

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

=====



GIÁO TRÌNH
HỆ CƠ SỞ DỮ LIỆU
(Lưu hành nội bộ)

Đà Nẵng, 06/2008

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

=====

GIÁO TRÌNH
HỆ CƠ SỞ DỮ LIỆU
(Lưu hành nội bộ)

Chủ biên: ThS.Phí Đắc Hải

Biên soạn: ThS.Trịnh Thị Ngọc Linh

Đà Nẵng, 06/2008

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 thế hệ thứ nhất của hệ quản trị cơ sở dữ liệu. Tiếp theo là thế 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 thế 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 “Hệ 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 ngôn ngữ cơ sở dữ liệu.
- Các mô hình dữ liệu, đặc biệt là mô hình quan hệ và mô hình thực thể - kết hợp.
- Quy hoạch, thiết kế và quản trị cơ sở dữ liệu.
- Chuẩn hoá cơ sở dữ liệu.
- Tính toàn vẹn và bảo mật cơ sở dữ liệu.
- Quản lý các giao dịch, điều khiển tương tranh và khôi phục cơ sở dữ liệu.
- Hệ cơ sở dữ liệu phân tán

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, như thiết kế cơ sở dữ liệu, nghiên cứu cơ sở dữ liệu hướng đối tượng, cơ sở dữ liệu di động, cơ sở dữ liệu quan hệ và SQL, kho dữ liệu và khai phá dữ liệu, cơ sở dữ liệu XML...

Trong quá trình biên soạn giáo trình, tôi đã tham khảo các tài liệu hiện đang được giảng dạy tại các trường đại học và cao đẳng trong và ngoài nước. Mỗi chương đều có phần tóm tắt, các câu hỏi và bài tập giúp sinh viên củng cố lại kiến thức.

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 và TS.Suh Dong Rin – giáo sư Trường Cao đẳng Dongseoul – Hàn Quốc, đã giúp tôi hoàn thành giáo trình này.

Đà Nẵng, tháng 06 năm 2008

Tác giả

MỤC LỤC

THUẬT NGỮ VÀ TỪ VIẾT TẮT

CHƯƠNG 1 GIỚI THIỆU VỀ CƠ SỞ DỮ LIỆU	1
1.1. Hệ xử lý tập tin truyền thống	1
1.1.1. Giới thiệu.....	1
1.1.2. Hạn chế của hệ xử lý tập tin truyền thống	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
1.2.3. Hệ cơ sở dữ liệu.....	6
1.2.3.1. Định nghĩa và phân loại.....	6
1.2.3.2. Sự cần thiết của các hệ cơ sở dữ liệu	9
1.3. Đối tượng sử dụng.....	10
1.4. Ưu và nhược điểm của hệ CSDL	11
1.4.1. Ưu điểm.....	11
1.4.2. Nhược điểm	11
CHƯƠNG 2 MÔI TRƯỜNG CƠ SỞ DỮ LIỆU	14
2.1. Kiến trúc ANSI-PARC 3-mức	14
2.1.1. Mức vật lý (mức trong).....	14
2.1.2. Mức logic (mức khái niệm).....	14
2.1.3. Mức khung nhìn (mức ngoài).....	15
2.1.4. Lược đồ và thể hiện của CSDL	17
2.1.5. Tính độc lập dữ liệu	18
2.2. Các ngôn ngữ cơ sở dữ liệu	19
2.2.1. Ngôn ngữ định nghĩa dữ liệu.....	19
2.2.2. Ngôn ngữ thao tác dữ liệu	19
2.3. Mô hình dữ liệu.....	19
2.3.1. Mô hình thực thể liên kết	20
2.3.2. Mô hình hướng đối tượng	21
2.3.3. Mô hình mạng.....	22

2.3.4. Mô hình phân cấp.....	22
2.3.5. Mô hình quan hệ.....	23
2.3.6. Lựa chọn giữa các mô hình dữ liệu.....	24
2.4. Từ điển dữ liệu	24
CHƯƠNG 3 MÔ HÌNH QUAN HỆ	29
3.1. Mở đầu	29
3.2. Các khái niệm của mô hình quan hệ.....	29
3.2.1. Miền (Domain).....	29
3.2.2. Thuộc tính (Attribute)	29
3.2.3. Quan hệ (Relation)	30
3.2.4. Lược đồ quan hệ (Relation scheme)	31
3.2.5. Bộ (tuple).....	32
3.2.6. Thể hiện của quan hệ (view).....	32
3.2.7. Khóa (key)	32
3.3. Tính toàn vẹn của quan hệ	33
3.3.1. Ràng buộc toàn vẹn thực thể.....	34
3.3.2. Ràng buộc toàn vẹn tham chiếu	34
3.3.2. Ràng buộc toàn vẹn ngữ nghĩa.....	35
3.4. Các ngôn ngữ quan hệ.....	35
3.4.1. Đại số quan hệ.....	35
3.4.1.1. Phép chiếu	35
3.4.1.2. Phép chọn	36
3.4.1.3. Phép hợp.....	36
3.4.1.4. Phép hiệu	37
3.4.1.5. Phép giao	37
3.4.1.6. Tích Đề-các	38
3.4.1.7. Phép nối.....	38
3.4.1.8. Phép chia	39
3.4.2. Phép tính quan hệ.....	40
3.4.2.1. Phép tính quan hệ bộ (Codd 1970)	40
3.4.2.2. Phép tính quan hệ miền.....	43
3.5. Khung nhìn (View).....	43
3.5.1. Khái niệm quan hệ cơ sở và khung nhìn	43
3.5.2. Mục đích của khung nhìn	43

3.5.3. Vấn đề cập nhật qua khung nhìn	44
CHƯƠNG 4 MÔ HÌNH THỰC THỂ LIÊN KẾT.....	49
4.1. Các thành phần cơ bản.....	49
4.1.1. Thực thể.....	49
4.1.2. Liên kết.....	50
4.2. Các ràng buộc trên các kiểu liên kết	51
4.2.1. Ràng buộc về tỉ số lực lượng.....	52
4.2.1.1. Tỉ số lực lượng 1:1	52
4.2.1.2. Tỉ số lực lượng 1:N hay N:1	52
4.2.1.3. Tỉ số lực lượng N:M.....	53
4.2.2. Ràng buộc về sự tham gia	53
4.3. Lược đồ thực thể liên kết (lược đồ E-R)	54
4.3.1. Các thành phần của lược đồ	54
4.3.2. Phương pháp thiết kế	55
4.3.3. Chuyển đổi từ lược đồ E-R sang lược đồ quan hệ.....	55
4.4. Các tính chất mở rộng của mô hình E-R	58
4.4.1. Chuyên biệt hoá	58
4.4.2. Khái quát hoá.....	59
4.4.3. Phép gộp.....	60
4.5. Một ví dụ về thiết kế lược đồ E-R.....	62
4.5.1. Mô tả bài toán	62
4.5.2. Xác định các kiểu thực thể, các thuộc tính và kiểu liên kết.....	62
4.5.3. Lược đồ E-R	64
CHƯƠNG 5 HOẠCH ĐỊNH, THIẾT KẾ VÀ QUẢN TRỊ CƠ SỞ DỮ LIỆU	68
5.1. Các giai đoạn phát triển của hệ CSDL	68
5.1.1. Lập kế hoạch CSDL.....	69
5.1.2. Định nghĩa hệ thống.....	70
5.1.3. Tập hợp và phân tích các yêu cầu.....	72
5.1.4. Thiết kế CSDL.....	73
5.1.5. Chọn hệ quản trị CSDL (tùy chọn).....	73
5.1.6. Thiết kế ứng dụng	74
5.1.6.1. Thiết kế giao dịch (transaction)	74
5.1.6.2. Thiết kế giao diện người dùng	74
5.1.7. Mô hình hoá (tùy chọn).....	75

5.1.8. Thực hiện	75
5.1.9. Chuyển đổi dữ liệu (tùy chọn)	75
5.1.10. Kiểm thử	75
5.1.11. Vận hành và bảo dưỡng	75
5.2. Tổng quan về thiết kế cơ sở dữ liệu	75
5.2.1. Thiết kế CSDL mức khái niệm	76
5.2.2. Thiết kế CSDL mức logic	77
5.2.3. Thiết kế CSDL mức vật lý	77
5.3. Quản trị dữ liệu và quản trị CSDL	77
5.3.1. Người quản trị dữ liệu	77
5.3.2. Người quản trị CSDL	77
CHƯƠNG 6 CHUẨN HOÁ	81
6.1. Mục đích của việc chuẩn hoá	81
6.2. Dư thừa thông tin và cập nhật dị thường	81
6.2.1. Sự dư thừa dữ liệu	81
6.2.2. Các dị thường cập nhật dữ liệu	81
6.3. Phụ thuộc hàm	83
6.3.1. Định nghĩa phụ thuộc hàm	83
6.3.2. Các qui tắc phụ thuộc hàm	84
6.3.2.1. Hệ tiên đề Armstrong cho các phụ thuộc hàm	84
6.3.2.2. Các tính chất của phụ thuộc hàm	84
6.4. Suy diễn lô-gíc	85
6.5. Bao đóng của tập phụ thuộc hàm	85
6.5.1. Định nghĩa	85
6.5.2. Phụ thuộc hàm đầy đủ	86
6.5.3. Phụ thuộc hàm dư thừa	86
6.5.4. Phủ của tập phụ thuộc hàm, phụ thuộc hàm tương đương	87
6.6. Bao đóng của tập thuộc tính	87
6.6.1. Định nghĩa	87
6.6.2. Thuật toán tìm bao đóng	87
6.8. Phép tách lược đồ quan hệ	88
6.8.1. Định nghĩa	88
6.8.2. Phép tách bảo toàn thông tin	88
6.8.3. Thuật toán kiểm tra phép tách bảo toàn thông tin	89

6.9. Qui trình chuẩn hoá	89
6.9.1. Dạng chuẩn một (1NF)	90
6.9.2. Dạng chuẩn thứ 2 (2NF)	91
6.9.3. Dạng chuẩn thứ 3 (3NF)	92
6.9.3.1. Định nghĩa.....	92
6.9.3.2. Thuật toán đưa về dạng chuẩn 3 bảo toàn thông tin	93
6.9.4. Dạng chuẩn Boyce-Codd (BCNF).....	94
6.9.4.1. Định nghĩa.....	94
6.9.4.2. Thuật toán đưa về dạng chuẩn Boyce-Codd bảo toàn thông tin.....	94
CHƯƠNG 7 AN TOÀN VÀ TOÀN VỆ DỮ LIỆU	100
7.1. An toàn dữ liệu.....	100
7.1.1. Xuất trình căn cước và xác minh người truy cập	101
7.1.2. Kiểm tra truy cập	102
7.1.3. Sử dụng các khung nhìn.....	102
7.1.4. Các lệnh an toàn dữ liệu.....	102
7.2. Toàn vẹn dữ liệu.....	104
7.2.1. Toàn vẹn thực thể	104
7.2.2. Toàn vẹn tham chiếu.....	104
7.2.3. Các ràng buộc miền	106
7.2.4. Thao tác bất	107
CHƯƠNG 8 QUẢN LÝ GIAO DỊCH.....	110
8.1. Hỗ trợ giao dịch.....	110
8.1.1. Giao dịch	110
8.1.2. Các tính chất của giao dịch	112
8.2. Điều khiển tương tranh.....	113
8.2.1. Mục đích của việc điều khiển tương tranh.....	113
8.2.2. Tính khả tuần tự của các lịch và việc sử dụng chúng.....	114
8.2.3. Các kỹ thuật điều khiển tương tranh.....	122
8.2.3.1. Phương pháp khóa.....	122
8.2.3.2. Phương pháp nhãn thời gian	126
8.3. Khôi phục CSDL.....	129
8.3.1. Sự cần thiết phải có dịch vụ khôi phục CSDL	129
8.3.2. Giao dịch và sự khôi phục.....	130
8.3.3. Phương tiện khôi phục	130

8.3.4. Một số kỹ thuật khôi phục	130
8.3.4.1. Khôi phục dựa vào nhật ký (file log) của giao dịch	130
8.3.4.2. Kỹ thuật khôi phục theo kiểu cập nhật trì hoãn.....	131
8.3.4.3. Kỹ thuật khôi phục theo kiểu cập nhật ngay	131
8.3.4.4. Kỹ thuật khôi phục sử dụng trang ghi vết.....	131
CHƯƠNG 9 HỆ CƠ SỞ DỮ LIỆU PHÂN TÁN	135
9.1. Khái niệm và kiến trúc.....	135
9.1.1. Khái niệm.....	135
9.1.2. Kiến trúc của một hệ CSDL phân tán	137
9.1.2.1. CSDL từ xa.....	139
9.1.2.2. CSDL liên hiệp	139
9.1.2.3. CSDL song song.....	139
9.2. Các ưu và nhược điểm của việc phân tán dữ liệu.....	139
9.2.1. Các ưu điểm của việc phân tán dữ liệu	139
9.2.2. Các nhược điểm của việc phân tán dữ liệu.....	140
9.3. Quản lý giao dịch phân tán.....	141
9.3.1. Điều khiển tương tranh.....	141
9.3.1.1. Thời dấu (time stamp).....	141
9.3.1.2. Sự đặt khóa (locking).....	141
9.3.2. Sự hợp thức hoá giao dịch	143
TÀI LIỆU THAM KHẢO.....	148

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

CHƯƠNG 1

GIỚI THIỆU 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), 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

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.

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*.

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. Chẳng hạn, hãy xét Ví dụ 1.1 sau đây.

Ví dụ 1.1:

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 trưởng 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)

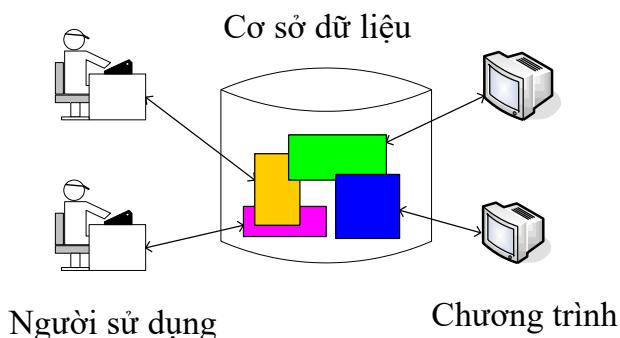
Dữ liệu là các sự kiện, văn bản, hình ảnh, âm thanh, phim ảnh... có ý nghĩa được ghi lại và lưu trữ trên máy tính.

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.

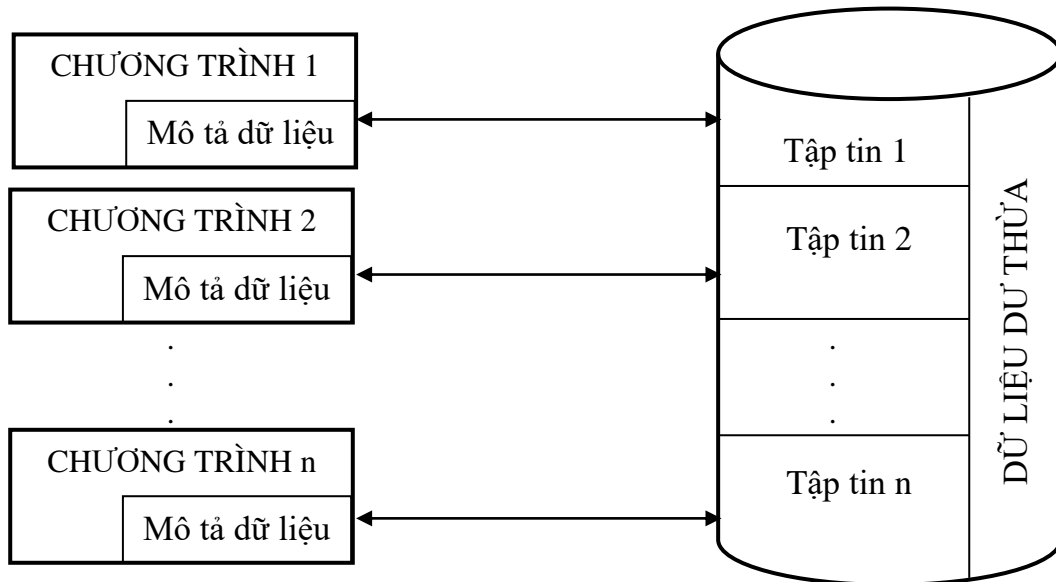


Hình 1.1. Sơ đồ tổng quát về một CSDL

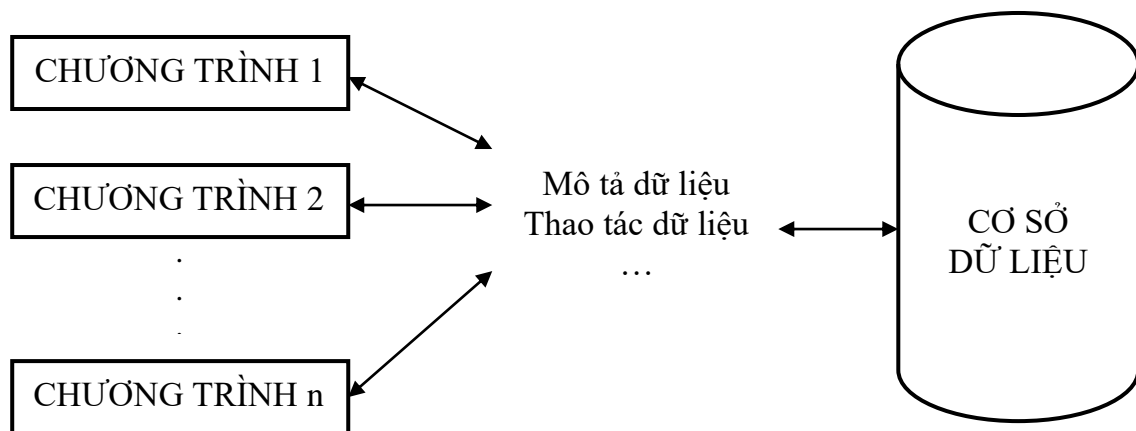
Trên Hình 1.1, chúng ta nhìn thấy các thành phần CSDL hợp nhất là một bộ sưu tập các dữ liệu chứa trên các phương tiện lưu trữ (như đĩa từ hay băng từ), những NSD trực tuyến hay các chương trình ứng dụng có thể sử dụng (tìm kiếm, sửa đổi, bổ sung, loại bỏ thông tin...), chia sẻ CSDL này.

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 dấu 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.

1.2.3. 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.

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

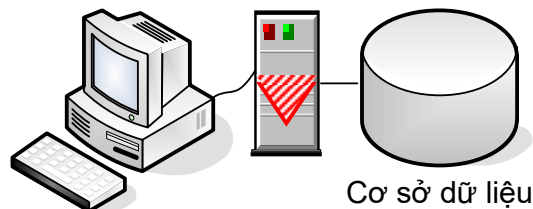
a. Các hệ CSDL tập trung

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.

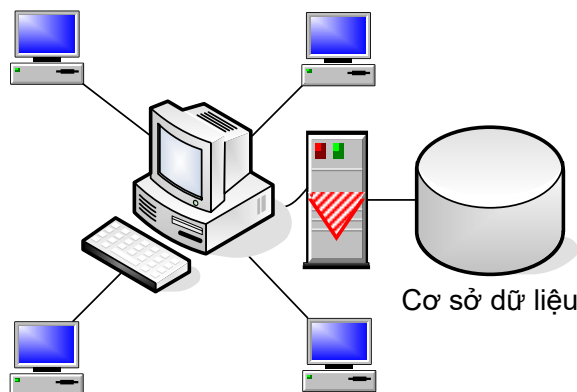
Có 3 kiểu hệ CSDL tập trung: hệ CSDL cá nhân, hệ CSDL trung tâm, hệ CSDL khách/chủ.

- **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.



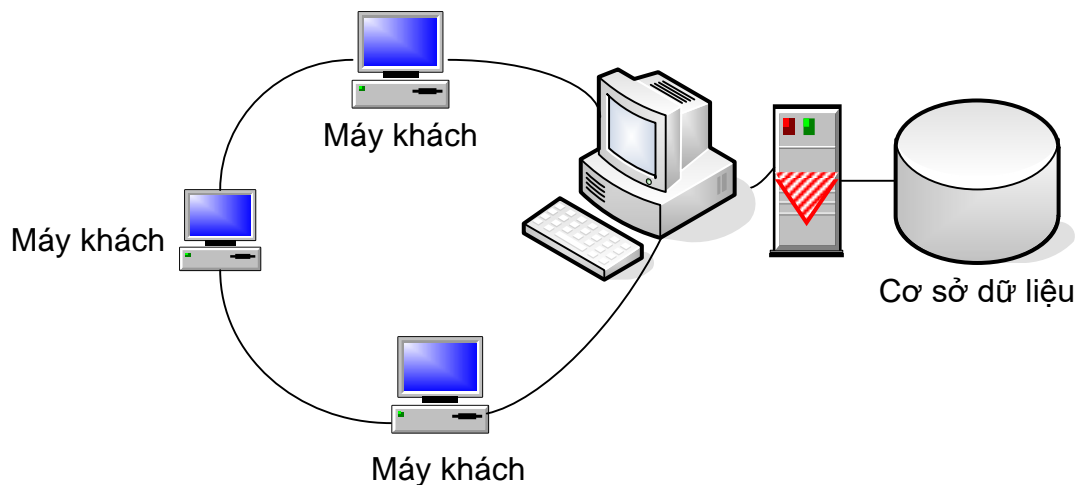
Hình 1.4. Hệ cơ sở dữ liệu cá nhân

- **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 làm 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 1.5. 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. Máy chủ cung cấp các dịch vụ quản lý tập tin hay CSDL, quản lý truyền thông...mà máy khách yêu cầu.



Hình 1.6. 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 1.7).

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:

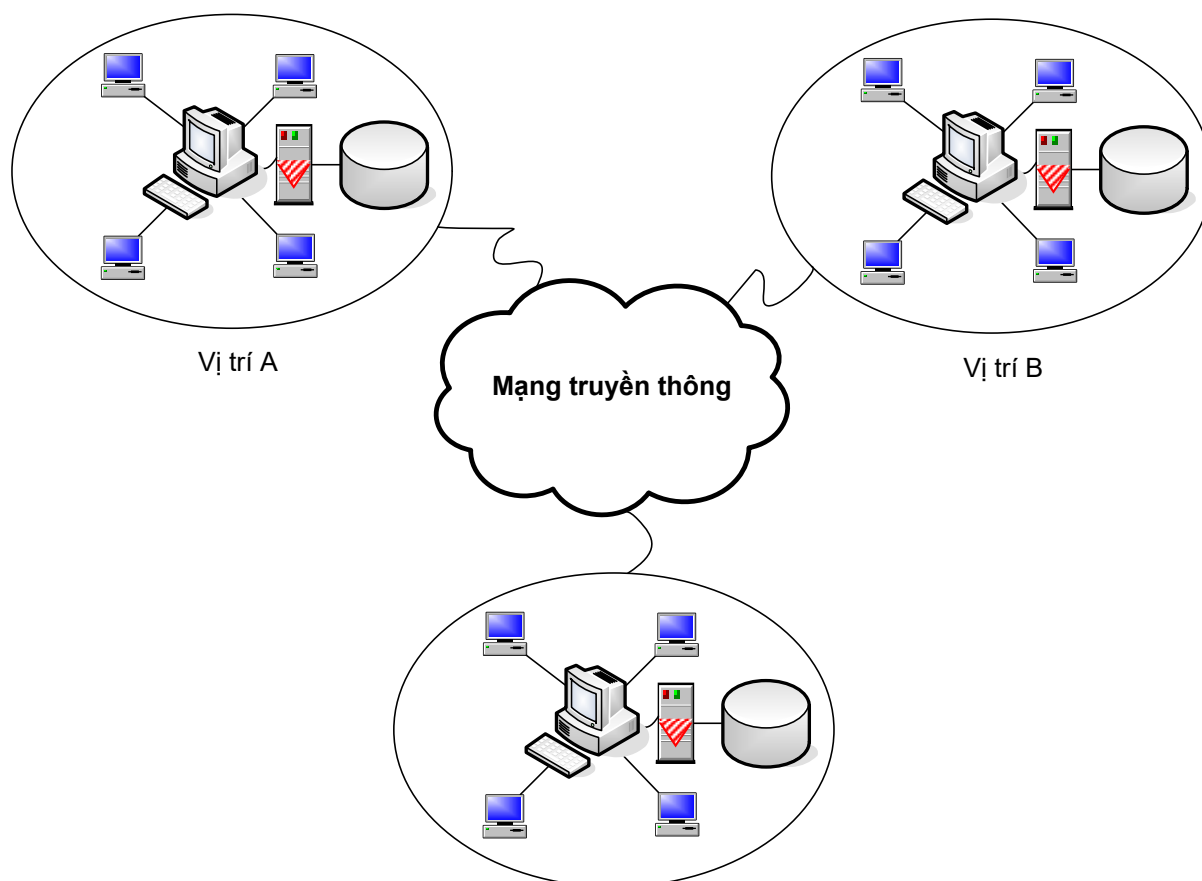
- + Các hệ điều hành máy tính tại mỗi vị trí địa lý là như nhau hay ít nhất chúng có khả năng tương thích cao.
- + Các mô hình dữ liệu được sử dụng tại mỗi một vị trí địa lý là như nhau (mô hình quan hệ được sử dụng chung nhất đối với các hệ CSDL phân tán ngày nay)
- + Các hệ quản trị CSDL được sử dụng 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 điển hình 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 1.7. 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.

TỔNG KẾT CHƯƠNG 1

Chương 1 đã giới thiệu về hệ xử lý tập tin truyền thống và các hạn chế của nó, trên cơ sở đó nêu lên sự cần thiết của hệ CSDL. Chương 1 cũng đưa ra các khái niệm chính và các ưu nhược điểm của hệ CSDL. Sau đây là tóm tắt một số nội dung chính của chương 1:

- Hạn chế của hệ xử lý tập tin truyền thống là sự cô lập dữ liệu, dư thừa dữ liệu và tính bất nhất, khó khăn trong việc truy xuất dữ liệu, các vấn đề về tính nguyên tử, tính toàn vẹn, tính bất thường trong truy xuất cạnh tranh, vấn đề an toàn.

- 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.

- 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 CSDL.

- Một hệ CSDL là một hệ thống gồm có bốn thành phần là CSDL hợp nhất, những NSD, phần mềm hệ quản trị CSDL, phần cứng.

- Hệ CSDL bao gồm hai loại kiến trúc là hệ CSDL tập trung và hệ CSDL phân tán. Hệ CSDL tập trung có ba dạng là hệ CSDL cá nhân, hệ CSDL trung tâm, hệ CSDL khách chủ. Hệ CSDL phân tán gồm hai dạng là hệ CSDL thuần nhất và hệ CSDL không thuần nhất.

- Ưu điểm nổi bật của CSDL là giảm sự trùng lặp thông tin, 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, dễ tích hợp, tiết kiệm không gian lưu trữ, có thể truy xuất dữ liệu theo nhiều cách khác nhau, 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.

- Nhược điểm của việc sử dụng CSDL là 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 2.1: 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 2.2: 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 2.3:

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 ...

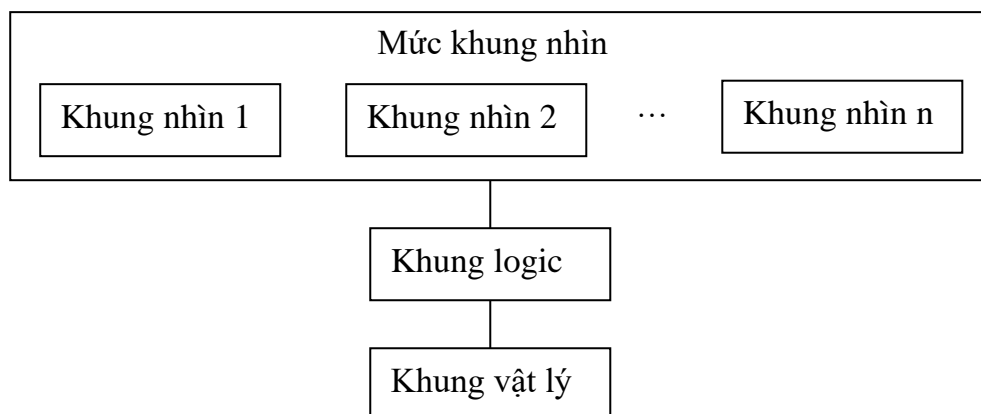
Như vậy, ở mức khung nhìn:

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 thuận lợi 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 hoá hay bảng băm).

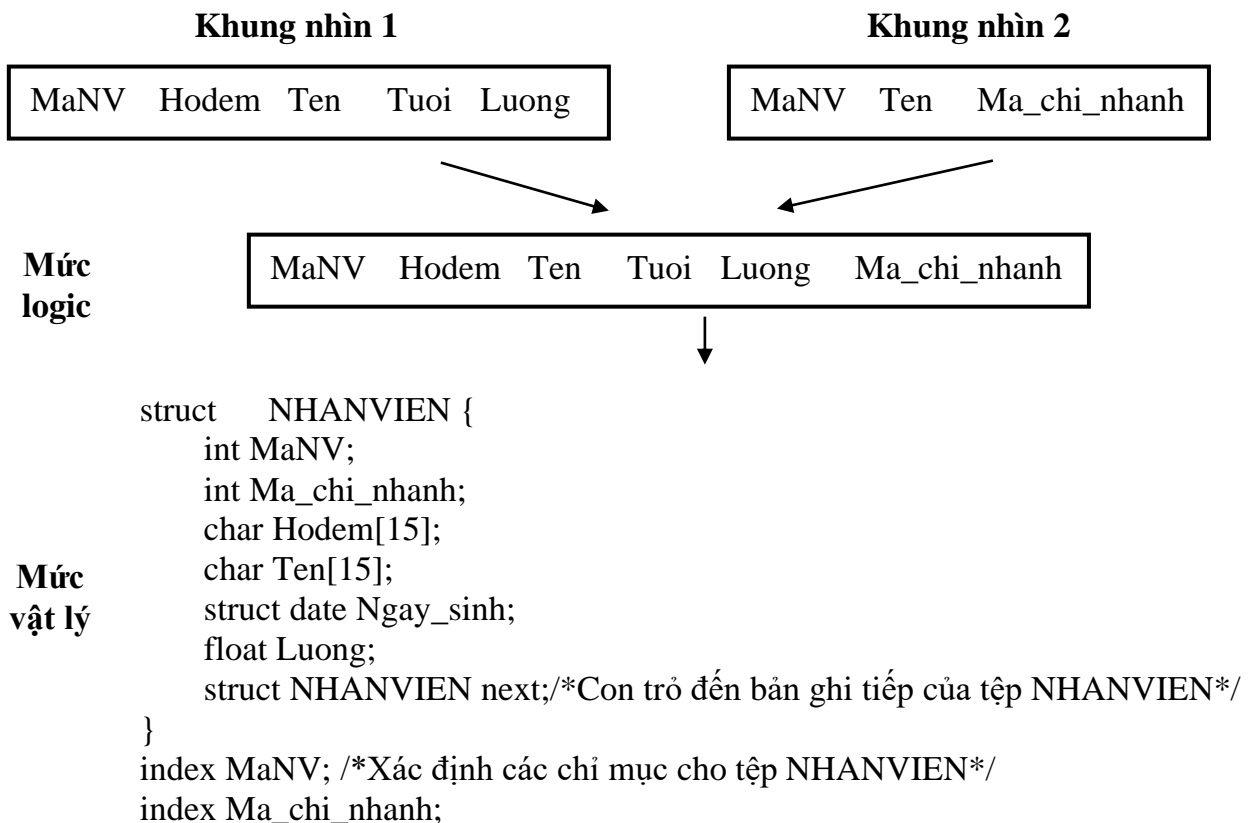
- Người quản trị CSDL có thể thay đổi cấu trúc lưu trữ của CSDL mà không làm ảnh hưởng đến những khung nhìn của NSD.
- Những thay đổi về khía cạnh vật lý trong lưu trữ, chẳng hạn như thay một thiết bị nhỏ thứ cấp mới, có thể không làm ảnh hưởng đến cấu trúc bên trong của CSDL.
- Người quản trị CSDL có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

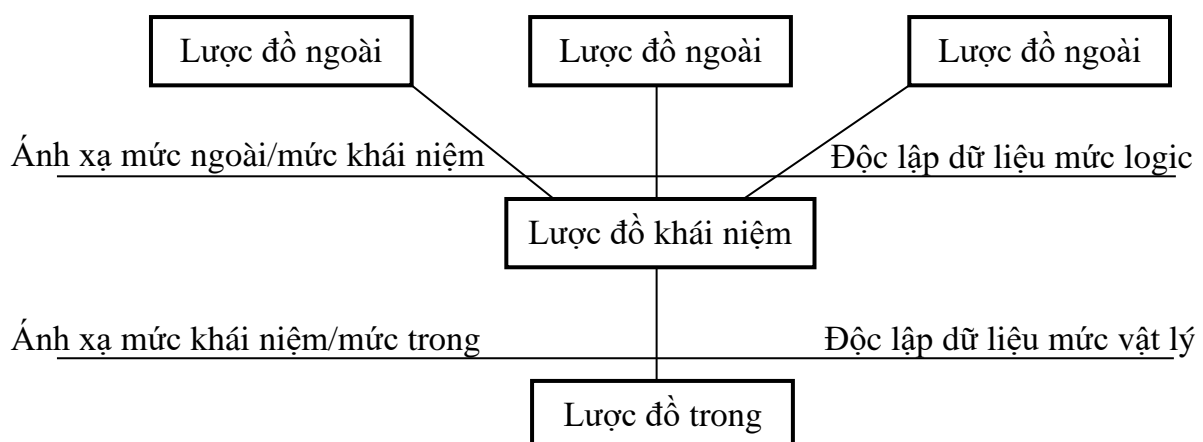
Như vậy nhiều thể hiện của CSDL có thể tương ứng với cùng một lược đồ CSDL. Đôi khi, lược đồ còn được gọi là nội hàm của CSDL và một thể hiện còn được gọi là một mở rộng hay một trạng thái của CSDL.

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. Các ngôn ngữ cơ sở dữ liệu

Một hệ CSDL cung cấp hai kiểu ngôn ngữ khác nhau:

- Một ngôn ngữ đặc tả sơ đồ dữ liệu, gọi là ngôn ngữ định nghĩa dữ liệu.
- Một ngôn ngữ biểu diễn các truy vấn và cập nhật CSDL, gọi là ngôn ngữ thao tác dữ liệu.

2.2.1. Ngôn ngữ định nghĩa dữ liệu (Data Definition Language - DDL)

Ngôn ngữ định nghĩa dữ liệu cho phép khai báo, hiệu chỉnh cấu trúc CSDL, mô tả các mối quan hệ của dữ liệu, các quy tắc áp đặt lên dữ liệu.

Thông thường, người ta sử dụng ngôn ngữ định nghĩa dữ liệu để thực hiện tạo cơ sở dữ liệu, tạo bảng, xoá bảng...

Kết quả biên dịch các lệnh của DDL là tập hợp các bảng được lưu trữ trong một tập tin đặc biệt được gọi *từ điển dữ liệu* hay *thư mục dữ liệu*. Từ điển dữ liệu là một tập tin được tra cứu trước khi dữ liệu hiện hành được đọc hay sửa đổi.

2.2.2. Ngôn ngữ thao tác dữ liệu (Data Manipulation Language - DML)

Ngôn ngữ thao tác dữ liệu cho phép người dùng thực hiện các thao tác trên dữ liệu như tìm kiếm, chèn, sửa đổi, xoá bỏ thông tin.

Có hai kiểu ngôn ngữ thao tác dữ liệu:

- **DML thủ tục (procedural DML):** Yêu cầu NSD phải xác định dữ liệu nào họ đang cần và xác định cách thức để có được dữ liệu đó.
- **DML phi thủ tục (Nonprocedural DML):** Chỉ yêu cầu NSD xác định dữ liệu nào họ đang cần, chứ không yêu cầu NSD xác định cách thức để có dữ liệu đó. Ngôn ngữ thông dụng nhất của kiểu DML phi thủ tục là ngôn ngữ SQL (Structured Query Language).

2.3. 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.

Mô hình dữ liệu được chia thành các nhóm sau:

- Mô hình logic trên cơ sở đối tượng: Gồm mô hình thực thể kết hợp, mô hình hướng đối tượng, mô hình ngữ nghĩa, mô hình dữ liệu chức năng.
- Mô hình logic trên cơ sở bản ghi: Gồm mô hình quan hệ, mô hình mạng, mô hình phân cấp.

Sau đây là một số dạng cơ bản của mô hình dữ liệu.

2.3.1. Mô hình thực thể liên kết

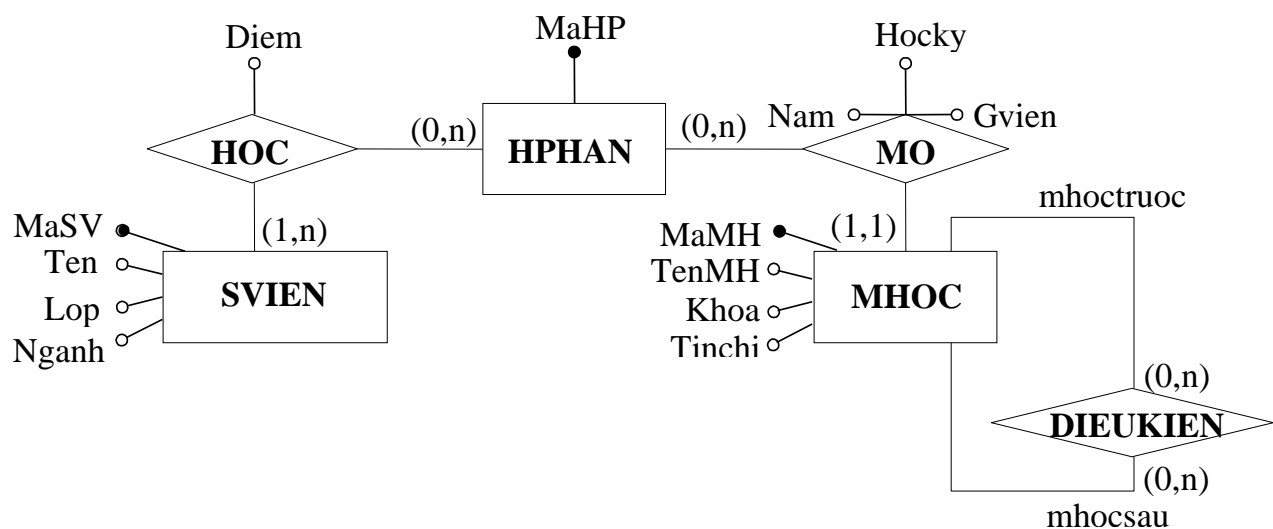
Mô hình thực thể liên kết được xây dựng dựa trên nhận thức rằng thế giới thực mà chúng ta muốn phản ánh là một tập hợp các đối tượng cơ sở và các mối liên kết giữa chúng.

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

Thực thể có thể là một vật cụ thể (như một cá nhân, một quyển sách, một mặt hàng...), có thể là một vật trừu tượng (như một tổ chức, một dự án, một môn học...). Một mối liên kết thể hiện một liên kết giữa nhiều thực thể. Chẳng hạn, mối liên kết “*người mượn*” liên kết một người với một quyển sách người đó mượn.

Tập tất cả các thực thể thuộc cùng một kiểu được gọi là *tập thực thể*. Tập tất cả các mối liên kết thuộc cùng một kiểu được gọi là *tập mối liên kết*. Thực thể hay liên kết được mô tả bởi một tập các *thuộc tính* của nó.

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 mối liên kết.



Hình 2.4. Ví dụ mô hình thực thể liên kết

Hình 2.4 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), Nganh (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).

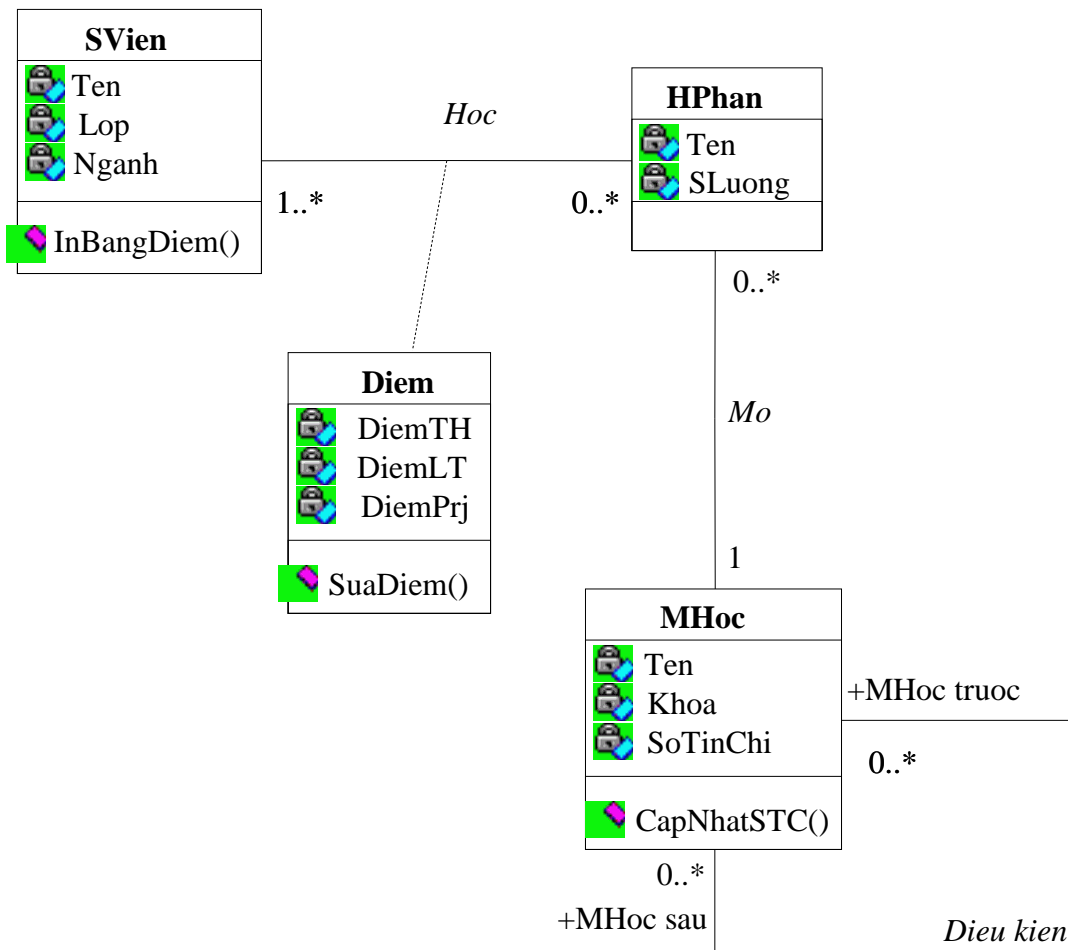
- Giữa thực thể SVIEN và HPHAN có liên kết HOC (học), liên kết này có thuộc tính Diem (điểm của sinh viên khi học học phần đó).
- Giữa thực thể HPHAN và MHOC có liên kết MO (mở), liên kết này có các thuộc tính Nam (năm), HKy (học kỳ), Gvien (giáo viên).
- Ngoài ra thực thể MHOC còn có liên kết đệ quy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

(Xem thêm chương 4 – Mô hình thực thể liên kết)

2.3.2. 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 2.5. Ví dụ mô hình CSDL hướng đối tượng

Hình 2.5 bao gồm:

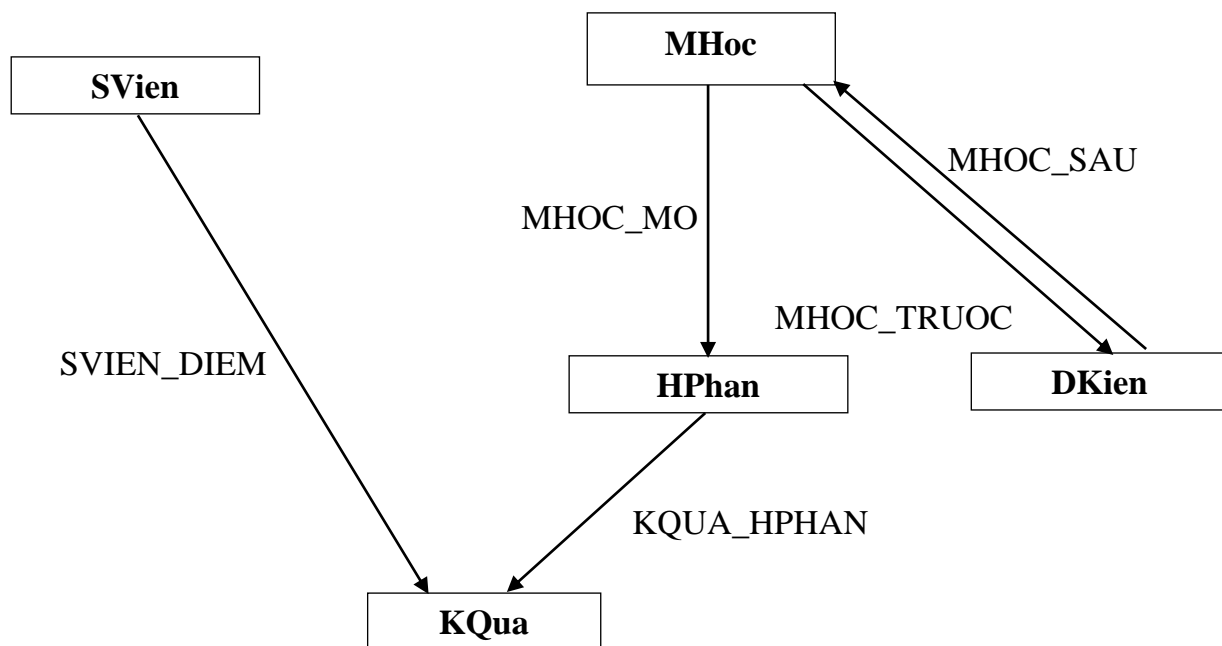
- Đối tượng SVien (sinh viên): gồm các biến Ten (tên sinh viên), Lop (lớp), Nganh (ngành) và các phương thức LapTKB (lập thời khóa biểu), InBangDiem (in bảng điểm).
- Đối tượng HPhan (học phần) gồm các biến Ten (tên học phần), SLuong (số lượng).
- Đối tượng Diem (điểm) gồm các biến DiemTH (điểm thực 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).

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.

2.3.3. Mô hình 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).

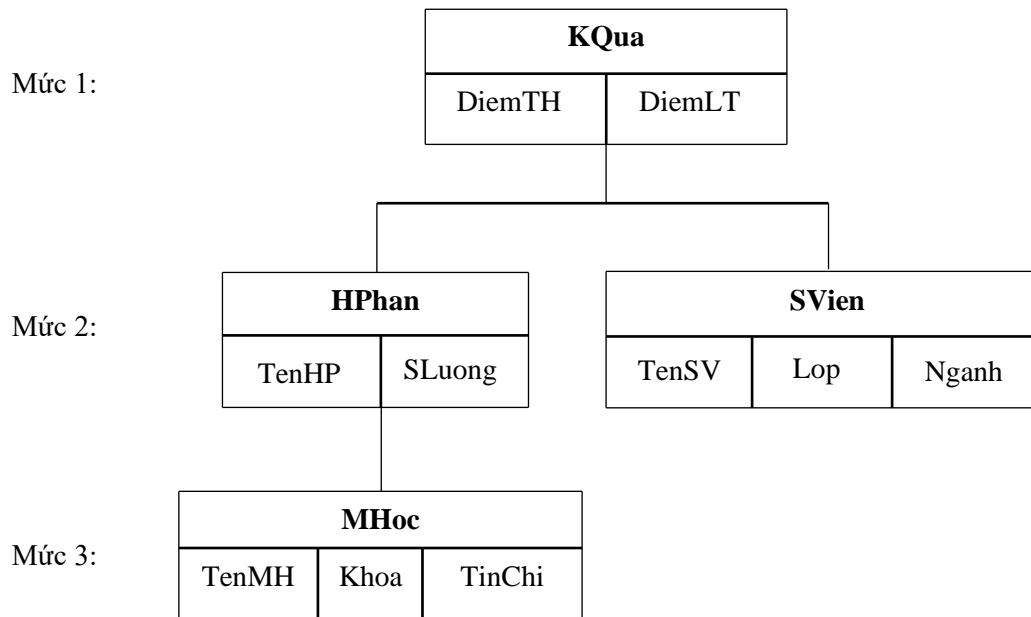
Xuất phát từ một đối tượng (biểu diễn bằng một bản ghi) có thể có nhiều mối quan hệ đến những đối tượng khác nhau. Trong những quan hệ đó luôn phân biệt đối tượng là chủ của quan hệ và những đối tượng thành phần của quan hệ.



Hình 2.6. Ví dụ mô hình CSDL mạng

2.3.4. Mô hình phân cấp

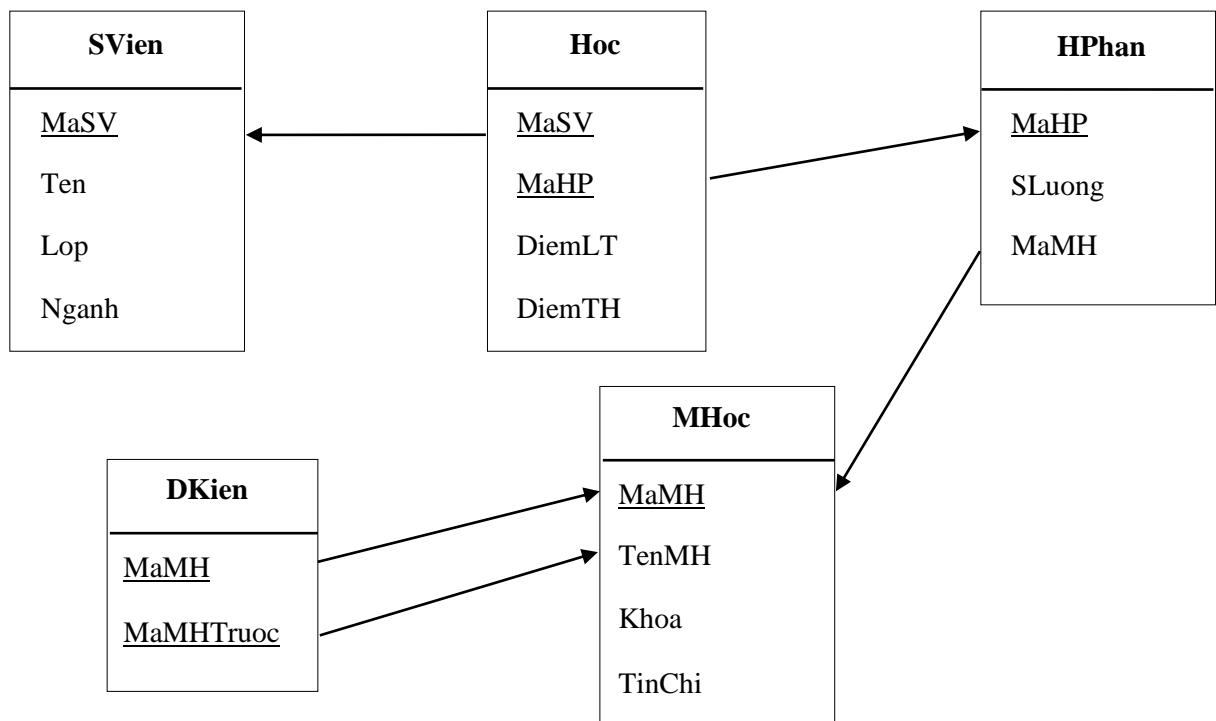
Mô hình phân cấp tương tự như mô hình mạng ở chỗ dữ liệu biểu diễn bằng tập các bản ghi, và mối quan hệ giữa dữ liệu biểu diễn bằng các mối nối như các con trỏ. Nhưng khác với mô hình mạng, mối quan hệ giữa hai đối tượng trong mô hình phân cấp thể hiện theo kiểu cha-con, và sơ đồ các bản ghi cùng các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



Hình 2.7. Ví dụ mô hình CSDL phân cấp

2.3.5. 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*) và các *dòng* (thường gọi là *bản ghi* hay *bộ*). Mỗi cột có một tên duy nhất.



Hình 2.8. Ví dụ mô hình CSDL quan hệ

Hình 2.8 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

<div>Dòng</div> <div></div>		Cột			
		MaSV	Ten	Lop	Nganh
		T001	Trần Thị Lan	03T1	CNTT
		T002	Nguyễn Văn Dũng	03T1	CNTT
		T003	Lý Hải	03T1	CNTT

(Xem thêm ở phần chương 3 - Mô hình quan hệ).

2.3.6. 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.

2.4. Từ điển dữ liệu

Trong một hệ quản trị CSDL, một từ điển dữ liệu là một tập hợp chỉ đọc (read only) của các bảng (table) và khung nhìn (view).

Thông thường một hệ thống từ điển dữ liệu chứa các thông tin:

- Tên, kiểu, kích thước các bản ghi.
- Tên của các mối quan hệ, các ràng buộc toàn vẹn trên dữ liệu.
- Tên những người có quyền truy cập vào CSDL.
- Các lược đồ trong, lược đồ khái niệm, lược đồ ngoài và các ánh xạ giữa chúng.

Những người đã làm quen với hệ quản trị CSDL Microsoft Access có thể thấy các từ điển dữ liệu này thông qua các bảng (Table) có tên bắt đầu bằng chữ MSys như MSysACEs, MSysColumn, MSysIMEXColumn, MSysIMEXSpecs, MSysIndexes, MSysMacros, MSysObjects, MSysQueries, MSysRelationships... Tuy nhiên các tập tin này thường bị ẩn nên chúng ta có thể thấy khi chọn chức năng **Tools / Options / View tabs / System Objects**.

TỔNG KẾT CHƯƠNG 2

Chương 2 trình bày một số nguyên lý của hệ CSDL, giúp hiểu rõ hơn về hệ CSDL. Sau đây là tóm tắt một số nội dung chính của chương 2 :

- Kiến trúc ANSI-PARC 3 mức gồm có 3 mức là mức vật lý (mức trong), mức logic (mức khái niệm) và mức khung nhìn (mức ngoài).
- Mức vật lý là mức truy xuất thấp nhất này mô tả dữ liệu được thực sự lưu trữ như thế nào trong CSDL.
- Mức logic là mức mô tả những dữ liệu nào được lưu trữ trong CSDL và có những mối quan hệ nào giữa chúng.
- Mức khung nhìn là mức của NSD và các chương trình ứng dụng.
- Các ngôn ngữ CSDL gồm có ngôn ngữ định nghĩa dữ liệu (DDL), ngôn ngữ thao tác dữ liệu (DML)
- Ngôn ngữ định nghĩa dữ liệu (Data Definition Language - DDL) cho phép khai báo, hiệu chỉnh cấu trúc CSDL, mô tả các mối quan hệ của dữ liệu cũng như các quy tắc áp đặt lên dữ liệu.
- Ngôn ngữ thao tác dữ liệu (Data Manipulation Language - DML) là ngôn ngữ cho phép NSD truy xuất hoặc thao tác dữ liệu.
- Có hai kiểu ngôn ngữ thao tác dữ liệu: DML thủ tục (Procedural DML) yêu cầu NSD đặc tả dữ liệu nào cần và làm thế nào để nhận được nó, DML phi thủ tục (Nonprocedural DML) yêu cầu NSD đặc tả dữ liệu nào cần nhưng không cần đặc tả làm thế nào để nhận được nó.
- Các mô hình dữ liệu cơ bản gồm có: mô hình thực thể liên kết, mô hình hướng đối tượng, mô hình mạng, mô hình phân cấp, mô hình quan hệ.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 2

- 2.1. Nêu rõ các mức của kiến trúc ANSI-PARC 3 mức.
- 2.2. Nêu mục đích của các ngôn ngữ DDL, DML.
- 2.3. Nêu các mô hình dữ liệu chính.
- 2.4. Hãy mô tả các chức năng và các thành phần của DBMS.
- 2.5. 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.
- 2.6. Từ điển dữ liệu được sử dụng để tổ chức các thông tin nào?
- 2.7. 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.

- 2.8. 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 chữ

c. Thông tin về phiếu đặt báo gồm

- Ngày khách đặt báo, chí
- Mã hiệu khách hàng đặt mua báo
- Mã báo khách đặt mua
- Số lượng báo, chí 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

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 3.1:

- 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 3.2:

- 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 3.3:

- 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 3.4:

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 3.5:

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 3.6: 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 3.7: Đây là 4 bộ dựa trên các thuộc tính của quan hệ HocVien:

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

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

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

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

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 3.8: 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 3.9: Với lược đồ ở ví dụ 3.6:

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 3.10: Với lược đồ ở ví dụ 3.6:

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

Ràng buộc toàn vẹn (Integrity Constraint/Rule) và kiểm tra sự vi phạm ràng buộc toàn vẹn là hai trong những vấn đề rất quan trọng trong quá trình phân tích, thiết kế và khai thác CSDL. Trong quá trình phân tích thiết kế CSDL, nếu không quan tâm đúng mức đến những vấn đề trên, thì có thể dẫn đến những hậu quả rất nghiêm trọng về tính an toàn và toàn vẹn dữ liệu, đặc biệt trong những CSDL tương đối 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.

3.3.1. 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 3.11: 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

3.3.2. 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 3.12:

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

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

3.3.2. 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 3.13:

Trong CSDL về quản lý sinh viên của một trường học (Ví dụ 3.6), chúng ta có một số ràng buộc toàn vẹn ngữ nghĩa như sau:

- R_1 : Mỗi lớp học phải thuộc một Khoa của trường.
- R_2 : Mỗi lớp học phải có một mã số duy nhất để phân biệt với mọi lớp học khác trong trường.
- R_3 : Mỗi sinh viên có một mã số riêng biệt, không trùng với bất cứ sinh viên nào khác.
- R_4 : Mỗi sinh viên phải đăng ký vào một lớp của trường.
- R_5 : Mỗi sinh viên được thi tối đa 2 lần cho mỗi môn học.
- R_6 : 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 3.14:

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 3.15:

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 3.16:

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 3.17: 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 3.18: 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	$r \cap s$
<u>A B C</u>	<u>A B C</u>	<u>A B 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 3.19:

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			$r \times s$					
A	B	C	D	E	F	A	B	C	D	E	F
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:

$$r \bowtie_F s$$

F

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 ta được giá trị đúng.

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

F

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:

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

F

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 3.20:

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 3.21: 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.

➤ 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 3.22:

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

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

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.

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 phần 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.

➤ Ngôn ngữ truy vấn SQL

SQL thuộc loại ngôn ngữ thể hệ thứ tư (4GL) được nghiên cứu nhiều năm và trở thành tiêu chuẩn quốc tế về kiểm soát dữ liệu. SQL kế thừa tính phi thủ tục của 4GL: Xử lý đồng thời hàng loạt câu lệnh. Người dùng chỉ cần nêu ra yêu cầu về dữ liệu mà không cần biết máy tính xử lý bên trong như thế nào. Người dùng có thể truy xuất nhanh chóng với những CSDL lớn, yêu cầu những xử lý phức tạp tinh vi mà không cần lập trình.

Sau đây là một số lệnh cơ bản của ngôn ngữ SQL.

a. Lệnh truy vấn dữ liệu

Một cách tổng quát khối select bao gồm ba mệnh đề chính:

Select: Xác định nội dung của các cột cần đưa ra kết quả

From: Xác định các bảng cần lấy thông tin ra

Where: Xác định các bản ghi thỏa mãn yêu cầu chọn lọc để đưa ra kết quả

Ngoài ra, để mở rộng khả năng của ngôn ngữ, khối select – from – where còn được bổ sung thêm các mệnh đề group by, having, order by.

Các lệnh truy vấn dữ liệu, gọi là câu vấn tin có cú pháp tổng quát như sau

SELECT [* | DISTINCT] *danhsach_chieu*

FROM *danhsach_ten_bang* | *ten_cac_view*

[WHERE *biểu_thức_điều_kiện*]

[GROUP BY *danhsach_ten_cot*]

[HAVING *biểu_thức_điều_kiện*]

[ORDER BY {*ten_cot* | *sot_thutur_cot* | *biểu_thức*} [ASC | DESC]]

Ví dụ 3.23:

Với các quan hệ như Ví dụ 3.22:

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

```
Select  EMP.ENAME
From    EMP, ASG, PROJ
Where   (EMP.ENO = ASG.ENO)
        AND (ASG.PNO = PROJ.PNO)
        AND (PROJ.PNAME = "CAD/CAM")
```

- Tìm tên tất cả nhân viên là lập trình viên (Programmer)

```
SELECT  ENAME
FROM    EMP
WHERE   TITLE = "Programmer"
```

b. Các lệnh cập nhật dữ liệu

Các lệnh cập nhật dữ liệu gồm có lệnh UPDATE (hiệu chỉnh), INSERT (thêm) và DELETE (xóa).

Hiệu chỉnh dữ liệu:

```
UPDATE [tên_bảng]
SET [tên_cột = biểu_thức,...]
[FROM tên_bảng]
[WHERE biểu_thức_điều_kiện]
```

Thêm dữ liệu:

```
INSERT INTO tên_bảng(danh_sách_tên_cột)
VALUES(các_giá_trị)
```

Xóa dữ liệu:

```
DELETE tên_bảng
[FROM {tên_bảng | tên_view}]
[WHERE biểu_thức_điều_kiện]
```

Ví dụ 3.24: Với các quan hệ như Ví dụ 3.22:

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

```
UPDATE  PAY
SET     SAL = 25000
WHERE   TITLE = "Programmer"
```

- Thêm nhân viên mới vào EMP

```
INSERT INTO EMP
VALUES ('E10', 'John Smith', 'Programmer')
```

- Xóa dự án có mã dự án là 'P1'

```
DELETE FROM PROJ
WHERE PNO = 'P1'
```

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 3.25: Với các quan hệ như Ví dụ 3.22, 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. Khung nhìn (View)

Trong kiến trúc 3 mức (theo ANSI-SPARC), 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.

3.5.1. 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.

3.5.2. 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.

3.5.3. 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.

TỔNG KẾT CHƯƠNG 3

Mô hình quan hệ được sử dụng khá phổ biến trong các hệ quản trị CSDL. Nội dung của chương đã trình bày những vấn đề rất quan trọng của mô hình quan hệ như các khái niệm và các ràng buộc toàn vẹn, ngôn ngữ quan hệ. Sau đây là tóm tắt một số nội dung chính của chương 3:

- Mô hình cơ sở dữ liệu quan hệ bao gồm: một hệ thống các ký hiệu để mô tả dữ liệu dưới dạng dòng và cột, một tập hợp các phép toán thao tác trên dữ liệu, ràng buộc toàn vẹn quan hệ.
- Các khái niệm trong mô hình quan hệ gồm miền, thuộc tính, quan hệ, lược đồ quan hệ, bộ, thể hiện của quan hệ, khóa.
- 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.
- 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.
- Một quan hệ trên một tập các thuộc tính là một tập con của tích đề-các của một hay nhiều miền.
- Tập tất cả các thuộc tính cần quản lý của một đối tượng cùng với tập các ràng buộc toàn vẹn giữa chúng được gọi là lược đồ quan hệ.
- Một bộ là các thông tin của một đối tượng thuộc quan hệ.
- Thể hiện (hoặc còn gọi là tình trạng) của quan hệ R, ký hiệu bởi T_R , là tập hợp các bộ giá trị của quan hệ R vào một thời điểm.
- Siêu khóa 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.
- Khóa 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 nhiều khóa thì các khóa này gọi là khóa dự tuyển.
- Một thuộc tính được gọi là khóa ngoại nếu nó là thuộc tính của một lược đồ quan hệ này nhưng lại là khóa chính của lược đồ quan hệ khác.
- 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.
- 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.
- Đại số quan hệ bao gồm phép chiếu, phép chọn, phép hợp, phép giao, phép hiệu, tích đề-các, phép nối, phép chia.
- Phép tính quan hệ bao gồm phép tính quan hệ bộ và phép tính quan hệ miền. Ngôn ngữ quan trọng thuộc phép tính quan hệ bộ là SQL.
- 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.

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)

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, Tinh)

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)

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 và SQL
 - a. Lập danh sách sinh viên gồm MaSV, HoTen
 - b. Lập danh sách sinh viên nữ khoa 'CNTT', danh sách cần MaSV, HoTen
 - c. Lập bảng điểm cho tất cả sinh viên khoa 'CNTT', bảng điểm gồm các cột MaSV, HoTen, TenMH, LanThi, DiemThi
 - d. Lập phiếu điểm cho sinh viên có MaSV='99001'
 - e. Lập danh sách sinh viên gồm MaSV, HoTen, TenLop, TenKhoa

f. Lập bảng điểm môn học có mã môn học là 'CSDL' cho tất cả sinh viên có mã lớp là 'CĐTH2B'

g. 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 và SQL:

1/ Thêm khách hàng mới: ss# = "A-12345", name = "David", address = "35 Chevre Road", license = "109283", year = "2002", model = "FORD LASER" vào CSDL.

2/ Xoá 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"

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 và SQL:

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

6/ Xoá tất cả các thông tin liên quan tới công ty "Bad Bank"

3.8. 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)

Tân 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)

Tân 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)

Tân 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)

Tân 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 thực hiện các câu hỏi sau bằng SQL:

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”

CHƯƠNG 4

MÔ HÌNH THỰC THỂ LIÊN KẾT

Chương này giới thiệu 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.

Nội dung của chương sẽ giới thiệu các thành phần của mô hình E-R, các vấn đề phát sinh trong thiết kế mô hình E-R, các ràng buộc trong mô hình, và cuối cùng là các mở rộng của mô hình.

4.1. Các thành phần cơ bản

Mô hình E-R do P.P.Chen đề xuất vào năm 1976. Mô hình này 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 quan hệ giữa chúng.

Các khái niệm chủ yếu được sử dụng trong lý thuyết của mô hình này là thực thể, liên kết.

4.1.1. Thực thể

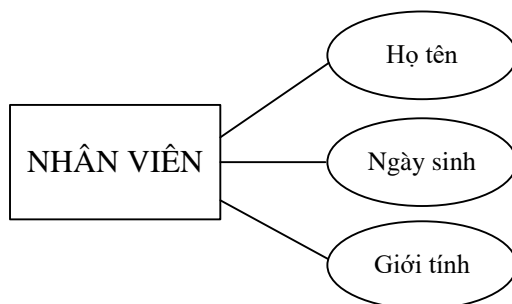
Một thực thể (entity) là một “vật” hay một “đối tượng” trong thế giới thực, phân biệt được với những đối tượng khác.

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



Hình 4.1. 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 nhân viên” 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.

Có những kiểu thực thể mà không có bất cứ một tập thuộc tính nào tạo thành khóa cho nó, điều đó nghĩa là sẽ không thể phân biệt được hai thực thể thuộc kiểu này nếu chỉ dựa vào các tổ hợp giá trị các thuộc tính của chúng, một kiểu thực thể như vậy được gọi là kiểu *thực thể yếu*. Một kiểu thực thể có khóa được gọi là *kiểu thực thể mạnh*.

4.1.2. 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”.

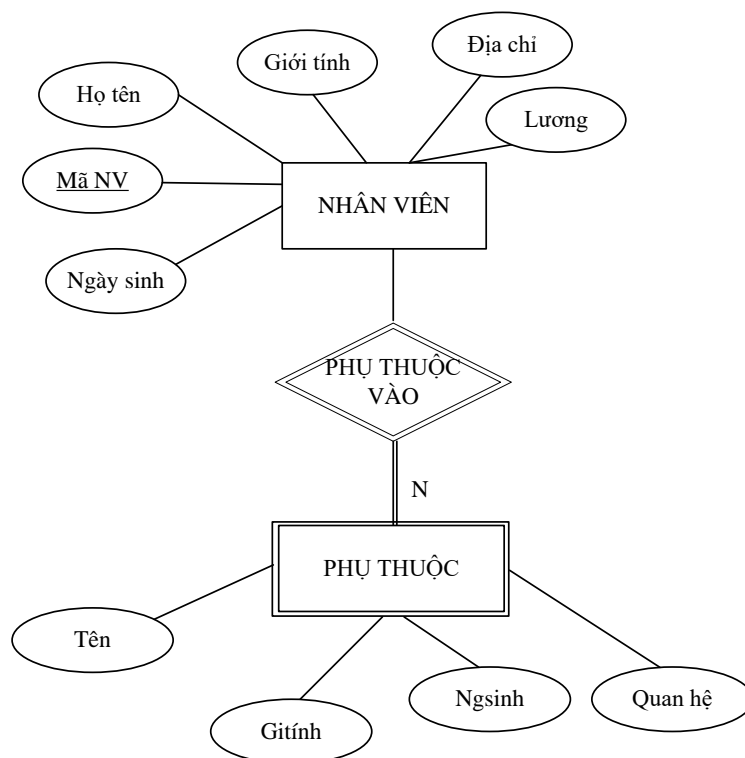
Sau đây là biểu diễn liên kết “Làm việc cho” kết hợp một thực thể “Nhân viên” với một thực thể Phòng:



Hình 4.2. Quan hệ giữa hai thực thể

Cùng một tập thực thể có thể có hơn một vai trò trong cùng một kiểu liên kết, kiểu liên kết như vậy gọi là *kiểu liên kết đệ quy*. Ví dụ, kiểu 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”.

Trong một liên kết, có thể có thực thể chủ và thực thể phụ thuộc vào thực thể chủ đó, trong đó thực thể phụ thuộc là thực thể yếu, quan hệ như vậy người ta gọi là *kiểu liên kết xác định*. Chẳng hạn, trong hình 4.3 là liên kết “Phụ thuộc vào” giữa “Nhân viên” và “Phụ thuộc”, trong đó “Phụ thuộc” chỉ những người thân nhân phụ thuộc vào nhân viên đó.



Hình 4.3. Kiểu liên kết xác định “Phụ thuộc vào” giữa “Nhân viên” và “Phụ thuộc”

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 mối liên kết, ngoài ra còn có thêm những thuộc tính bổ sung khác. Chẳng hạn trong mối liên kết “Tham gia” giữa “Nhân viên” và “Dự án”, mối 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.

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

4.2. Các ràng buộc trên các kiểu liên kết

Các kiểu liên kết thường có những ràng buộc nhằm hạn chế số các tổ hợp có thể của các thực thể tham gia liên kết. Ràng buộc trên một kiểu liên kết được xác định theo hiện trạng của thế giới thực mà liên kết này biểu diễn. Ví dụ, nếu trên thực thể mỗi nhân viên chỉ được làm việc trong một phòng thì ràng buộc này cũng cần được mô tả trong lược đồ. Có hai loại ràng buộc trên kiểu liên kết, đó là ràng buộc về tỉ số lực lượng và ràng buộc về sự tham gia.

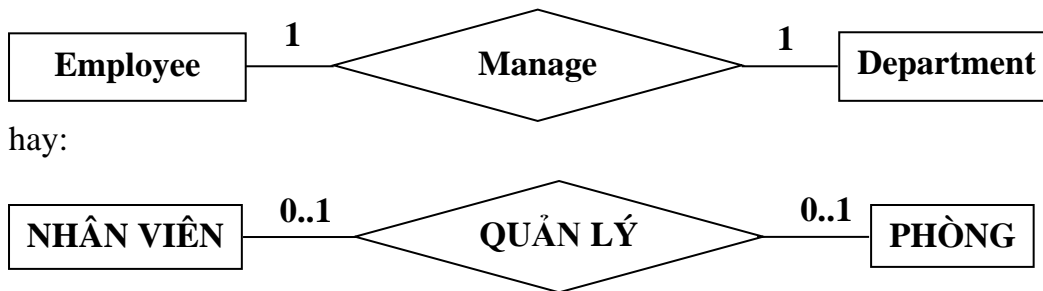
4.2.1. Ràng buộc về tỉ số lực lượng

Tỉ số lực lượng của một quan hệ cấp hai cho biết số các liên kết (của kiểu liên kết này) mà một thực thể có thể tham gia.

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

4.2.1.1. Tỉ số lực lượng 1:1

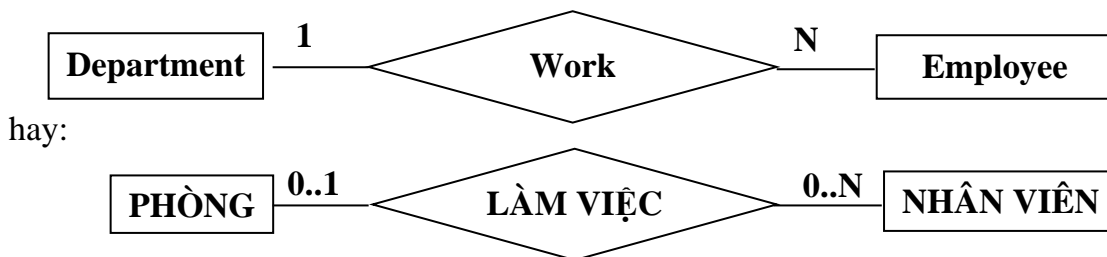
Một ví dụ về tỉ số liên kết 1:1 là kiểu liên kết “Quản lý” giữa “Nhân viên” và “Phòng”, phản ánh mỗi liên quan giữa một thực thể phòng với một thực thể nhân viên là người quản lý của phòng đó. Nếu như một phòng chỉ có thể có một người quản lý và một nhân viên chỉ là người quản lý của tối đa một phòng thì tỉ số của kiểu liên kết là 1:1. (Hình 4.4).



Hình 4.4. Hai cách biểu diễn tỉ số lực lượng 1:1 giữa Nhân viên và Phòng

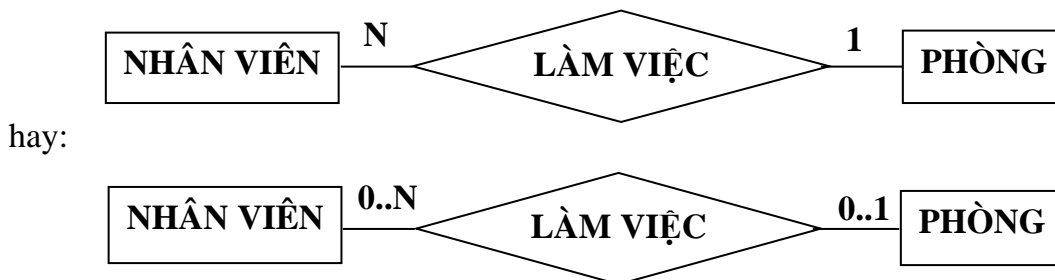
4.2.1.2. Tỉ số lực lượng 1:N hay N:1

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 4.5).



Hình 4.5. Hai cách biểu diễn tỉ số lực lượng 1:N giữa Phòng và Nhân viên

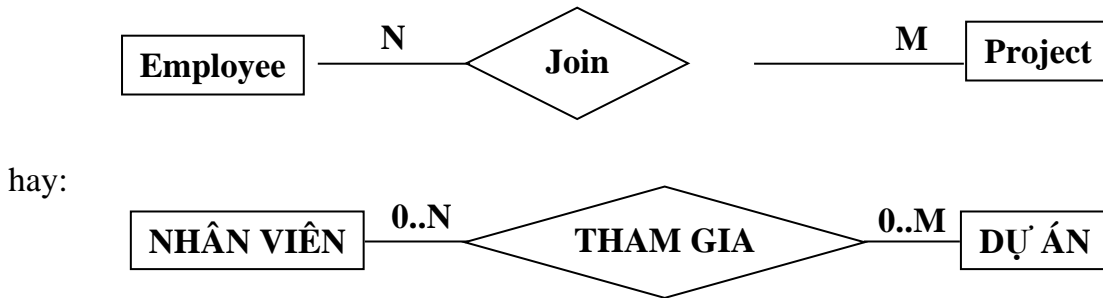
Nếu chúng ta biểu diễn ngược lại thì tỉ số lực lượng là N:1, N là số bất kỳ (Hình 4.6).



Hình 4.6. Hai cách biểu diễn tỉ số lực lượng N:1 giữa Nhân viên và Phòng

4.2.1.3. Tỉ số lực lượng $N:M$

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 4.7).



Hình 4.7. Hai cách biểu diễn tỉ số lực lượng $N:M$ giữa “Nhân viên” và “Dự án”

4.2.2. Ràng buộc về sự tham gia

Ràng buộc về sự tham gia trên một kiểu quan hệ cho biết sự tồn tại của một thực thể có thể phụ thuộc vào mối quan hệ kiểu này giữa nó với một thực thể khác hay không. Ràng buộc tham gia chia thành hai loại: toàn bộ và bộ phận.

Nếu quy định một công ty là mỗi nhân viên phải làm việc cho một phòng nào đó thì mỗi thực thể nhân viên của công ty chỉ có thể tồn tại nếu có tham gia vào một quan hệ “Làm việc cho”. Chúng ta nói rằng sự tham gia của kiểu thực thể “Nhân viên” và kiểu liên kết “Làm việc cho” là sự *tham gia toàn bộ* bởi mỗi thực thể trong toàn bộ tập thực thể “Nhân viên” đều có tham gia một kiểu liên kết thuộc kiểu này. Sự tham gia toàn bộ còn được gọi là sự phụ thuộc tồn tại.

Quay lại ví dụ trước đây về liên kết “Quản lý” giữa “Nhân viên” và “Phòng”. Không phải nhân viên nào cũng là người quản lý của một phòng, như vậy chỉ một bộ phận nào đó của tập thực thể “Nhân viên” tham gia vào kiểu quan hệ “Quản lý”. Chúng ta nói rằng sự tham gia của kiểu thực thể “Nhân viên” vào kiểu quan hệ “Quản lý” là sự *tham gia bộ phận*.

Tỉ số lực lượng và các ràng buộc về sự tham gia còn được gọi chung là các ràng buộc cấu trúc của một kiểu quan hệ.

Có một cách khác nữa để đặc tả các ràng buộc cấu trúc. Với mỗi kiểu liên kết R , chúng ta có thể đưa ra một cặp số nguyên (min, max) kèm theo mỗi kiểu thực thể E trong sự tham gia vào kiểu quan hệ R đó. Cặp số nguyên (min, max), với $\min \geq 0$, và $\max \geq 1$, mang một ý nghĩa như sau: ở mọi thời điểm, mỗi thực thể e thuộc kiểu E phải tham gia ít nhất min quan hệ trong R và chỉ tham gia nhiều nhất max quan hệ trong R .

Ví dụ, mỗi phòng chỉ có 1 nhân viên quản lý, và một nhân viên không tham gia quản lý hoặc chỉ quản lý tối đa 1 phòng.

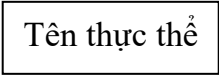
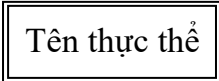


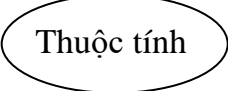
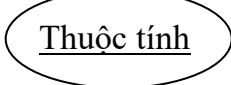
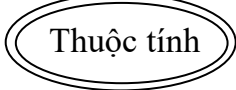
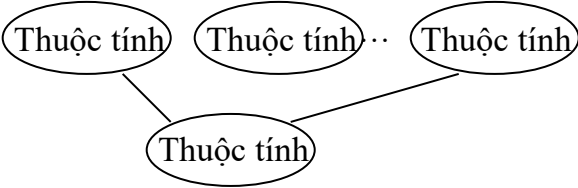
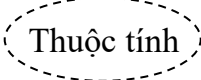

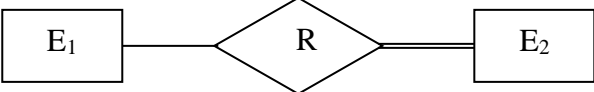
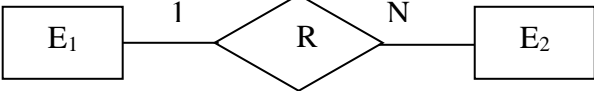
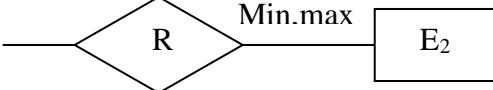


Hình 4.8. Lược đồ sử dụng (min,max)

4.3. Lược đồ thực thể liên kết (lược đồ E-R)

4.3.1. Các thành phần của lược đồ

Cấu trúc logic tổng thể của một CSDL có thể được biểu thị bởi một lược đồ thực thể liên kết, được thành lập từ các thành phần sau đây:

Kiểu thực thể	
Kiểu thực thể yếu	
Kiểu liên kết	
Kiểu liên kết xác định	
Thuộc tính	
Thuộc tính khóa	
Thuộc tính đa trị	
Thuộc tính phức hợp	
Thuộc tính suy diễn được	
Nối các thuộc tính với các tập thực thể và các tập thực thể với các mối liên kết	
Sự tham gia toàn bộ của E ₂ vào kiểu liên kết R	
Tỉ số lược lượng E ₁ :E ₂ là 1:N	
Ràng buộc cấu trúc (min,max) của kiểu thực thể E trong sự tham gia vào kiểu thực thể R	

Việc đặt lựa chọn để đặt tên cho các kiểu thực thể, thuộc tính, kiểu liên kết đòi hỏi chúng ta phải có sự cân nhắc, đặt tên như thế nào để có thể dễ dàng phân biệt các cấu trúc khác nhau trong lược đồ.

Xem phần 4.5 - ví dụ về thiết kế lược đồ thực thể liên kết để thấy rõ hơn việc sử dụng các thành phần nêu trên.

Tuy nhiên, đối với những lược đồ phức tạp có thể vẽ như sau (Xem Ví dụ 2.4 ở phần 2.3.1):

—○ Tên thuộc tính

—● Tên thuộc tính khoá

Hoặc có thể vẽ thực thể và các thuộc tính của thực thể theo dạng:

Tên thực thể	Chẳng hạn:	NHÂN VIÊN
Tên các thuộc tính		Họ tên Ngày sinh Giới tính

4.3.2. Phương pháp thiết kế

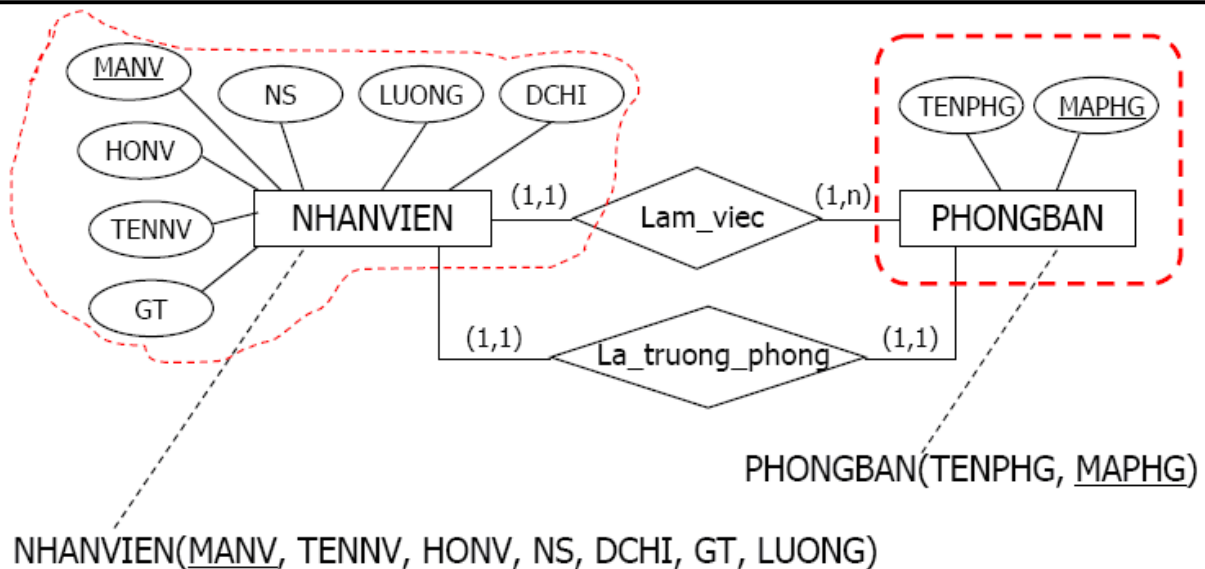
Khi thiết kế, có thể sử dụng các phương pháp thiết kế như sau:

- Thiết kế từ trên xuống: Bắt đầu sự phát triển các mô hình dữ liệu mà nó chứa đựng một vài thực thể và liên kết mức cao, từ đó định nghĩa các thực thể và liên kết mức thấp hơn. Đây là phương pháp thường sử dụng nhất.
- Dưới lên trên: Bắt đầu từ mức cơ bản của thuộc tính (thuộc tính của các thực thể và các liên kết), qua phân tích các mối liên kết giữa các thuộc tính, nhóm chúng lại và thể hiện lại các thực thể và liên kết. Phương pháp này thường sử dụng để thiết kế các CSDL nhỏ.
- Trong ra ngoài: Trước tiên phải định nghĩa một tập hợp các thực thể chính và sau đó xác định các thực thể, các liên kết, các thuộc tính liên hệ với định nghĩa đầu tiên đó.
- Pha trộn: Sử dụng cả hai phương pháp từ dưới lên và từ trên xuống cho các phần khác nhau của mô hình và cuối cùng kết hợp chúng lại với nhau.

4.3.3. Chuyển đổi từ lược đồ E-R sang lược đồ quan hệ

Sau khi thiết kế một lược đồ ở mức khái niệm (dưới dạng lược đồ E-R), chúng ta chuyển sang lược đồ ở mức logic (dưới dạng lược đồ quan hệ) dựa vào các quy tắc sau đây:

- Mỗi thực thể (không phải thực thể yếu) trong lược đồ ER được chuyển đổi thành một quan hệ. Các thuộc tính của thực thể được chuyển đổi thành thuộc tính của quan hệ tương ứng. Thuộc tính khóa của thực thể được chuyển đổi thành thuộc tính khóa của quan hệ tương ứng.



Hình 4.9. Chuyển đổi thực thể và các thuộc tính

– Mỗi liên kết trong lược đồ ER được chuyển theo quy tắc sau:

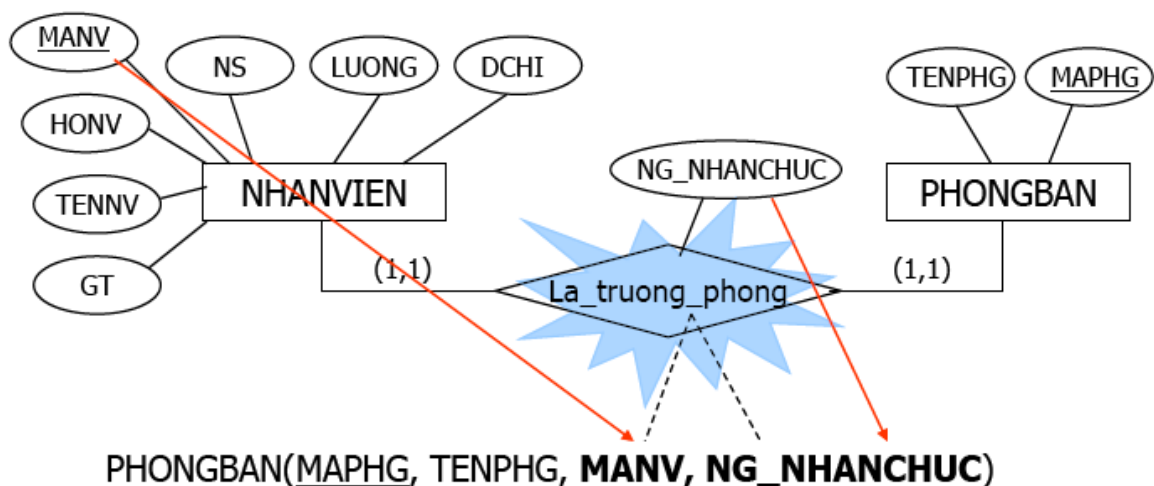
+ Liên kết 1:1: Có các cách sau:

Cách 1: Nếu khóa chính của hai thực thể trùng nhau thì kết hợp chúng thành một quan hệ bằng cách kết hợp tất cả các thuộc tính. Khóa chính được giữ nguyên.

Cách 2: Nếu khóa của hai thực thể khác nhau thì vẫn kết hợp chúng thành một quan hệ bằng cách kết hợp tất cả các thuộc tính. Một trong hai khóa chính sẽ được chọn làm khóa chính của quan hệ kết quả.

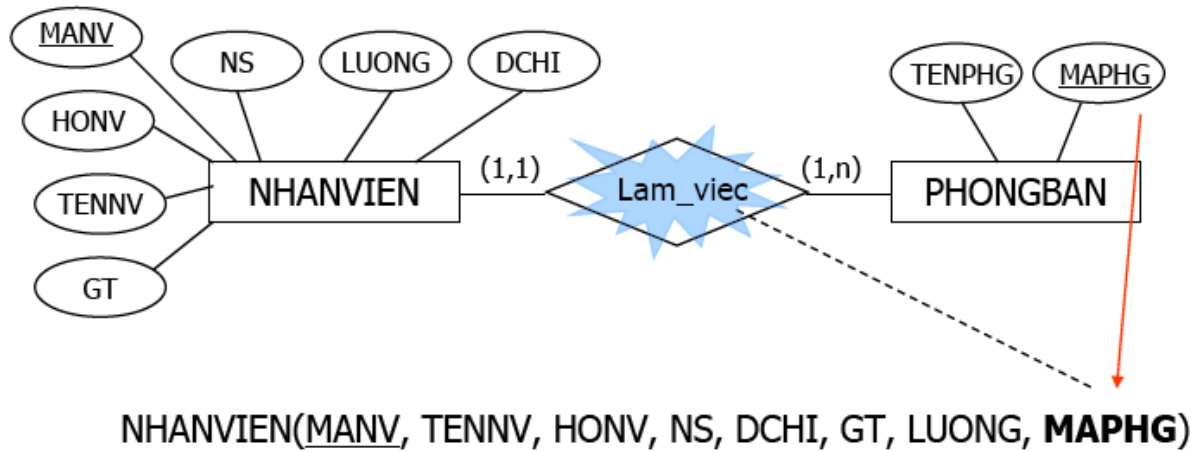
Cách 3: Tạo một quan hệ mới có: Tên quan hệ là tên của mối liên kết. Thuộc tính là các thuộc tính khóa của các thực thể liên quan và thuộc tính của liên kết (nếu có).

Cách 4: Tạo các quan hệ riêng lẻ. Sau đó thêm vào quan hệ này thuộc tính khóa của quan hệ kia.



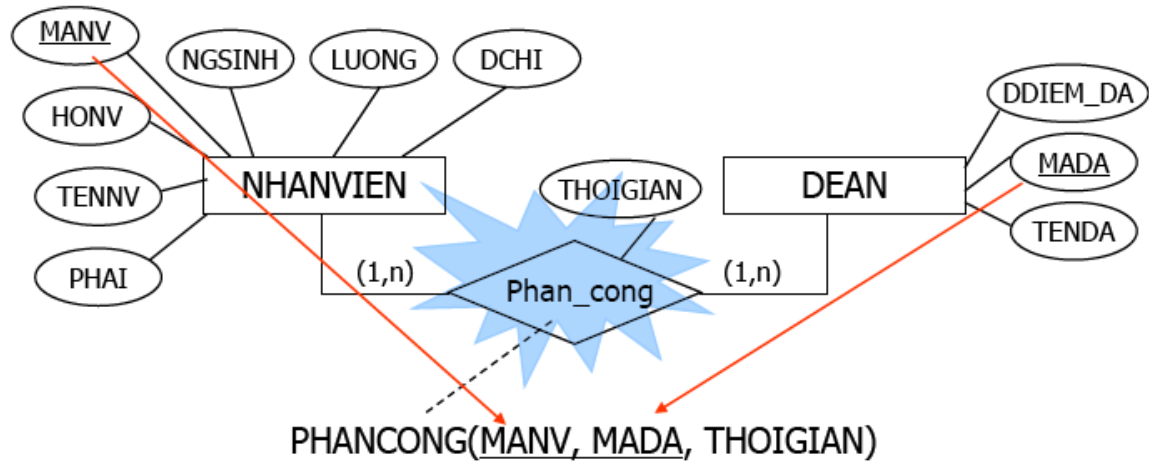
Hình 4.10. Chuyển đổi liên kết một-một theo cách 4

+ Liên kết một:nhiều: Thêm thuộc tính khóa của quan hệ nhiều vào quan hệ một.



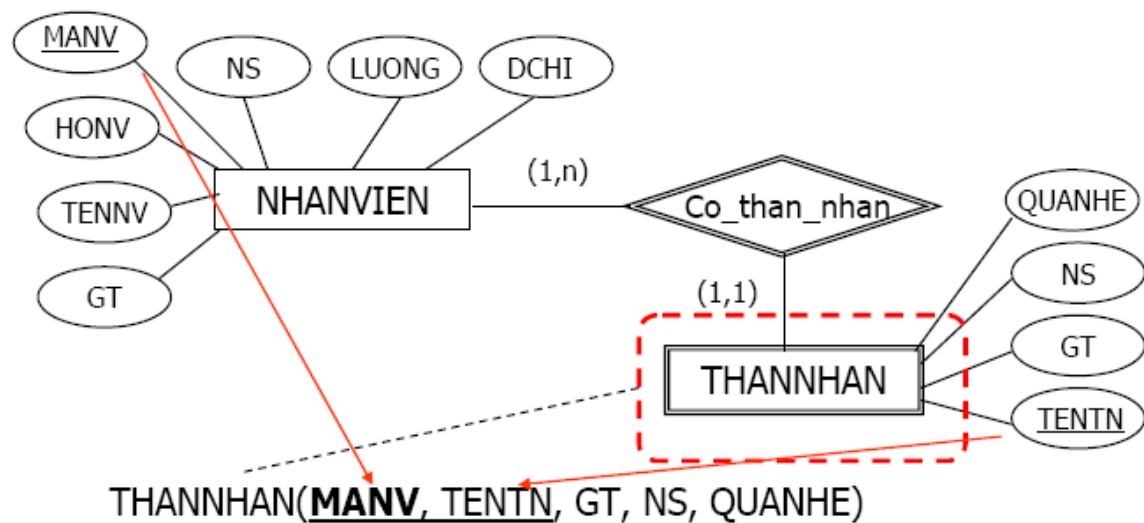
Hình 4.11. Chuyển đổi liên kết một-nhiều

+ Liên kết nhiều-nhiều: Tạo một quan hệ mới có: Tên quan hệ là tên của mối liên kết. Thuộc tính là các thuộc tính khóa của các thực thể liên quan.



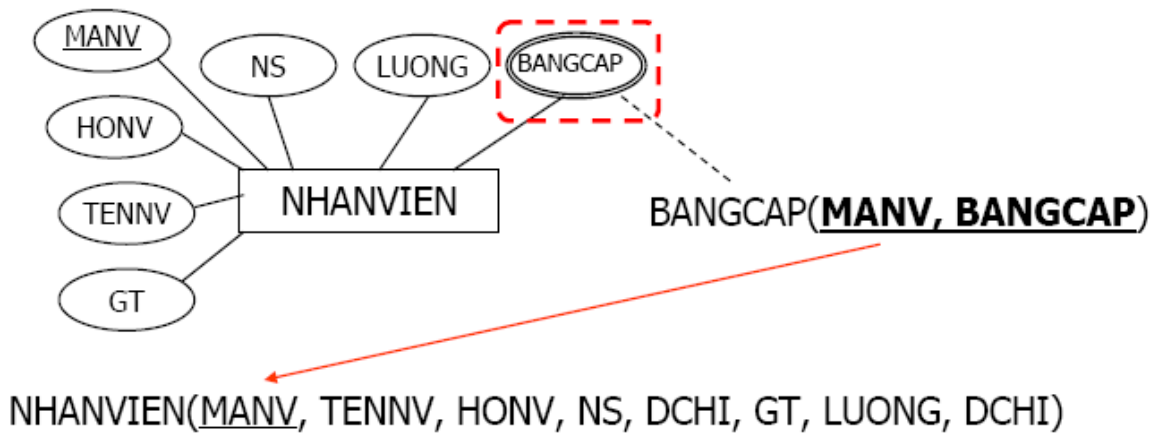
Hình 4.12. Chuyển đổi liên kết nhiều-nhiều

- Thực thể yếu: Chuyển thành một quan hệ có cùng tên với thực thể yếu. Thêm vào thuộc tính khóa của quan hệ liên quan.



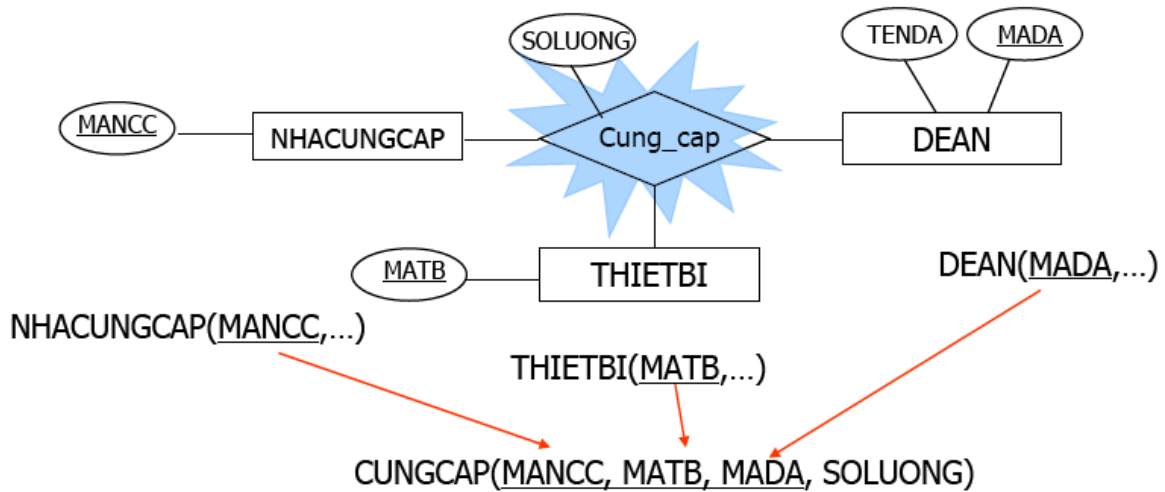
Hình 4.13. Chuyển đổi thực thể yếu

- Thuộc tính đa trị: Chuyển thành một quan hệ có cùng tên với thuộc tính đa trị.



Hình 4.14. Chuyển đổi thuộc tính đa trị

- Liên kết đa ngôi: Chuyển thành một quan hệ có cùng tên với mỗi liên kết đa ngôi. Khóa chính là tổ hợp các khóa của các thực thể tham gia liên kết.



Hình 4.15. Chuyển đổi liên kết đa ngôi

4.4. Các tính chất mở rộng của mô hình E-R

4.4.1. Chuyên biệt hoá

Một tập thực thể có thể chứa các tập con gồm các thực thể phân biệt với các thực thể khác theo một cách nào đó. Nói một cách cụ thể hơn, một tập con của một tập thực thể có thể có một số thuộc tính mà những thực thể ngoài tập con này không có.

Ví dụ 4.1:

Xem xét kiểu thực thể TÀI KHOẢN có các thuộc tính Số hiệu tài khoản và Số dư.

Trên thực tế một tài khoản còn được phân biệt theo một trong hai loại sau: tài khoản tiết kiệm và tài khoản kiểm tra. Mỗi một trong hai loại tài khoản như vậy được mô tả bởi một tập các thuộc tính gồm tất cả các thuộc tính mà kiểu thực thể TÀI KHOẢN vốn có, ngoài ra còn có các thuộc tính riêng của loại tài khoản đó. Cụ thể là các thực thể tài khoản tiết kiệm được mô tả thêm bằng thuộc tính tỉ suất (tỉ số lãi suất), trong khi đó thực thể loại kiểm tra được mô tả thêm bằng thuộc tính vượt trội (số lương rút vượt quá số lượng gửi).

Việc thiết kế các nhóm con trong một tập thực thể được gọi là sự chuyên biệt hoá. Việc chuyên biệt hoá thực thể TÀI KHOẢN như vậy cho phép chúng ta phân biệt các loại tài khoản trên cơ sở kiểu tài khoản.

Một kiểu thực thể có thể được chuyên biệt hoá nhờ hơn một tính chất phân biệt. Ở ví dụ trên tính chất phân biệt là loại tài khoản. Có một sự chuyên biệt hoá khác, chẳng hạn có thể dựa vào tính chất phân biệt là loại chủ tài khoản, kết quả là có hai loại, tài khoản kinh doanh và tài khoản cá nhân. Khi có hơn một sự chuyên biệt hoá áp dụng cho một kiểu thực thể thì một thực thể có thể thuộc vào cả hai chuyên biệt hoá. Ví dụ có tài khoản vừa là tài khoản kinh doanh, vừa là tài khoản kiểm tra. Chúng ta có thể dùng chuyên biệt hoá nhiều lần để làm mịn dần lược đồ thiết kế. Nếu áp dụng một chuyên biệt hoá nữa cho kiểu thực thể TÀI KHOẢN KIỂM TRA, một số loại đặc biệt trong kiểu thực thể này sẽ được tạo nên.

Trong lược đồ E-R, sự chuyên biệt hoá được mô tả bằng một hình tam giác có nhãn là ISA. Nhãn ISA mang ý nghĩa là "là một" ("is a"), một thực thể kiểu TÀI KHOẢN KIỂM TRA là một thực thể kiểu TÀI KHOẢN. Kiểu liên kết ISA cũng có thể được tham chiếu đến như một liên kết giữa kiểu thực thể ở mức cao với kiểu thực thể ở mức thấp. Các kiểu thực thể ở mức cao và mức thấp vẫn được biểu thị bằng một hình chữ nhật có nhãn như một kiểu thực thể bình thường.

(Xem Hình 4.16)

4.4.2. Khái quát hoá

Quá trình làm mịn dần, từ một kiểu thực thể ban đầu chia thành những nhóm thực thể ở những mức tiếp theo, đã thể hiện một tiến trình thiết kế trên xuống. Trong quá trình thiết kế đó sự phân biệt được thực hiện một cách tường minh. Chúng ta cũng có thể thiết kế theo kiểu dưới lên, nghĩa là nhiều kiểu thực thể có thể khái quát hoá thành một kiểu thực thể ở mức cao hơn trên cơ sở những đặc tính chung.

Đối chiếu với ví dụ đã đề cập khi nói về chuyên biệt hoá, theo một cách nào khác, người thiết kế thoạt đầu có thể xác định kiểu thực thể TÀI KHOẢN KIỂM TRA với các thuộc tính Số hiệu tài khoản, số dư, số vượt trội và kiểu thực thể TÀI KHOẢN TIẾT KIỆM với các thuộc tính Số hiệu tài khoản, số dư, tỉ suất. Rõ ràng hai kiểu thực thể đó có các thuộc tính chung. Những tính chất chung đó có thể biểu diễn bằng sự khái quát hoá với kết quả là có một liên kết tồn tại giữa một kiểu thực thể ở mức cao với một hay nhiều kiểu thực thể ở mức thấp. Ở ví dụ trên, TÀI KHOẢN là kiểu thực thể ở mức cao, TÀI KHOẢN TIẾT KIỆM và TÀI KHOẢN KIỂM TRA là các kiểu thực thể ở mức thấp.

Khi một kiểu thực thể mới được tạo ra là kết quả của chuyên biệt hoá hay khái quát hoá thì có những thuộc tính của thực thể ở mức cao được thực thể ở mức thấp kế thừa. Chẳng hạn các thuộc tính Số hiệu tài khoản, Số dư của TÀI KHOẢN được hai thực thể ở mức thấp TÀI KHOẢN TIẾT KIỆM và TÀI KHOẢN KIỂM TRA kế thừa. Các kiểu thực thể ở mức thấp cũng kế thừa sự tham gia vào những kiểu liên kết mà những thực thể ở mức cao đã tham gia. Những tính chất kế thừa như vậy được áp dụng cho tất cả các thực thể ở mức thấp.

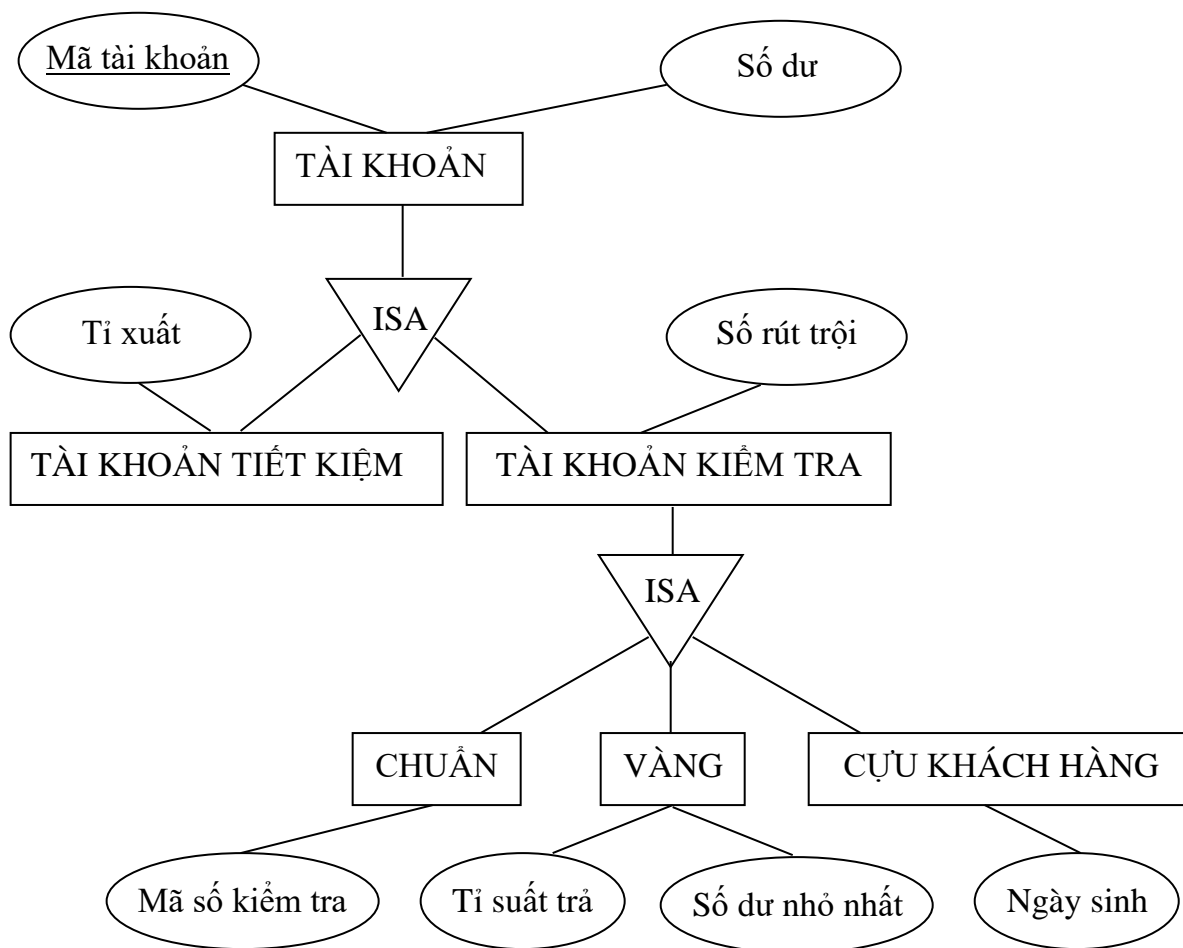
Dễ thấy rằng khái quát hoá là quá trình ngược lại của chuyên biệt hoá. Trong quá trình thiết kế lược đồ E-R có thể vận dụng phối hợp cả hai phương pháp này. Các lược đồ E-R không phân biệt khái quát hoá với chuyên biệt hoá.

Sự khác nhau cơ bản giữa phương pháp khái quát hoá và phương pháp chuyên biệt hoá là ở điểm khởi đầu và mục đích tổng thể của chúng:

- Chuyên biệt hoá bắt đầu từ một kiểu thực thể, nó làm nổi bật lên sự khác nhau giữa các thực thể trong tập thực thể này bằng việc tạo ra các tập thực thể ở mức thấp hơn. Các kiểu thực thể ở mức thấp hoặc có những thuộc tính hoặc tham gia vào những liên kết không áp dụng được cho mọi thực thể mức cao. Người thiết kế thực hiện chuyên biệt hoá để biểu diễn những tính chất phân biệt như vậy.

- Khái quát hoá lại bắt đầu từ sự nhận biết rằng một số kiểu thực thể có tính chất chung. Nói một cách khác, một số kiểu thực thể được mô tả bằng một số thuộc tính giống nhau hoặc chúng tham gia vào cùng các kiểu liên kết. Người thiết kế dựa vào những đặc tính chung đó để tổng hợp những kiểu thực thể này lại trong một kiểu thực thể ở mức cao hơn. Khái quát hoá là để làm nổi bật lên sự tương tự giữa các kiểu thực thể ở mức thấp và giấu đi sự khác biệt giữa chúng.

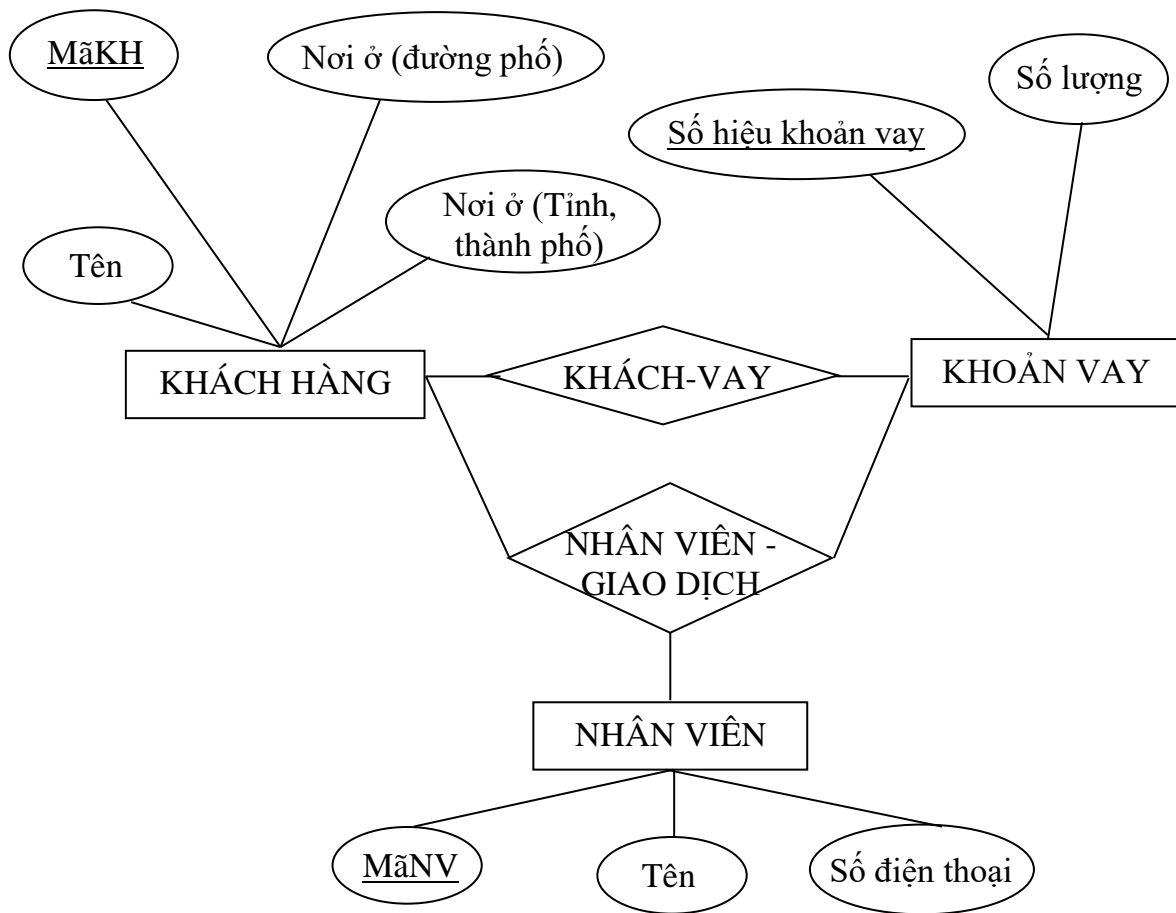
(Xem Hình 4.16)



Hình 4.16. Chuyên biệt hoá và khái quát hoá

4.4.3. Phép gộp

Mô hình E-R có hạn chế ở chỗ nó không biểu diễn mối liên kết giữa các liên kết. Chúng ta hãy xem xét lại ví dụ về một CSDL ngân hàng với thông tin về các khách hàng và những lần vay tiền của họ. Giả sử rằng mỗi cặp thông tin (khách hàng, vay) có một nhân viên ngân hàng thực hiện giao dịch này. Ta có lược đồ như Hình 4.17.

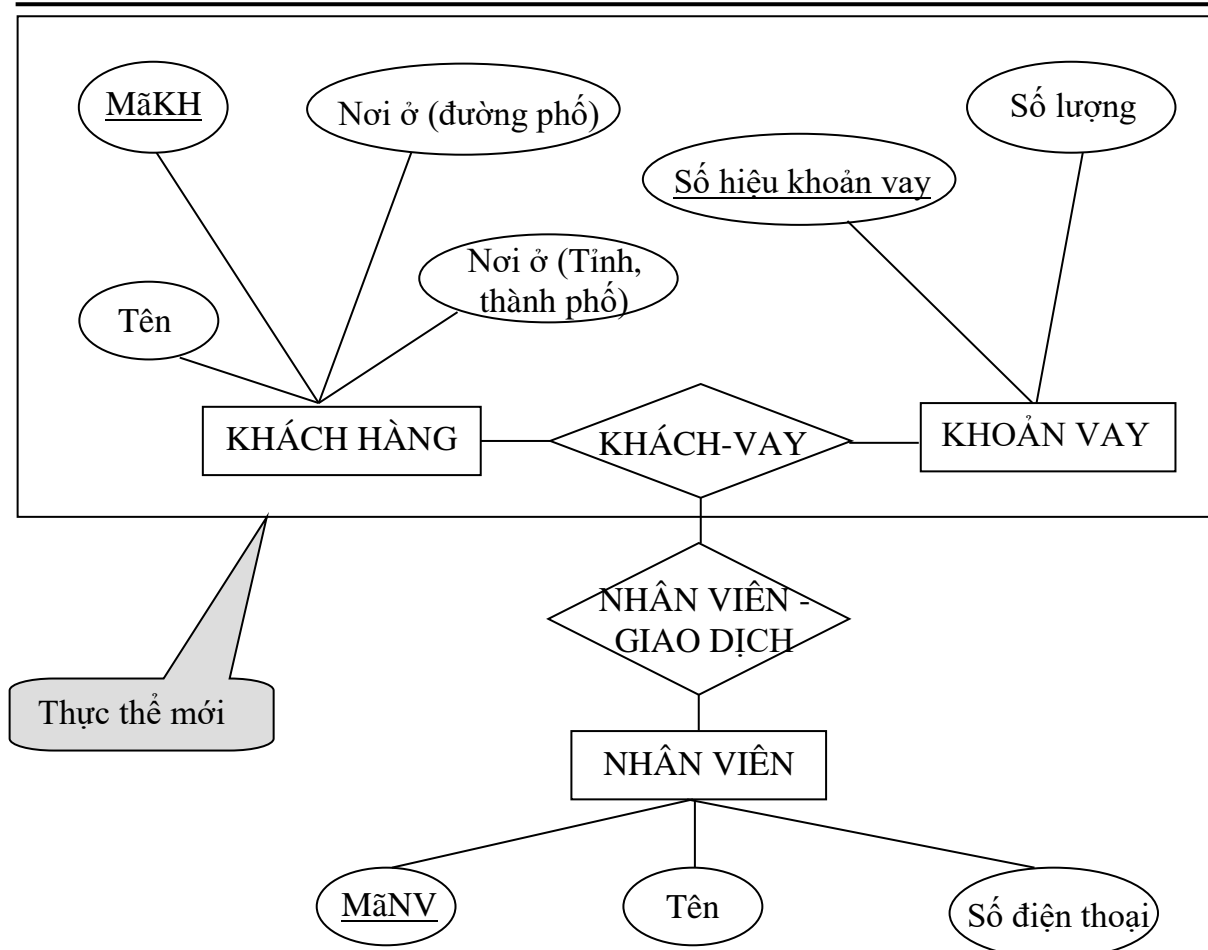


Hình 4.17. Lược đồ E-R có các liên kết thừa

Trong lược đồ đó xuất hiện kiểu liên kết KHÁCH-VAY và kiểu liên kết NHÂN VIÊN-GIAO DỊCH.

Có thể gộp hai kiểu liên kết này thành một kiểu liên kết, nhưng chúng ta không làm vậy vì điều đó làm cho cấu trúc logic trong lược đồ trở nên khó hiểu. Sự kết hợp hai kiểu liên kết đó sẽ cho thấy một liên kết nhân viên-giao dịch phải được gắn với mỗi cặp khách-vay, điều này không đúng. Tuy nhiên, ở đây có một sự dư thừa thông tin: mỗi cặp khách hàng (khách hàng, vay) xuất hiện trong tập liên kết KHÁCH-VAY cũng xuất hiện trong tập liên kết NHÂN VIÊN-GIAO DỊCH.

Phép gộp là công cụ của mô hình E-R để giải quyết những tình huống trên đây. Phép gộp thực chất là sự trừu tượng hoá những liên kết vốn được xử lý như những thực thể ở mức cao. Trong tình huống cụ thể trên đây, chúng ta sẽ xem kiểu liên kết KHÁCH-VAY và kiểu thực thể KHÁCH HÀNG, KHOẢN VAY như một kiểu thực thể ở mức cao có tên gọi là KHÁCH-VAY. Cần nhấn mạnh rằng: KHÁCH-VAY đó tham gia vào mô hình E-R với tư cách là một kiểu thực thể.



Hình 4.18. Lược đồ E-R với phép gộp

4.5. Một ví dụ về thiết kế lược đồ E-R

4.5.1. Mô tả bài toán

Giả sử sau bước tập hợp các yêu cầu và phân tích, người thiết kế CSDL mô tả lại công ty như sau:

1/ Công ty được tổ chức thành các phòng. Mỗi phòng có một tên duy nhất, một mã số duy nhất và một trưởng phòng. Việc một nhân viên được giao làm trưởng phòng được ghi lại ngày mà nhân viên đó được bổ nhiệm. 2/ Một phòng quản lý một số dự án. Mỗi dự án có một tên duy nhất, một mã số duy nhất và một địa điểm.

3/ Với mỗi nhân viên của công ty cần lưu trữ những thông tin sau: họ tên, mã số bảo hiểm xã hội, địa chỉ, lương, giới tính, ngày sinh. Một nhân viên được phân công vào một phòng nào đó, nhưng có thể làm việc với một số dự án mà không nhất thiết tất cả các dự án này đều do phòng anh ta quản lý. Chúng ta cũng lưu trữ số giờ làm việc/tuần của mỗi nhân viên trên mỗi dự án và lưu thông tin người giám sát trực tiếp của mỗi nhân viên.

4/ Mỗi nhân viên có những người phụ thuộc, những người này được hưởng chế độ bảo hiểm. Thông tin về những người phụ thuộc có: họ tên, giới tính, ngày sinh, mối liên kết đối với nhân viên (là con hay là bố, mẹ...).

4.5.2. Xác định các kiểu thực thể, các thuộc tính và kiểu liên kết

Theo mô tả ở trên, chúng ta có thể xác định 4 kiểu thực thể:

1/ Kiểu thực thể PHÒNG với các thuộc tính Tên phòng, Mã phòng, Địa điểm, Trưởng phòng, Ngày bổ nhiệm. Các thuộc tính Tên phòng và Mã phòng đều là các thuộc tính khóa do mỗi giá trị của mỗi thuộc tính này xác định duy nhất một phòng. Địa điểm là thuộc tính đa trị vì một phòng có thể có nhiều địa điểm.

2/ Kiểu thực thể DỰ_ÁN với các thuộc tính Tên dự án, Mã dự án, Địa điểm, Phòng quản lý. Cả hai thuộc tính Tên dự án và Mã dự án đều là thuộc tính khóa.

3/ Kiểu thực thể NHÂN VIÊN với các thuộc tính Họ tên, Mã BHXH (mã số bảo hiểm xã hội), Giới tính, Địa chỉ, Lương, Ngày sinh, Tên phòng và Người hướng dẫn. Hai thuộc tính Họ tên và Địa chỉ là hai thuộc tính phức hợp. Thuộc tính Mã BHXH là thuộc tính khóa.

4/ Kiểu thực thể PHỤ THUỘC với các thuộc tính Nhân viên, Họ tên PT (họ tên của người phụ thuộc), Giới tính, Ngày sinh, Liên kết (đối với nhân viên).

Giữa các kiểu thực thể trên, chúng ta phát hiện được các liên kết sau:

1/ Mỗi phòng có một nhân viên lãnh đạo (trưởng phòng), nên giữa NHÂN VIÊN và PHÒNG có một kiểu liên kết chúng ta đặt tên là QUẢN LÝ. Mỗi nhân viên chỉ có thể làm trưởng phòng của một phòng và một phòng không có hơn một trưởng phòng nên đây là kiểu liên kết 1:1. PHÒNG tham gia toàn phần vào liên kết này (phòng nào cũng có trưởng phòng) còn NHÂN VIÊN chỉ tham gia bộ phận (không phải nhân viên nào cũng làm trưởng một phòng nào đó). Kiểu liên kết QUẢN LÝ có thuộc tính Ngày bổ nhiệm (ngày bổ nhiệm nhân viên đó làm trưởng phòng).

2/ Mỗi nhân viên đều được phân công vào một phòng nào đó, một phòng có thể có nhiều nhân viên. Điều đó cho thấy có một kiểu liên kết N:1 giữa NHÂN VIÊN và PHÒNG. Cả hai kiểu thực thể tham gia liên kết này đều tham gia toàn phần bởi phòng nào cũng có nhân viên và không có nhân viên nào lại không được phân công vào một phòng cụ thể. Chúng ta đặt tên cho kiểu liên kết này là LÀM VIỆC CHO.

3/ Giữa PHÒNG và DỰ_ÁN có một kiểu liên kết gọi là ĐIỀU HÀNH (do phòng điều hành các dự án). Đây là kiểu liên kết 1:N vì một phòng có thể điều hành nhiều dự án nhưng mỗi dự án chỉ chịu sự điều hành của một phòng. Kiểu thực thể DỰ_ÁN tham gia toàn phần vào liên kết này bởi dự án nào cũng phải chịu sự điều hành của một phòng nào đó. Kiểu thực thể PHÒNG tham gia bộ phận hay toàn phần tùy thuộc vào việc có hay không một phòng không có chức năng điều hành dự án.

4/ Có những nhân viên được giao nhiệm vụ giám sát (hay hướng dẫn) một số nhân viên khác. Bởi vậy có một kiểu liên kết đệ quy, chúng ta gọi là HƯỚNG DẪN, giữa kiểu thực thể NHÂN VIÊN (với vai trò là người hướng dẫn) với chính nó (với vai trò người được hướng dẫn). Một người có thể hướng dẫn một số người nhưng không có nhân viên nào có hơn một người hướng dẫn, bởi vậy đây là kiểu liên kết 1:N. Sự tham gia của kiểu thực thể NHÂN VIÊN ở cả hai phía của liên kết đều là bộ phận do không phải nhân viên nào cũng được hướng dẫn và không phải nhân viên nào cũng được hướng dẫn và không phải nhân viên nào cũng là người hướng dẫn.

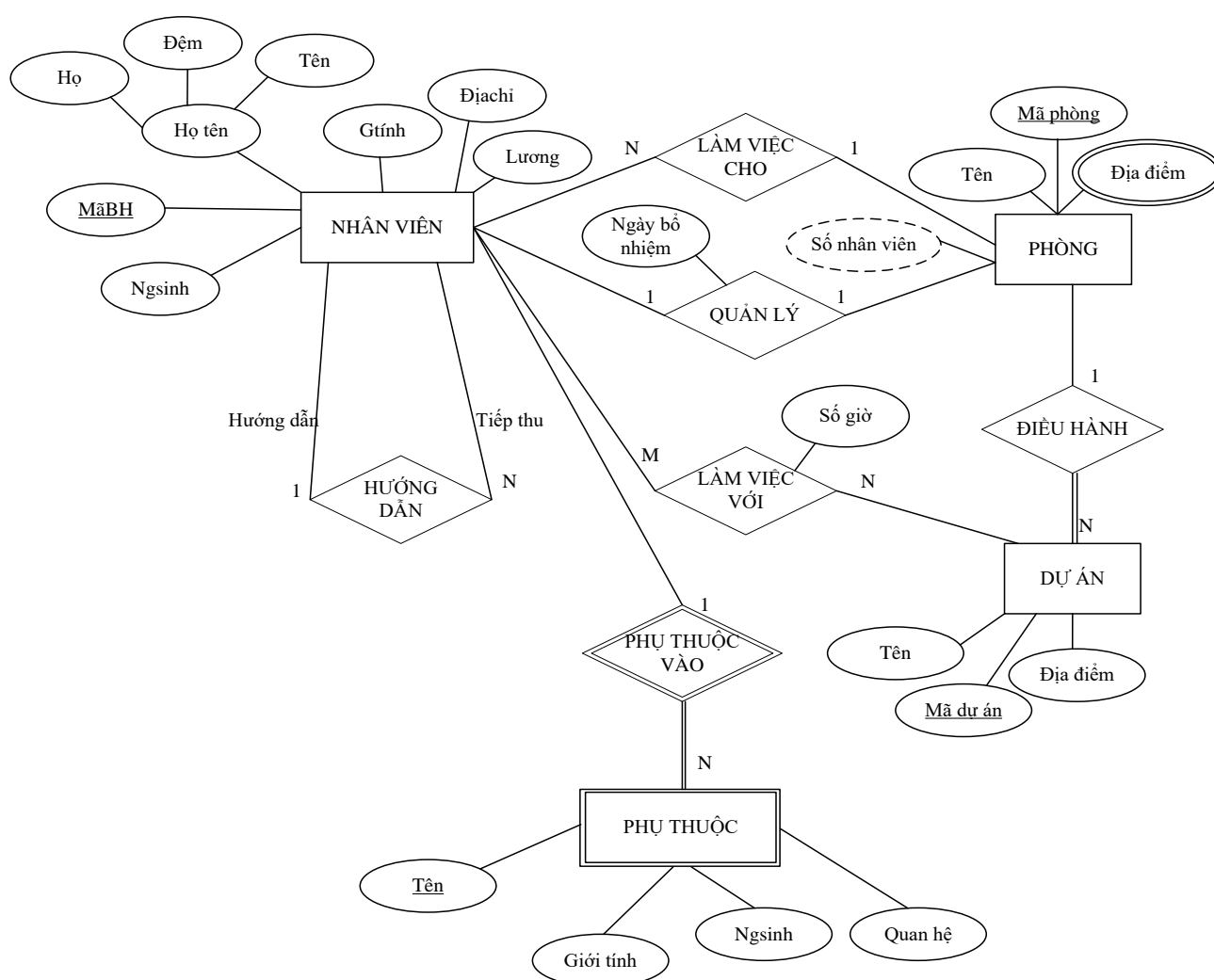
5/ Một số nhân viên tham gia vào các dự án. Liên kết LÀM VIỆC VỚI giữa hai kiểu thực thể NHÂN VIÊN và DỰ_ÁN là kiểu liên kết M:N do một nhân viên có thể tham gia nhiều dự án và một dự án có sự tham gia của một số nhân viên. Nếu có nhân viên không tham gia một dự án nào và có dự án (do công ty quản lý) nhưng

không có nhân viên nào tham gia, thì sự tham gia của hai kiểu thực thể thành phần vào liên kết đều là bộ phận. Liên kết này có thuộc tính Số giờ cho biết số giờ công/tuần tương ứng với cặp thông tin (nhân viên, dự án).

6/ Giữa kiểu thực thể NHÂN VIÊN và kiểu thực thể PHỤ THUỘC có một liên kết: một nhân viên có thể có những người phụ thuộc. Đây là kiểu liên kết 1:N. Sự tham gia của NHÂN VIÊN là bộ phận nhưng PHỤ THUỘC là tham gia toàn bộ vào liên kết này. Chúng ta đặt tên kiểu liên kết này là PHỤ THUỘC VÀO.

4.5.3. Lược đồ E-R

Sau các phân tích nói trên, trong mỗi kiểu thực thể chúng ta cần loại bỏ các thuộc tính được xem như các kiểu liên kết. Chúng ta sẽ loại bỏ thuộc tính Trưởng phòng, Ngày bỏ nhiệm khỏi kiểu thực thể PHÒNG, Phòng quản lý khỏi DỰ ÁN, Tên phòng, Người hướng dẫn khỏi NHÂN VIÊN và Nhân viên khỏi kiểu thực thể PHỤ THUỘC. Cuối cùng chúng ta có một lược đồ E-R như Hình 4.12.



Hình 4.19. Lược đồ E-R

TỔNG KẾT CHƯƠNG 4

- Các thành phần cơ bản của mô hình E-R là tập thực thể và tập liên kết.
- Một thực thể (entity) là một “vật” hay một “đối tượng” trong thế giới thực, phân biệt được với những đối tượng khác. Một tập thực thể (entity type) 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 (bao gồm 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).
- Một liên kết là một sự kết hợp của một số thực thể. Cấp của một kiểu liên kết là số kiểu thực thể tham gia vào kiểu liên kết đó.
- Tỉ số lực lượng của một liên kết cấp hai cho biết số các liên kết (của kiểu liên kết này) mà một thực thể có thể tham gia. Tỉ số lực lượng trên một kiểu liên kết cấp hai có thể gặp là 1:1, 1:N hay N:1, N:M.
- Các ràng buộc trên các kiểu liên kết: Ràng buộc về tỉ số lực lượng, Ràng buộc về sự tham gia.
- Các vấn đề phát sinh: Dùng kiểu thực thể hay kiểu thuộc tính, Dùng kiểu thực thể hay kiểu liên kết, Kiểu liên kết cấp hai và kiểu liên kết cấp n.
- Các tính chất mở rộng của mô hình E-R: Đặc biệt hoá, Khái quát hoá, Phép gộp.
- Chuyên biệt hoá bắt đầu từ một kiểu thực thể, nó làm nổi bật lên sự khác nhau giữa các thực thể trong tập thực thể này bằng việc tạo ra các tập thực thể ở mức thấp hơn. Các kiểu thực thể ở mức thấp hoặc có những thuộc tính hoặc tham gia vào những liên kết không áp dụng được cho mọi thực thể mức cao. Người thiết kế thực hiện đặc biệt hoá để biểu diễn những tính chất phân biệt như vậy.
- Khái quát hoá lại bắt đầu từ sự nhận biết rằng một số kiểu thực thể có tính chất chung. Nói một cách khác, một số kiểu thực thể được mô tả bằng một số thuộc tính giống nhau hoặc chúng tham gia vào cùng các kiểu liên kết. Người thiết kế dựa vào những đặc tính chung đó để tổng hợp những kiểu thực thể này lại trong một kiểu thực thể ở mức cao hơn. Khái quát hoá là để làm nổi bật lên sự tương tự giữa các kiểu thực thể ở mức thấp và giấu đi sự khác biệt giữa chúng.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 4

4.1. Hãy nêu các khái niệm cơ bản của mô hình E-R

4.2. Cho thêm ví dụ về các thực thể, các liên kết có tỉ số lực lượng 1:1, 1:n, n:m

4.3. Với điều kiện nào thì chúng ta có thể dùng một số các liên kết cấp 2 để biểu diễn một kiểu liên kết cấp 3?

4.4. Hãy lập thiết kế lược đồ thực thể liên kết cho các ví dụ sau:

a. 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.

b. QUẢN LÝ THƯ VIỆN

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

Mỗi quyền sách được đánh một mã sách (MASH) dùng để phân biệt với các quyền 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 (MAĐG), 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 quyền 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 quyền

sách mượn và ngày trả (NGAYTRA). Các quyền sách trong cùng một phiếu mượn không nhất thiết phải trả trong cùng một ngày.

c. 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.

d. 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.

CHƯƠNG 5

HOẠCH ĐỊNH, THIẾT KẾ VÀ QUẢN TRỊ CƠ SỞ DỮ LIỆU

Từ những năm 1970, các hệ CSDL ra đời đã thay thế các tập tin truyền thống, như một phần của hệ thông tin của một tổ chức. Cũng từ đây người ta nhận thức rằng dữ liệu là một tài nguyên quan trọng cần phải được xem trọng như tất cả các tài nguyên khác của tổ chức. Dữ liệu cần được hoạch định, thiết kế và quản trị để phục vụ cho các mục đích lâu dài của tổ chức. Từ đó đã thiết lập một bộ phận gọi là quản trị dữ liệu (Data Administrator - DA) và quản trị cơ sở dữ liệu (Database Administrator - DBA) nhằm đáp ứng các yêu cầu về quản lý và điều khiển dữ liệu và CSDL.

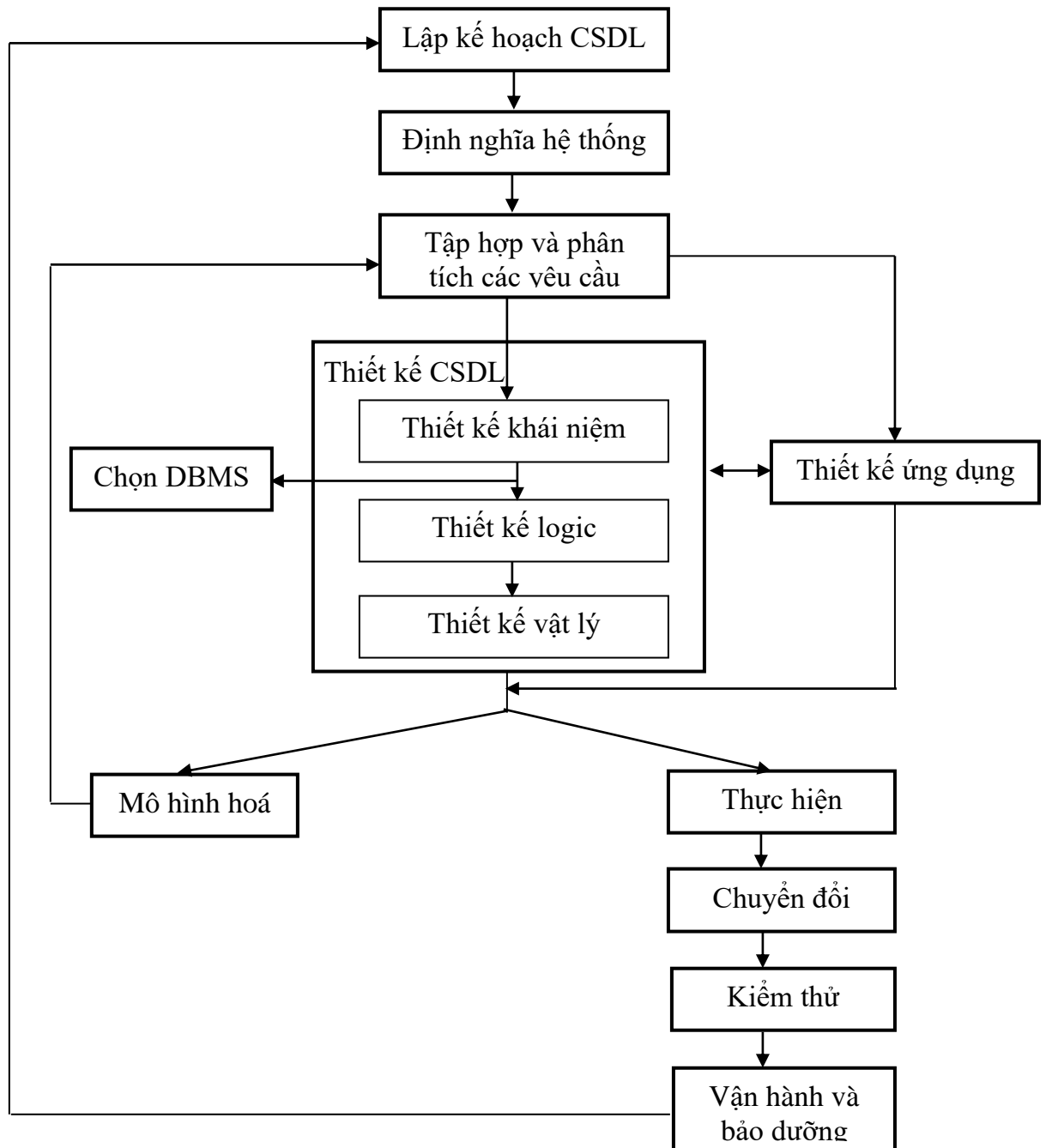
Nội dung của chương này trình bày về các giai đoạn phát triển của hệ CSDL, quá trình thiết kế CSDL.

5.1. Các giai đoạn phát triển của hệ CSDL

Các giai đoạn của một chu kỳ phát triển của hệ CSDL gồm có: lập kế hoạch CSDL, định nghĩa hệ thống, tập hợp và phân tích các yêu cầu, thiết kế CSDL, lựa chọn hệ quản trị CSDL (tùy chọn), thiết kế ứng dụng, mô hình hoá (tùy chọn), thực hiện, chuyển đổi dữ liệu (tùy chọn), kiểm thử, vận hành và bảo dưỡng (Hình 5.1)

Các giai đoạn của một chu kỳ phát triển của một hệ CSDL không tuân theo một tuần tự chặt chẽ mà có sự lặp lại. Chẳng hạn việc tập hợp và phân tích các yêu cầu có thể được thực hiện trong suốt quá trình thiết kế CSDL. Hình 5.1 chỉ thể hiện một số sự lặp lại này.

Đối với các hệ thống nhỏ có ít NSD, chu kỳ sống không cần quá hoàn chỉnh. Tuy nhiên, khi thiết kế một hệ CSDL vừa hoặc lớn với hàng chục ngàn NSD, sử dụng hàng trăm các truy vấn và chương trình ứng dụng thì chu kỳ sống cần phải hoàn chỉnh.



Hình 5.1. Các giai đoạn của chu kỳ phát triển của hệ CSDL

5.1.1. Lập kế hoạch CSDL

Lập kế hoạch CSDL là để quản lý các hoạt động, làm cho các giai đoạn của chu kỳ phát triển thật sự mang lại hiệu quả.

Việc lập kế hoạch CSDL phải được hợp nhất với chiến lược của toàn bộ hệ thống tin của tổ chức. Đối với chiến lược của hệ thống tin, có ba vấn đề chính cần xác định rõ ràng, đó là:

- Xác định kế hoạch, mục đích, yêu cầu của hệ thống tin.
- Xác định những điểm mạnh và yếu của các hệ thống tin hiện tại.
- Sự hỗ trợ của công nghệ thông tin mang đến những thuận lợi gì.

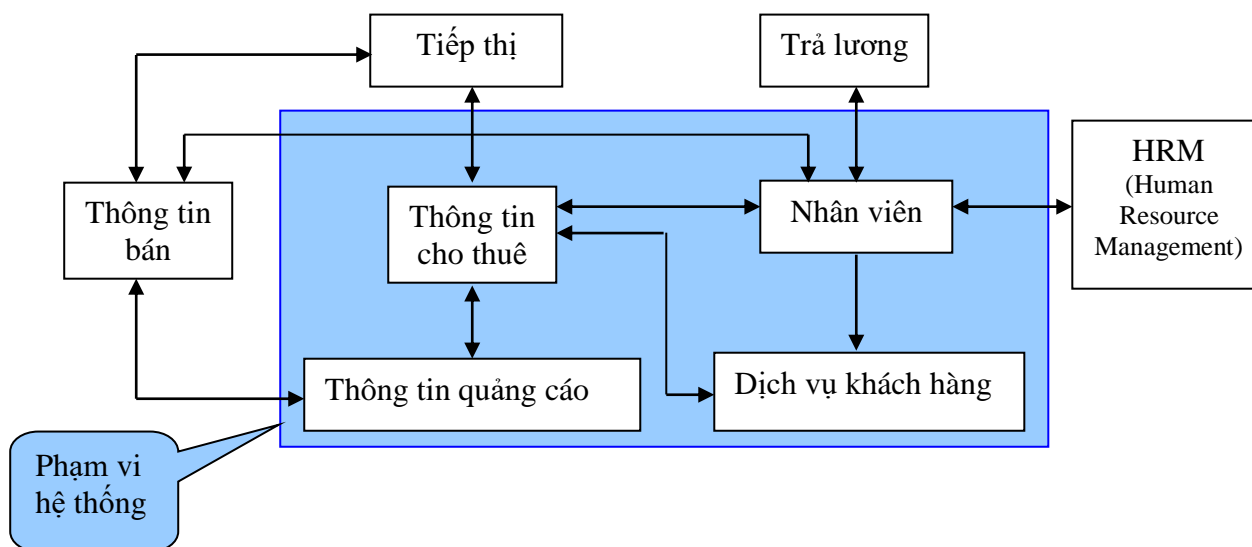
5.1.2. Định nghĩa hệ thống

Định nghĩa hệ thống là mô tả phạm vi của ứng dụng CSDL và quan điểm người dùng chính.

Bước đầu tiên của định nghĩa hệ thống là xác định phạm vi của ứng dụng CSDL.

Ví dụ 5.1:

Phạm vi của ứng dụng cho hệ CSDL DreamHome của một công ty cho thuê nhà:



Hình 5.2. Phạm vi hệ thống của hệ CSDL DreamHome

Với Hình 5.2, có thể thấy mục tiêu của hệ thống là quản lý các thông tin quảng cáo, thông tin cho thuê nhà, các nhân viên cho thuê nhà và khách hàng, không quản lý các thông tin khác như tiếp thị, trả lương, thông tin bán nhà, quản lý nhân sự.

Bước tiếp theo là xác định quan điểm người dùng. Quan điểm người dùng là xác định yêu cầu của hệ CSDL dựa trên một vai trò cụ thể (chẳng hạn: người quản lý, người giám sát) hay lĩnh vực ứng dụng (chẳng hạn: marketing, quản lý nhân sự, quản lý kho...).

Mỗi ứng dụng CSDL có thể có nhiều quan điểm người dùng. Việc xác định quan điểm người dùng có thể giúp chắc chắn là không có một NSD chính nào bị bỏ sót trong quá trình tổng hợp yêu cầu.

Bảng 5.1. Ví dụ về quan điểm người dùng chính

Dữ liệu	Loại cập nhật	Giám đốc	Quản lý	Người giám sát	Trợ lý
Tất cả các nhánh Nhánh đơn	Bảo trì				
	Truy vấn	x	x		
	Báo cáo	x	x		
	Bảo trì		x		
	Truy vấn		x		
	Báo cáo		x		

Tất cả nhân viên	Bảo trì				
	Truy vấn	x	x		
	Báo cáo	x	x		
Nhân viên chi nhánh	Bảo trì		x		
	Truy vấn		x	x	
	Báo cáo		x	x	
Tất cả chủ sở hữu	Bảo trì				
	Truy vấn	x			
	Báo cáo	x	x		
Chủ sở hữu chi nhánh	Bảo trì		x	x	
	Truy vấn		x	x	x
	Báo cáo		x		
Tất cả khách hàng	Bảo trì				
	Truy vấn	x			
	Báo cáo	x	x		
Khách hàng chi nhánh	Bảo trì		x	x	
	Truy vấn		x	x	x
	Báo cáo		x		
Tất cả hợp đồng	Bảo trì				
	Truy vấn	x			
	Báo cáo	x	x		
Hợp đồng chi nhánh	Bảo trì		x	x	
	Truy vấn		x	x	x
	Báo cáo		x	x	
Tất cả báo tin tức	Bảo trì				
	Truy vấn	x			
	Báo cáo	x	x		
Báo tin tức của chi nhánh	Bảo trì		x		
	Truy vấn		x		
	Báo cáo		x		

Ở cột thứ nhất là loại dữ liệu (chẳng hạn dữ liệu về nhân viên, khách hàng), cột thứ hai là loại cập nhật thực hiện trên dữ liệu đó (chẳng hạn như bảo trì dữ liệu, truy vấn dữ liệu, hay xem các báo cáo), các cột sau là những NSD chính (giám đốc, người

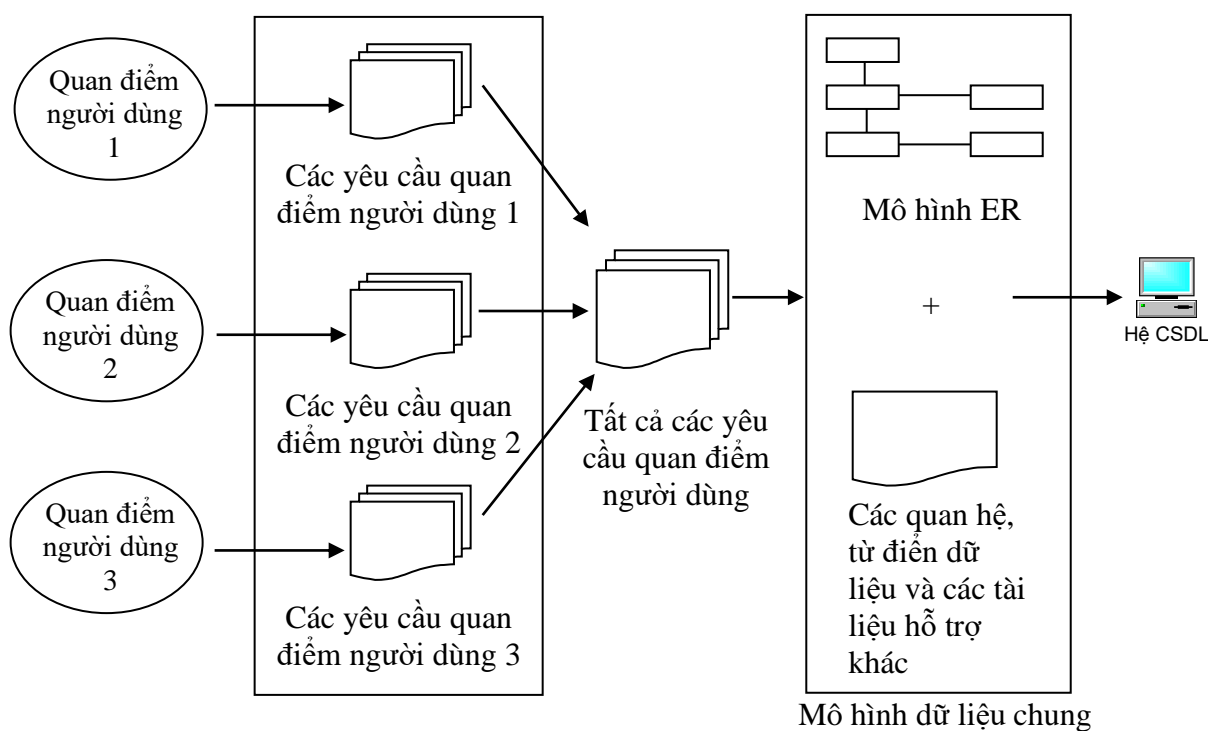
quản lý các chi nhánh, người giám sát, trợ lý). Đánh dấu x nếu NSD có yêu cầu sử dụng loại cập nhật trên dữ liệu nào đó.

5.1.3. Tập hợp và phân tích các yêu cầu

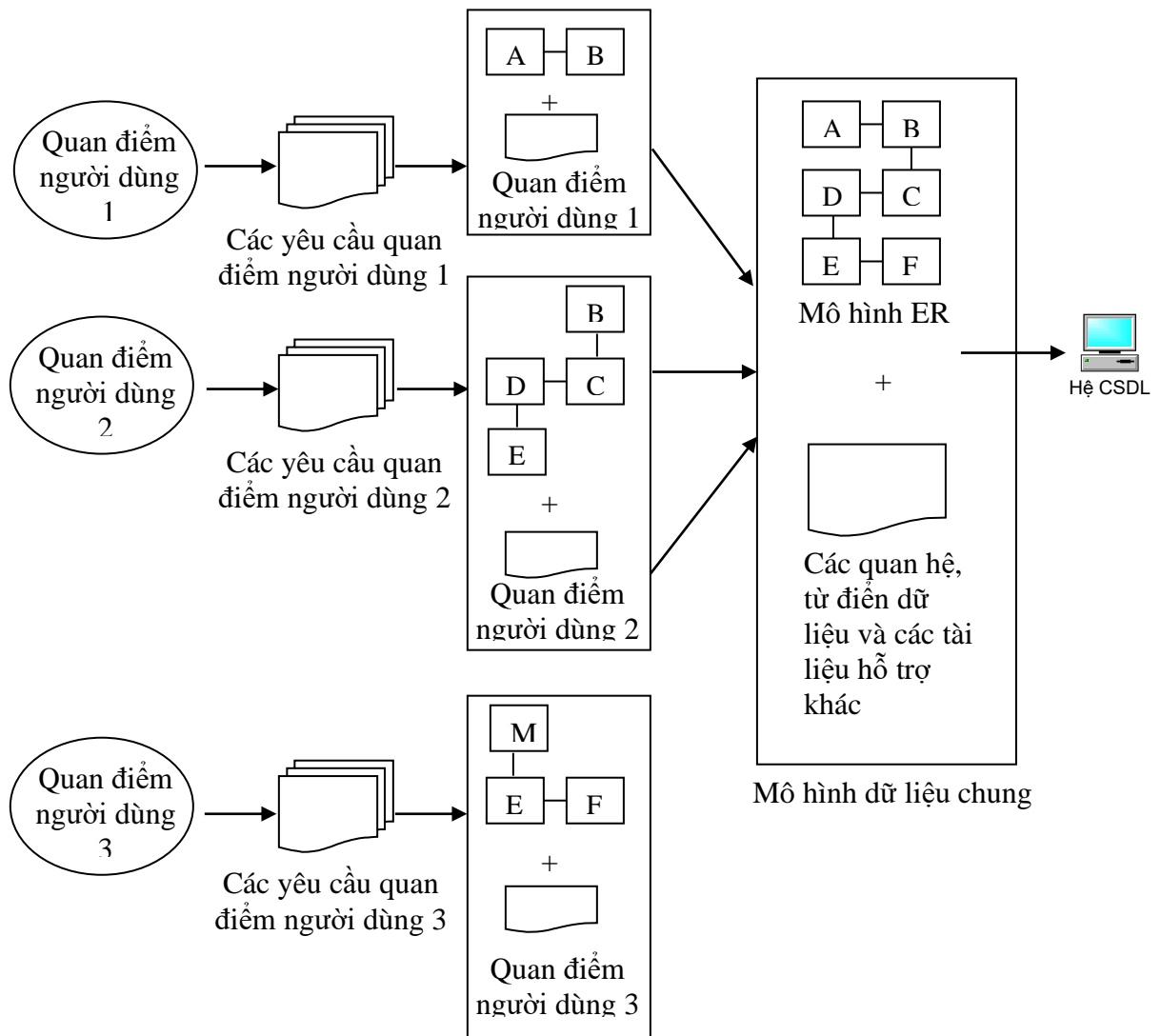
Đây là quá trình tập hợp và phân tích các thông tin về từng phần của tổ chức cần xây dựng ứng dụng CSDL, và sử dụng các thông tin này để xác định các yêu cầu cho hệ thống mới.

Có ba phương pháp chính:

- Phương pháp tập trung: Các yêu cầu cho mỗi quan điểm người dùng được kết hợp thành một tập duy nhất các yêu cầu cho hệ CSDL mới. Một mô hình dữ liệu thể hiện tất cả các quan điểm người dùng được tạo ra suốt giai đoạn thiết kế CSDL.
- Phương pháp kết hợp quan điểm: Các yêu cầu cho mỗi người dùng thể hiện như một danh sách riêng lẻ. Các mô hình dữ liệu thể hiện mối quan điểm người dùng được tạo ra và sau đó kết hợp sau quá trình thiết kế CSDL.
- Kết hợp cả hai phương pháp trên.



Hình 5.3. Phương pháp tập trung để quản lý nhiều quan điểm người dùng



Hình 5.4. Phương pháp tích hợp quan điểm để quản lý nhiều quan điểm người dùng

5.1.4. Thiết kế CSDL

Quá trình thiết kế cơ sở dữ liệu bao gồm các giai đoạn: thiết kế mức logic, thiết kế mức khái niệm, thiết kế mức vật lý. Kết quả của quá trình thiết kế là mô hình dữ liệu (thường dưới dạng mô hình thực thể liên kết hoặc mô hình quan hệ).

(Xem thêm phần 2.1 và 5.3).

5.1.5. Chọn hệ quản trị CSDL (tùy chọn)

Đây là quá trình chọn lựa một hệ quản trị CSDL thích hợp để trợ giúp Hệ cơ sở dữ liệu.

Cho đến nay có khá nhiều hệ quản trị CSDL mạnh được đưa ra thị trường như: Visual FoxPro, MicroSoft Access, SQL-Server, DB2, Sybase, Paradox, Informix, Oracle...

Các bước chính để lựa chọn một hệ quản trị CSDL:

- Định nghĩa các điều kiện.
- Đưa ra hai hoặc ba hệ quản trị CSDL.
- Đánh giá các hệ quản trị CSDL.

- Đưa ra kết quả lựa chọn

Tuỳ theo yêu cầu của ứng dụng, một hệ quản trị CSDL phải đảm bảo các yếu tố sau:

- Có biện pháp bảo mật tốt khi có yêu cầu.
- Có cơ chế giải quyết các vấn đề về tranh chấp dữ liệu. Ví dụ: cấp quyền ưu tiên, đánh dấu yêu cầu truy xuất dữ liệu...
- Có cơ chế sao lưu và phục hồi.
- Phải có giao diện tốt, dễ sử dụng.
- Đáp ứng được tính độc lập giữa dữ liệu và chương trình.

5.1.6. Thiết kế ứng dụng

Thiết kế ứng dụng là quá trình thiết kế giao diện NSD và các chương trình ứng dụng mà chúng sử dụng và xử lý cơ sở dữ liệu.

Thiết kế CSDL và thiết kế ứng dụng là các hoạt động song song với nhau. Thiết kế ứng dụng gồm hai hoạt động: thiết kế giao dịch (hay hoạt động) và thiết kế giao diện người dùng.

5.1.6.1. Thiết kế giao dịch (transaction)

Giao dịch là một hoạt động hay một chuỗi các hoạt động, được thực hiện bởi một NSD hay chương trình ứng dụng, cập nhật hoặc làm thay đổi nội dung của CSDL.

Khi thiết kế giao dịch, nên định nghĩa và viết tài liệu chi tiết về các yêu cầu của giao dịch. Các chi tiết quan trọng cần định nghĩa là:

- Dữ liệu được sử dụng.
- Mô tả các chức năng của giao dịch.
- Kết quả đầu ra của các giao dịch.
- Tầm quan trọng đối với NSD.
- Hiệu quả mong đợi.

5.1.6.2. Thiết kế giao diện người dùng

Thiết kế giao diện là thiết kế phần giao tiếp với người dùng.

Thiết kế giao diện bao gồm các yêu cầu sau:

- Tiêu đề phải thể hiện được ý nghĩa.
- Sắp xếp và nhóm các field (trường) một cách logic.
- Bố trí các form/report một cách hợp lý, đẹp mắt.
- Đánh nhãn các field theo cách thông dụng, quen thuộc.
- Ký hiệu và viết tắt phải phù hợp.
- Sử dụng màu sắc phù hợp.
- Sự phân chia ranh giới cho các field đối tượng dữ liệu.
- Thông báo lỗi về các giá trị không thích hợp.

- Các field tùy chọn phải được đánh dấu rõ ràng.
- Các thông báo giải thích cho các field.
- Báo hiệu hoàn tất.

5.1.7. Mô hình hoá (tùy chọn)

Mô hình hoá là xây dựng mô hình làm việc của một hệ cơ sở dữ liệu.

Có thể mô hình hoá hệ thống hoặc bỏ qua quá trình này. Quá trình này sẽ xây dựng một mô hình để kiểm tra trước khi chính thức thực hiện, nhằm hạn chế các thiếu sót của hệ thống.

Mục đích của việc mô hình hoá là:

- Để xác định các chức năng của hệ thống làm việc tốt hay tồn tại các thiếu sót.
- Đưa ra các đề nghị cải tiến.
- Chọn lọc các yêu cầu của NSD.
- Ước lượng tính khả thi của thiết kế hệ thống.

5.1.8. Thực hiện

Đây là quá trình sử dụng các hệ quản trị cơ sở dữ liệu để xây dựng cơ sở dữ liệu.

Trong quá trình này, có thể sử dụng các Hệ quản trị CSDL như Microsoft Access, Microsoft SQL Server, Oracle, DB2, My SQL... để xây dựng CSDL dựa trên các bảng thiết kế.

5.1.9. Chuyển đổi dữ liệu (tùy chọn)

Quá trình này để chuyển đổi dữ liệu đã tồn tại vào cơ sở dữ liệu mới và và chuyển ứng dụng đã tồn tại chạy trên hệ thống mới.

Quá trình này chỉ tiến hành khi có một hệ thống mới thay thế cho hệ thống cũ, và việc chuyển đổi tùy thuộc vào đặc tính của các hệ thống mới và hệ thống cũ.

5.1.10. Kiểm thử

Đây là quá trình tìm lỗi của hệ thống.

Cách kiểm thử thông dụng nhất là chạy thử từng phần để tìm ra các lỗi và chỉnh sửa các lỗi này. Sau đó chạy thử toàn bộ hệ thống để xem hệ thống có ổn định không.

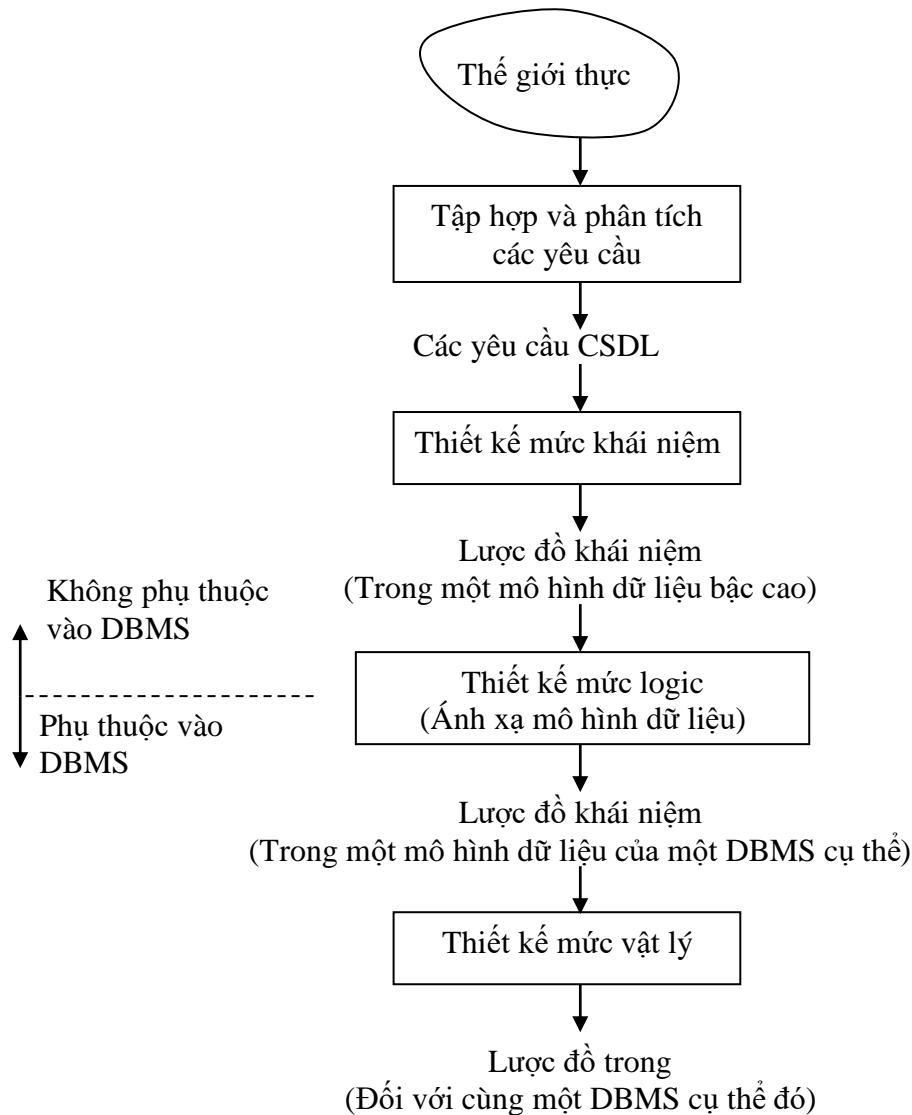
5.1.11. Vận hành và bảo dưỡng

Đây là quá trình kiểm tra và bảo trì cơ sở dữ liệu sau khi cài đặt

Quá trình này bao gồm các hoạt động như kiểm tra hoạt động của hệ thống, bảo trì và cải tiến ứng dụng CSDL khi có yêu cầu, tích hợp khi có các yêu cầu mới.

5.2. Tổng quan về thiết kế cơ sở dữ liệu

Quá trình thiết kế CSDL được minh họa như ở Hình 5.5.



Hình 5.5. Quá trình thiết kế một CSDL

Trước khi thiết kế CSDL, người thiết kế cần tập hợp các yêu cầu và phân tích các yêu cầu này. Các yêu cầu về CSDL được xác định bằng cách phỏng vấn cả người quản lý và NSD dữ liệu, nghiên cứu các tài liệu (chẳng hạn như các báo cáo), và tạo ra một đặc tả cho các yêu cầu. Đặc tả này bao gồm những dữ liệu cần xử lý, mối liên hệ tự nhiên giữa các dữ liệu và hạ tầng phần mềm cho bản cài đặt CSDL. Những đặc tả như vậy càng chi tiết và càng đầy đủ càng tốt. Chẳng hạn đặc tả về sản phẩm, người bán, khách hàng...của một tổ chức.

Tiếp theo, người thiết kế sẽ tiến hành thiết kế CSDL, bao gồm thiết kế mức khái niệm, thiết kế mức logic và thiết kế mức vật lý (xem thêm ở phần 2.1).

5.2.1. Thiết kế CSDL mức khái niệm

Thiết kế CSDL mức khái niệm là quá trình xây dựng mô hình dữ liệu tổng thể, độc lập với các kiến trúc vật lý.

Ở bước này người thiết kế lựa chọn một mô hình dữ liệu, dùng các khái niệm của mô hình đã chọn để chuyển những đặc tả yêu cầu của người dùng sang thành một lược đồ mức khái niệm. Lược đồ mức khái niệm là một mô tả cô đọng về yêu cầu dữ liệu của người dùng bao gồm: mô tả chi tiết các kiểu dữ liệu, các quan hệ, các ràng buộc.

Các khái niệm do mô hình dữ liệu bậc cao cung cấp được sử dụng trong những mô tả của lược đồ khái niệm. Do những khái niệm đó không chứa các chi tiết cài đặt, chúng thường dễ hiểu đối với NSD. Lược đồ khái niệm bậc cao thường được dùng để đảm bảo rằng kết quả của quá trình thiết kế CSDL sẽ đáp ứng được tất cả đòi hỏi của NSD và đảm bảo rằng những đòi hỏi đó không chứa mâu thuẫn. Với bước thiết kế này, người thiết kế CSDL có thể tập trung vào việc đặc tả các tính chất của dữ liệu mà chưa cần quan tâm đến các chi tiết về lưu trữ.

Một lược đồ mức khái niệm được thiết kế một cách đầy đủ cũng bao gồm cả những đặc tả yêu cầu về chức năng, đó là các thao tác (hay giao dịch) được thực hiện trên dữ liệu. Điều này cũng giúp khẳng định rằng lược đồ mức khái niệm thỏa mãn những yêu cầu chức năng đã xác định.

5.2.2. Thiết kế CSDL mức logic

Thiết kế CSDL mức logic hay còn gọi là ánh xạ mô hình dữ liệu là quá trình xây dựng một mô hình dữ liệu tổng thể dựa trên một mô hình dữ liệu đặc trưng, chẳng hạn như mô hình quan hệ.

Hầu hết các hệ quản trị CSDL dùng một mô hình dữ liệu thể hiện (chẳng hạn như mô hình quan hệ hay mô hình hướng đối tượng...), do vậy lược đồ mức khái niệm được chuyển đổi từ mô hình dữ liệu bậc cao sang mô hình dữ liệu thể hiện. Kết quả của bước này là một lược đồ CSDL dưới dạng một mô hình dữ liệu thể hiện của hệ quản trị CSDL.

5.2.3. Thiết kế CSDL mức vật lý

Các đặc điểm về mặt vật lý của CSDL phải được đặc tả ở giai đoạn này, chúng bao gồm các cấu trúc lưu trữ bên trong và kiểu tổ chức tập tin cho CSDL.

5.3. Quản trị dữ liệu và quản trị CSDL

Một trong các lý do chính đối với việc sử dụng Hệ quản trị CSDL là có sự điều khiển trung tâm cho cả dữ liệu lẫn các chương trình truy cập các dữ liệu này. *Người quản trị dữ liệu (Data Administrator - DA) và người quản trị CSDL (DataBase Administrator - DBA) là người quản lý và điều khiển dữ liệu và cơ sở dữ liệu.*

5.3.1. Người quản trị dữ liệu

Người quản trị dữ liệu thực hiện các công việc như sau:

- Lập kế hoạch CSDL.
- Phát triển và bảo trì các tiêu chuẩn, chính sách, thủ tục. Thiết kế CSDL mức khái niệm và mức logic.
- Chọn các công cụ thích hợp.
- Liên lạc với NSD để xác nhận các yêu cầu mới và giải quyết các khó khăn về cập nhật.
- Phát triển chính sách bảo mật.

5.3.2. Người quản trị CSDL

Người quản trị CSDL là người có trách nhiệm quản lý các tài nguyên của hệ CSDL, các tài nguyên đó là các CSDL, hệ quản trị CSDL và các phần mềm liên quan. Đây là người có vai trò thiết kế và cài đặt CSDL về mặt vật lý, cấp phát các quyền truy

cập CSDL, cấp phần mềm và phần cứng theo yêu cầu, duy trì các hoạt động hệ thống hệ thống đảm bảo thỏa mãn yêu cầu của các trình ứng dụng và của người dùng. Như vậy DBA phải là người có những hiểu biết chi tiết và kỹ thuật trong lĩnh vực CSDL, hệ quản trị CSDL và môi trường hệ thống.

Như vậy, các chức năng chính của DBA là:

- Định nghĩa sơ đồ: DBA tạo ra sơ đồ CSDL gốc bằng cách viết một tập các định nghĩa mà nó sẽ được dịch bởi trình biên dịch DDL thành một tập các bảng được lưu trữ thường trực trong tự điển dữ liệu.
- Định nghĩa cấu trúc lưu trữ và phương pháp truy xuất: DBA tạo ra một cấu trúc lưu trữ thích hợp và các phương pháp truy xuất bằng cách viết một tập hợp các định nghĩa mà nó sẽ được dịch bởi trình biên dịch lưu trữ dữ liệu và ngôn ngữ định nghĩa dữ liệu.
- Sửa đổi sơ đồ và tổ chức vật lý.
- Cấp quyền truy xuất dữ liệu: Việc cấp các dạng quyền truy cập khác nhau cho phép người quản trị CSDL điều hoà những phần của CSDL mà nhiều người có thể truy xuất. Thông tin về quyền được lưu giữ trong một cấu trúc hệ thống đặc biệt mà nó được tham khảo bởi hệ CSDL mỗi khi có sự truy xuất dữ liệu của hệ thống.
- Đặc tả ràng buộc toàn vẹn: Các giá trị dữ liệu được lưu trữ trong CSDL phải thỏa mãn một số các ràng buộc nhất quán nhất định. Chẳng hạn, số giờ làm việc của một nhân viên trong một tuần không quá 80 giờ. Một ràng buộc như vậy phải được đặc tả một cách tường minh bởi DBA. Các ràng buộc toàn vẹn được lưu giữ trong một cấu trúc hệ thống đặc biệt được tham khảo bởi hệ CSDL mỗi khi có sự cập nhật dữ liệu.

TỔNG KẾT CHƯƠNG 5

- Chu kỳ phát triển của hệ CSDL bao gồm các giai đoạn: Lập kế hoạch CSDL, định nghĩa hệ thống, tập hợp và phân tích các yêu cầu, thiết kế CSDL, thiết kế ứng dụng, chọn DBMS, mô hình hoá, thực hiện, chuyển đổi, kiểm thử, bảo trì.
- Lập kế hoạch CSDL là để quản lý các hoạt động, cho phép các giai đoạn của chu kỳ sống của ứng dụng CSDL hoạt động hiệu quả.
- Định nghĩa hệ thống là mô tả phạm vi của ứng dụng CSDL và quan điểm người dùng chính.
- Tập hợp và phân tích các yêu cầu là quá trình tập hợp và phân tích các thông tin về từng phần của tổ chức cần xây dựng ứng dụng CSDL, và sử dụng các thông tin này để xác định các yêu cầu cho hệ thống mới.
- Thiết kế CSDL là quá trình tạo một thiết kế cho cơ sở dữ liệu dựa vào các phương pháp thiết kế như trên xuống dưới, dưới lên trên, trong ra ngoài, pha trộn.
- Chọn hệ quản trị CSDL (tuỳ chọn) là hoạt động chọn lựa một hệ quản trị cơ sở dữ liệu thích hợp để trợ giúp hệ thống cơ sở dữ liệu.
- Thiết kế ứng dụng là quá trình thiết kế giao diện NSD và các chương trình ứng dụng mà chúng sử dụng và xử lý cơ sở dữ liệu. Bao gồm thiết kế giao dịch và thiết kế giao diện người dùng.
- Mô hình hoá là xây dựng mô hình làm việc của một hệ cơ sở dữ liệu.
- Thực hiện là sử dụng các hệ quản trị cơ sở dữ liệu để xây dựng cơ sở dữ liệu.
- Chuyển đổi dữ liệu là quá trình để chuyển đổi dữ liệu đã tồn tại vào cơ sở dữ liệu mới và vận chuyển ứng dụng đã tồn tại chạy trên hệ thống mới.
- Kiểm thử là quá trình thực thi hệ thống cơ sở dữ liệu để tìm ra lỗi.
- Thiết kế cơ sở dữ liệu bao gồm ba giai đoạn là Thiết kế CSDL mức khái niệm, Thiết kế CSDL mức logic, Thiết kế CSDL mức vật lý
- Thiết kế CSDL mức khái niệm là quá trình xây dựng mô hình dữ liệu tổng thể, độc lập với các kiến trúc vật lý.
- Thiết kế CSDL mức logic là quá trình xây dựng một mô hình dữ liệu tổng thể dựa trên một mô hình dữ liệu đặc trưng, chẳng hạn như mô hình quan hệ.
- Thiết kế CSDL mức vật lý là quá trình đặc tả các cấu trúc lưu trữ bên trong và kiểu tổ chức tập tin cho CSDL.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 5

- 5.1.** Nêu các giai đoạn phát triển của hệ CSDL
- 5.2.** Nêu các giai đoạn thiết kế CSDL.
- 5.3.** Mục đích của thiết kế ứng dụng.
- 5.4.** Các bước chính để lựa chọn hệ quản trị CSDL.
- 5.5.** Mục đích và các bước để quản trị dữ liệu.
- 5.6.** Giả sử bạn phải chọn một hệ quản trị CSDL cho một nhóm NSD trong tổ chức của bạn. Hãy thiết lập tập hợp các yêu cầu cho nhóm và xác định một tập hợp các chức năng mà một hệ quản trị CSDL phải đáp ứng đầy đủ.
- 5.7.** Bạn hãy phát triển CSDL cho ứng dụng Quản lý sinh viên.
- 5.8.** Bạn hãy phát triển CSDL cho ứng dụng Đặt báo.
- 5.9.** Bạn hãy phát triển CSDL cho ứng dụng Quản lý kho.
- 5.10.** Bạn hãy phát triển CSDL cho ứng dụng Quản lý thư viện.

CHƯƠNG 6 CHUẨN HOÁ

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.

6.1. 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.

6.2. 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.

6.2.1. Sự dư thừa dữ liệu

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

Ví dụ 6.1:

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*.

6.2.2. 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 6.2:

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 6.3:

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 6.4:

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 6.5: 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*.

6.3. 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.

6.3.1. Định nghĩa phụ thuộc hàm

Cho lược đồ quan hệ $R=(A_1, A_2, \dots, A_n)$ và X, Y là các tập con của $\{A_1, A_2, \dots, A_n\}$. 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 6.6: 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:

$MaVT \rightarrow TenVT, DonGia$

$X \qquad \qquad Y$

Ví dụ 6.7: 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 6.8: 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 6.9: Xét lược đồ quan hệ KháchHang(MaKhach, HoTen, DiaChi)

MaKhach \rightarrow HoTen, DiaChi

MaKhach, HoTen \rightarrow HoTen, DiaChi (Dư thừa)

6.3.2. Các qui tắc phụ thuộc hàm

6.3.2.1. 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 $A \rightarrow C$.

6.3.2.2. 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 6.10:

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 6.11:

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): $AB \Rightarrow GH$

6.4. 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 6.12: $\{A \rightarrow B, B \rightarrow C\} \models A \rightarrow C$

6.5. Bao đóng của tập phụ thuộc hàm

6.5.1. 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 6.13: 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:

- | | | | |
|--------------------------------|-----------------------------|-----------------------------|-----------------------------|
| $A, B, C \rightarrow A, B, C;$ | $A, B, C \rightarrow A, B;$ | $A, B, C \rightarrow A, C;$ | $A, B, C \rightarrow B, C;$ |
| $A, B, C \rightarrow A;$ | $A, B, C \rightarrow B;$ | $A, B, C \rightarrow C;$ | |
| $A, B \rightarrow A, B, C;$ | $A, B \rightarrow A, B;$ | $A, B \rightarrow A, C;$ | $A, B \rightarrow B, C;$ |
| $A, B \rightarrow A;$ | $A, B \rightarrow B;$ | $A, B \rightarrow C;$ | |
| $A, C \rightarrow A, B, C;$ | $A, C \rightarrow A, B;$ | $A, C \rightarrow A, C;$ | $A, C \rightarrow B, C;$ |
| $A, C \rightarrow A;$ | $A, C \rightarrow B;$ | $A, C \rightarrow C;$ | |
| $A \rightarrow A, B, C;$ | $A \rightarrow A, B;$ | $A \rightarrow A, C;$ | $A \rightarrow B, C;$ |

$A \rightarrow A;$ $A \rightarrow B;$ $A \rightarrow B;$ $A \rightarrow C$

(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 6.14:

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 $A \rightarrow D \in F^+$

6.5.2. 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 6.15:

Cho $F = \{A \rightarrow B, AC \rightarrow B, B \rightarrow C\}$

$AC \rightarrow B$ là không đầy đủ vì $A \rightarrow B$ ($A \subset AC$)

6.5.3. 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 6.16:

Cho $F = \{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$

$A \rightarrow C$ là dư thừa vì $A \rightarrow B$ và $B \rightarrow C \Rightarrow A \rightarrow C$

6.5.4. 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 $G \subset F^+$.

6.6. Bao đóng của tập thuộc tính

6.6.1. 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 6.17: 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 6.18: Cho bảng Chứng từ vật tư có các trường như sau CTVT(A, B, C, D, E, F)

$$A \rightarrow B, C$$

$$C \rightarrow D$$

$$A, C, E \rightarrow F$$

$$X = \{A, C, E\}$$

$$X^+ = \{A, B, C, D, E, F\} = \text{CTVT}$$

6.6.2. 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 6.19:

Cho $R=ABCDEF$

$F = \{A \rightarrow BC, C \rightarrow D, AC \rightarrow F\}$, $X = ACE$. Hãy tính X^+

Ta có:

$$X_0 = ACE$$

$$X_1 = ACEB \text{ vì } A \rightarrow BC$$

$$X_2 = ABCED \text{ vì } C \rightarrow D$$

$$X_3 = ABCDE \text{ vì } ACE \rightarrow F$$

$$\text{Vậy } X^+ = ABCDEF$$

6.7. Khóa và siêu khóa

Cho lược đồ quan hệ $R=(A_1,...,A_n)$ và tập phụ thuộc hàm F trên R .

Tập con $X \subset \{A_1,...,A_n\}$ là khóa của R nếu $X \rightarrow A_1,...,A_n \in F^+$ là phụ thuộc hàm nguyên tố.

Tập $S \subset \{A_1,...,A_n\}$ 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 6.20: 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.

Ví dụ 6.21: Cho $R=ABCDEG$ và $F = \{AE \rightarrow C, CG \rightarrow A, BD \rightarrow G, GA \rightarrow E\}$

$K = ABD$ là siêu khoá của R

$(ABD)^+ = ABDGEC$

Ví dụ 6.22: 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 $(BC)^+ = ABCDEG$.

6.8. Phép tách lược đồ quan hệ

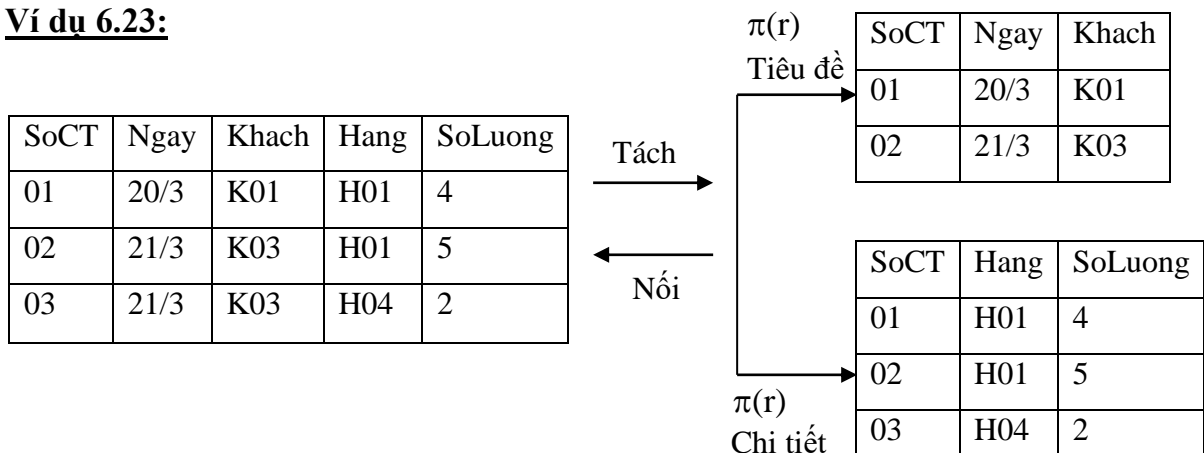
6.8.1. Định nghĩa

Cho lược đồ quan hệ $R = A_1A_2...A_n$. Tách lược đồ quan hệ R là thay thế R bằng các lược đồ con $R_1, R_2, ..., R_m$ sao cho $R_1 \cup R_2 \cup ... \cup R_m = R$ và $R_i \neq R_j$ khi $i \neq j$

6.8.2. Phép tách bảo toàn thông tin

$\rho(R) = (R_1, R_2, ..., R_m)$ bảo toàn thông tin $\Leftrightarrow \forall r(R) = \pi_{R_1}(r) * \pi_{R_2}(r) * ... * \pi_{R_m}(r)$

Ví dụ 6.23:



6.8.3. 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_i nếu $A_j \in R_i$, ngược lại ký hiệu $b(i,j)$

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 6.24: 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
ABC	a ₁	a ₂	a ₃	b ₁₄	b ₁₅
ADE	a ₁	b ₂₂	b ₂₃	a ₄	a ₅

A→BC
→

	A	B	C	D	E
ABC	a ₁	a ₂	a ₃	b ₁₄	b ₁₅
ADE	a ₁	a ₂	a ₃	a ₄	a ₅

Vậy $\rho(R)$ bảo toàn thông tin

Ví dụ 6.25: Cho $R = ABCD$ và $F = \{A \rightarrow B, AC \rightarrow D\}$

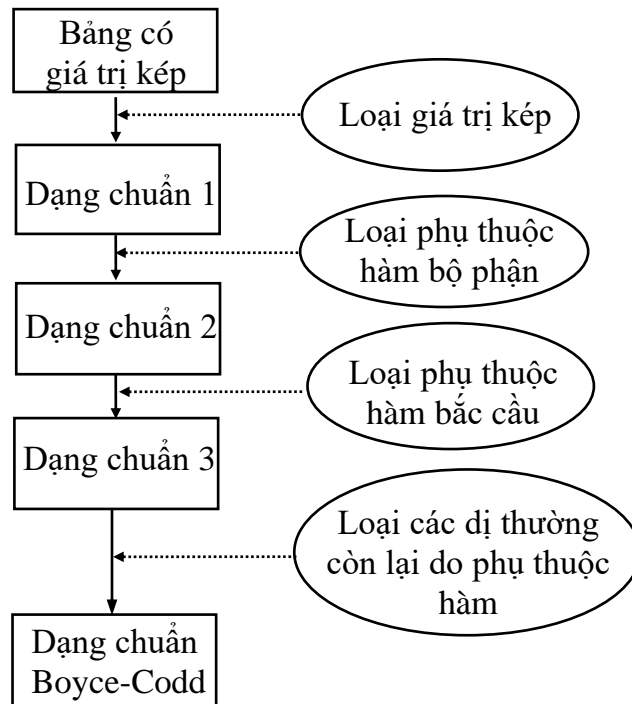
$\rho(R) = (AB, ACD)$ có bảo toàn thông tin hay không?

	A	B	C	D	$A \rightarrow B$ →		A	B	C	D
AB	a ₁	a ₂	b ₁₃	b ₁₄		AB	a ₁	a ₂	b ₁₃	b ₁₄
ACD	a ₁	b ₂₂	a ₃	a ₄		ACD	a ₁	a ₂	a ₃	a ₄

Vậy $\rho(R)$ bảo toàn thông tin

6.9. 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 6.1 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 6.1). 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, Boyce-Codd còn có dạng chuẩn 4 và dạng chuẩn 5, tuy nhiên chúng ta sẽ không nghiên cứu trong phạm vi giáo trình này.

6.9.1. 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 6.26:

Xét quan hệ:

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, TOANROIRAC, COSODULIEU.

Chúng ta có thể đưa quan hệ trên về dạng chuẩn 1 như sau:

MASV	HOVATEN	KHOA	TENMONHOC	DIEMTHI
99023	NGUYENTHITHU	CONG NGHE THÔNG TIN	KY THUAT LAP TRINH	6
99023	NGUYENTHITHU	CONG NGHE THÔNG TIN	TOAN ROI RAC	8
99023	NGUYENTHITHU	CONG NGHE THÔNG TIN	CO SO DU LIEU	4
99030	LE VAN THANH	DIEN TU	VI XULY	4

Chú ý rằng khi xét các dạng chuẩn, nếu chúng ta không nói gì thêm, ta hiểu dạng chuẩn đang xét ít nhất là ở dạng chuẩn 1.

Chú ý: Cách chuyển sang dạng 1NF:

- Điền đủ dữ liệu vào các cột
- Biến cột đa trị thành các dòng

6.9.2. 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 6.27: 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 về 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 6.28:

Cho lược đồ quan hệ HocVien(MaHV, MaMH, TenHV, Diem)

$MaHV \rightarrow TenHV$

$MaHV, MaMH \rightarrow Diem$

Với các thuộc tính không khoá TenHV, Diem thì TenHV phụ thuộc một phần của khoá 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 khoá chính gồm 2 thuộc tính trở lên.
- Để chuyển quan hệ từ dạng 1NF sang dạng 2NF, chúng ta dùng phép chiếu.

Ví dụ 6.29:

Bảng R gồm các thuộc tính MF, Tenfim, NSX, Giathue, HSX, NPP, MaKH, TenKH, Diachi, Ngaydat. R có các phụ thuộc hàm sau:

$MF \rightarrow Tenfim, NSX, Giathue, HSX, NPP$

$MaKH \rightarrow TenKH, Diachi$

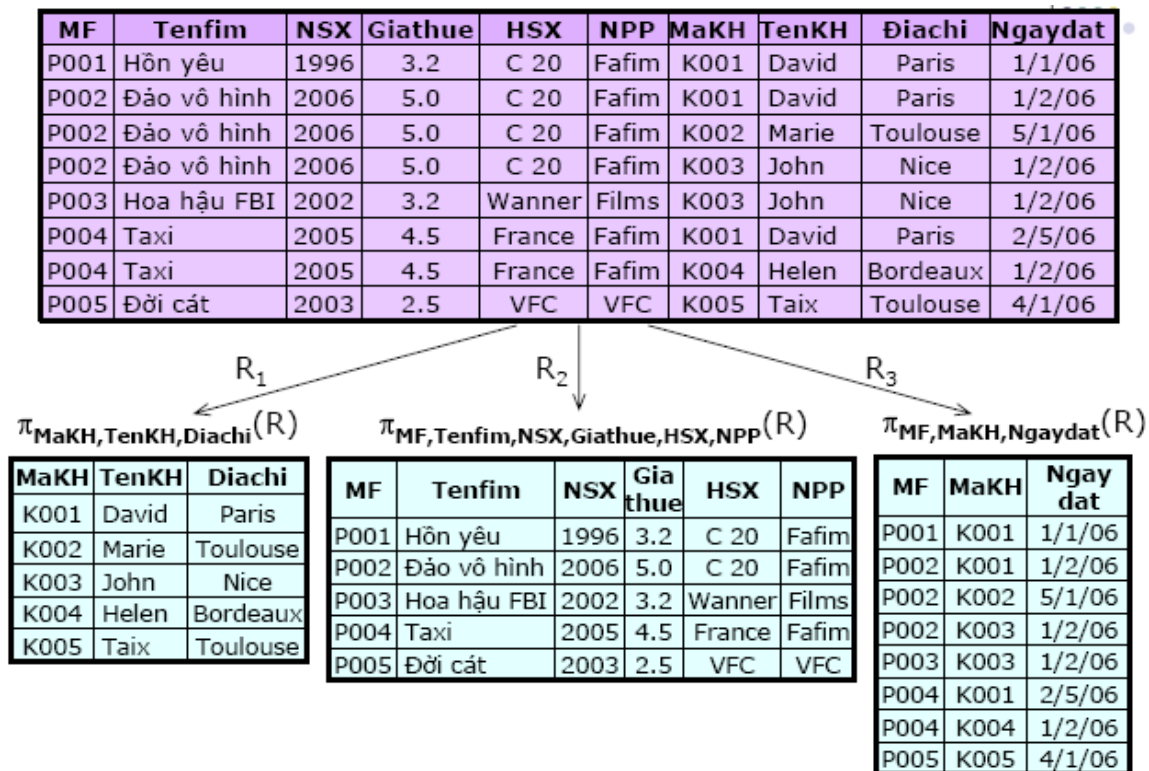
$MF, MaKH \rightarrow Ngaydat$

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

→ R không đạt chuẩn 2

Để chuyển về dạng chuẩn 2, chúng ta thường sử dụng phép chiếu. Chẳng hạn với Ví dụ 6.29 có thể sử dụng phép chiếu như sau (Hình 6.2):



Hình 6.2. Sử dụng phép chiếu tách về 2NF

6.9.3. Dạng chuẩn thứ 3 (3NF)

6.9.3.1. Định nghĩa

Phụ thuộc hàm $X \rightarrow A$ gọi là phụ thuộc hàm bắc 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 6.30:

- 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.

6.9.3.2. Thuật toán đưa về dạng chuẩn 3 bảo toàn thông tin

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 $X \rightarrow A_1A_2 \dots A_n$

Bước 3: Mọi phụ thuộc hàm chuyển thành một lược đồ con

Ví dụ 6.31:

$R = ABCDEGHIJ$ 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 6.32:

$R = \text{MTGPSL}$, $F = \{M \rightarrow T, GP \rightarrow M, GT \rightarrow P, MS \rightarrow L, GS \rightarrow P\}$

Khoá của R là GS

Ta có:

$M \rightarrow T \Rightarrow R_1 = \text{MT}$, $R = \text{MGPSL}$

$MS \rightarrow L \Rightarrow R_2 = \text{MSL}$, $R = \text{MGPS}$

$GP \rightarrow M \Rightarrow R_{23} = \text{GPM}$, $R = \text{GPS}$

$\Rightarrow \rho(R) = (\text{MT}, \text{MSL}, \text{GPM}, \text{GPS})$

6.9.4. Dạng chuẩn Boyce-Codd (BCNF)

6.9.4.1. Định nghĩa

Lược đồ quan hệ $s = \langle R, F \rangle$ được gọi là lược đồ dạng chuẩn Boyce - Codd (BCNF), nếu với mọi phụ thuộc $X \rightarrow Y \in F^+$, thì khi đó hoặc $Y \subseteq X$ (phụ thuộc tầm thường), hoặc X là một khoá của lược đồ quan hệ. Tức là nếu $X \rightarrow Y \in F^+$, $Y \not\subseteq X$ thì $X^+ = R$.

Từ định nghĩa trên có thể suy ra rằng:

- Các thuộc tính không khoá phụ thuộc hoàn toàn vào khoá.
- Các thuộc tính khoá phụ thuộc hoàn toàn vào tất cả khoá khác.

Ví dụ 6.33:

Lược đồ của quan hệ PROJ(PNO, PNAME, BUDGET) chỉ có phụ thuộc hàm duy nhất $PNO \rightarrow (PNAME, BUDGET)$, vậy nó ở dạng chuẩn Boyce-Codd.

Chú ý:

- Một quan hệ ở BCNF thì cũng đạt 3NF.
- Trong thực hành các quan hệ đạt chuẩn 3NF là đủ. Tuy nhiên một quan hệ ở 3NF không đảm bảo đã loại bỏ được tất cả các lỗi khi thao tác dữ liệu.

6.9.4.2. Thuật toán đưa về dạng chuẩn Boyce-Codd bảo toàn thông tin

- Đầu vào: $\langle R, F \rangle$
- Đầu ra: $\rho(R)$ thoả BCNF bảo toàn thông tin
- Phương pháp:

Phương pháp chủ yếu của thuật toán là tách lược đồ $s = \langle R, F \rangle$ thành 2 lược đồ.

Chọn bất kỳ $X \rightarrow A \in F^+$ sao cho X không là khoá và $A \notin X$. Khi đó lược đồ có tập các thuộc tính XA sẽ có dạng chuẩn BCNF và phụ thuộc hàm $X \rightarrow A$ sẽ thoả trên nó. Lược đồ thứ 2 có tập các thuộc tính $R \setminus A$. Hiển nhiên, khi kết nối lược đồ có tập thuộc tính $R \setminus A$ với lược đồ có tập thuộc tính XA không tổn thất thông tin. Tiếp tục tách $R \setminus A$ cho đến trở thành lược đồ có dạng chuẩn BCNF.

Ví dụ 6.34:

Cho $\Omega = \text{CTHRSG}$, trong đó:

C : Khoa học, T: Thầy giáo, H: Giờ học R: Phòng học, S : Sinh viên G: Lớp

Biết rằng:

- Mỗi khoá học chỉ có một thầy dạy.
- Một phòng học tại giờ xác định chỉ có một khoá học.
- Thầy dạy tại giờ học cụ thể xác định phòng học cụ thể.
- Khoá học với một sinh viên cụ thể xác định lớp học cụ thể.
- Mỗi một sinh viên học trong một giờ xác định tại phòng học cụ thể.

Khi đó $F = \{C \rightarrow T, HR \rightarrow C, HT \rightarrow R, CS \rightarrow G, HS \rightarrow R\}$.

Hiển nhiên, $s = \langle \Omega, F \rangle$ không là Boyce Codd, khoá của nó là thuộc tính HS.

Bước 1: Xét $CS \rightarrow G$: CS không phải là khóa, có thể tách $s = \langle \Omega, F \rangle$ thành 2 lược đồ quan hệ có dạng như sau:

$$\begin{aligned} s_1 &= \langle \Omega_1, F_1 \rangle & s_2 &= \langle \Omega_2, F_2 \rangle \\ \Omega_1 &= \{ \underline{C}, \underline{S}, G \} & \Omega_2 &= \{ C, T, \underline{H}, \underline{S}, R \} \\ F_1 &= \{ CS \rightarrow G \} & F_2 &= \{ C \rightarrow T, HR \rightarrow C, HT \rightarrow R, HS \rightarrow R \}. \end{aligned}$$

Sau bước 1, $s_1 = \langle \Omega_1, F_1 \rangle$ ở dạng Boyce Codd, $s_2 = \langle \Omega_2, F_2 \rangle$ ở dạng 3NF nhưng vẫn chưa ở dạng Boyce Codd.

Bước 2: Xét $C \rightarrow T$: T không phải là thuộc tính khóa, tách $s_2 = \langle \Omega_2, F_2 \rangle$ thành 2 lược đồ quan hệ sau:

$$\begin{aligned} s_{21} &= \langle \Omega_{21}, F_{21} \rangle & s_{22} &= \langle \Omega_{22}, F_{22} \rangle \\ \Omega_{21} &= \{ \underline{C}, T \} & \Omega_{22} &= \{ C, \underline{H}, \underline{S}, R \} \\ F_{21} &= \{ C \rightarrow T \} & F_{22} &= \{ HR \rightarrow C, HS \rightarrow R \}. \end{aligned}$$

Bước 3: Xét $HR \rightarrow C$: HR không phải là thuộc tính khóa, tách $s_{22} = \langle \Omega_{22}, F_{22} \rangle$ thành 2 lược đồ quan hệ sau:

$$S_{221} = \langle \Omega_{221}, F_{221} \rangle \quad s_{222} = \langle \Omega_{222}, F_{222} \rangle$$

$$\Omega_{221} = \{ C, \underline{H}, \underline{R} \} \quad \Omega_{222} = \{ \underline{H}, \underline{S}, R \}$$

$$F_{221} = \{ HR \rightarrow C \} \quad F_{222} = \{ HS \rightarrow R \},$$

Như vậy, từ lược đồ quan hệ $s = \langle \Omega, F \rangle$ chưa được chuẩn hoá, trong đó

$$\Omega = \{ C, T, H, R, S, G \},$$

$$F = \{ C \rightarrow T, HR \rightarrow C, HT \rightarrow R, CS \rightarrow G, HS \rightarrow R \}$$

Thuật toán đã cho một phép tách $\rho [\Omega_1, \Omega_{21}, \Omega_{221}, \Omega_{222}]$ lược đồ quan hệ $s = \langle \Omega, F \rangle$ về nhóm các lược đồ quan hệ chiếu $s_1 = \langle \Omega_1, F_1 \rangle$, $s_{21} = \langle \Omega_{21}, F_{21} \rangle$, $s_{221} = \langle \Omega_{221}, F_{221} \rangle$ và $s_{222} = \langle \Omega_{222}, F_{222} \rangle$ ở dạng chuẩn BCNF khi kết nối không tổn thất thông tin.

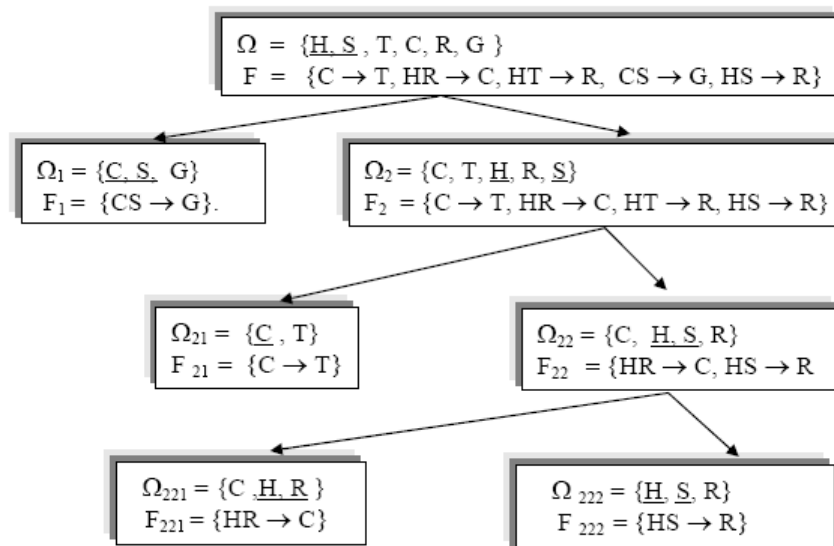
$$\Omega_1 = \{ \underline{C}, \underline{S}, G \}, F_1 = \{ CS \rightarrow G \}$$

$$\Omega_{21} = \{ \underline{C}, T \}, F_{21} = \{ C \rightarrow T \}$$

$$\Omega_{221} = \{ C, \underline{H}, \underline{R} \}, F_{221} = \{ HR \rightarrow C \}$$

$$\Omega_{222} = \{ \underline{H}, \underline{S}, R \}, F_{222} = \{ HS \rightarrow R \}$$

Sơ đồ tóm tắt quá trình tách như sau:



Hình 6.2. Quá trình tách về BCNF

TỔNG KẾT CHƯƠNG 6

- Một lược đồ quan hệ có cấu trúc tốt là lược đồ không chứa đựng sự dư thừa dữ liệu.
- Dư thừa dữ liệu là sự trùng lặp thông tin trong cơ sở dữ liệu.
- Các dị thường cập nhật dữ liệu: dị thường cập nhật dữ liệu, dị chen bộ, dị thường xoá bộ, dị thường sửa bộ.
- Quan hệ gọi là ở dạng chuẩn thứ nhất hay quan hệ chuẩn hoá nếu miền giá trị của mỗi thuộc tính chỉ chứa những giá trị nguyên tử, tức là không phân chia được nữa. Như vậy mỗi giá trị trong quan hệ cũng là nguyên tử.
- Một lược đồ quan hệ gọi 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.
- 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 bắc cầu.
- Lược đồ quan hệ $s = \langle R, F \rangle$ được gọi là lược đồ dạng chuẩn Boyce - Codd (BCNF), nếu với mọi phụ thuộc $X \rightarrow Y \in F^+$, thì khi đó hoặc $Y \subseteq X$ (phụ thuộc tầm thường), hoặc X là một khoá của lược đồ quan hệ. Tức là nếu $X \rightarrow Y \in F^+$, $Y \not\subseteq X$ thì $X^+ = R$.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 6

6.1. Nêu các mục đích của việc chuẩn hoá

6.2. Cho ví dụ về dư thừa dữ liệu

6.3. Cho ví dụ về cập nhật dị thừa

6.4. Cho ví dụ về độc lập chức năng

6.5. Nêu các bước quy trình chuẩn hoá

6.6. Kể tên các loại chuẩn hoá và nêu ví dụ cụ thể

6.7. Cho $F = \{AB \rightarrow E, AG \rightarrow I, E \rightarrow G, GI \rightarrow H\}$.

Chứng minh rằng $AB \rightarrow GH \in F^+$?

6.8. 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 $AB \rightarrow G \in F^+$?

6.9. Cho $F = \{AB \rightarrow E, AG \rightarrow I, BE \rightarrow I, E \rightarrow G, GI \rightarrow H\}$

Chứng minh rằng $AB \rightarrow GH \in F^+$.

6.10. 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 $AB \rightarrow G \in F^+$.

6.11. 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 $X = \{AB\}, \{BC\}, \{CD\}, \{ACD\}, \{DEG\}$.

6.12. 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 $AB \rightarrow E \in F^+$.

6.13. 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. $Q = ABCDEGHI$

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

6.14. Cho lược đồ $R = ABCDEGH$ và $F = \{C \rightarrow AB, D \rightarrow E, B \rightarrow G\}$.

Xác định xem lược đồ ở dạng chuẩn nào?

CHƯƠNG 7

AN TOÀN VÀ TOÀN VẸN DỮ LIỆU

Chương này của cuốn sách trình bày hai vấn đề liên quan với nhau là an toàn và toàn vẹn dữ liệu trong một hệ CSDL. Trong bất kỳ một hệ quản trị CSDL đầy đủ nào, chúng ta đều tìm thấy các công cụ để đảm bảo tính đúng đắn của các dữ liệu được lưu trữ trong CSDL, và ngăn chặn việc đọc ra hay phá hủy các dữ liệu trong CSDL một cách bất hợp pháp từ người sử dụng (NSD). Có hai nguyên nhân dẫn đến dữ liệu không đúng đắn: các sự cố (chẳng hạn như đánh sai dữ liệu hay lỗi lập trình) và việc sử dụng bất thường CSDL. Chúng ta có thể tách vấn đề bảo vệ CSDL chống lại những truy nhập sửa đổi hay phá hủy thành hai vấn đề: đảm bảo tính an toàn và toàn vẹn dữ liệu.

7.1. An toàn dữ liệu

Chúng ta sử dụng thuật ngữ "an toàn" để nói đến sự bảo vệ dữ liệu trong CSDL chống lại sự truy cập, sửa đổi hay phá hủy bất hợp pháp.

Ví dụ 7.1:

Chúng ta bắt đầu bằng một ví dụ minh họa một chừng mực nào đó bản chất của vấn đề: Giả sử CSDL của một đơn vị cụ thể nào đó có chứa:

Một quan hệ EMP (nhân viên) với thuộc tính: E# (số hiệu nhân viên), ENAME (tên nhân viên), ADDRESS (địa chỉ), D# (số hiệu phòng), SAL (lương), YEAR (năm tăng lương lần cuối) và EVALUATION (đánh giá của trưởng phòng về công việc của nhân viên)

Một quan hệ DEPT (phòng) với các thuộc tính: D# (số hiệu phòng), DNAME (tên phòng), MGR (trưởng phòng), ALOCATION (vị trí phòng).

Khi đó, mỗi mệnh đề sau đây chỉ ra một mức truy cập đến CSDL đó và có thể được trao cho một nhóm NSD:

1. NSD được phép truy nhập không điều kiện đến toàn bộ CSDL và thực hiện bất kỳ thao tác nào trên CSDL.

2. NSD không được phép truy nhập đến bất kỳ bộ phận nào của CSDL và không được phép thực hiện bất kỳ thao tác nào trên CSDL đó.

3. NSD có thể đọc một bộ phận của CSDL nhưng không được thay đổi nội dung của bộ phận đó.

4. NSD có thể đọc đúng một bộ trong quan hệ EMP (là bộ riêng của chính NSD) nhưng không được sửa đổi nó.

5. NSD có thể đọc đúng một bộ trong quan hệ EMP (là bộ riêng của chính NSD) và sửa đổi không phải tất cả các giá trị trong đó.

6. NSD chỉ được đọc các thuộc tính E#, ENAME, ADDRESS, và D#, nhưng trong mỗi bộ chỉ được sửa đổi các giá trị ENAME và ADDRESS.

7. NSD có thể đọc các thuộc tính E# và SAL, nhưng trong mỗi bộ chỉ được sửa đổi giá trị của SAL nhưng chỉ được làm trong khoảng thời gian từ 5 giờ sáng tới 5 giờ chiều từ một thiết bị đầu cuối đặt trong phòng tài vụ.

8. NSD có thể đọc các thuộc tính E# và SAL và trong mỗi bộ có thể sửa đổi giá trị của SAL nếu và chỉ nếu giá trị hiện tại của SAL nhỏ hơn 5000\$.

9. NSD có thể áp dụng các phép toán thống kê cho thuộc tính SAL (chẳng hạn để tính lương trung bình cho từng phòng) nhưng không được đọc hay sửa đổi các giá trị cá biệt.

10. NSD có thể đọc (xem) các thuộc tính E# và EVALUATION và trong mỗi bộ có thể sửa đổi giá trị của EVALUATION về các nhân viên của một phòng nếu và chỉ nếu NSD là trưởng phòng của phòng đó.

11. NSD có thể bổ sung thêm một số nhân viên vào trong một phòng hay xóa một số nhân viên từ một phòng nào đó.

Tóm lại, NSD có thể được cấp một số quyền truy cập như sau:

- Quyền đọc dữ liệu: được phép đọc ra một số dữ liệu được lưu trữ trong CSDL.
- Quyền cập nhật dữ liệu: được phép sửa đổi một số giá trị dữ liệu nhưng không được xóa dữ liệu trong CSDL.
- Quyền bổ sung dữ liệu: được phép bổ sung thêm dữ liệu mới vào trong CSDL nhưng không thay đổi bất kỳ dữ liệu nào có sẵn trong CSDL.
- Quyền xóa dữ liệu: được phép xóa một số dữ liệu trong CSDL.

Ngoài quyền truy cập, NSD còn có thể được cho phép sửa đổi sơ đồ CSDL, bao gồm các quyền sau:

- Quyền tạo chỉ dẫn: được phép tạo các chỉ dẫn đối với các quan hệ trong CSDL.
- Quyền quản lý tài nguyên: được phép tạo thêm các quan hệ mới.
- Quyền thay đổi: được phép thêm, xóa một số thuộc tính trong một quan hệ.
- Quyền loại bỏ: loại bỏ một quan hệ trong CSDL.

Danh sách các quyền nêu trên, không có nghĩa đã vét cạn tất cả, có thể cho chúng ta một vài khái niệm về phạm vi của vấn đề và tính mềm dẻo cần thiết của sơ đồ an toàn tổng quát.

Trong các mục sau chúng ta sẽ xét chi tiết hơn các đặc điểm mà một sơ đồ an toàn cần phải có. Không có đặc điểm nào vượt ra ngoài khả năng của công nghệ hiện thời. Tuy nhiên, giá thành của việc cài đặt tất cả các đặc điểm đó hiện nay quá cao là một lý do khiến không có hệ thống hiện hành nào có tất cả đặc điểm đó.

Tuy nhiên, sự giảm giá thành và những nhân tố khác sẽ thúc đẩy các hệ trong tương lai tiến tới cung cấp một sơ đồ bảo vệ dữ liệu đầy đủ theo các nét được gợi ý trong các mục sau.

7.1.1. Xuất trình căn cước và xác minh người truy cập

Những NSD khác nhau tùy theo vai trò, vị trí, trách nhiệm và quyền hạn trong hệ thống sẽ có các quyền khác nhau đối với CSDL hay các bộ phận khác nhau của CSDL, chẳng hạn như quan hệ và các thuộc tính của quan hệ. Các quyền này bao gồm đọc, thêm, xóa hay sửa đổi CSDL.

Các hệ quản trị CSDL phải đảm bảo không cho phép NSD thực hiện một thao tác nào trên CSDL nếu không được phép. Đây là trách nhiệm của người quản trị CSDL. Có nghĩa là người quản trị phải:

a. Xác định cho hệ thống những quyền hay cụ thể những thao tác mà mỗi NSD được phép thực hiện.

b. Cung cấp một phương tiện cho NSD để hệ thống nhận biết được họ.

Sơ đồ đơn giản nhất và chung nhất để xác minh những NSD là dùng mật khẩu mà chỉ hệ thống và cá nhân NSD được biết.

Như vậy, trước khi truy nhập dữ liệu, NSD phải xuất trình căn cước nói rõ anh ta là ai (chẳng hạn, cung cấp một số hiệu thao tác viên, hay sử dụng các phiếu hoặc thẻ đọc được bằng máy). Một cách lý tưởng, họ cũng phải xác minh sự khai báo đó (chẳng hạn nhập mật khẩu hay trả lời câu hỏi nào đó của hệ thống).

7.1.2. Kiểm tra truy cập

Với mỗi một NSD, hệ CSDL sẽ quản lý một hồ sơ NSD được phát sinh từ việc xác định NSD do người quản trị cung cấp, và cho các chi tiết về các thủ tục xuất trình căn cước và xác minh, chi tiết về các thao tác mà NSD được phép thực hiện. Khi đó, hệ quản trị CSDL phải kiểm tra mỗi thao tác NSD có vi phạm các ràng buộc an toàn nào không và nếu có phải hủy bỏ thao tác đó.

Nhiều vấn đề liên quan đến tính an toàn không phải là duy nhất đối với các hệ CSDL mà phải được giải quyết bởi người thiết kế hệ điều hành. Do vậy, chúng ta chỉ đề cập đến một số kỹ thuật chung đối với tính an toàn của các hệ CSDL.

a. Phân quyền NSD: Nói chung, những NSD khác nhau phù hợp với các quyền khác nhau đối với các CSDL khác nhau hay các bộ phận khác nhau của CSDL, chẳng hạn như các quan hệ hay các thuộc tính. Các quyền này có thể bao gồm đọc các bộ phận của CSDL, thêm, xóa, hay sửa đổi CSDL.

b. Bảo trì và lan truyền quyền: Hệ thống cần phải bảo trì một danh sách các quyền được tuyên bố cho mỗi NSD trên mỗi phần được bảo vệ của CSDL. Một trong các quyền này có thể là quyền lan truyền các quyền cho những NSD khác.

7.1.3. Sử dụng các khung nhìn

Các khung nhìn, ngoài việc cho phép viết các chương trình ứng dụng dễ dàng hơn bởi cho phép định nghĩa lại CSDL mức logic theo cách nhìn của người viết chương trình ứng dụng và tăng cường tính độc lập dữ liệu mức logic, chúng còn phục vụ như một công cụ bảo vệ thuận lợi trong nhiều trường hợp.

7.1.4. Các lệnh an toàn dữ liệu

Các phần trên đã trình bày một sơ đồ bảo vệ an toàn đơn giản với các bước xuất trình căn cước, xác minh NSD và kiểm tra truy nhập, trong đó khung nhìn được xem như một cơ chế bảo vệ an toàn thích hợp và hiệu quả đối với các hệ CSDL. Trong phần này, chúng ta sẽ làm quen với các lệnh an toàn được hỗ trợ bởi SQL. Qua đó, chúng ta thấy được các hệ CSDL có khả năng đảm bảo được an toàn dữ liệu.

a. Tạo khung nhìn cho NSD

Dạng tổng quát:

CREATE VIEW <tên view> [(<d/s tên cột>)] AS <Câu truy vấn SQL>

Ví dụ 7.2:

Với CSDL gồm hai quan hệ:

EMP(E#, ENAME, ADDRESS, D#, SAL, YEAR, EVALUATION): quan hệ nhân viên gồm mã nhân viên (E#), tên (ENAME), địa chỉ (ADDRESS), Mã phòng (D#), lương (SAL), năm lên lương (YEAR), đánh giá (EVALUATION).

DEPT(D#, DNAME, ALOCATION): quan hệ về phòng ban gồm mã phòng (D#), tên phòng (DNAME), địa chỉ (ALOCATION).

Chúng ta tạo khung nhìn KT cho đối tượng NSD là trưởng phòng kỹ thuật như sau:

CREATE VIEW KT (TênNV, ĐChỉ, Lương, NămLênLương, ĐánhGiá)

AS

SELECT ENAME, ADDRESS, SAL, YEAR, EVALUATION

FROM EMP

WHERE D# IN

SELECT D# FROM DEPT WHERE DNAME='Kỹ thuật'

b. Tuyên bố và kiểm tra quyền truy nhập

Dạng tổng quát:

GRANT <danh sách các thao tác> ON <đối tượng> TO <danh sách NSD> [WITH GRANT OPTION]

Trong đó:

<danh sách các thao tác>: read (đọc), select (tìm kiếm), write (ghi), insert (thêm), update (sửa đổi), delete (xoá), create (tạo lập), run (thực hiện). Đây là các thao tác được cung cấp trong SQL.

<đối tượng>: tên bảng, tên khung nhìn hay tên chương trình ứng dụng nào đó.

<danh sách NSD>: tên một NSD, một nhóm hay một danh sách, chúng ta có thể sử dụng từ khóa public, world cho mọi NSD.

Từ khóa WITH GRANT OPTION đảm bảo cho phép NSD trong <danh sách NSD> có thể tiếp tục lan truyền các quyền vừa được tuyên bố cho những NSD khác.

Ví dụ 7.3:

Tuyên bố quyền cho ông Huy, trưởng phòng kỹ thuật:

GRANT read, select, write, update, create, run ON KT TO Huy

WITH GRANT OPTION

c. Huỷ bỏ quyền truy cập

Dạng tổng quát:

REVOKE <danh sách các thao tác> ON <đối tượng> FROM <danh sách NSD>

Ví dụ 7.4:

Huỷ bỏ các quyền đã tuyên bố cho ông Huy, trưởng phòng kỹ thuật:

REVOKE read, select, write, update, create, run ON KT FROM Huy

7.2. Toàn vẹn dữ liệu

Quy tắc toàn vẹn (integrity rule) là các ràng buộc đảm bảo trạng thái nhất quán của cơ sở dữ liệu.

Giữ gìn tính toàn vẹn của CSDL có thể xem là sự bảo vệ dữ liệu trong CSDL chống lại sự sửa đổi hay phá hủy không có căn cứ (khác với không hợp pháp). Như vậy, toàn vẹn khác với an toàn dù rằng hai lĩnh vực này liên quan rất mật thiết với nhau và thực sự có thể dùng chung một cơ chế để thực hiện bảo vệ cả hai, ít nhất tới một chừng mực nào đó. Giống như các ràng buộc an toàn, các ràng buộc toàn vẹn sẽ được lưu giữ trong từ điển dữ liệu như một phần của sơ đồ mức logic.

Nói chung, việc mất tính toàn vẹn có thể do những nguyên nhân sau:

- Hỏng hóc về phần cứng ở một chỗ nào đó của hệ thống (chẳng hạn ở bộ xử lý trung tâm, trên một kênh dữ liệu, hay ở một thiết bị vào/ra)
- Sai sót về phía người thao tác máy tính
- Sai sót về lập trình ở ứng dụng CSDL

Sau đây là một số loại quy tắc toàn vẹn:

- Toàn vẹn thực thể (Entity integrity)
- Toàn vẹn tham chiếu (Referential integrity)
- Các ràng buộc miền (Domains)
- Thao tác bẫy (Triggering operations).

7.2.1. Toàn vẹn thực thể

Quy tắc toàn vẹn thực thể yêu cầu thực thể phải có khóa chính, các thuộc tính khóa phải có giá trị duy nhất và khác null. Quy tắc này không cho phép hai bản ghi trùng khóa.

Ví dụ 7.5:

Xét cơ sở dữ liệu

EMP(ENO, ENAME, TITLE, SAL, PNO, RESP, DUR)

PROJ(PNO, PNAME, BUDGET)

Quy tắc toàn vẹn thực thể ràng buộc:

- Trong lược đồ PROJ, thuộc tính khóa PNO không thể nhận giá trị null và phải có giá trị không trùng nhau.
- Trong lược đồ EMP, cặp thuộc tính khóa (EMP, PNO) không thể nhận giá trị null và phải có giá trị không trùng nhau.

7.2.2. Toàn vẹn tham chiếu

Toàn vẹn tham chiếu là ràng buộc đảm bảo tính hợp lệ của sự tham chiếu của một đối tượng trong cơ sở dữ liệu (gọi là đối tượng tham chiếu) đến đối tượng khác (gọi là đối tượng được tham chiếu) trong cơ sở dữ liệu đó. Các thuộc tính tương ứng gọi thuộc tính cặp ghép của ràng buộc tham chiếu.

Ví dụ 7.6:

Xét các quan hệ

EMP(ENO, ENAME, TITLE, SAL, PNO, RESP, DUR)

PROJ(PNO, PNAME, BUDGET)

Với mỗi bộ $e \in \text{EMP}$ phải tồn tại bộ $p \in \text{PROJ}$ sao cho

$$e(\text{PNO}) = p(\text{PNO})$$

Thuộc tính PNO của quan hệ EMP là khóa ngoại tham chiếu đến khóa chính PNO của quan hệ PROJ: $\text{EMP.PNO} \rightarrow \text{PROJ.PNO}$

Quan hệ EMP là quan hệ tham chiếu và quan hệ PROJ là quan hệ được tham chiếu, với cặp ghép: (EMP.PNO, PROJ.PNO).

Quy tắc toàn vẹn tham chiếu được xét đến trong khi cập nhật quan hệ tham chiếu hoặc quan hệ được tham chiếu. Ta xét các quy tắc con sau.

- **Quy tắc chèn:** Không thể chèn hàng mới vào quan hệ tham chiếu nếu quan hệ được tham chiếu chưa có dữ liệu cặp ghép tương ứng.

Ví dụ 7.7:

Xét các quan hệ EMP và PROJ. Giả sử ta muốn chèn bản ghi

$$e = (E1, J.Doe, \text{Elect.Eng.}, 40000, P5, \text{Manager}, 20)$$

trong đó P5 là dự án Elect.Commerce (thương mại điện tử) với kinh phí 500000 USD.

Khi đó, nếu quan hệ PROJ chưa có bản ghi

$$p = (P5, \text{Elect.Commerce}, 500000)$$

thì quy tắc chèn đảm bảo không thể chèn bản ghi e vào quan hệ EMP được.

Muốn chèn bản ghi e vào quan hệ EMP, trước hết ta phải chèn bản ghi p vào quan hệ PROJ.

- **Quy tắc xóa:** Không thể xóa hàng của quan hệ được tham chiếu nếu hàng đó có dữ liệu cặp ghép tương ứng trong quan hệ tham chiếu.

Ví dụ 7.8:

Xét các quan hệ EMP và PROJ.

EMP

ENO	ENAME	TITLE	SAL	PNO	RESP	DUR
E1	J.Doe	Elect.Eng.	40000	P1	Manager	12
E2	M.Smith	Syst.Anal.	34000	P1	Analyst	24
E2	M.Smith	Syst.Anal.	34000	P2	Analyst	6

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Development	135000

P3	CAD/CAM	250000
----	---------	--------

Giả sử dự án P1 đã kết thúc và ta muốn xóa nó khỏi quan hệ PROJ. Tuy nhiên các bản ghi một và hai của EMP tham chiếu đến dự án P1, vì thế ta không thể xóa dự án P1 được.

Tuy nhiên, qui tắc toàn vẹn tham chiếu cho phép các phương án lựa chọn sau:

- 1) Restrict: Không cho xóa
- 2) Nullify: Gán giá trị null cho thuộc tính cặp ghép của các bản ghi của quan hệ tham chiếu tham chiếu đến bản ghi bị xóa.
- 3) Cascade: Xóa tất cả các bản ghi của quan hệ tham chiếu tham chiếu đến bản ghi bị xóa.

7.2.3. Các ràng buộc miền

Các ràng buộc miền là loại ràng buộc lên các giá trị hợp lệ của thuộc tính. Miền giá trị là tập hợp tất cả các loại dữ liệu và phạm vi giá trị được thuộc tính thừa nhận.

Định nghĩa miền giá trị xác định các tham số đặc trưng của thuộc tính:

- 1) Kiểu dữ liệu (data type)
- 2) Độ dài (length)
- 3) Khuôn dạng (format)
- 4) Phạm vi (range)
- 5) Giá trị cho phép (allowable values)
- 6) Ý nghĩa (meaning)
- 7) Tính duy nhất (uniqueness)
- 8) Chấp nhận giá trị null (null support).

Ví dụ 7.9:

Xét quan hệ

PROJ(PNO, PNAME, BUDGET)

Các thuộc tính PNAME và BUDGET có ràng buộc miền giá trị như sau

Tên:	PNAME	BUDGET
Ý nghĩa:	Tên dự án	Kinh phí dự án
Kiểu dữ liệu:	Ký tự (Character)	Số (numeric)
Độ dài:	20	10
Định dạng (Format):		
Phạm vi:		> 0
Giá trị cho phép:		
Duy nhất:	Có	Không
Trợ giúp rỗng (Null support):	Non-null	Null

7.2.4. Thao tác bất

Thao tác bất là qui tắc yêu cầu tính hợp pháp của dữ liệu trong các tác nghiệp cập nhật như xoá, chèn và sửa.

Thao tác bất có thể liên quan đến các thuộc tính của một quan hệ hoặc nhiều quan hệ. Các ràng buộc phức tạp thường được phát biểu dạng thao tác bất.

Một thao tác bất thường có các thành phần sau:

- 1) Qui tắc người dùng: là yêu cầu ngắn gọn của ràng buộc
- 2) Sự kiện: là các thao tác xử lý dữ liệu (chèn, sửa hoặc xoá) kích hoạt thao tác bất
- 3) Tên thực thể: tên các thực thể liên quan
- 4) Điều kiện: là các lý do dẫn đến việc các bất thao tác
- 5) Hành động: là công việc thực thi khi thao tác được bất

Ví dụ 7.10:

Cho thực thể

NHANVIEN(Manv, HoTen, NgaySinh, NgayBC, ...).

Hiển nhiên là NgayBC (ngày vào biên chế) không được sớm hơn NgaySinh. Ta có thể đảm bảo điều kiện này bằng thao tác bất sau:

Qui tắc người dùng: NgayBC không sớm hơn NgaySinh

Sự kiện: Chèn, Sửa

Tên thực thể: NHANVIEN

Điều kiện: NgayBC < NgaySinh

Hành động: Phủ nhận thao tác cập nhật

Ví dụ 7.11: Xét hai thực thể

KHACH(Makhach, TenKhach, TaiKhoan, SoDu)

THANHTOAN(MaKhach, SoTien)

Ta thấy rằng SoTien của THANHTOAN không thể vượt quá SoDu của KHACH. Ta có thể đảm bảo điều kiện này bằng thao tác bất sau:

Qui tắc người dùng: SoTien không lớn hơn SoDu

Sự kiện: Chèn, Sửa

Tên thực thể: THANHTOAN, KHACH

Điều kiện: THANHTOAN.SoTien > KHACH.SoDu

Hành động: Phủ nhận thao tác cập nhật

TỔNG KẾT CHƯƠNG 7

- Chúng ta sử dụng thuật ngữ "an toàn" để nói đến sự bảo vệ dữ liệu trong CSDL chống lại sự truy cập, sửa đổi hay phá hủy bất hợp pháp.
- Các đặc điểm mà một sơ đồ an toàn cần có: xuất trình căn cước và xác minh người truy cập, kiểm tra truy cập, các khung nhìn như các cơ chế bảo vệ, các lệnh an toàn dữ liệu trong SQL.
- Cơ chế xuất trình căn cước và xác minh NSD của các hệ QTCSDL đảm bảo không cho phép NSD thực hiện một thao tác nào đó trên CSDL nếu không được phép.
- Cơ chế kiểm tra truy cập của hệ QTCSDL dùng để kiểm tra mỗi thao tác của NSD có vi phạm các ràng buộc an toàn nào đó không và nếu có thì phải huỷ bỏ thao tác đó.
- Sử dụng các khung nhìn như các cơ chế bảo vệ có tác dụng cho phép những NSD nào đó có thể thấy dữ liệu nhưng không thể thay đổi dữ liệu để đảm bảo an toàn cho dữ liệu.
- Các lệnh an toàn được hỗ trợ bởi SQL bao gồm lệnh tạo khung nhìn cho NSD, tuyên bố và kiểm tra quyền truy cập, huỷ bỏ quyền truy cập.
- Quy tắc toàn vẹn (integrity rule) là các ràng buộc đảm bảo trạng thái nhất quán của cơ sở dữ liệu .
- Có các loại quy tắc toàn vẹn sau: Toàn vẹn thực thể (Entity integrity), Toàn vẹn tham chiếu (Referential integrity), Miền giá trị (Domains), Thao tác bất kỳ (Triggering operations).
- Quy tắc toàn vẹn thực thể yêu cầu thực thể phải có khóa chính, các thuộc tính khóa phải có giá trị duy nhất và khác null. Quy tắc này không cho phép hai bản ghi trùng khóa.
- Toàn vẹn tham chiếu là ràng buộc đảm bảo tính hợp lệ của sự tham chiếu của một đối tượng trong cơ sở dữ liệu (gọi là đối tượng tham chiếu) đến đối tượng khác (gọi là đối tượng được tham chiếu) trong cơ sở dữ liệu đó. Các thuộc tính tương ứng gọi thuộc tính cặp ghép của ràng buộc tham chiếu.
- Các ràng buộc miền là loại ràng buộc lên các giá trị hợp lệ của thuộc tính. Miền giá trị là tập hợp tất cả các loại dữ liệu và phạm vi giá trị được thuộc tính thừa nhận.
- Thao tác bất kỳ là quy tắc yêu cầu tính hợp pháp của dữ liệu trong các tác nghiệp cập nhật như xoá, chèn và sửa.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 7

7.1. Hãy phân tích sự giống nhau và khác nhau giữa an toàn và toàn vẹn dữ liệu.

7.2. NSD có thể được cấp các quyền gì khi truy cập vào CSDL.

7.3. Hãy phân tích các đặc điểm của một sơ đồ an toàn.

7.4. Nêu các nguyên nhân dẫn đến mất tính toàn vẹn dữ liệu.

7.5. Nêu các loại ràng buộc toàn vẹn.

7.6. Giả sử có một bài toán quản lý như sau:

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.

Hãy phân tích các mức truy cập có thể được trao cho một nhóm NSD.

7.7. Với CSDL gồm hai quan hệ:

EMP(E#, ENAME, ADDRESS, D#, SAL, YEAR, EVALUATION)

DEPT(D#, DNAME, ALOCATION)

(Xem thêm ví dụ 7.1 để hiểu các quan hệ và các thuộc tính)

a. Hãy tạo khung nhìn cho đối tượng sử dụng là trưởng phòng KHQT (kế hoạch quản trị).

b. Tuyên bố các quyền đọc, tìm kiếm, ghi, thêm, sửa đổi, thực hiện đối với bà Thoa, trưởng phòng KHQT.

c. Huỷ bỏ các quyền đã tuyên bố đối với bà Thoa, trưởng phòng KHQT.

7.8. Cho các quan hệ sau:

Quan hệ KHOA (MAKHOA, TENKHOA)

Quan hệ LOPHOC (MALOP, TENLOP, MAKHOA, NIENKHOA, SOHOCVIEN).

Quan hệ HOCVIEN (MAHOCVIEN, TENHOCVIEN, NGAYSINH, QUEQUAN)

Hãy tìm các toàn vẹn thực thể, toàn vẹn tham chiếu, ràng buộc miền, thao tác bất.

CHƯƠNG 8

QUẢN LÝ GIAO DỊCH

Như chúng ta đã biết, một trong các chức năng của hệ quản trị CSDL là phải đảm bảo tính tin cậy, tính nhất quán của CSDL, cả khi có nhiều người dùng đồng thời truy cập vào CSDL, thậm chí cả khi xảy ra sự cố. Trong chương này chúng ta xem xét ba dịch vụ mà một hệ quản trị CSDL cần cung cấp: hỗ trợ các giao dịch, điều khiển tương tranh và khôi phục CSDL. Những dịch vụ đó phụ thuộc vào nhau.

Điều khiển tương tranh và khôi phục CSDL mỗi khi có sự cố đều nhằm bảo đảm sự nhất quán về dữ liệu và tránh tổn thất thông tin.

8.1. Hỗ trợ giao dịch

8.1.1. Giao dịch

Một giao dịch có thể là toàn bộ chương trình (viết trong ngôn ngữ thao tác dữ liệu bậc cao hay trong một ngôn ngữ lập trình SQL, COBOL, PASCAL,...), có thể là một phần chương trình hay một câu lệnh đơn lẻ (chẳng hạn lệnh INSERT hay UPDATE của SQL), nó có thể bao gồm một số thao tác trên CSDL (xem định nghĩa về giao dịch ở phần 5.2.6.1).

Trong ngữ cảnh CSDL, có thể xem sự thực hiện chương trình ứng dụng như một dãy các giao dịch và xen kẽ giữa chúng là các thao tác không đòi hỏi xử lý CSDL (thao tác của CPU).

Ví dụ 8.1:

Cho hai lược đồ quan hệ NHAN_VIEN và CHAM_CONG sau đây:

NHAN_VIEN(Hodem, Ten, MSNV, Ng-sinh, Diachi, Luong, MSTT, MSP)

CHAM_CONG(MSNV,MSDA,Giocong)

Trong đó:

Hodem: Họ đệm

Ten: Tên

MSNV: Mã số nhân viên

Ng-sinh: Ngày sinh

Diachi: Địa chỉ

Luong: Lương

MSTT: Mã số thủ trưởng trực tiếp

MSP: Mã số phòng quản lý nhân viên (MSP)

MSDA: Mã số dự án

Giocong: Số giờ công.

Một giao dịch đơn giản là nâng lương cho một nhân viên có mã số x thêm 10% lương hiện tại của anh ta. Chúng ta có thể viết giao dịch này như sau:

Read(MSNV=x, Luong)

*Luong_moi=Luong*1.1*

Write(MASNV=x, Luong_moi)

Trong đó:

Read(x) dùng để đọc một mục dữ liệu có tên là *x* ở CSDL vào một biến của chương trình. Việc thực hiện một lệnh *Read(x)* sẽ bao gồm các bước sau đây:

- Tìm địa chỉ của khối (trên đĩa) có chứa mục dữ liệu *x*.
- Sao chép khối đó vào vùng đệm của bộ nhớ chính (nếu nó chưa có sẵn trong vùng đệm của bộ nhớ chính).
- Sao chép mục dữ liệu *x* từ vùng đệm của biến có tên là *x* của chương trình.

Write(x) để ghi giá trị biến *x* của chương trình vào mục dữ liệu *x* của CSDL. Việc thực hiện một lệnh *Write(x)* sẽ bao gồm các bước sau đây:

- Tìm địa chỉ của khối (trên đĩa) có chứa mục dữ liệu *x*.
- Sao chép khối đó vào vùng đệm của bộ nhớ chính (nếu nó chưa có sẵn trong vùng đệm của bộ nhớ chính).
- Sao chép mục dữ liệu *x* từ biến chương trình có tên là *x* vào chỗ chính xác của nó trong vùng đệm.
- Lưu trữ khối vừa được cập nhật trong vùng đệm vào lại đĩa (ngay tức thời hoặc thời điểm muộn hơn).

Ví dụ 8.2:

Hãy xem xét một giao dịch phức tạp hơn. Giả sử có một giao dịch xoá bỏ bản ghi về nhân viên có *MSNV=x* khỏi quan hệ *NHAN_VIEN*. Do ràng buộc tham chiếu từ quan hệ *NHAN_VIEN* đến quan hệ *CHAM_CONG*, để có thể xoá bỏ bản ghi đó, trong *CHAM_CONG* cần gán lại một giá trị khác (chẳng hạn *y*, $y \neq x$) cho tất cả các bản ghi vốn có *MSNV=x*.

Giao dịch này được cho như sau:

Delete(MSNV=x)

Đối với mọi bản ghi của CHAM_CONG giá trị khóa là (msnv,msda)

begin

read(MSNV=msnv, MSDA=msda)

if (msnv=x) then begin

msnv=y

write(msnv=msnv, MSDA=msda)

end

end

Ở đây nếu việc cập nhật lại cho các bản ghi tương ứng trong *CHAM_CONG* không được tiến hành thì CSDL sẽ rơi vào trạng thái không nhất quán, vì phạm ràng buộc tham chiếu, một số bản ghi chấm công nói về một người không còn tồn tại trong CSDL nữa.

Rất có thể trong quá trình thực hiện dãy hành động của một giao dịch, một sự cố nào đó xảy ra và việc thực hiện giao dịch không thành công. Như vậy, việc thực hiện giao dịch có thể có kết quả thành công hoặc không thành công. Trong trường hợp thứ nhất, chúng ta nói rằng *giao dịch đã được chuyển giao*, trường hợp thứ hai nói rằng *giao dịch đã được huỷ bỏ*.

Khi giao dịch bị huỷ bỏ thì CSDL phải được khôi phục trạng thái nhất quán của nó trước khi bắt đầu thực hiện giao dịch này (còn gọi là giao dịch bị phục hồi). Một giao dịch bị huỷ bỏ có thể được khởi động thực hiện lại ngay sau đó và tùy thuộc vào nguyên nhân gây lỗi, nó vẫn có thể thực hiện thành công tại một thời điểm khác.

Những giao dịch đã được chuyển giao thì không thể huỷ bỏ. Nếu ta nhận thấy giao dịch này thực hiện có sai sót thì chỉ còn cách thực hiện một giao dịch khác điều chỉnh lại, huỷ đi tác động của giao dịch ngay trước đó đã gây ra sai sót.

Các hệ quản trị CSDL không thể tự hiểu được rằng những phép cập nhật nào sẽ được nhóm lại thành một giao dịch, bởi vậy người dùng phải *định biên* cho giao dịch bằng các từ khóa theo quy định của ngôn ngữ DML:

BEGIN TRANSACTION

COMMIT

ROLLBACK

Nếu không có sự định biên cho các giao dịch thì hệ thống sẽ coi toàn bộ chương trình là một giao dịch. Hệ quản trị CSDL sẽ tự động thực hiện COMMIT khi chương trình thực hiện xong không có lỗi nào xuất hiện và ROLLBACK trong trường hợp ngược lại.

8.1.2. Các tính chất của giao dịch

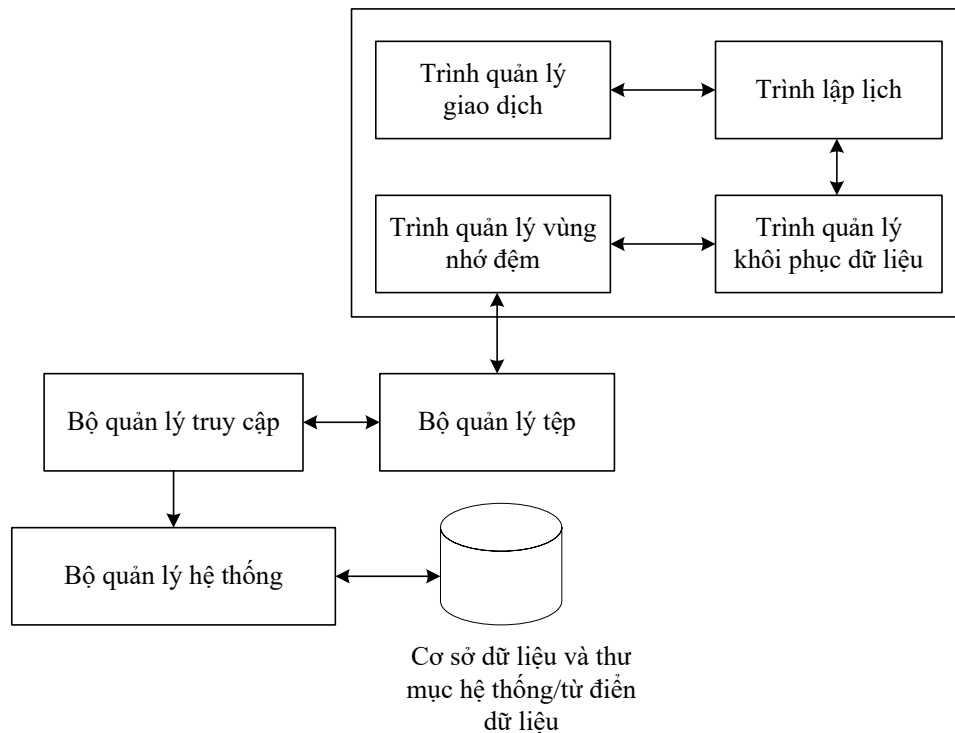
Để bảo toàn được tính nhất quán của dữ liệu, các hệ CSDL cần phải đảm bảo được tính chất sau đây cho các giao dịch:

- (1) Tính nguyên tử (*Atomicity*): Toàn bộ các thao tác trong một giao dịch đều được thực hiện hoặc không một giao dịch nào trong chúng được thực hiện cả. Một giao dịch là một đơn vị công việc không thể phân chia.
- (2) Tính nhất quán (*Consistency*): Một giao dịch phải chuyển CSDL từ trạng thái nhất quán này sang một trạng thái nhất quán khác.
- (3) Tính cô lập (*Isolation*): Các giao dịch phải được thực hiện một cách độc lập với nhau. Nói một cách khác, những tác động của một giao dịch sẽ không thể thấy được đối với những giao dịch khác, khi giao dịch này chưa được thực hiện xong hoàn toàn.
- (4) Tính bền vững (*Durability*): Những thay đổi của CSDL do tác động của một giao dịch thực hiện thành công là bền vững, không bị mất đi, kể cả khi có những lỗi xảy ra sau đó.

Các tính chất trên thường gọi là tính chất *ACID*.

Trong kiến trúc của một hệ quản trị CSDL, bộ quản lý giao dịch thay mặt các chương trình ứng dụng điều phối các giao dịch. Nó liên hệ với bộ lập lịch, nơi có trách nhiệm thực hiện một chiến lược điều khiển tương tranh. Nếu có sự cố xảy ra trong quá trình thực hiện giao dịch thì tính nhất quán của CSDL cần được bảo toàn, đây là trách

nhiệm của bộ quản lý khôi phục dữ liệu. Bộ quản lý vùng đệm có nhiệm vụ chuyển giao dữ liệu qua lại giữa thiết bị nhớ thứ cấp và bộ nhớ chính.



Hình 8.1. Kiến trúc của một hệ quản trị CSDL

8.2. Điều khiển tương tranh

8.2.1. Mục đích của việc điều khiển tương tranh

Một mục đích quan trọng đặt ra khi phát triển các hệ CSDL là sự cho phép nhiều người dùng truy cập đồng thời vào CSDL. Nếu tất cả các truy cập đồng thời đều chỉ là đọc dữ liệu thì không xảy ra vấn đề gì vì các giao dịch đọc không ảnh hưởng lẫn nhau. Tuy nhiên chỉ cần một trong các giao dịch có chứa thao tác cập nhật dữ liệu thì rất có thể chúng ảnh hưởng đến nhau, dẫn đến những kết quả không mong muốn.

Ví dụ 8.3: Tồn thất do cập nhật

Một phép cập nhật của một người dùng có vẻ đã hoàn thành nhưng có thể bị mất bởi cập nhật của một người dùng khác.

Thời điểm	T1	T2	sodu _x
t ₁		Begin_transaction	100
t ₂	Begin_transaction	Read(sodu _x)	100
t ₃	Read(sodu _x)	sodu _x =sodu _x +100	100
t ₄	sodu _x =sodu _x -10	Write(sodu _x)	200
t ₅	Write(sodu _x)	Commit	90
t ₆	Commit		90

Hình 8.2. Tồn thất do cập nhật

Trong hình 8.2, giao dịch T1 và T2 được thực hiện đồng thời.

Trước thời điểm thực hiện hai giao dịch T1 và T2, số dư tài khoản x là 100\$. Giao dịch T1 thực hiện việc rút 10\$ khỏi tài khoản x, trong khi giao dịch T2 lại đưa thêm vào đó 100\$. Nếu hai giao dịch này thực hiện tuần tự thì kết quả số dư trong tài khoản

là 190\$. Trong trường hợp bắt đầu thực hiện T1 và T2 gần như cùng thời điểm thì kết quả lại là 90\$.

Giải thích:

Tại thời điểm t_2 , T2 đọc số dư tài khoản x , thời điểm t_3 , T1 đọc số dư này và cả hai phép đọc đó đều cho kết quả là 100\$.

T2 tăng $sodu_x$ (số dư tài khoản x) thêm 100\$ và ghi kết quả cập nhật này vào CSDL.

Trong khi đó T1 giảm số dư đi 10\$ và ghi kết quả cập nhật này vào CSDL ngay sau thời điểm cập nhật của T2 và do vậy làm mất đi kết quả cập nhật của T2. Nếu chúng ta không cho phép T1 đọc giá trị của $sodu_x$ cho đến khi T2 ghi xong kết quả cập nhật thì đã không xảy ra tổn thất nói trên.

Ví dụ 8.4: Kết quả sai khi có giao dịch bị huỷ bỏ

Thời điểm	T3	T4	$sodu_x$
t_1		Begin_transaction	100
t_2		Read($sodu_x$)	100
t_3		$sodu_x = sodu_x + 100$	100
t_4	Begin_transaction	Write($sodu_x$)	200
t_5	Read($sodu_x$)		200
t_6	$sodu_x = sodu_x - 10$	Rollback	100
t_7	Write($sodu_x$)		190
t_8	Commit		190

Hình 8.3. Lỗi do có giao dịch bị huỷ bỏ

Trong hình 8.3, hai giao dịch T3 và T4 cũng được thực hiện gần như đồng thời. T3 rút 10\$ khỏi tài khoản x , T4 đưa thêm vào tài khoản này 100\$, số dư tài khoản x trước khi thực hiện cả hai giao dịch đó cũng là 100\$. Sự khác biệt với ví dụ trên là ở hai điểm sau:

T3 được bắt đầu thực hiện sau thời điểm T4 thực hiện ghi cập nhật (tăng số tiền trong tài khoản x lên 100\$), do vậy trước khi giảm $sodu_x$ T3 đọc được $sodu_x = 200$ và dùng giá trị này để tính giá trị mới cho $sodu_x$.

T4 đã không được chuyển giao mà bị huỷ bỏ tại thời điểm t_6 (rollback)

Đúng ra kết quả cuối cùng đối với việc thực hiện T3, T4 (thời điểm t_8) giá trị của $sodu_x$ phải là 90 chứ không phải 190\$. Nếu T4 không bị huỷ bỏ thì không xuất hiện lỗi này, 20\$ của $sodu_x$ do T3 đọc không có giá trị nữa khi T4 bị huỷ bỏ. Muốn tránh một lỗi kiểu như thế chúng ta phải ngăn cấm không cho T3 đọc $sodu_x$ cho đến khi có quyết định chuyển giao hay huỷ bỏ các động tác của T4.

8.2.2. Tính khả tuần tự của các lịch và việc sử dụng chúng

Mục đích của giao dịch điều khiển tương tranh là xếp lịch thực hiện sao cho không xảy ra sự tác động lẫn nhau giữa chúng. Có một giải pháp đơn giản: chỉ cho phép một giao dịch thực hiện tại một thời điểm, hiểu theo nghĩa là chỉ khi giao dịch này chuyển giao xong thì giao dịch tiếp theo mới được bắt đầu thực hiện. Nhưng mục đích của hệ QTCSDBL đa người dùng lại là tối ưu hoá sự thực hiện đồng thời trong hệ thống sao cho những giao dịch thực hiện đồng thời không ảnh hưởng lẫn nhau. Bởi

vậy những giao dịch truy cập đến những phần khác nhau của CSDL phải được sắp xếp trong một lịch thực hiện sao cho chúng không ảnh hưởng lẫn nhau.

Lịch là một dãy (có thứ tự) các thao tác của một tập các giao dịch tương tranh mà trong đó thứ tự của các thao tác trong mỗi một giao dịch được bảo toàn.

Như vậy, giả sử S là một lịch của tập giao dịch trong đó có sự tham gia của giao dịch T , nên thao tác thứ i được thực hiện trước thao tác j trong T thì trong lịch S thao tác i cũng phải đứng trước thao tác j . Nhưng chúng ta cũng thấy rằng không phải lịch nào cũng "đúng", nghĩa là không để xảy ra bất cứ sự ảnh hưởng lẫn nhau nào giữa các giao dịch tương tranh.

Lịch tuần tự là lịch mà trong đó các thao tác của mỗi một giao dịch được thực hiện kế tiếp nhau không có bất cứ một thao tác nào của một giao dịch khác xen vào.

Chẳng hạn nếu phải thực hiện hai giao dịch T_1 và T_2 trong một lịch tuần tự thì hoặc T_1 thực hiện xong rồi đến T_2 hoặc T_2 thực hiện xong rồi đến T_1 . Nhưng cần phải chú ý rằng các lịch tuần tự khác nhau của cùng một tập giao dịch có thể đem đến những kết quả khác nhau. Với mỗi tập n giao dịch, tồn tại $n!$ lịch tuần tự có kết quả có thể khác nhau.

Lịch không tuần tự là lịch mà trong đó các thao tác của một tập các giao dịch tương tranh được xen lẫn nhau.

CSDL sẽ không bao giờ rơi vào trạng thái không nhất quán sau khi các giao dịch được thực hiện theo lịch tuần tự. Bởi vậy đối với một tập giao dịch, chúng ta muốn tìm được những lịch không tuần tự nhưng cho phép các giao dịch thực hiện đồng thời không ảnh hưởng đến nhau và bởi vậy kết quả trạng thái CSDL giống như khi thực hiện một lịch tuần tự các giao dịch này. Việc tìm ra những lịch như vậy cho một tập giao dịch được gọi là tuần tự hoá (serializability) và chúng ta có khái niệm khả tuần tự (serializable). Một lịch được gọi là tuần tự nếu nó là không tuần tự nhưng đưa ra kết quả như kết quả của một lịch tuần tự (đối với cùng một tập giao dịch).

Việc tuần tự hoá, thứ tự của các thao tác đọc và ghi rất quan trọng:

- Nếu hai thao tác đều chỉ đọc một mục dữ liệu thì chúng sẽ không ảnh hưởng đến nhau và thứ tự giữa chúng không quan trọng.

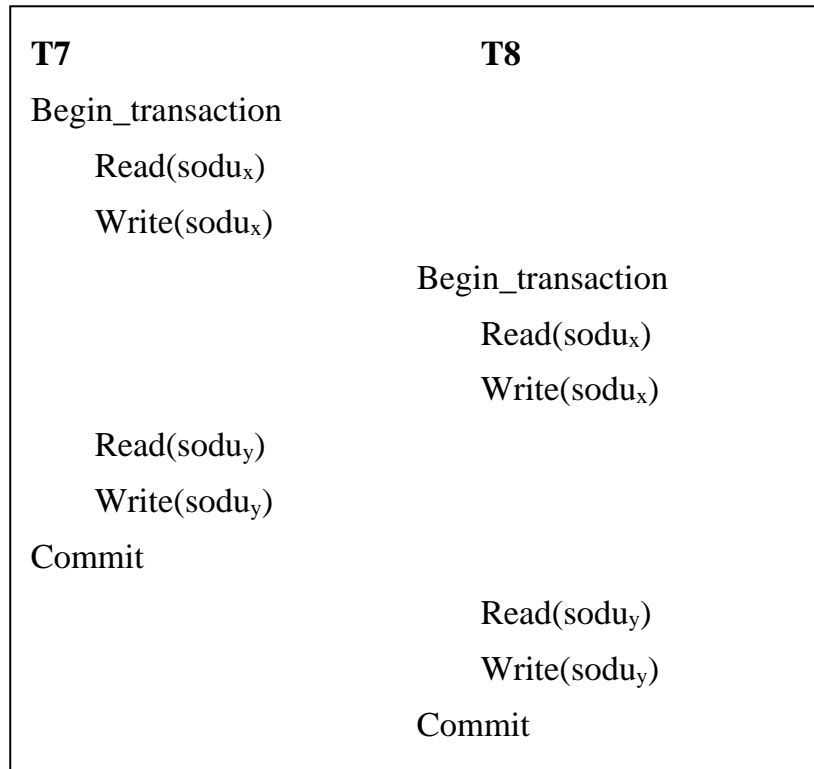
- Nếu hai thao tác đọc hay ghi trên hai mục dữ liệu hoàn toàn khác nhau thì chúng không ảnh hưởng đến nhau và thứ tự giữa chúng không quan trọng.

- Nếu một thao tác ghi một mục dữ liệu và một thao tác khác đọc hay ghi trên chính mục dữ liệu này thì thứ tự giữa chúng rất quan trọng.

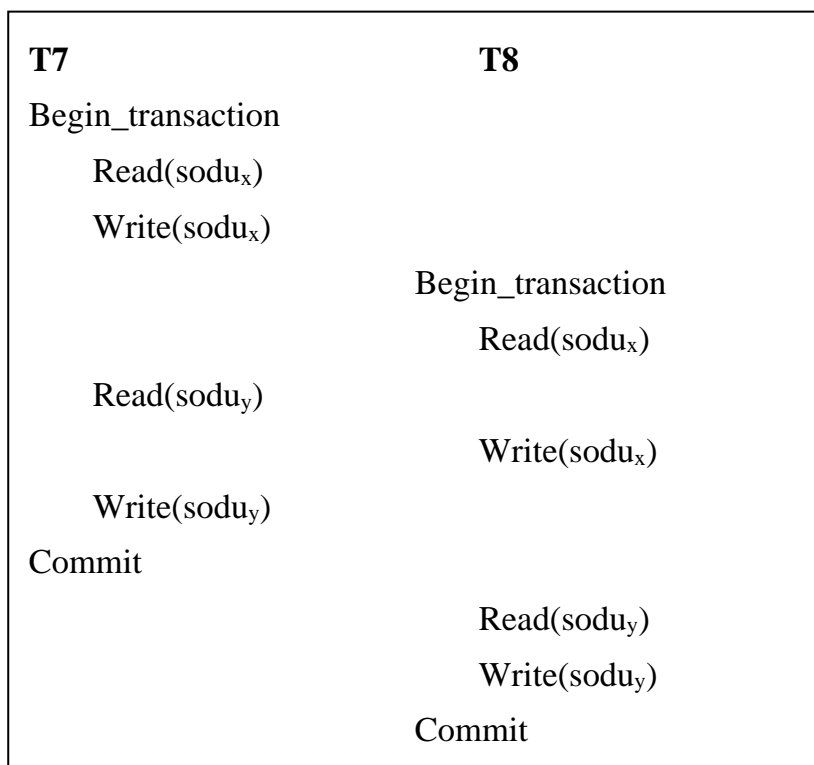
Chúng ta nói rằng hai thao tác kế tiếp là *xung đột* nếu chúng thuộc hai giao dịch khác nhau, tác động trên cùng một mục dữ liệu và ít nhất một trong hai thao tác đó là thao tác ghi.

Nếu một lịch S biến đổi được về lịch S' nhờ một dãy trao đổi thứ tự các thao tác (lệnh) không xung đột thì chúng ta nói rằng S và S' là *tương đương xung đột*.

Ví dụ 8.5: Ba lịch tương đương xung đột

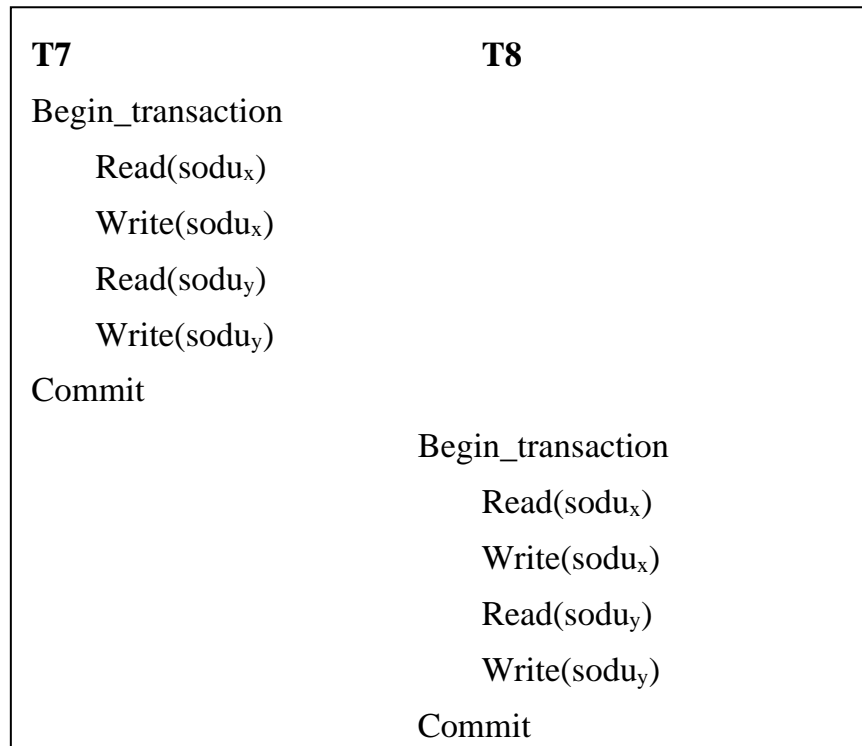


a.



b.

Hình 8.4. Các lịch tương đương



c.

Hình 8.4. Các lịch tương đương (tt)

Lịch (a) là lịch không tuần tự, do read(sodu_y) của T7 và write(sodu_x) của T8 làm việc trên hai mục dữ liệu tách biệt nên thứ tự giữa chúng có thể được thay đổi mà không ảnh hưởng đến kết quả. Do vậy chúng ta có lịch (a) tương đương xung đột với lịch (b).

Xét lịch (b), để có một lịch tương đương xung đột với (b), chúng ta có thể thay đổi thứ tự của một số cặp lệnh không xung đột như sau:

- Đổi thứ tự của write(sodu_x) của T8 với write(sodu_y) của T7
- Đổi thứ tự của read(sodu_x) của T8 với write(sodu_y) của T7
- Đổi thứ tự của read(sodu_x) của T8 với write(sodu_y) của T7

Lịch (c) là kết quả sau nhưng thay đổi thứ tự giữa các lệnh nói trên. Rõ ràng (c) là lịch tuần tự, do (a) và (b) tương đương với (c) nên chúng là các lịch khả tuần tự. Kiểu tuần tự hoá bằng cách trao đổi thứ tự các lệnh không xung đột, được gọi là *tuần tự hoá xung đột* (conflict serializability). Chúng ta cũng nói rằng các lịch (a) và (b) là *khả tuần tự xung đột*.

Có những lịch không có tính khả tuần tự xung đột. Chẳng hạn một lịch đơn giản như ở Hình 8.5 không tương đương xung đột với lịch tuần tự $\langle T, T' \rangle$, cũng không tương đương xung đột với lịch tuần tự $\langle T', T \rangle$.

T	T'
Begin_transaction	
Read(sodu _x)	Begin_transaction
	Write(sodu _x)
Write(sodu _x)	
Commit	
	Commit

Hình 8.5. Một lịch không khả tuần tự xung đột

Dưới giả thiết về ràng buộc cho thao tác ghi (đọc trước ghi), có một thuật toán kiểm tra xem một lịch có khả tuần tự xung đột hay không.

Thuật toán:

Input: Lịch S

Output: Trả lời S khả tuần tự xung đột hay không

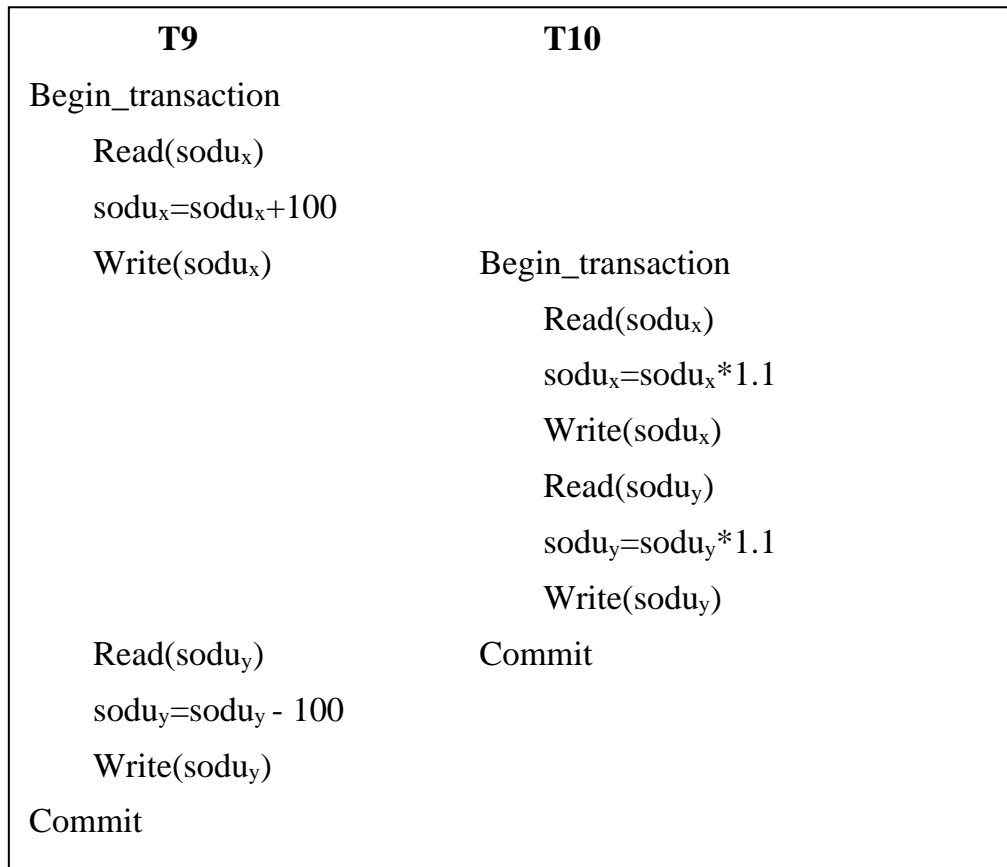
Phương pháp: (xây dựng một đồ thị có hướng G)

- (1) Với mỗi giao dịch T_i trong lịch S tạo một nút T_i trong đồ thị G
- (2) Vẽ một cung đi từ T_i đến T_j nếu giao dịch T_j thực hiện read(x) sau khi T_i thực hiện write(x)
- (3) Vẽ một cung đi từ T_i đến T_j nếu giao dịch T_j thực hiện write(x) sau khi T_i thực hiện read(x)
- (4) Vẽ một cung đi từ T_i đến T_j nếu giao dịch T_j thực hiện write(x) sau khi T_i thực hiện write(x)
- (5) Nếu đồ thị G không có chu trình thì S là khả tuần tự xung đột, ngược lại S không khả tuần tự xung đột.

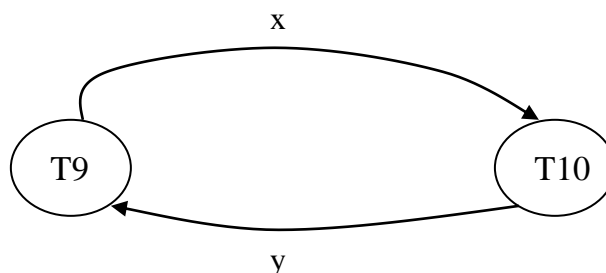
Ví dụ 8.6:

Lịch S trên hình 8.5 không khả tuần tự xung đột.

Giao dịch T9 chuyển 100\$ từ tài khoản y sang tài khoản x. Trong đó giao dịch T10 tăng mỗi tài khoản đó lên 10%



Hình 8.6. Một lịch không khả tuần tự xung đột



Hình 8.7. Đồ thị kiểm tra tính khả tuần tự xung đột của lịch S cho ở Hình 8.5

Như vậy, việc kiểm tra tính khả tuần tự xung đột của một lịch đưa ra về bài toán xây dựng một đồ thị có hướng từ lịch đã cho và kiểm tra xem có tồn tại một chu trình trong đồ thị đó hay không.

* Tuần tự nhìn thấy được

Có thể có hai lịch (trên cùng một tập giao dịch xác định) không tương đương xung đột với nhau nhưng lại cho kết quả như nhau. Điều này có ý nghĩa ở chỗ, để có một lịch khả tuần tự, nghĩa là tránh được sự tác động lẫn nhau của các giao dịch thực hiện đồng thời, không nhất thiết phải dùng các biến đổi tương đương xung đột.

Ví dụ 8.7:

Lịch đã cho ở Hình 8.7 không tương đương xung đột với lịch tuần tự <T,T'> nhưng cho cùng kết quả như lịch tuần tự <T,T'>

T	T'
Begin_transaction	
Read(sodu _x)	
sodu _x =sodu _x -50	
Write(sodu _x)	
	Begin_transaction
	Read(sodu _y)
	sodu _y =sodu _y -10
	Write(sodu _y)
Read(sodu _y)	
sodu _y =sodu _y + 50	
Write(sodu _y)	
Commit	
	Read(sodu _x)
	sodu _x =sodu _x +10
	Write(sodu _x)
	Commit

Hình 8.8. Một lịch khả tuần tự

Có một vài kiểu tuần tự hoá đưa ra định nghĩa tương đương giữa các lịch không quá chặt chẽ như ở trường hợp các lịch khả tuần tự xung đột. Dưới đây xin giới thiệu một định nghĩa như vậy:

Hai lịch S1 và S2 có cùng một tập các thao tác của n giao dịch T₁, T₂, ..., T_n được gọi là *tương đương thấy được* nếu ba điều kiện sau được thoả mãn:

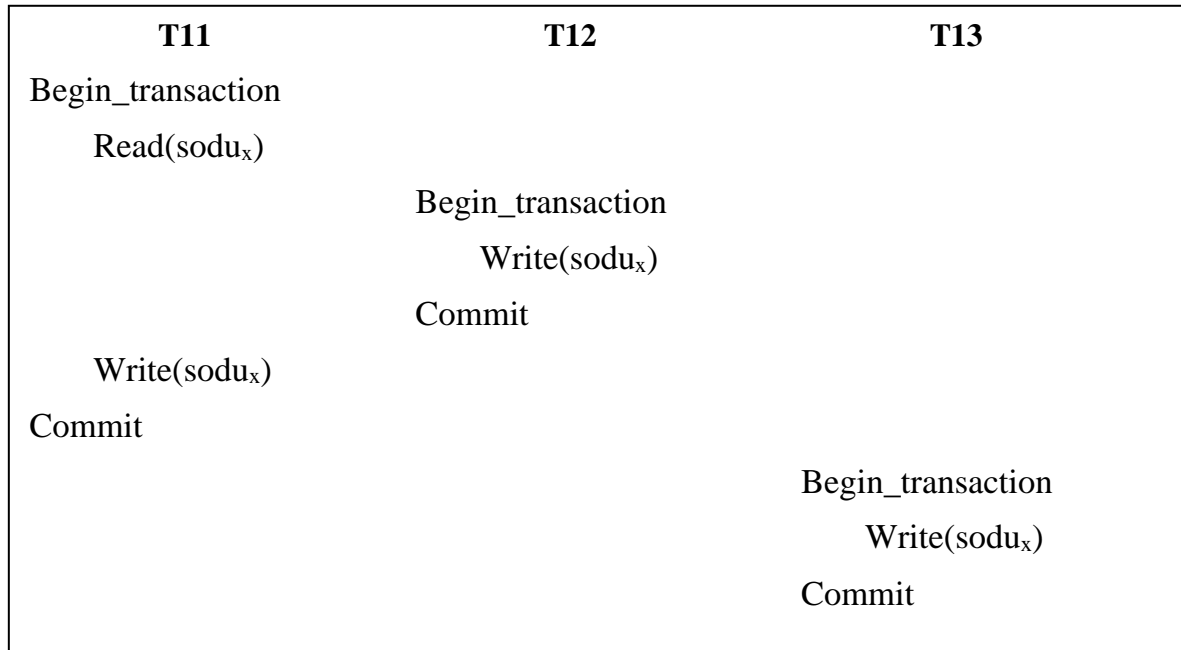
1. Với mỗi mục dữ liệu x, nếu giao dịch T_i đọc giá trị lần đầu của x trong lịch S1 thì giao dịch T_i cũng phải đọc giá trị lần đầu của x trong lịch S2.
2. Với mỗi thao tác đọc mục dữ liệu x của giao dịch T_i trong lịch S1, nếu giá trị đọc được này do giao dịch T_j nào đó ghi (vào CSDL) thì trong lịch S2, giao dịch T_j cũng phải đọc giá trị của x được ghi bởi giao dịch T_j
3. Với mỗi mục dữ liệu x, nếu trong lịch S1 thao tác ghi lần cuối được thực hiện bởi giao dịch T_i nào đó thì trong lịch S2 giao dịch T_i cũng là giao dịch ghi lần cuối cho x.

Khái niệm tương đương thấy được giữa các lịch dẫn đến khái niệm khả tuần tự thấy được. Một lịch được gọi là *khả tuần tự thấy được* (view serializable) nếu nó tương đương thấy được với một lịch tuần tự.

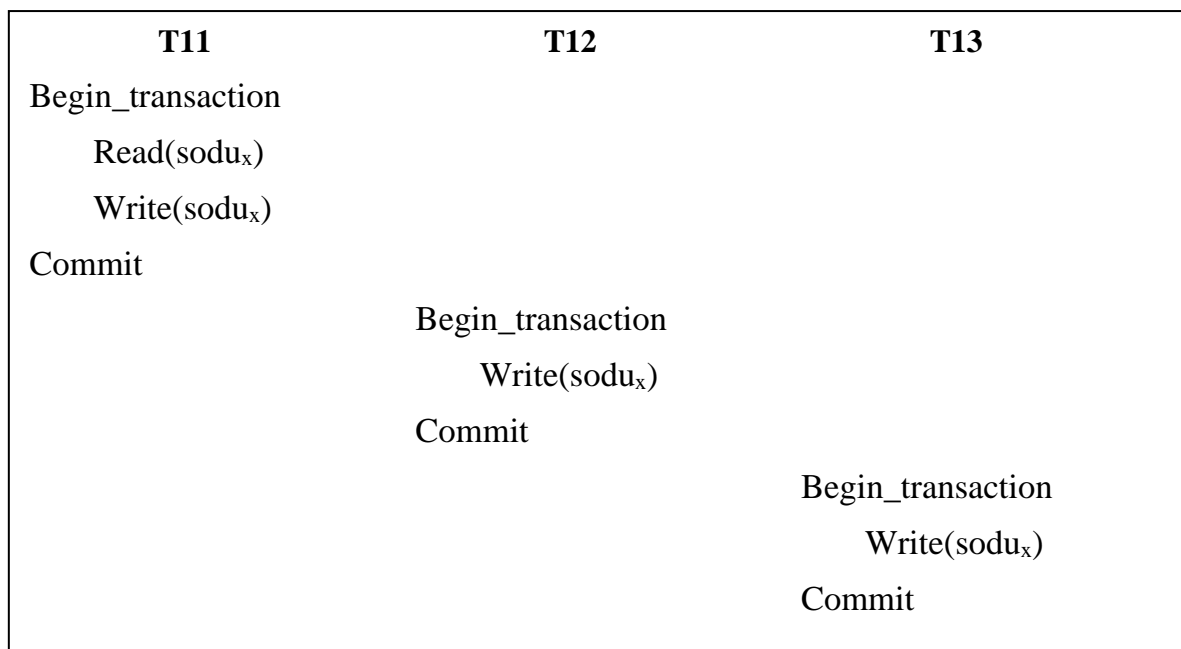
Mọi lịch khả tuần tự xung đột đều có khả năng khả tuần tự thấy được nhưng điều ngược lại không đúng, không phải lịch khả tuần tự thấy được nào cũng là lịch khả tuần tự xung đột.

Ví dụ 8.8:

Hãy xét lịch S1 cho ở Hình 8.8. S1 không tuân theo điều kiện ràng buộc thao tác ghi, hai giao dịch T12 và T13 đều có thao tác ghi write(sodu_x) mà trước đó không thực hiện read(sodu_x) (còn được gọi là *thao tác ghi nhắm mắt*). S1 không khả tuần tự xung đột, nhưng tương đương thấy được với lịch tuần tự S2 nên S1 là khả tuần tự thấy được. Người ta đã chỉ ra rằng bất cứ lịch nào khả tuần tự thấy được mà không khả tuần tự xung đột đều có chứa một hay nhiều thao tác ghi mù.



Hình 8.9. Lịch S1



Hình 8.10. Lịch S2

Trong thực tế, hệ quản trị CSDL không kiểm tra tính khả tuần tự của một lịch, bởi hệ điều hành sẽ định đoạt sự xen kẽ thực hiện thao tác của các giao dịch tương tranh. Thay vì vậy có những kỹ thuật được sử dụng để đưa ra những lịch khả tuần tự. Hai kỹ thuật như vậy được đề cập sau đây.

8.2.3. Các kỹ thuật điều khiển tương tranh

8.2.3.1. Phương pháp khóa

Một khóa là một biến gắn với một mục dữ liệu trong CSDL để biểu diễn trạng thái của mục dữ liệu này trong mối liên quan đến thao tác thực hiện trên đó.

Thông thường mỗi mục dữ liệu có một khóa. Chúng ta ký hiệu khóa gắn với mục dữ liệu X là LOCK(X). Tại mỗi thời điểm, mục dữ liệu X có một trong 3 trạng thái:

- Có khóa đọc (Read lock): nếu một giao dịch có khóa đọc trên một mục dữ liệu, nó có thể đọc nhưng không được cập nhật mục dữ liệu này.
- Có khóa ghi (Write lock): nếu một giao dịch có một khóa ghi trên một mục dữ liệu thì nó vừa được phép đọc vừa được phép cập nhật mục dữ liệu này.
- Không có khóa

Các khóa được sử dụng theo cách sau:

- Bất cứ một giao dịch nào cần truy cập vào một mục dữ liệu trước hết phải khóa mục dữ liệu đó lại. Giao dịch sẽ yêu cầu một khóa đọc nếu chỉ cần đọc dữ liệu vào, sẽ yêu cầu một khóa ghi nếu vừa cần đọc vừa cần ghi dữ liệu.
- Nếu mục dữ liệu đó chưa bị khóa bởi một giao dịch nào khác thì khóa sẽ được cấp phát theo đúng yêu cầu.
- Nếu mục dữ liệu đó đang bị khóa, hệ quản trị CSDL sẽ xác định xem khóa được yêu cầu có tương thích với khóa hiện hành hay không. Khi một giao dịch yêu cầu cấp một khóa đọc cho nó trên một mục dữ liệu mà trên mục đó đang có một khóa đọc (của giao dịch khác) thì khóa yêu cầu này sẽ được cấp phát. Trường hợp khóa được yêu cầu là khóa ghi thì giao dịch yêu cầu khóa sẽ phải chờ đợi cho đến khi khóa hiện hành được giải phóng mới được cấp khóa.
- Một giao dịch tiếp tục giữ một khóa cho đến thời điểm khóa đó được giải phóng. Thời điểm này hoặc nằm trong quá trình thực hiện giao dịch hoặc là thời điểm giao dịch được chuyển giao hay bị hủy bỏ. Chỉ khi khóa ghi được giải phóng thì kết quả của thao tác ghi mới thấy được đối với các giao dịch khác.

Một số hệ thống còn cho phép giao dịch đưa ra khóa đọc trên một mục dữ liệu và sau đó nâng cấp khóa lên thành khóa ghi. Điều này cho phép một giao dịch kiểm tra dữ liệu trước, sau đó mới quyết định có cập nhật hay không. Với hệ thống không hỗ trợ sự nâng cấp khóa như vậy, thì một giao dịch phải giữ các khóa ghi trên tất cả các mục dữ liệu mà nó muốn cập nhật tại một số thời điểm trong quá trình giao dịch được thực hiện. Điều này tiềm ẩn dẫn đến giảm sút mức hoạt động đồng thời trong hệ thống. Một số hệ thống cho phép giao dịch đưa ra một khóa ghi sau đó hạ mức xuống thành khóa đọc, việc sử dụng khóa cho các giao dịch như đã nói trên chưa đủ đảm bảo được tính khả tuần tự cho các lịch.

Ví dụ 8.9:

Xét lịch S cho hai giao dịch T9 và T10 ở Hình 8.4.

T9	T10
Begin_Transaction	Begin_Transaction
Read(sodu _x)	Read(sodu _x)
sodu _x = sodu _x + 100	sodu _x = sodu _x * 1.1
Write(sodu _x)	Write(sodu _x)
	Read(sodu _y)
	sodu _y = sodu _y * 1.1
	Write(sodu _y)
Read(sodu _y)	Commit
sodu _y = sodu _y - 100	
Write(sodu _y)	
Commit	

Hình 8.11. Ví dụ về phương pháp khóa

Theo những luật về sử dụng khóa đã nêu trên chúng ta có một lịch dùng với khóa như sau (không kể đến các thao tác của CPU):

```
{ write_lock(T9, sodux),
  read(T9, sodux),
  write(T9, sodux),
  unlock(T9, sodux),
  write_lock(T10, sodux),
  read(T10, sodux),
  write(T10, sodux),
  unlock(T10, sodux),
  write_lock(T10, soduy),
  read(T10, soduy),
  write(T10, soduy),
  unlock(T10, soduy),
  commit(T10)
  write_lock(T9, soduy),
  read(T9, soduy),
  write(T9, soduy),
  unlock(T9, soduy),
  commit(T9) }
```

Giả sử trước khi thực hiện T9 và T10 theo lịch trên, sodu_x = 100, sodu_y = 400. Nếu thực hiện tuần tự T9 xong rồi đến T10 thì kết quả sẽ là sodu_x=220, sodu_y=330.

Nếu thực hiện T10 trước rồi đến T9 thì kết quả là $sodu_x=210$, $so\ du_y=340$. Nhưng kết quả thực hiện tương tranh như theo lịch trên sẽ là $sodu_x=220$, $sodu_y=340$. Lịch S với việc sử dụng khóa như trên vẫn không phải là khả tuần tự.

Chúng ta hãy phân tích ví dụ này thêm, để thấy vì sao sử dụng khóa như vậy vẫn không đưa ra được lịch khả tuần tự. Theo lịch trên, khóa trên $sodu_x$ được giải phóng vì không cần truy cập vào đó nữa (lệnh `write_lock(T10, $sodu_x$)`). Ngay sau đó $sodu_x$ mới được khóa lại bởi T10, điều này có vẻ làm tăng lượng tương tranh nhưng thực ra làm hỏng dữ liệu. Đáng lẽ $sodu_x$ cũng phải được khóa bởi T9 cho đến khi T9 thực hiện xong sự truy cập vào vào mục dữ liệu này.

Để đảm bảo tính khả tuần tự, cần thiết phải yêu cầu mỗi giao dịch tuân theo một tập quy tắc được gọi là giao thức khóa, giao thức này chỉ định khi nào một giao dịch có thể khóa và giải phóng khóa trên một mục dữ liệu. Các giao thức khóa hạn chế số các lịch có thể dùng. Trong các giao thức khóa sinh ra các lịch khả tuần tự xung đột, giao thức khóa hai pha là giao thức được biết đến nhiều nhất.

b. Giao thức khóa hai pha

Chúng ta nói rằng một giao dịch tuân theo giao thức khóa hai pha nếu như tất cả các thao tác khóa đều xảy ra trước lần giải phóng khóa đầu tiên trong giao dịch.

Theo quy định của giao dịch này, mỗi giao dịch đều chia thành hai pha

- (1) Pha mở rộng: trong pha này giao dịch yêu cầu tất cả các khóa cần thiết nhưng không giải phóng một khóa nào.
- (2) Pha rút gọn: trong pha này giao dịch giải phóng các khóa của nó nhưng không yêu cầu thêm bất cứ một khóa mới nào.

Ban đầu giao dịch ở pha mở rộng, sau đó mới sang pha rút gọn. Không đòi hỏi tất cả các khóa đều được cấp đồng thời. Thông thường, giao dịch yêu cầu một số khóa, nó thực hiện một số xử lý nào đó và tiếp tục yêu cầu thêm những khóa cần thiết. Nhưng các giao dịch không bao giờ giải phóng bất kỳ một khóa nào cho đến khi nó không cần thêm một khóa mới nào nữa. Như vậy mỗi giao dịch đều phải yêu cầu một khóa trên một mục dữ liệu trước khi thao tác trên mục dữ liệu này, khóa đó là đọc hay ghi tùy thuộc vào mục đích truy cập. Mỗi giao dịch khi đã giải phóng một khóa sẽ không được yêu cầu thêm bất kỳ khóa nào nữa.

Trong những hệ thống cho phép nâng mức khóa, việc nâng mức khóa trên một mục dữ liệu chỉ có thể xảy ra trong pha mở rộng và có thể đòi hỏi giao dịch đó chờ cho đến khi giao dịch khác giải phóng khóa đọc trên mục dữ liệu này. Việc hạ mức khóa chỉ có thể xảy ra ở pha rút gọn.

Ví dụ 8.10:

Để giảm tổn thất do cập nhật ở Ví dụ 8.3, có thể sử dụng khóa với giao thức khóa hai pha như ở Hình 8.11.

Thời điểm	T1	T2	sodu _x
t ₁		Begin_transaction	100
t ₂	Begin_transaction	Write_lock(sodu _x)	100
t ₃	Write_lock(sodu _x)	Read(sodu _x)	100
t ₄	WAIT	sodu _x =sodu _x +100	100
t ₅	WAIT	Write(sodu _x)	200
t ₆	WAIT	Commit/unlock(sodu _x)	200
t ₇	Read(sodu _x)		200
t ₈	sodu _x =sodu _x -10		200
t ₉	Write(sodu _x)		190
t ₁₀	Commit/unlock(sodu _x)		190

Hình 8.12. Giao thức khóa hai pha

Trước tiên T2 phải yêu cầu khóa ghi trên sodu_x, sau đó nó có thể đọc được giá trị của sodu_x từ CSDL, tăng thêm 100 cho giá trị sodu_x và viết giá trị mới của sodu_x vào lại CSDL. Khi T1 bắt đầu thực hiện, nó cũng yêu cầu một khóa ghi trên sodu_x, nhưng do sodu_x cũng đồng thời có khóa ghi bởi T2 nên yêu cầu đó của T1 không được thỏa ngay lập tức. T1 phải chờ đến thời điểm T2 giải phóng khóa ghi của nó trên mục dữ liệu này, đó là thời điểm T2 được chuyển giao.

Ví dụ 8.11:

Để tránh lỗi do có giao dịch bị huỷ bỏ đã nêu trong Ví dụ 8.4, có thể sử dụng khóa với giao thức khóa-hai-pha như ở Hình 8.12. Trước tiên T4 yêu cầu khóa ghi trên sodu_x, sau đó nó mới đọc giá trị của sodu_x từ CSDL, tăng thêm 100 cho giá trị sodu_x và viết giá trị mới của sodu_x vào lại CSDL. Khi T4 bị huỷ bỏ (rollback) thì sự cập nhật của nó bị mất tác dụng, giá trị 100 của sodu_x trong CSDL trước khi thực hiện cập nhật đó được khôi phục lại. T1 bắt đầu yêu cầu một khóa ghi trên sodu_x, nhưng nó phải chờ đến thời điểm T2 giải phóng khóa ghi của T2 trên mục dữ liệu này, đó là thời điểm T2 bị huỷ bỏ hay được chuyển giao.

Thời điểm	T3	T4	sodu _x
t ₁		Begin_transaction	100
t ₂		Write_lock(sodu _x)	100
t ₃		Read(sodu _x)	100
t ₄	Begin_transaction	sodu _x =sodu _x +100	100
t ₅	Write_lock(sodu _x)	Write(sodu _x)	200
t ₆	WAIT	Rollback/unlock(sodu _x)	100
t ₇	Read(sodu _x)		100
t ₈	sodu _x =sodu _x -10		90
t ₉	Write(sodu _x)		90
t ₁₀	Commit/unlock(sodu _x)		90

Hình 8.13. Giao thức khóa hai pha

c. Nghẽn khóa

Nghẽn khóa là tình trạng bế tắc khi có hai hay nhiều giao dịch, trong đó giao dịch này đợi sự giải phóng khóa của giao dịch kia. Hình 8.14 cho thấy sự nghẽn khóa của hai giao dịch T17 và T18.

Thời điểm	T17	T18
t ₁	Begin_transaction	
t ₂	Write_lock(sodu _x)	Begin_transaction
t ₃	Read(sodu _x)	Write_lock(sodu _y)
t ₄	sodu _x =sodu _y -10	Read(sodu _y)
t ₅	Write(sodu _x)	sodu _y =sodu _y +100
t ₆	Write_lock(sodu _y)	Write(sodu _y)
t ₇	WAIT	Write_lock(sodu _x)
t ₈	WAIT	WAIT
t ₉	WAIT	WAIT
t ₁₀		WAIT
t ₁₁		

Hình 8.14. Nghẽn khóa giữa hai giao dịch

Tại thời điểm t₂, T17 yêu cầu và được cấp một khóa ghi trên mục dữ liệu sodu_x. tại thời điểm t₃, T18 yêu cầu và được cấp một khóa ghi trên mục dữ liệu sodu_y. Sau đó đến thời điểm t₆, T17 yêu cầu một khóa ghi trên sodu_y. Do T18 đang giữ khóa trên sodu_y trên T17 phải chờ. Trong khi đó ở thời điểm t₇, T18 yêu cầu ghi trên sodu_x mà khóa này đang do T17 nắm giữ. Cả hai thao tác này đều không tiếp tục thực hiện được, mỗi trong chúng đều chờ giao dịch kia giải phóng khóa. Khi xuất hiện tình huống nghẽn khóa như vậy, những ứng dụng liên quan không thể giải quyết được bế tắc này, hệ QTCSDL phải phát hiện thấy nghẽn khóa đã xảy ra và phải giải quyết gỡ bỏ nghẽn khóa.

Chỉ có một cách là huỷ bỏ một hay nhiều giao dịch để giải quyết bế tắc nói trên. Chẳng hạn ta có thể quyết định huỷ bỏ giao dịch T18 trong hai giao dịch ở hình 8.14. Khi thực hiện xong điều đó, các khóa do T18 nắm giữ được giải phóng và T17 sẽ được tiếp tục thực hiện.

Người dùng không nhận thấy sự xuất hiện tình trạng nghẽn khóa, do vậy hệ quản trị CSDL phải tự động khởi động lại hay huỷ bỏ một (hay một số) giao dịch.

Có hai phương pháp thông dụng để đối phó với tình trạng nghẽn khóa:

- Phòng ngừa nghẽn khóa: hệ quản trị CSDL xem xét trước rằng liệu có xảy ra nghẽn khóa hay không và không cho phép điều đó xảy ra. Một cách tiếp cận để tránh nghẽn khóa là sử dụng nhãn thời gian cho các giao dịch.
- Phát hiện nghẽn khóa và khôi phục: hệ quản trị CSDL cho phép nghẽn khóa có thể xuất hiện nhưng nhận ra được sự xuất hiện của nó và phá vỡ bế tắc do nghẽn khóa đó gây ra.

8.2.3.2. Phương pháp nhãn thời gian

Phương pháp nhãn thời gian là sử dụng các giao dịch được gán nhãn thời gian để xếp thứ tự thực hiện các thao tác cho một lịch tương đương với lịch tuần tự.

Do không sử dụng khóa, theo phương pháp này không có tính huống nghẽn khóa. Để tránh xung đột, phương pháp dùng khóa buộc các giao dịch phải chờ đợi. Trong khi đó, phương pháp nhãn thời gian không bắt các giao dịch chờ đợi, các giao dịch có xung đột sẽ được hủy bỏ hay bắt đầu lại từ đầu.

Một *nhãn thời gian* là một nhãn xác định duy nhất được tạo ra bởi hệ quản trị CSDL để xác định một giao dịch. Có hai cách thường dùng để sinh ra nhãn thời gian:

- Dùng một bộ đếm tăng mỗi lần giá trị của nó được gán cho một giao dịch. Các nhãn thời gian của các giao dịch sẽ có các giá trị là 1, 2, 3... Một bộ đếm trong máy tính có một số hữu hạn giá trị đếm, bởi vậy hệ thống phải khởi trị lại cho bộ đếm (gán bằng 0) một cách định kỳ khi các giao dịch không được thực hiện trong một khoảng thời gian ngắn nào đó.
- Dùng giá trị tức thời của đồng hồ hệ thống khi giao dịch được bắt đầu thực hiện.

Phương pháp điều khiển tương tranh được nói tới ở đây là sắp xếp thứ tự các thao tác dựa trên nhãn thời gian của chúng. Nếu $TS(T_i) < TS(T_j)$ thì hệ thống phải đảm bảo rằng lịch sinh ra tương đương với một lịch tuần tự trong đó T_i thực hiện trước T_j . Để có được điều này, mỗi mục dữ liệu X được kết hợp với hai giá trị nhãn thời gian:

(1) Read_TS(X) (còn được gọi là nhãn đọc): đây là nhãn thời gian lớn nhất của giao dịch đã thực hiện thành công thao tác Read(X), cho biết thời điểm lần cuối đọc X .

(2) Write_TS(X) (còn được gọi là nhãn ghi): đây là nhãn thời gian lớn nhất của giao dịch đã thực hiện thành công thao tác Write(X), cho biết thời điểm lần cuối ghi vào X .

Mỗi khi có một giao dịch T nào đó thực hiện một thao tác read(X) (hay write(X)), nhãn thời gian của T lại được so sánh với nhãn thời gian đọc X (hay nhãn thời gian ghi X) để kiểm tra xem thứ tự các giao dịch thực hiện có đúng với thứ tự nhãn thời gian của chúng hay không.

Nếu thứ tự nhãn thời gian bị vi phạm bởi các thao tác đó thì giao dịch T vi phạm tính tương đương với lịch tuần tự. Trong trường hợp này, giao dịch T phải bị hủy bỏ và bất cứ tác động nào do nó gây ra trên CSDL cũng phải được tháo gỡ. Sau đó T sẽ lại được đưa vào hệ thống như một giao dịch mới với nhãn thời gian mới. Cần chú ý rằng sau khi T bị hủy bỏ có thể sẽ kéo theo một số giao dịch khác bị hủy bỏ (hiện tượng phục hồi lan truyền).

Giao thức nhãn thời gian đảm bảo rằng các thao tác đọc, ghi thực hiện theo đúng thứ tự của nhãn thời gian. Đối với một giao dịch T có nhãn thời gian $TS(T)$, giao thức quy định như sau:

(1) Khi giao dịch T phát ra lệnh READ(X):

- Nếu T cần đọc mục dữ liệu X mà mục dữ liệu này đã bị cập nhật bởi một giao dịch có nhãn thời gian lớn hơn, nghĩa là $TS(T) < Write_TS(X)$, thì T phải bị phục hồi và sẽ được bắt đầu lại với một nhãn thời gian mới.

- Ngược lại, nếu $TS(T) > Write_TS(X)$ thì thao tác Read(X) này được thực hiện và nhãn thời gian đọc của X được gán giá trị lớn hơn trong hai giai đoạn Read_TS(X) và TS(X).

(2) Khi giao dịch T phát ra lệnh Write(X):

- Nếu T cần ghi vào mục dữ liệu X mà mục dữ liệu này đã bị đọc bởi một giao dịch có nhãn thời gian lớn hơn, nghĩa là $TS(T) < Read_TS(X)$, thì T phải bị phục hồi và sẽ được bắt đầu lại với một nhãn thời gian mới.

- Nếu T cần ghi vào mục dữ liệu X mà mục dữ liệu này đã bị ghi bởi một giao dịch có nhãn thời gian lớn hơn, nghĩa là $TS(T) < Write_TS(X)$, thì T phải bị phục hồi và sẽ được bắt đầu lại với một nhãn thời gian mới.

- Ngược lại, thao tác $Write(X)$ này được thực hiện và $Write_TS(T)$ được đặt là $TS(T)$

Giao thức dựa trên nhãn thời gian trên đây đảm bảo các giao dịch được thực hiện theo tiêu chuẩn khả tuần tự xung đột.

Nếu biến đổi giao thức nhãn thời gian một chút, về tiềm ẩn có thể đạt được mức thực hiện đồng thời cao hơn nữa, khi những quy định trên có chút thay đổi sau:

Khi giao dịch T phát ra lệnh $Write(X)$:

- Nếu $TS(T) < Read_TS(X)$, thì giá trị mà T cần ghi vào mục dữ liệu X đã được đọc trước đó bởi một giao dịch có nhãn thời gian lớn hơn, bởi vậy thao tác $Write(X)$ của T bị huỷ bỏ và T phải bị phục hồi, T sẽ được bắt đầu lại với một nhãn thời gian mới.

- Nếu $TS(T) < Write_TS(X)$, thì T đang cố gắng ghi một giá trị quá hạn vào mục dữ liệu X (quá hạn vì nó đã bị ghi bởi một giao dịch có nhãn thời gian lớn hơn), bởi vậy có thể huỷ bỏ thao tác $Write(X)$ này của T

- Ngược lại, thao tác $Write(X)$ này được thực hiện và $Write_TS(T)$ được đặt là $TS(T)$

Những thay đổi trong giao dịch nhãn thời gian như vậy được gọi là *luật ghi của Thomas*.

Ví dụ 8.12:

Xét 3 giao dịch T17, T18, T19 thực hiện đồng thời, với $TS(T17) < TS(T18) < TS(T19)$

	Thao tác	T17	T18	T19
t ₁		Begin_transaction		
t ₂	Read(sodu _x)	Read(sodu _x)		
t ₃	sodu _x =sodu _x +10	sodu _x =sodu _x +10		
t ₄	Write(sodu _x)	Write(sodu _x)	Begin_transaction	
t ₅	Read(sodu _y)		Read(sodu _y)	
t ₆	sodu _y =sodu _y +20		sodu _y =sodu _y +20	Begin_transaction
t ₇	Read(sodu _y)			Read(sodu _y)
t ₈	Write(sodu _y)		Write(sodu _y)	
t ₉	sodu _y =sodu _y +30			sodu _y =sodu _y +30
t ₁₀	Write(sodu _y)			Write(sodu _y)
t ₁₁	sodu _y =100			sodu _y =100
t ₁₂	Write(sodu _z)			Write(sodu _z)
t ₁₃	sodu _z =50	sodu _z =50		Commit
t ₁₄	Write(sodu _z)	Write(sodu _z)	Begin_transaction	
t ₁₅	Read(sodu _y)	Commit	Read(sodu _y)	
t ₁₆	sodu _y =sodu _y +20		sodu _y =sodu _y +20	
t ₁₇	Write(sodu _y)		Write(sodu _y)	
t ₁₈			Commit	

Hình 8.15. Một lịch tuân theo giao thức nhãn thời gian với luật Thomas

8.3. Khôi phục CSDL

Khôi phục dữ liệu là một dịch vụ mà Hệ quản trị CSDL phải cung cấp để đảm bảo rằng CSDL ở một trạng thái toàn vẹn, đúng đắn sau mỗi lần có sự cố xảy ra.

8.3.1. Sự cần thiết phải có dịch vụ khôi phục CSDL

Có nhiều loại sự cố khác nhau có thể tác động đến việc xử lý CSDL:

- Hệ thống gặp sự cố do lỗi phần cứng hoặc lỗi phần mềm, kết quả là mất thông tin ở bộ nhớ chính
- Thiết bị nhớ trung gian bị hỏng, chẳng hạn như gây vỡ hay không đọc được, kết quả là mất một số phần dữ liệu lưu trữ ở đây
- Lỗi phần mềm ứng dụng làm cho một hay nhiều giao dịch bị hỏng
- Sự cố vật lý như mất điện, cháy, lụt, động đất...
- Sự cố do tính bất cẩn vô ý của người vận hành hay người dùng dẫn đến việc hủy bỏ dữ liệu hay các phương tiện

- Sự phá hoại dữ liệu, phần cứng hay phần mềm một cách cố ý thức.

Sau đây chúng ta sẽ đề cập đến những kỹ thuật nhằm hạn chế những tác hại và cho phép khôi phục CSDL sau khi xảy ra sự cố.

8.3.2. Giao dịch và sự khôi phục

Giao dịch cũng là một đơn vị cơ sở trong việc khôi phục của hệ thống CSDL. Khi xảy ra sự cố thì bộ quản lý khôi phục phải đảm bảo được hai trong bốn tính chất của ACID của các giao dịch, đó là tính nguyên tử và tính bền vững. Bộ quản lý khôi phục dữ liệu phải đảm bảo được rằng trong sự hồi phục CSDL, hoặc tất cả các tác động của một giao dịch được ghi vào CSDL hoặc không một tác động nào được ghi nhận.

8.3.3. Phương tiện khôi phục

Một hệ quản trị CSDL cần cung cấp các phương tiện cho việc khôi phục CSDL như sau:

- Một cơ chế sao lưu dự phòng (backup) tạo các bản sao CSDL theo định kỳ. Có thể sao lưu toàn bộ CSDL và cũng có thể chỉ sao chép những thay đổi được thực hiện kể từ lần sao lưu gần nhất trước đó. Thông thường những bản sao này được lưu trữ trong thiết bị bộ nhớ ổn định, chẳng hạn như băng từ.
- Hệ quản trị CSDL bảo trì một tệp tin đặc biệt gọi là nhật ký (log) chứa thông tin về tất cả các cập nhật thực hiện trên CSDL, nhằm lưu giữ vết các thao tác.
- Đánh dấu điểm kiểm tra (checkpoint) để giới hạn việc duyệt và dãy xử lý phải thực hiện theo nhật ký nhằm khôi phục CSDL mỗi khi sự cố xảy ra. Mặc dù thông tin trong nhật ký cho phép lần theo các xử lý đã có, nhưng khó khăn ở chỗ không biết phải lùi lại điểm nào trong nhật ký để bắt đầu thực hiện theo vết. Bởi vậy cần đánh dấu một điểm kiểm tra, đó là điểm đồng bộ giữa CSDL và nhật ký, nghĩa là mục dữ liệu được ghi trước điểm đó đã được ghi vào bộ nhớ thứ cấp rồi.
- Một bộ quản lý khôi phục cho phép hệ thống khôi phục lại trạng thái toàn vẹn của CSDL sau khi xảy ra sự cố.

8.3.4. Một số kỹ thuật khôi phục

Việc sử dụng một thủ tục khôi phục cụ thể tùy thuộc vào mức độ thiệt hại của CSDL. Chúng ta xét hai trường hợp sau:

- Nếu CSDL thiệt hại nhiều, chẳng hạn đĩa hỏng và CSDL bị phá hủy thì cần phải khôi phục lại bản sao dự phòng của CSDL và thực hiện các giao dịch đã được chuyển giao theo nhật ký (với giả sử nhật ký không bị hỏng)
- Nếu CSDL không bị hư hỏng về mặt vật lý nhưng trở nên không nhất quán do hệ thống bị hư hỏng trong lúc thực hiện một số giao dịch, thì cần phải gỡ bỏ những cập nhật dẫn đến sự không nhất quán đó.

Sau đây là một số kỹ thuật khôi phục:

8.3.4.1. Khôi phục dựa vào nhật ký (file log) của giao dịch

Một kỹ thuật thường được sử dụng để thực hiện khôi phục lại dữ liệu dựa vào quá khứ giao dịch đã diễn ra trước đó.

- Ghi lại thông tin về quá trình xảy ra giao dịch trong một nhật ký tính từ trạng thái đúng đắn của dữ liệu cuối cùng.
- Do vậy CSDL nhận biết trạng thái đúng đắn của nó trước và sau mỗi giao dịch. Mỗi khi CSDL được trả về trạng thái đúng đắn sau khi giao dịch chấm dứt, nhật ký sẽ loại bỏ những thông tin giao dịch trước đó. Khi CSDL được trả lại trạng thái đúng đắn, quá trình này thường được gọi là điểm kiểm tra "checkpoint".

8.3.4.2. Kỹ thuật khôi phục theo kiểu cập nhật trì hoãn

Khi tuân theo giao thức này, các cập nhật chưa được ghi ngay vào CSDL cho đến tận khi một giao dịch đạt được điểm chuyển giao của nó.

Nếu một giao dịch bị thất bại trước khi đạt được điểm chuyển giao thì nó chưa thật sự làm thay đổi CSDL và vì vậy không cần thiết phải gỡ bỏ các cập nhật.

Nhưng nếu giao dịch đã được chuyển giao, việc gỡ bỏ các cập nhật là cần thiết và phải sử dụng nhật ký.

Trong quá trình thực hiện giao dịch, các cập nhật chỉ được ghi vào vùng làm việc của giao dịch đó (transaction workspace) trong nhật ký. Như vậy, một giao dịch sẽ không làm thay đổi CSDL cho đến khi nó đạt được điểm chuyển giao và một giao dịch sẽ không đạt được điểm chuyển giao (của nó) cho đến khi tất cả các cập nhật nó thực hiện được ghi vào nhật ký và nhật ký buộc phải ghi vào đĩa.

8.3.4.3. Kỹ thuật khôi phục theo kiểu cập nhật ngay

Trong kỹ thuật này, khi một giao dịch đưa ra một lệnh cập nhật, CSDL có thể cập nhật ngay không chờ đến lúc giao dịch đạt được điểm chuyển giao. Tuy nhiên, phép cập nhật này phải được ghi vào nhật ký (trên đĩa) trước khi nó được thực hiện trên CSDL. Quy tắc này được gọi là giao thức *nhật - ký - ghi - trước*.

Để áp dụng được các quy tắc đó, hệ thống phải gỡ bỏ tác động của các thao tác cập nhật trên CSDL do một giao dịch đưa ra nếu giao dịch này thất bại trong khi thực hiện.

8.3.4.4. Kỹ thuật khôi phục sử dụng trang ghi vết

Đối với mỗi giao dịch, trong khoảng thời gian tồn tại của nó, hệ thống duy trì hai bảng trang (page table):

- Bảng hiện hành
- Bảng vết

Khi giao dịch bắt đầu thực hiện, hai bảng này giống nhau, sau đó bảng vết không thay đổi và được dùng để khôi phục CSDL khi có sự cố xảy ra.

Trong quá trình thực hiện giao dịch, bảng hiện hành được dùng để ghi lại tất cả các cập nhật cho CSDL. Khi việc thực hiện giao dịch hoàn thành thì bảng hiện hành lại trở thành bảng ghi vết.

Trong môi trường có nhiều người sử dụng với nhiều giao dịch thực hiện đồng thời, duy trì các nhật ký và các điểm kiểm tra phải được kết hợp cùng với kỹ thuật trang ghi vết.

Kỹ thuật này có ưu điểm là tốc độ khôi phục nhanh do không cần các phép gỡ bỏ (undo) hay làm lại (redo). Tuy nhiên nhược điểm ở chỗ đòi hỏi những chiến lược lưu trữ phức tạp nhằm giữ được các trang CSDL có liên quan đặt gần nhau trên đĩa. Ngoài

ra, nếu băng (dùng để ghi vết) mà lớn thì mỗi khi giao dịch được chuyển giao, vùng nhớ dùng thêm trên đĩa cho nó là đáng kể.

TỔNG KẾT CHƯƠNG 8

- Một giao dịch là một hành động hay một dãy hành động truy cập vào CSDL hoặc làm thay đổi nội dung CSDL, được đưa ra bởi một NSD hay một chương trình ứng dụng.
- Các tính chất của giao dịch: tính nguyên tố, tính nhất quán, tính cô lập, tính bền vững.
- Các kỹ thuật điều khiển tương tranh: Phương pháp khóa, Phương pháp nhãn thời gian.
- Phương pháp khóa là sử dụng khóa đọc (Read) và khóa ghi (Write) để khóa những giao dịch đang thực thi.
- Phương pháp nhãn thời gian là sử dụng các giao dịch được gán nhãn thời gian để xếp thứ tự thực hiện các thao tác cho một lịch tương đương với lịch tuần tự.
- Khôi phục dữ liệu là một dịch vụ mà hệ quản trị CSDL phải cung cấp để đảm bảo rằng CSDL ở một trạng thái toàn vẹn, đúng đắn sau mỗi lần có sự cố xảy ra.
- Một số kỹ thuật khôi phục: Khôi phục dựa vào file log của giao dịch, Kỹ thuật khôi phục theo kiểu cập nhật trì hoãn, Kỹ thuật khôi phục theo kiểu cập nhật ngay, Kỹ thuật khôi phục sử dụng trang ghi vết
- Khôi phục dựa vào file log của giao dịch là Ghi lại thông tin về quá trình xảy ra giao dịch trong một file log tính từ trạng thái đúng đắn của dữ liệu cuối cùng.
- Kỹ thuật khôi phục theo kiểu cập nhật trì hoãn: các cập nhật chưa được ghi ngay vào CSDL cho đến tận khi một giao dịch đạt được điểm chuyển giao của nó.
- Kỹ thuật khôi phục theo kiểu cập nhật ngay: khi một giao dịch đưa ra một lệnh cập nhật, CSDL có thể cập nhật ngay không chờ đến lúc giao dịch đạt được điểm chuyển giao. Tuy nhiên, phép cập nhật này phải được ghi vào nhật ký (trên đĩa) trước khi nó được thực hiện trên CSDL
- Kỹ thuật khôi phục sử dụng trang ghi vết: Đối với mỗi giao dịch, trong khoảng thời gian tồn tại của nó, hệ thống duy trì hai bảng trang (page table) là Bảng hiện hành, và Bảng vết. Khi giao dịch bắt đầu thực hiện, hai bảng này giống nhau, sau đó bảng vết không thay đổi và được dùng để khôi phục CSDL khi có sự cố xảy ra.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 8

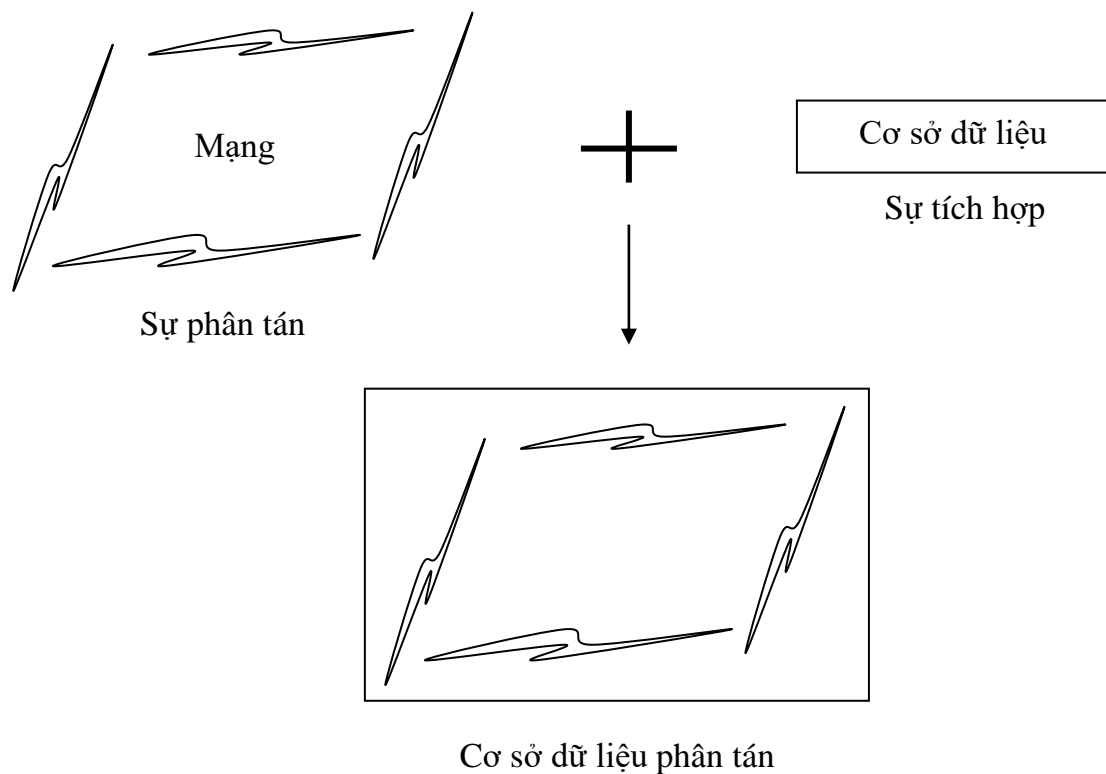
- 8.1.** Hãy giải thích thuật ngữ "giao dịch" và ý nghĩa quan trọng của giao dịch với tư cách là đơn vị công việc trong hoạt động của một hệ QTCSDL.
- 8.2.** Hãy phân tích một trong số bốn tính chất ACID cần được đảm bảo của giao dịch trong điều khiển tương tranh và khôi phục dữ liệu. Cho ví dụ minh họa.
- 8.3.** Cho biết các loại vấn đề (dẫn đến kết quả không theo ý muốn) trong một môi trường nhiều người dùng và cho phép nhiều người truy cập đồng thời. Cho ví dụ minh họa.
- 8.4.** Cho biết những nguyên nhân dẫn đến việc hầu hết các CSDL cài đặt giao thức khóa hai pha.
- 8.5.** Trong điều khiển tương tranh, với kỹ thuật dùng nhãn thời gian, khi nào một giao dịch bị buộc phục hồi và nó có được gán một nhãn thời gian mới hay không? Vì sao?
- 8.6.** Làm thế nào để tránh tình huống nghẽn khóa (deadlock)? Làm thế nào để kiểm tra có xuất hiện nghẽn khóa hay không?
- 8.7.** So sánh sự khác nhau giữa các kỹ thuật khôi phục: cập nhật trì hoãn và cập nhật ngay.

CHƯƠNG 9

HỆ CƠ SỞ DỮ LIỆU PHÂN TÁN

Công nghệ CSDL phân tán là sự hợp nhất của hai cách tiếp cận xử lý dữ liệu dường như đối lập nhau: công nghệ CSDL và công nghệ mạng máy tính.

Như chúng ta đã biết, một trong những động cơ chính đứng phía sau việc sử dụng các hệ CSDL là mong muốn tích hợp các dữ liệu tác nghiệp của một tổ chức, cho phép truy cập tập trung và có điều khiển đến các dữ liệu đó. Các hệ CSDL đầu tiên, xuất hiện vào những năm 1960, đã thực sự thay thế phương thức xử lý dữ liệu cổ điển, trong đó mỗi chương trình ứng dụng định nghĩa và duy trì một tệp dữ liệu riêng, bằng phương thức định nghĩa và quản lý dữ liệu tập trung cho nhiều ứng dụng khác nhau của cùng một tổ chức. Mặt khác, công nghệ mạng máy tính xúc tiến một kiểu làm việc đi ngược lại với mọi nỗ lực tập trung hoá. Hai cách tiếp cận trái ngược đó có thể được tổng hợp để cho ra đời một công nghệ mạnh mẽ và nhiều hứa hẹn: công nghệ CSDL phân tán (Hình 9.1).



Hình 9.1. Sự hình thành của các hệ CSDL phân tán

Nội dung của chương này bao gồm các khái niệm và kiến trúc của hệ CSDL phân tán, các ưu và nhược điểm của hệ CSDL phân tán, và cuối cùng là cách thức quản lý các giao dịch phân tán.

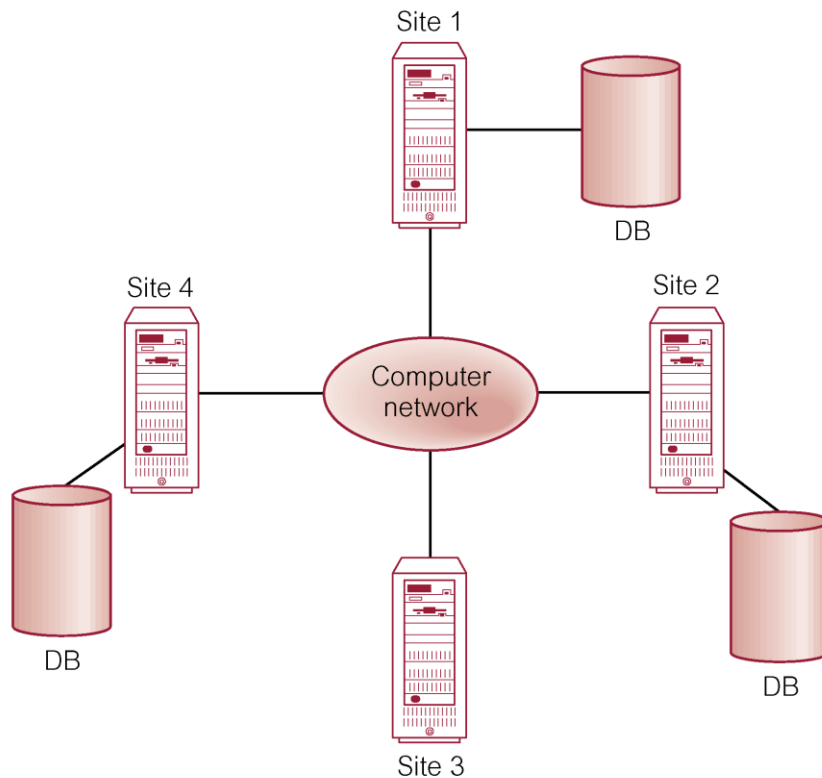
9.1. Khái niệm và kiến trúc

9.1.1. Khái niệm

Một CSDL phân tán là một tập các CSDL cùng hợp tác hoạt động, được lưu trữ trên các máy tính khác nhau, gọi là các trạm (sites), và được nối với nhau bằng một mạng truyền thông. NSD tại một trạm bất kỳ có thể truy cập và thao tác dữ liệu trên các trạm khác.

Hệ quản trị CSDL phân tán là một tập các phần mềm hệ thống hỗ trợ việc tạo lập và bảo trì các CSDL phân tán.

Như vậy, có thể xem một CSDL phân tán là một bộ sưu tập các dữ liệu về mặt logic thuộc cùng một hệ thống nhưng về mặt vật lý được phân bố trên các trạm của một mạng máy tính, và hệ quản trị CSDL phân tán là hệ thống phần mềm cho phép quản lý CSDL phân tán, làm cho việc phân tán trở nên thông suốt đối với người dùng (Hình 9.2).



Hình 9.2. Hệ cơ sở dữ liệu phân tán

Mục tiêu quan trọng nhất của CSDL phân tán là có tích hợp mà không cần tập trung hoá. Do yêu cầu của các đơn vị là phải tổ chức sao cho kinh doanh có hiệu quả nhất và nắm bắt thông tin nhanh nhất khi các cơ sở của công ty ở những địa điểm xa nhau, nên việc xây dựng một hệ thống làm việc trên CSDL phân tán là phù hợp xu hướng hiện nay.

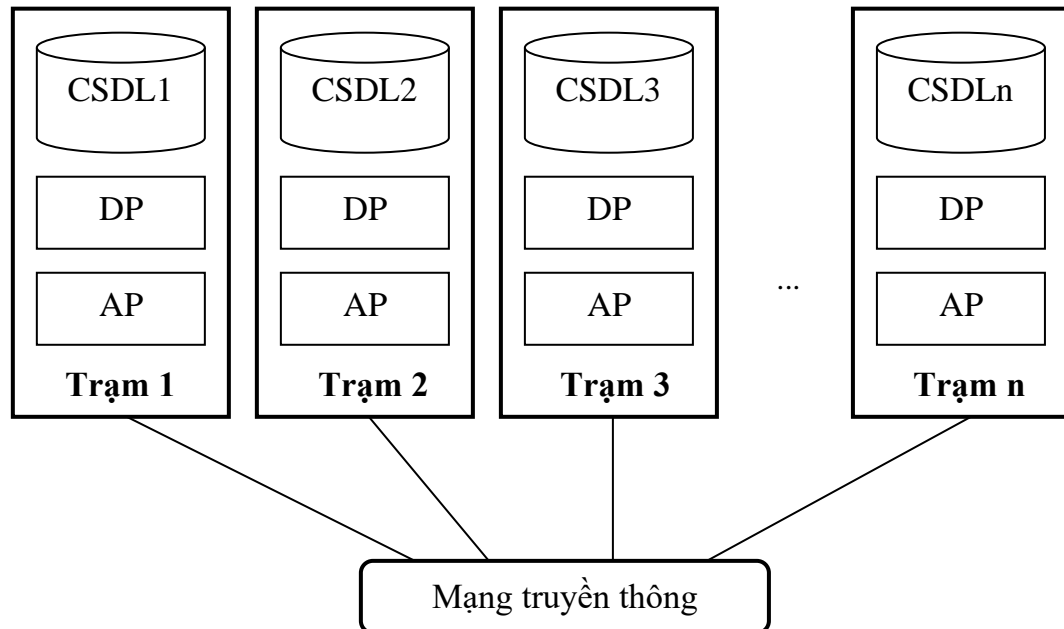
CSDL phân tán có hai đặc điểm với tầm quan trọng tương đương nhau:

- **Việc phân tán:** Trong thực tế, dữ liệu trong CSDL phân tán không đặt trên cùng một vị trí. Đây là đặc điểm để phân biệt CSDL phân tán với CSDL tập trung.
- **Liên quan logic:** Trong CSDL phân tán, dữ liệu có một số đặc tính quan hệ chặt chẽ với nhau như tính kết nối, tính liên quan logic... Trong CSDL tập trung, mỗi vị trí quản lý một CSDL và NSD phải truy cập đến CSDL ở những vị trí khác nhau để lấy thông tin tổng hợp, dữ liệu thiếu tính kết nối với nhau.

9.1.2. Kiến trúc của một hệ CSDL phân tán

Ở mức phân cứng vật lý, những nhân tố để phân biệt một hệ CSDL phân tán với một hệ CSDL tập trung là:

- Hệ CSDL phân tán gồm các trạm hay nút (node).
- Các trạm phải được kết nối bởi một kiểu mạng truyền thông nào đó để truyền dữ liệu và các lệnh giữa các trạm (Hình 9.3).



Hình 9.3. Kiến trúc đơn giản hoá của một hệ thống vật lý cho một hệ CSDL phân tán

Để quản lý được tính phức tạp của một Hệ CSDL phân tán, phần mềm tại các trạm được chia làm ba mô đun chính:

- Phần mềm xử lý dữ liệu (DP - Data Processor): Chịu trách nhiệm quản lý dữ liệu cục bộ (địa phương) tại một trạm, giống với phần mềm CSDL tập trung.
- Phần mềm xử lý ứng dụng (AP - Application Processor): Chịu trách nhiệm với phần lớn chức năng phân tán, nó truy cập thông tin phân tán từ thư mục CSDL phân tán và chịu trách nhiệm xử lý mọi yêu cầu đòi hỏi truy cập đến nhiều hơn một trạm.
- Phần mềm truyền thông (thường quan hệ với một hệ điều hành phân tán): Cung cấp các truyền thông nguyên thủy được AP dùng để truyền lệnh và dữ liệu giữa các trạm khác nhau khi cần.

Từ đó, có thể thấy hệ quản trị CSDL phân tán là một tập các phần mềm hệ thống bao gồm phần mềm quản trị các dữ liệu phân tán, các phần mềm quản trị truyền thông, và các hệ quản trị CSDL địa phương / cục bộ lưu trữ trên mỗi trạm của hệ CSDL phân tán. Ngoài chức năng của các hệ quản trị CSDL và của phần mềm quản trị truyền thông, hệ quản trị CSDL phân tán còn có các chức năng đặc biệt sau:

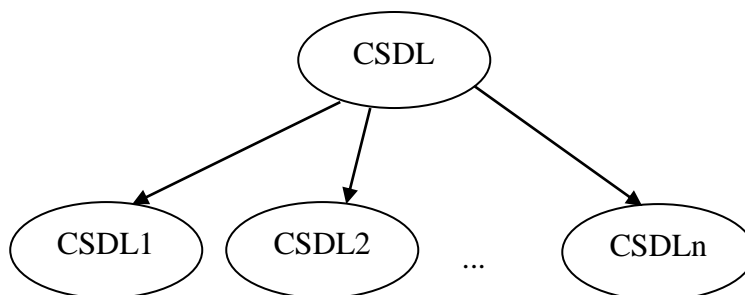
- Quản trị một từ điển dữ liệu tổng thể lưu giữ thông tin liên quan tới các dữ liệu phân tán.
- Định nghĩa các dữ liệu phân tán

- Kiểm tra ngữ nghĩa của các dữ liệu phân tán
- Đánh giá các câu hỏi phân tán do người dùng đưa ra: là chức năng biến đổi một câu hỏi phân tán về một phương án thực thi phân tán.
- Quản lý các giao dịch phân tán, các chương trình nguyên tố trong đó có chứa các câu hỏi phân tán.

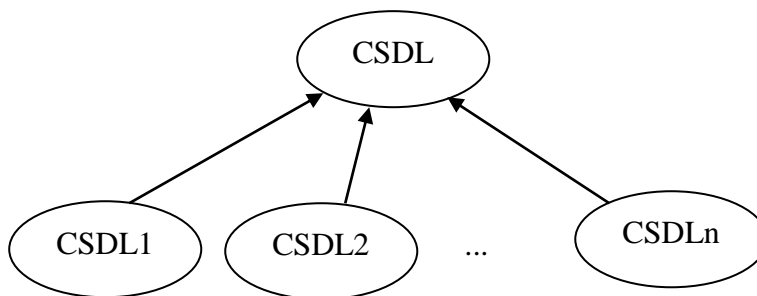
Một tính chất quan trọng của các CSDL phân tán là *tính thuần nhất* hay *không thuần nhất*. Có thể chia CSDL phân tán thành hai loại: CSDL phân tán thuần nhất và CSDL phân tán không thuần nhất.

Một CSDL phân tán thuần nhất có được bằng cách chia một CSDL thành một tập các CSDL cục bộ, mỗi CSDL cục bộ này được quản trị bởi cùng một hệ quản trị CSDL (Hình 9.4a). Một CSDL phân tán thuần nhất thường là kết quả của cách tiếp cận thiết kế từ trên xuống, trong đó chúng ta thiết kế một CSDL phân tán từ một CSDL tập trung. Nó tạo thuận lợi khi cần thêm một trạm mới vào mạng, đồng thời còn tăng hiệu năng của việc quản trị dữ liệu bằng cách khai thác các khả năng xử lý song song được cung cấp bởi sự có mặt của nhiều trạm.

Một CSDL phân tán không thuần nhất là một CSDL phân tán có được bằng cách tích hợp một tập các CSDL cục bộ được quản trị bởi các hệ quản trị CSDL khác nhau, thành một CSDL duy nhất (Hình 9.4b). Rõ ràng cách tiếp cận này phức tạp hơn nhiều so với cách tiếp cận thuần nhất. Nó thích hợp cho cách tiếp cận thiết kế từ dưới lên, cho phép tích hợp các CSDL hiện có và độc lập với nhau, không phải tạo ra một CSDL phân tán hoàn toàn mới. Khi đó, ngoài các chức năng đã chỉ rõ, phần mềm quản trị CSDL phân tán không thuần nhất phải cung cấp các giao diện để các hệ quản trị CSDL giao tiếp được với nhau.



a. Thuần nhất: Chia một CSDL thành các CSDL cục bộ



b. Không thuần nhất: Tích hợp các CSDL hiện có thành một CSDL

Hình 9.4. CSDL thuần nhất và không thuần nhất

Vì có nhiều cách tiếp cận khác nhau cho việc phân tán dữ liệu, chúng ta cần phân biệt các khái niệm sau đây: CSDL từ xa (remote database), CSDL liên hiệp (federated database), CSDL song song (parallel database).

9.1.2.1. CSDL từ xa

CSDL từ xa là một CSDL ở trên một máy tính khác với máy tính của người dùng và được truy cập nhờ vào các lệnh truyền thông được xác định bởi người dùng.

Thường là máy tính của người dùng duy trì một CSDL riêng được nạp từ CSDL từ xa. Những tiền bộ của các mạng cục bộ và các trạm làm việc cho phép tổ chức máy khách – máy chủ, trong đó CSDL được quản trị bởi một hệ quản trị CSDL trên máy chủ, còn các CSDL riêng được quản trị với một phiên bản của cùng hệ quản trị CSDL trên một trạm làm việc.

Người dùng phải đảm bảo sự nhất quán giữa dữ liệu trong CSDL riêng với các dữ liệu của CSDL từ xa.

Nói chung, việc truy cập từ xa được giới hạn trong việc tra cứu thông tin, các phép cập nhật được làm theo cách tập trung.

9.1.2.2. CSDL liên hiệp

Một CSDL liên hiệp không nhất thiết phải phân tán trên nhiều máy và cách tiếp cận này nhằm ưu tiên người dùng thao tác dữ liệu trên CSDL riêng của người đó và xem việc truy cập đến quá một CSDL chỉ là đột xuất.

9.1.2.3. CSDL song song

Một kiểu khai thác trong những năm gần đây của công nghệ CSDL phân tán là phân tán các dữ liệu trên một tập các nút của một máy tính song song (bộ đa xử lý) để tăng hiệu quả của việc quản lý dữ liệu. Khi các bộ xử lý của máy tính đó liên lạc, trao đổi với nhau bằng cách thông báo mà không phải bởi một bộ nhớ chung thì ta gọi cách tiếp cận đó là CSDL song song, về nhiều mặt giống với một CSDL phân tán thuần nhất. Như vậy, một CSDL song song có thể hiểu là một CSDL phân tán thuần nhất, trong đó các trạm là các nút của một máy tính song song và trao đổi/liên lạc với nhau bằng các thông báo. Sự khác nhau cơ bản giữa một CSDL song song và một CSDL phân tán thuần nhất là một nút của một máy tính song song không là một trạm mà tại đó người dùng có thể thực thi được một chương trình ứng dụng. Các chương trình ứng dụng được thực hiện bởi các máy chủ hay các trạm làm việc, trao đổi/liên lạc với máy tính song song nhờ vào một giao diện đặc biệt.

9.2. Các ưu và nhược điểm của việc phân tán dữ liệu

9.2.1. Các ưu điểm của việc phân tán dữ liệu

- **Lợi điểm về tổ chức và tính kinh tế:** Việc dùng CSDL phân tán phù hợp đối với tổ chức phân tán thành nhiều chi nhánh. Điều này có vai trò thúc đẩy kinh tế thương mại phát triển rộng hơn, vì việc phát triển các trung tâm máy tính phân tán ở nhiều vị trí đang trở thành nhu cầu cần thiết.
- **Tận dụng những cơ sở dữ liệu sẵn có:** Có thể hình thành CSDL phân tán từ các CSDL tập trung có sẵn.

- **Thuận lợi cho nhu cầu phát triển:** Xu hướng dùng cơ sở dữ liệu phân tán sẽ cung cấp khả năng phát triển thuận lợi hơn, giảm được xung đột về chức năng giữa các đơn vị đã tồn tại, và giảm xung đột giữa các chương trình ứng dụng khi truy cập đến CSDL. Với hướng tập trung hoá, nhu cầu phát triển trong tương lai sẽ gặp khó khăn.
- **Giảm chi phí truyền thông:** Trong CSDL phân tán, chương trình ứng dụng có thể giảm bớt được chi phí truyền thông khi thực hiện bằng cách khai thác CSDL tại chỗ.
- **Tăng số công việc thực hiện:** Hệ CSDL phân tán có thể tăng số lượng công việc thực hiện qua áp dụng nguyên lý xử lý song song với hệ thống xử lý đa nhiệm. Tuy nhiên CSDL phân tán cũng có tiện lợi trong việc phân tán dữ liệu như tạo ra các chương trình ứng dụng phụ thuộc vào tiêu chuẩn mở rộng vị trí làm cho các nơi xử lý có thể hỗ trợ lẫn nhau. Do đó tránh được hiện tượng tắc nghẽn cổ chai trong mạng truyền thông hoặc trong các dịch vụ thông thường của toàn bộ hệ thống.
- **Tính dễ hiểu và sẵn sàng:** Hướng phát triển CSDL phân tán cũng nhằm đạt được tính dễ hiểu và tính sẵn sàng cao hơn. Tuy nhiên để đạt được mục tiêu này không phải là dễ làm và đòi hỏi sử dụng kỹ thuật phức tạp. Khả năng xử lý tự trị của các điểm làm việc khác nhau không đảm bảo tính dễ sử dụng.

Hai nguyên nhân về mặt kỹ thuật đáp ứng cho sự phát triển hệ CSDL phân tán:

- Công nghệ tạo ra máy tính nhỏ và nền tảng phần cứng có khả năng phục vụ xây dựng hệ thống thông tin phân tán.
- Kỹ thuật thiết kế hệ CSDL phân tán được phát triển vững chắc dựa trên hai kỹ thuật thiết kế chính là Từ trên xuống và Từ dưới lên từ những năm thập kỷ 60.

Kỹ thuật thiết kế CSDL phân tán phức tạp nhưng hệ CSDL phân tán cần thiết cho xu hướng phát triển kinh tế hiện nay.

9.2.2. Các nhược điểm của việc phân tán dữ liệu

Nhược điểm đầu tiên của hệ thống CSDL phân tán là việc tăng thêm sự phức tạp khi xử lý phân tán. Phức tạp tăng lên dưới các dạng sau:

- **Giá phát triển phần mềm:** Do các khó khăn khi cài đặt một hệ thống phân tán, giá thành sẽ tăng lên.
- **Dễ mắc lỗi hơn:** Vì các trạm trong hệ phân tán làm việc song song, khó có thể đảm bảo thuật toán được thực hiện đúng trên tất cả các trạm. Do vậy mà số lỗi sẽ tăng lên. Các thuật toán phù hợp với CSDL phân tán còn đang được nghiên cứu.
- **Khối lượng các xử lý tăng:** Hệ thống phân tán cần truyền nhiều thông báo, nhiều tính toán phụ. Do vậy khối lượng xử lý tăng lên so với hệ thống tập trung. Khi chọn thiết kế hệ thống CSDL, người thiết kế cần phải cân đối các ưu điểm và nhược điểm của việc phân tán các dữ liệu. Có dải rộng các thiết kế rất phân tán đến các hệ thống tập trung.

9.3. Quản lý giao dịch phân tán

Một giao dịch phân tán gồm nhiều giao dịch con, mỗi giao dịch này thực hiện ở một trạm khác nhau. Môi trường phân tán ảnh hưởng đến các khía cạnh của việc quản lý các giao dịch: điều khiển tương tranh, xử lý hợp thức hoá (validation)...

9.3.1. Điều khiển tương tranh

Điều khiển tương tranh phân tán là chức năng ngăn chặn việc sản sinh ra các thực hiện không khả tuần tự của các giao dịch phân tán.

Hai cách tiếp cận cơ sở cho việc điều khiển tương tranh là thời dấu (time stamp) và đặt khóa (locking).

9.3.1.1. Thời dấu (time stamp)

Thời dấu bao gồm việc sắp xếp các giao dịch phân tán khi cho thực hiện chúng và áp đặt các thao tác truy cập tới dữ liệu tôn trọng thứ tự được xác định trước.

Với cách tiếp cận phân tán, mỗi trạm đánh thời dấu một cách tự trị theo đồng hồ địa phương (hay bộ đếm địa phương) và số hiệu của trạm. Mỗi giao dịch được khởi phát ở trạm i tại thời điểm t sẽ có thời dấu $\langle t, i \rangle$.

Khái niệm 1: Các đồng hồ được đồng bộ hoá

Các đồng hồ của các trạm khác nhau được đồng bộ nếu:

- (1) Mỗi thông báo đều có thời dấu là giá trị hiện thời của đồng hồ trạm phát.
- (2) Khi nhận được thông báo của có thời dấu t lớn hơn giá trị của đồng hồ địa phương của nó, trạm thu tự đồng bộ với trạm phát bằng cách gán giá trị $(t+1)$ cho đồng hồ địa phương của nó.

Bằng cách trao đổi có hệ thống giá trị hiện có của đồng hồ của chúng, các trạm trao đổi thông báo như vậy nhận được các đồng hồ được đồng bộ hoá. Trường hợp khi hai trạm không có sự hợp tác, việc đồng bộ hoá các đồng hồ của chúng đương nhiên là không cần thiết.

Thuật toán sắp trình tự bởi thời dấu có thể tóm tắt như sau:

- Hãy xét sự truy cập của giao dịch T_i tới một hạt dữ liệu mà thời dấu (của giao dịch cuối cùng đã truy cập nó để đọc hay viết) là j .
- Nếu $j \leq i$, khi đó truy cập bởi T_i tôn trọng thứ tự tới các giao dịch và có thể được thực hiện.
- Trong trường hợp ngược lại, giao dịch T_i phải bị huỷ bỏ và khôi phục với một thời dấu mới lớn hơn j .

Điều khiển tương tranh bằng thời dấu cho phép một giao dịch được tiến triển tự do, chừng nào nó không vi phạm thứ tự các giao dịch.

Vấn đề chính của cách tiếp cận này là số các huỷ bỏ và phục hồi các giao dịch (là những giao dịch tồn kém trong một hệ phân tán) có thể là rất lớn khi có mức truy cập tương tranh cao.

9.3.1.2. Sự đặt khóa (locking)

Việc truy cập tới các hạt dữ liệu được chi phối bởi một giao thức nói chung bao gồm hai thao tác nguyên thủy:

- LOCK: Yêu cầu thao tác trên các hạt dữ liệu theo một cách thức nhất định.
- UNLOCK: báo hiệu sự kết thúc của thao tác trên hạt.

Việc truy cập tới một hạt bị khóa dẫn đến việc giao dịch truy cập phải chờ cho tới khi giải phóng khóa bởi một giao dịch nào đó tác động lên hạt.

Để chỉ sinh ra các thực hiện khả tuần tự, giao thức áp đặt hai giai đoạn cho các giao dịch:

- Giai đoạn đặt khóa, trong đó nhận được tất cả các khóa cần thiết.
- Giai đoạn bỏ khóa, trong đó một khóa đều được giải phóng.

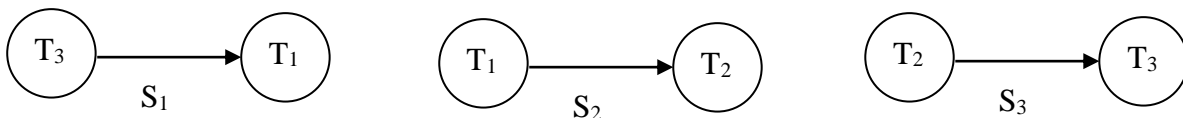
Vấn đề khó nhất do việc đặt khóa là tình huống bế tắc (deadlock - khóa chết hay nghẽn khóa). Ký hiệu " $T_1 \rightarrow T_2$ " biểu thị quan hệ T_1 "chờ đợi" sự giải phóng của một khóa nhận được bởi T_2 . Trong tình huống đơn giản nhất, bế tắc nảy sinh bởi ($T_1 \rightarrow T_2$, $T_2 \rightarrow T_1$), trong đó mỗi giao dịch đều bị kẹt nghẽn (vô hạn) trong chờ đợi việc giải phóng một khóa được đặt bởi giao dịch kia.

Việc phát hiện một chu trình trong đồ thị đợi chỉ ra sự có mặt của bế tắc (còn gọi là nghẽn). Việc giải quyết tắc nghẽn là loại bỏ chu trình bằng việc hủy bỏ một trong các giao dịch có liên quan.

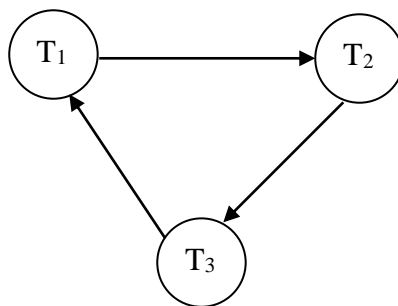
Ví dụ 9.1: Đồ thị đợi tại các trạm S_1, S_2, S_3

Nếu chúng ta xây dựng đồ thị đợi cho từng trạm riêng thì không có chu trình nào trong đồ thị, nghĩa là không tồn tại khóa chết (Hình 9.4). Tuy nhiên, nếu chúng ta kết hợp các đồ thị đợi (Hình 9.5), chúng ta có thể thấy khóa chết tồn tại với chu trình:

$$T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_1$$



Hình 9.4. Đồ thị đợi tại các trạm S_1, S_2, S_3



Hình 9.5. Kết hợp đồ thị đợi tại các trạm S_1, S_2, S_3

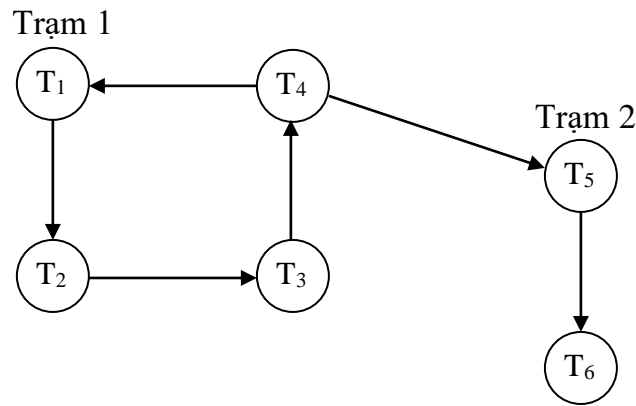
Khái niệm 2: Bế tắc toàn cục

Tình huống thường trực của một nhóm các giao dịch thực hiện ở nhiều hơn một trạm sao cho mỗi giao dịch của nhóm bị nghẽn lại trong khi đợi một hạt, và không một giao dịch nào của nhóm có thể được giải toả bởi việc thực hiện của một giao dịch lạ không thuộc nhóm.

Giải pháp thông dụng nhất là sự phát hiện và giải quyết vấn đề tắc nghẽn toàn cục, bằng cách dùng một *đồ thị đợi toàn cục* biểu thị tất cả các quan hệ đợi giữa các giao dịch đang hoạt động trong CSDL phân tán.

Ví dụ 9.2:

Hình 9.6 cho ví dụ về một đồ thị đợi toàn cục bao gồm có bốn giao dịch trong tình huống bế tắc, đó là T₁, T₂, T₃, T₄. Bốn giao dịch này tạo thành một chu trình.



Hình 9.6. Đồ thị đợi toàn cục có các giao dịch trong tình huống bế tắc

9.3.2. Sự hợp thức hoá giao dịch

Sự hợp thức hoá một giao dịch là tích hợp thực sự các cập nhật của nó vào CSDL.

Một phương pháp trực tiếp để hợp thức một giao dịch phân tán là xem mỗi giao dịch con thành phần như một giao dịch cục bộ có thể hợp thức hoá độc lập với các giao dịch cục bộ khác.

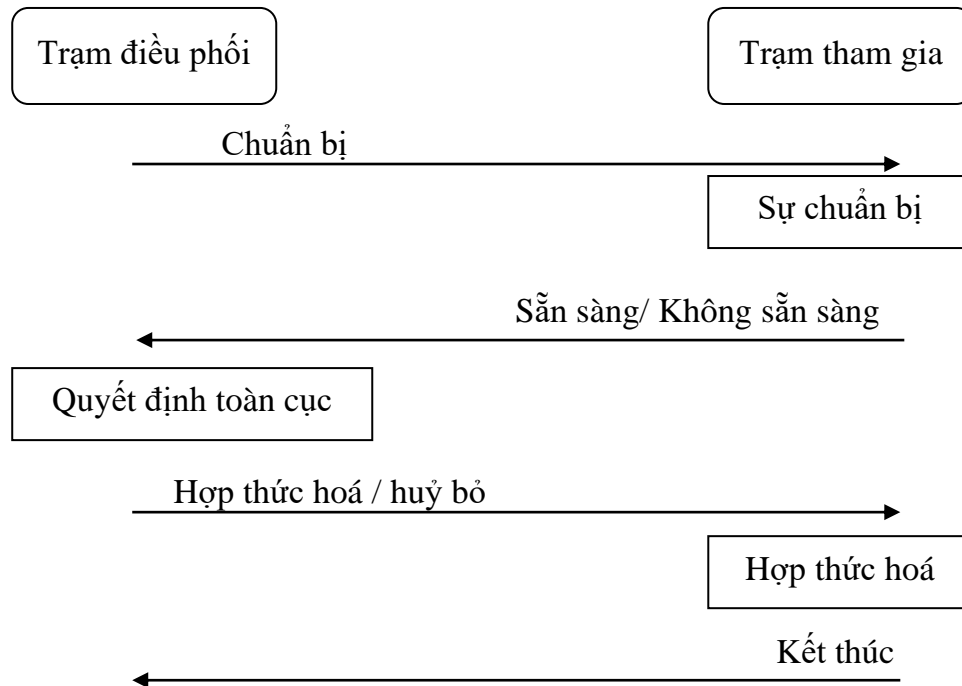
Tuy nhiên, một số giao dịch con có thể được hợp thức hoá trong khi những giao dịch con khác lại bị huỷ bỏ. Giải pháp là sử dụng một giao thức hợp thức hoá hai giai đoạn. Giao thức này được thực hiện bởi các trạm tham gia giao dịch dưới sự điều khiển của một trạm chủ được gọi là trạm điều phối.

Khái niệm 3: Giao thức hợp thức hoá hai giai đoạn

Giao thức gồm hai giai đoạn sau:

- (1) Trong giai đoạn chuẩn bị, trạm điều phối yêu cầu mỗi trạm tham gia chuẩn bị cho sự hợp thức hoá
- (2) Trong giai đoạn hợp thức, trạm điều phối ra lệnh cho tất cả các trạm tham gia hợp thức hoá các cập nhật của chúng nếu như tất cả chúng đều hoàn thành đúng giai đoạn thứ nhất, nếu không thì huỷ bỏ chúng.

Tóm tắt giao thức:

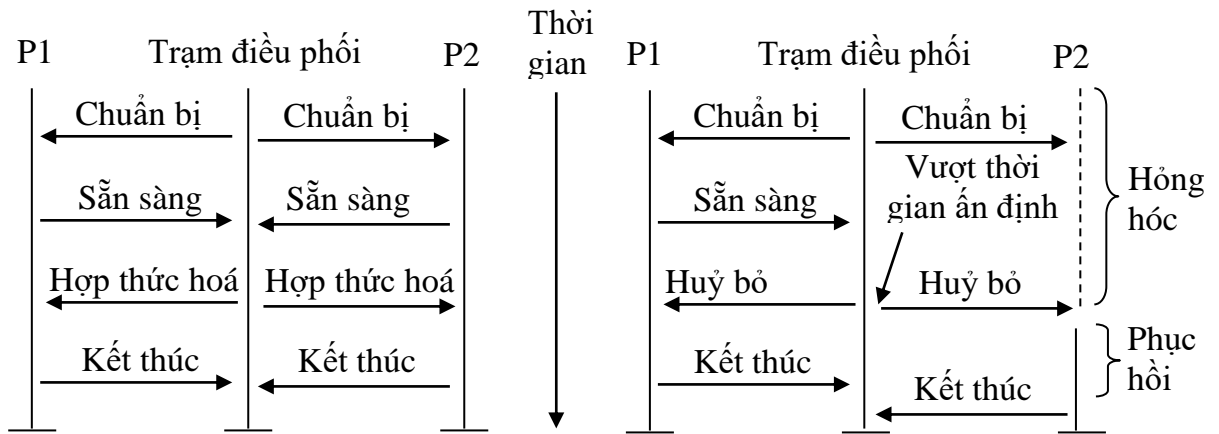


Hình 9.7. Hợp thức hoá hai giai đoạn

- Trạm điều phối khởi phát giai đoạn chuẩn bị bằng việc gửi thông báo "chuẩn bị" cho mọi trạm tham gia.
- Nhận được thông báo "chuẩn bị", mỗi trạm tham gia thực hiện các thao tác đảm bảo cho nó có thể hợp thức hay huỷ bỏ giao dịch con khi xảy ra bất kỳ sự cố nào, đó là thao tác ghi nhật ký và điều khiển tương tranh.
- Nếu trạm tham gia có khả năng hợp thứ, nói gửi thông báo "sẵn sàng" cho trạm điều phối, trường hợp ngược lại là thông báo "không sẵn sàng", chẳng hạn do sự cố hay do bế tắc cục bộ.
- Khi đó trạm điều phối sẽ có quyết định toàn cục là hợp thức hoá giao dịch nếu nhận được thông báo "sẵn sàng". Trường hợp có ít nhất một trạm tham gia trả lời "không sẵn sàng" hay đã vượt quá thời gian quy định mà không có trả lời (do có hồng hóc) thì quyết định toàn cục là huỷ bỏ.
- Nhận được quyết định toàn cục, mỗi trạm tham gia sẽ hợp thức hoặc huỷ bỏ các cập nhật của giao dịch con của nó và thông báo cho trạm điều phối là đã kết thúc công việc.

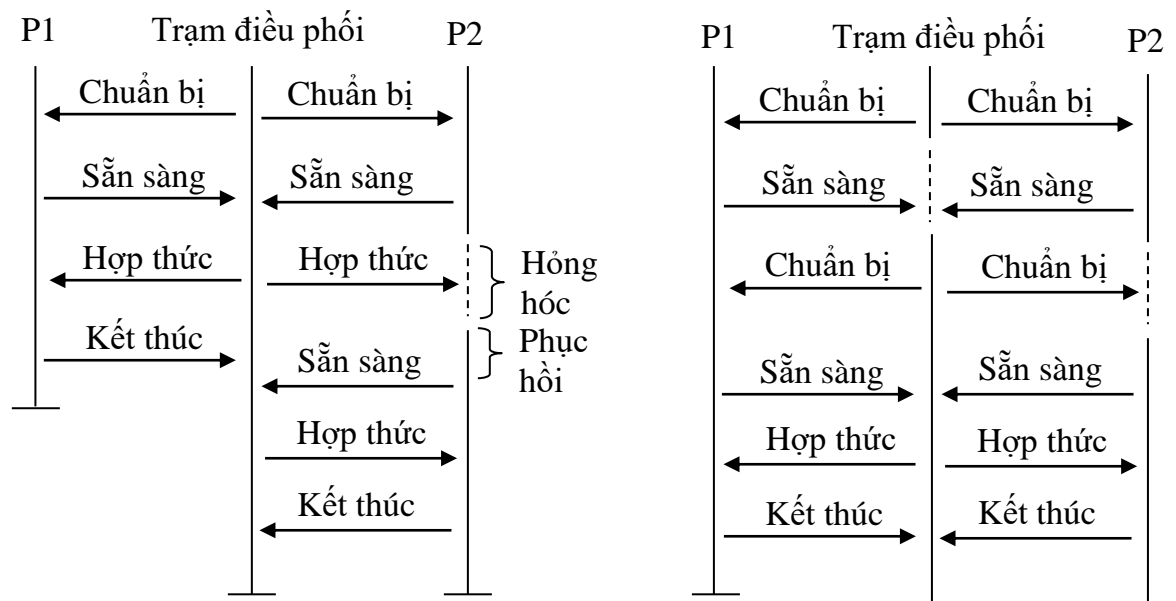
Giao thức hợp thức hoá hai giai đoạn đảm bảo tính nguyên tố của các giao dịch chống lại mọi kiểu hồng hóc trừ ra hồng hóc thảm hoạ (phá huỷ cả nhật ký).

Hình 9.8 mô tả nhiều lịch sử của các hợp thức hoá bao gồm một trạm điều phối và hai trạm tham gia P1 và P2.



a. Sự hợp thức bình thường

b. Hỏng hóc của P2 trước khi sẵn sàng



c. Hỏng hóc của P2 sau khi tuyên bố sẵn sàng

d. Hỏng hóc của trạm điều phối trước nhận được thông báo "sẵn sàng"

Hình 9.8. Các ví dụ của sự hợp thức hoá hai giai đoạn

Giao thức hợp thức hóa hai giai đoạn bị kẹt vì khi trạm điều phối không hoạt động, các trạm tham gia đã tuyên bố sẵn sàng để hợp thức sẽ bị kẹt. Chúng phải chờ sự khôi phục của trạm điều phối. Vấn đề nảy sinh khi một trạm tham gia chuyển trực tiếp từ trạng thái “sẵn sàng” sang trạng thái “hợp thức hóa” hay “hủy bỏ”.

Giao thức có thể trở nên không bị kẹt nhờ vào một dịch chuyển giữa các trạng thái “sẵn sàng” và “hợp thức hóa”. Trạng thái trung gian “sẵn sàng để hợp thức hóa” một khi đã đạt được, chỉ còn một kết cục là sự hợp thức hóa.

Như vậy, chúng ta nhận được giao thức hợp thức hóa trong ba giai đoạn. Nếu trạm điều phối có trục trặc (bị hỏng hóc) và không có trạm tham gia nào nhận được thông báo “sẵn sàng hợp thức hóa”, khi đó nó có thể hủy bỏ các giao tác con để giải phóng các tài nguyên. Giao tác có thể được phục hồi đúng đắn khi trạm điều phối lại hoạt động trở lại.

TỔNG KẾT CHƯƠNG 9

- Một CSDL phân tán là một tập các CSDL cùng hợp tác hoạt động, được lưu trữ trên máy tính khác nhau, gọi là các trạm (sites), được nối với nhau bằng một mạng truyền thông. Một cách lý tưởng, người dùng tại một trạm bất kỳ có thể truy cập và thao tác dữ liệu trên bất cứ một trạm nào.
- Kiến trúc của một hệ CSDL phân tán: có nhiều máy tính được gọi là các trạm hay nút, các trạm được kết nối bởi một kiểu mạng truyền thông nào đó để truyền dữ liệu và các lệnh giữa các trạm.
- Hệ quản trị CSDL phân tán là một tập các phần mềm hệ thống bao gồm các phần mềm quản trị các dữ liệu phân tán, các phần mềm quản trị truyền thông và các hệ quản trị CSDL địa phương/cục bộ lưu trữ trên mỗi trạm của hệ CSDL phân tán.
- Ưu điểm của việc phân tán: chia sẻ dữ liệu và điều khiển phân tán, tin cậy và sẵn sàng, tăng tốc xử lý câu hỏi
- Nhược điểm của việc phân tán: giá phát triển phần mềm cao, dễ mắc lỗi hơn, khối lượng các xử lý tăng.
- Một giao dịch phân tán gồm nhiều giao dịch con, mỗi giao dịch này thực hiện ở một trạm khác nhau. Môi trường phân tán ảnh hưởng đến các khía cạnh của việc quản lý các giao dịch: điều khiển tương tranh, xử lý hợp thức hoá...
- Điều khiển tương tranh phân tán là chức năng ngăn chặn việc sản sinh ra các thực hiện không khả tuần tự của các giao dịch phân tán. Hai cách tiếp cận cơ sở cho việc điều khiển tương tranh là thời dấu (time stamp) và đặt khóa (locking).
- Sự hợp thức hoá một giao dịch là tích hợp thực sự các cập nhật của nó vào CSDL.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 9

- 9.1.** Nêu khái niệm về hệ CSDL phân tán (DDBMS), hệ quản trị CSDL phân tán.
- 9.2.** Hãy phân tích các ưu điểm tương đối của các CSDL tập trung và các CSDL phân tán.
- 9.3.** Nêu các cách thức để quản lý các giao dịch dữ liệu phân tán.
- 9.4.** Cho ví dụ về sự hợp thức hoá hai giai đoạn.

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] Nguyễn Trung Trực, Cơ sở dữ liệu phân bố, Nhà xuất bản quốc gia TP. Hồ Chí Minh, 2004
- [6] Phương Lan, Giáo trình nhập môn cơ sở dữ liệu, Nhà xuất bản lao động xã hội
- [7] Toby J.Teorey - Trần Đức Quang biên dịch, Mô hình hoá và thiết kế cơ sở dữ liệu, Nhà xuất bản thống kê, 2002
- [8] Thomas Connolly - Carolyn Begg, Database systems - a practical approach to design, implementation, and management, Forth edition
- [9] Các tài liệu khác trên internet