

Mô hình dữ liệu thực thể – kết hợp (Entity – Relationship Model)

Cơ sở dữ liệu cơ bản – COS212

Lê Thành Văn

Khoa Hệ thống Thông tin Quản lý
HUTECH

17.11.2025

1. Quá trình thiết kế Cơ sở dữ liệu
2. Định nghĩa
3. Thành phần của mô hình
4. Cách biểu diễn sơ đồ
5. Mô hình thực thể – kết hợp mở rộng (Enhanced ER Model)

Nhắc lại

Mô hình dữ liệu là **sự trừu tượng hóa môi trường thực**, nó là sự biểu diễn dữ liệu ở mức quan niệm.

Quá trình thiết kế Cơ sở dữ liệu

Các kỹ thuật phát triển hệ thống

Các kỹ thuật phát triển hệ thống phổ biến:

- Mô hình hóa chức năng (Functional Modeling)
- Mô hình hóa dữ liệu (Data Modeling)
- Tham chiếu chéo (Cross-referencing)

Mô hình hóa chức năng

Mô hình hóa chức năng tập trung vào các quy trình xử lý dữ liệu trong hệ thống, bao gồm:

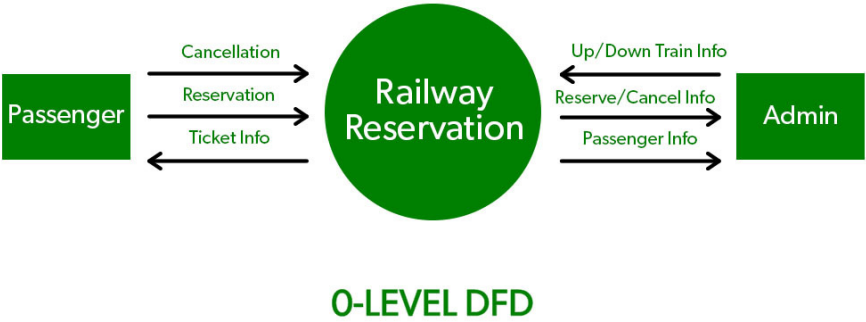
- Biểu đồ luồng dữ liệu (Data Flow Diagram, DFD)
- Biểu đồ cấu trúc (Structure Chart)

Mô hình hóa dữ liệu

Cố gắng để tạo một biểu diễn của thế giới thực:

- Bỏ qua một ít sự phức tạp của thế giới thực.
- Sự đơn giản dựa vào một tập nhỏ các ký hiệu.

Cố gắng để rút gọn tổ chức dữ liệu thành sự mô tả của các thực thể và các mối liên hệ giữa chúng.



Ví dụ về biểu đồ luồng dữ liệu mức 0

Tiến trình mô hình hóa độc lập với nền tảng phát triển (hay phần mềm)
Mô hình được sử dụng để trao đổi giữa người thiết kế CSDL và người dùng trong giai đoạn phân tích thiết kế

Tham chiếu chéo nhằm đảm bảo tính nhất quán giữa mô hình hóa chức năng và mô hình hóa dữ liệu.

- Thiết kế CSDL bao gồm ba mô hình chính:
- Mô hình quan niệm (Conceptual Model): mô tả các yêu cầu về dữ liệu của người dùng cuối.
 - Mô hình lô-gíc (Logical Model): mô tả cấu trúc dữ liệu theo cách mà hệ quản trị CSDL có thể hiểu được.
 - Mô hình vật lý (Physical Model): mô tả cách dữ liệu được lưu trữ trên thiết bị lưu trữ vật lý.

Lưu ý: Các mô hình này khác với kiến trúc ba mức của CSDL (mức ngoài, mức quan niệm, mức trong).

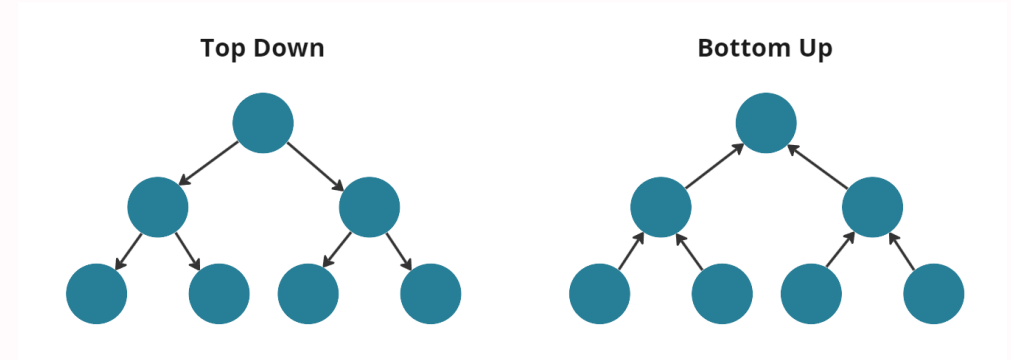
- Quá trình thiết kế CSDL bao gồm các bước chính sau:
- ➊ Thu thập và phân tích nhu cầu (yêu cầu về CSDL, yêu cầu chức năng, ...)
 - ➋ Thiết kế quan niệm và phân tích chức năng.
 - ➌ Thiết kế mô hình dữ liệu mức lô-gíc.
 - ➍ Thiết kế mô hình dữ liệu mức vật lý.
 - ➎ Triển khai và bảo trì CSDL.

Hướng tiếp cận thiết kế CSDL

Hai hướng tiếp cận chính trong thiết kế CSDL:

- Thiết kế từ trên xuống (Top-down Design).
- Thiết kế từ dưới lên (Bottom-up Design).

Hướng tiếp cận thiết kế CSDL (tiếp)



Hướng tiếp cận thiết kế

Thiết kế từ trên xuống

- Bắt đầu từ việc xác định các yêu cầu tổng thể của hệ thống.
- Phân tích các yêu cầu này để xác định các thực thể chính, mối quan hệ và thuộc tính.
- Tạo mô hình dữ liệu quan niệm dựa trên các thực thể và mối quan hệ đã xác định.
- Chuyển đổi mô hình dữ liệu quan niệm sang mô hình lô-gíc và sau đó sang mô hình vật lý.

Thiết kế từ trên xuống (tiếp)

Ví dụ: Thiết kế CSDL quản lý thư viện:

- Xác định các yêu cầu tổng thể: quản lý sách, quản lý độc giả, quản lý mượn trả sách.
- Phân tích các yêu cầu để xác định các thực thể chính: SÁCH, ĐỘC_GIẢ, MƯỢN_TRẢ.
- Tạo mô hình dữ liệu quan niệm dựa trên các thực thể và mối quan hệ đã xác định.
- Chuyển đổi mô hình dữ liệu quan niệm sang mô hình lô-gíc và sau đó sang mô hình vật lý.

Thiết kế từ dưới lên

- Bắt đầu từ việc thu thập dữ liệu hiện có và các yêu cầu chi tiết từ người dùng.
- Xác định các thực thể, mối quan hệ và thuộc tính dựa trên dữ liệu thu thập được.
- Tạo mô hình dữ liệu quan niệm từ các thực thể và mối quan hệ đã xác định.
- Chuyển đổi mô hình dữ liệu quan niệm sang mô hình lô-gíc và sau đó sang mô hình vật lý.

Thiết kế từ dưới lên (tiếp)

Ví dụ: Thiết kế CSDL quản lý bán hàng:

- Thu thập dữ liệu hiện có: thông tin sản phẩm, thông tin khách hàng, thông tin đơn hàng.
- Xác định các thực thể: SẢN_PHẨM, KHÁCH_HÀNG, ĐƠN_HÀNG.
- Tạo mô hình dữ liệu quan niệm từ các thực thể và mối quan hệ đã xác định.
- Chuyển đổi mô hình dữ liệu quan niệm sang mô hình lô-gíc và sau đó sang mô hình vật lý.

Lưu ý: Trong thực tế, quá trình thiết kế CSDL thường kết hợp cả hai hướng tiếp cận từ trên xuống và từ dưới lên để đảm bảo tính toàn diện và hiệu quả của mô hình dữ liệu.

Định nghĩa

Mô hình thực thể – kết hợp

Mô hình thực thể – kết hợp (Entity – Relationship Model, ER Model) là một mô hình dữ liệu mức lô-gíc nhằm mô tả các đối tượng trong thế giới thực và quan hệ giữa chúng.

Thực thể – Entity

Định nghĩa

Thực thể là một đối tượng trong thế giới thực, có sự tồn tại độc lập:

- *Thực thể cụ thể*: có thể cảm nhận bằng giác quan (xe đạp, bàn, ghế, ...).
- *Thực thể trừu tượng*: có thể nhận biết bằng nhận thức (khóa học, buổi học, ...).

Mối kết hợp – Relationship

Định nghĩa

Mối kết hợp là sự liên kết giữa hai hay nhiều thực thể với nhau, dùng để mô tả mối liên quan, quan hệ của các thực thể.

Thuộc tính – Attribute

Định nghĩa

Thuộc tính là đặc điểm, tính chất của thực thể hay mối kết hợp.

Bảng thuật ngữ

Thuật ngữ		Ý nghĩa
Anh	Việt	
Entity	Thực thể	Một đối tượng trong thế giới thực
Relationship	Mối kết hợp	Sự liên kết giữa hai hay nhiều thực thể với nhau
Attribute	Thuộc tính	Đặc điểm, tính chất của thực thể hay mối kết hợp

Bảng chữ viết tắt

Chữ viết tắt	Ý nghĩa
ER Model (Entity – Relationship Model)	Mô hình thực thể – kết hợp
ERD (Entity – Relationship Diagram)	Sơ đồ thực thể – kết hợp
EER Model (Enhanced Entity – Relationship Model)	Mô hình thực thể – kết hợp mở rộng
EERD (Enhanced Entity – Relationship Diagram)	Sơ đồ thực thể – kết hợp mở rộng

Thành phần của mô hình

Thực thể

Kiểu thực thể – Entity Type

Kiểu thực thể để chỉ các thực thể có cùng đặc điểm, tính chất. *Ví dụ:* Kiểu thực thể **SINH_VIÊN** để chỉ tất cả cá nhân (trong trường đại học) có các đặc điểm chung như mã số sinh viên, họ tên, ngày sinh, địa chỉ, ...
Ta sẽ ghi tên kiểu thực thể bằng chữ in hoa và gạch dưới thay cho khoảng trắng. *Ví dụ:* SINH_VIÊN, GIẢNG_VIÊN, MÔN_HOC, ...

Tập thực thể – Entity Set

Tập thực thể là tập hợp các thực thể cùng kiểu. *Ví dụ:* Tập thực thể **SINH_VIÊN** bao gồm tất cả các cá nhân có kiểu **SINH_VIÊN** như: SV₀₀₁, SV₀₀₂, SV₀₀₃, ...
Ký hiệu: **SINH_VIÊN** = {SV₀₀₁, SV₀₀₂, SV₀₀₃, ...}

Nếu điều kiện cho phép, ta sẽ dùng ký hiệu **E** để chỉ tập thực thể có kiểu thực thể **E**.

Mối kết hợp

Kiểu mối kết hợp – Relationship Type

Kiểu mối kết hợp để chỉ các mối kết hợp có cùng đặc điểm, tính chất. *Ví dụ:* Kiểu mối kết hợp **ĐĂNG_KÝ** để chỉ tất cả các mối kết hợp giữa thực thể kiểu **SINH_VIÊN** và thực thể kiểu **MÔN_HOC**.
Tên của kiểu mối kết hợp cũng được ghi bằng chữ in hoa và gạch dưới thay cho khoảng trắng. *Ví dụ:* ĐĂNG_KÝ, GIẢNG_DẠY, ...

Một cách hình thức, ta ký hiệu kiểu mối kết hợp **R** giữa các kiểu thực thể **E₁**, **E₂**, ..., **E_n** là:

$$\mathbf{R} \subseteq \mathbf{E}_1 \times \mathbf{E}_2 \times \cdots \times \mathbf{E}_n$$

Tập mối kết hợp – Relationship Set

Tập mối kết hợp là tập hợp các mối kết hợp cùng kiểu. *Ví dụ:* Tập mối kết hợp **ĐĂNG_KÝ** bao gồm tất cả các mối kết hợp giữa các thực thể kiểu **SINH_VIÊN** và thực thể kiểu **MÔN_HỌC**, như: (SV₀₀₁, CSDL), (SV₀₀₂, TOÁN), (SV₀₀₃, LẬP TRÌNH), ...

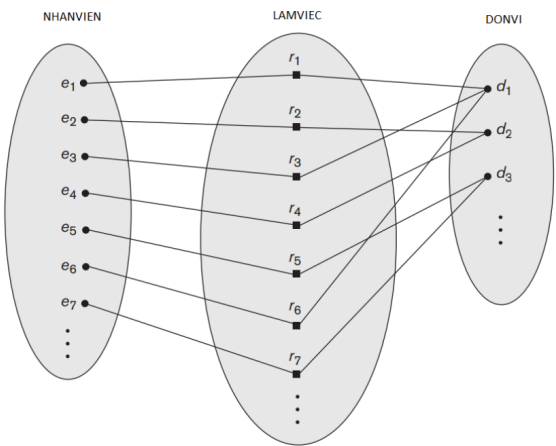
Định nghĩa hình thức Tập mối kết hợp

Cho kiểu mối kết hợp $R \subseteq E_1 \times E_2 \times \dots \times E_n$, tập mối kết hợp \mathbb{R} của kiểu mối kết hợp R được định nghĩa như sau:

$$\mathbb{R} = \{r = (e_1, e_2, \dots, e_n) \mid e_1 \in \mathbb{E}_1, e_2 \in \mathbb{E}_2, \dots, e_n \in \mathbb{E}_n\}$$

Trong đó:

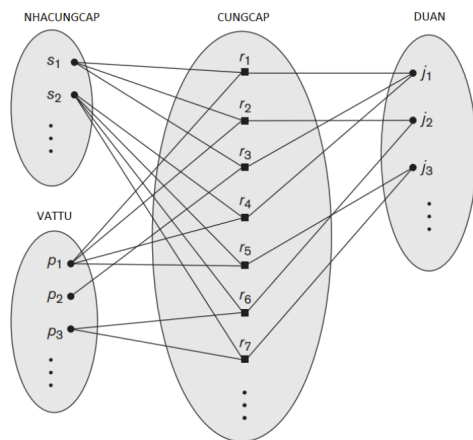
- e_i là thực thể thuộc tập thực thể \mathbb{E}_i với $1 \leq i \leq n$.
- r là thể hiện kết hợp thuộc tập mối kết hợp \mathbb{R} .



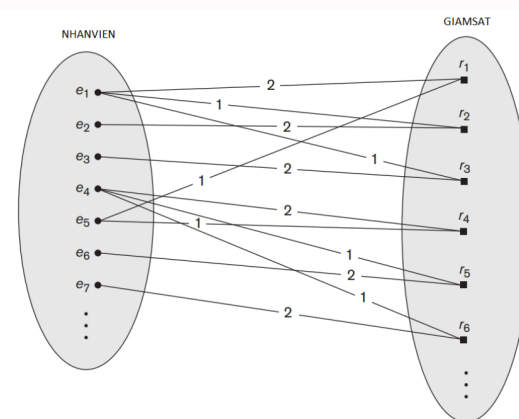
Mối kết hợp và tập mối kết hợp

Cấp kết hợp, tên vai trò và kiểu kết hợp đệ quy

- **Cấp kết hợp** (Degree of Relationship): số lượng các kiểu thực thể tham gia vào kiểu mối kết hợp.
- **Tên vai trò** (Role Name): tên của vai trò mà một thực thể đóng trong một mối kết hợp. Trong nhiều trường hợp, các vai trò là rõ ràng và không cần đặt tên.
- **Kiểu kết hợp đệ quy** (Recursive Relationship Type): kiểu mối kết hợp mà trong đó một kiểu thực thể tham gia vào một kiểu liên kết với **nhiều vai trò** khác nhau, tên vai trò là cần thiết để phân biệt ý nghĩa của việc tham gia.



Mối kết hợp cấp 3



Mối kết hợp đệ quy

Thuộc tính

Nhắc lại: Thuộc tính là đặc điểm, tính chất của thực thể hay mối kết hợp. *Ví dụ:* Thuộc tính của thực thể kiểu **SINH_VIÊN** bao gồm mã số sinh viên, họ tên, ngày sinh, địa chỉ, ...

Ta sẽ ghi tên thuộc tính bằng chữ in thường và gạch dưới thay cho khoảng trắng. *Ví dụ:* mã_số_sinh_viên, họ_tên, ngày_sinh, địa_chỉ, ... Ngoài ra, ta ký kiểu thuộc tính **attr** của kiểu thực thể **E** là **E.attr** hoặc **E[attr]**.

Phân loại thuộc tính

- Thuộc tính được phân loại như sau:
- **Thuộc tính đơn** (Simple Attribute) là thuộc tính không thể phân chia ra được thành các thành phần nhỏ hơn.
 - **Thuộc tính phức** (Composite Attribute) là thuộc tính có thể phân chia ra được thành các thành phần nhỏ hơn với các ý nghĩa độc lập.

Phân loại thuộc tính (tiếp)

- **Thuộc tính đơn trị** (Single-valued Attribute) là thuộc tính chỉ có một giá trị duy nhất đối với mỗi thực thể hay mỗi kết hợp.
- **Thuộc tính đa trị** (Multi-valued Attribute) là thuộc tính có thể có nhiều giá trị đối với mỗi thực thể hay mỗi kết hợp.
- **Thuộc tính suy diễn** (Derived Attribute) là thuộc tính có giá trị được suy ra từ các thuộc tính khác.

Thuộc tính khóa

Định nghĩa

Thuộc tính khóa (Key Attribute) là thuộc tính (hoặc tập hợp các thuộc tính) có khả năng xác định **duy nhất** một thực thể trong tập thực thể.

Ví dụ: Trong tập thực thể **SINH_VIÊN**, thuộc tính **mã_số_sinh_viên** có thể được sử dụng làm thuộc tính khóa vì nó xác định duy nhất mỗi sinh viên.

Thuộc tính khóa (tiếp)

- **Khóa đơn** (Simple Key) là thuộc tính khóa bao gồm một thuộc tính duy nhất.
- **Khóa phức** (Composite Key) là thuộc tính khóa bao gồm nhiều thuộc tính. Tập hợp các thuộc tính này cùng nhau xác định duy nhất một thực thể. Số lượng thuộc tính trong khóa phức phải là nhỏ nhất có thể.
- **Kiểu thực thể yếu** (Weak Entity Type) là kiểu thực thể không có thuộc tính khóa riêng.

Thuộc tính khóa (tiếp)

- Một kiểu thực thể có thể có một hoặc nhiều thuộc tính khóa.
- Ví dụ:* Trong cơ sở dữ liệu quản lý trường đại học:
- Kiểu thực thể **MÔN_HOC** có một khóa đơn là mã_môn_học.
 - Kiểu thực thể **LỚP_HOC** có một khóa phức là (mã_môn_học, mã_lớp).
 - Kiểu thực thể **GIẢNG_VIÊN** có nhiều khóa đơn như mã_giảng_viên, email_giảng_viên.
 - Kiểu thực thể **PHÒNG_HOC** là kiểu thực thể yếu, không có **thuộc tính khóa riêng** (nhưng vẫn có thuộc tính khóa).

Kiểu thực thể yếu và mối kết hợp nhận diện

Định nghĩa

Kiểu thực thể yếu (Weak Entity Type) là kiểu thực thể không có thuộc tính khóa riêng, nó phụ thuộc vào một kiểu thực thể khác (kiểu thực thể mạnh) để xác định duy nhất các thực thể của nó.

Định nghĩa

Mối kết hợp nhận diện (Identifying Relationship) là mối kết hợp giữa kiểu thực thể yếu và kiểu thực thể mạnh, dùng để xác định duy nhất các thực thể của kiểu thực thể yếu.

Kiểu thực thể yếu và mối kết hợp nhận diện (tiếp)

- Ví dụ:* Trong cơ sở dữ liệu quản lý trường đại học, ta có kiểu thực thể **PHÒNG_HOC** là kiểu thực thể yếu:
- Khóa của **PHÒNG_HOC** là (mã_tòa_nhà, số_phòng).
 - mã_tòa_nhà là khóa của kiểu thực thể mạnh **TÒA_NHÀ**.
 - Mối kết hợp nhận diện là **THUỘC**, liên kết giữa **PHÒNG_HOC** và **TÒA_NHÀ**.

Ràng buộc của kiểu mối kết hợp

Ràng buộc của kiểu mỗi kết hợp

Các kiểu mỗi kết hợp thường có một số ràng buộc để chỉ ra số tổ hợp có thể của các thực thể tham gia trong tập hợp các thể hiện liên kết. Có hai loại ràng buộc chính:

- **Tỷ số lực lượng** (Cardinality Ratio).
- **Ràng buộc tham gia** (Participation Constraint).

Tỷ số lực lượng

Định nghĩa

Tỷ số lực lượng (Cardinality Ratio) của một kiểu mỗi kết hợp chỉ ra số lượng tối đa các thể hiện liên kết mà một thể hiện thực thể có thể tham gia trong kiểu mỗi kết hợp đó.

Đối với kiểu mỗi kết hợp giữa hai kiểu thực thể (kiểu mỗi kết hợp cấp hai), có ba loại tỷ số lực lượng phổ biến:

- Một – Một (1 : 1)
- Một – Nhiều (1 : N)
- Nhiều – Nhiều (M : N)

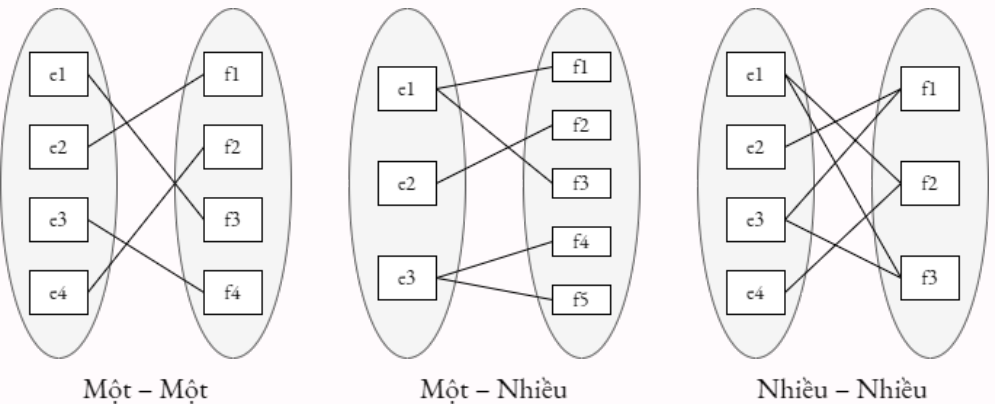
Tỷ số lực lượng (tiếp)

Định nghĩa

Cho kiểu mỗi kết hợp $R \subseteq E \times F$, ta nói kiểu mỗi kết hợp R có:

- Tỷ số lực lượng là 1 : 1 nếu với mỗi thực thể $e \in E$, e kết hợp với **nhieu nhất một** thực thể $f \in F$ và ngược lại.
- Tỷ số lực lượng là 1 : N nếu với mỗi thực thể $e \in E$, e có thể kết hợp với **nhieu hơn một** thực thể $f \in F$. Nhưng với mỗi thực thể $f \in F$, f chỉ kết hợp với **nhieu nhất một** thực thể $e \in E$.
- Tỷ số lực lượng là M : N nếu với mỗi thực thể $e \in E$, e có thể kết hợp với **nhieu hơn một** thực thể $f \in F$ và ngược lại.

Tỷ số lực lượng (tiếp)



Các loại tỷ số lực lượng

Tỷ số lực lượng (tiếp)

Trong cơ sở dữ liệu quản lý trường đại học:

- Kiểu mỗi kết hợp **QUẢN_LÝ** giữa kiểu thực thể **GIẢNG_VIÊN** và kiểu thực thể **KHOA** có tỷ số lực lượng là 1 : 1.
- Kiểu mỗi kết hợp **GIẢNG_DẠY** giữa kiểu thực thể **GIẢNG_VIÊN** và kiểu thực thể **MÔN_HOC** có tỷ số lực lượng là 1 : N.
- Kiểu mỗi kết hợp **ĐĂNG_KÝ** giữa kiểu thực thể **SINH_VIÊN** và kiểu thực thể **MÔN_HOC** có tỷ số lực lượng là M : N.

Ràng buộc tham gia

Định nghĩa

Ràng buộc tham gia (Participation Constraint) chỉ ra các thực thể của một kiểu thực thể có bắt buộc phải tham gia vào ít nhất một thể hiện kết hợp trong kiểu mỗi kết hợp hay không.

Có hai loại ràng buộc tham gia chính:

- Tham gia toàn phần (Total Participation).
- Tham gia một phần (Partial Participation).

Thuộc tính của mỗi kết hợp

Định nghĩa

Thuộc tính của mỗi kết hợp (Attributes of Relationship) là các thuộc tính gắn liền với một kiểu mỗi kết hợp, dùng để mô tả đặc điểm, tính chất của mỗi kết hợp đó.

Ví dụ: Trong cơ sở dữ liệu quản lý trường đại học, kiểu mỗi kết hợp **ĐĂNG_KÝ** giữa kiểu thực thể **SINH_VIÊN** và kiểu thực thể **MÔN_HOC** có thuộc tính **ngày_đăng_ký** để ghi lại ngày sinh viên đăng ký môn học.

Thực thể liên kết

Định nghĩa

Thực thể liên kết (Associative Entity) là thực thể được sử dụng để biểu diễn một mối kết hợp có thuộc tính.

Ví dụ: Trong cơ sở dữ liệu quản lý trường đại học, kiểu mỗi kết hợp **ĐĂNG_KÝ** giữa kiểu thực thể **SINH_VIÊN** và kiểu thực thể **MÔN_HOC** có thuộc tính **ngày_đăng_ký**. Ta có thể biểu diễn kiểu mỗi kết hợp này dưới dạng thực thể liên kết **ĐĂNG_KÝ** với các thuộc tính **ngày_đăng_ký**.

Thực thể liên kết (tiếp)

Thực thể liên kết giúp:

- Chuyển đổi mỗi kết hợp có thuộc tính thành một thực thể riêng biệt.
- Chuyển đổi mỗi kết hợp $M : N$ thành hai mối kết hợp $1 : N$.
- Chuyển đổi các kiểu kết hợp phức tạp thành các kiểu kết hợp đơn giản hơn.

Ví dụ thực thể liên kết

Trong cơ sở dữ liệu quản lý đại học, ta có kiểu mối kết hợp **GIẢNG_DẠY** giữa các kiểu thực thể **GIẢNG_VIÊN**, **SINH_VIÊN** và **MÔN_HỌC** với thuộc tính **ngày_học**. Trong thực tế, nhà trường sẽ tổ chức các lớp học cho sinh viên.

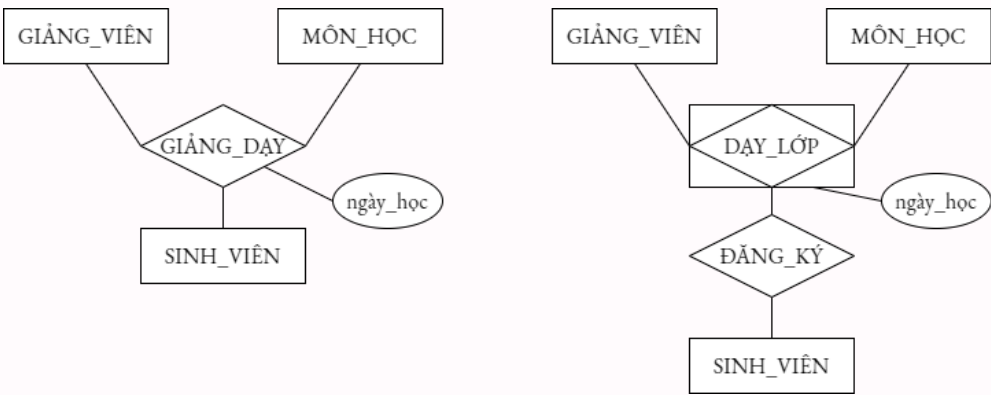
- Giảng viên dạy lớp học cho một môn học nào đó.
- Sinh viên đăng ký học lớp học.

Ví dụ thực thể liên kết (tiếp)

Ta có thể tách kiểu mối kết hợp **GIẢNG_DẠY** thành hai kiểu mối kết hợp **DẠY_LỚP** và **ĐĂNG_KÝ**, trong đó:

- Kiểu mối kết hợp **DẠY_LỚP** giữa kiểu thực thể **GIẢNG_VIÊN** và **MÔN_HỌC** với thuộc tính **ngày_học**.
- Kiểu mối kết hợp **DẠY_LỚP** sẽ trở thành kiểu thực thể liên kết **DẠY_LỚP**.
- Kiểu mối kết hợp **ĐĂNG_KÝ** giữa kiểu thực thể **SINH_VIÊN** và kiểu thực thể liên kết **DẠY_LỚP**.

Ví dụ thực thể liên kết (tiếp)



Chuyển đổi mỗi kết hợp có thuộc tính thành thực thể liên kết

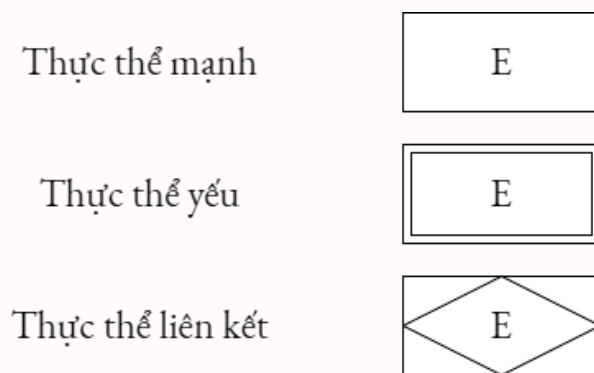
Cách biểu diễn sơ đồ

Cách biểu diễn sơ đồ

Sơ đồ thực thể – kết hợp (Entity – Relationship Diagram, ERD) được biểu diễn theo các quy ước cơ bản sau:

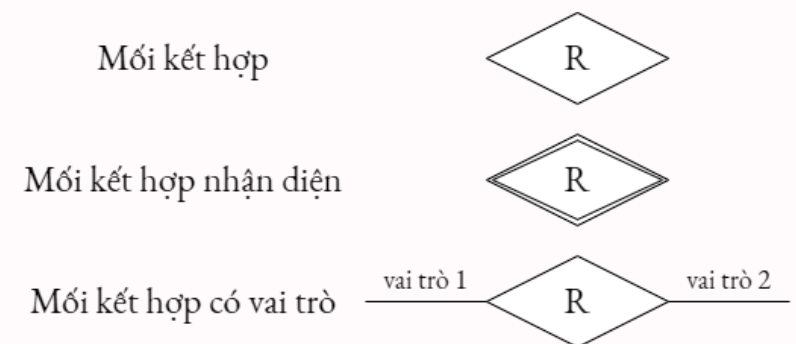
- Kiểu thực thể được biểu diễn bằng hình chữ nhật.
- Kiểu mối kết hợp được biểu diễn bằng hình thoi.
- Thuộc tính được biểu diễn bằng hình elip.
- Tên của kiểu thực thể, kiểu mối kết hợp và thuộc tính được ghi bên trong các hình tương ứng.
- Các liên kết giữa kiểu thực thể, kiểu mối kết hợp và thuộc tính được biểu diễn bằng các đường nối.
- Các ràng buộc của kiểu mối kết hợp được ghi trên các đường nối (dùng ký hiệu *(min, max)*).

Biểu diễn thực thể



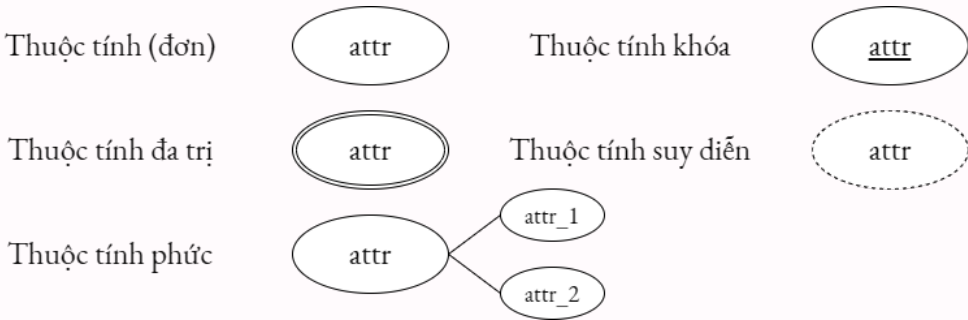
Biểu diễn kiểu thực thể

Biểu diễn mối kết hợp



Biểu diễn kiểu mối kết hợp

Biểu diễn thuộc tính



Biểu diễn thuộc tính

Biểu diễn ràng buộc của mối kết hợp

Trong sơ đồ thực thể – kết hợp, có nhiều cách để biểu diễn ràng buộc của kiểu mối kết hợp:

- Ký hiệu Chen.
- Ký hiệu Crow’s Foot (chân chim).
- **Ký hiệu (min, max).**
- ...

Trong bài này, ta sẽ sử dụng ký hiệu **(min, max)** để biểu diễn ràng buộc của kiểu mối kết hợp. Ký hiệu này có thể biểu diễn cả tỷ số lực lượng và ràng buộc tham gia.

Biểu diễn ràng buộc của mối kết hợp (tiếp)

Ký hiệu **(min, max)** là một cặp số nguyên không âm được ghi trên đường nối giữa kiểu thực thể và kiểu mối kết hợp, trong đó:

- **min** là số lượng tối thiểu các thể hiện kết hợp mà một thể hiện thực thể có thể tham gia trong kiểu mối kết hợp. *Giá trị thường dùng:* 0 hoặc 1.
- **max** là số lượng tối đa các thể hiện kết hợp mà một thể hiện thực thể có thể tham gia trong kiểu mối kết hợp. *Giá trị thường dùng:* 1 hoặc *n* (có thể tham gia với số lượng tùy ý).

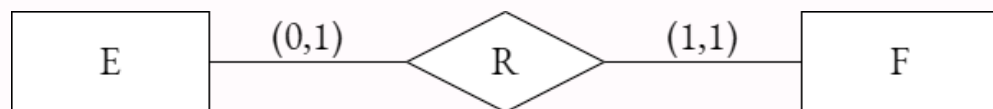
Biểu diễn ràng buộc của mối kết hợp (tiếp)

Ta có bảng sau:

Tỷ số lực lượng E:F	(min, max) bên E	(min, max) bên F
1 : 1	(0, 1)	(0, 1)
1 : N	(0, N)	(0, 1)
M : N	(0, N)	(0, N)

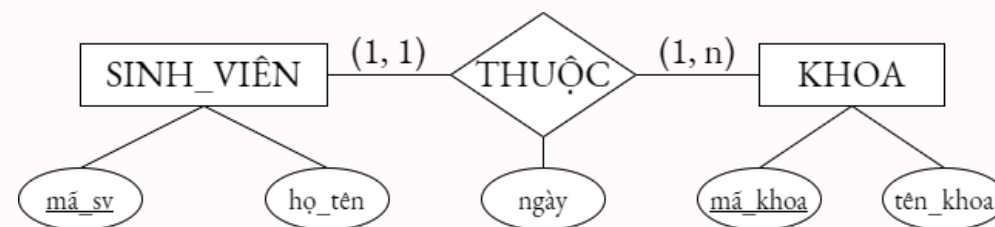
Lưu ý: Bảng trên đã đơn giản hóa các ký hiệu của tỷ số lực lượng và ràng buộc tham gia.

Biểu diễn ràng buộc của mối kết hợp (tiếp)



Biểu diễn ràng buộc của kiểu mối kết hợp

Sơ đồ thực thể – kết hợp đơn giản



Sơ đồ thực thể – kết hợp đơn giản

Các bước xây dựng sơ đồ thực thể – kết hợp

Các bước xây dựng sơ đồ thực thể – kết hợp bao gồm:

- 1 Xác định các thực thể và kiểu thực thể.
- 2 Xác định các mối kết hợp và kiểu mối kết hợp.
- 3 Xác định các thuộc tính của thực thể và mối kết hợp.
- 4 Xác định ràng buộc của kiểu mối kết hợp.
- 5 Vẽ sơ đồ thực thể – kết hợp.

Ví dụ

Thiết kế cơ sở dữ liệu quản lý forum:

- Các thành viên tham gia diễn đàn có mã số là duy nhất, họ tên thành viên và mật khẩu.
- Các bài viết có mã số bài viết, tiêu đề của bài viết và nội dung bài viết.
- Các chủ đề của các bài viết có mã số chủ đề (thuộc tính khóa) và tên của chủ đề.
- Mỗi bài viết được đăng bởi một thành viên và mỗi thành viên có thể đăng nhiều bài viết.
- Mỗi bài viết thuộc về một chủ đề và mỗi chủ đề có thể có nhiều bài viết.

Ví dụ (tiếp)

- 1 Xác định các thực thể và kiểu thực thể: THÀNH_VIÊN, BÀI_VIẾT, CHỦ_ĐỀ.

Ví dụ (tiếp)

- 1 Xác định các thực thể và kiểu thực thể: THÀNH_VIÊN, BÀI_VIẾT, CHỦ_ĐỀ.
- 2 Xác định các mối kết hợp và kiểu mối kết hợp: ĐĂNG_BÀI (giữa THÀNH_VIÊN và BÀI_VIẾT), THUỘC_CHỦ_ĐỀ (giữa BÀI_VIẾT và CHỦ_ĐỀ).

Ví dụ (tiếp)

- 1 Xác định các thực thể và kiểu thực thể: THÀNH_VIÊN, BÀI_VIẾT, CHỦ_ĐỀ.
- 2 Xác định các mối kết hợp và kiểu mối kết hợp: ĐĂNG_BÀI (giữa THÀNH_VIÊN và BÀI_VIẾT), THUỘC_CHỦ_ĐỀ (giữa BÀI_VIẾT và CHỦ_ĐỀ).
- 3 Xác định các thuộc tính của thực thể và mối kết hợp:
 - THÀNH_VIÊN(mã_số, họ_tên, mật_khẩu)
 - BÀI_VIẾT(mã_số_bài_viết, tiêu_đề, nội_dung)
 - CHỦ_ĐỀ(mã_số_chủ_đề, tên_chủ_đề)

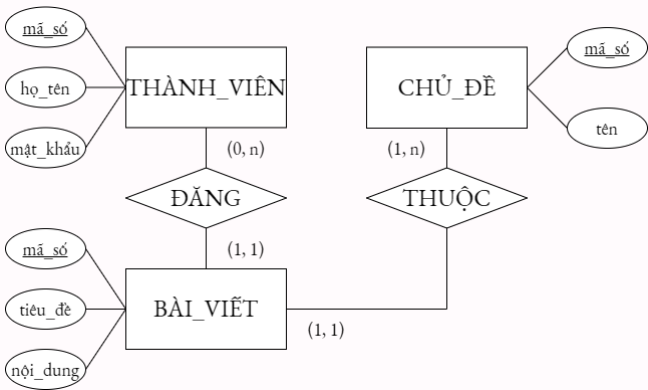
Ví dụ (tiếp)

- 1 Xác định các thực thể và kiểu thực thể: THÀNH_VIÊN, BÀI_VIẾT, CHỦ_ĐỀ.
- 2 Xác định các mối kết hợp và kiểu mối kết hợp: ĐĂNG_BÀI (giữa THÀNH_VIÊN và BÀI_VIẾT), THUỘC_CHỦ_ĐỀ (giữa BÀI_VIẾT và CHỦ_ĐỀ).
- 3 Xác định các thuộc tính của thực thể và mối kết hợp:
 - THÀNH_VIÊN(mã_số, họ_tên, mật_khẩu)
 - BÀI_VIẾT(mã_số_bài_viết, tiêu_đề, nội_dung)
 - CHỦ_ĐỀ(mã_số_chủ_đề, tên_chủ_đề)
- 4 Xác định ràng buộc của kiểu mối kết hợp:
 - ĐĂNG_BÀI: (0, N) bên THÀNH_VIÊN và (1, 1) bên BÀI_VIẾT.
 - THUỘC_CHỦ_ĐỀ: (1, N) bên CHỦ_ĐỀ và (1, 1) bên BÀI_VIẾT.

Ví dụ (tiếp)

- 1 Xác định các thực thể và kiểu thực thể: THÀNH_VIÊN, BÀI_VIẾT, CHỦ_ĐỀ.
- 2 Xác định các mối kết hợp và kiểu mối kết hợp: ĐĂNG_BÀI (giữa THÀNH_VIÊN và BÀI_VIẾT), THUỘC_CHỦ_ĐỀ (giữa BÀI_VIẾT và CHỦ_ĐỀ).
- 3 Xác định các thuộc tính của thực thể và mối kết hợp:
 - THÀNH_VIÊN(mã_số, họ_tên, mật_khẩu)
 - BÀI_VIẾT(mã_số_bài_viết, tiêu_đề, nội_dung)
 - CHỦ_ĐỀ(mã_số_chủ_đề, tên_chủ_đề)
- 4 Xác định ràng buộc của kiểu mối kết hợp:
 - ĐĂNG_BÀI: (0, N) bên THÀNH_VIÊN và (1, 1) bên BÀI_VIẾT.
 - THUỘC_CHỦ_ĐỀ: (1, N) bên CHỦ_ĐỀ và (1, 1) bên BÀI_VIẾT.
- 5 Vẽ sơ đồ thực thể – kết hợp.

Ví dụ (tiếp)



Sơ đồ thực thể – kết hợp của cơ sở dữ liệu quản lý forum

Mô hình thực thể – kết hợp mở rộng (Enhanced ER Model)

Giới thiệu chung

Mô hình thực thể – kết hợp mở rộng (Enhanced Entity – Relationship Model, EER Model) là một sự mở rộng của mô hình thực thể – kết hợp (ER Model). Mô hình EER bổ sung thêm các khái niệm và tính năng để mô tả các yêu cầu phức tạp hơn của hệ thống cơ sở dữ liệu.

Bên cạnh các khái niệm của mô hình thực thể – kết hợp, mô hình mở rộng bổ sung thêm các khái niệm:

- Kiểu cha (supertype), kiểu con (subtype) và các khái niệm liên quan: chuyên biệt hóa (specialization) và tổng quát hóa (generalization).
- Kiểu hợp (union) hoặc phân loại (category).
- Kế thừa thuộc tính và liên kết (attribute and relationship inheritance).

Kiểu cha (supertype) và kiểu con (subtype)

Kiểu cha (supertype) và kiểu con (subtype)

Định nghĩa

Kiểu cha (Supertype) là kiểu thực thể tổng quát, bao gồm các thuộc tính và mối kết hợp chung cho một tập hợp các kiểu con.

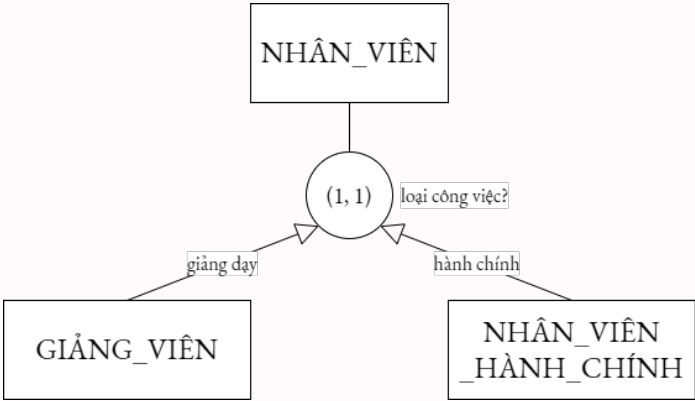
Định nghĩa

Kiểu con (Subtype) là kiểu thực thể chuyên biệt, kế thừa các thuộc tính và mối kết hợp từ kiểu cha và có thể có các thuộc tính và mối kết hợp riêng.

Trong cơ sở dữ liệu quản lý trường đại học, ta có kiểu thực thể **NHÂN_VIÊN** có thể được chia (*chuyên biệt hóa* – specialization) thành các kiểu con:

- Kiểu con **GIẢNG_VIÊN** có thêm thuộc tính chuyên môn như bằng_cấp, khoa.
- Kiểu con **NHÂN_VIÊN_HÀNH_CHÍNH** có thêm thuộc tính phòng_ban.

Hoặc ta có thể nói kiểu thực thể **GIẢNG_VIÊN** và **NHÂN_VIÊN_HÀNH_CHÍNH** được *tổng quát hóa* (generalization) thành kiểu cha **NHÂN_VIÊN**.



Biểu diễn kiểu cha và kiểu con

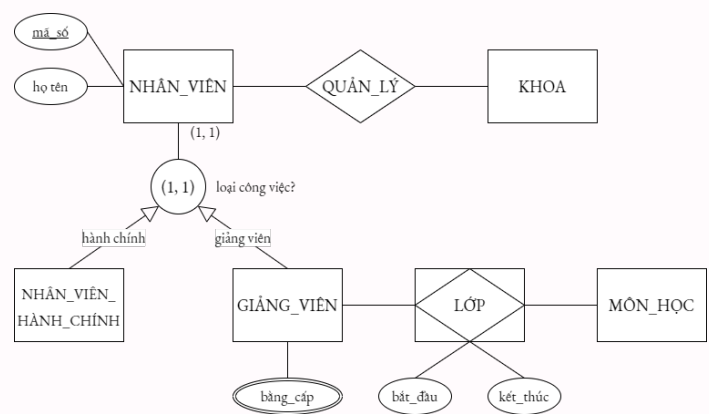
Một thực thể thuộc một kiểu con kế thừa (có) tất cả các thuộc tính của kiểu cha. Một thực thể thuộc kiểu con cũng có thể có các thuộc tính và tham gia vào các liên kết riêng của kiểu con đó.

Ví dụ: Một thực thể thuộc kiểu con **GIẢNG_VIÊN** kế thừa các thuộc tính như mã_số, họ_tên từ kiểu cha **NHÂN_VIÊN** và có thêm các thuộc tính riêng như bằng_cấp, khoa.

Một thực thể thuộc kiểu con cũng kế thừa (tham gia) mối kết hợp của kiểu cha. Ngoài ra, nó cũng có thể tham gia vào các mối kết hợp riêng của kiểu con đó (những kiểu con riêng biệt khác không tham gia).

Ví dụ: Một thực thể thuộc kiểu con **GIẢNG_VIÊN** tham gia vào mối kết hợp **QUẢN_LÝ** của kiểu cha **NHÂN_VIÊN** và có thể tham gia vào mối kết hợp riêng **GIẢNG_DẠY**. Kiểu cha **NHÂN_VIÊN** cũng có thể xem là tham gia vào mối kết hợp **GIẢNG_DẠY**, nhưng như thế không phản ánh đúng thực tế.

Kế thừa (inheritance) (tiếp)



Thuộc tính và mối kết hợp riêng

Chuyên biệt hóa (specialization)

Định nghĩa

Chuyên biệt hóa (Specialization) là quá trình xác định một tập các kiểu con từ một kiểu cha dựa trên các thuộc tính hoặc mối kết hợp riêng biệt.

Ví dụ: Trong cơ sở dữ liệu quản lý trường đại học, ta có thể chuyên biệt hóa kiểu thực thể **NHAN_VIEN** dựa trên loại công việc để tạo ra các kiểu con:

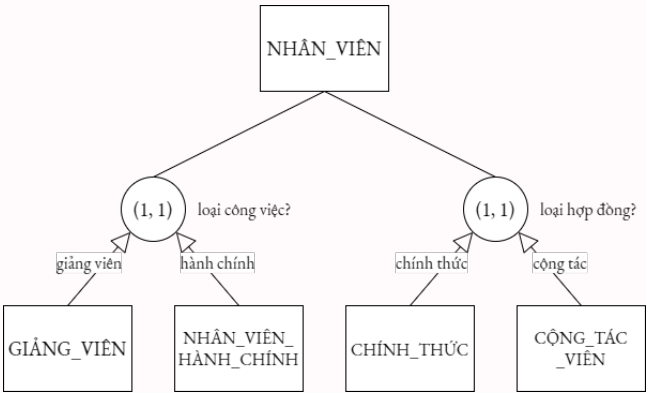
- Kiểu con **GIANG_VIEN** với loại công việc là giảng dạy.
- Kiểu con **NHAN_VIEN_HANH_CHINH** với loại công việc là hành chính.

Chuyên biệt hóa (specialization) (tiếp)

Một kiểu cha có thể được chuyên biệt hóa thành nhiều cách khác nhau.

Ví dụ: Kiểu thực thể **NHAN_VIEN** còn có thể được chuyên biệt hóa theo tính chất hợp đồng:

- Kiểu con **NHAN_VIEN_CHINH_THUC** với tính chất hợp đồng là chính thức.
- Kiểu con **CONG_TAC_VIEN** với tính chất hợp đồng là cộng tác viên.



Nhiều cách chuyên biệt hóa

Chuyên biệt hóa (specialization) (tiếp)

Tổng quát hóa (generalization)

Định nghĩa

Tổng quát hóa (Generalization) là quá trình kết hợp nhiều kiểu con thành một kiểu cha chung dựa trên các thuộc tính hoặc mối kết hợp chung. Có thể nói, tổng quát hóa là quá trình ngược với chuyên biệt hóa.

Ví dụ: Trong cơ sở dữ liệu quản lý trường đại học, ta có thể tổng quát hóa các kiểu thực thể **GIẢNG_VIÊN** và **NHÂN_VIÊN_HÀNH_CHÍNH** thành kiểu cha chung **NHÂN_VIÊN**, vì cả hai kiểu con đều chia sẻ các thuộc tính chung như mã_nhân_viên, họ_tên.

Phân cấp chuyên biệt hóa và dàn chuyên biệt hóa

Một kiểu con có thể tiếp tục được chuyên biệt hóa thành nhiều kiểu con, tạo thành phân cấp (hierarchy) hoặc dàn (lattice) chuyên biệt hóa.

Ví dụ: Kiểu thực thể **GIẢNG_VIÊN** có thể được chuyên biệt hóa theo loại hợp đồng thành:

- Kiểu thực thể **GIẢNG_VIÊN_CƠ_HỮU** nếu loại hợp đồng là chính thức.
- Kiểu thực thể **GIẢNG_VIÊN_THỈNH_GIẢNG** nếu loại hợp đồng là thỉnh giảng.

Phân cấp chuyên biệt hóa và dàn chuyên biệt hóa (tiếp)

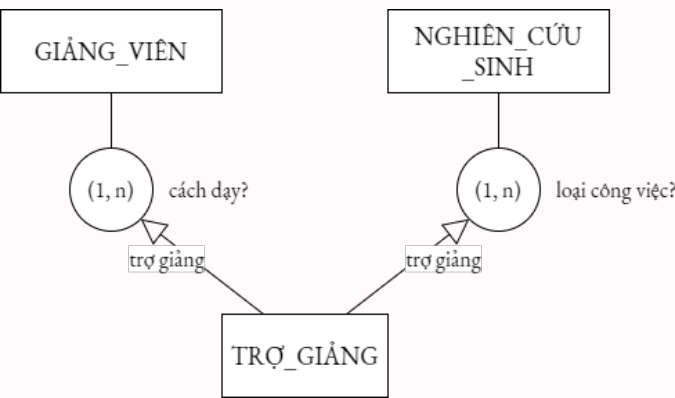
Định nghĩa

Phân cấp chuyên biệt hóa là khi mỗi kiểu con **chỉ có một** kiểu cha.

Định nghĩa

Dàn chuyên biệt hóa là khi có kiểu con có **từ hai** kiểu cha trở lên.

Phân cấp chuyên biệt hóa và dàn chuyên biệt hóa (tiếp)



Kiểu con có hai kiểu cha

Cách xác định các kiểu con

- Trong một số chuyên biệt hóa, có thể xác định chính xác các thực thể sẽ là thành viên của một kiểu con bằng một điều kiện trên một số thuộc tính của kiểu cha: *các kiểu con xác định bằng điều kiện*.
- Nếu các kiểu con của một chuyên biệt hóa có điều kiện thành viên trên cùng một thuộc tính của kiểu cha thì chuyên biệt hóa đó còn được gọi là chuyên biệt hóa *xác định bằng thuộc tính*.
- Nếu việc xác định một kiểu con không thực hiện được theo một điều kiện nào: các lớp con được *người sử dụng xác định*.

Ràng buộc của kiểu cha – kiểu con

Có hai loại ràng buộc áp dụng cho chuyên biệt hóa:

- Ràng buộc về tính đầy đủ** (Completeness constraint)
- Ràng buộc về tính phân ly** (Disjointness constraint)

Ràng buộc về tính đầy đủ

Có hai loại:

- Chuyên biệt hóa toàn phần** (Total specialization): một thực thể của kiểu cha phải thuộc một kiểu con nào đó.
- Chuyên biệt hóa một phần** (Partial specialization): một thực thể của kiểu cha không nhất thiết thuộc một kiểu con nào.

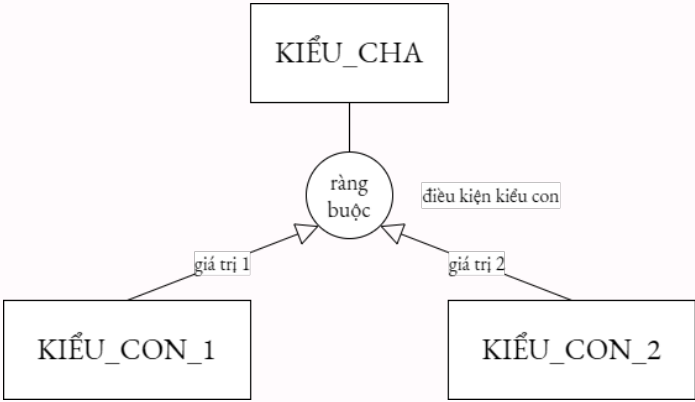
Ràng buộc về tính phân ly

Có hai loại:

- Chuyên biệt hóa rời rạc** (Disjoint): một thực thể của kiểu cha chỉ thuộc một kiểu con nào đó.
- Chuyên biệt hóa trùng lặp** (Overlap): một thực thể của kiểu cha có thể thuộc nhiều hơn một kiểu con.

Hai loại ràng buộc này là độc lập với nhau, tức là, ta có bốn loại ràng buộc:

- Rời rạc, toàn phần.
- Rời rạc, một phần.
- Trùng lặp, toàn phần.
- Trùng lặp, một phần.



Biểu diễn kiểu cha và kiểu con

Trong đó:

- **ràng buộc:** sẽ được ghi theo ký hiệu (*min*, *max*).
 - Rời rạc, toàn phần: (1, 1).
 - Rời rạc, một phần: (0, 1).
 - Trùng lặp, toàn phần: (1, *N*).
 - Trùng lặp, một phần: (0, *N*).
- **điều kiện kiểu con:** điều kiện dùng để chuyên biệt hóa.
- **giá trị:** giá trị của điều kiện để xác định chính xác kiểu con.

Kiểu hợp (union)

Kiểu hợp (union)

Trong nhiều trường hợp, ta cần phải tổng quát hoá nhiều kiểu thực thể không có nhiều điểm chung lại với nhau. Ta gọi kiểu được tổng quát hoá là kiểu hợp (union) hay kiểu phân loại (category).

Kiểu hợp (union) (tiếp)

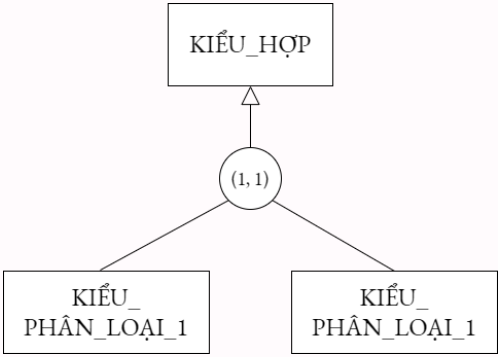
Ví dụ: Trong email, ta có thể đính kèm nhiều loại tập tin khác nhau như văn bản, hình ảnh, video, ... Ta có thể tạo một kiểu hợp là **TỆP_ĐÍNH_KÈM** để tổng quát hoá các kiểu **VĂN_BẢN**, **HÌNH_ẢNH**, **ÂM_THANH**, ...

Kiểu hợp (union) (tiếp)

Vì kiểu hợp là tổng quát hóa từ các kiểu khác nhau nên kiểu hợp:

- Không có thuộc tính để kế thừa.
- Ràng buộc là rời rạc, toàn phần.

Biểu diễn kiểu hợp



Biểu diễn kiểu hợp

- Một dàn nhạc giao hưởng muốn lưu trữ lại thông tin các buổi hòa nhạc, biết rằng:
- Trong năm, dàn nhạc tổ chức nhiều buổi hòa nhạc khác nhau với thời gian, địa điểm xác định trước và một mã số riêng.
 - Tại mỗi buổi hòa nhạc sẽ biểu diễn nhiều bài hòa tấu, các bài hòa tấu bao gồm tên và mô tả. Ngoài ra, trong một buổi hòa nhạc, các bài hòa tấu được chơi theo một thứ tự nhất định, cần lưu trữ lại thứ tự này.
 - Mỗi buổi hòa nhạc cần có một nhạc trưởng, cần lưu trữ tên, địa chỉ liên lạc cũng như email của nhạc trưởng này.

Chen, P. P.-S. (1976). The entity-relationship model – toward a unified view of data. *ACM Transactions on Database Systems*, 1(1), 9–36. <https://doi.org/10.1145/320434.320440>

Entity-Relationship-Diagram (ERD) – Michael Fuchs SQL. (2021). Netlify.app. <https://michael-fuchs-sql.netlify.app/2021/03/03/entity-relationship-diagram-erd>

Slide bài giảng CSDL (2025) TS. Nguyễn Hải Châu, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội. [https://uet.vnu.edu.vn/ chaunh/slide/](https://uet.vnu.edu.vn/chaunh/slide/)