**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 them những thuộc tính cho riêng mình.

**Ví dụ: A diagram of a company

AI-generated content may be incorrect.**

**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 , 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**: 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 ; 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:

A diagram of a company

AI-generated content may be incorrect.

* **Nhóm (category/union type)**: một dạng lớp con đặc biệt, là tập con của hợp nhất (union) nhiều loại lớp cha khác nhau .
* **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**: 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.
* **Predicate-defined subclass**: xác định lớp con dựa trên **điều kiện (predicate)** trên thuộc tính của lớp cha. Ví dụ: NHANVIEN có thuộc tính GioiTinh có lớp con là NHANVIENNAM -> Xác định lớp con theo giới tính của NHANVIEN.

**Khác biệt chính**:

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

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

* **User-defined specialization**: phân loại từ lớp cha thành các lớp con một cách tự do, không dựa vào giá trị thuộc tính cụ thể .
* **Attribute-defined specialization**: dựa trên **giá trị của một thuộc tính** làm tiêu chí phân loại. Ví dụ: thuộc tính CHUCVU trong NHANVIEN phân loại thành các lớp con THUKY , KYSU , KYTHUATVIEN.

**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**.

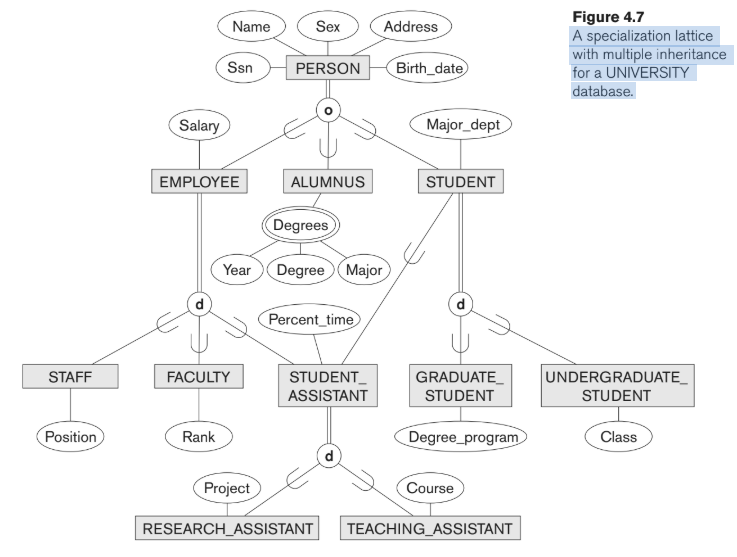
**7. Sự khác biệt giữa hệ thống cấp bậc chuyên biệt hóa và cấu trúc lưới chuyên biệt hóa**

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

A diagram of a company

AI-generated content may be incorrect.

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



**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.