

Chương 3: MÔ HÌNH QUAN NIỆM CỦA HỆ THỐNG THÔNG TIN

3.1 Giới thiệu về mô hình quan niệm

Mô hình quan niệm của một hệ thống thông tin được thiết lập từ hai mô hình liên quan đến nhau là *mô hình quan niệm về dữ liệu* và *mô hình quan niệm về xử lý*.

Mô hình quan niệm về dữ liệu: là sự mô tả toàn bộ dữ liệu của hệ thống, những mô tả này độc lập với các lựa chọn môi trường cài đặt, là công cụ cho phép người phân tích thể hiện dữ liệu của hệ thống ở mức quan niệm. Mô hình có thể mô tả bằng ngôn ngữ tự nhiên hoặc bằng hình vẽ.

Mô hình quan niệm về xử lý: mô tả toàn bộ các quy tắc xử lý được áp dụng cho dữ liệu của hệ thống.

Mô hình quan niệm cũng là cơ sở để trao đổi giữa những người phân tích thiết kế hệ thống.

3.2 Mô hình thực thể-mối quan hệ (mô hình ER)

3.2.1 Ý nghĩa của mô hình

Mô hình ER do Peter Chen đề xuất năm 1976, được sử dụng rộng rãi từ năm 1988. ANSI đã chọn nó làm mô hình chuẩn cho IRDS. Mô hình ER là một cách để mô tả thế giới thực gắn gũi với quan niệm và cách nhìn nhận bình thường. Mô hình này là một mô tả logic chi tiết dữ liệu của một tổ chức hoặc một lĩnh vực nghiệp vụ, nó còn là công cụ để phân tích thông tin nghiệp vụ. Mô hình được sử dụng rộng rãi trong các phần mềm thiết kế như ER Designer CASE (Chen và Associates 1988), trong các phần mềm trợ giúp thiết kế bằng máy tính.

3.2.2 Các thành phần của mô hình ER

Mô hình ER có các thành phần cơ bản sau:

- Các tập thực thể
- Các mối quan hệ giữa các thực thể
- Các thuộc tính của các thực thể và các mối quan hệ

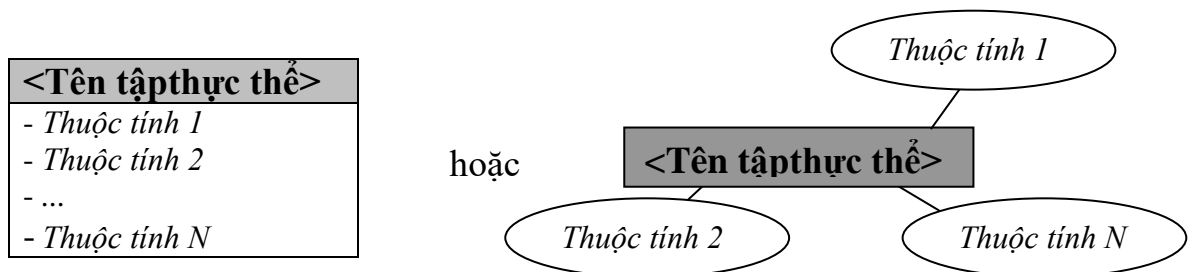
- Các mối quan hệ để mô tả kiểu kết nối giữa các thực thể (hoặc các bản số của các thực thể thông qua các mối quan hệ tương ứng)

3.2.1 Thực thể và tập thực thể

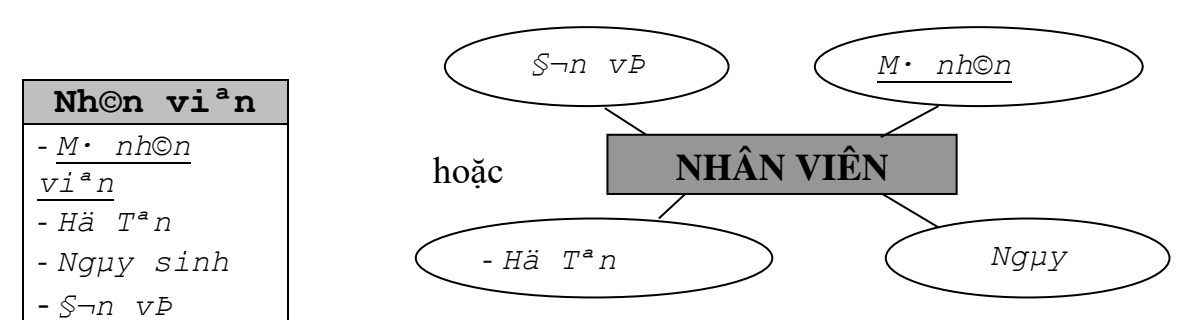
Một **tập thực thể** là mô hình của một lớp đối tượng cụ thể hoặc trừu tượng của thế giới thực. Mỗi thể hiện trong một tập thực thể được gọi là một **thực thể** hoặc cá thể (bản thể) của tập thực thể đó. Các đối tượng trong một tập thực thể tồn tại khách quan và độc lập tương đối lẫn nhau. Sự tồn tại của chúng không phụ thuộc vào hoạt động của hệ thống và chúng liên hệ với nhau thông qua tổ chức của hệ thống hoặc hoạt động của hệ thống.

Một thực thể được nhận diện bằng một số các đặc trưng của nó gọi là thuộc tính. Như vậy thuộc tính (Attribute) là các yếu tố thông tin cụ thể để nhận biết một tập thực thể.

Mỗi tập thực thể được đặc trưng bởi một *tên* và *danh sách các thuộc tính* của nó. Người ta dùng một trong các ký hiệu sau để mô tả một tập thực thể.



Ví dụ: Mã số nhân viên, họ tên, ngày sinh, đơn vị, nơi sinh là các yếu tố thông tin tạo thành tập thực thể NHÂN VIÊN.



3.2.2 Thuộc tính

Thuộc tính của một thực thể có thể phân thành các loại chủ yếu sau: thuộc tính đơn, thuộc tính lặp (đa trị), thuộc tính định danh.

a. Thuộc tính đơn

Thuộc tính đơn là thuộc tính mà giá trị của nó không thể phân tách được trong các xử lý theo một ý nghĩa tương đối nào đó.

Ví dụ: Thuộc tính HỌTÊN là thuộc tính đơn trong hệ thống thông tin “Quản lý nhân sự” bởi vì trong hệ thống này người ta không có nhu cầu tách thuộc tính HỌTÊN thành hai thuộc tính HỌLÓT và TÊN, tuy nhiên điều này không còn đúng nữa khi ở trong hệ thống thông tin “Quản lý Đào tạo”

b. Thuộc tính phức hợp

Thuộc tính phức hợp là thuộc tính được tạo từ những thuộc tính đơn khác nhau.

Ví dụ: Thuộc tính Ngày sinh là gộp của 3 thuộc tính ngày, tháng và năm sinh. Thuộc tính HỌTÊN được tạo từ hai thuộc tính HỌLÓT và TÊN

c. Thuộc tính lặp (đa trị): thuộc tính có thể nhận nhiều hơn một giá trị đối với mỗi thực thể.

Ví dụ: KỸNĂNG, TĐỘNGNGŨ là các thuộc tính lặp trong tập thực thể NHÂNVIÊN vì mỗi nhân viên có thể có nhiều kỹ năng và trình độ ngoại ngữ khác nhau.

d. Thuộc tính định danh (khóa)

Thuộc tính định danh là một hoặc một số tối thiểu các thuộc tính của một tập thực thể mà giá trị của nó cho phép phân biệt các thực thể khác nhau trong tập thực thể. Trong một tập thực thể có thể có nhiều thuộc tính định danh khác nhau. Thông thường người ta chọn thuộc tính định danh là một thuộc tính đơn duy nhất.

Ví dụ: Trong tập thực thể NHÂNVIÊN thuộc tính MÃNV, SÓCMND là các thuộc tính có thể làm thuộc tính định danh.

Để tiện cho việc tổ chức dữ liệu và xử lý sau này, khi chọn thuộc tính định danh nên chú ý đến các yếu tố sau:

- Chọn định danh sao cho giá trị của nó không thay đổi trong suốt vòng đời của thực thể. Ví dụ, SỐCMND ít khi được chọn làm thuộc tính định danh vì mỗi nhân viên có thể có nhiều chứng minh nhân dân khác nhau và có thể không có chứng minh nhân dân.
- Chọn định danh phải bảo đảm giá trị của nó đối với thực thể thuộc tập thực thể phải khác rỗng (NOT NULL). Nếu định danh là hợp bởi một số thuộc tính khác nhau thì phải bảo đảm mỗi thuộc tính thành phần phải khác rỗng.
- Tránh sử dụng các định danh áp đặt mà cấu trúc của nó có thành phần chỉ sự phân loại, địa điểm.

3.3 Mối quan hệ giữa các tập thực thể

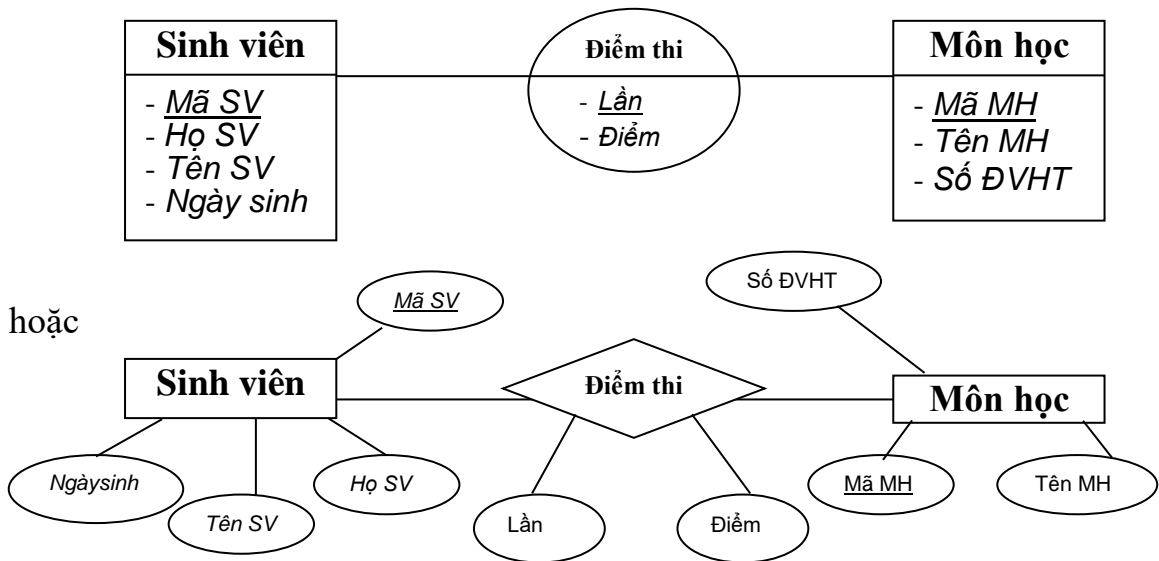
3.3.1 Mối quan hệ

Khái niệm tập thực thể với các thuộc tính không nói lên được mối liên quan giữa các tập thực thể với nhau. Mối quan hệ là sự mô tả sự liên hệ giữa các phần tử của các tập thực thể với nhau, chúng là các gắn kết các tập thực thể với nhau.

Chúng ta có thể diễn tả khái niệm mối quan hệ giữa các tập thực thể một cách hình thức như sau: Mối quan hệ R giữa các tập thực thể E_1, E_2, \dots, E_k là một tập con của tích Descartes $F_1 \times F_2 \times \dots \times F_n$, trong đó $F_i \in \{E_1, E_2, \dots, E_k\}$. Một thể hiện của mối quan hệ R là một tập các n -bộ (e_1, e_2, \dots, e_n) , trong đó $e_i \in F_i$ ($i=1 \dots n$). Nếu n -bộ (e_1, e_2, \dots, e_n) là một thể hiện của R thì ta nói rằng e_1, e_2, \dots, e_n có mối quan hệ R với nhau. Ta có thể phân loại các mối quan hệ giữa các tập thực thể như sau:

Mối quan hệ giữa các tập thực thể có thể là một *mối quan hệ sở hữu* hoặc *phụ thuộc* hoặc mô tả *sự tương tác giữa chúng*.

Một mối quan hệ có thể có thuộc tính riêng của nó. Để mô tả một mối quan hệ người ta dùng một hình ellip trong đó ghi tên của mối quan hệ và các thuộc tính riêng của nó nếu có.



Khi định nghĩa một mối quan hệ cần phải nêu ý nghĩa của nó.

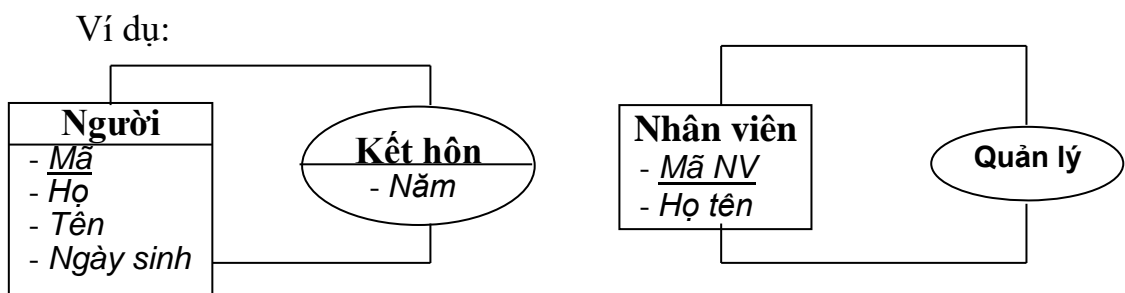
Ví dụ: $(e_1, e_2) \in \text{điểm thi}$ có ý nghĩa: *sinh viên e_1 thi một môn học e_2 lần thứ mấy và được bao nhiêu điểm.*

Một tập thực thể có thể tham gia nhiều mối quan hệ và giữa hai tập thực thể có thể có nhiều mối quan hệ khác nhau.

Ví dụ: Giữa hai tập thực thể **Sinh viên** và **Môn học** có hai mối quan hệ là ĐKMH và **Điểm thi**.

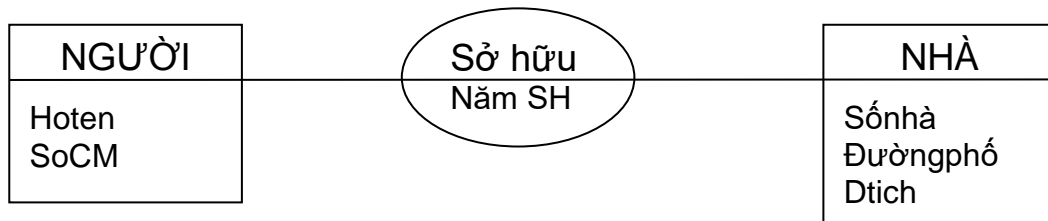
Chiều của mối quan hệ: là số tập thực thể tham gia vào mối quan hệ đó.

Mối quan hệ một chiều (đệ quy-phản xạ): mối quan hệ giữa các thực thể của cùng một tập thực thể. Ký hiệu:

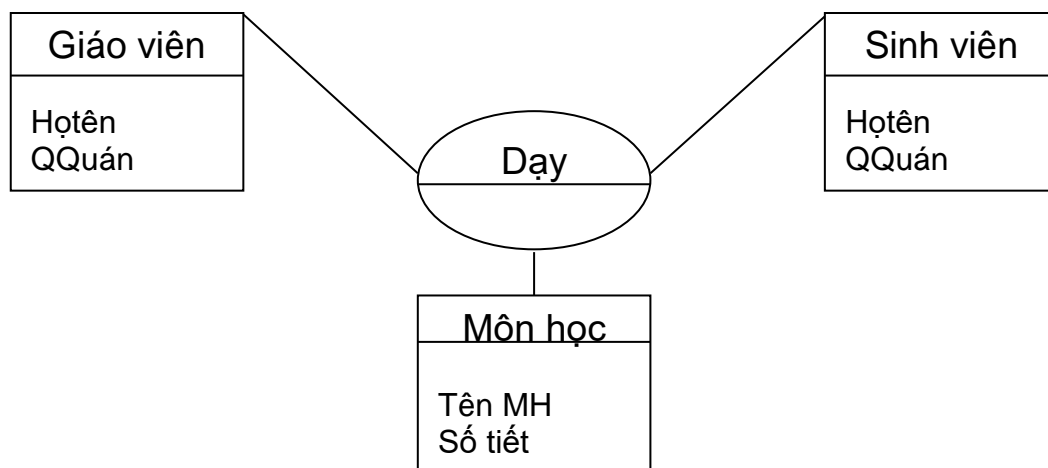


Mối quan hệ hai chiều: là sự kết nối giữa hai tập thực thể, còn gọi là mô hình nhị nguyên. Mối quan hệ này thường được sử dụng trong thực tế.

Ví dụ:



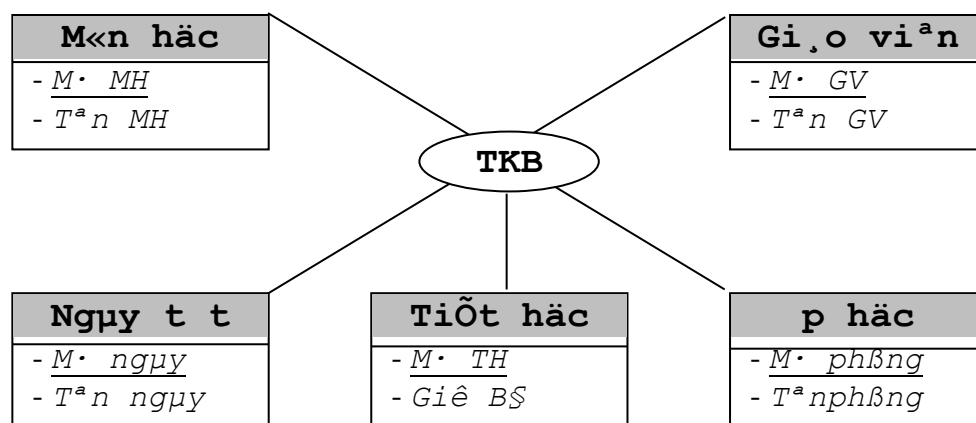
Mối quan hệ nhiều chiều: mối quan hệ có số tập thực thể tham gia lớn hơn 2, còn gọi là mô hình đa nguyên. Trong thực tế, người ta thường đưa các mối quan hệ nhiều chiều về mối quan hệ hai chiều.



Mối quan hệ ba chiều

Mối quan hệ Dạy có ý nghĩa:

$(a,b,c,d,e) \in \text{TKB}$: Giáo viên a Dạy Sinh viên b Môn học c tại Phòng học d vào tiết thứ e.



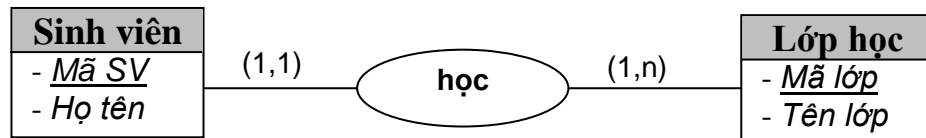
Mối quan hệ năm chiều

3.3.2 Bản số

Xác định mối quan hệ giữa các tập thực thể như ở trên chưa nói lên *độ phức tạp* của chúng. Chẳng hạn, khó có thể biết được mỗi thực thể của tập thực thể này có liên hệ với bao nhiêu thực thể của tập thực thể kia thông qua mối quan hệ. Để diễn tả tần suất xuất hiện của các phần tử của tập thực thể trong một mối quan hệ người ta dùng một khái niệm là bản số.

Bản số (*Cardinality*) là một cặp số nguyên (i,j) , chứa số tối thiểu và số tối đa trường hợp có thể có của các phần tử của tập thực thể tham gia vào mối quan hệ. Bản số của tập thực thể nào thì được ghi trên nhánh của tập thực thể đó. Nếu i,j nhận giá trị lớn hơn 1 thì quy ước thay chúng bởi ký tự n .

Ví dụ

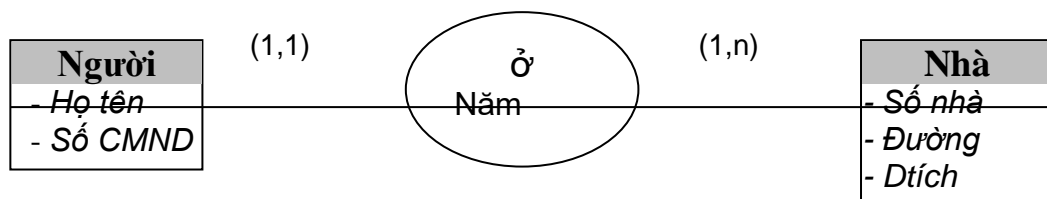


Bản số $(1,1)$: Một sinh viên học ít nhất là 1 lớp và nhiều nhất là 1 lớp.

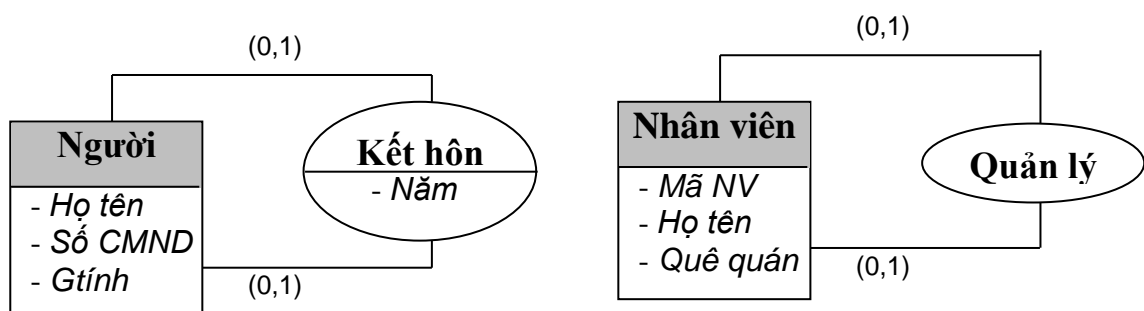
Bản số $(1,n)$: một lớp có ít nhất là 1 sinh viên và nhiều nhất là n sinh viên.

Các ví dụ:

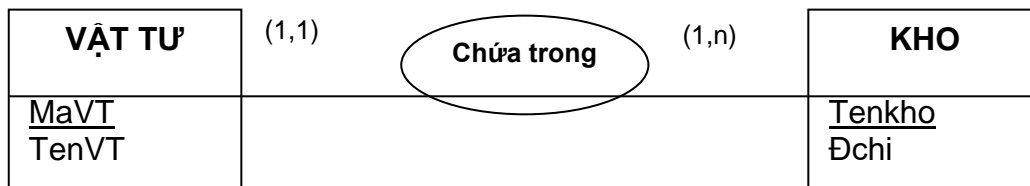
a. Giả sử một người *phải ở* và *chỉ ở* trong một nhà, khi đó bản số của các tập thực thể NGƯỜI và NHÀ qua mối quan hệ Ở là $(1,1)$ ----- $(1,n)$



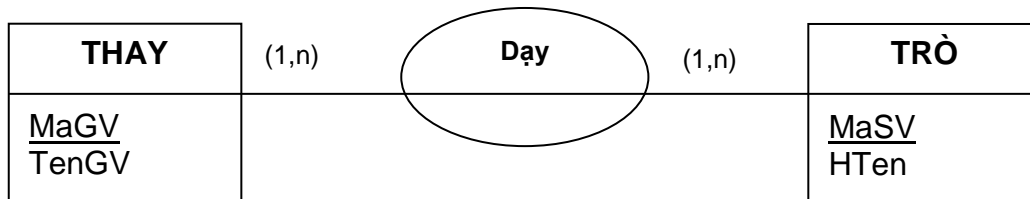
b. Bản số của mối quan hệ **Kết hôn** của công dân Việt nam



c. Giả sử mọi vật tư của công ty phải được chứa trong các kho hàng và một vật tư chỉ ở trong một kho mà thôi, khi đó bản số của các tập thực thể **VẬT TƯ** và **KHO** qua mối quan hệ **Chứa trong** là (1,1)------(1,n)



d. Bản số của các tập thực thể THẦY và TRÒ qua mối quan hệ **Dạy học**



Các trường hợp có thể xảy ra của các cặp bản số:

(1,1) — (1,1) hoặc (0,1) — (0,1) hoặc (0,1) — (1,1)

(1,1) — (1,n) hoặc (0,1) — (1,n) hoặc (1,1) — (0,n)

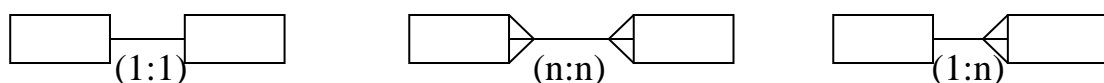
(1,n) — (1,n) hoặc (0,n) — (1,n) hoặc (1,1) — (0,n)

3.3.3 Bản số trực tiếp giữa các mối quan hệ

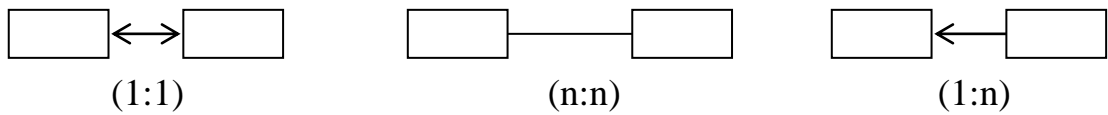
Trong một số phương pháp phân tích người ta không biểu diễn một cách rõ ràng bản số của một tập thực thể trong mỗi quan hệ mà chỉ biểu diễn bản số trực tiếp giữa hai tập thực thể. Bản số trực tiếp giữa hai tập thực thể:

Giả sử tập thực thể E_1 có bản số trong mỗi quan hệ là $(i_1:j_1)$; tập thực thể E_2 có bản số trong mỗi quan hệ là $(i_2:j_2)$. Khi đó bản số trực tiếp giữa hai tập thực thể E_1, E_2 là $(j_2:j_1)$.

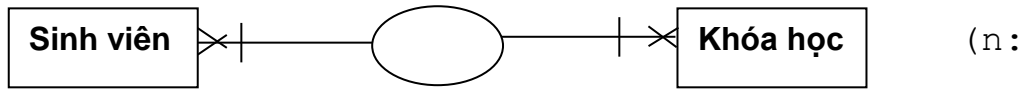
Người ta dùng các ký hiệu sau để mô tả bản số trực tiếp của hai tập thực thể:



hoặc

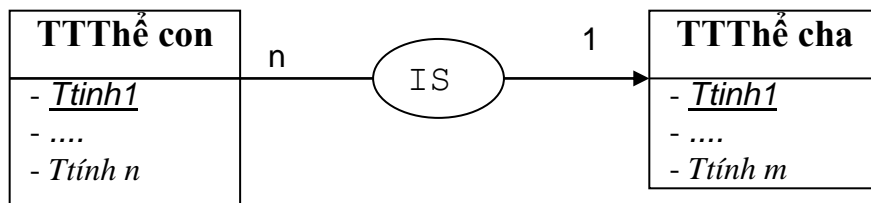


hoặc sử dụng các ký hiệu: \bigcirc , $|$, \rhd tương ứng với: 0, 1, n



Mối quan hệ ISA (cha-con): Cho hai tập thực thể A và B. Ta nói A có mối quan hệ ISA với B nếu mỗi thực thể trong A cũng là một thực thể trong B (còn gọi là A là con của B).

Ký hiệu



Ví dụ: Tập thực thể NHÂNVIÊN có tập thực thể ĐẢNGVIÊN là tập thực thể con. Vậy: ĐẢNGVIÊN ISA NHÂNVIÊN

3.3.4. Tách một mối quan hệ đa nguyên thành các mối quan hệ nhị nguyên

Trong mô hình ER người ta xem bậc (degree) của một mối quan hệ là số các tập thực thể tham gia vào mối quan hệ này. Chẳng hạn mối quan hệ bậc hai, còn được gọi là mối quan hệ nhị nguyên, là mối quan hệ giữa hai tập thực thể. Trên thực tế để đơn giản và tiện cho việc thiết kế cài đặt các CSDL, hầu hết các mối quan hệ đa nguyên đều được chuyển đổi thành các mối quan hệ nhị nguyên. Dựa vào khả năng chuyển đổi này của mối quan hệ đa nguyên, một số công cụ thiết kế CSDL chỉ cho phép sử dụng các mối quan hệ nhị nguyên. Cụ thể phương pháp chuyển đổi được thực hiện như sau.

Thuật toán 3.1. Tách một mối quan hệ đa nguyên thành các mối quan hệ nhị nguyên không kèm thuộc tính.

Input: Mỗi quan hệ R giữa các tập thực thể E_1, E_2, \dots, E_n là mỗi quan hệ bậc n ($n > 2$). Bản số đính kèm trên cung nối tập E_i và R là (\min_i, \max_i) , với $i=1, 2, \dots, n$.

Output: Các mỗi quan hệ nhị nguyên

Method: Thay R bởi n mỗi quan hệ R_i (không kèm thuộc tính) giữa các tập thực thể E_i với tập thực thể E (biểu diễn mỗi quan hệ R), với $i=1, 2, \dots, n$.

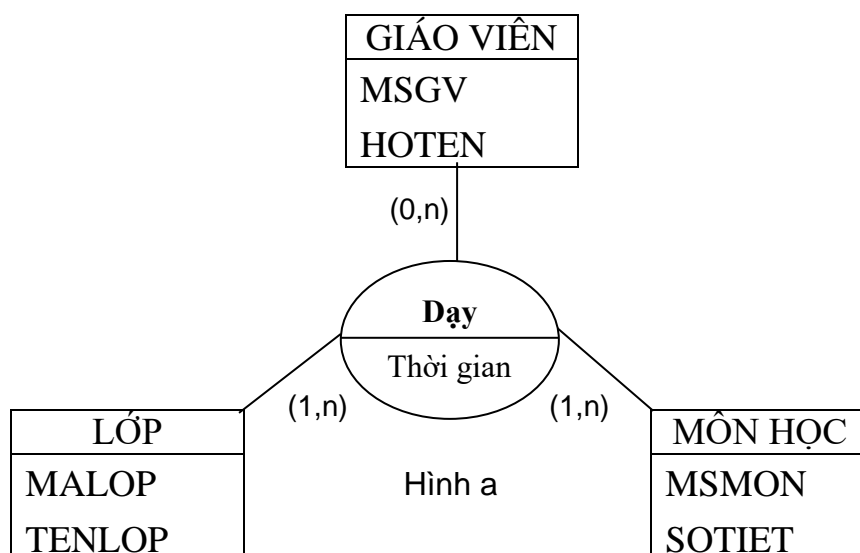
Procedure *Chuyen_doi_MQH_da_nguyen;*

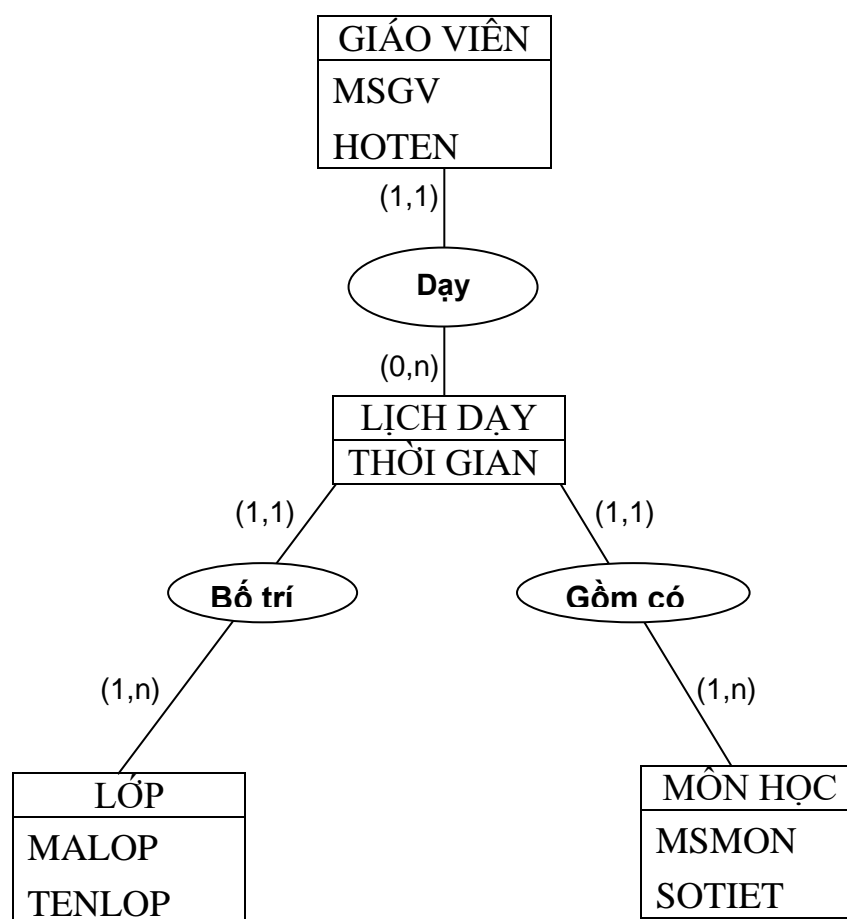
1. Mỗi quan hệ R được thay thế bởi tập thực thể E có cùng tập các thuộc tính;
2. **For** $i:=1$ **to** n **do**
3. Xây dựng mỗi quan hệ nhị nguyên R_i giữa tập thực thể E_i với tập thực thể E;
4. Gán bản số của cung nối R_i với E_i là (\min_i, \max_i) ;
5. Gán bản số của cung nối R_i và E là $(1, 1)$;
6. **Endfor**;

End *Chuyen_doi_MQH_da_nguyen;*

Lúc này ta hiểu rằng: mỗi thể hiện (e_1, e_2, \dots, e_n) thuộc R (mỗi quan hệ cũ), với $e_i \in E_i$ ($i = 1, 2, \dots, n$), sẽ cho tương ứng với một thực thể e thuộc E (tập thực thể mới tạo lập), sao cho các thể hiện (e, e_i) thuộc R_i (các mỗi quan hệ mới tạo lập).

Ví dụ. Hình sau đây trình bày một ví dụ về một mỗi quan hệ tam nguyên, và trình bày kết quả chuyển đổi thành ba mỗi quan hệ nhị nguyên.





Hình b

Hình 3.1. Minh họa việc chuyển đổi một mối quan hệ tam nguyên. (a) Ví dụ về một mối quan hệ tam nguyên. (b) Kết quả chuyển đổi thành ba mối quan hệ nhị nguyên.

Ví dụ trên cho thấy rằng, mối quan hệ DAY trong hình a được chuyển đổi thành tập thực thể LICHDAY đóng vai trò như một tập thực thể yếu không có khoá bộ phận và có ba mối quan hệ định danh là GIANG, BOTRI và GOMCO. Ba tập thực thể tham gia vào các mối quan hệ này là GIAOVIEN, LOP, MONHOC đều là các tập thực thể chủ của tập thực thể LICHDAY. Vì vậy, mỗi thực thể thuộc tập thực thể yếu LICHDAY được định danh bởi sự phối hợp của ba thực thể lần lượt thuộc ba tập thực thể chủ GIAOVIEN, LOP và MONHOC.

Rõ ràng, ta cũng có thể sử dụng thuật toán 3.1. để phân tách một mối quan hệ nhị nguyên (đặc biệt là mối quan hệ nhị nguyên có kèm thuộc tính) thành các mối quan hệ nhị nguyên không kèm thuộc tính.

3.3.5. Ràng buộc phụ thuộc hàm trên mỗi quan hệ đa nguyên

Xét mỗi quan hệ R là mỗi quan hệ đa nguyên bậc n ($n > 2$) giữa n tập thực thể E_1, E_2, \dots, E_n . Khác với mỗi quan hệ nhị nguyên, đối với mỗi quan hệ đa nguyên R , ngoài ràng buộc về bản số đính kèm trên mỗi cung nối tập E_i và R là (\min_i, \max_i) , với $i = 1, 2, \dots, n$; ta cần bổ sung một loại ràng buộc khác được gọi là ràng buộc phụ thuộc hàm giữa các tập thực thể tham gia vào mỗi quan hệ R (gọi tắt là ràng buộc phụ thuộc hàm trên R). Một cách hình thức, ta lần lượt xét các định nghĩa sau:

Thể hiện của một mỗi quan hệ: Gọi R là mỗi quan hệ đa nguyên bậc n ($n > 2$) giữa n tập thực thể E_1, E_2, \dots, E_n ; gọi tắt là mỗi quan hệ R trên $\{E_1, E_2, \dots, E_n\}$. Gọi e_1, e_2, \dots, e_n lần lượt là các thực thể thuộc các tập thực thể E_1, E_2, \dots, E_n . Khi đó, một bộ t gồm n thành phần (e_1, e_2, \dots, e_n) , ký hiệu $t = (e_1, e_2, \dots, e_n)$, được gọi là một thể hiện của mỗi quan hệ R , nếu và chỉ nếu (e_1, e_2, \dots, e_n) thuộc mỗi quan hệ R .

Chiều của một thể hiện: Xét mỗi quan hệ R trên $\Omega = \{E_1, E_2, \dots, E_n\}$, X là một tập con của Ω , và $t = (e_1, e_2, \dots, e_n) \in R$. Khi đó, chiều của t trên X , ký hiệu là $t[X]$, là bộ t chỉ chứa các thành phần tương ứng với các thực thể của các tập thực thể có trong X .

Phụ thuộc hàm giữa các tập thực thể: Xét mỗi quan hệ R trên Ω , gọi X và Y là các tập con của Ω . Khi đó, ta nói rằng R thoả phụ thuộc hàm $X \rightarrow Y$ (đọc là: X xác định Y , hoặc Y phụ thuộc hàm vào X) nếu và chỉ nếu: với mọi $t_1, t_2 \in R$ ta có: $t_1[X] = t_2[X]$ kéo theo $t_1[Y] = t_2[Y]$.

Một mỗi quan hệ R được gọi là thoả F - tập các phụ thuộc hàm trên R , nếu R thoả tất cả các phụ thuộc hàm trong F .

Từ những định nghĩa hình thức được nêu ở trên, ta có thể nhận thấy rằng, có sự tương ứng 1-1 giữa một số khái niệm này với các khái niệm của lý thuyết phụ thuộc hàm trong mô hình quan hệ. Cụ thể:

- Một mối quan hệ đa nguyên R tương ứng với một lược đồ quan hệ trên mô hình quan hệ. Và tập các thể hiện của mối quan hệ đa nguyên này tương ứng với một quan hệ trong lược đồ quan hệ trên. Tức là, một thể hiện của mối quan hệ R tương ứng với một bộ thuộc quan hệ đó.

- Các tập thực thể tham gia vào R tương ứng với các thuộc tính có trong một lược đồ quan hệ.

- Một ràng buộc phụ thuộc hàm trên một mối quan hệ R tương ứng với một ràng buộc phụ thuộc hàm trong một lược đồ quan hệ.

Thành phần của một bộ trong một quan hệ là giá trị của một thuộc tính, nhưng một thành phần của một thể hiện trong mối quan hệ R là một thực thể. Vì vậy, hai thực thể e và e' được gọi là bằng nhau, ký hiệu là $e = e'$, nếu e và e' cùng tham chiếu đến một đối tượng trong cùng một tập thực thể.

Từ việc so sánh này, cho phép một loạt các định nghĩa (bao đóng của tập các phụ thuộc hàm trên một mối quan hệ, bao đóng của tập các tập thực thể, khoá của một mối quan hệ, ...), các tính chất và các thuật toán (thuật toán tìm bao đóng của tập các tập thực thể, bài toán thành viên, thuật toán tìm khoá của một mối quan hệ, ...) được xây dựng một cách tương tự như đối với lý thuyết phụ thuộc hàm trên mô hình quan hệ. Chẳng hạn, chúng ta có thể định nghĩa khoá của một mối quan hệ đa nguyên như sau.

Khoá của một mối quan hệ đa nguyên: Xét mối quan hệ đa nguyên R trên Ω thoả tập phụ thuộc hàm F , gọi X là một tập con của Ω . Khi đó, X được gọi là *khóa* của R nếu thoả mãn hai điều kiện sau:

- (1). R thoả phụ thuộc hàm $X \rightarrow \Omega$, hay ta gọi X là một siêu khoá trên R .
- (2). Không tồn tại tập $X' \subset X$ (X' là tập con thực sự của X) thoả điều kiện trên.

Từ định nghĩa này, cho thấy rằng Ω là một siêu khoá của mối quan hệ R . Và thuật toán xác định một khoá của mối quan hệ được xây dựng như sau.

Thuật toán 1.2. Xác định một khoá của mối quan hệ R .

- + **Input:** Mỗi quan hệ R trên Ω thoả tập phụ thuộc hàm F
- + **Output:** Một khoá K của mỗi quan hệ R .
- + **Method:**

Function Key(Ω, F);

1. $K := \Omega$;
2. **for** mỗi tập thực thể E trong Ω **do**
3. **if** R thoả $K - \{E\} \rightarrow \Omega$ **then**
4. $K := K - \{E\}$
5. **endif**;
6. **endfor**;
7. **return** K ;

Xét mỗi quan hệ DAY giữa ba tập thực thể GIAOVIEN, MONHOC và LOP có ngữ nghĩa rằng: $(g, m, l) \in \text{DAY}$ nếu giáo viên g dạy môn m cho lớp l . Giả sử F chỉ có một phụ thuộc hàm $\{\text{MONHOC}, \text{LOP}\} \rightarrow \{\text{GIAOVIEN}\}$ để chỉ rằng một môn học của một lớp chỉ do một giáo viên phụ trách. Khi đó, theo thuật toán trên ta có $K = \{\text{MONHOC}, \text{LOP}\}$ là một khoá của mỗi quan hệ DAY.

Rõ ràng, ta có thể chỉ ra mối liên quan giữa các ràng buộc về bản số và ràng buộc phụ thuộc hàm trong mỗi quan hệ đa nguyên bởi các tính chất sau.

Tính chất 3.1: Xét mỗi quan hệ R trên $\Omega = \{E_1, E_2, \dots, E_n\}$, khi đó: chỉ số cực đại của bản số thuộc cung nối $E_k \in \Omega$ và R trên sơ đồ ER là bằng 1, khi và chỉ khi R thoả $E_k \rightarrow \Omega$ (hay $K = E_k$ là một khoá của R).

Tính chất 3.1 cho ta một tiêu chuẩn để xác định khoá của R dựa vào ràng buộc bản số. Dĩ nhiên tính chất này còn có thể áp dụng đối với mỗi quan hệ nhị nguyên (với $n = 2$). Dựa vào tính chất này, ta có thể chứng minh tính chất sau khi xét một mỗi quan hệ nhị nguyên.

Tính chất 3.2: Xét mỗi quan hệ nhị nguyên R giữa hai tập thực thể E_1 và E_2 , khi đó: mỗi quan hệ R là mỗi quan hệ nhiều-nhiều, khi và chỉ khi khoá của R là $\Omega = \{E_1, E_2\}$.

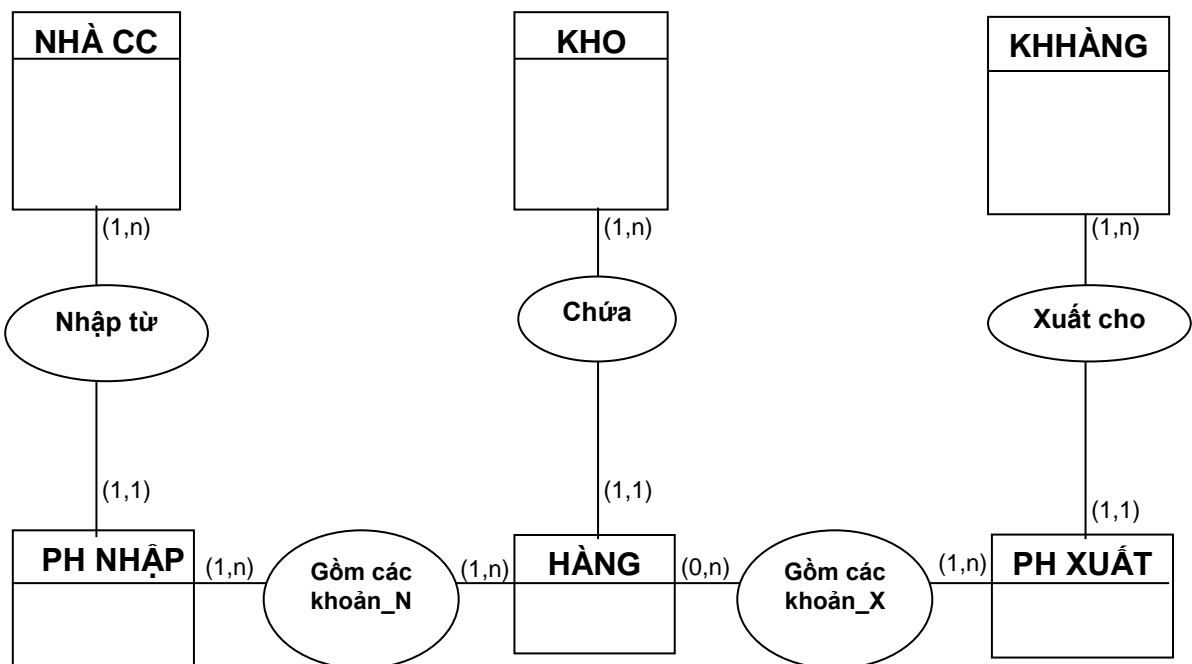
Tính chất 1.2. cho thấy rằng, đối với mỗi quan hệ nhị nguyên, ràng buộc phụ thuộc hàm là thật sự không cần thiết trong việc xác định khoá của mỗi quan hệ này.

Ngoài ra, liên quan đến việc phân tách một mối quan hệ đa nguyên trong một mô hình ER thành các mối quan hệ nhị nguyên, ta nhận thấy rằng, rõ ràng việc phân tách này có thể làm mất mát ngữ nghĩa của các ràng buộc phụ thuộc hàm giữa các tập thực thể tham gia vào mối quan hệ đa nguyên đó.

3.3.6 Mô hình thực thể-mối quan hệ (ER model)

Mô hình thực thể-mối quan hệ là mô hình liên hoàn các tập thực thể và các mối quan hệ trong hệ thống thông tin. Trên mô hình này sẽ thể hiện đầy đủ các tập thực thể và mối quan hệ giữa chúng trong hệ thống. Đây cũng chính là mô hình quan niệm về dữ liệu của hệ thống thông tin.

Ví dụ 1: Mô hình thực thể-mối quan hệ của HTTT "Quản lý Kho hàng"



Mô hình thực thể -mối quan hệ của HTTT "QL Kho hàng"

3.4 Một vài nhận xét để rà soát lại mô hình ER

3.4.1 Đối tượng nào có thể làm tập thực thể?

Một đối tượng có thể làm tập thực thể nếu nó được tạo thành từ một lớp các cá thể tương ứng. Ví dụ, tập thực thể SINHVIÊN được tạo từ các thực thể mà mỗi thực thể là một sinh viên.

3.4.2 *Yếu tố thông tin gì có thể làm thuộc tính cho một tập thực thể?*

Các thông tin đặc trưng để xác định các thực thể trong một tập thực thể đều có thể làm thuộc tính cho tập thực thể đó. Tuy nhiên cần phải chọn thông tin nào cần thiết và được sử dụng trong các xử lý. Để tránh nhầm lẫn khi xác định thuộc tính cho một tập thực thể, ta đặt hệ thống thông tin ở trạng thái tĩnh và xem thử thuộc tính đó có nguy cơ bị phá vỡ không. Nếu có thì thuộc tính đó là một tập thực thể hoặc là thuộc tính của một tập thực thể khác.

3.4.3 *Loại bỏ các thuộc tính vô nghĩa*

Loại bỏ các thông tin không bao giờ sử dụng đến. Ví dụ trong quản lý sinh viên thì thuộc tính anhem không cần.

3.4.4 *Tính độc lập của các thuộc tính*

Thuộc tính của một tập thực thể không được suy từ những thuộc tính khác của tập thực thể đó.

Ví dụ :

<table style="width: 100%; border-collapse: collapse;"> <tr style="background-color: #cccccc;"> <th style="padding: 2px 5px;">Hóa đơn</th> </tr> <tr> <td style="padding: 2px 5px;"> <ul style="list-style-type: none"> - Mã hóa đơn - Số lượng - Đơn giá - Thành tiền </td> </tr> </table>	Hóa đơn	<ul style="list-style-type: none"> - Mã hóa đơn - Số lượng - Đơn giá - Thành tiền 	chuyển thành	<table style="width: 100%; border-collapse: collapse;"> <tr style="background-color: #cccccc;"> <th style="padding: 2px 5px;">Hóa đơn</th> </tr> <tr> <td style="padding: 2px 5px;"> <ul style="list-style-type: none"> - Mã hóa đơn - Số lượng - Đơn giá </td> </tr> </table>	Hóa đơn	<ul style="list-style-type: none"> - Mã hóa đơn - Số lượng - Đơn giá
Hóa đơn						
<ul style="list-style-type: none"> - Mã hóa đơn - Số lượng - Đơn giá - Thành tiền 						
Hóa đơn						
<ul style="list-style-type: none"> - Mã hóa đơn - Số lượng - Đơn giá 						

Thuộc tính **Thành tiền** được tính toán từ hai Thuộc tính **Số lượng** và **Đơn giá**. Ta loại bỏ Thuộc tính **Thành tiền** khỏi tập thực thể **hóa đơn**.

3.4.5 *Xác định thuộc tính khóa*

Trong mỗi tập thực thể nên chọn khóa chỉ có một thuộc tính để tiện việc xử lý. Nếu trong tập thực thể không có một thuộc tính nào để làm khóa thì nên áp đặt một thuộc tính bên ngoài để làm khóa.

Thông thường thuộc tính áp đặt này có dạng: **Mã + <Tên tập thực thể>**,

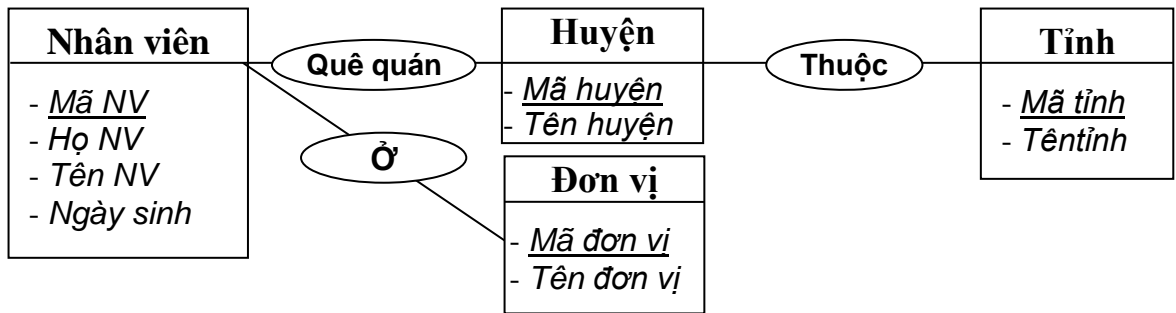
Ví dụ : Trong tập thực thể NHÂN VIÊN không tồn tại một thuộc tính nào để làm khóa, ta đưa thêm thuộc tính **Mã nhân viên** làm khóa.

Trong biểu diễn tập thực thể, những thuộc tính khóa được gạch dưới.

3.4.6 Tách thuộc tính có dung lượng lớn

Nếu một thuộc tính của tập thực thể có nhiều giá trị, mỗi giá trị chiếm một dung lượng lớn và lặp lại nhiều lần thì nên tách thành một tập thực thể riêng có tên là <tên thuộc tính> và có hai thuộc tính là: **Mã** + <tên thuộc tính> và **Tên** + <tên thuộc tính>.

Ví dụ : Thuộc tính Đơn vị, Nơi sinh trong tập thực thể **Nhân viên** với Nơi sinh bao gồm Huyện và Tỉnh được tách thành các tập thực thể riêng như sau:

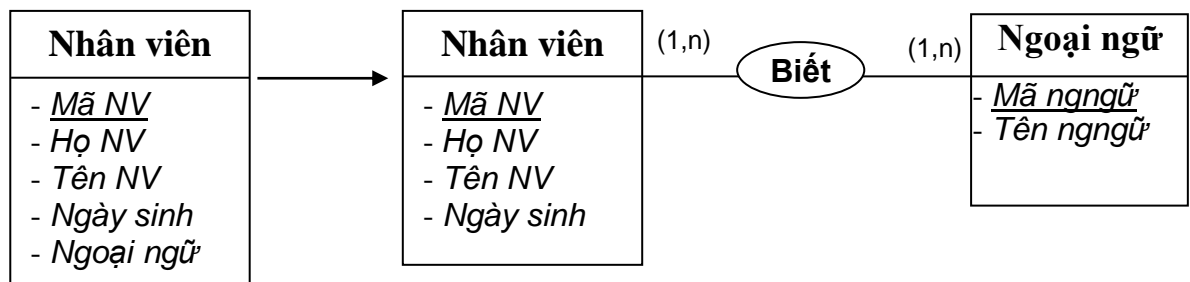


3.4.7 Xử lý một thuộc tính lặp (đa trị) nằm trong một tập thực thể

Nếu trong tập thực thể có thuộc tính đa trị thì tách thuộc tính này thành một tập thực thể có tên là <tên thuộc tính đa trị> và có hai thuộc tính là:

Mã + <tên thuộc tính> và **Tên** + <tên thuộc tính>.

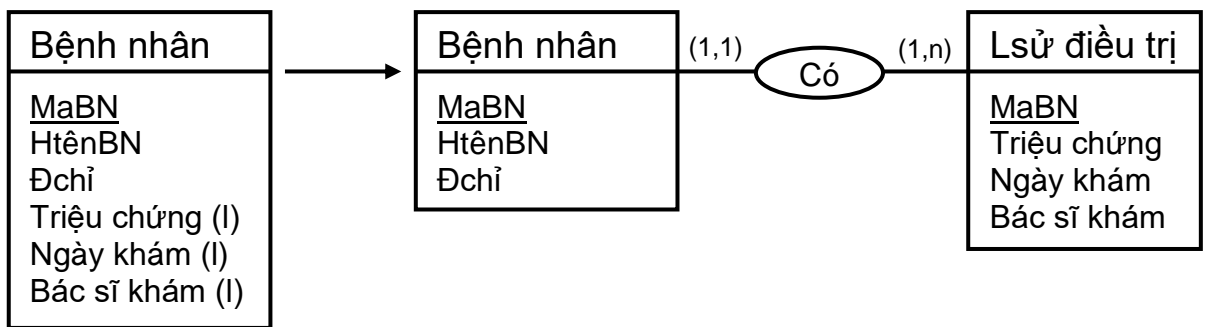
Ví dụ: một nhân viên có thể biết nhiều ngoại ngữ khác nhau (lặp). Khi đó thuộc tính Ngoại ngữ trong tập thực thể **Nhân viên** phải được chuyển thành một tập thực thể khác.



3.4.8 Xử lý một nhóm thuộc tính lặp nằm trong cùng một tập thực thể

Nếu trong một tập thực thể có một nhóm thuộc tính lặp thì tách chúng (các thuộc tính lặp) thành một tập thực thể riêng. Tập thực thể này nhận các thuộc tính lặp làm thuộc tính và nhận thuộc tính khóa của tập thực thể gốc làm khóa.

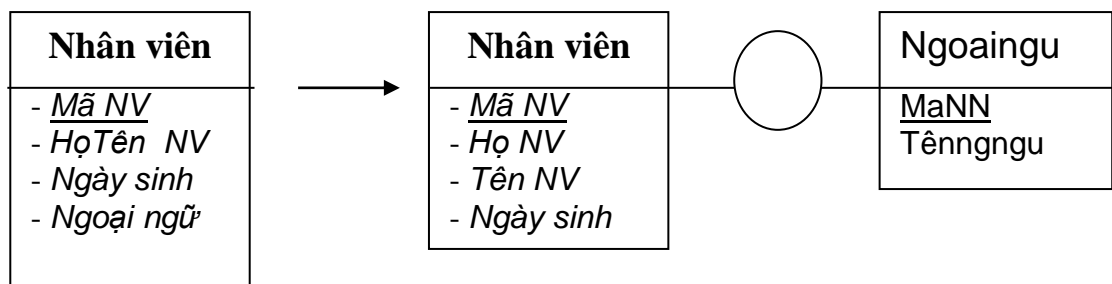
Ví dụ: một bệnh nhân có thể có nhiều triệu chứng, ngày khám và bác sĩ khám. Trong trường hợp dưới đây chúng ta chuyển các thuộc tính lặp này thành một tập thực thể riêng.



3.4.8 Xử lý các thuộc tính phức hợp

Thay thuộc tính phức hợp bởi các thuộc tính đơn tạo thành nó.

Ví dụ:



3.4.9 Các tập thực thể có mối quan hệ ISA

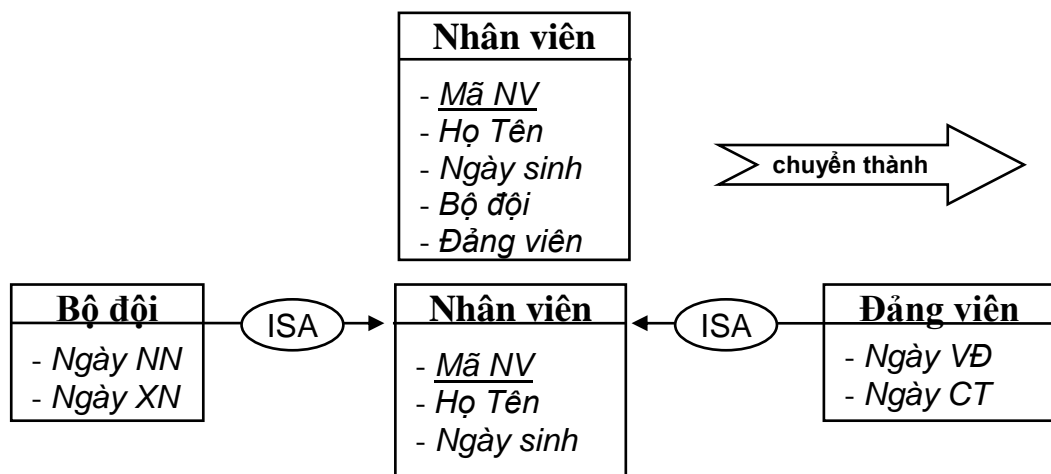
Khi một thuộc tính của tập thực thể mà chỉ có một số phần tử có giá trị, nếu phần tử nào có giá trị thì có thêm một số thuộc tính riêng của nó thì chuyển thành một tập thực thể riêng có tên là <tên thuộc tính> và có thuộc tính là các thuộc tính riêng của nó (trường hợp này giữa hai tập thực thể này có mối quan hệ ISA).

Tập thực thể gốc gọi là tập thực thể Cha, tập thực thể được tách ra gọi là tập thực thể Con.

Ví dụ: Trong hệ thống quản lý nhân viên của một cơ quan, với tập thực thể **Nhân viên**, ngoài những Thuộc tính chung như : Họ, tên, ngày sinh, giới tính, nơi sinh... còn có các Thuộc tính **Đảng viên**, **Bộ đội** ...

Thuộc tính **Đảng viên** để quản lý những Đảng viên trong cơ quan. Chỉ có một số nhân viên là Đảng viên, nếu là Đảng viên thì quản lý : Ngày vào Đảng, ngày chính thức, nơi vào Đảng. Nơi vào Đảng chỉ quản lý cấp tỉnh.

Thuộc tính **Bộ đội** để quản lý những nhân viên trong cơ quan từng đi bộ đội. Chỉ có một số nhân viên là Bộ đội. Nếu là Bộ đội thì quản lý các thuộc tính: Ngày nhập ngũ, ngày xuất ngũ, cấp bậc và binh chủng khi xuất ngũ. Như vậy, Thuộc tính **Đảng viên** và **Bộ đội** được tách thành các tập thực thể con.



Nếu trong tập thực thể con được tách ra tồn tại các thuộc tính có tính chất trên thì tiếp tục tách thuộc tính đó thành tập thực thể như phương pháp ở trên.

3.5 Mô hình quan niệm về dữ liệu

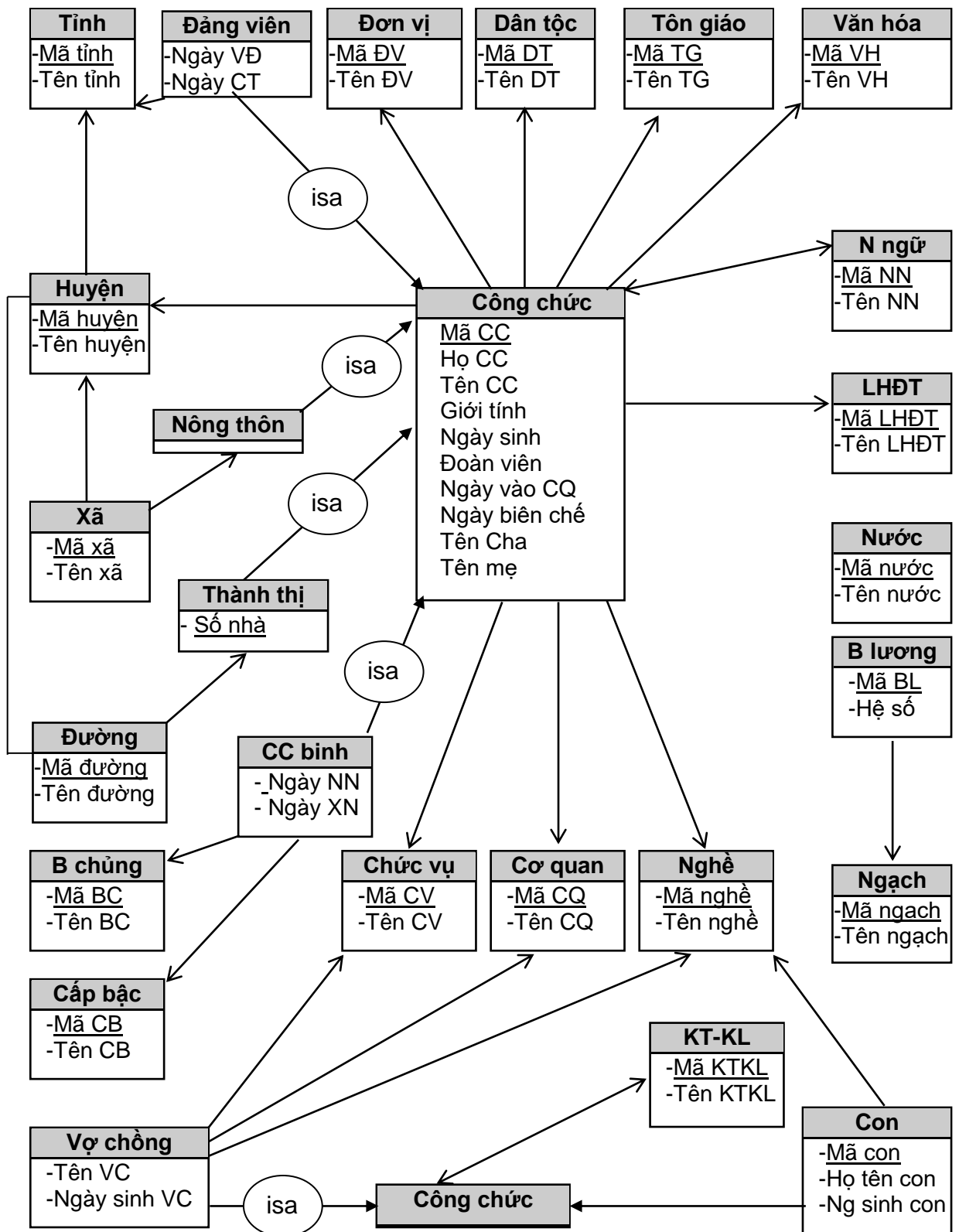
Mô hình quan niệm dữ liệu mô tả toàn bộ dữ liệu của hệ thống. Thực chất mô hình quan niệm dữ liệu là mô hình thực thể - mối quan hệ. Để mô tả mô hình quan niệm về dữ liệu của một hệ thống thông tin, cần mô tả thông tin theo các bước sau:

B1: Mô tả toàn bộ các tập thực thể và các thuộc tính tương ứng của chúng.

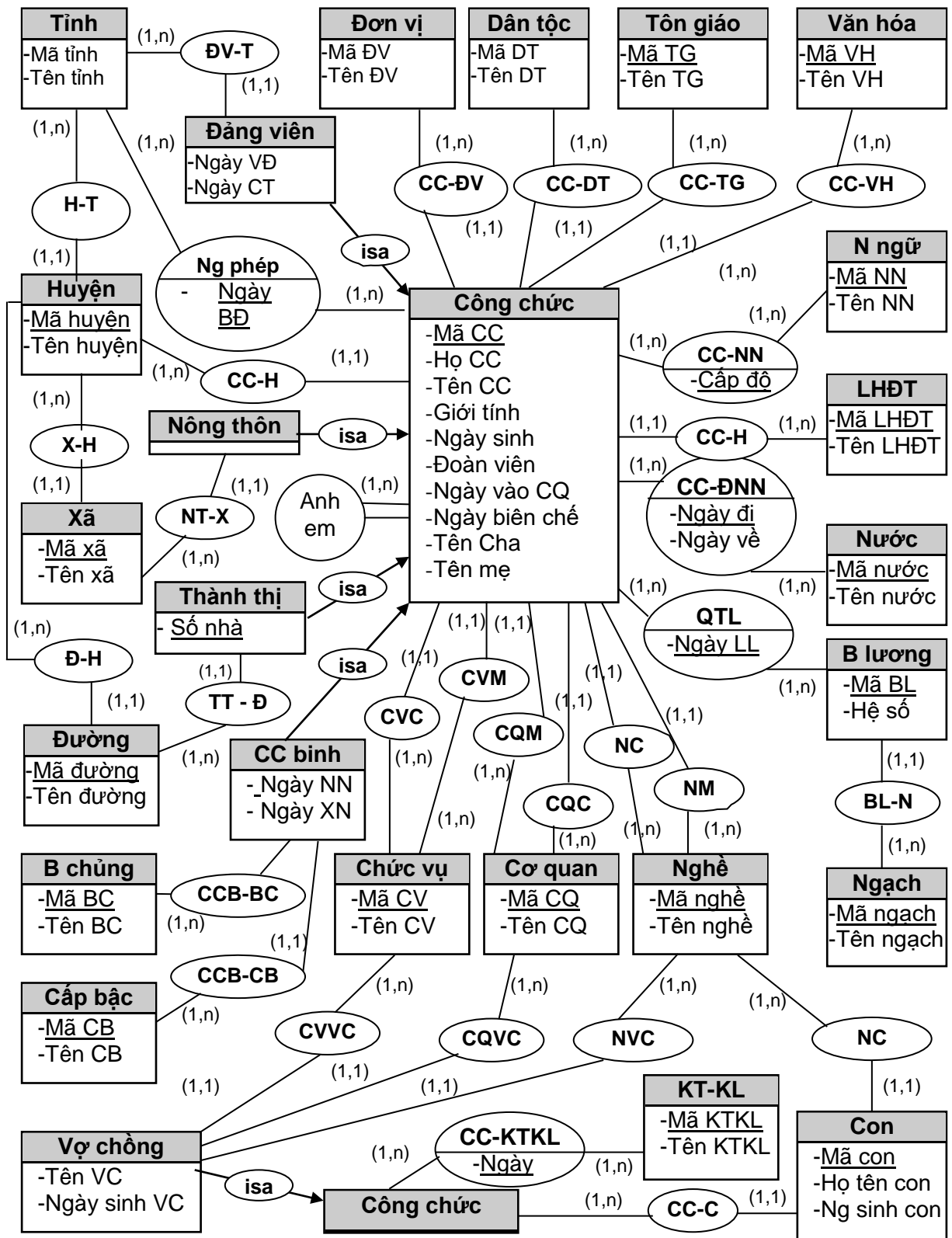
B2: Mô tả toàn bộ các mối quan hệ. Ý nghĩa của mỗi mối quan hệ và các thuộc tính tương ứng của chúng (nếu có). Bản số của mỗi tập thực thể qua mỗi quan hệ. Loại mỗi quan hệ: một chiều, hai chiều (1-1, 1-n, n-n, isa,...), nhiều chiều.

B3: Vẽ mô hình thực thể - mối quan hệ.

Ví dụ: Mô hình thực thể của hệ thống thông tin "Quản lý Công chức"



Mô hình thực thể -mối quan hệ của hтт "Quản lý Công chức"



3.6 Mô hình quan niệm xử lý:

3.6.1 **Mục đích:** Mô hình quan niệm xử lý nghiên cứu mặt động của hệ thống thông tin, nhằm xác định hệ thống gồm những chức năng gì, các chức năng đó liên hệ với nhau như thế nào? Ở mức này chưa quan tâm các chức năng đó do ai làm, làm khi nào, làm ở đâu?

3.6.2 Một số thuật ngữ và khái niệm

a. **Biến cố** (sự kiện): một sự việc gây ra sự thay đổi trạng thái của hệ thống. Một biến cố có thể xuất hiện bên trong hay bên ngoài hệ thống, tạo phản ứng cho hệ thống thông qua một qui tắc quản lý nào đó. Một biến cố sau khi kích hoạt một công việc thực hiện sẽ tạo một biến cố mới hay dữ liệu mới. Ví dụ, biến cố "*có độc giả mượn sách*" sẽ kích hoạt sự hoạt động của hệ thống.

Có thể phân loại biến cố theo nhiều khía cạnh

Biến cố vào: biến cố tham gia vào việc kích hoạt một công việc nào đó của hệ thống. Biến cố này còn gọi là biến cố khởi động. Ví dụ, biến cố "*có yêu cầu xuất kho*"

Biến cố ra: biến cố được sinh ra sau một hoặc nhiều công việc của hệ thống được thực hiện. Ví dụ, biến cố "*đủ tư cách độc giả*" sinh ra sau công việc "*Kiểm tra tư cách độc giả*".

Biến cố trong: biến cố xảy ra bên trong hệ thống để các hệ thống trao đổi thông tin cho nhau. Ví dụ, biến cố "*có sách theo yêu cầu*" xảy ra sau công việc "*Tìm sách*", biến cố này sẽ được công việc "*Cho mượn sách*" sử dụng.

Biến cố ngoài: biến cố đến từ môi trường bên ngoài hệ thống. Ví dụ, biến cố "*có sách mới*" đến từ "Nhà xuất bản"

Biến cố thời gian: biến cố gắn liền với thời gian, có tính chu kỳ. Ví dụ, việc thông báo ĐTB cho SV vào cuối năm học.

b. **Công việc:** là một xử lý nhỏ nhất mà hệ thống thực hiện khi một biến cố trong hệ thống xuất hiện. Thông thường một công việc chưa đủ để xác định được một chức năng hoặc một nhiệm vụ của hệ thống. Ví dụ, công việc "*Kiểm tra hàng nhập về*" chưa đủ điều kiện để thực hiện nhiệm vụ "nhập kho", bởi vì

nhiệm vụ này chỉ được hoàn tất sau khi các công việc *kiểm tra* và *làm phiếu nhập* được thực hiện. Một công việc còn được gọi là một *quy tắc quản lý*. Sau khi một công việc được thực hiện thì thông thường một trong hai trạng thái sẽ xảy ra: thành công (OK) và không thành công (\neg OK). Hai trạng thái này sẽ kèm theo các biến cố ra tương ứng.

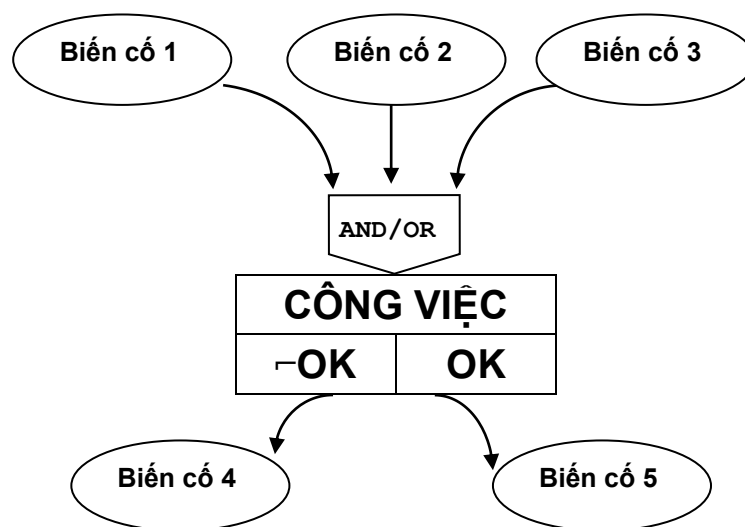
c. **Điểm đợi:** một công việc được thực hiện phải được kích hoạt bởi một hay nhiều biến cố. Các biến cố này có thể được sinh từ kết quả của những công việc khác hoặc những biến cố đã có sẵn. Thời điểm để đợi các biến cố xảy ra thì công việc mới thực hiện được gọi là *điểm đợi*. Các biến cố này kích hoạt công việc theo hai chế độ:

- **Chế độ AND:** khi tất cả các biến cố tại điểm đợi cùng xảy ra thì công việc mới được thực hiện.
- **Chế độ OR:** khi một trong các biến cố tại điểm đợi xảy ra thì công việc mới được thực hiện.

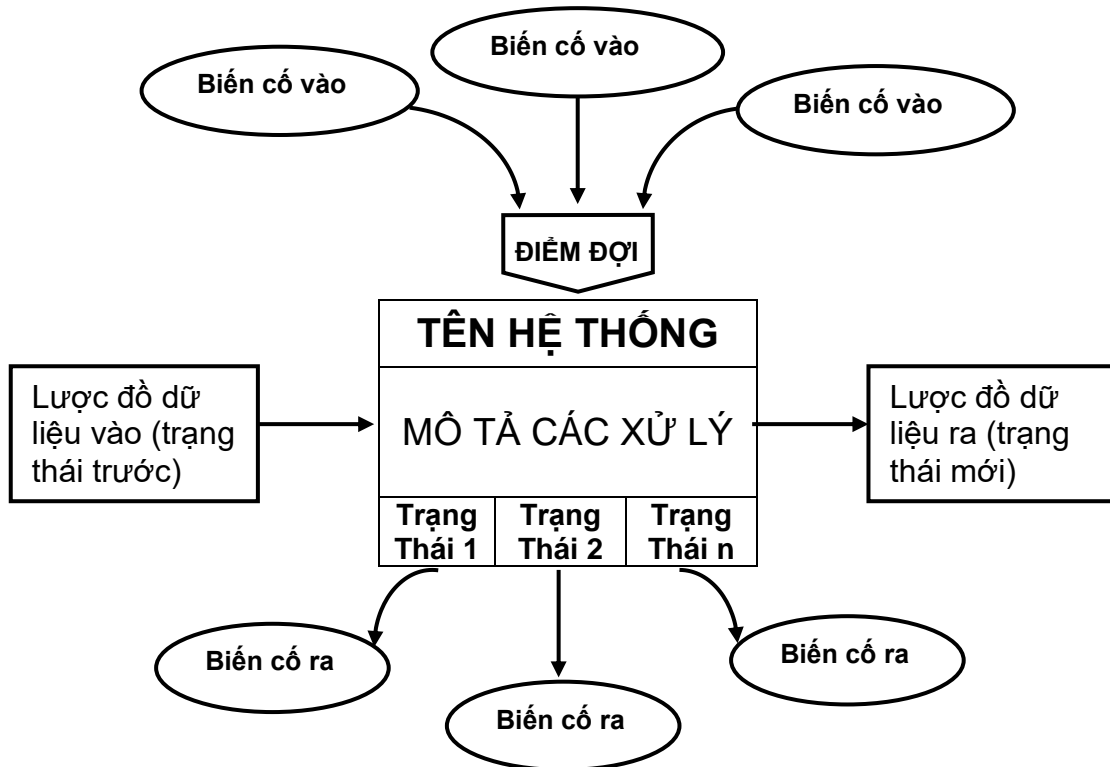
Ví dụ: Biến cố "*sách đã cho mượn*" được thực hiện bởi công việc "CHO MƯỢN SÁCH" nếu tại điểm đợi các biến cố xảy ra:

$$[(\text{Độc giả yêu cầu}) \wedge (\text{Đủ tư cách đọc giả}) \vee (\text{có lệnh của GD})] \wedge (\text{có sách})$$

Ký hiệu:



Tổng quát, Ở mức tổ chức một hệ thống thông tin hoặc một chức năng của hệ thống được mô tả như sau:

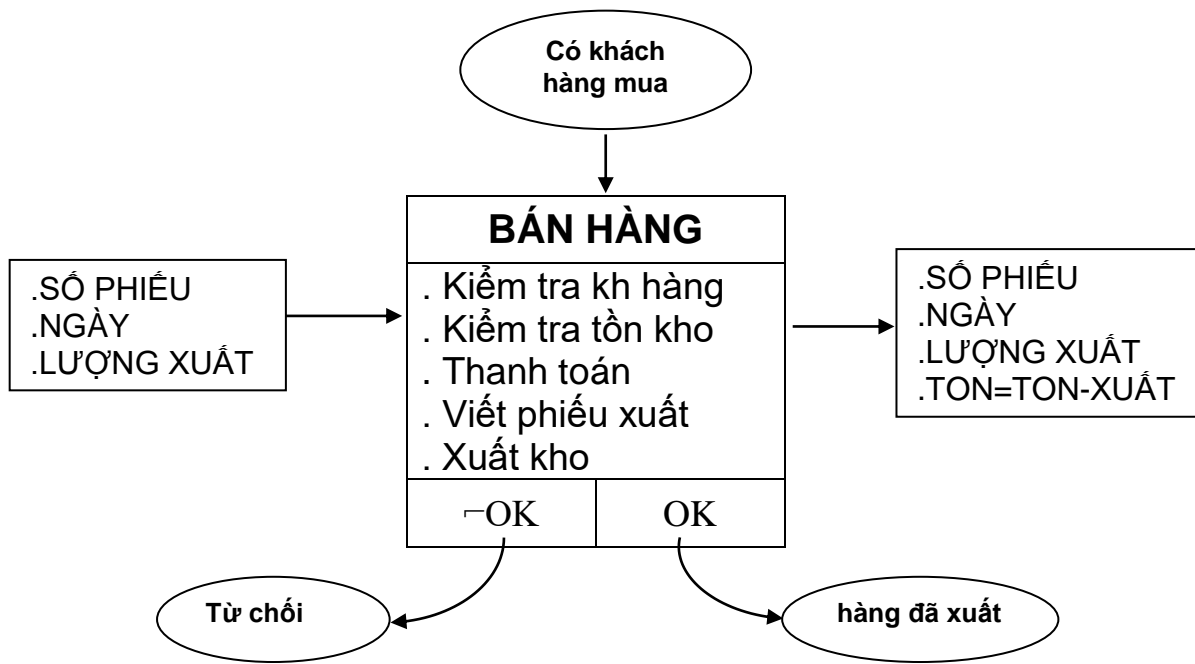


Biểu diễn một hệ thống

Một hệ thống có thể được phân rã thành các hệ thống con bằng cách chi tiết các xử lý thành các công việc để cuối cùng mỗi công việc sau khi thực hiện sẽ cho một trong hai trạng thái \neg OK và OK. Hai trạng thái này sẽ cho các biến cố ra khác nhau để làm biến cố vào cho các công việc tiếp theo.

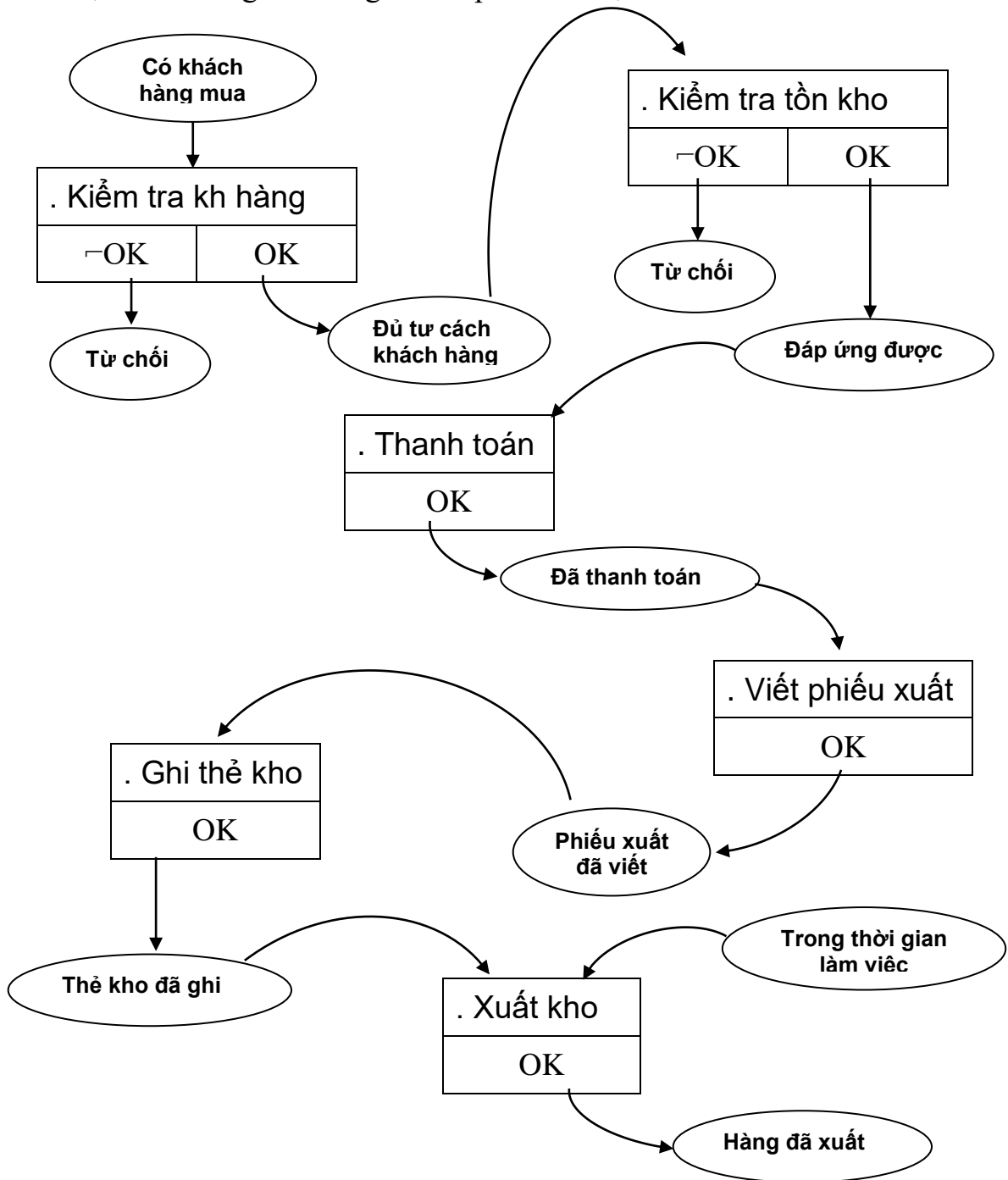
Ví dụ, trong hệ thống thông tin “Quản lý kho hàng” Chức năng “**Bán hàng**” sẽ bao gồm các công việc: kiểm tra tư cách khách hàng, kiểm tra hàng tồn kho, viết phiếu xuất, thanh toán, xuất kho.

Chúng ta có thể phân rã một hệ thống lớn thành các hệ thống nhỏ hơn, hoặc phân rã một chức năng thành các công việc.



Chức năng bán hàng khi chưa phân rã

Ví dụ: Chức năng bán hàng khi đã phân rã được mô tả như sau:



Chức năng bán hàng khi đã phân rã

3.6.3 Mô hình quan niệm về xử lý

Là mô hình liên hoàn các biến cố và các công việc của hệ thống thông tin.

Khi mô tả mô hình quan niệm xử lý cần phải liệt kê thứ tự thực hiện các công việc của hệ thống.

Ví dụ: ***Danh sách các công việc, theo thứ tự thực hiện*** của HTTT "***Quản lý tuyển sinh đại học***":

1. Thông báo tuyển sinh	2. Nhận hồ sơ dự thi
3. Đánh SBD	4. Lập danh sách TS trong phòng thi
5. In Giấy báo thi	6. Gửi Giấy báo thi
7. Thi tuyển sinh	8. Làm phách
9. Chấm thi	10. Nhập điểm
11. Ráp phách	12. Thống kê điểm
13. Lập danh sách đề nghị xét tuyển	14. Xét tuyển
15. In giấy báo kết quả	16. Thông báo kết quả trúng tuyển

Mô hình quan niệm xử lý của hệ thống thông tin “Quản lý tuyển sinh ĐH”

