

TRƯỜNG ĐẠI HỌC TRÀ VINH
KHOA KỸ THUẬT & CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN



TÀI LIỆU GIẢNG DẠY
MÔN CƠ SỞ DỮ LIỆU

Tác giả: Phan Thị Phương Nam

Trà Vinh, ngày 29 tháng 12 năm 2024

Mục lục

1	GIỚI THIỆU TỔNG QUAN VỀ CƠ SỞ DỮ LIỆU	11
1.1	Khái niệm về cơ sở dữ liệu (CSDL)	11
1.1.1	Giới thiệu hệ thống tập tin cổ điển	11
1.1.2	Định nghĩa cơ sở dữ liệu:	13
1.1.3	Các đối tượng sử dụng CSDL	14
1.2	Hệ quản trị CSDL	15
1.2.1	Khái niệm	15
1.2.2	Các tính năng của Hệ quản trị CSDL	15
1.3	Hệ CSDL	16
1.3.1	Khái niệm về hệ CSDL	17
1.3.2	Các mức biểu diễn một CSDL	17
1.3.3	Sơ đồ tổng quát của Hệ CSDL	19
1.3.4	Ví dụ về CSDL	19
1.3.5	Một số đặc tính của hệ CSDL	20
1.3.6	Các cách tiếp cận của một CSDL	22
2	MÔ HÌNH THỰC THỂ KẾT HỢP	29
2.1	Mô hình dữ liệu quan niệm và quá trình thiết kế CSDL	30
2.2	Các thành phần cơ bản của mô hình thực thể - kết hợp	31
2.2.1	Thực thể - Tập thực thể	31
2.2.2	Thuộc tính (Attribute)	32
2.2.3	Mối kết hợp (Relationship)	32
2.2.4	Lược đồ E/R	32
2.2.5	Thể hiện của lược đồ E/R	33
2.2.6	Thể hiện của một mối kết hợp	34

2.2.7	Bản số	35
2.2.8	Thuộc tính trên mỗi quan hệ	37
2.2.9	Khóa	37
2.2.10	Tập thực thể yếu	38
2.3	Thiết kế	39
2.3.1	Các bước thiết kế	39
2.3.2	Nguyên lý thiết kế	39
2.3.3	Ví dụ minh họa	39
3	MÔ HÌNH DỮ LIỆU QUAN HỆ	43
3.1	Các khái niệm cơ bản	43
3.1.1	Thuộc tính	43
3.1.2	Quan hệ	44
3.1.3	Bộ giá trị	45
3.1.4	Thể hiện của quan hệ	46
3.1.5	Khóa	46
3.1.6	Lược đồ quan hệ (Relation Schema)	48
3.2	Các bước biến đổi cấu trúc từ thực thể kết hợp sang quan hệ dữ liệu	49
3.3	Các thao tác trên quan hệ	50
3.3.1	Phép thêm một bộ mới vào quan hệ	50
3.3.2	Phép loại bỏ bộ ra khỏi quan hệ	51
3.3.3	Phép 3.3.3. Phép sửa đổi giá trị của các thuộc tính của quan hệ	52
4	NGÔN NGỮ ĐẠI SỐ QUAN HỆ	55
4.1	Các khái niệm	55
4.1.1	Giới thiệu	55
4.1.2	Khái niệm đại số quan hệ	56
4.2	Sử dụng các phép toán quan hệ, đại số quan hệ viết câu truy vấn	58
4.2.1	Phép toán tập hợp	58
4.2.2	Phép toán quan hệ	62
4.2.3	Các phép toán khác	67
4.2.4	Các thao tác cập nhật trên quan hệ	73
4.3	Tối ưu hóa câu truy vấn	75

5	PHỤ THUỘC HÀM VÀ RÀNG BUỘC TOÀN VỆN CỦA LƯỢC ĐỒ QUAN HỆ	81
5.1	Phụ thuộc hàm (PTH)	81
5.1.1	Định nghĩa	81
5.1.2	Hệ luật dẫn Armstrong	82
5.1.3	Bao đóng của tập thuộc tính	84
5.1.4	Giải thuật tìm bao đóng của tập thuộc tính (X^+)	85
5.2	Phủ tối thiểu của tập PTH	86
5.3	Ràng buộc toàn vẹn (Integrity Constraint)	89
5.3.1	Định nghĩa	89
5.3.2	Các yếu tố của RBTV	89
5.3.3	Phân loại RBTV	93
6	CHUẨN HÓA LƯỢC ĐỒ QUAN HỆ	101
6.1	Chuẩn hóa CSDL	101
6.2	Khóa trên lược đồ quan hệ	102
6.2.1	Định nghĩa	102
6.2.2	Thuật toán tìm một khóa	103
6.2.3	Thuật toán tìm tất cả các khóa	104
6.3	Dạng chuẩn của lược đồ quan hệ	105
6.4	Kiểm tra dạng chuẩn	114
7	NGÔN NGỮ SQL	117
7.1	Giới thiệu	117
7.2	Ngôn ngữ SQL	118
7.2.1	Lệnh tạo và xóa cơ sở dữ liệu	122
7.2.2	Ngôn ngữ định nghĩa dữ liệu DDL	123
7.2.3	Ngôn ngữ thao tác dữ liệu	128
7.2.4	Sao lưu và phục hồi dữ liệu	146

DANH MỤC TỪ VIẾT TẮT

CSDL	Cơ sở dữ liệu
CTƯD	Chương trình ứng dụng
HQT	Hệ quản trị
NSD	Người sử dụng
PTH	Phụ thuộc hàm
RBTV	Ràng buộc toàn vẹn

Danh sách hình vẽ

1.1	Hệ thống quản lý bán hàng	12
1.2	Ba mức trừu tượng dữ liệu	18
1.3	Sơ đồ tổng quát của một hệ CSDL	19
1.4	Mô hình dữ liệu mạng	23
1.5	Mô hình dữ liệu phân cấp	24
1.6	Mô hình dữ liệu thực thể - kết hợp	25
1.7	Mô hình dữ liệu quan hệ	26
1.8	Mô hình dữ liệu hướng đối tượng	27
2.1	Quá trình thiết kế CSDL	30
2.2	Lược đồ E/R	33
2.3	Thể hiện mối kết hợp	34
2.4	<i>Mối quan hệ vai trò</i>	35
2.5	Thể hiện bản số (min, max) giữa các thực thể	36
2.6	<i>Mối quan hệ vai trò</i>	37
2.7	Thuộc tính trên mối quan hệ	37
2.8	Thuộc tính làm khóa chính	38
2.9	Tập thực thể yếu	38
2.10	Thiết kế mô hình E/R	40
5.1	Đồ thị biểu diễn lược đồ CSDL	98
5.2	Đồ thị biểu diễn lược đồ CSDL ‘Quản lý đề án công ty’	98
6.1	Các mức độ chuẩn hóa CSDL	101
7.1	Cấu trúc bảng NHANVIEN không cài đặt RBTv	123
7.2	Cấu trúc bảng NHANVIEN sau khi thực thi câu lệnh	124

7.3	Câu lệnh thêm thuộc tính Ghi_Chú vào bảng NHANVIEN	125
7.4	Xóa thuộc tính Ghi_Chú ra khỏi bảng NHANVIEN	126
7.5	Thêm dữ liệu vào bảng NHANVIEN	129
7.6	Sửa dữ liệu của nhân viên 003 trong bảng NHANVIEN	129
7.7	Sửa dữ liệu trong bảng NHANVIEN	130
7.8	Xóa nhân viên 005 trong bảng NHANVIEN	130
7.9	Hiển thị toàn bộ thông tin của các nhân viên	132
7.10	Chiếu lấy họ tên và phái của nhân viên	132
7.11	Cho biết các mức lương trùng nhau bị bỏ bớt	133
7.12	Sắp xếp dữ liệu trong bảng	133
7.13	Xem tất cả thông tin của nhân viên theo điều kiện cụ thể	136
7.14	Xem thông tin của nhân viên có sự liên kết nhiều bảng	137
7.15	Kết quả so sánh giá trị thuộc tính	137
7.16	Kết quả so sánh giá trị thuộc tính với tập giá trị	138
7.17	Kết quả so sánh giá trị thuộc tính thuộc phạm vi	138
7.18	Tìm kiếm thông tin dạng chuỗi tương đương	138
7.19	Tìm kiếm thông tin có giá trị NULL hoặc NOT NULL	139
7.20	Sử dụng hàm kết hợp trong truy vấn	139
7.21	Sử dụng truy vấn con dạng 1	141
7.22	Sử dụng truy vấn con dạng 2	142
7.23	Sử dụng truy vấn con dạng 3	142
7.24	Sử dụng truy vấn con dạng 4	143
7.25	Sử dụng truy vấn con dạng 5	144
7.26	Gom nhóm theo từng phòng ban tính lương trung bình	145
7.27	Những phòng có lương trung bình lớn hơn 2.500.000	146

Danh sách bảng

1.1	Cơ sở dữ liệu ‘Quản lý đề án’	20
3.1	Thể hiện của quan hệ MonHoc	46
4.1	Thể hiện của quan hệ NHANVIEN	58
4.2	Ví dụ về tính khả hợp của quan hệ R và quan hệ S	59
4.3	Ví dụ thực hiện phép hợp giữa quan hệ R và quan hệ S	59
4.4	Ví dụ thực hiện phép giao giữa quan hệ R và quan hệ S	60
4.5	Ví dụ thực hiện phép trừ giữa quan hệ R và quan hệ S	61
4.6	Ví dụ thực hiện phép chia các quan hệ	62
4.7	Ví dụ thực hiện phép tích đề các giữa quan hệ R và quan hệ S	67
4.8	Ví dụ thực hiện phép kết theta của quan hệ R và S	68
4.9	Ví dụ thực hiện phép kết bằng của quan hệ R và S	68
4.10	Ví dụ thực hiện phép kết tự nhiên của quan hệ R và S	69
4.11	Sử dụng phép kết nối nội cho 2 quan hệ R và S	69
4.12	Sử dụng phép kết nối trái cho 2 quan hệ R và S	70
4.13	Sử dụng phép kết nối phải cho 2 quan hệ R và S	71
4.14	Sử dụng các hàm kết hợp trên quan hệ R	72
4.15	Sử dụng các hàm kết hợp trên quan hệ R, có gom nhóm	73
6.1	Ví dụ minh họa bằng HOADON	106
7.1	Cơ sở dữ liệu minh họa ‘Quản lý đề án’	121
7.2	Các kiểu dữ liệu thường gặp trong các hệ QT CSDL	125
7.3	Tóm tắt các dạng trả về từ Select con	144

Chương 1

GIỚI THIỆU TỔNG QUAN VỀ CƠ SỞ DỮ LIỆU

♦ **Mục tiêu học tập:** Sau khi học xong chương này, người học có thể:

- Nắm được các khái niệm về CSDL, Hệ QT CSDL, Hệ CSDL,....
- Nắm được các cách tiếp cận CSDL thông qua các mô hình
- Phân biệt được các đối tượng sử dụng CSDL

1.1 Khái niệm về cơ sở dữ liệu (CSDL)

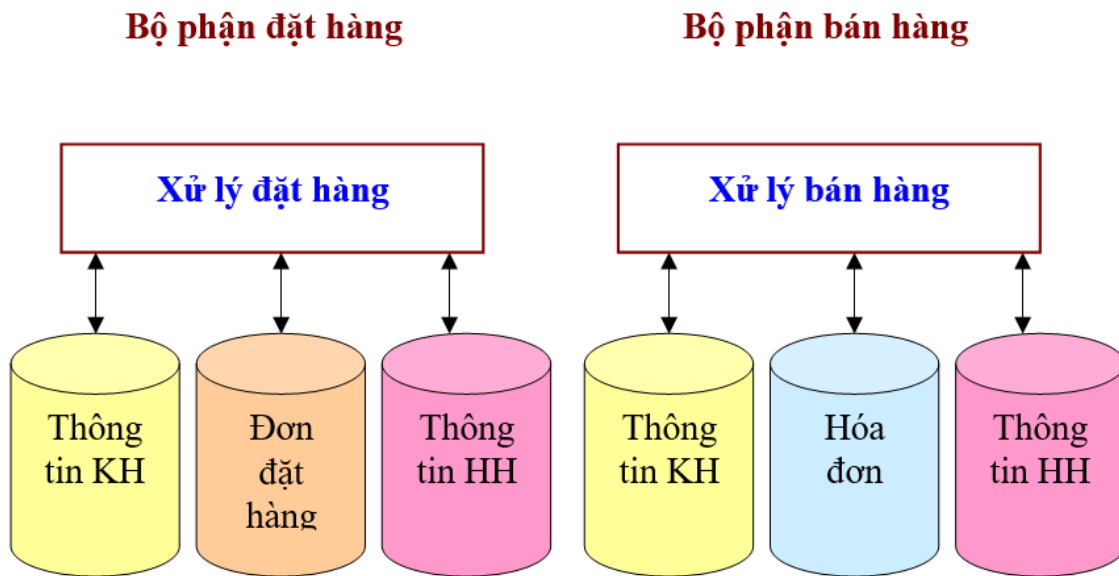
1.1.1 Giới thiệu hệ thống tập tin cổ điển

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

Tại một cửa hàng chuyên bán các mặt hàng, cửa hàng này cần có các bộ phận riêng lẻ để quản lý các việc khác nhau. Trong đó:

- Bộ phận đặt hàng cần lưu trữ thông tin về việc xử lý đơn đặt hàng từ thông tin khách hàng, thông tin về đơn đặt hàng, thông tin về hóa đơn và khách hàng đặt mua.
- Bộ phận bán hàng cần lưu trữ thông tin về việc xử lý bán hàng như thông tin khách hàng, thông tin về hóa đơn xuất hàng, thông tin về hàng hóa mua của khách hàng theo đơn khách hàng.

Cho hình minh họa:



Hình 1.1: Hệ thống quản lý bán hàng

a. **Ưu điểm:**

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

b. **Khuyết điểm:**

Do thông tin được tổ chức ở mỗi bộ phận đều khác nhau (thông tin và công cụ triển khai) nên rõ ràng sự phối hợp về mặt tổ chức và khai thác ở các bộ phận rất khó khăn. Thông tin ở bộ phận này không được sử dụng cho bộ phận khác, tại các chi nhánh, đại lý với trụ sở chính, cửa hàng gốc. Hơn nữa cùng một thông tin được cập nhật vào máy tính tại nhiều nơi sẽ khác nhau. Do đó có những hạn chế sau.

- Dữ liệu dư thừa và không nhất quán
 - Trùng lặp dữ liệu gây lãng phí không gian lưu trữ.
 - Dữ liệu không nhất quán do được cập nhật nhiều nơi khác nhau.

- Không uyển chuyển trong việc truy xuất dữ liệu, viết chương trình ứng dụng.
Chi phí bảo trì cao.
- Chia sẻ dữ liệu kém, khó khăn trong việc nâng cấp hệ thống.
 - Do hệ thống được tổ chức thành các tập tin riêng lẻ nên thiếu sự chia sẻ thông tin giữa các bộ phận.
 - Bên cạnh đó việc kết nối các hệ thống này với nhau hay nâng cấp ứng dụng sẽ gặp rất nhiều khó khăn.

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

1.1.2 Định nghĩa cơ sở dữ liệu:

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

Ưu điểm của CSDL:

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

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

- **Tính chủ quyền dữ liệu.** Do sự chia sẻ của CSDL nên tính chủ quyền của dữ liệu có thể bị lu mờ và làm mờ nhạt tinh thần trách nhiệm, được thể hiện trên vấn đề an toàn dữ liệu, khả năng biểu diễn các mối liên hệ ngữ nghĩa của dữ liệu và tính chính xác của dữ liệu. Nghĩa là người khai thác CSDL phải có nghĩa vụ cập nhật các thông tin mới nhất của CSDL.
- **Tính bảo mật và quyền khai thác thông tin của người sử dụng (NSD).** Do có nhiều người cùng khai thác CSDL một cách đồng thời nên cần phải có một cơ chế bảo mật và phân quyền khai thác CSDL.

- **Tranh chấp dữ liệu.** Nhiều người được phép truy cập vào cùng một tài nguyên dữ liệu (data source) của CSDL với những mục đích khác nhau như xem, thêm, xóa, sửa dữ liệu. Như vậy cần phải có cơ chế ưu tiên truy nhập dữ liệu cũng như cơ chế giải quyết tình trạng khoá chết (DeadLock) trong quá trình khai thác cạnh tranh.

Cơ chế ưu tiên có thể được thực hiện dựa trên:

- + Cấp quyền ưu tiên cho người khai thác CSDL, người nào có quyền ưu tiên cao hơn thì được ưu tiên cấp quyền truy cập dữ liệu trước.
 - + Thời điểm truy cập, ai có yêu cầu truy xuất trước thì có quyền truy cập dữ liệu trước.
 - + Hoặc dựa trên cơ chế lập lịch truy xuất hay cơ chế khoá.
 - + ...
- **Đảm bảo dữ liệu được an toàn khi có sự cố.** Việc quản lý tập trung có thể làm tăng khả năng mất mát hoặc sai lệch thông tin khi có sự cố như mất điện đột xuất, một phần hay toàn bộ đĩa lưu trữ dữ liệu bị hư, ... Một số hệ điều hành mạng có cung cấp dịch vụ sao lưu đĩa cứng (cơ chế sử dụng đĩa cứng dự phòng - RAID), tự động kiểm tra và khắc phục lỗi khi có sự cố. Tuy nhiên bên cạnh dịch vụ của hệ điều hành, một CSDL nhất thiết phải có một cơ chế khôi phục dữ liệu khi có sự cố bất ngờ xảy ra để đảm bảo CSDL luôn ổn định.

1.1.3 Các đối tượng sử dụng CSDL

- Người quản trị **CSDL** (Database Administrator - DBA)
Có trách nhiệm quản lý hệ CSDL
 - + Phải có kiến thức sâu về tin học, CSDL, hệ quản trị CSDL, hệ thống máy tính, hệ thống mạng. ...
 - + Có nhiệm vụ cấp quyền truy cập CSDL.
 - + Có nhiệm vụ điều phối và giám sát việc sử dụng CSDL.
 - + Có nhiệm vụ phục hồi dữ liệu khi có sự cố.
- Người phát triển ứng dụng (Database Designer, Developer, ...)
 - + Người phát triển ứng dụng (Database Designer, Developer, ...):

- Lựa chọn cấu trúc phù hợp để lưu trữ dữ liệu
 - Quyết định những dữ liệu nào cần được lưu trữ
 - Xây dựng ứng dụng để quản lý và khai thác CSDL
- + Khai thác CSDL qua chương trình ứng dụng và qua ngôn ngữ khai thác CSDL.
- Người dùng cuối (End User)
- + Không đòi hỏi phải có kiến thức về tin học, CSDL.
- + Khai thác CSDL thông qua **CTƯD**.

1.2 Hệ quản trị CSDL

1.2.1 Khái niệm

- Để giải quyết tốt những vấn đề đặt ra cho một CSDL như đã trình bày trong (mục 1.1.2) thì cần có một hệ thống các phần mềm chuyên dụng. Hệ thống này được gọi là hệ quản trị CSDL (Database Management System - DBMS), là *tập hợp các chương trình* cho phép người sử dụng định nghĩa, duy trì, khai thác và quản lý CSDL.
- Hiện nay có nhiều hệ quản trị CSDL trên thị trường như: MySQL, SQL Server, DB2, PostgreSQL, Oracle, Infomix, ... Phần lớn các hệ điều hành này hoạt động tốt trên nhiều hệ điều hành khác nhau như Linux, Unix, ... riêng SQL Server chỉ chạy trên hệ điều hành Windows.

1.2.2 Các tính năng của Hệ quản trị CSDL

Các hệ quản trị CSDL ngày nay đòi hỏi phải đảm bảo các tính năng sau:

- Ngôn ngữ giao tiếp giữa người sử dụng (NSD) và CSDL:
 - + Ngôn ngữ định nghĩa dữ liệu (Data Definition Language - DDL) cho phép NSD khai báo cấu trúc của CSDL, khai báo các mối liên hệ của dữ liệu (Data Relationship) và các quy tắc (Rules, Constraint) quản lý áp đặt trên dữ liệu.
 - + Ngôn ngữ thao tác dữ liệu (Data Manipulation Language - DML) cho phép NSD có thể thêm, xóa, dữ liệu trong CSDL.

- + Ngôn ngữ truy vấn dữ liệu (Structured Query Language - SQL) cho phép NSD truy vấn các thông tin cần thiết.
- + Ngôn ngữ quản lý dữ liệu (Data Control Language - DCL) cho phép những người quản trị hệ thống thay đổi cấu trúc của các bảng dữ liệu, khai báo bảo mật thông tin và cấp quyền khai thác CSDL cho NSD.
- **Từ** điển dữ liệu (Data dictionary) dùng để mô tả các ánh xạ liên kết, ghi nhận các thành phần cấu trúc của CSDL, các chương trình ứng dụng, mật mã, quyền sử dụng. . .
- Chia sẻ dữ liệu: Trong môi trường đa người dùng, các HQT phải cho phép truy xuất dữ liệu đồng thời.
- Hạn chế những truy cập không cho phép: Từng người dùng và nhóm người dùng có một tài khoản và mật mã để truy xuất dữ liệu.
- Cung cấp nhiều giao diện: HQT cung cấp ngôn ngữ giữa CSDL và người dùng.
- Đảm bảo các ràng buộc toàn vẹn (RBTV)
 - + RBTV (Integrity Constraints) là những qui định cần được thỏa mãn để đảm bảo dữ liệu luôn phản ánh đúng ngữ nghĩa của thế giới thực.
 - + Một số RBTV có thể được khai báo với HQT và HQT sẽ tự động kiểm tra. Một số RBTV khác được kiểm tra nhờ CTƯD.
- Khả năng sao lưu dự phòng khi gặp sự cố: Có khả năng khôi phục dữ liệu khi có sự hư hỏng về phần cứng hoặc phần mềm.
- Bảo đảm tính độc lập giữa dữ liệu và chương trình: khi có sự thay đổi dữ liệu (như sửa đổi cấu trúc lưu trữ các bảng dữ liệu, thêm các chỉ mục, . . .) thì các chương trình ứng dụng đang chạy trên CSDL vẫn không cần phải viết lại hay cũng không ảnh hưởng đến NSD khác.

1.3 Hệ CSDL

1.3.1 Khái niệm về hệ CSDL

Hệ cơ sở dữ liệu bao gồm Cơ sở dữ liệu và hệ quản trị cơ sở dữ liệu.

1.3.2 Các mức biểu diễn một CSDL

Để hệ thống có thể dùng được thì phải tìm dữ liệu một cách hiệu quả. Điều này dẫn đến việc cần phải thiết kế những cấu trúc dữ liệu phức tạp đối với việc biểu diễn dữ liệu trong cơ sở dữ liệu. Vì nhiều người sử dụng các hệ thống cơ sở dữ liệu không phải là người hiểu biết nhiều về máy tính nên những người phát triển hệ thống phải che dấu đi sự phức tạp để người sử dụng thông qua một số mức trừu tượng, nhằm làm đơn giản hóa sự tương tác của người sử dụng đối với hệ thống.

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

a. Mức ngoài (External Level)

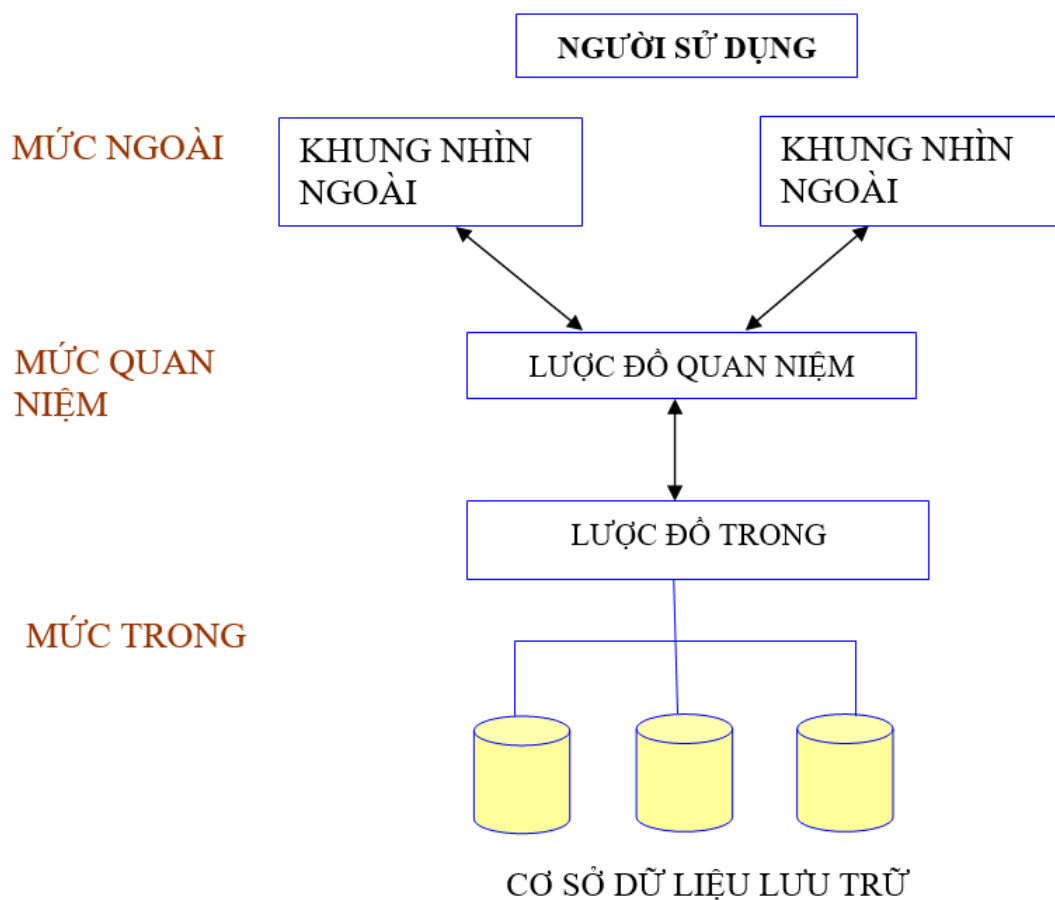
- Còn gọi là mô hình ngoài.
- Là mức của NSD và các chương trình ứng dụng.
- Làm việc tại mức này là các nhà chuyên môn, kỹ sư tin học, NSD không chuyên.
- Mỗi NSD hay mỗi chương trình ứng dụng có thể được nhìn CSDL theo một góc độ khác nhau. Tùy thuộc vào vai trò của NSD mà có thể thấy tất cả hoặc một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

b. Mức quan niệm (Conception level)

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

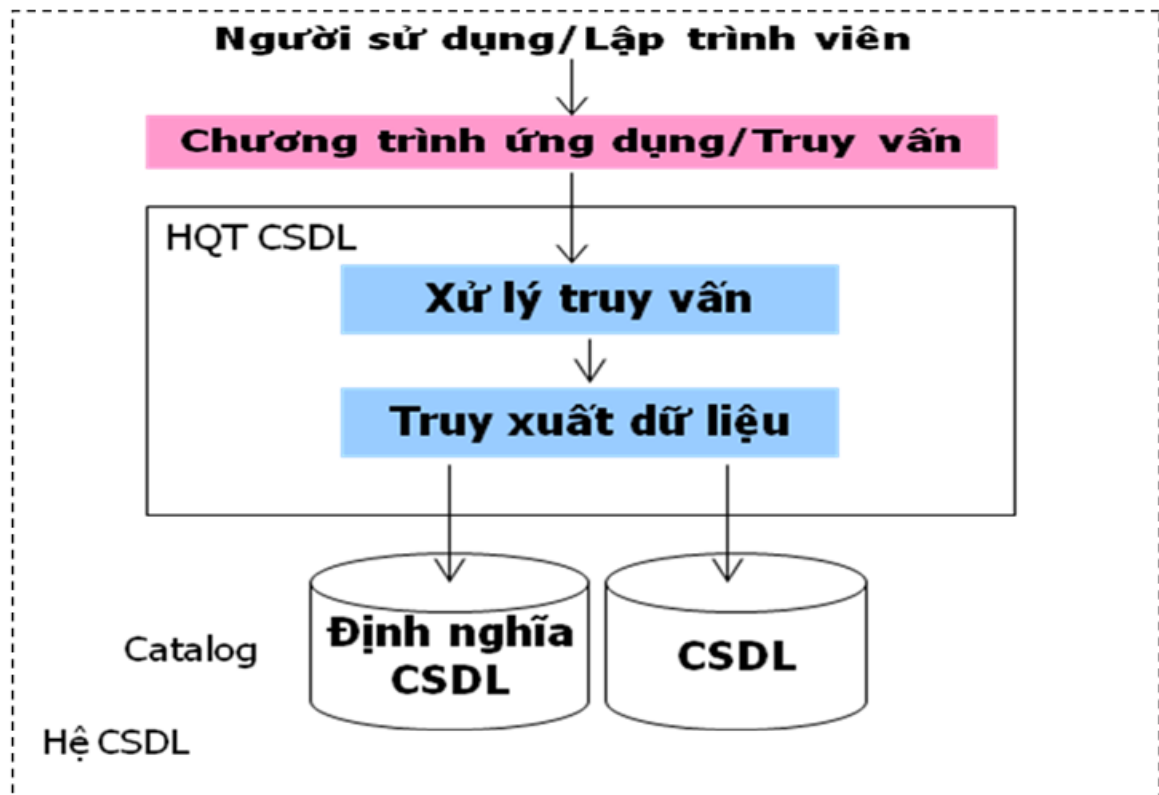
c. **Mức trong (Internal Level)**

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



Hình 1.2: Ba mức trừu tượng dữ liệu

1.3.3 Sơ đồ tổng quát của Hệ CSDL



Hình 1.3: Sơ đồ tổng quát của một hệ CSDL

1.3.4 Ví dụ về CSDL

Quản lý đề án của một công ty gồm ://không cần sửa vì chỉ là ví dụ để minh họa 1 CSDL.

NHANVIEN	HONV	TENLOT	TENNV	MANV	NGSINH	MA_NQL	PHG
	Tran	Hong	Quang	987987987	03/09/1969	987654321	4
	Nguyen	Thanh	Tung	333445555	12/08/1955	888665555	5
	Nguyen	Manh	Hung	666884444	09/15/1962	333445555	5
	Tran	Thanh	Tam	453453453	07/31/1972	333445555	5

DEAN	TENDA	MADA	DDIEM_DA	PHONG
	San pham X	1	VUNG TAU	5
	San pham Y	2	NHA TRANG	5
	San pham Z	3	TP HCM	5
	Tin hoc hoa	10	HA NOI	4

PHANCONG	MA_NVIENT	SODA	THOIGIAN
	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40.0
	453453453	1	20.0

Bảng 1.1: Cơ sở dữ liệu ‘Quản lý đề án’

- Định nghĩa CSDL
 - + Cấu trúc bảng, bao gồm các thành phần dữ liệu và kiểu dữ liệu tương ứng.
- Xây dựng CSDL
 - + Đưa dữ liệu vào các bảng.
- Xử lý CSDL
 - + Thực hiện các truy vấn: “Cho biết những nhân viên thuộc phòng 5”.
 - + Thực hiện các phép cập nhật: “Chuyển nhân viên Nguyễn Thanh Tùng sang phòng số 1”.

1.3.5 Một số đặc tính của hệ CSDL

- Tính tự mô tả:
 - + Hệ CSDL không chỉ chứa bản thân CSDL mà còn chứa định nghĩa đầy đủ (mô tả) của CSDL.
 - + Các định nghĩa được lưu trữ trong catalog, chứa các thông tin về cấu trúc tập tin, kiểu và dạng thức lưu trữ của mỗi thành phần dữ liệu và những ràng buộc dữ liệu.

- + Dữ liệu trong catalog gọi là meta-data.
- + Các CTƯD có thể truy xuất đến nhiều CSDL nhờ thông tin cấu trúc được lưu trữ trong catalog.

- Tính độc lập:

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

- Tính trừu tượng:

- + Hệ CSDL cho phép trình bày dữ liệu ở một mức trừu tượng cho phép, nhằm che bớt những chi tiết lưu trữ thật của dữ liệu.
- + Trừu tượng hóa dữ liệu – Mô hình dữ liệu.
- + Mô hình dữ liệu
 - Đối tượng (table)
 - Thuộc tính của đối tượng
 - Mối liên hệ

- Tính nhất quán

- + Lưu trữ dữ liệu thống nhất
- + Tránh được tình trạng trùng lặp thông tin
 - Có cơ chế điều khiển truy xuất dữ liệu hợp lý
- + Tránh được việc tranh chấp dữ liệu
- + Bảo đảm dữ liệu luôn đúng tại mọi thời điểm

1.3.6 Các cách tiếp cận của một CSDL

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

1.3.6.1 Mô hình dữ liệu mạng

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

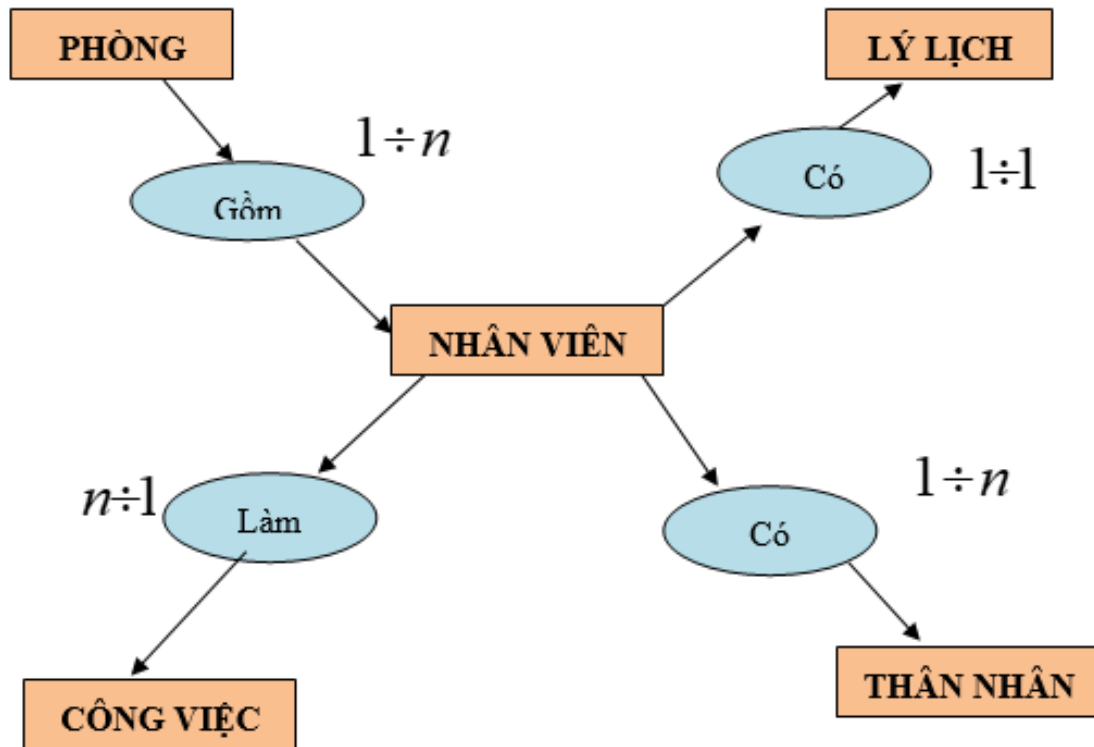
- **Loại mẫu tin:** đặc trưng cho một loại đối tượng riêng biệt. Chẳng hạn như trong quản lý nhân sự tại một đơn vị thì đối tượng cần phản ánh của thế giới thực có thể là phòng, nhân viên, công việc, lý lịch... do đó có các loại mẫu tin đặc trưng cho từng đối tượng này. Trong mô hình mạng, mỗi loại mẫu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

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

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

- + $1 \div 1$ (One to One): Mỗi mẫu tin của loại mẫu tin chủ kết hợp với đúng 1 mẫu tin của loại mẫu tin thành viên. Ví dụ, mỗi nhân viên có duy nhất một lý lịch cá nhân.
- + $1 \div n$ (One to Many): Mỗi mẫu tin của loại mẫu tin chủ kết hợp với 1 hay nhiều mẫu tin của loại mẫu tin thành viên. Ví dụ, mỗi phòng ban có thể có một hoặc nhiều nhân viên.

- + $n \div 1$ (Many to One): Nhiều mẫu tin của loại mẫu tin chủ kết hợp với đúng 1 mẫu tin của loại mẫu tin thành viên. Ví dụ, nhiều nhân viên cùng làm một công việc.
- + Đệ quy (Recursive): Một loại mẫu tin chủ cũng có thể đồng thời là mẫu tin thành viên với hính nó. Ví dụ, trưởng phòng cũng là một nhân viên thuộc phòng trong công ty.



Hình 1.4: Mô hình dữ liệu mạng

- + **Ưu điểm:** Đơn giản, dễ sử dụng.
- + **Hạn chế:** Không thích hợp trong việc biểu diễn CSDL lớn do hạn chế về khả năng diễn đạt ngữ nghĩa của dữ liệu, đặc biệt là các dữ liệu và mối liên hệ phức tạp của dữ liệu trong thực tế là rất hạn chế.

1.3.6.2 Mô hình dữ liệu phân cấp

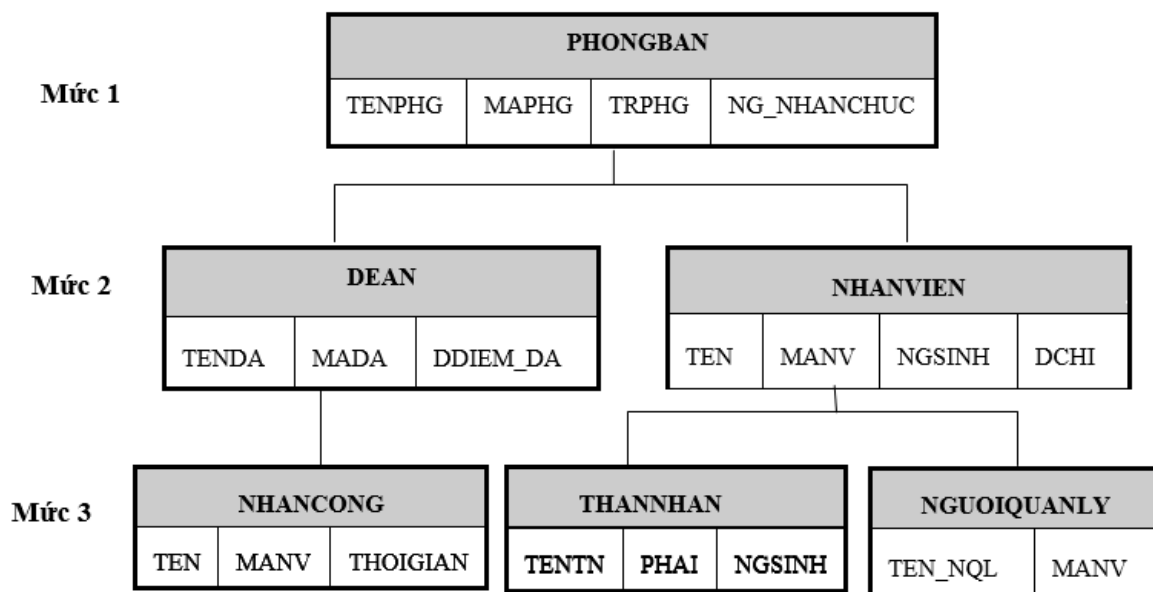
Mô hình dữ liệu phân cấp (**Hierarchical** Data Model), còn gọi là mô hình phân cấp (**Hierarchical** Model), được thực hiện thông qua sự kết hợp giữa IBM và North merican Rockwell vào khoảng năm 1965. Mô hình là một cây, trong đó mỗi nút của cây biểu diễn

một thực thể, giữa nút con với nút cha được liên hệ với nhau theo một mối quan hệ xác định.

- **Loại mẫu tin:** Giống trong mô hình dữ liệu mạng.
- **Loại mối liên hệ:** Kiểu liên hệ là phân cấp.

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

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

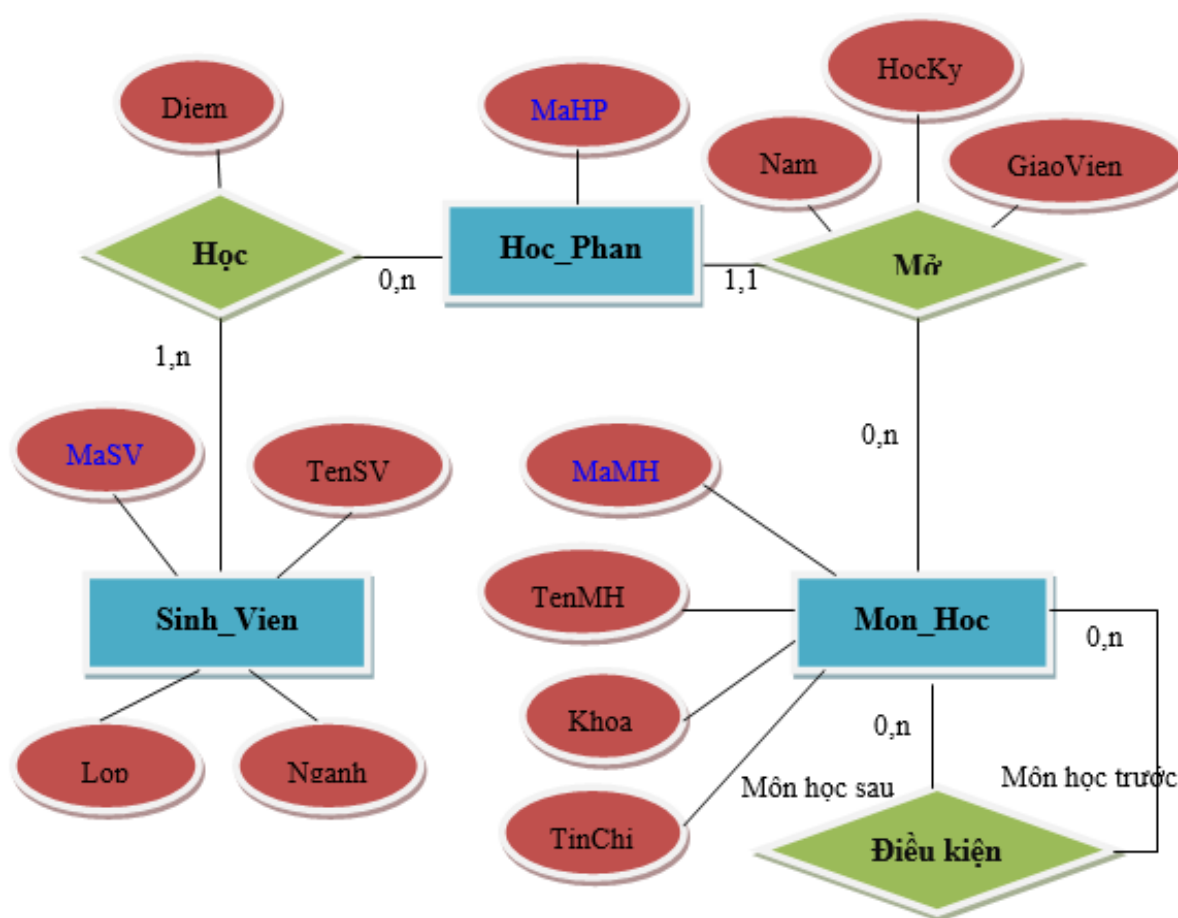


Hình 1.5: Mô hình dữ liệu phân cấp

1.3.6.3 Mô hình dữ liệu thực thể - kết hợp

Mô hình dữ liệu thực thể - kết hợp (Entity – Relationship Model) do Peter Pin_Shan Chen đề xuất năm 1976. Mô hình quan hệ-thực thể được dựa trên sự nhận thức thế giới gồm có một sưu tập các đối tượng căn bản, được gọi là các tập thực thể, và các mối quan hệ ở giữa các đối tượng này. Trong mô hình này các khái niệm được sử dụng là tập thực thể (entity set), thực thể (entity), thuộc tính của loại thực thể (entity attribute), khóa của loại thực thể (entity key), loại mối kết hợp (entity relationship), số ngôi của mối kết

hợp (relationship degree), thuộc tính của mỗi kết hợp (relationship attribute), bản số của mỗi kết hợp (relationship cardinal).



Hình 1.6: Mô hình dữ liệu thực thể - kết hợp

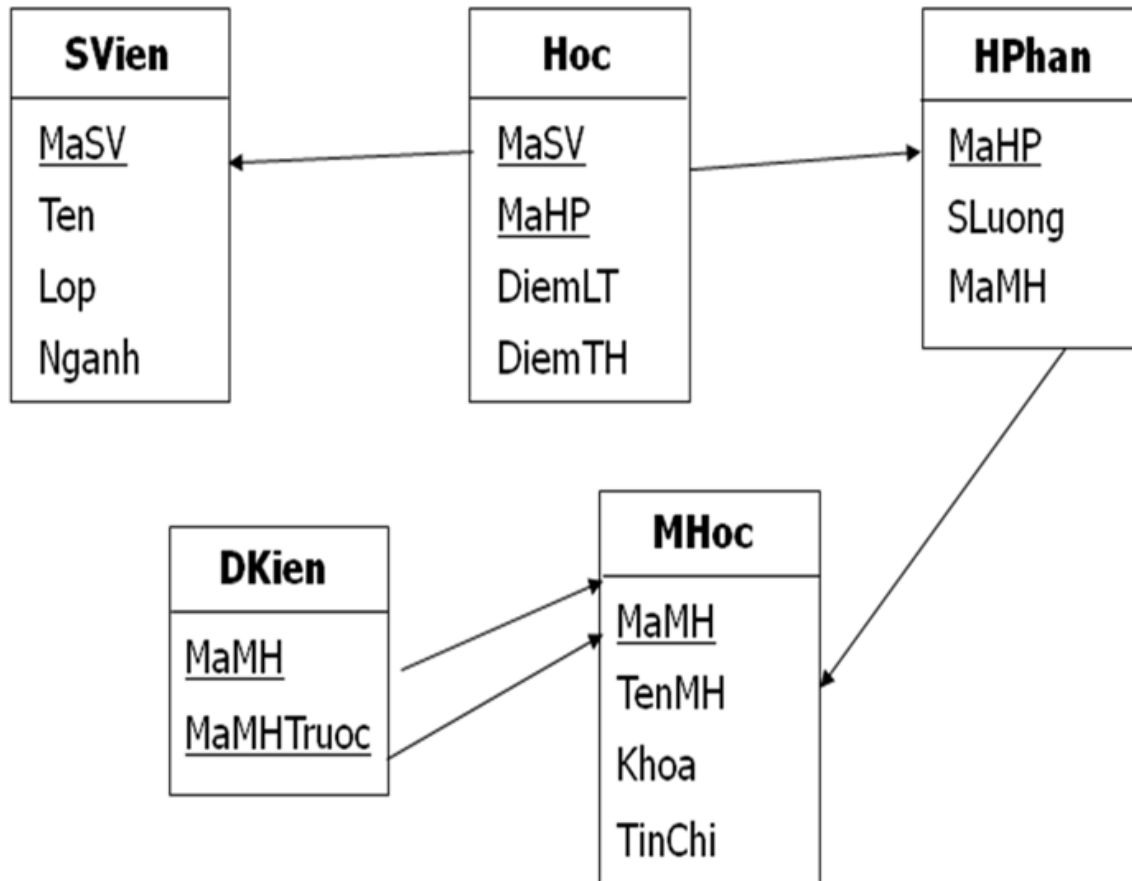
Chi tiết của mô hình này sẽ được nghiên cứu trong kết quả học tập 2.

1.3.6.4 Mô hình dữ liệu quan hệ

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

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

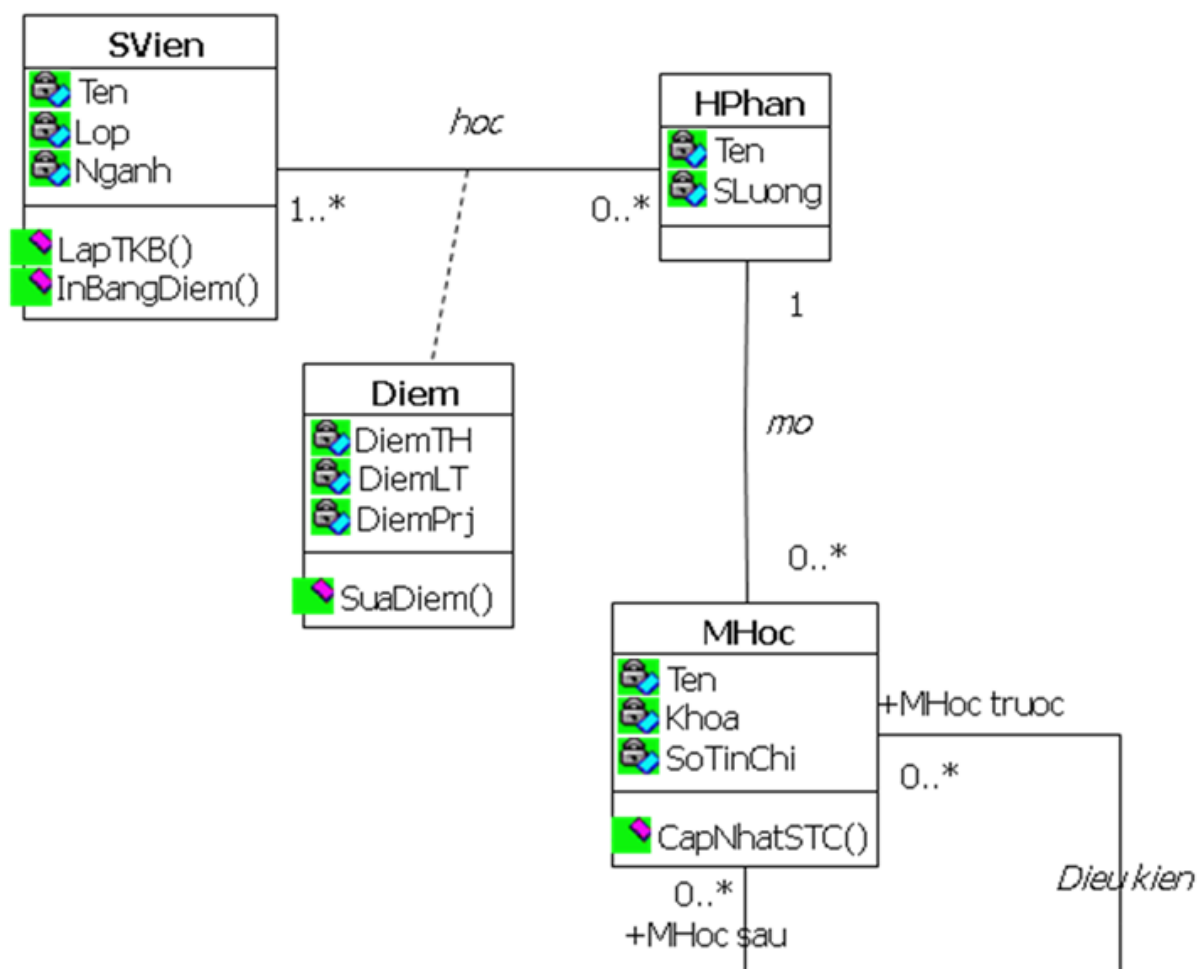
cứu chi tiết về mô hình này trong chương sau.



Hình 1.7: Mô hình dữ liệu quan hệ

1.3.6.5 Mô hình dữ liệu hướng đối tượng

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



Hình 1.8: Mô hình dữ liệu hướng đối tượng

► **Câu hỏi củng cố chương 1:**

Câu 1: CSDL đã được dùng rộng rãi trong cuộc sống. Ngoài hai ứng dụng CSDL vào lĩnh vực giáo dục và ngân hàng, hãy nêu thêm một số ví dụ khác mà anh/chị biết?

Câu 2: Hãy nêu ra 3 điểm khác nhau chủ yếu giữa một hệ thống xử lý tập tin cổ điển và một hệ quản trị CSDL? **Câu 3:** Phân biệt các thuật ngữ sau đây:

- Dữ liệu
- Cơ sở dữ liệu
- Hệ quản trị cơ sở dữ liệu
- Hệ cơ sở dữ liệu

Câu 4: 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 vật lý và tính độc lập logic.

Câu 5: Cho ví dụ về tính toàn vẹn dữ liệu và sự vi phạm tính toàn vẹn dữ liệu?

Câu 6: Hãy kể tên một số hệ quản trị CSDL mà anh/chị biết hoặc đã sử dụng?

Câu 7: Mô tả các thành phần chính của một hệ quản trị CSDL và chỉ ra các chức năng của chúng? **Câu 8:** Phân biệt vai trò của những nhóm người sau đây đối với một hệ CSDL:

- Người quản trị CSDL
- Người phát triển ứng dụng
- Người dùng cuối

Câu 9: Hãy cho biết mục đích của việc mô hình hóa các CSDL. Nêu các cách tiếp cận khác nhau trong mô hình hóa CSDL?

Chương 2

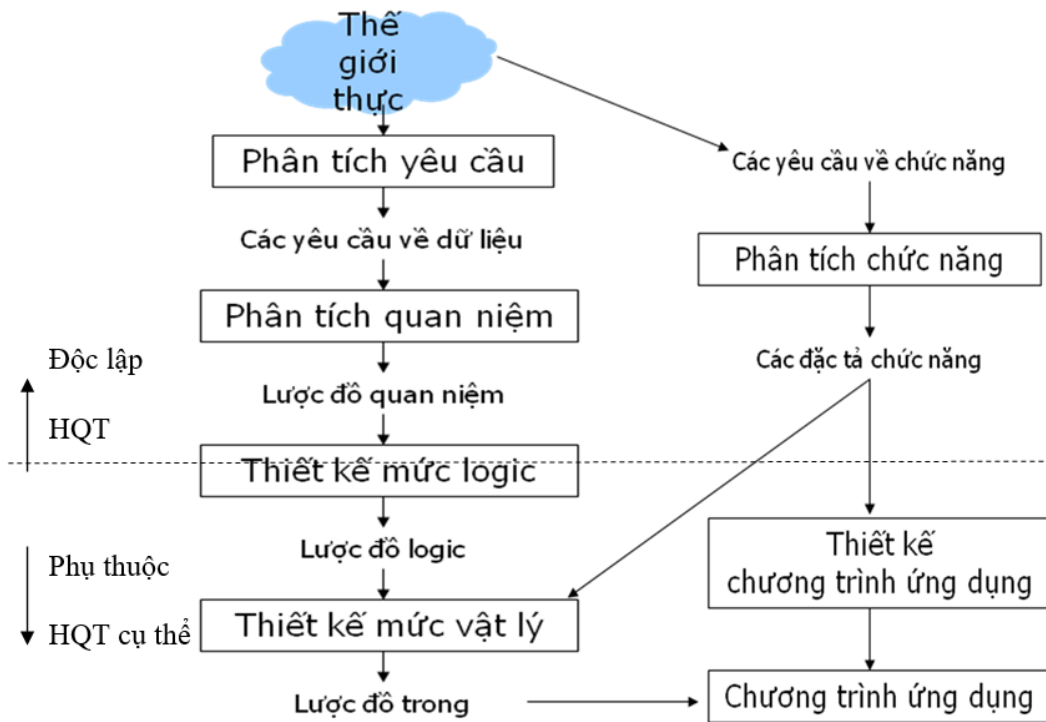
MÔ HÌNH THỰC THỂ KẾT HỢP

♦**Mục tiêu học tập:** Sau khi học xong chương này, người học có thể:

- Nắm được các khái niệm trong mô hình thực thể kết hợp.
- Thiết kế mô hình ở mức quan niệm.

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

2.1 Mô hình dữ liệu quan niệm và quá trình thiết kế CSDL



Hình 2.1: Quá trình thiết kế CSDL

- Mô hình thực thể - kết hợp được dùng để thiết kế CSDL ở mức quan niệm, biểu diễn trừu tượng cấu trúc của CSDL.
- Quá trình thiết kế CSDL được minh họa ở hình 2.1. Bước đầu tiên là tập hợp các yêu cầu và phân tích. Kết quả của bước này là một tập hợp các yêu cầu của người dùng
- Bước thứ hai là thiết kế quan niệm. 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 (kết quả ở bước thứ nhất) sang thành một lược đồ khái niệm. Lược đồ quan niệm là một mô tả cô đọng về yêu cầu dữ liệu của những người dùng bao gồm: mô tả chi tiết các kiểu dữ liệu, các liên kết và 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 quan niệm. Do những khái niệm đó không chứa các chi tiết cài

đặt, chúng thường dễ hiểu và có thể được dùng để đảm bảo rằng kết quả của quá trình thiết kế CSDL sẽ đáp ứng được tất cả đòi hỏi của người dùng 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.

- Bước thứ ba là thiết kế logic (hay còn gọi là ánh xạ mô hình dữ liệu). Ở bước này người thiết kế cài đặt CSDL bằng một HQT CSDL. Hầu hết các HQT CSDL dùng một mô hình dữ liệu thể hiện (chẳng hạn mô hình quan hệ hay mô hình hướng đối tượng, ...), do vậy lược đồ quan niệm được chuyển đổi từ mô hình dữ liệu bậc cao sang mô hình dữ liệu thể hiện.
- Bước cuối cùng là thiết kế 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.

2.2 Các thành phần cơ bản của mô hình thực thể - kết hợp

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

2.2.1 Thực thể - Tập thực thể

- Một *thực thể* (an entity) là một “sự vật” hoặc “đối tượng” mà nó tồn tại và có thể phân biệt được với các đối tượng khác. Ví dụ như một nhân viên trong một tổ chức là một thực thể.
- Một *tập thực thể* (an entity set) là một tập hợp các thực thể cùng loại mà chúng chia sẻ cùng những tính chất hoặc thuộc tính. Ví dụ như tập hợp tất cả những người mà họ là nhân viên của một tổ chức là một tập thực thể *NHANVIEN*.
- Mỗi *tập thực thể* được đặt một tên gọi, thông thường là danh từ. Ví dụ như *NHANVIEN*, *HANGHOA*,...

2.2.2 Thuộc tính (Attribute)

- Thuộc tính là những đặc tính riêng biệt của tập thực thể.

Ví dụ 2.1: Tập thực thể NHÂN_VIEN có các thuộc tính:

- + Họ
- + Tên
- + Ngày sinh

- Thuộc tính là những giá trị nguyên tố:

- + Kiểu chuỗi
- + Kiểu số nguyên
- + Kiểu số thực

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

2.2.3 Mối kết hợp (Relationship)

Mối kết hợp hay còn gọi là mối quan hệ, là sự liên kết giữa hai hay nhiều tập thực thể. Mỗi mối kết hợp cũng được đặt tên thể hiện mối kết hợp.

Ví dụ 2.2: Giữa tập thực thể NHÂN_VIEN và PHÒNG_BAN có các liên kết:

- Một nhân viên thuộc một phòng ban nào đó .
- Một phòng ban có một nhân viên làm trưởng phòng.

2.2.4 Lược đồ E/R

Là đồ thị biểu diễn các tập thực thể, thuộc tính và mối quan hệ.

Các ký hiệu trong lược đồ E/R:

+ Định

+ Tập thực thể

Tên tập thực thể

+ Thuộc tính

Tên thuộc tính

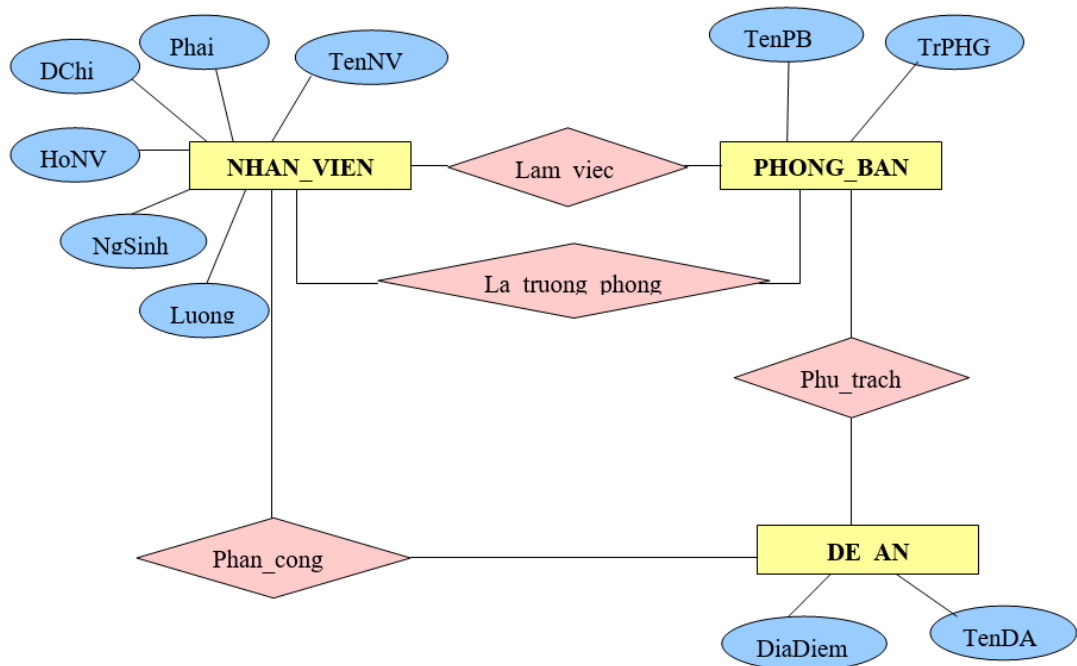


+ Quan hệ

+ Cạnh là đường nối giữa

- Tập thực thể và thuộc tính
- Mỗi quan hệ và tập thực thể

Ví dụ 2.3. Lược đồ E/R



Hình 2.2: Lược đồ E/R

2.2.5 Thể hiện của lược đồ E/R

Một CSDL được mô tả bởi lược đồ E/R sẽ chứa đựng những dữ liệu cụ thể gọi là thể hiện CSDL.

- Mỗi tập thực thể sẽ có tập hợp hữu hạn các thực thể

Giả sử tập thực thể **NHAN_VIEN** có các thực thể như NV_1, NV_2, \dots, NV_n

- Mỗi thực thể sẽ có 1 giá trị cụ thể tại mỗi thuộc tính

NV_1 có TENNV="Tung", NGSINH="08/12/1955", PHAI="Nam"

NV_2 có TENNV="Hang", NGSINH="07/19/1966", PHAI="Nu"

► Chú ý:

- Không lưu trữ lược đồ E/R trong CSDL
 - + Khái niệm trừu tượng
- Lược đồ E/R chỉ giúp ta thiết kế CSDL trước khi chuyển các quan hệ và dữ liệu xuống mức vật lý

2.2.6 Thể hiện của một mối kết hợp

- Thể hiện CSDL còn chứa các mối quan hệ cụ thể
- Xét mối quan hệ *Lam_viec* giữa **NHAN_VIEN** và **PHONG_BAN**, trong đó một thể hiện của quan hệ này sẽ tương ứng với việc một nhân viên làm việc tại phòng ban nào đó.



Hình 2.3: Thể hiện mối kết hợp

NHAN_VIEN	PHONG_BAN
Tùng	Nghiên cứu
Hằng	Điều hành
Vinh	Quản lý

(Tùng, Nghiên cứu)

(Hằng, Điều hành)

(Vinh, Quản lý)

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

2.2.7 Bản số

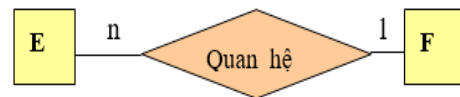
Bản số của một tập thực thể đối với một mối kết hợp là cặp (bản số tối thiểu, bản số tối đa). Trong đó chúng được định nghĩa như sau:

- Bản số tối thiểu: bằng 0 hoặc 1, là số lần tối thiểu mà một thể hiện bất kỳ của một tập thực thể tham gia vào các thể hiện của mối kết hợp.
- Bản số tối đa: bằng 1 hoặc n, là số lần tối đa mà một thể hiện bất kỳ của một tập thực thể tham gia vào các thể hiện của mối kết hợp.
- Xét mỗi quan hệ nhị phân R (binary relationship) giữa 2 tập thực thể E và F , tính multiplicity bao gồm:

+ Một - Nhiều

■ Một E có quan hệ với nhiều F

■ Một F có quan hệ với một E



+ Một - Một

■ Một E có quan hệ với một F

■ Một F có quan hệ với một E



+ Nhiều - Nhiều

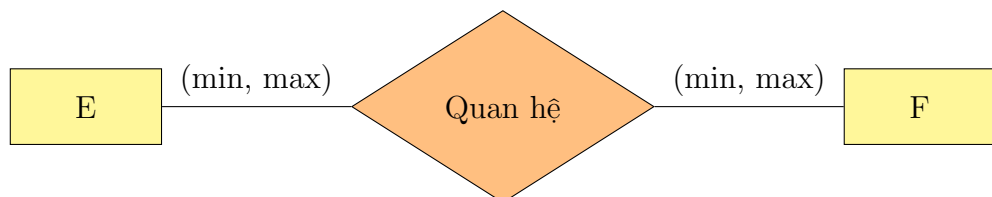
■ Một E có quan hệ với nhiều F

■ Một F có quan hệ với nhiều E



Hình 2.4: Mối quan hệ vai trò

- (\min, \max) chỉ định mỗi thực thể $e \in E$ tham gia ít nhất và nhiều nhất vào thể hiện của R



+ $(0,1)$ – không hoặc 1

+ $(1,1)$ – duy nhất 1

+ (0,n) – không hoặc nhiều

+ (1,n) – một hoặc nhiều

Ví dụ 2.4. Ghi bản số cho quan hệ

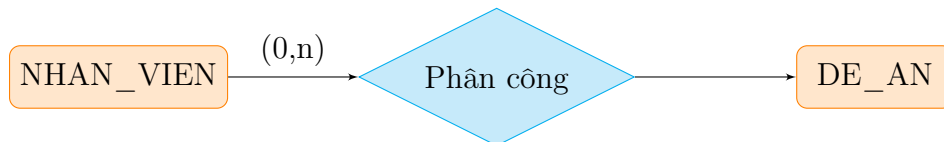
- Một phòng ban có nhiều nhân viên



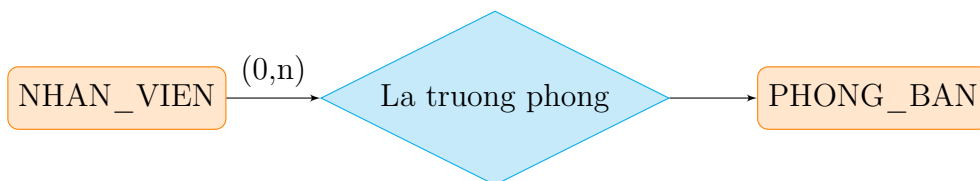
- Một nhân viên chỉ thuộc 1 phòng ban



- Một nhân viên có thể tham gia nhiều đề án hoặc không tham gia đề án nào

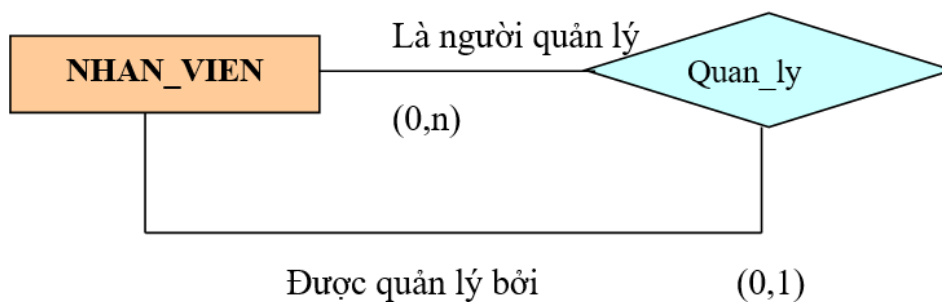


- Một nhân viên là trưởng phòng của 1 phòng ban hoặc không là trưởng phòng



Hình 2.5: Thể hiện bản số (min, max) giữa các thực thể

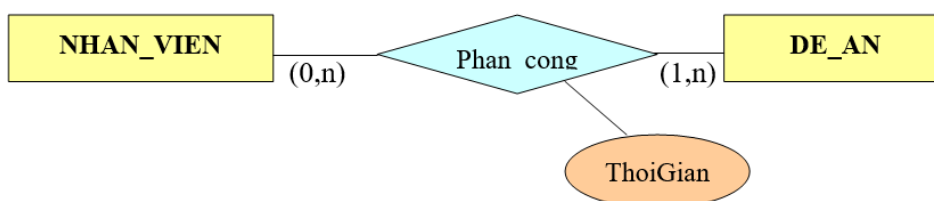
- Một loại thực thể có thể tham gia nhiều lần vào một quan hệ với nhiều vai trò khác nhau.



Hình 2.6: *Mối quan hệ vai trò*

2.2.8 Thuộc tính trên mối quan hệ

- Thuộc tính trên mối quan hệ mô tả tính chất cho mối quan hệ đó.
- Thuộc tính này không thể gắn liền với những thực thể tham gia vào mối quan hệ.



Hình 2.7: Thuộc tính trên mối quan hệ

2.2.9 Khóa

2.2.9.1 Khóa của một tập thể

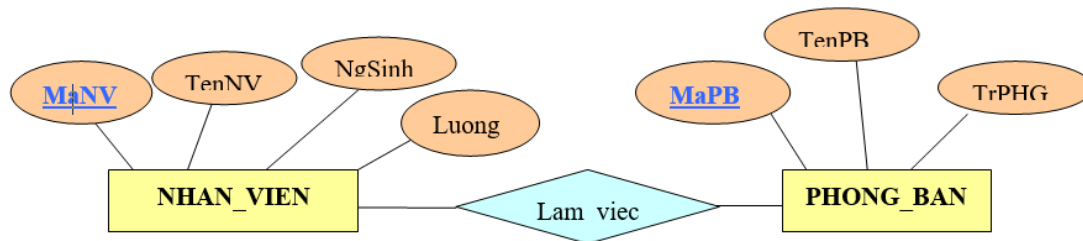
- Các thực thể trong tập thực thể cần phải được phân biệt
- Khóa K của tập thực thể E là một hay nhiều thuộc tính sao cho
 - + Lấy ra 2 thực thể bất kỳ e_1 , và e_2 trong E
 - + Thì e_1 , và e_2 không thể có các giá trị giống nhau tại các thuộc tính trong K

► Chú ý

- + Mỗi tập thực thể phải có 1 khóa
- + Một khóa có thể có 1 hay nhiều thuộc tính

- + Có thể có nhiều khóa trong 1 tập thực thể, ta sẽ chọn ra 1 khóa làm khóa chính cho tập thực thể đó, thuộc tính khóa chính được gạch dưới

Ví dụ:



Hình 2.8: Thuộc tính làm khóa chính

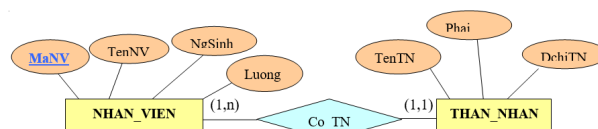
2.2.9.2 Khóa của một mối kết hợp

- Khóa của một mối kết hợp nhận được bằng cách kết hợp khóa của các tập thực thể tham gia vào mối kết hợp đó. Tập hợp tất cả các giá trị của các thuộc tính khóa của một mối kết hợp xác định duy nhất một thể hiện của mối kết hợp đó.
- Trong mô hình, khóa của mối kết hợp ngầm hiểu mà không được ghi ra nếu không quan tâm.

2.2.10 Tập thực thể yếu

- Là tập thực thể mà khóa có được từ những thuộc tính của tập thực thể khác.
- Tập thực thể yếu (weak entity set) phải tham gia vào mối quan hệ mà trong đó có một tập thực thể chính.

Ví dụ:



Hình 2.9: Tập thực thể yếu

Trong ví dụ này tập thực thể yếu là tập thực thể **THAN_NHAN**.

2.3 Thiết kế

2.3.1 Các bước thiết kế

- (1) Xác định tập thực thể
- (2) Xác định mối quan hệ
- (3) Xác định thuộc tính và gán thuộc tính cho tập thực thể và mối quan hệ
- (4) Quyết định miền giá trị cho thuộc tính
- (5) Quyết định thuộc tính khóa
- (6) Quyết định (min, max) cho mối quan hệ

2.3.2 Nguyên lý thiết kế

- Chính xác
- Tránh trùng lặp
- Dễ hiểu
- Chọn đúng mối quan hệ
- Chọn đúng kiểu thuộc tính

2.3.3 Ví dụ minh họa

Giả sử sau đây là một số yêu cầu dữ liệu đối với một công ty chuyên thực hiện các đề án, cần quản lý thông tin liên quan đến đề án như sau:

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

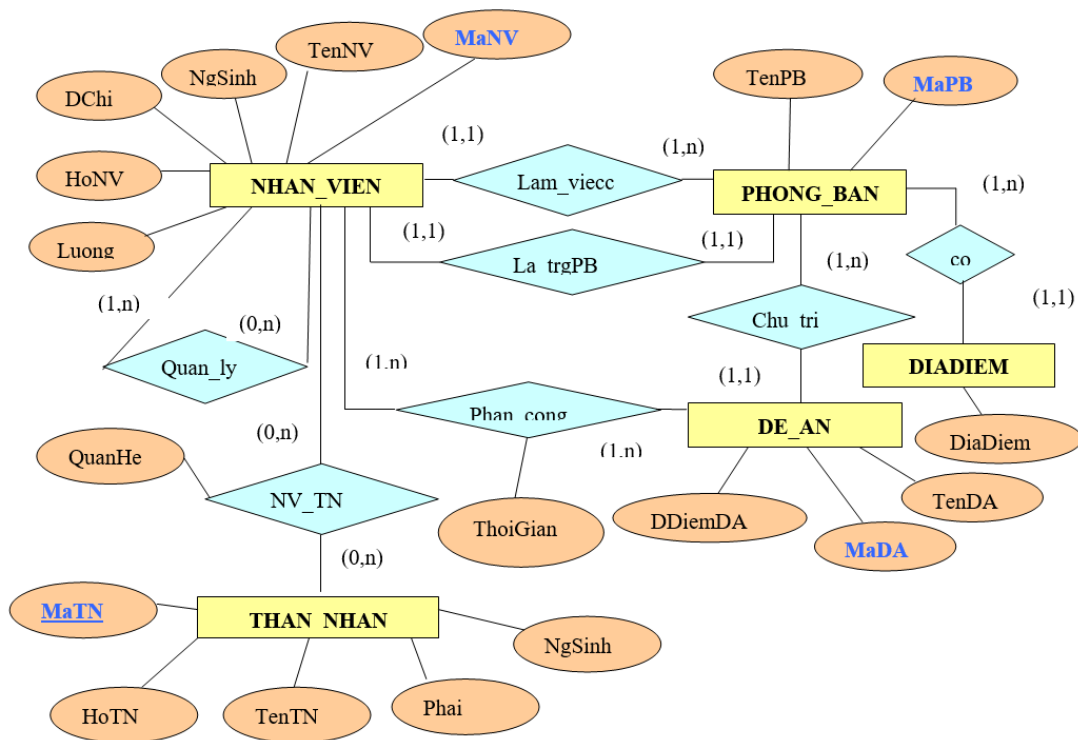
Do công ty thực hiện đề án, nên mỗi phòng có thể có nhiều văn phòng giao dịch hay làm việc khác nhau ở tại những địa điểm khác nhau.

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

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

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

Từ mô tả tình huống trên, mô hình thực thể kết hợp được thiết kế như sau:



Hình 2.10: Thiết kế mô hình E/R

► **Câu hỏi củng cố chương 2:**

Câu 1: Hãy cho biết vai trò của mô hình dữ liệu quan niệm trong quá trình thiết kế CSDL?

Câu 2: Hãy mô tả những khái niệm cơ bản của mô hình thực thể kết hợp?

Câu 3: Cho thêm ví dụ về kiểu thực thể yếu?

Câu 4: Thiết kế lược đồ E/R cho các bài tập sau:

Bài tập 1: Một công ty có một số nhân viên. Thuộc tính của nhân viên gồm có mã nhân viên (định danh), tên nhân viên, địa chỉ, ngày sinh. Hiện công ty thực hiện một vài dự án. Thông tin về dự án gồm có mã dự án (định danh), tên dự án và ngày bắt đầu. Một nhân viên có thể không tham gia hoặc tham gia một hay nhiều dự án. Một dự án phải có ít nhất một nhân viên. Tiền lương tham gia dự án của một nhân viên khác nhau theo từng dự án. Công ty cần lưu số tiền này cho mỗi nhân viên mỗi khi phân công nhân viên vào một dự án.

Bài tập 2: Một phòng thí nghiệm có một số nhà hóa học, mỗi nhà hóa học tham gia vào một hoặc nhiều dự án. Các nhà hóa học sử dụng một số thiết bị cho các dự án. Thông tin về nhà hóa học cần lưu trữ là mã số (định danh), tên và số điện thoại. Thông tin về dự án gồm mã số (định danh) và ngày bắt đầu. Thông tin về thiết bị gồm số thứ tự (serial_No) và giá thành. Hệ thống cần lưu thông tin về ngày giao thiết bị: thiết bị nào được giao cho nhà hóa học nào và để sử dụng vào dự án nào. Một nhà hóa học phải tham gia ít nhất một dự án và có ít nhất một thiết bị. Một thiết bị có thể chưa được sử dụng đến. Một dự án có thể chưa có nhà hóa học nào tham gia và do đó cũng chưa sử dụng một thiết bị nào.

Chương 3

MÔ HÌNH DỮ LIỆU QUAN HỆ

◆ **Mục tiêu học tập:** Sau khi học xong chương này, người học có thể:

- Nắm được các khái niệm cơ bản trong mô hình dữ liệu
- Xác định được khóa chính, khóa ngoại trong mô hình
- Chuyển đổi từ mô hình quan niệm sang mô hình dữ liệu quan hệ
- Chuyển đổi từ mô hình sang lược đồ quan hệ

3.1 Các khái niệm cơ bản

3.1.1 Thuộc tính

- 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 3.1:

Quan hệ Sinh viên có các thuộc tính Mã sinh viên, tên sinh viên, năm sinh, giới tính. . .

Quan hệ Môn học có các thuộc tính Mã môn học, tên môn học, số đơn vị học trình

- Thuộc tính được đợc trung bởi 3 yếu tố:

- + **Tên thuộc tính:** Dài không quá 255 kí tự, không có các kí tự đặc biệt, không được bắt đầu bằng kí tự số. Tên thường được đặt bằng chữ cái in hoa đầu tiên trong bảng chữ cái: A, B, C, D,...

Những chữ cái X, Y, W, Z, . . . dùng cho một nhóm gồm nhiều thuộc tính. Đôi

khi còn dùng các kí hiệu chữ cái với chỉ số A_1, A_2, \dots, A_n để chỉ các thuộc tính trong trường hợp tổng quát.

Trong thực tế, nhà phân tích hệ thống thường đặt tên gọi nhớ cho thuộc tính với các chữ in hoa đầu từ hoặc viết cách nhau bởi dấu gạch chân(_).

+ **Kiểu dữ liệu:** Mỗi thuộc tính đều phải có một kiểu dữ liệu nhất định. Kiểu dữ liệu có thể là vô hướng (đó là các kiểu dữ liệu cơ bản có sẵn như kiểu chuỗi – string, text hoặc character, kiểu số – int, number, kiểu luận lý – logical,...) hoặc các dữ liệu có cấu trúc được định nghĩa trên các kiểu dữ liệu đã có sẵn.//
Một số kiểu dữ liệu thường được dùng trong các HQT CSDL:

- text, string, char : kiểu kí tự
- number, float, double, int : kiểu số
- logical, boolean : kiểu luận lý
- date/time : kiểu ngày tháng năm + giờ phút giây
- varchar : kiểu chuỗi có độ dài thay đổi

Mỗi hệ quản trị CSDL có thể gọi tên các kiểu dữ liệu nói trên bằng các tên gọi khác nhau.

+ **Miền giá trị (domain):** Mỗi thuộc tính có thể chỉ lấy những giá trị trong một tập hợp con của kiểu dữ liệu. Tập hợp các giá trị mà một thuộc tính có thể nhận gọi là miền giá trị, kí hiệu Dom(thuộc tính).

Ví dụ: miền giá trị của thuộc tính A kí hiệu là Dom(A).

3.1.2 Quan hệ

Một quan hệ R có n ngôi được định nghĩa trên tập các thuộc tính $U = \{A_1, A_2, \dots, A_n\}$ (thứ tự các thuộc tính trong R không quan trọng) và kèm theo nó là một tân từ, tức là một qui tắc để xác định mối quan hệ giữa các thuộc tính A_i và được kí hiệu là $R(A_1, A_2, \dots, A_n)$.

Tập các thuộc tính của R được kí hiệu là R^+

- Số thuộc tính của R còn được gọi là số ngôi của quan hệ R.
- Quan hệ còn được gọi bằng thuật ngữ khác là bảng (table).

Ví dụ 3.2: Xét các quan hệ

SinhVien(MaSV, HoTen, GioiTinh, DiaChi, NgaySinh) là một quan hệ có 5 ngôi.

Tân từ: “Mỗi sinh viên có một mã số duy nhất để phân biệt với các sinh viên khác, ngoài ra còn có họ tên, địa chỉ, ngày sinh và giới tính để biết sinh viên đó là nam hay nữ”.

LopHoc(MaLop, TenLop, NienKhoa, SiSo, MaKhoa) là một quan hệ có 5 ngôi.

Tân từ: “Mỗi khoa có một tên gọi và một mã số duy nhất để phân biệt với các khoa khác của trường”.

MonHoc(MaMon, TenMon, DVHT) là một quan hệ có 3 ngôi.

Tân từ: “Mỗi môn học có một tên gọi cụ thể, có một số đơn vị học trình (DVHT) nhất định. Ứng với mỗi môn học có một mã số duy nhất để phân biệt với các môn học khác”.

3.1.3 Bộ giá trị

- Một bộ (tuple) chứa thông tin của một đối tượng cụ thể thuộc quan hệ. Bộ là một dòng dữ liệu trong một quan hệ. Bộ thường được gọi là mẫu tin, bản ghi hay record.
- Về mặt hình thức, một bộ giá trị q là một vector gồm n thành phần mà mỗi thành phần là kết quả của phép tích Đề - các miền giá trị của các thuộc tính và thoả mãn từ đã cho của quan hệ.

$$q = (a_1, a_2, \dots, a_n) \rightarrow (\text{Dom}(A_1) \times \text{Dom}(A_2) \times \dots \times \text{Dom}(A_n))$$

Ví dụ 3.3: Sau đây là các bộ giá trị dựa trên các thuộc tính của quan hệ SinhVien.

$q1 = (001, \text{Trần Quang Minh}, \text{Nam}, \text{Đồng Tháp}, 24/12/1985)$

$q2 = (002, \text{Lê Hoàng Ngọc Mai}, \text{Nữ}, \text{Tiểu Cần – Trà Vinh}, 12/9/1981)$

$q3 = (003, \text{Trương Hoài Anh}, \text{Nam}, \text{Vĩnh Long}, 8/5/1979)$

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

Đây còn gọi là *phép chiếu một bộ lên thuộc tính A_i* . **Ví dụ 3.4:**

$q1.HoTen = \text{'Trần Quang Minh'}$

$q2.DiaChi = \text{'Đồng Tháp'}$

Lưu ý: Trong Microsoft Access, chuỗi ký tự được viết trong cặp dấu nháy kép (“”), còn trong SQL Server và Oracle, chuỗi ký tự được viết trong cặp dấu nháy đơn (‘’).

3.1.4 Thể hiện của quan hệ

Thể hiện còn được gọi là tình trạng của quan hệ R , ký hiệu TR , là tập hợp các bộ giá trị của quan hệ R vào một thời điểm. Tại những thời điểm khác nhau thì quan hệ sẽ có những thể hiện khác nhau.

Ví dụ 3.5: Về thể hiện của quan hệ MonHoc

MaMon	TenMon	DVHT
CSDL	Cơ sở dữ liệu	4
CTDL_GT	Cấu trúc dữ liệu và giải thuật	5
LTCB	Lập trình căn bản	5

Bảng 3.1: Thể hiện của quan hệ MonHoc

3.1.5 Khóa

Có nhiều cách khác nhau để định nghĩa khóa.

3.1.5.1 Khóa (key)

Khóa của lược đồ quan hệ R định nghĩa trên tập thuộc tính $U = \{A_1, A_2, \dots, A_n\}$ là một tập con $K \subseteq U$ thỏa:

- Không tồn tại hai bộ có giá trị trùng nhau trên mọi thuộc tính của K . Tức là mỗi giá trị của khoá K phải xác định duy nhất trên quan hệ R
- Một lược đồ quan hệ có rất nhiều khoá. Việc xác định tất cả các khoá của một lược đồ quan hệ là rất khó.
- K được gọi là khoá của quan hệ R nếu thỏa hai điều kiện sau đây:

(i) K xác định được giá trị của $A_i \forall 1, 2, \dots, n$

(ii) Không tồn tại $K' \subseteq K$, mà K' xác định được giá trị của $A_i \subseteq 1, 2, \dots, n$

Nghĩa là K là tập con nhỏ nhất mà giá trị của nó có thể xác định được duy nhất một bộ giá trị của quan hệ. Khoá thỏa 2 điều kiện trên được gọi là khoá dự tuyển (candidate key) và là khoá nội của quan hệ.

3.1.5.2 Siêu khóa (Super key):

- Siêu khóa: là một tập con các thuộc tính của $+$ mà nhờ vào đó chúng ta có thể phân biệt 2 bộ khác nhau trong cùng một thể hiện T_Q bất kỳ.
- $\forall t_1, t_2 \in T_Q, t_1[K] \neq t_2[K] \iff K$ là siêu khóa của Q .
- Một quan hệ có ít nhất một siêu khóa (Q_+) và có thể có nhiều siêu khóa.

Ví dụ 3.6: Quan hệ LopHoc(MaLop, TenLop, NienKhoa, SiSo)

Lược đồ LopHoc có khoá là MaLop và một số siêu khoá sau:

$$K^1 = \{\text{MaLop}, \text{TenLop}\}$$

$$K^2 = \{\text{MaLop}, \text{TrnLop}, \text{SiSo}\}$$

$$K^3 = \{\text{MaLop}, \text{SiSo}\}$$

$$K^4 = \{\text{MaLop}, \text{NienKhoa}\}$$

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

3.1.5.3 Khóa chính (Primary key)

- Trên một lược đồ quan hệ luôn luôn tồn tại một thuộc tính hoặc một tập thuộc tính có khả năng xác định được các thuộc tính còn lại. Thông thường trên một lược đồ quan hệ có thể tồn tại nhiều khóa và tùy theo ngữ cảnh mà người ta sẽ chọn một khóa để sử dụng, gọi là *khóa chính*.
- Khóa chính phải là duy nhất và không chứa giá trị rỗng.
- Không được phép sửa đổi giá trị của thuộc tính khoá. Nếu muốn sửa đổi giá trị của thuộc tính khoá của một bộ q , người sử dụng phải huỷ bỏ bộ q và sau đó thêm mới giá trị q với giá trị khoá mới

Ví dụ 3.7: Xét các quan hệ:

Khoa(MaKhoa, TenKhoa)

MonHoc(MaMon, TenMon, DVHT)

3.1.5.4 Khóa ngoại (Foreign key)

Giả sử có hai quan hệ R và S. Một tập thuộc tính K của quan hệ R được gọi là khóa ngoại của quan hệ R nếu K là khóa chính của quan hệ S.

Ví dụ 3.8: MaKhoa trong quan hệ LopHoc là khóa ngoại vì nó là khóa chính của quan hệ Khoa

3.1.6 Lược đồ quan hệ (Relation Schema)

- **Cấu trúc của một quan hệ** là tập thuộc tính hình thành nên quan hệ đó.
- **Lược đồ quan hệ** nhằm mục đích **mô tả cấu trúc** của một quan hệ và **các mối liên hệ** giữa các thuộc tính trong quan hệ đó.
- Một lược đồ quan hệ gồm một tập thuộc tính của quan hệ kèm theo một mô tả để xác định ý nghĩa và mối liên hệ giữa các thuộc tính.
- Lược đồ quan hệ được đặc trưng bởi:
 - + Một tên phân biệt
 - + Một tập hợp hữu hạn các thuộc tính (A_1, \dots, A_n)
- Ký hiệu của lược đồ quan hệ **Q** gồm **n** thuộc tính (A_1, A_2, \dots, A_n) là: $Q(A_1, A_2, \dots, A_n)$

Ví dụ 3.10:

► Lược đồ quan hệ PhongBan

+ **PhongBan**(MaPHG, TenPHG, TrPHG, NG_NhanChuc)

+ Mô tả: Mỗi phòng ban được cấp một mã số duy nhất để phân biệt với các phòng ban khác trong công ty, và có một tên phòng ban, một trưởng phòng cùng ngày nhận chức của trưởng phòng.

► Lược đồ quan hệ NhanVien

+ **NhanVien**(MANV, HoTen, Phai, Luong, PHG)

+ Mô tả: Mỗi nhân viên được cấp mã số duy nhất để phân biệt với các nhân viên khác trong công ty, và có một họ tên nhân viên, một giới tính, một mức lương, và thuộc về một phòng ban nhất định.

3.2 Các bước biến đổi cấu trúc từ thực thể kết hợp sang quan hệ dữ liệu

- **Bước 1:** Chuyển các loại thực thể thành quan hệ.

Mỗi loại thực thể được chuyển đổi thành một quan hệ.

Với mô hình quan niệm ở mục 2.3.3 có các tập thực thể như NHAN_VIEN, PHONG_BAN, DE_AN, DIADIEM, THAN_NHAN được chuyển thành 5 quan hệ tương đương:

NHAN_VIEN(MaNV, HoNV, TenNV,...)

PHONG_BAN(MaPB, TenPB, ...)

DE_AN(MaDA, TenDA,...)

DIADIEM(DiaDiem,..)

THAN_NHAN(MaTN, HoTN, ...)

- **Bước 2:** Chuyển các mối kết hợp thành quan hệ

a. **Mối quan hệ 1 - 1:** Không cần chuyển đổi.

b. **Mối quan hệ 1 – n:**

Thêm các thuộc tính khóa chính ở vế n vào vế 1.

Ví dụ: NHAN_VIEN và PHONG_BAN có mối quan hệ 1-n nghĩa là mỗi nhân viên chỉ làm việc duy nhất ở 1 phòng ban và mỗi phòng ban có thể có nhiều nhân viên. Như vậy cần thêm MaPB vào quan hệ NHAN_VIEN(MaNV, TenNV, DChi, NgSinh, Luong, MaPB).

c. **Mối quan hệ n - n:**

Chuyển thành một quan hệ mới và lấy các thuộc tính khóa chính của các loại thực thể tham gia vào mối kết hợp để làm khóa chính.

Ví dụ: mối kết hợp Phan_cong giữa hai thực thể NHAN_VIEN và DE_AN được chuyển thành quan hệ PHAN_CONG(MaNV, MaDA, ThoiGian).

d. **Mối kết hợp đệ quy:**

Chuyển thành một quan hệ. Thuộc tính khóa chính của loại thực thể này được lấy 2 lần, nhưng phải được đổi tên lại cho khác nhau, được dùng làm thuộc tính khóa chính.

e. **Loại thực thể tổng quát hóa:**

Bổ sung vào quan hệ tất cả các thuộc tính của các loại thực thể chuyên biệt hóa mà chưa có trong thực thể tổng quát hóa, đồng thời bổ sung thêm một thuộc tính mới để phân biệt hai loại thực thể chuyên biệt hóa.

Ví dụ: chuyển đổi mô hình thực thể kết hợp mục 2.3.3 sang dữ liệu quan hệ:

NHANVIEN (MaNV, HoNV, TenNV, Luong, MaPB, NGSinh, DCHI, Ma_NQL)

PHONGBAN (MaPHG, TenPHG, TrPHG)

DEAN (MaDA, TenDA, DDIEM_DA, MaPB)

PHANCONG (MaNV, MaDA, ThoiGian)

DIADIEM_PHG (MaPB, DIADIEM)

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

NV_TN (MaNV, MaTN, QuanHe)

3.3 Các thao tác trên quan hệ

3.3.1 Phép thêm một bộ mới vào quan hệ

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

Dạng hình thức của phép thêm bộ mới là:

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

Trong đó:

A_i là các thuộc tính với $i = 1 \rightarrow n$

v_i là các giá trị của thuộc tính với $i = 1 \rightarrow n$

Lưu ý: Các thuộc tính không có tên trong danh sách gán giá trị của bộ t trong câu lệnh insert sẽ có giá trị NULL, tức là giá trị không xác định.

Ví dụ 3.11 Xét quan hệ SinhVien như sau:

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

Thêm bộ $q_4 = (004, \text{Trương Văn Chung, Nam, Cần Thơ, 25/5/1976})$ vào quan hệ SinhVien bởi phép thêm như sau:

INSERT(SinhVien; [MaSV] = 004, [HoTen] = Trương Văn Chung, [GioiTinh] = Nam, [DiaChi] = Cần Thơ, [NgàySinh] = 25/5/1976).

Thể hiện T của quan hệ SinhVien sẽ là:

$q_1 = (001, \text{Trần Quang Minh, Nam, Đồng Tháp, 24/12/1985})$

$q_2 = (002, \text{Lê Hoàng Ngọc Mai, Nữ, Tiểu Cần – Trà Vinh, 12/9/1981})$

$q_3 = (003, \text{Trương Hoài Anh, Nam, Vĩnh Long, 8/5/1979})$

$q_4 = (004, \text{Trương Văn Chung, Nam, Cần Thơ, 25/5/1976})$

Nếu xem các thuộc tính Ai là cố định và giá trị vi là hoàn toàn tương ứng thì phép chèn có thể viết dưới dạng tường minh như sau:

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

Phép chèn có thể không thực hiện được hoặc làm mất tính nhất quán của dữ liệu trong CSDL vì các lý do:

- Giá trị khoá của bộ mới là rỗng (Null) hoặc trùng với giá trị khoá của một bộ đã có trong CSDL. Trong trường hợp này hệ quản trị CSDL không cho bổ sung.
- Bộ mới không phù hợp với cấu trúc của lược đồ quan hệ. Trường hợp này có thể xảy ra khi người sử dụng nhầm lẫn thứ tự, kiểu hoặc độ lớn của các thuộc tính. Hệ quản trị CSDL có thể không cho bổ sung hoặc vẫn cho bổ sung bộ mới nhưng tính nhất quán của dữ liệu mới không được đảm bảo.
- Một số giá trị của bộ mới không thuộc miền giá trị của thuộc tính tương ứng. Trong trường hợp này, nếu quan hệ đã được đảm bảo tính nhất quán bởi các RBTV miền giá trị thì hệ quản trị CSDL sẽ không cho bổ sung. Nếu không có ràng buộc toàn vẹn thì tính nhất quán của dữ liệu sẽ bị vi phạm nhưng hệ quản trị CSDL không phát hiện được.

3.3.2 Phép loại bỏ bộ ra khỏi quan hệ

Phép loại bỏ một bộ t của quan hệ sẽ lấy đi bộ t khỏi thể hiện của quan hệ $T_R = T_R - \{t\}$, có thể viết lại như sau:

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

Trong đó: $A_i = v_i$ với $i = 1 \rightarrow n$ được gọi như những điều kiện thoả một số thuộc tính của bộ t để loại bỏ một bộ ra khỏi quan hệ.

Ví dụ 3.12: Xét quan hệ

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

Với phép loại bỏ như sau:

DELETE(SinhVien; [DiaChi] = Đồng Tháp)

bộ giá trị:

$q_1 = (001, \text{Trần Quang Minh, Nam, Đồng Tháp, 24/12/1985})$

Sẽ bị loại ra khỏi quan hệ SinhVien vì có thuộc tính thoả điều kiện, thể hiện T của quan hệ SinhVien lúc này sẽ là:

$q_2 = (002, \text{Lê Hoàng Ngọc Mai, Nữ, Tiểu Cần – Trà Vinh, 12/9/1981})$

$q_3 = (003, \text{Trương Hoài Anh, Nam, Vĩnh Long, 8/5/1979})$

$q_4 = (004, \text{Trương Văn Chung, Nam, Cần Thơ, 25/5/1976})$

3.3.3 Phép 3.3.3. Phép sửa đổi giá trị của các thuộc tính của quan hệ

Dữ liệu của CSDL cần được cập nhật theo thời gian nhằm đảm bảo tính nhất quán và chính xác của dữ liệu.

Ngôn ngữ hình thức để cập nhật giá trị của bộ trong quan hệ là:

UPDATE($R; A_1 = C_1, A_2 = C_2, \dots, A_n = C_n; A_1 = v_1, A_2 = v_2, \dots, A_n = v_n$)

Trong đó: R là quan hệ cần thực hiện việc sửa đổi: $A_i = c_i$ với $i = 1 \rightarrow n$ là điều kiện tìm kiếm bộ giá trị để sửa đổi và $A_i = v_i$ với $i = 1 \rightarrow n$ là giá trị mới của bộ. **Ví dụ 3.13:** Xét quan hệ

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

Với phép sửa đổi giá trị như sau:

UPDATE(SinhVien; [MaSV] = 002, [DiaChi] = Cần Long – Trà Vinh)

Thì giá trị bộ q_2 được sửa lại thành:

$q_2 = (002, \text{Lê Hoàng Ngọc Mai, Nữ, Cần Long – Trà Vinh, 12/9/1981})$

► **Câu hỏi củng cố chương 3:**

Câu 1: Hãy phân biệt các thuật ngữ sau đây:

- Thuộc tính, miền giá trị
- Quan hệ và lược đồ quan hệ
- Lược đồ quan hệ và lược đồ CSDL
- Siêu khóa và khóa

Câu 2: Vì sao không cho phép sự xuất hiện của hai bộ giống hệt nhau trong một quan hệ?

Câu 3: Hãy trình bày ý nghĩa của khóa ngoại, cho ví dụ minh họa?

Câu 4: Với các bài tập trong chương 2 hãy chuyển sang mô hình dữ liệu quan hệ và lược đồ quan hệ?

Câu 5: Hãy xác định khóa chính, khóa ngoại trong các lược đồ quan hệ của câu 4?

Chương 4

NGÔN NGỮ ĐẠI SỐ QUAN HỆ

♦ **Mục tiêu học tập:** Sau khi học xong chương này, người học có thể:

- Vận dụng các phép toán quan hệ, biểu thức Đại số quan hệ để viết câu truy vấn.
- Vận dụng các nguyên tắc tối ưu hóa biểu thức để viết câu truy vấn tối ưu.

4.1 Các khái niệm

4.1.1 Giới thiệu

- Xét một số xử lý trên quan hệ NHANVIEN (Ví dụ của KQHT1) như sau:
 - + Thêm mới một nhân viên
 - + Chuyển nhân viên có tên là “Tùng” sang phòng số 1
 - + Cho biết họ tên và ngày sinh các nhân viên có lương trên 20.000
- Chia thành 2 loại xử lý
 - + Làm thay đổi dữ liệu (cập nhật)
 - * Thêm mới, xóa và sửa
 - + Không làm thay đổi dữ liệu (rút trích)
 - * Truy vấn (query)
- Để thực hiện các xử lý ta sử dụng các cách sau:

- + Đại số quan hệ (Relational Algebra)
 - Biểu diễn câu truy vấn dưới dạng biểu thức
- + Phép tính quan hệ (Relational Calculus)
 - Biểu diễn kết quả
- + SQL (Structured Query Language)

Ở KQHT này, chúng ta sẽ sử dụng biểu thức đại số quan hệ và phép toán quan hệ để xử lý các vấn đề trên.

4.1.2 Khái niệm đại số quan hệ

- Đại số quan hệ là một mô hình toán học dựa trên lý thuyết tập hợp mà đối tượng xử lý là các quan hệ trong cơ sở dữ liệu quan hệ.
- Chức năng của Đại số quan hệ:
 - Cho phép mô tả các phép toán rút trích dữ liệu từ các quan hệ trong cơ sở dữ liệu quan hệ.
 - Cho phép tối ưu.
- Đại số quan hệ có nền tảng toán học (cụ thể là lý thuyết tập hợp) để mô hình hóa CSDL quan hệ. Ngoài ra, có thể áp dụng các phép toán có sẵn của lý thuyết tập hợp để tối ưu quá trình rút trích dữ liệu từ CSDL quan hệ.

a. Nhắc lại kiến thức về biểu thức toán học

- **Đại số**
 - Toán tử (operator)
 - Toán hạng (operand)
- **Trong số học**
 - Toán tử: $+$, $-$, $*$, $/$
 - Toán hạng - biến (variables): x , y , z
 - Hằng (constant)
 - Biểu thức $(x + 7)/(y - 3)$
 $(x + y) * z$ and/or $(x + 7)/(y - 3)$

b. Trong biểu thức đại số quan hệ

- Biến là các quan hệ
 - + Tập hợp (set)
- Toán tử là các phép toán (operations)
 - + Trên tập hợp
 - Hợp \cup (union)
 - Giao \cap (intersection)
 - Trừ $-$ (difference)
 - + Rút trích 1 phần của quan hệ
 - Chọn σ (selection)
 - Chiếu π (projection)
 - + Kết hợp các quan hệ
 - Tích Cartesian \times (Cartesian product)
 - Kết (join) \bowtie
 - + Phép đổi tên ρ
 - + Phép chia \div
 - + Phép gán \leftarrow
- Hằng số là thể hiện của quan hệ
- Biểu thức
 - + Được gọi là câu truy vấn
 - + Là chuỗi các phép toán đại số quan hệ
 - + Kết quả trả về là một thể hiện của quan hệ

Ví dụ 4.1: Thể hiện của quan hệ

Quan hệ **NHANVIEN** gồm:

NHANVIEN(MaNV, HoNV, TenLot, TenNV, NgSinh, Phai, MaNQL, Phong, MLuong)

NHANVIEN								
MANV	HONV	TENLOT	TENNV	NGSINH	PHAI	MANQL	PHONG	MLUONG
004	Đinh	Bá	Tiến	09/01/1968	Nam	002	NC	2.200.000
005	Bùi	Thuỳ	Vũ	19/07/1972	Nam	003	ĐH	2.200.000
006	Nguyễn	Mạnh	Hùng	15/09/1973	Nam	002	NC	2.000.000
007	Trần	Thanh	Tâm	31/07/1975	Nữ	002	NC	2.200.000
008	Trần	Hồng	Vân	04/07/1976	Nữ	004	NC	1.800.000

Bảng 4.1: Thể hiện của quan hệ NHANVIEN

4.2 Sử dụng các phép toán quan hệ, đại số quan hệ viết câu truy vấn

4.2.1 Phép toán tập hợp

Các phép toán cơ bản được áp dụng trên tập các bộ giá trị của các quan hệ, được hình thành từ lý thuyết tập hợp toán học: hợp hay hội (union), trừ (difference), giao (intersection).

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

► *Lưu ý: Để thực hiện được các phép toán trên tập hợp giữa quan hệ R và quan hệ S thì bắt buộc R và S phải khả hợp.*

Vậy khả hợp là gì?

Tính khả hợp

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

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

Kết quả của \cup , \cap , và $-$ là một quan hệ có cùng tên thuộc tính với quan hệ đầu tiên (R)

Ví dụ 4.2: Cho 2 quan hệ THANNHAN và NHANVIEN khả hợp vì:

THANNHAN	TENTN	NG_SINH	PHAITN
	Trinh	04/05/1986	Nu
	Khang	10/25/1983	Nam
	Phuong	05/03/1958	Nu
	Minh	02/28/1942	Nam
	Chau	12/30/1988	Nu

NHANVIEN	TENNV	NGSINH	PHAI
	Tung	12/08/1955	Nam
	Hang	07/19/1968	Nữ
	Nhu	06/20/1951	Nữ
	Hung	09/15/1962	Nam

Bảng 4.2: Ví dụ về tính khả hợp của quan hệ R và quan hệ S

Có cùng số thuộc tính là 3

$\text{DOM}(\text{TENNV}) = \text{DOM}(\text{TENTN})$

$\text{DOM}(\text{NGSINH}) = \text{DOM}(\text{NG_SINH})$

$\text{DOM}(\text{PHAI}) = \text{DOM}(\text{PHAITN})$

(DOM viết tắt của chữ DOMAIN là miền giá trị)

4.2.1.1 Phép Hợp (Union)

- Cho 2 quan hệ R và S khả hợp
- Phép hợp của R và S
 - + Ký hiệu $R \cup S$
 - + Kết quả trả về là một quan hệ gồm các bộ thuộc R hoặc thuộc S, hoặc cả hai (các bộ trùng lặp sẽ bị bỏ)

$$R \cup S = \{t/t \in R \vee t \in S\}$$

Ví dụ 4.3: Phép hợp giữa 2 quan hệ R và S

R	A	B
	α	1
	α	2
	β	1

S	A	B
	α	2
	β	3

$R \cup S$	A	B
	α	1
	α	2
	β	1
	β	3

Bảng 4.3: Ví dụ thực hiện phép hợp giữa quan hệ R và quan hệ S

Nói cách khác, hợp của hai quan hệ R và S là một quan hệ có cùng ngôi với quan hệ R và S, với các bộ giá trị bằng gộp các bộ của cả R và S, những bộ giá trị trùng nhau chỉ

giữ lại 1 bộ. Trong ví dụ trên bộ α , 2 xuất hiện trong cả R và S, do đó chỉ xuất hiện 1 lần trong $R \cup S$

4.2.1.2 Phép Giao (Intersection)

- Cho 2 quan hệ R và S khả hợp
- Phép giao của R và S
 - + Ký hiệu $R \cap S$
 - + Kết quả trả về là một quan hệ gồm các bộ thuộc R đồng thời thuộc S

$$R \cap S = \{t/t \in R \wedge t \in S\}$$

Ví dụ 4.4: Phép giao giữa 2 quan hệ R và S

R	A	B
	α	1
	α	2
	β	1

S	A	B
	α	2
	β	3

$R \cap S$	A	B
	α	2

Bảng 4.4: Ví dụ thực hiện phép giao giữa quan hệ R và quan hệ S

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

4.2.1.3 Phép Trừ (Difference)

- Cho 2 quan hệ R và S khả hợp
- Phép trừ của R và S
 - + Ký hiệu $R - S$
 - + Kết quả trả về là một quan hệ gồm các bộ thuộc R và không thuộc S

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

Ví dụ 4.5: Phép trừ giữa 2 quan hệ R và S

R	A	B
	α	1
	α	2
	β	1

S	A	B
	α	2
	β	3

$R - S$	A	B
	α	1
	β	1

Bảng 4.5: Ví dụ thực hiện phép trừ giữa quan hệ R và quan hệ S

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

Chú ý: Tính chất của các phép toán

- Phép hợp, phép giao có tính chất giao hoán và kết hợp

- Giao hoán

$$R \cup S = S \cup R$$
$$R \cap S = S \cap R$$

- Kết hợp

$$R \cup (S \cap T) = (R \cup S) \cap T$$
$$R \cap (S \cup T) = (R \cap S) \cup T$$

- Phép trừ không có tính chất giao hoán và kết hợp

$$R - S \neq S - R$$

4.2.1.4 Phép chia

- Được dùng để lấy ra một số bộ trong quan hệ R sao cho thỏa với tất cả các bộ trong quan hệ S

- Ký hiệu: $R \div S$

Ta có: $R(Z)$ và $S(X)$

Trong đó, Z là tập thuộc tính của quan hệ R, X là tập thuộc tính của hệ quan S

Và $X \subset Z$

- Kết quả của phép chia là một quan hệ $T(Y)$

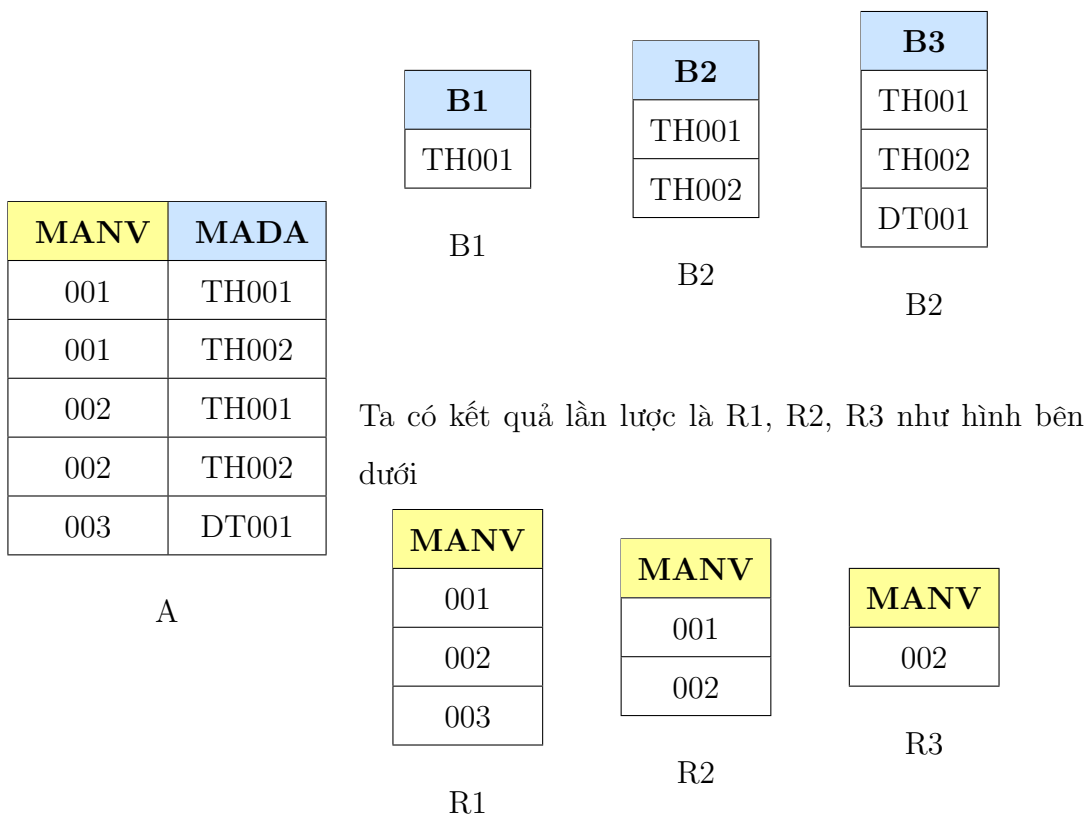
Với $Y = Z - X$

Có t là một bộ của T nếu với mọi bộ $t_S \in S$, tồn tại bộ $t_R \in R$ thỏa 2 điều kiện

$$t_R(Y) = t$$

$$t_R(X) = t_S(X)$$

Ví dụ 4.7: Thực hiện lần lượt A chia cho B1, B2, B3



Bảng 4.6: Ví dụ thực hiện phép chia các quan hệ

4.2.2 Phép toán quan hệ

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

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

Tên từ: Mỗi nhân viên có Mã nhân viên (MaNV) duy nhất để phân biệt với các nhân viên khác, có họ tên (HoNV, TenNV), ngày sinh (NgaySinh), địa chỉ (DiaChi), phái Nam

hoặc Nữ (Phai), mức lương (Luong), người quản lý trực tiếp (MaNQL) và thuộc về một phòng ban (Phong)

PHONGBAN (MaPhong, TenPhong, TruongPhong, NgayNhanChuc)

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

DIADIEMPHONG (MaPhong, DiaDiem)

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

DEAN (MaDA, TenDA, DdiemDA, Phong)

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

PHANCONG (MaNV, MaDA, ThoiGian)

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

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

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

NVIEN_TNHAN (MaNV, MaTN, QuanHe)

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

4.2.2.1 Phép chọn (selection)

- Được dùng để lấy ra các bộ của quan hệ R
- Các bộ được chọn phải thỏa mãn điều kiện chọn P
- Ký hiệu:

$$\sigma_P(R)$$

Trong đó:

- P là biểu thức gồm các mệnh đề có dạng
 - + $\langle \text{tên thuộc tính} \rangle \langle \text{phép so sánh} \rangle \langle \text{hằng số} \rangle$
 - + $\langle \text{tên thuộc tính} \rangle \langle \text{phép so sánh} \rangle \langle \text{tên thuộc tính} \rangle$
 - Phép so sánh gồm $<, >, \leq, \geq, \neq, =$.
 - Các mệnh đề được nối lại nhờ các phép \wedge, \vee, \neg
 - Kết quả trả về của phép chọn là một quan hệ
 - + Có cùng danh sách thuộc tính với R
 - + Có số bộ luôn ít hơn hoặc bằng số bộ của R
- ♦**Chú ý:** Phép chọn có tính giao hoán và kết hợp.

$$\sigma_{p1}(\sigma_{p2}(R)) = \sigma_{p2}(\sigma_{p1}(R))$$

Ví dụ 4.8:

- a. Chọn những nhân viên có lương ≥ 500000

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

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

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

4.2.2.2 Phép chiếu (projection)

- Được dùng để lấy ra một vài cột của quan hệ R
- Ký hiệu: $\pi_{A1, A2, \dots, Ak}(R)$
- Kết quả trả về là một quan hệ
 - + Có k thuộc tính
 - + Có số bộ luôn bằng số bộ của R

♦**Chú ý:** Phép chiếu không có tính giao hoán và kết hợp.

$$\pi_{X,Y}(R) \neq \pi_X(\pi_Y(R))$$

Ví dụ 4.9: Cho biết họ tên và lương của các nhân viên?

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

4.2.2.3 Phép gán (assignment)

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

Ký hiệu: \leftarrow

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

■ Ví dụ:

- B1 $S \leftarrow \sigma_P(R)$
- B2 $KQ \leftarrow \Pi_{A_1, A_2, \dots, A_K}(S)$

4.2.2.4 Phép đổi tên (Rename)

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

Các phép đổi tên sau:

- Đổi tên quan hệ:

Xét quan hệ R có 3 thuộc tính B, C và D, biểu thức $\rho_S(R)$ trả về kết quả là quan hệ S có 3 thuộc tính B, C, D.

- Đổi tên thuộc tính:

Xét biểu thức $\rho_{X,C,D}(R)$, kết quả trả về là quan hệ R gồm 3 thuộc tính X, C, D. Thuộc tính X thay thế cho thuộc tính B

- Đổi tên quan hệ và tên thuộc tính:

Xét biểu thức $\rho_{S(X,C,D)}(R)$, kết quả trả về là quan hệ R được thay thế bằng quan hệ S và thuộc tính B được thay thế bằng thuộc tính X.

4.2.2.5 Chuỗi các phép toán

Kết hợp các phép toán đại số quan hệ, lồng các biểu thức lại với nhau.

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

- Lồng các biểu thức lại với nhau

$$\boxed{\pi_{A_1, A_2, \dots, A_k}(\sigma_F(R))} \quad \text{Hoặc} \quad \boxed{\sigma_F(\pi_{A_1, A_2, \dots, A_k}(R))}$$

- Thực hiện từng phép toán một

<u>B1</u>	$\sigma_P(R)$
<u>B2</u>	$\pi_{A_1, A_2, \dots, A_k}(\text{Quan hệ kết quả ở B1})$

Ví dụ 4.10: Cho biết mã nhân viên, họ tên và lương của nhân viên làm việc trong phòng số 4.

- Cách 1:

$$\pi_{MaNV, HoNV, TenNV, Luong}(\sigma_{Phong=4}(NHANVIEN))$$

- Cách 2:

$$\begin{aligned} NVIEN_P4 &\leftarrow \sigma_{Phong=4}(NHANVIEN) \\ KQ &\leftarrow \pi_{MaNV, HoNV, TenNV, Luong}(NVIEN_P4) \end{aligned}$$

- Cách 3:

$$\begin{aligned} NVIEN_P4 &\leftarrow \sigma_{Phong=4}(NHANVIEN) \\ KQ &\leftarrow \pi_{MaNV, HoNV, TenNV, Luong}(NVIEN_P4) \end{aligned}$$

Trong đó có sử dụng phép gán và phép đặt tên

4.2.2.6 Phép tích đề các (Cartesian Product)

- Được dùng để kết hợp các bộ của các quan hệ lại với nhau.
- Cho 2 quan hệ $R(A_1, A_2, \dots, A_n)$ và $S(B_1, B_2, \dots, B_m)$
- Tích của R và S ký hiệu là: $R \times S$

- Kết quả trả về là một quan hệ Q

- Mỗi bộ của Q là tổ hợp giữa 1 bộ trong R và 1 bộ trong S
- Nếu R có u bộ và S có v bộ thì Q sẽ có $u \times v$ bộ
- Nếu R có n thuộc tính và S có m thuộc tính thì Q sẽ có $n + m$ thuộc tính
($R^+ \cap Q^+ \neq \phi$)

Ví dụ: Phép tích đề các giữa 2 quan hệ R và S

C	D
c1	d1
c2	d2

R

A	B
c1	d1
c2	d2

S

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

R x S

Bảng 4.7: Ví dụ thực hiện phép tích đề các giữa quan hệ R và quan hệ S

4.2.3 Các phép toán khác

4.2.3.1 Phép kết hai quan hệ (Join)

- Được dùng để tổ hợp 2 bộ có liên quan từ 2 quan hệ thành 1 bộ

- Ký hiệu: $R \bowtie S$

- Kết quả của $R \bowtie S$ là một quan hệ Q

+ Mỗi bộ của Q là tổ hợp của 2 bộ trong R và S, thỏa mãn một số điều kiện kết nào đó

- Có dạng $A_i \ominus B_j$
- A_i là thuộc tính của R, B_j là thuộc tính của S
- A_i và B_j có cùng miền giá trị
- \ominus là phép so sánh $\neq, =, <, >, \leq, \geq$

- Phân loại

+ Kết theta (theta join) là phép kết có điều kiện

- Ký hiệu $R \bowtie_C S$
- C gọi là điều kiện kết trên thuộc tính

+ Kết bằng (equi join) khi C là điều kiện so sánh bằng

+ Kết tự nhiên (natural join)

- Ký hiệu $R \bowtie S$ hay $R * S$
- $R^+ \cap Q^+ \neq \emptyset$
- Kết quả của phép kết bằng bỏ bớt đi 1 cột giống nhau

Ví dụ 4.12:

■ Phép kết theta

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

R

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

S

A	B	C	D	E
a2	2	1	d1	e1
a3	2	2	d1	e1

$$R^{R.B} \bowtie S^{S.C} = Q$$

Bảng 4.8: Ví dụ thực hiện phép kết theta của quan hệ R và S

■ Phép kết bằng

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

r

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

S

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

$$R^{R>B} \bowtie S^{S.C} = Q$$

Bảng 4.9: Ví dụ thực hiện phép kết bằng của quan hệ R và S

■ Phép kết tự nhiên

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

R

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

S

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

$R^{R>C} \bowtie S_{S>C} = Q$

Bảng 4.10: Ví dụ thực hiện phép kết tự nhiên của quan hệ R và S

■ Với mỗi phòng ban hãy cho biết thông tin của người trưởng phòng?

$$NHANVIEN \bowtie_{MANV=TruongPhong} PHONGBAN$$

4.2.3.2 Phép kết nối nội (inner join)

Thực chất của phép kết nối nội là phép kết nối bằng đã nêu trên. Tuy nhiên, ngay cả khi hai thuộc tính có cùng tên thì kết quả vẫn giữ lại 2 tên thuộc tính đó. **Ví dụ 4.15**
Phép kết nối nội

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3
a5	b5	c5
a7	b7	c7

R

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

S

A	B	C	D	E
a1	b1	c1	d1	e1
a2	b2	c2	d2	e2
a7	b7	c7	d7	e7

$R^{R.A} \bowtie S^{S>A} = Q$

Bảng 4.11: Sử dụng phép kết nối nội cho 2 quan hệ R và S

4.2.3.3 Phép kết ngoài (outer join)

Gồm phép kết nối trái và phép kết nối phải, cho phép giữ lại tất cả các bộ của hai quan hệ không tìm được bộ giá trị giống nhau trên các thuộc tính kết nối.

a. Phép kết nối trái (left join) \implies

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

$$R \stackrel{R.A = S.B}{\bowtie} S = \{v = (t, u) | (t \in R, u \in S : t.A \ominus u.B) \vee (t \in R, u \in u_{null} : t.A \in S[B])\}$$

Ví dụ: 4.16: Phép kết nối trái

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3
a5	b5	c5
a7	b7	c7

R

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

S

A	B	C	A	D	E
a1	b1	c1	a1	b1	c1
a2	b2	c2	a2	b2	c2
a3	b3	c3	null	null	null
a5	b5	c5	null	null	null
a7	b7	c7	a7	d7	e7

$$R \stackrel{R.A = S.B}{\bowtie} S \stackrel{S > A}{\bowtie} Q$$

Bảng 4.12: Sử dụng phép kết nối trái cho 2 quan hệ R và S

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

b. Phép kết nối phải (right join)

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

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

Ví dụ 4.17: Phép kết nối phải

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3
a5	b5	c5
a7	b7	c7

R

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

S

A	B	C	A	D	E
a1	b1	c1	a1	b1	c1
a2	b2	c2	a2	b2	c2
null	null	null	a4	d4	e4
null	null	null	a6	d6	e6
a7	b7	c7	a7	d7	e7

$$R^{R.A} \bowtie S^{S>A} = Q$$

Bảng 4.13: Sử dụng phép kết nối phải cho 2 quan hệ R và S

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

4.2.3.4 Các hàm kết hợp và gom nhóm

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

Trong đó:

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

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

Ví dụ 4.13: Cho quan hệ R như sau:

A	B
1	2
3	4
1	2
1	2

$$\text{SUM}(\mathbf{B}) = 10$$

$$\text{AVG}(\mathbf{A}) = 1.5$$

$$\text{MIN}(\mathbf{A}) = 1$$

$$\text{MAX}(\mathbf{B}) = 4$$

$$\text{COUNT}(\mathbf{A}) = 4$$

Bảng 4.14: Sử dụng các hàm kết hợp trên quan hệ R

Phép gom nhóm: Được dùng để phân chia quan hệ thành nhiều nhóm dựa trên điều kiện gom nhóm nào đó

Ký hiệu

$$G1, G2, \dots, Gn \vartheta_{F1(A1), F2(A2), \dots, Fn(An)}(E)$$

Trong đó:

- + E là biểu thức DSQH
- + G1, G2, ..., Gn là các thuộc tính gom nhóm
- + F1, F2, ..., Fn là các hàm
- + A1, A2, ..., An là các thuộc tính tính toán trong hàm F

Ví dụ 4.14: Cho quan hệ R sau:

A	B	C
α	2	7
α	4	7
β	2	3
γ	2	10

$$\sigma_{\text{SUM}(C)}(R)$$

$$A\sigma_{\text{SUM}(C)}(R)$$

Bảng 4.15: Sử dụng các hàm kết hợp trên quan hệ R, có gom nhóm

4.2.4 Các thao tác cập nhật trên quan hệ

- Nội dung của CSDL có thể được cập nhật bằng các thao tác:
 - + Thêm (insertion)
 - + Xóa (deletion)
 - + Sửa (updating)
- Các thao tác cập nhật được diễn đạt thông qua phép toán gán

$$R_{\text{new}} \leftarrow \text{Các phép toán trên } R_{\text{old}}$$

4.2.4.1 Thao tác thêm

- Được diễn đạt như sau:

$$R_{\text{new}} \leftarrow R_{\text{old}} \cup E$$

- Trong đó:
 - + R là quan hệ
 - + E là một biểu thức ĐSQH
- Phân công nhân viên có mã 123456789 làm thêm đề án mã số 20 với số giờ là 10.
Được viết bằng BT ĐSQH như sau:

$$\text{PHANCONG} \leftarrow \text{PHANCONG} \cup ('123456789', 20, 10)$$

4.2.4.2 Thao tác xóa

- Được diễn đạt như sau:

$$R_{new} \leftarrow R_{old} - E$$

- Trong đó:

- + R là quan hệ
- + E là một biểu thức ĐSQH

- Chú ý rằng phép xóa thực hiện xóa một hoặc nhiều bộ mà không thể xóa đi giá trị của các thuộc tính.

Ví dụ 4.19: Xóa các phân công đề án của nhân viên 123456789

$$\text{PHANCONG} \leftarrow \text{PHANCONG} - \sigma_{MANV='123456789'}(\text{PHANCONG})$$

4.2.4.3 Thao tác sửa

- Được diễn đạt như sau:

$$R_{new} \leftarrow \pi_{F1, F2, \dots, Fn}(R_{old})$$

- Trong đó:

- + R là quan hệ
- + Fi là biểu thức tính toán cho ra giá trị mới của thuộc tính

- Chú ý mỗi Fi có giá trị trả về là giá trị mới cho thuộc tính thứ i của R, thuộc tính này có thể được giữ nguyên hoặc cập nhật với giá trị mới.

Ví dụ 4.20: Tăng thời gian làm việc cho tất cả nhân viên lên 1.5 lần

$$\text{PHANCONG} \leftarrow \pi_{MANVIEN, MADA, THOIGIAN * 1.5}(\text{PHANCONG})$$

4.3 Tối ưu hóa câu truy vấn

- Nhìn chung có thể thấy được một quan hệ có “kích thước” càng nhỏ: có ít thuộc tính và ít bộ thì việc thực hiện các thao tác, các phép toán quan hệ trên đó càng ít tốn thời gian. Do đó cần biến đổi hợp lý để giảm thời gian tính toán đối với các biểu thức. Người ta gọi công việc biến đổi này là “Tối ưu hóa”. Một biểu thức quan hệ là tối ưu khi thời gian để thực thi nó là nhỏ và thực hiện bằng cách chuyển đổi vị trí của các phép toán để phép chiếu và phép chọn được thực hiện sớm nhất.
- Tuy nhiên cần tuân theo một số **nguyên tắc tổng quát tối ưu hóa biểu thức** như sau:
 1. Thực hiện phép chọn càng sớm càng tốt
 2. Gộp những phép chọn xác định với phép tích Đề-các thành phép kết nối.
 3. Gộp dãy các phép toán quan hệ một ngôi như các phép chọn và phép chiếu
 4. Tìm các biểu thức con chung trong một biểu thức
 5. Tiền xử lý các quan hệ tức là sắp xếp trước các bộ giá trị theo thứ tự lập chỉ mục.
 6. Đánh giá trước khi thực hiện tính toán.

Ví dụ 4.21: Cho CSDL quản lý việc thực hiện đề tài của sinh viên như sau:

SINHVIEN(MaSV, HoTen, NamSinh, QueQuan, HocLuc)

Tân từ: Mỗi sinh viên có một mã số duy nhất để phân biệt với các sinh viên khác, có họ tên sinh viên, năm sinh và quê quán của sinh viên **cùng** với học lực của sinh viên đó, học lực được lưu trữ dưới hình thức là xếp loại như Giỏi, khá, ...

DETAI(MaDT, TenDT, KinhPhi, ChuNhiem)

Tân từ: Mỗi đề tài có một mã đề tài duy nhất để phân biệt với những đề tài khác, có một tên đề tài, kinh phí cho đề tài đó và chủ nhiệm của đề tài (lưu họ tên chủ nhiệm đề tài). Kinh phí được tính bằng đơn vị triệu đồng.

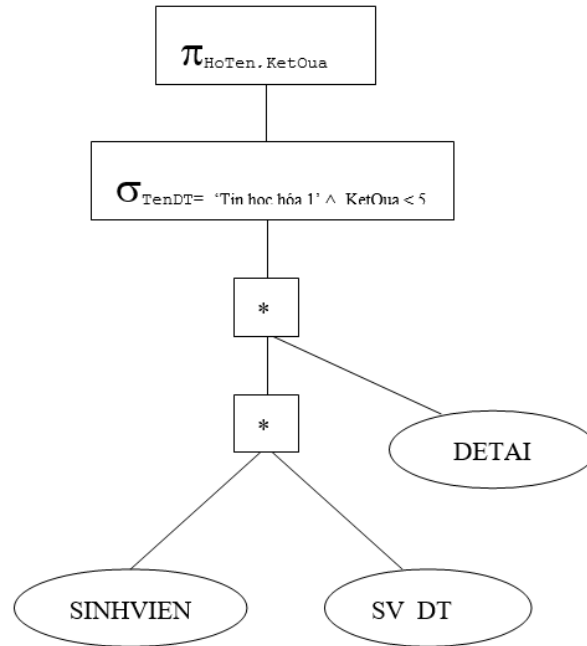
SV_DT(MaSV, MaDT, NoiAD, KetQua)

Tân từ: Mỗi sinh viên có thể tham gia nhiều đề tài và mỗi đề tài có thể có nhiều sinh viên tham gia, và đề tài đó được sinh viên thực hiện (hay áp dụng) ở một tỉnh thành nào đó, cũng cần phải lưu lại kết quả thực hiện đề tài đó, kết quả thực hiện đề tài được quy ra điểm từ 0 đến 10.

Hãy cho biết họ tên và kết quả thực hiện đề tài của những sinh viên tham gia đề tài ‘Tin học hóa 1’ có kết quả nhỏ hơn 5.

$\pi_{HoTen, KetQua}(\sigma_{TenDT='Tin học hóa 1' \wedge KetQua < 5}(SINHVIEN * SV_DT * DETAI))$

Cây thực hiện trong biểu thức này là:



Có thể giải thích việc thực hiện biểu thức như sau:

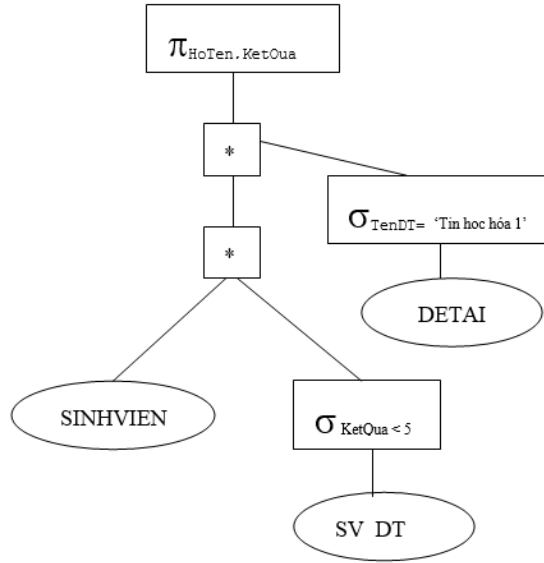
$$R1 \leftarrow SV * SV_DT$$

$$R2 \leftarrow R1 * DT$$

$$R3 \leftarrow \sigma_{TenDT='Tin học hóa 1' \wedge KetQua < 5}(R2)$$

$$KQ \leftarrow \pi_{HoTen, KetQua}(R3)$$

Chú ý: TenDT là thuộc tính của quan hệ DETAI nên người ta có thể đưa ra phép chọn điều kiện TenDT = ‘Tin học hóa 1’ trong quan hệ DETAI. Nghĩa là đẩy một phần phép chọn về phía lá. Tương tự như vậy, phép chọn KetQua < 5 được đẩy xuống để thực hiện sớm trong quan hệ SV_DT. Khi đó ta có cây biểu thức:



Giải thích việc thực hiện biểu thức trên như sau:

$$R1 \leftarrow \sigma_{TenDT='Tin học hóa 1'}(DETAI)$$

$$R2 \leftarrow \sigma_{KetQua < 5}(SV_DT)$$

$$R3 \leftarrow SINHVIEN * R2$$

$$R4 \leftarrow R3 * R1$$

$$KQ \leftarrow \pi_{HoTen, KetQua}(R4)$$

Ta có biểu thức tương ứng là:

$$\pi_{HoTen, KetQua}((SINHVIEN * \sigma_{KetQua < 5}(SV_DT)) * \sigma_{TenDT='Tin học hóa 1'}(DETAI))$$

► **Câu hỏi (bài tập) củng cố chương 4:**

Bài tập 1: Với CSDL ‘Quản lý đề án’ đã mô tả ở phần 4.2.2, hãy viết biểu thức DSQH cho các câu hỏi sau đây:

1. Cho biết thông tin cá nhân về những nhân viên có tên ‘Tùng’.
2. Tìm mã nhân viên, họ tên và địa chỉ của tất cả nhân viên làm việc phòng ‘Điều hành’.
3. Tìm mã nhân viên, họ tên và địa chỉ của tất cả nhân viên làm việc phòng ‘Quản lý’ và ‘Nghiên cứu’.
4. Cho biết mã nhân viên, họ tên nhân viên và tên các đề án mà nhân viên tham gia.
5. Tìm mã đề án, tên đề án, tên phòng ban chủ trì đề án cùng mã trưởng phòng, tên trưởng phòng đó.
6. Cho biết mã nhân viên, họ tên của những nhân viên tham gia vào đề án có mã là ‘TH001’ và có thời gian làm việc cho đề án trên 30giờ/tuần.
7. Cho biết mã nhân viên, họ tên của những nhân viên có cùng tên với người thân.
8. Cho biết mã nhân viên, họ tên của những nhân viên có trưởng phòng có họ tên là ‘Lê Thị Nhàn’.
9. Cho biết mã nhân viên, họ tên của những nhân viên có người quản lý có họ tên là ‘Trần Hồng Vân’.
10. Cho biết mã nhân viên, họ tên của những nhân viên tham gia mọi đề án của công ty.
11. Cho biết mã nhân viên, họ tên của những nhân viên không tham gia đề án nào của công ty.
12. Cho biết số lượng nhân viên từng phòng ban.
13. Cho biết số lượng nhân viên tham gia đề án ‘TH001’.
14. Cho biết họ tên những nhân viên có lương lớn hơn lương của các nhân viên ở phòng ‘Nghiên cứu’.
15. Cho biết họ tên các nhân viên có lương từ 2000000 đến 2500000.

Bài tập 2: Cho CSDL về việc thực hiện đề tài của sinh viên gồm các bảng sau:

SINHVIEI(MASV, HOTEN, NAMSINH, QUEQUAN, HOCLUC)

Tên từ: Mỗi sinh viên có một mã số phân biệt, một họ tên, năm sinh, quê quán của sinh viên, học lực của sinh viên được ghi nhận lại với hình thức là xếp loại của sinh viên đó như giỏi, khá,...

DETAI(MADT, TENDT, KINHPHI, CHUNHIEM)

Tên từ: Mỗi đề tài có một mã đề tài phân biệt, một tên đề tài, kinh phí dự trù cho đề tài đó, và họ tên chủ nhiệm của đề tài.

THUCHIEN(MASV, MADT, NOITH, DIEM)

Tên từ: Mỗi sinh viên có thể thực hiện nhiều đề tài khác nhau, mỗi đề tài cũng có thể do nhiều sinh viên thực hiện, khi thực hiện đề tài cũng cần phải ghi nhận lại nơi thực hiện đề tài, nơi này là tên một trong các tỉnh thành của đất nước Việt Nam. Ngoài ra cần lưu lại điểm số thực hiện đề tài của sinh viên.

Dùng các phép toán trên ĐSQH để trả lời các câu truy vấn sau:

1. Hãy cho biết tên đề tài do ‘Nguyễn Trung Kiên’ làm chủ nhiệm và có kinh phí hơn 4 triệu đồng?
2. Hãy cho biết họ tên sinh viên thực hiện đề tài được áp dụng ở ‘Hậu Giang’?
3. Hãy cho biết họ tên sinh viên không hề tham gia bất kỳ đề tài nào?
4. Hãy cho biết mã số sinh viên của những sinh viên có học lực ‘Giỏi’ nhưng điểm thực hiện đề tài nhỏ hơn 7?
5. Hãy cho biết kinh phí lớn nhất của các đề tài được thực hiện?
6. Hãy cho biết tổng kinh phí của các đề tài được áp dụng theo từng nơi?
7. Hãy cho biết họ tên các sinh viên đã thực hiện đề tài tại quê nhà?
8. Hãy cho biết danh sách các sinh viên thực hiện nhiều hơn 3 đề tài?
9. Hãy cho biết có bao nhiêu tỉnh thành có đề tài được thực hiện tại đó?

Chương 5

PHỤ THUỘC HÀM VÀ RÀNG BUỘC TOÀN VỆN CỦA LƯỢC ĐỒ QUAN HỆ

◆ **Mục tiêu học tập:** Sau khi học xong chương này, người học có thể:

- Vận dụng các tính chất của phụ thuộc hàm tìm chuỗi suy diễn
- Xác định được bao đóng của tập thuộc tính
- Xác định được phủ tối thiểu của tập phụ thuộc hàm
- Xác định được các yếu tố của ràng buộc toàn vẹn.

Phụ thuộc hàm (Functional Dependency) dùng để biểu diễn một cách hình thức các ràng buộc toàn vẹn (RBTV). Phụ thuộc hàm có tầm quan trọng rất lớn trong việc giải quyết các bài toán tìm khóa, phủ tối thiểu và chuẩn hóa cơ sở dữ liệu. Nội dung chương này cũng trình bày ràng buộc toàn vẹn (RBTV), các yếu tố liên quan đến ràng buộc toàn vẹn nhằm bảo đảm tính đúng đắn của dữ liệu.

5.1 Phụ thuộc hàm (PTH)

5.1.1 Định nghĩa

Quan hệ R được định nghĩa trên tập thuộc tính $U = \{A_1, A_2, \dots, A_n\}$. $A, B \subset U$ là hai tập con của thuộc tính U . Nếu tồn tại một ánh xạ $f: A \rightarrow B$ ta nói rằng A xác định hàm B ,

hay B phụ thuộc vào A. Kí hiệu : $A \rightarrow B$

Định nghĩa hình thức của phụ thuộc hàm như sau:

Quan hệ $Q(A, B, C)$ có phụ thuộc hàm A xác định B (Kí hiệu: $A \rightarrow B$) nếu:

$$\forall q, q' \in Q, \text{ sao cho } q.A = q'.A \text{ thì } q.B = q'.B$$

(Nghĩa là ứng với 1 giá trị của A thì có 1 giá trị duy nhất của B).

Ví dụ 5.1: Trong lược đồ CSDL quản lý hóa đơn bán hàng

Quan hệ HOA_DON(*MaHD*, MatHang, SoLoai, TongTriGia)

$$f_1 : MaHD \rightarrow MatHang, SoLoai$$

$$f_2 : MaHD \rightarrow TongTriGia$$

MaHD là khóa của Quan hệ HOA_DON, nếu biết được MaHD thì ta có thể xác định được tất cả các thông tin còn lại liên quan đến HOA_DON.

5.1.2 Hệ luật dẫn Armstrong

Gọi F là tập các PTH đối với quan hệ R định nghĩa trên tập thuộc tính U và $A \rightarrow B$ là một PTH; $A, B \subset U$. Ta nói, $A \rightarrow B$ được suy diễn logic từ F nếu R thỏa các PTH của F thì cũng thỏa $A \rightarrow B$ và ký hiệu là $F \mid A \rightarrow B$

Ví dụ 5.2: Với $F = \{X \rightarrow Y, X \rightarrow Z, Y \rightarrow Z\}$

Thì ta có các PTH $X \rightarrow YZ, X \rightarrow T$

- Bao đóng của tập phụ thuộc hàm

- + Bao đóng của tập phụ thuộc hàm F, ký hiệu là F^+ là tập tất cả các phụ thuộc hàm được suy diễn từ F.
- + Trong trường hợp $F = F^+$ thì F là họ đầy đủ của các phụ thuộc hàm.

- Các tính chất của tập bao đóng F^+

Bao đóng F^+ của tập hợp phụ thuộc hàm F có các tính chất:

1. Tính phản xạ: $F \subset F^+$
2. Tính đơn điệu: Nếu $F \subset G$ thì $F^+ \subset G^+$
3. Tính lũy đẳng: $(F^+)^+ = F^+$
4. $(FG)^+ \supseteq F^+G^+$
5. $(F^+G)^+ = (FG^+)^+ = (FG)^+$

Ví dụ 5.3: Với $F = \{A \rightarrow B, B \rightarrow C, A \rightarrow D, B \rightarrow D\}$ ta có:

$$F^+ = \{A \rightarrow B, B \rightarrow C, A \rightarrow D, B \rightarrow D, A \rightarrow C, B \rightarrow DC, A \rightarrow BC, \dots\}$$

Lưu ý: Trên lý thuyết, người ta hoàn toàn có thể xác định được thủ tục tìm bao đóng F^+ . Nhưng dù tập hợp thuộc tính U và tập hợp phụ thuộc hàm F là hữu hạn thì bài toán tìm bao đóng F^+ vẫn khó thực hiện vì có thể dẫn đến sự bùng nổ tổ hợp. Thay vào đó người ta thường xét bài toán “Kiểm tra một phụ thuộc hàm $X \rightarrow Y$ có thuộc bao đóng F^+ hay không?, nghĩa là nó có được suy diễn từ F hay không?”

- Các tính chất của PTH

- A1. Tính phản xạ: $X \rightarrow Y$, hay $Y \subset X$ thì $X \rightarrow Y$
- A2. Tính bắc cầu: $X \rightarrow Y$ và $Y \rightarrow Z \Rightarrow X \rightarrow Z$
- A3. Tính mở rộng 2 vế: $X \rightarrow Y \Rightarrow XZ \rightarrow YZ$
- A4. Tính tựa bắc cầu: $X \rightarrow Y, YZ \rightarrow W \Rightarrow XZ \rightarrow W$
- A5. Tính cộng đầy đủ: $X \rightarrow Y, Z \rightarrow W \Rightarrow XZ \rightarrow W$
- A6. Tính tích lũy: $x \rightarrow Y, Y \rightarrow ZW \Rightarrow X \rightarrow YZW$
- A7. Tính hợp: $X \rightarrow Y, X \rightarrow Z \Rightarrow X \rightarrow YZ$
- A8. Tính phân rã: $X \rightarrow YZ$ thì $X \rightarrow Y, X \rightarrow Z$

- Hệ luật dẫn Armstrong và các phép suy dẫn

+ Hệ A bao gồm ba tính chất {A1, A2, A3} đã nêu trong phần các tính chất của PTH ở trên được gọi là hệ luật tiên đề Armstrong của lớp các PTH.

Vậy $A = \{A1, A2, A3\}$ là *hệ tiên đề Armstrong*.

+ F là tập các PTH và f là một PTH trên R. Ta nói PTH f được suy dẫn theo hệ tiên đề Armstrong từ tập PTH F. Ký hiệu: $F \models f$ Nếu f có thể nhận được

từ F sau một số hữu hạn các bước áp dụng các luật A1, A2, A3 của hệ tiên đề Armstrong.

Ví dụ 5.4: Cho quan hệ $R = \{A, B, C, D\}$

$F = \{a \rightarrow B, A \rightarrow C, A \rightarrow D\}$, f là $A \rightarrow BCD$

Ta thấy f có thể nhận được từ phép cộng đầy đủ.

Định lý 1: Hệ tiên đề Armstrong là đúng đắn và đầy đủ.

Ví dụ 5.5: Cho lược đồ quan hệ R (A, B, C, D, E, G, H) và tập phụ thuộc hàm

$F = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A\}$

Áp dụng các tính chất của hệ tiên đề Armstrong tìm chuỗi suy diễn $AB \rightarrow E$

Giải:

Từ đề bài ta có:

(1) $AB \rightarrow C$

(2) $B \rightarrow D$

(3) $CD \rightarrow E$

(4) $CE \rightarrow GH$

(5) $G \rightarrow A$

(6) $AB \rightarrow BC$ mở rộng (1)

(7) $BC \rightarrow CD$ mở rộng (2)

(8) $AB \rightarrow CD$ bắc cầu (3)

(9) $AB \rightarrow E$ bắc cầu (3)(8)(đpcm)

5.1.3 Bao đóng của tập thuộc tính

Bao đóng của tập thuộc tính X đối với PTH F kí hiệu X^+ (hoặc X_F^+) là tập tất cả các thuộc tính A có thể suy dẫn từ X, nhờ tập bao đóng của các PTH F^+ .

$$X_F^+ = \{A | X \rightarrow A \neq F^+\}$$

5.1.4 Giải thuật tìm bao đóng của tập thuộc tính (X^+)

Bước 1: Gán $\Omega = F$

Bước 2: Gán $T = X$

Bước 3: Trong khi tồn tại $L \rightarrow R \in \Omega \wedge L \subseteq T \wedge R \not\subseteq T$ thực hiện:

$$T = T \cup \{R\}$$

$$\Omega = \Omega \setminus \{L \rightarrow R\}$$

Bước 4: Kết luận $(X)^+ = T$

Ví dụ 5.6: Cho quan hệ R (A, B, C, D, E, G) và tập phụ thuộc hàm $F = \{AB \rightarrow C, D \rightarrow EG, D \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG\}$

Hãy xác định bao đóng $(DB)^+$

Giải

B1: Gán $\Omega = F$

B2: Gán $T = DB$

B3: 3.1: Xét $D \rightarrow EG$ thỏa, nên thực hiện:

$$T = T \cup \{EG\} = BDEG$$

$$\Omega = \Omega \setminus \{D \rightarrow EG\} = \{AB \rightarrow C, D \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG\}$$

3.2: Xét $D \rightarrow A$ thỏa, nên thực hiện:

$$T = T \cup \{A\} = BDEGA$$

$$\Omega = \Omega \setminus \{D \rightarrow A\} = \{AB \rightarrow C, BE \rightarrow C, BC \rightarrow D, CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG\}$$

3.3: Xét $AB \rightarrow C$ thỏa, nên thực hiện:

$$T = T \cup \{C\} = BDEGAC$$

$$\Omega = \Omega \setminus \{AB \rightarrow C\} = \{BE \rightarrow C, BC \rightarrow D, CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG\}$$

B4: Dừng và kết luận $(DB)^+ = T = ABCDEG$

5.2 Phủ tối thiểu của tập PTH

- PTH có thuộc tính về trái dư thừa

Cho F là tập các PTH trên lược đồ quan hệ Q . Khi đó, PTH có dạng $\alpha A \rightarrow \beta \in F$, với A là một thuộc tính đơn lẻ. Ta nói A là thuộc tính dư thừa nếu có thể suy dẫn ra β từ α , Tức là $\alpha^+ \supseteq \beta$. Ngược lại $\alpha A \rightarrow \beta$ là PTH có thuộc tính về trái không dư thừa hay β phụ thuộc đầy đủ vào αA . $\alpha A \rightarrow \beta$ còn được gọi là PTH đầy đủ.

Ví dụ 5.7: Cho lược đồ quan hệ $R(A, B, C, D, E)$ và tập PTH

$$F = \{A \rightarrow B, BC \rightarrow D, C \rightarrow D\}$$

- Khi đó $A \rightarrow B, C \rightarrow D$ là những PTH đầy đủ.

+ Xét $BC \rightarrow D$

- Giả sử B thừa thì $C^+ = CD, D \subset C^+$ nên B dư thừa
- Giả sử C thừa thì $B^+ = B, D \not\subset B^+$ nên C không dư thừa

- Vậy $A \rightarrow B, C \rightarrow D$ và $B \rightarrow D$ là các PTH có thuộc tính về trái không dư thừa.

- Vấn đề là tìm các PTH đầy đủ tương ứng bằng cách loại bỏ các thuộc tính dư thừa.

- Thuật toán tìm các phụ thuộc hàm đầy đủ tương ứng

Thực hiện với mỗi phụ thuộc hàm $X \rightarrow Y \in F$

Bước 1: $X = A_1, A_2, \dots, A_n$ ($n \geq 2$, với $n=1$ thì $X \rightarrow Y$ là đầy đủ)

Đặt $Z = X$

Bước 2: Với mỗi A_i , thực hiện

$Tam = Z \setminus A_i$

Nếu $Tam \rightarrow Y \in F^+$ thì $Z = Tam$

- **Tập PTH có vẻ phải một thuộc tính**

Mỗi tập phụ thuộc hàm F đều tương đương với một tập phụ thuộc hàm G mà vẻ phải của các phụ thuộc hàm thuộc G chỉ gồm một thuộc tính

Ví dụ 5.8: Cho lược đồ quan hệ $R(A, B, C, D, E)$ và tập phụ thuộc hàm $F = \{A \rightarrow BC, A \rightarrow D, CD \rightarrow E\}$

Khi đó $G = \{A \rightarrow B, A \rightarrow C, A \rightarrow D, CD \rightarrow E\}$ là tập phụ thuộc hàm có vẻ phải một thuộc tính và $F \equiv G$

- **Tập phụ thuộc hàm không dư thừa**

F là tập phụ thuộc hàm không dư thừa nếu không tồn tại $F' \subset F$ sao cho $F' \equiv F$. Ngược lại F được gọi là tập phụ thuộc hàm dư thừa.

- **Thuật toán loại những phụ thuộc hàm dư thừa**

Với mỗi phụ thuộc hàm $X \rightarrow Y \in F$, nếu $X \rightarrow Y$ là thành viên của $F - \{X \rightarrow Y\}$ thì loại $X \rightarrow Y$ khỏi F .

- **Phủ tối thiểu của tập phụ thuộc hàm**

F được gọi là phủ tối thiểu của tập phụ thuộc hàm (hay tập phụ thuộc hàm tối thiểu) nếu thỏa:

- (i) F là tập phụ thuộc hàm có thuộc tính về trái không dư thừa
- (ii) F là tập phụ thuộc hàm có vẻ phải một thuộc tính
- (iii) F là tập phụ thuộc hàm không dư thừa

- **Thuật toán tìm phủ tối thiểu của tập phụ thuộc hàm**

Bước 1: Loại các thuộc tính có vẻ trái dư thừa của mọi phụ thuộc hàm

Bước 2: Phân rã các phụ thuộc hàm có vẻ phải nhiều thuộc tính thành các phụ thuộc hàm có vẻ phải một thuộc tính

Bước 3: Loại các phụ thuộc hàm dư thừa khỏi F

Ví dụ 5.9: Cho lược đồ quan hệ $R(A, B, C, D, E, G, H)$ và tập phụ thuộc hàm $F = \{B \rightarrow A, DA \rightarrow CE, D \rightarrow H, GH \rightarrow C, AC \rightarrow D\}$

Tìm phủ tối thiểu của F .

Giải

Bước 1: Loại các thuộc tính có vẻ trái dư thừa của mọi phụ thuộc hàm

+ Với $DA \rightarrow CE$:

- Giả sử D thừa thì $A \rightarrow CE \in F^+$: ta có $A_F^+ = A, CE \notin A_F^+$ nên D không thừa
- Giả sử A thừa thì $D \rightarrow CE \in F^+$: ta có $D_F^+ = DH, CE \notin D_F^+$ nên A không thừa

+ Với $GH \rightarrow C$:

- Giả sử G thừa thì $H \rightarrow C \in F^+$: ta có $H_F^+ = H, C \notin H_F^+$ nên G không thừa
- Giả sử H thừa thì $G \rightarrow C \in F^+$: ta có $D_F^+ = DH, C \notin G_F^+$ nên H không thừa

+ Với $AC \rightarrow D$:

- Giả sử A thừa thì $C \rightarrow D \in F^+$: ta có $C_F^+ = C, D \notin C_F^+$ nên G không thừa
- Giả sử C thừa thì $A \rightarrow D \in F^+$: ta có $A_F^+ = A, D \notin C_F^+$ nên H không thừa

Vậy mọi phụ thuộc hàm đều là đầy đủ.

Bước 2: Phân rã về phải ta có $F = \{B \rightarrow A, DA \rightarrow C, DA \rightarrow E, D \rightarrow H, GH \rightarrow C, AC \rightarrow D\}$

Bước 3: Loại các phụ thuộc hàm dư thừa:

- Với $B \rightarrow A$: ta có $B_{F \setminus \{B \rightarrow A\}}^+ = B, A \notin B_{F \setminus \{B \rightarrow A\}}^+$ nên $B \rightarrow A$ là không thừa
- Với $DA \rightarrow C$: ta có $AD_{F \setminus \{DA \rightarrow C\}}^+ = ADEH, C \notin AD_{F \setminus \{DA \rightarrow C\}}^+$ nên $DA \rightarrow C$ là không thừa
- Với $DA \rightarrow E$: ta có $AD_{F \setminus \{DA \rightarrow E\}}^+ = ADC H, E \notin AD_{F \setminus \{DA \rightarrow E\}}^+$ nên $DA \rightarrow E$ là không thừa
- Với $D \rightarrow H$: ta có $D_{F \setminus \{D \rightarrow H\}}^+ = D, H \notin D_{F \setminus \{D \rightarrow H\}}^+$ nên $D \rightarrow H$ là không thừa
- Với $GH \rightarrow C$: ta có $GH_{F \setminus \{GH \rightarrow C\}}^+ = GH, C \notin GH_{F \setminus \{GH \rightarrow C\}}^+$ nên $GH \rightarrow C$ là không thừa
- Với $AC \rightarrow D$: ta có $AC_{F \setminus \{AC \rightarrow D\}}^+ = AC, D \notin AC_{F \setminus \{AC \rightarrow D\}}^+$ nên $AC \rightarrow D$ là không thừa

Vậy phủ tối thiểu của $F := \{B \rightarrow A, DA \rightarrow C, DA \rightarrow E, D \rightarrow H, GH \rightarrow C, AC \rightarrow D\}$

5.3 Ràng buộc toàn vẹn (Integrity Constraint)

5.3.1 Định nghĩa

- RBTV là các điều kiện từ ứng dụng thực tế đảm bảo cho CSDL luôn ở trạng thái an toàn.
- Các điều kiện này phải luôn đúng sau mỗi thao tác làm thay đổi trạng thái của CSDL

Ví dụ 5.10: Trong CSDL quản lý sinh viên, ta có một số RBTV sau:

- R1: Mỗi lớp học phải có một mã số duy nhất để phân biệt với các lớp học khác trong trường.
- R2: Mỗi lớp học phải thuộc một khoa của trường
- R3: Mỗi sinh viên có một mã số riêng biệt, không trùng với bất cứ sinh viên nào khác.
- R4: Mỗi sinh viên phải đăng ký vào một lớp của trường.
- R5: Mỗi sinh viên được thi tối đa ba lần cho mỗi môn học.

Trong suốt quá trình khai thác CSDL, các RBTV phải được thoả mãn ở bất kỳ thời điểm nào nhằm đảm bảo cho CSDL luôn ở trạng thái an toàn và nhất quán về dữ liệu.

5.3.2 Các yếu tố của RBTV

Khi cần xác định RBTV ta cần chỉ rõ:

- Nội dung (tức là điều kiện) của RBTV, từ đó xác định các biểu diễn.
- Bối cảnh xảy ra RBTV: trên một hay nhiều quan hệ, cụ thể trên các quan hệ nào.
- Tầm ảnh hưởng của RBTV khả năng tính toàn vẹn dữ liệu bị vi phạm.
- Hành động cần phải có khi phát hiện có RBTV bị vi phạm.

5.3.2.1 Nội dung của RBTV

Điều kiện của RBTV là sự mô tả và biểu diễn hình thức nội dung của nó, có thể được biểu diễn bằng ngôn ngữ tự nhiên hoặc ngôn ngữ hình thức (mã giả, phép tính quan hệ, đại số quan hệ, ...).

- Ngôn ngữ tự nhiên: Dễ hiểu nhưng không chặt chẽ, logic.
- Ngôn ngữ hình thức: Chặt chẽ, cô đọng nhưng đôi lúc khó hiểu.

Ví dụ 5.11: Xét các RBTV

Giả sử ta có CSDL Quản lý hoá đơn bán hàng gồm các bảng sau:

HoaDon(*MaHD*, *SoLoaiMH*, *TongThanhTien*)

DM_Hang(*MaMH*, *TenMH*, *DVT*)

ChiTietHD(*MaHD*, *MaMH*, *SoLuong*, *DonGia*, *ThanhTien*)

Điều kiện của RBTV có thể biểu diễn như sau:

- Xét (R1):

- + Ngôn ngữ tự nhiên: “Mỗi hoá đơn có một mã số hoá đơn (*MaHD*) riêng biệt không trùng với hoá đơn khác”.
- + Ngôn ngữ hình thức: $\forall hd_1, hd_2 \in HoaDon, hd_1 \neq hd_2 \Rightarrow hd_1.MaHD \neq hd_2.MaHD$

- Xét (R2):

- + Ngôn ngữ tự nhiên: “Số chủng loại mặt hàng = số bộ của *ChiTietHD* có cùng *MaHD*”
- + Ngôn ngữ hình thức:
 $\forall hd \in HoaDon$ sao cho
 $hd.SoLoaiMH = count(cthd \in ChitietHD : cthd.MaHD = hd.MaHD)$

- Xét (R3):

- + Ngôn ngữ tự nhiên: “*TongThanhTien* của các mặt hàng trong *ChiTietHD* có cùng *MaHD* phải bằng *TongThanhTien*”.

+ Ngôn ngữ hình thức:

$\forall hd \in HoaDon$, thì : $hdTongThanhTien = SUM(cthd.ThanhTien)$ với $cthd \in ChiTietHD$ sao cho : $cthd.MaHD = hd.MaHD$

- Xét (R4):

+ Ngôn ngữ tự nhiên: “Mỗi bộ của ChiTietHD phải có MaMH thuộc về danh mục hàng”.

+ Ngôn ngữ hình thức: $ChiTietHD.MAHD \subseteq DM_Hang.MaMH$

5.3.2.2 Bối cảnh xảy ra RBTV

Bối cảnh của một RBTV là tập các quan hệ mà khi thao tác trên các quan hệ đó có khả năng làm cho ràng buộc toàn vẹn bị vi phạm. Xét bối cảnh của RBTV trong ví dụ 5.6 như sau:

- R1 có bối cảnh là quan hệ HoaDon
- R2 và R3 có bối cảnh liên quan đến hai quan hệ ChiTietHD và HoaDon
- R4 có bối cảnh là hai quan hệ ChiTietHD và DM_Hang

5.3.2.3 Tầm ảnh hưởng của RBTV

- Một RBTV có thể liên quan đến một số quan hệ, và chỉ khi có thao tác cập nhật (thêm, xoá, sửa) mới có nguy cơ dẫn đến vi phạm RBTV. Do đó cần xác định thao tác nào dẫn đến việc cần phải kiểm tra RBTV bằng cách lập bảng xác định tầm ảnh hưởng cho mỗi RBTV.
- Bảng này gồm 4 cột: cột 1 chứa tên quan hệ liên quan tới RBTV; ba cột tiếp theo là thao tác Thêm/ Xoá/ Sửa bộ giá trị của quan hệ.
- Nếu thao tác nào vi phạm RBTV đối với quan hệ nào thì tại giao điểm dòng và cột đó được ghi dấu (+). Nếu chỉ ảnh hưởng trên các thuộc tính thì ghi tên các thuộc tính đó ở phía dưới dấu (+). Nếu không ảnh hưởng thì ghi dấu (-); và nếu không ảnh hưởng vì lý do không sửa được thì ghi thêm dấu (*) ở phía trên của dấu (-) đó.

Ví dụ 5.12: Bảng tầm ảnh hưởng của RBTV R1

Quan hệ	Thêm	Xóa	Sửa
HoaDon	$+(MaHD)$	+	$- (*)$

Bảng tầm ảnh hưởng của RBTV R2

Quan hệ	Thêm	Xóa	Sửa
HoaDon	-	+	$+(SoLoaiMH)$
ChiTietHD	+	+	-

Bảng tầm ảnh hưởng của RBTV R3

Quan hệ	Thêm	Xóa	Sửa
HoaDon	-	+	$+(TongThanhTien)$
ChiTietHD	+	+	$+(ThanhTien)$

Bảng tầm ảnh hưởng của RBTV R4

Quan hệ	Thêm	Xóa	Sửa
ChiTietHD	$+(MaMH)$	+	$- (*)$
DM_Hang	-	+	$- (*)$

Để xác định tất cả các RBTV cần kiểm tra trên từng quan hệ \rightarrow xây dựng một bảng tầm ảnh hưởng tổng hợp các RBTV.

Quan hệ	HoaDon			ChiTietHD			DM_Hang		
RBTV	T	S	X	T	S	X	T	S	X
R1	$+(MaMH)$	$- (*)$	+						
R2	-	$+(SoLoaiMH)$	+	+	$- (*)$	+			
R3	-	$+(TongThanhTien)$	+	+	$+(ThanhTien)$	+			
R4				+	-	+	-	$- (*)$	

5.3.2.4 Hành động cần tiến hành khi RBTV bị vi phạm

Thông thường có 2 giải pháp:

- Đưa ra thông báo yêu cầu sửa dữ liệu. Thông báo phải đầy đủ, rõ ràng và lịch sự.
- Từ chối thao tác cập nhật. Giải pháp này là phù hợp đối với việc xử lý theo lô.

5.3.3 Phân loại RBTV

Trong CSDL có thể phát hiện nhiều RBTV, tuy nhiên có thể chia chúng thành hai loại:

- Ràng buộc toàn vẹn trong bối cảnh là một quan hệ cơ sở
- Ràng buộc toàn vẹn trong bối cảnh trên nhiều quan hệ cơ sở.

5.3.3.1 RBTV có bối cảnh là một quan hệ

a. RBTV về miền giá trị

Thuộc tính không chỉ đặc trưng bởi kiểu dữ liệu mà nó còn bị giới hạn bởi miền giá trị trong kiểu dữ liệu đó.

Ví dụ 5.13: Xét lược đồ quan hệ

NHANVIEN (MANV, HONV, TENLOT, TENNV, NGSINH, PHAI, DCHI, MA_NQL, PHONG, MLUONG)

Yêu cầu: Nêu đầy đủ các đặc trưng của RBTV về miền giá trị của quan hệ trên.

- Xét RBTV R1:

+ Nội dung:

- Phái của nhân viên chỉ có thể là ‘Nam’ hoặc ‘Nữ’
- $\forall n \in \text{NHANVIEN} : n.\text{PHAI} \in \{ 'Nam', 'N' \}$

+ Bối cảnh: quan hệ NHANVIEN

+ Bảng tầm ảnh hưởng:

R1	Thêm	Xóa	Sửa
NHANVIEN	+	-	+(PHAI)

b. RBTV liên thuộc tính

Đó là RBTV có liên quan đến nhiều thuộc tính của một quan hệ. Đó là các phụ thuộc tính toán hoặc một suy diễn từ giá trị của một hay nhiều thuộc tính trong cùng một bộ giá trị.

Ví dụ 5.14: Xét lược đồ quan hệ

DEAN (MADA, TENDA, DDIEM_DA, PHONG, NGBD_DK, NGKT_DK)

Yêu cầu: Nêu đầy đủ các đặc trưng của RBTV liên thuộc tính của quan hệ trên.

- Xét RBTV R2:

+ Nội dung:

- Với mọi đề án, ngày bắt đầu dự kiến (NGBD_DK) phải nhỏ hơn ngày kết thúc dự kiến (NGKT_DK)
- $\forall d \in \text{DEAN}, d[\text{NGBD_DK}] \leq d[\text{NGKT_DK}]$

+ Bối cảnh: quan hệ DEAN

+ Bảng tầm ảnh hưởng:

R2	Thêm	Xóa	Sửa
DEAN	+	-	+(NGBD_DK,NGKT_DK)

c. RBTV liên bộ

RBTV liên bộ là điều kiện ràng buộc giữa các bộ trên cùng một quan hệ

Ví dụ 5.10: Xét lược đồ quan hệ:

PHONGBAN(MAPHG, TENPHG, TRPHG,NGNC)

- Xét RBTV R3:

+ Nội dung:

- Hai bộ khác nhau sẽ khác nhau tại giá trị khóa
- $\forall p1, p2 \in \text{PHONGBAN} : p1 \neq p2$
 $\Rightarrow p1, p2 \in \text{PHONGBAN} : p1 \neq p2$

+ Bối cảnh: quan hệ PHONGBAN

+ Bảng tầm ảnh hưởng:

R3	Thêm	Xóa	Sửa
PHONGBAN	+	-	-(*)

5.3.3.2 RBTV có bối cảnh trên nhiều quan hệ

a. RBTV tham chiếu (RBTV về phụ thuộc tồn tại hay RBTV khoá ngoại)

- RBTV tham chiếu là ràng buộc quy định giá trị của thuộc tính trong một bộ của quan hệ R phải thuộc tập giá trị của thuộc tính khóa trong quan hệ S khác.
- RBTV tham chiếu còn gọi là ràng buộc phụ thuộc tồn tại hay ràng buộc khóa ngoại.

Ví dụ 5.11: Xét các lược đồ quan hệ

PHONGBAN(MAPHG, TENPHG, TRPHG, NGNC)

NHANVIEN (MANV, HONV, TENLOT, TENNV, NGSINH, PHAI, DCHI, MA_NQL, PHONG, MLUONG)

- Xét RBTV R4:

+ Nội dụng:

- Mỗi trưởng phòng phải là một nhân viên trong công ty.
- $\forall p \in PHONGBAN, \exists n \in NHANVIEN : n[MANV] = p[TRPHG]$
(hay: $PHONGBAN[TRPHG] \subseteq NHANVIEN[MANV]$)

+ Bối cảnh: NHANVIEN, PHONGBAN

+ Bảng tầm ảnh hưởng:

R4	Thêm	Xóa	Sửa
PHONGBAN	+	-	+(TRPHG)
NHANVIEN	-	+	-(*)

b. RBTV liên bộ - liên quan hệ

RBTV liên bộ, liên quan hệ là điều kiện giữa các bộ trên nhiều quan hệ khác nhau.

Ví dụ 5.12: Xét các lược đồ quan hệ

PHONGBAN(MAPHG, TENPHG, TRPHG, NGNC)

DIADIEM_PHG(MAPHG, DIADIEM)

- Xét RBTV R5:

+ Nội dung

- Mỗi phòng ban phải có ít nhất một địa điểm phòng
- $\forall p \in PHONGBAN, \exists d \in DIADIEM_PHG : p.MAPHG = d.MAPHG$

+ Bối cảnh: PHONGBAN, DIADIEM_PHG

+ Bảng tầm ảnh hưởng:

R5	Thêm	Xóa	Sửa
PHONGBAN	+	-	+(MAPHG)
DIADIEM_PHG	-	+	+(MAPHG)

c. RBTV liên thuộc tính - liên quan hệ

RBTV liên thuộc tính, liên quan hệ là điều kiện giữa các thuộc tính trên nhiều quan hệ khác nhau.

Ví dụ 5.13: Xét các lược đồ quan hệ

DATHANG(MADH, MAKH, NGAYDH)

GIAOHANG_PHG(MAGH, MADH, NGAYGH)

- Xét RBTV R6:

+ Nội dung

- Ngày giao hàng không được trước ngày đặt hàng
- $\forall g \in GIAOHANG, \exists! d \in DATHANG : d.[MADH] = g.[MADH] \wedge d[NGAYDH] \leq g[NGAYGH]$

+ Bối cảnh: PHONGBAN, DIADIEM_PHG

+ Bảng tầm ảnh hưởng:

R6	Thêm	Xóa	Sửa
DATHANG	-	-	+(ngaydh)
GIAOHANG	+	-	+(ngaydh)

d. RBTV do thuộc tính tổng hợp

- Thuộc tính tổng hợp là thuộc tính được tính toán từ giá trị của các thuộc tính khác hoặc trạng thái của CSDL.
- Khi có thuộc tính tổng hợp, cần phải có RBTV để đảm bảo mối quan hệ giữa nó và nguồn mà nó được tính toán từ đó.

Ví dụ 5.14: Xét các lược đồ quan hệ

PXUAT(SOPHIEU, NGAY, TONGTRIGIA)

CTIET_PX(SOPHIEU, MAHANG, SL, DG)

- Xét RBTV R7:

+ Nội dung

- Tổng trị giá của 1 phiếu xuất phải bằng tổng trị giá các chi tiết xuất.
- $\forall px \in PXUAT, px.TONGTRIGIA = \Sigma(ct \in CTIET_PX \wedge ct.SOPHIEU = px.SOPHIEU) ct.SL * ct.DG$

+ Bối cảnh: PXUAT, CTIET_PX

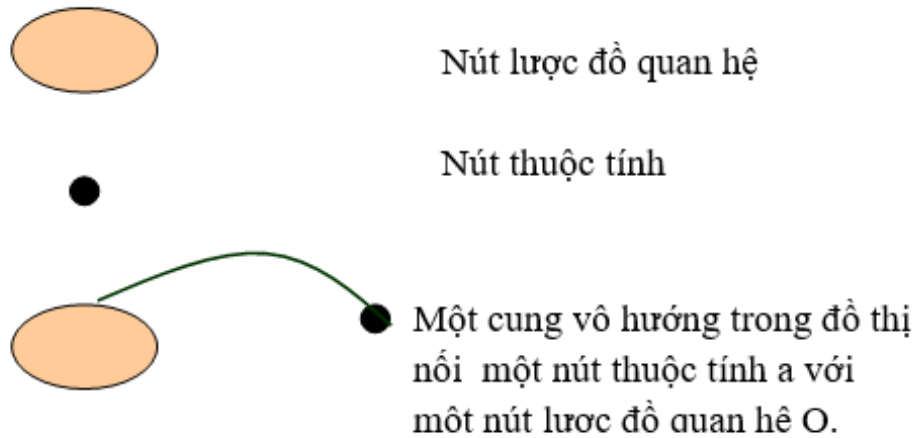
+ Bảng tầm ảnh hưởng:

R7	Thêm	Xóa	Sửa
PXUAT	-(*)	+	+(tongtrigia)
CTIET_PX	+	+	+(sl,dg)

e. RBTV có sự hiện diện của chu trình

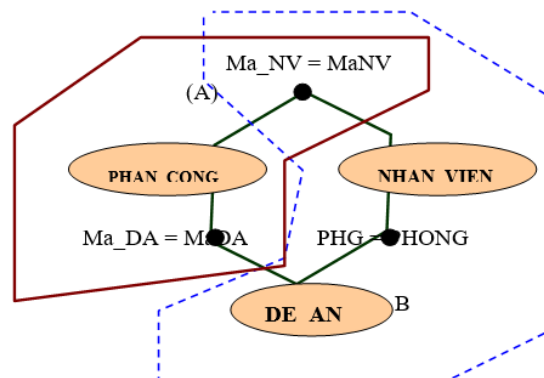
Đồ thị biểu diễn của lược đồ CSDL

Một lược đồ CSDL có thể được biểu diễn bằng một đồ thị vô hướng



Hình 5.1: Đồ thị biểu diễn lược đồ CSDL

Lược đồ CSDL có sự hiện diện của một chu trình khi xuất hiện một đường khép kín trên đồ thị.



Hình 5.2: Đồ thị biểu diễn lược đồ CSDL ‘Quản lý đề án công ty’

Một phần của đồ thị biểu diễn lược đồ CSDL, Quản lý đề án công ty gồm có các lược đồ quan hệ NHAN_VIEN, DE_AN, PHAN_CONG

Trong hình vẽ trên, ta nhận thấy đồ thị biểu diễn chứa một chu trình gồm 2 lược đồ quan hệ NHAN_VIEN, DE_AN, PHAN_CONG. Do đó, xuất hiện một RBTV thuộc một trong 3 trường hợp sau:

- Một nhân viên chỉ tham gia những đề án do phòng mình chủ trì, và phải tham gia đầy đủ tất cả các đề án đó.
- Một nhân viên chỉ tham gia những đề án do phòng mình chủ trì, và có thể không tham gia đầy đủ tất cả các đề án đó.

- Một nhân viên có thể tham gia tùy ý các đề án nào trong công ty, dù đề án đó không phải do phòng ban của mình chủ trì.

Ví dụ 5.15: Xét các lược đồ quan hệ

NHAN_VIEN(MaNV, HoTen, Gioi, PHG)

DE_AN(MaDA, TenDA, DiaDiem, PHONG)

PHAN_CONG(Ma_NV, Ma_DA)

- Xét RBTV R8:

+ Nội dung

- Một nhân viên chỉ được phân công vào các đề án do phòng mình chủ trì, và có thể không tham gia đầy đủ tất cả các đề án đó.
- $\forall pc \in PHANCONG(\exists nvda \in NV_DA(nvda.MaNV = pc.MaNV \wedge nvda.MaDA = pc.MaDA))$

+ Bối cảnh: NHANVIEN, DEAN, PHANCONG

+ Bảng tầm ảnh hưởng:

R8	Thêm	Xóa	Sửa
NHANVIEN	-	+	+(MaNV, Phong)
DEAN	-	+	+(MaDA, Phong)
PHANCONG	+	-	+(MaDA, MaNV)

► Câu hỏi (bài tập) củng cố:

Bài tập 1: Cho $R = (A\ B\ C\ D\ E\ G\ H\ I)$ và tập phụ thuộc hàm

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

Hãy áp dụng các tính chất của phụ thuộc hàm để tìm chuỗi suy diễn $CG \rightarrow BD$

Bài tập 2: Cho $R(ABCDEFGH)$ và tập phụ thuộc hàm

$$F = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A\}$$

Tìm chuỗi suy diễn $AB \rightarrow BG$

Bài tập 3: Cho $U = A\ B\ C\ D\ E\ G$ và tập phụ thuộc hàm

$$F = \{C \rightarrow G, BG \rightarrow CD, AEG \rightarrow BC, CG \rightarrow AE, B \rightarrow CG\}$$

a) Tính $(C)^+$

b) Tính $(B)^+$

c) Tính $(AEG)^+$

Bài tập 4: Cho $R(U, F)$ với $U = A\ B\ C\ D\ E\ G\ H\ K$ và

$$F = \{B \rightarrow ACDEG, D \rightarrow C, E \rightarrow GA\}. \text{ Tìm phủ tối thiểu của } F?$$

Bài tập 5: Tìm phủ tối thiểu của $F = \{AB \rightarrow CD, BC \rightarrow DEF, D \rightarrow CEF, ADE \rightarrow G, ABC \rightarrow G, ABD \rightarrow G\}$?

Bài tập 6: Cho $R(U, F)$ với $U = A\ B\ C\ D\ E\ I$ và $F = \{BE \rightarrow C, ET \rightarrow B, C \rightarrow DI, BC \rightarrow AB, EI \rightarrow C\}$. Tìm phủ tối thiểu?

Bài tập 7: Với CSDL ‘Quản lý đề án’ đã cho trong chương 4, hãy nêu đầy đủ các đặc trưng của tất cả các RBTV có trong bài. Với các RBTV tìm được hãy phân loại theo từng loại RBTV.

Chương 6

CHUẨN HÓA LƯỢC ĐỒ QUAN HỆ

♦ **Mục tiêu học tập:** Sau khi học xong chương này, người học có thể:

- Tìm khóa cho lược đồ quan hệ
- Xác định các dạng chuẩn của lược đồ quan hệ

6.1 Chuẩn hóa CSDL

Để mô tả một cách tổng quát các mức độ chuẩn hóa CSDL quan hệ người ta có hình vẽ sau đây:



Hình 6.1: Các mức độ chuẩn hóa CSDL

- Một cách chung nhất, quan hệ chuẩn hóa là quan hệ mà mỗi giá trị thuộc tính trong một bộ là những thuộc tính nguyên tố, không phân chia ra được. Nói cách khác, mỗi tọa độ của hàng và cột trong quan hệ chỉ có đúng một giá trị chứ không phải một tập hợp các giá trị. Cơ sở của việc chuẩn hóa là khái niệm phụ thuộc hàm, phụ thuộc hàm đầy đủ, khóa, ... Một CSDL được coi là chuẩn hóa tốt khi trong đó mỗi thuộc tính không khóa phụ thuộc hàm vào khóa.

6.2 Khóa trên lược đồ quan hệ

6.2.1 Định nghĩa

- Cho lược đồ quan hệ $Q(A_1, A_2, \dots, A_n)$, Q^+ là tập thuộc tính của quan hệ Q, F là tập phụ thuộc hàm trên Q, K là tập con của Q^+ . Khi đó K gọi là một khóa của Q nếu:

(roman*) $K_F^+ = Q^+$

(roman*) Không tồn tại $K' \subset K$ sao cho $K_F'^+ = Q^+$

- Thuộc tính A được gọi là **thuộc tính khóa** nếu $A \in K$, trong đó K là khóa của Q. Ngược lại thuộc tính A được gọi là **thuộc tính không khóa**.
- K' được gọi là siêu khóa nếu $K \subseteq K'$.

6.2.2 Thuật toán tìm một khóa

Bước a: Trong lược đồ quan hệ Q

- Xác định tập thuộc tính nguồn (thuộc tính chỉ nằm bên vế trái của PTH F)
- Xác định tập thuộc tính đích (thuộc tính chỉ nằm bên vế phải của PTH F)
- Xác định tập thuộc tính trung gian (thuộc tính chỉ nằm cả hai bên của PTH F)

Bước b: Xuất phát từ tập thuộc tính nguồn K tìm bao đóng của tập nguồn $(K)^+$.

- Nếu $(K)^+ = Q^+$ thì kết luận K là khóa của Q
- Ngược lại nếu $(K)^+ \subset Q^+$ thì bỏ từng thuộc tính trung gian X_i vào $K' = K \cup X_i$ rồi thực hiện tìm bao đóng của K' . Dừng khi tìm được một khóa của quan hệ Q .

Bước c: Kết luận khóa của Q

Ví dụ 6.1: Cho lược đồ quan hệ $Q(A, B, C, D, E, G, H)$ và tập PTH $F = \{B \rightarrow A, DA \rightarrow CE, D \rightarrow H, GH \rightarrow C, AC \rightarrow D\}$. Tìm một khóa của Q . **Giải**

Phân rã vế phải, ta có: $F = \{B \rightarrow A, DA \rightarrow C, DA \rightarrow E, D \rightarrow H, GH \rightarrow C, AC \rightarrow D\}$

Bước 1:

- Tập nguồn: BG
- Tập đích: E
- Tập trung gian: ACDH $\{A, C, D, H, AC, AD, AH, \dots CADH\}$

Bước 2:

- $(BG)^+ = BGA \subset Q^+$ nên BG không là khóa. Thêm C vào
- $(BGC)^+ = BCGADHE = Q^+$ nên BCG là khóa của Q

Bước 3: Kết luận BCG là khóa của quan hệ Q

6.2.3 Thuật toán tìm tất cả các khóa

- Tập thuộc tính nguồn, ký hiệu là N, là tập chứa những thuộc tính chỉ xuất hiện ở vế trái, không nằm bên vế trái và vế phải của mọi phụ thuộc hàm
- Tập thuộc tính trung gian, ký hiệu là TG, là tập chứa những thuộc tính vừa xuất hiện ở vế trái, vừa xuất hiện ở vế phải trong các phụ thuộc hàm

Thuật toán tìm tất cả các khóa

Bước 1: Tính tập nguồn N. Nếu $N_F^+ = Q^+$ thì chỉ có 1 khóa là N, ngược lại qua bước 2.

Bước 2: Tính tập trung gian (TG). Tính tập tất cả các tập con X_i của tập TG.

Bước 3: Tìm tập S chứa mọi siêu khóa S_i :

Với mỗi X_i nếu $(N \cup X_i)_F^+ = Q^+$ thì $S_i = (N \cup X_i)$.

Bước 4: Loại các siêu khóa không tối thiểu: $\forall D_i, S_j \in S$, nếu $S_i \subset S_j$ thì $S = S - S_j$.

Ví dụ 6.2: Cho lược đồ quan hệ $Q(A, B, C)$ và tập PTH $F = \{AB \rightarrow C, C \rightarrow A\}$

Tìm mọi khóa của Q ?

TN: $B(B)^+ = B$

TD: RỖNG

TTG: $AC\{\emptyset, A, C, AC\}$

Giải

Bước 1: $N = \{B\}, N_F^+ = B \neq Q^+$

Bước 2: $TG = \{AC\}$, tập các tập con trung gian là $CTG = \{\emptyset, A, C, AC\}$

Bước 3: Thêm từng tập con trung gian vào như bảng bên dưới

TN	X_i	$X_i \cup TN$	$(X_i \cup TN)^+$	Siêu khóa	Khóa
B	\cup	B	B		
B	A	AB	BAC	AB	AB
B	C	BC	BCA	BC	BC
B	AC	ABC	ABC	ABC	

Bước 4: Loại các siêu khóa không tối tiểu:

Nhận thấy rằng $BA \subset BAC$ nên loại siêu khóa BAC.

Tập các khóa còn lại là $S = BA, BC$

6.3 Dạng chuẩn của lược đồ quan hệ

Để dễ dàng trình bày các dạng chuẩn, cần nắm rõ các khái niệm:

- **Thuộc tính khóa, thuộc tính không khóa** (đã trình bày trong mục 6.2.1 định nghĩa khóa của lược đồ quan hệ)

Ví dụ 6.3: $R(A, B, C, D), F = \{AB \rightarrow C, B \rightarrow D, BC \rightarrow A\}$

Trong ví dụ trên, lược đồ R có 2 khóa là AB, BC . Khi đó A, B, C là thuộc tính khóa, D là thuộc tính không khóa.

- **Giá trị nguyên tố**

Giá trị nguyên tố là giá trị không phân nhỏ được nữa.

a. **Dạng chuẩn 1 (*First Normal Form* – 1NF)**

Lược đồ Q ở dạng chuẩn 1 nếu mọi thuộc tính đều mang giá trị nguyên tố.

Trong bài toán xét dạng chuẩn, dạng chuẩn thấp nhất là dạng chuẩn 1.

Ví dụ 6.4: Cho lược đồ HOADON(MaHD, MaKH, NgayHD, CtietMua, SoTien), và có thể hiện như sau:

MaHD	MaKH	NgayHD	CtietMua			SoTien
			Tên hàng	Số lượng	ĐVT	
HD01	KH01	15-10-05	Bánh Orion	1	Gói	25.000
			Kẹo mút	2	Cây	2.000
HD02	KH01	18-10-05	Gạo	2	Kg	30.000
HD03	KH02	24-10-05	Đường	1	Kg	15.000
			Bánh AFC	2	Gói	24.000

Bảng 6.1: Ví dụ minh họa bảng HOADON

Ở ví dụ này CTiếtMua không mang giá trị nguyên tố, do đó HOADON không đạt dạng chuẩn 1. Người ta tách các giá trị này ra và được một bảng mới như sau:

MaHD	MaKH	NgayHD	CtietMua			SoTien
			Tên hàng	Số lượng	ĐVT	
HD01	KH01	15-10-05	Bánh Orion	1	Gói	25.000
HD01	KH01	15-10-05	Kẹo mút	2	Cây	2.000
HD02	KH01	18-10-05	Gạo	2	Kg	30.000
HD03	KH02	24-10-05	Đường	1	Kg	15.000
HD03	KH02	24-10-05	Bánh AFC	2	Gói	24.000

Tuy nhiên khi chọn thuộc tính MaHD làm khóa thì bảng mới này vẫn không thỏa dạng chuẩn thứ nhất vì có những thuộc tính không phụ thuộc hàm vào khóa. Người ta tách bảng này thành hai bảng như sau:

<u>MaHD</u>	<u>MaKH</u>	<u>NgayHD</u>
HD01	KH01	15-10-05
HD02	KH01	18-10-05
HD03	KH02	24-10-05

CtietMua				
<u>MaHD</u>	<u>Tên hàng</u>	<u>Số lượng</u>	<u>ĐVT</u>	<u>SoTien</u>
HD01	Bánh Orion	1	Gói	25.000
HD01	Kẹo mút	2	Cây	2.000
HD02	Gạo	2	Kg	30.000
HD03	Đường	1	Kg	15.000
HD03	Bánh AFC	2	Gói	24.000

Hai bảng kết quả sau cùng đều thỏa dạng chuẩn thứ nhất.

b. Dạng chuẩn 2 (*Second Normal Form - 2NF*)

- Định nghĩa

Lược đồ Q ở dạng chuẩn 2 nếu thỏa:

- (1) Q đạt dạng chuẩn 1
- (2) Mọi thuộc tính không khóa của Q đều phụ thuộc đầy đủ vào khóa

- Kiểm tra dạng chuẩn 2

Để kiểm tra dạng chuẩn 2 thực hiện:

Bước 1: Tìm mọi khóa của Q

Bước 2: Với mỗi khóa K, tìm bao đóng của tập tất cả các tập con thực sự S_i của K

Bước 3: Nếu tồn tại bao đóng S_i^+ chứa thuộc tính không khóa thì Q không đạt dạng chuẩn 2, ngược lại Q đạt dạng chuẩn 2.

Xét quan hệ

<u>MaSV</u>	<u>MaDT</u>	<u>TenDT</u>	<u>KetQua</u>
001	1	CSDL	8
002	2	PTHT	7
003	1	CSDL	6
004	2	PTHT	8

Trên quan hệ này ta có:

- Khóa: MaSV, MaDT
- - Có MaDT \rightarrow TenDT: phụ thuộc không đầy đủ vào khóa

Nên quan hệ trên không đạt dạng chuẩn thứ hai. Người ta tách quan hệ này thành hai quan hệ có dạng như sau:

MaDT	TenDT
1	CSDL
2	PTHT

MaSV	MaDT	KetQua
001	1	8
002	2	7
003	1	6
004	2	8

Hai quan hệ kết quả sau cùng này đều đạt dạng chuẩn thứ hai.

Ví dụ 6.5: Cho Q1 (A, B, C, D), và phụ thuộc hàm $F = \{A \rightarrow B, B \rightarrow DC\}$. Tìm khóa và xác định dạng chuẩn cho Q1?

Giải

- Tập nguồn (TN): A
 - Tập trung gian (TG): B
 - Tập đích (TĐ): DC
- $$(A)^+ = ABCD$$

Vậy A là khóa của quan hệ Q1

- Xét dạng chuẩn 2NF

Ta có $A \rightarrow CD$ thuộc tính không khóa phụ thuộc đầy đủ vào khóa ($A \rightarrow B, B \rightarrow CD$ (phụ thuộc bắc cầu)) nên Q1 đạt dạng chuẩn 2NF

Ví dụ 6.6: Cho $Q2 (A, B, C, D)$, $F = \{AB \rightarrow C, B \rightarrow D, BC \rightarrow A\}$. Tìm khóa và xác định dạng chuẩn cho $Q2$?

Giải

- TN: B
- TG: AC
- TD: D

$$(B)^+ = BD \subset Q2$$

X_i	$X_i \cup TN$	$(X_i \cup TN)^+$	Siêu Khóa	Khóa
\emptyset	B	BD		
A	AB	ABCD	AB	AB
C	BC	ABCD	BC	BC
AC	ABC	ABCD	ABC	

Khóa là $K_1 = AB$ và $K_2 = BC$. Ta thấy $B \subset K_1, B \rightarrow D$, D là thuộc tính không khóa

\Rightarrow thuộc tính không khóa không phụ thuộc đầy đủ vào khóa nên $Q2$ không đạt dạng chuẩn 2NF.

c. Dạng chuẩn 3 (*Three Normal Form - 3NF*)

- Định nghĩa 1

Với một lược đồ Q ; X, Y là hai tập con của Q^+ ; A là một thuộc tính. Khi đó A được gọi là phụ thuộc bắc cầu vào X nếu thỏa:

- (1) $X \rightarrow Y, Y \rightarrow A$
- (2) $Y \rightarrow X$
- (3) $A \in XY$

- Định nghĩa 2

Lược đồ Q ở dạng chuẩn 3 nếu mọi phụ thuộc hàm $X \rightarrow A \in F^+$, với $A \notin X$ đều có:

- (1) X là siêu khóa, hoặc
- (2) A là thuộc tính khóa

-Kiểm tra dạng chuẩn 3

Từ định nghĩa 2, để kiểm tra dạng chuẩn 3 thực hiện các bước sau: **Bước 1:**

Tìm mọi khóa của Q

Bước 2: Phân rã vế phải của mọi phụ thuộc hàm trong F để tập F trở thành tập phụ thuộc hàm có vế phải một thuộc tính

Bước 3: Nếu mọi phụ thuộc hàm $X \rightarrow A \in F$, mà $A \notin X$ đều thỏa

(1) X là siêu khóa (vế trái chứa một khóa), hoặc

(2) A là thuộc tính khóa (vế phải là tập con của khóa)

thì Q đạt dạng chuẩn 3, ngược lại Q không đạt dạng chuẩn 3.

Xét quan hệ

<u>MaSV</u>	TenSV	QueQuan	KCach
001	A	VL	30
002	B	ST	60
003	C	VL	30
004	D	HG	60

Trên quan hệ này ta có:

- Khóa: MaSV
- $QueQuan \rightarrow KCach$: phụ thuộc bắc cầu vào khóa MaSV, nên không đạt dạng chuẩn thứ ba nhưng thỏa dạng chuẩn thứ 2. Người ta tách quan hệ này thành hai quan hệ như sau:

Hai quan hệ kết quả sau cùng này đều đạt dạng chuẩn thứ ba.

Ví dụ 6.7: Cho Q3 (A, B, C, D), và phụ thuộc hàm $F = \{A \rightarrow B, B \rightarrow DC\}$. Tìm khóa và xác định dạng chuẩn cho Q3?

Giải

- Tập nguồn(TN): A
 - Tập trung gian(TG): B
 - Tập đích(TĐ): DC
- $$(A)^+ = ABCD$$

Vậy A là khóa của quan hệ Q3

- Xét dạng chuẩn 2NF

Ta có $A \rightarrow CD$ thuộc tính không khóa phụ thuộc đầy đủ vào khóa ($A \rightarrow B, B \rightarrow CD$ (phụ thuộc bắc cầu)) nên Q3 đạt dạng chuẩn 2NF

- Xét dạng chuẩn 3NF

Do $A \rightarrow CD$ (thuộc tính không khóa phụ thuộc bắc cầu vào khóa) nên Q3 không đạt dạng chuẩn 3NF

Ví dụ 6.8: Cho lược đồ quan hệ Q4(A, B, C, D) và $F = \{AB \rightarrow C, D \rightarrow B, C \rightarrow ABD\}$.

Tìm khóa và xác định dạng chuẩn của Q4 ?

Giải

- TN: \emptyset
- TG: ABCD
- TD: \emptyset

X_i	$X_i \cup TN$	$(X_i \cup TN)^+$	Siêu Khóa	Khóa
\emptyset	\emptyset	\emptyset		
A	A	A		
B	B	B		
AB	AB	ABCD	AB	AB
C	C	ABCD	C	C
AC	AC	ABCD	AC	
BC	BC	ABCD	BC	
ABC	ABC	ABCD	ABC	
D	D	BD		
AD	AD	ABCD	AD	AD
BD	BD	BD		
ABD	ABD	ABCD	ABD	
CD	CD	ABCD	CD	
ACD	ACD	ABCD	ACD	
BCD	BCD	ABCD	BCD	
ABCD	ABCD	ABCD	ABCD	

$K_1 = AB, K_2 = AD, K_3 = C$ là các khóa của Q4, tất cả các thuộc tính đều là khóa, không có thuộc tính không khóa, nên Q4 đạt dạng chuẩn 3.

d. Dạng chuẩn BC (Boyce Codd)

- Định nghĩa

Lược đồ Q ở dạng chuẩn BC nếu mọi phụ thuộc hàm $X \rightarrow A \in F^+$, với $A \rightarrow X$ đều có X là siêu khóa.

Kiểm tra dạng chuẩn BC

Từ định nghĩa, để kiểm tra dạng chuẩn BC thực hiện các bước sau:

Bước 1: Tìm mọi khóa Q

Bước 2: Phân rã vế phải của mọi phụ thuộc hàm trong F để tập F trở thành tập phụ thuộc hàm có vế phải một thuộc tính

Bước 3: Nếu mọi phụ thuộc hàm $X \rightarrow A \in F$, mà $A \notin X$ đều thỏa X là siêu khóa (vế trái chứa một khóa), thì Q đạt dạng chuẩn BC, ngược lại Q không đạt dạng chuẩn BC

Ví dụ 6.9: Cho Q (A, B, C, D, E, I), và tập PTH $F = ACD \rightarrow EBI, CE \rightarrow AD$.
Xác định dạng chuẩn cho quan hệ Q?

Giải

Bước 1: Q có hai khóa là {ACD, CE}

Bước 2: Phân rã vế phải của các phụ thuộc hàm trong F, ta có:

$F = \{ACD \rightarrow E, ACD \rightarrow B, ACD \rightarrow I, CE \rightarrow A, CE \rightarrow D\}$

Bước 3: Mọi phụ thuộc hàm trong F đều có vế trái là một siêu khóa
Vậy Q đạt dạng chuẩn BC.

■ Nếu một lược đồ quan hệ không thỏa mãn điều kiện BCNF, thủ tục chuẩn hóa bao gồm:

- Loại bỏ các thuộc tính khóa phụ thuộc hàm vào thuộc tính không khóa ra khỏi quan hệ
- Tách chúng thành một quan hệ riêng có khoá chính là thuộc tính không khóa gây ra phụ thuộc.

Ví dụ 6.10: Cho R (A1, A2, A3, A4, A5)

Với các phụ thuộc hàm:

$F = \{A1, A2 \rightarrow A3, A4, A5; A4 \rightarrow A2\}$

- Lược đồ quan hệ R không thỏa dạng chuẩn BC vì tồn tại $A4 \rightarrow A2$ (vế trái của PTH không phải là siêu khóa)
- Lược đồ được tách ra như sau:
 - R1(A4, A2)
 - R2(A1, A4, A3, A5)

6.4 Kiểm tra dạng chuẩn

Kiểm tra dạng chuẩn của lược đồ quan hệ Q trải qua các bước sau:

- **Bước 1:** Tìm mọi khóa của Q
- **Bước 2:** Kiểm tra dạng chuẩn BC, nếu đúng thì Q đạt dạng chuẩn BC, ngược lại qua bước 3.
- **Bước 3:** Kiểm tra dạng chuẩn 3, nếu đúng thì Q đạt dạng chuẩn 3, ngược lại qua bước 4.
- **Bước 4:** Kiểm tra dạng chuẩn 2, nếu đúng thì Q đạt dạng chuẩn 2, ngược lại Q đạt dạng chuẩn 1.

► Câu hỏi (bài tập) củng cố:

Bài tập 1: Cho $Q1(A\ B\ C\ D\ E\ H)$ và tập phụ thuộc hàm

$F = \{A \rightarrow E, C \rightarrow D, E \rightarrow DH\}$. Tìm khóa, xác định dạng chuẩn của quan hệ $Q1$?

Bài tập 2: Cho $Q2(A\ B\ C\ D\ M)$ và tập PTH

$F = \{CD \rightarrow M, M \rightarrow AC, M \rightarrow B\}$.

Tìm khóa, xác định dạng chuẩn của quan hệ $Q2$?

Bài tập 3: Cho $Q3(A\ B\ C\ D\ E\ G\ H)$ và tập PTH

$F = \{A \rightarrow D; AB \rightarrow DE; CE \rightarrow G; E \rightarrow CH; H \rightarrow BE\}$. Tìm khóa, xác định dạng chuẩn của $Q3$?

Bài tập 4: Cho $Q4(A\ B\ C\ D)$ và tập PTH $F = \{AB \rightarrow C, C \rightarrow ABD\}$.

Tìm phủ tối thiểu của F ?

Chương 7

NGÔN NGỮ SQL

(Structured Query Language)

◆ **Mục tiêu học tập:** Sau khi học xong chương này, người học có thể:

- Sử dụng ngôn ngữ SQL để:
 - Tạo cấu trúc bảng, sửa và xóa cấu trúc bảng
 - Thêm, sửa, xóa dữ liệu
 - Cài đặt các ràng buộc toàn vẹn dữ liệu
 - Viết câu truy vấn cơ bản, truy vấn lồng
 - Sao lưu và phục hồi cơ sở dữ liệu

7.1 Giới thiệu

- Năm 1974, D.Chamberlin (IBM San Jose Lab) định nghĩa ngôn ngữ SEQUEL (Structured English Query Language).
- Một phiên bản sửa chữa, SEQUEL/2, đã được đưa ra vào năm 1976 nhưng sau đó đổi tên thành SQL vì lí do pháp lý.
- SQL (Structured Query Language) là ngôn ngữ CSDL quan hệ chuẩn dành cho các CSDL quan hệ thương mại.
- Từ những năm 70, SQL được phát triển từ IBM với hệ quản trị CSDL quan hệ **System R** với ngôn ngữ giao tiếp CSDL là Sequel (Structured English Query Language).

- Năm 1986, phiên bản chuẩn được chấp nhận bởi ANSI (American National Standards Institute) gọi là SQL-86 hay SQL1 và năm 1987 được chấp nhận bởi ISO (International Standards Organization). Từ đó các phiên bản đã được đưa ra vào 1992, 1999, 2003.

7.2 Ngôn ngữ SQL

- Là ngôn ngữ chuẩn để truy vấn và thao tác trên CSDL quan hệ
- Là một ngôn ngữ phi thủ tục với 2 thành phần chính:
 - a. Ngôn ngữ định nghĩa dữ liệu DDL (Data Definition Language): dùng để định nghĩa cấu trúc của CSDL. ANSI chia DDL gồm 2 phần:
 - + DDL gồm các lệnh để định nghĩa cấu trúc của CSDL:
 - o Lệnh tạo bảng (CREATE...)
 - o Lệnh sửa cấu trúc bảng (ALTER...)
 - o Lệnh xóa bảng (DROP...)
 - + DCL (Data Control Language) gồm các lệnh để điều khiển quyền truy cập trên dữ liệu:
 - o Lệnh cấp quyền cho người sử dụng cơ sở dữ liệu (GRANT...)
 - o Lệnh thu hồi quyền hạn của người sử dụng cơ sở dữ liệu (REVOKE...)
 - b. Ngôn ngữ thao tác dữ liệu (DML – Data Manipulation Language): Dùng để truy xuất và cập nhật dữ liệu, gồm các lệnh:
 - + Lệnh thêm dữ liệu (INSERT...)
 - + Lệnh sửa dữ liệu (UPDATE...)
 - + Lệnh xóa dữ liệu (DELETE...)
 - + Truy vấn dữ liệu (SELECT...)

Lưu ý: Hầu hết các thành phần trong câu lệnh SQL không phân biệt chữ hoa chữ thường, trừ các ký tự trong chuỗi dữ liệu. Tuy câu lệnh SQL có hình thức tự do nhưng để dễ đọc người sử dụng nên:

- Viết hoa cho từ khóa
- Từ do người dùng định nghĩa thì viết hoa ký tự đầu

- Viết mỗi mệnh đề của lệnh trên một dòng riêng
- Viết bắt đầu mệnh đề bằng hàng với các mệnh đề khác
- Nếu một mệnh đề có nhiều vế thì nên viết mỗi vế trên một dòng và thực vào trong mệnh đề đó.

Ví dụ 7.1: Câu lệnh sau đây hiển thị họ tên, công việc của các nhân viên nam làm việc tại phòng số 3 hoặc phòng số 4 trong bảng NHANVIEN có lược đồ như sau:

NHANVIEN (MaNV, HoTen, Phai, CongViec, Luong, PhuCap, MaPhong)

Hình thức câu lệnh viết như sau:

```
SELECT HoTen, CongViec
FROM NHANVIEN
WHERE Phai = 'nam'
AND (MaPhong = 3
OR MaPhong = 4)
```

- Trong các ví dụ sau chúng ta sử dụng **CSDL minh họa có tên là “QL_DEAN”** (CSDL quản lý đề án) gồm có các bảng được đặc tả như sau:

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

Tên từ: Mỗi nhân viên có Mã nhân viên duy nhất để phân biệt với các nhân viên khác, có họ tên, ngày sinh, địa chỉ, phái Nam hoặc Nữ, mức lương, người quản lý trực tiếp và thuộc về một phòng ban

PHONGBAN (MaPB, TenPB, Trg_PB, NgayNC)

Tên từ: Mỗi một phòng ban có một mã phòng duy nhất để phân biệt với các phòng ban khác, có tên phòng, người trưởng phòng, và ngày nhận chức của trưởng phòng.

DIADIEMPHONG (MaPB, DiaDiem)

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

DEAN (MaDA, TenDA, DdiemDA, Phg, NgayBD, NgayKT)

Tên từ: Mỗi một đề án có một mã đề án duy nhất để phân biệt với các đề án khác, có tên đề án, địa điểm thực hiện, và do một phòng ban chủ trì đề án đó

PHANCONG (MaNV, MaDA, ThoiGian)

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

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

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

NV_TNHAN(MaNV, MaTN, QuanHe)

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

Thể hiện CSDL như sau:

NHANVIEN									
MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	Dchi	MaNQL	Phg	Luong
002	Nguyen	Thanh	Tung	9/1/1955	Nam	TPHCM	001	NC	2500000
003	Le	Thi	Nhan	18/12/1960	Nu	TPHCM	001	DH	2500000
004	Dinh	Ba	Tien	9/1/1968	Nam	Dong Thap	002	NC	2200000
005	Bui	Thuy	Vu	8/6/1973	Nu	Tra Vinh	003	DH	2200000
006	Nguyen	Manh	Hung	30/8/1981	Nam	Dong Nai	002	NC	2000000
007	Tran	Hong	Van	2/8/1976	Nu	Vinh Long	004	NC	1800000

DEAN					
MaDA	TenDA	DdiemDA	Phg	NgayBD	NgayKT
TH00	Tin hoc hoa 1	Ha Noi	NC	1/2/2003	1/2/2004
TH002	Tin hoc hoa 2	TPHCM	NC	4/6/2003	4/2/2004
TH003	Tin hoc hoa 3	Da Lat	QL	1/12/2006	1/12/2006
DT001	Dao tao 1	Nha Trang	DH	4/2/2002	4/2/2006
DT002	Dao tao 2	Da Nang	DH	1/2/2006	2/12/2006

PHONGBAN			
MaPB	TenPB	TrPHG	NgayNC
QL	Quan Ly	001	22/5/2000
DH	Dieu Hanh	003	10/10/2002
NC	Nghien Cuu	002	15/3/2002

THANNHAN			
MaTN	TenTN	Phai	NgSinh
1	Tran Minh Tien	Nam	11/12/1990
2	Tran Ngoc Linh	Nu	10/3/1993
3	Tran Minh Long	Nam	10/10/1957
4	Le Nhat Minh	Nam	27/4/1955
5	Le Hoai Thuong	Nu	5/12/1960
6	Le Phi Phung	Nu	24/12/1960

PHANCONG		
MaNV	MaDA	ThoiGian
001	TH001	30
001	TH002	12.5
002	TH001	10
002	TH002	10
002	DT001	10
002	DT002	10
003	TH001	37.5
004	DT001	22.5
004	DT002	10
006	DT001	30.5
007	TH001	20
007	TH002	10

NV_THANHAN		
MaNV	MaTN	QuanHe
003	1	Con
003	2	Con
003	3	Vo Chong
001	1	Vo Chong
002	1	Vo Chong
004	1	Vo Chong
005	1	Vo Chong

Bảng 7.1: Cơ sở dữ liệu minh họa ‘Quản lý đề án’

7.2.1 Lệnh tạo và xóa cơ sở dữ liệu

7.2.1.1 Để tạo một CSDL người ta dùng câu lệnh

- Cú pháp:

```
CREATE DATABASE Database_Name
```

- Diễn giải: *<Database_Name>* là tên CSDL được tạo

Ví dụ 7.2: Tạo CSDL tên QL_DEAN

```
CREATE DATABASE QL_DEAN
```

7.2.1.2 Để sử dụng CSDL người ta dùng câu lệnh

- Cú pháp:

```
USE Database_Name
```

- Diễn giải: *<Database_Name>* là tên CSDL được sử dụng

Ví dụ 7.2: Sử dụng CSDL tên QL_DEAN

```
USE QL_DEAN
```

7.2.1.3 Để hủy bỏ một CSDL người ta dùng câu lệnh

- Cú pháp:

```
DROP DATABASE Database_Name
```

- Diễn giải: *<Database_Name>* là tên CSDL bị xóa

Ví dụ 7.2: Hủy CSDL tên QL_DEAN

```
DROP DATABASE QL_DEAN
```

7.2.2 Ngôn ngữ định nghĩa dữ liệu DDL

7.2.2.1 Lệnh tạo cấu trúc bảng

- Cú pháp: Tạo bảng không có RBTV

```
CREATE TABLE tên_bảng
(
    Tên_thuộc_tính 1 Kiểu dữ liệu <giá trị khác>,
    Tên_thuộc_tính 2 Kiểu dữ liệu <giá trị khác>,
    Tên_thuộc_tính 3 Kiểu dữ liệu <giá trị khác>,
    ...
)
```

- Diễn giải:

<giá trị khác>: là từ khóa NULL hoặc NOT NULL

Ví dụ 7.4: Tạo bảng NHANVIEN không có RBTV

```
CREATE TABLE tên_bảng
(
    MaNV varchar(10) NOT NULL,
    HoNV varchar(50) NOT NULL,
    TenLot varchar(50) NOT NULL,
    TenNV varchar(50) NOT NULL,
    NgSinh datetime,
    Phai varchar(3) NOTNULL,
    DChi varchar(100),
    MaNQL varchar(10),
    Phg varchar(10),
    Luong float
)
```

- Sau khi thực thi câu lệnh trên sẽ trả về kết quả như hình bên dưới

MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	Phg	Luong
------	------	--------	-------	--------	------	------	-------	-----	-------

Hình 7.1: Cấu trúc bảng NHANVIEN không cài đặt RBTV

- **Cú pháp:** Tạo bảng có RBTV khóa chính và khóa ngoại

```
CREATE TABLE tên_bảng
(
    Tên_thuộc_tính 1 Kiểu dữ liệu <giá trị khác>,
    Tên_thuộc_tính 2 Kiểu dữ liệu <giá trị khác>,
    Tên_thuộc_tính 3 Kiểu dữ liệu <giá trị khác>,
    ...
    CONSTRAINT [tên khóa chính] PRIMARY KEY ([danh sách các thuộc tính làm khóa chính, cách nhau bằng dấu phẩy])
    CONSTRAINT [Tên Khóa Ngoại] FOREIGN KEY ([Danh sách các thuộc tính khóa ngoại]) REFERENCES [Tên bảng tham chiếu] ([Danh sách các thuộc tính khóa chính của bảng tham chiếu đến])
)
```

Ví dụ 7.5: Tạo bảng NHANVIEN có RBTV khóa chính và khóa ngoại

```
CREATE TABLE NHANVIEN
(
    MaNV varchar(10) NOT NULL,
    HoNV varchar(50) NOT NULL,
    TenLot varchar(50) NOT NULL,
    TenNV varchar(50) NOT NULL,
    NgSinh datetime,
    Phai varchar(3) NOTNULL,
    DChi varchar(100),
    MaNQL varchar(10),
    Phg varchar(10),
    Luong float
    CONSTRAINT PK_NV PRIMARY KEY (MANV),
    CONSTRAINT FK_NV_PB FOREIGN KEY PHONG REFERENCES PHONG-
    BAN (MAPHG)
)
```

- Sau khi thực thi câu lệnh trên sẽ trả về kết quả như hình bên dưới

MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	Phg	Luong
------	------	--------	-------	--------	------	------	-------	-----	-------

Hình 7.2: Cấu trúc bảng NHANVIEN sau khi thực thi câu lệnh

- **Lưu ý:** Giá trị NULL cho phép giá trị thuộc tính có thể rỗng, NOT NULL bắt buộc giá trị của thuộc tính không được phép rỗng. Đối với thuộc tính làm khóa chính thì giá trị không được rỗng.

◆ Các kiểu dữ liệu thường gặp

TÊN KIỂU	GIẢI THÍCH
Bit	Số nguyên 0 hoặc 1
Int	Số nguyên từ -2^{31} đến $2^{31} - 1$ tức là từ (-2.147.483.648) đến (2.147.483.647)
SmaillInt	Số nguyên từ -2^{15} đến $2^{15} - 1$ tức là từ (-32.768) đến (32.767)
TinyInt	Số nguyên từ 0 đến 255
Decimal/ numeric	Các số thập phân từ $-10^{38} - 1$ đến $10^{38} - 1$
Money	Tiền tệ từ -2^{63} đến $2^{63} - 1$
SmallMoney	Tiền tệ từ -214.748,3648 đến 214.748,3647
Float	Số thực từ -1,79E+308 đến 1,79E+308
Real	Số thực từ -3.40E+38 đến 3.40E+38
DateTime	Ngày giờ từ 1/1/1753 đến 31/10/9999
SmallDateTime	Ngày giờ từ 1/1/1900 đến 06/6/2079
Char	Ký tự có độ dài ô nhớ cố định và tối đa là 8000 ký tự (không hỗ trợ Unicode)
Nchar	Ký tự có độ dài ô nhớ cố định và tối đa là 8000 ký tự (hỗ trợ Unicode)
Varchar	Ký tự có độ dài ô nhớ không cố định và tối đa là 8000 ký tự (không hỗ trợ Unicode)
Nvarchar	Ký tự có độ dài ô nhớ không cố định và tối đa là 8000 ký tự (hỗ trợ Unicode)

Bảng 7.2: Các kiểu dữ liệu thường gặp trong các hệ QT CSDL

7.2.2.2 Lệnh sửa cấu trúc bảng

- **Cú pháp thêm thuộc tính:**

```
ALTER TABLE tên_bảng ADD Tên_cột kiểu_dữ_liệu
```

Ví dụ 7.5: Thêm cột Ghi_chú vào bảng nhân viên

```
ALTER TABLE NHANVIEN ADD Ghi_chu varchar(20)
```

- Sau khi thực thi câu lệnh thêm cột Ghi_Chỉ sẽ trả về kết quả như hình bên dưới

MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	Phg	Luong	Ghi Chu
------	------	--------	-------	--------	------	------	-------	-----	-------	---------

Hình 7.3: Câu lệnh thêm thuộc tính Ghi_Chú vào bảng NHANVIEN

```
ALTER TABLE tên_bảng ALTER COLUMN Tên_cột kiểu dữ liệu mới
```

Ví dụ 7.6: Sửa kiểu dữ liệu của cột Ngày sinh

```
ALTER TABLE NHANVIEN ALTER COLUMN NgaySinh  
SMALLDATETIME
```

- Cú pháp xóa thuộc tính:

```
ALTER TABLE tên_bảng DROP COLUMN Tên_cột
```

Ví dụ 7.7: Xóa cột Ghi_chú từ bảng nhân viên

```
ALTER TABLE NHANVIEN DROP COLUMN Ghi_chu
```

- Sau khi thực thi câu lệnh xóa thuộc tính Ghi_Chú sẽ trả về kết quả như hình bên dưới

MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	Phg	Luong
------	------	--------	-------	--------	------	------	-------	-----	-------

Hình 7.4: Xóa thuộc tính Ghi_Chú ra khỏi bảng NHANVIEN

- Cú pháp thêm ràng buộc toàn vẹn:

```
ALTER TABLE <tên_bảng> ADD CONSTRAINT <tên_ràng_buộc>  
<loại ràng buộc>
```

- Diễn giải: <loại ràng buộc> có các loại RBTV sau
- + **[PRIMARY KEY]** - là ràng buộc ấn định toàn vẹn thực thể cho một hoặc nhiều cột thông qua một chỉ mục duy nhất. Mỗi bảng chỉ có một ràng buộc PRIMARY KEY, tức là ràng buộc khóa chính.
 - + **[UNIQUE]** - là ràng buộc ấn định toàn vẹn thực thể cho một hoặc nhiều cột thông qua chỉ mục duy nhất. Một bảng có thể có nhiều ràng buộc UNIQUE (khóa ứng viên).
 - + **DEFAULT** - đặt tả giá trị được cung cấp cho cột khi giá trị không được đáp ứng tường minh trong quá trình xen dòng.

- + **CHECK** - Là ràng buộc ấn định toàn vẹn miền giá trị bằng cách giới hạn các giá trị có thể đưa vào một cột.
- + **FOREIGN KEY** *tên_cột* **REFERENCES** *tên_bảng_tham_chiếu* (*tên_cột_làm_khóa_chính*) - Là ràng buộc ấn định toàn vẹn tham chiếu. Trong phần sau sẽ nói thêm một vài tùy chọn có liên quan.

Ví dụ 7.8

- + Thêm RBTV kiểm tra miền giá trị của thuộc tính Phai trong bảng NHANVIEN

```
ALTER TABLE NHAN_VIEN ADD CONSTRAINT CHK
CHECK (PHAI IN ('Nam','Nu'))
```

- + Thêm ràng buộc toàn vẹn khóa chính vào bảng NHANVIEN

```
ALTER TABLE NHAN_VIEN ADD CONSTRAINT PK_NV
PRIMARY KEY(MaNV)
```

- + Thêm ràng buộc toàn vẹn khóa chính vào bảng PHANCONG

```
ALTER TABLE PHANCONG ADD CONSTRAINT PK_PC
PRIMARY KEY(MaNV, MaDA)
```

- + Thêm ràng buộc toàn vẹn khóa ngoại vào bảng PHANCONG

```
ALTER TABLE PHANCONG ADD CONSTRAINT FK_PC_NV
FOREIGN KEY(MaNV) REFERENCES NHANVIEN(MaNV)
ALTER TABLE PHANCONG ADD CONSTRAINT FK_PC_DA
FOREIGN KEY(MaDA) REFERENCES DEAN(MaDA)
```

Cú pháp xóa ràng buộc toàn vẹn :

```
ALTER TABLE tên_bảng DROP CONSTRAINT tên_ràng_buộc
```

Ví dụ 7.9: Xóa RBTV *FK_NV_PB* của bảng NHANVIEN

```
ALTER TABLE NHANVIEN DROP CONSTRAINT FK_NV_PB
```

7.2.2.3 Lệnh xóa cấu trúc bảng

- Cú pháp:

```
DROP TABLE tên_bảng
```

Ví dụ 7.10: Xóa bảng (table) nhân viên

```
DROP TABLE NHANVIEN
```

7.2.3 Ngôn ngữ thao tác dữ liệu

7.2.3.1 Thêm dữ liệu vào bảng

■ Cú pháp: có 2 cách để thêm dữ liệu vào bảng

Cách 1: `INSERT INTO tên_bảng VALUES(giá_trị_1, giá_trị_2, ..., giá_trị_n)`

Cách 2: `INSERT INTO tên_bảng(cột1, cột2, ...) VALUES(giá_trị_1, giá_trị_2, ...)`

Ví dụ 7.11: Thêm nhân viên mới cùng với thông tin của nhân viên này

Cách 1:

```
INSERT INTO NHANVIEN  
VALUES('001','VUONG','NGOC','QUYEN','10/10/1957','nu','HA  
NOI','','QL','3000000')
```

Cách 2:

```
INSERT INTO  
NHANVIEN(MaNV,HoNV,TenNV,NgSinh,DChi,Phai,MaNQL,Phg,Luong)  
VALUES('001','VUONG','NGOC','QUYEN','10/10/1957','nu','HA  
NOI','','QL','3000000')
```

- *Dữ liệu bảng NHANVIEN trước khi thêm vào*

	MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	PHG	LUONG
1	002	NGUYEN	THANH	TUNG	1955-09-01 00:00:00.000	nam	TPHCM	001	NC	2500000.0
2	003	LE	THI	NHAN	1960-10-12 00:00:00.000	nu	TPHCM	001	DH	2500000.0
3	004	DINH	BA	TIEN	1968-09-01 00:00:00.000	nam	DONG THAP	002	NC	2200000.0
4	005	BUI	THUY	VU	1973-08-06 00:00:00.000	nu	TRAVINH	003	DH	2200000.0
5	006	NGUYEN	MANH	HUNG	1981-10-08 00:00:00.000	nam	DONG NAI	002	NC	2000000.0
6	007	TRAN	HONG	VAN	1976-02-08 00:00:00.000	nu	VINH LONG	004	NC	1800000.0

- Dữ liệu bảng NHANVIEN sau khi thực thi câu lệnh thêm nhân viên

	MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	PHG	LUONG
1	001	VUONG	NGOC	QUYEN	1957-10-10 00:00:00.000	nu	HA NOI		QL	3000000.0
2	002	NGUYEN	THANH	TUNG	1955-09-01 00:00:00.000	nam	TPHCM	001	NC	2500000.0
3	003	LE	THI	NHAN	1960-10-12 00:00:00.000	nu	TPHCM	001	DH	2500000.0
4	004	DINH	BA	TIEN	1968-09-01 00:00:00.000	nam	DONG THAP	002	NC	2200000.0
5	005	BUI	THUY	VU	1973-08-06 00:00:00.000	nu	TRAVINH	003	DH	2200000.0
6	006	NGUYEN	MANH	HUNG	1981-10-08 00:00:00.000	nam	DONG NAI	002	NC	2000000.0
7	007	TRAN	HONG	VAN	1976-02-08 00:00:00.000	nu	VINH LONG	004	NC	1800000.0

Hình 7.5: Thêm dữ liệu vào bảng NHANVIEN

7.2.3.2 Để sửa đổi dữ liệu của bảng ta sử dụng lệnh sau

- Cú pháp:

```
UPDATE tên_bảng SET cột_1 = giá_trị_1, cột_2 = giá_trị_2 [WHERE điều_kiện]
```

Ví dụ 7.12: Sửa đổi dữ liệu trong bảng

■ Sửa họ nhân viên có mã số '003' thành 'Nguyen'

```
UPDATE NHANVIEN SET HONV='Nguyen' WHERE MANV='003'
```

- Dữ liệu bảng NHANVIEN sau khi sửa thông tin nhân viên 003

	MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	PHG	LUONG
1	001	VUONG	NGOC	QUYEN	1957-10-10 00:00:00.000	nu	HA NOI		QL	3000000.0
2	002	NGUYEN	THANH	TUNG	1955-09-01 00:00:00.000	nam	TPHCM	001	NC	2500000.0
3	003	Nguyen	THI	NHAN	1960-10-12 00:00:00.000	nu	TPHCM	001	DH	2500000.0
4	004	DINH	BA	TIEN	1968-09-01 00:00:00.000	nam	DONG THAP	002	NC	2200000.0
5	005	BUI	THUY	VU	1973-08-06 00:00:00.000	nu	TRAVINH	003	DH	2200000.0
6	006	NGUYEN	MANH	HUNG	1981-10-08 00:00:00.000	nam	DONG NAI	002	NC	2000000.0
7	007	TRAN	HONG	VAN	1976-02-08 00:00:00.000	nu	VINH LONG	004	NC	1800000.0

Hình 7.6: Sửa dữ liệu của nhân viên 003 trong bảng NHANVIEN

- Sửa họ tên của nhân viên có mã số '001' thành 'Nguyen Thanh Tung' và ngày sinh mới là 1/1/1978

```
UPDATE NHANVIEN SET HONV = 'Nguyen', TENLOT = 'Bao', TENNV = 'Tran', NgSinh='1/1/1978' WHERE MANV='001'
```

	MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	PHG	LUONG
1	001	Nguyen Bao	Tran	1978-01-01	00:00:00.000	nu	HA NOI		QL	3000000.0
2	002	NGUYEN THANH	TUNG	1955-09-01	00:00:00.000	nam	TPHCM	001	NC	2500000.0
3	003	Nguyen THI	NHAN	1960-10-12	00:00:00.000	nu	TPHCM	001	DH	2500000.0
4	004	DINH BA	TIEN	1968-09-01	00:00:00.000	nam	DONG THAP	002	NC	2200000.0
5	005	BUI THUY	VU	1973-08-06	00:00:00.000	nu	TRAVINH	003	DH	2200000.0
6	006	NGUYEN MANH	HUNG	1981-10-08	00:00:00.000	nam	DONG NAI	002	NC	2000000.0
7	007	TRAN HONG	VAN	1976-02-08	00:00:00.000	nu	VINH LONG	004	NC	1800000.0

Hình 7.7: Sửa dữ liệu trong bảng NHANVIEN

7.2.3.3 Xóa dữ liệu trong bảng

- Cú pháp:

```
DELETE FROM tên_bảng [WHERE điều_kiện]
```

Ví dụ 7.13:

- Xóa nhân viên có mã số '005'

```
DELETE FROM NHANVIEN
WHERE MANV = '005'
```

	MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	PHG	LUONG
1	001	Nguyen Bao	Tran	1978-01-01	00:00:00.000	nu	HA NOI		QL	3000000.0
2	002	NGUYEN THANH	TUNG	1955-09-01	00:00:00.000	nam	TPHCM	001	NC	2500000.0
3	003	Nguyen THI	NHAN	1960-10-12	00:00:00.000	nu	TPHCM	001	DH	2500000.0
4	004	DINH BA	TIEN	1968-09-01	00:00:00.000	nam	DONG THAP	002	NC	2200000.0
5	006	NGUYEN MANH	HUNG	1981-10-08	00:00:00.000	nam	DONG NAI	002	NC	2000000.0
6	007	TRAN HONG	VAN	1976-02-08	00:00:00.000	nu	VINH LONG	004	NC	1800000.0

Hình 7.8: Xóa nhân viên 005 trong bảng NHANVIEN

- Xóa toàn bộ nhân viên

```
DELETE FROM NHANVIEN
```

7.2.3.4 Truy vấn dữ liệu

a. Truy vấn cơ bản

- Cú pháp lệnh truy vấn có dạng:

```
SELECT <danh sách thuộc tính>  
FROM <danh sách các bảng>  
WHERE <điều kiện>  
GROUP BY <các thuộc tính gom nhóm>  
HAVING <điều kiện gom nhóm>  
ORDER BY <danh sách thuộc tính>
```

Với:

- ▶ **SELECT** <danh sách thuộc tính>: tên các thuộc tính được lấy giá trị
- ▶ **FROM** <danh sách các bảng>: tên các bảng cần để xử lý câu truy vấn
- ▶ **WHERE** <điều kiện>: biểu thức điều kiện chọn và điều kiện kết các bộ trong các quan hệ được chỉ ra trong mệnh đề **FROM**
- ▶ **GROUP BY** <các thuộc tính gom nhóm>: chỉ ra các thuộc tính gom nhóm
- ▶ **HAVING** <điều kiện gom nhóm>: chỉ ra điều kiện để trích chọn trên các nhóm
- ▶ **ORDER BY** <danh sách thuộc tính>: thứ tự hiển thị kết quả câu truy vấn
 - **ASC** tăng dần
 - **DESC** giảm dần
 - Nếu không ghi thứ tự hiển thị thì mặc định là tăng dần.
- Trong câu lệnh, **SELECT** và **FROM** là bắt buộc, những thành phần còn lại có thể không có.
- Từ khóa **DISTINCT** trong **SELECT** chỉ rõ rằng chỉ có một trong các dòng giống nhau xuất hiện trong kết quả truy vấn. Dấu (*) chỉ rõ rằng truy vấn tất cả các cột và thể hiện trong kết quả truy vấn.
- Khi câu truy vấn cần tham chiếu tới cùng một quan hệ 2 lần thì dùng bí danh cho tên quan hệ. SQL cho phép đổi tên quan hệ và tên thuộc tính được thực hiện bằng **AS** (có thể dùng hoặc không) theo quy tắc: Tên cũ [as] tên mới.

- Tên thuộc tính có thể kèm theo tên bảng nếu cần làm rõ bằng cách thêm dấu chấm (.) trước tên thuộc tính. Ví dụ: NHANVIEN.MaNV.
- Câu lệnh SELECT còn cho phép thực hiện tính toán theo công thức dựa trên các cột của bảng.

Ví dụ 7.14: Tìm mã nhân viên, họ tên và lương nhân viên, 20% cho mọi nhân viên

```
SELECT MaNV, HoNV, TenNV, Luong*1.2
FROM NHANVIEN
```

Ví dụ 7.15: Truy vấn dữ liệu trong mệnh đề SELECT và FROM

- Hiển thị toàn bộ thông tin của các nhân viên

```
SELECT*
FROM NHANVIEN
```

	MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	PHG	LUONG
1	001	Nguyen	Bao	Tran	1978-01-01 00:00:00.000	nu	HA NOI		QL	3000000.0
2	002	NGUYEN	THANH	TUNG	1955-09-01 00:00:00.000	nam	TPHCM	001	NC	2500000.0
3	003	Nguyen	THI	NHAN	1960-10-12 00:00:00.000	nu	TPHCM	001	DH	2500000.0
4	004	DINH	BA	TIEN	1968-09-01 00:00:00.000	nam	DONG THAP	002	NC	2200000.0
5	005	BUI	THUY	VU	1973-08-06 00:00:00.000	nu	TRAVINH	003	DH	2200000.0
6	006	NGUYEN	MANH	HUNG	1981-10-08 00:00:00.000	nam	DONG NAI	002	NC	2000000.0
7	007	TRAN	HONG	VAN	1976-02-08 00:00:00.000	nu	VINH LONG	004	NC	1800000.0

Hình 7.9: Hiển thị toàn bộ thông tin của các nhân viên

- Hiển thị danh sách gồm họ tên nhân viên và phái của nhân viên đó

```
SELECT HoNV, TenNV, Phai
FROM NHANVIEN
```

- Sau khi thực hiện câu lệnh ta có kết quả như hình bên dưới

	honv	tenlot	tennv	phai
1	Nguyen	Bao	Tran	nu
2	NGUYEN	THANH	TUNG	nam
3	Nguyen	THI	NHAN	nu
4	DINH	BA	TIEN	nam
5	NGUYEN	MANH	HUNG	nam
6	TRAN	HONG	VAN	nu

Hình 7.10: Chiếu lấy họ tên và phái của nhân viên

- Hiển thị danh sách các mức lương có thể có của các nhân viên. Trong trường hợp này cần loại bỏ các mức lương giống nhau.

```
SELECT Luong
FROM NHANVIEN
```

- Sau khi thực hiện câu lệnh ta có kết quả như hình bên dưới

	Luong
1	1800000.0
2	2000000.0
3	2200000.0
4	2500000.0
5	3000000.0

Hình 7.11: Cho biết các mức lương trùng nhau bị bỏ bớt

Ví dụ 7.16: Hiển thị danh sách nhân viên theo mã phòng tăng dần, và lương giảm dần

```
SELECT*
FROM NHANVIEN
ORDER BY Phg, Luong DESC
```

- Sau khi thực hiện câu lệnh ta có kết quả như hình bên dưới

	MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DCh1	MaNQL	Phg	LUONG
1	003	Nguyen	THI	NGAN	1960-10-12 00:00:00.000	nu	TPHCM	001	DH	2500000.0
2	002	NGUYEN	THANH	TUNG	1955-09-01 00:00:00.000	nam	TPHCM	001	HC	2500000.0
3	004	DINH	BA	TIEN	1968-09-01 00:00:00.000	nam	DONG TRAP	002	HC	2200000.0
4	006	NGUYEN	MANH	HUNG	1981-10-08 00:00:00.000	nam	DONG HAI	002	HC	2000000.0
5	007	TRAN	HONG	VAN	1974-02-08 00:00:00.000	nu	VINH LONG	004	HC	1800000.0
6	001	Nguyen	Bao	Tran	1978-01-01 00:00:00.000	nu	HA NOI		QL	3000000.0

Hình 7.12: Sắp xếp dữ liệu trong bảng

- Trong nhiều trường hợp người ta chỉ muốn truy vấn một số dòng thỏa một điều kiện nào đó trong bảng. Người ta sử dụng mệnh đề WHERE sau mệnh đề FROM để chọn các dòng thỏa điều kiện.

► **Mệnh đề FROM** tên các bảng cần để xử lý câu truy vấn, nếu có từ 2 bảng trở lên phải thực hiện kết nối các bảng lại với nhau.

◆ Các phép nối

- Phép nối nội

Một phép nối nội (inner join) thực chất là tìm giao của hai bảng dữ liệu. Đây là loại kết hợp thường dùng nhất và được xem là phép kết mặc định.

Ví dụ 7.17: Cho 2 bảng NHANVIEN và PHONGBAN, hãy cho biết thông tin về phòng ban mà nhân viên đó làm việc.

NHANVIEN	
TenNV	MaPB
Nam	NC
Hùng	NC
Mai	QL
Thái	DH
Hà	TC

PHONGBAN	
MAPB	TenPB
NC	Nghiên Cứu
DH	Điều Hành
QL	Quản lý
TV	Tài Vụ

- Sử dụng cú pháp kết nối nội

o Cách viết thứ nhất

```
SELECT*  
FROM NHANVIEN.MaPB = PHONGBAN.MaPB
```

o Cách viết thứ hai

```
SELECT*  
FROM NHANVIEN INNER JOIN PHONGBAN  
ON NHANVIEN.MaPB = PHONGBAN.MaPB
```

- Kết quả kết hợp:

TenNV	MaPB	MaPB	TenPB
Nam	NC	NC	Nghiên Cứu
Hùng	NC	NC	Nghiên Cứu
Mai	QL	QL	Quản lý
Thái	DH	DH	Điều Hành

Lưu ý: Khi sử dụng kết nối nội thì kết quả sẽ loại bỏ những nhân viên có phòng ban không tồn tại trong bảng PHONGBAN

- **Phép nối ngoại bên trái (Left Outer Join):** Trả về tất cả những giá trị từ bảng bên trái + những giá trị tương ứng với bảng bên phải hoặc là NULL (khi những giá trị ở bảng bên phải không tương ứng).

Chẳng hạn, nó cho phép ta tìm phòng ban của nhân viên, nhưng vẫn trả về nhân viên ngay cả khi phòng ban của người đó là NULL hoặc không tồn tại.

Ví dụ phép kết ngoại bên trái

```
SELECT distinct *  
FROM employee LEFT OUTER JOIN department  
ON employee.DepartmentID = department.DepartmentID
```

Kết quả kết hợp

TenNV	MaPB	MaPB	TenPB
Nam	NC	NC	Nghiên Cứu
Hùng	NC	NC	Nghiên Cứu
Mai	QL	QL	Quản lý
Thái	DH	DH	Điều Hành
Hà	TC	NULL	NULL

- **Phép kết nối ngoại bên phải:** Trả về tất cả những giá trị từ bảng bên phải + những giá trị tương ứng với bảng bên trái hoặc là NULL (khi những giá trị ở bảng bên trái không tương ứng).

Ví dụ phép kết ngoại bên phải

```
SELECT distinct *  
FROM employee RIGHT OUTER JOIN department  
ON employee.DepartmentID = department.DepartmentID
```

Kết quả kết hợp

TenNV	MaPB	MaPB	TenPB
Nam	NC	NC	Nghiên Cứu
Hùng	NC	NC	Nghiên Cứu
Mai	QL	QL	Quản lý
Thái	DH	DH	Điều Hành
NULL	NULL	TV	Tài Vụ

► **Mệnh đề WHERE** có các dạng điều kiện như sau:

Dạng 1: So sánh giá trị thuộc tính với 1 giá trị cụ thể

<cột> so sánh <giá trị>

Ví dụ 7.18:

- Cho biết thông tin của nhân viên “Nguyễn Thanh Tùng”

```
SELECT *
FROM nhanvien
WHERE HoNV = 'Nguyen'
AND TenLot = 'Thanh'
AND TenNV = 'Tung'
```

- Sau khi thực thi câu lệnh xem thông tin nhân viên, kết quả trả về như hình bên dưới

	MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	PHG	LUONG
1	002	NGUYEN	THANH	TUNG	1955-09-01 00:00:00.000	nam	TPHCM	001	NC	2500000.0

Hình 7.13: Xem tất cả thông tin của nhân viên theo điều kiện cụ thể

- Cho biết tên phòng ban mà nhân viên “Nguyễn Thanh Tùng” đang làm việc.

```
SELECT TenPB
FROM PhongBan p,NhanVien n
WHERE p.MaPB = n.Phg
AND HoNV= 'Nguyen'
AND TenLot = 'Thanh'
AND TenNV = 'Tung'
```

- Sau khi thực thi câu lệnh xem thông tin nhân viên, kết quả trả về như hình bên dưới

	TenPb
1	NGHIEN CUU

Hình 7.14: Xem thông tin của nhân viên có sự liên kết nhiều bảng

Dạng 2: So sánh giá trị các thuộc tính với nhau

<cột> so sánh <cột>

Ví dụ 7.19: Hãy cho biết tên các đề án dự kiến sẽ kết thúc trong cùng một ngày

```
SELECT TenDA
FROM DEAN
WHERE NgayBD = NgayKT
```

- Dữ liệu của bảng DEAN
- Sau khi thực thi câu lệnh xem thông tin nhân viên, kết quả trả về như hình bên dưới

	MaDA	TenDA	DdiemDA	Phg	NgayYBD	NgayKT
1	DT001	DAO TAO 1	NHA TRANG	DH	2002-04-02...	2006-04-02...
2	DT002	DAO TAO 2	DA NANG	DH	2006-01-02...	2006-12-12...
3	TH001	TIN HOC HOA 1	HA NOI	NC	2003-01-02...	2004-01-02...
4	TH002	TIN HOC HOA 2	TPHCM	NC	2003-04-02...	2004-04-02...
5	TH003	TIN HOC HOA 3	DA LAT	QL	2006-01-12...	2006-01-12...

	TenDA
1	TIN HOC HOA 3

Hình 7.15: Kết quả so sánh giá trị thuộc tính

Dạng 3: So sánh cột với một tập gồm nhiều giá trị:

<cột> IN (<danh sách các giá trị>)
<cột> NOT IN (<danh sách các giá trị>)

Ví dụ 7.20: Cho biết thông tin những nhân viên làm việc tại các phòng có mã là DH, NC, TC

```
SELECT *
FROM NhanVien
WHERE Phong IN ('DH', 'NC', 'TC')
```

- Sau khi thực hiện câu lệnh, kết quả trả về như hình bên dưới

	MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	PHG	LUONG	
1	003	Nguyen	THI	NHAN	1960-10-12 00:00:00.000	nu	TPHCM	001	DH	2500000.0	
2	002	NGUYEN	THANH	TUNG	1955-09-01 00:00:00.000	nam	TPHCM	001	NC	2500000.0	
3	004	DINH	BA	TIEN	1968-09-01 00:00:00.000	nam	DONG THAP	002	NC	2200000.0	
4	006	NGUYEN	MANH	HUNG	1981-10-08 00:00:00.000	nam	DONG NAI	002	NC	2000000.0	
5	007	TRAN	HONG	VAN	1976-02-08 00:00:00.000	nu	VINH LONG	004	NC	1800000.0	
6	001	Nguyen	Bao	Tran	1978-01-01 00:00:00.000	nu	HA NOI		QL	3000000.0	

Hình 7.16: Kết quả so sánh giá trị thuộc tính với tập giá trị

Dạng 4: Kiểm tra điều kiện theo phạm vi:

<cột> BETWEEN <giá trị đầu> AND <giá trị cuối>
 < cột > NOT BETWEEN < giá trị đầu > AND < giá trị cuối >

Ví dụ 7.21: Cho biết những nhân viên có năm sinh trong khoảng từ năm 1964 đến 1981

```
SELECT *
FROM NhanVien
WHERE year(NgaySinh) BETWEEN 1964 AND 1981
```

- Thực thi câu lệnh ta có kết quả

	MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	PHG	LUONG	
1	001	Nguyen	Bao	Tran	1978-01-01 00:00:00.000	nu	HA NOI		QL	3000000.0	
2	004	DINH	BA	TIEN	1968-09-01 00:00:00.000	nam	DONG THAP	002	NC	2200000.0	
3	006	NGUYEN	MANH	HUNG	1981-10-08 00:00:00.000	nam	DONG NAI	002	NC	2000000.0	
4	007	TRAN	HONG	VAN	1976-02-08 00:00:00.000	nu	VINH LONG	004	NC	1800000.0	

Hình 7.17: Kết quả so sánh giá trị thuộc tính thuộc phạm vi

Dạng 5: Kiểm tra điều kiện theo mẫu dạng chuỗi tương đương:

<cột> LIKE <chuỗi>
 <cột> NOT LIKE <chuỗi>

Ví dụ 7.22: Cho biết thông tin của những nhân viên có ký tự thứ 2 của tên là u?

```
SELECT *
FROM NHANVIEN
WHERE hoten LIKE '_u%'
```

- Thực thi câu lệnh ta có kết quả

	MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	PHG	LUONG	
1	002	NGUYEN	THANH	TUNG	1955-09-01 00:00:00.000	nam	TPHCM	001	NC	2500000.0	
2	006	NGUYEN	MANH	HUNG	1981-10-08 00:00:00.000	nam	DONG NAI	002	NC	2000000.0	

Hình 7.18: Tìm kiếm thông tin dạng chuỗi tương đương

Dạng 6: Kiểm tra điều kiện theo giá trị NULL

<Cột> IS NULL
<Cột> IS NOT NULL

Ví dụ 7.23: Hiển thị các nhân viên không chịu sự quản lý của ai

```
SELECT *  
FROM NHANVIEN  
Hiển thị các nhân viên không chịu sự quản lý của ai
```

- Thực thi câu lệnh ta có kết quả

	MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	PHG	LUONG
1	001	Nguyen	Bao	Tran	1978-01-01 00:00:00.000	nu	HA NOI		QL	3000000.0

Hình 7.19: Tìm kiếm thông tin có giá trị NULL hoặc NOT NULL

►Nhóm lệnh thực hiện tính toán

Hàm kết hợp

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

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

Ví dụ 7.24: Cho biết những mức lương trung bình và cao nhất của các nhân viên phòng có mã là 'NC'

```
SELECT AVG(Luong)As Luong_TB, MAX(Luong)AS Luong_CN  
FROM NHANVIEN  
WHERE Phg = 'NC'
```

- Thực thi câu lệnh có kết quả như sau:

	Luong_TB	Luong_CN
1	2125000.0	2500000.0

Hình 7.20: Sử dụng hàm kết hợp trong truy vấn

b. Truy vấn con

Trong nhiều trường hợp điều kiện chọn ở mệnh đề WHERE cần truy cập thông tin ở bảng khác với bảng đang truy vấn để kiểm tra điều kiện. Để thực hiện việc này người ta có thể sử dụng một câu SELECT khác lồng trong điều kiện ở mệnh đề WHERE. SELECT này được gọi là SELECT con hay truy vấn con.

Cấu trúc của truy vấn con có dạng như sau:

```
SELECT < ds các thuộc tính>
FROM < ds các bảng>
WHERE < điều kiện chứa select con 1 >(
    SELECT < ds các thuộc tính>
    FROM < ds các bảng>
    WHERE < điều kiện >
)
[AND | OR < điều kiện chứa select con 2>(
    SELECT < ds các thuộc tính>
    FROM < ds các bảng>
    WHERE < điều kiện>
)[...]
]
```

<Điều kiện chứa SELECT con 1> nằm sau mệnh đề WHERE trong SELECT cha thường có các dạng như sau:

Dạng 1:

<Tên cột> <so sánh> (<select con>)

Điều

kiện là đúng khi giá trị của cột so sánh đúng với giá trị trả về từ select con.

Ví dụ 7.25: Hiển thị họ tên của nhân viên có lương cao nhất

```
SELECT *
FROM NHANVIEN
WHERE Luong = (
    SELECT Max(Luong)
    FROM NHANVIEN ))
```

- Thực thi câu lệnh kết quả trả về

	MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	PHG	LUONG
1	001	Nguyen	Bao	Tran	1978-01-01 00:00:00.000	nu	HA NOI		QL	3000000.0
2	008	Vo	Thai	NGUYEN	1957-10-10 00:00:00.000	Nam	HA NOI		QL	3000000.0
3	009	NGUYEN	THANH	PHONG	1955-09-01 00:00:00.000	Nam	TPHCM	001	NC	3000000.0
4	012	BUI	TUAN	VU	1973-08-06 00:00:00.000	Nam	TRAVINH	003	DH	3000000.0

Hình 7.21: Sử dụng truy vấn con dạng 1

Dạng 2:

<Tên cột> <so sánh> **ALL** (<select con>)

Điều kiện đúng khi giá trị cột so sánh đúng với tất cả các giá trị trả về từ select con

Ví dụ 7.26: Hiển thị họ tên nhân viên có lương cao hơn tất cả các nhân viên của phòng NC

```
SELECT HONV,TENNV
FROM NHANVIEN
WHERE Luong > ALL (
    SELECT Luong
    FROM NHANVIEN
    WHERE Phg ='NC'
)
```

- Thực thi câu lệnh kết quả trả về là:

	HONV	TENNV
1	Nguyen	Tran
2	Vo	NGUYEN
3	BUI	VU

Hình 7.22: Sử dụng truy vấn con dạng 2

Dạng 3:

<Tên cột> <so sánh> **ANY** (<select con>)

Điều kiện đúng khi giá trị của cột so sánh đúng với bất kỳ một giá trị nào trả về từ select con

Ví dụ 7.27: Hiển thị họ tên của trưởng phòng ở tất cả các phòng ban

```
SELECT HoNV, TENLOT, TENNV
FROM NHANVIEN
WHERE MaNV= ANY (
        SELECT Trphg
        FROM PHONGBAN )
```

- Thực thi câu lệnh kết quả trả về là:

	HoNV	TENLOT	TENNV
1	Nguyen	Bao	Tran
2	NGUYEN	THANH	TUNG
3	Nguyen	THI	NHAN

Hình 7.23: Sử dụng truy vấn con dạng 3

Dạng 4:

<tên cột> **IN** (<select con>)

<tên cột> **NOT IN** (<select con>)

- **IN:** DK đúng khi giá trị của cột **nằm trong** tập hợp các giá trị trả về của select con

- **NOT IN:** ĐK đúng khi giá trị của cột **không nằm trong** tập hợp các giá trị trả về của select con

Ví dụ 7.28: Hiển thị họ tên của trưởng phòng ở tất cả các phòng ban

```
SELECT HoNV, TENLOT,TENNV
FROM NHANVIEN
WHERE MaNV IN (
        SELECT Trphg
        FROM PHONGBAN )
```

- Thực thi câu lệnh kết quả trả về là:

	MaNV	HoNV	TenLot	TenNV	NgSinh	Phai	DChi	MaNQL	PHG	LUONG
1	001	Nguyen	Bao	Tran	1978-01-01 00:00:00.000	nu	HA NOI		QL	3000000.0
2	002	NGUYEN	THANH	TUNG	1955-09-01 00:00:00.000	nam	TPHCM	001	NC	2500000.0
3	003	Nguyen	THI	NHAN	1960-10-12 00:00:00.000	nu	TPHCM	001	DH	2500000.0

Hình 7.24: Sử dụng truy vấn con dạng 4

Dạng 5:

```
EXISTS (<select con>)
NOT EXISTS (<select con>)
```

- **EXISTS** : Điều kiện đúng khi kết quả trả về của select con l khác rỗng
- **NOT EXISTS:** Điều kiện đúng khi kết quả trả về của select con là rỗng

Ví dụ 7.29: Cho biết họ tên những nhân viên làm chung một phòng và có mức lương bằng nhau.

```
SELECT HoNV, TENLOT,TENNV
FROM NHANVIEN
WHERE EXISTS (SELECT *
        FROM NHANVIEN nv2
        WHERE nv1.phg = nv2.phg
        AND nv1.luong = nv2.luong
        AND nv1.manv <> nv2.manv
    )
```

- Thực thi câu lệnh kết quả trả về như sau:

	honv	tenlot	tennv
1	Nguyen	Bao	Tran
2	Nguyen	THI	NHAN
3	DINH	BA	TIEN
4	NGUYEN	MANH	HUNG
5	TRAN	HONG	VAN
6	Vo	Thai	NGUYEN
7	LE	THI	THAM
8	DOAN	THUY	TIEN
9	CAO	MANH	TOAN
10	TRAN	HONG	HAI

Hình 7.25: Sử dụng truy vấn con dạng 5

Chú ý

Kết quả trả về của SELECT con được tóm tắt như sau:

SELECT con trả về	Một cột	Nhiều cột
Một dòng	Dạng 1	Dạng 5
Nhiều dòng	Dạng 2, 3, 4	

Bảng 7.3: Tóm tắt các dạng trả về từ Select con

► Mệnh đề GROUP BY

- Cú pháp:

```
SELECT <danh sách thuộc tính[với hàm kết hợp]>
FROM <danh sách các bảng>
WHERE <điều kiện>
GROUP BY <danh sách thuộc tính cần gom nhóm>
```

- Mệnh đề SELECT và mệnh đề GROUP BY liên quan chặt chẽ với nhau: Mỗi mục trong danh sách SELECT chỉ được **có một giá trị cho mỗi nhóm**. Vì vậy, mệnh đề SELECT chỉ có thể chứa: các cột phân nhóm, các hàm kết hợp, các hằng, các biểu thức kết hợp các loại trên.

Ví dụ 7.30: Hiển thị mã phòng và mức lương trung bình của từng phòng.

```
SELECT PHG, AVG(Luong)
FROM NHANVIEN
GROUP BY PHG
```

- Kết quả sau khi thực thi câu lệnh trên:

	Phg	luong_TB
1	DH	2666666.6666666665
2	NC	2071428.5714285714
3	QL	3000000.0

Hình 7.26: Gom nhóm theo từng phòng ban tính lương trung bình

► Mệnh đề HAVING

- Mệnh đề HAVING lấy các giá trị của hàm gom nhóm chỉ trên những nhóm nào thỏa điều kiện nhất định. Mệnh đề HAVING chỉ ra điều kiện lọc trên các nhóm, không phải điều kiện lọc trên từng bộ.

- Cú pháp:

```
SELECT <danh sách thuộc tính[với hàm kết hợp]>
FROM <danh sách các bảng>
WHERE <điều kiện>
GROUP BY <danh sách thuộc tính cần gom nhóm>
HAVING <điều kiện gom nhóm>
```

Ví dụ 7.31: Hãy cho biết các phòng ban có lương trung bình lớn hơn 2500000

```
SELECT PHG, AVG(Luong) as luong_TB
FROM NHANVIEN
GROUP BY PHG
HAVING AVG(Luong) > 2500000
```

- Thực thi câu lệnh kết quả như hình bên dưới

	PHG	luong_TB
1	DH	26666666.6666666665
2	QL	30000000.0

Hình 7.27: Những phòng có lương trung bình lớn hơn 2.500.000

7.2.4 Sao lưu và phục hồi dữ liệu

a. Sao lưu lại CSDL

Để sao lưu một CSDL ta thực hiện các bước như sau:

- **Bước 1.** Mở CSDL muốn sao lưu bằng lệnh
USE <Tên CSDL>
- **Bước 2.** Tạo thiết bị sao lưu gắn với một tập tin sao lưu bằng lệnh
EXEC sp_addumpdevice 'disk' , '<tên thiết bị>', '<đường dẫn và tên tập tin>'
- **Bước 3.** Ra lệnh sao lưu CSDL vào thiết bị sao lưu
BACKUP DATABASE <tên CSDL muốn chép> TO <Tên thiết bị>

Ví dụ 7.32 Sao lưu CSDL có tên là CSDL1 vào ổ đĩa D và đặt tên là CSDL1_backup.bak

- **Bước 1.** Mở CSDL cần chép
USE CSDL1
- **Bước 2.** Tạo thiết bị sao lưu tên BAK1 gắn với tập tin D:\CSDL1_backup.bak
EXEC sp_addumpdevice 'disk' , 'BAK1', 'D:\CSDL1_backup.bak '
- **Bước 3.** Ra lệnh sao lưu CSDL1 vào thiết bị sao lưu vừa tạo là BAK1
BACKUP DATABASE CSDL1 TO BAK1

♦**Lưu ý:** Không thể tạo hai thiết bị sao lưu cùng tên. Nghĩa là nếu đã tạo thiết bị BAK1 rồi thì những lần sao chép sau không cần phải thực hiện bước 2 nữa.

b. Phục hồi CSDL từ tập tin đã sao lưu

Giả sử ta muốn chép CSDL về máy nhà (đã có cài SQL Server). Thì ta sẽ thực hiện các bước sau:

- Bước 1. Chép tập tin CSDL đã được sao lưu như trên về máy tính nhà.
- Bước 2. Mở Query Analyzer
- Bước 3. Ra lệnh phục hồi CSDL từ tập tin sao lưu

RESTORE DATABASE <Tên CSDL> FROM DISK = ‘< đường dẫn và tên tập tin sao lưu >’

Ví dụ 7.33 Phục hồi CSDL1 đã sao lưu ở trên thành tên CSDL là QLSV, ta thực hiện câu lệnh:

RESTORE DATABASE QLSV FROM DISK = ‘D:\CSDL1_backup.bak’

► **Câu hỏi (bài tập) củng cố:**

- **Bài tập 1:** Với CSDL minh họa ‘Quản lý đề án’ ở mục 7.2.

- a. Hãy sử dụng các câu lệnh đã hướng dẫn ở trên để tạo các bảng, cài đặt RTBV
- b. Nhập dữ liệu như mẫu
- c. Viết câu truy vấn SQL cho các câu hỏi sau đây:
 1. Tìm những nhân viên sinh từ 1965 đến 1970
 2. Tìm họ tên NV và tên phòng ban NV đó trực thuộc có mức lương từ 2.000.000 đến 3.000.000
 3. Tìm những nhân viên có họ “Nguyen” hoặc tên bắt đầu bằng chữ “T”
 4. Tìm những NV thuộc một trong các phòng ban có mã là “NC” hoặc “QL”
 5. Lập DS thân nhân dưới 18 tuổi của các nhân viên
 6. In DSNV nữ trên 30 tuổi
 7. Tìm tên và địa chỉ của nhân viên thuộc phòng “Nghiên cứu”
 8. Tìm mã số và tên những nhân viên thuộc phòng ‘Nghiên Cứu’ tham gia đề án ‘Tin học hóa’ với thời gian làm việc 20 giờ/tuần.
 9. Tìm họ tên trưởng phòng đã chủ trì các đề án ở Hà Nội.
 10. Tìm những nhân viên được ‘Nguyễn Thanh Tùng’ trực tiếp phụ trách.
 11. Tính thời gian thấp nhất, thời gian cao nhất, trung bình và tổng số giờ làm việc trong tuần của tất cả các nhân viên.
 12. Tính thời gian làm việc trung bình trong tuần của các nhân viên
 13. Tìm họ tên nhân viên có mức lương trên mức lương trung bình của phòng ‘Nghiên Cứu’
 14. Tìm số lượng nhân viên của tất cả các phòng ban.
 15. Tính số lượng nhân viên chịu sự quản lý trực tiếp của người khác
 16. Tính số lượng nhân viên quản lý trực tiếp nhân viên khác.
 17. Tính số lượng nhân viên của từng phòng.
 18. Tính thời gian tham gia đề án cao nhất, thấp nhất, trung bình của tất cả các nhân viên trong từng phòng ban.
 19. Liệt kê tên phòng và số lượng nhân viên của các phòng ban có mức lương trung bình trên 2.200.000

- **Bài tập 2:** Viết câu truy vấn SQL trả lời cho các câu hỏi của Bài tập 2, trong KQHT4.

TÀI LIỆU THAM KHẢO

TÀI LIỆU THAM KHẢO ĐỂ BIÊN SOẠN NỘI DUNG MÔN HỌC:

1. Nguyễn Đăng Ty, Đỗ Phúc, *Giáo trình Cơ sở dữ liệu*, NXB Đại học Quốc Gia, 2010.
2. Hồ Thuận, Hồ Cẩm Hà, *Các hệ Cơ sở dữ liệu*, NXB Giáo dục, 2004.
3. Cao Thị Nhạn (2008), *Bài giảng Cơ sở dữ liệu*, Đại học Đà Lạt.
4. Nguyễn Xuân Huy, Lê Hoài Bắc, *Bài tập Cơ sở dữ liệu*, NXB Thống Kê, 2003.
5. Lê Tiến Vượng, *Nhập môn Cơ sở dữ liệu*, NXB Khoa học và kỹ thuật, 2000.
6. Tài liệu tham khảo Cơ sở dữ liệu, Đại học Cần Thơ.

TÀI LIỆU THAM KHẢO ĐỀ NGHỊ CHO HỌC VIÊN:

1. Nguyễn Xuân Huy, Lê Hoài Bắc, *Bài tập Cơ sở dữ liệu*, NXB Thống Kê, 2003.
2. Cao Thị Nhạn (2008), *Bài giảng Cơ sở dữ liệu*, Đại học Đà Lạt.
3. Tài liệu tham khảo Lý thuyết Cơ sở dữ liệu, Đại học Cần Thơ.
4. Tài liệu tham khảo Thực hành SQL Server, Đại học Cần Thơ.
5. *Giáo trình tham khảo môn Cơ sở dữ liệu*, Đại học Trà Vinh, Giáo viên bộ môn CNTT biên soạn.
6. www.w3schools.com