

**ĐẠI HỌC ĐÀ NẴNG**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**GIÁO TRÌNH**  
**PHÂN TÍCH VÀ THIẾT KẾ**  
**HỆ THỐNG**

**PGS.TS. PHAN HUY KHÁNH**



**ĐÀ NẴNG 8/2001**

# Mở đầu

*Trong lĩnh vực ứng dụng công nghệ thông tin, nếu như trước đây không lâu, máy tính điện tử (MTĐT) còn đóng vai trò của người làm công (taskmaster) thì hiện nay, MTĐT đã trở thành công cụ (tool) cần thiết cho hầu hết các lĩnh vực hoạt động của một quốc gia. Trong tương lai không xa của thiên niên kỷ mới này, MTĐT sẽ trở thành người bạn đồng hành (companion) không thể thiếu của mỗi con người trong liên lạc, giao tiếp và việc làm hàng ngày.*

*Ở Việt Nam, MTĐT, chủ yếu là máy vi tính (PC – Personal Computer) đã và đang xuất hiện ngày càng nhiều trong các xí nghiệp, doanh nghiệp, các cơ quan hành chính xã hội..., ngày càng thâm nhập vào hầu khắp các mặt hoạt động của nền kinh tế quốc dân. Tuy nhiên, MTĐT chỉ mới phục vụ công việc văn phòng như soạn thảo văn bản là chính mà chưa thực sự đóng vai trò chủ đạo giúp con người trong các lĩnh vực quản lý, tự động hoá để tăng năng suất lao động.*

*Một trong những nguyên nhân chính là Việt Nam còn thiếu rất nhiều những nhà phân tích (analyste). Đó là những chuyên gia tin học có thể phân tích (tìm hiểu, khảo sát...) sự hoạt động của các xí nghiệp, doanh nghiệp, các tổ chức hành chính xã hội... để thiết kế các hệ thống Tin học phục vụ công tác quản lý trong mọi lĩnh vực.*

*Môn học «Các phương pháp phân tích và thiết kế hệ thống thông tin » (Information Systems Analysis and Design Methods), hay gọn hơn, « Phân tích và thiết kế hệ thống », đóng vai trò quan trọng trong quá trình đào tạo những cán bộ phân tích nói trên.*

## CHƯƠNG 1

# Khái niệm về hệ thống thông tin quản lý

---

## I. Khái niệm về hệ thống

### I.1. Định nghĩa hệ thống

Thuật ngữ *hệ thống* (system) là một khái niệm rộng và được định nghĩa rất nhiều cách khác nhau. Trong cuộc sống hàng ngày, con người tiếp xúc với những hiện tượng, những sự kiện, những hoạt động..., tất cả đều nhắc tới, hoặc liên quan tới thuật ngữ hệ thống.

Ví dụ :

1. Hệ thống nước sinh hoạt ở thành phố, hệ thống điện lưới, hệ thống dịch vụ mua bán hàng, hệ thống điện thoại, hệ thống nhà ở...
2. Hệ thống xã hội, hệ thống tổ chức, hệ thống tư tưởng, hệ thống chính trị, hệ thống kinh tế, hệ thống xí nghiệp, hệ thống đường sắt...
3. Hệ thống thiên nhiên, hệ thống thần kinh, hệ thống triết học, hệ thống máy tính, hệ thống thông tin...

Có nhiều định nghĩa về hệ thống :

Từ điển Tiếng Việt 1997 định nghĩa hệ thống :

Tập hợp nhiều yếu tố, đơn vị cùng loại hoặc cùng chức năng, có quan hệ hoặc liên hệ với nhau chặt chẽ, làm thành một thể thống nhất

Tập hợp những tư tưởng, những nguyên tắc, quy tắc liên kết với nhau một cách logic, làm thành một thể thống nhất

Từ điển Larousse 1995 định nghĩa hệ thống là :

Tập hợp có thứ tự của những tư tưởng khoa học hay triết học

Tập hợp các cơ quan hay các cấu tạo có cùng bản chất cùng chức năng

Tập hợp các thành phần được xác định bởi những quan hệ qua lại giữa chúng

V.v...

Tuy nhiên, định nghĩa *hệ thống như một tập hợp các phần tử tác động qua lại lẫn nhau* là phổ biến nhất.. Hệ thống còn bao hàm ý nghĩa về kế hoạch, phương pháp, tổ chức các đối tượng một cách có trật tự để tạo thành một chỉnh thể.

Với mỗi hệ thống, một tính chất vượt trội lên tất cả được gọi là “tính trội” (emergence) mà khi một phần tử nào đó đứng riêng sẽ không thể có được. Tính trội là một trong những hình thức biểu hiện của nguyên lý biện chứng : “những sự thay đổi về lượng dẫn đến những sự thay đổi về chất”.

Như vậy đối nghịch với hệ thống là sự hỗn loạn (chaos), là trạng thái mà mọi phần tử không tuân theo một quy luật nào. Một cách tổng quát :

*Hệ thống là tập hợp các phần tử hay đối tượng (M) trên đó thực hiện một hay nhiều quan hệ (R) cho trước với những tính chất (P) nhất định.*

Từ định nghĩa sau, có thể phân loại hệ thống theo nhiều cách khác nhau theo tính chất P, các quan hệ R và các đối tượng M.

## I.2. Tính chất của hệ thống

Một hệ thống thường có ba tính chất cơ bản :

Tính chất 1 :

*Mối quan hệ giữa các phần tử có tính tác động qua lại ảnh hưởng với nhau*

Tính chất 2 :

*Mọi sự thay đổi về lượng hay về chất của một phần tử nào đó đều làm ảnh hưởng tới phần tử khác của hệ thống. Ngược lại, mọi sự thay đổi về lượng hay về chất của hệ thống đều có thể làm ảnh hưởng đến các phần tử của hệ thống đó*

Tính chất 3 :

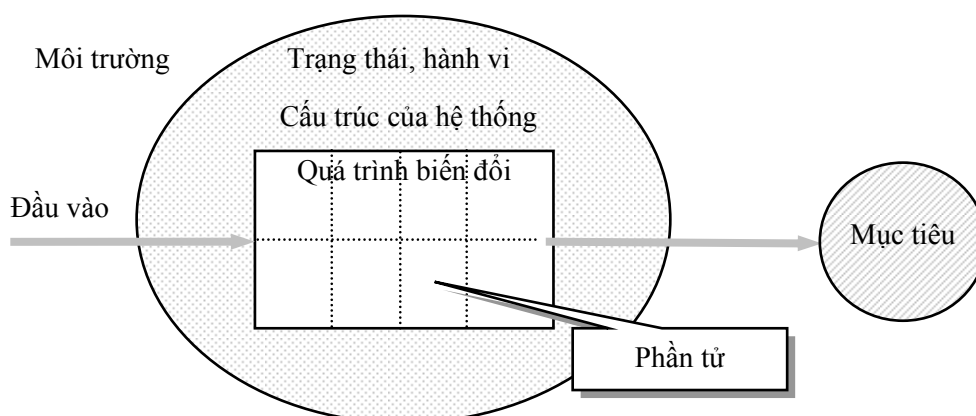
*Khi sắp xếp các phần tử của hệ thống theo một cách nào đó, hệ thống sẽ có tính trôi, đó là khả năng mà một phần tử đứng riêng sẽ không thể tạo ra được*

## I.3. Các thành phần cơ bản của hệ thống

Một hệ thống có thể được biểu diễn bởi nhiều thành phần, gồm :

1. Các phần tử
2. Môi trường của hệ thống
3. Các đầu vào và đầu ra
4. Trạng thái và hành vi
5. Cấu trúc
6. Mục tiêu

Hình dưới đây mô tả các thành phần của một hệ thống.



Hình 1.1 Các thành phần của hệ thống

### a) *Phần tử của hệ thống*

Phần tử là thành phần nhỏ nhất, có tính độc lập tương đối. Mỗi phần tử đều có những thuộc tính riêng và có thể được biểu diễn bởi một biến số (hay đại lượng biến thiên). Để hiểu về một hệ thống cần phải biết trạng thái của các phần tử và mối liên hệ giữa chúng.

### b) *Môi trường của hệ thống*

Môi trường của hệ thống là những gì nằm ngoài hệ thống nhưng liên quan đến việc thực hiện mục tiêu của hệ thống. Giữa hệ thống và môi trường có các tác động tương hỗ và những ranh giới. Nghiên cứu hệ thống kèm theo việc nghiên cứu môi trường. Những yếu tố bất lợi của môi trường làm ảnh hưởng đến việc thực hiện mục tiêu của hệ thống được gọi là *nhieều*.

### c) *Đầu vào và đầu ra của hệ thống*

Đầu vào là bất kỳ những gì mà môi trường có thể tác động vào hệ thống. Đầu ra là bất kỳ những gì mà hệ thống có thể tác động trở lại môi trường. Để làm tăng hiệu quả hoạt động của hệ thống, cần thỏa mãn ba yếu tố :

Chọn đầu vào và đầu ra hợp lý trong những điều kiện cụ thể

Thời gian biến đổi đầu vào thành đầu ra hợp lý

Hình thức hay phương pháp biến đổi hợp lý

### d) *Trạng thái của hệ thống*

Giả sử các phần tử của hệ thống được biểu diễn bởi các biến  $q_1, q_2, \dots, q_n$ , và hệ thống được biểu diễn bởi vector  $Q$  :

$$Q = (q_1, q_2, \dots, q_n)$$

Các biến  $q_i, i = 1..n$ , thay đổi theo thời gian :

$$q_i = q_i(t)$$

Khi đó trạng thái của hệ thống được biểu diễn bởi giá trị  $Q$  là bộ giá trị của các biến tại một thời điểm  $t$  :

$$Q(t) = (q_1(t), q_2(t), \dots, q_n(t))$$

Tại thời điểm  $t = 0$  là trạng thái ban đầu của hệ thống. Khi  $t$  biến thiên, vector hàm  $Q(t)$  xác định quỹ đạo hành vi của hệ thống.

Nếu tồn tại một số biến không thay đổi, hay thay đổi không đáng kể trong khoảng thời gian đang xét, thì những biến đó được gọi là các tham số của hệ thống và được ký hiệu bởi một vector :

$$a = (a_1, a_2, \dots, a_k)$$

Khi đó, quỹ đạo hành vi của hệ thống  $Q$  là vector hàm hai biến  $Q = Q(a, t)$ .

### **1.3.2. Hành vi của hệ thống**

Hành vi của hệ thống là tập hợp các đầu ra có thể của hệ thống trong một khoảng thời gian xác định. Khi hệ thống là đóng, có nghĩa hệ thống tách biệt với môi trường bên ngoài, các phần tử không biến đổi theo thời gian, khi đó hành vi của hệ thống được xác định bởi trạng thái ban đầu. Hành vi của hệ thống sẽ thay đổi khi các phần tử và mối liên hệ giữa chúng thay đổi, khi đó vector hàm  $Q(t)$  được xác định như sau :

$$Q(t) = f(Q(0), a, t)$$

Bây giờ xét tác động qua lại giữa hệ thống và môi trường, gọi  $X(t)$  là các biến đầu vào theo thời gian, hành vi của hệ thống được mô tả bởi hệ thức :

$$Q(t) = y(Q(0), X(t), a)$$

Trạng thái của đầu ra được mô tả bởi hệ thức :

$$Y(t) = F(Q(0), X(t), a)$$

Trong đó  $X = (X_1, X_2, \dots, X_n)$  là biến đầu vào,  $Y = (Y_1, Y_2, \dots, Y_m)$  là biến đầu ra.

### 1.3.3. Mục tiêu của hệ thống

Là trạng thái mong đợi, cần đạt được của hệ thống sau một khoảng thời gian hoặc tại một thời điểm nhất định nào đó.

Bên trong hệ thống, mỗi phần tử cũng có mục tiêu riêng. Những mục tiêu riêng có thể thống nhất hoặc không thống nhất với mục tiêu chung của hệ thống.

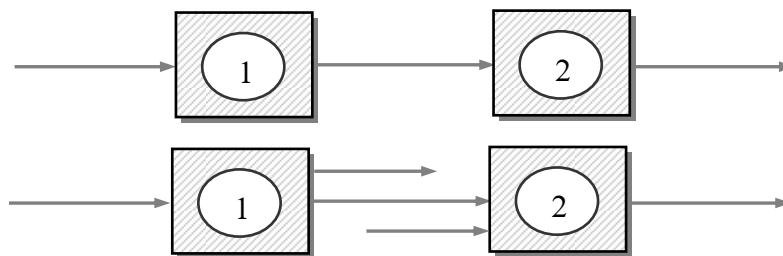
### 1.3.4. Cấu trúc của hệ thống

Cấu trúc là yếu tố bất biến của hệ thống, liên quan đến hình thức tổ chức hệ thống. Đó là cách sắp đặt bố trí hay ghép các phần tử và cách xác định mối quan hệ giữa chúng theo một dấu hiệu hay tiêu chuẩn nào đó.

Có nhiều cách tổ chức hệ thống khác nhau. Về cơ bản, có 3 cách ghép là *ghép nối tiếp*, *ghép song song* và *ghép có mối liên hệ ngược*.

#### a) Ghép nối tiếp

Ghép nối tiếp là đầu vào của phần tử này là đầu ra của phần tử kia và ngược lại.

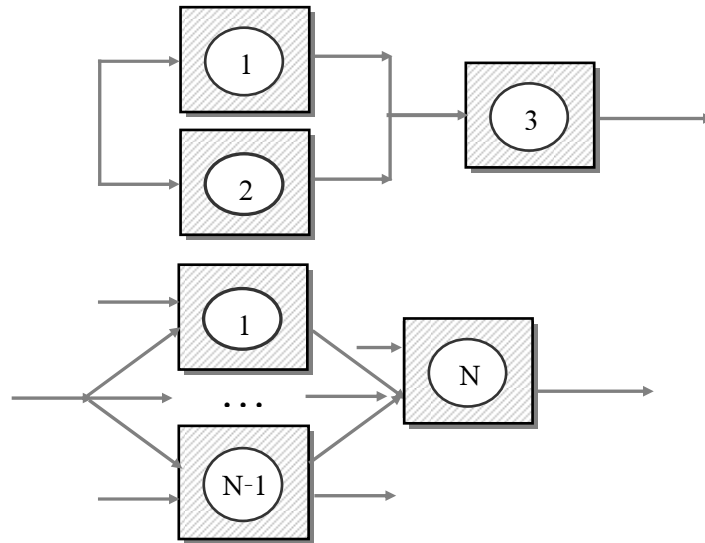


Hình 1.2 Ghép nối tiếp các phần tử

Phương pháp ghép nối tiếp đơn giản, rõ ràng nhưng độ tin cậy kém. Khi số lượng phần tử tăng lên thì độ tin cậy giảm xuống. Hệ thống chỉ làm việc tốt khi tất cả các phần tử đều làm việc tốt.

#### b) Ghép song song

Là cách ghép mà đầu vào của một phần hay toàn bộ các phần tử cùng chung một biến số vào mà đầu ra của chúng lại là đầu vào của một phần hay nhiều phần tử khác của hệ thống.

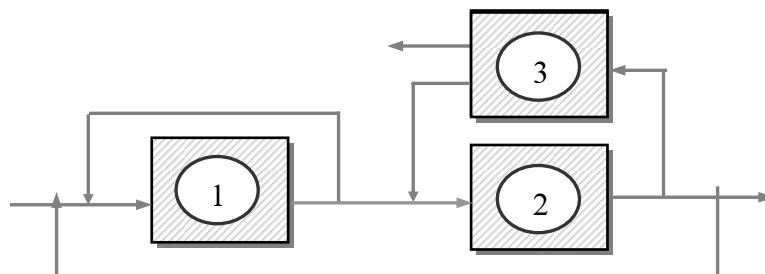


Hình 1.3 Ghép song song các phần tử

Phương pháp ghép song song có độ tin cậy cao vì hệ thống chỉ ngừng trệ khi toàn bộ các phần tử ngừng trệ. Tuy nhiên cách ghép nối này làm tăng mối quan hệ cho nên tính phức tạp của hệ thống cũng tăng lên.

#### c) Ghép có mối liên hệ ngược

Ghép có mối liên hệ ngược là một dạng kết hợp các phần tử. Trong cách ghép này, đầu ra của một phần tử lại có thể là đầu vào của chính phần tử đó, được thực hiện trực tiếp hay thông qua những phần tử khác của hệ thống.



Hình 1.4 Ghép có mối liên hệ ngược

Liên hệ ngược dương làm tăng thêm tác động tích cực của đầu vào, trái lại, liên hệ ngược âm sẽ làm giảm tính tích cực của đầu vào.

## I.4. Phân loại hệ thống

Lý thuyết hệ thống chia ra nhiều loại hệ thống như sau :

1. Hệ thống con hay phân hệ (hệ thống thứ yếu)
2. Hệ thống lớn
3. Hệ thống đóng và hệ thống mở
4. Hệ thống tĩnh và hệ thống động
5. Hệ thống trừu tượng và hệ thống cụ thể
6. Hệ thống bảo trì trạng thái
7. Hệ thống có chủ định
8. Những hệ thống tìm kiếm mục tiêu đa dạng

## **I.5.Nghiên cứu lý thuyết hệ thống**

### ***I.5.1.Lý thuyết tổng quát về hệ thống***

Lý thuyết tổng quát tiếp cận hệ thống bởi 9 mức độ hay trình độ tăng dần theo độ phức tạp và tính trừu tượng hoá.

- Mức 1 Là mức tĩnh, mức của những nền tảng. Sự nghiên cứu chính xác mức này là cơ sở của các mức sau cao hơn.
- Mức 2 Là mức của hệ thống đơn giản bằng cách xem những đối tượng tĩnh là động, giống như cơ cấu hoạt động của đồng hồ.
- Mức 3 Hệ thống được đưa vào các cơ chế kiểm soát hay điều khiển giống như cơ chế của máy điều hoà nhiệt độ. Lúc này hệ thống không có tính chất quân bình ổn định mà có sự chuyển giao và tiếp nhận của thông tin.
- Mức 4 Hệ thống được xem là mở và ở trạng thái bảo toàn, còn được gọi là trình độ của tế bào.
- Mức 5 Là trình độ xã hội phát sinh ở mức độ thực vật, có tính phát triển nhưng chưa có tính thực tiễn.
- Mức 6 Là trình độ thế giới động vật có đặc tính di động và khả năng tiếp nhận với sự hoạt động của hệ thống thần kinh.
- Mức 7 Là trình độ con người có ý thức, mang tính cá biệt. Mức độ này gắn liền với khả năng trao đổi ngôn ngữ và sử dụng ký hiệu, vượt lên khỏi giới động vật thuần tuý.
- Mức 8 Là trình độ tổ chức xã hội với sự hoạt động phong phú của khoa học nghệ thuật và tình cảm con người.
- Mức 9 Là trình độ của hệ thống tổ chức bậc cao, mang tính phát triển và thích nghi với môi trường.

### ***I.5.2.Quan điểm nghiên cứu hệ thống***

Nghiên cứu hệ thống phải dựa trên nền tảng khoa học, hiện thực và có hiệu quả. Đó là phải :

Tôn trọng mối quan hệ biện chứng giữa vật chất và ý thức.

Thừa nhận các hiện tượng luôn có sự tác động qua lại và chi phối lẫn nhau.

Thừa nhận các sự vật luôn luôn biến đổi không ngừng.

Các sự vật hiện tượng phát triển nhờ động lực nội tại của chúng.

Người ta thường sử dụng 3 phương pháp :

Phương pháp mô hình hoá.

Phương pháp hộp đen.

Phương pháp tiếp cận (phân tích).

#### ***a) Phương pháp mô hình hoá***

Phương pháp này đòi hỏi phải *biết cả ba yếu tố là đầu vào, đầu ra và cấu trúc* của hệ thống. Ưu điểm là dễ thực hiện, thời gian nghiên cứu ngắn và chi phí thấp. Nhược điểm là dễ gây hiểu sai và ngộ nhận, từ đó dẫn đến bảo thủ và cố chấp.

#### ***b) Phương pháp hộp đen***



Phương pháp được sử dụng khi *biết đầu vào và đầu ra nhưng chưa biết cấu trúc bên trong* của hệ thống. Quá trình nghiên cứu xác định mối quan hệ giữa đầu vào và đầu ra để tìm ra những quy luật hoạt động hay những cấu trúc hành vi của hệ thống.

Các bước nghiên cứu như sau :

Quan sát những yếu tố đầu vào X và ghi nhận những yếu tố của đầu ra Y.

Từ các cặp trạng thái (X, Y) tìm ra những quy luật hay những cấu trúc có thể có của hệ thống.

Tiến hành kiểm tra mặt thực tiễn của những cấu trúc giả định để từng bước hoàn thiện.

Chỉnh lý sửa đổi các kết quả, hoàn thiện cấu trúc và đem vào áp dụng thực tiễn.

### c) *Phương pháp tiếp cận hay phân tích hệ thống*

Phương pháp được sử dụng khi không biết gì về hệ thống, *chỉ biết được mục tiêu* của hệ thống mà thôi. Người ta chia hệ thống ra thành các hệ thống con có mối liên hệ ràng buộc lẫn nhau, từ đó tìm ra quy luật hoạt động của các hệ thống con để khái quát lên thành quy luật hoạt động của cả hệ thống.

Tư tưởng chủ đạo của phương pháp gồm 3 yếu tố :

Cái gì cần khảo sát và nghiên cứu ?

Phải giải quyết những vấn đề gì (phải làm thế nào ?)

Hệ thống làm việc như thế nào ?

Những đòi hỏi của phương pháp :

Chọn lựa kỹ lưỡng tiêu chuẩn, cách thức phân chia hệ thống ban đầu thành các hệ thống con

Chú ý tính trôi của hệ thống, không để làm lu mờ hoặc làm mất đi.

Xác định rõ các mối quan hệ khi phân chia vì mỗi hệ thống con lại có thể tiếp tục được phân chia thành các phân hệ nhỏ hơn.

Nghiên cứu hệ thống trong mối tương quan giữa hệ thống với môi trường theo quan điểm hệ thống là mở.

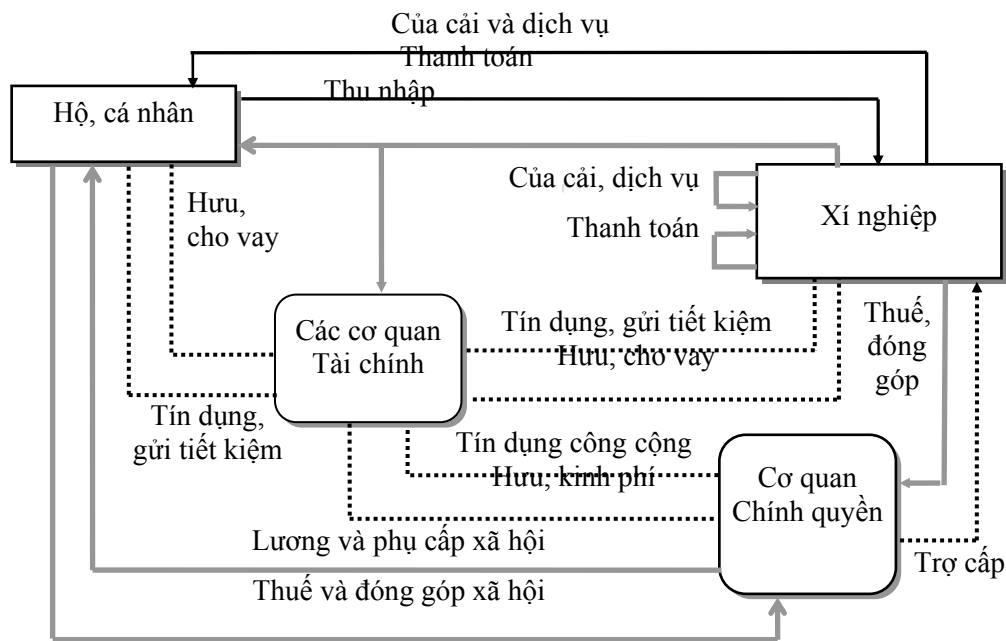
Quan sát hệ thống dưới nhiều góc độ để tìm ra những khía cạnh khác nhau của cơ cấu và hành vi của hệ thống.

Phương pháp phân tích hệ thống hay được sử dụng để nghiên cứu các hệ thống phức tạp. Để triển khai được phương pháp, đòi hỏi người phân tích phải có những trình độ hiểu biết và kiến thức nhất định, biết chủ động sáng tạo.

## II. Xí nghiệp và vai trò của xí nghiệp trong nền kinh tế

### II.1.Xí nghiệp và các tổ chức bên trong

Trong quản lý kinh tế, các xí nghiệp (XN) là những đơn vị có cơ cấu cơ bản trong hệ thống sản xuất vật chất. Hoạt động hiệu quả của XN có vai trò thúc đẩy nhịp độ phát triển của nền kinh tế quốc dân. Sơ đồ tổng quát dưới đây thể hiện chu trình kinh tế của XN với một số tác nhân bên ngoài XN :



Hình 1.5 Chu trình kinh tế của xí nghiệp

Các XN là những hệ thống được tổ chức sắp xếp theo đặc thù về chuyên môn kỹ thuật, công nghệ... Cách tổ chức của XN quyết định phạm vi hoạt động, các mục tiêu và chức năng của XN, tạo ra cho nó tính tự quản về tổ chức trong một cơ cấu phân cấp (theo ngành, theo Bộ...).

Các yếu tố sản xuất (chỗ làm việc, dây chuyền công nghệ, xưởng, phân xưởng...) và các yếu tố phụ trợ (kho tàng, phòng thí nghiệm...) được tổ chức theo những tiêu chuẩn riêng làm phong phú đa dạng hệ thống sản xuất nhưng cũng phản ánh tính đồng nhất của các XN. Đó là một mục tiêu và cùng nằm trong một cơ cấu quản lý.

Những tác nhân bên ngoài của XN là các nhà thầu, các XN khác, các đại lý, cơ quan chính quyền, các cơ sở tài chính trung gian, các khách hàng trực tiếp... tạo thành một môi trường của XN. Môi trường tác động tương hỗ với sự hoạt động bên trong của XN. Căn cứ vào sự hoạt động trao đổi này, mỗi XN đều có những quyết định mang tính chiến lược về sản xuất, tài chính, thương mại...

Thông thường, XN được tổ chức phân cấp theo chức năng sản xuất, kinh doanh hoặc vị trí địa lý thành các đơn vị, phòng ban, phân xưởng...

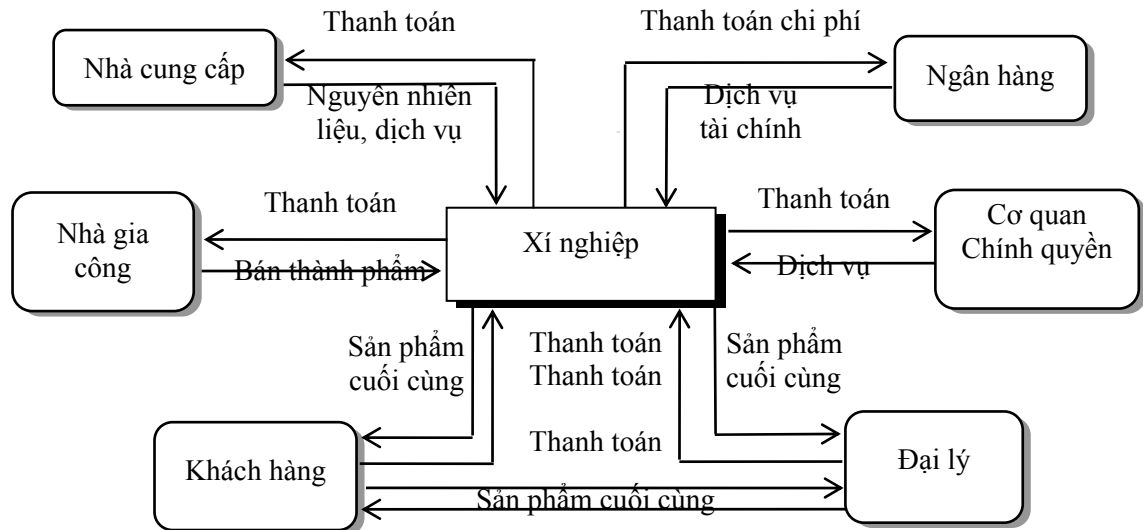
Các lĩnh vực quản lý (quản lý tài chính kế toán, quản lý thương mại, quản lý sản xuất...) được đặt lên phía trên của hệ thống các đơn vị này.

Mỗi đơn vị lại được phân ra thành các bộ phận nhỏ hơn. Ví dụ phòng Kế toán-Tài chính có thể gồm các bộ phận kế toán công nợ, kế toán vật tư, kế toán tài sản cố định và thiết bị...

### II.1.1. Liên hệ giữa xí nghiệp với môi trường

XN tạo thành một *hệ thống mở* (open system) đối với môi trường. Các phần tử trong hệ thống (nguồn nhân lực, vật chất...) một mặt tương tác với nhau, một mặt tương tác với bên ngoài (cung ứng vật tư, buôn bán...). Các XN là những hệ thống sống và phát triển, vì vậy mặt động là cơ bản.

Tập hợp gồm XN và môi trường tạo thành một *siêu hệ thống* (meta-system) được chỉ ra như hình dưới đây :



Hình 1.6 Xí nghiệp và môi trường kinh tế trực tiếp của xí nghiệp

### II.1.2. Phân tích các liên hệ với môi trường

Mối liên hệ giữa XN và môi trường được biểu diễn bởi các dòng (flux). Các dòng đi từ bên ngoài vào XN và đi từ XN ra lại môi trường. Lại có các dòng tồn tại bên trong XN. Có 4 loại dòng :

Dòng của cải (nguyên liệu, nhiên liệu, sản phẩm cuối cùng)

Dòng dịch vụ (cho vay tiền, tư vấn, bảo trì...)

Dòng tiền tệ (thanh toán với khách hàng hoặc với người cung cấp vật tư)

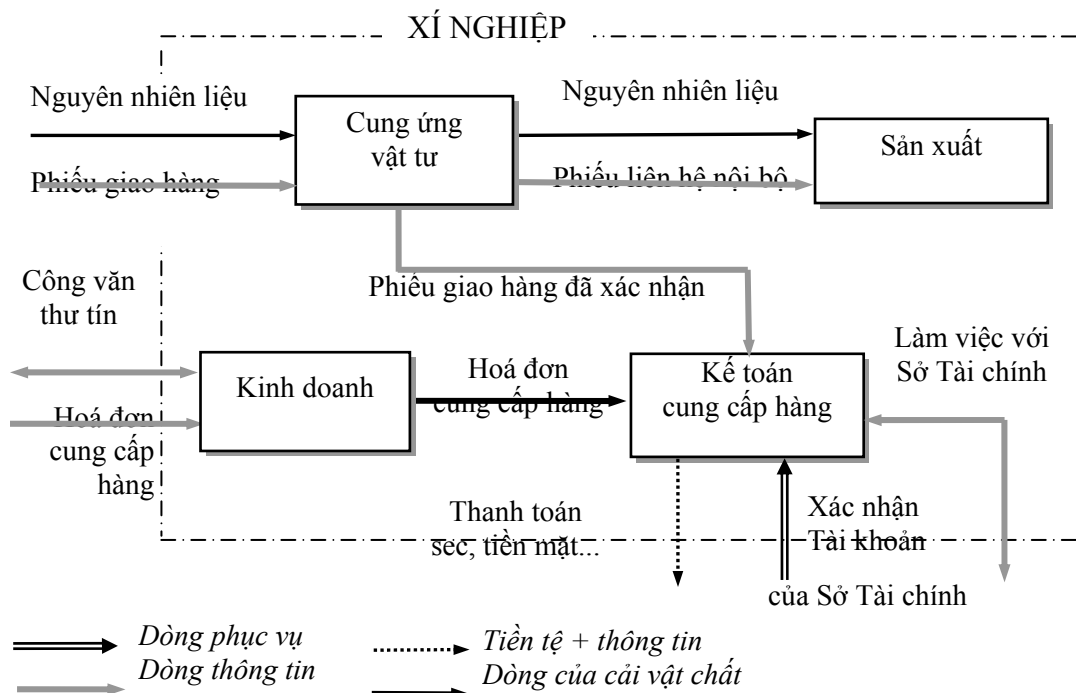
Dòng thông tin (ghi chép, thông báo, quảng cáo...)

Tập hợp các dòng xuất phát từ các quyết định của XN. Ví dụ :

- Tiếp nhận nguyên vật liệu để sản xuất sau khi phòng Vật tư thảo đơn đặt hàng và được ban giám đốc thông qua.
- Thanh toán khách hàng sau khi gửi sản phẩm + hoá đơn giao hàng, v.v...

Tập hợp các đơn vị trao đổi với nhau thông qua các dòng thông tin và dòng của cải vật chất nhằm đáp ứng các nhu cầu của XN.

Bốn loại dòng nói trên được biểu diễn như sau :



Hình 1.7 Xí nghiệp liên hệ với môi trường bởi bốn loại dòng

Sự tồn tại dòng của cái vật chất dẫn đến sự có mặt, ở đầu dòng và cuối dòng, các dòng thông tin hình thức hoặc phi hình thức.

Ví dụ tại nhà máy bia-nước ngọt Đà Nẵng, dòng vật chất là các loại chai do nhà máy thủy tinh Hoà khánh cung cấp, ta sẽ gặp những dòng thông tin sau :

Không chính thức : trao đổi điện thoại hoặc bằng miệng...

Chính thức : thư, fax báo giá hoặc các phiếu đặt hàng, giao nhận hàng...

Nghiên cứu hoạt động của các dòng có vai trò quan trọng trong việc thể hiện cấu trúc ba hệ thống của XN : hệ thống quyết định, hệ thống tác nghiệp và hệ thống thông tin.

## II.2.Hệ thống tổ chức xí nghiệp

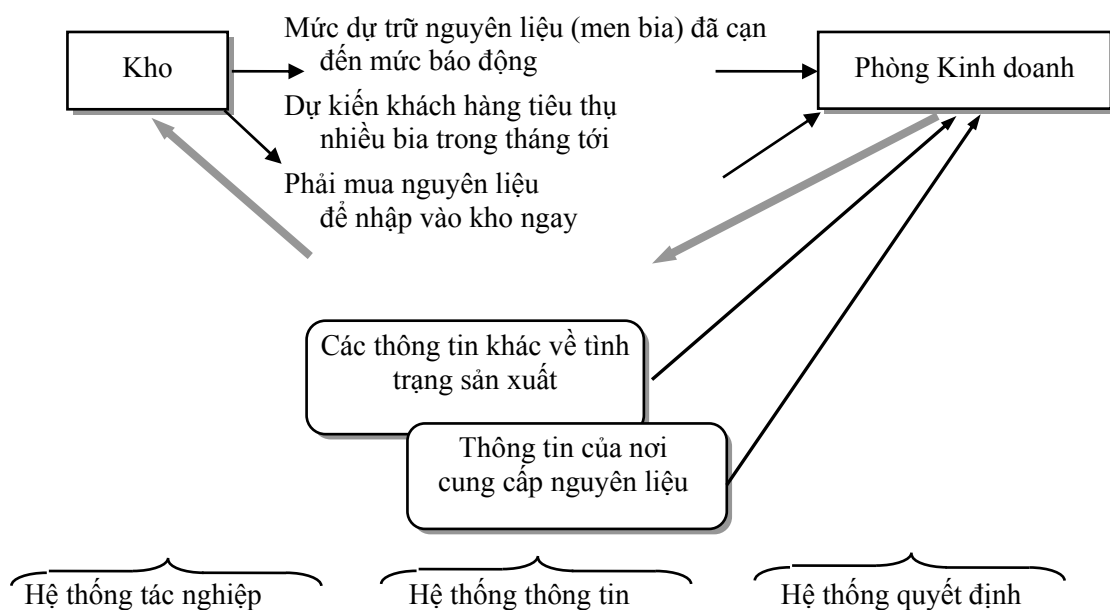
Sử dụng phương pháp của K.Boulding để tiếp cận hệ thống là tổ chức XN bằng cách khảo sát các loại dòng để làm rõ các tương tác bên trong một hệ thống và tương tác của hệ thống với môi trường.

Theo K.Boulding, có 9 mức để tiếp cận hệ thống :

- Mức 1 Bắt đầu, người ta xem xét một đối tượng nào đó như là tĩnh, độc lập với tất cả các đối tượng khác. Ví dụ : xem xét đối tượng là bia chai, giả sử có mã là 2453 trong danh mục các sản phẩm.
- Mức 2 Đối tượng được xem như là động, chẳng hạn đối tượng xuất hiện trong đơn đặt hàng của một đại lý nào đó.
- Mức 3 Người ta đưa vào các hiện tượng điều tiết : đơn đặt hàng chỉ được giải quyết đúng thời hạn khách hàng đề nghị nếu như kế hoạch sản xuất tại XN đảm bảo được.
- Mức 4 Xem xét các thông tin liên quan đến đối tượng. Ví dụ về tình trạng dự trữ trong kho hàng, về tính chất của khách hàng sẽ phục vụ (có giả tiền đúng kỳ hạn không, v.v...).

- Mức 5 Xuất hiện khái niệm quyết định và khái niệm ghi nhớ : đơn đặt hàng có được chấp nhận không ?
- Mức 6 Không thể quyết định một cách ngẫu nhiên, phải căn cứ tình hình thực tại cũng như trong quá khứ đã ghi nhớ. Mục đích là vừa đảm bảo tính kinh tế của XN, vừa làm hài lòng khách hàng.
- Mức 7 Giai đoạn làm rõ (xem hình dưới đây), gồm các yếu tố :
- Cấu trúc ba hệ thống : hệ thống quyết định (decision system), hệ thống thông tin (information system) và hệ thống tác nghiệp (operation system).
  - Sự tương tác cần thiết giữa các hệ thống.
  - Tính quan trọng của hệ thống thông tin :  
Giữ thông tin của hai hệ thống kia và của môi trường.  
Băng truyền giữa hai hệ thống kia và của môi trường.
- Mức 8, 9 Tính đến sự phức tạp của hệ thống bằng cách xem xét :
- Tính độc lập của quyết định : hệ thống quyết định được phân chia thành hệ thống mệnh lệnh và hệ thống tổ chức.
  - Khả năng của hệ thống quyết định đối với các dự án, tự điều chỉnh (khi kho nguyên liệu cạn đến mức báo động...).

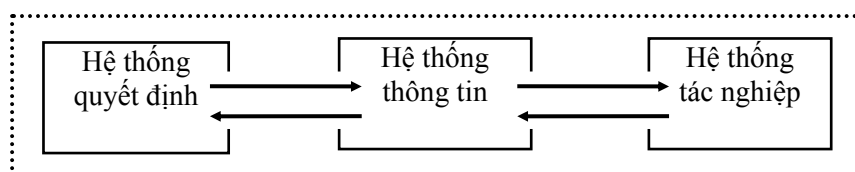
Ví dụ :



Hình 1.8 Tiếp cận và làm rõ cấu trúc ba hệ thống của xí nghiệp

### II.3. Ba hệ thống của một tổ chức xí nghiệp

Người ta quan niệm XN được tạo thành từ ba hệ thống con : *hệ thống tác nghiệp*, *hệ thống quyết định* (hay hệ thống lãnh đạo) và *hệ thống thông tin* :



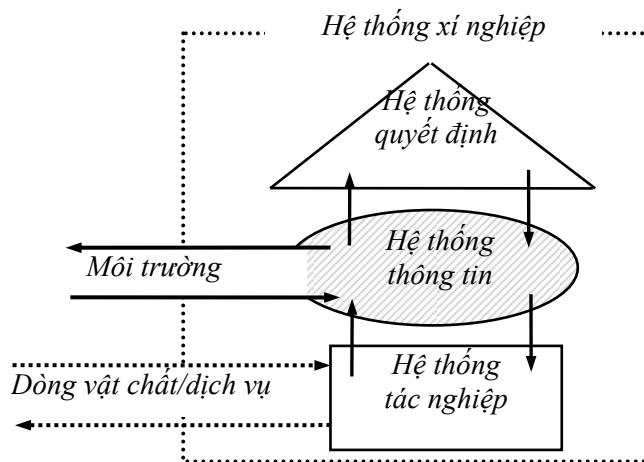
Hình 1.9 Cấu trúc ba hệ thống của xí nghiệp

### a) Hệ thống tác nghiệp

Hệ thống tác nghiệp liên quan đến mọi hoạt động sản xuất, tìm kiếm khách hàng mới. nhằm đạt được mục tiêu do hệ thống quyết định đưa ra. Hệ thống tác nghiệp gồm các nhân lực và phương tiện (máy móc, thiết bị, dây chuyền công nghệ...) có tác động tương hỗ với nhau để đáp ứng mục tiêu.

### b) Hệ thống quyết định

Hệ thống quyết định ảnh hưởng đến mọi hình thức quản lý XN, có tính chiến lược và tính chiến thuật. Tính chiến lược có tầm vực dài hạn hoặc trung hạn, thể hiện phương sách cải tiến sản xuất : thay đổi mẫu mã, nâng cao chất lượng nhưng lại hạ giá thành sản phẩm, tăng cường thị trường tiêu thụ của XN. Tính chiến thuật có tầm vực ngắn hạn, thể hiện ở các giải pháp tác nghiệp thường ngày, với mục đích nhằm thay đổi cách sử dụng thiết bị, nghiên cứu đáp ứng thị hiếu khách hàng, tiếp thị sản phẩm, v.v...

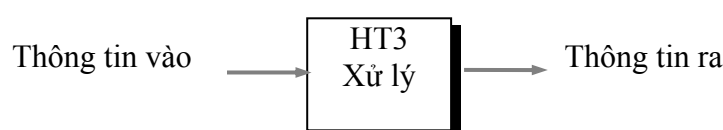


Hình 1. 10 Cách thể hiện khác cấu trúc ba hệ thống của tổ chức XN

### c) Hệ thống thông tin (HT3)

HT3 triển khai mối liên hệ giữa hệ thống tác nghiệp và hệ thống quyết định, đảm bảo sự hoạt động của XN và đạt được các mục tiêu đã đề ra. HT3 gồm các thành phần cơ bản sau đây :

1. *Con người* : Gồm những người sử dụng, người quản trị, người phát triển HT, v.v..., là yếu tố quyết định và can thiệp vào mọi quá trình phân tích, thiết kế và khai thác HT3.
2. *Dữ liệu* : Dữ liệu là thành phần cơ bản thể hiện cách nhìn tĩnh của HT3. Có thể xem đó là những “bức ảnh tĩnh” về thông tin có mặt trong HT3 mà người ta có thể có “chụp” được ở một thời điểm nào đó. HT3 có chức năng thu nhận, hợp thức hoá, tổ chức lưu trữ, khai thác và phân phối sử dụng dữ liệu.
3. *Quá trình xử lý* : Quá trình xử lý thể hiện mặt động của HT3. Xử lý biến đổi liên tục một cách tự động hay thủ công các dữ liệu có mặt HT3. Dữ liệu đến từ môi trường, sau khi được xử lý sử dụng có thể trở lại môi trường tạo thành các thông tin phục vụ sự hoạt động tác nghiệp của XN.



Hình 1.11 Xử lý thông tin

*Thiết bị và kỹ thuật* : Là nguồn tài nguyên (phần cứng và phần mềm) cho phép tiến hành quá trình xử lý dữ liệu. Thực tế, do chi phí đầu tư cao (nhân công, thời gian đặt hàng, lắp đặt thiết bị, nối mạng...), nên cần phải quan tâm đúng mức khi ước lượng giá thành.

Hiệu quả hoạt động của XN phụ thuộc vào chất lượng của HT3.

### **III. Hệ thống thông tin quản lý (HTTTQL)**

#### **III.1. Khái niệm HTTTQL**

HTTT nằm ở trung tâm của hệ thống tổ chức đang xét và là phần tử kích hoạt các quyết định (mệnh lệnh, chỉ thị, thông báo, chế độ tác nghiệp, v.v...). Do vai trò của HTTT trong lĩnh vực quản lý XN mà người ta nói đến HTTTQL (Management Information System). Một HTTTQL có thể được định nghĩa theo hai khía cạnh :

Khía cạnh thông tin và phương tiện truyền thông tin : “Tập hợp các thông tin luân chuyển trong XN và tập hợp các phương tiện, các thủ tục tìm kiếm, nắm giữ, ghi nhớ và xử lý thông tin”.

Khía cạnh mục đích chính đặt ra đối với XN : “Truyền đạt thông tin cho những người có liên quan (nhân viên) dưới dạng thích hợp và đúng đắn để đề ra quyết định hoặc cho phép thi hành một công việc.

HTTTQL cũng được định nghĩa theo cách khác : Là một hệ thống tích hợp “người-máy” tạo ra thông tin giúp con người trong sản xuất, quản lý và ra quyết định. HTTTQL sử dụng các thiết bị tin học, các phần mềm, cơ sở dữ liệu, các mô hình phân tích, lập kế hoạch, kiểm tra và ra quyết định.

HTTTQL được tiếp cận một cách tổng thể có logic trong một đơn vị mà không thể được nhìn nhận theo quan niệm chỉ một người sử dụng. Mỗi người sử dụng có cách nhìn riêng của mình về HTTTQL, tùy thuộc vào chức trách mà họ đảm nhận, kinh nghiệm nghề nghiệp, v.v... Chính vì vậy, HTTT của người sử dụng chỉ là một cái nhìn bộ phận trong thực tiễn.

#### **III.2. Cấu trúc của HTTTQL**

HTTTQL gồm 4 thành phần : các phân hệ hay hệ thống con (sub-systems), dữ liệu (data), mô hình (models) và các quy tắc quản lý (management rules).

##### ***III.2.1. Các phân hệ***

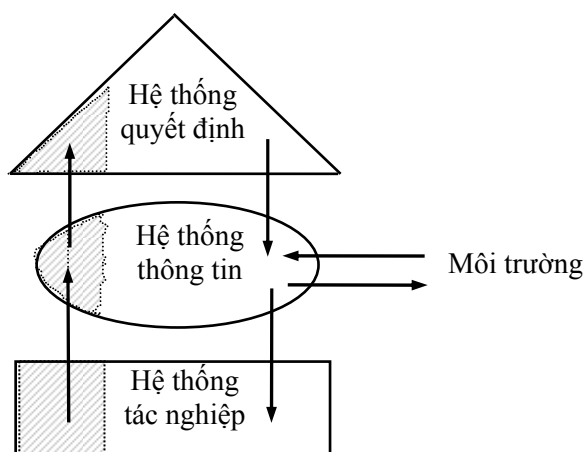
###### ***a) Định nghĩa phân hệ***

Phân hệ hay còn gọi là lĩnh vực quản lý (management domain) nhóm các hoạt động có cùng một mục tiêu trong nội bộ một đơn vị, như sản xuất, kinh doanh, hành chính, kế toán, nghiên cứu...

Người ta phân biệt 4 mức sau :

1. *Mức giao dịch* : các hoạt động thường nhật của XN.
2. *Mức tác nghiệp* : một số hoạt động thường nhật có thể đưa đến những quyết định ban đầu. Ví dụ cần có biện pháp xử lý thích hợp khi gặp trường hợp nợ đáo hạn của khách hàng.
3. *Mức chiến thuật* : ứng với các hoạt động đôn đốc, kiểm tra. Ví dụ theo dõi quy cách tiếp thị của nhân viên sau khi giao nhiệm vụ cho họ.
4. *Mức chiến lược* : đôn đốc kiểm tra sản xuất kinh doanh để duy trì sự phát triển lâu dài của XN.

Có cấu trúc giống hệ thống, phân hệ gồm một hệ thống tác nghiệp, một HTTT và một hệ thống quyết định như hình vẽ dưới đây (phần có gạch chéo).



Hình 1.12 Lĩnh vực quản lý là một phân hệ

Việc phân chia một hệ thống thông tin của một đơn vị thành các phân hệ cần tuân theo tính tổ chức và các quy tắc sau :

Biểu diễn một hoạt động của đơn vị có mục tiêu xác định.

Không dựa theo sự phân chia theo chức năng hoạt động của đơn vị tại một thời điểm nào đó, mà phải dựa trên tính hiệu quả và việc lựa chọn chiến thuật hay chiến lược.

Không căn cứ vào các mối liên hệ phân cấp vì các mối liên hệ này không phải lúc nào cũng mô tả các tình huống quản lý hay sản xuất.

Cần phân tích các dòng bên trong và bên ngoài trong quan hệ với môi trường. Để bảo đảm tính độc lập, phân hệ phải được xác định sao cho sự trao đổi thông tin với các lĩnh vực khác là tối thiểu.

*Ví dụ :*

Trong một XN, HTTT có thể gồm có các phân hệ :

Quản lý vật tư

Quản lý nguyên nhiên liệu

Quản lý tài sản cố định

Quản lý nhân sự - tiền lương

Quản lý tài chính kế toán

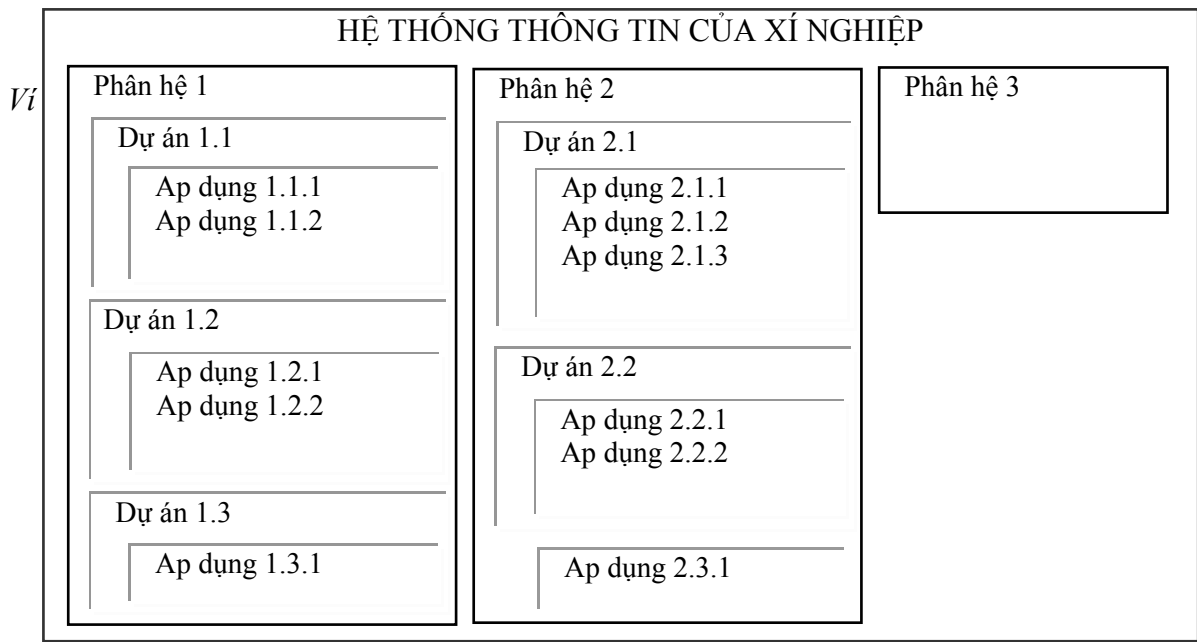
Quản lý tiêu thụ sản phẩm - công nợ khách hàng

Quản lý công văn thư tín hành chính

#### b) Sự phân chia thành các dự án và áp dụng

Một phân hệ, nếu như là một tập hợp độc lập với các phân hệ khác, thì vẫn còn là một khái niệm tương đối rộng. Vì vậy, khi Tin học hóa, cần tiếp tục phân chia các phân hệ thành các thành phần nhỏ hơn.





Quản lý nâng bậc

Quản lý khen thưởng, v.v...

Như vậy, mỗi phân hệ, hay lĩnh vực quản lý, đã được phân chia thành các hoạt động riêng rẽ và được xem như là các dự án. Mỗi dự án lại có thể tiếp tục được phân chia thành các áp dụng để dễ dàng Tin học hóa.

*Ví dụ :*

Dự án *Quản lý lương* trong phân hệ *Quản lý nhân sự tiền lương* có thể được phân chia thành các áp dụng :

Lương sản phẩm

Lương thời gian (hành chính)

Bảo hiểm xã hội

Phụ cấp nghề nghiệp...

### **III.2.2. Dữ liệu**

Dữ liệu (data) là cơ sở của thông tin. Nói đến thông tin là nói đến dữ liệu. Dữ liệu nhận giá trị trong một miền xác định.

*Ví dụ :*

Khách hàng (có tên là) *Đào*, (có địa chỉ là) *17 Lê Duẩn Đà Nẵng*.

Mặt hàng *bia chai Tiger*.

Ngày đặt hàng là *15/11/1998*, v.v...

Các dữ liệu trên đây mặc dù đã được cho giá trị cụ thể nhưng khó biết được mối liên hệ giữa như thế nào. Tuy nhiên, khi đặt chúng trong một mối liên hệ, người ta có được thông tin về yêu cầu đặt hàng của khách hàng.

*Ví dụ :*

*Cô Đào ngụ ở 17 Lê Duẩn Đà Nẵng đã đặt mua bia chai Tiger ngày 15/11/1998*

Như vậy, khái niệm dữ liệu hẹp hơn khái niệm thông tin. Thông tin luôn mang ý nghĩa và gồm nhiều giá trị dữ liệu.

Các dữ liệu có thể được biểu diễn dưới nhiều dạng khác nhau (chữ viết, lời nói...), thể hiện trên giấy (công văn, hóa đơn, thư, fax...) hoặc trên màn hình của máy vi tính, dễ hoặc khó sử dụng tùy theo tính chất hay hoàn cảnh thu nhận.

Có ba loại dữ liệu tương ứng với ba tình huống hay gặp khi thu nhận dữ liệu là tình huống chắc chắn, tình huống ngẫu nhiên và tình huống chưa biết trước :

*Ví dụ :*

Dữ liệu chắc chắn :

số ngày công của một công nhân trong tháng,

thuế suất áp dụng cho một mặt hàng...

Dữ liệu có tính ngẫu nhiên hay chưa chắc chắn :

doanh số dự báo theo phân tích thị trường,

thuế suất sẽ áp dụng cho một mặt hàng mới nhập...

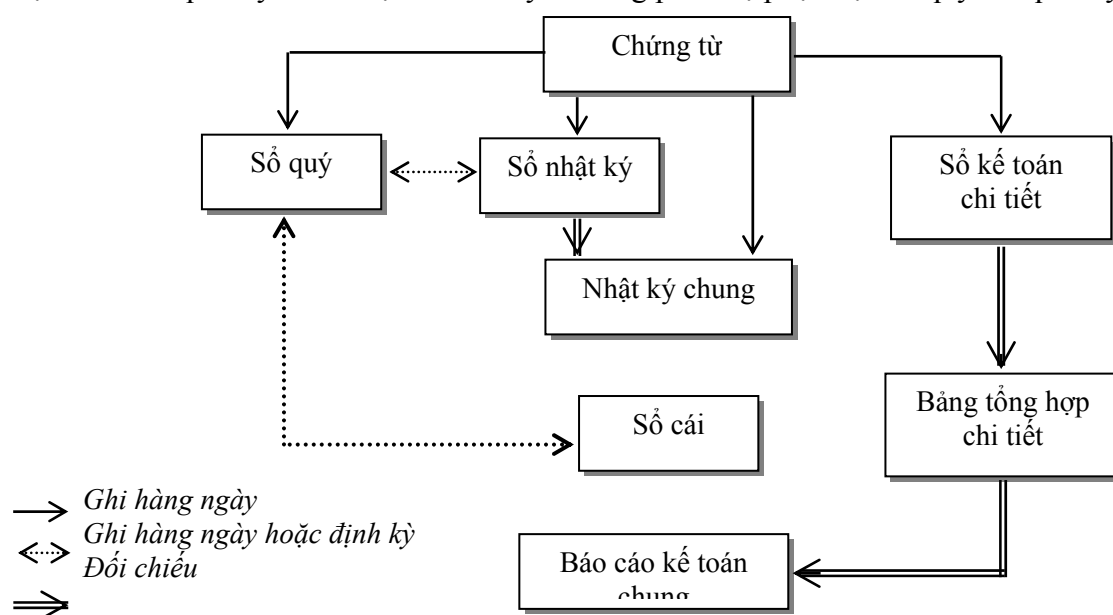
Dữ liệu chưa biết :

rủi ro do các sự cố sản xuất,

yếu tố con người...

### III.2.3. Mô hình quản lý

Mô hình quản lý bao gồm tập hợp các thủ tục, quy trình và phương pháp đặc thù cho mỗi phân hệ. Mô hình quản lý và dữ liệu luân chuyển trong phân hệ phục vụ các quy tắc quản lý.



Hình 1.14 Quy trình hạch toán kế toán

*Ví dụ :*

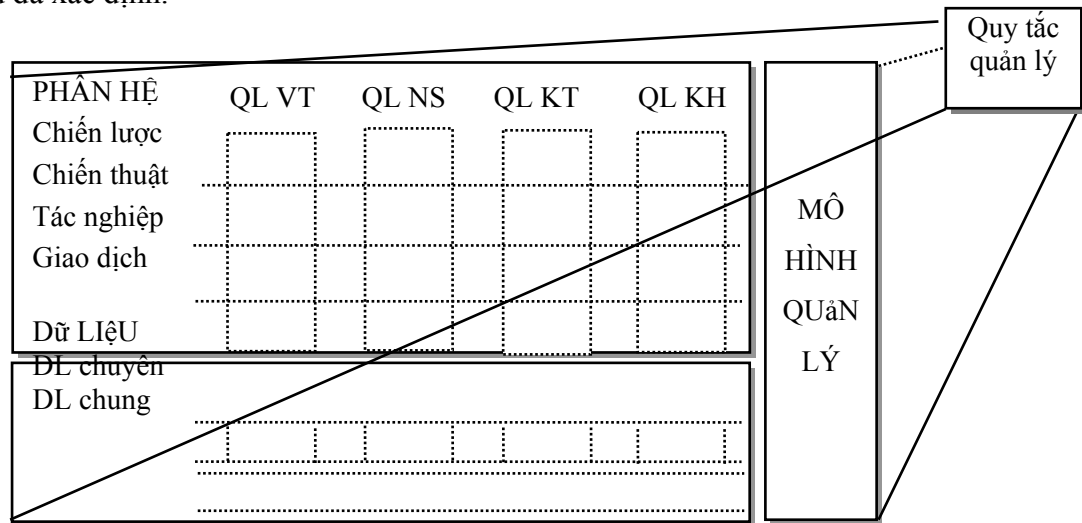
Kế hoạch lập báo cáo tài chính và các chế độ kế toán (chẳng hạn chế độ kế toán mới của Bộ Tài chính) trong phân hệ tài chính kế toán của công ty Xổ số Kiến thiết Đà Nẵng.

Quy trình sản xuất bia chai loại nhỏ sử dụng sản phẩm vỏ chai (theo dây chuyền thiết bị mới nhập của nhà máy thủy tinh) trong phân hệ quản lý sản xuất tại nhà máy bia-nước ngọt Đà Nẵng.

Chế độ quản lý nhập hàng và lưu kho trong phân hệ quản lý nhập cảng tại cảng Đà Nẵng, v.v...

### III.2.4. Quy tắc quản lý

Quy tắc quản lý, hay công thức tính toán, cho phép biến đổi hoặc xử lý dữ liệu phục vụ cho mục tiêu đã xác định.



Giải thích : QL KD quản lý vật tư QL NS quản lý nhân sự  
QL SX quản lý kế toán QL HC quản lý khách hàng

Hình 1.15 Mối liên hệ giữa các thành phần của HTTTQL

Ví dụ :

$$SL_{tồn\_cuối\_kỳ} = SL_{tồn\_đầu\_kỳ} + SL_{nhập\_trong\_kỳ} - SL_{xuất\_trong\_kỳ}$$

$$Số\_Tiền\_Ttoán = Giá\_Đơn\_vị \times Số\_lượng$$

$$Lương\_Tgian = (Lương\_CBản \times Số\_Ngày\_công) / 26$$

Sau đây là hình ảnh về mối liên hệ giữa quy tắc quản lý, các phân hệ, dữ liệu và mô hình quản lý.

### III.3.Vai trò và chất lượng của HTTTQL

HTTTQL phải có chức năng thu nhận, xử lý và phân phát thông tin đúng lúc đúng nơi nhận. Đồng thời HTTTQL phải được thiết kế sao cho XN quản lý tối ưu các nguồn thông tin.

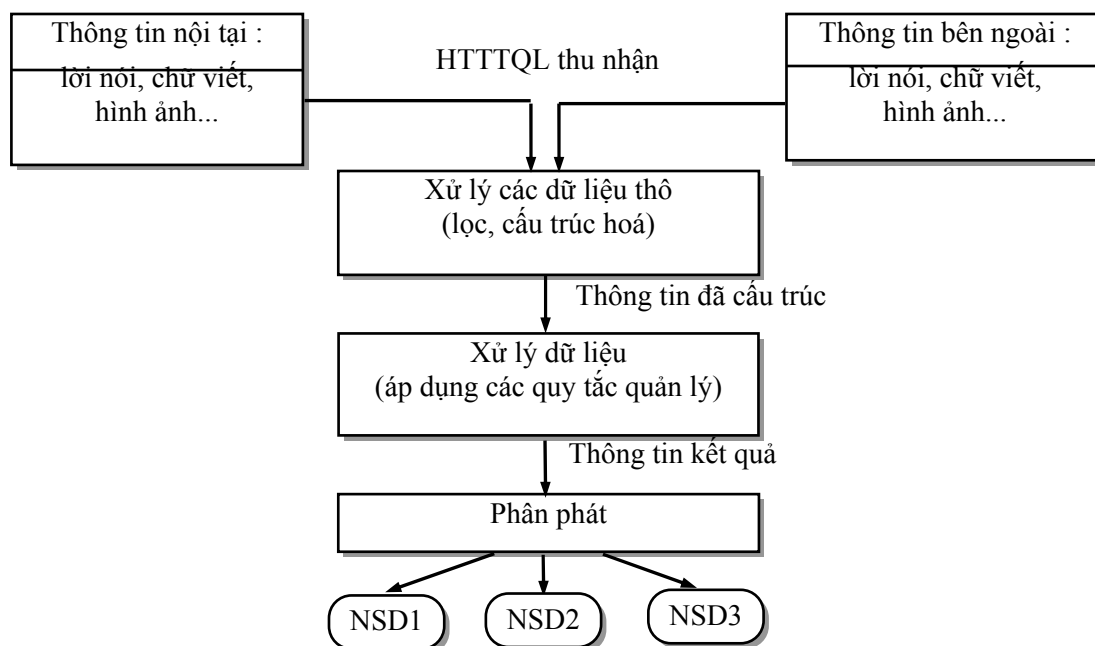
#### a) Các dạng thông tin

HTTTQL thu nhận nhiều dạng thông tin khác nhau :

**Thông tin nói :** là phương tiện liên lạc phổ biến giữa con người và giữa bất kỳ một tổ chức nào. Đặc trưng của thông tin nói là phi hình thức và rất khó xử lý bằng Tin học. Phương tiện liên lạc là điện thoại hoặc các máy móc nhận dạng tiếng nói.

**Thông tin viết :** là nguồn dữ liệu chính của HTTTQL trong bất kỳ một XN nào.

**Thông tin hình ảnh :** thu được từ các thông tin khác của hệ thống (chẳng hạn biểu đồ, đồ thị rút ra từ một bảng số liệu) hoặc từ các nguồn khác (ví dụ, ảnh chụp sản phẩm của hãng cạnh tranh, quảng cáo...).



Hình 1.16 Chức năng thu nhận, xử lý và phân phát thông tin

Các dạng thông tin khác : được cảm nhận từ vị giác, xúc giác, khứu giác và không xét trong HTTTQL, mặc dầu đôi khi chúng có vai trò quan trọng trong các ứng dụng công nghiệp.

#### b) Thông tin có cấu trúc

Giả sử rằng các thông tin vô ích hoặc có hại đã được loại bỏ thì những thông tin vừa được liệt kê ở trên là thành phần của HTTTQL của XN.

Một số thông tin trong chúng có thể khai thác ngay để đưa ra một quyết định (lập kế hoạch sản xuất, cải tiến trang thiết bị...), một số khác dùng để xử lý sơ bộ hoặc thủ công, hoặc tự động (ví dụ, thông tin trên các phim ảnh quảng cáo, hoặc trên đĩa từ, đĩa quang : CD-ROM...)

Việc xử lý tự động các thông tin chỉ có thể thực hiện được từ các dữ liệu có cấu trúc. Chính từ các thông tin có cấu trúc và sử dụng các quy tắc quản lý phù hợp (tính toán doanh thu hàng tháng, quản lý vốn, v.v...) mà thực hiện việc xử lý thông tin.

#### c) Chất lượng của HTTTQL

Chất lượng của HTTTQL phụ thuộc 3 yếu tố : tính nhanh chóng (speed - rapidity), tính tin cậy (reliability), tính toàn vẹn (integrity) và tính thích đáng (pertinence).

##### Tính nhanh chóng

HTTTQL xử lý thông tin quá khứ (lưu trữ) và hiện tại. HTTTQL phải giúp mỗi phần tử của XN có thông tin có ích và nhanh nhất có thể được. Tính nhanh chóng liên quan đến sự tiến bộ và sự phát triển của công nghệ mới về phần cứng, phần mềm (các bộ vi xử lý, mạng máy tính, Internet, cáp quang, v.v...).

##### Tính tin cậy

Để đảm bảo tính tin cậy, HTTTQL xử lý và phát hiện các thông tin sai lệch để chỉ luân chuyển các thông tin hợp thức. Các kết quả xử lý đưa ra luôn luôn đúng đắn, không phụ thuộc vào thời gian, điều kiện xử lý hoặc người xử lý...

#### *Tính toàn vẹn*

Yêu cầu về tính toàn vẹn của dữ liệu thường gặp trong các hệ quản trị cơ sở dữ liệu. Chẳng hạn dữ liệu phải được lưu giữ an toàn, NSD không được tự ý sửa đổi nếu không có quyền truy cập, v.v...

#### *Tính thích đáng*

HTTTQL thu nhận mọi thông tin đến nó và phải tìm được những thông tin cần thiết cho sự hoạt động của XN. Việc chỉ lựa chọn những thông tin thích đáng, không dư thừa là một trong những nguyên lý cơ bản của lĩnh vực phân tích và thiết kế.

### **III.4.HTTTQL - công cụ điều phối và kiểm soát hệ thống**

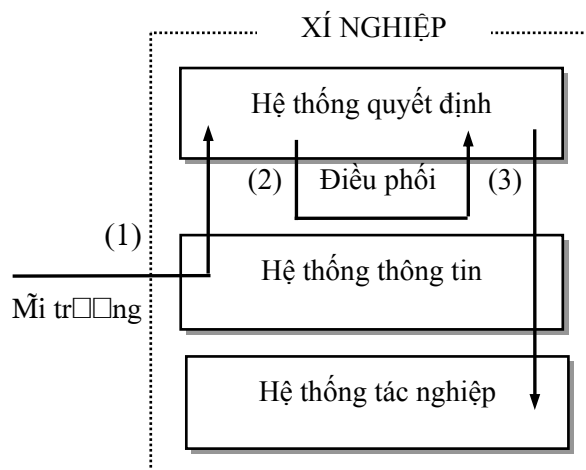
Nguồn gốc của HTTTQL nằm ở các mệnh lệnh tạo ra bởi hệ thống quyết định, đồng thời, HTTTQL là công cụ điều phối và kiểm soát hệ thống.

Hệ thống mệnh lệnh thu nhận mọi loại thông tin đến từ môi trường và bản thân XN để hoạt động. Có ba trường hợp như sau :

#### *a) Điều khiển theo “chu trình mở”*

Thông tin đến từ môi trường chuyển trực tiếp đến hệ thống quyết định, sau đó ảnh hưởng đến hệ thống tác nghiệp.

Ví dụ : Sự cạnh tranh đã làm sụt giá một sản phẩm của XN, hoặc việc sản xuất một sản phẩm gặp phải khó khăn về tài chính...



*Hình 1.17 Điều khiển theo “chu trình mở”*

Theo hình vẽ, ta thấy : thông tin đến từ bên ngoài (môi trường) theo đường (1).

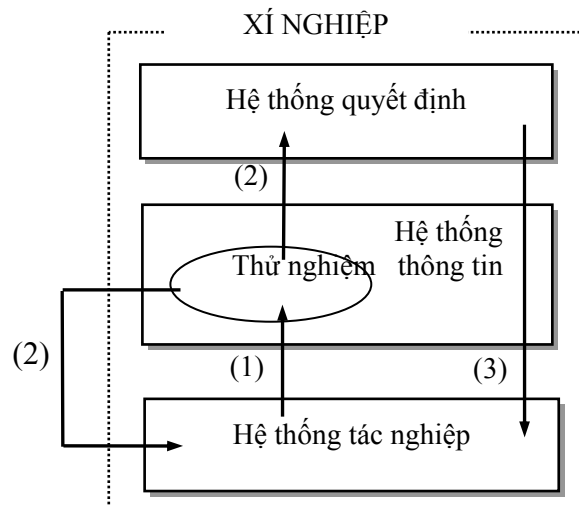
Theo (2) hệ thống quyết định hành động (tự điều phối) để trả lời thông tin này. Việc điều phối đòi hỏi phải có thêm các thông tin khác để có quyết định thật phù hợp.

Theo (3), quyết định được chuyển tới hệ thống tác nghiệp.

#### *b) Điều khiển theo chu trình đóng (khử hồi)*

Hệ thống quyết định nhận được các thông tin “đi lên” từ hệ thống tác nghiệp qua hệ thống thử nghiệm của HTTT. XN phải hành động theo cơ chế đi tới - khử hồi để giải quyết vấn đề.

Hiện tượng “khử hồi” xảy ra nhiều lần, cho đến khi vấn đề được xử lý. Tuy nhiên, trong một số trường hợp, người ta phải chấp nhận sự chênh lệch (dùng sai) về chất lượng giữa sản phẩm thành quả và sản phẩm nghiên cứu được để tránh “khử hồi” nhiều lần.



Hình 1.18 Điều khiển theo “chu trình đóng”

Từ hệ thống tác nghiệp, thông tin theo (1) đến hệ thống thử nghiệm. Theo (2), một quyết định được hành động (đi tới) hay là không (khử hồi). Theo (3), kết quả quyết định được chuyển đến hệ thống tác nghiệp.

c) Điều khiển bằng mệnh lệnh “báo động”

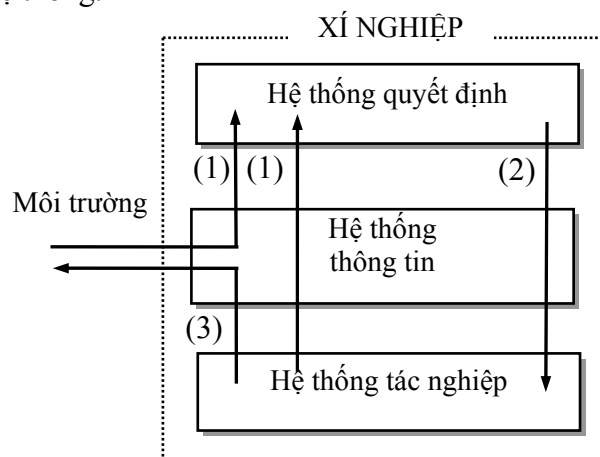
Trường hợp điều khiển bằng mệnh lệnh “báo động” được hình thành từ hai trường hợp trên. Thông tin đến hệ thống quyết định có thể có nguồn gốc từ nội bộ (hệ thống tác nghiệp) hoặc đến từ bên ngoài (môi trường).

Ví dụ :

Một sản phẩm mới được tung ra thị trường có thể gây ra sự xáo trộn về tổ chức nhân sự hoặc lao động trực tiếp.

Thiết bị phụ tùng sẽ bị cạn trong một thời gian ngắn tới đây.

Các chỉ dẫn về tình trạng báo động cho phép đưa ra các biện pháp xử lý thích hợp, để tránh mọi sự xáo trộn của hệ thống.



Hình 1.19 Điều khiển mệnh lệnh “báo động”

Theo hình vẽ, ta thấy : các thông tin đến từ bên ngoài XN hoặc từ hệ thống tác nghiệp (1). Một quyết định được đưa ra (2) và có thể chuyển ra bên ngoài (3).

### III.5. Phân loại các hệ thống thông tin

Có thể nhận thức một HTTTQL theo nhiều góc độ khác nhau tùy theo cách xử lý thông tin, cách khoanh vùng để xử lý, hoặc mức độ chính xác của thông tin.

#### a) Phân loại theo mức độ tự động hóa

Thông tin có thể được xử lý một cách thủ công, hoặc có sự trợ giúp của máy móc (điện thoại, photocopy, fax...) hoặc một cách tự động mà không có sự can thiệp của con người (MTĐT).

Tuy nhiên không phải lúc nào việc tự động hoá cũng hợp lý mà phải lựa chọn phương án xử lý thích hợp. Việc tự động hóa bằng Tin học chỉ có ý nghĩa khi thực sự có yêu cầu. Lựa chọn tự động hoá phụ thuộc vào các yếu tố :

- Độ lớn của XN,
- Khối lượng thông tin cần xử lý,
- Tốc độ mong muốn để nhận được kết quả, khái niệm thời gian trả lời,
- Chi phí xử lý tự động hoá,
- Thu lợi về thời gian hoặc tài chính...

#### b) Phân loại theo mức độ tích hợp các phương tiện xử lý

Khái niệm tích hợp (integra-tion) dựa trên hai yếu tố :

Khoanh vùng (localization) để xử lý.

Kiến trúc các phương tiện xử lý thông tin. Việc tích hợp chỉ được đặt ra khi việc xử lý tự động hóa gia tăng.

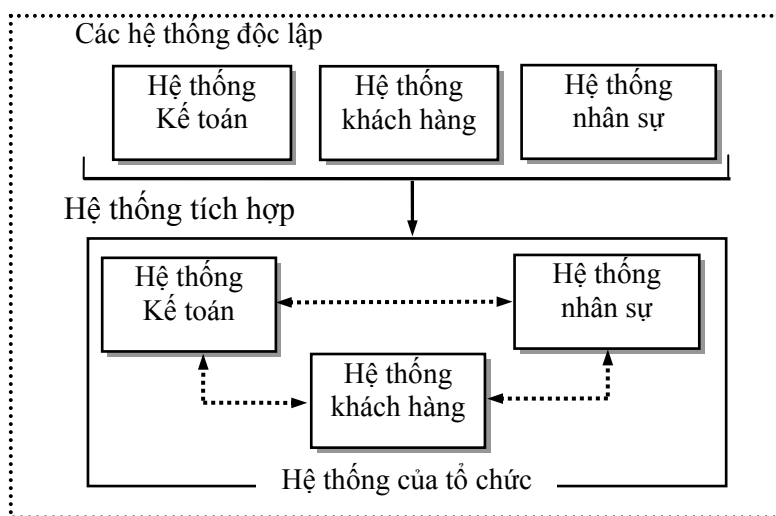
Trong cách tiếp cận này, các hệ thống xử lý thông tin tạo thành các hệ thống độc lập. Có thể xảy ra hiện tượng sai sót không hiệu quả của cả hệ thống do :

Có các thông tin dư thừa, trùng lặp trong mỗi hệ thống độc lập (Ví dụ, thông tin về khách hàng trong sơ đồ trên).

Có sự trùng lặp về xử lý từ mỗi hệ thống mà chỉ cần xử lý một lần (Ví dụ, thủ tục trao đổi khách hàng).

Hệ thống tích hợp

Với cách tiếp cận này, HTTTQL được xem như là một phần tử duy nhất. Mọi thông tin chỉ thu nhận một lần vào hệ thống và được sử dụng trong nhiều xử lý.



Hình 1.20 Tích hợp các phương tiện xử lý

### Các hệ thống độc lập

Ví dụ : Thông tin về khách hàng sẽ được thu nhận một lần và được xử lý cho từng hệ thống thành phần.

Hệ thống tích hợp đòi hỏi một cơ sở dữ liệu duy nhất với các phương tiện kỹ thuật thích hợp (mạng, viễn thông...). Như vậy, việc lựa chọn tích hợp sẽ tác động lên kiến trúc của các phương tiện xử lý thông tin.

#### Kiến trúc các phương tiện xử lý khác

Kiến trúc các phương tiện xử lý gắn liền với cấu trúc của các XN theo 3 loại :

##### *Kiến trúc tập trung (ít gặp)*

Thông tin được xử lý và hoàn thiện tại một nơi duy nhất trong XN rồi được phân bổ cho các nơi. Ưu : Xử lý nhất quán các dữ liệu, tránh được sự dư thừa thông tin.

Nhược : Hệ thống hoạt động nặng nề vì khối lượng thông tin lớn, dẫn đến khoảng thời gian giữa thời điểm thu nhận và thời điểm khai thác kết quả xử lý có thể tăng lên đáng kể.

##### *Kiến trúc phân tán*

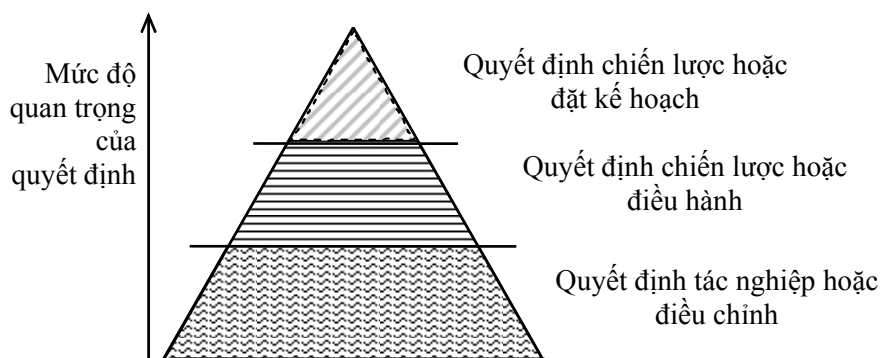
Máy tính được đặt tại các vị trí khác nhau theo tổ chức XN để xử lý độc lập rồi được nối với nhau (nhờ mạng) để trao đổi thông tin. Ưu điểm : Linh hoạt, uyển chuyển trong vận hành nên hay được sử dụng. Nhược điểm : Dữ liệu phát sinh nhiều, xử lý trùng lặp và không tương thích về thiết bị trong XN.

##### *Kiến trúc phân phối*

Kết hợp cả hai kiến trúc trên : xử lý thông tin tại điểm trung tâm trong khi thu nhận và phân phát thông tin lại được thực hiện một cách phân tán. Các phương tiện xử lý là các trạm cuối (terminal) nối với máy chủ (hote, main frame)

### c) Phân loại theo mức độ các quyết định

Từ các mức quyết định : chiến lược, chiến thuật, tác nghiệp, theo mức giảm dần trong tổ chức, xây dựng sơ đồ như sau :



Hình 1.21 Phân loại theo mức độ các quyết định

#### *Mức chiến lược (Strategic Level)*

Nhằm đưa XN hoạt động ngắn hạn hoặc dài hạn. Cần nhiều thông tin từ môi trường. Một số thông tin có thể xử lý tin học để đưa ra quyết định, nhưng thông thường có thể xử lý thủ công. Ví dụ : Tung ra thị trường sản phẩm mới, hoặc xây dựng cơ sở sản xuất mới, hoặc tuyển lựa cán bộ kỹ thuật cao cấp...



*Mức chiến thuật (Tactical Level)*

Các quyết định chiến thuật được đưa ra thường xuyên hơn nhằm đáp ứng nhu cầu hoạt động và hoàn thiện hệ thống. Ví dụ : Chọn giá bán sản phẩm, tuyển dụng nhân sự tạm thời, thay đổi cách cung ứng nguyên nhiên liệu, v.v...

*Mức tác nghiệp (Operational Level)*

Do nhân viên trong XN đưa ra hàng ngày. Ví dụ : gửi thư từ giao dịch, soạn thảo hóa đơn, thu nhận thông tin khách hàng, sản phẩm...

## CHƯƠNG 2

# Các phương pháp phân tích và thiết kế hệ thống

## I. Thế nào là phân tích hệ thống ?

### I.1. Khái niệm

Theo từ điển *Computer Dictionary*, Microsoft Press<sup>®</sup>, *phân tích hệ thống* (systems analysis) là sự khảo sát một hệ thống hay một vấn đề để cải tiến hệ thống đang tồn tại hoặc thiết kế và cài đặt hệ thống mới (nguyên văn tiếng Anh : the examination of system or problem, with the goal of either improving an existing system or designing and implementing a new one).

Phân tích hệ thống gắn liền với việc sử dụng phần cứng và phần mềm Tin học, bao gồm việc nghiên cứu chi tiết vấn đề, thiết kế, xây dựng những phương pháp tốt để giải quyết, nhằm đạt được mục đích theo những hạn chế và khả năng có thể. Những tiếp cận hay phân tích hệ thống đã có từ rất lâu, trước khi MTĐT ra đời.

*Ví dụ 1 :*

Khi xây dựng các Kim tự tháp cổ ở Ai Cập thì những người thiết kế được xem là các nhà kiến trúc sư, còn những người tổ chức việc vận chuyển nguyên vật liệu và huy động nhân lực được xem như là những người phân tích hệ thống.

*Ví dụ 2 :*

Gần đây hơn, khi xuất hiện các nhà máy, công sở (quá trình tư bản hóa công nghiệp) thì người chủ trì phải tìm hiểu cách tổ chức lao động, tìm kiếm các phương pháp tốt để tăng năng suất, tăng lợi nhuận... Đó là những hoạt động của người phân tích hệ thống.



Hình 2.1 Người tổ chức lao động là người phân tích hệ thống

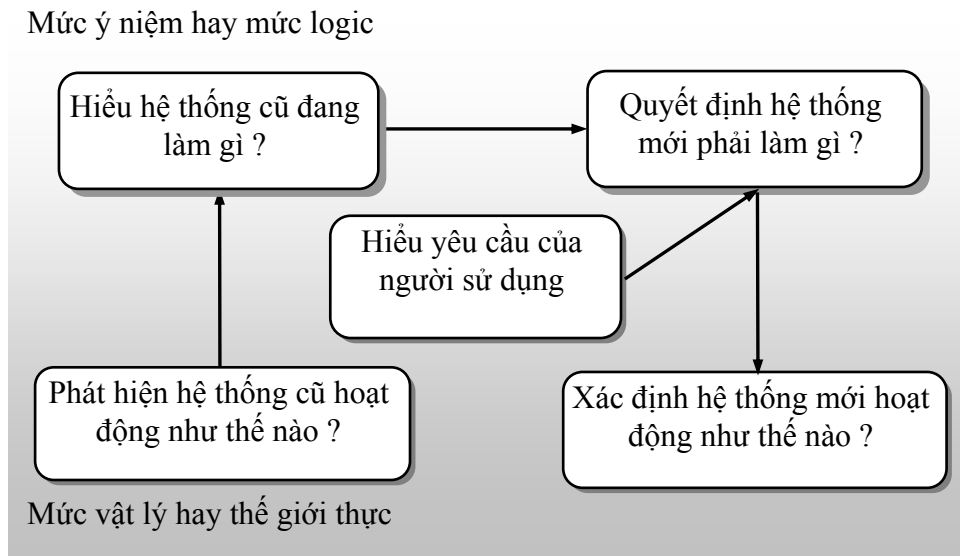
Nhu cầu về sản xuất thương mại, sự phát triển nhanh chóng của lĩnh vực Tin học đã dẫn đến việc phát triển ngành phân tích hệ thống áp dụng Tin học. Lĩnh vực này luôn luôn được nghiên cứu và phát triển nhằm hoàn thiện việc xây dựng các hệ thống thông tin.

Để thấy được vai trò của phân tích hệ thống, sau đây là những số liệu do Công ty IBM đã thống kê trong giai đoạn 1970-1980 :

Phân tích sai sót		Phân tích chi phí		Phân tích phân bổ hoạt động	
Mức ý niệm	45	Bảo trì	54	Lập trình	15
Lập trình	25	Phát triển	46	Thử nghiệm	50
Soạn thảo hồ sơ	7			Cài đặt	35
Các sai sót khác	20				
	100%		100%		100%

## I.2. Bản chất và yêu cầu của phân tích hệ thống

Phân tích là quá trình triển khai các giai đoạn mà nhà thiết kế hệ thống phải làm việc ở hai mức khái niệm khác nhau : “cái gì ?” (what?) và “như thế nào ?” (how?).



Hình 2.2 Mô hình theo mức của quá trình phân tích

Các yêu cầu của phân tích hệ thống :

1. *Tiếp cận toàn cục* bằng cách khảo sát mỗi phần tử (phòng, ban, xưởng, vị trí làm việc...) để tạo ra các dòng thông tin về hoạt động, quản lý và điều khiển trong một tổng thể toàn vẹn của hệ thống (xí nghiệp).
2. Sử dụng *phương pháp tiếp cận từ trên xuống* (top-down) để nhận thức, hiểu và đề ra biện pháp, từ tổng quát đến đặc thù, từ cái chung đến cái riêng... theo những tiêu chuẩn nhất quán.
3. Lĩnh hội được *tính trừu tượng, tính đặc thù* của mỗi thành phần trong hệ thống, từ đó sử dụng các công cụ thích hợp, hoặc tự động hóa, hoặc thủ công, trong quá trình phân tích.
4. Nắm được nhu cầu thực tiễn của người sử dụng cuối cùng.

### I.3.Đánh giá các phương pháp

Những thiếu sót mà các phương pháp phân tích hệ thống “cổ điển” mắc phải :

1. *Thiếu tiếp cận toàn cục*

Các chuyên gia (phân tích viên) làm việc một cách tự do, không có liên hệ gì với nhau dẫn đến khó có thể tích hợp các công việc

2. *Thiếu hợp tác với người sử dụng*

Sản phẩm phần mềm khó áp dụng, không phù hợp với công thái học (Ergonomie), không cùng cách suy nghĩ với NSD...

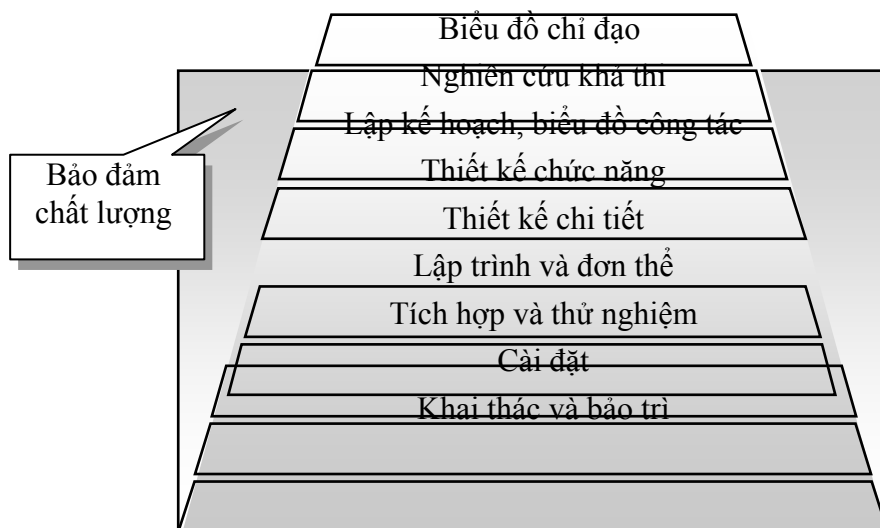
3. *Thiếu tiêu chuẩn thống nhất*

4. *Trùng lặp hoặc dư thừa thông tin, cùng một khái niệm mà có nhiều thuật ngữ..., không có tiêu chuẩn thống nhất về các đối tượng xử lý.*

Trong số những nguyên lý đặc trưng cơ bản của các phương pháp phân tích hiện nay đang có mặt trên thị trường, người ta chú ý đến :

1. Cơ sở lý thuyết trên một hệ thống Tin học hoá.
2. Chiến lược phát triển hệ thống nhưng tôn trọng các yếu tố liên quan đến chu kỳ sống (life cycle) là :
  - Trí tuệ (intelligence)
  - Thiết kế (design)
  - Triển khai (realization, achievement)
  - Cài đặt (implementation)
  - Bảo trì (maintenance)

Các giai đoạn khác nhau trong một chu kỳ sống của một dự án Tin học có thể được trình bày dưới dạng mô hình như sau :



Hình 2.3 Chu kỳ sống của một dự án Tin học

3. Tách rời tính cấu trúc và chức năng, các mức ý niệm, mức logic và mức vật lý của hệ thống để giảm độ phức tạp.
4. Xây dựng biểu đồ chỉ đạo triển khai thực hiện các giai đoạn khác nhau của quá trình phân tích thiết kế hệ thống (PTTKHT).

## II. Một số phương pháp PTTKHT “cổ điển”

Hiện nay, có rất nhiều phương pháp PTTKHT đã được đề xuất và được tiếp tục áp dụng. Bảng dưới đây liệt kê một số phương pháp.

STT	Tên phương pháp	Nguồn gốc	Hiện trạng thương mại
1	AXIAL	(Pháp)	IBM bán ra thị trường
2	CIAM (Conceptual Informa-tion Analysis Methodology)	Syslab (Thụy điển)	Đang tiếp tục được nghiên cứu
3	IDA (Interactive Design Approach)	Đại học Tổng hợp Namur (Bỉ)	METSI (Pháp)
4	JSD (Jackson System Development)	Michael Jackson	Cty Michael Jackson Ltd. (Anh)
5	MERISE	Sema-Matra (Pháp)	Nhiều công ty
6	METHOD/1	Arthur Andersen (Mỹ)	Arthur Andersen
7	REMORA	Đại học Tổng hợp Paris 1	Thomson (Pháp)
8	SADT	Softech (Mỹ)	Softech Inc. (Mỹ), Thomson IGL (Pháp)
9	SDM (Structured Design Methods)	Yourdon Inc. (Mỹ)	McDonnell Douglas (Mỹ)

Để hình dung về sự khác nhau giữa các quan điểm thiết kế HTTT, bảng dưới đây trình bày cách triển khai các giai đoạn của một số phương pháp phân tích hệ thống hay gặp.

Phương pháp	Lập kế hoạch	Phân tích hiện trạng	Thiết kế chi tiết	Triển khai	Cài đặt
SDM					
MERISE					
AXIAL					
JSD Jackson					

## II.1. Phương pháp SADT

Phương pháp SADT (Structured Analysis and Design Technique) là kỹ thuật phân tích và thiết kế có cấu trúc, do công ty Softech Inc. (Mỹ) phát triển, nhưng được áp dụng tương đối phổ biến ở châu Âu và ở Pháp. Ý tưởng cơ bản là phân rã hệ thống lớn thành các phân hệ nhỏ hơn và đơn giản hơn.

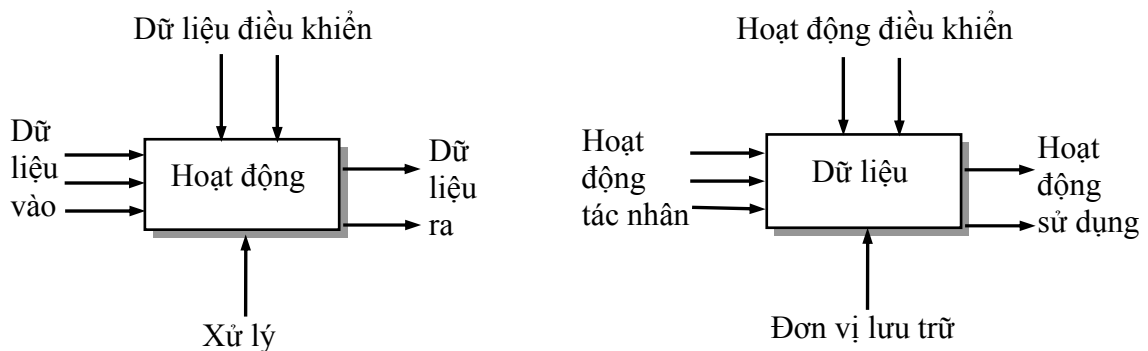
Theo quan điểm của SADT, mọi hệ thống được xem như một bộ sưu tập của các chức năng. Từ đó, SADT được sử dụng để xây dựng một mô hình biểu diễn mọi chức năng của một hệ thống và quan hệ của chúng với thế giới bên ngoài.

Phương pháp SADT đưa ra các lời khuyên “vàng” như sau :

1. Tính rõ ràng (trong sáng) quan trọng hơn là tính đúng đắn.
2. Một khía cạnh chưa tốt nhưng được diễn tả rõ ràng thì vẫn có thể được chấp nhận vì có thể được khắc phục sau đó.
3. Một khía cạnh chưa tốt nhưng không được diễn tả rõ ràng thì có thể không được chấp nhận vì có thể trở nên không tốt.
4. Cần phải biết nơi đến trước khi xuất phát.
5. Cần viết ra (giấy) hơn là chỉ nói ra (lời) và không nên kéo dài các buổi họp hành quá 60 phút chỉ vì một chủ đề.

Một mô hình SADT bao gồm các đơn thể (moduls) được tổ chức theo kiểu phân cấp (hierachical structure), tiếp cận từ trên xuống (top-down). SADT cho phép xây dựng các hệ thống phức tạp nhưng vẫn đảm bảo được tính tin cậy, tính đúng đắn.

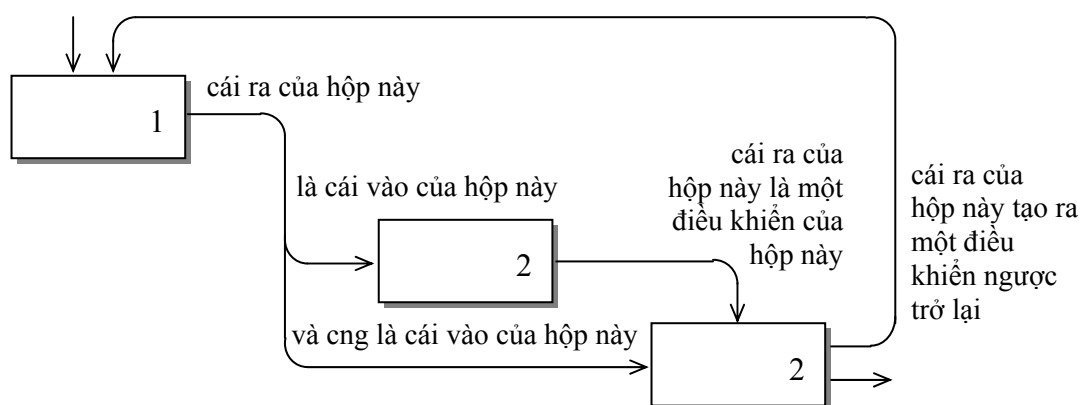
Về mặt cú pháp, mỗi đơn thể được biểu diễn bởi một trong hai dạng sơ đồ, sơ đồ hoạt động (activity diagram) và sơ đồ dữ liệu (data diagram). Sơ đồ hoạt động nhận dữ liệu vào, dữ liệu điều khiển, quy trình xử lý và cho dữ liệu ra. Sơ đồ dữ liệu nhận vào các hoạt động tác nhân và điều khiển, cho ra là hoạt động sử dụng :



Hình 2.4 Hai dạng sơ đồ SADT

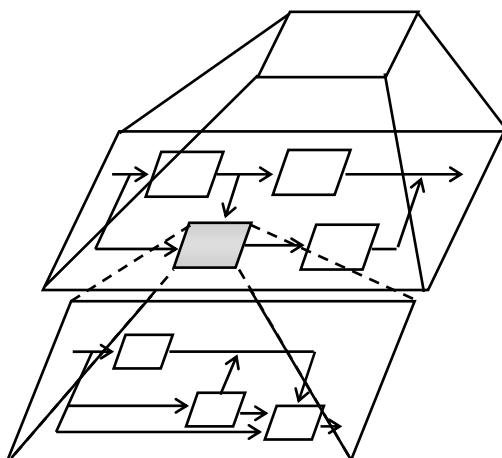
Một sơ đồ SADT thường có từ 3 đến 6 hộp (box) hình chữ nhật được liên kết với nhau bởi các mũi tên gắn nhãn (labeled arrow) thể hiện các giao diện (interface) hay các ràng buộc giữa các hộp. SADT đưa ra lời khuyên rằng một sơ đồ SADT mà có ít hơn 3 hộp sẽ làm nghèo hoặc không đặc tả đủ thông tin, nhưng nếu có nhiều hơn 6 hộp sẽ làm sơ đồ trở nên phức tạp khó theo dõi.

Nguyên tắc vẽ như sau :



Hình 2.22 Nguyên tắc vẽ sơ đồ SADT

Mỗi cạnh của hộp đều mang một ý nghĩa đặc biệt. Mỗi sơ đồ con là sự chi tiết hoá của một trong các hộp của sơ đồ cha. Một cha có thể có nhiều con. Mỗi sơ đồ con lại có thể có các sơ đồ con khác, v.v...



Hình 2.6 Cấu trúc phân cấp “một cha nhiều con”

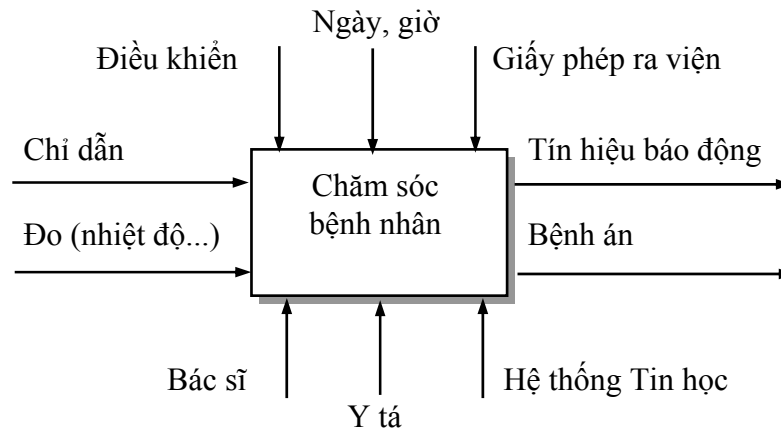
Sơ đồ SADT biểu diễn sự phân tích chủ đề ban đầu thành các thành phần nhỏ hơn. Mỗi thành phần là những *đối tượng* (objects) và những *sự kiện* (events), tương ứng với dữ liệu và hoạt động.

Ví dụ :

Dữ liệu :
Bệnh nhân
Bệnh án
Đơn thuốc

Hoạt động :
Thăm hỏi bệnh nhân
Xử lý bệnh án
Thanh toán tiền

Từ hai đối tượng trên, người ta vẽ được một sơ đồ SADT như sau :



Hình 2.7 Một mô hình xử lý của SADT

Nguyên lý làm việc theo nhóm của phương pháp SADT như sau :

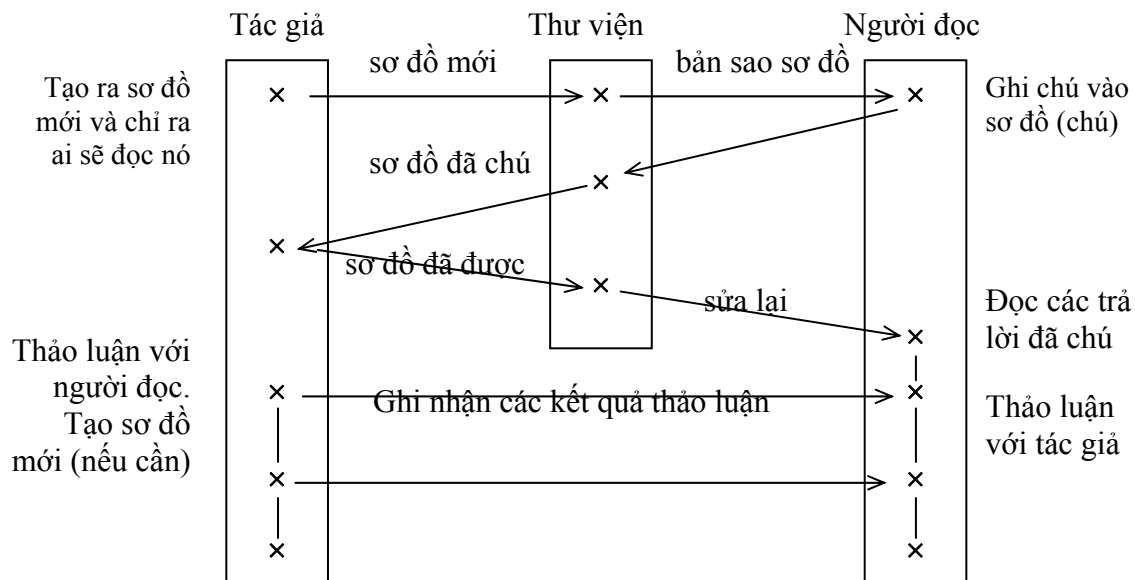
Mỗi sơ đồ được tạo ra bởi một tác giả (quy ước vẽ màu đen).

Sơ đồ được đọc và ghi chú (câu hỏi, gợi ý, điểm chưa rõ...) bởi người đọc (quy ước vẽ màu đỏ).

Sơ đồ sau đó được trả lại cho tác giả để thay đổi theo yêu cầu (quy ước vẽ màu xanh). Tác giả thay đổi xong lại đưa lại cho người đọc.

Thiết lập chu trình thảo luận tác giả – người đọc cho đến khi thoả mãn.

Trong quá trình luân chuyển sơ đồ giữa tác giả và người đọc, luôn luôn giữ lại một bản copy ở thư viện để lưu trữ.



Hình 2.8 Nguyên lý làm việc theo nhóm của SADT

## II.2.Phương pháp MERISE

Phương pháp MERISE (Méthode pour Rassembler les Idées Sans Effort, tạm dịch phương pháp tập hợp những ý tưởng dễ dàng) được đề xuất bởi CETE (Centre d'Etude Technique de l'Équipement d'Aix-en-Provence), INRIA (Institut Nationale de Recherche en Informatique et



Automatique) và Viện Đại học Marseilles III tại Pháp vào năm 1974. Đây là một phương pháp có cơ sở khoa học vững chắc, được sử dụng nhiều ở Pháp và châu Âu.

MERISE đưa ra một cách nhìn tổng quan về HTTT của xí nghiệp (XN) hay của một tổ chức, dựa trên mô hình ba hệ thống : hệ thống quyết định (hay hệ thống lãnh đạo), hệ thống thông tin và hệ thống tác nghiệp. Từ quan niệm này, HTTT được kiến trúc theo ba mức : *mức ý niệm* (conceptual level), *mức logic* hay *mức tổ chức* (organizational level) và *mức vật lý* hay *mức kỹ thuật* (technical level).

Mức ý niệm xây dựng mục đích và mục tiêu cuối cùng của XN trên cơ sở định nghĩa các ràng buộc, các quy tắc quản lý và cách xử lý chúng.

Mức tổ chức định nghĩa cách tổ chức hệ thống để XN đạt được mục đích

Mức kỹ thuật liên quan đến các phương tiện cần thiết của hệ thống : phần cứng, phần mềm, mạng, v.v...

Những đặc trưng cơ bản của phương pháp MERISE :

Tiếp cận theo mức nhằm hình thức hoá hệ thống tương lai

Tiếp cận theo giai đoạn nhằm phân cấp các quyết định.

Mức	Mô hình Dữ liệu	Mô hình Xử lý
Quan niệm	Mô hình ý niệm dữ liệu	Mô hình ý niệm xử lý
Tổ chức	Mô hình tổ chức dữ liệu	Mô hình tổ chức xử lý
Kỹ thuật	Mô hình vật lý dữ liệu	Mô hình tác nghiệp xử lý

Bảng thống kê sau đây chỉ ra tỷ lệ phần trăm trung bình cho mỗi mức :

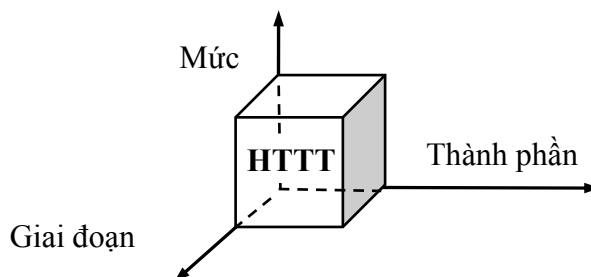
Số TT	Các giai đoạn		Tỷ lệ %
	Dữ liệu	Xử lý	
	Phân tích hiện trạng		± 50 %
	Mô hình ý niệm dữ liệu	Mô hình ý niệm xử lý	± 25 %
	Hợp thức hoá		± 5 %
	Mô hình logic dữ liệu		± 5 %
	Mô hình vật lý dữ liệu	Mô hình chức năng xử lý	± 15 %

Trong bảng trên, cột dữ liệu bên trái được xem là tĩnh so với cột bên phải được xem là động.

## II.3.PTTKHT theo quan điểm ba trục tọa độ

### II.3.1.Mô hình phân tích và thiết kế HTTT

Để nhìn nhận một HTTT cần thiết kế, hầu hết các phương pháp PTTKHT hiện nay đều sử dụng quan điểm ba trục tọa độ thuộc hệ quy chiếu “không gian ba chiều” là *mức*, *giai đoạn* và *thành phần*.



Hình 2.9 Ba góc nhìn khác nhau của HTTT

Phương pháp ba góc độ cho phép tập hợp các mục tiêu cần đạt được và những nội dung cần triển khai :

1. Trục *giai đoạn* xác định các bước dẫn đến một lời giải khả thi. Có chín giai đoạn cơ bản : lập kế hoạch, phân tích hiện trạng, phân tích khả thi, đặc tả, thiết kế, lập trình, thử nghiệm, khai thác và bảo trì.
2. Trục *mức* thể hiện cách tiếp cận và phương pháp luận để lựa chọn công cụ trên cơ sở chu kỳ sống của hệ thống. Đó là các mức ý niệm, logic và vật lý.
3. Trục *thành phần* xác định các thành phần cơ bản của một HTTT, là : dữ liệu, xử lý, thiết bị và nhân lực.

Hai trục *mức* và *giai đoạn* được định hướng một cách rõ ràng và tự nhiên do phương pháp tiếp cận từ trên xuống (từ tổng quát đến riêng biệt) và mức độ từ trừu tượng đến cụ thể (từ mức ý niệm đến mức vật lý).

Các thành phần dữ liệu, xử lý, bộ xử lý và nhân lực đặt trên trục thành phần có tính quy ước, vì trên thực tế, không hề có ưu tiên nào giữa chúng.

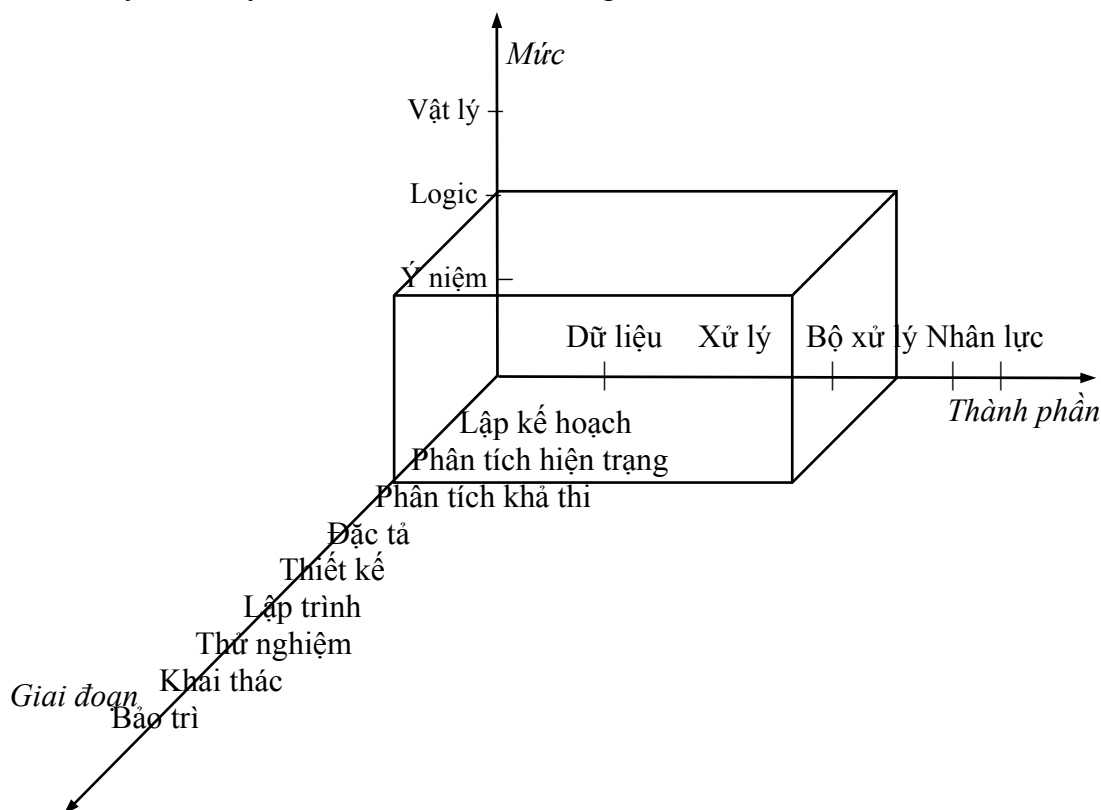
Từ quan điểm ba trục tọa độ, người ta cũng nhận thấy rằng, có hai yếu tố tham gia vào quá trình phân tích thiết kế HTTT là *chất lượng* và *giá thành*. Hai yếu tố này không tương thích với nhau.

Rõ ràng để giảm giá thành, cần xem xét hai trục thành phần và giai đoạn, để nâng cao chất lượng, cần chú ý trục mức là độ sâu sắc của sản phẩm. Trên thực tế, người ta phải ước tính giá thành (cost estimation).

Sau đây là bảng ước tính giá thành của phương pháp SDM.

STT	Nội dung công việc	Tỷ lệ % giá thành	Nhân lực
	Lập kế hoạch. phân tích hiện trạng	8 % (trên tổng giá thành)	80 ngày/người (trên 1000 ngày)
	Lựa chọn kiến trúc của HTTT	8 %	80 ngày/người
	Đặc tả bên ngoài của hệ thống. Phân tích tổng quan các xử lý	24 %	240 ngày/người
	Đặc tả bên trong của hệ thống. Phân tích chi tiết các xử lý, thiết kế CSDL	9 %	90 ngày/người
	Lập trình, thử nghiệm đơn thể	37.8 %	378 ngày/người
	Chuyển đổi HTTT cũ vào hệ thống mới, nhập dữ liệu ban đầu	7.2 %	72 ngày/người
	Thử nghiệm tích hợp	3 %	30 ngày/người
	Cài đặt vận hành	3 %	30 ngày /người
	Tổng cộng	100 %	1000 ngày

Hình vẽ dưới đây trình bày hệ trục tọa độ với nội dung của các trục.



Hình 2.10 HTTT được phân tích và thiết kế theo ba trục tọa độ

### II.3.2. Các giai đoạn phân tích và thiết kế hệ thống

Toàn bộ quá trình phân tích và thiết kế, từ giai đoạn ý niệm đến lúc khai thác HTTT, cần phải xác định và xây dựng ba mức của HTTT tương lai, đặc trưng hóa chính xác bốn thành phần cơ bản và triển khai lần lượt các giai đoạn. Để làm được điều này, phải tiếp cận chuẩn xác HTTT.

#### a) Lập kế hoạch

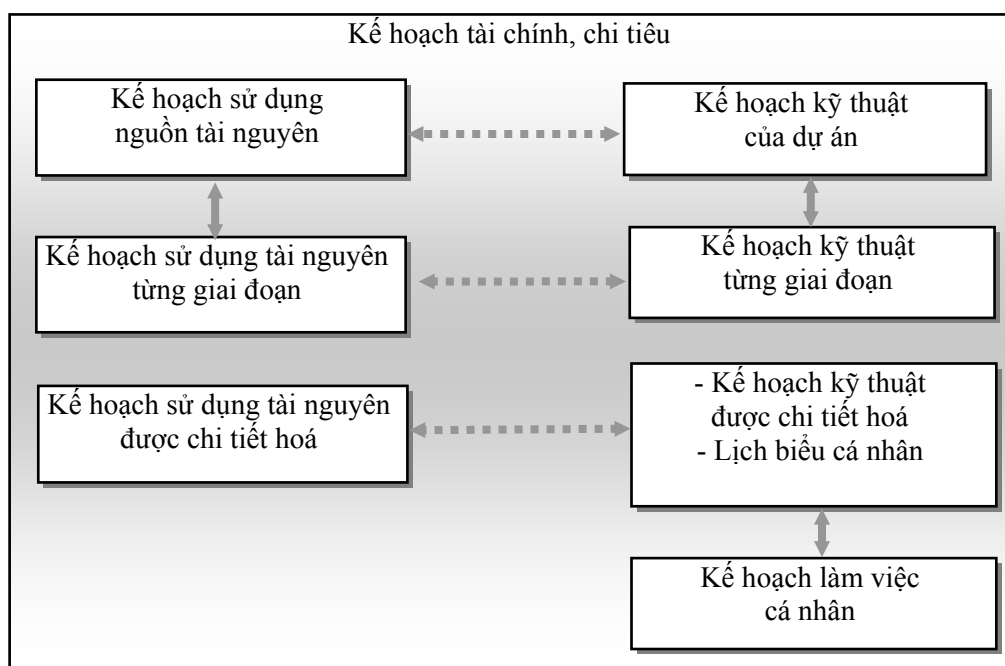
Thực hiện một dự án Tin học hóa có thể rất tốn kém, đòi hỏi nhiều công sức và thời gian (có thể mất rất nhiều tháng, nhiều năm) trước khi mang lại lợi nhuận.

Các nhân tố thường ảnh hưởng đến việc lập kế hoạch là : thời gian, mức đầu tư (investment), những yếu tố không chắc chắn của dự án, nguồn nhân lực (số lượng, trình độ, khả năng... của người thiết kế và những người sử dụng cuối), những tình huống bất ngờ, những đánh giá sai lệch thực tế...

Người ta thường cấu trúc hoá việc lập kế hoạch bằng cách :

- Tách riêng các phân bổ nhân lực, thời gian và kinh phí.
- Lập dự án tổng thể, kế hoạch cho một giai đoạn và các kế hoạch chi tiết.

Song song với việc lập kế hoạch là việc kiểm tra, báo cáo định kỳ.



Hình 2.11 Lập kế hoạch

Kết quả của giai đoạn lập kế hoạch là xác định rõ ràng các phân hệ, chức năng của chúng trong HTTT tương lai, xác định các khả năng ứng dụng trên mạng hoặc truyền thông, bố trí công việc theo nhóm chuyên gia, phân chia kinh phí...

### **b) Phân tích hiện trạng**

Phân tích (hay khảo sát) hiện trạng là giai đoạn phân tích các hoạt động của HTTT vật lý hiện hữu. Mục tiêu cần đạt được là làm sao có được các thông tin (liên quan đến những yêu cầu đặt ra trong bước lập kế hoạch) với độ tin cậy cao và chuẩn xác nhất, mới nhất.

Có nhiều phương pháp phân tích hiện trạng :

Phỏng vấn, trực tiếp hoặc gián tiếp, các đối tượng liên quan (giám đốc, nhân viên, vị trí làm việc...

Lập phiếu điều tra, thăm dò

Quan sát, thu thập mẫu biểu

Mỗi phương pháp đều có ưu điểm, nhược điểm riêng và được áp dụng sao cho phù hợp với tình hình thực tế.

Nguyên tắc : biết cách đặt các câu hỏi thiết thực thì biết càng nhiều thông tin về môi trường hoạt động của một tổ chức, càng dễ hiểu các vấn đề đang được đặt ra và tìm được phương án để giải quyết.

Sau khi có được các kết quả phân tích hiện trạng, phân tích viên phải biết cách tổng hợp các dữ liệu, các xử lý thu thập được và hợp thức hoá.

### **c) Phân tích khả thi**

Giai đoạn này có vai trò quyết định vì nó sẽ dẫn đến các lựa chọn sẽ quyết định HTTT tương lai cùng các bảo đảm tài chính. Gồm 5 bước :

Bước 1 : Phân tích, phê phán HTTT hiện hữu nhằm làm rõ các điểm yếu hoặc mạnh. Sắp xếp các vấn đề cần giải quyết theo thứ tự mức độ quan trọng của chúng.

Bước 2 : Xác định các mục tiêu mới của các hay dự án), khả năng sinh lãi, thời gian trả lãi, v.v..., nếu như việc này chưa được thực hiện ở giai đoạn lập kế hoạch.

Bước 3 : Xác định một cách tổng quát các giải pháp về chi phí triển khai phân hệ (dự án), chi phí hoạt động trong tương lai, kết hợp phân tích ưu điểm và khuyết điểm của từng giải pháp.

Bước 4 : Lựa chọn những người chịu trách nhiệm phù hợp với giải pháp nào đó đã xác định. Nếu không tìm được những người như vậy hoặc chi phí ước tính cao so với mục tiêu đề ra thì phải quay lên bước 2. Bước 4 trong trường hợp này thường lặp đi lặp lại nhiều lần.

### **d) Đặc tả**

Đặc tả (tiếng Anh : specifications, tiếng Pháp : cahier des charges) là việc mô tả chi tiết kỹ thuật các thành phần bên trong hệ thống, bao gồm :

Kiến trúc dữ liệu (data architecture) và xử lý kiểu dữ liệu tương ứng, các chỉ dẫn về tên (identifiers) dữ liệu, các sơ đồ, biểu đồ hay đồ thị.

Giao diện giữa HTTT và NSD : xác định HTTT cung cấp những gì cho NSD và ngược lại, NSD có thể khai thác được những gì từ HTTT ?

Các công việc và các cài đặt cần thực hiện.

Diễn biến tiến trình từ mức ý niệm đến lúc thể hiện : triển khai kế hoạch, phân công nhóm làm việc, v.v...

Kết quả của đặc tả là tập hợp các văn bản hồ sơ hay tư liệu về quá trình phân tích và thiết kế HTTT.

**e) Thiết kế**

Giai đoạn này xác định :

Kiến trúc chi tiết của HTTT, liên quan đến các giao diện với NSD và các đơn thể tin học cần áp dụng : các quy tắc quản lý, cấu trúc dữ liệu.

Thiết kế các đơn thể chương trình, chuẩn bị lập trình.

Quy cách thử nghiệm chương trình, sử dụng các thư viện

Quy cách khai thác, ứng dụng bảo trì, hướng dẫn sử dụng, v.v...

Các phương tiện và thiết bị liên quan

**f) Lập trình**

Giai đoạn này là thể hiện vật lý của HTTT bằng việc chọn công cụ phần mềm để xây dựng các tệp dữ liệu (database files), viết các đơn thể chương trình, chạy thử, kiểm tra, ráp nối, lập hồ sơ hướng dẫn, chú thích chương trình.

**g) Thử nghiệm**

Giai đoạn này bao gồm việc định nghĩa các thử nghiệm (testing) các đơn thể chương trình, thử nghiệm hệ thống, hoàn thiện quá trình đào tạo người sẽ sử dụng hệ thống, sửa chữa các chương trình nguồn, hoàn thiện các văn bản báo cáo và hướng dẫn sử dụng.

Việc thử nghiệm cho phép kết quả nhận được là phù hợp với các đặc tả ban đầu. Các phương pháp thử nghiệm được nghiên cứu chi tiết trong lĩnh vực công nghệ phần mềm (Software Engineering).

Các yếu tố liên quan đến thử nghiệm bao gồm :

Kế hoạch thử nghiệm

Danh mục (thư viện) thử nghiệm

Dữ liệu thử nghiệm

Các công cụ thử nghiệm

Các thử nghiệm đơn thể do người lập trình thực hiện :

- các nhánh chương trình
- tập hợp các thông số khác nhau
- các dữ liệu thường (normal data) và các dữ liệu đặc biệt, các giá trị bất thường

Các thử nghiệm tích hợp hệ thống do người phụ trách dự án triển khai.

**h) Khai thác**

Đây là giai đoạn quyết định đến kết quả tương lai của HTTT. Tùy theo kết quả khai thác, NSD sẽ quyết định có sử dụng HTTT vừa xây dựng để thay thế HTTT thủ công hay không ?

**i) Bảo trì**

Giai đoạn này gồm các công việc : bảo trì, cải tiến và thích nghi hóa HTTT với những thay đổi nội tại cũng như với môi trường chung quanh.

Nếu có kết quả phân tích ý niệm chính xác, xây dựng mô hình phù hợp và thể hiện vật lý hoàn hảo thì việc việc bảo trì HTTT sẽ dễ dàng. Ngược lại sẽ dẫn đến chi phí bảo trì tốn kém, khó làm thỏa mãn yêu cầu của NSD.

### II.3.3. Tiếp cận ba mức

Tiếp cận ba mức là để đáp ứng được các yêu cầu sau đây :

Sử dụng mô hình đơn giản và ngôn ngữ trong sáng, dễ hiểu để mô tả các kết quả đạt được trong từng giai đoạn.

Mô hình và ngôn ngữ đó dùng được cho những NSD khác nhau không nhất thiết phải là cán bộ Tin học.

Nhận thức HTTT ở mức thâm nhập đang xét.

#### a) **Mức ý niệm**

*Mức ý niệm* (hay quan niệm) mô tả các quy tắc quản lý (nhân sự, kế toán, chuyển giao sản phẩm...), các mục tiêu và những ràng buộc đặt ra đối với XN.

Phương pháp MERISE xây dựng hai mô hình ở mức ý niệm là mô hình ý niệm dữ liệu và mô hình ý niệm xử lý.

##### **Mô hình ý niệm dữ liệu :**

Nghiên cứu tổng quan dữ liệu, định nghĩa và hình thức hoá nhờ các yếu tố :

Thực thể

Đặc tính (thuộc tính) dữ liệu

Khoá

Quan hệ giữa các thực thể

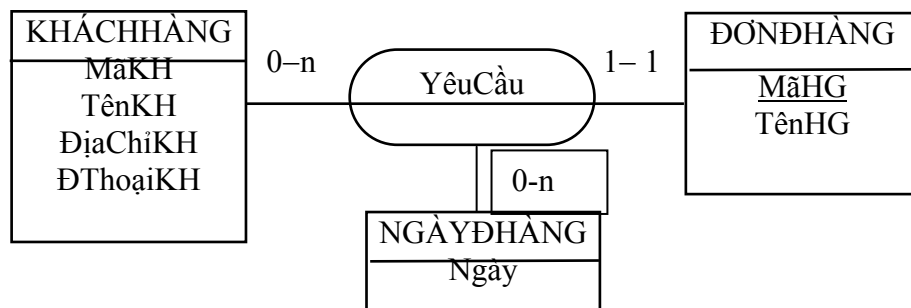
Các ràng buộc chức năng (functional constraints) giữa các mô hình :

Bản số (lực lượng)

Ràng buộc toàn vẹn chức năng (hàm)

Quá trình hợp thức hoá (validation), chuẩn hoá (normalization), phân rã (decomposition) và lượng hoá (quantification).

*Ví dụ :*



Hình 2.12 Mô hình ý niệm dữ liệu

##### **Mô hình ý niệm xử lý :**

Nghiên cứu tổng quan dữ liệu, định nghĩa và hình thức hoá :

Các phép toán

Các sự kiện

Các quy tắc

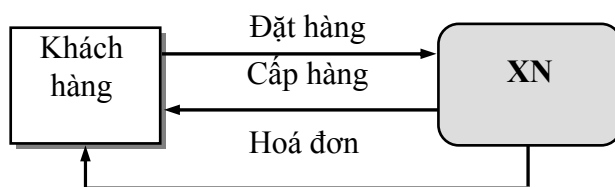
### Mô hình hoá

Mô hình hoá là sử dụng sơ đồ biểu diễn sự lưu chuyển và xử lý thông tin giữa các dòng dữ liệu (data flow), các kho dữ liệu (data stores) trong một XN.

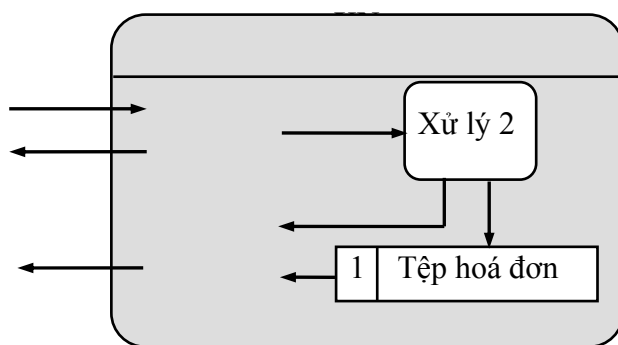
### Hợp thức hoá mô hình

Xem xét tập hợp các đặc tính để chỉ giữ lại những dữ liệu có ích cho xử lý, kiểm tra tính đúng đắn và hợp thức của các mô hình ý niệm xử lý.

Ví dụ :



Hình 2.13 Một mô hình xử lý ở mức đỉnh



Hình 2.14 Mô hình xử lý được phân tích chi tiết

### b) Mức logic

Các mô hình ý niệm cho phép biểu diễn dữ liệu độc lập với cách cài đặt vật lý. Vì vậy, trước khi chuyển sang mức vật lý, vấn đề là chuyển đổi các mô hình ý niệm sang mô hình ở mức logic tùy theo mức độ tự động hoá xử lý và cách chọn lựa các hệ quản trị CSDL. Có ba cách chọn kỹ thuật là :

Mô hình (hay quan điểm) CODASYL<sup>1</sup>

Mô hình quan hệ (sẽ được xét ở chương sau)

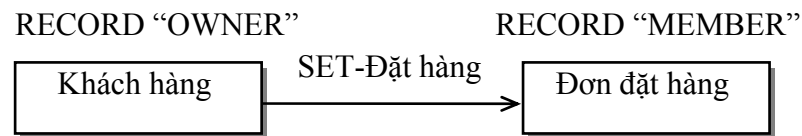
Các ngôn ngữ lập trình cổ điển

Ví dụ :

CODASYL đưa ra những khái niệm *trường* (field) - đơn vị thông tin nhỏ nhất hay dữ liệu sơ cấp, *bản ghi* (record) - tập hợp hữu hạn các trường, và, các *quan hệ* (set) được thiết lập giữa các bản ghi.

<sup>1</sup> CODASYL (viết tắt của Conference on Data Systems Languages, một tổ chức do Bộ Quốc phòng Mỹ sáng lập, mục đích phát triển các hệ quản trị dữ liệu và ngôn ngữ. Về COBOL.





Hình 2.15 Một mô hình CODASYL

### c) **Mức vật lý**

Hiện nay, chưa có những tiếp cận để chuẩn hoá việc mô tả và biểu diễn mức vật lý của dữ liệu. Việc mô tả một mô hình vật lý dữ liệu liên quan trực tiếp đến việc lựa chọn kỹ thuật. Có thể hiểu *mức vật lý* là sự kết hợp các phương tiện kỹ thuật cần thiết (phần cứng và phần mềm) để hệ thống có thể hoạt động, tùy theo sự phát triển của công nghệ. Các công cụ ảnh hưởng đến môi trường phát triển HTTT là :

Các công cụ quản lý màn hình

Các công cụ mô phỏng

Các công cụ quản lý mã nguồn

Ví dụ :

Mô hình CODASYL : IDS2, CLIO, TOTAL

Mô hình quan hệ : DB2, INGRES, ORACLE, Microsoft Access, FoxPro...  
các công cụ bảng tính...

## III. Phương pháp phân tích thiết kế hướng đối tượng, UML

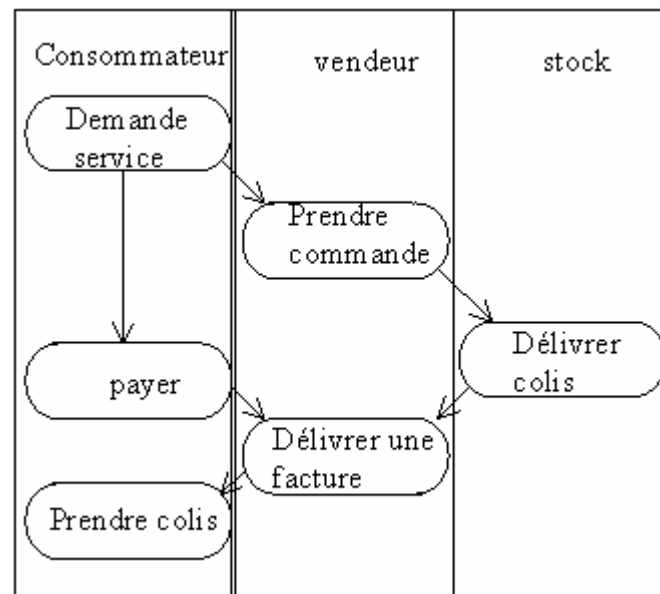
Hiện nay, dựa trên nguyên lý của lập trình hướng đối tượng (HĐT), người ta đã và đang tiếp tục nghiên cứu và ứng dụng các phương pháp phân tích hệ thống HĐT. Một trong các phương pháp tương đối phổ biến gần đây là sử dụng công cụ ngôn ngữ mô hình hoá thống nhất UML (Unified Modeling Language).

UML được phát triển năm 1994 bởi Grady Booch và Jim Rumbaugh tại hãng Rational Software Corporation, dựa trên các phương pháp đã được phát triển trước đó là BOOT và OMT (Object Modeling Technique). UML dễ học, đơn giản và thuần nhất hơn các phương pháp này.

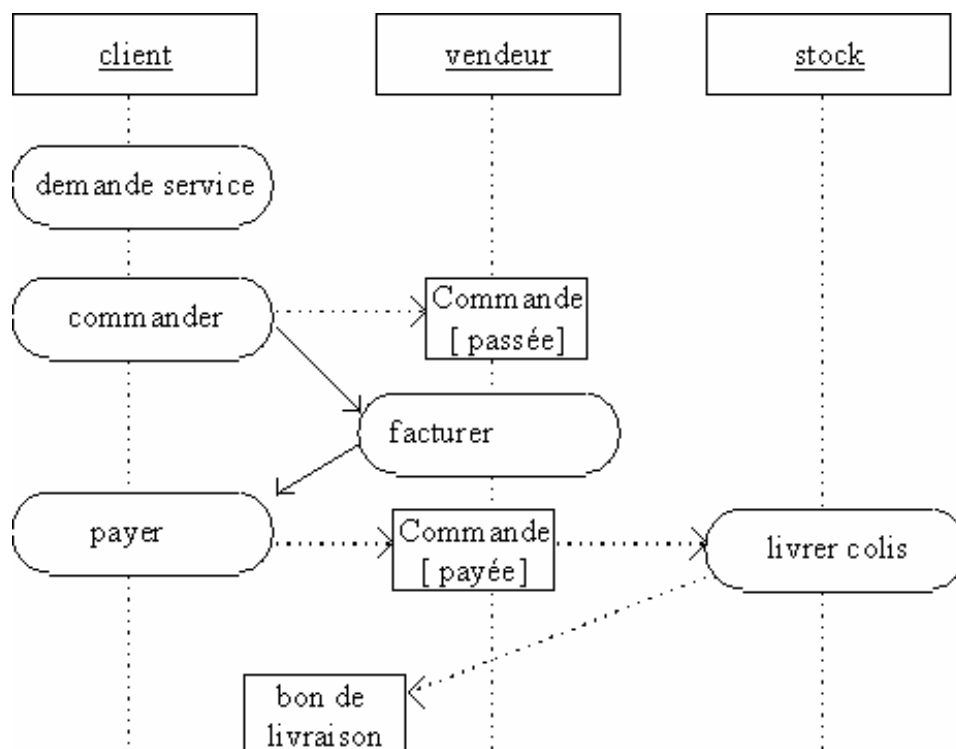
UML định nghĩa 8 sơ đồ cho phép biểu diễn các cách nhìn khác nhau về một mô hình của hệ thống, để từ đó, NSD có thể thao tác trên các phần tử của mô hình. Chín sơ đồ đó là :

1. *Các sơ đồ hoạt động* : Biểu diễn các phần tử tác nghiệp, thực chất là trạng thái thực hiện một phương pháp (chức năng) của một lớp đối tượng.
2. *Các sơ đồ sử dụng* : Biểu diễn các chức năng của hệ thống theo quan điểm của NSD.
3. *Các sơ đồ lớp* : Biểu diễn cấu trúc tĩnh của các lớp và mối liên hệ giữa chúng.
4. *Các sơ đồ hợp tác* : Biểu diễn không gian các đối tượng, các mối liên hệ và các tương tác của hệ thống.
5. *Các sơ đồ triển khai* : Biểu diễn sự triển khai các phần tử lên các thiết bị phần cứng, cũng như cách thực hiện các chương trình trên các thiết bị này.
6. *Các sơ đồ trạng thái–dịch chuyển* : Biểu diễn không gian các đối tượng, các mối liên hệ và các tương tác của hệ thống.

7. *Các sơ đồ đối tượng*: Biểu diễn các đối tượng, các mối liên hệ giữa chúng tương ứng với một sơ đồ hợp tác đơn giản, mà không thể hiện các trao đổi thông điệp.
  8. *Các sơ đồ dãy* : Biểu diễn theo thời gian các đối tượng và các tương tác giữa chúng.
- Sau đây là một ví dụ về mô hình hoạt động trong UML :



Hình 2.16 Một sơ đồ hoạt động trong UML



Hình 2.16 Sơ đồ hoạt động chứa các đối tượng trong UML

## **Câu hỏi chương 2**

1. Thế nào là phân tích hệ thống (PTTKHT) ? Bản chất và yêu cầu của PTTKHT là gì ? Mục đích của PTTKHT ứng dụng trong tin học hoá xí nghiệp. Hãy liên hệ với một bài toán tin học hoá xí nghiệp nào đó mà anh (chị) biết.
2. Hãy đánh giá về các phương pháp PTTKHT. Trình bày những ý tưởng cơ bản của hai phương pháp “cổ điển” SADT và MERISE.
3. Trình bày các quan điểm PTTKHT theo ba trục tọa độ.
4. Tóm tắt phương pháp PTTKHT hướng đối tượng sử dụng UML.
5. Trình bày các thành phần cơ bản của một hệ thống thông tin.

## CHƯƠNG 3

### Phân tích hiện trạng

---

*When I examine myself and my methods of thought,  
I come to the conclusion that the gift of fantasy  
has meant more to me than my talent  
for absorbing positive knowledge.*

*Albert Einstein*

Hiện nay, có rất nhiều phương pháp phân tích hiện trạng, mà trong khoa học xã hội và nhân văn, người ta thường gọi đó là các phương pháp và kỹ thuật điều tra xã hội học. Việc chọn sử dụng một phương pháp, hay sự nhấn mạnh nhiều hơn đến một phương pháp nào đó so với những phương pháp khác hoàn toàn tùy thuộc vào thói quen hoặc nhu cầu cụ thể của người phân tích. Thực tế, người ta thường sử dụng bốn phương pháp phân tích hiện trạng sau đây :

1. Phương pháp phân tích tài liệu
2. Phương pháp quan sát
3. Phương pháp thực nghiệm
4. Phương pháp phát vấn (hay trưng cầu ý kiến), gồm ba loại :
  - Phương pháp ăngket.
  - Phương pháp métric
  - Phương pháp phỏng vấn

Chương này giới thiệu phương pháp *phỏng vấn* (interview), một phương pháp rất phổ biến trong việc phân tích hiện trạng của hệ thống và giới thiệu một số công cụ cho phép hợp thức hoá kết quả phỏng vấn. Đây là giai đoạn đầu của quá trình phân tích và thiết kế.

Về mặt lý luận, mặc dù có thể áp dụng nhiều kỹ thuật khác nhau, như dùng phiếu điều tra, gửi câu hỏi đến các đối tượng cần nghiên cứu, v.v..., nhưng phương pháp phỏng vấn bằng cách tiếp xúc trực tiếp hệ thống cần phân tích vẫn là cách dễ thực hiện nhất và có hiệu quả nhất.

## I. Phương pháp phỏng vấn (interview)

### I.1. Nguyên lý của phương pháp

Phỏng vấn là tìm cách thu nhập thông tin qua các câu hỏi và các câu trả lời. Người phân tích (NPT) đặt câu hỏi cho đối tượng cần được khảo sát, sau đó ghi vào phiếu hay báo cáo kết quả phỏng vấn khi kết thúc cuộc phỏng vấn.

Yêu cầu đối với NPT là phải có nhiều kinh nghiệm, có kiến thức chuyên môn, ở đây là kiến thức về Tin học, và có sự am hiểu sâu sắc lĩnh vực cần được khảo sát cũng như trình độ điều luyện và thành thạo một cách nhuần nhuyễn nghệ thuật phỏng vấn.

Tính nghệ thuật trong phỏng vấn liên quan đến ba yếu tố là : đặt câu hỏi, lắng nghe và điều tra sáng tạo.

### ***Nghệ thuật đặt câu hỏi***

Trong thực tế, nếu NPT chỉ lắng nghe các câu trả lời của người được phỏng vấn một cách thụ động thì sẽ không đạt được kết quả. NPT rất dễ sa vào những chi tiết lan man, thiếu trọng tâm, lạc đề hoặc rơi vào những sự kiện hay những ý tưởng vụn vặt không gắn gì với những vấn đề cần nắm bắt của hệ thống.

Để khắc phục tình trạng trên, NPT phải thực hiện các yêu cầu sau :

Sắp xếp theo thứ tự rõ ràng và chính xác các khía cạnh đưa ra để hỏi.

Nội dung câu hỏi phải cụ thể tường minh, chỉ có thể hiểu theo một nghĩa, tránh những câu hỏi mập mờ hàm ý nhiều nghĩa bên trong.

Các câu hỏi phải đi thẳng vào vấn đề, tránh dẫn dắt người theo ý muốn chủ quan của mình, thiếu tính tế nhị.

Chỉ nên hỏi từng câu hỏi và phải chú ý đến những chi tiết chưa rõ hay người trả lời chưa trả lời đúng trọng tâm vì cố tình hay vô ý.

### ***Nghệ thuật lắng nghe***

Biết cách lắng nghe là một việc rất khó khăn, cần được rèn luyện và phát triển theo thời gian. Lắng nghe một cách chủ động, sáng tạo đòi hỏi phải có sự nhạy cảm cao trong tư duy, kết hợp giữa trực giác và cảm giác một cách chính xác.

NPT cần chú ý khi lắng nghe :

Chăm chú, chủ động thể hiện sự đồng cảm với người nói, biểu thị khả năng có thể thấu hiểu được những ý nghĩ và hành động của người nói.

Biết thu nhận (cả nghĩa bóng lẫn nghĩa đen), biết phán đoán và suy luận, biết chất lọc và hiểu được những điều được nghe, được cảm nhận.

Biết kiên nhẫn chờ đợi, với một tình cảm chân tình, và có khả năng chia sẻ, gợi ý những điều chưa rõ.

### ***Nghệ thuật biến phỏng vấn thành một cuộc điều tra sáng tạo***

Phỏng vấn đòi hỏi phải tiến hành như một quá trình ứng xử sáng tạo “thiên biến vạn hóa”. Như đã biết, sẽ không bao giờ có hai đối tượng được phỏng vấn trả lời hoàn toàn đồng nhất với nhau. Vì vậy, kết quả một cuộc phỏng vấn hoàn toàn tùy thuộc vào khả năng của NPT.

## **I.2. Phân tích hiện trạng**

Phân tích (hay tìm hiểu) hiện trạng sử dụng phương pháp phỏng vấn để nắm được từ tổng quát đến chi tiết về một số lĩnh vực thuộc ví nghiệp (XN) có dự định Tin học hoá. Đó là quá trình tiếp xúc giữa NPT với nhiều loại người khác nhau tùy theo chức năng công tác của họ trong XN : là lãnh đạo, quản lý hay thừa hành.

Người lãnh đạo cho biết một cách tổng thể về tổ chức XN, nhiệm vụ cụ thể của từng bộ phận, các mục tiêu trung hạn và dài hạn của XN.

Người quản lý cho biết các chức năng thuộc lĩnh vực họ đang phụ trách.

Người thừa hành cho biết thông tin về một công việc cụ thể.

Khi phân tích hiện trạng, NPT sẽ làm việc với hai nhóm đối tượng : nhóm lãnh đạo và nhóm các vị trí làm việc.

### **I.3. Phỏng vấn lãnh đạo**

#### *a) Mục tiêu*

- Phỏng vấn lãnh đạo nhằm hiểu được tính tổng thể của tổ chức, của từng lĩnh vực và của từng bộ phận.
- Phỏng vấn được tiến hành tuần tự theo cấu trúc phân cấp của hệ thống tổ chức : đầu tiên là ban giám đốc, sau đó đến các bộ phận (phòng, ban...) chức năng.
- Qua phỏng vấn, càng nắm được nhiều hồ sơ của nhiều lĩnh vực, nhiều báo cáo biểu mẫu thống kê hàng tháng, hàng quý, hàng năm, v.v... càng tốt.

#### *b) Kết quả phỏng vấn*

Kết quả phỏng vấn là :

Phát biểu lại vấn đề (cần Tin học hoá) một cách chính xác.

Liệt kê các mục tiêu XN cần đạt được, các con số có tính toàn cục.

Giới hạn những vị trí làm việc chính liên quan đến lĩnh vực quan tâm.

Xác định phạm vi phân tích khả thi.

Phạm vi phân tích khả thi (những gì cần làm, làm như thế nào và những gì thì không cần làm ?) thường không rõ ràng lúc tiến hành phỏng vấn, tuy nhiên, cùng với quá trình nhận thức hiện trạng, phạm vi này càng lúc càng rõ ra.

Liệt kê những hạn chế, ràng buộc về phương tiện, thời gian và kinh phí.

### **I.4. Phỏng vấn các vị trí làm việc**

#### *a) Mục đích*

Phỏng vấn các vị trí làm việc nhằm tiếp thu được tất cả những công việc cùng các thông tin cần xử lý ở tất cả các vị trí làm việc thuộc lĩnh vực nghiên cứu.

Các cuộc phỏng vấn và các buổi củng cố kết quả phỏng vấn tiếp theo giúp NPT nhận thức được hiện trạng và dựa trên hiện trạng này, tiếp tục phát triển các giai đoạn tiếp theo.

#### *b) Phỏng vấn*

Tại mỗi vị trí làm việc liên quan đến lĩnh vực đang xét, cần phỏng vấn người phụ trách và những người thừa hành ở đó.

*Ví dụ :*

Khi nghiên cứu hiện trạng lĩnh vực quản lý vật tư, hàng hóa, những vị trí làm việc sau đây sẽ được tiến hành phỏng vấn : thủ kho, kế toán, nhân viên cung ứng, nhân viên thống kê và lập phiếu, v.v...

NPT cần đặt ra những câu hỏi thường có dạng như sau :

Đối với mỗi hồ sơ :  
- “Hồ sơ này dùng để làm gì ?”  
- “Nguồn gốc (xuất xứ) của hồ sơ ?”  
- “Hồ sơ này sẽ giao cho ai ?”, v.v...

Đối với mỗi công việc :  
- “Cái gì đã khởi động công việc này ?”  
- “Công việc này tạo ra kết quả gì ?”  
- “Công việc này được làm như thế nào ?”

Câu trả lời của người được hỏi cho phép NPT mô tả từng công việc như sau :

- Những sự kiện khởi động một công việc,
- Chu kỳ và thời lượng thực hiện,
- Khối lượng dữ liệu liên quan,
- Các qui tắc cần áp dụng để thực hiện công việc.



Hình 3.23 Phỏng vấn các vị trí làm việc

NPT quan sát sự luân chuyển thông tin là các hồ sơ giữa các vị trí làm việc. Vật mang thông tin có thể là giấy in, lời nói... thể hiện dưới nhiều dạng : hồ sơ in, hoặc viết tay, điện thoại, telex, fax., v.v.... Hiện nay, giấy vẫn là vật mang chủ yếu.

### c) Báo cáo kết quả phỏng vấn

Chú ý : Khái niệm *vấn đề* chưa được định nghĩa chặt chẽ ở đây, được hiểu là tập hợp các hành động thực hiện liên tục tại một vị trí.

Nếu một trong những vấn đề trên không được trả lời, NPT phải tìm cách phát hiện ra các điểm không bình thường. Ngoài ra, NPT phải ghi lại những ý kiến, mong muốn hoặc khó khăn của những người được phỏng vấn về việc làm của họ.

Để tránh các trở ngại tâm lý, người phụ trách phỏng vấn cần tiếp xúc trước với những người sẽ được đối thoại để giới thiệu với họ về cách thức làm việc, sự cần thiết của phỏng vấn, v.v...

Kết quả phỏng vấn được NPT tóm lược và trình bày trong *báo cáo kết quả phỏng vấn*. Ví dụ dưới đây mô tả một mẫu báo cáo kết quả phỏng vấn :

Tên cơ quan Tin học, Địa chỉ, điện thoại...	Đề án số : ..... Người thực hiện : .....
<b>BÁO CÁO KẾT QUẢ PHỎNG VẤN</b>	
Cơ quan : .....	Địa chỉ, điện thoại : .....
Người được hỏi : .....	Chức danh : ..... Chức vụ : .....
Vấn đề 1 : .....	Trả lời : .....
Vấn đề 2 : .....	Trả lời : .....
Tài liệu : .....	Tóm tắt : .....
Các thỏa thuận đạt được : .....	
Nhận xét : .....	
Xác nhận của cơ quan	Ngày.....tháng.....năm..... Ký tên :

Sau bước phỏng vấn, NPT thu nhận được cách thức luân chuyển hồ sơ, sự trao đổi thông tin và cách lưu giữ chúng.

## **I.5.Củng cố các phỏng vấn**

Chấm dứt công việc ở “hiện trường” chưa phải là kết thúc phỏng vấn. Sau các buổi phỏng vấn, NPT cần phải xem lại những gì thu thập được để hệ thống lại và sắp xếp chúng, dự kiến những gì phải tiến hành tiếp theo.

Như vậy mục tiêu của bước củng cố phỏng vấn, ngoài việc chuẩn bị cho các giai đoạn tiếp theo (xác định các ý niệm cơ bản về dữ liệu, công thức tính toán...) còn nhằm để phát hiện được những điểm còn mơ hồ, từ đó tiếp tục nêu ra các câu hỏi phụ khi gặp lại những người đối thoại trong các buổi phỏng vấn kế tiếp.

Thực tế, nếu người phân tích thực hiện việc phỏng vấn buổi sáng tại nơi liên quan, thì buổi chiều, họ có thể ở nhà để củng cố phỏng vấn.

### *a) Thể hiện các mục tiêu và các hạn chế*

Với mỗi công việc tại một vị trí, kết quả thao tác trên các dữ liệu phụ thuộc vào một số quy tắc quản lý thể hiện những mục tiêu hoặc những hạn chế. Như đã trình bày, để nhận thức một HTTT, cần thâm nhập theo 3 mức : mức ý niệm, mức logic và mức vật lý. Mức sau sẽ bổ khuyết và chi tiết hoá mức trước. Có 3 loại quy tắc cho phép quá trình thâm nhập này.

### **1. Quy tắc quản lý**

Quy tắc quản lý là sự thể hiện các mục tiêu đã được chọn và những hạn chế được chấp nhận bởi XN. Quy tắc quản lý có nguồn gốc từ :

Bên ngoài XN : luật lệ, quy định, tương quan lực lượng với khách hàng và với nhà cung cấp hàng hoá vật liệu...

Bên trong XN (xử lý, tính toán...) thể hiện những mục tiêu cần thực hiện.

Ví dụ : - Kiểm kê được thực hiện theo chu kỳ (tuần, tháng, quý, năm...).

- Các sản phẩm trước khi tiêu thụ phải được nhập kho thành phẩm.

Các công thức tính toán mô tả các hành động của mỗi công việc.

Ví dụ : - Lương cơ sở bằng lương cơ bản nhân với hệ số trượt giá.

- Giá đơn vị của hàng hoá tồn kho theo công thức bình quân.

### **2. Quy tắc tổ chức**

Quy tắc tổ chức được định nghĩa khi đã nắm được các thành phần của hệ thống tổ chức XN, mô tả cái “ở đâu ?”, “ai ?” và cái “khi nào ?”.

*Ví dụ :*

- Thủ kho theo dõi trạng thái tồn trữ kho hàng. Đây là quy tắc quản lý vật tư - hàng hóa theo quy định của Nhà nước.

- Phiếu giao hàng chỉ được giao nhận vào cuối ngày. Trường hợp này, quy tắc tổ chức phản ánh lề lối làm việc, không nhằm thể hiện mục tiêu cần phải đạt.

- Đơn đặt hàng gửi cho một nơi cung cấp hàng hoá nào đó chỉ có thể gửi đi vào những ngày nhất định trong tuần.

Trong nghiên cứu hiện trạng, NPT tiếp xúc với tổ chức cũ. Nhưng do những mục tiêu mới và mối quan tâm đến tối ưu hóa, NPT có thể đề xuất các quy tắc mới trong khuôn khổ các hạn chế của tổ chức.

### **3. Quy tắc kỹ thuật**

Quy tắc kỹ thuật liên quan đến mức tác nghiệp, mô tả các giải pháp kỹ thuật nhằm đạt được mục đích, trả lời câu hỏi “cách thực hiện thế nào?”.



- Ví dụ : - Cấu hình của máy như thế nào (CPU, Memory, HDD, FDD...) ?  
 - Có cần thiết nối mạng không ?  
 - Nên chọn loại máy in nào ?

*b) Liệt kê các công việc*

NPT ghi kèm theo mỗi công việc những mô tả như sau :

*Tên công việc* : được chọn sao cho dễ nhận dạng, không bị nhầm lẫn về sau.

*Điều kiện khởi động* công việc thể hiện qua các sự kiện xảy ra. Các sự kiện này có thể đến từ bên ngoài hoặc đến từ vị trí làm việc khác..

*Mục đích* : mọi công việc đều sản sinh ra ít nhất một kết quả.

*Tần suất công việc* : được ước tính theo các giá trị trung bình.

*Thời gian cần thiết* : được ước tính theo các giá trị trung bình.

*Các quy tắc liên quan* : liệt kê theo thứ tự (quản lý, tổ chức, kỹ thuật).

*Lời bình* : ghi những khó khăn, những đề nghị liên quan đến công việc.

*c) Liệt kê các công thức tính toán*

Tùy theo đặc tính (hành động hoặc tính toán) và tùy theo độ phức tạp của các công thức tính toán mà NPT lựa chọn một cách thức thể hiện thích hợp. Dưới đây là một số phương pháp thể hiện :

Dùng ngôn ngữ thông thường : tuy có tính trực quan, nhưng đôi khi nặng nề, thiếu chính xác, vì vậy thường được dùng cho các công thức đơn giản.

Dùng ngôn ngữ đặc tả : nhằm chuẩn bị cho bước lập trình tiếp theo, cho phép phân rã một công thức phức tạp thành những công thức đơn giản hơn.

Dùng công thức toán học : thể hiện chính xác nhưng đòi hỏi đặt tên dữ liệu.

*Ví dụ* : Số lượng vật tư, hàng hóa tồn kho cuối kỳ được tính theo công thức :

$$N_{\text{cuối kỳ}} = N_{\text{tồn đầu kỳ}} + N_{\text{nhập trong kỳ}} - N_{\text{xuất trong kỳ}}$$

Các phương pháp khác : dùng bảng quyết định, cây quyết định hoặc lưu đồ thể hiện mặt tính của hệ thống.

*d) Liệt kê dữ liệu*

Tiếp tục nghiên cứu tập hợp các hồ sơ đã được photocopy. Thông thường rất ít hồ sơ có đủ lượng thông tin cần thiết. Cần sắp xếp, phân loại những hồ sơ có mối liên quan họ hàng với nhau và ấn định thứ tự trước khi phân tích.

Ở đây, cần phân biệt thông tin mà vật mang “chuyên chở” với dữ liệu mà hồ sơ chứa đựng. Để thể hiện thông tin, người ta sử dụng dữ liệu. Thông tin được thể hiện qua các giá trị của các dữ liệu.

*Ví dụ* :

“Hóa đơn N<sup>o</sup> 88512 của nhà cung cấp A phải được thanh toán chậm nhất là vào ngày 10/09/1996” là thông tin.

Ở ví dụ trên, nhóm “N<sup>o</sup> 88512” là giá trị của dữ liệu “số thứ tự hóa đơn”, “A” là giá trị của “tên nhà cung cấp” và “10/09/1996” là giá trị của “ngày giới hạn thanh toán”. Như vậy toàn bộ thông tin được thể hiện qua ba dữ liệu : số thứ tự hóa đơn, tên nhà cung cấp và ngày giới hạn thanh toán.

Quá trình trên giúp NPT lập được hồ sơ về danh sách, chuẩn bị cho việc xây dựng từ điển dữ liệu. Hồ sơ này xác định một ngôn ngữ chung cho mỗi người tham gia đề án, được sử dụng cho bất kỳ áp dụng tin học nào và luôn được bổ sung dữ liệu mới.

Sau khi tinh chế tập hợp dữ liệu trong danh sách dữ liệu, NPT chuẩn bị xây dựng từ điển dữ liệu. Với mỗi dữ liệu thu nhập được, NPT mô tả trên tờ phiếu gồm các mục sau :

Tên dữ liệu : chọn tên tượng trưng phù hợp với ngôn ngữ thường dùng của XN.

Định nghĩa : phân tích sâu hơn bản chất của dữ liệu, dễ hiểu nhưng tổng quát, xác định được phạm vi các giá trị dữ liệu có thể nhận.

Cấu trúc dữ liệu : chữ cái, chữ số (ký tự), số, hay kiểu logic..., đồng thời chỉ rõ số ký tự cần thiết để biểu diễn cấu trúc đó.

Kiểu : để xác định kiểu dữ liệu. Dữ liệu có thể là :

- *hiện diện* trong tổ chức như thế nào ?
- *để chế biến* theo công thức tính toán ;
- *kết nối* nhiều dữ liệu với nhau thành chuỗi ;
- *sơ cấp* (không được chế biến).

Dữ liệu sơ cấp không tính chất tuyệt đối, tùy theo cách xử lý dữ liệu mà có thể thay đổi. Ví dụ : “địa chỉ khách hàng” có thể được xem là dữ liệu kiểu chuỗi để chế biến (tách riêng đường phố, phường...), hoặc được xem là kiểu sơ cấp, v.v...

Định lượng : ước tính số các giá trị khác nhau mà dữ liệu có thể nhận được.

Ví dụ : liệt kê một số giá trị dữ liệu làm ví dụ.

Lời bình : có thể diễn đạt tùy ý nhằm giải thích dữ liệu, hoặc đề nghị một định nghĩa khác.

Ngoài ra, NPT có thể ghi thêm hai mục :

*Mức* : chính xác hóa dữ liệu nằm ở mức nào : ý niệm, logic, hoặc vật lý.

*Ngày tạo* : ghi ngày tháng năm tạo dữ liệu.

Chú ý : Quản lý danh sách dữ liệu và từ điển dữ liệu cho phép tiết kiệm thời gian đáng kể trong việc cập nhật cũng như tham khảo.

Cũng cố phòng vấn giúp NPT xác định nhiều điểm cần được làm sáng tỏ. Một lần nữa cần lưu ý đến tâm lý của NSD trong các buổi tiếp xúc liên quan đến tương lai của ứng dụng Tin học. Vì ở đây, NPT đã đụng chạm đến cách thức, lề lối làm việc và những vấn đề thường nhật của họ. NPT chú ý giải thích càng nhiều càng tốt về những điểm mà Tin học có thể tham gia, làm tăng hiệu quả công việc và lợi ích của XN. Mặt khác, NPT cũng phải lưu tâm uốn nắn những ngộ nhận, cũng như những ảo tưởng về Tin học (cái gì làm được, cái gì thì không, v.v...).

## II. Tổng hợp các kết quả phân tích hiện trạng

Giống như một người quan sát hiệu kỳ tìm hiểu một cỗ máy cơ khí phức tạp bằng cách tháo rời ra từng bộ phận, NPT thông qua các buổi phỏng vấn đã phát hiện các vị trí làm việc khác nhau.

Bằng cách “tháo rời”, NPT thu nhận được từ các vị trí làm việc nhiều “chi tiết rời” và nhóm chúng lại thành ba loại : *quy tắc*, *công việc* và *dữ liệu*. Từ đó, NPT xây dựng dần các kế hoạch “lắp ráp” nhằm thể hiện hoạt động của hệ thống XN trong lĩnh vực nghiên cứu.

Do khó có thể tiếp cận trực tiếp toàn bộ hệ thống, người ta thường chọn cách tiếp cận phân tích. Từ kết quả phân tích, người ta bắt đầu quá trình tổng hợp.

### II.1. Xác định các phân hệ

Ý tưởng chủ đạo cho việc ý niệm hóa các phân hệ là nhóm những hoạt động có sự phù hợp nào đó giữa chúng về ba mặt : *mục tiêu hướng tới*, *các quy tắc tác động đến* và *các dữ liệu thao tác*.

Định nghĩa : Một phân hệ xác định một hoặc nhiều mục tiêu mà XN phải đạt được. Phân hệ được mô tả bởi một tập hợp các hoạt động, các quy tắc quản lý và các dữ liệu tác nghiệp.

Để xác định một phân hệ :

Nhóm các hoạt động phù hợp với nhau về mục tiêu cần đạt được.

Liên hệ tập hợp các quy tắc quản lý với những hoạt động này.

Liên hệ “cộng đồng” dữ liệu với những hoạt động này : tài chính, nhân lực, nguyên liệu hoặc một kiểu chức năng nào đó : sản xuất, kiểm soát ngân sách, quan hệ con người.

*Ví dụ* : Một XN sửa chữa nhỏ có thể được chia ra thành bốn phân hệ :

Quản lý vật tư hàng hóa

Theo dõi hoạt động sửa chữa

Quản lý nhân sự, tiền lương

Kế toán.

Nhận xét :

Thông thường, việc phân chia tập hợp các hoạt động và tập hợp các quy tắc để ghép chúng vào một phân hệ đặc thù nào đó không phức tạp. Tuy nhiên, việc tách dữ liệu riêng cho từng phân hệ lại khó thực hiện.

Người ta thừa nhận sự có mặt của các dữ liệu chung cho nhiều phân hệ. Các dữ liệu chung này có vai trò hạt nhân của các giao diện và là cơ sở cho việc xây dựng một ý niệm tổng thể các dữ liệu, độc lập với các xử lý, hợp thành một ngân hàng dữ liệu phục vụ cho mọi phân hệ.

Việc phân chia thành các phân hệ có ưu điểm sau :

Nếu đối tượng nghiên cứu có “kích thước” nhỏ, nó là một lĩnh vực duy nhất.

Trái lại, nếu kích thước đối tượng lớn, lúc ấy khó lòng hình dung đầy đủ toàn bộ các xử lý xảy ra.

Mỗi phân hệ được giao cho một nhóm NPT. Các nhóm làm việc song song với nhau. Việc phân nhóm không nhất thiết cố định trong toàn bộ tiến trình phân tích và thiết kế.

Ví dụ : có thể giữ nguyên một lĩnh vực duy nhất về phương diện ý niệm, sau đó tách rời nó ra khi thực hiện hoặc áp dụng.

Việc phân chia đối tượng nghiên cứu không làm mất tính tổng thể của phương pháp. Thật vậy, đối với dữ liệu, người ta xây dựng một mô hình duy nhất trước khi tạo ra các lĩnh vực. Chính vì để xác định các lĩnh vực, người ta ghép cho mỗi lĩnh vực một tập hợp các dữ liệu thuần nhất. Như vậy , người ta tách mô hình dữ liệu thành các tiểu mô hình phục vụ cho từng lĩnh vực, không kết hợp nhiều mô hình dữ liệu đặc trưng cho từng lĩnh vực thành một mô hình dữ liệu duy nhất.

## II.2. Phân tích dữ liệu

Phân tích dữ liệu là để :

Thanh lọc các dữ liệu đồng nghĩa và nhiều nghĩa từ danh sách dữ liệu “thô” thu nhập qua các buổi phỏng vấn

Xây dựng từ điển dữ liệu (Data Dictionary) là tập hợp từ vựng chính xác mà mọi người tham gia dự án đều đồng ý.

Các bước phân tích dữ liệu được cho ở .

### II.2.1. Khái niệm về dữ liệu sơ cấp

Một dữ liệu được xem là sơ cấp khi toàn bộ các ký tự tạo nên dữ liệu luôn luôn được sử dụng một cách có hệ thống. Xét 2 ví dụ sau :

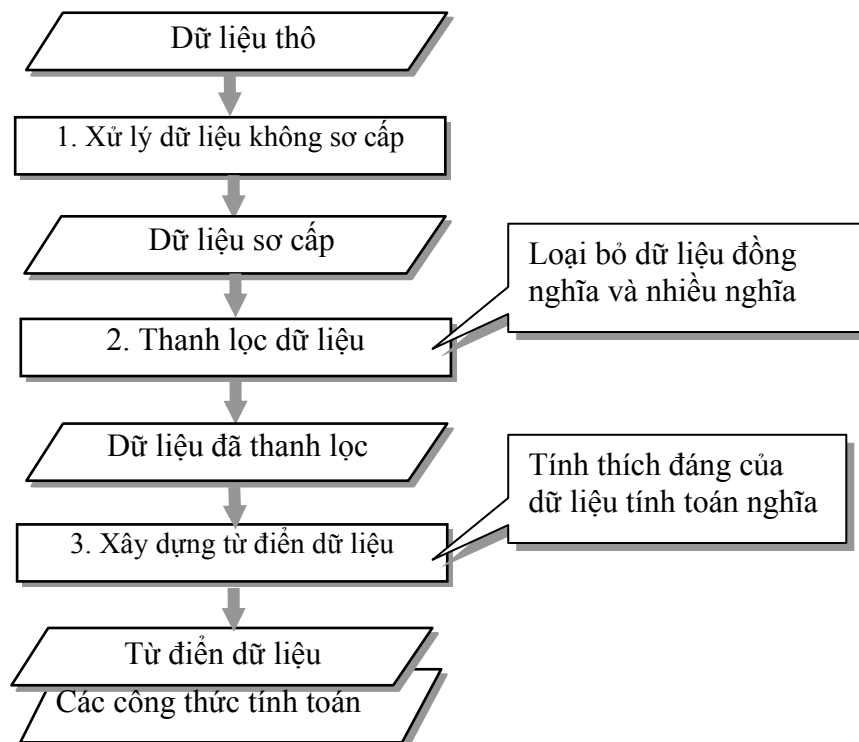
*Ví dụ:*

Giả sử dữ liệu mã sản phẩm có tên MãHàng có giá trị H025 có mặt trong cơ sở dữ liệu là do hai dữ liệu ghép lại : chữ H có ý nghĩa rằng sản phẩm thuộc về một loại đặc biệt nào đó và 025 là số của sản phẩm. Như vậy :

Loại sản phẩm có giá trị thay đổi, là một chữ cái từ A đến Z.

Số của sản phẩm có giá trị thay đổi từ 0 đến 999.

Nếu như khái niệm về loại sản phẩm không có mặt trong ứng dụng đang xét, ta có thể xem rằng MãHàng là sơ cấp.



Hình 3.24 Phân tích dữ liệu

*Ví dụ :*

Dữ liệu địa chỉ khách hàng ĐịaChỉKH được xem là dữ liệu sơ cấp nếu được dùng làm địa chỉ thư tín, công văn.

Trái lại, nếu cần lập danh sách khách hàng của một thành phố, hoặc của một tỉnh, thì bản thân ĐịaChíKH là dữ liệu hỗn hợp và phải được phân chia thành các dữ liệu sơ cấp. Ta có thể phân chia như sau :

SốNhàKH    Số nhà của khách hàng. Ví dụ : 17.  
PhốKH        Tên phố (hoặc tên đường, hẻm...) khách hàng lưu trú. Ví dụ Lê Duẩn.  
MãBĐ        Mã bưu điện của khách hàng (Việt nam chưa có).  
TPKH        Tên thành phố nơi khách hàng lưu trú. Ví dụ Đà Nẵng.  
QuốcGia    Tên nước (nếu có). Ví dụ Việt nam.

Ta thấy rằng mã bưu điện MãBĐ nếu có là dữ liệu hỗn hợp do mã tỉnh và mã huyện ghép lại.

### ***II.2.2.Thanh lọc dữ liệu***

Khi xây dựng một từ điển dữ liệu, cần phải phân tích dữ liệu thành các dữ liệu sơ cấp theo quan hệ dữ liệu – tên gọi dữ liệu. Sự không phù hợp của dữ liệu thể hiện ở hai dạng cần xử lý : các dữ liệu đồng nghĩa và các dữ liệu nhiều nghĩa

#### ***a) Dữ liệu đồng nghĩa***

Dữ liệu đồng nghĩa là tên gọi khác để chỉ cùng một loại dữ liệu, một sự kiện. Như vậy, một dữ liệu có thể có hai tên gọi khác nhau, dẫn đến sự dư thừa thông tin. Cho nên, với một dữ liệu chỉ nên có một tên dữ liệu.

Ví dụ :        Mã số mặt hàng, danh mục mặt hàng.  
                 Đơn giá, giá đơn vị.

Trong ví dụ đang xét, những tên dữ liệu MãKH và SốHĐĐặtHg đều có nghĩa rằng những khách hàng này đã đặt hàng tại XN DanaFood. Ở đây chỉ nên có một tên dữ liệu là MãKH.

#### ***b) Các dữ liệu có nhiều nghĩa***

Dùng cùng một tên để chỉ các đối tượng, sự kiện khác nhau.

Ví dụ :    Tài khoản trong một cơ sở sản xuất kinh doanh để theo dõi các hoạt động sản xuất kinh doanh. Tài khoản trong ngân hàng để theo dõi tiền rút, tiền gửi của khách hàng.

Dữ liệu SLSX sử dụng trong phòng Kinh doanh chỉ ra số lượng sản phẩm cần sản xuất để đáp ứng một đơn đặt hàng nhưng trong kho lưu trữ không còn đủ bán.

SLSX cũng được dùng để chỉ ra số lượng cần sản xuất thực sự của phân xưởng sản xuất theo nhu cầu của đơn đặt hàng và những ràng buộc về sản xuất.

Như vậy, số lượng sản xuất là một tên dữ liệu, nhưng có hai cách sử dụng khác nhau, gọi là dữ liệu có nhiều nghĩa.

Trong một hệ quản trị CSDL, tên dữ liệu chỉ có thể tương ứng với một dữ liệu.

Như vậy, có thể dùng hai tên dữ liệu :

- SLĐặt : Số lượng do phòng Kinh doanh đặt hàng.
- SLSX : Số lượng thực sự phải sản xuất.

### ***II.2.3.Xây dựng từ điển dữ liệu***

Sau khi đã thanh lọc dữ liệu, cần phải giữ lại những dữ liệu thật sự có ích, đó là *nguyên lý thích đáng*, và giữ lại những dữ liệu không thể được xác định bởi những dữ liệu sơ cấp có mặt trong cơ sở dữ liệu, đó là những *dữ liệu tính toán*.

### a) Nguyên lý thích đáng

Ví dụ :

Xét dữ liệu số lượng đặt hàng SLĐặt. Phân xưởng sản xuất nhận tất cả đơn đặt hàng của khách hàng chuyển đến (từ phòng Kinh doanh). Có thể suy ra số lượng cần sản xuất từ những dữ liệu cần có (trạng thái kho lưu trữ hiện hành, số lượng đang sản xuất, và số lượng đặt hàng) để thỏa mãn nhu cầu của khách hàng.

Như vậy, dữ liệu SLĐặt là vô ích và sẽ không có mặt trong từ điển dữ liệu.

### b) Các dữ liệu tính toán

Cần phân biệt trong số những dữ liệu tính toán :

1. Những dữ liệu có thể được xác định ở mọi thời điểm nhờ những dữ liệu sơ cấp có trong CSDL.

Ví dụ 1 :

Số tiền giảm giá cho mỗi đơn đặt hàng STGiảm phụ thuộc vào doanh số của năm trước và doanh số có được các tháng từ đầu năm cho đến tháng khách hàng đăng ký đặt hàng.

Trong năm, những dữ liệu này luôn có mặt và không bị thay đổi, cho nên, để dữ liệu STGiảm trong cơ sở dữ liệu là không cần thiết.

Ví dụ 2 :

Giá trị của dữ liệu SLDựTrữ có thể được xác định tại mọi thời điểm buôn bán thỏa thuận. Như vậy, dữ liệu này cũng không cần để trong cơ sở dữ liệu.

2. Những dữ liệu làm cho các dữ liệu trong cơ sở dữ liệu có thể tiến triển.

Ví dụ : Số tiền trong một đơn đặt hàng GiáTiền được tính từ số lượng sản phẩm đặt hàng và giá đơn vị của sản phẩm. Số lượng đặt hàng chỉ liên quan đến đơn đặt hàng và không thể thay đổi theo thời gian.

Ngược lại, giá đơn vị của sản phẩm có thể thay đổi (do nhiều lý do), GiáTiền của đơn đặt hàng sẽ được tính lại sau một thời gian nào đó và sẽ khác với số tiền xác định tại thời điểm đặt hàng. SôTiền phải có mặt trong cơ sở dữ liệu (có thể lưu giữ giá đơn vị của sản phẩm liên quan tại thời điểm đặt hàng).

Trong ví dụ này, mọi dữ liệu dù tính toán hay không đều có mặt trong từ điển dữ liệu. Tuy nhiên, trong những bước phân tích về sau, những dữ liệu tính toán có thể được xác định ở mọi thời điểm, chẳng hạn STGiảm, sẽ không còn trong từ điển.

### c) Cách biểu diễn từ điển dữ liệu

Từ điển dữ liệu là một danh sách các dữ liệu được sử dụng trong hệ thống thông tin, với một số đặc trưng tên, kiểu, lĩnh vực sử dụng, các quy tắc tác động. Dưới đây là mẫu mô tả từ điển dữ liệu :

STT	Tên dữ liệu	Loại	Mô tả kiểu dữ liệu	Công thức
-----	-------------	------	--------------------	-----------

Trong đó :

STT                      Số thứ tự của dữ liệu.

Dữ liệu                Diễn giải dữ liệu đã liệt kê trong bảng dữ liệu sơ cấp, ở đây chỉ liệt kê tên dữ liệu theo thứ tự ABC để dễ tìm kiếm.

Loại                    TT chỉ dữ liệu tính toán, KTT chỉ dữ liệu không tính toán.

*Mô tả kiểu dữ liệu* có hai cách xác định :

hoặc kiểu dữ liệu và độ rộng của dữ liệu :

N	Dữ liệu số
N6	Số có 6 chữ số.
N6.2	Số có 8 chữ số với 2 chữ số lẻ (sau dấu chấm thập phân).
C	Dữ liệu văn bản (ký tự)
D	Ngày tháng năm.
L	Dữ liệu Logic (true, false)

hoặc chỉ kiểu mở rộng :

1, 2, 3	Chỉ nhận 3 giá trị số là 1, 2, hoặc 3
xanh, đỏ, vàng, tím	Chỉ nhận 4 giá trị loại văn bản là xanh, đỏ, vàng, tím.
v.v...	

*Công thức* Mỗi dữ liệu có giá trị nhận được bằng cách áp dụng một công thức sẽ được ký hiệu bởi CT1, CT2, ... và được trình bày sau từ điển.

Mỗi công thức tương ứng với một công thức tính toán hoặc một cấu trúc lập trình (lựa chọn), hoặc tổ hợp cả hai.

*Ví dụ :*

Công thức tính toán :

$$\begin{aligned}\text{GiáTiền} &= \text{ĐơnGiá} * \text{SốLượng} \\ \text{STGiảm} &= \text{GiáTiền} \times \text{tỷ lệ giảm}\end{aligned}$$

Cấu trúc lập trình :

```
IF DoanhSố > 10 000 000      THEN STGiảm = 10%
                               ELSE STGiảm = 0%

ENDIF
```

Các phép toán logic NOT, AND và OR có thể sử dụng trong các công thức.

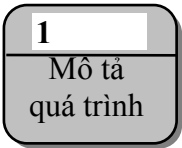

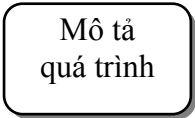
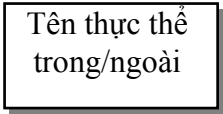
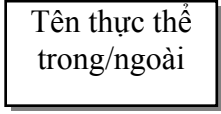
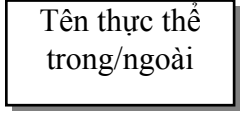
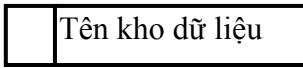
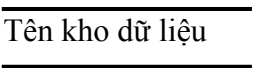
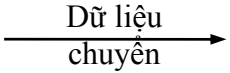
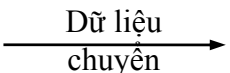
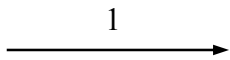
## II.3.Sơ đồ dòng dữ liệu

### II.3.1.Khái niệm

Sơ đồ (hay lưu đồ) *dòng dữ liệu* (DFD - Data Flow Diagram) là công cụ để mô tả các dòng thông tin của hệ thống đang xét. DFD dễ viết, dễ đọc, dễ hiểu và được ứng dụng rất phổ biến. DFD được xây dựng từ các hình vẽ và ký hiệu quy ước.

Có nhiều cách xây dựng DFD, thông dụng là phương pháp De Macro-Yourdon, Gane Sarson và MERISE (ở Pháp).

Sau đây là bảng trình bày các phương pháp :

	Gane-Sarson	DeMarco-Yourdon	MERISE
Quá trình			
Thực thể			
Kho dữ liệu			
Dòng dữ liệu			

Một DFD gồm bốn thành tố : quá trình, thực thể, kho dữ liệu và dòng dữ liệu chuyển.

#### Quá trình

Quá trình (process) mô tả hoạt động (activities) hay phép biến đổi (transform) một hoặc nhiều dòng dữ liệu vào (input) thành một hoặc nhiều dòng dữ liệu ra (output). Quá trình không chỉ ra chi tiết logic hay thủ tục xử lý. Trong sơ đồ DFD, một quá trình có thể là một người sử dụng hay máy tính.

#### Thực thể

Các thực thể (entities) xác định ranh giới (boundary), hay phạm vi (scope), hay ngữ cảnh (context) của hệ thống đang xét, để cung cấp cái vào cho hệ thống và lấy cái ra từ hệ thống.

Các thực thể có thể nằm bên trong (internal) hay bên ngoài (external), tạo thành các nguồn và các đích của hệ thống. Mỗi thực thể có thể là người, tổ chức hoặc là một hệ thống khác tương tác với hệ thống đang xét.

#### Kho dữ liệu

Kho dữ liệu (data stores) chỗ chứa những thông tin được lưu lại theo thời gian. Đó là các tập tin xử lý thủ công hay tự động, các cơ sở dữ liệu, hay bất cứ hình thức tập trung dữ liệu nào (bảng biểu báo cáo, danh mục tra cứu, từ điển, hộp thư...) đều có thể được xem như kho dữ liệu.

#### Dòng dữ liệu

Dòng dữ liệu (data flows) là phương tiện lưu chuyển thông tin thể hiện cái vào và cái ra. Dòng dữ liệu có thể là báo cáo, biểu mẫu, văn bản, thư tín, thông điệp hay dữ liệu nói chung.



Các dòng dữ liệu thể hiện sự tương tác trong hệ thống. Chúng được tạo thành từ tập hợp các vật mang thông tin (giấy, màn hình... ) có cùng bản chất, đi từ nơi phát (nguồn) đến nơi nhận (đích).

DFD chỉ ra cách chuyển tiếp thông tin từ một quá trình này sang một quá trình khác. Như vậy, cần phải có sẵn những thông tin trước khi thực thi một quá trình. DFD là một phần của quá trình phân tích chứ không phải một phần của quá trình điều tra.

### II.3.2. Phân biệt DFD với sơ đồ khối

DFD khác với sơ đồ khối, hay lưu đồ (Flowcharts) :

Các quá trình trong DFD có thể đặt song song, một số hoạt động đồng thời.

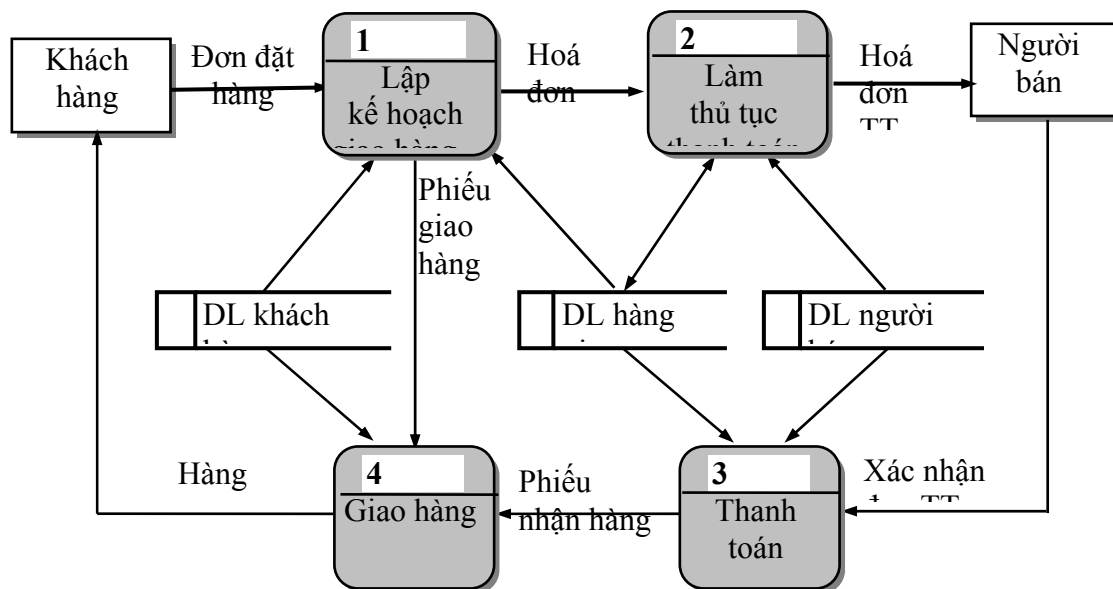
DFD thể hiện các dòng dữ liệu luân chuyển trong một hệ thống. Sơ đồ khối thể hiện tuần tự thực hiện dãy các bước xử lý trong một thuật toán.

DFD thể hiện quá trình ở các thời điểm khác nhau.

DFD không giống sơ đồ khối về cách thể hiện các cấu trúc vòng lặp (do-while) và cấu trúc lựa chọn (if-then-else).

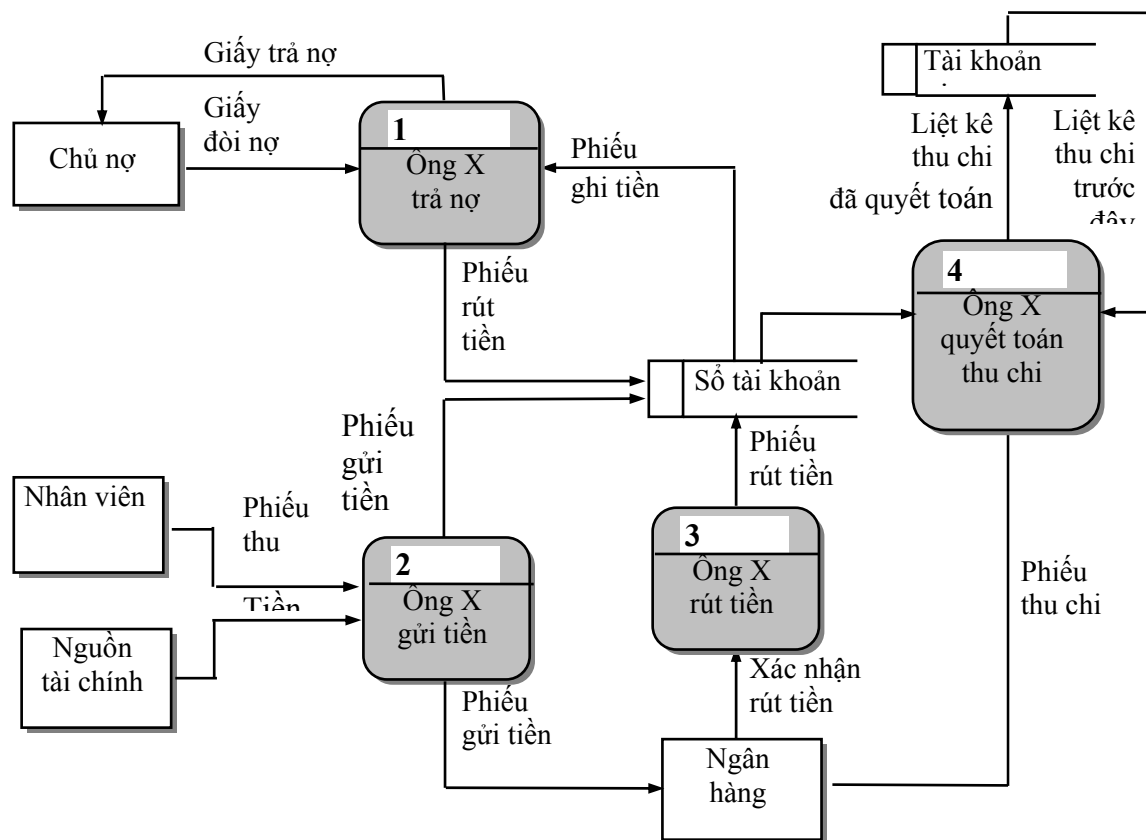
### II.3.3. Ví dụ :

DFD mô tả quá trình khách hàng đặt hàng, thanh toán tiền và nhận :



Hình 3.25 Hệ thống mua bán hàng

DFD dưới đây mô tả hệ thống tài chính cá nhân, giả sử của ông X nào đó :



Hình 3.26 Hệ thống tài chính cá nhân

### II.3.4. Xây dựng sơ đồ dòng dữ liệu

#### a) Các mức khác nhau của DFD

DFD cấp cao nhất, gọi là sơ đồ dòng dữ liệu cấp 0 được xem như một sơ đồ toàn cảnh (context diagram). Các sơ đồ dòng dữ liệu sẽ được lần lượt xây dựng từ trên xuống (top - down) bằng cách triển khai mỗi quá trình hay phép biến đổi của DFD cấp cao hơn thành một sơ đồ dòng dữ liệu chi tiết hơn cấp thấp hơn cho tới khi các phép biến đổi trở thành nguyên tố (nghĩa là chỉ thực hiện một tác vụ duy nhất).

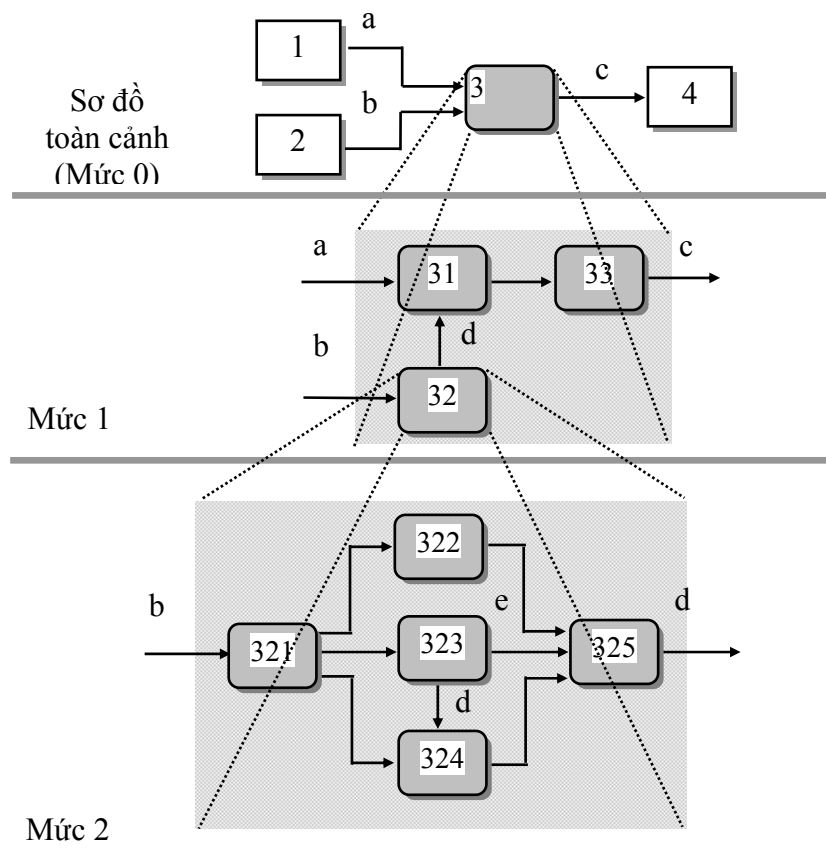
Tại mỗi mức, mỗi DFD có thể được vẽ trên một trang giấy (khổ A4 chẳng hạn).

#### b) Các nguyên tắc

Trong khi triển khai các mức DFD, cần tuân theo các nguyên tắc sau đây :

Cân đối giữa DFD cấp cao (DFD mẹ) và DFD cấp thấp hơn (DFD con) :

- Đầu vào và đầu ra của DFD con phải đồng nhất với đầu vào và đầu ra của phép biến đổi trong DFD mẹ mà từ đó DFD con được triển khai.
- Triển khai từng cấp theo chiều ngang, không theo chiều dọc.
- Tránh vẽ các lưu đồ quá lớn : một lưu đồ chỉ nên có tổng số các phép biến đổi và kho dữ liệu không quá 7.



*Hình 3.27 Sơ đồ nhiều mức của DFD*

Từ mỗi phép biến đổi, xét xem nó đã là nguyên tố chưa, nếu chưa, lặp lại từ việc liệt kê sự kiện và đáp ứng, bổ sung vào danh mục sự kiện. Xem phép biến đổi như một hệ thống và tiếp tục triển khai cấp dưới.

Bảo toàn dữ liệu : không có phép biến đổi hay kho dữ liệu nào có thể cho ra dữ liệu mà không nhận dữ liệu dữ liệu vào.

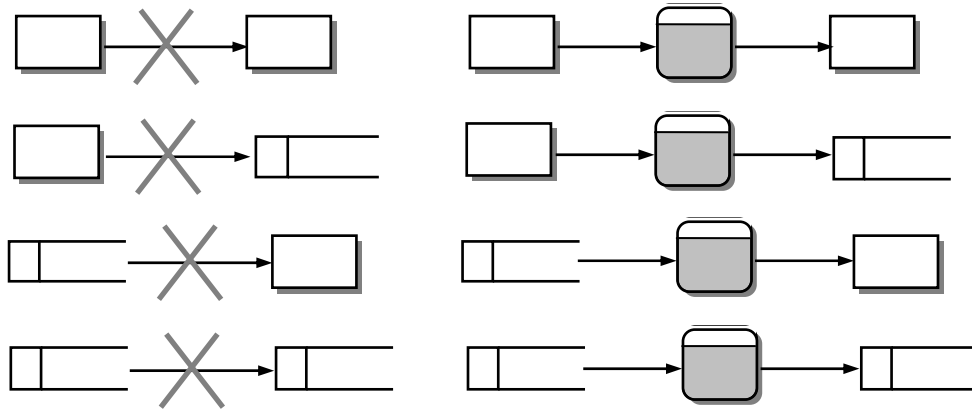
Tránh các hố đen (black hole) : Không có phép biến đổi chỉ nhận dữ liệu vào mà không cho dữ liệu ra. Không có kho dữ liệu chỉ ghi mà không có truy xuất, nếu có các tệp cơ sở dữ liệu được lưu trữ dự phòng (backup dbf) thì phải thể hiện dòng dữ liệu phục hồi (restore).

Tuy nhiên, trong thực tiễn, không nên thể hiện các tệp backup trong DFD vì đó chỉ là biện pháp mang tính đối phó và làm rắc rối DFD.

Miêu tả dòng dữ liệu chứ không phải dòng điều khiển. Tránh các dòng dữ liệu "xuyên qua" là những dòng dữ liệu vào và ra khỏi một phép biến đổi mà không bị biến đổi gì cả.

Ưu tiên khai báo kho dữ liệu ở cấp cao nhất trong đó nhiều phép biến đổi tham chiếu đến.

Mọi DFD phải bắt đầu và/hoặc kết thúc từ một quá trình. Trong hình vẽ dưới đây, cột bên trái là vẽ sai vì vi phạm quy tắc, cột bên phải là vẽ đúng quy tắc.



Hình 3.28 Quy tắc vẽ DFD (cột bên phải là vẽ đúng)

### c) Đặt tên và đánh số các thành phần của DFD

Các dòng dữ liệu, kho dữ liệu và các phép biến đổi đều phải được đặt tên. Tên là danh từ hoặc cụm danh từ.

Các quá trình là các động từ hoặc cụm động từ. Không sử dụng các từ đồng nghĩa hoặc vô ích trong DFD

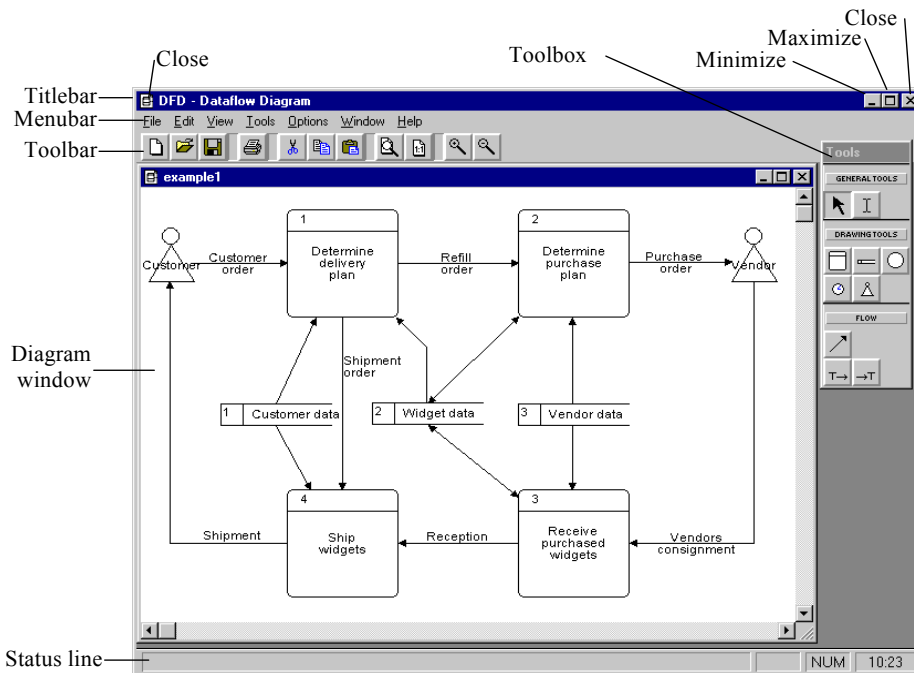
Sơ đồ ngữ cảnh : thường chỉ có một quá trình được đánh số là 0

Mức 1 được đánh số là 1, 2, 3...

Mức 2 được đánh số ví dụ cho quá trình 2 : 2.1, 2.2, 2.3...

Một cách tổng quát, mức DFD của quá trình i được đánh số i.1, i.2...

### II.3.5.Trình soạn thảo PPP DFD editor



Hình 3.29 Màn hình trình soạn thảo DFD (PPP DFD editor)

PPP DFD editor là trình soạn thảo DFD được download từ Internet, do Khoa Khoa học máy tính và Thông tin, viện Đại học Khoa học và Kỹ thuật Na Uy (Department of Computer and Information Science of the Norwegian University of Science and Technology-NTNU).

PPP DFD editor version 1.10 cho phép dễ dàng tạo ra các mô hình ý niệm DFD. Sau đây là giao diện (cửa sổ) ứng dụng của PPP DFD editor.

Tài liệu PPP DFD editor User's Guide kèm hệ thống cho phép sử dụng PPP DFD editor để tham khảo các hướng dẫn.

*Ví dụ* : Sơ đồ DFD mô tả hệ thống quản lý giáo vụ tại một trường Đại học

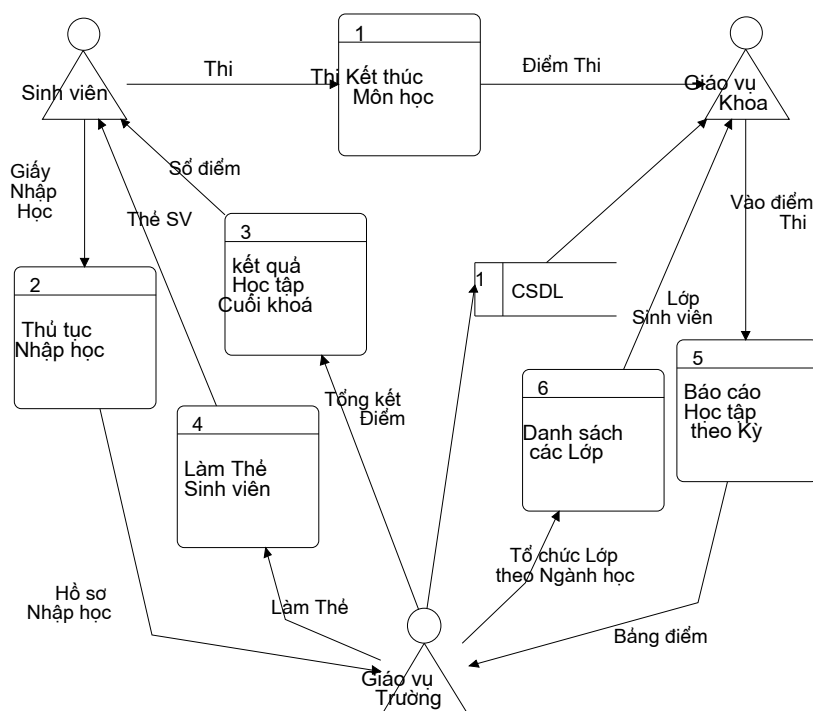
Hệ thống gồm :

3 thực thể là Sinh viên, Giáo vụ trường và Giáo vụ Khoa.

6 quá trình là Thủ tục nhập học, Làm thẻ sinh viên, Kết quả học tập, Danh sách lớp và Báo cáo học tập theo kỳ.

1 CSDL về quản lý giáo vụ

Các dòng thông tin liên kết các thực thể và các quá trình.



Hình 3.30 Sơ đồ DFD biểu diễn bài toán quản lý giáo vụ

### III. Ví dụ : xí nghiệp đóng hộp DanaFood

Mục này trình bày một ví dụ về phân tích hiện trạng của XN đóng hộp DanaFood. XN DanaFood sản xuất các mặt hàng thực phẩm đóng hộp. XN cần áp dụng Tin học để quản lý thị trường, quản lý đơn đặt hàng và quản lý kho lưu trữ.

Sau khi phân tích hiện trạng, NPT xây dựng được sơ đồ DFD và từ điển dữ liệu để phục vụ cho bước xây dựng mô hình ý niệm dữ liệu tiếp theo.

#### III.1. Mô tả hoạt động của xí nghiệp DanaFood

##### a) Các loại sản phẩm

DanaFood sản xuất hai loại sản phẩm : loại sản phẩm theo mùa và các sản phẩm khác. Loại sản phẩm theo mùa được lưu cất vào kho mỗi năm một lần theo đợt sản xuất. Khi hết đợt, phân xưởng sản xuất thông báo cho phòng kinh doanh.

Các sản phẩm khác được sản xuất và lưu cất vào kho khi có nhu cầu. Theo đặt hàng của phòng kinh doanh, phân xưởng sản xuất xác định số lượng cần sản xuất và báo cho phòng kinh doanh thời điểm có hàng.

##### b) Bán buôn

XN DanaFood phục vụ khách hàng theo hai cách: buôn bán theo thoả thuận (giao kèo) và buôn bán bình thường.

Buôn bán thoả thuận không phân biệt loại sản phẩm. Mỗi thoả thuận, qua thương lượng với khách hàng, xác định mà sản phẩm liên quan, giá vốn, số lượng và kỳ hạn giao hàng.

Việc buôn bán bình thường cũng dựa trên cả hai sản phẩm và được ghi chép, xử lý theo tiến độ các đơn đặt hàng của khách hàng gửi đến nhà máy. Mỗi đơn đặt hàng mỗi lần chỉ dựa trên một sản phẩm.

Sự giao hàng được chấp nhận phụ thuộc vào doanh số năm trước và doanh số hàng tháng kể từ đầu năm của khách hàng. Các doanh số này có được từ các thoả thuận và các đơn đặt hàng bình thường.

Một đơn đặt hàng cho sản phẩm theo mùa chỉ có giá trị khi trong kho lưu trữ, số lượng thực có ít hơn số lượng dành cho thoả thuận và cao hơn (hoặc bằng) số lượng đặt hàng.

Với các loại sản phẩm khác, đơn đặt hàng luôn luôn được chấp nhận, vì rằng các sản phẩm này có thể được sản xuất ngay tại mọi thời điểm.

##### c) Giao hàng

Cách giao hàng phụ thuộc vào loại sản phẩm.

Loại sản phẩm theo mùa

Có hai trường hợp : giao hàng theo thoả thuận và giao hàng buôn bán bình thường. Việc giao hàng theo thoả thuận tiến hành theo thời hạn quy định với số lượng tương ứng đã được ấn định. Số lượng đặt trước (giữ trước) này được tích lũy lại và được vào sổ lưu trữ. Nếu là buôn bán bình thường, chỉ giao hàng các sản phẩm theo mùa khi đơn đặt hàng đã được phê duyệt.

Các loại sản phẩm khác

Nếu là buôn bán bình thường thì việc giao hàng được thực hiện ngay lập tức nếu kho lưu trữ còn đủ, nếu không, có hai trường hợp xảy ra :

Nếu tổng cộng (số lượng tồn kho + số lượng đang sản xuất) là đủ, thì sẽ giao hàng ngay khi sản lượng đang sản xuất vừa đủ.

Nếu mặt hàng này chưa được sản xuất, thì phân xưởng sản xuất sẽ xác định số lượng cần sản xuất và quay lại trường hợp trên.

Trong cả hai trường hợp, khách hàng được báo trước thời hạn giao nhận hàng.

Chú ý rằng với mỗi sản phẩm loại này cần xác định mức báo động về số lượng lưu trữ. Phân xưởng sản xuất sẽ căn cứ vào trạng thái kho lưu trữ được cập nhập hàng ngày mà có biện pháp sản xuất kịp thời, đó là việc so sánh giữa số lượng thực còn và số lượng báo động.

### **III.2. Giao tiếp giữa cơ sở dữ liệu với người sử dụng**

Tập hợp thông tin về quản lý đơn đặt hàng, giao hàng và quản lý sản xuất, lưu trữ phải được truy cập không những bởi phòng Kinh doanh, các phân xưởng mà còn bởi phòng Kế toán Thống kê và phòng Giao nhận hàng.

Chẳng hạn, NPT có thể đặt ra những câu hỏi sau :

Với mỗi đơn đặt hàng, chẳng hạn đơn đặt hàng số 4123, hãy cho biết thông tin về khách hàng, về tỷ lệ giảm giá ?

Với mỗi khách hàng, hãy cho biết các thoả thuận đã có và ngày giao hàng tương ứng.

Với mỗi sản phẩm theo mùa, ví dụ “Dưa chuột tươi” có mã RQ-37 và số lượng 1 kg, hãy cho biết thông tin về khách hàng đã mua theo thoả thuận ?

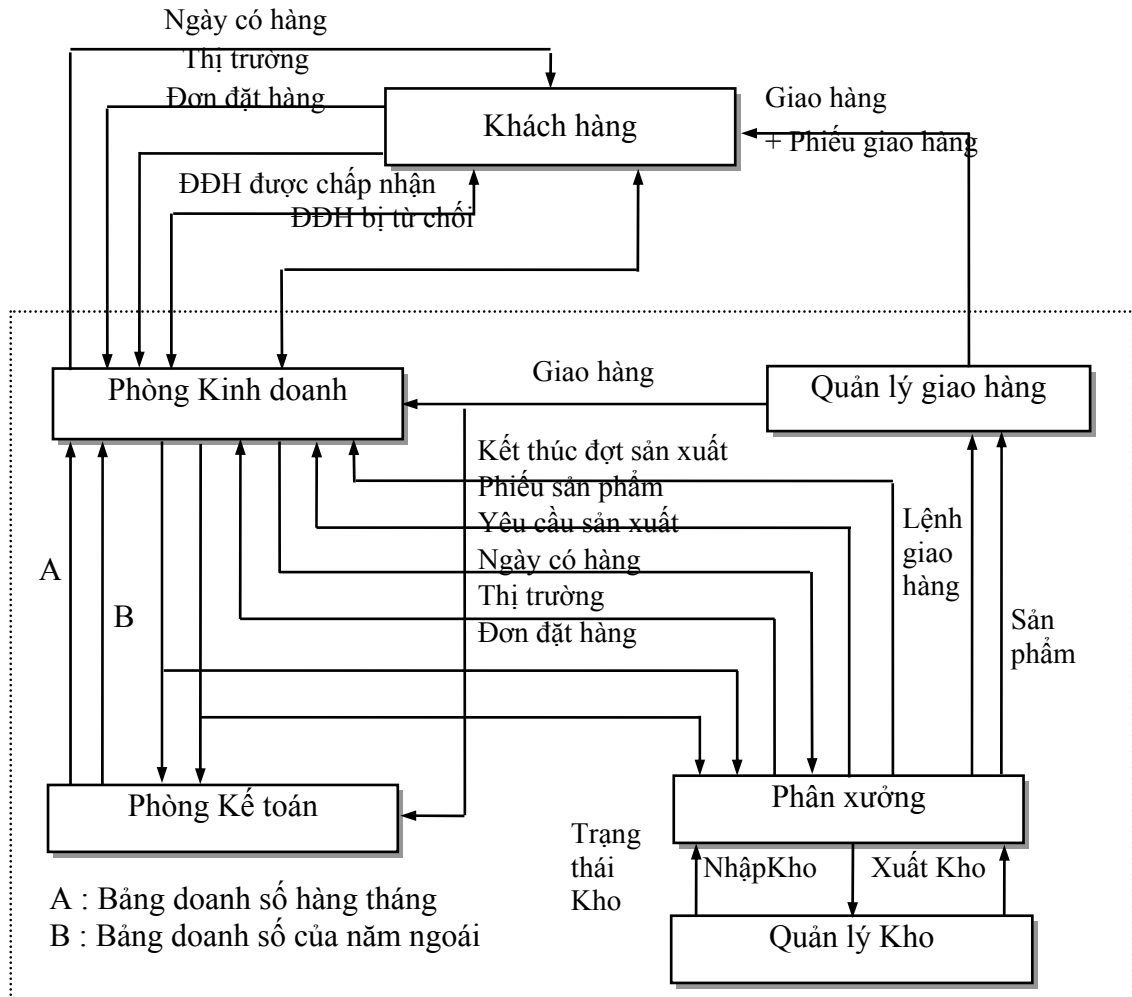
Với mỗi sản phẩm không theo mùa, ví dụ “Cá hộp” có mã sản phẩm DH-03 và số lượng 200g, hãy cho biết thông tin về khách hàng đang chờ giao hàng theo đơn đặt hàng ?

### III.3. Phân tích các dòng thông tin

Nhờ các cuộc phỏng vấn, nghiên cứu hồ sơ nội bộ và nghiên cứu dòng luân chuyển thông tin tại XN, NPT lập ra được hai hồ sơ phân tích như sau :

#### a) Sơ đồ dòng thông tin

Sơ đồ DFD thể hiện sự trao đổi các dòng thông tin trong XN Dana Food.



Hình 3.31 Sơ đồ DFD của XN Dana Food

#### b) Bảng các dòng thông tin

Sau đây ta xây dựng bảng các dòng thông tin làm cơ sở cho việc phân tích dữ liệu và phân tích xử lý tiếp theo.



Mỗi dòng thông tin có một số thứ tự (cột \*) cho biết nguồn và đích và dòng thông tin đi sát trước đó. Trong mỗi dòng thông tin, các dữ liệu có giá trị lặp lại nhiều lần thì được đặt trong cặp dấu { }, các dữ liệu đã có mặt ở một dòng thông tin khác trước đó thì được đặt trong cặp dấu [ ].

Stt	Dòng	(*)	Dữ liệu	Nguồn → Đích	
1	Kết thúc đợt sản xuất		MãHàng	Phân xưởng	Phòng Kinh doanh
2	Nơi bán hàng (NB)	1	NơiBánHg MãHàng, NgàyBán SLBán, GiáBán HạnGiaoHg TênKH, ĐịaChiKH	Khách hàng (KH)	Phòng Kinh doanh
3	Thị trường được đăng ký	2	[Thị trường]	Phòng Kinh doanh	Phân xưởng & Phòng Kế toán
4	Đơn đặt hàng (ĐDH)		TênKH ĐịaChiKH NgàyĐặtHg, MãHàng SLĐặt, GiáTiền	Khách hàng	Phòng Kinh doanh
5	ĐDH được đăng ký (ĐK)	4	[ĐDH] + MãKH SốHĐĐặtHg, STGiảm	Phòng Kinh doanh	Phân xưởng & Phòng Kế toán
6	Yêu cầu sản xuất	5	MãHàng SLSX	Phòng Kinh doanh	Phân xưởng
7	Sản phẩm đã có	6	MãHàng SLSX, NgàyCóHg	Phân xưởng	Phòng Kinh doanh
8	ĐDH được chấp nhận	5	MãKH SốHĐĐặtHg	Phòng Kinh doanh	Khách hàng
9	ĐDH bị từ chối (trả lại)	4	TênKH	Phòng Kinh doanh	Khách hàng
10	Trạng thái Kho		{MãHàng} {SLCó} {SLBáoĐộng}	Quản lý Kho	Phân xưởng
11	Lệnh giao hàng	2 4	NơiBánHg SốĐK	Phân xưởng	Quản lý Giao hàng
12	Phiếu giao hàng (PGH)	11	SốHĐĐặtHg SốHĐCấpHg NgàyGiao	Quản lý Giao hàng	Khách hàng
13	Phiếu giao hàng	12	[PGH)	Quản lý Giao hàng	Phòng KD & Phòng Kế toán
14	Bảng doanh số năm – 1		{MãKH} {TênKH} {DSốNăm-1}	Phòng Kế toán	Phòng Kinh doanh
15	Bảng doanh số tháng		{MãKH} {TênKH} {DSốTháng} {Số tháng}	Phòng Kế toán	Phòng Kinh doanh
16	Phiếu sản phẩm		MãHàng, GiáĐơnVị SLBáoĐộng SLCó, SLDựTrữ	Phân xưởng	Phòng Kinh doanh

Ví dụ 2 : Danh sách dữ liệu sơ cấp thu lượm được từ XN DanaFood :

Stt	Dữ liệu	Giải thích
1	DSốNăm-1	Doanh số của khách hàng năm ngoái
2	DSốTháng	Doanh số hàng tháng của khách hàng từ đầu năm nay
3	NgàyĐặtHg	Ngày đặt hàng của khách hàng
4	NgàyCóHg	Ngày xuất hiện mặt hàng (sau khi sản xuất)
5	NgàyKThúc	Ngày kết thúc sản xuất mặt hàng
6	NgàyGiao	Ngày giao hàng
7	NgàyBán	Ngày bán hàng
8	HạnGiaoHg	Hạn giao hàng nơi bán
9	ChỉDẫnHg	Chỉ dẫn về mặt hàng
10	TênTháng	Tên tháng
11	GiaTiền	Gia tiền của đơn đặt hàng của khách hàng
12	SốHĐCấpHg	Số hoá đơn cấp hàng
13	SốHĐĐặtHg	Số hoá đơn đặt hàng
14	MãKH	Mã khách hàng do XN ấn định
15	NơiBánHg	Nơi bán hàng
16	Tháng	Tháng thứ (1..12)
17	TênKH	Tên khách hàng
18	GiáNơiBán	Giá bán thực sự của mặt hàng tại nơi bán
19	GiáĐơnVị	Giá đơn vị của mặt hàng do XN ấn định
20	SLĐặt	Số lượng đặt hàng đặt hàng của khách hàng
21	SLSX	Số lượng sản xuất của mặt hàng
22	SLBán	Số lượng bán của mặt hàng tại nơi bán
23	MãHàng	Mã mặt hàng do XN ấn định
24	ĐịaChỉKH	Địa chỉ khách hàng
25	SLBáoĐộng	Số lượng báo động sắp cạn của mặt hàng
26	SLCó	Số lượng mặt hàng hiện có trong kho
27	ĐiệnThKH	Số điện thoại của khách hàng

c) Ví dụ từ điển dữ liệu của DanaFood

STT	Dữ liệu	Loại	Mô tả kiểu dữ liệu	Công thức
1	ChiDẫnHg	KTT	C20	CT1
2	DSốNăm-1	TT	N12	
3	DSốTháng	TT	N10	
4	ĐịaChỉKH	TT	C30	
5	ĐiệnThKH	KTT	C10	CT3
6	GiáĐơnVị	TT	N10	
7	GiáNơiBán	KTT	N10	
8	GiaTiền	TT	N10	
9	HạnGiaoHg	KTT	N4	
10	MãHàng	KTT	C5	
11	MãKH	KTT	C5	
12	NơiBánHg	KTT	C20	
13	NgàyĐặtHg	KTT	D	
14	NgàyBán	KTT	D	
15	NgàyCóHg	KTT	D	
16	NgàyGiao	KTT	D	
17	NgàyKThúc	KTT	D	
18	SốHĐĐặtHg	KTT	N4	
19	SốHĐCấpHg	KTT	N4	
20	SLĐặt	TT	N4	CT4
21	SLBán	KTT	N4	
22	SLBáoĐộng	KTT	N4	
23	SLCó	TT	N4	
24	SLDựTrữ	TT	N4	CT5
25	SLSX	KTT	N4	CT6
26	STGiảm	TT	N10	
27	TênKH	KTT	C20	
28	TênTháng	KTT	C6	
29	Tháng	KTT	N2	

Công thức tính toán :

CT1 :  $DSốNăm-1 = SUM(DSốTháng \text{ của năm trước đó})$ .

CT2 : Nhận được bằng cách cộng dồn các GiáTiền từng tháng của năm hiện hành.

CT3 :  $GiaTiền = SLĐặt \times GiáĐơnVị$  của sản phẩm tương ứng.

CT4 : Được cập nhật mỗi lần xuất / nhập kho.

CT5 :  $SLDựTrữ = SUM(\text{số lượng yêu cầu trên thị trường theo sản phẩm})$ .

CT6 :  $STGiảm = GiáTiền \times tỷ\_lệ\_giảm$ .

Tỷ lệ giảm được xác định như sau :

CộngDồnDsốTháng SốTháng	DsốNăm-1		
	≥ 120 000 000	≥ 100 000 000	< 100 000 000
≥ 10 000 000	10%	8%	5%
≥ 8 000 000	8%	5%	2%
< 8 000 000	2%	1%	0%

Thuật toán như sau :

$tmp = \text{CộngDồnDsốTháng} / \text{SốTháng}$

IF DsốNăm-1 >= 120 000 000

THEN IF tmp >= 10 000 000

THEN STGiảm = GiáTiền \* 10%

ELSE IF tmp >= 8 000 000

THEN STGiảm = GiáTiền \* 8%

ELSE STGiảm = GiáTiền \* 2%

ENDIF

ENDIF

ELSE IF DsốNăm-1 >= 100 000 000

THEN IF tmp >= 10 000 000

THEN STGiảm = GiáTiền \* 8%

ELSE IF tmp >= 8 000 000

THEN STGiảm = GiáTiền \* 5%

ELSE STGiảm = GiáTiền \* 1%

ENDIF

ENDIF

ELSE IF tmp >= 10 000 000

THEN STGiảm = GiáTiền \* 5%

ELSE IF tmp >= 8 000 000

THEN STGiảm = GiáTiền \* 2%

ELSE STGiảm = 0%

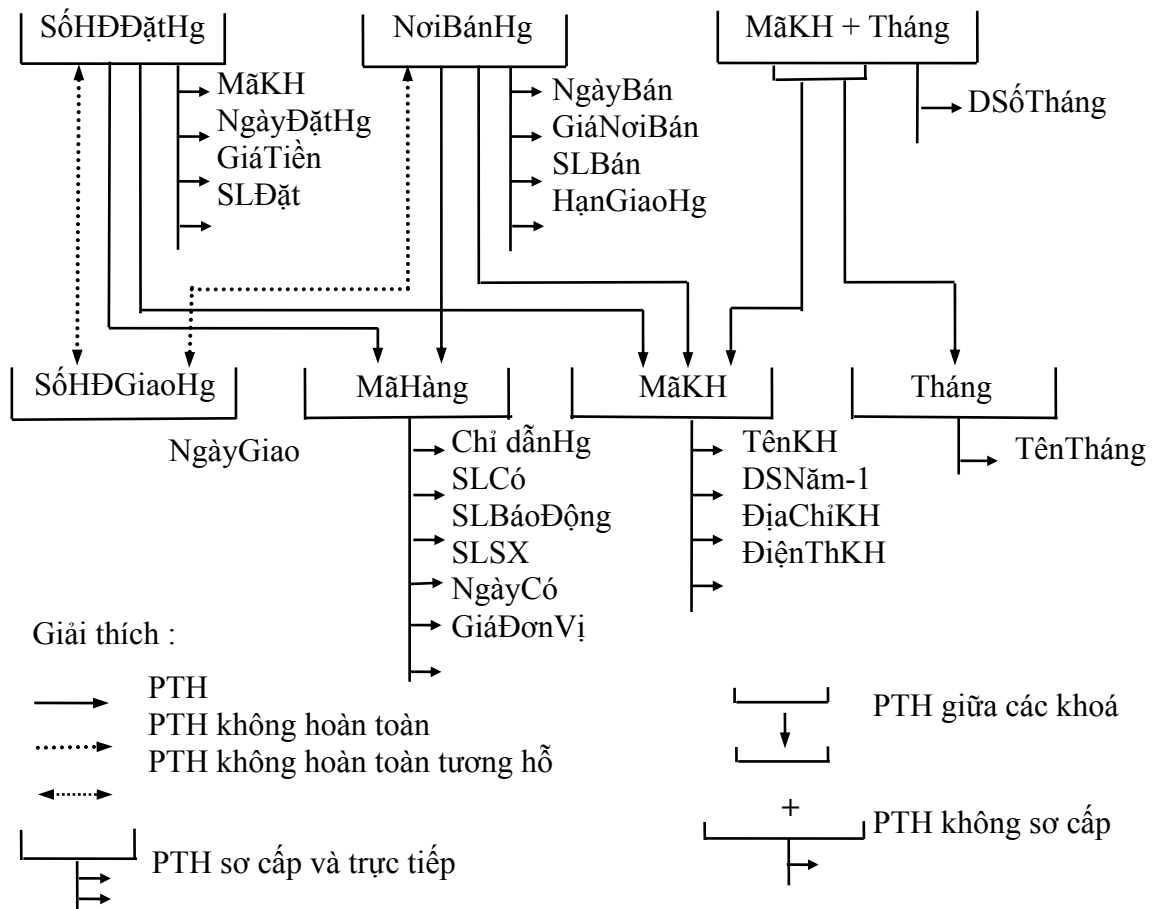
ENDIF

ENDIF

ENDIF

ENDIF

d) Đồ thị các PTH biểu diễn CSDL của nhà máy đóng hộp DanaFood



**Hình 3.10 Đồ thị các PTH của CSDL DanaFood**

## Bài tập chương 3

---

- a) Các bài tập sau đây yêu cầu sử dụng công cụ soạn thảo DFD PPP DFD editor để vẽ các sơ đồ dòng dữ liệu. Chú ý đọc kỹ yêu cầu và tìm hiểu bài toán thực tế.

### *1. Hệ thống quản lý đăng ký bầu cử*

Ủy ban Nhân dân (UBND) thành phố N. chuẩn bị cho một đợt bầu cử hội đồng thành phố. UBND sử dụng các phương tiện tuyên thông đại chúng như báo, đài, tivi, ... để thông báo về đợt đăng ký bầu cử.

Cử tri đến đăng ký bầu cử phải xuất trình giấy tờ phòng thân (CMND, hộ chiếu, giấy khai sinh...). Yêu cầu của Ban Tổ chức bầu cử là mọi đối tượng từ 18 tuổi trở lên đã có hộ khẩu tại thành phố hoặc những người mới chuyển đến cư trú trong thành phố.

Vẽ sơ đồ luân chuyển các dòng thông tin và bảng các dòng liên quan đến đợt đăng ký bầu cử của thành phố N. Danh sách các thành phần tổ chức liên quan trong hệ thống gồm :

- Hội đồng UBND
- Ban quản lý hộ khẩu
- Văn thư UB
- Ban Tổ chức bầu cử
- Các Cơ quan báo, đài...
- Cử tri

### *2. Hệ thống tiếp thị và bán sản phẩm*

Vẽ sơ đồ luân chuyển các dòng thông tin và bảng các dòng liên quan đến hệ thống tiếp thị và bán sản phẩm của một công ty X.

### *3. Hệ thống quản lý tồn kho vật tư*

Vẽ sơ đồ luân chuyển các dòng thông tin và bảng các dòng liên quan đến hệ thống quản lý tồn kho vật tư tại một công ty X.

- b) Xây dựng từ điển dữ liệu cho các bài toán trên đây.

## CHƯƠNG 4

### Phân tích ý niệm dữ liệu và các phương pháp mô hình hoá

Khi phân tích và thiết kế các HTTT, người ta đưa vào ba loại hiện tượng thực (real phenomena) tương tác lẫn nhau để thể hiện sự tương tác của thế giới thực :

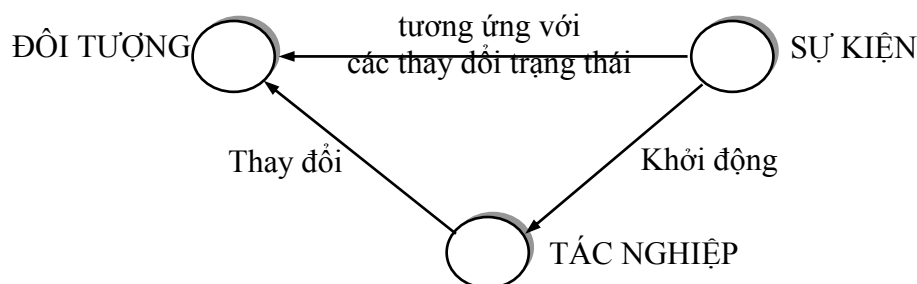
**ĐÔI TƯỢNG** thể hiện các lớp thực thể cụ thể hay trừu tượng của hệ thống đang xét, ví dụ SẢN PHẨM, KHÁCH HÀNG, ĐƠN HÀNG...

**TÁC NGHIỆP** thể hiện các hoạt động tác nghiệp của hệ thống, ví dụ phân tích một đơn đặt hàng, thanh toán hoá đơn...

**SỰ KIỆN** thể hiện các sự kiện xảy ra trong một hệ thống, ví dụ nhận được đơn đặt hàng gọi tới, hoá đơn đến hạn phải thanh toán...

Người ta xem trạng thái của hệ thống tại một thời điểm đang xét là trạng thái của các đối tượng của hệ thống. Hệ thống phát triển theo thời gian nhờ các hoạt động tác nghiệp (sản xuất, kinh doanh...). Các tác nghiệp được khởi động bởi các sự kiện diễn ra từ bên trong hoặc bên ngoài của hệ thống. Các hoạt động tác nghiệp làm thay đổi các trạng thái và đến lượt sự thay đổi các trạng thái này dẫn đến các sự kiện.

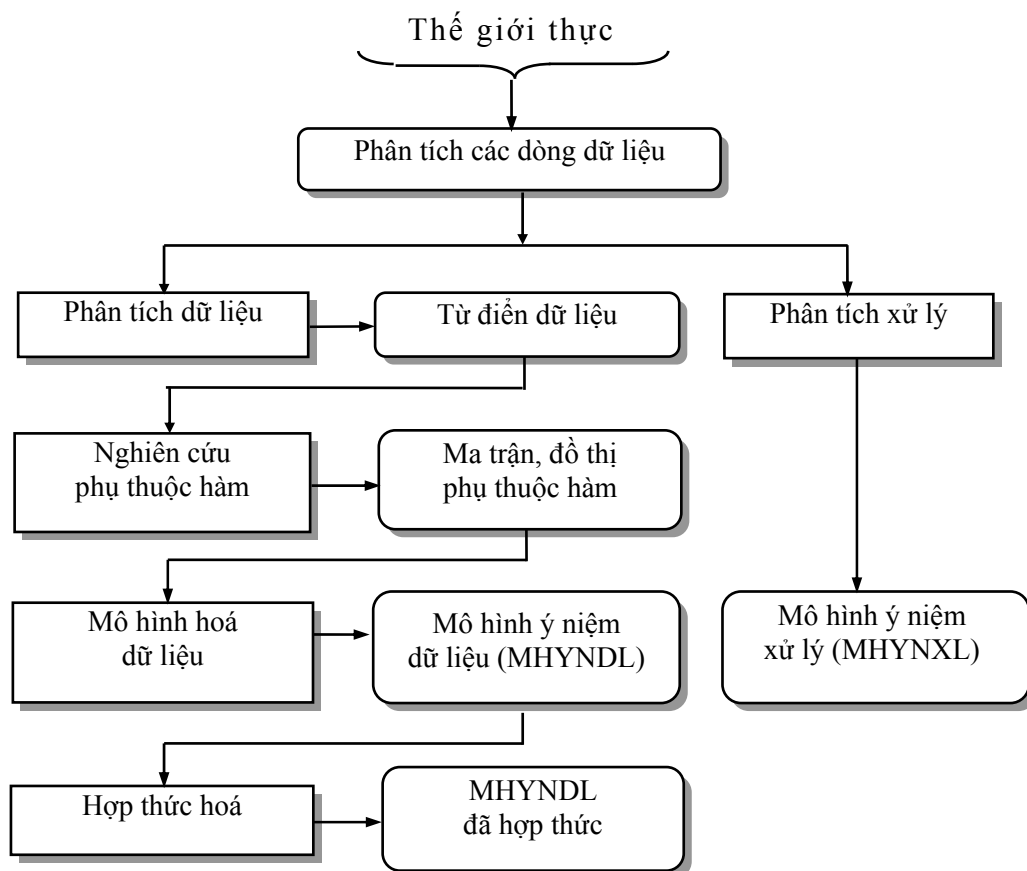
Sự tương tác giữa ba phạm trù được biểu diễn như sau :



Hình 4.32 Sự tương tác giữa ba hiện tượng thực

Chương này trình bày một phương pháp mô hình hóa dữ liệu, gọi là *phương pháp phân tích tiến* hay *phân tích từ dưới lên*. Cơ sở của phương pháp là xuất phát từ những dữ liệu thô thu nhận được từ hệ thống thông tin đang xét, phân tích các phụ thuộc hàm để xây dựng một mô hình ý niệm dữ liệu.

Hai ví dụ minh họa cho chương này là xí nghiệp chế biến thực phẩm DanaFood và khu du lịch Non Nước. Các giai đoạn mô hình hóa dữ liệu theo phương pháp tiến được giới thiệu trong hình sau đây :



Hình 4.33 Phương pháp phân tích từ dưới lên

## I. Mô hình thực thể - kết hợp

### I.1. Khái niệm về mô hình thực thể - kết hợp

#### I.1.1. Khái niệm về thực thể

*Thực thể* theo định nghĩa của từ điển tiếng Việt là *cái có sự tồn tại độc lập*. Ví dụ : con người là một thực thể xã hội. Như vậy, thực thể là một đối tượng (object) cụ thể hay trừu tượng của thế giới thực, có sự tồn tại ổn định theo thời gian để có thể được ghi nhận, biểu diễn và xử lý trong HTTT.

Ví dụ :

DH025 là Bia chai Tiger.

Số lượng của Bia chai Tiger là 10.

Mỗi thực thể có thể có một hoặc có nhiều *đặc tính* (Property) hay *thuộc tính* (Attribute). Mỗi đặc tính đặc trưng cho một khía cạnh thực của thực thể trong HTTT và được chỉ định bởi một *tên gọi* và một *giá trị*.

Ví dụ :

Địa chỉ của Mơ là 54 Lê Lợi Đà Nẵng



Tên mặt hàng của đơn đặt hàng số DH025 là Bia chai Tiger

Số lượng của Bia chai Tiger là 10

Tên đặc tính cũng là tên của một *kiểu đặc tính* (property type). Một kiểu đặc tính có thể thuộc về nhiều thực thể.

*Ví dụ :*

Địa chỉ của sinh viên

Tên mặt hàng của đơn đặt hàng

Số lượng của mặt hàng bán

Một *kiểu thực thể* (entity type) là một tập hợp các thực thể được xác định bởi cùng một tập hợp các kiểu đặc tính đóng giống nhau trong HTTT. Kiểu thực thể thể hiện một lớp đối tượng thực trong thế giới thực. Trong thực tế, người ta hay nhầm lẫn giữa thực thể và kiểu thực thể.

*Ví dụ :*

Kiểu thực thể Sinh viên

Kiểu thực thể Đơn đặt hàng

Kiểu thực thể Mặt hàng bán

Tuy nhiên, để đơn giản, người ta nói thực thể Sinh viên để chỉ định kiểu thực thể Sinh viên, v.v...

Một *trường hợp cụ thể* (occurrence) của một kiểu thực thể được tạo thành từ tập hợp các giá trị mà các kiểu đặc tính của kiểu thực thể này có thể lấy.

*Ví dụ :*

Giả sử kiểu thực thể Sinh viên gồm các kiểu đặc tính Tên, Phái, Địa chỉ và Ngành học, ta có các trường hợp cụ thể như sau :

STT	Tên	Phái	Địa chỉ	Ngành học
1.	Mơ	Nữ	54 Lê Lợi Đà Nẵng	Tin học
2.	Mận	Nữ	12/7 Phú Cam, tp. Huế	Điện tử
3.	Đào	Nữ	Quế an, Quế sơn, Quảng nam	Kế toán

Trong các kiểu đặc tính của một kiểu thực thể, người ta nhấn mạnh một kiểu đặc tính được gọi là *khoá* hay *định danh* (identifier).

*Ví dụ :*

Tên của sinh viên là khoá của kiểu thực thể Sinh viên

Số của đơn đặt hàng là khoá của kiểu thực thể Đơn đặt hàng

Tên của mặt hàng là khoá của kiểu thực thể Mặt hàng bán

Khoá của một kiểu thực thể cho phép phân biệt các trường hợp cụ thể khác nhau của kiểu thực thể đó. Khi gán một giá trị cụ thể cho khoá, ta có thể nhận được duy nhất một thực thể. Việc xác định khoá cho thực thể không phải luôn luôn dễ dàng. Trong một kiểu thực thể, có thể chọn được nhiều kiểu đặc tính dùng làm khoá nhờ đặc trưng của bản thân dữ liệu (giá trị của đặc tính). Chẳng hạn, đối với một thực thể khách hàng, vị trí xã hội cũng như số điện thoại của khách hàng đều có thể chỉ định được khách hàng đó.

Tuy nhiên có thể xảy ra trường hợp không có kiểu đặc tính nào của kiểu thực thể được chọn làm khoá. Trong kiểu thực thể Sinh viên ở trên, nếu ta chọn khoá là Tên thì sẽ không hợp lý vì có thể xảy ra sự trùng tên. Nhiều sinh viên có thể có cùng một tên. Điều gì sẽ xảy ra nếu có sự nhầm lẫn giữa hai sinh viên trong việc quản lý đào tạo ? Lúc này, cần đưa ra được ít nhất một khoá thỏa mãn những điều kiện xử lý tin học của hệ thống.

Ví dụ, người ta thường đưa vào Mã sinh viên làm khoá cho kiểu thực thể Sinh viên, chẳng hạn thực thể sinh viên Mơ có mã là DDK10996027, thực thể sinh viên Mận có mã là DDK10396054, v.v...

Để dễ dàng thiết kế các cơ sở dữ liệu quan hệ về sau, người ta đưa vào hai ràng buộc (constraints) cho các kiểu đặc tính là *tính duy nhất* (uniqueness) và *tính sơ cấp* (elementarity)

Tính duy nhất giá trị của các đặc tính bắt buộc mỗi đặc tính của một thực thể chỉ nhận một và chỉ một giá trị, không cho phép có nhiều giá trị. Ví dụ kiểu đặc tính Ngành học của thực thể sinh viên Đào không thể lấy giá trị nào khác ngoài giá trị “Kế toán”.

Tính sơ cấp của kiểu đặc tính có nghĩa là không thể định nghĩa một hoặc nhiều kiểu đặc tính này từ những kiểu đặc tính khác bằng cách tổ hợp hoặc rút gọn. Ví dụ kiểu đặc tính Địa chỉ của kiểu thực thể Sinh viên không thể xác định từ các kiểu đặc tính Số nhà, Phố và Tỉnh thành. Tính sơ cấp là bắt buộc cho mọi kiểu đặc tính của một kiểu thực thể.

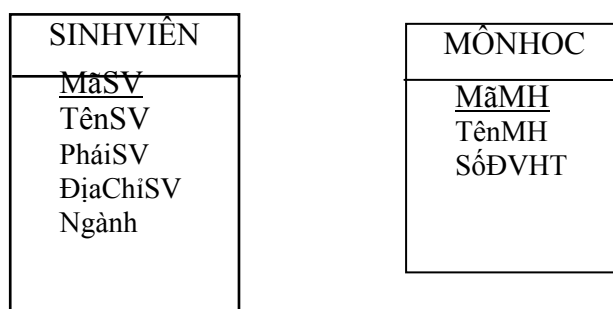
Các kiểu thực thể được biểu diễn bởi các hình chữ nhật gồm hai phần, phần trên chứa tên của kiểu thực thể và phần dưới chứa danh sách các kiểu đặc tính của kiểu thực thể đó.

Tên của kiểu thực thể được quy ước viết toàn chữ hoa, ví dụ SINHVIÊN. Tên của các kiểu đặc tính viết hoa thường xen kẽ giống tên riêng (proper), ví dụ TênSV. Riêng tên kiểu đặc tính là khoá có gạch chân, ví dụ MãSV. Các loại tên đều không chứa dấu cách, tuy nhiên có thể sử dụng dấu gạch dưới dòng (\_) để thay cho dấu cách, chẳng hạn Tên\_SV.

Để dễ đọc, tên dữ liệu liên quan đến thực thể và kết hợp đều có dấu tiếng Việt.

Ví dụ :

Dưới đây là các kiểu thực thể SINHVIÊN và MÔNHOc :



Hình 4.34 Biểu diễn đồ thị của kiểu thực thể

### 1.1.2. Khái niệm về kết hợp

Trong cấu trúc kiểu thực thể, các dữ liệu không phản ánh được những mối liên hệ (nếu có) giữa chúng. Tồn tại những mối liên hệ hay sự phụ thuộc về tác nghiệp giữa các khóa của nhiều kiểu thực thể.

Như vậy, *kết hợp* (association) là mối liên hệ hay sự phụ thuộc giữa các thực thể với nhau trong đó mỗi thực thể đóng một vai trò xác định. Kết hợp phản ánh một tình huống thực trong HTTT đang xét.

Ví dụ :

Sinh viên Mơ học môn Phân tích và Thiết kế Hệ thống

Sinh viên Đào ở (thuộc về) khoa Kế toán

Khách hàng Hải Vân yêu cầu Đơn đặt hàng số DH025

Một *kiểu kết hợp* (association type) là một tập hợp các kết hợp có cùng ngữ nghĩa được định nghĩa giữa nhiều kiểu thực thể. Ta cũng quy ước tên của kiểu kết hợp được viết chữ hoa chữ thường xen kẽ tương tự kiểu đặc tính nhưng in nghiêng.

Ví dụ :

Kết hợp *Học* giữa các kiểu thực thể SINHVIÊN và MÔNHOc.

Kết hợp *Thuộc Về* giữa các kiểu thực thể SINHVIÊN và KHOA

Kết hợp *Yêu Cầu* giữa các kiểu thực thể KHÁCH HÀNG và ĐƠN HÀNG

Một kiểu kết hợp có thể có nhiều kiểu đặc tính. Ví dụ kết hợp *Học* có kiểu đặc tính là *Học Kỳ*, kết hợp *Yêu Cầu* có kiểu đặc tính là *Số Lượng Đặt*, v.v...

Một kiểu kết hợp được gọi là *nhị nguyên* (binary) nếu chỉ có hai kiểu thực thể tham gia vào kết hợp. Đây là một ánh xạ giữa hai tập hợp thực thể hay giữa hai kiểu thực thể tương ứng. Ánh xạ cũng được xem là có hai chiều ngược nhau trong phép kết hợp giữa các thực thể của hai kiểu thực thể.

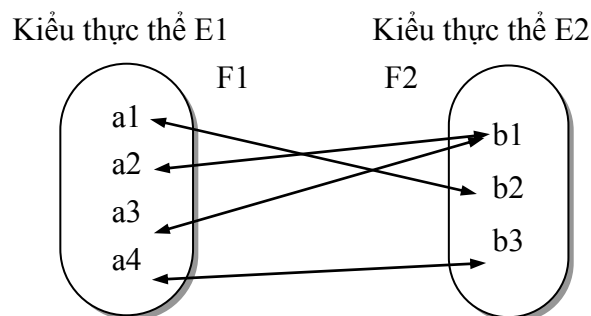
Một kiểu kết hợp có thể được biểu diễn dưới dạng các bảng cụ thể như sau :

	b1	b2	b3	E1	E2			
a1		1		a1	b2	a1	b1	0
a2	1			a2	b1	a1	b2	1
a3	1			a3	b1	a1	b3	0
a4			1	a4	b3	a2	b1	1
						a2	b2	0
						a2	b3	0
						a3	b1	1
						a3	b2	0
						a3	b3	0
						a4	b1	0
						a4	b2	0
						a4	b3	1

Hình 4.4 Biểu diễn dạng bảng của kiểu kết hợp

Người ta đưa vào khái niệm *bản số* (cardinality) hay còn gọi là *lực lượng*. Các bản số thể hiện mối quan hệ tồn tại giữa các trường hợp cụ thể của các kiểu thực thể thông qua các kết hợp. Với mỗi kiểu thực thể và với mỗi kiểu kết hợp liên quan, hai bản số được chỉ định, bản số thứ nhất là *cực tiểu* và bản số thứ hai là *cực đại*.

Giả sử cho hai kiểu thực thể E1 và E2 với hai ánh xạ F1 và F2 ngược nhau qua kết hợp A như hình vẽ dưới đây.



Hình 4.5 Xác định bản số của kiểu kết hợp

Ánh xạ F1 đặt tương ứng mỗi phần tử (hay mỗi trường hợp cụ thể) của E1 với các phần tử của E2 bởi kết hợp A. Ánh xạ F2 đặt tương ứng mỗi phần tử của E2 với các phần tử của E1 bởi kết hợp A.

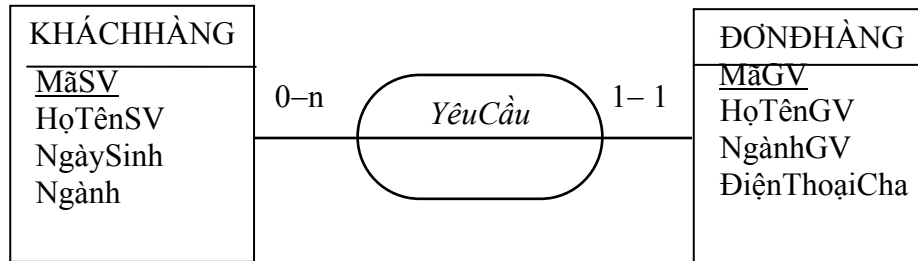
Bản số cực tiểu của F1 là số ảnh nhỏ nhất của F1 (có thể là 0, 1 hoặc  $n > 1$ ), còn bản số cực đại của F1 là số ảnh lớn nhất của F1 (có thể là 0, 1 hoặc  $n$ ). Đối với F1, hai số này thể hiện rằng có ít nhất và có nhiều nhất số phần tử của E2 được kết hợp với một phần tử của E1 trong kết hợp A.

Một cách tương tự ta cũng định nghĩa bản số cực tiểu và bản số cực đại của F2. Như vậy, mỗi kiểu kết hợp được xác định bởi hai cặp bản số được ký hiệu 0-1, 1-1, 0-n hoặc 1-n. Chữ

n thể hiện một số không xác định, lớn hơn hoặc bằng 1. Đôi khi có thể là một con số xác định nhưng trong phân tích ý niệm, người ta không đặt thành vấn đề, miễn là một con số được ghi nhận.

Trong thực tế, người ta dùng đồ thị để biểu diễn kiểu kết hợp bằng cách dùng một hình oval ghi tên kiểu kết hợp và đường nối giữa hai kiểu thực thể liên quan.

Ví dụ :



Hình 4.35 Các bản số của kiểu kết hợp Yêu Cầu

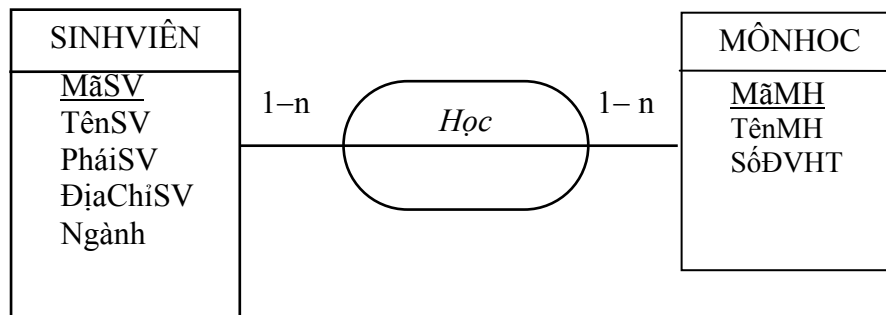
Các bản số của kiểu kết hợp *Yêu Cầu* giữa KHÁCH HÀNG và ĐƠN HÀNG được xác định như sau :

Giữa KHÁCH HÀNG và ĐƠN HÀNG có bản số 0-n vì có thể không có khách hàng nào (ít nhất) và có thể yêu cầu nhiều đơn đặt hàng (nhiều nhất).

Trong một số trường hợp, bản số 1-n với giá trị 1 chỉ ra rằng mọi khách hàng thuộc cơ sở dữ liệu đều đã đặt tối thiểu một đơn đặt hàng.

Bản số giữa ĐƠN HÀNG và KHÁCH HÀNG là 1-1 vì mỗi đơn đặt hàng là của một khách hàng (ít nhất) và chỉ thuộc về một khách hàng mà thôi (nhiều nhất).

Ví dụ :



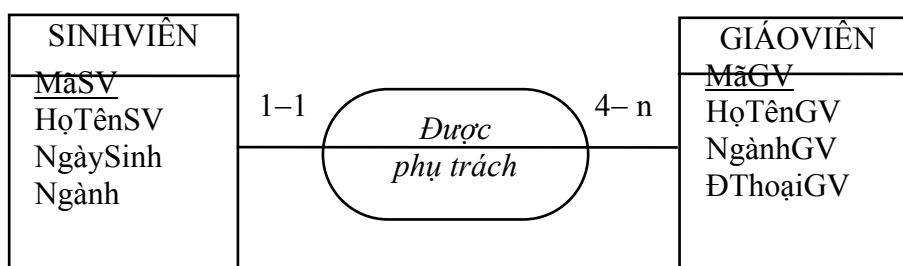
Hình 4.36 Các bản số của kiểu kết hợp Học

Các bản số của kiểu kết hợp *Học* giữa SINHVIÊN và MÔN HỌC được xác định như sau : Giữa SINHVIÊN và MÔN HỌC có bản số 1-n vì mỗi sinh viên đều phải học ít nhất một môn học và sinh viên bắt buộc phải học nhiều môn học.

Giữa MÔN HỌC và SINHVIÊN cũng có bản số 1-n vì mỗi môn học đều có sinh viên theo học và có nhiều sinh viên theo học một môn học.

Khi xảy ra trường hợp ngoại lệ là bản số cực tiểu có giá trị lớn hơn 1, ta sẽ ghi cụ thể số đó.

Ví dụ :



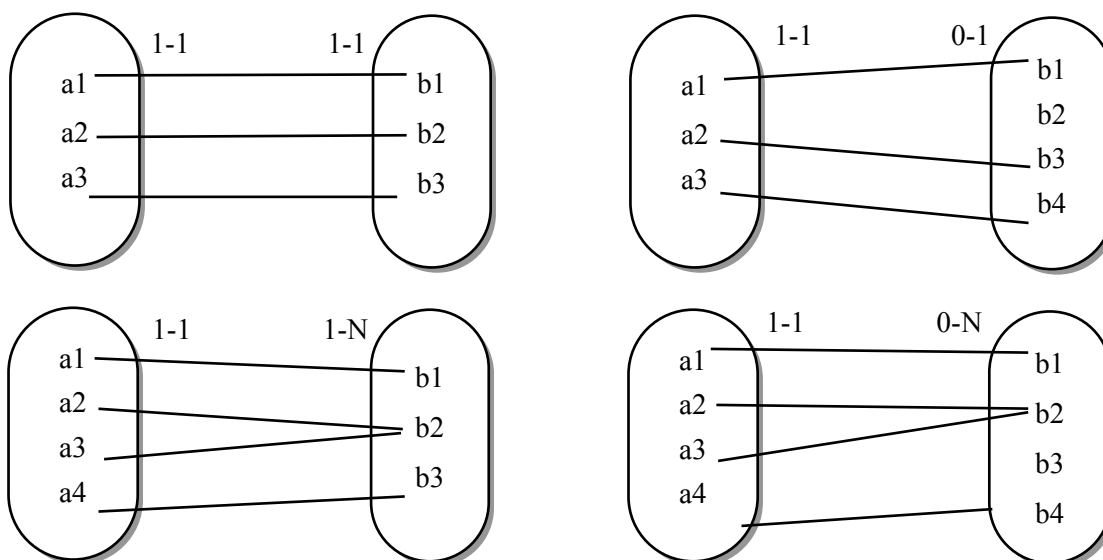
Hình 4.37 Bản số cực tiểu có giá trị lớn hơn 1

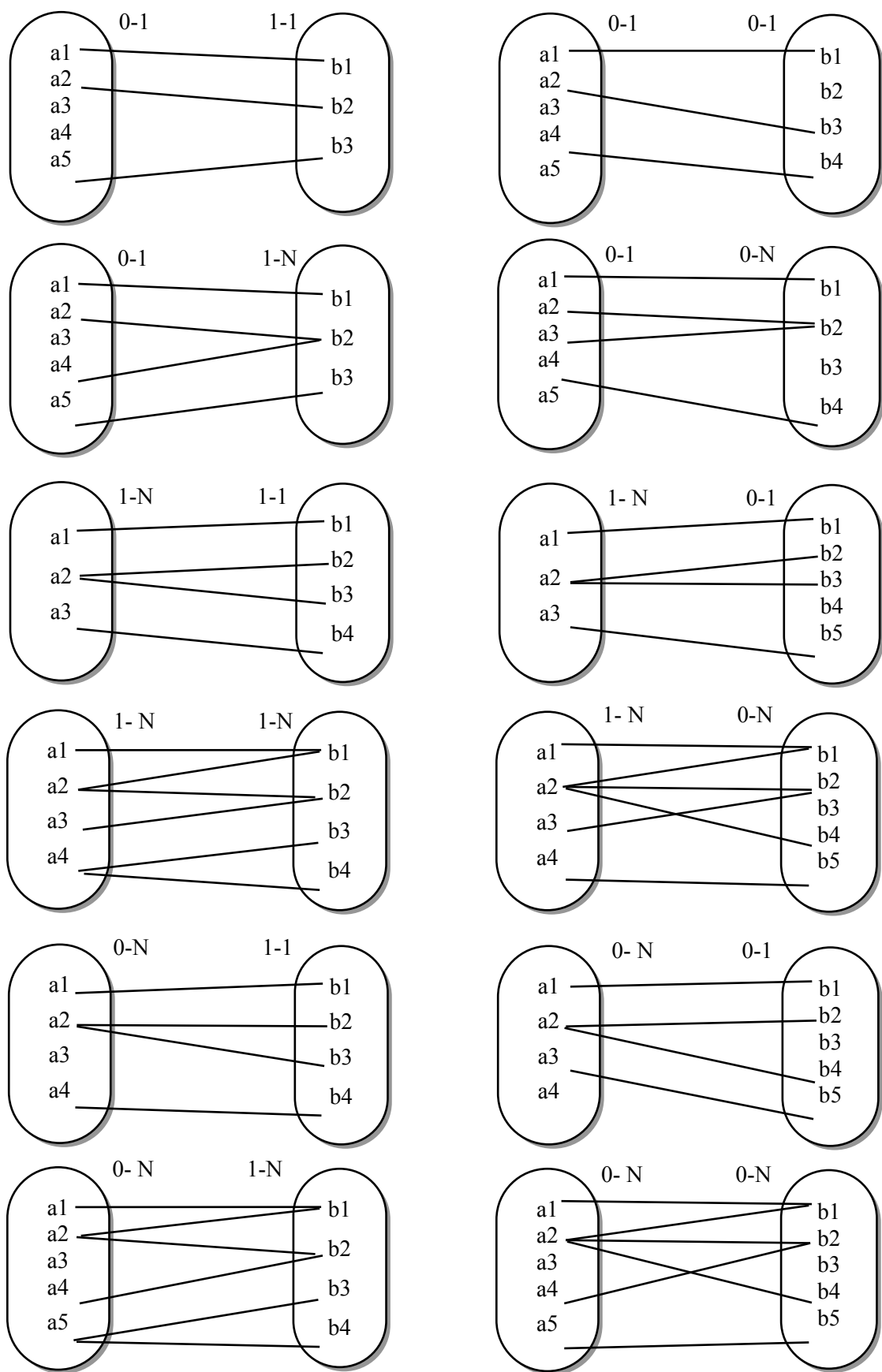
Một SINHVIÊN chỉ có một và chỉ một GIÁOVIÊN phụ trách, bản số (1 – 1).

Một GIÁOVIÊN phụ trách tối thiểu 4 sinh viên, hoặc hơn, bản số (4 – n).

### 1.1.3.16 khả năng của kiểu kết hợp nhị phân

Cho hai kiểu thực thể E1 (trái) và E2 (phải). Có tất cả 16 khả năng cho kiểu kết hợp nhị phân như sau :





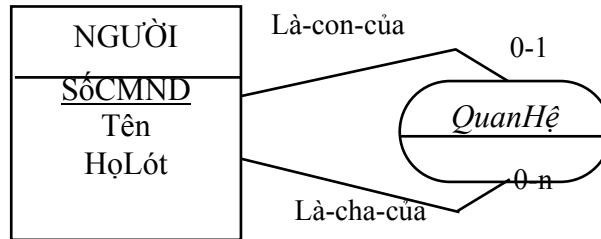
Hình 4.38 Có 16 khả năng của kiểu kết hợp nhị phân

### 1.1.4. Các kiểu kết hợp

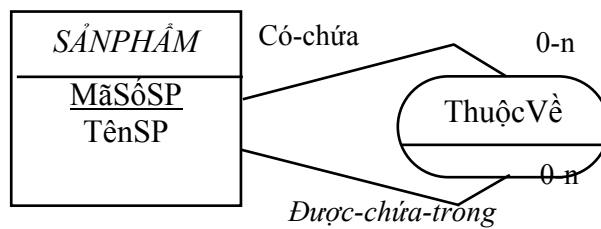
Tồn tại nhiều kiểu kết hợp, sau đây là những kiểu kết hợp hay gặp.

#### a) Kiểu kết hợp phản xạ

Kết hợp phản xạ thể xảy ra trên cùng một thực thể. Trong trường hợp này, tên kiểu kết hợp đóng vai trò quan trọng để phân biệt chiều ánh xạ.



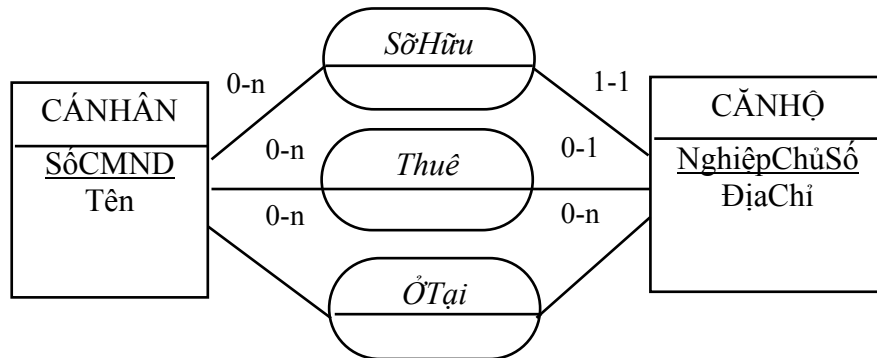
Hình 4.39 Quan hệ gia đình cha-con có kiểu phản xạ



Hình 4.40 Quan hệ danh mục sản phẩm có kiểu phản xạ

#### b) Nhiều kiểu kết hợp giữa cùng thực thể

Có thể có nhiều kiểu kết hợp giữa cùng thực thể. Chẳng hạn giữa hai kiểu thực thể CÁN NHÂN và CĂN HỘ có thể có các kiểu kết hợp *Sở Hữu*, *Thuê*, *Ở Tại* :

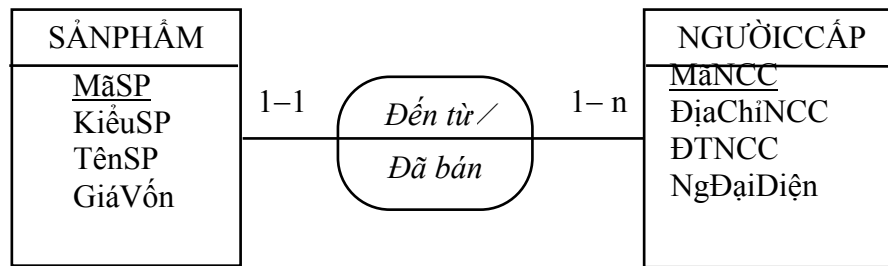


Hình 4.41 Nhiều kiểu kết hợp giữa cùng thực thể

#### c) Sự kết hợp có phân cấp (ràng buộc toàn vẹn hàm)

Một kiểu kết hợp phản ánh sự phụ thuộc về tác nghiệp giữa các khóa của những kiểu thực thể tương ứng được gọi là *kết hợp có phân cấp* hay *ràng buộc toàn vẹn hàm* (FIC : Functional Integrity Constraint). Kiểu kết hợp có phân cấp có thể xảy ra cho một thực thể duy nhất với kiểu kết hợp là phản xạ.

Ví dụ :



Hình 4.42 Kết hợp có phân cấp giữa hai kiểu thực thể

Sự kết hợp có phân cấp ở đây là một sản phẩm SẢN PHẨM do người cung cấp NGƯỜI CẤP bán (đến từ) và người cung cấp bán một sản phẩm (đã bán).

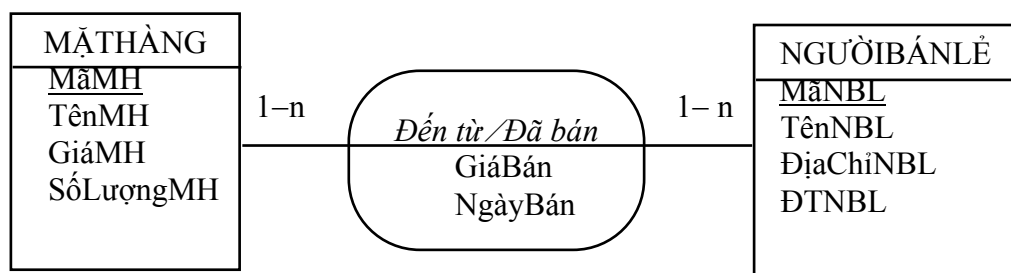
Các bản số giải thích một sản phẩm đến từ một và duy nhất một người cung cấp (1 – 1), trong khi đó một người cung cấp có thể bán một hoặc nhiều sản phẩm (1 – n).

Nói cách khác, với một giá trị của MãSP, tồn tại một và chỉ một giá trị của MãNCC, trong khi đó với một giá trị của MãNCC, tồn tại một hoặc nhiều giá trị của MãSP. Người ta cũng nói kết hợp có phân cấp là kết hợp *cha con*, *một-nhiều*.

#### d) Kết hợp không phân cấp (ràng buộc toàn vẹn bội)

Một kiểu kết hợp được gọi là *không phân cấp*, hay *ràng buộc toàn vẹn bội* (MIC : Multiple Integrity Constraint) nếu kiểu kết hợp đó có chứa dữ liệu (các thuộc tính của kiểu kết hợp) và các dữ liệu này chỉ được xác định khi xác định được tất cả các khoá của các thực thể liên quan.

Ví dụ :



Hình 4.43 Kết hợp không phân cấp giữa hai kiểu thực thể

Ở đây, một mặt hàng MẶT HÀNG do người bán lẻ NGƯỜI BÁN LẺ bán ra (đến từ) và người bán lẻ bán một mặt hàng (đã bán). Hai dữ liệu của kiểu kết hợp là GiáBán và NgàyBán phụ thuộc vào NGƯỜI BÁN LẺ và MẶT HÀNG. Có bao nhiêu người bán lẻ thì sẽ có bấy nhiêu giá bán cho mặt hàng này, bản số (1 – n), và, với một người bán lẻ, có bao nhiêu mặt hàng người bán lẻ này đã bán, thì sẽ có bấy nhiêu giá bán lẻ, bản số (1 – n).

Các kết hợp không phân cấp có thể không mang dữ liệu. Trong trường hợp này, sự phân cấp cho phép thiết lập sự tương ứng giữa các khóa của các thực thể liên quan. Trong ví dụ trên, nếu không có dữ liệu về giá bán và ngày bán, kiểu kết hợp bán cho biết những người bán lẻ đã bán một mặt hàng, nhưng cũng biết được những mặt hàng mà một người bán lẻ đã bán ra.

Người ta cũng nói kết hợp không phân cấp là kết hợp *nhiều-nhiều*

Chú ý :

Khi phân tích thiết kế HTTTT, người ta thường gặp một số dữ liệu được đặt riêng biệt, không thuộc vào bất kỳ một kiểu thực thể hoặc kiểu kết hợp nào. Đó là những tham số dữ liệu, những dữ liệu mà chỉ có một giá trị cho mọi thời điểm trong quá trình sống của CSDL.

Ví dụ :



Tỷ suất thuế đánh thêm trên giá trị (V.A.T - Value Added Tax)

Hệ số trượt giá

Các giá trị này có thể thay đổi, được cập nhật, nhưng tại mỗi thời điểm đã cho, chỉ có một giá trị duy nhất, xác định trong HTTTT đang xét.

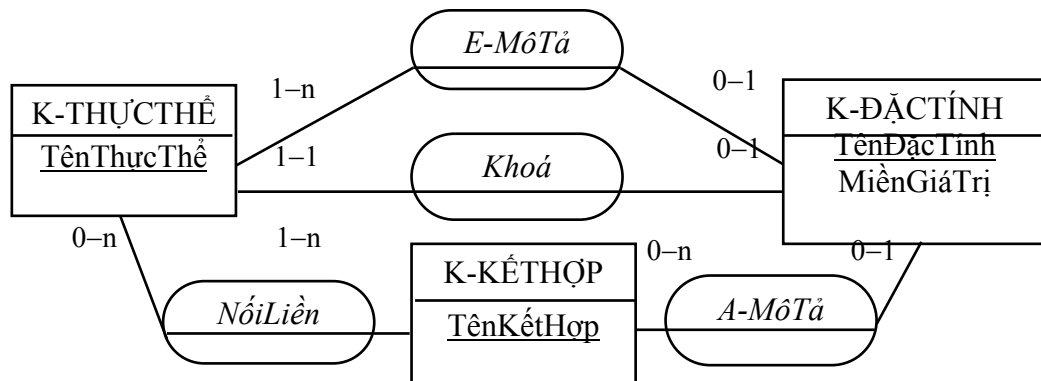
### 1.1.5. Các thành phần của từ điển dữ liệu

Từ điển dữ liệu của một MHYNDL có thể chứa các thành phần như sau :

Từ điển con (subdirectory) chứa các kiểu thực thể

Từ điển con chứa các kiểu kết hợp

Từ điển con chứa các kiểu đặc tính



Hình 4.44 Siêu sơ đồ của MHYN của CSDL

Từ điển dữ liệu lại cũng có thể được xem như một cơ sở dữ liệu gọi là *siêu sơ đồ* (meta-diagram). Siêu sơ đồ gồm ba kiểu thực thể : K-THỰC THỂ, K-KẾT HỢP và K-ĐẶC TÍNH được kết hợp với nhau bởi :

Kiểu kết hợp *E-MôTả* giữa K-THỰC THỂ và K-ĐẶC TÍNH

Kiểu kết hợp *A-MôTả* giữa K-KẾT HỢP và K-ĐẶC TÍNH

Kiểu kết hợp *Nói Liền* giữa K-KẾT HỢP và K-THỰC THỂ

Kiểu kết hợp *Định Danh* giữa K-THỰC THỂ và K-ĐẶC TÍNH

Sơ đồ ý niệm ở ví dụ Hình 4.41 có thể được biểu diễn theo siêu sơ đồ như sau :

Hai trường hợp cụ thể của K-THỰC THỂ có Tên Thực Thể lần lượt là CÁN NHÂN và CÁN HỘ.

Ba trường hợp cụ thể của K-KẾT HỢP có Tên Kết Hợp lần lượt là *Sở Hữu*, *Thuê* và *Ở Tại*.

Bốn trường hợp cụ thể của K-ĐẶC TÍNH có Tên Đặc Tính lần lượt là *Số CMND*, *Tên*, *Nghịệp Chủ Số* và *Địa Chỉ*.

## 1.2. Mô hình thực thể - kết hợp mở rộng

Các mô hình thực thể - kết hợp vừa được giới thiệu trên đây đã được giảng dạy trong các trường Đại học và đã được sử dụng rất rộng rãi trong thực tế. Tuy nhiên, từ những năm 1980, người ta đã thấy được những điểm yếu của các mô hình cổ điển này do không đáp ứng được cho những ứng dụng lớn có nhiều đối tượng phức tạp.

Người ta đã có nhiều phương pháp để nghiên cứu phát triển mô hình này, như xây dựng mô hình ngữ nghĩa của dữ liệu, các mô hình hướng đối tượng dựa trên các ngôn ngữ lập trình hướng đối tượng, v.v...

Dưới đây, ta sẽ giới thiệu các mô hình thực thể - kết hợp được mở rộng theo ba nội dung chính : kiểu đặc tính nhiều giá trị, kiểu đặc tính kết tụ và các kiểu con.

#### a) Kiểu đặc tính nhiều giá trị

Trong mô hình thực thể - kết hợp cổ điển, các kiểu đặc tính phải tuân thủ ràng buộc là dữ liệu sơ cấp. Do đó, trong một số trường hợp, khi mô hình hoá, người ta phải đưa vào một số kiểu thực thể bổ sung, nhưng ít có nghĩa. Ví dụ một cơ quan có nhiều số điện thoại thì phải xây dựng kiểu thực thể Điện thoại trong đó có đặc tính Số điện thoại.

Trong mô hình thực thể - kết hợp mở rộng, người ta đưa vào các kiểu đặc tính có thể có nhiều giá trị cho mỗi trường hợp cụ thể của kiểu thực thể.

Ví dụ :

Tuổi của các con của một nhân viên  
Các số điện thoại của một cơ quan

#### b) Kiểu đặc tính kết tụ

Trong mô hình thực thể - kết hợp cổ điển, không thể xây dựng một kiểu đặc tính được tổ hợp từ các kiểu đặc tính khác. Kiểu đặc tính kết tụ cho phép khắc phục khiếm khuyết này bằng cách cho phép kết một kiểu đặc tính từ các kiểu đặc tính khác.

Trong một trường hợp cụ thể của thực thể, giá trị của một kiểu đặc tính kết tụ là sự ghép (concatenation) các giá trị của các kiểu đặc tính sơ cấp. Ví dụ địa chỉ của một Khách hàng là sự ghép (kết tụ) của các kiểu đặc tính Số nhà, Tên Phố và Tên Tỉnh thành.

#### c) Các kiểu con

Khi cần sự khái quát (generalization) và sự đặc tả (specification), người ta đưa vào các kiểu con để nhận được từ một tập hợp các đối tượng các tập hợp con có các đặc tính chung của kiểu thực thể và các đặc tính đặc trưng cho kiểu con.

Một kiểu thực thể con (entity sub-type) B của một kiểu thực thể A là một tập hợp các thực thể thuộc về A, xác định bởi các kiểu đặc tính của A và bởi các kiểu đặc tính riêng. Người ta nói kiểu con thực thể B chuyên môn hoá (specialize) kiểu thực thể A và thừa kế (inherit) các kiểu đặc tính của A.

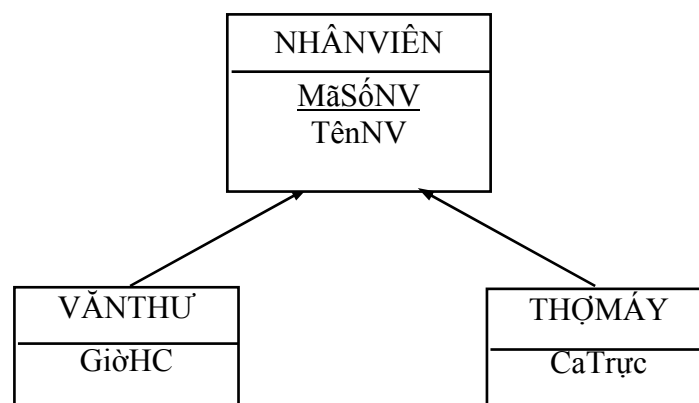
Ví dụ :

Kiểu thực thể NHÂNVIÊN được xác định bởi hai kiểu đặc tính MãSốNV và TênNV.

Các kiểu con thực thể của NHÂNVIÊN là VÃNTHƯ và THỢMÁY.

Kiểu con VÃNTHƯ có các kiểu đặc tính MãSốNV và TênNV, nhưng có thêm đặc tính đặc trưng cho kiểu con là GiờHC. Kiểu con THỢMÁY có thêm đặc tính CaTrực.

Để vẽ các kiểu con, người ta sử dụng phương pháp hướng đối tượng bằng cách dùng một mũi tên vẽ ngược từ kiểu con lên kiểu thực thể.



Hình 4.45 Mô hình của một kiểu thực thể có hai kiểu con

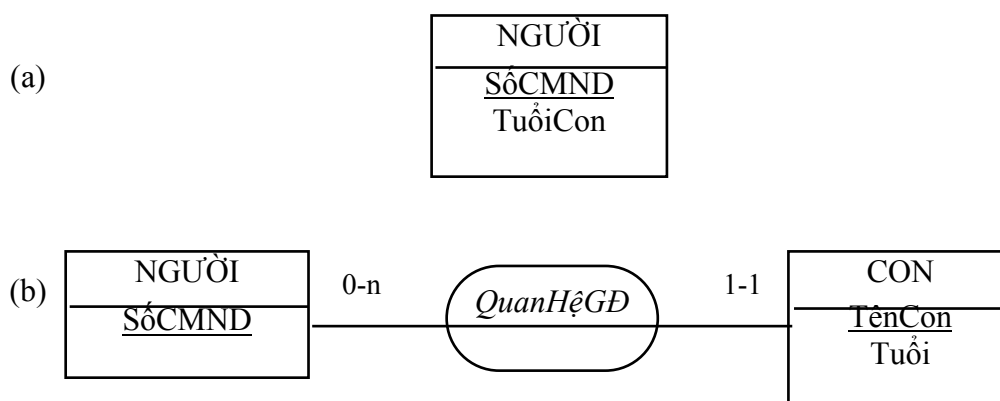
### I.3.Chuyển đổi các mô hình thực thể kết hợp

Do các hệ quan trị CSDL quan hệ hiện nay không cho phép sự thừa kế giữa các quan hệ nên cần phải chuyển đổi các mô hình thực thể kết hợp mở rộng về mô hình cổ điển. Sau đây là các quy tắc chuyển đổi.

Quy tắc 1 : Xử lý một kiểu thực thể có các kiểu đặc tính đa trị

Thay thế một kiểu đặc tính đa trị P của kiểu thực thể A đã cho bởi một kiểu thực thể mới E-P. Tạo ra trong E-P một đặc tính làm khoá, giả sử là I-E-P. Gán cho E-P một kiểu đặc tính sơ cấp p tương ứng với một giá trị của P. Thiết lập một kiểu kết hợp R giữa E-P và A. Sau đó tính bản số tương ứng của kết hợp R này.

Ví dụ, kiểu thực thể NGƯỜI trong mô hình (a) chứa đặc tính TuổiCon là đa trị, cần chuyển (a) thành mô hình (b).



Hình 4.46 Thay kiểu đặc tính đa trị thành một kiểu thực thể và một kiểu kết hợp

Quy tắc 2 : Xử lý một kiểu kết hợp có các đặc tính đa trị

Thay thế kiểu đặc tính đa trị P của kiểu kết hợp R đã cho bởi một kiểu thực thể E-R và hai kiểu kết hợp tương ứng. Sau đó áp dụng quy tắc 1 cho P trong ER.

Quy tắc 3 : Xử lý các đặc tính kết tụ

Thay thế mỗi kiểu đặc tính kết tụ P thành những kiểu đặc tính riêng mà những đặc tính này đã tham gia kết tụ ra P.

Quy tắc 4 : Xử lý các kiểu con thực thể

Khi thực thể A có kiểu con thực thể B, chọn một trong ba cách chuyển đổi sau :

Quy tắc 4.1 :

Bỏ B, thêm vào trong A các kiểu đặc tính riêng của B và tạo ra một kiểu đặc tính mới của A sao cho mỗi trường hợp cụ thể của A thì chỉ ra các kiểu đặc tính đặc trưng của kiểu con B. Đưa lên mức A tất cả các kết hợp ở mức B. Tính toán lại các bản số cho các kết hợp này.

Ví dụ :

Áp dụng quy tắc 4.1 trên đây để thay thế ba kiểu thực thể cho trong ví dụ ở Hình 4.45 thành một kiểu thực thể duy nhất như sau :



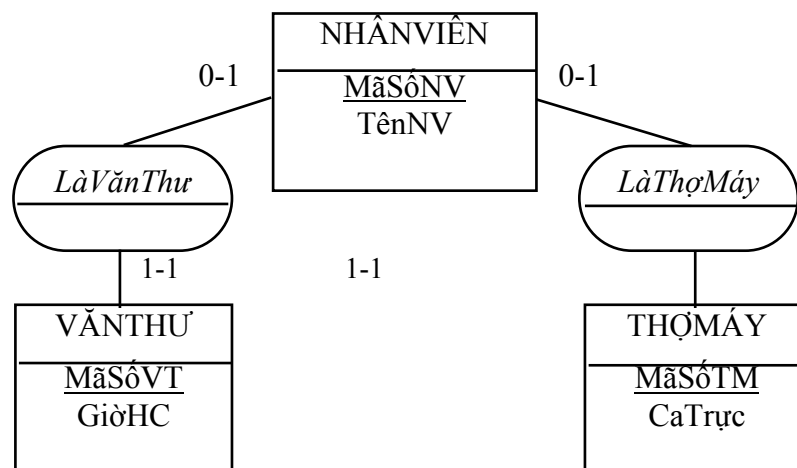
Hình 4.47 Thay thế một kiểu con thực thể thành một kiểu đặc tính

Quy tắc 4.2 :

Thay thế quan hệ thừa kế bởi một kết hợp giữa kiểu thực thể A và kiểu con B mà các bản số cực đại chỉ là 1, sau đó tính lại các bản số cực tiểu.

Ví dụ :

Áp dụng quy tắc 4.2 để thay thế mô hình kiểu con cho đã trong ví dụ ở Hình 4.45 thành mô hình các kiểu kết hợp như sau :



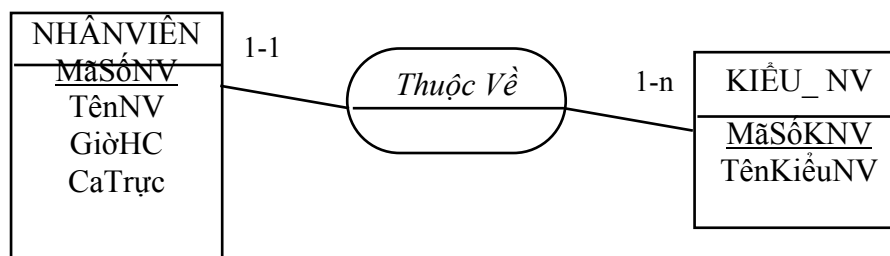
Hình 4.48 Thay thế quan hệ thừa kế bởi các kiểu kết hợp

Quy tắc 4.3 :

Giả sử kiểu thực thể A chứa một số kiểu con thực thể là B, C và D. Đưa vào trong A các kiểu đặc tính của các kiểu con B, C và D. Tạo ra một kiểu thực thể mới T-A cho phép biểu diễn các kiểu con cho mỗi trường hợp cụ thể của A. Thiết lập một kiểu kết hợp R giữa T-A và A. Tính các bản số tương ứng của R.

Ví dụ :

Ap dụng quy tắc 4.3 để thay thế mô hình kiểu con cho đã trong ví dụ ở Hình 4.45 thành một kiểu kết hợp như sau :



Hình 4.49 Thay thế các kiểu con bởi một kiểu đặc tính

## II. Mô hình quan hệ

Mô hình quan hệ (Relational Model) được Codd đề xuất vào năm 1970 dựa trên nền tảng Toán học lý thuyết về quan hệ để mô hình hoá dữ liệu. Mô hình quan hệ được nghiên cứu kỹ về mặt lý thuyết cũng như về mặt ứng dụng trong lĩnh vực CSDL để thiết kế các hệ quản trị CSDL quan hệ (RDBMS - Relational DataBase Management System). Hai mô hình khác cũng được nghiên cứu và ứng dụng trong lĩnh vực CSDL là mô hình phân cấp (Hierarchical Model) và mô hình mạng lưới (Network Model) chuẩn CODASYL và SOCRATE.

### II.1. Các định nghĩa

Cho  $n$  miền giá trị (hay tập hợp không có thứ tự) không nhất thiết rời nhau.

Một *quan hệ*  $R$  là một bộ phận (hay tập hợp con) tích Đềcac (Cartesian product)  $D_1 \times D_2 \times \dots \times D_n$ . Mỗi phần tử của  $R$  là một bộ- $n$  ( $n$ -tuple) có dạng  $\langle d_1 \times d_2 \times \dots \times d_n \rangle$  sao cho  $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$ .

Ví dụ :

Cho  $n=2$ ,  $D_1$  là tên nhân viên và  $D_2$  là tên tỉnh thành quê quán của nhân viên đó.

TÊN	TỈNHTHÀNH
Bảy	Hồ Chí Minh
Năm	Hà nội
Mười	Đà Nẵng

Tích Đềcac  $D_1 \times D_2$  sẽ tạo ra các cặp giá trị giữa hai miền giá trị  $D_1$  (TÊN) và  $D_2$  (TỈNHTHÀNH). Ta có các cặp giá trị :

(Bảy, Hồ Chí Minh), (Năm, Hà nội), (Mười, Đà Nẵng)

Một *lược đồ quan hệ* hay *sơ đồ quan hệ* (relational diagram) của một quan hệ  $R$  ( $D_1, D_2, \dots, D_n$ ) là một bộ- $n$  gồm các thành phần phân biệt, còn được gọi là các *đặc tính*, hay *thuộc tính* (Attributes)  $\langle A_1, A_2, \dots, A_n \rangle$ .  $R$  được gọi là tên của quan hệ.

Ví dụ :

Lược đồ quan hệ của ví dụ trên đây được viết NHÂNVIÊN (TÊN, TỈNHTHÀNH)

Các thành phần  $A_1, A_2, \dots, A_n$  trong lược đồ quan hệ mang tính ngữ nghĩa và cho phép chỉ định nhất quán cùng một đặc tính của đối tượng thì có thể xuất hiện trong nhiều lược đồ quan hệ khác nhau, không phụ thuộc vào vị trí của đặc tính đó trong tập hợp các thành phần của mỗi lược đồ.

Ví dụ :

Trong các lược đồ quan hệ sau đây :

ĐƠNĐHÀNG (ĐĐHSỐ, MÃKH, TÊNKH, MÃHG, SỐLGĐ, NGÀYĐH)

KHÁCHHÀNG (MÃKH, TÊNKH, ĐỊACHỈKH, ĐTHOẠIKH)

thì MÃKH và TÊNKH là hai đặc tính xuất hiện trong cả hai lược đồ quan hệ.

Một lược đồ quan hệ xác định *nội hàm* (intension) của một lớp quan hệ (hay tập hợp các quan hệ có cùng đặc tính) và được viết quy ước R ( $A_1, A_2, \dots, A_n$ ) với R là tên của lớp quan hệ đó và các  $A_i$  là tên đặc tính,  $i = 1..n$ .

Tại một thời điểm đã cho, tập hợp các bộ-n của một quan hệ thuộc lớp quan hệ đang xét được gọi là *ngoại diện* (extension) của lớp quan hệ đó. Ngoại diện xuất hiện như một bảng dữ liệu trong đó mỗi dòng tương ứng với một bộ-n và một cột tương ứng với một thành phần.

*Ví dụ :*

Quan hệ ĐƠNĐHÀNG (ĐĐHSỐ, MÃKH, TÊNKH, MÃHG, SỐLGĐ, NGÀYĐH)

xác định một lớp quan hệ là tập hợp các quan hệ thể hiện các đơn đặt hàng của khách hàng trong một doanh nghiệp. Ngoại diện của lớp quan hệ này như sau :

ĐĐHSỐ	MÃKH	TÊNKH	MÃHG	SỐLGĐH	NGÀYĐH
15	25	Đào	324Z	20	12/10/97
15	25	Đào	014Z	10	12/10/97
15	25	Đào	765Z	15	12/10/97
16	30	Mơ	014Z	30	09/11/97
16	30	Mơ	345Z	60	09/11/97
17	40	Mận	345Z	40	15/09/97
17	40	Mận	248Z	17	15/09/97
18	25	Đào	879Z	45	25/09/97

Từ bảng trên ta có :

$\langle 15, 25, \text{Đào}, 324Z, 20, 12/10/97 \rangle$  là một bộ-n của quan hệ ĐƠNĐHÀNG

(15, 16, 17, 18) là miền giá trị của thành phần ĐĐHSỐ

(Đào, Mận, Mơ) là miền giá trị của thành phần TÊNKH, v.v...

Người ta đưa vào các khái niệm :

*Bậc* (degree) của một quan hệ là số thành phần (hay số cột của bảng).

Trong ví dụ trên, bậc của ĐƠNĐHÀNG là 6.

*Bản số* (hay lực lượng) của một quan hệ là số bộ-n (hay số dòng của bảng).

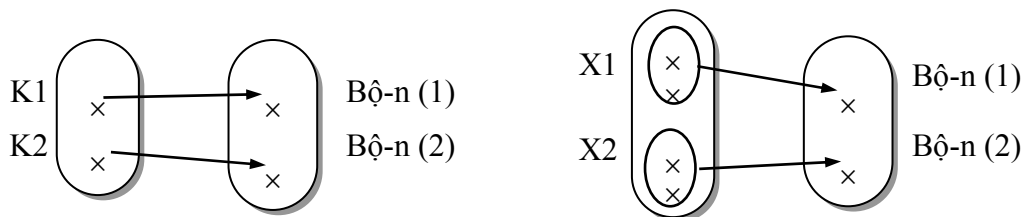
Trong ví dụ trên, bản số của ĐƠNĐHÀNG là 8.

#### a) *Khái niệm về khoá và siêu khoá*

Mọi lược đồ quan hệ đều có một *khóa* (key) định danh một cách duy nhất mỗi bộ-n trong mỗi ngoại diện.

Một tập hợp X các thành phần là một *siêu khoá* (super-key) nếu hai bộ-n phân biệt của R có các phép chiếu khác nhau trên X (hay có các giá trị khác nhau trên X).

X là một *khóa* nếu nó là một siêu khoá nhỏ nhất, nghĩa là không tồn tại tập hợp con  $Y \subset X$  mà Y là một siêu khoá.



Hình 4.50 Khoá và siêu khoá

*Ví dụ :*

TÊN trong quan hệ NHÂNVIÊN (TÊN, TỈNHTHÀNH) không phải là khoá đúng vì có thể có nhiều người cùng tên. Do đó cần đưa vào một đặc tính làm khoá là MÃNV để được quan hệ NHÂNVIÊN (MÃNV, TÊN, TỈNHTHÀNH).

Tuy nhiên, có thể lấy cặp MÃNV, TÊN làm khoá của quan hệ nhưng đây là một siêu khoá mà không phải là một khoá nhỏ nhất.

Quan hệ KHÁCHHÀNG (MÃKH, SỐCMND, ĐỊACHỈKH) có thể có hai khoá là MÃKH và SỐCMND, tuy nhiên chỉ nên chọn MÃKH làm khoá.

Quan hệ ĐƠNĐHÀNG (ĐĐHSỐ, MÃKH, TÊNKH, MÃHG, SỐLGĐ, NGÀYĐH) có khoá là cặp ĐĐHSỐ và MÃHG.

#### b) Trùng lặp các bộ-n

Về mặt Toán học, các bộ-n phải hoàn toàn phân biệt nhau, nghĩa là không tồn tại các bộ-n giống hệt nhau. Tuy nhiên, việc kiểm tra trùng lặp trong thực tiễn là rất khó khăn vì độ lớn của các CSDL. Do đó nên chọn nhiều khoá để tránh sự trùng lặp.

#### c) Ràng buộc toàn vẹn

Người ta đưa vào khái niệm *ràng buộc toàn vẹn* (integrity constraint) gồm hai dạng như sau :

Sự độc lập giữa các đặc tính.

Các quy tắc về tính chắc chắn (consistency) và tính đầy đủ (completeness) của các lớp quan hệ.

Các ràng buộc toàn vẹn dùng để kiểm tra mọi ngoại diện của một lược đồ quan hệ xác định trên các miền dữ liệu  $D_1, D_2, \dots, D_n$ .

#### d) Thứ tự viết

Theo định nghĩa Toán học của quan hệ, không tồn tại thứ tự ưu tiên giữa các đặc tính cũng như giữa các bộ-n. Tuy nhiên để dễ theo dõi, thường người ta viết theo thứ tự như sau :

$R(A_1, \underline{A_2, \dots, A_p}, A_{p+1}, \dots, A_n, B_1, \dots, B_k)$

$A_1, A_2, \dots, A_p$  là những đặc tính dùng làm khoá của R

(thông thường được tạo thành từ nhiều đặc tính của các quan hệ khác)

$A_{p+1}, \dots, A_n$  là những đặc tính đơn của R

$B_1, \dots, B_k$  là các khoá ngoại (import keys) đến từ các quan hệ khác

*Ví dụ :*

Trong quan hệ ĐƠNĐHÀNG (ĐĐHSỐ, MÃHG, SỐLGĐ, NGÀYĐH, MÃKH) thì :

ĐĐHSỐ (số của đơn đặt hàng) là một phần của khoá,

MÃHG (mã hàng) là khoá của quan hệ HÀNG,

SỐLGĐ và NGÀYĐH là những đặc tính đơn.

MÃKH là khoá ngoại có trong quan hệ KHÁCHHÀNG (cho biết khách hàng nào đã đặt hàng).

e) *Sơ đồ quan hệ của một CSDL*

Một lược đồ quan hệ của một CSDL được tạo thành từ hai phần tử :

Các lược đồ quan hệ (hay sơ đồ ý niệm) đặc trưng cho CSDL tu cho CSDL tương ứng với định nghĩa nội hàm của các quan hệ.

Tập hợp các ràng buộc toàn vẹn tham gia vào các quan hệ.

Các ràng buộc toàn vẹn đối với các khoá của các quan hệ vừa trình bày ở trên đưa đến nguyên tắc : *hai bộ-n phân biệt phải có một phép chiếu khác nhau trên các khoá, hay một giá trị khoá chỉ tương ứng với một và chỉ một bộ-n*. Từ nguyên tắc này, người ta đưa vào khái niệm phụ thuộc hàm và các dạng chuẩn của lược đồ quan hệ.

## II.2. Phụ thuộc hàm

### II.2.1. Khái niệm

*Phụ thuộc hàm* (Functional Dependency), viết tắt là PTH, do Codd đề xuất để phân tích đặc trưng của các quan hệ mà không làm mất mát thông tin.

Giả sử cho quan hệ R ( $A_1, A_2, \dots, A_n$ ) và hai tập hợp con X và Y gồm các thành phần là các đặc tính. Người ta nói rằng Y là PTH của X hay tồn tại một PTH giữa X và Y nếu và chỉ nếu, khi hai bộ-n của R có cùng một phép chiếu lên X, thì chúng sẽ có cùng một phép chiếu như vậy lên Y. PTH giữa X và Y được viết :

$$X \rightarrow Y$$

Nói cách khác, với mọi ngoại diện r của R, mọi bộ  $t_1, t_2$  của R, ta có :

$$P_X(t_1) = P_X(t_2) \Rightarrow P_Y(t_1) = P_Y(t_2)$$

trong đó P là ký hiệu phép chiếu trên một hoặc nhiều thành phần của quan hệ R.

Một cách đơn giản, trong một sơ đồ R, người ta nói rằng Y là PTH vào X, hay X là PTH với Y, với mọi ngoại diện của R. Như vậy, PTH đặt mối liên hệ giữa hai dữ liệu, một dữ liệu là *nguồn* và dữ liệu kia là *đích* :

$$\text{Nguồn} \rightarrow \text{Đích}$$

sao cho một giá trị dữ liệu nguồn chỉ có thể tương ứng với một giá trị đích.

Quan niệm về PTH rất cơ bản trong việc phân tích cấu trúc dữ liệu. Nghiên cứu PTH là bước đầu tiên để xây dựng một mô hình dữ liệu, nghĩa là để đưa ra một sự biểu diễn hình thức của cấu trúc dữ liệu này. Việc xác định các PTH chỉ có được sau khi đã phân tích HTTT.

*Ví dụ :*

Cho quan hệ KHÁCH HÀNG (MÃKH, TÊNKH, ĐỊACHỈKH), ta có thể xây dựng các PTH như sau :

$$\text{MÃKH} \rightarrow \text{TÊNKH}$$

$$\text{MÃKH} \rightarrow \text{ĐỊACHỈKH}$$

Quan hệ ĐƠN HÀNG (ĐĐHSỐ, MÃHG, SỐLGĐ, NGÀYĐH, MÃKH, TÊNKH) có các PTH như sau :

$$\text{ĐĐHSỐ} \rightarrow \text{MÃKH}$$

$$\text{ĐĐHSỐ} \rightarrow \text{TÊNKH}$$

$$\text{ĐĐHSỐ} \rightarrow \text{NGÀYĐH}$$

$$\text{ĐĐHSỐ, MÃHG} \rightarrow \text{SỐLGĐ}$$



Ba PTH trên cùng được giải thích như sau : một đơn đặt hàng có số (number) là ĐĐHSỐ do một khách hàng có mã số MÃKH và tên TÊNKH nào đó, đặt hàng trong một ngày NGÀYĐH nào đó.

PTH cuối cùng được giải thích như sau : với một đơn đặt hàng và một mặt hàng cố định nào đó, khách hàng đã đặt một số lượng đã cho. PTH này còn được viết :

$$\text{ĐĐHSỐ} + \text{MÃHG} \rightarrow \text{SÓLGĐ}$$

Phép + trong nguồn là phép ghép (concatenation) của các đặc tính.

Đề ý rằng còn một PTH khác là MÃKH  $\rightarrow$  TÊNKH cho biết một mã số khách hàng chỉ tương ứng với một tên khách hàng và ngược lại.

Chú ý rằng PTH giữa X và Y trong một quan hệ R là một ràng buộc toàn vẹn tham gia vào lược đồ quan hệ R.

### II.2.2. Các tính chất của phụ thuộc hàm

Cho quan hệ R ( $A_1, A_2, \dots, A_n$ ) và một số PTH giữa các đặc tính  $A_1, A_2, \dots, A_n$ , ta có các tính chất sau đây (được minh họa qua các ví dụ) :

#### a) Phản xạ (reflexivity)

Ta luôn có  $A_i \rightarrow A_i$ , hay nếu  $X \subseteq A_i$  thì  $X \rightarrow A_i$

Tính chất phản xạ được hiểu là mọi đặc tính được xác định bởi chính chúng hoặc một phần của chúng.

#### b) Tính chiếu (projection)

Nếu  $A_i \rightarrow A_j + A_k$  thì  $A_i \rightarrow A_j$  và  $A_i \rightarrow A_k$

$\text{MÃKH} \rightarrow \text{TÊNKH} + \text{ĐỊACHỈKH} \Rightarrow \text{MÃKH} \rightarrow \text{TÊNKH}$  và  $\text{MÃKH} \rightarrow \text{ĐỊACHỈKH}$

Nếu một đặc tính là PTH với một tập hợp các đặc tính thì nó cũng PTH với mỗi đặc tính (phần tử) của tập hợp đó.

#### c) Tăng thêm (increase)

Nếu  $A_i \rightarrow A_j$  thì  $\forall A_k, A_i + A_k \rightarrow A_j$

$\text{ĐĐHSỐ} \rightarrow \text{NGÀYĐH} \Rightarrow \text{ĐĐHSỐ} + \text{MÃHG} \rightarrow \text{NGÀYĐH}$

Có nghĩa là PTH sẽ không thay đổi nếu thêm một tập hợp các đặc tính vào nguồn (vế bên trái) của PTH.

#### d) Tính cộng được (additivity)

Nếu  $A_i \rightarrow A_j$  và  $A_i \rightarrow A_k$  thì  $A_i \rightarrow A_j + A_k$

$\text{MÃKH} \rightarrow \text{TÊNKH}$  và  $\text{MÃKH} \rightarrow \text{ĐỊACHỈKH} \Rightarrow \text{MÃKH} \rightarrow \text{TÊNKH} + \text{ĐỊACHỈKH}$

Đây là tính chất ngược lại với tính chất xạ ảnh. Nghĩa là PTH sẽ không thay đổi nếu thêm vào đích (vế bên phải) của PTH một tập hợp các đặc tính với điều kiện là tồn tại PTH giữa nguồn (vế bên trái) với mỗi đặc tính của đích.

#### e) Tính bắc cầu hay truyền ứng (transitivity)

Nếu  $A_i \rightarrow A_j$  và  $A_j \rightarrow A_k$ , thì  $A_i \rightarrow A_k$

$\text{ĐĐHSỐ} \rightarrow \text{MÃKH}$  và  $\text{MÃKH} \rightarrow \text{TÊNKH} \Rightarrow \text{ĐĐHSỐ} \rightarrow \text{TÊNKH}$

#### f) Tính giả bắc cầu (pseudo-transitivity)

Nếu  $A_i \rightarrow A_j$  và  $A_j + A_k \rightarrow A_l$ , thì  $A_i + A_k \rightarrow A_l$

$$\begin{array}{cccc} \text{ĐĐHSỐ} & \rightarrow & \text{MÃKH} & \text{và} & \text{MÃKH} & \rightarrow & \text{TÊNKH} \\ (A_i) & & (A_j) & & (A_j) & & (A_l) \end{array}$$

$$\text{Nhờ tính chất cộng thêm, ta có : } \begin{array}{ccc} \text{MÃKH} & + & \text{MÃHG} \\ (A_j) & & (A_k)(A_l) \end{array} \rightarrow \text{TÊNKH}$$

$$\text{Từ đó ta có : } \begin{array}{ccc} \text{ĐĐHSỐ} & + & \text{MÃHG} \\ (A_i) & & (A_k) \end{array} \rightarrow \text{TÊNKH} \quad (A_l)$$

Các tính chất trên đây có ích gì và được ứng dụng như thế nào ? Người ta khuyên không nên liệt kê ra hết mọi PTH mà trong đó, một số suy ra được từ các PTH đã có mặt trong CSDL. Các tính chất của PTH cho phép hạn chế tính dư thừa quan hệ.

### II.2.3. Các loại hình của phụ thuộc hàm

Cho A là tập hợp các đặc tính  $A_1, A_2, \dots, A_n$ . Giả sử  $R(A)$  là một quan hệ, X và Y là hai tập hợp con của A sao cho tồn tại một PTH  $X \rightarrow Y$ . Người ta đưa vào các loại hình (typography) PTH như sau :

Phụ thuộc hàm *chấp nhận được* (canonical FD) hay đúng quy tắc :

$X \rightarrow Y$  là chấp nhận được nếu Y là một đặc tính duy nhất của A.

Phụ thuộc hàm *tầm thường* (trivial FD) :

$X \rightarrow Y$  là tầm thường nếu  $Y \subset X$ .

Phụ thuộc hàm *sơ cấp* (elementary FD) :

$X \rightarrow Y$  là sơ cấp nếu không tồn tại  $X' \subset X$  sao cho tồn tại  $X' \rightarrow Y$

Nói cách khác  $X \rightarrow Y$  là sơ cấp nếu dữ liệu nguồn X là sơ cấp, không thể phân chia được nữa.

Ví dụ trong quan hệ :

$\text{ĐƠNĐHÀNG} (\text{ĐĐHSỐ}, \text{MÃHG}, \text{SỐLGĐ}, \text{NGÀYĐH}, \text{MÃKH}, \text{TÊNKH})$

thì  $\text{ĐĐHSỐ} + \text{MÃHG} \rightarrow \text{MÃKH}$  không là PTH sơ cấp vì  $\text{ĐĐHSỐ} \rightarrow \text{MÃKH}$ .

PTH trực tiếp (direct FD) :

$X \rightarrow Y$  là *trực tiếp* nếu không tồn tại dữ liệu Z nào đó,  $Z \subset A$ , sao cho tồn tại PTH  $X \rightarrow Z, Z \rightarrow Y$  mà  $Z \rightarrow X$  không tồn tại trong R.

Trong ví dụ đơn đặt hàng,  $\text{ĐĐHSỐ} \rightarrow \text{TÊNKH}$  không là trực tiếp vì tồn tại dữ liệu  $\text{MÃKH}$  sao cho  $\text{ĐĐHSỐ} \rightarrow \text{MÃKH}$  và  $\text{MÃKH} \rightarrow \text{TÊNKH}$ .

Việc chỉ ra loại hình của PTH có ích ở chỗ các PTH không sơ cấp hoặc không trực tiếp sẽ tương ứng với sự dư thừa dữ liệu trong một quan hệ cần phải loại bỏ.

### II.2.4. Đồ thị của các phụ thuộc hàm

Cho R là một lược đồ quan hệ, người ta nói *đồ thị* (graph) của các PTH của R là một đồ thị mà các nút của nó là các đặc tính đơn hay có thể là các đặc tính ghép của R, các cung là các PTH của R tồn tại giữa các đặc tính đó.

Để vẽ đồ thị PTH của lược đồ một đồ quan hệ, xây dựng tập hợp F là các PTH sơ cấp nhận được từ lược đồ quan hệ đã cho. Từ tập hợp F, vẽ đồ thị PTH như sau :

Đầu tiên chọn đặt các nút là các khoá đơn

(khóa được tạo thành từ chỉ một đặc tính duy nhất)

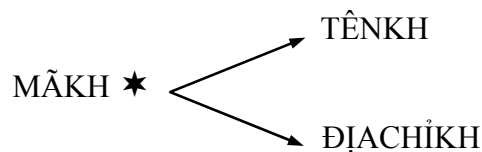
Nối lần lượt mỗi nút khoá với các nút là các đặc tính nếu tồn tại PTH xác định giữa chúng

*Ví dụ :*

Từ quan hệ KHÁCHHÀNG (MÃKH, TÊNKH, ĐỊACHỈKH), ta có thể xây dựng tập hợp F như sau :

$F = \{ \text{MÃKH} \rightarrow \text{TÊNKH}, \text{MÃKH} \rightarrow \text{ĐỊACHỈKH} \}$

Đồ thị PTH được tạo ra một cách đơn giản như sau :



Hình 4.51 Đồ thị PTH đơn giản

Đối với các lược đồ quan hệ phức tạp hơn, rõ ràng đồ thị PTH sẽ phức tạp hơn, do có thể có các khoá ghép.

Ví dụ : Cho các quan hệ :

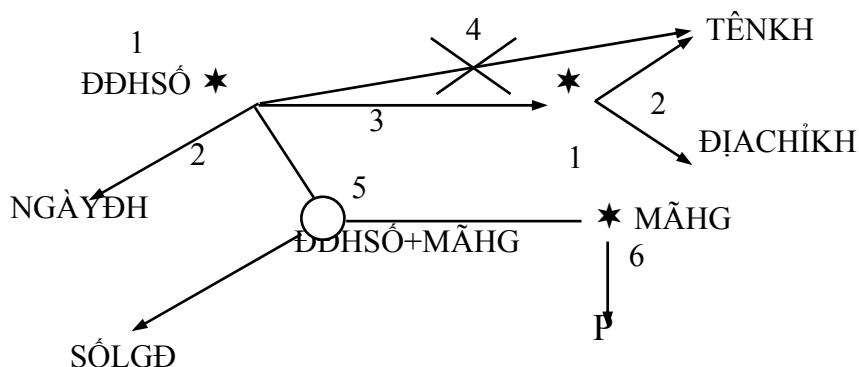
KHÁCHHÀNG (MÃKH, TÊNKH, ĐỊACHỈKH),

ĐƠNĐHÀNG (ĐĐHSỐ, MÃHG, SỐLGD, NGÀYĐH, MÃKH, TÊNKH),

Quan hệ ĐƠNĐHÀNG có khoá ghép là (ĐĐHSỐ, MÃHG). Ta có tập hợp F các PTH như sau :

$F = \{ \text{MÃKH} \rightarrow \text{TÊNKH}, \text{MÃKH} \rightarrow \text{ĐỊACHỈKH};$   
 $\text{ĐĐHSỐ} \rightarrow \text{MÃKH}, \text{ĐĐHSỐ} \rightarrow \text{NGÀYĐH}, \text{ĐĐHSỐ} \rightarrow \text{TÊNKH},$   
 $\text{ĐĐHSỐ} + \text{MÃHG} \rightarrow \text{SÓLGD} \}$

Từ F, xây dựng đồ thị PTH như hình dưới đây :



Hình 4.52 Đồ thị PTH đầy đủ

Chú ý rằng đồ thị PTH trên đây có thể bỏ PTH  $\text{ĐĐHSỐ} \rightarrow \text{TÊNKH}$  vì  $\text{TÊNKH}$  nhận được từ  $\text{ĐĐHSỐ}$  bởi các PTH bắc cầu  $\text{ĐĐHSỐ} \rightarrow \text{MÃKH}$ ,  $\text{MÃKH} \rightarrow \text{TÊNKH}$ .

Trong đồ thị PTH có xuất hiện một quan hệ mới là HÀNG với khoá MÃHG và các thành phần được biểu diễn bởi P.

### II.3. Các dạng chuẩn của lược đồ quan hệ

Cho  $R(A_1, A_2, \dots, A_n)$  là một lược đồ quan hệ và  $K$  là khoá của  $R$ . Người ta định nghĩa các dạng chuẩn như sau :

#### a) Dạng chuẩn 1

Quan hệ  $R$  được gọi là ở dạng chuẩn 1 (1NF - first normal form) nếu và chỉ nếu mỗi thành phần  $A_i$  của  $R$  không có mặt trong khoá  $K$  là PTH của  $K$ . Nói cách khác, ứng mỗi giá trị của khoá  $K$ , không tồn tại một thành phần khác  $K$  là đa trị.

Ví dụ :

Quan hệ KHÁCH HÀNG (MÃKH, TÊNKH, ĐỊACHỈKH) là 1 NF.

Quan hệ ĐƠNĐHÀNG (ĐĐHSỐ, MÃHG, SÓLGĐ, NGÀYĐH, MÃKH, TÊNKH) sẽ là 1NF nếu (ĐĐHSỐ, MÃHG) là khoá (ghép) của nó.

Tuy nhiên nếu chọn ĐĐHSỐ là khoá thì quan hệ ĐƠNĐHÀNG sẽ không còn ở dạng chuẩn 1 vì với mỗi giá trị của ĐĐHSỐ, tồn tại nhiều giá trị của MÃHG (là tập hợp các mặt hàng ứng với đơn đặt hàng do khách hàng yêu cầu).

#### b) Dạng chuẩn 2

Quan hệ  $R$  được gọi là ở dạng chuẩn 2 (2NF) nếu và chỉ nếu mỗi thành phần  $A_i$  của  $R$  không có mặt trong khoá  $K$  là PTH sơ cấp của  $K$  (1NF + tính sơ cấp). Nói cách khác, không tồn tại một PTH giữa một thành phần của  $K$  và một thành phần  $A_i$ .

Như vậy, các quan hệ có khoá ghép sẽ không ở dạng chuẩn 2.

Ví dụ :

Quan hệ KHÁCH HÀNG (MÃKH, TÊNKH, ĐỊACHỈKH) là 2 NF.

Quan hệ ĐƠNĐHÀNG (ĐĐHSỐ, MÃHG, SÓLGĐ, NGÀYĐH, MÃKH, TÊNKH)

không là 2NF vì ta có PTH ĐĐHSỐ  $\rightarrow$  (MÃKH, TÊNKH, NGÀYĐH), nghĩa là một thành phần của khoá ghép có PTH với một tập hợp các thành phần của quan hệ.

Một lý do khác là có sự mập mờ giữa đơn đặt hàng nói riêng và nội dung đặt hàng của khách hàng trong quan hệ ĐƠNĐHÀNG. Để tránh sự không rõ ràng này, người ta tách quan hệ ĐƠNĐHÀNG thành hai quan hệ con (sub-relation) :

ĐƠNĐHÀNG (ĐĐHSỐ, NGÀYĐH, MÃKH, TÊNKH)

NỘI DUNGĐH (ĐĐHSỐ, MÃHG, SÓLGĐ)

Lúc này, các hai quan hệ đều ở dạng chuẩn 2NF.

#### c) Dạng chuẩn 3

Quan hệ  $R$  được gọi là ở dạng chuẩn 3 (3NF) nếu và chỉ nếu mỗi thành phần  $A_i$  của  $R$  không có mặt trong khoá  $K$  là PTH sơ cấp trực tiếp của  $K$  (2NF + tính không bắc cầu). Nói cách khác, không tồn tại PTH giữa các thành phần còn lại của quan hệ.

Ví dụ :

Các quan hệ KHÁCH HÀNG và NỘI DUNGĐH là 3NF. Quan hệ ĐƠNĐHÀNG không là 3NF vì ta có các PTH :

ĐĐHSỐ  $\rightarrow$  MÃKH, ĐĐHSỐ  $\rightarrow$  TÊNKH, nhưng MÃKH  $\rightarrow$  TÊNKH

Nghĩa là không tồn tại PTH trực tiếp giữa ĐĐHSỐ và TÊNKH, hay tồn tại PTH giữa các thành phần MÃKH và TÊNKH.

Tuy nhiên người ta không tiếp tục thu nhỏ quan hệ như trước. Từ quan hệ KHÁCH HÀNG, có thể nhận được tên khách hàng từ mã số của khách hàng.

#### d) Các dạng chuẩn khác

Từ dạng chuẩn 3, người ta định nghĩa dạng chuẩn 3BCKNF (BCK - Boyce Codd Kent), dạng chuẩn 4 và dạng chuẩn 5. Tuy nhiên trong khuôn khổ giáo trình này, chúng ta dừng lại ở dạng chuẩn 3 với dạng chuẩn 1 là bắt buộc.

## II.4. Ví dụ khu du lịch Non Nước

### II.4.1. Giới thiệu cơ sở dữ liệu

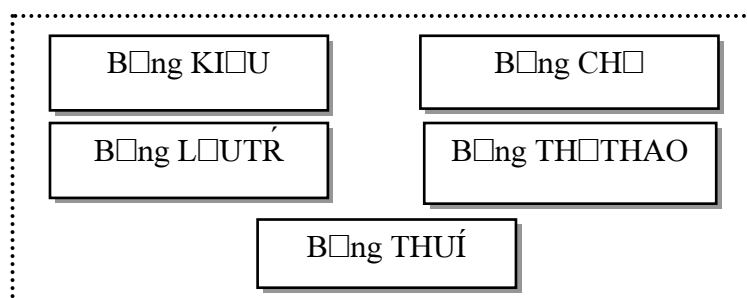
Để quản lý khu du lịch Non Nước, ở đây chỉ xét việc quản lý các lưu trú, báo gồm việc thuê chỗ nghỉ, địa điểm cắm trại và thuê phương tiện thể thao. Để lưu giữ thông tin về lưu trú của một nhóm khách, ban quản lý sẽ lập một hóa đơn thanh toán vào cuối kỳ. Mỗi nhóm khách được chỉ định bởi một số đăng ký lưu trú và tên của một người khách trong nhóm.

#### a) Định nghĩa bảng

Mỗi bảng có một tên gồm nhiều dòng và cột. Mỗi cột mang một tên dữ liệu, nằm ở đầu cột. Mỗi dòng chứa tập hợp các giá trị của các dữ liệu, mỗi giá trị cho một cột (giao giữa hàng và cột). Như vậy, bảng có bao nhiêu cột mang tên dữ liệu thì có bấy nhiêu giá trị. Thường người ta thêm một cột phụ bên trái cùng để ghi thứ tự các dòng dữ liệu của bảng.

#### b) Các bảng của cơ sở dữ liệu ví dụ

Cơ sở dữ liệu gồm 5 bảng như sau :



Bảng KIỂU gồm 2 cột : KiểuChỗ và GiáNgàyNgười.

Bảng KIỂU		
	KiểuChỗ	GiáNgàyNgười
1	Lều bạt	70 000
2	Xe nhà ở	82 500
3	Xe cắm trại	95 000
4	Nhà lán	128 500

Bảng CHỖ gồm 4 cột :

ChỗSố      số thứ tự của chỗ,  
DiệnTích      diện tích của chỗ,  
SốNgMax      số khách tối đa có thể ở,  
KiểuChỗ      kiểu chỗ.

Bảng CHỖ				
	<u>ChỗSố</u>	DiệnTích	SốNgMax	KiểuChỗ
1	1	10	4	Lều bạt
2	2	12	4	Lều bạt
—	— —	— —	— —	— — —
15	15	14	4	Xe nhà ở
16	16	19	6	Xe nhà ở
—	— —	— —	— —	— — —
33	33	16	4	Xe cắm trại
34	34	16	4	Xe cắm trại
—	— —	— —	— —	— — —
58	58	30	4	Nhà lán
59	59	34	5	Nhà lán
—	— —	— —	— —	— — —

Bảng LUÛTRÚ gồm 6 cột :

LưuTrúSố      mỗi nhóm khách có một số đăng ký lưu trú.  
TênKhách      Tên khách chịu trách nhiệm trong nhóm.  
NgàyĐến      Ngày đến lưu trú.  
NgàyĐi      Ngày đi khỏi (ngày lưu trú cuối cùng).  
SốNgười      Số người lưu trú.  
ChỗSố      Chỉ số thứ tự chỗ đã lưu trú.

Bảng LUÛTRÚ						
	<u>LưuTrúSố</u>	TênKhách	NgàyĐến	NgàyĐi	SốNgười	ChỗSố
1	100	Quỳnh	02/07/96	09/07/96	3	4
2	101	Liên	02/07/96	09/07/96	5	12
3	102	Thái	03/07/96	05/07/96	4	3
4	103	Dũng	05/07/96	014/07/96	5	16
5	104	Hiền	05/07/96	010/07/96	2	31

Bảng THỀTHAO gồm 3 cột :

TênThềThao      Quần vợt, đi xe đạp trên mọi địa hình, chèo thuyền, lướt ván...  
ĐơnVịTính      Thời gian thuê tính theo đơn vị giờ, 1/2 ngày, ngày...  
GiáTiền      Giá tiền theo đơn vị thời gian.

Bảng THỀTHAO			
	<u>TênThềThao</u>	ĐơnVịTính	GiáTiền
1	Quần vợt	1 giờ	50 000
2	Đi xe đạp	1 ngày	55 000
3	Lướt ván	1/2 ngày	65 000
4	Chèo thuyền	1 giờ	25 000
5	Bóng bàn	1 giờ	15 000

Bảng THUÊ gồm 4 cột :

Bảng THUÊ				
	<u>LưuTrúSố</u>	<u>TênThểThao</u>	<u>NgàyThuê</u>	<u>SốĐơnVị</u>
1	102	Quần vợt	06/07/96	1
2	101	Chèo thuyền	07/07/96	2
3	104	Bóng bàn	06/07/96	2
4	103	Lướt ván	06/07/96	1
5	107	Đi xe đạp	08/07/96	1

#### **II.4.2. Quan sát dữ liệu**

Sau khi lập các bảng dữ liệu, ta quan sát theo các giai đoạn như sau :

Lúc đầu, ta thấy các bảng độc lập với nhau, không có mối liên hệ nào giữa chúng.

Tiếp theo, ta chú ý đến toàn bộ các bảng, xem xét những mối liên hệ giữa chúng để tạo thành cơ sở dữ liệu.

##### **a) Quan sát các bảng**

Với mỗi bảng, chọn một, hoặc nhiều dữ liệu, có chức năng khóa (gạch chân trong các bảng).

Ví dụ 1 : Trong bảng KIỂU, để xác định giá thuê một ngày cho một người theo kiểu chỗ cho thuê, cần phải biết kiểu chỗ liên quan. Như vậy giá trị của KiểuChỗ là khóa dữ liệu.

Ví dụ 2 : Trong bảng CHỖ, để xác định diện tích và số người tối đa có thể thuê chỗ, cần biết vị trí (chỗ số mấy). Như vậy, giá trị của ChỗSố là khóa.

Như vậy :

– Khóa là một dữ liệu đặc biệt của bảng (hoặc của một tập hợp dữ liệu) mà khi cho nó giá trị thì có thể truy cập đến các giá trị dữ liệu tương ứng khác của bảng (trên cùng một dòng của bảng).

Các giá trị của khóa là duy nhất. Ví dụ : Chỉ có một giá trị của dữ liệu KiểuChỗ trong bảng CHỖ. Trong khi đó, các giá trị của dữ liệu không phải khóa thì có thể có nhiều dòng có cùng giá trị là số người, vì mỗi dòng ứng với một nhóm, mà các nhóm này có thể có số người giống nhau.

– Chú ý rằng bảng THUÊ có 3 khóa là LưuTrúSố, TênThểThao và NgàyThuê (các bảng khác chỉ có 1). Có nghĩa rằng, để định danh một dòng một cách chắc chắn, cần cung cấp cho mỗi khóa một giá trị dữ liệu. Bởi vì với cùng một lưu trú, có thể có nhiều môn thể thao khác nhau được thuê cho nhiều ngày khác nhau. Chẳng hạn đợt lưu trú số 5 có 2 ngày thuê canoe.

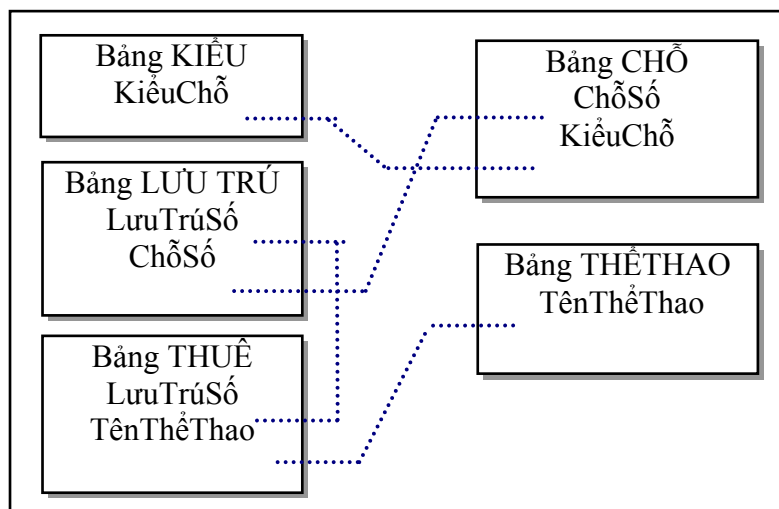
##### **b) Quan sát các mối liên hệ**

###### **Dữ liệu**

Dữ liệu khóa của một bảng có thể xuất hiện trong một hoặc nhiều bảng khác nhau. Trong khi đó, dữ liệu không phải là khóa chỉ xuất hiện đúng một lần trong một bảng, và trong CSDL.

Chẳng hạn, dữ liệu KiểuChỗ là khóa của bảng KIỂU có mặt trong bảng CHỖ. Như vậy, với một chỗ nào đó, có thể tìm ra giá tiền thuê một ngày cho một người, nhờ dữ liệu KiểuChỗ trong bảng CHỖ, rồi truy cập bảng KIỂU để tìm ra GiáNgàyNgười tương ứng.

Tương tự, dữ liệu ChỗSố là khóa của bảng CHỖ có mặt trong bảng LUUTRÚ. Lợi ích ở đây là có thể tìm ra các thông tin liên quan đến một chỗ cho một kỳ lưu trú đang xét, rồi tìm ra giá tiền bởi KiểuChỗ như vừa nói ở trên.



Nhớ rằng các dữ liệu khóa của bảng THUÊ, LưuTrúSố, TênThểThao và NgàyThuê, đều là khóa của các bảng khác. Ở đây, NgàyThuê là trường hợp đặc biệt sẽ giải thích ở mục sau.

### **Giá trị**

– Giá trị dữ liệu có mặt trong các bảng không thể không tương thích với nhau. Trong bảng CHỖ, giá trị của KiểuChỗ không thể khác với giá trị của KiểuChỗ trong bảng KIỂU. Vì lúc đó sẽ không xác định được giá tiền tương ứng để thanh toán.

Tương tự, giá trị của TênThểThao trong bảng THUÊ, phải có mặt trong bảng THỂTHAO. Nếu không, sẽ không thanh toán được những thuê bao về thể thao.

– Các giá trị dữ liệu khác cũng không thể không tương thích.

Chẳng hạn, giá trị NgàyThuê trong bảng THUÊ bắt buộc phải nằm giữa hai giá trị NgàyĐến và NgàyĐi trong bảng LUUTRÚ. Bởi vì chỉ có thể có thuê bao thể thao trong kỳ lưu trú.

Mặt khác, giá trị SốNgười trong bảng LUUTRÚ bắt buộc phải nhỏ hơn hoặc bằng giá trị SốNgMax trong bảng CHỖ với chỗ thuê tương ứng ChỗSố.

### **c) Khai thác cơ sở dữ liệu**

Sau khi tổ chức CSDL, vấn đề là khai thác nó. Có thể có những câu hỏi sau :

Cho biết giá thuê xe cắm trại (camping-car) cho một người trong một ngày ?

- Từ bảng KIỂU, tìm giá trị KiểuChỗ là “Xe cắm trại”.
- Xác định được GiáNgàyNgười là 95 000.

Cho biết thuê lều bạt ở những chỗ nào (tìm chỗ số) với diện tích lớn hơn 10m<sup>2</sup> ?

- Từ bảng CHỖ, tìm giá trị KiểuChỗ là “Lều bạt”.
- Chọn ra được một dòng ứng với lều bạt có diện tích lớn hơn 10 là chỗ số 2.

Cho biết giá áp dụng cho chỗ số 15 ?

- Giá tiền thuê nằm trong bảng KIỂU, tuy nhiên chưa có giá trị của KiểuChỗ.
- Giá trị KiểuChỗ được tìm thấy trong bảng Chỗ, là “Xe nhà ở”, ứng với chỗ số 15. Từ đó, dễ dàng tìm ra giá trị GiáNgàyNgười là 82 500.

Cho biết những môn thể thao nào nhóm khách đăng ký lưu trú số 104 đã thuê ?



- Từ bảng THUÊ, tìm những dòng có giá trị LưuTrúSố là 104. Ở đây tìm được 1 dòng có giá trị TênThẻThao là “Bóng bàn”. Ngoài ra còn biết ngày tháng nhóm khách này đã thuê và số đơn vị thời gian thuê phải thanh toán.
- Chọn ra được một dòng ứng với lều bạt có diện tích lớn hơn 10 là chỗ số 2.

Chú ý : Trên đây chỉ là những ví dụ tùy ý. Thông thường, người ta phải tìm ra những thông tin cần thiết để xác định giá tiền phải trả theo quy định cho một nhóm khách đã lưu trú vào thời điểm lên đường của họ.

#### II.4.3. Mô hình quan hệ tương ứng

Những bảng dữ liệu vừa xét trên đây thể hiện khía cạnh áp dụng thực tế của một hệ QLCSDL. Chúng được biểu diễn dưới dạng mô hình quan hệ kiểu logic của CSDL.

##### a) Sơ đồ biểu diễn

Mỗi bảng thể hiện một quan hệ. Các dữ liệu tham gia vào quan hệ tạo thành sơ đồ biểu diễn quan hệ này. Tập hợp các sơ đồ biểu diễn quan hệ tạo thành sơ đồ biểu diễn CSDL.

KIẾU (KiểuChỗ, GiáNgàyNgười)

CHỖ (ChỗSố, DiệnTích, SốNgMax, #KiểuChỗ)

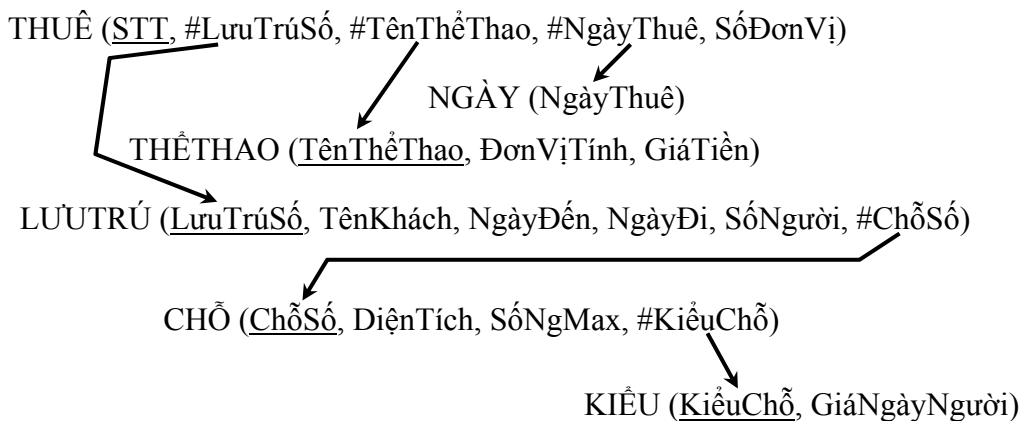
LU'UTRÚ (LưuTrúSố, TênKhách, NgàyĐến, NgàyĐi, SốNgười, #ChỗSố)

THÊTHAO (TênThẻThao, ĐơnVịTinh, GiáTiền)

THUÊ (STT, #LưuTrúSố, #TênThẻThao, #NgàyThuê, SốĐơnVị)

NGÀY (NgàyThuê)

Biểu diễn không gian :



Hình 4.53 Biểu diễn không gian của các quan hệ

##### b) Xét các quan hệ

Một cách tổng quan, ta nhìn thấy cấu trúc dữ liệu của các bảng, nhưng không nhìn thấy các giá trị. Mỗi bảng tương ứng với một sơ đồ biểu diễn quan hệ giữa các dữ liệu. Các dữ liệu khoá được gạch chân.

Một số dữ liệu có dấu # đặt trước. Đó là những dữ liệu có vai trò cụ thể hoá mối ràng buộc giữa các quan hệ. Khoá trong một quan hệ (chẳng hạn KiểuChỗ trong bảng KIẾU) không là dữ liệu khoá trong một quan hệ khác (KiểuChỗ trong bảng CHỖ), được xem là khoá ngoại lai.

Bảng THUÊ là trường hợp đặc biệt có khoá được tạo bởi 3 khoá ngoại lai LưuTrúSố, TênThểThao và NgàyThuê.

Những quan hệ bao gồm một hoặc nhiều dữ liệu (thường được gọi là thuộc tính trong mô hình quan hệ) có dấu # đứng trước được gọi là những quan hệ động, vì chúng cho phép thiết lập mối ràng buộc giữa các quan hệ (các quan hệ CHỖ, LƯU TRÚ, và THUÊ). Những quan hệ khác được gọi là tĩnh (các quan hệ KIỂU, THỂTHAO và NGÀY).

**Chú ý :**

Quan hệ NGÀY chỉ có một dữ liệu NgàyThuê mà không có bảng tương ứng. Trong CSDL, khoá NgàyThuê là ngoại lai đối với quan hệ THUÊ. Sử dụng quan hệ NGÀY để tham chiếu kiểm tra.

#### ***II.4.4. Mô hình thực thể – kết hợp***

**a) Nhận xét**

Trong MHYNDL, không có sự dư thừa dữ liệu. Các mối ràng buộc (khóa ngoại lai) trong mô hình quan hệ được thể hiện bởi các kết hợp giữa các thực thể.

Kết hợp “thuộc về/tính tiền” giữa các thực thể CHỖ và KIỂU cho biết một chỗ chỉ thuộc về 1 và chỉ 1 KiểuChỗ, bản số (1-1), một KiểuChỗ ứng với từ 1 đến n chỗ, bản số (1-n).

Kết hợp “tiếp nhận/triển khai” giữa các thực thể CHỖ và LƯU TRÚ chỉ rằng một đợt lưu trú chỉ thực hiện trên 1 và chỉ 1 chỗ, bản số (1-1), và một chỗ có thể không có, hoặc có nhiều khách đến lưu trú trong một mùa du lịch, bản số (0-n).

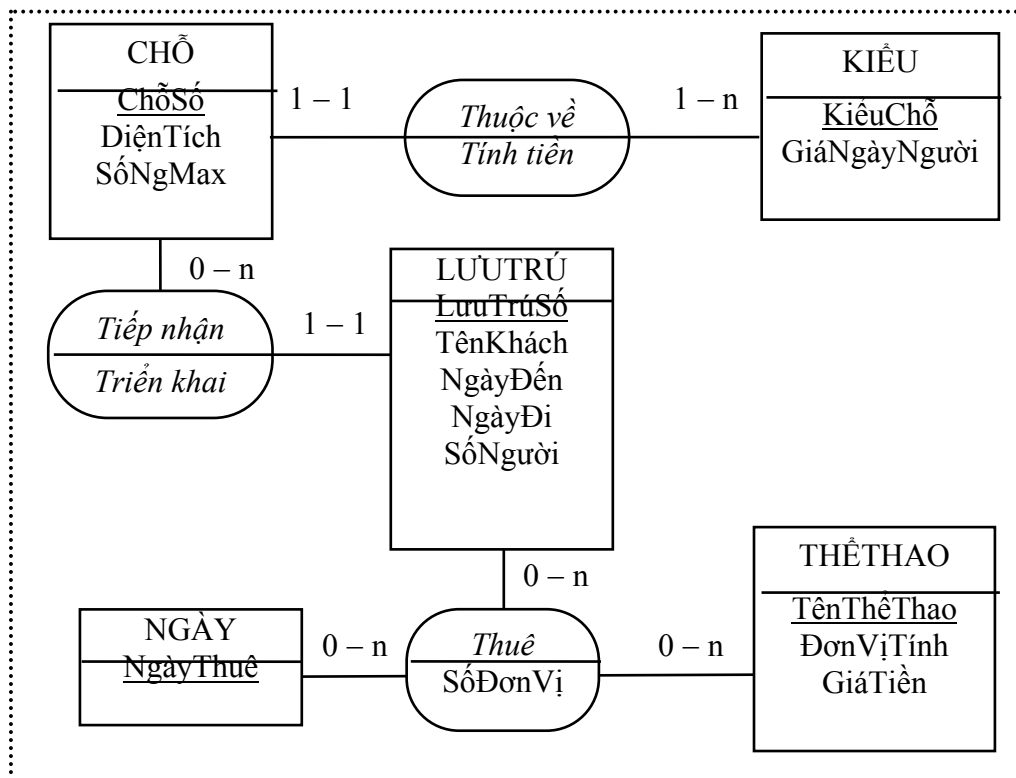
Kết hợp “thuê” có nghĩa trong một đợt lưu trú, người ta ghi lại một số lượng đơn vị thời gian cho một phương tiện thể thao và vào một ngày đã cho nào đó.

Trong mùa du lịch, với một lưu trú, không có hoặc có nhiều, với một thể thao có thể không có hoặc có nhiều cuộc thuê, bản số (0-n), và cũng có thể vào một ngày nào đó, không có hoặc có nhiều người thuê phương tiện thể thao, bản số (0-n).

Nhưng cũng có thể trong một mùa du lịch, có lưu trú nhưng không có hoặc có nhiều người thuê phương tiện thể thao, một thể thao có thể không ai thuê hoặc có nhiều người thuê, và vào một ngày nào đó, có thể không có hoặc có nhiều người thuê phương tiện thể thao, đều bản số (0-n).

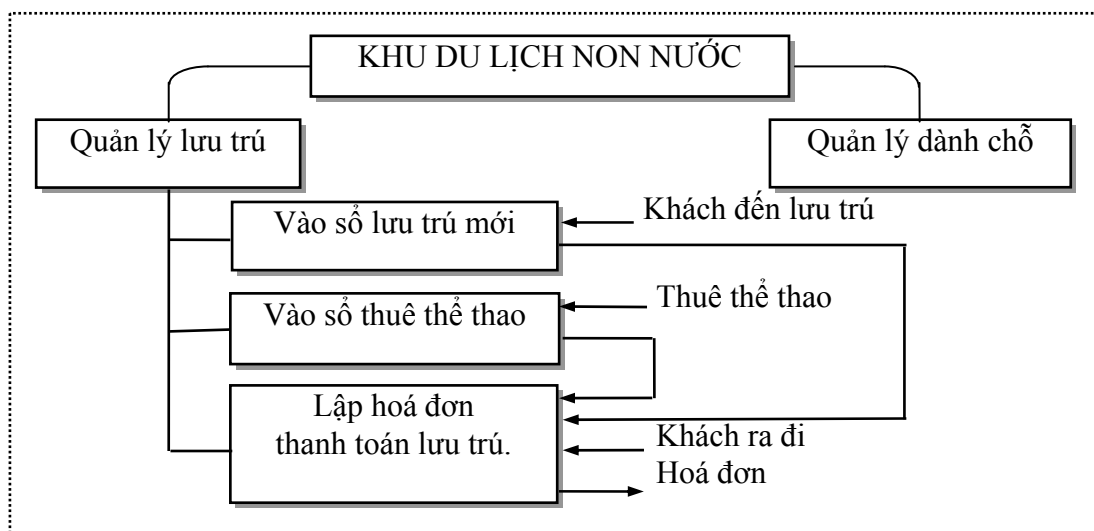
**b) Biểu diễn mô hình**

Mô hình thực thể - kết hợp cho CSDL “Khu du lịch Non Nước” như sau :



Hình 4. 54 Mô hình ý niệm dữ liệu “Khu du lịch Non Nước”

**c) Mở rộng cơ sở dữ liệu**



Hình 4.55 Mở rộng cơ sở dữ liệu Khu Du lịch Non Nước

Ngoài những câu hỏi ví dụ đã đưa ra ở mục trước, mục đích cuối cùng của cơ sở dữ liệu là lập hóa đơn thanh toán cho khách hàng vào thời điểm họ ra đi. Trong ví dụ về cơ sở dữ liệu “Khu du lịch Non Nước”, ta mới chỉ xét việc quản lý lưu trú. Lĩnh vực chưa được xem xét đến là quản lý dành chỗ.

### III. Các công cụ biểu diễn PTH cho mô hình E–A

Khái niệm về PTH cũng được áp dụng trong các mô hình thực thể - kết hợp. Nếu trong một thực thể, tồn tại PTH giữa dữ liệu là khoá A và một dữ liệu là đặc tính B, người ta biểu diễn PTH dưới dạng :  $A \rightarrow B$

Nếu giá trị của A đã biết (nguồn) thì giá trị của B (đích) cũng được biết.

Trong trường hợp tồn tại nhiều PTH, có hai công cụ để biểu diễn là *ma trận các PTH* và *đồ thị các PTH*. Hai công cụ này bổ sung cho nhau.

#### III.1.1. Ma trận các phụ thuộc hàm

Để xây dựng ma trận các PTH, người ta sử dụng lại từ điển dữ liệu đã xây dựng trong bước phân tích hiện trạng. Mọi dữ liệu sơ cấp trong từ điển được đặt lần lượt trên các dòng để làm đích của các PTH. Sau đó chúng lại được đặt lần lượt trên các cột để biểu diễn các nguồn của PTH. Có 2 kiểu ma trận :

*Ma trận đầy đủ* : mọi dữ liệu sơ cấp được đặt hết trên dòng và trên cột tạo thành một ma trận vuông cho phép xét hết mọi khả năng của PTH.

*Ma trận rút gọn* : chỉ những dữ liệu có thuộc tính khoá mới được đặt trên cột. Như vậy, ma trận trở nên dễ đọc nhưng trước đó phải hiểu ma trận dạng đầy đủ và không có sự nhập nhằng trên các khoá.

Ma trận đầy đủ :

Đích	Nguồn				
	1	2	3	4	5
A	1			1	
B	1	1			
C			1		
D			1	1	
E					1

Ma trận rút gọn :

Đích	Nguồn		
	1	3	4
<u>A</u>	1		1
B	1		
<u>C</u>		1	
<u>D</u>		1	1
E			

Đối với ma trận đầy đủ :

Đường chéo chính mặc nhiên chứa các số 1 thể hiện tính chất phản xạ của các PTH.

Các cột có chứa nhiều hơn một số 1 tương ứng với các dữ liệu là khóa, trong đó, một số 1 trên một hàng thể hiện tính chất phản xạ (số thứ tự hàng và cột giống nhau) và một hoặc nhiều số 1 trên các hàng khác tương ứng với các dữ liệu đích của PTH mà dữ liệu trên cột tương ứng đóng vai trò nguồn. Ở đây, các khóa là A, C và D.

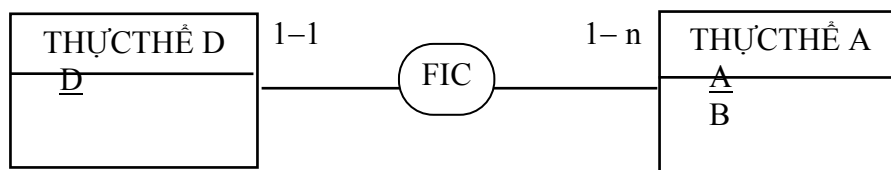
Đối với ma trận rút gọn :

Ma trận có dạng hình chữ nhật, đường chéo chính biến mất cùng với các PTH phản xạ.

Các dữ liệu khóa được gạch chân (in đậm).

Nguồn của mỗi PTH bắt buộc là một khóa, tuy nhiên đích có thể không phải là khóa. Nếu đích không là khóa, thì nguồn phải là khóa của một thực thể, như trường hợp dữ liệu B ( $A \rightarrow B$ ).

Nếu đích là một khóa, vấn đề là một kết hợp có phân cấp giữa hai thực thể. Đó là trường hợp giữa A và D.



Hình 4.56 Kết hợp phân cấp xây dựng từ ma trận rút gọn

Các dữ liệu có thể bị đặt tách biệt, nghĩa là chúng không phải đích của bất kỳ một PTH nào có nguồn là dữ liệu sơ cấp. Đó là trường hợp của dữ liệu E. Cần phải tìm một phụ thuộc hàm sao cho nguồn là các khóa ghép lại, còn E là đích.

Chẳng hạn A + C (tức 1+3) cho một kết hợp không phân cấp giữa các thực thể tương ứng. Có thể bổ sung một cột mới để biểu diễn PTH này với nguồn là A + C, đích là E. Nếu muốn chỉ PTH này là phản xạ, có thể thêm một dòng A + C.

Đích	Nguồn			
	1	3	4	1+3
1. <u>A</u>	1		1	1
2. <u>B</u>	1			
3. <u>C</u>		1		1
4. <u>D</u>		1	1	
5. <u>E</u>				1

- Một dữ liệu khi có mặt trong một thực thể hay trong một kết hợp, sẽ chỉ có một số 1 trên dòng tương ứng. Tuy nhiên, có hai khả năng :
  - Một mặt, *các dữ liệu không phải là khóa chỉ có thể có một số 1 trên các dòng tương ứng*. Nếu không phải như vậy, thì chắc chắn rằng đó là dấu hiệu của PTH có tính bắc cầu cần loại bỏ (ở đây là hai dữ liệu B và E).
  - Mặt khác, các dữ liệu là khóa có thể có nhiều số 1 trên các dòng tương ứng. Một số 1 thể hiện vai trò là khóa (cùng số thứ tự dòng và cột), còn các số 1 khác chỉ ra các kết hợp phân cấp (dòng 1, cột 4) hoặc các kết hợp không phân cấp (cột 1 + 3 và các dòng 1, 3).

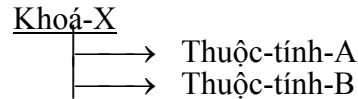
Chú ý các dữ liệu là tham số, hay các hệ số tính toán, không phải là nguồn cũng không phải là đích của PTH. Có thể đánh dấu chúng bằng cách ghi thêm “tham số” trên những dòng tương ứng.

### III.1.2. Đồ thị các PTH

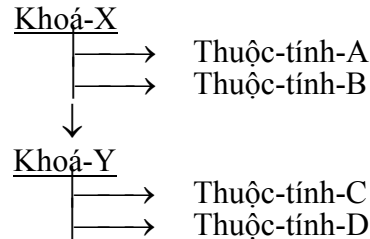
Để xác định và loại bỏ các PTH có tính bắc cầu giữa các khóa, người ta biểu diễn các PTH bằng đồ thị, mà ở dạng biểu diễn ma trận không thể xác định được.

Cách biểu diễn như sau :

PTH giữa khóa và các thuộc tính khác trong cùng một thực thể.

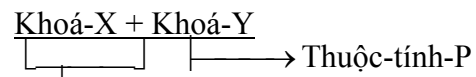


PTH giữa các khóa của nhiều thực thể nối với nhau bởi một kết hợp phân cấp (FIC).

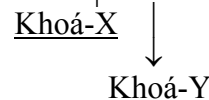


Các PTH có nguồn là dữ liệu ghép được biểu diễn :

dữ-liệu-1 + dữ-liệu-2.



Giữa các khóa của nhiều thực thể nối với nhau bởi các kết hợp không phân cấp (MIC).

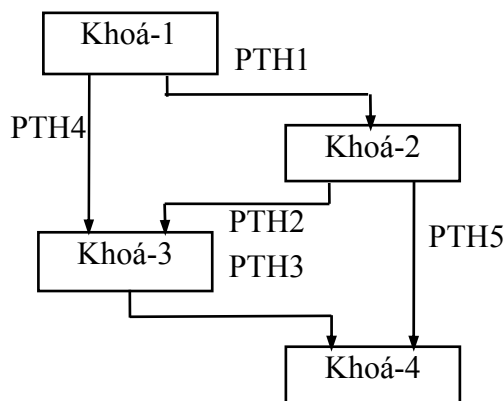


Với các PTH có đích là dữ liệu không phải khóa, việc kiểm tra rất dễ dàng : trong ma trận rút gọn các PTH, những dữ liệu này chỉ có thể có duy nhất một số 1 trên các dòng tương ứng, chỉ ra rằng chúng không là đích của một PTH nào. Do vậy ta cũng dễ dàng biểu diễn được các PTH giữa các khóa bằng đồ thị.

Ngược lại, đối với các dữ liệu đồng thời vừa là nguồn, vừa là đích của PTH, có thể tồn tại các PTH bắc cầu tương ứng với các số 1 trên cùng một dòng trong ma trận các PTH. Vấn đề này thường xảy ra khi cấu trúc dữ liệu làm xuất hiện nhiều PTH giữa các khóa có dạng “thác đổ” (cascade) :

khoá-1 → khoá-2 → khoá-3 → khoá-4

Đồ thị biểu diễn các PTH bắc cầu này như sau :



Hình 4.57 Các phụ thuộc hàm bắc cầu

Các PTH1, PTH2 và PTH3 thể hiện tính bắc cầu như trên. Các PTH4 và PTH5 là kết quả của tính chất bắc cầu của các PTH. Cần phải loại bỏ các PTH bắc cầu trong đồ thị các PTH, cũng như loại bỏ các số 1 tương ứng trong ma trận các PTH để đảm bảo tính thuần nhất giữa ma trận, đồ thị và mô hình dữ liệu.

## III.2. Ví dụ ứng dụng phụ thuộc hàm

Ta tiếp tục xét ví dụ về ứng dụng PTH cho CSDL Xí nghiệp DanaFood.

### III.2.1. Ma trận PTH

#### a) Ma trận PTH dạng đầy đủ

Lập ma trận theo cột, mỗi cột tương ứng với một dữ liệu là nguồn của PTH. Sau đó cần đặt câu hỏi cho mỗi dữ liệu ứng với mỗi dòng. Như vậy, với mỗi cột, nếu dữ liệu tương ứng thực sự là nguồn của một PTH thì ta ghi 1 trên những dòng chứa dữ liệu là đích của PTH này. Cần chú ý tính đối xứng của phương pháp. Tất cả những PTH sơ cấp, nghĩa là những PTH có nguồn là một dữ liệu sơ cấp, phải được làm rõ.

STT	Dữ liệu	Quy tắc quản lý
1	ChiDẫnHg	Không là nguồn của PTH
2	DSốNăm-1	Không là nguồn của PTH
3	DSốTháng	Không là nguồn của PTH
4	ĐịaChỉKH	Không là nguồn của PTH
5	GiáĐơnVị	Không là nguồn của PTH
6	GiáNơiBán	Không là nguồn của PTH
7	GiaTiền	Không là nguồn của PTH
8	HạnGiaoHg	Không là nguồn của PTH
9	MãHàng	Nguồn của các PTH liên quan đến chỉ dẫn về mặt hàng, SLCó, ...
10	MãKH	Nguồn của các PTH liên quan đến khách hàng.
11	NơiBánHg	Nguồn của nhiều PTH, kéo theo sự nhận biết NgàyBán, GiáNơiBán, SLBán, HạnGiaoHg, về khách hàng, mặt hàng và việc giao hàng tương ứng.
12	NgàyĐặtHg	Không là nguồn của PTH
13	NgàyBán	Không là nguồn của PTH
14	NgàyCóHg	Không là nguồn của PTH
15	NgàyGiao	Không là nguồn của PTH
16	NgàyKThúc	Không là nguồn của PTH
17	SốHĐĐặtHg	Nguồn của nhiều PTH, kéo theo sự nhận biết những thông tin về đơn đặt hàng : NgàyĐặtHg, GiaTiền, SLĐặt, về khách hàng, mặt hàng và việc giao hàng tương ứng.
18	SốHĐGiaoHg	Nguồn của nhiều PTH, kéo theo sự nhận biết NgàyGiao, những thông tin liên quan đến khách hàng hay nơi bán và mặt hàng.
19	SLĐặt	Không là nguồn của PTH
20	SLBán	Không là nguồn của PTH
21	SLSX	Không là nguồn của PTH
22	TênKH	Không là nguồn của PTH, do không xác định rõ được địa chỉ của khách hàng.
23	TênTháng	Không là nguồn của PTH
24	Tháng	Duy nhất PTH cho biết tên tháng
25	SLBảoĐộng	Không là nguồn của PTH
26	SLCó	Không là nguồn của PTH
27	ĐiệnThKH	Không là nguồn của PTH, mặc dù có thể xác định được khách hàng

Trong ma trận PTH đầy đủ, các chữ N (Nguồn) được đặt ở đầu một số cột để chỉ nguồn của các PTH. Đó là :

12	SốHĐGiaoHg	15	NơiBánHg
13	SốHĐĐặtHg	16	Tháng
14	MãKH	23	MãHàng

Những dữ liệu trên là những nguồn duy nhất của PTH, chúng đóng vai trò quan trọng đặc biệt trong việc thiết kế CSDL. Mỗi một trong chúng cho phép định danh các đối tượng có quan hệ tới. Ví dụ, SốHĐGiaoHg cho phép định danh duy nhất một hoá đơn giao hàng, vì không thể có nhiều hoá đơn giao hàng có cùng SốHĐGiaoHg trong CSDL. Ma trận PTH dạng đầy đủ của XN DanaFood :

	Đích	Nguồn																											
														N	N	N	N	N							N				
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	
1	DSốNăm-1	1											1	1	1	1													
2	DSốTháng		1																										
3	NgàyĐặtHg			1					1				1	1															
4	NgàyCóHg				1								1	1		1								1					
5	NgàyKThúc					1							1	1										1					
6	NgàyGiao						1						1	1		1													
7	NgàyBán							1					1			1													
8	HạnGiaoHg								1							1													
9	ChỉDẫnHg									1			1	1		1								1					
10	TênTháng										1						1												
11	GiaTiền											1	1	1															
12	SốHĐGiaoHg												1	1		1													
13	SốHĐĐặtHg												1	1															
14	MãKH												1	1	1	1													
15	NơiBánHg												1			1													
16	Tháng																1												
17	TênKH												1	1	1			1											
18	GiáNơiBán															1			1										
19	GiáĐơnVị												1	1		1				1				1					
20	SLĐặt												1	1							1								
21	SLSX												1	1		1						1		1					
22	SLBán															1							1						
23	MãHàng												1	1		1								1					
24	ĐịaChiKH												1	1	1	1									1				
25	SLBáoĐộng												1	1		1							1			1			
26	SLCó												1	1		1							1				1		
27	ĐiệnThKH												1	1	1	1												1	



*b) Vấn đề không dư thừa dữ liệu*

Trong các mô hình thực thể - kết hợp, các dữ liệu được đặt trong các thực thể hoặc các kết hợp. Tuy nhiên, mỗi dữ liệu chỉ có thể xuất hiện trong một thực thể hoặc trong một kết hợp. Một dữ liệu chỉ có thể được đặt trong thực thể này, mà không được đặt trong thực thể kia, bởi nó là đích của một PTH. Nguồn của PTH này là khoá định danh của thực thể.

Hậu quả là, một dữ liệu không phải khoá định danh của thực thể thì chỉ có thể là đích của một PTH. Điều này có nghĩa rằng trong ma trận các PTH, mỗi dòng chỉ có thể có một số 1 duy nhất.

Trong ma trận đầy đủ, các số 1 trên đường chéo chính thể hiện tính chất phản xạ của các PTH. Nhưng đối với các số 1 khác, có một sự lựa chọn giữa các nguồn khác nhau của PTH đã xác định.

Ví dụ :

Trên dòng 1, DSốNăm-1 là đích của 4 PTH :

SốHĐGiaoHg	→	DSốNăm-1
SốHĐĐặtHg	→	DSốNăm-1
MãKH	→	DSốNăm-1
NơiBánHg	→	DSốNăm-1

Ta cũng có MãKH là đích của 3 PTH :

SốHĐGiaoHg	→	MãKH
SốHĐĐặtHg	→	MãKH
NơiBánHg	→	MãKH

Do tính chất bắc cầu của PTH, nếu  $a \rightarrow b$  và  $b \rightarrow c$ , thì  $a \rightarrow c$ , do đó :  
nếu :

SốHĐĐặtHg  $\rightarrow$  MãKH và MãKH  $\rightarrow$  DSốNăm-1,

thì :

SốHĐĐặtHg  $\rightarrow$  SốNăm-1.

Tương tự như vậy đối với SốHĐGiaoHg và NơiBánHg. Do sự vô ích của các PTH có tính bắc cầu mà người ta phải bỏ chúng đi. Như vậy, 4 PTH của DSốNăm-1 chỉ còn lại MãKH  $\rightarrow$  DSốNăm-1.

Trong ma trận rút gọn của PTH, người ta chỉ giữ lại những cột dữ liệu nguồn của PTH và chỉ chú ý đến những PTH không có tính bắc cầu.

Các PTH phản xạ sẽ không được đưa vào ma trận, ngoại trừ các khoá (dữ liệu nguồn sơ cấp của PTH).

*c) Các PTH giữa các khoá*

Khi trên dòng dữ liệu là khoá có nhiều 1, có thể dẫn đến các PTH bắc cầu và vấn đề là phải khử bỏ chúng đi như đã thấy. Tuy nhiên trong trường hợp này cũng có những kết hợp phân cấp (hay ràng buộc toàn vẹn hàm - CIF) giữa các thực thể mà khoá của chúng là PTH.

Xét dòng 23 có MãHàng là đích của 3 PTH với các nguồn là SốHĐGiaoHg, SốHĐĐặtHg và NơiBánHg.

Ta có : SốHĐĐặtHg  $\longrightarrow$  MãHàng vì đơn đặt hàng chỉ có thể đặt một sản phẩm. Ngược lại, một sản phẩm có thể tìm thấy trong nhiều đơn đặt hàng. Sự PTH này dẫn đến một kết hợp phân cấp giữa các thực thể ĐDHÀNG và SẢN PHẨM.

Đối với NơiBánHg, ta có PTH là NơiBánHg  $\longrightarrow$  MãHàng. Nghĩa là một nơi bán chỉ liên quan đến một sản phẩm, còn một sản phẩm có thể bán ở nhiều nơi, hoặc không có nơi nào.

Như vậy có một kết hợp phân cấp giữa các thực thể NƠI BÁN và SẢN PHẨM

Ta cũng thấy PTH :      SốHĐGiaoHg       $\longrightarrow$  MãHàng      có tính bắc cầu vì :  
SốHĐGiaoHg       $\longrightarrow$  SốHĐĐặtHg

và :

SốHĐĐặtHg       $\longrightarrow$  MãHàng.

Trong ma trận rút gọn các PTH, ta chỉ giữ lại các PTH giữa các khoá mà không giữ lại các PTH có tính bắc cầu.

**III.2.2. Ma trận rút gọn các PTH**

*a) Các PTH sơ cấp*

Một PTH được gọi là sơ cấp nếu nguồn là một dữ liệu sơ cấp, không phải dạng ghép (tổ hợp) của nhiều dữ liệu.

Giống ma trận đầy đủ các PTH, ma trận rút gọn được xây dựng từ các PTH sơ cấp.

*b) PTH trực tiếp*

Một PTH là trực tiếp nếu nó không là bắc cầu. Ma trận rút gọn chỉ chứa các PTH trực tiếp.

c) Ma trận rút gọn

Đích		Nguồn						
		12	13	14	15	16	23	A
1	DSốNăm-1			1				
2	DSốTháng							1
3	NgàyĐặtHg		1					
4	NgàyCóHg						1	
5	NgàyKThúc						1	
6	NgàyGiao	1						
7	NgàyBán				1			
8	HạnGiaoHg				1			
9	ChỉDẫnHg						1	
10	TênTháng					1		
11	GiaTiền		1					
12	SốHĐGiaoHg	<u>1</u>	1		1			
13	SốHĐĐặtHg	1	<u>1</u>					
14	MãKH		1	<u>1</u>	1			1
15	NơiBánHg	1			<u>1</u>			
16	Tháng					<u>1</u>		1
17	TênKH			1				
18	GiáNơiBán				1			
19	GiáĐơnVị						1	
20	SLĐặt		1					
21	SLSX						1	
22	SLBán				1			
23	MãHàng		1		1		1	
24	ĐịaChỉKH			1				
25	SLBảoĐộng						1	
26	SLCó						1	
27	ĐiệnThKH			1				

$$A = 14 + 16 (\text{MãKH} + \text{Tháng})$$

Các PTH giữa các khoá liên quan đến SốHĐGiaoHg, có hai vấn đề như sau :

PTH SốHĐĐặtHg  $\rightarrow$  SốHĐGiaoHg là một PTH không hoàn toàn, bởi vì biết một giá trị của SốHĐĐặtHg có thể kéo theo sự nhận biết nhiều giá trị của SốHĐGiaoHg. Nhưng chừng nào mà việc giao hàng ứng với một đơn đặt hàng chưa được ghi nhận thì chưa thể kéo theo sự nhận biết giá trị của SốHĐGiaoHg. Tương tự như vậy cho PTH : NơiBánHg  $\rightarrow$  SốHĐGiaoHg.

Hai PTH trên có tính tương hỗ qua lại. Nghĩa là ta có các PTH :

SốHĐGiaoHg  $\rightarrow$  SốHĐĐặtHg và SốHĐGiaoHg  $\rightarrow$  NơiBánHg.

Đây là trường hợp đặc biệt. Hơn nữa, hai PTH này là sơ cấp vì biết được giá trị của SốHĐGiaoHg thì biết được hoặc một giá trị của SốHĐĐặtHg, hoặc một giá trị của NơiBánHg. Như vậy, một giao hàng có thể chỉ ứng với một nơi bán, hoặc chỉ ứng với một đơn đặt hàng (loại trừ lẫn nhau).

### III.2.3. Các PTH không sơ cấp

Ta đã xét các PTH sơ cấp. Nhưng không phải luôn luôn mọi PTH tạo nên CSDL đều có nguồn là một dữ liệu duy nhất. Một số PTH không sơ cấp có nguồn là sự ghép nhiều dữ liệu lại với nhau, được gọi là PTH về trái ghép.

Trong ma trận rút gọn, các khoá và các số 1 được gạch chân để dễ phân biệt. Các dòng dữ liệu không là khoá chỉ có một số 1 duy nhất thể hiện các PTH sơ cấp và trực tiếp. Trường hợp dòng 2 cho dữ liệu DSốTháng sẽ xét sau.

Đối với các dòng tương ứng với dữ liệu khoá, trừ một số 1 thể hiện tính phản xạ, các số 1 khác thể hiện các kết hợp phân cấp (CIF).

Chẳng hạn, dòng 14 cho MãKH, hai số 1 không gạch chân thể hiện hai CIF :

một số 1 tương ứng với thực thể ĐDHÀNG có khoá là SốHĐĐặtHg :

SốHĐĐặtHg  $\rightarrow$  MãKH,  
một số 1 sẽ tương ứng với thực thể

NƠIBÁN có khoá là NơiBánHg :

NơiBánHg  $\rightarrow$  MãKH

Chú ý :

Trước hết cần nhớ rằng các đơn đặt hàng, cũng như các nơi bán, thường được yêu cầu giao hàng toàn bộ một lúc, và, một giao hàng chỉ liên quan đến một đơn đặt hàng hoặc một nơi bán mà thôi.

Tuy nhiên, đơn đặt hàng hay nơi bán được ghi nhận rồi mà việc giao hàng có thể chưa được thực thi.

Các PTH không sơ cấp xuất hiện trong ma trận các PTH là do không tồn tại những PTH có nguồn là dữ liệu sơ cấp.

Đó là trường hợp của DSốTháng. Không có PTH mà MãKH là nguồn vì với một khách hàng, có bao nhiêu tháng đã qua thì cũng có bấy nhiêu DSốTháng. Tương tự như vậy, đối với PTH có nguồn là Tháng, vì với một tháng đã cho, có bao nhiêu khách hàng hiện diện trong CSDL trong tháng thì cũng có bấy nhiêu DSốTháng.

Như vậy, để biết duy nhất một giá trị DSốTháng, cần biết MãKH và Tháng. Cấu trúc này được thể hiện bởi một bảng hai chiều, một chiều là các giá trị của MãKH, và chiều kia là các giá trị của Tháng. Ta có một kết hợp không phân cấp (hay ràng buộc toàn vẹn bội CIM) như sau :

MãKH, Tháng  $\longrightarrow$  DSốTháng

Đây là những PTH về trái ghép. Trong ma trận rút gọn các PTH, ta thêm một cột và một dòng cho PTH không sơ cấp. Thực ra thì chỉ cần thêm một cột, là cột A = 14 + 16, với số 1 ở dòng 2, dòng 14 và dòng 16. Sau khi xây dựng xong ma trận rút gọn các PTH theo cách này, các dữ liệu đều được đặt, hoặc trong một thực thể (chúng là đích của các PTH có nguồn là sơ cấp), hoặc trong một kết hợp không phân cấp (chúng là đích của các PTH có nguồn là do nhiều dữ liệu ghép lại).

*Chú ý :*

Mọi kết hợp không phân cấp (CIM) không mang dữ liệu, như trong ví dụ chúng ta vừa xét. Vấn đề là không tồn tại những dữ liệu tách biệt khỏi những PTH sơ cấp. Chúng được biểu diễn như sau :

khoá-1, khoá-2  $\rightarrow$  0

### **III.2.4. Kết luận**

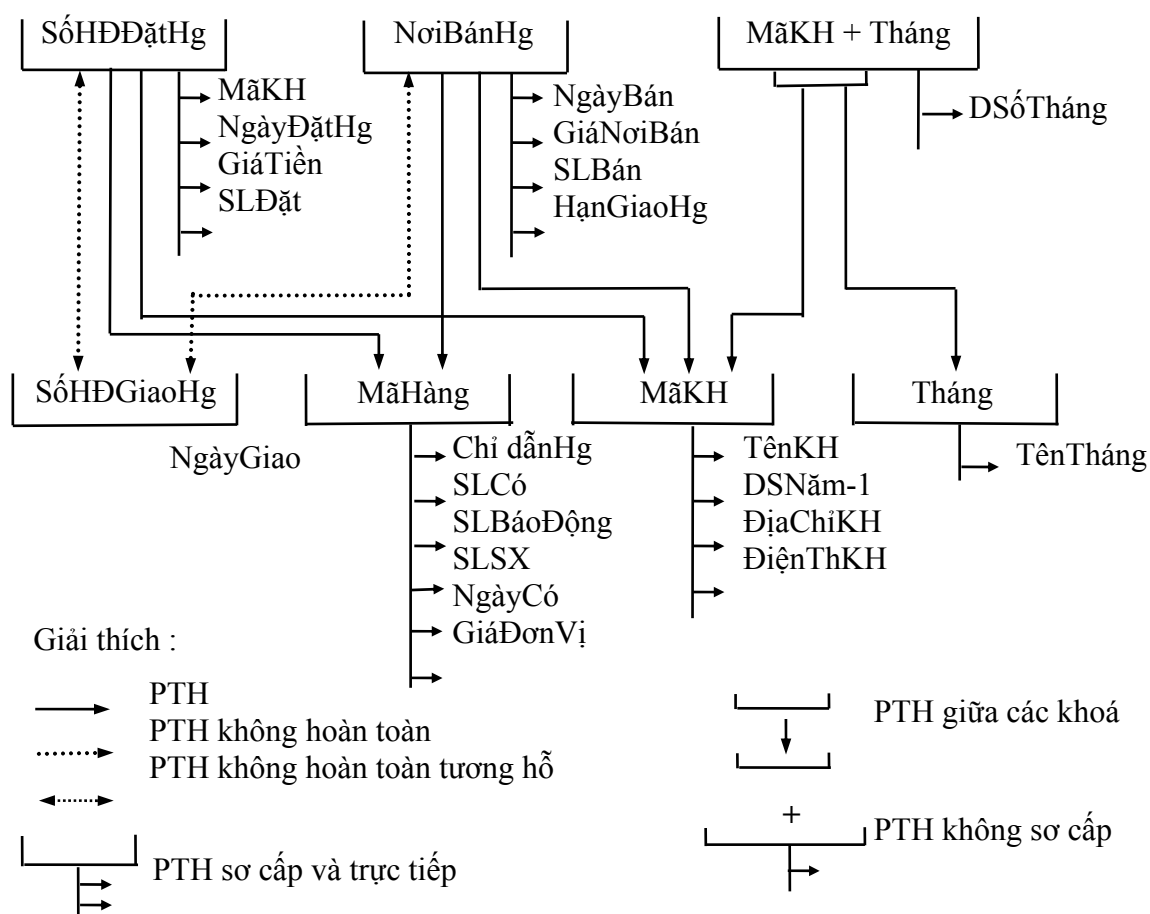
Mỗi thực thể được thể hiện bởi một khoá và các dữ liệu là thuộc tính của thực thể trong đó, khoá là nguồn, các dữ liệu là các đích của PTH (sơ cấp và trực tiếp). Ví dụ, tại cột 14, thực thể là KHHÀNG, khoá là MãKH, và các thuộc tính là DSốNăm-1, MãKH, TênKH và ĐịaChỉKH.

Hai kiểu kết hợp được xem xét như sau :

– Kết hợp phân cấp (kiểu CHA-CON, sẽ xét ở chương sau) với các PTH giữa các khoá. Chẳng hạn, cột 14 có số 1 dòng 14 biểu diễn một kết hợp phân cấp giữa các thực thể ĐDHÀNG và KHHÀNG. Một đơn đặt hàng chỉ liên quan đến một khách hàng nhưng một khách hàng lại có thể đặt nhiều đơn đặt hàng.

– Kết hợp không phân cấp (kiểu BẢNG, sẽ xét ở chương sau) với các PTH về trái ghép. Chẳng hạn, duy nhất trong ví dụ, cột 14+16, hai thực thể liên quan là KHHÀNG và THÁNG. Kết hợp này có một dữ liệu là DSốTháng.

### III.3. Đồ thị PTH biểu diễn CSDL của nhà máy đóng hộp DanaFood



Hình 4.58 Đồ thị các PTH của CSDL DanaFood

### III.4. Chuyển đổi giữa mô hình dữ liệu và đồ thị PTH

Làm thế nào để chuyển đổi một đồ thị PTH thành mô hình thực thể - kết hợp ?

#### III.4.1. PTH có nguồn là dữ liệu sơ cấp

Nếu PTH có nguồn là dữ liệu sơ cấp thì xây dựng thực thể có khoá là nguồn của PTH đã cho. Ví dụ : SốHĐĐH là khoá của thực thể ĐDH-KH

#### III.4.2. PTH sơ cấp giữa khoá và các dữ liệu sơ cấp

Nếu là PTH sơ cấp giữa khoá và các dữ liệu sơ cấp thì xây dựng thực thể có các thuộc tính là các dữ liệu sơ cấp và khoá của thực thể là nguồn của PTH đã cho.

Ví dụ :

Thực thể :

Khoá :

Các thuộc tính :

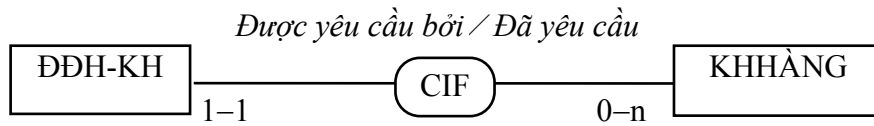
ĐDH-KH
<u>SốHĐĐặtHg</u>
MãKH
NgàyĐặtHg
GiáTiền
SLĐặt

SẢN PHẨM
<u>MãKH</u>
SLCó
SLBáoĐộng
SLSX
NgàyCó
GiáĐơnVị

### III.4.3.PTH sơ cấp giữa các khoá

Nếu là PTH sơ cấp giữa các khoá thì tạo ra một kết hợp phân cấp (CIF).

Ví dụ : PTH giữa các khoá : SốHĐặtHg → MãKH được chuyển thành :



Hình 4.59 Kết hợp phân cấp

Kết hợp “Được yêu cầu bởi/Đã yêu cầu” là phân cấp vì rằng bản số cực đại bên trái bằng 1 và bản số bên phải bằng n. Một đơn đặt hàng chỉ có thể được yêu cầu bởi một và chỉ một khách hàng, bản số (1-1), trong khi đó, một khách hàng có thể yêu cầu nhiều đơn đặt hàng, bản số cực đại sẽ là n.

Khách hàng có thể đã yêu cầu tối thiểu 1 hoặc 0 đơn đặt hàng. Ở đây, với mỗi xử lý, CSDL đặt giá trị 0 cho mỗi dữ liệu liên quan đến việc quản lý đơn đặt hàng và nơi bán hàng. Ngược lại, giá trị của mỗi dữ liệu thuộc các thực thể KHHÀNG và SẢN PHẨM được chép lại từ xử lý trước đó.

Như vậy, một khách hàng dù có mặt trong CSDL nhưng có thể vẫn chưa yêu cầu một đơn đặt hàng nào. Điều này giải thích vì sao bản số tối thiểu là 0.

### III.4.4.PTH không sơ cấp

Với các PTH loại về trái ghép này, xây dựng kết hợp không phân cấp (CIM). Các thành phần khoá chỉ ra các thực thể liên quan.

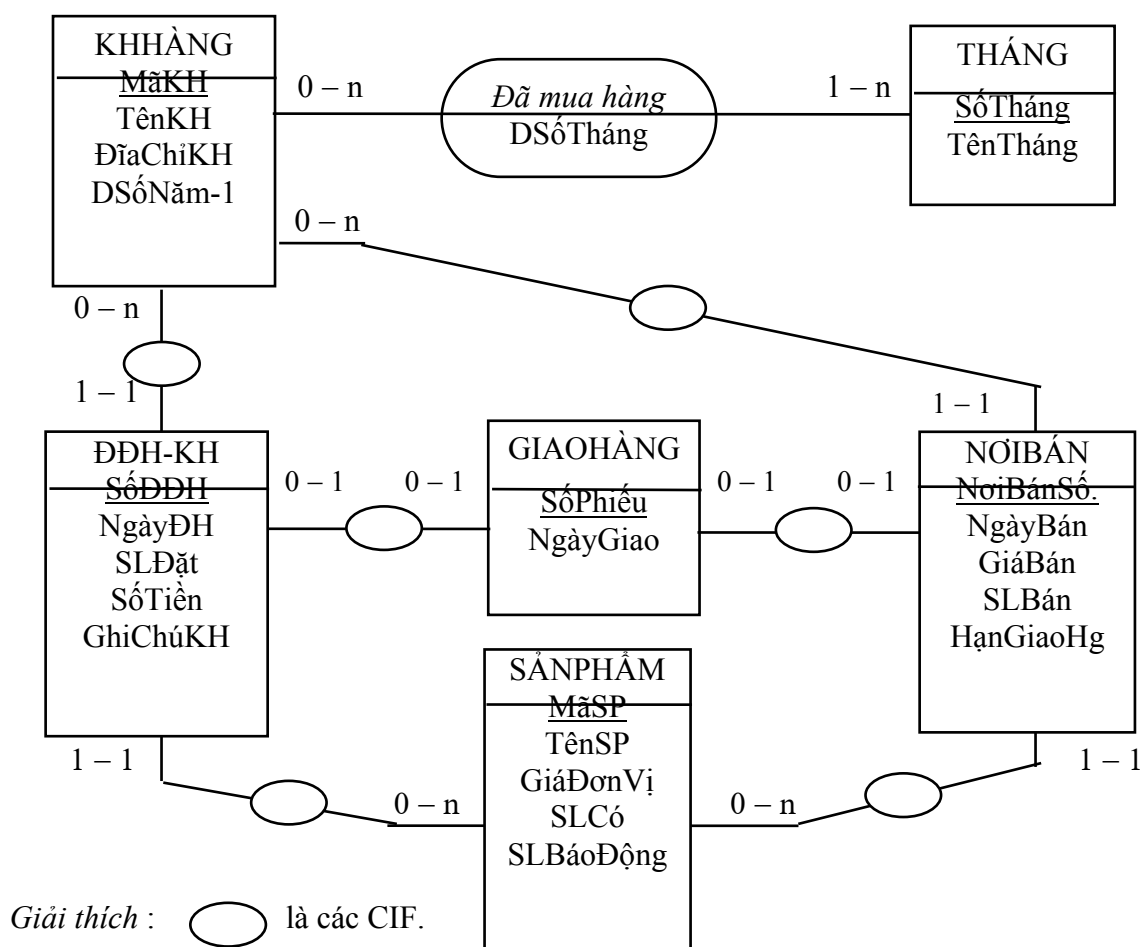
Ví dụ : MãKH + Tháng → DSốTháng được chuyển thành :



Hình 4.60 Kết hợp không phân cấp

Kết hợp “Đã mua hàng” là không phân cấp vì bản số cực đại của cả hai đều bằng n. Bản số 0-n (KHHÀNG) chỉ ra rằng một khách hàng có thể không có doanh số tháng, nếu khách hàng này chưa yêu cầu một đơn đặt hàng nào. Ngược lại, khách hàng này có thể có doanh số tháng tùy theo số tháng đã trải qua từ khi bắt đầu xuất hiện trong CSDL. Bản số 1-n (THÁNG) chỉ ra rằng với một tháng đã cho, có ít nhất một khách hàng đã yêu cầu một hoặc nhiều đơn đặt hàng.

Mô hình thực thể - kết hợp cho nhà máy đóng hộp DanaFood như sau :



Hình 4.61 Mô hình ý niệm dữ liệu của DanaFood

## Bài tập chương 4

### 1. Công ty bảo hiểm tài sản BAVICO

Công ty bảo hiểm tài sản BAVICO bảo hiểm các loại bất động sản như nhà, cửa hàng, garage, kho bãi... Để xây dựng HTTT, công ty cần những thông tin sau :

Tên, họ, địa chỉ và điện thoại của chủ sở hữu.

Tên, tình trạng và địa chỉ bất động sản của chủ sở hữu.

Ví dụ ông Trần Văn A ngụ tại ..., điện thoại... có một căn nhà cấp 2 tại ...

Yêu cầu :

- Xây dựng mô hình thực thể - kết hợp biểu diễn các thực thể, các thuộc tính, các khoá, các kết hợp và kiểu của kết hợp, các bản số tương ứng về sở hữu.
- Tương tự câu a) nhưng về quan hệ thuê bất động sản (vẽ trên cùng một mô hình). Ví dụ ông Trần Văn A ngụ tại ..., điện thoại... thuê (để hành nghề) cửa hàng ăn uống bình dân tại tại ...

### 2. Thi nấu ăn

Trong một cuộc thi nấu ăn của những người đầu bếp, ban Tổ chức cần những thông tin sau đây :

Món ăn dự thi :

- Tên món ăn (*bò lúc lắc, sôt vang thỏ, ốc hấp gừng, ...*)
- Kiểu (*khai vị, món chính, tráng miệng, ...*)

Thực phẩm dùng để làm món ăn :

- Loại (*thịt, cá, rau, bột, ...*)
- Tên thực phẩm (*gà công nghiệp, chim bồ câu, bò, ...*)
- Đặc điểm (*tươi, ướp lạnh, hun khói, ...*)

Tên, họ, địa chỉ, cửa hàng đang hành nghề ... của mỗi người đầu bếp dự thi.

Với mỗi món ăn, mỗi đầu bếp dự thi có cách chuẩn bị riêng. Ví dụ cùng làm món *Trứng Vương miện*, ông A dùng 6 quả trứng và 200 gram mỡ lợn, nhưng ông B lại chỉ dùng 5 quả trứng và 150 gram beure.

Chú ý : Mỗi món ăn phải nấu cho nhiều người ăn, không làm cho một người.

Mỗi đầu bếp không làm tất cả các món ăn dự thi.

Yêu cầu : Xây dựng mô hình thực thể - kết hợp cho HTTT vừa nêu.



## CHƯƠNG 5

### Mô hình hóa dữ liệu bằng phương pháp lùi

---

Trong chương trước, chúng ta đã nghiên cứu cách xây dựng các mô hình ý niệm dữ liệu dựa trên việc phân tích các phụ thuộc hàm, xuất phát từ những dữ liệu thô thu nhận được từ HTTT đang xét. Phương pháp mô hình hoá dữ liệu như vậy được gọi là *phương pháp tiến*, hay từ dưới lên (méthode ascendante).

Trong chương này, chúng ta sẽ nghiên cứu một phương pháp khác nhanh hơn, gọi là *phương pháp lùi*, hay từ trên xuống (méthode descendante), dựa trên việc phân tích các cấu trúc kiểu (structures types).

Trước hết, chúng ta sẽ giới thiệu các cấu trúc kiểu và sau đó, dùng các ví dụ minh hoạ, chúng ta chỉ ra cách xây dựng một mô hình dữ liệu nhờ các cấu trúc kiểu trong HTTT đang xét.

#### I. Các cấu trúc kiểu

Để xây dựng các mô hình dữ liệu, người ta đưa vào các cấu trúc kiểu đặc thù cho hệ thống thông tin mà nó biểu diễn. Các cấu trúc kiểu gồm :

Các cấu trúc kiểu cơ sở : PHIẾU, CHA-CON, BẢNG.

Các cấu trúc kiểu đặc biệt được tổ hợp từ các cấu trúc kiểu cơ sở, đó là :

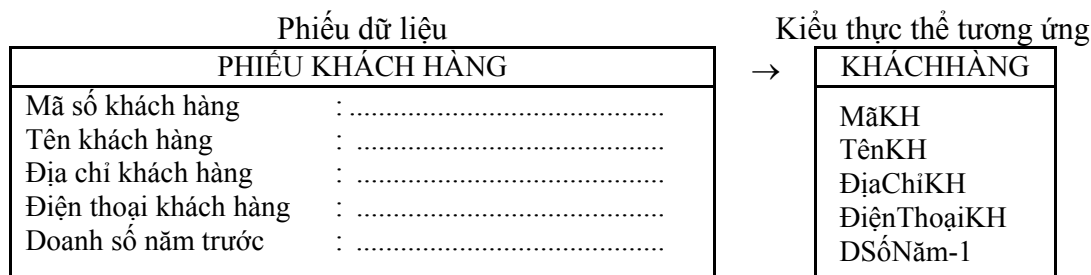
HOẠCH ĐỊNH, CÓ-KHÔNG, PHẢN XẠ, BÌNH ĐẲNG, THỪA KẾ và KẾT TỤ.

##### I.1. Cấu trúc kiểu PHIẾU

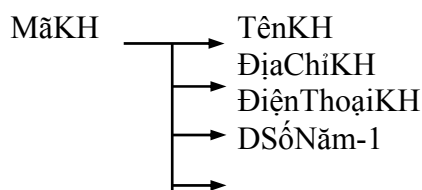
Cấu trúc kiểu PHIẾU (FICHE) cho phép mô tả tường minh một thực thể và các thành phần của thực thể như khóa, các thuộc tính và sự PTH giữa khóa và các thuộc tính của thực thể.

Từ một phiếu dữ liệu (thu lượm được từ bài toán đang phân tích), có thể dễ dàng chuyển thành kiểu thực thể tương ứng.

Ví dụ



Thực thể KHÁCH HÀNG nhận được từ phiếu khách hàng cho biết : mỗi khách hàng được đồng nhất bởi một MãKH, vì với một MãKH, chỉ có một và chỉ một khách hàng. Như vậy MãKH là khóa của thực thể KHÁCH HÀNG.



Những dữ liệu khác là các thuộc tính của thực thể KHÁCH HÀNG. Vì rằng tồn tại sự PTH giữa khóa MãKH và mỗi một thuộc tính này, sao cho MãKH là nguồn và các thuộc tính là đích. Như vậy, mỗi giá trị MãKH chỉ tương ứng với một giá trị của hoặc TênKH, hoặc ĐịaChỉKH, hoặc ĐiệnThoạiKH, hoặc DSốNăm-1.

## I.2.Cấu trúc kiểu CHA-CON

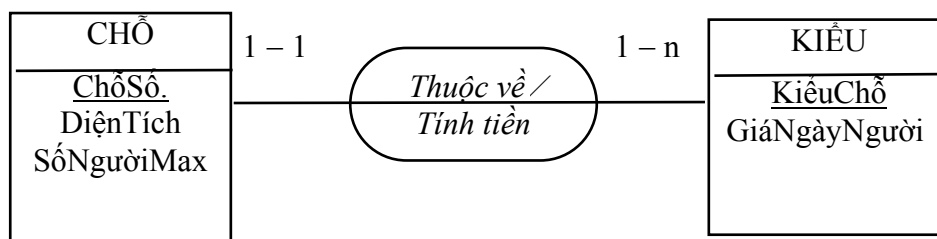
Cấu trúc kiểu CHA-CON (père-fils) liên quan đến sự kết hợp phân cấp (hay phụ thuộc toàn vẹn hàm – CIF) và sự PTH giữa khóa và nhiều thực thể.

Ví dụ

PHIẾU KIỂU CHỖ			
Kiểu chỗ	:	.....	
Giá/Ngày/Người	:	.....	
Danh sách các chỗ	:	.....	
Số TT	Chỗ số.	Diện tích	Số người tối đa
.....	.....	.....	.....
.....	.....	.....	.....

Phiếu trên tạo ra các thực thể CHỖ, KIỂU và kết hợp phân cấp *Thuộc về/Tính tiền*. Việc phân tích phiếu chỉ ra rằng, một số dữ liệu chỉ nhận một giá trị cho một phiếu, ở đây là cho một phiếu kiểu chỗ (“lều bạt”, “xe nhà ở”...). Đó là trường hợp của các dữ liệu KiểuChỗ và GiáNgàyNgười (giá cho một ngày và cho một người).

Trong khi đó, các dữ liệu khác có thể nhận nhiều giá trị, như là ChỗSố, DiệnTích và SốNgườiMax. Vì rằng có thể có nhiều chỗ cùng kiểu chỗ. Người ta nói trường hợp này có hai luồng thông tin. Do đó, không thể xem những dữ liệu vừa nêu là thuộc về một thực thể duy nhất, mà là thuộc về hai thực thể.



Hình 5.1 Mô hình dữ liệu cấu trúc kiểu CHA-CON

Như vậy, hai thực thể quan hệ với nhau bởi kết hợp *Thuộc về / Tính tiền* và các bản số có nghĩa rằng một chỗ chỉ thuộc về một và chỉ một kiểu chỗ, một kiểu chỗ tính toán cho tối thiểu một chỗ và có thể cho nhiều chỗ (n). Các bản số chỉ ra kiểu kết hợp phân cấp hay CIF.

Hai thực thể nối nhau bởi một CIF tạo thành cấu trúc kiểu CHA-CON nói lên rằng, một cha có thể có nhiều con, nhưng mỗi con chỉ có một cha. Các thực thể CHỖ và KIỂU nói riêng là những cấu trúc kiểu PHIẾU. Mặt khác, trong cấu trúc kiểu CHA-CON, tồn tại sự PTH giữa hai khóa : ChỗSố → KiểuChỗ.

### 1.3. Cấu trúc kiểu BẢNG

Ví dụ :

Mỗi khách hàng được ghi chép vào một phiếu gồm mã (do công ty tự ấn định), tên họ, địa chỉ, điện thoại và doanh số hàng tháng như sau :

PHIẾU KHÁCH HÀNG		
Mã khách hàng	:	.....
Tên khách hàng	:	.....
Địa chỉ	:	.....
Điện thoại	:	.....
Doanh số năm trước	:	.....
<i>Doanh số hàng tháng :</i>		
Giêng : .....	Năm : .....	Chín : .....
Hai : .....	Sáu : .....	Mười : .....
Ba : .....	Bảy : .....	Mười Một : .....
Tư : .....	Tám : .....	Mười Hai : .....

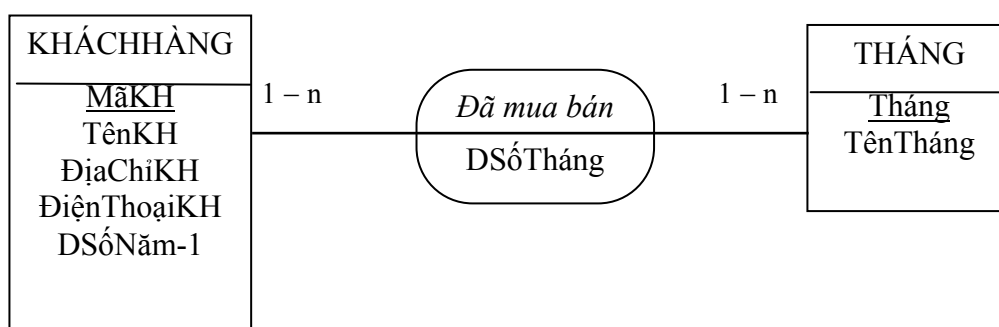
Cấu trúc kiểu BẢNG (TABLEAU) liên quan đến sự kết hợp không phân cấp hay CIM (phụ thuộc toàn vẹn bộ), sự ghép các khóa và các PTH giữa các khóa của ít nhất hai thực thể và các thuộc tính của kết hợp không phân cấp.

Với mỗi khách hàng, có bao nhiêu tháng giao dịch trong năm thì sẽ có bấy nhiêu doanh số tháng. Mặt khác, lấy một tháng nào đó, có bao nhiêu khách hàng trong CSDL thì sẽ có bấy nhiêu doanh số tương ứng. Từ đó có bảng hai chiều như sau :

Khách hàng	Tháng			
	<i>Giêng</i>	<i>Hai</i>	<i>Ba</i>	...
Khách hàng 1024	12 543 000	8 654 000	19 231 000	
Khách hàng 1134	5 467 000	6 785 000	11 942 000	
Khách hàng 1145	15 687 000	13 232 000	20 980 000	
.....				

Trong bảng trên, dữ liệu DSốTháng phụ thuộc vào hai khóa là MãKH và Tháng. Có nghĩa giá trị của một doanh số tháng liên quan đến một khách hàng và tại một tháng đã cho.

Phiếu khách hàng được chuyển thành hai thực thể KHÁCHHÀNG và THÁNG. Tồn tại một kết hợp không phân cấp giữa hai thực thể này :



Hình 5.2 Mô hình dữ liệu cấu trúc kiểu BẢNG

Ta có một PTH với đích là DSốTháng và nguồn là ghép của các khóa của các thực thể tham gia vào kết hợp “Đã mua bán” :

$$\text{Mã-KH} + \text{Tháng} \rightarrow \text{DSốTháng}$$

Trong thí dụ trên, ta tự giới hạn ở bảng hai chiều do các kết hợp phân cấp chỉ liên quan đến hai thực thể. Trong khi đó, các kết hợp không phân cấp lại có thể liên quan đến nhiều thực thể. Với ba thực thể, ta có bảng ba chiều. Với bốn thực thể, ta có bảng bốn chiều, v.v...

Tập hợp từ 2 đến n thực thể có quan hệ với nhau bởi một kết hợp không phân cấp được gọi là cấu trúc kiểu BẢNG.

## I.4.Cấu trúc kiểu HOẠCH ĐỊNH

Cấu trúc kiểu HOẠCH ĐỊNH (planning) là trường hợp đặc biệt của cấu trúc kiểu BẢNG. Một trong những thực thể bắt buộc hiện diện là thời gian. Trong thí dụ về cấu trúc kiểu BẢNG ở trên, có một thực thể thể hiện thời gian là tháng. Tuy nhiên đó không phải là một hoạch định. Ta sẽ thấy rằng có sự phân biệt giữa cấu trúc HOẠCH ĐỊNH và các cấu trúc kiểu khác.

Trong cấu trúc kiểu HOẠCH ĐỊNH, có thực thể thể hiện THỜI GIAN. Người ta thường quan tâm đến chu kỳ thời gian là ngày.

Để quản lý việc thuê phòng trong một khách sạn, người ta cần biết hiện trạng của các phòng nhờ MãHiệnTrg có thể nhận các giá trị như sau :

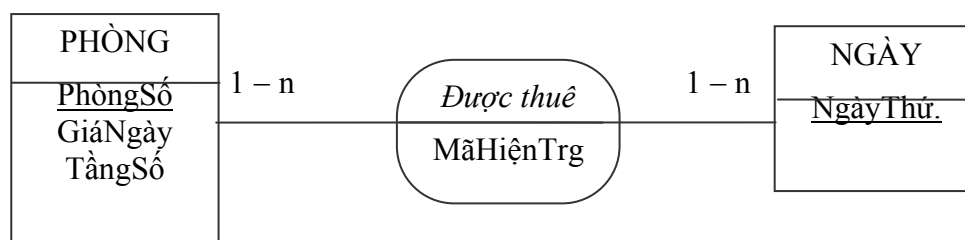
T : phòng trống                      B : có người ở (phòng bận)                      Đ : phòng đã đặt trước

Trong bảng dưới đây, ta thấy MãHiệnTrg phụ thuộc vào hai dữ liệu là NgàyThứ và PhòngSố.

Phòng số	Ngày										
	1	2	3	4	5	6	7	8	9	10	...
	T	T	T	T	T	Đ	Đ	Đ	T	T	
	B	B	B	B	T	T	T	Đ	Đ	Đ	
	B	B	B	B	T	T	T	T	T	T	
	T	T	T	T	T	T	Đ	Đ	Đ	Đ	
	T	T	T	T	Đ	Đ	Đ	Đ	Đ	T	
	B	B	B	B	Đ	Đ	Đ	Đ	Đ	Đ	

Ngày hôm nay : 05/07/199 .....

Các thực thể PHÒNG và NGÀY nối với nhau bởi kết hợp không phân cấp “Được thuê”. Dữ liệu MãHiệnTrg là thuộc tính của kết hợp này.



Hình 5.3 Mô hình dữ liệu cấu trúc kiểu HOẠCH ĐỊNH

Tồn tại PTH với đích là MãHiệnTrg, nguồn là do hai khóa PhòngSố và NgàyThứ ghép lại :  
 PhòngSố + NgàyThứ → MãHiệnTrg

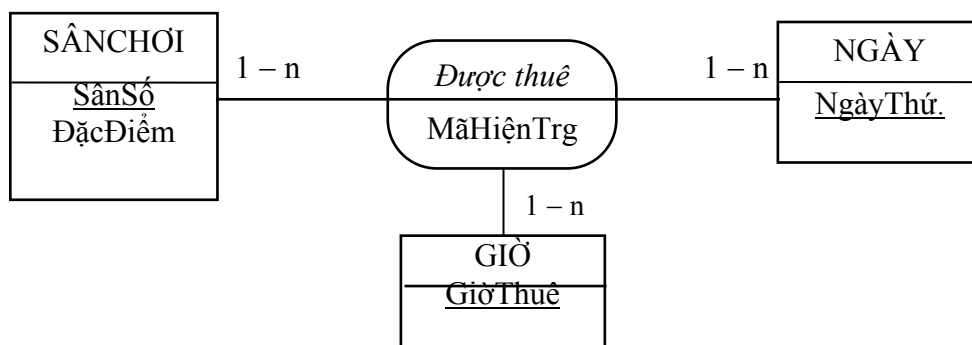
Cấu trúc kiểu HOẠCH ĐỊNH có thể được tạo ra từ nhiều bảng (hai chiều, ba chiều hoặc nhiều chiều) tương ứng, tương tự cấu trúc kiểu BẢNG. Nếu cần quản lý chi tiết hơn về thời gian hoạt động, có thể sử dụng thêm những thực thể khác.

Chẳng hạn, nếu người ta muốn quản lý việc dành chỗ sân quần vợt vào những giờ giấc nào đó, cần phải thêm vào thực thể GIỜ và ta có mô hình dữ liệu sau :

Như vậy, với mỗi sân, mỗi ngày và mỗi giờ, ta có một giá trị của thuộc tính MãHiệnTrg :

Sân số + NgàyThứ + GiờThuê → MãHiệnTrg

Chú ý : Vậy khi nào thì phải tạo ra một thực thể thời gian ?



Hình 5.4 Cấu trúc kiểu HOẠCHĐỊNH nhiều chiều

Nếu như việc quản lý chỉ liên quan đến những ngày đặc biệt, chẳng hạn NgàyĐH (ngày đặt hàng) hoặc NgàyTToán (ngày thanh toán), thì không nên tạo thực thể thời gian, mà tạo ra các dữ liệu tương ứng, sau đó đặt chúng vào các thực thể liên quan với vai trò là những thuộc tính. Ví dụ, NgàyĐH đưa vào thực thể ĐẶTHÀNG và NgàyTToán vào HÓAĐƠN. Khi đó, ta có thể sử dụng các PTH.

Tuy nhiên, khi thời gian là nguồn của một PTH, với vai trò là dữ liệu sơ cấp, hoặc thời gian là thành phần của một nguồn gồm nhiều thành phần ghép lại, thì phải tạo ra một thực thể thời gian. Chính vì vậy mà tên cấu trúc kiểu là HOẠCHĐỊNH.

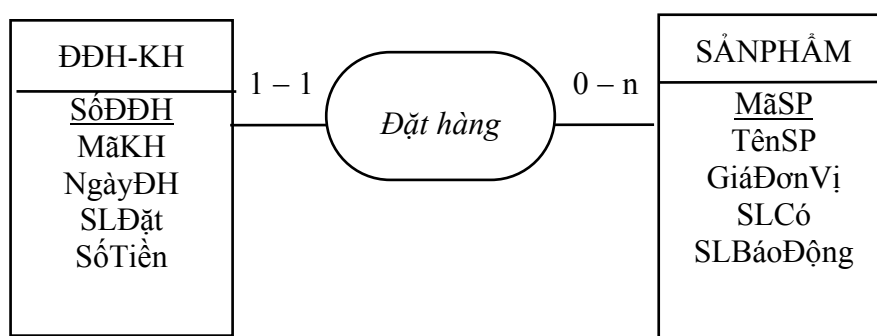
## I.5.Cấu trúc kiểu CÓ-KHÔNG

Được tạo thành từ các cấu trúc kiểu cơ sở BẢNG và CHA-CON.

Cấu trúc kiểu CÓ-KHÔNG (oui-non) dựa trên ý tưởng rằng một trường hợp cụ thể của một thực thể này ít khi tương ứng với một trường hợp cụ thể của một thực thể kia, mặc dầu giữa chúng có liên hệ với nhau bởi một kết hợp.

Người ta đưa vào khái niệm *trường hợp cụ thể tùy ý* của thực thể và sử dụng khái niệm ràng buộc toàn vẹn trong cấu trúc kiểu này.

Ví dụ



Hình 5.5 Mô hình dữ liệu cấu trúc kiểu CÓ-KHÔNG

Ở đây, sử dụng cấu trúc kiểu CHA-CON. Các bản số của ĐDH-KH chỉ ra rằng, một đơn đặt hàng của khách hàng chỉ liên quan đến một và chỉ một sản phẩm. Bản số tối thiểu 1 có nghĩa tính ràng buộc toàn vẹn phải được kiểm tra : một trường hợp cụ thể của ĐDH-KH phải tương ứng với một trường hợp cụ thể của SẢN PHẨM.

Các bản số của SẢN PHẨM chỉ ra rằng, một sản phẩm nào đó có thể không được đặt hàng, nhưng cũng có thể được đặt hàng nhiều lần. Bản số tối thiểu 0 chỉ ra rằng ở đây không có ràng buộc toàn vẹn. Đây là đặc trưng của cấu trúc kiểu CÓ-KHÔNG : một trường hợp cụ thể của

một thực thể này có thể không tương ứng, hoặc tương ứng với một hoặc nhiều trường hợp cụ thể khác. Các bản số có thể là 0–1, hoặc 0–n.

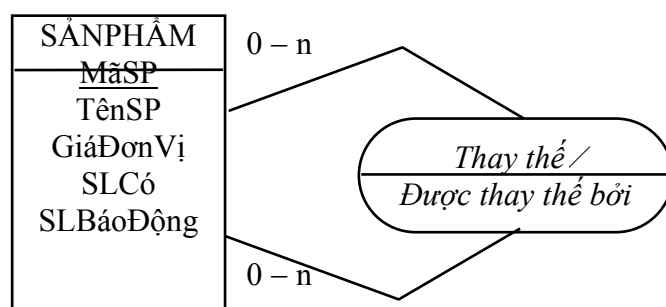
Tuỳ theo quy tắc quản lý mà cấu trúc kiểu này tồn tại, tương ứng với bản số tối thiểu có giá trị 1, hoặc không tồn tại, bản số tối thiểu có giá trị 0. Cần phải hiểu được điều muốn diễn tả. Trong thí dụ trên, các sản phẩm có thể được nhiều đơn đặt hàng đến nhưng cũng có thể được tiêu thụ ở nơi bán, điều đó giải thích giá trị 0 của bản số tối thiểu.

## I.6.Cấu trúc kiểu PHẢNXẠ

Cấu trúc kiểu PHẢNXẠ (reflex) có thể được kết hợp với các cấu trúc kiểu khác. Một số kết hợp không tạo ra mối liên hệ giữa hai (hay nhiều) thực thể. Tuy nhiên kết hợp có thể tồn tại trên cùng một thực thể. Khi đó kết hợp có tính phản xạ.

Có hai cấu trúc kiểu trong ví dụ này là cấu trúc kiểu BẢNG (tập hợp đặc biệt vì rằng chỉ có duy nhất một thực thể, trong khi đó kết hợp là không phân cấp) và cấu trúc kiểu CÓ-KHÔNG.

Mô hình này xác định một ma trận thay thế giữa các sản phẩm. Một sản phẩm có thể không dùng để thay thế hoặc thay thế cho nhiều sản phẩm khác. Trái lại, một sản phẩm có thể không bị thay thế hoặc bị thay thế bởi nhiều sản phẩm khác.

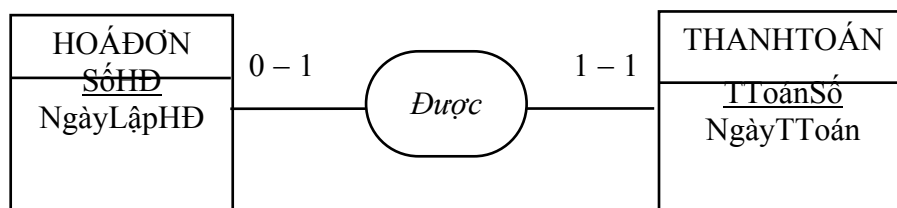


Hình 5.6 Mô hình dữ liệu cấu trúc kiểu PHẢNXẠ

## I.7.Cấu trúc kiểu BÌNHĐẲNG

Cấu trúc kiểu BÌNHĐẲNG (ÉGALITÉ) đặt một kết hợp giữa hai cấu trúc kiểu PHIẾU. Đây là một kết hợp đặc biệt mà người ta gọi là CIF qua lại (thuận nghịch) do bản số tối đa ở hai đầu đều bằng 1. Người ta cũng nói đây là trường hợp một cha có một con duy nhất.

Ví dụ : Ta thừa nhận rằng một hóa đơn chỉ có thể thanh toán một lần và một lần thanh toán chỉ căn cứ trên một hóa đơn.



Hình 5.7 Mô hình dữ liệu cấu trúc kiểu BÌNHĐẲNG

Mỗi thực thể có các đặc tính riêng khẳng định sự tồn tại của thực thể đó : khóa và các thuộc tính. Mỗi trường hợp cụ thể của mỗi một trong hai thực thể tương ứng với nhiều nhất là một trường hợp cụ thể của thực thể kia. Ví dụ trên cũng thể hiện cấu trúc kiểu CÓ-KHÔNG. Mô hình được giải thích như sau :

Một hóa đơn chỉ được thanh toán nhiều nhất một lần. Nhưng chừng nào thanh toán này chưa được thực hiện, sẽ không có trường hợp cụ thể của THANHTOÁN tương ứng với trường hợp cụ thể của HÓAĐƠN đang xét. Từ đó ta có bản số 0 (cấu trúc kiểu CÓ-KHÔNG).

Một thanh toán, một khi đã được ghi nhận, sẽ tồn tại trong CSDL và có quan hệ đến một và chỉ một hóa đơn.

Chú ý rằng do chỉ cần làm việc với một thực thể duy nhất, việc mô hình hóa đặt ra những vấn đề sau :

HÓAĐƠN
<u>SỐHĐ</u>
NgàyLậpHĐ
TToánSố
NgàyTToán

Các thuộc tính TToánSố và NgàyTToán có giá trị 0 khi việc thanh toán cho hóa đơn tương ứng chưa được thực hiện.

Không thể thanh toán nếu chưa biết số hóa đơn liên quan.

NgàyTToán là đích của một PTH mà TToánSố là nguồn, và cũng là đích của một PTH khác mà SỐHĐ là nguồn. PTH sau cùng này có tính bắc cầu.

TToánSố → NgàyTToán

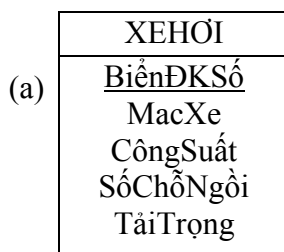
SỐHĐ → NgàyTToán và ta cũng có SỐHĐ → TToánSố

## I.8.Cấu trúc kiểu THỪA KẾ

Việc đưa vào khái niệm thừa kế (héritage) trong các mô hình dữ liệu thực thể - kết hợp thực sự đã làm phong phú thêm về mặt ngữ nghĩa. Khái niệm cấu trúc kiểu THỪAKẾ liên quan đến sự chuyên môn hóa và sự khái quát hoá.

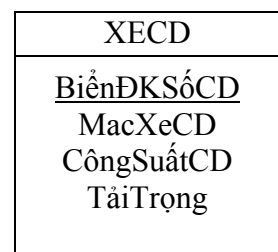
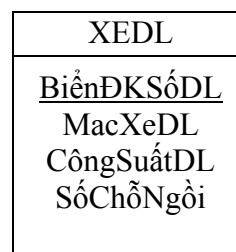
Ví dụ : Trong một mô hình dữ liệu, ta quan tâm đến nhiều loại xe : loại xe du lịch (DL), loại xe chuyên dùng (CD). Thoạt tiên, ta dự kiến hai mô hình hóa :

Mô hình dùng một thực thể duy nhất :

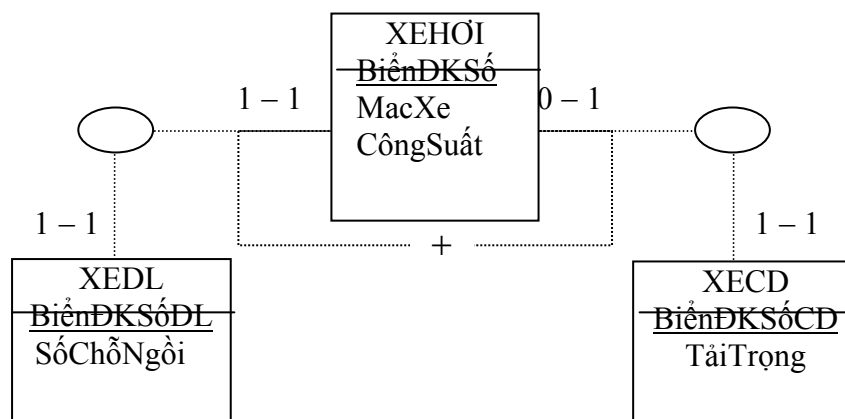


(b)

Mô hình dùng hai thực thể phân biệt :



Cả hai mô hình trên đều không thỏa mãn. Trong mô hình dùng một thực thể duy nhất (a), dữ liệu SốChỗNgồi chỉ có nghĩa với loại xe du lịch và dữ liệu TảiTrọng chỉ có nghĩa với các xe chuyên dùng. Còn trong mô hình dùng hai thực thể phân biệt (b), các thuộc tính chung cho cả hai loại xe (MacXe và CôngSuất) lại nằm trong cả hai thực thể.



Hình 5.8 Mô hình dữ liệu cấu trúc kiểu THỪAKẾ



Để khắc phục, ta đưa các dữ liệu chung cho cả hai loại xe vào trong một thực thể XEHƠI và các dữ liệu đặc trưng của mỗi loại xe được đặt vào các thực thể XEDL và XECD. Ta có các PTH như sau :

Mô hình (a) thể hiện sự chuyên môn hóa : các dữ liệu đặc trưng cho mỗi loại xe được đặt trong thực thể tương ứng).

Mô hình (b) thể hiện khái quát hóa : nhóm các dữ liệu chung cho cả hai loại xe vào trong cùng một thực thể đặc chủng.

Vậy thì tính thừa kế là gì trong mô hình dữ liệu này ?

Hai thực thể XEDL và XECD thừa kế các thuộc tính MacXe và CôngSuất của thực thể đặc chủng XEHƠI.

Chúng là các cấu trúc kiểu BÌNHĐẲNG và CÓ-KHÔNG, tạo thành cấu trúc kiểu THỪAKẾ. Một trường hợp cụ thể của XEDL phải tương ứng với một trường hợp cụ thể của XEHƠI (ràng buộc toàn vẹn). Tương tự như vậy đối với thực thể XECD.

Với thực thể XEHƠI, vấn đề phức tạp hơn. Một trường hợp cụ thể của XEHƠI không nhất thiết tương ứng với một trường hợp cụ thể của XEDL (cấu trúc kiểu CÓ-KHÔNG). Cũng lý luận tương tự cho thực thể XECD. Tuy nhiên, cần có một trường hợp cụ thể của XEHƠI tương ứng với một trường hợp cụ thể của XEDL, hoặc của XECD.

Thực tế, một xe nào đó hoặc là loại xe du lịch, hoặc là loại xe chuyên dụng (trong thí dụ này). Điều này thể hiện bởi dấu + (hay “hoặc có loại trừ”). Đây là một ràng buộc có loại trừ lẫn nhau giữa hai kết hợp cho phép nối kết XEHƠI với XEDL và XEHƠI với XECD.

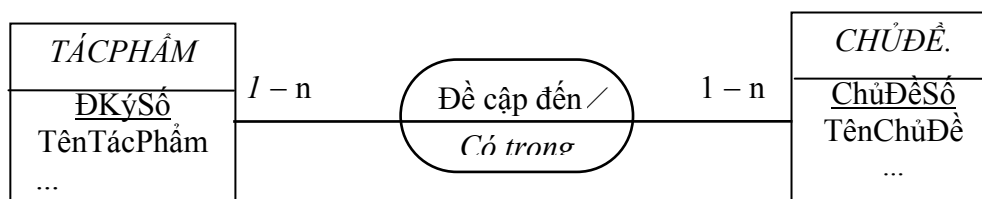
## I.9.Cấu trúc kiểu KẾT TỤ

Cấu trúc kiểu cuối cùng này chỉ ra rằng đôi khi cần phải mô hình hóa một hệ thống thông tin bằng cách thực hiện từng giai đoạn.

Ở đây đề cập đến khái niệm kết tụ (AGRÉGATION) và cũng đề cập đến khái niệm cá thể hóa để giải quyết cùng một vấn đề theo một cách khác.

Ví dụ : Xét một thư viện phục vụ sinh viên Tin học. Các tác phẩm (sách, tài liệu...) thuộc nhiều chủ đề khác nhau (ngôn ngữ lập trình, hệ điều hành, cơ sở dữ liệu, ...). Đương nhiên, một chủ đề có thể có nhiều tác phẩm.

Ta có mô hình dữ liệu kiểu BẢNG thiết lập sự tương ứng giữa các tác phẩm và các chủ đề như sau :



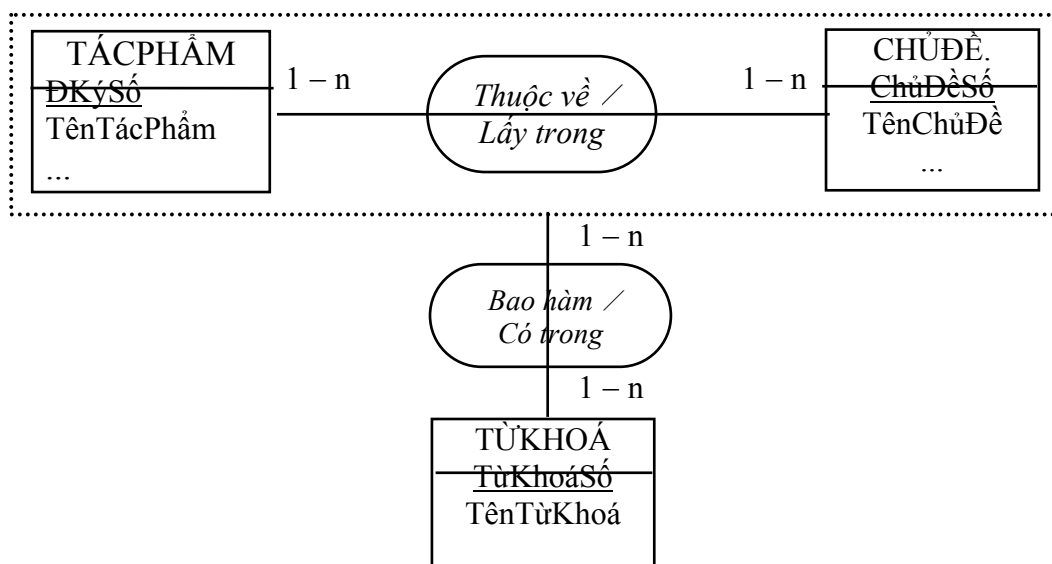
Hình 5.9 Mô hình dữ liệu cấu trúc kiểu BẢNG

Để quản lý thư viện dễ dàng và hiệu quả, sinh viên đã đưa ra sáng kiến thiết lập một danh mục các từ khóa có ý nghĩa nhất về tình trạng kiến thức Tin học. Họ muốn rằng có thể tìm được những từ khóa ở trong mỗi phần của tác phẩm đã được đặt mã tham chiếu, nghĩa là cho mỗi chủ đề trong mỗi tác phẩm.

Chẳng hạn, hãy tìm những từ khóa về đề tài “Mô hình quan hệ”, mã MHQH, có mặt trong tác phẩm có tên “Cơ sở dữ liệu”, mã tác phẩm là BD143. Ta thêm thực thể mới TỪKHÓA và nối kết với mô hình trên.

Không thể nối trực tiếp TỪ KHOÁ với TÁC PHẨM được, vì như vậy, khái niệm về chủ đề sẽ bị loại trừ. Ta cũng không thể nối với thực thể CHỦ ĐỀ, vì khái niệm về tác phẩm sẽ bị loại ra. Ở đây, cần phải nối TỪ KHOÁ với toàn bộ mô hình, hoặc bởi kết tụ, hoặc bởi việc cá thể hóa kết hợp “Bao hàm/Có trong”.

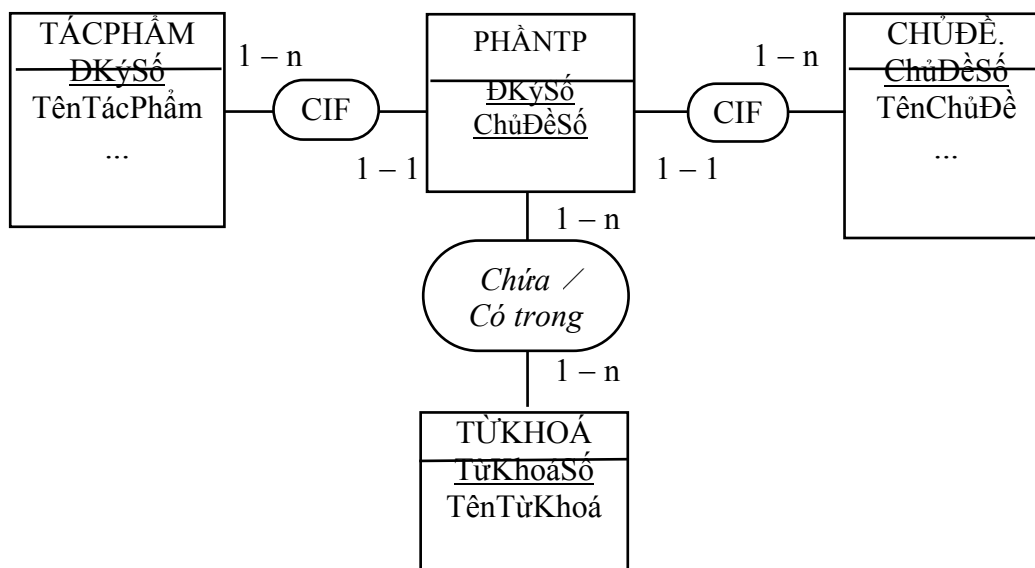
**a) Kết tụ**



Hình 5.10 Mô hình dữ liệu cấu trúc kiểu KẾT TỤ

Nếu coi một phần tác phẩm là một chủ đề được đề cập đến trong tác phẩm, thì mô hình trên chỉ ra rằng, với mỗi phần của tác phẩm, tìm thấy từ 1 đến n từ khóa và ngược lại, một từ khóa có thể được tìm thấy trong 1 hoặc n phần của các tác phẩm khác nhau. Ở đây, mô hình không thể được thiết lập nếu như không trải qua giai đoạn kết tụ.

**b) Cá thể hóa**



Hình 5.11 Mô hình dữ liệu cấu trúc kiểu cá thể hoá

Cá thể hóa một kết hợp không phân cấp là thay thế kết hợp này bởi một thực thể và bởi một số kết hợp phân cấp bằng số các thực thể tham gia vào kết hợp không phân cấp này. Như vậy, ta đã chuyển về mô hình nhị nguyên (mô hình hai chiều, mọi kết hợp là phân cấp). Tiếp theo, có thể đặt một kết hợp phân cấp hoặc không giữa thực thể mới này với một thực thể khác.

Kết quả của cá thể hóa là cho ra một mô hình giống mô hình trên.

## II. Ứng dụng phương pháp từ trên xuống

Trong mục này, ta sẽ đưa vào một ví dụ để minh họa phương pháp từ trên xuống, thấy được sự gắn gũi của phương pháp với những cấu trúc thu lượm được từ HTTT.

Vận dụng các cấu trúc kiểu để xây dựng một mô hình dữ liệu, phương pháp này thực sự nhanh chóng và ít trừu tượng hơn phương pháp từ dưới lên đã xét.

### II.1. Giới thiệu công ty xây dựng nhà ở BKCO

#### II.1.1. Các quy tắc quản lý

Công ty xây dựng nhà ở BKCO, có nhiệm vụ xây dựng các căn hộ cho khách hàng. Sau khi phỏng vấn những người có trách nhiệm của BKCO, phân tích viên (PTV) thu nhận được những thông tin về quy tắc quản lý như sau :

1. BKCO có hơn 10 kiểu nhà ở (mẫu thiết kế). Việc xây dựng một kiểu nhà bao gồm một số giai đoạn liên tiếp nhau. Mỗi giai đoạn được đặc trưng bởi tên gọi và thời gian thi công.
2. Mỗi công trình có một người phụ trách. Tuy nhiên, tùy tình hình thực tế mà một người có thể phụ trách nhiều hoặc không phụ trách công trình nào.
3. BKCO chỉ quản lý thông tin đối với những khách hàng đang có công trình thi công (đến 6 tháng sau khi giao chìa khoá nhà). BKCO chỉ thi công một công trình cho mỗi khách hàng.
4. Với mỗi mẫu nhà thi công, trước khi tiến hành một giai đoạn xây dựng, khách hàng phải thanh toán trước một số tiền.
5. Vào ngày thoả thuận, khách hàng thanh toán trước số tiền và giai đoạn tương ứng được bắt đầu. Thời hạn thi công được đảm bảo nếu khách hàng thanh toán đúng kỳ hạn và không xảy ra sự cố (vì lý do thời tiết, cấp vật liệu trễ, v.v...).

#### II.1.2. Hồ sơ

PTV thu nhận được ba loại hồ sơ như sau :

##### a) *Phiếu khách hàng*

Mỗi phiếu khách hàng bao gồm :

- Số của khách hàng
- Tên khách hàng
- Địa chỉ của khách hàng
- Số tài khoản
- Tên và địa chỉ của ngân hàng
- Mã số kiểu nhà được chọn
- Số của công trình tương ứng
- Ngày cam kết trả tiền trước

##### b) *Phiếu kiểu nhà thi công*

Mỗi phiếu kiểu nhà bao gồm :

- Mã số kiểu nhà
- Tên kiểu nhà



**e) Theo dõi công trình**

Để theo dõi công trình, người ta dùng một phiếu ghi tổng hợp bắt đầu từ giai đoạn 1 cho đến khi bàn giao chìa khoá cho khách hàng.

BKCO ✉ ..., ☎ ...		
PHIẾU THEO DÕI CÔNG TRÌNH		
Mã số kiểu nhà : .....		Họ tên người phụ trách : .....
Ngày bàn giao chìa khoá : .....		
<i>Số TT giai đoạn</i>	<i>Ngày bắt đầu dự kiến</i>	<i>Ngày kết thúc thực tế.</i>
1		
2		
3		
4		
5		

**II.1.3. Nghiên cứu các cấu trúc kiểu**

Mỗi loại phiếu công tác có thể tạo thành một hoặc nhiều cấu trúc kiểu phục vụ xây dựng một mô hình dữ liệu biểu diễn HTTT của BKCO.

**a) Phiếu KHÁCH HÀNG**

Dữ liệu KHSố là khoá của thực thể KHÁCH HÀNG vì nó cho phép chỉ định một khách hàng duy nhất.

Các dữ liệu khác đều là thuộc tính của thực thể KHÁCH HÀNG vì có các phụ thuộc hàm (PTH) giữa khoá KHSố (nguồn) và mỗi một thuộc tính (đích) của nó. Ta nhận được cấu trúc kiểu PHIẾU là KHÁCH HÀNG.

Như vậy, mỗi giá trị của khoá KHSố tương ứng với một và chỉ một giá trị thuộc tính khác của thực thể KHÁCH HÀNG.

KHÁCH HÀNG	KIỂU NHÀ
<u>KHSố</u>	<u>KiểuNhàSố</u>
TênKH	TênKiểu
ĐịaChiKH	GiáKiểu
SốTàiKhoản	DiệnTích
TênNgânHàng	SốPhòng
KiểuNhàSố	
CôngTrìnhSố	
NgàyCamKết	

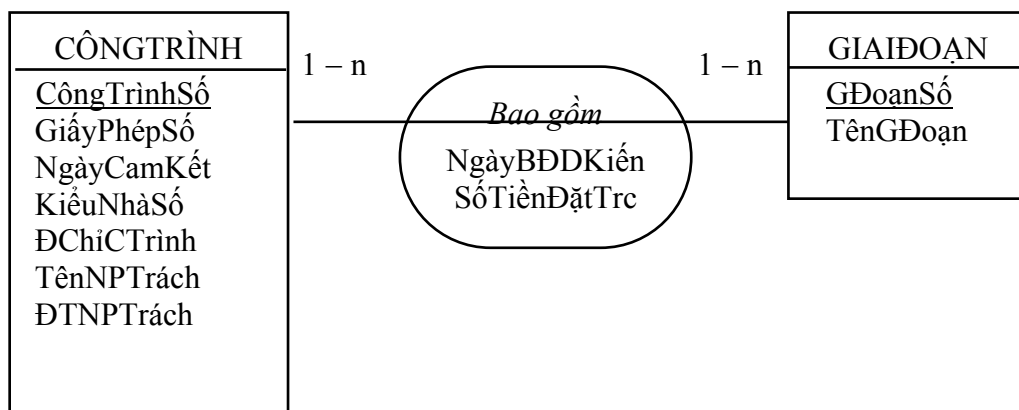
**b) Phiếu KIỂU NHÀ**

Đây cũng là một cấu trúc kiểu PHIẾU. KiểuNhàSố là khoá của thực thể KIỂU NHÀ vì nó cho phép chỉ định một kiểu nhà duy nhất.

Các dữ liệu khác đều là thuộc tính của thực thể này vì có các PTH giữa khoá KiểuNhàSố (nguồn) và mỗi một thuộc tính (đích) của nó. Mỗi giá trị của KiểuNhàSố tương ứng với một và chỉ một giá trị của TênKiểu, một và chỉ một giá trị của GiáKiểu, một và chỉ một giá trị của DiệnTích và một và chỉ một giá trị của SốPhòng.

**c) Chương trình thi công**

Một số dữ liệu gắn liền với công trình (phần đầu phiếu), còn những dữ liệu khác thì liên quan đến các giai đoạn thi công khác nhau. Các giai đoạn không liên quan đến công trình, nhưng liên quan đến các kiểu nhà là đối tượng của các công trình (phần bảng phía dưới phiếu)



Từ phiếu thi công, ta nhận được cấu trúc kiểu BẢNG.

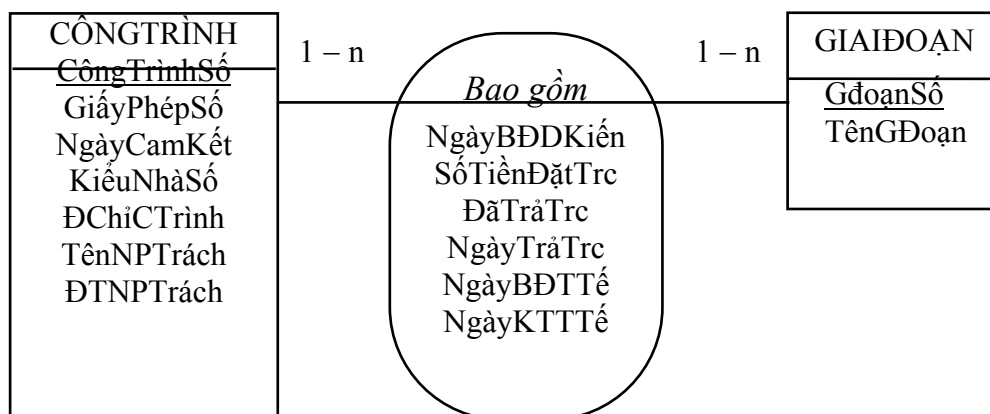
Tại BKCO, một công trình được thi công trong nhiều giai đoạn, còn một giai đoạn có thể liên quan đến nhiều công trình.

#### d) Tiến độ các công trình

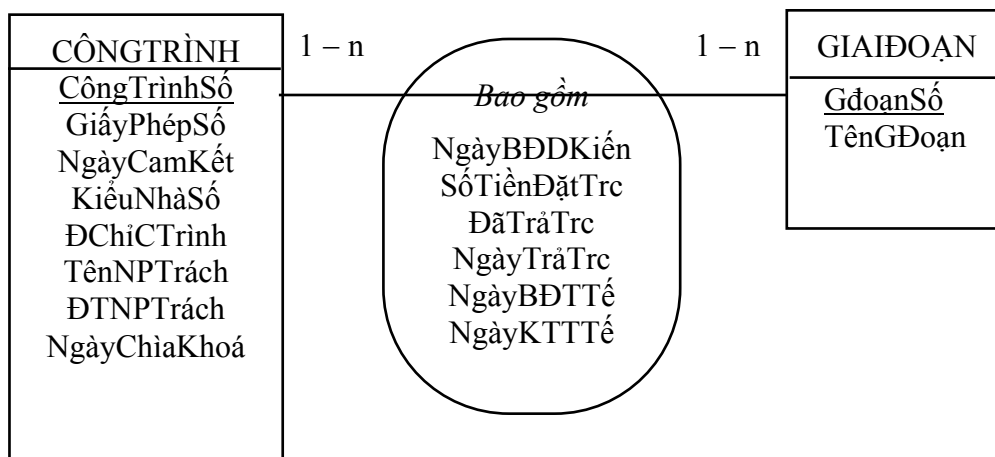
Cấu trúc kiểu cho tiến độ các công trình tương tự cấu trúc kiểu chương trình thi công đã trình bày ở trên.

Tuy nhiên cần thêm một số thuộc tính vào kết hợp “Bao gồm”. Đó là các dữ liệu cho biết khách hàng đã trả trước chưa (ĐãTrảTrc), ngày trả trước (NgàyTrảTrc), ngày bắt đầu thực tế (NgàyBĐTTế) và ngày kết thúc thực tế (NgàyKTTTế).

Ta có mô hình đầy đủ như sau :



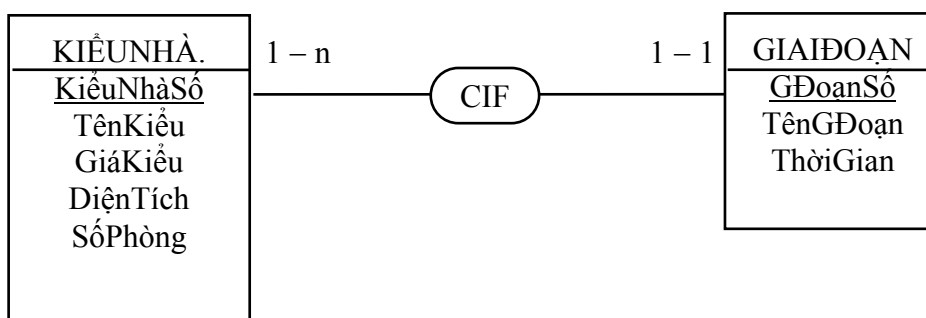
Phiếu theo dõi công trình có cùng kiểu cấu trúc dữ liệu với hồ sơ tiến độ các công trình. Tuy nhiên, thực thể CÔNG TRÌNH được thêm một dữ liệu thuộc tính mới là NgàyChiaKhoá (ngày giao chia khoá). Mỗi công trình có một giá trị dữ liệu này. Ta có mô hình mới đầy đủ như sau :



Ta xét lại nội dung các quy tắc quản lý đã phỏng vấn từ a) đến d) trong mục II.1.1. Riêng nội dung e) không có gì đặc biệt để xem xét.

Đây là cấu trúc kiểu CHA-CON giữa các thực thể KIỂUNHÀ (cha) và GIAĐOẠN (các con).

Trong mô hình này, khoá của thực thể GIAIĐOẠN đặt ra vấn đề. Trong các phiếu đang xét, giả sử các giai đoạn có số thứ tự đi từ 1 đến 8 (giá trị cuối cùng có vai trò quan trọng). Điều này không có nghĩa phải chú trọng vào từng giai đoạn của mỗi kiểu nhà.



Tuy nhiên, trong mô hình dữ liệu đang xây dựng, cần chú ý rằng giá trị khoá của mỗi thực thể phải là duy nhất trong CSDL. Đó không phải là trường hợp của GĐoạnSố vì rằng có bao nhiêu giá trị 1, 2, ... thì có bấy nhiêu kiểu nhà khác nhau trong CSDL.

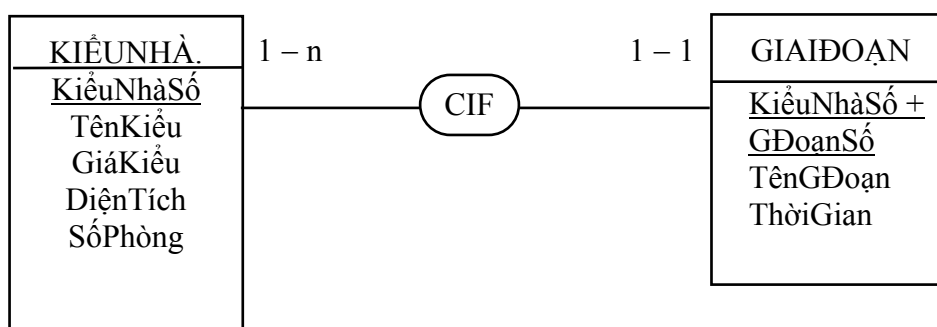
Do đó, nhìn từ các phiếu thực tế, chọn GĐoạnSố là khoá tự nhiên của thực thể GIAĐOẠN là hoàn toàn không phù hợp. Để chọn một khoá khác phù hợp hơn, cần trả lời câu hỏi sau đây : cần những dữ liệu nào hoặc cần ghép những dữ liệu nào để định danh một cách chắc chắn một giai đoạn trong CSDL ?

125

Câu trả lời thứ nhất có tính lý thuyết, dễ hiểu nhưng ít khi thực hiện được : ta thấy rằng GĐoạnSố là duy nhất trong CSDL, dẫn đến có thể có các giai đoạn từ 1 đến 8 cho kiểu nhà thứ nhất, từ 9 đến 18 cho kiểu nhà thứ hai, v.v...

Câu trả lời thứ hai có tính thực dụng vì cho rằng để định danh một giai đoạn, cần biết số thứ tự của nó và số của kiểu nhà tương ứng. Như vậy, khoá của thực thể GIAIDOẠN sẽ là KiểuNhàSố + GĐoạnSố. Đây là lời giải phù hợp vì tính thực hiện được của nó. Các giai đoạn và tên gọi tương ứng trong chương trình thi công hoàn toàn đặc trưng cho mỗi kiểu nhà xây dựng.

Mô hình bây giờ như sau :



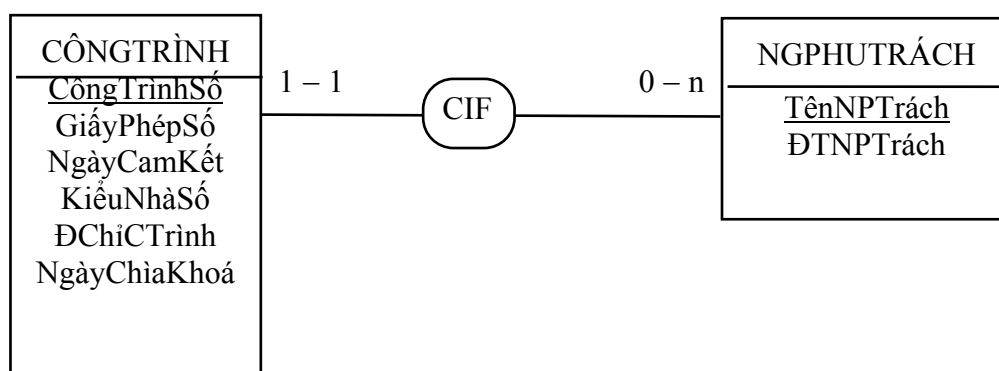
## b) Quan hệ giữa CÔNGTRÌNH và NGPHUTRÁCH

Ta có cấu trúc kiểu CHA-CON khác bổ sung cho cấu trúc kiểu CÓ-KHÔNG.

Chúng ta giải thích cấu trúc kiểu này như sau :

Mỗi công trình do một và chỉ một người phụ trách. Ngược lại, một người phụ trách có thể không phụ trách một công trình nào ở thời điểm nào đó (cấu trúc kiểu CÓ-KHÔNG), nhưng có thể đồng thời phụ trách nhiều công trình. Ta có cấu trúc kiểu CHA-CON, trong đó, thực thể NGPHUTRÁCH có vai trò là cha và thực thể CÔNGTRÌNH có vai trò là các con.

Từ đó, các dữ liệu TênNPTrách và ĐTNPTrách, vốn dĩ là các thuộc tính của CÔNGTRÌNH, bây giờ trở thành thuộc tính của NGPHUTRÁCH mà không còn thuộc về CÔNGTRÌNH nữa.

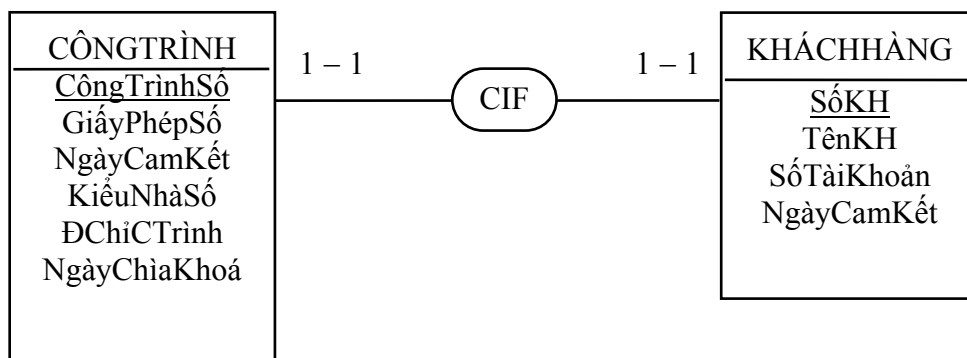


Lúc này, dữ liệu ĐTNPTrách là đích của một phụ thuộc hàm mà TênNPTrách là nguồn. Ta lại có PTH với TênNPTrách là đích và CôngTrìnhSố là nguồn. Đây là các PTH bắc cầu.



c) **Quan hệ giữa CÔNG TRÌNH và KHÁCH HÀNG**

Đây là cấu trúc kiểu BÌNHĐẲNG.

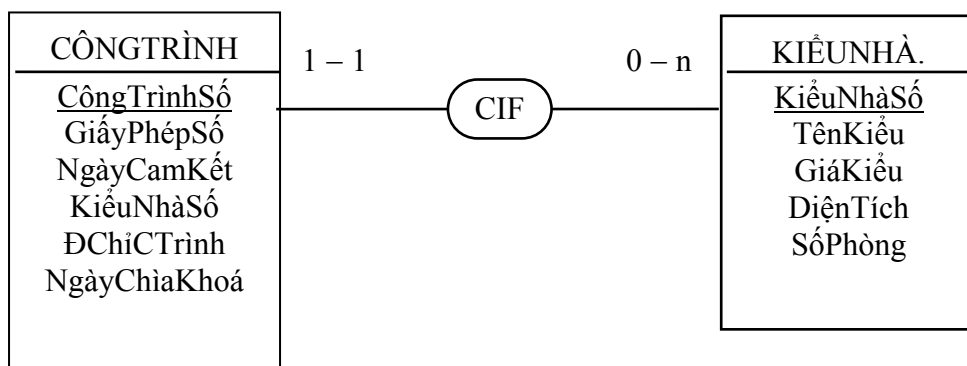


Đối với BKCO, mỗi khách hàng chỉ có một và chỉ một công trình, một công trình chỉ thuộc về một và chỉ một khách hàng mà thôi.

d) **Chú ý**

Mặc dầu chưa mô tả tường minh, nhưng tồn tại quan hệ giữa các thực thể CÔNG-TRÌNH và KIỂUNHÀ. Mỗi công trình chỉ liên quan đến một và chỉ một kiểu nhà. Ở đây, loại bỏ khả năng một khách hàng có thể xây dựng trên cùng chỗ hai nhà cùng kiểu hay theo hai kiểu nhà khác nhau.

Ngược lại, mỗi kiểu nhà có thể không có công trình nào động tới (cấu trúc kiểu CÓ-KHÔNG), hoặc có nhiều công trình thực thi, hoặc có nhiều khách hàng cùng lựa chọn. Từ đó, ta có cấu trúc kiểu CHA-CON kết hợp với cấu trúc kiểu CÓ-KHÔNG sau đây :



### II.1.5. Hỗn hợp các cấu trúc kiểu

Bây giờ ta có thể xây dựng một mô hình dữ liệu tổng thể bằng cách tổ hợp các cấu trúc kiểu hay các mô hình con đã xây dựng ở mục trước. Ta bắt đầu bằng một cấu trúc kiểu, sau đó lần lượt thêm vào mô hình các cấu trúc kiểu khác.

Ta bắt đầu xây dựng mô hình tổng thể như sau :

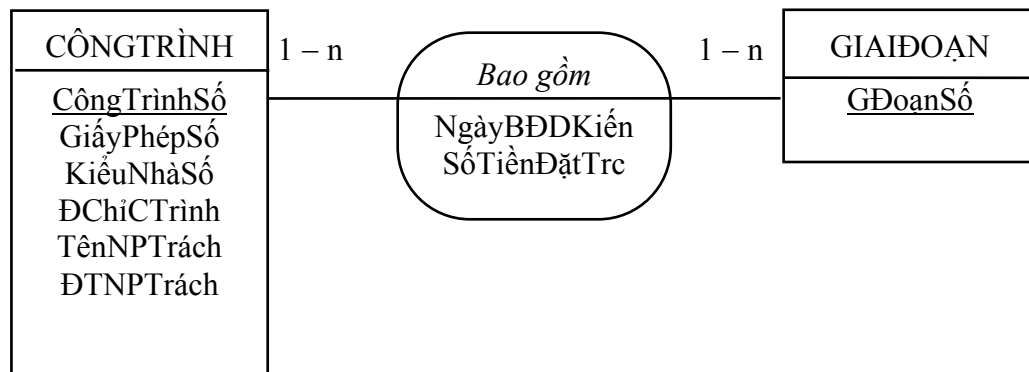
1. Phiếu khách hàng

KHÁCH HÀNG
<u>SốKH</u>
TênKH
ĐịaChỉKH
SốTàiKhoản
TênNgânHàng
KiểuNhàSố
CôngTrìnhSố
NgàyCamKét

2. Phiếu kiểu nhà

KIỂU NHÀ
<u>KiểuNhàSố</u>
TênKiểu
GiáKiểu
DiệnTích
SốPhòng

3. Chương trình thi công



Trước khi tiến hành tổ hợp, cần chú ý hai điểm sau :

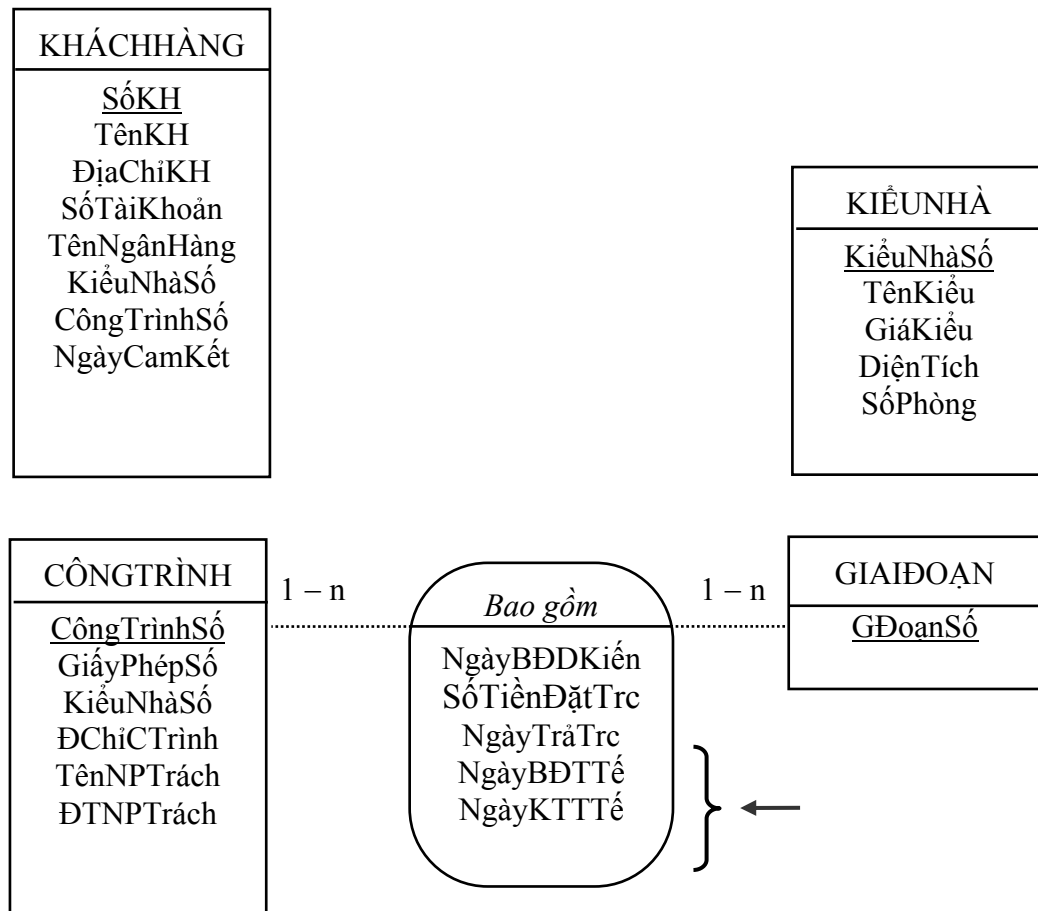
Dầu rằng một số thực thể có thể xuất hiện trong nhiều mô hình con, chúng chỉ có thể xuất hiện một lần trong mô hình tổng thể.

Một số dữ liệu có thể xuất hiện trong nhiều thực thể hoặc trong nhiều kết hợp không phân cấp. Điều này hoàn toàn hợp lý đối với các mô hình con nhưng không còn hợp lý đối với mô hình tổng thể.

Dữ liệu NgàyCamKét không thể xuất hiện trong cả hai thực thể, mà chỉ có thể xuất hiện trong KHÁCH HÀNG, vì đây là điều cam kết của khách hàng.

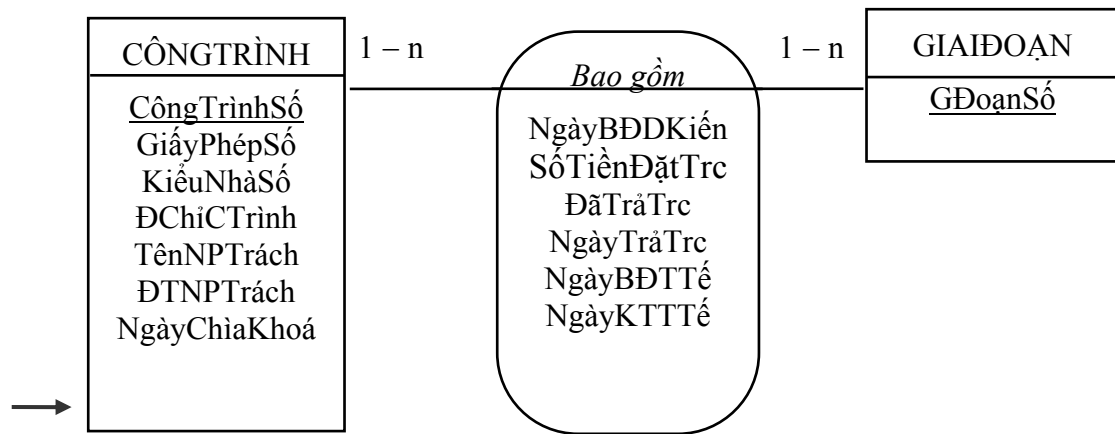
Ta tiếp tục qua bước 4 : Thêm các thuộc tính mới vào kết hợp “Bao gồm”.

4. Tiến độ thi công các công trình :



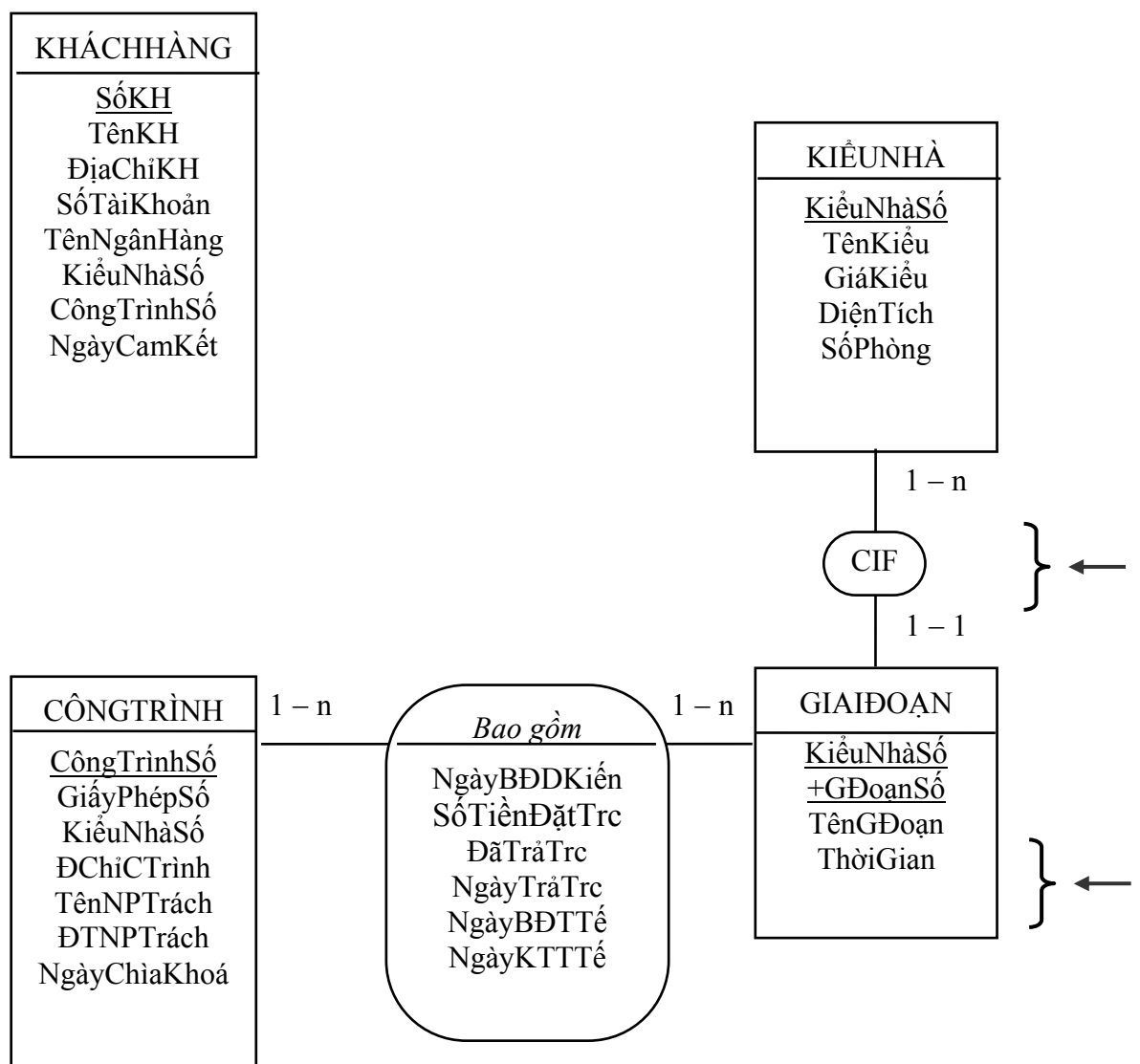
5. Theo dõi công trình :

Hai thực thể KHÁCH HÀNG và KIỂU NHÀ chưa có gì thay đổi. Lúc này, thêm thuộc tính NgàyChiaKhoá vào thực thể CÔNG TRÌNH :

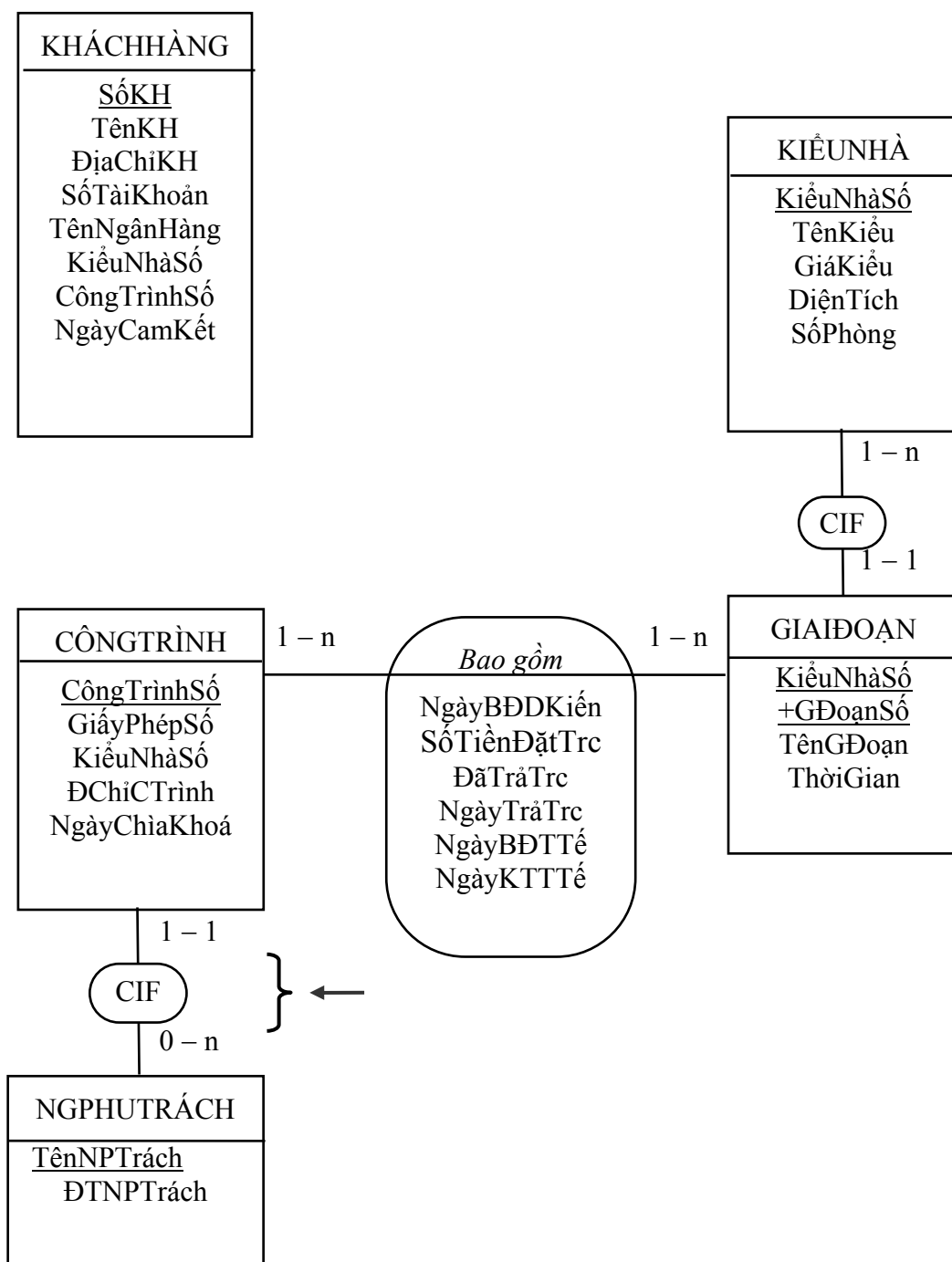


6. Hoàn thiện từng bước

a. Xác lập quan hệ giữa KIỂUNHÀ và GIAI ĐOẠN và hoàn thiện GIAI ĐOẠN

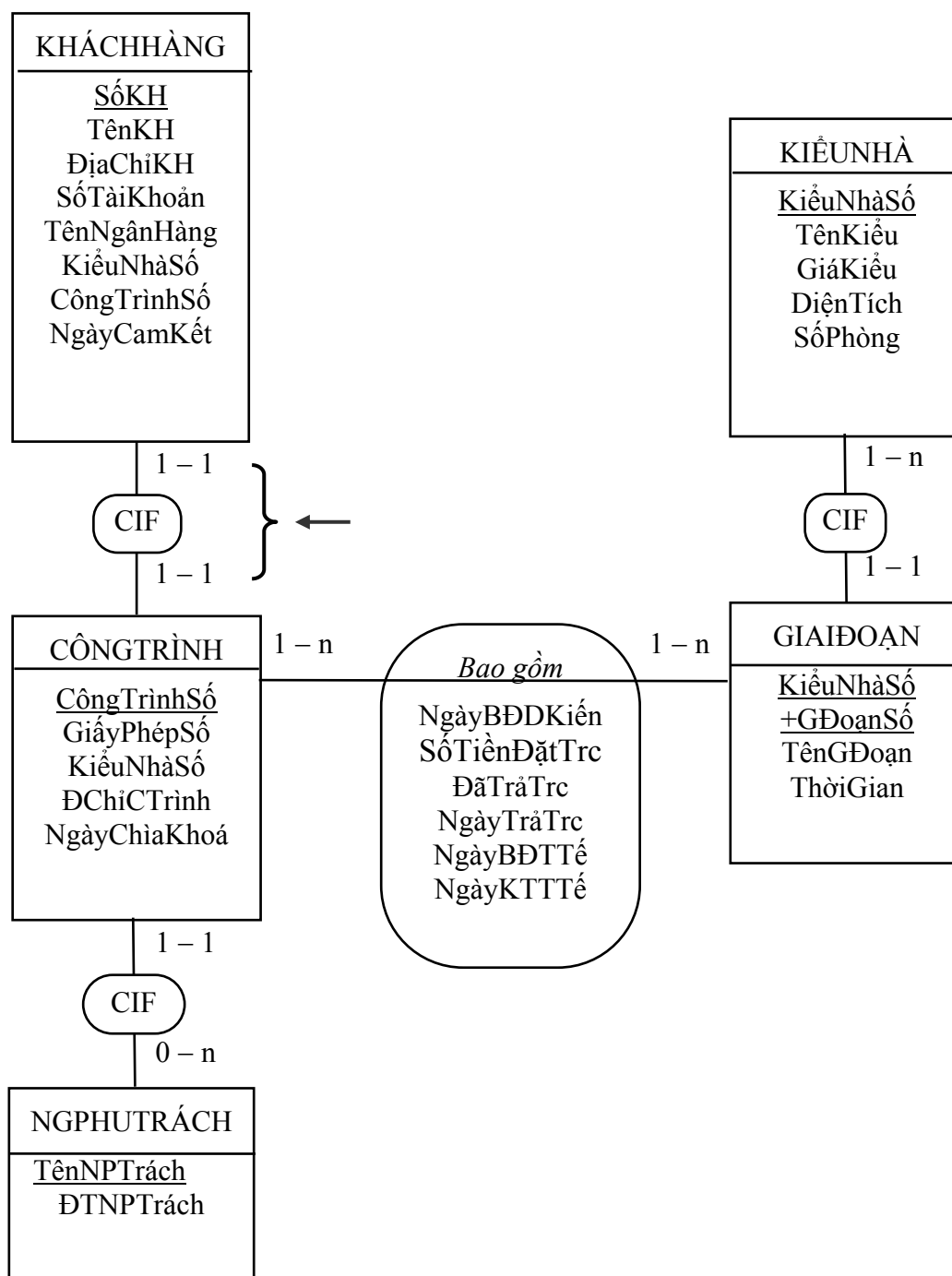


b) Thêm thực thể NGPHUTRÁCH và một kết hợp phân cấp giữa thực thể này với CÔNGTRÌNH



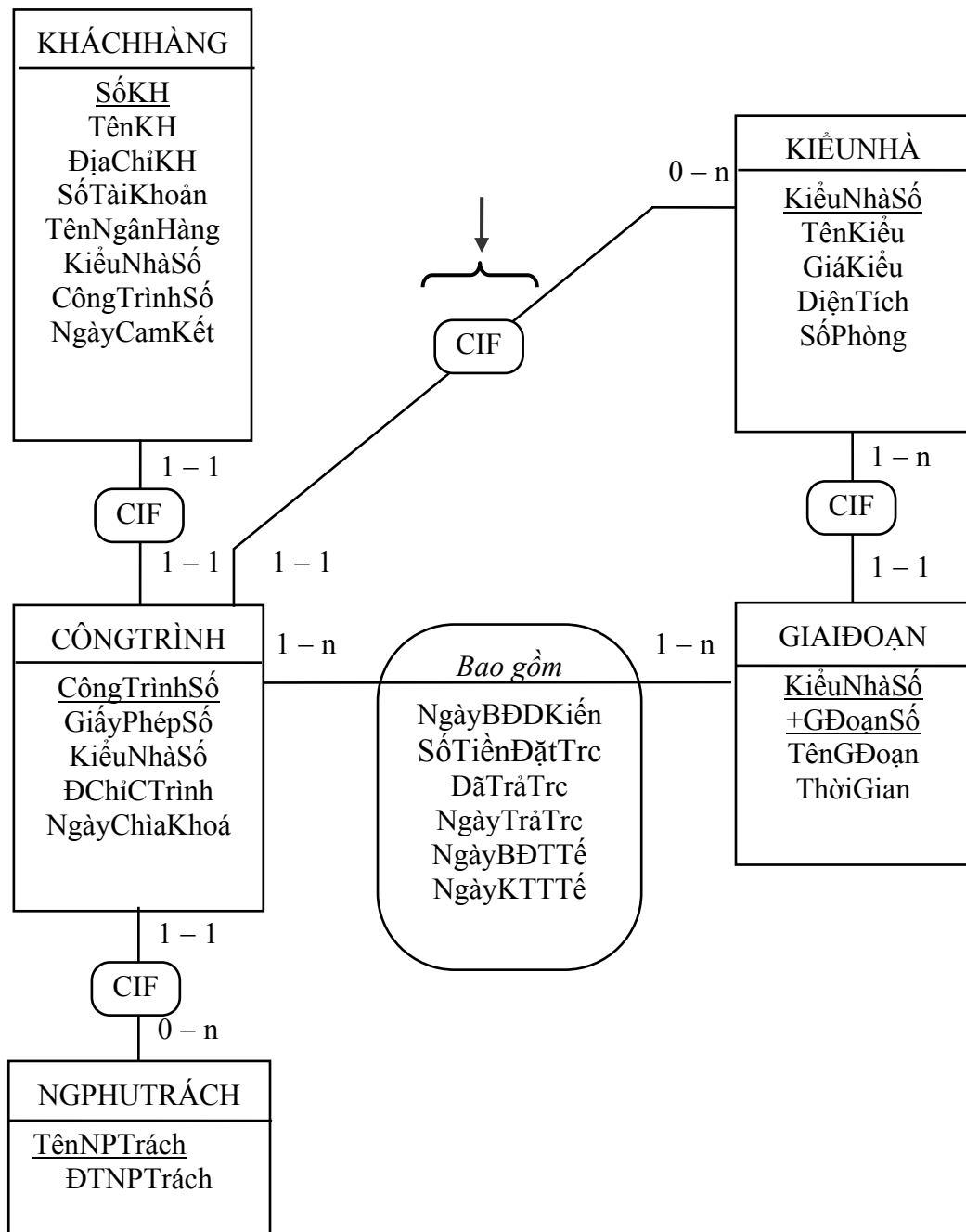
Các dữ liệu TênNPTrách và ĐTNPTrách không còn trong CÔNGTRÌNH nữa mà trở thành các thuộc tính của thực thể NGPHUTRÁCH

c) Xác lập mối liên hệ giữa thực thể CÔNG TRÌNH và KHÁCH HÀNG



Ta thấy thuộc tính CôngTrìnhSố của KHÁCH HÀNG là khoá của thực thể CÔNG TRÌNH. Dữ liệu này không còn nằm trong KHÁCH HÀNG nữa. Mô hình chỉ ra rằng với mỗi khách hàng, chỉ có một và chỉ một công trình. Sự nhận biết một khách hàng kéo theo sự nhận biết công trình tương ứng và ngược lại.

d) Xác lập mối liên hệ giữa thực thể CÔNGTRÌNH và KIỂUNHÀ.



Dữ liệu KiểuNhàSố, khoá của KIỂUNHÀ, cũng là thuộc tính của KHÁCHHÀNG và của CÔNGTRÌNH. Dữ liệu này cần được xoá khỏi hai thực thể trên. Bởi vì các kết hợp KHÁCHHÀNG – CÔNGTRÌNH và CÔNGTRÌNH – KIỂUNHÀ, chỉ ra rằng, khi biết một khách hàng thì biết công trình tương ứng, và, khi biết công trình thì cũng biết khách hàng tương ứng.

Do tính bắc cầu, khi biết một khách hàng thì cũng biết kiểu nhà mà khách hàng đó đã lựa chọn.

*e) Kết quả của tổ hợp các cấu trúc kiểu*

Ta đã xây dựng xong mô hình dữ liệu căn cứ vào việc phân tích các phiếu và thông tin thu lượm được từ công ty BKCO. Nếu như vẫn còn những chỗ hoài nghi về tính đúng đắn của mô hình, những trao đổi trực tiếp với BKCO vẫn còn là cần thiết sau giai đoạn phân tích hiện trạng. Bước tiếp theo là hợp thức hoá mô hình dữ liệu này.

## II.2. Hợp thức hoá mô hình ý niệm dữ liệu

Sau khi thành lập từ điển dữ liệu và ma trận các PTH, vấn đề là áp dụng các quy tắc hợp thức hoá đối với mô hình ý niệm dữ liệu đã xây dựng.

### II.2.1. Từ điển dữ liệu

Từ điển dữ liệu cho HTTT của BKCO được thành lập như sau :

Stt	Dữ liệu	Loại	Kiểu
1	CôngTrìnhSố		N
2	DiệnTích		N
3	ĐãTrảTrc		L
4	ĐChỉCTrình		C
5	ĐịaChỉKH		C
6	ĐTNPTrách		N
7	GĐoạnSố		N
8	GiáKiểu	TT	N
9	GiấyPhépSố		N
10	KHSố		N
11	KiểuNhàSố		N
12	NgàyBĐDKiến		D
13	NgàyBĐTTế		D

Stt	Dữ liệu	Loại	Kiểu
1	NgàyCamKết		D
2	NgàyChiaKhoá		D
3	NgàyKTTTế		D
4	NgàyTrảTrc		D
5	SốPhòng		N
6	SốTàiKhoản		N
7	SốTiềnĐặtTrc	TT	N
8	TênGĐoạn		C
9	TênKH		C
10	TênKiểu		C
11	TênNgânHàng		C
12	TênNPTrách		C
13	ThờiGian		N

*Ghi chú :* Các dữ liệu có *Loại* = TT là các dữ liệu tính toán theo công thức



### II.2.2. Ma trận rút gọn các PTH

Ta thêm vào ma trận rút gọn các PTH các cột mới có nội dung là ghép các dữ liệu khoá. Trong ma trận có thể có các PTH phân xạ.

		Nguồn (khoá)					
Đích		1	10	11	25	11+7	1+11+7
1	<u>CôngTrìnhSố</u>	1	1				1
2	DiệnTích			1			
3	ĐãTrảTrc						1
4	ĐChỉCTrình	1					
5	ĐịaChỉKH		1				
6	ĐTNPTrách				1		
7	<u>GĐoanSố</u>					1	1
8	GiáKiểu			1			
9	GiấyPhépSố	1					
10	<u>KHSố</u>	1	1				
11	<u>KiểuNhàSố</u>	1		1		1	1
12	NgàyBĐDKiến						1
13	NgàyBĐTTế						1
14	NgàyCamKết		1				
15	NgàyChìaKhoá	1					
16	NgàyKTTTế						1
17	NgàyTrảTrc						1
18	SốPhòng			1			
19	SốTàiKhoản		1				
20	SốTiềnĐặtTrc						1
21	TênGĐoạn					1	
22	TênKH		1				
23	TênKiểu			1			
24	TênNgânHàng		1				
25	<u>TênNPTrách</u>	1			1		
26	ThờiGian					1	

### ***II.2.3. Quy tắc hợp thức hoá mô hình ý niệm dữ liệu***

#### ***a) Quy tắc thống nhất các thuộc tính***

*Một thuộc tính chỉ có nghĩa đối với một thực thể hoặc một kết hợp duy nhất.*

Ma trận các PTH chỉ rõ quy tắc này. Vì rằng đối với các dữ liệu khoá, sẽ có nhiều con 1 trên dòng tương ứng, còn các dòng khác chỉ có duy nhất một con 1.

#### ***b) Quy tắc thống nhất các kết hợp***

*Với mỗi trường hợp cụ thể của một kết hợp, chỉ có thể tồn tại một và chỉ một trường hợp cụ thể của mỗi thực thể tham gia vào kết hợp này và ngược lại.*

Quy tắc này chỉ đặt ra đối với các kết hợp không phân cấp. Như đã thấy, kết hợp không phân cấp “Bao gồm” gồm một tập hợp các thuộc tính nhận giá trị với mỗi công trình và mỗi giai đoạn.

Quy tắc thống nhất các kết hợp cũng được kiểm chứng trong ma trận các PTH, trong đó, khoá của kết hợp không phân cấp được ghép từ các khoá của các thực thể tham gia vào kết hợp này : CôngTrìnhSố + KiểuNhàSố + GĐoạnSố.

Các cấu trúc kiểu BẢNG cho phép giải thích tường minh quy tắc này.

#### ***c) Quy tắc các thuộc tính và khoá của một thực thể***

*Nếu một thuộc tính phụ thuộc vào nhiều khoá của nhiều thực thể, thuộc tính này phải thuộc về một kết hợp giữa các thực thể này.*

Đó là trường hợp của kết hợp “Bao gồm”.

*Nếu một thuộc tính phụ thuộc vào khoá của thực thể chứa nó và đồng thời phụ thuộc vào một thuộc tính khác cũng trong thực thể này, thì tồn tại một thực thể con (bị bao hàm) trong đó.*

Đó là trường hợp của các thực thể TênNPTrách và ĐTNPTrách lúc đầu xuất hiện trong thực thể CÔNGTRÌNH. Việc tạo ra thực thể NGPHUTRÁCH đã khắc phục vấn đề này.

## **II.3. Ràng buộc toàn vẹn**

Cấu trúc dữ liệu không là đặc trưng duy nhất của một mô hình dữ liệu. Các ràng buộc tồn tại trên và giữa các dữ liệu làm phong phú thêm và tăng cường tính hiệu quả của mô hình đang xét.

Như vậy, các ràng buộc toàn vẹn đảm bảo tính chặt chẽ của dữ liệu, dù cách thao tác trên dữ liệu là như thế nào.

#### ***a) Ràng buộc toàn vẹn của thực thể***

Mỗi trường hợp cụ thể của thực thể phải được xác định.

Mỗi thực thể phải có một khoá. Giá trị của khoá không thể là 0, dù đó là dữ liệu sơ cấp hay dữ liệu ghép. Nếu là dữ liệu ghép thì không có thành phần nào trong đó có giá trị 0.

#### ***b) Ràng buộc toàn vẹn tham chiếu***

Ràng buộc toàn vẹn tham chiếu liên quan đến các kết hợp.

Trong kết hợp có phân cấp, ràng buộc toàn vẹn hàm (CIF) giữa thực thể KHÁCHHÀNG và thực thể ĐẶTHÀNG có nghĩa rằng không thể tồn tại một trường hợp cụ thể của ĐẶTHÀNG quan hệ với một khách hàng mà một trường hợp cụ thể tương ứng của KHÁCHHÀNG lại chưa được tạo ra.

Một các tương tự, không thể loại bỏ một trường hợp cụ thể của KHÁCHHÀNG chừng nào các trường hợp cụ thể của ĐẶTHÀNG tương ứng lại chưa được loại bỏ.

Vấn đề ràng buộc toàn vẹn tham chiếu cũng đặt ra đối với các kết hợp không phân cấp. Chúng ta lấy lại ví dụ về ràng buộc toàn vẹn bội (CIM) giữa các thực thể KHÁCHHÀNG và THÁNG. Ràng buộc này có mang dữ liệu là DSỐTháng.

Rõ ràng, mỗi trường hợp cụ thể của ràng buộc này, tức là mỗi DSỐTháng, phải tương ứng với một trường hợp cụ thể của KHÁCHHÀNG và với một trường hợp cụ thể của THÁNG.

**c) Ràng buộc toàn vẹn trên miền xác định của dữ liệu**

Mỗi dữ liệu chỉ có thể nhận giá trị trong miền xác định đã định nghĩa.

**d) Ràng buộc toàn vẹn biến đổi**

Giá trị của một dữ liệu nào đó chỉ có thể biến đổi khi có sự cập nhật. Ví dụ, giá của một mặt hàng nào đó có thể biến đổi khi tính toán.

**e) Ràng buộc toàn vẹn liên kết**

Giá trị của một dữ liệu nào đó không thể biến đổi một cách hoàn toàn độc lập. Ví dụ, việc quản lý thuê chỗ đã đăng ký trong bảng THUÊ : không thể có một ngày thuê không nằm trong khoảng thời gian khách hàng đã lưu trú thực tế (ngày đến - ngày đi).

**f) Ràng buộc toàn vẹn tổng thể**

Xuất hiện ràng buộc toàn vẹn tổng thể khi bản số tối thiểu của một thực thể gắn với một kết hợp có giá trị 1. Điều này có nghĩa mọi trường hợp cụ thể của thực thể này đều tham gia vào kết hợp đang xét.

Trong ràng buộc toàn vẹn hàm giữa thực thể XEHƠI và thực thể KIỂUXE, bản số tối thiểu của XEHƠI phải bằng 1 vì xe hơi nào cũng thuộc về một kiểu.

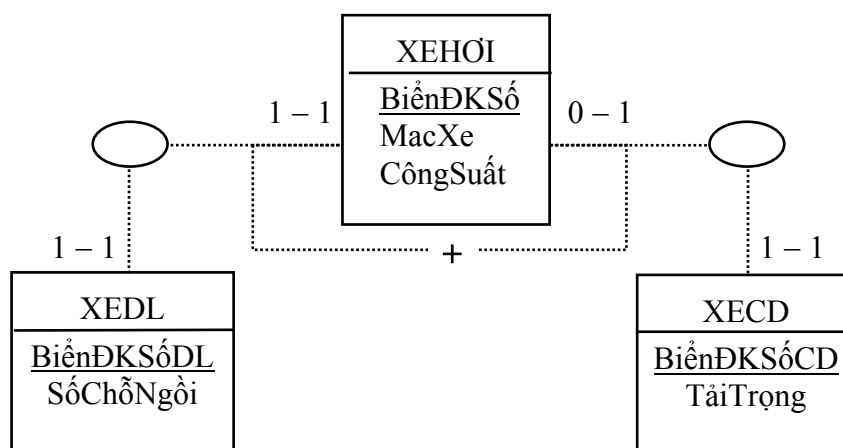
**g) Ràng buộc toàn vẹn bình đẳng**

Xuất hiện khi một trường hợp cụ thể của một kết hợp tự động kéo theo một trường hợp cụ thể của một thực thể khác và ngược lại.

Đây chính là cấu trúc kiểu BÌNHĐẲNG mà ta đã xét.

**h) Ràng buộc toàn vẹn loại trừ**

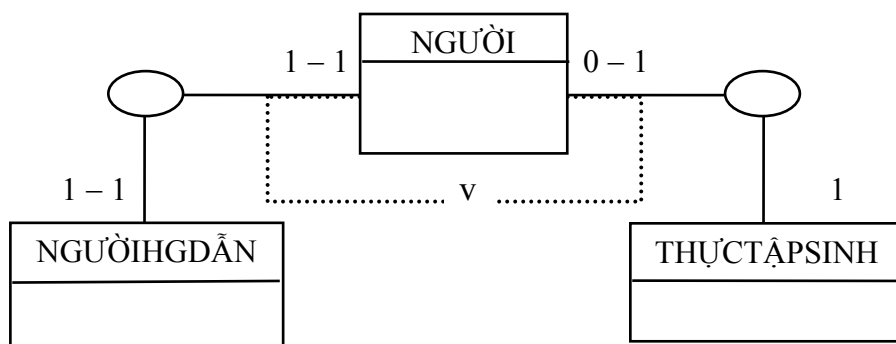
Ta đã xét cấu trúc kiểu THỪAKẾ có mô hình như sau :



Dấu “+” chỉ ra rằng đây là ràng buộc toàn vẹn loại trừ. Có nghĩa rằng một xe hơi, hoặc là loại xe du lịch (DL), hoặc là loại xe chuyên dụng (CD), nhưng không thể cùng lúc là cả hai. Mỗi trường hợp cụ thể của XEHƠI tương ứng với một trường hợp cụ thể của XEDL, hoặc (hoặc “loại trừ”) một trường hợp cụ thể của XECD.

**i) Ràng buộc toàn vẹn bao hàm**

Xét ví dụ sau đây :



Trong dịp thực tập nghề tại một cơ sở, mỗi thành viên có thể là một thực tập sinh nào đó, hoặc là hướng dẫn viên duy nhất hoặc là hướng dẫn viên cho một giai đoạn thực tập nhưng lại là thực tập sinh ở một giai đoạn khác. Chính vì vậy ở đây ta có dấu “v”.

Một trường hợp cụ thể của thực thể NGƯỜI tương ứng với một trường hợp cụ thể của thực thể NGƯỜIHƯỚNGDẪN hoặc (hoặc “bao hàm”) một trường hợp cụ thể của thực thể THỰCTẬP SINH.

Chú ý đối với các ràng buộc toàn vẹn loại trừ và bao hàm : Trong cả hai trường hợp ví dụ, ta thấy có sự bao hàm. Một mặt, tập hợp các xe du lịch và tập hợp các xe chuyên dụng được bao hàm trong tập hợp các xe hơi. Mặt khác, tập hợp các hướng dẫn viên và tập hợp các thực tập sinh được bao hàm trong tập hợp người.

Việc định nghĩa các ràng buộc toàn vẹn loại trừ và bao hàm đã làm phong phú thêm các mô hình thực thể kết hợp trên bình diện ngữ nghĩa.

## II.4.Ví dụ : Bài toán quản lý du lịch

Một công ty du lịch tổ chức các chuyến du lịch vòng quanh thế giới. Hoạt động của công ty được mô tả qua các qui tắc quản lý như sau :

### a) Qui tắc quản lý

- QT 1 Mọi hồ sơ về khách hàng của công ty được lưu giữ lâu dài.
- QT 2 Trong một hành trình (chặng), tại mỗi thành phố, nếu phải lưu trú qua đêm thì khách du lịch chỉ lưu trú tại một khách sạn duy nhất. Ví dụ khi đến thành phố Đà Nẵng, Việt Nam, khách du lịch luôn luôn ở khách sạn Tre Xanh (Bamboo Green).
- QT3 Nếu khách du lịch phải qua đêm ở một thành phố nào đó trong hành trình thì lưu trú ở khách sạn chỉ định ở QT 2.
- QT 4 Tên các thành phố không được trùng nhau. Ví dụ : trong hệ thống, chỉ có duy nhất một thành phố tên Đà Nẵng.
- QT 5 Mỗi hành trình luôn có một và chỉ một người hướng dẫn du lịch của công ty.
- QT 6 Mỗi hành trình phải qua ít nhất hai thành phố.
- QT 7 Mỗi hành trình không nhất thiết phải đi qua hết tất cả các thành phố theo kế hoạch dự kiến trong hành trình.
- QT 8 Trong cùng một ngày của hành trình, với mỗi thành phố, chỉ đến và từ đấy đi duy nhất một lần.
- QT 9 Trước mỗi vòng du lịch dự kiến, thông tin về vòng du lịch lần trước liên quan đến các hãng hàng không, đường sắt, khách sạn, v.v... được phổ biến rộng rãi cho khách hàng.
- QT 10 Khi khách hàng yêu cầu đặt chỗ trước thì tùy theo tình hình thực tế phải trả lời có chấp thuận hay không (từ chối nếu hết chỗ).

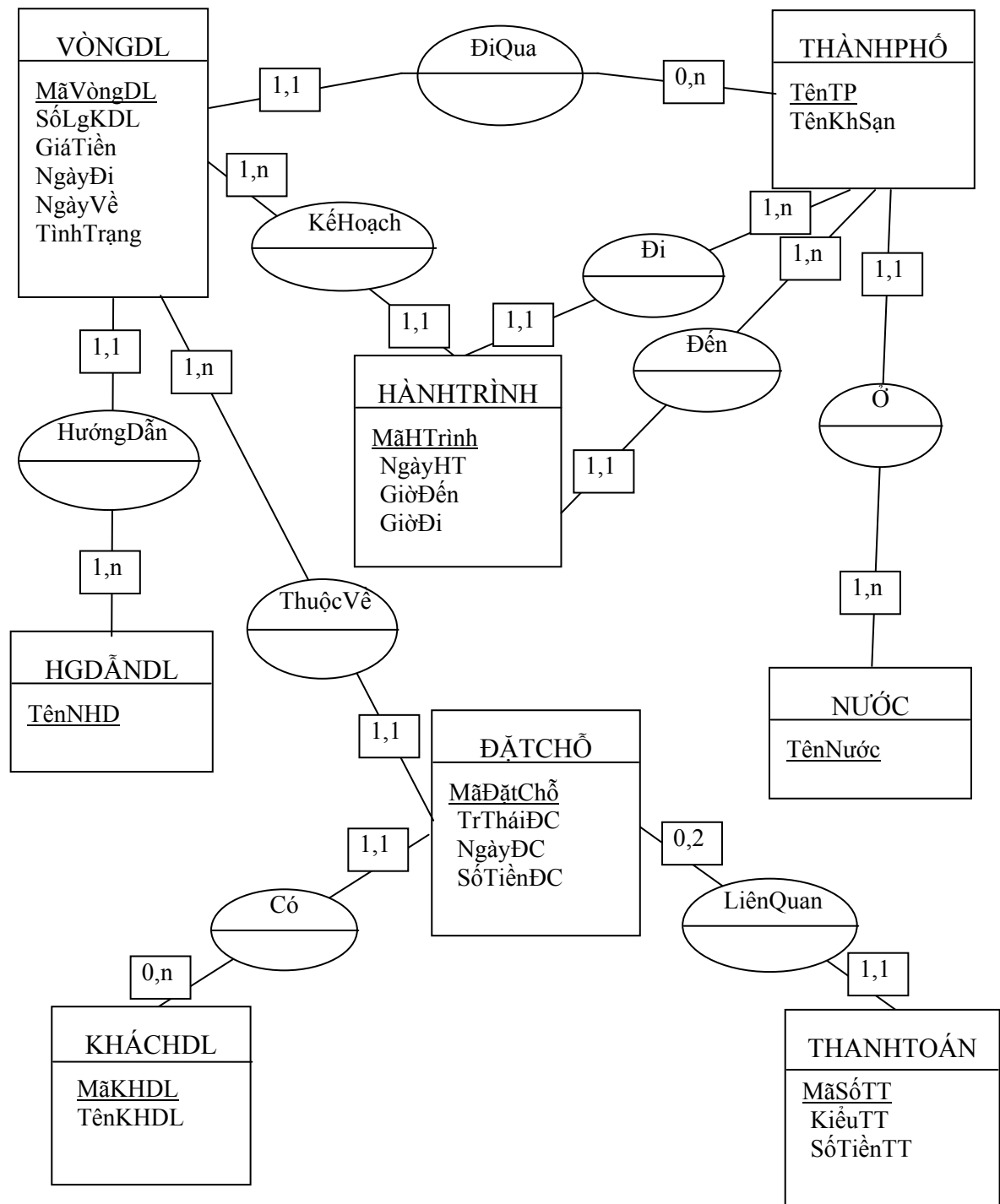
- QT 11 Khách hàng chỉ đặt chỗ được nếu còn chỗ (QT 10) và lúc đó phải đặt cọc trước một số tiền, sau đó trả dần phần còn lại + lãi suất nếu không trả hết một lần.
- QT 12 Việc đặt chỗ chỉ có hiệu lực khi lần trả tiền sau cùng đã hoàn đủ lãi suất của các lần trả trước.
- QT 13 Sau một ngày đã định N1 nào đó (chẳng hạn cách một tháng trước ngày khởi hành), những khách hàng nào chưa trả đủ tiền sẽ bị loại để bố trí cho khách hàng khác, dù họ có phàn nàn hoặc sau đó có yêu cầu trả tiền tiếp.
- QT 14 Sau một ngày quy định N2 nào đó (chẳng hạn cách đúng 15 ngày trước ngày khởi hành) :
- Nếu không đủ số khách theo kế hoạch, công ty phải hủy bỏ hành trình và hoàn trả tiền lại cho những khách hàng đã trả đủ.
  - Nếu không có khách hàng nào trả tiền theo đúng yêu cầu thì vòng du lịch dự kiến cũng bị hủy bỏ.
  - Trường hợp có đủ số khách theo kế hoạch, vòng du lịch được ghi nhận.

**b) Xây dựng mô hình E–A**

Xây dựng từ điển các thực thể như sau :

Stt	Kiểu thực thể	Khóa	Thuộc tính	Giải thích
1	HGDẪNDL	TênNHD	TênNHD	Tên người hướng dẫn du lịch
2	VÒNGDL	MãVòngDL	MãVòngDL SốLgKDL GiáTiền NgàyĐi NgàyVề TìnhTrạng	Mã số thứ tự vòng du lịch Số chỗ dự kiến cho một vòng DL Giá tiền Ngày khởi hành Ngày trở về Tình trạng của chuyến DL
3	KHÁCHDL	MãKHDL	MãKHDL TênKHDL	Mã số khách DL Họ tên khách DL
4	HÀNHTRÌNH	MãHTrình	MãHTrình NgàyHT GiờĐến GiờĐi	Mã số của hành trình Ngày đi và đến trong hành trình Giờ đến đích hành trình Giờ xuất phát
5	THANHTOÁN	MãSốTT	MãSốTT KiểuTT SốTiềnTT	Mã số thanh toán Kiểu thanh toán Số tiền thanh toán
6	NƯỚC	TênNước	TênNước	Tên nước đến và đi trong hành trình
7	ĐẶTCHỖ	MãĐặtChỗ	MãĐặtChỗ TrTháiĐC NgàyĐC SốTiềnĐC	Mã số của việc đặt chỗ Trạng thái đặt chỗ Ngày đặt chỗ Số tiền đặt chỗ đợt này
8	THÀNHPHỐ	TênTP	TênTP TênKhSạn	Tên thành phố Tên khách sạn nơi lưu trú qua đêm

Mô hình thực thể-kết hợp cho bài toán quản lý du lịch :



### c) **Lược đồ quan hệ**

Từ sơ đồ E-A, ta có thể chuyển thành LĐQH như sau :

- Kiểm tra các qui tắc 1 và 2 : mọi kiểu đặc tính trong sơ đồ E-A trên đây đều là dữ liệu đơn và sơ cấp.
- Kiểm tra qui tắc 3 : mọi kiểu thực thể đều có một kiểu đặc tính khóa.
- Kiểm tra qui tắc 4 : mọi kiểu kết hợp đều nhị nguyên.
- Kiểm tra qui tắc 5 : mọi kiểu kết hợp đều có ít nhất một bản số cực đại là 1.
- Áp dụng các qui tắc 6, 7, 8, 9 để đưa đến các lược đồ quan hệ như sau (để cho tiện theo dõi, ta sử dụng toàn bộ chữ in hoa cho các tên thuộc tính) :

VÒNGDL (MÃVÒNGDL, SÓLGKDL, GIÁTIỀN, TÊNTTP, NGÀYĐI, NGÀYVỀ,  
TÌNHTRẠNG, TÊNNHD)

HÀNHTRÌNH (MÃVÒNGDL, MÃHTRÌNH, NGÀYYHT, GIỜĐẾN, GIỜĐI, TÊNTPĐI,  
TÊNTPĐẾN)

HGDÃNDL (TÊNNHD)

THÀNHPHỐ (TÊNTTP, TÊNKHSAN, TÊNNƯỚC)

NƯỚC (TÊNNƯỚC)

ĐẶTCHỖ (MÃĐẶTCHỖ, TRTHÁIĐC, NGÀYĐC, SÓTIỀNĐC, VÒNGDLSỐ,  
MÃKHDL)

THANHTOÁN (MÃSỐTT, KIỂUTT, SÓTIỀNTT, MÃĐẶTCHỖ)

KHÁCHDL (MÃKHDL, TÊNKHDL)

Chú ý : trong các chuyển đổi trên đây, do giữa hai kiểu thực thể HÀNHTRÌNH và THÀNHPHỐ tồn tại hai kiểu kết hợp, là Đi và Đến, nên ta đã phải thêm vào lược đồ quan hệ HÀNHTRÌNH hai đặc tính đồng nghĩa với khóa TÊNTTP (của thực thể THÀNHPHỐ) là TÊNTPĐI và TÊNTPĐẾN.

Mặt khác, lược đồ quan hệ HÀNHTRÌNH đã sửa đổi lại khóa bằng cách ghép hai đặc tính khóa MÃVÒNGDL và MÃHTRÌNH. Khóa ghép này cho phép liệt kê thứ tự các hành trình trong một vòng du lịch.

## II.5.Các ràng buộc toàn vẹn dữ liệu

Sau đây là một số ràng buộc toàn vẹn áp dụng cho các lược đồ quan hệ đã nêu

### II.5.1.Các ràng buộc tĩnh đối với các quan hệ

#### a) Các ràng buộc đối với quan hệ VÒNGDL

VÒNGDL (MÃVÒNGDL, SÓLGKDL, GIÁTIỀN, TÊNTTP, NGÀYĐI, NGÀYVỀ,  
TÌNHTRẠNG, TÊNNHD)

Khóa chính : MÃVÒNGDL

Khóa ngoại : TÊNNHD, tham chiếu đến quan hệ HGDÃNDL

Miền giá trị của TÌNHTRẠNG : { ‘Dự kiến’, ‘Hủy bỏ’, ‘Đang thực hiện’, ‘Đã qua’ }

Miền giá trị của GIÁTIỀN: một số nguyên lớn hơn 500 (USD)

Ràng buộc cho các bộ-n : NGÀYĐI < NGÀYVỀ

Ràng buộc khác : hai đặc tính GIÁTIỀN và TÊNNHD không được thay đổi một cách tùy tiện.

#### b) Các ràng buộc đối với quan hệ HÀNHTRÌNH

HÀNHTRÌNH (MÃVÒNGDL, MÃHTRÌNH, NGÀYYHT, GIỜĐẾN, GIỜĐI, TÊNTPĐI,  
TÊNTPĐẾN)

Khóa chính : MÃVÒNGDL, MÃHTRÌNH

Khóa ngoại : MÃVÒNGDL tham chiếu đến quan hệ VÒNGDL,  
TÊNTPĐI và TÊNTPĐẾN tham chiếu đến quan hệ THÀNHPHỐ

Miền giá trị của MÃVÒNGDL: một số nguyên  $\in 1..30$

Miền giá trị của GIỜĐẾN : một số  $\in 12..21$  đứng trước chữ G

Miền giá trị của GIỜĐI : một số  $\in 6..11$  đứng trước chữ G

Ràng buộc cho các bộ-n : luôn luôn TÊNTPĐI  $\neq$  TÊNTPĐẾN

**c) Ràng buộc đặc trưng cho tất cả các quan hệ**

**RB1** : Trong một vòng du lịch, các mã số hành trình chỉ định thứ tự các giai đoạn của vòng du lịch và do đó phải có tính tương thích giữa các mã số hành trình, ngày thực hiện hành trình và các thành phố đi, thành phố đến.

Ràng buộc được mô tả như sau :

$\forall d1 \text{ và } d2 \in \text{HÀNHTRÌNH}$  sao cho :

$(d1.MÃVÒNGDL = d2.MÃVÒNGDL)$  và  $(d2.MÃHTRÌNH = d1.MÃHTRÌNH + 1)$ ,

thì ta phải có :

$(d1.TÊNTPĐẾN = d2.TÊNTPĐI)$  và  $(d2.NGÀYHT > d1.NGÀYHT)$

Nghĩa là trong một vòng du lịch, thành phố đến ở hành trình thứ  $i$  là thành phố đi ở hành trình thứ  $i+1$  và ngày thực hiện hành trình thứ  $i+1$  phải sau ngày thực hiện hành trình thứ  $i$ .

**d) Các ràng buộc kết hợp với các quan hệ khác**

Sau đây là các ràng buộc kết hợp với các quan hệ HGDÃNDL, THÀNHPHỐ, ĐẶTCHỖ, THANHTOÁN và KHÁCHDL

Quan hệ	Khóa chính	Khóa ngoại	Miền giá trị
HGDÃNDL	TÊNNHD		
THÀNHPHỐ	TÊNTP	TÊNNƯỚC	
ĐẶTCHỖ	MÃĐẶTCHỖ	MÃVÒNGDL TÊNKHDL	TRTHÁIĐC = {‘Chấp nhận’, ‘Từ chối’, ‘Chắc chắn’, ‘Hủy bỏ’, ‘Hoàn trả’} Giá trị mặc nhiên : SỐTIỀNĐC = 0
THANHTOÁN	MÃSỐTT	MÃĐẶTCHỖ	Kiểu TT = {‘Tiền mặt’, ‘chuyển khoản’}
KHÁCHDL	MÃKHDL		

**II.5.2. Các ràng buộc đối với nhiều quan hệ**

**a) Ràng buộc toàn vẹn tham chiếu**

Ràng buộc này được áp dụng cho tất cả các đặc tính của các quan hệ, trừ đặc tính TÊNNHD thuộc quan hệ HGDÃNDL trong quan hệ VÒNGDL. Ở đây, người ta triển khai lập kế hoạch cho một vòng du lịch trước khi chỉ định người hướng dẫn.

**b) Ràng buộc đặc trưng**

**RB2** : Giữa các quan hệ VÒNGDL và HÀNHTRÌNH.

Thành phố khởi hành (đi) của mỗi một vòng du lịch phải là thành phố khởi hành của hành trình đầu tiên thuộc vòng du lịch đó và thành phố đích (về) của hành trình cuối cùng.

$\forall c \in \text{VÒNGDL} : \exists d1 \text{ và } d2 \in \text{HÀNHTRÌNH}$  sao cho :

$(d1.MÃVÒNGDL = d2.MÃVÒNGDL = c.MÃVÒNGDL)$  và  $(d1.MÃHTRÌNH = 1)$  và

$(d1.TÊNTPĐI = c.TÊNTP)$  và  $(d2.TÊNTPĐẾN = c.TÊNTP)$

**II.5.3. Các ràng buộc toàn vẹn động**

**RB3** : Khi một vòng du lịch ở tình trạng ‘Hủy bỏ’, cần phải chuyển đổi các đặt chỗ ‘Chắc chắn’ thành ‘Hoàn trả’ (quy tắc 14).



Để đặc tả kiểu ràng buộc này, cần sử dụng một ngôn ngữ cho phép sử dụng các phép toán trên các quan hệ. Tuy nhiên, việc thiết kế CSDL trong một ngôn ngữ như vậy không đơn giản và tùy theo môi trường CSDL đang sử dụng mà có cách giải quyết phù hợp.

Ví dụ trong ngôn ngữ FoxPro, ta có thể viết :

```
...  
N2 = NGÀYĐI-15  
SL-OK = 30 && Số khách đặt chỗ tối thiểu theo kế hoạch  
* Tính TSTDC = tổng tất cả SÓTIỀNĐC  
IF (DATE()>= N2 AND DATE()< NGÀYĐI) AND  
  ((SÓLGKDL < SL-OK) OR (TSTDC = 0))  
  TRTHÁIĐC = 'Hoàn trả'  
ELSE  
  TRTHÁIĐC = 'Chắc chắn'  
ENDIF  
...
```



## Bài tập chương 5

Yêu cầu xây dựng mô hình thực thể – kết hợp đối với các bài tập sau đây :

### 1. Quản lý phân phát báo tạp chí hàng ngày

Để quản lý việc phân phát báo và tạp chí hàng ngày của các nhân viên bưu điện, người ta sử dụng những thông tin tình huống như sau :

Mỗi tờ báo hay tạp chí đều có : mã, tên (tiêu đề), thể loại và đơn giá. Ví dụ : báo 01–*Nhân dân* ra hàng ngày (nhật báo) có giá bán 1 000 đ, tạp chí 13–*Thế giới Vi tính* ra hàng tháng (nguyệt san) có giá bán 8 000 đ, phụ san *Kiến thức ngày nay* ra 10 ngày một số, giá bán 4 500 đ, v.v...

Mỗi nơi nhận báo tạp chí được quản lý bởi số mã, tên người nhận (hoặc tên quầy báo), địa chỉ và số điện thoại (nếu có). Ví dụ ông Đinh Quang Vụ có mã DN325 ở tại địa chỉ 43b, Bạch Đằng. Quầy báo Tuổi Hoa có mã DN037 ở tại 64 Hoàng Diệu, v.v...

Nhân viên bưu điện có nhiệm vụ phân phát báo chí hàng ngày cho nơi nhận đã được đặt đăng ký theo lộ trình cố định và được ấn định sẵn. Mỗi nhân viên phụ trách phân phát không quá 20 địa chỉ. Cơ quan Bưu điện quản lý họ bởi mã nhân viên, họ tên, địa chỉ và số điện thoại.

Tuy nhiên nơi nhận báo có thể không nhận một số lượng báo cố định vì tùy theo ngày, theo tuần, theo tháng trong năm mà số lượng báo chí bán ra tại quầy có thể khác nhau. Theo kinh nghiệm hoặc nhu cầu thực tế mà nơi nhận yêu cầu cấp phát số lượng báo cụ thể cho từng ngày.

### 2. Quản lý tàu vào ra và bốc hàng tại cảng Đà Nẵng

Để quản lý theo dõi tàu vào ra và bốc hàng tại cảng Đà Nẵng, người ta sử dụng những quy tắc quản lý như sau :

Mỗi tàu cập bến cảng đều ghi nhận mã tàu, tên tàu, tên thuyền trưởng, nước sở hữu, loại tàu, trọng tải, v.v... Mỗi con tàu có thể chở nhiều loại hàng hoá và được chuyên chở trong

những loại kiện hàng khác nhau (container, lưới bao bì, thùng gỗ, v.v...). Mỗi con tàu đều có nơi đi (có thể là cảng Đà Nẵng), nơi đến và được ghi nhận ngày đi, ngày đến cảng Đà Nẵng.

Thời gian bốc hàng lên hoặc xuống tàu được ghi nhận dự kiến trong hợp đồng bốc hàng và ảnh hưởng đến việc thanh toán thực tế trong bản thanh lý hợp đồng bốc hàng. Việc bốc hàng do các đội bốc dỡ của cảng Đà Nẵng đảm nhiệm. Mỗi đội bốc dỡ được quản lý bởi tên đội, tên người phụ trách đội, số lượng công nhân trong đội. Mỗi con tàu có thể do nhiều đội bốc dỡ bốc hàng. Mỗi đội có thể tham gia bốc dỡ cho nhiều con tàu.

### 3. Quản lý chấm thi

Để quản lý việc chấm bài thi tập trung một số môn học cho các trường thành viên của Đại học Đà Nẵng, hàng năm, Ban Đào tạo và CTCT yêu cầu mỗi cán bộ giảng dạy đến Đại học Đà Nẵng chấm thi phải điền vào một phiếu như sau :

ĐẠI HỌC ĐÀ NẴNG Ban Đào tạo và CTCT		PHIẾU CHẤM THI	Số .....
Họ và tên cán bộ chấm thi	:	.....	
Chức danh	:	.....	
Tên Khoa công tác	:	.....	
Tên Trường công tác	:	.....	
Mã môn thi	:	.....	
Tên môn thi	:	.....	
Số đơn vị học trình	:	.....	
Kiểu thi (đánh dấu vào ô)	:	Viết <input type="checkbox"/>	Trắc nghiệm <input type="checkbox"/>
Ngày chấm thi	:	.....	
Tên lớp thi	:	.....	
Thuộc Trường	:	.....	
Số lượng sinh viên	:	.....	

Cho biết mỗi môn thi do nhiều người chấm, nhưng tối thiểu phải có hai người. Mỗi người chấm chỉ được chấm một môn. Mỗi Khoa có thể cử nhiều người đi chấm và có thể chấm thi cho nhiều Trường khác nhau. Các Trường khác nhau có thể có cùng một môn thi.

### 4. Quản lý thi sản phẩm phần mềm “tài năng trẻ Quốc Gia”

Trong một cuộc thi sản phẩm phần mềm để bình chọn ra tài năng trẻ Quốc Gia về lĩnh vực Công nghệ Phần mềm, ban Tổ chức cần đến những thông tin như sau :

Sản phẩm phần mềm dự thi : tên phần mềm, công cụ lập trình, hệ điều hành, phương tiện cài đặt (đĩa mềm, đĩa cứng, CD ROM, v.v...), dung lượng chiếm chỗ và thời gian kiến tạo phần mềm (số tháng ước chừng).

Tác giả của phần mềm dự thi : tên, họ, chủ trì (là người chủ trì hoặc thành viên cùng tham gia kiến tạo phần mềm), tuổi, phái (nam/nữ), địa chỉ, điện thoại, nghề nghiệp, nơi làm việc (tên trường học, cơ quan công tác hoặc nhà riêng) và khu vực (tỉnh, thành hoặc đang tên nước ngoài đang cư trú...).

Lĩnh vực liên quan của phần mềm dự thi : mã số, tên lĩnh vực, loại hình áp dụng (lý thuyết, nghiên cứu cơ bản, thực tiễn, đào tạo, v.v...)

Thành viên ban giám khảo : tên, họ, học hàm, nơi làm việc.

Mỗi người dự thi có thể gửi tham gia tối đa 2 sản phẩm phần mềm chưa từng được công bố. Với mỗi phần mềm dự thi có thể có nhiều tác giả, trong đó có một chủ trì, những người còn lại là thành viên. Ban giám khảo duyệt chấm lần lượt từng phần mềm và mỗi thành viên phải cho cho 3 loại điểm : điểm khoa học (về lĩnh vực liên quan), điểm ứng dụng và điểm giao tiếp.

## 5. Quản lý chấm thi năng khiếu

Đài Truyền hình TW tổ chức một cuộc thi năng khiếu giữa các Đội tuyển. Mỗi Đội tuyển tham gia đều phải thi hết cả 5 tiết mục được chấm điểm.

Mỗi thành viên của Hội đồng chấm thi được quản lý bởi tên họ, chức danh (tước hiệu), tên Cơ quan đang công tác và số điện thoại.

Để quản lý chấm và công bố kết quả, Ban Tổ chức Đài Truyền hình yêu cầu mỗi thành viên của Hội đồng khi chấm thi phải điền vào một phiếu như sau :

ĐÀI TRUYỀN HÌNH TW			
Ban Tổ chức		PHIẾU CHẤM THI	Số .....
Tên Đội tuyển : .....			
Họ tên Đội trưởng : .....			
Số người tham gia : .....			
Mã	Tên tiết mục	Thể loại	Điểm
01			
02			
03			
04			
05			
Địa điểm, Ngày chấm thi : .....			
Trưởng Ban Tổ chức (ký)		Thành viên Hội đồng chấm (ký)	

## 6. Quản lý tiêu thụ sản phẩm công nghiệp tại một đại lý

Một đại lý nhận uỷ quyền tiêu thụ sản phẩm công nghiệp cho một số nhà máy, xí nghiệp trong nước và nước ngoài. Đại lý yêu cầu mỗi khách hàng bắt buộc phải làm thủ tục đăng ký trước một lần với đại lý để được mua hàng. Mỗi khi khách hàng đặt mua các loại sản phẩm, đại lý phải liên hệ với nơi cung cấp để thoả mãn yêu cầu.

Mỗi khách hàng được quản lý bởi mã số đăng ký, tên, họ, địa chỉ (của người đại diện cơ quan đến đặt mua hàng), điện thoại và tên cơ quan (nhà máy, xí nghiệp...). Mỗi khách hàng có thể đặt mua với số lượng tuỳ ý nhiều loại sản phẩm công nghiệp khác nhau, đồng thời yêu cầu thời gian cụ thể nhận được hàng sau khi thanh toán.

Nơi cung cấp sản phẩm công nghiệp được quản lý bởi tên cơ quan, điện thoại, người đại diện (tên, họ, điện thoại và e-mail). Nơi cung cấp có thể giao nhiều loại sản phẩm công nghiệp cho đại lý kèm theo giá gốc (giá đại lý mua vào). Tuỳ theo nơi cung cấp mà giá gốc của cùng một sản phẩm có thể khác nhau.

Mỗi sản phẩm công nghiệp có thể được đặt mua ở nhiều nơi cung cấp khác nhau và được ấn định một giá bán cho khách hàng, không phụ thuộc khách hàng đó là ai và số lượng đặt mua là bao nhiêu.

Thời gian và lý do nơi cung cấp giao hàng trễ hẹn hoặc do chính đại lý gây ra cho khách hàng đều phải được ghi nhận trong hệ thống quản lý.

## CHƯƠNG 6

# Xây dựng mô hình logic dữ liệu

---

Mục đích của chương này là tìm chọn một công cụ phần mềm phù hợp để thể hiện các cấu trúc dữ liệu đã định nghĩa sau khi xây dựng mô hình ý niệm dữ liệu (MHYNDL). Chú ý rằng các giải pháp lựa chọn ở đây vẫn chỉ dừng lại ở tiếp cận “cổ điển”, nghĩa là những công cụ thuộc thế hệ 3, thế hệ của lập trình cấu trúc.

Như đã biết, hiện nay, các ứng dụng thuộc lĩnh vực phân tích và thiết kế hệ thống thông tin quản lý đã bước sang giai đoạn “hướng đối tượng”, với các công cụ mạnh hơn, tiện lợi hơn, và tất nhiên, dễ sử dụng hơn, chẳng hạn UML.

### I. Chọn phần mềm

Những công cụ phần mềm phục vụ cài đặt và khai thác HTTT hiện nay có rất nhiều, đó là : Bảng tính điện tử (WorkSheets). Ví dụ Tableur, Lotus-123, Quatro Pro, Microsoft Excel...

Hệ quản lý tệp (FMS – Files Management System) bao gồm các ngôn ngữ lập trình cấp cao như C, Cobol, Pascal, Java, ...

Có thể coi hệ quản lý tệp là công cụ phần mềm thế hệ 3, ngôn ngữ máy là thế hệ 1 và hợp ngữ là thế hệ 2.

Hệ quản lý cơ sở dữ liệu quan hệ (RDBMS – Relational DataBase Management System), công cụ phần mềm thế hệ 4, cho phép thể hiện sự độc lập giữa dữ liệu và chương trình. Ví dụ FoxPro, Oracle, Lotus Notes, Paradox...

Mỗi công cụ đều có những đặc điểm và ích lợi riêng. Các yếu tố để so sánh là :

Cách thể hiện các cấu trúc dữ liệu (CTDL) và các ràng buộc toàn vẹn (CIF - CIM) đã được định nghĩa ở mức ý niệm.

Khối lượng dữ liệu xử lý được.

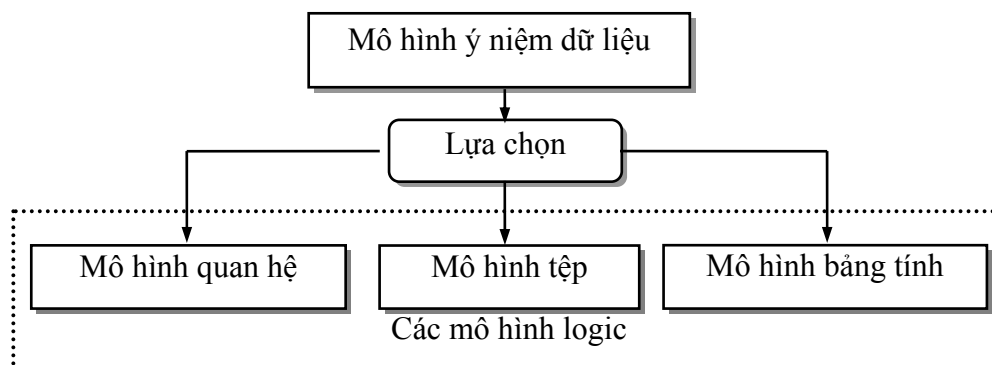
Độ phức tạp tính toán.

Cách thể hiện các quy tắc quản lý.

Tính độc lập giữa ứng dụng cần triển khai và công cụ phần mềm sử dụng.

Sử dụng các ngôn ngữ khác nhau (định nghĩa, thao tác, truy vấn và điều khiển).

Dưới đây là các khả năng lựa chọn một công cụ phần mềm thích hợp cho một ứng dụng phân tích và thiết kế hệ thống thông tin quản lý dựa trên một số mô hình đã xét :



Hình 6.12 Chọn một công cụ phần mềm

## I.1.Chuyển đổi các cấu trúc dữ liệu

### a) Bảng tính

Bảng tính không thích hợp với những CTDL phức tạp vừa khó cài đặt, vừa khó quản lý. Bảng tính không có những cấu trúc cơ sở như tệp hoặc các quan hệ. Mỗi vấn đề cần giải quyết được thể hiện trên một hoặc nhiều bảng, có cấu trúc là tổ hợp của các cấu trúc kiểu như CON, CHA-CON, BẢNG... Vì vậy bảng tính thường được dùng cho các mô hình dữ liệu đơn giản.

Mặt khác, bảng tính không đề cập đến các ràng buộc toàn vẹn. Tuy nhiên, nếu đưa được MHYNDL đang xét về sử dụng bảng tính sẽ làm cho việc cài đặt ứng dụng trở nên dễ dàng, vận hành và xử lý có hiệu quả.

### b) Hệ quản lý tệp (HQLT)

Về mặt lý thuyết, các HQLT không đưa ra hạn chế gì về độ phức tạp của CTDL. Tuy nhiên, một chương trình sử dụng CTDL phức tạp phải định nghĩa tất cả các tệp dữ liệu liên quan, lúc đó, sự vận hành của hệ thống sẽ trở nên nặng nề, tính trong sáng, dễ hiểu có thể bị hạn chế.

Phần lớn các ràng buộc toàn vẹn lại được đề cập đến trong các HQLT.

### c) Hệ quản lý cơ sở dữ liệu (HQLCSDL)

Các HQLCSDL cho phép biểu diễn bất kỳ một MHYNDL dù phức tạp đến đâu. Vì rằng cấu trúc bên trong của HQLCSDL không cứng nhắc như các HQLT. Mặt khác, các ràng buộc toàn vẹn được đề cập đến trong các HQLCSDL. Cả người thiết kế và NSD đều có thể sử dụng HQLCSDL một cách dễ dàng và hiệu quả.

## I.2.Khối lượng dữ liệu xử lý

### a) Bảng tính

Do mọi dữ liệu dùng cho xử lý đều có mặt trong bảng tính và xuất hiện trên màn hình, nên khối lượng dữ liệu không lớn. Số lượng các bảng tính phục vụ một ứng dụng nào đó còn phụ thuộc vào dung lượng của đĩa cứng.

Bảng tính thường được dùng cho những ứng dụng không đòi hỏi khối lượng dữ liệu xử lý lớn vì việc quản lý chúng sẽ trở nên nặng nề.

### b) Hệ quản lý tệp

HQLT thích hợp cho những ứng dụng có khối lượng dữ liệu lớn, đặc biệt là những ứng dụng đòi hỏi sử dụng thường xuyên những tệp thực thi, như soạn thảo hoá đơn, chứng từ thanh toán...

*c) Hệ quản lý cơ sở dữ liệu*

Về mặt lý thuyết, không có hạn chế gì về khối lượng dữ liệu. Tuy nhiên, cơ sở dữ liệu (CSDL) càng lớn thì thời gian xử lý tìm kiếm dữ liệu để trả lời các câu hỏi (thông qua các ngôn ngữ vấn tin) càng lớn. Để có được thời gian trả lời chấp nhận được, cần dự kiến trang thiết bị phần cứng và phần mềm phù hợp.

### **I.3.Mức độ tính toán**

*a) Bảng tính*

Bảng tính thích hợp cho những công việc có nhiều tính toán đồng thời, dễ dàng mô phỏng các hàm toán học phức tạp, các hàm tài chính và thống kê.

*b) Hệ quản lý tệp*

Tốc độ tính toán phụ thuộc vào ngôn ngữ lập trình đang sử dụng. Các hàm toán học phức tạp đòi hỏi phải lập trình và vẫn còn ách tắc. NSD không thể thay đổi, tối ưu được gì hơn với những phương tiện mà họ đang có.

Ví dụ, ngôn ngữ COBOL (COmmon Business Oriented Language, ra đời năm 1964) được sử dụng rộng rãi trong lĩnh vực thương mại do thích hợp với những tính toán đơn giản, dễ cập nhật.

*c) Hệ quản lý cơ sở dữ liệu*

Các ngôn ngữ vấn tin, như SQL (Structured Query Language), Oracle cho phép thực hiện những tính toán đơn giản. Nhưng để tính toán những công thức phức tạp hơn từ những dữ liệu được trích ra từ CSDL, cần sử dụng một công cụ sản sinh ứng dụng (AG – Application Generator).

### **I.4.Chuyển đổi các quy tắc quản lý**

Ngoài các quy tắc tính toán, các quy tắc quản lý đòi hỏi sử dụng những cấu trúc thuật toán phức tạp, được cấu thành từ các cấu trúc điều khiển cơ sở, như tuần tự, lựa chọn và lặp.

*a) Bảng tính*

Những quy tắc quản lý có cấu trúc thuật toán đơn giản dễ dàng chuyển đổi thành các công thức của bảng tính. Tuy nhiên, những cấu trúc thuật toán phức tạp, ví dụ như các xử lý lặp..., lại khó sử dụng trong bảng tính.

*b) Hệ quản lý tệp*

Các ngôn ngữ lập trình thích hợp cho mọi thuật toán, dù phức tạp đến đâu.

*c) Hệ quản lý cơ sở dữ liệu*

Bộ sản sinh ứng dụng GA có mặt trong một HQLCSDL dùng để chuyển đổi các quy tắc quản lý thành các cấu trúc thuật toán. Đối với những trường hợp đơn giản, GA được dùng tương tự như một bảng tính. Đối với những trường hợp phức tạp hơn, cần phải sử dụng một ngôn ngữ lập trình đủ mạnh và có cấu trúc, ví dụ ngôn ngữ Pal trong Paradox.

## **I.5. Tính độc lập của các ứng dụng**

### *a) Bảng tính*

Để phát triển một ứng dụng trên bảng tính, cần có phần mềm bảng tính và những kiến thức cơ sở để sử dụng. Thông thường, người thiết kế bảng tính đồng thời cũng là người sử dụng.

### *b) Hệ quản lý tệp*

Các HQLT hoàn toàn độc lập với các ứng dụng. Người thiết kế lập trình trên các ngôn ngữ bậc cao (như Cobol, Pascal, C...) và cho biên dịch thành ngôn ngữ máy để gọi chạy trực tiếp.

### *c) Hệ quản lý cơ sở dữ liệu*

Tồn tại hai khả năng :

Ứng dụng được thiết kế trên HQLCSDL và AG tương ứng. NSD biết cách “vào” CSDL và “gọi chạy” ứng dụng của mình. Lúc này NSD làm việc với một chương trình đã được biên dịch.

NSD không cần đến HQLCSDL mà cho chạy một chương trình độc lập với HQLCSDL.

## **I.6. Các kiểu ngôn ngữ khác nhau**

Người ta phân biệt bốn loại ngôn ngữ như sau :

Ngôn ngữ định nghĩa dữ liệu (DDL – Data Description Language) để chuyển đổi MHYNDL thành CTDL tương ứng với phần mềm sử dụng.

Ngôn ngữ thao tác dữ liệu (DML – Data Manipulation Language) để nhập, sửa đổi (cập nhật), thêm hoặc huỷ bỏ dữ liệu.

Ngôn ngữ truy vấn dữ liệu (DIL – Data Inquiry Language) để truy tìm xem xét dữ liệu thông qua các quy tắc quản lý.

Ngôn ngữ điều khiển dữ liệu (DCL – Data Control Language) để nhiều NSD cùng tham gia khai thác sử dụng CSDL tùy theo quyền hạn của mình.

### *a) Bảng tính*

Bảng tính không có những ngôn ngữ kể trên. Tùy theo mức độ sử dụng thành thạo bảng tính mà tự NSD tiến hành :

Thiết kế và tạo ra các mối quan hệ giữa các dữ liệu (vai trò DDL).

Thao tác trực tiếp trên các ô của bảng tính, tự thiết kế các công thức tính toán (vai trò DML).

Tìm kiếm dữ liệu trực tiếp trên bảng tính nhờ các hàm chuyên dụng có sẵn trong bảng tính, hoặc nhờ các lệnh vĩ mô (macro) tự lập lấy (vai trò DIL).

Quản lý dữ liệu và phân quyền truy xuất trên các bảng tính để đảm bảo tính toàn vẹn dữ liệu (vai trò DCL).

### *b) Hệ quản lý tệp*

Hệ quản lý tệp cũng không có những ngôn ngữ kể trên, hầu hết các ngôn ngữ lập trình đều có :

Phần mô tả, khai báo (định nghĩa) các biến, các hàm, các tệp dữ liệu liên quan... có mặt ở đầu chương trình (vai trò DDL).

Phần thao tác, xử lý trong chương trình. Người thiết kế phải lường trước mọi khả năng cần thiết trong khi lập trình để đảm bảo tính chặt chẽ, đúng đắn và hợp lý khi sử dụng. Đây là một hệ thống nặng nề, chắc chắn nhưng phiền phức khi cần sửa đổi (vai trò DML).

Việc truy tìm dữ liệu cũng được đảm bảo bởi chương trình (vai trò DIL).

HQLT cho phép phân loại NSD theo quyền hạn trách nhiệm của họ nhờ các mật mã (keyword). Hệ thống tuy nặng nề, nhưng đảm bảo mức độ an toàn (vai trò DCL).

### c) Hệ quản lý cơ sở dữ liệu

Các DDL, DML, DIL và DCL có mặt trong các HQLCSDL, đơn giản và dễ sử dụng. Ví dụ, các ngôn ngữ như SQL, hay QBE (Query By Exemple), được cài vào Paradox, Oracle, FoxPro..., cho phép định nghĩa, thao tác, truy vấn và điều khiển khai thác dữ liệu một cách tương tác.

## I.7.Kết luận

Ba kiểu công cụ phần mềm vừa kể trên không có cùng đặc tính, khả năng và mục đích. Sau đây là bảng đánh giá tổng hợp (DL là “dữ liệu” được viết tắt):

	Bảng tính	HQLT	HQLCSDL
1.Cấu trúc dữ liệu	Chỉ thích hợp với những cấu trúc đơn giản (có vấn đề đối với các ràng buộc toàn vẹn)	Xử lý được các CTDL phức tạp nhưng thao tác nặng nề	Thích hợp với những cấu trúc phức tạp và các ràng buộc toàn vẹn
2.Khối lượng dữ liệu xử lý	Không nhiều quá	Khối lượng DL lớn	Không hạn chế khối lượng DL nhưng hạn chế thời gian trả lời
3.Mức độ tính toán	Thích hợp với tính toán lặp, chuyên nghiệp	Chức năng của ngôn ngữ lập trình	Khả năng hạn chế, trừ trường hợp có GA đi kèm
4.Chuyển đổi các quy tắc quản lý	Phù hợp với những QTQL tương đối đơn giản	Phù hợp với những QTQL phức tạp	GA đi kèm QLCSDL có thể xử lý các QTQL phức tạp
5.Tính độc lập ứng dụng	NSD phải có bảng tính và biết cách sử dụng	Hoàn toàn độc lập	Độc lập nếu có cá Run- Time
6.Kiểu ngôn ngữ	Hạn chế	Nhiều khả năng nhưng hệ thống nặng nề	Nhiều khả năng và mềm dẻo



## II. Chuyển đổi mô hình E-A về mô hình quan hệ

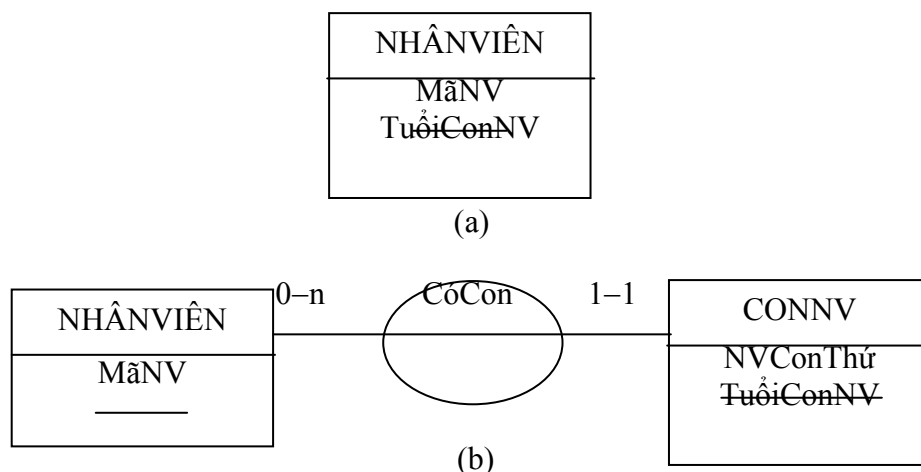
Dưới đây trình bày chín qui tắc chuyển đổi một sơ đồ E-A thành một lược đồ quan hệ. Trước khi chuyển đổi, sơ đồ E-A đã phải được thiết kế đúng và đầy đủ, không dư thừa các kết hợp, các kiểu đặc tính khóa và các bản số đều hữu hạn. Sơ đồ E-A chỉ chứa các kết hợp nhị nguyên mà ít nhất một trong hai bản số cực đại phải bằng 1. Người ta nói sơ đồ E-A đã được chuẩn hóa.

Từ nói sơ đồ E-A đã được chuẩn hóa, xây dựng lược đồ quan hệ bằng cách đặt tương ứng mỗi kiểu thực thể thành một sơ đồ của quan hệ. Thực tiễn cho thấy những người phân tích thiết kế hệ thống có kinh nghiệm có khả năng xây dựng được các mô hình ý niệm dữ liệu để chuyển thành các cơ sở dữ liệu ở dạng 3 NF.

### a) Quy tắc 1

Kiểm tra các kiểu đặc tính đều có kiểu đơn. Thay thế các kiểu đặc tính đa trị thành một kiểu thực thể và một kiểu kết hợp. Khi ca có thể đưa vào một kiểu đặc tính khóa cho kiểu thực thể mới và xác định lại các bản số của kết hợp mới này.

Trong kiểu thực thể NHÂNVIÊN, đặc tính TuổiConNV là một danh sách tuổi của các con của nhân viên. Thay thế đặc tính này bởi kiểu thực thể CONNV. CONNV sẽ có các đặc tính TuổiConNV và NVConThứ (con thứ mấy). Khóa của CONNV là NVConThứ được xác định từ MãNV của NHÂNVIÊN và NVConThứ của nhân viên này.



### b) Quy tắc 2

Kiểm tra các kiểu đặc tính sơ cấp. Thay thế các kiểu đặc tính tổ hợp (aggregate) bởi các đặc tính mới là những thành phần đã tổ hợp thành kiểu đặc tính này.

Ví dụ kiểu đặc tính Địa chỉ thường được thay thế bởi danh sách các kiểu thuộc tính Phố (xã), Quận (huyện), ThànhPhố (tỉnh), QuốcGia.

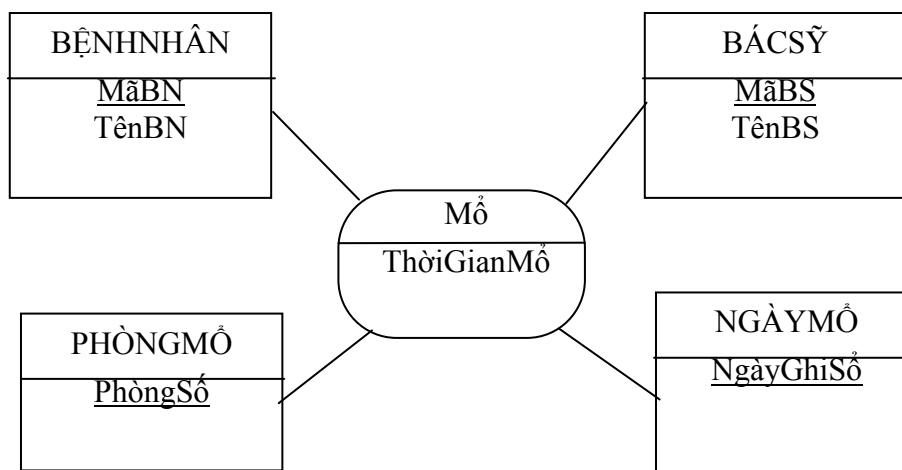
### c) Quy tắc 3

Kiểm tra các kiểu thực thể có các kiểu đặc tính khóa đơn. Đưa vào các dữ liệu đồng nghĩa (synonymous) cho các khóa tổ hợp. Ví dụ trong qui tắc 1, thay thế khóa tổ hợp mã NVConThứ thứ bởi dữ liệu đồng nghĩa MãConNV.

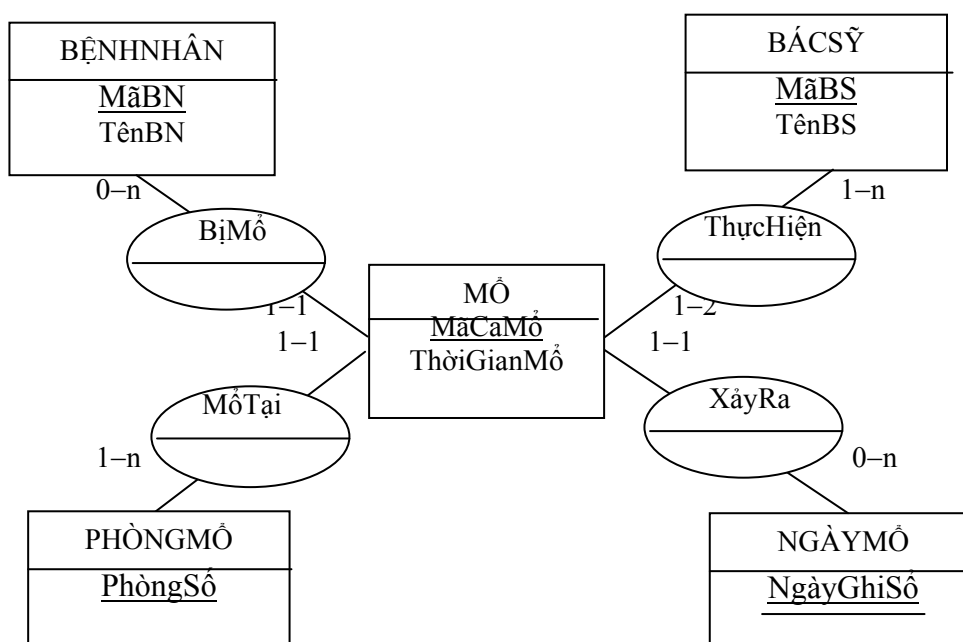
### d) Quy tắc 4

Thay thế tất cả các kiểu kết hợp không nhị nguyên bởi một kiểu thực thể và nhiều kiểu kết hợp. Tạo khóa mới cho kiểu thực thể mới và xác định lại các bản số cho các kiểu kết hợp mới.

Ví dụ :



Thay thế kiểu kết hợp Mô bởi kiểu thực thể mới MÔ và 4 kiểu kết hợp mới cùng các bản số tương ứng được tạo thành như sau :

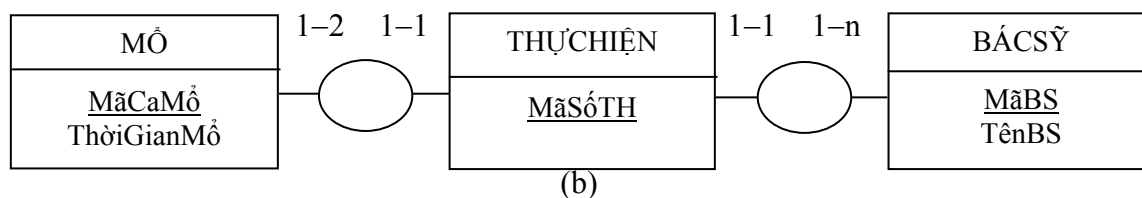
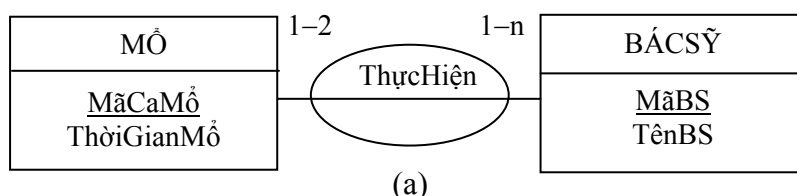


### e) Quy tắc 5

Đưa kết hợp không phân cấp (n:m) về kết hợp phân cấp (1:n) : Thay thế các kiểu kết hợp có bản số (\*-n) và (\*-n), nghĩa là quan hệ n:m, bởi một kiểu thực thể và hai kiểu kết hợp. Tạo khóa mới cho kiểu thực thể mới và xác định các bản số của hai kiểu kết hợp mới này.

Một trong hai bản số cực đại phải là 1, nghĩa là kiểu kết hợp phải tương ứng với một PTH giữa các khóa và hai thực thể.

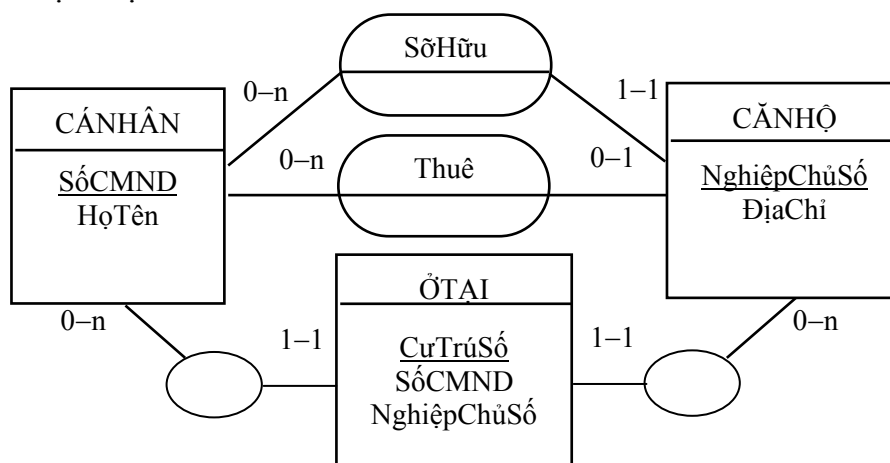
Trong ví dụ ở quy tắc 4, kết hợp TIỀN HÀNH phải được thay thế bởi một kiểu thực thể và hai kiểu kết hợp.



### f) Quy tắc 6

Trong trường hợp giữa hai kiểu thực thể có nhiều kiểu kết hợp, thêm vào trong các kiểu thực thể được tạo ra ở quy tắc 5, một kiểu thuộc tính là ghép của các khóa của các kiểu thực thể liên quan.

Ví dụ : trong ví dụ hình 52 chương 4, giữa hai kiểu thực thể CÁN HÂN và CĂN HỘ có các kiểu kết hợp Sở Hữu, Thuê và Ở Tại. Riêng kiểu kết hợp Ở Tại là không phân cấp. Áp dụng hai quy tắc 5 và 6 để nhận được sơ đồ như sau :



Trong mô hình dữ liệu, chủ sở hữu và người thuê nhà (CÁN HÂN) đều đồng nghĩa với số chứng minh nhân dân (SốCMND).

### g) Quy tắc 7

Vẽ đồ thị phụ thuộc hàm nối các khóa của các kiểu thực thể và tìm kiếm các PTH bắc cầu có thể tồn tại trong đồ thị. Loại bỏ các bắc cầu và thay đổi lại sơ đồ E-A nếu cần.

### h) Quy tắc 8

Tạo một lược đồ quan hệ cho mỗi một kiểu thực thể. Khóa của quan hệ là khóa của kiểu thực thể. Các thuộc tính của quan hệ tương ứng với các kiểu đặc tính của kiểu thực thể.

Như vậy, ví dụ cho ở qui tắc 6 cho ta các lược đồ quan hệ sau :

CÁNHÂN (SốCMND, HọTên, ...)

CÃNHỘ (NghịệpChủSố, ĐịaChỉ, ...)

ỞTAI (CưTrúSố, SốCMND, NghịệpChủSố, ...)

### **i) Qui tắc 9**

Loại bỏ các khóa đơn đồng nghĩa được tạo ra từ các kiểu khóa tổ hợp và kiểm tra quan hệ nhận được ở dạng 2 NF và 3 NF. Sử dụng phương pháp phân chia nếu cần cho những quan hệ như ở dạng chuẩn 2 NF và 3 NF, sau đó sửa đổi lại mô hình E-A.

## **III.Sử dụng các ngôn ngữ lập trình**

Các ngôn ngữ lập trình hay hệ quản lý tệp là công cụ phần mềm thể hệ ba.

Hệ quản lý tệp đảm đương mối liên hệ giữa các tệp dữ liệu và cách tổ chức chúng trên một thiết bị nhớ (mức vật lý) trong hệ điều hành đang sử dụng. Các tệp dữ liệu có thể được tạo ra bởi NSD, bởi chương trình hay bởi một HQLCSDL và/hoặc bởi một công cụ tạo sinh ứng dụng (mức logic),.

Ví dụ về “Khu du lịch Non nước” đã xét ở chương trước có thể được phát triển trong một HQLCSDL quan hệ. Tuy nhiên, do MHYNDL đã xây dựng trước đây không quá phức tạp, khối lượng dữ liệu không quá lớn, các chức năng cần xử lý không quá nhiều và tương đối đơn giản, ta có thể sử dụng một hệ quản lý tệp.

Ta chọn FoxPro, do đây là một phần mềm (phổ biến ở Việt nam) nằm giữa các ngôn ngữ thể hệ ba và các HQLCSDL quan hệ thuộc thể hệ 4. Sử dụng FoxPro, ta thấy được tính tuần tự có chỉ mục của hệ quản lý tệp này, tương tự DBase.

### **III.1.Chuyển đổi MHYNDL thành mô hình logic dữ liệu**

Trước khi chuyển đổi MHYNDL thành mô hình logic dữ liệu (MHLGDL), ta cần tìm hiểu cách thể hiện các tệp dữ liệu của FoxPro.

#### **III.1.1. Các tệp dữ liệu của FoxPro**

Mỗi tệp dữ liệu (DBF – DataBase File) gồm hai yếu tố : bản thân DBF và một (hoặc nhiều) tệp chỉ mục (Index File) tương ứng. Tệp chỉ mục không là tất yếu, cũng không bắt buộc phải có. Nếu như tệp chỉ mục không được tạo ra thì bản thân DBF được tổ chức theo kiểu tuần tự.

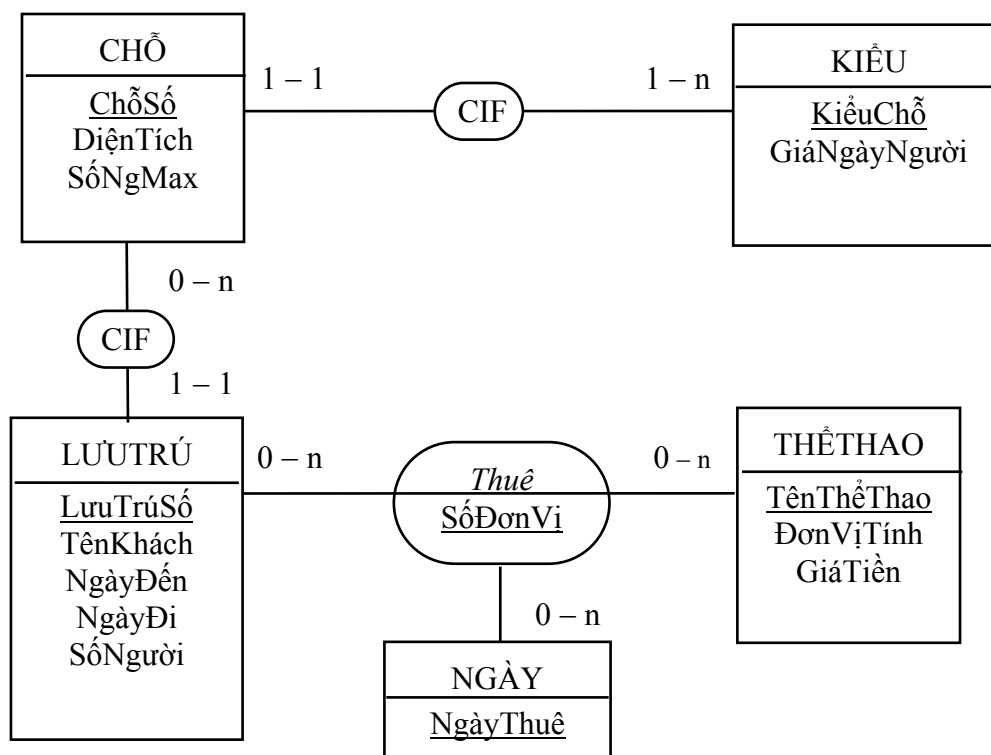
Có hai kiểu chỉ mục thường sử dụng :

*Chỉ mục đơn* hay chuẩn (standard index), tương tự khái niệm *khoá* : ứng với một giá trị chỉ mục duy nhất thì xác định được giá trị của một dữ liệu khác.

*Chỉ mục kết hợp* (compound index) là ghép của nhiều chỉ mục đơn (đính mục).

Mỗi DBF gồm các bản ghi (Record). Mỗi bản ghi gồm các trường (Field) là dữ liệu cơ sở, không chứa các cấu trúc con. Các trường dữ liệu tạo thành bản ghi là các thuộc tính của một thực thể,. Mỗi thuộc tính là đích của một PTH mà chỉ mục đơn (hay khoá của thực thể) là nguồn.

Sau đây là một MHYNDL trước khi chuyển :

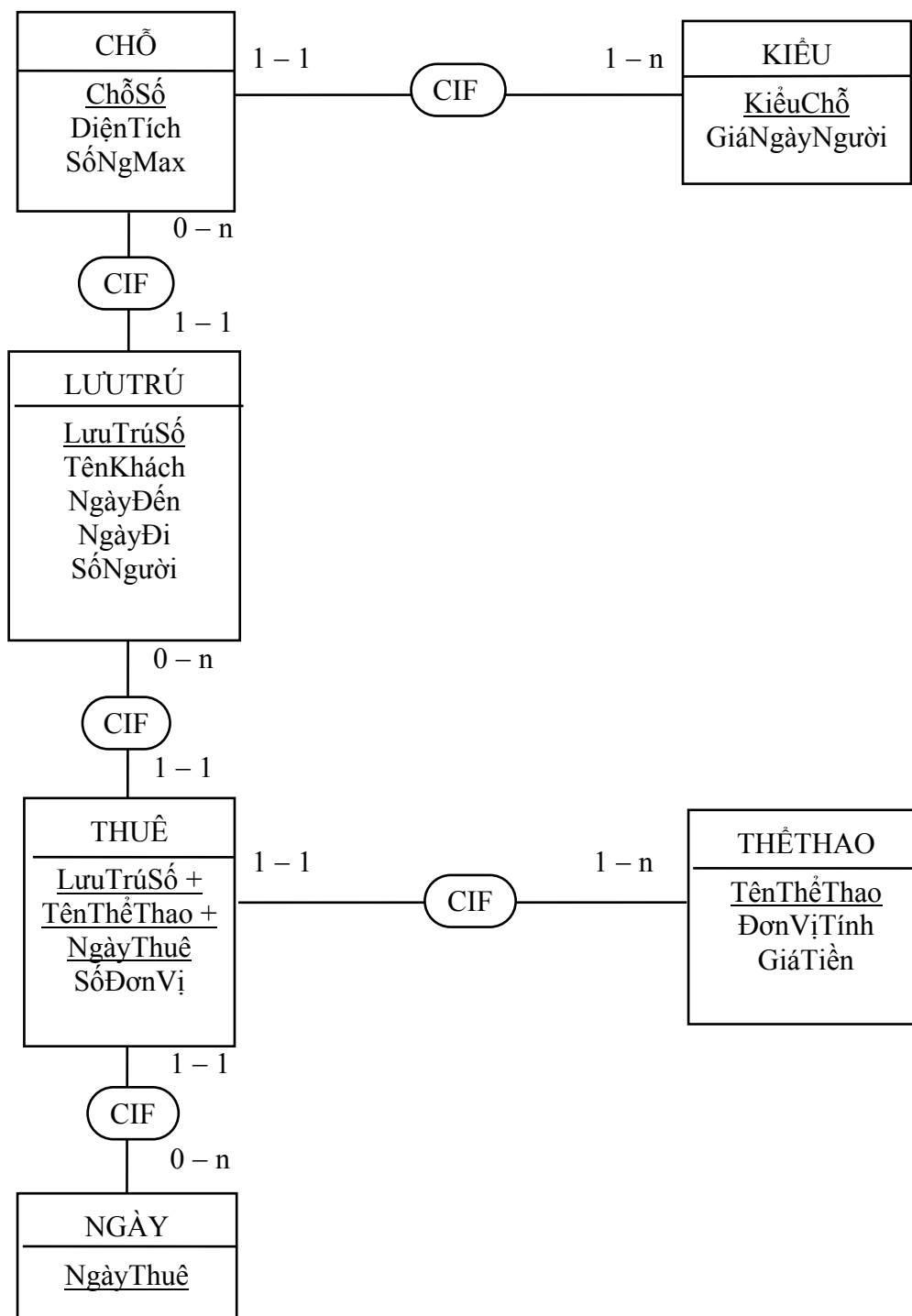


### III.1.2. Chuyển đổi MHYNDL $\longrightarrow$ MHLGDL

Để đơn giản hóa việc chuyển đổi, trước tiên ta chuyển MHYNDL đã cho thành mô hình nhị nguyên, một mô hình chỉ gồm các kết hợp phân cấp hai chiều.

#### a. Chuyển đổi thành mô hình nhị nguyên

Các kết hợp không phân cấp phải được đơn giản hóa (hay được cá thể hóa). Trong ví dụ **MHYNDL** trên đây, chỉ có kết hợp Thuê là không phân cấp, phải được đơn giản hóa. MHYNDL sau khi chuyển đổi thành mô hình nhị nguyên :



Như vậy, kết hợp Thuê đã được đơn giản hóa và được thay thế bởi :

Thực thể THUÊ có khóa là khóa của kết hợp Thuê, là do ghép các khóa của các thực thể tham gia vào kết hợp này.

Kết hợp Thuê là ba chiều nên phải tạo ra ba CIF giữa thực thể mới THUÊ và các thực thể LƯU TRÚ, THỂ THAO và NGÀY.

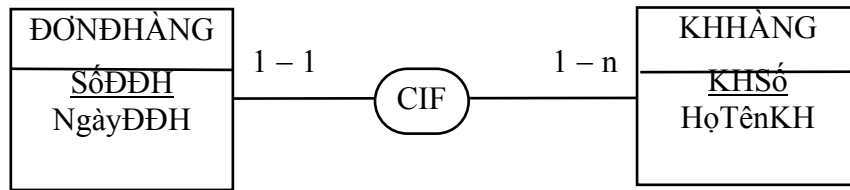
Ta nhận thấy rằng mô hình chỉ còn lại các CIF, và đó là mô hình nhị nguyên.

*b. Nguyên tắc chuyển đổi các thực thể thành các tệp*

Mỗi thực thể được tạo thành hai tệp dữ liệu là DBF và tệp chỉ mục của nó. Các thuộc tính của thực thể tạo thành các trường tin của DBF. Khóa có vai trò là giá trị chỉ mục đơn của tệp chỉ mục. Tuy nhiên, mô hình logic các tệp sẽ không đầy đủ nếu mối liên hệ giữa các thực thể (CIF) không được chuyển đổi.

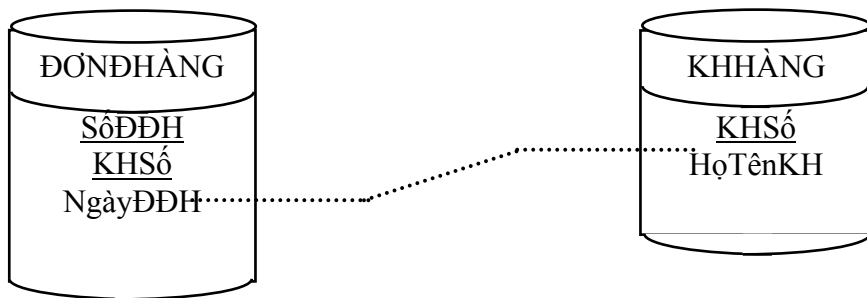
*c. Nguyên tắc chuyển đổi các mối liên hệ giữa các thực thể*

MHYNDL

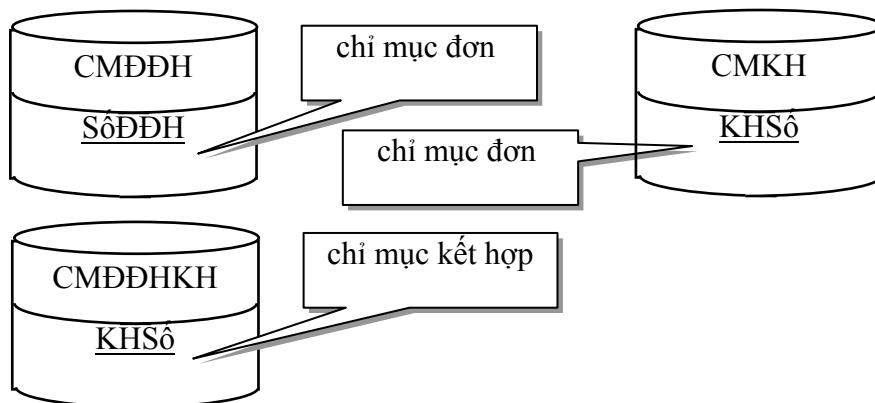


Mô hình logic các tệp

Các tệp cơ sở dữ liệu (DBF) :



Các tệp chỉ mục (IDX) :



Mỗi liên hệ CIF giữa hai thực thể được chuyển đổi như sau : tệp ĐƠN HÀNG được thêm dữ liệu KHSố là khóa của thực thể KH HÀNG. Trong trường hợp xây dựng tệp chỉ mục kết hợp CM ĐDH KH thì KHSố sẽ là khóa của tệp chỉ mục này, vì từ đó có thể :

– Tìm lại các thông tin về một khách hàng từ một đơn đặt hàng,

d. – Tìm lại tất cả các đơn đặt hàng của một khách hàng (từ khách hàng số).

e. Chuyển đổi MHYNDL thành MHLGDL các tệp

Thực thể KIẾU

DBF	Tệp chỉ mục
KIẾU	CMKIẾU
Các trường : <i>KiểuChỗ</i> <i>GiáNgàyNgười</i>	Khoá chỉ mục đơn : <i>KiểuChỗ</i>

Thực thể CHỖ

DBF	Tệp chỉ mục
CHỖ	CMCHỖ
Các trường : <i>ChỗSố</i> <i>KiểuChỗ</i> <i>DiệnTích</i> <i>SốNgMax</i>	Khoá chỉ mục đơn : <i>ChỗSố</i>
	CMCHỖKIẾU
	Khoá chỉ mục kết hợp : <i>KiểuChỗ</i>

Thực thể THUÊ

DBF	Tệp chỉ mục
THUÊ	CMTHUÊ
Các trường : <i>LưuTrúSố</i> <i>TênThẻThao</i> <i>NgàyThuê</i> <i>SốĐơnVị</i>	Khoá chỉ mục : <i>LưuTrúSố</i> <i>TênThẻThao</i> <i>NgàyThuê</i>   ghép các khoá

Khoá chỉ mục của tệp THUÊ do các khoá *LưuTrúSố*, *TênThẻThao* và *NgàyThuê* ghép lại để từ đó xác định giá trị của dữ liệu *SốĐơnVị*.

Thực thể NGÀY

Thực thể NGÀY không cần chuyển thành DBF vì chỉ có mỗi dữ liệu *NgàyThuê* nên *NgàyThuê* cũng đồng thời là khoá chỉ mục của nó.

### III.2. Hợp thức hóa mô hình dữ liệu bởi xử lý

Ta sẽ chứng tỏ rằng các tệp dữ liệu vừa xây dựng ở trên thỏa mãn các xử lý cho bài toán quản lý lưu trú trong ví dụ Khu du lịch Non Nước như sau :

- Đăng ký một lưu trú mới khi khách đến.
- Đăng ký thuê bao thẻ thao.
- Lập hóa đơn thanh toán khi khách đi.

Như vậy, mô hình thực thể - kết hợp khi chuyển thành MHLGDL các tệp sẽ được hợp thức hoá bởi các xử lý.

a. Đăng ký một lưu trú mới



Tập dữ liệu LU'UTRÚ.DBF được thêm một bản ghi mới :

```
SELECT 0
USE LU'UTRÚ INDEX CMLU'UTRÚ && Khoá LưuTrúSố
APPEND BLANK
SCATTER MEMVAR
*INPUT m.LưuTrúSố,m.ChỗSố,m.TênKhách,m.NgàyĐến,m.NgàyĐi,m.SốNgười
GATHER MEMVAR
USE
```

*b. Đăng ký thuê thể thao*

Tập dữ liệu THUÊ.DBF được thêm một bản ghi mới :

```
SELECT 0
USE THUÊ INDEX CMTHUÊ && Khoá LưuTrúSố+TênThểThao+NgàyThuê
APPEND BLANK
SCATTER MEMVAR
*INPUT m.LưuTrúSố,m.TênThểThao,m.NgàyThuê,m.SốĐơnVị
GATHER MEMVAR
USE
```

*c. Lập hóa đơn*

Đây là thủ tục phức tạp hơn cả, mọi tập dữ liệu và chỉ mục đều dùng đến :

```
STORE 0 TO m.LưuTrúSố, m.SốTiền
INPUT "Cho biết Lưu Trú Số:" TO m.LưuTrúSố
SELECT 0
USE LU'UTRÚ INDEX CMLU'UTRÚ && Khoá LưuTrúSố
SEEK m.LưuTrúSố
IF FOUND() && Tìm thấy bản ghi có khoá m.LưuTrúSố
    *Nhận các giá trị ChỗSố,TênKhách,NgàyĐến,NgàyĐi và SốNgười
    SCATTER MEMVAR
    SELECT 0
    USE CHỖ INDEX CMCHỖ && Khoá ChỗSố
    SEEK m.ChỗSố
    IF FOUND() && Tìm thấy bản ghi có khoá m.LưuTrúSố
        SCATTER MEMVAR && Nhận giá trị m.KiểuChỗ
        SELECT 0
        USE KIỂU INDEX CMKIỂU && Khoá KiểuChỗ
        SCATTER MEMVAR && Nhận giá trị m.GiáNgàyNgười
        SELECT 0
        USE THUÊ
        *Tạo tập DBF trung gian khoá LưuTrúSố
        COPY TO THUÊTG.DBF FOR LưuTrúSố=m.LưuTrúSố
        *Mở tập THỂTHAO để lấy GiáTiền tính toán
        SELECT 0
        USE THỂTHAO INDEX CMTHỂTHAO && Khoá TênThểThao
        SELECT 0
        USE THUÊTG
        *Tạo tập chỉ mục trung gian khoá TênThểThao+NgàyThuê
        INDEX ON TênThểThao+NgàyThuê TO CMTHUÊTG
        SCAN
```

```

SELECT THẺTHAO
SEEK THUÊTG.TênThẻThao
m.SốTiền=m.SốTiền+THẺTHAO.GiáTiền*THUÊTG.SốĐơnVị
SELECT THUÊTG
ENDSCAN
*Tiếp tục tính số tiền lưu trú và tổng số tiền phải trả
*In hoá đơn thanh toán
ELSE
? “Không tìm thấy bản ghi có chỗ “+STR(m.ChỗSố,4)
ENDIF
ELSE
? “Không tìm thấy bản ghi có khoá “+STR(m.LưuTrúSố,4)
ENDIF
CLOSE DATABASES

```

## Bài tập chương 6

1. Làm lại các bài tập ở chương 5 với các yêu cầu sau đây :

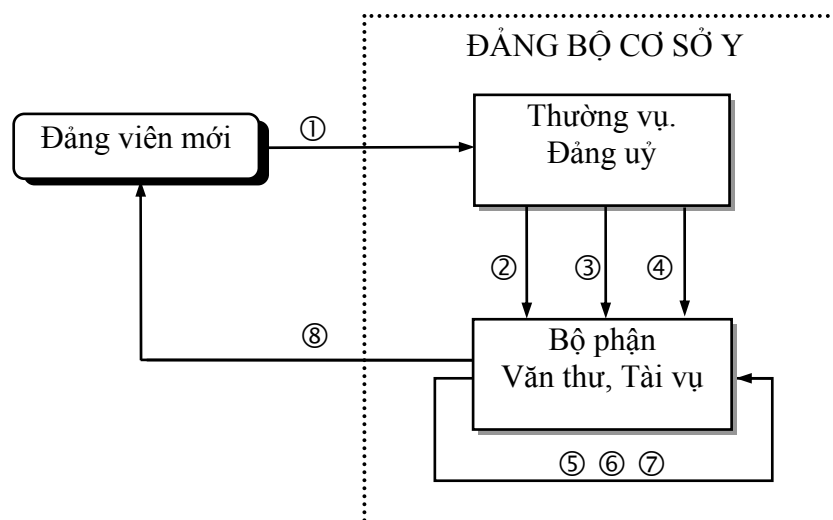
1. Xây dựng mô hình thực thể – kết hợp.
2. Chuyển đổi mô hình thực thể - kết hợp về mô hình logic dữ liệu
3. Mô tả các tệp CSDL và các quan hệ giữa chúng (nếu có), chỉ ra các tệp chủ (master), tệp tớ (slave) và các khoá sắp xếp tương ứng.
4. Cho 3 ví dụ về các câu hỏi vấn tin trên CSDL đã xây dựng.  
Có thể sử dụng lệnh SQL (select...) để viết câu trả lời.

### 2. Viết các lệnh FoxPro :

Để chuyển đổi các thực thể LU'UTRÚ và THỂTHAO trong ví dụ đã cho ở chương 6 thành các tệp CSDL (DBF) và tệp chỉ mục (IDX hoặc CDX) tương ứng.

### 3. Quản lý đảng viên tại đảng bộ cơ sở Y

a. Sơ đồ dòng thông tin



- ① Hồ sơ gia nhập Đảng (khoảng 10 mục tin).
- ② Hồ sơ gia nhập Đảng đã vào sổ Đảng + mã số Đảng viên + chức vụ Đảng + Đảng phí phải đóng hàng tháng.
- ③ Phiếu sửa đổi hồ sơ Đảng viên : thông tin mới về Đảng viên cần sửa đổi.
- ④ Phiếu xoá bỏ hồ sơ Đảng viên (do khai trừ, chuyển chuyên...) : mã số Đảng viên.
- ⑤ Danh sách Đảng viên trong Đảng bộ (thứ tự ABC).
- ⑥ Danh sách Ban Thường vụ Đảng uỷ.
- ⑦ Danh sách đóng Đảng phí của Đảng viên trong Đảng bộ (thứ tự ABC).
- ⑧ Giấy biên nhận đã đóng Đảng phí.

b. Yêu cầu công việc

Lập bảng các dòng.

Từ điển dữ liệu cùng các quy tắc quản lý.

Mô hình ý niệm dữ liệu - mô hình logic dữ liệu

Sơ đồ xử lý công việc quản lý Đảng viên.

#### 4. Quản lý thư viện

##### a. Mô tả

Để quản lý thư viện của một trường Đại học, người ta sử dụng những thông tin như sau :

<i>Tác phẩm</i>	Gồm các thông tin về tác phẩm số, tiêu đề, năm xuất bản, tên nhà xuất bản, đơn giá...
<i>Tác giả (của tác phẩm)</i>	Họ và tên tác giả, quốc tịch, nơi làm việc... Một tác phẩm có thể được viết bởi nhiều tác giả.
<i>Sinh viên mượn các tác phẩm này</i>	Mã sinh viên, họ và tên sinh viên, lớp, ngành chuyên môn...
<i>Mượn tác phẩm</i>	Ngày mượn, ngày trả, mã sinh viên mượn, tác phẩm số

Thư viện quy định rằng mỗi sinh viên chỉ có thể được mượn không quá 3 tác phẩm mỗi lần. Điều này có nghĩa nếu một sinh viên nào đó đã mượn và còn giữ lại một vài tác phẩm (chưa trả) thì anh ta chỉ được mượn tiếp cho đủ 3 tác phẩm. Còn nếu anh ta đã trả hết (hoặc chưa mượn) thì có thể mượn cùng lúc tối đa 3 tác phẩm.

##### b. Yêu cầu công việc

Anh (hay chị) hãy lập mô hình thực thể – quan hệ (có ghi bản số min – max tương ứng với CIF hoặc CIM) cho các trường hợp cụ thể sau :

Số lượng mỗi tác phẩm chỉ có một.

Số lượng mỗi tác phẩm có nhiều.

Thư viện muốn quản lý thêm nhà xuất bản (gồm tên và địa chỉ nhà xuất bản).

Lập ma trận phụ thuộc hàm để hợp thức hoá cho trường hợp 3 vừa nêu.

Chuyển mô hình dữ liệu trên về cấu trúc dữ liệu trong FoxPro.

## 5. Thực tập tốt nghiệp

### a. Mô tả

Tại một trường Đại học, Khoa Công nghệ Thông tin (CNTT) tổ chức cho sinh viên thực tập tốt nghiệp hàng năm trong thời gian 10 tuần tại các cơ sở sản xuất – nghiên cứu thuộc về lĩnh vực Tin học.

Khoa CNTT báo cho sinh viên danh sách những cơ quan đồng ý tiếp nhận sinh viên để họ có thể tự do lựa chọn nơi phù hợp với đề tài của họ. Sinh viên chọn xong, Khoa phân công giáo viên trực tiếp hướng dẫn sinh viên thực tập. Sau đó, Khoa đánh quyết định thực tập cho mỗi sinh viên.

Các giáo viên của Khoa sau khi được phân công sẽ xuống các cơ sở thực tế để kiểm tra việc thực tập tốt nghiệp của sinh viên. Mỗi giáo viên phải hướng dẫn ít nhất bốn sinh viên.

Để phân phối giáo viên cho phù hợp, Khoa CNTT cần biết những cơ sở thực tế mà giáo viên đã từng đến trong những kỳ thực tập của các năm trước.

Khoa CNTT cần có những thông tin như sau :

- Với mỗi cơ quan : tên cơ quan, địa chỉ, số điện thoại, loại cơ quan (nhà nước, tư nhân, liên doanh...), họ tên người phụ trách nhận quản lý sinh viên thực tập.

- Với mỗi sinh viên : mã số, họ tên, ngày sinh, giới tính, nhóm đề tài, và, những thông tin sau khi nhận quyết định thực tập : tên cơ quan sinh viên sẽ đến, ngày ký quyết định thực tập và họ tên giáo viên trực tiếp hướng dẫn.

- Với mỗi giáo viên : mã số, họ tên, giới tính, chức danh, số điện thoại, tên cơ quan đã từng đến hướng dẫn sinh viên và năm tương ứng (gần đây nhất).

Chú ý : Một cơ quan có thể tiếp nhận đồng thời nhiều sinh viên, tuy nhiên, mỗi sinh viên có thể có một giáo viên hướng dẫn riêng (các sinh viên tại cùng một cơ quan không nhất thiết có cùng giáo viên hướng dẫn).

### b. Yêu cầu công việc

Xây dựng mô hình thực thể – quan hệ (có ghi bản số min – max tương ứng với CIF hoặc CIM) biểu diễn việc tổ chức thực tập tốt nghiệp của Khoa CNTT tại một năm học đã cho.



# Mục lục

---

Mở đầu

<b>CHƯƠNG 1</b>	<b>KHÁI NIỆM VỀ HỆ THỐNG THÔNG TIN QUẢN LÝ .....</b>	<b>1</b>
I.	KHÁI NIỆM VỀ HỆ THỐNG .....	3
I.1.	Định nghĩa hệ thống .....	3
I.2.	Tính chất của hệ thống .....	2
I.3.	Các thành phần cơ bản của hệ thống .....	2
I.3.2.	Hành vi của hệ thống .....	3
I.3.3.	Mục tiêu của hệ thống .....	4
I.3.4.	Cấu trúc của hệ thống .....	4
I.4.	Phân loại hệ thống .....	5
I.5.	Nghiên cứu lý thuyết hệ thống .....	6
I.5.1.	Lý thuyết tổng quát về hệ thống .....	6
I.5.2.	Quan điểm nghiên cứu hệ thống .....	6
II.	XÍ NGHIỆP VÀ VAI TRÒ CỦA XÍ NGHIỆP TRONG NỀN KINH TẾ .....	8
II.1.	Xí nghiệp và các tổ chức bên trong .....	8
II.1.1.	Liên hệ giữa xí nghiệp với môi trường .....	9
II.1.2.	Phân tích các liên hệ với môi trường .....	9
II.2.	Hệ thống là tổ chức xí nghiệp .....	10
II.3.	Ba hệ thống của một tổ chức xí nghiệp .....	11
III.	HỆ THỐNG THÔNG TIN QUẢN LÝ (HTTTQL) .....	13
III.1.	Khái niệm HTTTQL .....	13
III.2.	Cấu trúc của HTTTQL .....	13
III.2.1.	Các phân hệ .....	13
III.2.2.	Dữ liệu .....	15
III.2.3.	Mô hình quản lý .....	16
III.2.4.	Quy tắc quản lý .....	17
III.3.	Vai trò và chất lượng của HTTTQL .....	17
III.4.	HTTTQL - công cụ điều phối và kiểm soát hệ thống .....	19
III.5.	Phân loại các hệ thống thông tin .....	21
<b>CHƯƠNG 2</b>	<b>CÁC PHƯƠNG PHÁP PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....</b>	<b>24</b>
I.	THẾ NÀO LÀ PHÂN TÍCH HỆ THỐNG ? .....	24
I.1.	Khái niệm .....	24
I.2.	Bản chất và yêu cầu của phân tích hệ thống .....	25
I.3.	Đánh giá các phương pháp .....	26
II.	MỘT SỐ PHƯƠNG PHÁP PTTKHT “CỖ ĐIỀN” .....	27
II.1.	Phương pháp SADT .....	28
II.2.	Phương pháp MERISE .....	30
II.3.	PTTKHT theo quan điểm ba trục tọa độ .....	32
II.3.1.	Mô hình phân tích và thiết kế HTTT .....	32

II.3.2.	Các giai đoạn phân tích và thiết kế hệ thống .....	34
II.3.3.	Tiếp cận ba mức .....	37
III.	PHƯƠNG PHÁP PHÂN TÍCH THIẾT KẾ HƯỚNG ĐỐI TƯỢNG, UML .....	39
<b>CHƯƠNG 3</b>	<b>PHÂN TÍCH HIỆN TRẠNG .....</b>	<b>42</b>
I.	PHƯƠNG PHÁP PHÒNG VẤN (INTERVIEW) .....	42
I.1.	<i>Nguyên lý của phương pháp</i> .....	42
I.2.	<i>Phân tích hiện trạng</i> .....	43
I.3.	<i>Phòng vấn lãnh đạo</i> .....	44
I.4.	<i>Phòng vấn các vị trí làm việc</i> .....	44
I.5.	<i>Củng cố các phỏng vấn</i> .....	46
II.	TỔNG HỢP CÁC KẾT QUẢ PHÂN TÍCH HIỆN TRẠNG .....	48
II.1.	<i>Xác định các phân hệ</i> .....	48
II.2.	<i>Phân tích dữ liệu</i> .....	50
II.2.1.	Khái niệm về dữ liệu sơ cấp.....	50
II.2.2.	Thanh lọc dữ liệu .....	51
II.2.3.	Xây dựng từ điển dữ liệu .....	51
II.3.	<i>Sơ đồ dòng dữ liệu</i> .....	54
II.3.1.	Khái niệm.....	54
II.3.2.	Phân biệt DFD với sơ đồ khối .....	55
II.3.3.	Ví dụ : .....	55
II.3.4.	Xây dựng sơ đồ dòng dữ liệu.....	56
II.3.5.	Trình soạn thảo PPP DFD editor .....	58
III.	Ví Dụ : XÍ NGHIỆP ĐÓNG HỢP DANAFOOD .....	60
III.1.	<i>Mô tả hoạt động của xí nghiệp DanaFood</i> .....	60
III.2.	<i>Giao tiếp giữa cơ sở dữ liệu với người sử dụng</i> .....	61
III.3.	<i>Phân tích các dòng thông tin</i> .....	62
<b>CHƯƠNG 4</b>	<b>PHÂN TÍCH Ý NIỆM DỮ LIỆU VÀ CÁC PHƯƠNG PHÁP MÔ HÌNH HOÁ.....</b>	<b>69</b>
I.	MÔ HÌNH THỰC THỂ - KẾT HỢP.....	70
I.1.	<i>Khái niệm về mô hình thực thể - kết hợp</i> .....	70
I.1.1.	Khái niệm về thực thể .....	70
I.1.2.	Khái niệm về kết hợp .....	72
I.1.3.	16 khả năng của kiểu kết hợp nhị phân.....	75
I.1.4.	Các kiểu kết hợp .....	77
I.1.5.	Các thành phần của từ điển dữ liệu.....	79
I.2.	<i>Mô hình thực thể - kết hợp mở rộng</i> .....	79
I.3.	<i>Chuyển đổi các mô hình thực thể kết hợp</i> .....	81
II.	MÔ HÌNH QUAN HỆ.....	83
II.1.	<i>Các định nghĩa</i> .....	83
II.2.	<i>Phụ thuộc hàm</i> .....	86
II.2.1.	Khái niệm.....	86
II.2.2.	Các tính chất của phụ thuộc hàm .....	87
II.2.3.	Các loại hình của phụ thuộc hàm .....	88
II.2.4.	Đồ thị của các phụ thuộc hàm.....	88
II.3.	<i>Các dạng chuẩn của lược đồ quan hệ</i> .....	90
II.4.	<i>Ví dụ khu du lịch Non Nước</i> .....	91
II.4.1.	Giới thiệu cơ sở dữ liệu.....	91
II.4.2.	Quan sát dữ liệu .....	93

II.4.3.	Mô hình quan hệ tương ứng .....	95
II.4.4.	Mô hình thực thể – kết hợp .....	96
III.	CÁC CÔNG CỤ BIỂU DIỄN PTH CHO MÔ HÌNH E–A .....	98
III.1.1.	Ma trận các phụ thuộc hàm .....	98
III.1.2.	Đồ thị các PTH .....	100
III.2.	<i>Ví dụ ứng dụng phụ thuộc hàm</i> .....	101
III.2.1.	Ma trận PTH .....	101
III.2.2.	Ma trận rút gọn các PTH .....	104
III.2.3.	Các PTH không sơ cấp .....	105
III.2.4.	Kết luận .....	106
III.3.	<i>Đồ thị PTH biểu diễn CSDL của nhà máy đóng hộp DanaFood</i> .....	107
III.4.	<i>Chuyển đổi giữa mô hình dữ liệu và đồ thị PTH</i> .....	107
III.4.1.	PTH có nguồn là dữ liệu sơ cấp .....	107
III.4.2.	PTH sơ cấp giữa khoá và các dữ liệu sơ cấp .....	107
III.4.3.	PTH sơ cấp giữa các khoá .....	108
III.4.4.	PTH không sơ cấp .....	108
<b>CHƯƠNG 5</b>	<b>MÔ HÌNH HÓA DỮ LIỆU BẰNG PHƯƠNG PHÁP LUI</b> .....	<b>111</b>
I.	CÁC CẤU TRÚC KIỂU .....	111
I.1.	<i>Cấu trúc kiểu PHIẾU</i> .....	111
I.2.	<i>Cấu trúc kiểu CHA-CON</i> .....	112
I.3.	<i>Cấu trúc kiểu BẢNG</i> .....	113
I.4.	<i>Cấu trúc kiểu HOẠCH ĐỊNH</i> .....	115
I.5.	<i>Cấu trúc kiểu CÓ-KHÔNG</i> .....	116
I.6.	<i>Cấu trúc kiểu PHÁN XẠ</i> .....	117
I.7.	<i>Cấu trúc kiểu BÌNH ĐẲNG</i> .....	117
I.8.	<i>Cấu trúc kiểu THỪA KẾ</i> .....	118
I.9.	<i>Cấu trúc kiểu KẾT TỤ</i> .....	119
II.	ỨNG DỤNG PHƯƠNG PHÁP TỪ TRÊN XUỐNG .....	121
II.1.	<i>Giới thiệu công ty xây dựng nhà ở BKCO</i> .....	121
II.1.1.	Các quy tắc quản lý .....	121
II.1.2.	Hồ sơ .....	121
II.1.3.	Nghiên cứu các cấu trúc kiểu .....	123
II.1.4.	Xem xét các quy tắc quản lý .....	125
II.1.5.	Hỗn hợp các cấu trúc kiểu .....	127
II.2.	<i>Hợp thức hoá mô hình ý niệm dữ liệu</i> .....	134
II.2.1.	Từ điển dữ liệu .....	134
II.2.2.	Ma trận rút gọn các PTH .....	135
II.2.3.	Quy tắc hợp thức hoá mô hình ý niệm dữ liệu .....	136
II.3.	<i>Ràng buộc toàn vẹn</i> .....	136
II.4.	<i>Ví dụ : Bài toán quản lý du lịch</i> .....	138
II.5.	<i>Các ràng buộc toàn vẹn dữ liệu</i> .....	141
II.5.1.	Các ràng buộc tĩnh đối với các quan hệ .....	141
II.5.2.	Các ràng buộc đối với nhiều quan hệ .....	142
II.5.3.	Các ràng buộc toàn vẹn động .....	142
<b>CHƯƠNG 6</b>	<b>XÂY DỰNG MÔ HÌNH LOGIC DỮ LIỆU</b> .....	<b>146</b>
I.	CHỌN PHẦN MỀM .....	146
I.1.	<i>Chuyển đổi các cấu trúc dữ liệu</i> .....	147
I.2.	<i>Khối lượng dữ liệu xử lý</i> .....	147



I.3.	Mức độ tính toán.....	148
I.4.	Chuyển đổi các quy tắc quản lý.....	148
I.5.	Tính độc lập của các ứng dụng.....	149
I.6.	Các kiểu ngôn ngữ khác nhau.....	149
I.7.	Kết luận.....	150
II.	CHUYỂN ĐỔI MÔ HÌNH E–A VỀ MÔ HÌNH QUAN HỆ .....	151
III.	SỬ DỤNG CÁC NGÔN NGỮ LẬP TRÌNH.....	154
III.1.	Chuyển đổi MHYNDL thành mô hình logic dữ liệu.....	154
III.1.1.	Các tệp dữ liệu của FoxPro .....	154
III.1.2.	Chuyển đổi MHYNDL $\longrightarrow$ MHLGDL .....	155
III.2.	Hợp thức hóa mô hình dữ liệu bởi xử lý .....	158

## Tài liệu tham khảo

---

- [ 1] E. M. Bennatan, *Software Project Management : A Practitioner Approach*, McGraw-Hill. 1992.
- [ 2] **J. Claviez, Diriger un Projet Informatique, Édition J.C.I. Inc., Montréal (Québec) 1993.**
- [ 3] **A. Collongues, J. Hugues, B. Laroche, MERISE phương pháp thiết kế hệ thống thông tin Tin học hoá phục vụ quản lý doanh nghiệp, sách dịch. Nhà XB Khoa học Kỹ thuật, Hà nội 1994.**
- [ 4] **G. Devoivre, La vie d'un projet informatique en contexte industriel, Cours de 3ème année, ENSIMAG, 1996.**
- [ 5] O. Foucaut, O. Thiéry, K. Smáli, *Conception des Systèmes d'information et Programmation Événementielle*, InterEditions, Paris 1996.
- [ 6] **J. Gabay, Apprendre et Pratiquer MERISE, MASSON, Paris 1993.**
- [ 7] Harozyszkiewicz, *Introduction to System Analysis and Design*, Prentice Hall 1989.
- [ 8] H. Habrias, *Introduction à la Spécification*, MASSON, Paris 1993.
- [ 9] J. Hugues. *MERISE vers la Conduite de Projet*. DUNOD Informatique, 1990.
- [ 10] F. Jolivet, G. Reboul, *Informatique Appliquée à la Gestion*, DUNOD, Paris 1992.
- [ 11] Page-Jones, *The Practical Guide to Structured Systems Design*, Yourdon Press 1980.
- [ 12] Pressaman, *Software Engineering*, McGrawHill, 1982.
- [ 13] C. Smart, R. Sims, *Phân tích, thiết kế, cài đặt hệ thống thông tin quản lý*, Viện Công nghệ Thông tin biên dịch và xuất bản, Hà nội 1990.
- [ 14] Weinberg, *Structured Analysis*, Prentice Hall 1980.
- [ 15] Ngô Trung Việt (sách dịch), *Phân tích và Thiết kế Tin học - Hệ thống Quản lý, Kinh doanh, Nghiệp vụ*, Nhà XB Giao thông vận tải 1995.
- [ 16] Các tài liệu trên internet.