

ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC SÀI GÒN

KHOA CÔNG NGHỆ THÔNG TIN



CƠ SỞ DỮ LIỆU NÂNG CAO

BÁO CÁO

CHUYÊN ĐỀ 2

Thiết Kế Cơ Sở Dữ Liệu Với Mô Hình E-R Và Mô Hình E-R Mở Rộng

BVHD : Nguyễn Duy Hàm

MSSV : 3123410143

Họ Võ Tên : Hồ Đắc Khả

SĐT:0941712219

Mục Lục

Phần 1: Giới thiệu và Khái niệm cơ bản của Mô hình E-R

1.1. Tại sao cần mô hình E-R?

1.2. Các khái niệm cơ bản

- Thực thể và Tập thực thể
- Thuộc tính
- Khóa
- Thực thể yếu

1.3. Mối kết hợp và Tập mối kết hợp

- Độ của mối kết hợp
- Vai trò (Role Names)
- Tính liên kết (Cardinality)
- Hạn chế tham gia (Participation)

1.4. Ví dụ minh họa

Phần 2: Mô hình Thực thể – Kết hợp mở rộng (E-R mở rộng)

2.1. Lớp con là gì? Khi nào thì cần lớp con?

2.2. Định nghĩa các thuật ngữ

2.3. Cơ chế thừa kế thuộc tính/mối quan hệ

2.4. Lớp con do người dùng định nghĩa và do vị từ định nghĩa

2.5. Chuyên môn hóa do người dùng định nghĩa và do thuộc tính định nghĩa

2.6. Hai loại ràng buộc chính đối với chuyên biệt hóa và tổng quát hóa

2.7. Sự khác biệt giữa hệ thống cấp bậc và cấu trúc lưới chuyên biệt hóa

2.8. Sự khác biệt giữa chuyên biệt hóa và tổng quát hóa; lý do không hiển thị trong sơ đồ

¶ Phụ lục từ mới

- Part 1: Basic E-R Model
 - Part 2: Extended E-R Model
-

Phần 1: Giới thiệu và Khái niệm cơ bản của Mô hình E-R

1.1. Tại sao cần mô hình E-R?

- Là công cụ thiết kế cơ sở dữ liệu **ở mức quan niệm** (Conceptual Level).
- Giúp **biểu diễn trực quan yêu cầu dữ liệu** bằng sơ đồ thay vì ngôn ngữ kỹ thuật.
- Là **cầu nối** giữa:
- Yêu cầu thực tế (người dùng, doanh nghiệp)
- Mô hình cơ sở dữ liệu quan hệ (bảng, cột, khóa trong SQL).

1.2. Các khái niệm cơ bản

a) Thực thể (Entity) và Tập thực thể (Entity Set)

- **Thực thể**: một đối tượng cụ thể, nhận diện được .(VD: Sinh viên Nguyễn Văn A , Khách hàng Nguyễn Văn Tèo hay là Sản phẩm Guitar Bass Fender ,.v.v).
- **Tập thực thể**: tập hợp các thực thể cùng loại, có chung thuộc tính. (VD: tất cả sinh viên → SinhVien , tất cả các khách hàng -> Customers).

b) Thuộc tính (Attribute)

- **Thuộc tính đơn (Simple attribute)**: không thể chia nhỏ (VD: Ngày sinh , Số CCCD).
- **Thuộc tính phức hợp (Composite attribute)**: gồm nhiều thành phần (VD: Địa chỉ = Số nhà + Đường + Thành phố , Họ Tên = Họ + Chữ Lót + Tên).
- **Thuộc tính đa trị (Multivalued attribute)**: có thể có nhiều giá trị (VD: Số điện thoại ,Email , Bút danh , Tài khoản mạng xã hội,...).
- **Thuộc tính dẫn xuất (Derived attribute)**: được tính toán từ thuộc tính khác (VD: Tuổi từ Ngày sinh , Giá đơn hàng = Giá bán * Số lượng mua hàng , Thâm niên = Ngày hiện tại – Ngày bắt đầu,...).

c) Khóa (Key)

- **Khóa ứng viên (Candidate Key)**: tập thuộc tính có thể xác định duy nhất một thực thể.
- **Khóa chính (Primary Key)**: khóa ứng viên được chọn.

- **Khóa ngoại (Foreign Key)**: thuộc tính tham chiếu sang tập thực thể khác.

d) Thực thể yếu (Weak Entity)

- Không có khóa chính riêng, phụ thuộc vào **thực thể mạnh**.
 - Xác định nhờ **khóa của thực thể mạnh + thuộc tính phân biệt riêng**.
 - VD: ChiTietHoaDon phụ thuộc vào HoaDon , NguoiThan phụ thuộc vào NhanVien.
-

1.3. Mối kết hợp (Relationship) và Tập mối kết hợp (Relationship Set)

- **Mối kết hợp (Relationship)**: liên kết giữa các thực thể (VD: Sinh viên → Học → Môn học).
- **Tập mối kết hợp (Relationship Set)**: tập hợp các mối kết hợp cùng loại.

a) Độ tuổi của mối kết hợp (Degree)

- Nhị phân (Binary): giữa 2 thực thể.
- Ba ngôi (Ternary): giữa 3 thực thể.

b) Vai trò (Role Names)

- Khi cùng một tập thực thể tham gia nhiều vai trò khác nhau.
- VD: Giảng viên – dạy – Sinh viên (giảng viên = role “Teacher”) nếu không sử dụng role thì cả hai đều chỉ là thực thể Người , dễ gây hiểu nhầm.

c) Tính liên kết (Cardinality)

- **1:1** → Một thực thể A chỉ liên kết với tối đa một thực thể B và ngược lại.
Ví dụ : Một nhân viên có **một** thẻ nhân viên.
- **1:N** → Một thực thể A có thể liên kết với nhiều thực thể B , nhưng B chỉ có thể liên kết với thực thể A.
Ví dụ: Một khoa có **nhiều** giảng viên.
- **M:N** → Một thực thể A có thể liên kết với nhiều thực thể B và ngược lại.
Ví dụ : Sinh viên học **nhiều** môn, môn có **nhiều** sinh viên.

d) Hạn chế tham gia (Participation)

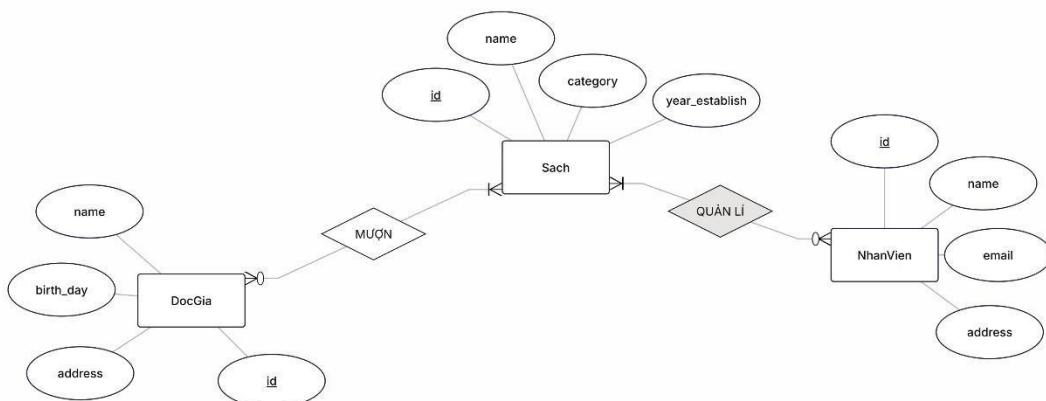
- **Toàn bộ (Total Participation)**: mọi thực thể đều phải tham gia (VD: Mọi hóa đơn đều có chi tiết hóa đơn ; Mọi sản phẩm điều có nhà cung cấp).

- **Một phần (Partial Participation)**: có thể có hoặc không (VD: Sinh viên có thể tham gia câu lạc bộ hoặc không).

1.4. Ví dụ minh họa.

Hệ thống quản lý thư viện:

- Tập thực thể: DocGia, Sach, NhanVien.
- Mối kết hợp: Muon (giữa DocGia và Sach), QuanLy (giữa NhanVien và Sach).
- Ràng buộc:
 - Một độc giả có thể mượn nhiều sách (1:N).
 - Một sách có thể được nhiều độc giả mượn (M:N).

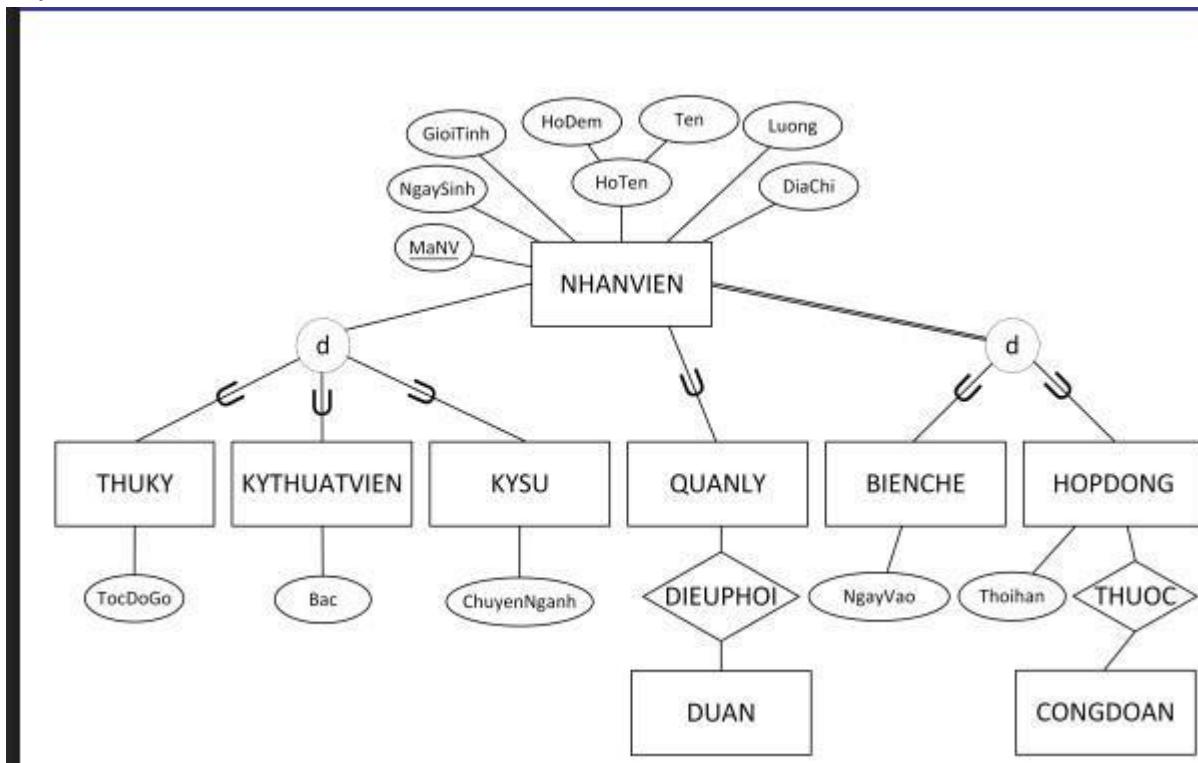


Phần 2: Mô hình Thực thể-Kết hợp mở rộng (E-R mở rộng)

1. Lớp con là gì? Khi nào thì cần lớp con trong mô hình hóa dữ liệu?

Lớp con (subclass) là một tập hợp con của **lớp cha (superclass)** chứa các thực thể có những đặc điểm chung và chia sẻ thuộc tính hoặc mối quan hệ chung, đồng thời có thêm các đặc tính riêng biệt. Nói cách khác nó kế thừa tất cả các thuộc tính của lớp cha và có thêm những thuộc tính cho riêng mình.

Ví dụ:



- THUKY là con của NHANVIEN kế thừa MaNV , NgaySinh ,... và có thêm thuộc tính TocDoGo.
- KYTHUATVIEN là con của NHANVIEN kế thừa MaNV , NgaySinh,... và có thêm thuộc tính Bac.

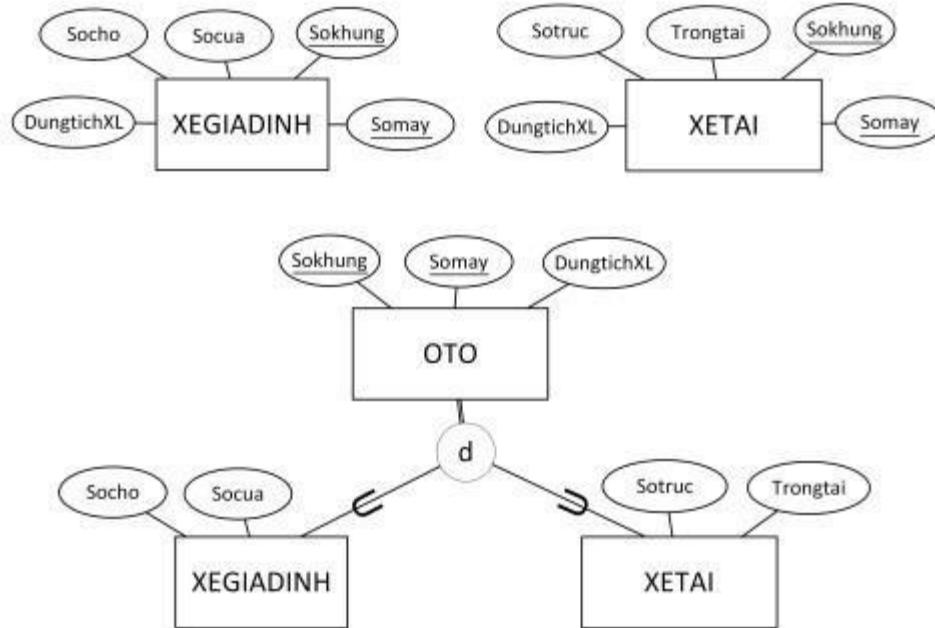
Khi cần dùng lớp con:

- Khi một số thực thể trong lớp cha sở hữu các **thuộc tính riêng** mà chỉ áp dụng cho chúng (ví dụ: *TocDoGo chỉ áp dụng riêng cho THUKY , Bac chỉ áp dụng cho KYTHUATVIEN,....*).

- Khi có **mối quan hệ riêng**: chỉ một số loại thực thể trong lớp cha tham gia vào một quan hệ đặc thù .
- Ví dụ : QUANLY -DIEU PHOI – DUAN -> Chỉ có lớp con QUANLY có quan hệ DIEUPHOI với DUAN.
- Ví dụ : HOPDONG – THUOC – CONGDOAN -> Chỉ có HOPDONG tham gia vào quan hệ THUOC với CONGDOAN.

2. Định nghĩa các thuật ngữ

- **Lớp cha của một lớp con (superclass)**: lớp tổng quát chứa tất cả thực thể chung mà lớp con là tập con của nó .
- **Mối quan hệ lớp cha/lớp con**: quan hệ kế thừa—lớp con kế thừa thuộc tính và mối quan hệ từ lớp cha .
- **Mối quan hệ IS-A (là một)** : diễn giải quan hệ giữa lớp con và lớp cha . Subclass thể hiện tính “IS-A” và được dùng khi chỉ một số thực thể trong superclass cần thuộc tính hoặc quan hệ riêng .
 - Ví dụ : THUKY IS-A NHANVIEN : có nghĩa là THUKY là một NHANVIEN.
- **Chuyên biệt hóa (specialization)**: quá trình chia lớp cha thành các lớp con cụ thể dựa trên đặc tính riêng .
 - Ví dụ : THUKY , KYSU , KYTHUATVIEN là một chuyên biệt hóa của lớp cha qua loại công việc
 - Ví dụ: BIENCHE , HOPDONG là một chuyên biệt mới của lớp NHANVIEN căn cứ vào cách trả lương.
- **Tổng quát hóa (generalization)**: quá trình ngược lại—gom nhiều lớp con lại thành một lớp cha chung dựa trên các thuộc tính chung.
 - Ví dụ:
Ta có thực thể XEGIADINH và XETAI có những thuộc tính chung .
Ta có thể gom chúng thành lớp:



- **Nhóm (category/union type)**: một dạng lớp con đặc biệt, là tập con của hợp nhất (union) nhiều loại lớp cha khác nhau.
 - Ví dụ: Có các lớp cha : CongTy và CaNhan. Nhóm ChuSoHuu -> chủ sở hữu có thể là CongTy hoặc CaNhan.
- **Thuộc tính riêng (specific/local attributes)**: thuộc tính chỉ áp dụng cho một lớp con nhất định, không có ở lớp cha .
- **Các mối quan hệ riêng (specific/local relationships)**: chỉ áp dụng cho lớp con, không dùng chung với lớp cha .

3. Cơ chế thừa kế thuộc tính/mối quan hệ. Tại sao nó lại hữu ích?

- **Cơ chế thừa kế**: Lớp con tự động kế thừa tất cả **thuộc tính** và **mối quan hệ** của lớp cha, đồng thời có thể bổ sung thêm **thuộc tính riêng** hoặc **quan hệ riêng** .
- **Tại sao hữu ích**:
 - **Không lặp lại**: tránh mô hình hóa lại thuộc tính/mối quan hệ đã có ở lớp cha.
 - **Tổ chức rõ ràng**: cấu trúc dữ liệu trở nên dễ hiểu hơn, phân chia rõ ràng giữa chung và riêng.
 - **Dễ mở rộng và bảo trì**: khi cần thêm loại con, chỉ cần tạo lớp con mới, vẫn kế thừa mọi thứ từ lớp cha.

4. Lớp con do người dùng định nghĩa và do vị từ định nghĩa

- **User-defined subclass (Lớp con do người dùng định nghĩa):** được định nghĩa thủ công bởi người dùng hoặc nhà thiết kế—liệt kê những thực thể thuộc lớp con mà không dựa vào điều kiện nào nhất định.
 - Ví dụ: Tập thực thể NhanVien

Người thiết kế tự tạo lớp con: - NhanVienDuAn (nhân viên tham gia dự án lớn)
 - Ví dụ: Lớp con là NhanVienNam -> Xác định lớp con theo điều kiện GioiTinh='Nam'
 - Ví dụ : NHANVIEN có thuộc tính LoaiHopDong

Lớp con NhanVienBienChe ->Xác định lớp con theo điều kiện LoaiHopDong = 'Biên chế'.
- **Predicate-defined subclass(Lớp con do vị từ định nghĩa):** xác định lớp con dựa trên **điều kiện (predicate)** trên thuộc tính của lớp cha.
 - Ví dụ: Tập thực thể NhanVien có thuộc tính GioiTinh

Khác biệt chính:

- User-defined: xác định thủ công, không có điều kiện hay thuộc tính.
- Predicate-defined: dựa trên điều kiện logic gắn với thuộc tính, tự động phân loại thực thể.

5. Chuyên môn hóa do người dùng định nghĩa và do thuộc tính định nghĩa

- **User-defined specialization (Chuyên môn hóa do người dùng định nghĩa):** phân loại từ lớp cha thành các lớp con một cách tự do, không dựa vào giá trị thuộc tính cụ thể .
 - Ví dụ : Lớp cha NhanVien

Người dùng tự định nghĩa các lớp con : QuanLy (Nhân viên làm quản lý)
- **Attribute-defined specialization (Chuyên môn hóa do thuộc tính định nghĩa):** chia lớp cha thành các lớp con dựa trực tiếp vào giá trị của một thuộc tính , tự động phân loại dựa trên dữ liệu.

- Ví dụ: Lớp cha NhanVien có thuộc tính ChucVu
Dự vào chức vụ , phân loại : ThuKy -> ChucVu ="Thư ký".

Khác biệt:

- User-defined: không dùng thuộc tính làm cơ sở.
- Attribute-defined: sử dụng một thuộc tính xác định rõ ràng để phân loại.

6. Hai loại ràng buộc chính đối với chuyên biệt hóa và tổng quát hóa

1. Ràng buộc rời rạc (Disjointness Constraint):

- **Disjoint**: mỗi thực thể chỉ thuộc tối đa một lớp con.
- **Overlapping**: thực thể có thể thuộc nhiều lớp con cùng lúc .

2. Ràng buộc đầy đủ (Completeness / Totalness Constraint):

- **Total specialization**: tất cả thực thể trong lớp cha phải thuộc ít nhất một lớp con (kết nối bằng **đường đôi**).
- **Partial specialization**: một số thực thể có thể không thuộc lớp con nào (kết nối bằng **đường đơn**) .

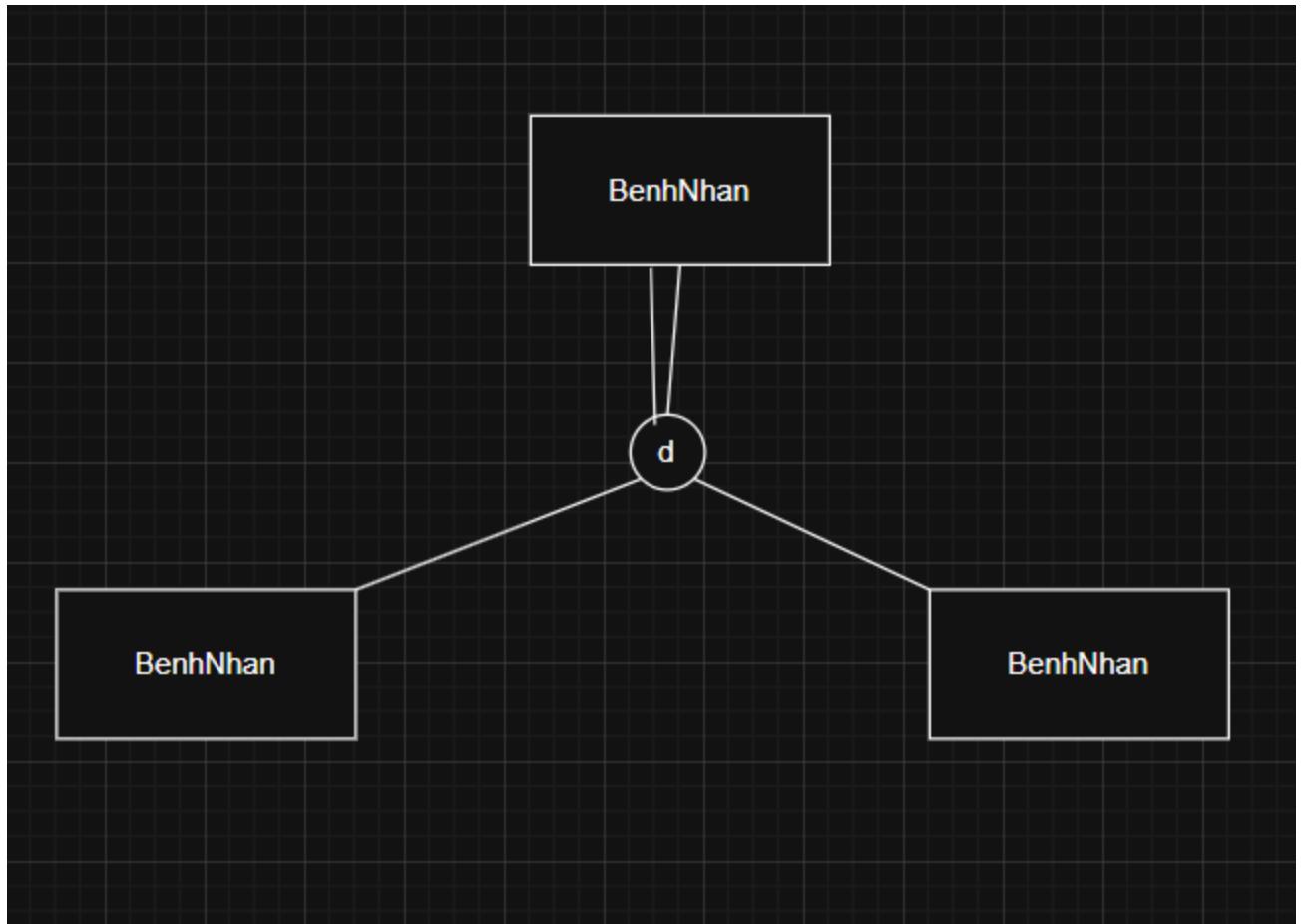
Và hai loại ràng buộc này độc lập với nhau.

Ví dụ : về Ràng buộc Rời rạc và Toàn bộ (Disjoint and Total)

- **Quy tắc**: Một thực thể của lớp cha **phải** thuộc về **đúng một** trong các lớp con.
- **Ký hiệu trên sơ đồ**: Kết nối bằng **đường đôi** và có chữ 'd' (disjoint) trong vòng tròn.

Tình huống: Tại một bệnh viện, một bệnh nhân (BENHNHAN) khi nhập viện phải được phân loại ngay lập tức là **Bệnh nhân nội trú** (NOITRU) hoặc **Bệnh nhân ngoại trú** (NGOAITRU).

- **Lớp cha**: BENHNHAN
- **Các lớp con**: NOITRU, NGOAITRU

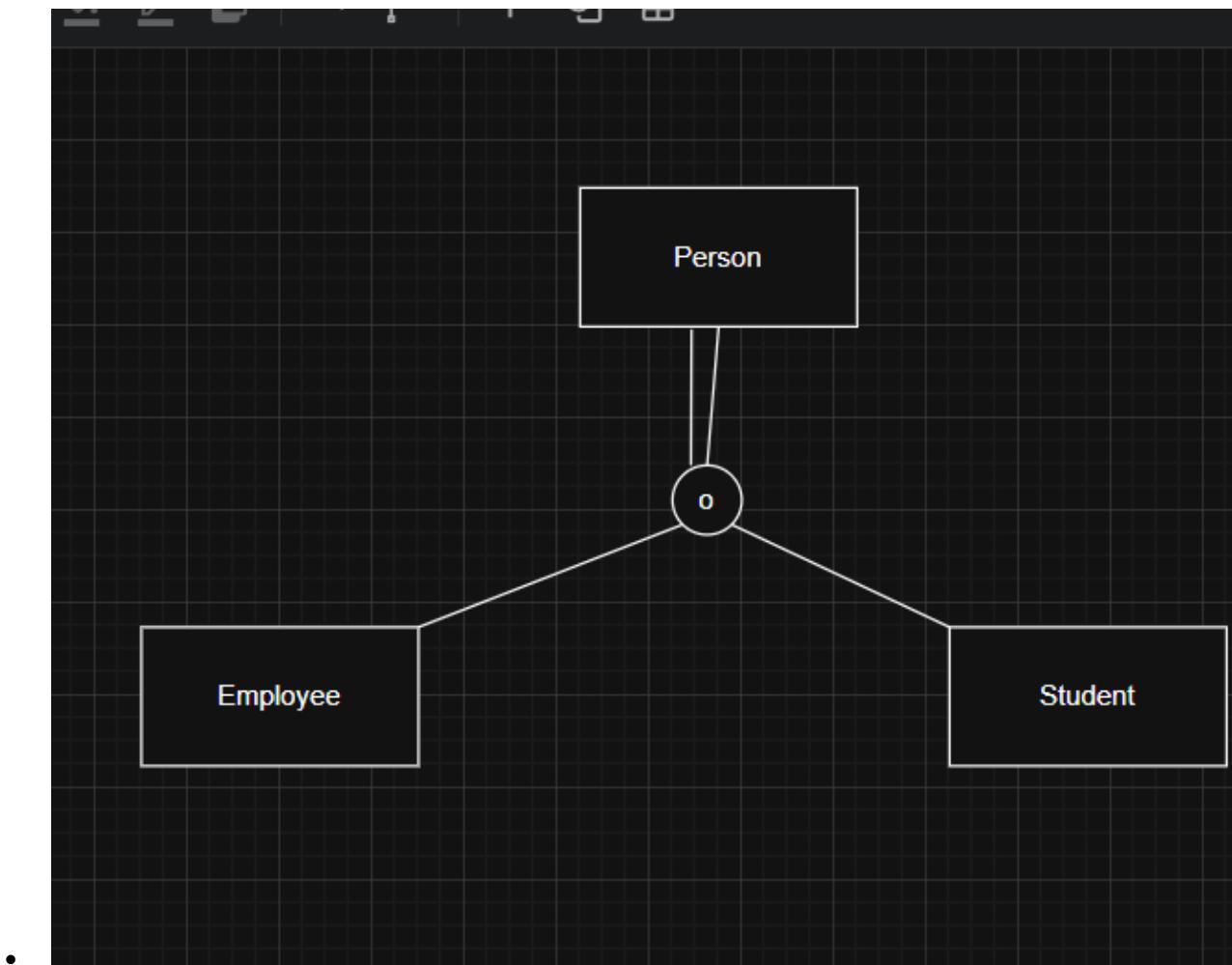


Ví dụ về Ràng buộc Giao nhau và Toàn bộ (Overlapping and Total)

- **Quy tắc:** Một thực thể của lớp cha **phải** thuộc về **ít nhất một** lớp con, và **có thể thuộc nhiều** lớp con cùng lúc.
- **Ký hiệu trên sơ đồ:** Kết nối bằng **đường đôi** và có chữ 'o' (overlapping) trong vòng tròn.

Tình huống: Trong cơ sở dữ liệu của một trường đại học, một người (PERSON) trong trường phải là **Nhân viên** (EMPLOYEE) hoặc là **Sinh viên** (STUDENT). Một số người có thể là cả hai. (Đây chính là ví dụ trong sơ đồ trang 12 của bạn).

- **Lớp cha:** PERSON
- **Các lớp con:** EMPLOYEE, STUDENT

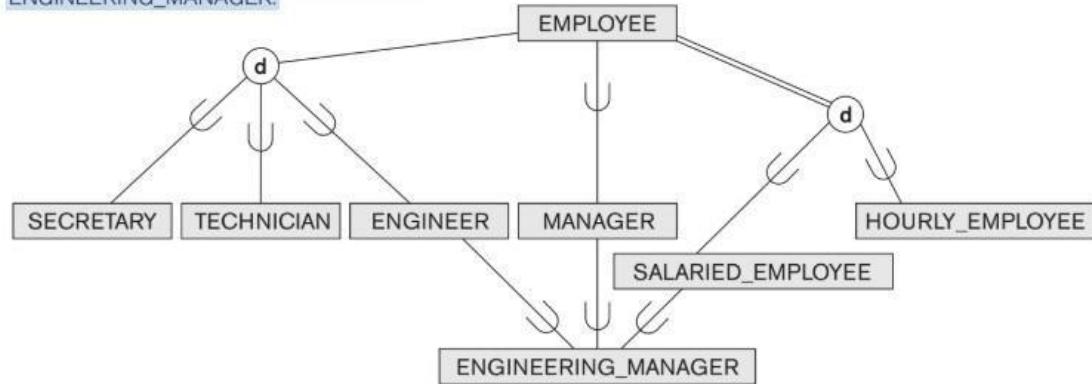


7. Sự khác biệt giữa hệ thống cấp bậc chuyên biệt hóa và cấu trúc lưới chuyên biệt hóa

- **Specialization hierarchy:** cấu trúc cây—mỗi lớp con chỉ có **một lớp cha duy nhất** (thừa kế đơn).

Figure 4.6

A specialization lattice with shared subclass
ENGINEERING_MANAGER.

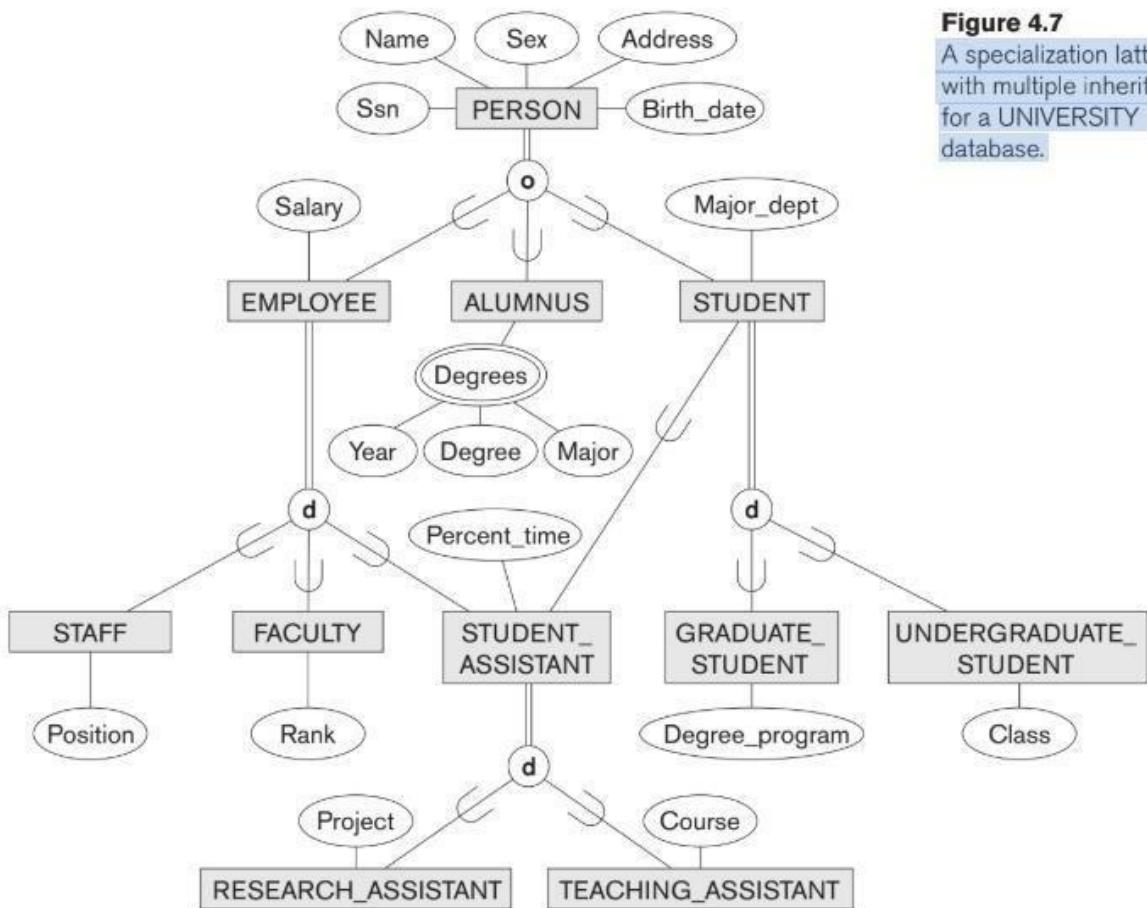


Các lớp con chỉ kế thừa một lớp cha duy nhất.

- **Specialization lattice:** cấu trúc lưới—cho phép lớp con có **nhiều lớp cha trực tiếp** (thừa kế đa).

Figure 4.7

A specialization lattice with multiple inheritance for a UNIVERSITY database.



Lớp con STUDENT-ASSISTANT vừa kế thừa EMPLOYEE và STUDENT.

8. Sự khác biệt giữa chuyên biệt hóa và tổng quát hóa; tại sao không hiển thị trong sơ đồ?

- **Chuyên môn hóa (specialization)**: quy trình đi từ lớp cha → tạo ra các lớp con cụ thể hơn.
- **Tổng quát hóa (generalization)**: quy trình ngược lại—từ các lớp con → tổng hợp thành lớp cha chung.

Cả hai quá trình đều kết thúc với cùng cấu trúc lớp cha–lớp con giống nhau. Vì vậy, trong các sơ đồ EER, **không cần thể hiện rõ hướng chuyển đổi (từ trên xuống hay từ dưới lên)** — biểu diễn là cùng một định nghĩa cấu trúc, cách thể hiện giống nhau.

=====Phụ lục từ mới=====

Part 1: Basic E-R Model

1. **Conceptual Level**: Mức quan niệm
2. **Simple Attribute**: Thuộc tính đơn
3. **Composite Attribute**: Thuộc tính phức hợp
4. **Multivalued Attribute**: Thuộc tính đa trị
5. **Derived Attribute**: Thuộc tính dẫn xuất
6. **Ternary Relationship**: Mối kết hợp ba ngôi (between 3 entities)
7. **Cardinality**: Tính liên kết (or Cardinality Ratio)
8. **Participation Constraint**: Hạn chế tham gia
9. **Total Participation**: Toàn bộ (tham gia)
10. **Partial Participation**: Một phần (tham gia)

Part 2: Extended E-R Model (EER Model)

1. **IS-A Relationship**: Mỗi quan hệ IS-A (là một)
2. **Specialization**: Chuyên biệt hóa
3. **Generalization**: Tổng quát hóa

4. **Specific/Local Attributes:** Thuộc tính riêng
5. **Specific/Local Relationships:** Các mối quan hệ riêng
6. **User-defined subclass:** Lớp con do người dùng định nghĩa
7. **Predicate-defined subclass:** Lớp con do vị từ định nghĩa
8. **User-defined specialization:** Chuyên môn hóa do người dùng định nghĩa
9. **Attribute-defined specialization:** Chuyên môn hóa do thuộc tính định nghĩa
10. **Disjointness Constraint:** Ràng buộc rời rạc
11. **Disjoint:** Rời rạc
12. **Overlapping:** Giao nhau (có thể thuộc nhiều)
13. **Completeness / Totalness Constraint:** Ràng buộc đầy đủ
14. **Total Specialization:** Chuyên biệt hóa toàn bộ
15. **Partial Specialization:** Chuyên biệt hóa một phần
16. **Specialization Hierarchy:** Hệ thống cấp bậc chuyên biệt hóa
17. **Specialization Lattice:** Cấu trúc lưới chuyên biệt hóa

=====END=====