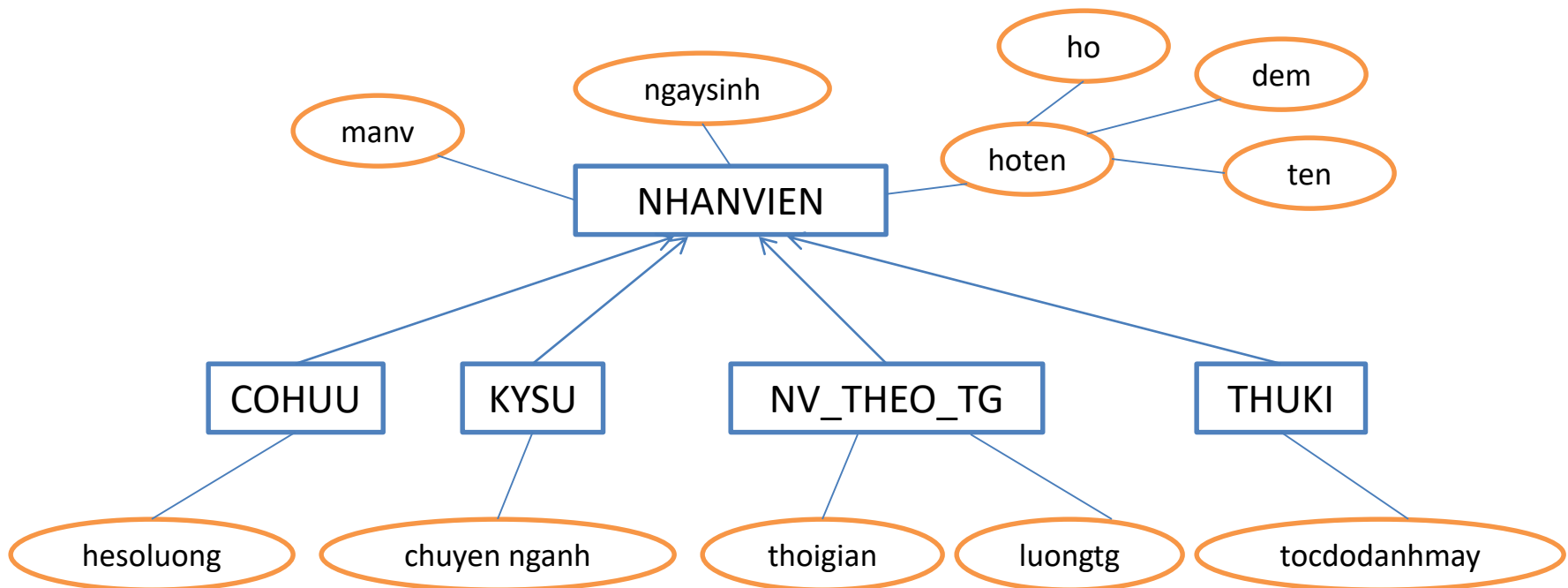


2.3- Mô hình thực thể - Liên kết mở rộng (EER:Enhanced Entity-Relationship)

• Tại sao cần EER?

Ví dụ: NHANVIEN gồm các loại: nhân viên cơ hữu (hưởng lương tháng), nhân viên hưởng theo thời gian, kỹ sư, thư kí,..



2.3 - Mô hình thực thể - Liên kết mở rộng (EER:Enhanced Entity-Relationship)

● Tại sao cần EER?

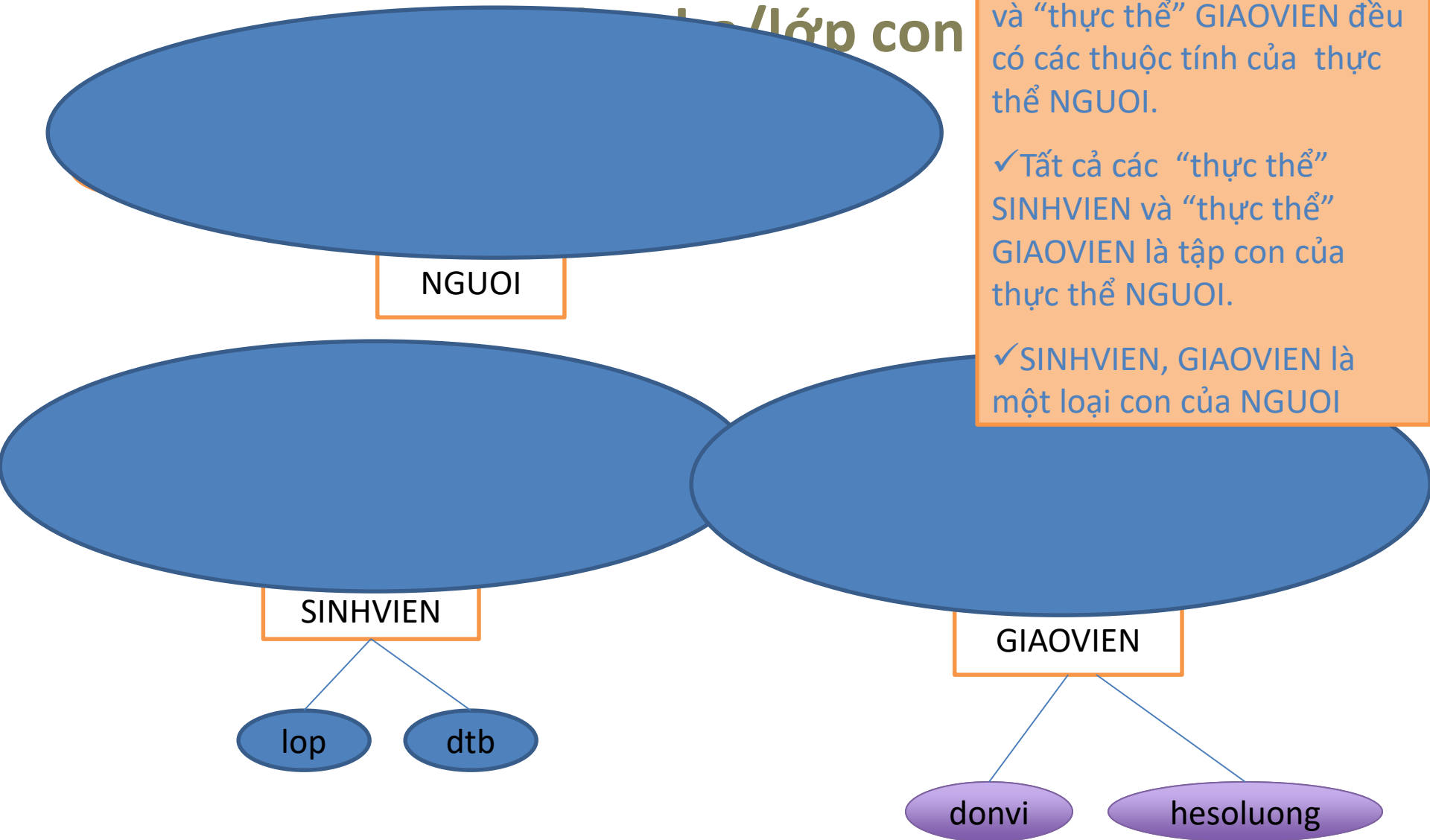
- ✓ Các khái niệm cơ bản về ER không đủ để biểu diễn một số các ứng dụng phức tạp,
- ✓ Cần thêm một số khái niệm để tăng khả năng mô tả đối tượng rõ ràng và chính xác hơn,
- ✓ EER là sự mở rộng của ER bằng cách thêm vào một số các khái niệm trừu tượng (abstraction) và thể hiện các ràng buộc rõ ràng hơn.

2.3 - Mô hình thực thể - liên kết mở rộng

Một số khái niệm cơ bản của EER

1. Lớp cha/con
2. Chuyên biệt hóa và Tổng quát hóa
3. Các ràng buộc trên chuyên biệt hóa
4. Phân cấp chuyên biệt và lưới
5. Giới thiệu về kiểu hợp (Union type)

2.3 - Mô hình thực thể - liên kết mở rộng



✓ Các “thực thể” SINHVIEN và “thực thể” GIAOVIEN đều có các thuộc tính của thực thể NGUOI.

✓ Tất cả các “thực thể” SINHVIEN và “thực thể” GIAOVIEN là tập con của thực thể NGUOI.

✓ SINHVIEN, GIAOVIEN là một loại con của NGUOI

■ Lớp cha/lớp con

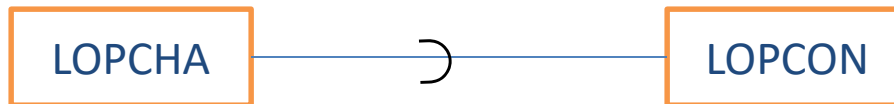


- Lớp con: Quản lý, kế toán, thư ký....
- Lớp cha: Nhân viên

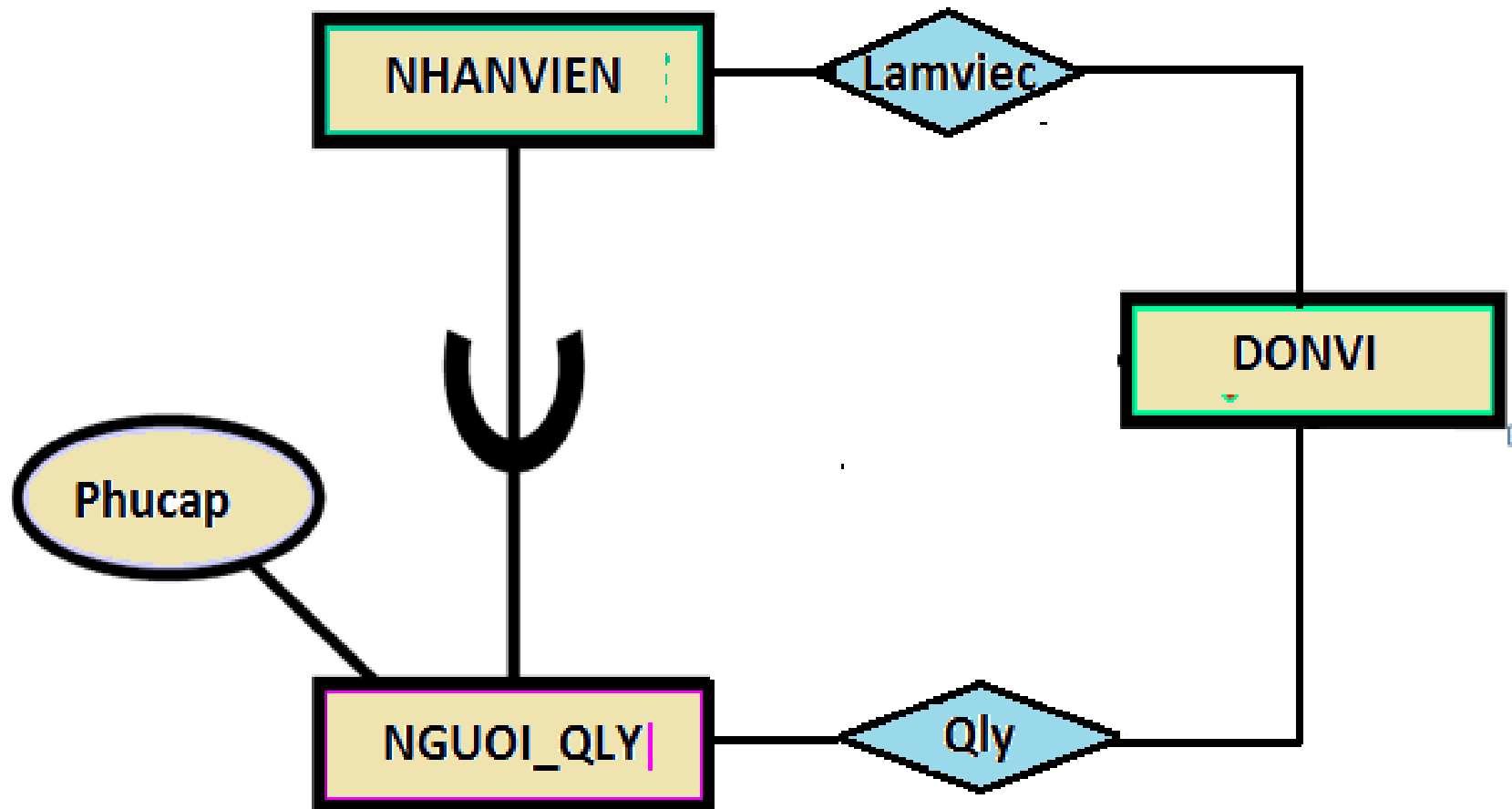
- Lớp cha: là kiểu thực thể bao gồm các thực thể mang đặc tính chung (tổng quát hóa) cho các nhóm thực thể.
- Lớp con: là các thực thể thành viên của lớp cha nhưng có vai trò riêng biệt (được chuyên biệt hóa).
- Một thực thể không thể tồn tại chỉ trong vai trò của lớp con.

■ Tính kế thừa

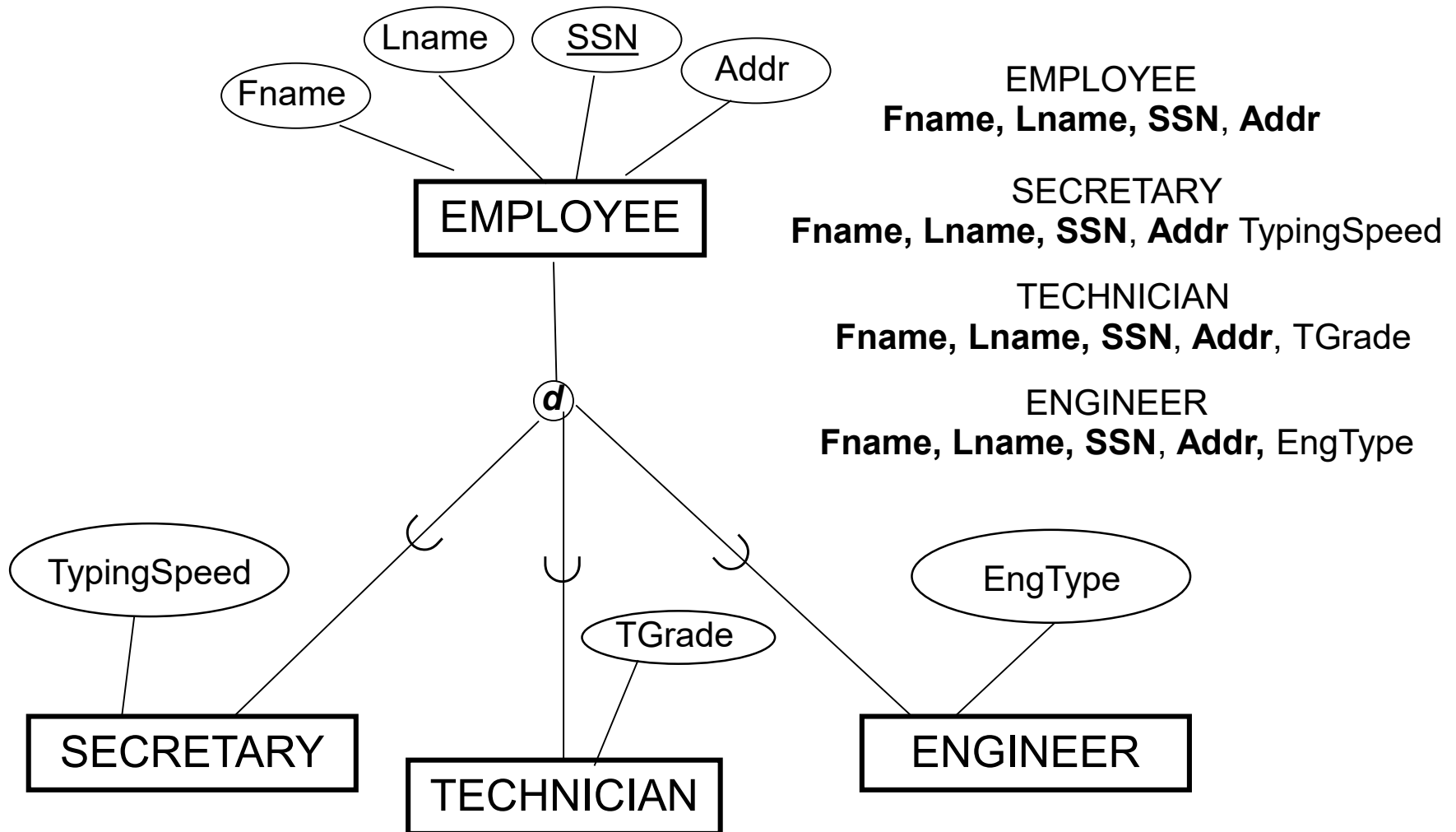
- Lớp con thừa kế thuộc tính và quan hệ của lớp cha và có thể có một số thuộc tính và quan hệ của riêng nó
- Lợi ích
 - Tránh mô tả các định nghĩa trùng lặp nhau
 - Thêm thông tin về ngữ nghĩa vào trong thiết kế
- Mỗi quan hệ giữa lớp cha và lớp con gọi là *kiểu liên kết ISA, là liên kết 1:1*
- *Kí hiệu*



Ví dụ: Lớp Nhân viên và lớp Người quản lý



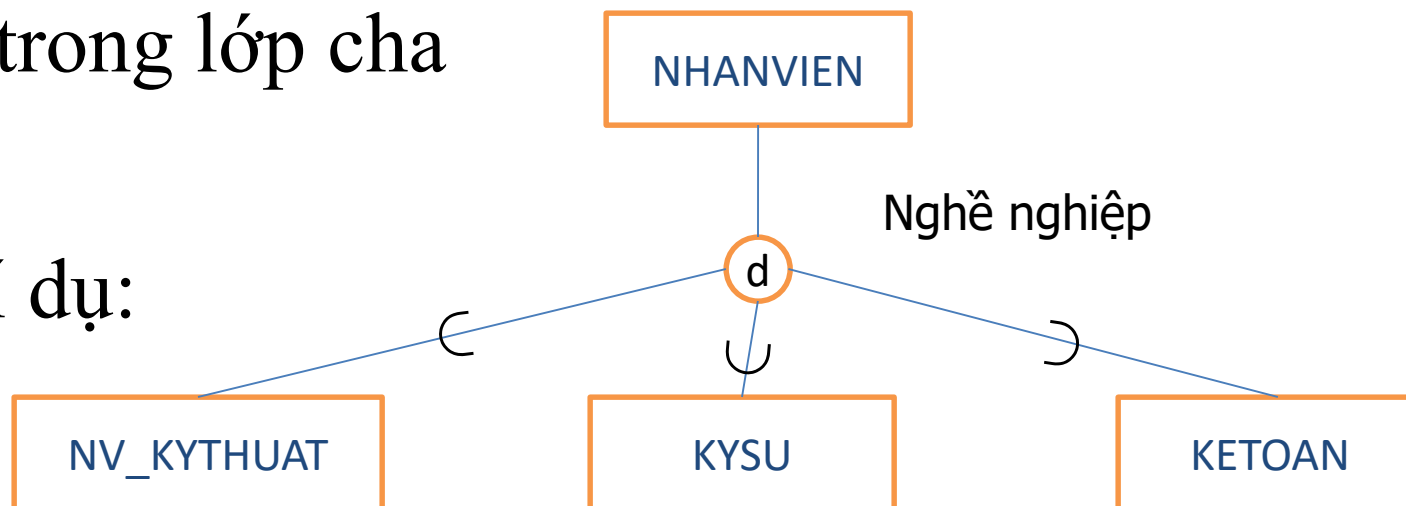
Ví dụ



■ Chuyên biệt hóa (cụ thể hóa)

- Là quá trình xác định tập các lớp con của một kiểu thực thể (lớp cha)
- Tập các lớp con được tạo ra dựa trên một số các đặc tính riêng biệt nào đó của các thực thể trong lớp cha

Ví dụ:



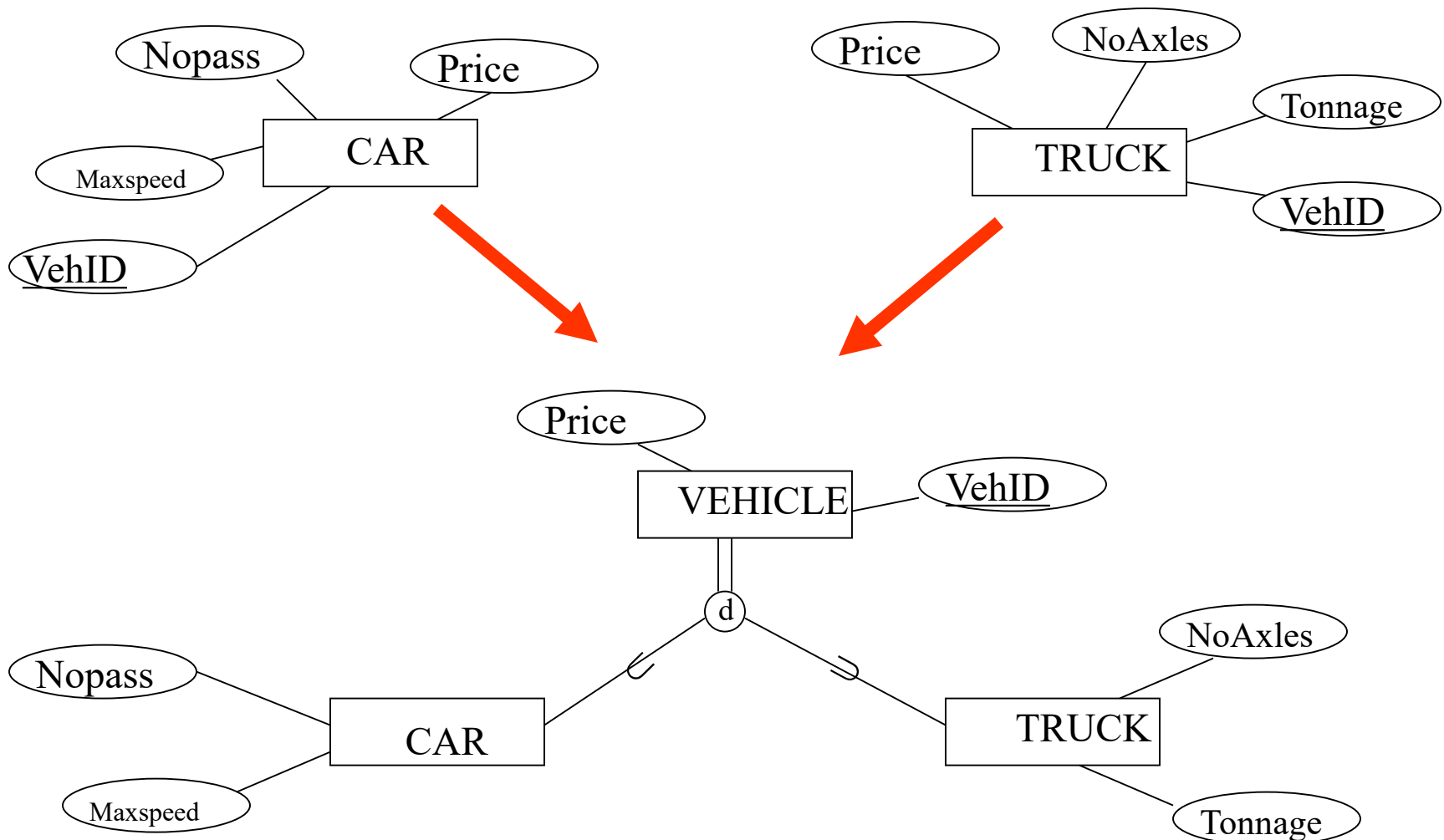
■ Tổng quát hóa

- Quá trình xác định lớp cha từ các lớp con có chung một số thuộc tính.
- Một số các lớp chia sẻ một số thuộc tính chung được nhóm lại vào một lớp cha.


Ví dụ

- **Sinh viên** và **giáo viên** chia sẻ nhau chung một số thuộc tính như tên, tuổi, địa chỉ, giới tính,
- Nhóm lại thành 1 lớp cha: **Người**
- *Chuyên biệt hóa và Tổng quát hóa là 2 cách tiếp ngược nhau*
 - *Tổng quát hóa = bottom up*
 - *Chuyên biệt hóa = Top down*

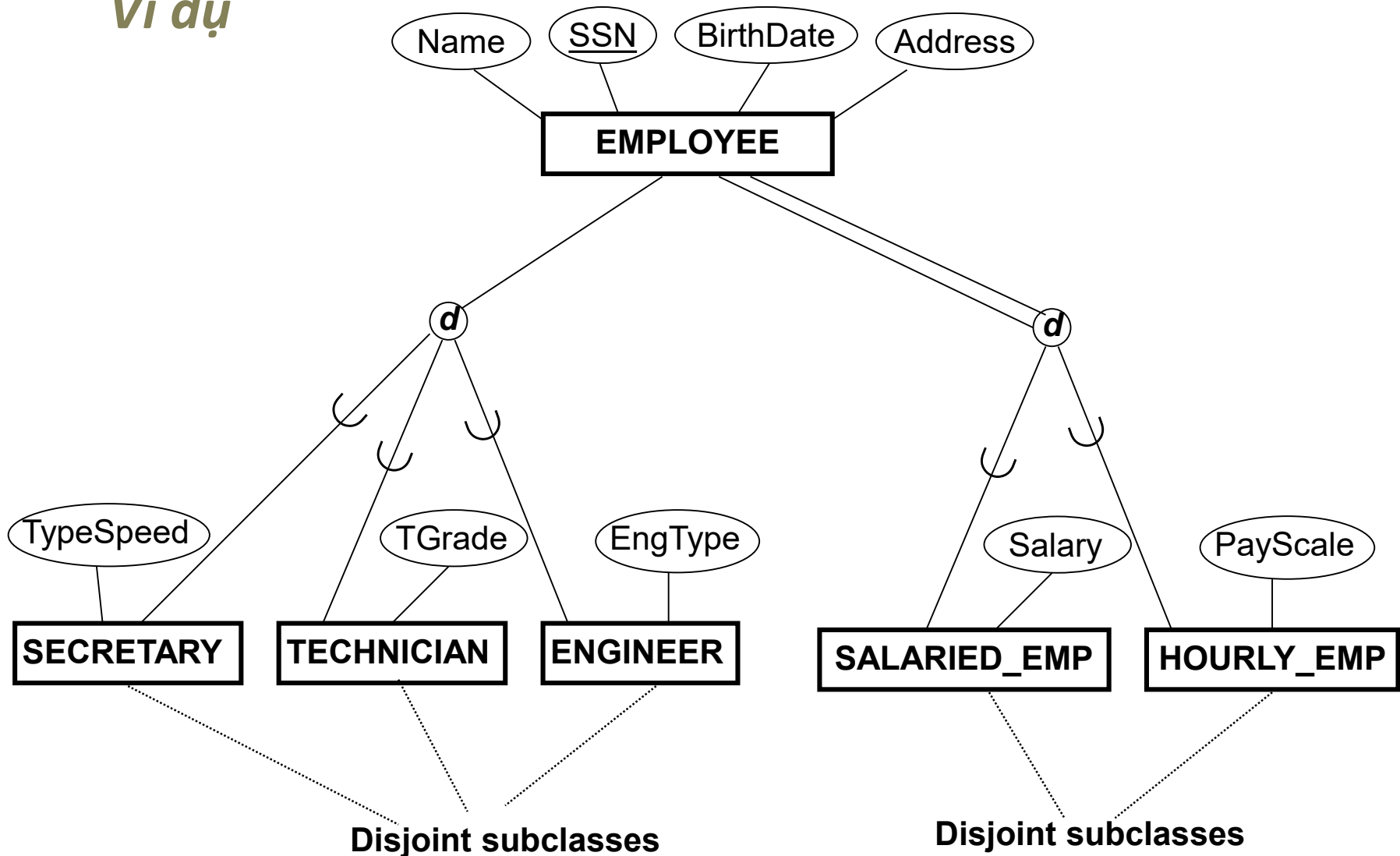
Ví dụ




■ Ràng buộc rời rạc

- Ràng buộc rời rạc(disjointness constraint) mô tả quan hệ giữa lớp cha và các lớp con; các lớp con phải độc lập hoàn toàn với nhau(một thực thể là thành viên của lớp cha chỉ thuộc một lớp con được chuyên biệt hóa)
- Trong lược đồ EER, ràng buộc rời rạc được ký hiệu bởi chữ d (**d**isjoint) nằm trong vòng tròn  d

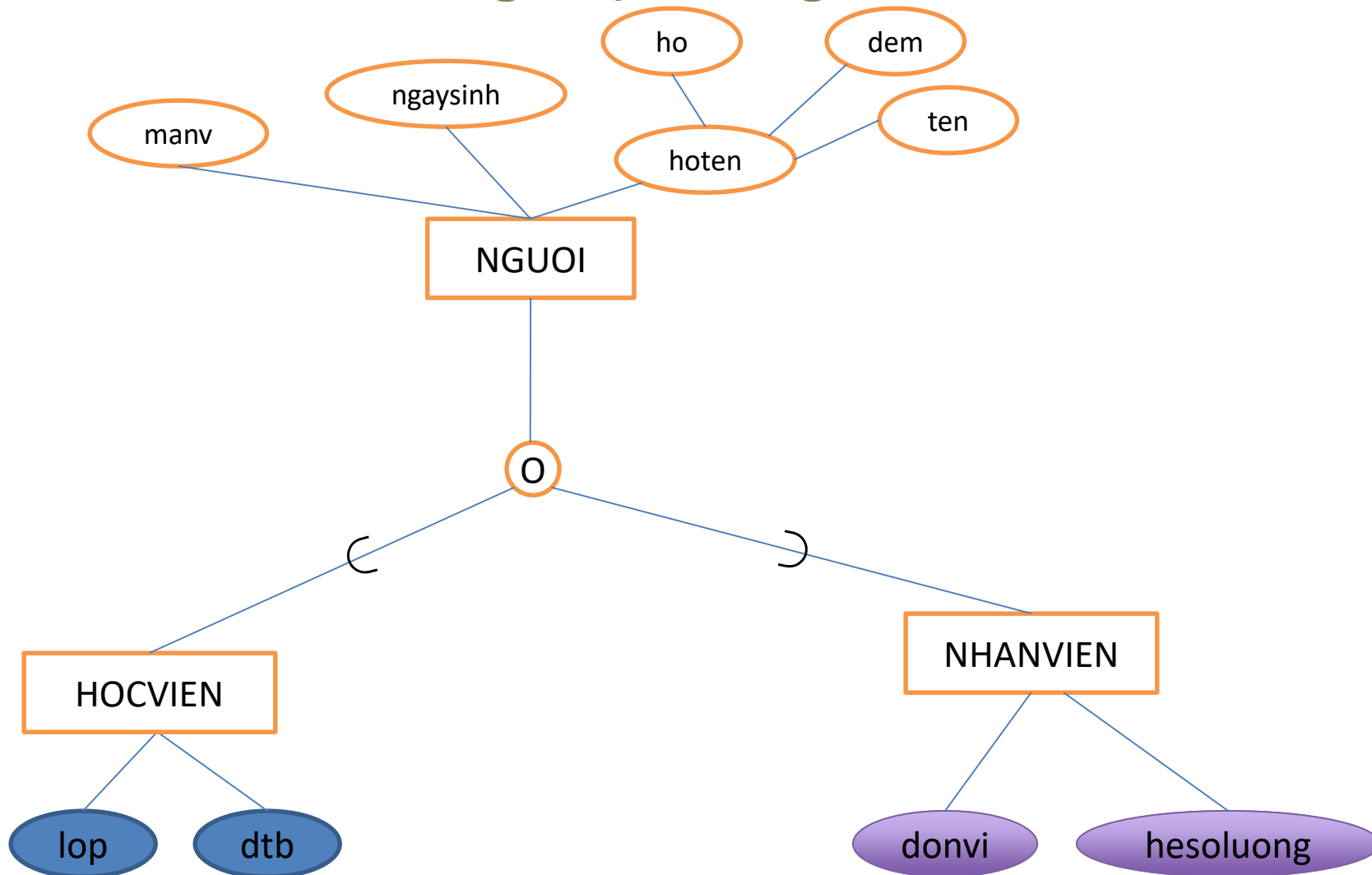
Ví dụ



■ Ràng buộc chồng chéo

- Ràng buộc chồng chéo (Overlapping constraint) cho biết quan hệ giữa lớp cha và các thực thể ở lớp con; các thực thể lớp con là không tách biệt (một thực thể có thể là thành viên của nhiều lớp con theo sự chuyên biệt hóa)
- Trong lược đồ EER, ràng buộc này được mô tả bởi chữ o (overlap) bên trong vòng tròn 

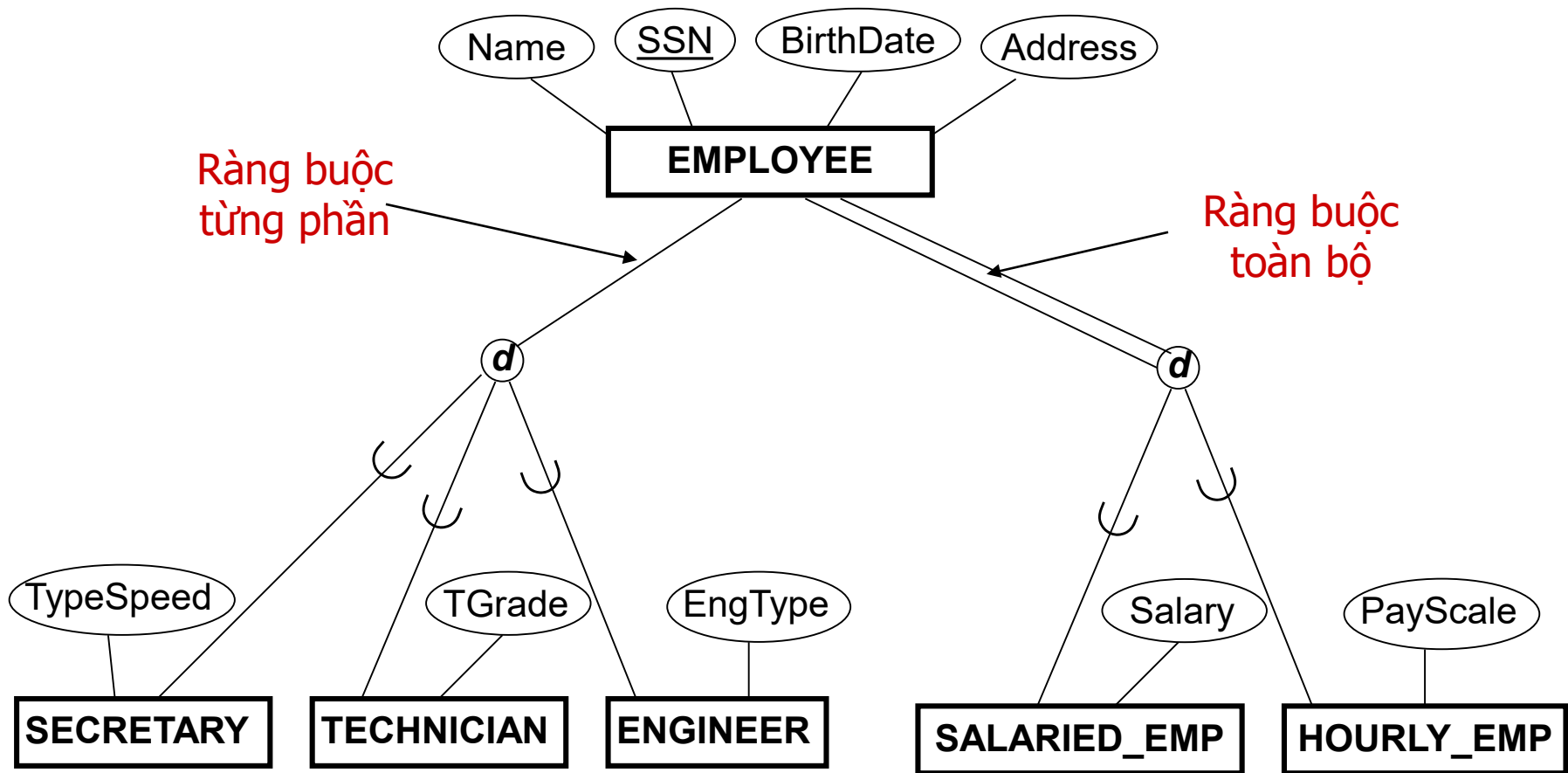
■ Ràng buộc chéo



■ Ràng buộc đầy đủ

- **Ràng buộc đầy đủ** (completeness constraint) bao gồm
 - **Ràng buộc toàn bộ** (total) cho biết tất cả các thực thể trong lớp cha phải là thành viên của ít nhất một lớp con nào đó trong chuyên biệt
 - Thể hiện bằng một đường nét đôi nối giữa lớp cha và vòng tròn chuyên biệt
 - **Ràng buộc từng phần** (partial) cho phép một thực thể ở lớp cha có thể không thuộc bất kỳ một lớp con nào trong chuyên biệt
 - Thể hiện bằng đường nét đơn

Ví dụ ràng buộc đầy đủ

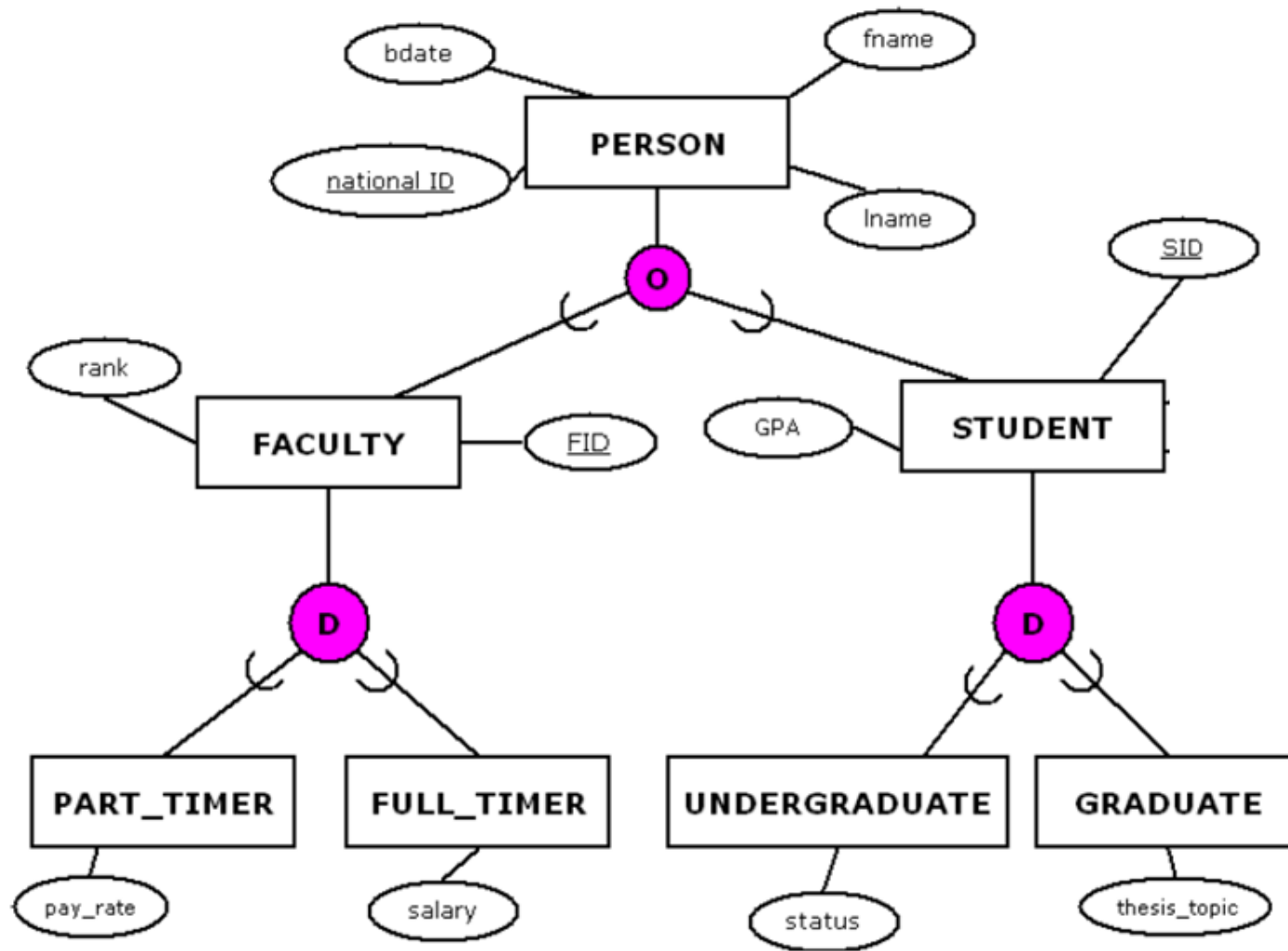


- Có 4 khả năng ràng buộc
 - Disjoint, total
 - Disjoint, partial
 - Overlapping, total
 - Overlapping, partial
- Các ràng buộc độc lập với nhau

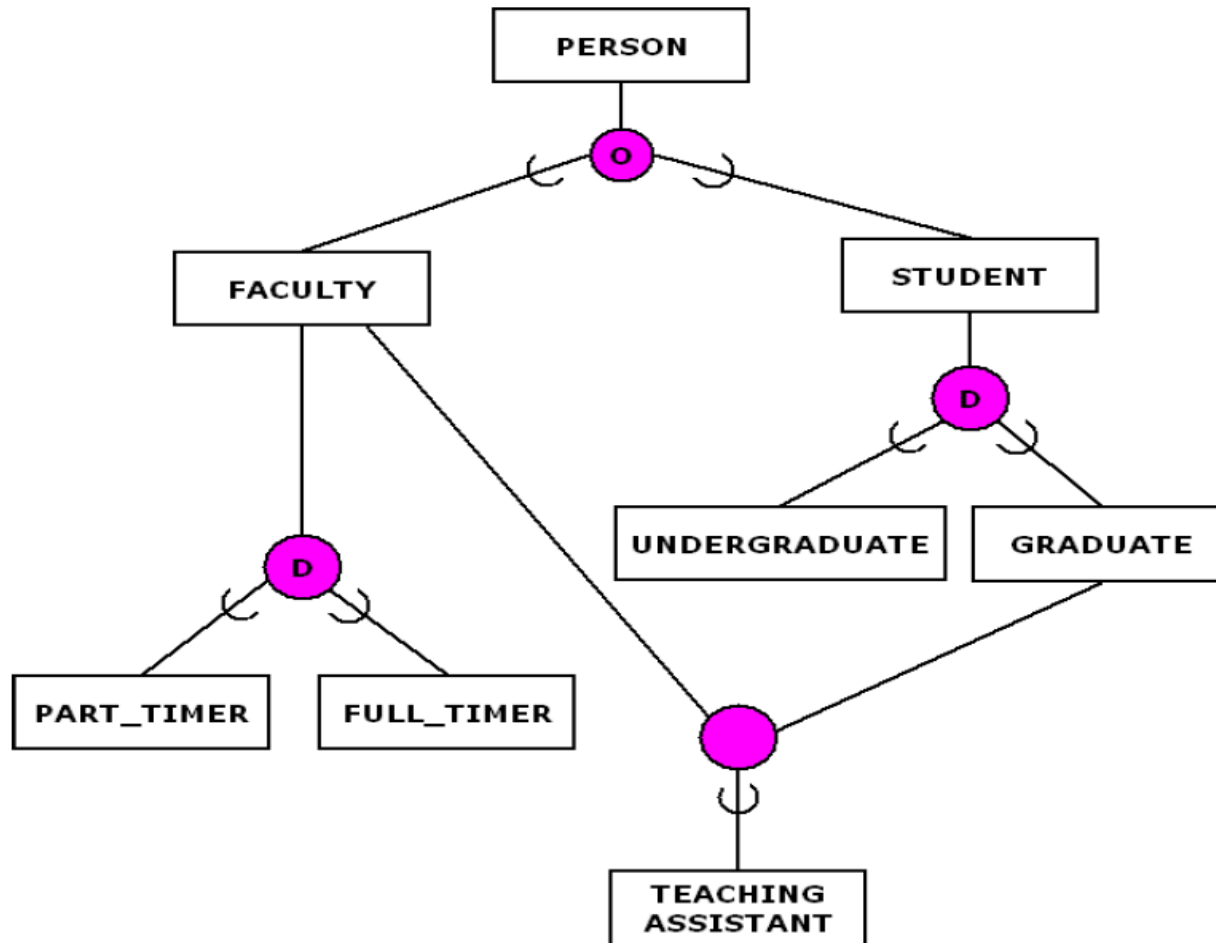
■ Chuyên biệt phân cấp và lưới

- Trong chuỗi quan hệ chuyên biệt, một lớp con có thể lại là lớp cha của lớp khác :
 - **Phân cấp** (hierarchy) là ràng buộc trong đó tất cả các lớp con chỉ tham gia vào 1 liên kết lớp cha/con (thừa kế đơn ánh)
 - **Lưới** (Lattice) là ràng buộc trong đó lớp con có thể tham gia vào nhiều hơn 1 liên kết cha/con (thừa kế bội)
- Lớp con không chỉ kế thừa thuộc tính của lớp cha trực tiếp mà còn kế thừa thuộc tính của lớp bên trên (cha của lớp cha nó)

Ví dụ chuyên biệt phân cấp



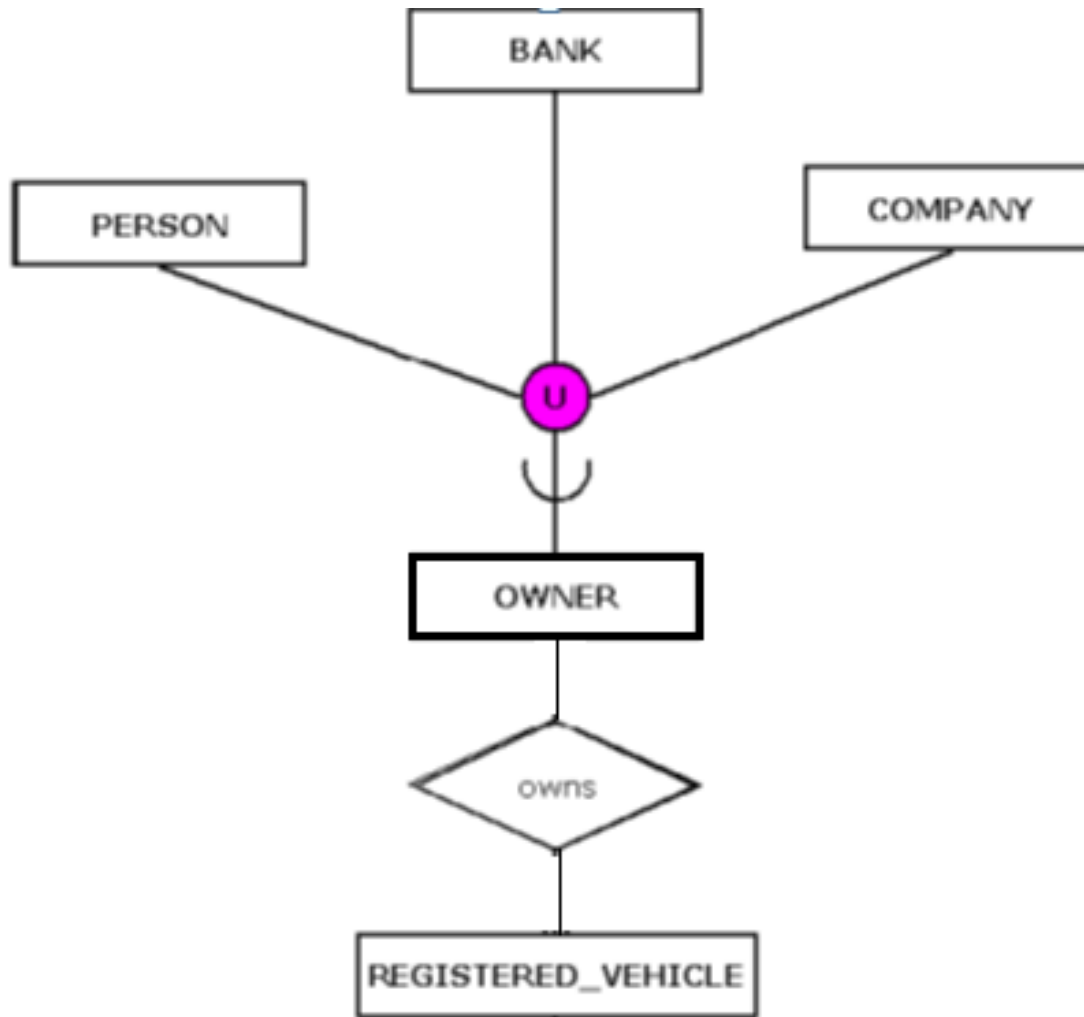
Ví dụ Chuyên biệt lười



■ Kiểu hợp (union)

- Trong chuyên biệt lưới lớp con thừa kế từ nhiều lớp cha, thừa hưởng tất cả các thuộc tính của lớp cha
- Làm thế nào để mô tả quan hệ lớp con/cha của 1 con với một tập các lớp cha?
- Ví dụ:
- CSDL đăng ký xe có người sở hữu là cá nhân, công ty hoặc ngân hàng

Ví dụ



Tóm tắt

- Tại sao ER phải mở rộng thành EER
- Chuyên biệt hóa và tổng quát hóa
- Cách thể hiện hệ thống phân cấp trên lược đồ EER
- Cách thể hiện các ràng buộc

Bài tập: Xây dựng lược đồ EER cho cơ sở dữ liệu ĐAOTAO (xác định các thực thể, quan hệ, ràng buộc) được mô tả như sau:

- Trường có nhiều khoa: Thông tin về Khoa gồm mã khoa, tên khoa, địa chỉ, số điện thoại.
- Mỗi Khoa quản lý nhiều môn học. Mỗi môn học gồm có tên môn học, mã số, số tín chỉ.
- Cán bộ của khoa có thể là Hành chính hoặc Giảng viên; Giảng viên có thể là Cơ hữu hoặc trợ giảng (trợ giảng là học viên cao học). Mỗi khoa có nhiều cán bộ làm việc, nhưng mỗi cán bộ chỉ làm việc cho một khoa. Mỗi một khoa có một chủ nhiệm khoa, đó là một Giảng viên.
- Mỗi giáo viên có thể dạy nhiều nhất là 4 môn học và cũng có thể không dạy môn học nào.
- Học viên có thể là sinh viên ĐH hoặc cao học, mỗi học viên phải học nhiều môn học.
- Mỗi một khoa có nhiều học viên, mỗi học viên chỉ thuộc về một khoa.
- Mỗi sinh viên đại học có một giáo viên hướng dẫn, một giáo viên có thể hướng dẫn nhiều sinh viên.

Tài liệu tham khảo

- ***Giáo trình CSDL***
 - Chương 2,
- ***Fundamentals of Database system***
 - Chapter 4