

## Bài 6

Bài này giới thiệu sơ lược về mô hình hóa dữ liệu và biến đổi mô hình đó về các bảng của mô hình quan hệ.

### 1./ Tổng quan về mô hình hóa dữ liệu

Mô hình hóa là một bước rất quan trọng khi phát triển phần mềm và xây dựng CSDL.

Một mô hình dữ liệu là biểu diễn ở mức khái niệm cấu trúc dữ liệu của CSDL. Cấu trúc dữ liệu bao gồm các đối tượng, mối liên kết giữa các đối tượng và các qui tắc chi phối các đối tượng. Mô hình chỉ tập trung vào việc nhận thức rõ cái gì là cần và cần được tổ chức như thế nào. Người ta thường không quan tâm nhiều đến các phép toán thực hiện trên các đối tượng, chủ yếu quan tâm chúng là gì và chúng liên kết với nhau như thế nào. Có thể hình dung một cách trực quan là mô hình dữ liệu giống với kiến trúc một tòa nhà.

Mô hình dữ liệu hoàn toàn độc lập với phần cứng và phần mềm. Mô hình dữ liệu không phải là biểu diễn theo một HQT CSDL cụ thể nào, mục đích của nó là biểu diễn cho người dùng xem và nó phải thể hiện gần với thế giới thực nhất. Mô hình dữ liệu đồng thời phải đóng vai trò cầu nối giữa các sự kiện và qui trình của thế giới thực với các khái niệm cơ bản trong CSDL.

### § Phương pháp tiếp cận

Trong thực tế hiện nay, mô hình hóa dữ liệu có hai phương pháp tiếp cận chính là cách tiếp cận theo mô hình quan hệ thực thể (Entity Relationship – ER) và cách tiếp cận theo mô hình đối tượng dữ liệu. Bài này chỉ quan tâm đến mô hình ER.

### § Mô hình hóa dữ liệu đặt trong ngữ cảnh của Thiết kế CSDL

Trước hết ta phải cùng thống nhất về khái niệm Thiết kế CSDL. Thiết kế CSDL là gì?

*“Thiết kế CSDL là thiết kế cấu trúc logic và cấu trúc vật lý của một hoặc nhiều CSDL nhằm thỏa mãn nhu cầu thông tin của người dùng trong một tổ chức nào đó với một tập hợp các ứng dụng đã được xác lập rõ ràng”.*

Qui trình thiết kế gồm các bước chính sau:

1. Khảo sát, phân tích, lập kế hoạch
2. Thiết kế mức khái niệm
3. Thiết kế logic
4. Thiết kế vật lý
5. Thiết kế cơ chế cài đặt, chuyển giao, chạy thử và vận hành

Bước 2 trên đây của thiết kế CSDL có mối quan hệ chặt chẽ với mô hình hóa dữ liệu. Các phần khác ít nhiều liên quan đến việc thiết kế chức năng. Nói cụ thể hơn trong mô hình CSDL quan hệ, mô hình hóa dữ liệu chính là thiết kế các bảng, còn thiết kế chức năng chính là thiết kế các câu truy vấn, trong đó bao gồm cả truy vấn cập nhật và truy vấn kết xuất thông tin.

## § Các hợp phần của mô hình hóa dữ liệu

Đầu vào của mô hình hóa dữ liệu là kết quả khảo sát, phân tích và lập kế hoạch (bước 1 nói trên). Người lập mô hình đồng thời cũng là người thu thập thông tin, thu thập các đặc tả về yêu cầu của người dùng cuối và có thể phải tiếp xúc, phỏng vấn người dùng về các nhu cầu trong các ứng dụng sau này của họ.

Mô hình hóa phải có 2 kết quả đầu ra. Đầu ra thứ nhất là sơ đồ quan hệ thực thể (Entity Relationship Diagram - ERD) và sơ đồ đồ này phải mô tả được cấu trúc và quan hệ dữ liệu dưới dạng hình vẽ. Sơ đồ là một công cụ rất hữu hiệu trong giao lưu thông tin với người dùng cuối. Đầu ra thứ hai là tài liệu thiết kế. Tài liệu thiết kế mô tả một cách chi tiết các đối tượng, các mối quan hệ và các ràng buộc giữa chúng. Bên cạnh đó, tùy theo mức độ, người ta có thể cần đến cả từ điển dữ liệu, để xây dựng các bảng cụ thể theo HQT CSDL cụ thể.

Tóm lại:

Đầu vào:      khảo sát, phân tích, kế hoạch  
Đầu ra:        ERD, tài liệu thiết kế, từ điển dữ liệu

## § Tại sao lại cần đến mô hình hóa dữ liệu

Mô hình hóa là một công đoạn rất tốn thời gian và công sức trong quá trình phát triển ứng dụng. Câu hỏi đặt ra là liệu có cần đến công đoạn này hay không, đặc biệt là đối với các dự án cần thực hiện nhanh do sức ép về tiến độ và thời gian? Câu trả lời rất đơn giản: ***xây dựng một CSDL thiếu mô hình hóa dữ liệu cũng giống như xây nhà không có thiết kế.***

Mục tiêu của một mô hình dữ liệu là tất cả các đối tượng cần thiết đều có trong bản thiết kế. Vì mô hình dữ liệu sử dụng các thuật ngữ thông thường và ký hiệu dễ hiểu nên rất dễ kiểm tra, sửa đổi và người dùng có thể kiểm tra tính đúng đắn của nó so với nghiệp vụ của họ.

Mặt khác, mô hình dữ liệu cũng đạt được mức độ chi tiết cần thiết và đó là cơ sở để đưa ra bản thiết kế trong HQT CSDL một cách dễ dàng. Các số liệu trong mô hình sẽ được sử dụng để thiết lập các bảng, các khóa chính, khóa ngoại, các thủ tục lưu (stored procedures) và các bẫy (*triggers*). Một bản thiết kế tồi sẽ dẫn đến các công việc sau này bị rối và làm cho việc thực hiện bị chậm trễ. Các bản thiết

kế tài thường dẫn đến thiếu thông tin, mối quan hệ không đúng dẫn đến dữ liệu bất nhất và làm cho các nghiệp vụ không được đảm bảo.

Nói tóm lại, mô hình dữ liệu là một kế hoạch cụ thể xây dựng CSDL. Để nâng cao tính hiệu quả, mô hình phải đơn giản để cho người dùng có thể hiểu được cấu trúc và đồng thời cũng phải đủ chi tiết để thiết lập dễ dàng CSDL quan hệ. Mô hình quan hệ thực thể (ER) là mô hình được sử dụng một cách rộng rãi trong thiết kế CSDL quan hệ.

## 2./ **Mô hình quan hệ thực thể (Entity-Relationship Model)**

Năm 1976, Peter Cheng đề xuất mô hình ER với mục tiêu là thống nhất mô hình mạng với mô hình quan hệ.

Một cách đơn giản, mô hình ER phân chia thế giới thành các thực thể và quan hệ giữa chúng. Tiêu điểm của mô hình là sơ đồ quan hệ thực thể (ERD) và sơ đồ này dùng để biểu diễn các đối tượng. Từ thời điểm bài báo đưa ra đến nay đã có nhiều cải tiến thay đổi và hiện nay mô hình này được chấp nhận như là một công cụ thiết kế CSDL cơ bản. Công cụ này có các ưu điểm chính như sau:

- Mô hình ER là một ánh xạ của mô hình quan hệ. Các đơn thể trong mô hình ER có thể dễ dàng chuyển sang thành các bảng.
- Nó rất đơn giản và dễ hiểu, và vì vậy trở thành công cụ rất hữu hiệu nhằm giao lưu ý tưởng giữa nhà thiết kế và người sử dụng.

## 3./ **Các đơn thể của mô hình ER**

### 3.1.- **Thực thể**

Thực thể là đối tượng dữ liệu mà ở đó ta cần lưu thông tin. *Thực thể là khái niệm có thể nhận biết được, trừu tượng hoặc cụ thể, như người, vật, địa danh, các sự kiện, ... miễn là nó có mối liên quan đến CSDL.* Ví dụ về thực thể: [sinh\\_vien](#), [mon\\_hoc](#), [khoa](#), [bo\\_mon](#). Một thực thể trong mô hình ER giống như một bảng trong mô hình quan hệ.

Thực thể được chia thành thực thể *độc lập* (**independent**) và thực thể *phụ thuộc* (**dependent**) - thuật ngữ tương đương là thực thể *mạnh* và thực thể *yếu*. Thực thể độc lập có thể nhận dạng một cách độc lập - thực thể phụ thuộc phải nhờ vào một thực thể khác để xác định.

Một **hiện hữu** (occurrence) là một cá thể của thực thể đó. Một hiện hữu của thực thể giống như một bản ghi của một bảng trong mô hình quan hệ.

### 3.2.- Thực thể đặc biệt

- Thực thể liên hợp (còn gọi là thực thể giao) là thực thể dùng để liên kết quan hệ nhiều-nhiều (xem các phần sau).
- **Thực thể con - thực thể mẹ** nằm trong mối tương quan phân cấp

### 3.3.- Quan hệ

Quan hệ biểu diễn mối tương quan giữa 2 hoặc nhiều thực thể. Ví dụ:

*Nhân viên tham gia vào các dự án*

*Dự án có nhiều công đoạn*

*Khoa quản lý nhiều dự án*

### 3.4.- Thuộc tính

Thuộc tính mô tả các đặc trưng của một thực thể. Một hiện hữu của thuộc tính được gọi là giá trị. Ví dụ, “Nguyễn Sáng Tạo” là một giá trị của trường [ho\\_va\\_ten](#). Mỗi một thuộc tính có giá trị thuộc một miền giá trị nhất định. Ví dụ [ho\\_va\\_ten](#) có miền giá trị là chuỗi ký tự.

Có hai loại thuộc tính: **thuộc tính khóa** và **thuộc tính mô tả**. Thuộc tính khóa là loại thuộc tính có giá trị duy nhất cho mỗi hiện hữu.

### 3.5.- Phân loại quan hệ

Các quan hệ được phân loại theo **bậc** (degree), **tính giao kết** (connectivity), **số lượng** (cardinality), và **tính tồn tại** (existence).

**Bậc của quan hệ (degree):** số lượng thực thể tham gia vào mối quan hệ đó. Quan hệ bậc 2 gọi là quan hệ nhị nguyên (tay đôi), quan hệ bậc 3 là quan hệ tam nguyên, ... Quan hệ nhị nguyên chính là quan hệ phổ biến nhất trong thế giới thực.

Quan hệ nhị nguyên có thể đệ quy: ví dụ thực thể người có các quan hệ anh em, quan hệ bố mẹ, ....

Quan hệ tam nguyên (bậc 3) gồm 3 thực thể. Ví dụ, giáo viên – môn học – giáo trình là một mối quan hệ tam nguyên. Quan hệ n-nguyên gồm n thực thể. Phần

lớn các mô hình dữ liệu đều tách các quan hệ n-nguyên thành các quan hệ nhị nguyên.

### Tính giao kết (connectivity) và số lượng (cardinality)

Tính giao kết thể hiện ánh xạ giữa các hiện hữu trong mỗi quan hệ đó. Các giá trị của nó được đặt tên là “một”, “nhiều”. Cardinality là số lượng cụ thể của các giá trị đó. Người ta gọi chung các loại quan hệ là quan hệ một-một (1:1), quan hệ một-nhiều (1:N) và quan hệ nhiều-nhiều (M:N).

Quan hệ một-một (1:1): khi một hiện hữu của thực thể A tương ứng với đúng một hiện hữu của thực thể B. Ví dụ:

Một nhân viên tương ứng với một bàn làm việc.

Quan hệ một-nhiều (1:N): khi một hiện hữu của thực thể A tương ứng với 0, 1 hoặc nhiều hiện hữu của thực thể B nhưng một hiện hữu của thực thể B chỉ tương ứng với duy nhất một hiện hữu của thực thể A. Ví dụ:

Một đơn vị có nhiều nhân viên;  
Một nhân viên chỉ thuộc một đơn vị.

Quan hệ nhiều-nhiều (M:N): khi một hiện hữu của thực thể A tương ứng với 0, 1 hoặc nhiều hiện hữu của thực thể B và ngược lại một hiện hữu của thực thể B tương ứng với 0, 1 hoặc nhiều hiện hữu của thực thể A. Ví dụ:

Một nhân viên đồng thời có thể tham gia nhiều công việc  
Một công việc có thể có nhiều nhân viên cùng thực hiện

### Tính tồn tại

Trong quan hệ nhị nguyên, khi *một hiện hữu của thực thể A tồn tại có suy ra sự tồn tại của một hiện hữu của thực thể B hay không* (A và B có mối giao kết nào đó) được gọi là tính tồn tại. Nếu B phải tồn tại thì gọi là “bắt buộc”. Nếu B không bắt buộc phải tồn tại thì gọi là “tùy chọn”. Ví dụ:

Một nhân viên tham gia vào các dự án

thì có thể xảy ra trường hợp là một nhân viên X nào đó có thể không tham gia vào dự án nào cả.

### 3.6.- Các ký hiệu

Cho đến thời điểm hiện nay, chưa tồn tại một “chuẩn” nào cho các ký hiệu. Tuy nhiên, chúng ta sẽ theo qui ước sau đây:

**Thực thể** được vẽ bằng hình chữ nhật có nhãn và nhãn là tên của thực thể. Tên của thực thể cần được thể hiện bằng danh từ số ít.

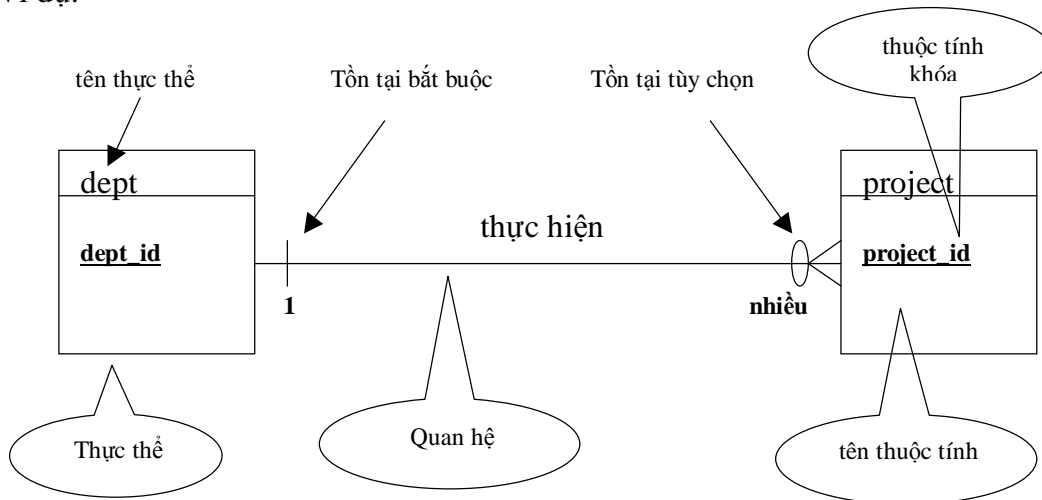
**Quan hệ** được thể hiện bằng đường nét liền nối 2 thực thể. Tên của quan hệ được viết trên nét liền đó. Quan hệ được thể hiện bằng động từ.

**Thuộc tính**, nếu được thể hiện, cần được liệt kê phía trong hình chữ nhật của thực thể. Thuộc tính là danh từ số ít. Các thuộc tính khóa được gạch dưới (hoặc ghi kèm các chữ viết tắt như PK, FK).

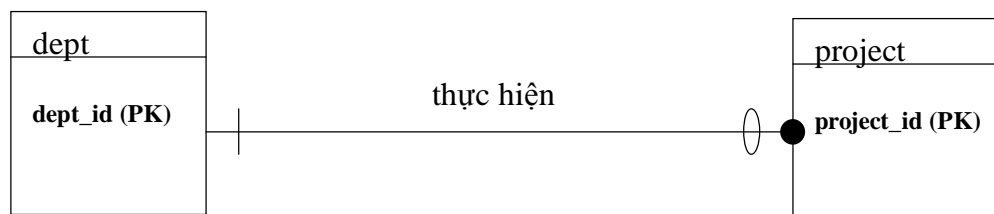
**Giao kết “nhiều”** được thể hiện bằng chân quạ 3 ngón (hoặc dấu tròn đậm – ký hiệu của IDEFIX). Nếu không phải chân quạ 3 ngón (hoặc dấu tròn đậm) thì đó là **giao kết “một”**.

**Tính tồn tại** được thể hiện bằng một **vòng tròn** hoặc **vạch đứng**. Nếu thực thể đó là bắt buộc thì ta vẽ vạch đứng. Nếu thực thể đó là tùy chọn thì ta vẽ vòng tròn. Đọc vạch đứng là 1 và đọc vòng tròn là 0 (không).

Ví dụ:



Hình 1: Các ký hiệu trong ERD



Hình 2: Sơ đồ ERD lược giản (theo qui ước IDEFIX)

#### 4./ Nhận dạng các đối tượng và quan hệ của chúng

Trước khi xây dựng mô hình, chúng ta phải nhận biết đâu là thực thể, đâu là thuộc tính và đâu là mối quan hệ giữa chúng.

Thông tin lấy từ đâu? Thông tin được lấy từ kết quả khảo sát phân tích yêu cầu của bài toán, từ nghiệp vụ, từ các tài liệu, từ các lần phỏng vấn, ...

Khi có thông tin rồi, làm thế nào ta nhận biết đâu là **thực thể**, đâu là **thuộc tính**, đâu là mối **quan hệ**? Trong lý thuyết mô hình hóa quan hệ thực thể, người ta không thể biến đổi được từ một lô các tài liệu và thông tin thành mô hình quan hệ thực thể. Người thiết kế phải làm việc này.

Quan hệ tương đối dễ nhận biết (quan hệ là các động từ) nhưng thực thể và thuộc tính thường dễ bị lẫn lộn. Ví dụ, ta cần mô hình hóa bài toán “**các chuyên viên làm việc cho dự án**”. Rõ ràng, dự án sẽ là một thực thể. Liệu chuyên viên là một thực thể độc lập hay chỉ là một thuộc tính của dự án? Câu trả lời còn tùy vào trường hợp cụ thể. Chẳng hạn, nếu văn phòng của dự án có một số lượng ít chuyên viên (5-7 người) thì chuyên viên sẽ trở thành một **thuộc tính** của thực thể dự án. Trong trường hợp đơn vị thực hiện dự án có đông các chuyên viên và các chuyên viên đồng thời có thể tham gia nhiều dự án thì chuyên viên lúc đó trở thành một **thực thể**.

Chỉ có một số qui tắc:

- **Thực thể** thường hàm chứa thông tin mô tả
- **Thuộc tính** thường đóng vai trò mô tả thực thể hoặc là thông tin dùng để nhận biết thực thể
- **Mối quan hệ** chỉ áp dụng đối với thực thể, không áp dụng cho thuộc tính

#### 4.1.- Nhận biết thực thể

Sau đây là một số qui tắc giúp ta nhận biết thực thể:

- Thực thể là đối tượng hoặc khái niệm;
- Nếu ta có đối tượng A và ta lại có nhiều đối tượng khác mô tả đối tượng A thì A chắc chắn là một thực thể. Nếu ta không tìm thấy bất cứ một đối tượng khác mô tả A thì A không phải là thực thể;
- Một thực thể thường là đại diện cho một lớp đối tượng và các đối tượng đó có một số đặc điểm chung. Ví dụ *Hamlet* và *Vua Lia* là các vở kịch và các vở kịch đều có đặc điểm chung là có **tên**, có **tác giả**, có các **nhân vật**. Từ đó ta có thực thể **vở kịch**, và “Hamlet” và “Vua Lia” là các **hiện hữu** của **vở kịch**.
- Có một số thực thể là lõi của một số các thực thể khác. Ví dụ thực thể **người** là lõi chung cho thực thể **giáo viên** và thực thể **sinh viên**;

- Thực thể không nên để phụ thuộc vào thời gian. Ví dụ, ta không nên có các thực thể **quí 1**, **quí 2**, **quí 3**, **quí 4** và chỉ nên có thực thể **quí**;
- Người dùng cuối thường áp đặt thông tin lên mô hình vì họ cho đó là quan trọng đối với họ. Chú ý là các loại thông tin này không nhất thiết phải trở thành thực thể, phần lớn các trường hợp ta phải tách chúng ra;

#### 4.2.- Nhận biết thuộc tính

Các thuộc tính hoặc là loại dữ liệu có tính duy nhất (như số *Chứng minh thư*, số *hiệu sinh viên*) hoặc là loại dữ liệu mô tả các đối tượng khác (ngày tháng năm sinh, quê quán).

Loại dữ liệu có tính duy nhất gọi là *thuộc tính khóa*. Các thuộc tính khác gọi là *thuộc tính mô tả*.

Các thuộc tính phải nguyên tử - nghĩa là một thuộc tính không thể phân tích thành tổ hợp của các thuộc tính khác. Quan điểm thế nào là nguyên tử cũng tùy theo ngữ cảnh. Ví dụ, thuộc tính họ tên. Liệu có cần tách tên ra khỏi họ và đệm không? Trong ngữ cảnh nào thì tách và trong ngữ cảnh nào thì không?

#### § Thuộc tính suy diễn

***Thuộc tính suy diễn là thuộc tính mà giá trị của nó là kết quả của tính toán trên các thuộc tính khác.*** Đây là loại thuộc tính mà về mặt lý thuyết mô hình quan hệ, ta không được lưu vì nó gây ra dư thừa thông tin. Tuy nhiên, trong mô hình ER, có hai trường phái: một trường phái ủng hộ việc có mặt thuộc tính suy diễn trong mô hình ER và trường phái kia không ủng hộ việc đó. Lý do gì cần sự có mặt của thuộc tính suy diễn?

- Dưới góc độ nghiệp vụ, người ta cần các thông tin đó
- Cần nêu rõ, ít ra là trong mô hình, sự có mặt của các thuộc tính đó vì mô hình ER là mô hình dành cho cả người thiết kế lẫn người dùng. Các thuộc tính gắn với nghiệp vụ của người dùng sẽ thuyết phục hơn.

#### § Giá trị là mã

Có một số loại thuộc tính, người thiết kế hay đưa các mã vào. Chẳng hạn, thay vì đưa vào các giá trị đầy đủ, người ta hay viết tắt tên các thành phố bằng 3 chữ cái như Hà Nội có mã là HAN, TP. Hồ Chí Minh có mã là HCM, ... Hay như mã ngôn ngữ cũng thường chỉ có hai chữ cái: EN = tiếng Anh, VI = tiếng Việt, FR = tiếng Pháp, ...



### 4.3.- Nhận biết quan hệ

Quan hệ là liên kết giữa các thực thể. Thông thường ta dùng động từ để biểu diễn quan hệ.

**Nhân viên** được *chỉ định* thực hiện **dự án**.

Quan hệ có đặc trưng về số lượng (**1:1, 1:N, M:N**), tính tồn tại (**bắt buộc, tùy chọn**).

Ví dụ về số lượng: một nhân viên tham gia nhiều nhất vào 3 dự án, mỗi một dự án có ít nhất 5 người thực hiện, ...

Ví dụ về tính tồn tại: nhân viên có thể tham gia dự án hoặc không, sinh viên bắt buộc phải có mặt ở lớp học.

### 4.4.- Đặt tên và qui ước

Qui tắc chung:

- Tên là duy nhất
- Tên phải có nghĩa
- Phải gồm đủ số từ để hiểu được đối tượng đó là gì

Qui ước:

- Tất cả chữ cái của tên đều viết thường và không dấu
- Khi tên gồm nhiều từ, các từ nối với nhau bằng dấu gạch dưới
- Trừ trường hợp đặc biệt, tên của thuộc tính lấy từ đầu tiên của thực thể hoặc viết tắt của thực thể đó làm tiếp đầu ngữ

### 4.5.- Định nghĩa đối tượng

Tất cả các thực thể và quan hệ cần được định nghĩa dưới hình thức là bảng, như trong Hình 3.

Ví dụ:

Tên	Mô tả
nhan_vien	Là người làm việc cho cơ quan và được cơ quan trả lương
du_an	Là nội dung công việc của cơ quan nhằm đạt đến một mục tiêu xác định cụ thể trong một khoảng thời gian cụ thể.

chi_dinh	Nhân viên trong cơ quan có thể được chỉ định tham gia vào các dự án. Cùng một thời điểm, mỗi một nhân viên chỉ tham gia vào nhiều nhất là 3 dự án. Cùng một thời điểm, mỗi dự án phải có ít nhất 5 người thực hiện.
----------	---

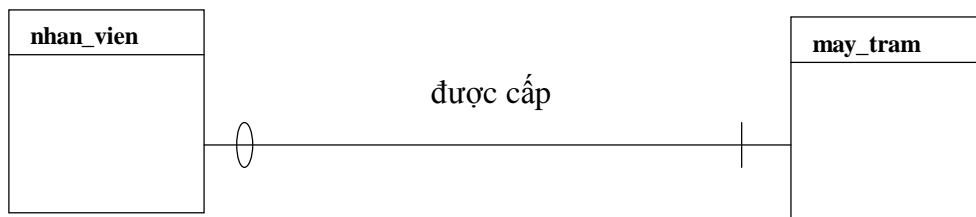
**Hình 3: Mẫu bảng định nghĩa thực thể, quan hệ**

## 5./ Phát triển sơ đồ cơ bản

Khi đã xác định được các thực thể và các quan hệ, bước tiếp theo là vẽ sơ đồ quan hệ thực thể (ERD).

Ví dụ:

Mỗi một nhân viên được cấp một máy trạm để làm việc. Tuy nhiên, không phải máy trạm nào cũng cấp cho nhân viên. Chú ý rằng quan hệ 1:1 rất ít được sử dụng trong thực tế và cách làm thông thường là ghép hai thực thể đó thành một thực thể duy nhất, biến một trong số đó thành thuộc tính.



Hình 4: Quan hệ một - một

(SV ghi chép tại lớp về cách hiểu hình vẽ)

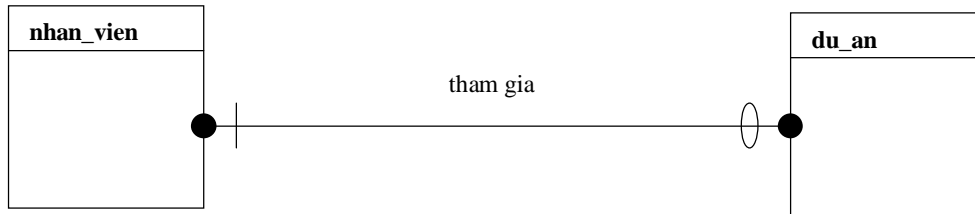
Mỗi một đơn vị chịu trách nhiệm về các dự án do mình đảm trách và mỗi một dự án có đúng một đơn vị chịu trách nhiệm. Trong Hình 5 thực thể don\_vi được gọi là **thực thể mẹ**, và thực thể du\_an là **thực thể con** (đọc theo chiều 1:N).



Hình 5: Quan hệ một - nhiều

(SV ghi chép tại lớp về cách hiểu hình vẽ)

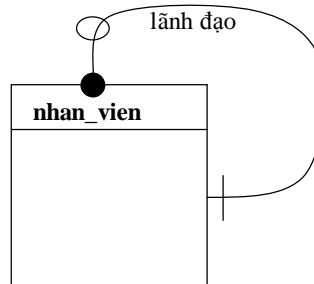
Mỗi một nhân viên có thể tham gia 0, 1 hoặc nhiều dự án, mỗi một dự án phải có ít nhất một nhân viên tham gia. Quan hệ nhiều - nhiều chỉ xuất hiện trong bản thảo đầu tiên, về sau ta phải biến đổi thành các quan hệ 1 - nhiều (xem dưới đây).



Hình 6: Quan hệ nhiều - nhiều

(SV ghi chép tại lớp về cách hiểu hình vẽ)

Một nhân viên có thể phải lãnh đạo một nhóm các nhân viên khác.



Hình 7: Quan hệ đệ qui

## 6./ Tinh lọc sơ đồ

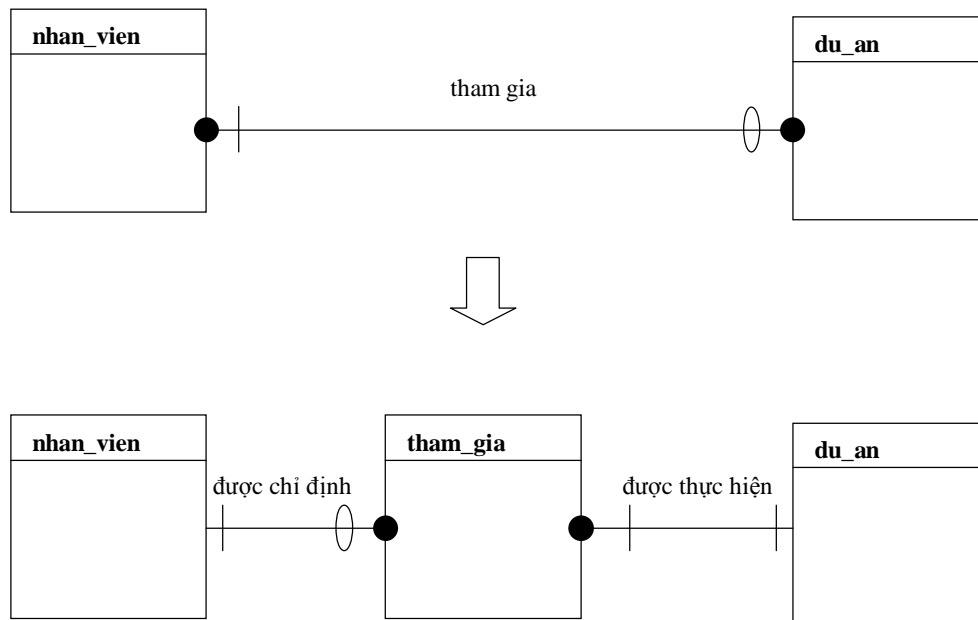
Sau khi đã có lược đồ cơ bản, bước tiếp theo là tinh lọc nó, biến đổi quan hệ M:N về các quan hệ 1:N và giải quyết các quan hệ n-nguyên với  $n > 2$ .

### 6.1.- Tất cả các thực thể phải tham gia vào quan hệ

Trừ trường hợp CSDL chỉ có một bảng, còn lại bất cứ thực thể nào cũng phải tham gia vào các quan hệ. Nếu không thấy thực thể đó có bất cứ mối quan hệ nào với thực thể khác thì ta phải xem xét lại sự tồn tại của nó.

## 6.2.- Giải quan hệ M:N

Quan hệ M:N không thể đưa về bảng trong CSDL quan hệ được, do đó phải biến đổi loại quan hệ này thành các quan hệ 1:N.



Hình 8: Biến đổi quan hệ M:N về các quan hệ 1:N

### *Ghi nhớ về tính tồn tại khi biến đổi M:N về 1:N*

Nếu cả 2 phía M và N đều là bắt buộc:



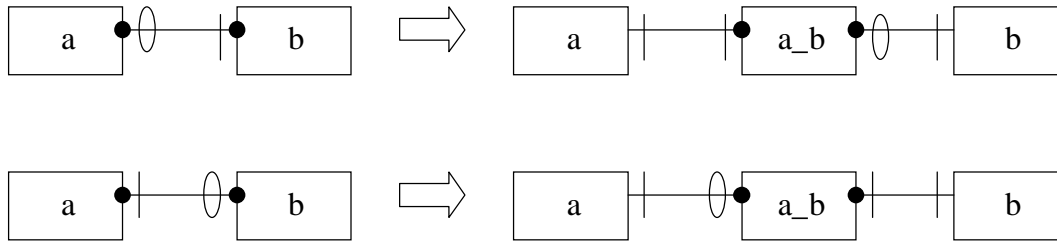
Hình 9: Biến đổi M:N về 1:N nếu cả 2 phía M và N đều bắt buộc

Nếu cả 2 phía M và N đều là tùy chọn:



Hình 10: Biến đổi M:N về 1:N nếu cả 2 phía M và N đều tùy chọn

Nếu một trong 2 phía M và N là bắt buộc và phía còn lại là tùy chọn:



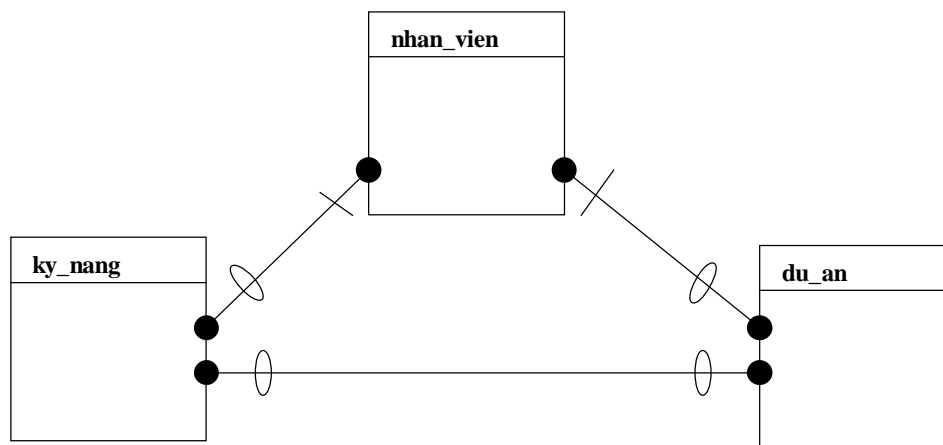
Hình 11: Biến đổi M:N về 1:N khi một trong 2 phía là bắt buộc và phía còn lại là tùy chọn

### 6.3.- Giải các quan hệ phức tạp

Tiếp tục ví dụ trên, giả thiết ta có tình huống như sau.

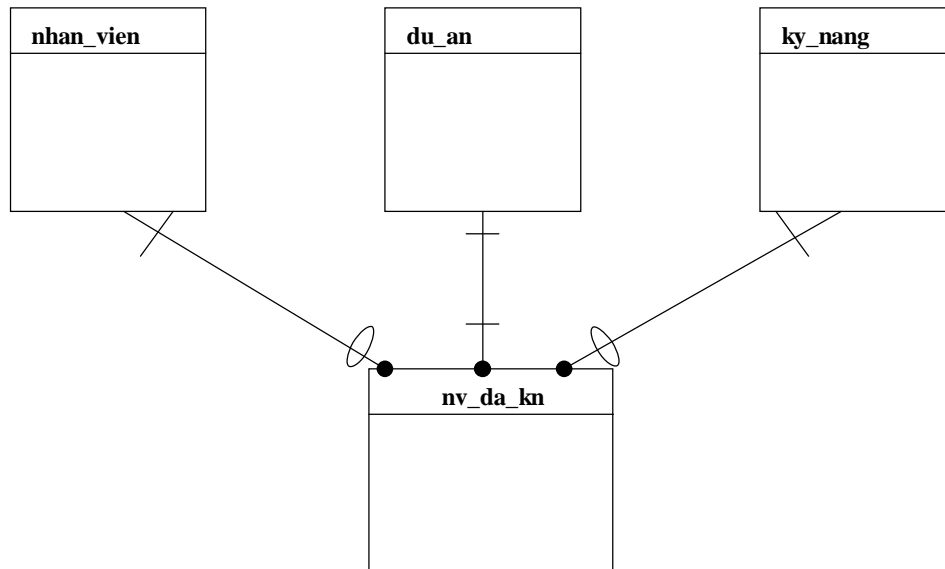
*Người ta nhận thấy mỗi dự án có một số vấn đề cần giải quyết trên cơ sở là người thực thi dự án phải có kỹ năng tương ứng. Do đó, cần có một cơ chế quản lý các kỹ năng và phải quản lý trong mối tương quan là mỗi dự án cần liệt kê ra các kỹ năng cần thiết, và mỗi nhân viên cũng phải có các bằng, chứng chỉ chứng minh mình có các kỹ năng gì trước khi được chỉ định tham gia vào một dự án nào đó.*

Để giải quyết vấn đề trên, người thiết kế CSDL thiết kế thêm thực thể, gọi là thực thể **ky\_nang** và lập các mối quan hệ với các thực thể khác. Chúng được thể hiện như trong Hình 12.



Hình 12: Quan hệ tam nguyên

Vấn đề là quan hệ tam nguyên như trên không biến đổi được về thành các bảng trong mô hình quan hệ. Vì vậy, ngay trong quá trình mô hình hóa này, ta phải biến đổi các mối quan hệ phức tạp về thành các quan hệ 1:N.



Hình 13: Kết quả biến đổi của quan hệ tam nguyên thành các quan hệ nhị nguyên

## 7./ Nhận biết và gán khóa chính, khóa ngoại

Khóa chính và khóa ngoại trong mô hình quan hệ thực thể giống với khóa chính và khóa ngoại trong mô hình quan hệ. Khóa chính và khóa ngoại được lấy ra từ các thuộc tính.

### 7.1.- Nhận biết khóa và khóa chính

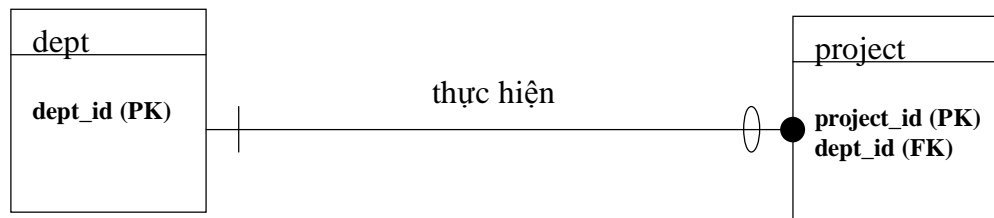
Không có lý thuyết chung về tìm khóa, khóa chính, khóa ngoại từ tập hợp các thuộc tính. Chúng ta chỉ có một vài qui tắc từ kinh nghiệm thực tế:

- Tìm khóa trong số các thuộc tính (hiển nhiên là như vậy)
- Lấy một trong số đó làm khóa chính

- Nếu không tìm thấy một thuộc tính nào làm khóa, hãy thử tìm tổ hợp các thuộc tính làm khóa
- Nếu không tìm thấy thuộc tính nào làm khóa cũng như không tìm thấy tổ hợp nào làm khóa, hãy tạo ra một thuộc tính và lấy thuộc tính đó làm khóa chính
- Nguyên tắc thực tế:
  - các thực thể: chỉ nên lấy một thuộc tính làm khóa;
  - các quan hệ: lấy số thuộc tính bằng số các mũi tên chỉ đến quan hệ đó làm khóa chính (như vậy tổ hợp các khóa ngoại sẽ được làm khóa chính)

## 7.2.- Nhận biết khóa ngoại

- Tất cả các quan hệ phải được thể hiện bằng khóa ngoại
- Khóa ngoại là một thuộc tính nằm ở thực thể con và có cấu trúc giống với khóa chính của thực thể mẹ



Hình 14: Qui ước về thể hiện khóa chính, khóa ngoại trong các thực thể

## 8./ Thêm thuộc tính vào mô hình

Sau khi đã xác định được các thuộc tính khóa, ta thêm các thuộc tính khác vào mô hình, tùy theo nhu cầu của vấn đề. Quá trình này thường gắn với người dùng hệ thống và nghiệp vụ.

Nguyên tắc đầu tiên là thêm các thuộc tính phụ thuộc hàm vào khóa chính. Nếu ta không thấy quan hệ này thì rất có thể thuộc tính đó phải thuộc vào một thực thể khác, và rất có thể đó là thực thể con.

Nếu ta nhận thấy có **thuộc tính đa trị** thì hãy nghĩ đến việc lập ra một thực thể mới. Ví dụ, đối với thực thể **người**, nếu ta xét thuộc tính con cái thì xảy ra vấn đề là: *người đó có thể có nhiều con*. Trong trường hợp này nên suy xét kỹ dưới góc độ ứng dụng. Nếu người dùng cần biết rõ về gia đình, thì rõ ràng việc thêm một



thực thể là chuyện tự nhiên. Và quan hệ giữa hai thực thể này là quan hệ 1:N. Còn nếu người dùng không chú ý lắm đến vấn đề này thì có thể lập một thuộc tính có tên là con.

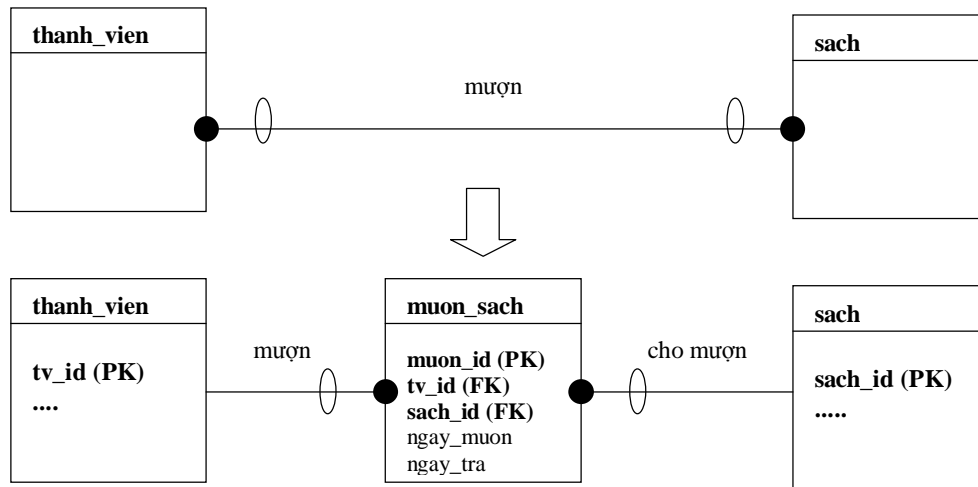
### 8.1.- Thuộc tính của quan hệ

Trong một số trường hợp quan hệ có thể có thuộc tính.

Ví dụ

#### Thành viên mượn sách.

Quan hệ *mượn*, trong trường hợp này có thuộc tính. Đó là: *ngày mượn, thời gian phải trả*. Thực chất, đây là một quan hệ M:N như trong hình vẽ sau:



Hình 15: Quan hệ cũng có thuộc tính và cách biến đổi

**Câu hỏi:** Tại sao không dùng tổ hợp (**tv\_id, sach\_id**) làm khóa cho thực thể **muon\_sach** mà phải dùng **muon\_id**?

### 8.2.- Có nên để thuộc tính vào sơ đồ ERD?

Vấn đề này nhiều chuyên gia có quan điểm trái ngược nhau. Người nói có, người nói không. Có một điểm lưu ý là khi số lượng các thuộc tính quá lớn (từ 30 trở lên) thì việc đưa các thuộc tính vào sơ đồ là không thực tế và làm cho sơ đồ rắc rối, không rõ ràng.

Tác giả soạn tài liệu này ủng hộ việc không đưa tất cả các thuộc tính vào sơ đồ ERD. Tuy nhiên, người lập mô hình có thể (*không bắt buộc*) đưa các thuộc tính *khóa chính* và *khóa ngoại* vào sơ đồ.

## 9./ Các thực thể có cấu trúc phân cấp tổng quát hóa

Đến thời điểm này, có thể coi như ta biết lập một thực thể và các thuộc tính của nó. Ví dụ, đối với một nhân viên ta có các thuộc tính **nhân\_vien\_id**, **ten**, **vi\_tri\_cong\_tac**, và một tập hợp các kỹ năng.

Trong thực tế, có thể xảy ra trường hợp là một tập hợp các thực thể có rất nhiều thuộc tính chung. Ta lấy ví dụ, trong một cơ quan lớn, các đơn vị có thể thực hiện 2 loại dự án, ta tạm gọi là *dự án ngoài* và *dự án nội bộ*. Rõ ràng, hai thực thể **du\_an\_ngoai** và **du\_an\_noi\_bo** có rất nhiều điểm chung như mã dự án, tên dự án, thời gian bắt đầu, thời gian kết thúc, ....

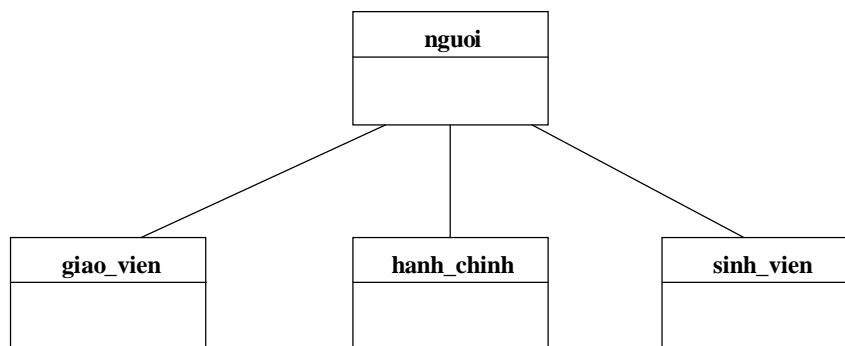
Bên cạnh các điểm chung, ta dễ dàng nhận thấy có một số điểm khác biệt. Ví dụ, dự án ngoài cần xác định khách hàng, chi phí của dự án là bao nhiêu, ...

Như vậy, người lập mô hình vừa đồng thời phải tìm ra các điểm chung vừa phải tìm ra các điểm khác biệt. Quá trình này có tên gọi là quá trình tổng quát hóa các thực thể. Các thực thể thuộc quá trình này có tên gọi là các **thực thể có cấu trúc phân cấp tổng quát hóa**.

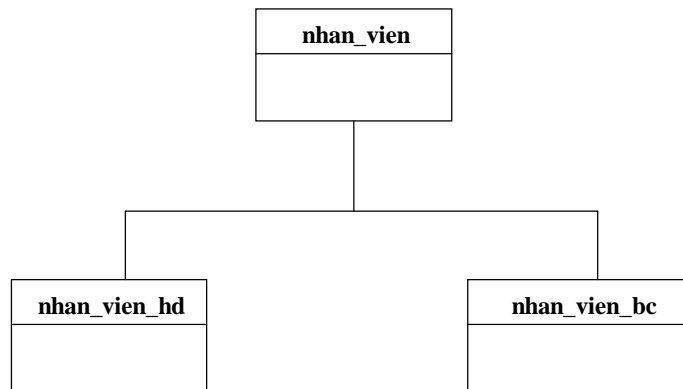
Người ta gộp các thuộc tính chung và lập ra một thực thể và thực thể đwoj phân loại là thực thể **cấp trên (supertype)**. Các thực thể thể hiện sự khác biệt được phân loại là thực thể **cấp dưới (subtype)**.

**Cấp trên** và **cấp dưới** có hai loại phân cấp. Loại phân cấp **phủ nhau** và loại phân cấp **loại trừ nhau**. Loại phân cấp phủ nhau là loại phân cấp như ví dụ trên đây.

Ta lấy ví dụ là *nhân viên biên chế* và *nhân viên hợp đồng*. Các thuộc tính của nhân viên biên chế và nhân viên hợp đồng đều chung và chỉ có một khác biệt duy nhất là *biên chế* hay *hợp đồng*. Người ta gọi các loại thực thể này là thực thể loại trừ nhau.



Hình 16: Thực thể phân cấp phủ nhau



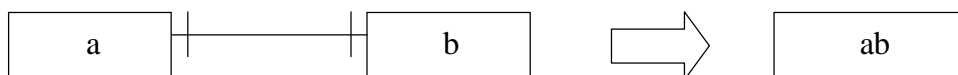
Hình 17: Thực thể phân cấp loại trừ nhau

## 10./ Chuyển đổi sơ đồ ERD thành CSDL quan hệ

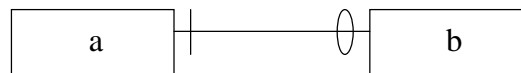
Mục tiêu của phần này là lập qui trình chuyển đổi sơ đồ ERD thành CSDL quan hệ. Phần này chỉ chuyển đổi các quan hệ nhị nguyên. Tác giả không đề cập đến các quan hệ phức tạp hơn. Tác giả cũng giả thiết là các quan hệ M:N đã được biến đổi về 1:N trước khi bước vào quá trình này.

### 10.1.- Chuyển đổi quan hệ 1:1

Nếu quan hệ 1:1 mà tính tồn tại của cả 2 thực thể đều là bắt buộc, thì cách tốt nhất là ghép 2 thực thể thành một.



Như vậy, trường hợp còn lại sẽ là (không mất tính tổng quát):



Cách làm:

- Tên thực thể → tên bảng (có 2 bảng: bảng **a** và bảng **b**)
- Chuyển các thuộc tính trên sơ đồ thành các cột của bảng
- Bảng **a** lấy khóa chính của bảng **b** làm khóa ngoại. Khóa ngoại trong bảng **a** cho phép nhận giá trị NULL.

**Chú ý:** Do hiện nay MySQL không xử lý từ khóa FOREIGN KEY một cách qui chuẩn nên chúng ta sẽ không khai báo FOREIGN KEY trong các ví dụ và

bài tập. Tuy nhiên, chúng ta vẫn khai báo trường khóa ngoại như một trường bình thường và vẫn hiểu theo cơ chế khóa ngoại tham chiếu đến khóa chính của một bảng khác.

## 10.2.- Chuyển đổi quan hệ 1:N



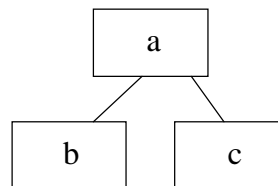
Cách làm:

- α Tên thực thể → tên bảng (có 2 bảng: bảng **a** và bảng **b**)
- α Chuyển các thuộc tính trên sơ đồ thành các cột của bảng
- α Bảng **b** lấy khóa chính của bảng **a** làm khóa ngoại. Khóa ngoại trong bảng **b** cho phép nhận giá trị NULL nếu tính tồn tại của phía **a** là tùy chọn.

**Chú ý:** Do hiện nay MySQL không xử lý từ khóa FOREIGN KEY một cách qui chuẩn nên chúng ta sẽ không khai báo FOREIGN KEY trong các ví dụ và bài tập. Tuy nhiên, chúng ta vẫn khai báo trường khóa ngoại như một trường bình thường và vẫn hiểu theo cơ chế khóa ngoại tham chiếu đến khóa chính của một bảng khác.

## 10.3.- Chuyển đổi các thực thể phân cấp

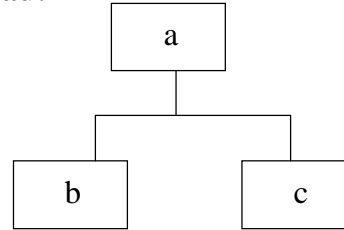
Trường hợp thực thể phân cấp phủ nhau:



Cách làm:

- α Tên thực thể → tên bảng (có n bảng: bảng **a, b, ...**)
- α Chuyển các thuộc tính chung thành các thuộc tính của bảng cấp trên (**a**)
- α Các bảng cấp dưới lấy khóa chính giống với khóa chính của bảng cấp trên
- α Các bảng cấp dưới khai báo khóa ngoại trùng với khóa chính
- α Biến các thuộc tính khác biệt thành các cột của bảng cấp dưới

**Trường hợp thực thể phân cấp loại trừ nhau:**



Cách làm:

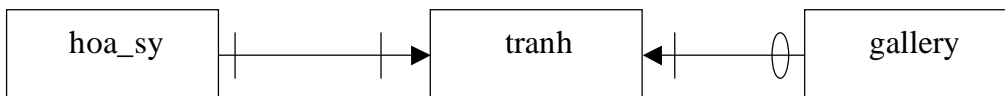
- q Tên thực thể cấp trên → tên bảng (trong ví dụ trên là bảng **a**)
- q Chuyển đổi các thuộc tính của thực thể cấp trên thành các cột của bảng (trong đó bao gồm cả khóa chính)
- q Lập một cột mới có giá trị mã nhằm phân biệt các thực thể cấp dưới. Ví dụ:
  - 1 ⇒ nhân viên thuộc biên chế chính thức
  - 2 ⇒ nhân viên hợp đồng dài hạn
  - 3 ⇒ nhân viên hợp đồng ngắn hạn

### 11./ Ví dụ

#### CSDL họa sỹ

Một họa sỹ vẽ nhiều tranh. Mỗi một bức tranh chỉ do một họa sỹ vẽ. Một bức tranh có thể được đem triển lãm ở một gallery nào đó (cũng có thể không).

ERD:



hoa\_sy (họa sỹ):

STT	Tên thuộc tính	Kiểu dữ liệu	NULL?	Mô tả
1	hoa_sy_id	INT	NOT NULL AUTO_INCREMENT	Trường khóa chính
2	hoa_sy_ten	CHAR(35)	NOT NULL	Tên của họa sỹ
3	hoa_sy_phone	CHAR(15)		Số điện thoại của họa sỹ

gallery (gallery):

STT	Tên thuộc tính	Kiểu dữ liệu	NULL?	Mô tả
1	gallery_id	CHAR(4)	NOT NULL UNIQUE	Trường khóa chính
2	gallery_owner	CHAR(35)		Chủ gallery
3	gallery_address	VARCHAR(250)	NOT NULL	Địa chỉ gallery
4	gallery_phone	VARCHAR(15)	NOT NULL	Điện thoại gallery
5	gallery_rate	DECIMAL(18,2)		Đơn giá thuê

tranh (tranh):

STT	Tên thuộc tính	Kiểu dữ liệu	NULL?	Mô tả
1	tranh_id	CHAR(10)	NOT NULL UNIQUE	Trường khóa chính
2	tranh_de_tua	VARCHAR(250)		Đề tựa của tranh
3	tranh_don_gia	DECIMAL(18,2)		Đơn giá bán tranh
4	hoa_sy_id	INT	NOT NULL	Khóa ngoại tham chiếu hoa_sy(hoa_sy_id)
5	gallery_id	CHAR(4)		Khóa ngoại tham chiếu gallery(gallery_id)

Chuyển đổi thành CSDL:

```
CREATE TABLE hoa_sy (  
    hoa_sy_id INT NOT NULL AUTO_INCREMENT,  
    hoa_sy_ten VARCHAR(35) NOT NULL,  
    hoa_sy_phone VARCHAR(15),  
    PRIMARY KEY(hoa_sy_id)  
);
```

```
CREATE TABLE gallery (  
    gallery_id CHAR(4) NOT NULL UNIQUE,  
    gallery_owner VARCHAR(35),  
    gallery_address VARCHAR(250) NOT NULL,  
    gallery_phone VARCHAR(15) NOT NULL,  
    gallery_rate DECIMAL(18,2),  
    PRIMARY KEY(gallery_id)  
);
```

```
CREATE TABLE tranh (  
    tranh_id CHAR(10) NOT NULL UNIQUE,  
    tranh_de_tua VARCHAR(250),  
    tranh_don_gia DECIMAL(18,2),  
    hoa_sy_id INT NOT NULL,  
    gallery_id CHAR(4),  
    PRIMARY KEY(tranh_id)  
);
```

## 12./ **Bài tập**

*Bài tập 6.1* – Vẽ sơ đồ ERD cho:

Một trường đại học có nhiều khoa. Mỗi khoa có một thầy (cô) chủ nhiệm khoa. Thầy (cô) chủ nhiệm đồng thời cũng thuộc bộ phận hành chính của trường. Thầy (cô) chủ nhiệm khoa đồng thời cũng là giáo viên và giáo viên dạy các lớp học. Giáo viên và bộ phận hành chính đều là các nhân viên của trường (nhân viên chưa chắc là giáo viên hoặc bộ phận hành chính).

*Bài tập 6.2* – Vẽ sơ đồ ERD và chuyển thành bảng trong CSDL:

Mỗi một khoa có nhiều bộ môn và mỗi bộ môn chỉ thuộc một khoa.

*Bài tập 6.3* – Vẽ sơ đồ ERD và chuyển thành bảng trong CSDL:

Mỗi một khoa có nhiều môn học. Mỗi một môn học có thể có nhiều lớp.

*Bài tập 6.4* – Vẽ sơ đồ ERD và chuyển thành bảng trong CSDL:

Mỗi bộ môn có nhiều giáo viên. Một trong số họ là trưởng bộ môn.

*Bài tập 6.5*– Vẽ sơ đồ ERD và chuyển thành bảng trong CSDL:

Mỗi giáo viên có thể dạy nhiều nhất là 4 lớp. Có một số giáo viên chỉ làm công tác nghiên cứu, không dạy.

*Bài tập 6.6*– Vẽ sơ đồ ERD và chuyển thành bảng trong CSDL:

Mỗi sinh viên có thể ghi danh học một học kỳ nhiều nhất là 6 lớp. Mỗi một lớp tối đa 35 sinh viên.

*Bài tập 6.7*– Vẽ sơ đồ ERD và chuyển thành bảng trong CSDL:

Mỗi bộ môn có nhiều chuyên ngành cho sinh viên. Sinh viên chỉ được chọn một chuyên ngành duy nhất và phải thuộc vào một bộ môn nào đó.

*Bài tập 6.8*– Vẽ sơ đồ ERD và chuyển thành bảng trong CSDL:

Mỗi sinh viên có một giáo viên hướng dẫn và một giáo viên có thể hướng dẫn nhiều sinh viên. Không phải giáo viên nào cũng hướng dẫn sinh viên.

*Bài tập 6.9*– Vẽ sơ đồ ERD tổng hợp từ bài tập 6.2 đến 6.8