

TRƯỜNG ĐẠI HỌC ĐÀ LẠT
KHOA CÔNG NGHỆ THÔNG TIN

CAO THỊ NHẬN

NGUYỄN THỊ THANH BÌNH

GIÁO TRÌNH

CƠ SỞ DỮ LIỆU

Dành cho sinh viên ngành Công nghệ Thông tin

Đà Lạt 2010

LỜI MỞ ĐẦU

Để đáp ứng nhu cầu học tập của sinh viên chuyên ngành Công nghệ Thông tin, bài giảng Cơ sở dữ liệu được biên soạn theo chương trình hệ thống tín chỉ của Trường Đại học Đà Lạt, cung cấp các kiến thức cơ bản về lý thuyết cơ sở dữ liệu.

Giáo trình gồm 8 chương sau:

Chương 1: giới thiệu chung

Chương 2: trình bày mô hình thực thể kết hợp E_R (Entity Relationship) để mô hình hóa các hoạt động trong thế giới thực, nhìn thế giới thực như là một tập các đối tượng căn bản được gọi là các thực thể, và các mối quan hệ ở giữa các đối tượng này. Mô hình được phát triển để làm thuận tiện cho việc thiết kế cơ sở dữ liệu bằng cách đặc tả một tổ chức.

Chương 3: trình bày mô hình dữ liệu quan hệ, các quy tắc chuyển đổi cơ sở dữ liệu biểu diễn dạng lược đồ E_R sang mô hình dữ liệu quan.

Chương 4, 5 và 6: trình bày các ngôn ngữ truy vấn trên cơ sở dữ liệu. Trong chương 4 trình bày ngôn ngữ đại số quan hệ, chương 5 trình bày ngôn ngữ tân từ và chương 6 trình bày ngôn ngữ truy vấn SQL.

Chương 7: trình bày các khái niệm liên quan đến phụ thuộc hàm, khóa, các thuật toán tìm phủ tối thiểu của tập phụ thuộc hàm, tìm khóa và cuối chương trình bày về ràng buộc toàn vẹn dữ liệu.

Chương 8: Chương này giới thiệu các dạng chuẩn, phân rã bảo toàn thông tin, bảo toàn phụ thuộc hàm, qua đó cũng trình bày cách phân rã bảo toàn bảo toàn thông tin và bảo toàn phụ thuộc.

Mặc dù đã rất cố gắng trong quá trình biên soạn bài giảng nhưng chắc chắn bài giảng sẽ còn nhiều thiếu sót và hạn chế. Rất mong nhận được sự đóng góp ý kiến quý báu của sinh viên và các bạn đọc để giáo trình ngày một hoàn thiện hơn.

MỤC LỤC

<i>Lời mở đầu</i>	2
Mục lục	3
Chương 1: Giới thiệu chung	8
1. Giới thiệu chung.....	8
1.1. Giới thiệu	8
1.2. Định nghĩa CSDL	10
1.3. Các đối tượng sử dụng CSDL.	11
1.4. Hệ quản trị CSDL	12
1.5. Các mức biểu diễn một CSDL.....	12
1.6. Sơ đồ tổng quát một hệ quản trị CSDL	14
1.7. Tính độc lập giữa dữ liệu và chương trình.	15
2. Các cách tiếp cận của một CSDL.....	15
2.1. Cách tiếp cận theo mô hình dữ liệu mạng	16
2.2. Mô hình dữ liệu phân cấp	17
2.3. Mô hình dữ liệu quan hệ thực thể.....	18
2.4. Mô hình dữ liệu quan hệ.....	18
2.5. Mô hình dữ liệu hướng đối tượng	19
3. Bài tập.....	19
Chương 2: Mô hình thực thể kết hợp	21
1. Mô hình thực thể kết hợp.	21
1.1. Thực thể - tập thực thể.....	21
1.2. Thuộc tính.....	21
1.3. Mối kết hợp.....	22
1.4. Bản số	23
1.5. Khoá.....	24

1.6. Số chiều của mỗi kết hợp.....	25
1.7. Tổng quát hóa và chuyên biệt hóa.....	25
1.8. Tập thực thể yếu	26
2. Ví dụ	27
3. Bài tập.....	29
 Chương 3: Mô hình dữ liệu quan hệ	30
1. Các khái niệm cơ bản	30
1.1. Thuộc tính.....	30
1.2. Quan hệ n ngôi.....	31
1.3. Bộ.....	31
1.4. Lược đồ quan hệ	32
1.5. Khóa của một quan hệ	33
1.6. Ràng buộc toàn vẹn	35
2. Các thao tác cơ bản trên quan hệ.....	35
2.1. Phép thêm	35
2.2. Phép xóa	36
2.3. Phép sửa.....	36
3. Các bước chuyển đổi từ mô hình thực thể kết hợp sang mô hình quan hệ	37
3.1. Biến các tập thực thể chuyên biệt hóa về dạng bình thường.....	37
3.2. Chuyển tất cả các tập thực thể thành quan hệ	38
3.3. Mỗi kết hợp.....	39
3.4. Nhập tất cả các quan hệ có cùng khóa.....	39
 Chương 4: Ngôn ngữ đại số quan hệ	40

1. Các phép toán cơ sở	40
1.1. Các phép toán tập hợp	40
1.2. Các phép toán quan hệ	41
2. Các phép toán khác	46
2.1. Phép kết hai quan hệ	46
2.2. Phép kết nối nội	47
2.3. Phép kết nối trái	48
2.4. Phép kết nối phải	49
2.5. Hàm kết hợp và gom nhóm	49
2.6. Các phép toán cập nhật trên quan hệ	50
3. Bài tập.....	52

Chương 5: Ngôn ngữ tân từ 55

1. Ngôn ngữ tân từ có biến là bộ	55
1.1. Một số khái niệm	55
1.2. Định nghĩa hình thức của phép tính bộ	55
1.3. Lượng từ tồn tại \exists và với mọi \forall	57
2. Ngôn ngữ tân từ có biến là miền giá trị.....	58
3. Bài tập.....	59

Chương 6: Ngôn ngữ truy vấn SQL 61

1. Các lệnh hỏi.....	61
1.1. Cú pháp lệnh truy vấn.....	61
1.2. Phép chiếu.....	62
1.3. Phép chọn	62
1.4. Phép kết	63

1.5. Một số lưu ý.....	63
2. Truy vấn lồng	65
3. Hàm kết hợp và gom nhóm	68
4. Các lệnh khai báo cấu trúc CSDL	69
5. Các thao tác cập nhật dữ liệu.....	72
6. Bài tập.....	73
Chương 7: Phụ thuộc hàm, khóa, ràng buộc toàn vẹn	74
1. Phụ thuộc hàm	74
1.1. Khái niệm phụ thuộc hàm.....	74
1.2. Hệ luật dẫn Amstrong.....	75
1.3. Thuật toán tìm bao đóng của tập thuộc tính	77
1.4. Bài toán thành viên.....	78
1.5. Phủ tối thiểu của một tập phụ thuộc hàm	78
2. Khóa	82
2.1. Định nghĩa	82
2.2. Thuật toán tìm khóa.....	82
3. Ràng buộc toàn vẹn	85
3.1. Định nghĩa – các yếu tố của ràng buộc toàn vẹn.....	85
3.2. Các loại ràng buộc toàn vẹn	87
4. Bài tập.....	93
Chương 8: Dạng chuẩn và chuẩn hóa CSDL	96
1. Dạng chuẩn của lược đồ quan hệ	96
1.1. Dạng chuẩn 1	96
1.2. Dạng chuẩn 2	97
1.3. Dạng chuẩn 3	100

1.4.	Dạng chuẩn BC	101
1.5.	Kiểm tra dạng chuẩn	101
2.	Phép phân rã	102
2.1.	Phân rã bảo toàn thông tin	102
2.2.	Phân rã bảo toàn phụ thuộc hàm.....	103
3.	Thiết kế CSDL bằng cách phân rã	104
3.1.	Phân rã thành dạng chuẩn BC (hoặc dạng chuẩn 3) bảo toàn thông tin.....	104
3.2.	Phân rã thành dạng chuẩn 3 vừa bảo toàn thông tin vừa bảo toàn phụ thuộc hàm	108
4.	Bài tập.....	108
	<i>Tài liệu tham khảo</i>	<i>109</i>

Chương 1

Giới Thiệu Chung

Chương 1 giới thiệu cho học viên các khái niệm cơ bản về cơ sở dữ liệu (CSDL) và hệ quản trị CSDL.

1. Giới thiệu chung

1.1. Giới thiệu

Xét một hệ thống các tập tin cổ điển: được tổ chức một cách riêng rẽ, phục vụ cho một mục đích của một đơn vị hay một đơn vị con trực thuộc cụ thể. Ứng dụng là một hệ quản lý nhân sự của một công ty.

Tại một công ty đã được trang bị máy vi tính cho tất cả các phòng ban. Trong đó:

- Phòng Tổ chức lưu giữ những thông tin về lý lịch của nhân viên như họ tên, bí danh, giới tính, ngày sinh, ngày vào đoàn, ngày vào đảng, ngày tuyển dụng, hệ số lương, hệ số phụ cấp, hoàn cảnh gia đình ... Phần mềm được sử dụng để quản lý là Microsoft Access.
- Phòng Tài vụ lưu giữ những thông tin về họ tên, hệ số lương, hệ số phụ cấp, đơn vị ... để tính toán lương và phụ cấp dựa trên danh sách do phòng tổ chức gửi sang. Các bảng tính này được làm bởi Microsoft Excel.
- Bộ phận Văn thư sử dụng Microsoft word để thảo các văn bản báo cáo về tình hình của đơn vị trong đó có nêu tổng số công nhân viên và trình độ chuyên môn được đào tạo.
- Trong khi đó tại tổng công ty các phòng ban cũng được trang bị máy vi tính và cũng quản lý những thông tin của các cán bộ chủ chốt của công ty con là giám đốc, phó giám đốc, trưởng phó phòng.

Việc phát triển của hệ thống quản lý như sau:

- Những chương trình ứng dụng mới phải được viết khi một nhu cầu nảy sinh.
- Các tập tin thường trực mới được tạo ra theo yêu cầu.
- Trong một thời gian dài các tập tin có thể ở các dạng thức khác nhau.
- Những chương trình ứng dụng có thể viết bởi các ngôn ngữ khác nhau.

Ưu điểm:

- Việc xây dựng những hệ thống tập tin riêng lẻ tại từng đơn vị quản lý ít tốn thời gian bởi lượng thông tin cần quản lý và khai thác là nhỏ, không đòi hỏi đầu tư nhiều nên triển khai nhanh.
- Thông tin được khai thác chỉ phục vụ cho mục đích hẹp nên khả năng đáp ứng nhanh chóng, kịp thời.

Hạn chế:

Do thông tin được tổ chức ở mỗi phòng ban mỗi khác (thông tin và công cụ triển khai) nên rõ ràng sự phối hợp về mặt tổ chức và khai thác ở các phòng ban rất khó khăn. Thông tin ở phòng ban này không được sử dụng cho phòng ban khác, tại công ty con với tổng công ty. Hơn nữa cùng một thông tin được cập nhật vào máy tại nhiều nơi khác nhau. Do đó có những hạn chế sau:

- **Dữ liệu dư thừa và không nhất quán**
 - ✓ Những thông tin giống nhau có thể bị trùng lặp ở một số nơi.
 - ✓ Tất cả những bản sao dữ liệu có thể không được cập nhật đúng.
- **Khó khăn trong việc truy xuất dữ liệu**
 - ✓ Có thể phải viết một trình ứng dụng mới thỏa yêu cầu mới đặt ra.
 - ✓ Có thể tạo ra dữ liệu này một cách thủ công nhưng mất thời gian.
- **Khó khăn khi viết chương trình ứng dụng đòi hỏi thông tin liên quan đến nhiều đơn vị**
 - ✓ Dữ liệu ở những tập tin khác nhau.
 - ✓ Dữ liệu ở các dạng thức khác nhau.

Ngoài ra còn một số vấn đề chưa được giải quyết như:

- **Nhiều người sử dụng**
 - ✓ Muốn thời gian trả lời nhanh hơn cho các tác vụ đồng thời.
 - ✓ Cần sự bảo vệ đối với những cập nhật đồng thời.
- **Các vấn đề bảo mật**
 - ✓ Mỗi người sử dụng hệ thống chỉ có thể truy xuất những dữ liệu mà họ được phép thấy.

- ✓ Chẳng hạn như những người tính lương và phụ cấp chỉ giải quyết các mẫu tin nhân viên, và không thể thấy được thông tin về lý lịch nhân viên; những thủ quỹ chỉ truy xuất những dữ liệu về các khoản chi...
- ✓ Khó khăn đề ràng buộc điều này trong những chương trình ứng dụng.
- **Các vấn đề toàn vẹn.**
 - ✓ Dữ liệu có thể được yêu cầu thỏa mãn những ràng buộc.
 - ✓ Chẳng hạn như nhân viên đang hưởng chế độ nào đó ở phòng tổ chức thì sẽ không được hưởng chế độ trợ cấp cao nhất của phòng công đoàn.
 - ✓ Với cách tiếp cận xử lý-tập tin, khó khăn để thực hiện hoặc thay đổi những ràng buộc như vậy.
- **Khó khăn khi nâng cấp hệ thống.**
 - ✓ Do hệ thống được tổ chức thành các tập tin riêng lẻ nên thiếu sự chia sẻ thông tin giữa các phòng ban. Bên cạnh đó việc kết nối các hệ thống này với nhau hay nâng cấp ứng dụng sẽ gặp rất nhiều khó khăn.

Những nhược điểm này đã dẫn tới việc phát triển các hệ thống quản trị cơ sở dữ liệu.

1.2. Định nghĩa cơ sở dữ liệu

Cơ sở dữ liệu (database) là một hệ thống các thông tin có cấu trúc được lưu trữ trên các thiết bị lưu trữ thứ cấp (băng từ, đĩa từ...) nhằm thỏa mãn yêu cầu khai thác thông tin đồng thời của nhiều người sử dụng hay nhiều chương trình ứng dụng với nhiều mục đích khác nhau.

Ưu điểm của CSDL:

- Giảm sự trùng lặp thông tin xuống mức thấp nhất và do đó bảo đảm được tính nhất quán và toàn vẹn dữ liệu
- Dữ liệu có thể được truy xuất theo nhiều cách khác nhau
- Khả năng chia sẻ thông tin cho nhiều người sử dụng và nhiều ứng dụng khác nhau

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

- **Tính chủ quyền của dữ liệu.** Do sự chia sẻ của CSDL nên tính chủ quyền của dữ liệu có thể bị lu mờ và làm mờ nhạt tinh thần trách nhiệm, được thể hiện trên vấn đề an toàn dữ liệu, khả năng biểu diễn các mối liên hệ ngữ nghĩa của dữ liệu và tính chính xác của dữ liệu. Nghĩa là người khai thác CSDL phải có nghĩa vụ cập nhật các thông tin mới nhất của CSDL.

- **Tính bảo mật và quyền khai thác thông tin của người sử dụng (NSD).** Do có nhiều người cùng khai thác CSDL một cách đồng nên cần phải có một cơ chế bảo mật và phân quyền khai thác CSDL.
- **Tranh chấp dữ liệu.** Nhiều người được phép truy cập vào cùng một tài nguyên dữ liệu (data source) của CSDL với những mục đích khác nhau như xem, thêm, xóa, sửa dữ liệu. Như vậy cần phải có cơ chế ưu tiên truy nhập dữ liệu cũng như cơ chế giải quyết tình trạng khoá chết (DeadLock) trong quá trình khai thác cạnh tranh. Cơ chế ưu tiên có thể được thực hiện dựa trên:
 - ✓ Cấp quyền ưu tiên cho người khai thác CSDL, người nào có quyền ưu tiên cao hơn thì được ưu tiên cấp quyền truy nhập dữ liệu trước.
 - ✓ Thời điểm truy nhập, ai có yêu cầu truy xuất trước thì có quyền truy nhập dữ liệu trước.
 - ✓ Hoặc dựa trên cơ chế lập lịch truy xuất hay cơ chế khoá.
 - ✓ ...
- **Đảm bảo dữ liệu khi có sự cố.** Việc quản lý tập trung có thể làm tăng khả năng mất mát hoặc sai lệch thông tin khi có sự cố như mất điện đột xuất, một phần hay toàn bộ đĩa lưu trữ dữ liệu bị hư... Một số hệ điều hành mạng có cung cấp dịch vụ sao lưu 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, một CSDL nhất thiết phải có một cơ chế khôi phục dữ liệu khi các sự cố bất ngờ xảy ra để đảm bảo CSDL luôn ổn định.

1.3. Các đối tượng sử dụng CSDL

- Những người sử dụng không chuyên về lĩnh vực tin học và CSDL, do đó CSDL cần có các công cụ để giúp cho người sử dụng không chuyên có thể khai thác hiệu quả CSDL
- Các chuyên viên tin học biết khai thác CSD, đây là những người có thể xây dựng các ứng dụng khác nhau phục vụ cho những yêu cầu khác nhau trên CSDL.
- Những người quản trị CSDL, là những người hiểu biết về tin học, các hệ quản trị CSDL và hệ thống máy tính. Họ là người tổ chức CSDL (khai báo cấu trúc CSDL, ghi nhận các yêu cầu bảo mật cho các dữ liệu cần bảo vệ...), do đó họ cần phải nắm rõ các vấn đề về kỹ thuật để có thể phục hồi dữ liệu khi có sự cố. Họ là những người cấp quyền hạn khai thác CSDL nên họ có thể giải quyết các vấn đề tranh chấp dữ liệu.

1.4. Hệ quản trị CSDL

Để giải quyết tốt những vấn đề đặt ra cho một CSDL như đã trình bày trong 1.2 thì cần có một hệ thống các phần mềm chuyên dụng. Hệ thống này được gọi là hệ quản trị CSDL (Database Management System - DBMS), là công cụ hỗ trợ tích cực cho các nhà phân tích, thiết kế và khai thác CSDL. Hiện nay có nhiều hệ quản trị CSDL trên thị trường như: Visual Foxpro, SQL server, DB2, Microsoft Access, Oracle...

Một hệ quản trị CSDL phải có:

- Ngôn ngữ giao tiếp giữa NSD và CSDL:
 - ✓ Ngôn ngữ mô tả dữ liệu (Data Definition Language - DDL) để cho phép khai báo cấu trúc của CSDL, khai báo các mối liên hệ của dữ liệu (Data Relationship) và các quy tắc (Rules, Constraint) quản lý áp đặt trên dữ liệu.
 - ✓ Ngôn ngữ thao tác dữ liệu (Data Manipulation Language - DML) cho phép NSD có thể thêm, xóa, dữ liệu trong CSDL.
 - ✓ Ngôn ngữ truy vấn dữ liệu (Structured Query Language - SQL) cho phép NSD truy vấn các thông tin cần thiết.
 - ✓ Ngôn ngữ quản lý dữ liệu (Data Control Language - DCL) cho phép những người quản trị hệ thống thay đổi cấu trúc của các bảng dữ liệu, khai báo bảo mật thông tin và cấp quyền khai thác CSDL cho NSD.
- Tự điển dữ liệu (Data dictionary) dùng để mô tả các ánh xạ liên kết, ghi nhận các thành phần cấu trúc của CSDL, các chương trình ứng dụng, mật mã, quyền sử dụng...
- Có biện pháp bảo mật tốt.
- Có cơ chế giải quyết tranh chấp dữ liệu.
- Có cơ chế sao lưu và phục hồi dữ liệu khi có sự cố xảy ra.
- Có giao diện tốt, dễ sử dụng.
- Bảo đảm tính độc lập giữa dữ liệu và chương trình: khi có sự thay đổi dữ liệu (như sửa đổi cấu trúc lưu trữ các bảng dữ liệu, thêm các chỉ mục,...) thì các chương trình ứng dụng đang chạy trên CSDL vẫn không cần phải viết lại hay cũng không ảnh hưởng đến NSD khác.

1.5. Các mức biểu diễn một CSDL

Để hệ thống có thể dùng được thì phải tìm dữ liệu một cách hiệu quả. Điều này dẫn đến việc cần phải thiết kế những cấu trúc dữ liệu phức tạp đối với việc biểu diễn dữ liệu trong

cơ sở dữ liệu. Vì nhiều người sử dụng các hệ thống cơ sở dữ liệu không phải là người hiểu biết nhiều về máy tính nên những người phát triển hệ thống phải che dấu đi sự phức tạp khỏi người sử dụng thông qua một số mức trừu tượng, nhằm làm đơn giản hóa sự tương tác của người sử dụng đối với hệ thống.

Có 3 mức biểu diễn dữ liệu, cụ thể như sau:

Mức trong (mức vật lý - Physical level)

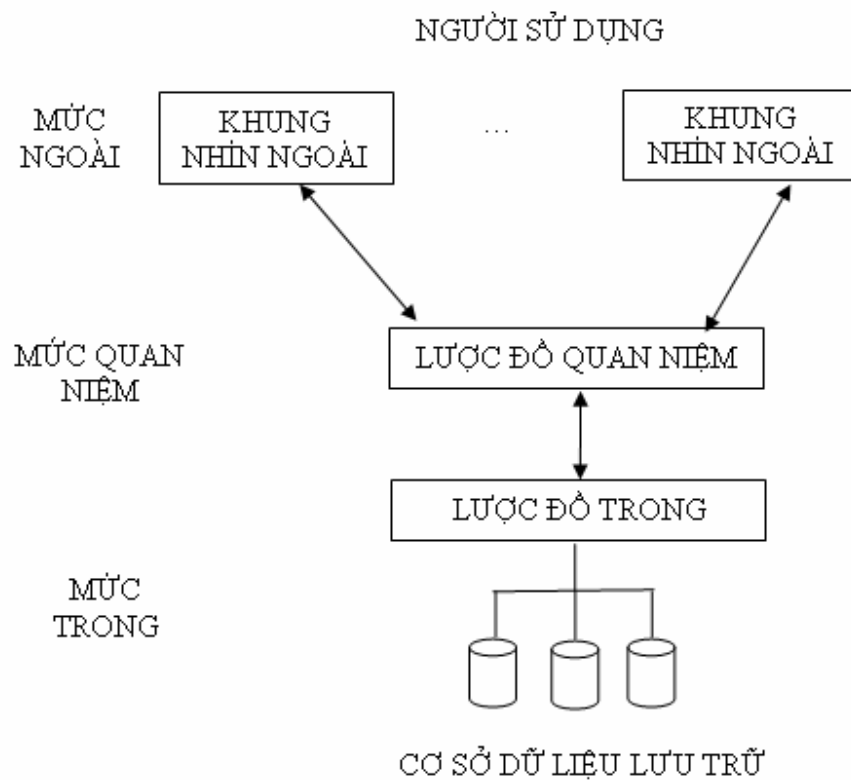
- Nói đến cách thức dữ liệu được lưu trữ thực sự. Trả lời câu hỏi dữ liệu gì và lưu trữ chúng như thế nào? Cần các chỉ mục gì?
- Ví dụ như chỉ mục, B-cây, băm.
- Những người thao tác tại mức này là người quản trị CSDL và những NSD chuyên môn.

Mức quan niệm hay mức logic (conception level, logical level)

- Trả lời câu hỏi cần phải lưu trữ bao nhiêu loại dữ liệu? Đó là những dữ liệu gì? Mối quan hệ giữa chúng như thế nào?
- Như vậy CSDL mức vật lý là sự cài đặt cụ thể của CSDL mức quan niệm.

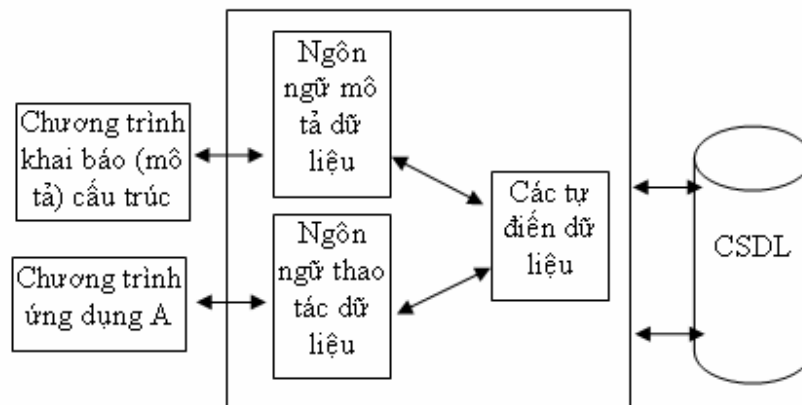
Mức ngoài hay mức nhìn (view level)

- Là mức của NSD và các chương trình ứng dụng.
- Làm việc tại mức này là các nhà chuyên môn, kỹ sư tin học, NSD không chuyên.
- Mỗi NSD hay mỗi chương trình ứng dụng có thể được nhìn CSDL theo một góc độ khác nhau. Tùy thuộc vào vai trò của NSD mà có thể thấy tất cả hoặc một phần của CSDL. NSD hay chương trình ứng dụng có thể không được biết về cấu trúc tổ chức lưu trữ thông tin trong CSDL. Họ chỉ có thể làm việc trên một phần CSDL theo cách nhìn do người quản trị hay chương trình ứng dụng quy định, gọi là khung nhìn.



Hình 1.1. Ba mức trừu tượng dữ liệu

1.6. Sơ đồ tổng quát một hệ quản trị CSDL



Hình 1.2. Sơ đồ tổng quát của một hệ quản trị CSDL

Hình 1.2 minh họa sơ đồ tổng quát của một hệ quản trị CSDL. Có 3 mức: mức chương trình khai báo cấu trúc và chương trình ứng dụng; mức mô tả CSDL, thao tác CSDL và các tự điển dữ liệu; mức CSDL.

Mỗi hệ quản trị CSDL có một ngôn ngữ khai báo hay mô tả (Data Definition Language - DDL) cấu trúc CSDL riêng. Những người thiết kế và quản trị CSDL thực hiện các công việc khai báo cấu trúc CSDL.

Các chương trình khai báo cấu trúc CSDL được viết bằng ngôn ngữ mà hệ quản trị CSDL cho phép. Hai công việc khai báo cấu trúc logic (là việc khai báo các loại dữ liệu và các mối liên hệ giữa các loại dữ liệu đó, cùng các ràng buộc toàn vẹn dữ liệu) và khai báo vật lý (dữ liệu được lưu trữ theo dạng nào, có bao nhiêu chỉ mục)

Các chương trình ứng dụng được viết bằng ngôn ngữ thao tác CSDL (Data Manipulation Language - DML) với mục đích:

- Truy xuất dữ liệu
- Cập nhật dữ liệu (thêm, xóa, sửa)
- Khai thác dữ liệu
- Ngôn ngữ thao tác CSDL còn được sử dụng cho những người sử dụng thao tác trực tiếp với CSDL.

Tự điển dữ liệu (Data Dictionary - DD) là một CSDL của hệ quản trị CSDL sử dụng để lưu trữ cấu trúc CSDL, các thông tin bảo mật, bảo đảm an toàn dữ liệu và các cấu trúc ngoài. Tự điển dữ liệu còn được gọi là siêu CSDL (Meta-Database)

1.7. Tính độc lập giữa dữ liệu và chương trình

- **Độc lập dữ liệu vật lý**
 - ✓ Khả năng cập nhật lược đồ vật lý không làm cho các chương trình ứng dụng phải bị viết lại.
 - ✓ Việc cập nhật ở mức này thường cần thiết để cải tiến hiệu suất.
- **Độc lập dữ liệu logic**
 - ✓ Khả năng cập nhật lược đồ logic mà không làm cho các chương trình ứng dụng phải bị viết lại.
 - ✓ Những cập nhật thường được thực hiện khi cấu trúc logic của cơ sở dữ liệu thay đổi.

2. Các cách tiếp cận của một CSDL

Nền tảng của cấu trúc cơ sở dữ liệu là mô hình dữ liệu. Mô hình dữ liệu được định nghĩa là một sưu tập các công cụ khái niệm dùng cho việc mô tả dữ liệu, các mối quan hệ dữ liệu, các ngữ nghĩa dữ liệu và các ràng buộc dữ liệu.

2.1. Cách tiếp cận theo mô hình dữ liệu mạng

Mô hình dữ liệu mạng (Network Data Model) hay gọi tắt là mô hình mạng (Network Model) là mô hình được biểu diễn bởi một đồ thị có hướng. Mô hình này được xây dựng bởi Honeywell vào năm 1964-1965. Trong mô hình này, có 3 khái niệm được sử dụng: mẫu tin hay bản ghi (record), loại mẫu tin (record type) và loại liên hệ (set type).

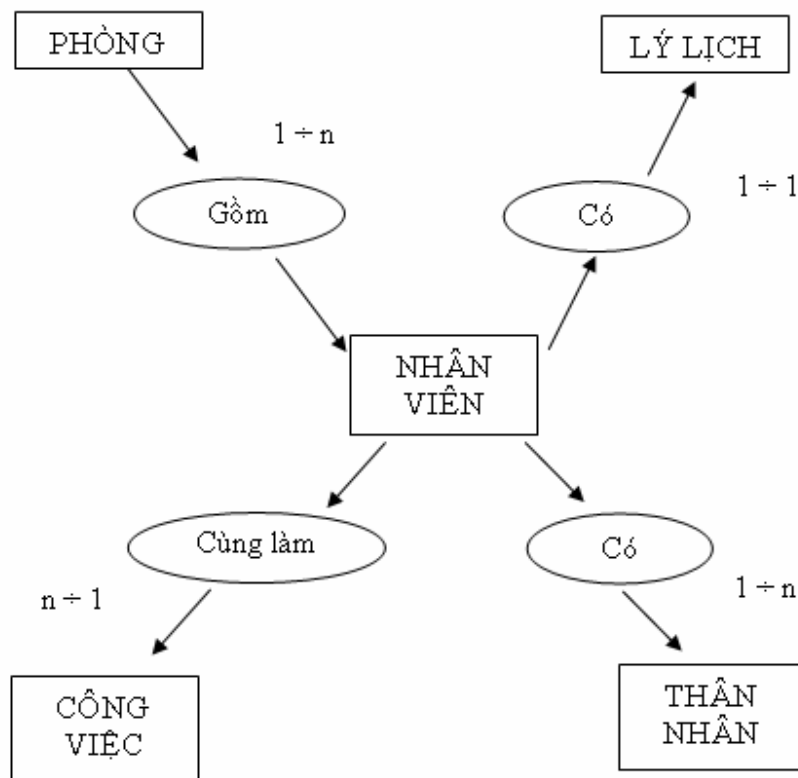
Loại mẫu tin: đặc trưng cho một loại đối tượng riêng biệt. Chẳng hạn như trong quản lý nhân sự tại một đơn vị thì đối tượng cần phản ánh của thế giới thực có thể là Phòng, nhân viên, công việc, lý lịch... do đó có các loại mẫu tin đặc trưng cho từng đối tượng này. Trong mô hình mạng, mỗi loại mẫu tin được thể hiện bằng một hình chữ nhật, một thể hiện (instance) của một loại mẫu tin được gọi là bản ghi (record). Ở ví dụ trên loại mẫu tin Phòng có các mẫu tin là các phòng ban trong công ty, loại mẫu tin Nhân viên có các mẫu tin là các nhân viên đang làm việc tại các phòng ban của công ty...

Loại liên hệ: là sự liên kết giữa một loại mẫu tin chủ với một mẫu tin thành viên. Trong mô hình mạng, một loại liên hệ được thể hiện bằng hình bầu dục và sự liên kết giữa hai loại mẫu tin được thể hiện bằng các cung có hướng đi từ loại mẫu tin chủ đến loại liên hệ và từ loại liên hệ đến loại mẫu tin thành viên.

Trong loại liên kết người ta còn chỉ ra số lượng các mẫu tin tham gia vào trong mỗi kết hợp. Có một số loại sau:

- $1 \div 1$ (One to One): Mỗi mẫu tin của loại mẫu tin chủ kết hợp với đúng 1 mẫu tin của loại mẫu tin thành viên. Ví dụ, mỗi nhân viên có duy nhất một lý lịch cá nhân.
- $1 \div n$ (One to Many): Mỗi mẫu tin của loại mẫu tin chủ kết hợp với 1 hay nhiều mẫu tin của loại mẫu tin thành viên. Ví dụ, mỗi phòng ban có thể có một hoặc nhiều nhân viên.
- $n \div 1$ (Many to One): Nhiều mẫu tin của loại mẫu tin chủ kết hợp với đúng 1 mẫu tin của loại mẫu tin thành viên. Ví dụ, nhiều nhân viên cùng làm một công việc.
- Đệ quy (Recursive): Một loại mẫu tin chủ cũng có thể đồng thời là mẫu tin thành viên với chính nó. Ví dụ, trưởng phòng cũng là một nhân viên thuộc phòng trong công ty.

Hình 1.4 minh họa một mô hình dữ liệu mạng.



Hình 1.4. Mô hình dữ liệu mạng

Ưu điểm: đơn giản, dễ sử dụng.

Hạn chế: không thích hợp trong việc biểu diễn CSDL lớn do hạn chế về khả năng diễn đạt ngữ nghĩa của dữ liệu, đặc biệt là các dữ liệu và mối liên hệ phức tạp của dữ liệu trong thực tế là rất hạn chế.

2.2. Mô hình dữ liệu phân cấp

Mô hình dữ liệu phân cấp (Hierarchical Data Model), còn gọi là mô hình phân cấp (Hierarchical Model), được thực hiện thông qua sự kết hợp giữa IBM và North American Rockwell vào khoảng năm 1965. Mô hình là một cây, trong đó mỗi nút của cây biểu diễn một thực thể, giữa nút con với nút cha được liên hệ với nhau theo một mối quan hệ xác định.

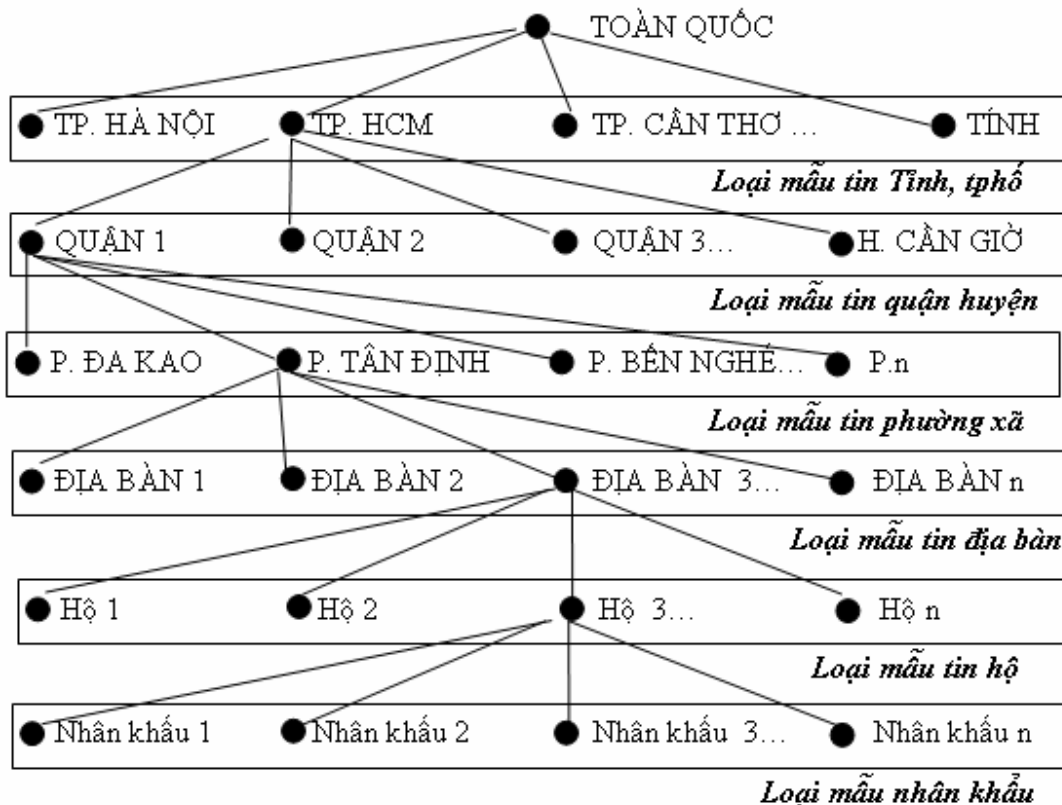
Loại mẫu tin: giống trong mô hình dữ liệu mạng

Loại mối liên hệ: kiểu liên hệ là phân cấp:

Mẫu tin thành viên chỉ đóng vai trò thành viên của một mối liên hệ duy nhất, tức là nó thuộc một chủ duy nhất. Như vậy, mối liên hệ từ mẫu tin chủ tới các mẫu tin thành viên là $1 \div n$ và từ mẫu tin thành viên với mẫu tin chủ là $1 \div 1$

Giữa hai loại mẫu tin chỉ tồn tại một mối quan hệ duy nhất.

Hình 1.5 minh họa mô hình dữ liệu phân cấp một CSDL về điều tra dân số.



Hình 1.5. Mô hình dữ liệu phân cấp

2.3. Mô hình dữ liệu quan hệ

Mô hình dữ liệu quan hệ (Relational Data Model) còn được gọi là mô hình quan hệ (Relational Model) do E.F.Codd đề xuất năm 1970. Nền tảng cơ bản là khái niệm lý thuyết tập hợp trên các quan hệ, tức là tập của các bộ giá trị (value tuples). Trong mô hình dữ liệu này những khái niệm được sử dụng là thuộc tính (attribute), quan hệ (relation), lược đồ quan hệ (relation schema), bộ (tuple), khóa (key).

Mô hình quan hệ là mô hình được nghiên cứu nhiều nhất, và có cơ sở lý thuyết vững chắc nhất. Mô hình quan hệ cùng với mô hình dữ liệu thực thể kết hợp đang được sử dụng rộng rãi nhất hiện nay trong việc phân tích và thiết kế CSDL. Chúng ta sẽ nghiên cứu chi tiết về mô hình này trong chương sau.

2.4. Mô hình dữ liệu thực thể - kết hợp

Mô hình dữ liệu thực thể - kết hợp (Entity – Relationship Model) do Peter Pin-Shan Chen đề xuất năm 1976. Mô hình quan hệ-thực thể được dựa trên sự nhận thức thể giới gồm có

một sưu tập các đối tượng căn bản, được gọi là các tập thực thể, và các mối quan hệ ở giữa các đối tượng này. Trong mô hình này các khái niệm được sử dụng là tập thực thể (entity set), thực thể (entity), thuộc tính của loại thực thể (entity attribute), khóa của loại thực thể (entity key), loại mối kết hợp (entity relationship), số ngôi của mối kết hợp (relationship degree), thuộc tính của mối kết hợp (relationship attribute), bản số của mối kết hợp (relationship cardinal).

Chi tiết của mô hình này sẽ được nghiên cứu trong chương sau.

2.5. Mô hình dữ liệu hướng đối tượng

Mô hình dữ liệu hướng đối tượng (Object Oriented Data Model) ra đời vào cuối những năm 80 và đầu những năm 90. Đây là loại mô hình tiên tiến nhất hiện nay dựa trên cách tiếp cận hướng đối tượng. Mô hình này sử dụng các khái niệm như lớp (class), sự kế thừa (inheritance), kế thừa bội (multi - inheritance). Đặc trưng cơ bản của cách tiếp cận này là tính đóng gói (encapsulation), tính đa hình (polymorphism) và tính tái sử dụng (reusability).

3. Bài tập

Bài 1:

Dựa vào những khái niệm đã học, hãy biểu diễn CSDL có các loại mẫu tin Phòng, Nhân viên, Công việc, lý lịch đã trình bày trong mô hình mạng theo cách tiếp cận phân cấp. Biết:

- Loại liên hệ là phân cấp.
- Phòng có nhiều nhân viên, mỗi nhân viên chỉ thuộc một phòng duy nhất
- Công việc có nhiều nhân viên cùng làm; mỗi nhân viên chỉ làm một công việc duy nhất
- Mỗi nhân viên có một lý lịch, mỗi lý lịch chỉ thuộc duy nhất một nhân viên.

Bài 2:

Dựa vào những khái niệm đã học, hãy biểu diễn CSDL về tổng điều tra dân số toàn quốc có các loại mẫu tin tỉnh – thành phố, quận huyện, phường xã, địa bàn, hộ điều tra và nhân khẩu đã trình bày trong mô hình phân cấp theo cách tiếp cận mạng. Biết:

- Nhân khẩu thuộc một hộ điều tra
- Hộ điều tra thuộc một địa bàn
- Địa bàn điều tra thuộc một phường xã

- Phường xã thuộc một quận huyện
 - Quận huyện thuộc một tỉnh, thành phố
-

Chương 2

Mô Hình Thực Thể Kết Hợp

Mô hình dữ liệu thực thể kết hợp (E-R - entity-relationship data model) do Peter Pin-Shan Chen đề xuất năm 1976, nhìn thế giới thực như là một tập các đối tượng căn bản được gọi là các *thực thể*, và *các mối quan hệ* ở giữa các đối tượng này. Mô hình đã được phát triển để làm thuận tiện cho việc thiết kế cơ sở dữ liệu bằng cách đặc tả một tổ chức. Một lược đồ như vậy biểu diễn một cấu trúc logic tổng quát của cơ sở dữ liệu.

1. Mô hình thực thể kết hợp

Những khái niệm căn bản mà mô hình thực thể kết hợp dùng: thực thể, tập thực thể, khóa, mối quan hệ, và thuộc tính.

1.1. Thực thể - Tập thực thể

Một *thực thể* (an entity) là một “sự vật” hoặc “đối tượng” mà nó tồn tại và có thể phân biệt được với các đối tượng khác. Ví dụ như một nhân viên trong một tổ chức là một thực thể.

Một *tập thực thể* (an entity set) là một tập hợp các thực thể cùng loại mà chúng chia sẻ cùng những tính chất hoặc thuộc tính. Ví dụ như tập hợp tất cả những người mà họ là nhân viên của một tổ chức là một tập thực thể *khách hàng*.

Mỗi tập thực thể được đặt một tên gọi, thông thường là danh từ. Ví dụ như KHACHHANG, HOADON,...

Ký hiệu: hình chữ nhật với tên gọi



Hình 2.1. Tập thực thể

1.2. Thuộc tính

Mỗi một tập thực thể có nhiều đặc trưng riêng được gọi là các thuộc tính. Mỗi một thuộc tính được đặt một tên, chẳng hạn như MaKH (mã khách hàng), HoTenKH (họ tên khách hàng)...

Ký hiệu: hình oval với tên gọi, có đường nối với thực thể



Hình 2.2. Thuộc tính của tập thực thể

Khi phân tích một thuộc tính, thông thường cần xét đến kiểu dữ liệu và miền giá trị tương ứng của thuộc tính đó. Chẳng hạn các thuộc tính Mã nhân viên (MANV), Họ tên (HoTen), Địa chỉ (DiaChi) là những chuỗi ký tự phản ánh những thông tin trên về nhân viên; Đơn giá là số nguyên dương,...

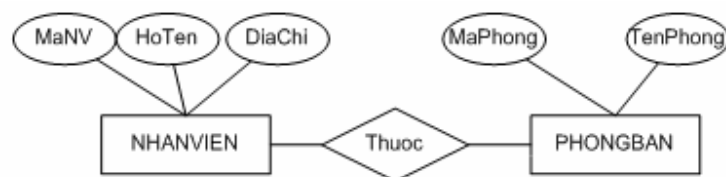
Có một số loại thuộc tính sau:

- **Thuộc tính đơn, kết hợp:** Thuộc tính đơn là thuộc tính không thể chia ra thành những phần con nhỏ hơn. Còn thuộc tính kết hợp thì có thể chia ra thành những phần con (tức là thành những thuộc tính khác). Ví dụ *tên nhân viên* có thể cấu trúc như là thuộc tính kết hợp bao gồm *tên*, *chữ lót*, và *họ*. Thuộc tính kết hợp có thể xuất hiện theo sự phân cấp. Ví dụ thuộc tính *địa chỉ nhân viên* có các thuộc tính thành phần là *số nhà*, *đường*, *thành phố*, *tỉnh*.
- **Thuộc tính rỗng.** Một giá trị rỗng (null value) được dùng đến khi một thực thể không có giá trị đối với một thuộc tính. Ví dụ một nhân viên nào đó không có người trong gia đình thì giá trị của thuộc tính *tên người trong gia đình* đối với nhân viên đó phải là rỗng. Giá trị rỗng cũng có thể được dùng để chỉ ra rằng giá trị của thuộc tính là chưa biết. Một giá trị chưa biết có thể hoặc là giá trị tồn tại nhưng chúng ta không có thông tin đó, hoặc là không biết được (không biết giá trị thực sự có tồn tại hay không).
- **Thuộc tính suy ra.** Giá trị của loại thuộc tính này có thể được suy ra từ các thuộc tính hoặc thực thể liên hệ khác. Ví dụ tập thực thể *khách hàng* có thuộc tính *số lượng tài khoản vay* cho biết bao nhiêu tài khoản vay mà khách hàng có từ ngân hàng. Chúng ta có thể suy ra giá trị của thuộc tính này bằng cách đếm số lượng các thực thể *tài khoản vay* được kết hợp với khách hàng. Một ví dụ khác về tập thực thể *nhân viên* có các thuộc tính *ngày bắt đầu*, cho biết ngày bắt đầu làm việc tại ngân hàng của nhân viên, và thuộc tính *thời gian làm việc*, cho biết tổng số giờ đã làm việc của nhân viên. Giá trị của *thời gian làm việc* có thể được suy ra từ giá trị của *ngày bắt đầu* và ngày hiện thời.

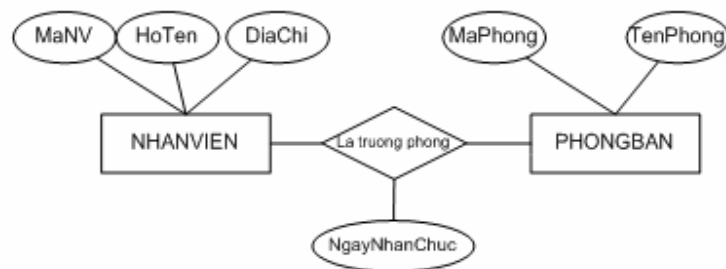
1.3. Môi kết hợp

Quan hệ giữa các tập thực thể được gọi là môi kết hợp. Mỗi môi kết hợp cũng được đặt tên thể hiện môi kết hợp, chẳng hạn như mỗi hóa đơn chỉ bán cho một khách hàng, một khách hàng có thể mua hàng nhiều lần (nhiều hóa đơn)

Ký hiệu là hình thoi nối với những thực thể tham gia vào mối kết hợp



Hình 2.3. a. Mối kết hợp không có thuộc tính



Hình 2.3.b. Mối kết hợp có thuộc tính

1.4. Bản số

1.4.1. Thể hiện của một thực thể

Trong tập thực thể KHACHHANG có nhiều khách hàng, ví dụ như {KH01, Nguyễn Trọng, Vũ, 11 Nguyễn Công Trứ, 063552540} là một khách hàng, đây được gọi là một thể hiện của thực thể KHACHHANG.

Một thể hiện của một tập thực thể được nhận biết bằng tập hợp tất cả các giá trị của tất cả các thuộc tính của thể hiện thuộc tập thực thể, đó chính là dữ liệu của đối tượng trong thế giới thực.

1.4.2. Thể hiện của một mối kết hợp

Xét mối kết hợp *bán* giữa HÓA ĐƠN và MẶT HÀNG, trong đó một thể hiện của quan hệ này sẽ tương ứng với việc một hóa đơn bán mặt hàng với số lượng và đơn giá là bao nhiêu.

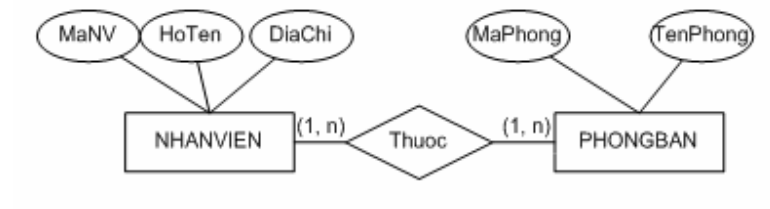
Như vậy, một thể hiện của một mối kết hợp là tập hợp các thể hiện của các tập thực thể tham gia vào mối kết hợp đó.

1.4.2.1. Bản số

Bản số của một tập thực thể đối với một mối kết hợp là cặp (bản số tối thiểu, bản số tối đa). Trong đó chúng được định nghĩa như sau:

- Bản số tối thiểu: bằng 0 hoặc 1, là số lần tối thiểu mà một thể hiện bất kỳ của một tập thực thể tham gia vào các thể hiện của mỗi kết hợp.
- Bản số tối đa: bằng 1 hoặc n, là số lần tối đa mà một thể hiện bất kỳ của một tập thực thể tham gia vào các thể hiện của mỗi kết hợp.

Ký hiệu:



Hình 2.4. Bản số

Ví dụ một nhân viên thuộc về một hoặc nhiều phòng ban, một phòng ban có thể có một hoặc nhiều nhân viên trực thuộc.

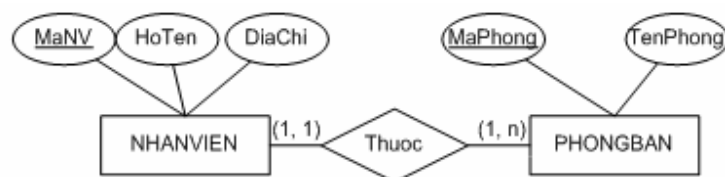
1.5. Khóa

1.5.1. Khóa của một tập thực thể

Khóa của một tập thực thể là một thuộc tính hoặc một số thuộc tính của thực thể, sao cho với mỗi giá trị của các thuộc tính này, tương ứng một và chỉ một thể hiện của tập thực thể (xác định một thực thể duy nhất). Trong nhiều trường hợp khóa của tập thực thể thường là thuộc tính chỉ định của tập thực thể đó.

Một thực thể có thể có nhiều khóa. Khi đó cần chọn ra một khóa để làm khóa chính.

Ký hiệu: Là thuộc tính được gạch dưới.



Hình 2.5. Khóa

1.5.2. Khóa của một mối kết hợp

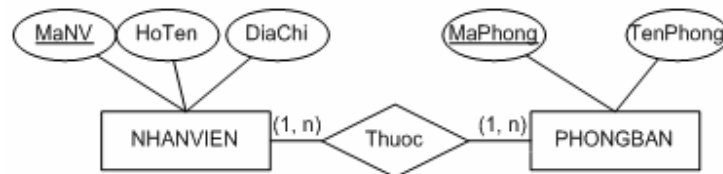
Khóa của một mối kết hợp nhận được bằng cách kết hợp khóa của các tập thực thể tham gia vào mối kết hợp đó. Tập hợp tất cả các giá trị của các thuộc tính khóa của một mối kết hợp xác định duy nhất một thể hiện của mối kết hợp đó.

Trong mô hình, khóa của mỗi kết hợp ngầm hiểu mà không được ghi ra nếu không quan tâm.

1.6. Số chiều (bậc) của một mối kết hợp, mối kết hợp tự thân (đệ quy)

1.6.1. Số chiều của một mối kết hợp

Số chiều của một mối kết hợp là số tập thực thể tham gia vào mối kết hợp đó.

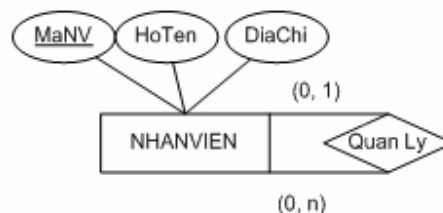


Hình 2.6. Mối kết hợp nhị phân

1.6.2. Mối kết hợp tự thân (đệ quy, vai trò)

Mối kết hợp tự thân là một mối kết hợp từ một tập thực thể đi đến chính tập thực thể đó.

Ví dụ một nhân viên có thể không chịu sự quản lý của ai hoặc một người, một người có thể không quản lý ai hoặc nhiều người.



Hình 2.7. Mối kết hợp đệ quy

1.7. Tổng quát hóa và chuyên biệt hóa

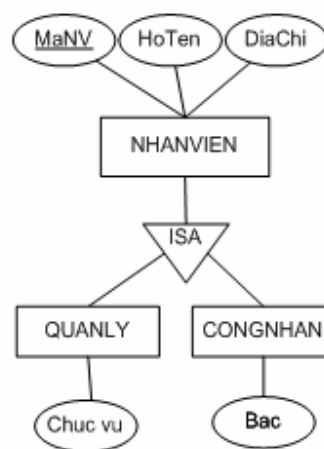
Mặc dù khái niệm bản số của tập thực thể đối với mối kết hợp cho chúng ta nhận biết mỗi thể hiện của tập thực thể tham gia tối thiểu là bao nhiêu, và tối đa là bao nhiêu vào mối kết hợp. Nhưng trong thực tế, một lớp các đối tượng trong tổ chức có khi tồn tại tình trạng là: một số đối tượng (tập con) của nó tham gia vào một mối kết hợp này, số còn lại có thể tham gia hoặc không vào những mối kết hợp khác, trong khi có thể tất cả các phần tử của chúng lại cùng tham gia vào mối kết hợp khác nữa. Hoặc một tập con này có những đặc tính này, còn những phần tử khác thì có thêm những đặc tính khác hoặc không. Chẳng hạn cùng là nhân viên trong nhà máy, thì công nhân trực tiếp tham gia sản xuất, nhân viên quản lý tham gia công tác quản lý. Đối với nhân viên quản lý người ta quan tâm đến chức vụ, còn đối với nhân viên công nhân thì người ta lại quan tâm đến bậc. Để phản ánh tình

trạng đó trong phương pháp mô hình hóa, người ta dùng khái niệm chuyên biệt hóa / tổng quát hóa.

Chuyên biệt hóa nghĩa là phân hoạch một thực thể thành các tập (thực thể) con.

Tổng quát hóa là gộp các tập thực thể thành một tập thực thể bao hàm tất cả các thể hiện của các tập thực thể con. Các chuyên biệt được thừa hưởng tất cả các thuộc tính của các tập thực thể mức trên và chính nó có thể có những thuộc tính khác. Các tập thực thể chuyên biệt có thể có những mối kết hợp khác nhau với những tập thực thể khác và do đó các xử lý sẽ có thể khác nhau tùy theo từng chuyên biệt thành phần.

Ký hiệu:



Hình 2.8. Chuyên biệt hóa / tổng quát hóa

1.8. Tập thực thể yếu

Một tập thực thể có thể không có đủ các thuộc tính để cấu thành một khóa chính, được gọi là *tập thực thể yếu*. Một tập thực thể mà nó có khóa chính được gọi là *tập thực thể mạnh*. Tập thực thể yếu phải tham gia vào mỗi quan hệ mà trong đó có một tập thực thể mạnh (tập thực thể mà tập thực thể yếu phụ thuộc)

Coi tập thực thể *thân nhân (THANNHAN)* có các thuộc tính là *tên thân nhân (TenTN)*, *ngày sinh (NTNS)*, *phái (Phai)*, *quan hệ với nhân viên (QuanHe)*. Như vậy tập thực thể này không có khóa chính nên nó là tập thực thể yếu.

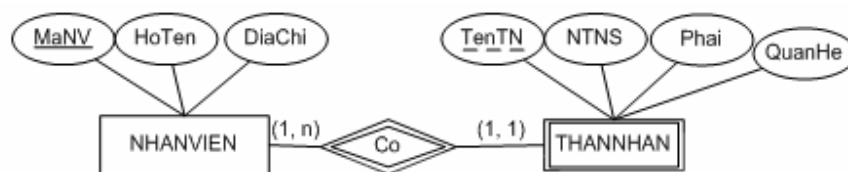
Mặc dù tập thực thể yếu không có khóa chính nhưng chúng ta cần một phương tiện để phân biệt trong số những thực thể thuộc tập thực thể này mà chúng phụ thuộc vào một thực thể mạnh nào đó.

- *Phân phân biệt* của một tập thực thể yếu là tập hợp các thuộc tính cho phép sự phân biệt nói trên được thực hiện. Ví dụ thuộc tính *tên thân nhân (TenTN)* là phân phân biệt của tập thực thể yếu *thân nhân (THANNHAN)*.

- Phân phân biệt của một tập thực thể yếu cũng còn được gọi là *khóa bộ phận* của tập thực thể yếu.
- Khóa chính của tập thực thể yếu được hình thành bởi khóa chính của tập thực thể mạnh mà tập thực thể yếu phụ thuộc vào, cộng với phân phân biệt của tập thực thể yếu. Ví dụ khóa chính của *THANNHAN* là $\{MaNV, TenTN\}$.

Kí hiệu:

Một tập thực thể yếu được chỉ ra bởi một hộp nét đôi trong lược đồ, và quan hệ định danh tương ứng được vẽ bởi hình thoi nét đôi. Trong hình, *phân phân biệt* được gạch dưới gián đoạn.



Hình 2.9. Tập thực thể yếu

2. Ví dụ

2.1. Quản lý đề án

Giả sử sau đây là một số yêu cầu dữ liệu đối với một công ty chuyên thực hiện các đề án:

Công ty gồm nhiều nhân viên, mỗi nhân viên được gán mã nhân viên để tiện việc quản lý, có họ tên, ngày sinh, mức lương được hưởng. Công ty gồm nhiều phòng ban, mỗi phòng ban có chức năng riêng của mình, có mã phòng, tên phòng, có một trưởng phòng. Mỗi nhân viên chỉ thuộc vào một phòng ban và một phòng có thể có nhiều nhân viên. Mỗi nhân viên trong phòng còn có thể chịu sự quản lý trực tiếp từ một nhân viên khác.

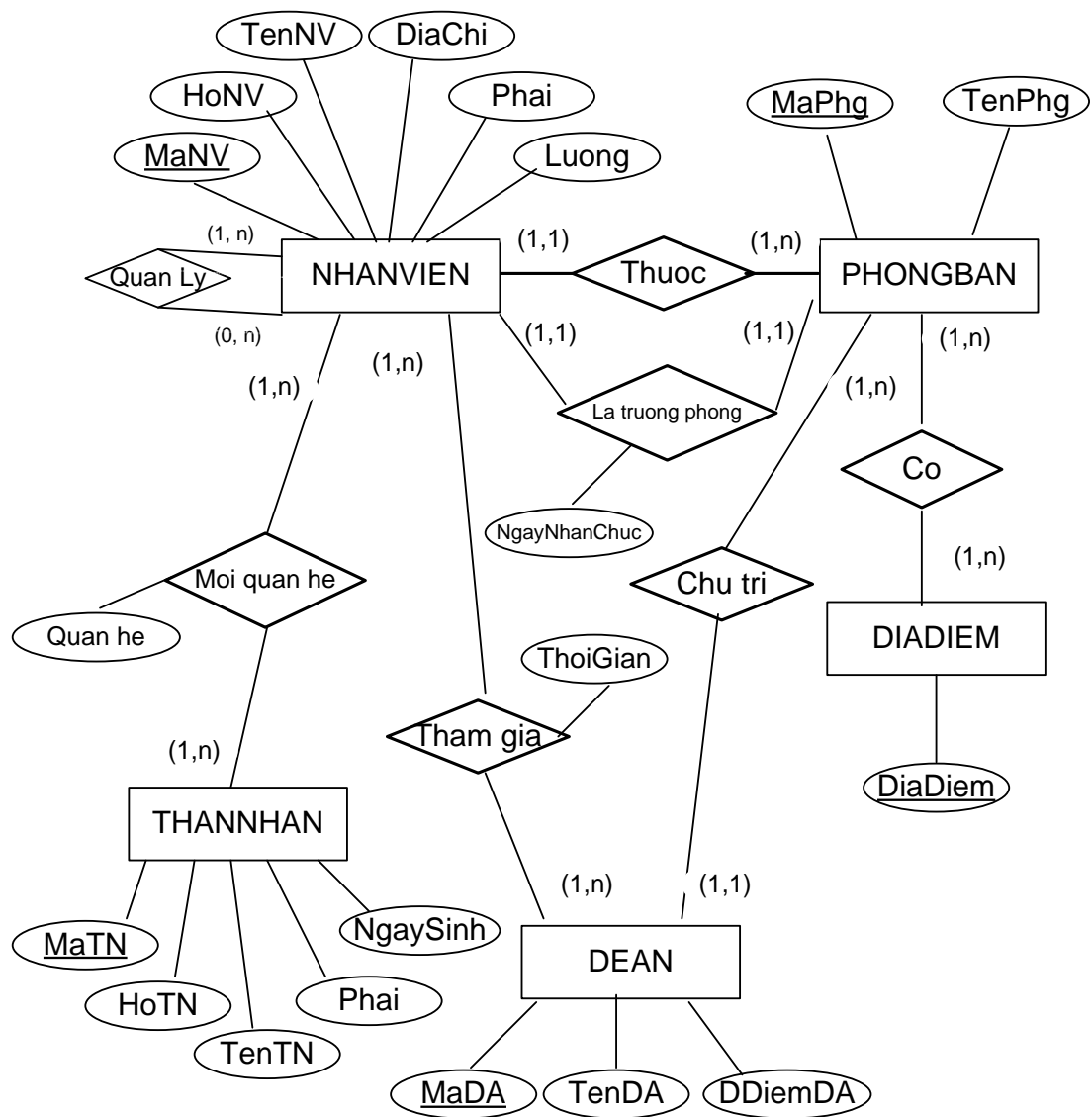
Do công ty thực hiện đề án, nên mỗi phòng có thể có nhiều văn phòng giao dịch hay làm việc khác nhau ở tại những địa điểm khác nhau.

Mỗi một đề án khi được xây dựng, có mã đề án, tên đề án, địa điểm thực hiện đề án đó và do một phòng ban chịu trách nhiệm chủ trì đề án.

Quá trình thực hiện đề án có thể được chia nhỏ thành nhiều công việc và phân công cho các nhân viên thực hiện, khi đó công ty sẽ ghi nhận lại thời gian phân công công việc cho nhân viên (tính bằng số giờ / tuần) để theo dõi tiến độ thực hiện.

Nhằm có thể chăm lo đời sống của nhân viên, công ty có ghi nhận lại những thông tin về những thân nhân của nhân viên, bao gồm những người như cha mẹ, chồng vợ, và con cái.

Từ mô tả tình huống trên, mô hình quan hệ thực thể như sau:



3. Bài tập

Hãy xây dựng mô hình thực thể kết hợp cho tình huống sau:

Giả sử sau đây là một số yêu cầu dữ liệu đối với ví dụ hoạt động ngân hàng:

Ngân hàng được tổ chức thành các chi nhánh. Mỗi chi nhánh tọa lạc tại một thành phố và được định danh bởi một tên duy nhất. Ngân hàng theo dõi tài sản của mỗi chi nhánh.

Các khách hàng của ngân hàng được định danh bởi số CMND của họ. Ngân hàng lưu trữ các tên khách hàng, đường phố và thành phố mà khách hàng sinh sống. Khách hàng có thể có tài khoản gửi và tài khoản vay. Một khách hàng có thể được kết hợp với một nhân viên ngân hàng. Nhân viên này có thể là nhân viên cho vay hoặc nhân viên bình thường.

Các nhân viên ngân hàng được định danh bởi mã số nhân viên của họ. Bộ phận quản lý ngân hàng lưu trữ tên và số phone của mỗi nhân viên, tên của các người phụ thuộc nhân viên và mã số nhân viên của người quản lý nhân viên. Ngân hàng cũng theo dõi ngày bắt đầu làm việc của nhân viên, và thời gian thuê nhân viên làm việc.

Ngân hàng đưa ra các loại tài khoản gửi, tài khoản tiết kiệm và tài khoản séc. Các tài khoản gửi có thể được nắm giữ bởi nhiều hơn một khách hàng, và một khách hàng có thể có nhiều hơn một tài khoản. Mỗi tài khoản gửi được gán bởi một số tài khoản duy nhất. Ngân hàng duy trì thông tin về cân đối của mỗi tài khoản gửi và ngày gần nhất mà tài khoản gửi được truy cập bởi mỗi khách hàng nắm giữ tài khoản đó. Ngoài ra, mỗi tài khoản tiết kiệm còn có mức lãi suất, và tài khoản séc có số tiền rút vượt mức.

Một tài khoản vay đầu tiên xuất phát từ một chi nhánh nào đó, và nó có thể được nắm giữ bởi một hoặc nhiều khách hàng. Mỗi tài khoản vay được gán bởi một số tài khoản duy nhất. Với mỗi tài khoản vay ngân hàng theo dõi số tiền vay và số tiền trả. Mặc dù số thứ tự lần trả tiền vay (gọi tắt là số lần trả) không xác định duy nhất lần trả đối với các tài khoản vay của ngân hàng nhưng nó xác định duy nhất lần trả đối với một tài khoản vay cụ thể. Ngày và số tiền trả đối với mỗi lần trả tiền vay cũng được ngân hàng theo dõi ghi nhận.

Mô Hình Dữ Liệu Quan Hệ

Mô hình dữ liệu quan hệ lần đầu tiên được đề nghị bởi Edgar F. Codd vào năm 1970. Hiện nay mô hình quan hệ là mô hình ưu thế đối với các ứng dụng xử lý dữ liệu thương mại. Chương này sẽ trình bày chi tiết về các khái niệm đã nhắc tới trong chương 1 và coi đó như là những cơ sở nền tảng để tiếp tục nghiên cứu các phần tiếp theo.

1. Các khái niệm cơ bản

1.1. Thuộc tính

Thuộc tính (attribute) là một tính chất riêng biệt của một đối tượng cần được lưu trữ trong CSDL để phục vụ cho việc khai thác dữ liệu về đối tượng.

Ví dụ:

- Đối tượng LOPHOC có các thuộc tính mã lớp, tên lớp, khóa, số học viên.
- Đối tượng SINHVIEN có các thuộc tính mã sinh viên, họ tên, ngày sinh, quê quán.

Các thuộc tính được đặc trưng bởi một tên thuộc tính, kiểu giá trị (data type) và miền giá trị (domain).

Trong các ứng dụng thực tế, người phân tích – thiết kế thường đặt *tên thuộc tính* một cách gợi nhớ, tuy nhiên không nên đặt tên quá dài (vì làm cho việc viết câu lệnh truy vấn vất vả hơn) nhưng cũng không nên quá ngắn (vì không thể hiện được ngữ nghĩa một cách rõ ràng).

Ví dụ: nếu có hai đối tượng HOCVIEN và GIAOVIEN đều có thuộc tính tên thì nên đặt tên một cách rõ ràng là Tên_học_viên và Tên_giáo_viên vì chúng mang ngữ nghĩa hoàn toàn khác nhau.

Mỗi một thuộc tính đều phải thuộc một *kiểu dữ liệu*. Kiểu dữ liệu có thể là vô hướng - là các kiểu dữ liệu cơ bản như chuỗi, số, logic, ngày tháng... hoặc các kiểu có cấu trúc được định nghĩa dựa trên các kiểu dữ liệu đã có sẵn.

Mỗi hệ quản trị CSDL có thể gọi tên các kiểu dữ liệu nói trên bằng các tên gọi khác nhau, ngoài ra còn bổ sung thêm một số kiểu dữ liệu riêng của mình. Ví dụ, Microsoft Access

có kiểu dữ liệu text, memo là kiểu chuỗi; SQL Server có kiểu dữ liệu text, char, varchar, nvarchar là kiểu chuỗi.

Mỗi một thuộc tính có thể chỉ chọn lấy những giá trị trong một tập hợp con của kiểu dữ liệu. Tập hợp các giá trị mà một thuộc tính A có thể nhận được gọi là *miền giá trị* của thuộc tính A, thường được ký hiệu là $MGT(A)$ hoặc $Dom(A)$.

Ví dụ:

Điểm của sinh viên là một số, nhưng luôn nằm trong đoạn từ 0 đến 10.

Với kiểu dữ liệu cấu trúc thì miền giá trị chính là tích đề các (hoặc tập con của tích đề các) của các miền giá trị thành phần.

Trong nhiều hệ quản trị CSDL, thường đưa thêm vào miền giá trị của các thuộc tính một giá trị đặc biệt gọi là giá trị null. Tùy theo ngữ cảnh mà giá trị này có thể là một giá trị không thể xác định được hay một giá trị chưa được xác định ở thời điểm nhập tin và có thể được xác định vào một thời điểm khác.

Nếu thuộc tính có kiểu dữ liệu vô hướng thì nó được gọi là thuộc tính đơn hoặc nguyên tố, nếu thuộc tính có kiểu dữ liệu có cấu trúc thì ta nói rằng nó là thuộc tính kép.

1.2. Quan hệ n ngôi

Một quan hệ R có n ngôi được định nghĩa trên tập các thuộc tính $U = \{A_1, A_2, \dots, A_n\}$ và kèm theo nó là một tân từ, để xác định mối quan hệ giữa các thuộc tính A_i , và được ký hiệu là $R(A_1, A_2, \dots, A_n)$. Tập thuộc tính của quan hệ R còn được ký hiệu là R^+ .

Với A_i là một thuộc tính có miền giá trị là $MGT(A_i)$, như vậy $R(A_1, A_2, \dots, A_n)$ là tập con của tích đề các: $MGT(A_1) \times MGT(A_2) \times \dots \times MGT(A_n)$

Quan hệ còn được gọi là bảng (table).

Ví dụ:

KHOA(Mã_khoa, Tên_khoa) là một quan hệ 2 ngôi với tân từ : “*Mỗi khoa có một mã khoa duy nhất để phân biệt với các khoa khác, có một tên gọi*”.

SINHVIEN (Mã_sinh_viên, Tên_sinh_viên, Ngày_sinh, Quê_quán, Khoa) là một quan hệ 5 ngôi với tân từ : “*Mỗi sinh viên có một mã sinh viên duy nhất để phân biệt với các sinh viên khác, có họ tên, ngày tháng năm sinh, quê quán và học tại một khoa trong trường*”.

1.3. Bộ

Một bộ (tuple) giá trị là các thông tin của một đối tượng thuộc quan hệ. Bộ giá trị cũng thường được gọi là một mẫu tin hay bản ghi (record), dòng của bảng (row).

Một bộ q là một vectơ gồm n thành phần thuộc tập hợp con của tích đề các miền giá trị của các thuộc tính và thỏa mãn tất cả các ràng buộc của quan hệ.

Ví dụ: các bộ giá trị dựa trên các thuộc tính của quan hệ SINHVIEN

$q_1 = (\text{SV001}, \text{Trần Văn Mạnh}, 10/10/1980, \text{Lâm Đồng}, \text{CTK27})$

$q_2 = (\text{SV002}, \text{Nguyễn Thị Hoa Huệ}, 25/11/1985, \text{Khánh Hòa}, \text{MTK27})$

$q_3 = (\text{SV003}, \text{Tăng Thanh Hà}, 11/11/1982, \text{Tp. Hồ Chí Minh}, \text{NVK27})$

Để lấy thành phần A_i – là giá trị thuộc tính A_i của một bộ giá trị q , ký hiệu $q.A_i$. Đây được gọi là *phép chiếu một bộ lên thuộc tính A_i*

Ví dụ:

$q_1.\text{Tên_sinh_viên} = \text{“Trần Văn Mạnh”}$

$q_2.\text{Khoa} = \text{“MTK27”}$

1.4. Lược đồ quan hệ

Lược đồ quan hệ (Relation schema) là sự trừu tượng hóa của quan hệ, một sự trừu tượng hóa ở mức cấu trúc của một bảng hai chiều. Khi nói đến lược đồ quan hệ tức là đề cập tới cấu trúc tổng quát của một quan hệ; khi nói đến một quan hệ thì hiểu rằng đó là một bảng có cấu trúc cụ thể trên một lược đồ quan hệ với các bộ giá trị của nó.

Lược đồ cơ sở dữ liệu là tập hợp các lược đồ quan hệ con $\{R_i\}$, ký hiệu là ζ .

Thể hiện (hay tình trạng) của quan hệ R , ký hiệu là T_R , là tập hợp các bộ giá trị của quan hệ R vào một thời điểm. Như vậy, tại những thời điểm khác nhau thì quan hệ có những thể hiện khác nhau.

Thể hiện của các lược đồ quan hệ con T_{R_i} gọi là tình trạng của lược đồ CSDL ζ

Ví dụ về thể hiện của quan hệ KHOA và LOPHOC

Bảng 3.1. Thể hiện của quan hệ KHOA

Mã khoa	Tên khoa	Ngày thành lập
CNTT	Công nghệ thông tin	10/10/1994
TH	Toán học	20/10/1976
VL	Vật lý	20/10/1976
HH	Hóa học	20/10/1976

Bảng 3.2. Thể hiện của quan hệ LOPHOC

Mã lớp	Tên lớp	Số học viên	Mã khoa
CTK27	Công nghệ thông tin K27	75	CNTT
HHK18	Hóa học K18	95	HH
THK20	Toán học K20	120	TH

1.5. Khóa của một quan hệ

Quan hệ R định nghĩa trên tập các thuộc tính $U = \{A_1, A_2, \dots, A_n\}$

Khi đó $K \subseteq U$ là khóa của quan hệ R nếu thỏa:

- (i) K xác định được giá trị của A_j , với mọi $j = 1, 2, \dots, n$
- (ii) Không tồn tại $K' \subseteq K$ mà K' có thể xác định được giá trị của A_j , với mọi $j = 1, 2, \dots, n$

Theo định nghĩa trên, K là tập con nhỏ nhất mà giá trị của nó có thể xác định được duy nhất một bộ giá trị của quan hệ.

Khóa theo định nghĩa trên gọi là *khóa đề nghị* (candidate key).

K được gọi là *siêu khóa* (superkey) của quan hệ R nếu $K' \subseteq K$ là một khóa của quan hệ. Như vậy một quan hệ Q luôn có ít nhất một siêu khóa và có thể có nhiều siêu khóa.

Ví dụ:

Với quan hệ LOPHOC (MaLop, TenLop, SoSV, MaKhoa)

Siêu khoá :

$$K1 = \{\text{MaLop}, \text{TenLop}\}$$

$$K2 = \{\text{MaLop}, \text{SoSV}\}$$

$$K3 = \{\text{MaLop}, \text{TenLop}, \text{MaKhoa}\}$$

$$K4 = \{\text{MaLop}, \text{SoSV}, \text{MaKhoa}\}$$

Ý nghĩa thực tế của khóa là dùng để nhận diện một bộ trong một quan hệ, khi cần thiết tìm thông tin một bộ q nào đó ta chỉ cần biết giá trị của khóa của q là đủ để dò tìm và hoàn toàn xác định được nó trong quan hệ.

Trong thực tế, đối với các loại thực thể tồn tại khách quan như NHANVIEN, SINHVIEN, MATHANG,... người thiết kế CSDL thường gán thêm một thuộc tính giả là mã số để làm khóa như MaNV, MSSV, MaHang,... Đối với các lược đồ quan hệ biểu diễn cho sự trừu tượng hóa thường có khóa chỉ định là một tổ hợp của hai hay nhiều thuộc tính.

Trong trường hợp lược đồ quan hệ có nhiều khóa đề nghị, khi cài đặt lên một hệ quản trị CSDL người ta chọn ra một khóa trong số các khóa đề nghị này để sử dụng. Khi đó khóa này được gọi là *khóa chính* (primary key) và các khóa còn lại là *khóa tương đương*. Khóa chính chỉ thật sự có ý nghĩa trong quá trình khai thác CSDL, khóa chính hoàn toàn không có vai trò gì khác so với các khóa chỉ định còn lại.

Trong các hệ quản trị CSDL có cài đặt cơ chế tự động kiểm tra tính duy nhất của khóa chính. Khi người sử dụng thêm một bộ mới q_2 có giá trị khóa chính trùng với giá trị khóa chính của một bộ q_1 đã có trong quan hệ thì hệ thống sẽ báo lỗi và yêu cầu nhập lại giá trị khác.

Các thuộc tính tham gia vào một khóa được gọi là thuộc tính khoá. Về mặt ký hiệu, trong lược đồ quan hệ các thuộc tính khóa được gạch dưới.

Trong một bộ của quan hệ, các thuộc tính khóa không chứa giá trị rỗng.

Không được phép sửa đổi giá trị của thuộc tính khoá, nếu người sử dụng muốn sửa giá trị thuộc tính khoá của bộ q , cần phải hủy bỏ bộ q sau đó thêm vào bộ q' với giá trị khóa đã được sửa đổi.

Ví dụ: Một lược đồ CSDL như sau:

KHOA (MaKhoa, TenKhoa, NgayThanhLap)

LOPHOC (MaLop, TenLop, NienKhoa, SoHocvien, MaKhoa)

MONHOC (MaMon, TenMon, SoTC)

HOCVIEN (MaHV, HoHV, TenHV, NgaySinh, QueQuan, MaLop)

GIAOVIEN (MaGV, HoGV, TenGV, NgaySinh, HocVi, ChuyenNganh)

KQUATHI (MaHV, MaMon, LanThi, NgayThi, DiemThi, GhiChu)

DAY (MaGV, MaLop, MaMon)

Với hai quan hệ R và S , một tập thuộc tính K của quan hệ R được gọi là *khóa ngoại* (foreign key) của quan hệ R nếu K là khóa của quan hệ S .

Ví dụ:

- Trong quan hệ LOPHOC, MaKhoa là khoá ngoại vì MaKhoa là khóa của quan hệ KHOA.
- Trong quan hệ HOCVIEN, MaLop là khoá ngoại vì MaLop là khóa của quan hệ LOPHOC.

1.6. Ràng buộc toàn vẹn

Ràng buộc toàn vẹn (RBTV) là một quy tắc định nghĩa trên một hay nhiều quan hệ do môi trường ứng dụng quy định. Đó chính là quy tắc để bảo đảm tính nhất quán của dữ liệu trong CSDL.

Thông thường mỗi RBTV được định nghĩa bằng một thuật toán trong CSDL.

Ví dụ:

- Trong quan hệ KQUATHI, DiemThi là một số nguyên nằm trong khoảng từ 0 đến 10.
- Trong quan hệ KQUATHI, LanThi là 1 hoặc 2.

2. Các thao tác cơ bản trên quan hệ

Các thao tác cơ bản trên một quan hệ là thêm (insert), xóa (delete), sửa (update) các bộ giá trị của quan hệ.

2.1. Phép thêm

Việc thêm một bộ mới t vào quan hệ $R(A_1, A_2, \dots, A_n)$ làm cho thể hiện T_R tăng thêm một phần tử mới $T_R = R \cup t$

Dạng thức của phép thêm bộ mới là:

$$INSERT(R; A_1 = v_1, A_2 = v_2, \dots, A_n = v_n),$$

với: A_1, A_2, \dots, A_n là các thuộc tính và v_1, v_2, \dots, v_n là giá trị muốn thêm vào (với điều kiện là các giá trị này thuộc $MGT(A_1), MGT(A_2), \dots, MGT(A_n)$ tương ứng)

Trong trường hợp này nếu như các thuộc tính không được liệt kê trong danh sách gán giá trị của bộ t trong câu lệnh INSERT sẽ nhận giá trị null.

Nếu xem thứ tự của các thuộc tính là cố định và các giá trị v_1, v_2, \dots, v_n là hoàn toàn tương ứng thì phép chèn có thể viết dưới dạng tường minh như sau:

$$INSERT(R; v_1, v_2, \dots, v_n)$$

Chú ý rằng có thể phép chèn không được thực hiện hoặc làm mất tính nhất quán của dữ liệu vì các lý do sau:

- Giá trị khóa của bộ mới là null hoặc trùng với giá trị khóa của một bộ đã có trong CSDL. Hệ quản trị CSDL sẽ không cho thêm mới trong trường hợp này.
- Bộ mới không phù hợp với lược đồ quan hệ. Trường hợp này xảy ra khi người sử dụng thêm mới các giá trị sai thứ tự, sai kiểu hoặc độ lớn của các thuộc tính trong lược đồ. Hệ quản trị CSDL có thể sẽ không cho bổ sung nếu không tương thích kiểu giá trị, hoặc vẫn cho bổ sung bộ mới nhưng tính nhất quán của dữ liệu không được đảm bảo.
- Một số giá trị của bộ mới không thuộc miền giá trị của thuộc tính tương ứng. Trong trường hợp này, nếu quan hệ đã được đảm bảo tính nhất quán bởi các RBTV về miền giá trị thì hệ quản trị CSDL sẽ không cho bổ sung; ngược lại, nếu không có RBTV về miền giá trị thì tính nhất quán của CSDL bị vi phạm mà hệ quản trị CSDL không phát hiện được.

2.2. Phép xóa

Phép xóa bộ t của quan hệ sẽ xóa đi một (hoặc nhiều) bộ t khỏi thể hiện của quan hệ $T_R = R \setminus t$

Dạng thức của phép xóa là:

$$DELETE(R; A_1 = v_1, A_2 = v_2, \dots, A_n = v_n),$$

với $A_i = v_j; j = 1..n$ chính là điều kiện thỏa một số thuộc tính của bộ t để loại một bộ t ra khỏi quan hệ.

Ví dụ: Với quan hệ:

HOCVIEN (MaHV, HoHV, TenHV, NgaySinh, QueQuan, Khoa),

và phép loại bỏ : $DELETE(HOCVIEN; QueQuan = "NhaTrang")$

thì tất cả các bộ trong thể hiện HOCVIEN có quê quán ở Nha trang sẽ bị loại bỏ.

2.3. Phép sửa

Dạng thức của phép sửa là:

$$UPDATE(R; A_1 = c_1, A_2 = c_2, \dots, A_n = c_n; A_1 = v_1, A_2 = v_2, \dots, A_n = v_n),$$

với $A_i = c_j; j = 1..n$ là điều kiện thỏa tìm kiếm bộ muốn sửa và $A_i = v_j; j = 1..n$ là giá trị mới cần cập nhật

Ví dụ: Với quan hệ:

HOCVIEN (MaHV, HoHV, TenHV, NgaySinh, QueQuan, MaLop),

Trong thể hiện của HOCVIEN có bộ:

$q = \{ \text{"HV001"}, \text{"Nguyễn Văn"}, \text{"Mạnh"}, 20/10/85, \text{"Nghệ An"}, \text{"CNTT"} \}$

và phép cập nhật: $UPDATE(HOCVIEN; MaHV = HV001; QueQuan = \text{Lâm Đồng})$

khi đó kết quả đạt được sẽ là bộ q được sửa lại với giá trị :

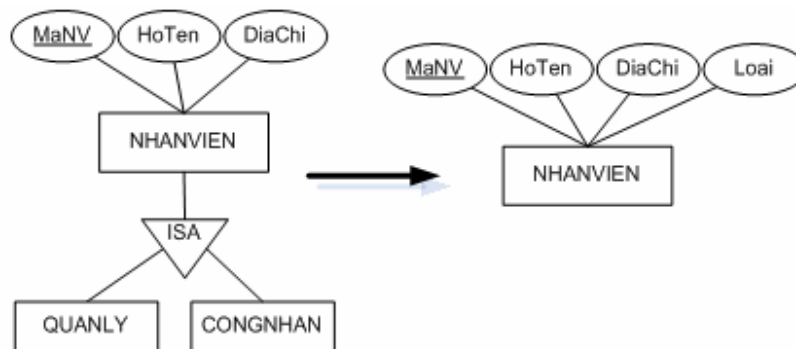
$q = \{ \text{"HV001"}, \text{"Nguyễn Văn"}, \text{"Mạnh"}, 20/10/85, \text{"Lâm Đồng"}, \text{"CNTT"} \}$

3. Các bước chuyển đổi từ mô hình thực thể kết hợp sang mô hình quan hệ

3.1. Biến các tập thực thể chuyên biệt hóa về dạng bình thường

3.1.1. Tập thực thể chuyên biệt không có thuộc tính riêng

Trong trường hợp này, đưa tập thực thể chuyên biệt lên làm thuộc tính của tập thực thể mức tổng quát, khi đó có thuộc tính mới cho biết loại của thực thể chuyên biệt.

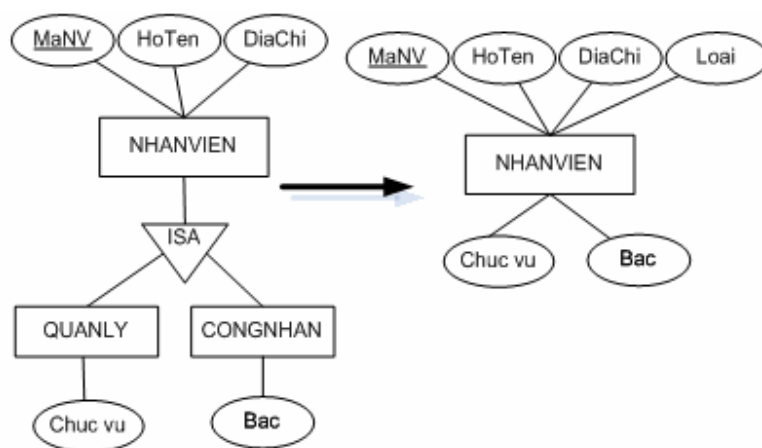


Hình 3.1. Biến đổi tập thực thể chuyên biệt không có thuộc tính riêng

Trong ví dụ trên $Loai = \{ NV_QL, NV_CN \}$

3.1.2. Tập thực thể chuyên biệt có thuộc tính riêng

Nếu số lượng thuộc tính riêng ở tập thực thể chuyên biệt ít, gom lên mức tổng quát, và bổ sung thêm ràng buộc.



Hình 3.2. Biến đổi tập thực thể chuyên biệt có thuộc tính riêng

Trong ví dụ trên $Loai = \{NV_QL, NV_CN\}$ và các ràng buộc:

- RBTV1: “Nếu $Loai = NV_QL$ thì thuộc tính *Chucvu* mới có giá trị”
- RBTV2: “Nếu $Loai = NV_CN$ thì thuộc tính *Bac* mới có giá trị”

3.2. Chuyển tất cả các tập thực thể thành quan hệ

Tập thực thể mạnh

Với mỗi tập thực thể mạnh, chuyển thành quan hệ với khóa là khóa của tập thực thể.

NHANVIEN (MaNV, HoTen, DiaChi)

PHONGBAN (MaPhong, TenPhong)

Tập thực thể yếu

Với mỗi tập thực thể yếu, chuyển thành quan hệ với:

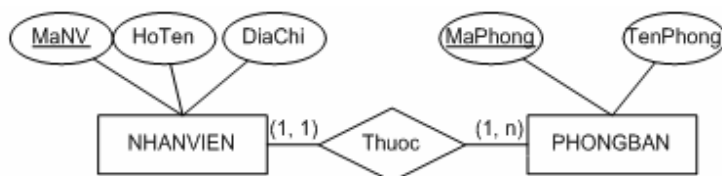
- Các thuộc tính là các thuộc tính của tập thực thể yếu và khóa của tập thực thể mạnh mà nó phụ thuộc
- Khóa của quan hệ là khóa của tập thực thể mạnh và phần phân biệt của tập thực thể yếu.

THANNHAN (MaNV, TenTN, NTNS, Phai, QuanHe)

3.3. Mỗi kết hợp

Mỗi kết hợp có bản số (1,1) và (1, n), chuyển thành các quan hệ với:

- Tập thực thể có sự tham gia (1, n) chuyển như bình thường
- Tập thực thể có sự tham gia (1, 1) chuyển thành quan hệ gồm *tất cả các thuộc tính của thực thể và thuộc tính khóa của thực thể có sự tham gia (1, n)*.

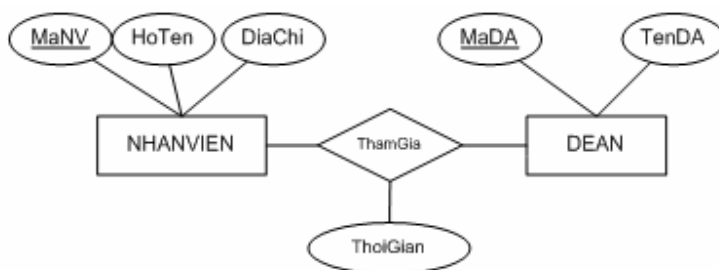


NHANVIEN (MaNV, HoTen, DiaChi, MaPhong)

PHONGBAN (MaPhong, TenPhong)

Mỗi kết hợp khác, chuyển thành một quan hệ với:

- Thuộc tính là khác khóa của các tập thực thể tham gia vào mỗi kết hợp và mọi thuộc tính của mỗi kết hợp
- Khóa là khóa của các tập thực thể tham gia vào mỗi kết hợp và có thể có thêm thuộc tính của mỗi kết hợp



THAMGIA (MaNV, MaDA, ThoiGian): *Khóa là khóa của các tập thực thể tham gia vào mỗi kết hợp*

3.4. Nhập tất cả các quan hệ có cùng khóa

Sau khi thực hiện chuyển đổi, có thể có một số quan hệ có cùng khóa, khi đó thực hiện việc nhập tất cả các quan hệ có cùng khóa lại thành một quan hệ.

Chương 4

Ngôn Ngữ Đại Số Quan Hệ

Ngôn ngữ đại số quan hệ là ngôn ngữ phi thủ tục. Nó bao gồm tập hợp các phép toán được áp dụng trên các thể hiện của quan hệ, kết quả của một câu truy vấn là một thể hiện của quan hệ. Ngôn ngữ đại số quan hệ có ưu điểm trong việc thể hiện kế hoạch thực hiện câu truy vấn và các kỹ thuật tối ưu hóa câu truy vấn.

1. Các phép toán cơ sở

1.1. Các phép toán tập hợp

Các phép toán cơ bản được áp dụng trên tập các bộ giá trị của các quan hệ, được hình thành từ lý thuyết tập hợp toán học: hợp hay hội (union), hiệu (minus), giao (intersection), tích đề các (cartesian product operation), chia (division), bù (complement). Quan hệ kết quả của các phép toán hợp, hiệu, trừ có cùng tên thuộc tính với quan hệ đầu tiên (quy ước).

Giả thiết: $R(A_1, A_2, \dots, A_n)$ và $S(B_1, B_2, \dots, B_n)$

Tính khả hợp

Quan hệ $R(A_1, A_2, \dots, A_n)$ và $S(B_1, B_2, \dots, B_n)$ được gọi là khả hợp khi:

- Số bậc của R và S là bằng nhau, nghĩa là cùng số lượng thuộc tính.
- Miền giá trị của thuộc tính phải tương thích $MGT(A_i) = MGT(B_i)$, với $1 \leq i \leq n$

1.1.1. Phép hợp (union)

Hợp (hay còn gọi là hội) của hai quan hệ khả hợp R và S , ký hiệu $R \cup S$, là quan hệ Q được định nghĩa như sau:

$$Q = R \cup S = \{t \mid t \in R \vee t \in S\}$$

Ví dụ:

A	B
A1	b1
a1	b2
a2	b1

R

A	B
a1	b2
a2	b3

S

A	B
a1	b1
a1	b2
a2	b1
a2	b3

 $R \cup S$

Nói cách khác, hợp của hai quan hệ R và S là một quan hệ có cùng ngôi với quan hệ R và S, với các bộ giá trị bằng gộp các bộ của cả R và S, những bộ giá trị trùng nhau chỉ giữ lại 1 bộ. Trong ví dụ trên bộ {a1, b2} xuất hiện trong cả R và S, do đó chỉ xuất hiện 1 lần trong $R \cup S$

1.1.2. Phép trừ (minus)

Phép trừ (hay còn gọi là hiệu) của hai quan hệ khả hợp R và S, ký hiệu $R - S$, là quan hệ Q được định nghĩa như sau:

$$Q = R - S = \{t \mid t \in R \wedge t \notin S\}$$

Ví dụ:

A	B
a1	b1
a1	b2
a2	b1

R

A	B
a1	b2
a2	b3

S

A	B
a1	b1
a2	b1

 $R - S$

Nói cách khác, hợp của hai quan hệ R và S là một quan hệ có cùng ngôi với quan hệ R và S, với các bộ giá trị là các bộ của cả R sau khi đã loại bỏ các bộ có mặt trong S.

1.1.3. Phép giao (intersect)

Giao của hai quan hệ khả hợp R và S, ký hiệu $R \cap S$, là quan hệ Q được định nghĩa như sau:

$$Q = R \cap S = \{t \mid t \in R \wedge t \in S\}$$

Ví dụ:

A	B
a1	b1
a1	b2
a2	b1

R

A	B
a1	b2
a2	b3

S

A	B
a1	b2

$R \cap S$

Nói cách khác, giao của hai quan hệ R và S là một quan hệ có cùng ngôi với quan hệ R và S, với các bộ giá trị là các bộ giống nhau trong cả R và S.

1.1.4. Phép tích đề các (Cartesian Product Operation)

Giả thiết: $R(A_1, A_2, \dots, A_n)$ và $S(B_1, B_2, \dots, B_m)$

Phép tích đề các của hai quan hệ của hai quan hệ R và S, ký hiệu $R \times S$, là quan hệ Q được định nghĩa như sau:

$$Q = R \times S = \{tq \mid t \in R \wedge q \in S\}$$

Ví dụ:

A	B
a1	b1
a1	b2
a2	b1

C	D
c1	d2
c2	d3

S

A	B	C	D
a1	b1	c1	d2
a1	b1	c2	d3
a1	b2	c1	d2
a1	b2	c2	d3
a2	b1	c1	d2
a2	b1	c2	d3

$R \times S$

Vậy tích đề các của hai quan hệ R và S là một quan hệ gồm (n+m) ngôi với n thuộc tính đầu là một bộ của R và m thuộc tính sau là một bộ thuộc S.

1.1.5. Phép chia (division)

Giả thiết: $R(A_1, A_2, \dots, A_n)$ và $S(B_1, B_2, \dots, B_m)$ ($n > m, S \neq \emptyset$), có m thuộc tính chung. Khi đó phép chia trên 2 quan hệ R và S , ký hiệu $R \div S$, là quan hệ Q có $(n-m)$ ngôi được định nghĩa như sau:

$$Q = R \div S = \{t \mid \forall u \in S, (t, u) \in R\}$$

Ví dụ 1:

A	B	C	D
a1	b1	c1	d1
a1	b1	c2	d3
a2	b1	c2	d3
a2	b1	c1	d1
a3	b1	c1	d3

R

C	D
C1	d1
C2	d3

S

A	B
a1	b1
a2	b1

$R \div S$

Nói cách khác, phép chia của quan hệ R cho S đưa ra tất cả các bộ trong quan hệ R , sao cho khớp với tất cả các bộ trong quan hệ S

1.2. Các phép toán quan hệ

Phần này trình bày các phép toán trên quan hệ và ví dụ minh họa dựa trên lược đồ cơ sở dữ liệu Quản lý đề án:

NHANVIEN (MaNV, HoNV, tenNV, NgaySinh, DiaChi, Phai, Luong, MaNQL, Phong)

Tân từ: Mỗi nhân viên có Mã nhân viên (MaNV) duy nhất để phân biệt với các nhân viên khác, có họ tên (HoNV, TenNV), ngày sinh (NgaySinh), địa chỉ (DiaChi), phái Nam hoặc Nữ (Phai), mức lương (Luong), người quản lý trực tiếp (MaNQL) và thuộc về một phòng ban (Phong)

PHONGBAN (MaPhong, TenPhong, TruongPhong, NgayNhanChuc)

Tân từ: Mỗi một phòng ban có một mã phòng duy nhất (MaPhong) để phân biệt với các phòng ban khác, có tên phòng (TenPhong), người trưởng phòng (TruongPhong), và ngày nhận chức của trưởng phòng (NgayNhanChuc)

DIADIEMPHONG (MaPhong, DiaDiem)

Tân từ: Mỗi một phòng ban (MaPhong) có thể có nhiều địa điểm làm việc khác nhau (DiaDiem)

DEAN (MaDA, TenDA, DdiemDA, Phong)

Tân từ: Mỗi một đề án có một mã đề án duy nhất (MaDA) để phân biệt với các đề án khác, có tên đề án (TenDA), địa điểm thực hiện (DdiemDA), và do một phòng ban chủ trì đề án đó (Phong)

PHANCONG (MaNV, MaDA, ThoiGian)

Tân từ: Mỗi một nhân viên (MaNV) được phân công tham gia đề án (MaDA) dưới dạng tham gia số giờ trên 1 tuần (ThoiGian)

THANNHAN(MaTN, HoTN, TenTN, Phai, NgaySinh)

Tân từ: Mỗi thân nhân có Mã thân nhân (MaTN) duy nhất để phân biệt với các thân nhân khác, có họ tên (HoTN, TenTN), phái (Phai) ngày sinh (NgaySinh)

NVIEN_TNHAN(MaNV, MaTN, QuanHe)

Tân từ: Mỗi nhân viên (MaNV) có thể có nhiều thân nhân (MaTN), được diễn giải bởi quan hệ (QuanHe) như vợ, chồng, con, anh em...

1.2.1. Phép chọn (selection)

Cho phép chọn ra những bộ trong R thỏa mãn biểu thức điều kiện chọn P cho trước. Ký hiệu là $\sigma_P(R)$ với định nghĩa:

$$\sigma_P(R) = \{t \mid t \in R \wedge P(t)\}$$

với R là quan hệ được chọn, P là biểu thức logic chứa các phép so sánh ($\neq, \geq, \leq, =, \dots$), các phép toán logic (\wedge, \vee, \neg) dạng:

<Thuộc tính> <phép so sánh> <Thuộc tính> hay <Hằng số>

Như vậy kết quả của phép chọn là một quan hệ có cùng danh sách thuộc tính với quan hệ R.

Ví dụ:

- Chọn những nhân viên có lương ≥ 500000

$$\sigma_{Luong \geq 500000}(NHANVIEN)$$

- Cho biết những nhân viên thuộc phòng số 5 và có lương ≥ 500000

$$\sigma_{Luong \geq 500000 \wedge Phong = 5}(NHANVIEN)$$

1.2.2. Phép chiếu (projection)

Cho phép trích chọn ra những cột (thuộc tính) trong R chỉ ra trong danh sách thuộc tính. Ký hiệu là $\pi_{A_1, A_2, \dots, A_k}(R)$ với A_1, A_2, \dots, A_k là danh sách các thuộc tính cần chọn và R là quan hệ cần trích chọn.

Nhận thấy rằng số lượng các bộ kết quả luôn nhỏ hơn hoặc bằng số lượng các bộ trong R. Các bộ trùng nhau sẽ loại đi và chỉ giữ lại bộ.

Ví dụ:

- Cho biết mã nhân viên, họ tên của tất cả các nhân viên

$$\pi_{MaNV, HoNV, TenNV}(NHANVIEN)$$

- Cho biết mã nhân viên, họ tên, phòng làm việc và mức lương của tất cả các nhân viên

$$\pi_{MaNV, HoNV, TenNV, Phong, Luong}(NHANVIEN)$$

- Cho biết các đề án cùng với các phòng phụ trách đề án đó

$$\pi_{MaDA, TenDA, Phong}(DEAN)$$

1.2.3. Phép gán (assignment)

Khi gặp những truy vấn phức tạp, phép gán cho phép diễn tả một cách rõ ràng hơn câu truy vấn. Khi đó, câu truy vấn chính là một chuỗi các phép gán theo sau đó là một biểu thức có giá trị như là kết quả của câu truy vấn.

Ký hiệu: \leftarrow

Việc gán được thực hiện cho một biến quan hệ tạm và được sử dụng cho các biểu thức theo sau.

1.2.4. Phép đổi tên:

Vì cho phép đặt tên nên có thể tham chiếu tới kết quả của biểu thức đại số quan hệ, và cho phép tham chiếu tới một quan hệ bằng nhiều tên

Các phép đổi tên sau:

- Đổi tên quan hệ và tên thuộc tính: Cho biểu thức đại số quan hệ E có n thuộc tính, biểu thức $\rho_{\lambda(A_1, A_2, \dots, A_n)}(E)$ trả về kết quả biểu thức E dưới tên λ và các tên của thuộc tính đổi thành A_1, A_2, \dots, A_n
- Đổi tên quan hệ: Cho biểu thức đại số quan hệ E, biểu thức $\rho_{\lambda}(E)$ trả về kết quả biểu thức E dưới tên λ
- Đổi tên thuộc tính: Cho biểu thức đại số quan hệ E có n thuộc tính, biểu thức $\rho_{(A_1, A_2, \dots, A_n)}(E)$ trả về kết quả biểu thức E với các tên của thuộc tính đổi thành A_1, A_2, \dots, A_n

1.2.5. Chuỗi các phép toán

Kết hợp các phép toán đại số quan hệ với nhau để tạo ra một quan hệ kết quả theo yêu cầu.

Có nhiều cách để thể hiện một truy vấn cho trước.

Ví dụ: Cho biết mã nhân viên, họ tên cùng lương của nhân viên làm việc trong phòng số 4

- Cách 1: $\pi_{MaNV, HoNV, TenNV, Luong}(\sigma_{Phong=4}(NHANVIEN))$
- Cách 2:

$$NVIEN_P4 \leftarrow \sigma_{Phong=4}(NHANVIEN)$$

$$KQ \leftarrow \pi_{MaNV, HoNV, TenNV, Luong}(NVIEN_P4)$$

- Cách 3:

$$NVIEN_P4 \leftarrow \sigma_{Phong=4}(NHANVIEN)$$

$$KQ(MaNV, HoNV, TenNV, LuongNV) \leftarrow \pi_{MaNV, HoNV, TenNV, Luong}(NVIEN_P4)$$

Trong đó có sử dụng phép gán và phép đặt lại tên.

2. Các phép toán khác

2.1. Phép kết hai quan hệ

Giả thiết: $R(A_1, A_2, \dots, A_n)$ và $S(B_1, B_2, \dots, B_m)$, việc ghép bộ $t = (a_1, a_2, \dots, a_n) \in R$ vào $v = (b_1, b_2, \dots, b_m) \in S$ được định nghĩa như sau:

$A \in R^+$ và $B \in S^+$ là hai thuộc tính có thể so sánh được.

Gọi θ là một trong các phép so sánh $\{<, >, =, \geq, \leq, \neq\}$

Khi đó, phép kết nối hai quan hệ R và S trên các thuộc tính A, B với phép so sánh θ được định nghĩa:

$$R \bowtie_{A\theta B} S = \{v = (t, u) \mid t \in R, u \in S, t.A \theta u.B\}$$

Có thể nhận thấy rằng phép kết nối được thực hiện qua 2 bước: (1) tích đề các hai quan hệ R và S, (2) chọn các bộ thỏa điều kiện $A\theta B$.

Nếu θ là phép toán so sánh bằng nhau được gọi là phép *kết nối bằng* (equi join). Nếu các thuộc tính so sánh là giống tên nhau thì trong kết quả của phép nối sẽ loại đi một cột (thuộc tính), khi đó phép kết được gọi là phép *kết tự nhiên* (natural join) và về mặt ký hiệu bỏ đi $A\theta B$. Các trường hợp còn lại được gọi là phép kết nối *theta* (θ join)

Ví dụ:

A	B	C			
a1	1	1			
a2	2	1			
a3	2	2			

R

C	D	E			
1	d1	e1			
2	d2	e2			
3	d3	e3			

S

A	B	C	C	D	E
a1	1	1	1	d1	E1
a2	2	1	1	d1	E1
A2	2	1	2	D2	E2
A3	2	2	1	d1	E1
a3	2	2	2	d2	E2

$R \bowtie_{R.B \geq S.C} S$

Và kết quả của phép nối tự nhiên (kết bằng trên thuộc tính C) là

A	B	C	D	E
a1	1	1	d1	e1
a2	2	1	d1	e1
a3	2	2	d2	e2

2.2. Phép kết nối nội (inner join)

Thực chất của phép kết nối nội là phép kết nối bằng đã nêu trên. Tuy nhiên, ngay cả khi hai thuộc tính có cùng tên thì kết quả vẫn giữ lại 2 tên thuộc tính đó.

Ví dụ:

A	B	C	A	D	E	
a1	B 1	c1	a1	d1	e1	
a2	B 2	c1	a2	d2	e2	
a3	B 3	c2	a4	d4	e4	
a5	B 5	c5	a6	d6	e6	
a7	B 7	c7	a7	d7	e7	

A	B	C
a1	B 1	c1
a2	B 2	c1
a3	B 3	c2
a5	B 5	c5
a7	B 7	c7

R

A	D	E
a1	d1	e1
a2	d2	e2
a4	d4	e4
a6	d6	e6
a7	d7	e7

S

$$R \underset{A}{\overset{R.A=S.}{\bowtie}} S$$

2.3. Phép kết nối trái (left join)

Phép kết nối trái hai quan hệ R và S trên các thuộc tính A và B với phép so sánh bằng được định nghĩa là tất cả các bộ v đạt được bằng cách xếp bộ giá trị của R và S cạnh nhau, nếu có giá trị giống nhau trên hai thuộc tính kết nối; và các bộ v đạt được nhờ cách đặt bộ R với các bộ null của S, nếu không tìm được giá trị tương ứng của thuộc tính kết nối trên quan hệ S.

$$R.A=S.B$$

$$R \ltimes S = \{v = (t, u) \mid (t \in R, u \in S : t.A = u.B) \vee (t \in R, u = u_{null} : t.A \notin S[B])\}$$

Với $S[B]$ là tập tất cả các giá trị của thuộc tính B của S

Ví dụ:

A	B	C
a1	b1	c1
a2	b2	c1
a3	b3	c2
a5	b5	c5
a7	b7	c7

R

A	D	E
a1	d1	e1
a2	d2	e2
a4	d4	e4
a6	d6	e6
a7	d7	e7

S

A	B	C	A	D	E
a1	b1	c1	a1	d1	e1
a2	b2	c1	a2	d2	e2
a3	b3	c2	null	null	null
a5	b5	c5	null	null	null
a7	b7	c7	a7	d7	e7

$R \bowtie_{R.A=S.A} S$



Ý nghĩa của phép kết nối này là xác định được các bộ giá trị của quan hệ bên trái nhưng không có bộ giá trị tương ứng trong quan hệ bên phải.

2.4. Phép kết nối phải (right join)

Phép kết nối phải hai quan hệ R và S trên các thuộc tính A và B với phép so sánh bằng được định nghĩa là tất cả các bộ v đạt được bằng cách xếp bộ giá trị của R và S cạnh nhau, nếu có giá trị giống nhau trên hai thuộc tính kết nối; và các bộ v đạt được nhờ cách đặt bộ null của R với các bộ của S, nếu không tìm được giá trị tương ứng của thuộc tính kết nối trên quan hệ R.

$R.A=S.B$

$$R \bowtie S = \{v = (t, u) \mid (t \in R, u \in S : t.A \theta u.B) \vee (t = t_{null}, u \in S, : u.B \notin R[A])\}$$

Với $R[A]$ là tập tất cả các giá trị của thuộc tính A của R

Ví dụ:

A	B	C
a1	b1	c1
a2	b2	c1
a3	b3	c2
a5	b5	c5
a7	b7	c7

R

A	D	E
a1	d1	e1
a2	d2	e2
a4	d4	e4
a6	d6	e6
a7	d7	e7

S

A	B	C	A	D	E
a1	b1	c1	a1	d1	e1
a2	b2	c1	a2	d2	e2
null	null	Null	a4	d4	e4
null	null	Null	a6	d6	e6
a7	b7	c7	a7	d7	e7

$R \bowtie_{R.A=S.A} S$



Ý nghĩa của phép kết nối này là xác định được các bộ giá trị của quan hệ bên phải nhưng không có bộ giá trị tương ứng trong quan hệ bên trái.

Chú ý rằng trong một số ngôn ngữ truy vấn CSDL, người ta gọi hai phép kết nối trái và phải chung lại là phép *kết nối ngoài* (outer join) vì cho phép giữ lại tất cả các bộ của hai quan hệ không tìm được bộ giá trị giống nhau trên các thuộc tính kết nối.

2.5. Hàm kết hợp và gom nhóm

Dùng để tính toán các giá trị mang tính chất tổng hợp trong đại số quan hệ. Trong đó:

Hàm kết hợp: đầu vào là một tập giá trị và trả về một giá trị đơn

- Avg(): giá trị trung bình
- Min(): giá trị nhỏ nhất
- Max(): giá trị lớn nhất
- Sum(): tính tổng
- Count(): đếm số mẫu tin

Gom nhóm: công thức như sau: $\mathfrak{F}_{G_1, G_2, \dots, G_N, F_1(A_1), F_2(A_2), \dots, F_n(A_n)}(E)$, với:

- E là biểu thức đại số quan hệ
- G_i là tên thuộc tính gom nhóm (có thể không có)
- F_i là hàm gom nhóm
- A_i là tên thuộc tính tính toán trong hàm gom nhóm

Ví dụ:

- Cho biết số nhân viên trong công ty và mức lương trung bình

$$\mathfrak{F}_{COUNT(MaNV), AVG(Luong)}(NHANVIEN)$$

- Cho biết số lượng nhân viên và lương trung bình của mỗi phòng ban

$$Phong \mathfrak{F}_{COUNT(MaNV), AVG(Luong)}(NHANVIEN)$$

2.6. Các phép toán cập nhật trên quan hệ

Các thao tác được viết thông qua phép toán gán. Cụ thể như sau:

2.6.1. Thêm

Phép thêm: $r \leftarrow r \cup E$, với r là một quan hệ và E là một biểu thức đại số quan hệ.

Thông thường, đưa ra bộ cần chèn một cách tường minh hoặc viết một câu truy vấn mà kết quả truy vấn chính là một tập các bộ cần chèn.

Ví dụ: Chèn một bộ tường minh

$$DEAN \leftarrow DEAN \cup \{ 'DA01', 'Phổ cập tin học', 'Đức Trọng', 4 \}$$

2.6.2. Xóa

Phép xóa: $r \leftarrow r - E$, với r là một quan hệ và E là một biểu thức đại số quan hệ. Chú ý rằng phép xóa thực hiện xóa một hoặc nhiều bộ mà không thể xóa đi giá trị của các thuộc tính.

Ví dụ:

- Xóa tất cả các phân công công tác tham gia đề án của nhân viên mang mã số NV01

$$PHANCONG \leftarrow PHANCONG - (\sigma_{MaNV='NV01'}(NHANVIEN))$$

- Xóa tất cả các đề án do phòng mang tên ‘Quản Lý’ chủ trì

$$r1 \leftarrow (\sigma_{TenPhong='Quản Lý'}(PHONGBAN \bowtie_{MaPhong=Phong} DEAN))$$

$$r2 \leftarrow \pi_{MaDA, TenDA, DDiemDA, Phong}(r1)$$

$$DEAN \leftarrow DEAN - r2$$

2.6.3. Sửa

Phép sửa: $r \leftarrow \pi_{F1, F2, \dots, Fn}(r)$, với:

- F_i là một biểu thức, gồm hằng và thuộc tính của r để đưa ra giá trị mới cho thuộc tính này.
- Mỗi F_i có giá trị trả về là giá trị mới cho thuộc tính thứ i của r , thuộc tính này có thể được giữ nguyên hoặc cập nhật với giá trị mới.

Phép sửa có thể được viết thông qua phép xóa và thêm. Khi đó, phép xóa sẽ xóa đi các bộ chứa giá trị cũ và phép thêm sẽ thêm những bộ chứa giá trị mới.

Ví dụ:

- Cộng thêm lương mỗi nhân viên với số tiền là 120000

$$NHANVIEN \leftarrow (\pi_{MaNV, HoNV, TenNV, NgaySinh, DiaChi, Phai, Luong+120000, MaNQL, Phong}(NHANVIEN))$$

- Với nhân viên nam, cộng thêm lương với số tiền là 100000; với nhân viên nữ, cộng thêm lương với số tiền là 150000

$$NHANVIEN \leftarrow \left(\pi_{MaNV, HoNV, TenNV, NgaySinh, DiaChi, Phai, Luong+100000, MaNQL, Phong} (\sigma_{Phai='Nam'}(NHANVIEN)) \right) \cup \left(\pi_{MaNV, HoNV, TenNV, NgaySinh, DiaChi, Phai, Luong+150000, MaNQL, Phong} (\sigma_{Phai='Nu'}(NHANVIEN)) \right)$$

3. Bài tập

Bài 1:

Với lược đồ cơ sở dữ liệu Quản lý đề án trong 2.2. Hãy viết các biểu thức đại số quan hệ theo yêu cầu:

1. Cho biết thông tin cá nhân về những nhân viên có tên ‘Mai’
2. Tìm mã nhân viên, họ tên và địa chỉ của tất cả nhân viên làm việc phòng ‘Hành Chính’
3. Tìm mã nhân viên, họ tên và địa chỉ của tất cả nhân viên làm việc phòng ‘Hành Chính’ và ‘Tài Vụ’
4. Cho biết mã nhân viên, họ tên nhân viên và tên các đề án mà nhân viên tham gia.
5. Tìm mã đề án, tên đề án, tên phòng ban chủ trì đề án cùng mã trưởng phòng, tên trưởng phòng đó.
6. Cho biết mã nhân viên, họ tên của những nhân viên tham gia vào đề án có mã là ‘DA01’ và có thời gian làm việc cho đề án trên 30giờ/tuần
7. Cho biết mã nhân viên, họ tên của những nhân viên có cùng tên với người thân.
8. Cho biết mã nhân viên, họ tên của những nhân viên có người trưởng phòng có họ tên là ‘Nguyễn’ ‘Mai’
9. Cho biết mã nhân viên, họ tên của những nhân viên có người quản lý có họ tên là ‘Nguyễn’ ‘Mai’
10. Cho biết mã nhân viên, họ tên của những nhân viên tham gia mọi đề án của công ty.
11. Cho biết mã nhân viên, họ tên của những nhân viên không tham gia đề án nào của công ty.

12. Cho biết mức lương trung bình của nhân viên trong công ty.
13. Cho biết mức lương trung bình của nhân viên nam trong công ty.
14. Cho biết tổng số đề án của công ty.
15. Với mỗi đề án, cho biết tổng số nhân viên tham gia vào đề án.
16. Với mỗi đề án, cho biết tổng số nhân viên nữ tham gia vào đề án.
17. Tăng thời gian tham gia đề án của các nhân viên nam thêm 4giờ/tuần
18. Xóa tất cả những nhân viên có mức lương dưới 500000

Bài 2:

Cho lược đồ CSDL Quản lý sinh viên sau:

SINHVIEN (MaSV, HoSV, TenSV, NgaySinh, DiaChi, Phai, Nam, Khoa)

Tân từ: Mỗi sinh viên có Mã sinh viên (MaSV) duy nhất để phân biệt với các sinh viên khác, có họ tên (HoSV, TenSV), ngày sinh (NgaySinh), địa chỉ (DiaChi), phái Nam hoặc Nữ (Phai), năm nhập học (Nam) và thuộc về một khoa (Khoa)

GIANGVIEN (MaGV, HoGV, TenGV, NgaySinh, DiaChi, Phai, ChuyenNganh, Khoa)

Tân từ: Mỗi giảng viên có Mã giảng viên (MaGV) duy nhất để phân biệt với các giảng viên khác, có họ tên (HoGV, TenGV), ngày sinh (NgaySinh), địa chỉ (DiaChi), phái Nam hoặc Nữ (Phai), chuyên ngành (ChuyenNganh) và thuộc về một khoa (Khoa)

MONHOC (MaMH, TenMH, STC, Loai, Khoa)

Tân từ: Mỗi môn học có mã môn học (MaMH) duy nhất để phân biệt với các môn học khác, có tên môn học (TenMH), số tín chỉ (STC), là loại bắt buộc hay tự chọn (Loai), và do một khoa (Khoa) chịu trách nhiệm giảng dạy.

DIEUKIEN (MaMH, MaMHTruoc)

Tân từ: Một số môn học có điều kiện tiên quyết, sinh viên muốn học môn học (MaMH) thì phải đạt được môn tiên quyết của môn học này (MaMHTruoc)

KHOAHOC (MaKH, MaMH, HocKy, NamHoc, MaGV)

Tân từ: Một môn học (MaMH) được tổ chức trong học kỳ (HocKy) của một năm học (NamHoc) và do một giảng viên chịu trách nhiệm giảng dạy (MaGV). Lưu ý rằng một môn học có thể được mở nhiều lần (chẳng hạn năm học 2007-2008 mở cho khoá CTK29, năm học 2008-2009 mở cho khoá CTK30). Trong quan hệ này, mã khoá học (MaKH) thể hiện việc một lần tổ chức giảng dạy môn học.

KETQUA (MaSV, MaKH, Diem, KetQua)

Tân từ: Khi sinh viên (MaSV) tham gia học môn học tại một khoá học (MaKH) sẽ có điểm đánh giá (Diem) của học viên, từ điểm đánh giá sẽ có kết quả (KetQua) đạt hay không đạt.

Hãy viết các biểu thức đại số quan hệ theo yêu cầu:

1. Cho biết mã sinh viên, họ tên của mọi sinh viên
 2. Cho biết mã môn học, tên môn học và số tín chỉ tương ứng
 3. Cho biết mã môn học, tên môn học phải học trước môn có mã là 'CT101'
 4. Cho biết mã sinh viên, họ tên sinh viên cùng với các môn học mà sinh viên đạt trên 5 điểm.
 5. Cho biết mã sinh viên, họ tên sinh viên học tất cả các khóa học.
 6. Cho biết tổng số sinh viên của mỗi khoa.
 7. Cho biết mã sinh viên, họ tên sinh viên đạt điểm cao nhất trong mỗi khóa học
 8. Cho biết mã sinh viên, họ tên sinh viên và điểm trung bình của sinh viên trong từng học kỳ của từng niên học
 9. Cho biết mã giáo viên, họ tên giáo viên và chuyên ngành của những giáo viên tham gia dạy năm 2004-2005
 10. Tăng số tín chỉ cho những môn học được học trong học kỳ 1, năm 2004-2005
 11. Từ điểm của sinh viên, hãy điền vào cột KetQua thỏa: nếu điểm \geq 5: đạt, ngược lại: không đạt.
-