

## NHÀ XUẤT BẢN THÔNG TIN VÀ TRUYỀN THÔNG

Trụ sở chính: 18 Nguyễn Du, Hà Nội

ĐT Biên tập: 04.35772143, 04.35772145

E-mail: nxb.tttt@mic.gov.vn

Website: www.nxbthongtintruyenthong.vn

ĐT Phát hành: 04.35772138

Fax: 04.35772037

Chi nhánh TP. Hồ Chí Minh: 8A đường D2, Phường 25, Quận Bình Thạnh, TP. Hồ Chí Minh

Điện thoại: 08.35127750, 08.35127751

Fax: 08.35127751

E-mail: cnsng.nxbtttt@mic.gov.vn

Chi nhánh TP. Đà Nẵng: 42 Trần Quốc Toản, quận Hải Châu, TP. Đà Nẵng

Điện thoại: 0511.3897467

Fax: 0511.3843359

E-mail: cndn.nxbtttt@mic.gov.vn

### MỜI CÁC BẠN TÌM ĐỌC

1. GIÁO TRÌNH MẠNG MÁY TÍNH
2. GIÁO TRÌNH KỸ THUẬT LẬP TRÌNH
3. GIÁO TRÌNH LÝ THUYẾT THÔNG TIN
4. GIÁO TRÌNH CƠ SỞ DỮ LIỆU PHÂN TÁN
5. GIÁO TRÌNH KỸ THUẬT VI XỬ LÝ (02 TẬP)
6. LÝ THUYẾT HỆ THỐNG VÀ ĐIỀU KHIỂN HỌC
7. CƠ SỞ DỮ LIỆU WEB - CƠ BẢN VÀ NÂNG CAO
8. GIÁO TRÌNH CƠ SỞ DỮ LIỆU: LÝ THUYẾT VÀ THỰC HÀNH
9. KỸ THUẬT PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN HƯỚNG CẤU TRÚC

### SÁCH CỦA NHÀ XUẤT BẢN THÔNG TIN VÀ TRUYỀN THÔNG CÓ BÁN TẠI:

1. Nhà sách Tiên phong  
175 Nguyễn Thái Học, Hà Nội
2. Nhà sách Nguyễn Văn Cừ  
36 Xuân Thủy, Cầu Giấy, Hà Nội
3. Nhà sách Minh Châu  
Số 10 và 14/40 Tạ Quang Bửu, Hai Bà Trưng, Hà Nội
4. Nhà sách PTIT  
Km 10, Nguyễn Trãi, Hà Nội
5. Nhà sách Bách Khoa  
Số 1, Đường Giải Phóng, Hà Nội  
86 - 107 Tô Hiến Thành, Quận 10, TP. HCM
6. Nhà sách Thăng Long  
2 Bis Nguyễn Thị Minh Khai, Quận 1, TP. HCM

Giá: 39.000đ

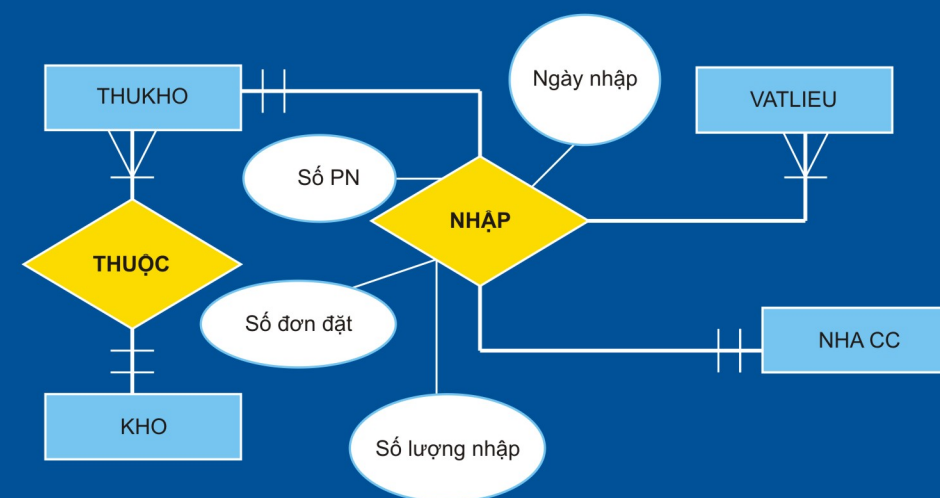
TS. LÊ VĂN PHÙNG

CƠ SỞ DỮ LIỆU QUAN HỆ  
VÀ CÔNG NGHỆ PHÂN TÍCH - THIẾT KẾ

THÔNG TIN VÀ TRUYỀN THÔNG  
NHÀ XUẤT BẢN

TS. LÊ VĂN PHÙNG

# CƠ SỞ DỮ LIỆU QUAN HỆ VÀ CÔNG NGHỆ PHÂN TÍCH - THIẾT KẾ



NHÀ XUẤT BẢN THÔNG TIN VÀ TRUYỀN THÔNG

TS. LÊ VĂN PHÙNG

**CƠ SỞ DỮ LIỆU QUAN HỆ  
VÀ CÔNG NGHỆ  
PHÂN TÍCH - THIẾT KẾ**

NHÀ XUẤT BẢN THÔNG TIN VÀ TRUYỀN THÔNG



## LỜI NÓI ĐẦU

Cơ sở dữ liệu (CSDL) là một trong những môn học chuyên ngành quan trọng của chuyên ngành công nghệ thông tin. Mục đích của cuốn sách nhằm giúp bạn đọc nắm bắt nhanh những vấn đề cốt lõi nhất về mặt lý thuyết CSDL quan hệ, trên cơ sở đó vận dụng vào thực tế trong việc thiết kế cơ sở dữ liệu logic cũng như thiết kế hệ thống thông tin nói chung. Do đó, nội dung tài liệu không đi sâu vào việc chứng minh các định lý mà chú trọng đến việc làm sáng tỏ ý nghĩa thực tế của các công thức để từ đó hướng dẫn học viên cách tiếp cận tư duy logic, nắm vững kỹ thuật tính toán cũng như các bước triển khai giải quyết các bài toán thực tế trên khía cạnh công nghệ.

Với mục đích cung cấp những kiến thức cơ bản về cơ sở dữ liệu quan hệ cho sinh viên, học viên ngành công nghệ thông tin và các ngành kỹ thuật khác, Nhà xuất bản Thông tin và Truyền thông trân trọng giới thiệu cuốn sách ***“Cơ sở dữ liệu quan hệ và công nghệ phân tích - thiết kế”*** do TS. Lê Văn Phùng, Viện Công nghệ thông tin, Viện Khoa học và Công nghệ Việt Nam biên soạn.

Nội dung sách được biên soạn sau nhiều năm thể nghiệm giảng dạy tại các trường đại học, cao đẳng và Trung tâm đào tạo tin học tại nhiều tỉnh thành trong cả nước. Cuốn sách gồm 3 chương:

*Chương 1 - Tổng quan về cơ sở dữ liệu:* trích lọc những khái niệm cơ bản nhất về dữ liệu, cơ sở dữ liệu, hệ thống xử lý dữ liệu, hệ quản trị CSDL, các phương tiện diễn tả dữ liệu, từ điển dữ liệu, cũng như những khái niệm cốt lõi về mô hình dữ liệu sẽ được sử dụng trong quá trình phân tích, thiết kế CSDL.

*Chương 2 - Mô hình cơ sở dữ liệu quan hệ:* trình bày những khái niệm cơ bản nhất về mô hình dữ liệu quan hệ của E.F.Codd, chúng được dùng nhiều trong việc thiết kế các hệ quản trị CSDL hiện nay. Chương này cũng giới thiệu những thuật toán quan trọng trong CSDL như các

thuật toán tính bao đóng và khoá tối thiểu trên 1 quan hệ cũng như trên 1 sơ đồ quan hệ... Nội dung chương còn đặc biệt quan tâm đến các dạng chuẩn và các thuật toán có liên quan. Việc chuẩn hoá các quan hệ cũng như các sơ đồ quan hệ đóng một vai trò quan trọng trong việc thiết kế các hệ quản trị cơ sở dữ liệu trên mô hình dữ liệu của E.F.Codd. Bên cạnh đó, chương 2 còn tập trung hướng dẫn áp dụng các khái niệm và phương pháp tiên tiến để chuẩn hoá các quan hệ đến 3NF (3NF) trong các hệ quản trị CSDL hiện hữu.

*Chương 3 - Quy trình phân tích thiết kế CSDL:* được trình bày một cách hệ thống toàn bộ quy trình phân tích, thiết kế một CSDL, bao gồm các giai đoạn chủ chốt như phân tích yêu cầu dữ liệu, phát triển mô hình phân tích dữ liệu, thiết kế CSDL mức logic và mức vật lý, thiết kế an toàn bảo mật cho CSDL. Quá trình phân tích, thiết kế nhằm xây dựng và quản trị CSDL được trình bày ở đây tập trung vào CSDL quan hệ, một loại CSDL được xây dựng dựa trên mô hình dữ liệu quan hệ có lý thuyết toán học vững chắc nhất. Ngoài ra, đây cũng là loại CSDL phổ biến nhất, được sử dụng hầu hết trong các hệ quản trị CSDL thông dụng trên thế giới.

Các nội dung được gói gọn trong 3 chương trên là những kiến thức thực hành cần thiết cho bất cứ bạn đọc nào yêu và say mê với công nghệ thông tin.

Hy vọng cuốn sách sẽ thực sự hữu ích đối với các kỹ sư, kỹ thuật viên, cán bộ giảng dạy và sinh viên ngành Công nghệ Thông tin, khi thực hiện các đề tài, đồ án, các dự án, cũng như trong giảng dạy, học tập và vận hành các hệ thống thông tin. Ngoài ra, cuốn sách cũng là tài liệu tham khảo bổ ích cho bạn đọc quan tâm tới lĩnh vực này.

Nhà xuất bản xin trân trọng giới thiệu cùng bạn đọc và rất mong nhận được ý kiến đóng góp của quý vị. Mọi ý kiến đóng góp xin gửi về **Nhà xuất bản Thông tin và Truyền thông** - 18 Nguyễn Du, Hà Nội hoặc gửi trực tiếp cho tác giả theo địa chỉ: [lvphung@ioit.ac.vn](mailto:lvphung@ioit.ac.vn).

Trân trọng cảm ơn./.

**NXB THÔNG TIN VÀ TRUYỀN THÔNG**

## **TỪ VIẾT TẮT**

CSDL	Cơ sở dữ liệu
DBMS	Hệ quản trị cơ sở dữ liệu
E_R	Thực thể - mối quan hệ
KT_XH	Kinh tế - Xã hội
NSD	Người sử dụng
PC	Máy tính cá nhân
PT_TK	Phân tích và thiết kế
RBTV	Ràng buộc toàn vẹn



## Chương 1

# TỔNG QUAN VỀ CƠ SỞ DỮ LIỆU

### 1.1. CÁC KHÁI NIỆM CƠ BẢN

#### 1.1.1. Dữ liệu

Trong máy tính, thuật ngữ dữ liệu được xem như là các đặc tính được biết đến mà có thể ghi lại và lưu trữ trên các thiết bị ghi nhớ của máy tính. Dữ liệu là những mô tả về sự vật, con người và sự kiện trong thế giới thực.

*Dữ liệu bao gồm số, kí tự, văn bản, hình ảnh, đồ họa, âm thanh, đoạn phim,... có một giá trị nào đó đối với người sử dụng (NSD) chúng và được lưu trữ, xử lý trong máy tính.*

*Ví dụ:*

- Dữ liệu về khách: tên, địa chỉ, điện thoại, thẻ tín dụng
- Dữ liệu về xe ô tô của khách: hãng xe, đời xe, năm sản xuất
- Dữ liệu về nhật ký sửa chữa: ngày phục vụ, tên thợ sửa chữa, số tiền thanh toán

Trong hoạt động kinh tế xã hội (KT\_XH) của con người, người ta thường chia ra hai loại dữ liệu là loại dữ liệu phản ánh cấu trúc nội bộ của cơ quan (nhân sự, nhà xưởng, thiết bị,... dữ liệu ít biến động) và loại dữ liệu phản ánh hoạt động của tổ chức (sản xuất, mua bán, giao dịch,...). Trong doanh nghiệp, không kể con người và thiết bị, dữ liệu cùng với xử lý là 2 thành phần cơ bản của hệ thống: dữ liệu thường dùng để ghi nhận thực trạng.

*Dữ liệu có thể được chia thành nhiều cấp theo kiểu cây phân cấp lưu trữ dữ liệu (Data storage hierarchy)*



Database - cơ sở dữ liệu
File - tệp
Record - bản ghi
Field- trường
Character or byte - Ký tự hoặc byte
bit

**Bit** (binary digit) là một số có 1 ký tự số (digit), được biểu diễn bằng số 0/1. Đây là đơn vị lưu trữ nhỏ nhất trong máy tính.

**Một ký tự** (character) hoặc byte là một tổ hợp 8 bit liên tục nhau

Ký tự gồm có:

- ký tự chữ cái A - Z, a - z
- ký tự số 0 - 9
- ký tự đặc biệt (+, %, \$,...)

Cấu trúc bit nhỏ nhất cho 1 ký tự là 0000 0000

Cấu trúc bit lớn nhất cho 1 ký tự là 1111 1111

Như vậy có 256 byte biểu diễn  $2^8 = 256$  ký tự khác nhau

*Có 2 hệ thống mã để chuyển đổi mã 8 bit thành các ký tự:*

- **E**xtended **b**inary **c**oded **d**ecimal **i**nterchange **c**ode (dùng cho máy mainframe và máy IBM lớn)
- **A**merican **s**tandard **c**ode for **i**nformation **i**nterchange (dùng cho máy mini và máy vi tính)

Việc chuyển đổi dữ liệu ở dạng mã này sang dạng mã kia nhờ vào những trình tiện ích

*Sức chứa ở bộ nhớ chính và bộ nhớ phụ được tính bằng:*

- Kilobyte (K)            1K = 1024 byte
- Megabyte (M)        1M = 1024K
- Gigabyte (G)         1G = 1024M
- Têtabyte (T)          1T = 1024G

Phần lớn máy PC (personal computer) thời kỳ đầu đều dựa trên chip 8086/8088 là loại có dung lượng 16 bit.

PC mới hiện nay dùng chip có dung lượng lớn như 32, 64 bit.

**Field** (Trường) là tập các byte có liên quan

*Ví dụ:* tên người có thể được cấu tạo từ các trường tên, tên lót, họ

Trường là đơn vị dữ liệu nhỏ nhất có nghĩa đối với NSD

Trường có thể chỉ chứa 1 ký tự F (female)/M (male)

Trường số có thể có dấu chấm (.) và có dấu (âm/dương). Một số phần mềm không được phép sử dụng số âm và dấu chấm thập phân.

Trường phải được khai báo trước

**Record** (bản ghi) là tập các trường có liên hệ với nhau

Bản ghi về hợp đồng thuê nhân công có thể có trường tên, địa chỉ, số giờ làm việc trong tuần. Trường trong bản ghi được phân loại (type) xác định độ dài tối đa (size). Nhờ vậy bản ghi xác định được số byte tổng cộng được chứa trong một vùng liên tục trong bộ nhớ.

**File** (tệp) là tập các bản ghi chứa các dữ liệu có quan hệ với nhau được lưu vào thiết bị nhớ thứ cấp (đĩa mềm, đĩa cứng, băng caset).

Các phương pháp chủ yếu để tổ chức lưu trữ và đọc các tệp trên đĩa hoặc các thiết bị lưu trữ gồm:

- Tổ chức tệp kiểu tuần tự: các bản ghi được lưu theo thứ tự
- Tổ chức tệp truy xuất ngẫu nhiên (trực tiếp): lưu các tệp tin theo thứ tự ngẫu nhiên.
- Tổ chức tệp theo chỉ mục: tổ chức một trường khoá trong mỗi bản ghi nên có thể truy xuất bản ghi này theo kiểu tuần tự hoặc ngẫu nhiên.

*Phép băm (hashing) là một kỹ thuật được sử dụng để tính địa chỉ của một bản ghi trên thiết bị vật lý (đĩa từ). Còn tệp được tạo ra theo kỹ thuật này được gọi là tệp băm (hash file).*

**Database** (cơ sở dữ liệu) là hệ thống tích hợp của các tệp, được thiết kế nhằm mục đích làm giảm thiểu sự trùng lặp các dữ liệu. Ngoài ra chúng còn có một cấu trúc có thứ tự để kết nối các tệp lại với nhau. Theo nghĩa phổ thông, CSDL là tập hợp dữ liệu về một đơn vị tổ chức nào đó, lưu trên máy, có cách tổ chức quản lý theo một kiểu mô hình nào đó. Trên thực tế, người ta đã đưa ra một định nghĩa CSDL đầy đủ hơn dưới đây.

### 1.1.2. Cơ sở dữ liệu

*Định nghĩa: CSDL được xác định như là một bộ sưu tập các dữ liệu có liên quan logic với nhau; nó được tổ chức sắp xếp theo một cách nào đó và được các hệ ứng dụng của một đơn vị/cơ quan cụ thể nào đó sử dụng.*

*Ví dụ:* một CSDL của cửa hàng sửa chữa xe ô tô có thể chứa các dữ liệu về khách hàng (bao gồm các khoản mục dữ liệu như tên khách hàng, địa chỉ, số điện thoại, thẻ tín dụng), các dữ liệu về xe ô tô của khách hàng (bao gồm các khoản dữ liệu như hãng xe, đời xe, năm sản xuất) và nhật ký các lần sửa chữa cho mỗi chiếc xe (như ngày phát sinh phục vụ, tên thợ sửa, loại hình sửa chữa và số tiền đã thanh toán).

CSDL thường được sử dụng với những ứng dụng cài đặt riêng lẻ trong máy tính cá nhân, hoặc dưới dạng một CSDL tập trung trên mạng LAN, hoặc dưới dạng một CSDL phân tán trên mạng WAN.

Các thành phần của môi trường sử dụng CSDL bao gồm:

- Tệp và CSDL,
- Phần cứng và hệ điều hành,
- Cấu trúc dữ liệu và phương pháp truy xuất dữ liệu.

### 1.1.3. Hệ thống xử lý dữ liệu

Từ khi có máy tính, người ta đã tìm cách *lưu trữ* dữ liệu và *xử lý* chúng, một trong những hệ thống xử lý dữ liệu đầu tiên đó là các “*hệ thống xử lý tệp*”. Mặc dù ngày nay những hệ thống xử lý tệp này không

còn được sử dụng rộng rãi nhưng nó vẫn tồn tại trong một số lĩnh vực hẹp. Để có thể thấy được những ưu nhược điểm của hệ thống xử lý tệp cũng như có thể hiểu sâu hơn tại sao người ta lại mong muốn xây dựng những CSDL với các ưu điểm của cách tiếp cận dựa trên hệ thống xử lý tệp, chúng ta sẽ xem xét một cách khái quát các khái niệm quan trọng.

#### **a. Các hệ thống xử lý tệp truyền thống**

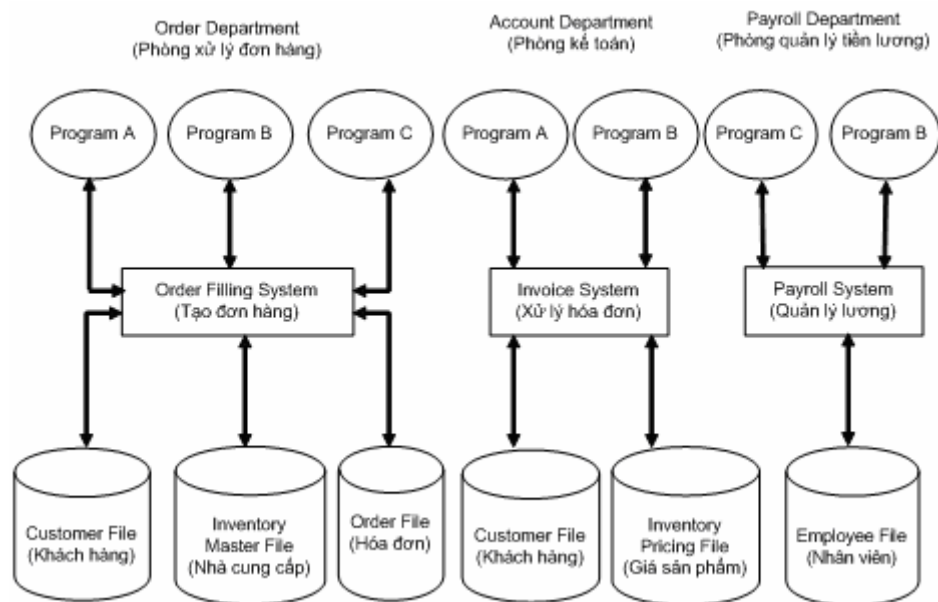
Khi chưa có CSDL, các máy tính của chúng ta phải xử lý dữ liệu thông qua các hệ thống xử lý tệp truyền thống. Ngày nay, khi khả năng của các máy tính đã tăng lên một cách đáng kể, có thể giải quyết các bài toán lớn, đòi hỏi xử lý những khối dữ liệu đồ sộ, phức tạp thì các hệ thống xử lý tệp truyền thống tỏ ra không còn thích hợp và dần dần nó được thay thế bởi các hệ thống xử lý dữ liệu dựa trên CSDL. Tuy nhiên sự hiểu biết về các hệ thống xử lý tệp vẫn là cần thiết bởi hai lý do cơ bản sau:

- Ngày nay các hệ thống xử lý tệp vẫn còn được sử dụng trong một số hệ thống, đặc biệt là các hệ thống phục vụ sao lưu CSDL.
- Hiểu được các vấn đề cũng như các giới hạn của các hệ thống xử lý tệp sẽ giúp chúng ta thiết kế các CSDL tốt hơn.

*Ví dụ về một ứng dụng dựa trên hệ thống xử lý tệp:*

Một công ty chuyên sản xuất các đồ dùng bằng gỗ đã tiến hành tin học hoá các nghiệp vụ của mình trên máy tính. Thoạt đầu, mỗi một ứng dụng trong công ty đều được xây dựng theo cách tiếp cận xử lý tệp truyền thống. Với cách tiếp cận này, *các hệ thống thông tin đều tập trung vào việc xử lý dữ liệu của mỗi một phòng ban đơn lẻ*, trong đó các hệ thống cố gắng đáp ứng các yêu cầu của từng nhóm người cụ thể. Do vậy, nhìn toàn bộ vào hệ thống thông tin của toàn công ty thì không có một mô hình cũng như kế hoạch hay sơ đồ tổng thể nào để có thể hoạch định được cho việc phát triển tiếp theo của các ứng dụng. Hệ thống ứng dụng dựa trên cách tiếp cận xử lý tệp này được minh họa như trong hình 1.1, đó là các hệ thống tạo lập đơn hàng (Order Filling

System), hệ thống xử lý hoá đơn (Invoice System) và hệ thống quản lý lương cho nhân viên (Payroll System). Nhìn vào hình vẽ chúng ta thấy rằng, các tệp dữ liệu được kết nối trực tiếp với mỗi chương trình ứng dụng, mỗi tệp là một tập các bản ghi liên quan. Ví dụ, với hệ thống Order Filling System có 3 tệp: Khách hàng (Customer), Nhà cung cấp (Inventory Master) và Hoá đơn (Order). Chúng ta nhận thấy rằng có thể có những tệp được lưu lặp lại và sử dụng bởi các ứng dụng khác nhau (như Customer), đây chính là một trong những nhược điểm của các hệ thống dựa trên việc xử lý tệp.



Hình 1.1. Một hệ thống xử lý tệp truyền thống

Các nhược điểm của các hệ thống xử lý tệp:

- *Tính phụ thuộc giữa chương trình và dữ liệu:* Các mô tả về tệp dữ liệu được lưu trong mỗi một chương trình ứng dụng sử dụng tệp đó. Như ví dụ trên, chương trình Program A truy nhập cả hai tệp Inventory Master và Customer, do vậy chương trình này sẽ chứa các mô tả chi tiết của cả 2 tệp này. Điều này dẫn đến, khi có bất cứ một thay đổi nào đến cấu trúc tệp sẽ đòi hỏi cũng phải có các thay đổi tương ứng đến

các mô tả trong các chương trình ứng dụng sử dụng tệp đó. Ngoài ra, như trong hình 1.1, nếu tệp Customer bị sửa đổi thì cả hai hệ thống Order System và Invoice System đều bị ảnh hưởng.

- *Lưu trữ trùng lặp dữ liệu*: Trong hình 1.1 chúng ta nhận thấy rằng, hệ thống Order Filling có chứa một tệp Inventory Master, trong khi đó hệ thống Invoice cũng chứa một tệp là Inventory Pricing. Rõ ràng giữa hai tệp này sẽ có sự lưu trữ trùng lặp dữ liệu, ví dụ các thông tin về mô tả sản phẩm, giá mỗi sản phẩm cũng như số lượng đang có trong kho. Việc lưu trữ dư thừa dữ liệu này gây ra rất nhiều bất cập và đòi hỏi không gian lưu trữ lớn, nó rất dễ vi phạm tính toàn vẹn của dữ liệu.

- *Việc chia sẻ dữ liệu bị hạn chế*: Trong cách tiếp cận xử lý tệp này, mỗi một ứng dụng có các tệp dữ liệu riêng của nó và do vậy giữa những người sử dụng, họ có ít cơ hội để có thể chia sẻ dữ liệu cho nhau nếu không sử dụng chung một phân hệ ứng dụng. Như trong hình 1.1, các nhân viên trong phòng kế toán (Accounting Department) chỉ có thể truy nhập được những tệp trong ứng dụng của các phòng xử lý đơn hàng (Order Department) hay phòng quản lý tiền lương (Payroll Department). Do vậy, khi người lãnh đạo muốn tạo ra những báo cáo tổng hợp liên phòng thì vô cùng khó khăn bởi dữ liệu được lưu riêng lẻ cũng như có thể không thống nhất giữa các phòng.

- *Tốn thời gian xây dựng*: Với mỗi một ứng dụng mới, người phát triển chương trình phải bắt đầu lại từ đầu với việc xây dựng các tệp, mô tả chúng cũng như xây dựng các mô-đun chương trình để truy nhập tới các tệp đó. Quá trình này tốn khá nhiều thời gian mặc dù nó luôn lặp lại giữa các hệ thống.

- *Chi phí cho bảo trì chương trình lớn*: Trên thực tế, có đến 80% chi phí được dành cho giai đoạn bảo trì chương trình đối với những ứng dụng dựa trên các hệ thống xử lý tệp truyền thống này.

Tuy nhiên chúng ta cũng cần phải lưu ý rằng, những nhược điểm đối với cách tiếp cận dựa trên các hệ thống xử lý tệp truyền thống này

cũng có thể gặp phải trong cách tiếp cận dựa trên CSDL (sẽ được trình bày dưới đây), nếu như CSDL không được thiết kế tốt, cũng như chúng được xây dựng một cách riêng rẽ, không thống nhất.

### ***b. Cách tiếp cận CSDL***

Cách tiếp cận CSDL nhấn mạnh đến tính chia sẻ và tích hợp của dữ liệu trong toàn bộ hoạt động của một tổ chức. Theo cách tiếp cận này, dữ liệu được lưu trữ có tổ chức, tập trung dưới dạng một CSDL và giữa các dữ liệu có sự liên kết logic với nhau.

Các ưu điểm mà cách tiếp cận dựa trên CSDL mang lại so với cách tiếp cận dựa trên các hệ thống xử lý tệp truyền thống:

- *Tối thiểu hoá dư thừa dữ liệu*: Mục đích thiết kế trong cách tiếp cận CSDL chính là cố gắng tích hợp các tệp dữ liệu riêng rẽ (và cũng là dư thừa) trong mô hình xử lý tệp trước đây vào một cấu trúc logic duy nhất. Mỗi một sự kiện cụ thể sẽ được lưu trữ tại một vị trí quy định trong CSDL. Cũng cần lưu ý rằng, cách tiếp cận CSDL có thể không loại bỏ hoàn toàn vấn đề dư thừa dữ liệu, nhưng nó đã cho phép người thiết kế kiểm soát được vấn đề này. Ví dụ, đối với mỗi đơn hàng trong bảng ORDER có chứa một trường Customer\_id để biểu diễn mối liên kết giữa nâng cao hiệu quả truy vấn của CSDL, chúng ta có thể chấp nhận hình thức lưu trữ dư thừa dữ liệu.

- *Tính độc lập chương trình - dữ liệu*: Tính độc lập với dữ liệu được thể hiện ở sự tách biệt rõ ràng giữa chương trình ứng dụng và các mô tả dữ liệu (siêu dữ liệu), do vậy nó cho phép dữ liệu của một doanh nghiệp được phép sửa đổi (trong một giới hạn nào đó) mà không phải thay đổi lại chương trình thao tác trên các dữ liệu đó.

- *Nâng cao tính nhất quán dữ liệu*: Với việc loại bỏ tính dư thừa dữ liệu, cách tiếp cận CSDL đã giảm bớt được đáng kể tính không nhất quán của dữ liệu. Ví dụ, nếu địa chỉ của khách hàng được lưu trữ duy nhất, khi đó, nếu khách hàng thay đổi địa chỉ, chúng ta chỉ việc làm một động tác là đơn giản sửa đổi lại địa chỉ của khách hàng ở vị trí duy nhất đó, các truy vấn liên quan đến giá trị này sẽ được tự động nhận

giá trị mới khi được thực hiện. Như vậy, vấn đề về cập nhật, sửa đổi dữ liệu rất đơn giản và nhất quán. Một điểm quan trọng hơn, cách tiếp cận CSDL sẽ làm giảm đáng kể không gian nhớ so với phương pháp lưu trữ dữ liệu trong các hệ thống xử lý tệp truyền thống.

- *Nâng cao tính chia sẻ dữ liệu*: CSDL được thiết kế nhằm mục đích chia sẻ dữ liệu. Mỗi người sử dụng hợp pháp được gán những quyền sử dụng CSDL khác nhau, đối với mỗi người (hoặc nhóm người) sử dụng sẽ được cung cấp những khung nhìn sử dụng khác nhau, do vậy họ có thể chủ động trong việc chia sẻ dữ liệu thông qua các quyền kiểm soát dữ liệu.

- *Nâng cao tính sản phẩm của ứng dụng*: Một lợi thế đáng kể của cách tiếp cận CSDL đó là giảm đáng kể chi phí và thời gian cho việc phát triển những ứng dụng nghiệp vụ.

- *Áp đặt dễ dàng các tiêu chuẩn*: Với cách tiếp cận CSDL, các tiêu chuẩn như quy ước đặt tên, sử dụng quy trình thống nhất trong việc truy nhập, cập nhật và bảo vệ dữ liệu dễ dàng được thực hiện. Các chức năng quản trị CSDL thường cung cấp một loạt các công cụ nhằm đảm bảo những tiêu chuẩn này luôn phải được thoả mãn. Tính chất này làm cho việc bảo trì và trao đổi dữ liệu trở nên dễ dàng và thuận lợi hơn.

- *Nâng cao chất lượng dữ liệu*: Với cách tiếp cận CSDL, người thiết kế có thể xây dựng các ràng buộc toàn vẹn nhằm đảm bảo dữ liệu khi được đưa vào CSDL phải thoả mãn một số quy tắc nào đó. Hoặc đối với những CSDL lớn như kho dữ liệu, thì thông thường dữ liệu được làm sạch trước khi được đưa vào kho dữ liệu.

- *Nâng cao khả năng truy xuất dữ liệu*: Với các CSDL quan hệ, người sử dụng không cần biết kỹ năng về lập trình cũng có thể dễ dàng tìm kiếm và hiển thị dữ liệu thông qua một ngôn ngữ hỏi đáp dữ liệu rất thân thiện đã được tiêu chuẩn hoá.

- *Giảm chi phí bảo trì chương trình*: Tính chất này là hệ quả của tính độc lập chương trình - dữ liệu. Tính độc lập chương trình - dữ liệu cho phép xây dựng chương trình tách biệt khỏi dữ liệu ở mức độ nhất



định, do vậy khi có sự sửa đổi, chúng ta chỉ việc sửa đổi một số mô-đun thao tác trên dữ liệu mà không nhất thiết phải sửa đổi lại toàn bộ chương trình. Chính điều này đã dẫn đến việc giảm bớt được chi phí bảo trì chương trình.

Trên đây là những ưu điểm mà cách tiếp cận CSDL mang lại. Một CSDL thường được tích hợp vào trong một *hệ quản trị CSDL* (đó là một chương trình máy tính hay chính xác hơn là một bộ các chương trình máy tính được xây dựng để quản lý một CSDL).

#### 1.1.4. Sự khác biệt giữa dữ liệu và thông tin

Hai thuật ngữ dữ liệu và thông tin liên quan chặt chẽ với nhau, trên thực tế chúng thường được sử dụng thay đổi cho nhau. Tuy nhiên, giữa chúng cũng có sự khác biệt và chúng ta nên phân biệt giữa dữ liệu và thông tin.

*Chúng ta quan niệm thông tin là những gì được tạo nên nhằm giảm sự không xác định. Thông tin thường được chiết xuất từ dữ liệu. Nói cách khác chúng ta có thể hiểu thông tin như là dữ liệu đã được xử lý theo cách mà chúng có thể làm tăng hàm lượng tri thức cho NSD nó.*

*Ví dụ:*

Hoá đơn N<sup>0</sup>- 852 của nhà cung cấp A cần phải thanh toán chậm nhất là vào ngày 01/10/2005 đó là một thông tin.

Để thể hiện thông tin người ta sử dụng dữ liệu. Thông tin được thể hiện qua các giá trị của dữ liệu. Với ví dụ trên, thông tin toàn bộ được thể hiện qua 3 loại dữ liệu: STT hoá đơn, tên nhà cung cấp, ngày giới hạn thanh toán.

Có dữ liệu nhưng chưa thể hiện được thông tin.

*Ví dụ:*

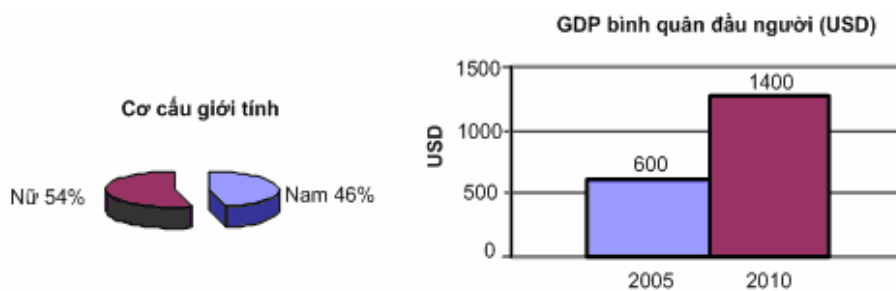
Mai Linh	240685
Đông Lan	060980
Minh Phương	211279

Để chuyển dữ liệu thành thông tin, cần tóm lược chúng lại hoặc xử lý và trình bày chúng dưới các dạng trực quan hơn như bảng, đồ thị hay biểu đồ (đặt danh sách trong một ngữ cảnh, bổ sung thêm một vài dữ liệu và cấu trúc của các dữ liệu đó hoặc tóm lược và biểu diễn dưới dạng trực quan - dạng thông tin đồ thị):

Bảng 1.1. Danh sách lớp Tin cơ bản

Tên	Ngày tháng năm sinh	Mã sinh viên
Mai Linh	240685	
Đông Lan	060980	
Minh Phương	211279	

Có một số cách diễn tả dữ liệu như bằng đồ thị hình bánh hoặc đồ thị hình cột. Đồ thị hình bánh biểu diễn cơ cấu giới tính. Đồ thị hình cột biểu diễn tăng trưởng GDP bình quân đầu người,...



Hình 1.2. Các cách diễn tả dữ liệu

### 1.1.5. Siêu dữ liệu

Dữ liệu chỉ trở nên hữu ích khi chúng được đặt trong một ngữ cảnh nào đó. Cơ chế chính cung cấp ngữ cảnh cho dữ liệu đó chính là siêu dữ liệu, do vậy chúng ta có thể hiểu *siêu dữ liệu là dữ liệu mô tả các tính chất hoặc đặc trưng của dữ liệu*. Các tính chất này có thể là các dữ liệu về định nghĩa, về cấu trúc dữ liệu, về các luật hay các ràng buộc.

Siêu dữ liệu cho phép người thiết kế CSDL và NSD biết có dữ liệu gì, tồn tại ở đâu, các khoản mục dữ liệu khác nhau ở điểm gì. Việc

quản lý siêu dữ liệu cũng quan trọng như việc quản lý dữ liệu liên quan với nó, bởi dữ liệu nếu không được phân biệt rõ ràng sẽ gây ra nhầm lẫn, sai sót và không thể diễn giải nổi. Trong CSDL, siêu dữ liệu thường được lưu trữ trong một vùng quy định mà chúng ta thường gọi là từ điển dữ liệu (Data Dictionary).

Dữ liệu tạo nên nội dung của CSDL, tuy nhiên nó không quyết định tính hiệu quả của CSDL. Bởi tính hiệu quả được thể hiện qua việc thiết kế và khai thác CSDL.

*Bảng 1.2. Ví dụ về một siêu dữ liệu cho bảng kết quả học tập*

Data item		Length	Value		Description
Name	Type		Min	max	
Course	Char	30			Course_id và tên
Section	Integer	1	1	9	Số hiệu học phần
Semester	Char	10			Học kỳ và năm
Name	Char	30			Tên Sinh viên
ID	Integer	9			Số hiệu sinh viên
Major	Char	4			Môn học
GPA	Decimal	3	0.0	10.0	Điểm môn học

Cần phân biệt rõ dữ liệu và siêu dữ liệu: siêu dữ liệu là phần được tách khỏi dữ liệu, siêu dữ liệu mô tả các tính chất của dữ liệu, siêu dữ liệu không bao gồm dữ liệu.

### 1.1.6. Kiểu thực thể và bảng dữ liệu

#### a. Thực thể

Thực thể là một khái niệm được sử dụng thường xuyên trong quá trình phân tích, thiết kế CSDL. Chúng ta cần nắm vững bản chất của thực thể.

Chúng ta nhận thấy rằng trong thế giới thực, tồn tại rất nhiều các đối tượng khác nhau, ví dụ như một người trong tổ chức, một sinh viên

trong một trường đại học, hay thậm chí một phòng ban trong một viện nghiên cứu, tất cả những đối tượng đó đều được coi là các thực thể khác nhau. Ngoài ra những đối tượng trừu tượng như “một ngành khoa học” cũng có thể được coi là một thực thể.

Vậy ta có thể định nghĩa: *thực thể như là một đối tượng cụ thể hay trừu tượng trong thế giới thực mà nó tồn tại và có thể phân biệt với các đối tượng khác.*

### ***b. Kiểu thực thể***

Từ định nghĩa về thực thể nêu trên, chúng ta nhận thấy rằng trong thế giới thực có rất nhiều các thực thể giống nhau ở một số khía cạnh nào đó.

Như vậy, chúng ta có thể coi một nhóm các thực thể giống nhau hình thành lên một kiểu thực thể hay còn gọi là tập thực thể.

*Ví dụ:*

Tất cả các sinh viên trong một trường đại học tạo thành một kiểu thực thể là STUDENT.

Tuy nhiên chúng ta cũng nhận thấy rằng khái niệm “*thực thể giống nhau*” không được xác định một cách chính xác và người ta có thể đưa ra rất nhiều các tính chất khác nhau để hình thành lên một kiểu thực thể, điều này phụ thuộc vào nhu cầu quản lý của bài toán.

Vậy chúng ta *có thể hiểu một kiểu thực thể như là một nhóm các thực thể giống nhau ở một số điểm nhất định nào đó mà chúng ta cần quan tâm.*

Một trong các bước chính trong quá trình phân tích, thiết kế CSDL đó là: xây dựng một mô hình đặc tả thế giới thực, nói một cách cụ thể hơn tức là lựa chọn và mô tả các kiểu thực thể cũng như các mối quan hệ giữa các kiểu thực thể đó.

Chúng ta cần phải phân biệt sự khác nhau giữa kiểu thực thể và thể hiện của kiểu thực thể (hay thực thể):

- Một kiểu thực thể là một tập các thực thể có chung một số tính chất hoặc đặc trưng nào đó. Mỗi một kiểu thực thể phải được đặt một tên duy nhất không trùng với các kiểu thực thể đã có.

- Còn thể hiện của thực thể là một tập giá trị cụ thể của kiểu thực thể.

*Ví dụ:*

*Bảng 1.3. Kiểu thực thể EMPLOYEE và 2 thể hiện của nó*

**Kiểu thực thể: EMPLOYEE**

Các thuộc tính:

Emp_id	Char(10)
Emp_name	Char(50)
Address	Char(50)
Date_hired	Date

**Và hai thể hiện:**

642 - 17 - 125	781 - 78 - 157
Kim Luyến	Ngọc Lan
10A Tây Hồ	2 Hà Đông
01/01/2007	01/06/2008

Khi phân tích thiết kế một CSDL, một kiểu thực thể chỉ được mô tả một lần (sử dụng siêu dữ liệu), trong khi đó các thể hiện của kiểu thực thể đó lại được biểu diễn bởi dữ liệu lưu trong CSDL.

Để đặt tên cho kiểu thực thể, chúng ta dùng các danh từ viết liền với các chữ cái được viết hoa, bao quanh bởi một hình chữ nhật:



*Hình 1.3. Kiểu thực thể*

Các kiểu thực thể được đặc tả bởi một tập các đặc trưng mà chúng ta quy ước gọi là các thuộc tính. Do vậy muốn thành lập một kiểu thực thể đòi hỏi phải xác định được tập các thuộc tính chung nhất.

**c. Bảng dữ liệu hay quan hệ**

Trong quá trình phân tích CSDL, khái niệm kiểu thực thể được sử dụng để minh họa cho một lớp các thực thể giống nhau. Bảng dữ liệu

chính là hình thức thể hiện cụ thể của kiểu thực thể khi xây dựng CSDL, và bảng dữ liệu được sử dụng để lưu dữ liệu về các thực thể trong lớp thực thể đó.

Chúng ta có thể hình dung bảng dữ liệu là một tập các bộ dữ liệu, hay bản ghi dữ liệu, mỗi bộ có cùng một số lượng thuộc tính như nhau nhưng có thể khác nhau về miền giá trị. Bảng dữ liệu trong cách tiếp cận CSDL quan hệ, được hiểu chính xác bởi cụm từ quan hệ. Một quan hệ bao gồm một lược đồ quan hệ và một thể hiện cột của bảng đó. Trong một quan hệ không thể tồn tại hai bộ dữ liệu giống nhau ở tất cả các thuộc tính. Hình dưới đây minh họa cho một bảng dữ liệu, các cột của bảng tương ứng với các thuộc tính hay các trường dữ liệu, các hàng của bảng tương ứng với các bản ghi hay các bộ dữ liệu.

*Trường dữ liệu (hay thuộc tính, cột dữ liệu)*

*Bảng 1.4. Dữ liệu hay quan hệ*

	Emp_id	Emp_name	Address	Date_hired	Birth_date
	784-123	Hồng Đào	10K Nguyễn Trãi	01/01/2008	12/04/1982
	123-541	Thanh Hằng	45 Hàm Long	01/06/2007	15/03/1983
	145-458	Thu Hoa	18A Đường Láng	01/01/2008	14/08/1984

Đôi khi chúng ta có thể biểu diễn các bảng dưới dạng lược đồ quan hệ loại bỏ các dữ liệu và chỉ đưa ra tên bảng cùng các tên cột như sau:

AUTHOR (Au\_id, Au\_iname, Au\_fname, Address, City, Country)

TITLE (Title\_id, Title, Type, Price, Pub\_id)

PUBLISHER (Pub\_id, Pub\_name, City)

AUTHOR\_TITLE (Au\_id, Title\_id)

*Mỗi bảng muốn trở thành quan hệ cần có một số tính chất sau:*

- Các dữ liệu trong bảng là các giá trị đơn, được hiểu theo nghĩa là các giá trị không cần chia nhỏ hơn nữa mà vẫn đáp ứng được nhu cầu nghiệp vụ.

- Các giá trị trong cùng một cột dữ liệu phải có cùng một kiểu.

- Mỗi một hàng trong bảng là duy nhất. Trong bảng không thể tồn tại hai hàng có cùng giá trị ở tất cả các cột.

- Thứ tự liệt kê các cột trong bảng không quan trọng.

- Thứ tự liệt kê các hàng trong bảng không quan trọng.

- Mỗi một cột dữ liệu trong bảng phải có một tên duy nhất.

Như vậy một quan hệ là một bảng hai chiều gồm các giá trị, đó là tập hợp các bộ của quan hệ tại một thời điểm nào đó. Tuy nhiên, quan hệ lại thường được cập nhật (bổ sung, loại bỏ, điều chỉnh các bộ), do đó quan hệ có tính chất là thay đổi theo thời gian.

Tuy thay đổi về giá trị thông qua việc cập nhật, song quan hệ vẫn phải giữ một số đặc điểm về ngữ nghĩa, hay nói một cách khác nó phải tuân thủ một lược đồ quan hệ nhất định như qui ước số trường trong quan hệ, kiểu dữ liệu trong quan hệ,...

#### ***d. Thuộc tính và trường dữ liệu***

##### *Thuộc tính*

Như trong khái niệm về kiểu thực thể chúng ta đã biết, một kiểu thực thể được mô tả bằng cách sử dụng một tập các thuộc tính. Tất cả các thực thể trong một kiểu thực thể đã được xác định đều có cùng số lượng các thuộc tính đó.

Mỗi một kiểu thực thể luôn có một tập các thuộc tính đi kèm. Dưới đây là ví dụ về hai kiểu thực thể và các thuộc tính đi kèm với chúng:

STUDENT:      Student\_id, Student\_name, Address, Tel\_no, Class

EMPLOYEE:      Emp\_id, Emp\_name, Address, Skill, Start\_date

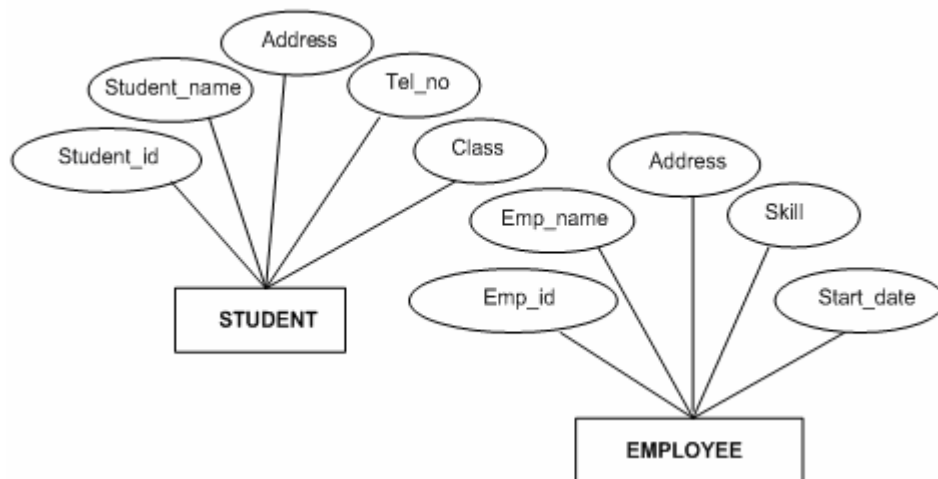
Với mỗi thuộc tính trong kiểu thực thể, chúng ta phải xác định miền giá trị của thuộc tính đó. Ví dụ miền giá trị cho thuộc tính

Emp\_name của kiểu thực thể EMPLOYEE có thể là một chuỗi ký tự có độ dài không quá 30 ký tự.

Thể hiện của thuộc tính chính là các giá trị. Ví dụ “Tân Huệ” là một giá trị cho thuộc tính tên nhân viên Emp\_name.

Để đặt tên cho các thuộc tính, chúng ta dùng các danh từ viết liền với chữ cái đầu tiên được viết hoa, nếu một thuộc tính có tên gồm 2 từ thì chúng ta sử dụng dấu gạch dưới để nối 2 từ đó lại. Chúng ta sử dụng hình Elíp để biểu diễn cho một thuộc tính và một đường thẳng kết nối thuộc tính đó với kiểu thực thể mà nó mô tả.

Ví dụ:



Hình 1.4. Thuộc tính và kiểu thực thể

#### Trường dữ liệu

Cũng như giữa bảng dữ liệu và kiểu thực thể, khi phân tích CSDL, chúng ta sử dụng các thuộc tính để mô tả cho các kiểu thực thể, còn khi xây dựng CSDL, chúng ta sử dụng các trường dữ liệu để xây dựng lên các quan hệ. Do vậy chúng ta có thể hiểu trường dữ liệu như là đơn vị nhỏ nhất có tên gọi của dữ liệu lưu trữ trong CSDL. Trường dữ liệu còn được gọi là cột dữ liệu khi chúng ta sử dụng với khái niệm bảng dữ liệu.



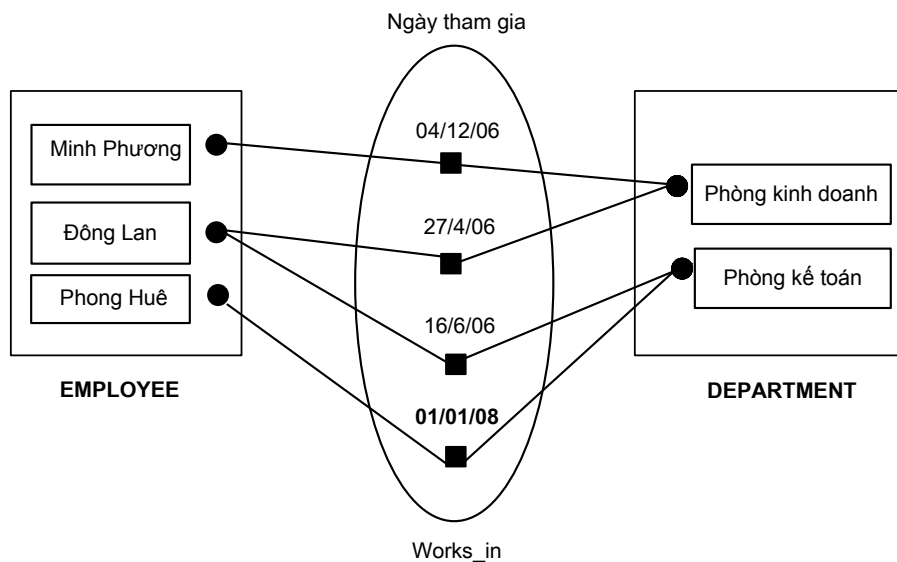
Qua đó chúng ta có thể ánh xạ các khái niệm trong khâu phân tích CSDL và các khái niệm được sử dụng trong khâu thiết kế, xây dựng CSDL:

- Từ kiểu thực thể chúng ta xây dựng được lên các quan hệ hay bảng dữ liệu.
- Từ các thuộc tính chúng ta xây dựng được lên các trường hay cột dữ liệu.

#### ***e. Liên kết giữa các thực thể***

Một liên kết là một kết nối giữa các thể hiện của một hoặc nhiều kiểu thực thể. Ví dụ, chúng ta có một liên kết các nhân viên Minh Phương, Đông Lan, Phong Huê với các phòng kinh doanh, kế toán.

Chúng ta cũng có thể sử dụng các thuộc tính để mô tả cho liên kết. Các thuộc tính mô tả được sử dụng để lưu lại các thông tin về mỗi liên kết hơn là những thông tin về các thực thể tham gia vào liên kết đó. Ví dụ, chúng ta có thể ghi lại nhân viên Đông Lan làm việc ở phòng kinh doanh từ ngày 27/4/2006 thông qua một thuộc tính Start\_date được đặt bên cạnh liên kết Works\_in như trong hình 1.5.



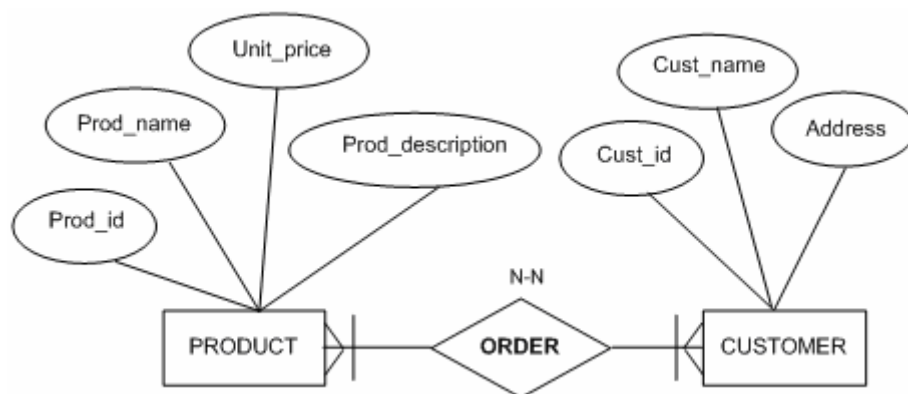
*Hình 1.5. Liên kết giữa các thực thể*

Một liên kết phải được xác định duy nhất bởi các thực thể tham gia liên kết mà không cần phải tham chiếu đến các thuộc tính mô tả. Như hình 1.5, mỗi một liên kết trong Work\_in được xác định duy nhất bởi việc kết hợp thuộc tính đại diện Emp\_name của kiểu thực thể EMPLOYEE và thuộc tính đại diện Dept\_name của kiểu thực thể DEPARTMENT. Do vậy với mỗi cặp nhân viên - phòng ban trên chúng ta không thể có hơn một kết nối.

*Chú ý:* Trong ví dụ trên chúng ta giả thiết rằng giữa các thực thể của kiểu thực thể có thể phân biệt duy nhất với nhau thông qua các giá trị trong các thuộc tính Emp\_name và Dept\_name tương ứng với hai kiểu thực thể EMPLOYEE và DEPARTMENT.

Một nhóm các liên kết cùng kiểu, hình thành lên một tập liên kết. Khi dùng với khái niệm kiểu thực thể, chúng ta dùng thuật ngữ liên kết để mô tả cho sự kết nối giữa chúng thay vì sử dụng thuật ngữ tập liên kết.

Để đặt tên cho liên kết, chúng ta dùng các động từ viết liền với chữ cái đầu tiên được viết hoa, nếu một động từ có tên gồm 2 từ thì chúng ta sử dụng dấu gạch dưới để nối 2 từ đó lại. Chúng ta sử dụng hình thoi để biểu diễn cho một liên kết và các đường nối liên kết đó với các kiểu thực thể tham gia vào liên kết. Một ví dụ về mối liên kết được đặc tả như sau:



Hình 1.6. Liên kết giữa kiểu thực thể

Liên kết đặt mua “Orders” giữa hai kiểu thực thể là khách hàng CUSTOMER và mặt hàng PRODUCT được đặc tả với ý nghĩa: mỗi khách hàng có thể đặt mua một hoặc nhiều mặt hàng, ngược lại mỗi mặt hàng cũng có thể được mua bởi nhiều khách hàng khác nhau.

#### 1.1.7. Khái niệm toàn vẹn dữ liệu

Như đã trình bày trong phần đầu, một trong những ưu điểm của cách tiếp cận CSDL đó là dữ liệu khi được đưa vào CSDL sẽ phải tuân theo một số quy định nào đó. Những quy định này được xem là những ràng buộc trên dữ liệu và nó đảm bảo tính toàn vẹn của dữ liệu. Tính toàn vẹn dữ liệu giúp cho NSD có thể tin cậy ít nhất về mặt nguyên tắc đối với các dữ liệu trong CSDL.

##### *a. Ràng buộc toàn vẹn*

*Ràng buộc toàn vẹn (RBTV) là một điều kiện bất biến không được vi phạm trong một CSDL.*

RBTV còn được gọi là *quy tắc quản lý (Rule)* được áp đặt lên trên các đối tượng của thế giới thực.

*Ví dụ:*

- Mỗi lớp học phải có 1 mã số duy nhất để phân biệt với mọi lớp học khác trong trường
- Mỗi lớp phải thuộc 1 khoa trong trường
- Mỗi học sinh phải có một mã số riêng biệt, không trùng nhau trong trường
- Mỗi học viên phải thuộc 1 lớp ...

Trong một CSDL, RBTV cần được người phân tích thiết kế mô tả đầy đủ và chính xác rõ ràng trong hồ sơ phân tích thiết kế (PT\_TK).

RBTV được xem như 1 *công cụ* để diễn đạt *ngữ nghĩa* của CSDL đó. Trong quá trình khai thác CSDL, các RBTV đều phải được thỏa mãn ở bất kỳ thời điểm nào nhằm đảm bảo cho CSDL luôn ở trạng thái an toàn và nhất quán về dữ liệu.

Các hệ quản trị thường có các cơ chế tự động kiểm tra RBTV về miền giá trị của khoá, giá trị NOT NULL qua các khai báo cấu trúc các bảng (Mô hình dữ liệu quan hệ) hoặc thông qua những thủ tục kiểm tra và xử lý vi phạm RBTV (do người PT\_TK cài đặt).

*Việc kiểm tra có thể tiến hành vào 1 trong 2 thời điểm sau:*

- Kiểm tra ngay sau khi thực hiện một thao tác cập nhật CSDL (thêm, sửa, xoá). Thao tác cập nhật chỉ được coi là hợp lệ nếu nó không vi phạm bất kỳ RBTV nào. Nếu vi phạm RBTV nghĩa là không hợp lệ và sẽ bị hệ thống huỷ bỏ.

- Kiểm tra định kỳ hay đột xuất, nghĩa là việc kiểm tra độc lập với thao tác cập nhật dữ liệu. Nếu vi phạm RBTV, hệ thống có những xử lý ngầm định hoặc yêu cầu NSD xử lý.

### ***b. Các yếu tố của RBTV***

Cụ thể, xét trong một cơ sở quan hệ, chúng ta thấy luôn tồn tại nhiều mối quan hệ ảnh hưởng qua lại lẫn nhau giữa các thuộc tính của 1 quan hệ, giữa các bản ghi trong 1 quan hệ và giữa các thuộc tính của các bản ghi trong các quan hệ với nhau.

Khi xác định một RBTV cần chỉ rõ:

- Nội dung RBTV và cách biểu diễn thích hợp
- Bối cảnh xảy ra RBTV: trên 1 hay nhiều quan hệ, cụ thể các quan hệ nào
- Phạm vi ảnh hưởng của RBTV. Khả năng tính toán vẹn dữ liệu bị vi phạm.
- Các phương án xử lý khi phát hiện RBTV bị vi phạm

Nội dung RBTV là mô tả quy tắc phải tuân theo. Việc biểu diễn nó có thể bằng ngôn ngữ tự nhiên, thuật giải (ngôn ngữ có cấu trúc), ngôn ngữ đại số tập hợp, đại số quan hệ hoặc bằng phụ thuộc hàm.

HOADON (sốhd, số chủng loại mặt hàng, tổng trị giá)

HANG (Mã hàng, tên hàng, đv tính)

CHITIETHOADON (sốhd, mã hàng, SL, đơn giá, trị giá)

RBTV1: mỗi hoá đơn có một số riêng biệt không trùng với các hoá đơn khác

$$\forall \text{hd1,hd2} \in \text{HOADON}, \text{hd1} \neq \text{hd2} \Rightarrow \text{hd1.sohd} \neq \text{hd2.sohd}$$

Bối cảnh có thể định nghĩa trên 1 cơ sở quan hệ hay nhiều cơ sở quan hệ. Đó là những quan hệ mà RBTV được áp dụng trên đó:

- Bối cảnh RBTV1 (mỗi hoá đơn có một số riêng biệt không trùng với các hoá đơn khác) chỉ trong 1 quan hệ HOADON
- Bối cảnh RBTV2 (số chủng loại mặt hàng bằng số bộ của CHITIETHOADON có cùng số hoá đơn) trong 2 quan hệ HOADON và CHITIETHOADON

Một RBTV có thể liên quan đến một số quan hệ, chỉ khi có thao tác cập nhật (chèn, sửa, xoá) mới có nguy cơ vi phạm RBTV, do vậy cần chỉ rõ thao tác nào dẫn đến việc cần phải kiểm tra RBTV.

Trong quá trình phân tích thiết kế, người phân tích cần lập bảng xác định phạm vi ảnh hưởng cho mỗi RBTV nhằm xác định khi nào phải tiến hành kiểm tra các RBTV đó. Sau khi xây dựng các bảng ảnh hưởng của từng RBTV trên các quan hệ liên quan, cần tổng hợp bằng cách xây dựng bảng tổng hợp các RBTV nhằm xác định tất cả các RBTV cần phải kiểm tra trên từng quan hệ.

Cần có hành động thích hợp khi vi phạm RBTV. Thường có 2 giải pháp:

- Đưa thông báo và yêu cầu sửa chữa dữ liệu của các thuộc tính cho phù hợp với quy tắc đảm bảo tính nhất quán dữ liệu. Thông báo cần đầy đủ, thân thiện với NSD. Giải pháp này phù hợp với việc xử lý thời gian thực.
- Từ chối thao tác cập nhật. Giải pháp này phù hợp với việc xử lý theo lô. Việc từ chối cũng phải được thông báo đầy đủ rõ ràng vì sao thao tác bị từ chối và cần sửa dữ liệu nào.

### c. Phân loại RBTV

Trong CSDL quan hệ, người ta thường chia RBTV thành 2 loại chính theo bối cảnh RBTV là trên 1 quan hệ cơ sở hay trên nhiều quan hệ cơ sở.

RBTV trên bối cảnh 1 quan hệ cơ sở có các loại ràng buộc về miền giá trị, liên thuộc tính, liên bộ, liên thuộc tính.

*Ví dụ:*

KQUATHI (mãhv, mã môn, lần thi, ngày thi, điểm thi)

$\forall kq \in KQUATHI: 0 \leq kq.lần\ thi \leq 3$

$\forall kq \in KQUATHI: 0 \leq kq.điểm\ thi \leq 10$

RBTV liên thuộc tính có liên quan tới nhiều thuộc tính của 1 quan hệ. Thông thường đó là các phụ thuộc tính toán hoặc một suy diễn từ giá trị của 1 hay nhiều thuộc tính trong cùng một bộ giá trị:

CHITIETHOADON (số hd, mã hàng, SL, đơn giá, Trị giá)

$\forall cthd \in CHITIETHOADON, cthd.trị\ giá = cthd.SL * cthd.đơn\ giá$

KQUATHI (mãhv, mã môn, lần thi, ngày thi, điểm thi)

$\forall kq \in KQUATHI, nếu\ kq.lần\ thi = 1\ thì\ 0 \leq kq.điểm\ thi \leq 10.0$

hoặc: nếu  $kq.lần\ thi > 1$

thì  $\exists kq' \in KQUATHI: kq'.lần\ thi = kq.lần\ thi - 1$  và  $kq'.điểm\ thi < 5$

Có 2 trường hợp RBTV phụ thuộc tồn tại (phụ thuộc về khoá ngoại):

- Có sự hiện diện của khoá ngoại
- Có sự lồng khoá giữa các quan hệ

*Ví dụ:*

RBTV1: mỗi lớp học phải thuộc 1 khoa nhất định

KHOA (mã khoa, tên khoa)

LOPHOC (mã lớp, tên lớp, niên khoá, số học viên, *mã khoa*)

$\forall lh \in LOPHOC, \exists kh \in KHOA: lh.mã\ khoa = kh.mã\ khoa$

**RBTV2**: mỗi bộ của chi tiết hoá đơn phải có 1 hoá đơn với số hoá đơn tương ứng

HOADON (sốhd, số chủng loại mặt hàng, tổng trị giá)

CHITIETHOADON (số hd, mã hàng, SL, đơn giá)

$\forall \text{cthd} \in \text{CHITIETHOADON}, \exists \text{hd} \in \text{HOADON}: \text{cthd.sohd} = \text{hd.sohd}$

Khi có sự hiện diện của 1 thuộc tính mang tính tổng hợp hay trung gian (tính được nhờ giá trị của nhiều thuộc tính khác trên 1 hay nhiều quan hệ khác nhau trong CSDL) hay phụ thuộc tồn tại lồng khoá thì có RBTV liên bộ, liên quan hệ:

Với mọi bộ giá trị của LOPHOC, nếu số lượng học viên  $> 0$  thì số lượng này phải bằng tổng số bộ giá trị đếm được của các học viên có cùng mã lớp.

LOPHOC (mã lớp, tên lớp, niên khoá, số học viên, *mã khoa*)

HOCVIEN (mã học viên, tên học viên, mã lớp)

$\forall \text{lh} \in \text{LOPHOC}$ : nếu  $\text{lh.số học viên} > 0$ :

$\text{lh.số học viên} = \text{COUNT} (\text{hv} \in \text{HOCVIEN}, \text{hv.mã lớp} = \text{lh.mã lớp})$

### ***e. Biểu diễn RBTV bằng phụ thuộc hàm (function dependency)***

Phụ thuộc hàm (PTH) có tầm quan trọng rất lớn trong việc phân tích và thiết kế mô hình dữ liệu. Phụ thuộc hàm là một trong những khái niệm cốt lõi nhất trong mô hình cơ sở dữ liệu quan hệ sẽ được trình bày chi tiết sau.

Chúng ta cũng cần lưu ý toàn vẹn dữ liệu có thể đảm bảo tính hợp lệ của dữ liệu khi được lưu trong CSDL, nhưng lại khó có thể kiểm tra mỗi một giá trị như vậy về tính đúng. Ví dụ, khi nhập giá trị giờ làm việc của một nhân viên trong tuần, không có cách nào có thể phát hiện sự kiện là một giá trị vào bằng 35 là sai, trong khi đó thực sự phải là giá trị 38. Tuy nhiên, nếu một giá trị là 350 đối với giờ làm việc rõ ràng là sai và hệ thống sẽ không chấp nhận giá trị đó, bởi một tuần không thể có 350 giờ.

Giữ gìn tính toàn vẹn của CSDL có thể xem là sự bảo vệ dữ liệu trong CSDL chống lại sự cập nhật hay phá huỷ không có căn cứ. Chú ý, chúng ta cũng phải phân biệt tính toàn vẹn dữ liệu, với tính an toàn dữ liệu dù rằng hai khái niệm này quan hệ rất mật thiết với nhau và thực sự có thể dùng chung một cơ chế để thực hiện bảo vệ cả hai ở một mức độ nào đó.

Nói chung mỗi một bảng dữ liệu trong CSDL quan hệ đều có một tập các ràng buộc toàn vẹn đi kèm với nó. Các ràng buộc toàn vẹn này cũng được lưu trữ trong từ điển dữ liệu.

#### **1.1.8. Hệ quản trị CSDL (DBMS)**

Hệ quản trị CSDL là hệ thống bao gồm nhiều phần mềm cho phép mô tả, lưu giữ, thao tác các dữ liệu trên 1 CSDL. Nó đảm bảo tính an toàn, bí mật của dữ liệu trong môi trường nhiều người sử dụng.

Có thể đưa ra định nghĩa Hệ quản trị CSDL rõ ràng hơn. Ví dụ như là:

- Phần mềm quản lý CSDL, giúp cho việc thực hiện các thao tác như chèn, sửa, xóa và tìm kiếm dữ liệu
- Hệ thống bao gồm nhiều phần mềm cho phép mô tả, lưu giữ, thao tác các dữ liệu trên một CSDL, đảm bảo an toàn, bí mật của dữ liệu trong môi trường nhiều NSD
- Bộ phần mềm và những công cụ có sẵn do một số nhà sản xuất phần mềm cung cấp để tạo lập và quản lý CSDL

Hiện nay, chúng ta thường gặp các hệ quản trị CSDL cỡ vừa và nhỏ như Foxpro, Access, cỡ lớn như SQL Server, Oracle.

#### **1.1.9. Kho dữ liệu**

Như chúng ta đã biết, CSDL đã được sử dụng từ những năm 60 và chúng ngày càng chứng tỏ là những thành phần không thể thiếu trong các hoạt động của doanh nghiệp. Một xu hướng mới đã xuất hiện và phát triển khá nhanh trong các hệ CSDL đó là các CSDL hỗ trợ việc phân tích và khai thác dữ liệu, người ta gọi đó là những kho dữ liệu,



chúng chủ yếu được phục vụ cho các ứng dụng hỗ trợ ra quyết định DSS (Decision Support System). Kho dữ liệu có những đặc điểm cơ bản sau:

- *Tính tích hợp*: dữ liệu tập hợp trong kho được thu thập từ nhiều nguồn và được tích hợp với nhau tạo thành một thể thống nhất.

- *Tính hướng chủ đề*: Các dữ liệu được thu thập, trích lọc và làm sạch để đưa vào kho theo những chủ đề nhất định.

- *Gắn thời gian*: Mỗi hàng thông tin trong kho dữ liệu đều chứa một yếu tố thời gian như một phần của khoá chính để đảm bảo tính đơn nhất của mỗi hàng và cung cấp một đặc trưng về thời gian cho dữ liệu.

- *Tính lịch sử*: Các thông tin trong kho dữ liệu được tập trung theo thời gian và thể hiện một khung nhìn về một lĩnh vực chủ đề trong một giai đoạn dài.

- *Tính chỉ đọc*: Dữ liệu trong kho là dữ liệu chỉ đọc và không thể bị sửa đổi.

- *Tính bất biến*: Dữ liệu được lưu trữ lâu dài trong kho, mặc dù những dữ liệu mới vẫn bị sửa đổi, điều đó cho phép cung cấp thông tin về một khoảng thời gian dài, đủ số liệu cần thiết cho các mô hình nghiệp vụ phân tích, dự báo.

- *Bao gồm cả dữ liệu tổng hợp và chi tiết*: Dữ liệu chi tiết là thông tin mức thấp nhất được lưu trữ trong kho. Dữ liệu tổng hợp được tích hợp lại qua nhiều giai đoạn khác nhau.

## 1.2. CÁC PHƯƠNG TIỆN DIỄN TẢ DỮ LIỆU

### 1.2.1. Khái niệm diễn tả dữ liệu

Diễn tả dữ liệu được xem như việc xác định tên, dạng dữ liệu và tính chất của dữ liệu; nó không phụ thuộc vào người sử dụng đồng thời không phụ thuộc vào yêu cầu tìm kiếm và thay đổi thông tin.

Chúng ta tập trung đề cập đến các phương tiện và mô hình diễn tả dữ liệu. Đó là các thông tin được quan tâm đến trong quản lý; nó được lưu trữ lâu dài, được xử lý và sử dụng trong hệ thống kinh doanh.

Có nhiều công cụ để diễn tả dữ liệu, đó là các cách trừu tượng hoá dữ liệu, đặc biệt là mối quan hệ của dữ liệu nhằm phổ biến những cái chung nhất mà người ta có thể trao đổi lẫn nhau.

Có nhiều công cụ diễn tả dữ liệu, ở đây chúng ta đề cập tới 4 công cụ chủ yếu:

- Mã hoá dữ liệu (coding)
- Từ điển dữ liệu (Data Dictionary)
- Mô hình thực thể liên kết E- R (Entity\_Relationship)
- Mô hình cơ sở dữ liệu quan hệ (Relational Data Base Modeling)

### 1.2.2. Mã hoá

#### a. Khái niệm mã hoá

Mã là tên viết tắt gắn cho một đối tượng nào đó hay nói cách khác mỗi đối tượng cần có tên và vấn đề đặt ra ta sẽ đặt tên cho đối tượng như thế nào. Trong mỗi đối tượng gồm nhiều thuộc tính khác nhau thì yêu cầu mã hoá cho các thuộc tính cũng là yêu cầu cần thiết. Ngoài ra mã hoá còn là hình thức chuẩn hoá dữ liệu đặc biệt trong các hệ thống thông tin xử lý bằng máy tính.

*Một số ví dụ về mã hoá:*

Khi ta cần xác định một công dân thì số chứng minh thư hoặc số hộ chiếu là mã của công dân đó. Khi cần xác định xe ô tô hay xe máy thì biển số xe là mã của xe đó.

#### b. Chất lượng của việc mã hoá

Trong thực tế ta gặp rất nhiều đối tượng cần mã hoá như mã hoá các ngành nghề đào tạo, mã hoá các bệnh, mã số điện thoại, mã thẻ sinh viên, thẻ bảo hiểm y tế. Chúng ta có nhiều phương pháp mã khác

nhau. Do vậy, cần xác định một số tiêu chí để đánh giá chất lượng của việc mã hoá:

- Mã không được nhập nhằng: Thể hiện ánh xạ 1\_1 giữa mã hoá và giải mã, mỗi đối tượng được xác định rõ ràng với một mã nhất định.

- Thích ứng với phương thức sử dụng: Việc mã có thể tiến hành bằng *thủ công* nên dễ hiểu để giải mã, việc mã hoá *bằng máy* đòi hỏi cú pháp chặt chẽ.

- Có khả năng mở rộng mã:

- + Thêm phía cuối (sau) của các mã đã có.

- + Xen mã mới vào giữa các mã đã có, thường mã xen phải dùng phương pháp cóc nhảy, nhảy đều đặn dựa vào thống kê để tránh tình trạng “bùng nổ” mã.

- Mã phải ngắn gọn làm giảm kích cỡ của mã, đây cũng là mục tiêu của mã hoá. Tuy nhiên điều này đôi khi mâu thuẫn với khái niệm mở rộng mã sau này.

- Có tính gợi ý: Tính ngữ nghĩa của mã; Đôi khi tính gợi ý là yêu cầu đối với mã công khai, làm cho việc mã hoá thuận tiện dễ dàng

### **c. Các kiểu mã hoá**

**Mã hoá kiểu liệt kê:** dùng các số nguyên liên tiếp 000,001,002... để mã hoá.

*Ưu điểm:*

- + Không nhập nhằng

- + Đơn giản

- + Thêm phía sau

*Nhược điểm:*

- + Không xen được

- + Thiếu tính gợi ý, cần phải có bảng tương ứng

- + Không phân theo nhóm

**Mã hóa theo lát:** dùng các số nguyên nhưng phân ra từng lát (lớp) cho từng loại đối tượng, trong mỗi lát dùng mã liên tiếp.

*Ví dụ:* Mã hoá Ngũ kim

Vùng 1	0001 - 0999	ngũ kim bé
Vùng 2	0010 - 0099	vít
Vùng 3	0100 - 0299	ê cu
	0300 - 0499	bulông
	0500 - 0599	đinh
Vùng 4	1000 - 1999	chi tiết kim loại
	1000 - 1099	sắt U

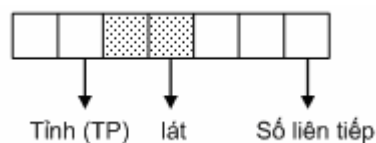
*Ưu điểm:*

- + Không nhập nhằng
- + Đơn giản
- + Mở rộng xen thêm được

*Nhược điểm:* Thiếu gợi ý

**Mã phân đoạn:** Bản thân mã phân thành nhiều đoạn, mỗi đoạn mang một ý nghĩa riêng.

*Ví dụ:* Số đăng ký xe máy



*Ưu điểm:*

- + Không nhập nhằng
- + Mở rộng xen thêm được
- + Được dùng khá phổ biến
- + Cho phép thiết lập các phương thức kiểm tra gián tiếp đối với mã của các đối tượng.

*Nhược điểm:*

- + Quá dài → thao tác nặng nề
- + Không cố định
- + Có thể bị bão hoà

**Mã phân cấp:** Các đối tượng bị mã hoá theo chế độ phân cấp các chi tiết nhỏ dần. Một hình ảnh khá quen thuộc của mã phân cấp là đánh số chương, tiết, mục trong một quyển sách.

## **1. Chương 1**

### 1.1. Bài 1

### 1.2. Bài 2

## **2. Chương 2**

### 2.1. Bài 3

#### 2.1.1. Mục 1

#### 2.1.2. Mục 2

### 2.2. Bài 4

### 2.3. Bài 5

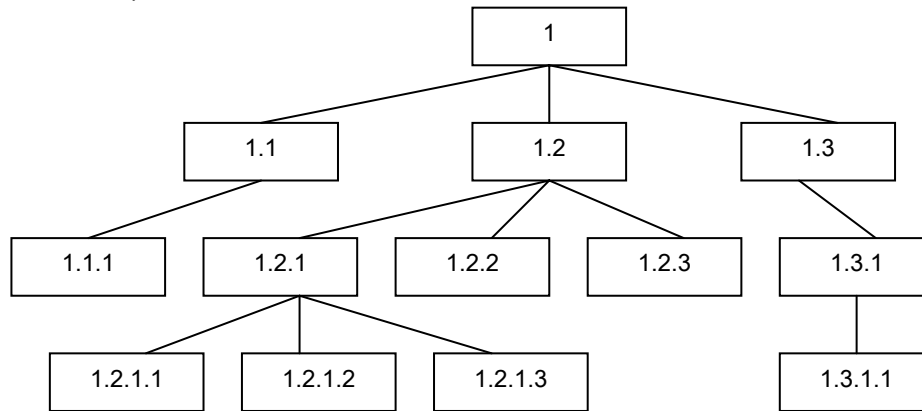
*Ưu điểm:* Giống như phân đoạn

    Tìm kiếm dễ dàng

*Nhược điểm:*

    Giống phân đoạn

Ví dụ:



Hình 1.7. Cấu trúc của mã phân cấp

**Mã diễn nghĩa:** Gắn với một tên ngắn gọn nhưng hiểu được cho một đối tượng.

*Ví dụ:* Đội bóng các nước tham gia SEA GAMES được mã bằng cách lấy 3 ký tự đầu như sau:

VIE: Vietnam, THA: Thailand, SIN: Singapore, IND: Indonesia, MAL: Malaysia...

**Ưu điểm:** tiện dụng cho NSD trong xử lý công việc hàng ngày.

**Nhược điểm:** càng dài nếu số lượng thành phần tham gia mã hoá càng lớn.

**Mã vạch - mã số hàng hoá** (bar code):

Mã số hàng hóa là một dãy các con số được ghi ở dưới mã vạch dùng để phân định sản phẩm hàng hóa. Mỗi dãy số như vậy sẽ đặc trưng cho một loại hàng hóa của cơ sở sản xuất và của quốc gia sản xuất loại hàng hóa đó. Mã số sẽ tạo thuận lợi cho quản lý trong quá trình vận chuyển, lưu kho, mua, bán. Bao giờ mã số và mã vạch được in trên bao bì sản phẩm thì sản phẩm đó mới coi là đủ tiêu chuẩn để nhập khẩu vào một nước nào đó. Hiện nay, ở Việt Nam tổ chức mã số, mã vạch được thành lập ngày 29/3/1995 gọi tắt là EAN- VN (tên tiếng Anh là Europe Article Numbering). Việt Nam là thành viên của Hội

mã số, mã vạch Quốc tế (EAN International). Như vậy, các sản phẩm của các nhà sản xuất Việt Nam muốn xuất khẩu sang nước ngoài có thể đăng ký mã số, mã vạch do EAN- VN cấp.

Hiện nay, mã số quốc gia của Việt Nam được cấp (3 số đầu) là 893, nhóm số tiếp sau chỉ mã của nhà sản xuất và nhóm số tiếp nữa là mã của sản phẩm, số cuối cùng là số kiểm tra.

Mã số của Việt Nam hiện nay sử dụng có thể là 8 hoặc 13 chữ số viết tắt là EAN- 8 và EAN- 13. EAN- 8 dùng cho sản phẩm có chủng loại ít và EAN- 13 dùng cho sản phẩm có chủng loại nhiều.

Mã vạch là một dãy các vạch đen trắng dùng để thể hiện mã số dưới dạng ký hiệu mà máy quét có thể đọc được, để lưu vào bộ nhớ của hệ thống máy tính hay máy tính tiền một cách nhanh chóng và chính xác. Dãy các vạch đen trắng này có độ chính xác tới 1/100mm và được in bằng chất liệu đặc biệt để chống làm giả.

*Ví dụ:*



*Hình 1.8*

Mã số của một sản phẩm trong hình trên là 893 5025 33457 6, trong đó 893 là Việt Nam, 5025 là mã tên đơn vị sản xuất - Công ty Trà và Cà phê Tâm Châu, 33457 là mã của sản phẩm - cà phê pha phin, 6 là mã kiểm tra.

#### ***d. Cách lựa chọn sự mã hoá***

Có nhiều phương pháp mã hoá khác nhau, có thể sử dụng kết hợp nhiều kiểu để đạt chất lượng mã tốt nhất. Việc lựa chọn mã hoá cần dựa vào những yếu tố sau:

- Nghiên cứu việc sử dụng mã sau này
- Nghiên cứu số lượng các đối tượng mã hoá để lường trước được sự phát triển.
- Nghiên cứu sự phân bố thống kê các đối tượng để phân bố theo lớp.
- Tìm xem đã có những mã hoá nào được dùng trước đó cho các đối tượng này để kế thừa.
- Thỏa thuận người dùng.
- Thử nghiệm trước khi dùng chính thức.

### 1.2.3. Từ điển dữ liệu

Từ điển dữ liệu (còn gọi từ điển yêu cầu) là bộ phận của tư liệu trong PT\_TK, nó là văn phạm giả hình thức mô tả nội dung của các sự vật, đối tượng theo định nghĩa có cấu trúc. Trong biểu đồ luồng dữ liệu các chức năng xử lý, kho dữ liệu, luồng dữ liệu chỉ mô tả ở mức khái quát thường là tập hợp các khoản mục riêng lẻ. Các khái quát này cần được mô tả chi tiết hoá hơn qua công cụ từ điển dữ liệu.

*Cấu tạo từ điển:* Từ điển dữ liệu là sự liên kết có tổ chức các phần tử dữ liệu thuộc hệ thống, liệt kê các mục từ chỉ tên gọi theo một thứ tự nào đó và giải thích các tên một cách chính xác, chặt chẽ, ngắn gọn để cho cả người dùng và người phân tích hiểu chung cái vào, cái ra, cái luân chuyển.

Các mục từ tương ứng với một tên gọi kèm theo các giải thích. Tên gọi có thể đi cùng với tên đồng nghĩa như “hoá đơn” (hoá đơn kiêm phiếu thu). Các giải thích bao gồm giải thích về cấu trúc (đơn giản hay phức tạp), về bản chất (liên tục hay rời rạc), về chi tiết (miền giá trị, đơn vị đo, độ chính xác,...), cũng như về mối liên hệ với các thuật ngữ khác. Nội dung các mục từ thường thay đổi theo loại đối tượng: luồng dữ liệu (hoá đơn), kho/tập dữ liệu (nhân viên), phần tử dữ



liệu hoặc thuộc tính (ngày mở tài khoản), chức năng xử lý (kiểm tra đơn hàng). Ngoài ra, trong nội dung các mục từ có cả phần định nghĩa dữ liệu.

Ký pháp mô tả nội dung cho từ điển dữ liệu tuân theo bảng 1.5:

*Bảng 1.5. Bảng ký pháp trong từ điển dữ liệu*

Kết cấu dữ liệu	Ký pháp	Ý nghĩa
Tuần tự	=	được tạo từ
Tuyển chọn	+	và
Lặp	[/]	hoặc
	*	Lặp
	hoặc	hoặc
	{ } <sup>n</sup>	Lặp n lần
	()	Dữ liệu tùy chọn
	**	Giới hạn chú thích

Các dạng biểu thị từ điển:

### 1. Dạng thuộc tính

*Bảng 1.6. Ví dụ từ điển dữ liệu dạng thuộc tính*

STT	Tên gọi	Kiểu	Cỡ	Lĩnh vực	Quy tắc, ràng buộc và khuôn dạng
1	Số phiếu xuất	Ký tự	7	Kế toán, kho	Chữ hoa hoặc số
2	Ngày xuất	Ngày tháng	8	Kế toán, kho	Ngày/tháng/năm
3	Tên hàng	Ký tự	25	Kế toán, kho	Chữ hoa + số
4	Giá xuất	Số	8	Kế toán	Đơn vị: nghìn đồng
5	Số lượng xuất	Số	5	Kho	Đơn vị: chiếc
6	Số tiền	Số	12	Kế toán	Hai số lẻ sau dấu phẩy
7	Mã kho	Ký tự	4	Kho	Chữ hoa + số

### 2. Dạng thực thể

Giả sử có tờ hoá đơn bán hàng như sau:

Công ty: Hoàng Hà

### HOÁ ĐƠN BÁN HÀNG

Số HĐ: 1199

Ngày: 01- 01- 2009

Tên khách hàng: Hương Giang

Tài khoản: KMBĐ1111983

Địa chỉ: Đường Ngựa Vàng

Tel: (043) 8460201

Loại tiền thanh toán: USD

Tỷ giá: 18780

*Bảng 1.7. Ví dụ từ điển dữ liệu dạng thực thể*

Mã thiết bị	Tên thiết bị	Mô tả	Đơn giá	Kho
CPUSK478P30	Bộ vi xử lý Pentium 4	Pentium IV 3.0C GHz (Box/512Kb/Bus800)	240	A
ASUSP4V533MX	Bo mạch chủ Pentium 4	VIA chipset/SK478/Bus533 VGA/Sound/NIC onboard, support DDRAM, Huper threading có AGP mở rộng	54	B
VGA8	Card màn hình	VGA 8Mb (2X) Ati	10	C
<b>Tổng tiền thanh toán</b>			<b>304</b>	

Người nhận

Kế toán

Thủ kho

Ta có một phần từ điển dữ liệu sau:

*Xác định một tờ hoá đơn như sau:*

HOÁ ĐƠN = Số HĐ + Ngày + Tên khách hàng + Tài khoản + Địa chỉ + Telephone + Loại tiền thanh toán + Tỷ giá + Mã thiết bị\* + tên thiết bị\* + mô tả\* + đơn giá\* + kho\* + tổng tiền thanh toán + Người nhận + Kế toán + Thủ kho.

*Xác định thông tin về khách hàng*

Khách hàng = Tên khách hàng + Tài khoản + Địa chỉ + Telephone

*Xác định thông tin về từng loại thiết bị*

Thiết bị = Mã thiết bị + tên thiết bị + mô tả + đơn giá

*Họ tên khách cần được tách tên để thuận tiện cho Tiếng Việt*

Tên khách hàng = Họ đệm + Tên

### 3. Dạng đặc tả

Từ điển xác định thuê bao di động gọi dịch vụ cố định:

Số điện thoại = 0 + mã vùng + mã dịch vụ

Mã vùng = [76/64/281/75/56/62/26/71/50/...]

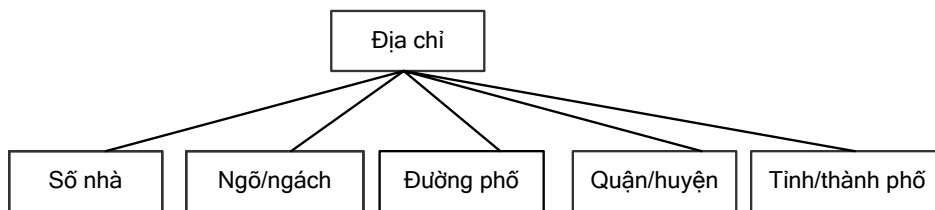
\*76 - AnGiang, 64 - Bà Rịa Vũng Tàu, 281 - Bắc Kạn, ...\*

Mã dịch vụ = 1 + số hiệu dịch vụ

Số hiệu dịch vụ = [01/02/03/04/05/.../20]

\*01 - đăng ký điện thoại đường dài trong nước, 113 - gọi cảnh sát, 114 - gọi cứu hỏa, 115 - gọi cấp cứu y tế, ....\*

### 4. Dạng đồ thị

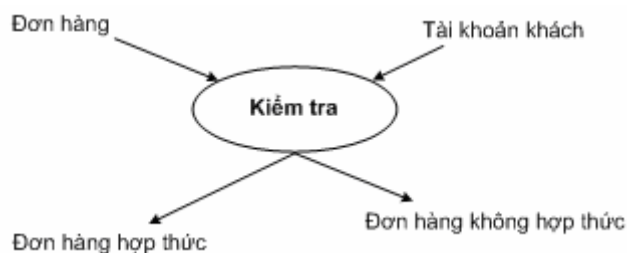


Hình 1.9. Ví dụ dạng đồ thị của từ điển dữ liệu

**5. Dạng tệp:** bao gồm tên tệp, mô tả, từ đồng nghĩa, từ hợp thành, cách tổ chức (tuần tự, nhị phân, hàm băm) và các xử lý.

**6. Dạng chức năng xử lý:** bao gồm mô tả, từ đồng nghĩa, cái vào, cái ra, tên chức năng: ví dụ như kiểm tra đơn hàng.

Lưu đồ thể hiện chức năng xử lý như hình 1.10.



Hình 1.10. Lưu đồ thể hiện chức năng xử lý

### 1.3. MÔ HÌNH DỮ LIỆU

#### 1.3.1. Khái niệm mô hình dữ liệu

Mô hình dữ liệu là cách biểu diễn các cấu trúc dữ liệu cho một CSDL dưới dạng các khái niệm. Các cấu trúc dữ liệu bao gồm các đối tượng dữ liệu, mối liên hệ giữa các dữ liệu, ngữ nghĩa của dữ liệu và các ràng buộc trên các đối tượng dữ liệu đó.

Mô hình dữ liệu không phụ thuộc vào các ràng buộc về cả phần cứng và phần mềm. Nó tập trung vào việc mô tả các dữ liệu và cách tổ chức chúng như thế nào, không tập trung vào các thao tác sẽ được thực hiện trên các dữ liệu đó (việc này được thực hiện bởi mô hình chức năng). Việc tập trung vào việc biểu diễn dữ liệu giúp cho NSD có thể “nhìn thấy” được chúng trong thế giới thực.

#### 1.3.2. Các loại mô hình CSDL

Có 3 loại mô hình CSDL:

##### *1. Mô hình CSDL quan niệm*

- Là mô hình mô tả dữ liệu của thế giới thực gắn với hoạt động nghiệp vụ của tổ chức sử dụng nó.
- Mô tả các cấu trúc và mối liên hệ giữa các đơn vị thông tin cơ bản.
- Là phương tiện để giao tiếp với NSD nhằm xác định đúng đắn và đầy đủ các yêu cầu thông tin của hệ thống.
- Hoàn toàn độc lập với mọi hệ quản trị dữ liệu và cách thức sử dụng nó.
- Cung cấp các khái niệm gắn liền với cách cảm nhận dữ liệu của NSD. Nó tập trung vào bản chất logic của biểu diễn dữ liệu, quan tâm đến cái được biểu diễn, chứ không quan tâm đến cách biểu diễn.
- Mô hình khái niệm cơ bản như mô hình E\_R. Mô hình E\_R dùng để mô tả cấu trúc logic tổng thể (lược đồ) của một CSDL bằng hình ảnh (đặc tả). Người ta quan niệm thế giới thực bao gồm tập các E

và R. Trong đó, E - “sự vật”/”đối tượng” tức là thực thể trong thế giới thực và phải phân biệt được, còn R là mối quan hệ (Relationship) giữa một nhóm các thực thể.

*Ví dụ:* người, tài khoản ngân hàng,...

Thực thể được mô tả trong CSDL bằng tập các thuộc tính. Thuộc tính định danh dùng để xác định duy nhất một bản thể/thể hiện/dòng dữ liệu.

- Mô hình E\_R chủ yếu xác định cấu trúc dữ liệu hơn là diễn đạt các quy tắc nghiệp vụ (*1. Toàn vẹn thực thể: trong một bản thể tồn tại định danh duy nhất và duy nhất, 2. Toàn vẹn tham chiếu, 3. Thuộc cùng miền giá trị, 4. Các hoạt động kích hoạt: thực hiện các ràng buộc đúng cho các giá trị các thuộc tính thuộc các thực thể có quan hệ với nhau khi tiến hành các thao tác dữ liệu*).

2. *Mô hình CSDL logic:* cung cấp khái niệm cho NSD có thể hiểu được và không xa so với cách tổ chức dữ liệu trong máy tính. Chúng che dấu một số chi tiết về việc lưu trữ dữ liệu nhưng có thể cài đặt trực tiếp trên hệ thống máy tính. Mô hình dữ liệu logic cho 1 hệ quản trị CSDL:

+ Mô tả các dữ liệu bằng cách sử dụng các kí hiệu tương ứng với mô hình dữ liệu mà 1 hệ quản trị CSDL xây dựng trên nó.

+ Có 4 loại mô hình dữ liệu logic: mô hình dữ liệu phân cấp, mạng, quan hệ, hướng đối tượng.

+ Hiện nay, được tổ chức theo mô hình dữ liệu quan hệ là chủ yếu.

3. *Mô hình CSDL vật lý:* cung cấp các khái niệm mô tả chi tiết về việc các dữ liệu được lưu trữ trong máy như thế nào.

### 1.3.3. Mô hình dữ liệu mạng và mô hình dữ liệu phân cấp

Trong mô hình mạng (network) và mô hình phân cấp (hierarchy), các thực thể được biểu diễn dưới dạng các bản ghi và các liên kết được biểu diễn bởi các móc nối hay con trỏ. Về cơ bản, các móc nối có khả

năng biểu diễn các liên kết giữa một thực thể và nhiều thực thể của các kiểu thực thể liên quan. Tuy nhiên, với cách nhìn dữ liệu theo kiểu phân cấp bởi cấu trúc cây, các bản ghi dữ liệu chỉ có ý nghĩa khi nó được xem xét trong cấu trúc cây nếu không có phần tử bậc trên nó. Tuy nhiên, với mô hình phân cấp, chúng ta có một cách rất tự nhiên để mô hình hoá những cấu trúc phân cấp thực sự của thế giới thực. Cấu trúc biểu diễn mối quan hệ của CSDL theo kiểu cây phân cấp tồn tại trên máy lớn Mainframe.

Mô hình dữ liệu mạng có cấu trúc tổng quát hơn so với mô hình dữ liệu phân cấp bởi một bản ghi dữ liệu đã cho có thể có một số bất kỳ các phần tử bậc trên trực tiếp cũng như một số bất kỳ các phần tử phụ thuộc trực tiếp. Cấu trúc nội tại của mô hình mạng phức tạp hơn so với mô hình phân cấp nhưng mô hình mạng lại đối xứng hơn mô hình phân cấp. Điểm bất lợi chủ yếu của mô hình này đó là sự quá phức tạp trong bản thân mô hình và trong các thao tác xử lý dữ liệu. Do vậy, những NSD và người viết chương trình sẽ phải hao tổn nhiều thời gian vào việc giải quyết những vấn đề nảy sinh trong quá trình phát triển và bảo trì chương trình. Cùng với cấu trúc quan hệ, cấu trúc mạng được sử dụng trên cả máy Mainframe và PC.

Về phương diện cấu trúc CSDL, mối liên hệ giữa các dữ liệu có mối quan hệ chặt chẽ với sự liên hệ của thực thể với thực thể:

- Nếu mỗi khoá có một bản ghi duy nhất đi kèm với nó và nếu với mỗi bản ghi có một giá trị khoá chính duy nhất thì mối liên hệ giữa các dữ liệu được gọi là 1-1. Ví dụ: số chứng minh thư (CMT) và công dân có mối quan hệ 1-1

- Nếu mỗi giá trị khoá có nhiều bản ghi duy nhất đi kèm, mối liên hệ này là 1-N. Trường họ có nhiều bản ghi đi kèm, mỗi bản ghi chỉ có một khoá

- Nếu là mối liên hệ không duy nhất cho cả khoá và bản ghi, nó được xem là mối liên hệ N-N. Ví dụ mối quan hệ giữa nam sinh và nữ sinh.

Như vậy, trong cấu trúc cây phân cấp: có mối liên hệ 1-1/1-N (không có mối liên hệ N-N như các cấu trúc khác). Ở đây không nhóm dữ liệu vào bản ghi mà nhóm thành segment. Mỗi Segment có 1/n trường. Quan hệ giữa các segment là quan hệ cha - con, con phụ thuộc cha. Vì vậy phải truy xuất segment cha tương ứng trước khi truy xuất đến con (truy xuất segment gốc đầu tiên). Trong cấu trúc CSDL mạng: có mối quan hệ N-N. Quan hệ cha con gọi là quan hệ chủ và thành viên. Trong cấu trúc này có các bản ghi. Vì vậy truy xuất bản ghi chủ trước, xong mới đến thành viên. Trong CSDL mạng, cần tạo ra khoá/chỉ mục trên các bản ghi thành viên và truy xuất chúng nhờ khoá/chỉ mục.

*Chúng ta thấy 2 loại cấu trúc trên bộc lộ một số nhược điểm sau:*

- Không có khái niệm cho phép NSD tạo lập, xây dựng và thay đổi các bản ghi trong CSDL
- NSD phụ thuộc hoàn toàn vào nhóm phát triển hệ thống để hiệu chỉnh và tạo báo cáo. Người lập trình phải có nhiều kỹ năng để viết những chương trình truy xuất CSDL
- Trong mô hình cây phân cấp: các câu vấn tin đòi hỏi người lập trình phải có nhiều kinh nghiệm
- Trong CSDL mạng, nhiều CSDL không có tiện ích vấn tin dành cho NSD hoặc nếu có thì cũng rất hạn chế.

#### **1.3.4. Mô hình dữ liệu quan hệ và mô hình dữ liệu hướng đối tượng**

*Mô hình dữ liệu quan hệ* được Codd đề xuất năm 1970. Nó đã tạo ra một cuộc cách mạng mới trong lĩnh vực CSDL và nhanh chóng thay thế các mô hình dữ liệu trước đó.

Mô hình dữ liệu quan hệ tương đối đơn giản và dễ hiểu. Mô hình dữ liệu quan hệ là mô hình dữ liệu mà cốt lõi của nó là CSDL quan hệ. Một CSDL quan hệ là một tập của một hoặc nhiều quan hệ, trong đó mỗi một quan hệ là một bảng. Mô hình quan hệ sử dụng một tập các

bảng để biểu diễn cả dữ liệu và mối liên hệ giữa những dữ liệu này. Bảng có n cột và mỗi cột có một tên duy nhất.

*Các ưu điểm cơ bản của mô hình dữ liệu quan hệ so với các mô hình khác:*

- Được xem là mô hình có cơ sở toán học vững chắc nhất.
- Đang giữ vai trò thống trị và là nền tảng cho hàng loạt các hệ quản trị CSDL nổi tiếng và phổ biến như Oracle, DB2, MS SQL server, Access.
- Tương đối đơn giản dễ hiểu, một CSDL quan hệ là một tập quan hệ, biểu diễn đơn giản bằng bảng gồm các cột và hàng giúp NSD mới làm quen với CSDL có thể hiểu được nội dung CSDL.
- Cung cấp các khái niệm chặt chẽ được hình thức hoá cao, cho phép áp dụng các công cụ toán học, các thuật toán tối ưu trên dữ liệu.
- Tuy được trừu tượng hoá cao nhưng chỉ dừng ở mức logic, nghĩa là độc lập với mức vật lý - mức cài đặt, với các thiết bị lưu trữ nên đảm bảo được tính độc lập giữa dữ liệu và chương trình ứng dụng.
- Nhờ biểu diễn dữ liệu thống nhất, nó có thể cho phép sử dụng các ngôn ngữ thao tác dữ liệu ở mức cao, dễ sử dụng và dễ chuẩn hoá.
- NSD có thể tạo và sửa các bản ghi trong CSDL, có thể tạo báo cáo, độc lập với nhóm phát triển phần mềm.
- CSDL được lưu trong bảng (ứng với tệp) chứa các hàng (ứng với các bản ghi) và các cột (ứng với các trường).
- Tạo ra các bảng và các cột bằng cách sử dụng ngôn ngữ định nghĩa dữ liệu (DDL - Data Definition Languages). NSD có thể học dễ dàng được ngôn ngữ này trong thời gian ngắn cùng với việc có thể tạo và sửa bản ghi, tạo báo cáo, sửa đổi cấu trúc, trích rút dữ liệu. Nhờ vậy NSD có thể làm việc độc lập với nhóm lập trình.
- Những CSDL quan hệ thông dụng nhất đều sử dụng ngôn ngữ SQL (Structured Query Language).



*Mô hình dữ liệu hướng đối tượng* so với các mô hình dữ liệu khác là một mô hình cho phép biểu diễn các đối tượng khá tự nhiên và sát với thực tế hơn cả. Các đối tượng phức tạp và các liên kết giữa chúng được phản ánh đầy đủ và giàu ngữ nghĩa. Do có khả năng biểu diễn dữ liệu khá tốt, nên mô hình dữ liệu hướng đối tượng cũng hỗ trợ các ngôn ngữ thao tác dữ liệu hướng đối tượng mạnh, cho phép người sử dụng dễ dàng biểu diễn cả các truy vấn khá phức tạp. Tuy nhiên, cho đến nay, chưa có một sự thống nhất và chuẩn hoá đối với các khái niệm của mô hình hướng đối tượng cũng như chưa có một cơ sở toán học tốt hình thức hoá ở mức cao, chặt chẽ đối với mô hình này, do vậy nó vẫn được tiếp tục nghiên cứu và phát triển trong tương lai.

Ngày nay, người ta đã nghĩ đến một mô hình dữ liệu kết hợp giữa mô hình dữ liệu quan hệ và mô hình dữ liệu hướng đối tượng gọi tắt là mô hình dữ liệu quan hệ - hướng đối tượng nhằm tận dụng được các ưu điểm của mỗi mô hình trên. Nhưng đây thực sự là một vấn đề khá khó khăn nên mô hình này vẫn đang trong giai đoạn nghiên cứu và thử nghiệm.

### **1.3.5. Phương pháp xây dựng mô hình dữ liệu**

Có hai quy trình chính được sử dụng để tạo nên hai mô hình dữ liệu mức khái niệm và mức logic dựa trên cách tiếp cận thực thể - liên kết và phương pháp mô hình hoá đối tượng.

#### ***Quy trình phát triển 1 mô hình khái niệm dữ liệu***

Các bước xây dựng theo phương pháp mô hình:

1. Chính xác hoá dữ liệu,
2. Xác định các thực thể,
3. Xác định các mối quan hệ,
4. Vẽ mô hình khái niệm dữ liệu,
5. Chuẩn hoá (loại các thuộc tính lặp) và thu gọn (thực thể treo có 1 thuộc tính nằm ở phía 1 trong mối quan hệ 1-N)

**Quy trình phát triển 1 mô hình dữ liệu logic**

1. Chuyển mô hình khái niệm dữ liệu sang mô hình dữ liệu logic-Mô hình quan hệ (dựa vào 3 quy tắc chuyển mô hình khái niệm dữ liệu sang hệ lược đồ quan hệ),
2. Chuẩn hoá đến 3NF và hợp nhất các quan hệ (nếu cần),
3. Vẽ sơ đồ E\_R tương ứng với mô hình quan hệ đã đạt 3NF

Trong quá trình phân tích, thiết kế và xây dựng CSDL, có thể nói quá trình xây dựng mô hình dữ liệu chiếm nhiều thời gian và công sức nhất. Bởi mục đích của nó là đảm bảo chắc chắn rằng tất cả các đối tượng dữ liệu trong bài toán phải được trình bày đầy đủ và chính xác. Người thiết kế CSDL có thể sử dụng mô hình dữ liệu như một bản thiết kế chi tiết có thể tiến hành thiết kế một CSDL mức vật lý.

## Chương 2

# MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ

### 2.1. CÁC KHÁI NIỆM CƠ BẢN

#### 2.1. 1. Định nghĩa quan hệ

Cho  $R = \{a_1, a_2, \dots, a_n\}$  là một tập hữu hạn, không rỗng các thuộc tính.

Mỗi thuộc tính  $a_i$  có một miền giá trị là  $D_{a_i}$ . Khi đó  $r$  - một tập các bộ  $\{h_1, h_2, \dots, h_m\}$  được gọi là một quan hệ trên  $R$ , với  $h_j$  ( $j=1, 2, \dots, m$ ) là một hàm:  $h_j : R \rightarrow \bigcup_{a_i \in R} D_{a_i}$  sao cho:  $h_j(a_i) \in D_{a_i}$

Có thể biểu diễn quan hệ  $r$  ở dạng bảng:

*Bảng 2.1. Biểu diễn quan hệ dạng bảng*

	$a_1$	$a_2$	...	$a_n$
$h_1$	$h_1(a_1)$	$h_1(a_2)$	...	$h_1(a_n)$
$h_2$	$h_2(a_1)$	$h_2(a_2)$	...	$h_2(a_n)$
			...	
$h_m$	$h_m(a_1)$	$h_m(a_2)$	...	$h_m(a_n)$

*Nhận xét:*

- Định nghĩa này là quan trọng. Toàn bộ lý thuyết cơ sở dữ liệu dựa trên định nghĩa này. Nó được coi là hạt nhân của cơ sở dữ liệu quan hệ.

- Vì  $h_1, h_2, \dots, h_m$  là các thành phần trong tập hợp quan hệ trong  $r$ . Trong mô hình quan hệ, không chấp nhận hai bản ghi trùng nhau trong một file dữ liệu.

*Ví dụ*

Để lưu giữ thông tin về các mặt hàng, cần sử dụng bảng thiết bị như sau:

*Bảng 2.2. Biểu diễn quan hệ THIETBI*

Mã thiết bị	Tên thiết bị	Mô tả	Đơn giá	Kho
CPUSK478P30	Bộ vi xử lý Pentium 4	Pentium IV 3.0C GHz (Box/512Kb/Bus800)	240	A
ASUSP4V533MX	Bo mạch chủ Pentium 4	VIA chipset/SK478/Bus533 VGA/Sound/NIC onboard, support DDRAM, Huper threading có AGP mở rộng	54	B
VGA8	Card màn hình	VGA 8Mb (2X) Ati	10	C
HDDQ30	Ổ đĩa cứng	Maxtor Quantum 30Gb 5400rpm	53	D

Có thể quy định kích thước cho các thuộc tính (các trường) như sau:

*Bảng 2.3. Bảng thuộc tính của quan hệ THIETBI*

Tên thuộc tính	Kiểu	Kích thước
Mã thiết bị	Ký tự	15
Tên thiết bị	Ký tự	30
Mô tả	Ký tự	100
Đơn giá	Số	5
Kho	Ký tự	2

Có nghĩa là chúng ta xác định miền giá trị cho các thuộc tính:

- Mã thiết bị là một chuỗi ký tự có độ dài không quá 15
- Tên thiết bị là một chuỗi ký tự có độ dài không quá 30
- Mô tả là một chuỗi ký tự có độ dài không quá 100
- Đơn giá là một số nguyên không quá 5
- Kho là một chuỗi ký tự không quá 2

Như vậy ta có tập thuộc tính:

THIẾT BỊ = {Mã thiết bị, Tên thiết bị, Mô tả, Đơn giá, Kho}

Ở đây:

- $D_{\text{Mã thiết bị}}$  là tập các xâu ký tự độ dài không quá 15
- $D_{\text{Tên thiết bị}}$  là tập các xâu ký tự độ dài không quá 30
- $D_{\text{Mô tả}}$  là tập các xâu ký tự độ dài không quá 100
- $D_{\text{Đơn giá}}$  là tập các số có độ dài không quá 5
- $D_{\text{Kho}}$  là tập các xâu ký tự độ dài không quá 2

Khi đó chúng ta có các quan hệ  $r = \{h_1, h_2, h_3, h_4\}$ , ở đây đối với bản ghi thứ nhất (dòng thứ nhất) chúng ta có:

$h_1(\text{Mã thiết bị}) = \text{CPUSK478P30}$

$h_1(\text{Tên thiết bị}) = \text{Bộ vi xử lý Pentium 4}$

$h_1(\text{Mô tả}) = \text{Pentium IV 3.0C GHz (Box/512Kb/Bus800)}$

$h_1(\text{Đơn giá}) = 240$

$h_1(\text{Kho}) = A$

Khi viết  $R(a_1, a_2, \dots, a_n)$  ta có một lược đồ quan hệ  $R$

### 2.1.2. Định nghĩa phụ thuộc hàm

Khái niệm về phụ thuộc hàm trong một quan hệ là rất quan trọng trong việc thiết kế mô hình dữ liệu. Năm 1970 E.F Codd đã mô tả phụ thuộc hàm trong mô hình dữ liệu quan hệ, nhằm giải quyết việc phân rã không mất thông tin.

Cho  $R = \{a_1, a_2, \dots, a_n\}$  là tập các thuộc tính,  $r = \{h_1, h_2, \dots, h_m\}$  là một quan hệ trên  $R$ , và  $A, B \subseteq R$  ( $A, B$  là tập cột hay tập thuộc tính). Khi đó ta nói  $A$  xác định hàm cho  $B$  hay  $B$  phụ thuộc hàm vào  $A$  trong  $r$  (ký pháp  $A \xrightarrow[r]{f} B$ ) nếu:  $(\forall h_i, h_j \in r) ((\forall a \in A) (h_i(a) = h_j(a)) \Rightarrow (\forall b \in B) (h_i(b) = h_j(b)))$  nghĩa là đối số trùng nhau thì hàm có cùng

giá trị. Đặt  $F_r = \{(A, B): A, B \subseteq R, A \xrightarrow[r]{f} B\}$ . Lúc đó  $F_r$  được gọi là họ đầy đủ các phụ thuộc hàm của  $r$ .

*Nhận xét:*

Ta có thể thấy rằng  $B$  mà phụ thuộc hàm vào  $A$ , nếu hai dòng bất kỳ mà các giá trị của tập thuộc tính  $A$  mà bằng nhau từng cặp một, thì kéo theo các giá trị trên tập thuộc tính  $B$  cũng phải bằng nhau từng cặp một.

*Ví dụ:*

Xét quan hệ:

*Bảng 2.4. Quan hệ THISINH*

SBD	Hoten	Điachi	Tinh	Khuvuc
PD711001	Nguyễn Thái Bình	12 Bản Nhân	Lạng Sơn	0
PD711002	Trần Nam Ninh	3 Kim Mã	Hà Nội	3
PD711003	Lê Thanh Hoa	53 Hai Bà Trưng	Hà Nội	3
PD711004	Vũ Thúy Hồng	89 Đồng Đăng	Lạng Sơn	0
PD711005	Phạm Như Em	40 Trần Hưng Đạo	Hải Dương	2

Trong quan hệ THISINH, dựa vào định nghĩa phụ thuộc hàm của quan hệ ta có:

$\{\text{tinh}\} \rightarrow \{\text{khuvuc}\}$

$\{\text{sbd}\} \rightarrow \{\text{hoten, diachi, tinh, khuvc}\}$

*Ý nghĩa:* Khái niệm phụ thuộc hàm miêu tả một loại ràng buộc (phụ thuộc dữ liệu) xảy ra tự nhiên nhất giữa các tập thuộc tính.

### 2.1.3. Hệ tiên đề Armstrong

Gọi  $F$  là tập xác định các phụ thuộc hàm đối với lược đồ quan hệ  $R$  và  $X \rightarrow Y$  là một phụ thuộc hàm.  $X, Y \subseteq R$ . Nói rằng  $X \rightarrow Y$  được

suy diễn logic từ  $F$  nếu mỗi quan hệ  $r$  trên  $R$  đều thoả mãn phụ thuộc hàm của  $F$  thì cũng thoả mãn  $X \rightarrow Y$ . Chẳng hạn  $F = \{A \rightarrow B, B \rightarrow C\}$  thì  $A \rightarrow C$  suy ra từ  $F$ . Gọi  $F^+$  là *bao đóng* (closure) của  $F$ , tức là tập tất cả các phụ thuộc hàm được suy diễn logic từ  $F$ . Nếu  $F = F^+$  thì  $F$  là *họ đầy đủ* (full family) của các phụ thuộc hàm.

Để có thể xác định khoá của một lược đồ quan hệ và các suy diễn logic giữa các phụ thuộc hàm cần thiết phải tính được  $F^+$  từ  $F$ . Do đó đòi hỏi phải có các hệ tiên đề. Tập các quy tắc của hệ tiên đề được Armstrong (1974) đưa ra, thường được gọi là hệ tiên đề Armstrong.

**Hệ tiên đề Armstrong (Armstrong's Axioms):**

Cho  $R = \{a_1, \dots, a_n\}$  là tập các thuộc tính.  $X, Y, Z \subseteq R$ .

Hệ tiên đề Armstrong có 3 tính chất cơ bản sau:

- A1 (phản xạ): Nếu  $Y \subseteq X$  thì  $X \rightarrow Y$
- A2 (tăng trưởng): Nếu  $Z \subseteq R$  và  $X \rightarrow Y$  thì  $XZ \rightarrow YZ$ .

Trong đó ký hiệu  $XZ$  là hợp của hai tập  $X$  và  $Z$  thay cho ký hiệu  $X \cup Z$ .

- A3 (bắc cầu): Nếu  $X \rightarrow Y$  và  $Y \rightarrow Z$  thì  $X \rightarrow Z$ .

**Bổ đề 1:**

Hệ tiên đề Armstrong là đúng. Có nghĩa  $F$  là tập các phụ thuộc hàm đúng trên quan hệ  $r$ . Nếu  $X \rightarrow Y$  là tập các phụ thuộc hàm được suy diễn từ  $F$  nhờ hệ tiên đề Armstrong thì  $X \rightarrow Y$  là đúng trên quan hệ  $r$ .

**Bổ đề 2:**

Hệ tiên đề Armstrong có thể được mở rộng thành:

1. Tính phản xạ (Reflexivity):

$Y \subseteq X \subseteq R \Rightarrow X \rightarrow Y$  (phụ thuộc tầm thường: trivial dependency)

2. Tính tăng trưởng (augmentation):

$X \rightarrow Y, Z \subseteq R \Rightarrow XZ \rightarrow YZ$

3. Tính “bắc cầu”:

$$X \rightarrow Y, Y \rightarrow Z \Rightarrow X \rightarrow Z$$

4. Tính hợp (union rule):

$$X \rightarrow Y, X \rightarrow Z \Rightarrow X \rightarrow YZ$$

5. Tính “nở trái”:

$$X \rightarrow Y; X, Y, Z \in R \Rightarrow XZ \rightarrow Y$$

6. Tính phân rã/tách (decomposition rule) (thu phải):

$$X \rightarrow Y, Z \subseteq Y \Rightarrow X \rightarrow Z$$

7. Tính “hợp dọc”:

$$X \rightarrow Y, P \rightarrow Q \Rightarrow XP \rightarrow YQ$$

8. Tính tựa bắc cầu:

$$X \rightarrow Y, WY \rightarrow Z \Rightarrow WX \rightarrow Z$$

9. Tính tách và hợp

$$\text{Cho } A_1, A_2, \dots, A_p \subseteq R. X \rightarrow A_1 A_2 \dots A_p \Leftrightarrow X \rightarrow A_i$$

10. Tính tích luỹ

$$X \rightarrow YZ, Z \rightarrow AV \Rightarrow X \rightarrow YZA$$

*Nhận xét:*

- Người ta còn viết  $(A, B)$  hay  $A \rightarrow B$  thay cho  $A \xrightarrow[r]{f} B$ .

- Lúc đó tập hợp tất cả  $(A, B)$  như thế xác định một họ  $f$  trên  $R$  khi chúng thỏa mãn hệ tiên đề Armstrong.

-  $F_r$  là một họ  $f$  trên  $R$ . Armstrong đã chứng minh một kết quả rất quan trọng như sau: nếu  $Y$  là một họ  $f$  bất kỳ thì tồn tại một quan hệ  $r$  trên  $R$  sao cho  $F_r = Y$ .

- Việc nghiên cứu phụ thuộc hàm không lệ thuộc vào các quan hệ (bảng) cụ thể. Vì vậy, áp dụng được các công cụ toán nhằm sáng tỏ *cấu trúc logic của mô hình dữ liệu quan hệ*



- Có nhiều quan hệ khác nhau nhưng các họ đầy đủ các phụ thuộc hàm của chúng lại như nhau

Ví dụ:

Cho  $r_1, r_2$  là các quan hệ (bảng 2.5):

Bảng 2.5

$r_1$		$r_2$	
A	B	A	B
0	0	0	0
1	1	1	1
2	1	2	1
3	2	3	1

Có thể thấy rằng  $r_1$  và  $r_2$  khác nhau nhưng  $F_{r_1} = F_{r_2}$  vì chỉ có  $A \rightarrow B$

#### 2.1.4. Định nghĩa hàm đóng

Một hàm  $L: P(R) \rightarrow P(R)$ , ( $P(R)$  là tập các tập con của  $R$ ) được gọi là hàm đóng trên  $R$  nếu với mọi  $A, B \in P(R)$ :

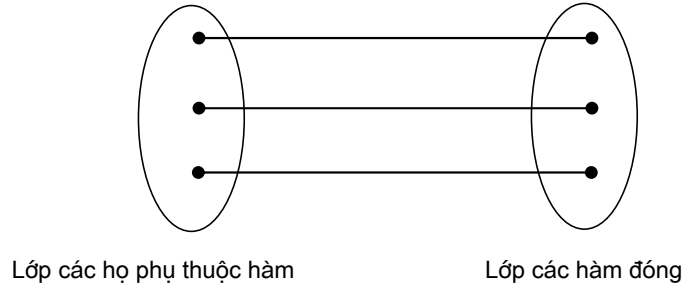
- $A \subseteq L(A)$
- Nếu  $A \subseteq B$  thì  $L(A) \subseteq L(B)$
- $L(L(A)) = L(A)$ .

**Định lý:**

Nếu  $F$  là một họ  $f$  và chúng ta đặt  $L_F(A) = \{a: a \in R: (A, \{a\}) \in F\}$  thì  $L_F$  là một hàm đóng. Ngược lại nếu  $L$  là một hàm đóng thì tồn tại duy nhất một họ  $f: F$  trên  $R$  sao cho  $L = L_F$ .

Ở đây,  $F = \{(A, B): A, B \subseteq R, B \subseteq L(A)\}$ .

Như vậy, chúng ta thấy có sự tương ứng 1-1 giữa lớp các hàm đóng và lớp các họ  $f$ . Sự tương ứng này được minh họa bằng hình 2.1:



Hình 2.1.

Chúng ta có thể dễ dàng chứng minh được 3 tính chất sau của hàm đóng:

- +  $L(X)L(Y) \subseteq L(XY)$
- +  $L(X \cap Y) \subseteq L(X) \cap L(Y)$
- +  $L(L(X)Y) = L(XL(Y)) = L(XY)$

### 2.1.5. Định nghĩa sơ đồ quan hệ

Cho trước  $R = \{a_1, a_2, \dots, a_n\}$  là tập các thuộc tính.

Khi đó  $s$  là một sơ đồ quan hệ,  $s = \langle R, F \rangle$

$$F = \begin{cases} A_1 \rightarrow B_1 \\ A_2 \rightarrow B_2 \\ \dots \\ \dots \\ A_t \rightarrow B_t \end{cases}$$

Ở đây  $A_i, B_i \subseteq R$  ( $i = 1, \dots, t$ ) và  $A_i \rightarrow B_i$  là phụ thuộc hàm

Ví dụ: Cho sơ đồ quan hệ  $s = \langle R, F \rangle$ , với  $R = \{a_1, a_2, a_3, a_4\}$

$$F = \begin{cases} \{a_1\} \rightarrow \{a_3, a_4\} & \text{cột 1 xác định hàm với cột 3, cột 4} \\ \{a_2\} \rightarrow \{a_3\} & \text{cột 2 xác định hàm với cột 3} \\ \dots & \\ \dots & \\ \{a_3\} \rightarrow \{a_4\} & \text{cột 3 xác định hàm với cột 4} \end{cases}$$

### 2.1.6. Định nghĩa bao đóng

#### a. Định nghĩa bao đóng của tập thuộc tính trên sơ đồ quan hệ

Giả sử  $F$  là tập các phụ thuộc hàm trên sơ đồ quan hệ  $s = \langle R, F \rangle$ . Gọi  $F^+$  là tập tất cả các phụ thuộc hàm có thể suy diễn logic từ  $F$  bởi các luật của hệ tiên đề Armstrong.

Giả sử  $A \subseteq R$ , ký hiệu  $A^+ = \{a: A \rightarrow \{a\} \in F^+\}$ .  $A^+$  được gọi là bao đóng của  $A$  trên  $s$ .

*Ý nghĩa:*  $A^+$  là tập tất cả các thuộc tính  $a$  mà phụ thuộc hàm  $A \rightarrow \{a\}$  có thể suy diễn logic từ  $F$  nhờ hệ tiên đề Armstrong.

Mệnh đề  $A \rightarrow B \in F^+$  nếu và chỉ nếu  $B \subseteq A^+$  đúng nhờ vào bổ đề dưới đây.

*Bổ đề*

$X \rightarrow Y$  suy diễn từ hệ tiên đề Armstrong khi và chỉ khi  $Y \subseteq X^+$ .

*Chứng minh:*

Điều kiện đủ: Giả sử  $Y = A_1 \dots A_n$  với  $A_1, \dots, A_n$  là tập các thuộc tính và  $Y \subseteq X^+$ .

Từ định nghĩa  $X^+$  có  $X \rightarrow A_i$ . áp dụng tiên đề Armstrong cho mọi  $i$ , luật hợp suy ra  $X \rightarrow Y$ . Ngược lại, giả sử có  $X \rightarrow Y$ , áp dụng hệ tiên đề Armstrong cho mỗi  $i$  có  $X \rightarrow A_i$ ,  $A_i \in Y$  nhờ luật tách. Từ đó suy ra  $Y \subseteq X^+$ .

Chúng ta có thể dễ dàng chứng minh được mệnh đề quan trọng sau:

*Mệnh đề*

Phép lấy bao đóng của tập thuộc tính là một hàm đóng thỏa mãn các tính chất sau:

1. Tính phản xạ:  $X \subseteq X^+$

Thật vậy, Từ  $a \in X$  suy ra  $X \rightarrow a \in F^+$  suy ra  $a \in X^+$

2. Tính đơn điệu: nếu  $X \subseteq Y$  thì  $X^+ \subseteq Y^+$

Thật vậy, theo thuật toán tính bao đóng

3. Tính lũy đẳng:  $X^{++} = X^+$

Thật vậy, theo thuật toán tính bao đóng

4.  $X^+Y^+ \subseteq (XY)^+$

Thật vậy, dựa vào  $L(X)L(Y) \subseteq L(XY)$

5.  $(X^+Y)^+ = (XY^+)^+ = (XY)^+$

Thật vậy, dựa vào  $L(L(X)Y) = L(XL(Y)) = L(XY)$

6.  $X \rightarrow Y$  khi và chỉ khi  $Y \subseteq X^+$  (bổ đề)

7.  $X \rightarrow Y$  khi và chỉ khi  $Y^+ \subseteq X^+$

Thật vậy:

Điều kiện cần:

+  $X \rightarrow Y$  theo tính chất 6 suy ra  $Y \subseteq X^+$ ,

+ dựa theo tính chất 2 suy tiếp ra  $Y^+ \subseteq X^{++}$

+ dựa theo tính chất 3 suy ra  $Y^+ \subseteq X^{++} = X^+$  tức là  $Y^+ \subseteq X^+$

Điều kiện đủ:

$Y^+ \subseteq X^+$ , dựa theo tính chất 1 suy ra  $Y \subseteq Y^+ \subseteq X^+$ , tức là  $Y \subseteq X^+$

8.  $X \rightarrow X^+, X^+ \rightarrow X$

Thật vậy:  $X \rightarrow X^+$  (theo định nghĩa bao đóng)

$X^+ \rightarrow X$  (theo tính chất 1)

9.  $X^+ = Y^+$  khi và chỉ khi  $X \rightarrow Y$  và  $Y \rightarrow X$

Thật vậy:

Điều kiện cần:

$X \subseteq X^+ = Y^+$  suy ra  $X \subseteq Y^+$  suy ra  $Y \rightarrow X$

$Y \subseteq Y^+ = X^+$  suy ra  $Y \subseteq X^+$  suy ra  $X \rightarrow Y$

Điều kiện đủ:

$Y \rightarrow X$  suy ra  $X \subseteq Y^+$  suy ra  $X^+ \subseteq Y^{++} = Y^+$  suy ra  $X^+ \subseteq Y^+$  (1)

$$\begin{aligned} \text{Mặt khác } X \rightarrow Y \text{ suy ra } Y \subseteq X^+ \text{ suy ra } Y^+ \subseteq X^{++} = X^+ \\ \text{suy ra } Y^+ \subseteq X^+ \end{aligned} \quad (2)$$

Kết hợp (1) và (2) suy ra  $X^+ = Y^+$

**b. Định nghĩa bao đóng của một tập thuộc tính trên một quan hệ**

Giả sử  $r = \{h_1, h_2, \dots, h_m\}$  là một quan hệ trên  $R = \{a_1, a_2, \dots, a_n\}$ .

Ta đặt:  $A_r^+ = \{a : A \xrightarrow[r]{f} \{a\}\}$

$A_r^+$  được gọi là bao đóng của A trên r.

**2.1.7. Định nghĩa khoá của quan hệ, sơ đồ quan hệ, họ F**

**a. Định nghĩa khoá**

Giả sử  $r = \{h_1, h_2, \dots, h_m\}$  là một quan hệ,  $s = \langle R, F \rangle$  là một sơ đồ quan hệ, trong đó  $R = \{a_1, a_2, \dots, a_n\}$  là tập các thuộc tính, F là tập xác định các phụ thuộc hàm trên R. Gọi Y là một họ f trên R và  $A \subseteq R$ . Khi ấy A là một **khóa** của r (tương ứng là một khoá của s, một khóa của Y) nếu:  $A \xrightarrow[r]{f} R$  ( $A \rightarrow R \in F^+$ ,  $(A, R) \in Y$ ), nghĩa là A phải thoả mãn các tính chất sau đây:

- Với bất kỳ hai bộ  $h_1, h_2 \in r$  đều tồn tại một thuộc tính  $a \in A$  sao cho  $h_1(a) \neq h_2(a)$ . Nói cách khác, không tồn tại hai bộ mà có giá trị bằng nhau trên mọi tập thuộc tính của A. Điều kiện này có thể viết  $t_1(A) \neq t_2(A)$ . Do vậy, mỗi giá trị của A xác định là duy nhất. Khi biết giá trị thuộc tính trong A sẽ biết được các giá trị của thuộc tính khác.

- Theo định nghĩa của Codd: Nếu có hai dòng bằng nhau trên các giá trị của khoá A thì sẽ kéo theo bằng nhau trên tất cả các cột còn lại. Như vậy sẽ có hai cột bằng nhau, điều này không thể có được và nếu có thì đây là dữ liệu nhầm lẫn.

- Dễ dàng thấy rằng, với lược đồ  $R(A,B,C,D)$  mà  $A \rightarrow C$ ,  $B \rightarrow D$  thì tập hợp các thuộc tính ở vế trái của phụ thuộc hàm  $\{A,B\}$  chính là khoá vì tất cả các thuộc tính của lược đồ đều có mặt trên phụ thuộc hàm.

### b. Khoá tối thiểu

Chúng ta gọi  $A$  ( $A \subseteq R$ ) là một **khóa tối thiểu** của  $r$  (tương ứng của  $s$ , của  $Y$ ) nếu:

- +  $A$  là một khoá của  $r(s, Y)$  tức  $A \rightarrow R$
- + Bất kỳ một tập con thực sự của  $A$  không là khoá của  $r(s, Y)$  hay không tồn tại  $A'$  là tập con thực sự của  $A$  ( $A' \subset A$ ) mà  $A' \rightarrow R$ .

*Nhận xét:*

Khoá chính là hình ảnh của cột mã số hay số thứ tự (vì số thứ tự không thể trùng nhau được).

Khóa đóng một vai trò rất quan trọng vì nhờ có nó người ta mới tìm kiếm được (tìm kiếm bản ghi). Phép toán tìm kiếm bản ghi trong file dữ liệu là phép toán quan trọng nhất vì chỉ sau khi tìm kiếm xong thì người ta mới tiến hành loại bỏ bản ghi ấy hoặc bổ sung một bản ghi mới vào trước hoặc sau bản ghi mà ta đã tìm được.

Dù rằng dễ thấy  $A$  có thể chính bằng  $R$  nhưng người ta vẫn phải đi tìm khóa tối thiểu, tức là khóa nhỏ nhất mà không thể nhỏ hơn được nữa để việc so sánh các giá trị khóa với nhau trong quá trình tìm kiếm bản ghi là nhanh nhất.

Một sơ đồ quan hệ có thể có nhiều khóa, thậm chí còn có nhiều khóa tối thiểu.

*Ví dụ:*

Bảng 2.6. Quan hệ BAN\_HANG

Mã hàng	Tên hàng	Số lượng (chiếc)
VT0001	Vô tuyến	1000
TL0002	Tủ lạnh	500
RA2012	Radio	2000

Trong bảng trên mã số mặt hàng (Mã hàng) là khóa và là khóa tối thiểu. Mỗi giá trị mã hàng đều xác định duy nhất một loại mặt hàng trong quan hệ BAN\_HANG.

**c. Hệ Sperner**

Ký hiệu  $K_r, K_s, K_Y$  là tập tất cả các khoá tối thiểu của  $r(s, Y)$ .

Ta gọi  $K$  (với  $K$ -tập con của  $P(R)$ ) là một **hệ Sperner** trên  $R$  nếu với  $\forall A, B \in K \Rightarrow \neg(A \subseteq B)$ ,

Để thấy  $K_r, K_s, K_Y$  là hệ Sperner trên  $R$ .

**d. Tập phản khoá**

Giả sử  $K = \{K_1, K_2, \dots, K_t\}$  là một hệ Sperner trên  $R = \{a_1, a_2, \dots, a_n\}$ . Ta định nghĩa tập các phản khoá của  $K$  ký hiệu  $K^{-1}$  như sau:

$$K^{-1} = \{A \subset R : (B \in K) \Rightarrow (B \not\subset A) \text{ và } (A \subset C) \Rightarrow (\exists B \in K) (B \subseteq C)\}$$

(tức là không chứa tập con thuộc  $K$  và nói rộng nó ra sẽ có 1 phần tử của  $K$  lọt vào)

Tập phản khoá đóng vai trò quan trọng trong quá trình nghiên cứu cấu trúc logic của các họ phụ thuộc hàm, khóa, dạng chuẩn, quan hệ Armstrong, đặc biệt đối với các bài toán tổ hợp trong mô hình dữ liệu quan hệ.

*Nhận xét:*

Nếu  $K$  đóng vai trò là một tập các khoá tối thiểu của một sơ đồ quan hệ nào đó thì theo định nghĩa  $K^{-1}$  là tập tất cả các tập không phải là khoá lớn nhất.

- Dễ thấy rằng  $K^{-1}$  cũng là hệ Sperner trên  $R$ .

- Ở đây ta luôn giả thiết rằng nếu một hệ Sperner đóng vai trò tập các khoá tối thiểu (tập các phản khoá), thì hệ Sperner này không rỗng (Không chứa  $R$ ).

- Demertovics.J. đã chứng tỏ rằng nếu  $K$  là một hệ Sperner tùy ý, thì tồn tại một sơ đồ quan hệ  $s$  sao cho  $K_s = K$  và tồn tại quan hệ  $r$  để  $K_r = K$ .

**2.1.8. Hệ bằng nhau**

Giả sử  $r = \{h_1, h_2, \dots, h_m\}$  là một quan hệ trên  $R = \{a_1, a_2, \dots, a_n\}$

Đặt  $E_r = \{E_{ij}: 1 \leq i < j \leq |r|\}$  trong đó  $E_{ij} = \{a \in R: h_i(a) = h_j(a)\}$ ,  $|r| = m$ ,  $E_r$  được gọi là hệ bằng nhau của  $r$

Giả sử  $M_r = \{E_r/E_{ij} \text{ với } E_{ij}: \exists E_{pq}: E_{ij} \subset E_{pq}\}$ . Khi đó  $M_r$  được gọi là hệ bằng nhau cực đại của  $r$ .

*Nhận xét:*

Hệ bằng nhau và hệ bằng nhau cực đại đóng một vai trò quan trọng trong các thuật toán thiết kế cũng như mối quan hệ giữa các lớp quan hệ và lớp các phụ thuộc hàm trong quá trình nghiên cứu cấu trúc logic của lớp các phụ thuộc hàm.

*Ví dụ:* Cho quan hệ  $r$  (bảng 2.7).

Bảng 2.7: Quan hệ  $r$

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
1	1	0	1	0
1	0	0	3	0
3	1	1	3	1
5	1	0	3	0

$$E_r = \begin{cases} E_{12} = \{a_1, a_3, a_5\} \\ E_{13} = \{a_2\} \\ E_{14} = \{a_2, a_3, a_5\} \\ E_{23} = \{a_4\} \\ E_{24} = \{a_3, a_4, a_5\} \\ E_{34} = \{a_2, a_4\} \end{cases}$$

$$M_r = \begin{cases} \{a_1, a_3, a_5\} \\ \{a_2, a_4\} \\ \{a_2, a_3, a_5\} \\ \{a_3, a_4, a_5\} \end{cases}$$

Vậy  $M_r$  là hệ bằng nhau cực đại của  $r$ .



## 2.2. MỘT SỐ THUẬT TOÁN LIÊN QUAN ĐẾN KHOÁ

### 2.2.1. Một số thuật toán liên quan đến bao đóng

Một vấn đề thường xuyên xảy ra là đối với một sơ đồ quan hệ cho trước  $(s = \langle R, F \rangle)$ , và một phụ thuộc hàm  $A \rightarrow B$ , chúng ta muốn biết  $A \rightarrow B$  có là phân tử của  $F^+$  hay không. Để trả lời câu hỏi này chúng ta cần tính bao đóng  $F^+$  của tập các phụ thuộc hàm  $F$ .

Tuy nhiên tính  $F^+$  trong trường hợp tổng quát là rất khó khăn và tốn kém thời gian vì các tập phụ thuộc hàm thuộc  $F^+$  là rất lớn cho dù  $F$  có thể là nhỏ. Chẳng hạn,  $F = \{A \rightarrow B_1, A \rightarrow B_2, \dots, A \rightarrow B_n\}$ , khi đó  $F^+$  bao gồm cả những phụ thuộc hàm  $A \rightarrow Y$  với  $Y \subseteq \{B_1 \cup B_2 \cup \dots \cup B_n\}$ , như vậy ta sẽ có  $2^n$  tập con  $Y$ . Trong khi đó việc tính bao đóng của tập thuộc tính  $A$  lại không khó. Theo kết quả đã trình bày ở trên thì việc kiểm tra  $A \rightarrow B \in F^+$  sẽ được thế bởi việc tính  $A^+$ .

**Thuật toán 1: Tính bao đóng của một tập các thuộc tính đối với tập các phụ thuộc hàm trên sơ đồ quan hệ**

Vào:  $s = \langle R, F \rangle$  là một sơ đồ quan hệ

trong đó:

$R = (a_1, a_2, \dots, a_n)$  là tập hữu hạn các thuộc tính.

$F$  là tập các phụ thuộc hàm và  $A \subseteq R$

Ra:  $A^+$  là bao đóng của  $A$  đối với  $F$ .

Nhớ rằng  $A^+ = \{a: A \rightarrow \{a\} \in F^+\}$ .

$A \rightarrow B \in F^+$  nếu và chỉ nếu  $B \subseteq A^+$ .

*Phương pháp:*

Lần lượt tính các tập thuộc tính  $A_0, A_1$  như sau:

1.  $A_0 = A$
2.  $A_i = A_{i-1} \cup \{a\}$  nếu  $\exists (C \rightarrow D) \in F, \{a\} \in D$  và  $C \subseteq A_{i-1}$

3. Rõ ràng  $A = A_0 \subseteq A_1 \subseteq \dots \subseteq A_i \subseteq R$  và  $R$  hữu hạn nên tồn tại  $i$  sao cho  $A_i = A_{i+1}$

Khi ấy thuật toán dừng và  $A_i$  chính là  $A^+$

*Ví dụ:*

Xét sơ đồ quan hệ  $s = \langle R, F \rangle$

trong đó:

$$F = \begin{cases} \{c\} \rightarrow \{t\} \\ \{h, r\} \rightarrow \{c\} \\ \{h, t\} \rightarrow \{r\} \\ \{c, s\} \rightarrow \{g\} \\ \{h, s\} \rightarrow \{r\} \end{cases} \quad R = \{c, t, h, r, s, g\}$$

Tính  $\{h, r\}^{+}$ ?

$$A_0 = \{h, r\}$$

$$A_1 = \{h, r, c\} \text{ do } \{h, r\} \rightarrow \{c\} \in F$$

$$A_2 = \{h, r, c, t\} \text{ do } \{c\} \rightarrow \{t\} \in F$$

$$A_3 = \{h, r, c, t\} = A_2$$

$$\text{Vậy } \{h, r\}^+ = \{h, r, c, t\}$$

**Thuật toán 2: Tính bao đóng cho một tập bất kỳ trên quan hệ  $r$**

Vào:  $r = \{h_1, h_2, \dots, h_m\}$  là một quan hệ trên  $R = \{a_1, a_2, \dots, a_n\}$ ,  $A \subseteq R$

Ra:  $A_r^+$

Bước 1: Từ  $r$  xây dựng một tập  $E_r = \{E_{ij} : 1 \leq i < j \leq m\}$

$$E_{ij} = \{a : a \in R \text{ và } h_i(a) = h_j(a)\}$$

Bước 2: Từ  $E_r$  xây dựng một tập

$$M = \{B \in P(R) : \text{Tồn tại } E_{ij} \in E_r : E_{ij} = B\}$$

ở đây  $P(R)$  là tập các tập con của  $R$

Bước 3:  $A_r^+$  được tính như sau:

$$A_r^+ = \begin{cases} \cap B & \text{nếu tồn tại } B \in M: A \subseteq B, \text{ (giao của tất cả các} \\ & \text{tập } A \subseteq B \text{ trong } M \text{ chứa nó (A))} \\ R & \text{ngược lại} \end{cases}$$

Có thể thấy rằng  $E_{ij} = \emptyset$  (trong trường hợp hai dòng  $i$  và  $j$  không có cột nào trùng nhau về giá trị, tức là chúng khác nhau hoàn toàn.

Không bao giờ có  $E_{ij} = R$  (có nghĩa rằng  $E_{ij}$  = toàn bộ không gian), vì nếu xảy ra thì có hai dòng trùng nhau, theo định nghĩa quan hệ thì không cho phép có hai dòng trùng nhau.

Ví dụ: Xét quan hệ  $r$  (bảng 2.8)

Bảng 2.8

A	B	C	D	E
1	1	0	0	0
0	1	0	1	1
0	0	2	0	0
2	0	2	0	1

Tính  $\{B, C\}_r^+$

$E_{12} = \{B, C\}$ ,  $E_{13} = \{D, E\}$ ,  $E_{14} = \{D\}$

$E_{23} = \{A\}$ ,  $E_{24} = \{E\}$

$E_{34} = \{B, C, D\}$

$M = \{\{A\}, \{B, C\}, \{B, C, D\}, \{D\}, \{D, E\}, \{E\}\}$

Vậy  $\{B, C\}_r^+ = \{B, C\} \cap \{B, C, D\} = \{B, C\}$

### 2.2.2. Một số thuật toán liên quan đến khoá

Khi giải quyết các bài toán thông tin quản lý, người ta thường sử dụng các hệ quản trị cơ sở dữ liệu mà trong đó chứa cơ sở dữ liệu quan hệ. Các phép xử lý đối với bài toán này thường là tìm kiếm bản ghi sau đó thêm bản ghi mới, thay đổi nội dung bản ghi hoặc xoá bản ghi.

Trong các thao tác trên, việc tìm kiếm bản ghi là rất quan trọng. Muốn tìm được bản ghi trong file dữ liệu thì chúng ta phải xây dựng khoá của file dữ liệu đó.

Có hai thuật toán tìm khoá của quan hệ và lược đồ quan hệ. Tìm khoá ở đây chính là tìm khoá tối thiểu.

**Thuật toán 3: Tìm khoá tối thiểu của một quan hệ**

Vào:  $r = \{h_1, \dots, h_m\}$  là một quan hệ trên tập thuộc tính  $R = \{a_1, \dots, a_n\}$

Ra:  $K$  là một khoá tối thiểu của  $r$ .

*Phương pháp:*

Bước 1: Tính  $E_r = \{A_1, A_2, \dots\}$  trong đó  $E_r$  là các hệ bằng nhau.

Bước 2: Tính  $M_r$  là các hệ bằng nhau cực đại.

Bước 3: Lần lượt tính các thuộc tính  $K_0, K_1, \dots, K_n$  theo qui tắc:

$K_0 = R = \{a_1, \dots, a_n\}$  hoặc  $K_0$  là một khoá đã biết.

$$K_i = \begin{cases} K_{i-1} - \{a_i\} & \text{nếu: không tồn tại } A \in M_r: K_{i-1} - \{a_i\} \subseteq A \\ K_{i-1} & \text{trong trường hợp ngược lại} \end{cases} \quad (\text{bao } K_i)$$

Bước 4: Đặt  $K = K_n$ . Khi đó  $K$  là khoá tối thiểu.

*Nhận xét:* Nếu ta thay đổi thứ tự các thuộc tính của  $r$  bằng thuật toán này chúng ta có thể tìm được một khoá tối thiểu khác.

*Ví dụ:* Cho quan hệ  $r$  (bảng 2.9):

Bảng 2.9

A	B	C	D	E
1	1	0	1	0
1	0	0	1	1
2	1	1	0	2
1	0	1	1	0

Bước 1:  $E_{12} = \{A, C, D\}$        $E_{13} = \{B\}$        $E_{14} = \{A, D, E\}$

$E_{23} = \emptyset$        $E_{24} = \{A, B, D\}$        $E_{34} = \{C\}$

Bước 2:  $M_r = \{\{A, C, D\}, \{A, D, E\}, \{A, B, D\}\}$

Bước 3: (KH:  $P(M_r)$  - phân tử của  $M_r$ )

$K_0 = \{A, B, C, D, E\}$

Xét  $K_1 = K_0 - \{A\} = \{B, C, D, E\} \not\subset P(M_r) \Rightarrow K_1 = \{B, C, D, E\}$

Xét  $K_2 = K_1 - \{B\} = \{C, D, E\} \not\subset P(M_r) \Rightarrow K_2 = \{C, D, E\}$

Xét  $K_3 = K_2 - \{C\} = \{D, E\} \subset P(M_r) \Rightarrow K_3 = \{C, D, E\}$

Xét  $K_4 = K_3 - \{D\} = \{C, E\} \not\subset P(M_r) \Rightarrow K_4 = \{C, E\}$

Xét  $K_5 = K_4 - \{E\} = \{C\} \subset P(M_r) \Rightarrow K_5 = \{C, E\}$

Vậy  $\{C, E\}$  là một khoá tối thiểu của  $r$ .

Cũng ví dụ trên nhưng ta thay đổi tập thuộc tính theo thứ tự là:  $A, C, E, B, D$ . Ta đi tìm khoá tối thiểu của  $r$ .

Bước 1 và bước 2 giống như trên

Bước 3:

$K_0 = \{A, C, E, B, D\}$

Xét  $K_1 = K_0 - \{A\} = \{C, E, B, D\} \not\subset P(M_r) \Rightarrow K_1 = \{C, E, B, D\}$

Xét  $K_2 = K_1 - \{C\} = \{E, B, D\} \not\subset P(M_r) \Rightarrow K_2 = \{E, B, D\}$

Xét  $K_3 = K_2 - \{E\} = \{B, D\} \subset P(M_r) \Rightarrow K_3 = K_2 = \{E, B, D\}$

Xét  $K_4 = K_3 - \{B\} = \{E, D\} \subset P(M_r) \Rightarrow K_4 = K_3 = \{E, B, D\}$

Xét  $K_5 = K_4 - \{D\} = \{E, B\} \not\subset P(M_r) \Rightarrow K_5 = \{E, B\}$

Vậy khoá tối thiểu của  $r$  là  $\{E, B\}$ .

Như vậy thay đổi thứ tự của các thuộc tính, ta sẽ có những tập khoá tối thiểu khác nhau.

Ví dụ: Cho quan hệ  $r$  (bảng 2.10):

Bảng 2.10

A	B	C	D	E
2	2	1	4	1
1	2	1	2	2
1	1	2	2	1
3	1	3	1	2

Bước 1:  $E_{12} = \{B, C\}$        $E_{13} = \{E\}$        $E_{14} = \emptyset$   
 $E_{23} = \{A, D\}$        $E_{24} = \{E\}$        $E_{34} = \{B\}$

Bước 2:  $M_r = \{\{B, C\}, \{E\}, \{A, D\}\}$

Bước 3:

\* Các thuộc tính được lấy theo thứ tự  $r_1 = \{A, B, C, D, E\}$

$K_0 = R = \{A, B, C, D, E\}$

Xét  $K_1 = K_0 - \{A\} = \{B, C, D, E\} \not\subset P(M_r) \Rightarrow K_1 = \{B, C, D, E\}$

Xét  $K_2 = K_1 - \{B\} = \{C, D, E\} \not\subset P(M_r) \Rightarrow K_2 = \{C, D, E\}$

Xét  $K_3 = K_2 - \{C\} = \{D, E\} \not\subset P(M_r) \Rightarrow K_3 = \{D, E\}$

Xét  $K_4 = K_3 - \{D\} = \{E\} \subset P(M_r) \Rightarrow K_4 = K_3 = \{D, E\}$

Xét  $K_5 = K_4 - \{E\} = \{D\} \subset P(M_r) \Rightarrow K_5 = K_4 = \{D, E\}$

Vậy khoá tối thiểu của  $r$  là  $\{D, E\}$ .

\* Các thuộc tính được lấy theo thứ tự  $r_2 = \{E, D, C, B, A\}$

$K_0 = R = \{E, D, C, B, A\}$

Xét  $K_1 = K_0 - \{E\} = \{D, C, B, A\} \not\subset P(M_r) \Rightarrow K_1 = \{D, C, B, A\}$

Xét  $K_2 = K_1 - \{D\} = \{C, B, A\} \not\subset P(M_r) \Rightarrow K_2 = \{C, B, A\}$

Xét  $K_3 = K_2 - \{C\} = \{B, A\} \not\subset P(M_r) \Rightarrow K_3 = \{B, A\}$

Xét  $K_4 = K_3 - \{B\} = \{A\} \subset P(M_r) \Rightarrow K_4 = K_3 = \{B, A\}$

Xét  $K_5 = K_4 - \{A\} = \{B\} \subset P(M_r) \Rightarrow K_5 = K_4 = \{B, A\}$

Vậy khoá tối thiểu khác của quan hệ là  $\{B, A\}$ .

**Thuật toán 4: Tìm khoá tối thiểu cho một sơ đồ quan hệ**

Vào: sơ đồ quan hệ  $s = \langle R, F \rangle$

trong đó:  $F$  là tập các phụ thuộc hàm

$R = \{a_1, \dots, a_n\}$  là tập các thuộc tính

Ra:  $K$  là tối thiểu của  $s$

Phương pháp:

Tính liên tiếp các tập thuộc tính  $K_0, K_1, \dots, K_n$  như sau:

$$K_0 = R = \{a_1, \dots, a_n\}$$

$$K_i = \begin{cases} K_{i-1} & K_{i-1} \text{ nếu } K_{i-1} - \{a_i\} \rightarrow R \notin F^+ \\ K_{i-1} - \{a_i\} & K_{i-1} - \{a_i\} \text{ nếu ngược lại.} \end{cases}$$

$K = K_n$  là khoá tối thiểu.

Ta có thể dùng công thức tương đương:

$$K_i = \begin{cases} K_{i-1} - \{a_i\} & \text{nếu } \{K_{i-1} - a_i\}^+ = R \\ K_{i-1} & \text{nếu ngược lại.} \end{cases}$$

Nhận xét:

- Thay đổi thứ tự các thuộc tính của  $R$  bằng thuật toán trên chúng ta có thể tìm được một khoá tối thiểu khác.

- Nếu như đã biết  $A$  là một khoá nào đó thì có thể đặt  $K_0 = A$ , ta vẫn tìm ra được khoá tối thiểu và thời gian tìm nhanh hơn.

Ví dụ: Giả sử  $s = \langle F, R \rangle$  là một lược đồ quan hệ trong đó:

$$R = \{a, b, c, d\}$$

$$F = \{ \{a, b\} \rightarrow \{d\}, \{c\} \rightarrow \{b\} \}$$

Tìm khoá tối thiểu của sơ đồ quan hệ.

Áp dụng thuật toán trên ta có:

$$+ K_0 = R = \{a, b, c, d\}$$

$$+ \text{Tính } K_1$$

Xét  $K_1 = K_0 - \{a\} = \{b, c, d\}$

$$\{b, c, d\}^+ = \{b, c, d\} \neq R$$

Vậy  $K_1 = \{a, b, c, d\}$ . ( $K_1 = K_0$ )

+ Tính  $K_2$

Xét  $K_2 = K_1 - \{b\} = \{a, c, d\}$

$$\{a, c, d\}^+ = \{a, b, c, d\} = R$$

Vậy  $K_2 = \{a, c, d\}$

+ Tính  $K_3$

Xét  $K_3 = K_2 - \{c\} = \{a, d\}$

$$\{a, d\}^+ = \{a, d\} \neq R$$

Vậy  $K_3 = \{a, c, d\}$  ( $K_3 = K_2$ )

+ Tính  $K_4$

Xét  $K_4 = K_3 - \{d\} = \{a, c\}$

$$\{a, c\}^+ = \{a, b, c, d\} = R$$

Vậy  $K_4 = \{a, c\}$

Vậy khoá tối thiểu là  $\{a, c\}$ .

### 2.2.3. Thuật toán xác định các thuộc tính cơ bản

Giả sử  $r$  là một quan hệ trên  $R$  và  $K_r$  là tập tất cả các khoá tối thiểu của  $r$ . Ta nói  $a$  là *thuộc tính cơ bản* của  $r$  nếu tồn tại một khoá tối thiểu  $K$  ( $K \in K_r$ ) để  $a$  là một phân tử của  $K$ .

Nếu  $a$  không là thuộc tính cơ bản,  $a$  sẽ được gọi là *thuộc tính thứ cấp*.

Đối với lược đồ quan hệ, ta cũng có định nghĩa tương tự.

#### **Thuật toán 5: Tìm tất cả các thuộc tính cơ bản của một quan hệ trên $R$**

Vào:  $r = \{h_1, \dots, h_m\}$  là một quan hệ trên  $R$ .

Ra:  $V$  là tập tất cả các thuộc tính cơ bản của  $r$ .



*Phương pháp:*

Bước 1: Từ  $r$  ta xây dựng một tập  $E_r = \{E_{ij} : m \geq j > i \geq 1\}$

Trong đó  $E_{ij} = \{a \in R : h_i(a) = h_j(a)\}$

Bước 2: Từ  $E_r$  ta xây dựng tập

$M = \{B \in P(R) : \exists E_{ij} \in E_r : E_{ij} = B\}$

Bước 3: Từ  $M$  ta xây dựng tập

$M_r = \{B \in M : \forall B' \in M : B \not\subset B'\}$

(chỉ lấy tập nào trong  $M$  mà không có tập khác bao nó, tức là bỏ bớt tập con thực sự)

Bước 4: Xây dựng tập  $V = R - \bigcap B$  mà  $B \in M_r$ .

*Ví dụ:*

Xét quan hệ (bảng 2.11)

*Bảng 2.11*

A	B	C	D
1	2	1	1
2	2	1	0
1	1	0	0
2	1	0	1

Bước 1: Tính  $E_r$

$E_{12} = \{B, C\}$

$E_{13} = \{A\}$

$E_{14} = \{D\}$

$E_{23} = \{D\}$

$E_{24} = \{A\}$

$E_{34} = \{B, C\}$

Bước 2: Tính  $M = \{\{B, C\}, \{A\}, \{D\}\}$

Bước 3: Tính  $M_r = \{\{B, C\}, \{A\}, \{D\}\}$ . Vì giao của các tập con trong  $M_r$  bằng rỗng nên sang,

Bước 4:  $V = R - \phi = \{A, B, C, D\}$ .

### 2.3. CÁC PHÉP TOÁN XỬ LÝ FILE DỮ LIỆU

Cho 2 file dữ liệu sau (bảng 2.12):

Bảng 2.12

File r			File t		
A	B	C	A	D	E
a	b	c	a	b	c
a	a	d	b	c	d
a	d	c	a	d	c
			a	g	f

### 2.3.1. Phép hợp ( $r \cup t$ ) (Union)

Giả sử  $r, t$  là 2 file dữ liệu có cùng số cột  $n$  khi đó file dữ liệu là hợp 2 file  $r$  và  $t$  ký pháp là  $r \cup t$ , gồm  $n$  cột bao gồm các bản ghi, các dòng của cả file  $r$  và file  $t$ . Những dòng giống nhau chỉ lấy một lần. Nếu  $r$  và  $t$  là các file có tên các cột khác nhau thì cột của quan hệ hợp không có tên.

Ví dụ:

Với  $r$  và  $t$  như trên ta có  $r \cup t$  là (bảng 2.13):

Bảng 2.13

a	b	c
a	a	d
a	d	c
b	c	d
a	g	f

### 2.3.2. Phép trừ ( $r - t$ ) (Difference)

Giả sử  $r$  và  $t$  là 2 file dữ liệu có cùng  $n$  cột.

File dữ liệu gọi là hiệu, ký pháp  $r - t$  là một file dữ liệu  $n$  cột bao gồm các bản ghi của  $r$  nhưng không có mặt trong  $t$ . Nếu  $r$  và  $t$  có tên các cột khác nhau thì file dữ liệu hiệu không có tên các cột.

Ví dụ: Bảng 2.14

Bảng 2.14

$r-t=$

a	a	d

### 2.3.3. Phép giao ( $r \cap t$ ) (Intersection)

Giả sử  $r$  và  $t$  là 2 file dữ liệu cùng  $n$  cột. File dữ liệu giao  $r \cap t$  bao gồm file dữ liệu có  $n$  cột gồm các bản ghi có cả trong  $t$  và  $r$ . Nếu file dữ liệu có các cột khác nhau thì file dữ liệu giao không có tên các cột.

Ví dụ:

Với  $r$  và  $t$  cho trước ta có bảng 2.15.

Bảng 2.15

$r \cap t=$

a	b	C
a	d	c

### 2.3.4. Phép tích Đề-các ( $r \times t$ ) (Cartesian product)

Cho  $r$  là file dữ liệu trên  $R_1 = \{a_1, a_2, \dots, a_n\}$

$t$  là file dữ liệu trên  $R_2 = \{b_1, b_2, \dots, b_m\}$

Khi đó file dữ liệu tích Đề-các ký pháp  $r \times t$  là file dữ liệu gồm  $n + m$  cột. Đó là  $n$  cột của  $r$  và  $m$  cột của  $t$ . Các dòng của tích Đề-các có dạng  $x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m$ . Ở đây  $x_1, x_2, \dots, x_n \in r$  và  $y_1, y_2, \dots, y_m \in t$ .

Số lượng các dòng của tệp tích  $r \times t$  bằng số lượng dòng của  $r$  nhân với số lượng dòng của  $t$ . Ký hiệu:  $|r \times t| = |r| \times |t|$

Với  $r$  và  $t$  như trước ta có bảng 2.16:

Bảng 2.16

r x t =

rA	B	C	tA	D	E
a	b	c	a	b	c
a	b	c	b	c	d
a	b	c	a	d	c
a	b	c	a	g	f
a	a	d	a	b	c
a	a	d	b	c	d
a	a	d	a	d	c
a	a	d	a	g	f
a	d	c	a	b	c
a	d	c	b	c	d
a	d	c	a	d	c
a	d	c	a	g	f

### 2.3.5. Phép chiếu (Projection)

Giả sử  $r$  là file dữ liệu có tập cột là  $R=\{a_1, a_2, \dots, a_n\}$  khi đó phép chiếu ký hiệu là  $\Pi$  lên tập  $r$  là  $\Pi_{i_1, i_2, \dots, i_p}(r)$ ,  $i$  là số thứ tự lấy trong tập từ 1 đến  $n$ .  $j = 1, 2, \dots, p$  (chỉ số  $p \leq n$ )

Khi đó ta thực hiện phép chiếu như sau: Giữ lại  $p$  cột có số hiệu là  $i_1, i_2, \dots, i_p$  và loại bỏ các dòng trùng nhau.

Ví dụ:

Với  $r$  và  $t$  cho trước ta có bảng 2.17:

Bảng 2.17

 $\Pi_{1,2}(r)=$ 

A	B
a	b
a	a
a	d

### 2.3.6. Phép chọn (Restriction)

Giả sử  $r$  là file dữ liệu trên  $R=\{a_1, a_2, \dots, a_n\}$  phép chọn hình thành file dữ liệu ký pháp  $\delta_F(r)$  đó là file dữ liệu có cùng số cột như  $r$  nhưng số dòng chỉ bao gồm các dòng của  $r$  thoả mãn điều kiện  $F$ .

Ở đây  $F$  là biểu thức có điều kiện gồm:

- Các toán hạng (tên cột và hằng)
- Các phép toán quan hệ số học ( $<, =, >, \neq, ?$ )
- Các phép toán logic ( $\wedge, \vee, \neg$  là phép và, hoặc, phủ định)
- Hằng được đặt trong dấu nháy.

Ví dụ: Với  $r$  đã cho trước (trong bảng 2.12)

$F$  là điều kiện  $A = 'a' \wedge B = 'b'$

khi đó  $\delta_F(r)$  là phép chọn được thực hiện bằng cách rút ra các bản ghi thoả mãn  $F$  từ  $r$  và loại bỏ các bản ghi dòng trùng nhau.

Bảng 2.18

A	B	C
a	b	c

### 2.3.7. Phép chia (Division)

Giả sử  $r$  là file dữ liệu  $n$  cột

$t$  là file dữ liệu  $m$  cột, trong đó  $n > m$  và  $t \neq \phi$

File dữ liệu thường ký pháp là:  $r \div t$  được tạo ra như sau:

$$A_1 = \Pi_{1, \dots, n-m}(r)$$

$$A_2 = A_1 \times t$$

$$A_3 = A_2 - r$$

$$A_4 = \Pi_{1, 2, \dots, n-m}(A_3)$$

$$r \div t = A_1 - A_4$$

Như vậy:  $r \div t = \Pi_{1, \dots, n-m}(r) - \Pi_{1, \dots, n-m}((\Pi_{1, \dots, n-m}(r) \times t) - r)$

Phép chia là phép phức tạp nhất mà Codd đề ra, bản chất của phép này là sàng lọc trong n-m cột đầu của file dữ liệu bị chia (chính là r) mà không bị ràng buộc bởi file dữ liệu chia (t). Có thể thấy rằng cách tính file dữ liệu thường chỉ dùng phép chiếu và phép tích Đề-các.

Ví dụ:

Cho file r như trong bảng 2.19:

Bảng 2.19

A	B	C	D
a	b	c	d
a	b	e	f
b	c	e	f
e	d	c	d
e	d	e	f
a	b	d	c
b	b	b	b

Bảng 2.20: file dữ liệu t

E	F
c	d
e	f

Bảng 2.21: file dữ liệu  $r \div t$

E	F
a	b
e	d

### 2.3.8. Phép nối điều kiện 2 file dữ liệu (Join $\theta$ )

Giả sử r là file dữ liệu n cột

t là file dữ liệu m cột

$\theta$  là một trong những quan hệ số học:  $<$ ,  $\leq$ ,  $>$ ,  $\geq$ ,  $\neq$ . Khi đó ký pháp phép nối 2 file dữ liệu r và t có điều kiện là  $r \bowtie_{\theta} t$

Khi đó kết quả của phép nối  $\theta$  của 2 file dữ liệu r và t là file dữ liệu có n+m cột bao gồm các bản ghi của tích Đề các  $r \times t$ , trong đó các bản ghi này thoả mãn quan hệ số học  $\theta$ .

Ví dụ: Với r và t đã cho, ta có phép nối  $\theta$ :

Bảng 2.22

$r.A = t.A$

r.A	B	C	t.A	D	E
a	b	c	a	b	c
a	b	c	a	d	c
a	b	c	a	g	f
a	a	d	a	b	c
a	a	d	a	d	c
a	a	d	a	g	f
a	d	c	a	b	c
a	d	c	a	d	c
a	d	c	a	g	f

### 2.3.9. Phép nối 2 file dữ liệu (Join)

Giả sử  $r$  là file dữ liệu  $n$  cột

$t$  là file dữ liệu  $m$  cột và có  $q$  cột có tên trùng nhau  $A_1, \dots, A_q$ , ta ký pháp phép nối  $r$  với  $t$  như sau:  $r \bowtie t$

Phép nối được thực hiện như sau:

1. Tính  $r \times t$
2. Thực hiện chọn trong tích Đề-các lấy ra các bản ghi thoả mãn điều kiện các cột có tên trùng nhau thì có cùng giá trị.
3. Dùng phép chiếu loại bỏ  $q$  cột (tên giống nhau), mỗi cặp giống nhau chỉ giữ lại một

Như vậy số cột của phép nối là  $n+m-q$

$$r \bowtie t = \Pi_{i_1 i_2 \dots i_{n+m-q}} (\delta_{rA_1=tA_1 \wedge \dots \wedge rA_q=tA_q} (r \times t))$$

Ví dụ:

Bảng 2.23: file  $r$ 

A	B	C
0	1	0
1	1	1
0	1	1

Bảng 2.24: file  $t$ 

D	A	E
0	0	0
1	1	0

Bảng 2.25

r x t=

rA	B	C	D	tA	E
0	1	0	0	0	0
0	1	0	1	1	0
1	1	1	0	0	0
1	1	1	1	1	0
0	1	1	0	0	0
0	1	1	1	1	0

Bảng 2.26

 $(\delta_{rA1=tA1 \wedge \dots \wedge rAq=tAq}) = \Delta$ 

rA	B	C	D	tA	E
0	1	0	0	0	0
1	1	1	1	1	0
0	1	1	0	0	0

Bảng 2.27

 $\Pi(\Delta) =$ 

rA	B	C	D	E
0	1	0	0	0
1	1	1	1	0
0	1	1	0	0

**2.3.10. Ví dụ áp dụng**

Có bảng tổng kết bán hàng đồ chơi

Ngày tháng	Mã hàng	Tên hàng	Đơn giá	Số lượng
2905	MH1	Tủ lạnh	2	2
	MH2	Quạt điện	5	3
	MH6	Bếp ga	2	2
			Thành tiền	23
			Đã thanh toán	7

Từ đây, xây dựng bảng ứng với lược đồ quan hệ sau:

BANHANG (Ngày tháng, Mã hàng, Tên hàng, Đơn giá, Số lượng, tổng theo ngày, thanh toán).

Khoá tối thiểu là {ngày tháng, mã hàng}

Người ta tách được thành các bảng chuẩn 3 như sau:



## 1. KHOILUONG

Ngày tháng	Mã hàng	Số lượng
2905	MH1	2
2905	MH3	3
2905	MH6	2
3005	MH2	3
3105	MH1	4
3105	MH4	3
3105	MH9	2

## 2. DOANH SO

Ngày tháng	Tổng	Thanh toán
2905	23	7
3005	12	3
3105	32	21

## 3. HÀNG

Mã hàng	Tên hàng
MH1	Tủ lạnh
MH3	Quạt điện
MH6	Bếp ga
MH2	Bàn là
MH4	Xe máy
MH9	Nồi cơm điện

## 4. MATHANG

Tên hàng	Đơn giá
Tủ lạnh	2
Quạt điện	5
Bếp ga	2
Bàn là	4
Xe máy	6
Nồi cơm điện	3

## 1. Tìm lại BANHANG?

BANHANG = khối lượng ▷◁ Doanh số ▷◁ Hàng ▷◁ Mặt hàng

<u>Ngày tháng</u>	<u>Mã hàng</u>	Tên hàng	Đơn giá	Số lượng	Tổng theo ngày	Thanh toán
2905	MH1	Tủ lạnh	2	2	23	7
2905	MH3	Quạt điện	5	3	23	7
2905	MH6	Bếp ga	2	2	23	7
3005	MH2	Bàn là	4	3	12	3
3105	MH1	Tủ lạnh	2	4	32	21
3105	MH4	Xe máy	6	3	32	21
3105	MH9	Nồi cơm điện	3	2	32	21

2. Cho biết *doanh số* bán ra sau ngày 29 tháng 5?

$\Pi_{\text{ngày tháng, tổng}}(\sigma_{\text{ngày tháng} > '2905'}(\text{DOANH SO}))$

Ngày tháng	Tổng
3005	12
3105	32

3. Cho biết *tên hàng* và *số lượng* đã bán trong ngày 29 tháng 5?

$\Pi_{\text{tên hàng, số lượng}}(\sigma_{\text{ngày tháng} = '2905'}(\text{KHOILUONG}) \triangleright \triangleleft \text{HANG})$

Tên hàng	Số lượng
Tủ lạnh	2
Quạt điện	3
Bếp ga	2

4. Cho biết các *ngày* mà trong đó những ngày này *doanh số* bán ra ít nhất là 20?

$\Pi_{\text{ngày tháng}}(\sigma_{\text{tổng} \geq '20'}(\text{DOANH SO}))$

Ngày tháng
2905
3105

5. Cho biết mã hàng và đơn giá của những mặt hàng bán ra trong ngày 31/5?

$\Pi_{\text{mã hàng}}(\sigma_{\text{ngày tháng} = '31/05'}(\text{KHOILUONG})) \triangleright \triangleleft \Pi_{\text{mã hàng, đơn giá}}(\text{HANG} \triangleright \triangleleft \text{MATHANG})$

Mã hàng	Đơn giá
MH1	2
MH4	6
MH9	3

6. Cho biết mã hàng và tên hàng mà đơn giá của nó nhỏ hơn 4?

$\Pi_{\text{tên hàng, mã hàng}}(\sigma_{\text{đơn giá} < '4'}(\text{MATHANG}) \triangleright \triangleleft \text{HANG})$

Tên hàng	Mã hàng
Tủ lạnh	MH1
Bếp ga	MH6
Nồi cơm điện	MH9

7. Tìm tên hàng và đơn giá của mã hàng MH1 và số lượng bán ra của mặt hàng này trong ngày 31/5?

$\Pi_{\text{tên hàng, Đơn giá, số lượng}}(\Pi_{\text{mã hàng, số lượng}}(\sigma_{\text{ngày tháng} = '31/05', \text{mã hàng} = \text{MH1}}(\text{KHOILUONG})) \triangleright \triangleleft \text{HANG} \triangleright \triangleleft \text{MATHANG})$

Tên hàng	Đơn giá	Số lượng
Tủ lạnh	2	4

*Nhận xét:*

Để tăng tốc độ truy vấn dữ liệu, chúng ta cần thực hiện theo trình tự sau:

- Xác định các bảng tham gia
- Giảm độ lớn của một tệp trước khi ghép nối (bằng cách ưu tiên phép chọn có điều kiện được thực hiện trước, thực hiện phép chiếu ngay nếu thấy không ảnh hưởng đến kết quả bài toán)

### 2.3.11. Các chiến lược tối ưu tổng quát

Vì sao một số câu hỏi truy vấn lại thực hiện quá lâu như vậy?

Trong các ngôn ngữ hỏi dựa trên đại số quan hệ, các câu hỏi liên quan đến phép tính tích Đề-các và phép kết nối là rất tốn kém thời gian. Có thể ví dụ, có 2 quan hệ  $r_1$  và  $r_2$  (coi hai quan hệ này là 2 tập dữ liệu). Tập  $r_1$  có  $n$  bản ghi, tập  $r_2$  có  $m$  bản ghi. Tích Đề-các của  $r_1 \times r_2$  thu được có  $n \times m$  bản ghi. Phép tính trên không những tốn kém về thời gian mà còn tốn cả về ô nhớ nữa.

Dựa vào ví dụ sau, chúng ta sẽ làm rõ hơn tư tưởng cần thiết phải thao tác ra sao để giảm thời gian xử lý câu hỏi, đặc biệt cần thiết nhất là phải giảm thời gian xử lý câu hỏi, giảm tới mức có thể đối với các truy xuất ngoài (tức là các thao tác với bộ nhớ thứ cấp như băng, đĩa từ)

Cho câu hỏi trên lược đồ  $R(A,B)$  và  $S(C,D)$  dưới dạng ngôn ngữ đại số quan hệ:  $\Pi_A(\sigma_{B=C \wedge D=100}(AB \times CD))$

Nếu đưa phép chọn  $D = 100$  vào bên trong phép tích Đề-các sẽ được:  $\Pi_A(\sigma_{B=C}(AB \times \sigma_{D=100}(CD)))$

Và sau đó chuyển phép chọn  $\sigma_{B=C}$  của tích Đề-các thành phép “kết nối bằng” sẽ được:  $\Pi_A(AB \bowtie_{B=C} \sigma_{D=100}(CD))$

Rõ ràng phép tính cuối cùng sẽ đỡ tốn kém thời gian hơn rất nhiều.

Do vậy, việc tổ chức lại câu hỏi như ví dụ nêu trên là một minh họa rõ ràng cho thấy số lần cần truy nhập tới bộ nhớ thứ cấp sẽ giảm đi rất nhiều. Trình tự thực hiện các phép tính sẽ đóng một vai trò quan trọng trong quá trình tổ chức câu hỏi

Ullman J.D trong các kết quả nghiên cứu công bố lần đầu tiên của mình đã trình bày 6 chiến lược tổng quan cho việc tối ưu hoá, câu hỏi như sau:

#### 1. Thực hiện phép chọn sớm như có thể.

Biến đổi câu hỏi để đưa phép chọn vào thực hiện trước nhằm làm giảm bớt kích cỡ của kết quả trung gian và do vậy chi phí phải trả giá cho việc truy nhập bộ nhớ thứ cấp cũng như lưu trữ của bộ nhớ chính sẽ nhỏ đi.

*2. Tổ hợp những phép chọn xác định với phép tích Đề-các thành phép kết nối*

Như đã biết, phép kết nối, đặc biệt là phép kết nối bằng có thể thực hiện được “rẻ” hơn là thực hiện phép tích Đề-các trên cùng các quan hệ. Nếu kết quả của tích Đề-các  $R \times S$  là đối số của phép chọn và phép chọn liên quan tới các phép so sánh giữa các thuộc tính của  $R$  và  $S$  thì rõ ràng phép tích Đề-các là phép kết nối.

*3. Tổ hợp dãy các phép tính một ngôi như các phép chọn và phép chiếu.*

Một dãy các phép một ngôi như phép chọn hoặc phép chiếu mà kết quả của chúng phụ thuộc vào các bộ của một quan hệ độc lập thì có thể nhóm các phép tính đó lại

*4. Tìm các biểu thức con chung trong một biểu thức*

Nếu kết quả của một biểu thức con chung (biểu thức xuất hiện hơn một lần) là một quan hệ không lớn và nó có thể được đọc từ bộ nhớ thứ cấp với ít thời gian thì nên tính toán trước biểu thức đó chỉ một lần. Biểu thức con chung có liên quan tới một phép kết nối thì trong trường hợp tổng quát không thể được thay đổi nhờ việc đẩy phép chọn vào trong.

Điều đáng quan tâm là các biểu thức con chung có tần số xuất hiện lớn được biểu diễn trong các view (khung nhìn) của người sử dụng vì để thực hiện các câu hỏi đó cần thay thế một biểu thức cố định cho khung nhìn.

*5. Xử lý các tệp trước*

Có hai vấn đề cần xử lý trước quan trọng cho các tệp số là sắp xếp trước các tệp và thiết lập các tệp chỉ số. Như vậy, rõ ràng khi thực hiện các phép tính có liên quan tới 2 tệp (phép tính 2 ngôi) sẽ nhanh hơn rất nhiều.

### 6. Đánh giá trước khi thực hiện tính toán

Một khi cần chọn trình tự thực hiện các phép tính trong biểu thức hoặc chọn một trong hai đối số của một phép hai ngôi cần tính toán xem chi phí thực hiện các phép tính đó (thường là số phép tính, thời gian, dung tích bộ nhớ theo một tỷ lệ giữa kích cỡ các quan hệ ...). Từ đó sẽ có được các chi phí (giá thành) phải trả cho các cách khác nhau để thực hiện các câu hỏi.

## 2.4. CÁC DẠNG CHUẨN VÀ CÁC THUẬT TOÁN LIÊN QUAN

### 2.4.1. Các định nghĩa chuẩn hoá

#### **Định nghĩa 1 (Dạng chuẩn 1 - 1NF)**

Giả sử  $r = \{h_1, h_2, \dots, h_m\}$  là file dữ liệu trên tập cột  $R = \{a_1, a_2, \dots, a_n\}$

Khi đó  $r$  là 1NF nếu các giá trị  $h_i(a_j)$  là *sơ cấp* với mọi  $i, j$

Khái niệm *sơ cấp* hiểu ở đây là giá trị  $h_i(a_j)$  ( $i = 1, \dots, m$ ;  $j = 1, \dots, n$ ) không phân chia được nữa.

*Ví dụ:*

Xét quan hệ - Trình độ ngoại ngữ

*Bảng 2.28. Quan hệ TDNN*

MNS	Hoten	Ngoaingu
A199001	Hoàng Diệu	Anh, Nhật
A199002	Quốc Trung	Anh, Trung
A199003	Chí Hiếu	Anh, Đức
A199004	Lan Anh	Anh
A199005	Phương Lan	Anh, Hàn
A199006	Bảo Quỳnh	Pháp, Nga, Anh

Có thể thấy rằng thuộc tính NGOAINGU còn có thể được chia nhỏ hơn ra thành từng ngoại ngữ một và sau đó có thể phân thành hai bộ phận là tên ngoại ngữ và trình độ ngoại ngữ. Do vậy quan hệ ngoại ngữ chưa ở dạng chuẩn 1.

**Định nghĩa 2 (Dạng chuẩn 2 - 2NF)**

Quan hệ  $r$  được gọi là dạng chuẩn 2 nếu:

- Quan hệ  $r$  là dạng chuẩn 1
- Với mọi khoá tối thiểu  $K$ :

$A \rightarrow \{a\} \notin F_r$  với  $A \subset K$  và  $a$  là thuộc tính thứ cấp.

**Định nghĩa 3 (Dạng chuẩn 3 - 3NF)**

Quan hệ  $r$  là dạng chuẩn 3 nếu:

$A \rightarrow \{a\} \notin F_r$  đối với  $A$  mà  $A^+ \neq R$ ,  $a \notin A$ ,  $a \notin \cup K$

Có nghĩa là:

- $K$  là một khóa tối thiểu
- $a$  là thuộc tính thứ cấp
- $A$  không là khóa
- $A \rightarrow \{a\}$  không đúng trong  $r$ .

**Định nghĩa 4 (Dạng chuẩn Boyce-Codd - BCNF)**

Quan hệ  $r = \{h_1, h_2, \dots, h_m\}$  được gọi là dạng chuẩn Boyce-Codd nếu:  $A \rightarrow \{a\} \notin F_r$ , đối với những tập thuộc tính  $A$  mà  $A^+ \neq R$ ,  $a \notin A$ .

Nhận xét:

Qua định nghĩa ta có thể thấy dạng chuẩn BCNF là 3NF và 3NF là 2NF. Chúng ta có thể đưa ra các ví dụ chứng tỏ quan hệ là 2NF nhưng không là 3NF và có quan hệ là 3NF nhưng không là BCNF.

Nói cách khác là lớp các quan hệ BCNF là lớp con thực sự của lớp các quan hệ 3NF và lớp các quan hệ 3NF này lại là lớp con thực sự của lớp các quan hệ 2NF.

Đối với  $s = \langle R, F \rangle$  thì các dạng chuẩn 2NF, 3NF, BCNF trong đó ta thay  $F_r = F^+$ .

Chú ý: Đối với sơ đồ quan hệ chúng ta không có dạng chuẩn 1NF.

*Ví dụ 1:*

Cho  $s = \langle R, F \rangle$  là sơ đồ quan hệ, với  $R = (a_1, a_2, a_3, a_4)$

$F = \{ \{a_1\} \rightarrow \{a_2\}, \{a_3\} \rightarrow \{a_4\}, \{a_2\} \rightarrow \{a_1, a_3, a_4\} \}$

Dễ thấy  $a_1, a_2$  là các khóa tối thiểu của  $s$ ,  $a_3, a_4$  là thuộc tính thứ cấp. Do đó,  $s$  là 2NF, nhưng không là 3NF.

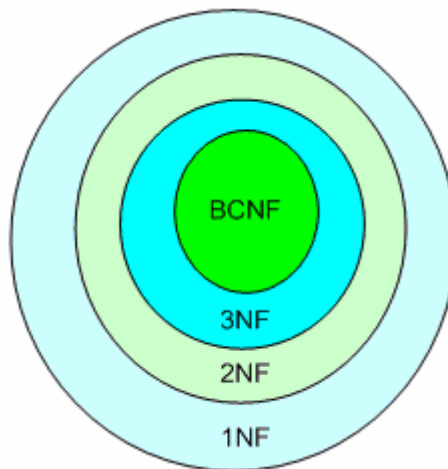
*Ví dụ 2:*

Cho  $t = \langle R, F \rangle$  là sơ đồ quan hệ, với  $R = (a_1, a_2, a_3)$

$F = \{ \{a_1, a_2\} \rightarrow \{a_3\}, \{a_3\} \rightarrow \{a_2\} \}$

Ta nhận thấy  $\{a_1, a_2\}$  là một khóa tối thiểu của  $t$ . Hiển nhiên  $t$  là 3NF. Vì có:  $\{a_3\} \rightarrow \{a_2\}$ ,  $a_3$  không thuộc khoá,  $a_2$  cơ bản, nên  $t$  không là BCNF.

Như vậy việc phân lớp các dạng chuẩn có thể được thể hiện qua hình 2.2:



Hình 2.2. Phân lớp các dạng chuẩn

#### 2.4.2. Một số đặc trưng dạng chuẩn 2NF

Bây giờ chúng ta nêu ra loại phụ thuộc hàm đặc biệt, mà phụ thuộc dữ liệu này đóng vai trò quan trọng trong dạng chuẩn 2.



**Định nghĩa 5 (phụ thuộc hoàn toàn, phụ thuộc bộ phận)**

Một phụ thuộc hàm  $A \rightarrow B$  được gọi là **sơ cấp** nếu không tồn tại một tập hợp  $A' \subset A$  sao cho  $A' \rightarrow B$ . Trong trường hợp này ta cũng nói B phụ thuộc **hoàn toàn** vào A. Như vậy nếu A là một *thuộc tính sơ cấp* thì phụ thuộc hàm  $A \rightarrow B$  cũng là sơ cấp. Trong trường hợp ngược lại, ta nói B phụ thuộc bộ phận vào A.

Trên cơ sở định lý này, người ta đưa ra hai mệnh đề quan trọng sau:

**Mệnh đề 1:**

Cho r là một quan hệ trên R. Khi đó r là 2NF khi và chỉ khi:

- r là 1NF
- Mỗi thuộc tính thứ cấp của r đều phụ thuộc hoàn toàn vào **mọi** khóa tối thiểu.

Vì sơ đồ quan hệ không có dạng chuẩn 1NF, từ định nghĩa 2 ta có mệnh đề sau:

**Mệnh đề 2:**

Cho s là một sơ đồ quan hệ trên R. Khi đó s là 2NF khi và chỉ khi mọi thuộc tính thứ cấp của s đều phụ thuộc hoàn toàn vào khóa tối thiểu bất kỳ.

Có thể thấy, bản chất dạng chuẩn 2NF là loại bỏ các phụ thuộc bộ phận giữa các thuộc tính thứ cấp với các khóa tối thiểu.

**2.4.3. Một số đặc trưng dạng chuẩn 3NF**

Trong mục này có khái niệm quan trọng mô tả dạng 3NF.

**Định nghĩa 6**

Một phụ thuộc hàm  $A \rightarrow C$  được gọi là **trực tiếp** nếu không có B ( $B \neq A$  và  $B \neq C$ ) sao cho  $A \rightarrow B$  và  $B \rightarrow C$  (tức là B không phụ thuộc hàm vào A hoặc C không phụ thuộc hàm vào B). Trong trường hợp nếu

có B như vậy thì B được gọi là tập thuộc tính bắc cầu và  $A \rightarrow C$  là phụ thuộc **bắc cầu**.

Trên cơ sở định nghĩa này, người ta đưa ra hai mệnh đề quan trọng sau:

**Mệnh đề 3**

Giả sử  $r = \{h_1, h_2, \dots, h_m\}$  là một quan hệ trên  $R = \{a_1, a_2, \dots, a_n\}$ . Khi ấy  $r$  ở dạng chuẩn 3NF nếu và chỉ nếu:

- Quan hệ  $r$  đã ở dạng chuẩn 2NF.
- Không có thuộc tính thứ cấp nào của  $r$  phụ thuộc bắc cầu vào một khoá tối thiểu.

**Mệnh đề 4**

Giả sử  $s$  là một sơ đồ quan hệ trên  $R$ . Khi đó  $s$  là 3NF nếu và chỉ nếu:

- $s$  là 2NF
- Mọi thuộc tính thứ cấp của  $s$  phụ thuộc trực tiếp vào **mỗi** khoá tối thiểu.

Trong dạng chuẩn 3NF chúng ta loại bỏ các phụ thuộc bộ phận, phụ thuộc bắc cầu giữa các thuộc tính thứ cấp với các khoá tối thiểu.

#### 2.4.4. Một số đặc trưng dạng chuẩn BCNF

**Định nghĩa 7 (tập sinh)**

Giả sử  $r$  là một quan hệ trên  $R$ , còn  $A, B \subseteq R$  và  $A \rightarrow B$ . Khi đó ta nói  $A$  là tập sinh của  $B$  nếu:

- $|A| < |B|$ ,
- Không tồn tại tập con thực sự của  $A$  mà xác định hàm cho  $B$

Tập  $C$  là tập sinh của quan hệ  $r$  nếu có một tập  $D$  nào đó để  $C$  là tập sinh của  $D$ .

Trên cơ sở định nghĩa này, người ta đưa ra các mệnh đề quan trọng sau:

**Mệnh đề 5**

Giả sử  $r$  là quan hệ trên  $R$ . Khi đó  $r$  là BCNF khi và chỉ khi mọi tập sinh của  $r$  đều là khóa.

**Mệnh đề 6**

Giả sử  $r$  là một quan hệ trên  $R$ . Khi đó  $r$  là BCNF nếu và chỉ nếu với mọi  $B \in M_r$ ,  $a \in B$  thì  $\{B - a\}_r^+ = B - a$ , ở đây  $M_r$  là hệ bằng nhau cực đại của quan hệ  $r$ .

Giả sử  $A \rightarrow B$  là một phụ thuộc hàm. Chúng ta gọi phụ thuộc hàm này là *tầm thường* nếu  $B \subseteq A$ . Ngược lại trong trường hợp này, chúng ta gọi nó là phụ thuộc hàm *không tầm thường*.

**Mệnh đề 7**

Giả sử  $s = \langle R, F \rangle$  là một sơ đồ quan hệ trên  $R$ . Khi đó  $s$  là BCNF nếu và chỉ nếu với mọi  $A \rightarrow B \in F$  và  $A \rightarrow B$  không tầm thường thì  $A^+ = R$ .

### 2.4.5. Các thuật toán

**Thuật toán 1 (kiểm tra một quan hệ  $r$  có là BCNF hay không)**

Vào:  $r = \{h_1, h_2, \dots, h_m\}$  là một quan hệ trên  $R = \{a_1, a_2, \dots, a_n\}$ .

Ra:  $r$  có phải là BCNF hay không?

*Phương pháp:*

Bước 1: Từ  $r$  ta xây dựng tập  $E_r$ :  $E_r = \{E_{ij}: 1 \leq i < j \leq m\}$ .

trong đó  $E_{ij} = \{a \in R: h_i(a) = h_j(a)\}$

Bước 2: Từ  $E_r$  xây dựng tập  $M$ .

$$M = \{B \in P(R): \text{Tồn tại } E_{ij} \in E_r, E_{ij} = B\}$$

Bước 3: Từ  $M$  xây dựng tập  $M_r$ ,  $M_r = \{B \in M: \forall B' \in M: B \not\subset B'\}$

Bước 4: Nếu với  $\forall B \in M_r, \forall a \in B$  ta có:  $\{B - a\}_r^+ = B - a$ , ta kết luận  $r$  đã ở BCNF, ngược lại  $r$  chưa ở BCNF.

Ví dụ:

Bảng 2.29: Quan hệ  $r$

A	B	C	D	E
1	0	1	0	1
2	1	0	1	1
2	0	1	1	0
1	1	3	1	3

Khi đó:  $E_{12} = \{E\}$ ,  $E_{13} = \{B, C\}$ ,  $E_{14} = \{A\}$

$E_{23} = \{A, D\}$ ,  $E_{24} = \{B, D\}$ ,  $E_{34} = \{D\}$

$M = \{\{E\}, \{A\}, \{D\}, \{B, C\}, \{A, D\}, \{B, D\}\}$

$M_r = \{\{B, C\}, \{A, D\}, \{E\}, \{B, D\}\}$

Ta có thể kiểm tra rằng  $\{B, C\} - B = C$  và  $\{C\}_r^+ = \{B, C\}$ . Vì vậy  $r$  không là BCNF.

**Thuật toán 2 (Kiểm tra một sơ đồ quan hệ  $s$  có là BCNF hay không)**

Vào:  $s = \langle R, F \rangle$ ,  $R = \{a_1, a_2, \dots, a_n\}$  và  $F = \begin{cases} A_1 \rightarrow B_1 \\ \dots & A_i, B_i \subseteq R \\ A_m \rightarrow B_m \end{cases}$

Ra:  $s$  có là BCNF hay không?

Bước 1: Nếu  $A_1 \rightarrow B_1$  là phụ thuộc hàm không tầm thường ( $B_i \subseteq A_i$ ) và  $A_1^+ \neq R$  thì dừng và kết luận  $s$  không là BCNF, ngược lại chuyển sang bước sau.

...

Bước  $m$ : Giống bước 1 nhưng đối với  $A_m \rightarrow B_m$

Bước  $m+1$ :  $s$  là BCNF.

Ví dụ:

Cho sơ đồ quan hệ  $s = \langle R, F \rangle$ , với  $R = \{a_1, a_2, a_3, a_4, a_5, a_6\}$

$$F = \begin{cases} \{a_1\} \rightarrow \{a_2\} \\ \{a_1, a_2\} \rightarrow \{a_3, a_4\} \\ \{a_4\} \rightarrow \{a_5, a_6\} \end{cases}$$

(nhận xét:  $m = 3$  phụ thuộc hàm không tầm thường)

Bước 1: Xét phụ thuộc hàm thứ nhất, có bao đóng vế trái:

$$\{a_1\}^+ = \{a_1, a_2, a_3, a_4, a_5, a_6\} = R.$$

Bước 2: Xét phụ thuộc hàm thứ 2, có bao đóng vế trái:

$$\{a_1, a_2\}^+ = \{a_1, a_2, a_3, a_4, a_5, a_6\} = R.$$

Bước 3: Xét tiếp phụ thuộc hàm thứ 3, có bao đóng vế trái:

$$\{a_4\}^+ = \{a_4, a_5, a_6\} \neq R.$$

Như vậy tại bước 3 ( $m = 3$ ) phải dừng và kết luận:  $s$  không là BCNF.

#### 2.4.6. Chuẩn hoá một quan hệ

Như ta đã biết, thực thể là hình ảnh tượng trưng cho một đối tượng cụ thể hay một khái niệm trừu tượng nhưng có mặt trong thế giới thực; còn kiểu thực thể là một nhóm các thực thể có chung nhau một số đặc tính mà chúng ta quan tâm. Ví dụ như dự án, con người, sản phẩm,...

Trong thực tế, để đơn giản hoá, người ta thường dùng khái niệm thực thể thay cho khái niệm kiểu thực thể. Vì vậy, từ đây trở đi, chúng ta sẽ dùng khái niệm “thực thể” thay vì khái niệm “kiểu thực thể”.

Do vậy, trong một hệ thông tin, cần lựa chọn một số tính chất đặc trưng để diễn tả một thực thể, các tính chất này được gọi là thuộc tính của thực thể được mô tả và đây cũng chính là các loại thông tin dữ liệu cần quản lý.

Giá trị các thuộc tính của một thực thể cho phép diễn tả một trường hợp cụ thể của thực thể, gọi là một **thể hiện** của thực thể đó.

*Ví dụ:*

(Lê Thanh Hà, 53 Hai Bà Trưng - Hà Nội, 1-5-1978) là một thể hiện của 'SINH VIÊN'. (Máy in, LASER SHOT, LBP\_1120) là một thể hiện của SẢN PHẨM.

Một *thuộc tính là sơ cấp* khi ta không cần phân tích nó thành nhiều thuộc tính khác, tùy theo nhu cầu xử lý trong hệ thông tin đối với một thực thể.

Thông thường một thực thể ứng với một bảng (hay một quan hệ của Codd).

Mỗi thực thể phải có ít nhất một thuộc tính mà mỗi giá trị của nó vừa đủ cho phép nhận diện một cách duy nhất một thể hiện của thực thể gọi là thuộc tính nhận dạng hay là khoá. Có nhiều trường hợp chúng ta phải dùng một tập các thuộc tính để nhận diện thực thể. Khi một thực thể có nhiều khoá, người ta chọn một trong số đó làm khoá chính (khóa tối thiểu). Giá trị của một khóa luôn luôn được xác định.

*Ví dụ:*

Số hoá đơn là thuộc tính nhận dạng của thực thể HOÁ ĐƠN.

Không thể có hai hay nhiều hoá đơn có cùng số hoá đơn trong cùng một hệ thông tin.

Trên cơ sở đơn giản hoá như vậy, chúng ta xét các dạng chuẩn đang hiện hành trên thực tế sau đây.

#### **2.4.6.1. Chuẩn hoá quan hệ ở dạng chuẩn 1**

Ta nói rằng *một thực thể hay quan hệ ở dạng chuẩn 1* nếu tất cả giá trị các thuộc tính của nó đều là sơ cấp. Điều kiện ràng buộc giống như 1NF ở trên. Định nghĩa dạng chuẩn 1 mang tính mô tả.

Bảng 2.30. Quan hệ BANHANG

Ngày tháng	Mã hàng	Tên hàng	Đơn giá	Số lượng	Tổng tiền theo ngày	Thanh toán
210397	M1	Radio	1000	1	10 000	6 000
210397	M3	Tv	4000	2	10 000	6 000
210397	M6	Xe đạp	1000	1	10 000	6 000
220397	M2	Máy giặt	7500	2	15 000	6 000
230397	M1	Radio	1000	3	15 000	11 000
230397	M4	Video	5000	2	15 000	11 000
230397	M9	Máy ảnh	2000	1	15 000	11 000

Chúng ta xác định được một khoá chính (khoá tối thiểu) cho thực thể BÁN HÀNG là tập {Ngày tháng, Mã hàng}.

#### 2.4.6.2. Chuẩn hoá quan hệ ở dạng chuẩn 2

Một thực thể hay quan hệ 1NF được xem là dạng chuẩn 2 nếu tất cả phụ thuộc hàm giữa khoá chính và các thuộc tính khác của nó đều là hoàn toàn.

Chú ý rằng định nghĩa dạng chuẩn 2 trong phần trước chặt hơn vì điều kiện phụ thuộc hoàn toàn liên quan đến mọi khoá tối thiểu, chứ không chỉ liên quan đến một khoá tối thiểu được chọn làm khoá chính.

Trong ví dụ trên thực thể BÁN HÀNG đã là 1NF ta nhận thấy đối với khoá chính {Ngày tháng, Mã hàng} các thuộc tính Tổng và Thanh toán phụ thuộc hàm vào thuộc tính ngày tháng; các thuộc tính tên hàng, đơn giá phụ thuộc hàm vào thuộc tính Mã hàng. Ngày tháng, Mã hàng là hai thuộc tính của khoá chính. Do đó dẫn đến trùng lặp dữ liệu. Thực thể BÁN HÀNG không là 2NF, ta phải tách nó ra làm 3 thực thể riêng biệt:

Bảng 2.31. Hàng hoá

Mã hàng	Tên hàng	Đơn giá
M1	Radio	1000
M2	Máy giặt	7500
M3	Tv	4000
M4	Video	5000
M6	Xe đạp	1000
M9	Máy ảnh	2000

Ta xác định được khoá chính của thực thể HÀNG HOÁ là Mã hàng.

Bảng 2.32. Khối lượng

Ngày tháng	Mã hàng	Số Lượng
210397	M1	1
210397	M3	2
210397	M6	1
220397	M2	2
230397	M1	3
230397	M4	2
230397	M9	1

Ta dễ dàng xác định được khoá chính (khoá tối thiểu) cho thực thể KHỐI LƯỢNG là {Ngày tháng, Mã hàng},

Bảng 2.33. Doanh số

Ngày tháng	Tổng	Thanh toán
210397	10 000	6000
220397	15 000	6 000
230397	15 000	11 000

Khoá chính là Ngày tháng.

Có thể thấy thực thể KHỐI LƯỢNG → thực thể HÀNG HOÁ



### 2.4.6.3. Chuẩn hoá quan hệ ở dạng chuẩn 3

Một thực thể (hay quan hệ) đã là 2NF được xem là dạng chuẩn 3NF nếu tất cả các phụ thuộc hàm giữa khoá chính và các thuộc tính khác của nó đều trực tiếp. Hay nói cách khác, mọi thuộc tính không nằm trong khoá chính đều không phụ thuộc hàm vào một thuộc tính không phải là khoá chính.

Ta có nhận xét: Một thực thể có nhiều khoá nhận dạng dễ có thể không thoả mãn dạng chuẩn 3NF. Mặt khác định nghĩa 3NF trong chương trước chặt hơn vì điều kiện phụ thuộc hoàn toàn và phụ thuộc trực tiếp liên quan đến mọi khoá tối thiểu, chứ không chỉ liên quan đến một khoá tối thiểu được chọn làm khoá chính.

Trong thực thể HÀNG HOÁ là 2NF ở trên, ta thấy trên đồ thị của các phụ thuộc hàm có hai con đường để đi từ 'Mã hàng' đến 'Đơn giá' hoặc đi qua thuộc tính 'Tên hàng'.



Hình 2.3. Ràng buộc phụ thuộc hàm giữa ba thuộc tính

Điều này chứng tỏ thực thể HÀNG HOÁ chưa đạt 3NF, dẫn đến trùng lặp đơn giá của tên hàng. Để đạt 3NF ta tách nó thành 2 thực thể riêng biệt:

Bảng 2.34. HÀNG

Mã hàng	Tên hàng
M1	Radio
M2	Máy giặt
M3	Tv
M4	Video
M6	Xe đạp
M9	Máy ảnh

Bảng 2.35. MATHANG

Tên hàng	Đơn giá
Radio	1000
Máy giặt	7500
Tv	4000
Video	5000
Xe đạp	1000
Máy ảnh	2000

Khoá chính là {Tên hàng}

Có thể thấy thực thể HÀNG  $\rightarrow$  thực thể MẶT HÀNG

#### 2.4.6.4. Chuẩn hoá quan hệ ở dạng chuẩn Boyce-Codd

Dạng chuẩn 3NF cho phép một thuộc tính thành phần của khoá chính phụ thuộc hàm vào một thuộc tính không phải là khoá.

Ví dụ:

Bảng 2.36. Quan hệ LOP - MON - THẦY

Lớp	Môn	Thầy
12	Toán	A
11	Toán	B
10	Toán	A
12	Địa	C
11	Địa	C
10	Sử	D

Thực thể này thoả mãn dạng 3NF. Khoá chính của nó bao gồm các thuộc tính 'Lớp' và 'Môn'.

Nhưng do quy tắc 'Mỗi thầy chỉ dạy một môn' ta thấy có sự phụ thuộc hàm của Môn (là một thành phần của khoá chính) vào Thầy (là một thuộc tính bình thường): 'Thầy'  $\rightarrow$  'Môn'

Ta nói rằng thực thể thoả mãn dạng chuẩn Boyce-Codd (BCNF) khi tất cả các phụ thuộc hàm của nó đều thuộc dạng  $K \rightarrow a$ , trong đó  $K$  là khoá chính còn  $a$  là một thuộc tính bất kỳ.

Để thoả dạng BCNF, ta có thể tách thực thể LOP - MON - THAY thành hai thực thể riêng biệt:

Bảng 2.37. Thực thể 'LỚP' và □THẦY□

Thầy	Lớp	Thầy	Môn
A	12	A	Toán
B	11	B	Toán
A	10	C	Địa
C	12	D	Sử
C	11		
D	10		

Thực thể 'LỚP' có khoá chính là {Thầy, lớp} và "THẦY" có khoá chính là {Thầy}

#### 2.4.7. Chuẩn hoá một sơ đồ quan hệ

##### 2.4.7.1. Phép phân rã một sơ đồ quan hệ

###### 2.4.7.1.1. Định nghĩa phép phân rã một sơ đồ quan hệ $s = \langle R, F \rangle$

Phép phân rã một sơ đồ quan hệ  $s = \langle R, F \rangle$  trên tập thuộc tính  $R = \{a_1, a_2, \dots, a_n\}$  là việc thay thế sơ đồ quan hệ đó bằng tập các sơ đồ quan hệ con  $s_1, s_2, \dots, s_k$  tương ứng trên  $R_1, R_2, \dots, R_k$ , trong đó  $R_i \subseteq R$ , ( $i = 1, \dots, k$ ) và  $R = \cup_{i=1, \dots, k} R_i$ , ở đây không đòi hỏi các  $R_i$  phải phân biệt nhau, còn  $F_i$  là phép chiếu của  $F$  trên  $R_i$ .

Mục đích của phép phân rã nhằm loại bỏ các file dữ liệu dư thừa (redundancy) và loại bỏ các dị thường: Không nhất quán, dị thường khi thêm dòng, dị thường khi xoá dòng của quan hệ, khi thực hiện phép cập nhật (sửa, thêm xoá).

Ví dụ: Xét quan hệ *HANG* (tên\_hãng, địa\_chỉ, mặt\_hàng, đơn\_giá) ta thấy:

a. Dư thừa dữ liệu: Khi có tên hãng thì lại có địa chỉ của hãng đó trong quan hệ (dư thừa)

a'. Không nhất quán (inconsistency), là hệ quả của (a): Giả sử trong quan hệ có nhiều tên hãng, địa\_chỉ thì địa chỉ của hãng thay đổi

nếu ta chỉ sửa địa chỉ trong một bản ghi (dòng) thì địa chỉ trong bản ghi khác không thay đổi dẫn đến 1 nhà cung cấp không có địa chỉ duy nhất. Dị thường này xuất hiện khi sửa dữ liệu.

*b. Dị thường khi thêm dòng (insertion anomalies):* Một hãng khi chưa cấp mặt hàng nào thì chưa thể thêm dòng ứng với hãng đó được, vì nếu ta thêm một dòng với hãng đó thì ta không thể đưa mặt hàng và đơn giá vào dòng đó được.

*b') Dị thường khi xoá bản ghi, là vấn đề ngược lại của (b):* Không thể xoá tất cả các mặt hàng được cung cấp bởi 1 hãng nào đó vì mặt hàng đó có thể được cung cấp bởi nhiều hãng.

Người ta có thể phân rã quan hệ trên thành một tập quan hệ. Để tránh các nhược điểm nói trên, nhưng đảm bảo không mất thông tin theo nghĩa dưới đây:

#### 2.4.7.1.2. Phép phân rã không mất thông tin

##### **Định nghĩa**

Giả sử sơ đồ quan hệ  $s = \langle R, F \rangle$  đã được phân rã thành các sơ đồ con  $s_1, s_2, \dots, s_k$  với  $s_i = \langle R_i, F_i \rangle$ , ( $i = 1, \dots, k$ ),  $R_i \subseteq R$ ,  $F_i$  là phép chiếu của  $F$  trên  $R_i$ . Khi đó ta nói rằng, phép phân rã trên là không mất thông tin đối với  $F$  nếu đối với mỗi quan hệ  $r$  trên  $R$  thoả mãn  $F$ :

$$r = \Pi R_1(r) \bowtie \Pi R_2(r) \bowtie \dots \bowtie \Pi R_k(r)$$

tức là  $r$  được tạo nên từ phép kết nối tự nhiên của các hình chiếu của nó trên các  $R_i$ , ( $i = 1, \dots, k$ ).

##### **Thuật toán kiểm tra phép phân rã không mất thông tin**

Vào:  $s = \langle R, F \rangle$ , phép phân rã  $p = (s_1, s_2, \dots, s_k)$ ,  $s_i = \langle R_i, F_i \rangle$ .

Ra: Kết luận phép phân rã  $p$  có mất thông tin hay không?

##### **Phương pháp:**

Bước 1: Thiết lập một bảng với  $n$  cột và  $k$  hàng, hàng  $i$  ứng với lược đồ quan hệ  $R_i$  cột  $j$  ứng với thuộc tính  $a_j$ . Tại ô  $(i, j)$  điền  $A_j$  nếu  $a_j \in R_i$ , ngược lại điền ký hiệu  $b_{ij}$ .

Bước 2: xét phụ thuộc hàm là  $F$  áp dụng cho bảng vừa thiết lập được. Giả sử  $X \rightarrow Y \in F$ , xét các dòng có giá trị bằng nhau trên thuộc tính  $X$  thì làm bằng giá trị của chúng trên thuộc tính  $Y$  theo cách sau: Nếu thấy một giá trị tương ứng với  $Y$  là  $A_j$  thì chuyển các giá trị  $b_{ij}$  thành  $A_j$ .

Bước 3: Tiếp tục áp dụng các phụ thuộc hàm cho bảng (kể cả việc lặp lại phụ thuộc hàm đã được áp dụng) cho đến khi không còn áp dụng được nữa.

Bước 4: Cuối cùng xem bảng kết quả: Nếu xuất hiện một hàng có đủ  $A_1, A_2, \dots, A_n$  thì phép phân rã trên là không mất thông tin, ngược lại là phép phân rã bị mất thông tin.

*Ví dụ:*

Cho một sơ đồ quan hệ  $s = \langle R, F \rangle$  với  $R = \{a, b, c, d, e, g\}$ ,

Phép tách  $p = \{s_1, s_2, \dots, s_5\}$  với:

$R_1 = \{a, b\}, \quad F_1: a \rightarrow b$

$R_2 = \{a, c, d\}, \quad F_2: cd \rightarrow a$

$R_3 = \{b, c, d\}, \quad F_3: bc \rightarrow d$

$R_4 = \{a, c, g\}, \quad F_4: ac \rightarrow g$

$R_5 = \{c, d, e\}, \quad F_5: ce \rightarrow d$

Kiểm tra xem phép phân rã trên có mất thông tin không?

- Lập bảng:

*Bảng 2.38. Bảng xuất phát*

	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>g</b>
S1	A1	A2	b13	b14	b15	b16
S2	A1	b22	A3	A4	b25	b26
S3	b31	A2	A3	A4	b35	b36
S4	A1	b42	b43	b44	A5	A6
S5	b51	b52	A3	A4	A5	b56

Xét phụ thuộc hàm  $a \rightarrow b$  có bảng sau (các hàng 1, 2, 4 bằng nhau trên cột  $a = A1$ ) nên ở cột  $b$  chúng được làm bằng và bằng A2.

Bảng 2.39.

	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>g</b>
S1	A1	A2	b13	b14	b15	b16
S2	A1	A2	A3	A4	b25	b26
S3	b31	A2	A3	A4	b35	b36
S4	A1	A2	b43	b44	A5	A6
S5	b51	b52	A3	A4	A5	b56

Xét phụ thuộc hàm  $cd \rightarrow a$  ta có bảng sau (hai hàng 2 và 5 bằng nhau trên cột  $cd$  (bằng A3, A4) nên trên cột  $a$  chúng được làm bằng và bằng A1).

Bảng 2.40.

	<b>A</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>g</b>
S1	A1	A2	b13	b14	b15	b16
S2	A1	A2	A3	A4	b25	b26
S3	A1	A2	A3	A4	b35	b36
S4	A1	A2	b43	b44	A5	A6
S5	A1	b52	A3	A4	A5	b56

Tương tự xét tiếp phụ thuộc hàm  $bc \rightarrow d$  bảng không thay đổi

Xét phụ thuộc hàm  $ac \rightarrow g$ , ta có bảng sau:

Bảng 2.41.

	<b>a</b>	<b>b</b>	<b>C</b>	<b>d</b>	<b>e</b>	<b>g</b>
S1	A1	A2	b13	b14	b15	b16
S2	A1	A2	A3	A4	b25	b26
S3	A1	A2	A3	A4	b35	b36
S4	A1	A2	b43	b44	A5	A6
S5	A1	b52	A3	A4	A5	A6

- Xét phụ thuộc hàm  $ce \rightarrow d$  bảng không thay đổi.

- Quay lại xét phụ thuộc hàm  $a \rightarrow b$  ta có bảng sau: (bảng 2.42):

*Bảng 2.42.*

	a	b	c	d	e	g
S1	A1	A2	b13	b14	b15	b16
S2	A1	A2	A3	A4	b25	b26
S3	A1	A2	A3	A4	b35	b36
S4	A1	A2	b43	b44	A5	A6
S5	A1	A2	A3	A4	A5	A6

- Tiếp tục xét phụ thuộc hàm  $cd \rightarrow a$ ,  $bc \rightarrow d$ ,  $ae \rightarrow g$ ,  $ce \rightarrow d$ , còn lại ta thấy bảng không thay đổi. Vậy bảng kết quả là bảng 2.42. Ta nhận thấy trong bảng này hàng thứ 5 có đủ A1, A2, ..., A6 nên ta kết luận phép phân rã p trong ví dụ là không mất thông tin.

#### **2.4.7.2. Sơ đồ quan hệ dạng chính tắc**

##### **2.4.7.2.1. Các định nghĩa**

*Phụ thuộc hàm đầy đủ/phụ thuộc hoàn toàn*

Giả sử  $s = \langle R, F \rangle$  là một sơ đồ quan hệ trên tập thuộc tính R và  $A, B \in P(R)$  - tập các tập con của R, khi đó B được gọi là phụ thuộc hàm đầy đủ vào A nếu B phụ thuộc hàm A nhưng không phụ thuộc hàm vào bất kỳ tập con nào của A.

*Thuộc tính dư thừa (vế trái của phụ thuộc hàm)*

Một thuộc tính của vế trái một phụ thuộc hàm được gọi là dư thừa nếu loại bỏ nó khỏi vế trái không làm thay đổi bao đóng của tập phụ thuộc hàm, tức là: Nếu  $X \rightarrow Y \in F$ ,  $a \in X$ , a là dư thừa nếu  $\{X - a\} \rightarrow Y \in F^+$ .

*Tập phụ thuộc hàm không dư thừa*

Phụ thuộc hàm  $A \rightarrow B$  được gọi là dư thừa trong tập phụ thuộc hàm F của sơ đồ quan hệ  $s = \langle R, F \rangle$  nếu:  $F^+ = \{F - \{A \rightarrow B\}\}^+$

Tập phụ thuộc hàm F không chứa một phụ thuộc hàm dư thừa nào trong nó được gọi là tập phụ thuộc hàm không dư thừa.

*Phủ của phụ thuộc hàm*

Gọi F và G là các tập của phụ thuộc hàm. Nói rằng F và G là tương đương nếu  $F^+ = G^+$ . Nếu F và G là tương đương, đôi khi cũng nói F phủ G (hoặc G phủ F).

*Phủ không dư thừa*

G là phủ không dư thừa của F, nếu  $G^+ = F^+$  và không chứa một phụ thuộc hàm dư thừa nào.

*Định nghĩa phụ thuộc hàm tối thiểu*

Tập phụ thuộc hàm F được gọi là **tối thiểu** nếu:

- Mỗi vế phải của phụ thuộc hàm thuộc F chỉ có một thuộc tính.
- Không tồn tại một phụ thuộc hàm  $X \rightarrow A$  thuộc F mà:

$$F^+ = \{F - \{X \rightarrow A\}\}^+$$

- Không tồn tại một phụ thuộc hàm  $X \rightarrow A$  thuộc F và tập con thực sự (Z) của X mà  $F^+ = (F - \{X \rightarrow A\} \cup \{Z \rightarrow A\})^+$

Thực vậy, điều kiện b đảm bảo cho tập F không có một phụ thuộc hàm nào là dư thừa và điều kiện c bảo đảm không có một thuộc tính nào tham gia phía trái của phụ thuộc hàm là dư thừa. Vế phải của phụ thuộc hàm ở điều kiện a chỉ có một thuộc tính bảo đảm chắc chắn không có một thuộc tính nào bên vế phải là dư thừa.

*Định nghĩa chính tắc (Sơ đồ quan hệ dạng chính tắc)*

Chúng ta nói một sơ đồ quan hệ  $s = \langle R, F \rangle$  là chính tắc nếu F là tối thiểu, nghĩa là F thỏa mãn đồng thời 3 tiêu chuẩn sau đây:

- Mỗi vế phải của phụ thuộc hàm thuộc F chỉ có một thuộc tính.
- Không có  $X \rightarrow a$  nào ở trong tập F để  $F - \{X \rightarrow a\}$  tương đương với F theo nghĩa  $\{F - \{X \rightarrow a\}\}^+ = F^+$  (không có phụ thuộc hàm nào dư thừa).



c. Không có  $X \rightarrow a$  và một tập con thực sự (Z) của X để:

$F - \{X \rightarrow a\} \cup \{Z \rightarrow a\}$  tương đương với F (tập X được gọi là tối ưu).

#### 2.4.7.2.2. Các thuật toán

##### **Thuật toán loại bỏ thuộc tính dư thừa ở vế trái trên mỗi phụ thuộc hàm của F**

Cho sơ đồ quan hệ  $s = \langle R, F \rangle$ , thuật toán sau đây sẽ loại bỏ thuộc tính dư thừa ở vế trái trên mỗi phụ thuộc hàm của F (đảm bảo các phụ thuộc hàm là đầy đủ). Người ta thường gọi đó là thuật toán tối ưu trên tập F của sơ đồ quan hệ

Thuật toán tối ưu trên tập F của sơ đồ quan hệ  $s = \langle R, F \rangle$

trong đó  $F = \{A^*_1 \rightarrow B_1, \dots, A^*_m \rightarrow B_m\}$ .

Vào:  $A \rightarrow B \in F^+$ , A là một trong những  $A^*_i$  ( $i=1, \dots, m$ )

Ra:  $A' \rightarrow B \in F^+$

trong đó:  $A' \subseteq A$  và không tồn tại  $A'' \subset A'$  và  $A'' \rightarrow B \in F^+$ .

Bước 0:

$$A_0 = A = \{a'_1, a'_2, \dots, a'_t\}$$

Bước 1:

$$A_1 = \begin{cases} A_0 - \{a'_1\} & \text{Nếu } A_0 - \{a'_1\} \rightarrow B \in F^+ \\ A_0 & \text{Nếu ngược lại} \end{cases}$$

Bước i:

$$A_i = \begin{cases} A_{i-1} - \{a'_i\} & \text{Nếu } A_{i-1} - \{a'_i\} \rightarrow B \in F^+ \quad (i = 2, \dots, t) \\ A_{i-1} & \text{Nếu ngược lại} \end{cases}$$

**Nhận xét:** Giống như thuật toán tìm khoá tối thiểu, ở đây không xác định hàm với R mà xác định hàm với B. Dùng  $A \rightarrow B \in F^+ \Leftrightarrow B \in A^+$

##### **Thuật toán tìm tập phụ thuộc hàm không dư thừa**

Vào: Sơ đồ quan hệ  $s = \langle R, F \rangle$

$$F = \{A_1 \rightarrow B_1, \dots, A_m \rightarrow B_m\}$$

$B_j$  chỉ gồm một thuộc tính ( $j=1, \dots, m$ ).

Ra: Sơ đồ quan hệ  $s^* = \langle R, F^* \rangle$  trong đó  $F^*$  không dư thừa.

Bước 1: Đặt  $F - \{A_1 \rightarrow B_1\} = G_1$

Kiểm tra:  $F^+ = G_1^+$

$$\text{Quyết định } F_1 = \begin{cases} G_1 & \text{Nếu } F^+ = G^+ \\ F & \text{Nếu } F^+ \neq G^+ \end{cases}$$

...

Bước i: Kiểm tra ( $i = 2, \dots, m$ )

$$\{F_{i-1} - \{A_i \rightarrow B_i\}\}^+ = F^+ \quad (1)$$

$$F_i = \begin{cases} F_{i-1} - \{A_i \rightarrow B_i\} & \text{Nếu (1) thỏa mãn} \\ F_{i-1} & \text{Nếu (1) không thỏa mãn} \end{cases}$$

Đặt  $F^* = F_m$

*Nhận xét:*

- Hầu hết các sơ đồ quan hệ cho trước phần lớn là dư thừa vì nó lưu trữ các phụ thuộc hàm dẫn xuất từ các phụ thuộc hàm khác.

- Các phụ thuộc hàm được xếp theo các thứ tự khác nhau sẽ có thể cho kết quả khác nhau.

### Bổ đề 1

Mỗi tập phụ thuộc hàm  $F$  đều được phủ bằng các phụ thuộc hàm  $G$  có vế phải bao gồm không quá một thuộc tính.

*Chứng minh:*

Gọi  $G$  là tập các phụ thuộc hàm  $X \rightarrow a$  sao cho với  $X \rightarrow Y$  thuộc  $F$  thì  $a \in Y$ . Từ  $X \rightarrow Y$  suy ra  $X \rightarrow a$  (theo luật tách).

Do vậy  $G \subseteq F^+$ .

Ngược lại, có  $F \subseteq G^+$  vì nếu  $Y = a_1, \dots, a_n$  thì  $X \rightarrow Y$  được suy ra từ  $X \rightarrow a_1, \dots, X \rightarrow a_n$  nhờ luật hợp.

**Định lý 1**

Mỗi tập phụ thuộc hàm  $F$  đều tương đương với tập  $F'$  tối thiểu.

*Chứng minh:*

Theo bổ đề trên, giả sử rằng không vế phải nào của các phụ thuộc hàm của  $F$  có nhiều hơn một thuộc tính. Để kiểm tra điều kiện  $b$  trong định nghĩa chính tắc, xét  $(X \rightarrow Y) \in F$ .

Nếu  $(F - \{X \rightarrow Y\})^+ = F^+$  loại bỏ  $X \rightarrow Y$  khỏi  $F$ .

Chú ý rằng các phụ thuộc hàm  $F$  được sắp xếp theo một thứ tự khác nhau thì sẽ cho kết quả khác nhau.

*Ví dụ* (loại bỏ các hàm dư thừa)

Cho tập  $F$  gồm:

$$\begin{array}{lll} A \rightarrow B & A \rightarrow C & B \rightarrow C \\ B \rightarrow A & C \rightarrow A & \end{array}$$

Có thể loại bỏ khỏi  $F$ :  $B \rightarrow A$  và  $A \rightarrow C$  hoặc loại bỏ  $B \rightarrow C$  nhưng không thể đồng thời loại bỏ cả ba phụ thuộc hàm.

Như vậy điều kiện  $b$  được thoả mãn. Cần kiểm tra điều kiện  $c$  cho tập phụ thuộc còn lại của  $F$ . Ở đây cũng cần lưu ý tới thứ tự các thuộc tính xuất hiện bên vế trái của các phụ thuộc hàm.

Loại bỏ các thuộc tính vế trái của các phụ thuộc hàm sao cho tập các thuộc tính vẫn là tương đương. Quá trình tiếp tục cho tới khi không thể loại bỏ được thuộc tính nào nữa. Như vậy tập các phụ thuộc còn lại của  $F$  sẽ tạo nên  $F'$  và thoả mãn ba điều kiện của một tập tối thiểu.

*Ví dụ:*

Cho tập  $F$

$$\begin{array}{lll} AB \rightarrow C & D \rightarrow E & CG \rightarrow D \\ C \rightarrow A & D \rightarrow G & CE \rightarrow A \\ BC \rightarrow D & BE \rightarrow C & CE \rightarrow G \end{array}$$

$$ACD \rightarrow B \quad CG \rightarrow B$$

Áp dụng thuật toán tính bao đóng của tập các thuộc tính đối với tập F theo thứ tự từ phải qua trái và từ dưới lên trên. Rõ ràng  $CE \rightarrow A$  là dư thừa vì có thể suy ra từ  $C \rightarrow A$ .  $CG \rightarrow B$  là dư thừa vì  $CG \rightarrow D$ ,  $C \rightarrow A$  và  $ACD \rightarrow B$  suy ra  $CG \rightarrow B$ .

Từ đó dễ dàng tính được  $(CG)^+$  và không còn phụ thuộc hàm nào là dư thừa.  $ACD \rightarrow B$  có thể thay thế bởi  $CD \rightarrow B$  vì  $C \rightarrow A$ .

Từ đó có tập tối thiểu là:

$$AB \rightarrow C$$

$$C \rightarrow A$$

$$BC \rightarrow D$$

$$CD \rightarrow B$$

$$D \rightarrow E$$

$$D \rightarrow G$$

$$BE \rightarrow C$$

$$CG \rightarrow D$$

$$CE \rightarrow G$$

Nếu loại bỏ  $CE \rightarrow A$ ,  $CG \rightarrow D$  và  $ACD \rightarrow B$  sẽ có một tập tối thiểu khác:

$$AB \rightarrow C$$

$$D \rightarrow G$$

$$C \rightarrow A$$

$$BE \rightarrow C$$

$$BC \rightarrow D$$

$$CG \rightarrow B$$

$$D \rightarrow E$$

$$CE \rightarrow G$$

### **Thuật toán xác định sơ đồ dạng chính tắc**

Cho sơ đồ quan hệ  $s = \langle R, F \rangle$

Vào:  $F = \langle A_1 \rightarrow B_1, \dots, A_m \rightarrow B_m \rangle$

Ra:  $s^* = \langle R, F^* \rangle$  chính tắc.

Bước 1: Nhờ bổ đề trên, có  $s = \langle R, F' \rangle$  sao cho:

$A \rightarrow B \in F$  có dạng  $\{A \rightarrow \{b\}\}$

Bước 2: Nhờ kết quả trên, ta tối ưu từng phụ thuộc hàm của  $F'$

Giả sử  $F'' = \{C_1 \rightarrow \{b_1\}, \dots, C_t \rightarrow \{b_t\}\}$

$C_i$  tối ưu  $\in R$  và  $b_i \in R$ ,  $i = (1, \dots, t)$

Bước 3: Cũng nhờ kết quả trên ta loại bỏ các hàm dư thừa, ta thu được:

$F_i = \{D_1 \rightarrow \{c_1\}, \dots, D_t \rightarrow \{c_t\}\}$

$D_i \subset R$  và  $c_i \in R$ ,  $i = (1, \dots, p)$ , trong đó  $p \leq t$ .

### 2.4.7.3. Phép tách một sơ đồ quan hệ thành hệ lược đồ đạt 3NF

#### *Thuật toán tách một sơ đồ quan hệ thành hệ lược đồ đạt 3NF*

Vào: Sơ đồ quan hệ  $R$ , tập các phụ thuộc hàm  $F$ , không làm mất tính tổng quát, được giả sử là phụ thuộc hàm tối thiểu.

Ra: Phép tách không mất thông tin trên  $R$  bảo toàn các phụ thuộc hàm sao cho mỗi sơ đồ con đều ở 3NF.

*Phương pháp:*

i - Loại bỏ tất cả các thuộc tính của  $R$  nếu các thuộc tính đó không liên quan đến phụ thuộc hàm nào của  $F$ , hoặc về trái, hoặc về phải.

ii - Nếu có một phụ thuộc hàm nào của  $F$  mà liên quan tới tất cả các thuộc tính của  $R$  thì kết quả ra chính là  $R$ .

iii - Ngoài ra, phép tách  $p$  đưa ra các sơ đồ gồm các thuộc tính  $XA$  cho phụ thuộc hàm  $X \rightarrow A$  thuộc  $F$ , nếu  $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n$  thuộc  $F$  thì thay thế tập thuộc tính  $XA_1A_2\dots A_n$  cho  $XA_i$  ( $1 \leq i \leq n$ ). Quá trình tiếp tục.

*Chú ý:* Tại mỗi bước kiểm tra sơ đồ R, nếu mỗi thuộc tính ngoài khoá không phụ thuộc bắc cầu vào khoá chính, khi đó R sẽ ở 3NF, ngược lại cần áp dụng bước (iii) để tách tiếp.

*Ví dụ:*

Cho lược đồ quan hệ R(CTHRSG) với tập phụ thuộc hàm tối thiểu:  $C \rightarrow T$ ,  $HR \rightarrow C$ ,  $HT \rightarrow R$ ,  $CS \rightarrow G$  và  $HS \rightarrow R$ .

Dùng thuật toán trên cho tập lược đồ ở dạng chuẩn thứ 3:

R1(CT), R2(CHR), R3(HRT), R4(CGS) và R5(HRS)

*Định lý:*

Thuật toán trên thoả mãn là một phép tách không mất thông tin, bảo toàn các phụ thuộc hàm và các lược đồ đích ở dạng chuẩn 3.

#### **2.4.7.4. Phép tách một sơ đồ quan hệ thành hệ lược đồ đạt BCNF**

**Bổ đề 2:**

a. Giả sử R là một lược đồ quan hệ với tập phụ thuộc hàm F.

Gọi  $p = (R_1, R_2, \dots, R_k)$  là một phép tách không mất thông tin của R đối với F. Gọi  $F_i$  là hình chiếu của F trên  $R_i$  ( $i = 1, \dots, k$ ), tức là tập của  $(X \rightarrow Y) \in F^+$  sao cho:  $X \subseteq R_i$  và  $Y \subseteq R_i$

Gọi  $@ = (s_1, \dots, s_m)$  là phép tách không mất thông tin của  $R_i$  đối với  $F_i$ . Khi đó phép tách của R thành  $(R_1, \dots, R_{i-1}, S_1, \dots, S_m, R_{i+1}, \dots, R_k)$  đối với F là không mất thông tin.

b. Giả sử R, F và p như ở mục (a); gọi  $t = (R_1, \dots, R_k, R_{k+1}, \dots, R_n)$  là một phép tách của R thành tập các lược đồ bao gồm cả p thì t cũng là phép tách không mất thông tin đối với F.

#### **Thuật toán tách một lược đồ quan hệ thành hệ lược đồ đạt BCNF**

*Vào:* Lược đồ quan hệ R và tập phụ thuộc hàm F

*Ra:* Phép tách của R không mất thông tin sao cho mỗi lược đồ quan hệ trong phép tách đều ở BCNF đối với phép chiếu của F trên lược đồ đó.

*Phương pháp:*

Cấu trúc phép tách  $p$  trên  $R$  theo phương pháp lặp liên tiếp. Tại mỗi bước phép tách  $p$  là bảo đảm không mất thông tin đối với  $F$ .

Bước đầu:  $p$  chỉ bao gồm  $R$

Các bước tiếp: nếu  $S$  là một sơ đồ thuộc  $p$ ,  $S$  chưa ở BCNF: chọn  $X \rightarrow A$  là phụ thuộc hàm thoả mãn trên  $S$ , trong đó  $X$  không chứa khoá của  $S$ ,  $A \in X$ . Rõ ràng cần phải có một số thuộc tính khác của  $S$  vừa không phải  $A$ , vừa không thuộc  $X$  hoặc nếu không thì  $X$  phải chứa một khoá của  $S$ . Thay thế  $S$  trong  $p$  bởi  $S_1$  và  $S_2$  với  $S_1 = XA$ ,  $S_2 = S - A$

Theo định lý trước phép tách  $s$  thành  $S_1$  và  $S_2$  là không mất thông tin đối với tập phụ thuộc hàm trên  $S$  vì rằng:  $S_1 \cap S_2 = X$ ,  $X \rightarrow S_1 - S_2 = A$ .

Theo bổ đề trên phần (a),  $p$  được thay  $S$  bằng  $S_1$  và  $S_2$  là không mất thông tin. Mỗi sơ đồ  $S_1$ ,  $S_2$  đều có số thuộc tính ít hơn  $S$ .

Quá trình tiếp tục cho tới khi tất cả các lược đồ đều ở BCNF. Chú ý tại mọi thời điểm  $p$  luôn bảo đảm không mất thông tin, vì rằng  $p$  ban đầu là  $R$ , mà bước thay đổi  $p$  đều bảo toàn tính chất đó.

*Ví dụ:*

Cho lược đồ  $R(CTHRSG)$ , trong đó C: Giáo trình, T: thầy giáo, H: giờ, R: phòng học, S: sinh viên, G: lớp.

Tập các phụ thuộc hàm  $F$ :

$C \rightarrow T$ : Mỗi giáo trình có một thầy dạy

$HR \rightarrow C$ : Chỉ một môn học (Giáo trình) ở một phòng học tại một thời điểm

$HT \rightarrow R$ : Tại mỗi thời điểm mỗi thầy giáo chỉ có thể dạy ở một phòng học.

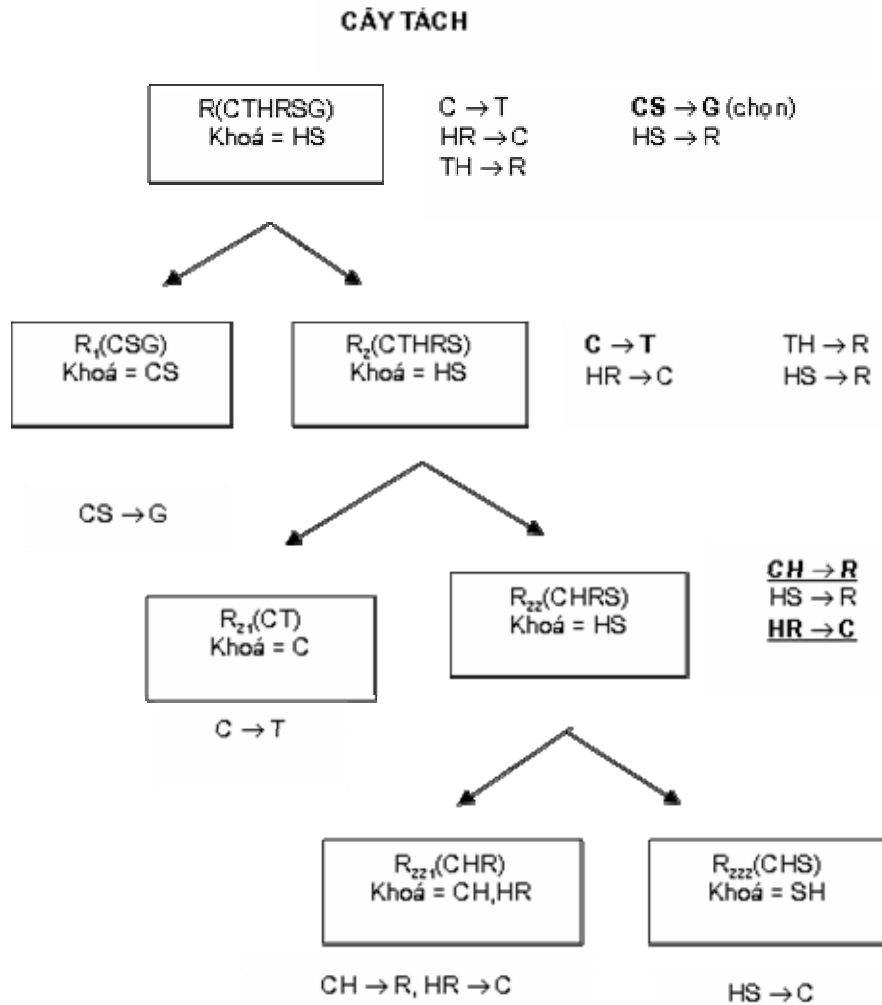
$CS \rightarrow G$ : Mỗi sinh viên chỉ ở một lớp theo học mỗi giáo trình

$HS \rightarrow R$ : Mỗi sinh viên chỉ có thể ở một phòng học tại mỗi thời điểm.

Khoá của  $R$  là **HS**.

Tách lược đồ  $R$  thành các lược đồ đạt BCNF:

Xét  $CS \rightarrow G$  cho  $R$ . Vi phạm điều kiện BCNF vì  $CS$  không chứa khoá. Do vậy, tách thành  $R_1(CSG)$  và  $R_2(CTHRS)$ . Bước tiếp cần tính  $F^+$  và chiếu xuống  $R_1$  và  $R_2$ , sau đó kiểm tra các lược đồ đã ở BCNF chưa. Có thể biểu diễn quá trình tách qua sơ đồ sau:



Hình 2.4. Sơ đồ tách

Phép tách cuối cùng được  $R = (R_1, R_{21}, R_{221}, R_{222})$ .



**2.4.7.5. Chuẩn hoá bằng phép tổng hợp**

Trong phần chuẩn hoá trên đã xem xét quá trình chuẩn hoá một lược đồ quan hệ thành 3NF nhờ phép tách không mất mát thông tin đối với tập các phụ thuộc hàm. Phần này sẽ trình bày quá trình chuẩn hoá nhờ phép tổng hợp. Điều khác cơ bản của phép tổng hợp so với phép tách là thông tin ban đầu gồm tập các thuộc tính (hiểu theo nghĩa người sử dụng chỉ biết tên các thuộc tính) và các tập phụ thuộc hàm, còn trong phương pháp tách, thông tin ban đầu là một lược đồ cụ thể. Qua phép tổng hợp (hoặc phép tách) kết quả đều cho một tập các lược đồ đều ở 3NF.

Gọi  $F$  và  $G$  là các tập của các phụ thuộc hàm. Nói rằng  $F$  và  $G$  là tương đương nếu  $F^+ = G^+$ . Nếu  $F$  và  $G$  tương đương đôi khi còn nói  $F$  phủ  $G$  hoặc ngược lại  $G$  phủ  $F$ .

**Thuật toán kiểm tra  $F$  và  $G$  có hay không tương đương với nhau**

Phương pháp kiểm tra:

Lấy mỗi phụ thuộc hàm  $Y \rightarrow Z$  thuộc  $F$ , kiểm tra xem liệu  $Y \rightarrow Z$  có thuộc  $G^+$  không? dùng thuật toán tính bao đóng của tập các thuộc tính đối với tập  $G$  để tính  $Y^+$  và kiểm tra liệu  $Z \subseteq Y^+$  không? Nếu tồn tại 1 phụ thuộc hàm  $Y \rightarrow Z$  thuộc  $F$  mà không thuộc  $G^+$  thì  $F^+ \neq G^+$ .

Nếu mỗi phụ thuộc hàm thuộc  $F$  cũng thuộc  $G^+$  thì mỗi phụ thuộc hàm  $V \rightarrow W$  thuộc  $F^+$  cũng thuộc  $G^+$ .

Để kiểm tra mỗi phụ thuộc  $G$  là thuộc  $F^+$ , quá trình làm hoàn toàn tương tự. Do đó  $F$  và  $G$  là tương đương khi và chỉ khi mỗi phụ thuộc hàm thuộc  $F$  là thuộc  $G^+$  và mỗi phụ thuộc hàm thuộc  $G$  là thuộc  $F^+$ .

**Thuật toán tổng hợp thành các lược đồ 3NF**

Vào: Tập các thuộc tính  $R = \{a_1, a_2, \dots, a_n\}$ , tập phụ thuộc hàm  $F$ .

Ra: Tập các lược đồ quan hệ ở 3NF.

*Phương pháp:*

- Tìm phủ không dư thừa H của F (tức là tìm phủ của F và loại bỏ các hàm dư thừa).
- Loại bỏ các thuộc tính dư thừa trong F
- Phân chia phụ thuộc H thành các nhóm sao cho các phụ thuộc trong một nhóm là có cùng vế trái.
- Mỗi cặp nhóm, ví dụ  $H_1$  và  $H_2$  có vế trái là X và Y tồn tại song ánh  $(X \rightarrow Y) \in H^+$  (tức là  $X \rightarrow Y$  và  $Y \rightarrow X$ ) thì hoà hai nhóm đó lại với nhau với  $A \in Y$ . Nếu  $X \rightarrow A \in H$  thì loại bỏ nó khỏi H. Tương tự cho mỗi  $(Y \rightarrow B) \in H$  với  $B \in X$ , ở mỗi nhóm đạt được lược đồ quan hệ bao gồm tất cả các thuộc tính xuất hiện trong nhóm đó. Mỗi thuộc tính xuất hiện trong vế trái của một phụ thuộc hàm trong nhóm là một khoá của lược đồ quan hệ. Các khoá tìm được gọi là khoá tổng hợp. Mỗi tập được cấu trúc như lược đồ quan hệ.

*Ví dụ:*

Cho tập thuộc tính  $\{A, B_1, B_2, C_1, C_2, D, E, I_1, I_2, I_3, J\}$ , tập hợp phụ thuộc hàm F:

$$A \rightarrow B_1 B_2 C_1 C_2 D E I_1 I_2 I_3 J$$

$$B_1 B_2 C_1 \rightarrow A C_2 D E I_1 I_2 I_3 J$$

$$B_1 B_2 C_2 \rightarrow A C_1 D E I_1 I_2 I_3 J$$

$$E \rightarrow I_1 I_2 I_3$$

$$C_1 D \rightarrow J$$

$$C_2 D \rightarrow J$$

$$I_1 I_2 \rightarrow I_3$$

$$I_2 I_3 \rightarrow I_1$$

$$I_1 I_3 \rightarrow I_2$$

Rõ ràng F là chưa tối thiểu vì dư thừa. Dùng các thuật toán đã biết để đảm bảo các phụ thuộc hàm trong F là không dư thừa và loại bỏ các thuộc tính ở vế trái phụ thuộc hàm trong H để không dư thừa:

$$\begin{aligned}
 H = \{ & A \rightarrow B_1 B_2 C_1 C_2 D E \\
 & B_1 B_2 C_1 \rightarrow A \\
 & B_1 B_2 C_2 \rightarrow A \text{ vì } A \text{ có thể suy ra } C_1 C_2 DE \text{ I}_1 I_2 I_3 J \\
 & E \rightarrow I_1 I_2 \\
 & C_1 D \rightarrow J \\
 & C_2 D \rightarrow J \\
 & I_1 I_2 \rightarrow I_3 \\
 & I_2 I_3 \rightarrow I_1 \\
 & I_1 I_3 \rightarrow I_2 \}
 \end{aligned}$$

Ở đây có 9 nhóm phụ thuộc hàm. Hoà các nhóm được:

$$\begin{aligned}
 H = \{ & (A, B_1 B_2 C_1, B_1 B_2 C_2) \rightarrow D E \text{ (Hoà 3 nhóm đầu)} \\
 & (E) \rightarrow I_1 I_2 \\
 & (C_1 D) \rightarrow J \\
 & (C_2 D) \rightarrow J \\
 & (I_1 I_2, I_2 I_3, I_1 I_3) \rightarrow I_1, I_2, I_3 \} \text{ (Hoà 3 nhóm cuối)}
 \end{aligned}$$

Thiết lập các lược đồ:

$$\begin{aligned}
 R_1 = & AB_1 B_2 C_1 C_2 DE \text{ với khoá} & K_1 = \{ & A, B_1 B_2 C_1, B_1 B_2 C_2 \} \text{ (3 khoá)} \\
 R_2 = & EI_1 I_2 & K_2 = \{ & E \} \\
 R_3 = & C_1 DJ & K_3 = \{ & C_1 D \} \\
 R_4 = & C_2 DJ & K_4 = \{ & C_2 D \} \\
 R_5 = & I_1 I_2 I_3 & K_5 = \{ & I_1 I_2, I_2 I_3, I_1 I_3 \}
 \end{aligned}$$

Như vậy được tập các lược đồ:  $R = \{R_1 R_2 R_3 R_4 R_5\}$  đều ở 3NF

*Định lý 2:*

Lược đồ quan hệ  $R(A_1, \dots, A_n)$  được tổng hợp qua thuật toán trên từ tập hợp phụ thuộc hàm  $F$  đạt 3NF. Nói cách khác, không một thuộc tính ngoài khoá nào của  $R$  là phụ thuộc hàm bậc cầu vào một khoá chính của  $R$ .

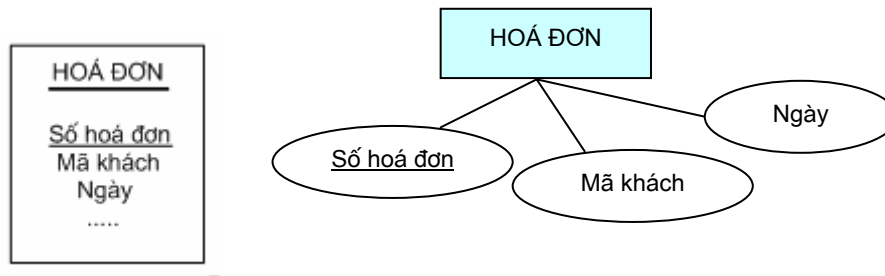
## 2.5. MỐI QUAN HỆ (RELATIONSHIP)

Khi xây dựng mô hình dữ liệu, các thực thể thường được biểu diễn bằng những hình chữ nhật:



Hình 2.5. Biểu diễn thực thể SANPHAM

Chúng ta có thể biểu diễn một thực thể bằng một trong hai cách “truyền thống” hay “đồ hoạ”:



Hình 2.6. Biểu diễn thực thể HOADON

### 2.5.1. Đặc tả mối quan hệ

Khái niệm quan hệ ở mục này (khác với khái niệm quan hệ của Codd) được dùng để nhóm hợp hai hay nhiều thực thể với nhau nhằm biểu hiện một mối liên quan tồn tại trong thế giới thực giữa các thực thể này. (Nếu sợ nhầm lẫn, chúng ta có thể dùng thuật ngữ “mối quan hệ” để phân biệt với thuật ngữ “quan hệ” của Codd).

Kích thước của một quan hệ là số thực thể cấu thành nên quan hệ.

Trong một mô hình dữ liệu các quan hệ được biểu diễn bằng hình thoi hoặc hình tròn hoặc elíp.

Trong mỗi thực thể bao giờ chúng ta cũng xác định được ít nhất một thuộc tính, còn mỗi quan hệ có thể không có thuộc tính. Trong một số trường hợp, mỗi quan hệ cũng có thể có những thuộc tính riêng (nếu có thì các thuộc tính riêng đó là rất quan trọng, không thể bỏ sót).

Ví dụ:

Hoá đơn dùng để thanh toán một số loại sản phẩm bán ra. Mỗi dòng hoá đơn cho biết số lượng và giá của từng loại sản phẩm. Đây là một quan hệ có kích thước là 2 (có 2 thực thể tham gia vào mỗi quan hệ), còn gọi là quan hệ nhị nguyên.



Hình 2.7. Đặc tả mối quan hệ DONGHOADON

### 2.5.2. Các đặc trưng của mỗi quan hệ

Xét  $R$  là một tập các quan hệ và  $E$  là một thực thể cấu thành của  $R$ , mỗi cặp  $(E, R)$  được biểu thị trên sơ đồ khái niệm dữ liệu bằng một đoạn thẳng. Với thực thể  $E$ , ta có thể xác định được:

- $X$  là số tối thiểu các thể hiện tương ứng với  $E$  mà  $R$  có thể có trong thực tế. Giá trị  $X$  như vậy chỉ có thể bằng 0 hay 1.
- $Y$  là số tối đa các thể hiện tương ứng với  $E$  mà  $R$  có thể có trong thực tế. Giá trị của  $Y$  có thể bằng 1 hay một số nguyên  $N > 1$ .

Cặp số  $(X, Y)$  được định nghĩa là **bản số** của đoạn thẳng  $(E, R)$  và có thể lấy các giá trị sau:  $(0, 1)$ ,  $(1, 1)$ ,  $(0, N)$ , hay  $(1, N)$ , với  $N > 1$ .

Đối với loại quan hệ nhị nguyên  $R$  liên kết giữa hai thực thể  $A$  và  $B$ , ta phân thành ba loại quan hệ cơ bản sau:

- Quan hệ 1-1 (một-một): mỗi thể hiện của thực thể A được kết hợp với 0 hay 1 thể hiện của B và ngược lại.

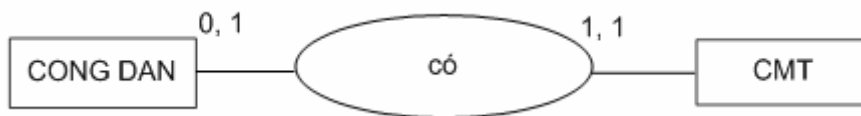


Hình 2.8. Đặc tả mối quan hệ 1-1

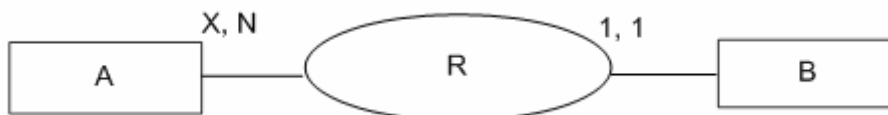
X,Y có thể lấy các giá trị 0 và 1.

Ví dụ:

Mỗi công dân trưởng thành đều được cấp một số chứng minh thư. Mỗi một số CMT chỉ dành cho 1 công dân trưởng thành.



- Quan hệ 1-N (một-nhiều): Mỗi thể hiện của thực thể A được kết hợp với 0,1 hay nhiều thể hiện của B và mỗi thể hiện của B được kết hợp với một thể hiện duy nhất của A. Đây là một loại quan hệ thông dụng và đơn giản nhất.



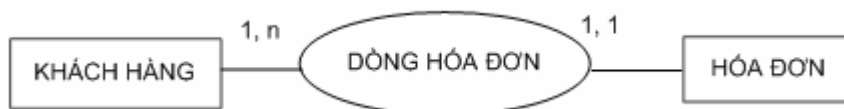
Hình 2.9. Đặc tả mối quan hệ 1-N

X có thể lấy các giá trị 0 và 1

Ví dụ:

Một khách hàng có thể có nhiều hoá đơn

Một hoá đơn chỉ mang tên một khách hàng.



- Quan hệ N-P (nhiều-nhiều): Mỗi thể hiện của một thực thể A được kết hợp với 0,1 hay nhiều thể hiện của B và ngược lại, mỗi thể hiện của B được kết hợp với 0, 1 hay nhiều thể hiện của A.



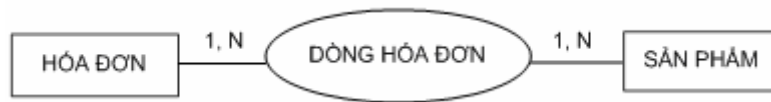
Hình 2.10. Đặc tả mối quan hệ N-N

X và Y có thể lấy giá trị 0, 1

Ví dụ:

Một hoá đơn dùng để thanh toán một hay nhiều loại sản phẩm.

Một loại sản phẩm có thể xuất hiện trong 1 hay nhiều hoá đơn.



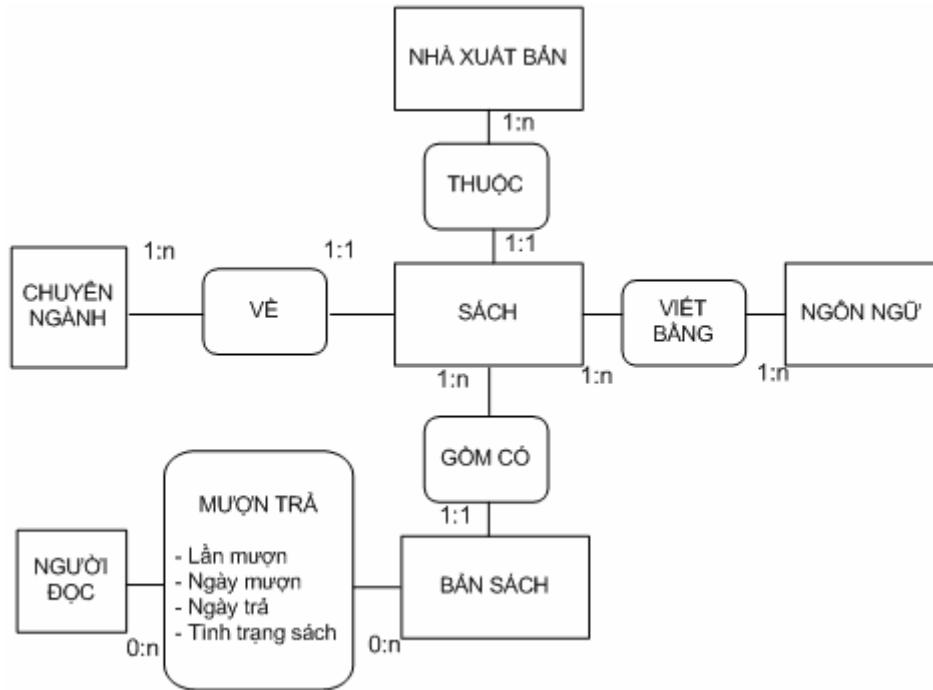
Thông thường quan hệ N-N chứa các thuộc tính (ví dụ như số lượng, giá). Chúng ta biến đổi loại quan hệ này thành các thực thể và thực thể này cần được nhận dạng bởi một khoá chính (gồm một nhóm các thuộc tính).

### 2.5.3. Mô hình khái niệm dữ liệu

Mô hình khái niệm dữ liệu là kết quả của quá trình phân tích dữ liệu mà chúng ta sẽ nghiên cứu kỹ trong chương 3. Mô hình này được xây dựng từ thực thể và các mối quan hệ giữa các thực thể đã được xác định bằng cách lắp ghép các đặc tả mối quan hệ lại với nhau thành mạng.

Ví dụ

Như trong bài toán quản lý thư viện đơn giản:



Danh sách các thực thể và thuộc tính tương ứng:

**NGƯỜI ĐỌC:** Số thẻ, Họ tên, Ngày sinh, Địa chỉ, Thời hạn hết giá trị, Số chứng minh thư, Số điện thoại, Ngày làm thẻ

**BẢN SÁCH:** Số hiệu lưu trữ, Ngày nhập về, Loại (cho mượn về nhà/đọc tại chỗ)

**SÁCH:** Số xuất bản, Tên sách, Tên tác giả (chủ biên), Tên đồng tác giả, Tập, Lần xuất bản, Số trang, Giá tiền, Đa ngữ, Năm xuất bản.

**NHÀ XUẤT BẢN:** Tên NXB, Địa chỉ NXB

**CHUYÊN NGÀNH:** Mã phân loại chuyên ngành, Tên phân loại chuyên ngành

**NGÔN NGỮ:** Mã loại ngôn ngữ, Tên loại ngôn ngữ (tiếng Việt, tiếng Anh, ...)



## Chương 3

# QUY TRÌNH PHÂN TÍCH THIẾT KẾ CSDL

Việc phân tích thiết kế CSDL bao gồm những giai đoạn cơ bản sau:

1. Phân tích yêu cầu
2. Phân tích dữ liệu và xây dựng mô hình khái niệm dữ liệu
3. Thiết kế CSDL mức logic
4. Thiết kế CSDL mức vật lý
5. Thiết kế an toàn bảo mật cho CSDL

### 3.1. PHÂN TÍCH YÊU CẦU

#### 3.1.1. Mục đích của việc phân tích yêu cầu

Trong quá trình phân tích và thiết kế CSDL thì phân tích yêu cầu là công việc phải thực hiện đầu tiên.

Chúng ta quan tâm đến những vấn đề sau:

- Các dữ liệu nào sẽ được lưu giữ trong CSDL, trong đó các dữ liệu nào là không thể thiếu?
- Các ứng dụng nào có thể sẽ được phát triển mà có thể khai thác được CSDL đã xây dựng?
- Các thao tác gì sẽ được thực hiện trên những dữ liệu đã lưu giữ?

Để đáp ứng được những vấn đề này chúng ta cần phải tìm hiểu hiện trạng, phân tích các yêu cầu của đối tượng sử dụng từ những người thừa hành, đến người quản lý và lãnh đạo vì người thừa hành có thể cung cấp các thông tin chi tiết về một công việc cụ thể, người quản lý cung cấp các nhiệm vụ, một lĩnh vực trong hệ thống, còn người lãnh

đạo cung cấp các thông tin tổng thể cùng các mục tiêu trước mắt và lâu dài của toàn hệ thống.

Có nhiều kỹ thuật và công cụ như phỏng vấn trực tiếp, phiếu điều tra, bảng hỏi, tài liệu,...được sử dụng riêng lẻ hay tích hợp trong giai đoạn này tùy thuộc vào điều kiện cụ thể khi xây dựng CSDL.

*Mục đích chính của khâu phân tích yêu cầu:*

1. Xác định yêu cầu dữ liệu của CSDL: danh sách các thuộc tính cần quản lý và gộp chúng theo các đối tượng
2. Phân loại và mô tả thông tin về các đối tượng đó
3. Xác định mối liên kết giữa các đối tượng
4. Xác định các giao dịch sẽ được thực hiện trên các đối tượng dữ liệu đó
5. Xác định các quy tắc nghiệp vụ (luật) đảm bảo tính toàn vẹn của dữ liệu

Việc phân tích yêu cầu được tiến hành thông qua phương pháp mô hình hoá với mức trừu tượng logic hay khái niệm. Kết quả của việc phân tích yêu cầu và phân tích dữ liệu là xây dựng được mô hình khái niệm dữ liệu. Trong quá trình xây dựng mô hình này, vai trò của NSD rất quan trọng. Họ có thể bổ sung cho chúng ta những khiếm khuyết của hệ thống hiện tại. Nhờ vậy, chúng ta sẽ có cơ sở để hoàn thiện mô hình với các dữ liệu đầy đủ và chính xác hơn.

*Một số lưu ý khi phân tích yêu cầu của NSD:*

1. Khi trao đổi với NSD, cần sử dụng các thuật ngữ trong thế giới thực cho dễ trao đổi và gợi mở được nhiều chi tiết nghiệp vụ
2. Tìm hiểu kỹ quy trình nghiệp vụ đủ chi tiết để mô hình hoá nó một cách đầy đủ và chính xác nhất
3. Phỏng vấn đầy đủ các đối tượng sử dụng vì đây là những người được thừa hưởng và phát huy được hiệu quả sử dụng CSDL

### 3.1.2. Nội dung phân tích yêu cầu

Việc phân tích yêu cầu về dữ liệu cho hệ thống gồm những nội dung sau:

#### 3.1.2.1. Xác định dữ liệu cần lưu giữ

Đối với hệ thống CSDL, mấu chốt vấn đề là xác định *đúng* và *đủ* danh sách thuộc tính cần quản lý. Danh sách này quyết định thành công của việc xây dựng CSDL.

Đối với mỗi dữ liệu trong danh sách, cần mô tả đầy đủ các chỉ mục dữ liệu:

- Tên dữ liệu (sát với thực tế)
- Định nghĩa (người dùng hiểu được)
- Kiểu dữ liệu (chỉ rõ kiểu của dữ liệu là ký tự, số, ngày tháng, logic,... và kích cỡ của chúng)
- Loại dữ liệu (sơ cấp: không thể chia nhỏ được nữa, tích hợp/kết nối: được gộp lại từ việc ghép nhiều dữ liệu với nhau, tính toán: được suy ra từ một số dữ liệu khác)
- Định lượng (số các giá trị khác nhau mà dữ liệu có thể nhận)
- Ghi chú thêm: cho phép chúng ta dùng hình thức diễn đạt tự do để có thể nêu thêm các dữ kiện bổ sung cho dữ liệu.

*Ví dụ:* Tên: Tên nhà cung cấp

Định nghĩa: dùng để đặt tên, thường tóm tắt, cho phép xác định nhà cung cấp

Cấu trúc: kiểu ký tự C(30)

Loại: sơ cấp

Định lượng: khoảng 50

*Ví dụ:* Công ty LƯƠNG THỰC THỰC PHẨM I

Định nghĩa: Công ty Lương thực thực phẩm ở phu vực phía Bắc (tương ứng II - miền Trung, III - miền Nam)

*Ghi chú:* có thể chỉ sử dụng 15 ký tự.

### **3.1.2.2. Xác định ứng dụng sẽ được cài đặt trên CSDL**

Việc xác định ứng dụng sẽ được cài đặt trên CSDL là rất quan trọng vì đó là cơ sở xác định tính đúng và đủ của các dữ liệu và giúp cho việc nhìn nhận tổng quát hơn về CSDL sẽ được xây dựng.

### **3.1.2.3. Xác định các thao tác thường xuyên được thực hiện**

Việc xác định các thao tác thường xuyên được thực hiện trên dữ liệu đóng vai trò quan trọng. Yếu tố này đôi khi lại quyết định tính thành công của một CSDL. Nhờ có yếu tố này mà việc thiết kế một CSDL mới được coi là đầy đủ.

Trên thực tế, khi lượng dữ liệu tăng lên thì thời gian thực hiện các thao tác trên dữ liệu cũng tăng lên. Do vậy, nếu trong khâu khảo sát, phân tích yêu cầu, chúng ta không xác định được các thao tác thường xuyên được thực hiện thì chúng ta không thể đưa ra được thiết kế hiệu quả, không thể tạo ra những chỉ mục phù hợp, cách phân chia dữ liệu hợp lý.

Đặc biệt, việc xác định các thao tác thường xuyên được thực hiện trên dữ liệu rất có ích trong quá trình tinh chỉnh sơ đồ CSDL logic cũng như thiết kế CSDL vật lý.

### **3.1.3. Các kỹ thuật phân tích yêu cầu**

Một số kỹ thuật phân tích yêu cầu:

#### **3.1.3.1. Phỏng vấn trực tiếp**

Trước hết, chúng ta phải chọn lựa đúng đối tượng cần phỏng vấn. Sau đó tiến hành phỏng vấn theo cấu trúc phân cấp: ban lãnh đạo, các bộ phận nghiệp vụ phòng ban, và sau cùng là các đơn vị cơ sở (các vị trí làm việc). Trong quá trình phỏng vấn, chúng ta cố gắng thu được đầy đủ các mẫu hồ sơ dữ liệu (đầu vào, trung gian), các báo cáo thống kê (đầu ra). Việc phỏng vấn ban lãnh đạo nhằm mục đích tìm hiểu các chỉ tiêu tổng thể mang tính chiến lược của cả tổ chức, trong từng lĩnh vực, từng bộ phận. Việc phỏng vấn các phòng ban nghiệp vụ nhằm

mục đích thu thập các chỉ tiêu trọng yếu, cần thiết cho mọi hoạt động của tổ chức mà họ thường dùng để theo dõi, quản lý các hoạt động nghiệp vụ và chỉ đạo trong tổ chức. Việc phỏng vấn các vị trí làm việc nhằm tiếp nhận các thông tin cụ thể, chi tiết cho từng hoạt động nghiệp vụ trong từng lĩnh vực cụ thể, từng bộ phận nhỏ trong phạm vi bao quát của các chỉ tiêu đã thu được từ phỏng vấn lãnh đạo và các ban ngành chức năng. Chúng ta cần chi tiết từng xử lý, tương ứng các dữ liệu đầu vào, nguồn dữ liệu cung cấp, các dữ liệu đầu ra, nơi gửi thông tin đến. Chúng ta có thể tiến hành phỏng vấn cá nhân tham gia trực tiếp vào từng công việc cụ thể. Để có thể liệt kê và mô tả toàn bộ công việc phải thực hiện, đối với mỗi công việc cần nắm bắt các sự kiện khởi động, chu kỳ thực hiện, thời lượng thực hiện, những dữ liệu có liên quan, khối lượng của chúng, các quy tắc cần áp dụng để thực hiện công việc, quan sát sự luân chuyển dữ liệu giữa các vị trí công việc.

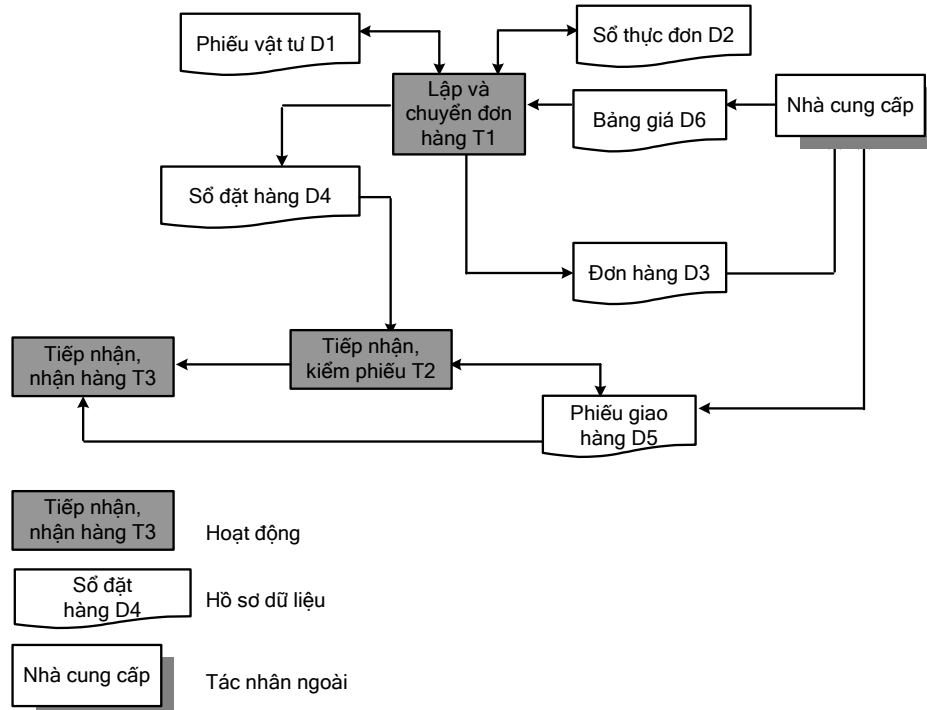
Để nâng cao chất lượng và tốc độ phỏng vấn, chúng ta có thể dùng các *phiếu phỏng vấn* đã chuẩn bị sẵn. Sau khi phỏng vấn cần tổng hợp, hiệu chỉnh dữ liệu thu thập được, sắp xếp chúng dưới dạng các hồ sơ khảo sát và xác định các dữ liệu còn chưa chính xác, các dữ liệu còn thiếu, cần thu thập tiếp.

Phiếu phỏng vấn bao gồm các thông tin về dự án, người được phỏng vấn, nội dung cần phỏng vấn (thông qua câu hỏi chuẩn bị trước), đánh giá sau phỏng vấn, lịch biểu tiếp tục (thỏa thuận), thời gian phỏng vấn, tên người phỏng vấn.

*Khi phỏng vấn, chúng ta cần:*

+ *Vẽ Lưu đồ công việc* nhằm xác định đầy đủ các hồ sơ dữ liệu:

Dự án:	Tiểu dự án: QL kho	Trang:
Loại: Lưu đồ	Nhiệm vụ: <i>Lập đơn hàng</i>	Ngày:
Thủ kho	Người quản lý	Bên ngoài



Hình 3.1. Lưu đồ công việc bài toán quản lý kho

+ Lập *Bảng tổng hợp hồ sơ* (Rút từ các phiếu phỏng vấn, tài liệu thu thập) nhằm mô tả quan hệ giữa xử lý và dữ liệu và dùng để phục vụ cho việc tổng hợp dữ liệu sau này.

Bảng 3.1. Bảng tổng hợp hồ sơ

Mã tài liệu	Tên - Vai trò	Công việc liên quan (mã chức năng/nhiệm vụ)
D1	Phiếu vật tư: ghi hàng hoá xuất hay nhập	T1
D2	Sổ thực đơn: Định mức hàng hoá làm sản phẩm	T1
D3	Đơn đặt hàng: Theo mẫu nhà cung cấp	T1
D4	Sổ đặt hàng: tập hợp các đơn hàng	T1,T2
D5	Phiếu giao hàng: Nhà cung cấp phát ra	T2,T3
...		...

+ Lập *Bảng tổng hợp công việc* (rút ra từ phiếu phỏng vấn) nhằm phục vụ cho việc tổng hợp xử lý sau này.

*Bảng 3.2. Bảng tổng hợp công việc*

STT	Mô tả công việc	Vị trí làm việc	Tần suất	Hồ sơ vào	Hồ sơ ra
T1	<i>Lập đơn hàng</i> : Xuất phát từ nhu cầu cung ứng, thực đơn sản xuất, báo giá, đơn hàng lập và chuyển đi bằng điện thoại (80%), viết (20%) sắp các đơn hàng vào sổ đặt để đối chiếu, theo dõi	Quản lý kho hàng	4-5 đơn/ngày 5-10 dòng/đơn	D1 D2 D6	D3 D4
T2					

+ Lập *Bảng mô tả chi tiết tài liệu* (từ điển dữ liệu) nhằm phục vụ cho việc xây dựng bảng tổng hợp từ điển dữ liệu sau này.

*Bảng 3.3. Bảng mô tả chi tiết tài liệu*

Dự án:	Tiểu dự án: <i>Lập đơn hàng</i>	Trang
Loại: phân tích hiện trạng	Mô tả dữ liệu	Số TT: Ngày:
Định nghĩa:	Tên dữ liệu: <i>Nhà cung cấp</i> Dùng để chỉ những người cung cấp hàng thường xuyên. Nó cho phép xác định mỗi nhà cung cấp	
Cấu trúc:	Kiểu ký tự, độ dài 20-30 ký tự, một số chữ đầu viết hoa	
Loại:	Sơ cấp	
Số lượng:	50 nhà cung cấp	
Ví dụ:	Công ty NETNAM	
Lời bình:	Tên có thể viết đầy đủ hoặc viết tắt	

*Sau phỏng vấn, chúng ta cần tổng hợp khảo sát:*

- Một tổ chức lớn, phức tạp thường không thể quan sát tất cả một lúc, bằng cách phân tích và tách nhỏ để quan sát, thu thập thông tin. Lúc này cần lắp ghép lại để có được một cách nhìn tổng thể

- Thực hiện tổng hợp hai loại

- + Tổng hợp **xử lý**: để thấy được tổng thể các xử lý diễn ra trong tổ chức và phát hiện xử lý nào không thay đổi trong mọi trường hợp

- + Tổng hợp các **dữ liệu**: đảm bảo tính phù hợp và chặt chẽ của dữ liệu đã sử dụng

### ***Tổng hợp dữ liệu***

*Mục tiêu*: liệt kê ra tất cả các dữ liệu có liên quan đến miền khảo sát của tổ chức và sàng lọc để thu được những dữ liệu đầy đủ, chính xác và gán cho tên gọi thích hợp mà mọi người tham gia dự án đồng ý.

*Sàng lọc dữ liệu*:

- Liệt kê các thông tin thu được, tiến hành so sánh, đối chiếu, xem xét sự phù hợp của nó về mặt nội dung, cũng như tên gọi.

Sự không phù hợp của dữ liệu thường xảy ra khi:

- + Dữ liệu **đồng nghĩa**: hai tên khác nhau nhưng chỉ cùng một loại dữ liệu.

*Ví dụ*:

Đơn giá và giá đơn vị.

- + Dữ liệu **đồng danh/đa nghĩa**: cùng một tên gọi nhưng lại chỉ nhiều loại dữ liệu khác nhau.

*Ví dụ*:

Tài khoản ở một chỗ có nghĩa là số của một tài khoản, ở chỗ khác có nghĩa là tên đầy đủ của một tài khoản thanh toán. Ngày (xuất/nhập), Giá (bán/mua), Số lượng (nhập/xuất)

- Loại bỏ tên trùng, cho tên khác nhau với các dữ liệu khác nhau

### ***Lập một từ điển dữ liệu***

- Tập hợp các dữ liệu thu được từ các cuộc phỏng vấn điều tra, sau khi đã sàng lọc lập thành một tài liệu chung. Rõ ràng trong bảng tổng hợp này, các dữ liệu đều có tên khác nhau.



- Từ điển dữ liệu là 1 tư liệu về mọi tên gọi của mọi đối tượng được dùng trong hệ thống ở cả 4 giai đoạn: phân tích, thiết kế, cài đặt và bảo trì. Tài liệu này là cơ sở để tra cứu và sử dụng sau này cho việc hình thành CSDL.

*Mẫu bảng tổng hợp từ điển dữ liệu có dạng (thuộc tính):*

*Bảng 3.4. Bảng tổng hợp từ điển dữ liệu*

STT	Tên gọi	Kiểu	Cỡ	Lĩnh vực	Quy tắc, ràng buộc và khuôn dạng
1	Số hoá đơn	Ký tự	8	Kế toán	Chữ hoa hoặc số
2	Tên hàng hoá	Ký tự	20	Kế toán	Chữ hoa + số
3	Ngày vào sổ đơn hàng	Ngày tháng	8	Quản lý kho	ngày/tháng/năm
4	Số lượng nhận hàng	Số	6	Thủ kho	Đơn vị: tạ, tấn.
5	Số tiền	Số	7	Kế toán	Giữ 2 số lẻ sau dấu phẩy

*Sau phỏng vấn, chúng ta cần hợp thức hoá kết quả phỏng vấn.*

Hợp thức hoá là việc *hiểu và thể hiện* các thông tin phỏng vấn ở các dạng khác nhau được nhiều người sử dụng và tổ chức chấp nhận sự mô tả nhu cầu thông tin của tổ chức là *đúng và đủ*.

Mục tiêu của việc hợp thức hoá kết quả phỏng vấn nhằm chính xác hoá các thông tin được khảo sát đồng thời người được phỏng vấn và người đại diện xác nhận sự phù hợp của nó với hiện trạng tổ chức của họ và phù hợp với mục tiêu đề ra.

*Sau phỏng vấn, chúng ta cũng cần đánh giá vấn đề và phân tích tổng hợp các thông tin hữu ích* nhằm xác định được một cách tiếp cận hay giải pháp tổng thể.

Khi đánh giá, tổng hợp, người phân tích cần xây dựng các mô hình hệ thống nhằm hiểu rõ hơn luồng dữ liệu và các xử lý, thao tác dữ liệu và nội dung thông tin. Các mô hình này sẽ là nền tảng cho thiết kế sau này và là cơ sở cho việc tạo ra một đặc tả hệ thống và phần mềm.

### 3.1.3.2. Phỏng vấn gián tiếp

Phỏng vấn gián tiếp thông qua bảng hỏi hay *phiếu khảo sát* là hình thức khảo sát bổ sung cho kỹ thuật phỏng vấn trực tiếp. Các câu hỏi được liệt kê trong bảng hỏi hay phiếu khảo sát. Người được khảo sát sẽ tự điền vào phiếu này những thông tin được hỏi. Ưu điểm của phương pháp này là đỡ tốn kém và có thể mở rộng được diện khảo sát nhưng thiếu sự giao tiếp giữa chúng ta và người được phỏng vấn. Hơn nữa, người được khảo sát có thể bỏ qua những thông tin mà chúng ta cần.

Ví dụ: Hình 3.2.

PHIẾU THU THẬP THÔNG TIN  
VỀ ỨNG DỤNG CNTT TRONG CÁC DOANH NGHIỆP

**A. THÔNG TIN CHUNG**

1. Tên doanh nghiệp: .....

2. Địa chỉ: .....

3. Điện thoại: .....

4. Fax: .....

5. E-mail: .....

6. Web site: .....

7. Họ và tên Giám đốc: .....

8. Tổng số nhân viên của doanh nghiệp: .....

**B. PHẦN CÂU HỎI CHÍNH**

9. Tổng số máy tính (PC) hiện có của doanh nghiệp: .....

Trong đó:

a. Số máy chủ (Server): .....

b. Số máy tính cá nhân (PC): .....

c. Số máy xách tay (Laptop/Notebook) .....

d. Thiết bị ngoại vi:

- Máy in laser.....

Dưới HP 1100.....

Trên HP1100 (trên 8 trang/phút).....

- Máy in kim.....
- Scanner.....
- Flotter.....
- Modem.....
- Switch.....
- Router.....
- Hub.....

e. Thiết bị hỗ trợ:

- Điều hòa nhiệt độ:.....
- Máy hút ẩm:.....
- Ổn áp:.....
- Máy lưu điện:.....
  - Dưới 3kVA.....
  - Trên 3kVA.....
- Chống sét.....

**10. Doanh nghiệp có Trung tâm/Phòng máy tính không?**

- |          |                          | Số lượng |
|----------|--------------------------|----------|
| 1. Có    | <input type="checkbox"/> | .....    |
| 2. Không | <input type="checkbox"/> |          |

**11. Các máy tính của doanh nghiệp đã được kết nối mạng nội bộ (LAN) chưa?**

- |          |                          | Từ bao giờ |
|----------|--------------------------|------------|
| 1. Đã có | <input type="checkbox"/> | .....      |
| 2. Chưa  | <input type="checkbox"/> |            |

**12. Việc kết nối mạng nội bộ (LAN) của doanh nghiệp**

- |   |                          |
|---|--------------------------|
| 1. Tất cả các máy tính đều được nối mạng                            | <input type="checkbox"/> |
| 2. Chỉ có các máy tính trong Trung tâm/Phòng máy tính được nối mạng | <input type="checkbox"/> |
| 3. Không có máy nào được nối mạng                                   | <input type="checkbox"/> |
| 4. Số máy tính được nối mạng  | <input type="checkbox"/> |
| 5. Số máy chủ trong mạng  | <input type="checkbox"/> |
| 6. Số máy in trong mạng   | <input type="checkbox"/> |

**13. Hệ điều hành mạng**

- |                                    |                          |
|------------------------------------|--------------------------|
| WINDOW NT                          | <input type="checkbox"/> |
| Windows 2000 Server/Advance Server | <input type="checkbox"/> |

NOVELL

UNIX/LINUX

Khác (ghi tên cụ thể)


**14. Doanh nghiệp hiện đang sử dụng dịch vụ kết nối Internet nào?**

Từ bao giờ

1. Kết nối gián tiếp (Dial-up)

--

.....

2. Internet tốc độ cao (ADSL)

--

**15. Việc kết nối mạng Internet của doanh nghiệp**

1. Tất cả các máy tính đều được nối mạng Internet

--

2. Chỉ có các máy tính trong Trung tâm/Phòng máy tính được nối mạng Internet

--

3. Không có máy nào được nối mạng Internet

--

4. Số máy tính được nối mạng Internet

--

**16. Việc sử dụng địa chỉ E-mail của doanh nghiệp**

Có

Không

a.	Doanh nghiệp có địa chỉ e-mail giao dịch chính thức			
----	---	--	--	--

Địa chỉ E- mail:.....

**17. Doanh nghiệp có website không?**

Từ bao giờ

Địa chỉ website

1. Đã có

	.....	.....
--	-------	-------

2. Chưa

--	--	--

**18. Có bao nhiêu nhân viên của doanh nghiệp biết sử dụng máy tính?****19. Có bao nhiêu nhân viên của doanh nghiệp biết sử dụng Internet?****20. Doanh nghiệp có bao nhiêu nhân viên chuyên trách về CNTT?****21. Doanh nghiệp sử dụng máy tính cho những mục đích nào dưới đây**

1. Soạn thảo văn bản

--

2. Kế toán

--

3. Quản lý, điều hành công việc

--

4. Dịch vụ

--

5. Phục vụ một số công đoạn trong dây chuyền sản xuất

--

**22. Doanh nghiệp sử dụng Internet cho những mục đích nào dưới đây?**

1. Gửi E-mail

--

2. Quản lý, điều hành công việc

--

3. Thu thập thông tin

--

4. Quảng cáo cho doanh nghiệp
5. Mua bán hàng hóa, vật tư....
6. Dịch vụ


**23. Doanh nghiệp có cơ sở dữ liệu không?**

1. Có
2. Không


Các hệ quản trị CSDL đang dùng (nếu có thì đánh dấu x vào ô dưới):

Foxpro	Access	Notus note	DB2	SQL server	Oracle	CSDL địa lý	Infomix	Khác

**24. Đầu tư cho ứng dụng CNTT tại doanh nghiệp trong giai đoạn**

**2005-2010.....đồng**

Ngân sách trung ương: .....đồng

Ngân sách địa phương: .....đồng

Đầu tư phát triển mạng diện rộng, mạng cục bộ, mua sắm phần cứng:.....đồng

Đầu tư phát triển ứng dụng CNTT (mua, xây dựng cơ sở dữ liệu, phần mềm...):.....đồng

Đầu tư cho đào tạo, bồi dưỡng về CNTT cho cán bộ, nhân viên: .....đồng

**25. Các phần mềm ứng dụng đang sử dụng trong đơn vị:**

Dạng ứng dụng (ghi rõ tên ứng dụng)	Hiện trạng sử dụng		Trên máy đơn	Ngôn ngữ lập trình	Nhà cung cấp
	Năm bắt đầu sử dụng	Đáp ứng được yêu cầu			
a) Phần mềm ứng dụng cơ bản					
- Soạn thảo văn bản					
- Quản lý công văn					
- Quản lý VB PQ					
- Quản lý nhân sự					
- Quản lý tài chính					
- Quản lý vật tư					
- Đơn khiếu tố					
- Điều hành tác nghiệp					

b) Các ứng dụng và CSDL chuyên ngành phục vụ cho lĩnh vực chuyên môn					
c) Các ứng dụng trong sản xuất (điều hành sản xuất)					
d) Khác (nếu có)					

**26. Đào tạo**

Nội dung đào tạo	Số học viên được đào tạo qua các năm					Nơi đào tạo		
	Năm ...					Tại địa phương	Ngoài địa phương	Nước ngoài
Tổng số								
1. CNTT cho cán bộ lãnh đạo, quản lý								
2. Tin học cho NSD								
3. Chuyên viên tin học								
4. Phát triển và quản trị hệ thống								

**27. Nhu cầu ứng dụng và phát triển CNTT của đơn vị đến năm ...***1. Nhân lực về CNTT*

Nhu cầu tổng số cán bộ nhân viên toàn đơn vị đến năm .....: ... người

Trong đó, nhu cầu số cán bộ về CNTT đến năm .....:

Chỉ tiêu	Tổng số (người)	Trong đó				
		ĐH và trên ĐH	Cao đẳng	Trung cấp	Chuyên viên	Khác
Cán bộ phần cứng						
Cán bộ phần mềm						
Phát triển và quản trị hệ thống						
Cán bộ có sử dụng PC trong công việc						
Nhập dữ liệu						

*2. Nhu cầu đào tạo từng năm*

Chỉ tiêu	Năm...				
Cán bộ quản lý					
Cán bộ quản trị mạng					

Chuyên viên lập trình					
Chuyên viên phân tích thiết kế hệ thống					
Chuyên viên phát triển và quản trị hệ thống					
Chuyên viên TMDT					

3. Nhu cầu nhân lực có trình độ cao:

ĐV tính: người

Năm	Số cán bộ						
	Phổ cập	ĐH			Trên ĐH		
		Lập trình	Phân tích và thiết kế	Mạng	Lập trình	Phân tích và thiết kế	Mạng

4. Nhu cầu về thiết bị

ĐV: chiếc

STT	Chủng loại	Số lượng				
		Năm....				
1.	Máy tính PC					
2.	Máy in					
3.	Scanner					
4.	Plotter					
5.	Modem					
6.	Router					
7.	Switch					
8.	Máy chủ CSDL					
9.	Máy chủ truyền thông					

5. Nhu cầu các hệ quản trị CSDL chủ yếu (nếu có thì đánh dấu x vào ô dưới):

Foxpro	Access	Notus note	DB2	SQL server	Oracle	CSDL địa lý	Infomix	Khác

6. Nhu cầu các phần mềm ứng dụng sẽ sử dụng trong đơn vị:

Dạng ứng dụng (ghi rõ tên ứng dụng, năm bắt đầu sử dụng)	Nhu cầu		Trên máy đơn	Ngôn ngữ lập trình	Nhà cung cấp
	Nâng cấp	Mới			
a) PM ứng dụng cơ bản					
- Soạn thảo văn bản					
- Quản lý công văn					
- Quản lý VB PQ					

- Quản lý nhân sự					
- Quản lý tài chính					
- Quản lý vật tư					
- Đơn khiếu tố					
- Điều hành tác nghiệp					
b) Các ứng dụng và CSDL chuyên ngành phục vụ cho lĩnh vực chuyên môn					
c) Các ứng dụng phục vụ sản xuất (điều hành sản xuất)					
d) Khác (nếu có)					

#### 7. Nhu cầu về tổ chức

Có nhu cầu thành lập đơn vị chuyên trách về CNTT:

Nếu có: Số lượng cán bộ: .....

Trong đó:

- Cán bộ kỹ thuật: .....
- Cán bộ lập trình: .....
- Cán bộ phân tích và thiết kế hệ thống: .....

### 28. Đánh giá và kiến nghị

#### 1. Lý do phát triển chậm CNTT của doanh nghiệp

- Nhu cầu sử dụng CNTT còn ít
- Không hoạch định được lộ trình áp dụng CNTT
- Thiếu sự kết hợp chặt chẽ của các hiệp hội
- Nhà nước chưa chú trọng đến lớp ứng dụng cho doanh nghiệp
- Ngân sách có hạn
- Lo ngại về hiệu quả sử dụng phần mềm
- Sự chỉ đạo của cấp trên còn ít
- Lãnh đạo doanh nghiệp còn bận quá nhiều việc
- Đầu tư không hợp lý
- Thiếu sự liên kết, phối hợp, không theo kế hoạch tổng thể
- Thiếu nhân lực (lập trình viên, cán bộ quản lý, cán bộ xúc tiến thương mại)
- Năng lực của doanh nghiệp cung cấp sản phẩm CNTT còn hạn chế
- Chính sách, đề án hỗ trợ doanh nghiệp ứng dụng CNTT còn thiếu



- Thiếu Trung tâm cung cấp dữ liệu, thông tin kinh tế chuyên ngành
- Thiếu kích thích (chương trình phát triển TMĐT, tin học hoá QLNN)
- Chi phí xây dựng phần mềm quá cao
- Nhân viên ngại không muốn thay đổi thói quen làm việc cũ
- Quy trình sản xuất kinh doanh ở nhiều đơn vị thành viên không chuẩn
- Chưa coi thông tin là 1 yếu tố không thể thiếu trong điều hành sản xuất kinh doanh
- Dịch vụ không tốt của các nhà cung cấp dịch vụ CNTT trong, ngoài nước

**2. Các trở ngại chính khi thực hiện ứng dụng**

(ghi theo thứ tự quan trọng: 1 - quan trọng nhất, 9 - ít quan trọng nhất)

Trình độ chung về tin học của cán bộ còn thấp

Thiếu các chuyên gia tin học

Các phần mềm đã xây dựng không phù hợp

Kế hoạch không phù hợp

Trình độ tổ chức quản lý còn hạn chế

Không đủ kinh phí

Thiếu sự quan tâm của lãnh đạo

Không được sự ủng hộ hợp tác của cán bộ công chức trong cơ quan

Khác (nếu có ghi rõ)

**3. Các kiến nghị của doanh nghiệp về ứng dụng CNTT tại doanh nghiệp**

Ngày tháng năm...

Thủ trưởng đơn vị

**3.1.3.3. Phân tích tài liệu**

Các tài liệu đã có là nguồn dữ liệu và thông tin dồi dào và đã được tích lũy qua nhiều năm. Chúng có thể tồn tại dưới dạng cứng (sổ, sách, tập tư liệu) hay mềm (các tệp dữ liệu). Chúng ta cố gắng có đầy đủ danh sách của chúng. Sau đó phân loại dữ liệu trong từng tài liệu thành các dữ liệu cơ bản hay các dữ liệu có cấu trúc, xem xét, phát hiện các dữ liệu trùng lặp, thiếu nhất quán trong lưu trữ, tên gọi. Nếu tổ chức đó đã có CSDL thì cố gắng tìm hiểu nguyên do tại sao họ không muốn dùng CSDL này nữa.

*Ví dụ:* về Hồ sơ dữ liệu Bài toán quản lý kinh doanh:

**Bảng 3.5. Hệ thống bảng Vào**

<b>1. ĐƠN ĐẶT MUA HÀNG</b>					
Số:.....			Ngày giao hàng:.....		
Ngày:.....			Ngày thanh toán:.....		
Tên nhà cung cấp:.....			Tổng số tiền:.....đồng		
Địa chỉ nhà cung cấp:.....					
Điện thoại nhà cung cấp:...					
E-mail nhà cung cấp:.....					
Số thứ tự	Tên hàng	Đơn vị tính	Số lượng	Đơn giá	Thành tiền
Tổng					

<b>2. PHIẾU NHẬP</b>					
Số:.....					
Theo số đơn đặt mua hàng:.....					
Tên nhà cung cấp:.....					
Ngày:.....					
Tổng tiền nhập:.....đồng					
Số thứ tự	Tên hàng	Đơn vị tính	Số lượng	Đơn giá	Thành tiền
Tổng					

<b>3. PHIẾU CHI</b>					
Số:.....					
Theo số phiếu nhập:.....					
Tên nhà cung cấp:.....					
Ngày:.....					
Tổng tiền chi:.....đồng					

#### 4. ĐƠN ĐẶT HÀNG

Số:..... Ngày:.....  
 Tên khách:..... Ngày để nghị giao:  
 Số nhà:..... Ngày thanh toán:.....  
 Địa chỉ:..... Tổng số tiền:.....đồng  
 Điện thoại:.....  
 E-mail:.....

STT	Tên hàng	Đơn vị tính	Số lượng	Đơn giá	Thành tiền
Tổng					

#### 5. PHIẾU GIAO

Số:.....  
 Theo số đơn đặt hàng.....  
 Tên khách:.....  
 Ngày:.....  
 Tổng tiền giao:..... đồng

STT	Tên hàng	Đơn vị tính	Số lượng	Đơn giá	Thành tiền
Tổng					

#### 6. PHIẾU THU

(phát cho khách hàng)

Số:.....  
 Theo số phiếu giao:.....  
 Tên khách:.....  
 Ngày:.....  
 Tổng tiền thu:.....đồng

**Bảng 3.6. Hệ thống bảng Ra****7. BÁO CÁO HÀNG GIAO**

Số:

Trang:

Ngày:

STT	Mã khách	Mã hàng	Đơn vị tính	Đơn giá	Số lượng	Thành tiền
Tổng						

**8. BÁO CÁO HÀNG NHẬP**

Số báo cáo:

Trang:

Ngày:

STT	Mã nhà cung cấp	Mã hàng	Đơn vị tính	Đơn giá	Số lượng	Thành tiền
Tổng						

**9. BÁO CÁO HÀNG BÁN CHẠY**

Số báo cáo:

Ngày:

Trang:

STT	Mã hàng	Đơn vị tính	Đơn giá	Số lượng	Thành tiền
Tổng					

**10. BÁO CÁO HÀNG BÁN CHẬM**

Số báo cáo:

Ngày:

Trang:

STT	Mã hàng	Đơn vị tính	Đơn giá	Số lượng	Thành tiền
Tổng					

**11. BÁO CÁO KHÁCH HÀNG THÂN THIẾT**

Số báo cáo:

Trang:

STT	Mã khách	Mã hàng	Đơn vị tính	Đơn giá	Số lượng	Thành tiền
Tổng						

Ngày....tháng...năm.....

### 12. BÁO CÁO NỢ CỦA KHÁCH

Số báo cáo:

Trang:

STT	Mã khách	Tên	Địa chỉ	Số điện thoại	E-mail	Hạn thanh toán	Tiền
	Tổng						

Ngày....tháng...năm...

### 13. BÁO CÁO NỢ NHÀ CUNG CẤP

Số báo cáo:

Trang:

STT	Mã nhà cung cấp	Tên	Địa chỉ	Điện thoại	E-mail	Hạn thanh toán	Tiền
	Tổng						

Ngày....tháng.... năm....

### 14. BÁO CÁO DANH SÁCH HÀNG VÀ LƯỢNG TỒN

Ngày kiểm kê:

STT	Mã hàng	Tên hàng	Đơn vị tính	Giá hiện tại	Số lượng	Giá trị hàng tồn
	Tổng					

Tên người báo cáo

Ngày....tháng.... năm...

### 15. BÁO CÁO VỀ NHÀ CUNG CẤP CHỦ YẾU

STT	Mã nhà cung cấp	Tên NCC	DS mặt hàng chính cung cấp	Ngày bắt đầu quan hệ	VIP	Địa chỉ liên hệ	Ghi chú

Tên người báo cáo

Ngày....tháng..... năm....

### 16. DANH SÁCH KHÁCH HÀNG TIỀM NĂNG

STT	Mã khách	Tên khách	Địa chỉ	VIP	DS mặt hàng mua chủ yếu	Tên người đại diện	Địa chỉ người đại diện

Ngày....tháng..... năm ....

### 3.2. PHÂN TÍCH DỮ LIỆU VÀ XÂY DỰNG MÔ HÌNH KHÁI NIỆM DỮ LIỆU

*Ý nghĩa của mô hình khái niệm dữ liệu:*

Mô hình khái niệm dữ liệu:

- Là công cụ kết nối giữa người PT\_TK và NSD
- Là mô hình dữ liệu mức khái niệm, mô tả cấu trúc và các ràng buộc của dữ liệu trong CSDL
- Độc lập với hệ quản trị dữ liệu và mô hình dữ liệu sẽ được sử dụng để cài đặt.
- Được biểu diễn dạng sơ đồ với 3 thành phần chính là kiểu thực thể, thuộc tính, liên kết (mối quan hệ)

Quá trình xây dựng mô hình khái niệm dữ liệu có thể được chia làm các giai đoạn sau đây:

#### *1. Khảo sát thực tế*

- Thu thập thông tin
- Trình bày có hệ thống dưới dạng sơ đồ luân chuyển các tài liệu

#### *2. Thiết lập mô hình dữ liệu*

- Kiểm kê các dữ liệu
- Xác định các phụ thuộc hàm
- Xây dựng mô hình khái niệm dữ liệu

#### *3. Kiểm soát và chuẩn hóa mô hình*

##### **3.2.1. Khảo sát thực tế**

Mục tiêu của giai đoạn này bao gồm quá trình quan sát, phỏng vấn, tìm hiểu và phân tích, mô tả đầy đủ hiện trạng, mô tả các bài toán nghiệp vụ và các nhu cầu của người sử dụng mà hệ thống tin khi thiết kế được đòi hỏi. Do đó, nó không chỉ giới hạn trong việc xây dựng mô hình dữ liệu mà còn là cơ sở phát triển những thông tin cần thiết cho việc xây dựng mô hình chức năng.

Để đạt mục tiêu này cần quan sát được dưới mọi dạng: *Tĩnh* (dữ liệu sơ cấp, tài liệu, báo cáo,...), *động* (sự luân chuyển các thông tin, tài liệu) và các *nguyên nhân biến đổi* của chúng (thủ tục, quy tắc quản lý, công thức tính toán,...).

Các thông tin thu thập phải đầy đủ và chính xác, vì chúng là nền tảng của hệ thống tin tương lai. Nhưng cũng không nên đi quá sâu vào chi tiết, phải biết gạt bỏ những thông tin không cần thiết để không làm chệch hướng và gây khó khăn, nặng nề cho việc phân tích và thiết kế.

Công việc khảo sát không chỉ tập trung hoàn toàn vào giai đoạn đầu của quá trình phân tích - thiết kế, mà có thể trải dài trong suốt quá trình để thu thập thêm thông tin, đào sâu vấn đề hay kiểm chứng một giả thiết cùng với người sử dụng.

Khi hệ thống tin cần xây dựng quá lớn và phức tạp, ta nên chia nhỏ nó ra thành nhiều tiểu hệ. Mỗi tiểu hệ có thể được khảo sát, phân tích hay thiết kế độc lập với nhau, trước khi được tổng hợp lại. Có khi người ta đi đến phát triển cả các tiểu hệ này, những phần mềm độc lập nhưng có thể trao đổi thông tin với nhau.

Để chia một thông tin thành nhiều tiểu hệ, người ta thường sử dụng một trong hai phương pháp:

- + Phương pháp “giếng”: Các tiểu hệ độc lập được định ra, dựa trên cơ sở bài toán, chức năng nghiệp vụ chủ yếu của tổ chức. Đôi khi dựa trên một kế hoạch thực hiện theo thứ tự ưu tiên hay để thỏa mãn những yêu cầu về thời gian.

- + Phương pháp “ao”: Một cuộc khảo sát tổng quát sơ khởi sẽ cho phép nhận diện những tiểu hệ tương đối độc lập với nhau.

#### **3.2.1.1. Thu thập thông tin**

Công việc này chủ yếu là tham khảo tài liệu sưu tập được và tiếp xúc với những người sử dụng, đòi hỏi những khả năng nhân sinh như: Kinh nghiệm, óc quan sát, phê phán, giao tiếp và ứng biến,... Các phương pháp gò bó, cứng nhắc sẽ chẳng đem lại kết quả mong muốn.

Do đó phần này chỉ liệt kê và phân loại thông tin có thể gặp được trong quá trình khảo sát để trợ giúp trí nhớ.

Một sơ đồ phân loại thông tin sẽ được trình bày ở trang sau.

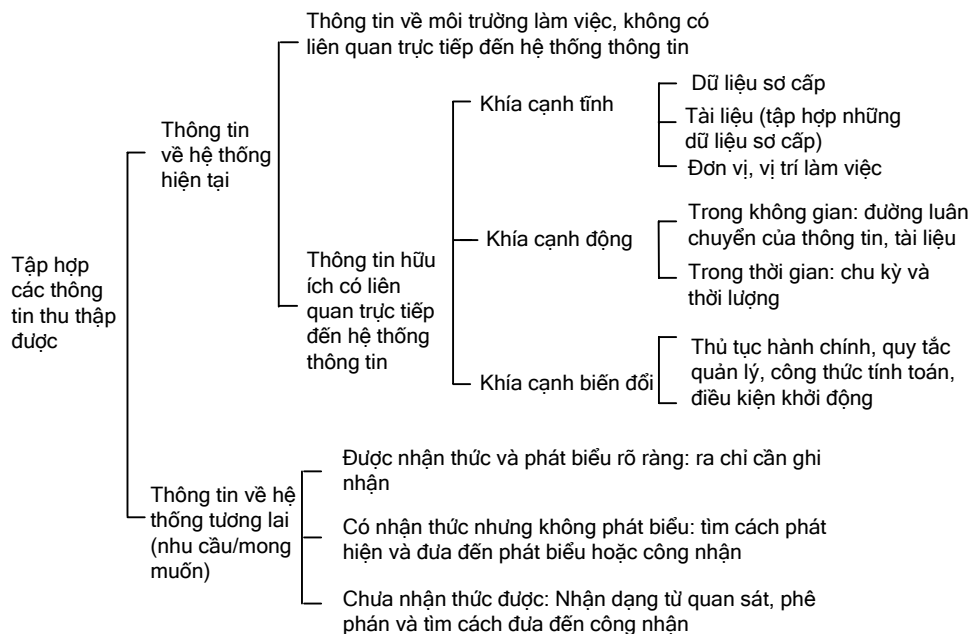
Với mỗi loại thông tin nêu lên trong sơ đồ này, nếu cần ta có thể tìm hiểu thêm về một số khía cạnh khác như: Số lượng, chu kỳ, độ chính xác cần có, người chịu trách nhiệm,...

Trong quá trình khảo sát cần lưu ý các điểm sau:

+ Nếu có thể, các cuộc phỏng vấn phải được tiến hành tuân tự theo cấu trúc phân cấp của tổ chức theo từng bộ phận, lĩnh vực, chức năng hay đi từ cấp lãnh đạo qua cấp quản lý đến người thừa hành.

+ Phải luôn nhớ sao chụp mẫu các hồ sơ, tài liệu để có được cấu trúc chính xác các thông tin làm căn bản cho việc xây dựng mô hình dữ liệu sau này.

+ Cần thiết nhất là luôn phân biệt những thông tin *nói về* hệ thống tin đang xây dựng với những thông tin *thuộc về* hệ này.



Hình 3.3. Sơ đồ phân loại các thông tin



**3.2.1.2. Trình bày hệ thống sơ đồ luân chuyển tài liệu**

Các thông tin thu thập, sau khi được tổng hợp và củng cố, sẽ được trình bày trong tài liệu đặc tả tổng thể dưới hai dạng:

- Mô tả các bài toán nghiệp vụ, các chức năng và tổ chức của cơ quan, các nhu cầu và mong muốn của người sử dụng một cách đầy đủ, nhưng ngắn gọn và mạch lạc, bằng một ngôn ngữ thông thường, gần gũi với mọi người.

- Minh họa và hệ thống hóa phân trình bày trên bằng một ngôn ngữ hình thức, thường là dưới dạng phiếu mô tả, danh sách và đồ họa.

Có nhiều phương pháp trình bày thông dụng khác nhau. Người ta thường dùng sơ đồ luân chuyển tài liệu vì một số ưu điểm của nó như sau:

- Đây là một dạng sơ đồ tổng hợp thông tin về tài liệu cũng như về chức năng.

- Nó được trình bày đúng theo thứ tự thời gian, công việc của từng đơn vị, có thể dùng làm “kim chỉ nam” trong lúc phỏng vấn người sử dụng.

- Sơ đồ này dễ hiểu, tương đối gần với người sử dụng, do đó có thể cùng họ làm chung các công tác xây dựng cũng như kiểm chứng.

Sơ đồ này thường được trình bày kèm theo một số tài liệu khác như:

- Danh sách các đơn vị hay vị trí làm việc: mã số và định nghĩa

- Phiếu mô tả các đơn vị hay vị trí làm việc: định nghĩa, các tài liệu vào và ra, các chức năng xử lý.

- Danh sách các tài liệu (in trên giấy hay màn hình, tệp in, điện thoại,...): mã số và định nghĩa.

- Danh sách các chức năng xử lý: mã số và định nghĩa.

Cách trình bày các loại sơ đồ và tài liệu này không đòi hỏi quy chuẩn chặt chẽ, và cũng khá đơn giản để không cần phải giải thích dài dòng.

### 3.2.2. Thiết lập mô hình dữ liệu

Thiết lập một mô hình khái niệm dữ liệu là liệt kê và định nghĩa chính xác những dữ liệu có liên quan đến các chức năng, hoạt động của tổ chức, sau đó nhóm chúng lại thành thực thể và quan hệ giữa các thực thể rồi dùng một số qui ước đã định trước để trình bày dưới dạng mô hình khái niệm.

#### 3.2.2.1. Xây dựng mô hình khái niệm dữ liệu bằng phương pháp *Blanpre*

Phương pháp Blanpre là phương pháp có độ tin cậy cao. Cơ sở của phương pháp này là đặc tả ràng buộc toàn vẹn bằng phụ thuộc hàm. Các bước thực hiện phương pháp Blanpre bao gồm các công việc sau đây.

##### 3.2.2.1.1. Kiểm kê dữ liệu

Danh sách này chủ yếu được rút tĩa từ những thông tin thu thập được trong giai đoạn khảo sát ban đầu: tài liệu thu thập được; nhu cầu, giải thích của người sử dụng. Danh sách này sẽ làm nền cho việc thiết lập một từ điển dữ liệu về sau.

Có thể phân biệt hai loại dữ liệu:

- Loại dữ liệu xuất hiện trực tiếp trên các tài liệu, màn hình, tệp in thu thập được.
- Loại dữ liệu không hề xuất hiện nhưng cần thiết để chứa các kết quả trung gian, các thông tin đang chờ được xử lý, hay để tính toán ra các dữ liệu thuộc loại thứ nhất.

Một công cụ thông dụng, hữu ích cho giai đoạn này là “Ma trận của Blanpre”, dùng để phân tích các tài liệu thu thập và liệt kê ra danh sách các dữ liệu. Trong ma trận này, ta trình bày mỗi cột là một tài liệu và mỗi hàng là một loại dữ liệu. Tại mỗi ô giao điểm, ta đánh dấu loại dữ liệu có xuất hiện trên tài liệu. Nên dùng hai loại dấu hiệu khác nhau để phân biệt loại dữ liệu trực tiếp (số 1 chẳng hạn) với loại được tính toán thành (số 2 chẳng hạn).

Khi xây dựng ma trận này, ta nên bắt đầu bằng những tài liệu cơ bản, quan trọng nhất và chỉ cần trình bày một loại tài liệu khi nó cho phép nhận dạng ít nhất một loại dữ liệu mới.

*Ví dụ:*

Hai chứng từ sau là cần thiết cho quản lý hoạt động kinh doanh:

**A. ĐƠN ĐẶT HÀNG**

Số:

Mã khách hàng:

Ngày:

Tên khách hàng:

Địa chỉ:

STT	Mã hàng	Tên hàng	Đơn vị	Số lượng

**B. PHIẾU GIAO HÀNG**

Số:

Ngày:

Mã khách hàng:

Nơi giao:

STT	Mã hàng	Đơn giá	Số lượng	Thành tiền

*Giả thiết rằng đơn giá chỉ phụ thuộc vào hàng*

TONGTIEN

*Bảng 3.7. Ma trận Blaupre*

Loại dữ liệu	Đơn đặt hàng	Phiếu giao hàng
1. Số đơn	1	
2. Ngày đặt	1	
3. Mã khách	1	1
4. Tên khách	1	
5. Địa chỉ	1	
6. STT	2	2
7. Mã hàng	1	1
8. Tên hàng	1	
9. Đơn vị	1	
10. SL đặt	1	
11. Số phiếu		1
12. Ngày giao		1

13. Nơi giao		1
14. Đơn giá giao (Giả thiết: chỉ phụ thuộc vào hàng)		1
15. SL giao		1
16. Thành tiền		2
17. Tổng tiền		2

*Ghi chú:*

1. Dữ liệu trực tiếp
2. Dữ liệu gián tiếp/trung gian (tính được thông qua các dữ liệu khác)

Từ danh sách này, người ta cần kiểm tra bằng một công tác thanh lọc như:

- Bỏ bớt các dữ liệu đồng nghĩa nhưng khác tên, chỉ giữ lại một

*Ví dụ:*

Mã số sản phẩm = Danh mục đặt hàng

- Phân biệt các dữ liệu cùng tên nhưng khác nghĩa và tách thành nhiều loại dữ liệu khác nhau.

*Ví dụ:*

- Giá bán của một cửa hiệu khác với giá bán của một công ty sản xuất.

- Nhập chung các loại dữ liệu luôn xuất hiện đồng thời với nhau trên mọi loại tài liệu thành một dữ liệu sơ cấp.

*Ví dụ:*

Số nhà và tên đường; ngày, tháng và năm sinh.

- Loại bỏ những loại dữ liệu có thể xác định được một cách duy nhất từ các dữ liệu khác, hoặc bằng công thức tính toán, do các quy luật của tổ chức.

*Ví dụ:*

- Tổng giá đơn đặt hàng = Số lượng giá\* đơn giá

- Giả sử do quy luật tổ chức, mọi đề nghị mua hàng phải được giải quyết nội trong ngày, ta suy ra: Ngày đề nghị mua hàng = Ngày đặt hàng.

Sau khi chính xác hoá dữ liệu (loại đồng danh, đồng nghĩa, thêm bớt thuộc tính, loại bỏ dữ liệu gián tiếp,...), ta có:

*Bảng 3.8. Ma trận Blanpre rút gọn*

Loại dữ liệu	Đơn đặt hàng	Phiếu giao
Số đơn	1	
Ngày đặt	1	
Mã khách	1	1
Tên khách	1	
Địa chỉ	1	
Mã hàng	1	1
Tên hàng	1	
Đơn vị	1	
SL đặt	1	
Số phiếu		1
Ngày giao		1
Nơi giao		1
Đơn giá giao		1
SL giao		1

### 3.2.2.1.2. Xác định các phụ thuộc hàm giữa các dữ liệu

Từ danh sách các loại dữ liệu đã thanh lọc của hệ thống thông tin đặt được qua giai đoạn trên, ta phải xác định tất cả các phụ thuộc hàm hiện hữu giữa chúng.

*Cụ thể, ta phải tự đặt câu hỏi:*

*Mỗi giá trị của một loại dữ liệu A có tương ứng với một giá trị duy nhất của loại dữ liệu B không?*

Nếu “có” thì B phụ thuộc hàm vào A:  $A \rightarrow B$

Ngoài các phụ thuộc hàm có vế trái A là một loại dữ liệu sơ cấp (gọi là phụ thuộc hàm sơ cấp), tương đối dễ xác định, ta còn phải nhận

diện cả các hàm trong đó về trái A là một tập hợp của nhiều loại dữ liệu (gọi là phụ thuộc hàm đa phần).

*Ta tiếp tục đặt câu hỏi:*

*Cần ấn định giá trị của những loại dữ liệu nào để có thể suy ra một giá trị duy nhất của loại dữ liệu B?*

Các hàm phụ thuộc sẽ được trình bày dưới dạng một bảng các phụ thuộc hàm như sau:

*Bảng 3.9. Ma trận phụ thuộc hàm Blanpre*

Loại dữ liệu	Phụ thuộc hàm sơ cấp	Phụ thuộc hàm đa phần
<b>Số đơn</b>		
Ngày đặt	↓	Nhận
<b>Mã khách</b>	↓	Có
Tên khách	↓	
Địa chỉ	↓	
<b>Mã hàng</b>	↓	
Tên hàng	↓	
Đơn vị	↓	
SL đặt		↓ ↓ ↓ DATCACLOAI
<b>Số phiếu</b>	↓	
Ngày giao	↓	
Nơi giao	↓	
Đơn giá giao		
SL giao		GIAOCAC LOAI ↓ ↓ ↓

### 3.2.2.1.3. Xây dựng mô hình khái niệm dữ liệu

Giai đoạn này bao gồm 5 bước:

- Xác định tập hợp các khóa chính
- Nhận diện các thực thể
- Nhận diện các quan hệ
- Phân bố hết các thuộc tính
- Vẽ mô hình khái niệm dữ liệu

### **1. Xác định tập hợp các khóa chính**

Tập hợp K của những khóa chính là tập hợp tất cả những loại dữ liệu có đóng vai trò nguồn (thuộc về trái) trong ít nhất một phụ thuộc hàm.

Trong ví dụ trên, ta có:

$$K = \{\text{Số đơn, mã khách, mã hàng, số phiếu}\}$$

### **2. Nhận diện các thực thể**

Mỗi phần tử của tập hợp K sẽ là khóa chính của một thực thể.

Trong ví dụ trên, ta nhận ra được 4 thực thể:

ĐƠN HÀNG (Số đơn,....)

KHÁCH (Mã khách,....)

HÀNG (Mã hàng,...)

PHIẾU GIAO (Số phiếu,.....)

### **3. Nhận diện các quan hệ**

Có hai trường hợp:

A. Nếu gốc của một phụ thuộc hàm bao gồm ít nhất 2 phần tử thuộc tập hợp K thì nó tương ứng với một quan hệ N-N giữa các thực thể có khóa chính là các phần tử này.

Trong ví dụ trên, ta nhận ra được 2 quan hệ:

ĐẶT CÁC LOẠI (số lượng đặt)

GIAO CÁC LOẠI (số lượng giao)

B. Sự hiện hữu của một phụ thuộc hàm giữa hai phần tử của tập hợp K xác định một quan hệ nhị nguyên kiểu 1-N giữa hai thực thể có khóa chính là các phần tử này.

Trong ví dụ trên, ta nhận ra được 2 quan hệ:

CÓ

NHẬN

#### 4. Phân bổ các thuộc tính còn lại

Tất cả các loại dữ liệu không tập hợp K đều là những thuộc tính thông thường và có thể được phân bổ một cách dễ dàng vào các thực thể hay quan hệ, tùy theo các hàm phụ thuộc.

Phân bổ các thuộc tính còn lại cho các thực thể ta thấy vừa hết:

ĐƠN HÀNG (Số đơn, ngày đặt)

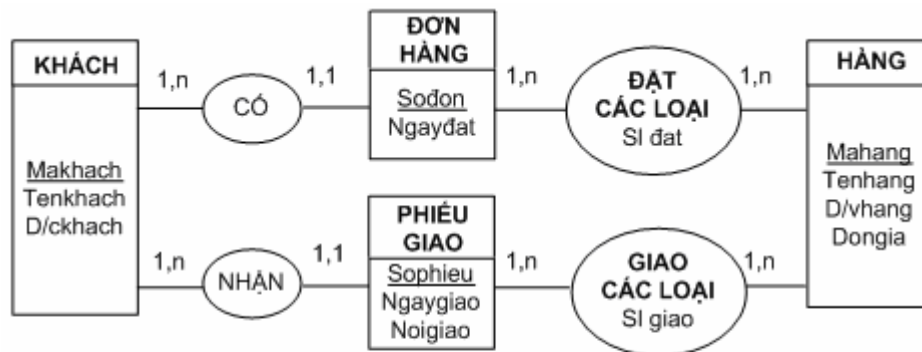
KHÁCH (Mã khách, tên khách, Đ/Ckhách)

HÀNG (Mã hàng, Tên hàng, Mô tả hàng, Đ/Vhàng, Đơn giá)

PHIẾU GIAO (Số phiếu, ngày giao, nơi giao)

#### 5. Vẽ mô hình khái niệm dữ liệu

Từ các thực thể và quan hệ đã nhận diện, ta có thể vẽ lên một mô hình khái niệm dữ liệu như sau:



Hình chữ nhật biểu diễn thực thể

Hình biểu diễn mối quan hệ

Thuộc tính khóa được gạch chân

Hình 3.4. Mô hình khái niệm dữ liệu bài toán đặt và giao hàng

#### 3.2.2.2. Xây dựng mô hình khái niệm dữ liệu bằng phương pháp mô hình

Cơ sở của phương pháp này là các quy tắc quản lý. Các bước thực hiện của phương pháp mô hình bao gồm các công việc sau:



1. Xác định danh sách các thuộc tính cần quản lý
2. Xác định các thực thể
3. Xác định các mối quan hệ giữa các thực thể
4. Thiết lập mô hình khái niệm dữ liệu

*Ví dụ:*

Một cơ sở bán hàng sử dụng hai loại chứng từ sau để theo dõi hoạt động kinh doanh của mình:

**A. ĐƠN ĐẶT HÀNG**

Số hoá đơn:

Ngày đặt:

Người đặt hàng:

Địa chỉ:

STT	Tên hàng	Mô tả hàng	Đơn vị tính	Số lượng (đặt)

**B. PHIẾU GIAO HÀNG**

Số phiếu:

Ngày giao:

Tên khách hàng:

Nơi giao hàng:

Địa chỉ:

STT	Tên hàng	Đơn vị tính	Đơn giá	Số lượng (giao)	Thành tiền

**TONGTIEN:**

Cần thiết kế CSDL loại quan hệ (bao gồm các file dữ liệu và sơ đồ mô hình dữ liệu) từ các tài liệu trên để quản lý hoạt động bán hàng.

*Giả thiết:* Đơn giá giao không chỉ phụ thuộc vào mã hàng (có thể phụ thuộc vào khách, số lượng, thời gian)

**Thủ tục tiến hành:**

1. Liệt kê, chính xác hoá thông tin

Bảng 3.10. Danh sách thuộc tính cần quản lý

Thực thể -Thuộc tính	Ghi chú
ĐƠN ĐẶT HÀNG	DONHANG
<u>Số đơn hàng</u>	Định danh
Người đặt hàng $\Rightarrow$ tên khách hàng	Thay bằng tên khách hàng (0)
Mã khách hàng	Thêm mới (1)
Địa chỉ khách	
Ngày đặt hàng	
* Số thứ tự (2)	Không cần
* <u>Mã hàng</u>	Thêm mới, Định danh (1)
* Tên hàng	
* Mô tả hàng	
* Đơn vị tính	
* Số lượng đặt	
PHIẾU GIAO HÀNG	PHIEUGIAO
<u>Số phiếu giao</u>	Định danh
Tên khách hàng $\Rightarrow$ mã khách	Thay bằng mã khách (3)
Nơi giao hàng	
Ngày giao	
* Số thứ tự (2)	Không cần
* Tên hàng $\rightarrow$ <u>mã hàng</u>	Thay bằng mã hàng (3)
* Đơn vị tính	Không cần (4)
* Đơn giá hàng giao	
* Số lượng giao	
* Thành tiền	Không cần (5)
<b>Tổng cộng</b>	Không cần

*Chú thích:*

- (0) “người đặt hàng” ở ĐƠN ĐẶT HÀNG và “Tên khách hàng” ở PHIẾU GIAO HÀNG chỉ cùng một đối tượng nên ta thống nhất chọn tên gọi cho đối tượng này là: “Tên khách hàng”.
- (1) “Tên khách hàng” là thuộc tính tên gọi của thực thể “khách hàng”, không xác định mỗi khách hàng cụ thể vì khách có thể trùng tên.

Vì vậy phải thêm thuộc tính “Mã khách hàng” làm thuộc tính định danh cho “khách hàng”. Cũng tương tự như vậy cần thêm “Mã hàng” làm thuộc tính định danh cho “hàng”.

- (2) Số thứ tự để đánh số mỗi dòng hoá đơn chỉ có ý nghĩa trong hoá đơn này, không cho thông tin gì khác, không cần lưu trữ.
  - (3) Tên khách hàng, tên hàng là thuộc tính tên gọi đã có ở trên, cần thay nó bằng thuộc tính định danh tương ứng.
  - (4) Các thuộc tính này là thuộc tính mô tả, đã gặp ở trên nên bỏ đi.
  - (5) Thuộc tính *thành tiền* = (*đơn giá giao*) x (*số lượng giao*). Hai thuộc tính này đã được chọn ở trên nên có thể tính được “*thành tiền*”, do đó không cần lưu giữ thuộc tính này.
- (\*) Chỉ các thuộc tính lặp/đa trị (không sơ cấp).

## 2. Xác định các thực thể và các thuộc tính

Từ hai hồ sơ, chúng ta nhận biết được các thực thể cũng như các thuộc tính.

Các thực thể:

- E1-KHÁCH

*#mã khách*

*Tên khách*

*Địa chỉ khách*

- E2-HÀNG

*#mã hàng*

*Tên hàng*

*Mô tả hàng*

*Đơn vị hàng*

## E4-ĐƠN ĐẶT

#Số đơn đặt

Ngày đặt hàng

## - E5-PHIẾU GIAO

# Số phiếu giao

Nơi giao

Ngày giao

Chú ý:

- Thực thể KHACH có hai thuộc tính: *tên khách, địa chỉ*. Tên khách không thể dùng làm định danh cho thực thể nên cần thêm định danh cho nó: mã khách

- Thực thể HANG có 3 thuộc tính: *tên hàng, đơn vị, mô tả hàng*. Cũng như thực thể KHACH ta cần thêm định danh: mã hàng cho thực thể này.

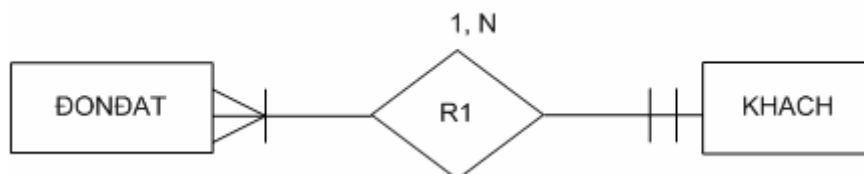
## 3. Xác định các mối quan hệ và các thuộc tính của quan hệ

Nhận xét: còn 3 thuộc tính sau chưa ghép vào thực thể nào cả là: Đơn giá hàng giao, Số lượng hàng giao và Số lượng đặt.

Trong đơn đặt hàng, chúng ta xác định được quy tắc quản lý sau:

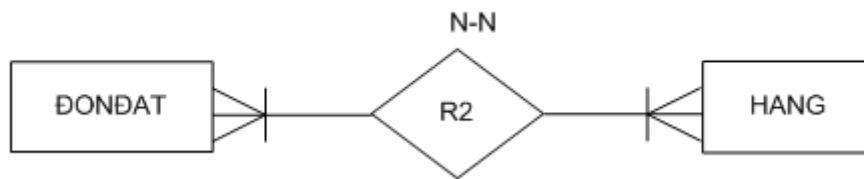
- Trong mỗi đơn đặt hàng người ta chỉ viết cho một khách. Ngược lại một khách có thể có nhiều đơn đặt hàng.

Vậy ta có đặc tả:



- Trong mỗi đơn đặt hàng người ta có thể viết nhiều loại hàng (mỗi loại hàng trên một dòng hàng). Ngược lại, mỗi loại hàng có thể có mặt trong nhiều đơn đặt hàng.

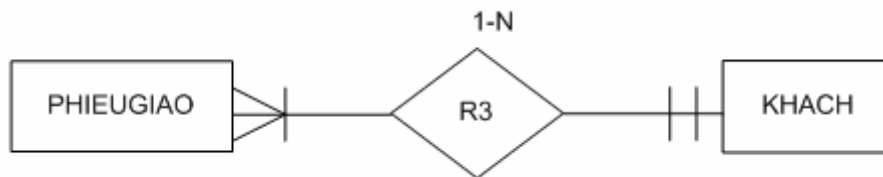
Vậy ta có đặc tả:



Trong phiếu giao hàng, chúng ta cũng xác định được quy tắc quản lý sau:

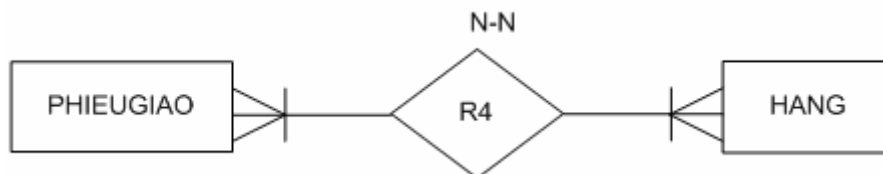
- Trong mỗi phiếu giao hàng người ta chỉ viết cho một khách. Ngược lại một khách có thể có nhiều phiếu giao hàng.

Vậy ta có đặc tả:



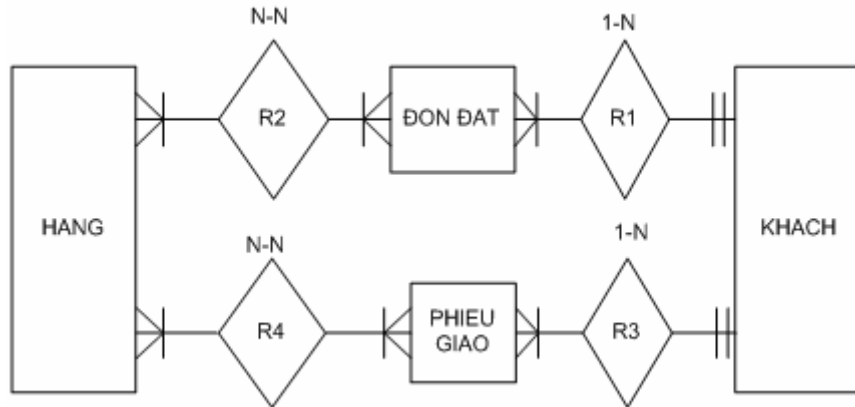
- Trong mỗi phiếu giao hàng người ta có thể viết nhiều loại hàng (mỗi loại hàng trên một dòng hàng). Ngược lại, mỗi loại hàng có thể có mặt trong nhiều phiếu giao hàng.

Vậy ta có đặc tả:



#### 4. Vẽ mô hình khái niệm dữ liệu

Ghép nối tất cả 4 đặc tả trên, chúng ta có được mô hình khái niệm dữ liệu cho bài toán:



Hình 3.5. Mô hình khái niệm dữ liệu

### 3.2.2.3. Xây dựng mô hình khái niệm dữ liệu bằng phương pháp trực giác

Phương pháp phân tích hệ thống nêu trên là một công cụ hữu hiệu và chuẩn xác để xây dựng phần lớn các mô hình dữ liệu. Nhưng nếu áp dụng hoàn toàn trong một hệ thống tin cỡ lớn sẽ đòi hỏi nhiều thời gian và công sức. Trong thực tế, các thiết kế viên chuyên nghiệp - sau khi đã nhận thức được vấn đề khảo sát - thường chọn cách xây dựng trực tiếp một mô hình sơ khởi rồi đi thẳng vào giai đoạn sau để kiểm soát, bổ túc và chuẩn hóa mô hình.

Phương pháp trực giác này có ưu điểm là ít tốn thời gian và đôi khi tạo ra mô hình đơn giản và thực tế hơn. Nhưng ngược lại, nó cũng chứa nhiều rủi ro hơn.

Ví dụ 1: Cho phiếu nhập kho có dạng:

#### PHIẾU NHẬP KHO

(Theo đơn đặt hàng số:.....)

Số PN:

Ngày nhập:

Mã Nhà cung cấp:

Mã kho:

Tên NCC:

Địa chỉ kho:

Địa chỉ NCC:

STT	Tên vật liệu	Mã vật liệu	Đơn vị tính	Số lượng	Nơi sản xuất

Thủ kho

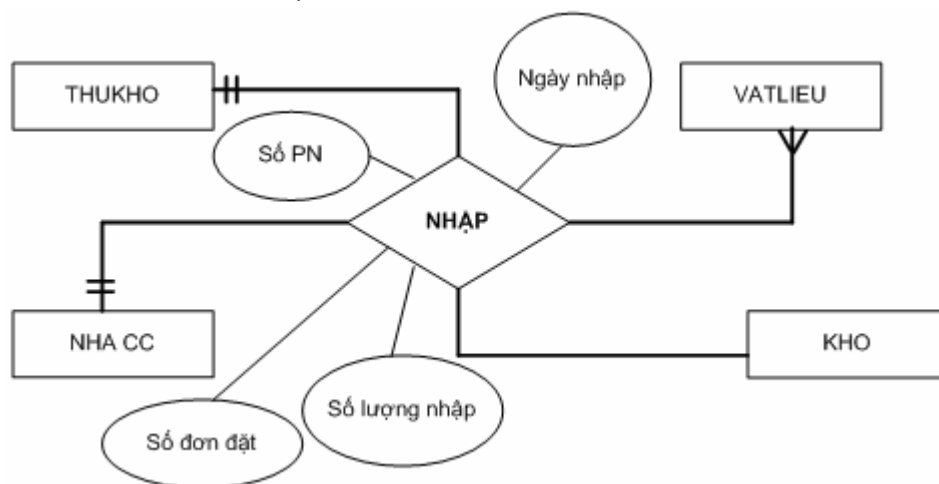
Khi nghiên cứu PHIEUNHAPKHO chúng ta có nhận xét nó biểu hiện mối quan hệ của 4 thực thể NHACC, KHO, VATLIEU và THUKHO, vì vậy chúng ta xác nhận nó như một mối quan hệ.

Khi xác định mối quan hệ NHAP và các thuộc tính của nó, chúng ta đặt các câu hỏi khi nhìn vào phiếu nhập:

Bảng 3.11. Bảng phân tích quan hệ NHAP

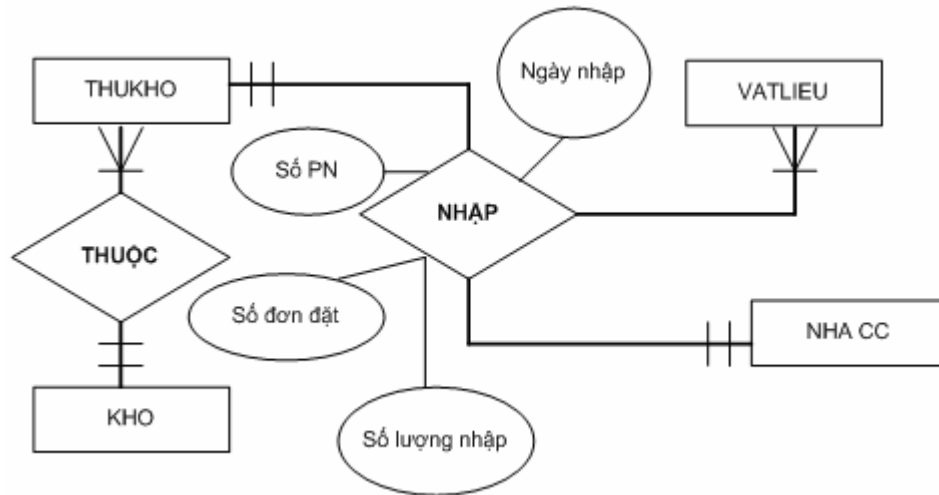
Câu hỏi cho động từ “nhập”	Trả lời	
	Thực thể	Thuộc tính
Ai nhập?	THUKHO	
Nhập cái gì?	VATLIEU	
Từ Ai?	NHACUNGCAP	
Ở đâu?	KHO	
Lý do nhập		Đơn đặt (theo đơn đặt số)
Bằng cách nào?		Phiếu nhập (Số PN)
Khi nào?		Ngày nhập
Bao nhiêu?		Số lượng nhập
Nguồn gốc vật liệu?		Nơi sản xuất

Ta có đặc tả mối quan hệ này như sau: THUKHO nhận VATLIEU từ NHACUNGCAP tại KHO:



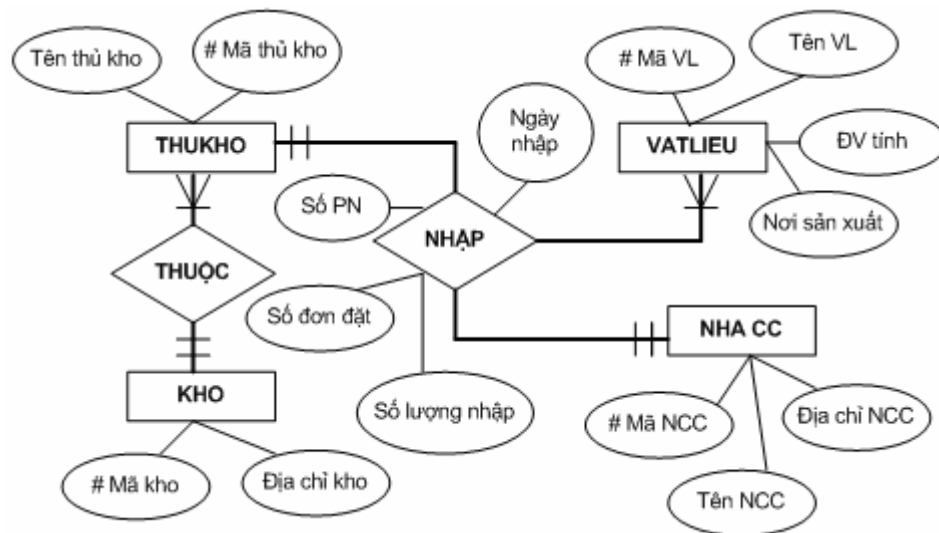
Hình 3.6. Mối quan hệ bậc 4 dựa trên PHIEUNHAPKHO

Giả sử rằng: mỗi kho có thể có nhiều thủ kho, nhưng mỗi thủ kho chỉ làm việc trong 1 kho. Như vậy giữa KHO và THU KHO có mối quan hệ 1-N. Vì vậy, chúng ta chính xác hoá quan hệ này bằng cách tách khỏi nó mối quan hệ sở hữu:



Hình 3.7. Đặc tả mối quan hệ

Đặc tả này cũng chính là mô hình khái niệm dữ liệu của bài toán:



Hình 3.8. Mô hình khái niệm dữ liệu



Ví dụ 2: Chúng ta sử dụng lại ví dụ trong phương pháp mô hình và cũng thực hiện lần lượt theo các bước sau:

a. Liệt kê, chính xác hoá thông tin: (giống như phương pháp mô hình)

b. Xác định các thực thể và các thuộc tính

Từ hai hồ sơ, qua các tên gọi (tên khách hàng, người đặt hàng, tên hàng) ta nhận biết được các thực thể cũng như là các thuộc tính là:

- Các thực thể: KHÁCH, HÀNG

- Các thuộc tính trích từ các hồ sơ trên theo thứ tự sau khi đã chất lọc (như cột đầu trong bảng trên sau khi đã loại đi những thuộc tính lặp lại) là:

Số đơn đặt

Tên khách

Địa chỉ khách

Ngày đặt hàng

\* Tên hàng

\* Mô tả hàng

\* Đơn vị hàng

\* Số lượng đặt

Số phiếu giao

Nơi giao

Ngày giao

\* Đơn giá hàng giao

\* Số lượng hàng giao

Tổng tiền của phiếu giao (bỏ vì có thể suy ra từ các thuộc tính khác)

c. Xác định các thuộc tính của quan hệ

- Thực thể KHÁCH có hai thuộc tính: *tên khách*, *địa chỉ*. *Tên khách* không thể dùng làm định danh cho thực thể nên cần thêm định danh cho nó: *mã khách*

- Thực thể *HANG* có 3 thuộc tính: *tên hàng*, *đơn vị*, *mô tả hàng*.  
Cũng như thực thể *KHACH* ta cần thêm định danh: *mã hàng* cho thực thể này.

Các thuộc tính còn lại:

*Số đơn đặt*

*Ngày đặt*

\* *Số lượng đặt*

*Số phiếu giao*

*Nơi giao*

*Ngày giao*

\* *Đơn giá hàng giao*

\* *Số lượng hàng giao*

*d. Xác định các mối quan hệ của hai thực thể trên*

- Mối quan hệ 1:

*KHÁCH - ĐẶT - HÀNG* (bằng *Đơn hàng*) nên có các thuộc tính liên quan đến quan hệ này là:

+ *Số đơn đặt*

+ *Ngày đặt*

+ *Số lượng hàng đặt* \*

- Mối quan hệ 2:

*HÀNG - (được) GIAO (cho) - KHÁCH* (bằng *Phiếu giao hàng*) nên có các thuộc tính liên quan đến mối quan hệ này là:

+ *Số phiếu giao*

+ *Nơi giao*

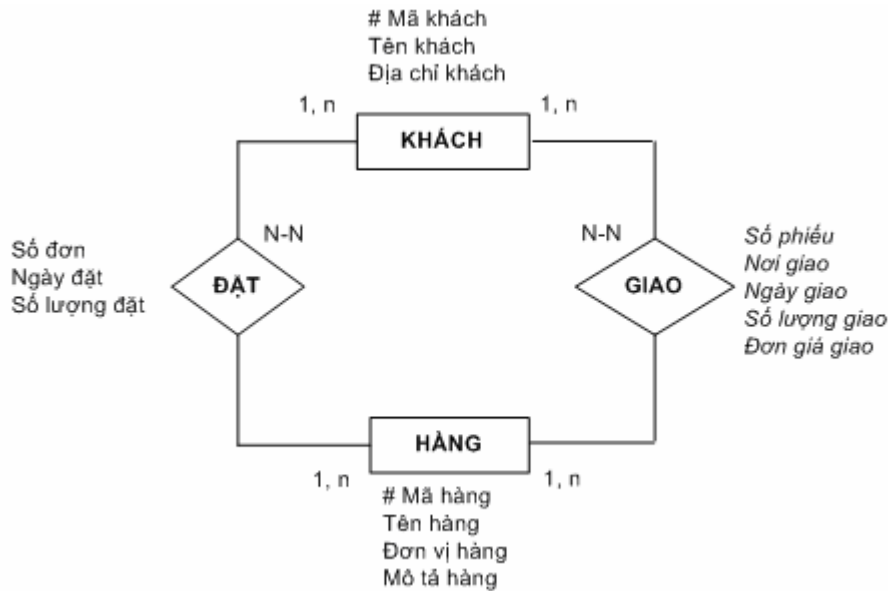
+ *Ngày giao*

+ *Số lượng hàng giao* \*

+ *Đơn giá hàng giao* \*

Như vậy, các thông tin từ hai chứng từ đã được sử dụng hết.

e. Xây dựng mô hình khái niệm dữ liệu



Hình 3.9. Mô hình khái niệm dữ liệu của bài toán đặt và giao hàng

### 3.2.3. Kiểm soát và chuẩn hóa mô hình

Để đơn giản hóa và đồng thời đảm bảo tính nhất quán của mô hình dữ liệu, ta cần kiểm soát lại mô hình vừa xây dựng bằng một số qui tắc thực tiễn sau đây:

#### 3.2.3.1. Chuẩn hóa mô hình

Mọi thực thể, quan hệ của mô hình đều nên được chuẩn hóa theo các dạng 1FN, 2FN, 3FN và BCFN, ngoại trừ trường hợp có yêu cầu đặc biệt khác của hệ thông tin.

Việc chuẩn hóa toàn bộ theo các dạng 3FN và BCFN không bắt buộc, còn các dạng 1FN và 2FN phải luôn được tuân thủ.

Ví dụ:

Mối quan hệ xác định trên hình 3.8 trong mục xây dựng mô hình dữ liệu bằng trực giác sẽ sinh thành một lược đồ quan hệ chưa đạt chuẩn 1:

PHIEUNHAP (Số PN, ngày nhập, Mã Nhà cung cấp, Số đơn đặt hàng, Mã vật liệu\*, Số lượng\*, mã thủ kho)

Để chuẩn hoá mô hình, chúng ta phải tách PHIEUNHAP thành các lược đồ quan hệ đạt 1NF:

PHIEUNHAP (Số PN, ngày nhập, Mã Nhà cung cấp, Số đơn đặt, mã thủ kho)

DONGPHIEUNHAP (Số PN, Mã vật liệu, Số lượng)

Hai quan hệ này đương nhiên cũng đạt 2NF, 3NF.

### 3.2.3.2. Tạo thêm một thực thể

- Việc tạo thêm một thực thể là cần thiết khi có ít nhất một quan hệ hiện hữu tham chiếu tới nó.

- Việc tạo thêm một thực thể là hợp lý khi:

a. Thuộc tính sẽ được chọn làm khóa chính của thực thể này là một loại dữ liệu thông dụng trong hoạt động của tổ chức

b. Ngoài khóa chính này, quan hệ còn có chứa những thuộc tính khác.

### 3.2.3.3. Biến một quan hệ thành thực thể

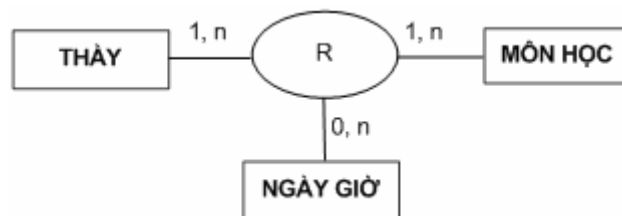
Một quan hệ có kích thước lớn hơn 3 nên được biến thành những thực thể đơn giản hơn.

Có thể biến một quan hệ thành thực thể khi hội đủ các điều kiện sau:

- Quan hệ này có một khóa chính độc lập.

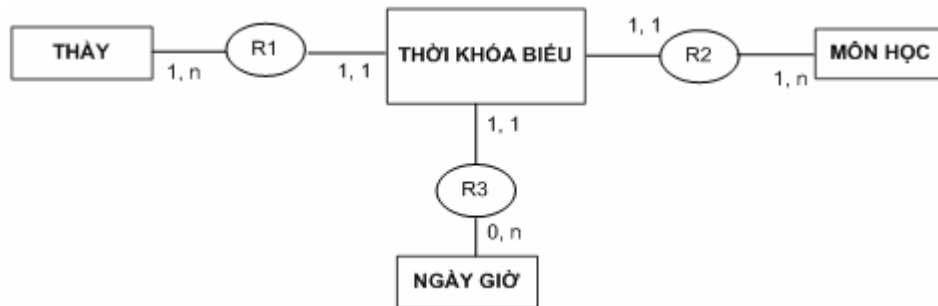
- Quan hệ này tương ứng với một khái niệm quen thuộc, thông dụng trong hoạt động của tổ chức.

Ví dụ:



Hình 3.10. Đặc tả mối quan hệ thời khoá biểu

Quan hệ R được biến thành thực thể “Thời khóa biểu”

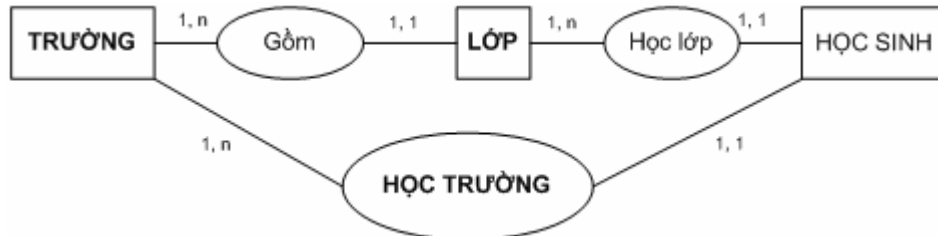


Hình 3.11. Đặc tả thực thể thời khóa biểu

### 3.2.3.4. Xóa một quan hệ

Một quan hệ 1-N phải được loại bỏ khỏi mô hình dữ liệu nếu nó là tổng hợp của hai hay nhiều quan hệ 1-N.

Ví dụ: Quan hệ “HỌC TRƯỜNG” trong ví dụ sau đây có thể, và nên, được loại bỏ.



Hình 3.12. Đặc tả mối quan hệ “học trường”

### 3.2.3.5. Phân tách một quan hệ phức tạp

Xét một quan hệ có kích thước lớn hơn hoặc bằng 3. Quan hệ có thể phân tách thành nhiều quan hệ khác với kích thước nhỏ hơn mà không mất thông tin nếu tồn tại ít nhất một hàm phụ thuộc giữa các thực thể cấu thành quan hệ.

#### 3.2.3.5.1. Trường hợp phụ thuộc hàm ẩn

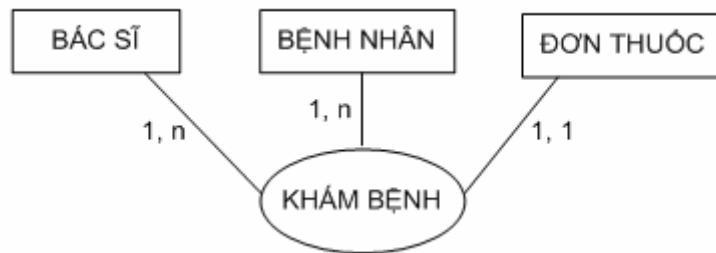
Trong trường hợp này, một trong các bản số của quan hệ bằng (1.1) hoặc (0.1) điều này chứng tỏ sự tồn tại của một số phụ thuộc hàm ẩn.

Trong ví dụ sau đây, bản số (1,1) giữa quan hệ “Khám bệnh” với thực thể “Đơn thuốc” đã chỉ ra được các phụ thuộc hàm:

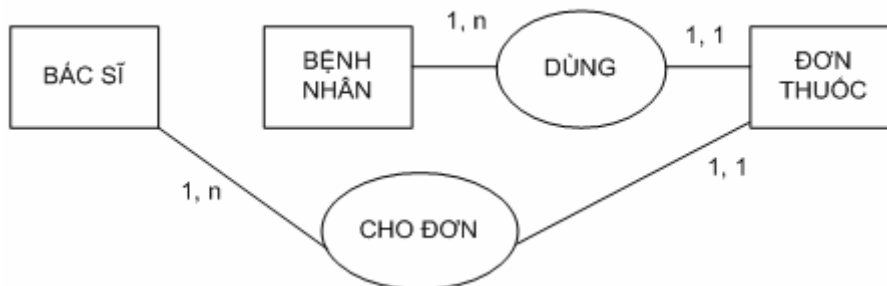
Đơn thuốc ---> Bệnh nhân

và Đơn thuốc ---> Bác sĩ

Sơ đồ thứ hai tương đương với sơ đồ thứ nhất, nhưng đơn giản hơn:



Hình 3.13. Đặc tả mối quan hệ “khám bệnh”



Hình 3.14. Đặc tả mối quan hệ “cho đơn” và “dùng thuốc”

### 3.2.3.5.2. Trường hợp phụ thuộc hàm hiện

Cho một quan hệ R giữa 3 thực thể A, B và C. Nếu tồn tại một hàm phụ thuộc  $A \rightarrow B$  thì R có thể được phân thành hai quan hệ giữa A với B và giữa A với C.

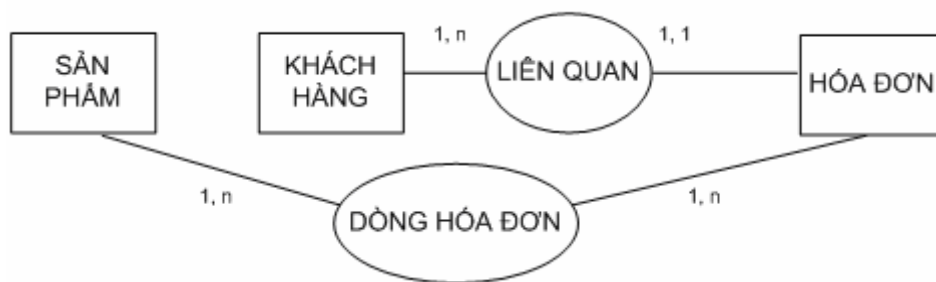
Trong ví dụ sau đây, hàm phụ thuộc Hóa đơn ---> Khách hàng (mỗi hóa đơn chỉ liên quan đến một khách hàng duy nhất) cho phép ta đưa vào một quan hệ mới để diễn tả sự phụ thuộc này và đơn giản hóa mô hình.

Ví dụ:

Sơ đồ thứ 2 tương đương với sơ đồ thứ nhất, nhưng đơn giản hơn:



Hình 3.15. Đặc tả mối quan hệ “dòng hoá đơn”



Hình 3.16. Đặc tả mối quan hệ “dòng hoá đơn” và “liên quan”

*Nhận xét:* Phương pháp xây dựng mô hình bằng trực giác thường cho phép ta đi thẳng đến sơ đồ thứ hai vì sơ đồ này gần với thực tế hoạt động của tổ chức hơn.

### 3.3. THIẾT KẾ CSDL LOGIC

Thiết kế CSDL logic là vô cùng quan trọng trong khâu thiết kế CSDL. Kết quả thiết kế CSDL logic cho cả người phân tích thiết kế hệ thống, NSD và người quản lý hình dung được CSDL của HTTT trong một tổ chức bao gồm bao nhiêu tệp, tên từng tệp, danh sách các trường mỗi tệp, nhóm thuộc tính khoá chính trong mỗi tệp cũng như tổng thể các mối quan hệ logic giữa các tệp (kết nối thông qua khoá ngoại). Các kỹ thuật được sử dụng ở đây thể hiện tính công nghệ rõ nét.

### 3.3.1. Khuôn cảnh thiết kế cơ sở dữ liệu logic

Vấn đề đặt ra là thế nào là một thiết kế cơ sở dữ liệu tốt?

Để làm rõ vấn đề này chúng ta xét ví dụ sau.

Cho một lược đồ:

THÔNG TIN VỀ NHÀ CUNG CẤP (tên NCC, địa chỉ NCC, mặt hàng, giá)

Lược đồ này chứa một số các thông tin về nhà cung cấp. Nhiều vấn đề có thể nảy sinh trong đó như:

1. Dư thừa (redundancy). Địa chỉ của nhà cung cấp được lập lại mỗi lần cho mỗi mặt hàng được cung cấp.

2. Mâu thuẫn tiềm ẩn (potential inconsistency) hay bất thường khi cập nhật. Do hậu quả của dư thừa, chúng ta có thể cập nhật địa chỉ của một nhà cung cấp trong một bản ghi nhưng vẫn để lại địa chỉ cũ trong một bản ghi khác. Vì vậy chúng ta có thể không có một địa chỉ duy nhất đối với mỗi nhà cung cấp.

3. Bất thường khi chèn (insertion anomaly). Chúng ta không thể biết địa chỉ một nhà cung cấp nếu hiện tại họ chưa cung cấp ít một mặt hàng. Chúng ta có thể đặt giá trị null trong các thành phần mặt hàng và giá của một bản ghi cho người đó, nhưng khi chúng ta nhập mặt hàng cho nhà cung cấp đó, chúng ta có nhớ xoá đi bản ghi mang giá trị null hay không? Điều tệ hại là mặt hàng và tên NCC cùng tạo ra một khoá cho quan hệ đó, và có lẽ không thể tìm ra các bản ghi nhờ chỉ mục sơ cấp được, nếu có những giá trị null trong trường khoá mặt hàng.

4. Bất thường khi xoá (deletion anomaly). Ngược lại với vấn đề (3) là vấn đề chúng ta có thể xoá tất cả các mặt hàng được cung cấp bởi một người, vô ý làm mất dấu vết để tìm ra địa chỉ của nhà cung cấp này.

Trong ví dụ này, tất cả các vấn đề nảy sinh trên sẽ được giải quyết triệt để khi thay nó bằng hai lược đồ quan hệ mới:

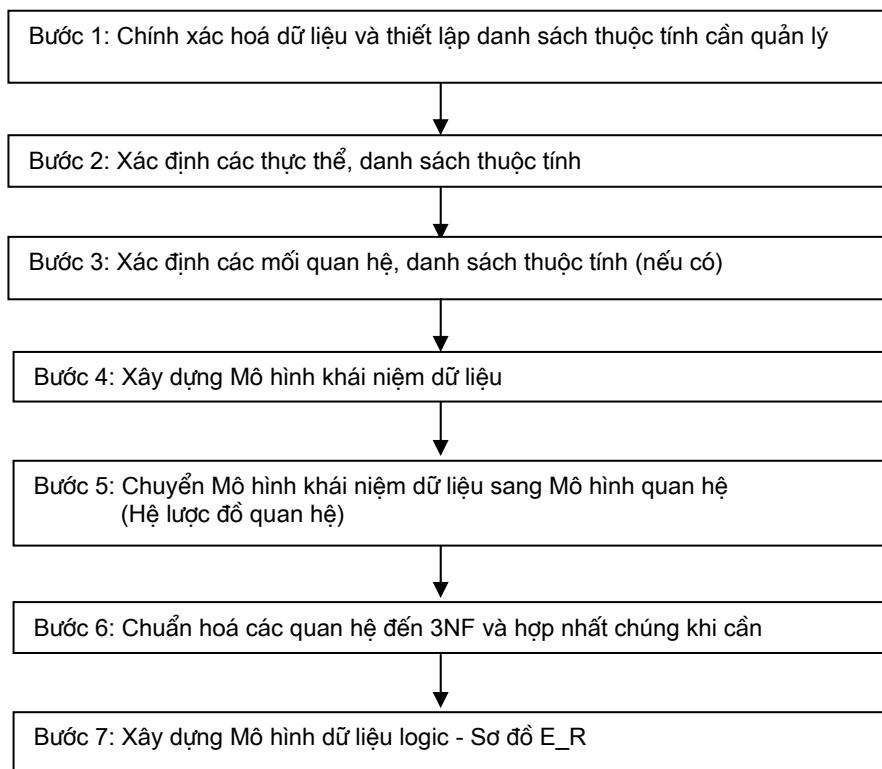
NHÀCÇ (tên NCC, địa chỉ NCC)

GIÁ\_NCC (tên NCC, mặt hàng, giá)



Ở đây quan hệ NHÀCC cung cấp địa chỉ của mỗi NCC đúng một lần, do vậy không có dư thừa. Ngoài ra, chúng ta cũng có thể nhập địa chỉ của nhà cung cấp dù hiện tại họ chưa cung cấp một loại mặt hàng nào. Như vậy tư tưởng nảy sinh tự nhiên là làm thế nào để có một CSDL tốt?

Để đảm bảo quá trình thiết kế dẫn đến một CSDL tốt (theo nghĩa tránh được những nhược điểm trên) người ta thường thực hiện đầy đủ các bước theo một trình tự công nghệ được trình bày dưới đây.



Hình 3.17. Sơ đồ khuôn cảnh chung các bước thiết kế CSDL logic

Đối với phương pháp Blaupre, bước 2 và bước 3 dựa vào ma trận phụ thuộc hàm. Đối với phương pháp trực giác, việc xác định các mối quan hệ không dựa trên ràng buộc phụ thuộc hàm, hay quy tắc quản lý mà bằng trực giác. Trong đó, các hồ sơ dữ liệu thường được xét tới như một mối quan hệ n ngôi.

### 3.3.2. Các kỹ thuật sử dụng trong thiết kế cơ sở dữ liệu logic

Phần trên, chúng ta đã biết các kỹ thuật xây dựng mô hình khái niệm dữ liệu. Bây giờ chúng ta sẽ tìm hiểu các kỹ thuật còn lại sẽ sử dụng khi thiết kế CSDL.

#### 3.3.2.1. Kỹ thuật xác định các thực thể

Một thực thể được xác định nếu xác định được 3 thành phần: tên của thực thể, danh sách thuộc tính (ít nhất là một) và nhóm thuộc tính định danh (ít nhất là một thuộc tính).

Có nhiều cách xác định thực thể. Chúng ta xem xét một ví dụ sau đây.

Cho một hoá đơn điển hình:

#### HOÁ ĐƠN BÁN HÀNG

Số:

Ngày:

Mã khách hàng:

Tên khách hàng:

Địa chỉ:

STT	Mã hàng	Tên hàng	Đơn vị	Đơn giá	Số lượng	Thành tiền

Tổng tiền

1. Trường hợp 1: Coi cả HOADON là một thực thể, ta có:

HOADON

#Số

#Mã hàng\* (2 thuộc tính {số, mã hàng} hợp thành nhóm thuộc tính định danh của thực thể)

Ngày

Mã khách hàng

Tên khách hàng

Địa chỉ

Tên hàng\*

Đơn vị tính\*

Đơn giá\*

Số lượng\*

Thành tiền\*

Tổng tiền (các thuộc tính có dấu \* là thuộc tính lặp/đa trị)

2. Trường hợp 2: Xác định 2 thực thể: một nhóm thuộc tính đơn và 1 nhóm thuộc tính lặp.

E1-HOADON

E2- HANG

#Số

Ngày

Mã khách hàng

Tên khách hàng

Địa chỉ

Tổng tiền

#Mã hàng

Tên hàng

Đơn vị tính

3. Trường hợp 3: Xác định 3 thực thể: hai nhóm thuộc tính đơn và 1 nhóm thuộc tính lặp.

E1-HOADON

E2- KHACH

E3-HANG

#Số

Ngày

Tổng tiền

#Mã khách hàng

Tên khách hàng

Địa chỉ

#Mã hàng

Tên hàng

Đơn vị tính

Chú ý:

- Mỗi thuộc tính chỉ phân cho 1 thực thể
- Có thể có thuộc tính không phân được cho một thực thể nào cả

### 3.3.2.2. Kỹ thuật đặc tả

Chúng ta đã biết kỹ thuật đặc tả mối quan hệ giữa 2 thực thể dựa vào mô tả bằng ngôn ngữ tự nhiên. Ngoài cách này chúng ta còn có thể đặc tả mối quan hệ giữa 2 thực thể dựa trên những kỹ thuật sau đây:

1. Dựa vào quy tắc quản lý hoặc những quy tắc toàn vẹn

Ví dụ: Cho các thực thể và thuộc tính sau:

VẬT TƯ (mã vt, tên vt, ĐV, đơn giá),

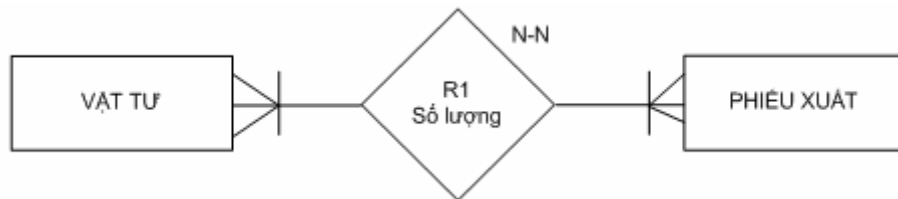
PHIẾU XUẤT (số phiếu, ngày, mã kho, mã khách),

KHÁCH (mã khách, tên khách, địa chỉ, E\_mail),

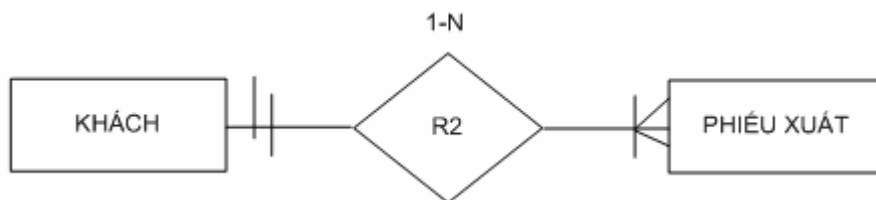
KHO (mã kho, tên kho)

Dựa vào những quy tắc được phát biểu trước, chúng ta vẽ được một đặc tả mối quan hệ giữa 2 thực thể:

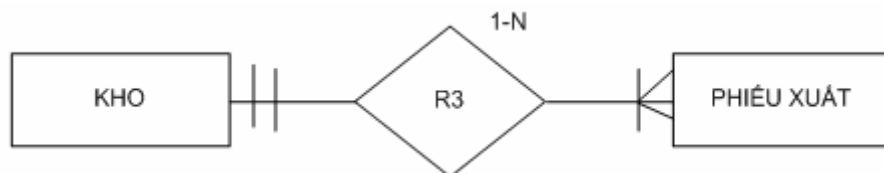
*“Trong mỗi phiếu xuất, ghi nhiều loại vật tư. Ngược lại mỗi loại vật tư có thể xuất hiện trong nhiều phiếu xuất với dữ liệu quan trọng là số lượng”.*



*“Một phiếu xuất chỉ viết cho một khách; ngược lại một khách có thể có nhiều phiếu xuất”*



*“Một phiếu xuất chỉ viết cho một kho, ngược lại mỗi kho có thể có nhiều phiếu xuất”*



2. Dựa vào khoá của các lược đồ quan hệ (Xác định qua khoá liên kết)

Ví dụ:

+ Trường hợp hệ lược đồ quan hệ đã chuẩn hoá

VẬT TƯ (mã vt, tên vt, ĐV, đơn giá),

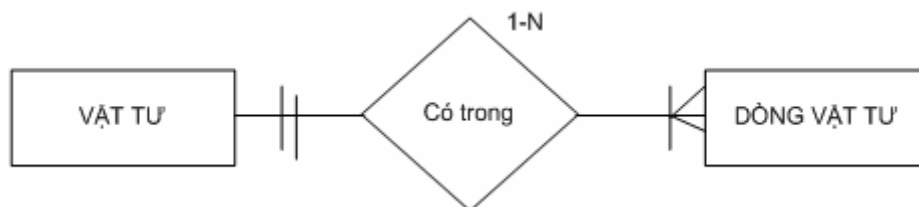
PHIẾU XUẤT (số phiếu, ngày, mã kho, mã khách),

KHÁCH (mã khách, tên khách, địa chỉ, E\_mail),

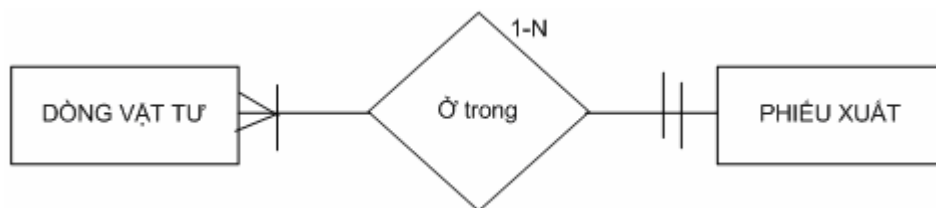
KHO (mã kho, tên kho)

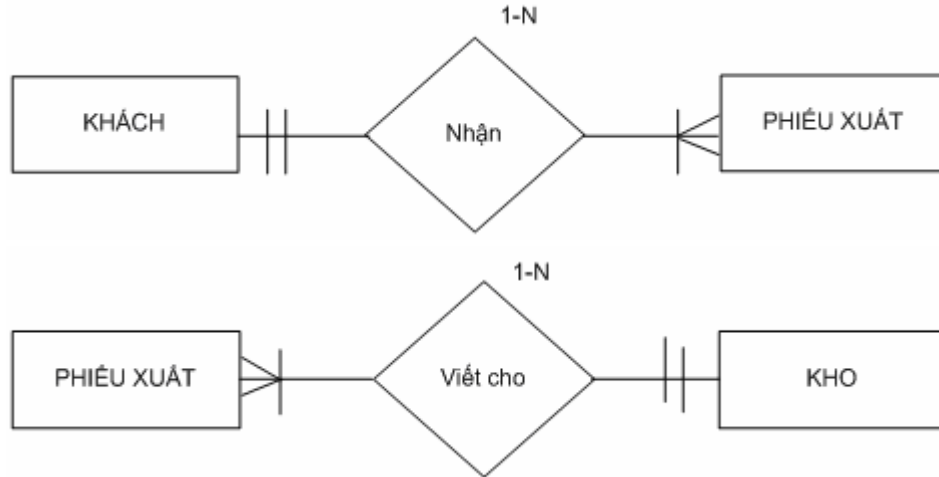
DÒNG VẬT TƯ (số phiếu, mã vt, số lượng)

Từ VẬT TƯ và DÒNG VẬT TƯ ta thấy có nhóm thuộc tính chung là “mã vt”, ở thực thể VẬT TƯ chúng là khoá chính, vậy ở thực thể DÒNG VẬT TƯ chúng là khoá ngoại. Sự tồn tại của khoá ngoại chứng tỏ giữa 2 lược đồ quan hệ đó có mối quan hệ với nhau. Ta vẽ được:



Phân tích tương tự, ta có:





**Khoá ngoại** (foreign key: FK) của 1 quan hệ là một nhóm thuộc tính trong quan hệ đó mà là khoá chính của một quan hệ khác liên kết với nó (các FK cũng có thể tham chiếu đến khoá chính trong cùng 1 quan hệ). Khoá ngoại còn gọi là khoá liên kết.

+ Trường hợp hệ lược đồ quan hệ chưa chuẩn hoá

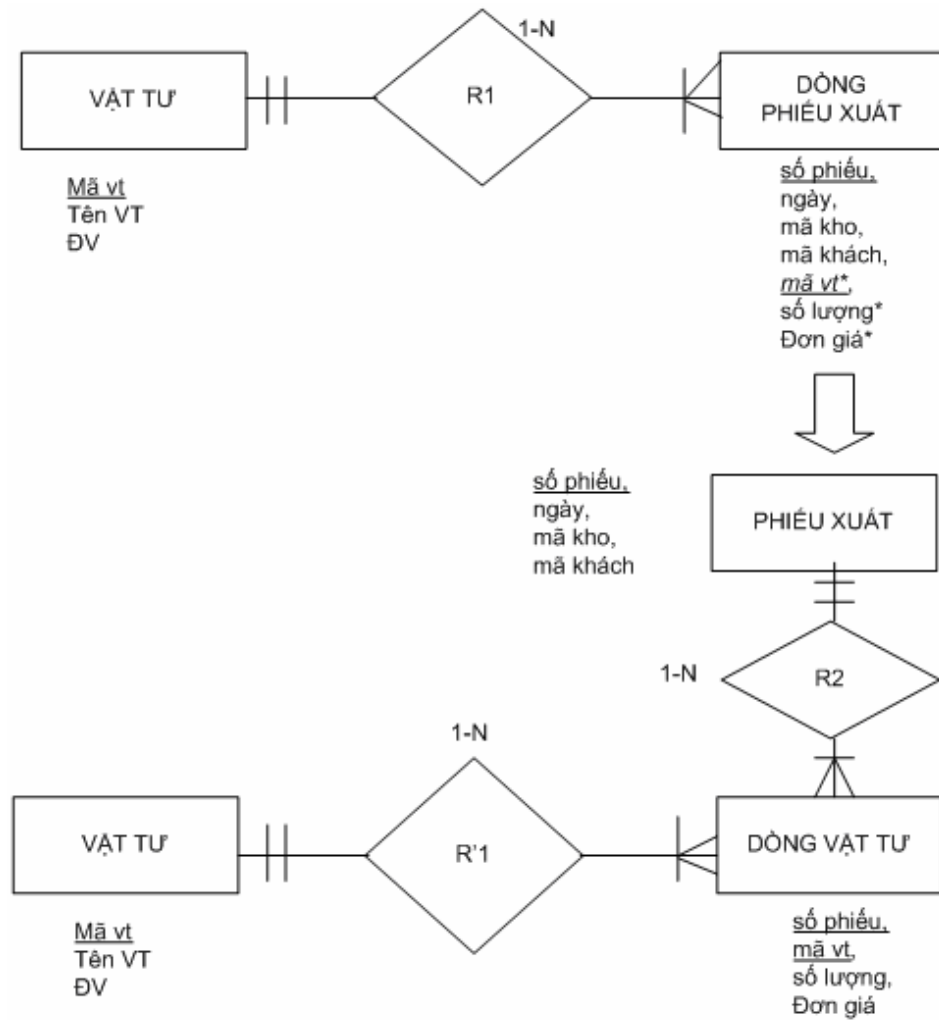
VẬT TƯ (mã vt, tên vt, ĐV),

DÒNG PHIẾU XUẤT (số phiếu, ngày, mã kho, mã khách, mã vt\*, số lượng\*, đơn giá)

KHÁCH (mã khách, tên khách, địa chỉ, E\_mail),

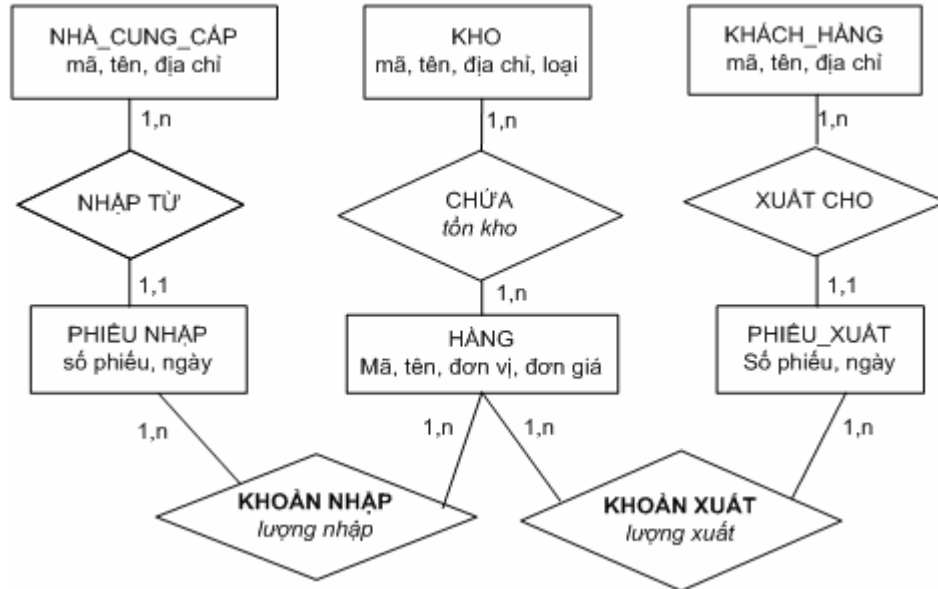
KHO (mã kho, tên kho)

Ta thấy “mã vt” là khoá ngoại của DÒNG PHIẾU XUẤT trong mối quan hệ với VẬT TƯ nên có thể đặc tả mối quan hệ giữa chúng trước, sau đó chuẩn hoá:



### 3.3.2.3. Kỹ thuật chuyển mô hình khái niệm dữ liệu về hệ lược đồ quan hệ

#### 1. Cho mô hình khái niệm dữ liệu của hệ quản lý kho



Hình 3.18. Mô hình khái niệm dữ liệu của hệ quản lý kho

**📖** Có 3 quy tắc chuyển:

*Quy tắc 1 (quy tắc biến đổi cơ bản):*

Mỗi **thực thể** được chuyển thành một quan hệ trong đó các thuộc tính của thực thể được chuyển thành thuộc tính của quan hệ, định danh của thực thể trở thành khóa của quan hệ.

*Quy tắc 2:*

Mỗi **mối quan hệ** 1-N mà không có thuộc tính riêng sẽ không được chuyển thành một quan hệ. Nhưng thực thể tham gia vào mối quan hệ về phía N (phía 1:1 trong mô hình) sẽ đổi mới bằng cách sau khi dùng phép biến đổi cơ bản sẽ nhận thêm khoá của thực thể tham gia vào mối quan hệ ở phía 1 làm khoá liên kết. Còn thực thể tham gia vào mối quan hệ ở phía 1 sẽ biến đổi theo quy tắc 1.

*Quy tắc 3:*

Mỗi **mối quan hệ** N-N hoặc mối quan hệ có thuộc tính riêng sẽ được chuyển thành một quan hệ mới. Quan hệ mới này có thuộc tính



gồm định danh của tất cả các thực thể trong mối quan hệ và các thuộc tính riêng của nó. Khóa của quan hệ được xác định lại sau đó. Các thực thể tham gia vào mối quan hệ đều biến đổi theo quy tắc 1.

Giả thiết ở đây (dựa theo bản số của mô hình) là **mỗi mặt hàng có thể để ở nhiều kho**. Ta thấy rằng:

+ Mối quan hệ NHẬP TỪ là mối quan hệ loại 1-N và không có thuộc tính riêng nên không chuyển thành một quan hệ mới. Thực thể NHÀ CUNG CẤP tham gia vào mối quan hệ này được chuyển thành quan hệ theo quy tắc biến đổi cơ bản. Thực thể PHIẾU NHẬP sau khi biến đổi theo quy tắc cơ bản sẽ được bổ sung thuộc tính Mã NCC làm thuộc tính liên kết. Kết quả ta có hai lược đồ sau:

NHA\_CUNG\_CAP (Mã NCC, Tên NCC, Địa chỉ NCC)

PHIEU\_NHAP (Số phiếu nhập, Ngày nhập, Mã NCC)

+ Mối quan hệ CHỨA là mối quan hệ loại N-N và có thuộc tính riêng nên được chuyển thành một quan hệ mới. Danh sách thuộc tính của nó bao gồm thuộc tính riêng “tồn kho” và các thuộc tính định danh của các thực thể tham gia vào mối quan hệ này (mã kho và mã hàng). Các thực thể KHO và HÀNG tham gia vào mối quan hệ đều được biến thành quan hệ theo quy tắc cơ bản. Kết quả ta có 3 lược đồ quan hệ sau:

KHO (Mã kho, Tên kho, Địa chỉ, Loại kho)

CHUA (Mã kho, Mã hàng, Tồn kho)

HANG (Mã hàng, Tên hàng, Đơn vị, Đơn giá)

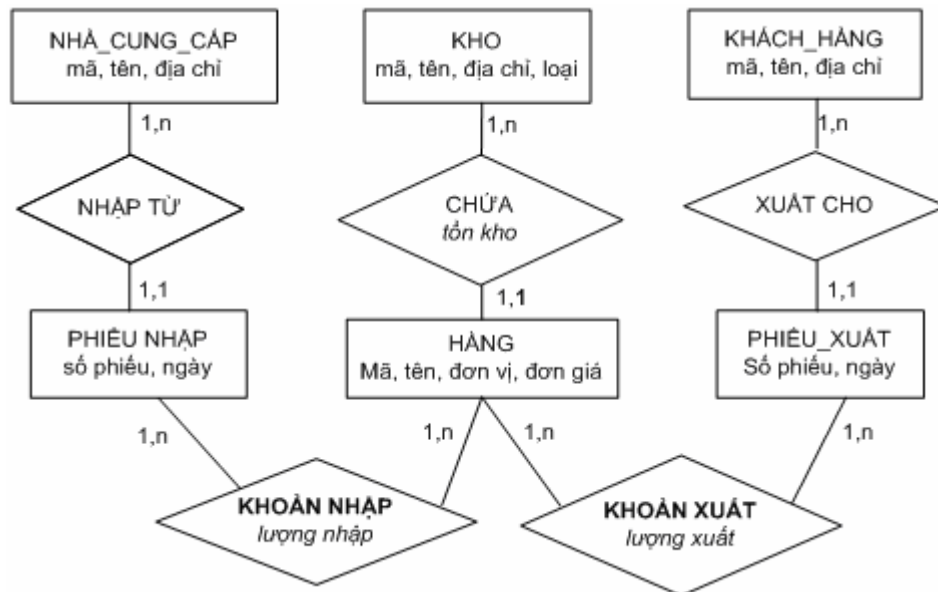
+ Các mối quan hệ khác đều được phân tích tương tự sẽ cho chúng ta hệ lược đồ quan hệ. Sau khi loại trừ các quan hệ bị bao hàm để đảm bảo là hệ Sperner ta sẽ còn lại 9 lược đồ quan hệ sau (những thuộc tính gạch chân là khóa):

1. NHA\_CUNG\_CAP (Mã NCC, Tên NCC, Địa chỉ NCC)

2. PHIEU\_NHAP (Số phiếu nhập, Ngày nhập, Mã NCC)

3. GOM\_CAC\_KHOAN\_NHAP (Số phiếu nhập, Mã hàng, Lượng nhập)
4. KHACH\_HANG (Mã khách, Tên khách, Địa chỉ khách)
5. PHIEU\_XUAT (Số phiếu xuất, Ngày xuất, Mã khách)
6. GOM\_CAC\_KHOAN\_XUAT (số phiếu xuất, mã hàng, Lượng xuất)
7. KHO (Mã kho, Tên kho, Địa chỉ, Loại kho)
8. CHUA (Mã kho, Mã hàng, Tồn kho)
9. HANG (Mã hàng, Tên hàng, Đơn vị, Đơn giá)

Chú ý: Nếu mỗi mặt hàng chỉ để ở 1 kho thì có:



Hình 3.19. Mô hình khái niệm dữ liệu

Và có CHUA (Mã kho, Mã hàng, Tồn kho)

Nên 2 quan hệ:

HANG (Mã hàng, Tên hàng, Đơn vị, Đơn giá)

Và: CHUA (Mã kho, Mã hàng, Tồn kho)

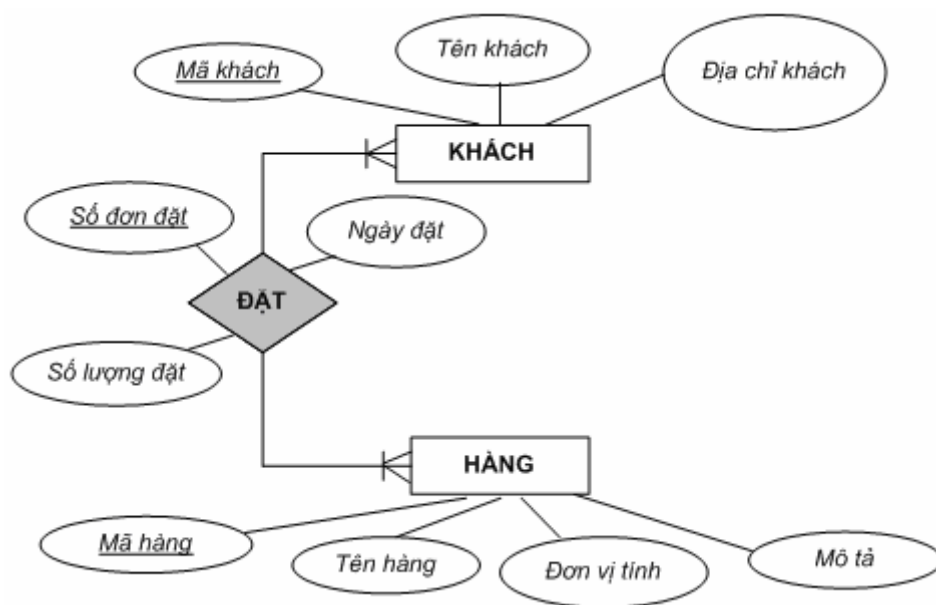
sẽ được hợp lại thành một:

HANG (Mã hàng, Tên hàng, Đơn vị, Đơn giá, Mãkho, Tồn kho)

Vậy hệ lược đồ trong trường hợp này chỉ còn có 8 lược đồ quan hệ sau:

1. NHA\_CUNG\_CAP (Mã NCC, Tên NCC, Địa chỉ NCC)
2. PHIEU\_NHAP (Số phiếu nhập, Ngày nhập, Mã NCC)
3. GOM\_CAC\_KHOAN\_NHAP (Số phiếu nhập, Mã hàng, Lượng nhập)
4. KHACH\_HANG (Mã khách, Tên khách, Địa chỉ khách)
5. PHIEU\_XUAT (Số phiếu xuất, Ngày xuất, Mã khách)
6. GOM\_CAC\_KHOAN\_XUAT (số phiếu xuất, mã hàng, Lượng xuất)
7. KHO (Mã kho, Tên kho, Địa chỉ, Loại kho)
8. HANG (Mã hàng, Tên hàng, Đơn vị, Đơn giá, Mãkho, Tồnkho)

## 2. Cho mô hình khái niệm bài toán quản lý kinh doanh



Hình 3.20. Mô hình khái niệm dữ liệu

*Chuyển sang mô hình quan hệ:*

Mối quan hệ ĐAT là mối quan hệ loại N-N, có thuộc tính riêng nên được chuyển thành một quan hệ:

ĐAT (số đơn, ngày đặt, mã khách, mã hàng \*, số lượng đặt\*)

*Chú ý:*

Mã hàng được đánh dấu \* trong ĐAT vì thuộc tính “số lượng đặt” là đa trị, tức là ứng với nhiều loại hàng.

Thực thể KHACH đã được tách riêng nên *mã khách* trong ĐAT không giữ vai trò là thành phần của khoá chính mà chỉ là khoá ngoại.

Còn thực thể KHACH và HANG tham gia vào mối quan hệ trên cũng được chuyển sang lược đồ quan hệ tương ứng theo quy tắc cơ bản:

KHACH (Mã khách, Tên khách, địa chỉ khách)

HANG (Mã hàng, Tên hàng, Mô tả hàng, đơn vị tính)

Kết quả ta có hệ lược đồ sau:

ĐAT (số đơn, ngày đặt, mã khách, mã hàng\*, số lượng đặt\*)

KHACH (Mã khách, Tên khách, địa chỉ khách)

HANG (Mã hàng, Tên hàng, Mô tả hàng, đơn vị tính)

#### 3.3.2.4. Kỹ thuật chuẩn hoá

**Chuẩn hoá** là một quá trình chuyển một cấu trúc dữ liệu phức hợp thành các cấu trúc dữ liệu tốt và đơn giản hơn.

Chuẩn hoá thường được hoàn thành sau một số bước, mỗi bước nhận được các quan hệ tương ứng với một dạng chuẩn. Dạng chuẩn được hiểu là một trạng thái của quan hệ có thể được xác định nhờ áp dụng các quy tắc để phát hiện sự phụ thuộc hàm (mối quan hệ) giữa các thuộc tính của quan hệ.

Chuẩn hoá dựa trên cơ sở phân tích các phụ thuộc hàm. Phụ thuộc hàm như đã nói ở trên là một mối quan hệ cụ thể giữa hai thuộc tính (hay nhóm thuộc tính) trong một quan hệ.

Các dữ liệu trong quan hệ không dễ cho thấy về sự tồn tại của các phụ thuộc hàm. Chỉ có sự hiểu biết và tri thức của con người về các đối tượng mà chính quan hệ đó mô tả mới cho phép xác định các phụ thuộc hàm vốn tồn tại trong nó. Tuy có những thuật toán (thường phức tạp) cho phép phát hiện ra có hay không các sự phụ thuộc hàm giữa hai hay nhiều thuộc tính của nó nhưng thường ít dùng.

*a. Các dạng chuẩn cơ bản*

Có 3 dạng chuẩn cơ bản là:

*Chuẩn 1* (first-normal-form: 1NF): Một quan hệ đạt chuẩn 1 nếu nó không chứa các thuộc tính lặp.

*Chuẩn 2* (second-normal-form: 2NF) (Các thuộc tính ngoài khóa phụ thuộc hoàn toàn vào khóa)

Một quan hệ đạt chuẩn 2 nếu:

- Đạt chuẩn 1
- Không tồn tại thuộc tính ngoài khóa mà phụ thuộc vào một phần của khóa.

*Chuẩn 3* (third-normal-form: 3NF) (Các thuộc tính ngoài khóa phụ thuộc trực tiếp vào khóa)

Một quan hệ đạt chuẩn 3 nếu:

- Đạt chuẩn 2
- Không tồn tại thuộc tính ngoài khóa mà phụ thuộc bắc cầu vào khóa (qua một thuộc tính gọi là *thuộc tính cầu* (cũng là thuộc tính ngoài khóa)).

Để chuẩn hoá một hệ lược đồ quan hệ, ta xét lần lượt từng quan hệ và kiểm tra tính chuẩn của nó. Muốn vậy, trước hết ta *xác định các phụ thuộc hàm và khóa chính* (khóa tối thiểu) của quan hệ. Sau đó tiến hành kiểm tra lần lượt các loại chuẩn đối với quan hệ:

*Ví dụ:*

- Cho lược đồ quan hệ:

PHIEUNHAP (sophieu, ngaynhap, makhach, ten\_kh, diachi\_kh,

1                      2                      3                      4                      5

đienthoai, makho, điachikho, hinhthucthanhtoan,loaitien,

6              7                      8                      9                      10

mavattu\*, tenvattu\*, *soluong\**, donvitinh\*, đongia\*, tyleVAT\*)

11              12                      13                      14                      15                      16

- Các bước thực hiện:

+ *Bước 1: Xác định phụ thuộc hàm*

Việc xác định phụ thuộc hàm dựa vào 2 yếu tố: nghiệp vụ của các nhà chuyên môn và kinh nghiệm của nhà phân tích. Đối với ví dụ đơn này, ta thấy lược đồ quan hệ gồm 2 nhóm: nhóm thuộc tính 1 gồm danh sách các thuộc tính không lặp (từ 1 đến 10), nhóm thuộc tính 2 gồm các thuộc tính lặp (từ 11 đến 16). Trên cơ sở đó ta xác định 3 nhóm phụ thuộc hàm tương ứng:

+ Nhóm phụ thuộc hàm thứ nhất: xác định phụ thuộc hàm trong nhóm thuộc tính không lặp. Theo ví dụ ta có:

(1) -> (2, 3, 4, 5, 6, 7, 8, 9, 10)

+ Nhóm phụ thuộc hàm thứ hai: xác định phụ thuộc hàm trong nhóm thuộc tính lặp. Theo ví dụ ta có:

(11) -> (12, 14, 15, 16)

+ Nhóm phụ thuộc hàm thứ ba: xác định phụ thuộc hàm trong nhóm hỗn hợp cả thuộc tính lặp và không lặp. Theo ví dụ ta có:

(1, 11) -> (13)

“số lượng” ở đây là số lượng vật tư và có giá trị khác nhau trên các PHIEUNHAP khác nhau.

Sau này, mở rộng trong các bài toán lớn như bài toán quản lý nhân sự chẳng hạn, có nhiều nhóm lặp khác nhau, thì chúng ta sẽ xác định phụ thuộc hàm riêng cho từng nhóm lặp và xác định phụ thuộc hàm hỗn hợp của từng nhóm lặp riêng với nhóm các thuộc tính không lặp.

+ *Bước 2: Xác định K chính (K tối tiểu) (dùng thuật toán)*

Khoá chính, phần lớn các trường hợp, thường là các thuộc tính nằm phía bên trái các phụ thuộc hàm mà chúng ta xác định được. Gọi nhóm thuộc tính phía trái các phụ thuộc hàm là K. Nhờ thuật toán tính bao đóng ta xác định được K có là khoá (thường) hay không. Thông thường với người phân tích có kinh nghiệm xác định phụ thuộc hàm thì K là khoá ngay (nếu chưa đạt khoá thì ta dễ dàng bổ sung các thành phần thuộc tính còn thiếu để đạt là khoá). Sau đó dùng thuật toán tính khoá tối tiểu áp dụng cho ngay khoá vừa tìm được ta sẽ thu được một khoá tối tiểu mà ta gọi là khoá chính.

Ví dụ ở đây: nhóm thuộc tính  $K = (1, 11)$

có thể suy ra toàn không gian thuộc tính của quan hệ. Hơn nữa, kiểm tra được chúng chính là khoá tối tiểu (lần lượt bớt từng thuộc tính và kiểm tra lại bao đóng). Cuối cùng thu được:

PHIEUNHAP (sophieu, ngaynhap, makhach, ten\_kh, diachi\_kh, dienthoai, makho, diachikho, hinhthucthanhtoan, loaitien, mavattu\*, tenvattu\*, soluong\*, donvitinh\*, dongia\*, tyleVAT\*)

+ *Bước 3: Chuẩn hoá 1, 2, 3*

Vì các thuộc tính: 11,...,16 (có dấu \*) là các thuộc tính lặp nên quan hệ này chưa đạt chuẩn 1.

*Nếu quan hệ không phải là chuẩn 1: Phân rã quan hệ thành hai quan hệ:*

Quan hệ 1 (QH1): Các thuộc tính lặp và khoá chính của lược đồ gốc (chưa tách).

Quan hệ 2 (QH2): Các thuộc tính đơn (các thuộc tính không lặp).

QH1: gồm các thuộc tính lặp (11,...,16) và (1) là khoá tức là:

VATTU ( <u>Số phiếu</u> , <u>mavattu</u> , tenvattu, <u>soluong</u> , donvitinh, dongia, tyleVAT)	<u>1</u>	<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	<u>15</u>
						<u>16</u>

QH2: gồm (2, 3, 4, 5,...,10) và (1) tức là:

PHIEUNHAP (sophieu, ngaynhap, makhach, ten\_kh, diachi\_kh,

1 2 3 4 5

đienthoai, makho, diachikho, hinhthucthanhtoan, loaitien)

6 7 8 9 10

*Nếu quan hệ không phải là chuẩn 2: Phân rã thành hai quan hệ:*

Quan hệ 1: Các thuộc tính phụ thuộc vào một phần khoá chính và phần khoá chính xác định chúng.

Quan hệ 2: Các thuộc tính còn lại và khoá chính của lược đồ gốc (chưa tách).

*Xét QH1:*

VATTU (Số phiếu, mavattu, tenvattu, soluong, donvitinh, đongia, tyleVAT)

1 11 12 13 14 15

16

Ta thấy 11 -> 12, 14, 15, 16, mà 11 là bộ phận của khoá nên quan hệ này không đạt chuẩn 2.

Để đạt chuẩn 2 ta tách thành 2 quan hệ:

QH1\_1: VATTU (mavattu, tenvattu, donvitinh, đongia, tyleVAT)

11 12 14 15 16

QH 1\_2: DONGVATTU (Số phiếu, mavattu, soluong)

1 11 13

*Xét tiếp QH2:*

PHIEUNHAP (sophieu, ngaynhap, makhach, ten\_kh, diachi\_kh,

1 2 3 4 5

đienthoai, makho, diachikho, hinhthucthanhtoan, loaitien)

6 7 8 9 10

Quan hệ này đương nhiên đạt chuẩn 2 vì nhóm thuộc tính khoá chỉ có một thuộc tính.



Nếu quan hệ không phải là chuẩn 3: phân rã quan hệ thành 2 quan hệ:

Quan hệ 1: Các thuộc tính phụ thuộc bắc cầu và thuộc tính cầu.

Quan hệ 2: Các thuộc tính còn lại và các thuộc tính cầu.

Xét:

QH1\_1: 1. VATTU (mavattu, tenvattu, donvitinh, đongia, tyleVAT)

11      12      14      15      16

đạt chuẩn 3 vì không có thuộc tính cầu.

Xét:

QH 1\_2: 2. DONGVATTU (Số phiếu, mavattu, soluong)

1      11      13

cũng đạt chuẩn 3 vì thuộc tính ngoài khoá chỉ có một nên không có thuộc tính cầu.

Xét tiếp QH2:

PHIEUNHAP (sophieu, ngaynhap, makhach, ten\_kh, diachi\_kh,

1      2      3      4      5

đienthoai, makho, diachikho, hinhthucthanhtoan, loaitien)

6      7      8      9      10

Ta thấy 3 -> 4, 5, 6

7 -> 8

mà 3 và 7 là thuộc tính ngoài khoá nên chúng là những thuộc tính cầu.

Vì vậy để có chuẩn 3 ta tách thành những quan hệ sau:

QH2\_1: 3. KHACH (makhach, ten\_kh, diachi\_kh, đienthoai)

3      4      5      6

QH2\_2: 4. KHO (makho, diachikho)

7      8

và QH2\_3: 5. PHIEUNHAP (sophieu, ngaynhap, makhach, makho, hinhthucthanhtoan, loaitien)      1      2      3      7

9      10

*Kết luận:*

Ta có hệ lược đồ quan hệ sau đạt 3NF:

1. VATTU (mavattu, tenvattu, donvitinh, dongia, tyleVAT)
2. DONGVATTU (Số phiếu, mavattu, soluong)
3. KHACH (makhach, ten\_kh, diachi\_kh, dienthoai)
4. KHO (makho, diachikho)
5. PHIEUNHAP(sophieu, ngaynhap, *makhach*, *makho*, hinhthucthanhtoan, loaitien)

Ở đây, chúng ta lưu ý một số kỹ thuật tách chúng.

*Xét chuẩn 1:*

+ Nếu quan hệ còn các thuộc tính có dấu \* (thuộc tính lặp) nghĩa là quan hệ đó chưa đạt chuẩn 1. Ta sử dụng quy tắc tách bình thường.

+ Trong trường hợp có nhiều nhóm thuộc tính lặp (xen kẽ với các thuộc tính không lặp) ta nên đồng thời tách theo từng nhóm lặp. Hay nói rõ hơn, khi có m nhóm lặp ta sẽ tách lược đồ đó theo quy tắc đã biết để thu được đồng thời m+1 lược đồ quan hệ mới đạt chuẩn 1.

*Xét chuẩn 2:*

+ Mọi quan hệ đã đạt chuẩn 1, chưa đạt chuẩn 2, đều dễ dàng tách thành các quan hệ đạt chuẩn 2 theo quy tắc đã biết.

+ Trường hợp nhóm thuộc tính khoá chỉ có một thuộc tính, điều này có nghĩa là phụ thuộc bộ phận không thể xảy ra, ta kết luận ngay là quan hệ đó đương nhiên đạt chuẩn 2.

+ Trong trường hợp có m nhóm phụ thuộc bộ phận, ta sẽ tách đồng thời thành m+1 quan hệ đạt chuẩn 2 dựa theo quy tắc đã biết.

*Xét chuẩn 3:*

Cũng giống như các chuẩn trên, khi có  $m$  thuộc tính cầu, ta sẽ tách được  $m+1$  quan hệ mới đạt chuẩn 3.

+ Trường hợp nhóm thuộc tính ngoài khoá chỉ có 1 thuộc tính, điều này có nghĩa không thể tồn tại thuộc tính cầu, nên quan hệ đó đương nhiên đạt chuẩn 3.

**3.3.2.5. Kỹ thuật chuyển từ hệ lược đồ quan hệ sang sơ đồ E-R**  
**(ERD-mô hình dữ liệu logic)**

Để dễ nhận thức và trao đổi, mô hình E-R thường được biểu diễn dưới dạng một đồ thị, trong đó các nút là các thực thể, còn các cung là các mối quan hệ (các kiểu liên kết các thực thể).

*Mô hình được lập như sau:*

Mỗi thực thể được biểu diễn bằng một hình chữ nhật có 2 phần: phần trên là tên thực thể (viết in), phần dưới chứa danh sách các thuộc tính, trong đó thuộc tính khoá được đánh dấu (mỗi thực thể chỉ xác định một khoá-tối thiểu). Tên thực thể thường là danh từ (chỉ đối tượng).

Một mối quan hệ được biểu diễn thường bằng hình thoi/elíp, được nối bằng nét liền tới các thực thể tham gia vào mối quan hệ đó. Trong hình thoi là tên của mối quan hệ cũng được viết in, danh sách các thuộc tính của nó thì được viết thường. Tên của mối quan hệ thường là động từ chủ động hay bị động. Trong phương pháp MERISE, mối quan hệ thường được biểu diễn bằng hình elíp. Mô hình E\_R cuối cùng thường mối quan hệ không còn loại N-N. Trong mối quan hệ nhị nguyên thì ở hai đầu mút các đường nối, sát với thực thể, người ta vẽ đường ba chẽ (còn gọi là đường chân gà) về phía có khoá ngoại (khoá liên kết) thể hiện nhiều, còn phía kia thể hiện một. Bản số trong mỗi đặc tả mối quan hệ giữa 2 thực thể là cặp Max của hai bản số xác định trong đặc tả và được gọi là bản số trực tiếp.

*Chú ý:*

+ Mỗi quan hệ có thể không có thuộc tính. Khi có, ta thường gọi là thuộc tính riêng và cũng được viết trong hình thoi song nhớ rằng chỉ viết chữ thường (phân biệt tên của mỗi quan hệ viết bằng chữ in)

+ Giữa 2 thực thể có thể có nhiều mối quan hệ và chúng cần vẽ riêng rẽ, không chập vào nhau.

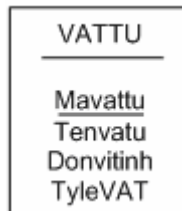
Việc chuyển từ hệ lược đồ quan hệ sang sơ đồ E-R được thực hiện theo các bước sau:

*Bước 1:*

Từ mỗi lược đồ quan hệ, vẽ một hình chữ nhật bao gồm tên lược đồ cùng với các thuộc tính của nó:

*Ví dụ:* Cho lược đồ quan hệ:

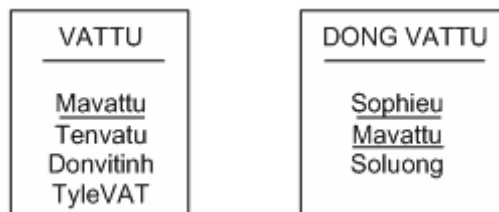
VATTU (Mavattu, Tenvatu, Donvitinh, TyleVAT)



*Bước 2:* Xác định quan hệ (thể hiện bằng đường nối) giữa 2 thực thể (thể hiện bằng hình chữ nhật) bất kỳ.

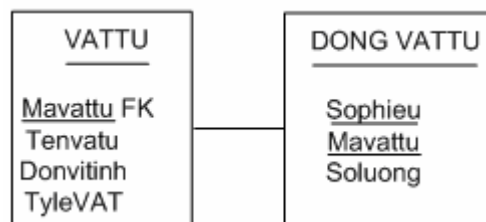
*Bước 2.1:* Xác định nhóm thuộc tính chung.

Nếu không có nhóm thuộc tính chung thì kết luận: không có quan hệ

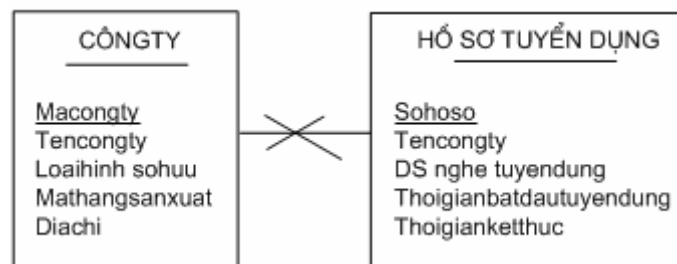


Ở đây, hai thực thể VẬT TƯ và DÒNG VẬT TƯ có nhóm thuộc tính chung là mã vật tư.

*Bước 2.2:* Kiểm tra nhóm thuộc tính chung đó có phải là khoá chính của 1 trong 2 thực thể? Ở đây có mã vật tư là khoá chính trong VẬT TƯ.

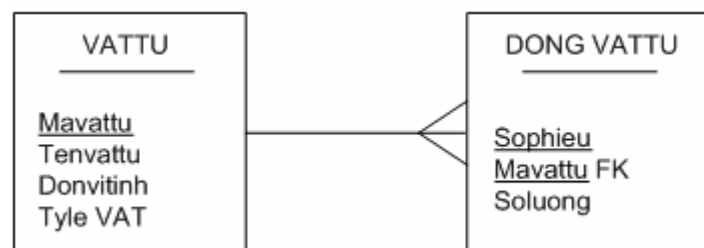


Nếu không có thì kết luận không có quan hệ: tên công ty là thuộc tính chung nhưng không phải là khoá chính của thực thể nào cả.



*Bước 2.3:* Kết luận tồn tại khoá ngoại: Nhóm thuộc tính chung ở bên thực thể không phải là khóa chính mà sẽ là khóa ngoại.

*Bước 3:* Vẽ đường nối giữa 2 hình chữ nhật và đường 3 chẽ gắn với thực thể chứa khoá ngoại.



Kết quả, từ hệ 5 lược đồ quan hệ đã chuẩn hoá 3NF:

KHACH (Makhach, Tenkhach, Diachi, Dienthoai)

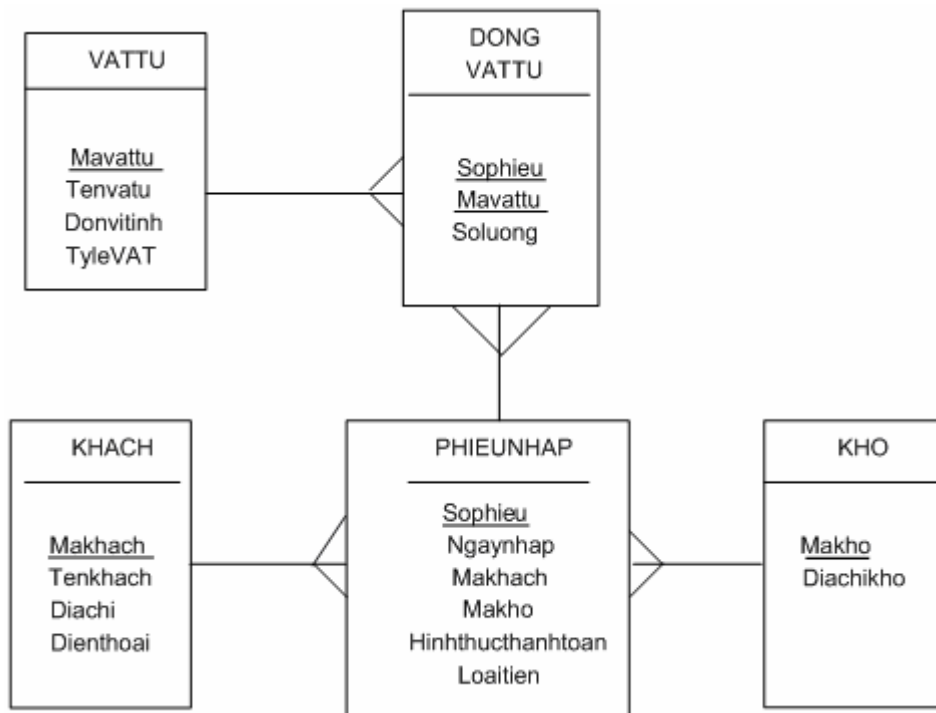
KHO (Mã kho, địa chỉ kho)

VATTU (Mavattu, Tenvatu, Donvitinh, TyleVAT)

DONGVATTU (Sophieu, Mavattu, Soluong)

PHIEUNHAP (Sophieu, Ngaynhap, Makhach, Makho, hinhthucthanhtoan, Loaitien)

ta xây dựng được sơ đồ E-R như sau:



Hình 3.21. Sơ đồ E-R tương ứng với hệ lược đồ quan hệ đã chuẩn hoá đến 3NF

Chú ý:

1. CHINHANH (Mã chi nhánh, tên chi nhánh, mã công ty)
2. CONGTY (Mã công ty, tên công ty, địa chỉ công ty)

3. DUAN (Mã dự án, tên dự án, mã chi nhánh)

4. NHANVIEN (Mã nhân viên, tên nhân viên, địa chỉ nhân viên, mã chi nhánh, Mã nghề\*, trình độ nghề\*) (thuộc tính có dấu \* là thuộc tính lập)

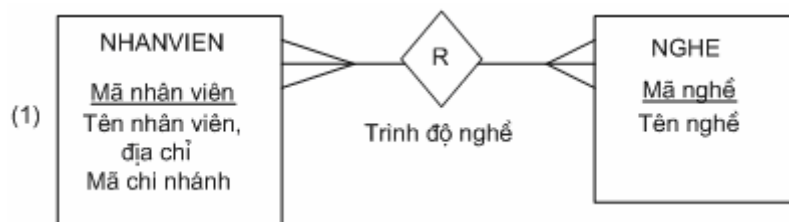
5. NGHE (Mã nghề, tên nghề)

Đặc tả quan hệ giữa các thực thể:

✂ Nhận thấy:

4. NHANVIEN (Mã nhân viên, tên nhân viên, địa chỉ nhân viên, mã chi nhánh, Mã nghề\*, trình độ nghề\*) cho biết 1 nhân viên có nhiều nghề.

5. NGHE (Mã nghề, tên nghề): 1 nghề có thể có nhiều nhân viên được đào tạo. Vì vậy ta có:



Nhận xét: Cách giải quyết này đòi hỏi kinh nghiệm của người phân tích.

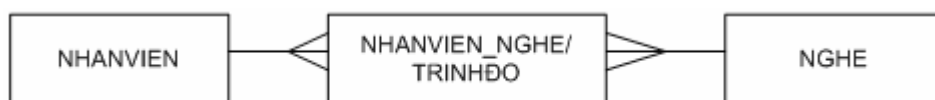
✂ Nhận thấy:

Quan hệ NHANVIEN nếu tách thành 2 quan hệ đạt chuẩn 1:

4. NHANVIEN (Mã nhân viên, tên nhân viên, địa chỉ nhân viên, mã chi nhánh)

6. TRINHDO (Mã nhân viên, Mã nghề, trình độ nghề)

thì ta cũng có tương đương (1):

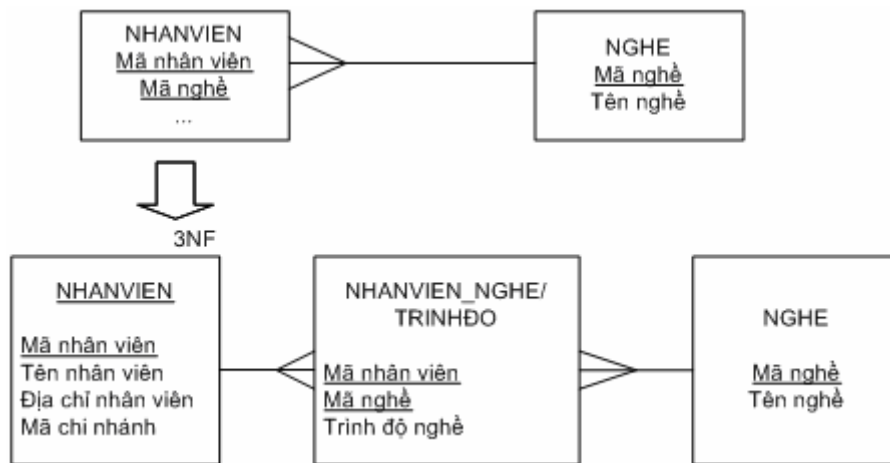


Hệ mới có 6 lược đồ (thêm TRINHDO) nhưng sơ đồ E\_R như nhau.

*Nhận xét:* dễ đặc tả hơn: đầu tiên là chuẩn hoá; sau đó xác định khoá liên kết để vẽ sơ đồ đặc tả mối quan hệ giữa 2 thực thể.

*Nhận thấy:*

Từ quan hệ 4. NHANVIEN và 5. NGHE, có đặc tả quan hệ giữa chúng (dựa vào việc xác định K ngoại):



*Nhận xét:*

Phương pháp này rất tự nhiên: đầu tiên dựa vào khoá liên kết để đặc tả mối quan hệ giữa hai thực thể, sau đó chuẩn hoá.

Trường hợp hệ lược đồ đạt 3NF có nguồn gốc từ mô hình khái niệm dữ liệu thì ta dựa vào chính mô hình khái niệm dữ liệu để xác định mô hình E\_R (khung thiết kế).

### 3.3.2.6. Ví dụ

*Ví dụ 1- Phương pháp mô hình:*

Một cơ sở bán hàng sử dụng hai loại chứng từ sau để theo dõi hoạt động kinh doanh của mình:



### A. ĐƠN ĐẶT HÀNG

Số hoá đơn:

Ngày đặt:

Người đặt hàng:

Địa chỉ:

STT	Tên hàng	Mô tả hàng	Đơn vị tính	Số lượng (đặt)

### B. PHIẾU GIAO HÀNG

Số phiếu:

Ngày giao:

Tên khách hàng:

Địa chỉ:

Nơi giao hàng:

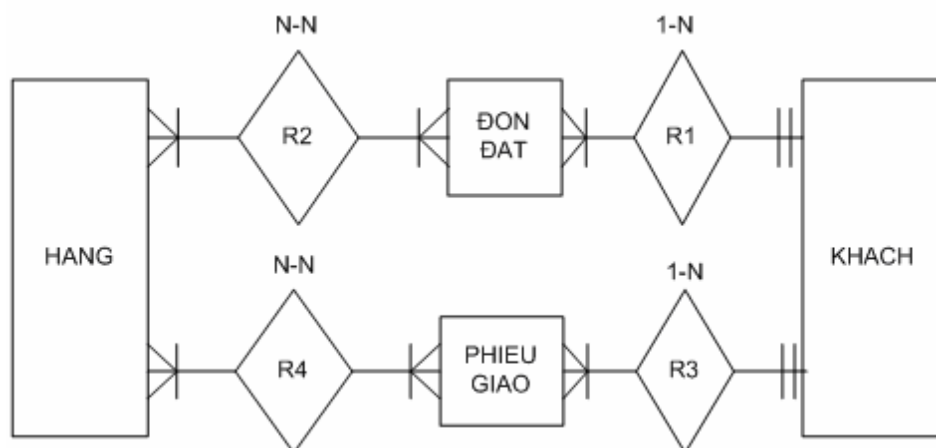
STT	Tên hàng	Đơn vị tính	Đơn giá	Số lượng (giao)	Thành tiền

TONGTIEN:

Cần thiết kế CSDL loại quan hệ (bao gồm các file dữ liệu và sơ đồ mô hình dữ liệu) từ các tài liệu trên để quản lý hoạt động bán hàng.

*Giả thiết:* Đơn giá giao không chỉ phụ thuộc vào mã hàng (có thể phụ thuộc vào khách, số lượng, thời gian).

*Thủ tục tiến hành:* Phần trên, các bước 1, 2, 3, 4 đã thực hiện và chúng ta đã có mô hình khái niệm dữ liệu sau (hình 3.5):



Chúng ta tiếp tục các bước còn lại:

5. Chuyển mô hình khái niệm dữ liệu sang mô hình quan hệ

HANG, KHACH chuyển sang dạng lược đồ theo quy tắc cơ bản ta có:

HANG (mã hàng, Tên hàng, Mô tả hàng, Đơn vị hàng)

KHACH (mã khách, Tên khách, Địa chỉ khách)

DONDAT, PHIEUGIAO chuyển sang lược đồ quan hệ theo quy tắc 2 ta có:

DONDAT (Số đơn đặt, Ngày đặt hàng, mã khách)

PHIEUGIAO (Số phiếu giao, Nơi giao, Ngày giao, mã khách)

R2, R4 chuyển sang lược đồ quan hệ theo quy tắc 3 ta có:

DONGDON (Số đơn đặt, mã hàng, SL đặt)

DONGPHIEU (Số phiếu giao, mã hàng, giá, SL giao)

Kết quả, chúng ta có hệ lược đồ quan hệ sau:

1. HANG (Mã hàng, Tên hàng, Mô tả hàng, Đơn vị hàng)

2. KHACH (Mã khách, Tên khách, Địa chỉ khách)

3. DONDAT (Số đơn đặt, Ngày đặt hàng, mã khách)

4. PHIEUGIAO (Số phiếu giao, Nơi giao, Ngày giao, mã khách)

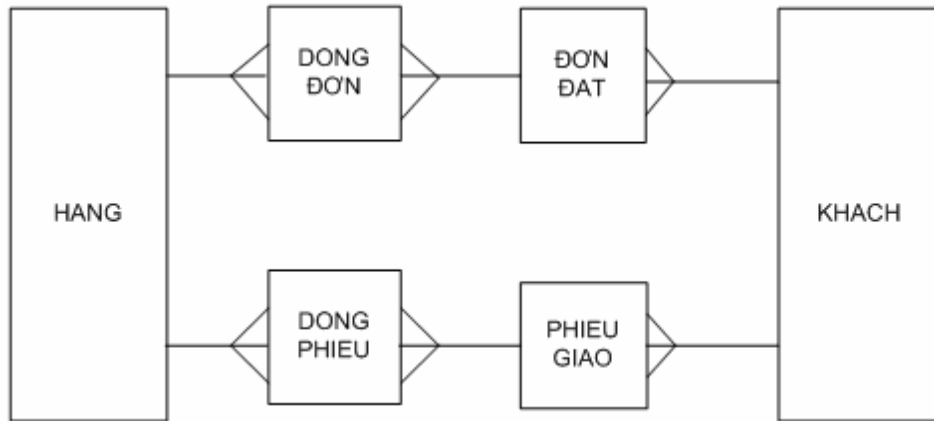
5. DONGDON (Số đơn đặt, mã hàng, SL đặt)

6. DONGPHIEU (Số phiếu giao, mã hàng, giá, SL giao)

6. Tiến hành chuẩn hoá các quan hệ nhận được đến 3NF

Cả 6 lược đồ trên đều đạt chuẩn 3NF

7. Vẽ sơ đồ E\_R: Hình 3.22.

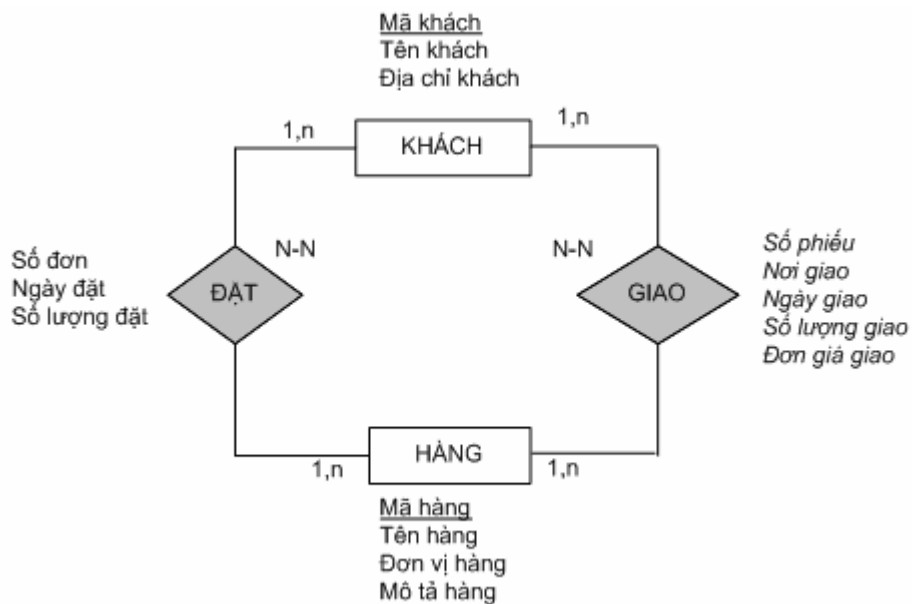


Hình 3.22

Ví dụ 2: Phương pháp trực giác

Thủ tục tiến hành:

Phần trên, các bước 1, 2, 3, 4 đã thực hiện và chúng ta đã có mô hình khái niệm dữ liệu sau (hình 3.9):



Chuyển mô hình khái niệm dữ liệu sang mô hình quan hệ:

HANG, KHACH chuyển sang dạng lược đồ quan hệ thành:

HANG (Mã hàng, Tên hàng, Mô tả hàng, Đơn vị hàng)

KHACH (Mã khách, Tên khách, Địa chỉ khách)

DAT, GIAO chuyển sang lược đồ quan hệ thành:

DONDAT (Số đơn, Ngày đặt, số lượng đặt\*, mã khách, mã hàng\*)

PHIEUGIAO (Số phiếu, Nơi giao, Ngày giao, số lượng giao\*, đơn giá giao\*, mã khách, mã hàng\*)

*Chuẩn hoá: Hai quan hệ:*

HANG (Mã hàng, Tên hàng, Mô tả hàng, Đơn vị hàng)

KHACH (Mã khách, Tên khách, Địa chỉ khách)

đều đạt 3NF.

*Còn 2 quan hệ:*

DONDAT (Số đơn, Ngày đặt, số lượng đặt\*, mã khách, mã hàng\*)

PHIEUGIAO (Số phiếu, Nơi giao, Ngày giao, số lượng giao\*, đơn giá giao\*, mã khách, mã hàng\*)

không đạt chuẩn 1 vì còn thuộc tính lặp. Để đạt 1NF, ta tách chúng thành:

DONGDON (Số đơn, số lượng đặt, mã hàng)

DONDAT (Số đơn, Ngày đặt, mã khách,)

DONGPHIEU (Số phiếu, số lượng giao, đơn giá giao, mã hàng)

PHIEUGIAO (Số phiếu, Nơi giao, Ngày giao, mã khách)

*Kết quả, chúng ta có hệ lược đồ quan hệ sau giống như ví dụ trên:*

1. HANG (Mã hàng, Tên hàng, Mô tả hàng, Đơn vị hàng)

2. KHACH (Mã khách, Tên khách, Địa chỉ khách)

3. DONDAT (Số đơn, Ngày đặt, mã khách)

4. PHIEUGIAO (Số phiếu, Nơi giao, Ngày giao, mã khách)

5. DONGDON (Số đơn đặt, mã hàng, số lượng đặt)

6. DONGPHIEU (Số phiếu, mã hàng, đơn giá giao, số lượng giao)

### 3.3.3. Thiết kế cơ sở dữ liệu logic dựa trên “Phương pháp từ điển”

Trong một số trường hợp, các dữ liệu thu được là những hồ sơ, đầu vào chứa tất cả dữ liệu cần thiết để xây dựng cơ sở dữ liệu. Khi đó người ta có thể *lập các từ điển mà mỗi tài liệu được xem như một thực thể vốn đã tồn tại*. Quá trình xây dựng cơ sở dữ liệu được bắt đầu từ “từ điển dữ liệu” thu được. Đây là một phương pháp được sử dụng rất sớm từ khi có mô hình dữ liệu quan hệ.

***Quy trình thiết kế một CSDL có thể được mô tả như sau:***

1. *Mỗi hồ sơ gốc được xem như một thực thể*. Liệt kê từng thực thể và tất cả các thông tin đặc trưng trong nó xem như những thuộc tính của thực thể đó.

2. *Chính xác hoá thông tin* là những thuộc tính của các thực thể, sao cho hai thông tin có cùng tên gọi trong toàn từ điển phải cùng nghĩa. Nếu chúng có nghĩa khác nhau thì tên gọi phải khác nhau.

3. *Chọn lọc và mã hoá các thông tin* thu được theo nguyên tắc sau:

- Đi lần lượt từ trên xuống dưới.
- Các thực thể và thuộc tính được mã hoá bằng tên mới đảm bảo yêu cầu của hệ quản trị CSDL sẽ sử dụng để xây dựng các file cho nó.
- Nếu một thuộc tính là thuộc tính *tên gọi* mà lần đầu tiên gặp nó và có thể sử dụng làm *định danh* cho thực thể tương ứng với nó thì sử dụng làm định danh và đánh dấu nó. Nếu không thể làm định danh thì phải thêm một thuộc tính định danh cho thuộc tính này và tiến hành mã hoá chúng.
- Nếu thuộc tính là *tên gọi* gặp lần thứ 2 trở đi thì thay nó bằng thuộc tính định danh tương ứng ở trên.
- Nếu một thuộc tính không phải là *định danh* hoặc *tên gọi* (tức là *thuộc tính mô tả*) gặp lần đầu thì nó được chọn và mã hoá. Nếu là thuộc tính mô tả gặp lần thứ 2 trở đi thì bỏ qua.

- Nếu một *thuộc tính mô tả* là thuộc tính trung gian (có thể suy ra từ các thuộc tính đã được mã hoá) thì cũng bỏ qua. Ví dụ như “thành tiền” được suy từ “đơn giá” và “số lượng”

4. Xác định các thuộc tính lặp trong các thực thể nhận được (thuộc tính trong bảng).

5. Xác định thuộc tính định danh của các thực thể (hồ sơ gốc) (đánh dấu bằng gạch dưới thuộc tính)

6. Tiến hành chuẩn hoá các thực thể đã mã hoá để thu được các quan hệ đạt chuẩn 3NF

7. Tiến hành tích hợp các quan hệ đã được chuẩn hóa, xác định các khoá chính, khoá ngoại của chúng. Kết quả thu về 1 hệ lược đồ quan hệ đã được chuẩn hoá đến 3NF sao cho hệ các lược đồ quan hệ đó tạo thành một hệ Sperner (K là một hệ Sperner nếu với  $\forall A, B \in K \Rightarrow$  không xảy ra:  $A \subseteq B$  hoặc  $B \subseteq A$ )

8. Vẽ sơ đồ E\_R

- Mỗi quan hệ tương ứng với một hình chữ nhật, phần trên là tên quan hệ, phần dưới là các thuộc tính của nó

- Đánh dấu thuộc tính khóa (in đậm, gạch dưới, ...)

- Hai quan hệ được nối với nhau nếu chúng có mối quan hệ (thể hiện chúng có khoá ngoại)

- Bản số trực tiếp giữa 2 thực thể được kiểm nghiệm dựa trên ý nghĩa thực tiễn của nó hoặc dựa vào khoá ngoại.

*Chú ý:*

Việc xác định mối liên kết (vẽ được đường nối) giữa 2 thực thể dựa trên các kết quả của công việc truy xét sau:

+ Tìm được nhóm thuộc tính chung giữa 2 quan hệ: nếu tại 1 quan hệ, nhóm thuộc tính đó là khoá chính thì chúng sẽ là khoá ngoại của quan hệ còn lại, tức là tồn tại liên kết giữa 2 quan hệ đó. Khi đó ta kẻ

được 1 đường nối liên kết giữa chúng. Đường ba chẽ sẽ gắn với quan hệ chứa khoá ngoại.

+ Nếu không tìm được nhóm thuộc tính chung hoặc có nhóm thuộc tính chung nhưng chúng không là khoá chính của một quan hệ nào cả thì 2 quan hệ đó sẽ không có liên kết, tức là không có đường nối giữa chúng.

*Ví dụ:*

Một công ty kinh doanh quan tâm đến hai loại chứng từ sau để theo dõi hoạt động xuất nhập vật tư của mình:

**PHIẾU NHẬP**

Số phiếu:.....

Ngày:

Tên Nhà cung cấp:

Địa chỉ:

Điện thoại:

Mã kho:

Địa chỉ kho:

STT	Tên vật tư	Số lượng	Đơn vị tính	Đơn giá	Tỷ lệ VAT	Thành tiền
						Tổng cộng tiền

**PHIẾU XUẤT**

Số phiếu:.....

Ngày:

Tên khách hàng:

Địa chỉ:

Điện thoại:

Mã kho:

Địa chỉ kho:

STT	Tên vật tư	Số lượng	Đơn vị tính	Đơn giá	Tỷ lệ VAT	Thành tiền
						Tổng cộng tiền

*Bước 1: Liệt kê, chính xác hoá thông tin, chất lọc thông tin và mã hoá*

*Bảng 3.12. Danh sách thuộc tính cần quản lý*

Thực thể -Thuộc tính	Chính xác hoá dữ liệu
PHIẾU NHẬP	PHIEUNHAP
Số phiếu (nhập)	Sophieunhap
Ngày (nhập)	Ngaynhap
Tên Nhà cung cấp	TenNCC
+ Mã Nhà cung cấp (1)	MaNCC (thêm)
Địa chỉ (nhà cung cấp)	Đ/cNCC
Điện thoại (nhà cung cấp)	TeleNCC
Mã kho	Makho
Địa chỉ kho	Đ/ckho
Số thứ tự (lập) (2)	(bỏ)
Tên vật tư (lập)	Tenvattu*
+ Mã vật tư (1)	Mavattu* (thêm)
Số lượng (nhập) (lập)	Slnhap*
Đơn vị tính (lập)	Đ/Vtinh*
Đơn giá (nhập) (lập)	Đongianhap*
Tỷ lệ VAT (lập)	TyleVAT
Thành tiền (lập)	(bỏ)
Tổng tiền (nhập) (5)	(bỏ)
PHIẾU XUẤT	PHIEUXUAT
Số phiếu (xuất)	Sophieuxuat
Ngày (xuất)	Ngayxuat
+ Mã khách hàng (1)	Makhach (thêm)
Tên khách hàng	Tenkhach
Địa chỉ (khách)	Đ/ckhach
Điện thoại (khách)	Telekhach
Mã kho	Makho
Địa chỉ kho (4)	(bỏ)
Số thứ tự (lập) (2)	(bỏ)
Tên vật tư → Mã vật tư (3)	Mavattu*
Số lượng (xuất) (lập)	Slxuat*



Đơn vị tính (lập) (4)	(bỏ)
Đơn giá (xuất) (lập)	Dongiaxuat*
Tỷ lệ VAT (lập) (4)	(bỏ)
Thành tiền (lập)	(bỏ)
Tổng tiền (xuất) (5)	(bỏ)

*Chú thích:*

1. “Tên nhà cung cấp” là thuộc tính tên gọi của thực thể “Nhà cung cấp”, không xác định mỗi nhà cung cấp cụ thể vì Nhà cung cấp có thể trùng tên. Vì vậy phải thêm thuộc tính “Mã Nhà cung cấp” (MaNCC) làm thuộc tính định danh cho “Nhà cung cấp”. Tương tự thêm “Mã vật tư” và “mã khách hàng”

2. Số thứ tự để đánh số mỗi dòng hoá đơn chỉ có ý nghĩa trong hoá đơn này, không cho thông tin gì khác, không cần lưu trữ.

3. Thuộc tính tên gọi, đã có thuộc tính mã hoá đi kèm, đã gặp một lần ở trên nên bỏ đi. Nếu không có thuộc tính mã hoá đi kèm thì thay bằng mã hoá.

4. Các thuộc tính mô tả, đã gặp một lần ở trên nên bỏ đi.

5. Thuộc tính *thành tiền* = (*đơn giá giao*) x (*số lượng giao*), hai thuộc tính này đã được chọn ở trên ta có thể tính được “*thành tiền*” nên không cần lưu trữ thuộc tính này. Tương tự như vậy, thuộc tính tổng tiền cũng không cần lưu trữ.

Thuộc tính có dấu (\*) biểu thị các thuộc tính lập.

*Bước 2: Xác định các thuộc tính cần quản lý và các phụ thuộc hàm*

Theo bảng liệt kê trên ta có tập các thuộc tính cần quản lý với các phụ thuộc hàm như sau:

$R1 = \{Sophieunhap, Ngaynhap, MaNCC, TenNCC, Đ/cNCC, TeleNCC, Makho, Đ/ckho, Mavattu^*, Tenvattu^*, Slnhap^*, Đ/v\ tính^*, Đongianhap^*, TyleVAT^*\}$

$$F1 = \begin{cases} \text{Sophieunhap} \rightarrow \text{Ngaynhap, MaNCC, TenNCC, Đ/cNCC,} \\ \quad \text{TeleNCC, Makho, Đ/ckho.} \\ \text{Mavattu} \rightarrow \text{Tenvattu, Đ/vtinh, TyleVAT} \\ \text{Sophieunhap, Mavattu} \rightarrow \text{Sinhap, Dongianhap} \end{cases}$$

$R2 = \{\text{Sophieuxuat, Ngayxuat, Makhach, tenkhach, Đ/ckhach, telekhach, makho, Mavattu}^*, \text{Slxuat}^*, \text{dongiaxuat}^*\}$

$$F2 = \begin{cases} \text{Sophieuxuat} \rightarrow \text{Ngayxuat, Makhach, Tenkhach,} \\ \quad \text{Đ/ckhach, Telekhach, Makho} \\ \text{Sophieuxuat, Mavattu} \rightarrow \text{SLxuat, Dongiaxuat} \end{cases}$$

Từ đây, ta xác định được khoá trong lược đồ quan hệ tương ứng cho R1 và R2 như sau:

PHIEUNHAP (Sophieunhap, Ngaynhap, MaNCC, TenNCC, Đ/cNCC, TeleNCC, Makho, Đ/ckho, Mavattu<sup>\*</sup>, Tenvattu<sup>\*</sup>, Sinhap<sup>\*</sup>, Đ/v tính<sup>\*</sup>, Dongianhap<sup>\*</sup>, TyleVAT<sup>\*</sup>)

PHIEUXUAT (Sophieuxuat, Ngayxuat, Makhach, Tenkhach, Đ/ckhach, Telekhach, Makho, Mavattu<sup>\*</sup>, Slxuat<sup>\*</sup>, dongiaxuat<sup>\*</sup>)

vì bao đóng  $\{\text{Sophieunhap, Mavattu}\}^+ = R1$

và bao đóng  $\{\text{Sophieuxuat, Mavattu}\}^+ = R2$

Khoá trong hai lược đồ chính là khoá tối thiểu.

### Bước 3: Tiến hành chuẩn hoá

Việc tiến hành chuẩn hoá sẽ dựa theo các quy tắc đã nêu trong các chương trước.

Nhận thấy rằng hai lược đồ:

PHIEUNHAP (Sophieunhap, Ngaynhap, MaNCC, TenNCC, Đ/cNCC, TeleNCC, Makho, Đ/ckho, Mavattu<sup>\*</sup>, Tenvattu<sup>\*</sup>, Sinhap<sup>\*</sup>, Đ/v tính<sup>\*</sup>, Dongianhap<sup>\*</sup>, TyleVAT<sup>\*</sup>)

PHIEUXUAT (Sophieuxuat, Ngayxuat, Makhach, tenkhach, Đ/ckhach, telekhach, makho, Mavattu<sup>\*</sup>, Slxuat<sup>\*</sup>, dongiaxuat<sup>\*</sup>)

chưa đạt chuẩn 1 vì còn có thuộc tính lặp.

Để đạt chuẩn 1NF ta tách chúng thành các lược đồ con:

PHIEUNHAP1 (Sophieunhap, Mavattu, Tenvattu, Slnhap, Đ/v tính, Dongianhap, TyleVAT)

PHIEUNHAP2 (Sophieunhap, Ngaynhap, MaNCC, TenNCC, Đ/cNCC, TeleNCC, Makho, Đ/ckho)

PHIEUXUAT1 (Sophieuxuat, Mavattu, Slxuat, Dongiaxuat)

PHIEUXUAT2 (Sophieuxuat, Ngayxuat, Makhach, tenkhach, Đ/ckhach, telekhach, makho)

*Xét chuẩn 2:*

PHIEUNHAP1 (Sophieunhap, Mavattu, Tenvattu, Slnhap, Đ/v tính, Dongianhap, TyleVAT)

Có Mavattu → Tenvattu, Đ/v tính, TyleVAT), mà Mavattu là một phần của khoá nên PHIEUNHAP1 không đạt 2NF. Để đạt 2NF ta tách nó thành các lược đồ con:

(1) VATTU (Mavattu, Tenvattu, Đ/v tính, TyleVAT)

(2) DONGNHAP (Sophieunhap, Mavattu, Slnhap, Dongianhap,)

PHIEUNHAP2 (Sophieunhap, Ngaynhap, MaNCC, TenNCC, Đ/cNCC, TeleNCC, Makho, Đ/ckho)

PHIEUNHAP2 đương nhiên đạt chuẩn 2NF vì nhóm thuộc tính khoá chỉ có một thuộc tính. Vậy:

(3) PHIEUNHAP2 (Sophieunhap, Ngaynhap, MaNCC, TenNCC, Đ/cNCC, TeleNCC, Makho, Đ/ckho)

(4) DONGXUAT (Sophieuxuat, Mavattu, Slxuat, Dongiaxuat)

đạt 2NF vì không tồn tại thuộc tính ngoài khoá nào phụ thuộc vào một phần của khoá cả.

PHIEUXUAT2 (Sophieuxuat, Ngayxuat, Makhach, Tenkhach, Đ/ckhach, telekhach, makho)

PHIEUXUAT2 đương nhiên đạt chuẩn 2NF vì nhóm thuộc tính khoá chỉ có một thuộc tính. Vậy:

(5) PHIEUXUAT2 (Sophieuxuat, Ngayxuat, Makhach, Tenkhach, Đ/ckhach, Telekhach, Makho)

Kết quả ta có hệ lược đồ sau đạt chuẩn 2NF:

(1) VATTU (Mavattu, Tenvattu, Đ/v tính, TyleVAT)

(2) DONGNHAP (Sophieunhap, Mavattu, Slnhap, Dongianhap)

(3) PHIEUNHAP2 (Sophieunhap, Ngaynhap, MaNCC, TenNCC, Đ/cNCC, TeleNCC, Makho, Đ/ckho)

(4) DONGXUAT (Sophieuxuat, Mavattu, SLxuat, Dongiaxuat)

(5) PHIEUXUAT2 (Sophieuxuat, Ngayxuat, Makhach, Tenkhach, Đ/ckhach, Telekhach, Makho)

*Xét chuẩn 3:*

Các quan hệ (1), (2), (4) đương nhiên đạt chuẩn 3NF vì không có thuộc tính cầu

(1) VATTU (Mavattu, Tenvattu, Đ/v tính, TyleVAT)

(2) DONGNHAP (Sophieunhap, Mavattu, Slnhap, Dongianhap)

(4) DONGXUAT (Sophieuxuat, Mavattu, SLxuat, Dongiaxuat)

Có quan hệ (3):

(3) PHIEUNHAP2 (Sophieunhap, Ngaynhap, MaNCC, TenNCC, Đ/cNCC, TeleNCC, Makho, Đ/ckho)

MaNCC → TenNCC, Đ/cNCC, TeleNCC

Makho → Đ/ckho

Mà MaNCC và Makho là các thuộc tính ngoài khoá nên nó chính là các thuộc tính cầu. Vậy quan hệ này không đạt 3NF.

Để quan hệ này đạt được 3NF ta tách nó thành các quan hệ con:

(3.1) NHACC (MaNCC, TenNCC, Đ/cNCC, TeleNCC)

(3.2) KHO (Makho, Đ/ckho)

(3.3) PHIEUNHAP (Sophieunhap, Ngaynhap, MaNCC, Makho)

Có quan hệ 5:

(5) PHIEUXUAT2 (Sophieuxuat, Ngayxuat, Makhach, Tenkhach, Đ/ckhach, Telekhach, Makho)

Makhach → Tenkhach, Đ/ckhach, Telekhach

Mà Makhach là thuộc tính ngoài khoá nên nó chính là thuộc tính cầu. Vậy quan hệ này không đạt 3NF.

Để quan hệ này đạt được 3NF ta tách nó thành các quan hệ con:

(5.1) KHACH (Makhach, Tenkhach, Đ/ckhach, Telekhach)

(5.2) PHIEUXUAT (Sophieuxuat, Ngayxuat, Makhach, Makho)

*Tổng hợp kết quả trên, ta có hệ lược đồ sau đạt chuẩn 3:*

(1) VATTU (Mavattu, Tenvattu, Đ/v tính, TyleVAT)

(2) DONGNHAP (Sophieunhap, Mavattu, Slnhap, Đongianhap)

(3) NHACC (MaNCC, TenNCC, Đ/cNCC, TeleNCC)

(4) KHO (Makho, Đ/ckho)

(5) PHIEUNHAP (Sophieunhap, Ngaynhap, MaNCC, Makho)

(6) DONGXUAT (Sophieuxuat, Mavattu, SLxuat, Đongiaxuat)

(7) KHACH (Makhach, Tenkhach, Đ/ckhach, Telekhach)

(8) PHIEUXUAT (Sophieuxuat, Ngayxuat, Makhach, Makho)

*Bước 4: Vẽ sơ đồ E-R*

Sơ đồ E-R được vẽ theo nguyên tắc sau:

- Mỗi lược đồ được biểu diễn bằng một hình chữ nhật gồm tên (viết hoa) và các thuộc tính, thuộc tính khoá được gạch chân hoặc in đậm, khoá liên kết/khoá ngoại được đánh dấu ✓ hoặc in nghiêng hoặc dùng chữ FK (foreign key).

- Xác định liên kết giữa 2 thực thể: chúng có liên kết nếu đảm bảo đủ 2 dấu hiệu sau:

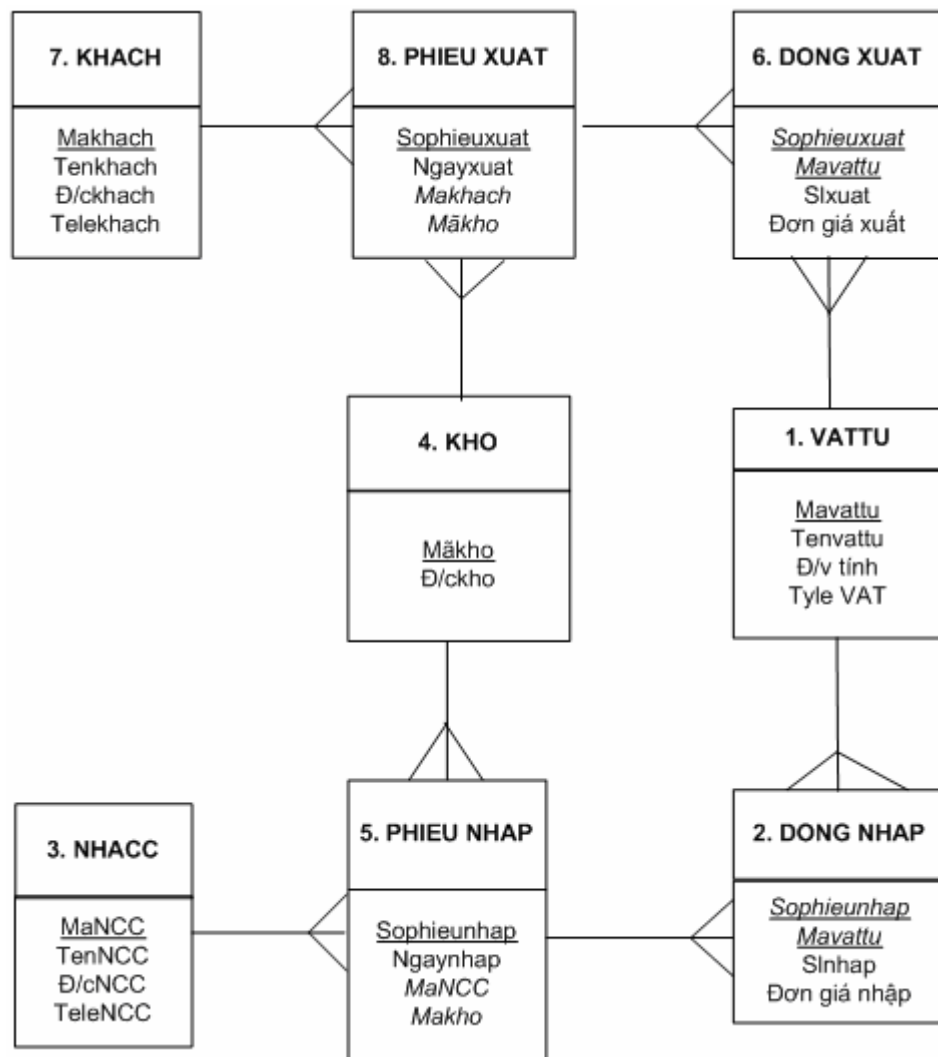
+ Có nhóm thuộc tính chung

+ Tại một thực thể nào đó, nhóm thuộc tính chung đó là khoá chính.

Như vậy nhóm thuộc tính ở thực thể còn lại sẽ chính là khoá ngoại/khoá liên kết

- Khi đã xác định được khoá liên kết ta có thể vẽ đường nối giữa chúng và đường ba chẽ sẽ dính vào thực thể có chứa khoá liên kết.

Kết quả vận dụng các nguyên tắc trên thu được sơ đồ E-R:



Thuộc tính khoá được gạch chân, khoá liên kết/khoá ngoại được in nghiêng.

Hình 3.23. Sơ đồ E-R bài toán xuất nhập vật tư trong kho

### 3.4. THIẾT KẾ CSDL MỨC VẬT LÝ

#### 3.4.1. Chuyển thiết kế dữ liệu mức logic sang mức vật lý

Có hai cách chuyển từ mức logic sang mức vật lý:

- Sử dụng một ngôn ngữ lập trình nào đó
- Sử dụng một hệ quản trị CSDL.

Ví dụ theo cách thứ nhất ta có thể dùng một ngôn ngữ lập trình nào đó ví dụ như TURBO PASCAL với các kiểu dữ liệu FILE và RECORD để mô tả các bản ghi và dùng các thủ tục READ, WRITE, SEEK, POS... để sinh các thao tác cập nhật và truy nhập bản ghi. Cách làm như vậy hiệu quả rất thấp. Ngày nay ít người dùng phương pháp thứ nhất vì nó đòi hỏi người lập trình phải thực hiện toàn bộ các chức năng tạo lập cấu trúc dữ liệu, truy nhập, cập nhật và xử lý câu hỏi.

Giải pháp tốt nhất là nên sử dụng một hệ quản trị CSDL sẵn có. Như đã biết hầu hết các hệ quản trị CSDL đều cung cấp các phương tiện mô tả dữ liệu, thao tác, đảm bảo toàn vẹn dữ liệu và tìm kiếm. Với các hệ quản trị CSDL, một khi đã thiết kế xong mức logic thì hãy sử dụng các công cụ thích hợp của hệ QTCSDL để tạo hệ thống ở mức vật lý. Các hệ CASE (Computer Aided SoftWare Engineering) ngày nay còn cho phép tự động sinh hệ thống (tạo mức vật lý) từ mức logic.

#### 3.4.2. Đánh giá không gian lưu trữ

Mặc dù đã có phần mềm tổ chức CSDL và phương pháp truy nhập nhưng người phân tích và thiết kế vẫn phải quan tâm đến một công việc khác có liên quan đến mức vật lý là đánh giá không gian lưu trữ cần thiết và tốc độ truy nhập. Điều này cần được xem là một phần của công việc thiết kế vì nhờ đó mới xác định được các quy tắc kỹ thuật liên quan đến khả năng lưu trữ của hệ thống và phần nào có liên quan đến thời gian trả lời hệ thống. Rõ ràng là mức logic về dữ liệu có liên quan trực tiếp đến các yếu tố không gian lưu trữ và thời gian trả lời của hệ thống.

Với mô hình CSDL được chọn là mô hình quan hệ thì ta phải tính toán được không gian cần thiết theo số bản ghi, độ dài các tệp phụ (như tệp chỉ dẫn).

Ví dụ ta cần tính không gian lưu trữ cho hệ thống thông tin quản lý kho phụ tùng đã nêu trong ví dụ trước đây của một nhà máy chỉ gồm một kho. Ta nhắc lại mô hình logic của hệ thống với sự chi tiết hoá về kiểu và độ dài của thuộc tính. Ta quy ước kiểu chữ là A, kiểu số là N, kiểu ngày là D: số để trong ngoặc đơn là độ dài tính theo byte. Khi đó có thể ước tính không gian cần thiết cho năm đầu tiên như trong bảng. Cũng phải tính không gian cần thiết cho các tệp index. Ngoài ra còn ước tính độ tăng trưởng của CSDL sau mỗi chu kỳ thời gian (tháng hay năm) để có một khuyến cáo kỹ thuật cho phía chủ đầu tư.

Với mô hình CSDL là mạng thì ta phải tính không gian đĩa không chỉ cho các bản ghi mà còn phải tính không gian cần thiết cho các liên kết. Để làm việc này ta cần ước tính mật độ trung bình của các liên kết đối với mỗi bản ghi chủ. Mỗi bản ghi chủ ngoài không gian cần thiết cho chính nó còn phải cộng thêm không gian cần thiết cho các con trỏ móc nối với các bản ghi thành viên. Ta có thể nghĩ rằng nên tính không gian tối đa căn cứ vào bản ghi số của bản ghi chủ trong mối quan hệ. Điều này không phải bao giờ cũng làm được nếu bản số này là 1: n với n không xác định.

*Bảng 3.13. Bảng đánh giá không gian lưu trữ*

Quan hệ	Thuộc tính	Độ dài bản ghi	Số bản ghi	Độ dài tổng
HÀNG (mỗi loại hàng chỉ để ở 1 kho)	Mã hàng A(12)	80	3000	240000
	Tên hàng A(20)			
	Đơn vị A (12)			
	Đơn giá N(8)			
	Tên kho A(20)			
	Tồn kho N(8)			
PHIEU_NHAP	Số_phiếu_nhập A(6)	20	60	1200
	Ngày_nhập D(8)		(Một năm)	
	Mã_NCC A(6)			
CAC_KHOAN_NHAP (Trung bình 50 khoản một phiếu nhập)	Số_phiếu_xuất A(6)	24	50x60	72000
	Mã hàng A(10)			
	Lượng nhập N(8)			



NHA_CUNG_CAP	Mã_nhà_cung_cấp A(5)	100	40	4000
	Tên_NCC A(45)			
	Địa_chỉ_NCC A(50)			
PHIEU_XUAT	Số_phiếu_xuất A(6)	20	1200	2400
	Ngày_xuất D(8)		(Một năm)	
	Mã_khách A(6)			
CAC_KHOAN_XUAT (Trung bình 2 khoản một phiếu xuất)	Số_phiếu_xuất A(6)	14	2x1200	57600
	Mã hàng A(10)			
	Lượng xuất N(8)			
KHACH_HANG	Mã_khách A(5)	100	300	3000
	Tên_khách A(45)			
	Địa_chỉ_khách A(50)			
INDEX	(20 byte/một bản ghi)		10000	20000
TONG CONG				628800
Lượng tăng mỗi năm	(ước tính)			300000

### 3.4.3. Biến đổi các lược đồ khái niệm thành lược đồ vật lý

Trong giai đoạn phân tích, ta có các lược đồ khái niệm. Đó là các lược đồ dữ liệu theo mô hình E\_R hay mô hình quan hệ. *Sang giai đoạn thiết kế, cần biến đổi các lược đồ khái niệm thành các lược đồ vật lý*, tức là một cấu trúc lưu giữ thực sự của dữ liệu trong bộ nhớ ngoài. Cấu trúc này thường được chọn lựa trong số các dạng sau: các tệp tuần tự, các bảng hàm băm, các mạng có con trỏ, các cây, các quan hệ,... Mỗi cấu trúc trên đều có tính ưu việt và hạn chế riêng nên tùy hoàn cảnh cụ thể của bài toán mà chọn cho thích hợp. Sự cân nhắc dựa trên yêu cầu truy nhập dữ liệu của mỗi chức năng trong hệ thống đảm bảo nhanh và dễ sử dụng, dựa trên các đặc điểm và ràng buộc của cấu hình vật lý hệ thống (phần cứng và phần mềm sử dụng) sao cho thích ứng được với cấu hình đó.

Việc biến đổi các lược đồ khái niệm thành các lược đồ vật lý kéo theo việc thiết kế dữ liệu thành 2 bước:

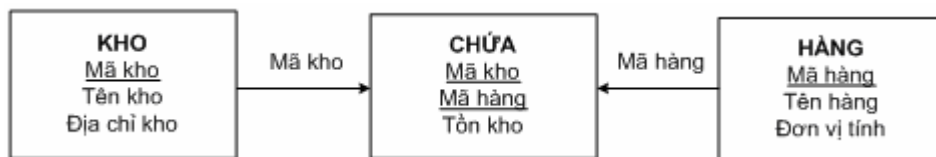
1. Thông qua việc nghiên cứu các yêu cầu truy nhập mà biến đổi lược đồ khái niệm thành một dạng trung gian gọi là *lược đồ logic* về dữ liệu.

2. Thông qua việc nghiên cứu cấu hình của hệ thống, đặc biệt là ngôn ngữ lập trình, các hệ quản trị CSDL đã được chọn dùng mà biến đổi lược đồ logic thành lược đồ vật lý thích hợp với cấu hình đó.

### 3.4.3.1. Thành lập lược đồ logic

Mọi cấu trúc lưu trữ vật lý, dù là tệp (tệp tuần tự, tệp tuần tự có chỉ dẫn, tệp trực truy, ...) hay là CSDL (mô hình mạng, mô hình phân cấp, mô hình quan hệ) thì cũng đều tạo nên từ các đơn vị cơ sở là các bản ghi (một bản ghi cấu tạo từ nhiều trường). Vì vậy người ta chọn *lược đồ logic*, một dạng trung gian trước khi đến lược đồ vật lý, là một *cấu trúc các kiểu bản ghi*. Mỗi kiểu bản ghi là một tập hợp những bản ghi có cấu trúc trường giống nhau, thường được gọi tắt là *bảng*. Bảng được biểu diễn bằng hình chữ nhật có tên bảng và danh sách các thuộc tính (trường). Giữa 2 bảng, thiết lập một kết nối, diễn tả bằng một mũi tên nếu 2 bảng đó có thuộc tính chung, và trong một bảng nào đó nhóm thuộc tính chung đó là khoá chính. Nhóm trường chung đó là điều kiện tiên quyết của kết nối, hơn nữa nhóm thuộc tính này chỉ có một giá trị duy nhất cho mỗi bản ghi trong bảng mà nhóm thuộc tính chung đó giữ vai trò khoá chính và ở gốc mũi tên (phía 1). Mũi tên sẽ gắn vào thực thể có khoá ngoại (phía Nhiều).

Ví dụ cấu trúc các kiểu bản ghi (lược đồ logic) như sau:



Hình 3.24. Lược đồ logic

Mục đích thiết kế logic: xuất phát từ lược đồ khái niệm, tìm một cấu trúc các kiểu bản ghi phù hợp với yêu cầu truy nhập của các chức năng xử lý trong hệ thống.

### 3.4.3.2. Thành lập lược đồ vật lý

Lược đồ vật lý là cấu trúc lưu trữ thực sự của dữ liệu trong bộ nhớ ngoài, phụ thuộc vào cấu hình của hệ thống (các ngôn ngữ lập trình, các hệ quản trị CSDL,...). Có 2 phương án chọn lựa chính là các tệp và CSDL, theo đó mà ta phải chuyển đổi lược đồ logic thu được thành lược đồ vật lý thích hợp.

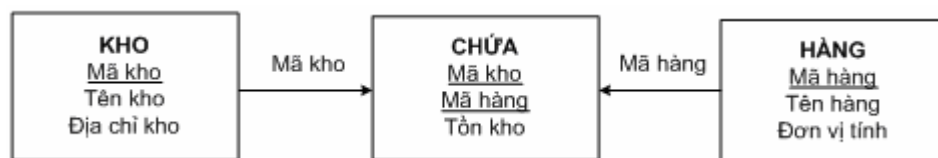
Điểm khác biệt cơ bản giữa tệp và CSDL là: đối với tệp, người dùng phải trực tiếp thành lập nó và khai thác nó, nghĩa là phải hiểu rõ và chịu trách nhiệm trực tiếp về nó. Như vậy, chương trình và dữ liệu ràng buộc chặt với nhau. Còn đối với CSDL, sự có mặt của hệ quản trị CSDL đã cách ly chương trình người dùng với cấu trúc lưu trữ của dữ liệu làm cho chương trình có ưu điểm là độc lập tương đối với dữ liệu. Đổi lại thì các ngôn ngữ định nghĩa dữ liệu cũng như các ngôn ngữ thao tác dữ liệu cung cấp bởi hệ quản trị CSDL lại phải nhúng được vào ngôn ngữ lập trình được chọn để cài đặt hệ thống, điều này không phải lúc nào cũng thuận tiện.

Đối với thiết kế CSDL, Cấu trúc lưu trữ vật lý phụ thuộc theo mô hình của CSDL, ví dụ như mô hình quan hệ:

Mỗi bảng trong lược đồ logic chuyển đổi trực tiếp thành một quan hệ (mỗi trường thành một thuộc tính của quan hệ). Không có gì cài đặt đặc biệt đối với các kết nối giữa các bảng, ngoại trừ sự có mặt của khoá ngoại. Một ngôn ngữ định nghĩa dữ liệu cho phép định nghĩa các quan hệ. Các lệnh trong một ngôn ngữ thao tác dữ liệu cho phép lưu trữ và tìm kiếm các dữ liệu dựa trên các phép toán của đại số quan hệ (thay cho sự lần dò theo các con trỏ).

### 3.4.3.3. Ví dụ

Dựa theo ví dụ trên, từ lược đồ logic (hình 3.24).



ta chuyển về cấu trúc lưu trữ:

#### KHO

Mã kho	Tên kho	Địa chỉ kho

#### CHUA

Mã kho	Mã hàng	Tồn kho

#### HANG

Mã hàng	Tên hàng	Đơn vị tính

Nếu biểu diễn dưới dạng có bổ sung liên kết giữa khoá chính và khoá ngoại của hai quan hệ bằng mũi tên như hình 3.26, chúng ta sẽ có được một *sơ đồ CSDL mức logic*:

#### KHO

Mã kho	Tên kho	Địa chỉ kho

#### CHUA

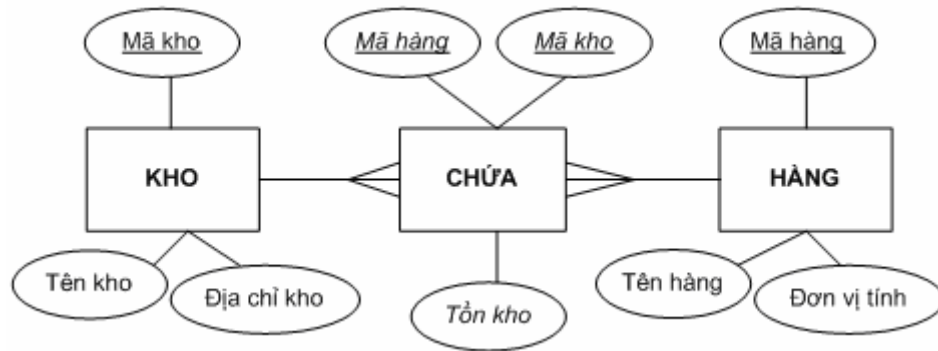
Mã kho	Mã hàng	Tồn kho

#### HANG

Mã hàng	Tên hàng	Đơn vị tính

Hình 3.26. Sơ đồ CSDL mức logic

Trong phương pháp hiện đại, người ta dùng luôn mô hình CSDL logic - sơ đồ E\_R (ERD) để làm căn cứ thiết kế vật lý:



Hình 3.27. Sơ đồ E\_R của bài toán ví dụ Kho - hàng

Và tiến hành thiết kế vật lý như sau:

Bảng 3.14. KHO

Tên trường	Kiểu	Độ dài	Khuôn dạng	Ràng buộc
<b>Mã kho</b>	Ký tự	6	Chữ hoa + số	Khoá chính NOTNULL
Tên kho	Ký tự	5	Chữ hoa	
Địa chỉ kho	Ký tự	30	Chữ đầu viết hoa	

Bảng 3.15. CHỨA

Tên trường	Kiểu	Độ dài	Khuôn dạng	Ràng buộc
<b>Mã kho</b>	Ký tự	6	Chữ hoa + số	Thành phần của khoá chính NOTNULL Khoá ngoại liên kết với KHO
<b>Mã hàng</b>	Ký tự	6	Chữ hoa + số	Thành phần của khoá chính NOTNULL Khoá ngoại liên kết với HÀNG
Tồn kho	Số	9	Số nguyên	

Bảng 3.16. HÀNG

Tên trường	Kiểu	Độ dài	Khuôn dạng	Ràng buộc
<b>Mã hàng</b>	Ký tự	6	Chữ hoa + số	Khoá chính NOTNULL
Tên hàng	Ký tự	15	Chữ đầu viết hoa	
Đơn vị tính	Ký tự	10	Chữ thường	

### 3.5. THIẾT KẾ AN TOÀN BẢO MẬT CHO CSDL

#### 3.5.1. An toàn thông tin và an toàn hệ thống

Mục đích của công tác an toàn và bảo mật dữ liệu là đảm bảo tính chính xác, độ tin cậy cao và phục hồi được trong các trường hợp phát sinh sự cố do một số nguyên nhân chính sau:

- NSD không thành thạo vô tình sửa, xóa số liệu tùy tiện làm mất mát, sai lạc số liệu.
- Mất điện đột ngột khi hệ thống đang tiến hành tính toán hoặc cập nhật làm hỏng tệp dữ liệu.
- Hỏng đĩa cứng hoặc máy làm tê liệt hoàn toàn bộ số liệu và chương trình.
- Cháy, nổ, va đập mạnh làm hỏng máy móc, chương trình và số liệu.
- Kẻ gian lấy trộm hệ thống máy tính, hoặc thâm nhập trái phép xóa bỏ hệ thống chương trình và số liệu, việc truyền số liệu không đảm bảo an toàn.
- Các nguyên nhân khách quan khác,...

Trong trường hợp xảy ra sự cố kỹ thuật đối với máy móc, chương trình hoặc số liệu, bản sao lưu có thể được phục hồi trở lại trên một hệ thống máy khác đã cài lại chương trình và sẵn sàng hoạt động như một hệ thống dự phòng. Nếu tiến hành sao lưu thường xuyên hàng ngày thì bản sao lưu sẽ sai khác bản dữ liệu đã mất chỉ có một ngày thao tác nghiệp vụ thay vì phải nhập lại một khối lượng số liệu lớn.

#### 3.5.2. Các biện pháp thực hiện an toàn dữ liệu

*Thiết lập các nguyên tắc quản lý:*

- Đối với máy chủ: Mọi thao tác trên máy chủ đều nằm dưới sự kiểm soát của người phụ trách trung tâm CSDL thông qua các cán bộ kỹ thuật được giao nhiệm vụ cụ thể. Các cán bộ kỹ thuật phải tuân thủ nghiêm ngặt chế độ sao lưu và kiểm tra dữ liệu định kỳ trên các thiết bị mang tin.

- Đối với các máy trạm: Phải giao trách nhiệm sao lưu và kiểm tra dữ liệu của từng nghiệp vụ cho các cán bộ cụ thể. Các cán bộ này sẽ được hướng dẫn cơ chế sao lưu, kiểm tra dữ liệu định kỳ trên các vật mang tin và phục hồi dữ liệu khi có sự cố.

- Tăng cường công tác kiểm tra, kiểm soát và hạn chế đến mức cao nhất việc đưa thông tin từ bên ngoài vào hệ thống thông qua đĩa mềm để giảm thiểu khả năng lây nhiễm virus lên hệ thống.

*Sử dụng các thiết bị lưu trữ an toàn:*

Dùng các đĩa lưu trữ, các ổ CD, các Tape Backup (sao lưu băng từ)... Các thiết bị này dùng để lưu trữ song song, đề phòng mất mát dữ liệu khi có sự cố xảy ra. Tuy nhiên, đối với hệ CSDL của cơ quan chủ quản và hệ thống chính có dung lượng lưu trữ lớn và phát sinh thường xuyên cần có cơ chế sao lưu nhanh với sự hỗ trợ của phần mềm ứng dụng.

Tuỳ theo khả năng đánh giá về tầm quan trọng khác nhau của số liệu ta có thể chọn chu kỳ sao lưu là ngày, vài ngày, tuần hoặc vài tuần, tháng...

*Sử dụng các trang thiết bị an toàn:*

- Thiết lập cơ chế Clustering (cơ chế quản lý CSDL nhân đôi song song) hoàn chỉnh và có độ tin cậy cao.

- Sử dụng hệ thống lưu điện tổng thông minh.

- Lựa chọn các máy chủ với khả năng hỗ trợ RAID và thiết bị nguồn cung cấp có khả năng chuyển đổi nóng.

Ngoài ra, việc tuân thủ chế độ bảo trì thiết bị định kỳ, thường xuyên cập nhật các phần mềm chống virus là hết sức cần thiết.

### **3.5.3. Bảo mật thông tin**

*Thiết lập cơ chế bảo mật*

- Ban hành qui chế bảo mật trong toàn bộ hệ thống

- Qui định chức năng quyền hạn truy xuất cho NSD đối với dữ liệu trong hệ thống

- Trách nhiệm của quản trị hệ thống trong việc phân quyền truy nhập vào hệ thống

Những hệ thống CSDL điển hình thường có những thủ tục truy xuất riêng. Để có thể thâm nhập hệ thống, NSD phải được cấp một tài khoản (user account) trên hệ thống đó và bắt buộc khai báo đúng tên và mật khẩu của mình. NSD được quản trị viên CSDL (Database Administrator) cấp quyền xem một hiển thị (view) của CSDL. NSD chỉ được phép sử dụng một số màn hình, một số lệnh để truy xuất dữ liệu trong một số tệp, một số bản ghi hoặc trường. Ở đây, quản trị viên CSDL là một cá nhân hoặc một nhóm chịu trách nhiệm chính trong công việc quản lý CSDL. Số những người sử dụng chương trình sẽ khác nhau đối với các CSDL khác nhau. Hệ thống luôn luôn lưu trữ lại một số lần truy nhập dữ liệu gần nhất của từng NSD.

Mỗi NSD phải có trách nhiệm ghi nhớ và bảo mật mật khẩu của mình trong quá trình sử dụng. Trong trường hợp có sự cố xảy ra, những người lãnh đạo sẽ căn cứ vào biên bản về lần truy cập gần nhất để thẩm vấn.

*Ví dụ:* Trong lĩnh vực quản lý kinh doanh sách trên mạng, thông thường, có 3 nhóm người dùng có nhu cầu truy nhập CSDL, đó là: khách mua, nhân viên cửa hàng, chủ cửa hàng. Ngoài ra, cũng cần có người quản trị hệ thống, chịu trách nhiệm đối với các hoạt động của cơ sở dữ liệu và có quyền truy nhập vào toàn bộ cơ sở dữ liệu.

Chủ cửa hàng có đầy đủ các quyền trên tất cả các quan hệ, tuy nhiên không được phép tạo dựng thêm các đối tượng dữ liệu mới. Do vậy, người quản trị sẽ phân quyền thao tác đầy đủ trên các đối tượng dữ liệu trong CSDL.

Khách mua chỉ được phép tìm kiếm trên quan hệ có liên quan và có thể đặt hàng trực tuyến, nhưng họ không được phép truy nhập đến các bản ghi của khách hàng khác cũng như các đơn đặt hàng khác. Do vậy chúng ta phải giới hạn theo 2 cách. Cách thứ nhất, phải sử dụng các đặc trưng an toàn của CSDL để giới hạn phạm vi truy xuất đến các dữ liệu nhạy cảm. Ví dụ như khi tạo lập một khoản mục mới cho



khách hàng, người quản trị CSDL sẽ tạo dựng luôn những khung nhìn chỉ cho phép anh ta xem các dữ liệu của riêng mình thông qua các khung nhìn đó. Cách thứ hai, chúng ta có thể giới hạn thông qua việc tạo các giao diện trong ứng dụng trên web. Và mọi truy vấn đều được gắn thêm giá trị Khách-id của khách mua hiện thời.

Nhân viên cửa hàng cũng tùy theo từng đối tượng mà được người quản trị phân cho họ các quyền thao tác trên từng đối tượng dữ liệu cụ thể. Ví dụ với người quản lý kho, anh ta chỉ được phép cập nhật trên quan hệ SACH. Với người quản lý các thông tin về khách hàng thì chỉ có thể thao tác trên quan hệ KHACH mà thôi. Còn đối với các người quản lý đơn hàng, họ có các quyền cập nhật trên quan hệ DON\_HANG\_MOI và DONG\_DON\_HANG\_MOI, nhưng với các quan hệ còn lại như SACH, KHACH thì chỉ có quyền xem mà thôi. Để hạn chế quyền truy nhập của khách mua, chúng ta có thể tạo ra một tài khoản có tên là khách\_mua và gán cho tài khoản này các quyền nào đó.

- *Tăng cường cơ chế kiểm soát đột nhập trên mạng*: sử dụng các công cụ mã hoá để bảo mật thông tin trên đường truyền, các chức năng về an toàn dữ liệu của hệ quản trị mạng và nên sử dụng giao thức SSL (Security Socket Layer).

- *Đảm bảo có hệ thống theo dõi tự động*: phần lớn CSDL, trên mainframe đều có những hệ thống theo dõi tự động nhằm ghi nhận mọi hoạt động có thể làm tổn hại đến CSDL.

- *Sử dụng trang thiết bị bảo mật ở mức vật lý*:

Lắp đặt các thiết bị bảo mật đầu cuối trong toàn bộ hệ thống

Trang bị hệ thống khoá bảo vệ an toàn.

## TÀI LIỆU THAM KHẢO

- [1]. Armstrong W.W. *Dependency Structure of Database Relationships*. Information Processing 74, Holland publ.Co (1974)
- [2]. Beerl C. Dowd M., Fagin R... *On the structure of Armstrong relational for Functional Dependencies* J.ACM 31,1 (1984)
- [3]. Codd E.F. *A relational model for large shared data banks*. Communication ACM 13 (1970)
- [4]. Collongues A., Hugues J., Laroche B. *MERISE method*. Science - technique Public 1994.
- [5]. Demetrovics J. *Logical and structural investigation of Relational Data model*. MTA-SZTAKI Tanulmanyok, Budapest, 114 (1980)
- [6]. Jeffrey D. Ullman, *Principles of Database and Knowledge base Systems*, 1988-1989. Biên dịch: Trần Đức Quang, Nhà xuất bản Thống kê, 1998.
- [7]. Peter Rob, Carlos Coronel, *Database System*. Wadsworth Publishing company, 1993.
- [8]. Nguyễn Kim Anh, *Nguyên lý của các hệ cơ sở dữ liệu*. Nhà xuất bản Đại học Quốc gia Hà Nội, 2004.
- [9]. Đề án 112. *Giáo trình phân tích, thiết kế và xây dựng các hệ thống cơ sở dữ liệu*. Hà Nội, 2004.
- [10]. Nguyễn Địch, Bùi Công Cường, Lê Văn Phùng, Thái Thanh Sơn. *Lý thuyết hệ thống và điều khiển học*. Nhà xuất bản Thông tin và Truyền thông, 2009.
- [11]. Hồ Sĩ Khoa, *Các phương pháp xây dựng các mô hình khái niệm dữ liệu*, 1990.

- [12]. Lê Văn Phùng, *Bài giảng cơ sở dữ liệu*. Nhà xuất bản Lao động-Xã hội, 2004.
- [13]. Lê Văn Phùng, *Phân tích và thiết kế hệ thống thông tin - Kiến thức và thực hành*. Nhà xuất bản Lao động-Xã hội, 2004.
- [14]. Lê Văn Phùng, *Kỹ thuật phân tích và thiết kế hệ thống thông tin hướng cấu trúc*. Nhà xuất bản Thông tin và Truyền thông, 2009.
- [15]. Trần Đức Quang, *Quản trị cơ sở dữ liệu*. Nhà xuất bản Xây dựng, 2002.
- [16]. Vũ Đức Thi, *Cơ sở dữ liệu - Kiến thức và thực hành*. NXB Thống kê, 1997.
- [17]. Vũ Đức Thi, *Thuật toán trong tin học*. Nhà xuất bản Khoa học Kỹ thuật, 1999.
- [18]. Nguyễn Quốc Toàn, Nguyễn Văn Vy, Vũ Đức Thi, Lê Văn Phùng, *Bài giảng Kỹ nghệ phần mềm*. Khoa Công nghệ, Đại học Quốc gia Hà Nội, 2000.
- [19]. Đỗ Trung Tuấn, *Database*. NXB Giáo dục, 1998.
- [20]. Đinh Mạnh Tường, *Cấu trúc dữ liệu và thuật toán*. Nhà xuất bản Khoa học Kỹ thuật, 2000.
- [21]. Nguyễn Bá Tường, *Cơ sở dữ liệu: Lý thuyết - thực hành*. Nhà xuất bản Khoa học Kỹ thuật, 2001.
- [22]. Trần Thành Trai, *Nhập môn cơ sở dữ liệu*. Nhà xuất bản Trẻ.
- [23]. Nguyễn Đăng Ty, Đỗ Phúc, *Giáo trình cơ sở dữ liệu*. Đại học Quốc gia TP. Hồ Chí Minh, 2003.
- [24]. Lê Tiến Vương, *Nhập môn cơ sở dữ liệu quan hệ*. Nhà xuất bản Khoa học Kỹ thuật, 1997.
- [25]. Nguyễn Văn Vy, Lê Văn Phùng, *Giáo trình Phân tích thiết kế hệ thống thông tin*. Hà Nội, 2000.

# MỤC LỤC

<i>Lời nói đầu</i> .....	3
<b>Chương 1. Tổng quan về cơ sở dữ liệu</b> .....	7
1.1. Các khái niệm cơ bản .....	7
1.1.1. Dữ liệu .....	7
1.1.2. Cơ sở dữ liệu .....	10
1.1.3. Hệ thống xử lý dữ liệu .....	10
1.1.4. Sự khác biệt giữa dữ liệu và thông tin .....	16
1.1.5. Siêu dữ liệu .....	17
1.1.6. Kiểu thực thể và bảng dữ liệu .....	18
1.1.7. Khái niệm toàn vẹn dữ liệu .....	26
1.1.8. Hệ quản trị CSDL (DBMS) .....	31
1.1.9. Kho dữ liệu .....	31
1.2. Các phương tiện diễn tả dữ liệu .....	32
1.2.1. Khái niệm diễn tả dữ liệu .....	32
1.2.2. Mã hoá .....	33
1.2.3. Từ điển dữ liệu .....	39
1.3. Mô hình dữ liệu .....	43
1.3.1. Khái niệm mô hình dữ liệu .....	43
1.3.2. Các loại mô hình CSDL .....	43
1.3.3. Mô hình dữ liệu mạng và mô hình dữ liệu phân cấp .....	44
1.3.4. Mô hình dữ liệu quan hệ và mô hình dữ liệu hướng đối tượng .....	46
1.3.5. Phương pháp xây dựng mô hình dữ liệu .....	48
<b>Chương 2. Mô hình cơ sở dữ liệu quan hệ</b> .....	51
2.1. Các khái niệm cơ bản .....	51

2.1.1. Định nghĩa quan hệ .....	51
2.1.2. Định nghĩa phụ thuộc hàm .....	53
2.1.3. Hệ tiên đề Armstrong .....	54
2.1.4. Định nghĩa hàm đóng .....	57
2.1.5. Định nghĩa sơ đồ quan hệ.....	58
2.1.6. Định nghĩa bao đóng .....	59
2.1.7. Định nghĩa khoá của quan hệ, sơ đồ quan hệ, họ F.....	61
2.1.8. Hệ bằng nhau.....	63
2.2. Một số thuật toán liên quan đến khoá.....	65
2.2.1. Một số thuật toán liên quan đến bao đóng .....	65
2.2.2. Một số thuật toán liên quan đến khoá .....	67
2.2.3. Thuật toán xác định các thuộc tính cơ bản.....	72
2.3. Các phép toán xử lý file dữ liệu .....	73
2.3.1. Phép hợp ( $r \cup t$ ) (Union) .....	74
2.3.2. Phép trừ ( $r - t$ ) (Difference) .....	74
2.3.3. Phép giao ( $r \cap t$ ) (Intersection) .....	75
2.3.4. Phép tích Đề-các ( $r \times t$ ) (Cartesian product).....	75
2.3.5. Phép chiếu (Projection) .....	76
2.3.6. Phép chọn (Restriction).....	77
2.3.7. Phép chia (Division) .....	77
2.3.8. Phép nối điều kiện 2 file dữ liệu (Join $\theta$ ) .....	78
2.3.9. Phép nối 2 file dữ liệu (Join) .....	79
2.3.10. Ví dụ áp dụng .....	80
2.3.11. Các chiến lược tối ưu tổng quát.....	83
2.4. Các dạng chuẩn và các thuật toán liên quan .....	86
2.4.1. Các định nghĩa chuẩn hoá .....	86
2.4.2. Một số đặc trưng dạng chuẩn 2NF .....	88
2.4.3. Một số đặc trưng dạng chuẩn 3NF .....	89
2.4.4. Một số đặc trưng dạng chuẩn BCNF .....	90

2.4.5. Các thuật toán.....	91
2.4.6. Chuẩn hoá một quan hệ.....	93
2.4.7. Chuẩn hoá một sơ đồ quan hệ .....	99
2.5. Mối quan hệ (Relationship) .....	116
2.5.1. Đặc tả mối quan hệ .....	116
2.5.2. Các đặc trưng của mối quan hệ .....	117
2.5.3. Mô hình khái niệm dữ liệu .....	119
<b>Chương 3. Quy trình phân tích thiết kế CSDL .....</b>	<b>121</b>
3.1. Phân tích yêu cầu .....	121
3.1.1. Mục đích của việc phân tích yêu cầu .....	121
3.1.2. Nội dung phân tích yêu cầu.....	123
3.1.3. Các kỹ thuật phân tích yêu cầu.....	124
3.2. Phân tích dữ liệu và xây dựng mô hình khái niệm dữ liệu .....	142
3.2.1. Khảo sát thực tế.....	142
3.2.2. Thiết lập mô hình dữ liệu .....	146
3.2.3. Kiểm soát và chuẩn hóa mô hình .....	163
3.3. Thiết kế CSDL logic.....	167
3.3.1. Khuôn cảnh thiết kế cơ sở dữ liệu logic .....	168
3.3.2. Các kỹ thuật sử dụng trong thiết kế cơ sở dữ liệu logic.....	170
3.3.3. Thiết kế cơ sở dữ liệu logic dựa trên “Phương pháp từ điển” ...	197
3.4. Thiết kế CSDL mức vật lý .....	207
3.4.1. Chuyển thiết kế dữ liệu mức logic sang mức vật lý.....	207
3.4.2. Đánh giá không gian lưu trữ .....	207
3.4.3. Biến đổi các lược đồ khái niệm thành lược đồ vật lý .....	209
3.5. Thiết kế an toàn bảo mật cho CSDL .....	214
3.5.1. An toàn thông tin và an toàn hệ thống.....	214
3.5.2. Các biện pháp thực hiện an toàn dữ liệu.....	214
3.5.3. Bảo mật thông tin .....	215
<i>Tài liệu tham khảo .....</i>	<i>219</i>

# CƠ SỞ DỮ LIỆU QUAN HỆ VÀ CÔNG NGHỆ PHÂN TÍCH - THIẾT KẾ

Chịu trách nhiệm xuất bản

NGUYỄN THỊ THU HÀ

Biên tập: NGÔ MỸ HẠNH  
TRỊNH THU CHÂU  
Trình bày sách: BÙI CHÂU LOAN  
Sửa bản in: TRỊNH THU CHÂU  
Thiết kế bìa: TRẦN HỒNG MINH

In 700 bản, khổ 16 x 24 cm tại Công ty CP In và Thương mại Ngọc Hưng  
Số đăng ký kế hoạch xuất bản: 237-2010/CXB/3-125/TTTT  
Số quyết định xuất bản: 72/QĐ-NXB TTTT ngày 31 tháng 3 năm 2010  
In xong và nộp lưu chiểu tháng 4 năm 2010.