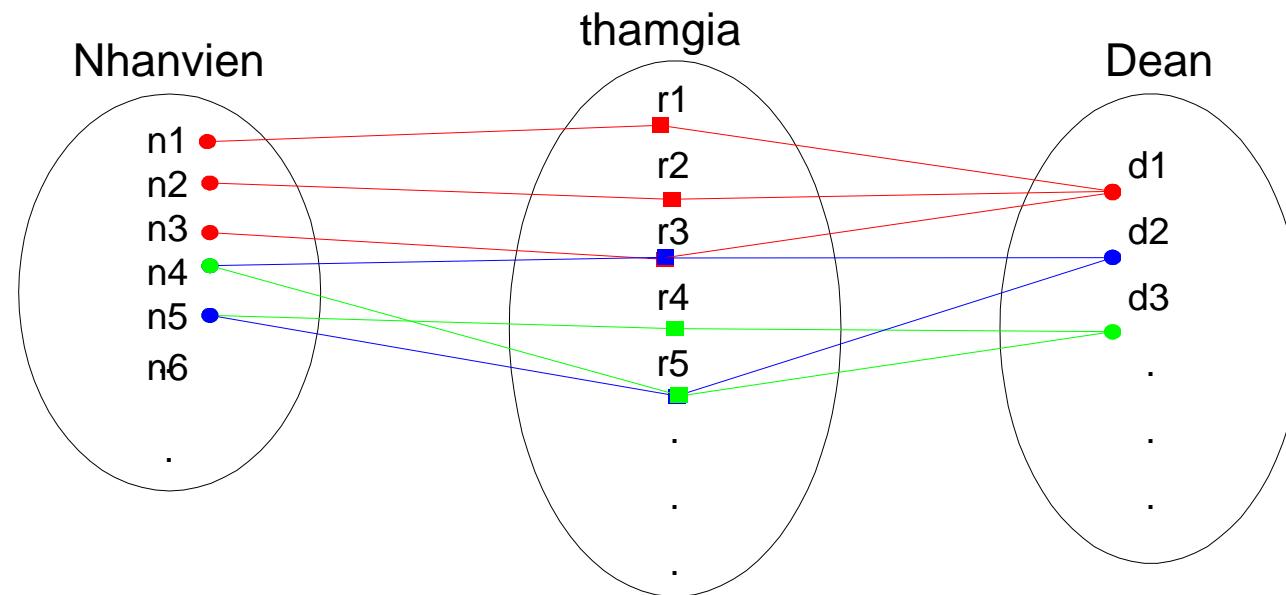
Mối Kết hợp 1-1



Mối Kết hợp 1-n



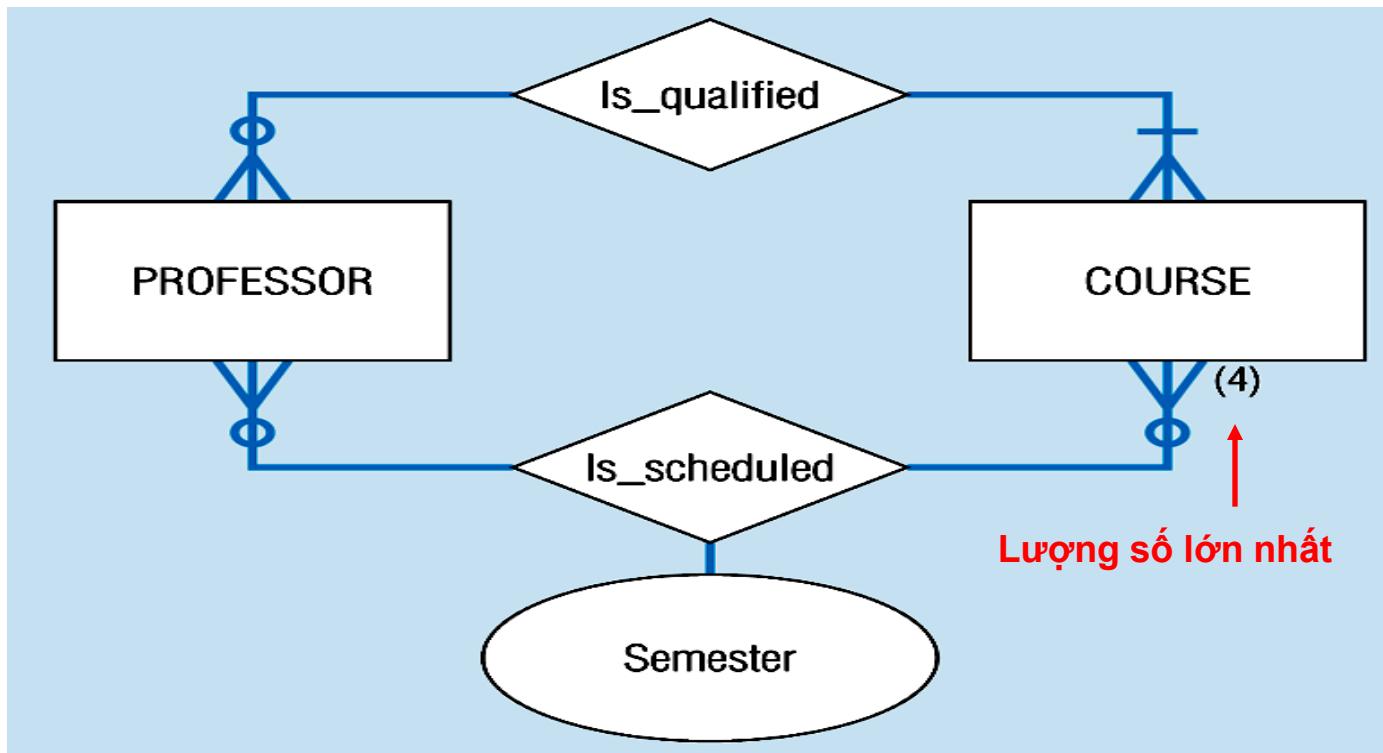
Mối Kết hợp n-n



Ví dụ mối liên kết



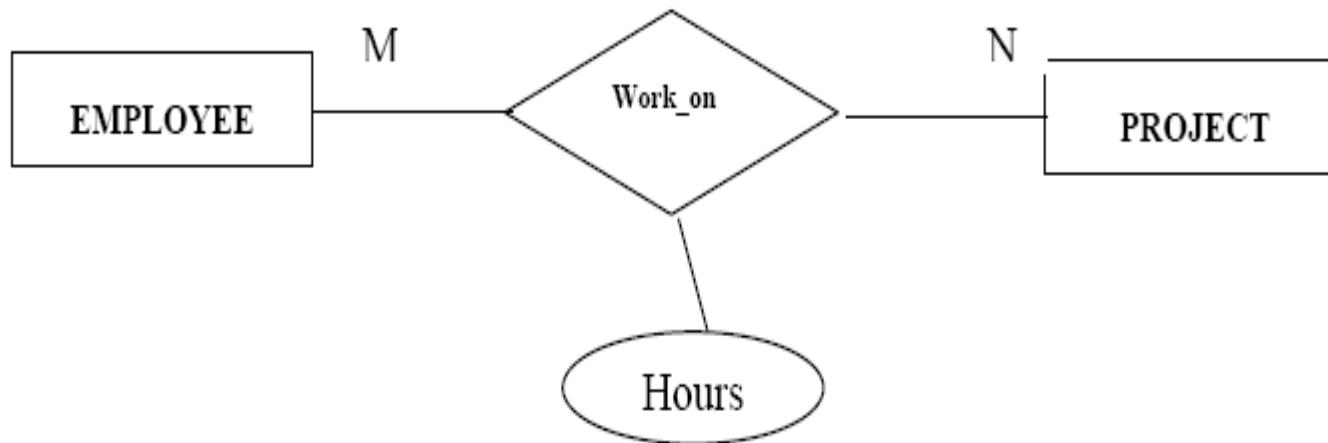
Ví dụ mối liên kết



Mối liên kết có lượng số tối đa xác định

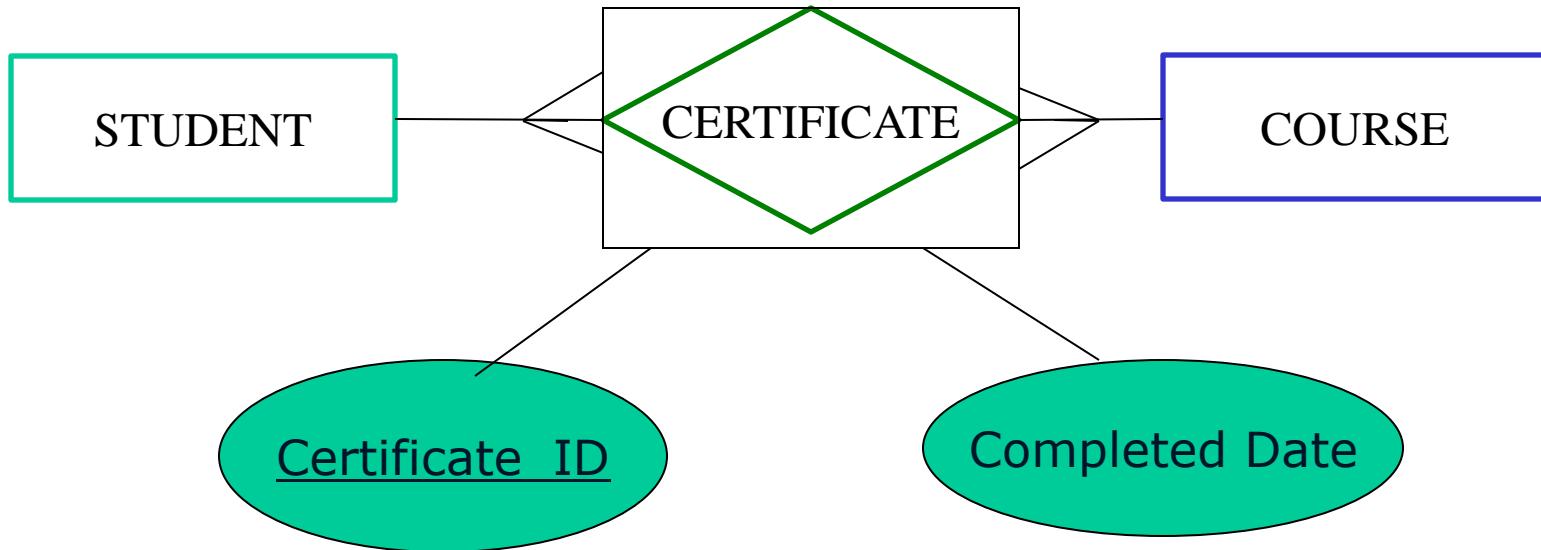
Thuộc tính của kiểu liên kết

- ⇒ Kiểu liên kết cũng có thể có thuộc tính.
- ⇒ Ví dụ: Số giờ nhân viên làm việc cho dự án (Hours) là thuộc tính của mối liên kết giữa hai kiểu thực thể EMPLOYEE và PROJECT.



Kiểu thực thể kết hợp Associative entity type

- ⇒ Là một kiểu thực thể dùng liên kết một hay nhiều kiểu thực thể và có chứa thêm một số thuộc tính riêng biệt của mối liên kết này



Ràng buộc lượng số của liên kết ba ngôi

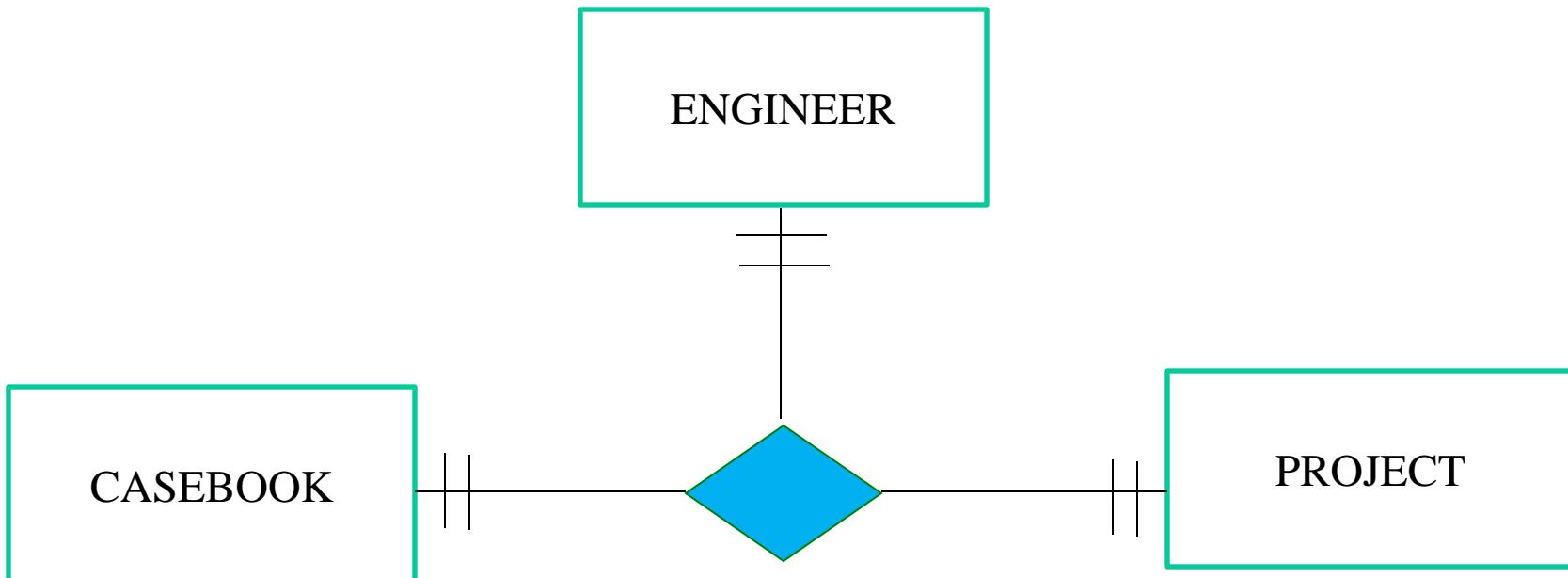
⇒ Trường hợp liên kết 1-1-1

Xét quy tắc nghiệp vụ (business rule) sau:

“Mỗi kỹ sư dùng chỉ 1 số ghi chép cho 1 đề án. Những kỹ sư khác nhau sẽ dùng những số ghi chép khác nhau khi làm việc cho cùng một đề án. Không có kỹ sư nào dùng cùng một số ghi chép cho nhiều đề án khác nhau”

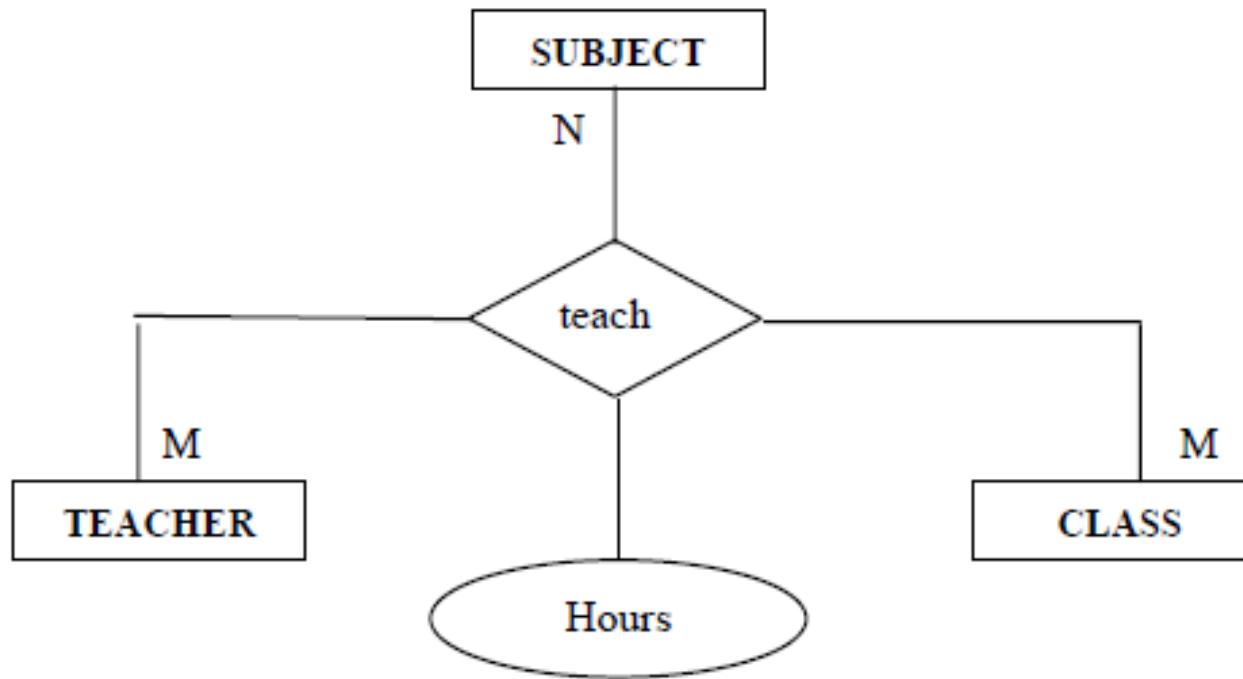
Ràng buộc lượng số của liên kết ba ngôi

⇒ Trường hợp liên kết 1-1-1



Ràng buộc lượng số của liên kết ba ngôi

⇒ Trường hợp liên kết n-n-n



Kỹ thuật thiết kế - Design Techniques

1. Tránh dư thừa (Avoid redundancy)
2. Hạn chế việc sử dụng thực thể yếu
3. Đừng dùng những thực thể mà chỉ có mỗi một thuộc tính

Tránh dư thừa

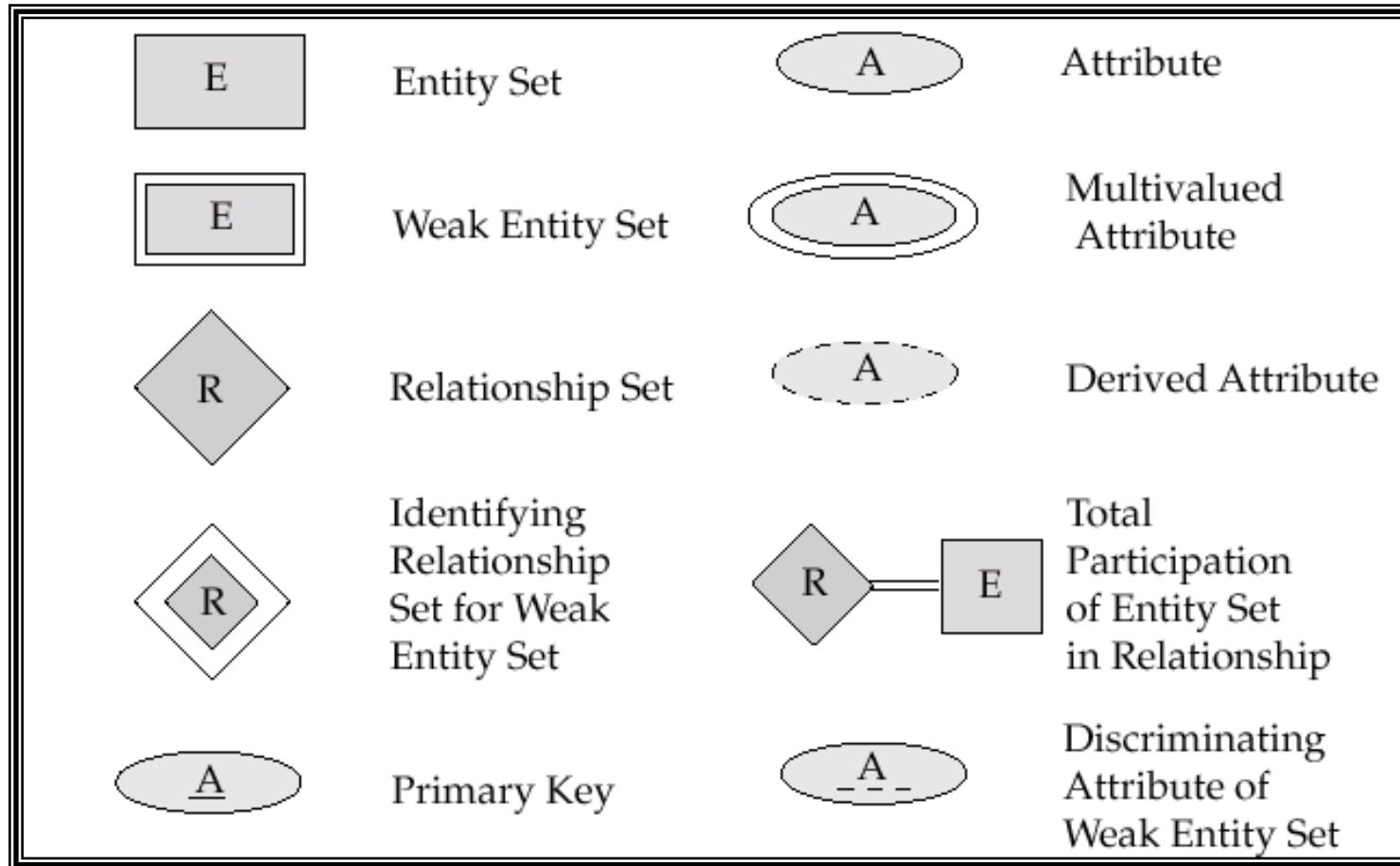
- ⇒ *Dư thừa (Redundancy) xảy ra khi sử dụng cùng 1 vật trong 2 hay nhiều cách khác nhau.*
- ⇒ Dư thừa sẽ làm lãng phí không gian lưu trữ và gây ra mâu thuẫn (inconsistency).
 - Hai instances của cùng 1 kiểu thực thể có thể bị inconsistent nếu thay đổi 1 instance này mà quên thay đổi instance còn lại.

Kiểu thực thể kết hợp Associative entity type

⇒ Bốn điều kiện để chuyên đổi mối liên kết thành kiểu thực thể kết hợp

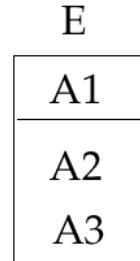
- Là mối liên kết nhiều – nhiều
- Có thuộc tính xác định riêng
- Có thêm vài thuộc tính khác
- Kiểu thực thể kết hợp sẽ tham gia vào 1 số mối liên kết khác trong sơ đồ ER

Tổng kết: Các ký hiệu dùng trong mô hình ER

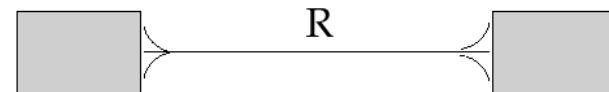
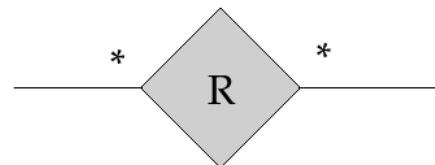


Tổng kết: Các ký hiệu dùng trong mô hình ER

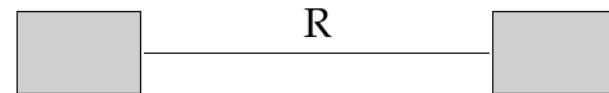
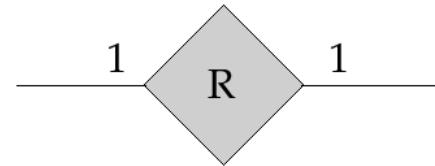
Entity set E with
attributes A1, A2, A3
and primary key A1



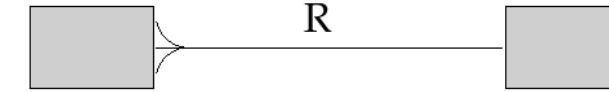
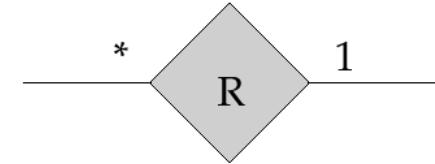
Many to Many
Relationship



One to One
Relationship



Many to One
Relationship



Tổng kết: Các bước để tiến hành thiết kế ER

- ⇒ [Bước 1] Xác định các loại thực thể (loại thực thể hay thuộc tính)
- ⇒ [Bước 2] Xác định loại mối kết hợp
- ⇒ [Bước 3] Xác định và gắn thuộc tính với loại thực thể và loại mối kết hợp
- ⇒ [Bước 4] Quyết định miền giá trị của thuộc tính
- ⇒ [Bước 5] Quyết định các thuộc tính khóa cho loại thực thể
- ⇒ [Bước 6] Gắn (tinh chế) bản số vào loại mối kết hợp
- ⇒ [Bước 7] Thiết kế phân cấp tổng quát hóa/chuyên biệt hóa trong các ràng buộc

Entity Relationship Diagram

- ⇒ DFD và Processing Logic chỉ ra *làm thế nào, ở đâu và khi nào* dữ liệu được xử lý, nhưng không chỉ ra *định nghĩa, cấu trúc và các quan hệ* của dữ liệu.
- ⇒ Entity Relationship Diagram (ERD) là lược đồ thể hiện cấu trúc trừu tượng hóa của dữ liệu trong tổ chức, dựa trên khái niệm thực thể (entity) và quan hệ (relationship) giữa các thực thể, để nhằm thể hiện nội dung, ý nghĩa của dữ liệu trong hệ thống.

Thực thể (Entity)

- ⇒ Nhân viên, Sinh viên, Môn học,... là các thực thể, là một khái niệm tổng quát hóa cho một nhóm các đối tượng (thể hiện, entity instance) trong thế giới thực có chung một số đặc điểm (thuộc tính). Vd: môn “PTTK”, môn “THQL” là các thể hiện của thực thể Môn học.
 1. *Thực thể xác thực* mô tả các đối tượng tồn tại thực sự trong thế giới thực: những chiếc xe đạp, các quyển sách...
 2. *Thực thể chức năng* mô tả mục đích, chức năng, hoặc nhiệm vụ của con người, thiết bị hoặc tổ chức: Sinh viên, nhân viên, khách hàng, nhà kho,...
 3. *Thực thể sự kiện* mô tả các sự kiện hoặc biến cố: biên nhận, biên bản họp, buổi thi,...
 4. *Thực thể quan hệ* mô tả các quan hệ giữa các thực thể: chuyển hàng nhiều lần, thi nhiều lần

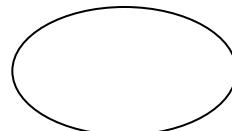
Thực thể (Entity)

- ⇒ **Instance** (thể hiện): là đối tượng cụ thể của thực thể
- ⇒ **Attribute** (thuộc tính) là đặc điểm chung của các thể hiện trong thực thể. Vd: MãNV, Tên, NgàySinh, GiớiTính, NghềNghiệp, KỹNăng là các đặc điểm được chú ý khi nghĩ về người nhân viên và thực thể NhânViên.
- ⇒ **Key** (khóa) là 1 thuộc tính (hoặc kết hợp nhiều thuộc tính) để phân biệt các thể hiện trong thực thể.
 - MãNV là 1 thuộc tính dùng để phân biệt các nhân viên trong tập thực thể NhânViên: Nếu biết MãNV của 1 nhân viên, ta sẽ xác định được tên, địa chỉ và kỹ năng của nhân viên đó.
 - Một giá trị của khóa không bao giờ bị rỗng, trùng nhau, hoặc thay đổi khi thể hiện tương ứng vẫn còn tồn tại.

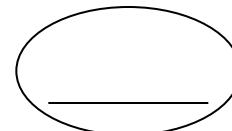
Các ký hiệu



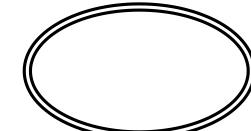
Thực thể



Thuộc tính

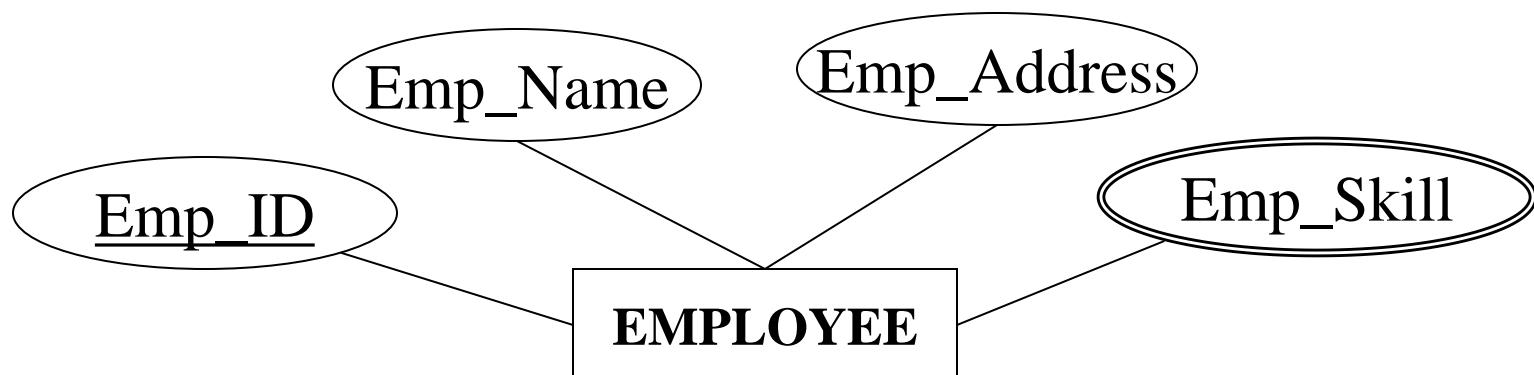


Khóa



Thuộc tính đa trị

Ví dụ: một nhân viên có 1 mã nhân viên dùng để phân biệt. Cơ quan chỉ quan tâm quản lý tên nhân viên, địa chỉ nhà riêng, và các kỹ năng của từng nhân viên → thực thể nhân viên được diễn tả như sau:



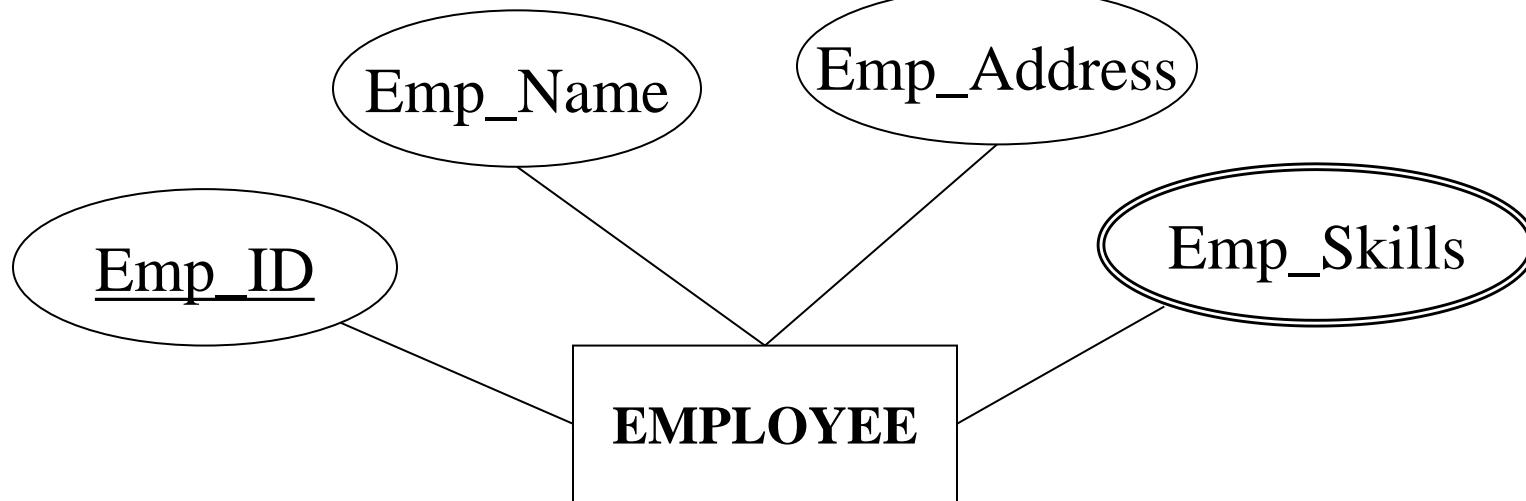
Multivalue Attribute

⇒ là một thuộc tính có nhiều giá trị được dùng để mô tả cho một thể hiện trong thực thể. Vd: thuộc tính “Skill” của 1 nhân viên. Một nhân viên thường có nhiều kỹ năng, do đó EMP_Skill có nhiều giá trị cho từng nhân viên:

EMP_ID	EMP_Name
--------	----------

0210-67	Susan
---------	-------

0210-67	Susan
---------	-------



- ⇒ Là mối liên kết giữa một hoặc nhiều thực thể để chỉ ra sự liên kết về nội dung (và ý nghĩa) giữa các thực thể trong mối liên kết. Ví dụ: “Mỗi SINH VIÊN đăng ký nhiều MÔN HỌC”. Sự liên kết nội dung giữa thực thể Sinh viên và thực thể Môn học là việc đăng ký môn để học của mỗi sinh viên.
- ⇒ **Cardinality:** Theo quan hệ R giữa 2 thực thể A và B, số của quan hệ (cardinality) ở phía B là số thể hiện của thực thể B có thể (hoặc phải) liên kết với mỗi thể hiện ở thực thể A.
Vd: một sinh viên phải đăng ký học ít nhất là 1 môn, và nhiều nhất là 6 môn trong một học kỳ: ở phía thực thể môn, số của quan hệ học (số môn được 1 sinh viên nào đó học) sẽ là từ 1 đến 6.

Ý nghĩa của cardinality



Mỗi thể hiện của thực thể A có đúng 1 thể hiện tương ứng ở thực thể B theo quan hệ R1 (cardinality = [1,1]).



Mỗi thể hiện của A chỉ có 1 thể hiện tương ứng ở B, hoặc không có thể hiện tương ứng theo quan hệ R2 (cardinality = [0,1]).

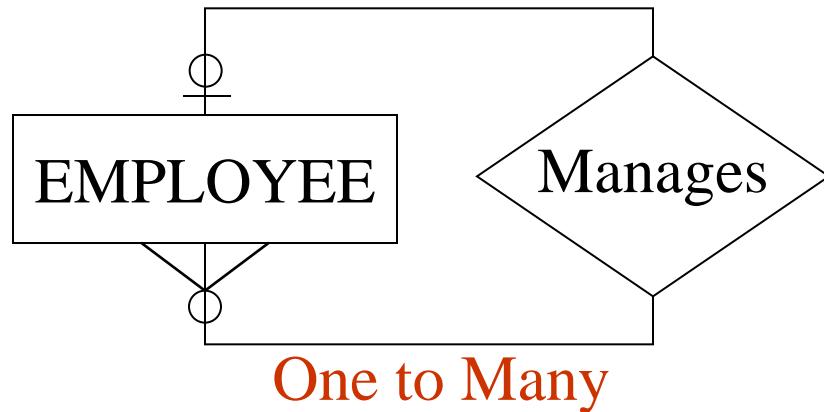


Mỗi thể hiện của A có ít nhất là 1 và tối đa là N thể hiện tương ứng ở B theo quan hệ R3 (cardinality = [1,N]).

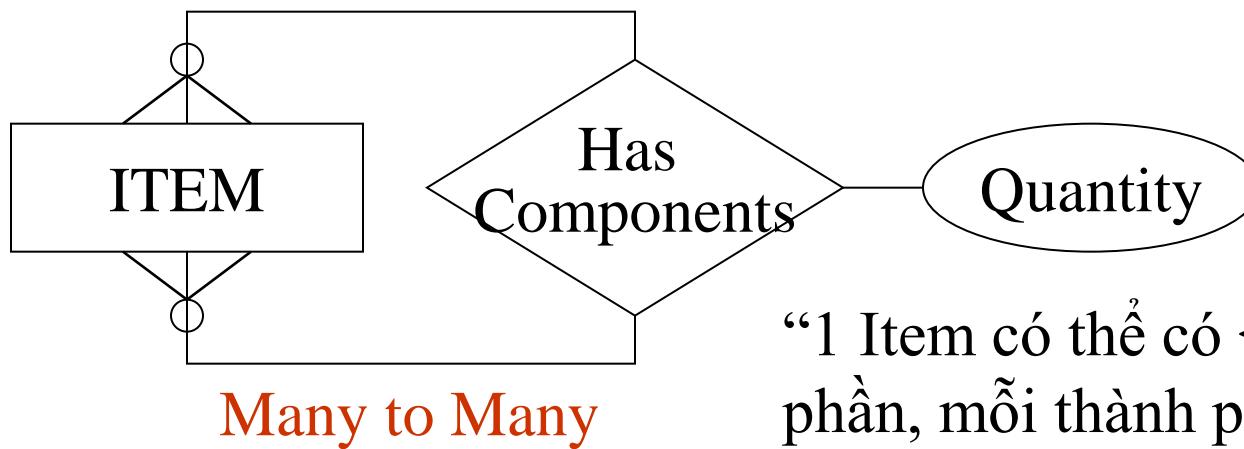


Mỗi thể hiện của A có tối đa là N thể hiện tương ứng ở B, hoặc không có thể hiện tương ứng theo quan hệ R4 (cardinality = [0,N]).

Unary Relationship

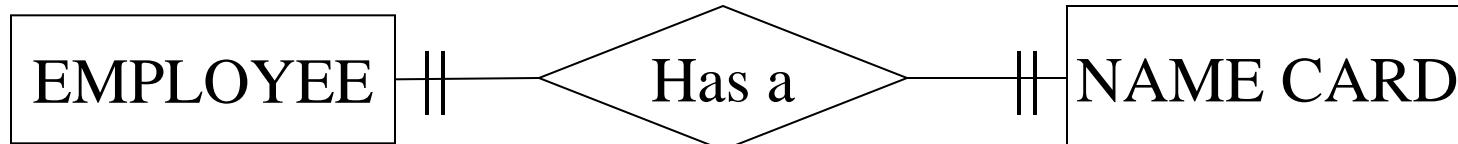


“Một nhân viên có thể quản lý nhiều nhân viên”



“1 Item có thể có <quantity> thành phần, mỗi thành phần cũng là 1 Item”

Binary Relationship



One to One (1:1)

“Một nhân viên phải có (duy nhất) 1 bảng tên.”



One to Many (1:N)

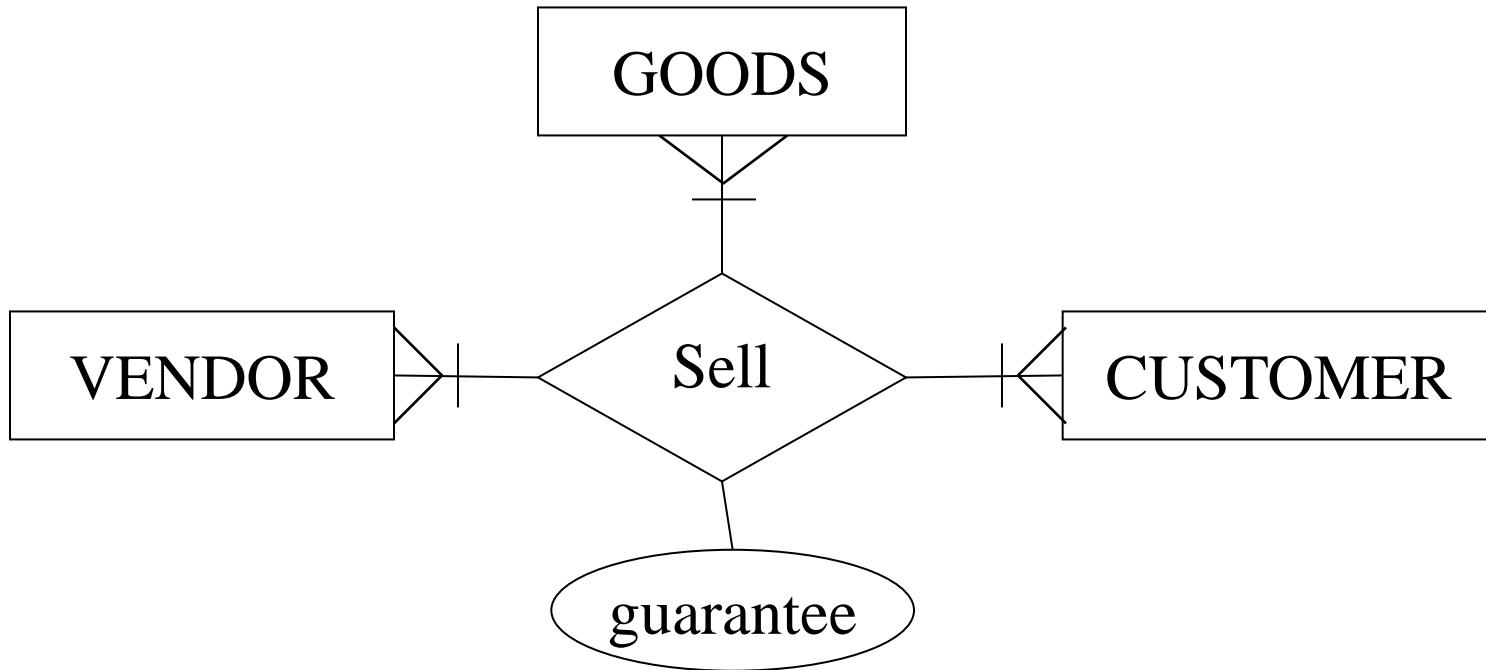
“Một dây chuyền sản phẩm phải chứa 1 hoặc nhiều sản phẩm. Một sản phẩm phải thuộc 1 dây chuyền sản xuất.”



Many to Many (M:N)

“Một sinh viên phải đăng ký 1 hoặc nhiều môn học. Một môn học có thể có nhiều sinh viên đăng ký, hoặc không có sinh viên đăng ký”

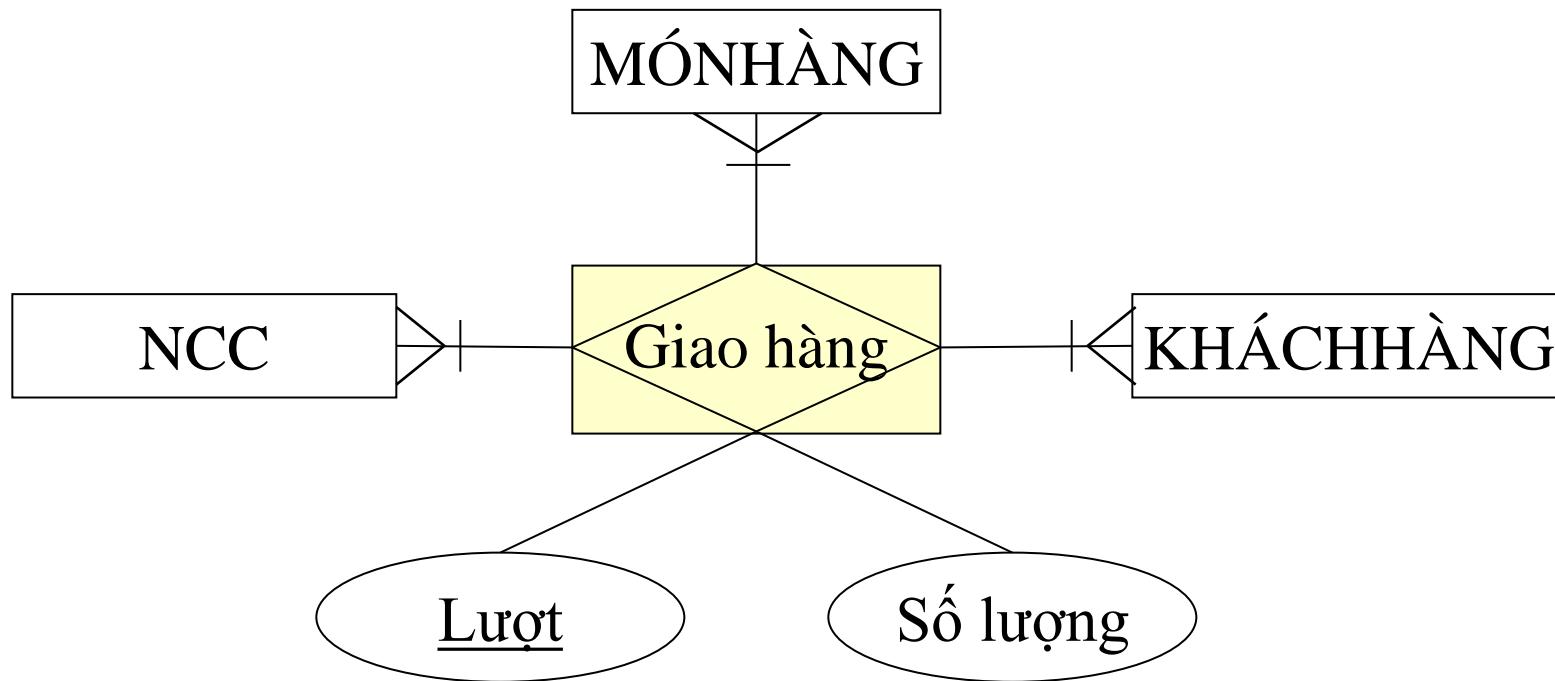
Ternary Relationship



“Một nhà cung cấp có thể bán nhiều mặt hàng cho nhiều khách hàng; khách hàng có thể mua hàng từ nhiều nhà cung cấp khác nhau và có bảo hành”

Associative Entity (thực thể liên kết)

⇒ Là thực thể diễn tả các quan hệ (liên kết) cụ thể giữa các thẻ hiện tham gia vào quan hệ.



1. *Định nghĩa các thực thể*, dựa trên vai trò, ý nghĩa của thực thể đối với hệ thống. Nên chọn danh từ để dùng làm tên cho thực thể, vd: MONHOC, SINHVIEN, KHOA,..
2. *Định nghĩa các quan hệ giữa các thực thể*. Tên của các quan hệ thường được diễn tả bằng động từ để chỉ các hành động, sự kiện liên kết các thể hiện trong các thực thể có quan hệ nhau.
3. *Xác định các thuộc tính của thực thể và quan hệ*. Thuộc tính của thực thể (hoặc quan hệ) là những đặc tính mà tất cả các thể hiện của thực thể (hoặc quan hệ) đều có. Thêm thuộc tính để tăng tính mô tả, hoặc để có thể dữ liệu phân biệt các thể hiện. Bỏ bớt thuộc tính nếu chúng dư thừa hoặc không liên quan đến vai trò, ý nghĩa của thực thể trong hệ thống.
4. *Xác định cardinality cho mỗi quan hệ*.

XIN CẢM ƠN!