**Tổng Kết: CSDL** là tập hợp dữ liệu liên quan với nhau được lưu trữ có cấu trúc. Người dùng có thể dễ dàng cập nhật dữ liệu hoặc trích xuất thông tin từ **CSDL**. Ban đầu dữ liệu lưu trữ rời rạc dưới dạng các file, gọi là **mô hình dữ liệu** file phẳng. Sau đó, các mô hình dữ liệu khác được thiết kế cho phép mô tả cách thức lưu trữ dữ liệu và cách thức để truy nhập dữ liệu dễ dàng.

**Hệ quản trị CSDL (DBMS)** là tập hợp các chương trình cho phép người dùng lưu trữ, cập nhật và trích xuất thông tin từ CSDL.

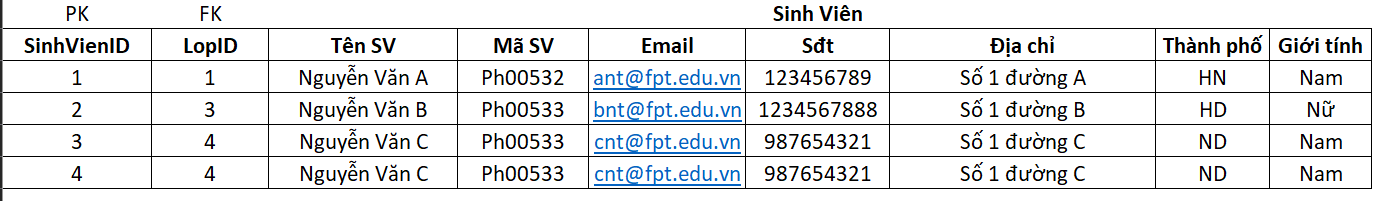
**Hệ quản trị CSDL quan hệ (RDBMS)** là tập hợp các chương trình cho phép tạo và thao tác với CSDL quan hệ.

**Có nhiều đối tượng người dùng RDBMS như**: quản trị CSDL, thiết kế CSDL, phân tích và thiết kế ứng dụng, cài đặt CSDL, người dùng cuối

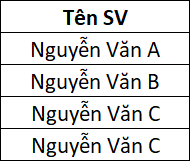
Những thuật ngữ RDBMS

**Database:** Một cơ sở dữ liệu là một tập hợp các bảng dữ liệu, với dữ liệu có liên quan.

**Bảng dữ liệu (Table):** Một bảng là một ma trận dữ liệu. Một bảng trong một cơ sở dữ liệu trông giống như một bảng tính đơn giản.



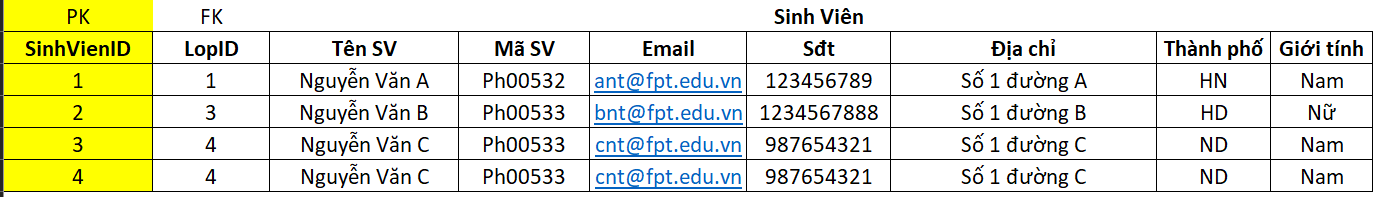
**Cột:** Một cột chứa cùng một kiểu dữ liệu, ví dụ như tên **Sinh Viên**.



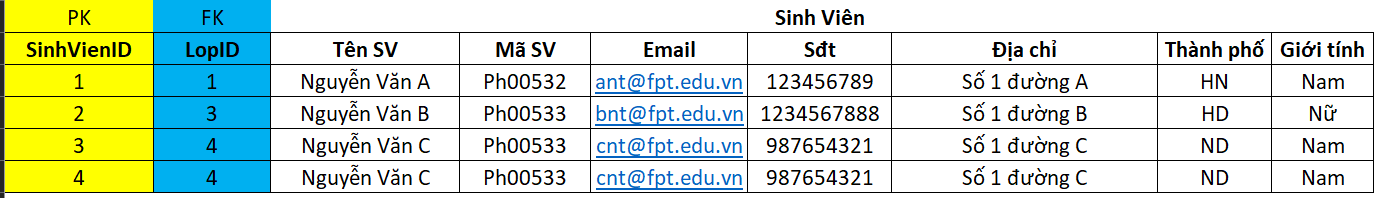
**Hàng:** Một hàng (row, entry, record) là một nhóm dữ liệu có liên quan.



**Primary Key:** Một **P**rimary **K**ey (Khóa chính) là duy nhất. Một giá trị key không thể xuất hiện hai lần trong một bảng. Với một key, bạn có thể tìm thấy phần lớn trên một hàng.

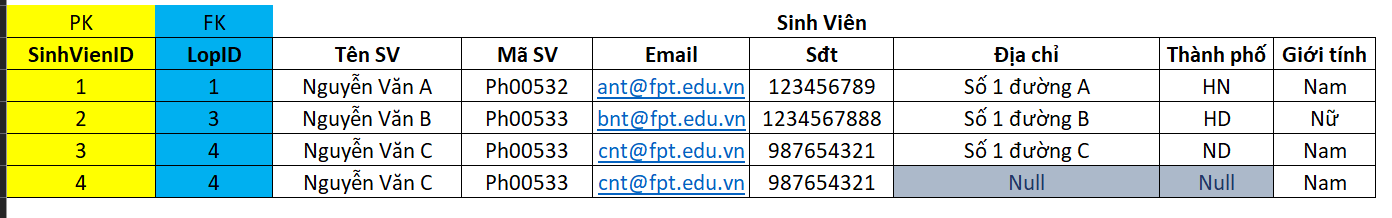


**Foreign Key:** Bạn tưởng tượng về Foreign Key như là cái ghim liên kết giữa hai bảng.



**Giá trị NULL là gì?**

Giá trị NULL của một bảng xác định rằng bản trường đã bị để trống trong khi tạo ghi. Nó là khác hoàn toàn với giá trị 0 hoặc một trường mà chứa khoảng trống (space).



1. Các bước để xây dựng được một cơ sở dữ liệu:
   1. **Thiết kế một CSDL được phân thành các mức khác nhau:**



* + 1. **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 (ERD)** 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 văn bản 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:**

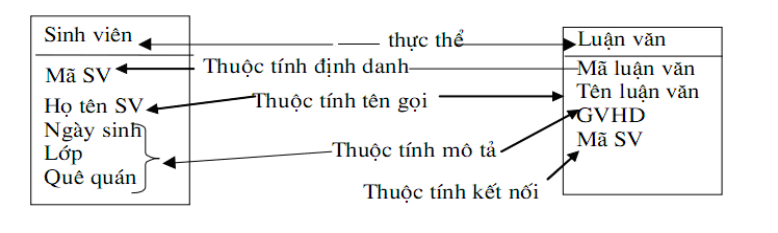
Lựa chọn đúng đối tượng phỏng vấn: lãnh đạo, phòng ban, các bộ phận, người tham gia trực tiếp.

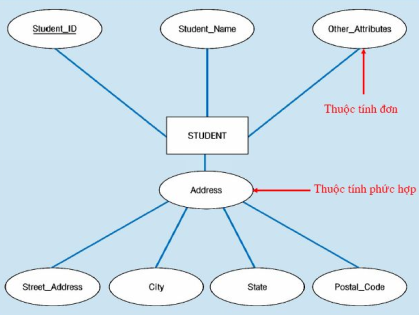
Tìm hiểu bức tranh tổng thể trong từng lĩnh vực từng bộ phận. Cần nắm rõ quy trình nghiệp vụ, mô tả toàn bộ các công việc phải thực hiện, quy tắc xử lý.

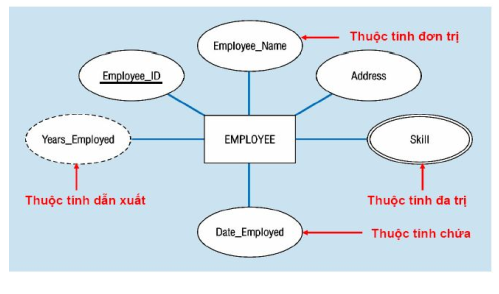
* + - * + **Kỹ thuật sử dụng phiếu khảo sát**
        + **Kỹ thuật phân tích tài liệu**
      * **Sau khi chúng ta thực hiện hết các bước trên sẽ có 1 tài liệu thiết kế cơ sở dữ liệu mức khái niệm:**
        + Lược đồ quan niệm là một mô tả súc tích về các yêu cầu dữ liệu của những người sử dụng.
        + 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.
    1. **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.
    2. **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.
  1. **Khái niệm thực thể**
     1. **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)
    1. Ví dụ: Một sinh viên được gọi một thực thể, **tập hợp (nhiều) các sinh viên là một tập thực thể.**
  1. **Thuộc tính:**
     1. **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.
     2. **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**
     3. **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.
     4. **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.
     5. **Thuộc tính đơn:** là thuộc tính không thể phân chia ra được thành các thành phần nhỏ hơn.
     6. **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)







Các kiểu dữ liệu trong MySQL

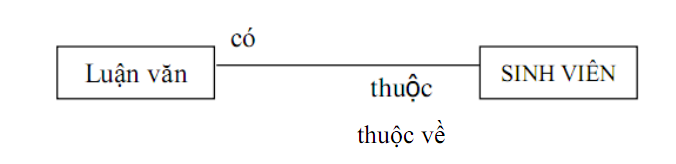
Trong MySQL, kiểu dữ liệu được chia làm ba loại chính: kiểu số, kiểu văn bản, kiểu ngày tháng

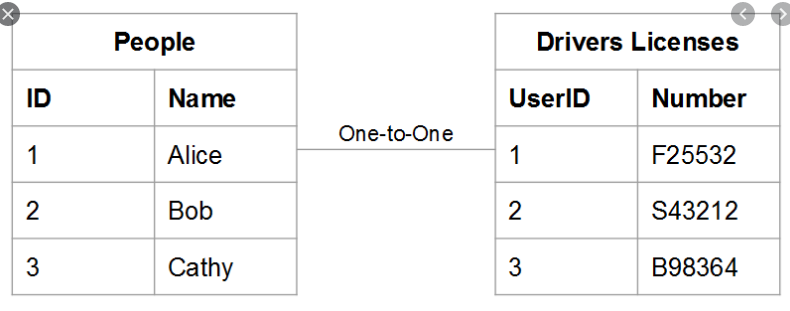
|  |  |
| --- | --- |
| **Kiểu dữ liệu (Số)** | **Mô tả** |
| **TINYINT(size)** | Lưu trữ một số nguyên có giá trị từ -128 đến -127 hoặc 0 đến 255 |
| **SMALLINT(size)** | Lưu trữ một số nguyên có giá trị từ -32768 đến 32767 hoặc 0 đến 65535 |
| **MEDIUMINT(size)** | Lưu trữ một số nguyên có giá trị từ -8388608 đến 8388607 hoặc 0 đến 16777215 |
| **INT(size)** | Lưu trữ một số nguyên có giá trị từ -2147483648 đến 2147483647 hoặc 0 đến 4294967295 |
| **BIGINT(size)** | Lưu trữ một số nguyên có giá trị từ -9223372036854775808 đến 9223372036854775807 hoặc 0 đến 18446744073709551615. |
| **FLOAT(size,d)** | Lưu trữ một số thập phân loại nhỏ (Ví dụ: 567.25). Tham số “size” dùng để xác định kích thước tối đa của phần nguyên (nằm bên trái dấu chấm). Tham số “d” dùng để xác định kích thước tối đa của phần thập phân (nằm bên phải dấu chấm). |
| **DOUBLE(size,d)** | Lưu trữ một số thập phân loại lớn. Tham số “size” dùng để xác định kích thước tối đa của phần nguyên (nằm bên trái dấu chấm). Tham số “d” dùng để xác định kích thước tối đa của phần thập phân (nằm bên phải dấu chấm). |
| **DECIMAL(size,d)** | Lưu trữ như một chuỗi, cho phép một dấu thập phân cố định. Tham số “size” dùng để xác định kích thước tối đa của phần nguyên (nằm bên trái dấu chấm). Tham số “d” dùng để xác định kích thước tối đa của phần thập phân (nằm bên phải dấu chấm). |

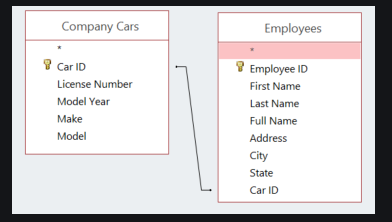
|  |  |
| --- | --- |
| **Kiểu dữ liệu (Chuỗi)** | **Mô tả** |
| **CHAR(size)** | Dùng để lưu trữ một chuỗi ký tự có chiều dài tối đa là 8000 ký tự |
| **VARCHAR(size)** | Dùng để lưu trữ một chuỗi ký tự có chiều dài tối đa là 8000 ký tự. Nếu đặt “size” lớn hơn 8000 thì nó sẽ chuyển sang kiểu TEXT |
| **NVARCHAR(size)** | **UNICODE** Dùng để lưu trữ một chuỗi ký tự có chiều dài tối đa là 4000 ký tự. Nếu đặt “size” lớn hơn 4000 thì nó sẽ chuyển sang kiểu TEXT |
| **TINYTEXT** | Dùng để lưu trữ một chuỗi ký tự có chiều dài tối đa là 255 ký tự |
| **TEXT** | Dùng để lưu trữ một chuỗi ký tự có chiều dài tối đa là 65,535 ký tự |
| **BLOB** | Dùng để lưu trữ dữ liệu nhị phân tối đa là 65,535 byte |
| **MEDIUMTEXT** | Dùng để lưu trữ một chuỗi ký tự có chiều dài tối đa là 16,777,215 ký tự |
| **MEDIUMBLOB** | Dùng để lưu trữ dữ liệu nhị phân tối đa là 16,777,215 byte |
| **LONGTEXT** | Dùng để lưu trữ một chuỗi ký tự có chiều dài tối đa là 4,294,967,295 ký tự |
| **LONGBLOB** | Dùng để lưu trữ dữ liệu nhị phân tối đa là 4,294,967,295 byte |

|  |  |
| --- | --- |
| **Kiểu dữ liệu (thời gian)** | **Mô tả** |
| **DATE()** | Lưu trữ một ngày theo định dạng YYYY-MM-DD (Ví dụ: 2016-09-12 tức là lưu ngày 12 tháng 9 năm 2016) |
| **TIME()** | Lưu trữ thời gian theo định dạng HH:MI:SS (Ví dụ 17:25:36 tức là lưu 17 giờ 25 phút 36 giây) |
| **YEAR()** | Lưu trữ một năm theo định dạng hai số hoặc bốn số |
| **DATETIME()** | Lưu trữ một ngày cùng với thời gian theo định dạng YYYY-MM-DD HH:MI:SS (Ví dụ: 2016-09-12 17:25:36 tức là lưu ngày 12 tháng 9 năm 2016 lúc 17 giờ 25 phút 36 giây) |

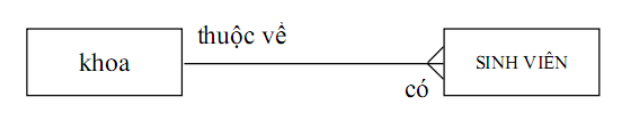
* 1. **Mỗi quan hệ (Relationship)**
     1. **Liên kết (còn gọi là quan hệ)** là sự kết hợp giữa hai hay nhiều thực thể phản ánh sự ràng buộc trong quản lý. **Đặc biệt: Một thực thể có thể liên kết với chính nó ta thường gọi là tự liên kết.** Giữa hai thực thể có thể có nhiều hơn một liên kết. Có **ba** **kiểu** liên kết: **một - một, một - nhiều, nhiều - nhiều.**
     2. **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.

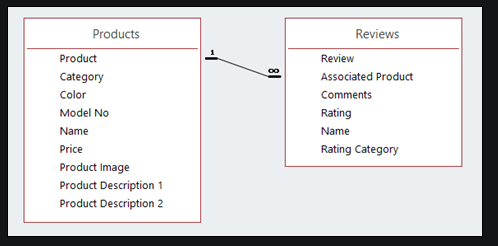


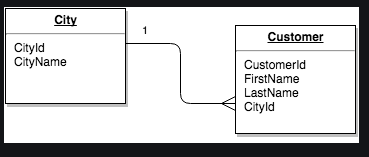


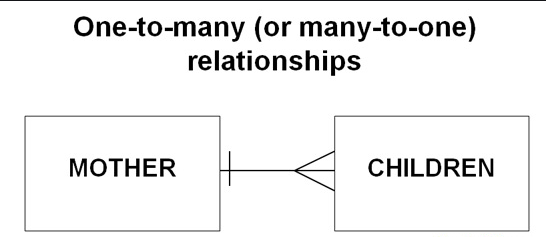


* + 1. **Liên kết một nhiều (1 - N):** Quan hệ 1-N là quan hệ giữa hai tập thực thể trong đó mỗi thực thể của tập này có thể liên kết với nhiều thực thể của tập còn lại.
       - **Ví dụ:** Một khoa có nhiều sinh viên. Một sinh viên thuộc về một khoa.

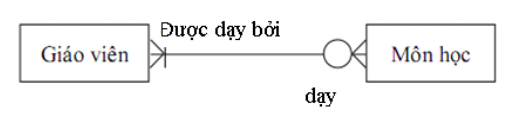


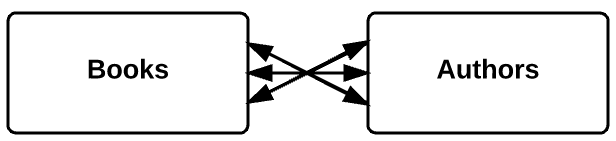


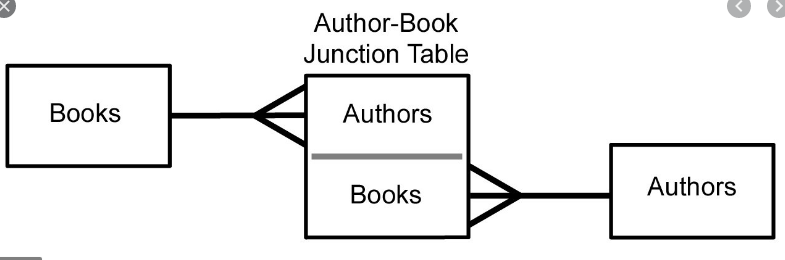


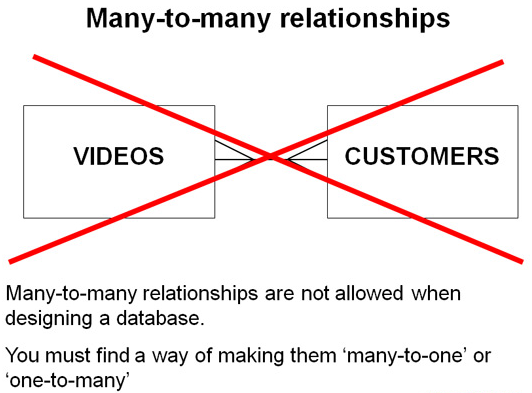


* + 1. **Liên kết nhiều nhiều (N - N):** Quan hệ **N-N** là quan hệ giữa hai tập thực thể trong đó một thực thể của tập này có thể liên kết với 0, 1 hoặc nhiều thực thể của tập kia, và ngược lại. Thường quan hệ **N-N** có thêm phần dữ liệu giao nhau để thêm thông tin cụ thể cho mối quan hệ. **Hoặc** 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 tuỳ 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:** 
         * **Tuỳ 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.”



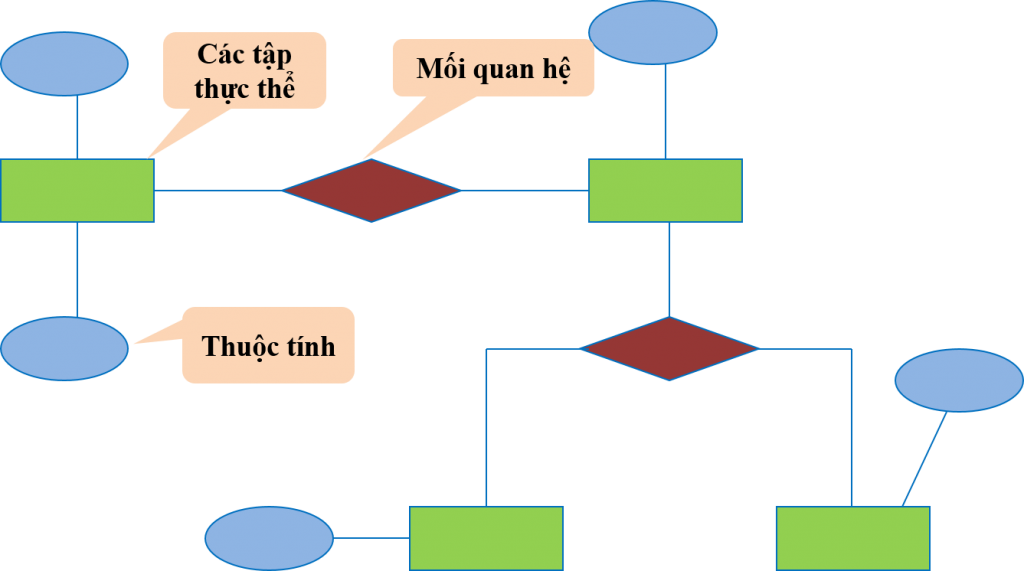


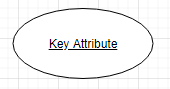
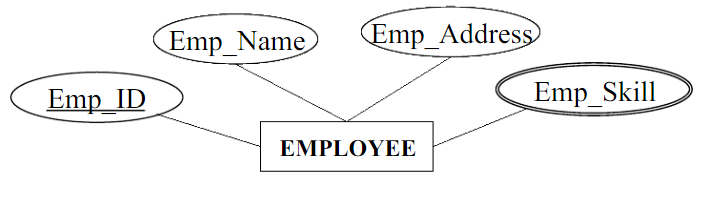




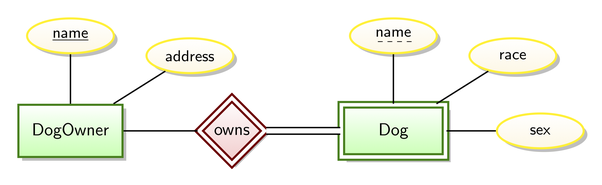
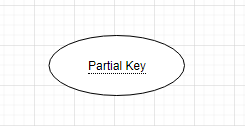
* + 1. **Lưu ý:** Mô hình dữ liệu không chỉ là công cụ phân tích thiết kế mà còn như một phương pháp kiểm tra chặt chẽ các yêu cầu nghiệp vụ của người sử dụng.
    2. **Nếu hai thực thể có quan hệ một - một** thường có ít lý do để coi chúng như hai bảng tách biệt nên người ta thường gộp hai thực thể làm một
    3. **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.
    4. **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.**

1. **sXây dựng một database:**
   1. **Thực thể và tập thực thể (Entity)**
2. 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 thiển hiện mỗi quan hệ nhiều (Chính là bảng tạm – bảng trung gian)**
3. Một nhóm bao gồm các thực thể tương tự nhau tạo thành một tập thực thể
4. 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**.)
5. **Xác định được thực thể thông qua việc đọc và phân tích tài liệu.**
   1. **Xác định thuộc tính (Attribute)**
6. **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)
7. **Relational Database** được tổ chức thành nhiều bảng, và giữa các bảng quan hệ với nhau bằng các **khóa**.
8. **Cột** chính là thuộc tính (**attribute**) của đối tượng Customer, bao gồm cột: Last Name, First Name, Email.....
9. **Dòng (Row)** là các **“bản ghi**”, mình hay gọi là các **record**, là số lượng dữ liệu mà table đó lưu trữ trong **database**.
10. 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.**
11. 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.
12. **Foreign Key** sẽ là **Primary Key** ở một **table**, nhưng lại là **Foreign Key** ở một **table** khác mà **table** đó liên kết đến.
    1. Mô hình quan hệ - thực thể **(Entity – Relationship Model)**
13. **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.
14. **Các thành phần cơ bản của mô hình E-R** bao gồm các thực thể và thuộc tính của nó.
15. **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)**

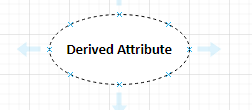
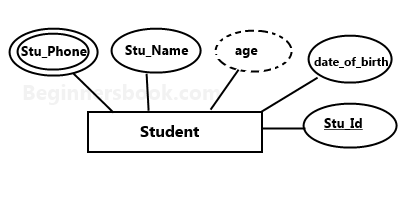


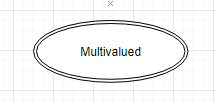
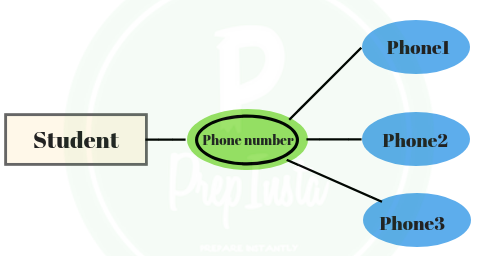
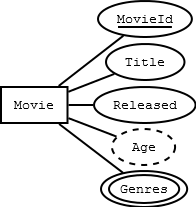
Nếu chữ trong hình Ovan được **gạch dưới**, nghĩa là với attribute này, mỗi dòng dữ liệu chỉ có một giá trị duy nhất. Thường nó sẽ là ID của bảng dữ liệu đó. Nó còn gọi **là Unique key, unique attribute, Primary** **Key**, hay **Indentifier**. Chú ý là mỗi Entity có thể có nhiều **Unique key**. Khi đó 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 gía trị duy nhất cho từng sinh viên nhưng ta chỉ chọn một trong hai làm Primary Key.



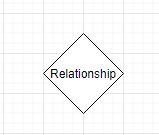
Nhưng nếu chữ trong hình Ovan **được gạch dưới đứt quãng** và **hình chữ nhật có 2 viền**, điều này nghĩa là **Weaky** **Key Attribute** đó là **Partial Key**. Tức là một mình nó không thể dùng để xác định được Entity vì nó phụ thuộc vào Key (unique attribute) của một Entity khác. Ta phải dùng Unique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

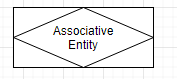
Nếu **hình Ovan (Attribute)** có viền **gạch đứt quãng**, nó nghĩa là **Derived (Thuộc tính dẫn xuất)**. Nghĩa là ta không cần thu thập data cho nó mà có thể suy ra từ một **Attribute** khác trong **Database**. Ví dụ dựa vào ngày sinh, ta có thể suy ra người đó bao nhiêu tuổi. (Thuộc tính dẫn xuất không 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 thầy bằng cách tính toán chênh lệch giữa ngày hiện tại và ngày của quá khứ.)

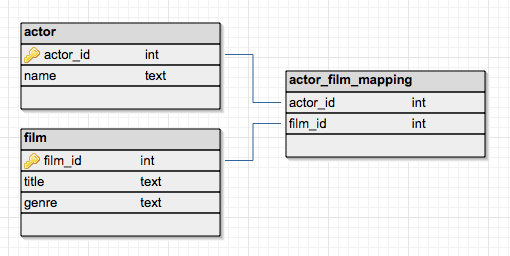
  

Nếu hình **Ovan (Attribute)** có **2 viền** đây là một Attribute có nhiều giá trị khác nhau cho cùng một Entity. Ví dụ attribute có tên Kỹ Năng Nhân Viên. Mỗi Nhân viên sẽ có vài kỹ năng khác nhau chứ không chỉ có một kỹ năng làm việc.

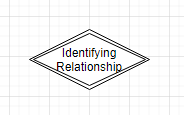
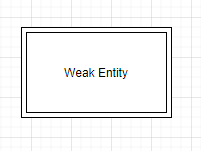


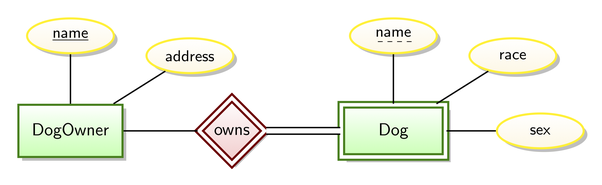
Một thực thể A **quan hệ** với một thực thể B bởi hình **THOI**. Thể hiện mối liên quan giữa hai hay nhiều thực thể. **Mỗi liên kết có một tên gọi và thường dùng động từ.**



**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 bẳng 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.

s 



**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** phục thuộc vào một **bảng cha.** Ví dụ trên2: **owns: sở hữu –** Bảng chó bị phụ thuộc vào bảng chủ của con 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 dụ2: 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.

