

TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THÔNG TIN HỮU NGHỊ VIỆT - HÀN
KHOA: KHOA HỌC MÁY TÍNH

=====



GIÁO TRÌNH
CƠ SỞ DỮ LIỆU

(Lưu hành nội bộ - Dành cho hệ đào tạo trung cấp chuyên nghiệp)

Đà Nẵng, 04/2012

TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THÔNG TIN HỮU NGHỊ VIỆT - HÀN

KHOA: KHOA HỌC MÁY TÍNH

=====

GIÁO TRÌNH
CƠ SỞ DỮ LIỆU

(Lưu hành nội bộ - Dành cho hệ đào tạo Trung cấp chuyên nghiệp)

Chủ biên: ThS.Phí Đắc Hải

Biên soạn: ThS.Trịnh Thị Ngọc Linh

Đà Nẵng, 04/2012

LỜI NÓI ĐẦU

Trước khi các hệ cơ sở dữ liệu ra đời, mỗi chương trình ứng dụng đều có một tập tin dữ liệu tương ứng, và mỗi khi chương trình ứng dụng cần được sửa đổi hoặc mở rộng thì tập tin dữ liệu tương ứng cũng thay đổi theo, điều này dẫn đến sự cô lập về dữ liệu, dư thừa dữ liệu và không nhất quán, khó đảm bảo các vấn đề về bảo mật hay truy cập đồng thời...

Các hệ cơ sở dữ liệu đầu tiên xuất hiện vào giữa năm 1960, được xây dựng theo mô hình phân cấp và mô hình mạng, được xem là thuộc thể hệ thứ nhất của hệ quản trị cơ sở dữ liệu. Tiếp theo là thể hệ thứ hai, các hệ quản trị cơ sở dữ liệu quan hệ, được xây dựng theo mô hình dữ liệu quan hệ do E.F.Codd đề xuất năm 1970. Các hệ quản trị cơ sở dữ liệu có mục tiêu tổ chức dữ liệu, truy cập và cập nhật những khối lượng lớn dữ liệu một cách thuận lợi, an toàn và hiệu quả. Trong những năm 1990, xuất hiện các hệ quản trị cơ sở dữ liệu thể hệ thứ ba, đó là các hệ hướng đối tượng, có khả năng hỗ trợ các ứng dụng đa phương tiện.

Các hệ cơ sở dữ liệu ngày càng đóng vai trò quan trọng, không còn là một ứng dụng chuyên biệt mà đã trở thành một thành phần trung tâm của một môi trường tính toán hiện đại. Một quyển danh bạ điện thoại, danh mục khách hàng, hàng trăm địa chỉ email, hay các thông tin cần ghi nhớ khác đều cần đến cơ sở dữ liệu để lưu trữ và truy xuất.

Chính vì vậy, việc trang bị các kiến thức nền tảng về hệ cơ sở dữ liệu cho các sinh viên ở bậc cao đẳng và đại học đã trở thành một phần quan trọng trong giáo dục và đào tạo khoa học máy tính.

Mục đích của cuốn giáo trình “Cơ sở dữ liệu” nhằm trình bày các khái niệm và thuật toán cơ sở về hệ cơ sở dữ liệu, bao gồm:

- Các khái niệm cơ bản về hệ cơ sở dữ liệu
- Các mô hình dữ liệu, đặc biệt là mô hình quan hệ, lý thuyết thiết kế cơ sở dữ liệu quan hệ
- Các ngôn ngữ cơ sở dữ liệu, đặc biệt là ngôn ngữ truy vấn dữ liệu SQL

Sau khi học xong học phần này, sinh viên sẽ có những kiến thức cơ bản để có thể nghiên cứu sâu hơn về cơ sở dữ liệu, thiết kế cơ sở dữ liệu cho các chương trình ứng dụng, hoặc lưu trữ thông tin tại các đơn vị.

Tuy đã rất cố gắng nhưng giáo trình này chắc chắn không tránh khỏi sai sót. Rất mong nhận được ý kiến đóng góp để giáo trình hoàn thiện hơn.

Tôi xin chân thành cảm ơn các giảng viên Trường Cao đẳng Công nghệ thông tin Hữu nghị Việt – Hàn đã giúp tôi hoàn thành giáo trình này.

Đà Nẵng, tháng 04 năm 2012

Tác giả

MỤC LỤC

THUẬT NGỮ VÀ TỪ VIẾT TẮT	vii
DANH MỤC BẢNG BIỂU	viii
DANH MỤC HÌNH VẼ.....	ix
CHƯƠNG 1 TỔNG QUAN VỀ CƠ SỞ DỮ LIỆU	1
1.1. Hệ xử lý tập tin truyền thông.....	1
1.1.1. Giới thiệu.....	1
1.1.2. Hạn chế của hệ xử lý tập tin truyền thông	2
1.2. Hệ cơ sở dữ liệu.....	3
1.2.1. Cơ sở dữ liệu (Database)	3
1.2.2. Hệ quản trị cơ sở dữ liệu (DataBase Management System - DBMS).....	5
1.2.2.1. Vài nét về quá trình phát triển các hệ quản trị CSDL.....	5
1.2.2.2. Các chức năng của hệ quản trị CSDL.....	6
1.2.2.3. Giới thiệu về hệ quản trị CSDL Access	6
1.2.3. Hệ cơ sở dữ liệu	12
1.2.3.1. Định nghĩa và phân loại	12
1.2.3.2. Sự cần thiết của các hệ cơ sở dữ liệu	16
1.3. Đối tượng sử dụng	17
1.4. Ưu và nhược điểm của hệ CSDL.....	17
1.4.1. Ưu điểm	17
1.4.2. Nhược điểm	18
CHƯƠNG 2 MÔI TRƯỜNG CƠ SỞ DỮ LIỆU	20
2.1. Kiến trúc ANSI-PARC 3-mức.....	20
2.1.1. Mức vật lý (mức trong)	20
2.1.2. Mức logic (mức khái niệm)	20
2.1.3. Mức khung nhìn (mức ngoài)	21
2.1.4. Lược đồ và thể hiện của CSDL	23
2.1.5. Tính độc lập dữ liệu	24
2.2. Mô hình dữ liệu	25
2.2.1. Mô hình thực thể liên kết.....	25
2.2.1.1. Thực thể.....	26
2.2.1.2. Liên kết	27
2.2.1.3. Tỷ số	28

2.2.2. Mô hình hướng đối tượng	30
2.2.3. Mô hình mạng	31
2.2.4. Mô hình phân cấp.....	32
2.2.5. Mô hình quan hệ.....	32
2.2.6. Lựa chọn giữa các mô hình dữ liệu	33
CHƯƠNG 3 MÔ HÌNH QUAN HỆ	39
3.1. Mở đầu.....	39
3.2. Các khái niệm của mô hình quan hệ.....	39
3.2.1. Miền (Domain)	39
3.2.2. Thuộc tính (Attribute).....	39
3.2.3. Quan hệ (Relation)	40
3.2.4. Lược đồ quan hệ (Relation scheme).....	41
3.2.5. Bộ (tuple).....	42
3.2.6. Thể hiện của quan hệ (view)	42
3.2.7. Khóa (key)	42
3.3. Tính toàn vẹn của quan hệ.....	43
3.3.1. Khái niệm ràng buộc toàn vẹn	43
3.3.2. Ràng buộc toàn vẹn thực thể.....	46
3.3.3. Ràng buộc toàn vẹn tham chiếu	46
3.3.4. Ràng buộc toàn vẹn ngữ nghĩa	47
3.4. Các ngôn ngữ quan hệ	47
3.4.1. Đại số quan hệ	47
3.4.1.1. Phép chiếu	47
3.4.1.2. Phép chọn	48
3.4.1.3. Phép hợp.....	49
3.4.1.4. Phép hiệu	49
3.4.1.5. Phép giao.....	49
3.4.1.6. Tích Đề-các	50
3.4.1.7. Phép nối.....	50
3.4.1.8. Phép chia	51
3.4.1.9. Các chương trình đại số quan hệ	52
3.4.2. Phép tính quan hệ.....	53
3.4.2.1. Phép tính quan hệ bộ (Codd 1970).....	53
3.4.2.2. Phép tính quan hệ miền	54
3.5. Chuyển đổi mô hình thực thể liên kết sang mô hình quan hệ	54

3.6. Khung nhìn (View).....	54
3.6.1. Khái niệm quan hệ cơ sở và khung nhìn	55
3.6.2. Mục đích của khung nhìn	55
3.6.3. Vấn đề cập nhật qua khung nhìn.....	55
CHƯƠNG 4 NGÔN NGỮ TRUY VẤN DỮ LIỆU.....	59
4.1. Giới thiệu.....	59
4.1.1. Lịch sử phát triển	59
4.1.2. Chuẩn SQL	59
4.1.3. Đặc điểm của SQL	59
4.1.4. Vai trò của SQL.....	60
4.1.5. Phân loại các lệnh.....	61
4.2. Ngôn ngữ định nghĩa dữ liệu	61
4.2.1. Tạo cơ sở dữ liệu.....	61
4.2.2. Tạo bảng.....	61
4.2.3. Sửa đổi cấu trúc bảng	63
4.2.4. Xóa đối tượng.....	63
4.2.5. Các kiểu thuộc tính.....	64
4.2.6. Các loại ràng buộc trong bảng dữ liệu.....	65
4.2.6.1. NOT NULL - Không rỗng.....	65
4.2.6.2. UNIQUE - Duy nhất	65
4.2.6.3. PRIMARY KEY - Khoá chính	66
4.2.6.4. FOREIGN KEY - Khoá ngoại.....	66
4.2.6.5. CHECK - Ràng buộc kiểm tra giá trị	67
4.2.6.6. DEFAULT - Mặc định	67
4.3. Ngôn ngữ thao tác dữ liệu	68
4.3.1. Thêm bản ghi (INSERT)	68
4.3.2. Xóa bản ghi (DELETE).....	68
4.3.3. Sửa bản ghi (UPDATE)	69
4.3.4. Truy vấn dữ liệu (SELECT).....	69
4.3.4.1. Tránh các giá trị trùng lặp (DISTINCT)	70
4.3.4.2. Sử dụng bí danh cột	70
4.3.4.3. Sắp xếp thứ tự (ORDER BY).....	71
4.3.4.4. Hiển thị dữ liệu theo điều kiện (WHERE)	71
4.3.4.4. Nhóm các thông tin (GROUP BY).....	73
4.3.4.5. Mệnh đề HAVING.....	74

4.4. Ngôn ngữ điều khiển dữ liệu (DCL)	75
4.4.1. Lệnh GRANT	75
4.4.2. Lệnh DENY.....	76
4.4.3. Lệnh REVOKE	76
4.5. Các hàm thư viện của SQL.....	77
4.5.1. Hàm số học.....	77
4.5.2. Các hàm ký tự.....	77
4.5.3. Các hàm ngày.....	78
4.5.4. Các hàm chuyển đổi kiểu	78
4.5.4. Hàm nhóm	78
CHƯƠNG 5 LÝ THUYẾT THIẾT KẾ CƠ SỞ DỮ LIỆU QUAN HỆ	85
5.1. Mục đích của việc chuẩn hoá	85
5.2. Dư thừa thông tin và cập nhật dị thường.....	85
5.2.1. Sự dư thừa dữ liệu	85
5.2.2. Các dị thường cập nhật dữ liệu.....	85
5.3. Phụ thuộc hàm	87
5.3.1. Định nghĩa phụ thuộc hàm	87
5.3.2. Các qui tắc phụ thuộc hàm	88
5.3.2.1. Hệ tiên đề Armstrong cho các phụ thuộc hàm.....	88
5.3.2.2. Các tính chất của phụ thuộc hàm	88
5.4. Suy diễn lô-gíc.....	89
5.5. Bao đóng của tập phụ thuộc hàm.....	89
5.5.1. Định nghĩa.....	89
5.5.2. Phụ thuộc hàm đầy đủ.....	90
5.5.3. Phụ thuộc hàm dư thừa	90
5.5.4. Phủ của tập phụ thuộc hàm, phụ thuộc hàm tương đương.....	91
5.6. Bao đóng của tập thuộc tính	91
5.6.1. Định nghĩa.....	91
5.6.2. Thuật toán tìm bao đóng	91
5.8. Phép tách lược đồ quan hệ.....	92
5.8.1. Định nghĩa.....	92
5.8.2. Phép tách bảo toàn thông tin	92
5.8.3. Thuật toán kiểm tra phép tách bảo toàn thông tin.....	93
5.9. Qui trình chuẩn hoá	94
5.9.1. Dạng chuẩn một (1NF)	94

5.9.2. <i>Dạng chuẩn thứ 2 (2NF)</i>	95
5.9.3. <i>Dạng chuẩn thứ 3 (3NF)</i>	97
5.9.4. <i>Thuật toán chuẩn hóa một lược đồ cơ sở dữ liệu</i>	98
TÀI LIỆU THAM KHẢO	101

THUẬT NGỮ VÀ TỪ VIẾT TẮT

CSDL, DB	Cơ sở dữ liệu - Database
CNTT	Công nghệ thông tin
DBA	Người quản trị CSDL - Database Administrator
DCL	Ngôn ngữ quản lý dữ liệu - Data Control Language
DDL	Ngôn ngữ định nghĩa dữ liệu - Data Definition Language
DML	Ngôn ngữ thao tác dữ liệu - Data Manipulation Language
HĐH	Hệ điều hành
Hệ QTCSDL, DBMS	Hệ quản trị cơ sở dữ liệu - Database management systems
NSD	Người sử dụng - User
File	Tập tin, tệp
SQL	Ngôn ngữ truy vấn có cấu trúc - Structured Query Language

DANH MỤC BẢNG BIỂU

Bảng 2.1. Thẻ hiện CSDL gồm tên khách hàng, số chứng minh thư, nơi ở, tài khoản.	24
Bảng 2.2. Một bảng trong mô hình CSDL quan hệ.....	33
Bảng 2.3. Bảng HocSinh	35
Bảng 2.4. Bảng Lớp.....	35
Bảng 3.1. Quan hệ LopHoc	42
Bảng 3.2. Quan hệ MonHoc	42
Bảng 3.3. Quan hệ SinhVien	46
Bảng 3.4. Quan hệ Lop và SinhVien.....	46
Bảng 4.1. Các thành phần của lệnh tạo bảng.....	62
Bảng 4.2. Các loại dữ liệu của Access	64
Bảng 4.3. Các kiểu dữ liệu số trong Access	65
Bảng 4.4. Bảng dữ liệu Nhân viên	70

DANH MỤC HÌNH VẼ

Hình 1.1. Sơ đồ tổng quát về một CSDL.....	3
Hình 1.2. Hệ xử lý tập tin truyền thống.....	4
Hình 1.3. Hệ cơ sở dữ liệu.....	4
Hình 1.4. Môi trường hệ cơ sở dữ liệu	13
Hình 1.5. Hệ cơ sở dữ liệu cá nhân	14
Hình 1.6. Hệ cơ sở dữ liệu trung tâm	14
Hình 1.7. Hệ cơ sở dữ liệu khách/chủ	15
Hình 1.8. Các hệ CSDL phân tán	16
Hình 2.1. Kiến trúc ANSI-PARC 3-mức.....	22
Hình 2.2. Ví dụ về lược đồ CSDL	23
Hình 2.3. Sự độc lập dữ liệu của kiến trúc ba mức ANSI-SPARC	24
Hình 2.4. Thực thể “Nhân viên” và các thuộc tính	26
Hình 2.5. Quan hệ giữa hai thực thể.....	27
Hình 2.6. Quan hệ đệ quy	27
Hình 2.7. Thuộc tính của liên kết	28
Hình 2.8. Hai cách biểu diễn tỉ số lực lượng 1:1 giữa Nhân viên và Phòng	28
Hình 2.9. Hai cách biểu diễn tỉ số lực lượng 1:N giữa Phòng và Nhân viên	29
Hình 2.10. Hai cách biểu diễn tỉ số lực lượng N:1 giữa Nhân viên và Phòng	29
Hình 2.11. Hai cách biểu diễn tỉ số lực lượng N:M giữa “Nhân viên” và “Dự án”	29
Hình 2.12. Ví dụ mô hình thực thể liên kết.....	30
Hình 2.13. Ví dụ mô hình CSDL hướng đối tượng.....	31
Hình 2.14. Ví dụ mô hình CSDL mạng.....	31
Hình 2.15. Ví dụ mô hình CSDL phân cấp	32
Hình 2.16. Ví dụ mô hình CSDL quan hệ	33
Hình 5.1 Quy trình chuẩn hoá	94
Hình 5.2. Sử dụng phép chiếu tách về 2NF.....	97

CHƯƠNG 1

TỔNG QUAN VỀ CƠ SỞ DỮ LIỆU

Phần đầu tiên của chương 1 sẽ trình bày về những hạn chế của kiểu tập tin truyền thống và sự cần thiết phải ứng dụng hệ cơ sở dữ liệu (CSDL), phần tiếp theo giới thiệu các khái niệm chính của hệ CSDL, và phần cuối cùng trình bày về những đối tượng sử dụng và các ưu, nhược điểm của hệ CSDL. Nội dung của chương này giúp chúng ta hiểu được hệ CSDL là gì và tầm quan trọng của nó trong đời sống hiện nay.

1.1. Hệ xử lý tập tin truyền thống

1.1.1. Giới thiệu

Trước khi các hệ CSDL ra đời (khoảng đầu những năm 60 của thế kỷ 20), mỗi chương trình ứng dụng đều có một tập tin dữ liệu tương ứng, và mỗi khi chương trình ứng dụng cần được sửa đổi hoặc mở rộng thì tập tin dữ liệu tương ứng cũng phải thay đổi theo. Cách quản lý dữ liệu như vậy gọi là hệ xử lý tập tin truyền thống.

Hệ xử lý tập tin là một tập hợp các chương trình ứng dụng đáp ứng các yêu cầu của người sử dụng (NSD). Mỗi chương trình định nghĩa và quản lý dữ liệu của chính nó.

Đối với một công ty, cách tiếp cận truyền thống đối với hệ thống thông tin thường chỉ tập trung vào các nhu cầu xử lý dữ liệu của các phòng riêng lẻ trong một công ty, mà không xem xét công ty này như một tổng thể. Các phòng sử dụng các ứng dụng khác nhau và mỗi ứng dụng có các tập tin dữ liệu riêng của nó (Xem ví dụ dưới đây).

Ví dụ:

Tại một công ty người ta trang bị máy vi tính cho tất cả các phòng, ban nghiệp vụ.

Phòng Kế toán sử dụng máy tính để tính lương và in danh sách lương của từng bộ phận trong công ty, dựa trên danh sách cán bộ viên chức cùng hệ số lương và các hệ số phụ cấp của họ do phòng Tổ chức cung cấp. Thông tin mà phòng Kế toán quản lý và khai thác là: Họ và Tên, Hệ số lương, Hệ số phụ cấp, Phụ cấp khác của các công nhân viên chức (CNVC) xếp theo từng phòng ban, và sử dụng công cụ văn phòng là Microsoft Excel.

Phòng Tổ chức quản lý thông tin lý lịch của CNVC chi tiết hơn gồm Họ CNVC, Tên CNVC (để riêng thành một cột "Tên" để tiện sắp xếp theo vần Alphabet), Bí danh, Giới tính, Ngày sinh, Ngày tuyển dụng, Hoàn cảnh gia đình, Quá trình được đào tạo, Hệ số lương, Hệ số phụ cấp... nhưng thiếu thông tin về Phụ cấp khác của CNVC. Phần mềm được sử dụng để quản lý là FoxPro for Windows.

Trong khi đó, tại Tổng công ty của họ, các phòng ban nghiệp vụ cũng được trang bị vi tính. *Phòng Tổ chức cán bộ tại Tổng công ty sử dụng phần mềm Microsoft Access để quản lý CNVC* gồm các cán bộ chủ chốt từ trường phó phòng, quản đốc và phó quản đốc xí nghiệp trở lên của các công ty con trực thuộc. Thông tin quản lý tại đây cũng giống như thông tin quản lý tại phòng tổ chức của công ty con.

Việc quản lý như trên có những ưu điểm sau:

- Việc xây dựng hệ thống các tập 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 cho mục đích hẹp nên *khả năng đáp ứng nhanh chóng, kịp thời*.

Tuy nhiên, quản lý như vậy sẽ tồn tại các hạn chế:

- Do thông tin được tổ chức ở mỗi phòng ban mỗi khác, cũng như phần mềm công cụ để triển khai mỗi nơi cũng khác nhau nên sự phối hợp tổ chức và khai thác ở các phòng ban là khó khăn. Cùng một thông tin được nhập vào máy tại nhiều nơi khác nhau gây ra *lãng phí công sức nhập tin và không gian lưu trữ thông tin*. Sự trùng lặp thông tin có thể dẫn đến tình trạng *không nhất quán dữ liệu*. Chẳng hạn, ở phòng Tổ chức ghi đầy đủ họ tên nhân viên là Nguyễn Văn Quang nhưng tại phòng Kế toán chỉ ghi tắt là Nguyễn v Quang, hay một cán bộ chủ chốt của công ty có thay đổi về hoàn cảnh gia đình (mới cưới vợ / lấy chồng, sinh thêm con...) có thể được cập nhật ngay tại đơn vị nhưng sau một thời gian mới được cập nhật tại Tổng công ty.

- Do hệ thống được tổ chức thành các tập tin riêng lẻ nên việc kết nối các hệ thống này hay việc *nâng cấp ứng dụng* sẽ là *rất khó khăn*.

1.1.2. Hạn chế của hệ xử lý tập tin truyền thống

- **Cô lập và hạn chế chia sẻ dữ liệu:** Dữ liệu nằm rải rác trong nhiều tập tin và các tập tin có thể có định dạng khác nhau, nên NSD khó khai thác các dữ liệu chứa trong các tập tin đang tồn tại, trong khi các tập tin mới được thiết kế lặp lại các dữ liệu đang tồn tại.

- **Dư thừa dữ liệu và tính không nhất quán:** Cùng một dữ liệu có thể được lưu trữ trong nhiều tập tin khác nhau, khi tiến hành cập nhật có thể bỏ sót và dẫn đến không nhất quán.

- **Khó khăn trong việc truy xuất dữ liệu:** Môi trường của hệ thống xử lý tập tin không cung cấp các công cụ cho phép truy xuất dữ liệu một cách hiệu quả và thuận lợi.

- **Các vấn đề về tính nguyên tử:** Tính nguyên tử của một giao dịch (hoạt động) là: hoặc thực hiện hoàn toàn hoặc không thực hiện gì cả. Trong hệ xử lý tập tin truyền thống khó đảm bảo tính nguyên tử này và khó đưa được hệ thống trở về trạng thái như trước khi xảy ra sự cố.

- **Tính dị thường trong truy cập tương tranh:** Một hệ thống cho phép nhiều NSD cập nhật dữ liệu đồng thời và như thế có thể dẫn đến dữ liệu không nhất quán, điều này đòi hỏi một sự giám sát, nhưng hệ xử lý tập tin truyền thống không cung cấp chức năng này.

- **Các vấn đề về toàn vẹn:** Các giá trị dữ liệu được lưu trữ trong CSDL phải thỏa mãn các kiểu ràng buộc toàn vẹn nhất định. Các nhà phát triển làm hiệu lực các ràng buộc này trong hệ thống bằng cách thêm các đoạn mã tương ứng vào các chương trình ứng dụng. Tuy nhiên, khi các ràng buộc mới được bổ sung thêm đối với các tập tin, sẽ rất khó thay đổi các chương trình để làm hiệu lực các ràng buộc mới này. Vấn đề này phức tạp hơn khi các ràng buộc đó lôi kéo một số khoản mục dữ liệu từ các tập tin khác nhau.

- **Vấn đề an toàn:** Hệ thống phải đảm bảo được tính phân quyền, chống truy xuất trái phép. Hệ xử lý tập tin truyền thống khó đảm bảo tính chất này.

Qua phân tích trên chúng ta nhận thấy việc tổ chức dữ liệu theo hệ xử lý tập tin truyền thống có nhiều nhược điểm.

Việc xây dựng một hệ thống tin đảm bảo được tính chất nhất quán dữ liệu, không trùng lặp thông tin mà vẫn đáp ứng được nhu cầu khai thác dữ liệu đồng thời là thực sự cần thiết, điều này đã dẫn đến sự ra đời của hệ cơ sở dữ liệu.

1.2. Hệ cơ sở dữ liệu

1.2.1. Cơ sở dữ liệu (Database)

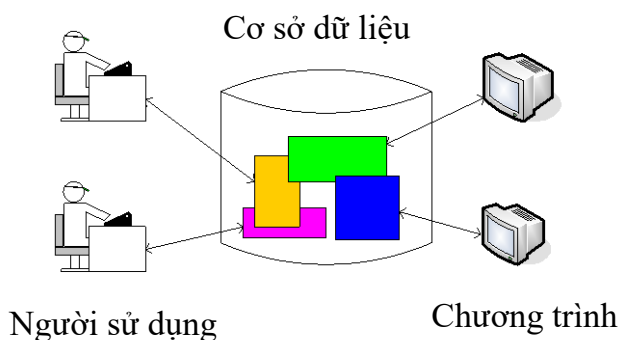
Dữ liệu là các sự kiện, văn bản, hình ảnh, âm thanh, phim ảnh... có ý nghĩa được ghi lại và lưu trữ trên máy tính.

Thông tin là các dữ liệu đã được xử lý, hữu ích cho việc đưa ra các quyết định.

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

Chúng ta có thể hiểu một cách đơn giản CSDL là một danh sách thông tin. Như trang niên giám điện thoại chẳng hạn, mỗi trang là một danh sách chứa các mục thông tin gồm tên, địa chỉ, số điện thoại, mô tả về người thuê bao điện thoại trong một vùng nào đó (thông tin mô tả đối tượng). Tất cả thông tin của người thuê bao dùng chung một mẫu (cấu trúc). Theo thuật ngữ của CSDL, các trang niên giám tương đương với một bảng (table) dữ liệu mà trong đó thông tin mỗi người thuê bao được đại diện hay biểu diễn bởi một bản ghi (record). Thông tin bản ghi mô tả về người thuê bao chứa ba mục: tên, địa chỉ và số điện thoại. Các bản ghi được sắp xếp theo thứ tự abc và được gọi là khóa dùng để tìm kiếm khi cần.

Các ví dụ khác về CSDL là danh sách khách hàng, danh mục hay danh sách sinh viên, danh sách hàng hóa, và ngay cả nội dung trang Web cũng có thể xem là một CSDL.

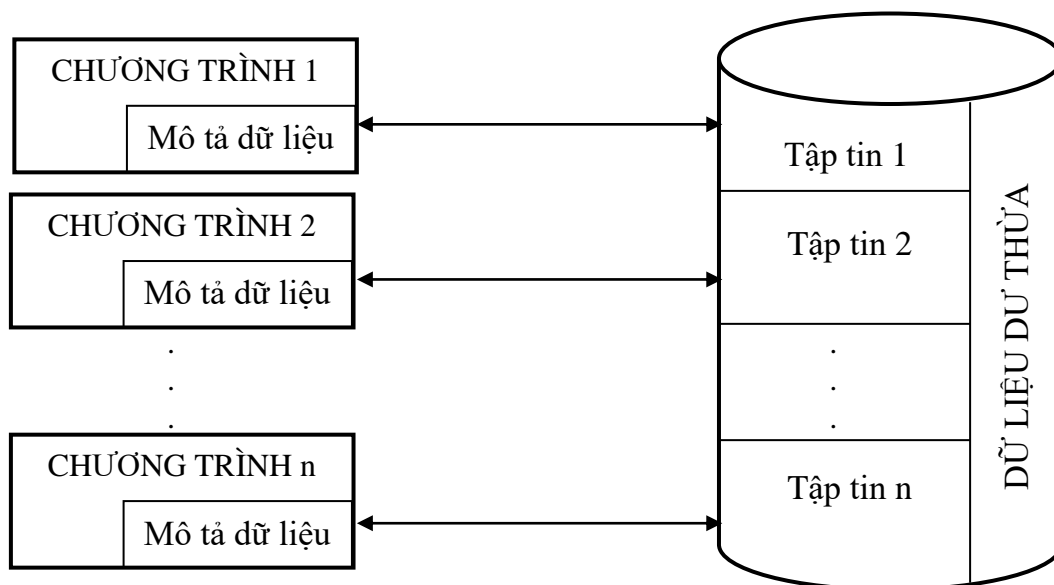


Hình 1.1. Sơ đồ tổng quát về một CSDL

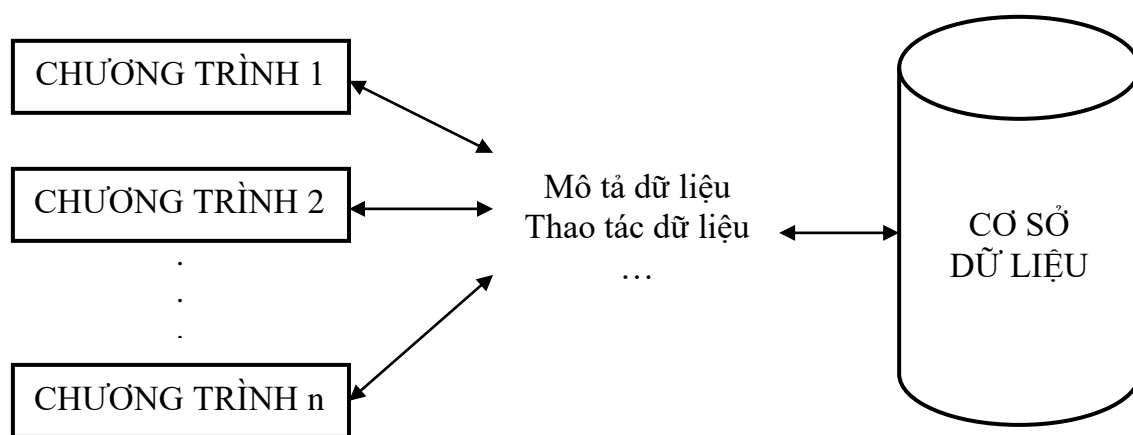
Trên Hình 1.1, chúng ta nhìn thấy các thành phần CSDL hợp nhất là một bộ sưu tập các dữ liệu chứa trên các phương tiện lưu trữ (như đĩa từ hay băng từ), những NSD trực tuyến hay các chương trình ứng dụng có thể sử dụng (tìm kiếm, sửa đổi, bổ sung, loại bỏ thông tin...), chia sẻ CSDL này.

Một trong những động cơ chính phía sau việc sử dụng các hệ CSDL là mong muốn tích hợp các dữ liệu tác nghiệp của công ty, cho phép truy cập tập trung và có điều khiển đến các dữ liệu đó, tránh việc từng phòng ban xử lý dữ liệu riêng lẻ như đối với hệ xử lý tập tin truyền thống.

Hình 1.2 và 1.3 cho thấy rõ những khác biệt của hệ xử lý tập tin truyền thống và hệ CSDL.



Hình 1.2. Hệ xử lý tập tin truyền thống



Hình 1.3. Hệ cơ sở dữ liệu

Trong Hình 1.2, mỗi chương trình sử dụng một tập tin dữ liệu nên dẫn đến dư thừa dữ liệu. Trong Hình 1.3, dữ liệu được quản lý tập trung và các chương trình có thể truy cập khi cần thiết.

Cần lưu ý những điểm sau đây:

- **Cơ sở dữ liệu là tài nguyên thông tin dùng chung cho nhiều người.** Bất kỳ người sử dụng nào trên mạng máy tính, tại các thiết bị đầu cuối, về nguyên tắc có quyền truy nhập khai thác toàn bộ hay một phần dữ liệu theo chế độ trực tuyến hay tương tác mà không phụ thuộc vào vị trí địa lý của người sử dụng với các tài nguyên đó.

- **Cơ sở dữ liệu được các hệ ứng dụng khai thác bằng ngôn ngữ con dữ liệu hoặc bằng các chương trình ứng dụng để xử lý, tìm kiếm, tra cứu, sửa đổi, bổ sung hay loại bỏ dữ liệu.** Tìm kiếm và tra cứu thông tin là một trong những chức năng quan trọng và phổ biến nhất của các dịch vụ cơ sở dữ liệu. Hệ quản trị CSDL (Phần 1.3.2) là phần mềm điều khiển các chiến lược truy nhập CSDL. Khi người sử dụng đưa ra yêu cầu truy nhập bằng một ngôn ngữ con dữ liệu nào đó, hệ quản trị CSDL tiếp nhận và thực hiện các thao tác trên CSDL lưu trữ.

- **Đối tượng nghiên cứu của CSDL là các thực thể và mối quan hệ giữa các thực thể.** Thực thể và mối quan hệ giữa các thực thể là hai đối tượng khác nhau về căn bản. Mối quan hệ giữa các thực thể cũng là một loại thực thể đặc biệt. Trong cách tiếp cận CSDL quan hệ (Chương 3), người ta dựa trên cơ sở lý thuyết đại số quan hệ để xây dựng các quan hệ chuẩn, khi kết nối không tổn thất thông tin và khi biểu diễn dữ liệu là duy nhất. Dữ liệu được lưu trữ trong bộ nhớ của máy tính không những phải tính đến yếu tố về tối ưu không gian lưu trữ, mà phải đảm bảo tính khách quan, trung thực của dữ liệu hiện thực, nghĩa là phải đảm bảo tính nhất quán của dữ liệu và giữ được sự toàn vẹn của dữ liệu.

1.2.2. Hệ quản trị cơ sở dữ liệu (DataBase Management System - DBMS)

Để giải quyết tốt các vấn đề đặt ra cho một CSDL như tính chủ quyền, cơ chế bảo mật hay phân quyền khai thác CSDL, giải quyết tranh chấp trong quá trình truy cập dữ liệu, phục hồi dữ liệu khi có sự cố...thì cần phải có một hệ thống các phần mềm chuyên dụng. Hệ thống các phần mềm đó được gọi là hệ quản trị CSDL. Đó là các công cụ hỗ trợ tích cực cho các nhà phân tích và thiết kế CSDL, hoặc những người khai thác CSDL.

Phần mềm cho phép người dùng giao tiếp với CSDL, cung cấp một môi trường thuận lợi và hiệu quả để tìm kiếm và lưu trữ thông tin của CSDL được gọi là hệ quản trị cơ sở dữ liệu (hệ quản trị CSDL).

Có hai khả năng cho phép phân biệt hệ quản trị CSDL với các kiểu hệ thống lập trình khác, đó là hệ quản trị CSDL có khả năng *quản lý dữ liệu tồn tại lâu dài* và khả năng *truy cập các khối lượng dữ liệu lớn một cách hiệu quả*.

1.2.2.1. Vài nét về quá trình phát triển các hệ quản trị CSDL

Các hệ quản trị CSDL đầu tiên ra đời vào đầu những năm 60 của thế kỷ 20 dựa trên mô hình dữ liệu phân cấp và mạng, trong số đó có hệ quản trị CSDL có tên là IMS của hãng IBM dựa trên mô hình dữ liệu phân cấp.

Năm 1976, hệ quản trị CSDL đầu tiên dựa trên mô hình dữ liệu quan hệ của hãng IBM mang tên System-R ra đời. Từ năm 1980, hãng IBM cho ra đời hệ quản trị CSDL trên các máy MainFrame mang tên DB2, tiếp theo là các hệ quản trị CSDL Dbase, Sybase, Oracle, Informix, Microsoft SQL Server...Các hệ quản trị CSDL có mục tiêu tổ chức dữ liệu, truy cập và cập nhật những khối lượng lớn dữ liệu một cách thuận lợi, an toàn và hiệu quả.

Từ những năm 1990 người ta bắt đầu cố gắng xây dựng các hệ quản trị CSDL hướng đối tượng (Oriented Object DataBase Management System) như Orion, Illustra, Itasca...Tuy nhiên hầu hết các hệ này đều vẫn là quan hệ - hướng đối tượng, nghĩa là, xét về bản chất, chúng vẫn dựa trên nền tảng của mô hình quan hệ. Hệ quản trị CSDL hướng đối tượng thuần nhất có thể là hệ ODMG (Object Data Management Group) ra đời vào năm 1996. Hệ quản trị hướng đối tượng có khả năng hỗ trợ các ứng dụng đa

phương tiện, vì có thể quản lý các đối tượng có cấu trúc phức tạp (văn bản, âm thanh, hình ảnh) và động (các chương trình, mô phỏng).

Như vậy, cho đến nay có khá nhiều hệ quản trị CSDL mạnh được đưa ra thị trường như: Visual FoxPro, Microsoft Access, Microsoft SQL Server, Oracle, mySQL, DB2, Sybase, Paradox, Informix,...với các chất lượng khác nhau.

1.2.2.2. Các chức năng của hệ quản trị CSDL

- Cung cấp cho người dùng khả năng lưu giữ, truy xuất và cập nhật dữ liệu. Với chức năng này, hệ quản trị CSDL đã che giấu người dùng những chi tiết mang tính vật lý.

- Cung cấp cho người dùng một từ điển dữ liệu, đó là mô tả về dữ liệu được lưu trữ và người dùng truy cập được vào từ điển dữ liệu này.

- Hỗ trợ các giao dịch bằng cách cung cấp một cơ chế đảm bảo rằng: hoặc tất cả các cập nhật trong một giao dịch làm việc được thực hiện, hoặc không thao tác cập nhật nào trong giao dịch này được thực hiện.

- Cung cấp các dịch vụ điều khiển tương tranh để đảm bảo tính nhất quán dữ liệu khi có nhiều phiên làm việc với CSDL, có nhiều người đồng thời truy cập vào CSDL, đặc biệt là những truy cập làm thay đổi thông tin lưu trong đó.

- Cung cấp một cơ chế để khôi phục dữ liệu khi xảy ra sự cố làm hỏng CSDL theo một kiểu nào đó.

- Cung cấp các dịch vụ bản quyền: Điều này có nghĩa là một cơ chế chỉ đảm bảo những người có quyền mới được truy cập CSDL ở những mức độ khác nhau.

- Hỗ trợ cho truyền thông dữ liệu: Hệ quản trị CSDL phải có khả năng tích hợp được với các phần mềm truyền thông. Hầu hết người dùng truy cập vào CSDL từ trạm làm việc đầu cuối. Có thể những trạm làm việc đầu cuối này được nối trực tiếp với máy tính chủ của hệ quản trị CSDL, nhưng nhiều trường hợp trạm làm việc đầu cuối đặt tại vùng xa và truyền thông với máy tính chủ thông qua mạng. Trong những trường hợp như vậy, hệ quản trị nhận được các yêu cầu trong các thông điệp truyền thông gửi đến và trả kết quả lại cũng theo cách truyền thông điệp như vậy.

- Cung cấp các dịch vụ đảm bảo tính toàn vẹn dữ liệu: Điều này nghĩa là đảm bảo dữ liệu trong CSDL và những thay đổi dữ liệu tuân theo những luật đã xác định. Tính toàn vẹn dữ liệu là để nhằm đảm bảo sự chính xác và nhất quán của dữ liệu được lưu trữ.

1.2.2.3. Giới thiệu về hệ quản trị CSDL Access

Microsoft Access là một thành phần trong bộ Microsoft Office của hãng Microsoft. MS Access cung cấp cho người dùng giao diện thân thiện và các thao tác đơn giản, trực quan trong việc xây dựng và quản trị cơ sở dữ liệu cũng như xây dựng các ứng dụng cơ sở dữ liệu.

Access là một hệ quản trị cơ sở dữ liệu quan hệ (RDMS- Relational Database Management System), rất phù hợp cho các bài toán quản lý vừa và nhỏ. Hiệu năng cao và đặc biệt dễ sử dụng do giao diện giống các phần mềm khác trong bộ MS Office như MS Word, MS Excel.

Access còn cung cấp hệ thống công cụ phát triển khá mạnh đi kèm (Development Tools) giúp các nhà phát triển phần mềm đơn giản trong việc xây dựng trọn gói các dự án phần mềm quản lý qui mô vừa và nhỏ.

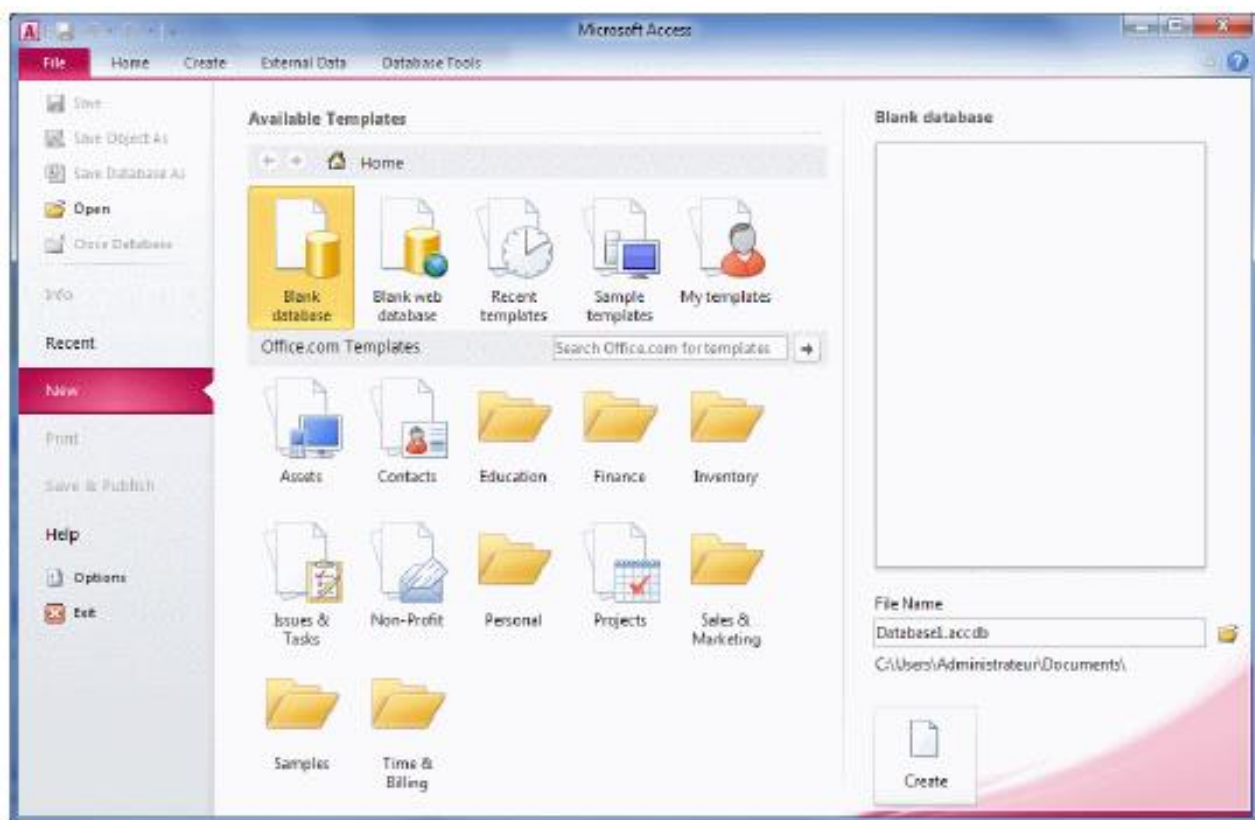
Có rất nhiều phiên bản của Access. Trong giáo trình này sẽ trình bày về MS Access 2010.

- Khởi động Access 2010:

Để khởi động MS Access ta có thể dùng một trong các cách sau:

Cách 1: Start → (All) Programs → Microsoft Office → Microsoft Office Access 2010.

Cách 2: Double click vào shortcut Ms Access trên desktop, xuất hiện cửa sổ khởi động Access như Hình 1.4.



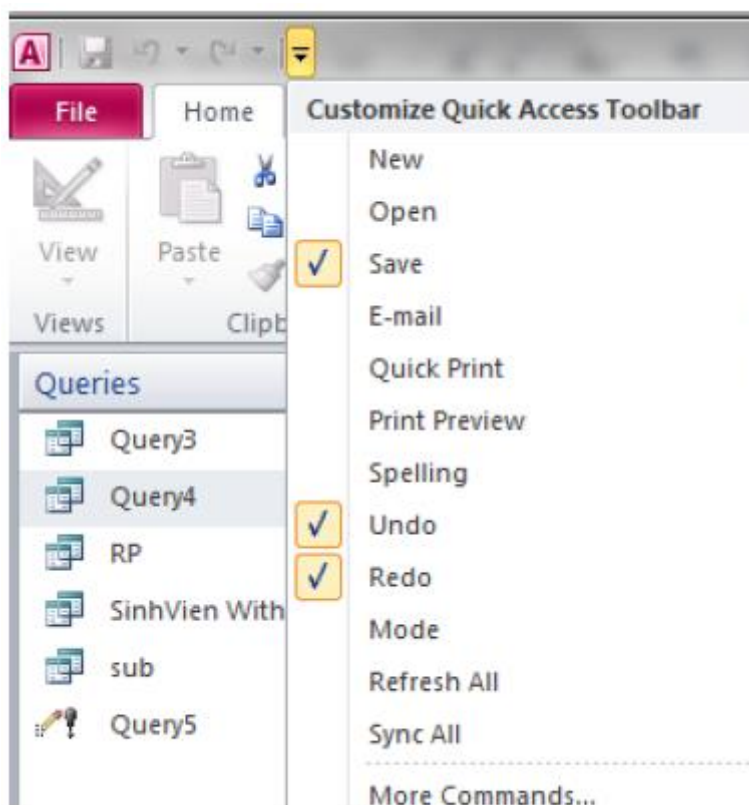
Hình 1.4. Giao diện của Access 2010 khi khởi động

- Các thành phần trong cửa sổ khởi động:

+ Thanh Quick Access:

Thanh công cụ Quick Access hiển thị bên trái của thanh tiêu đề, mặc định thanh Quick Access gồm các nút công cụ Save, Undo, and Redo.

Bên phải của Quick Access chứa nút Customize, khi cơ sở dữ liệu đang mở, nếu click nút Customize sẽ xuất hiện một menu giúp chúng ta chỉ định các nút lệnh hiển thị trên thanh Quick Access.



Hình 1.5. Thanh Quick Access

+ Vùng làm việc:

Khi khởi động Access, trong cửa sổ khởi động, mặc định tab File và lệnh New trong tab File được chọn, cửa sổ được chia thành 3 khung.

Khung bên trái gồm các lệnh trong tab File.

Khung giữa: chứa các loại tập tin cơ sở dữ liệu mà bạn có thể tạo mới.

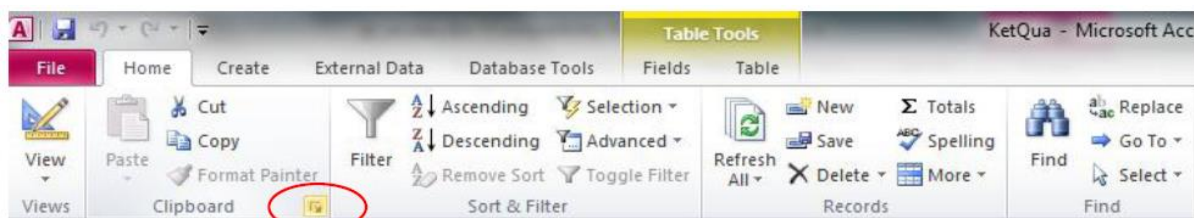
Khung bên phải: để nhập tên và chọn vị trí lưu tập tin mới tạo và thực thi lệnh tạo mới CSDL.

+ Thanh Ribbon:

Bên dưới thanh tiêu đề, Access 2010 hiển thị các nút lệnh trên một thanh dài được gọi là Ribbon, thanh Ribbon có thể chiếm nhiều không gian màn hình, chúng ta có thể thu nhỏ kích thước của Ribbon bằng cách click nút Minimize The Ribbon.

Thanh Ribbon được tạo bởi nhiều tab khác nhau, để truy cập vào một tab chúng ta có thể click trên nhãn của tab hoặc dùng phím tắt. Nhấn phím Alt hoặc F10 để hiển thị tên của các phím tắt của các tab.

Hầu hết các tab trên Ribbon được chia thành từng nhóm khác nhau, mỗi nhóm hiển thị tiêu đề con bên dưới của nhóm. Một số nhóm trong Ribbon có hiển thị nút, khi click nút này sẽ xuất hiện cửa sổ cho phép thiết lập các thuộc tính của nhóm tương ứng.



Hình 1.6. Thanh Ribbon

+ Cửa sổ Properties:

Cửa sổ Properties giúp chúng ta có thể hiệu chỉnh thuộc tính của đối tượng, tùy theo đối tượng đang được chọn mà cửa sổ thuộc tính sẽ chứa những thuộc tính tương ứng của đối tượng đó.

Để mở cửa sổ Properties, chúng ta chọn đối tượng muốn thay đổi thuộc tính, chọn tab Design hoặc nhấn tổ hợp phím Atl +Enter.

Cửa sổ Properties được chia thành năm nhóm:

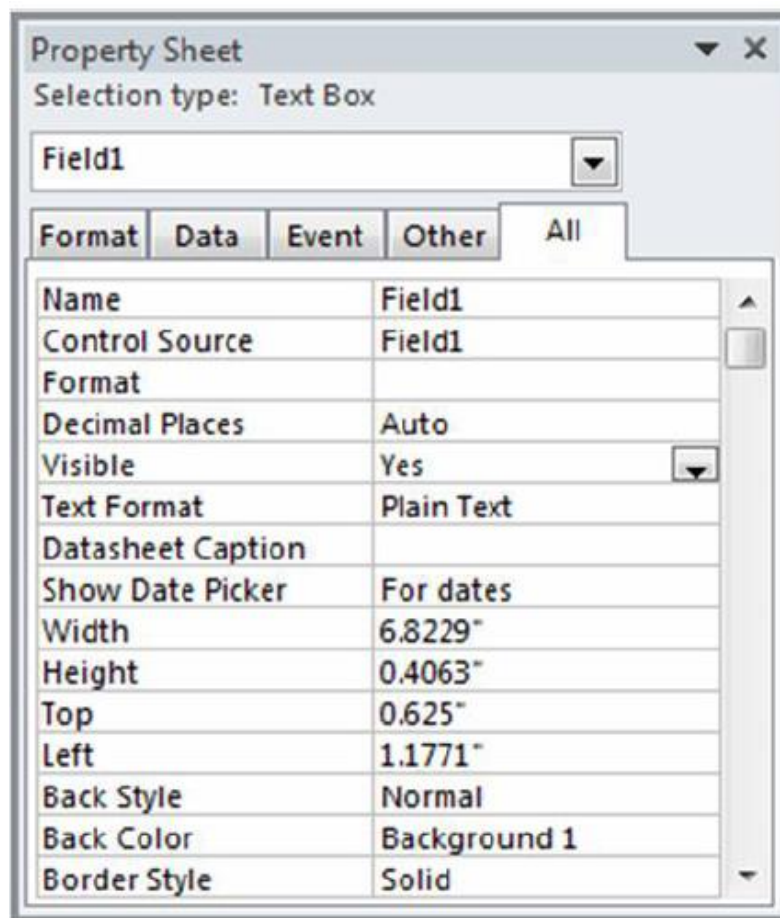
Format: Gồm các thuộc tính định dạng đối tượng

Data: Gồm các thuộc tính truy suất dữ liệu của đối tượng.

Event: Gồm các sự kiện (event) của đối tượng.

Other: Gồm các thuộc tính giúp bạn tạo hệ thống menu, toolbars,...

All: Gồm tất cả các thuộc tính trong bốn nhóm trên.



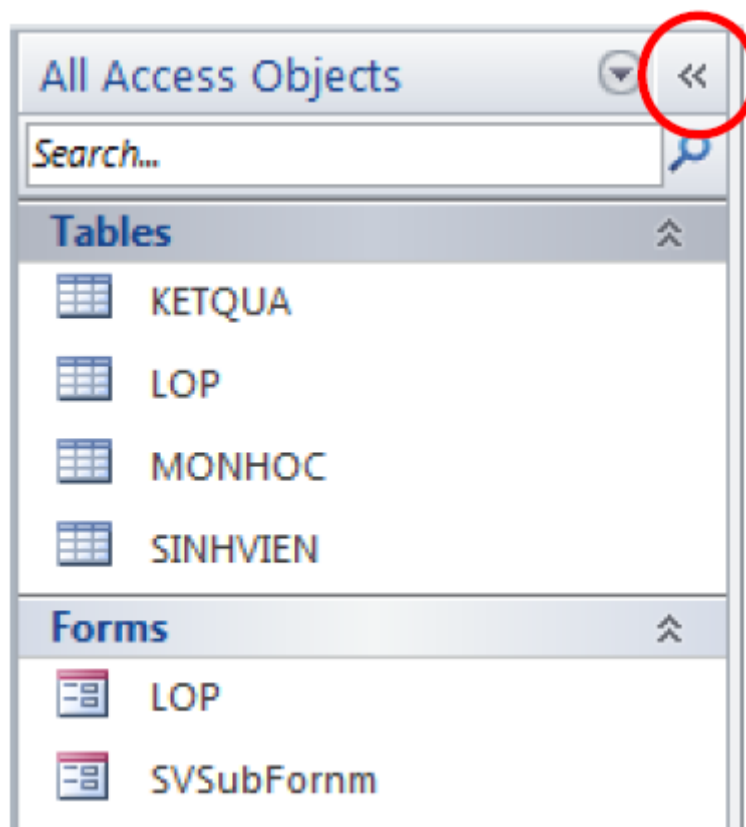
Hình 1.6. Cửa sổ Properties

+ Thanh Navigation Pane:

Navigation Pane là khung chứa nội dung chính của cơ sở dữ liệu. Từ khung Navigation Pane, bạn có thể mở bất kỳ Table, Query, Form, Report, Macro, hoặc module trong cơ sở dữ liệu bằng cách double click vào tên của đối tượng.

Bằng cách click phải vào tên của đối tượng trong Navigation Pane, chúng ta có thể thực hiện các thao tác với đối tượng như: đổi tên, sao chép, xoá, import, export một đối tượng...

Nhấn phím F11 hoặc click vào mũi tên kép ở góc trên bên phải của khung Navigation Pane để hiển thị hoặc ẩn khung Navigation Pane.



Hình 1.7. Thanh Navigation Pane

- Thoát khỏi access:

Có thể thoát khỏi Access bằng một trong các cách:

Cách 1: Chọn menu File → Exit

Cách 2: Nhấn tổ hợp phím Alt + F4;

Cách 3: Hoặc sử dụng nút Close trên cửa sổ Access đang mở.

- Mở một cơ sở dữ liệu:

Để mở một cơ sở dữ liệu đã có ta thực hiện một trong các cách sau:

Cách 1:

+ Tại cửa sổ khởi động, trong tab File → Open...

+ Chọn tập tin cơ sở dữ liệu cần mở → Open.

Cách 2: Double click vào tên tập tin cần mở.

- Các đối tượng trong cơ sở dữ liệu trong Access:

Cơ sở dữ liệu là một tập hợp những số liệu liên quan đến một mục đích quản lý, khai thác dữ liệu nào đó, CSDL trong Access là cơ sở dữ liệu quan hệ gồm các thành phần: Tables, Queries, Forms, Reports, Pages, Macros, Modules.



Hình 1.8. Các đối tượng trong cơ sở dữ liệu Access

+ **Bảng (Tables):**

Table là thành phần quan trọng nhất của tập tin cơ sở dữ liệu Access, dùng để lưu trữ dữ liệu. Do đó đây là đối tượng phải được tạo ra trước. Bên trong một bảng, dữ liệu được lưu thành nhiều cột và nhiều dòng.

Cột

	Mã sinh viên	Họ Sinh Viên	Tên Sinh Viên	Phá	Ngày Sinh	Địa chỉ	Điện Thoại	Mã Lớp
	A101	Lê Kiều	Oanh	Nữ	12/10/78	12 Nguyễn Trãi Q3	()7896746	CDTH1A
	A102	Trần văn	An	Nam	15/11/76	14 Đinh Tiên Hoàng Q1	()7613123	CDTH1A
	A103	Nguyễn thị Thù	Hiền	Nữ	12/02/78	13 Trần Bình Trọng QPN	()9123126	CDTH1A
	A104	Nguyễn Thành	Nhân	Nam	11/11/80	15 Nguyễn Trãi Q3	()7213546	CDTH1A

Dòng

Hình 1.9. Các cột và các dòng trong bảng dữ liệu về sinh viên

+ **Truy vấn (Queries):**

Query là công cụ để người sử dụng truy vấn thông tin và thực hiện các thao tác trên dữ liệu. Người sử dụng có thể sử dụng ngôn ngữ SQL hoặc công cụ QBE để thao tác trên dữ liệu.

+ **Biểu mẫu (Forms):**

Form là công cụ để thiết kế giao diện cho chương trình, dùng để cập nhật hoặc xem dữ liệu. Biểu mẫu giúp thân thiện hóa quá trình nhập, thêm, sửa, xóa và hiển thị dữ liệu.

+ **Báo cáo (Reports):**

Report là công cụ giúp người dùng tạo các kết xuất dữ liệu từ các bảng, sau đó định dạng và sắp xếp theo một khuôn dạng cho trước và có thể in ra màn hình hoặc máy in.

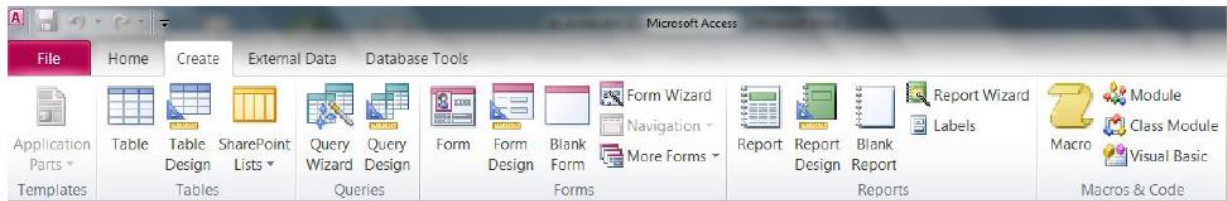
+ **Tập lệnh (Macros):**

Macro là một tập hợp các lệnh nhằm thực hiện một loạt các thao tác được qui định trước. Tập lệnh của Access có thể được xem là một công cụ lập trình đơn giản đáp ứng các tình huống cụ thể.

+ Bộ mã lệnh (Modules):

Là công cụ lập trình trong môi trường Access mà ngôn ngữ nền tảng của nó là ngôn ngữ Visual Basic for Application. Đây là một dạng tự động hóa chuyên sâu hơn tập lệnh, giúp tạo ra những hàm người dùng tự định nghĩa. Bộ mã lệnh thường dành cho các lập trình viên chuyên nghiệp.

Công cụ để tạo các đối tượng trong Access (thêm, thiết kế, xem nội dung các đối tượng) được tổ chức thành từng nhóm trong tab Create của thanh Ribbon.



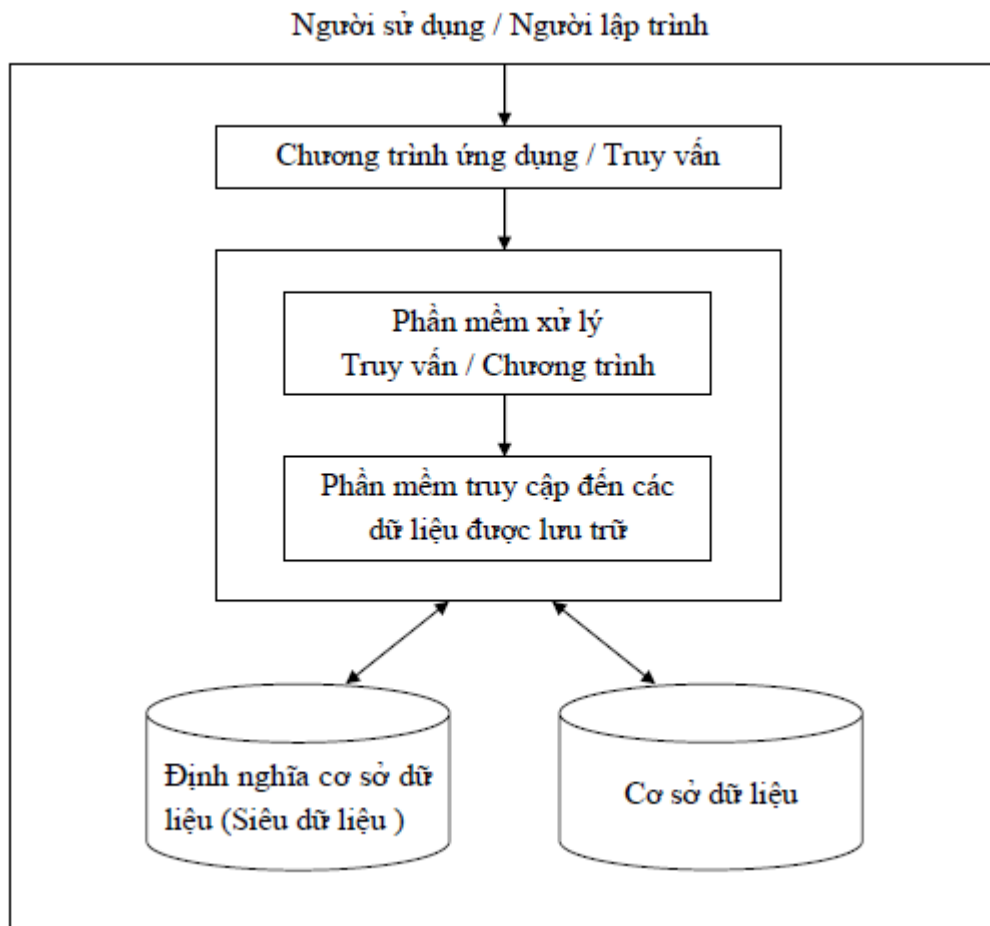
Hình 1.9. Thanh Ribbon cho phép tạo các đối tượng

1.2.3. Hệ cơ sở dữ liệu

1.2.3.1. Định nghĩa và phân loại

Hệ CSDL là một hệ thống gồm có bốn thành phần:

- Cơ sở dữ liệu hợp nhất: CSDL của hệ có hai tính chất là tối thiểu hoá dư thừa và tính chia sẻ.
- Người sử dụng.
- Phần mềm hệ quản trị CSDL.
- Phần cứng: Phần cứng của hệ bao gồm các thiết bị nhớ thứ cấp được sử dụng để lưu trữ dữ liệu.



Hình 1.4. Môi trường hệ cơ sở dữ liệu

Có thể chia hệ CSDL thành hai loại: hệ CSDL tập trung và hệ CSDL phân tán.

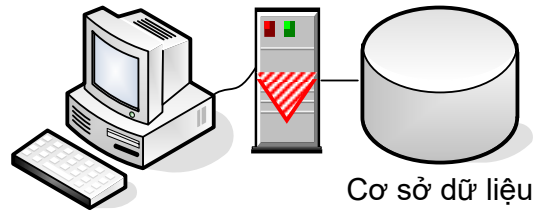
a. Các hệ CSDL tập trung

Đối với một hệ CSDL tập trung, tất cả các dữ liệu được định vị tại một trạm đơn lẻ, và NSD tại các trạm từ xa có thể truy nhập CSDL thông qua các công cụ truyền thông dữ liệu.

Các hệ CSDL tập trung cung cấp một sự kiểm soát lớn hơn đối với việc truy nhập và cập nhật dữ liệu so với hệ CSDL phân tán, nhưng chúng có thể bị mắc lỗi nhiều hơn do chúng phụ thuộc vào tính sẵn sàng của tài nguyên.

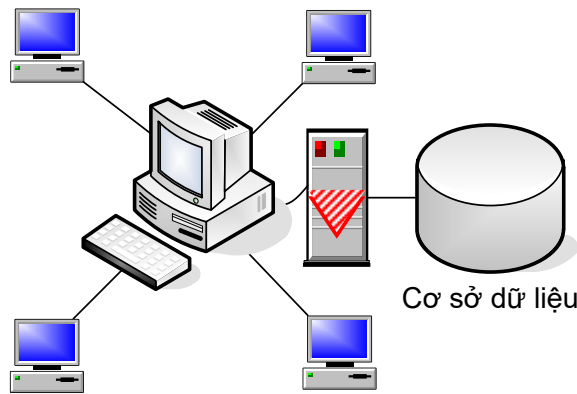
Có 3 kiểu hệ CSDL tập trung: hệ CSDL cá nhân, hệ CSDL trung tâm, hệ CSDL khách/chủ.

- Hệ CSDL cá nhân: Hệ CSDL cá nhân thường có một NSD đơn lẻ, vừa thiết kế, tạo lập CSDL, cập nhật CSDL và bảo trì dữ liệu, đồng thời cũng là người lập và hiển thị báo cáo. Nói cách khác, NSD vừa là người quản trị CSDL, vừa là người viết chương trình ứng dụng, đồng thời cũng là người sử dụng cuối của hệ. Thông thường, hệ CSDL này hỗ trợ một ứng dụng hay một số giới hạn các ứng dụng đơn lẻ, khó chia sẻ dữ liệu cho nhiều ứng dụng khác nhau.



Hình 1.5. Hệ cơ sở dữ liệu cá nhân

- **Hệ CSDL trung tâm:** Dữ liệu mà các ứng dụng có thể truy cập được lưu trữ trên một máy tính trung tâm. NSD từ xa có thể truy cập CSDL này thông qua các thiết bị đầu cuối và các kết nối truyền thông dữ liệu. Tùy theo quy mô của tổ chức, máy tính trung tâm này thường là một dàn máy hay chỉ một máy vi tính. Các hệ CSDL trung tâm thường lưu trữ các CSDL tích hợp rất lớn và được nhiều người sử dụng truy nhập. Việc sử dụng thường có cường độ lớn với hàng trăm giao dịch trong một giây đang được xử lý (một số hệ thống có thể hỗ trợ hàng nghìn giao dịch trong một giây). Các ứng dụng điển hình bao gồm các hệ thống đăng ký đặt chỗ máy bay, các hệ thống thông tin của các cơ quan tài chính và các công ty phát triển nhanh.



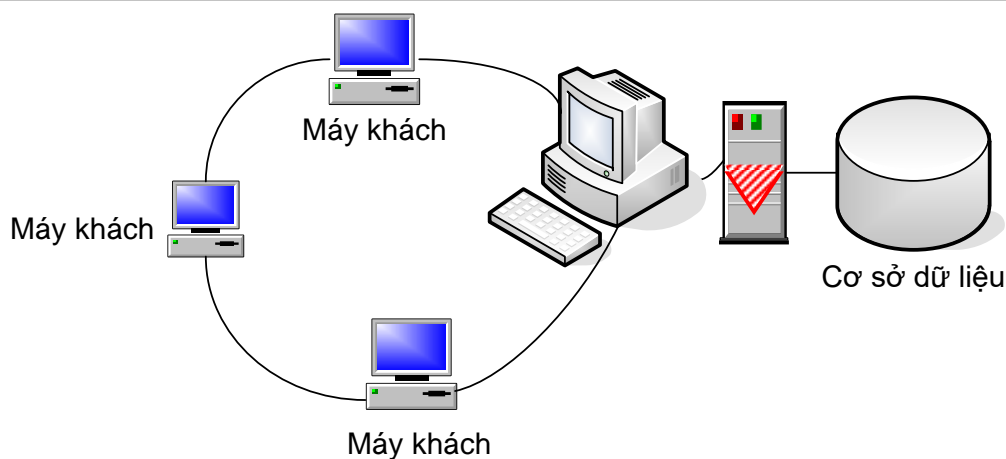
Hình 1.6. Hệ cơ sở dữ liệu trung tâm

- **Hệ CSDL khách/chủ:** Các máy tính trung tâm thường rất đắt so với các máy vi tính nhỏ và các máy trạm. Do vậy, nhiều tổ chức đã giảm kích thước các ứng dụng để có thể cài đặt trên các máy tính nhỏ với giá thành hiệu quả hơn. Máy khách sẽ yêu cầu các dịch vụ từ một hay nhiều máy chủ. Thành phần quản trị CSDL trên máy chủ tiếp nhận các yêu cầu, sau đó cung cấp các dịch vụ quản lý tập tin hay CSDL, quản lý truyền thông... mà máy khách yêu cầu.

Đối với hệ CSDL trung tâm, máy tính trung tâm chỉ đóng vai trò như một ổ cứng chứa dữ liệu, khi NSD cần dữ liệu thì phải truy cập vào thông qua các thiết bị đầu cuối. Trong khi đối với hệ CSDL khách/chủ thì sẽ có sự lưu trữ và quản lý dữ liệu ở máy chủ và máy chủ chỉ cung cấp dữ liệu theo yêu cầu của máy khách.

Ưu điểm của hệ CSDL khách/chủ so với các hệ CSDL khác là:

- + Nâng cao khả năng thực hiện: Các CPU ở máy khách và máy chủ hoạt động riêng rẽ và cùng hoạt động song song. Các máy tính thực hiện nhiệm vụ riêng của nó.
- + Chi phí cho phần cứng có thể giảm do chỉ cần máy chủ đủ mạnh để lưu trữ và quản lý CSDL.
- + Có thể thêm máy khách dễ dàng.



Hình 1.7. Hệ cơ sở dữ liệu khách/chủ

b. Các hệ CSDL phân tán

Ngày nay, nhiều tổ chức phân bố trên nhiều vị trí địa lý khác nhau như các thành phố khác nhau hay các quốc gia khác nhau. Trong những trường hợp như vậy, việc xây dựng các hệ CSDL tập trung đối với các tổ chức này thường là không thực tế và không kinh tế.

Một CSDL phân tán là tập hợp dữ liệu logic thuộc về cùng một hệ thống nhưng trải rộng ra nhiều điểm trên mạng máy tính (Hình 1.8).

Có 2 kiểu hệ CSDL phân tán là: hệ CSDL phân tán thuần nhất, hệ CSDL phân tán không thuần nhất.

- Hệ CSDL phân tán thuần nhất:

Hệ CSDL phân tán thuần nhất có các đặc điểm sau:

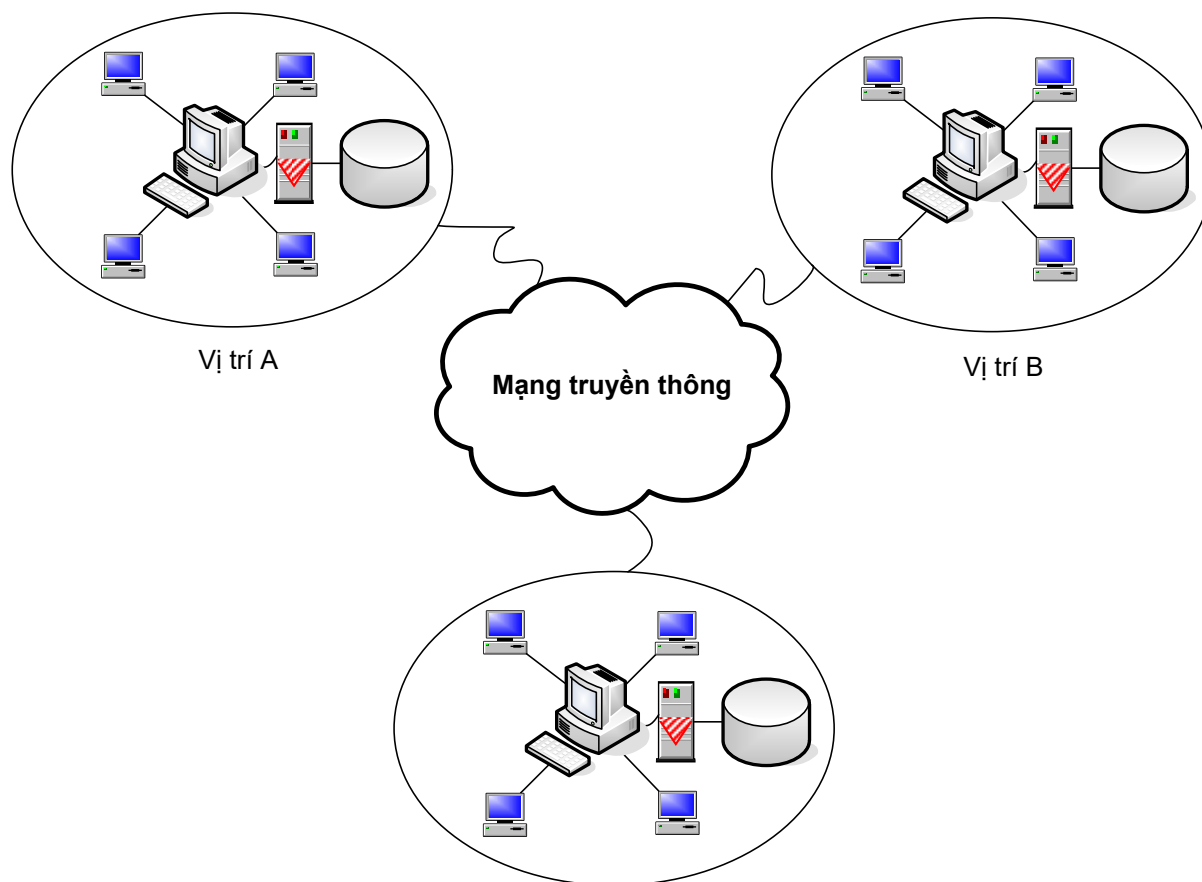
- + Các hệ điều hành máy tính tại mỗi vị trí địa lý là như nhau hay ít nhất chúng có khả năng tương thích cao.
- + Các mô hình dữ liệu được sử dụng tại mỗi một vị trí địa lý là như nhau (mô hình quan hệ được sử dụng chung nhất đối với các hệ CSDL phân tán ngày nay)
- + Các hệ quản trị CSDL được sử dụng tại mỗi vị trí địa lý là như nhau hay ít nhất chúng có khả năng tương thích cao.
- + Dữ liệu tại các vị trí khác nhau có các định nghĩa và định dạng chung.

- Hệ CSDL phân tán không thuần nhất:

Đối với hệ CSDL phân tán không thuần nhất, các máy tính khác nhau và các hệ điều hành khác nhau có thể được sử dụng tại mỗi vị trí. Các mô hình dữ liệu khác nhau và các hệ quản trị CSDL khác nhau cũng có thể được lựa chọn sử dụng. Phức tạp hơn, dữ liệu trên các vị trí thường không tương thích. Các mâu thuẫn diễn hành bao gồm các khác biệt về cú pháp (sự biểu diễn khác nhau của các khoản mục dữ liệu tại hai vị trí) và các khác biệt về ngữ nghĩa (các ngữ nghĩa khác nhau đối với cùng một khoản mục dữ liệu tại các vị trí khác nhau)

Đến một thời điểm nào đó, NSD tại các vị trí khác nhau sẽ nhận ra rằng họ cần chia sẻ dữ liệu cho dù có sự không tương thích. Một giải pháp là phát triển một CSDL mới hoàn toàn dựa trên việc hợp nhất tất cả các hệ đang tồn tại, tuy nhiên đây là một giải pháp không dễ thực hiện về mặt kỹ thuật hay kinh tế. Thay vào đó, đôi khi các

CSDL được kết nối với nhau và kết quả là tạo ra một tập các CSDL không thuần nhất (còn được gọi là các CSDL liên hiệp). Một hệ thống như vậy nói chung hạn chế các kiểu xử lý mà NSD có thể thực hiện, chẳng hạn, một NSD tại một vị trí có thể đọc nhưng không thể cập nhật dữ liệu tại một vị trí khác.



Hình 1.8. Các hệ CSDL phân tán

1.2.3.2. Sự cần thiết của các hệ cơ sở dữ liệu

Tổ chức lưu trữ dữ liệu theo lý thuyết cơ sở dữ liệu có những ưu điểm:

- **Giảm bớt dư thừa dữ liệu trong lưu trữ:** Trong các ứng dụng lập trình truyền thống, phương pháp tổ chức lưu trữ dữ liệu vừa tốn kém, lãng phí bộ nhớ và các thiết bị lưu trữ, vừa dư thừa thông tin lưu trữ. Nhiều chương trình ứng dụng khác nhau cùng xử lý trên các dữ liệu như nhau, dẫn đến sự dư thừa đáng kể về dữ liệu. Chẳng hạn, trong các bài toán nghiệp vụ quản lý "Cước thuê bao điện thoại" và "Doanh thu & sản lượng", tương ứng với mỗi một chương trình là một hay nhiều tệp dữ liệu được lưu trữ riêng biệt, độc lập với nhau. Trong cả 2 chương trình cùng xử lý một số thuộc tính của một cuộc đàm thoại như "số máy gọi đi", "số máy gọi đến", "hướng cuộc gọi", "thời gian bắt đầu" và "thời gian kết thúc"...Nhiều thuộc tính được mô tả và lưu trữ nhiều lần độc lập với nhau. Nếu tổ chức lưu trữ theo lý thuyết CSDL thì có thể hợp nhất các tệp lưu trữ của các bài toán trên, các chương trình ứng dụng có thể cùng chia sẻ tài nguyên trên cùng một hệ CSDL.

- **Tổ chức lưu trữ dữ liệu theo lý thuyết CSDL sẽ tránh được sự không nhất quán trong lưu trữ dữ liệu và bảo đảm được tính toàn vẹn của dữ liệu:** Nếu một thuộc tính được mô tả trong nhiều tập tin dữ liệu khác nhau và lặp lại nhiều lần trong các bản ghi, khi thực hiện việc cập nhật, bổ sung sẽ không sửa hết nội dung các mục đó. Nếu dữ liệu càng nhiều thì sự sai sót khi cập nhật, bổ sung càng lớn. Khả năng xuất

hiện mâu thuẫn, không nhất quán thông tin càng nhiều, dẫn đến không nhất quán dữ liệu trong lưu trữ. Tất yếu kéo theo sự dị thường thông tin, thừa, thiếu và mâu thuẫn thông tin.

Thông thường, trong một thực thể, giữa các thuộc tính có mối quan hệ ràng buộc lẫn nhau, tác động ảnh hưởng lẫn nhau. Cước của một cuộc đàm thoại phụ thuộc vào khoảng cách và thời gian cuộc gọi, tức là phụ thuộc hàm vào các thuộc tính máy gọi đi, máy gọi đến, thời gian bắt đầu và thời gian kết thúc cuộc gọi. Các trình ứng dụng khác nhau cùng xử lý cước đàm thoại trên các thực thể lưu trữ tương ứng khác nhau chưa hẳn cho cùng một kết quả về sản lượng phút và doanh thu. Điều này lý giải tại sao trong một doanh nghiệp, cùng xử lý trên các chỉ tiêu quản lý mà số liệu báo cáo của các phòng ban, các công ty con lại cho các kết quả khác nhau, thậm chí còn trái ngược nhau. Như vậy, có thể khẳng định, nếu dữ liệu không tổ chức theo lý thuyết cơ sở dữ liệu, tất yếu không thể phản ánh thế giới hiện thực dữ liệu, không phản ánh đúng bản chất vận động của dữ liệu.

Sự không nhất quán dữ liệu trong lưu trữ làm cho dữ liệu mất đi tính toàn vẹn của nó. Tính toàn vẹn dữ liệu đảm bảo cho sự lưu trữ dữ liệu luôn luôn đúng. Không thể có mã vùng ngoài quy định của cơ quan quản lý, hoặc ngày sinh của một nhân viên không thể xảy ra sau ngày tốt nghiệp ra trường của nhân viên đó...

- **Tổ chức lưu trữ dữ liệu theo lý thuyết CSDL có thể triển khai đồng thời nhiều ứng dụng trên cùng một CSDL:** Điều này có nghĩa là các ứng dụng không chỉ chia sẻ chung tài nguyên dữ liệu mà còn trên cùng một CSDL có thể triển khai đồng thời nhiều ứng dụng khác nhau tại các thiết bị đầu cuối khác nhau.

- **Tổ chức dữ liệu theo lý thuyết cơ sở dữ liệu sẽ thống nhất các tiêu chuẩn, thủ tục và các biện pháp bảo vệ, an toàn dữ liệu:** Các hệ CSDL sẽ được quản lý tập trung bởi một cá nhân hay một nhóm quản trị CSDL, bằng các hệ quản trị CSDL. Người quản trị CSDL có thể áp dụng thống nhất các tiêu chuẩn, quy định, thủ tục chung như quy định thống nhất về mẫu biểu báo cáo, thời gian bổ sung, cập nhật dữ liệu. Điều này làm dễ dàng cho công việc bảo trì dữ liệu. Người quản trị CSDL có thể bảo đảm việc truy nhập tới CSDL, có thể kiểm tra, kiểm soát các quyền truy nhập của người sử dụng. Ngăn chặn các truy nhập trái phép, sai quy định từ trong ra hoặc từ ngoài vào...

1.3. Đối tượng sử dụng

- **Người sử dụng không chuyên về lĩnh vực tin học và CSDL:** Là những người không cần có kiến thức cao về tin học, dùng các công cụ để khai thác CSDL.

- **Người phát triển hệ thống:** Là những người có thể khai thác CSDL để xây dựng các ứng dụng phục vụ cho các mục đích khác nhau.

- **Người thiết kế, quản trị CSDL:** Đây là những người hiểu biết về tin học, về 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 phải nắm rõ các vấn đề kỹ thuật về CSDL để có thể phục hồi dữ liệu khi có sự cố. Họ là những người cấp quyền hạn khai thác CSDL, do vậy họ có thể giải quyết được các vấn đề tranh chấp dữ liệu nếu có.

1.4. Ưu và nhược điểm của hệ CSDL

1.4.1. Ưu điểm

- Giảm sự trùng lặp thông tin xuống mức thấp nhất và do đó tránh dư thừa dữ liệu, bảo đảm được tính nhất quán và toàn vẹn dữ liệu.
- Quản lý được khối lượng dữ liệu lớn.
- Có khả năng chia sẻ thông tin cho nhiều NSD và nhiều ứng dụng khác nhau.
- Tăng cường tính bảo mật.
- Các CSDL tuân theo các tiêu chuẩn nên dễ tích hợp.
- Việc kết hợp các dữ liệu vào cùng một CSDL có thể tiết kiệm không gian lưu trữ, thời gian và công sức nhập dữ liệu.
- Đảm bảo dữ liệu có thể được truy xuất theo nhiều cách khác nhau.
- Hệ quản trị CSDL cung cấp một giao diện dễ sử dụng, hỗ trợ sẵn nhiều chức năng nên thuận tiện hơn cho việc phát triển các ứng dụng.
- Tăng cường tính độc lập giữa dữ liệu và chương trình
- Hạn chế tranh chấp dữ liệu khi có nhiều người dùng đồng thời.
- Có cơ chế sao lưu và phục hồi dữ liệu khi có sự cố.

1.4.2. Nhược điểm

- Sử dụng CSDL sẽ phức tạp hơn so với sử dụng tập tin thông thường.
- Kích thước lớn vì quản lý dữ liệu tập trung.
- Phải chi phí cho hệ quản trị CSDL và các phần cứng.
- CSDL lưu trữ một khối lượng dữ liệu lớn có thể làm cho các ứng dụng hoạt động chậm.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 1

- 1.1.** Cho một ví dụ và phân tích các ưu điểm, nhược điểm của một hệ xử lý tập tin truyền thống.
- 1.2.** Tại sao người ta lại sử dụng hệ CSDL để thay thế hệ xử lý tập tin truyền thống?
- 1.3.** Hãy nêu ra các điểm khác nhau chính giữa một hệ xử lý tập tin truyền thống và một hệ CSDL.
- 1.4.** Phân biệt các thuật ngữ sau: dữ liệu, thông tin, cơ sở dữ liệu, hệ quản trị cơ sở dữ liệu, hệ cơ sở dữ liệu.
- 1.5.** Hãy nêu các nhóm đối tượng sử dụng cơ sở dữ liệu.
- 1.6.** Hãy phân tích các nhóm đối tượng sử dụng CSDL tại một trường học. Giả sử trường học bao gồm cán bộ quản lý, giáo viên, các chuyên viên quản trị CSDL, các chuyên viên phát triển phần mềm ứng dụng cho trường, học sinh.
- 1.7.** Nêu các ưu và nhược điểm của hệ CSDL.
- 1.8.** CSDL được sử dụng rộng rãi trong cuộc sống, hãy nêu ví dụ về một số lĩnh vực ứng dụng CSDL, nêu rõ ứng dụng CSDL vào việc gì.

CHƯƠNG 2

MÔI TRƯỜNG CƠ SỞ DỮ LIỆU

Chương 1 đã giới thiệu hệ CSDL là gì và tại sao cần phải sử dụng các hệ CSDL. Mục đích của chương 2 sẽ tìm hiểu sâu hơn nữa về môi trường CSDL.

Nội dung chương này sẽ trình bày về kiến trúc ANSI-PARC 3 mức của CSDL, theo đó, CSDL được chia thành mức vật lý (mức trong), mức logic (mức khái niệm) và mức khung nhìn (mức ngoài). Kiến trúc ba mức của hệ CSDL thể hiện các mức trừu tượng khi nhìn nhận dữ liệu, điều đó cho chúng ta thấy rằng: *mục đích của một hệ CSDL là giải phóng đa số người dùng khỏi sự quan tâm về lưu trữ và bảo trì dữ liệu.*

Tương ứng với các mức trừu tượng đó, chương này đề cập đến các khái niệm về lược đồ ngoài, lược đồ khái niệm, lược đồ trong, dùng để mô tả ba mức thiết kế một CSDL.

Chương này cũng giới thiệu các ngôn ngữ dùng để định nghĩa và thao tác dữ liệu, giới thiệu về các mô hình dữ liệu và giới thiệu về hệ quản trị CSDL đa người dùng.

2.1. Kiến trúc ANSI-PARC 3-mức

Theo kiến trúc ANSI-PARC (American National Standards Institute - Planning and Requirements Committee: Viện tiêu chuẩn quốc gia Mỹ - uỷ ban nhu cầu và kế hoạch Mỹ), một CSDL có 3 mức biểu diễn: mức vật lý (mức trong), mức logic (mức khái niệm) và mức khung nhìn (mức ngoài).

2.1.1. Mức vật lý (mức trong)

Mức vật lý mô tả dữ liệu được lưu trữ như thế nào trong CSDL.

Đây là mức thể hiện các cài đặt có tính chất vật lý của CSDL để đạt được sự tối ưu khi thực hiện các thao tác tìm kiếm và lưu trữ, và để tận dụng được các vùng nhớ còn trống.

Mức vật lý cũng phản ánh cấu trúc dữ liệu, các tổ chức tập tin được dùng cho việc lưu trữ dữ liệu trên các thiết bị nhớ thứ cấp. Điều đó cũng có nghĩa là mức này tiếp xúc với các phương thức truy nhập của hệ điều hành để đặt dữ liệu vào các thiết bị nhớ, xây dựng các tập chỉ mục, truy xuất dữ liệu..., liên quan đến các vấn đề như sự cấp phát vùng nhớ cho dữ liệu và chỉ mục, các mô tả bản ghi để lưu trữ, các kỹ thuật nén dữ liệu và giải mã dữ liệu.

Những người hiểu và làm việc với CSDL tại mức này là người quản trị CSDL (Database Administrator), những NSD chuyên môn.

Ví dụ: Biểu diễn mức vật lý đơn giản các dữ liệu về nhân viên của một công ty: mã nhân viên - kiểu số nguyên, mã chi nhánh mà nhân viên đó làm việc - kiểu số nguyên, họ đệm - kiểu chuỗi, tên - kiểu chuỗi, ngày sinh - kiểu ngày, lương - kiểu số thực, và một con trỏ đến bản ghi tiếp theo.

2.1.2. Mức logic (mức khái niệm)

Mức logic mô tả những dữ liệu nào được lưu trữ trong CSDL và có những mối quan hệ nào giữa chúng.

Mức logic biểu diễn các thực thể, các thuộc tính, và các mối quan hệ giữa các thực thể đó. Mức logic cũng cho thấy các ràng buộc trên dữ liệu, các thông tin về ngữ nghĩa

của dữ liệu, các thông tin về an ninh và toàn vẹn của dữ liệu. Tuy nhiên mức logic chỉ quan tâm đến cái gì được lưu trữ trong CSDL chứ không quan tâm đến cách thức để lưu trữ.

Từ môi trường thế giới thực, xuất phát từ nhu cầu quản lý, việc xác định các loại thông tin cần lưu trữ và các mối quan hệ giữa các thông tin đó như thế nào chính là công việc ở mức logic.

Ví dụ: CSDL quản lý các phòng ban và nhân viên của một công ty có thể biểu diễn mức logic như sau:

Môi trường (thế giới thực) của công ty ở đây gồm có các phòng ban (Department), mỗi phòng ban có một tên gọi khác nhau (Department_Name), một địa chỉ trụ sở chính (Location), các số điện thoại (Telephone) để liên lạc, có một người làm trưởng phòng ban (Manager), hàng năm được cấp một khoản kinh phí để hoạt động (Expense_Budget), và phải đạt được một doanh thu (Revenue_Budget). Để tránh viết tên phòng ban dài dễ dẫn đến viết sai, người ta thường đặt cho mỗi phòng ban một giá trị số gọi là số hiệu phòng ban (Department_Number) và sử dụng số hiệu này để xác định tên và các thông tin khác của nó.

Công ty có một số công việc có thể sắp xếp cho các nhân viên trong công ty. Để thuận lợi cho việc theo dõi công việc cũng như trong công tác tuyển chọn nhân viên mới, người ta lập thành một bảng các công việc (Jobs) gồm các thông tin: tên tắt công việc (Job), tên công việc (Job_Name), mức lương tối thiểu (Min_Salary) và tối đa (Max_Salary) của công việc này và cho biết công việc này cần có người lãnh đạo không. Một công việc có thể có nhiều người cùng làm.

Mỗi phòng ban có thể có từ một đến nhiều nhân viên (Employee). Mỗi nhân viên có một tên gọi (Name), một công việc làm (Job), một khoản tiền lương hàng tháng (Salary), số hiệu phòng ban mà anh ta đang công tác. Nếu muốn, người ta có thể theo dõi thêm các thông tin khác như ngày sinh (BirthDay), giới tính (Sex)... Để tránh viết tên nhân viên dài dễ dẫn đến sai sót, mỗi nhân viên có thể được gán cho một con số duy nhất, gọi là mã số nhân viên (EmpNo).

Nếu yêu cầu quản lý của công ty chỉ dừng ở việc theo dõi danh sách nhân viên trong từng phòng ban cùng các công việc của công ty thì cần 3 loại thông tin: Phòng ban (Department), Công việc (Jobs) và Nhân viên (Employee) với các thông tin như trên là đủ.

Nếu công ty có thêm yêu cầu quản lý cả quá trình tuyển dụng và nâng lương thì cần có thêm một (hoặc một số) loại thông tin về quá trình công tác: mã số nhân viên, lần thay đổi, thời gian bắt đầu và kết thúc sự thay đổi, mức lương...

2.1.3. Mức khung nhìn (mức ngoài)

Mức khung nhìn mô tả phần CSDL liên quan đến NSD hay các chương trình ứng dụng.

Mỗi NSD hay mỗi chương trình ứng dụng có thể được "nhìn" CSDL theo một góc độ khác nhau. Có thể "nhìn" thấy toàn bộ hay chỉ một phần, hoặc chỉ là các thông tin tổng hợp từ CSDL hiện có. NSD hay chương trình ứng dụng có thể hoàn toàn không được biết về cấu trúc tổ chức lưu trữ thông tin trong CSDL, thậm chí ngay cả tên gọi của các loại dữ liệu hay tên gọi của các thuộc tính. Họ chỉ có thể làm việc trên phần

CSDL theo cách "nhìn" do người quản trị hay chương trình ứng dụng quy định, gọi là khung nhìn (View).

Ví dụ:

Với CSDL quản lý các phòng ban và nhân viên của một công ty như Ví dụ 2.2, có thể thêm các biểu diễn mức logic như sau:

Phòng Tổ chức nhân sự giờ đây còn quản lý thêm cả các thông tin chi tiết trong lý lịch của nhân viên trong công ty: quá trình đào tạo chuyên môn kỹ thuật - kinh tế - chính trị - quản lý nhà nước, quá trình được khen thưởng, các lần bị kỷ luật, quá trình hoạt động Cách mạng bị địch bắt - bị tù đầy, quá trình công tác, quá trình nâng lương, sơ lược tiểu sử cha mẹ - anh chị em ruột - vợ chồng - con ...

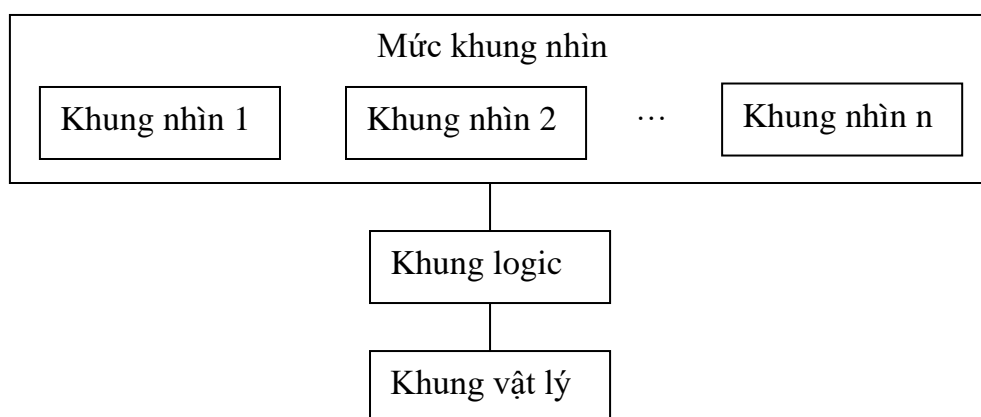
Như vậy, ở mức khung nhìn:

Phòng Kế toán có thể chỉ “nhìn” thấy danh sách nhân viên đang làm các công việc cụ thể trong từng Phòng ban với các mức lương thỏa thuận, mà không được thấy lý lịch của các nhân viên.

Lãnh đạo công ty có thể chỉ cần “nhìn” thấy số lượng nhân viên, tổng số lương phải trả và ai là người lãnh đạo của từng Phòng ban.

Ngay cả những người trong Phòng Tổ chức nhân sự cũng có thể có người được xem lý lịch của tất cả cán bộ, công nhân viên của công ty, nhưng cũng có thể có người chỉ được xem lý lịch của những cán bộ, công nhân viên với mức lương từ n đồng trở xuống (n là số bất kỳ)...

Như vậy, cấu trúc CSDL vật lý và mức logic thì chỉ có một, nhưng mức khung nhìn thì có thể có rất nhiều cấu trúc ngoài tương ứng.



Hình 2.1. Kiến trúc ANSI-PARC 3-mức

Có thể thấy mục đích của kiến trúc ba mức nêu trên chính là sự tách biệt quan niệm CSDL của nhiều NSD với những chi tiết biểu diễn về vật lý của CSDL. Điều đó dẫn đến những ưu điểm sau:

- Đối với một CSDL, mỗi người dùng có một khung nhìn riêng của mình. Họ có thể thay đổi khung nhìn của họ và sự thay đổi này không làm ảnh hưởng đến những khung nhìn dữ liệu của những người dùng khác đang dùng chung CSDL này.

- Những tương tác của người dùng với CSDL không phụ thuộc vào những vấn đề chi tiết trong lưu trữ dữ liệu (chẳng hạn vấn đề chỉ mục hóa hay bảng băm).

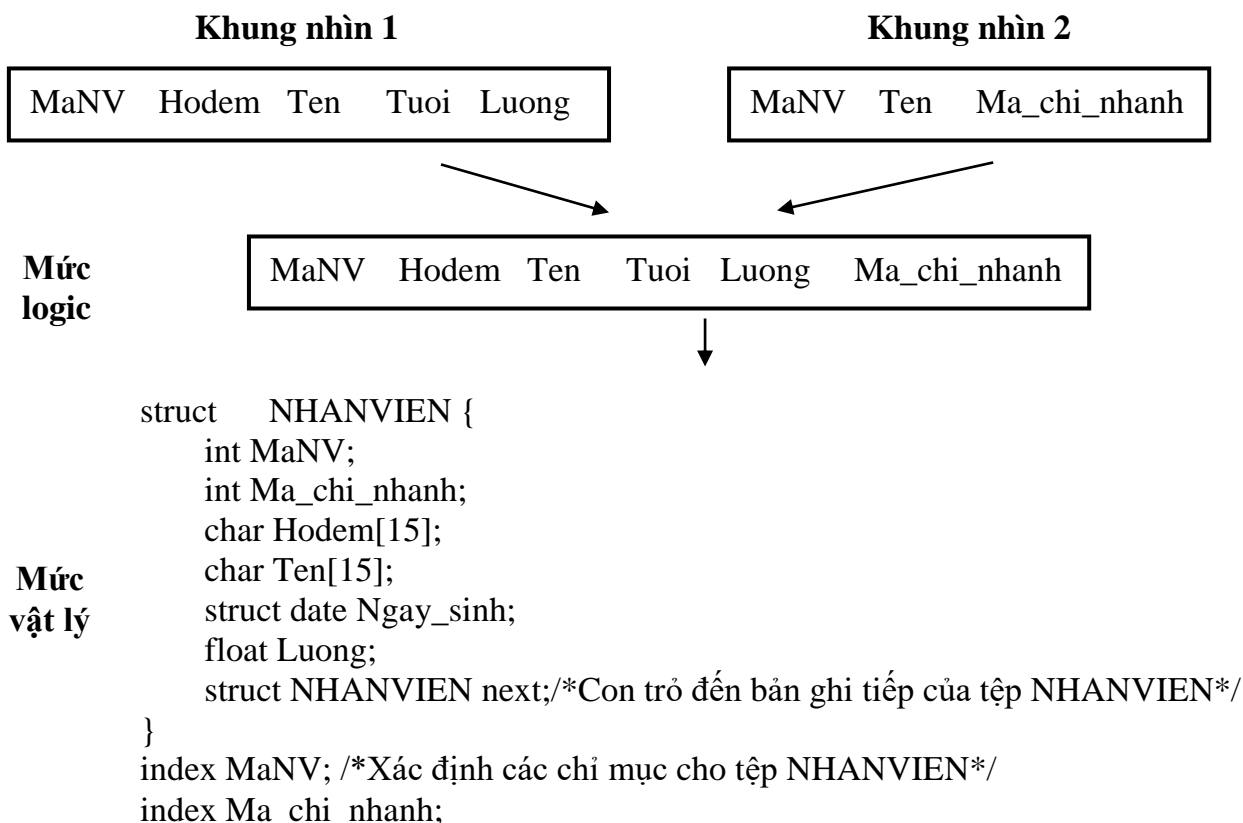
- Người quản trị CSDL có thể thay đổi cấu trúc lưu trữ của CSDL mà không làm ảnh hưởng đến những khung nhìn của NSD.
- Những thay đổi về khía cạnh vật lý trong lưu trữ, chẳng hạn như thay một thiết bị nhỏ thứ cấp mới, có thể không làm ảnh hưởng đến cấu trúc bên trong của CSDL.
- Người quản trị CSDL có thể thay đổi cấu trúc tổng quát hay cấu trúc khái niệm của CSDL mà không làm ảnh hưởng đến tất cả người dùng.

2.1.4. *Lược đồ và thể hiện của CSDL*

Toàn bộ mô tả CSDL được gọi là lược đồ CSDL (database schema).

Tương ứng với ba mức truy xuất dữ liệu nói trên có ba loại lược đồ:

- Ở mức cao nhất (khung nhìn) chúng ta có nhiều lược đồ ngoài (hay còn gọi là lược đồ con) cho những cách nhìn dữ liệu khác nhau của những NSD khác nhau.
- Ở mức logic chúng ta có lược đồ khái niệm (hay còn gọi là lược đồ logic).
- Ở mức thấp nhất (vật lý) chúng ta có lược đồ trong (hay còn gọi là lược đồ vật lý).



Hình 2.2. Ví dụ về lược đồ CSDL

Thông thường các hệ CSDL hỗ trợ một lược đồ trong, một lược đồ khái niệm và nhiều lược đồ ngoài.

Cần phải phân biệt lược đồ CSDL với bản thân CSDL. *Lược đồ được xác định trong quá trình thiết kế CSDL và người ta không muốn nó thay đổi thường xuyên, trong khi đó bản thân CSDL sẽ thay đổi theo thời gian do dữ liệu được thêm vào, xóa*

hay sửa đổi. Hình 2.2 là ví dụ về lược đồ CSDL, gồm các mức vật lý, logic và khung nhìn.

Toàn bộ dữ liệu trong CSDL tại một thời điểm nhất định được gọi là một thể hiện của CSDL (database instance).

Bảng 2.1. Thể hiện CSDL gồm tên khách hàng, số chứng minh thư, nơi ở, tài khoản

TenKH	SoCMT	NoiO	TaiKhoan
Trần Văn Ban	031803491	Hà Nội	A-101
Nguyễn Thị Giao	044803581	Hải Phòng	A-215
Hoàng Thị Kim Dung	037120582	Hà Nội	A-102
Trần Thị Lan Anh	035671241	Bắc Ninh	A-305

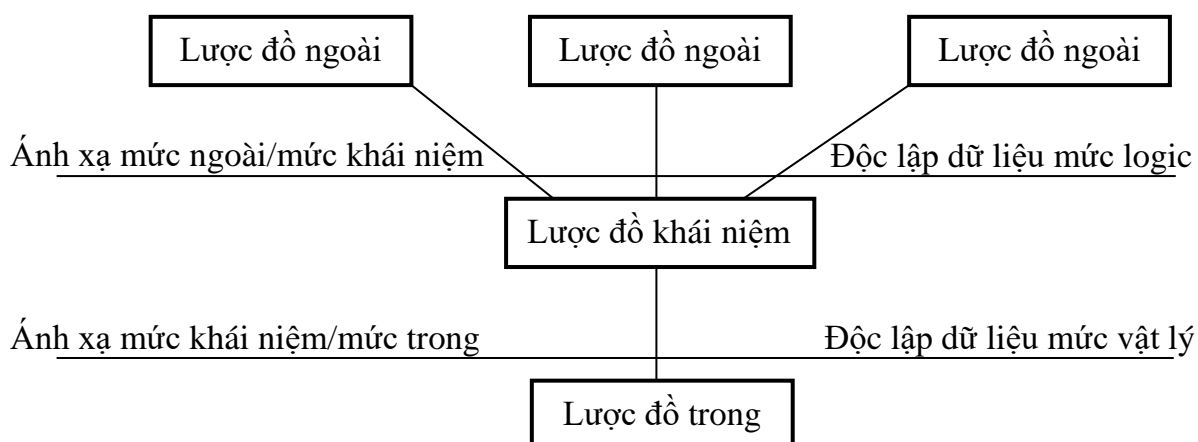
Như vậy nhiều thể hiện của CSDL có thể tương ứng với cùng một lược đồ CSDL. Đôi khi, lược đồ còn được gọi là *nội hàm* của CSDL và một thể hiện còn được gọi là một *mở rộng* hay một *trạng thái* của CSDL.

2.1.5. Tính độc lập dữ liệu

Có thể nói một cách khác về mục đích của kiến trúc ba mức của CSDL, đó là sự độc lập dữ liệu. Độc lập dữ liệu được hiểu theo nghĩa các lược đồ ở mức trên không bị ảnh hưởng khi có sự thay đổi các lược đồ ở các mức dưới. Có hai loại độc lập dữ liệu:

- **Độc lập dữ liệu mức vật lý:** Là khả năng sửa đổi lược đồ vật lý mà không thay đổi lược đồ logic, như vậy không đòi hỏi viết lại các trình ứng dụng. Để tăng tính hiệu quả, nhiều khi cần có những thay đổi ở mức vật lý, chẳng hạn như sử dụng các tổ chức tập tin khác, dùng thiết bị nhớ khác, thay đổi các chỉ mục hay thay đổi thuật toán băm.

- **Độc lập dữ liệu mức logic:** Là khả năng sửa đổi lược đồ logic mà không làm thay đổi các lược đồ ngoài, như vậy không đòi hỏi viết lại các trình ứng dụng. Các sửa đổi ở mức logic là cần thiết mỗi khi cấu trúc logic của CSDL cần phải thay đổi, chẳng hạn cần thêm hay bớt các thực thể nào đó, các thuộc tính hay các mối quan hệ của chúng. Những người dùng sử dụng những thông tin thay đổi này sẽ được thông báo về sự thay đổi, nhưng những người dùng khác sẽ không bị ảnh hưởng gì.



Hình 2.3. Sự độc lập dữ liệu của kiến trúc ba mức ANSI-SPARC

Độc lập dữ liệu logic khó thực hiện hơn độc lập dữ liệu vật lý vì các chương trình ứng dụng phụ thuộc nhiều vào cấu trúc logic của dữ liệu mà chúng truy cập.

2.2. Mô hình dữ liệu

Trên thực tế, một lược đồ được viết trong ngôn ngữ định nghĩa dữ liệu của một hệ quản trị CSDL cụ thể. Để mô tả các yêu cầu dữ liệu của một tổ chức sao cho mô tả đó dễ hiểu đối với nhiều NSD khác nhau thì ngôn ngữ này lại ở mức quá thấp. Như vậy cần phải có một mô tả lược đồ ở mức cao hơn, nói cách khác cần phải có một mô hình dữ liệu. Khi dùng mô hình dữ liệu chúng ta có thể biểu diễn dữ liệu theo một cách dễ hiểu.

Mô hình dữ liệu là một tập các khái niệm và ký pháp dùng để mô tả dữ liệu, các mối quan hệ của dữ liệu, và ràng buộc trên dữ liệu của một tổ chức.

Như vậy, có thể xem một mô hình dữ liệu có ba thành phần:

- Phần mô tả cấu trúc của CSDL.
- Phần mô tả các thao tác, định nghĩa các phép toán được phép trên dữ liệu.
- Phần mô tả các ràng buộc toàn vẹn để đảm bảo sự chính xác của dữ liệu.

Có rất nhiều mô hình dữ liệu. Chúng ta có thể phân loại các mô hình dữ liệu dựa trên các khái niệm mà chúng sử dụng để mô tả các cấu trúc cơ sở dữ liệu.

Các mô hình dữ liệu bậc cao hoặc mô hình dữ liệu mức khái niệm cung cấp các khái niệm gắn liền với cách cảm nhận dữ liệu của nhiều người sử dụng. Các mô hình này tập trung vào bản chất logic của biểu diễn dữ liệu, nó quan tâm đến cái được biểu diễn trong cơ sở dữ liệu chứ không phải cách biểu diễn dữ liệu.

Các mô hình dữ liệu bậc thấp hoặc các mô hình dữ liệu vật lý cung cấp các khái niệm mô tả chi tiết về việc các dữ liệu được lưu trữ trong máy tính như thế nào. Các khái niệm do mô hình dữ liệu vật lý cung cấp nói chung có ý nghĩa đối với các chuyên gia máy tính chứ không có ý nghĩa mấy đối với người sử dụng thông thường.

Ở giữa hai loại mô hình này là một lớp các mô hình dữ liệu mức logic, chúng cung cấp những khái niệm mà người sử dụng có thể hiểu được và không xa với cách tổ chức dữ liệu bên trong máy tính. Các mô hình dữ liệu mức logic che giấu một số chi tiết về việc lưu trữ dữ liệu nhưng có thể được cài đặt trực tiếp trên hệ thống máy tính.

Như vậy, các mô hình dữ liệu được phân loại như sau:

- Mô hình mức khái niệm: Gồm mô hình thực thể kết hợp, mô hình hướng đối tượng, mô hình ngữ nghĩa, mô hình dữ liệu chức năng.
- Mô hình mức logic: Gồm mô hình quan hệ, mô hình mạng, mô hình phân cấp.

2.2.1. Mô hình thực thể liên kết

Mô hình thực thể liên kết (hay còn gọi là mô hình thực thể kết hợp, mô hình thực thể quan hệ), viết tắt là mô hình E-R (Entity-Relationship model). Đây là mô hình dữ liệu khái niệm bậc cao hỗ trợ cho việc thiết kế CSDL.

Mô hình thực thể liên kết được xây dựng dựa trên nhận thức rằng thế giới thực mà chúng ta muốn phản ánh là một tập hợp các đối tượng cơ sở và các mối liên kết giữa chúng.

Các mô hình loại này dùng các khái niệm “thực thể” (*entity*) và “mối liên kết” (*relationship*).

2.2.1.1. Thực thể

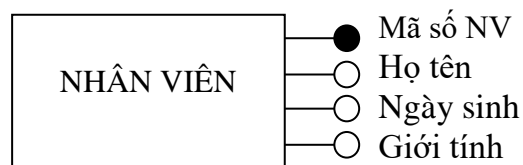
Một thực thể (*entity*) là một “vật” hay một “đối tượng” trong thế giới thực, phân biệt được với những đối tượng khác.

Chẳng hạn, mỗi nhân viên là một thực thể, các tài khoản ở ngân hàng là các thực thể, mỗi khoa hay sinh viên là một thực thể ...

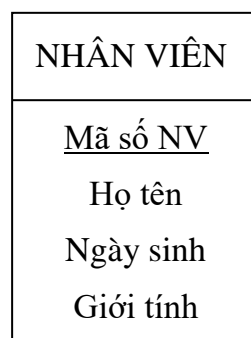
Một tập thực thể (*entity set*) là một tập hợp các thực thể cùng kiểu, nghĩa là cùng được thể hiện bởi một tập đặc trưng hay thuộc tính. Thuộc tính của thực thể (*entity attribute*) là các đặc tính riêng biệt cơ bản của thực thể.

Một tập thực thể thường được tham chiếu đến bằng cách dùng tên. Ví dụ, tập thực thể gồm tất cả thực thể nhân viên của công ty có thể được tham chiếu bằng tên “Nhân viên”. Như vậy, tên “Nhân viên” vừa là tên của một tập thực thể, đồng thời có thể coi đó là tên chỉ một kiểu thực thể nào đó.

Ví dụ: Một tập thực thể có tên là “Nhân viên”, gồm các thuộc tính “Mã số NV”, “Họ tên”, “Ngày sinh”, “Giới tính” được biểu diễn như sau:



Hay



Hình 2.4. Thực thể “Nhân viên” và các thuộc tính

Trong mô hình E-R có nhiều kiểu thuộc tính: Thuộc tính đơn và thuộc tính phức hợp, thuộc tính đơn trị và thuộc tính đa trị, thuộc tính được lưu trữ và thuộc tính được suy diễn, thuộc tính khóa.

- *Thuộc tính đơn* là thuộc tính không phân chia được thành những thành phần nhỏ hơn. Ví dụ, thuộc tính “Bậc lương” của kiểu thực thể “Nhân viên”.
- *Thuộc tính phức hợp* là thuộc tính có thể phân chia thành những thành phần nhỏ hơn, tức là chia thành những thuộc tính khác nữa. Ví dụ, thuộc tính Họ tên của kiểu thực thể “Nhân viên” có thể chia thành “Họ”, “Tên đệm”, “Tên”.

- *Thuộc tính đơn trị* là thuộc tính có một giá trị duy nhất cho một thực thể cụ thể. Ví dụ, thuộc tính “Tuổi” của kiểu thực thể “Nhân viên”, vì một nhân viên chỉ có một số tuổi.
- *Thuộc tính đa trị* là thuộc tính có thể có một tập hợp các giá trị cho cùng một thực thể. Ví dụ, thuộc tính “Số điện thoại” của kiểu thực thể “Nhân viên”, vì một nhân viên có thể có hơn một số điện thoại.
- Trong một số trường hợp, một số thuộc tính liên quan đến nhau theo kiểu giá trị của thuộc tính này có thể tính được giá trị của thuộc tính kia. Ví dụ, nếu biết ngày tháng năm sinh của một người, chúng ta có thể biết được tuổi của người đó. Thuộc tính Ngày tháng năm sinh được gọi là *thuộc tính được lưu trữ*, còn thuộc tính Tuổi gọi là *thuộc tính được suy diễn*.
- Có một ràng buộc trên một kiểu thực thể gọi là ràng buộc khóa hay còn gọi là ràng buộc về tính duy nhất trên các thuộc tính, nghĩa là mỗi kiểu thực thể thường có một thuộc tính mà bất cứ hai thực thể khác nhau trong tập thực thể đều có giá trị khác nhau trên thuộc tính đó. Thuộc tính như vậy gọi là *thuộc tính khóa*. Có thể dùng thuộc tính khóa để xác định (nhận diện) một thực thể duy nhất. Ví dụ, thuộc tính “Mã số NV” là thuộc tính khóa của kiểu thực thể “Nhân viên”. Đôi khi, vài thuộc tính kết hợp với nhau tạo thành một khóa, nghĩa là tổ hợp giá trị của các thuộc tính này ở các thực thể khác nhau phải khác nhau.

2.2.1.2. Liên kết

Một liên kết (relationship) là một sự kết hợp của một số thực thể.

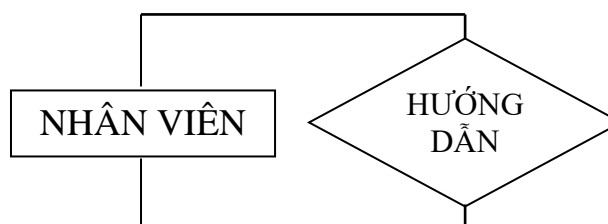
Ví dụ, liên kết “Làm việc cho” kết hợp một thực thể “Nhân viên” với một thực thể “Phòng”, liên kết “Điều hành” kết hợp một thực thể “Phòng” với một thực thể “Dự án”, liên kết “Có” kết hợp một thực thể “Chi nhánh” với một thực thể “Nhân viên”.

Sau đây là biểu diễn liên kết “Làm việc cho” kết hợp một thực thể “Nhân viên” với một thực thể Phòng:



Hình 2.5. Quan hệ giữa hai thực thể

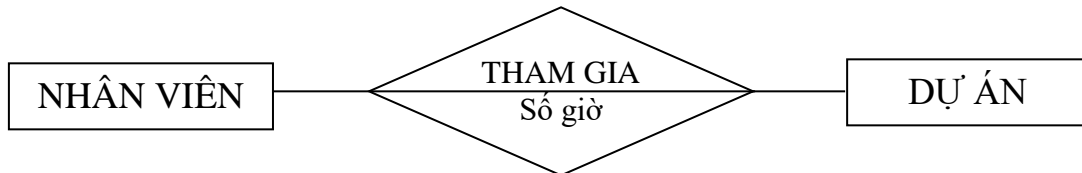
Một liên kết từ một thực thể đến chính thực thể ấy gọi là *liên kết đệ quy*. Ví dụ, thực thể “Nhân viên” có hai vai trò khác nhau trong liên kết “Hướng dẫn”, đó là vai trò “hướng dẫn” và vai trò “tiếp thu sự hướng dẫn”.



Hình 2.6. Quan hệ đệ quy

Bậc của một liên kết là số kiểu thực thể tham gia vào liên kết đó. Ví dụ, giữa 3 kiểu thực thể “Nhân viên”, “Phòng”, “Dự án” có thể có một liên kết cấp 3.

Liên kết có thể có các *thuộc tính* riêng của nó. Thông thường liên kết có các thuộc tính là khóa của các loại thực thể tham gia vào mỗi liên kết, ngoài ra còn có thêm những thuộc tính bổ sung khác. Chẳng hạn trong mỗi liên kết “Tham gia” giữa “Nhân viên” và “Dự án”, mỗi liên kết “Tham gia” có thể có thêm thuộc tính “Số giờ” để chỉ số giờ mà nhân viên đó tham gia vào dự án.



Hình 2.7. Thuộc tính của liên kết

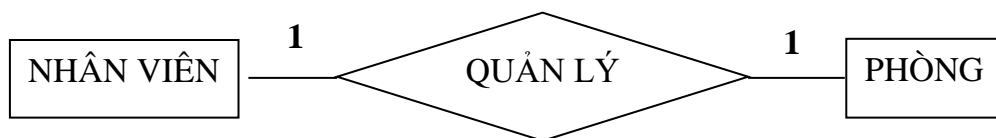
2.2.1.3. Tỉ số

Mô hình thực thể liên kết còn biểu diễn một số ràng buộc mà dữ liệu trong CSDL phải tuân thủ. Một ràng buộc quan trọng là các *tỉ số (lực lượng) của ánh xạ*, biểu thị số các thực thể được liên kết với một thực thể khác thông qua một tập mỗi liên kết.

Tỉ số lực lượng trên một kiểu quan hệ cấp hai có thể gặp là 1:1, 1:N hay N:1, N:M.

- Tỉ số 1:1:

Một ví dụ về tỉ số liên kết 1:1 là kiểu liên kết “Quản lý” giữa “Nhân viên” và “Phòng”, phản ánh mối liên quan giữa một thực thể phòng với một thực thể nhân viên là người quản lý của phòng đó. Nếu như một phòng chỉ có thể có một người quản lý và một nhân viên chỉ là người quản lý của tối đa một phòng thì tỉ số của kiểu liên kết là 1:1. (Hình 2.8).



hay:

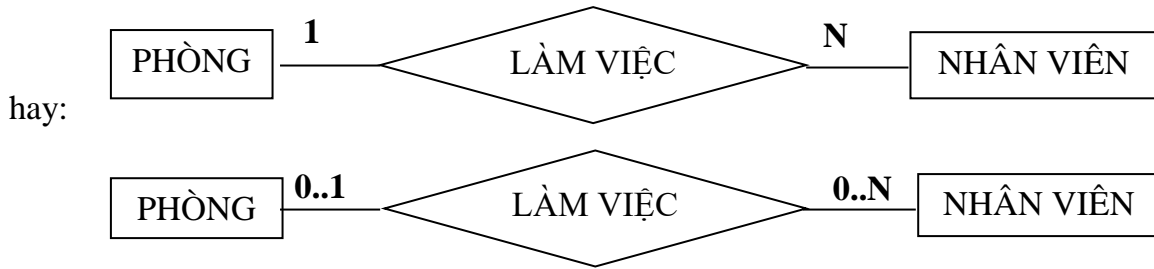


Hình 2.8. Hai cách biểu diễn tỉ số lực lượng 1:1 giữa Nhân viên và Phòng

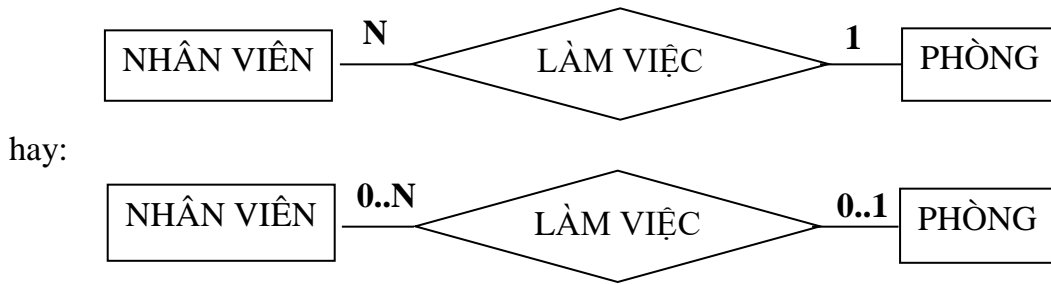
- Tỉ số 1:N hay N:1:

Trường hợp một phòng ban có thể có nhiều nhân viên, nhưng một nhân viên chỉ có thể làm việc ở duy nhất một phòng ban, thì tỉ số lực lượng của kiểu liên kết “Làm việc” giữa “Nhân viên” và “Phòng ban” là 1:N, N là số bất kỳ (Hình 2.9).

Nếu chúng ta biểu diễn ngược lại thì tỉ số lực lượng là N:1, N là số bất kỳ (Hình 2.10).



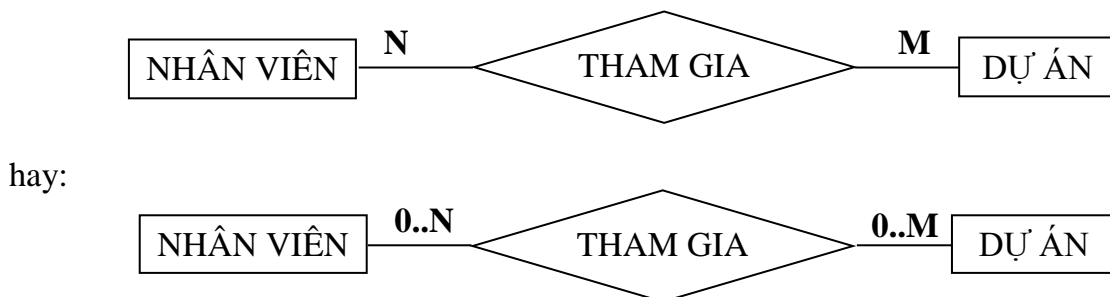
Hình 2.9. Hai cách biểu diễn tỉ số lực lượng 1:N giữa Phòng và Nhân viên



Hình 2.10. Hai cách biểu diễn tỉ số lực lượng N:1 giữa Nhân viên và Phòng

- Tỉ số N:M:

Trường hợp một nhân viên có thể tham gia nhiều dự án và một dự án có thể tham gia bởi nhiều nhân viên, thì tỉ số lực lượng của liên kết “Tham gia” giữa “Nhân viên” và “Dự án” là N:M, với N và M là số bất kỳ (Hình 2.11).

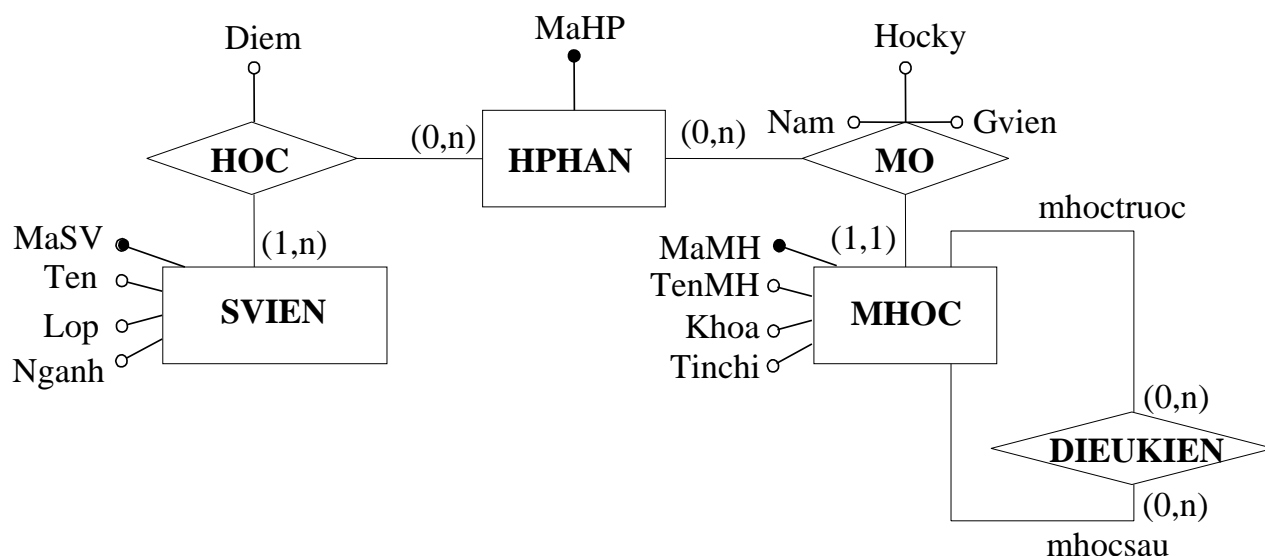


Hình 2.11. Hai cách biểu diễn tỉ số lực lượng N:M giữa “Nhân viên” và “Dự án”

Hình 2.12 là ví dụ về mô hình thực thể liên kết, bao gồm:

- Thực thể SVIEN (sinh viên): gồm thuộc tính khóa là MaSV (mã sinh viên), các thuộc tính Ten (tên sinh viên), Lop (lớp), Ngành (ngành).
- Thực thể HPHAN (học phần): gồm thuộc tính khóa là MaHP (mã học phần).
- Thực thể MHOC (môn học): gồm thuộc tính khóa là MaMH (mã môn học), các thuộc tính TenMH (tên môn học), Khoa (Khoa), Tinchi (tín chỉ).
- Giữa thực thể SVIEN và HPHAN có liên kết HOC (học), liên kết này có thuộc tính Diem (điểm của sinh viên khi học học phần đó).

- Giữa thực thể HPHAN và MHOC có liên kết MO (mở), liên kết này có các thuộc tính Nam (năm), HKy (học kỳ), Gvien (giáo viên).
- Ngoài ra thực thể MHOC còn có liên kết đệ quy DIEUKIEN (điều kiện) cho biết môn nào học trước, môn nào học sau.



Hình 2.12. Ví dụ mô hình thực thể liên kết

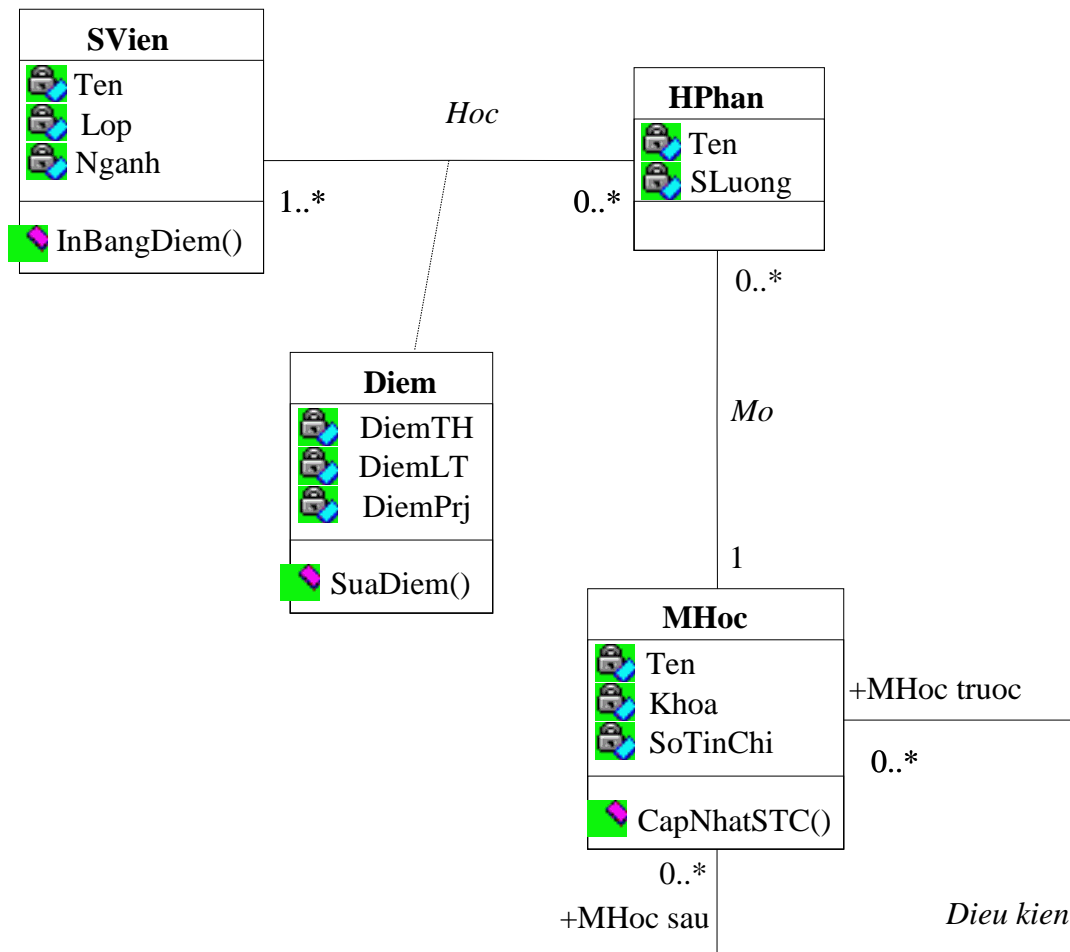
2.2.2. Mô hình hướng đối tượng

Mô hình hướng đối tượng dựa trên cơ sở các đối tượng. Một đối tượng chứa các *thuộc tính* được lưu trữ trong các *biến thể hiện* ở bên trong đối tượng. Một đối tượng còn chứa các phần mã thao tác trên đối tượng, phần mã đó gọi là *phương thức*. Các đối tượng chứa cùng các kiểu thuộc tính và cũng các phương thức như nhau được nhóm thành các *lớp*. Một lớp có thể xem như một định nghĩa kiểu cho các đối tượng.

Chỉ có một cách theo đó một đối tượng có thể truy cập dữ liệu của một đối tượng khác là gọi tới phương thức của đối tượng khác đó. Hành động này gọi là *gửi một thông báo* tới đối tượng. Như vậy, giao diện của các phương thức của một đối tượng định nghĩa phần nhìn thấy từ bên ngoài của đối tượng. Phần bên trong của đối tượng - các biến thể hiện và mã của phương thức - là không thấy được từ bên ngoài, kết quả là có hai mức truy xuất dữ liệu.

Hình 2.13 là ví dụ về mô hình hướng đối tượng, bao gồm:

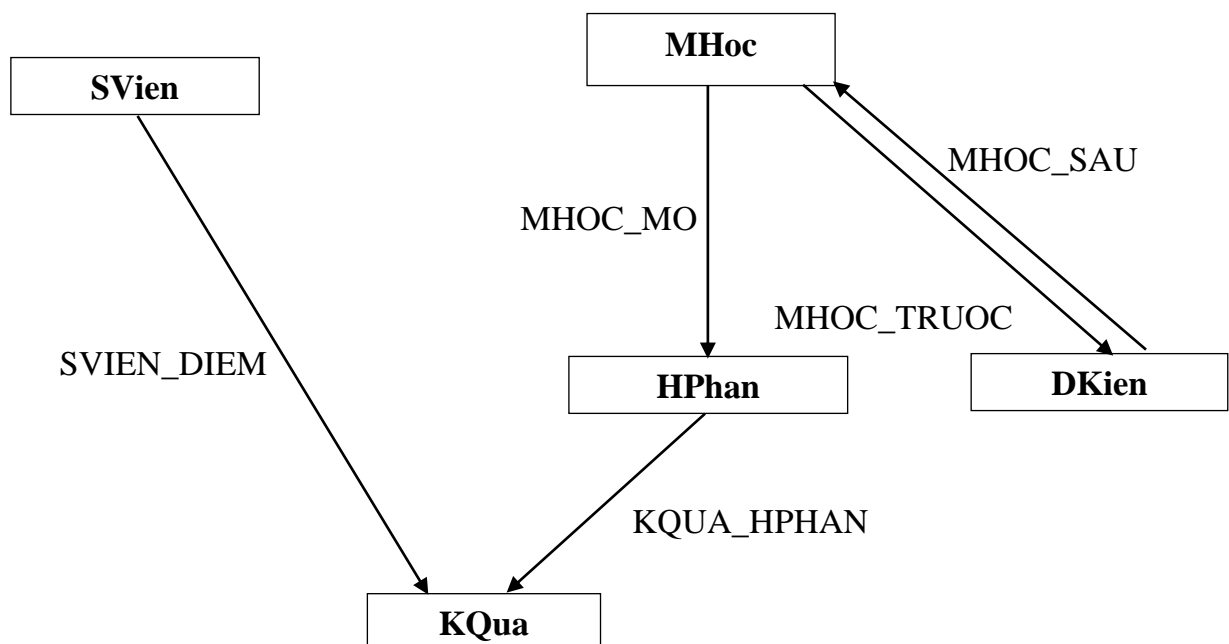
- Đối tượng SVien (sinh viên): gồm các biến Ten (tên sinh viên), Lop (lớp), Nganh (ngành) và các phương thức LapTKB (lập thời khóa biểu), InBangDiem (in bảng điểm).
- Đối tượng HPhan (học phần) gồm các biến Ten (tên học phần), SLuong (số lượng).
- Đối tượng Diem (điểm) gồm các biến DiemTH (điểm thực hành), DiemLT (điểm lý thuyết), DiemPrj (điểm đồ án), và phương thức SuaDiem (sửa điểm).
- Đối tượng MHoc gồm các biến Ten (tên), Khoa, SoTinChi (số tín chỉ) và phương thức CapNhatSTC (cập nhật số tín chỉ).



Hình 2.13. Ví dụ mô hình CSDL hướng đối tượng

Trong mô hình hướng đối tượng, một đối tượng độc lập với các giá trị mà nó chứa đựng. Như vậy hai đối tượng chứa cùng các giá trị vẫn có thể là khác nhau.

2.2.3. Mô hình mạng



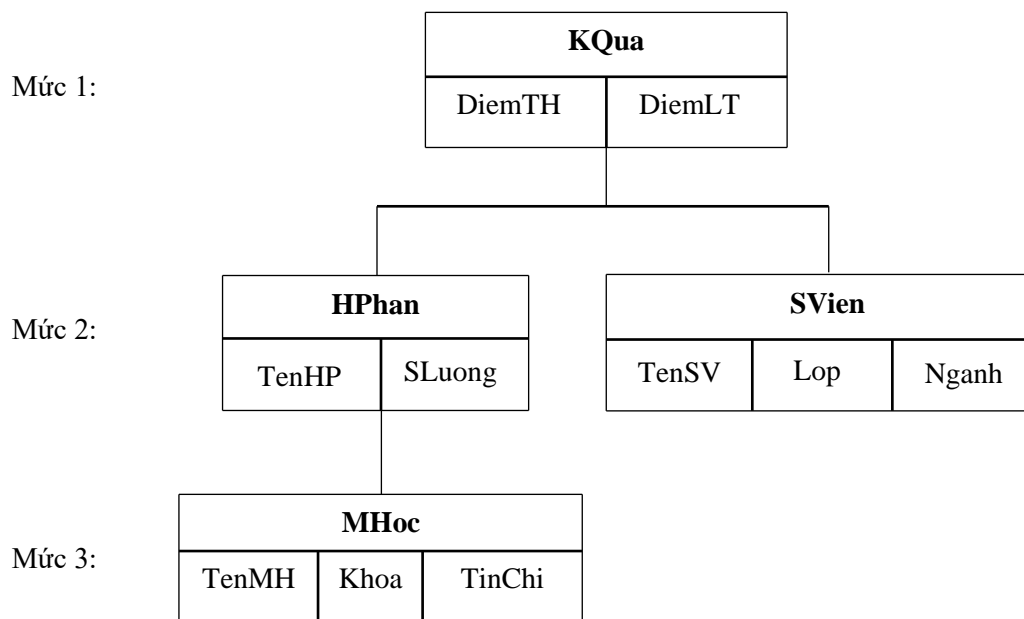
Hình 2.14. Ví dụ mô hình CSDL mạng

Trong mô hình mạng, dữ liệu được biểu diễn bởi một tập các bản ghi, còn các mối quan hệ được biểu diễn bởi các mối nối có thể xem như những con trỏ (giống như đồ thị).

Xuất phát từ một đối tượng (biểu diễn bằng một bản ghi) có thể có nhiều mối quan hệ đến những đối tượng khác nhau. Trong những quan hệ đó luôn phân biệt đối tượng là chủ của quan hệ và những đối tượng thành phần của quan hệ.

2.2.4. Mô hình phân cấp

Mô hình phân cấp tương tự như mô hình mạng ở chỗ dữ liệu biểu diễn bằng tập các bản ghi, và mối quan hệ giữa dữ liệu biểu diễn bằng các mối nối như các con trỏ. Nhưng khác với mô hình mạng, mỗi quan hệ giữa hai đối tượng trong mô hình phân cấp thể hiện theo kiểu cha-con, và sơ đồ các bản ghi cùng các quan hệ giữa chúng có cấu trúc như các cây chứ không phải các đồ thị.



Hình 2.15. Ví dụ mô hình CSDL phân cấp

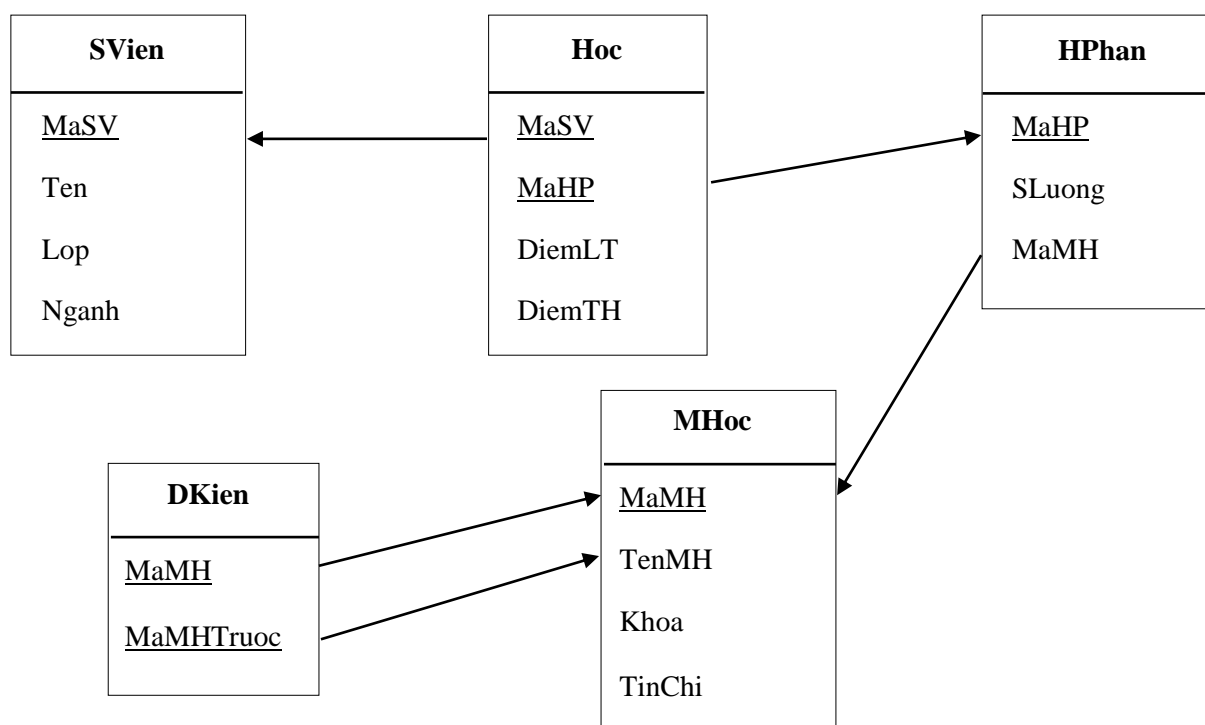
2.2.5. Mô hình quan hệ

Mô hình quan hệ là một trong những mô hình CSDL có cơ sở lý thuyết được xây dựng vững chắc nhất. Trong mô hình quan hệ, dữ liệu được thể hiện trong các *bảng*. Mỗi bảng gồm:

- Các *cột* (thường gọi là *trường*). Mỗi cột có một tên duy nhất, có kiểu dữ liệu và các ràng buộc.
- Các *dòng* (thường gọi là *bản ghi* hay *bộ*).

Giữa các bản có quan hệ với nhau thông qua các khóa.

(Xem chi tiết hơn ở chương 3).



Hình 2.16. Ví dụ mô hình CSDL quan hệ

Hình 2.16 là một mô hình CSDL quan hệ gồm các bảng SVien, Hoc, HPhan, DKien, MHoc.

Bảng 2.2. Một bảng trong mô hình CSDL quan hệ

Cột			
MaSV	Ten	Lop	Nganh
T001	Trần Thị Lan	03T1	CNTT
T002	Nguyễn Văn Dũng	03T1	CNTT
T003	Lý Hải	03T1	CNTT

2.2.6. Lựa chọn giữa các mô hình dữ liệu

Tổ chức dữ liệu theo mô hình nào là tốt nhất? Thực tế chưa có mô hình dữ liệu nào là tốt nhất. Tốt nhất phụ thuộc vào yêu cầu truy xuất và khai thác thông tin của đơn vị quản lý nó. Nó được sử dụng ở đâu và vào lúc nào là tốt nhất. Tuy nhiên, thường người ta dựa vào các tiêu chí sau để nói rằng mô hình dữ liệu tốt nhất khi:

- **Mục đích:** Phần lớn các mô hình dữ liệu sử dụng hệ thống ký hiệu để biểu diễn dữ liệu và làm nền tảng cho các hệ ứng dụng và ngôn ngữ thao tác dữ liệu. Các mô hình thực thể kết hợp không có hệ thống ký hiệu để xây dựng các phép toán thao tác dữ liệu, mà sử dụng để thiết kế lược đồ khái niệm, cài đặt trong một mô hình dữ liệu với một hệ quản trị cơ sở dữ liệu nào đó.

- **Hướng giá trị hay hướng đối tượng:** Các mô hình dữ liệu quan hệ là mô hình dữ liệu hướng giá trị, do đó trong mô hình dữ liệu này có tính khai báo (declarativeness) và có tác động đến các ngôn ngữ được nó hỗ trợ. Các mô hình mạng, mô hình phân cấp, mô hình hướng đối tượng cung cấp đặc tính nhận dạng đối tượng, nên có thể xem chúng là các mô hình hướng đối tượng. Mô hình thực thể kết hợp cũng có đặc tính nhận dạng hướng đối tượng.

- **Tính dư thừa:** Tất cả các mô hình dữ liệu đều có khả năng hỗ trợ lưu trữ dữ liệu vật lý và hạn chế sự dư thừa dữ liệu. Tuy nhiên các mô hình dữ liệu hướng đối tượng giải quyết sự dư thừa tốt hơn, bằng cách tạo ra con trỏ trỏ đến nhiều vị trí khác nhau.

- **Giải quyết mối quan hệ nhiều – nhiều:** Phần lớn trong các mô hình dữ liệu có chứa các mối quan hệ nhiều – nhiều, một – nhiều hay quan hệ một – một. Một quan hệ có nhiều phần tử của các quan hệ khác và ngược lại. Tuy nhiên trong mô hình mạng không chấp nhận mối quan hệ nhiều – nhiều.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 2

2.1. Nêu rõ các mức của kiến trúc ANSI-PARC 3 mức.

2.2. Nêu các mô hình dữ liệu chính.

2.3. Giải thích thuật ngữ “độc lập dữ liệu” và sự quan trọng của tính độc lập dữ liệu trong môi trường của một hệ CSDL. Hãy giải thích sự khác nhau giữa tính độc lập dữ liệu vật lý và tính độc lập dữ liệu logic.

2.4. Cho mô hình quan hệ gồm 2 bảng như sau:

Bảng 2.3. Bảng HọcSinh

TenHocSinh	GioiTinh	NgaySinh	NoiSinh	Lop
Trần Thanh	Nam	03/08/1990	Đà Nẵng	10/1
Thái Thị Huyền	Nữ	05/07/1990	Đà Nẵng	10/2
Lê Anh	Nam	05/04/1990	Quảng Bình	10/2
Trần Thị Huệ	Nữ	14/04/1990	Huế	10/2

Bảng 2.4. Bảng Lớp

Lop	GVCN
10/1	Trần Thị Hương
10/2	Nguyễn Thị Thuỳ My

Hãy vẽ các mô hình thực thể liên kết, mô hình hướng đối tượng, mô hình phân cấp và mô hình mạng, có thể bổ sung một số thông tin nếu cần thiết.

2.5. Giả sử trong nghiệp vụ quản lý phát hành báo chí, thông tin gồm có:

a. Thông tin về khách hàng đặt mua báo (tạp chí) :

- Mã khách hàng đặt mua, 5 ký tự, kiểu character
- Họ và tên khách hàng, 21 ký tự, kiểu character
- Địa chỉ khách hàng, 21 ký tự, kiểu character
- Số điện thoại, 7 ký tự, kiểu character
- Địa điểm giao nhận báo (tạp chí) hàng ngày, 25 ký tự, kiểu character

b. Thông tin về các loại báo (tạp chí) bao gồm:

- Mã báo khách đặt mua, 3 ký tự, kiểu character
- Tên báo (tạp chí), 15 ký tự, kiểu character
- Giá báo, 5 số
- Kỳ phát hành báo (báo ngày, báo tuần, báo tháng...), 3 ký tự, , kiểu character

c. Thông tin về phiếu đặt báo gồm

- Ngày khách đặt báo, chỉ
- Mã hiệu khách hàng đặt mua báo

- Mã báo khách đặt mua
- Số lượng báo, chỉ khách đặt mua
- Thành tiền từng loại báo.

Yêu cầu:

- Hãy phát hoạ mô hình CSDL quan hệ với các dữ liệu trên.
- Hãy phát hoạ mô hình CSDL mạng
- Hãy phát hoạ mô hình CSDL phân cấp

2.6. Hãy thiết kế lược đồ thực thể liên kết cho ví dụ về Quản lý lao động:

Để quản lý việc phân công các nhân viên tham gia vào xây dựng các công trình. Công ty xây dựng ABC tổ chức quản lý như sau:

Cùng lúc công ty có thể tham gia xây dựng nhiều công trình, mỗi công trình có một mã số công trình duy nhất (MACT), mỗi mã số công trình xác định các thông tin như: tên gọi công trình (TENCT), địa điểm (ĐIADIEM), ngày công trình được cấp giấy phép xây dựng (NGAYCAPGP), ngày khởi công (NGAYKC), ngày hoàn thành (NGAYHT).

Mỗi nhân viên của công ty ABC có một mã số nhân viên (MANV) duy nhất, một mã số nhân viên xác định các thông tin như: Họ tên (HOTEN), ngày sinh (NGAYSINH), phái (PHAI), địa chỉ (ĐIACHI). Mỗi nhân viên phải chịu sự quản lý hành chính bởi một phòng ban. Tất nhiên một phòng ban quản lý hành chính nhiều nhân viên. Công ty có nhiều phòng ban (Phòng kế toán, phòng kinh doanh, phòng kỹ thuật, phòng tổ chức, phòng chuyên môn, Phòng phục vụ,...). Mỗi phòng ban có một mã số phòng ban (MAPB) duy nhất, mã phòng ban xác định tên phòng ban (TENPB).

Công ty phân công các nhân viên tham gia vào các công trình, mỗi công trình có thể được phân cho nhiều nhân viên và mỗi nhân viên cùng lúc cũng có thể tham gia vào nhiều công trình. Với mỗi công trình một nhân viên có một số lượng ngày công (SLNGAYCONG) đã tham gia vào công trình đó.

2.7. Hãy thiết kế lược đồ thực thể liên kết cho ví dụ về Quản lý thư viện:

Một thư viện tổ chức việc cho mượn sách như sau:

Mỗi quyển sách được đánh một mã sách (MASH) dùng để phân biệt với các quyển sách khác (giả sử nếu một tác phẩm có nhiều bản giống nhau hoặc có nhiều tập thì cũng xem là có mã sách khác nhau), mỗi mã sách xác định các thông tin khác như: tên sách (TENSACH), tên tác giả (TACGIA), nhà xuất bản (NHAXB), năm xuất bản (NAMXB).

Mỗi độc giả được thư viện cấp cho một thẻ thư viện, trong đó có ghi rõ mã đọc giả (MADG), cùng với các thông tin khác như: họ tên (HOTEN), ngày sinh (NGAYSINH), địa chỉ (ĐIACHI), nghề nghiệp (NGHENGHIEP).

Cứ mỗi lượt mượn sách, độc giả phải ghi các quyển sách cần mượn vào một phiếu mượn, mỗi phiếu mượn có một số phiếu mượn (SOPM) duy nhất, mỗi phiếu mượn xác định các thông tin như: ngày mượn (NGAYMUON), độc giả mượn, các quyển sách mượn và ngày trả (NGAYTRA). Các quyển sách trong cùng một phiếu mượn không nhất thiết phải trả trong cùng một ngày.

2.8. Hãy thiết kế lược đồ thực thể liên kết cho ví dụ về Quản lý bán hàng:

Mỗi khách hàng có một mã khách hàng (MAKH) duy nhất, mỗi MAKH xác định được các thông tin về khách hàng như: họ tên khách hàng (HOTEN), địa chỉ (ĐIACHI), số điện thoại (ĐIENTHOAI).

Các mặt hàng được phân loại theo từng nhóm hàng, mỗi nhóm hàng có một mã nhóm (MANHOM) duy nhất, mỗi mã nhóm hàng xác định tên nhóm hàng (TENNHOM), tất nhiên một nhóm hàng có thể có nhiều mặt hàng. Mỗi mặt hàng được đánh một mã số (MAHANG) duy nhất, mỗi mã số này xác định các thông tin về mặt hàng đó như : tên hàng (TENHANG), đơn giá bán (ĐONGIA), đơn vị tính (ĐVT). Mỗi hóa đơn bán hàng có một số hóa đơn (SOHD) duy nhất, mỗi hóa đơn xác định được khách hàng và ngày lập hóa đơn (NGAYLAPHĐ), ngày bán hàng (NGAYBAN). Với mỗi mặt hàng trong một hóa đơn cho biết số lượng bán (SLBAN) của mặt hàng đó.

2.9. Hãy thiết kế lược đồ thực thể liên kết cho ví dụ về Quản lý lịch dạy học:

Để quản lý lịch dạy của các giáo viên và lịch học của các lớp, một trường tổ chức như sau:

Mỗi giáo viên có một mã số giáo viên (MAGV) duy nhất, mỗi MAGV xác định các thông tin như: họ và tên giáo viên (HOTEN), số điện thoại (DTGV). Mỗi giáo viên có thể dạy nhiều môn cho nhiều khoa nhưng chỉ thuộc sự quản lý hành chánh của một khoa nào đó.

Mỗi môn học có một mã số môn học (MAMH) duy nhất, mỗi môn học xác định tên môn học (TENMH). Cùng với mỗi lớp thì mỗi môn học chỉ được phân cho một giáo viên.

Mỗi phòng học có một số phòng học (SOPHONG) duy nhất, mỗi phòng có một chức năng (CHUCNANG), chẳng hạn như phòng lý thuyết, phòng thực hành máy tính, phòng nghe nhìn, xưởng thực tập cơ khí,...

Mỗi khoa có một mã khoa (MAKHOA) duy nhất, mỗi khoa xác định các thông tin như: tên khoa (TENKHOA), số điện thoại của khoa (DTKHOA).

Mỗi lớp có một mã lớp (MALOP) duy nhất, mỗi lớp có một tên lớp (TENLOP), sĩ số lớp (SISO).

Mỗi lớp có thể học nhiều môn của nhiều khoa nhưng chỉ thuộc sự quản lý hành chính của một khoa nào đó.

Hàng tuần, mỗi giáo viên phải lập lịch báo giảng cho biết giáo viên đó sẽ dạy những lớp nào, ngày nào (NGAYDAY), môn gì?, tại phòng nào, từ tiết nào (TUTIET) đến tiết nào (DENTIET), tựa đề bài dạy (BAIDAY), ghi chú (GHICHU) về các tiết dạy này, đây là giờ dạy lý thuyết (LYTHUYET) hay thực hành - giả sử nếu LYTHUYET=1 thì đó là giờ dạy thực hành và nếu LYTHUYET=2 thì đó là giờ lý thuyết, một ngày có 16 tiết, sáng từ tiết 1 đến tiết 6, chiều từ tiết 7 đến tiết 12, tối từ tiết 13 đến 16.

2.10. Hãy thiết kế lược đồ thực thể liên kết cho ví dụ về Quản lý trường học:

Trường được chia thành các trường con: Trường KHTN, Trường KHXH, Trường Công nghệ,... Mỗi trường có một hiệu trưởng quản lý. Mỗi hiệu trưởng quản lý một trường.

Mỗi trường có nhiều khoa. Chẳng hạn, trường KHTN có các khoa Toán, Lý, Hoá,... Mỗi một khoa chỉ thuộc về một trường. Thông tin về Khoa gồm Mã khoa, tên khoa, địa chỉ, số điện thoại, tên trường.

Mỗi Khoa cung cấp nhiều môn học. Mỗi môn học gồm có Tên môn học, mã số, số đơn vị học trình, trình độ, tên Khoa.

Mỗi môn học có thể có nhiều học phần. Mỗi học phần được lưu giữ bằng các thông tin: Mã học phần, Tên môn học, Tên giáo viên dạy, học kỳ.

Mỗi khoa có nhiều giáo viên làm việc, nhưng mỗi giáo viên chỉ làm việc cho một khoa. Mỗi một khoa có một chủ nhiệm khoa, đó là một giáo viên.

Mỗi giáo viên có thể dạy nhiều nhất là 4 học phần và cũng có thể không dạy học phần nào.

Mỗi sinh viên phải học nhiều học phần.

Mỗi khoa có nhiều sinh viên, mỗi sinh viên chỉ thuộc về một khoa. Thông tin về mỗi sinh viên gồm: Mã sinh viên, Họ tên, địa chỉ, ngày sinh, giới tính, Lớp, Tên Khoa và chế độ đào tạo.

Mỗi sinh viên có một người giám sát (giáo viên chủ nhiệm), người đó là một giáo viên.

Sau mỗi học kỳ sẽ có một danh sách điểm để phân loại, gồm các thông tin: Mã sinh viên, mã học phần, điểm bằng chữ, điểm bằng số.

CHƯƠNG 3

MÔ HÌNH QUAN HỆ

Từ năm 1980, các hệ CSDL dựa trên mô hình quan hệ được sử dụng rộng rãi bởi tính đơn giản và các cơ sở toán học của nó.

Mô hình quan hệ biểu thị dữ liệu trong một CSDL như một tập các quan hệ. Có thể coi một quan hệ là một bảng giá trị gồm các hàng và các cột. Mỗi hàng trong bảng là một tập các giá trị có liên quan đến nhau, giá trị này biểu thị một sự kiện tương ứng với một thực thể hay một mối quan hệ trong thế giới thực. Tên bảng và tên cột được dùng để giúp cho việc hiểu nghĩa của mỗi hàng trong bảng.

Nội dung của chương này là trình bày *kiến trúc tổng quát về mô hình quan hệ* và áp dụng nó để lập mô hình dữ liệu quan hệ có *hiệu quả trong lưu trữ và khai thác*.

Nội dung đầu tiên mà chương đề cập đến là các lịch sử của mô hình quan hệ, các khái niệm trong mô hình quan hệ và các ràng buộc toàn vẹn. Tiếp theo là trình bày về các ngôn ngữ quan hệ, và cuối cùng là các khung nhìn.

3.1. Mở đầu

Mô hình cơ sở dữ liệu quan hệ (gọi tắt là Mô hình quan hệ) do E.F Codd đề xuất năm 1971. Mô hình này bao gồm:

- Một hệ thống các ký hiệu để mô tả dữ liệu dưới dạng dòng và cột như quan hệ, bộ, thuộc tính, khóa chính, khóa ngoại, ...
- Một tập hợp các phép toán thao tác trên dữ liệu như phép toán tập hợp, phép toán quan hệ.
- Ràng buộc toàn vẹn quan hệ.

Các hệ quản trị CSDL sử dụng phổ biến như Oracle, MS Access, MS SQL Server ngày nay được xây dựng dựa vào lý thuyết của mô hình quan hệ.

3.2. Các khái niệm của mô hình quan hệ

3.2.1. Miền (Domain)

Một miền D là một tập hợp các giá trị nguyên tố, hiểu theo nghĩa mỗi giá trị trong miền không thể phân chia trong phạm vi mô hình quan hệ.

Để đặc tả một miền người ta chỉ ra một tên miền, một kiểu dữ liệu và khuôn dạng dữ liệu.

Ví dụ:

- Miền của Mã sinh viên là một tập hợp các dãy ký tự có độ dài từ 5 đến 8, bắt đầu là một chữ cái.
- Miền của Họ tên là một tập các dãy chữ cái có độ dài không quá 35.

Ngoài ra có một số thông tin phụ để thể hiện các giá trị của miền, chẳng hạn như các đơn vị tính (tiền, khối lượng) cũng có thể được chỉ ra trong việc đặc tả một miền.

3.2.2. Thuộc tính (Attribute)

Thuộc tính 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 Khoa có các thuộc tính Mã khoa, Tên khoa.
- Đối tượng Lớp học có thuộc tính Mã lớp, Tên lớp, Niên khóa, Số học viên...
- Đối tượng Môn học có thuộc tính Mã môn, Tên môn, Số tín chỉ...
- Đối tượng Sinh viên có các thuộc tính Mã sinh viên, Tên sinh viên, Ngày sinh, Giới tính, Quê quán, Địa chỉ...
- Đối tượng Giảng viên có các thuộc tính Mã giảng viên, Tên giảng viên, Học vị, Chuyên ngành...

Các thuộc tính được *phân biệt qua tên gọi và phải thuộc vào một kiểu dữ liệu nhất định* (số, chuỗi, ngày tháng, logic, hình ảnh,...). Kiểu dữ liệu ở đây là kiểu đơn. Trong cùng một đối tượng không được có hai thuộc tính cùng tên.

Một số kiểu dữ liệu thường dùng:

- Text (hoặc Character, String, hoặc Char) – kiểu văn bản
- Number (hoặc Numeric, hoặc float) – kiểu số
- Logical (hoặc Boolean) – kiểu luận lý
- Date/Time – kiểu thời gian: ngày tháng năm + giờ phút
- Memo (hoặc VarChar) – kiểu văn bản có độ dài thay đổi

Thông thường mỗi thuộc tính chỉ chọn lấy giá trị trong một tập con của kiểu dữ liệu và tập hợp con đó gọi là *miền giá trị của thuộc tính đó*.

Ví dụ:

- Thuộc tính ngày (trong một tháng) có kiểu dữ liệu là số nguyên, miền giá trị từ 1 đến (tối đa là) 31.
- Điểm thi có kiểu dữ liệu số nguyên, miền giá trị từ 0 đến 10.

3.2.3. Quan hệ (Relation)

Một quan hệ trên một tập các thuộc tính là một tập con của tích Đề-các của một hay nhiều miền.

Cho một quan hệ r có n ngôi được định nghĩa trên tập các thuộc tính $U = \{A_1, A_2, \dots, A_n\}$.

Khi đó $r \subseteq \text{MGT}(A_1) \times \text{MGT}(A_2) \times \dots \times \text{MGT}(A_n)$, trong đó $\text{MGT}(A_i)$ là miền của thuộc tính (A_i) , là tập các giá trị có thể của thuộc tính đó.

Như vậy quan hệ r là một tập hợp các n -bộ có dạng:

$$r = \{ (a_1, a_2, \dots, a_n) \mid a_i \in \text{MGT}(A_i), i = 1, 2, \dots, n \}.$$

Quan hệ còn được gọi bằng thuật ngữ khác là *bảng* (table).

Ví dụ:

Khoa(MaKhoa, TenKhoa) là một quan hệ 2 ngôi.

GiangVien (MaGiangVien, TenGiangVien, CapHocVi, MaKhoa) là quan hệ 4 ngôi.

LopHoc(MaLop, TenLop, NienKhoa, SoSinhVien, Makhoa) là quan hệ 5 ngôi.

MonHoc (MaMon, TenMon, SoTC) là quan hệ 3 ngôi.

SinhVien (MaSinhVien, TenSinhVien, NgaySinh, QueQuan, MaLop) là quan hệ 5 ngôi.

Bậc (degree) của một quan hệ là số thuộc tính của quan hệ đó. Bản số (cardinality) của một quan hệ là số các bộ (tuple – xem phần 3.2.5) mà nó chứa đựng.

3.2.4. Lược đồ quan hệ (Relation scheme)

Một lược đồ quan hệ R là một cặp có thứ tự $R = \langle U, F \rangle$, trong đó U là tập hữu hạn các thuộc tính của quan hệ, F là tập các điều kiện giữa các thuộc tính (F còn gọi là tập các ràng buộc toàn vẹn).

Một ràng buộc trên tập các thuộc tính $\{A_1, A_2, \dots, A_n\}$ là một tính chất (một tân từ) trên tập tất cả các quan hệ xác định trên tập thuộc tính đó.

Chẳng hạn, lược đồ quan hệ Sinh viên (đặt tên là Sv) với các thuộc tính Mã sinh viên, Họ sinh viên, Tên sinh viên, Giới tính, Ngày sinh, Mã lớp và Tỉnh được viết như sau: Sv(MaSV, HoSV, TenSV, GioiTinh, NgaySinh, MaLop, Tinh).

Ví dụ:

Lược đồ quan hệ Khoa(MaKhoa, TenKhoa) với tân từ: "Mỗi khoa có một tên gọi và một mã số duy nhất để phân biệt với tất cả các khoa khác của trường".

Lược đồ quan hệ GiangVien(MaGiangVien, TenGiangVien, CapHocVi, ChuyenNganh) với tân từ: "Mọi giảng viên đều có họ tên, cấp học vị thuộc một chuyên ngành nhất định và được gán cho một mã số duy nhất, gọi là MaGiangVien, để phân biệt với mọi giảng viên khác trong trường".

Lược đồ quan hệ LopHoc(MaLop, TenLop, NienKhoa, SoSinhVien, Makhoa) với tân từ: "Mỗi lớp học trong trường có một mã số quy ước duy nhất để phân biệt với tất cả các lớp học khác trong trường, có một tên gọi của lớp học, một số lượng học viên theo học và thuộc một khoa của trường".

Lược đồ quan hệ MonHoc(MaMon, TenMon, SoTC) với tân từ: "Mỗi môn học có một tên gọi cụ thể, được học trong một số tín chỉ nhất định và ứng với môn học là một mã số duy nhất để phân biệt với mọi môn học khác".

SinhVien(MaSinhVien, TenSinhVien, NgaySinh, QueQuan, MaLop) với tân từ: "Mỗi sinh viên có một họ và tên, ngày sinh, quê quán...và được cấp một mã số duy nhất để phân biệt với các sinh viên khác trong trường. Sinh viên được ghi danh vào một lớp học duy nhất trong trường".

Nhiều lược đồ quan hệ cùng nằm trong một hệ thống quản lý được gọi là *một lược đồ CSDL*.

Ví dụ: Lược đồ CSDL để quản lý điểm sinh viên có thể gồm các lược đồ quan hệ sau:

Khoa(MaKhoa, TenKhoa)

LopHoc(MaLop, TenLop, NienKhoa, SoSinhVien, Makhoa)

MonHoc(MaMon, TenMon, SoTC)

SinhVien(MaSinhVien, TenSinhVien, NgaySinh, QueQuan, MaLop)

KetQua(MaSinhVien, MaMon, LanThi, DiemThi)

3.2.5. Bộ (tuple)

Một bộ là các thông tin của một đối tượng thuộc quan hệ.

Bộ cũng thường được gọi là mẫu tin hay bản ghi (record) hoặc dòng của bảng (Row).

Ví dụ: Đây là 4 bộ dựa trên các thuộc tính của quan hệ HocVien:

$q_1 = (\text{SV001}, \text{Nguyễn Văn Nam}, 27/03/1970, \text{Cần Thơ}, \text{QTKD1})$

$q_2 = (\text{SV005}, \text{Vũ Thị Tuyết Mai}, 26/02/1968, \text{Đồng Nai}, \text{KTKC1})$

$q_3 = (\text{SV014}, \text{Hồng Đăng}, 30/04/1975, \text{Đồng Nai}, \text{CNTK3})$

$q_4 = (\text{SV015}, \text{Lê Hoài Nhở}, 23/03/1965, \text{Long An}, \text{CNTK4})$

Để lấy thành phần A_i (tức là giá trị thuộc tính A_i) của bộ giá trị q , ta viết $q.A_i$.

3.2.6. Thể hiện của quan hệ (view)

Thể hiện (hoặc còn gọi là tình trạng) của quan hệ R , ký hiệu bởi T_R , là tập hợp các bộ của quan hệ R vào một thời điểm.

Ví dụ: Bảng 3.1 và 3.2 là các thể hiện của quan hệ LopHoc và MonHoc:

Bảng 3.1. Quan hệ LopHoc

MaLop	TenLop	NienKhoa	SoSinhVien	MaKhoa
QTKD1	Quản trị kinh doanh QT01	96-99	145	QTKD
KTCK1	Tài chính - Kế toán KT4	96-99	230	TCKT
KTCK2	Tài chính - Kế toán KT5	97-2000	120	TCKT

Bảng 3.2. Quan hệ MonHoc

MaMon	TenMon	SoTC
TCKT	Tài chính - kế toán	3
KTCT	Kinh tế chính trị	3
TOANC	Toán Cơ sở	3
LTCBC	Lập trình căn bản C	3

3.2.7. Khóa (key)

Siêu khóa (superkey) của một lược đồ quan hệ R là một tập hợp gồm một hay nhiều thuộc tính của lược đồ R có tính chất xác định duy nhất một bộ trong mỗi thể hiện của R .

Cho lược đồ quan hệ R , $S \subseteq R^+$. S được gọi là một siêu khóa của lược đồ quan hệ R nếu với hai bộ tùy ý trong quan hệ R thì giá trị của các thuộc tính trong S là khác nhau.

Một lược đồ quan hệ có thể có nhiều siêu khóa. Siêu khóa chứa ít thuộc tính nhất được gọi là khóa chỉ định, trong trường hợp lược đồ quan hệ có nhiều khóa chỉ định, thì khóa được chọn để cài đặt gọi là *khóa chính* (primary key) (trong các phần sau khóa chính được gọi tắt là khóa).

Khóa (primary key) của một lược đồ quan hệ là một siêu khóa của lược đồ này sao cho mọi tập con thực sự của nó không phải là siêu khóa.

Một lược đồ quan hệ có thể có nhiều hơn một khóa, khi đó mỗi một khóa được gọi là một *khóa dự tuyển*.

Các khóa này được phân biệt bằng cách thêm dấu gạch chân phía dưới.

Ví dụ:

MonHoc(MaMon, TenMon, SoTC): Khóa chính là MaMon. Như vậy với tất cả các bộ trong lược đồ quan hệ này thì MaMon không trùng nhau.

KetQua(MaSinhVien, MaMon, LanThi, DiemThi): Khóa dự tuyển là MaSinhVien, MaMon. Như vậy không có bộ nào trong lược đồ quan hệ này có cả MaSinhVien và MaMon trùng với bộ khác.

Các thuộc tính tham gia vào một khóa được gọi là *thuộc tính khóa* (prime key), ngược lại được gọi là *thuộc tính không khóa* (non prime key).

Một thuộc tính được gọi là khóa ngoại (foreign key) nếu nó là thuộc tính của một lược đồ quan hệ này nhưng lại là khóa chính của lược đồ quan hệ khác.

Ví dụ: Với các lược đồ:

LopHoc(MaLop, TenLop, NienKhoa, SoSinhVien, Makhoa)

SinhVien (MaSinhVien, TenSinhVien, NgaySinh, QueQuan, MaLop)

MaLop là khóa ngoại trong lược đồ quan hệ SinhVien, vì nó là thuộc tính trong lược đồ quan hệ này nhưng lại là khóa chính trong lược đồ quan hệ LopHoc.

Chú ý:

- Một bảng được gọi là một quan hệ (relation).
- Một dòng trong bảng được gọi là một bộ (tuple).
- Một cột trong bảng gọi là một thuộc tính (attribute).
- Số lượng các cột trong bảng được gọi là bậc (degree) của bảng.
- Một cột hoặc tập hợp một số cột xác định duy nhất một dòng bên trong bảng được gọi là khóa chính của bảng (primary key).

3.3. Tính toàn vẹn của quan hệ

3.3.1. Khái niệm ràng buộc toàn vẹn

Trong mỗi CSDL luôn tồn tại nhiều mối liên hệ giữa các thuộc tính, giữa các bộ; sự liên hệ này có thể xảy ra trong cùng một quan hệ hoặc trong các quan hệ của một lược đồ CSDL. Các mối liên hệ này là những điều kiện bất biến mà tất cả các bộ của những quan hệ có liên quan trong CSDL đều phải thỏa mãn ở mọi thời điểm. Những điều kiện bất biến đó được gọi là ràng buộc toàn vẹn.

Trong thực tế ràng buộc toàn vẹn là các quy tắc quản lý được áp đặt trên các đối tượng của thế giới thực. Chẳng hạn mỗi sinh viên phải có một mã sinh viên duy nhất, hai thí sinh dự thi vào một trường phải có số báo danh khác nhau, một sinh viên dự thi một môn học không quá 3 lần,...

Nhiệm vụ của người phân tích thiết kế là phải phát hiện càng đầy đủ các ràng buộc toàn vẹn càng tốt và mô tả chúng một cách chính xác trong hồ sơ phân tích thiết kế - đó là một việc làm rất quan trọng. Ràng buộc toàn vẹn được xem như là một công cụ để diễn đạt ngữ nghĩa của CSDL. Một CSDL được thiết kế công kênh nhưng nó thể hiện được đầy đủ ngữ nghĩa của thực tế vẫn có giá trị cao hơn rất nhiều so với một cách thiết kế gọn nhẹ nhưng nghèo nàn về ngữ nghĩa vì thiếu các ràng buộc toàn vẹn của CSDL.

Công việc kiểm tra ràng buộc toàn vẹn thường được tiến hành vào thời điểm cập nhật dữ liệu (thêm, sửa, xóa). Những ràng buộc toàn vẹn phát sinh phải cần được ghi nhận và xử lý một cách tường minh (thường là bởi một hàm chuẩn hoặc một đoạn chương trình).

Ràng buộc toàn vẹn và kiểm tra sự vi phạm ràng buộc toàn vẹn là hai trong số những vấn đề quan trọng trong quá trình phân tích thiết kế cơ sở dữ liệu, nếu không quan tâm đúng mức đến những vấn đề trên, thì có thể dẫn đến những hậu quả nghiêm trọng về tính an toàn và toàn vẹn dữ liệu, đặc biệt là đối với những cơ sở dữ liệu lớn.

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

Null biểu thị một giá trị đặc biệt cho một thuộc tính của một bộ trong trường hợp không biết giá trị của thuộc tính này (tính đến thời điểm đang xét) hay không thể áp dụng thuộc tính này cho bộ đó.

➤ Các yếu tố của ràng buộc toàn vẹn :

Mỗi ràng buộc toàn vẹn có bốn yếu tố: điều kiện, bối cảnh, bảng tầm ảnh hưởng và hành động phải cần thực hiện khi phát hiện có ràng buộc toàn vẹn bị vi phạm.

- Điều kiện:

Điều kiện của ràng buộc toàn vẹn là sự mô tả, và biểu diễn hình thức nội dung của nó.

Điều kiện của một ràng buộc toàn vẹn R có thể được biểu diễn bằng ngôn ngữ tự nhiên, ngôn ngữ đại số quan hệ, ngôn ngữ mã giả, ngôn ngữ truy vấn SQL,... ngoài ra điều kiện của ràng buộc toàn vẹn cũng có thể được biểu diễn bằng phụ thuộc hàm.

Sau đây là một số ràng buộc toàn vẹn trên lược đồ CSDL quản lý sinh viên:

+ Mỗi lớp học phải có một mã số duy nhất để phân biệt với các lớp học khác trong trường.

+ Mỗi lớp học phải thuộc về một khoa của trường.

+ Mỗi sinh viên có một mã số sinh viên duy nhất, không trùng với bất cứ sinh viên nào trong trường.

+ Mỗi học viên phải đăng ký vào một lớp học trong trường. Mỗi học viên chỉ được thi tối đa 3 lần cho mỗi môn học.

+ Tổng số học viên của một lớp phải lớn hơn hoặc bằng số lượng đếm được của một lớp tại một thời điểm nào đó.

- Bối cảnh:

Bối cảnh của ràng buộc toàn vẹn là những quan hệ mà ràng buộc đó có hiệu lực hay nói một cách khác, đó là những quan hệ cần phải được kiểm tra khi tiến hành cập nhật dữ liệu. Bối cảnh của một ràng buộc toàn vẹn có thể là một hoặc nhiều quan hệ.

Chẳng hạn với ràng buộc toàn vẹn R trên thì bối cảnh của nó là quan hệ Sinhvien.

- Bảng tầm ảnh hưởng:

Trong quá trình phân tích thiết kế một CSDL, người phân tích cần lập bảng tầm ảnh hưởng cho một ràng buộc toàn vẹn nhằm xác định thời điểm cần phải tiến hành kiểm tra khi tiến hành cập nhật dữ liệu.

Thời điểm cần phải kiểm tra ràng buộc toàn vẹn chính là thời điểm cập nhật dữ liệu.

Một bảng tầm ảnh hưởng của một ràng buộc toàn vẹn có dạng sau :

Tên RBTV	Thêm(T)	Sửa(S)	Xoá(X)
r_1	+		
r_2		-	
r_3			-(*)
r_n			

Bảng này chứa toàn các ký hiệu + , – hoặc -(*).

Chẳng hạn + tại (dòng r_1 , cột Thêm) thì có nghĩa là khi thêm một bộ vào quan hệ r_1 thì RBTV bị vi phạm.

Dấu - Tại ô (dòng r_2 , cột sửa) thì có nghĩa là khi sửa một bộ trên quan hệ r_2 thì RBTV không bị vi phạm.

Quy ước :

+ Không được sửa thuộc tính khoá.

+ Nếu không bị vi phạm do không được phép sửa đổi thì ký hiệu là -(*)

- Hành động cần phải có khi phát hiện có RBTV bị vi phạm:

Khi một ràng buộc toàn vẹn bị vi phạm, cần có những hành động thích hợp. Thông thường có 2 giải pháp :

Thứ nhất: Đưa ra thông báo và yêu cầu sửa chữa dữ liệu của các thuộc tính cho phù hợp với quy tắc đảm bảo tính nhất quán dữ liệu. Thông báo phải đầy đủ và phải thân thiện với người sử dụng. Giải pháp này là phù hợp cho việc xử lý thời gian thực.

Thứ hai: Từ chối thao tác cập nhật. Giải pháp này là phù hợp đối với việc xử lý theo lô. Việc từ chối cũng phải được lưu lại bằng những thông báo đầy đủ, rõ ràng vì sao thao tác bị từ chối và cần phải sửa lại những dữ liệu nào?

Khóa nội, khóa ngoại, giá trị NOT NULL là những ràng buộc toàn vẹn miền giá trị của các thuộc tính. Những ràng buộc toàn vẹn này là những ràng buộc toàn vẹn đơn giản trong CSDL.

Các hệ quản trị cơ sở dữ liệu thường có các cơ chế tự động kiểm tra các ràng buộc toàn vẹn về miền giá trị của khóa nội, khóa ngoại, giá trị NOT NULL.

Việc kiểm tra ràng buộc toàn vẹn có thể tiến hành vào những thời điểm sau đây:

Thứ nhất: Kiểm tra ngay sau khi thực hiện một thao tác cập nhật CSDL. Thao tác cập nhật chỉ được xem là hợp lệ nếu như nó không vi phạm bất cứ một ràng buộc toàn vẹn nào, nghĩa là nó không làm mất tính toàn vẹn của CSDL. Nếu vi phạm ràng buộc toàn vẹn, thao tác cập nhật bị coi là không hợp lệ và sẽ bị hệ thống hủy bỏ (hoặc có một xử lý thích hợp nào đó).

Thứ hai: Kiểm tra định kỳ hay đột xuất, nghĩa là việc kiểm tra ràng buộc toàn vẹn được tiến hành độc lập với thao tác cập nhật dữ liệu. Đối với những trường hợp vi phạm ràng buộc toàn vẹn, hệ thống có những xử lý ngầm định hoặc yêu cầu người sử dụng xử lý những sai sót một cách tường minh.

3.3.2. Ràng buộc toàn vẹn thực thể

Toàn vẹn thực thể là ràng buộc phát biểu rằng: *Trong một quan hệ, giá trị của các bộ tại thuộc tính có tham gia vào khóa chính không được là rỗng (null).*

Ví dụ: Cho quan hệ: SinhVien(MaSinhVien, TenSinhVien, MaLop)

Quan hệ SinhVien có khóa chính là MaSinhVien. Với tất cả các bộ, giá trị tại thuộc tính MaSinhVien không được là rỗng.

Bảng 3.3. Quan hệ SinhVien

Không
được rỗng

<u>MaSinhVien</u>	TenSinhVien	MaLop
001	Trần Thị Lan Anh	CNTT-01
002	Thái Thiện Kỳ	CNTT-01
003	Nguyễn Văn Hải	CNTT-01

3.3.3. Ràng buộc toàn vẹn tham chiếu

Toàn vẹn tham chiếu là ràng buộc phát biểu rằng: *Giá trị của các bộ tại khóa ngoại hoặc phải bằng một giá trị tại khóa chính của một bộ nào đó trong quan hệ khác hoặc phải bằng rỗng.*

Ví dụ:

Bảng 3.4. Quan hệ Lop và SinhVien

<u>MaLop</u>	TenLop
CNTT-01	Công nghệ thông tin
TMĐT-02	Thương mại điện tử
CNTTUD-03	Công nghệ thông tin ứng dụng

<u>MaSinhVien</u>	TenSinhVien	MaLop
001	Trần Thị Lan Anh	CNTT-01
002	Thái Thiện Kỳ	CNTT-01
003	Nguyễn Văn Hải	CNTT-01

Quan hệ Lop(MaLop,TenLop) có MaLop là khóa chính.

Quan hệ SinhVien(MaSinhVien,TenSinhVien,MaLop) có MaLop là khóa ngoại.

Như vậy với tất cả các bộ thuộc bảng SinhVien, giá trị của MaLop phải bằng giá trị của MaLop tại một bộ nào đó của bảng Lop hoặc rỗng.

3.3.4. Ràng buộc toàn vẹn ngữ nghĩa

Ràng buộc toàn vẹn ngữ nghĩa, hay còn gọi là các ràng buộc chung, là các quy luật được chỉ định bởi NSD hay người quản trị CSDL của một CSDL nào đó, nhằm định nghĩa hay ràng buộc một số khía cạnh nào đó của hoạt động.

Ví dụ:

Trong CSDL về quản lý sinh viên của một trường học đã trình bày ở phần 3.3.4, chúng ta có một số ràng buộc toàn vẹn ngữ nghĩa như sau:

- R₁ : Mỗi lớp học phải thuộc một Khoa của trường.
- R₂ : Mỗi lớp học phải có một mã số duy nhất để phân biệt với mọi lớp học khác trong trường.
- R₃ : Mỗi sinh viên có một mã số riêng biệt, không trùng với bất cứ sinh viên nào khác.
- R₄ : Mỗi sinh viên phải đăng ký vào một lớp của trường.
- R₅ : Mỗi sinh viên được thi tối đa 2 lần cho mỗi môn học.
- R₆ : Tổng số sinh viên của một lớp phải lớn hơn hoặc bằng số lượng đếm được của lớp tại một thời điểm.

3.4. Các ngôn ngữ quan hệ

Các ngôn ngữ thao tác dữ liệu được phát triển cho mô hình quan hệ (thường gọi là ngôn ngữ vấn tin, query language) được chia làm hai nhóm căn bản:

- Các ngôn ngữ dựa trên đại số quan hệ (relational algebra)
- Các ngôn ngữ dựa trên phép tính quan hệ (relational calculus).

Cả hai ngôn ngữ được Codd đưa ra năm 1970 và ông đã chứng minh rằng chúng tương đương về khả năng diễn tả.

3.4.1. Đại số quan hệ

Đại số quan hệ có một tập các phép toán trên các quan hệ. Chúng có nguồn gốc từ lý thuyết tập hợp (mỗi quan hệ thực chất là một tập hợp). Mỗi toán tử nhận một hoặc hai quan hệ làm toán hạng và cho ra một quan hệ mới (quan hệ kết quả), đến lượt nó quan hệ kết quả có thể dùng làm toán hạng cho một toán tử khác. Những phép toán này cho phép vấn tin và cập nhật cơ sở dữ liệu quan hệ.

3.4.1.1. Phép chiếu

Phép chiếu của một quan hệ r trên tập thuộc tính X dùng để xây dựng một quan hệ mới từ quan hệ r đã cho bằng cách loại bỏ đi một số thuộc tính của quan hệ đó. Những thuộc tính bị loại bỏ là những thuộc tính không thuộc tập X.

Cho r là một quan hệ n ngôi xác định trên tập thuộc tính $U=\{A_1, \dots, A_n\}$ và một tập con thuộc tính X . Phép chiếu của quan hệ r trên tập thuộc tính X , ký hiệu $\pi_X(r)$, là tập các bộ của r xác định trên tập thuộc tính X .

Biểu diễn hình thức của phép toán này là:

$$\pi_X(r) = \{t[X] \mid t \in r\}$$

Ví dụ:

Cho quan hệ r với các thuộc tính A, B, C . Phép chiếu của r lên hai thuộc tính A và C , ký hiệu $\pi_{A,C}(r)$, cho kết quả như sau:

r			\Rightarrow	$\pi_{A,C}(r)$	
<u>A</u>	<u>B</u>	<u>C</u>		<u>A</u>	<u>C</u>
a	b	c		a	c
d	a	f		d	f
c	b	d		c	d

Việc chiếu quan hệ r lên hai thuộc tính A và C thực chất là chỉ giữ lại các giá trị của A và C , bỏ đi các giá trị còn lại.

3.4.1.2. Phép chọn

Phép chọn là phép toán lọc ra một tập con các bộ của quan hệ đã cho thỏa mãn một điều kiện xác định. Điều kiện đó được gọi là điều kiện chọn hay biểu thức chọn.

Biểu thức chọn F được định nghĩa là một tổ hợp logic các toán hạng, mỗi toán hạng là một phép so sánh đơn giản giữa hai biến là hai thuộc tính hoặc giữa một biến là thuộc tính và một giá trị hằng. Biểu thức chọn F cho giá trị đúng hoặc sai đối với mỗi bộ đã cho của quan hệ khi kiểm tra riêng bộ đó.

- Các phép toán so sánh trong biểu thức F là: $<, =, >, \neq, \leq, \geq$
- Các phép toán logic trong biểu thức F là: $\&, \wedge$ (và), \vee (hoặc), \neg (phủ định).

Cho r là một quan hệ và F là một biểu thức logic trên các thuộc tính của r . Phép chọn của r theo biểu thức F , ký hiệu $\sigma_F(r)$, là tập hợp tất cả các bộ của r thỏa mãn F .

Biểu diễn hình thức của phép chọn trên quan hệ r được định nghĩa như sau:

$$\sigma_F(r) = \{t \mid t \in r \wedge F(t) = \text{đúng}\}$$

Ví dụ:

Cho quan hệ r với các thuộc tính A, B, C . Phép chọn của r theo biểu thức $A=a$, ký hiệu $\sigma_{A=a}(r)$, cho kết quả như sau:

r			\Rightarrow	$\sigma_{A=a}(r)$		
<u>A</u>	<u>B</u>	<u>C</u>		<u>A</u>	<u>B</u>	<u>C</u>
a	b	c		a	b	c
d	a	f				

c b d

Trong tất cả các bộ của quan hệ r , chỉ có bộ đầu tiên thuộc tính A có giá trị a , do đó chúng ta loại bỏ đi các bộ còn lại.

3.4.1.3. Phép hợp

Cho r, s là quan hệ n ngôi xác định trên tập thuộc tính $U=\{A_1, \dots, A_n\}$. Phép hợp của hai quan hệ r và s , ký hiệu $r \cup s$, là tập hợp tất cả các bộ thuộc r hoặc thuộc s hoặc thuộc cả hai quan hệ.

Biểu diễn hình thức của phép hợp có dạng: $r \cup s = \{t \mid t \in r \vee t \in s\}$

Ví dụ:

Cho quan hệ r và s với các thuộc tính A, B, C . Phép hợp của r và s , ký hiệu $r \cup s$, như sau:

r	s	$r \cup s$
<u>A B C</u>	<u>A B C</u>	<u>A B C</u>
a b c	b g a	a b c
d a f	d a f	d a f
c b d	c b d	b g a
		c b d

Kết quả của phép hợp r và s là tất cả các bộ của r và s , các bộ trùng nhau chỉ tính một lần.

3.4.1.4. Phép hiệu

Cho r, s là quan hệ n ngôi xác định trên tập thuộc tính $U=\{A_1, \dots, A_n\}$. Hiệu của r và s , ký hiệu $r - s$, là tập hợp tất cả các bộ thuộc r nhưng không thuộc s .

Biểu diễn hình thức của phép trừ có dạng: $r - s = \{t \mid t \in r \wedge t \notin s\}$

Ví dụ: Cho quan hệ r và s với các thuộc tính A, B, C . Phép hiệu của r và s , ký hiệu $r - s$, cho kết quả như sau:

r	s	$r - s$
<u>A B C</u>	<u>A B C</u>	<u>A B C</u>
a b c	b g a	a b c
d a f	d a f	

Kết quả của phép hiệu r và s là tất cả các bộ thuộc r mà không thuộc s . Trong ví dụ trên chỉ có bộ đầu tiên của r thỏa mãn điều này.

3.4.1.5. Phép giao

Cho r, s là quan hệ n ngôi xác định trên tập thuộc tính $U=\{A_1, \dots, A_n\}$. Giao của r và s , ký hiệu $r \cap s$, là tập hợp tất cả các bộ thuộc cả hai quan hệ r và s .

Biểu diễn hình thức phép giao có dạng: $r \cap s = \{t \mid t \in r \wedge t \in s\}$

Công thức: Phép giao suy ra từ phép hiệu: $r \cap s = r - (r - s)$

Ví dụ: Cho quan hệ r và s với các thuộc tính A, B, C . Phép giao của r và s , ký hiệu $r \cap s$, cho kết quả như sau:

r			s			r ∩ s			
<u>A</u>	<u>B</u>	<u>C</u>	<u>A</u>	<u>B</u>	<u>C</u>	\Rightarrow	<u>A</u>	<u>B</u>	<u>C</u>
a	b	c	b	g	a		d	a	f
d	a	f	d	a	f				

Kết quả của phép giao r và s là tất cả các bộ vừa thuộc r vừa thuộc s , chỉ có bộ thứ hai của r , cũng là bộ thứ hai của s thoả mãn điều này.

3.4.1.6. Tích Đề-các

Cho quan hệ r xác định trên tập thuộc tính $\{A_1, \dots, A_n\}$ và quan hệ s xác định trên tập thuộc tính $\{B_1, \dots, B_m\}$. Tích Đề-các của hai quan hệ r và s , ký hiệu $r \times s$, là tập hợp tất cả các $(n+m)$ bộ có n thành phần đầu là một bộ thuộc r và m thành phần sau là một bộ thuộc s .

Biểu diễn hình thức của phép toán này:

$$r \times s = \{t \mid t = (a_1, \dots, a_n, b_1, \dots, b_m) \wedge (a_1, \dots, a_n) \in r \wedge (b_1, \dots, b_m) \in s\}$$

Ví dụ:

Cho quan hệ r với các thuộc tính A, B, C và s với các thuộc tính D, E, F . Tích Đề-các của r và s , ký hiệu $r \times s$ cho kết quả như sau:

r			s			r × s						
<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>F</u>	⇒	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>F</u>
a	b	c	b	g	a		a	b	c	b	g	a
d	a	f	d	a	f		a	b	c	d	a	f
c	b	d					d	a	f	b	g	a
							d	a	f	d	a	f
							c	b	d	b	g	a
							c	b	d	d	a	f

Kết quả của phép hợp r và s là các bộ, mỗi bộ gồm 3 thành phần thuộc một bộ của r được ghép với 3 thành phần thuộc một bộ của s .

3.4.1.7. Phép nối

Cho quan hệ r xác định trên tập thuộc tính $\{A_1, \dots, A_n\}$ và quan hệ s xác định trên tập thuộc tính $\{B_1, \dots, B_m\}$. Cho biểu thức logic F . Phép nối của quan hệ r và quan hệ s theo điều kiện nối F , ký hiệu:

$$r \bowtie_F s$$

là tập hợp tất cả các $(n+m)$ bộ (t, u) thoả $t \in r, u \in s$ và khi thay thế các giá trị của t và u vào F thì chúng ta được giá trị đúng.

Biểu diễn hình thức: $r \bowtie_F s = \{t \mid t=(u,v) \wedge u \in r \wedge v \in s \wedge F(t)=\text{đúng}\}$

Ở đây F là biểu thức logic gồm các toán hạng dạng $A\theta B$, trong đó A là thuộc tính của r và B là thuộc tính s và θ là phép toán so sánh.

Chúng ta có thể biểu diễn:

$$r \bowtie_F s = \sigma_F(r \times s)$$

Nếu các toán tử so sánh trong biểu thức F đều là phép bằng ($=$) thì phép nối theo F được gọi là phép đẳng nối (equijoin).

Ví dụ:

Cho quan hệ r với các thuộc tính A, B, C và s với các thuộc tính D, E . Phép nối của r với s theo biểu thức $B < D$, ký hiệu:

$$r \bowtie_{B < D} s$$

cho kết quả như sau:

r			s		$r \bowtie_{B < D} s$				
<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>
1	2	3	3	1	1	2	3	3	1
4	5	6	6	2	1	2	3	6	2
7	8	9			4	5	6	6	2

Chẳng hạn khi xét bộ đầu tiên của r và bộ đầu tiên của s thì thỏa mãn biểu thức $B < D$, kết quả là bộ đầu tiên của phép nối.

Tương tự như vậy, nhưng nếu chúng ta xét với điều kiện $B = D$ thì gọi là phép đẳng nối.

3.4.1.8. Phép chia

Cho quan hệ r xác định trên tập thuộc tính $U = \{A_1, \dots, A_n\}$ và quan hệ s xác định trên tập thuộc tính $V = \{B_1, \dots, B_m\}$, trong đó $m < n$. Phép chia quan hệ r cho quan hệ s , ký hiệu $r \div s$, là tập hợp tất cả các bộ t sao cho với mọi bộ $v \in s$ thì t ghép với v sẽ thuộc r .

Biểu diễn hình thức: $r \div s = \{t \mid \forall v \in s \Rightarrow (t, v) \in r\}$

Ví dụ: Cho quan hệ r với các thuộc tính A, B, C, D và s với các thuộc tính C, D . Phép chia của r cho s , ký hiệu $r \div s$, cho kết quả như sau:

r				s		$r \div s$	
<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>C</u>	<u>D</u>	<u>A</u>	<u>B</u>
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

Ở ví dụ trên, với mỗi bộ của r, nếu hai thành phần C và D tạo thành một bộ thuộc s, thì hai thành phần A và B sẽ tạo thành một bộ thuộc kết quả của phép chia.

3.4.1.9. Các chương trình đại số quan hệ

Vì tất cả các phép toán đại số đều nhận quan hệ làm đối biến và sinh ra các quan hệ kết quả, chúng ta có thể lồng ghép các phép toán này bằng các dấu ngoặc đơn và sinh ra các chương trình đại số quan hệ.

Ví dụ:

Cho các quan hệ sau:

EMP(ENO, ENAME, TITLE): quan hệ về nhân viên, gồm các thuộc tính Mã nhân viên (ENO), tên nhân viên (ENAME), chức vụ (TITLE).

PAY(TITLE, SAL): quan hệ về mức lương của từng chức vụ, gồm các thuộc tính chức vụ (TITLE), mức lương (SAL).

PROJ(PNO, PNAME, BUDGET): quan hệ về dự án, gồm các thuộc tính mã dự án (PNO), tên dự án (PNAME), kinh phí của dự án (BUDGET).

ASG(ENO, PNO, RESP, DUR): quan hệ về việc nhân viên tham gia vào các dự án, gồm các thuộc tính mã nhân viên (ENO), mã dự án (PNO), vai trò của nhân viên trong dự án (RESP), khoảng thời gian nhân viên tham gia dự án (DUR).

- Tìm tên tất cả nhân viên đang làm việc cho dự án CAD/CAM

$$\pi_{ENAME}(((\sigma_{PNAME='CAD/CAM'}(PROJ)) \times ASG) \times EMP)$$

Thứ tự thực hiện như sau: thực hiện phép chọn trên PROJ, sau đó nối tự nhiên với ASG, theo sau là nối tự nhiên với EMP, và cuối cùng là chiếu lên thuộc tính ENAME.

- Tăng lương các lập trình viên (programmer) lên 25000 USD.

$$(PAY - (\sigma_{TITLE='Programmer'}(PAY))) \cup (<'Programmer', 25000>)$$

Trước hết sử dụng phép chọn để tìm chức danh lập trình viên trong PAY. Sau đó sử dụng phép hiệu để loại bỏ chức danh lập trình viên đi bằng cách chỉ giữ lại các chức danh không phải lập trình viên. Cuối cùng sử dụng phép hợp để thêm chức danh lập trình viên với mức lương 25000 vào.

Ví dụ: Cho các quan hệ sau:

NHÂNVIÊN(Mã số NV, Họ đệm, Tên, Ngày sinh, Địa chỉ, Giới tính, Lương, Mã số NGS, Mã số ĐV)

ĐƠN VỊ(Mã số ĐV, Tên ĐV, Mã số NQL, Ngày bắt đầu)

DỰ ÁN(Tên DA, Mã số DA, Địa điểm DA, Mã số ĐV)

NHÂNVIÊN_DỰ ÁN(Mã số NV, Mã số DA, Số giờ)

- Truy vấn 1: Đưa ra Họ đệm, Tên và địa chỉ của tất cả các nhân viên làm việc cho đơn vị có tên là “Nghiên cứu”: (Các quan hệ TG1, TG2 là các kết quả trung gian)

$$TG1 \leftarrow \sigma_{Tên ĐV = "Nghiên cứu"}(ĐƠN VỊ)$$

$$TG2 \leftarrow (TG1 * NHÂNVIÊN)$$

$$KETQUA \leftarrow \pi_{Họđệm, Tên, Địachỉ} (TG2)$$

Theo cách thực hiện này, quan hệ TG1 chứa thông tin về đơn vị có tên “Nghiên cứu”, quan hệ TG2 chứa thông tin về các nhân viên làm việc cho đơn vị “Nghiên cứu” và quan hệ KẾTQUẢ chứa các thông tin theo yêu cầu của truy vấn.

Trong các bảng của chúng ta, các thuộc tính nối có tên như nhau nên có thể dùng phép nối tự nhiên.

- Truy vấn 2: Với mỗi dự án đặt tại Hà nội, hãy liệt kê MãSốDA, TênĐV, Tên, Địachỉ, Ngàysinh của người quản lý đơn vị:

$$TG1 \leftarrow \sigma_{DiadiemDA="Hanoi"} (DỰÁN)$$

$$TG2 \leftarrow (TG1 * ĐƠNVI)$$

$$TG3 \leftarrow (TG2 * NHÂNVIÊN)$$

$$KẾTQUẢ \leftarrow \pi_{MãSốDA, MãSốĐV, Họđệm, Địachỉ, Ngàysinh} (TG3)$$

- Truy vấn 3: Hãy tìm tên của các nhân viên làm việc trên tất cả các dự án do đơn vị có mã số = 5 kiểm soát.

$$TG1 \leftarrow \pi_{MãSốDA} (\sigma_{Mã sốĐV = 5} (DỰÁN))$$

$$TG2 \leftarrow \pi_{MãSốNV, MãSốDA} (NHÂNVIÊN_DỰÁN)$$

$$TG3 \leftarrow TG2 \div TG1$$

$$KETQUA \leftarrow \pi_{Họđệm, Tên} (TG3 * NHÂNVIÊN)$$

3.4.2. Phép tính quan hệ

Trong các ngôn ngữ dựa trên phép tính quan hệ, thay vì xác định xem phải làm thế nào để thu được kết quả, chúng ta sẽ xác định xem kết quả là gì bằng cách đưa ra mối liên hệ được giả sử là đúng đối với kết quả.

Ngôn ngữ phép tính quan hệ được chia làm 2 nhóm: *phép tính quan hệ bộ* (tuple relational calculus) và *phép tính quan hệ miền* (domain relational calculus). Sự khác biệt giữa chúng là ở các biến nguyên thủy được dùng khi xác định các câu vấn tin.

Một cơ sở dữ liệu quan hệ có thể xem như tập các bộ hoặc tập các miền. Phép tính quan hệ bộ diễn giải các biến trong công thức như một bộ của quan hệ, còn phép tính quan hệ miền diễn giải biến như giá trị của miền.

3.4.2.1. Phép tính quan hệ bộ (Codd 1970)

Biến nguyên thủy dùng trong phép tính quan hệ bộ là biến bộ (tuple variable), biểu thị một bộ của quan hệ. Nói cách khác biến này biến thiên trên các bộ của quan hệ.

Công thức nguyên tử có hai dạng:

(1) Biểu thức kiểu phân tử biến bộ

Nếu t là một biến bộ biến thiên trên các bộ của một quan hệ R , biểu thức “bộ t thuộc quan hệ R ” là công thức nguyên tử, và được viết là $R.t$ hoặc $R(t)$.

(2) Điều kiện

Loại công thức nguyên tử này có thể định nghĩa như sau:

(i) $s[A] \theta t[B]$, trong đó s và t là các biến bộ và A và B là các thành phần tương ứng của s và t , θ là một trong các toán tử so sánh $<, >, =, \leq, \geq$ và \neq .

(ii) $s[A] \theta c$, trong đó s , A và θ định nghĩa giống như trên (i) và c là hằng.

Hiện có nhiều ngôn ngữ dựa trên phép tính quan hệ bộ, và ngôn ngữ thông dụng nhất là SQL và QUEL. SQL hiện là chuẩn quốc tế (duy nhất) với nhiều phiên bản chuẩn hoá (Xem Chương 4 – Ngôn ngữ truy vấn dữ liệu).

3.4.2.2. Phép tính quan hệ miền

Thành công của ngôn ngữ phép tính quan hệ miền chủ yếu do QBE (Query by example) (Zloof, 1977) đem lại. QBE được thiết kế dành cho kiểu làm việc tương tác từ thiết bị đầu cuối trực quan và thân thiện, cho phép NSD xây dựng các truy vấn dữ liệu bằng cửa sổ thiết kế và bằng các công cụ có sẵn.

Khái niệm cơ bản là example: NSD đưa ra các câu văn tin bằng cách cung cấp một example có thể có của câu trả lời. Hành động gõ tên quan hệ sẽ kích hoạt việc hiển thị các lược đồ của chúng lên màn hình. Sau đó bằng cách cung cấp các từ khóa trong các cột (miền), người dùng đặc tả câu văn tin.

Ví dụ: Với các quan hệ ở phần 3.4.1.9, để tăng lương các lập trình viên (programmer) lên 25000 USD, sử dụng phép tính quan hệ miền như sau:

PAY	TITLE	SAL
	Programmer	U.25000

3.5. Chuyển đổi mô hình thực thể liên kết sang mô hình quan hệ

Sau khi xây dựng mô hình dữ liệu mức khái niệm, ta phải chuyển đổi mô hình đó thành một mô hình dữ liệu logic. Phần này sẽ trình bày về một số quy tắc thường được sử dụng chuyển đổi một mô hình ER thành ra mô hình quan hệ.

- Quy tắc 1: Chuyển đổi mỗi kiểu thực thể thành một lược đồ quan hệ, các thuộc tính của loại thực thể thành các thuộc tính của lược đồ quan hệ, thuộc tính khoá của loại thực thể là thuộc tính khoá của lược đồ quan hệ.

- Quy tắc 2: Nếu mỗi liên kết của cả hai nhánh đều là n thì mỗi liên kết này sẽ được chuyển thành một lược đồ quan hệ K' gồm các thuộc tính của mỗi liên kết K , cộng thêm các thuộc tính khoá của hai lược đồ quan hệ A, B tương ứng với hai thực thể tham gia vào mỗi liên kết. Khoá của lược đồ quan hệ K' gồm cả hai khoá của hai lược đồ quan hệ A và B .

- Quy tắc 3: Mỗi liên kết mà một nhánh là n (nhánh B) và nhánh còn lại là 1 (nhánh A) thì loại bỏ mỗi liên kết này khỏi mô hình thực thể liên kết và thêm các thuộc tính khoá của lược đồ tương ứng với kiểu thực thể ở nhánh B vào lược đồ tương ứng với kiểu thực thể ở nhánh A (khoá của B sẽ thành khoá ngoại của A). Nếu mỗi liên kết có các thuộc tính thì những thuộc tính này cũng được thêm vào lược đồ quan hệ tương ứng với loại thực thể ở nhánh A .

- Quy tắc 4: Nếu mỗi liên kết mà cả hai nhánh đều là 1 thì áp dụng quy tắc 3 cho một trong hai nhánh tùy chọn.

3.6. Khung nhìn (View)

Trong kiến trúc 3 mức (theo ANSI-PARC), chúng ta đã mô tả khung nhìn (hay mức ngoài) là cấu trúc của CSDL xuất hiện cho một người dùng cụ thể. Trong mô hình quan hệ, thuật ngữ khung nhìn có nghĩa khác đi một chút.

3.6.1. Khái niệm quan hệ cơ sở và khung nhìn

Một quan hệ cơ sở là một quan hệ được đặt tên, nó tương ứng với một thực thể trong lược đồ logic, các bộ của nó được lưu trữ vật lý trong CSDL.

Một khung nhìn là kết quả thực hiện một hay nhiều thao tác (phép toán) trên các quan hệ cơ sở nhằm đưa ra một quan hệ mới. Một khung nhìn là một quan hệ ảo, nghĩa là nó không thực sự tồn tại ở mức vật lý trong CSDL (như quan hệ cơ sở) mà là câu trả lời cho yêu cầu của người dùng, tại thời điểm yêu cầu.

Khung nhìn mang tính động, theo nghĩa những thay đổi trong quan hệ cơ sở sẽ tác động lập tức đến thể hiện của khung nhìn. *Một khung nhìn có thể được tạo dựng từ việc thực hiện các thao tác như các phép toán đại số quan hệ (chọn, chiếu, kết nối...) hay các phép tính quan hệ trên các quan hệ cơ sở.*

Như vậy mức ngoài của một CSDL có thể bao gồm cả các quan hệ cơ sở và cả các khung nhìn được suy dẫn, kết xuất từ các quan hệ cơ sở.

3.6.2. Mục đích của khung nhìn

Cơ chế sử dụng khung nhìn mang lại một số lợi ích sau đây:

- Cung cấp một cơ chế an ninh hữu hiệu và linh hoạt bởi che dấu một số phần của CSDL trước một số người dùng nào đó. Người dùng không hề được thông báo về sự tồn tại của một số thuộc tính hay một số bộ trong CSDL khi chúng không được xuất hiện trong khung nhìn.

- Cho phép người dùng truy cập được dữ liệu theo cách họ muốn, sao cho tại một thời điểm, cùng một dữ liệu có thể thấy bởi nhiều người dùng khác nhau trong những khuôn dạng khác nhau.

- Cho phép đơn giản hoá những phép toán trên các quan hệ cơ sở. Chẳng hạn, nếu một khung nhìn được định nghĩa là kết nối của hai quan hệ, thì người dùng có thể thực hiện một số phép toán một ngôi như chiếu, chọn trên khung nhìn.

3.6.3. Vấn đề cập nhật qua khung nhìn

Việc cập nhật cho các quan hệ cơ sở sẽ tác động ngay đến tất cả các khung nhìn có tham chiếu đến các quan hệ này. Tương tự, nếu một khung nhìn được cập nhật thì quan hệ cơ sở liên quan phải thay đổi. Tuy nhiên có những ràng buộc chặt chẽ cho việc cập nhật thông qua khung nhìn. Dưới đây là các nguyên tắc mà hầu hết các hệ thống dùng để kiểm soát và quyết định cho phép một cập nhật CSDL xảy ra qua khung nhìn:

- Được phép cập nhật thông qua một khung nhìn, nếu khung nhìn này được định nghĩa bằng một câu truy vấn đơn giản chỉ liên quan đến một quan hệ cơ sở và chứa hoặc khóa chính hoặc khóa dự tuyển của quan hệ cơ sở đó.

- Không được cập nhật thông qua một khung nhìn, nếu khung nhìn này liên quan đến nhiều quan hệ cơ sở.

- Không được cập nhật thông qua khung nhìn, nếu khung nhìn này liên quan đến (được xác định từ) các hàm kết tập và các hàm gộp nhóm.

Như vậy có những khung nhìn không thể cập nhật được, có những khung nhìn có thể cập nhật và có những khung nhìn chỉ cập nhật bộ phận.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 3

3.1. Hãy phân biệt các thuật ngữ sau:

- Thuộc tính và miền
- Quan hệ và lược đồ quan hệ
- Lược đồ quan hệ và lược đồ CSDL
- Siêu khóa, khóa, khóa dự tuyển, khóa ngoại
- Toàn vẹn thực thể và toàn vẹn tham chiếu

3.2. Vì sao sự xuất hiện hai bộ giống hệt nhau trong một quan hệ là không được phép?**3.3.** Hãy giải thích sự khác nhau giữa khóa dự tuyển và khóa chính. Vì sao phải có một khóa dự tuyển được chỉ định làm khóa chính.**3.4.** Hãy nói ra ý nghĩa của khóa ngoại. Cho ví dụ minh họa.**3.5.** Cho lược đồ cơ sở dữ liệu dùng để quản lý hồ sơ sinh viên được mô tả bởi các lược đồ quan hệ như sau:

Khoa(MaKhoa, TenKhoa)

Tên từ: Mỗi khoa có mỗi MAKHOA duy nhất. Mỗi MAKHOA xác định tất cả các thuộc tính còn lại của khoa đó.

Lop(MaLop, TenLop, SiSo, MaKhoa)

Tên từ: Mỗi lớp có một mã lớp duy nhất, mỗi lớp chỉ thuộc về một khoa nào đó.

MonHoc(MaMH, TenMH, SoTiet)

Tên từ: Mỗi Môn học có một MaMH duy nhất. Mỗi MaMH xác định tất cả các thuộc tính còn lại của môn học đó.

SinhVien(MaSV, HoTen, GioiTinh, NgaySinh, MaLop)

Tên từ: Mỗi sinh viên có mỗi MaSV duy nhất. Mỗi MaSV xác định tất cả các thuộc tính còn lại của sinh viên đó.

KetQua(MaSV, MaMH, LanThi, DiemThi)

YÊU CẦU:

- 1/ Tìm khóa chính cho mỗi lược đồ quan hệ trên.
 - 2/ Hãy chỉ ra các khóa ngoại (nếu có) của mỗi lược đồ.
 - 3/ Hãy thực hiện các câu hỏi sau bằng ngôn ngữ đại số quan hệ
- Lập danh sách sinh viên gồm MaSV, HoTen
 - Lập danh sách gồm MaSV, HoTen của các sinh viên nữ khoa 'CNTT'
 - Lập bảng điểm cho tất cả sinh viên khoa 'CNTT', bảng điểm gồm các cột MaSV, HoTen, TenMH, LanThi, DiemThi
 - Lập phiếu điểm cho sinh viên có MaSV='99001'
 - Lập danh sách sinh viên gồm MaSV, HoTen, TenLop, TenKhoa
 - Lập bảng điểm môn học có mã môn học là 'CSDL' cho tất cả sinh viên có mã lớp là 'CĐTH2B'

- Lập danh sách sinh viên của lớp có mã lớp là 'CĐTH2B' và có điểm thi môn học lần 1 lớn hơn hay bằng 8.

3.6. Xét CSDL bảo hiểm sau:

person(ss#, name, address): Số bảo hiểm ss# sở hữu bởi người tên name ở địa chỉ address

car(license, year, model): Xe hơi số đăng ký license, sản xuất năm year, nhãn hiệu model

accident(date, driver, damage_amount): tai nạn xảy ra ngày date, do người lái driver, mức hư hại damage_amount

owns(ss#, license): người mang số bảo hiểm ss# sở hữu chiếc xe mang số đăng ký license

log(license, date, driver): ghi sổ chiếc xe mang số đăng ký license, bị tai nạn ngày date do người lái driver

Các thuộc tính được gạch dưới là các khóa chính.

Hãy biểu diễn mỗi câu hỏi sau đây bằng ngôn ngữ đại số quan hệ:

- Thêm khách hàng mới: ss# = "A-12345", name = "David", address = "35 Chevre Road", license = "109283", year = "2002", model = "FORD LASER" vào CSDL.
- Xoá các thông tin liên quan đến xe nhãn hiệu "MAZDA" của "John Smith".
- Thêm thông tin tai nạn cho chiếc xe "TOYOTA" của khách hàng mang số bảo hiểm số "A-84626"

3.7. Xét CSDL nhân viên:

employee(E_name, street, city): Nhân viên có tên E_name, cư trú tại phố street, trong thành phố city

works(E_name, C_name, salary): Nhân viên tên E_name làm việc cho công ty C_name với mức lương salary

company(C_name, city): Công ty tên C_name đóng tại thành phố city

manages(E_name, M_name): Nhân viên E_name dưới sự quản lý của nhân viên M_name.

Hãy biểu diễn mỗi câu hỏi sau đây bằng ngôn ngữ đại số quan hệ:

- Tìm tên của tất cả các nhân viên làm việc cho công ty "First Bank"
- Tìm tên và thành phố cư trú của các nhân viên làm việc cho công ty "First Bank"
- Tìm tên, phố, thành phố cư trú làm việc cho "First Bank" và hưởng mức lương lớn hơn 10000\$
- Tìm tất cả nhân viên trong công ty CSDL sinh sống trong cùng thành phố của công ty nơi họ làm việc
- Xoá tất cả các thông tin liên quan tới công ty "Bad Bank"

CHƯƠNG 4

NGÔN NGỮ TRUY VẤN DỮ LIỆU

4.1. Giới thiệu

Ngôn ngữ hỏi có cấu trúc (SQL - Structured Query Language) và các hệ quản trị CSDL quan hệ là một trong những nền tảng kỹ thuật quan trọng trong công nghiệp máy tính. Cho đến nay, có thể nói rằng SQL đã được xem là ngôn ngữ chuẩn trong cơ sở dữ liệu. Các hệ quản trị CSDL quan hệ thương mại hiện có như Oracle, SQL Server, Informix, DB2,... đều chọn SQL làm ngôn ngữ cho sản phẩm của mình

Vậy thực sự SQL là gì? Tại sao nó lại quan trọng trong các hệ quản trị CSDL? SQL có thể làm được những gì và như thế nào? Nó được sử dụng ra sao trong các hệ quản trị CSDL quan hệ? Nội dung của chương này sẽ cung cấp cho chúng ta cái nhìn tổng quan về SQL và một số vấn đề liên quan.

4.1.1. Lịch sử phát triển

Ngôn ngữ SQL được IBM sử dụng đầu tiên trong hệ quản trị CSDL System R vào giữa những năm 70, hệ ngôn ngữ SQL đầu tiên (SEQUEL2) được IBM công bố vào tháng 11 năm 1976. Năm 1979, tập đoàn ORACLE giới thiệu thương phẩm đầu tiên của SQL, SQL cũng được cài đặt trong các hệ quản trị CSDL như DB2 của IBM và SQL/DS.

Ngày nay, SQL được sử dụng rộng rãi và được xem là ngôn ngữ chuẩn để truy cập CSDL quan hệ.

4.1.2. Chuẩn SQL

Năm 1989, viện tiêu chuẩn quốc gia Hoa kỳ (ANSI) công nhận SQL là ngôn ngữ chuẩn để truy cập CSDL quan hệ trong văn bản ANSI SQL89.

Năm 1989, tổ chức tiêu chuẩn quốc tế (ISO) công nhận SQL ngôn ngữ chuẩn để truy cập CSDL quan hệ trong văn bản ISO 9075-1989.

Tất cả các hệ quản trị CSDL lớn trên thế giới cho phép truy cập bằng SQL và hầu hết theo chuẩn ANSI.

4.1.3. Đặc điểm của SQL

SQL là công cụ sử dụng để tổ chức, quản lý và truy xuất dữ liệu được lưu trữ trong các CSDL. SQL là một hệ thống ngôn ngữ bao gồm tập các câu lệnh sử dụng để tương tác với CSDL quan hệ.

Tên gọi *ngôn ngữ hỏi có cấu trúc* phần nào làm chúng ta liên tưởng đến một công cụ (ngôn ngữ) dùng để truy xuất dữ liệu trong các CSDL. Thực sự mà nói, khả năng của SQL vượt xa so với một công cụ truy xuất dữ liệu, mặc dù đây là mục đích ban đầu khi SQL được xây dựng nên và truy xuất dữ liệu vẫn còn là một trong những chức năng quan trọng của nó. SQL được sử dụng để điều khiển tất cả các chức năng mà một hệ quản trị CSDL cung cấp cho người dùng bao gồm:

- **Định nghĩa dữ liệu:** SQL cung cấp khả năng định nghĩa các CSDL, các cấu trúc lưu trữ và tổ chức dữ liệu cũng như mối quan hệ giữa các thành phần dữ liệu.

- **Truy xuất và thao tác dữ liệu:** Với SQL, người dùng có thể dễ dàng thực hiện các thao tác truy xuất, bổ sung, cập nhật và loại bỏ dữ liệu trong các CSDL.
- **Điều khiển truy cập:** SQL có thể được sử dụng để cấp phát và kiểm soát các thao tác của người sử dụng trên dữ liệu, đảm bảo sự an toàn cho CSDL.
- **Đảm bảo toàn vẹn dữ liệu:** SQL định nghĩa các ràng buộc toàn vẹn trong CSDL nhờ đó đảm bảo tính hợp lệ và chính xác của dữ liệu trước các thao tác cập nhật cũng như các lỗi của hệ thống.

Như vậy, có thể nói rằng SQL là một ngôn ngữ hoàn thiện được sử dụng trong các hệ thống CSDL và là một thành phần không thể thiếu trong các hệ quản trị CSDL. Mặc dù SQL không phải là một ngôn ngữ lập trình như C, C++, Java,... song các câu lệnh mà SQL cung cấp có thể được nhúng vào trong các ngôn ngữ lập trình nhằm xây dựng các ứng dụng tương tác với CSDL.

Khác với các ngôn ngữ lập trình quen thuộc như C, C++, Java,... SQL là ngôn ngữ có tính khai báo. Với SQL, người dùng chỉ cần mô tả các yêu cầu cần phải thực hiện trên CSDL mà không cần phải chỉ ra cách thức thực hiện các yêu cầu như thế nào. Chính vì vậy, SQL là ngôn ngữ dễ tiếp cận và dễ sử dụng.

4.1.4. Vai trò của SQL

Bản thân SQL không phải là một hệ quản trị CSDL, nó không thể tồn tại độc lập. SQL thực sự là một phần của hệ quản trị CSDL, nó xuất hiện trong các hệ quản CSDL với vai trò ngôn ngữ và là công cụ giao tiếp giữa người sử dụng và hệ quản trị CSDL.

Trong hầu hết các hệ quản trị CSDL quan hệ, SQL có những vai trò như sau:

- **SQL là ngôn ngữ hỏi có tính tương tác:** Người sử dụng có thể dễ dàng thông qua các trình tiện ích để gửi các yêu cầu dưới dạng các câu lệnh SQL đến cơ sở dữ liệu và nhận kết quả trả về từ CSDL.
- **SQL là ngôn ngữ lập trình cơ sở dữ liệu:** Các lập trình viên có thể nhúng các câu lệnh SQL vào trong các ngôn ngữ lập trình để xây dựng nên các chương trình ứng dụng giao tiếp với CSDL.
- **SQL là ngôn ngữ quản trị cơ sở dữ liệu:** Thông qua SQL, người quản trị cơ sở dữ liệu có thể quản lý được cơ sở dữ liệu, định nghĩa các cấu trúc lưu trữ dữ liệu, điều khiển truy cập CSDL,...
- **SQL là ngôn ngữ cho các hệ thống khách/chủ (client/server):** Trong các hệ thống CSDL khách/chủ, SQL được sử dụng như là công cụ để giao tiếp giữa các trình ứng dụng phía máy khách với máy chủ CSDL.
- **SQL là ngôn ngữ truy cập dữ liệu trên Internet:** Cho đến nay, hầu hết các máy chủ Web cũng như các máy chủ trên Internet sử dụng SQL với vai trò là ngôn ngữ để tương tác với dữ liệu trong các CSDL.
- **SQL là ngôn ngữ cơ sở dữ liệu phân tán:** Đối với các hệ quản trị CSDL phân tán, mỗi một hệ thống sử dụng SQL để giao tiếp với các hệ thống khác trên mạng, gửi và nhận các yêu cầu truy xuất dữ liệu với nhau.
- **SQL là ngôn ngữ sử dụng cho các cổng giao tiếp cơ sở dữ liệu:** Trong một hệ thống mạng máy tính với nhiều hệ quản trị CSDL khác nhau, SQL

thường được sử dụng như là một chuẩn ngôn ngữ để giao tiếp giữa các hệ quản trị CSDL.

4.1.5. Phân loại các lệnh

Các lệnh của SQL chia làm các nhóm chính:

- Ngôn ngữ định nghĩa dữ liệu (DDL - Data Definition Language): Tạo và thay đổi cấu trúc các đối tượng trong CSDL (CREATE, ALTER...)
- Ngôn ngữ thao tác dữ liệu (DML - Data Manipulation Language): Cho phép tìm kiếm, thêm, sửa, xóa dữ liệu (SELECT, INSERT, UPDATE, DELETE...).
- Ngôn ngữ điều khiển dữ liệu (DCL - Data Control Language): Cho phép gán hoặc hủy các quyền truy cập dữ liệu (GRANT, REVOKE).

Ở các phần sau chúng ta sẽ lần lượt nghiên cứu các ngôn ngữ này.

4.2. Ngôn ngữ định nghĩa dữ liệu

Sau đây là một số lệnh chính của ngôn ngữ định nghĩa dữ liệu.

4.2.1. Tạo cơ sở dữ liệu

Mục đích: Đây là lệnh tạo ra một cơ sở dữ liệu với tên xác định.

Cú pháp: *CREATE DATABASE <Tên_CSDL>*

Ví dụ: Tạo một cơ sở dữ liệu có tên là QLNV (Quản lý nhân viên):

```
Create Database QLNV
```

4.2.2. Tạo bảng

Mục đích: Đây là lệnh tạo ra một bảng trong cơ sở dữ liệu với các thuộc tính xác định. Bảng này sẽ được nhận một tên gọi và một cấu trúc (danh sách tên các thuộc tính và một vài đặc trưng). Với các thuộc tính cần khai báo tên, kiểu dữ liệu, và cho phép rỗng hay không.

Khi mới được tạo, bảng chưa có dữ liệu, chỉ là một cấu trúc logic có thể tiếp nhận các dữ liệu.

Cú pháp:

CREATE TABLE <Tên_bảng> (

Tên_thuộc_tính1 Kiểu_thuộc_tính1 [các ràng buộc],

Tên_thuộc_tính2 Kiểu_thuộc_tính2 [các ràng buộc],

...

Tên_thuộc_tínhn Kiểu_thuộc_tínhn [các ràng buộc]

[, các ràng buộc trên bảng dữ liệu])

Trong đó:

Bảng 4.1. Các thành phần của lệnh tạo bảng

Tên_bảng	Tên của bảng cần tạo. Tên phải tuân theo qui tắc định danh và không được vượt quá 128 ký tự.
Tên_thuộc_tính (từ 1 đến n)	Là tên của cột (trường) cần định nghĩa, tên cột phải tuân theo qui tắc định danh và không được trùng nhau trong mỗi một bảng. Mỗi một bảng phải có ít nhất một cột. Nếu bảng có nhiều cột thì định nghĩa của các cột (tên cột, thuộc tính và các ràng buộc) phải phân cách nhau bởi dấu phẩy.
Kiểu_thuộc_tính (từ 1 đến n)	Kiểu dữ liệu của cột. Đây là thuộc tính bắt buộc phải có đối với mỗi cột. (Xem thêm phần 4.2.5)
Các ràng buộc	<p>Khi định nghĩa bảng, chúng ta phải đưa ra các ràng buộc. Chẳng hạn:</p> <ul style="list-style-type: none"> • Giá trị mặc định của cột: là giá trị được tự động gán cho cột nếu như người sử dụng không nhập dữ liệu cho cột một cách tường minh. Mỗi một cột chỉ có thể có nhiều nhất một giá trị mặc định. • Cột có tính chất IDENTITY hay không? tức là giá trị của cột có được tự động tăng mỗi khi có bản ghi mới được bổ sung hay không. Tính chất này chỉ có thể sử dụng đối với các trường kiểu số. • Cột có chấp nhận giá trị NULL hay không . • Cột có phải là khóa hay không. <p>Các ràng buộc (CONSTRAINT) được sử dụng trên mỗi cột hoặc trên bảng nhằm các mục đích sau:</p> <ul style="list-style-type: none"> - Quy định khuôn dạng hay giá trị dữ liệu được cho phép trên cột (chẳng hạn qui định tuổi của một học sinh phải lớn hơn 6 và nhỏ hơn 20, số điện thoại phải là một chuỗi bao gồm 6 chữ số,...) - Đảm bảo tính toàn vẹn dữ liệu trong một bảng và toàn vẹn tham chiếu giữa các bảng trong cơ sở dữ liệu. Những loại ràng buộc này nhằm đảm bảo tính đúng của dữ liệu như: số chứng minh nhân dân của mỗi một người phải duy nhất, nếu sinh viên học một lớp nào đó thì lớp đó phải tồn tại,... <p>(Xem thêm phần 4.2.6)</p>

Ví dụ: Khai báo dưới đây định nghĩa cột STT có kiểu dữ liệu là int và cột có tính chất IDENTITY:

stt INT IDENTITY

Hay định nghĩa cột NGÀY có kiểu datetime và không cho phép chấp nhận giá trị NULL:

ngay DATETIME NOT NULL

Định nghĩa cột SOLUONG kiểu int và có giá trị mặc định là 0:

soluong INT DEFAULT (0)

Ví dụ: Tạo bảng NHANVIEN, có các thuộc tính:

```
CREATE TABLE NHANVIEN(  
    MaNV      Text(10)    NOT NULL PRIMARY KEY  
    TenNV     Text(30)    NOT NULL,  
    DiaChi    Text(50)    NOT NULL,  
    Tuoi      NUMBER)
```

4.2.3. Sửa đổi cấu trúc bảng

Mục đích: Đây là lệnh sửa đổi cấu trúc của bảng hiện đang tồn tại. Chúng ta có thể thêm một thuộc tính (cột) mới, thay đổi cấu trúc của một thuộc tính (cột) đang có, bổ sung khoá, bổ sung ràng buộc.

- **Thêm một cột:**

Cú pháp:

ALTER TABLE <Tên_bảng>

ADD COLUMN Tên_cột , Kiểu_cột[(size)]

Ví dụ: Thêm cột chi chú vào bảng dữ liệu của nhân viên:

```
ALTER TABLE NHANVIEN  
ADD COLUMN GhiChu Text(50))
```

- **Thay đổi kiểu của một thuộc tính:**

Cú pháp:

ALTER TABLE <Tên_bảng>

ALTER COLUMN Tên_cột , Kiểu_cột_mới[(size)]

Ví dụ: Đổi cột ghi chú của bảng dữ liệu về nhân viên sang kiểu Memo

```
ALTER TABLE NHANVIEN ALTER COLUMN GhiChu Memo
```

- **Xóa một thuộc tính**

Cú pháp:

ALTER TABLE <Tên_bảng>

DROP COLUMN Tên_cột

Ví dụ: Xóa cột ghi chú khỏi bảng dữ liệu về nhân viên

```
ALTER TABLE NHANVIEN  
DROP COLUMN GhiChu
```

4.2.4. Xóa đối tượng

Mục đích: Xóa một đối tượng trong cơ sở dữ liệu.

Cú pháp:

DROP <Object_name>

Ví dụ: Xóa bảng dữ liệu về nhân viên:

DROP TABLE NHANVIEN

4.2.5. Các kiểu thuộc tính

Kiểu thuộc tính là các kiểu dữ liệu của các thuộc tính. Các kiểu dữ liệu được sử dụng tùy thuộc vào từng Hệ quản trị CSDL. Chẳng hạn trong bảng 4.2 là các kiểu dữ liệu được sử dụng trong MS Access.

Bảng 4.2. Các loại dữ liệu của Access

Kiểu dữ liệu	Miêu tả	Kích cỡ
Text	Sử dụng ký tự hoặc kết hợp giữa ký tự và số. Chẳng hạn như địa chỉ, hoặc những số không yêu cầu tính toán, như số điện thoại, mã nước, mã vùng...	Khả năng lưu trữ tối đa (FieldSize) là 255 ký tự.
Memo	Sử dụng khi bạn cần lưu trữ một lượng thông tin lớn, ví dụ như trường thông tin ghi chú về một cán bộ.	Khả năng lưu trữ tối đa là 65.536 ký tự.
Number	Number: Sử dụng cho những dữ liệu cần tính toán (loại trừ tính tiền, sử dụng Currency Type).	Khả năng lưu trữ có thể là 1, 2, 4, 8 tùy thuộc vào kiểu dữ liệu ta chọn (byte, integer, long integer, single, double, decimal), riêng đối với kiểu dữ liệu ReplicationID (GUI) thì khả năng lưu trữ là 16 byte.
Currency	Sử dụng Currency cho các dữ liệu cần tính toán. Phần thập phân có thể có từ 1 đến 4 số.	Khả năng lưu trữ là 8 byte.
AutoNumber	Đây là kiểu số tự động tăng với bước tăng là 1. Ta không thể cập nhật lại được trường này.	Sử dụng 4 byte để lưu trữ. Nếu chọn kiểu dữ liệu là thì khả năng lưu trữ có thể lên tới 16 byte
Yes/No	Kiểu dữ liệu YES/NO chỉ chứa một trong 2 giá trị (Yes/No, True/False, On/ Off)	Sử dụng 1 bite để lưu trữ.
OLE Object	Đối tượng (như là một văn bản trong Microsoft Word, dữ liệu đồ họa, âm thanh, hoặc một kiểu dữ liệu nhị	Sử dụng 1 GB để lưu trữ (tùy thuộc vào dung lượng của đĩa).

Ngoài ra còn 2 loại dữ liệu khác như Hyperlink, Lookup Wizard.

Đối với kiểu dữ liệu Number, ta còn có thể lựa chọn chi tiết như trong bảng 4.3.

Bảng 4.3. Các kiểu dữ liệu số trong Access

Kiểu dữ liệu	Miêu tả	Độ chính xác thập phân	Kích cỡ
Integer	Lưu trữ các số từ -32,768 đến 32,767 (không có phân số).	Không	2 byte
Long Integer	Lưu trữ các số từ -2,147,483,648 đến 2,147,483,647 (không có phân số)	Không	4 byte
Single	Lưu trữ các số từ -3.4×10^{38} đến 3.4×10^{38}	7	4 byte
Double	Lưu trữ số từ -1.797×10^{308} đến 1.797×10^{308}	15	8 byte

4.2.6. Các loại ràng buộc trong bảng dữ liệu

Các dạng ràng buộc (constraint) gồm:

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY (Referential)
- CHECK

4.2.6.1. NOT NULL - Không rỗng

Khi có mệnh đề NOT NULL trong định nghĩa của một cột thì bắt buộc thuộc tính này phải có giá trị. Nếu không chỉ thị gì trong định nghĩa của thuộc tính thì có thể có hoặc không có giá trị.

Ví dụ:

```
CREATE TABLE NHANVIEN(
    MaNV NUMBER(10) NOT NULL,
    TenNV CHAR(30))
```

Trong ví dụ trên, MaNV là thuộc tính bắt buộc phải có giá trị, không được để trống. TenNV là thuộc tính có thể để trống.

4.2.6.2. UNIQUE - Duy nhất

Chúng ta có thể tạo Unique Constraint để đảm bảo giá trị của một cột nào đó không bị trùng lặp. Tuy Unique Constraint và Primary Key Constraint (xem phần 4.2.4.3) đều đảm bảo tính duy nhất nhưng chúng ta nên dùng Unique Constraint trong những trường hợp sau:

- Nếu một cột (hay một sự kết hợp giữa nhiều cột) không phải là primary key. Nên nhớ chỉ có một Primary Key Constraint trong một table trong khi ta có thể có nhiều Unique Constraint trên một table.

- Nếu một cột cho phép chứa Null. Unique constraint có thể áp đặt lên một cột chứa giá trị Null trong khi primary key constraint thì không.

Ví dụ:

```
CREATE TABLE NHANVIEN(  
    MaNV NUMBER(10) NOT NULL,  
    TenNV CHAR(30),  
    DiachiNV CHAR(50)  
    CONSTRAINT UNQ_Ten_Diachi UNIQUE(Ten,Diachi))
```

4.2.6.3. PRIMARY KEY - Khoá chính

Chỉ ra ràng buộc duy nhất (giống UNIQUE), tuy nhiên khoá là dạng khoá UNIQUE cấp cao nhất. Một table chỉ có thể có một PRIMARY KEY. Các giá trị trong PRIMARY KEY phải NOT NULL.

Cú pháp:

```
[CONSTRAINT constraint_name ]  
PRIMARY KEY [CLUSTERED|NONCLUSTERED]  
[( colname [,colname2 [...colname16]])]
```

Ví dụ:

```
CREATE TABLE NHANVIEN (  
    MaNV char(10) NOT NULL primary key,  
    TenNV char(30), DiachiNV char(50))  
Hoặc ta có thể viết câu lệnh sau:  
CREATE TABLE NHANVIEN (  
    MaNV char(10) NOT NULL,  
    TenNV char(30),  
    DiachiNV char(50),  
    CONSTRAINT NV_P_K PRIMARY KEY (MaNV))
```

4.2.6.4. FOREIGN KEY - Khoá ngoại

Chỉ ra mối liên hệ ràng buộc tham chiếu giữa bảng này với bảng khác.

Từ khoá ON DELETE CASCADE được chỉ định trong dạng khoá này để chỉ khi dữ liệu cha bị xoá thì dữ liệu con cũng tự động bị xoá theo.

Cú pháp:

```
[CONSTRAINT constraint_name ]  
[FOREIGN KEY (colname [,colname2 [...colname16]])] REFERENCES  
reference_table [(ref_colname[,ref_colname2[...ref_colname 16]])]
```

Ví dụ:

Hai bảng DONVI và bảng NHANVIEN có mối quan hệ cha – con (1-n). Thuộc tính MaDV trong bảng NHANVIEN (bảng con) là khoá ngoại, được tham chiếu từ thuộc tính MaDV của bảng DONVI (bảng cha). Ta tạo 2 bảng như sau:

```
CREATE TABLE DONVI (  
    MaDV char(2) primary key,  
    TenDV char(20) not null)  
  
CREATE TABLE NHANVIEN (  
    MaNV char(10) primary key,  
    TenNV char(30) not null,  
    Diachi char(50),  
    madv char(2)  
  
    CONSTRAINT k_n_madv FOREIGN KEY(madv) REFERENCES  
    DONVI(MaDV))
```

4.2.6.5. CHECK - Ràng buộc kiểm tra giá trị

Ràng buộc CHECK được sử dụng để yêu cầu các giá trị trong cột, hoặc khuôn dạng dữ liệu trong cột phải theo một quy tắc nào đó. Trên một cột có thể có nhiều ràng buộc này.

Cú pháp:

[CONSTRAINT constraint_name] CHECK (expression)

Trong đó, expression là một biểu thức logic. Sau khi có ràng buộc này, giá trị nhập vào cho cột phải thỏa mãn điều kiện mới được chấp nhận.

Ví dụ:

```
CREATE TABLE NHANVIEN  
(MaNV CHAR(10) NOT NULL PRIMARY KEY,  
TenNV CHAR(30),  
Luong NUMBER(10,2)  
CONSTRAINT CK_Luong CHECK(Luong>500))
```

4.2.6.6. DEFAULT - Mặc định

Ràng buộc DEFAULT được sử dụng để quy định giá trị mặc định cho một cột. Giá trị này sẽ tự động gán cho cột nếu người sử dụng không nhập vào khi bổ sung bản ghi.

Cú pháp:

[CONSTRAINT constraint_name]

DEFAULT {const_expression/nonarguments_function/NULL}

Ví dụ:

```
CREATE TABLE NHANVIEN (  
    MaNV char(10) primary key,  
    TenNV char(30) not null,  
    Gioitinh char(3) DEFAULT 'Nam')
```

4.3. Ngôn ngữ thao tác dữ liệu**4.3.1. Thêm bản ghi (INSERT)**

Mục đích: Lệnh này được dùng để thêm một hoặc nhiều dòng (bản ghi) mới vào một bảng.

Dạng đơn giản nhất của lệnh này là thêm mỗi lần 1 dòng. Nó đòi hỏi phải nêu tên của bảng, tên các thuộc tính và giá trị cần gán cho chúng. Nếu không nêu tên các thuộc tính thì điều đó có nghĩa là tất cả các thuộc tính trong bảng đều cần được thêm giá trị theo thứ tự từ trái sang phải.

Cú pháp:

```
INSERT [INTO]<TableName> (Column1, Column2, ..., Columnn)  
VALUES (Values1, Values2, ..., Valuesn)
```

Ví dụ:

Giả sử ta đã có cấu trúc bảng NHANVIEN(MaNV, TenNV, Diachi, Tuoi)

- Thêm bản ghi mới có tất cả các trường cho bảng NHANVIEN. Vì tất cả các thuộc tính trong bảng đều được thêm giá trị nên ta không cần có danh sách các thuộc tính ngay sau tên bảng NHANVIEN:

```
INSERT INTO NHANVIEN  
VALUES('NV01','Nguyễn Công Thành', '01 LDD',22 )
```

- Thêm bản ghi mới vào bảng, với tuổi không xác định:

```
INSERT INTO NHANVIEN(MaNV,TenNV,DiaChi)  
VALUES('NV02','Nguyễn Phương Lan','23 HTK')
```

Ngoài ra chúng ta còn có thêm dữ liệu cho bảng từ giá trị của bảng khác như sau:

Cú pháp:

```
INSERT [INTO]<TableName> (Column1, Column2, ..., Columnn)  
SELECT Select_list FROM <Tables>
```

Ví dụ:

```
INSERT INTO NHANVIEN_tam (TenNV, Tuoi)  
SELECT TenNV, Tuoi FROM NHANVIEN WHERE Tuoi > 20
```

4.3.2. Xóa bản ghi (DELETE)

Mục đích: Lệnh này được dùng để xóa một hoặc nhiều dòng (bản ghi) trong một bảng.

Lệnh này gồm 1 mệnh đề DELETE FROM để chỉ ra tên gọi của bảng được xét, và một mệnh đề WHERE để chỉ ra các dòng cần phải xóa. Như vậy, ta có thể cùng lúc xóa được nhiều dòng nếu dòng đó thỏa mãn điều kiện. Muốn xóa tất cả các dòng của một bảng thì không cần đưa vào mệnh đề WHERE.

Cú pháp:

DELETE FROM <Table_name> WHERE <Conditions>

Ví dụ:

- Xóa một bản ghi (dòng) có MaNV='NV01' trong bảng NHANVIEN:

DELETE FROM NHANVIEN WHERE MaNV='NV01'

- Xóa những nhân viên có địa chỉ là 41NC có trong bảng NHANVIEN:

DELETE FROM NHANVIEN WHERE Diachi='41NC'

4.3.3. Sửa bản ghi (UPDATE)

Mục đích: Lệnh này dùng để sửa đổi dữ liệu trong bảng.

Cú pháp:

UPDATE <Table_name>

SET (Column_name= <new value>) WHERE <Condition>

Ví dụ:

UPDATE NHANVIEN

SET (Diachi= '24 LTT') WHERE MaNV= 'NV01'

4.3.4. Truy vấn dữ liệu (SELECT)

Mục đích: Lệnh này dùng để tìm kiếm các dữ liệu thỏa mãn điều kiện nào đó.

Cú pháp:

SELECT [DISTINCT]/Columns_list/Expression_list/*

FROM <Tables_list>

WHERE <Conditions>

GROUP BY <Columns>

HAVING <Conditions_for_group>

ORDER BY [ACS/ DESC]

Trong đó:

- Sau SELECT: Các thông tin cần đưa ra, đó chính là danh sách các thuộc tính
- Sau FROM: Danh sách các tên bảng, từ đó thông tin được lấy ra.
- Sau WHERE: Các biểu thức logic, chỉ ra thông tin được lấy ra từ hàm nào hoặc điều kiện nối giữa các bảng.
- Sau GROUP BY: Các cột mà trong đó được tính theo từng nhóm.
- Sau HAVING: Biểu thức logic chỉ ra thông tin được lấy ra từ nhóm nào.
- Sau ORDER BY: Chỉ ra các cột mà trong đó thông tin được sắp xếp theo

thứ tự:

- + ASC: thông tin được sắp xếp theo chiều tăng dần (ASCendent)
- + DESC: thông tin được sắp xếp theo chiều giảm dần (DESCendent)

Bảng 4.4. Bảng dữ liệu Nhân viên

NHANVIEN				
MaNV	HoTen	CongViec	Luong	MaDV
NV001	Phạm Thị Nhân	Thư ký	500	0001
NV002	Hoàng Thanh Vân	Giáo viên	600	0001
NV003	Hoàng Thị Lan	Giáo viên	200	0002
NV004	Đỗ Trung Dũng	Thư ký	700	0003

Ví dụ:

Sau đây là một số ví dụ đơn giản về truy vấn dữ liệu trong bảng 4.4:

- Hiện thị tất cả các thông tin về nhân viên:

```
SELECT * FROM NHANVIEN
```

- Hiện thị Hoten, Luong của các nhân viên:

```
SELECT Hoten, Luong FROM NHANVIEN
```

- Hiện thị Hoten, Luongnam (Luong *12) của tất cả các nhân viên:

```
SELECT Hoten, Luong*12 FROM NHANVIEN
```

4.3.4.1. Tránh các giá trị trùng lặp (DISTINCT)

Chúng ta có thể sử dụng DISTINCT để tránh các giá trị trùng lặp.

Ví dụ:

Hiện thị các công việc khác nhau trong bảng NHANVIEN (bảng 4.4):

```
SELECT DISTINCT Congviec FROM NHANVIEN
```

Sau đây là kết quả trong trường hợp không có lệnh DISTINCT và có DISTINCT:

Congviec	CongViec
Thư ký	Giáo viên
Giáo viên	Thư ký
Giáo viên	
Thư ký	

4.3.4.2. Sử dụng bí danh cột

Chúng ta có thể sử dụng một tên khác thay thế cho tên cột.

Ví dụ:

```
SELECT Hoten, Luong*12 AS Luongnam FROM NHANVIEN
```

Kết quả:

Hoten	LuongNam
Phạm Thị Nhân	6000
Hoàng Thanh Vân	7200
Hoàng Thị Lan	2400
Đỗ Trung Dũng	8400

4.3.4.3. Sắp xếp thứ tự (ORDER BY)

Để sắp xếp dữ liệu khi truy vấn, chúng ta sử dụng ORDER BY. Có thể sử dụng ASC(ascendent) để sắp xếp tăng dần, hoặc DESC(descendent) để sắp xếp giảm dần. Nếu để giá trị mặc định thì sẽ sắp xếp theo chiều tăng dần.

Ví dụ:

Hiển thị các thông tin về Hoten, Luong. Dữ liệu được sắp xếp theo thứ tự tăng dần/giảm dần của Luong:

```
SELECT Hoten, Luong
FROM NHANVIEN
ORDER BY Luong [ASC/ DESC]
```

4.3.4.4. Hiển thị dữ liệu theo điều kiện (WHERE)

Cú pháp:

```
SELECT [DISTINCT]|Columns_list|Expression_list|*
FROM <Tables_list>
WHERE <Conditions>
```

Điều kiện sau mệnh đề Where là một biểu thức, sử dụng các phép toán so sánh và phép toán logic.

➤ Các phép so sánh:

- = : Toán tử bằng hay tương đương
- != : Toán tử khác hay không tương đương
- > : Toán tử lớn hơn
- < : Toán tử nhỏ hơn
- >= : Toán tử lớn hơn hoặc bằng
- <= : Toán tử nhỏ hơn hoặc bằng

Ví dụ:

Hiển thị Hoten, Luong của các nhân viên có Luong>300

```
SELECT HoTen, Luong FROM NHANVIEN WHERE (Luong>300)
```

➤ Sử dụng các phép logic:

- AND** : Trả về kết quả đúng nếu tất cả các điều kiện đều đúng
- OR** : Trả về kết quả đúng nếu một trong các điều kiện đúng

NOT : Trả về kết quả phủ định của điều kiện

Ví dụ:

Hiển thị Hoten, Luong của những nhân viên có công việc là Giáo viên và mức lương >300:

```
SELECT HoTen, Luong
FROM NHANVIEN
WHERE (Luong>300) AND (Congviec='Giáo viên')
```

➤ **Sử dụng các toán tử của SQL:**

[NOT] BETWEEN x AND y : [Không] nằm giữa giá trị X và Y

IN (danh sách) : Thuộc bất kỳ giá trị nào trong danh sách

x [NOT] LIKE y : Đúng nếu x [không] giống khung mẫu y

Các ký tự dùng trong khuôn mẫu:

- Dấu gạch dưới (_) : Chỉ một ký tự bất kỳ
- Dấu phần trăm (% hoặc *) : Chỉ một nhóm ký tự bất kỳ

IS [NOT] NULL : Kiểm tra giá trị rỗng

EXISTS : Trả về TRUE nếu có tồn tại.

Ví dụ:

Hiển thị những nhân viên có Lương trong khoảng 300 đến 600:

```
SELECT HoTen, Luong
FROM NHANVIEN
WHERE Luong BETWEEN 300 AND 600
```

Ví dụ:

Hiển thị những nhân viên có lương hoặc 200, 300, 600:

```
SELECT HoTen, Luong
FROM NHANVIEN
WHERE Luong IN (200,500,600)
```

Ví dụ:

Hiển thị Hoten, Congviec của các nhân viên có Họ tên bắt đầu bằng chữ 'Hoàng':

```
SELECT HoTen, Congviec
FROM NHANVIEN
WHERE Hoten LIKE 'Hoàng*'
```

Ví dụ:

Hiển thị các nhân viên mà địa chỉ không rỗng:

```
SELECT * FROM NHANVIEN WHERE Diachi IS NULL
```

Ví dụ: Các câu hỏi lồng nhau:

Giả sử chúng ta có bảng dữ liệu về nhân viên, dự án, và thời gian các nhân viên tham gia dự án như sau:

NHANVIEN(MaNV, TenNV, Diachi, tuoi)

DUAN(MaDA, TenDA)

TGDUAN(MaNV, MaDA, BD, KT)

Tìm tên các nhân viên tham gia dự án có mã dự án là '01'

SELECT TenNV

FROM NHANVIEN

WHERE MaNV IN (SELECT MaDA

FROM TGDUAN

WHERE MaDA='01')

4.3.4.4. Nhóm các thông tin (GROUP BY)

Mệnh đề GROUP BY <các cột> cho phép đưa ra thông tin theo từng nhóm.

Ví dụ:

Hiển thị Công việc, Lương trung bình của từng loại công việc:

SELECT CongViec, AVG(Luong) AS LuongTB

FROM NHANVIEN

GROUP BY CongViec

Trong đó AVG là hàm tính trung bình (Xem phần 4.5)

Kết quả:

CongViec	LuongTB
Giáo viên	400
Thư ký	600

Có thể thêm vào một mệnh đề WHERE để đưa vào một tiêu chuẩn chọn lựa các dòng. SQL thực hiện cùng một cách xử lý, đầu tiên là loại bỏ các dòng không đáp ứng tiêu chuẩn đã được xác định trong mệnh đề WHERE.

Ví dụ:

SELECT CongViec, AVG(Luong) AS LuongTB

FROM NHANVIEN

WHERE Luong>200

GROUP BY CongViec

Kết quả:

CongViec	LuongTB
Giáo viên	600
Thư ký	600

Có thể Sử dụng mệnh đề GROUP BY để đưa ra các thông tin về các nhóm con trong các nhóm lớn.

Ví dụ:

Hiển thị tổng lương của từng nhóm công việc trong từng đơn vị:

```
SELECT MaDV, CongViec, SUM(Luong) AS TongLuong
FROM NHANVIEN
GROUP BY MaDV, CongViec
```

Kết quả:

MaDV	CongViec	TongLuong
0001	Giáo viên	600
0001	Thư ký	500
0002	Giáo viên	200
0003	Thư ký	700

4.3.4.5. Mệnh đề HAVING

Khi muốn hiển thị các nhóm trên cơ sở thông tin nhóm thì điều kiện phải được viết trong mệnh đề HAVING (Không viết trong mệnh đề WHERE).

Sau đây là một số ví dụ về HAVING. Tham khảo thêm các hàm ở phần 4.5.

Ví dụ:

Hiển thị những công việc và lương trung bình của các công việc có lương trung bình ≥ 300 :

```
SELECT CongViec, Avg(Luong) AS TBLuong
FROM NHANVIEN
GROUP BY CongViec
HAVING (Avg(Luong)>300)
```

Kết quả:

CongViec	TBLuong
Giáo viên	400
Thư ký	600

Ví dụ:

Hiển thị những đơn vị và lương lớn nhất của các đơn vị có lương lớn nhất ≥ 300 :

```
SELECT MaDV, Max(Luong) AS MaxLuong
FROM NHANVIEN
GROUP BY MaDV
HAVING Max(Luong)>300
```

Kết quả:

MaDV	MaxLuong
0001	600
0003	700

Ghi chú: Mệnh đề HAVING là mệnh đề tương đương với WHERE áp dụng cho các nhóm. Nói chung, mệnh đề này chỉ sử dụng nếu đã có chỉ thị một mệnh đề GROUP BY.

4.4. Ngôn ngữ điều khiển dữ liệu (DCL)

Ngôn ngữ điều khiển dữ liệu được sử dụng trong việc cấp phát hay huỷ bỏ quyền của người sử dụng.

4.4.1. Lệnh GRANT

Mục đích: Câu lệnh này dùng để cấp phát quyền cho người sử dụng trên đối tượng CSDL hoặc quyền thực thi các câu lệnh SQL.

Cú pháp có 2 dạng như sau:

Dạng 1: Cấp quyền đối với câu lệnh SQL:

GRANT ALL / statement [...,statementN] TO account [...,accountN]

Dạng 2: Cấp quyền đối với các đối tượng trong CSDL:

GRANT ALL / permission [...,permissionN]

ON table_name / view_name [(column1 [...,columnN])]

/ ON stored_procedure

TO account [..., accountN]

Trong đó:

- **ALL:** là từ khoá được sử dụng khi muốn cấp phát tất cả các quyền cho người sử dụng.

- **Account:** là tên tài khoản đăng nhận hệ thống

- **Permission:** là quyền cấp phát cho người sử dụng trên đối tượng cơ sở dữ liệu:

- + Các quyền có thể cấp phát trên một bảng hoặc một View: Select, Insert, Delete, Update.

- + Các quyền có thể cấp phát trên cột của bảng hoặc của View: Select, Update

- + Quyền có thể cấp phát với các thủ tục: EXECUTE (thực thi)

- **Statement:** Là câu lệnh được cấp phát cho người sử dụng. Các câu lệnh có thể cấp phát là:

- + CREATE DATABASE

- + CREATE TABLE

- + CREATE VIEW

- + CREATE PROCEDURE
- + CREATE RULE
- + CREATE DEFAULT
- + BACKUP DATABASE
- + BACKUP LOG

Ví dụ: Câu lệnh sau sẽ 3 cấp quyền SELECT, UPDATE, INSERT trên các thuộc tính (TenNV, DiaChi, Tuoi) của bảng NHANVIEN cho 2 người sử dụng phnhung, htvan:

```
GRANT SELECT, UPDATE, INSERT
ON NHANVIEN(TenNV, DiaChi, Tuoi) TO phnhung, htvan
```

Ví dụ: Câu lệnh sau sẽ cấp quyền tạo bảng, tạo View và tạo thủ tục cho người dùng phnhung:

```
GRANT CREATE TABLE, CREATE VIEW, CREATE PROCEDURE TO
phnhung
```

4.4.2. Lệnh DENY

Mục đích: Trái ngược với lệnh Grant, lệnh Deny từ chối quyền thực hiện các lệnh trên một bảng nào đó của CSDL đối với người sử dụng.

Cú pháp:

```
DENY ALL / statement [... , statementN]
ON Table_name
TO account [... , accountN]
```

Ví dụ:

Từ chối quyền Select trên bảng NHANVIEN với các user trong nhóm Public:
DENY SELECT ON NHANVIEN TO Public

4.4.3. Lệnh REVOKE

Mục đích: Lệnh REVOKE được dùng để huỷ bỏ quyền đã được cấp phát cho người sử dụng. Câu lệnh này cũng có 2 dạng tương tự như câu lệnh GRANT.

Dạng 1: Hủy quyền thực hiện câu lệnh:

```
REVOKE ALL / statement [... , statementN] FROM account [... , accountN]
```

Dạng 2: Hủy quyền thực hiện các đối tượng:

```
REVOKE ALL / permission [... , permissionN]}
ON table_name / view_name [(column [... , columnN])] / stored_procedure
FROM account [... , accountN]
```

Ví dụ:

Hủy quyền INSERT trên các thuộc tính (TenNV, DiaChi, Tuoi) của bảng NHANVIEN đối với htvan:

```
REVOKE SELECT, UPDATE, INSERT
```

ON NHANVIEN(TenNV, DiaChi, Tuoi) FROM htvan

4.5. Các hàm thư viện của SQL

Trong ngôn ngữ SQL có một danh sách các hàm thư viện, chúng ta có thể sử dụng trong câu lệnh SQL đã trình bày ở phần trên.

4.5.1. Hàm số học

Đầu vào và đầu ra của các hàm số học là các giá trị kiểu số.

- ROUND(n[,m]): Cho giá trị làm tròn của n (đến cấp m, mặc nhiên m=0).
- TRUNC(n[,m]): Cho giá trị n lấy m chữ số tính từ chấm thập phân.
- CEIL(n): Cho số nguyên nhỏ nhất lớn hơn hoặc bằng n.
- FLOOR(n): Cho số nguyên lớn nhất bằng hoặc nhỏ hơn n.
- POWER(m,n): Cho lũy thừa bậc n của m.
- EXP(n): Cho giá trị của e^n .
- SQRT(n): Cho căn bậc 2 của n, $n \geq 0$.
- SIGN(n): Cho dấu của n
 - $n < 0$ có SIGN(n) = -1
 - $n = 0$ có SIGN(n) = 0
 - $n > 0$ có SIGN(n) = 1
- ABS(n): Cho giá trị tuyệt đối.
- MOD(m,n): Cho phần dư của phép chia m cho n.
- LOG(m,n): Cho logarit cơ số m của n.
- SIN(n): Cho cosin của n (n tính bằng radian).
- COS(n): Cho cosin của n (n tính bằng radian).
- TAN(n): Cho cotang của n (n tính bằng radian).

4.5.2. Các hàm ký tự

- CONCAT(char1, char2): Cho kết hợp của 2 chuỗi ký tự, tương tự như sử dụng toán tử.
- INITCAP(char): Cho chuỗi với ký tự đầu các từ là ký tự hoa.
- LOWER(char): Cho chuỗi ký tự viết thường (không viết hoa).
- LPAD(char1, n [,char2]): Cho chuỗi ký tự có chiều dài bằng n. Nếu chuỗi char1 ngắn hơn n thì thêm vào bên trái chuỗi char2 cho đủ n ký tự. Nếu chuỗi char1 dài hơn n thì giữ lại n ký tự tính từ trái sang.
- LTRIM(char1, n [,char2]): Bỏ các ký tự trống bên trái.
- NLS_INITCAP(char): Cho chuỗi với ký tự đầu các từ là chữ hoa, các chữ còn lại là chữ thường.
- REPLACE(char,search_string[,replacement_string]): Thay tất cả các chuỗi search_string có trong chuỗi char bằng chuỗi replacement_string.

- RPAD(char1, n [,char2]): Giống LPAD(char1, n [,char2]) nhưng căn phải.
- RTRIM(char1, n [,char2]): Bỏ các ký tự trống bên phải.
- SOUNDEX(char): Cho chuỗi đồng âm của char.
- SUBSTR(char, m [,n]): Cho chuỗi con của chuỗi char lấy từ vị trí m về phải n ký tự, nếu không chỉ n thì lấy cho đến cuối chuỗi.
- TRANSLATE(char, from, to): Cho chuỗi trong đó mỗi ký tự trong chuỗi from thay bằng ký tự tương ứng trong chuỗi to, những ký tự trong chuỗi from không có tương ứng trong chuỗi to sẽ bị loại bỏ.
- UPPER(char): Cho chuỗi chữ hoa của chuỗi char.
- ASCII(char): Cho ký tự ASCII của byte đầu tiên của chuỗi char.
- INSTR(char1, char2 [,n[,m]]): Tìm vị trí chuỗi char2 trong chuỗi char1 bắt đầu từ vị trí n, lần xuất hiện thứ m.
- LENGTH(char): Cho chiều dài của chuỗi char.

4.5.3. Các hàm ngày

- MONTH_BETWEEN(d1, d2): Cho biết số tháng giữa ngày d1 và d2.
- ADD_MONTHS(d,n): Cho ngày d thêm n tháng.
- NEXT_DAY(d, char): Cho ngày tiếp theo ngày d có thứ chỉ bởi char.
- LAST_DAY(d): Cho ngày cuối cùng trong tháng chỉ bởi d.
- ROUND(date1): Trả về ngày date1 tại thời điểm giữa trưa 12:00 AM.
- ROUND(date1, 'MONTH'): Nếu date 1 nằm trong nửa tháng đầu trả về ngày đầu tiên của tháng, ngược lại sẽ trả về ngày đầu tiên của tháng sau.
- ROUND(date1, 'YEAR'): Nếu date 1 nằm trong nửa năm đầu trả về ngày đầu tiên của tháng, ngược lại sẽ trả về ngày đầu tiên của năm sau.
- TRUNC(date1, 'MONTH'): Trả về ngày đầu tiên của tháng chứa date1.
- TRUNC(date1, 'YEAR'): Trả về ngày đầu tiên của năm chứa date1.

4.5.4. Các hàm chuyển đổi kiểu

- TO_CHAR(number|date, 'fmt'): Chuyển kiểu số và ngày về kiểu ký tự.
- TO_NUMBER(char): Chuyển ký tự có nội dung số sang số.
- TO_DATE('chsr', 'fmt'): Chuyển ký tự sang kiểu ngày với định dạng đặt trong fmt.
- DECODE(EXPR, SEARCH1, RESULT1, SEARCH2, RESULT2, DEFAULT): So sánh biểu thức expr với giá trị search nếu đúng trả về giá trị result nếu không trả về giá trị default.
- NVL(COL|VALUE, VAL): Chuyển giá trị COL|VALUE thành val nếu null.
- GREATEST(col|value1, col|value2): Trả giá trị lớn nhất trong dãy giá trị.

4.5.4. Hàm nhóm

- COUNT(): Đếm số lượng các bộ hiện có trong một quan hệ.

- SUM(colume): Tính tổng các giá trị của thuộc tính (thuộc loại số học).
- AVG(colume): Tính giá trị trung bình các giá trị của thuộc tính (thuộc loại số học).
- MAX(colume): Tìm giá trị cực đại của thuộc tính.
- MIN(colume): Tìm giá trị cực tiểu của thuộc tính.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 4

4.1. Một cơ sở dữ liệu quản lý học viên có tên là DIEM và gồm 3 bảng như sau:

- Bảng dữ liệu về sinh viên gồm mã số sinh viên, tên, tuổi, giới tính như sau:

Student: Student(RN, Name, Age, Gender)

- Bảng dữ liệu về môn học gồm mã môn học, tên môn học như sau:

Bảng Subject (môn học): Subject(sID, sName)

- Bảng dữ liệu về kết quả học của sinh viên, gồm mã sinh viên, mã môn học, điểm, ngày như sau:

Bảng StudentSubject: StudentSubject(RN, SID, Mark, Date)

1. Sử dụng hệ quản trị cơ sở dữ liệu (Access hoặc Microsoft SQL server) để tạo các bảng dữ liệu trên.

a. Đặt khóa chính cho các bảng:

Student khóa là RN

Subject khóa là sID

StudentSubject khóa là (RN, sID)

b. Đặt ràng buộc để trường Mark chỉ nhận các giá trị trong khoảng [0->10]

c. Đặt ràng buộc khóa chính khóa ngoại giữa 2 bảng Student(RN) và StudentSubject(RN).

2. Hãy viết câu lệnh SQL để nhập dữ liệu vào các bảng Student và Subject (gồm các học viên và môn học có tên như ở câu 3).

3. Hãy viết câu lệnh SQL để nhập dữ liệu vào bảng StudentSubject:

a. Học viên Mỹ Linh đạt điểm 8 môn SQL vào ngày 7/28/2005

b. Học viên Đàm Vĩnh Hưng đạt điểm 3 môn LGC vào ngày 7/29/2005

c. Học viên Kim Tử Long đạt điểm 9 môn HTML vào ngày 7/31/2005

d. Học viên Tài Linh đạt điểm 5 môn SQL vào ngày 7/30/2005

e. Học viên Mỹ Lệ đạt điểm 10 môn CF vào ngày 7/19/2005

f. Học viên Ngọc Oanh đạt điểm 9 môn SQL vào ngày 7/25/2005

4. Hãy viết câu lệnh SQL để cập nhật lại giới tính cho các học viên:

a. Mỹ Linh, Tài Linh, Mỹ Lệ là 0

b. Kim Tử Long là 1

c. Ngọc Oanh vẫn để Null

5. Xóa các thông tin về học viên Đàm Vĩnh Hưng khỏi cơ sở dữ liệu.

4.2. Một cơ sở dữ liệu KHACHHANG, gồm bốn bảng dữ liệu sau:

KHACH_HANG (mã khách hàng, ngày sinh, họ và tên, địa chỉ)

DON_HANG (số đơn hàng, mã khách hàng, ngày đặt hàng)

DONG_DON (số đơn hàng, mã hàng, số lượng)

HANG_HOA (mã hàng, tên hàng, mô tả, đơn giá theo đơn vị)

Hãy tạo cơ sở dữ liệu và nhập dữ liệu cho các bảng. Dùng câu lệnh SQL để trả lời các câu truy vấn sau đây:

1. Tạo các bảng trong cơ sở dữ liệu KHACHHANG
2. Nhập vào theo thứ tự mỗi bảng của cơ sở dữ liệu 5 bản ghi
3. Tìm kiếm thông tin về các khách hàng (bao gồm: mã khách hàng, họ và tên, địa chỉ) không đặt hàng trong 01 tháng tính đến ngày hiện tại
4. Tìm kiếm thông tin về 05 mặt hàng (bao gồm: mã hàng, đơn hàng, tên hàng, tổng tiền) có tổng tiền đặt là bé nhất. Biết rằng: tổng tiền = đơn giá theo đơn vị * số lượng
5. tính tổng tiền của tất cả các khách hàng đã đặt hàng có tên là "nokia6610i"
6. liệt kê danh sách khách hàng đã đặt hàng, gồm các thông tin: mã khách hàng, ngày sinh, họ tên, địa chỉ, mã hàng, tên hàng, mô tả, đơn giá theo đơn vị, số đơn hàng, số lượng. Hãy cho biết câu lệnh SQL trên sử dụng phép toán đại số quan hệ nào?

4.3. Cho một lược đồ cơ sở dữ liệu như sau:

EMP(EmpNo, Ename, Job, Mgr, HireDate, Sal, Comm, DeptNo, Grade)

DEPT(DeptNo, Dname, Loc)

SALGRADE(Grade, Losal, Hisal)

Trong đó các bảng quan hệ có ý nghĩa như sau:

- EMP: Lưu các thông tin của nhân viên, bao gồm các thông tin: Mã nhân viên (EmpNo), Tên của nhân viên: Ename, Nghề nghiệp (Job), Mã người quản lý trực tiếp (Mgr), ngày vào làm (HireDate), lương hàng tháng (Sal), phụ cấp hàng tháng (Comm) và mã phòng ban (DeptNo) mà nhân viên này đang làm việc, Grade là mã mức lương của nhân viên.
- DEPT: Lưu các thông tin về phòng ban bao gồm các thông tin: Mã phòng ban (DeptNo), tên phòng ban (Dname), Nơi đặt văn phòng (Loc).
- SALGRADE: Lưu các thông tin về mức lương bao gồm các thông tin: Mã mức lương (Grade), lương thấp nhất (Losal), lương cao nhất (Hisal).

Hãy tạo cơ sở dữ liệu và nhập dữ liệu cho các bảng. Dùng câu lệnh SQL để trả lời các câu truy vấn sau đây:

1. Nêu tên phòng mà nhân viên "SMITH" làm việc
2. Liệt kê những nhân viên có tên bắt đầu bằng chữ "A"
3. Liệt kê tất cả các phòng ban đóng tại thành phố Hồ Chí Minh
4. Liệt kê mã số nhân viên, tên nhân viên, tên phòng, lương và bậc lương của mọi nhân viên
5. Liệt kê những nhân viên nào có thâm niên làm việc lâu nhất
6. Liệt kê những nhân viên không có phụ cấp hàng tháng. Biết rằng những nhân viên có giá trị của field Comm bằng 0 hay Null đều được xem là không có phụ cấp

7. Liệt kê mã số nhân viên, tên nhân viên, tên phòng ban và thu nhập hàng năm của anh ta. Biết rằng thu nhập hàng năm = (Salary + Comm) * 12 và nếu Comm là Null thì được xem là 0
8. Liệt kê những nhân viên có chức vụ cao nhất trong công ty. Biết rằng nhân viên có chức vụ cao nhất là nhân viên không chịu sự quản lý của bất kỳ một nhân viên nào khác
9. Liệt kê những nhân viên có chức vụ thấp nhất trong công ty: Nhân viên không làm quản lý cho nhân viên nào cả
10. Liệt kê tên của nhân viên và tên của người quản lý trực tiếp của họ
11. Cho biết phòng tài chính ("FINANCE") có bao nhiêu nhân viên
12. Liệt kê tên của các phòng ban, kèm theo tổng số nhân viên trong phòng ban đó
13. Liệt kê tên phòng ban, số lượng nhân viên của các phòng ban có nhiều nhân viên nhất
14. Cho biết tổng số tiền lương phải trả hàng tháng cho các nhân viên theo từng phòng ban
15. Liệt kê những phòng ban nào có tổng số tiền lương phải trả hàng tháng nhân viên hàng tháng là lớn nhất
16. Tính thu nhập bình quân của các nhân viên
17. Liệt kê tất cả các nhân viên có lương cao hơn mức lương trung bình của tất cả mọi nhân viên
18. Liệt kê những nhân viên có lương cao hơn lương của người quản lý mình
19. Liệt kê những nhân viên có lương thấp nhất
20. Liệt kê những nhân viên có lương thấp nhất trong mỗi phòng ban
21. Liệt kê 3 nhân viên có lương lớn nhất

4.4. Cho lược đồ CSDL dùng để quản lý lao động bao gồm các lược đồ quan hệ sau:

Nhanvien(MaNV, HoTen, NgaySinh, Phai, DiaChi, MaPB)

Tên từ: Mỗi nhân viên có một mã số nhân viên (MANV) duy nhất. Một mã số nhân viên xác định các thông tin như họ tên (HoTen), ngày sinh (NgaySinh), phái (Phai), địa chỉ (DiaChi) và phòng ban (MaPB) nơi quản lý nhân viên.

Phongban(MaPB, TenPB)

Tên từ: Mỗi phòng ban có một mã phòng ban (MaPB) duy nhất, mã phòng ban xác định tên phòng ban (TenPB).

Cong(MaCT, MaNV, SLNgayCong)

Tên từ: Lược đồ quan hệ Cong ghi nhận số lượng ngày công (SLNgayCong) của một nhân viên (MaNV) tham gia vào công trình (MaCT).

Congtrinh(MaCT, TenCT, DiaDiem, NgayCapGP, NgayKC, NgayHT)

Tên từ: Mỗi công trình có một mã số công trình (MaCT) duy nhất. Mã số công trình xác định các thông tin như tên công trình (TenCT), địa điểm (DiaDiem), ngày công

trình được cấp giấy phép xây dựng (NgayCapGP), ngày khởi công (NgayKC), ngày hoàn thành (NgayHT).

Hãy tạo cơ sở dữ liệu và nhập dữ liệu cho các bảng. Dùng câu lệnh SQL để trả lời các câu truy vấn sau đây:

1. Lập danh sách những nhân viên có tham gia vào công trình có mã số công trình là X. Yêu cầu các thông tin: MaNV, HoTen, SLNgayCong
2. Lập danh sách những nhân viên có sinh nhật trong tháng 8. Yêu cầu các thông tin: MaNV, TenNV, NgaySinh, DiaChi, TenPB
3. Lập danh sách nhân viên của phòng có MaPB là “Tổng hợp”

4.5. Cho lược đồ cơ sở dữ liệu dùng để quản lý hồ sơ sinh viên được mô tả bởi các lược đồ quan hệ như sau:

Khoa(MaKhoa, TenKhoa)

Tân từ: Mỗi khoa có mỗi MAKHOA duy nhất. Mỗi MAKHOA xác định tất cả các thuộc tính còn lại của khoa đó.

Lop(MaLop, TenLop, SiSo, MaKhoa)

Tân từ: Mỗi lớp có một mã lớp duy nhất, mỗi lớp chỉ thuộc về một khoa nào đó.

MonHoc(MaMH, TenMH, SoTiet)

Tân từ: Mỗi Môn học có một MaMH duy nhất. Mỗi MaMH xác định tất cả các thuộc tính còn lại của môn học đó.

SinhVien(MaSV, HoTen, GioiTinh, NgaySinh, MaLop)

Tân từ: Mỗi sinh viên có mỗi MaSV duy nhất. Mỗi MaSV xác định tất cả các thuộc tính còn lại của sinh viên đó.

KetQua(MaSV, MaMH, LanThi, DiemThi)

Hãy tạo cơ sở dữ liệu và nhập dữ liệu cho các bảng. Dùng câu lệnh SQL để trả lời các câu truy vấn sau đây:

1. Lập danh sách sinh viên gồm MaSV, HoTen
2. Lập danh sách gồm MaSV, HoTen của các sinh viên nữ khoa 'CNTT'
3. Lập bảng điểm cho tất cả sinh viên khoa 'CNTT', bảng điểm gồm các cột MaSV, HoTen, TenMH, LanThi, DiemThi
4. Lập phiếu điểm cho sinh viên có MaSV='99001'
5. Lập danh sách sinh viên gồm MaSV, HoTen, TenLop, TenKhoa
6. Lập bảng điểm môn học có mã môn học là 'CSDL' cho tất cả sinh viên có mã lớp là 'CĐTH2B'
7. Lập danh sách sinh viên của lớp có mã lớp là 'CĐTH2B' và có điểm thi môn học lần 1 lớn hơn hay bằng 8.

4.6. Xét CSDL bảo hiểm sau:

person(ss#, name, address): Số bảo hiểm ss# sở hữu bởi người tên name ở địa chỉ address

car(license, year, model): Xe hơi số đăng ký license, sản xuất năm year, nhãn hiệu model

accident(date, driver, damage_amount): tai nạn xảy ra ngày date, do người lái driver, mức hư hại damage_amount

owns(ss#, license): người mang số bảo hiểm ss# sở hữu chiếc xe mang số đăng ký license

log(license, date, driver): ghi sổ chiếc xe mang số đăng ký license, bị tai nạn ngày date do người lái driver

Các thuộc tính được gạch dưới là các khóa chính.

Hãy tạo cơ sở dữ liệu và nhập dữ liệu cho các bảng. Dùng câu lệnh SQL để trả lời các câu truy vấn sau đây:

1. Thêm khách hàng mới: ss# = "A-12345", name = "David", address = "35 Chevre Road", license = "109283", year = "2002", model = "FORD LASER" vào CSDL.
2. Xóa các thông tin liên quan đến xe nhãn hiệu "MAZDA" của "John Smith".
3. Thêm thông tin tai nạn cho chiếc xe "TOYOTA" của khách hàng mang số bảo hiểm số "A-84626"

4.7. Xét CSDL nhân viên:

employee(E_name, street, city): Nhân viên có tên E_name, cư trú tại phố street, trong thành phố city

works(E_name, C_name, salary): Nhân viên tên E_name làm việc cho công ty C_name với mức lương salary

company(C_name, city): Công ty tên C_name đóng tại thành phố city

manages(E_name, M_name): Nhân viên E_name dưới sự quản lý của nhân viên M_name.

Hãy tạo cơ sở dữ liệu và nhập dữ liệu cho các bảng. Dùng câu lệnh SQL để trả lời các câu truy vấn sau đây:

1. Tìm tên của tất cả các nhân viên làm việc cho công ty "First Bank"
2. Tìm tên và thành phố cư trú của các nhân viên làm việc cho công ty "First Bank"
3. Tìm tên, phố, thành phố cư trú làm việc cho "First Bank" và hưởng mức lương lớn hơn 10000\$
4. Tìm tất cả nhân viên trong công ty CSDL sinh sống trong cùng thành phố của công ty nơi họ làm việc
5. Xóa tất cả các thông tin liên quan tới công ty "Bad Bank"

CHƯƠNG 5

LÝ THUYẾT THIẾT KẾ CƠ SỞ DỮ LIỆU QUAN HỆ

Khi thiết kế cơ sở dữ liệu quan hệ, chúng ta thường đứng trước vấn đề lựa chọn giữa các lược đồ quan hệ: lược đồ nào tốt hơn và tại sao? Phần này sẽ nghiên cứu một số tiêu chuẩn đánh giá lược đồ quan hệ và các cách thức giúp chúng ta xây dựng được lược đồ cơ sở dữ liệu quan hệ có cấu trúc tốt. Có thể nói tổng quát một lược đồ quan hệ có cấu trúc tốt là lược đồ không chứa đựng sự dư thừa dữ liệu, tức là sự trùng lặp thông tin trong cơ sở dữ liệu.

5.1. Mục đích của việc chuẩn hoá

Chuẩn hoá là một kỹ thuật để tạo ra một tập hợp các quan hệ thích hợp để hỗ trợ các yêu cầu về dữ liệu của một hoạt động.

Một CSDL được thiết kế tốt là CSDL hạn chế tối đa việc dư thừa dữ liệu nhưng vẫn không làm mất đi bất kỳ dữ liệu nào. Có nghĩa là chúng ta sử dụng không gian lưu trữ trong CSDL ít nhất nhưng vẫn bảo đảm tất cả mối quan hệ cùng với nội dung dữ liệu. Hơn nữa, chuẩn hoá lược đồ CSDL sẽ tránh được những dị thường trong thao tác chèn, sửa, xoá dữ liệu sau này và do đó bảo đảm tính toàn vẹn trong CSDL.

Về cơ bản, các quy tắc chuẩn hoá loại bỏ các dư thừa dữ liệu và những quan hệ phụ thuộc mâu thuẫn nhau giữa các bảng.

5.2. Dư thừa thông tin và cập nhật dị thường

Một lược đồ quan hệ có cấu trúc tốt là lược đồ không chứa đựng sự dư thừa dữ liệu.

5.2.1. Sự dư thừa dữ liệu

Dư thừa dữ liệu là sự trùng lặp thông tin trong cơ sở dữ liệu.

Ví dụ:

Xét quan hệ EMP (ENO, ENAME, TITLE, SAL, PNO, RESP, DUR).

Nếu một nhân viên tham gia trong nhiều dự án, thì các dữ liệu như ENAME, TITLE, SAL phải lặp lại nhiều lần và kéo theo dư thừa dữ liệu.

Ngoài việc gây lãng phí dung lượng lưu trữ, sự dư thừa dữ liệu có thể gây ra những hậu quả nghiêm trọng đối với dữ liệu khi người dùng cập nhật dữ liệu làm cho dữ liệu không tương thích, bất định hoặc mất mát. Các sự cố như vậy gọi là *những dị thường*.

5.2.2. Các dị thường cập nhật dữ liệu

Ta sẽ minh hoạ các dị thường bằng các lược đồ

EMP(ENO, ENAME, TITLE, SAL, PNO, RESP, DUR): Lược đồ về nhân viên gồm các thuộc tính Mã nhân viên (ENO), tên nhân viên (ENAME), chức vụ (TITLE), lương (SAL), Mã dự án mà nhân viên đó tham gia (PNO), vai trò của nhân viên trong dự án (RESP), thời gian (DUR).

PROJ(PNO, PNAME, BUDGET): Lược đồ về dự án gồm các thuộc tính Mã dự án (PNO), tên dự án (PNAME), kinh phí (BUDGET).

a. Dị thường do dữ liệu lặp: Một số thông tin có thể được lặp lại một cách vô ích.

Ví dụ:

Trong quan hệ EMP thì tên (ENAME), chức vụ (TITLE), và lương (SAL) của nhân viên được lặp lại trong mỗi dự án mà họ tham gia. Điều này rõ ràng là làm lãng phí chỗ lưu trữ và đối nghịch với các nguyên lý của cơ sở dữ liệu.

b. Dị thường chèn bộ: Không thể chèn bộ mới vào quan hệ, nếu không có đầy đủ dữ liệu.

Ví dụ:

Xét quan hệ EMP. Giả sử một nhân viên mới được nhận vào công ty và chưa được phân công vào dự án nào cả. Khi đó chúng ta không thể nhập các thông tin về tên, chức vụ, lương của nhân viên này vào quan hệ, vì khóa của EMP là (ENO, PNO).

c. Dị thường xóa bộ: Trường hợp này ngược với dị thường chèn bộ. Việc xóa bộ có thể kéo theo mất thông tin.

Ví dụ:

Xét quan hệ EMP. Giả sử một nhân viên làm việc trong một dự án duy nhất. Khi dự án chấm dứt, chúng ta không thể xóa thông tin về dự án đó trong EMP được, vì nếu làm thế ta sẽ mất luôn thông tin về nhân viên đó.

d. Dị thường sửa bộ: Việc sửa đổi dữ liệu dư thừa có thể dẫn đến sự không tương thích dữ liệu.

Ví dụ: Xét quan hệ EMP. Giả sử một nhân viên làm việc trong nhiều dự án. Khi có sự thay đổi về lương, rất nhiều bộ phải cập nhật sự thay đổi này. Điều đó gây lãng phí thời gian công sức và là nguy cơ gây ra sự không thống nhất dữ liệu.

Trong các ví dụ trên ta thấy tác hại của sự dư thừa dữ liệu và sự cần thiết phải loại bỏ chúng khỏi các lược đồ quan hệ. Quá trình từng bước thay thế một lược đồ quan hệ bằng các tập lược đồ quan hệ đơn giản và chuẩn tắc hơn gọi là *chuẩn hoá*. Mục đích của chuẩn hoá là loại bỏ các dị thường (hoặc các khía cạnh không mong muốn khác) để có những quan hệ tốt hơn.

Cơ sở lý thuyết của việc thiết kế lược đồ cơ sở dữ liệu quan hệ tốt là khái niệm *phụ thuộc dữ liệu*. Phụ thuộc dữ liệu biểu diễn các quan hệ nhân quả giữa các thuộc tính trong quan hệ. Ví dụ trong bảng EMP, thuộc tính SAL phụ thuộc vào thuộc tính ENO, vì mỗi nhân viên chỉ có một lương duy nhất.

Cũng dựa trên khái niệm phụ thuộc dữ liệu người ta định nghĩa các *dạng chuẩn của lược đồ dữ liệu quan hệ*. Mỗi dạng chuẩn đáp ứng một yêu cầu nhất định đối với lược đồ quan hệ.

Quá trình biến đổi một lược đồ thành lược đồ tương đương (bảo toàn thông tin và phụ thuộc dữ liệu) thoả mãn dạng chuẩn gọi là quá trình *chuẩn hoá lược đồ quan hệ*.

5.3. Phụ thuộc hàm

Phụ thuộc hàm có tầm quan trọng đối với người quản trị cơ sở dữ liệu trong việc thiết kế và cài đặt các mô hình cơ sở dữ liệu quan hệ. Cơ sở lý thuyết về chuẩn hoá dữ liệu dựa trên các khái niệm phụ thuộc hàm và khoá của quan hệ. Phụ thuộc hàm là khái niệm được xây dựng để mô tả các ràng buộc trong cơ sở dữ liệu, chẳng hạn, mã mặt hàng xác định số lượng, đơn giá, ngày nhập kho...của một mặt hàng, hay nói cách khác, mỗi một giá trị của thuộc tính mã mặt hàng xác định duy nhất giá trị của thuộc tính số lượng, đơn giá...của mặt hàng. Ràng buộc này sẽ từ chối khi chèn thêm thông tin về một mặt hàng mới mà chưa được xác định mã mặt hàng, vì sẽ mâu thuẫn, không nhất quán trong tổ chức lưu trữ dữ liệu.

5.3.1. Định nghĩa phụ thuộc hàm

Cho lược đồ quan hệ $R=(A1, A2, ..., An)$ và X, Y là các tập con của $\{A1, A2, ..., An\}$. Ta nói rằng X xác định hàm Y hay Y phụ thuộc hàm X , ký hiệu $X \rightarrow Y$, nếu mọi quan hệ bất kỳ r của lược đồ R thoả mãn: $\forall u, v \in r : u(X) = v(X) \Rightarrow u(Y) = v(Y)$

Phụ thuộc hàm $X \rightarrow Y$ gọi là *phụ thuộc hàm tầm thường* nếu $Y \subset X$ (hiển nhiên là nếu $Y \subset X$ thì theo định nghĩa ta có $X \rightarrow Y$).

Phụ thuộc hàm $X \rightarrow Y$ gọi là *phụ thuộc hàm nguyên tố* nếu không có tập con thực sự $Z \subset X$ thoả $Z \rightarrow Y$.

Tập thuộc tính $K \subset R$ gọi là *khóa* nếu nó xác định hàm tất cả các thuộc tính và $K \rightarrow R$ là phụ thuộc hàm nguyên tố.

Ví dụ: Xét lược đồ quan hệ về danh mục vật tư gồm các thuộc tính về mã vật tư, tên vật tư, đơn giá: DMVT(MaVT, TenVT, DonGia). Có phụ thuộc hàm:

MaVT \rightarrow TenVT, DonGia

X Y

Ví dụ: Xét lược đồ quan hệ về chứng từ vật tư có các thuộc tính về số chứng từ, khách hàng, tên mặt hàng, số lượng: CTVT(SoCT, KhachHang, TenHang, SoLuong). Có phụ thuộc hàm:

SoCT \rightarrow KhachHang

SoCT, KhachHang, TenHang \rightarrow SoLuong

Ví dụ: Xét các quan hệ:

EMP(ENO, ENAME, TITLE, SAL, PNO, RESP, DUR)

PROJ(PNO, PNAME, BUDGET)

Đối với quan hệ PROJ. Ta có thể chấp nhận rằng mỗi dự án có tên và kinh phí xác định. Vậy có thể khẳng định

PNO \rightarrow PNAME, BUDGET

Trong quan hệ EMP ta có

ENO, PNO \rightarrow ENAME, TITLE, SAL, RESP, DUR

$ENO \rightarrow ENAME, TITLE, SAL$

Chúng ta có thể cho rằng lương của mỗi chức vụ là cố định, do đó sẽ tồn tại phụ thuộc hàm

$TITLE \rightarrow SAL$

Phụ thuộc hàm $X \rightarrow Y$ là đầy đủ khi và chỉ khi: $\nexists X' \subset X$ mà $X' \rightarrow Y$

Ví dụ: Xét lược đồ quan hệ KháchHang(MaKhach, HoTen, DiaChi)

$MaKhach \rightarrow HoTen, DiaChi$

$MaKhach, HoTen \rightarrow HoTen, DiaChi$ (Dư thừa)

5.3.2. Các qui tắc phụ thuộc hàm

5.3.2.1. Hệ tiên đề Armstrong cho các phụ thuộc hàm

Cho $\Omega := \{A_1, A_2, \dots, A_n\}$ là tập khác rỗng. Gọi F là tập các phụ thuộc hàm thỏa trên các quan hệ R trên tập các thuộc tính Ω . Khi đó nếu $\forall A, B, C, D \subseteq \Omega$

A1: Phản xạ: Nếu với mọi $B \subseteq A \Rightarrow A \rightarrow B$

A2: Gia tăng: Nếu $A \rightarrow B \Rightarrow AC \rightarrow B, AC \rightarrow BC$

A3: Bắc cầu: Nếu $A \rightarrow B$ và $B \rightarrow C$ thì suy ra $A \rightarrow C$

A4: Giả bắc cầu: Nếu $A \rightarrow B$ và $BC \rightarrow Z \Rightarrow AC \rightarrow Z$. Áp dụng A2 và A3 có thể suy ra A4: $A \rightarrow B \Rightarrow AC \rightarrow BC$ (A2), $BC \rightarrow Z \Rightarrow AC \rightarrow Z$ (A3)

A5: Hợp: Nếu $A \rightarrow B$ và $A \rightarrow C \Rightarrow A \rightarrow BC$. Áp dụng A2: $A \rightarrow B \Rightarrow AA \rightarrow AB$ và $A \rightarrow C \Rightarrow AB \rightarrow BC$. Áp dụng A3: $AA \rightarrow C$, tức là $A \rightarrow BC$.

A6: Tách: Nếu $A \rightarrow BC \Leftrightarrow A \rightarrow B$ và $A \rightarrow C$. Nghĩa là nếu vế phải bao gồm nhiều thuộc tính, khi đó thuộc tính vế trái sẽ xác định các thành phần trong vế phải. Tiên đề được suy dẫn từ các tiên đề A1, A2 và A3 như sau: $A \rightarrow BC \Leftrightarrow AA \rightarrow ABC \Leftrightarrow A \rightarrow ABC$, áp dụng quy tắc A1: $ABC \rightarrow B$ và $ABC \rightarrow C \Leftrightarrow A \rightarrow B$ và $A \rightarrow C$.

5.3.2.2. Các tính chất của phụ thuộc hàm

A1. Tính phản xạ: Nếu $B \subseteq A$ khi đó $A \rightarrow B$.

A2. Tính gia tăng: Nếu $A \rightarrow B$ và $C \subseteq \Omega$ khi đó $AC \rightarrow BC$.

A3. Tính bắc cầu: Nếu $A \rightarrow B$ và $B \rightarrow C$ khi đó $A \rightarrow C$.

A4. Quy tắc hợp: Nếu $A \rightarrow B$ và $A \rightarrow C$ khi đó $A \rightarrow BC$.

A5. Quy tắc tách: Nếu $A \rightarrow B$ và $C \subseteq B$ khi đó $A \rightarrow C$.

Ví dụ:

Cho lược đồ $R=ABC$ và $F=\{AB \rightarrow C, C \rightarrow A\}$. Chứng minh rằng $BC \rightarrow ABC$

Ta có

1. $C \rightarrow A$ (theo giả thiết)
2. $BC \rightarrow AB$ (luật 1 thêm B)
3. $AB \rightarrow C$ (giả thiết)
4. $AB \rightarrow ABC$ (luật 3 thêm AB)
5. $BC \rightarrow ABC$ (luật bắc cầu từ 2 đến 4)

Ví dụ:

Cho $\{AB \rightarrow E, AG \rightarrow I, BE \rightarrow I, E \rightarrow G, GI \rightarrow H\}$. Chứng minh $AB \rightarrow GH$

1. $AB \rightarrow E; E \rightarrow G \Rightarrow AB \rightarrow G$
2. $AB \rightarrow G \Rightarrow AB \rightarrow AG$ mà $AG \rightarrow I \Rightarrow AB \rightarrow I$
 $AB \rightarrow G \Rightarrow AB \rightarrow GI \Rightarrow AB \rightarrow H$

Từ (1) và (2): $AB \Rightarrow GH$

5.4. Suy diễn lô-gíc

Giả sử F là tập các phụ thuộc hàm trên lược đồ quan hệ R và X, Y là các tập con thuộc tính của R . Ta nói rằng F suy diễn lôgic phụ thuộc hàm $X \rightarrow Y$ hay phụ thuộc hàm $X \rightarrow Y$ được suy diễn lôgic từ F , ký hiệu

$$F \models X \rightarrow Y$$

nếu mọi quan hệ r thoả các phụ thuộc hàm trong F cũng thoả phụ thuộc hàm $X \rightarrow Y$.

Ví dụ: $\{A \rightarrow B, B \rightarrow C\} \models A \rightarrow C$

5.5. Bao đóng của tập phụ thuộc hàm

5.5.1. Định nghĩa

Bao đóng của tập phụ thuộc hàm F , ký hiệu là F^+ , là tập hợp tất cả các phụ thuộc hàm suy diễn lôgic từ F :

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

Ví dụ: Cho $R=(A,B,C)$ và $F = \{A \rightarrow B, B \rightarrow C\}$. Khi đó bao đóng F^+ gồm các phụ thuộc hàm $X \rightarrow Y$ thoả

(i) X chứa A , và Y bất kỳ:

- | | | | |
|--------------------------------|-----------------------------|-----------------------------|-----------------------------|
| $A, B, C \rightarrow A, B, C;$ | $A, B, C \rightarrow A, B;$ | $A, B, C \rightarrow A, C;$ | $A, B, C \rightarrow B, C;$ |
| $A, B, C \rightarrow A;$ | $A, B, C \rightarrow B;$ | $A, B, C \rightarrow C;$ | |
| $A, B \rightarrow A, B, C;$ | $A, B \rightarrow A, B;$ | $A, B \rightarrow A, C;$ | $A, B \rightarrow B, C;$ |
| $A, B \rightarrow A;$ | $A, B \rightarrow B;$ | $A, B \rightarrow C;$ | |

$A, C \rightarrow A, B, C;$	$A, C \rightarrow A, B;$	$A, C \rightarrow A, C;$	$A, C \rightarrow B, C;$
$A, C \rightarrow A;$	$A, C \rightarrow B;$	$A, C \rightarrow C;$	
$A \rightarrow A, B, C;$	$A \rightarrow A, B;$	$A \rightarrow A, C;$	$A \rightarrow B, C;$
$A \rightarrow A;$	$A \rightarrow B;$	$A \rightarrow C;$	

(ii) X chứa B nhưng không chứa A, Y không chứa A:

$B, C \rightarrow B, C;$	$B, C \rightarrow B;$	$B, C \rightarrow C;$
$B \rightarrow B, C;$	$B \rightarrow B;$	$B \rightarrow C$

(iii) $C \rightarrow C$

Ví dụ:

1. Cho $F = \{A \rightarrow B, B \rightarrow C, A \rightarrow D, B \rightarrow D\}$.

Khi đó $A \rightarrow C \in F^+$, $B \rightarrow DC \in F^+$ và $A \rightarrow BC \in F^+$

- Áp dụng quy tắc bắc cầu, từ $A \rightarrow B, B \rightarrow C$, suy ra. $A \rightarrow C \in F^+$
- Vì $B \rightarrow C$ và $B \rightarrow D$, suy ra $B \rightarrow DC \in F^+$
- Vì $A \rightarrow B$ và $A \rightarrow C \in F^+$, suy ra $A \rightarrow BC \in F^+$

2. Cho $F = \{A \rightarrow B, C \rightarrow X, BX \rightarrow Z\}$. Khi đó $AC \rightarrow Z \in F^+$?

- Vì $A \rightarrow B \Rightarrow AX \rightarrow BX$
- Từ $AX \rightarrow BX$, kết hợp $BX \rightarrow Z$, suy ra $AX \rightarrow Z$
- Từ $C \rightarrow X \Rightarrow AC \rightarrow AX$
- Áp dụng tính chất bắc cầu, $AC \rightarrow AX$ và $AX \rightarrow Z$ suy ra $AC \rightarrow Z \in F^+$

3. Cho $F = \{A \rightarrow B, C \rightarrow D\}$, $C \subset B$, chứng tỏ rằng $A \rightarrow D \in F^+$?

- Vì $C \subset B$, áp dụng tính chất phản xạ, suy ra $B \rightarrow C$
- Từ $A \rightarrow B$ và $B \rightarrow C$ suy ra $A \rightarrow C$
- Từ $A \rightarrow C$ và $C \rightarrow D$ suy ra $A \rightarrow D \in F^+$

5.5.2. Phụ thuộc hàm đầy đủ

Phụ thuộc hàm $X \rightarrow Y$ là đầy đủ khi và chỉ khi không tồn tại $X' \subset X$ mà $X' \rightarrow Y$

Ví dụ:

Cho $F = \{A \rightarrow B, AC \rightarrow B, B \rightarrow C\}$

$AC \rightarrow B$ là không đầy đủ vì $A \rightarrow B$ ($A \subset AC$)

5.5.3. Phụ thuộc hàm dư thừa

Cho tập phụ thuộc hàm F , $X \rightarrow Y \in F$ là dư thừa khi $F \setminus X \rightarrow Y \models X \rightarrow Y$

Ví dụ:

Cho $F = \{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$

$A \rightarrow C$ là dư thừa vì $A \rightarrow B$ và $B \rightarrow C \Rightarrow A \rightarrow C$

5.5.4. Phủ của tập phụ thuộc hàm, phụ thuộc hàm tương đương

Cho hai tập phụ thuộc hàm F và G . G được gọi là phủ của F nếu $F \subset G^+$ và F tương đương với G nếu $F \subset G^+$ và $G \subset F^+$.

5.6. Bao đóng của tập thuộc tính

5.6.1. Định nghĩa

Bao đóng của tập thuộc tính $X \subset R$ (đối với tập phụ thuộc hàm F), ký hiệu là X^+ , là tập hợp tất cả các thuộc tính phụ thuộc hàm vào X : $X^+ = \{A \mid X \rightarrow A \in F^+\}$

Ví dụ: Cho $R = (A, B, C)$ và $F = \{A \rightarrow B, B \rightarrow C\}$. Khi đó ta dễ dàng thấy bao đóng của thuộc tính B là $B^+ = \{B, C\}$.

Ví dụ: Cho bảng Chúng từ vật tư có các trường như sau CTVT(A, B, C, D, E, F)

$A \rightarrow B, C$

$C \rightarrow D$

$A, C, E \rightarrow F$

$X = \{A, C, E\}$

$X^+ = \{A, B, C, D, E, F\} = \text{CTVT}$

5.6.2. Thuật toán tìm bao đóng

- Đầu vào: Tập các thuộc tính R , tập các phụ thuộc hàm F trên R và tập $X \subset R$

- Đầu ra: X^+ (Bao đóng X^+ của X đối với F)

- **Phương pháp:** Ta tính lần lượt dãy các tập thuộc tính X_0, X_1, \dots, X_n như sau:

Đặt $X_0 = X$

Tính X_i như sau: $X_i = X_{i-1} \cup A$ nếu có $X_{i-1} \rightarrow A$, nếu không $X_i = X_{i-1}$

Kiểm tra điều kiện kết thúc: $X_i = R$ hoặc không có phụ thuộc hàm nào thỏa mãn.

Ví dụ:

Cho $R = ABCDEF$

$F = \{A \rightarrow BC, C \rightarrow D, AC \rightarrow F\}$, $X = ACE$. Hãy tính X^+

Ta có:

$X_0 = ACE$

$X_1 = ACEB$ vì $A \rightarrow BC$

$X_2 = ABCED$ vì $C \rightarrow D$

$X_3 = ABCDE$ vì $ACE \rightarrow F$

Vậy $X^+ = ABCDEF$

5.7. Khóa và siêu khóa

Cho lược đồ quan hệ $R = (A_1, \dots, A_n)$ và tập phụ thuộc hàm F trên R .

Tập con $X \subset \{A_1, \dots, A_n\}$ là khóa của R nếu $X \rightarrow A_1, \dots, A_n \in F^+$ là phụ thuộc hàm nguyên tố.

Tập $S \subset \{A_1, \dots, A_n\}$ là siêu khóa của R nếu S chứa khóa.

- Ghi chú: Nếu $A \rightarrow R$, thì A là khóa của lược đồ R .

Ví dụ: Xét lược đồ quan hệ $R = (A, B, C)$ với tập phụ thuộc hàm $F = \{A \rightarrow B, B \rightarrow C\}$. Ta có khóa duy nhất là (A) , vì $A \rightarrow (A, B, C)$. Mọi tập thuộc tính chứa A là siêu khóa.

Chúng ta có thể dùng bao đóng của tập thuộc tính để xác định khóa của một lược đồ quan hệ.

Định lý: Cho U là một tập các thuộc tính, F là tập phụ thuộc hàm trên U và $X \subseteq U$. X là khóa của U dưới F khi và chỉ khi $X^+ = U$.

Ví dụ: Cho $R = ABCDEG$ và $F = \{AE \rightarrow C, CG \rightarrow A, BD \rightarrow G, GA \rightarrow E\}$

$K = ABD$ là siêu khóa của R vì $(ABD)^+ = ABDGEC$

Ví dụ: Cho $\Omega = \{A, B, C, D, E, G\}$

$F = \{AB \rightarrow C, D \rightarrow EG, C \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG\}$

Khi đó tập khóa của lược đồ quan hệ là : $K = \{AB, CG, CD, EB, CE, BC\}$

$K_1 = AB$ vì $(AB)^+ = ABCDEG$ $K_2 = EB$ vì $(EB)^+ = ABCDEG$

$K_3 = CG$ vì $(CG)^+ = ABCDEG$ $K_4 = CE$ vì $(CE)^+ = ABCDEG$

$K_5 = CD$ vì $(CD)^+ = ABCDEG$ $K_6 = BC$ vì $(BC)^+ = ABCDEG$.

5.8. Phép tách lược đồ quan hệ

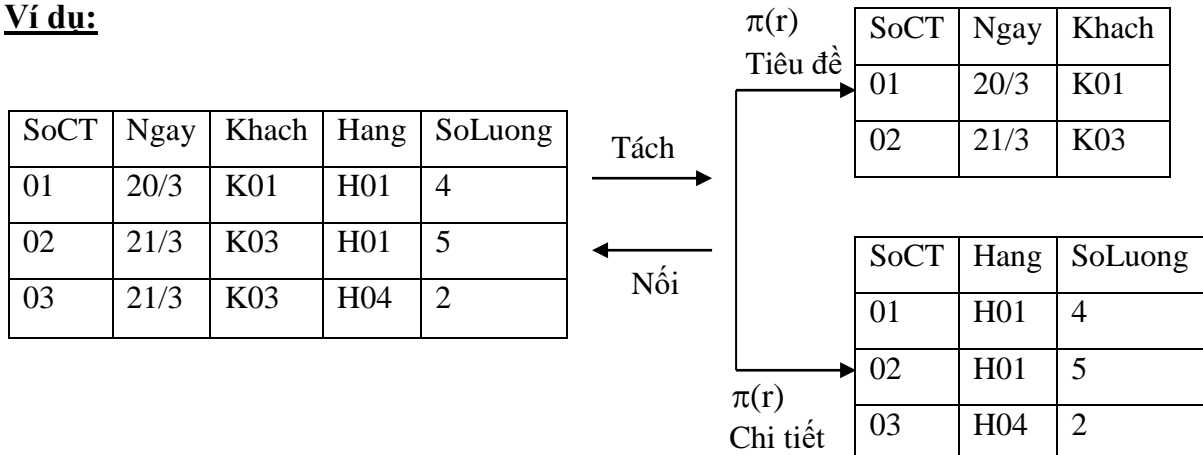
5.8.1. Định nghĩa

Cho lược đồ quan hệ $R = A_1 A_2 \dots A_n$. Tách lược đồ quan hệ R là thay thế R bằng các lược đồ con R_1, R_2, \dots, R_m sao cho $R_1 \cup R_2 \cup \dots \cup R_m = R$ và $R_i \neq R_j$ khi $i \neq j$

5.8.2. Phép tách bảo toàn thông tin

$\rho(R) = (R_1, R_2, \dots, R_m)$ bảo toàn thông tin $\Leftrightarrow \forall r(R) = \pi_{R_1}(r) * \pi_{R_2}(r) * \dots * \pi_{R_m}(r)$

Ví dụ:



5.8.3. Thuật toán kiểm tra phép tách bảo toàn thông tin

- Đầu vào: $R = A_1A_2...A_n$ và $\rho(R) = (R_1, R_2,...R_m)$

- Đầu ra: $\rho(R)$ bảo toàn thông tin hay không?

- Phương pháp:

Bước 1: Lập bảng gồm m dòng và n cột. Dòng thứ i tương ứng lược đồ con R_i , cột thứ j tương ứng thuộc tính A_j

Tại vị trí (i,j) ta ký hiệu a_{ij} nếu $A_j \in R_i$, ngược lại ký hiệu b_{ij}

Bước 2: Dựa vào các phụ thuộc hàm để làm bảng theo nguyên tắc: Xét $X \rightarrow Y$, nếu trên các dòng mà giá trị X bằng nhau ưu tiên cho ký hiệu a_j

Lặp lại bước 2 cho đến khi

+ Có một dòng chứa toàn ký hiệu a_j . Khi đó kết luận $\rho(R)$ bảo toàn thông tin

+ Không áp dụng được phụ thuộc hàm nào nữa. Khi đó kết luận $\rho(R)$ mất thông tin

Ví dụ: Cho $R = ABCDE$ và $F = \{A \rightarrow BC, ACD \rightarrow E\}$

$\rho(R) = (ABC, ADE)$ có bảo toàn thông tin hay không?

	A	B	C	D	E	$A \rightarrow BC$ \longrightarrow		A	B	C	D	E
ABC	a_1	a_2	a_3	b_{14}	b_{15}		ABC	a_1	a_2	a_3	b_{14}	b_{15}
ADE	a_1	b_{22}	b_{23}	a_4	a_5		ADE	a_1	a_2	a_3	a_4	a_5

Vậy $\rho(R)$ bảo toàn thông tin

Ví dụ 5.25: Cho $R = ABCD$ và $F = \{A \rightarrow B, AC \rightarrow D\}$

$\rho(R) = (AB, ACD)$ có bảo toàn thông tin hay không?

	A	B	C	D
AB	a ₁	a ₂	b ₁₃	b ₁₄
ACD	a ₁	b ₂₂	a ₃	a ₄

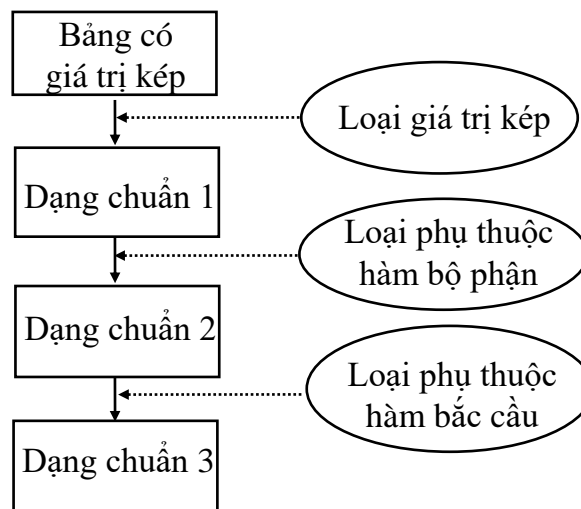
$$\xrightarrow{A \rightarrow B}$$

	A	B	C	D
AB	a ₁	a ₂	b ₁₃	b ₁₄
ACD	a ₁	a ₂	a ₃	a ₄

Vậy $\rho(R)$ bảo toàn thông tin

5.9. Quy trình chuẩn hoá

Khi thực hiện các phép lưu trữ trên các quan hệ chưa được chuẩn hoá thường xuất hiện các dị thường thông tin. Nghĩa là trong dữ liệu lưu trữ, sự dư thừa, mất dữ liệu, mâu thuẫn hay không nhất quán dữ liệu có thể xảy ra khi cập nhật, bổ sung hay sửa đổi dữ liệu. Dị thường thông tin là nguyên nhân gây cản trở cho việc tìm kiếm, hỏi đáp thông tin. Mục tiêu của chuẩn hoá dữ liệu là triệt tiêu mức cao nhất khả năng xuất hiện các dị thường thông tin khi thực hiện các phép lưu trữ. Có như vậy mục tiêu của các hệ cơ sở mới được bảo đảm. Dữ liệu lưu trữ phản ánh thế giới hiện thực khách quan, đầy đủ hơn và sinh động hơn.



Hình 5.1 Quy trình chuẩn hoá

Khi thiết kế và cài đặt các hệ CSDL, chuẩn hoá là quá trình khảo sát danh sách các thuộc tính và áp dụng tập các quy tắc phân tích vào danh sách đó, biến đổi chúng thành nhiều tập nhỏ hơn sao cho:

- Tối thiểu việc lặp lại
- Tránh dị thường thông tin
- Xác định và giải quyết được sự không rõ ràng, nhập nhằng trong suy diễn

Chuẩn hoá lược đồ quan hệ thường được thực hiện qua các giai đoạn tương ứng với các dạng chuẩn (Hình 5.1). Dạng chuẩn là trạng thái quan hệ được xác định bằng cách áp dụng các quy tắc đối với phụ thuộc hàm của quan hệ.

Ngoài các dạng chuẩn 1, 2, 3 còn có dạng chuẩn Boyce-Codd, chuẩn 4 và dạng chuẩn 5, tuy nhiên chúng ta sẽ không nghiên cứu trong phạm vi giáo trình này.

5.9.1. Dạng chuẩn một (1NF)

Một lược đồ quan hệ R được gọi là ở dạng chuẩn thứ nhất nếu và chỉ nếu toàn bộ các miền có mặt trong R đều chỉ chứa các giá trị nguyên tố (không phân chia được nữa).

Ví dụ:

Xét lược đồ quan hệ Sinh viên:

MASV	HOVATEN	KHOA	TENMONHOC	DIEMTHI
99023	NGUYENTHITHU	CONG NGHE THONG TIN	KY THUAT LAP TRINH	6
			TOAN ROI RAC	8
			CO SO DU LIEU	4
99030	LE VAN THANH	DIEN TU	VI XULY	4

Quan hệ này không đạt dạng chuẩn 1 vì các thuộc tính TENMONHOC, DIEMTHI của bộ thứ nhất không mang giá trị đơn. Chẳng hạn, sinh viên NGUYENTHITHU có thuộc tính TENMONHOC là KYTHUATLAPTRINH, TOANROIIRAC, COSODULIEU.

Ví dụ:

Xét lược đồ quan hệ Hóa đơn: HOADON(MaHang, MaKH, SoLuong, DonGia, ThanhTien) không có dạng chuẩn 1 vì các giá trị thuộc tính ThanhTien=SoLuong x DonGia.

Để đưa lược đồ quan hệ về dạng chuẩn 1:

- Loại bỏ các thuộc tính có thể tính toán từ thuộc tính khác
- Biến cột đa trị thành đơn trị
- Điền đủ dữ liệu vào các cột khác

Chẳng hạn, trong lược đồ quan hệ Sinh viên có thể đưa về:

SINHVIEN(MaSV, Hovaten, Khoa)

DIEM_SINHVIEN(MaSV, TenMonHoc, DiemThi)

Đối với quan hệ Hóa đơn: Loại bỏ thuộc tính về thành tiền.

5.9.2. Dạng chuẩn thứ 2 (2NF)

Giả sử K là khóa của lược đồ R . Khi đó mọi thuộc tính không khóa A của R đều phụ thuộc hàm vào khóa K : $K \rightarrow A$. Nếu A không phụ thuộc đầy đủ vào K thì tồn tại tập con thực sự H của K xác định A , tức $H \rightarrow A$. Khi đó phụ thuộc hàm $H \rightarrow A$ gọi là phụ thuộc hàm bộ phận.

Một lược đồ quan hệ R là ở dạng chuẩn thứ 2 nếu nó ở dạng chuẩn thứ 1 và không có phụ thuộc hàm bộ phận, tức là mọi thuộc tính không khóa đều phụ thuộc đầy đủ vào các khóa của lược đồ.

Ví dụ: Xét các lược đồ quan hệ sau:

EMP(ENO, ENAME, TITLE, SAL, PNO, RESP, DUR)

PROJ(PNO, PNAME, BUDGET)

Lược đồ của EMP có khóa là (ENO, PNO).

Phụ thuộc hàm $ENO \rightarrow ENAME, TITLE$ là phụ thuộc hàm bộ phận vì vế phải là tập con thực sự của khóa. Vậy EMP không ở dạng chuẩn thứ 2.

Lược đồ của PROJ không có phụ thuộc hàm bộ phận, vậy nó ở dạng chuẩn 2.

Ví dụ:

Cho lược đồ quan hệ HocVien(MaHV, MaMH, TenHV, Diem)

$MaHV \rightarrow TenHV$

$MaHV, MaMH \rightarrow Diem$

Với các thuộc tính không khóa TenHV, Diem thì TenHV phụ thuộc một phần của khóa chính mà không phụ thuộc hoàn toàn. Vậy lược đồ quan hệ HocVien không ở dạng chuẩn 2.

Chú ý:

- Chỉ kiểm tra các quan hệ có đạt 2NF nếu quan hệ đó có khóa chính gồm 2 thuộc tính trở lên.
- Để chuyển quan hệ từ dạng 1NF sang dạng 2NF, chúng ta dùng phép chiếu để tách các lược đồ.

Ví dụ:

Bảng R gồm các thuộc tính MF, Tenfim, NSX, Giathue, HSX, NPP, MaKH, TenKH, Diachi, Ngaydat. R có các phụ thuộc hàm sau:

$MF \rightarrow Tenfim, NSX, Giathue, HSX, NPP$

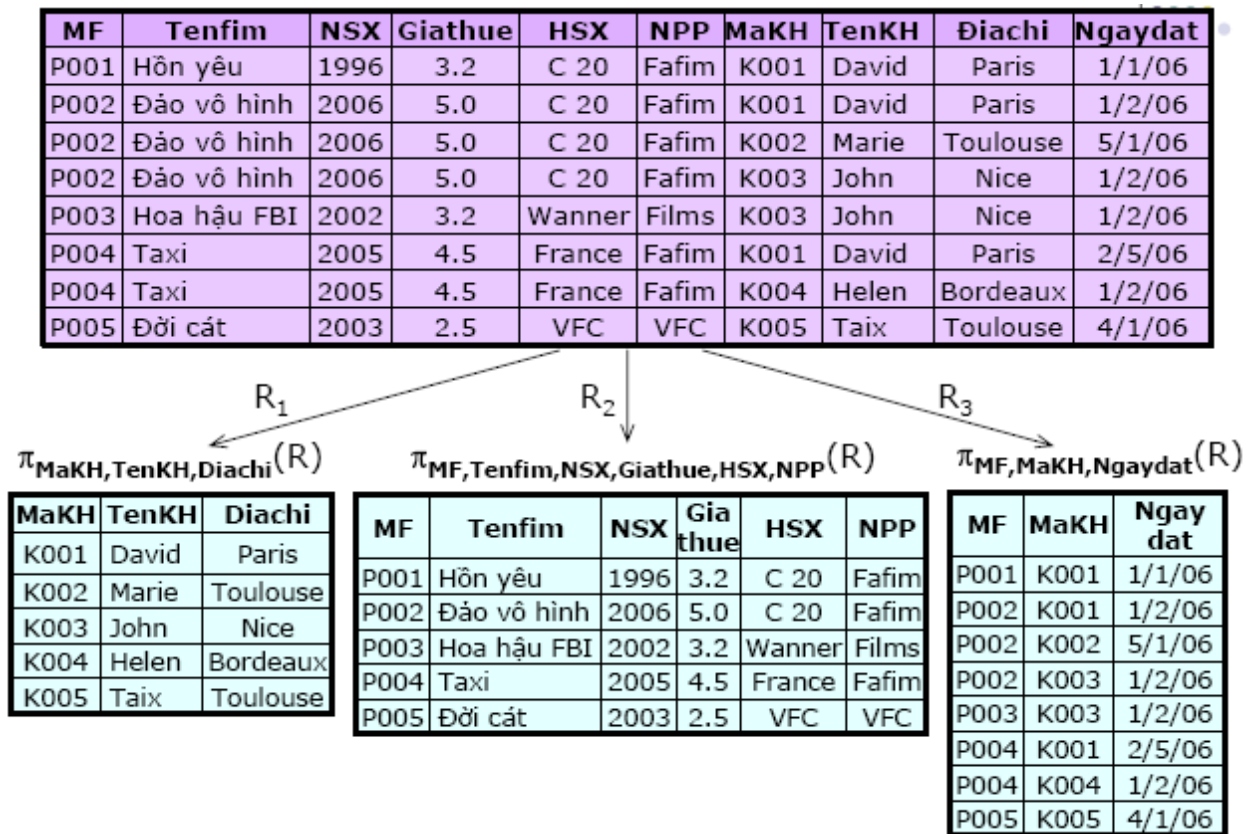
$MaKH \rightarrow TenKH, Diachi$

$MF, MaKH \rightarrow Ngaydat$

Khóa chính: MF, MaKH.

Các thuộc tính Tenfim, Giathue, TenKH, Diachi,...là các thuộc tính không khóa, chỉ phụ thuộc vào một bộ phận của khóa \rightarrow R không đạt chuẩn 2

Để chuyển về dạng chuẩn 2, chúng ta thường sử dụng phép chiếu. Chẳng hạn chúng ta có thể sử dụng phép chiếu như sau (Hình 5.2):



Hình 5.2. Sử dụng phép chiếu tách về 2NF

5.9.3. Dạng chuẩn thứ 3 (3NF)

Phụ thuộc hàm $X \rightarrow A$ gọi là phụ thuộc hàm bậc cầu, nếu nó là phụ thuộc hàm nguyên tố, A là thuộc tính không khóa, $A \rightarrow X$, và X chứa thuộc tính không khóa. Khi đó với mọi khóa K ta có các phụ thuộc hàm không tầm thường $K \rightarrow X$ & $X \rightarrow A$. Mặt khác không thể có $X \rightarrow K$ vì X chứa các thuộc tính không khóa và không chứa khóa (vì $X \rightarrow A$ là nguyên tố).

Nói một cách khác phụ thuộc hàm bậc cầu là sự phụ thuộc không tầm thường giữa các thuộc tính không khóa.

Một lược đồ quan hệ gọi là ở dạng chuẩn thứ 3 nếu nó ở dạng chuẩn thứ 2 và không có phụ thuộc hàm bậc cầu.

Ví dụ:

- Xét lược đồ quan hệ EMP(ENO, ENAME, TITLE, SAL, PNO, RESP, DUR). Lược đồ của quan hệ có TITLE \rightarrow SAL là phụ thuộc hàm bậc cầu. Vậy EMP không ở dạng chuẩn thứ 3.

- Lược đồ của quan hệ PROJ(PNO, PNAME, BUDGET) không có phụ thuộc hàm bậc cầu, vậy nó ở dạng chuẩn 3.

Để đưa lược đồ quan hệ về dạng chuẩn 3, chúng ta sử dụng các thuật toán chuẩn hóa lược đồ quan hệ ở phần 5.9.4.

5.9.4. Thuật toán chuẩn hóa một lược đồ cơ sở dữ liệu

Thuật toán 1:

- Đầu vào: $\langle R, F \rangle$
- Đầu ra: $\rho(R)$ thoả 3NF bảo toàn thông tin
- Phương pháp:

Bước 1: Loại bỏ trong R những thuộc tính không thuộc về phụ thuộc hàm nào

Bước 2: Thu gọn các phụ thuộc hàm

Nếu $X \rightarrow A_1, X \rightarrow A_2, X \rightarrow A_n$ Thì $X \rightarrow A_1A_2...A_n$

Bước 3: Mọi phụ thuộc hàm chuyển thành một lược đồ con

Ví dụ:

$R=ABCDEFGHIJ$ và $F=\{A \rightarrow BC, D \rightarrow AF, DG \rightarrow H, G \rightarrow IJ\}$

$A \rightarrow BC \Rightarrow R_1=ABC$

$D \rightarrow AF \Rightarrow R_2=DAF$

$DG \rightarrow H \Rightarrow R_3=DGH$

$G \rightarrow IJ \Rightarrow R_4=GIJ$

$\Rightarrow \rho(R) = (ABC, DAF, DGH, GIJ)$

Thuật toán 2:

- Đầu vào: $\langle R, F \rangle$
- Đầu ra: $\rho(R)$ thoả 3NF bảo toàn thông tin
- Phương pháp:

Bước 1: Tìm khoá của R và giả sử F là đầy đủ và không dư thừa

Bước 2: Nếu $X \rightarrow A$ và X không chứa khoá của R: $R=(XA, R \setminus A)$

Lặp lại bước 2 với $R \setminus A$ cho đến khi không tách được.

Ví dụ:

$R = MTGPSL, F = \{M \rightarrow T, GP \rightarrow M, GT \rightarrow P, MS \rightarrow L, GS \rightarrow P\}$

Khoá của R là GS

Ta có:

$M \rightarrow T \Rightarrow R_1 = MT, R = MGPSL$

$MS \rightarrow L \Rightarrow R_2 = MSL, R=MGPS$

$GP \rightarrow M \Rightarrow R_{23} = GPM, R=GPS$

$\Rightarrow \rho(R) = (MT, MSL, GPM, GPS)$

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 5

5.1. Nêu các mục đích của việc chuẩn hoá

5.2. Cho ví dụ về dư thừa dữ liệu

5.3. Cho ví dụ về cập nhật dị thừa

5.4. Cho ví dụ về độc lập chức năng

5.5. Nêu các bước quy trình chuẩn hoá

5.5. Kể tên các loại chuẩn hoá và nêu ví dụ cụ thể

5.7. Cho $F = \{AB \rightarrow E, AG \rightarrow I, E \rightarrow G, GI \rightarrow H\}$.

Chứng minh rằng $AB \rightarrow GH \in F^+$?

5.8. Cho $F = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A\}$.

Chứng minh rằng $AB \rightarrow E \in F^+$ và $AB \rightarrow G \in F^+$?

5.9. Cho $F = \{AB \rightarrow E, AG \rightarrow I, BE \rightarrow I, E \rightarrow G, GI \rightarrow H\}$

Chứng minh rằng $AB \rightarrow GH \in F^+$.

5.10. Cho $F = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A\}$

Chứng minh rằng $AB \rightarrow E \in F^+$ và $AB \rightarrow G \in F^+$.

5.11. Cho $\Omega = \{A, B, C, D, E, G\}$ và $F = \{AB \rightarrow C, D \rightarrow EG, C \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG\}$.

Tính X^+ với $X = \{AB\}, \{BC\}, \{CD\}, \{ACD\}, \{DEG\}$.

5.12. Cho $F = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A\}$.

Tính bao đóng của X với $X = \{AB\}$

Chứng minh rằng $AB \rightarrow E \in F^+$.

5.13. Cho biết dạng chuẩn của các lược đồ quan hệ sau. Sau đó đưa về các dạng chuẩn cao hơn (nếu có thể)

a. $Q = ABCDEG$

$F = \{A \rightarrow BC, C \rightarrow DE, E \rightarrow G\}$

b. $Q = ABCDEGH$

$F = \{C \rightarrow AB, D \rightarrow E, B \rightarrow G\}$

c. $Q = ABCDEGH$

$F = \{A \rightarrow BC, D \rightarrow E, H \rightarrow G\}$

d. $Q = ABCDEG$

$F = \{AB \rightarrow C, C \rightarrow B, ABD \rightarrow E, G \rightarrow A\}$

e. $Q = ABCDEGHI$

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

TÀI LIỆU THAM KHẢO

- [1] Hồ Thuần - Hồ Cẩm Hà, *Các hệ cơ sở dữ liệu - Lý thuyết và thực hành*, Nhà xuất bản giáo dục, 2004
- [2] Lê Tiến Vương, *Nhập môn cơ sở dữ liệu quan hệ*, Nhà xuất bản thống kê, 2005
- [3] Lê Văn Sơn, *Hệ tin học phân tán*, Nhà xuất bản quốc gia TP. Hồ Chí Minh, 2002
- [4] Nguyễn Kim Anh, *Nguyên lý các hệ cơ sở dữ liệu*, Nhà xuất bản đại học quốc gia Hà Nội
- [5] Phương Lan, *Giáo trình nhập môn cơ sở dữ liệu*, Nhà xuất bản lao động xã hội
- [6] Toby J.Teorey - Trần Đức Quang biên dịch, *Mô hình hoá và thiết kế cơ sở dữ liệu*, Nhà xuất bản thống kê, 2002
- [7] Các tài liệu khác trên internet