

CƠ SỞ DỮ LIỆU



PHẦN

I



- Các khái niệm cơ bản
- Kiến trúc hệ thống cơ sở dữ liệu
- Mô hình quan hệ thực thể
- Mô hình quan hệ
- Chuẩn hoá quan hệ
- Thiết kế cơ sở dữ liệu vật lý

GV. Phạm Thị Hoàng Nhung
Bộ môn Công nghệ phần mềm
Đại học Thủy lợi

MỤC LỤC

1	Chương 1. CÁC KHÁI NIỆM CƠ BẢN	5
1.1	Tại sao phải có một cơ sở dữ liệu	5
1.2	Định nghĩa một cơ sở dữ liệu.....	5
1.2.1	Khái niệm.....	5
1.2.2	Ưu điểm.....	6
1.2.3	Vấn đề cần giải quyết.....	6
1.3	Hệ quản trị cơ sở dữ liệu (DataBase Management System_DBMS)	7
1.3.1	Ví dụ.....	7
1.3.2	Khái niệm.....	7
1.4	Hệ thống cơ sở dữ liệu (Database System)	8
1.5	Các đối tượng sử dụng CSDL.....	8
1.5.1	Đối tượng trực tiếp.....	8
1.5.2	Đối tượng gián tiếp	9
1.6	Lợi ích của việc sử dụng HQTCSDL	9
2	Chương 2. NHỮNG KHÁI NIỆM VÀ KIẾN TRÚC CỦA HỆ THỐNG CƠ SỞ DỮ LIỆU	11
2.1	Mô hình dữ liệu, lược đồ và trường hợp (Data Models, Schemas, Instances) 11	
2.1.1	Phân loại mô hình dữ liệu	11
2.1.2	Lược đồ(Schema) , minh hoạ (instances), và trạng thái (State).....	14
2.2	Lược đồ kiến trúc của hệ quản trị cơ sở dữ liệu (DBMS Architecture) và sự độc lập dữ liệu (Data Independence)	15
2.2.1	Lược đồ kiến trúc 3 mức của HQTCSDL.....	16
2.2.2	Độc lập dữ liệu.....	17
2.3	Ngôn ngữ của HQTCSDL.....	17
2.4	Các tính năng của HQTCSDL	17
2.5	Phân loại HQTCSDL	17
3	Chương 3. MÔ HÌNH QUAN HỆ - THỰC THỂ (Entity – Relationship Model)	19
3.1	Sử dụng mô hình dữ liệu khái niệm mức cao để thiết kế cơ sở dữ liệu...19	
3.2	Mục đích của mô hình khái niệm ER(Entity – Relationship Model)	20

3.3	Ví dụ về một cơ sở dữ liệu ứng dụng	20
3.4	Kiểu thực thể(Entity Type), Thuộc tính (Attributes), Khoá (Keys)	22
3.4.1	Thực thể (Entities) và thuộc tính (Attributes).....	22
3.4.2	Kiểu thực thể, Khoá và tập giá trị	25
3.5	Liên kết, Kiểu liên kết và các Ràng buộc liên kết.....	25
3.5.1	Định nghĩa liên kết và kiểu liên kết	25
3.5.2	Bậc của kiểu liên kết	26
3.5.3	Ràng buộc liên kết.....	27
3.6	Kiểu thực thể yếu(Weak Entity)	29
3.7	Tổng quát hóa và chuyên biệt hóa	29
3.7.1	Thực thể con và thực thể chính.....	30
3.7.2	Các thực thể con loại trừ	30
3.8	Các ký hiệu và quy ước đặt tên trong mô hình ER.....	31
3.8.1	Các ký hiệu.....	31
3.8.2	Quy tắc đặt tên	31
3.9	Xây dựng một mô hình ER.....	32
3.9.1	Các bước xây dựng sơ đồ ER.....	32
3.9.2	Mô hình ER cho cơ sở dữ liệu COMPANY	33
3.9.3	Bài tập	34
4	Chương 4. MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ.....	37
4.1	Khái niệm mô hình quan hệ	37
4.2	Các thành phần cơ bản của mô hình	37
4.2.1	Một số khái niệm của mô hình quan hệ	37
4.2.2	Quan hệ:	37
4.2.3	Các tính chất của một quan hệ	38
4.2.4	Các ràng buộc toàn vẹn trên quan hệ	38
4.2.5	Các phép toán trên CSDL quan hệ.....	41
5	Chương 5. CHUYỂN TỪ MÔ HÌNH ER SANG MÔ HÌNH QUAN HỆ	48
6	Chương 6. PHỤ THUỘC HÀM VÀ CHUẨN HOÁ QUAN HỆ	55
6.1	Một số hướng dẫn khi thiết kế cơ sở dữ liệu quan hệ	55
6.2	Phụ thuộc hàm(Functional Dependencies).....	56
6.2.1	Định nghĩa phụ thuộc hàm.....	56

6.2.2	Hệ tiên đề Armstrong.....	57
6.2.3	Bao đóng của tập phụ thuộc hàm.....	57
6.2.4	Bao đóng của tập thuộc tính X trên F.....	57
6.2.5	Khoá của quan hệ.....	58
6.2.6	Tập phụ thuộc hàm tương đương.....	59
6.2.7	Tập phụ thuộc hàm tối thiểu.....	59
6.3	Các dạng chuẩn của quan hệ.....	60
6.3.1	Định nghĩa các dạng chuẩn.....	60
6.3.2	Phép phân rã các lược đồ quan hệ.....	66
6.4	Chuẩn hoá quan hệ.....	70
6.4.1	Thuật toán phân rã lược đồ quan hệ thành các lược đồ quan hệ con ở BCNF 70	
6.4.2	Thuật toán phân rã một lược đồ quan hệ thành các lược đồ con ở 3NF. 72	
7	Chương 7. THIẾT KẾ CƠ SỞ DỮ LIỆU VẬT LÝ (Tham khảo).....	75
7.1	Nội dung thiết kế file vật lý và cơ sở dữ liệu vật lý.....	75
7.1.1	Quá trình thiết kế.....	75
7.1.2	Sản phẩm thiết kế.....	76
7.2	Thiết kế các trường.....	77
7.2.1	Yêu cầu thiết kế trường.....	77
7.2.2	Chọn kiểu và cách biểu diễn dữ liệu.....	78
7.3	Thiết kế các bản ghi vật lý.....	80
7.3.1	Phi chuẩn.....	80
7.3.2	Quản lý trường có độ dài cố định.....	81
7.3.3	Quản lý trường có độ dài biến đổi.....	81
7.4	Thiết kế file vật lý.....	82
7.4.1	Các loại file.....	82
7.4.2	Các phương pháp truy cập.....	82
7.4.3	Tổ chức file.....	83
7.4.4	Ví dụ về thiết kế file.....	87

1 Chương 1. CÁC KHÁI NIỆM CƠ BẢN

Trong nhiều năm gần đây, thuật ngữ Cơ sở dữ liệu - Database đã trở nên quen thuộc trong nhiều lĩnh vực. Các ứng dụng tin học vào quản lý ngày càng nhiều và đa dạng, hầu hết các lĩnh vực kinh tế, xã hội... đều đã ứng dụng các thành tựu mới của tin học vào phục vụ công tác chuyên môn của mình. Chính vì lẽ đó mà ngày càng nhiều người quan tâm đến thiết kế, xây dựng và ứng dụng cơ sở dữ liệu (CSDL).

Trong chương này, chúng ta sẽ tìm hiểu thế nào là cơ sở dữ liệu và các khái niệm liên quan đến nó. Trước hết, chúng ta sẽ tìm hiểu lý do tại sao cần phải quản lý dữ liệu bằng CSDL?

1.1 Tại sao phải có một cơ sở dữ liệu

Hệ thống các tệp tin cổ điển

Cho đến nay vẫn còn một số đơn vị kinh tế, hành chính sự nghiệp... sử dụng mô hình hệ thống các tệp tin cổ điển: chúng được tổ chức riêng rẽ, phục vụ cho một mục đích của một đơn vị hay một đơn vị con trực thuộc cụ thể.

- Ưu điểm:

Việc xây dựng hệ thống các tệp tin riêng tại từng đơn vị quản lý ít tốn thời gian bởi khối lượng thông tin cần quản lý và khai thác là nhỏ, không đòi hỏi đầu tư vật chất và chất xám nhiều, do đó triển khai ứng dụng nhanh.

Thông tin được khai thác chỉ phục vụ mục đích hẹp nên khả năng đáp ứng nhanh chóng, kịp thời.

- Nhược điểm:

Thông tin được tổ chức riêng rẽ ở nhiều nơi nên việc cập nhật dễ làm mất tính nhất quán dữ liệu.

Hệ thống thông tin được tổ chức thành các hệ thống file riêng lẻ nên thiếu sự chia sẻ thông tin giữa các nơi.

Qua phân tích trên, chúng ta nhận thấy việc tổ chức dữ liệu theo hệ thống tệp tin hoàn toàn không phù hợp với những hệ thống thông tin lớn. Việc xây dựng một hệ thống thông tin đảm bảo được tính nhất quán dữ liệu, đáp ứng được nhu cầu khai thác đồng thời của nhiều người là thực sự cần thiết.

1.2 Định nghĩa một cơ sở dữ liệu

1.2.1 Khái niệm

CSDL và công nghệ CSDL đã có những tác động to lớn trong việc phát triển sử dụng máy tính. Có thể nói rằng CSDL ảnh hưởng đến tất cả các nơi có sử dụng máy tính:

Kinh doanh (thông tin về sản phẩm, khách hàng, ...)

Giáo dục (thông tin về sinh viên, điểm, ..)

Thư viện (thông tin về tài liệu, tác giả, độc giả...)

Y tế (thông tin về bệnh nhân, thuốc....)...

Như vậy, cơ sở dữ liệu là gì?

CSDL là tập hợp các dữ liệu có cấu trúc và liên quan với nhau được lưu trữ trên máy tính, được nhiều người sử dụng và được tổ chức theo một mô hình.

Ví dụ:

Danh bạ điện thoại là một ví dụ về CSDL.

- Là các thông tin có ý nghĩa
- Là tập hợp các thông tin có cấu trúc.
- Các thông tin này có liên quan với nhau và có thể hệ thống được.

Trong khái niệm này, chúng ta cần nhấn mạnh, CSDL là tập hợp các thông tin có tính chất hệ thống, không phải là các thông tin rời rạc, không có liên quan với nhau. Các thông tin này phải có cấu trúc và tập hợp các thông tin này phải có khả năng đáp ứng nhu cầu khai thác của nhiều người sử dụng một cách đồng thời. Đó cũng chính là đặc trưng của CSDL.

1.2.2 Ưu điểm

Từ khái niệm trên, ta thấy rõ ưu điểm nổi bật của CSDL là:

Giảm sự trùng lặp thông tin xuống mức thấp nhất và do đó đảm bảo được tính nhất quán và toàn vẹn dữ liệu (Cấu trúc của cơ sở dữ liệu được định nghĩa một lần. Phần định nghĩa cấu trúc này gọi là meta-data, và được Catalog của HQTCSDB lưu trữ).

Đảm bảo sự độc lập giữa dữ liệu và chương trình ứng dụng (Insulation between programs and data): Cho phép thay đổi cấu trúc, dữ liệu trong cơ sở dữ liệu mà không cần thay đổi chương trình ứng dụng.

Trừu tượng hoá dữ liệu (Data Abstraction): Mô hình dữ liệu được sử dụng để làm ẩn lưu trữ vật lý chi tiết của dữ liệu, chỉ biểu diễn cho người sử dụng mức khái niệm của cơ sở dữ liệu.

Nhiều khung nhìn (multi-view) cho các đối người dùng khác nhau: Đảm bảo dữ liệu có thể được truy xuất theo nhiều cách khác nhau. Vì yêu cầu của mỗi đối tượng sử dụng CSDL là khác nhau nên tạo ra nhiều khung nhìn vào dữ liệu là cần thiết.

Đa người dùng (multi-user): Khả năng chia sẻ thông tin cho nhiều người sử dụng và nhiều ứng dụng khác nhau.

1.2.3 Vấn đề cần giải quyết

Để đạt được các ưu điểm trên, CSDL đặt ra những vấn đề cần giải quyết. Đó là:

Tính chủ quyền của dữ liệu: Do tính chia sẻ của CSDL nên chủ quyền của CSDL dễ bị xâm phạm.

Tính bảo mật và quyền khai thác thông tin của người sử dụng: Do có nhiều người được phép khai thác CSDL nên cần thiết phải có một cơ chế bảo mật và phân quyền hạn khai thác CSDL.

Tranh chấp dữ liệu: Nhiều người được phép cùng truy cập vào CSDL với những mục đích khác nhau: Xem, thêm, xóa hoặc sửa dữ liệu. Cần phải có cơ chế ưu

tiên truy cập dữ liệu hoặc giải quyết tình trạng xung đột trong quá trình khai thác cạnh tranh. Cơ chế ưu tiên có thể được thực hiện bằng việc cấp quyền (hay mức độ) ưu tiên cho từng người khai thác.

Đảm bảo dữ liệu khi có sự cố: Việc quản lý dữ liệu tập trung có thể làm tăng nguy cơ mất mát hoặc sai lệch thông tin khi có sự cố mất điện đột xuất hoặc đĩa lưu trữ bị hỏng. Một số hệ điều hành mạng có cung cấp dịch vụ sao lưu ảnh đĩa cứng (cơ chế sử dụng đĩa cứng dự phòng - RAID), tự động kiểm tra và khắc phục lỗi khi có sự cố. Tuy nhiên, bên cạnh dịch vụ của hệ điều hành, để đảm bảo an toàn cho CSDL, nhất thiết phải có một cơ chế khôi phục dữ liệu khi có sự cố xảy ra.

1.3 Hệ quản trị cơ sở dữ liệu (DataBase Management System_DBMS)

1.3.1 Ví dụ

Như chúng ta đã biết, kích thước và độ phức tạp của CSDL rất khác nhau.

Ví dụ:

Danh bạ điện thoại của một quốc gia, một thành phố.. chứa tới hàng triệu số và những thông tin cần thiết về khách hàng.

Trong trường đại học có tới hàng ngàn sinh viên. Nhà trường phải quản lý tất cả những thông tin liên quan đến sinh viên như: tên, ngày sinh, quê quán, địa chỉ, kết quả học tập...

Xét một Ví dụ về CSDL quản lý tài liệu và độc giả trong thư viện quốc gia. Giả sử rằng có 100 triệu cuốn sách, mỗi cuốn sách cần lưu 10 thông tin liên quan, mỗi thông tin chứa tối đa 400 kí tự thì CSDL sẽ phải có tối thiểu $100 * 10^6 * 400 * 10$ kí tự (bytes). Như vậy, dung lượng bộ nhớ cần dùng là: $100 * 10^6 * 400 * 10 = 400$ GB.

Ta thấy, bộ nhớ cũng là vấn đề cần phải được giải quyết. Tuy nhiên, vấn đề quan trọng hơn ở đây lại là cách thức tổ chức dữ liệu trong một cơ sở dữ liệu để phục vụ cho việc truy cập, tìm kiếm, cập nhật,... nhanh chóng và an toàn hơn.

Việc tổ chức dữ liệu như thế nào được thực hiện thông qua Hệ quản trị cơ sở dữ liệu(HQTCSDL).

Vậy hệ quản trị cơ sở dữ liệu (HQTCSDL) là gì?

1.3.2 Khái niệm.

HQTCSDL là tập hợp các phần mềm cho phép định nghĩa các cấu trúc để lưu trữ thông tin trên máy, nhập dữ liệu, thao tác trên các dữ liệu đảm bảo sự an toàn và bí mật của dữ liệu.

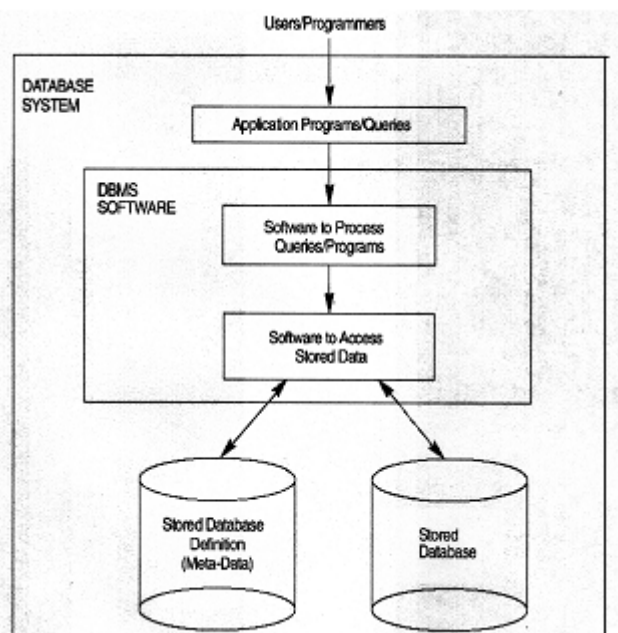
Định nghĩa cấu trúc: Định nghĩa cấu trúc CSDL bao gồm việc xác định kiểu dữ liệu, cấu trúc và những ràng buộc cho dữ liệu được lưu trữ trong CSDL.

Nhập dữ liệu: Là việc lưu trữ dữ liệu vào các thiết bị lưu trữ trung gian được điều khiển bằng HQTCSDL.

Thao tác dữ liệu: thao tác trên CSDL bao gồm những chức năng như truy xuất cơ sở dữ liệu để tìm kiếm thông tin cần thiết, cập nhật cơ sở dữ liệu và tổng hợp những báo cáo từ dữ liệu.

1.4 Hệ thống cơ sở dữ liệu (Database System)

Là phần mềm HQTCSDL cùng với dữ liệu của bản thân cơ sở dữ liệu đó.



Hình 1.1. Môi trường hệ thống cơ sở dữ liệu đơn giản

1.5 Các đối tượng sử dụng CSDL

Đối với các cơ sở dữ liệu nhỏ, mang tính cá nhân như lịch làm việc, danh bạ điện thoại cá nhân... thì chỉ cần một người để tạo ra và thao tác trên nó. Tuy nhiên, đối với các CSDL lớn như: quản lý tài chính của ngân hàng nhà nước, điều hành các chuyến bay cho các sân bay quốc tế... cần phải có rất nhiều người tham gia thiết kế, xây dựng, bảo trì CSDL và hàng trăm người sử dụng. Trong phần này, chúng ta tìm hiểu xem ai là người thao tác với CSDL hàng ngày. Và trong phần sau, chúng ta xem xét những người không trực tiếp tham gia một CSDL cụ thể, họ là người duy trì môi trường hệ thống CSDL.

1.5.1 Đối tượng trực tiếp

1.5.1.1 Quản trị cơ sở dữ liệu

Trong những tổ chức có nhiều người cùng sử dụng chung một nguồn dữ liệu thì nhất thiết phải có một người đứng đầu quản lý, chịu trách nhiệm đối với nguồn dữ liệu này. Đó chính là người quản trị cơ sở dữ liệu (Database Administrators _ DBA).

DBA có nhiệm vụ tổ chức nội dung của cơ sở dữ liệu, tạo và phân quyền cho người sử dụng, đưa ra yêu cầu về phần cứng và phần mềm... nếu cần thiết. DAB chịu trách nhiệm bảo vệ an toàn, Backup thông tin... khi có sự cố.

1.5.1.2 Thiết kế cơ sở dữ liệu

Người thiết kế CSDL chịu trách nhiệm:

- Xác định những dữ liệu nào cần lưu trữ trong CSDL
- Lựa chọn những cấu trúc thích hợp để biểu diễn và lưu trữ những dữ liệu này.

- Phỏng vấn tất cả những người sử dụng CSDL sau này để hiểu được những yêu cầu của họ đối với CSDL
- Tiến hành phân tích thiết kế hệ thống sau khi thống nhất được tất cả các yêu cầu của người sử dụng

1.5.1.3 Người sử dụng cuối

Người sử dụng cuối là những người truy cập CSDL để:

- Truy vấn
- Cập nhật
- Thống kê, báo cáo

1.5.1.4 Phân tích hệ thống và Lập trình ứng dụng

Phân tích hệ thống để định rõ những yêu cầu của người sử dụng cuối cùng, thống nhất để đưa ra khung nhìn cho từng đối tượng người sử dụng, quản lý các giao tác (transactions)...

Lập trình ứng dụng:

- Thực hiện các yêu cầu thông qua lập trình bằng những ngôn ngữ phù hợp
- Chạy thử chương trình (test)
- Chữa lỗi và gỡ rối chương trình (debug)
- Viết tài liệu, hướng dẫn sử dụng.
- Bảo trì hệ thống

1.5.2 Đối tượng gián tiếp

Ngoài những đối tượng trực tiếp tham gia vào một CSDL cụ thể như đã nói ở trên, còn có một đội ngũ những người phân tích, phát triển, và thực hiện tạo ra môi trường hệ thống và phần mềm của hệ quản trị cơ sở dữ liệu. Những người này không trực tiếp thao tác trên một hệ quản trị CSDL nào cụ thể. Họ là:

- Người phân tích và thực hiện tạo ra hệ thống của HQTCSDL
- Những nhà phát triển hệ công cụ (Tool developers)
- Người kiểm thử và bảo trì hệ thống

1.6 *Lợi ích của việc sử dụng HQTCSDL*

- Hạn chế dư thừa dữ liệu.
- Ngăn cản truy cập dữ liệu bất hợp pháp (bảo mật và phân quyền sử dụng).
- Cung cấp khả năng lưu trữ lâu dài cho các đối tượng và cấu trúc dữ liệu.
- Cho phép suy dẫn dữ liệu (từ dữ liệu này suy ra dữ liệu khác) sử dụng Rules.
- Cung cấp giao diện đa người dùng.
- Cho phép biểu diễn mối quan hệ phức tạp giữa các dữ liệu.

- Đảm bảo ràng buộc toàn vẹn dữ liệu (Enforcing Integrity Constraints).
- Cung cấp thủ tục sao lưu và phục hồi (backup và recovery)

2 Chương 2. NHỮNG KHÁI NIỆM VÀ KIẾN TRÚC CỦA HỆ THỐNG CƠ SỞ DỮ LIỆU

2.1 Mô hình dữ liệu, lược đồ và trường hợp (Data Models, Schemas, Instances)

Một trong những đặc điểm cơ bản của cơ sở dữ liệu là cung cấp một số mức độ trừu tượng hoá dữ liệu bằng cách làm ẩn đi cách thức tổ chức dữ liệu- cái mà hầu hết người dùng không cần biết đến.

Mô hình dữ liệu (Data Model): Là một tập những khái niệm dùng để biểu diễn cấu trúc của cơ sở dữ liệu-cung cấp những điều kiện cần thiết để đạt được mức độ trừu tượng dữ liệu. Cấu trúc dữ liệu bao gồm kiểu dữ liệu (data types) và mối quan hệ giữa các dữ liệu (relationships) và những ràng buộc (constraints) mà cơ sở dữ liệu phải tuân theo.

Hầu hết mô hình dữ liệu đều có một tập hợp các thao tác cơ bản (basic operations) để truy vấn và cập nhật dữ liệu.

- Thao tác chung (generic operations): Thêm (insert), Xoá (delete), Sửa (modify), Truy cập (retrieve)
- Thao tác do người dùng định nghĩa (user-defined operations)

2.1.1 Phân loại mô hình dữ liệu

Có rất nhiều mô hình dữ liệu đã được đưa ra, chúng ta có thể phân loại chúng theo những kiểu khái niệm mà họ đã dùng để biểu diễn cấu trúc cơ sở dữ liệu. Mô hình dữ liệu được chia làm 3 loại sau:

a. *Mô hình khái niệm (Conceptual (high-level, semantic) data models):*

Cung cấp những khái niệm gần gũi với đa số người sử dụng, mô hình này chỉ ra cái gì được đưa vào để quản lý. Mô hình này là phương tiện để những người phân tích thiết kế giao tiếp với người sử dụng, nhằm thu thập thông tin, xác định đúng đắn và đầy đủ yêu cầu của hệ thống.

Mô hình này sử dụng cấu trúc dữ liệu là: thực thể (entity), thuộc tính (attribute) và mối liên kết (relationship)

b. *Mô hình dữ liệu vật lý (Physical (low-level, internal) data models):*

Cung cấp những khái niệm để biểu diễn chi tiết cách thức dữ liệu được lưu trữ trong máy tính. Mô hình này chỉ ra định dạng bản ghi (record formats), thứ tự sắp xếp các bản ghi (record ordering) và đường dẫn để truy cập dữ liệu (access paths).

c. *Mô hình dữ liệu thể hiện (Implementation (record-oriented) data models):*

Mô tả các dữ liệu bằng cách sử dụng những ký pháp tương ứng với mô hình dữ liệu mà một hệ quản trị cơ sở dữ liệu sử dụng.

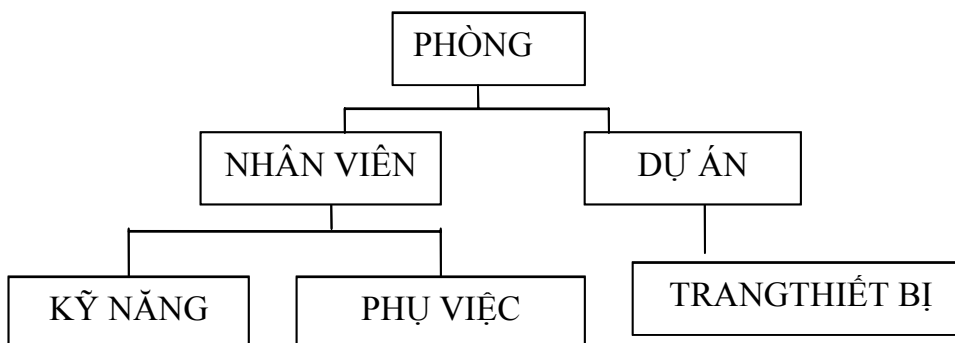
Các loại mô hình cơ sở dữ liệu thể hiện:

c1. Mô hình phân cấp

Mô hình CSDL phân cấp được biểu diễn dưới dạng cây và các đỉnh của cây là các bản ghi. Các bản ghi liên kết với nhau theo mối quan hệ cha-con.

- Một cha có nhiều con
- Một con chỉ có một cha

Ví dụ:



Hình 2.1. Minh họa mô hình cơ sở dữ liệu phân cấp

Ưu điểm:

- Thể hiện dễ dàng quan hệ 1-N.
- Việc phân chia dữ liệu dễ thể hiện, đảm bảo an toàn dữ liệu
- Tính độc lập của chương trình và các dữ liệu được đảm bảo

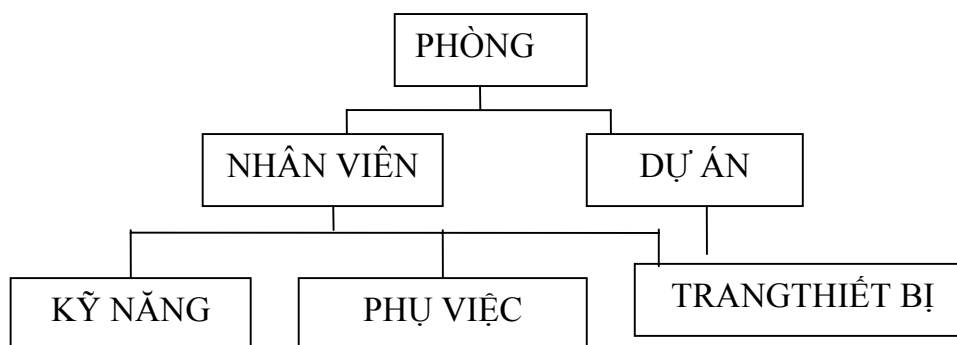
Nhược điểm:

- Không thể hiện được mối quan hệ M-N
- Trong một hệ thống phân cấp, dữ liệu được tổ chức như trên dẫn đến khó sửa đổi dữ liệu.

c2. Mô hình mạng

Cấu trúc cơ bản trong mô hình mạng là những tập hợp và mỗi tập hợp có bản ghi là bản ghi chủ và một số bản ghi thành viên. Mỗi thành viên có thể thuộc về nhiều tập hợp.

Ví dụ:



Hình 2.2. Minh họa mô hình cơ sở dữ liệu mạng

Ưu điểm:

- Dễ thể hiện mối liên kết M-N
- Kiểu truy cập dữ liệu mềm dẻo hơn kiểu phân cấp

Nhược điểm:

- Việc sửa đổi số liệu khó khăn.
- Với những lập trình viên, việc thiết kế CSDL khó.

c3. Mô hình quan hệ

Trong mô hình quan hệ, các dữ liệu được biểu diễn ở dạng các bảng với các dòng và các cột.

Trong mô hình quan hệ không có một cấu trúc vật lý nào của dữ liệu mô tả sự kết nối giữa các bảng. Thay vào đó, sự kết nối giữa các bảng được mô tả logic bằng các giá trị được lưu trữ trong các dòng của bảng. Chẳng hạn trong hình dưới đây, thuộc tính *ProCode*(*Mã tỉnh*) được lưu trong cả 2 bảng PROVINCE và bảng STUDENT, giá trị chung này cho phép người dùng liên kết được 2 bảng.

PROVINCE	
ProCode	ProName
04	Hà Nội
08	Tp Hồ Chí Minh
...	...

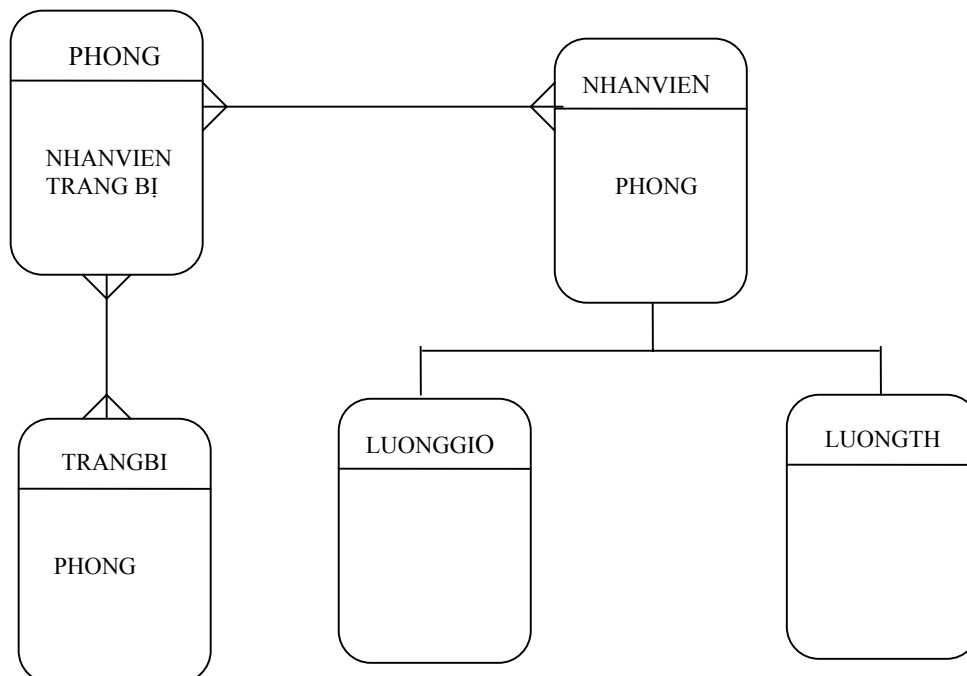
STUDENT			
StdNo	StdName	StdBird	ProCode
TD001	AA	9/16/1979	04
TD002	BB	6/19/1979	08
...

Hình 2.3. Minh họa mô hình cơ sở dữ liệu quan hệ

c4. Mô hình hướng đối tượng

Trong mô hình hướng đối tượng, các thuộc tính dữ liệu và các thao tác trên các dữ liệu này được bao gói trong một cấu trúc gọi là đối tượng.

Đối tượng có thể chứa các dữ liệu phức hợp như văn bản, hình ảnh, tiếng nói và hình ảnh động. Một đối tượng có thể yêu cầu hoặc xử lý dữ liệu từ một đối tượng khác bằng việc gửi đi một thông báo đến đối tượng đó. Mô hình hướng đối tượng biểu diễn một sơ đồ mới để lưu trữ và thao tác dữ liệu. Từ một đối tượng có thể sinh ra một đối tượng khác.



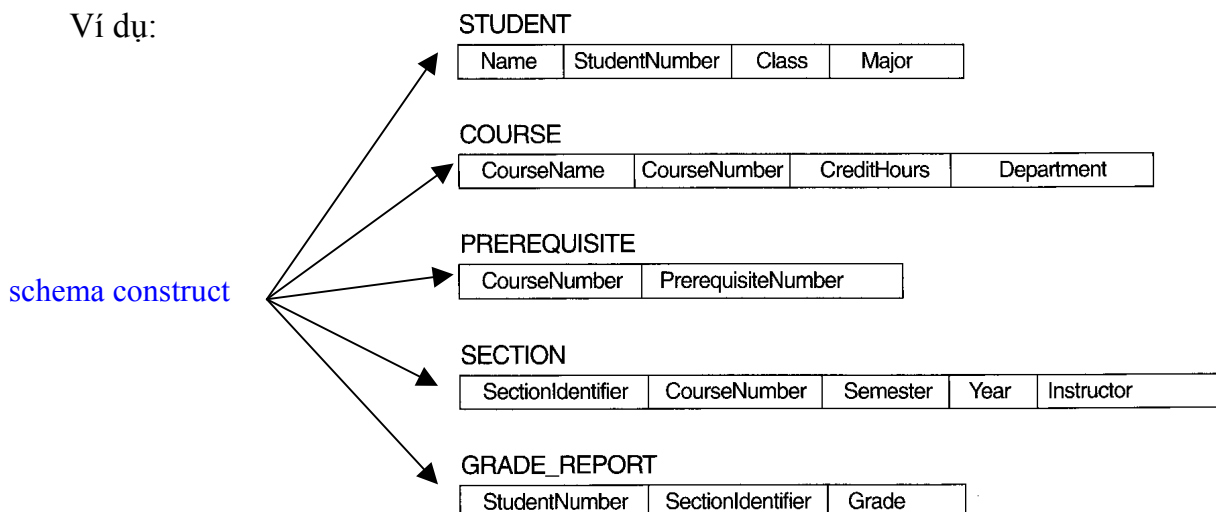
Hình 2.4. Minh họa mô hình cơ sở dữ liệu hướng đối tượng

2.1.2 Lược đồ (Schema) , minh họa (instances), và trạng thái (State)

Lược đồ cơ sở dữ liệu (Database Schema): là biểu diễn của cơ sở dữ liệu, bao gồm cấu trúc cơ sở dữ liệu và những ràng buộc trên dữ liệu.

Sơ đồ của lược đồ cơ sở dữ liệu (Schema Diagram): Là lược đồ cơ sở dữ liệu được biểu diễn thông qua sơ đồ.

Ví dụ:



Hình 2.5. Lược đồ cơ sở dữ liệu UNIVERSITY

Minh học cơ sở dữ liệu (Database Instance): Là dữ liệu thực sự được lưu trữ trong cơ sở dữ liệu ở thời điểm hiện tại. Database Instance cũng được gọi là trạng thái của cơ sở dữ liệu (**database state**)

Ví dụ:

STUDENT	Name	StudentNumber	Class	Major
	Smith	17	1	CS
	Brown	8	2	CS

COURSE	CourseName	CourseNumber	CreditHours	Department
	Intro to Computer Science	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Database	CS3380	3	CS

SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	85	MATH2410	Fall	98	King
	92	CS1310	Fall	98	Anderson
	102	CS3320	Spring	99	Knuth
	112	MATH2410	Fall	99	Chang
	119	CS1310	Fall	99	Anderson
	135	CS3380	Fall	99	Stone

GRADE_REPORT	StudentNumber	SectionIdentifier	Grade
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

PREREQUISITE	CourseNumber	PrerequisiteNumber
	CS3380	CS3320
	CS3380	MATH2410
	CS3320	CS1310

Hình 2.5. Cơ sở dữ liệu UNIVERSITY

Như vậy, **Database Schema** rất hiếm khi thay đổi, còn **Database State** thay đổi bất kỳ khi nào có sự cập nhập dữ liệu.

2.2 Lược đồ kiến trúc của hệ quản trị cơ sở dữ liệu (DBMS Architecture) và sự độc lập dữ liệu (Data Independence)

Như chúng ta đã biết, các tính chất quan trọng nhất của cơ sở dữ liệu là: (1) Đảm bảo sự độc lập giữa chương trình ứng dụng và dữ liệu. (2) Hỗ trợ nhiều khung nhìn cho các đối tượng người dùng khác nhau. (3) Sử dụng danh mục để lưu trữ biểu diễn dữ liệu (schema). Trong phần này, chúng ta sẽ tìm hiểu kiến trúc của hệ quản trị

cơ sở dữ liệu, gọi là **Lược đồ kiến trúc 3 mức** (three –schema architecture). Sau đó chúng ta sẽ tìm hiểu về khái niệm độc lập dữ liệu.

2.2.1 Lược đồ kiến trúc 3 mức của HQTCSDL

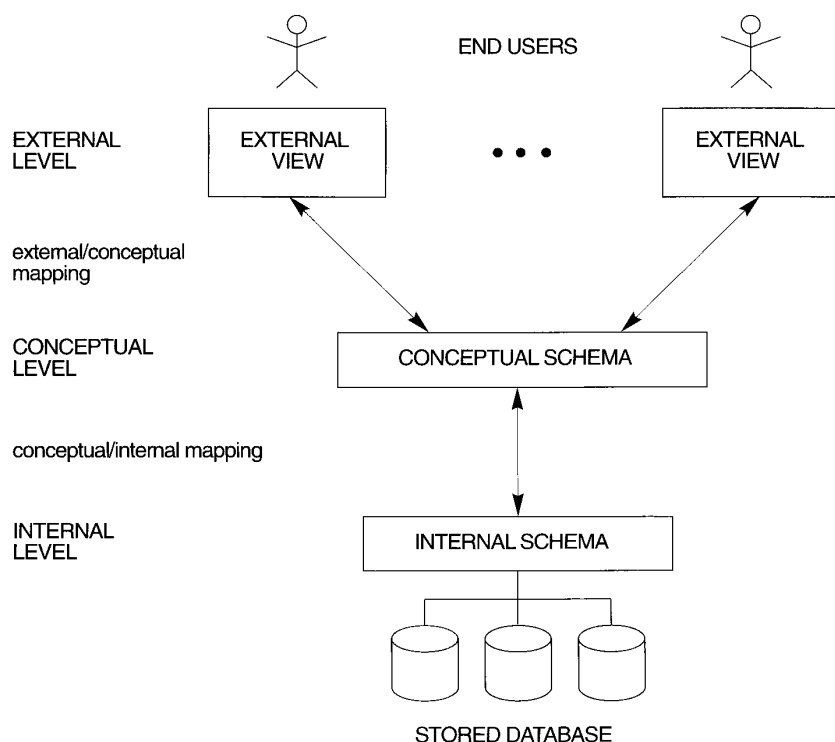
Mục đích của việc phân thành 3 mức trong kiến trúc của mô hình HQTCSDL là để tách biệt các ứng dụng của người sử dụng với cơ sở dữ liệu vật lý. Trong kiến trúc này, lược đồ có thể được định nghĩa ở 3 mức sau:

Lược đồ mức trong (Internal schema) ở Mức trong (Internal level) để biểu diễn chi tiết cấu trúc lưu trữ dữ liệu và cách thức truy cập dữ liệu. Lược đồ mức trong sử dụng mô hình dữ liệu vật lý (physical data model).

Lược đồ khái niệm (Conceptual schema) ở Mức khái niệm (Conceptual level) để biểu diễn cấu trúc và các ràng buộc trong toàn bộ cơ sở dữ liệu phục vụ cho việc giao tiếp với người sử dụng. Lược đồ khái niệm ẩn đi cách thức tổ chức vật lý của dữ liệu, chỉ tập trung vào việc biểu diễn các thực thể, các kiểu dữ liệu, mối quan hệ giữa các thực thể, các thao tác của người sử dụng và các ràng buộc giữa các dữ liệu. Mô hình dữ liệu mức khái niệm (Conceptual data model) hoặc Mô hình dữ liệu thể hiện (Implementation data model) có thể được sử dụng ở mức này.

Lược đồ mức ngoài (External Level) ở Mức ngoài (External level hoặc View level) để biểu diễn hàng loạt những khung nhìn của người sử dụng (user views). Mô hình dữ liệu mức cao (High-level data model) hoặc Mô hình dữ liệu thể hiện (Implementation data model) có thể được sử dụng ở mức này.

Ánh xạ giữa các mức này là cần thiết. Những chương trình làm việc với dữ liệu ở mức ngoài và được HQTCSDL ánh xạ tới dữ liệu vật lý ở mức trong để thực hiện.



Hình 2.6. Lược đồ kiến trúc 3 mức của HQTCSDL

2.2.2 Độc lập dữ liệu

Kiến trúc 3 mức của HQTCSDL có thể được sử dụng để giải thích khái niệm về độc lập dữ liệu. Độc lập dữ liệu là khả năng thay đổi lược đồ ở một mức nào đó của hệ thống cơ sở dữ liệu mà không cần phải thay đổi lược đồ ở mức cao hơn. Chúng ta có thể định nghĩa 2 kiểu của độc lập dữ liệu:

Độc lập dữ liệu logic (Logical data independence): cho phép thay đổi lược đồ khái niệm mà không cần phải thay đổi lược đồ mức ngoài hoặc những chương trình ứng dụng. Chúng ta có thể thay đổi lược đồ khái niệm để mở rộng (thêm các trường dữ liệu, các bản ghi) hoặc thu nhỏ cơ sở dữ liệu (xóa các trường dữ liệu, các bản ghi).

Độc lập dữ liệu vật lý (Physical data independence): cho phép thay đổi lược đồ mức trong mà không cần thay đổi lược đồ khái niệm. Có khi chúng ta cần thay đổi lược đồ mức trong vì các file vật lý đôi khi cần tổ chức lại để tăng hiệu quả thực hiện. Nếu kiểu dữ liệu không thay đổi thì chúng ta không cần thay đổi lại lược đồ khái niệm.

2.3 Ngôn ngữ của HQTCSDL

Vì HQTCSDL phục vụ có nhiều đối tượng người sử dụng khác nhau nên nó phải hỗ trợ ngôn ngữ để người sử dụng tương tác với nó. Trong phần này chúng ta sẽ tìm hiểu một số những ngôn ngữ được HQTCSDL hỗ trợ.

Ngôn ngữ định nghĩa dữ liệu (Data Definition Language - DDL): Là ngôn ngữ được các nhà quản trị cơ sở dữ liệu (DBA) và các nhà thiết kế cơ sở dữ liệu (database designers) dùng để xây dựng lược đồ khái niệm của cơ sở dữ liệu. Trong nhiều HQTCSDL, DDL cũng được sử dụng để định nghĩa lược đồ mức trong và mức ngoài (các khung nhìn). Một số HQTCSDL chia thành 2 ngôn ngữ: Ngôn ngữ định nghĩa lưu trữ (storage definition language – SDL) và Ngôn ngữ định nghĩa khung nhìn (view definition language -VDL) được dùng để định nghĩa lược đồ mức trong và mức ngoài.

Ngôn ngữ thực hiện dữ liệu (Data Manipulation Language -DML): Là ngôn ngữ được sử dụng để thao tác với dữ liệu bao gồm việc truy cập đến bản ghi và cập nhật dữ liệu cho bản ghi (thêm, sửa, xóa). Các lệnh DML có thể được nhúng trong ngôn ngữ lập trình hoặc thực hiện độc lập (ngôn ngữ truy vấn).

2.4 Các tính năng của HQTCSDL

- Nạp dữ liệu đang lưu trữ ở các tệp tin vào cơ sở dữ liệu (Conversion Tool).
- Cung cấp các thao tác truy xuất
- Đảm bảo tính độc lập dữ liệu
- Cung cấp thủ tục sao lưu và phục hồi (backup và recovery)
- Cung cấp các thủ tục điều khiển đồng thời (Do tính truy xuất đồng thời và cạnh tranh)
- Cung cấp các thủ tục kiểm soát bản quyền, kiểm tra tính đúng đắn của dữ liệu (để đảm bảo tính an toàn và toàn vẹn dữ liệu)

2.5 Phân loại HQTCSDL

Người ta phân loại HQTCSDL dựa trên một số tiêu chí:

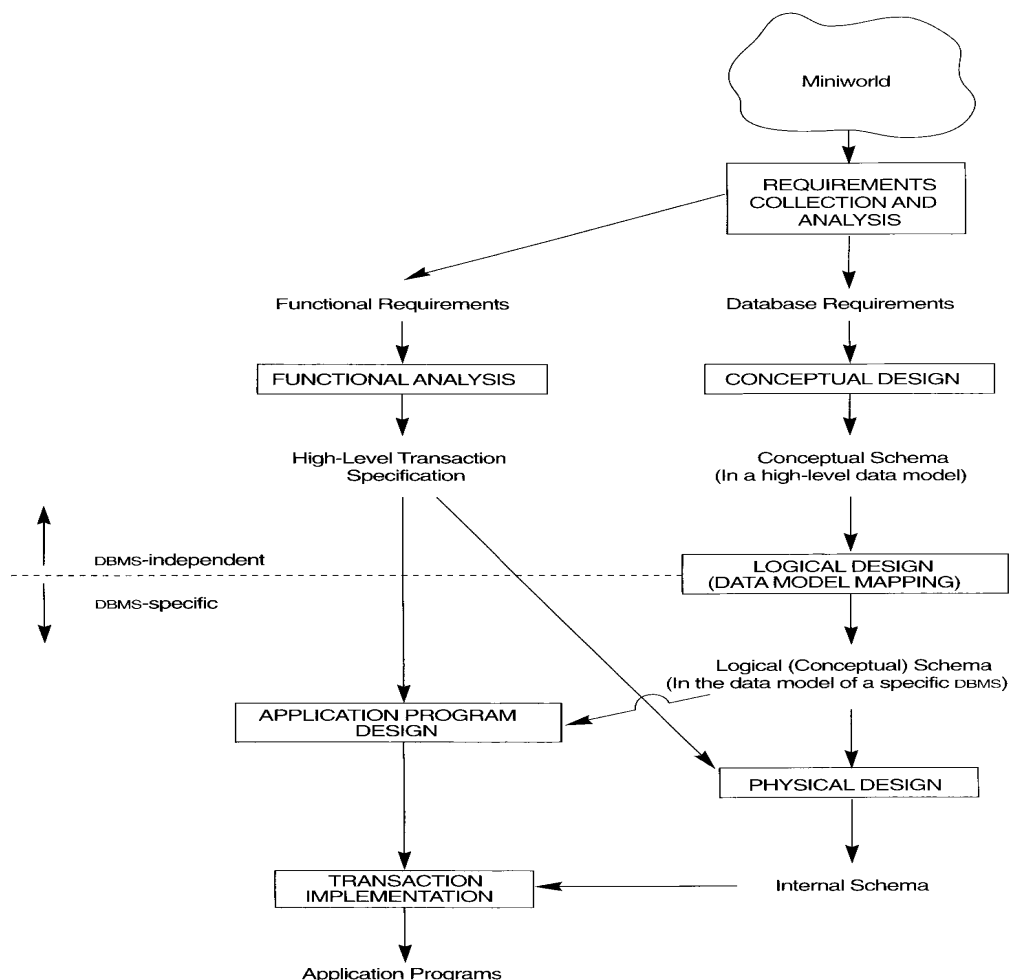
Dựa trên mô hình dữ liệu được HQTCSDL sử dụng: Mô hình quan hệ (Relational Data Model), Mô hình hướng đối tượng (Object Data Model), Mô hình mạng (Network Data Model), mô hình phân cấp (Hierarchical Data Model), mô hình quan hệ thực thể (Entity – Relationship Data Model)...

Tiêu chí khác:

- Số người sử dụng HQTCSDL: Đơn người dùng (Single-user), Đa người dùng (multi-user). Hầu hết các HQTCSDL hiện nay đều là HQTCSDL đa người dùng.
- Vị trí HQTCSDL: tập trung (sử dụng một máy đơn) hay phân tán (sử dụng nhiều máy tính).
- Giá của HQTCSDL: 100\$- 300\$; 10000\$- 100000\$

3 Chương 3. MÔ HÌNH QUAN HỆ - THỰC THỂ (Entity – Relationship Model)

3.1 Sử dụng mô hình dữ liệu khái niệm mức cao để thiết kế cơ sở dữ liệu



Hình 3.1. Các giai đoạn thiết kế cơ sở dữ liệu

Hình trên chỉ ra tiến trình thiết kế cơ sở dữ liệu một cách đơn giản. Bước đầu tiên là *tập hợp và phân tích yêu cầu hệ thống*. Trong bước này, người thiết kế cơ sở dữ liệu phải tiến hành thu thập yêu cầu của người sử dụng, sau đó viết tài liệu *những yêu cầu dữ liệu*. Kết quả của bước này là viết ra được tập hợp những yêu cầu tất cả các đối tượng người dùng một cách xúc tích. Từ đó, ta xác định được yêu cầu chức năng (*Functional Requirements*) của hệ thống.

Sau khi tất cả các yêu cầu đã được tập hợp và phân tích, bước tiếp theo là xây dựng *lược đồ khái niệm (conceptual schema)* cho cơ sở dữ liệu. Lược đồ khái niệm là nơi biểu diễn xúc tích những yêu cầu của người sử dụng và biểu diễn chi tiết những kiểu thực thể (*entity types*), quan hệ (*relationships*) và những ràng buộc (*constraints*) của dữ liệu, phần này sử dụng những khái niệm được cung cấp trong mô hình dữ liệu mức cao (*High level data model*). Mô hình dữ liệu mức cao là mô hình dữ liệu được dùng để giao tiếp với người sử dụng bình thường vì thế nó rất dễ hiểu, nó chỉ ra *cái gì*

cần được lưu trong cơ sở dữ liệu chứ không chỉ ra cụ thể dữ liệu được thực hiện *như thế nào*.

Bước tiếp theo trong quá trình thiết kế là cài đặt cơ sở dữ liệu trên một *mô hình dữ liệu thực hiện*, sử dụng các Hệ quản trị cơ sở dữ liệu nào đó (Hầu hết các hệ quản trị cơ sở dữ liệu hiện nay sử dụng mô hình dữ liệu quan hoặc hướng đối tượng). Vì thế, chúng ta cần thiết phải chuyển từ mô hình dữ liệu mức cao sang những mô hình dữ liệu thực hiện. Bước này được gọi là thiết kế logic (*Logical design*) hoặc ánh xạ mô hình dữ liệu (*Data model mapping*) và kết quả của bước này là *lược đồ cơ sở dữ liệu* trong mô hình cơ sở dữ liệu thực hiện.

Bước cuối cùng là thiết kế vật lý cho cơ sở dữ liệu (*physical design*), bao gồm việc thiết kế những cấu trúc lưu trữ dữ liệu bên trong, đường dẫn truy cập, tổ chức file của các file dữ liệu.

Trong chương này, chúng ta sẽ sử dụng *mô hình khái niệm ER* cho thiết kế lược đồ khái niệm (*Conceptual Schema*).

3.2 Mục đích của mô hình khái niệm ER(Entity – Relationship Model)

Qua bước xem xét tổng quát trên, ta thấy rằng, mô hình E-R là một mô tả logic chi tiết dữ liệu của một tổ chức hay một lĩnh vực nghiệp vụ. Nó giúp người thiết kế cơ sở dữ liệu mô tả thế giới thực gắn gũi với quan niệm và cách nhận nhìn nhận bình thường của con người. Nó là công cụ để phân tích thông tin nghiệp vụ.

Mục đích của mô hình E – R:

- Làm thống nhất quan điểm về dữ liệu của những người tham gia hệ thống: Người quản lý, người dùng cuối, người thiết kế hệ thống
- Xác định các xử lý về dữ liệu cũng như các ràng buộc trên các dữ liệu.
- Giúp đỡ việc thể hiện cơ sở dữ liệu về mặt cấu trúc: Sử dụng thực thể và các mối liên kết giữa các thực thể. Biểu diễn mô hình quan hệ thực thể bằng một sơ đồ.

3.3 Ví dụ về một cơ sở dữ liệu ứng dụng

Trong phần này, chúng ta sẽ tìm hiểu một ví dụ về cơ sở dữ liệu ứng dụng - gọi là *COMPANY* để minh họa các khái niệm trong mô hình ER và sử dụng nó trong khi thiết kế lược đồ khái niệm.

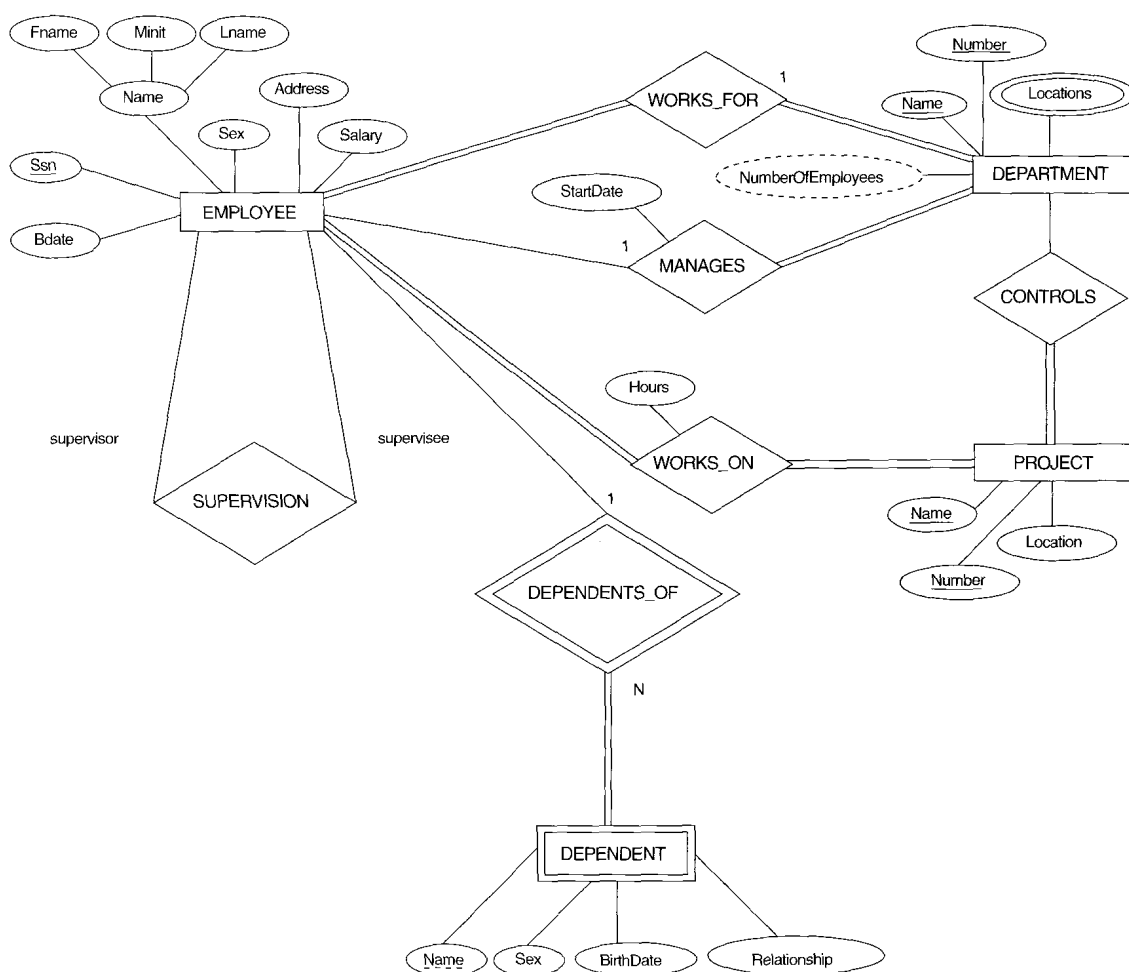
Cơ sở dữ liệu *COMPANY* cần lưu trữ thông tin về nhân viên (*employees*), phòng/ban (*departments*), và các dự án (*projects*) trong công ty. Sau khi tập hợp được tất cả các yêu cầu của hệ thống, người thiết kế cơ sở dữ liệu tiến hành mô tả lại *miniworld* bằng mô hình ER.

Sau đây là các quy tắc nghiệp vụ của hệ thống cơ sở dữ liệu *COMPANY*, chúng ta sẽ xây dựng mô hình ER từng bước để giới thiệu các khái niệm của mô hình ER.

1. Công ty (*COMPANY*) có nhiều phòng/ban (*DEPARTMENTS*). Mỗi phòng/ban có tên (*name*), mã số (*number*) duy nhất và có một nhân viên (*employee*) làm quản lý (*manages*) phòng/ban. Chúng ta lưu lại ngày bắt đầu (*start date*) làm quản lý phòng/ban của nhân viên đó.

2. Mỗi phòng/ban có thể có nhiều địa điểm khác nhau (*locations*).
3. Mỗi phòng/ban điều hành một số dự án (PROJECTs). Mỗi dự án có tên (*name*), mã số (*number*) duy nhất và chỉ có một địa điểm (*location*).
4. Với mỗi nhân viên, chúng ta lưu lại những thông tin sau: tên (*name*), số bảo hiểm xã hội (*social security number*), địa chỉ (*address*), lương (*salary*), giới tính (*sex*), và ngày sinh (*birth date*).
5. Mỗi nhân viên làm việc ở một phòng/ban, nhưng có thể làm việc cho nhiều dự án. Chúng ta lưu lại số giờ làm việc (*the number of hours per week*) của từng nhân viên trong từng dự án.
6. Chúng ta lưu lại thông tin về người quản lý trực tiếp (*direct supervisor*), của mỗi nhân viên. Người quản lý trực tiếp cũng là một nhân viên.
7. Mỗi nhân viên có những người phụ thuộc vào họ (DEPENDENTS). Mỗi người phụ thuộc ta lưu lại thông tin về tên (*name*), giới tính (*sex*), ngày sinh (*birth date*) và quan hệ (*relationship*).

Hình minh họa sau sẽ chỉ ra sơ đồ ER của cơ sở dữ liệu COMPANY dựa trên những ký hiệu đồ họa được quy định trong mô hình ER.



Hình 3.2. Sơ đồ ER của cơ sở dữ liệu COMPANY

Lưu ý, trên đây chỉ là giới thiệu sơ bộ về mô hình ER, các bạn chưa cần phải hiểu được toàn bộ sơ đồ trên.

Phần tiếp theo chúng ta sẽ giới thiệu về mô hình ER. Các khái niệm cơ bản trong mô hình ER gồm có **Thực thể (Entity)**, **Thuộc tính (Attributes)**, và **Quan hệ (Relationship)**.

3.4 Kiểu thực thể (Entity Type), Thuộc tính (Attributes), Khoá (Keys)

3.4.1 Thực thể (Entities) và thuộc tính (Attributes)

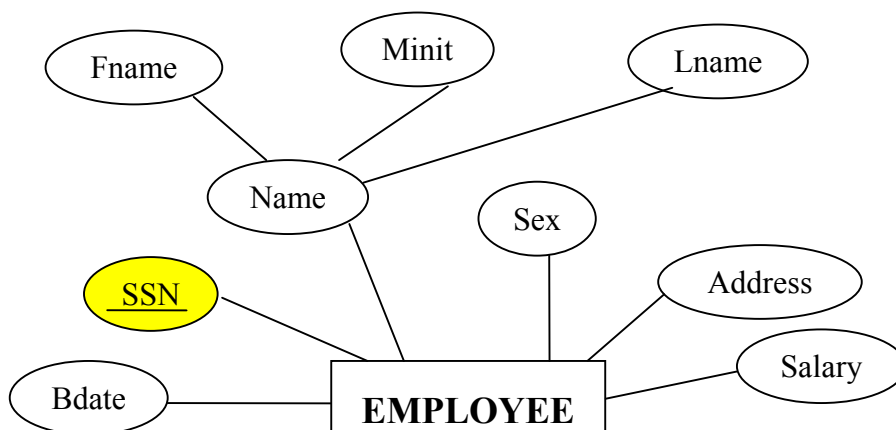
3.4.1.1 Thực thể

- Các thực thể vốn tồn tại trong thế giới thực. Một thực thể là một khái niệm để chỉ một lớp các đối tượng cụ thể hay các khái niệm có cùng những đặc tính chung mà ta quan tâm.
- Tên thực thể: Là tên của một lớp đối tượng. Trong 1 CSDL, tên thực thể không được trùng nhau.
- Ký hiệu: Trong mô hình E-R, thực thể được biểu diễn bằng một hình chữ nhật có tên bên trong.
- Ví dụ:

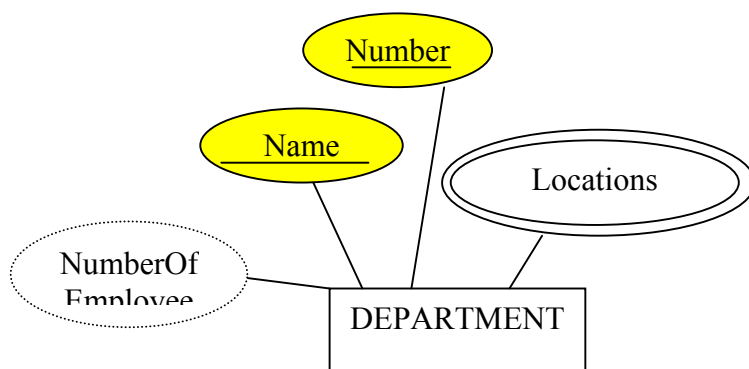


3.4.1.2 Thuộc tính

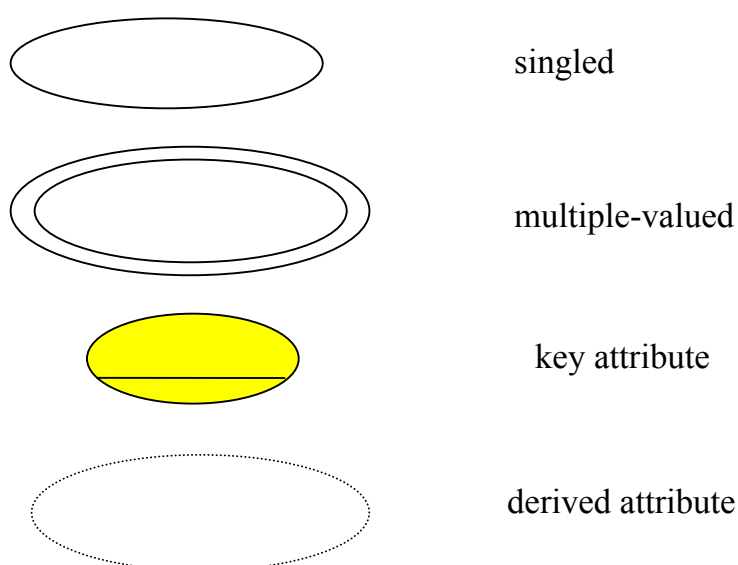
- Thuộc tính là các đặc trưng (*properties*) được sử dụng để biểu diễn thực thể.
- Ví dụ: Thực thể EMPLOYEE có các thuộc tính: *Name*, *SSN*, *Address*, *Sex*, *BirthDate*.
- Thuộc tính được ký hiệu bằng hình oval, bên trong ghi tên của thuộc tính. Thuộc tính của thực thể nào thì sẽ được gắn với thực thể đó.
- Ví dụ:



Hình 3.3. Thực thể EMPLOYEE và các thuộc tính của nó



Hình 3.4. Thực thể DEPARTMENT và các thuộc tính của nó



Hình 3.5. Một số ký hiệu của thuộc tính

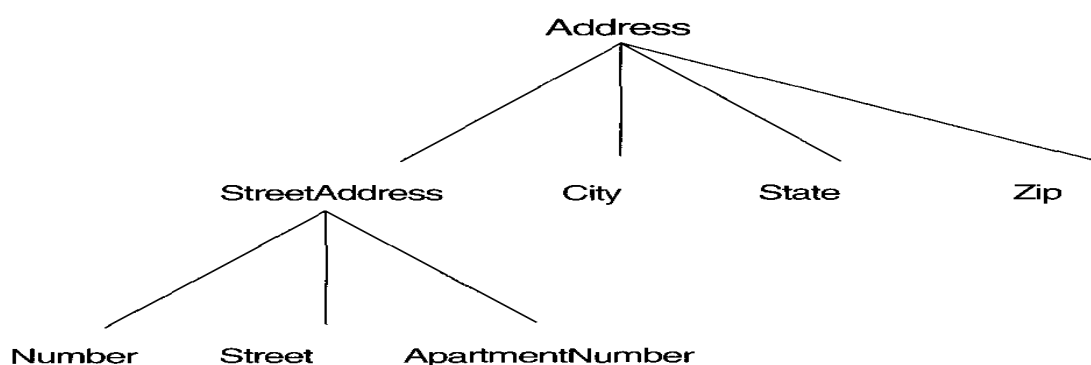
Các kiểu thuộc tính trong mô hình ER:

Thuộc tính đơn (*simple*) đối lập với **thuộc tính tổ hợp** (*composite*), **thuộc tính đơn trị** (*single-value*) đối lập với **thuộc tính đa trị** (*multivalued*), **thuộc tính lưu trữ** (*stored*) đối lập với **thuộc tính suy diễn** (*derived*). Sau đây ta sẽ tìm hiểu chi tiết về các loại thuộc tính này:

Thuộc tính đơn (*simple*) hay còn gọi là **thuộc tính nguyên tử** (*Atomic*): Chỉ có một giá trị trong một thuộc tính của một thực thể. Ví dụ: Thuộc tính *Birthdate*, *Sex...* của *Employee* là thuộc tính nguyên tử.

Thuộc tính tổ hợp (*Composite*): là thuộc tính được kết hợp của một số thành phần. Ví dụ: *Address*(*Apt#*, *House#*, *Street*, *City*, *State*, *ZipCode*, *County*) hoặc *Name* (*FirstName*, *MiddleName*, *LastName*) là thuộc tính tổ hợp.

Thuộc tính tổ hợp có thể được biểu diễn phân cấp như sau:



Hình 3.6. Biểu diễn phân cấp của thuộc tính tổ hợp

Thuộc tính đơn trị (single-value): Là thuộc tính chỉ có một giá trị duy nhất ở một thời điểm. Ví dụ: *Sex, Birthdate, ...*

Thuộc tính đa trị (multivalued): Là thuộc tính có thể có nhiều giá trị tại một thời điểm. Ví dụ: *PreviousDegrees...*

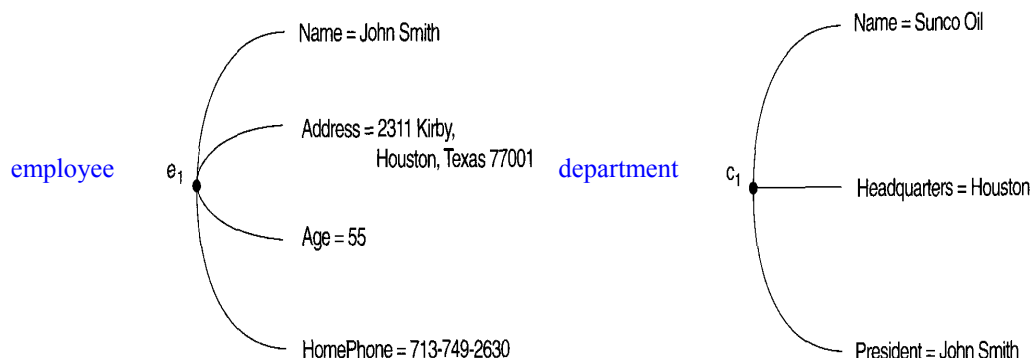
Ký hiệu: {*PreviousDegrees*}

Thuộc tính tổ hợp và thuộc tính đa trị có thể làm tổ ở nhiều mức, mặc dù ít gặp trường hợp này. Ví dụ: *PreviousDegrees* của thực thể STUDENT là loại thuộc tính đó. Ký hiệu: {*PreviousDegrees (College, Year, Degree, Field)*}.

Thuộc tính lưu trữ (stored attribute) và thuộc tính suy diễn (derived attribute): Thuộc tính lưu trữ là thuộc tính mà giá trị của nó phải được lưu trữ, còn thuộc tính suy diễn là thuộc tính mà giá trị của nó có thể suy ra từ giá trị của những thuộc tính khác. Ví dụ: *Age(derived attribute)* được suy diễn từ *BirthDate (stored attribute)*

Giá trị rỗng của thuộc tính (Null Values): Trong một vài trường hợp, một thực thể có thể không có giá trị tương ứng cho một thuộc tính, ví dụ thuộc tính *NameDependent* (Tên của người phụ thuộc), nếu một nhân viên nào đó trong thực thể EMPLOYEE chưa có người phụ thuộc thì thuộc tính *NameDependent* tương ứng với nhân viên đó sẽ không có giá trị. Hoặc trong trường hợp thuộc tính có giá trị nhưng chưa được biết, ví dụ thuộc tính *PhoneNumber* (Số điện thoại). Trong trường hợp này, một giá trị đặc biệt được tạo ra, đó là giá trị **Null**.

Mỗi thuộc tính trong thực thể luôn có giá trị, ví dụ các thuộc tính trong thực thể EMPLOYEE có các giá trị sau: *Name='John Smith', SSN='123456789', Address='731 Fondren, Houston, TX', Sex='M', BirthDate='09-JAN-55'*. Một bộ giá trị của một thực thể được gọi là một **bản ghi (record)**.



Hình 3.7. Hai thực thể và giá trị thuộc tính của nó

3.4.2 Kiểu thực thể, Khoá và tập giá trị

Kiểu thực thể: là tập hợp những thực thể có cùng các thuộc tính cơ bản. Ví dụ, kiểu thực thể EMPLOYEE, kiểu thực thể PROJECT.

Khoá: Mỗi một kiểu thực thể phải có một hoặc một tập các thuộc tính mang giá trị duy nhất (unique value) để phân biệt giữa bản ghi này với bản ghi khác. Thuộc tính đó gọi là khoá của kiểu thực thể (*Key attribute*). Ví dụ: thuộc tính SSN của kiểu thực thể EMPLOYEE, hoặc thuộc tính *NumberStudent* (*Mã sinh viên*) của kiểu thực thể STUDENT. Chú ý là khoá có thể gồm một hoặc một tập các thuộc tính.

Tập giá trị hay còn gọi là miền xác định (Domain): là tập những giá trị mà thuộc tính có thể nhận được. Ví dụ: Miền xác định của thuộc tính Sex là {Male, Female}, hoặc của Mark(Điểm) là từ 0..10.

Tập giá trị không được biểu diễn trong lược đồ ER.

3.5 Liên kết, Kiểu liên kết và các Ràng buộc liên kết

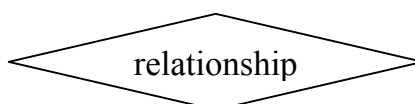
3.5.1 Định nghĩa liên kết và kiểu liên kết

Liên kết (Relationship) dùng để chỉ mối quan hệ giữa hai hay nhiều thực thể khác nhau. Ví dụ: Nhân viên (A) làm việc cho dự án (X), nhân viên B làm việc cho dự án (X)...

Những liên kết của cùng một kiểu được nhóm lại gọi là kiểu liên kết (Relationship Type), ví dụ kiểu liên kết WORK_ON (làm việc cho), kiểu liên kết MANAGES (làm quản lý)...

Trong lược đồ ER, người ta sử dụng hình thoi và bên trong ghi tên kiểu liên kết để ký hiệu kiểu liên kết.

Ký hiệu:



Lưu ý: Kiểu liên kết cũng có thể có thuộc tính của nó.

3.5.2 Bậc của kiểu liên kết

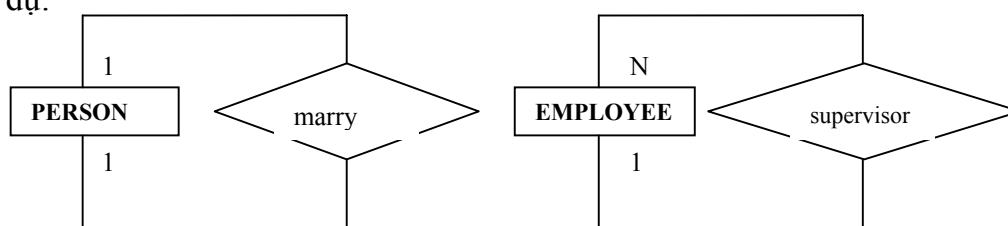
Là số lượng các kiểu thực thể tham gia vào liên kết. Có các kiểu liên kết sau:

- Kiểu liên kết bậc 1 (đệ quy) là mối quan hệ giữa cùng 1 kiểu thực thể.
- Kiểu liên kết bậc 2 là mối liên kết giữa hai kiểu thực thể
- Kiểu liên kết bậc 3 là mối liên kết giữa 3 kiểu thực thể

3.5.2.1 Mối quan hệ bậc 1

Mối quan hệ bậc 1 (đệ quy) là mối quan hệ giữa cùng 1 kiểu thực thể

Ví dụ:

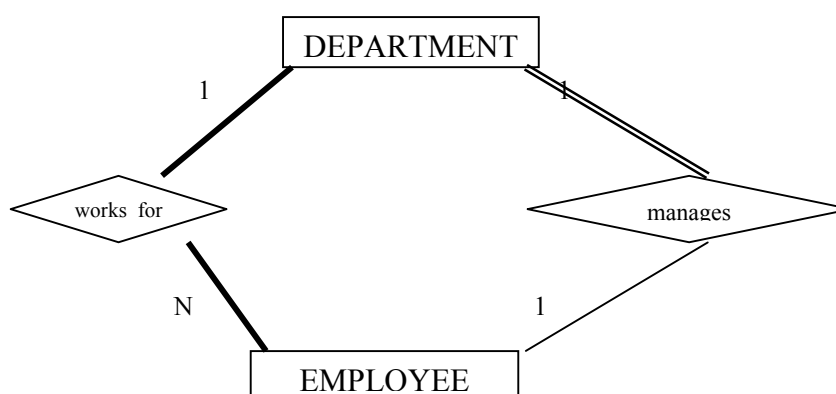


Hình 3.8. Minh họa mối quan hệ bậc 1

3.5.2.2 Mối quan hệ bậc 2

Là mối quan hệ giữa 2 kiểu thực thể khác nhau.

Ví dụ:



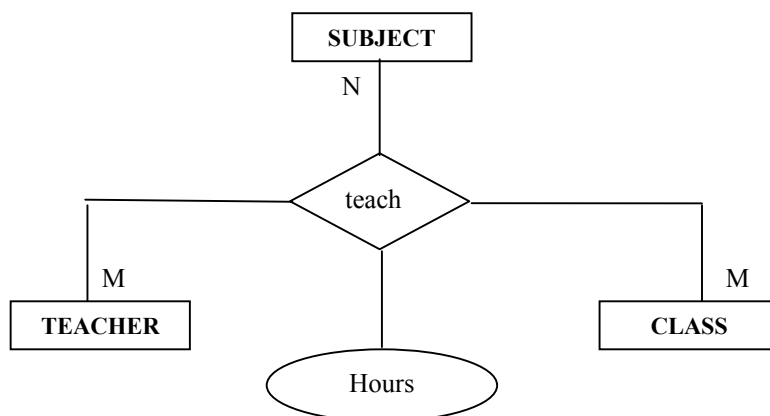
Hình 3.9. Minh họa mối quan hệ bậc 2

Ví dụ: Mối liên kết giữa hai kiểu thực thể DEPARTMENT và EMPLOYEE sau đây là kiểu liên kết bậc 2 vì nó có sự tham gia của hai kiểu thực thể.

Hình minh họa trên còn cho ta thấy, có thể có nhiều hơn một kiểu liên kết giữa hai kiểu thực thể khác nhau.

3.5.2.3 Mối quan hệ bậc 3

Là mối quan hệ giữa 3 kiểu thực thể khác kiểu.



Hình 3.10. Minh họa mối quan hệ bậc 3

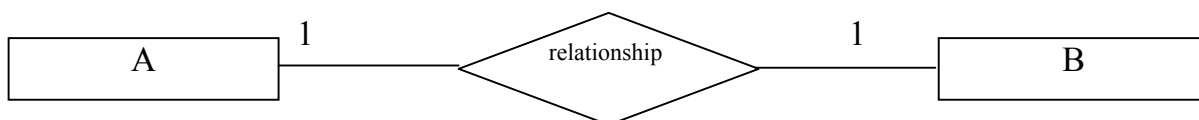
3.5.3 Ràng buộc liên kết

Các kiểu liên kết thường có một số ràng buộc nào đó về các thực thể có thể kết hợp với nhau tham gia trong một liên kết phù hợp. Các ràng buộc này xác định từ tình huống thực tế mà liên kết thể hiện. Có các loại ràng buộc như sau:

3.5.3.1 Tỷ số lực lượng:

Trong các kiểu liên kết bậc 2, tỷ số lực lượng chỉ rõ số thực thể tham gia vào liên kết. Các tỷ số lực lượng có thể là: 1:1, 1:N, N:1 và M:N.

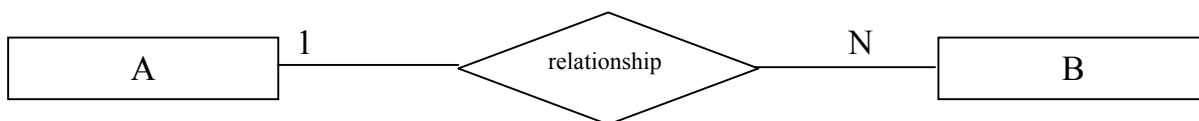
Tỷ số 1:1: Một thực thể của kiểu A có liên kết với một thực thể của kiểu B và ngược lại.



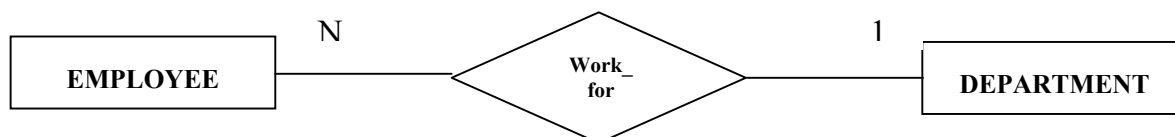
Ví dụ: Một nhân viên (EMPLOYEE) quản lý một phòng (DEPARTMENT), và một phòng chỉ có một nhân viên quản lý.



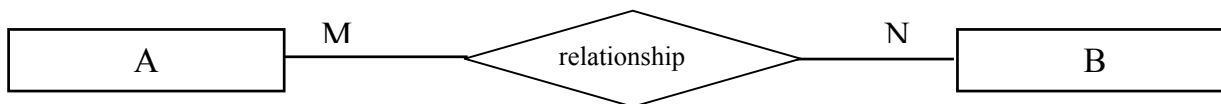
Tỷ số 1:N: Một thực thể của kiểu A có liên kết với nhiều thực thể của kiểu B. Nhưng một thực thể của kiểu B lại có liên kết duy nhất với thực thể của kiểu A.



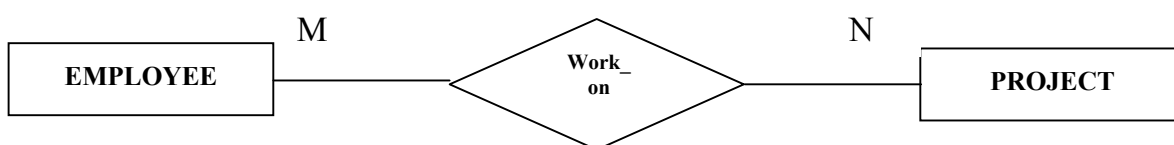
Ví dụ: Một nhân viên (EMPLOYEE) làm việc cho một phòng (DEPARTMENT), và một phòng có nhiều nhân viên làm việc.



Tỷ số M:N: Một thực thể của kiểu A có liên kết với nhiều thực thể của kiểu B và ngược lại.



Ví dụ:



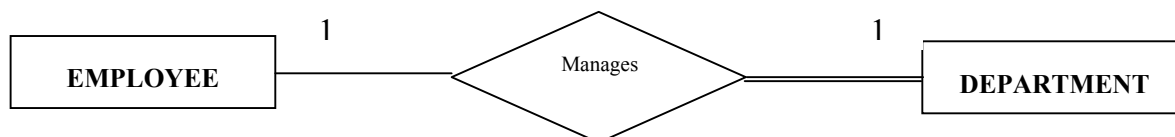
3.5.3.2 Ràng buộc về sự tham gia liên kết (Participation constraint)

Ràng buộc về sự tham gia liên kết được xác định trên từng thực thể trong từng kiểu liên kết mà thực thể đó tham gia, bao gồm: lực lượng tham gia toàn bộ (*total participation*) và lực lượng tham gia bộ phận (*partial participation*).

Ví dụ: Trong kiểu liên kết Manages giữa hai kiểu thực thể EMPLOYEE và DEPARTMENT, lực lượng tham gia của kiểu thực thể DEPARTMENT là toàn bộ, vì DEPARTMENT nào cũng có người quản lý, còn lực lượng tham gia của kiểu thực thể EMPLOYEE là bộ phận vì không phải EMPLOYEE nào cũng làm quản lý (manages) của DEPARTMENT.

Trong sơ đồ ER, kiểu thực thể có lực lượng tham gia liên kết toàn bộ được nối với kiểu liên kết bằng gạch nối kép, còn kiểu thực thể có lực lượng tham gia bộ phận được nối với kiểu liên kết bằng gạch nối đơn.

Ví dụ:



3.5.3.3 Lực lượng tham gia liên kết

Trong mỗi liên kết giữa các thực thể, ta cần quan tâm đến lực lượng tham gia liên kết, đó là số bản ghi lớn nhất và nhỏ nhất của thực thể tham gia vào liên kết đó.

Ký hiệu: Thêm (min,max) vào mỗi liên kết.

Trong đó:

- min là số bản ghi nhỏ nhất tham gia vào liên kết

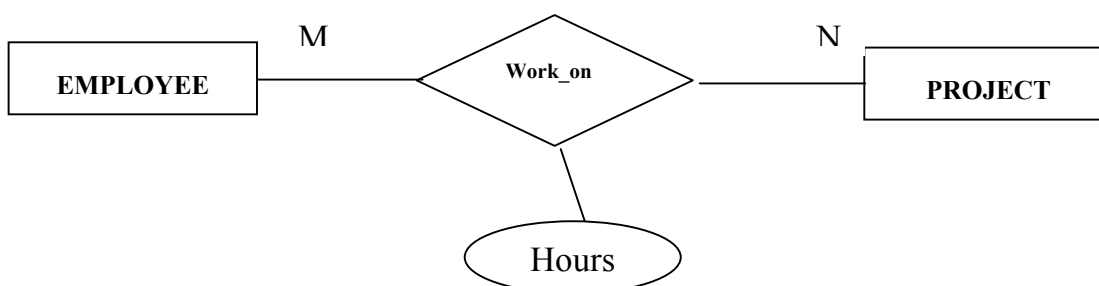
- max là số bản ghi lớn nhất tham gia vào liên kết
- Mặc định, min=0, max=n
- Chúng ta xác định lực lượng này từ khảo sát thực tế bài toán.

Ví dụ:

- Tại một thời điểm, một phòng có duy nhất một người quản lý-người đó là nhân viên, một nhân viên chỉ quản lý duy nhất một phòng. Vì vậy, (0,1) là lực lượng của EMPLOYEE và (1,1) là lực lượng của DEPARTMENT tham gia trong liên kết Manages(quản lý).
- Một nhân viên chỉ có thể làm việc cho một phòng nhưng một phòng có thể có bất kỳ số lượng nhân viên nào. Vì thế, (1,1) là lực lượng của EMPLOYEE và (0,n) là lực lượng của DEPARTMENT tham gia trong liên kết Works_For(làm việc cho).

3.5.3.4 Thuộc tính của kiểu liên kết

Kiểu liên kết cũng có thể có thuộc tính. Ví dụ: Số giờ nhân viên làm việc cho dự án (Hours) là thuộc tính của mỗi liên kết giữa hai kiểu thực thể EMPLOYEE và PROJECT.



3.6 Kiểu thực thể yếu(Weak Entity)

Kiểu thực thể yếu là kiểu thực thể tồn tại phụ thuộc vào thực thể khác (thực thể làm chủ hay còn gọi là xác định nó). Kiểu thực thể yếu không có khoá.

Kiểu thực thể yếu được xác định bằng:

- Một hay một tập các thuộc tính xác định kiểu thực thể yếu
- Và thực thể làm chủ (xác định) thực thể yếu.

Kiểu thực thể yếu luôn có lực lượng tham gia liên kết toàn bộ.

3.7 Tổng quát hóa và chuyên biệt hóa

Tổng quát hóa là khái niệm cho phép ta xem một vật thể nào đó (các thực thể) là một thực thể con của một vật thể khác tổng quát hơn.

Ví dụ: SÁCH là một loại con của loại tổng quát hơn là TAILIEU nói chung.

Chuyên biệt hóa là khái niệm ngược lại với tổng quát hóa.

Ví dụ:

Ôtô, xeca, taxi gộp lại thành một thực thể tổng quát hơn là Phuongtienvantai.

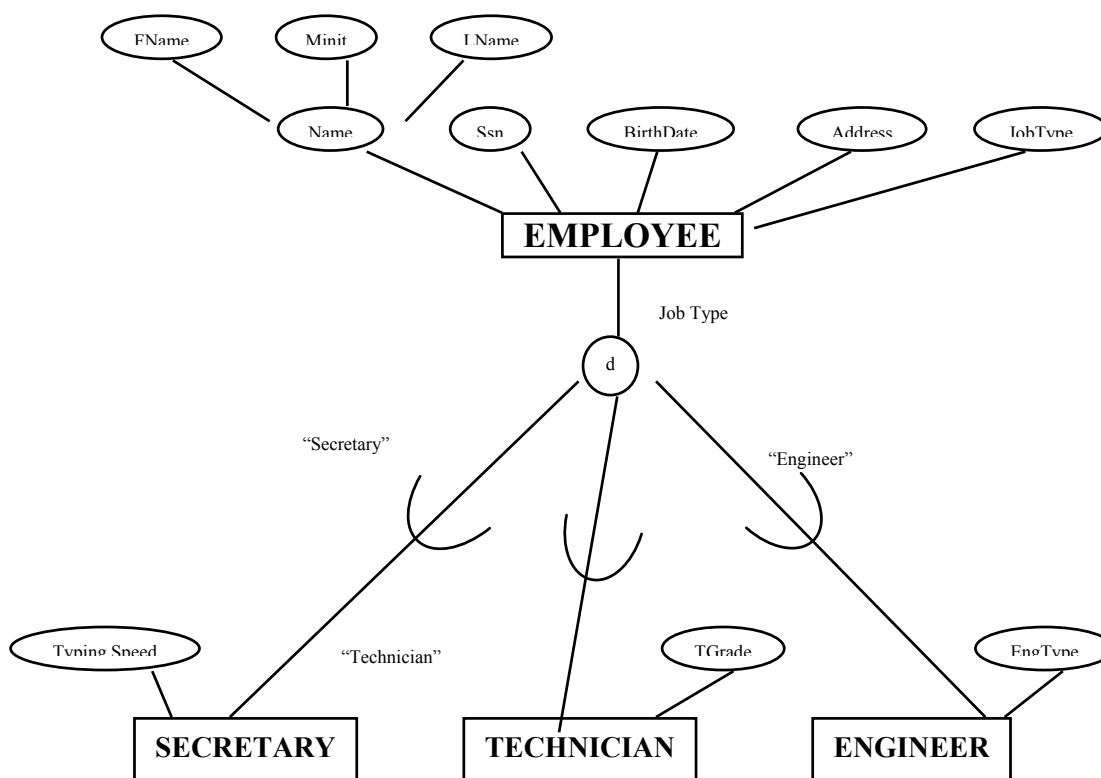
3.7.1 Thực thể con và thực thể chính

Trong mô hình dữ liệu, cần phải mô tả một cách rõ ràng các thực thể gần như nhau: đó là các thực thể có các thuộc tính chung, nhưng cũng có một số các thuộc tính khác nhau.

Ví dụ: Trong một Đơn vị có 3 loại nhân viên: SECRETARY, TECHNICIAN, ENGINEER. Các thực thể này có một số thuộc tính chung và một số thuộc tính riêng.

Trong trường hợp này, có thể có 3 hướng giải quyết sau:

- Cách đơn giản nhất là gộp tất cả các loại nhân viên vào một thực thể EMPLOYEE. Cách này dẫn đến dư thừa thông tin, vì có những thuộc tính luôn rỗng đối với mỗi loại nhân viên.
- Cách thứ hai, định nghĩa riêng rẽ từng loại thực thể: SECRETARY, TECHNICIAN, ENGINEER. Cách này không khai thác được những thuộc tính chung của nhân viên.
- Cách thứ ba, định nghĩa một thực thể chính gọi là EMPLOYEE, với 3 thực thể con là: SECRETARY, TECHNICIAN, ENGINEER. Những thuộc tính chung nằm trong thực thể chính, còn thực thể con sẽ chứa thuộc tính riêng của nó. Hình sau đây minh họa cách giải quyết này:



Hình 3.11. Minh họa mối quan hệ giữa thực thể con và thực thể chính

3.7.2 Các thực thể con loại trừ

Trong mô hình trên, các thực thể con là loại trừ lẫn nhau.

Thực thể con loại trừ gồm 2 loại:

Thực thể con đầy đủ: Là tất cả các thực thể con xác định một thực thể chính. Trong ví dụ trên, tất cả các thực thể con là đầy đủ vì không thể bỏ sung thực thể nào vào thực thể chính EMPLOYEE.

Thực thể con không đầy đủ: Tập các thực thể con không đầy đủ xác định thực thể chính. Ví dụ: Tập thực thể ÔTÔ, XEMAY chưa xác định được thực thể chính là PHUONGTIEN.

3.8 Các ký hiệu và quy ước đặt tên trong mô hình ER

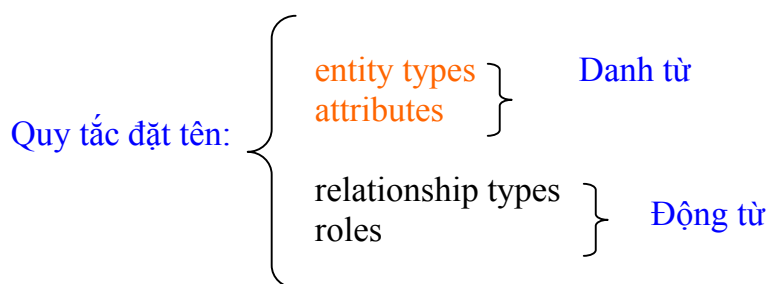
3.8.1 Các ký hiệu

Trong xây dựng mô hình E-R, ta sử dụng các ký hiệu trong hình 3.12:

Symbol	Meaning
	ENTITY Thực thể
	WEAK ENTITY Thực thể yếu
	RELATIONSHIP Mối quan hệ
	IDENTIFYING RELATIONSHIP Mối quan hệ xác định
	ATTRIBUTE Thuộc tính
	KEY ATTRIBUTE Thuộc tính khóa
	MULTIVALUED Thuộc tính đa trị
	COMPOSITE ATTRIBUTE Thuộc tính tổ hợp
	DERIVED ATTRIBUTE Thuộc tính suy diễn
	TOTAL PARTICIPATION OF E ₂ IN R E ₂ tham gia toàn bộ trong R
	CARDINALITY RATIO 1 : N FOR E ₁ :E ₂ IN R Tỷ số tham gia liên kết 1:N
	STRUCTURAL CONSTRAINT (min, max) ON PARTICIPATION OF E IN R Lực lượng của E ₂ trong R

Hình 3.12. Bảng tổng hợp các ký hiệu trong mô hình ER

3.8.2 Quy tắc đặt tên



3.9 Xây dựng một mô hình ER.

3.9.1 Các bước xây dựng sơ đồ ER

3.9.1.1 Liệt kê, chính xác hóa và lựa chọn thông tin cơ sở

Xác định một từ điển bao gồm tất cả các thuộc tính (không bỏ sót bất cứ thông tin nào).

Chính xác hóa các thuộc tính đó. Thêm các từ cần thiết để thuộc tính đó mang đầy đủ ý nghĩa, không gây nhầm lẫn, hiểu nhầm.

Chú ý: Để lựa chọn các đặc trưng cần thiết, ta duyệt từ trên xuống và chỉ giữ lại những thuộc tính đảm bảo yêu cầu sau:

Thuộc tính cần phải đặc trưng cho một lớp các đối tượng được xét.

Chọn một thuộc tính một lần, nếu lặp lại thì bỏ qua.

Một thuộc tính phải là sơ cấp (Nếu giá trị của nó có thể suy ra từ các thuộc tính khác thì bỏ qua).

3.9.1.2 Xác định các thực thể và các thuộc tính của nó, sau đó xác định thuộc tính định danh cho từng thực thể.

Duyệt danh sách các thuộc tính từ trên xuống để tìm ra thuộc tính tên gọi. Mỗi thuộc tính tên gọi sẽ tương ứng với một thực thể.

Gán các thuộc tính cho từng thực thể.

Xác định thuộc tính định danh cho từng thực thể.

3.9.1.3 Xác định các mối quan hệ và các thuộc tính riêng của nó

Xét danh sách các thuộc tính còn lại, hãy tìm tất cả các động từ (ứng với thuộc tính đó). Với mỗi động từ, hãy trả lời các câu hỏi: Ai? Cái gì? Ở đâu? Khi nào? Bằng cách nào?

3.9.1.4 Vẽ sơ đồ mô hình thực thể- mối quan hệ, xác định lực lượng tham gia liên kết cho các thực thể.

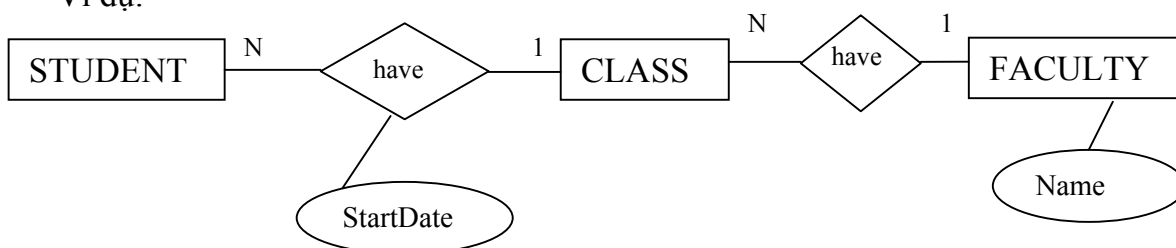
3.9.1.5 Chuẩn hóa sơ đồ và thu gọn sơ đồ

- Vẽ sơ đồ.
- Chuẩn hóa sơ đồ, nếu trong đó còn có chứa: các thuộc tính lặp, nhóm lặp và các thuộc tính phụ thuộc thời gian → sơ đồ chỉ còn các thực thể đơn và các thuộc tính đơn.

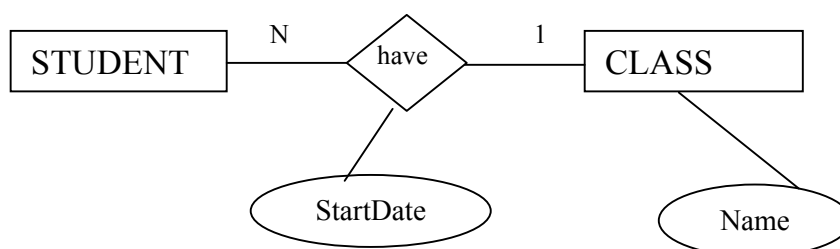
Thu gọn sơ đồ: Nếu một thực thể có tất cả các đặc trưng:

- Là thực thể treo: là thực thể chỉ tham gia vào một mối quan hệ và chỉ chứa một thuộc tính duy nhất thực sự là của nó (có thể có thuộc tính thứ 2 thêm vào làm định danh).
- Mối quan hệ là bậc hai và không có thuộc tính riêng.
- Mối quan hệ là 1: N hay 1:1.

Ví dụ:



Được thu gọn thành sơ đồ sau:



3.9.2 Mô hình ER cho cơ sở dữ liệu COMPANY

- Qua Bước 1 và Bước 2 ta xác định được danh sách **các thực thể và các thuộc tính** của từng thực thể.
- Qua bước 3 ta xác định được các **kiểu liên kết** như sau:

1. **MANAGES**: là kiểu liên kết 1:1 giữa hai kiểu thực thể EMPLOYEE và DEPARTMENT. Lực lượng tham gia liên kết của kiểu thực thể EMPLOYEE là bộ phận, vì không phải nhân viên nào cũng tham gia quản lý. Còn lực lượng tham gia của DEPARTMENT là toàn bộ, vì tại bất kỳ thời điểm nào một phòng cũng có một nhân viên làm quản lý. Thuộc tính StartDate được gắn vào kiểu liên kết để ghi lại thời điểm bắt đầu làm quản lý của nhân viên cho phòng đó.

2. **WORKS_FOR**: là kiểu liên kết 1:N giữa hai kiểu thực thể DEPARTMENT và EMPLOYEE. Cả hai kiểu thực thể này đều có lực lượng tham gia toàn bộ vào liên kết.

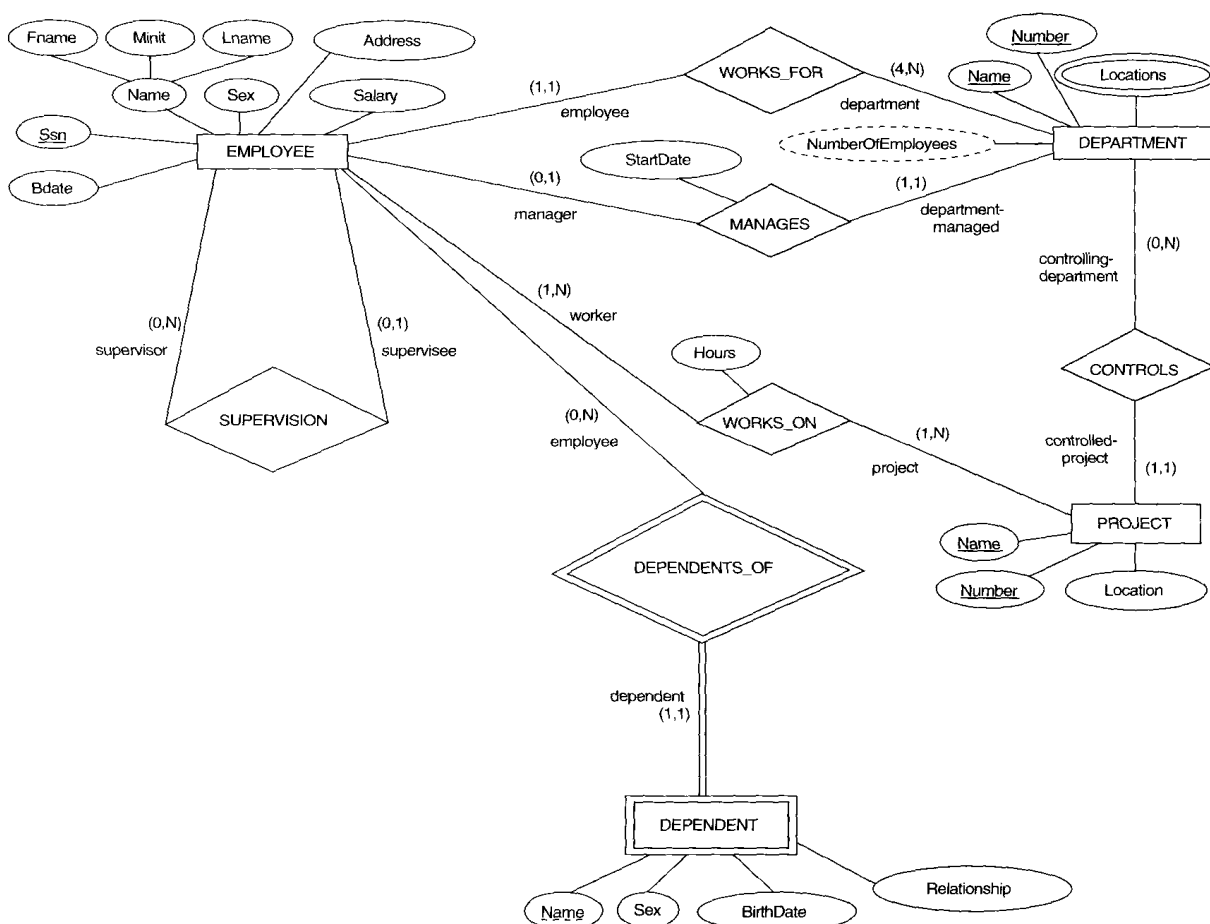
3. **CONTROLS**: là kiểu liên kết 1:N giữa hai kiểu thực thể DEPARTMENT và PROJECT. Lực lượng tham gia của PROJECT là toàn bộ, của DEPARTMENT là bộ phận.

4. **SUPERVISOR**: là kiểu liên kết 1:N giữa hai kiểu thực thể EMPLOYEE và EMPLOYEE (Mỗi nhân viên có người quản lý cấp trên của mình, người đó cũng là một nhân viên). Trong quá trình phỏng vấn các đối tượng người dùng, người thiết kế được trả lời rằng: Không phải nhân viên nào cũng làm quản lý nhân viên khác, và không phải nhân viên nào cũng có người quản lý trực tiếp mình. Vì vậy, cả hai kiểu thực thể này có lực lượng tham gia bộ phận.

5. **WORK_ON**: là kiểu liên kết M:N giữa hai kiểu thực thể EMPLOYEE và PROJECT, vì một dự án có nhiều nhân viên làm việc và một nhân viên có thể làm việc cho nhiều dự án. Thuộc tính Hours là thuộc tính của kiểu liên kết được dùng để ghi lại số giờ mỗi nhân viên làm việc cho một dự án nào đó. Cả hai kiểu thực thể này có lực lượng tham gia toàn bộ.

6. **DEPENDENTS_OF**: là kiểu liên kết 1:N giữa hai kiểu thực thể EMPLOYEE và DEPENDENT. Kiểu thực thể DEPENDENT là kiểu thực thể yếu, vì nó không tồn tại nếu không có sự tồn tại của kiểu thực thể EMPLOYEE. Lực lượng tham gia của EMPLOYEE là bộ phận, vì không phải nhân viên nào cũng có người phụ thuộc. Lực lượng tham gia của DEPENDENT là toàn bộ vì nó là kiểu thực thể yếu.

c. Qua bước 4 ta vẽ được mô hình ER:



Hình 3.13. Mô hình ER của bài toán COMPANY

3.9.3 Bài tập

Xây dựng mô hình ER để quản lý các đơn vị sau:

Bài tập 1: Quản lý hoạt động của một trung tâm đại học

Qua quá trình khảo sát, điều tra hoạt động của một trung tâm đại học ta rút ra các quy tắc quản lý sau:

- Trung tâm được chia làm nhiều **trường** và mỗi trường có 1 **hiệu trưởng** để quản lý nhà trường.
- Một trường chia làm nhiều **khoa**, mỗi khoa thuộc về một trường.
- Mỗi khoa cung cấp nhiều **môn học**. Mỗi môn học thuộc về 1 khoa (thuộc quyền quản lý của 1 khoa).
- Mỗi khoa thuê nhiều giáo viên làm việc. Nhưng mỗi **giáo viên** chỉ làm việc cho 1 khoa. Mỗi khoa có 1 chủ nhiệm khoa, đó là một giáo viên.
- Mỗi giáo viên có thể dạy nhiều nhất 4 môn học và có thể không dạy môn học nào.
- Mỗi sinh viên có thể học nhiều môn học, nhưng ít nhất là môn. Mỗi môn học có thể có nhiều sinh viên học, có thể không có sinh viên nào.
- Một khoa quản lý nhiều sinh viên chỉ thuộc về một khoa.
- Mỗi giáo viên có thể được cử làm chủ nhiệm của lớp, lớp đó có thể có nhiều nhất 100 sinh viên.

Bài tập 2:

Cho các thuộc tính, các quy tắc quản lý của một đơn vị.

1. Thuộc tính:

- Mã đơn vị, Tên đơn vị, Số điện thoại đơn vị, Địa chỉ đơn vị.
- Mã nhân viên, Tên nhân viên, Giới tính nhân viên, Địa chỉ nhân viên, Số điện thoại của nhân viên.
- Mã dự án, Tên dự án
- Mã khách hàng, tên khách hàng, Địa chỉ khách hàng, Số điện thoại của khách hàng.
- Mã hàng, Tên hàng, Số lượng trong kho.
- Lượng đặt hàng, Ngày đặt hàng

2. Các quy tắc

- Một đơn vị thuê 1 hoặc nhiều nhân viên
- Một đơn vị được quản lý bởi 1 người quản lý. Đó là một nhân viên.
- Một nhân viên chỉ làm việc cho 1 đơn vị
- Một nhân viên có thể làm việc cho 1 dự án
- Mỗi dự án có thể thuê 1 hoặc nhiều nhân viên

- Một nhân viên có thể phục vụ cho 1 hoặc nhiều khách hàng
- Một khách hàng có thể được 1 hoặc nhiều nhân viên phục vụ
- Một khách hàng có thể đặt 1 hoặc 1 vài hàng hóa (Khách hàng nào cũng đặt hàng: 1 hoặc nhiều mặt hàng)
- Mọi mặt hàng đều có ít nhất một khách hàng đặt mua
- Một đơn đặt hàng chỉ có 1 mặt hàng.

4 Chương 4. MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ

4.1 Khái niệm mô hình quan hệ

Mô hình CSDL quan hệ lần đầu tiên được E.F.Codd và tiếp sau đó được công ty IBM giới thiệu vào năm 1970. Ngày nay, hầu hết các tổ chức đã áp dụng CSDL quan hệ để quản lý dữ liệu trong đơn vị mình.

Mô hình cơ sở dữ liệu quan hệ là cách thức biểu diễn dữ liệu dưới dạng bảng hay còn gọi là quan hệ, mô hình được xây dựng dựa trên cơ sở lý thuyết đại số quan hệ.

Cấu trúc dữ liệu: dữ liệu được tổ chức dưới dạng quan hệ hay còn gọi là bảng.

Thao tác dữ liệu: sử dụng những phép toán mạnh (bằng ngôn ngữ SQL).

4.2 Các thành phần cơ bản của mô hình

4.2.1 Một số khái niệm của mô hình quan hệ

Mô hình quan hệ là cách thức biểu diễn dữ liệu dưới dạng các quan hệ (các bảng). Một quan hệ là một bảng dữ liệu 2 chiều (cột và dòng), mô tả một thực thể. Mỗi cột tương ứng với một thuộc tính của thực thể. Mỗi dòng chứa các giá trị dữ liệu của một đối tượng cụ thể thuộc thực thể

Một số khái niệm cơ bản:

Lược đồ quan hệ: $R(A_1, \dots, A_n)$, trong đó R là tên quan hệ, A_i là các thuộc tính, mỗi A_i có miền giá trị tương ứng $dom(A_i)$.

Lược đồ quan hệ được sử dụng để mô tả một quan hệ, bao gồm: Tên quan hệ, các thuộc tính và bậc của quan hệ (số lượng các thuộc tính)

4.2.2 Quan hệ:

Một quan hệ r của $R(A_1, \dots, A_n)$, ký hiệu $r(R)$ là một tập hợp n -bộ $r = \{ t_1, \dots, t_m \}$

Trong đó:

Mỗi $t_i = \langle v_1, \dots, v_n \rangle$, $v_i \in dom(A_i)$.

$r(R) \subseteq dom(A_1) \times \dots \times dom(A_n)$

$r = \{ (v_{i1}, v_{i2}, \dots, v_{in}) / i=1, \dots, m \}$

v_{11}	v_{12}	\dots	v_{1n}
v_{21}	v_{22}	\dots	v_{2n}
\dots	\dots	\dots	\dots
v_{m1}	v_{m2}	\dots	v_{mn}
A_1	A_2	\dots	A_m

Ta có A_i là các thuộc tính và miền giá trị của A_i là:

$D_1 = dom(A_1), D_2 = dom(A_2), \dots, D_n = dom(A_n)$.

Chú ý: - Các tập (D_1, D_2, \dots, D_n) là tập các miền trị của R

- n được gọi là bậc của quan hệ r .
- m được gọi là lực lượng của r .
- Quan hệ bậc 1 là quan hệ nhất nguyên, bậc 2 là quan hệ nhị nguyên, bậc n là quan hệ n nguyên.

Ví dụ: Quan hệ EMPLOYEE trên tập các thuộc tính $R = \{SSN, Name, BDate, Address, Salary\}$ là một quan hệ 5 ngôi.

SSN	Name	BDate	Address	Salary	
001	Đỗ Hoàng Minh	1960	Hà nội	425	t1
002	Đỗ Như Mai	1970	Hải Phòng	390	t2
003	Đặng Hoàng Nam	1973	Hà nội	200	t3

$t1(001, 'Đỗ Hoàng Minh', 1960, 'Hà nội', 425) = t1(R)$ là một bộ của quan hệ EMPLOYEE

4.2.3 Các tính chất của một quan hệ

- Giá trị đưa vào cột là đơn nhất
- Các giá trị trong cùng một cột phải thuộc cùng một miền giá trị (cùng kiểu)
- Thứ tự dòng cột tùy ý.

4.2.4 Các ràng buộc toàn vẹn trên quan hệ

Ràng buộc là những quy tắc được áp đặt lên trên dữ liệu đảm bảo *tính tin cậy* và *độ chính xác* của dữ liệu. Các luật toàn vẹn được thiết kế để giữ cho dữ liệu phù hợp và đúng đắn.

Có 4 kiểu ràng buộc chính: Ràng buộc miền giá trị (Domain Constraints), Ràng buộc khoá (Key Constraints), Ràng buộc thực thể (Entity Integrity Constraints), và Ràng buộc toàn vẹn tham chiếu (Referential Integrity Constraints).

4.2.4.1 Ràng buộc miền giá trị

Là một hợp các kiểu dữ liệu và những giá trị giới hạn mà thuộc tính có thể nhận được. Thông thường việc xác định miền giá trị của các thuộc tính bao gồm một số các yêu cầu sau: *Tên thuộc tính, Kiểu dữ liệu, Độ dài dữ liệu, khuôn dạng của dữ liệu, các giá trị giới hạn cho phép, ý nghĩa, có duy nhất hay không, có cho phép giá trị rỗng hay không.*

4.2.4.2 Ràng buộc khoá

4.2.4.2.1 Khóa chính (Primary Key)

Khóa chính là một (hoặc một tập) các thuộc tính đóng vai trò là nguồn của một phụ thuộc hàm mà đích lần lượt là các thuộc tính còn lại.

Ví dụ: $R = \{SSN, Name, BDate, Address, Salary\}$

$SSN \rightarrow Name, BDate, Address, Salary$

(Nguồn) \rightarrow (Đích)

Ta thấy, từ SSN ta có thể suy ra toàn bộ các thuộc tính ứng. Vậy SSN được gọi là khóa chính.

Một số gợi ý khi chọn khóa:

- Khóa không nên là tập hợp của quá nhiều thuộc tính. Trong trường hợp khóa có nhiều thuộc tính, có thể thêm một thuộc tính “nhân tạo” thay chúng làm khóa chính cho quan hệ.

- Nếu khóa chính được cấu thành từ một số thuộc tính, thì các thành phần nên tránh sử dụng thuộc tính có giá trị thay đổi theo thời gian: như tên địa danh, phân loại.

4.2.4.2.2 Khóa dự tuyển (Candidate Key)

Trong tập hợp các thuộc tính của một bảng, có thể có nhiều thuộc tính có thể dùng được làm khóa chính. Các thuộc tính đó được gọi là khóa dự tuyển.

Khóa dự tuyển cần thỏa mãn 2 tính chất sau:

- Xác định duy nhất.
- Không dư thừa: Khi xóa đi bất kỳ một thuộc tính nào của khóa đều phá hủy tính xác định duy nhất của khóa.

4.2.4.2.3 Khóa ngoại (Foreign Key)

Trong nhiều trường hợp, khóa chính của một bảng được đưa sang làm thuộc tính bên bảng khác, thuộc tính đó gọi là khóa ngoại. Khóa ngoại đóng vai trò thể hiện liên kết giữa 2 bảng.

4.2.4.2.4 Khóa phụ (Second Key)

Đóng vai trò khi ta muốn sắp xếp lại dữ liệu trong bảng.

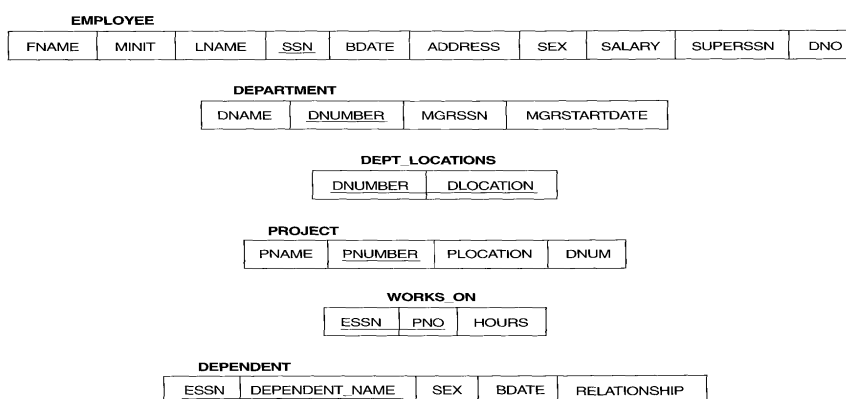
Ví dụ: Ta có bảng SINHVIEN (MaSV, Hoten, GioiTinh, Diem).

Muốn sắp xếp lại danh sách sinh viên theo thứ tự a, b, c.. của Họ tên. Khi đó thuộc tính Hoten được gọi là khóa phụ.

4.2.4.3 Ràng buộc thực thể

Mỗi một lược đồ quan hệ R, chúng ta phải xác định khoá chính của nó. Khoá chính trong lược đồ quan hệ được gạch chân ở phía dưới của thuộc tính.

Sau đây là danh sách các lược đồ quan hệ trong cơ sở dữ liệu COMPANY sau khi xác định ràng buộc thực thể:



Hình 4.1. Lược đồ cơ sở dữ liệu COMPANY

Chương 4. MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	1985-01-09	731 Fondren, Houston, TX	M	30000	333445555	5	
Franklin	T	Wong	333445555	1965-12-08	638 Voss, Houston, TX	M	40000	888665555	5	
Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4	
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4	
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5	
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5	
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4	
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1	

DEPT_LOCATIONS	DNUMBER	DLOCATION
	1	Houston
	4	Stafford
	5	Bellaire
	5	Sugarland
	5	Houston

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	1988-05-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-19

WORKS_ON	ESSN	PNO	HOURS
	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40.0
	453453453	1	20.0
	453453453	2	20.0
	333445555	2	10.0
	333445555	3	10.0
	333445555	10	10.0
	333445555	20	10.0
	999887777	30	30.0
	999887777	10	10.0
	987987987	10	35.0
	987987987	30	5.0
	987654321	30	20.0
	987654321	20	15.0
	888665555	20	null

PROJECT	PNAME	PNUMBER	PLOCATION	DNUM
	ProductX	1	Bellaire	5
	ProductY	2	Sugarland	5
	ProductZ	3	Houston	5
	Computerization	10	Stafford	4
	Reorganization	20	Houston	1
	Newbenefits	30	Stafford	4

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
	333445555	Alice	F	1986-04-05	DAUGHTER
	333445555	Theodore	M	1983-10-25	SON
	333445555	Joy	F	1958-05-03	SPOUSE
	987654321	Abner	M	1942-02-28	SPOUSE
	123456789	Michael	M	1988-01-04	SON
	123456789	Alice	F	1988-12-30	DAUGHTER
	123456789	Elizabeth	F	1967-05-05	SPOUSE

Hình 4.2. Một thể hiện của cơ sở dữ liệu COMPANY

Lưu ý: Ràng buộc khoá và ràng buộc thực thể được xác định cho từng quan hệ.

4.2.4.4 Ràng buộc toàn vẹn tham chiếu

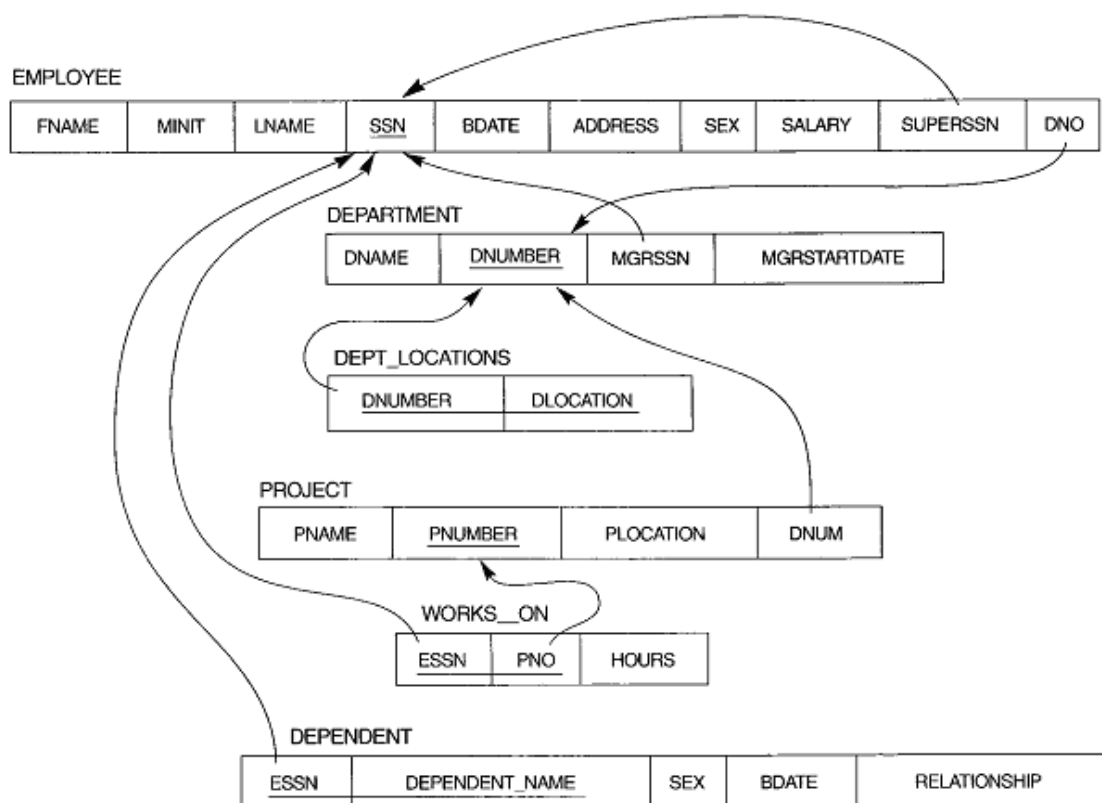
Một bộ giá trị trong một quan hệ tham chiếu tới một bộ giá trị đã tồn tại trong một quan hệ khác.

Ràng buộc toàn vẹn tham chiếu phải xác định trên 2 quan hệ: quan hệ tham chiếu (**referencing relation**) và quan hệ được tham chiếu (**referenced relation**).



Ràng buộc toàn vẹn tham chiếu còn được gọi là ràng buộc khoá ngoại.

Ví dụ: Thuộc tính DNo của quan hệ EMPLOYEE tham chiếu tới thuộc tính DNumber của quan hệ DEPARTMENT.



Hình 4.3. Các ràng buộc tham chiếu trong cơ sở dữ liệu COMPANY

4.2.5 Các phép toán trên CSDL quan hệ

4.2.5.1 Phép toán cập nhật

a. Phép chèn (INSERT): Là phép bổ xung thêm một bộ vào quan hệ r cho trước.

+ Biểu diễn: INSERT(r; $A_1=d_1, A_2=d_2, \dots, A_n=d_n$) với A_i là thuộc tính, di thuộc $\text{dom}(A_i)$, $i=1, \dots, n$.

Nếu thứ tự các trường là cố định, có thể biểu diễn phép chèn dưới dạng không trường minh INSERT(r; d_1, d_2, \dots, d_n).

+ Ví dụ : Chèn thêm một bộ $t_4=(\text{'004'}, \text{'Hoàng Thanh Vân'}, 1969, \text{'Hà nội'}, 235)$ vào quan hệ EMPLOYEE(SSN, Name, BDate, Address, Salary) ta có thể viết:

INSERT(EMPLOYEE; SSN= '004', Name= 'Hoàng Thanh Vân', BDate=1969, Address= 'Hà nội', Salary=235).

+ Chú ý : Kết quả của phép chèn có thể gây ra một số sai sót là :

- Bộ mới được thêm không phù hợp với lược đồ quan hệ cho trước
- Một số giá trị của một số thuộc tính nằm ngoài miền giá trị của thuộc tính đó.
- Giá trị khoá của bộ mới có thể là giá trị đã có trong quan hệ đang lưu trữ.

b. Phép loại bỏ (DEL): Là phép xoá một bộ ra khỏi một quan hệ cho trước.

- Biểu diễn : DEL(r; $A_1=d_1, A_2=d_2, \dots, A_n=d_n$) hay DEL((r, d_1, d_2, \dots, d_n).

Nếu $K=(E_1, E_2, \dots, E_m)$ là khoá thì có thể viết DEL(r; $E_1=e_1, E_2=e_2, \dots, E_m=e_m$)

- Ví dụ :

+ Để xoá bộ t_1 ra khỏi quan hệ r:

DEL(EMPLOYEE; SSN= '004', Name= 'Hoàng Thanh Vân', BDate=1969, Address= 'Hà nội', Salary=235).

+ Cần loại bỏ một nhân viên trong quan hệ EMPLOYEE mà biết SSN đó là '004' thì chỉ cần viết: DEL(EMPLOYEE; SSN= '004')

c. Phép cập nhật (UPDATE): Là phép tính dùng để sửa đổi một số giá trị nào đó tại một số thuộc tính.

+ Biểu diễn :

UPD (r; $A_1=d_1, A_2=d_2, \dots, A_n=d_n; B_1=b_1, B_2=b_2, \dots, B_k=b_k$)

Với $\{B_1, B_2, \dots, B_k\}$ là tập các thuộc tính mà tại đó các giá trị của bộ cần thay đổi. $\{B_1, B_2, \dots, B_k\}$ ứng với tập thuộc tính $\{A_1, A_2, \dots, A_n\}$

Hay UPD(r; $E_1=e_1, E_2=e_2, \dots, E_m=e; B_1=b_1, B_2=b_2, \dots, B_k=b_k$) với $K=(E_1, E_2, \dots, E_m)$ là khoá.

+ Ví dụ : Để thay đổi tên nhân viên có SSN= '003' trong quan hệ EMPLOYEE thành Nguyễn Thanh Mai ta có thể viết :

CH (EMPLOYEE; SSN= '03'; Name= 'Nguyễn Thanh Mai')

4.2.5.2 Phép toán đại số quan hệ

Đại số quan hệ gồm một tập các phép toán tác động trên các quan hệ và cho kết quả là một quan hệ.

Có 8 phép toán được chia làm 2 nhóm : Nhóm các phép toán tập hợp (hợp, giao, trừ, tích chập), nhóm các phép toán quan hệ (chọn, chiếu, kết nối, chia).

Định nghĩa : Hai quan hệ r và s được gọi là khả hợp nếu chúng được xác định trên cùng một tập các miền giá trị (Có nghĩa là chúng được xác định trên cùng một tập các thuộc tính).

a. Phép hợp:

- Phép hợp của hai quan hệ khả hợp $r \cup s = \{t / t \text{ thuộc } r \text{ hoặc } t \text{ thuộc } s\}$

Ví dụ:

$r(A, B, C)$	$s(A, B, C)$	$r \cup s(A, B, C)$
a1 b1 c1	a1 b1 c1	a1 b1 c1
a2 b2 c2	a1 b2 c2	a2 b2 c2
		a1 b2 c2

- Phép hợp của hai quan hệ là phép gộp các bộ của hai bảng của một quan hệ thành một bảng và bỏ đi các bộ trùng.

Ví dụ: EMPLOYEE1

SSN	Name	DNo
001	Hoàng	P001
002	Thiện	P002

EMPLOYEE2

SSN	Name	DNo
003	Huy	P001
002	Thiện	P002
004	Thiện	P003

EMPLOYEE1 \cup EMPLOYEE2 = EMPLOYEE3

SSN	Name	DNo
001	Hoàng	P001
002	Thiện	P002
003	Huy	P001
004	Thiện	P003

Hình 4.4. Minh họa dữ liệu phép hợp 2 quan hệ

b. Phép giao

- Phép giao của hai quan hệ khả hợp $r \cap s = \{t / t \text{ thuộc } r \text{ và } t \text{ thuộc } s\}$

Ví dụ :

$r \cap s(A, B, C)$
a1 b1 c1

- Phép giao của hai quan hệ là lấy ra các bộ cùng có mặt ở cả hai bảng của một quan hệ.

Ví dụ: EMPLOYEE1 \cap EMPLOYEE2 = 002, Thiện, P002

c. Phép trừ

- Phép trừ của hai quan hệ khả hợp $r - s = \{t / t \text{ thuộc } r \text{ và } t \text{ không thuộc } s\}$

Ví dụ :

$r - s(A, B, C)$
a2 b2 c2

Bậc $r =$ bậc r' ; lực lượng của $r \geq$ lực lượng của r'

-Phép chọn trên quan hệ là lấy ra các dòng của bảng quan hệ thoả mãn một điều kiện nào đó trên tập các cột thuộc tính.

Ví dụ : Chọn trên quan hệ EMPLOYEE3 các nhân viên thuộc phòng có DNo=P001

SSN	Name	DNo
001	Hoàng	P001
003	Huy	P001

f. Phép chiếu (cắt dọc) - 1 ngôi

- Là phép toán loại bỏ đi một số thuộc tính và chỉ giữ lại một số thuộc tính được chỉ ra của một quan hệ.

- Cho quan hệ $r(R)$, X là tập con của tập thuộc tính R . Phép chiếu của quan hệ r trên X : $\Pi_X(r) = \{ t[X] / \text{thuộc } r \}$; $t[X]$ là bộ t lấy trên tập thuộc tính X .

Ví dụ : Cho $r(A,B)$ như trên , $X=\{A\}$;

$$\begin{aligned} \Pi_X(r) &= u(A) \\ & \quad a1 \\ & \quad a2 \\ & \quad a3 \end{aligned}$$

- Bậc của $r >$ bậc của k . Lực lượng của $r >$ lực lượng của k

- Phép chiếu trên quan hệ là lấy một số cột (thuộc tính) nào đó của bảng quan hệ.

Ví dụ : Lấy danh sách mã NV của quan hệ NHANVIEN

$$\Pi_{SSN}(\text{EMPLOYEE3}) = \begin{array}{|c|} \hline \text{SSN} \\ \hline 001 \\ \hline 002 \\ \hline 003 \\ \hline 004 \\ \hline \end{array}$$

g. Phép kết nối - 2 ngôi

1. Phép kết nối

- Cho hai quan hệ $r(R)$, $R=\{A_1,A_2,\dots,A_n\}$ và quan hệ $s(U)$, $U=\{B_1,B_2,\dots,B_m\}$.

- Phép xếp cạnh nhau: cho hai bộ $d = (d_1,d_2,\dots,d_n)$ và $e = (e_1,e_2,\dots,e_m)$ phép xếp cạnh nhau của d và e là : $(d \wedge e) = (d_1,d_2,\dots,d_n, e_1,e_2,\dots,e_m)$

- Phép kết nối giữa quan hệ r có thuộc tính A và quan hệ s có thuộc tính B với một phép so sánh θ là :

$$r \bowtie_{\theta} s = \{ a \wedge b / a \text{ thuộc } r, b \text{ thuộc } s \text{ và } a(A) \theta b(B) \}$$

Ví dụ : Xét quan hệ r và s trong ví dụ phép tích đề các

$r(A, B)$;	$s(C, D)$;	$r \times s = k(A, B, C, D)$
a1 1	1 d1	($B \geq C$) a1 1 1 d1
a2 2	2 d2	a1 1 2 d2
a3 3		a2 2 2 d2
		a3 3 1 d1
		a3 3 2 d2

- Lực lượng của phép kết nối $k' \leq$ lực lượng của phép tích đề các k.

$$k' = \sigma_{B \geq C}(k) \rightarrow r \bowtie s = \sigma_F(r \times s)$$

- Chú ý :

+ Để phép kết nối có nghĩa, miền trị dom(A) phải so sánh được qua phép so sánh θ với miền trị dom(B)

+ Nếu phép so sánh θ là "=" thì phép kết nối gọi là kết nối bằng.

2. Phép kết nối tự nhiên

Phép toán kết nối bằng trên những thuộc tính cùng tên của hai quan hệ và sau khi kết nối thì cắt bỏ đi một thuộc tính cùng tên bằng phép chiếu của đại số quan hệ được gọi là phép kết nối tự nhiên ký hiệu *.

- Ví dụ :

$r(A, B, C)$	$s(C, D, E)$	$r*s(A, B, C, D, E)$
a1 1 1	1 d1 e1	a1 1 1 d1 e1
a2 2 1	2 d2 e2	a2 2 1 d1 e1
a1 2 2	3 d3 e3	a1 2 2 d2 e2

$r*s = \Pi_{ABCDE}(r \bowtie s)$
 $C=C$

- Ví dụ : Cho hai quan hệ NHANVIEN và PHONG

SSN	Name	DNo
001	Hoàng	P001
002	Thiện	P002
003	Huy	P001
004	Thiện	P003

DNo	DName
P001	Tổ chức
P002	Kinh doanh
P003	Nhân sự
P004	Tiếp thị

Kết nối tự nhiên của hai quan hệ :

SSN	Name	DNo	DName
001	Hoàng	P001	Tổ chức
002	Thiện	P002	Kinh doanh
003	Huy	P001	Tổ chức
004	Thiện	P003	Nhân sự

Hình 4.5. Minh họa dữ liệu phép kết nối tự nhiên 2 quan hệ

h. Phép chia

- Cho r là một quan hệ n- ngôi, s là quan hệ m- ngôi ($n > m$, s khác rỗng). Phép chia quan hệ r cho quan hệ s là tập tất cả các n-m bộ t sao cho với mọi u thuộc s thì bộ (t^u) thuộc r : $r \div s = \{t / \text{với mọi } u \text{ thuộc } s \text{ thì } (t^u) \text{ thuộc } r\}$

Ví dụ :

$r(A, B, C, D)$	$s(C, D)$	$r \div s (A, B)$
a b c d	c d	a b
a b e f	e f	e d
b c e f		
e d c d		
e d e f		
a b d e		

- Ví dụ với hai quan hệ : PRODUCT và SUPPORT

PNo	PName
h1	Đài
h2	TV
h3	tủ lạnh

SNo	PNo	PName
n1	h1	Đài
n1	h2	TV
n1	h3	Tủ lạnh
n2	h3	Tủ lạnh
n2	h1	Đài
n3	h1	Đài
n3	h2	TV
n3	h3	Tủ lạnh
n4	h1	Đài

Cung cấp Mặt hàng =

SNo
n1
n3

Hình 4.6. Minh họa dữ liệu phép chia 2 quan hệ

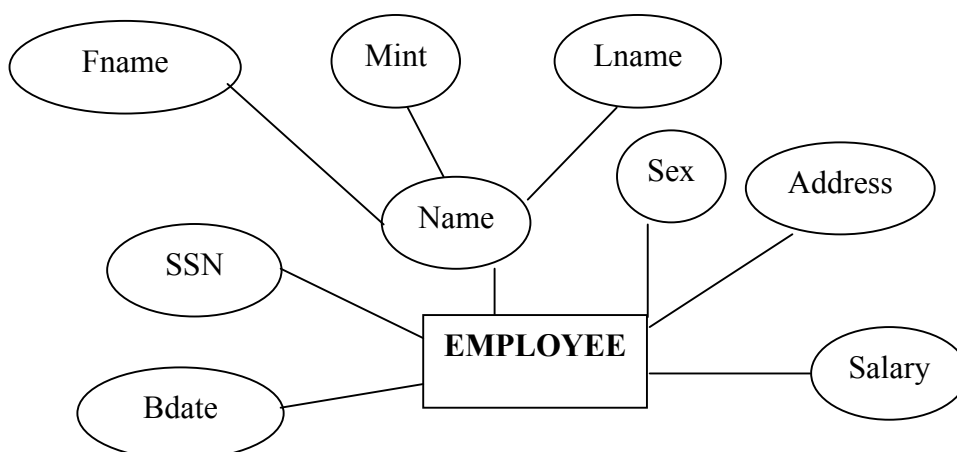
5 Chương 5. CHUYỂN TỪ MÔ HÌNH ER SANG MÔ HÌNH QUAN HỆ

Như chúng ta đã biết, mô hình ER là mô hình dữ liệu mức khái niệm. Sau quá trình khảo sát thiết kế, ta thu được mô hình này. Từ mô hình này ta có thể sử dụng các quy tắc chuyển sang mô quan hệ để thực hiện quản lý cơ sở dữ liệu trên máy tính.

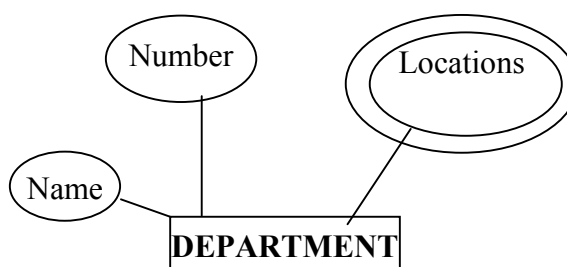
Sau đây là 8 bước được sử dụng để chuyển từ mô hình ER sang mô hình quan hệ:

Bước 1: Mỗi kiểu thực thể bình thường (không phải kiểu thực thể yếu) trong mô hình ER trở thành một quan hệ. Quan hệ đó bao gồm tất cả các thuộc tính đơn giản và thuộc tính tổ hợp của thực thể. Thuộc tính định danh của thực thể là khóa chính của quan hệ.

Ví dụ: Kiểu thực thể: EMPLOYEE, DEPARTMENT, PROJECT
 ==> quan hệ: EMPLOYEE, DEPARTMENT, PROJECT

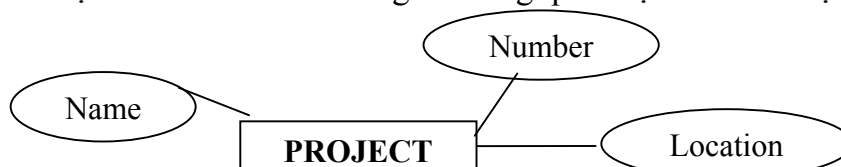


→ Quan hệ: EMPLOYEE (Ssn, fname, minit, lname, bdate, sex, address, salary)



→ Quan hệ: DEPARTMENT (Dnumber, Dname)

Lưu ý: Thuộc tính Locations không có trong quan hệ vì nó là thuộc tính đa trị.

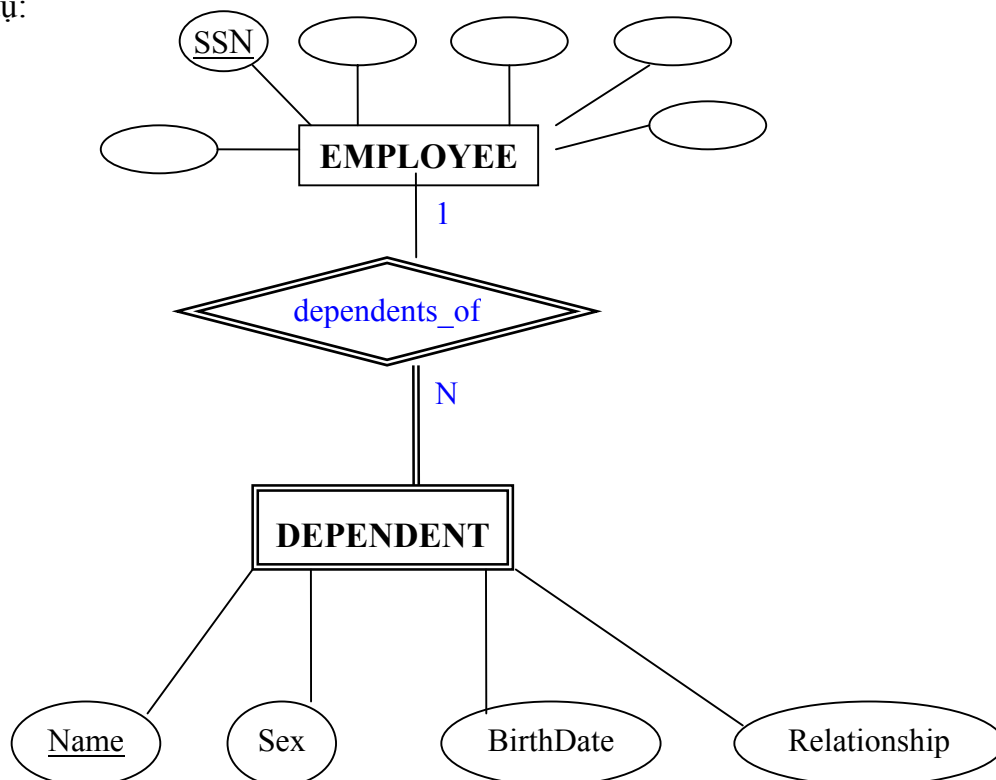


→ Quan hệ: PROJECT (Pnumber, Pname, Plocation)

Bước 2: Cho mỗi thực thể yếu (Weak Entity) trong mô hình ER, tạo thành một quan hệ R, tất cả thuộc tính đơn giản của thực thể yếu trở thành thuộc tính của R. Thêm vào đó, thuộc tính định danh của thực thể chủ trở thành khóa ngoại của R.

Khoá chính của R là sự kết hợp giữa thuộc tính định danh của thực thể chủ và thuộc tính định danh của thực thể yếu.

Ví dụ:

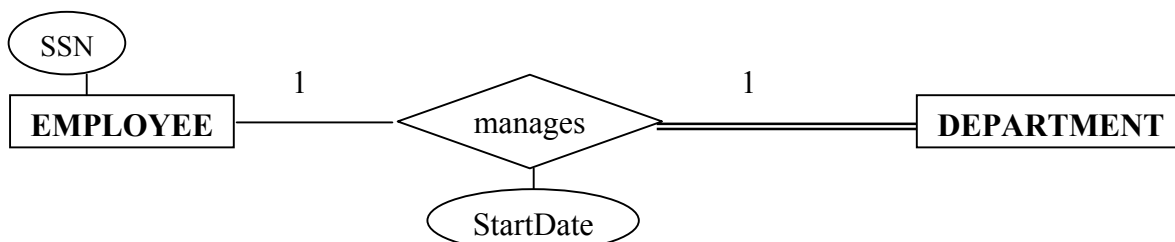


→ **DEPENDENT (Essn, Dependent name, sex, bdate, relationship)**

Bước 3: Cho mỗi mối liên kết 1-1 trong mô hình ER:

- Xác định một quan hệ S_T. Kiểu thực thể có sự tham gia toàn bộ vào liên kết trở thành quan hệ S, thực thể còn lại trở thành quan hệ T.
- Đưa khóa chính của T sang làm khóa ngoại của S.
- Thuộc tính của mỗi quan hệ S_T trở thành thuộc tính của S.

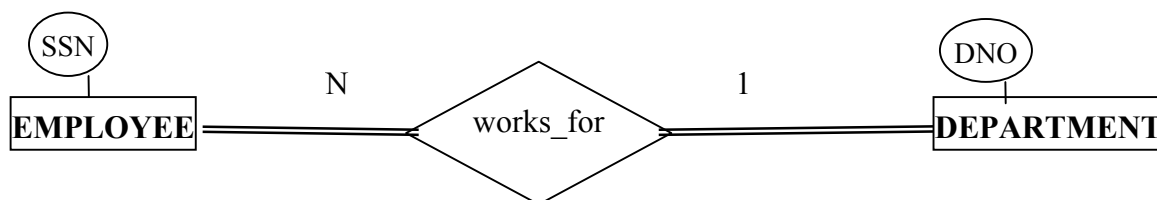
Ví dụ:



→ **DEPARTMENT(..., MGRSSN, MGRSTARTDATE)**

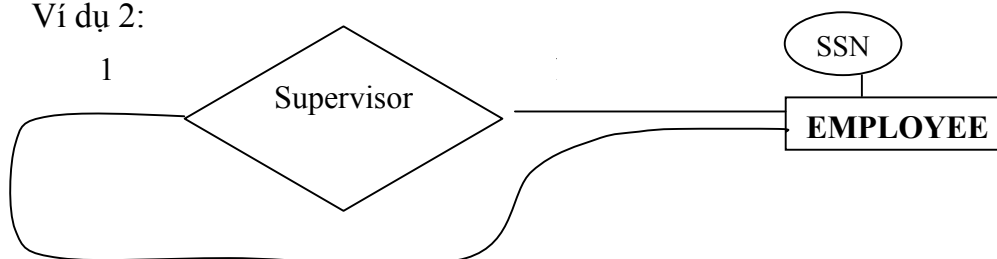
Bước 4: Cho mỗi mối liên kết 1_N trong mô hình ER. Chuyển khóa chính của quan hệ phía 1 sang làm khóa ngoại của quan hệ phía N.

Ví dụ 1:



→ **EMPLOYEE(..., DNO)**

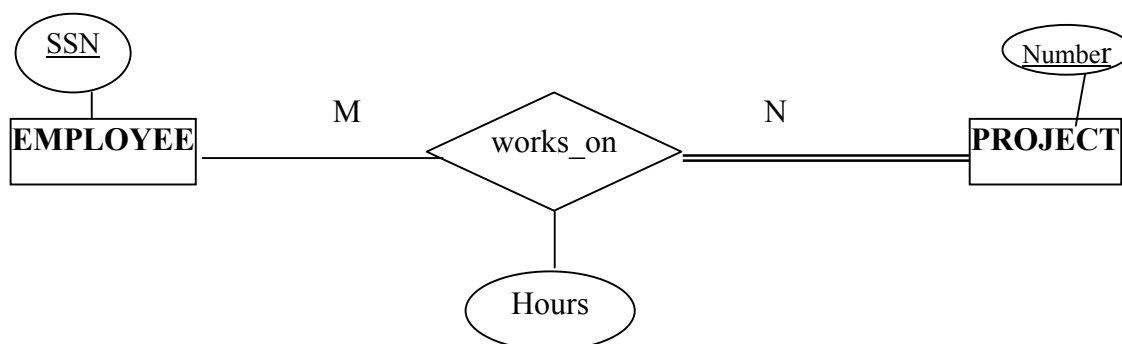
Ví dụ 2:



→ **EMPLOYEE(..., SuperSSN)**

Bước 5: Cho mỗi mối liên kết MN, sinh ra một quan hệ mới R, chuyển khóa chính của hai quan hệ phía M và N thành khóa ngoại của quan hệ R. Khóa chính của R là sự kết hợp của hai khóa ngoại.

Ví dụ:



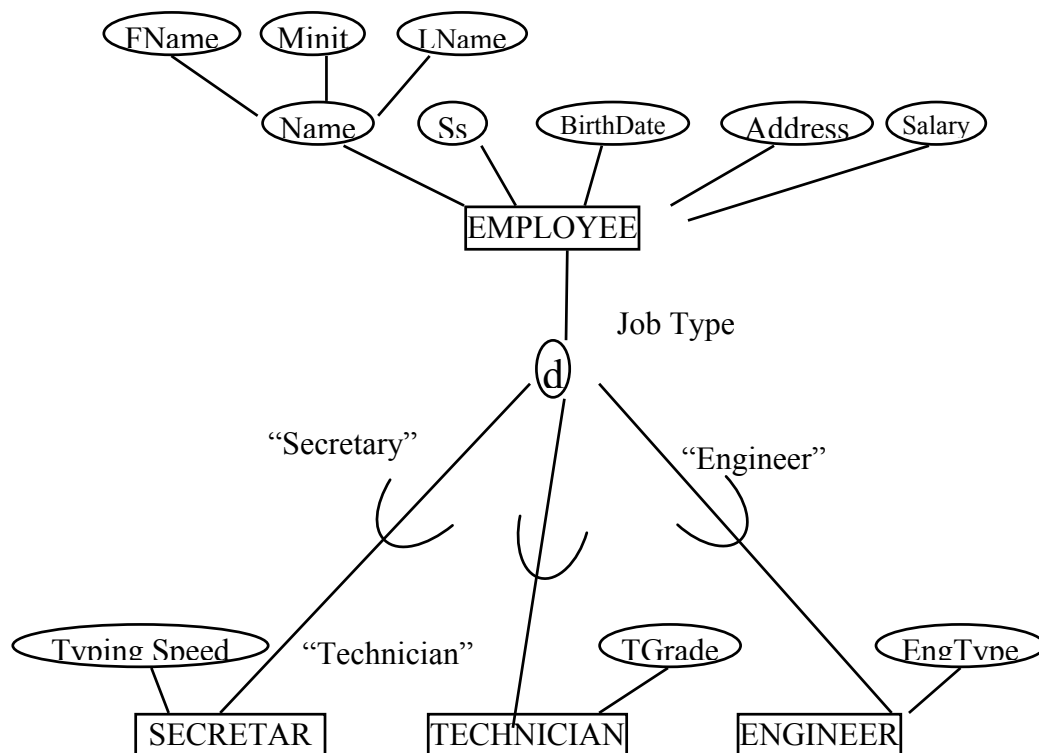
→ **WORKS_ON(PNO, ESSN, Hours)**

Bước 6: Nếu gặp thuộc tính đa trị:

- Chuyển thuộc tính đa trị thành quan hệ mới.
- Thuộc tính định danh (hoặc 1 phần thuộc tính định danh) của thực thể chính chuyển thành khóa ngoại của quan hệ mới.

- Tạo quan hệ L cho lớp cha C với các thuộc tính $Attrs(L)=\{k, a1, \dots, an\}$ và khoá chính của L là: $PK(L)=k$.
- Tạo quan hệ Li cho mỗi lớp con tương ứng Si với các thuộc tính $Attrs(Li)=\{k\} \cup \{thuộc\ tính\ của\ Si\}$ và $PK(Li)=k$.

Ví dụ:



→ Chuyển chuyên biệt hoá trên thành các quan hệ sau:

EMPLOYEE(SSN, FName, Minit, LName, BirthDate, Address, Salary)

SECRETARY(SSN, TypingSpeed)

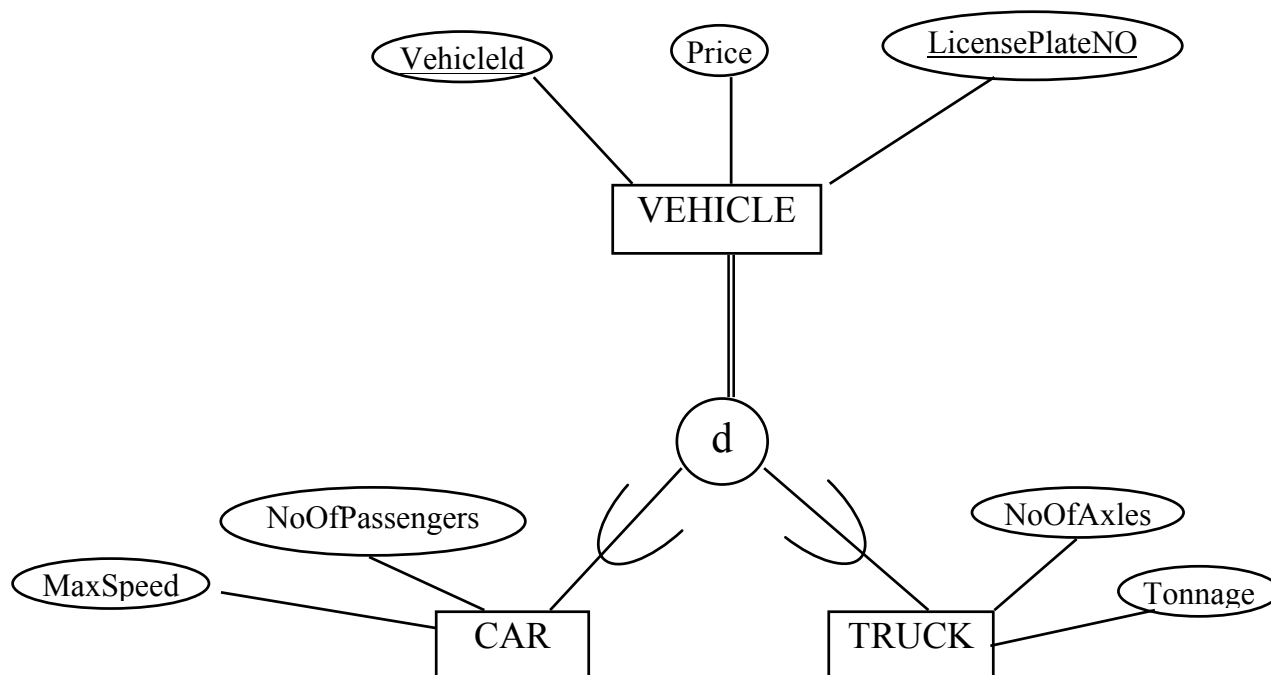
TECHNICIAN(SSN, TGrade)

ENGINEER(SSN, EngType)

2. Lựa chọn 8B:

Tạo một quan hệ Li cho mỗi lớp con Si, với các thuộc tính $Attr(Li) = \{k, a_1, a_2, \dots, a_m\} \cup \{thuộc\ tính\ của\ Si\}$ và $PK(Li) = k$.

Ví dụ:



→ Chuyển chuyên biệt hoá trên thành các quan hệ sau:

CAR(VehicleId, LicensePlateNo, Price, MaxSpeed, NoOfPassengers)

TRUCK(VehicleId, LicensePlateNo, Price, NoOfAxles, Tonnage)

3. Lựa chọn 8C:

Tạo một quan hệ L với các thuộc tính $Attr(L) = \{k, a_1, a_2, \dots, a_n\} \cup \{thuộc\ tính\ của\ S_i\} \cup \dots \cup \{thuộc\ tính\ của\ S_m\} \cup \{t\}$ và $PK(L) = k$. Trong đó, t là thuộc tính phân biệt chỉ ra bản ghi thuộc về lớp con nào, vì thế miền giá trị của $t = \{1, 2, \dots, m\}$.

Ví dụ: Đối với chuyên biệt hoá của EMPLOYEE, ta chỉ tạo ra một quan hệ L như sau:

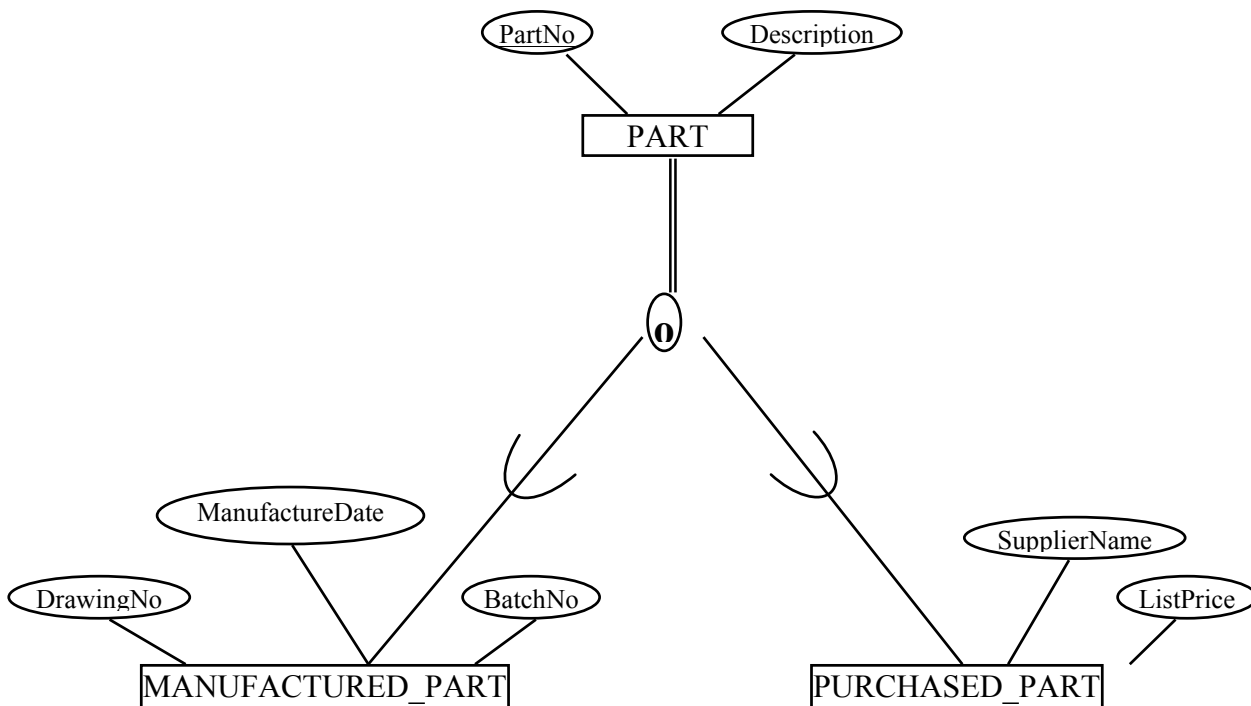
EMPLOYEE(SSN, FName, Minit, LName, BirthDate, Address, Salary, TypingSpeed, Tgrad, EngType, JobType)

Thuộc tính phân biệt

4. Lựa chọn 8D:

Tạo một quan hệ L với các thuộc tính $Attr(L) = \{k, a_1, a_2, \dots, a_n\} \cup \{thuộc\ tính\ của\ S_i\} \cup \dots \cup \{thuộc\ tính\ của\ S_m\} \cup \{t_1, t_2, \dots, t_m\}$ và $PK(L) = k$. Lựa chọn này cho chuyên biệt hoá của các lớp con được nạp chồng (nhưng cũng áp dụng cho một chuyên biệt tách rời), và với mỗi $t_i, 1 \leq i \leq m$, là thuộc tính BOOLEAN chỉ ra bộ theo lớp con S_i .

Ví dụ:



→ PART (ParNo, Description, Mflag, DrawingNo, ManufactureDate, BatchNo, PFlag, SupplierName, ListPrice)

Thuộc tính phân biệt

6 Chương 6. PHỤ THUỘC HÀM VÀ CHUẨN HOÁ QUAN HỆ

6.1 Một số hướng dẫn khi thiết kế cơ sở dữ liệu quan hệ

Việc quan trọng nhất khi thiết kế cơ sở dữ liệu quan hệ là ta phải chọn ra tập các lược đồ quan hệ tốt nhất dựa trên một số tiêu chí nào đó. Và để có được lựa chọn tốt, thì chúng ta cần đặc biệt quan tâm đến mối ràng buộc giữa các dữ liệu trong quan hệ, đó chính là các phụ thuộc hàm.

Để hiểu hơn về câu hỏi tại sao phải thiết kế một cơ sở dữ liệu tốt, chúng ta hãy cùng tìm hiểu ví dụ sau:

RESULT(StNo, StName, SubNo, SubName, Credit, Mark)

Quan hệ RESULT(Kết quả học tập) có các thuộc tính: StNo(Mã sinh viên), StName(Tên sinh viên), SubNo(Mã môn học), SubName(Tên môn học), Credit (Số đơn vị học trình) và Mark (điểm thi của sinh viên trong môn học).

Sau đây là minh hoạ dữ liệu của quan hệ RESULT:

StNo	StName	SubNo	SubName	Credit	Mark
St01	Mai	Sub04	CSDL	3	9
St01	Mai	Sub01	TRR	5	10
St01	Mai	Sub02	PPS	4	8
St02	Vân	Sub04	CSDL	3	10
St02	Vân	Sub01	TRR	5	9
St03	Thanh	Sub07	Tiếng Anh	4	8

Hình 6.1. Minh hoạ dữ liệu của quan hệ RESULT

Quan hệ trên thiết kế chưa tốt vì:

1. **Dư thừa dữ liệu (Redundancy):** Thông tin về sinh viên và môn học bị lặp lại nhiều lần. Nếu sinh viên có mã St01 thì 10 môn học thì thông tin về sinh viên này bị lặp lại 10 lần, tương tự đối với môn học có mã Sub04, nếu có 1000 sinh viên thì thông tin về môn học cũng lặp lại 1000 lần
2. **Không nhất quán (Inconsistency):** Là hệ quả của dư thừa dữ liệu. Giả sử sửa bản ghi thứ nhất, tên sinh viên được chữa thành Nga thì dữ liệu này lại không nhất quán với bản ghi thứ 2 và 3 (vẫn có tên là Mai).
3. **Di thường khi thêm bộ (Insertion anomalies):** Nếu muốn thêm thông tin một sinh viên mới nhập trường (chưa có điểm môn học nào) vào quan hệ thì không được vì khoá chính của quan hệ trên gồm 2 thuộc tính StNo và SubNo.

4. **Di thường khi xoá bộ (Deletion anomalies):** Giả sử xoá đi bản ghi cuối cùng, thì thông tin về môn học có mã môn học là SubNo=Sub07 cũng mất.

Nhận xét: Qua phân tích trên, ta thấy chúng ta nên tìm cách tách quan hệ trên thành các quan hệ nhỏ hơn.

Trong chương này chúng ta sẽ nghiên cứu về những khái niệm và các thuật toán để có thể thiết kế được những lược đồ quan hệ tốt.

6.2 Phụ thuộc hàm (Functional Dependencies)

- Phụ thuộc hàm (FDs) được sử dụng làm thước đo để đánh giá một quan hệ tốt.
- FDs và khoá được sử dụng để định nghĩa các dạng chuẩn của quan hệ.
- FDs là những ràng buộc dữ liệu được suy ra từ ý nghĩa và các mối liên quan giữa các thuộc tính.

6.2.1 Định nghĩa phụ thuộc hàm

Cho $r(U)$, với r là quan hệ và U là tập thuộc tính.

Cho $A, B \subseteq U$, phụ thuộc hàm $X \rightarrow Y$ (đọc là X xác định Y) được định nghĩa là:

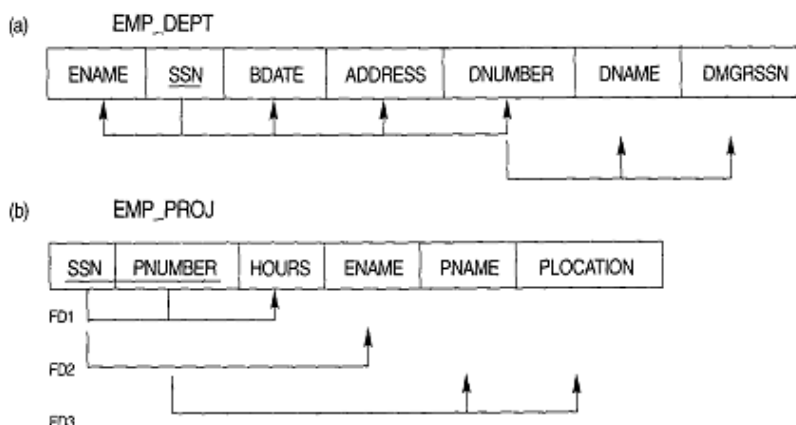
$$\forall t, t' \in r \text{ nếu } t.X = t'.X \text{ thì } t.Y = t'.Y$$

(Có nghĩa là: Nếu hai bộ có cùng trị X thì có cùng trị Y)

Phụ thuộc hàm được suy ra từ những quy tắc dữ liệu khi ta khảo sát yêu cầu của bài toán.

Ví dụ:

Từ mã số bảo hiểm xã hội, ta có thể suy ra được tên của nhân viên ($Ssn \rightarrow Ename$). Từ mã dự án, ta có thể suy ra tên và vị trí của dự án ($PNumber \rightarrow \{PName, PLocation\}$)



Hình 6.2. Biểu diễn FDs của 2 lược đồ quan hệ EMP_DEPT và EMP_PROJ

6.2.2 Hệ tiên đề Armstrong

Cho lược đồ quan hệ $r(U)$, U là tập thuộc tính, F là tập các phụ thuộc hàm được định nghĩa trên quan hệ r .

Ta có phụ thuộc hàm $A \rightarrow B$ được suy diễn logic từ F nếu quan hệ r trên U thỏa các phụ thuộc hàm trong F thì cũng thỏa phụ thuộc hàm $A \rightarrow B$.

Ví dụ:

Tập phụ thuộc hàm: $F = \{ A \rightarrow B, B \rightarrow C \}$

Ta có phụ thuộc hàm $A \rightarrow C$ là phụ thuộc hàm được suy từ F .

Hệ tiên đề Armstrong được sử dụng để tìm ra các phụ thuộc hàm suy diễn từ F .

Hệ tiên đề Armstrong bao gồm:

1. **Phản xạ:** Nếu $Y \subset X$ thì $X \rightarrow Y$
2. **Tăng trưởng:** Nếu $Z \subset U$ và $X \rightarrow Y$ thì $XZ \rightarrow YZ$ (Ký hiệu XZ là $X \cup Z$)
3. **Bắc cầu:** Nếu $X \rightarrow Y$ và $Y \rightarrow Z$ thì $X \rightarrow Z$
4. **Giả bắc cầu:** Nếu $X \rightarrow Y$ và $WY \rightarrow Z$ thì $XW \rightarrow Z$
5. **Luật hợp:** Nếu $X \rightarrow Y$ và $X \rightarrow Z$ thì $X \rightarrow YZ$
6. **Luật phân rã:** Nếu $X \rightarrow Y$ và $Z \subset Y$ thì $X \rightarrow Z$

Trong sáu luật trên thì a_4, a_5, a_6 suy được từ a_1, a_2, a_3 .

6.2.3 Bao đóng của tập phụ thuộc hàm

- Ta gọi f là một phụ thuộc hàm được suy dẫn từ F , ký hiệu là $F \vdash f$ nếu tồn tại một chuỗi phụ thuộc hàm: f_1, f_2, \dots, f_n sao cho $f_n = f$ và mỗi f_i là một thành viên của F hay được suy dẫn từ những phụ thuộc hàm $j=1, \dots, i-1$ trước đó nhờ vào luật dẫn.

- **Bao đóng của F :** ký hiệu là F^+ là tập tất cả các phụ thuộc hàm được suy từ F nhờ vào hệ tiên đề Armstrong. F^+ được định nghĩa:

$$F^+ = \{ X \rightarrow Y \mid F \vdash X \rightarrow Y \}$$

6.2.4 Bao đóng của tập thuộc tính X trên F

Bao đóng của tập thuộc tính X xác định trên tập phụ thuộc hàm F ký hiệu là X^+ là tập hợp tất cả các thuộc tính có thể suy ra từ X . Ký hiệu:

$$X^+ = \{ Y \mid F \vdash X \rightarrow Y \}$$

X^+ có thể được tính toán thông qua việc lặp đi lặp lại các quy tắc 1, 2, 3 của hệ tiên đề Armstrong.

Thuật toán xác định bao đóng của tập thuộc tính X^+ :

$X^+ := X;$
repeat

oldX⁺ := X⁺;
 for (mỗi phụ thuộc hàm Y → Z trong F) do
 if Y ⊆ X⁺ then X⁺ ∪ Z
 until (oldX⁺ = X⁺);

Ví dụ: Cho tập phụ thuộc hàm:

F = { SSN → ENAME, PNUMBER → {PNAME, PLOCATION}, {SSN, PNUMBER} → HOURS }

Suy ra:

{SSN}⁺ = {SSN, ENAME}

{PNUMBER}⁺ = {PNUMBER, PNAME, PLOCATION}

{SSN, PNUMBER}⁺ = {SSN, PNUMBER, ENAME, PNAME, PLOCATION, HOURS}

Như vậy, tập thuộc tính {SSN, PNUMBER} là khoá của quan hệ.

6.2.5 Khoá của quan hệ

Cho quan hệ r(R), tập K ⊂ R được gọi là khoá của quan hệ r nếu: K⁺ = R và nếu bớt một phần tử khỏi K thì bao đóng của nó sẽ khác R.

Như thế tập K ⊂ R là khoá của quan hệ nếu K⁺ = R và (K \ A)⁺ ≠ R, ∀ A ⊂ R.

Ví dụ: Cho R = { A, B, C, D, E, G } và tập phụ thuộc hàm:

F = { AB → C, D → EG, BE → C, BC → D, CG → BD, ACD → B, CE → AG }

Ta sẽ thấy các tập thuộc tính:

K1 = { A, B }, K2 = { B, E }, K3 = { C, G }, K4 = { C, E }, K5 = { C, D }, K6 = { B, C }
 đều là khoá của quan hệ.

Như vậy, một quan hệ có thể có nhiều khoá.

Thuật toán tìm khoá:

Ý tưởng: Bắt đầu từ tập U vì Closure(U⁺, F) = U. Sau đó ta bớt dần các phần tử của U để nhận được tập bé nhất mà bao đóng của nó vẫn bằng U.

Thuật toán:

Input: Lược đồ quan hệ r(U), tập phụ thuộc hàm F.

Output: Khoá K

Bước 1: Gán K = U

Bước 2: Lặp lại các bước sau:

Loại phần tử A khỏi K mà Closure(K - A, F) = U

Nhận xét:

- Thuật toán trên chỉ tìm được một khóa. Nếu cần tìm nhiều khóa, ta thay đổi trật tự loại bỏ các phần tử của K.
- Chúng ta có thể cải thiện tốc độ thực hiện thuật toán trên bằng cách: Trong bước 1 ta chỉ gán $K=Left$ (là tập các phần tử có bên tay trái của các phụ thuộc hàm)

Ví dụ:

Cho lược đồ quan hệ $R = \{ A,B,C,D,E,G,H,I \}$ và tập phụ thuộc hàm:

$F = \{ AC \rightarrow B, BI \rightarrow ACD, ABC \rightarrow D, H \rightarrow I, ACE \rightarrow BCG, CG \rightarrow AE \}$

Tìm khoá K?

Ta có $Left = \{A,B,C,H,E,G\}$

Bước 1: $K=Left = \{A,B,C,H,E,G\}$

Bước 2:

Tập thuộc tính	A	B	C	D	E	G	H	I	Ghi chú
ABCHEG	x	x	x	x	x	x	x	x	
BCHEG	x	x	x	x	x	x	x	x	Loại A
CHEG	x	x	x	x	x	x	x	x	Loại B
CHG	x	x	x	x	x	x	x	x	Loại E

Như vậy, $\{C,H,G\}$ là một khóa của R.

Nếu muốn tìm tất cả các khóa của R, ta cần thay đổi trật tự loại bỏ phần tử của khoá K.

6.2.6 Tập phụ thuộc hàm tương đương

Hai tập phụ thuộc hàm **F** và **G** là tương đương nếu:

- Tất cả các phụ thuộc hàm trong F có thể được suy ra từ G, và
- Tất cả các phụ thuộc hàm trong G có thể suy ra từ F.

Vì thế, F và G là tương đương nếu $F^+ = G^+$

Nếu F và G là tương đương thì ta nói **F phủ G** hay **G phủ F**.

Vì thế, thuật toán sau đây sẽ kiểm tra sự tương đương của hai tập phụ thuộc hàm:

- F phủ E: $\forall X \rightarrow Y \in E$, tính X^+ từ F, sau đó kiểm tra xem $Y \in X^+$
- E phủ F: $\forall X \rightarrow Y \in F$, tính X^+ từ E, sau đó kiểm tra xem $Y \in X^+$

6.2.7 Tập phụ thuộc hàm tối thiểu

Tập phụ thuộc hàm là tối thiểu nếu nó thoả mãn các điều kiện sau:

1. Chỉ có một thuộc tính nằm ở phía bên tay trái của tất cả các phụ thuộc hàm trong F.

DEPARTMENT

DNAME	<u>DNUMBER</u>	DMGRSSN	DLOCATIONS
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

Vi phạm dạng chuẩn 1

Hình 6.3. Dữ liệu của quan hệ DEPARTMENT vi phạm 1NF

- Chuyển quan hệ trên thành dạng chuẩn 1 (bằng cách xác định tập thuộc tính {DNumber, DLocation} là khoá chính), ta có:

DEPARTMENT

DNAME	<u>DNUMBER</u>	DMGRSSN	<u>DLOCATION</u>
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

Dư thừa

Hình 6.4. Dư thừa dữ liệu trong quan hệ ở dạng chuẩn 1

c. Nhận xét:

- Quan hệ ở dạng chuẩn 1 có tồn tại sự dư thừa dữ liệu, trong quan hệ DEPARTMENT, nếu như một phòng có nhiều địa điểm khác nhau thì dữ liệu của 3 thuộc tính (DName, DNumber, DMgrSsn) bị lặp lại nhiều lần.
- Chúng ta có thể tách quan hệ DEPARTMENT thành 2 quan hệ:

DEPARTMENT

DNAME	<u>DNUMBER</u>	DMGRSSN
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

DEPT_LOCATIONS

<u>DNUMBER</u>	<u>DLOCATION</u>
5	Bellaire
5	Sugarland
5	Houston
4	Stafford
1	Houston

Hình 6.5. Quan hệ DEPARTMENT được tách thành 2 quan hệ

Mô tả dữ liệu của 2 quan hệ này:

DEPARTMENT:

DName	<u>DNumber</u>	DMgrSsn
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

DEPT_LOCATIONS:

<u>DNumber</u>	<u>DLocation</u>
5	Bellaire
5	Sugarland
5	Houston
4	Stafford
1	Houston

Hình 6.6. Minh họa dữ liệu của DEPARTMENT và DEPT_LOCATIONS

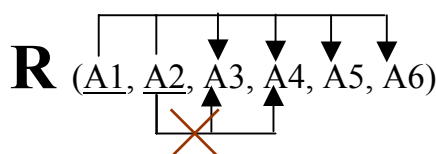
6.3.1.2 Dạng chuẩn 2 (Second Normal Form_2NF)

a. Định nghĩa:

Một quan hệ ở dạng chuẩn 2 nếu:

- Quan hệ đó ở dạng chuẩn 1
- Tất cả các thuộc tính không phải là khóa **phụ thuộc đầy đủ** vào khóa.
- Phụ thuộc đầy đủ: Phụ thuộc hàm $Y \rightarrow Z$ là phụ thuộc hàm đầy đủ nếu: $\forall A \in Y, (Y - \{A\}) \not\rightarrow Z$

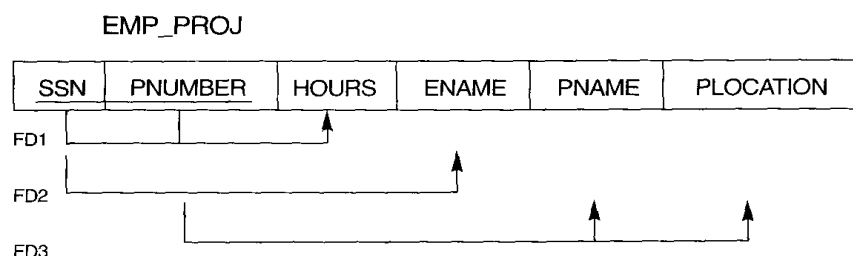
b. Sơ đồ mô tả:



Vi phạm chuẩn 2

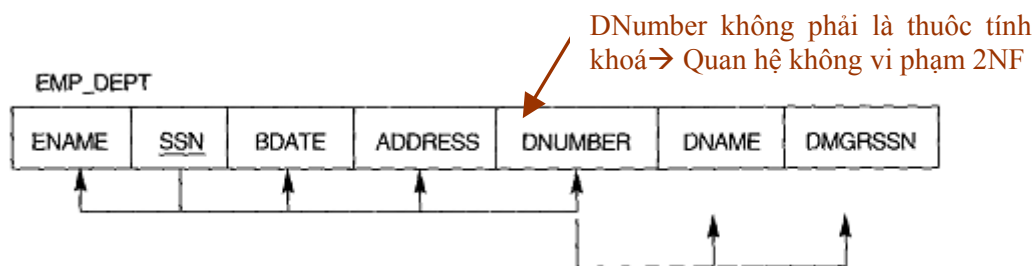
c. Ví dụ:

Ví dụ 1: Quan hệ EMP_PROJ **không phải ở dạng chuẩn 2** vì tồn tại 2 phụ thuộc hàm FD2, FD3 là **phụ thuộc hàm bộ phận** (trái với phụ thuộc hàm đầy đủ)



Hình 6.7. Lược đồ quan hệ EMP_PROJ và các phụ thuộc hàm

Ví dụ 2: Quan hệ sau đây **ở dạng chuẩn 2**:



Hình 6.8. Quan hệ EMP_DEPT ở dạng chuẩn 2

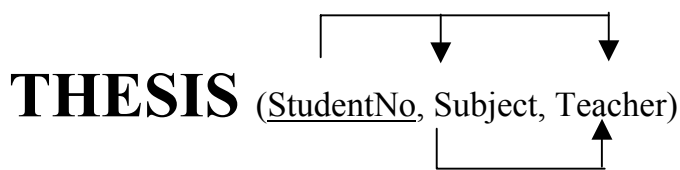
EMP_DEPT

ENAME	<u>SSN</u>	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston, TX	5	Research	333445555
Wong, Franklin T.	333445555	1955-12-08	638 Voss, Houston, TX	5	Research	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle, Spring, TX	4	Administration	987654321
Wallace, Jennifer S.	987654321	1941-06-20	291 Berry, Bellaire, TX	4	Administration	987654321
Narayan, Ramesh K.	666884444	1962-09-15	975 FireOak, Humble, TX	5	Research	333445555
English, Joyce A.	453453453	1972-07-31	5631 Rice, Houston, TX	5	Research	333445555
Jabbar, Ahmad V.	987987987	1969-03-29	980 Dallas, Houston, TX	4	Administration	987654321
Borg, James E.	888665555	1937-11-10	450 Stone, Houston, TX	1	Headquarters	888665555

Dư thừa dữ liệu

Hình 6.9. Minh họa dữ liệu của quan hệ EMP_DEPT

Ví dụ 3: Quan hệ sau đây ở **dạng 2NF**:



<u>StudentNo</u>	Subject	Teacher
SV01	1	Nguyễn Văn Hiệu
SV02	2	Ngô Lan Phương
SV03	1	Nguyễn Văn Hiệu
SV04	1	Nguyễn Văn Hiệu

Hình 6.10. Minh họa dữ liệu của quan hệ THESIS

d. Nhận xét:

- Quan hệ ở dạng chuẩn 2 có sự dư thừa thông tin.

- Dạng chuẩn 2 có thể bị vi phạm khi quan hệ có khóa **gồm hơn một thuộc tính**.

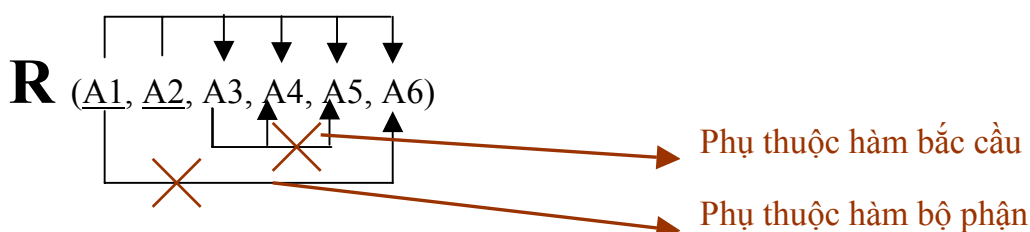
6.3.1.3 Dạng chuẩn 3 (Third Normal Form)

a. Định nghĩa

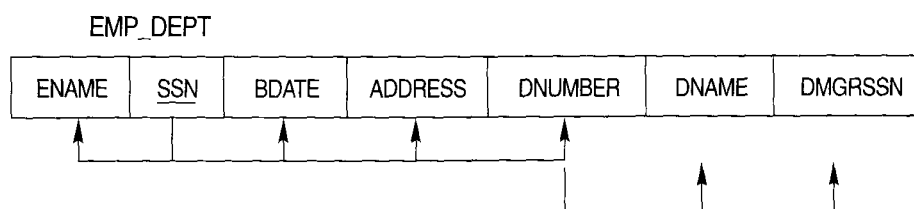
Một quan hệ ở dạng chuẩn 3 nếu:

- Quan hệ ở dạng chuẩn 2
- Và không có chứa các phụ thuộc hàm **phụ thuộc bắc cầu vào khoá**.
- **Phụ thuộc hàm phụ thuộc bắc cầu:** Phụ thuộc hàm $Y \rightarrow Z$ là phụ thuộc hàm bắc cầu nếu tồn tại hai phụ thuộc hàm: $Y \rightarrow X$ và $X \rightarrow Z$.

b. Biểu diễn bằng sơ đồ

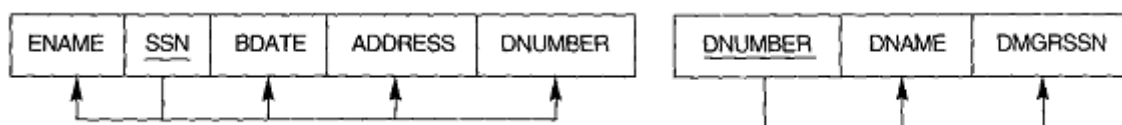


c. Ví dụ: Quan hệ EMP_DEPT **không phải ở dạng chuẩn 3** vì còn tồn tại phụ thuộc hàm $DNumber \rightarrow DName$, $DMgrSsn$ là phụ thuộc hàm phụ thuộc bắc cầu vào khoá.



Hình 6.11. Quan hệ EMP_DEPT không phải ở dạng chuẩn 3

Tách quan hệ trên thành 2 quan hệ: EMPLOYEE và DEPARTMENT. 2 quan hệ sau đều ở dạng chuẩn 3:



Hình 6.12. Tách quan hệ EMP_DEPT thành 2 quan hệ mới

EMPLOYEE

ENAME	SSN	BDATE	ADDRESS	DNUMBER
Smith,John B.	123456789	1965-01-09	731 Fondren,Houston,TX	5
Wong,Franklin T.	333445555	1955-12-08	638 Voss,Houston,TX	5
Zelaya,Alicia J.	999887777	1968-07-19	3321 Castle,Spring,TX	4
Wallace,Jennifer S.	987654321	1941-06-20	291 Berry,Bellaire,TX	4
Narayan,Remesh K.	666884444	1962-09-15	975 Fire Oak,Humble,TX	5
English,Joyce A.	453453453	1972-07-31	5631 Rice,Houston,TX	5
Jabbar,Ahmad V.	987987987	1969-03-29	980 Dallas,Houston,TX	4
Borg,James E.	888665555	1937-11-10	450 Stone,Houston,TX	1

DEPARTMENT

DNAME	DNUMBER	DMGRSSN
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

Hình 6.13. Mô tả dữ liệu của quan hệ EMPLOYEE và DEPARTMENT

d. Nhận xét:

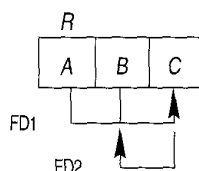
- Trong một cơ sở dữ liệu tốt, các quan hệ nên được chuyển về dạng chuẩn 3.
- Tuy nhiên, dữ liệu vẫn có khả năng dư thừa khi quan hệ có hai tập khóa dự tuyển gối lẫn nhau, hoặc quan hệ có thuộc tính không khóa xác định một thuộc tính khóa .

6.3.1.4 Dạng chuẩn Boyce _Codd(Boyce-Codd Normal Form)

a. Định nghĩa

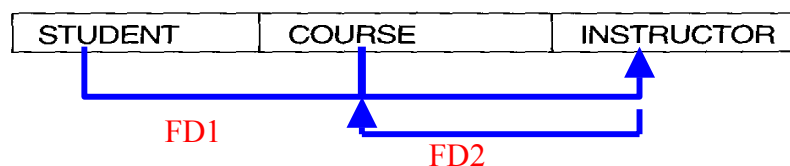
Quan hệ R ở dạng chuẩn BCNF khi tất cả các phụ thuộc hàm $X \rightarrow A$ trong R đều phải có X là khoá của R.

b. Ví dụ: Quan hệ sau ở dạng 3NF nhưng không phải BCNF.



A, B: thuộc tính khoá
C: không phải là thuộc tính khoá

TEACH



STUDENT	COURSE	INSTRUCTOR
Narayan	Database	Mark
Smith	Database	Navathe
Smith	Operating Systems	Ammar
Smith	Theory	Schulman
Wallace	Database	Mark
Wallace	Operating Systems	Ahamad
Wong	Database	Omięcinski
Zelaya	Database	Navathe

Hình 6.14. Minh hoạ dữ liệu của quan hệ TEACH vi phạm chuẩn Boyce -Codd

Để nhận được quan hệ ở BCNF, ta có thể tách quan hệ trên:

Cách 1: R1(Student, Instructor) và R2(Student, Course)

Cách 2: R1(Couse, Instructor) và R2(Course, Student)

Cách 3: R1(Instructor, Course) và R2(Instructor, Student)

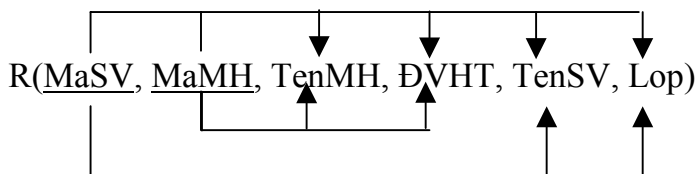
Lưu ý: Việc tách quan hệ như trên sẽ làm mất đi phụ thuộc hàm FD1.

6.3.2 Phép phân rã các lược đồ quan hệ

6.3.2.1 Định nghĩa

Phép phân rã các lược đồ quan hệ $R = \{A_1, A_2, \dots, A_n\}$ là việc thay thế lược đồ quan hệ R thành các lược đồ con $\{R_1, \dots, R_k\}$, trong đó $R_i \subseteq R$ và $R = R_1 \cup R_2 \dots \cup R_k$

Ví dụ: Cho quan hệ R với các phụ thuộc hàm như sau:



Ta có thể phân rã thành 3 lược đồ $R_1(\underline{MaSV}, TenSV, Lop)$ và $R_2(\underline{MaMH}, TenMH, ĐVHT)$ và $R_3(\underline{MaSV}, \underline{MaMH})$.

6.3.2.2 Phép phân rã không mất mát thông tin

Cho R là một lược đồ quan hệ, phép rã $\rho = (R_1, R_2, \dots, R_n)$ và D là tập các phụ thuộc dữ liệu. Phép phân rã ρ không mất mát thông tin nếu khi thực hiện phép toán kết nối tự nhiên các quan hệ thành phần R_1, R_2, \dots, R_n ta vẫn nhận được kết quả của quan hệ ban đầu.

Ví dụ về một phép phân rã có mất mát thông tin:

Cho quan hệ:

MaSV	MaMH	Điểm
1	A	3
2	A	5
3	A	6
4	B	6
5	C	9

Nếu ta phân rã quan hệ trên thành 2 quan hệ: R1(MaSV, MaMH) và R2(MaMH, Điểm) như sau:

R1:

MaSV	MaMH
1	A
2	A
3	A
4	B
5	C

R2:

MaMH	Điểm
A	3
A	5
A	6
B	6
C	9

Thực hiện phép kết nối tự nhiên 2 quan hệ R1 và R2:

$R1 * R2 =$

MaSV	MaMH	Điểm
1	A	3
1	A	5
1	A	6
2	A	3
2	A	5
2	A	6
3	A	3
3	A	5
3	A	6
4	B	6
5	C	9

Như vậy, khi nối tự nhiên 2 bảng, ta nhận được quan hệ không giống quan hệ ban đầu → Phép phân rã trên là mất mát thông tin.

Vấn đề đặt ra đối với người thiết kế là phải tìm ra những phép phân rã không làm mất mát thông tin (chi tiết sẽ được trình bày ở phần sau). Bây giờ chúng ta sẽ tìm hiểu một thuật toán để kiểm tra một phép phân rã có mất mát thông tin hay không.

6.3.2.3 Thuật toán kiểm tra phép phân rã không mất mát thông tin

Input:

- Lược đồ quan hệ $R = \{A_1, A_2, \dots, A_n\}$
- Tập các phụ thuộc hàm F
- Phép tách $\rho(R_1, R_2, \dots, R_k)$

Output: Kết luận phép tách ρ không mất mát thông tin.

Các bước của thuật toán:

Bước 1:

- Thiết lập một bảng với n cột (tương ứng với n thuộc tính) và k dòng (tương ứng với k quan hệ), trong đó cột thứ j ứng với thuộc tính A_j , dòng thứ i ứng với lược đồ R_i .
- Tại dòng i và cột j, ta điền ký hiệu a_j nếu thuộc tính $A_j \in R_i$. Ngược lại ta điền ký hiệu b_{ij} .

Bước 2:

- Xét các phụ thuộc hàm trong F và áp dụng cho bảng trên.
- Giả sử ta có phụ thuộc hàm $X \rightarrow Y \in F$, xét các dòng có giá trị bằng nhau trên thuộc tính X **thì làm bằng** các giá trị của chúng trên Y. Ngược lại làm bằng chúng bằng ký hiệu b_{ij} . Tiếp tục áp dụng các pth cho bảng (kể cả việc lặp lại các phụ thuộc hàm đã áp dụng) cho tới khi không còn áp dụng được nữa.

Bước 3:

Xem xét bảng kết quả. Nếu xuất hiện một dòng chứa toàn giá trị **a1, a2 ,...,an** thì kết luận phép tách ρ không mất mát thông tin.

Vi dụ: Cho quan hệ:

EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston, TX	5	Research	333445555
Wong, Franklin T.	333445555	1955-12-08	638 Voss, Houston, TX	5	Research	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle, Spring, TX	4	Administration	987654321
Wallace, Jennifer S.	987654321	1941-06-20	291 Berry, Bellaire, TX	4	Administration	987654321
Narayan, Ramesh K.	666884444	1962-09-15	975 FireOak, Humble, TX	5	Research	333445555
English, Joyce A.	453453453	1972-07-31	5631 Rice, Houston, TX	5	Research	333445555
Jabbar, Ahmad V.	987987987	1969-03-29	980 Dallas, Houston, TX	4	Administration	987654321
Borg, James E.	888665555	1937-11-10	450 Stone, Houston, TX	1	Headquarters	888665555

Hình 6.15. Minh họa dữ liệu của quan hệ EMP_DEPT

Tách quan hệ trên thành 2 quan hệ:

EMPLOYEE

ENAME	SSN	BDATE	ADDRESS	DNUMBER
Smith,John B.	123456789	1965-01-09	731 Fondren,Houston,TX	5
Wong,Franklin T.	333445555	1955-12-08	638 Voss,Houston,TX	5
Zelaya,Alicia J.	999887777	1968-07-19	3321 Castle,Spring,TX	4
Wallace,Jennifer S.	987654321	1941-06-20	291 Berry,Bellaire,TX	4
Narayan,Remesh K.	666884444	1962-09-15	975 Fire Oak,Humble,TX	5
English,Joyce A.	453453453	1972-07-31	5631 Rice,Houston,TX	5
Jabbar,Ahmad V.	987987987	1969-03-29	980 Dallas,Houston,TX	4
Borg,James E.	888665555	1937-11-10	450 Stone,Houston,TX	1

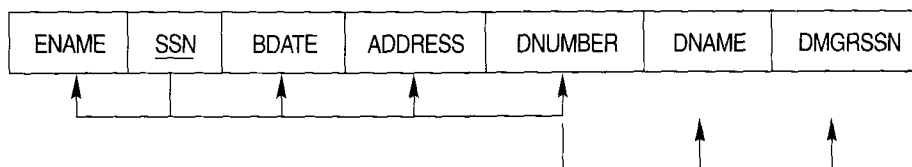
DEPARTMENT

DNAME	DNUMBER	DMGRSSN
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

Hình 6.16. Quan hệ EMPLOYEE được phân rã (tách) thành 2 quan hệ

Tập phụ thuộc hàm F:

EMP_DEPT



Kiểm tra phép tách trên là không mất mát thông tin:

Bước 1:

	ENAME	SSN	BDate	Address	DNumber	DName	DMgrSsn
EMPLOYEE	a1	a2	a3	a4	a5	b16	b17
DEPARTMENT	b21	b22	b23	b24	a5	a6	a7

Bước 2: Xét phụ thuộc hàm DNumber → DName, DMgrSsn. Ta nhận thấy có giá trị a5 ở dòng thứ 2, nên ta sẽ làm bằng giá trị a6, a7 cho dòng thứ 1.

Bước 3: Tồn tại một dòng chứa giá trị a1, a2,..a7. Kết luận, phép phân rã trên không mất mát thông tin.

	EName	SSN	BDate	Address	DNumber	DName	DMgrSsn
EMPLOYEE	a1	a2	a3	a4	a5	a6	a7
DEPARTMENT	b21	b22	b23	b24	a5	a6	a7

Ghi chú: Sinh viên thực hiện phép nối tự nhiên 2 quan hệ EMPLOYEE và DEPARTMENT trên để kiểm tra có bằng quan hệ ban đầu EMP_DEPT

6.4 Chuẩn hoá quan hệ

Chuẩn hoá quan hệ là việc phân rã một lược đồ quan hệ thành các lược đồ con ở dạng chuẩn 3 hoặc ở BCNF sao cho vẫn bảo toàn phụ thuộc và không mất mát dữ liệu.

6.4.1 Thuật toán phân rã lược đồ quan hệ thành các lược đồ quan hệ con ở BCNF

Input:

- Lược đồ quan hệ R
- Tập phụ thuộc hàm F

Output:

Phép phân rã của R không mất thông tin và mỗi lược đồ quan hệ trong phép tách đều ở dạng BCNF đối với phép chiếu của F trên lược đồ đó.

Các bước của thuật toán:

- Ban đầu phép tách ρ chỉ bao gồm R.
- Nếu S là một lược đồ thuộc ρ và S chưa ở dạng BCNF thì chọn phụ thuộc hàm $X \rightarrow A$ thỏa trong S, trong đó X không chứa khóa của S và $A \notin X$. {phụ thuộc hàm vi phạm định nghĩa dạng chuẩn BCNF}.
Thay thế S trong ρ bởi S1 và S2 như sau $S1 = XA$, $S2 = S \setminus A$.
- Quá trình trên tiếp tục cho đến khi tất cả các lược đồ quan hệ đều ở dạng BCNF

Ví dụ:

Cho lược đồ quan hệ R(CTHRSG).

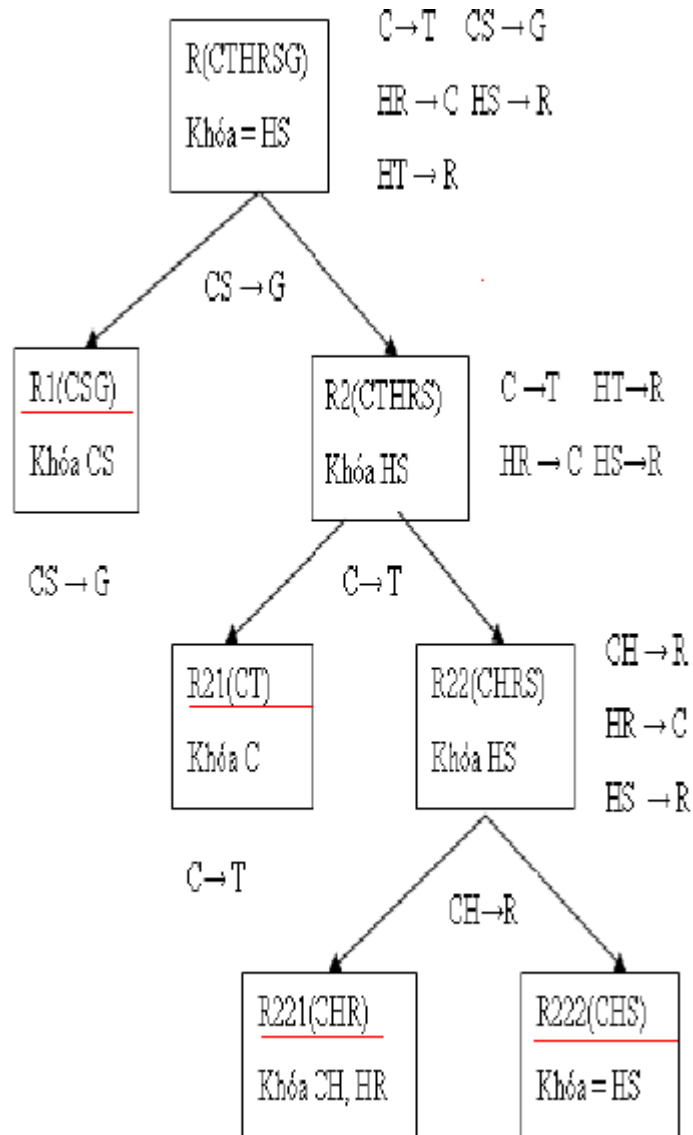
Trong đó:

- C: Course; T: Teacher; H: Hour; R: Room; S: Student; G: Group).
- Và tập các phụ thuộc hàm F:
 - o $C \rightarrow T$: Mỗi khoá học (course) có một thầy (teacher) duy nhất.
 - o $HR \rightarrow C$: Tại một thời điểm (Hour) ở tại phòng học (room) chỉ có một khoá học duy nhất.

- $HT \rightarrow R$: Tại một thời điểm và một giáo viên chỉ ở một phòng duy nhất
- $CS \rightarrow G$: Một sinh viên học một course thì chỉ ở một lớp duy nhất.
- $HS \rightarrow R$: Một sinh viên, ở một thời điểm nhất định chỉ ở trong một phòng duy nhất.

Dựa vào thuật toán tìm khoá \rightarrow Khóá của R là **HS**.

Yêu cầu: Tách lược đồ R thành các lược đồ con ở dạng BCNF.



Hình 6.17. Biểu diễn quá trình tách quan hệ R thành các quan hệ ở BCNF

Như vậy, quan hệ R được tách thành 4 quan hệ R1, R21, R221, R222 đều ở BCNF.

6.4.2 Thuật toán phân rã một lược đồ quan hệ thành các lược đồ con ở 3NF.

Input:

- Lược đồ quan hệ R
- Tập các phụ thuộc hàm F, không làm mất tính tổng quát giả sử đó là phủ tối thiểu.

Output:

Phép tách không mất mát thông tin trên R thành các lược đồ con ở dạng chuẩn 3 sao cho vẫn bảo toàn các phụ thuộc hàm.

Các bước của thuật toán:

- Bước 1: Loại bỏ các thuộc tính của R nếu thuộc tính đó không liên quan đến phụ thuộc hàm nào của F. (không có mặt ở cả hai vế của phụ thuộc hàm).
- Bước 2: Nếu có một phụ thuộc hàm của F liên quan đến tất cả các thuộc tính của R thì kết quả chính là R.
- Bước 3: Ngoài ra, phép tách ρ đưa ra các lược đồ gồm các thuộc tính XA ứng với phụ thuộc hàm $X \rightarrow A \in F$. Nếu tồn tại các phụ thuộc hàm $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n$ thuộc F thì thay thế $XA_i (1 \leq i \leq n)$ bằng $XA_1A_2 \dots A_n$. Quá trình tiếp tục.
- Chú ý: Tại mỗi bước kiểm tra lược đồ R, **nếu mỗi thuộc tính không khóa không phụ thuộc bắc cầu vào khóa chính**, thì R đã ở dạng 3NF, ngược lại cần áp dụng bước 3 để tách tiếp.

Ví dụ:

Cho lược đồ quan hệ R(C,T,H,R,S,G) với tập phụ thuộc hàm tối thiểu F:

$C \rightarrow T, HR \rightarrow C, HT \rightarrow R, CS \rightarrow G, HS \rightarrow R$.

Yêu cầu: Phân rã lược đồ quan hệ trên thành các quan hệ con đều ở dạng 3NF.

- Sử dụng thuật toán tìm khoá \rightarrow Khoá chính của R là HS.
- Thực hiện thuật toán:
 - o Bước 1: Không có thuộc tính bị loại bỏ
 - o Bước 2: Không có phụ thuộc hàm nào liên quan tới tất cả các thuộc tính
 - o Bước 3:
 - Phụ thuộc hàm $C \rightarrow T$ vi phạm 3NF (phụ thuộc bắc cầu vào khoá), vì vậy tách R thành $R_1(\underline{C}, T)$ và $R_2(C, \underline{H}, R, \underline{S}, G)$.
 - Phụ thuộc hàm $CS \rightarrow G$ vi phạm 3NF (phụ thuộc bộ phận vào khoá), tách R_2 thành $R_{21}(\underline{C}, \underline{S}, G)$ và $R_{22}(C, \underline{H}, R, \underline{S})$.

- Phụ thuộc hàm $HR \rightarrow C$ vi phạm 3NF, tách R22 thành $R221(\underline{H}, \underline{R}, C)$ và $R222(\underline{H}, \underline{S}, R)$

Như vậy, quan hệ R được tách thành các quan hệ sau: R1, R21, R221, R222

Lưu ý:

- Kết quả của phép tách có thể khác nhau phụ thuộc vào thứ tự áp dụng các phụ thuộc hàm khi thực hiện thuật toán.
- Sinh viên tự kiểm tra xem việc tách quan hệ như trên có mất mát thông tin không.

Bài tập:

1. Cho một quan hệ $R = \{A, B, C, D, E, F, G, H, I, J\}$ và tập phụ thuộc hàm

$F = \{ A, B \rightarrow C$
 $A \rightarrow D, E$
 $B \rightarrow F$
 $F \rightarrow G, H$
 $D \rightarrow I, J$
 $\}$

Yêu cầu:

- Tìm $\{A\}^+ = \{D, E, I, J\}$
- Tìm khóa của quan hệ R.
- Tách quan hệ R thành BCNF.
- Kiểm tra xem việc tách trên có mất mát thông tin không?

2. Lặp lại yêu cầu ở bài 1 với tập phụ thuộc hàm sau:

$G = \{ A, B \rightarrow C$
 $B, D \rightarrow E, F$
 $A, D \rightarrow G, H$
 $A \rightarrow I$
 $H \rightarrow J \}$

3. Cho một quan hệ $R = \{CourseNo, SecNo, OfferingDept, Credit_Hours, CourseLevel, InstructorSSN, Semester, Year, Days_Hours, RoomNo, NoOfStudents\}$ và tập phụ thuộc hàm:

$F = \{ CourseNo \rightarrow OfferingDept, Credit_Hours, CourseLevel;$
 $CourseNo, SecNo, Semester, Year \rightarrow Days_Hours, RoomNo,$
 $NoOfStudents, InstructorSSN;$
 $RoomNo, Days_Hours, Semester, Year \rightarrow InstructorSSN, CourseNo, SecNo \}$

Yêu cầu:

- Tìm khóa của quan hệ R.
- Quan hệ trên thuộc dạng chuẩn mấy?
- Tách quan hệ về dạng 3NF.
- Kiểm tra xem việc tách trên có mất mát thông tin không?

7 Chương 7. THIẾT KẾ CƠ SỞ DỮ LIỆU VẬT LÝ (Tham khảo)

Thiết kế cơ sở dữ liệu vật lý là quá trình chuyển các đặc tả dữ liệu logic thành các đặc tả kỹ thuật để lưu trữ dữ liệu. Gồm 2 nội dung sau:

Lựa chọn công nghệ lưu trữ (Hệ điều hành, HQTCSDL, các công cụ truy nhập dữ liệu).

Chuyển các quan hệ của mô hình logic thành các thiết kế vật lý.

Trong chương này sẽ trình bày những phần sau:

Thiết kế các trường, bản ghi vật lý

Thiết kế file vật lý

Thiết kế cơ sở dữ liệu vật lý

7.1 Nội dung thiết kế file vật lý và cơ sở dữ liệu vật lý

7.1.1 Quá trình thiết kế

Trong quá trình thiết kế hệ thống vật lý, vấn đề đặt ra hàng đầu là phải làm thế nào tối thiểu hóa không gian lưu trữ và thời gian người dùng tương tác với hệ thống.

Tuy nhiên, do dung lượng các thiết bị nhớ tăng nhanh, nên người ta tập trung nhiều vào việc xử lý các file và dữ liệu sao cho hiệu quả hơn đối với người sử dụng.

Các thông tin cần thiết để thiết kế file vật lý:

Các quan hệ đã được chuẩn hóa, kể cả ước lượng về số lượng dữ liệu cần lưu trữ

Định nghĩa chi tiết các thuộc tính

Các mô tả cho biết ở đâu và khi nào dữ liệu được sử dụng (xem, thêm, sửa, xóa).

Các yêu cầu và mong đợi về sử dụng dữ liệu và tích hợp dữ liệu, bao gồm các yêu cầu về thời gian đáp ứng, các mức độ an toàn, ghi tạm, phục hồi....

Các mô tả về công nghệ được sử dụng để triển khai file và CSDL (thiết bị lưu trữ, hệ điều hành, HQTCSDL...)

Một số các quyết định cơ bản có ý nghĩa đối với sự tích hợp và hiệu năng của hệ thống ứng dụng cần thực hiện:

Chọn định dạng lưu trữ (kiểu dữ liệu) cho mỗi thuộc tính sao cho tối thiểu hóa dư thừa thông tin và tối đa sự tích hợp dữ liệu.

Nhóm gộp các thuộc tính từ mô hình dữ liệu logic vào bản ghi vật lý.

Sắp xếp các bản ghi có quan hệ với nhau vào bộ nhớ ngoài sao cho từng bản ghi hay nhóm các bản ghi lưu trữ, cập nhật và lấy ra nhanh chóng (gọi là tổ chức file)

Lựa chọn phương tiện và cấu trúc để lưu trữ dữ liệu đảm bảo truy nhập hiệu quả hơn.

7.1.2 Sản phẩm thiết kế

Sản phẩm thiết kế là một tập các đặc tả mà các nhà lập trình và các nhà phân tích dữ liệu sẽ sử dụng để xác định định dạng và cấu trúc các file trong bộ nhớ thứ cấp của máy tính (bộ nhớ ngoài).

Khi sử dụng các công cụ CASE, kho dữ liệu của CASE hay từ điển dữ liệu dự án là nơi lưu trữ tất cả các đặc tả nêu ra ở trên.

Sau đây là các phần tử tiêu biểu của thiết kế được lưu trữ trong kho dữ liệu của CASE khi thiết kế file và cơ sở dữ liệu vật lý:

BẢNG MÔ TẢ CÁC TRƯỜNG

Loại đặc tả	Mô tả nội dung
Tên trường (field name)	Theo quy định về cách đặt tên trường của HQTCSDL.
Kiểu trường (data type)	Chọn kiểu dữ liệu mà HQTCSDL đó hỗ trợ
Kích cỡ (size)	Là kích thước tối đa dùng để lưu trữ dữ liệu của trường đó
Mã hóa (Coding)	Cách viết tắt giá trị của trường. Ví dụ, mỗi nước được biểu diễn bằng hai ký tự
Các quy tắc toàn vẹn dữ liệu (data integrity rules)	Các đặc tả về các hạn chế đặt lên giá trị của trường
Các kiểm soát bảo trì (maintenance controls)	Chỉ ra những giá trị nào được phép thay đổi
Công thức (Formular)	Mô tả công thức tính toán giá trị với những trường số cần tính toán.
Toàn vẹn tham chiếu (references integrity)	Đặc tả giá trị của trường có liên quan đến giá trị của trường khác
Sở hữu (Ownership)	Ai là người sở hữu trường đó (có quyền đối với dữ liệu)

BẢNG CÁC ĐẶC TẢ TIÊU BIỂU ĐỐI VỚI THIẾT KẾ BẢN GHI

Các trường (fields)	Danh sách các trường trong một bản ghi
Dữ liệu có cấu trúc (Structure Data)	Định nghĩa cấu trúc dữ liệu dùng để lưu trữ bản ghi (Thứ tự các trường, khóa chính, khóa ngoại...)
Sự lưu trữ lại (retention)	Đặc tả những bản ghi nào đó được giữ lại trong file bao lâu (dữ liệu về sinh viên không được lưu trữ quá 10 năm sau khi ra trường).

BẢNG CÁC ĐẶC TẢ TIÊU BIỂU ĐỐI VỚI THIẾT KẾ FILE

Tên file và định vị	Tên file theo quy định của HQTCSDL và thiết bị lưu trữ nó.
Các bản ghi (record)	Những bản ghi nào được lưu trữ trong file.
Khóa chính (Primary Key)	Là một hay một số trường được dùng để định danh duy nhất cho bản ghi.
Chỉ số hóa (index)	Chỉ ra các trường được dùng để lập chỉ số
Yếu tố khóa bản ghi (Record blocking factor)	Số các bản ghi theo mỗi trang hoặc khóa của bản ghi (Ví dụ: 10 bản ghi của ITEM được lưu trữ trong một trang bộ nhớ ngoài)
Lưu giữ lại và sao lưu (Retention and Backup)	File được lưu trữ trong bao lâu và các thủ tục sao lưu, thời gian định kỳ cần sao lưu (để đảm bảo an toàn khi có sự cố).
Tổ chức file (file organization)	Phương pháp truy nhập dữ liệu và sắp xếp các bản ghi trong file
Kiểm soát (controls)	Đặc tả về kiểm soát và phương pháp mã hóa

BẢNG CÁC ĐẶC TẢ TIÊU BIỂU ĐỐI VỚI THIẾT KẾ CSDL

Các file	Các file trong CSDL và nơi định vị nó
Kiến trúc (Architecture)	Loại hình cấu trúc (bao gồm cả mô hình) CSDL được dùng để tổ chức file.
Các mối quan hệ	Cơ chế để liên kết file với nhau.

7.2 Thiết kế các trường

Một thuộc tính trong mô hình dữ liệu logic được biểu diễn bằng một số trường (fields).

Ví dụ: HoTenSV được biểu diễn thành 2 trường HodemSV và TenSV

7.2.1 Yêu cầu thiết kế trường

Mỗi HQTCSDL sử dụng những kiểu dữ liệu nhất định để lưu trữ dữ liệu.

Trong yêu cầu thiết kế trường, quan trọng nhất là phải chọn kiểu dữ liệu phù hợp, ta thường quan tâm đến các mục tiêu sau khi chọn kiểu dữ liệu:

Tiết kiệm không gian lưu trữ

Biểu diễn được mọi giá trị có thể thuộc miền giá trị

Cải thiện tính toàn vẹn (tổ chức việc nhập dữ liệu, kiểm tra dữ liệu đầu vào)

Hỗ trợ thao tác dữ liệu (Ví dụ: thao tác với dữ liệu số nhanh hơn với ký tự)

7.2.2 Chon kiểu và cách biểu diễn dữ liệu

7.2.2.1 Kiểu dữ liệu

Các kiểu dữ liệu mà HQTCSDDL SQL hỗ trợ và ý nghĩa của nó

DECIMAL(m,n)	Số thập phân có độ dài là m chữ số và n số thập phân
INTEGER	Số nguyên lớn (độ dài tối đa là 11 chữ số)
SMALLINT	Số nguyên nhỏ (độ dài tối đa là 6 chữ số)
FLOAT(m,n)	Số thực có độ dài là m chữ số và n số thập phân
CHAR	Xâu ký tự có độ dài là m ký tự
DATE	Kiểu dữ liệu thời gian và có rất nhiều cách biểu diễn
LOGICAL	Giá trị logic (đúng/sai)

7.2.2.2 Các trường tính toán

Khi giá trị của một trường là giá trị nhận được từ các giá trị của trường khác thì trường đó gọi là trường tính toán.

Có các loại tính toán sau:

+ Tính toán số học: Lương= Hệ số lương * 210.

+ Tính toán logic: Tiền trợ cấp = $\begin{cases} 50.000 \text{ đ nếu cán bộ là nữ.} \\ 0 \text{ nếu cán bộ là nam.} \end{cases}$

+ Tính toán hỗn hợp:

Tiền điện= Số điện * 500đ nếu số điện < 100.

Số điện * 500 + (Số điện - 100)* 750 nếu số điện > 100.

7.2.2.3 Các kỹ thuật mã hóa dữ liệu và nén dữ liệu

Một số phương pháp mã hóa dùng để biểu diễn dữ liệu trong các trường lưu trữ:

Mã hóa phân cấp: để mô tả các dữ liệu phân cấp người ta dùng nhiều nhóm, mỗi nhóm đại diện cho cấp và các nhóm được sắp xếp lần lượt từ trái sang phải.

Ví dụ: Hệ thống phân loại sách trong thư viện:

Các cấp			Mã số	Tên tài khoản
1	2	3		
500			500	Khoa học tự nhiên
	1		5001	Toán học
	2		5002	Vật lý
	1	1	50011	Toán giải tích
	1	2	50012	Toán rời rạc

Mã liên tiếp: Mã này được tạo ra theo quy tắc một dãy liên tục, như 1, 2, 3 ... A, B, C.... Mã loại này dùng cho những dữ liệu là danh sách như danh sách sinh viên. Nó đơn giản, dễ tự động hóa, không nhầm lẫn. Tuy nhiên nó không gợi nhớ về đối tượng được mã hóa và không cho phép chèn thêm vào giữa.

Mã gợi nhớ: Căn cứ vào đối tượng được mã hóa để cấu tạo mã. Ví dụ: VND (Đồng Việt Nam), TL001 (Thủy lợi 001)...Loại này giúp ta nhận ra đối tượng được mã hóa, có thể nói rộng hoặc thu hẹp số lượng mã. Tuy nhiên khó tổng hợp và phân tích.

Mã thành phần ngữ nghĩa: Theo phương pháp này, mã được chia làm nhiều thành phần, mỗi phần mô tả một đặc trưng nhất định của đối tượng như phân loại, địa danh... Những phần này có thể sử dụng các nhóm ký tự khác nhau. Mã loại này rất thông dụng và được sử dụng nhiều trong công nghiệp cũng như giao tiếp quốc tế.

Ví dụ: Địa chỉ miền trên internet có dạng:

<Tên tổ chức>.<Loại tổ chức>.<Tên nước>

Ví dụ : **hwru.edu.vn**: Đại học Thủy Lợi, Tổ chức giáo dục, Tên nước

Mã loại này công kênh, và cần chọn các thành phần sao cho ổn định, nếu không việc sử dụng mã sẽ gặp nhiều khó khăn.

7.2.2.4 Kiểm tra tính toàn vẹn của dữ liệu

Để đảm bảo tính đúng đắn của dữ liệu người ta đặt các ràng buộc trên các dữ liệu đó.

Thường dùng các phương pháp sau để kiểm tra tính toàn vẹn:

Giá trị ngầm định (default value): Là giá trị được gán sẵn cho một trường nào đó khi bản ghi mới được nhập vào. Ví dụ: Trong hóa đơn bán hàng, trường ngày bán được mặc định là ngày hiện tại.

Kiểm tra khuôn dạng (picture control): Là mẫu định dạng bao gồm độ rộng, các giá trị có thể trong từng vị trí. Ví dụ: TLA006, \$999,999.99.

Kiểm tra giới hạn (range control): Các trường có thể đưa ra các giới hạn đối với các giá trị của nó. Ví dụ: Điểm môn học được giới hạn là các số và được giới hạn từ 0..10.

Tính toàn vẹn tham chiếu (reference integrity): là giá trị của thuộc tính đã cho có thể bị hạn chế bởi giá trị của những thuộc tính khác. Ví dụ: Trong mỗi quan hệ 1_N, nếu giá trị của bên 1 chưa có thì sẽ không được có bên N.

Kiểm tra giá trị rỗng (Null value control): Nếu đặt một thuộc tính nào đó là khác rỗng thì bắt buộc ta phải thêm giá trị cho trường đó.

Quản lý dữ liệu mất: Trong khi vận hành, nếu vì một lý do nào đó mà dữ liệu có thể bị mất. Khi thiết kế file vật lý, các nhà thiết kế phải chỉ ra cách thức mà hệ thống quản lý dữ liệu bị mất. Balad và Hofer đã đưa ra một số phương pháp sau đây dùng để quản lý dữ liệu của 1 trường bị mất:

Cho quy trình để ước lượng giá trị bị mất.

Theo dõi dữ liệu bị mất để báo cáo và sử dụng một phần tử hệ thống giúp con người mau chóng thay thế giá trị bị mất này.

Thực hiện một số kiểm tra để có thể bỏ qua dữ liệu bị mất hay phải phục hồi nó nếu nó thực sự ảnh hưởng đến kết quả của hệ thống.

7.3 Thiết kế các bản ghi vật lý

Một bản ghi vật lý là một nhóm các trường được lưu trữ ở các vị trí liên kề nhau và được gọi ra cùng nhau như một đơn vị thống nhất.

Thiết kế bản ghi vật lý là chọn một nhóm các trường của nó sẽ lưu trữ ở những vị trí liên kề nhau nhằm 2 mục tiêu: sử dụng hiệu quả không gian lưu trữ và tăng tốc độ truy nhập. Hệ điều hành đọc hay ghi dữ liệu vào bộ nhớ thứ cấp theo một đơn vị gọi là trang. Một trang này có dung lượng cụ thể phụ thuộc vào hệ điều hành và máy tính cụ thể.

Vấn đề đặt ra ở đây là phải thiết kế các bản ghi thế nào để tận dụng được dung lượng chứa của trang. Nếu dung lượng của trang tận dụng được càng nhiều thì số lần đọc càng ít và tốc độ truy cập càng nhanh.

Để làm được điều này người ta thường phi chuẩn hóa một số quan hệ nhận được.

7.3.1 Phi chuẩn

Việc phi chuẩn hóa các quan hệ đã chuẩn hóa trong nhiều trường hợp là cần thiết để tận dụng dung lượng trang của máy.

BENHNHAN(MaBN, TenBN, Diachi_BN, Ngay_nhap, Giuong_phong, Khoa, Tinh_trang, Ngayra, ThanhToan)

Ta có thể phân chia nó thành 2 quan hệ mới để có độ dài gần với dung lượng trang:

BENHNH1(MaBN, TenBN, Diachi_BN, Khoa)

BENHNH2(MaBN, Ngay_nhap, Giuong_phong, Tinh_trang, Ngayra, ThanhToan)

Có một số dạng phi chuẩn hóa, nhưng không có một quy tắc chặt chẽ nào. Rodger đã thảo luận đến một số trường hợp chung có thể xét phi chuẩn:

Hai thực thể có quan hệ một – một.

Ví dụ: Có 2 quan hệ có mối liên kết 1_1 như sau:

SINHVIEN(MaSV, TenSV, MaThe)

THEDOC(MaThe, DiaChi, NgayCap, MaSV)

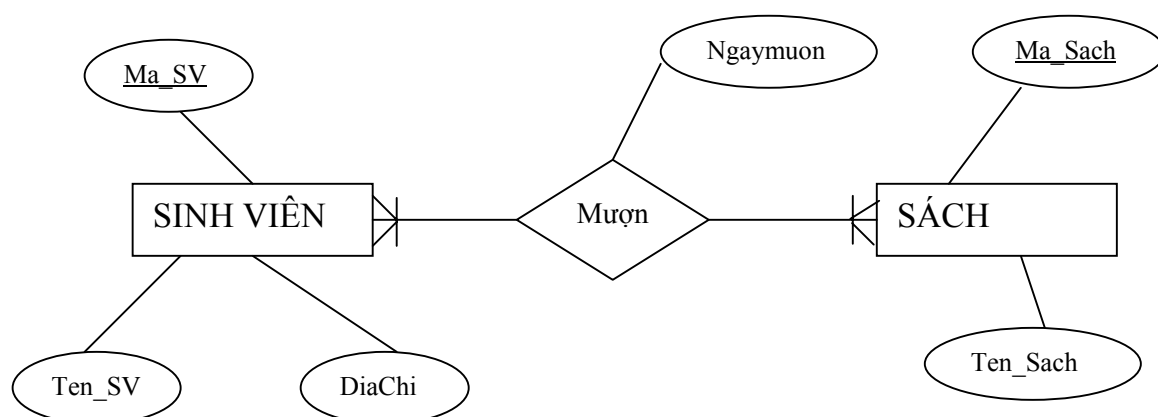
Phi chuẩn hóa ta có quan hệ sau:

SINHVIEN(MaSV, TenSV, MaThe, DiaChi, NgayCap)

Và trong trường hợp này MaThe, DiaChi, NgayCap có thể bỏ trống đối với những SV không có thẻ.

Hai thực thể có mối quan hệ M_N trong đó liên kết có thuộc tính riêng.

Ví dụ: Có quan hệ như sau:



Sau khi chuẩn hóa, ta nhận được 3 quan hệ sau:

SINHVIÊN(Ma_SV, Ten_SV, DiaChi)

SÁCH(Ma_Sach, Ten_Sach)

MƯỢN(Ma_SV, Ma_Sach, Ngay_muon)

Phi chuẩn hóa ta được:

SINHVIÊN(Ma_SV, Ten_SV, DiaChi)

MƯỢN(Ma_SV, Ma_Sach, TenSach, NgayMuon)

Dữ liệu tham chiếu: Trong quan hệ 1_N nếu bảng ở bên 1 không tham gia vào một quan hệ nào khác thì ta có thể hợp nhất 2 thực thể này thành 1.

Ví dụ:

KHO (Ma_Kho, Ten_Kho, Loai_Kho)

HANG(Ma_Hang, Ten_Hang)

Phi chuẩn hóa ta được:

HANG(Ma_Hang, Ten_Hang, Ma_Kho, Ten_Kho, Loai_Kho)

7.3.2 Quản lý trường có độ dài cố định

Việc thiết kế bản ghi sẽ rất dễ dàng nếu trường có độ dài cố định (vì tính ngay được độ dài bản ghi).

Trong trường hợp này việc xác định vị trí của một trường chỉ bằng phép toán: Vị trí con trở hiện thời + (độ dài bản ghi * số bản ghi)

7.3.3 Quản lý trường có độ dài biến đổi

Một trường có độ dài thay đổi như trường Memo thì định vị trí của một trường hay một bản ghi cụ thể không đơn giản.

Một cách chung để quản lý trường có độ dài thay đổi là đưa các bản ghi có độ dài cố định vào một bản ghi vật lý có độ dài cố định và đưa những bản ghi vật lý có độ dài thay đổi vào một bản ghi vật lý có độ dài thay đổi. Đó chính là kỹ thuật thiết kế

bản ghi vật lý tự động được sử dụng trong hầu hết các hệ quản trị CSDL cho máy tính nhỏ.

7.4 Thiết kế file vật lý

File vật lý là một phần nhỏ của bộ nhớ thứ cấp (đĩa cứng, băng từ...) lưu các bản ghi vật lý một cách độc lập. Việc lưu trữ các bản ghi vật lý ở vị trí nào đối với người dùng không quan trọng nhưng lại được các nhà thiết kế đặc biệt quan tâm.

7.4.1 Các loại file

Một hệ thống thông tin có thể cần đến 6 loại file sau:

File dữ liệu (Data file- master file): là file chứa dữ liệu liên quan với mô hình dữ liệu vật lý và logic. File này luôn tồn tại, nhưng nội dung thay đổi.

File lấy từ bảng (look up table file): Là danh sách các dữ liệu tham chiếu lấy từ một hay một số file khác theo một yêu cầu nào đó.

File giao dịch (Transaction file): là file dữ liệu tạm thời phục vụ các hoạt động hàng ngày của một tổ chức. File này thường được thiết kế để phục vụ các yêu cầu xử lý nhanh.

File làm việc (Work file): Là file tạm thời dùng để lưu kết quả trung gian, file này sẽ tự động xóa đi mỗi khi không cần thiết.

File bảo vệ (Protection file): là file được thiết kế để khắc phục những sai sót trong quá trình hệ thống hoạt động. Các file này cho hình ảnh của file dữ liệu trước và sau những hoạt động nhất định (cập nhật, sửa đổi, xử lý...) của hệ thống.

File lịch sử (History file): File này ghi lại quá trình hoạt động của hệ thống, cũng có thể là các dữ liệu cũ hiện không cần sử dụng.

Việc tổ chức các loại file khác nhau không chỉ liên quan đến việc tổ chức lưu trữ và khai thác dữ liệu, mà còn liên quan đến các hoạt động xử lý dữ liệu trong quá trình hoạt động của hệ thống. Về nguyên tắc, việc sử dụng càng ít file càng tốt. Tuy nhiên, việc đưa vào các file là cần thiết cho việc đảm bảo an toàn dữ liệu (file bảo vệ, file lịch sử), tăng tốc độ truy cập hay xử lý (file giao dịch, file lấy từ bảng, file làm việc)

7.4.2 Các phương pháp truy cập

Mỗi hệ điều hành trợ giúp một số kỹ thuật khác nhau để tìm kiếm và lấy thông tin ra gọi là các phương pháp truy cập.

Về cơ bản, có 2 loại phương pháp truy cập:

Phương pháp truy cập trực tiếp, sử dụng tính toán để xác định địa chỉ chính xác của một bản ghi và truy nhập trực tiếp đến bản ghi đó.

Phương pháp gián tiếp, hỗ trợ việc tìm kiếm bản ghi thứ n xuất phát từ một vị trí hiện thời của con trỏ hay điểm bắt đầu của 1 file.

Mặc dù có tồn tại 2 phương pháp như vậy, nhưng ít khi chúng ta sử dụng trực tiếp mà thường thông qua các công cụ có sẵn mà phần mềm hệ thống trợ giúp.

7.4.3 Tổ chức file

Cách tổ chức file là kỹ thuật sắp xếp các bản ghi vật lý của một file trên một thiết bị nhớ thứ cấp. Tổ chức một file cụ thể cần tính toán đến các yếu tố sau:

Lấy dữ liệu nhanh.

Thông lượng các giao dịch xử lý lớn

Sử dụng hiệu quả không gian nhớ.

Tránh được sai sót khi mất dữ liệu

Tối ưu hóa nhu cầu tổ chức file.

Đáp ứng được nhu cầu khi tăng dữ liệu

Trước khi nghiên cứu thiết kế tổ chức các CSDL cần xem xét sơ qua cách tổ chức của dữ liệu trong bộ nhớ ngoài.

Mô hình tổ chức bộ nhớ ngoài.

Bộ nhớ ngoài (hay còn gọi là bộ nhớ thứ cấp) là các thiết bị lưu trữ như đĩa từ, băng từ...

Đĩa từ được phân thành các khối vật lý (tổ chức đồng) có kích cỡ như nhau: khoảng 512 bytes đến 4K ($4 \times 1024 = 4096$ bytes) và được đánh địa chỉ khối. Địa chỉ này gọi là địa chỉ tuyệt đối trên đĩa.

Mỗi tệp dữ liệu trên đĩa từ chiếm 1 hoặc nhiều khối, mỗi khối chứa 1 hoặc nhiều bản ghi. Việc thao tác với tệp thông qua tên tệp thực chất là thông qua địa chỉ tuyệt đối của các khối.

Mỗi bản ghi đều có địa chỉ và thường được xem là địa chỉ tuyệt đối của byte đầu tiên của bản ghi hoặc là địa chỉ của khối chứa bản ghi đó.

Các phép toán đặc trưng trên tệp dữ liệu là:

Thêm một bản ghi.

Xóa một bản ghi

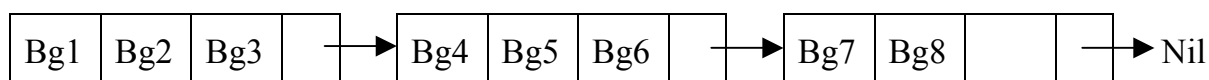
Sửa một bản ghi

Tìm một bản ghi theo điều kiện

7.4.3.1 Tổ chức file tuần tự (Sequential file)

Trong tổ chức file tuần tự, các bản ghi được sắp xếp tuần tự.

Giả sử: Hiện tại tệp cơ 8 bản ghi. 1 khối chứa 3 bản ghi. Con trỏ tệp trỏ đến bản ghi đầu tiên.



Thêm bản ghi: lần theo con trỏ đến kiểm tra khối cuối cùng (khối 3) xem có đủ bộ nhớ không. Nếu không đủ thì phải cấp thêm khối mới.

Xóa bản ghi (có giá trị khóa =k):

- Tìm đến bản ghi đó
- Xóa :
- Xóa logic: chỉ đánh dấu xóa
- Xóa vật lý: Xóa hẳn bản ghi đó

Tìm kiếm bản ghi (có giá trị khóa =k): Tìm lần lượt từ trên xuống dưới cho đến khi gặp khóa k.

Sửa đổi giá trị thuộc tính:

- Tìm đến thuộc tính cần sửa
- Ghi đè giá trị mới lên giá trị cũ

7.4.3.2 Tổ chức file băm (Hashed File)

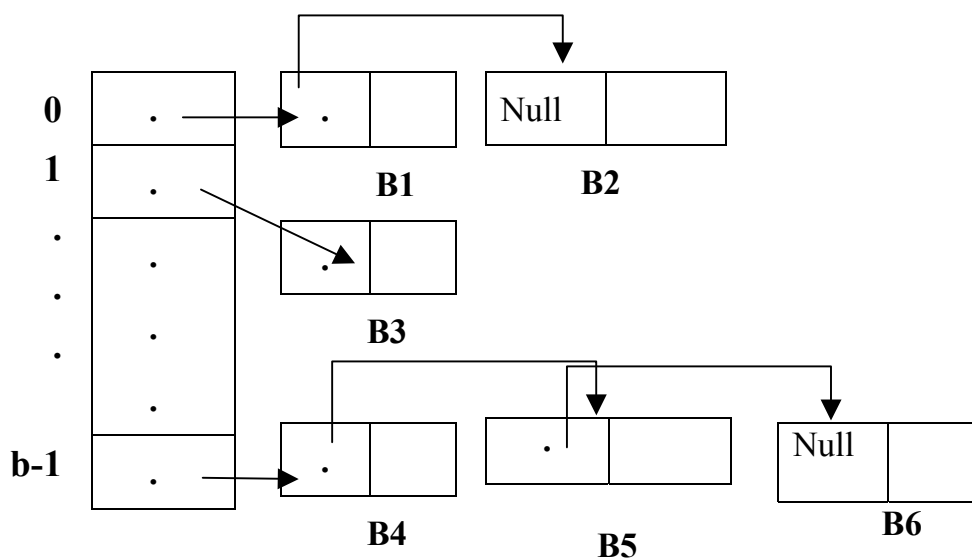
- Khái niệm hàm băm: Mỗi bản ghi đều có 1 khóa là giá trị số (giả sử là k)
- Hàm băm $H(k)=b$. Trong đó, b là số cụm (hàm băm sẽ tác động lên giá trị khóa và trả lại 1 số nguyên là số cụm)

Như vậy, tư tưởng của hàm băm là phân chia tập hợp các bản ghi của tệp dữ liệu thành các cụm (Buckets). Mỗi cụm bao gồm một hoặc nhiều khối, mỗi khối chứa một số lượng cố định các bản ghi.

Mỗi cụm ứng với một địa chỉ băm được đánh số từ 0..b-1. Ở mỗi đầu của khối đều chứa con trỏ trỏ đến khối tiếp theo trong cụm, khối cuối cùng trong cụm chứa con trỏ rỗng.

Có một bảng chỉ dẫn cụm (bucket directory): chứa k con trỏ, mỗi con trỏ chứa địa chỉ khối đầu tiên của từng cụm.

Ví dụ:



Hình 7.1. Tổ chức file băm

Thêm bản ghi mới (với giá trị khóa k):

Tính $H(k)$: Nếu $H(k)=b$ thì bổ sung vào nhóm b (Lần theo con trỏ, thêm vào như tổ chức tuần tự).

Tìm kiếm bản ghi (với giá trị khóa k):

Tính $H(k)$: Nếu $H(k)=b$ thì b chính là nhóm chứa bản ghi có khóa là k. Sau đó tìm kiếm tuần tự trên nhóm đó.

Xóa bản ghi (với giá trị khóa k):

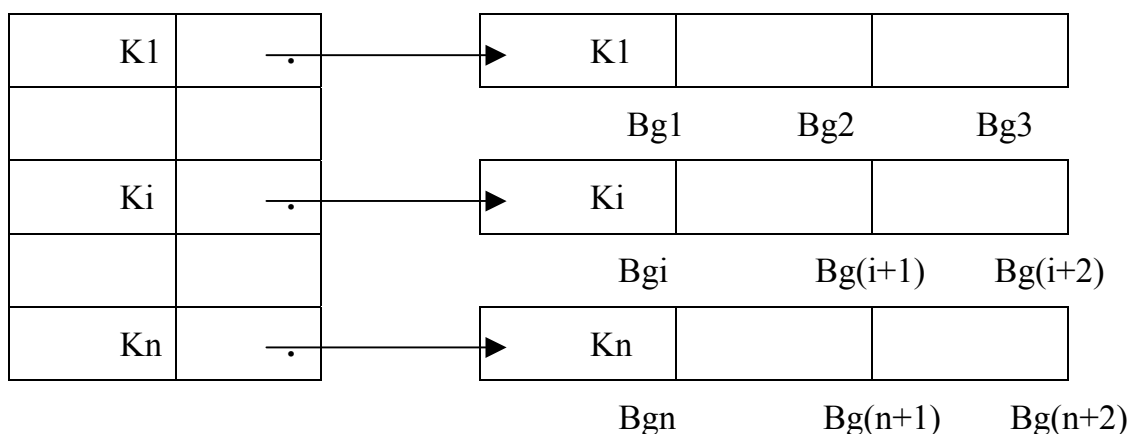
- Tìm đến bản ghi có khóa k bằng cách tính $H(k)$: Nếu $H(k)= b$ thì tìm bản ghi đó trong nhóm b.
- Xóa logic hoặc xóa vật lý.
- Nếu bản ghi là duy nhất trong khối thì khi xóa bản ghi sẽ đồng thời giải phóng khối khỏi cụm chứa khối.

Sửa đổi thuộc tính:

- Sửa đổi thuộc tính không phải là khóa: Ghi đè giá trị mới lên giá trị cũ.
- Sửa đổi thuộc tính khóa: Xóa bản ghi cũ, thêm bản ghi mới vào.
- Tính $H(k')$.
 - Nếu $h(k) =h(k')$ → Ghi đè lên
 - Nếu $h(k) \neq h(k')$ → Xóa bản ghi cũ, thêm bản ghi mới vào.

7.4.3.3 Tổ chức file chỉ số

Một kiểu tổ chức tệp dữ liệu truy nhập khóa rất thường dùng trong CSDL là tệp chỉ số.



Hình 7.2. Tổ chức file chỉ số

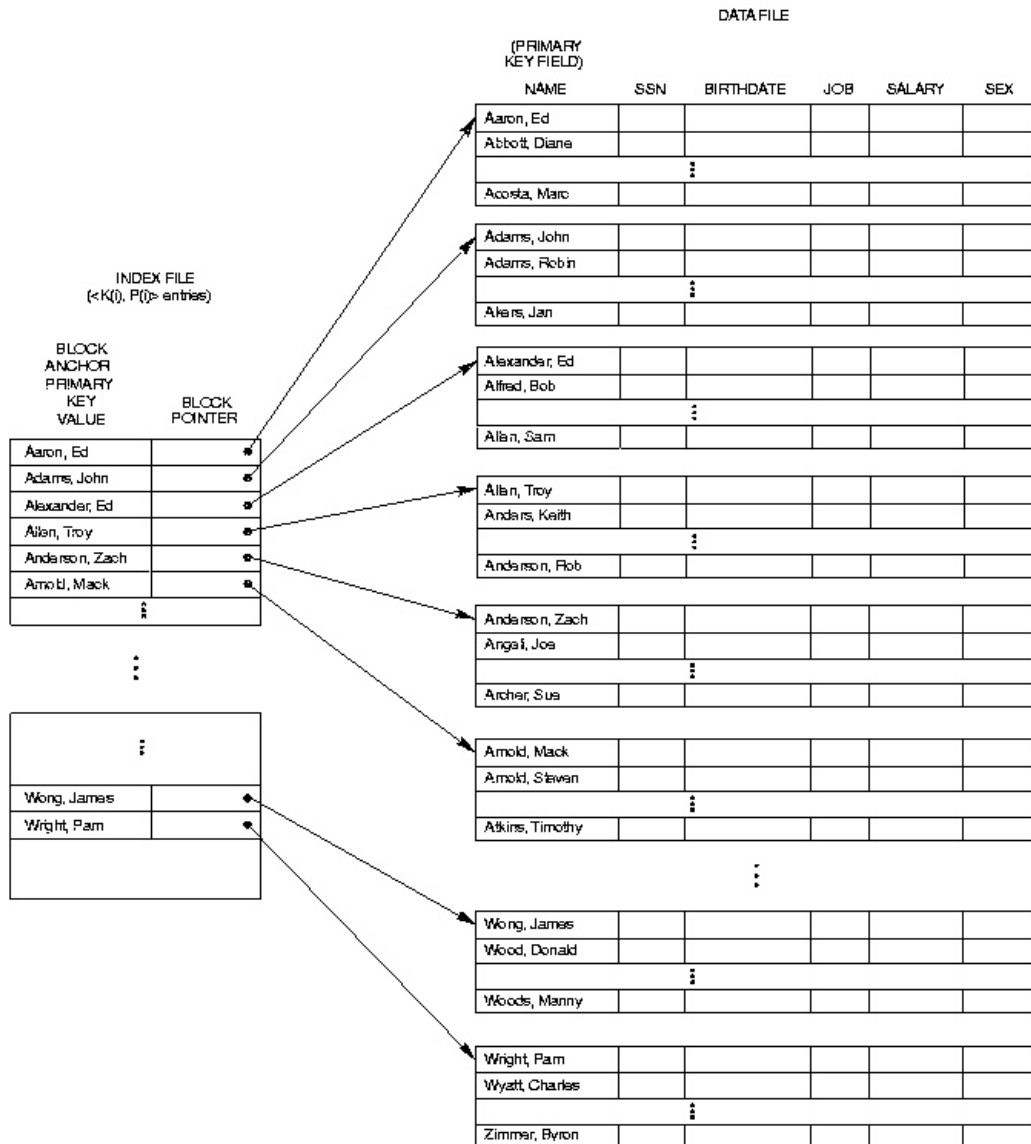
Trong đó:

K_1 : Giá trị khóa của bản ghi đầu tiên của khối thứ nhất.

Tệp chỉ dẫn có 2 trường: khóa và con trỏ

Tập chính có n khối, trong khối chứa các bản ghi.

Một minh họa về tập chỉ số:



Hình 7.3. Minh họa tập chỉ số

Thêm bản ghi (có giá trị khóa k):

- Tìm vị trí khối cần thêm ($H(k)=b$).
- Nếu khối đó còn chỗ trống thì thêm bản ghi vào theo đúng thứ tự sắp xếp và chú ý thay đổi khóa trong bảng chỉ dẫn khối nếu có sự thay đổi.

Xóa bản ghi (có giá trị khóa k):

- Quá trình giống như thêm một bản ghi. Tuy nhiên, nếu khi xóa tạo ra khối rỗng, khi đó có thể xóa bỏ toàn bộ khối.
- Tìm kiếm bản ghi (có giá trị khóa k): Có thể tìm kiếm tuần tự hoặc nhị phân.

Sửa đổi thuộc tính:

- Thuộc tính khóa: Xóa cũ, thêm mới.
- Thuộc tính không khóa: Ghi đè lên thuộc tính cũ.

Bảng so sánh các cách tổ chức file khác nhau

Yếu tố	Cách tổ chức file		
	Tuần tự	Chỉ số	Băm
Không gian lưu trữ	Không lãng phí	Cần nhiều không gian hơn vì phải lưu file chỉ số	Cần nhiều không gian khi thêm và xóa bản ghi (phải tính H(k))
Truy nhập tuần tự theo khóa chính	Rất nhanh	Trung bình	Không thực tế
Truy nhập ngẫu nhiên theo khóa chính	Không thực tế	Trung bình	Rất nhanh
Xóa bản ghi	Tạo ra khoảng trống và có thể yêu cầu tổ chức lại	Nếu không gian nhớ được bố trí động thì dễ dàng, nhưng yêu cầu phải bảo trì các chỉ số.	Rất dễ dàng
Thêm bản ghi	Yêu cầu sắp xếp lại file sau khi thêm	Nt	Nt
Sửa bản ghi	Yêu cầu sắp xếp lại file sau khi sửa	Dễ dàng, nhưng yêu cầu phải bảo trì các chỉ số	Rất dễ dàng

7.4.4 Ví dụ về thiết kế file

Xét Ví dụ sau: Có 2 chứng từ

<p>ĐƠN ĐẶT HÀNG</p> <p style="text-align: right;">Số hóa đơn: xxxxxx</p> <p>Người đặt hàng: (30 ký tự.....)</p> <p>Địa chỉ: (50 ký tự.....)</p> <p>Ngày đặt: dd/mm/yyyy</p>				
Số tt	Tên hàng	Mô tả hàng	Đơn vị tính	Số lượng
Xx	Char(15)	Char(50)	Char(10)	xxxxx
...

PHIẾU GIAO HÀNG

Số phiếu: xxxxxx

Tên khách hàng: (30 ký tự.....)

Địa chỉ: (50 ký tự.....)

Nơi giao hàng: (50 ký tự.....)

Ngày giao: dd/mm/yyyy

Số tt	Tên hàng	Đơn vị tính	Đơn giá	Số lượng	Thành tiền
Xx	Char(15)	Char(10)	Number	Number	Number
...

Sau quá trình thiết kế logic ta được các quan hệ sau:

KHACH(Ma_Khach, Ten_kh, DiaChi_kh)

HANG(Ma_hang, Ten_hang, Mota_hang, Donvi)

DONHANG(So_Don, Ma_Khach, NgayDon)

PHIEUGIAO(So_Phieu, Ma_Khach, Noi_Giao, Ngay_Giao)

DONGDON(So_Don, Ma_Hang, So_luongD)

DONGPHIEUGIAO(So_Phieu, MaHang, So_luonggi, DonGia)

Khi tiến hành thiết kế vật lý, ta được các file sau:

1. KHACH

Tên thuộc tính	Kiểu dữ liệu	Kích cỡ	Khuôn dạng
<u>Ma khach</u>	Ký tự	6	Chữ hoa +số
Ten_kh	Ký tự	30	Chữ đầu viết hoa
Diachi_kh	Ký tự	30	Chữ đầu viết hoa

2. HANG

Tên thuộc tính	Kiểu dữ liệu	Kích cỡ	Khuôn dạng
<u>Ma hang</u>	Ký tự	6	Chữ hoa+số
Ten_hang	Ký tự	15	Chữ đầu viết hoa
Mota_hang	Ký tự	30	Chữ đầu viết hoa
Don_vi	Ký tự	10	Chữ thường

3. DONHANG

TÀI LIỆU THAM KHẢO

Tên thuộc tính	Kiểu dữ liệu	Kích cỡ	Khuôn dạng
<u>So_don</u>	Ký tự	6	Chữ hoa +số
<u>Ma_khach</u>	Ký tự	15	Chữ hoa +số
Ngay_don	Ngày	8	Dd/mm/yy

4. DONGDON

Tên thuộc tính	Kiểu dữ liệu	Kích cỡ	Khuôn dạng
<u>So_don</u>	Ký tự	6	Chữ hoa +số
<u>Ma_hang</u>	Ký tự	6	Chữ hoa +số
So_luongd	Số	7	Số nguyên

5. PHIEUGIAO

Tên thuộc tính	Kiểu dữ liệu	Kích cỡ	Khuôn dạng
<u>So_phieu</u>	Ký tự	6	Chữ hoa +số
<u>Ma_khach</u>	Ký tự	6	Chữ hoa +số
Ngay_giao	Ngày	30	Dd/mm/yy
Noi_giao	Ký tự	30	
Tong_tien	Số	9	Số thực

6. DONGPHIEU

Tên thuộc tính	Kiểu dữ liệu	Kích cỡ	Khuôn dạng
<u>So_phieu</u>	Ký tự	6	Chữ hoa+ số
<u>Ma_hang</u>	Ký tự	6	Chữ hoa + số
Don_gia	Số	6	Số thực
So_luonggi	Số	7	Số thực

TÀI LIỆU THAM KHẢO

1. Elmasri & Navathe: *Fundamentals of Database Systems*, International Edition.
2. Nguyễn Tiên Vương: *Cơ sở dữ liệu quan hệ*
3. Date, C.J., and Darwen, H.: *A Guide to the SQL Standard, 3rd ed.*, Addison-Wesley.