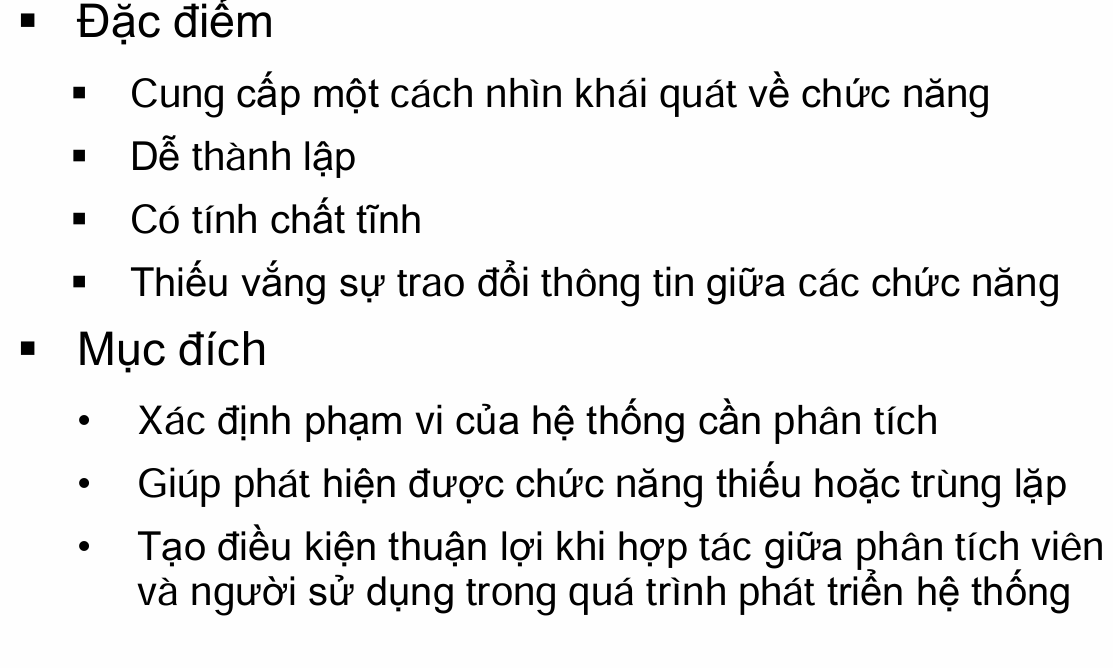
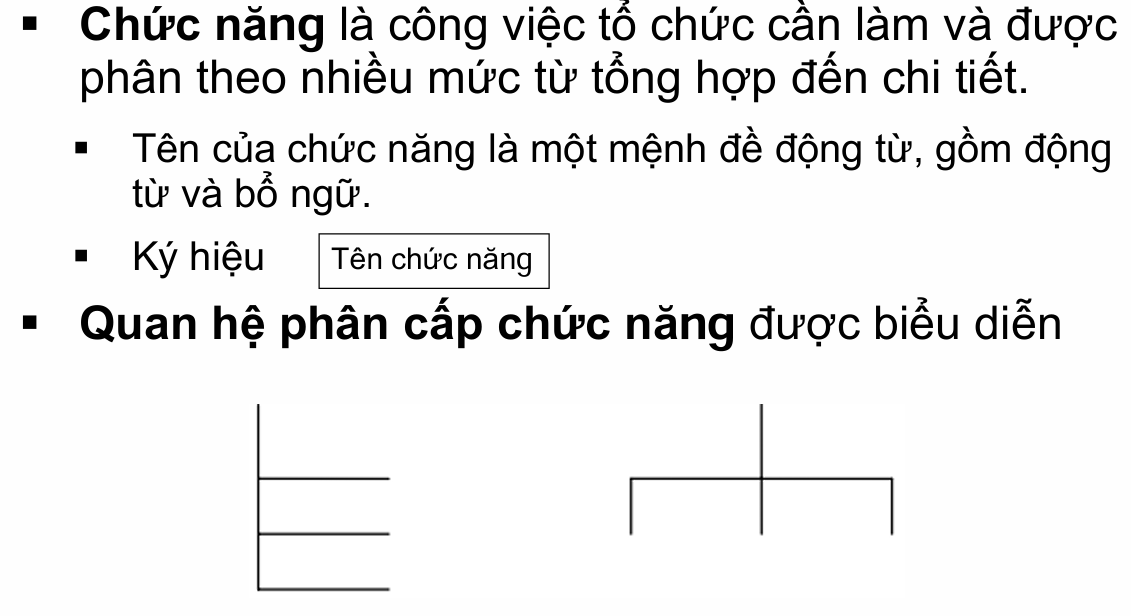
**Mô hình phân rã chức năng – BFD**

* Là công cụ Biểu diễn sự phân rã có thứ bậc đơn giản các chức năng cần thực hiện. *Mỗi công việc được chia ra làm các công việc con, số mức chia ra phụ thuộc kích cỡ và độ phức tạp của hệ thống.*
* **Đặc điểm & mục đích của BFD**

****

* **Các thành phần của BFD**



* **Nguyên tắc phân rã chức năng**

**Cách phân chia thường theo nguyên tắc:**

Trong quá trình tiếp cận một tổ chức theo phương pháp từ trên xuống (top- down) ta nhận được thông tin về các chức năng từ mức gộp (do lãnh đạo cung cấp) đến mức chi tiết (do các bộ phân chức năng cung cấp). Cách phân rã này cho là phù hợp với sự phân công các chức năng của một tổ chức nào đó.

* Cách phân chia này thường theo nguyên tắc sau:

• Mỗi chức năng con phải là một bộ phận thực sự tham gia thực hiện chức năng cha.

• Việc thực hiện tất cả các chức năng con phải đảm bảo thực hiện được các chức năng ở mức trên đã phân rã ra chúng.

**▪ Các bước tiến hành**

• B1 - Xác định chức năng

• B2 - Phân rã các chức năng

**MÔ HÌNH LUỒNG (DÒNG) DỮ LIỆU – DFD (DATA FLOW DIAGRAM)**

Mô hình luồng dữ liệu (DFD - Data Flow Diagram) là một công cụ mô tả mối quan hệ thông tin giữa các công việc

* **Đặc trưng:**
* Cách tiếp cận chủ yếu theo phương pháp top-down: xem các hoạt động xử lý như “hộp đen” và chỉ quan tâm đến việc lưu trữ xử lý thông tin giữa các “hộp đen” này.

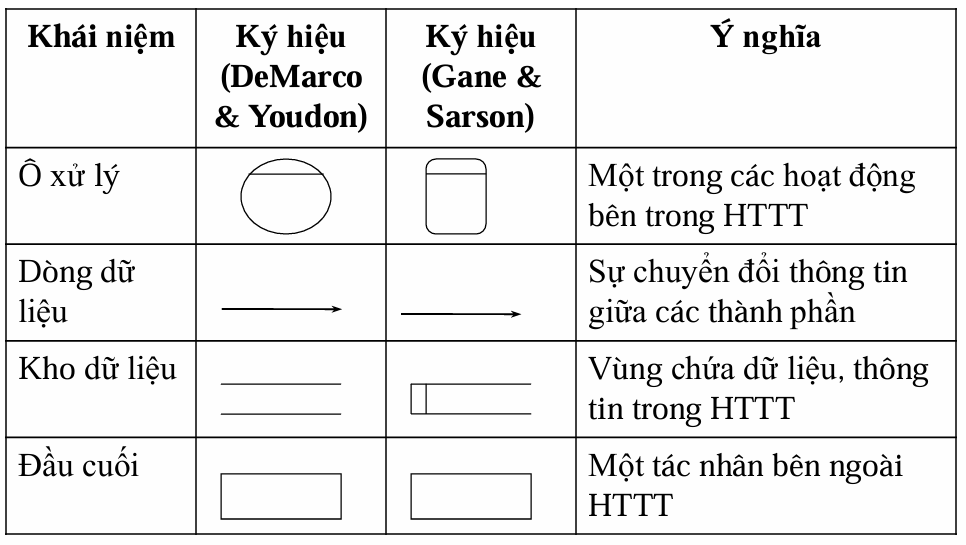
*Ví dụ, trong một hệ thống quản lý bán hàng, một quá trình xử lý "Tính tổng hóa đơn" có thể được xem như một "hộp đen". Chúng ta chỉ cần biết rằng dữ liệu đầu vào là các mặt hàng và số lượng của chúng, và đầu ra là tổng số tiền cần thanh toán, mà không cần biết chi tiết bên trong quá trình đó tính toán như thế nào.*

*Cách tiếp cận này giúp mô hình DFD trở nên dễ hiểu hơn và cho phép người thiết kế tập trung vào cách các quá trình tương tác với nhau và cách dữ liệu di chuyển trong toàn bộ hệ thống.*

* Mặc dù DFD biểu diễn cả xử lý lẫn dữ liệu của hệ thống, nhưng quan trong xử lý hơn là dữ liệu.

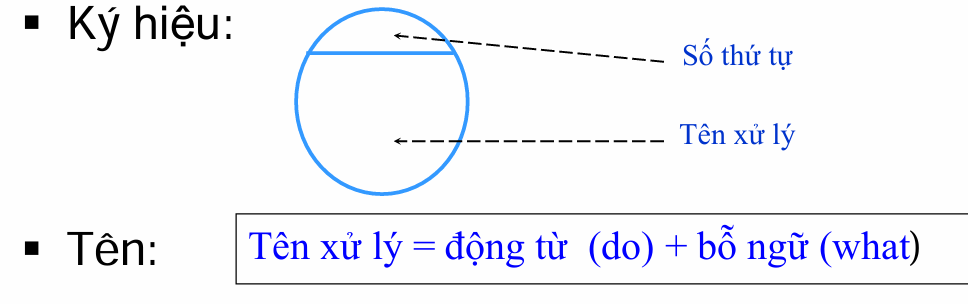
*Mặc dù DFD mô tả cả xử lý và dữ liệu, nhưng nó nhấn mạnh nhiều hơn vào cách xử lý dữ liệu diễn ra. Các quá trình xử lý (processes) được biểu diễn chi tiết, trong khi các kho dữ liệu (data stores) và luồng dữ liệu (data flows) chỉ được mô tả để hỗ trợ việc hiểu rõ quá trình xử lý.*

* Các nhiều cách tiếp cận và phương pháp luận khác nhau để tạo ra, sử dụng mô hình DFD. Các phương pháp này cung cấp các hướng dẫn, quy tắc và ký hiệu cụ thể để giúp người thiết kế hệ thống xây dựng và hiểu rõ hơn về các DFD. Gồm các phương pháp chính như : Tom DeMarco, Yourdon&Constantinie, Gane & Sarson.
* **CÁC THÀNH PHẦN TRONG DFD (4 TP)**

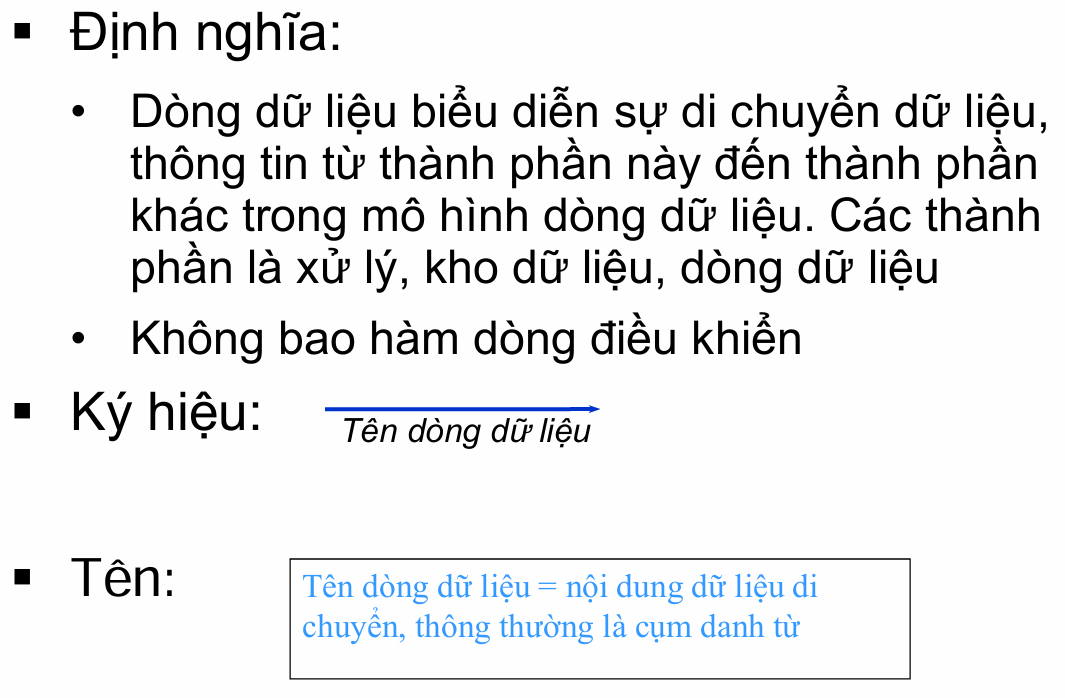


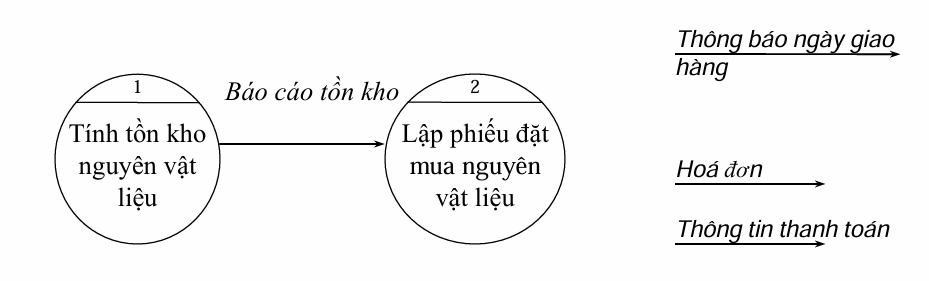
* **Ô xử lí (process):**

- là một hoạt động xử lý bên trong hệ thống thông tin. Bao gồm: Tạomới thông tin, Sửdụng thông tin, Cậpnhật thông tin, Huỷbỏthông tin

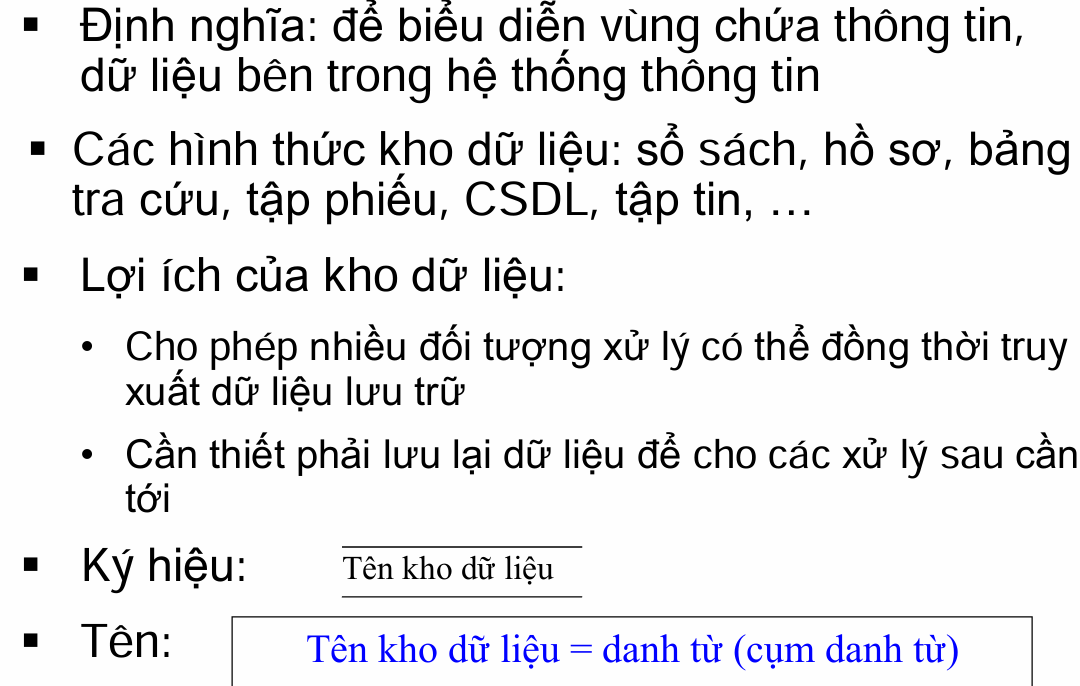


* **Dòng dữ liệu (data flow):**

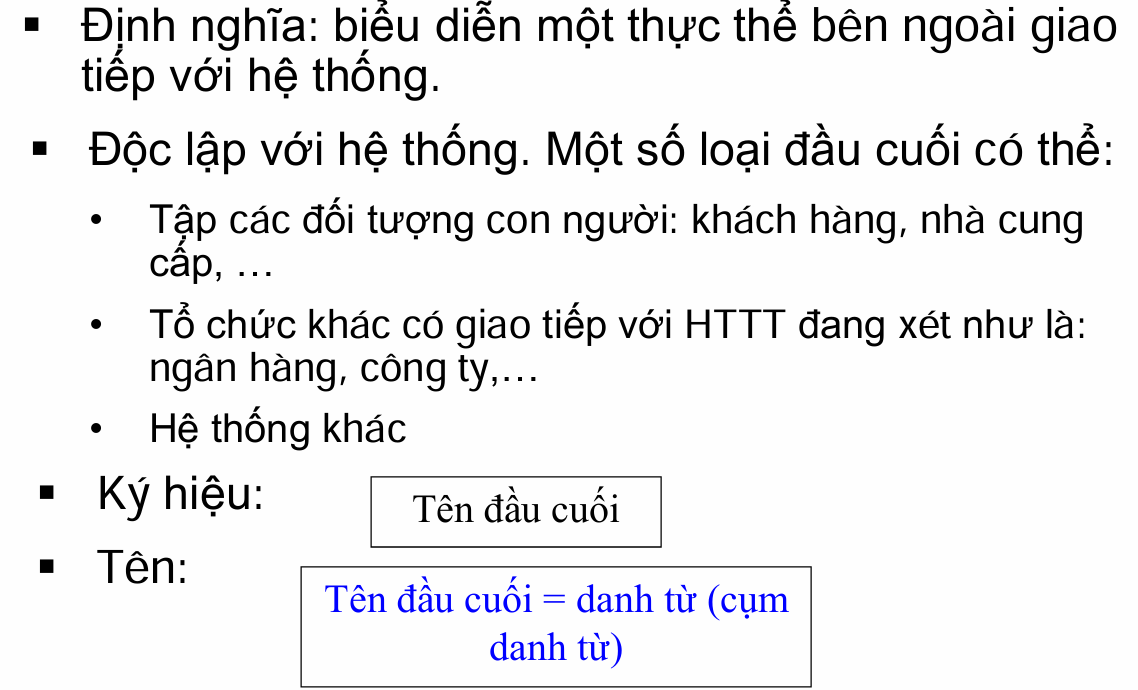




* **Kho dữ liệu (data store)**

****

* **Đầu cuối (terminal)**

****

**Trình tự xây dựng mô hình luồng dữ liệu:**

[(PDF) CHƯƠNG 3: PHÂN TÍCH CHỨC NĂNG CỦA HỆ THỐNG 3.1. MÔ HÌNH PHÂN RÃ CHỨC NĂNG | Kim Kim - Academia.edu](https://www.academia.edu/29164068/CH%C6%AF%C6%A0NG_3_PH%C3%82N_T%C3%8DCH_CH%E1%BB%A8C_N%C4%82NG_C%E1%BB%A6A_H%E1%BB%86_TH%E1%BB%90NG_3_1_M%C3%94_H%C3%8CNH_PH%C3%82N_R%C3%83_CH%E1%BB%A8C_N%C4%82NG)

**BƯỚC 1: XÂY DỰNG MÔ HÌNH LUỒNG DỮ LIỆU MỨC KHUNG CẢNH (MỨC 0)**

* Mô hình luồng dữ liệu mức khung cảnh bao gồm 1 chức năng duy nhất biểu thị toàn bộ hệ thống đang nghiên cứu, chức năng này được nối với mọi tác nhân ngoài của hệ thống.
* Các luồng dữ liệu giữa chức năng và tác nhân ngoài chỉ thông tin vào và ra của hệ thống.

**Bước 2**: xây dựng mô hình luồng dữ liệu mức đỉnh (mức 1)

* Với mức đỉnh các tác nhân ngoài của hệ thống ở mức khung cảnh được giữ nguyên với các luồng thông tin vào ra.
* Hệ thống được phân rã thành các chức năng mức đỉnh là các tiến trình chính bên trong hệ thống theo mô hình phân rã chức năng mức 1.
* Xuất hiện thêm các kho dữ liệu và luồng thông tin trao đổi giữa các chức năng mức đỉnh.

**Bước 3: xây dựng mô hình luồng dữ liệu mức dưới đỉnh (mức 2 và dưới 2)**

* ở mức này thực hiện phân rã đối với mỗi chức năng của mức đỉnh
* khi thực hiện mức phân rã này vẫn phải căn cứ vào mô hình phân rã chức năng để xác định các chức năng con sẽ xuất hiện trong mô hình luồng dữ liệu.
* việc phân rã có thể tiếp tục cho đến khi đủ số mức cần thiết
* khi phân rã các chức năng phải đảm bảo tất cả các luồng thông tin vào ra ở chức năng mức cao phải có mặt trong các chức năng mức thấp hơn và ngược lại.
* chú ý:

+ các kho dữ liệu không xuất hiện ở DFD mức khung cảnh

+ nên đánh số các chức năng theo sự phân cấp

+ các kho dữ liệu, các tác nhân ngoài có thể xuất hiện nhiều lần

+ số mức phụ thuộc vào độ phức tạp của hệ thống.

**LƯỢC ĐỒ THỰC THẾ KẾT HỢP – ERD**

Mô hình ER sử dụng 3 khái niệm để mô hình hóa dữ liệu trong thế giới thực:

* Tập thực thể:

Thực thể mạnh/ thực thể yếu.

Thực thể là một vật hoặc đối tượng (cụ thể hay trừu tượng) trong thế giới thực, có sự tồn tại độc lập và có thể phân biệt với các vật hoặc đối tượng khác.

* **Thực thể mạnh**: tồn tại độc lập với thực thể khác.

• **Thực thể yếu:** không có thuộc tính khóa, sự tồn tại của nó phải phụ thuộc vào thực thể khác.

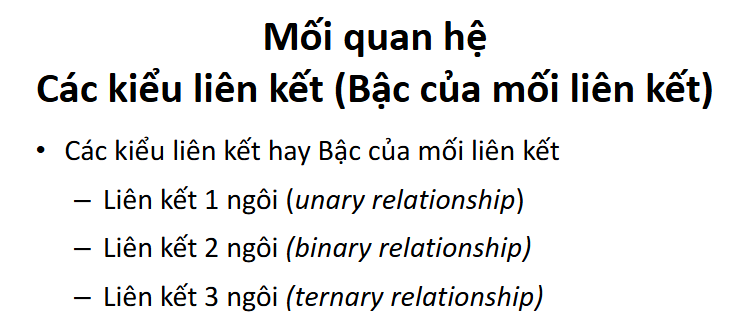
* Thuộc tính:

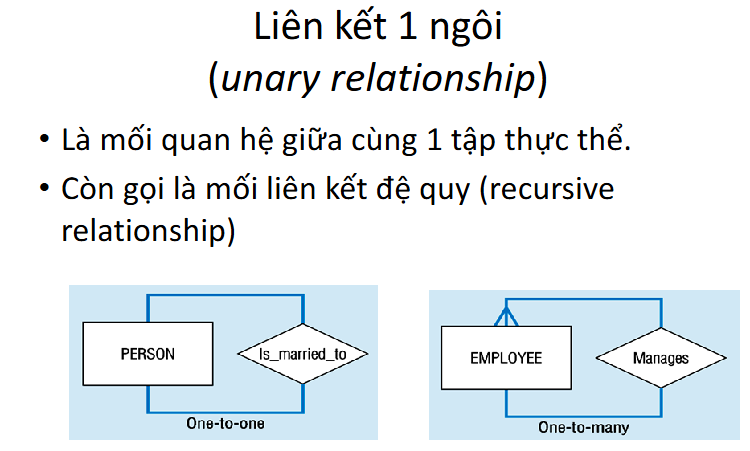
• Thuộc tính: là các tính chất đặc trung của thực thể, là yếu tố thông tin cho biết rõ hơn về thực thể.

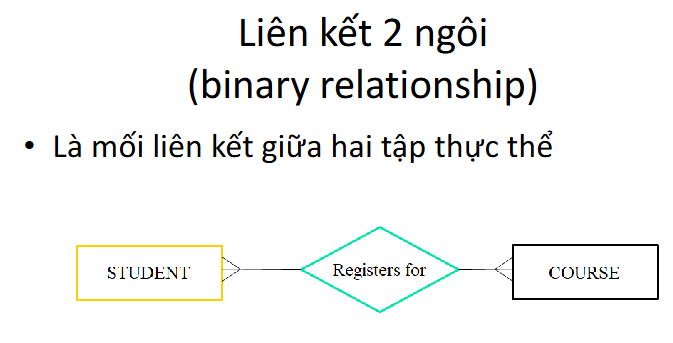
• Mỗi thuộc tính có một miền giá trị. VD: tên người là chuỗi ký tự, tuổi là số nguyên dương. Nếu thuộc tính chưa có giá trị thì qui ước giá trị đó là null

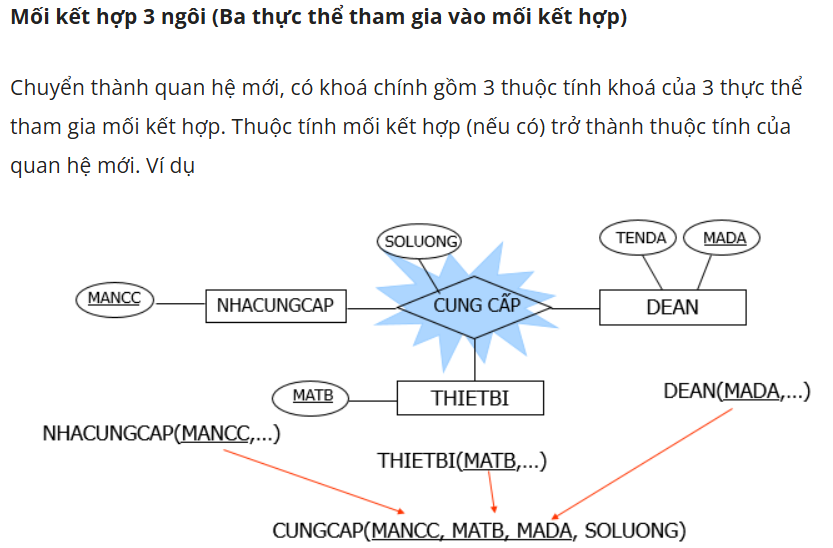
* **Thuộc tính đơn trị**: là thuộc tính chỉ có thể nhận một giá trị duy nhất cho một thực thể cụ thể.
* **Thuộc tính đa trị**: là thuộc tính có thể nhận một hoặc một vài giá trị cho một thực thể <=> nhận nhiều giá trị đồng thời
* Kí hiệu: vòn elip kép (elip nét đôi)
* Mối quan hệ (kiểu liên kết) – mối kết hợp:

Bậc của mối quan hệ





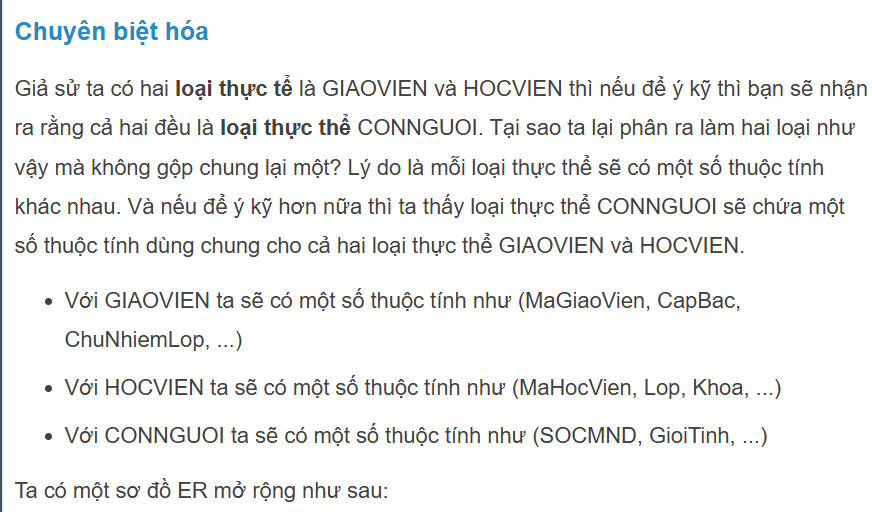


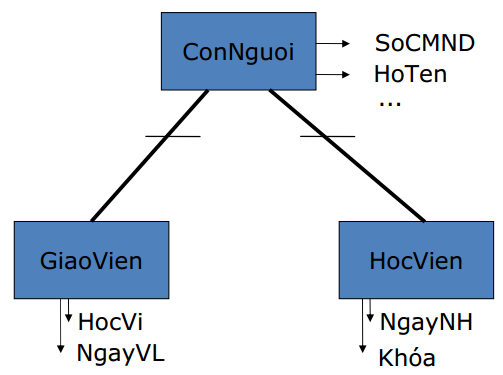


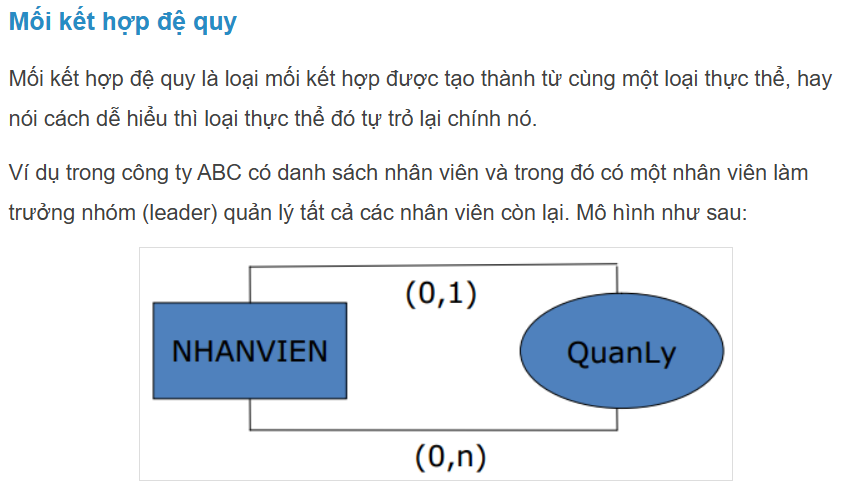
[Mô hình dữ liệu quan hệ (Bài 3) - Gia Sư Tin Học (giasutinhoc.vn)](https://giasutinhoc.vn/database/co-so-du-lieu/mo-hinh-du-lieu-quan-he-bai-3-2/)

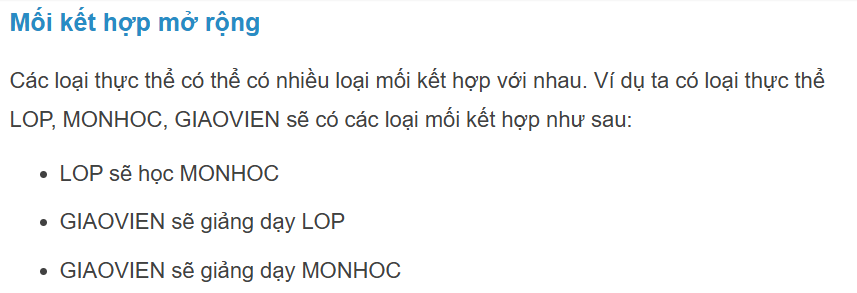
**Quy tắc chuyển đổi ERD sang mô hình dữ liệu quan hệ (mức vật lý)**

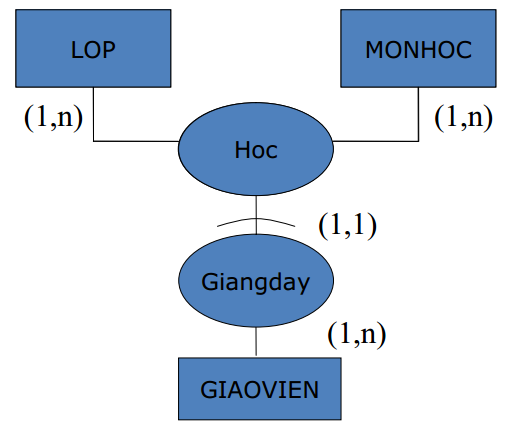
**Mô hình thực thể mối kết hợp mở rộng**

****



****

****



**Các bước PTTK TPDL mức logic – chuyển đổi mô hình quan niệm thành mô hình logic:**

* Xây dựng mô hình quan hệ dữ liệu (tổ chức dữ liệu) từ mô hình thực thể kết hợp (quan niệm dữ liệu)
* Tối ưu hóa mô hình dữ liệu quan hệ
* Mô tả các ràng buộc toàn vẹn ở mức thiết kế tổ chức (thường dùng ngôn ngữ toán học)
* Thiết kế hàm cửa sổ hay các khung hình dữ liệu, cũng như các chỉ mục được sử dụng trong xử lý sau này.

**Các bước chuyển đổi sang mô hình quan hệ:**

B1: chuyển các khái niệm chuyên biệt hóa, tổng quát hóa sang khái niệm quan hệ - trung gian

B2: chuyển thực thể sang quan hệ

B3: chuyển các mối kết hợp sang quan hệ

B4: chuẩn hóa các quan hệ

B5: xem xét các rbtv

**BƯỚC 1: chuyển các khái niệm chuyên biệt hóa, tổng quát hóa – TRUNG GIAN**

**Cách 1:** chuyển thành tổng quát hóa, gom tất cả vào một quan hệ (thường áp dụng khi ở mức chuyên biệt hóa không có hoặc <3 thuộc tính riêng)

* Các đặc trưng của thực thể chuyên biệt sẽ chuyển sang thực thể tổng quát.
* Thêm vào một thuộc tính phân loại (hoặc một thực thể) loại E, bổ sung rang buộc miền giá trị cho thuộc tính loại E.
* Loại bỏ các thực thể chuyên biệt

***Th1: Ở mức chuyên biệt hóa không có thuộc tính riêng***

▪ Bổ sung ràng buộc miền giá trị cho thuộc tính Loại E *(toànbộ(t)và bánphần (p), chồng lắp (o) và riêngbiệt(e))*

• (t,e): → Ràng buộc MGT(LoạiE) ≅ {E1, E2}

• (t,o): → Ràng buộc MGT(LoạiE) ≅ {E1, E2, E1E2}

• (p,e): → Ràng buộc MGT(LoạiE) ≅ {E, E1, E2} A E’ R E E (A, Loại E) Loại E 94

• (p,o): → Ràng buộc MGT(LoạiE) ≅ {E, E1, E2, E1E2}

***Th2: Ở mức chuyên biệt hóa có ít thuộc tính riêng (<= 3)***

▪ Bổ sung ràng buộc miền giá trị cho thuộc tính Loại E như TH1

▪ Bổ sung ràng buộc để đảm bảo ngữ nghĩa

**Cách 2:** Chuyển thành chuyên biệt hóa, chia thành các quan hệ riêng (thường áp dụng khi ở mức chuyên biệt hóa có nhiều thuộc tính riêng (>3))

• Chuyển đổi tất cả đặc trưng (thuộc tính, mối kết hợp, định danh) của thực thể tổng quát xuống lần lượt các thực thể chuyên biệt

• Hủy bỏ thực thể tổng quát

• Cách này chỉ có thể áp dụng cho cấu trúc tổng quát hóa toàn bộ (t) và riêng biệt (e), các cấu trúc còn lại không thể áp dụng

**Nhận xét chung:** việc lựa chọn C1 hay C2 tuỳ theo cấu trúc tổng quát hóa, số thuộc tính của các thực thể chuyên biệt, ngoài ra còn phụ thuộc vào các yếu tố sau:

• Dung lượng:

- Gộp: Dung lượng lớn => truy xuất chậm

- Tách: tùy trường hợp, thông thường dung lượng tối ưu hơn

• Xử lý: xác định các xử lý ưu tiên (các xử lý có tần xuất cao,…), tùy tình huống mà quyết định gộp hay tách

**Cách 3**: dùng mối kết hợp

• Hủy bỏ tổng quát – chuyên biệt

• Tạo mối kết hợp từ thực thể tổng quát đến các thực thể chuyên biệt

**BƯỚC 2: CHUYỂN THỰC THỂ SANG QUAN HỆ**

Thực thể thường:

Thực thể yếu: Thêm vào quan hệ - thực thể yếu thuộc tính khóa của quan hệ - thực thể chủ

**BƯỚC 3: CHUYỂN MỐI KẾT HỢP SANG QUAN HỆ**

▪ **Mối kết hợp Nhiều - Nhiều :**

• Tạo một quan hệ mới biểu diễn mối kết hợp với thuộc tính là các thuộc tính khóa của các thực thể liên quan và các thuộc tính riêng của mối kết hợp

**▪ Mối kết hợp Một - Một :**

• Thêm vào quan hệ - tham gia bắt buộc thuộc tính khóa của quan hệ - tham gia tùy chọn

**▪ Mối kết hợp bậc cao**

• Tạo một quan hệ mới với thuộc tính là các thuộc tính khóa của các thực thể liên quan, cộng với các thuộc tính riêng của mối kết hợp.

**BƯỚC 4: CHUẨN HÓA**

▪ Sau khi đã chuyển đổi mô hình ER sang danh sách các quan hệ, phải chuẩn hóa các quan hệ này nhằm mục đích:

• Giảm thiểu sự trùng lặp dữ liệu

• Tối ưu hóa thời gian cập nhật

• Tránh tình trạng không đồng bộ dữ liệu

**BƯỚC 5: XEM XÉT RBTV**

**1. Phương pháp của Tom DeMarco**

Tom DeMarco là một trong những người tiên phong trong lĩnh vực phân tích hệ thống. Phương pháp của ông có những đặc điểm sau:

* **Đơn giản và dễ hiểu**: Nhấn mạnh vào việc tạo ra các DFD dễ hiểu cho cả người kỹ thuật và người không kỹ thuật.
* **Sử dụng các ký hiệu cơ bản**: Chỉ sử dụng các ký hiệu cơ bản như hình tròn hoặc hình bầu dục cho các quá trình xử lý, mũi tên cho luồng dữ liệu, và các hình chữ nhật cho kho dữ liệu.
* **Tiếp cận từ trên xuống (Top-down)**: Bắt đầu từ mức độ tổng quát nhất và sau đó dần dần chi tiết hóa các thành phần của hệ thống.

**2. Phương pháp của Yourdon & Constantine**

Yourdon và Constantine đã phát triển một phương pháp chi tiết hơn cho việc tạo ra DFD, với những đặc điểm sau:

* **Module hóa**: Nhấn mạnh vào việc chia hệ thống thành các module nhỏ hơn, giúp quản lý phức tạp và bảo trì hệ thống dễ dàng hơn.
* **Quy tắc và hướng dẫn cụ thể**: Cung cấp các quy tắc chi tiết để đảm bảo các DFD được nhất quán và logic.
* **Sử dụng ký hiệu tiêu chuẩn**: Các ký hiệu bao gồm hình tròn hoặc hình bầu dục cho các quá trình xử lý, hình chữ nhật có hai đường thẳng đứng cho kho dữ liệu, và mũi tên cho luồng dữ liệu.

**3. Phương pháp của Gane & Sarson**

Phương pháp của Gane và Sarson được biết đến với sự chi tiết và sử dụng các ký hiệu cụ thể, với các đặc điểm chính sau:

* **Ký hiệu chi tiết và tiêu chuẩn hóa**: Sử dụng các ký hiệu đặc biệt như hình chữ nhật có góc bo tròn cho các quá trình xử lý, và ký hiệu cụ thể cho các kho dữ liệu và luồng dữ liệu.
* **Phân cấp rõ ràng**: Tạo ra các sơ đồ ở nhiều mức độ phân cấp khác nhau, từ tổng quát đến chi tiết.
* **Chú trọng vào mô hình hóa dữ liệu và xử lý**: Đảm bảo rằng cả dữ liệu và quá trình xử lý đều được mô tả một cách chi tiết và rõ ràng.

**Tổng Kết**

* **Sự khác biệt trong cách biểu diễn**:
  + **Tom DeMarco**: Đơn giản, dễ hiểu, tiếp cận từ trên xuống.
  + **Yourdon & Constantine**: Chi tiết, module hóa, có quy tắc cụ thể.
  + **Gane & Sarson**: Chi tiết, sử dụng ký hiệu tiêu chuẩn, phân cấp rõ ràng.
* **Tiếp cận hộp đen**:
  + **Tom DeMarco**: Ít chi tiết bên trong, tập trung vào luồng dữ liệu và chức năng.
  + **Yourdon & Constantine**: Chi tiết hơn, nhấn mạnh vào cách các module tương tác.
  + **Gane & Sarson**: Chi tiết nhất, với sự phân cấp rõ ràng và ký hiệu tiêu chuẩn.
* **Chú trọng vào quá trình xử lý**:
  + Tất cả các phương pháp đều tập trung vào việc mô tả quá trình xử lý, nhưng mức độ chi tiết và cách biểu diễn khác nhau.

Việc lựa chọn phương pháp nào phụ thuộc vào yêu cầu cụ thể của dự án và sự quen thuộc của nhóm phát triển với từng phương pháp.

* Tiếp cận chủ yếu theo hướng từ trên xuống: xem các hoạt động xử lý như hộp đen và quan tâm đến việc lưu trữ và xử lý thông tin giữa các hộp đen này • Mặc dù DDL biểu diễn cả xử lý lẫn dữ liệu hệ thống, nhưng chú ý đến xử lý hơn là dữ liệu • Gồm các nhóm phương pháp chính như sau: Tom DeMarco, Yourdon & Constantine, Gane & Sarson