
Cơ sở dữ liệu - COM2012

I. Giới thiệu về cơ sở dữ liệu

1. **Tại sao phải học môn cơ sở dữ liệu?** Bởi vì nó là một kỹ năng quan trọng và không thể thiếu trong lĩnh vực công nghệ thông tin. Giúp bạn phát triển ra các sản phẩm động có khả năng lưu trữ các thông tin và cuối cùng là khi đi ứng tuyển tại các vị trí công việc sau này.
2. **Lưu trữ dữ liệu dưới dạng hệ thống tập tin?** Lưu trữ thông tin dưới dạng file và mỗi file sẽ đặt nó trong những folder khác nhau nhằm mục đích dễ dàng tìm đến.
 - a. Ưu điểm:
 - Với việc lưu trữ này thì rõ ràng khi ta cần là lưu ngay nên tốc độ triển khai sẽ nhanh
 - Rõ ràng và trực quan với người không có chuyên môn công nghệ thông tin
 - b. Khuyết điểm:
 - Dữ liệu không nhất quán, đôi khi có trường hợp nhiều người sử dụng một file và sẽ xảy ra hiện tượng lưu đè.
 - Trùng lặp dữ liệu nhiều
 - c. Tính chia sẻ dữ liệu không cao (kém)
3. **Cơ sở dữ liệu là gì?** Cơ sở dữ liệu là một hệ thống các thông tin có cấu trúc, được lưu trữ trên các thiết bị lưu trữ nhằm thỏa mãn yêu cầu khai thác thông tin đồng thời của nhiều người sử dụng hay nhiều chương trình ứng dụng chạy cùng một lúc với những mục đích khác nhau.
 - a. Việc sử dụng hệ thống CSDL này sẽ khắc phục được những khuyết điểm của cách lưu trữ dưới dạng hệ thống tập tin:
 - Giảm trùng lặp thông tin ở mức thấp nhất, đảm bảo tính nhất quán và toàn vẹn dữ liệu
 - Đảm bảo dữ liệu được truy xuất theo nhiều cách khác nhau, từ nhiều người khác nhau và nhiều ứng dụng khác nhau.
 - Tăng khả năng chia sẻ thông tin. Ví dụ nếu ta đặt hệ thống dữ liệu online tại Server thì ở bất cứ đâu nếu có password logi vào thì ta hoàn toàn có thể vào hệ thống để truy vấn thông tin
 - b. Tuy nhiên việc sử dụng hệ quản trị CSDL lại có những phiền hà không hề nhỏ sau đây:
 - Phải đảm bảo tính chủ quyền của dữ liệu, vì khi sử dụng có tính chất chia sẻ cao
 - Bảo mật quyền khai thác thông tin
 - Bảo đảm vấn đề tranh chấp dữ liệu khi xảy ra
 - Khi gặp các trục trặc sự cố thì phải bảo đảm vấn đề an toàn dữ liệu, không bị mất dữ liệu
4. **Các hệ cơ sở dữ liệu phổ thông hiện nay?** có rất nhiều hệ quản trị CSDL như **MYSQL, SQL SERVER, Oracle, MS Access.**
5. **Tại sao lại chọn hệ CSDL MySQL vào môn học này?** Mysql là một hệ quản trị CSDL được sử dụng rất rộng rãi trên thế giới, đặc biệt là các ứng dụng website và nó thường đi kết hợp với ngôn ngữ lập trình **PHP** để xây dựng các ứng dụng website. Các hệ thống web ưa chuộng **MYSQL** là tại vì tốc độ xử lý của nó cao, tính dễ sử dụng và thương thích với các hệ điều hành

thông dụng hiện nay như **Linux, Window, ... Mysql** có sử dụng ngôn ngữ truy vấn **T-SQL** để thao tác dữ liệu.

6. Hiện nay đang đình đám các hệ quản trị CSDL với tên gọi **NO-SQL**, nghĩa là hệ quản trị không sử dụng ngôn ngữ có cấu trúc truy vấn (**T-SQL**), vì thế trong tương lai không biết **MYSQL** liệu có bị tiêu diệt hay không?

II. Các bước để xây dựng được một cơ sở dữ liệu:

1. Thiết kế một CSDL được phân thành các mức khác nhau:



a. **Thiết kế các thành phần dữ liệu mức khái niệm** là sự trừu tượng hóa của thế giới thực. **Sơ đồ thực thể - liên kết (ER)** dùng để mô tả lược đồ CSDL mức khái niệm. **Sơ đồ thực thể - liên kết (ERD)** sẽ được đề cập trong những bài tiếp theo. Để thực hiện được thiết kế mức khái niệm cần phải **đặc tả và phân tích yêu cầu** của bài toán dưới đây là cách làm và sử dụng **tài liệu đặc tả và phân tích yêu cầu**:

- **Mục đích:**
 - Nghe các yêu cầu và giải thích từ phía khách hàng – để ký được hợp đồng cần phải đầy đủ và chi tiết.
 - Làm tư liệu đầu vào cho thiết kế và triển khai **cần đầy đủ - chính xác- không mâu thuẫn**
- **Yêu cầu người sử dụng:**
 - Đơn giản – dễ hiểu
 - Diễn đạt bằng ngôn ngữ tự nhiên và sơ đồ về dịch vụ hệ thống cần cung cấp và các ràng buộc trong hoạt động của nó.
 - Dành cho người phát triển hệ thống
- **Yêu cầu hệ thống:**
 - Mô tả đủ chi tiết về các dịch vụ hệ thống cung cấp
 - Các đặc trưng hệ thống cần có
- **Đặc tả yêu cầu:**
 - Là các phát biểu chính thức về hệ thống cần xây dựng.
 - Đủ chi tiết làm cơ sở cho việc thiết kế và triển khai.
 - Dành cho nhà phát triển
- **Người thiết kế cơ sở dữ liệu cần**
 - Nghiên cứu thực trạng.
 - Ghi nhận những mong muốn của người sử dụng.
 - Phân tích những vấn đề có thể và những ứng dụng đang tồn tại.
 - Hiểu về các yêu cầu dữ liệu.
- **Kết quả là:**
 - Một tập hợp ghi chép súc tích về các yêu cầu của những người sử dụng.
 - Những yêu cầu sẽ được đặc tả càng đầy đủ và chi tiết càng tốt
- **Đặc tả yêu cầu chức năng:**

- Thực hiện song song với việc đặc tả các yêu cầu dữ liệu
- Các chức năng phục vụ hệ thống.
- Là các thao tác do người sử dụng định nghĩa sẽ được áp dụng đối với cơ sở dữ liệu.
- **Một số kỹ thuật phân tích yêu cầu:**
 - **Kỹ thuật phỏng vấn:**
 - Lựa chọn đúng đối tượng phỏng vấn: lãnh đạo, phòng ban, các bộ phận, người tham gia trực tiếp.
 - Tìm hiểu bức tranh tổng thể trong từng lĩnh vực từng bộ phận. Cần nắm rõ quy trình nghiệp vụ, mô tả toàn bộ các công việc phải thực hiện, quy tắc xử lý.
 - **Kỹ thuật sử dụng phiếu khảo sát**
 - **Kỹ thuật phân tích tài liệu**
- **Sau khi chúng ta thực hiện hết các bước trên sẽ có 1 tài liệu thiết kế cơ sở dữ liệu mức khái niệm:**
 - **Lược đồ quan niệm** là một mô tả súc tích về các yêu cầu dữ liệu của những người sử dụng.
 - Bao gồm các mô tả chi tiết của các kiểu thực thể, kiểu liên kết và các ràng buộc.
 - Mô hình dữ liệu mức quan niệm dễ hiểu và có thể sử dụng chúng để kết nối giữa người thiết kế CSDL với người dùng cuối trong quá trình phân tích CSDL.
 - Thường sử dụng mô hình liên kết thực thể.
- **Lựa chọn hệ quản trị cơ sở dữ liệu:**
 - Hầu hết các hệ quản trị cơ sở dữ liệu sử dụng một mô hình dữ liệu cài đặt.
 - Lựa chọn hệ quản trị cơ sở dữ liệu cụ thể để đưa ra bản thiết kế có thể cài đặt được.

b. Thiết kế các thành phần dữ liệu mức logic

- **Thiết kế CSDL mức logic** là quá trình chuyển CSDL mức khái niệm sang mô hình **Lược đồ quan hệ** và **chuẩn hóa các quan hệ**.
- Các khái niệm **Lược đồ quan hệ** và **chuẩn hóa** sẽ được đề cập trong các bài sau.
- Đưa ra mô hình dữ liệu logic đáp ứng được yêu cầu tối thiểu hóa sự dư thừa thông tin và đáp ứng được việc cài đặt.
- Kết quả của bước này là một lược đồ cơ sở dữ liệu dưới dạng một mô hình dữ liệu cài đặt của hệ quản trị cơ sở dữ liệu.

c. Thiết kế các thành phần dữ liệu mức vật lý

- Mức thấp nhất của kiến trúc một CSDL là **cơ sở dữ liệu vật lý**. CSDL vật lý là sự cài đặt cụ thể của CSDL mức khái niệm.
- **CSDL vật lý bao gồm** các **Bảng (Table)** và **mối quan hệ (Relationship)** giữa các bảng này.

2. Khái niệm thực thể

- a. **Định nghĩa:** Thực thể là một đối tượng, một địa điểm, con người... trong thế giới thực được lưu trữ thông tin trong CSDL. Hoặc Là tập hợp các thực thể có cùng

thuộc tính. Mỗi loại thực thể đều phải được đặt tên sao cho có ý nghĩa. Kí hiệu: hình chữ nhật

Sinh Viên

- Tập hợp các thực thể giống nhau tạo thành 1 tập thực thể (Nhớ lại khái niệm về lập trình OOP về CLASS)

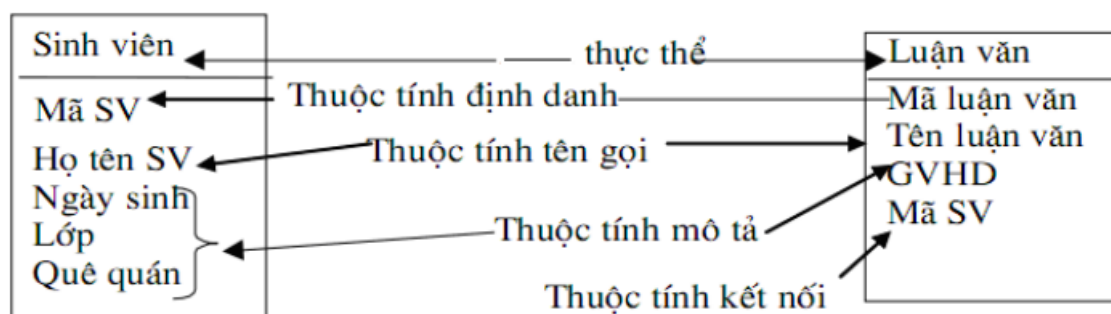
b. Ví dụ: Một sinh viên được gọi một thực thể, **tập hợp (nhiều) các sinh viên là một tập thực thể.**

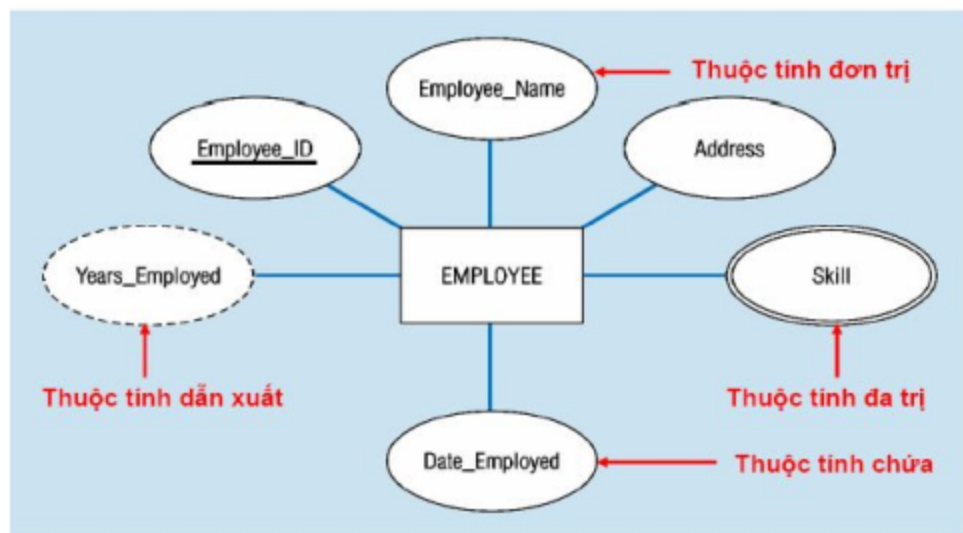
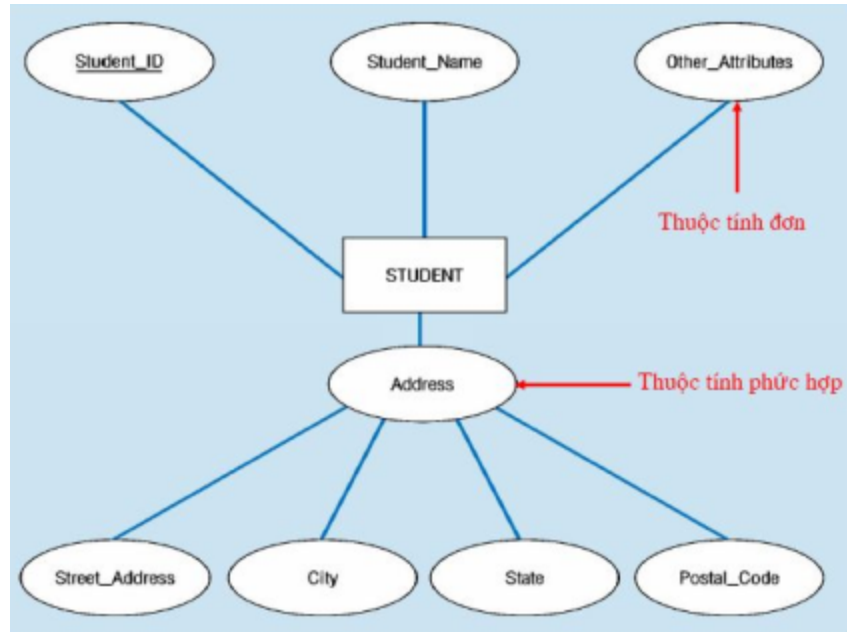
3. Thuộc tính:

- Định nghĩa:** Là những đặc tính riêng biệt của tập thực thể. Là tính chất của thực thể cần được quản lý. Chỉ quan tâm tới những tính chất có liên quan tới ứng dụng.
- Thuộc tính định danh (thuộc tính khóa) (PK khóa chính):** là một hoặc một số tối thiểu các thuộc tính của một tập thực thể mà giá trị của nó cho phép phân biệt các thực thể khác nhau trong tập thực thể. **Kí hiệu: tên thuộc tính có gạch chân**
- Thuộc tính mô tả:** Là các thuộc tính mà giá trị của chúng chỉ có tính mô tả cho thực thể hay liên kết mà thôi.
- Thuộc tính tên gọi:** là thuộc tính mô tả để chỉ tên các đối tượng thuộc thực thể. Thuộc tính tên gọi để xác định các thực thể.
- Thuộc tính đơn:** là thuộc tính không thể phân chia ra được thành các thành phần nhỏ hơn.
- Thuộc tính phức hợp:** là thuộc tính có thể phân chia được thành các thành phần nhỏ hơn, biểu diễn các thuộc tính cơ bản hơn với các ý nghĩa độc lập.

• Các loại giá trị của thuộc tính:

- **Thuộc tính Đơn trị:** các thuộc tính có giá trị duy nhất cho một thực thể (VD: số CMND, ...)
- **Thuộc tính Đa trị:** các thuộc tính có một tập giá trị cho cùng một thực thể (VD: bằng cấp, ...)
- **Suy diễn được** (năm sinh <----> tuổi)

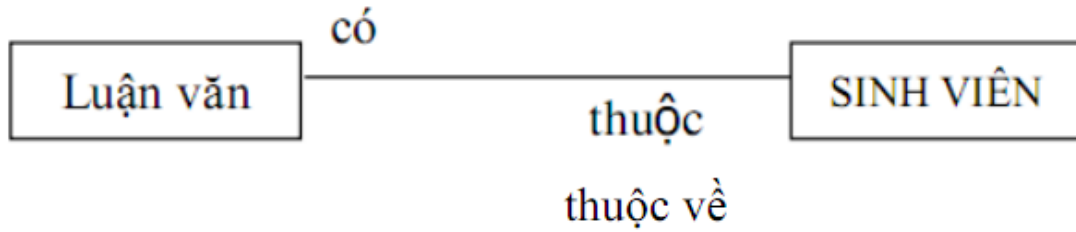




4. Mối quan hệ (Relationship)

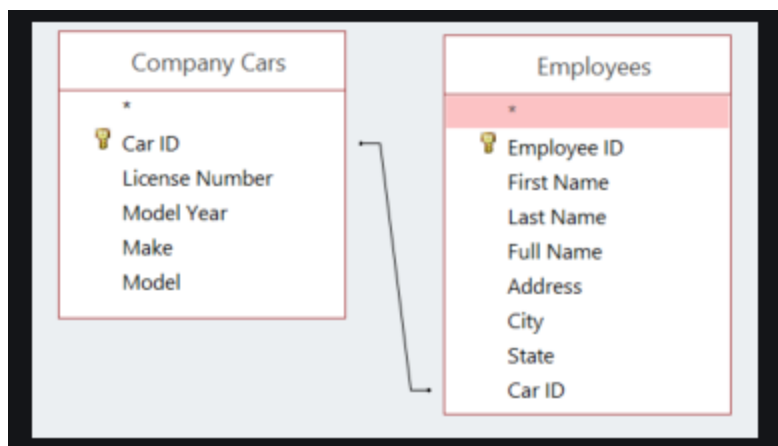
- a. **Liên kết (còn gọi là quan hệ)** là sự kết hợp giữa hai hay nhiều thực thể phản ánh sự ràng buộc trong quản lý. **Đặc biệt: Một thực thể có thể liên kết với chính nó ta thường gọi là tự liên kết.** Giữa hai thực thể có thể có nhiều hơn một liên kết. Có **ba kiểu** liên kết: **một - một, một - nhiều, nhiều - nhiều.**
- b. **Liên kết 1 – 1:** Mỗi thể hiện của thực thể A quan hệ với một thể hiện của thực thể B và ngược lại. Quan hệ 1-1 là quan hệ giữa hai tập thực thể trong đó mỗi thực thể của tập này chỉ có thể liên kết với nhiều nhất một thực thể của tập kia, và ngược lại.

- Ví dụ: Một sinh viên có một luận văn. Một luận văn thuộc về một sinh viên.



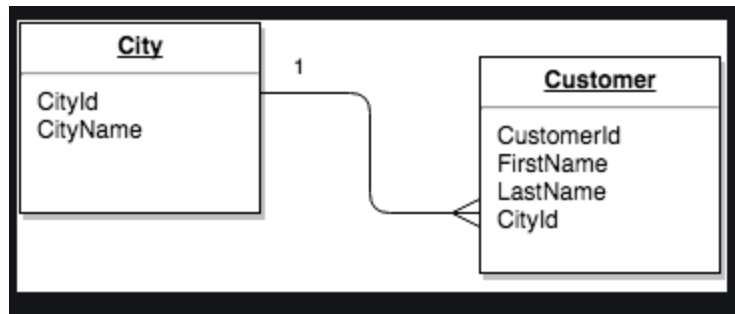
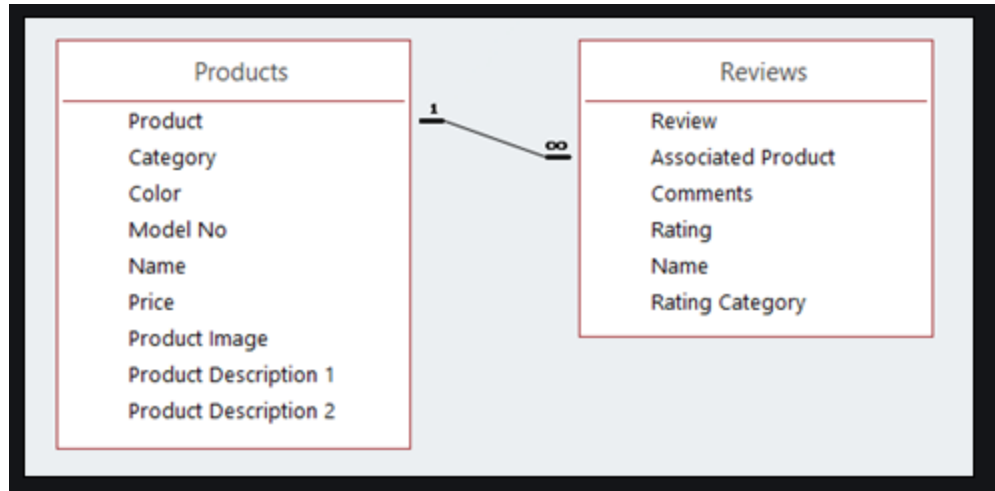
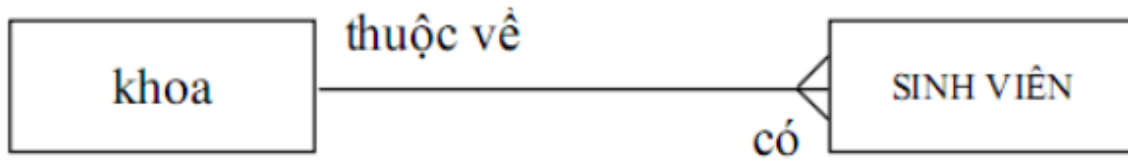
| People | | Drivers Licenses | |
|--------|-------|------------------|--------|
| ID | Name | UserID | Number |
| 1 | Alice | 1 | F25532 |
| 2 | Bob | 2 | S43212 |
| 3 | Cathy | 3 | B98364 |

One-to-One

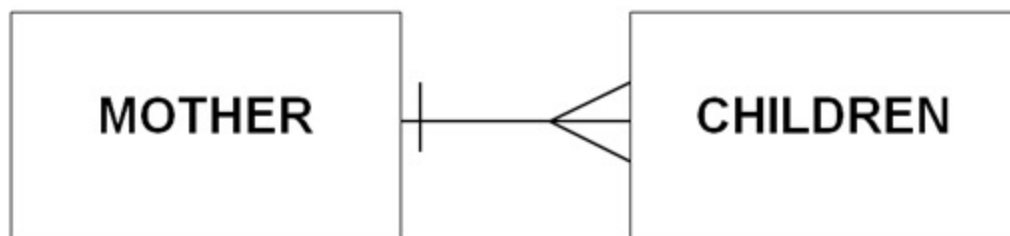


c. **Liên kết một nhiều (1 - N):** Quan hệ 1-N là quan hệ giữa hai tập thực thể trong đó mỗi thực thể của tập này có thể liên kết với nhiều thực thể của tập còn lại.

- **Ví dụ:** Một khoa có nhiều sinh viên. Một sinh viên thuộc về một khoa.



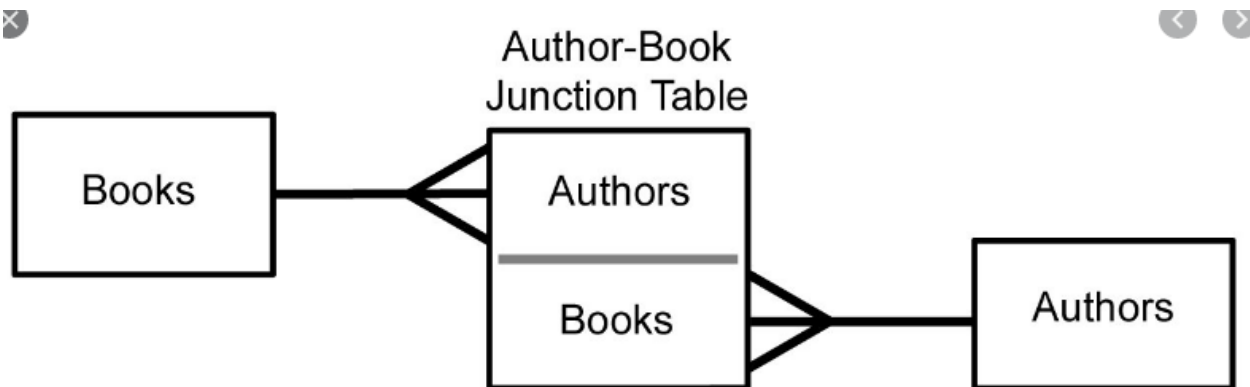
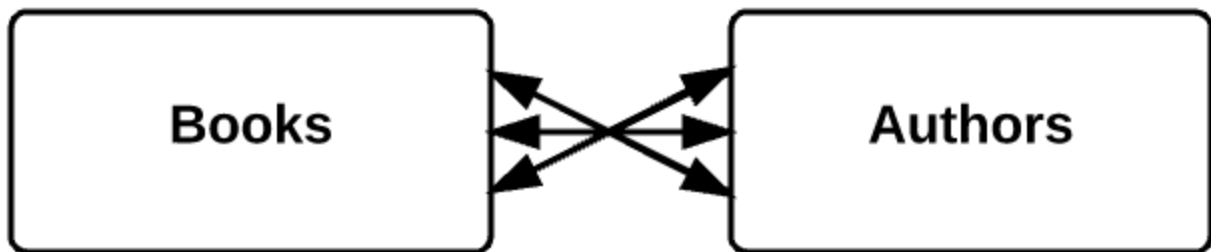
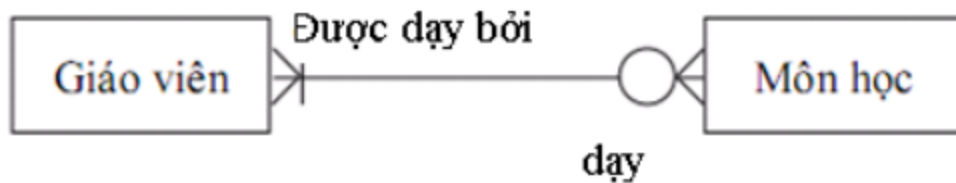
One-to-many (or many-to-one) relationships



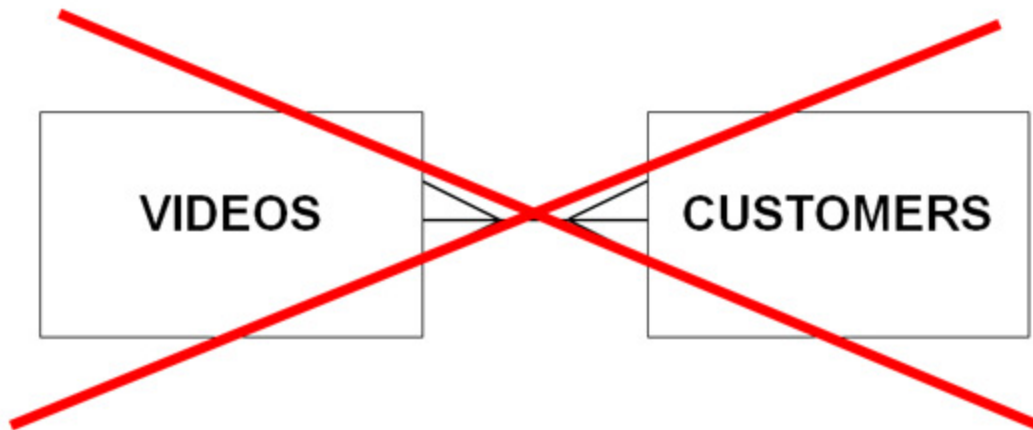
d. **Liên kết nhiều nhiều (N - N):** Quan hệ **N-N** là quan hệ giữa hai tập thực thể trong đó một thực thể của tập này có thể liên kết với 0, 1 hoặc nhiều thực thể của tập kia, và ngược lại. Thường quan hệ **N-N** có thêm phần dữ liệu giao nhau để thêm thông tin cụ thể cho mỗi quan hệ. **Hoặc** Điều kiện để một cá thể của thực thể tham gia vào liên kết với một thực thể khác gọi là loại thành viên. Nó có thể là bắt buộc hay tùy chọn trong quan hệ. Các loại thành viên cho biết số thể hiện nhỏ nhất của mỗi thực thể tham gia vào liên kết với một thể hiện của một thực thể khác.

- **Kí hiệu, Ví dụ:**

- **Tùy chọn (ít nhất 0)** – “một giáo viên có thể dạy không, một hoặc nhiều môn học.”
- **Bắt buộc (ít nhất 1)** – “một môn học cần phải được một hoặc nhiều giáo viên dạy.”



Many-to-many relationships



Many-to-many relationships are not allowed when designing a database.

You must find a way of making them 'many-to-one' or 'one-to-many'

- e. **Lưu ý:** Mô hình dữ liệu không chỉ là công cụ phân tích thiết kế mà còn như một phương pháp kiểm tra chặt chẽ các yêu cầu nghiệp vụ của người sử dụng.
- f. **Nếu hai thực thể có quan hệ một - một** thường có ít lý do để coi chúng như hai bảng tách biệt nên người ta thường gộp hai thực thể làm một
- g. **Nếu hai thực thể có quan hệ nhiều - nhiều** thì không có sự khác biệt về bản chất giữa các chiều vì vậy ít khi được sử dụng.
- h. **Tóm lại trong ba kiểu liên kết trên, liên kết một nhiều là quan trọng hơn cả và hầu như các mối quan hệ trong mô hình thực thể liên kết đều là một nhiều.**

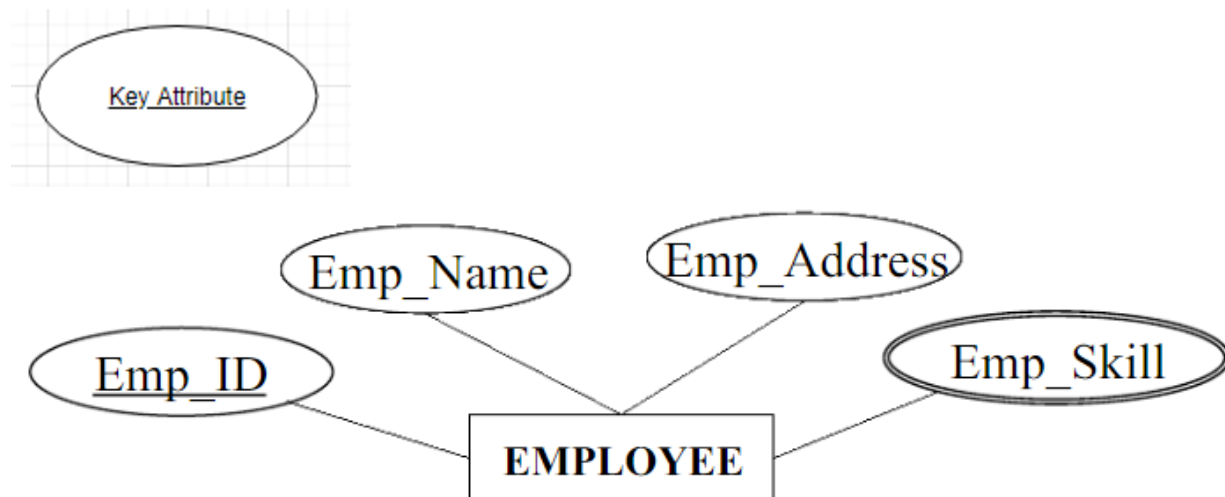
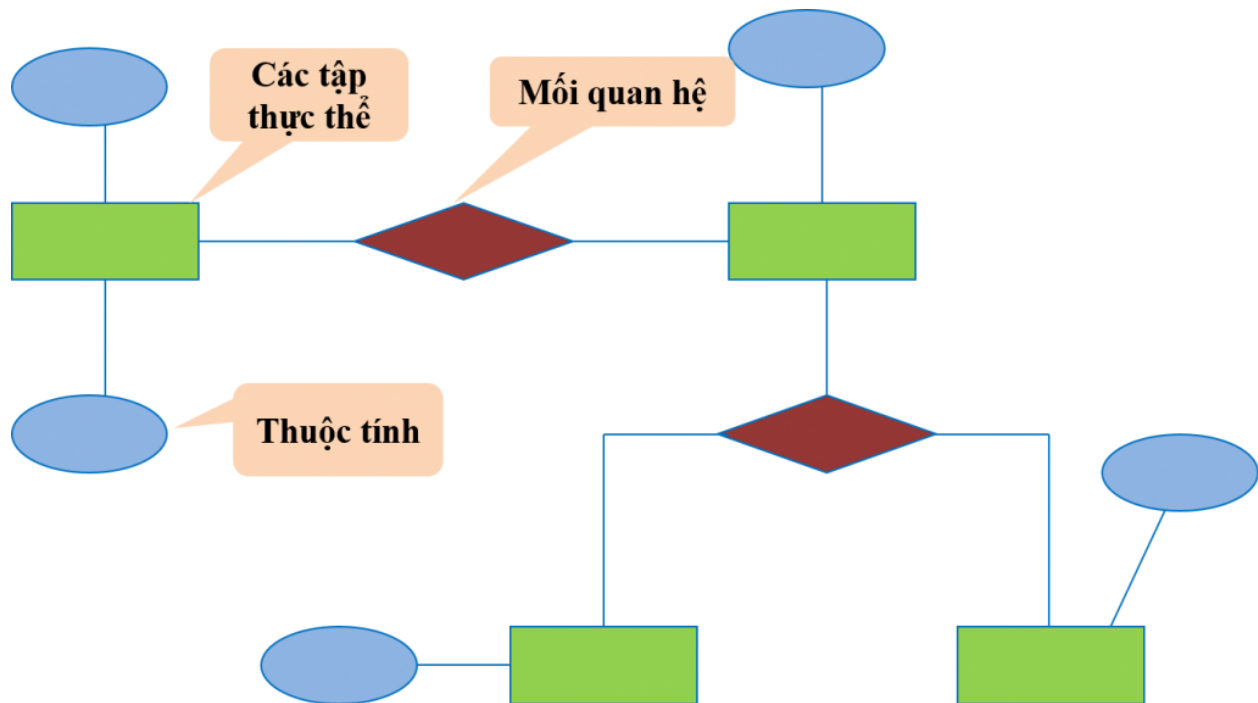
III. Xây dựng một database:

1. Thực thể và tập thực thể (Entity)

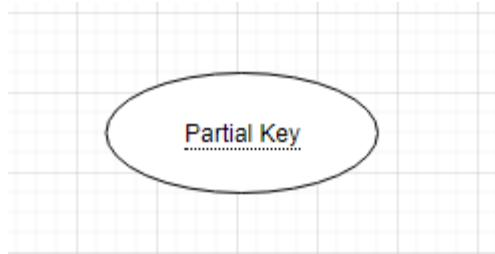
- a) Thực thể là một đối tượng trong thế giới thực. (**Entity PHẢI LUÔN là danh từ nhé – Và có một thực thể không tồn tại ở business thực tế ở bên ngoài. Nó là những thực thể trung gian nằm giữa 2 thực thể khác và thể hiện mối quan hệ nhiều (Chính là bảng tạm – bảng trung gian)**)
- b) Một nhóm bao gồm các thực thể tương tự nhau tạo thành một tập thực thể
- c) Việc lựa chọn các tập thực thể là một bước vô cùng quan trọng trong việc xây dựng sơ đồ về mối quan hệ thực thể
 - **Thực thể mạnh và thực thể yếu** (Kiểu thực thể yếu là kiểu thực thể tồn tại phụ thuộc vào thực thể khác (**thực thể làm chủ hay còn gọi là xác định nó**). Kiểu thực thể yếu không có khoá.)
- d) **Xác định được thực thể thông qua việc đọc và phân tích tài liệu.**

2. Xác định thuộc tính (Attribute)

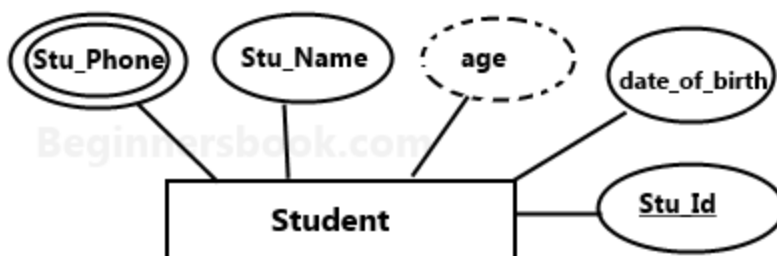
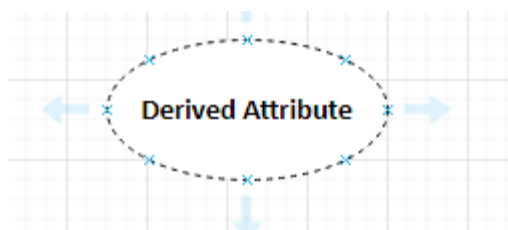
- a) **Attribute** nghĩa là thuộc tính. Thuộc tính của chính những thực thể đã được xác định ở bước trên. **(Là những thông tin riêng biệt của đối tượng đó mà mình sẽ lưu trữ. Khi xác định thuộc tính phân tích các kiểu dữ liệu của thuộc tính đó như là kiểu chuỗi, kiểu số nguyên, real)**
 - Ví dụ: **Khách hàng** (Họ và tên – Email – Ngày Sinh Nhật – Số Điện thoại – Sở thích)
 - b) **Relational Database** được tổ chức thành nhiều bảng, và giữa các bảng quan hệ với nhau bằng các **khóa**.
 - c) **Cột** chính là thuộc tính (**attribute**) của đối tượng Customer, bao gồm cột: Last Name, First Name, Email.....
 - d) **Dòng (Row)** là các “**bản ghi**”, mình hay gọi là các **record**, là số lượng dữ liệu mà table đó lưu trữ trong **database**.
 - e) Cần phân biệt các **record** với nhau ==> **khóa chính (Primary Key)** ra đời. (Bản chất thì khóa chính cũng chỉ là một cột)
 - **Khóa chính là thứ tồn tại độc lập duy nhất (và không được giống nhau). Tất cả những cột khác, giống nhau hay không, không quan trọng.**
 - f) Vì các table liên kết với nhau. Nhưng để liên kết với nhau thì nó cần có điểm chung nào đó. **Foreign Key** ra đời. Nó là **key** dùng để liên kết 2 tables lại với nhau.
 - g) **Foreign Key** sẽ là **Primary Key** ở một **table**, nhưng lại là **Foreign Key** ở một **table** khác mà **table** đó liên kết đến.
3. Mô hình quan hệ - thực thể (**Entity – Relationship Model**)
- a) **Mục tiêu** của mô hình **E-R** trong quá trình thiết kế cơ sở dữ liệu đó là phân tích dữ liệu, xác định các đơn vị thông tin cơ bản cần thiết của tổ chức, mô tả cấu trúc và mối liên hệ giữa chúng. **E-R** là mô hình trung gian để chuyển những yêu cầu quản lý dữ liệu trong thế giới thực thành mô hình cơ sở dữ liệu quan hệ.
 - b) **Các thành phần cơ bản của mô hình E-R** bao gồm các thực thể và thuộc tính của nó.
 - c) **Các loại hình sử dụng trong mô hình ER (Chỉ ra mối quan hệ giữa các thực thể hoặc một thực thể quan hệ với chính nó)**



Nếu chữ trong hình Oval được **gạch dưới**, nghĩa là với attribute này, mỗi dòng dữ liệu chỉ có một giá trị duy nhất. Thường nó sẽ là ID của bảng dữ liệu đó. Nó còn gọi là **Unique key, unique attribute, Primary Key**, hay **Identifier**. Chú ý là mỗi Entity có thể có nhiều **Unique key**. Khi đó ta chỉ cần chọn một trong số đó làm Primary Key. Ví dụ trong trường dữ liệu Sinh Viên, ta có thể có Mã số sinh viên và Số CMND đều chỉ có giá trị duy nhất cho từng sinh viên nhưng ta chỉ chọn một trong hai làm Primary Key.

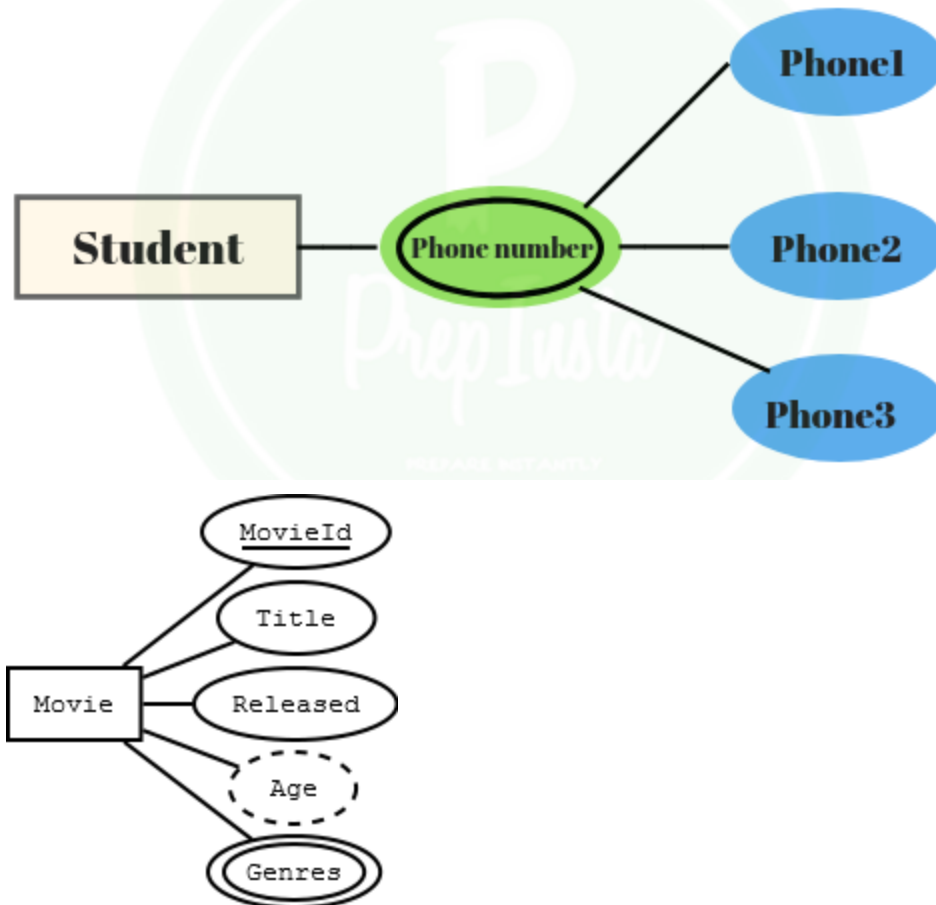
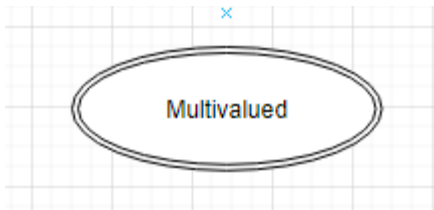


Nhưng nếu chữ trong hình Ovan **được gạch dưới đứt quãng** và **hình chữ nhật có 2 viền**, điều này nghĩa là **Weak Key Attribute** đó là **Partial Key**. Tức là một mình nó không thể dùng để xác định được Entity vì nó phụ thuộc vào Key (unique attribute) của một Entity khác. Ta phải dùng Unique Key của một Entity khác cùng với Partial Key của Entity này mới truy cập được đúng dữ liệu mình muốn.

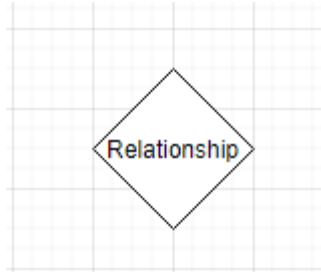


Nếu hình Ovan (Attribute) có viền **gạch đứt quãng**, nó nghĩa là **Derived (Thuộc tính dẫn xuất)**. Nghĩa là ta không cần thu thập data cho nó mà có thể suy ra từ một **Attribute** khác trong **Database**. Ví dụ dựa vào ngày sinh, ta có thể suy ra người đó bao nhiêu tuổi. (Thuộc tính dẫn xuất không cần phải lưu trữ vật lý trong cơ sở dữ liệu – thay vào đó nó có thể được bắt nguồn bằng cách sử dụng một thuật toán:

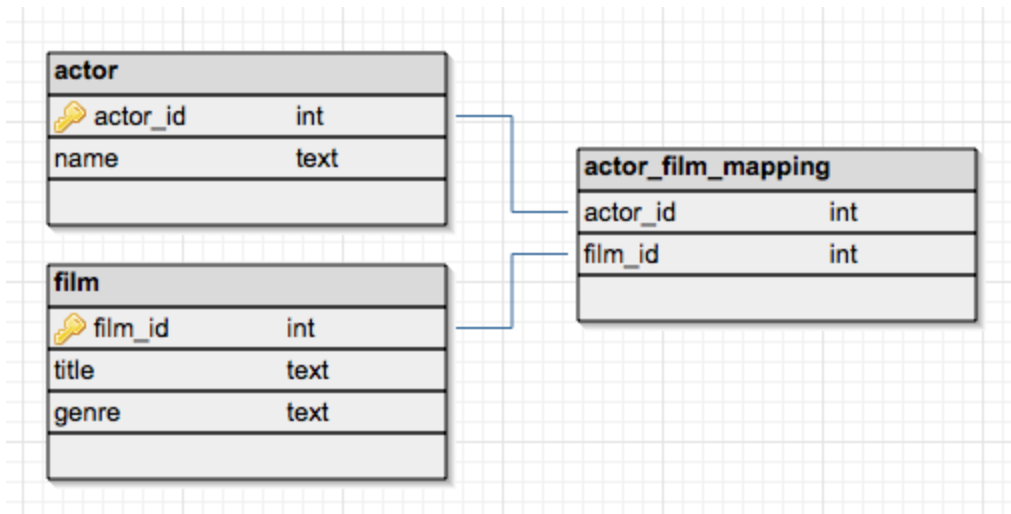
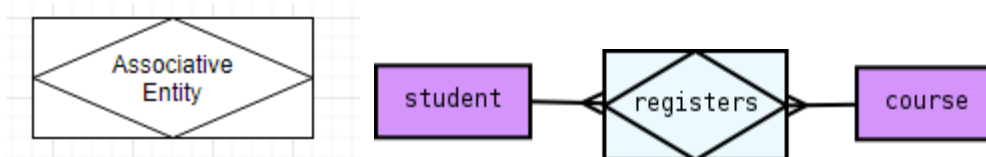
Ví dụ: Tuổi của nhân viên có thể được tìm thấy bằng cách tính toán chênh lệch giữa ngày hiện tại và ngày của quá khứ.)



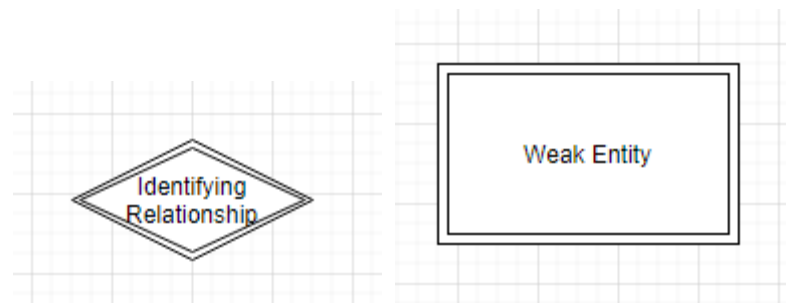
Nếu hình **Ovan (Attribute)** có **2 viền** đây là một Attribute có nhiều giá trị khác nhau cho cùng một Entity. Ví dụ attribute có tên Kỹ Năng Nhân Viên. Mỗi Nhân viên sẽ có vài kỹ năng khác nhau chứ không chỉ có một kỹ năng làm việc.



Một thực thể A **quan hệ** với một thực thể B bởi hình **THOI**. Thể hiện mối liên quan giữa hai hay nhiều thực thể. **Mỗi liên kết có một tên gọi và thường dùng động từ.**

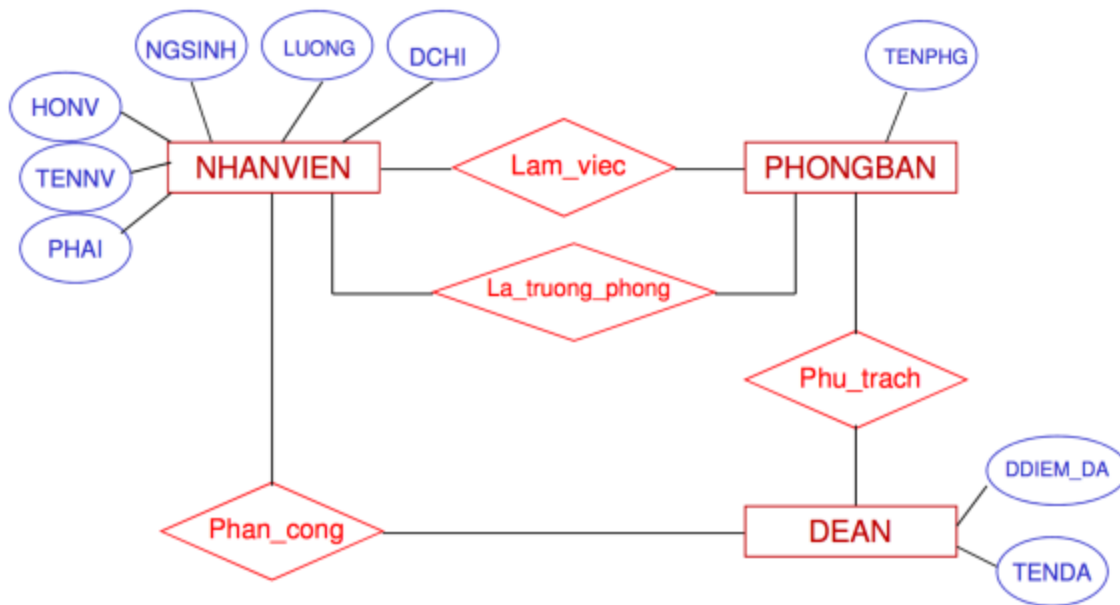


Hình thoi nằm trong hình chữ nhật là Một thực thể kết hợp **associative entity** là một định nghĩa sử dụng trong quan hệ giữa các thực thể để giải quyết một mối **quan hệ nhiều nhiều giữa hai thực thể**. Các thực thể kết hợp được triển khai trong cơ sở dữ liệu bằng cách sử dụng bảng kết hợp. Là bảng mà có thể chứa các tham chiếu đến từ cùng một hoặc các bảng khác nhau trong cơ sở dữ liệu.





Identifying Relationship là hình thoi kép với 2 đường kẻ. Thể hiện một mối quan hệ xác định khi sự tồn tại của một **bảng con** phụ thuộc vào một **bảng cha**. Ví dụ trên: **owns: sở hữu** – Bảng chó bị phụ thuộc vào bảng chủ của chó và không có khóa chính - ở đây ta sử dụng khóa chính của cha để thể hiện bên bảng của chó. Cuối cùng là thể hiện mối quan hệ. Ví dụ 2: Một người có một hoặc nhiều số điện thoại. Nếu họ chỉ có một số điện thoại, chúng ta có thể lưu nó vào một cột của Người. Vì chúng tôi muốn hỗ trợ nhiều số điện thoại, chúng tôi tạo một bảng thứ hai PhoneNumbers, có khóa chính bao gồm person_id tham chiếu bảng Person.



4. Chuẩn hóa cơ sở dữ liệu: (MỨC LOGIC)

- Giai đoạn này là 1 cách tiếp cận từ dưới lên (bottom-up approach) để thiết kế CSDL, bắt đầu từ các mối liên hệ giữa các thuộc tính. Mục đích chuẩn hóa là loại bỏ các bất thường của 1 quan hệ và để có được các quan hệ có cấu trúc tốt hơn, nhỏ hơn. **Giảm thiểu sự dư thừa dữ liệu ở mức thấp nhất và cho phép người dùng thêm, sửa, xóa mà không gây ra mâu thuẫn dữ liệu.** Các dạng chuẩn được sắp xếp theo thứ tự từ

thấp đến cao. Để chuẩn hóa 2NF thì cơ sở dữ liệu của bạn phải đạt chuẩn 1NF, tương tự nếu đạt chuẩn 3NF thì phải đạt chuẩn 1NF và 2NF. Và chuẩn Boyce-Codd sẽ bao gồm 3 loại chuẩn 1NF, 2NF và 3NF. Mục đích: **Giảm sự dư thừa của dữ liệu - đảm bảo tính nhất quán của dữ liệu - hỗ trợ dễ dàng cập nhật, truy xuất dữ liệu.**

| idKhachHang | tenKhachHang | gioiTinhKhachhang | dienThoai | diaChi | idHoaDon | ngayLapHoaDon | idSanPham | moTa | soLuong | donGia | tongTien |
|-------------|-------------------------|-------------------|-----------|------------------|----------|---------------|-----------|-----------|---------|----------|----------|
| 1 | Phùng Văn Hưng | Nam | 912141265 | Hà Nội, Hưng yên | 100 | 21/03/2020 | 15647 | Iphone 11 | 2 | 10.000 đ | 20.000 đ |
| 2 | Nguyễn Thị Thanh Phương | Nữ | 912141266 | HCM | 101 | 22/03/2020 | 15648 | Xe Đạp | 3 | 10.001 đ | 30.003 đ |
| 3 | Vũ Quốc Luật | Nam | 912141267 | Hà Nội | 102 | 19/03/2020 | 15649 | Loa | 1 | 10.002 đ | 10.002 đ |
| 4 | Vũ Quốc Luật | Nam | 912141267 | Hà Nội | 103 | 21/03/2020 | 15650 | Đài | 1 | 10.003 đ | 10.003 đ |
| 5 | Nguyễn Ngọc Quyết | Nam | 912141269 | Nghệ An, Hà Tĩnh | 104 | 19/03/2020 | 15651 | Mì Tôm | 4 | 10.004 đ | 40.016 đ |
| 6 | Trần Văn Duy | Nam | 912141271 | Hòa Bình | 105 | 18/03/2020 | 15652 | Bỉm | 5 | 10.005 đ | 50.025 đ |
| 7 | Trần Văn Duy | Nam | 912141271 | Hòa Bình | 106 | 21/03/2020 | 15653 | Sữa | 1 | 10.006 đ | 10.006 đ |

b) Các dạng chuẩn hóa:

• Dạng chuẩn 1 (1NF – first normal form)

Một bảng được gọi là ở dạng **1NF** nếu miền giá trị của một thuộc tính chỉ chứa giá trị nguyên tố đơn (không phân chia được) và giá trị của mỗi thuộc tính cũng là một giá trị đơn lấy từ miền giá trị của nó. Để bảng đạt chuẩn hóa dạng **1NF**:

- Các thuộc tính của bảng phải là nguyên tố không phải là thuộc tính đa trị tức là sự không thể phân chia một thuộc tính thành các phần nhỏ hơn.
- Giá trị của các thuộc tính trên bảng phải là đơn trị - không chứa nhóm lặp
- Không có một thuộc tính nào có giá trị có thể tính toán được từ một thuộc tính khác.
- Xác định được thuộc tính khóa chính.

| idKhachHang | tenKhachHang | dienThoai | diaChi | idHoaDon | ngayLapHoaDon | idSanPham | moTa | soLuong | donGia | tongTien |
|-------------|-------------------------|-----------|------------------|----------|---------------|-----------|-----------|---------|----------|----------|
| 1 | Phùng Văn Hưng | 912141265 | Hà Nội, Hưng yên | 100 | 21/03/2020 | 15647 | Iphone 11 | 2 | 10.000 đ | 20.000 đ |
| 2 | Nguyễn Thị Thanh Phương | 912141266 | HCM | 101 | 22/03/2020 | 15648 | Xe Đạp | 3 | 10.001 đ | 30.003 đ |
| 3 | Vũ Quốc Luật | 912141267 | Hà Nội | 102 | 19/03/2020 | 15649 | Loa | 1 | 10.002 đ | 10.002 đ |
| 4 | Vũ Quốc Luật | 912141267 | Hà Nội | 103 | 21/03/2020 | 15650 | Đài | 1 | 10.003 đ | 10.003 đ |
| 5 | Nguyễn Ngọc Quyết | 912141269 | Nghệ An, Hà Tĩnh | 104 | 19/03/2020 | 15651 | Mì Tôm | 4 | 10.004 đ | 40.016 đ |
| 6 | Trần Văn Duy | 912141271 | Hòa Bình | 105 | 18/03/2020 | 15652 | Bỉm | 5 | 10.005 đ | 50.025 đ |
| 7 | Trần Văn Duy | 912141271 | Hòa Bình | 106 | 21/03/2020 | 15653 | Sữa | 1 | 10.006 đ | 10.006 đ |

Bảng dữ liệu hóa đơn trên vi phạm điều kiện chuẩn 1NF vì: **Địa chỉ** chứa các giá trị trùng lặp, hơn thế nữa, giá trị **diaChi** trong từng hàng không phải là đơn trị. Thuộc tính **tongtien** có thể tính được bằng cách nhân số lượng*đơn giá không nhất thiết đưa vào bảng, gây ra dư thừa dữ liệu (Thuộc tính dẫn xuất). Vì vậy ta phải tách bảng thành bảng **KhachHang – HoaDon – DiaChi**.

KhachHang(idKhachHang, tenKhachHang, dienthoai).

HoaDon(idHoaDon, ngayLapHoaDon, idSanpham, moTa, soLuong, donGia).

DiaChi(idKhachHang, diachi)

• Dạng chuẩn 2 (2NF – second normal form)

- Phải thỏa mãn chuẩn **1NF**
- Phụ thuộc hàm đầy đủ vào khóa chính
- Với các quan hệ có tính khóa đơn thì không phải xét – chỉ kiểm tra lược đồ có chứa phụ thuộc hàm bộ phận

- Nói ngắn gọn hơn: **Các trường thuộc tính không phải khóa chính, phải phụ thuộc hoàn toàn vào khóa chính.** Không được phép phụ thuộc vào 1 phần của khóa chính.

Phân tích: **moTa**, **donGia** phụ thuộc vào **idSanPham** chứ không phụ thuộc vào cả 2 khóa còn lại như **idKhachHang**, **idHoaDon**. Thuộc tính **tenKhachHang** và **dienthoai** chỉ phụ thuộc vào khóa **idKhachHang**, thuộc tính **ngayLapHoaDon** phụ thuộc vào **idKhachHang** và **idHoaDon**. Thuộc tính **soLuong** phụ thuộc vào **idHoaDon** và **idSanpham**.

Sau khi phân tích để đạt chuẩn 2NF:

- Tách thuộc tính (idSanpham, moTa, donGia) thành một bảng SanPham.
- Tách thuộc tính (idHoaDon, idKhachHang, ngayLapHoaDon) thành một bảng HoaDon.
- Tách thuộc tính (idHoaDon, idSanpham, soLuong) thành một bảng trung gian giữa HoaDon và SanPham và đặt tên là HoaDonSanPham

KhachHang(idKhachHang, tenKhachHang, dienthoai).

DiaChi(idKhachHang, diachi)

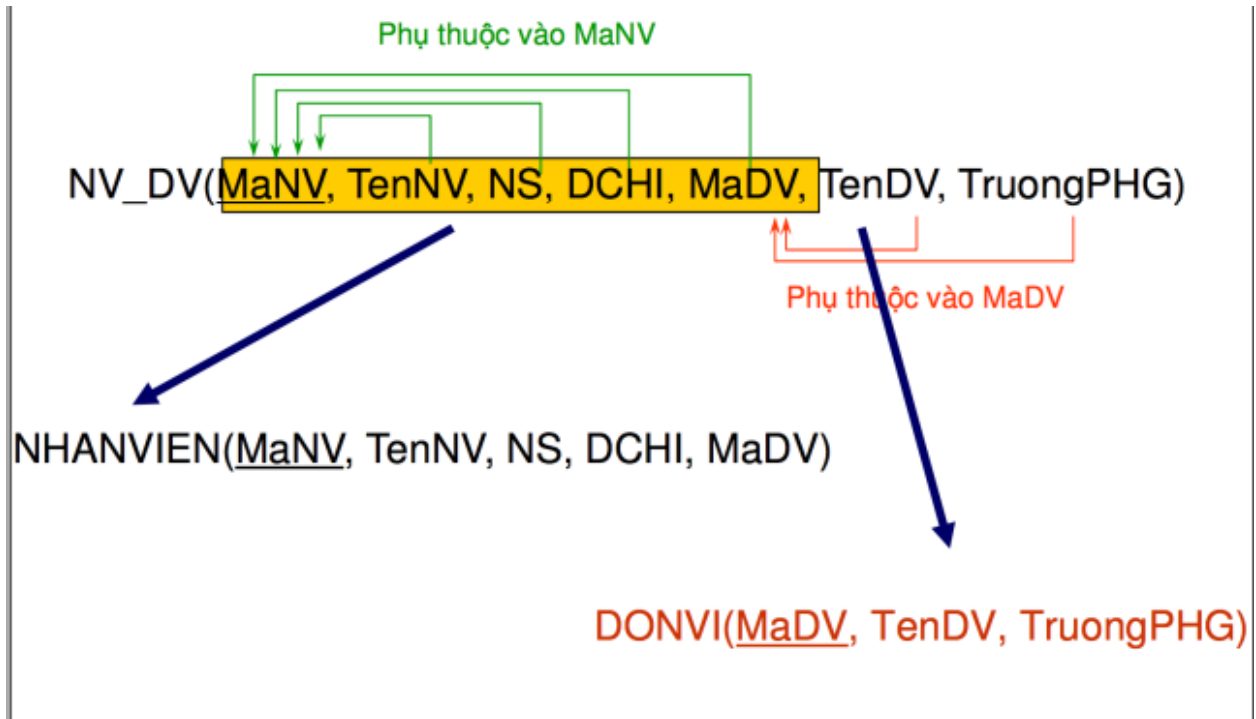
SanPham(idSanpham, moTa, donGia).

HoaDon(idHoaDon, idKhachHang, ngayLapHoaDon).

HoaDonSanPham(idHoaDon, idSanpham, soLuong)

Ví dụ bổ sung

- **Dạng chuẩn 3 (3NF – third normal form)**
- Phải thỏa mãn chuẩn **2NF**
- Các trường thuộc tính không phải là khóa chính, **phải phụ thuộc trực tiếp vào khóa chính.** **Không được phép phụ thuộc bắc cầu thông qua thuộc tính khác.** Nếu có phải tách nhóm các thuộc tính đó thành quan hệ mới.
- Ví dụ: NhanVien (id, taikhoan, **maTPthanhpho**, **tenthanhpho**.)...



- Dạng chuẩn BCNF (Boyce-Codd normal form):** Dạng chuẩn Boyce–Codd (hoặc BCNF hoặc 3.5NF) là một dạng chuẩn được sử dụng trong chuẩn hóa dữ liệu. Nó là phiên bản mạnh hơn một chút của dạng chuẩn 3 (3NF). BCNF được phát triển vào năm 1975 bởi Raymond F. Boyce và Edgar F. Codd để giải quyết một số loại dị thường không được xử lý bởi 3NF như được định nghĩa ban đầu.
 - o Phải đạt chuẩn **3NF**
 - o Không có thuộc tính khóa nào phụ thuộc vào thuộc tính không khóa
 - Bước 1: Loại bỏ các thuộc tính khóa phụ thuộc hàm vào thuộc tính không khóa ra khỏi quan hệ.
 - Bước 2: Tách thuộc tính vừa loại bỏ thành một quan hệ riêng có khóa chính là thuộc tính không khóa gây ra phụ thuộc.
 - o Giả sử bảng có các thuộc tính (A, B, C, D, E, F) với AB là khóa chính **AB**
 - => C **AB** => D **AB** => E **F** => B Ta thấy bảng trên vi phạm chuẩn Boyce-Codd vì B là thuộc tính khóa nhưng lại phụ thuộc vào F là thuộc tính không khóa. Từ đó, ta sẽ tách bảng ra để theo chuẩn như sau: Bảng 1: (A, F, C, D, E) Bảng 2: (F, B). (employee_id, first_name, last_name, title. Trong bảng này, cột employee_id giúp xác định first_name và last_name. Tương tự, cặp (first_name, last_name) cũng sẽ xác định được employee_id).
- Dạng chuẩn 4NF**
 - o Phải đạt chuẩn **3NF** – SV có thể tìm hiểu thêm.

5. ERD

Dungna29@fe.edu.vn

IV.