

Chương 3

MÔ HÌNH LIÊN KẾT THỰC THỂ MỞ RỘNG VÀ QUI TẮC NGHIỆP VỤ

(Enhanced Entity Relationship Model -EER)





Nội dung

- Nhắc lại ERD
- Mô hình ERR
 - Siêu kiểu và kiểu con
 - Chuyên biệt hóa và tổng quát hóa
 - Các loại ràng buộc trong mối liên kết
- Quy tắc nghiệp vụ
 - Phân loại
 - Biểu diễn qui tắc nghiệp vụ



Mô hình liên kết thực thể mở rộng – mô hình EER Enhanced Entity Relationship model

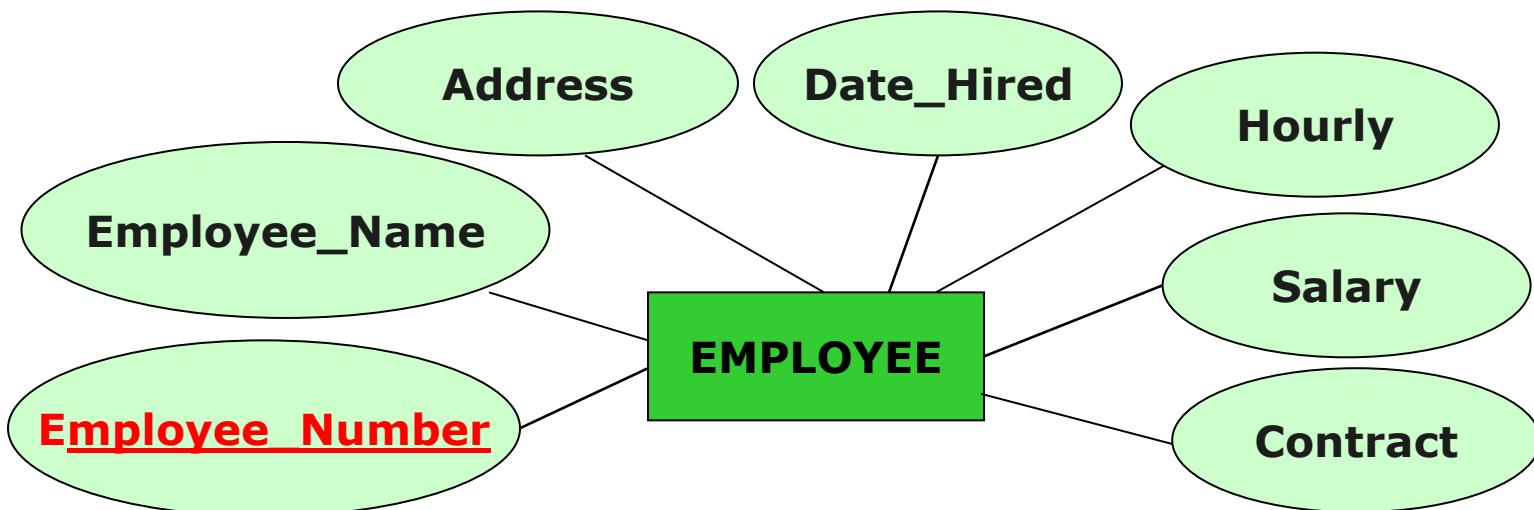
- Thực tế: yêu cầu nghiệp vụ của các tổ chức ngày càng phức tạp hơn
 - ➔ Mô hình ER cơ bản không đủ cấu trúc để diễn tả những hệ thống thông tin phức tạp
- Cân mô hình mở rộng để
 - Diễn tả được các quy tắc nghiệp vụ (business rules) phức tạp
 - Tạo được những kiểu thực thể tổng quát hơn (siêu kiểu)



Mô hình liên kết thực thể mở rộng – mô hình EER

Enhanced Entity Relationship model

- Ví dụ: một công ty có 3 loại nhân viên khác nhau: làm theo giờ, theo tháng và lương theo hợp đồng. Thể hiện quy tắc nghiệp vụ này trên ER như thế nào??
 - Cách 1: Tạo 1 kiểu thực thể EMPLOYEE có 3 thuộc tính HOURLY, SALARY, CONTRACT → mỗi thực thể chỉ có giá trị thuộc 1 trong 3 thuộc tính trên, 2 thuộc tính còn lại để trống





Mô hình liên kết thực thể mở rộng – mô hình EER

Enhanced Entity Relationship model

- Ví dụ: một công ty có 3 loại nhân viên khác nhau: làm theo giờ, theo tháng và lương theo hợp đồng. Thể hiện quy tắc nghiệp vụ này trên ER như thế nào??

→ Cách 1: Tạo 1 kiểu thực thể EMPLOYEE có 3 thuộc tính HOURLY, SALARY, CONTRACT → mỗi thực thể chỉ có giá trị thuộc 1 trong 3 thuộc tính trên, 2 thuộc tính còn lại để trống

<u>Number</u>	Name	Address	Date_Hired	Hourly	Salary	Contract
1	Lan	1 Lê Lợi	2/3/2000		600000	
2	Minh	1 ND	2/8/2016	4		
3	Hà	2 NVB	12/1/2018			1200000
4	Nga	3 QT	2/3/2015		500000	

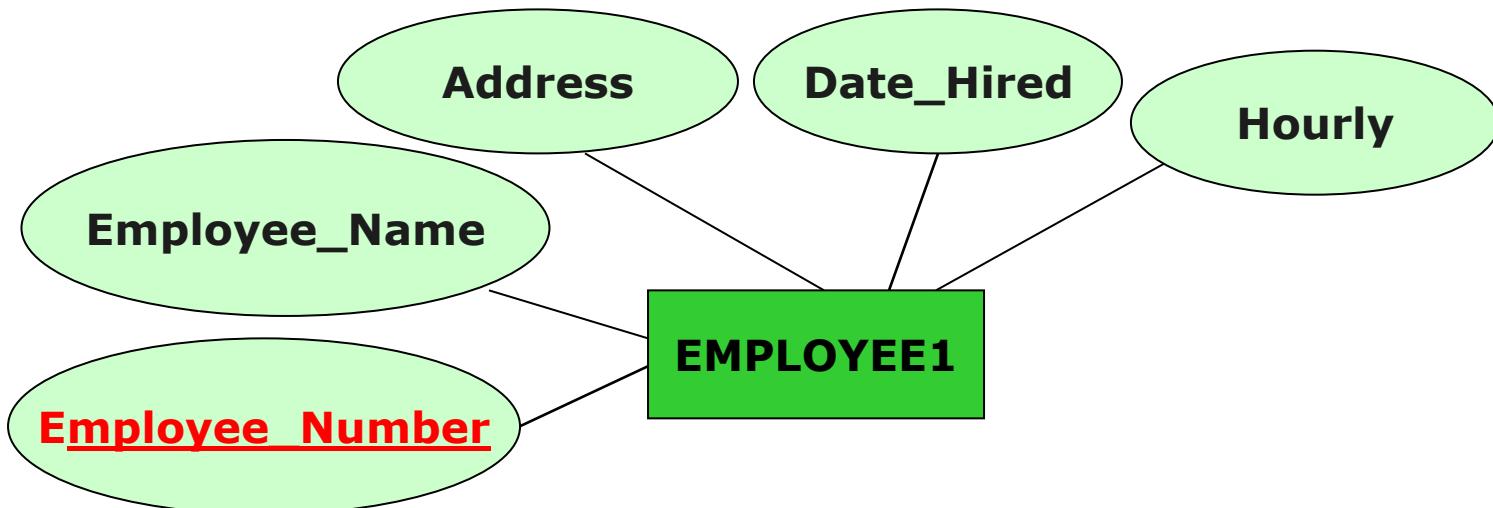


Mô hình liên kết thực thể mở rộng – mô hình EER

Enhanced Entity Relationship model

- **Ví dụ:** một công ty có 3 loại nhân viên khác nhau: làm theo giờ, theo tháng và lương theo hợp đồng. Thể hiện quy tắc nghiệp vụ này trên ER như thế nào??

→ Cách 2: Tạo 3 kiểu thực thể riêng biệt cho 3 loại nhân viên
→ không tận dụng được những thuộc tính chung



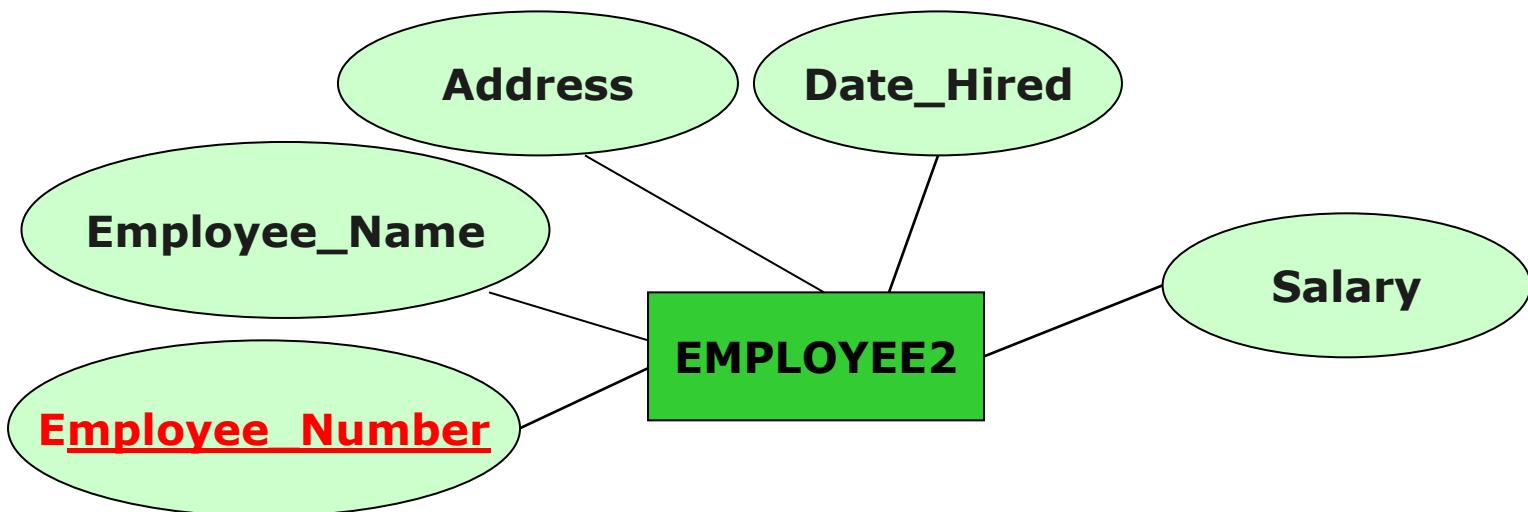


Mô hình liên kết thực thể mở rộng – mô hình EER

Enhanced Entity Relationship model

- Ví dụ: một công ty có 3 loại nhân viên khác nhau: làm theo giờ, theo tháng và lương theo hợp đồng. Thể hiện quy tắc nghiệp vụ này trên ER như thế nào??

→ Cách 2: Tạo 3 kiểu thực thể riêng biệt cho 3 loại nhân viên
→ không tận dụng được những thuộc tính chung



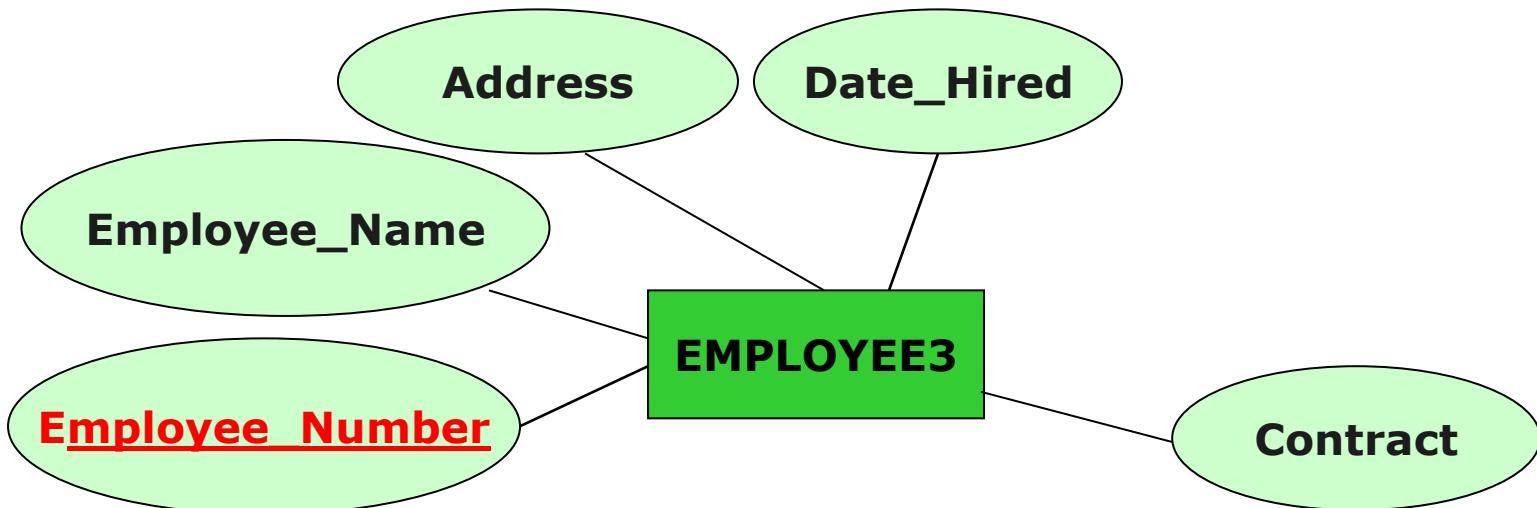


Mô hình liên kết thực thể mở rộng – mô hình EER

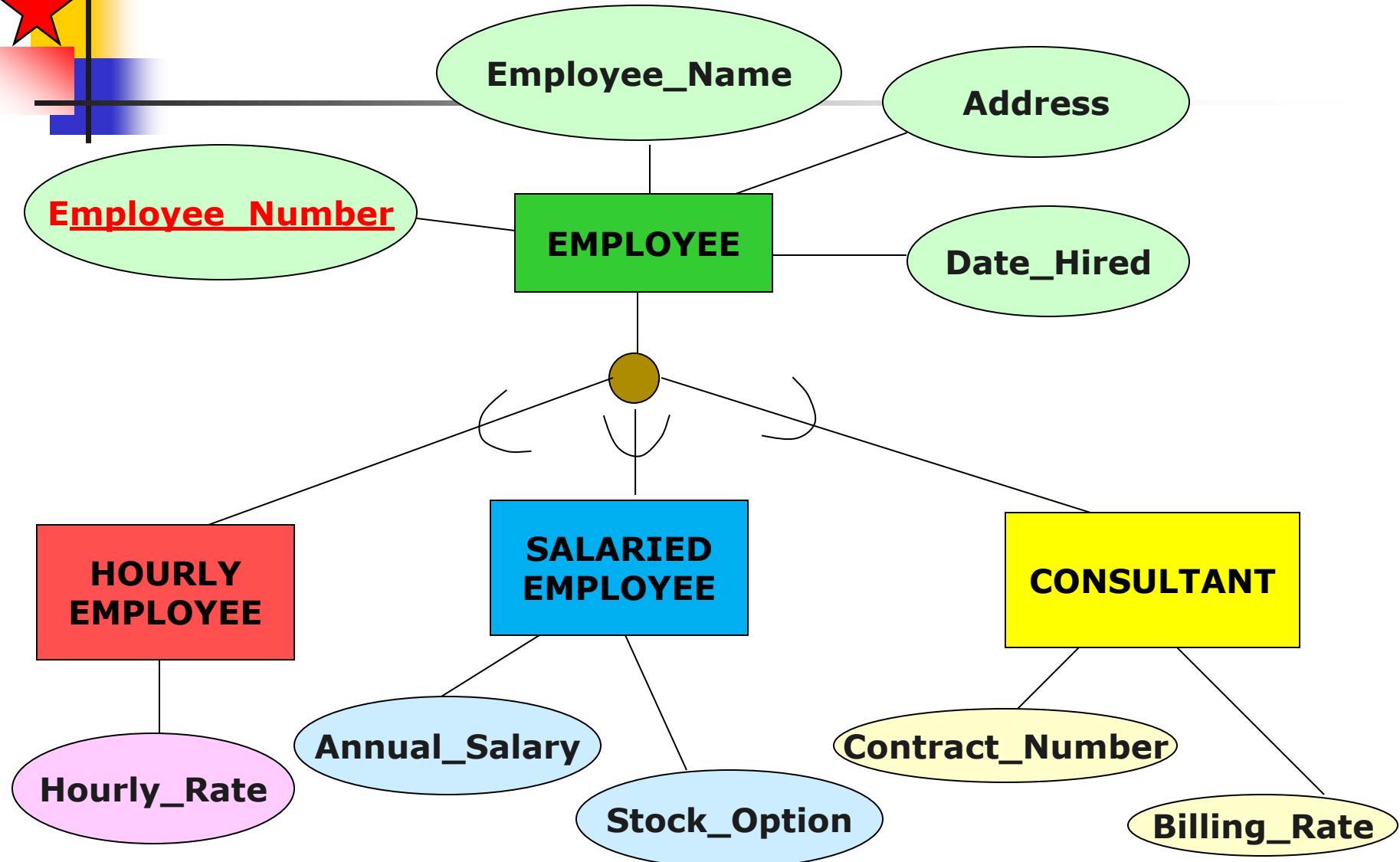
Enhanced Entity Relationship model

- Ví dụ: một công ty có 3 loại nhân viên khác nhau: làm theo giờ, theo tháng và lương theo hợp đồng. Thể hiện quy tắc nghiệp vụ này trên ER như thế nào??

→ Cách 2: Tạo 3 kiểu thực thể riêng biệt cho 3 loại nhân viên
→ không tận dụng được những thuộc tính chung



Ví dụ sơ đồ EER



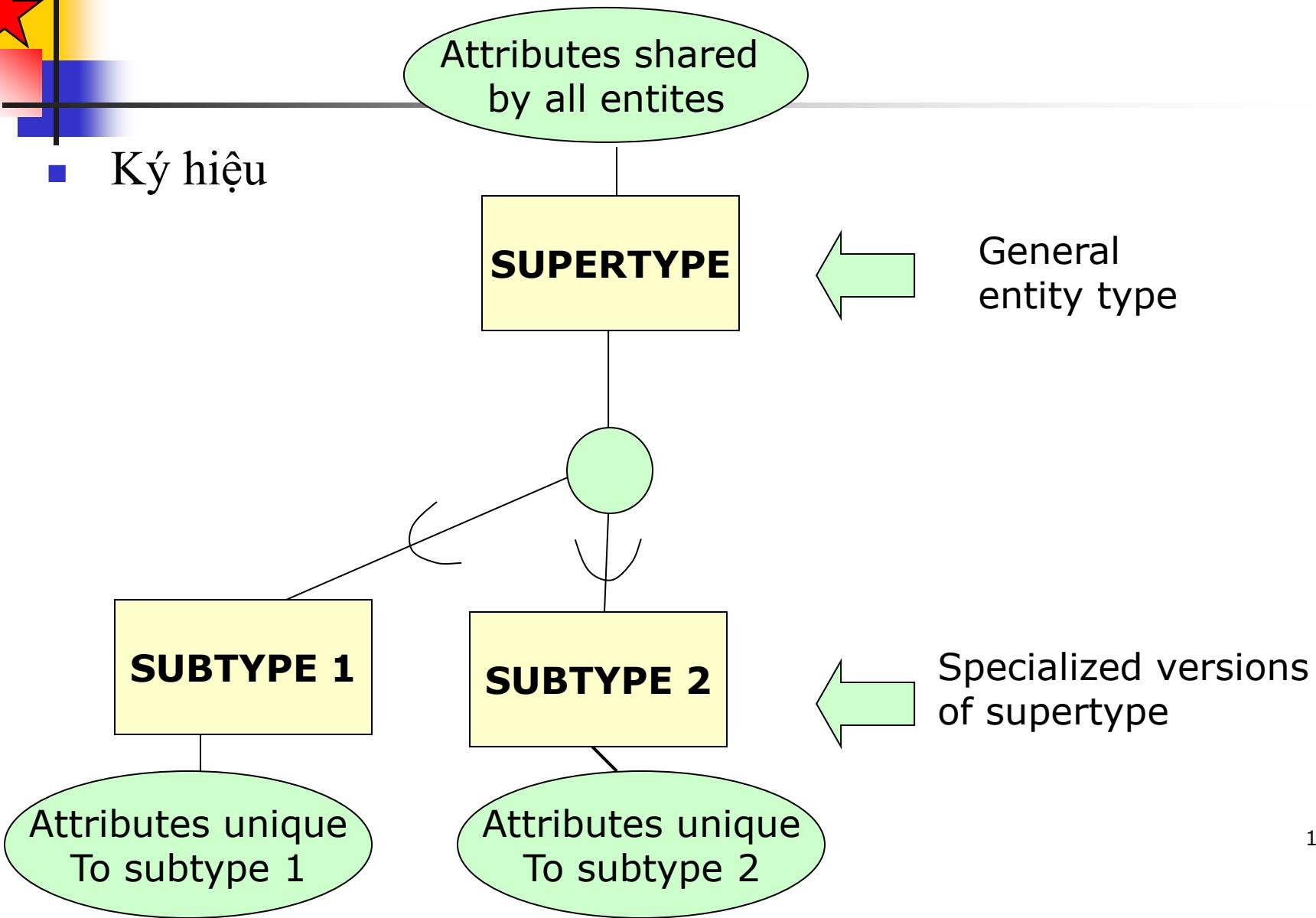


Siêu kiểu và kiểu con (Supertype và subtype)

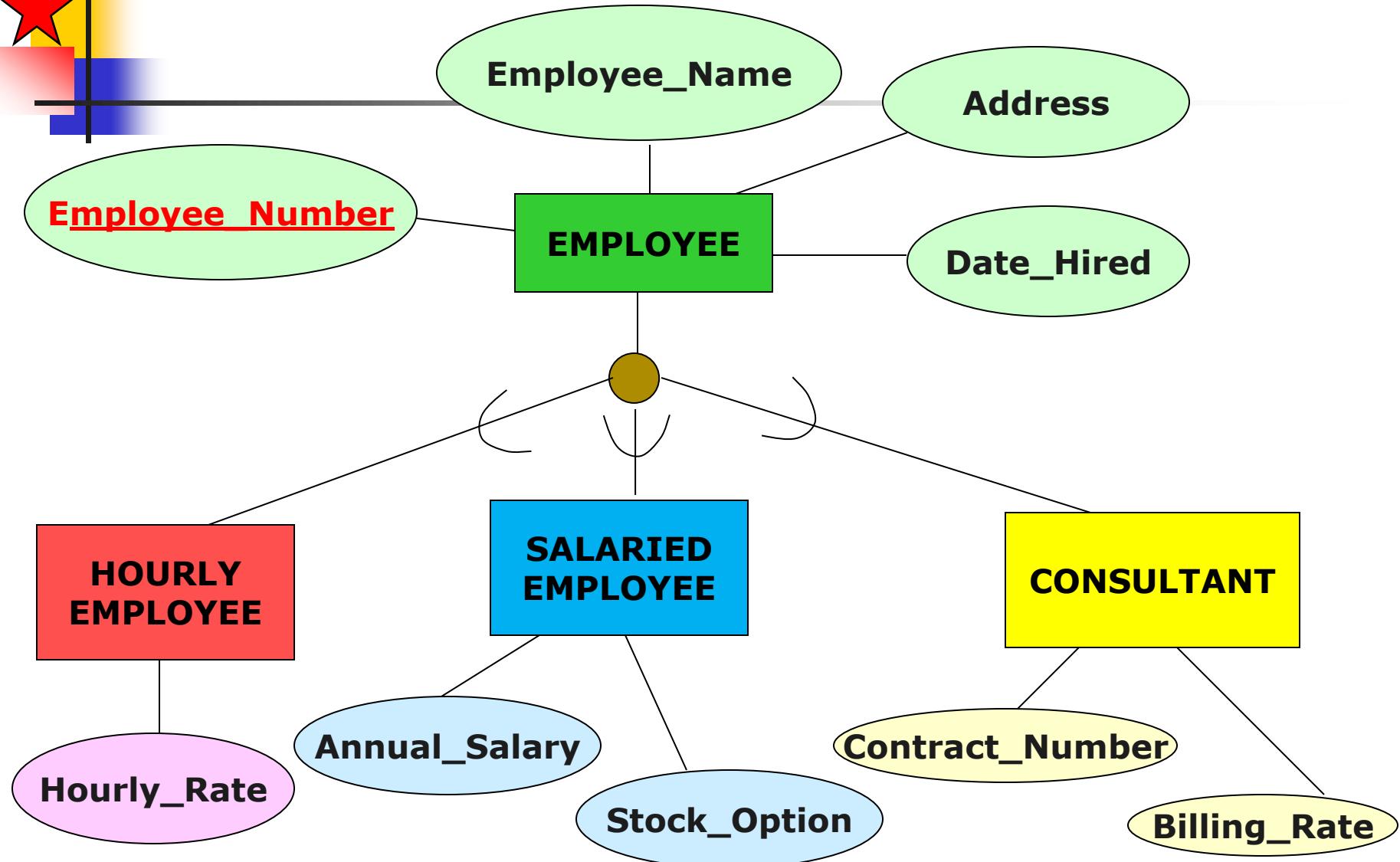
- **Siêu kiểu (supertype):** là kiểu thực thể tổng quát có mối liên kết với một hay nhiều kiểu con
- **Kiểu con (subtype):** là sự phân nhóm từ một kiểu thực thể thành nhiều kiểu thực thể

Siêu kiểu và kiểu con (tt)

- Ký hiệu



Ví dụ sơ đồ EER





Sự thừa kế thuộc tính

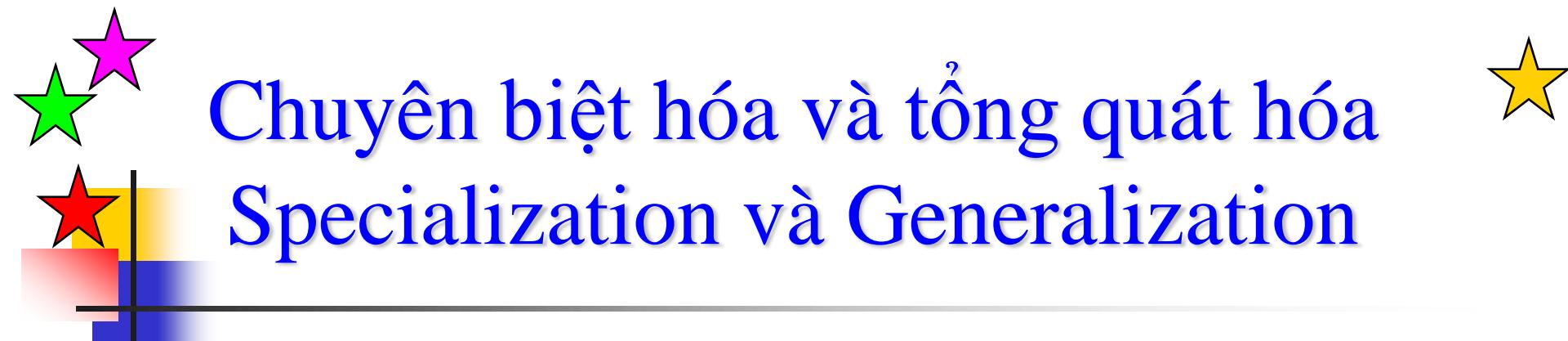
Attribute inheritance

- Sự thừa kế thuộc tính là tính chất mà theo đó các kiểu thực thể con thừa kế trị của mọi thuộc tính thuộc về siêu kiểu
 - Một thành viên của subtype cũng là 1 thành viên của supertype
 - Điều ngược lại không phải lúc nào cũng đúng mà phụ thuộc vào nghiệp vụ



Khi nào sử dụng mối quan hệ supertype/subtype

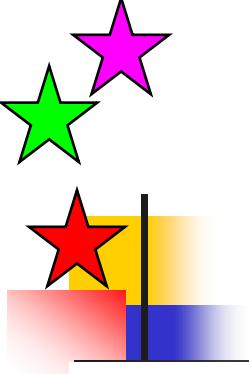
- Có các thuộc tính chỉ dành cho 1 số thể hiện (instance) của kiểu thực thể.
 - Ví dụ: siêu kiểu BenhNhan có 2 subtype:
 - BenhNhanNgoaiTru(*NgayTaiKham, ...*)
 - BenhNhanNoiTru(*SoPhong, SoGiuong, NgayNhapVien, ...*)
- Thể hiện của 1 kiểu con (subtype) tham gia vào mối quan hệ đó là duy nhất cho kiểu con đó
 - Ví dụ: BenhNhanNoiTru có quan hệ với kiểu thực thể GIUONG



Chuyên biệt hóa và tổng quát hóa

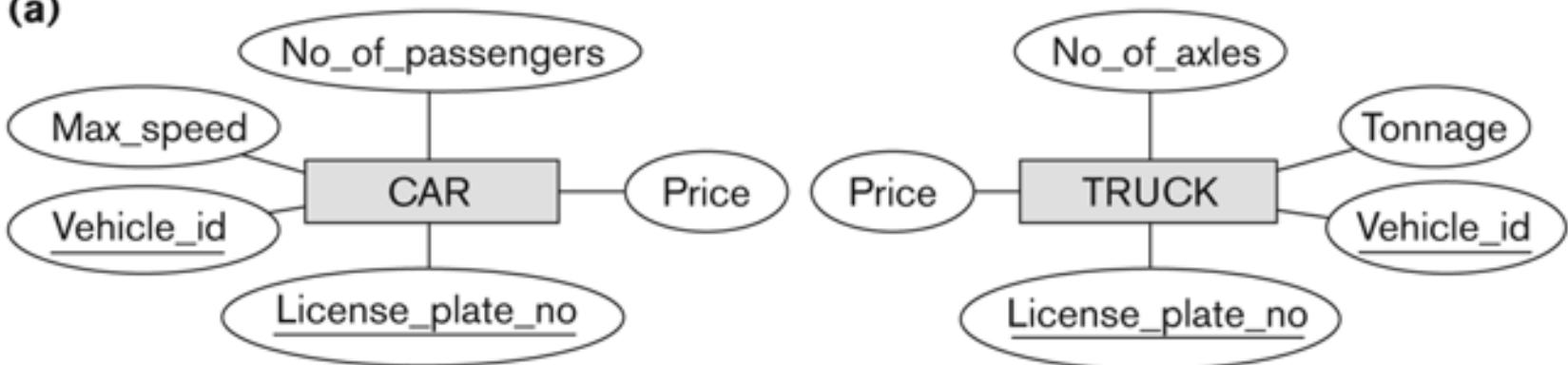
Specialization và Generalization

- **Tổng quát hóa** là quá trình định nghĩa một kiểu dữ liệu tổng quát hơn từ một tập hợp các kiểu dữ liệu chuyên biệt.
Đây là quá trình từ dưới lên (Bottom up)
- Ví dụ: Ba kiểu thực thể CAR, TRUCK và MOTOCYCLE có thể tổng quát hóa thành siêu kiểu VEHICLE chứa các thuộc tính chung là Vehicle_ID, Model, Price,...

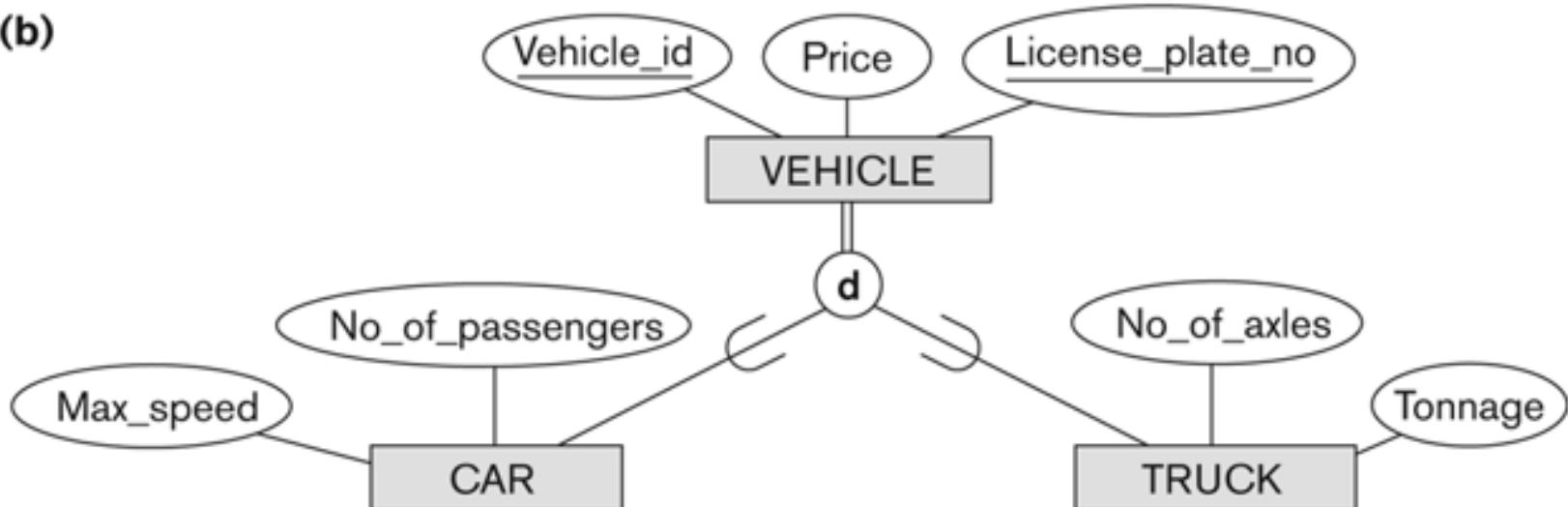


Tổng quát hóa- Generalization

(a)



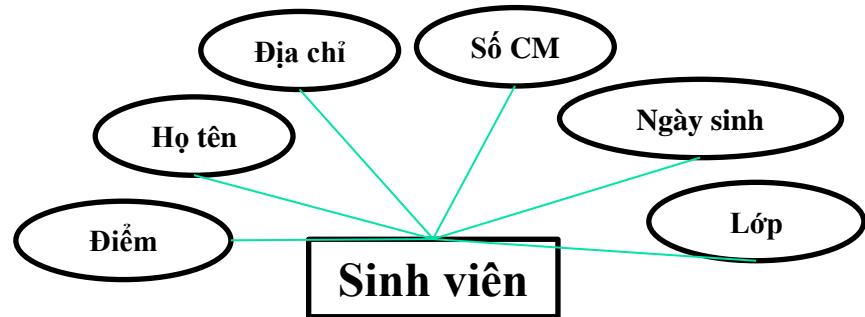
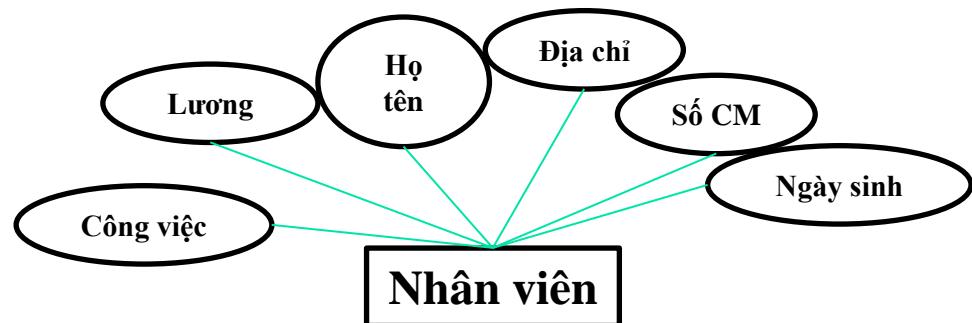
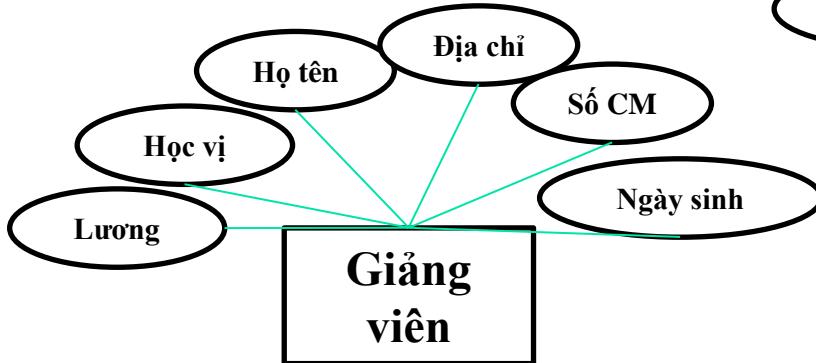
(b)





Tổng quát hóa- Generalization

■ Ví dụ:

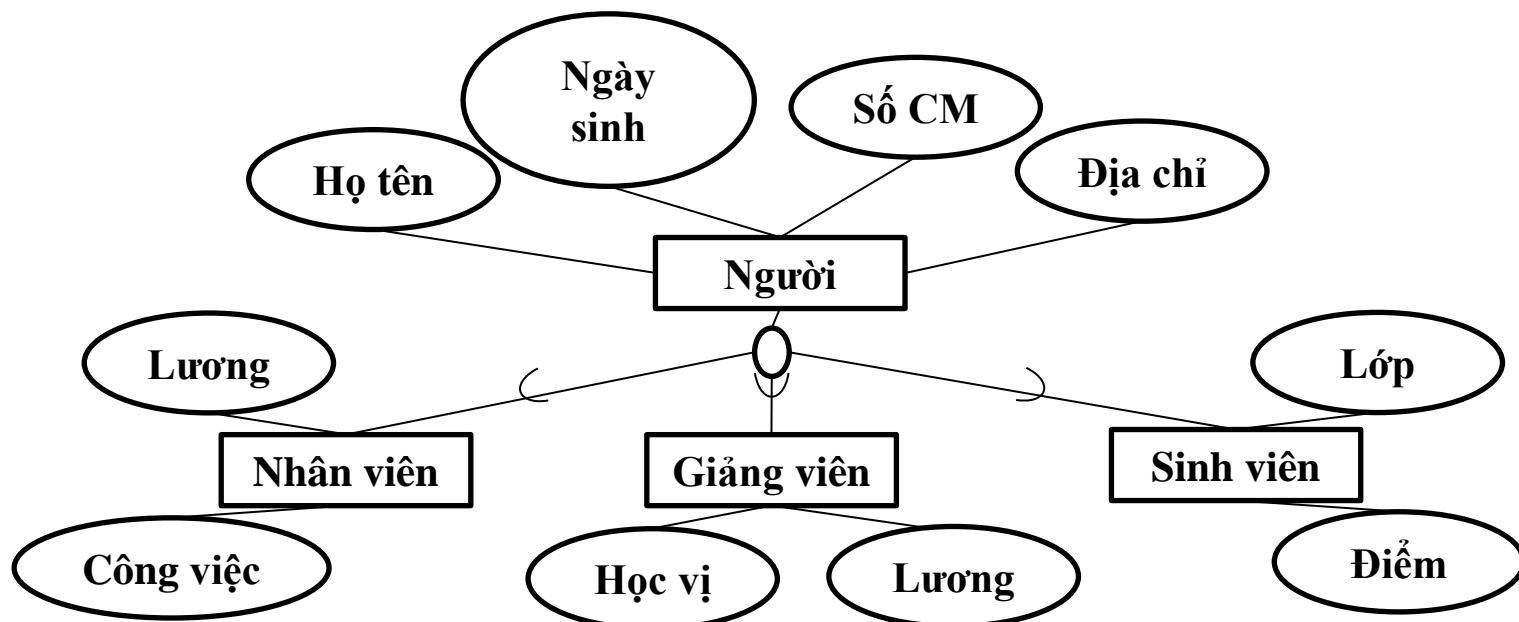


Các thực thể Nhân viên, giảng viên và sinh viên trước khi tổng quát hóa



Tổng quát hóa- Generalization

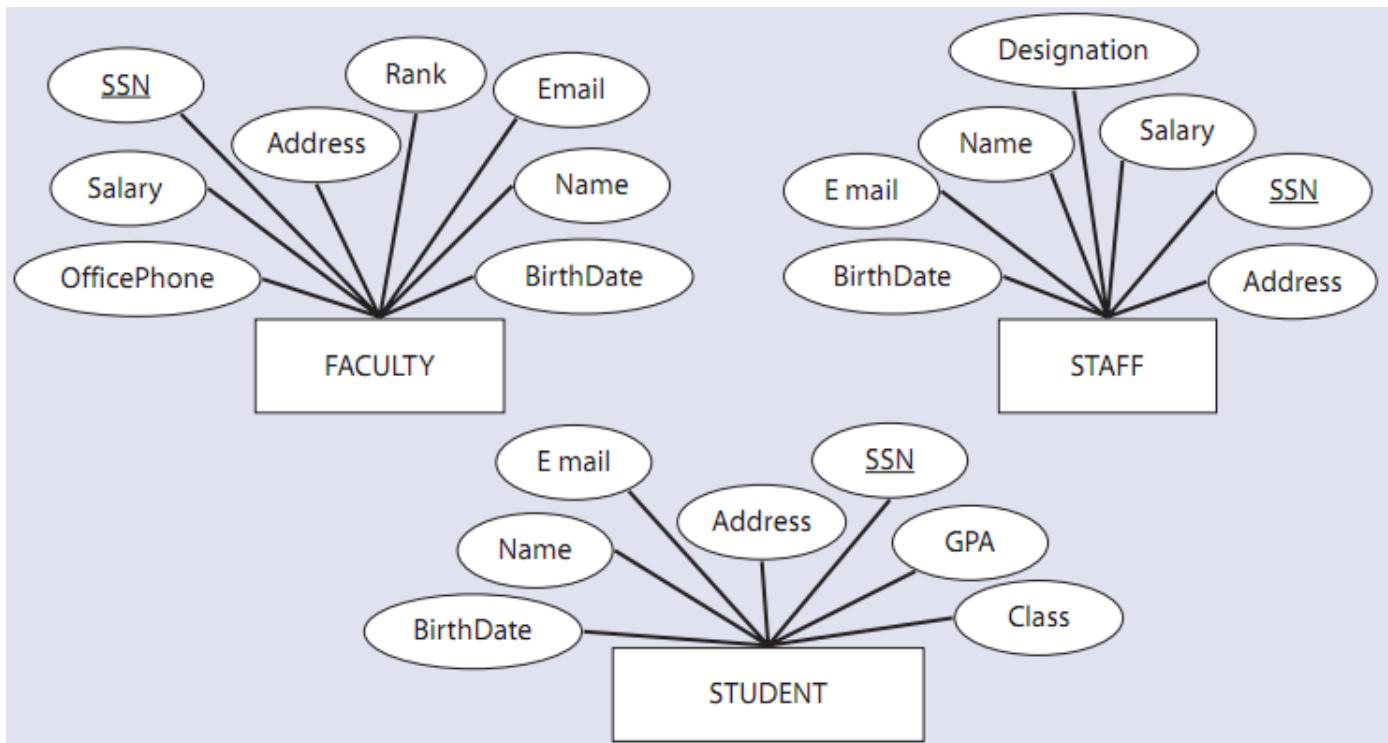
- Ví dụ: sau khi tổng quát hóa: những thuộc tính chung được đặt trong lớp cha.





Tổng quát hóa- Generalization

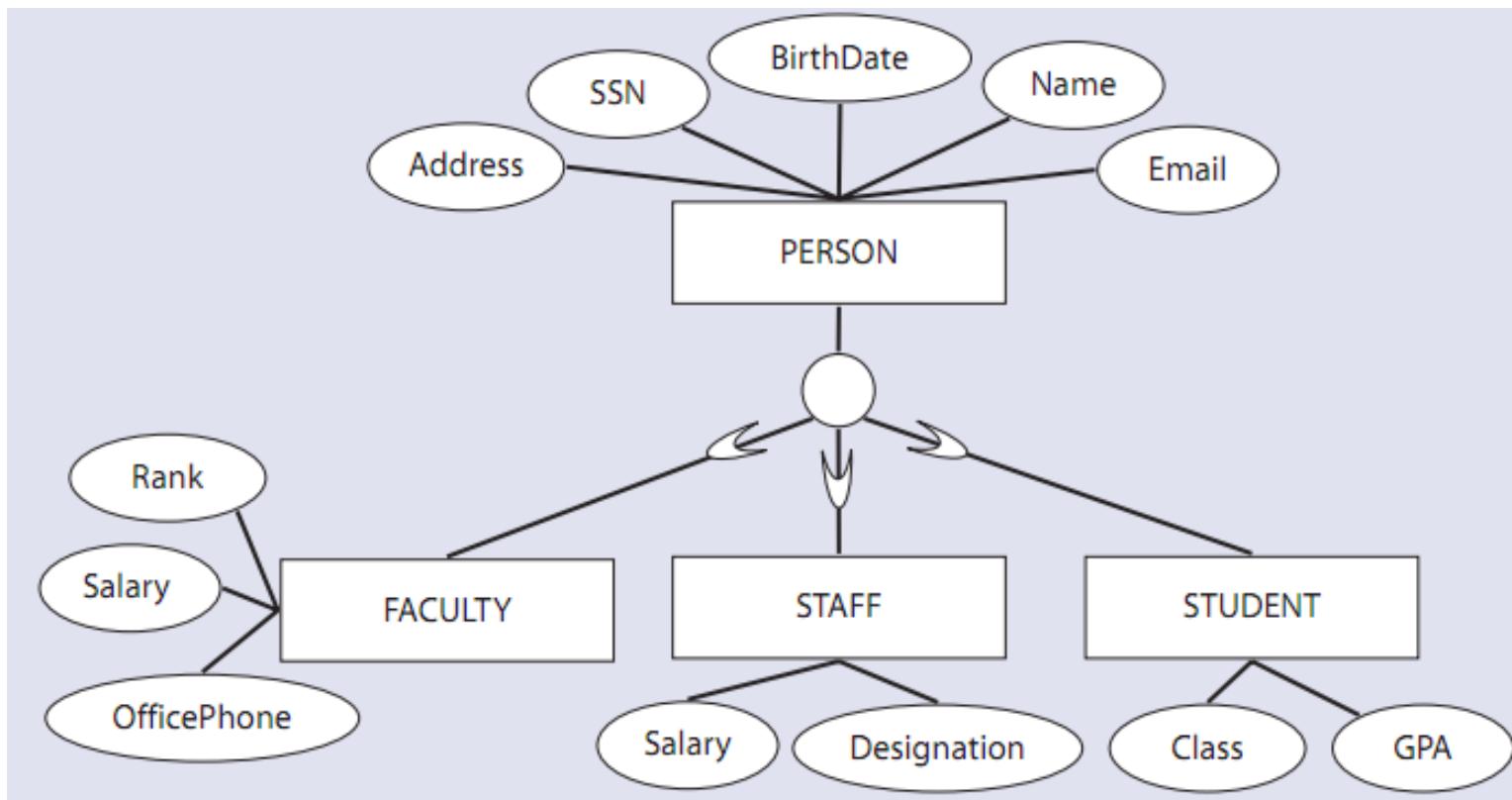
Ví dụ: loại thực thể FACULTY, STAFF, STUDENT trước khi tổng quát hóa





Tổng quát hóa- Generalization

Ví dụ: Các loại thực thể FACULTY, STAFF, STUDENT sau khi tổng quát hóa





Chuyên biệt hóa - Specialization

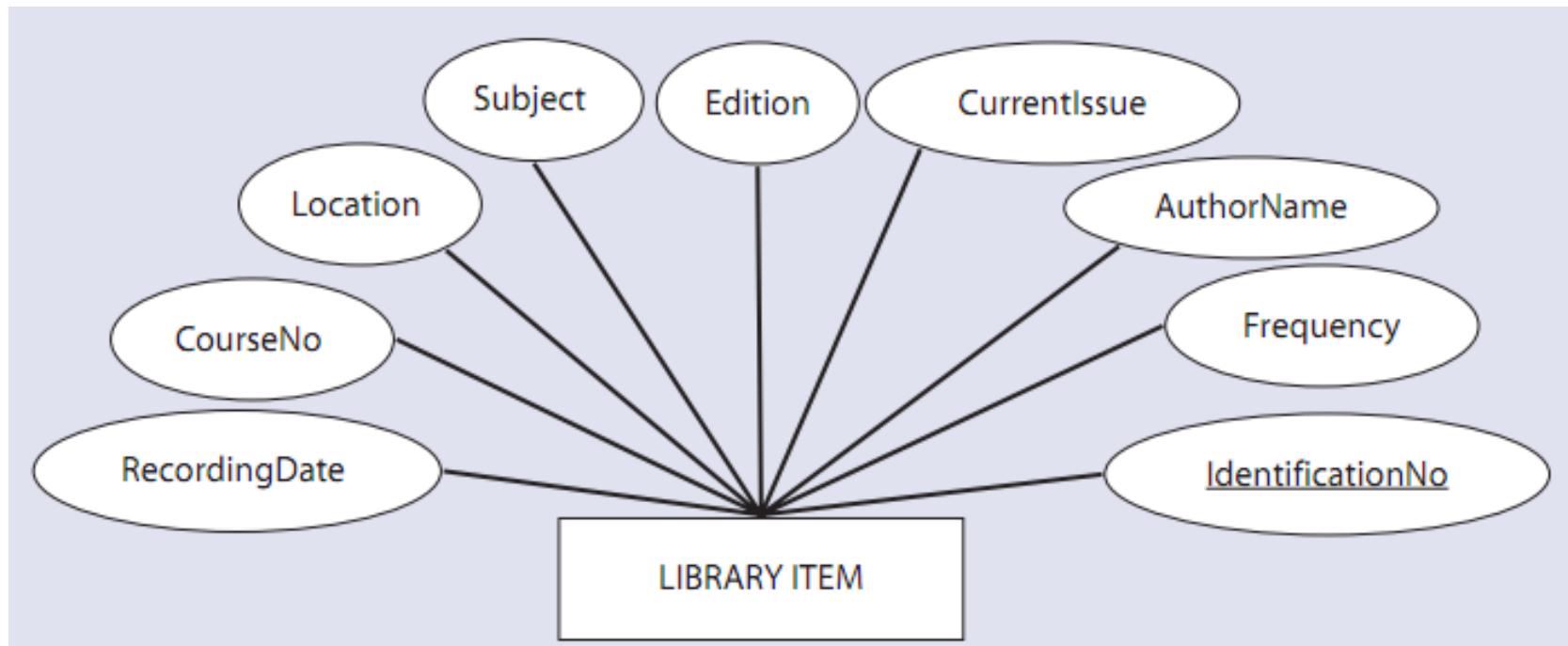
- **Chuyên biệt hóa** là quá trình định nghĩa một hay nhiều kiểu con từ một siêu kiểu và hình thành mối liên kết siêu kiểu/kiểu con

Là quá trình từ trên xuống (Top down), bắt đầu từ loại thực thể tổng quát (superclass) xác định những subclasses dựa trên những thuộc tính riêng hoặc mối quan hệ cụ thể của lớp con



Chuyên biệt hóa - Specialization

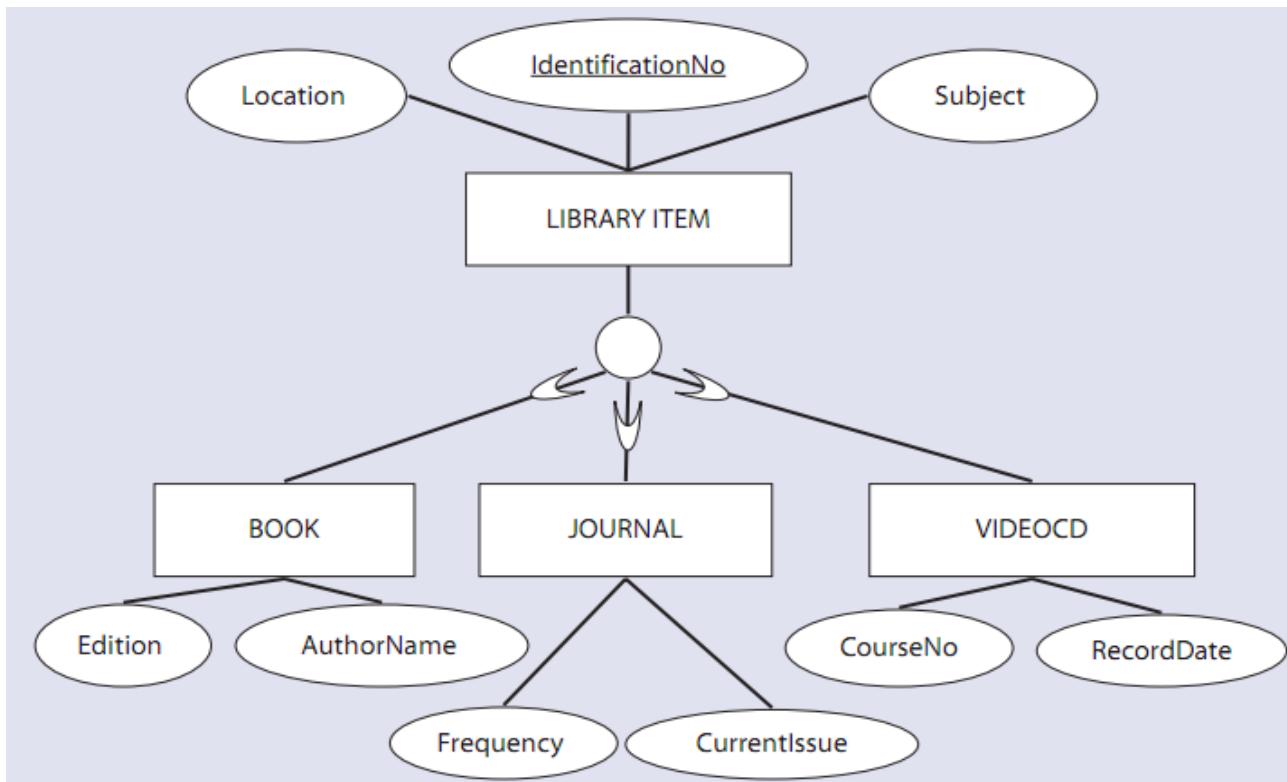
- Ví dụ: loại thực thể LIBRARY ITEM với các thuộc tính:





Chuyên biệt hóa - Specialization

- Ví dụ: Sau khi chuyên biệt hóa: superclass: LIBRARY ITEM và subclasses BOOK, JOURNAL, VIDEOCD.

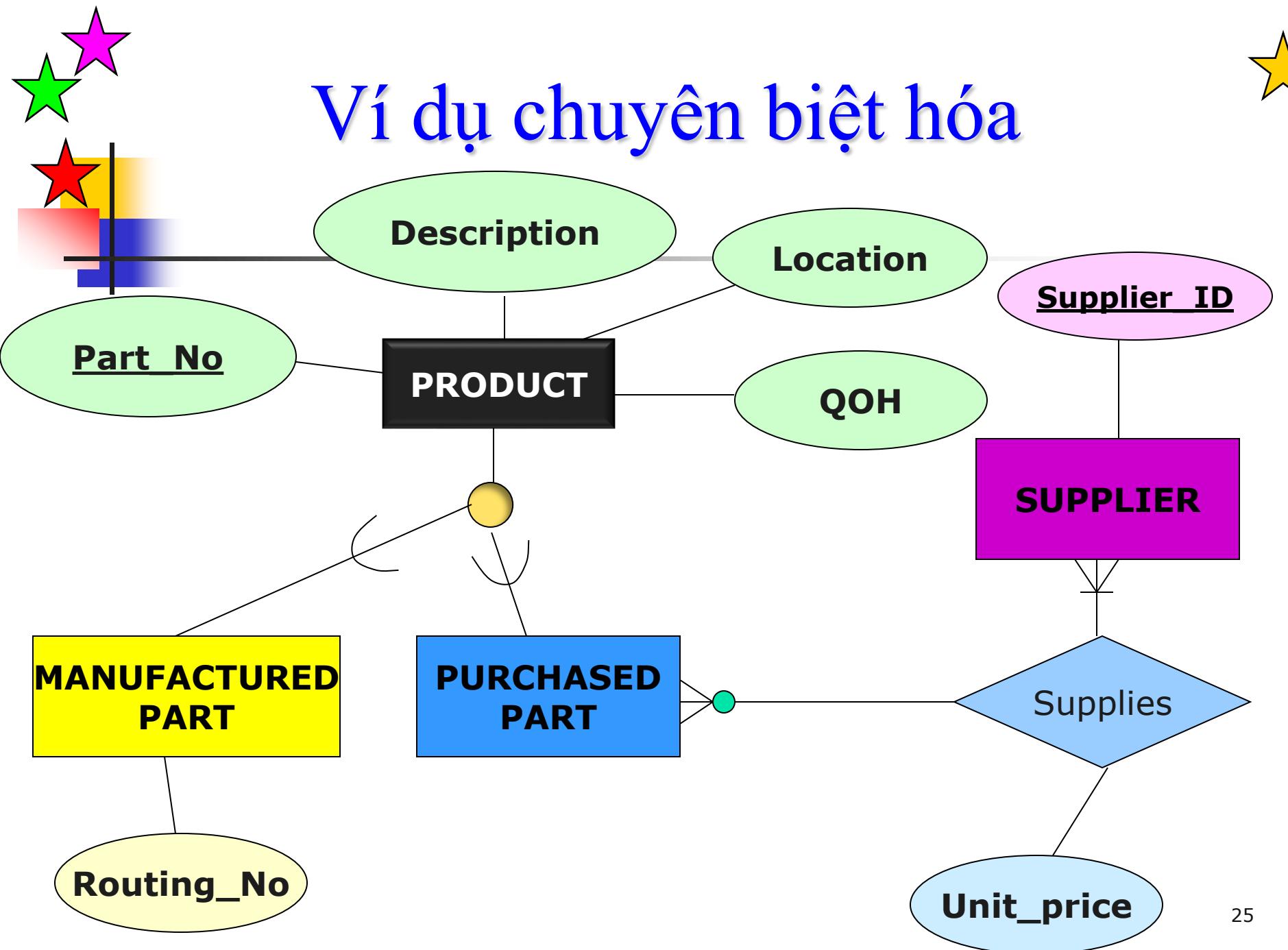




Chuyên biệt hóa - Specialization

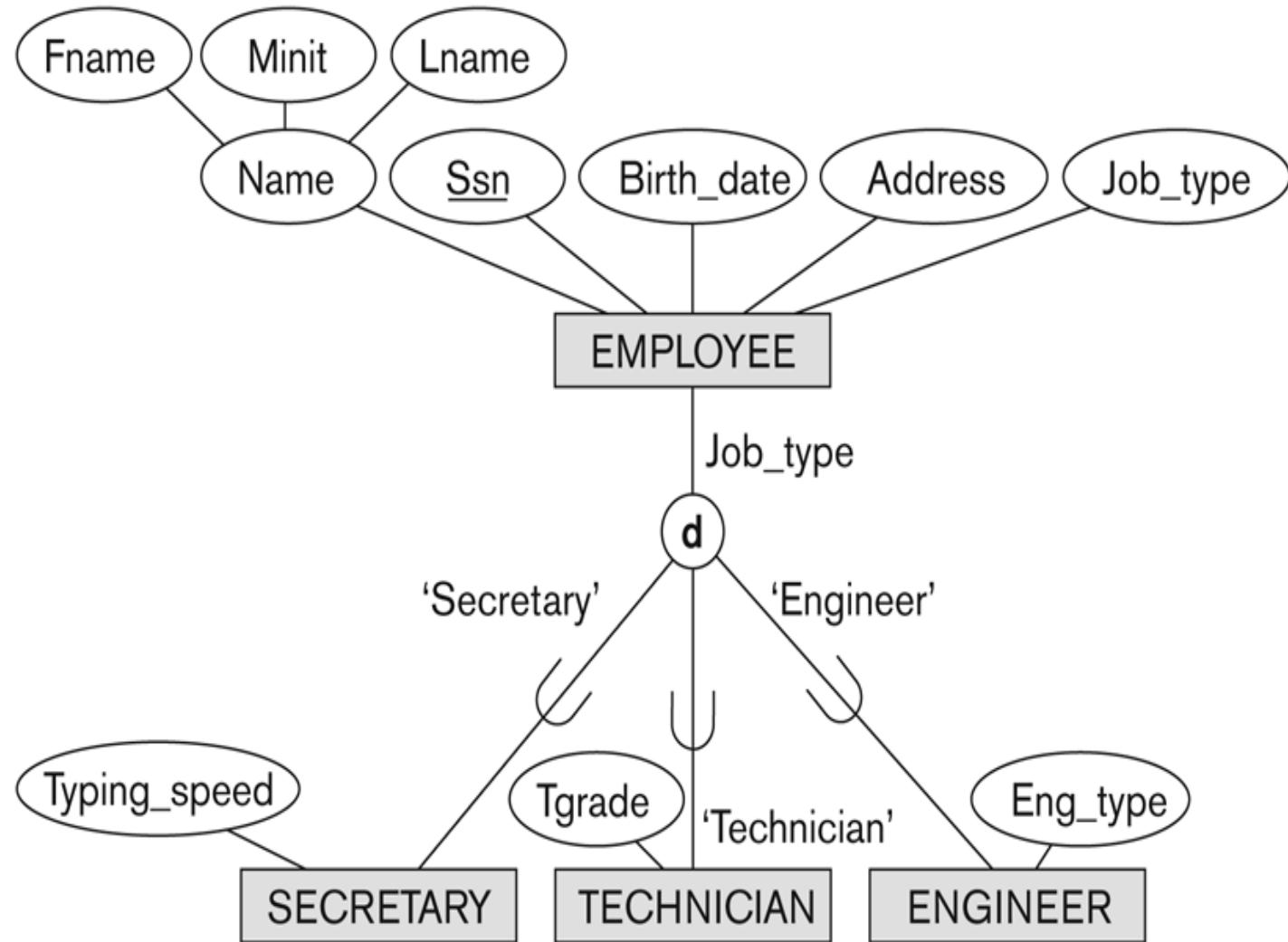
- **Ví dụ:** kiểu thực thể PRODUCT có 1 thuộc tính đa trị là Supplier (có thể được cung cấp tại chỗ hoặc từ nhà sản xuất bên ngoài) → PRODUCT nên được chuyên biệt hóa thành 2 kiểu con MANUFACTURED PART và PURCHASED PART

Ví dụ chuyên biệt hóa



Ví dụ chuyên biệt hóa

EER diagram notation
for an attribute-defined specialization
on Job_type.



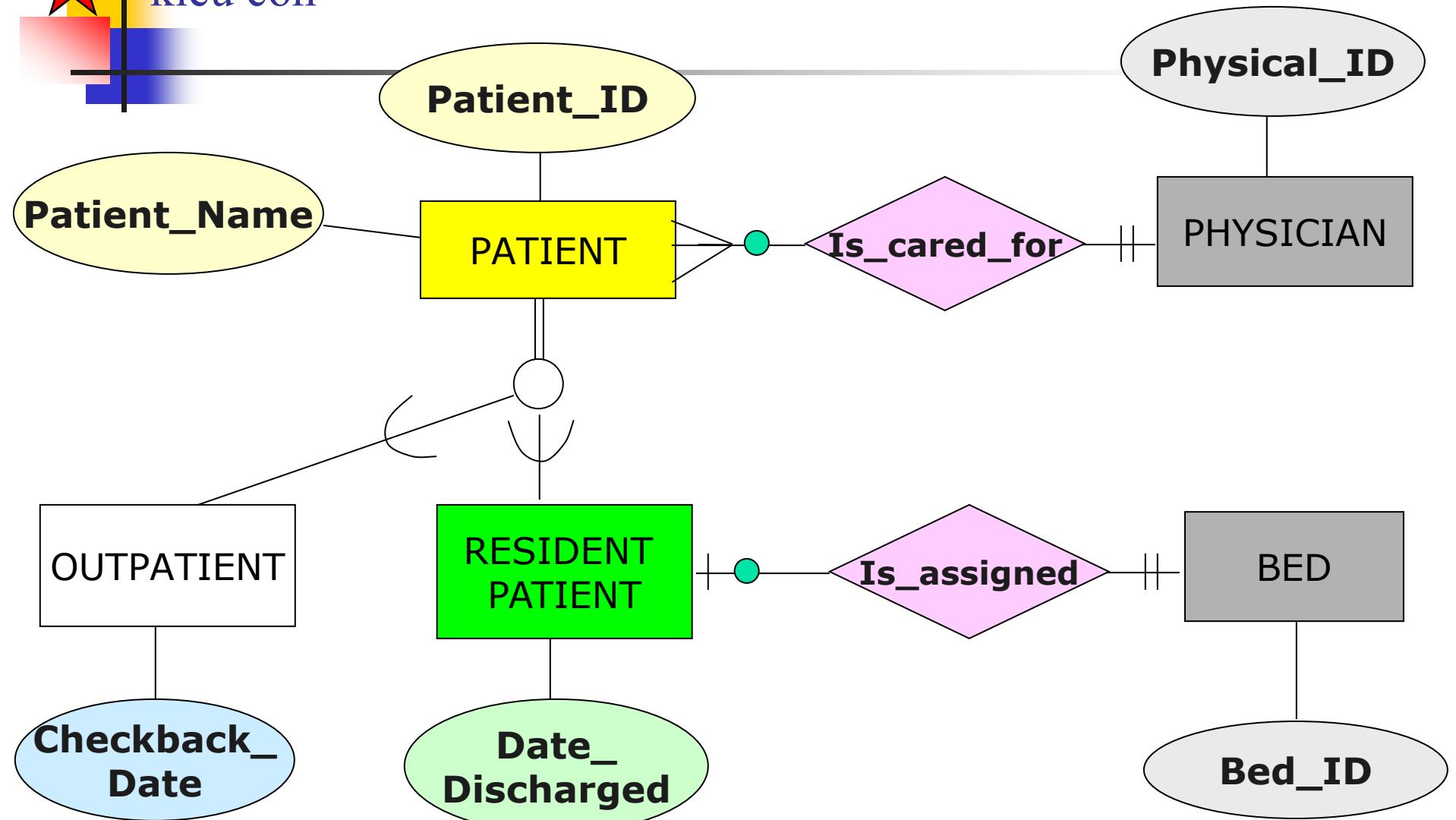


Chuyên biệt hóa

- **Có hai nguyên tắc (rule):**
 - Chuyên biệt hóa toàn phần (total specialization)
 - Chuyên biệt hóa riêng phần (partial specialization)

Chuyên biệt hóa toàn phần

Mỗi thể hiện của siêu kiểu tất yếu phải là một thể hiện của một kiểu con

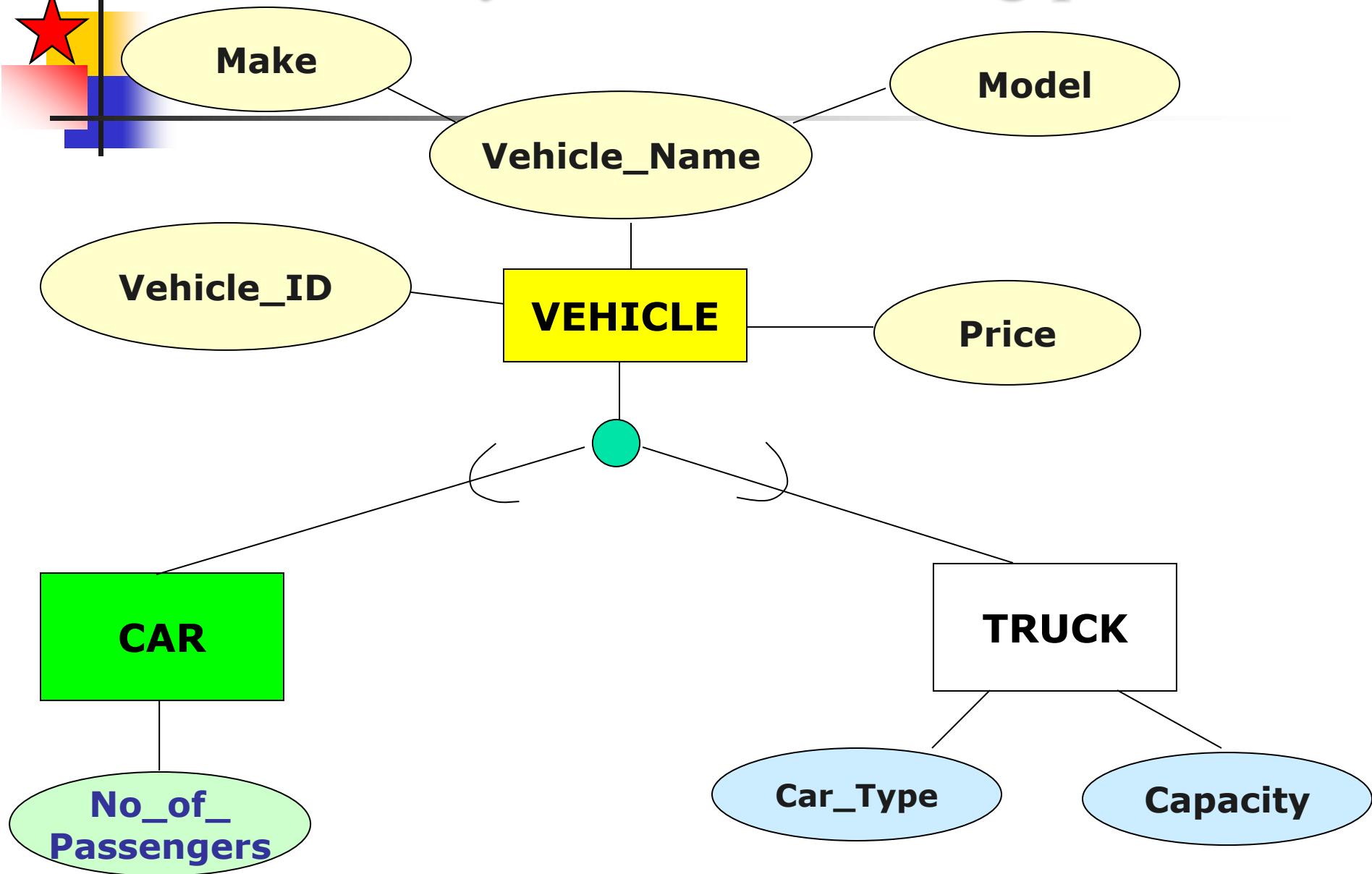


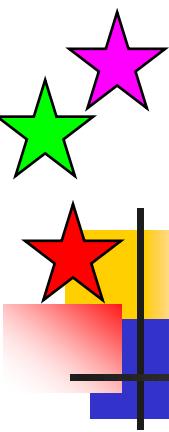


Chuyên biệt hóa riêng phần

- **Chuyên biệt hóa riêng phần:** mỗi thể hiện của siêu kiểu không nhất thiết phải là 1 thể hiện của một kiểu con
- Ví dụ: siêu kiểu VEHICLE có 2 kiểu con CAR và TRUCK. Kiểu thực thể MOTORCYCLE cũng là 1 loại xe cộ nhưng không được đưa vào mô hình

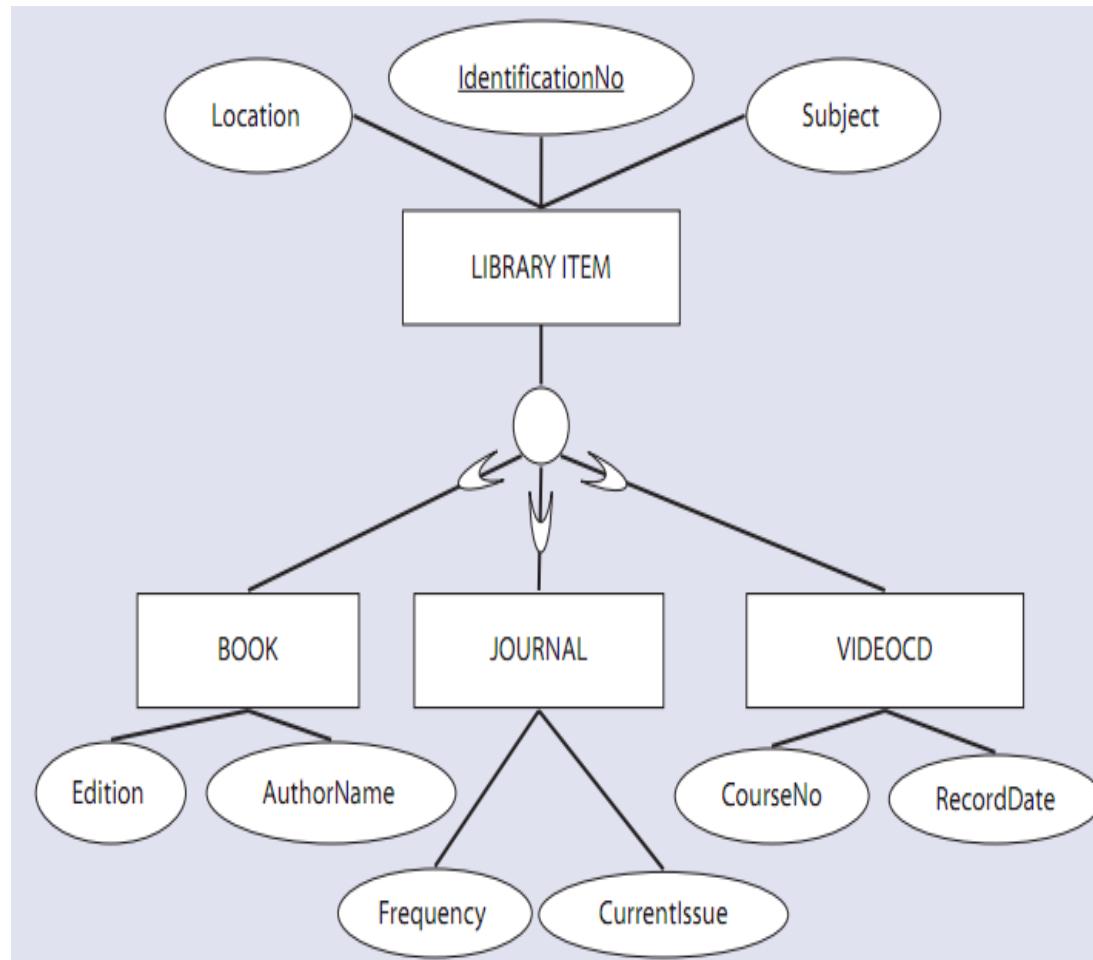
Ví dụ Chuyên biệt hoá riêng phần





Chuyên biệt hóa riêng phần

- 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 trong một trong các lớp con.





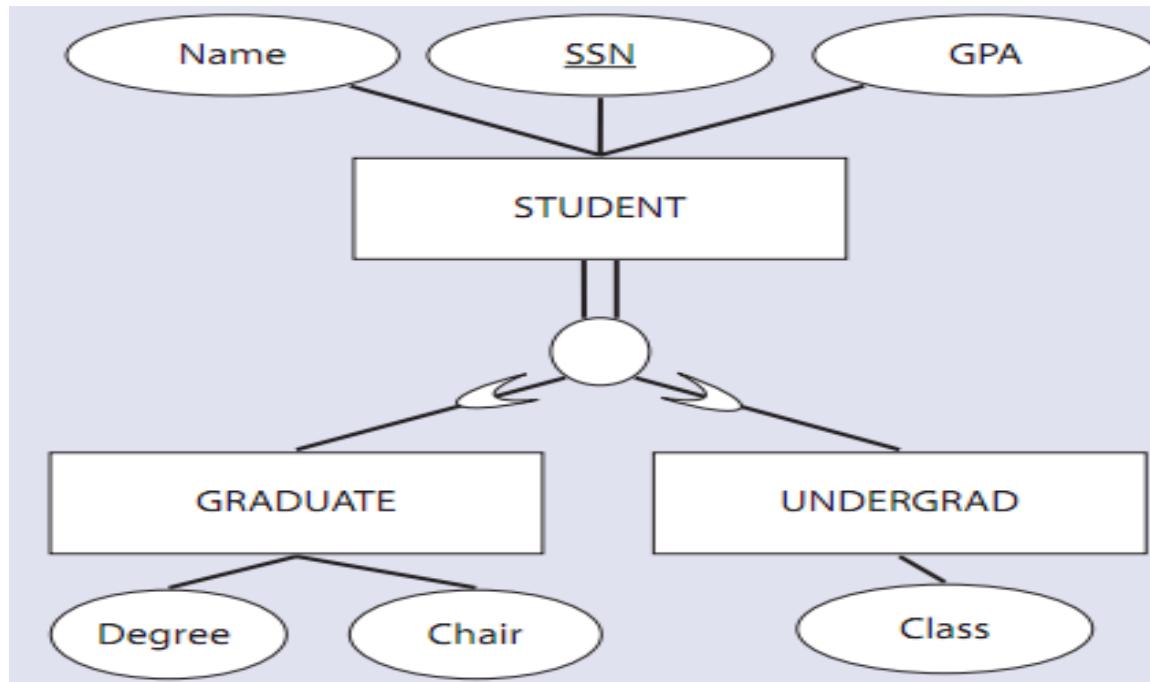
Ràng buộc trong môi liên kết siêu kiểu/ kiểu con

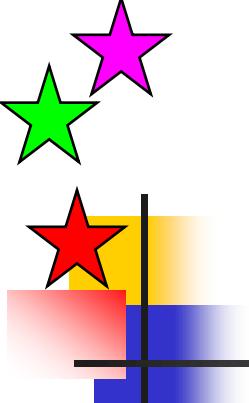
- Hai loại ràng buộc
 - Ràng buộc về tính đầy đủ (completeness constraint)
 - Ràng buộc về tính phân ly (Disjointness constraint)



Ràng buộc về tính đầy đủ

- Ràng buộc về tính đầy đủ dùng để trả lời cho câu hỏi: “**Một thể hiện của siêu kiểu có phải là thành viên của ít nhất một kiểu con hay không?**”





Ràng buộc về tính phân ly

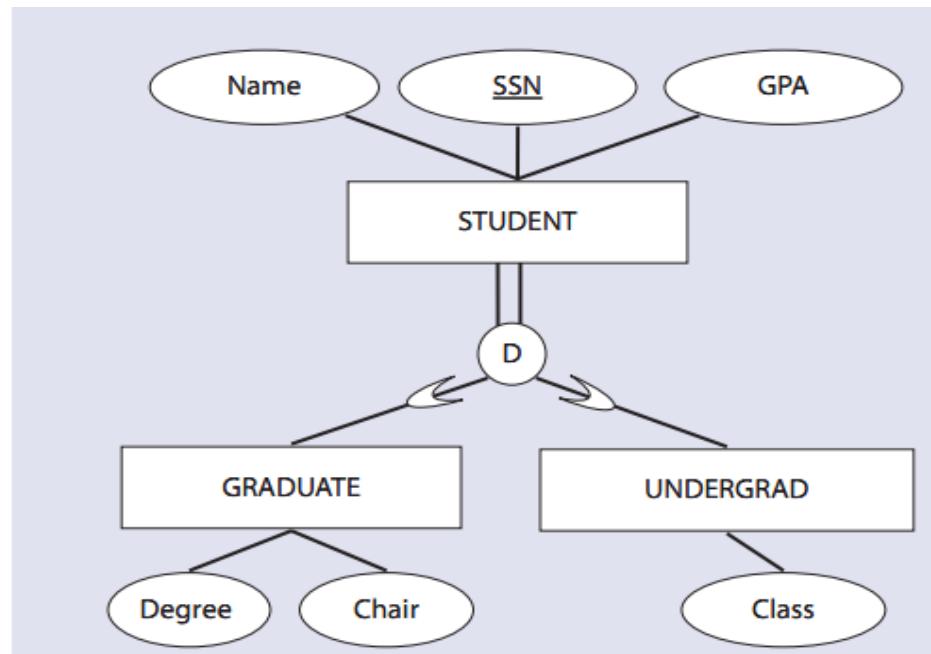
Disjointness constraint

- Ràng buộc về tính phân ly để trả lời cho câu hỏi “**một thể hiện (instance) của siêu kiểu có đồng thời là thành viên của cả 2 kiểu con hay không?**”.
- Hai nguyên tắc (rule):
 - **Phân ly (disjoint)**
 - **Trùng lặp (overlap)**

Ràng buộc về tính phân ly

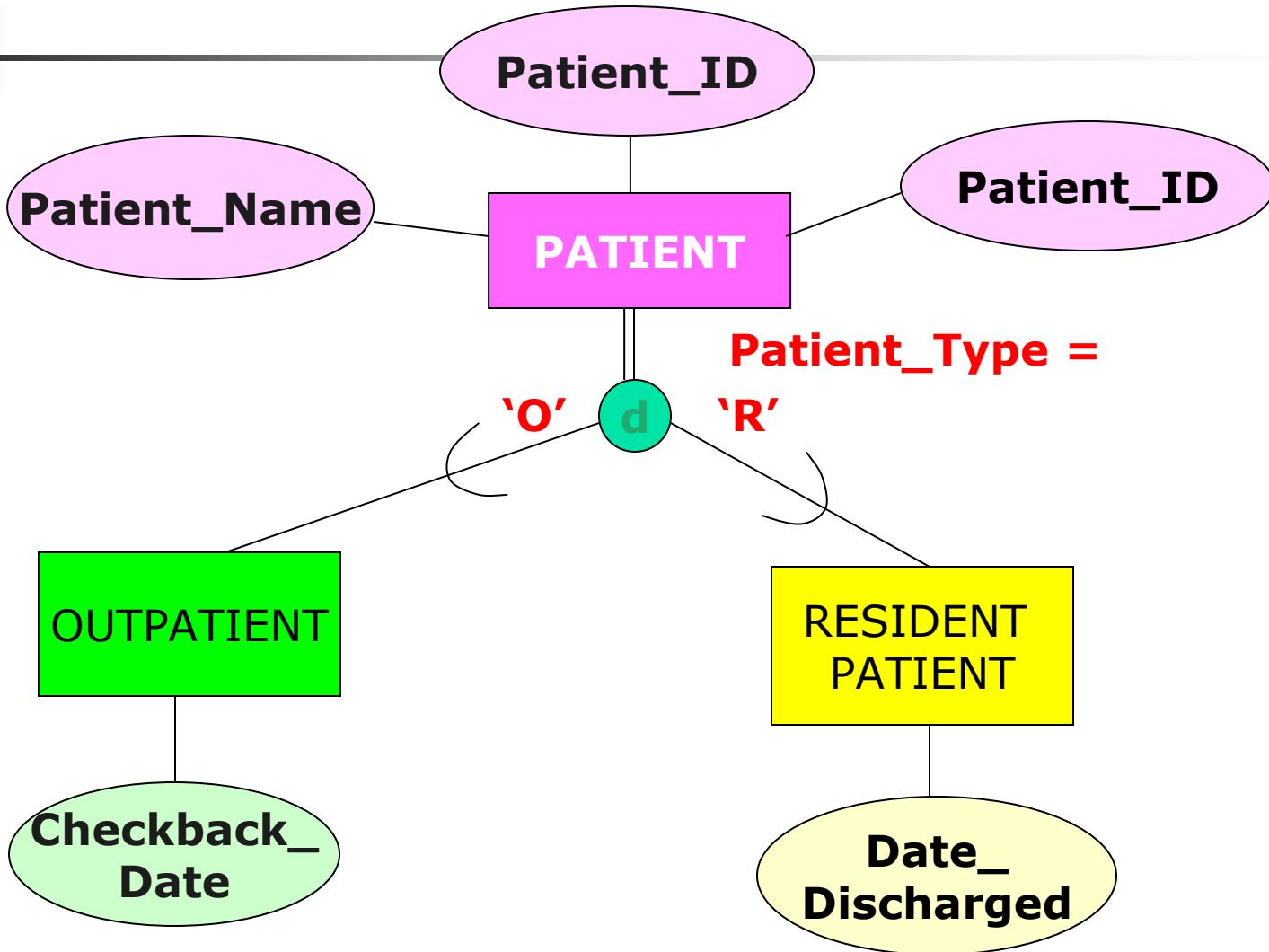
Disjointness constraint

- **Phân ly (disjoint):** một thể hiện của siêu kiểu là thành viên của chỉ một kiểu con
- Trong mỗi quan hệ superclass/subclass, ràng buộc Disjoint được ký hiệu là D.



Ví dụ thuộc tính kiểu phân ly

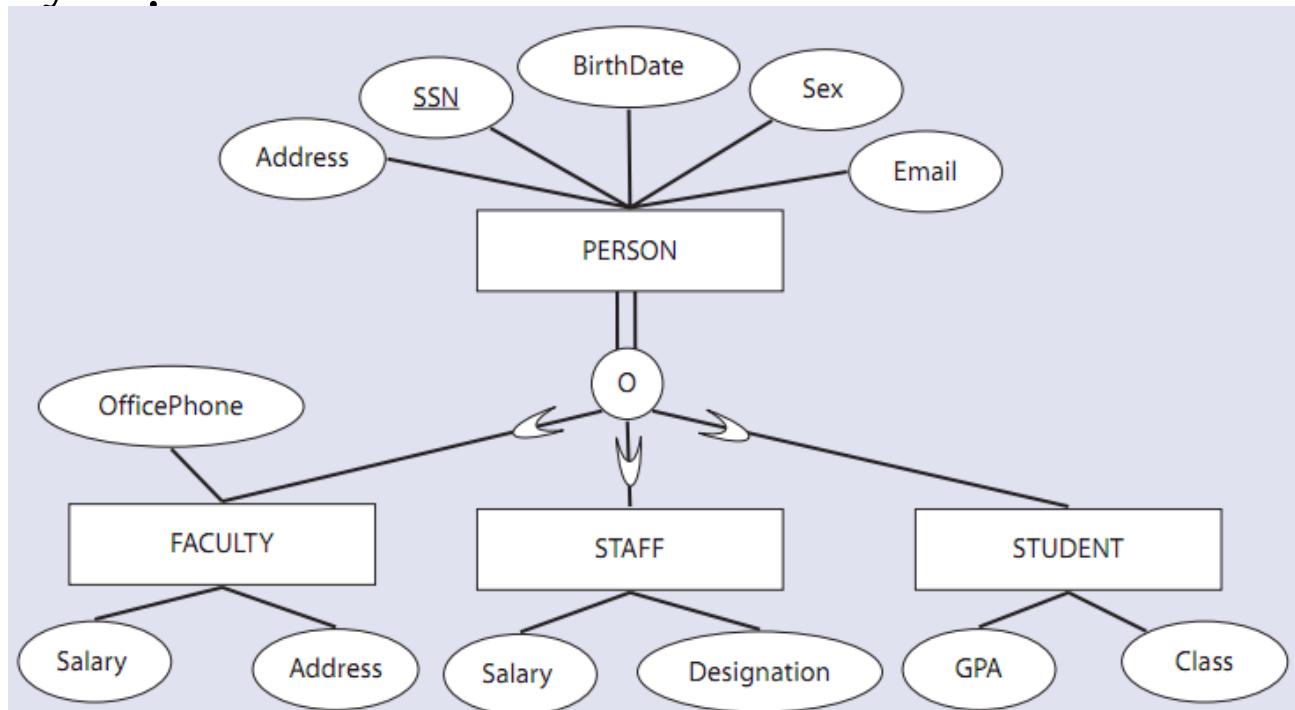
Ví dụ: PATIENT chỉ có thể hoặc là OUTPATIENT hoặc là RESIDENT PATIENT



Ràng buộc về tính phân ly

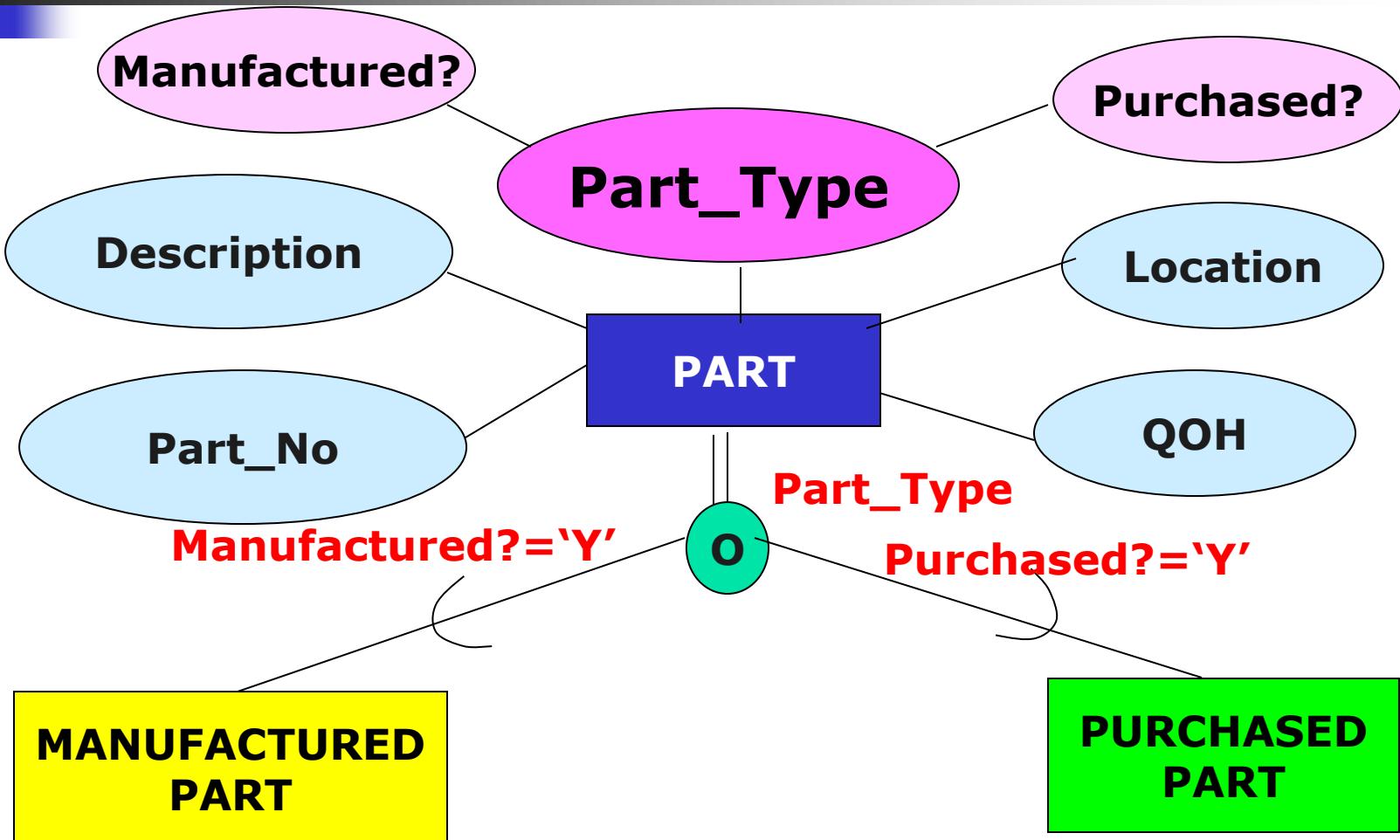
Disjointness constraint

- **Trùng lặp (overlap):** một thể hiện của siêu kiểu **có thể đồng thời là thành viên của nhiều hơn một kiểu con**
- Trong mối quan hệ superclass/subclass, overlap constraint được ký hiệu là O



Ví dụ thuộc tính kiểu trùng lắp

Ví dụ: một thể hiện của siêu kiểu PART có thể hoặc là thể hiện của kiểu con MANUFACTURED PART hoặc của kiểu con PURCHASED PART



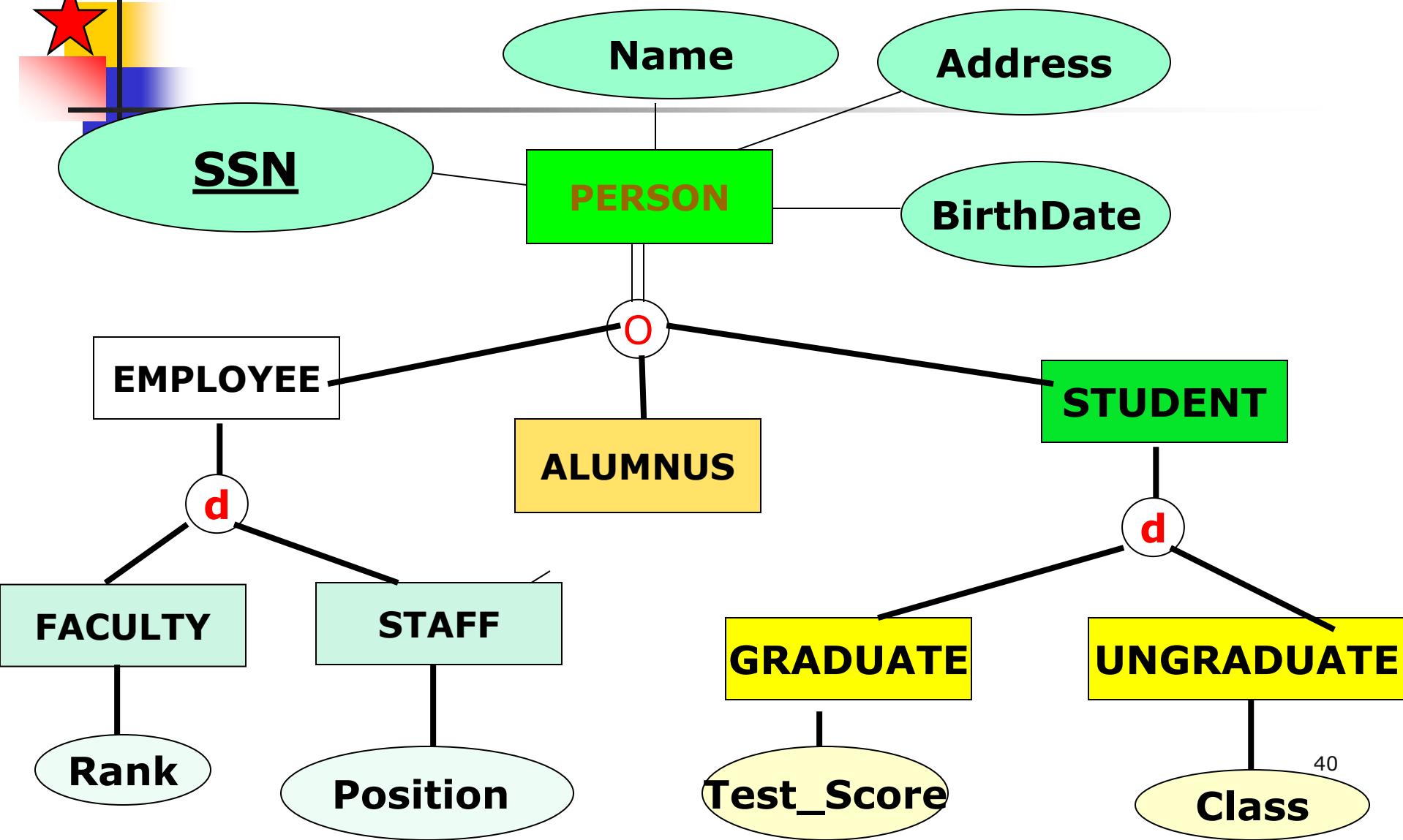


Thứ tự phân cấp (Hierarchy) của siêu kiểu/kiểu con

- Một kiểu con có thể trở thành siêu kiểu cho 1 số kiểu con khác
- Siêu kiểu ở mức cao nhất được gọi là root
- Ví dụ: hãy lập mô hình nhân lực (human resource) của 1 trường đại học

Một faculty thì sẽ có những thuộc tính gì?

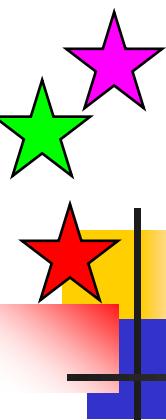
Ví dụ mô hình nhân lực trường đại học





Bài tập

- Một bệnh viện có nhiều **khu chữa trị**. Một khu chữa trị có số của khu (định danh) và tên. **Bệnh nhân** gồm hai loại: bệnh nhân nội trú và bệnh nhân ngoại trú. Bệnh nhân có mã bệnh nhân , tên và ngày sinh. Các bác sĩ có mã bác sĩ và tên. Bác sĩ có thể tiếp nhận hoặc điều trị bệnh nhân.
- Mỗi **giường bệnh** có số giường, số phòng và số khu chữa trị. Thuốc men hoặc dụng cụ dùng để chữa bệnh được gọi chung là vật tư, có mã, đặc tả và đơn giá.
- Các **nhân viên** trong bệnh viện có mã số và tên.
- Một **sự chữa trị** là bất kỳ một cuộc **xét nghiệm** hoặc một **công việc điều trị** nào đó mà một bác sĩ thực hiện cho một bệnh nhân. Một sự chữa trị có định danh là một mã số bao gồm số và tên của sự chữa trị đó.



Bài tập

- **Mỗi nhân viên** của bệnh viện *làm việc* trong một hoặc nhiều **khu chữa trị**. Mỗi khu **có** ít nhất một **nhân viên**. Bệnh viện cần thống kê *số giờ làm việc* trong một tuần của mỗi nhân viên tại một khu chữa trị.
- **Mỗi khu chữa trị có một y tá trưởng.**
- **Bệnh nhân ngoại trú** không có giường nằm. Giường có thể không có bệnh nhân. Một bệnh nhân đang được chữa trị có một bác sĩ theo dõi. Một bác sĩ có thể theo dõi nhiều bệnh nhân hoặc không theo dõi bệnh nhân nào.
- Một bác sĩ có thể thực hiện nhiều lần chữa trị cho nhiều bệnh nhân. Một bệnh nhân được chữa trị nhiều lần bởi nhiều bác sĩ. Bệnh viện cần biết mỗi lần chữa trị, bệnh nhân được chữa trị bởi bác sĩ nào, ngày chữa trị, thời gian chữa trị và kết quả.
- Một bệnh nhân cần nhiều thuốc men và dụng cụ. Mỗi loại thuốc men hay dụng cụ có thể được sử dụng cho nhiều bệnh nhân hoặc chưa được sử dụng. Bệnh viện cần thống kê ngày, thời gian, số lượng và tổng số tiền (số lượng x đơn giá) cho một loại thuốc men hay dụng cụ mỗi lần sử dụng cho bệnh nhân.



Quy tắc nghiệp vụ

Business Rules

- **Quy tắc nghiệp vụ** là “*một phát biểu (statement) dùng để định nghĩa hay ràng buộc một số ngữ cảnh của hoạt động nghiệp vụ. Quy tắc này dùng để khẳng định cấu trúc của hoạt động nghiệp vụ hoặc để điều khiển đến hoạt động nghiệp vụ*”.

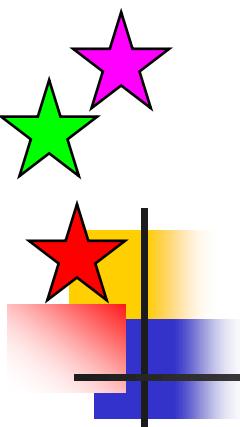
Ví dụ:

- Một sinh viên chỉ được phép đăng ký 1 môn học khi sinh viên đó đã đạt được những môn học tiên quyết cho môn học đó.
- Một khách quen được giảm giá 10% nếu không nợ quá hạn



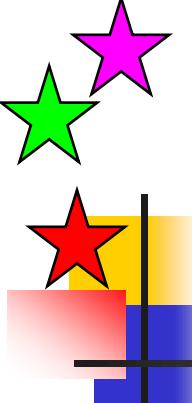
Quy tắc nghiệp vụ

- Thuật ngữ cũ “**data integrity constraints**” (ràng buộc toàn vẹn dữ liệu)
- Thuật ngữ “**business rule**” có phạm vi rộng hơn bao gồm mọi quy tắc có ảnh hưởng đến CSDL trong 1 tổ chức.



Quy tắc nghiệp vụ (tt)

- Mô hình quy tắc nghiệp vụ (business rule paradigm) được xem như mô hình mới trong việc xác định yêu cầu hệ thống thông tin, và có phạm vi rộng hơn.
- Giới hạn phạm vi của quy tắc nghiệp vụ: chỉ quan tâm đến các quy tắc nghiệp vụ có liên quan đến database. Các quy tắc này được thể hiện thông qua các ràng buộc toàn vẹn (**integrity constraint**) trong database.
- Hai loại chính:
 - **Ràng buộc về cấu trúc (structure constraint)**
 - **Ràng buộc về tác vụ (operational constraint)**



Business Rules



Structural Constraint

Operational Constraint

Domain Constraints

Procedural

Declarative

Definitions

Relationships

Terms

Facts

Constraints

SuperType /SubType

Base

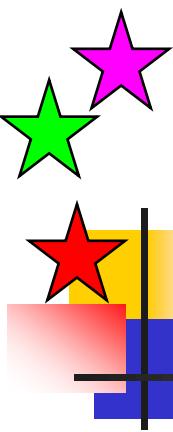
Derived

ER diagram⁴⁶



Phân loại quy tắc nghiệp vụ

- Chỉ có 3 loại quy tắc có thể được thể hiện trong lược đồ ER:
 - Terms → các thực thể, thuộc tính và mối quan hệ
 - Constraints → lượng số min và max
 - Supertype/subtype



Ràng buộc về cấu trúc

- Là các quy luật để ràng buộc về **cấu trúc tĩnh** (static) của một tổ chức.
- Ba loại ràng buộc về cấu trúc:
 - Các định nghĩa (definitions)
 - Miền trị (Domains)
 - Mối liên kết (Relationships)
- Các định nghĩa (definition):
 - Thuật ngữ (term): một từ hay một nhóm từ có ý nghĩa
 - Sự kiện (fact): sự kết hợp giữa hai hay nhiều thuật ngữ
 - Sự kiện dẫn xuất (derived fact): là sự kiện mà dẫn xuất ra từ những sự kiện hoặc quy tắc khác



Ràng buộc về cấu trúc (tt)

- Ràng buộc miền trị (domain): xác định tập các giá trị mà một hay nhiều thuộc tính có thể lấy.
- Ví dụ:
 - Xét thực thể SINHVIEN có các thuộc tính MaSv, TenSV, Phai (Ràng buộc miền trị là chỉ có 2 giá trị Nam hay Nữ), Diem (từ 0 đến 10).
 - Thực thể HOADON có các thuộc tính MaHD, LoaiHD (RBMT chỉ chứa các giá trị N, X, C, T).



Ràng buộc về tác vụ

- Là các quy tắc dùng để ràng buộc những tác vụ nghiệp vụ đang xảy ra
- Trước đây các ràng buộc tác vụ được thực hiện trong các thủ tục nằm sâu trong chương trình ứng dụng → khó sửa đổi
- Phương pháp mới: dùng khai báo (declarative approach) để xác định các quy tắc nghiệp vụ.
- Mỗi quy tắc được phát biểu như 1 sự khẳng định (assertion) mà không xác định xem quy luật đó thực thi như thế nào.
- Tất cả các quy tắc sẽ được lưu trữ trong một cơ sở ràng buộc (constraint base). Khi DBMS xử lý 1 transaction, nó truy xuất đến các quy tắc thích hợp trong cơ sở ràng buộc này để áp dụng cho transaction.



Ngôn ngữ để xác định ràng buộc

- Mỗi quy tắc sẽ được xác định bằng cú pháp của 1 ngôn ngữ đặc biệt có 2 tính chất sau:
 - Phải khá đơn giản để người dùng (end user) không chỉ hiểu được mà còn có thể tự mình tạo ra các quy tắc từ ngôn ngữ này
 - Ngôn ngữ phải có cấu trúc thích đáng để có thể chuyển đổi tự động thành mã máy
- Trong SQL server 2008, các quy tắc được thực hiện thông qua các constraint trong bảng và trigger.



Các đối tượng bị ràng buộc và đối tượng ràng buộc

- **Đối tượng bị ràng buộc (constrained object):** là 1 thực thể, thuộc tính hay mối quan hệ mà các thao tác (như tạo, xóa, cập nhật, đọc,...) trên đối tượng đó bị giới hạn
- **Đối tượng ràng buộc (constraining object):** là 1 thực thể, thuộc tính, hay mối quan hệ mà tác động đến khả năng thực thi tác vụ của 1 đối tượng khác

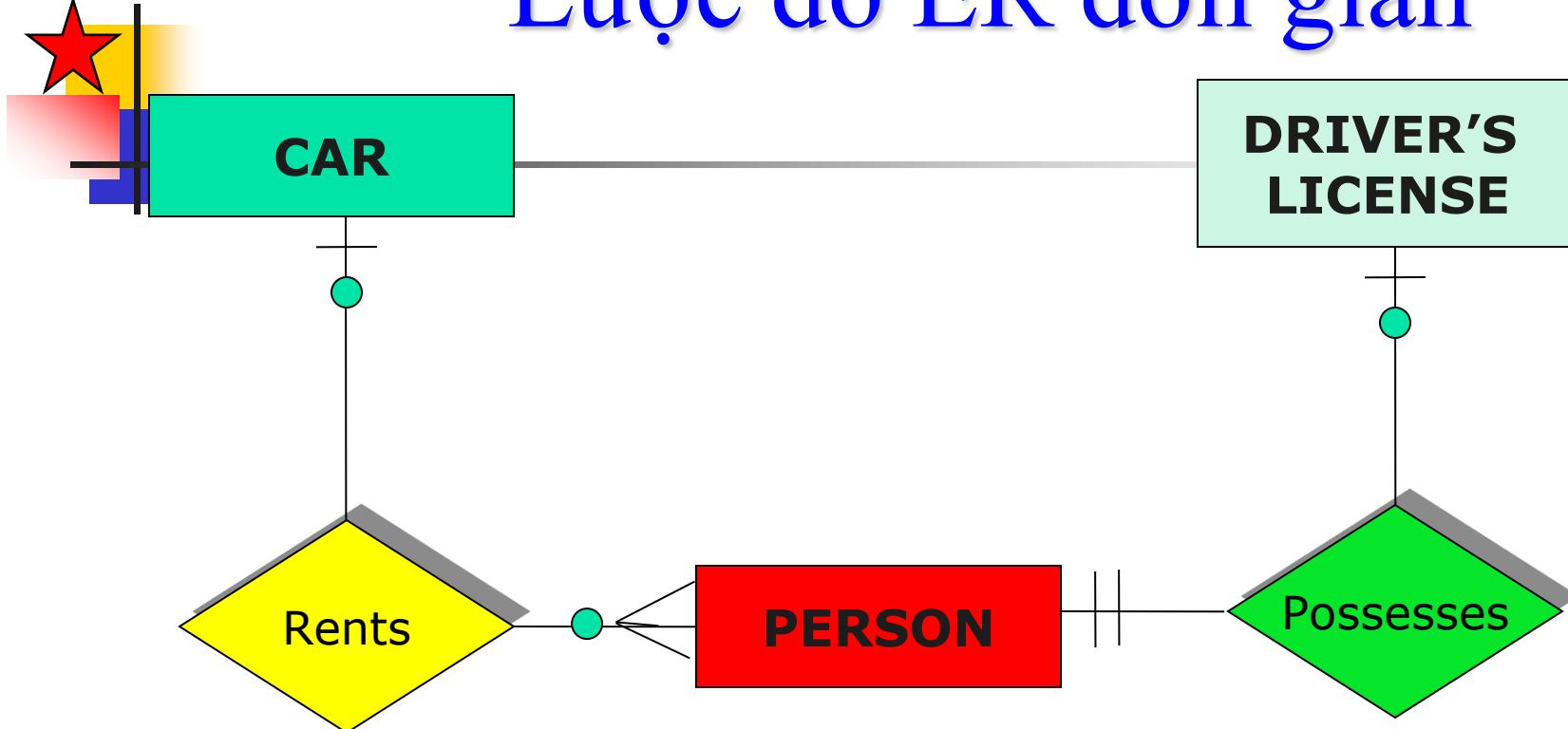


Các đối tượng bị ràng buộc và đối tượng ràng buộc

- Ví dụ: Xét quy tắc nghiệp vụ sau: “ A person can rent a car only if he or she possesses a valid driver’s license”
 - ➔ 3 thực thể: PERSON, CAR, DRIVER’S LICENSE
 - ➔ 2 mối kết nối: Rents (1-M optional), Possesses (1-1 optional)



Lược đồ ER đơn giản



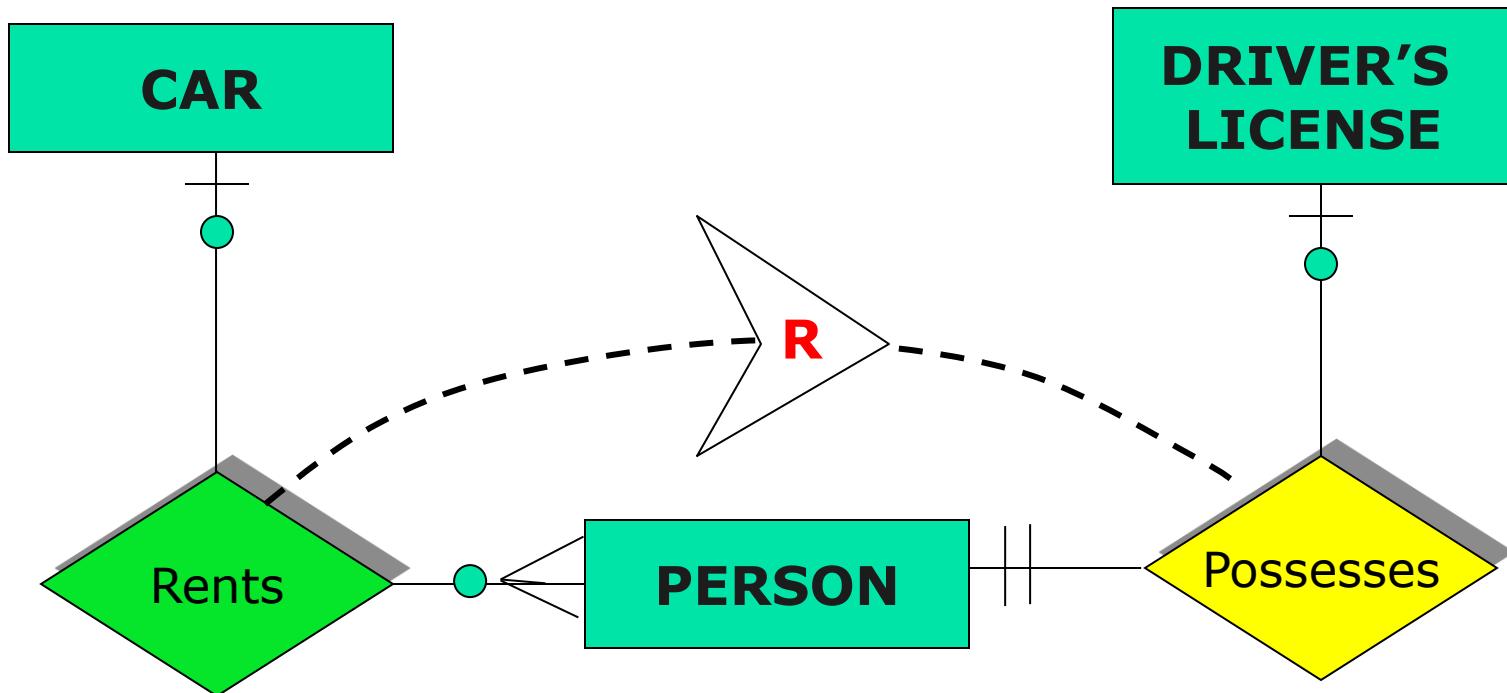
Chỉ ra cấu trúc của ngữ cảnh nhưng không chỉ ra
được các ràng buộc giữa các đối tượng

Đối tượng nào là đối tượng bị ràng buộc?? **Rents**

Đối tượng nào là đối tượng ràng buộc?? **Possesses**



Lược đồ ER có ràng buộc

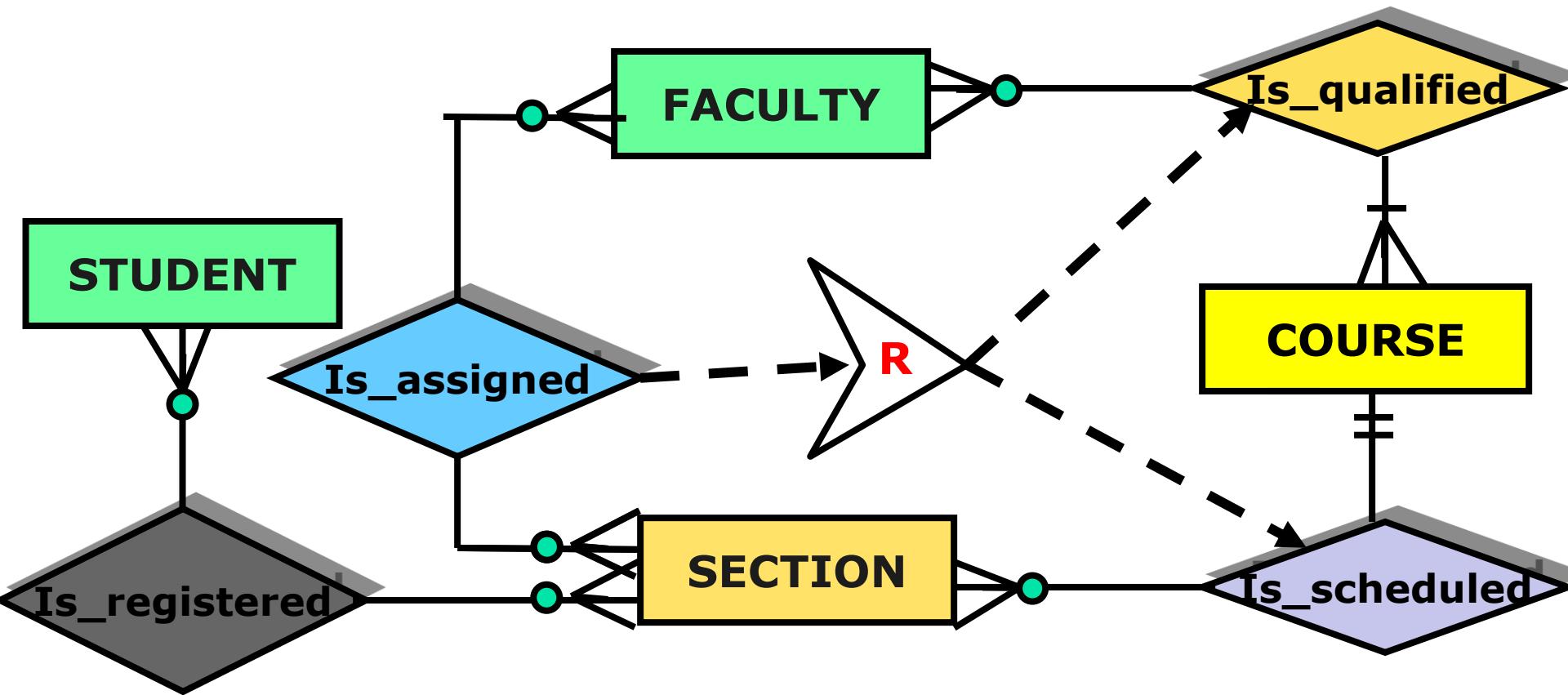




Ví dụ 2

- Bài toán lập lịch lớp học (class scheduling): quy tắc nghiệp vụ như sau:
 - For a faculty member to be assigned to teach a section of a course, the faculty member must be qualified to teach the course for which that section is scheduled
- 3 entities: FACULTY, COURSE, SECTION
- 1 Constrained entity: Is_assigned
- 2 Constraining entities: Is_qualified, Is_Scheduled

Ví dụ 2





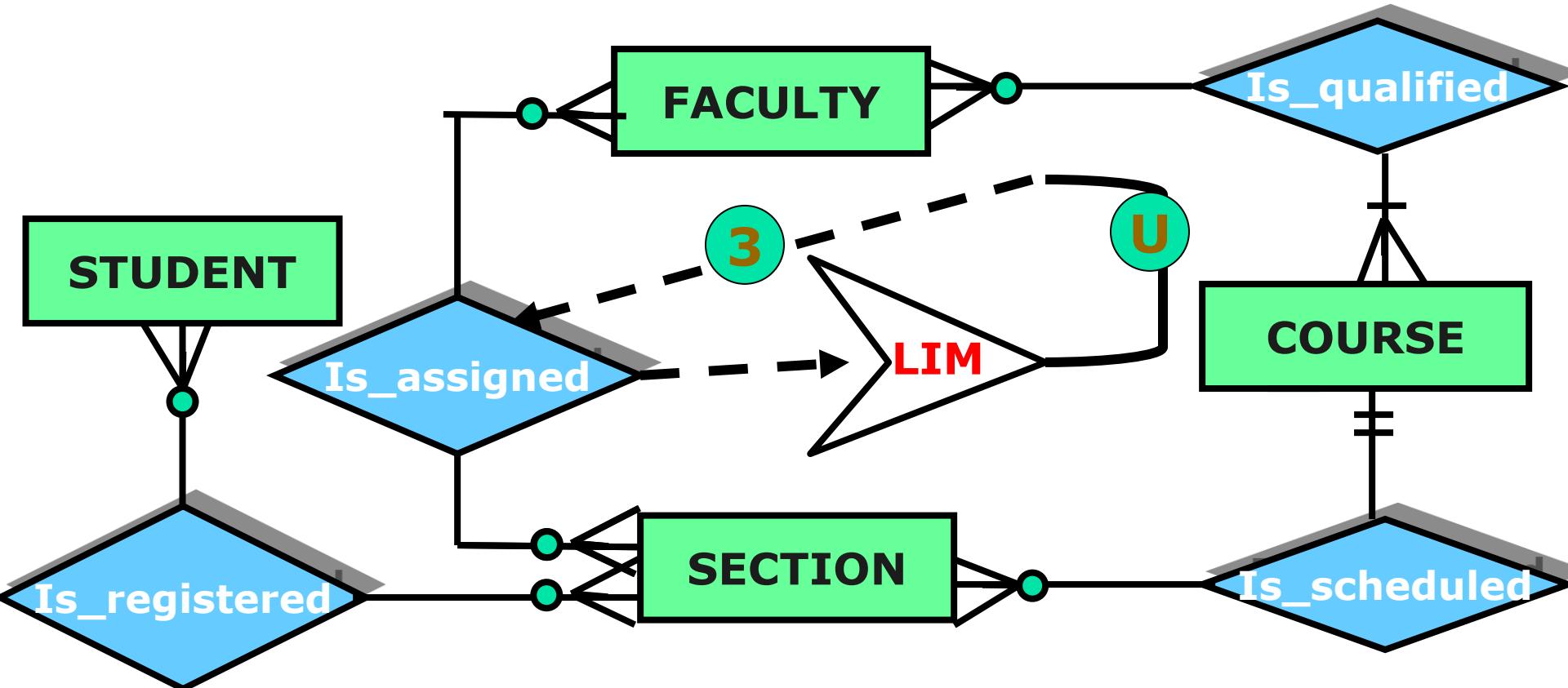
Ví dụ 2

- Quy tắc 2: For a faculty member to be assigned to teach a section of a course, the faculty member must not be assigned to teach a total of more than three course sections

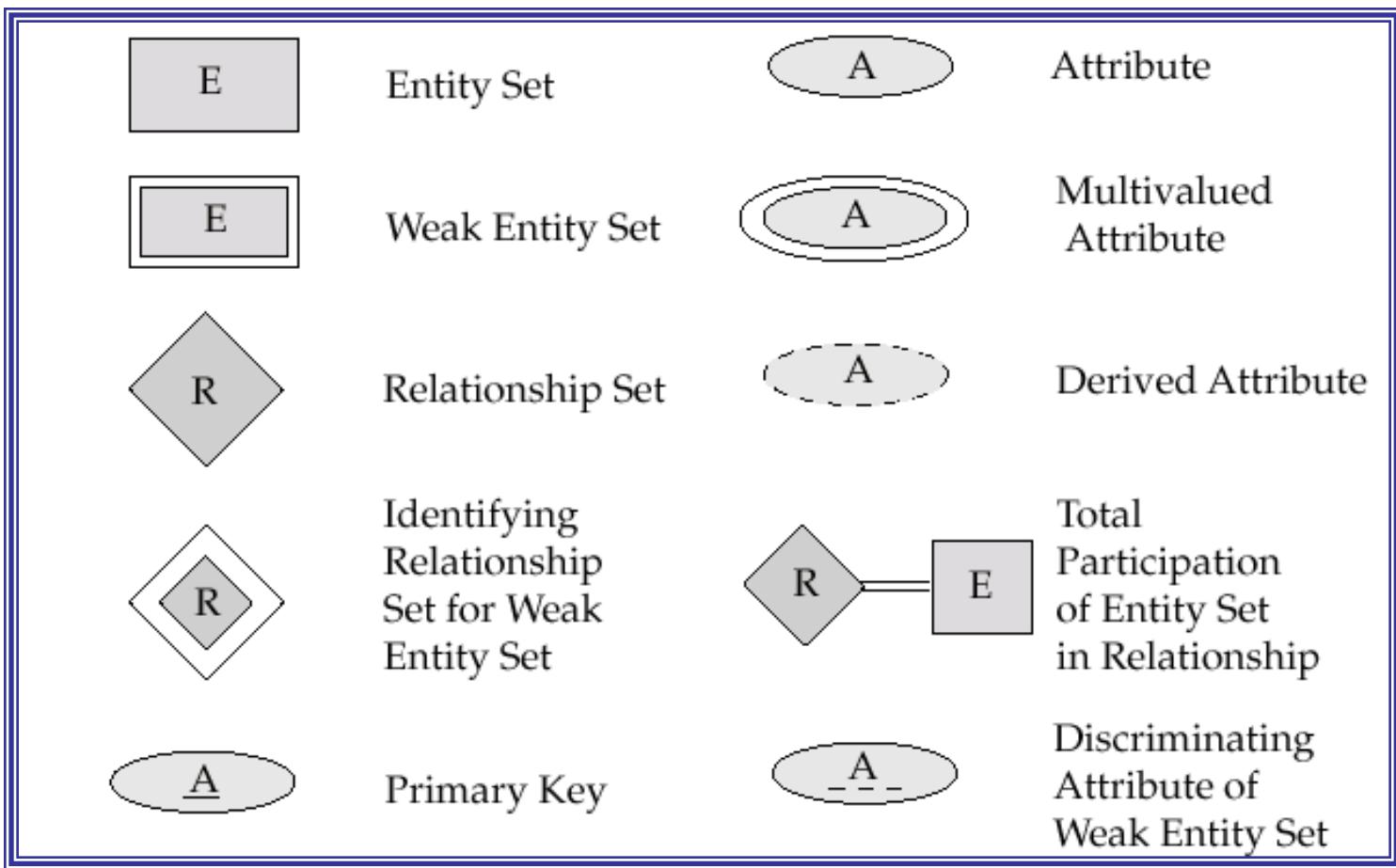
➔ Constrained entity: Is_assigned

➔ Constraining entity: Is_assigned

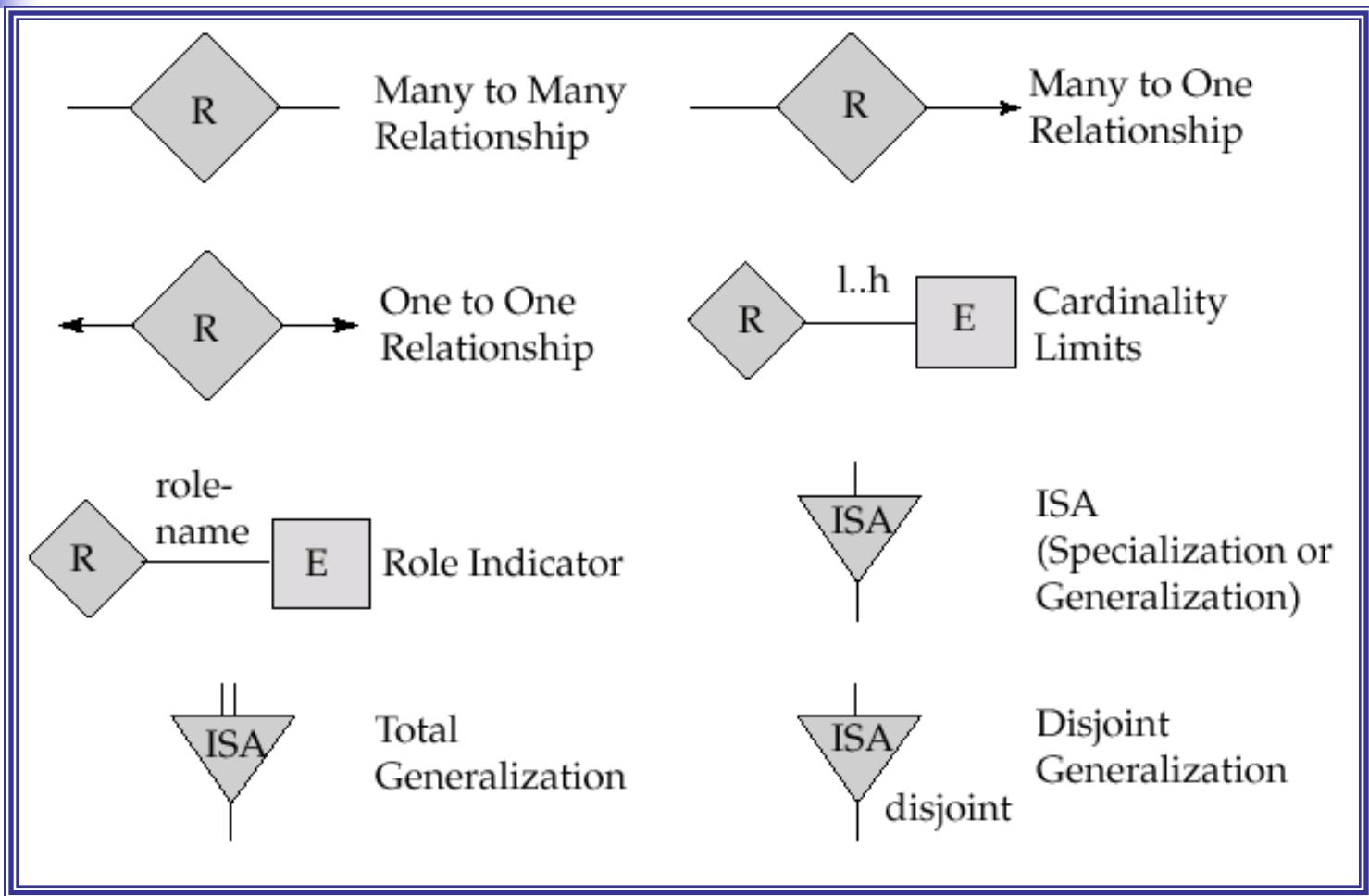
Ví dụ 2



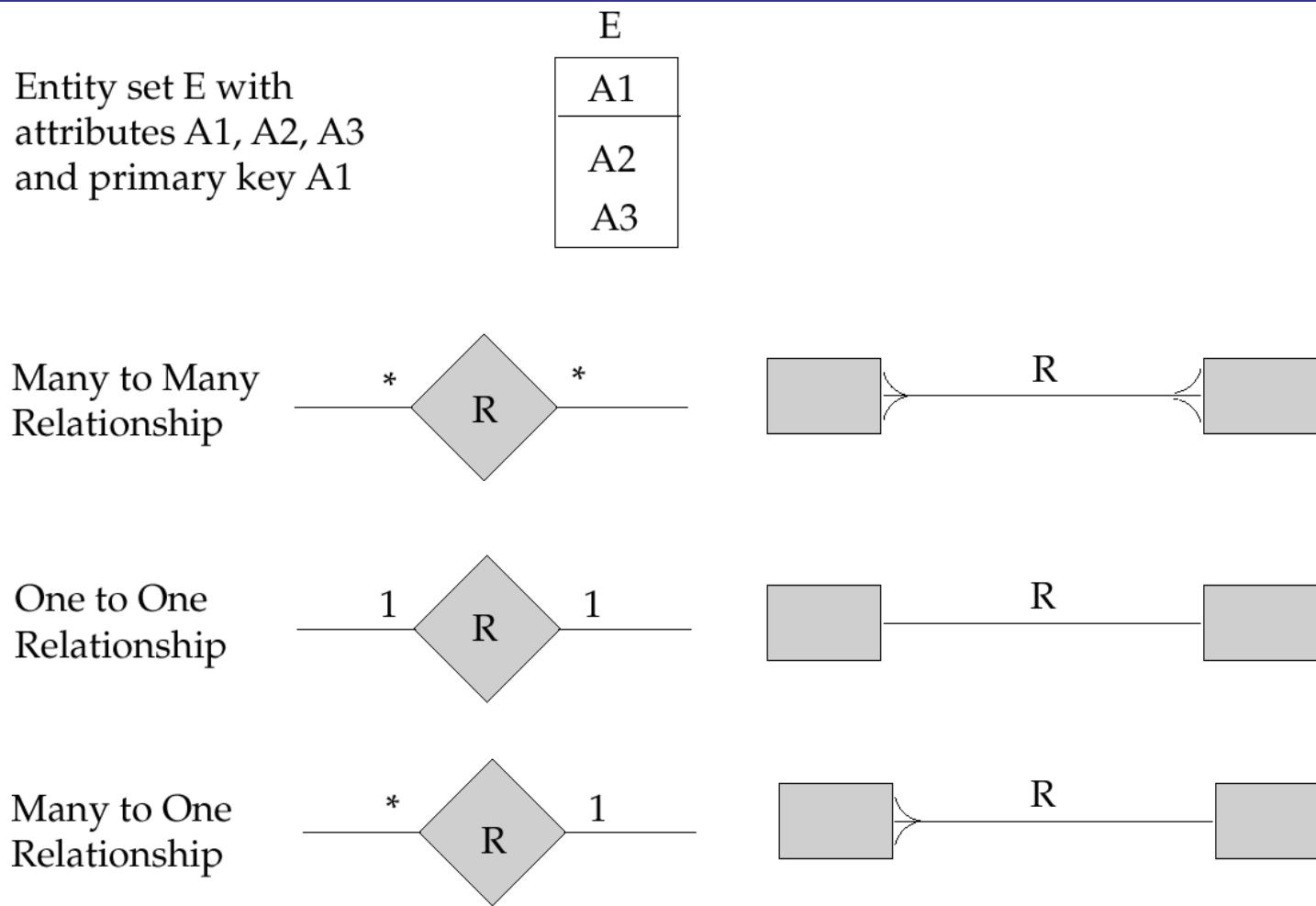
Tổng kết: Các ký hiệu dùng trong mô hình ER

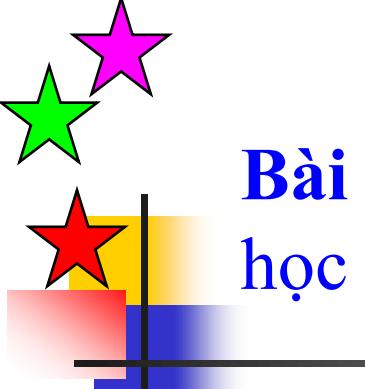


Tổng kết: Các ký hiệu dùng trong mô hình ER



Tổng kết: Các ký hiệu dùng trong mô hình ER





Xây dựng mô hình ER

Bài tập 1: Quản lý hoạt động của một trung tâm đại học

Qua quá trình khảo sát, điều tra hoạt động của một trung tâm đại học ta rút ra các quy tắc quản lý sau:

- Trung tâm được chia làm nhiều **trường** và mỗi trường có 1 **hiệu trưởng** để quản lý nhà trường.
- Một trường chia làm nhiều **khoa**, mỗi khoa thuộc về một trường.
- Mỗi khoa cung cấp nhiều **môn học**. Mỗi môn học thuộc về 1 khoa (thuộc quyền quản lý của 1 khoa).
- Mỗi khoa thuê nhiều giáo viên làm việc. Nhưng mỗi **giáo viên** chỉ làm việc cho 1 khoa. Mỗi khoa có 1 chủ nhiệm khoa, đó là một giáo viên.
- Mỗi giáo viên có thể dạy nhiều nhất 4 môn học và có thể không dạy môn học nào.
- Mỗi sinh viên có thể học nhiều môn học, nhưng ít nhất là môn. Mỗi môn học có thể có nhiều sinh viên học, có thể không có sinh viên nào.
- Một khoa quản lý nhiều sinh viên chỉ thuộc về một khoa.
- Mỗi giáo viên có thể được cử làm chủ nhiệm của lớp, lớp đó có thể có nhiều nhất 100 sinh viên.



Xây dựng mô hình ER

Bài tập 2: Quản lý nhân viên cho một đơn vị

1. Thuộc tính:

- Mã đơn vị, Tên đơn vị, Số điện thoại đơn vị, Địa chỉ đơn vị.
- Mã nhân viên, Tên nhân viên, Giới tính nhân viên, Địa chỉ nhân viên,
- Số điện thoại của nhân viên.
- Mã dự án, Tên dự án
- Mã khách hàng, tên khách hàng, Địa chỉ khách hàng, Số điện thoại của khách hàng.
- Mã hàng, Tên hàng, Số lượng trong kho.
- Lượng đặt hàng, Ngày đặt hàng

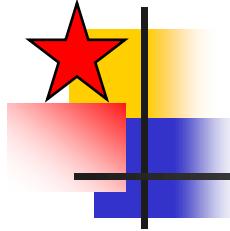


Xây dựng mô hình ER

Bài tập 2: Quản lý nhân viên cho một đơn vị

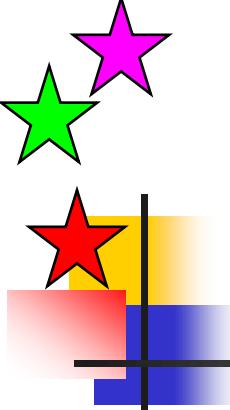
2. Các quy tắc

- Một đơn vị thuê 1 hoặc nhiều nhân viên
- Một đơn vị được quản lý bởi 1 người quản lý. Đó là một nhân viên.
- Một nhân viên chỉ làm việc cho 1 đơn vị
- Một nhân viên có thể làm việc cho 1 dự án
- Mỗi dự án có thể thuê 1 hoặc nhiều nhân viên
- Một nhân viên có thể phục vụ cho 1 hoặc nhiều khách hàng
- Một khách hàng có thể được 1 hoặc nhiều nhân viên phục vụ
- Một khách hàng có thể đặt 1 hoặc 1 vài hàng hóa (Khách hàng nào cũng đặt hàng: 1 hoặc nhiều mặt hàng)
- Mọi mặt hàng đều có ít nhất một khách hàng đặt mua
- Một đơn đặt hàng chỉ có 1 mặt hàng.



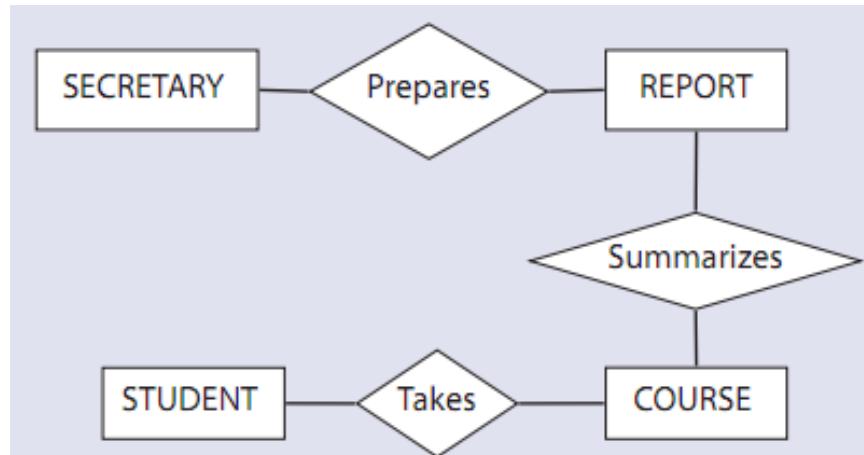
Review Questions

1. What is a subclass? When is a subclass needed in data modeling?
2. Define the following terms: superclass of a subclass, superclass/subclass relationship, IS-A relationship, specialization, generalization, category, specific (local) attributes, specific relationships.
3. What is the difference between a specialization hierarchy and a specialization lattice?



Review Questions

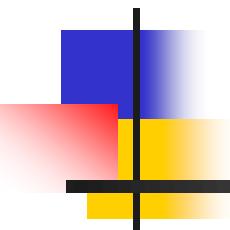
4. A department in a university stores the information about its students and courses in a database. The administrative assistant manages the database. At the end of the semester, he prepares a report about each course. Is the E-R diagram correct? If not, explain why and draw the correct diagram.





Review Questions

5. Consider the BANK ER schema of Figure 03.17, and suppose that it is necessary to keep track of different types of ACCOUNTS (SAVINGS_ACCTS, CHECKING_ACCTS, . . .) and LOANS (CAR_LOANS, HOME_LOANS, . . .). Suppose that it is also desirable to keep track of each account's TRANSACTIONS (deposits, withdrawals, checks, . . .) and each loan's PAYMENTS; both of these include the amount, date, and time. Modify the BANK schema, using ER and EER concepts of specialization and generalization. State any assumptions you make about the additional requirements.



Thank you

