

INT1313- Cơ Sở Dữ Liệu

Mô Hình Hóa Dữ Liệu Khái niệm và Thiết kế CSDL EER - Mô hình ER nâng cao

Giảng viên: Lê Hà Thanh
lehathanh@ptithcm.edu.vn

Nội dung

- Lớp con, lớp cha và sự thừa kế
- Chuyên môn hóa và Tổng quát hóa
- Các ràng buộc và đặc tính của Chuyên môn hóa và Tổng quát hóa
- Mô hình hóa các kiểu UNION bằng các danh mục
- Bài mẫu: CSDL UNIVERSITY
- Biểu diễn Chuyên môn hóa và Tổng quát hóa bằng sơ đồ lớp UML
- Khái niệm Trừu tượng hóa dữ liệu, biểu diễn tri thức, và ontology

Lớp con, lớp cha và sự thừa kế

Subclasses, superclasses, and inheritance

Lớp con, lớp cha và quan hệ IS-A

Lớp con (subclass/subtype): các nhóm con của một kiểu lớp thực thể, có ý nghĩa cụ thể trong các ứng dụng thực tế.

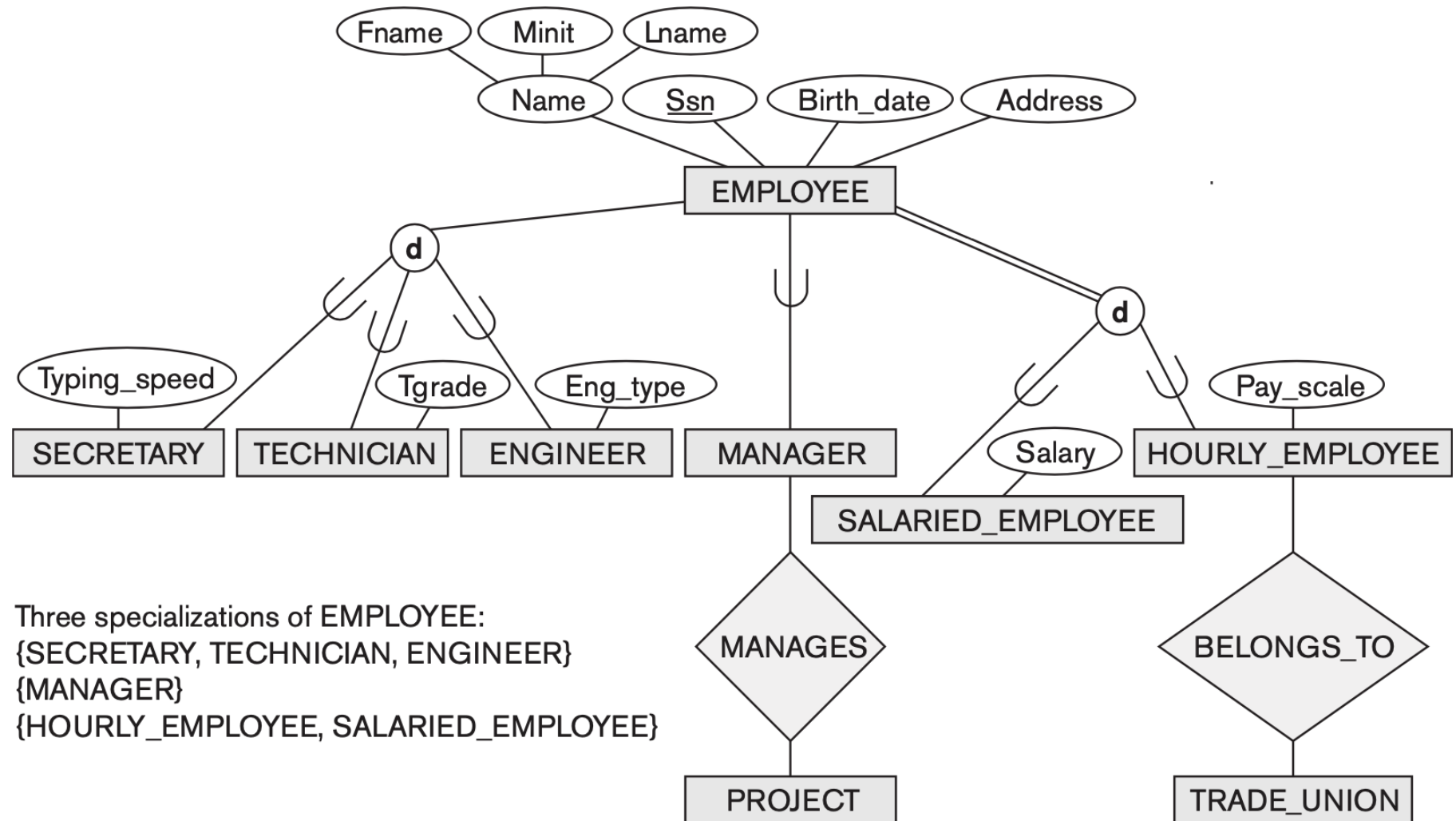
Ví dụ: **EMPLOYEE** có thể được phân biệt cụ thể hơn với **SECRETARY**, **ENGINEER**, **MANAGER**, **TECHNICIAN**, **SALARIED_EMPLOYEE**, **HOURLY_EMPLOYEE**, v.v.

Kiểu lớp gốc còn được gọi là **siêu lớp/lớp cha** (superclass/supertype).

Quan hệ giữa một lớp cha và lớp con của nó được gọi là **quan hệ lớp cha/lớp con** (superclass/subclass, supertype/subtype, hoặc class/subclass), hay còn gọi là quan hệ “là một” (**IS-A/IS-AN** relationship).

Ví dụ: - Một **SECRETARY** là một **EMPLOYEE**, một **TECHNICIAN** là một **EMPLOYEE**

Ví dụ



Thực thể của lớp con và siêu lớp

Một thành viên của lớp con cũng là một thành viên của siêu lớp, nhưng thể hiện một vai trò cụ thể riêng (so với các thành viên thuộc các lớp con khác).

Một thực thể của lớp con phải là thành viên của siêu lớp của nó. Thực thể này có thể được bao gồm như là một thành viên trong bất kỳ lớp con nào.

Ví dụ: "nhân sự được trả lương cũng là một kỹ sư thuộc về hai lớp con ENGINEER và SALARIED_EMPLOYEE của cùng lớp cha EMPLOYEE."

Ngược lại, mọi thực thể của siêu lớp không nhất thiết phải là một thành viên của một lớp con.

Sự kế thừa kiểu

- Kiểu của thực thể được định nghĩa bằng các thuộc tính và các kiểu quan hệ mà thực thể có tham gia.
- Một thực thể của lớp con có các giá trị của các thuộc tính đặc trưng riêng của nó, và các giá trị của các thuộc tính quy định nó là thành viên của siêu lớp.

Định nghĩa:

Một thực thể là thành viên của một lớp con kế thừa (inherits) tất cả thuộc tính quy định nó là thành viên của siêu lớp.

Thực thể cũng kế thừa tất cả mối quan hệ mà siêu lớp tham gia.

Một lớp con có thể được xem là một kiểu thực thể riêng.

Chuyên môn hóa và Tổng quát hóa

Specialization and Generalization

Chuyên môn hóa

Chuyên môn hóa là quá trình định nghĩa một tập hợp các lớp con của một kiểu thực thể; Kiểu thực thể này được gọi là siêu lớp (lớp cha) của chuyên môn hóa.

Tập hợp các lớp con tạo nên chuyên môn hóa được xác định dựa trên cơ sở một số đặc điểm phân biệt của các thực thể trong siêu lớp.

Các đặc điểm phân biệt khác nhau dẫn đến các chuyên môn hóa khác nhau.

Ví dụ:

- Tập hợp {**SECRETARY**, **ENGINEER**, **TECHNICIAN**} là một chuyên môn hóa của siêu lớp **EMPLOYEE**, phân biệt các thực thể nhân viên dựa trên loại công việc của mỗi nhân viên.
- Tập hợp {**SALARIED_EMPLOYEE**, **HOURLY_EMPLOYEE**} là một chuyên môn hóa của **EMPLOYEE** dựa trên phương thức trả lương.

Các ký hiệu trong sơ đồ EER

Chuyên môn hóa

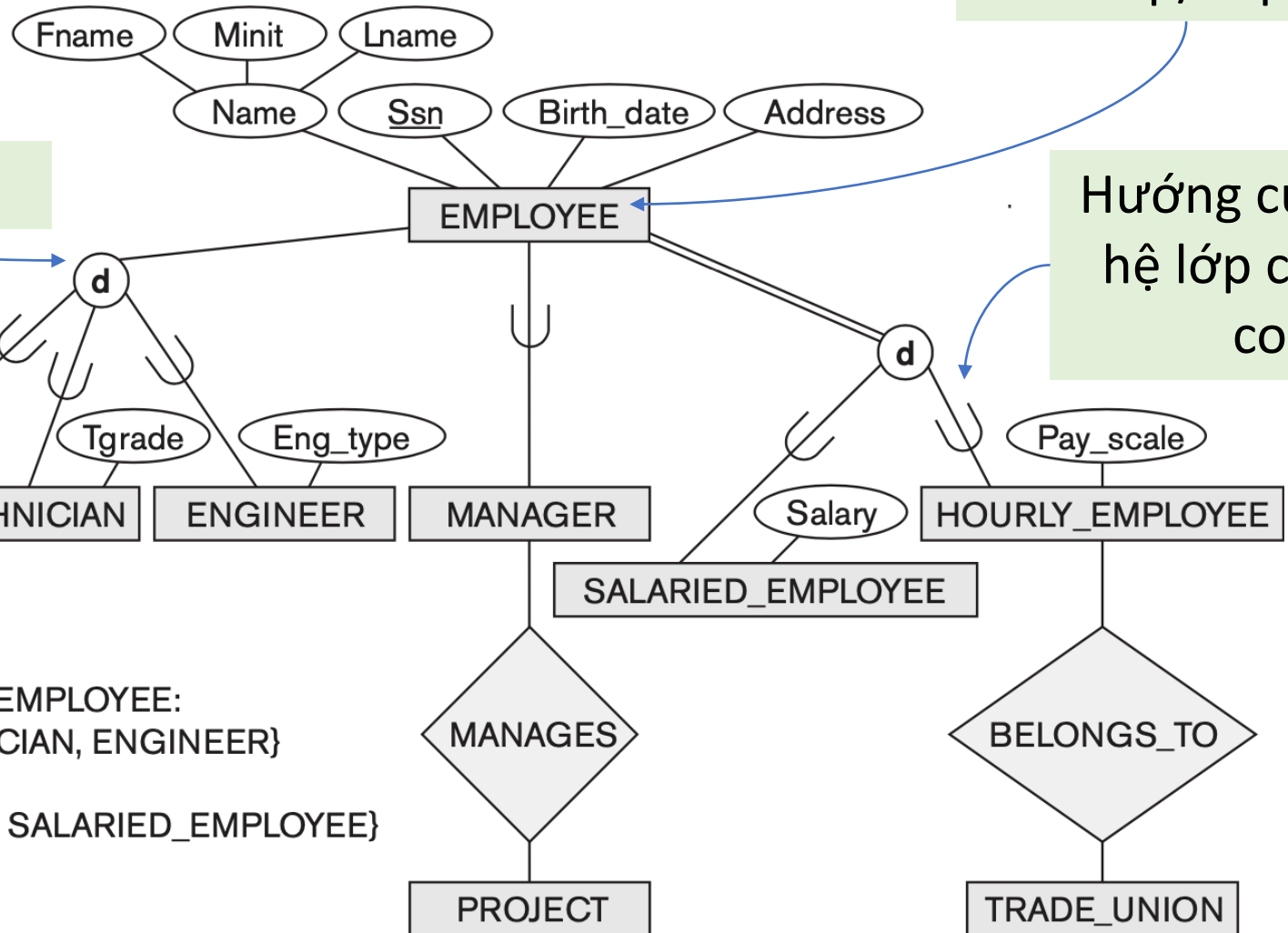
Siêu lớp/Lớp cha

Hướng của quan hệ lớp cha/lớp con

Thuộc tính chuyên biệt

Lớp con

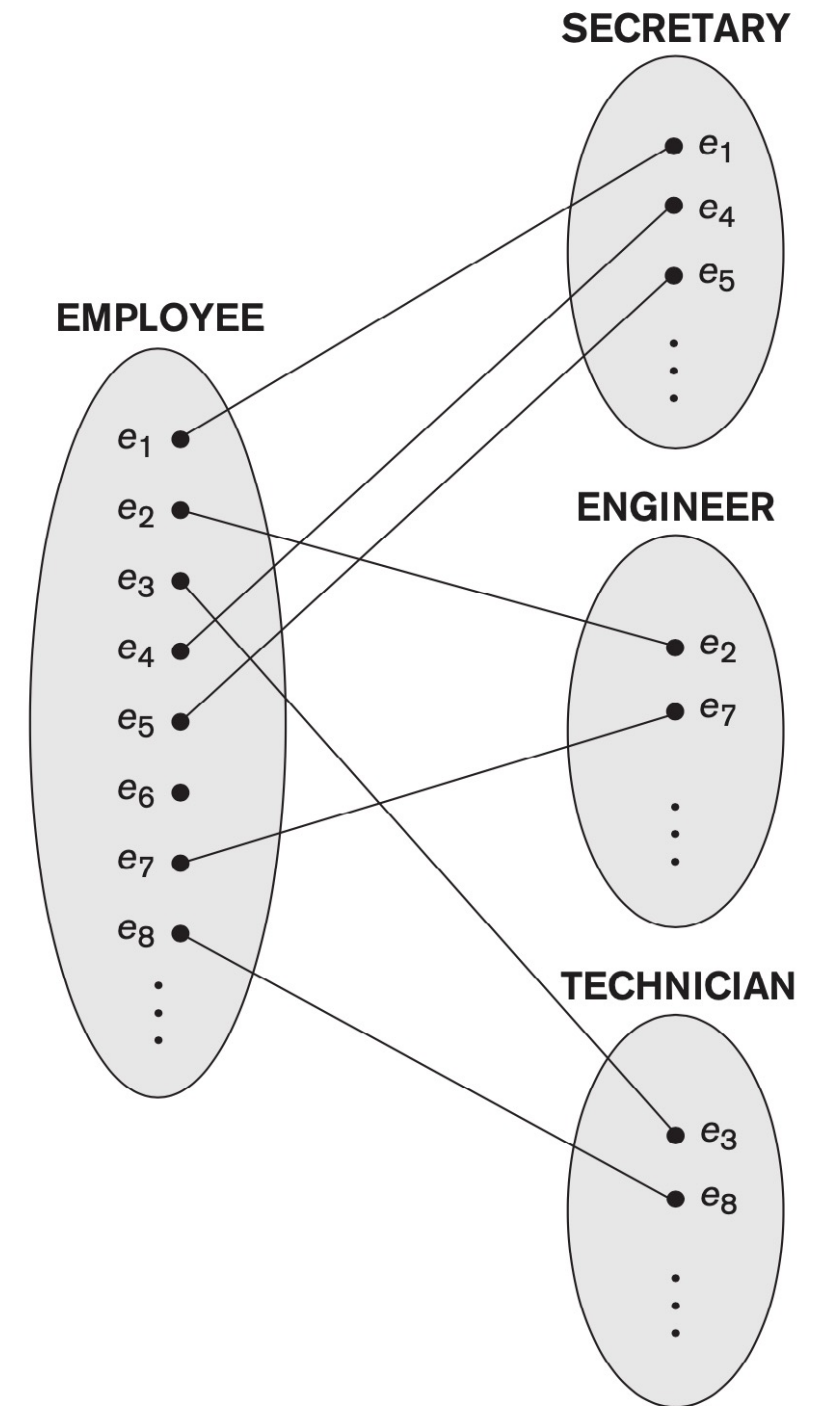
Three specializations of EMPLOYEE:
{SECRETARY, TECHNICIAN, ENGINEER}
{MANAGER}
{HOURLY_EMPLOYEE, SALARIED_EMPLOYEE}



Quan hệ lớp cha/lớp con và Chuyên môn hóa

Thực thể trong lớp con là cùng một thực thể trong thế giới thực với thực thể trong lớp cha, nhưng đóng vai trò chuyên môn hóa.

Ví dụ: một EMPLOYEE chuyên môn hóa trong vai trò là SECRETARY, hoặc là TECHNICIAN.



Quan hệ lớp cha/lớp con và Chuyên môn hóa

Một số thuộc tính nhất định có thể áp dụng cho một vài thực thể của kiểu thực thể lớp cha.

Một lớp con được định nghĩa để gom nhóm các thực thể theo các thuộc tính này.

Các thành viên của lớp con có thể chia sẻ phần lớn các thuộc tính của chúng với các thành viên khác trong lớp cha.

Ví dụ: lớp con **SECRETARY** có thuộc tính **Typing_speed**, lớp **ENGINEER** có thuộc tính **Eng_type**, và hai lớp này cùng chia sẻ các thuộc tính khác được kế thừa từ kiểu thực thể **EMPLOYEE**.

Quan hệ lớp cha/lớp con và Chuyên môn hóa

Một số kiểu quan hệ chỉ áp dụng được với các thành viên của lớp con.

Ví dụ: quan hệ giữa HOURLY_EMPLOYEES với TRAD_UNION qua BELONGS_TO

Tổng quát hóa

Quá trình đảo ngược của sự trừu tượng hóa, trong đó thực hiện loại bỏ các khác biệt giữa một số loại thực thể, xác định các điểm chung của chúng và khái quát chúng thành một siêu lớp duy nhất.

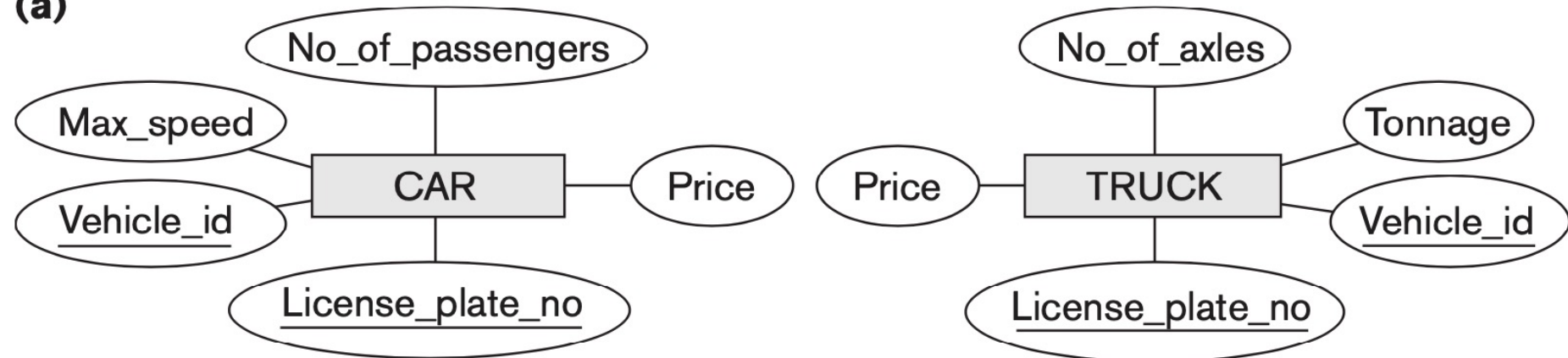
- Các lớp ban đầu trở thành các lớp con đặc biệt của siêu lớp này.

Ví dụ

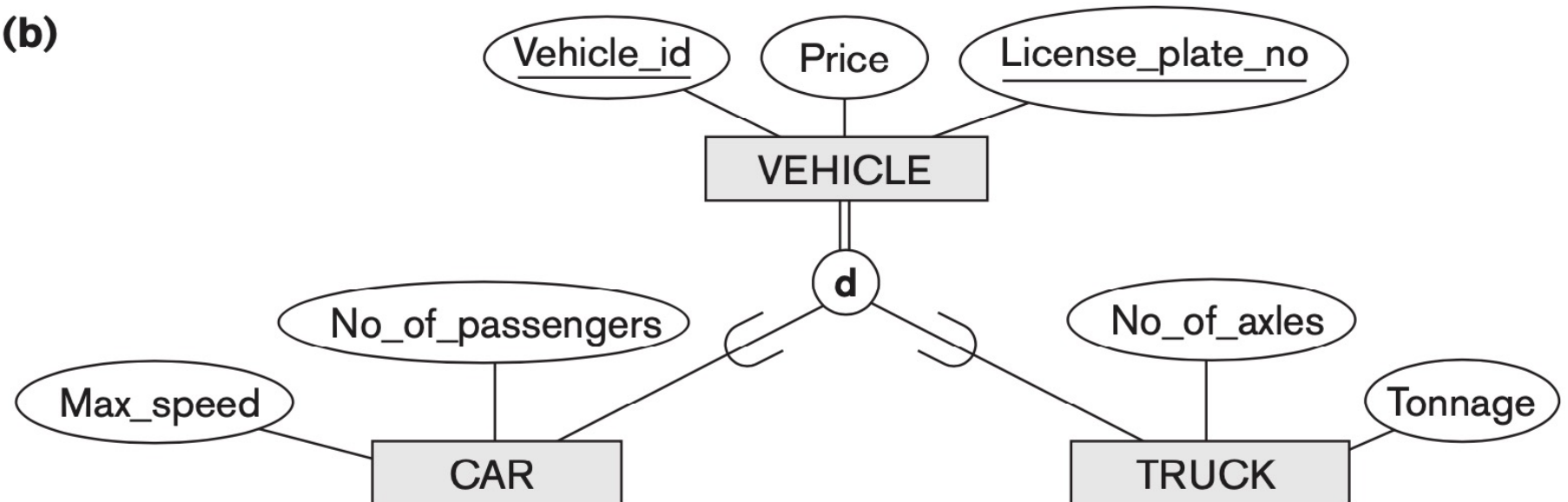
Khái quát kiểu lớp
CAR và TRUCK

→ Siêu lớp VEHICLE

(a)



(b)



Các ràng buộc và Đặc điểm của
hệ thống phân cấp Chuyên môn
hóa và Tổng quát hóa

Xác định chuyên môn hóa

- Có nhiều chuyên môn hóa được định nghĩa trên cùng một siêu lớp.
- Một chuyên môn hóa có thể bao gồm chỉ duy nhất một lớp con.

Chuyên môn hóa được định nghĩa bởi:

1. Vị từ điều kiện
2. Thuộc tính
3. Người dùng

Chuyên môn hóa xác định bằng vị từ điều kiện

Với một số chuyên môn hóa, ta có thể xác định chính xác các thực thể sẽ trở thành thành viên của mỗi lớp con bằng cách đưa thêm một điều kiện áp dụng vào giá trị của một vài thuộc tính của siêu lớp.

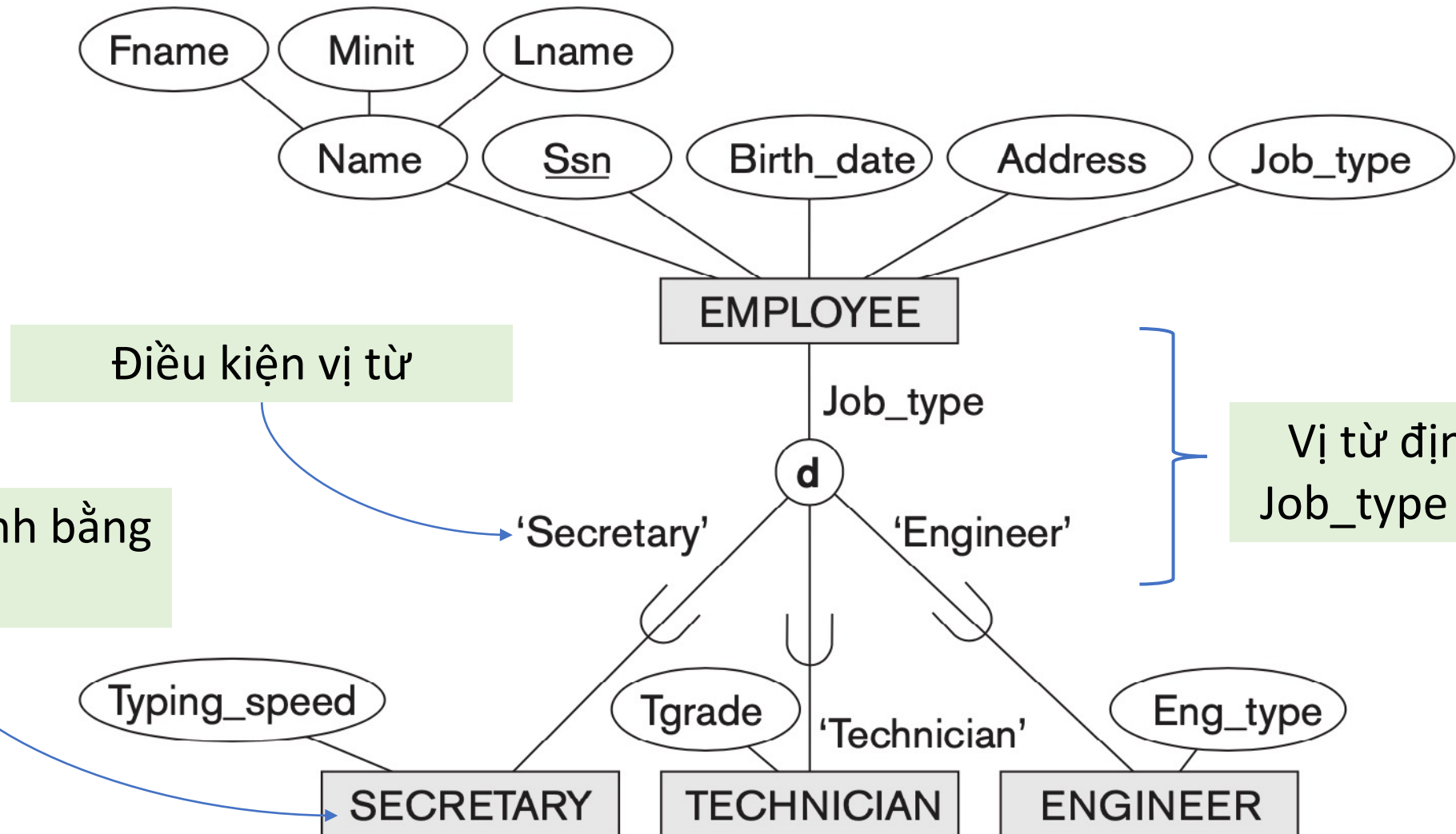
➔ Các lớp con **được xác định bởi vị từ** (hoặc **được xác định bằng điều kiện**)

Ví dụ: `EMPLOYEE.Job_type = 'Secretary'`

➔ **Vị từ định nghĩa** lớp con.

- Còn gọi là ràng buộc xác định thực thể nào của siêu lớp sẽ thuộc về lớp con.

Ví dụ và ký hiệu

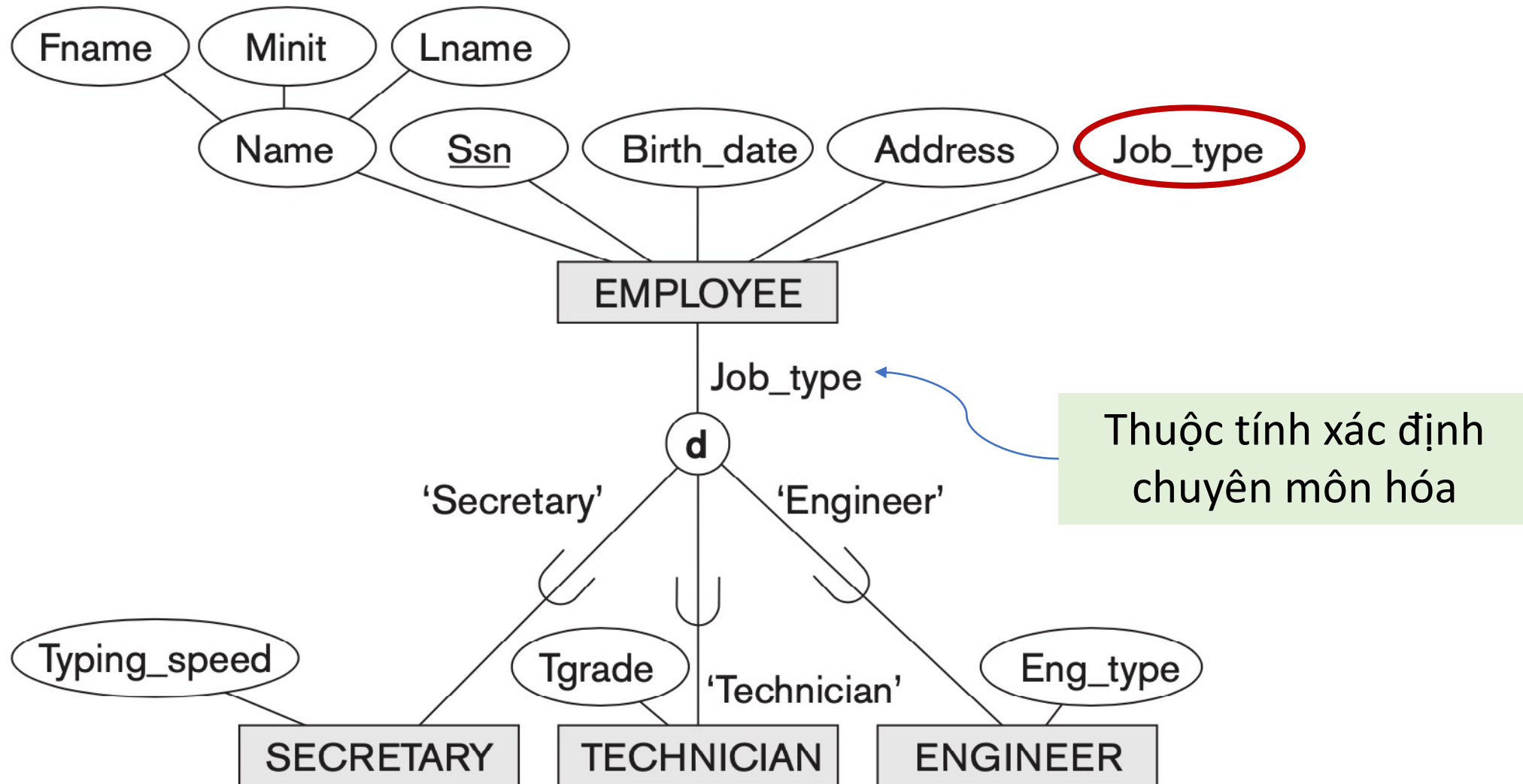


Chuyên môn hóa xác định bằng thuộc tính

Nếu tất cả các lớp con chuyên môn hóa có điều kiện thành viên của chúng trên cùng một thuộc tính của siêu lớp, thì phép chuyên môn hóa này được gọi là **chuyên môn hóa được xác định bằng thuộc tính** (attribute-defined specialization); và thuộc tính được gọi là thuộc tính xác định chuyên môn hóa.

Tất cả thực thể mà thuộc tính có cùng giá trị thì được xếp vào cùng một lớp con.

Ví dụ và ký hiệu



Chuyên môn hóa xác định theo người dùng

Khi không có điều kiện để xác định thành viên trong lớp con, thì lớp con được gọi là được xác định theo người dùng (**user-defined**): người dùng quy định và áp dụng thao tác xử lý để thêm một thực thể vào lớp con.

- Xác định thành viên được người dùng thực hiện riêng cho từng thực thể, không thông qua bất kỳ điều kiện đánh giá tự động nào.

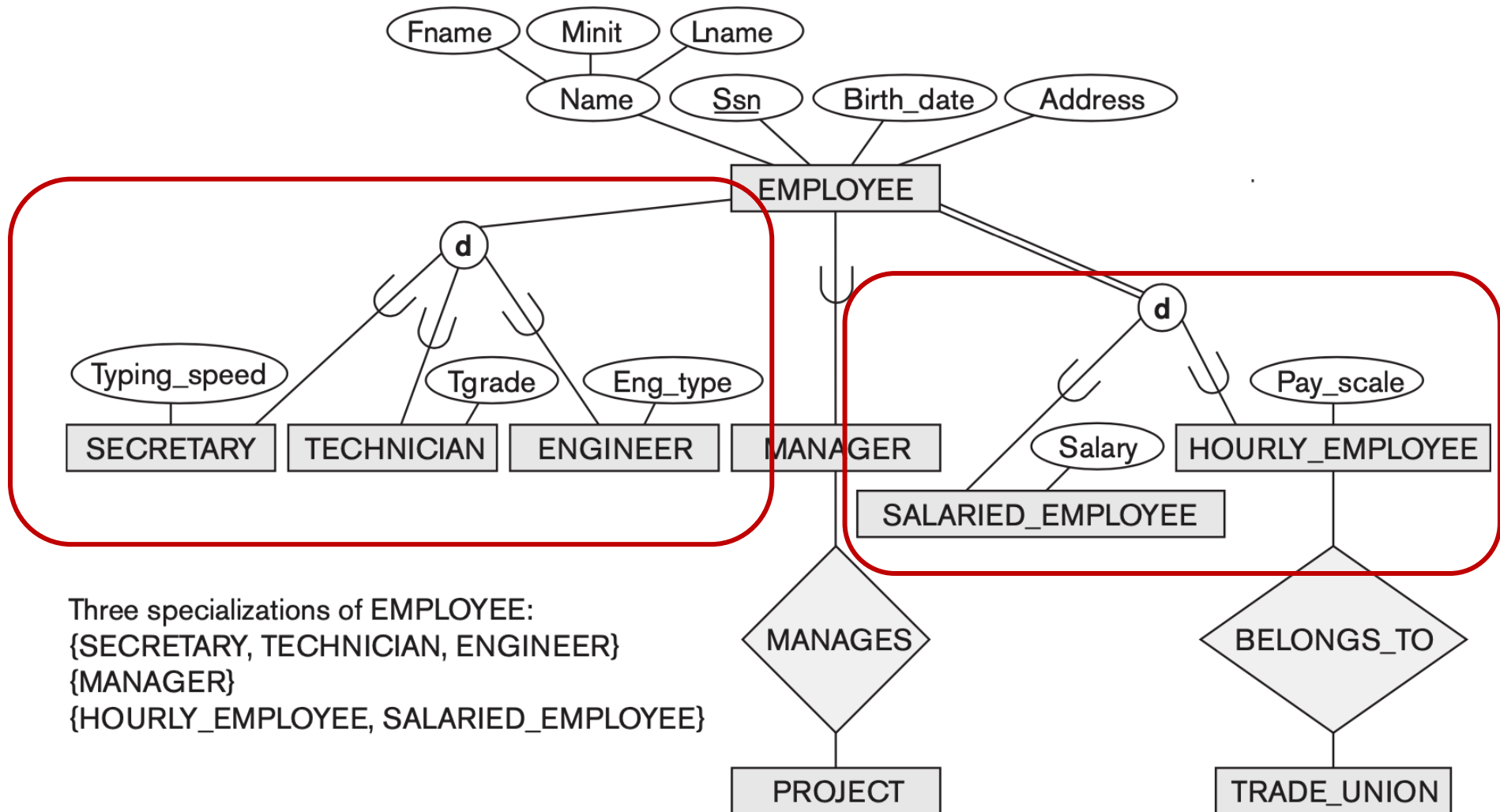
Chuyên môn hóa bằng ràng buộc rời rạc

Ràng buộc rời rạc (**disjoint constraint**) quy định các lớp con của chuyên môn hóa phải là các tập hợp rời rạc.

~ Một thực thể có thể là thành viên của *tối đa* một lớp con của chuyên môn hóa.

- Chuyên môn hóa xác định bằng thuộc tính ngầm định ràng buộc rời rạc
- Ký hiệu (**d**) - *disjoint* trong sơ đồ EER.

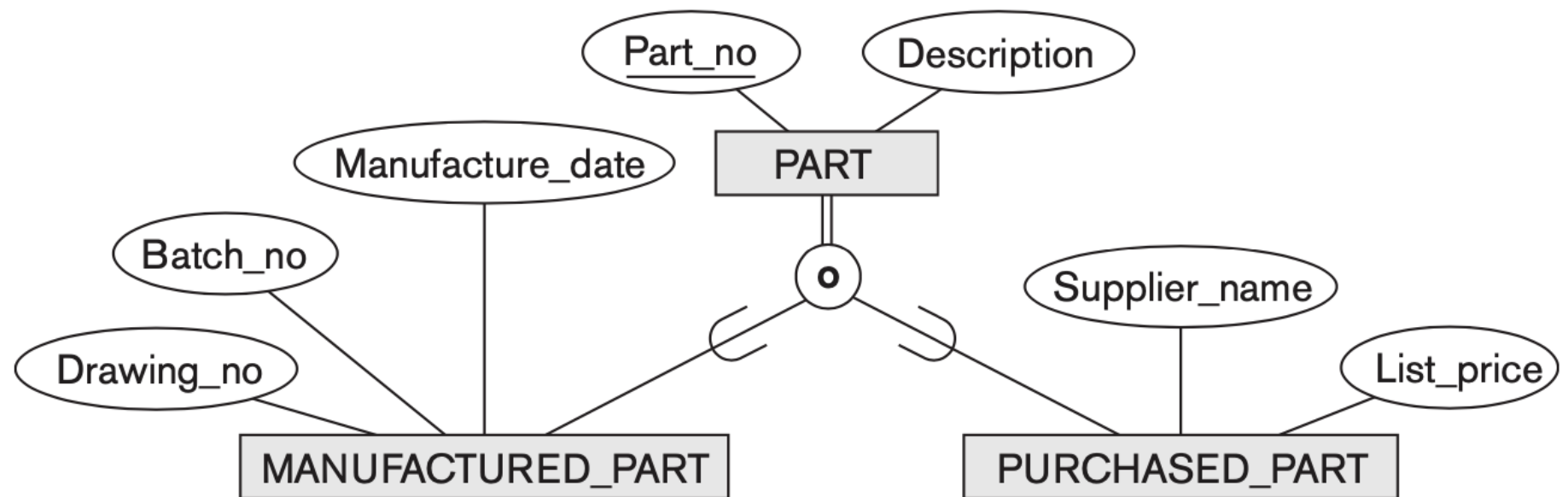
Ràng buộc rời rạc (d) trong sơ đồ EER



Sự chồng lấn trong chuyên môn hóa

Nếu các lớp con không bị ràng buộc rời rạc, tập hợp các thực thể của chúng có thể chồng lấn lên nhau (**overlapping**); nghĩa là, một thực thể trong thực tế có thể là một thành viên của nhiều hơn một lớp con của chuyên môn hóa.

Ký hiệu (o)
trong sơ
đồ EER.



Chuyên môn hóa bằng ràng buộc đầy đủ

Ràng buộc chuyên môn hóa toàn phần (**total specialization**) quy định rằng mọi thực thể trong siêu lớp phải là thành viên của ít nhất một lớp con trong chuyên môn hóa.

Ví dụ: Nếu mọi nhân viên **EMPLOYEE** phải là một **HOURLY_EMPLOYEE** hoặc là một **SALARIED_EMPLOYEE**, thì chuyên môn hóa {**HOURLY_EMPLOYEE**, **SALARIED_EMPLOYEE**} là một chuyên môn hóa toàn phần của **EMPLOYEE**.

Ký hiệu:

- ||: chuyên môn hóa toàn phần, nối siêu lớp tới ký hiệu chuyên môn hóa.

Chuyên môn hóa bằng ràng buộc một phần

Ràng buộc chuyên môn hóa một phần (**partial specialization**) cho phép một thực thể không thuộc về bất kỳ một lớp con nào.

Ví dụ: Nếu một số thực thể **EMPLOYEE** không thuộc về bất kỳ lớp con {**SECRETARY, ENGINEER, TECHNICIAN**} thì đây là phép chuyên môn hóa một phần.

Ký hiệu:

- |: chuyên môn hóa một phần.

Bốn ràng buộc chuyên môn hóa

- Các ràng buộc rời rạc và ràng buộc toàn phần là độc lập với nhau.

➔ Có bốn loại ràng buộc chuyên môn hóa:

- Rời rạc, toàn phần
- Rời rạc, một phần
- Chồng lấp, toàn phần
- Chồng lấp, một phần

Ràng buộc quy định chuyên môn hóa được xác định từ thực tế.

Một cách tổng quát, một siêu lớp được xác lập thông qua quá trình tổng quát hóa thường là chuyên môn hóa toàn phần, do siêu lớp được dẫn xuất từ các lớp con và chỉ chứa các thực thể trong các lớp con này.

Chèn và xóa thực thể trong chuyên môn hóa và tổng quát hóa

Áp dụng các luật chèn và luật xóa tổng quát sau:

- Việc xóa một thực thể khỏi một siêu lớp hàm ý rằng thực thể được xóa một cách tự động khỏi tất cả các lớp con chứa nó.
- Việc chèn một thực thể vào một siêu lớp hàm ý rằng thực thể được chèn bắt buộc vào tất cả các lớp con được xác định bằng vị từ điều kiện (hoặc xác định bằng thuộc tính) mà thực thể thỏa mãn vị từ xác định lớp.
- Việc chèn một thực thể vào một siêu lớp trong chuyên môn hóa toàn phần hàm ý rằng thực thể được chèn bắt buộc vào ít nhất một trong số các lớp con của chuyên môn hóa.

Cấu trúc phân cấp và cấu trúc lưới trong chuyên môn hóa và tổng quát hóa

Một lớp con có thể có các lớp con của riêng nó.

Các siêu lớp, và lớp con ở các cấp khác nhau tạo thành một cấu trúc phân cấp (hierachy) hoặc cấu trúc lưới (lattice) chuyên môn hóa.

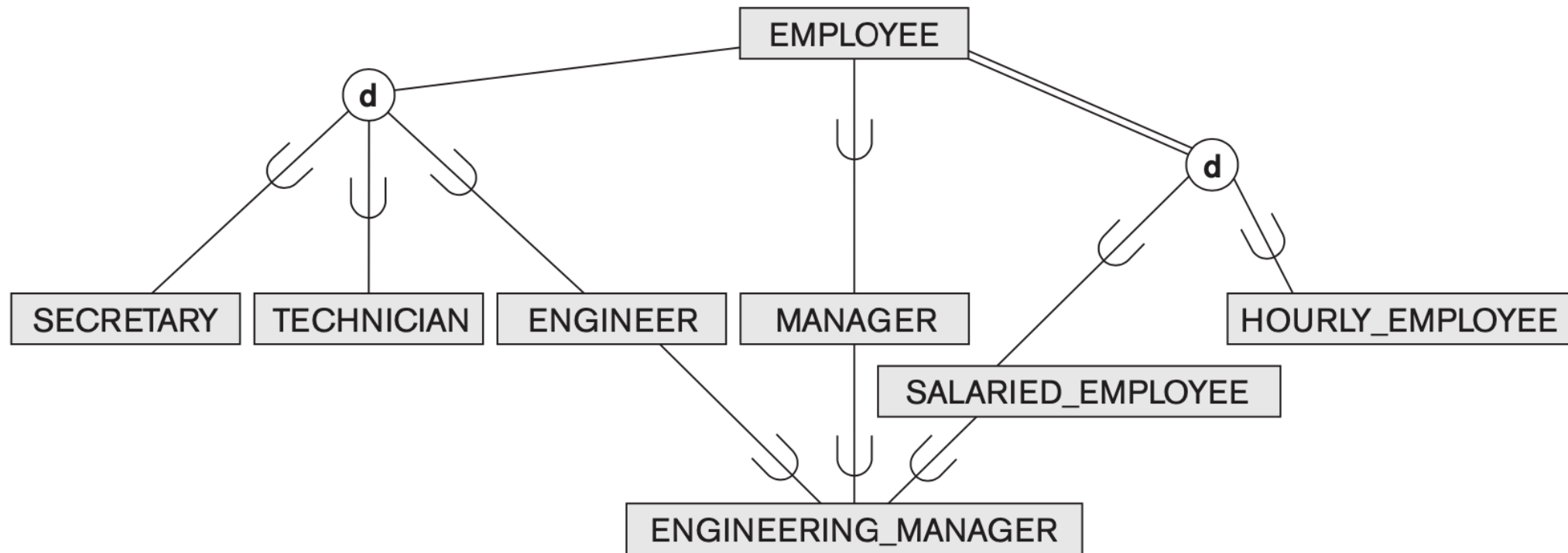
Cấu trúc phân cấp chuyên môn hóa có một ràng buộc quy định mọi lớp con tham gia vào mỗi quan hệ “là một” (as a) trong chỉ một quan hệ lớp/lớp con duy nhất. Nghĩa là mỗi lớp con chỉ có duy nhất một lớp cha.

➔ Cấu trúc dạng cây, hay còn gọi là cấu trúc phân cấp chặt chẽ.

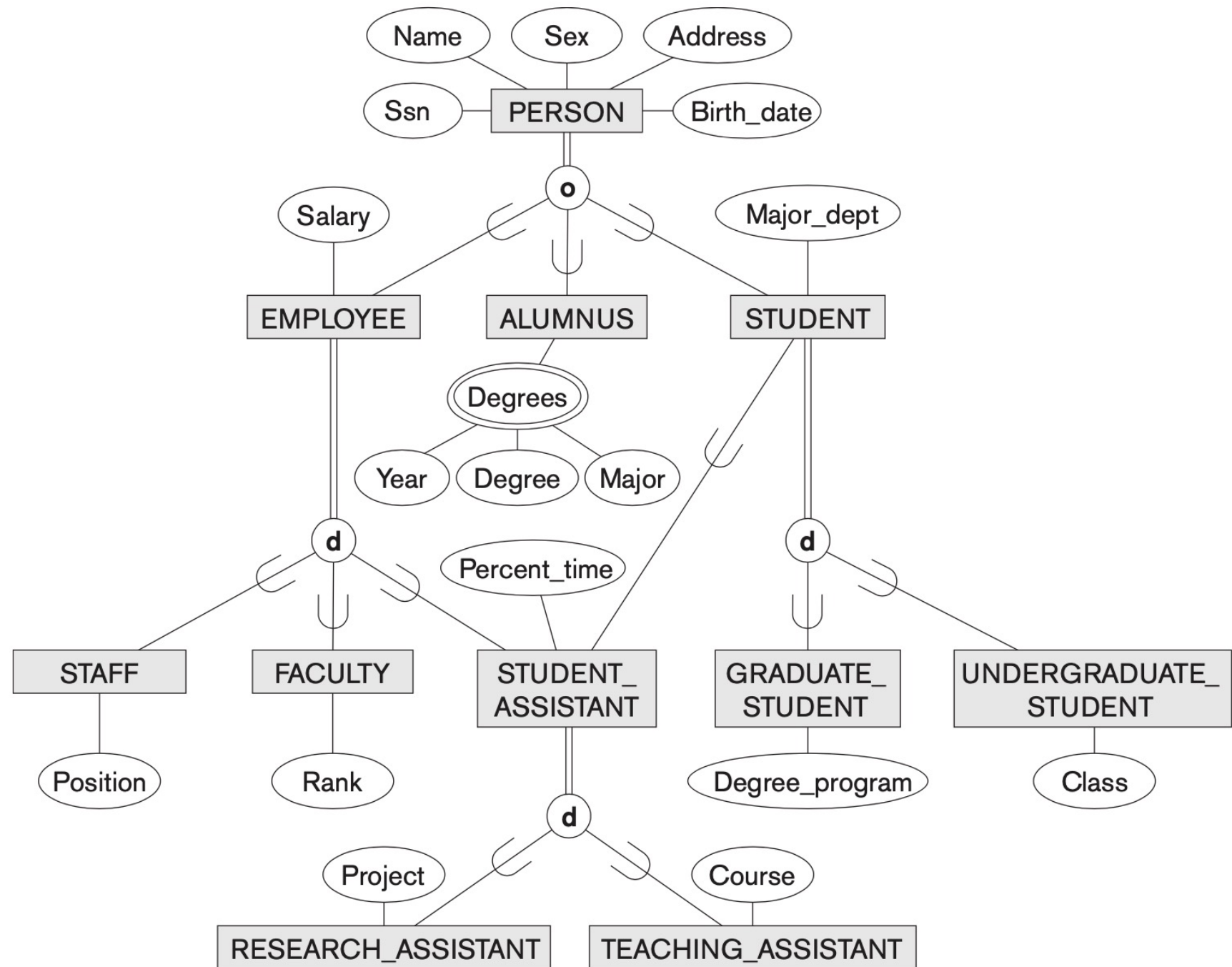
Cấu trúc lưới chuyên môn hóa cho phép một lớp con có thể là con trong từ một trở lên mỗi quan hệ lớp/lớp con.

Ví dụ 1

Đây là một cấu trúc chuyên môn hóa dạng lưới.



Ví dụ 2



Đặc tả yêu cầu của Ví dụ 2

1. Cơ sở dữ liệu theo dõi ba loại người: nhân viên, cựu sinh viên và sinh viên. Một người có thể thuộc một, hai hoặc cả ba loại này. Mỗi người có tên, SSN, giới tính, địa chỉ và ngày sinh.
2. Mỗi nhân viên đều có lương và có ba loại nhân viên: giảng viên, nhân viên và trợ lý sinh viên. Mỗi nhân viên thuộc chính xác một trong những loại này. Đối với mỗi cựu sinh viên, hồ sơ về bằng cấp hoặc các bằng cấp mà họ đạt được tại trường đại học được lưu giữ, bao gồm tên bằng cấp, năm cấp và khoa chính. Mỗi sinh viên có một khoa chính.
3. Mỗi khoa có một bậc xếp hạng, trong khi mỗi nhân viên có một vị trí nhân viên. Trợ lý sinh viên được phân loại thêm thành trợ lý nghiên cứu hoặc trợ lý giảng dạy và phần trăm thời gian họ làm việc được ghi lại trong cơ sở dữ liệu. Trợ lý nghiên cứu có dự án nghiên cứu được lưu trữ, trong khi trợ lý giảng dạy có khóa học hiện tại mà họ đang theo học.
4. Sinh viên được phân loại thành sau đại học hoặc đại học, với các thuộc tính cụ thể của chương trình cấp bằng (Thạc sĩ, Tiến sĩ, MBA, v.v.) dành cho sinh viên sau đại học và lớp (sinh viên năm nhất, năm hai, v.v.) dành cho sinh viên đại học.

Các tính chất của cấu trúc phân cấp và cấu trúc lưới chuyên môn hóa

- Một lớp con kế thừa các thuộc tính của siêu lớp trực tiếp, các thuộc tính của các siêu lớp phía trên nữa, cho đến siêu lớp gốc của cấu trúc.
- Một thực thể có thể tồn tại trong nhiều nút lá. Nút lá là một lớp không có các lớp con của riêng nó.
Ví dụ: một thành viên của **GRADUATE_STUDENT** có thể cũng là một thành viên của **RESEARCH_ASSISTANT**.
- Một lớp con có nhiều hơn một siêu lớp được gọi là **lớp con được chia sẻ** (shared subclass).
Ví dụ: **ENGINEERING_MANAGER**.

Đa Kế thừa

Đa kế thừa (multiple inheritance) là hình thức kế thừa trong đó lớp con được chia sẻ kế thừa trực tiếp các thuộc tính và các mối quan hệ từ nhiều siêu lớp.

Hình thức đa kế thừa xuất hiện trong cấu trúc lưới.

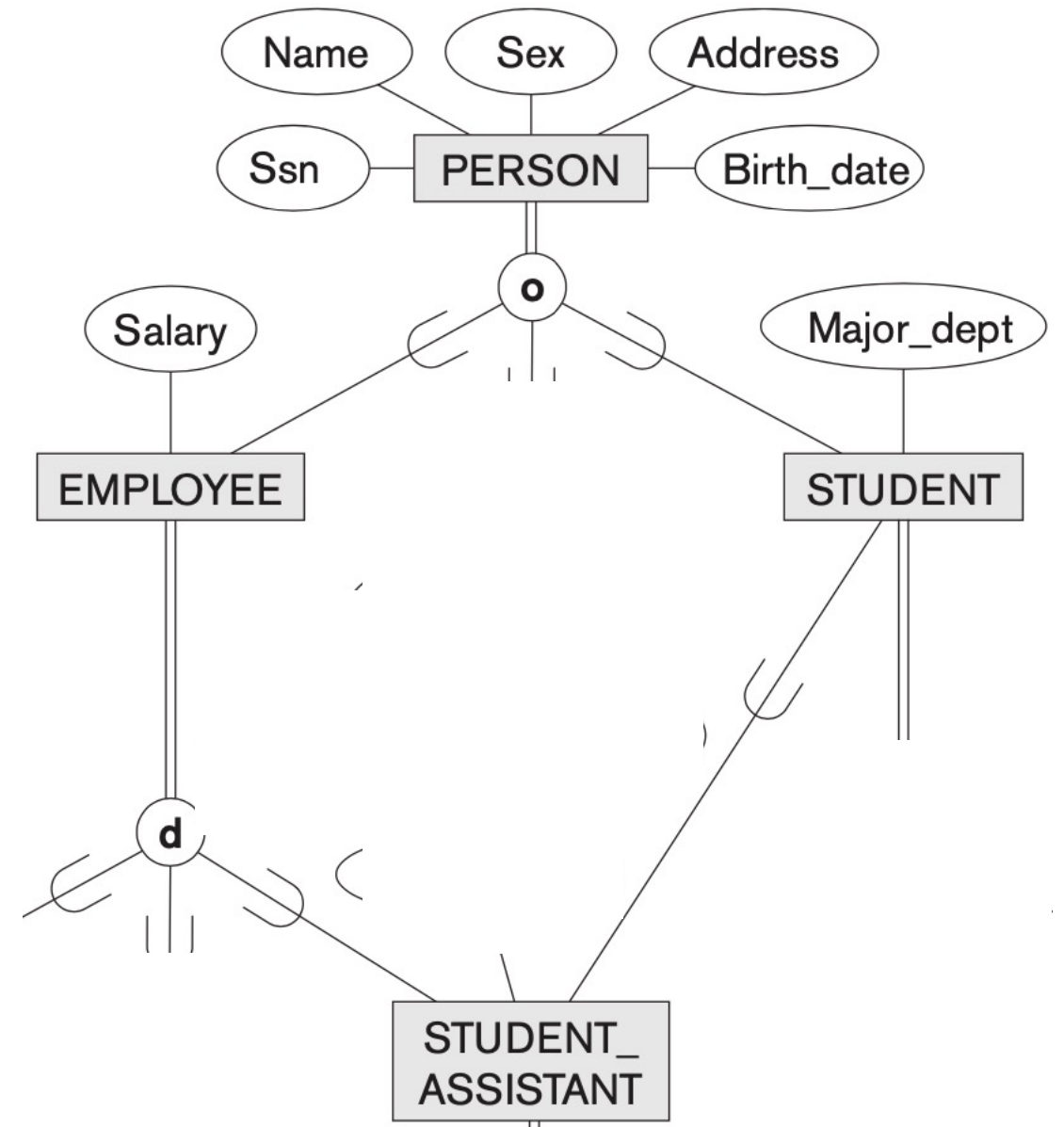
Ví dụ: Lớp con **ENGINEERING_MANAGER**

- Phân biệt với **kế thừa đơn lẻ** (single inheritance) trong cấu trúc phân cấp.

Luật giới hạn thuộc tính trong đa kế thừa

Nếu một thuộc tính (hoặc mối quan hệ) từ trong cùng một siêu lớp được kế thừa hơn một lần qua các luồng kế thừa khác nhau trong lưới chuyên môn hóa, thì thuộc tính (hoặc mối quan hệ) này chỉ được giữ lại một lần trong lớp con chia sẻ.

Ví dụ: Các thuộc tính của PERSON chỉ được kế thừa một lần tại STUDENT_ASSISTANT.



Chuyên môn hóa và Tổng quát hóa trong tinh chỉnh Lược đồ Khái niệm

- Tinh chỉnh khái niệm từ trên xuống (Top-down conceptual refinement)
- Tổng hợp khái niệm từ dưới lên (Bottom-up conceptual synthesis)

Mô hình hóa các kiểu UNION
bằng các danh mục phân loại

Định nghĩa

Đôi khi cần phải biểu diễn một tập hợp các thực thể từ các kiểu thực thể khác nhau.

Định nghĩa:

Kiểu hợp (union type) là một lớp con biểu diễn một tập hợp các thực thể là tập hợp con của phép hợp UNION của các kiểu thực thể phân biệt.

Một kiểu hợp còn được gọi là một **danh mục** (category).

Ví dụ

Cho ba kiểu thực thể: **PERSON**, **BANK**, và **COMPANY**.

Đặc tả: “Trong CSDL đăng ký xe cơ giới, chủ sở hữu xe có thể là một cá nhân, một ngân hàng (đang giữ thế chấp xe), hoặc một công ty.”

Yêu cầu: Tạo ra một lớp mô tả vai trò chủ sở hữu xe cơ giới chứa các thực thể của ba hình thức sở hữu trên.

Giải pháp: Tạo một danh mục (kiểu hợp) tên **OWNER** là lớp con của phép UNION của ba tập hợp **COMPANY**, **BANK**, **PERSON**.

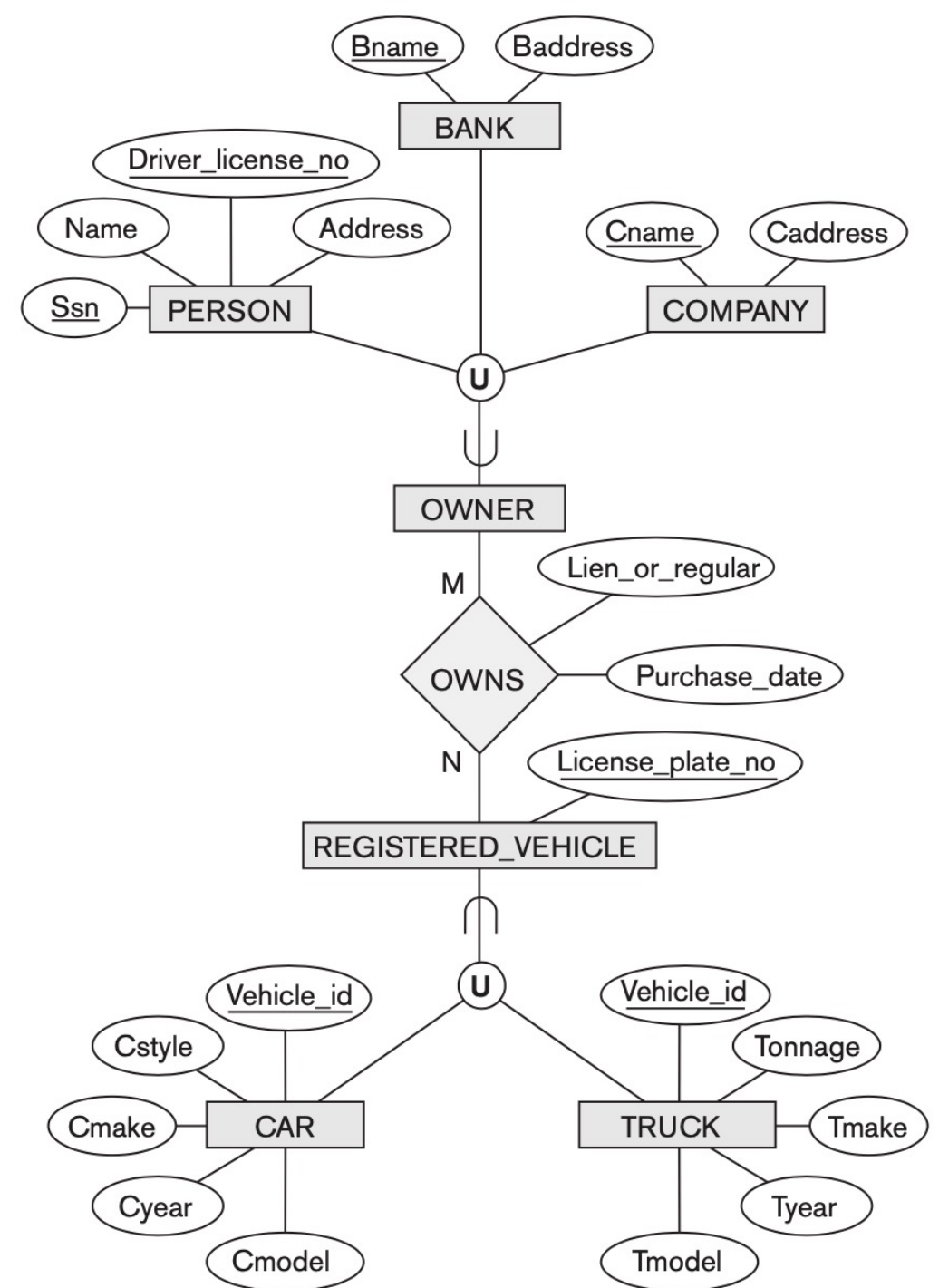
Giải pháp: mô hình EER

Ký hiệu:

- U: phép hợp UNION
- | : nối từ biểu tượng lớp con đến phép hợp.

Trong ví dụ có hai danh mục:

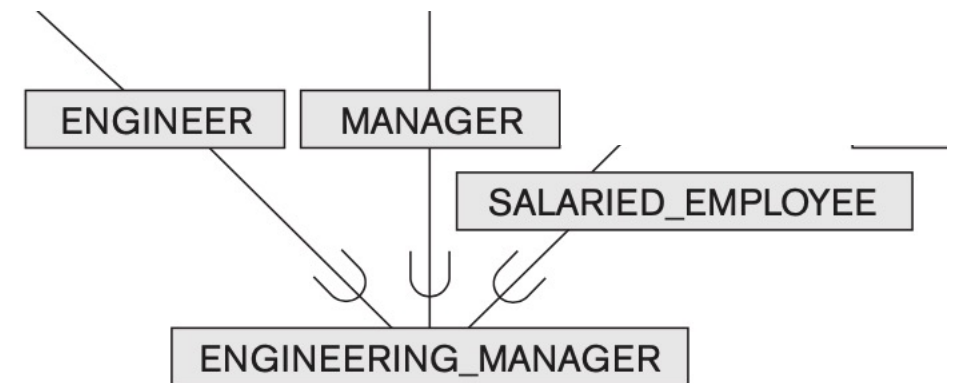
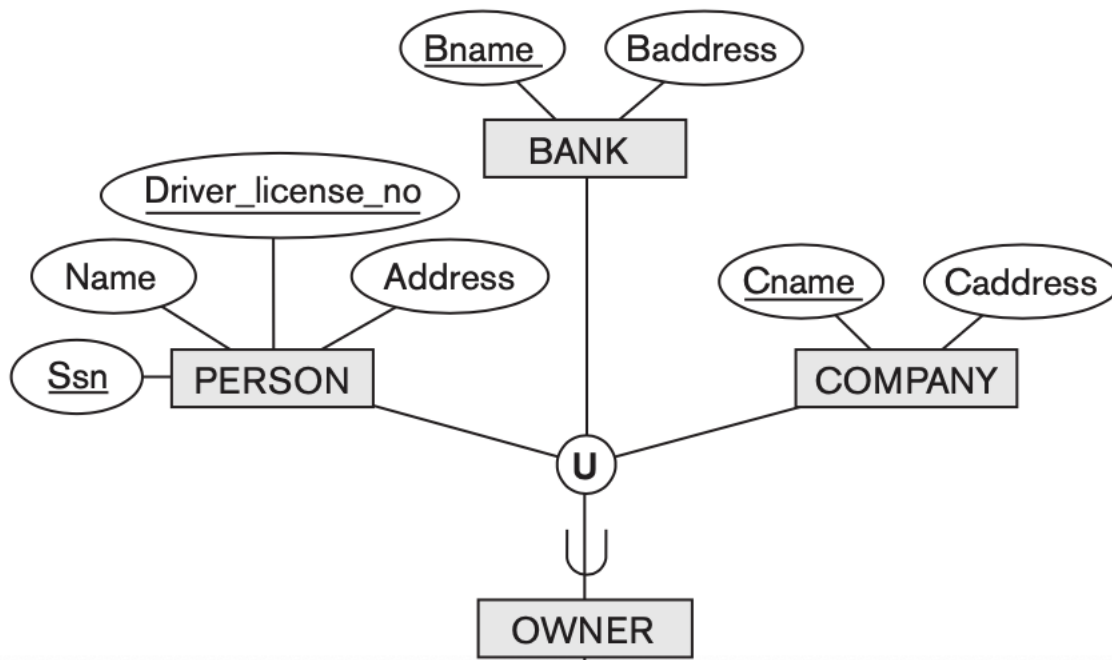
- OWNER
- REGISTER_VEHICLE



So sánh lớp Danh mục với lớp con Đa kế thừa

Yêu cầu: Hãy diễn giải ý nghĩa và sự tồn tại của một thực thể của lớp **OWNER** và một thực thể của lớp **ENGINEERING_MANAGER**.

Chú ý: Một lớp con là danh mục kế thừa *có chọn lọc* các thuộc tính từ các siêu lớp trong phép hợp; trong khi lớp con được chia sẻ kế thừa *toàn bộ* các thuộc tính từ tất cả siêu lớp của nó.



So sánh lớp Danh mục với lớp Tổng quát hóa

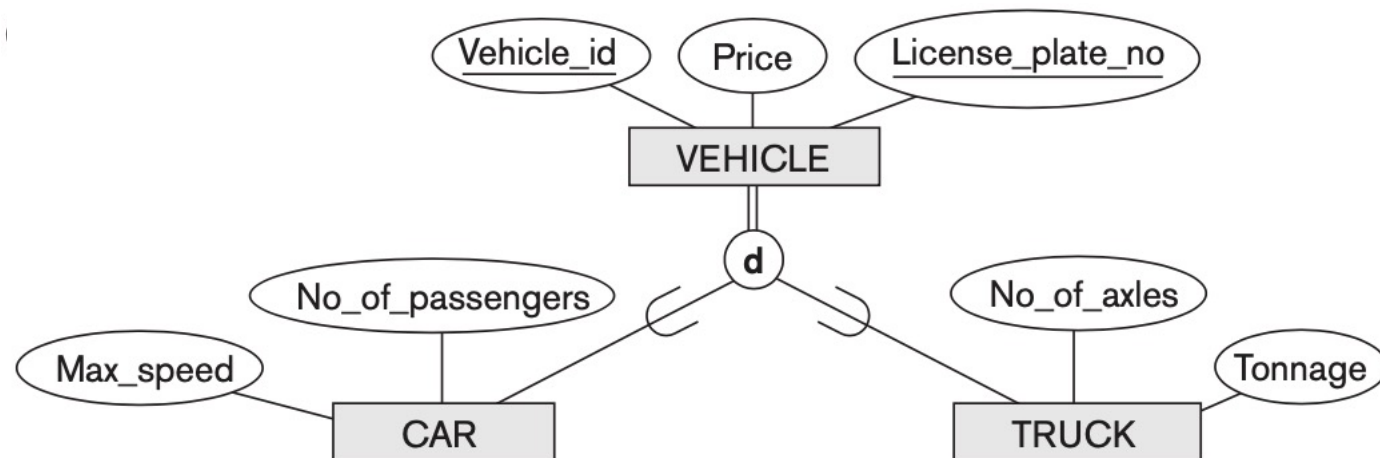
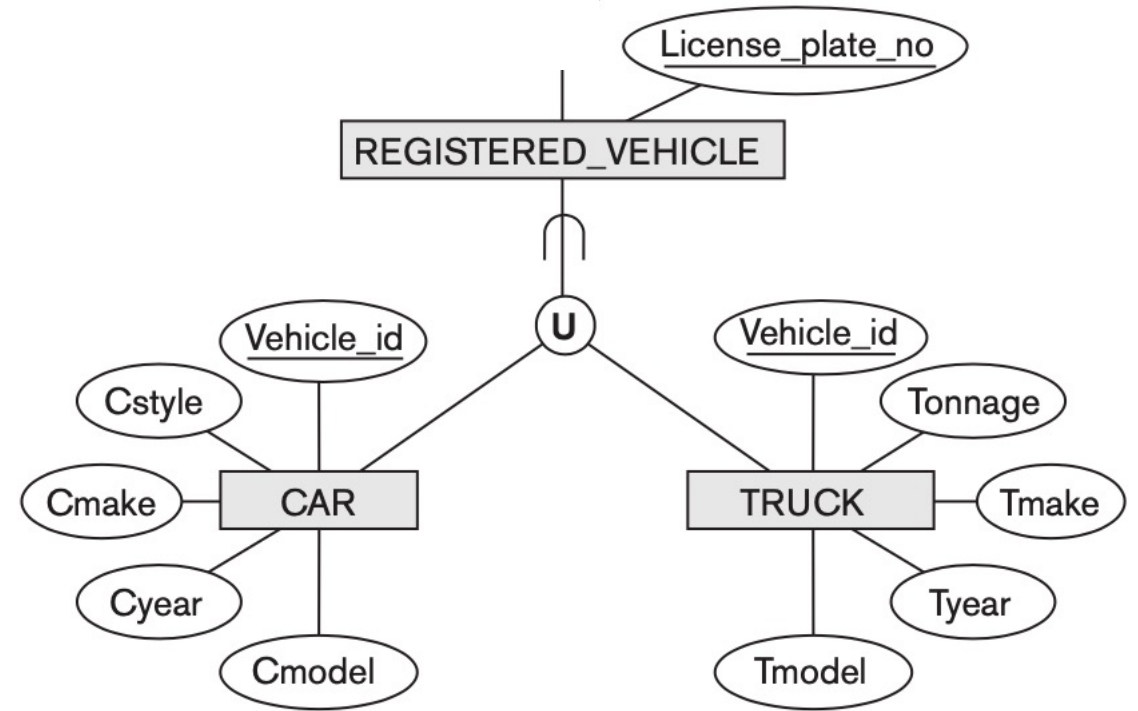
Yêu cầu: Hãy diễn giải ý nghĩa và sự tồn tại của lớp danh mục **REGISTERED_VEHICLE** và của siêu lớp được tổng quát hóa **VEHICLE**.

Một lớp chuyên môn hóa hoặc tổng quát hóa từng phần được phép chứa thêm các loại thực thể khác.

Ví dụ: **VEHICLE** có thể chứa các thực thể “xe máy”.

Một lớp danh mục được ngầm hiểu rằng chỉ các các thực thể thuộc các siêu lớp trong phép hợp mới có thể là thành viên của nó

Ví dụ: chỉ có xe hơi và xe tải mới trở thành thành viên của **REGISTERED_VEHICLE**.



Danh mục toàn phần, danh mục từng phần

Danh mục toàn phần giữ kết quả hợp nhất của *tất cả* thực thể có trong siêu lớp của danh mục.

Ký hiệu:

|| : nối danh mục đến phép hợp

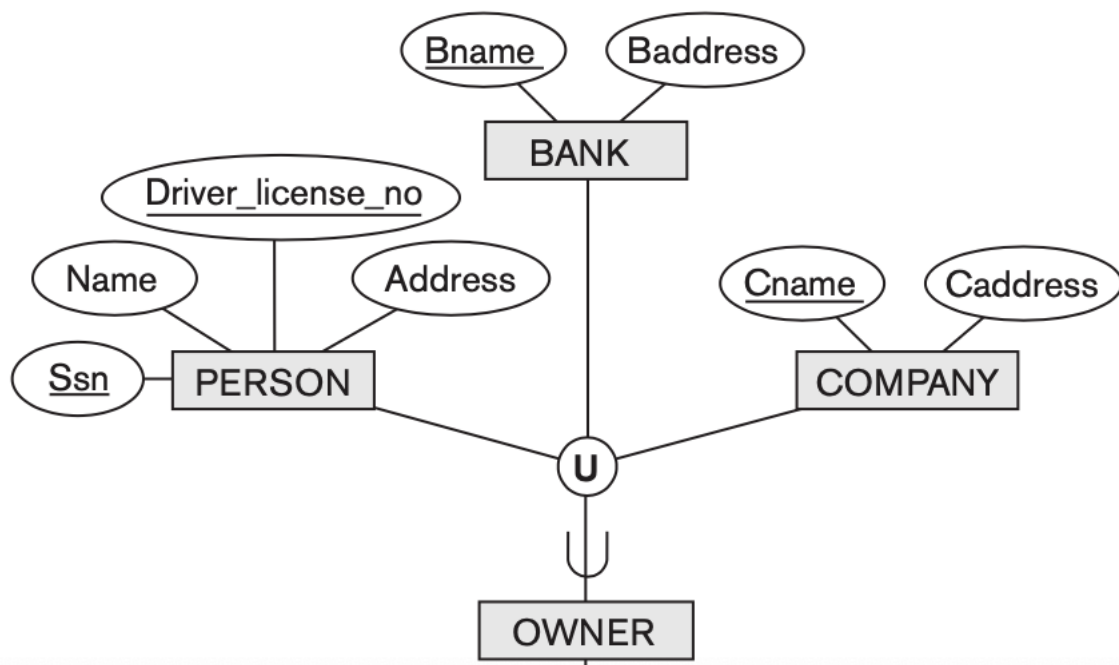
Danh mục từng phần chỉ giữ *một tập hợp con* của kết quả hợp.

Ký hiệu:

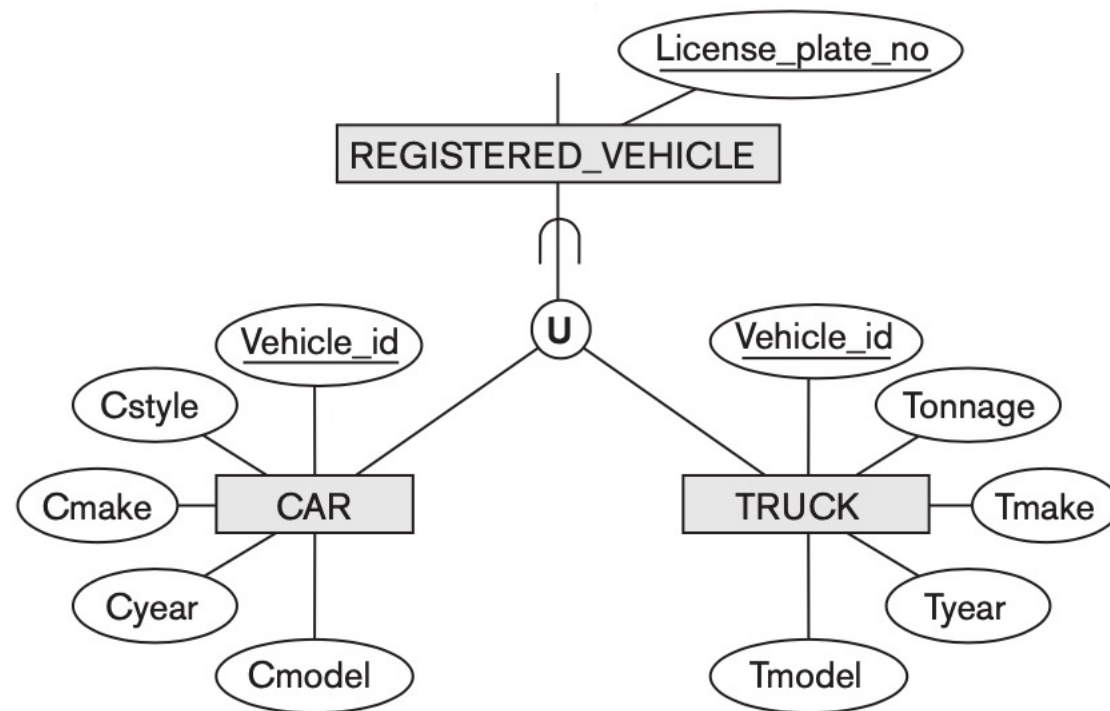
| : nối từ danh mục đến phép hợp.

Thuộc tính khóa trong lớp danh mục

Các siêu lớp có thuộc tính khóa khác nhau.



Các siêu lớp có thuộc tính khóa giống nhau.



Dùng Chuyên môn hóa/Tổng quát hóa hay dùng Danh mục phân loại?

Nếu một danh mục là toàn phần thì danh mục có thể được biểu diễn thay thế bằng một chuyên môn hóa toàn phần (hay bằng một tổng quát hóa toàn phần).

- Nếu hai lớp biểu diễn cùng một kiểu thực thể và chia sẻ nhiều thuộc tính, bao gồm cả thuộc tính khóa giống nhau, thì ưu tiên dùng chuyên môn hóa/tổng quát hóa.
- Ngược lại, dùng danh mục.

Chú ý: một số phương pháp mô hình hóa không có kiểu union.

Lược đồ EER cho UNIVERSITY

Mục đích của CSDL UNIVERSITY (được mở rộng)

- CSDL theo dõi sinh viên và chuyên ngành, bảng điểm, đăng ký và các khóa học của trường đại học.
- CSDL theo dõi các dự án nghiên cứu được tài trợ của giảng viên và sinh viên sau đại học.

Đặc tả yêu cầu (1/3)

Đối với mỗi người, cơ sở dữ liệu lưu giữ thông tin về Tên [Name], Số An sinh xã hội [SSN], Địa chỉ [Address], Giới tính [Sex] và Ngày sinh [Bdate] của người đó. Hai lớp con của loại thực thể PERSON được xác định: KHOA (FACULTY) và SINH VIÊN (STUDENT). Các thuộc tính cụ thể của KHOA là cấp bậc [Rank] gồm: trợ lý (assistant), cộng sự (associate), giảng viên thỉnh giảng (adjunct), nghiên cứu (research), thỉnh giảng (visiting), v.v.), văn phòng [Foffice], điện thoại văn phòng [Fphone] và lương [Salary].

Tất cả các thành viên khoa đều có liên quan đến các khoa mà họ trực thuộc [BELONGS] (một thành viên khoa có thể được liên kết với một số khoa, vì vậy mối quan hệ là M:N). Một thuộc tính cụ thể của SINH VIÊN là lớp theo năm học [Class] (sinh viên năm nhất (freshman) = 1, sinh viên năm hai (sophomore) = 2, ..., sinh viên cao học (master – MS student) = 5, sinh viên Tiến sĩ (PhD student) = 6). Mỗi SINH VIÊN (STUDENT) cũng liên quan đến các khoa chính và khoa phụ (nếu biết) [MAJOR] và [MINOR], đến các phần khóa học mà sinh viên hiện đang theo học [REGISTERED] và đến các khóa học đã hoàn thành [TRANSCRIPT]. Mỗi trường hợp TRANSCRIPT bao gồm điểm mà sinh viên nhận được [Grade] trong một phần của khóa học.

Đặc tả yêu cầu (2/3)

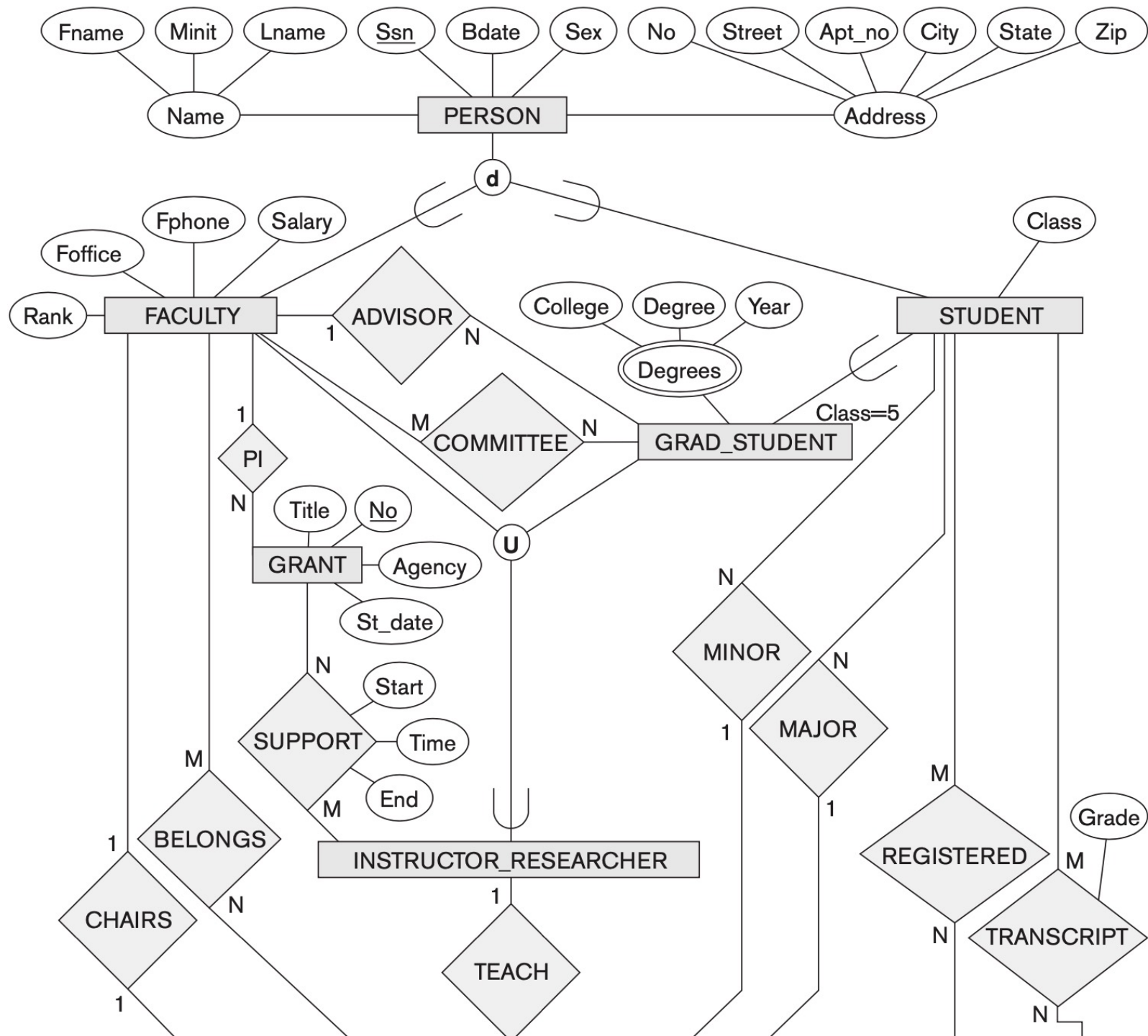
GRAD_STUDENT là một lớp con của STUDENT, với vị từ xác định (Class = 5 OR Class = 6). Đối với mỗi sinh viên sau đại học, chúng tôi lưu giữ danh sách các bằng cấp trước đây trong một thuộc tính tổng hợp, đa giá trị [Degrees]. Chúng tôi cũng liên hệ sinh viên sau đại học với cố vấn khoa [ADVISOR] và với ủy ban luận án [COMMITTEE], nếu có.

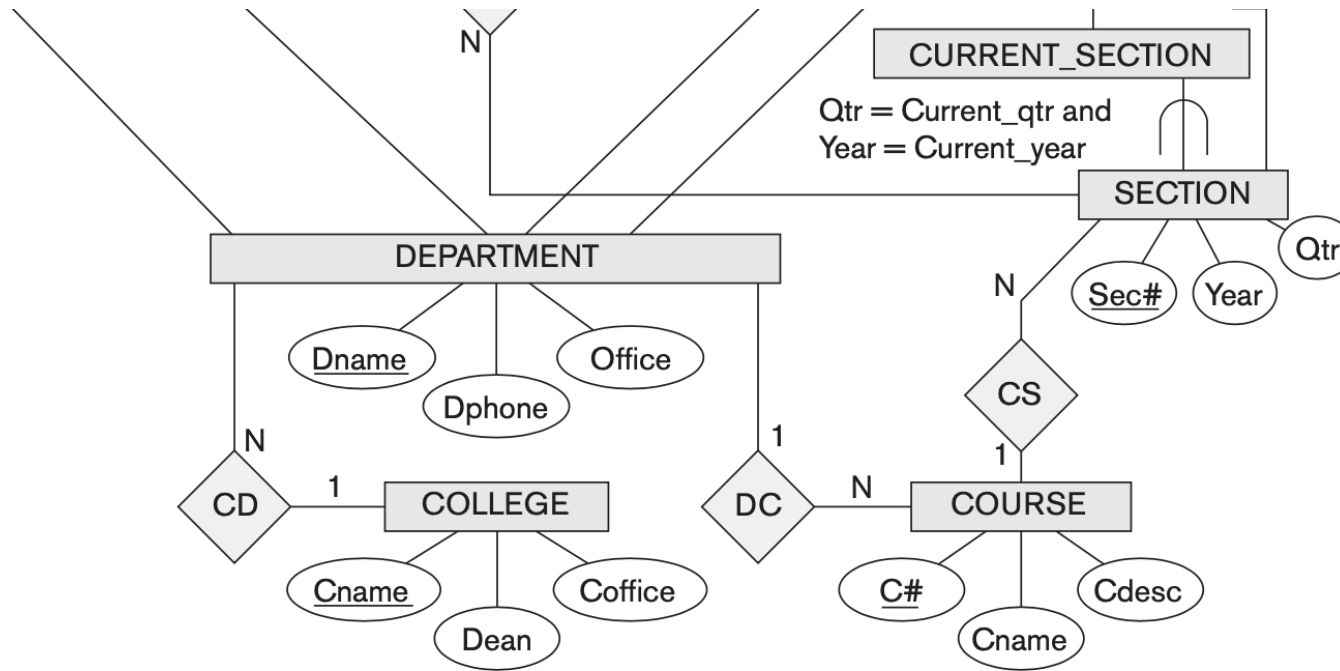
Một khoa học thuật có các thuộc tính tên khoa [Dname], điện thoại liên lạc [Dphone] và số hiệu văn phòng khoa [Office] và liên quan đến thành viên khoa là chủ tịch [CHAIRS] và trường đại học mà khoa đó trực thuộc [CD]. Mỗi trường cao đẳng có các thuộc tính tên trường [Cname], số phòng [Coffice] và tên khoa trưởng [Dean].

Đặc tả yêu cầu (3/3)

Một khóa học có các thuộc tính số khóa học [C#], tên khóa học [Cname] và mô tả khóa học [Cdesc]. Một số phần của mỗi khóa học được cung cấp, với mỗi phần có các thuộc tính số phần [Sec#] và năm và quý mà phần đó được cung cấp ([Year] và [Qtr]). Số phần xác định duy nhất từng phần. Các phần được cung cấp trong quý hiện tại năm trong một lớp con CURRENT_SECTION của SECTION, với vị từ xác định Qtr = Current_qtr và Year = Current_year. Mỗi phần có liên quan đến giảng viên đã hoặc đang giảng dạy phần đó ([TEACH]), nếu giảng viên đó có trong cơ sở dữ liệu.

Danh mục INSTRUCTOR_RESEARCHER là một tập hợp con của hợp FACULTY và GRAD_STUDENT và bao gồm tất cả các giảng viên, cũng như sinh viên sau đại học được hỗ trợ bằng việc giảng dạy hoặc nghiên cứu. Cuối cùng, loại thực thể GRANT theo dõi các khoản tài trợ nghiên cứu và hợp đồng được trao cho trường đại học. Mỗi khoản tài trợ có các thuộc tính là tiêu đề khoản tài trợ [Title], số khoản tài trợ [No], cơ quan trao [Agent] và ngày bắt đầu [St_date]. Một khoản tài trợ có liên quan đến một nhà nghiên cứu chính [PI] và tất cả các nhà nghiên cứu được hỗ trợ [SUPPORT]. Mỗi trường hợp hỗ trợ có các thuộc tính là ngày bắt đầu hỗ trợ [Start], ngày kết thúc hỗ trợ (nếu biết) [End] và tỷ lệ phần trăm thời gian dành cho dự án [Time] của nhà nghiên cứu được hỗ trợ.





Hướng dẫn thiết kế EER (1/3)

- Chỉ biểu diễn những lớp con cần thiết nhất để tránh làm lộn xộn quá mức lược đồ khái niệm.
- Nếu lớp con có ít thuộc tính (nội bộ) đặc biệt và không có các mối quan hệ cụ thể nào thì nó có thể được hợp nhất vào lớp cha. Các thuộc tính đặc biệt sẽ có giá trị NULL cho các thực thể không phải là thành viên của lớp con.
Dùng một thuộc tính đặt tên là “*kiểu*” (“*type*”) để chỉ định một thực thể có phải là thành viên của lớp con hay không.

Hướng dẫn thiết kế EER (2/3)

- Nếu tất cả lớp con của một chuyên môn hóa/tổng quát hóa có ít thuộc tính đặc biệt và không có các mối quan hệ cụ thể, chúng có thể được hợp nhất vào lớp cha và được thay thế bằng một hoặc nhiều thuộc tính “*kiểu*” (“*type*”) để xác định lớp con mà mỗi thực thể thuộc về.
- Nên tránh dùng các kiểu union và lớp danh mục trừ trường hợp bất khả kháng xảy ra trong một số tình huống thực tế.
Nếu có thể, hãy chuyển sang dùng mô hình chuyên môn hóa/tổng quát hóa.

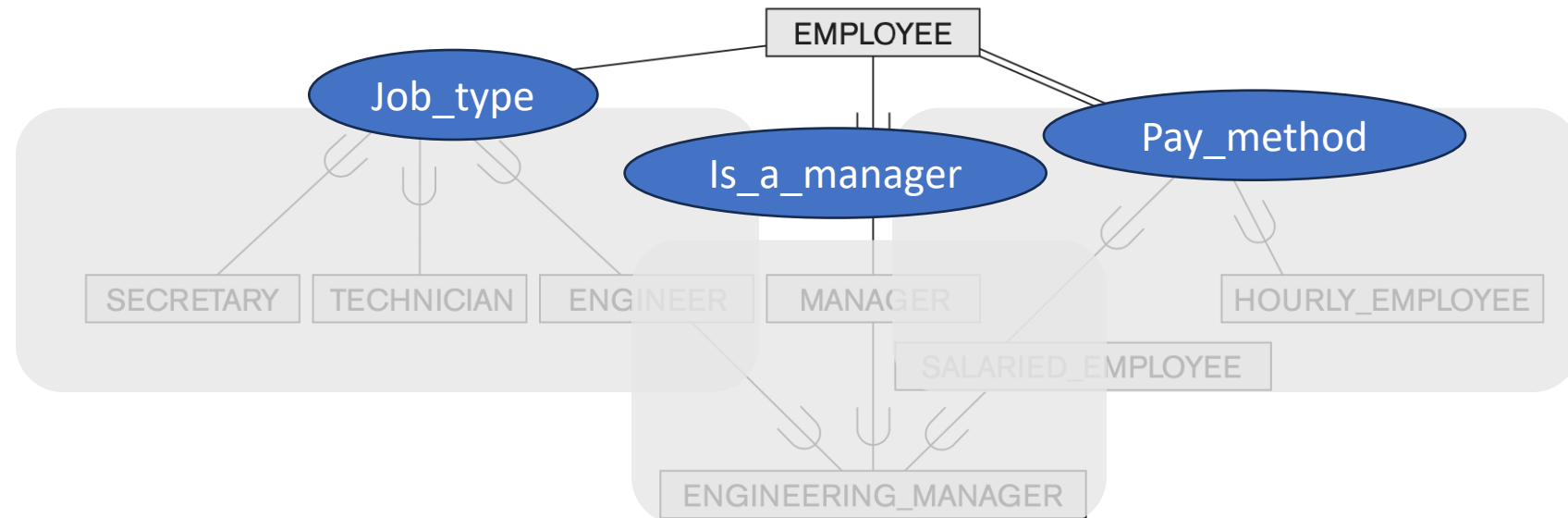
Hướng dẫn thiết kế EER (3/3)

- Việc chọn các ràng buộc rời rạc/chồng lấn và ràng buộc toàn thể/một phần đối với chuyên môn hóa/tổng quát hóa tùy thuộc vào các quy tắc quy định trong thực tế đang được mô hình hóa.
Nếu các yêu cầu không chỉ định loại ràng buộc cụ thể nào thì mặc định nên dùng ràng buộc chồng lấn và ràng buộc một phần vì các ràng buộc loại này không quy định bất kỳ một hạn chế nào lên các thành viên lớp con.

Áp dụng hướng dẫn vào thiết kế EER – Ví dụ

Trong mô hình dưới, có thể hợp nhất tất cả các lớp con vào kiểu thực thể **EMPLOYEE** và bổ sung vào **EMPLOYEE** các thuộc tính sau:

- **[Job_type]**: $\text{dom}([\text{Job_type}] = \{\text{'Secretary'}, \text{'Engineer'}, \text{'Technician'}\}$ để xác định một nhân viên thuộc về nhóm nghề nghiệp nào.
- **[Pay_method]**: $\text{dom}([\text{Pay_method}]) = \{\text{'Salaried'}, \text{'Hourly'}\}$ cho biết một nhân viên ở trong nhóm hình thức trả lương nào.
- **[Is_a_manager]**: $\text{dom}([\text{Is_a_manager}]) = \{\text{'Yes'}, \text{'No'}\}$ cho biết nhân viên có là một quản lý hay không.



Định nghĩa hình thức các khái niệm trong mô hình EER (1/4)

Một **lớp** (class) định nghĩa một kiểu thực thể và biểu diễn một tập hợp các thực thể thuộc cùng kiểu này.

Một **lớp con** (subclass) S là một lớp chứa các thực thể là tập hợp con của các thực thể trong một lớp khác, được gọi là **siêu lớp** (superclass) C .

Giữa S và C tồn tại **quan hệ cha/con** (superclass/subclass relationship) hay còn gọi là quan hệ “**là một**” (IS-A), kí hiệu: C/S .

Ta luôn có:

$$S \subseteq C$$

Định nghĩa hình thức các khái niệm trong mô hình EER (2/4)

Một **chuyên môn hóa** (specialization) $Z = \{S_1, S_2, \dots, S_n\}$ là tập hợp các lớp con có cùng siêu lớp G ; nghĩa là, G/S_i là một quan hệ cha/con với $i = 1, 2, \dots, n$.

G được gọi là **kiểu thực thể được tổng quát hóa** (generalized entity type), hoặc là **siêu lớp** của chuyên môn hóa; hoặc là **tổng quát hóa** của các lớp con $\{S_1, S_2, \dots, S_n\}$.

Z được gọi là **toàn phần** (total) nếu ta luôn có:

$$\bigcup_{i=1}^n S_i = G$$

Ngược lại, Z được gọi là **một phần** (spartial).

Z được gọi là **rời rạc** (disjoint) nếu ta luôn có:

$$S_i \cap S_j = \emptyset, \text{ với } i \neq j$$

Ngược lại, Z được gọi là **chồng lấn** (overlapping).

Định nghĩa hình thức các khái niệm trong mô hình EER (3/4)

Lớp con S của C được gọi là **xác định bằng vị từ** (predicate-defined) nếu một vị từ p trên các thuộc tính của C được sử dụng để xác định thực thể nào trong C là thành viên của S ; nghĩa là, $S = C[p]$, trong đó $C[p]$ là tập hợp các thực thể trong C thỏa mãn p .

Một lớp con không được xác định bằng vị từ được gọi là **xác định theo người dùng** (user-defined).

Một chuyên môn hóa Z (hay tổng quát hóa G) được gọi là **xác định bằng thuộc tính** (attribute-defined) nếu có một vị từ ($A = c_i$), trong đó A là một thuộc tính của G và c_i là một giá trị hằng lấy từ miền giá trị của A , được dùng để xác định tư cách thành viên trong mỗi lớp con S_i trong Z .
Nếu $c_i \neq c_j$ với $i \neq j$, và A là thuộc tính đơn trị, thì chuyên môn hóa sẽ là rời rạc.

Định nghĩa hình thức các khái niệm trong mô hình EER (4/4)

Một **danh mục** (category) T là một lớp và là một tập hợp con của phép hợp (union) của n siêu lớp $D_1, D_2, \dots, D_n, n > 1$ và được xác định như sau:

$$T \subseteq (D_1 \cup D_2 \dots \cup D_n)$$

Vị từ p_i trên các thuộc tính của D_i có thể được sử dụng để xác định các thành viên của từng D_i cũng là thành viên của T . Nếu một vị từ được xác định trên mọi D_i , ta có:

$$T = (D_1[p_1] \cup D_2[p_2] \dots \cup D_n[p_n])$$

Mở rộng định nghĩa **kiểu quan hệ** (relationship type) trong ER bằng cách cho phép lớp bất kỳ được tham gia vào một mối quan hệ. Dùng khái niệm lớp thay cho khái niệm kiểu thực thể.

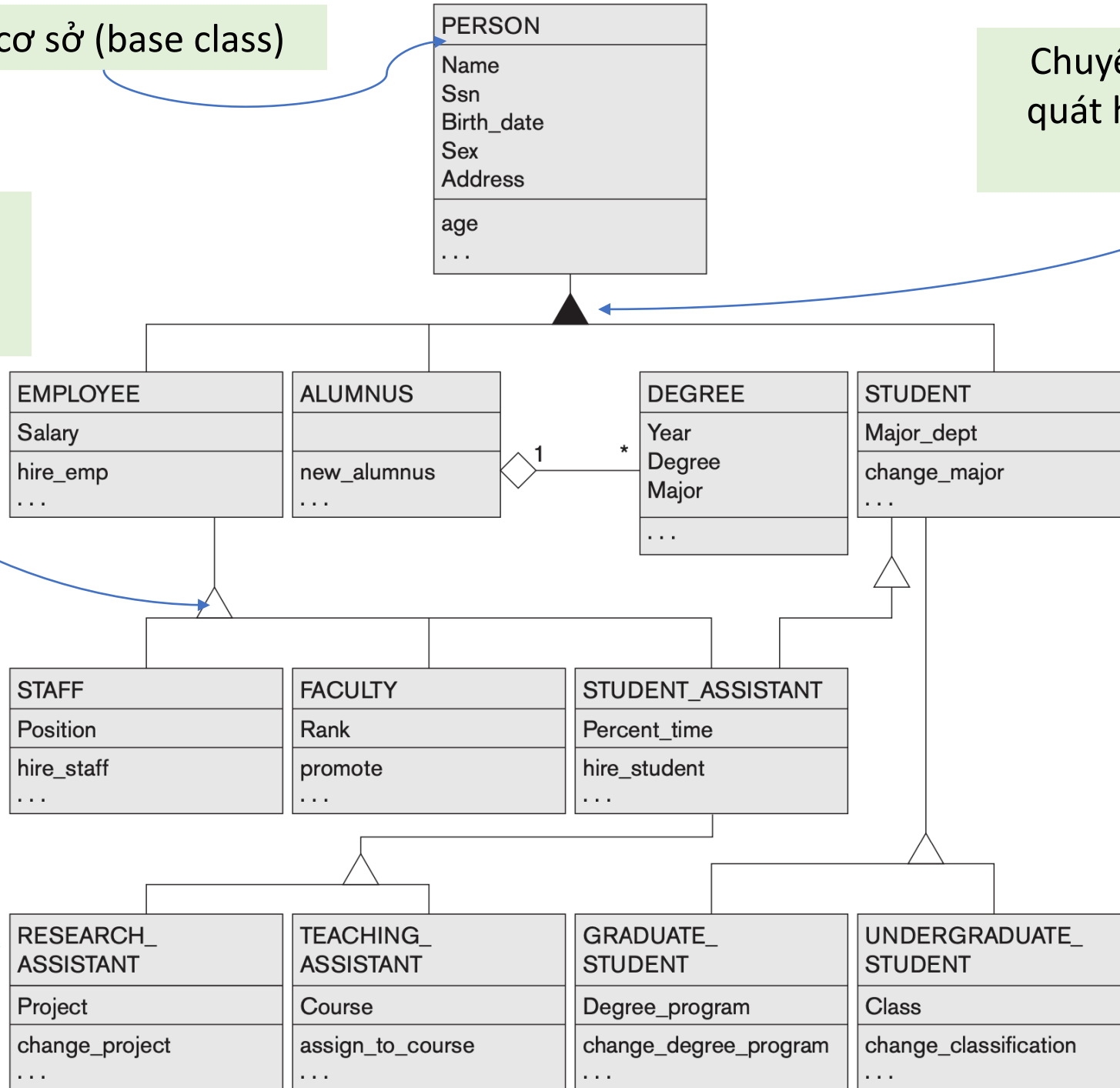
Biểu diễn Chuyên môn hóa/Tổng
quá hóa trong sơ đồ lớp UML

Lớp cơ sở (base class)

Chuyên môn hóa/Tổng quát hóa với ràng buộc chồng lấn

Chuyên môn hóa/Tổng quát hóa với ràng buộc rời rạc

Lớp lá (leaf class)



Trừu tượng hóa dữ liệu,
Biểu diễn tri thức,
Các khái niệm Ontology

Biểu diễn tri thức (KR) và Ontology

- Kỹ thuật biểu diễn tri thức (KR – Knowledge Representation) phát triển các khái niệm để mô hình hóa chính xác một số miền tri thức bằng cách tạo ra một ontology mô tả các khái niệm của miền tri thức đó và cách các khái niệm này liên quan với nhau.
- Ontology được sử dụng để lưu trữ và xử lý tri thức để rút ra suy luận, đưa ra quyết định hoặc trả lời các câu hỏi.

KR và mô hình dữ liệu ngữ nghĩa (SDM)

Mục tiêu của KR tương tự mục tiêu của các mô hình dữ liệu ngữ nghĩa (SDM - Semantic Data Model).

- Hai lĩnh vực đều sử dụng quy trình trừu tượng hóa để xác định các thuộc tính chung và các khía cạnh quan trọng của mô hình thực tế và loại bỏ các khác biệt không đáng kể cũng như các chi tiết không quan trọng.
- Đều cung cấp các khái niệm, mối quan hệ, ràng buộc, phép tính và ngôn ngữ định nghĩa dữ liệu và biểu diễn tri thức.
- KR thường có phạm vi rộng hơn SDM. Các dạng kiến thức khác nhau, chẳng hạn như các quy tắc (được sử dụng trong suy luận, diễn dịch và tìm kiếm), kiến thức không đầy đủ và mặc định, kiến thức về thời gian và không gian, được biểu diễn trong các lược đồ KR.

KR và mô hình dữ liệu ngữ nghĩa (SDM)

- Các lược đồ KR bao gồm các cơ chế lý luận suy ra các sự kiện bổ sung từ các sự kiện được lưu trữ trong cơ sở dữ liệu. Do đó, trong khi hầu hết các hệ thống cơ sở dữ liệu hiện tại chỉ giới hạn ở việc trả lời các truy vấn trực tiếp, thì các hệ thống dựa trên kiến thức sử dụng các lược đồ KR có thể trả lời các truy vấn liên quan đến suy luận trên dữ liệu được lưu trữ.
- Trong khi hầu hết các mô hình dữ liệu tập trung vào việc biểu diễn các lược đồ cơ sở dữ liệu hoặc siêu tri thức (meta-knowledge), các lược đồ KR thường trộn lẫn các lược đồ với chính các thể hiện để cung cấp tính linh hoạt trong việc biểu diễn các ngoại lệ. Điều này thường dẫn đến tình trạng kém hiệu quả khi các lược đồ KR này được triển khai, đặc biệt là khi so sánh với cơ sở dữ liệu và khi cần lưu trữ một lượng lớn dữ liệu có cấu trúc (là các sự kiện).

Các khái niệm trừu tượng trong SDM và KR

1. Sự phân loại và khởi tạo,
2. Định danh,
3. Sự chuyên môn hóa và tổng quát hóa,
4. Tổng hợp và liên kết.

(1) Sự phân loại và khởi tạo.

Quy trình **phân loại** (classification) thực hiện việc gán một cách có hệ thống các đối tượng/thực thể tương đồng (về các thuộc tính, các mối quan hệ, các ràng buộc) vào các kiểu lớp đối tượng/lớp thực thể.

- Mô tả (trong DB) hoặc suy diễn (trong KR) ở mức độ lớp thay vì cho từng cá thể đối tượng.
- Làm đơn giản hóa tiến trình khám phá thuộc tính của đối tượng

Quy trình **khởi tạo** (instatiation) gồm việc tạo ra và kiểm tra cụ thể từng đối tượng riêng biệt trong một lớp.

- Một thể hiện đối tượng liên quan đến lớp đối tượng qua mối quan hệ IS-A-INSTANCE-OF hoặc IS-A-MEMBER-OF.

Phân loại trong Mô hình EER (nhắc lại)

Trong mô hình EER, các thực thể được phân loại thành các kiểu thực thể căn cứ vào các thuộc tính và các mối quan hệ cơ sở của chúng.

Các thực thể tiếp tục được phân loại vào các lớp con và các danh mục dựa trên các đặc tính tương đồng hoặc khác biệt (dạng ngoại lệ) giữa chúng.

Các thể hiện mối quan hệ được phân loại thành các kiểu quan hệ.

Phân loại trong Mô hình KR

Dùng nhiều lược đồ phân lớp, trong đó mỗi lớp là một *hiện thực* (*instance*) của một lớp khác (được gọi là **meta-class** – siêu lớp).

Cấu trúc lớp của mô hình KR thể hiện các mối quan hệ lớp cha/lớp con, và cả các mối quan hệ giữa lớp/hiện thực lớp (điều này không thực hiện được trong EER).

(2) Định danh

Định danh (identification) là quy trình trừu tượng hóa trong đó các lớp và các đối tượng được nhận dạng duy nhất thông qua một số mã định danh.

- Định danh đối tượng dùng để phân biệt các đối tượng riêng biệt của một lớp.

Giúp xác định nhiều biểu hiện trong CSDL của cùng một đối tượng trong thế giới thực với các phép tham chiếu chéo giữa các bộ dữ liệu được xác lập khi thiết kế CSDL.

Ví dụ: Thông tin của cùng một người được thể hiện trong hai bộ <'Matthew Clarke', '610618', '376-9821'> trong quan hệ PERSON và bộ <'301-54-0836', 'CS', 3.8> trong quan hệ STUDENT.

(3) Chuyên môn hóa và Tổng quát hóa

Chuyên môn hóa là quá trình phân loại một lớp các đối tượng thành các lớp con chuyên biệt.

- Tinh chế khái niệm.

Tổng quát hóa là quá trình tổng quát nhiều lớp thành một lớp trừu tượng cấp trên, có chứa các đối tượng của tất cả các lớp này.

- Tổng hợp khái niệm.

Được biểu diễn bằng các quan hệ “là một” - **IS-A**.

(4) Tổng hợp và liên kết

Tổng hợp (aggregation) là khái niệm trừu tượng thể hiện sự tạo thành các đối tượng tổng hợp (composite objects) từ các đối tượng thành phần.

Trong mô hình EER:

1. Tổng hợp các giá trị thuộc tính của một đối tượng để tạo thành toàn bộ đối tượng.
2. Biểu diễn một mối quan hệ tổng hợp qua một mối quan hệ thông thường.
3. Kết hợp các đối tượng có liên quan với nhau để tạo thành một đối tượng tổng hợp ở mức cao hơn.
Các mối quan hệ **IS-A-PART-OF** hay ngược lại, là quan hệ **IS-A-COMPONENT-OF** giữa các đối tượng gốc và đối tượng tổng hợp.

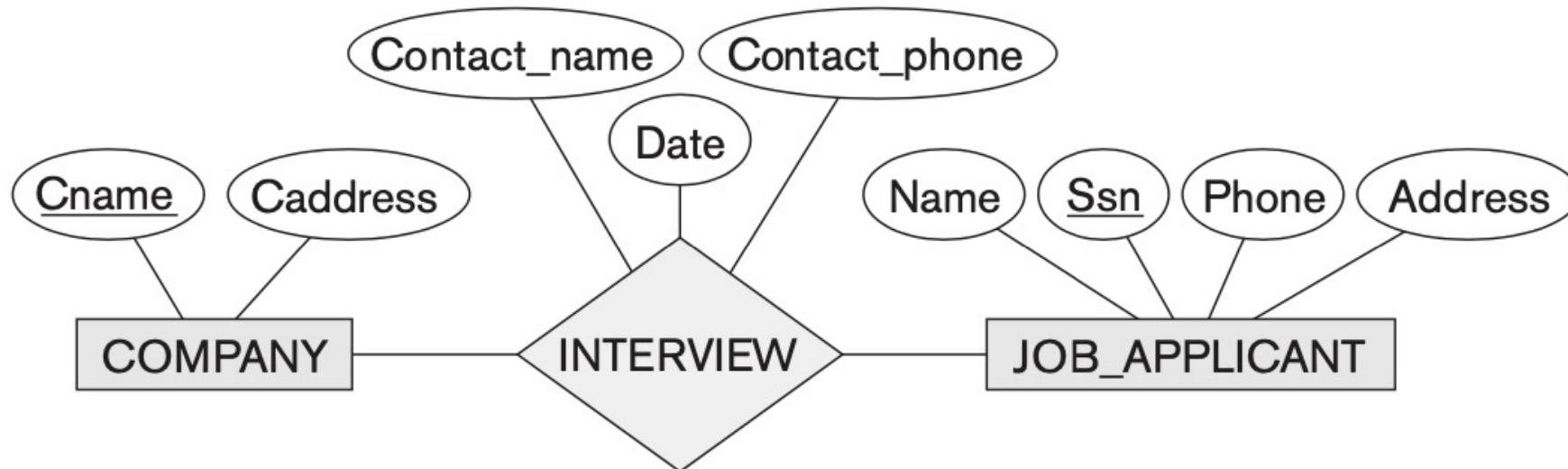
Liên kết (association) là khái niệm trừu tượng được sử dụng để liên kết các đối tượng từ nhiều lớp độc lập với nhau. Được gọi là mối quan hệ **IS-ASSOCIATED-WITH**.

Ví dụ các cách dùng khác nhau của tổng hợp (1/5)

Đặc tả: “Lưu trữ thông tin các buổi phỏng vấn của các ứng viên xin việc tại nhiều công ty khác nhau.”

Một số cuộc phỏng vấn có thể dẫn đến việc ứng viên có thể nhận được lời mời làm việc, trong khi các cuộc phỏng vấn khác thì không → thêm quan hệ JOB_OFFER.

(a)

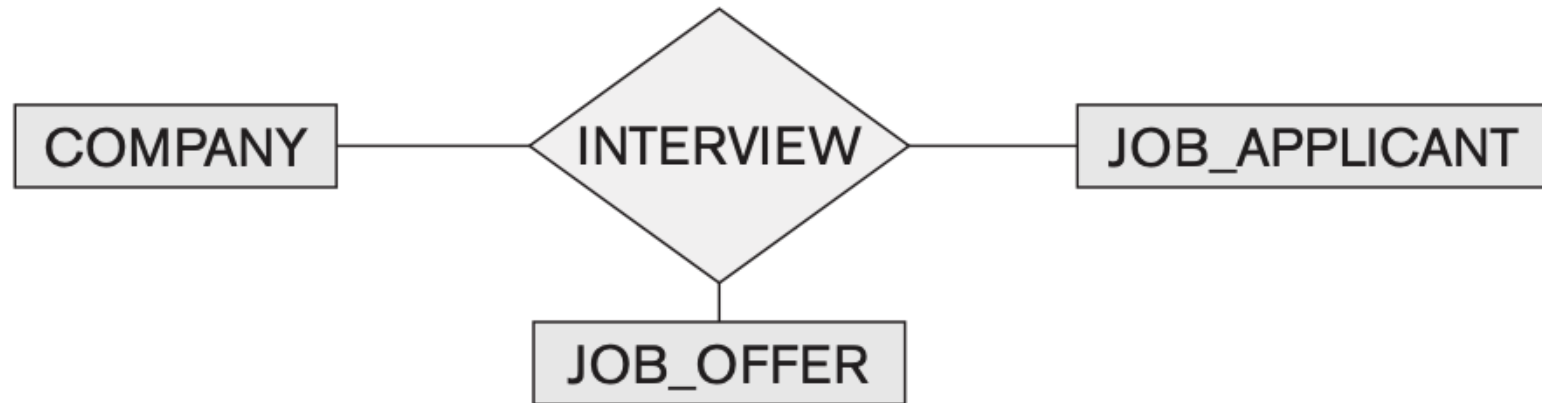


Ví dụ các cách dùng khác nhau của tổng hợp (2/5)

(b) Thêm lớp JOB_OFFER và biến INTERVIEW thành quan hệ bậc 3.

Cách này **KHÔNG ĐÚNG**. Lý do: ER này thể hiện một buổi phỏng vấn phải dẫn đến một lời mời nhận việc(!!)

(b)

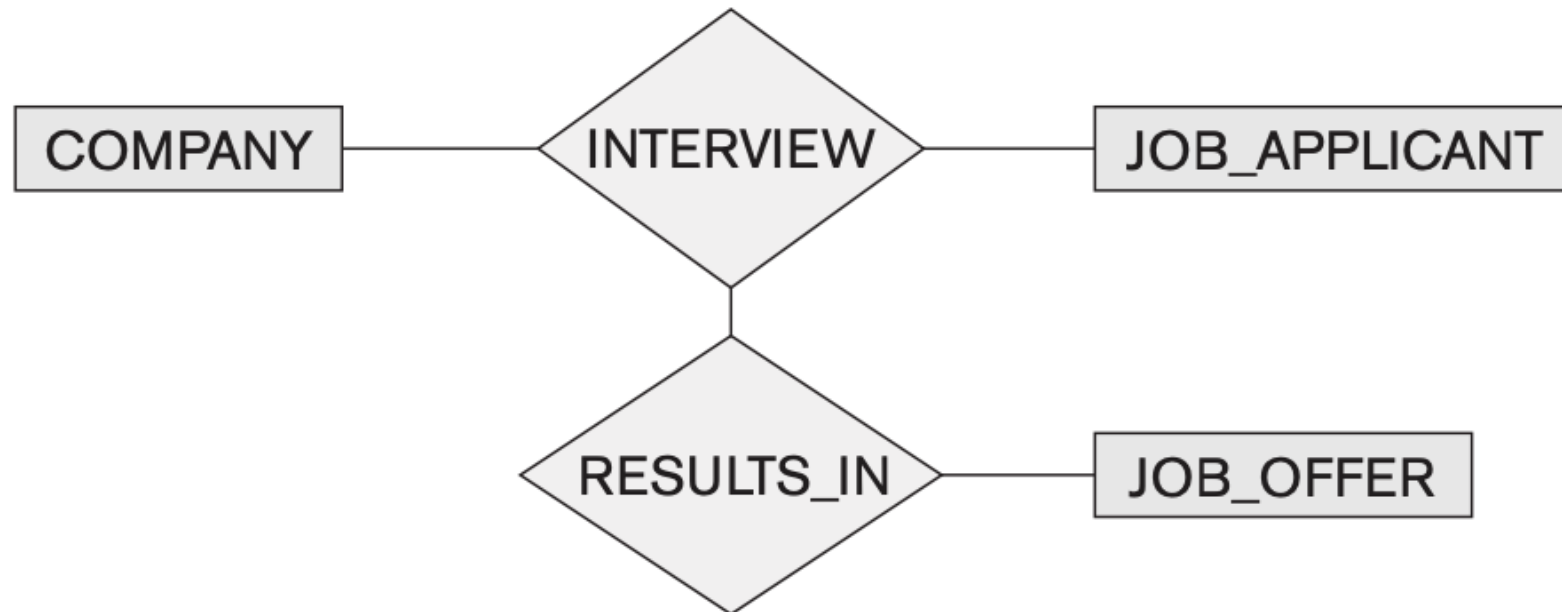


Ví dụ các cách dùng khác nhau của tổng hợp (3/5)

(c) Đưa thêm một quan hệ RESULTS_IN kết với INTERVIEW.

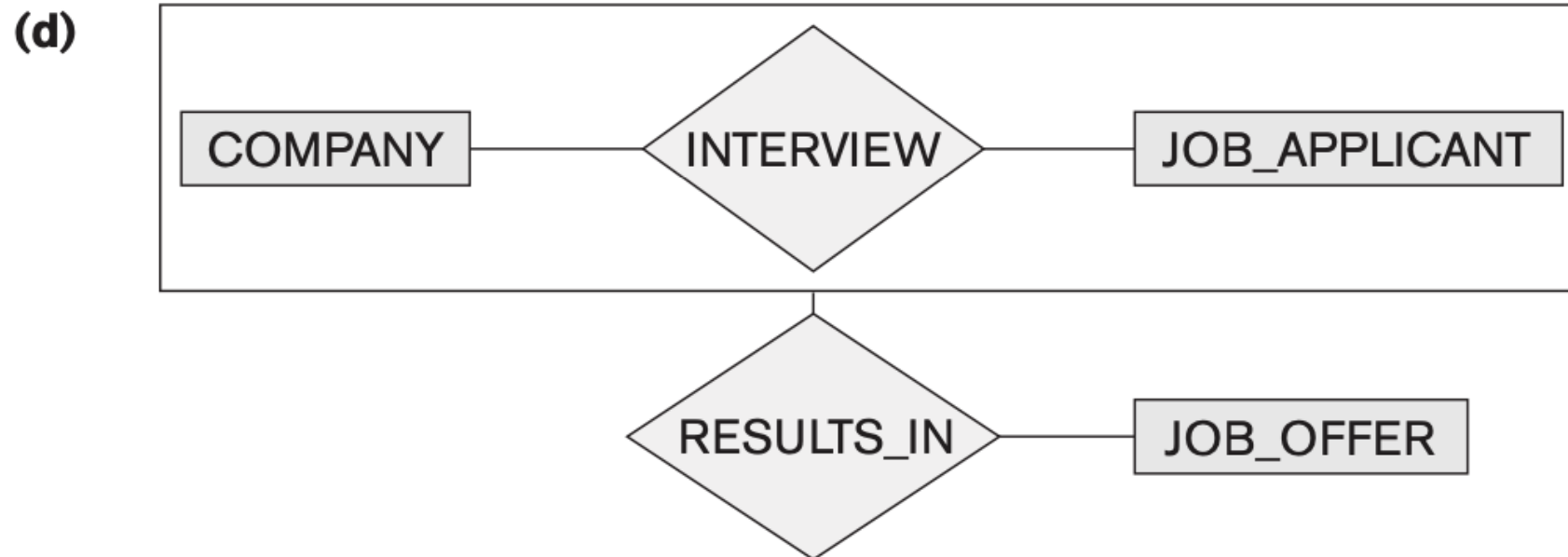
Cách này **KHÔNG THỰC HIỆN ĐƯỢC** được do vi phạm nguyên tắc trong ER mỗi quan hệ không được xuất hiện giữa các mối quan hệ.

(c)



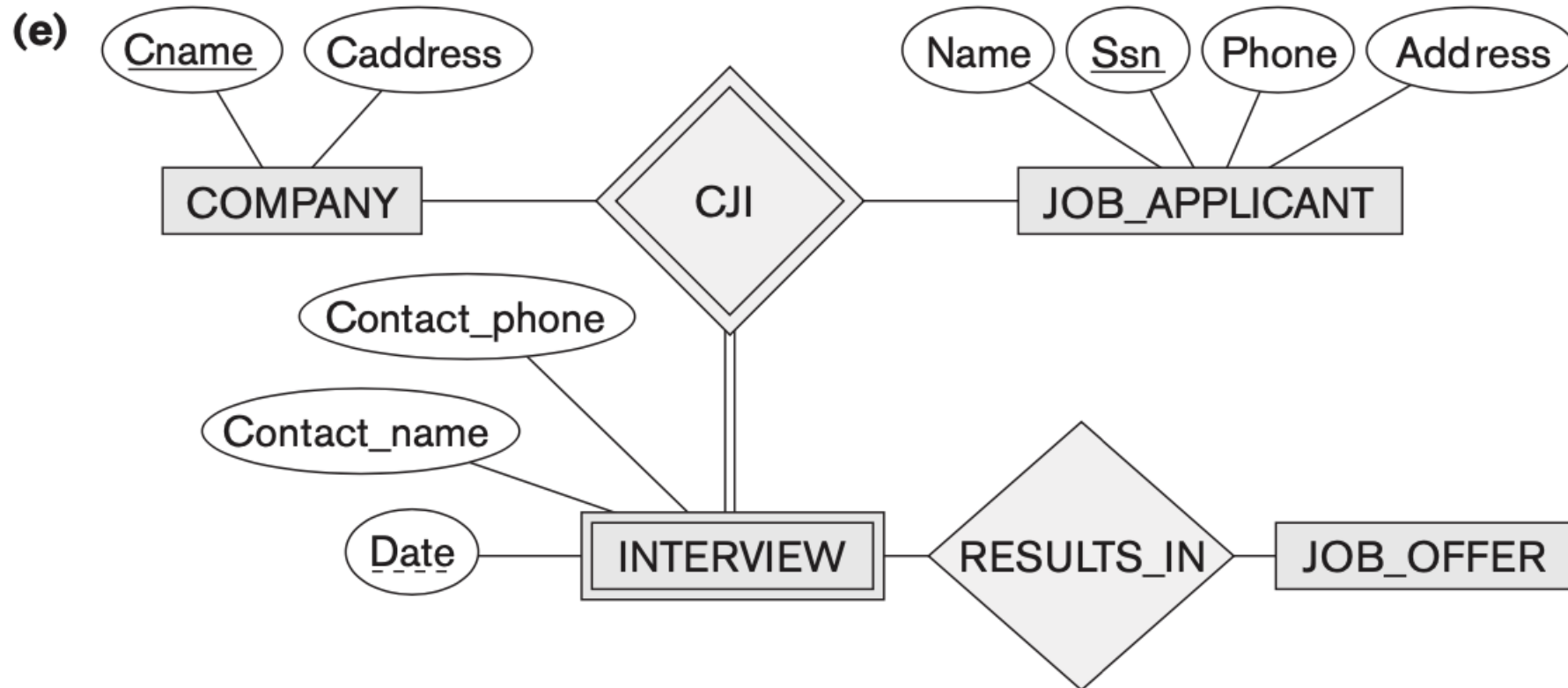
Ví dụ các cách dùng khác nhau của tổng hợp (4/5)

(d) Tạo một lớp tổng hợp bao gồm COMPANY, JOB_APPLICANT và INTERVIEW. Sau đó liên kết với JOB_OFFER thông qua RESULTS_IN.



Ví dụ các cách dùng khác nhau của tổng hợp (5/5)

Đáp án: (e) tạo thực thể yếu INTERVIEW để liên kết với JOB_OFFER qua RESULTS_IN.



Mạng ngữ nghĩa

Nhiều thông tin được lưu giữ dưới dạng các **tài liệu** (documents) không có cấu trúc như các thông tin biểu diễn được bằng CSDL.

Mạng ngữ nghĩa hình thành các mô hình KR phổ quát để trao đổi và tìm kiếm thông tin có ý nghĩa giữa các hệ thống máy.

Ontology

Bản thể học (ontology) bắt nguồn từ triết học và siêu hình học (metaphysics), chỉ các khái niệm khởi nguyên của sự vật, bao gồm các khái niệm trừu tượng về bản thể, sự hiểu biết, bản chất, nguyên nhân, danh tính, thời gian và không gian.

Ontology là một đặc tả của một khái niệm.

Khái niệm hóa (conceptualization) là tập hợp gồm các khái niệm và các mối quan hệ được sử dụng để diễn giải một phần của thực tế hay tri thức là mối quan tâm của cộng đồng.

Đặc tả (specification) chỉ ngôn ngữ và từ điển thuật ngữ được sử dụng để đặc tả sự khái niệm hóa.

Một số hình dung về Ontology

- Là một từ điển chuyên đề (thesaurus)
- Là một sự phân loại (taxonomy)
- Là một lược đồ CSDL chi tiết mô tả các khái niệm (thực thể và thuộc tính) và các mối quan hệ của thế giới thu nhỏ từ thực tế,
- Là lý thuyết logic dùng khái niệm logic toán học để xác định các khái niệm và quan hệ giữa chúng.

Ontology vs. DB

CSDL giới hạn ở mức mô tả một tập con của thế giới thực để phục vụ cho mục đích lưu trữ và quản lý dữ liệu.

Ontology có tính khái quát hơn CSDL ở chỗ nó cố gắng mô tả một phần của thực tế hoặc một lĩnh vực được quan tâm (vd, thuật ngữ y tế, các ứng dụng thương mại điện tử, thể thao, v.v...) một cách đầy đủ nhất có thể.

Hết phần này

Bài tập chương 4.