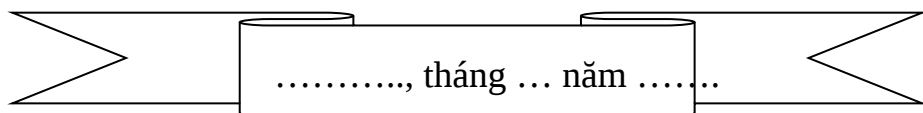




Giáo trình Cơ sở dữ liệu quan hệ Version 1.0



Mục lục

MỘT SỐ KÝ HIỆU VÀ QUY ƯỚC.....	3
Hình 1.2: Cấu trúc của một hệ cơ sở dữ liệu.....	4
1.1 Giới thiệu về hệ thống quản lý tệp truyền thống.....	5
1.2. Hệ cơ sở dữ liệu.....	6
1.2.1 Các thành phần của hệ cơ sở dữ liệu.....	6
1.2.2. Kiến trúc của một hệ cơ sở dữ liệu.....	8
1.3. Phân loại các hệ cơ sở dữ liệu.....	8
1.3.1. Các hệ tập trung	8
1.3.2. Các hệ cơ sở dữ liệu phân tán.....	9
1.4. Những ưu điểm của việc xây dựng một hệ cơ sở dữ liệu	
.....	10
1.5. Tính độc lập dữ liệu.....	11
1.6 Hệ quản trị cơ sở dữ liệu.....	11
1.7. Các mô hình dữ liệu.....	12
BÀI TẬP VÀ CÂU HỎI CHƯƠNG 1.....	18
Chương 2.....	19
2.4 Các phép toán của đại số quan hệ.....	21
3.1. Giới thiệu	36
3.2. Sơ đồ quan hệ.....	39
3.7. Các thuật toán.....	49
3.8 Phụ thuộc đa trị.....	60

MỘT SỐ KÝ HIỆU VÀ QUY ƯỚC

A, B, C, \dots là tên các thuộc tính đơn.

X, Y, Z, \dots là tập hợp các thuộc tính.

t, t_1, t_2, \dots là các bộ giá trị.

$t.[A]$: giá trị tại bộ t ứng với thuộc tính A .

$t.A$: giá trị tại bộ t ứng với thuộc tính A .

F : là tập các phụ thuộc hàm.

f : là kí hiệu của một phụ thuộc hàm

U : Tập hữu hạn các thuộc tính

R, S, \dots : Ký hiệu các quan hệ

r, s, \dots : Ký hiệu lược đồ quan hệ hoặc sơ đồ quan hệ.

$R(f)$: Ta nói quan hệ R thoả mãn phụ thuộc hàm f

PTH: Phụ thuộc hàm

$F \vdash f$: ta gọi f là một phụ thuộc hàm được suy dẫn logic từ F

$X \rightarrow Y$: Y phụ thuộc hàm vào X

$X \not\rightarrow Y$: Y không phụ thuộc hàm vào X

Sơ đồ quan hệ (lược đồ quan hệ).

DANH SÁCH CÁC HÌNH VẼ VÀ CÁC BẢNG DỮ LIỆU

Hình 1.1 :Các thành phần của một hệ cơ sở dữ liệu

Hình 1.2: Cấu trúc của một hệ cơ sở dữ liệu

Hình 1.3: Hệ cơ sở dữ liệu: a) Personal DB; b) Central DB

Hình 1.4: Client/Server Database

Hình 1.5: Hệ cơ sở dữ liệu phân tán

Bảng 1.1: Khách hàng

Bảng 1.2: Hàng hoá

Bảng 1.3: Hàng bán

Bảng 2.1: Chứa thông tin về học sinh

Bảng 2.2: Quan hệ LOPHOC

Bảng 2.3: Quan hệ SINHVIEN

Bảng 2.4: Quan hệ MONHOC

Bảng 2.5: Quan hệ DIEMTHI

Bảng 3.1: Sổ theo dõi việc bán hàng

Bảng 3.2: Chứa thông tin về hàng hoá

Bảng 3.3 Chứa thông tin về khách hàng

Bảng 3.4: Chứa thông tin về hoá đơn bán hàng

Bảng 3.5 : Chứa thông tin về chi tiết hoá đơn bán hàng

Bảng 3.6: Chứa thông tin về sinh viên

Bảng 3.7: Bảng chứng minh định lý của phép tách

Bảng 3.7: Bảng đăng ký học của sinh viên

Bảng 4.1: Các kiểu dữ liệu

Chương 1

NHẬP MÔN CƠ SỞ DỮ LIỆU

1.1 Giới thiệu về hệ thống quản lý tệp truyền thống

Hệ thống quản lý tệp truyền thống thường được tổ chức riêng rẽ, phục vụ cho một mục đích của một đơn vị hoặc một đơn vị con trực thuộc cụ thể.

Hệ thống quản lý tệp truyền thống cho phép ta tạo các tệp, truy cập và xử lý thông tin trong các tệp thông qua các chương trình ứng dụng. Các phần mềm ứng dụng này được viết bằng các ngôn ngữ lập trình đa năng như PASCAL, C ...

- Ưu điểm:

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

- Nhược điểm:

- Thông tin được tổ chức riêng rẽ ở nhiều nơi nên việc cập nhật dễ làm mất tính nhất quán dữ liệu.
- Hệ thống thông tin được tổ chức thành các hệ thống file riêng lẻ nên thiếu sự chia sẻ thông tin giữa các nơi.
- Có sự dư thừa dữ liệu rất lớn qua việc trùng lặp các tệp tin trong các ứng dụng khác nhau.
- Không gian đĩa bị lãng phí, khó khăn trong việc bảo trì hệ thống.
- Khó khăn trong việc truy xuất dữ liệu.

Một ví dụ điển hình về sự trùng lặp dữ liệu như trong Hệ quản lý nguồn nhân lực bao gồm ba hệ chính:

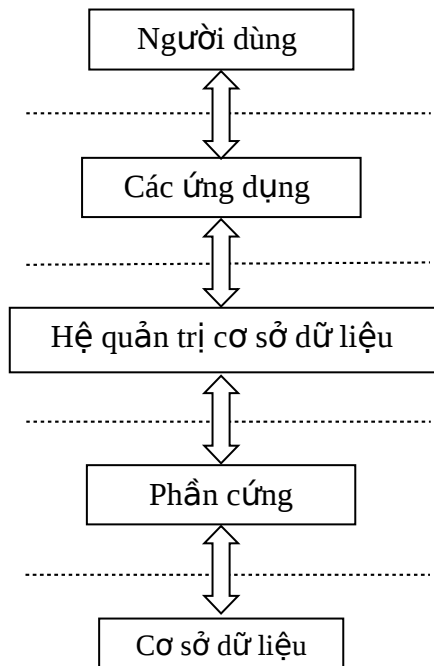
1. Hệ lương, hệ này duy trì ngày công và lương cho tất cả nhân viên.
2. Hệ nhân sự, hệ này duy trì lý lịch cá nhân, dữ liệu về tổ chức, công việc đào tạo và vị trí thăng tiến.
3. Hệ hưu, hệ này quản trị các qui tắc liên quan đến nghỉ hưu, loại nghỉ hưu. Chi tiết về hưu của từng nhân viên.

Vấn đề bất lợi là Hệ quản lý lương thông thường được quản lý bởi phòng Tài chính, trong khi Hệ quản lý nhân sự và Hệ quản lý hưu được quản lý bởi phòng Tổ chức cán bộ. Rõ ràng, có nhiều dữ liệu về nhân viên là chung cho cả ba hệ. Thường những hệ này thực hiện và lưu trữ riêng biệt nên chúng tạo ra sự trùng lặp dữ liệu.

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

1.2. Hệ cơ sở dữ liệu

1.2.1 Các thành phần của hệ cơ sở dữ liệu.



Hình 1.1 : Các thành phần của một hệ cơ sở dữ liệu

Các thành phần trong hệ CSDL gồm:

- **Người dùng (User), gồm có 4 đối tượng sử dụng:**

+ **Người quản trị cơ sở dữ liệu:** Trong những tổ chức có nhiều người cùng sử dụng chung một nguồn dữ liệu thì nhất thiết phải có một người đứng đầu quản lý, chịu trách nhiệm đối với nguồn dữ liệu này. Đó chính là người quản trị cơ sở dữ liệu (Database Administrators - DBA). DBA có nhiệm vụ tổ chức nội dung của cơ sở dữ liệu, tạo và cấp quyền truy cập cơ sở dữ liệu cho người dùng, đưa ra yêu cầu về phần cứng và phần mềm... nếu cần thiết. DAB cũng phải chịu trách nhiệm bảo vệ an toàn, Backup thông tin...khi có sự cố.

+ **Người phân tích và thiết kế hệ thống:** Là người chịu trách nhiệm: (a) xác định những dữ liệu nào cần lưu trữ trong CSDL; (b) lựa chọn những cấu trúc thích hợp để biểu diễn và lưu trữ; (c) phỏng vấn tất cả những người sử dụng CSDL sau này để hiểu được những yêu cầu của họ đối với CSDL; (d) tiến hành phân tích thiết kế hệ thống sau khi thống nhất được tất cả các yêu cầu của người sử dụng.

+ **Người viết chương trình ứng dụng:** Là người viết phần mềm phục vụ cho việc thực hiện các chức năng của hệ thống bằng những ngôn ngữ phù hợp, ngoài ra còn có các nhiệm vụ: (a) chạy thử chương trình (test); (b) chữa lỗi và gỡ rối chương trình (debug); (c) viết tài liệu, hướng dẫn sử dụng; (d) bảo trì hệ thống

+ **Người dùng cuối (EndUser):** Người dùng cuối là những người truy cập CSDL để: (a) cập nhật dữ liệu; (b) truy vấn dữ liệu; (c) thống kê, báo cáo. Mỗi EndUse chỉ có một quyền hạn trong phạm vi nhất định đối với cơ sở dữ liệu như quyền đọc, ghi, copy...)

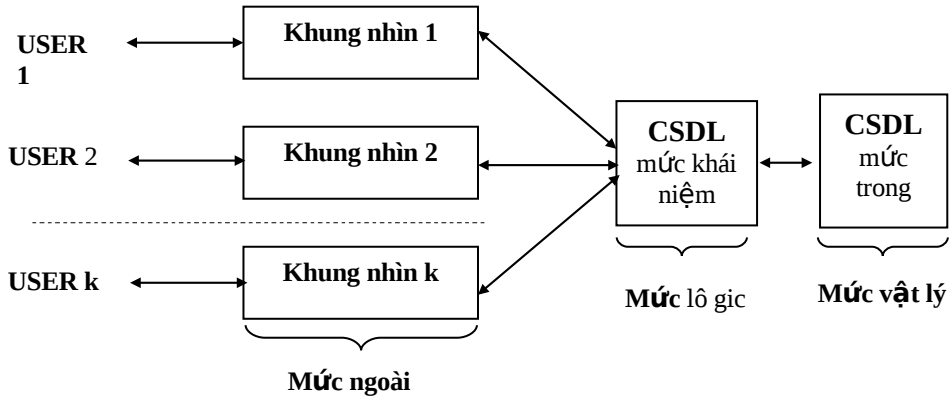
- **Các ứng dụng:** Các thao tác cần thiết truy cập vào cơ sở dữ liệu như tạo lập, xử lý, cập nhật dữ liệu.

- **Hệ quản trị cơ sở dữ liệu:** Hệ quản trị cơ sở dữ liệu là phần mềm cho phép định nghĩa các cấu trúc để lưu trữ dữ liệu và các thao tác trên dữ liệu sao cho đảm bảo sự an toàn và bí mật của dữ liệu. Hiện nay có một số hệ quản trị cơ sở dữ liệu thông dụng như FOXPRO, ACCESS, SQL SERVER, ORACLE...

- **Phần cứng:** Phần cứng là các thiết bị và các phương tiện được sử dụng để lưu trữ và truy cập vào cơ sở dữ liệu.

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

1.2.2. Kiến trúc của một hệ cơ sở dữ liệu



Hình 1.2: Cấu trúc của một hệ cơ sở dữ liệu

Cấu trúc một hệ cơ sở dữ liệu gồm ba mức:

+ **Mức ngoài:** Là mức sát với người sử dụng nhất, là cách nhìn, là quan niệm của từng người sử dụng đối với cơ sở dữ liệu mức khái niệm. Khả năng truy nhập tùy thuộc vào quyền hạn từng USER.

+ **Mức logic (CSDL mức khái niệm):** Là tập các dữ liệu được biểu diễn dưới dạng trừu tượng của cơ sở dữ liệu vật lý.

+ **Mức vật lý:** Là tập các dữ liệu được biểu diễn theo một cấu trúc nào đó, được lưu trên các thiết bị nhớ thứ cấp (như đĩa từ, băng từ ...).

1.3. Phân loại các hệ cơ sở dữ liệu

1.3.1. Các hệ tập trung

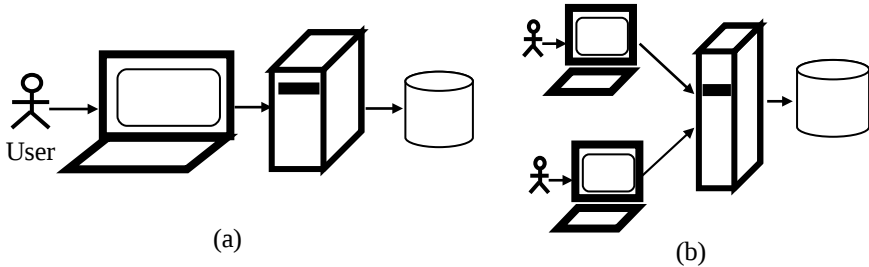
Hệ cơ sở dữ liệu tập trung là hệ trong đó CSDL được lưu trữ tại một vị trí nhất định, gồm các hệ cơ sở dữ liệu sau:

- **Hệ cơ sở dữ liệu cá nhân (Personal Database):** Mô hình này là một hệ cơ sở dữ liệu nhỏ chỉ gồm một máy tính cá nhân với một vài người sử dụng làm nhiệm vụ đơn lẻ với quy mô nhỏ.

- **Hệ cơ sở dữ liệu trung tâm (Central Database):** Hệ cơ sở dữ liệu trung tâm là một hệ đa người dùng từ thiết bị đầu cuối (terminal) tại đó có màn hình và

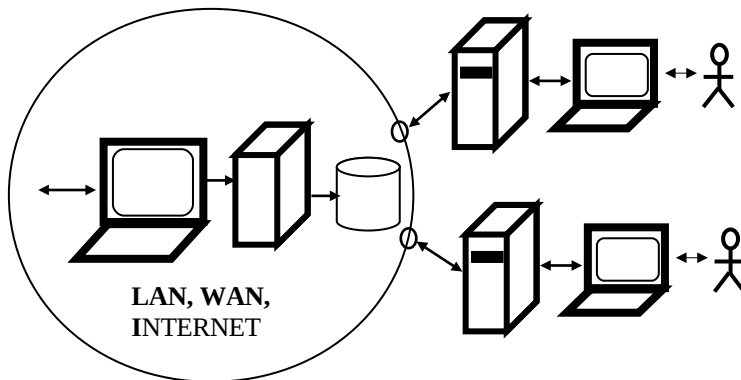
phím để trao đổi thông tin. Mọi xử lý, tính toán được thực hiện tại trung tâm với một máy tính mạnh có thể xử lý nhiều yêu cầu.

Một hệ như vậy khi máy tính trung tâm có sự cố, thì toàn bộ hệ thống sẽ ngừng hoạt động.



Hình 1.3: Hệ cơ sở dữ liệu; (a) Cá nhân; (b) Trung tâm

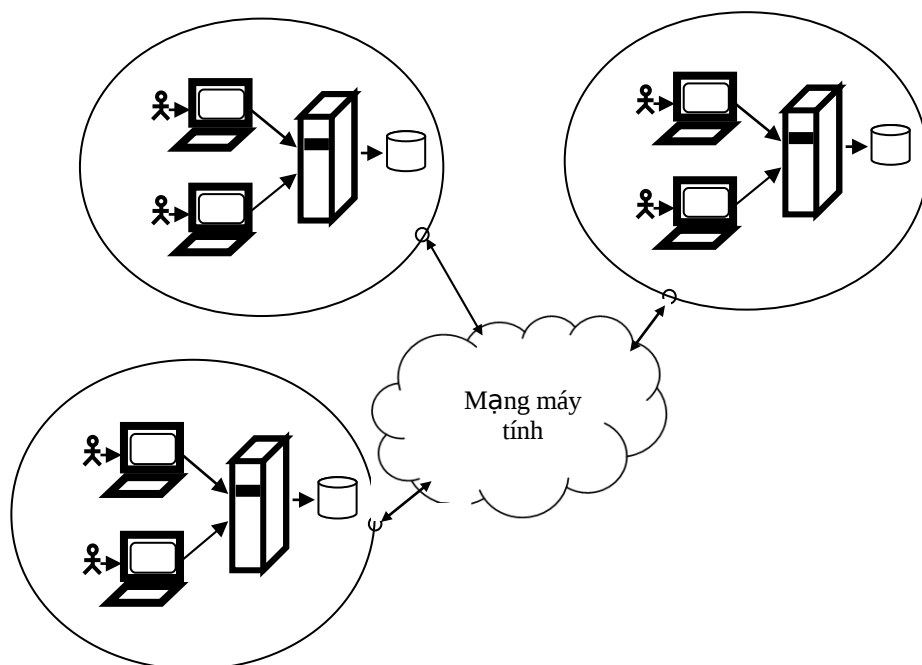
- Hệ cơ sở dữ liệu Client/Server (Client/Server Database): Cơ sở dữ liệu được lưu trữ tại máy chủ (Server) và nhiều máy trạm (Client) kết nối sử dụng chung cơ sở dữ liệu này.



Hình 1.4 Kiến trúc Client/Server Database

1.3.2. Các hệ cơ sở dữ liệu phân tán

Hệ cơ sở dữ liệu phân tán là hệ CSDL trong đó cơ sở dữ liệu được tổ chức phân bố thành nhiều cơ sở dữ liệu địa phương, được lưu trữ trên các máy tính ở các vị trí địa lý khác nhau nhưng vẫn cùng thuộc một hệ thống. Các cơ sở dữ liệu này được liên kết với nhau qua mạng máy tính phục vụ nhu cầu người dùng ở nhiều địa điểm khác nhau ở mức trong suốt.



Hình 1.5: Hệ cơ sở dữ liệu phân tán

Hệ cơ sở dữ liệu phân tán được phân thành hai loại

+ *Hệ thuần nhất*: Trong các hệ CSDL địa phương biểu diễn theo những mô hình giống nhau phương thức truy cập giống nhau.

+ *Hệ không thuần nhất*: Ngược lại với mô hình hệ thuần nhất là hệ không thuần nhất.

1.4. Những ưu điểm của việc xây dựng một hệ cơ sở dữ liệu

- Đảm bảo sự độc lập dữ liệu: Dữ liệu độc lập với chương trình làm cho dữ liệu được sử dụng rộng rãi và thuận lợi hơn.

- Giảm thiểu việc dư thừa dữ liệu: Khác với hệ thống tập, hệ thống cơ sở dữ liệu tổ chức theo cấu trúc thống nhất, hợp lý hạn chế việc lưu trữ tại nhiều nơi.

- Đảm bảo tính nhất quán và toàn vẹn dữ liệu: Do ít dư thừa nên hạn chế được sự dị thường khi thay đổi, cập nhật.

- Tăng tính dùng chung: Cơ sở dữ liệu có khả năng cho nhiều người truy cập sử dụng mỗi người nhìn vào cơ sở dữ liệu như nó là của riêng mình không bị ảnh hưởng bởi người khác.

- Tăng khả năng phát triển các ứng dụng: Do có sự mở rộng giao lưu nên khả năng sáng tạo cải tiến thuận lợi hơn.
- Tính chuẩn hoá cao.
- Chất lượng dữ liệu được cải thiện.
- Giảm bớt chi phí bảo trì hệ thống.

1.5. Tính độc lập dữ liệu

Tính độc lập dữ liệu là sự bất biến của chương trình ứng dụng đối với các thay đổi trong cấu trúc lưu trữ và chiến lược truy nhập vào cơ sở dữ liệu. Tính độc lập dữ liệu ở đây có hai mặt:

- Độc lập về vật lý: Là sự độc lập trong lưu trữ, chương trình ứng dụng không phụ thuộc vào việc dữ liệu được lưu giữ ở đâu hoặc lưu giữ như thế nào trên thiết bị nhớ thứ cấp.

- Độc lập về lôgic: Sự thay đổi, thêm bớt thông tin về các thực thể ở mức quan niệm không đòi hỏi thay đổi các khung nhìn của người sử dụng dẫn tới không cần thay đổi chương trình ứng dụng.

1.6 Hệ quản trị cơ sở dữ liệu

1.6.1 Các chức năng của một hệ quản trị CSDL

Một hệ quản trị CSDL thực hiện các chức năng sau:

- + Tạo cấu trúc dữ liệu tương ứng với mô hình dữ liệu được chọn.
- + Đảm bảo tính độc lập dữ liệu.
- + Cho phép cập nhật dữ liệu.
- + Kết xuất ra được các báo cáo từ các dữ liệu trong CSDL.
- + Đảm bảo tính an toàn và toàn vẹn dữ liệu trong CSDL.
- + Cung cấp các tiện ích sao lưu phục hồi dữ liệu.
- + Cung cấp các thủ tục điều khiển tương tranh.

1.6.2. Các thành phần của một hệ QTCSDL

Một hệ quản trị thông thường có các thành phần chính sau:

- + Ngôn ngữ định nghĩa dữ liệu (Data Definition Language).
- + Ngôn ngữ thao tác dữ liệu (Data Manipulation Language).
- + Ngôn ngữ hỏi đáp dữ liệu (Query Language).
- + Bộ viết báo cáo.

- + Từ điển dữ liệu.
- + Bộ phát sinh đồ họa.

1.7. Các mô hình dữ liệu.

Mô hình dữ liệu cho phép người dùng biểu diễn cơ sở dữ liệu dưới cấu trúc thuật ngữ dễ hiểu. Một mô hình dữ liệu là một hình thức mô tả toán học bao gồm:

- + Một hệ thống các ký hiệu để mô tả dữ liệu.
- + Tập các phép toán để thao tác trên cơ sở dữ liệu.

Vào những năm đầu của thập kỷ 60 (thế kỷ 20), mô hình mạng và mô hình phân cấp là thế hệ đầu tiên của họ các mô hình dữ liệu. Sang đầu thập kỷ 70. E.F. Codd đề xuất mô hình quan hệ mới, đó chính là thế hệ thứ hai. Mô hình quan hệ này có cấu trúc chặt chẽ, sáng sủa, nhất quán và có tính trực quan cao.

1.6.1 Khái niệm về thực thể và liên kết.

a) Thực thể

Thực thể (entity) là đối tượng cụ thể hay trừu tượng mà ta cần quan tâm trong công tác quản lý. Tên thực thể là danh từ.

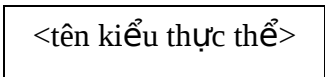
Thí dụ 1.1: Quản lý thư viện ta có các thực thể như: “Sách”, “Độc giả”... là các đối tượng cụ thể. Các đối tượng trừu tượng có thể là: Khoa công nghệ thông tin, Ngành toán ứng dụng....

b) Kiểu thực thể

Kiểu thực thể là tập hợp các thực thể (đối tượng) cùng được mô tả bằng những đặc trưng, tính chất giống nhau.

Thí dụ 1.2: Một nhân viên là một thực thể, tập hợp các nhân viên của cùng một hệ thống tạo thành một kiểu thực thể.

Biểu diễn một kiểu thực thể: Là một hình chữ nhật bên trong ghi tên của kiểu thực thể.



Thí dụ 1.3: Biểu diễn các thực thể « Nhân viên », « Sách », « Độc giả »:



Ghi chú: Thể hiện của kiểu thực thể là một thực thể, nó là một phần tử trong tập hợp hay lớp của kiểu thực thể. Vì vậy trong các ứng dụng để tránh sử dụng nhiều khái niệm ta đồng nhất thực thể và kiểu thực thể.

c) Thuộc tính (Attribute)

Thuộc tính là dữ liệu dùng để mô tả một đặc trưng của thực thể. Mỗi thực thể có một tập các thuộc tính. Tên thuộc tính phải là danh từ.

Thí dụ 1.4: Thực thể “Sách” có các thuộc tính: Tên sách, tên tác giả, nhà xuất bản...

1.6.2. Liên kết và kiểu liên kết

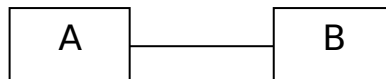
Liên kết: Là một sự ghép nối giữa hai hay nhiều thực thể phản ánh một thực tế quản lý.

Thí dụ 1.5: Ông Nguyễn Văn Hưng làm việc ở phòng Đào tạo; Hoá đơn số 60 gửi cho khách hàng Trần Văn Hùng; Sinh viên Dương Văn Việt thuộc lớp CNTT1A

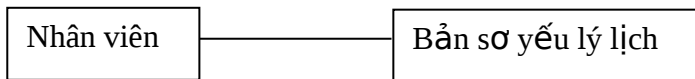
Phân loại liên kết:

+ Liên kết 1-1 (liên kết một - một): Hai kiểu thực thể A và B có mối liên kết 1-1 nếu một thực thể kiểu A tương ứng với một thực thể kiểu B và ngược lại.

Kí hiệu:



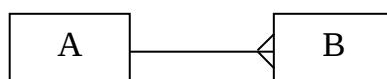
Thí dụ 1.7:



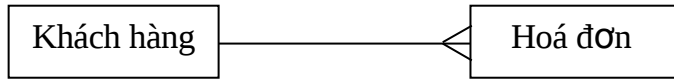
Ghi chú: Trong biểu đồ cấu trúc dữ liệu hai kiểu thực thể có mối liên kết 1-1 có thể được đồng nhất thành một kiểu thực thể.

+ Liên kết 1-n (một - nhiều): Hai kiểu thực thể A và B có mối liên kết 1-n nếu một thực thể kiểu A tương ứng với nhiều thực thể kiểu B và ngược lại một thực thể kiểu B tương ứng với duy nhất một thực thể kiểu A.

Kí hiệu:

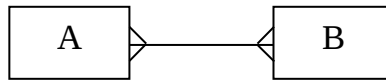


Thí dụ 1.8:

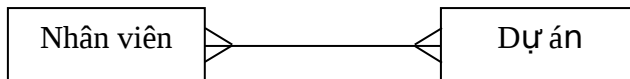


+ Liên kết n-n (nhiều - nhiều): Hai kiểu thực thể A và B có mối liên kết n - n nếu một thực thể kiểu A tương ứng với nhiều thực thể kiểu B và ngược lại.

Kí hiệu:

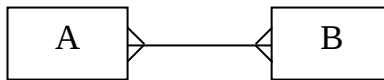


Thí dụ 1.9:

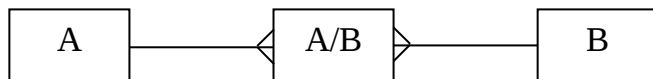


Ghi chú: Trong biểu đồ cấu trúc dữ liệu nếu tồn tại mối liên kết n-n giữa các kiểu thực thể, ta cần chuẩn hoá nó đưa về dạng liên kết một-nhiều:

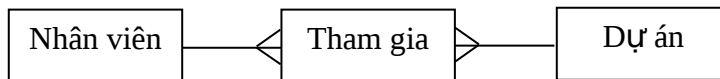
Dạng nhiều- nhiều



Đưa về dạng
một - nhiều



Thí dụ 1.10:



1.6.3. Các mô hình dữ liệu

- *Mô hình dữ liệu phân cấp:* Mô hình dữ liệu phân cấp (Hierarchical Data Model) - được gọi tắt là mô hình phân cấp được đưa ra vào những năm 60, trong mô hình dữ liệu này dữ liệu được tổ chức thành cấu trúc cây, trong đó các

nút (node) của cây biểu diễn các bản ghi, giữa các bản ghi liên kết với nhau theo mối quan hệ cha con:

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

Ưu điểm:

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

Nhược điểm:

- Không thể hiện được mối quan hệ n-n
- Trong một hệ thống phân cấp, dữ liệu được tổ chức như trên dẫn đến khó sửa đổi dữ liệu.
- Lặp lại dữ liệu, lãng phí bộ nhớ và tốn nhiều công sức tạo lập.

- *Mô hình dữ liệu mạng:*

Mô hình dữ liệu mạng (Network Data Model) được gọi tắt là mô hình mạng (Network Model) là mô hình dữ liệu được biểu diễn bởi một đồ thị có hướng. Trong mô hình mạng người ta dùng hai yếu tố là bản ghi và liên kết. Khái niệm bản ghi giống như mô hình phân cấp, liên kết là tập các con trỏ vật lý thiết lập quan hệ chủ sở hữu giữa tập bản ghi này với tập bản ghi khác. So sánh hai mô hình ta thấy bản ghi “đơn hàng” liên kết với bản ghi “số lượng” bản ghi “số lượng” cũng có liên kết với bản ghi “mặt hàng” và bản ghi “số lượng” là thành viên của hai bản ghi chủ khác nhau.

Mô hình mạng người ta đã khắc phục được việc dư thừa dữ liệu của mô hình phân cấp. Tuy vậy cấu trúc hệ thống phức tạp ngoài nội dung thông tin, mỗi bản ghi còn có thêm thông tin nữa là địa chỉ để truy nhập tới bản ghi thành viên. Với mỗi liên kết phải có nhãn để xác định liên kết.

Ưu điểm:

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

Nhược điểm:

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

- Mô hình quan hệ:

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

- Một hệ thống các ký hiệu để mô tả dữ liệu dưới dạng dòng và cột như quan hệ, bộ, thuộc tính, khóa chính, khóa ngoại, ...

- Một tập hợp các phép toán thao tác trên dữ liệu như phép toán tập hợp, phép toán quan hệ.

Vì tính chất chặt chẽ của toán học về lí thuyết tập hợp nên mô hình này đã mô tả dữ liệu một cách rõ ràng, uyển chuyển và trở thành rất thông dụng.

Ngày nay hầu hết các HQTCSDL đều tổ chức dữ liệu theo mô hình dữ liệu quan hệ.

Thí dụ 1.11:

Xét một hệ thông tin phân phối hàng, hệ này quản lý hoạt động bán hàng cho khách. Các kiểu thực thể chính của hệ thống bao gồm:

Kiểu thực thể **Khách Hàng** gồm các thuộc tính: Mã khách hàng (MaKH), Tên khách hàng (TenKH), tuổi (Tuoi), Địa chỉ khách hàng (DiaChi)

Kiểu thực thể **Hàng Hoá** gồm các thuộc tính: Mã hàng hoá (MaHang), Tên hàng hoá (TenHang), Giá (Gia), Màu sắc của mặt hàng (Mau), Đơn vị tính (DVT)

Kiểu thực thể **Bán Hàng** gồm các thuộc tính: MaKH, MaHang, số lượng (SoLuong)

Ứng với mỗi kiểu thực thể ta có một bảng dữ liệu sau:

Bảng Khách Hàng

Bảng 1.1: Khách hàng

Mã Khách	Tên Khách	Tuổi	Địa Chỉ
KH1	A	20	Hà nội
KH2	B	21	Hà tây
KH3	C	19	Thái Nguyên

Bảng Hàng Hoá

Bảng 1.2: Hàng hoá

Mã Hàng	Tên Hàng	Giá	ĐVT	Màu
MH1	Bóng rổ	500	Quả	Vàng
MH2	Bóng đá	150	Quả	Đỏ
MH3	Bóng chuyền	100	Quả	Xanh

Bảng Bán Hàng

Bảng 1.3 Hàng bán

Mã Khách	Mã Hàng	Số Lượng
KH1	MH2	50
KH1	MH3	40
KH2	MH3	60
KH2	MH1	90
KH3	MH 3	80

Một cơ sở dữ liệu theo mô hình quan hệ thực chất là một tập các bảng mà:

- Mỗi bảng gọi là một quan hệ/ kiểu thực thể/ tệp.
- Mỗi hàng gọi là một bộ/ thực thể/ bản ghi.
- Mỗi cột gọi là một thuộc tính/ trường.

1.6.4 Đánh giá các mô hình

-Về cách biểu diễn dữ liệu:

- Mô hình 1: Tập các cây quá lớn nếu dữ liệu nhiều và phức tạp
- Mô hình 2: Tập các đồ thị có hướng phức tạp
- Mô hình 3: Tập các bảng dễ quan sát vì các bảng độc lập với nhau

-Thao tác trên các mô hình:

+ Thao tác bổ sung thêm một bản ghi

- Mô hình 1, 2: Không thể bổ sung thêm các bản ghi thành viên mà không có bản ghi chủ. Ví dụ thêm mặt hàng 5 thì phải thuộc đơn hàng nào?.
- Mô hình 3: Có thể bổ sung vào bảng “hàng hoá” dễ dàng

+ Xoá một bản ghi

- Mô hình 1: Phải xoá toàn bộ cây mà nó là gốc, sửa lại cây mà nó là nhánh
- Mô hình 2: Phải xoá các bản ghi thành viên của nó, sửa lại dây chuyền mà nó là thành viên
- Mô hình 3: Đơn giản ta chỉ xoá dòng có bản ghi đó và các bản ghi ở bảng có liên kết với nó

+ Tìm kiếm bản ghi

- Mô hình 1: Tìm trên các nhánh của cây
- Mô hình 2 : Tìm trên toàn bộ dây chuyền
- Mô hình 3: Tìm trên các bảng

Như vậy so sánh các thao tác ta thấy với mô hình quan hệ đơn giản thuận tiện hơn so với các mô hình trên.

BÀI TẬP VÀ CÂU HỎI CHƯƠNG 1

1. Định nghĩa cơ sở dữ liệu.
2. Nêu các thành phần của một hệ cơ sở dữ liệu.
3. Nêu kiến trúc của một hệ cơ sở dữ liệu.
4. Phân loại các hệ cơ sở dữ liệu
5. Nêu ưu điểm của việc thiết kế một hệ cơ sở dữ liệu.
6. Nêu tính độc lập dữ liệu.
7. Trình bày khái niệm về hệ quản trị cơ sở dữ liệu? Các hệ quản trị cơ sở dữ liệu hiện nay đang được sử dụng.
8. Nêu các chức năng và các thành phần của một hệ quản trị cơ sở dữ liệu.
9. Thế nào là mô hình dữ liệu ? Các mô hình dữ liệu.

Chương 2 MÔ HÌNH DỮ LIỆU QUAN HỆ

2.1. Thuộc tính

-Thuộc tính (Attribute): Thuộc tính là dữ liệu dùng để mô tả một đặc trưng của thực thể. (Các thuộc tính đơn thường ký hiệu là các chữ cái A,B,C,... Tập thuộc tính thường ký hiệu là các chữ cái X, Y, Z...). Các thuộc tính được phân biệt qua tên gọi và phải thuộc 1 kiểu dữ liệu nhất định (kiểu dữ liệu là kiểu đơn). Tên nên đặt sát với ý nghĩa của nó, mang tính gợi nhớ và không nên quá dài.

-Miền thuộc tính: Là tập hợp các thuộc tính của thực thể, các thực thể thường có rất nhiều thuộc tính, tuy vậy để quản lý ta chỉ cần quản lý một số thuộc tính cần thiết cho thông tin về thực thể.

-Miền trị của thuộc tính (Domain): Là một tập hợp các giá trị của thuộc tính, ký hiệu là $DOM(A_i)$ với $i=1, \dots, n$.

Ví dụ 2.1: Thuộc tính GIOITINH có miền trị là $DOM(GIOITINH) = \{nam, nữ\}$

2.2. Quan hệ

Định nghĩa: Gọi $U = \{A_1, A_2, A_3, \dots, A_n\}$ là tập hữu hạn của các thuộc tính, mỗi thuộc tính A_i với $i=1, \dots, n$ có miền giá trị tương ứng là $DOM(A_i)$. Quan hệ R xác định trên tập thuộc tính U là tập con của tích Đề – Các.

$$R(U) \subseteq DOM(A_1) \times DOM(A_2) \times \dots \times DOM(A_n)$$

Ký hiệu quan hệ R xác định trên tập thuộc tính U là $R(U)$ hoặc $R(A_1, A_2, \dots, A_n)$.

Hay có thể viết dưới dạng sau:

$R(U) =$	A_1	A_2	\dots	A_n	
	a^1_1	a^1_2	\dots	a^1_n	}
	a^2_1	a^2_2	\dots	a^2_n	
	\dots	\dots	\dots	\dots	
	a^m_1	a^m_2	\dots	a^m_n	
	n thuộc tính				

Trong đó: A_1, A_2, \dots, A_n : Gọi là miền thuộc tính của quan hệ R

$DOM(A_1) = \{a^1_1, a^2_1, \dots, a^m_1\}$: Gọi là miền trị của thuộc tính A_1

n : Gọi là bậc của quan hệ R

m : Gọi là lực lượng của quan hệ R

Ta thấy so với một bảng thì:

- Mỗi quan hệ tương ứng với một bảng dữ liệu (là một tập dữ liệu)
- Mỗi thuộc tính tương ứng với một cột dữ liệu trong bảng (là một trường)
- Mỗi bộ tương ứng với một hàng của bảng dữ liệu (là một bản ghi)

Ví dụ 2.2 :

Cho tập thuộc tính U gồm có các thuộc tính: TÊN, Giới tính và Tuổi Ta có miền trị của chúng như sau:

$DOM(Tên) = \{Mai, Trung, Hoa, Anh\}$

$DOM(Giới\ tính) = \{Nam, Nữ\}$

$DOM(Tuổi) = \{15, 16, 17\}$

Từ đó, ta xây dựng được quan hệ **học sinh** là một tập con Tích Đề các của miền trị các thuộc tính trên như sau:

Bảng 2.1: Chứa thông tin về học sinh

Tên	Giới Tính	Tuổi
Mai	Nữ	15
Anh	Nam	16
Trung	Nam	15
Mai	Nữ	16
Anh	Nữ	15
Trung	Nam	17

2.3. Khoá của một quan hệ

Cho quan hệ R xác định trên tập thuộc tính U, K là một tập thuộc tính $K \subseteq U$. Gọi K là khoá của quan hệ R nếu với \forall bộ $t_i, t_j \in R$; $t_i \neq t_j$ thì $t_i[K] \neq t_j[K]$ (tức

là giá trị trên K của một bộ nào đó khác giá trị trên K của mọi bộ còn lại, hay nói cách khác bộ đó là xác định duy nhất).

Xét thêm rằng nếu với $\forall t_i, t_j \in R; i \neq j$ sao cho $t_i[K] = t_j[K]$ thì $t_i \equiv t_j$ thì K là khóa của quan hệ R

Ví dụ 2.3 : Xét quan hệ R có dạng sau

R	A	B	C	D
$t_1=$	a_1	b_1	c_1	d_1
$t_2=$	a_2	b_2	c_2	d_2
$t_3=$	a_3	b_3	c_2	d_2
$t_4=$	a_1	b_2	c_2	d_2

Xét thấy: $K_1=U, K_2=ABC, K_3=AB, \dots$ là các khóa của quan hệ R.

Nhưng $X=BC$ không phải là tập khóa của quan hệ R vì $t_2.X = t_4.X$ nhưng $t_2 \neq t_4$

2.4 Các phép toán của đại số quan hệ

2.4.1 Quan hệ khả hợp

Hai quan hệ R, S gọi là khả hợp nếu chúng có cùng bậc (số các thuộc tính bằng nhau) và miền trị của thuộc tính thứ i của quan hệ này bằng miền trị của thuộc tính thứ i trong quan hệ kia.

Ví dụ 2.4 :

Cho hai quan hệ $R = \{ A_1, A_2, \dots, A_n \}$ và $S = \{ A'_1, A'_2, \dots, A'_n \}$

nếu thỏa mãn $DOM(A_i) = DOM(A'_i)$, với $i=1, \dots, n$ thì R và S gọi là hai quan hệ khả hợp.

Chú ý: Nếu hai quan hệ có cùng bậc và tên thuộc tính thứ i trong quan hệ này khác tên với thuộc tính thứ i trong quan hệ kia nhưng chúng có cùng miền trị thì hai quan hệ này cũng là hai quan hệ khả hợp.

2.4.2 Các phép toán

(1) Phép hợp

Hợp của hai quan hệ R và S khả hợp là tập các bộ thuộc R hoặc thuộc S.

Ký hiệu phép hợp là $R \cup S$.

Biểu diễn hình thức phép hợp có dạng:

$$R \cup S = \{ t \mid t \in R \text{ hoặc } t \in S \}$$

Ví dụ 2.5: Cho hai quan hệ R và S có dạng sau:

R	A	B	C
	a ₁	b ₁	c ₁
	a ₁	b ₂	c ₂

S	A	B	C
	a ₁	b ₁	c ₁
	a ₂	b ₂	c ₂

Ta có :

R ∪ S =	A	B	C
	a ₁	b ₁	c ₁
	a ₁	b ₂	c ₂
	a ₂	b ₂	c ₂

Ví dụ 2.6: Cho hai quan hệ R và S có dạng sau:

R	A	B	C
	a ₁	2	c ₁
	a ₁	3	c ₂
	a ₂	4	c ₂

S	A'	B'	C
	a ₁	2	c ₁
	a ₂	4	c ₂
	a ₃	5	c ₃

Ta có :

R ∪ S =	A	B	C
	a ₁	2	c ₁
	a ₁	3	c ₂
	a ₂	4	c ₂
	a ₃	5	c ₃

(2) Phép giao

Giao của hai quan hệ R và S khả hợp là tập các bộ thuộc cả quan hệ R và S. Ký hiệu phép giao là $R \cap S$

Biểu diễn hình thức phép giao có dạng:

$$R \cap S = \{ t \mid t \in R \text{ và } t \in S \}$$

Ví dụ 2.7: Cho hai quan hệ R và S có dạng sau :

R	A	B	C
	a ₁	b ₁	c ₁
	a ₁	b ₂	c ₂
	a ₂	b ₂	c ₂

S	A	B	C
	a ₁	b ₁	c ₁
	a ₂	b ₂	c ₂

Ta có :

$R \cap S =$	A	B	C
	a ₁	b ₁	c ₁
	a ₂	b ₂	c ₂

Ví dụ 2.8: Cho hai quan hệ R và S có dạng sau :

R	A	B	C
	a ₁	2	c ₁
	a ₁	3	c ₂
	a ₂	4	c ₂

S	A'	B'	C'
	a ₁	2	c ₁
	a ₂	4	c ₂
	a ₃	5	c ₃

Ta có :

$R \cap S =$	A	B	C
	a ₁	2	c ₁
	a ₂	4	c ₂

(3) Phép trừ

Hiệu của hai quan hệ R và S khả hợp là tập các bộ thuộc R nhưng không thuộc S. Ký hiệu phép trừ là $R - S$.

Biểu diễn hình thức phép trừ có dạng:

$$R - S = \{ t \mid t \in R \text{ và } t \notin S \}$$

Ví dụ 2.9: Cho hai quan hệ R và S có dạng sau :

R	A	B	C
	a ₁	b ₁	c ₁

S	A	B	C
	a ₁	b ₁	c ₁

a_1	b_2	c_2
a_2	b_2	c_2

a_2	b_2	c_2

Ta có :

$R - S =$	A	B	C
	a_1	b_2	c_2

Ví dụ 2.10: Cho hai quan hệ R và S có dạng sau :

R	A	B	C
	a_1	2	c_1
	a_2	4	c_2

S	A'	B'	C'
	a_1	2	c_1
	a_2	4	c_2
	a_3	5	c_3
	a_1	3	c_2

Ta có :

$S - R =$	A'	B'	C'
	a_3	5	c_3
	a_1	3	c_2

Chú ý: Ta thấy $R - S = R - (R \cap S)$

(4) Phép chiếu

Phép chiếu của quan hệ R trên tập thuộc tính X ($X \subseteq U$), ký hiệu $\Pi_X(R)$ là một tập các bộ, được xây dựng bằng cách loại bỏ đi từ các bộ t trong quan hệ R những thuộc tính không nằm trong X.

Để thuận tiện cho việc biểu diễn hình thức phép chiếu, quy ước một số ký hiệu như sau: Gọi t là một bộ thuộc R, A là một thuộc tính ($A \subseteq U$), $t[A]$ là giá trị của bộ t tại thuộc tính A. Giả sử $X \subseteq U$ với $X = \{B_1, B_2, \dots, B_m\}$ khi đó $t[X] = (t[B_1], t[B_2], \dots, t[B_m])$. Vậy ta có $\Pi_X(R) = \{t[X] \mid t \in R\}$

Ví dụ 2.11: Cho quan hệ R và tập thuộc tính X (với $X=AB$)

R	A	B	C
	a ₁	b ₁	c ₁
	a ₂	b ₂	c ₂
	a ₁	b ₂	c ₂

Vậy phép chiếu trên tập thuộc tính X của quan hệ R có dạng sau:

$\Pi_X(R) =$	A	B
	a ₁	b ₁
	a ₂	b ₂
	a ₁	b ₂

(5) Phép chọn

-Phép chọn là phép toán lọc ra trong

một quan hệ một tập con các bộ thỏa mãn các điều kiện của biểu thức chọn F.

-Biểu thức chọn F: là một tổ hợp Boolean của các toán hạng, mỗi toán hạng là một phép so sánh đơn giản giữa hai biến là hai thuộc tính hoặc giữa một biến là một thuộc tính và một hằng, cho giá trị đúng hoặc sai đối với mỗi bộ dữ liệu.

Các phép so sánh: >, >, =, <, <=, <>

Các phép logic là: \wedge (và), \vee (hoặc), \neg (phủ định)

-Biểu diễn hình thức của phép chọn:

$$\sigma_F(R) = \{t / t \in R \text{ và } t(F)=\text{True}\}$$

Ví dụ 2.12: Cho quan hệ R có dạng sau:

R	A	B	C
	a ₁	b ₁	c ₁
	a ₂	b ₂	c ₂
	a ₃	b ₃	c ₃

Với biểu thức chọn F: B = "b₁", ta có

$\sigma_F(R) =$	A	B	C
	a ₁	b ₁	c ₁

Với biểu thức chọn F: $B \neq "b_1"$, ta có

$\sigma_F(R) =$	A	B	C
	a ₂	b ₂	c ₂
	a ₃	b ₃	c ₃

Với biểu thức chọn F: $(A = "b_2") \wedge (C = "c_2")$

$\sigma_F(R) =$	A	B	C
	a ₂	b ₂	c ₂

(6) Phép tích ĐỀ - các

Cho R là một quan hệ xác định trên tập thuộc tính (A_1, A_2, \dots, A_n) và quan hệ S xác định trên tập thuộc tính (B_1, B_2, \dots, B_m) . Tích ĐỀ - các của R và S là một quan hệ gồm $(n+m)$ thuộc tính và mỗi bộ của quan hệ kết quả có dạng n thành phần đầu là một bộ thuộc R và m thành phần sau là một bộ thuộc S

$R \times S = \{ t \mid t \text{ có dạng } (a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m) \text{ trong đó } \{a_1, a_2, \dots, a_n\} \in R; \{b_1, b_2, \dots, b_m\} \in S \}$

Ví dụ 2.13: Cho hai quan hệ R và S có dạng sau:

R	A	B	C
	a ₁	b ₁	c ₁
	a ₁	b ₂	c ₂

S	D	E	F
	d	e	f
	d'	e'	f'

Ta có kết quả của phép tích ĐỀ các:

R x S =	A	B	C	D	E	F
----------------	----------	----------	----------	----------	----------	----------

	a ₁	b ₁	c ₁	d	e	f
	a ₁	b ₁	c ₁	d'	e'	f'
	a ₁	b ₂	c ₂	d	e	f
	a ₁	b ₂	c ₂	d'	e'	f'

Ví dụ 2.14: Cho hai quan hệ R và S có dạng sau:

R	A	B
	α	1
	β	2

S	B	C	D
	α	4	d ₁
	β	5	d ₂
	β	3	d ₃
	γ	7	d ₄

Ta có phép Tích Đề Các của R và S là:

RxS=	A	R.B	S.B	C	D
	α	1	α	4	d ₁
	α	1	β	5	d ₂
	α	1	β	3	d ₃
	α	1	γ	7	d ₄
	β	2	α	4	d ₁
	β	2	β	5	d ₂
	β	2	β	3	d ₃
	β	2	γ	7	d ₄

(7) Phép kết nối

Phép kết nối là phép toán với hai quan hệ dựa trên các điều kiện để liên kết với nhau:

-Gọi θ là một trong các phép so sánh: $>$; $<$; \leq ; \geq ; $<>$; $=$

-Biểu thức kết nối có dạng: $F = A \theta B$

Từ đó ta có phép kết nối được xác định như sau:

$$R \bowtie_F S = \{ t(u,v) / u \in R; v \in S \text{ và thỏa mãn biểu thức chọn } F \}$$

Ví dụ 2.15: Cho hai quan hệ R và S có dạng sau:

R	A	B	C
	a ₁	b ₁	1
	a ₂	b ₂	2
	a ₃	b ₃	3

S	E	D
	1	d ₁
	2	d ₂

Gọi F là biểu thức kết nối hai quan hệ R và S (F = C ≥ E)

$$Ta \text{ có } R \bowtie_F S = \begin{array}{ccccc} \text{A} & \text{B} & \text{C} & \text{E} & \text{D} \\ \hline a_1 & b_1 & 1 & 1 & d_1 \\ a_2 & b_2 & 2 & 1 & d_1 \\ a_2 & b_2 & 2 & 2 & d_2 \\ a_3 & b_3 & 3 & 1 & d_1 \\ a_3 & b_3 & 3 & 2 & d_2 \end{array}$$

Chú ý:

-Hai quan hệ muốn kết nối được thì miền thuộc tính kết nối A của quan hệ R phải so sánh được với miền thuộc tính B của quan hệ S

-Nếu T' = R x S mà T = R ⋈ S thì T ⊆ T'. Như vậy phép kết nối có thể coi là phép chọn của phép tích Đề các.

-Nếu phép kết nối θ là “ = ” thì gọi là kết nối bằng. Nếu kết nối bằng qua hai thuộc tính cùng tên và một trong hai thuộc tính được loại bỏ thì gọi là kết nối tự nhiên, ký hiệu “*”

Ví dụ 2.16: Cho hai quan hệ R và S có dạng sau:

R	A	B	C
	a ₁	b ₁	1
	a ₂	b ₂	2
	a ₃	b ₃	3

S	C	D
	1	d ₁
	2	d ₂

Gọi F là biểu thức kết nối tự nhiên giữa R và S, (F có dạng: R.C=S.C)

Vậy phép kết nối tự nhiên giữa R và S là:

R*S=	A	B	C	D
	a ₁	b ₁	1	d ₁
	a ₂	b ₂	2	d ₂

(8) Phép chia

Cho R là quan hệ xác định trên tập thuộc tính U, với $U=\{A_1,A_2,\dots,A_n\}$ và quan hệ S xác định trên tập thuộc tính V với $V=\{B_1,B_2,\dots,B_m\}$ sao cho $n > m$, $S \neq \emptyset$ và $U \subset V$. Phép chia $R \div S$ là một quan hệ P(M) có dạng sau:

$$P(M)=R \div S = \{t.M \text{ với } t \in R, (t.M) \times S \subseteq R \text{ và } M=U-V\}$$

Ví dụ 2.17: Cho hai quan hệ R và S có dạng sau:

R	A	B	C	D
	a	b	c	d
	a	b	e	f
	b	c	e	f
	c	d	c	d
	c	d	e	f
	a	b	d	e

S	C	D
	c	d
	e	f

Ta có phép chia của R và S là:

R ÷ S=	A	B
	a	b
	c	d

Nhận xét:

- Nếu $P \times S = R$ thì phép chia không dư
- Nếu $P \times S \subset R$ thì phép chia có dư

(9) Một số hàm tiện ích:

1. Hàm SUM(R,A) cho tổng cá giá trị số trong cột A của R
 $SUM(R,A) = \Sigma \{t.A \mid t \in R\}$
2. Hàm AVG(R,A)
3. Hàm MAX(R,A)
4. Hàm (MIN(R,A)

2.4.3 Một số ví dụ

Cho cơ sở dữ liệu cung cấp hàng gồm các bảng dữ liệu sau:

Bảng Công Ty (CONGTY) gồm các thuộc tính: Mã công ty (MaCongTy), Tên công ty (TenCongTy), Ngân sách (NganSach), Địa chỉ (DiaChi).

Bảng Hàng Hoá (HANGHOA) gồm các thuộc tính: Mã hàng (MaHang), Tên hàng (TenHang), Màu sắc (Mau), Đơn vị tính (DonViTinh).

Bảng Cung Cấp hàng (CUNGCAP) gồm các thuộc tính: MaCongTy, MaHang, Số lượng (SoLuong), Đơn giá (DonGia).

Hãy viết biểu thức đại số quan hệ để thực hiện các câu hỏi sau:

- Cho biết danh sách các mặt hàng màu đỏ

$$\sigma_{\text{Mau} = \text{"Đỏ"}}(\text{HANGHOA})$$

- Cho biết mã các công ty cung cấp mặt hàng H1

$$\Pi_{\text{MaCongTy}} (\sigma_{\text{MaHang} = \text{"H1"}}(\text{CUNGCAP}))$$

- Cho biết tên các công ty cung cấp mặt hàng H1

$$\Pi_{\text{TenCongTy}} ((\sigma_{\text{MaHang} = \text{"H1"}}(\text{CUNGCAP})) *_{\text{MaCongTy}} \text{CONGTY})$$

- Cho biết những công ty cung cấp cả hai mặt hàng H1 và H2

$$\Pi_{\text{TenCongTy}} [(\text{CONGTY} *_{\text{MaCongTy}} (\sigma_{\text{MaHang} = \text{"H1"}}(\text{CUNGCAP})) \cap_{\text{MaCongTy}} ((\text{CONGTY} *_{\text{MaCongTy}} (\sigma_{\text{MaHang} = \text{"H2"}}(\text{CUNGCAP})))$$

Câu hỏi tối ưu hơn:

$$\Pi_{\text{TenCongTy}} \{ \text{CONGTY} * [\Pi_{\text{MaCongTy}} (\sigma_{\text{MaHang}=\text{'H1'}}(\text{CUNGCAP})) \cap \Pi_{\text{MaCongTy}} (\sigma_{\text{MaHang}=\text{'H2'}}(\text{CUNGCAP}))] \}$$

- Cho biết tên các công ty cung cấp ít nhất một mặt hàng màu đỏ

$$\Pi_{\text{TenCongTy}}(\text{CONGTY} * \text{CUNGCAP} * (\sigma_{\text{màu}=\text{'Đỏ'}}(\text{HANGHOA})))$$

- Cho biết tên những công ty cung cấp tất cả các mặt hàng

$$\Pi_{\text{TenCongTy}} \{ \text{CONGTY} * [\Pi_{\text{MaCongTy, MaHang}} \text{CUNGCAP}] \div \Pi_{\text{MaHang}}(\text{HANGHOA}) \}$$

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 2

- Định nghĩa quan hệ. Cho ví dụ minh họa.
- Định nghĩa khoá của một quan hệ. Cho ví dụ minh họa.
- Trình bày các phép toán của đại số quan hệ. Cho các ví dụ minh họa.
- Cho các quan hệ R, S và P có dạng sau:

R	A	B	C
	a ₁	b ₁	c ₁
	a ₂	b ₂	c ₂
	a ₃	b ₃	c ₃

S	D	E	F
	d ₁	e ₁	f ₁
	d ₂	e ₂	f ₂
	d ₃	e ₃	f ₃

P	A	B	D
	a ₁	b ₁	d ₁
	a ₂	b ₂	d ₂
	a ₁	b ₃	d ₃
	a ₂	b ₁	d ₂
	a ₂	b ₂	d ₁
	a ₃	b ₃	d ₃
	a ₂	b ₃	d ₃

Hãy tính giá trị của các biểu thức đại số quan hệ sau:

- $\Pi_{(AB)}(P)$
- $R * \Pi_{(AB)}(P)$
- $\sigma_F(R * \Pi_{(AB)}(P))$ với F có dạng $D = 'd_1'$

a. $d) R * S * T$

b. $e) \Pi_{CE}(\sigma_{C='c_2' \wedge F='f_2'}(R \times S))$

- Cho các quan hệ R, S và P có dạng sau:

R	A	B	C
---	---	---	---

S	D	E
---	---	---

P	A	B	D	H
---	---	---	---	---

a ₁	b ₁	c ₁
a ₂	b ₂	c ₂
a ₃	b ₃	c ₃

d ₁	e ₁
d ₂	e ₂
d ₃	e ₃

a ₁	b ₁	d ₁	h ₁
a ₂	b ₂	d ₁	h ₂
a ₂	b ₂	d ₂	h ₃
a ₂	b ₂	d ₃	h ₄

Hãy tính kết quả của các biểu thức đại số quan hệ sau:

a) $S \times P$

1. b) $\sigma_{S,D=P,D} (S \times P)$

2. c) $R^*(\sigma_{S,D=P,D} (S \times P))$

3. d) R^*S^*P

6. Cho quan hệ R, S và P có dạng sau:

R	A	B	C
	a ₁	b ₁	c ₁
	a ₂	b ₂	c ₂

S	D	E
	1	e ₁
	2	e ₂
	3	e ₃

P	B	D
	b ₁	1
	b ₂	2
	b ₂	2

Hãy tính kết quả của các phép toán đại số quan hệ sau:

- a) S^*P
- b) R^*S^*P
- c) $\sigma_{D=2} (S^*P)$
- d) $R * (\sigma_{D=2} (S^*P))$

7. Cho các quan hệ R, S và Q có dạng sau:

R	A	B	C
	a ₂	b ₂	c ₁
	a ₁	b ₂	c ₁
	a ₂	b ₃	c ₂

S	B	C	D
	b ₂	C ₁	d ₁
	b ₂	C ₁	d ₂
	b ₁	C ₂	d ₃

Q	D	E	F
	d ₁	e ₁	f ₁
	d ₂	e ₁	f ₂
	d ₂	e ₂	f ₃

Hãy tính kết quả của các phép toán đại số quan hệ sau:

- $\delta_{A=a_2}(R^*S)$
- $\Pi_{ABC}(R^*S^*Q)$
- $\Pi_{BC}(R) \cup \Pi_{BC}(S)$
- e. $R \times S$
- f. $R * \delta_{B=b_1}(S)$
- g. $\Pi_B(\delta_{B=b_2}(R^*S))$

$$\blacksquare \Pi_{BC}(R) \cap \Pi_{BC}(S)$$

$$h. \delta_{D=d1}(\Pi_{BCD}(S*Q))$$

8. Cho các quan hệ LOPHOC, SINHVIEN, DIEMTHI và MONHOC lần lượt như sau:

+Bảng 2.2: Quan hệ LOPHOC

MALOP	TENLOP
L01	ĐHCQK2C
L02	ĐHCQK2C
L03	ĐHCQK2B
L04	ĐHCQK2B

+ Bảng 2.3 Quan hệ SINHVIEN

MASV	HOTEN	GIOITINH	DIACHI	MALOP
CQK21001	Lê Hồng Vân	1	Thái Nguyên	L01
CQK21002	Nguyễn Văn Hà	1	Bắc giang	L01
CQK21003	Hoàng Thị Gấm	0	Hà Nội	L02
CQK21004	Lê Thị Thao	0	Thái Nguyên	L02

+Bảng 2.4: Quan hệ MONHOC

MAMH	TENMH	SOTINCHI
M01	Pascal	3
M02	Ngôn ngữ C	3
M03	Cơ sở dữ liệu	2

+ Bảng 2.5: Quan hệ DIEMTHI

MASV	MAMH	KYHOC	DIEMLAN1	DIEMLAN2
CQK21001	M01	1	4	7
CQK21001	M02	3	8	
CQK21001	M03	3	4	6
CQK21003	M01	1	8	
CQK21004	M02	3	2	6

Hãy viết các biểu thức đại số quan hệ để thực hiện các yêu cầu sau:

- c. Cho biết mã sinh viên, họ tên, giới tính của các sinh viên có địa chỉ tại Thái Nguyên?
- d. Cho biết mã sinh viên, họ tên, địa chỉ của các sinh viên có giới tính bằng 1?
- e. Cho biết mã và tên của các môn học có số tín chỉ bằng 3?
- f. Cho biết mã của các môn học có số tín chỉ ≤ 2 ?
- g. Cho biết mã sinh viên, họ tên, điểm thi lần 1 của các sinh viên đã học môn học có mã là 'M01' trong học kỳ 1?
- h. Cho biết mã sinh viên, họ tên, điểm thi lần 1 của các sinh viên đã học môn học có tên môn là 'cơ sở dữ liệu'?
- i. Cho biết mã của các sinh viên đã tích lũy được tất cả các môn học có ?
- j. Cho biết mã của các môn học chưa có sinh viên nào đăng ký học?

- k. Cho biết mã, họ tên, mã lớp của các sinh viên phải thi lại môn học có mã môn là 'M01'?
- l. Cho biết mã, họ tên, mã lớp của các sinh viên phải học lại môn học có tên môn là 'Ngôn ngữ C'?
- m. Cho biết mã, họ tên, mã lớp của các sinh viên phải thi lại môn học có tên môn là 'Pascal'?
- n. Cho biết mã sinh viên, mã môn học, điểm thi lần 1 của các sinh viên học trong học kỳ 3?

Chương 3

LÝ THUYẾT THIẾT KẾ CƠ SỞ DỮ LIỆU

3.1. Giới thiệu

3.1.1. Vấn đề thiết kế cơ sở dữ liệu

Một cơ sở dữ liệu quan hệ gồm tập các quan hệ. Muốn xây dựng một cơ sở dữ liệu quan hệ cần xác định trong cơ sở dữ liệu đó có những quan hệ gì, mỗi quan hệ có những thuộc tính nào, sự liên kết giữa các quan hệ như thế nào?...

Từ cơ sở phân tích chúng ta mới xây dựng nên sơ đồ thực thể liên kết, xác định các quan hệ và các liên kết cần thiết, chỉnh sửa chuẩn hoá các quan hệ trong hệ thống cơ sở dữ liệu

Bước cuối cùng là nhập dữ liệu theo dõi bảo trì cập nhật, hoàn thiện các quan hệ, các liên kết... trong hệ thống theo yêu cầu của người dùng

3.1.2 Bài toán ví dụ

Giả sử một cửa hàng bán buôn/lẻ các nhân viên mở sổ theo dõi việc bán hàng hàng ngày là một bảng (quan hệ) như sau:

Bảng 3.1: Sổ theo dõi việc bán hàng

Ngày	Số HD	Mã KH	Tên KH	Địa chỉ	Mã hàng	Tên hàng	Đơn giá	SL	Thành tiền
15/01/09	HD01	1	Anh	TN	A1	Xe đạp	800	3	2.400
15/01/09	HD01	1	Anh	TN	A2	Xe máy	1.200	2	2.400
17/01/09	HD02	2	Hùng	BG	A2	Xe máy	1.200	1	1.200
18/01/09	HD03	3	Hương	TB	A1	Xe đạp	800	2	1.600
18/01/09	HD03	3	Hương	TB	A2	Xe máy	1.200	4	4.800

Nhận xét: Quan hệ trên được thiết kế chưa tối ưu vì tồn tại một số dị thường về dữ liệu, cụ thể như:

-Dư thừa dữ liệu (Redundancy): Thông tin về khách hàng và hàng hoá bị lặp lại nhiều lần. Nếu khách hàng có mã 1 mua 15 mặt hàng thì thông tin về khách hàng này bị lặp lại 15 lần, tương tự đối với mặt hàng nếu mặt hàng có mã A1, nếu có 2000 khách hàng mua thì thông tin về mặt hàng đó cũng lặp lại 2000 lần

-Không nhất quán (Inconsistency): Là hệ quả của dư thừa dữ liệu. Giả sử sửa bản ghi thứ nhất, tên khách hàng được chữa thành An thì dữ liệu này lại không nhất quán với bản ghi thứ 2 (vẫn có tên là Anh).

-Dị thường khi thêm bộ (Insertion anomalies): Nếu muốn thêm thông tin về một mặt hàng mới nhập (chưa bán cho bất kỳ khách nào) vào quan hệ thì không được vì khoá chính của quan hệ trên gồm 2 thuộc tính Số hoá đơn, Mã hàng.

-Dị thường khi xoá bộ (Deletion anomalies): Giả sử muốn xoá thông tin về mặt hàng có mã là A1 thì ta phải rờ tất cả các dòng trong bảng có liên quan đến mặt hàng này để xoá, ngược lại thông tin về mặt hàng đó vẫn tồn tại.

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

Vậy ta tách quan hệ trên thành 4 quan hệ sau:

Bảng 3.2: Chứa thông tin về hàng hoá

Mã hàng	Tên hàng	Đơn giá
A1	Xe đạp	800000
A2	Xe máy	12000000

Bảng 3.3 Chứa thông tin về khách hàng

Mã khách	Tên khách	Địa chỉ
1	Anh	TN
2	Hùng	BG
3	Hương	TB

Bảng 3.4: Chứa thông tin về hoá đơn bán hàng

Số hoá đơn	Ngày	Mã khách
HD01	15/01/09	1
HD02	17/01/09	2
HD03	18/01/09	2

Bảng 3.5 : Chứa thông tin về chi tiết hoá đơn bán hàng

Số hoá đơn	Mã hàng	Số lượng
HD01	A1	3
HD01	A2	2
HD02	A2	1
HD03	A1	2
HD03	A2	4

Với cách tổ chức này ta thấy:

- Cơ sở dữ liệu gồm 4 bảng.
- Trong mỗi quan hệ không có sự dư thừa dữ liệu.

3.1.3. Kết luận

Cách tổ chức dữ liệu thứ hai (tách thành 4 quan hệ) tốt hơn thuận lợi hơn cho việc áp dụng máy tính vào xử lý, khắc phục những dị thường về dữ liệu khi cập nhật, sửa chữa dữ liệu như:

- Dư thừa dữ liệu
- Không nhất quán về dữ liệu....

Cơ sở để tách các quan hệ dựa trên sự phụ thuộc giữa các thuộc tính (gọi là phụ thuộc hàm) nghĩa là từ thuộc tính này có thể suy ra thuộc tính kia:

Ví dụ 3.1: Từ mã hàng ta có thể suy ra tên hàng

Mã hàng là “A1” thì “tên hàng” phải là xe đạp

Mã hàng là “A2” thì “tên hàng” phải là xe máy

Việc tách các quan hệ thành các quan hệ con ta gọi là phép chuẩn hoá

3.2. Sơ đồ quan hệ

3.2.1 Phụ thuộc hàm (Functional Dependencies)

Cho quan hệ $R(U)$; X, Y là 2 tập thuộc tính ($X, Y \subseteq U$) và một PTH $f: X \rightarrow Y$. Ta nói quan hệ R thỏa PTH f và viết $R(f)$ nếu với mọi 2 bộ bất kỳ $t_i, t_j \in R$ giống nhau trên X thì chúng cũng giống nhau trên Y . Hay ta viết:

$$R(X \rightarrow Y) \Leftrightarrow (\forall u, v \in R): u.X = v.X \Rightarrow u.Y = v.Y.$$

Trong đó u, v là hai bộ bất kỳ thuộc quan hệ R .

Nếu $f: X \rightarrow Y$ là một phụ thuộc hàm xác định trên $R(U)$ thì ta nói rằng tập thuộc tính Y phụ thuộc hàm vào tập thuộc tính X , (hay tập thuộc tính X xác định hàm tập thuộc tính Y).

Nếu Y không phụ thuộc hàm vào X ta có thể viết $X! \rightarrow Y$

Ví dụ 3.2: Cho bảng (quan hệ) SINH VIÊN sau:

Bảng 3.6: Chứa thông tin về sinh viên

Mã SinhViên	HỌTÊN	GiớiTính	Ngày Sinh	Quê Quán
SV01	Lan	NỮ	14/07/86	Hà Nội
SV02	Mai	NỮ	02/05/87	Thái Nguyên
SV04	Anh	Nam	12/05/85	Nam Định
SV05	Hoa	NỮ	20/01/86	Hà Nội

- Ký hiệu một phụ thuộc hàm là f . Ký hiệu một tập phụ thuộc hàm là F . Ví dụ trong bảng 3.6 ta có các phụ thuộc hàm sau:

- Tên sinh viên phụ thuộc vào mã sinh viên ($MãSinhViên \rightarrow HỌTÊN$)

- Quê quán phụ thuộc hàm vào mã sinh viên ($MãSinhViên \rightarrow QuêQuán$)

- Giới tính phụ thuộc hàm vào mã sinh viên ($MãSinhViên \rightarrow GiớiTính$)

- Ngày Sinh phụ thuộc hàm vào mã sinh viên ($MãSinhViên \rightarrow NgàySinh$)

Vậy ta có tập phụ thuộc hàm:

$$F = \{MãSinhViên \rightarrow HỌTÊN; MãSinhViên \rightarrow QuêQuán; MãSinhViên \rightarrow GiớiTính; MãSinhViên \rightarrow NgàySinh\}$$

Ghi chú:

- Phụ thuộc hàm là cơ sở cho việc chuẩn hoá lược đồ quan hệ.
- Phụ thuộc hàm là những ràng buộc dữ liệu được suy ra từ ý nghĩa và các mối liên quan giữa các thuộc tính.

3.2.2. Lược đồ quan hệ (sơ đồ quan hệ)

Lược đồ quan hệ r là một cặp gồm hai thành phần (U, F) trong đó U là tập hữu hạn các thuộc tính, F là tập các phụ thuộc hàm xác định trên U .

Ký hiệu là: $r(U, F)$

Ví dụ 3.3: Cho lược đồ quan hệ $r(U, F)$, với $U = \{A, B, C, D, E\}$

và $F = \{A \rightarrow BC, B \rightarrow D, AD \rightarrow E\}$

Ghi chú:

- Thể hiện của lược đồ quan hệ là quan hệ (bảng dữ liệu)
- Quan hệ luôn xác định trên một lược đồ quan hệ
- Thể hiện của một lược đồ quan hệ có thể khác nhau tại mỗi thời điểm
- Một lược đồ quan hệ có thể tương đương với một tập lược đồ quan hệ nhỏ hơn nhưng có cấu trúc tốt hơn trong việc áp dụng các thao tác dữ liệu.

3.3 Hệ tiên đề cho tập phụ thuộc hàm

3.3.1. Đặt vấn đề

Ta thấy với các bài toán quản lý khác nhau thì ta phải làm việc với các loại dữ liệu khác nhau, như vậy sẽ không có một phương pháp tổng quát cho mọi loại dữ liệu. Hay nói cách khác sẽ không có một lý thuyết mà có thể áp dụng cho mọi cơ sở dữ liệu. Điều đó dẫn đến bài toán tổ chức cơ sở dữ liệu chỉ là một bài toán thủ công không thể áp dụng các công cụ toán học và quá trình xử lý trên máy tính được.

Từ đó người ta tìm một giải pháp sao cho có thể khái quát hoá các cơ sở dữ liệu bằng mô hình toán học và có thể áp dụng được các công cụ toán học. Trong cơ sở dữ liệu khái quát đó, các thuật toán xử lý không phụ thuộc vào ý nghĩa của các thuộc tính cụ thể mà chỉ phụ thuộc vào các ràng buộc đã xác định qua tập thuộc tính và tập phụ thuộc hàm.

Ví dụ 3.4: Ta có lược đồ quan hệ $r(U, F)$ với U là tập hữu hạn các thuộc tính $U = \{A, B, C\}$, F là tập các PTH $F = \{A \rightarrow BC\}$

Ta có thể coi A là số báo danh; B là tên; C là tuổi

Cũng có thể coi A là tên hàng; B đơn giá; C là khối lượng

Dù tên cụ thể của A, B, C là gì thì tập U và F cũng vẫn đúng không phụ thuộc vào tên cụ thể của các thuộc tính.

Từ vấn đề trên Armstrong đã nghiên cứu và đưa ra mô hình bài toán khái quát với các tiên đề áp dụng cho mọi cơ sở dữ liệu

3.3.2. Hệ tiên đề Armstrong

Cho lược đồ quan hệ $r(U, F)$ với $U = \{A_1, A_2, \dots, A_n\}$ là tập các thuộc tính và F là tập PTH. Giả sử $X, Y, Z \in U$, ta có hệ tiên đề Armstrong sau:

1. Tiên đề phản xạ

Nếu $Y \subseteq X$ thì $X \rightarrow Y$ (Mọi tập con của X thì đều phụ thuộc hàm vào X)

2. Tiên đề tăng trưởng

Nếu $X \rightarrow Y$ và $Z \in U$ thì $XZ \rightarrow YZ$

3. Tiên đề bắc cầu

Nếu $X \rightarrow Y$ và $Y \rightarrow Z$ thì $X \rightarrow Z$

Từ các tiên đề ta có các tính chất trên sơ đồ quan hệ $r(U, F)$; $X, Y, Z, W \subseteq U$

1. Tính phản xạ chặt

Nếu $X \rightarrow X$

2. Tính tự bắc cầu:

Nếu $X \rightarrow Y$ và $YZ \rightarrow W$ thì $XZ \rightarrow W$

3. Tính mở rộng về trái và thu hẹp về phải

Nếu $X \rightarrow Y$ thì $XZ \rightarrow YW$

4. Tính cộng đầy đủ

Nếu $X \rightarrow Y$ và $Z \rightarrow W$ thì $XZ \rightarrow YW$

5. Tính mở rộng về trái

Nếu $X \rightarrow Y$ thì $XZ \rightarrow Y$

6. Tính cộng ở vế phải (Luật hợp)

Nếu $X \rightarrow Y$ và $X \rightarrow Z$ thì $X \rightarrow YZ$

7. Tính bộ phận ở vế phải (Luật tách)

Nếu $X \rightarrow YZ$ thì $X \rightarrow Y$ và $X \rightarrow Z$

8. Tính tích luỹ

Nếu $X \rightarrow YZ$, $Z \rightarrow W$ thì $X \rightarrow YZW$

Khi giải quyết các bài toán ta có thể áp dụng các tiên đề Amstrong và các tính chất trên.

3.3.3. Bài toán áp dụng

Cho lược đồ quan hệ $R(U,F)$ với

$U = \{A, B, C\}$

$F = \{AB \rightarrow C; C \rightarrow A\}$. Chứng minh $BC \rightarrow ABC$

Giải:

Từ $C \rightarrow A$ (gt)

Theo tiên đề tăng trưởng thêm vào hai vế B ta có: $BC \rightarrow AB$ (1)

Từ $AB \rightarrow C$ (gt)

Thêm AB vào hai vế ta có: $AB \rightarrow ABC$ (2)

Từ (1) và (2) theo tiên đề bắc cầu ta có:

$BC \rightarrow ABC$ đó là điều phải chứng minh

3.3.4. Kiểm tra tính đúng đắn của hệ tiên đề Amstrong

Giả sử có bảng DS cán bộ: MãCB, TênCB, MãLương, BậcLương

Trong đó: MãCB \rightarrow TênCB, MãLương, BậcLương

MãLương \rightarrow BậcLương

Mô hình hoá bằng các thuộc tính sau:

Cho lược đồ quan hệ $R(U,F)$. Trong đó

$U = \{A, B, C, D\}$

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

a). Kiểm tra tiên đề 1

Nếu đặt $X = AB$ rõ ràng $A \subseteq AB$

Với hai bộ bất kỳ t_1, t_2 ta đều có

$$\text{Nếu } t_1.[AB] = t_2.[AB]$$

$$\text{Thì } t_1.[A] = t_2.[A]$$

Hiển nhiên ta thấy $AB \rightarrow A$

b) Kiểm tra tiên đề 2

$$\text{Đặt } X = AB \text{ và } XC = ABC$$

$$\text{Đặt } Y = D \text{ và } YC = DC$$

Với hai bộ bất kỳ t_1, t_2 ta thấy

$$\text{Nếu } t_1.[ABC] = t_2.[ABC]$$

$$\text{Thì } t_1.[DC] = t_2.[DC]$$

Như vậy tiên đề thứ hai là đúng đắn

c). Kiểm tra tiên đề 3

Theo tiên đề 3 ta thấy $A \rightarrow C ; C \rightarrow D$ thì có thể suy ra $A \rightarrow D$

Với hai bộ bất kỳ t_1, t_2

$$\text{Nếu } t_1.[A] = t_2.[A]$$

$$\text{Thì } t_1.[D] = t_2.[D]$$

Vậy tiên đề này hoàn toàn đúng

3.4 Bao đóng của tập phụ thuộc hàm

Cho lược đồ quan hệ $r(U,F)$, Trong đó U là tập hữu hạn các thuộc tính và F là tập phụ thuộc hàm. X, Y là các tập thuộc tính ($X, Y \subseteq U$).

Nói rằng phụ thuộc hàm $X \rightarrow Y$ được suy dẫn logic từ F nếu quan hệ R xác định trên $r(U,F)$ thỏa mãn tất cả các phụ thuộc hàm của F thì cũng thỏa mãn phụ thuộc hàm $X \rightarrow Y$.

Bao đóng của tập phụ thuộc hàm F (kí hiệu là F^+) là tập tất cả các phụ thuộc hàm được suy dẫn logic là F

$$F^+ = \{f: X \rightarrow Y \mid X, Y \in U \text{ và } F \models f\}$$

Nếu có $F = F^+$ thì F là họ đầy đủ của các phụ thuộc hàm.

Ví dụ 3.5: Cho lược đồ quan hệ $r(U,F)$, với $U = \{A, C, B\}$ và $F = \{A \rightarrow B, B \rightarrow C\}$ ta có thể suy ra $A \rightarrow C$. Rõ ràng phụ thuộc hàm $A \rightarrow C$ được suy diễn ra từ F .

Ta có $F^+ = \{ A \rightarrow B, B \rightarrow C, A \rightarrow C \}$

3.5. Phép tách một quan hệ

3.5.1. Định nghĩa

Cho lược đồ quan hệ r xác định trên tập thuộc tính U và F là tập các phụ thuộc hàm. Phép tách lược đồ quan hệ $r(U, F)$ là việc thay thế lược đồ quan hệ $r(U, F)$ bằng các tập lược đồ $r_1(U_1), r_2(U_2), \dots, r_m(U_m)$, sao cho $U = U_1 \cup U_2 \cup \dots \cup U_m$ Trong đó $U_i \subseteq U$ và $r_i(U_i) = \Pi_{U_i}(R)$ với $i=1, \dots, m$

Ký hiệu phép tách của $r(U, F)$ là ρ . Vậy $\rho = \{r_1(U_1), r_2(U_2), \dots, r_m(U_m)\}$

Nói rằng ρ là phép tách – kết nối không mất mát thông tin đối với F nếu với mỗi quan hệ R xác định trên $r(U, F)$ thì $R = \Pi_{U_1}(R) * \Pi_{U_2}(R) * \Pi_{U_3}(R) * \dots * \Pi_{U_m}(R)$.

Định lý 3.1: Cho lược đồ quan hệ $r(U, F)$ nếu $\rho = \{R_1(U_1), R_2(U_2)\}$ là một phép tách của r thì ρ là phép tách không mất mát thông tin đối với F khi và chỉ khi:

$$U_1 \cap U_2 \rightarrow U_1 \setminus U_2$$

Hoặc $U_1 \cap U_2 \rightarrow U_2 \setminus U_1$

Chứng minh:

Ta phải chứng minh hai vấn đề:

$$-U_1 \cap U_2 \neq \phi$$

$$-Và R(U) = R(U_1) * R(U_2) \text{ theo tính chất trên}$$

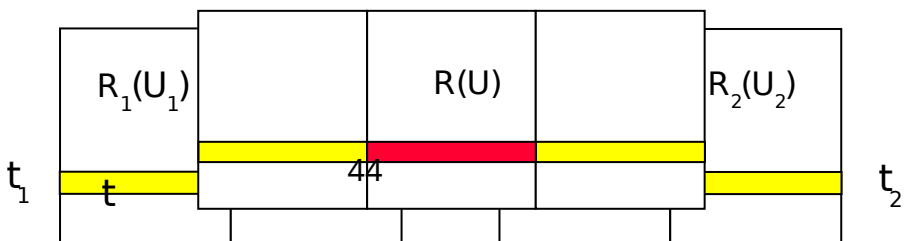
♦ Giả sử xét bảng dữ liệu sau:

Tên	Phách	Điểm
Nam	01	7
Bắc	02	6
Nam	03	4

$U = \{\text{tên, phách, điểm}\}$

Nếu tách $U_1 = \{\text{tên}\}; U_2 = \{\text{phách, điểm}\}$ Thì $U_1 \cap U_2 = \phi$

Rõ ràng ta thấy dữ liệu không còn chính xác. Minh họa bằng bảng sau ta thấy:



♦ Giả sử chọn bộ t nào đó thuộc R . Khi tách thành R_1, R_2 ta được t_1, t_2

Ta thấy $t = t_1 * t_2$ hay $R \subseteq R_1 * R_2$

Mặt khác $\forall t_1 \in R_1$; và $\forall t_2 \in R_2$ ta có:

$$t_1[U_1 \cap U_2] = t_2[U_1 \cap U_2]$$

Theo tính chất phép toán kết nối tự nhiên ta có:

$$t_1 * t_2 = t \quad \text{Hay } R_1 * R_2 \subseteq R$$

Như vậy ta có $R_1 * R_2 = R$

Định lý được chứng minh

Nhận xét: Nếu ta tách một lần được hai quan hệ, tách hai lần được 3 quan hệ vậy muốn tách m quan hệ phải tách $(m-1)$ lần.

3.5.3. Kiểm tra phép tách không mất mát thông tin

Cho lược đồ quan hệ $r(U, F)$, Trong đó, tập các thuộc tính $U = \{A_1, A_2, \dots, A_n\}$ và tập các phụ thuộc hàm F ; phép tách ρ . Hãy kiểm tra phép tách ρ :

$$\rho = (r_1, r_2, \dots, r_m) \text{ có mất mát thông tin không?}$$

Thuật toán 1:

Bước 1: Lập một bảng gồm có n cột, m hàng. Cột thứ j ứng với thuộc tính A_j hàng thứ i ứng với lược đồ r_i . Tại hàng i cột j điền ký hiệu a_j nếu $A_j \in r_i$; ngược lại điền ký hiệu b_{ij}

Bước 2: Đồng nhất giá trị trên bảng

Áp dụng quy trình thay thế đỏi trên bảng:

Xét các phụ thuộc hàm từ F : Giả sử xét PTH $X \rightarrow Y \in F$. Ta xét các hàng có giá trị bằng nhau tại thuộc tính X thì làm bằng giá trị tại thuộc tính Y giữa các hàng đó.

Chú ý: Khi làm bằng giá trị trên Y , nếu một trong các giá trị là a_j thì ưu tiên làm bằng về ký hiệu a_j ngược lại làm bằng chúng bằng một trong các ký hiệu b_{ij} .

Tiếp tục áp dụng các phụ thuộc hàm có trong F (kể cả việc lập lại các phụ thuộc hàm đã được áp dụng) cho tới khi không áp dụng được nữa hay trên bảng kết quả đã xuất hiện ít nhất một hàng có đủ các ký hiệu ($a_1, a_2, a_3, \dots a_n$) thì dừng và chuyển sang **bước 3**.

Bước 3: Xét bảng kết quả nếu xuất hiện ít nhất một hàng gồm các ký hiệu $a_1, a_2, a_3, \dots a_n$ thì ta kết luận phép tách ρ là không mất mát thông tin (bảo toàn thông tin). Ngược lại phép tách ρ không bảo toàn thông tin (mất mát thông tin).

Ví dụ 3.6:

Cho quan hệ: HOCSINH (SBD, TEN, DTOAN, DTIN)

Với các phụ thuộc hàm: $SBD \rightarrow TEN$; $SBD \rightarrow \{DTOAN, DTIN\}$

Tách thành hai quan hệ:

HS1(SBD, TEN)

HS2(SBD, DTOAN, DTIN)

+ Lập bảng kiểm tra như sau:

	SBD	TEN	DTOAN	DTIN
HS1	a_1	a_2	b_{13}	b_{14}
HS2	a_1	b_{22}	a_3	a_4

+ Làm bằng các giá trị

Ta thấy dòng 2 tại thuộc tính TEN có giá trị là a_2 và b_{22} mà SBD của hai dòng này có giá trị là a_1 . Vậy theo phụ thuộc hàm $SBD \rightarrow TEN$ nên ta thay giá trị b_{22} của thuộc tính TEN tại dòng 2 là a_2

Ta có bảng:

	SBD	TEN	DTOAN	DTIN
HS1	a_1	a_2	b_{13}	b_{14}
HS2	a_1	a_2	a_3	a_4

Vậy bảng có dòng 2 toàn là giá trị a_j ($j = 1..4$), nên phép tách trên là không mất mát thông tin

3.6 Chuẩn hoá lược đồ quan hệ

Khi thiết kế một lược đồ quan hệ phải tuân theo một số nguyên tắc để khi thao tác trên cơ sở dữ liệu không dẫn đến sự dị thường về dữ liệu. Công việc thiết kế dữ liệu theo một dạng chuẩn nào đó gọi là chuẩn hoá dữ liệu.

3.6.1. Các dạng chuẩn trong lược đồ quan hệ

Do việc cập nhật dữ liệu (qua phép chèn, loại bỏ và sửa đổi) gây nên những dị thường, cho nên các quan hệ nhất thiết phải được biến đổi thành các dạng phù hợp, quá trình đó gọi là quá trình chuẩn hoá.

Quan hệ được chuẩn hoá là quan hệ trong đó mỗi miền của một thuộc tính chỉ chứa những giá trị nguyên tố, tức là không phân nhỏ được nữa và do đó mỗi giá trị trong quan hệ cũng là nguyên tố.

Quan hệ có chứa các miền giá trị là không nguyên tố gọi là quan hệ chưa chuẩn hoá.

Chuẩn hoá dữ liệu là quá trình phân rã lược đồ quan hệ chưa chuẩn hoá (có dạng chuẩn thấp) thành các lược đồ quan hệ nhỏ hơn nhưng ở dạng chuẩn cao hơn (có cấu trúc tốt hơn) và không làm mất mát thông tin.

Có các dạng chuẩn sau: 1NF, 2NF, 3NF, BCNF

3.6.2. Một số định nghĩa

a) Thuộc tính khoá

Cho lược đồ quan hệ $r(U)$ với tập thuộc tính U , $A_i \in U$; A gọi là thuộc tính khoá của R nếu tồn tại $K \subseteq U$,

Nếu $A \in K$ mà K là khoá thì A là thuộc tính khoá (A có mặt trong ít nhất một tập khoá của $r(U,F)$).

Ngược lại A không có mặt trong bất kỳ tập khoá nào của $r(U,F)$ thì A là thuộc tính không khoá.

b) Phụ thuộc hàm đầy đủ

Cho $R(U)$ $X, Y \subseteq U$, Y gọi là phụ thuộc hàm đầy đủ vào X , Nếu $X \rightarrow Y$ và $\forall A \in X; (X - \{A\}) \not\rightarrow Y$

Phụ thuộc hàm đầy đủ ký hiệu là $X \twoheadrightarrow Y$

c) Phụ thuộc hàm bắc cầu

Cho $R(U)$ $X, Y \subseteq U$ Y gọi là phụ thuộc hàm bắc cầu vào X

Nếu $\exists Z \subseteq U: Y-Z \neq \emptyset, X \rightarrow Z, Z \not\rightarrow X, Z \rightarrow Y, Y \not\rightarrow X$

Phụ thuộc hàm bậc cầu ký hiệu là $X\% \rightarrow Y$

3.6.3. **Dạng chuẩn 1NF (1st Normal Form)**

Lược đồ quan hệ r gọi là ở dạng chuẩn một (1NF) nếu toàn bộ các miền trị có mặt trong quan hệ R đều chỉ chứa các giá trị nguyên tố.

Ví dụ 3.7: Cho thông tin của bảng đăng ký học của sinh viên:

Bảng 3.7: Bảng đăng ký học của sinh viên

Mã sinh viên	Tên SV	Ngày sinh	Kỳ học	Đăng ký học		
				Mã môn	Tên môn	Số TC
SV01	Hùng	09/12/1990	1	M01	CSDL	2
				M02	Tin ĐC	3
SV02	Trương	07/09/1990	1	M02	Tin ĐC	3
				M03	Tiếng Anh	3

Quan hệ này không phải là dạng chuẩn 1NF vì giá trị trong thuộc tính **Đăng ký học** không phải là nguyên tố.

3.6.4 **Dạng chuẩn 2NF (2nd Normal Form)**

Lược đồ quan hệ r gọi là dạng chuẩn hai (2NF) nếu r ở dạng chuẩn 1NF và mọi thuộc tính không khoá của r đều phụ thuộc hàm đầy đủ vào khoá.

3.6.5 **Dạng chuẩn 3NF (3rd Normal Form)**

Lược đồ quan hệ r gọi là dạng chuẩn 3NF nếu đã ở dạng chuẩn 2 và mọi thuộc tính không khoá của r không phụ thuộc hàm bậc cầu vào khoá.

Ví dụ 3.8 : Xét lược đồ quan hệ: $r(\text{SAIP})$; Trong đó $F = \{SI \rightarrow P; S \rightarrow A\}$

Ta thấy r là dạng chuẩn 1

Xét dạng chuẩn 2: Ta có $S \rightarrow A$

và $SI \rightarrow S \Rightarrow SI \rightarrow A$

Thuộc tính A không phụ thuộc đầy đủ vào khoá của r là SI , như vậy r không phải là 2 NF

Ví dụ 3.9:

Xét lược đồ quan hệ: $r(\text{SIDM}); F = \{SI \rightarrow D; SD \rightarrow M\}$

-Ta thấy r là dạng chuẩn 2

-Xét dạng chuẩn 3:

$SI \rightarrow D \Rightarrow SI \rightarrow SD$; (theo luật tăng trưởng); Mà $SD \rightarrow M \Rightarrow SI \rightarrow M$

Vậy M phụ thuộc bắc cầu vào SI nên r không là 3 NF

3.6.6 Dạng chuẩn BCNF (Boye - Code)

Lược đồ quan hệ r gọi là dạng chuẩn BCNF nếu $X \rightarrow A$ thoả trên r , nếu A không thuộc X và X là khoá của r .

Ghi chú:

- Các lược đồ quan hệ ở dạng chuẩn 1NF, 2NF vẫn tồn tại các dị thường về dữ liệu.

- Trong một cơ sở dữ liệu tốt, các quan hệ phải được chuẩn hoá về 3NF hoặc BCNF.

- Một lược đồ quan hệ r ở dạng chuẩn BCNF thì r là 3NF

3.7. Các thuật toán

3.7.1 Bao đóng của tập thuộc tính

Cho lược đồ quan hệ $r(U, F)$, với $U = \{A_1, A_2, \dots, A_n\}$ và F là tập phụ thuộc hàm, X là một tập thuộc tính ($X \subseteq U$). Bao đóng của tập thuộc tính X đối với tập phụ thuộc hàm F (ký hiệu là X^+) là tập các thuộc tính có thể suy dẫn logic từ X qua các phụ thuộc hàm có trong F .

$$X^+ = \{ A \mid A \subseteq U; X \rightarrow A \in F^+ \}$$

Thuật toán 2: Tìm bao đóng của tập thuộc tính

Cho lược đồ quan hệ $r(U, F)$. Trong đó U là tập hữu hạn các thuộc tính và F là tập phụ thuộc hàm. Tìm bao đóng của X (X^+)

Tính liên tiếp các tập X_0, X_1, X_2, \dots theo phương pháp sau:

Bước 0: Đặt $X_0 = X$

Bước i: Lần lượt xét các phụ thuộc hàm của F

Tính $X_i = \{ X_{i-1} \cup \{A\} / \text{nếu } \exists Y \rightarrow Z \in F; A \in Z \text{ và } A \notin X_{i-1}; Y \subseteq X_{i-1}; \text{loại } Y \rightarrow Z \text{ khỏi } F \}$

Vì $X_1 \subseteq X_2 \subseteq \dots \subseteq U$ nên $\exists j$ sao cho $X_j = X_{j-1}$ (tập X không tăng nữa)

Đặt $X^+ = X_j$; Gọi X^+ là bao đóng của X

Mô tả thuật toán bằng ngôn ngữ giả Pascal:

```
Proc Closure;  
Input:  $r=(U,F)$ ; Tập thuộc tính  $X \subseteq U$   
Output:  $Y = X^+ = \{A \in U \mid X \rightarrow A \in F^+\}$   
Begin  
Y:= X  
Repeat  
    Z:=Y;  
    For each  $f \in L \rightarrow R$  in F do  
        if  $L \subseteq Y$  then  
            Y:=Y  $\cup$  R;  
        endif;  
    endfor;  
Until Y=Z;  
return Y;  
End;
```

Ví dụ 3.9:

Cho lược đồ quan hệ $r(U,F)$ với $U = \{A,B,C,D,E,G,H\}$
và $F = \{A \rightarrow BC; C \rightarrow B; D \rightarrow EH; AD \rightarrow G\}$. Tính $(AD)^+$?

Bài giải:

Đặt $X_1 = (AD)$

Chọn các phụ thuộc hàm có vế trái là A,D,AD có $A \rightarrow BC$ nên

$$X_2 = X_1 \cup B = AD \cup B = ABD$$

$$X_3 = X_2 \cup C = ADB \cup C = ABCD$$

Vì $D \rightarrow EH$ nên

$$X_4 = X_3 \cup E = ADBC \cup E = ABCDE$$

$$X_5 = X_4 \cup H = ADBCE \cup H = ABCDEH$$

Vì $AD \rightarrow G$ nên

$$X_6 = X_5 \cup G = ABCDEH \cup G = ABCDEHG$$

$$X_7 = X_6 = ABCDEHG$$

Kết luận: $(AD)^+ = ABCDEFG$

Bài toán thành viên: Cho lược đồ quan hệ $r(U,F)$ với U là tập hữu hạn các thuộc tính và F là tập phụ thuộc hàm, X, Y là hai tập thuộc tính ($X, Y \subseteq U$). Hỏi rằng $X \rightarrow Y$ có là thành viên của F hay không? (có nghĩa là $X \rightarrow Y \in F^+$ hay không?)

Để trả lời cho câu hỏi này ta có thể tính F^+ rồi xác định xem $X \rightarrow Y$ có thuộc F^+ hay không. Nhưng việc tính F^+ đòi hỏi thời gian và công sức. Tuy nhiên, thay vì tính F^+ chúng ta có thể sử dụng định lý sau:

Định lý 3.2: $X \rightarrow Y \in F^+$ Khi và chỉ khi $Y \subseteq X^+$

3.7.2. Phủ của tập các phụ thuộc hàm

Cho hai tập phụ thuộc hàm F và G cùng xác định trên tập thuộc tính U . Nói rằng F và G là tương đương nếu $F^+ = G^+$. Nếu F và G tương đương thì ta có thể nói F là một phủ của G (hoặc G là một phủ của F)

Ký hiệu: $F \equiv G$

Phủ không dư thừa: Cho tập phụ thuộc hàm F xác định trên tập thuộc tính U . F được gọi là phủ không dư thừa nếu:

Không tồn tại phụ thuộc hàm $X \rightarrow Y \in G$ mà $(G - \{X \rightarrow Y\}) \equiv G$

Ví dụ 3.10: Cho lược đồ quan hệ $r(U,F)$ với $F = \{A \rightarrow B; B \rightarrow C; A \rightarrow C\}$

Ta thấy $A \rightarrow C$ là thừa vì $F - \{A \rightarrow C\}$ tương đương với F . Vậy F là phủ dư thừa.

Phủ tối thiểu: Gọi tập các phụ thuộc hàm F là tối thiểu nếu thỏa mãn ba điều kiện:

(1) Mọi phụ thuộc hàm thuộc F đều có dạng: $\{X_i \rightarrow A_i \mid i = 1..m\}$ (nói cách khác về phải mỗi phụ thuộc hàm thuộc F chỉ có một thuộc tính).

(2) F là phủ không dư thừa :

Có nghĩa là không tồn tại PTH $X \rightarrow A \in F$ mà $F^+ = (F - \{X \rightarrow A\})^+$

(3) F không dư thừa thuộc tính nào ở vế trái, nói cách khác không tồn tại một phụ thuộc hàm $X \rightarrow A \in F; Z \subset X$ mà : $F^+ = (F - \{X \rightarrow A\} \cup \{Z \rightarrow A\})^+$

Định lý 3.3: Mọi phụ thuộc hàm F đều tương đương với một phủ tối thiểu F' .

Thuật toán 3: Cho lược đồ quan hệ $r(U,F)$ với U là tập hữu hạn các thuộc tính và F là tập phụ thuộc hàm. Tìm phủ tối thiểu của F .

Bước 1: Tách các PTH sao cho vế phải của mỗi PTH chỉ có một thuộc tính.

Giả sử xét phụ thuộc hàm $X \rightarrow Y$, với $Y = A_1A_2A_3\dots A_n$. Ta có thể tách thành các phụ thuộc hàm sau:

$$X \rightarrow A_1$$

$$X \rightarrow A_2$$

...

$$X \rightarrow A_m$$

Kết quả ta được F^1 tương đương với F

Bước 2: Loại bỏ các phụ thuộc hàm (PTH) dư thừa.

Giả sử có F_i có dạng $X_j \rightarrow A_j \mid j = 1, 2, \dots, m$

Đặt $F_0 = F^1$

$$F_i = \begin{cases} F_{i-1} \setminus \{X_j \rightarrow A_j\} & \text{nếu } F_{i-1} \setminus \{X_j \rightarrow A_j\} \text{ tương đương với } F_{i-1} \\ F_{i-1} & \text{nếu ngược lại} \end{cases}$$

Sau m lần ta được $F_m = F_{m-1}$

Đặt $F^2 = F_m$ tương đương với F^1 .

Bước 3: Loại bỏ các thuộc tính dư thừa bên trái của mỗi phụ thuộc hàm

Sau bước 2 có $F^2 = \{X_i \rightarrow A_j \mid \text{với } i = 1..n \text{ và } X_i \text{ có dạng } X_i = A_1, A_2, \dots, A_n$

-Đặt $X_0 = X_i$

$$X_j = \begin{cases} X_{j-1} \setminus \{A_j\} & \text{nếu } \{F_2 \setminus (X_{i-1} \rightarrow A_i) \cup (X_{i-1} \setminus A_j) \rightarrow A_i\} \text{ tương đương với } F^2 \\ X_{j-1} & \text{nếu ngược lại} \end{cases}$$

Lặp lại quy tắc trên n lần thì ta xét xong phụ thuộc hàm $X_i \rightarrow A_j$ (Có nghĩa là đã loại bỏ tất cả các thuộc tính dư thừa bên trái trong phụ thuộc hàm trên).

Sau bước này ta được F^3 tương đương với F^2 . F^3 là phủ tối thiểu của F

Ví dụ 3.11:

Cho lược đồ quan hệ $r(U, F)$ với $U = \{A, B, C, D, E, G, H, L\}$

và $F = \{A \rightarrow BC; C \rightarrow B; D \rightarrow EL; ADC \rightarrow G\}$. Tìm phủ tối thiểu của F ?

Bước 1: Tách các phụ thuộc hàm:

Ta có $F_1 = \{A \rightarrow C; C \rightarrow B; A \rightarrow B; D \rightarrow E; D \rightarrow L; ADC \rightarrow G\}$

Bước 2: Loại bỏ các phụ thuộc hàm dư thừa :

$A \rightarrow C$ không dư thừa vì $C \notin A^+ = AB$

$C \rightarrow B$ không dư thừa vì $B \notin C^+ = C$

$A \rightarrow B$ dư thừa vì có $B \in A^+ = ABC$

$D \rightarrow E$ không dư thừa vì có $E \notin D^+ = D$

$D \rightarrow L$ không dư thừa vì có $L \notin D^+ = D$

$ADC \rightarrow G$ không dư thừa vì có $G \notin (ADC)^+ = (ABCDEL)$

Ta có $F_2 = \{C \rightarrow B; A \rightarrow C; D \rightarrow E; D \rightarrow L; ADC \rightarrow G\}$

Bước 3: Loại bỏ các thuộc tính dư thừa ở vế trái:

Vì $A \rightarrow C$ nên phụ thuộc hàm $ADC \rightarrow G$ thừa thuộc tính C nên ta có $AD \rightarrow G$

Ta có $F_3 = \{C \rightarrow B; A \rightarrow C; D \rightarrow E; D \rightarrow F; AD \rightarrow G\}$

Kết luận: Phủ tối thiểu của F là $F_{tt} = F_3 = \{C \rightarrow B; A \rightarrow C; D \rightarrow E; D \rightarrow F; AD \rightarrow G\}$

Thuật toán 4: Cho hai tập phụ thuộc hàm F và G xác định trên tập thuộc tính U. Với F và G có dạng sau : $F = \{X_i \rightarrow Y_i \mid i = 1..m\}$; $G = \{X_j \rightarrow Y_j \mid j = 1..n\}$. Hãy kiểm tra F và G có tương đương với nhau hay không ?

Bước 1: Với $\forall i = 1..m$ kiểm tra xem $X_i \rightarrow Y_i \in F$ có thuộc G^+ hay không. Theo bài toán thành viên ta kiểm tra xem có thoả $Y_i \subseteq X_j^+$ nếu thoả thì $F^+ \subseteq G^+$

Bước 2: Với $\forall j = 1..n$ kiểm tra xem $X_j \rightarrow Y_j \in G$ có thuộc F^+ không. Theo bài toán thành viên ta kiểm tra xem có thoả $Y_j \subseteq X_i^+$ nếu thoả thì $G^+ \subseteq F^+$

Nếu thoả cả hai điều kiện trên thì $G^+ = F^+$ và ta nói F và G tương đương, ngược lại trong quá trình kiểm tra nếu tồn tại một phụ thuộc hàm không thoả mãn thì F và G là không tương đương nhau.

3.7.4. Khoá tối thiểu của sơ đồ quan hệ.

Cho lược đồ quan hệ $r(U, F)$, $K \subseteq U$. K được gọi là khoá tối thiểu của lược đồ quan hệ nếu:

$$(1) K^+ = U$$

$$(2) \forall A \in K; (K - \{A\})^+ \neq U$$

Hai điều kiện trên tương ứng với:

$$(3) K \rightarrow U$$

$$(4) \forall A \in K; (K - \{A\})^+ \neq U$$

Chú ý:

- K chỉ thỏa mãn điều kiện (1) thì K gọi là siêu khoá.

- Trong một số tài liệu thuật ngữ *khóa* được dùng theo nghĩa *siêu khóa* và thuật ngữ *khóa tối thiểu* dùng theo nghĩa *khóa*

- Một sơ đồ quan hệ có ít nhất một tập khóa. Gọi M là giao của các khóa:

$$M = \cup (L \setminus R)$$

- $L \setminus R$ là thuộc tính chỉ xuất hiện ở vế trái và không xuất hiện ở vế phải của các phụ thuộc hàm có trong F

- Nếu $M^+ = U$ thì r có một khóa duy nhất ngược lại $M^+ \neq U$ thì r có nhiều hơn một tập khóa.

Thuật toán 5: Tìm một khóa tối thiểu của lược đồ quan hệ

Cho lược đồ quan hệ r (U, F). với $U = \{A_1, A_2, \dots, A_n\}$ và F là tập phụ thuộc hàm.

Tìm khóa tối thiểu của r(U,F)

Bước 1: Đặt $K_0 = U$

$$K_{i-1} \setminus A_j \text{ nếu } (K_{i-1} \setminus A_j)^+ = U, j=1,2,\dots,n$$

$$\text{Bước } i: \text{ Tính } K_i = \begin{cases} K_{i-1} \setminus A_j & \text{nếu } (K_{i-1} \setminus A_j)^+ = U, j=1,2,\dots,n \\ K_{i-1} & \text{nếu ngược lại} \end{cases}$$

Lặp lại bước i n lần

Kết luận: Khóa tối thiểu của r(U,F) là K_i .

Mô tả thuật toán bằng ngôn ngữ giả PASCAL

Proc Key;

Input: Tập thuộc tính U; Tập F

Output: $K \subseteq U$ thỏa điều kiện

$$(1) K^+ = U$$

$$(2) \forall A \in K; (K - \{A\})^+ \neq U$$

Begin

$K := U;$

For each attribute A in U do

 if $A \in (K - A)^+$ then

$K := K - A;$

 endif;

endfor;

return K;
End;

Ví dụ 3.12: Cho lược đồ quan hệ $r(U,F)$ có:

$$U = \{A,B,C,D,E,L,G,H\}$$

$$F = \{ A \rightarrow BC; C \rightarrow B; D \rightarrow EL; ADC \rightarrow G \}. \text{ Tìm một khoá cho } r(U,F)?$$

Bài giải:

Bước 0: Đặt $K_0 = U = \{A,B,C,D,E,L,G,H\}$

Bước 1: $K_1 = K_0 \setminus A = \{A,B,C,D,E,L,G,H\}$ vì $(K_0 \setminus A)^+ \neq U$

Bước 2: $K_2 = K_1 \setminus B = \{A,C,D,E,L,G,H\}$ vì $(K_1 \setminus B)^+ = U$

Bước 3: $K_3 = K_2 \setminus C = \{A,D,E,L,G,H\}$ vì $(K_2 \setminus C)^+ = U$

Bước 4: $K_4 = K_3 \setminus D = \{A,D,E,L,G,H\}$ vì $(K_3 \setminus D)^+ \neq U$

Bước 5: $K_5 = K_4 \setminus E = \{A,D,L,G,H\}$ vì $(K_4 \setminus E)^+ = U$

Bước 6: $K_6 = K_5 \setminus L = \{A,D,G,H\}$ vì $(K_5 \setminus L)^+ = U$

Bước 7: $K_7 = K_6 \setminus G = \{A,D,H\}$ vì $(K_6 \setminus G)^+ = U$

Bước 8: $K_8 = K_7 \setminus H = \{A,D,H\}$ vì $(K_7 \setminus H)^+ \neq U$

Vậy khoá là $K = \{A,D,H\}$

Thuật toán 6: Tìm tất cả các khoá của lược đồ quan hệ

Cho lược đồ quan hệ $r(U,F)$ với U là tập hữu hạn các thuộc tính và F là tập phụ thuộc hàm. Tìm tất cả các khoá cho $r(U,F)$?

Phương pháp: Thực hiện theo các bước sau:

Bước 1: Tạo các tập thuộc tính nguồn (TN) và tập thuộc tính trung gian (TG).

TN- Chứa các thuộc tính chỉ xuất hiện ở vế trái và không xuất hiện ở vế phải của các phụ thuộc hàm có trong F .

TG- Chứa các thuộc tính vừa xuất hiện ở vế trái vừa xuất hiện ở vế phải của các phụ thuộc hàm có trong F .

Bước 2:

Nếu $TG = \emptyset$ thì $r(U,F)$ chỉ có một khoá K

$$K = TN$$

kết thúc thuật toán

Ngược lại

Qua bước 3

Bước 3: Tìm tất cả các tập con X_i của tập trung gian TG.

Bước 4: Tìm các siêu khoá S_i bằng cách với $\forall X_i$

Nếu $(TN \cup X_i)^+ = U$ thì $S_i = TN \cup X_i$

Bước 5: Tìm khoá bằng cách loại các khoá không tối thiểu

Gọi S là tập siêu khoá xác định được ở bước 4, $S = (S_1, S_2, \dots, S_n)$

Nếu $S_i \subset S_j$ thì loại S_j ra khỏi tập siêu khoá S

S còn lại chính là tập khoá cần tìm.

Ví dụ 3.13:

Cho lược đồ quan hệ $r(U, F)$ với $U = \{CDEKGH\}$ và $F = \{CK \rightarrow H, C \rightarrow D, E \rightarrow C, E \rightarrow G, CK \rightarrow E\}$. Tìm tất cả các khoá cho $r(U, F)$?

Bước 1 : Xác định các tập thuộc tính TN, TG với $TN = \{K\}$ và $TG = \{CE\}$

Từ bước 2 đến bước 5: Minh hoạ qua bảng dữ liệu sau

Gọi X_i là tập con của TG

X_i	$TN \cup X_i$	$(TN \cup X_i)^+$	Siêu khoá	Khoá
ϕ	K	K		
C	KC	U	KC	K

				C
E	KE	U	KE	KE
EC	KCE	U	KCE	

Kết luận : Các khoá của $r(U,F)$ là: KC và KE

Ví dụ 3.14 :

Cho $r(U,F)$ với $U=(A, B, C, D, E, I)$.

Và $F=\{ACD \rightarrow EBI; CE \rightarrow AD\}$. Tìm tất cả các khoá cho $r(U,F)$?

Bài giải:

$$TN=\{ C \}; TG=\{ ADE \}$$

X_1	$(TN \cup X_1)$	$(TN \cup X_1)^+$	Siêu khoá	Khóa
θ	C	C		
A	AC	AC		
D	CD	CD		
AD	ACD	ABCDEI	ACD	ACD
E	CE	ABCDEI	CE	CE
AE	ACE	ABCDEI	ACE	
DE	CDE	ABCDEI	CDE	
ADE	ADCE	ABCDEI	ACDE	

Kết luận : Tất cả các khoá của $r(U,F)$ là : CE và ACD

3.7.5 Các bước chuẩn hoá một quan hệ đến 3NF

Thuật toán 7: Chuẩn hoá một quan hệ thành dạng 3NF

Cho lược đồ quan hệ $r(U,F)$ với U là tập hữu hạn các thuộc tính và F là tập phụ thuộc hàm. Hãy chuẩn hoá $r(U,F)$ về dạng 3NF và phép tách ρ là không mất mát thông tin

$$\rho = \{r_1(U_1), r_2(U_2) \dots r_n(U_n)\}, \text{ sao cho } r_i(U_i) \text{ là dạng chuẩn 3NF}$$

Bước 1: Tìm khoá của r

Bước 2: Sử dụng thuật toán 2 tìm phủ tối thiểu của F

Bước 3: Xác định các lược đồ con

Mỗi phụ thuộc hàm thuộc F' tương đương với một lược đồ con.

Giả sử xét $Y \rightarrow A_i \in F_{\text{tối thiểu}}$ ta có lược đồ $r_j(U_j)$, với $U_j = YA_i$ khoá của $r_j(U_j)$ là Y với $j \subseteq \{1, \dots, n\}$ và $Y \subseteq U$

Lưu ý:

Nếu $\exists X_i \rightarrow A_{i1}; X_i \rightarrow A_{i2}; \dots; X_i \rightarrow A_{in}$ thì $U_i = (X_i A_{i1} A_{i2} \dots A_{in})$ và ta có lược đồ quan hệ $r_i(U_i)$ với khoá là X_i , $i \subseteq \{1, \dots, n\}$ và $X_i \subseteq U$.

Kết quả ta có các lược đồ: $\{r_1(U_1), r_2(U_2) \dots r_n(U_n)\}$

Bước 4: Kết luận phép chuẩn hoá

Nếu tồn tại ít nhất một thuộc tính khoá không có mặt trong các lược đồ $\{r_1(U_1), r_2(U_2) \dots r_n(U_n)\}$ thì kết luận phép chuẩn hoá $r(U,F)$ về 3NF là

$$\rho = \{r_0(U_0), r_1(U_1), r_2(U_2) \dots r_n(U_n)\} \text{ với } r_0(U_0) = K$$

Ngược lại phép chuẩn hoá $r(U,F)$ về 3NF là $\rho = \{r_1(U_1), r_2(U_2) \dots r_n(U_n)\}$

Ví dụ 3.15:

Cho lược đồ quan hệ $r(U,F)$ với $U = \{A, B, C, D, E, L, G, H\}$

và $F = \{A \rightarrow BC; C \rightarrow B; D \rightarrow EL; ADC \rightarrow G\}$

Chuẩn hoá r thành dạng 3NF

Bước 1: Tìm khoá tối thiểu:

$K_0 = U = \{A, B, C, D, E, G, H, L\}$ dùng thuật toán 4 loại bỏ dần ta có

$K = ADH$

Bước 2:

Tìm phủ tối thiểu

2.1 Tách các phụ thuộc hàm

$$F = \{A \rightarrow C; C \rightarrow B; A \rightarrow B; D \rightarrow E; D \rightarrow L; ADC \rightarrow G\}$$

2.2 Loại bỏ các phụ thuộc hàm dư thừa: Có $A \rightarrow B$ thừa vì có $A \rightarrow C$ và $C \rightarrow B$

$$\text{Ta có } F = \{C \rightarrow B; A \rightarrow C; D \rightarrow E; D \rightarrow L; ADC \rightarrow G\}$$

2.3 Bỏ các thuộc tính thừa ở vế trái

Vì $A \rightarrow C$ nên phụ thuộc hàm $ADC \rightarrow G$ thừa thuộc tính C nên ta có: $AD \rightarrow G$

$\Rightarrow F = \{C \rightarrow B; A \rightarrow C; D \rightarrow E; D \rightarrow L; AD \rightarrow G\}$

Bước 3: Ta có các r_i như sau:

$A \rightarrow C \Rightarrow R_1(U_1) = (AC)$ khoá $K_1 = \{A\}$

$C \rightarrow B \Rightarrow R_2(U_2) = (CB)$ khoá $K_2 = \{B\}$

$D \rightarrow E; D \rightarrow L; \Rightarrow R_3(U_3) = (DEL)$ khoá $K_3 = \{D\}$

$AD \rightarrow G \Rightarrow R_4(U_4) = (ADG)$ khoá $K_4 = \{AD\}$

Bước 4: Kết quả

$\rho = \{r_0(ADH), r_1(AC), r_2(CB), r_3(DEL), r_4(ADG)\}$

Thoả mãn $R_i(U_i)$ là 3NF

Thuật toán 8: Chuẩn hoá một lược đồ quan hệ về dạng chuẩn Boye -Code

Cho lược đồ quan hệ $r(U,F)$ với U là tập hữu hạn các thuộc tính và F là tập phụ thuộc hàm. Hãy chuẩn hoá $r(U,F)$ về dạng chuẩn Boye - Code.

Bước 1: Gọi ρ là phép chuẩn hoá $r(U,F)$ về BCNF, $\rho = \{r(U,F)\}$

Bước i: Nếu s là một lược đồ thuộc ρ , s chưa ở BCNF chọn $X \rightarrow A$ là một phụ thuộc hàm thoả trên S mà X không phải là siêu khoá của S và $A \notin X$ thì tách S thành hai lược đồ quan hệ:

$s_i = X \cup Y = s_i(XY)$ Khoá là X

$s_{i+1} = (U \setminus Y)$, Xác định lại khoá và tập phụ thuộc hàm cho s_{i+1} .

Quá trình tiếp tục cho tới khi tất cả các lược đồ đều ở BCNF

Kết quả được phép tách:

$\rho = \{r_1(U_1), r_2(U_2), \dots, r_m(U_m)\}$ với mỗi r_i là quan hệ ở dạng BCNF

Ví dụ 3.16: Cho lược đồ quan hệ $r(U,F)$

$U = \{C, T, H, N, S, G\}$

$F = \{CS \rightarrow G; C \rightarrow T; HT \rightarrow N; HS \rightarrow N; HN \rightarrow C\}$. Hãy chuẩn hoá $r(U,F)$ về dạng chuẩn BCNF?

Khoá của r là HS

Bước 1: $r(U,F)$ không thoả mãn BCNF

Xét $CS \rightarrow G$ vì CS không là siêu khoá của r nên tách:

$r_1 = (CSG)$; $r_2 = (CTHNS)$ và $F = \{C \rightarrow T; HT \rightarrow N; HS \rightarrow N; HN \rightarrow C\}$, khoá của r_2 là HS . Trong đó r_1 thoả mãn BCNF, r_2 không thoả mãn BCNF.

Bước 2:

Xét r_2

Có $C \rightarrow T$ mà C không là khoá của r_2 nên tách:

$r_{21} = (CT)$; $r_{22} = (CHNS)$ và $F = \{ HC \rightarrow N ; HS \rightarrow N ; HN \rightarrow C ; \}$, khoá của r_{22} là HS . Trong đó r_{21} thoả mãn là BCNF, r_{22} không thoả mãn BCNF.

Bước 3:

Xét r_{22} :

Có $HN \rightarrow C$ mà CH không là khoá nên tách:

$r_{221} = (CHN)$; $r_{222} = (CHS)$; r_{221} thoả mãn CBNF; r_{222} thoả mãn BCNF

Thuật toán dừng vì $\forall r_i$ thoả mãn BCNF.

Cuối cùng ta có:

$$\rho = \{r_1(CSG); r_{21}(CT); r_{221}(CHN); r_{222}(CHS)\}$$

Trong đó mỗi r_i là BCNF.

3.8 Phụ thuộc đa trị

3.8.1 Khái niệm

Ta thấy dữ liệu có mối quan hệ với nhau đó là phụ thuộc hàm. Tuy vậy cũng có trường hợp quan hệ đó không có sự phụ thuộc hàm. ảnh xạ trên các thuộc tính không phải là đơn trị mà có nhiều giá trị. Mối quan hệ đó gọi là phụ thuộc đa trị (Multivalued Dependency-MVD)

Ví dụ: Quan hệ KHDH (kế hoạch dạy học)

Giáo viên	Môn	Lớp
A	M2	K1
A	M1	K2
A	M2	K2
A	M1	K1

Như vậy với một giáo viên ta chưa hẳn đã xác định được dạy lớp nào, môn gì cụ thể

3.8.2 Định nghĩa

Cho R là một lược đồ quan hệ X, Y là 2 tập con của R . $Z = R - XY$. Quan hệ $r(R)$ gọi là phụ thuộc đa trị nếu với bất kỳ 2 bộ $t_1, t_2 \in r$, với $t_1[X] = t_2[X]$ tồn tại một bộ $t_3 \in r$ sao cho:

$$t_3[X] = t_1[X]; t_3[Y] = t_1[Y]; t_3[Z] = t_2[Z];$$

Ký hiệu phụ thuộc đa trị

$$X \twoheadrightarrow Y$$

Ta nói X xác định đa trị Y ; hay Y phụ thuộc đa trị vào X

Ví dụ: Xét quan hệ KHDH trên là phụ thuộc đa trị

Nhận xét:

-Xét mô hình trên ta còn có: $t4[Z] = t1[Z]$; $t2[Y] = t4[Y]$

-Nếu $Y = \emptyset$ thì $X \twoheadrightarrow \emptyset$ đúng với mọi quan hệ

-Nếu $X = \emptyset$ thì $\emptyset \twoheadrightarrow Y$ đúng khi Y độc lập với các thuộc tính khác trong r

3.8.3 Hệ tiên đề:

(1) Tiên đề bù: $X \twoheadrightarrow Y \Rightarrow Y \twoheadrightarrow X \setminus Y$

(2) Tiên đề tưng trướng: $X \twoheadrightarrow Y$; $V \in W \Rightarrow WX \twoheadrightarrow WY$

(3) Tiên đề bắc cầu: $X \twoheadrightarrow Y$ và $Y \twoheadrightarrow Z \Rightarrow X \twoheadrightarrow Z \setminus Y$

(4) Tiên đề về quan hệ phụ thuộc đơn trị và đa trị: $X \twoheadrightarrow Y$ thì $X \twoheadrightarrow Y$

4. Nếu $X \twoheadrightarrow Y$, $Z \subseteq Y$, $W \cap Y = \emptyset$, $W \twoheadrightarrow Z$ thì $X \twoheadrightarrow Z$

3.8.4 Các luật suy diễn của phụ thuộc đa trị

Luật hợp:

Nếu $X \twoheadrightarrow Y$ và $X \twoheadrightarrow Z$ thì $X \twoheadrightarrow YZ$

Luật tựa bắc cầu

Nếu $X \twoheadrightarrow Y$ và $WY \twoheadrightarrow Z$ thì $WX \twoheadrightarrow Z \setminus WX$

Luật tựa bắc cầu hỗn hợp

Nếu $X \twoheadrightarrow Y$ và $XY \twoheadrightarrow Z$ thì $X \twoheadrightarrow Z \setminus Y$

Luật tách

Nếu $X \twoheadrightarrow Y$ và $X \twoheadrightarrow Z$ thì $X \twoheadrightarrow Y \setminus Z$; $X \twoheadrightarrow Z \setminus Y$; $X \twoheadrightarrow Y \cap Z$

Định lý: Cho $r(U)$ có phép tách $\rho = \{r_1(U1), r_2(U2)\}$ là phép tách hai không mất mát thông tin khi và chỉ khi:

$$U1 \cap U2 \twoheadrightarrow U1 \setminus U2$$

$$U1 \cap U2 \twoheadrightarrow U2 \setminus U1$$

3.8.5 Dạng chuẩn 4 NF

-Một phụ thuộc hàm đa trị $X \twoheadrightarrow Y$ gọi là sơ cấp nếu với $X, Y \neq \emptyset$ $X \cup Y \neq U$ mà $\forall X' < X \Rightarrow X' \not\rightarrow Y$

Quan hệ r gọi là dạng chuẩn 4 nếu mọi phụ thuộc đa trị sơ cấp đều được xác định bởi khoá chính

Ví dụ: Quan hệ KHDH trên

Khoá chính: GML

Tách: KHDH1: (GM) có $G \twoheadrightarrow M$

KHDH2: (GL) có $G \twoheadrightarrow L$

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 3

1. Định nghĩa phụ thuộc hàm và các khái niệm liên quan.
2. Định nghĩa lược đồ quan hệ và cho ví dụ minh họa
3. Phát biểu tiên đề Armstrong và các hệ quả.
4. Định nghĩa bao đóng của một tập thuộc tính.
5. Định nghĩa phủ của một tập phụ thuộc hàm. Phủ tối thiểu.
6. Định nghĩa phép tách một lược đồ quan hệ.
7. Nêu các dạng chuẩn 1NF, 2NF, 3NF, BCNF và cho ví dụ minh họa.
8. Cho lược đồ quan hệ $r(U,F)$. Tập thuộc tính $U = \{ABDEGIH\}$. Tập phụ thuộc hàm: $F = \{AB \twoheadrightarrow E, AG \twoheadrightarrow I, BE \twoheadrightarrow I, E \twoheadrightarrow G, GI \twoheadrightarrow H\}$. Chứng minh: $AB \twoheadrightarrow GH$.
9. Cho lược đồ quan hệ $r(U,F)$ với $U = \{ABCDEFGH\}$; $F = \{AB \twoheadrightarrow C, C \twoheadrightarrow D, CD \twoheadrightarrow E, CE \twoheadrightarrow GH, G \twoheadrightarrow A\}$. Chứng minh: $AB \twoheadrightarrow E$; $AB \twoheadrightarrow G$
10. Cho sơ đồ quan hệ $r(U,F)$ với $U = \{ABCDEFGH\}$; $F = \{A \twoheadrightarrow D, AB \twoheadrightarrow DE, CE \twoheadrightarrow G, E \twoheadrightarrow H\}$. Hãy tính $(AB)^+$
11. Cho sơ đồ quan hệ $r(U,F)$ với $U = \{ABCDEG\}$; $F = \{A \twoheadrightarrow D, AB \twoheadrightarrow E, BG \twoheadrightarrow E, CD \twoheadrightarrow G, E \twoheadrightarrow C\}$. Hãy tính $(AB)^+$
12. Cho sơ đồ quan hệ $r(U,F)$; $U = \{ABCDEH\}$; $F = \{BC \twoheadrightarrow E, D \twoheadrightarrow A, C \twoheadrightarrow A, AE \twoheadrightarrow D, BE \twoheadrightarrow CH\}$. Tìm một khoá tối thiểu cho $r(U,F)$
13. Cho sơ đồ quan hệ $r(U,F)$, với $U = \{DBIOQS\}$; $F = \{S \twoheadrightarrow D, I \twoheadrightarrow B, IS \twoheadrightarrow Q, B \twoheadrightarrow O\}$. Hãy chuẩn hoá $r(U,F)$ về dạng chuẩn 3NF.
14. Cho sơ đồ quan hệ $r(U,F)$; $U = (ABCDEFGH)$; $F = \{ABC \twoheadrightarrow D, AB \twoheadrightarrow E, BC \twoheadrightarrow DC, C \twoheadrightarrow DE, CE \twoheadrightarrow H, DC \twoheadrightarrow G, CH \twoheadrightarrow G, AD \twoheadrightarrow H\}$. Hãy chuẩn hoá $r(U,F)$ về dạng chuẩn 3NF.
15. Cho lược đồ quan hệ $r(U,F)$. $U = \{C\#, I, D, B, K, E, L\}$; $F = \{C\# \twoheadrightarrow IBKE, D \twoheadrightarrow B, K \twoheadrightarrow E\}$. Hãy chuẩn hoá $r(U,F)$ về dạng chuẩn 3NF.

16. Cho sơ đồ quan hệ $r(U,F)$, $U=\{ABCD\}$; $F=\{D \rightarrow B, C \rightarrow A, B \rightarrow ACD\}$. Hãy xác định dạng chuẩn cao nhất của $r(U)$? Giải thích.

17. Kiểm tra tính kết nối không mất mát thông tin của phép tách $r(ABCDE)$ thành các lược đồ quan hệ sau: $r_1=AD$; $r_2=AB$; $r_3=BE$; $r_4=CDE$; $r_5 = AE$. Với tập phụ thuộc hàm: $F=\{A \rightarrow C, B \rightarrow C; A \rightarrow B; DE \rightarrow C; CE \rightarrow A\}$.

18. Cho lược đồ quan hệ $r(U,F)$; $U=\{ABCDEFG\}$; $F=\{AB \rightarrow C; C \rightarrow B; ABD \rightarrow E; G \rightarrow A\}$. Chuẩn hoá r thành dạng BCNF.

19. Kiểm tra tính không mất mát thông tin của phép tách $r(ABCDEG)$ thành $\rho(r) = (BC, AC, ADBE, ADBF)$.

Với tập phụ thuộc hàm $F=\{AB \rightarrow C; C \rightarrow B; ABD \rightarrow E; G \rightarrow A\}$

20. Cho lược đồ quan hệ $r(U,F)$ với

$$U=(ABCDE)$$

$$F=\{AB \rightarrow C; AD \rightarrow B; B \rightarrow D\}.$$

a) Tìm giao của các khoá?

b) Tìm tất cả các khoá cho $r(U,F)$?

21. Cho lược đồ quan hệ $r(U,F)$

$$U=(ABCDEFGH)$$

$$F=\{B \rightarrow AC, DH \rightarrow AE, AC \rightarrow BE, E \rightarrow H, A \rightarrow D, G \rightarrow E\}.$$

a) Tìm một khoá cho $r(U,F)$.

b) Chứng minh rằng: $CG \rightarrow EH \in F^+$

c) Các tập thuộc tính CGH và $ABCG$ có phải là khoá của r hay không?

22. Cho lược đồ quan hệ $r(U,F)$:

$$U = \{A, B, C, D, E, G\}$$

$$F = \{AB \rightarrow C, D \rightarrow EG, BE \rightarrow C, BC \rightarrow D, CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG\}.$$

a). Tìm tất cả các khoá của $r(U,F)$

b). Tìm giao của khoá

Tìm phủ tối thiểu của F

Chuẩn hoá $r(U,F)$ về dạng chuẩn 3NF

Chương 4

NGÔN NGỮ ĐỊNH NGHĨA VÀ THAO TÁC DỮ LIỆU

4.1 Ngôn ngữ đại số quan hệ

4.1.1 Khái niệm

Là ngôn ngữ dựa trên các phép toán của đại số quan hệ mà ta đã đề cập tới trong Chương 2. Mỗi câu hỏi được biểu diễn bằng một tập các phép toán nào đó.

4.1.2 Các câu lệnh của ngôn ngữ đại số quan hệ

5. Phép hợp

<quan hệ 1> UNION <quan hệ 2>

6. Phép giao

<quan hệ 1 > INTERSECT <quan hệ 2>

7. Phép trừ

<quan hệ 1 > MINUS <quan hệ 2>

8. Phép tích ĐỀ- các

<quan hệ 1 > TIMES <quan hệ 2>

9. Phép chọn

SELECT <quan hệ> WHERE <điều kiện>

10. Phép chiếu

PROJECT <quan hệ> OVER <danh sách thuộc tính>

11. Phép kết nối

JOIN <quan hệ 1> AND <quan hệ 2> [OVER <danh sách thuộc tính>] [WHERE <danh sách thuộc tính>]

12. Phép chia

DIVIDE <quan hệ 1> BY <quan hệ 2> OVER <danh sách thuộc tính> [AND <danh sách thuộc tính>]

13. Đưa ra kết quả

GIVING <kết quả>

4.1.3 Ví dụ minh họa

-Bổ xung vào quan hệ CONGTY một công ty nữa

```
Congty UNION {"CT4", "HỒng Hà",1200000, "Nam định"}  
GIVING Congty
```

-Xóa tên công ty CT5

```
Congty MINUS {"CT5", , , }  
GIVING CongTy
```

-Sửa địa chỉ của công ty HỒng Hà thành Hà nội, thực chất là xoá bộ cũ thay bộ mới với nội dung mới

```
Congty MINUS ("CT4", , , )  
GIVING Tgian  
Tgian UNION ( "CT4", , "Hà nội")  
GIVING CongTy
```

Chú ý: Lệnh này cần đề phòng mất dữ liệu

-Tìm kiếm thông tin về công ty CT1

```
SELECT CongTy WHERE MaCongTy = "CT1"  
GIVING CongTy
```

4.1.4 Biểu diễn một số câu hỏi

14. Đưa ra danh sách các mặt hàng màu đỏ

```
SELECT Hanghoa WHERE Mau = "ĐỎ"  
GIVING Ketqua
```

15. Cho biết mã các công ty cung cấp mặt hàng H1

```
SELECT CungCap WHERE MaHang = "H1" GIVING Tgian  
PROJECT Tgian OVER MaCongTy GIVING Ketqua
```

Hoặc:

```
PROJECT (SELECT CungCap WHERE MaHang= "H1") OVER  
MaCongTy  
GIVING Ketqua
```

16. Cho biết tên công ty cung cấp mặt hàng H1

```
SELECT Cungcap WHERE MaHang = "H1" GIVING Tgian1  
JOIN Tgian1 AND Congty OVER MaCongTy GIVING Tgian2  
PROJECT Tgian2 OVER TenCongTy GIVING Ketqua
```

17. Cho biết tên công ty cung cấp cả hai mặt hàng H1 và H2

```
SELECT CungCap WHERE MaHang= "H1" GIVING Tgian1  
PROJECT Tgian1 OVER MaCongTy GIVING Tgian2
```

```
SELECT CungCap WHERE MaHang= "H2" GIVING Tgian1'
PROJECT Tgian1' OVER MaCongTy GIVING Tgian2'
Tgian2 INTERSECT Tgian2' GIVING Tgian
JOIN Tgian AND Congty OVER MaCongTy GIVING Tgian'
PROJECT Tgian' OVER TenCongTy GIVING Ketqua
```

4.2 Ngôn ngữ SQL (Structure Query Language)

4.2.1 Giới thiệu

SQL được phát triển từ ngôn ngữ SEQUEL-2, thử nghiệm và cài đặt tại Trung tâm nghiên cứu của hãng IBM ở San Joes (California-US), cho hệ thống Quản trị cơ sở dữ liệu lớn điển hình là System - R, SQL vừa đóng vai trò là một ngôn ngữ có thể thao tác độc lập của người sử dụng đầu cuối, đồng thời lại có khả năng là một ngôn ngữ con được nhúng trong ngôn ngữ chủ.

SQL là ngôn ngữ phi thủ tục, dễ sử dụng. Do vậy hiện nay rất nhiều hệ quản trị cơ sở dữ liệu sử dụng ngôn ngữ SQL như ACCESS, ORACLE, DB2...

SQL cho phép tạo lập và thực hiện các thao tác truy xuất trên cơ sở dữ liệu một cách rất dễ dàng.

4.2.2 Nhóm lệnh tạo lập cơ sở dữ liệu

a) Lệnh tạo bảng

```
CREATE TABLE <tên_bảng>
( <tên_cột> <kiểu_dữ_liệu> [<ràng_buộc_dữ_liệu>] [, ...n ]
    [, <ràng_buộc_bảng >])
```

Trong đó:

- + tên_bảng, tên_cột: do người sử dụng tự định nghĩa tuân theo quy tắc đặt tên
 - Tên cột bắt đầu bằng chữ hoa, sao cho ngắn gọn chính xác và đầy đủ.
 - Không nên đặt tên bảng và tên cột có khoảng trắng.
 - Không nên đặt tên bảng và tên cột trùng với các từ khóa.
- + Kiểu_dữ_liệu: Chọn một kiểu dữ liệu nào phù hợp với dữ liệu người dùng sẽ nhập vào. Bảng 4.1 cho thấy các kiểu dữ liệu được quy định:

Bảng 4.1 Các kiểu dữ liệu

Tên kiểu dữ liệu	Phạm vi biểu diễn	Kích thước
bigint	Từ 2^{63} (-9223372036854775808) đến $2^{63}-1$ (9223372036854775807)	Kiểu số nguyên, 8 byte
int	Từ -2^{31} (-2,147,483,648) đến $2^{31}-1$ (2,147,483,647)	Kiểu số nguyên, 4 byte
smallint	Từ 2^{15} (-32,768) đến $2^{15}-1$ (32,767)	Kiểu số nguyên, 2 byte
tinyint	Từ 0 đến 255	Kiểu số nguyên, 1 byte
bit	Biểu diễn giá trị 0 hoặc 1	Kiểu số nguyên
decimal(n,p)	Từ $(-10^{38}+1)$ đến $(10^{38}-1)$	Kiểu số thực tinh
numeric(n,p)	Từ $(-10^{38}+1)$ đến $(10^{38}-1)$	Kiểu số thực tinh
money	Từ -2^{63} (-922,337,203,685,477.5808) đến $2^{63}-1$ (+922,337,203,685,477.5807),	Kiểu dữ liệu tiền tệ
smallmoney	Từ -214,748.3648 đến +214,748.3647	Kiểu dữ liệu tiền tệ
float	Từ $(-1.79E+308)$ đến $(1.79E+308)$	Kiểu số thực động
real	Từ $(-3.40E+38)$ đến $(3.40E+38)$	Kiểu số thực động
datetime	Từ (January 1- 1753), đến (December 31- 9999)	Kiểu dữ liệu ngày giờ
smalldatetime	Từ (January 1, 1900), đến (June 6, 2079)	Kiểu dữ liệu ngày giờ
char(n)	Kiểu ký tự có độ dài cố định, $n_{\max}=8000$ ký tự	Kiểu ký tự
varchar(n)	Kiểu ký tự có độ dài thay đổi, $n_{\max}=8000$ ký tự	Kiểu ký tự

text	Kiểu ký tự có độ dài tối đa là $2^{31} - 1$ (2,147,483,647) ký tự	Kiểu ký tự
------	---	------------

+ Ràng_buộc_dữ_liệu: Là một số quy định dùng để kiểm tra dữ liệu khi thực hiện các thao tác nhập hoặc cập nhật dữ liệu, có các loại ràng buộc sau:

- Null (mặc định): Chấp nhận giá trị rỗng
- Not Null: Không chấp nhận giá trị rỗng
- Unique: giá trị nhập vào phải duy nhất.
- Primary key: Ràng buộc khoá chính dùng để xác định duy nhất một đối tượng nên giá trị của chúng phải duy nhất, không chấp nhận giá trị Null.

+ Ràng_buộc_bảng: gồm 2 loại là ràng buộc khoá chính và ràng buộc khoá ngoài

Ràng buộc khoá chính được định nghĩa theo cú pháp sau:

CONSTRAINT <tên_ràng_buộc> PRIMARY KEY (danh_sách_thuộc_tính_khoá)

Ràng buộc khoá ngoài: Dùng để kiểm tra sự tương quan về dữ liệu giữa khoá chính và khoá ngoài ở hai bảng dữ liệu có mối quan hệ với nhau, được định nghĩa theo cú pháp sau:

CONSTRAINT <tên_ràng_buộc> FOREIGN KEY (<thuộc_tính_khoá_ngoài>)

REFERENCES <tên_bảng_liên_kết> (<thuộc_tính_liên_kết>)

Ví dụ 4.1: Cho cơ sở dữ liệu quản lý **Thực tập** gồm các bảng dữ liệu sau:

+ Bảng **LopHoc** chứa thông tin về các lớp học (4.2)

MaLop	TenLop
L01	K7A-CNTT
L02	K7B-CNTT

Trong đó: MaLop là Mã lớp; TenLop là Tên lớp

+ Bảng **SinhVien** chứa danh sách sinh viên (4.3):

MaSV	HotenSV	NS	GT	Diachi	MaLop
08K7A1	Nguyễn Văn Dũng	09/11/1989	1	Hà Nội	L01

08K7A2	Lê Ngọc Dương	01/09/1989	1	Bắc Giang	L01
08K7A3	Trần Thị Hồng	17/06/1989	0	Bắc Kạn	L02

Trong đó: MaSV là “Mã số sinh viên”; HoTenSV là “Họ tên sinh viên”; NS là “Ngày sinh”; DiaChi là “Địa chỉ” của sinh viên; GT là “Giới tính”;

+ Bảng **DeTai** chứa danh sách các đề tài thực tập (4.4):

MaDT	TenDT	GVHD	KP
DT01	Pháp triển ứng dụng WEB	Nguyễn Hồng An	2000000
DT02	Xây dựng thư viện điện tử	Trần Văn Lâm	2500000
DT03	Truy vấn dữ liệu Multimedia	Ngô Hải Long	3000000

Trong đó: MaDT là “Mã đề tài”; TenDT là “Tên đề tài”; GVHD là “Họ tên giáo viên hướng dẫn đề tài”; KP là “Kinh phí”.

+ Bảng **SV_DeTai** chứa thông tin về tình hình thực tập của sinh viên (4.5):

MaSV	MaDT	NTT	KQ
08K7A1	DT01	Công ty phần mềm CNN	8
08K7A2	DT02	Khoa CNTT	7
08K7A3	DT03	Công ty phần mềm Thanh Niên	9
08K7A1	DT02	Trung tâm học liệu TN	8

Trong đó: MaDT là “Mã đề tài”; NTT là “Nơi sinh viên đến thực tập”; KQ là “Kết quả thực tập”

Yêu cầu: Hãy tạo cấu trúc ba bảng dữ liệu trên.

Câu lệnh tạo bảng của SQL:

Create table LopHoc(

MaLop char(6) Primary Key,

TenLop Varchar(35) Not Null)

Create table SinhVien(

MaSV Varchar(30) Primary Key,

HoTenSV Varchar(35) Not Null,

NS Datetime Not Null,

GT Bit Not Null
DiaChi Varchar(100) Not Null,
MaLop Char(6),
Constraint MaLop_FK Foreign key (MaLop) References LopHoc(MaLop))

Create table DeTai(
MaDT Char(9) Primary key,
TenDT Varchar(100) Not Null,
GVHD Varchar(30) Not Null,
KP SmallMoney)

Create table SV_DeTai(
MaDT Char(9) Primary key,
MaSV Char(10),
NTT Varchar(100) Not Null,
KQ Numeric(5,2) Not Null,
Constraint MaDT_FK Foreign key (MaDT) References DeTai(MaDT),
Constraint MaSV_FK Foreign key (MaSV) References SinhVien(MaSV))

b) Thêm một cột

```
ALTER TABLE <tên_bảng>  
ADD <tên_cột > <kiểu_dữ_liệu> [<ràng_buộc_cột>]
```

Ví dụ 4.2: Thêm vào bảng SinhVien cột số điện thoại

```
ALTER TABLE SinhVien  
ADD SoDT char(10)
```

c) Xoá cột

```
ALTER TABLE <tên_bảng>  
DROP COLUMN <tên_cột>
```

Ví dụ 4.3: Câu lệnh sau sẽ xoá cột số điện thoại trong bảng Sinh viên

```
ALTER TABLE SinhVien  
DROP COLUMN SoDT
```

d) Xoá ràng buộc

```
ALTER TABLE <tên_bảng>  
DROP CONSTRAINT <tên_ràng_buộc>
```

Ví dụ 4.4: Câu lệnh sau sẽ xoá một ràng buộc khoá ngoài trong bảng SV_DETAI

```
ALTER TABLE SV_DETAI DROP CONSTRAINT MaSV_FK
```

4.2.3. Nhãn lỜnh thao t,c đ÷ liỜu

4.2.3.1 LỜnh truy vấn dữ liỜu - SELECT

Việc truy cập và lấy các thông tin từ database được SQL cho phép thực hiện qua câu lệnh SELECT. Câu lệnh SELECT có phạm vi ứng dụng rất rộng, có thể truy cập dữ liệu từ một bảng (table), hay từ nhiều bảng.

Các từ khóa SELECT, FROM, WHERE được sử dụng để tạo nên một câu lệnh SELECT đơn giản nhất

Cú pháp tổng quát có dạng sau:

```
SELECT [ ALL | DISTINCT ] <danh_sách_chỌn>
FROM <danh_sách_bảng>
[ WHERE <điều_kiện > ]
[ GROUP BY <danh_sách_cột> ]
[ HAVING <điều_kiện_nhóm> ]
[ ORDER BY <tên_cột> [ ASC | DESC ] [,..n]]
```

a) Mệnh đề SELECT

Danh sách chọn trong câu lệnh SELECT được sử dụng để chỉ định các thuộc tính, các biểu thức xuất hiện trong bảng kết quả của câu truy vấn. Sử dụng danh sách chọn trong câu lệnh SELECT bao gồm các trường hợp sau:

S1: Chọn tất cả các cột trong bảng

Khi cần hiển thị tất cả các trường trong bảng, sử dụng ký tự * trong danh sách chọn thay vì phải liệt kê tất cả các cột. Trong trường hợp này, các cột được hiển thị trong bảng kết quả của câu truy vấn sẽ tuân theo thứ tự mà chúng xuất hiện trong bảng cấu trúc.

Ví dụ 4.7: Câu lệnh sau sẽ liệt kê danh sách các sinh viên

```
SELECT * FROM SinhVien
```

Kết quả:

MaSV	HotenSV	NS	GT	Diachi	MaLop
08K7A1	Nguyễn Văn Dũng	09/11/1989	1	Hà Nội	L01
08K7A2	Lê Ngọc Dương	01/09/1989	1	Bắc Giang	L01

08K7A3	Trần Thị Hồng	17/06/1989	0	Bắc Kạn	L02

S2: Liệt kê tên cột trong danh sách chọn

Trong trường hợp cần chỉ định cụ thể các cột cần hiển thị trong kết quả truy vấn, ta chỉ định danh sách các tên cột trong danh sách chọn. Thứ tự của các cột trong kết quả truy vấn tuân theo thứ tự của các cột trong danh sách chọn.

Ví dụ 4.8: Liệt kê danh sách sinh viên gồm các thuộc tính sau: MaSV, HoTenSV, DiaChi

Câu lệnh `SELECT MaSV, HoTenSV, DiaChi FROM SinhVien`

Kết quả:

MaSV	HotenSV	Diachi
08K7A1	Nguyễn Văn Dũng	Hà Nội
08K7A2	Lê Ngọc Dương	Bắc Giang
08K7A3	Trần Thị Hồng	Bắc Kạn

Chú ý: Nếu truy vấn được thực hiện trên nhiều bảng và các bảng có các thuộc tính trùng tên thì tên của những thuộc tính này nếu xuất hiện trong câu truy vấn phải được viết dưới dạng:

<Tên_bảng>.<Tên_thuộc tính>

Ví dụ 4.9: Liệt kê mã, họ và tên, các sinh viên đã tham gia ít nhất một lần thực tập

`SELECT SV.MaSV, HoTenSV`

`FROM SinhVien SV, SV_DeTai TT`

`WHERE SV.MaSV=TT.MaSV`

Kết quả:

MaSV	HotenSV
08K7A1	Nguyễn Văn Dũng
08K7A2	Lê Ngọc Dương
08K7A3	Trần Thị Hồng

S3: Hiển thị với việc thay đổi tiêu đề các cột

Trong kết quả truy vấn, tiêu đề của các cột mặc định sẽ là tên của các thuộc tính tương ứng trong bảng. Tuy nhiên, để tiêu đề trở thành thân thiện hơn, ta có thể đổi lại tên tiêu đề của các cột. Để đặt tiêu đề cho một cột nào đó là một chuỗi ký tự đặt trong dấu nháy kép “ ”.

<tên_thuộc_tính > AS <tiêu_đề_cột>

Ví dụ 4.10: Cho biết mã và tên của các đề tài thực tập.

SELECT MaDT AS “Mã đề tài”, TenDT AS “Tên đề tài”

FROM DeTai

Kết quả:

Mã đề tài	Tên đề tài
DT01	Pháp triển ứng dụng WEB
DT02	Xây dựng thư viện điện tử
DT03	Truy vấn dữ liệu Multimedia

S4: Hằng và biểu thức trong danh sách chọn

Ngoài danh sách thuộc tính, trong danh sách chọn của câu lệnh SELECT còn có thể sử dụng các biểu thức. Mỗi biểu thức trong danh sách chọn trở thành một cột trong kết quả truy vấn.

S5: Loại bỏ các bản ghi trùng nhau

Trong kết quả của truy vấn có thể xuất hiện các dòng dữ liệu trùng nhau. Để loại bớt các dòng này, ta chỉ định thêm từ khoá DISTINCT ngay sau từ khoá SELECT.

Ví dụ 4.11: Cho biết thông tin về mã của các sinh viên đã tham gia thực tập.

```
SELECT DISTINCT MaSV FROM SV_DeTai
```

Kết quả (không tồn tại các bản ghi trùng nhau):

MaSV
08K7A1
08K7A2
08K7A3

Nếu ta thực hiện câu lệnh không có tùy chọn DISTINCT như sau:

```
SELECT MaSV FROM SV_DeTai
```

MaSV
08K7A1
08K7A2
08K7A3
08K7A1

Thì kết quả tồn tại cả các bản ghi trùng nhau:

b) Mệnh đề FROM

Mệnh đề FROM nhằm chỉ định các bảng cần truy xuất dữ liệu có liên quan đến câu truy vấn. Sau FROM là danh sách tên của các bảng và khung nhìn tham gia vào truy vấn. Tên của các bảng và các khung nhìn được phân cách nhau bởi dấu phẩy.

Ví dụ 4.5: Câu lệnh dưới đây hiển thị mã và tên của các sinh viên.

```
SELECT MaSV, HoTenSV
```

```
FROM SinhVien
```

Kết quả:

MaSV	HotenSV
08K7A1	Nguyễn Văn Dũng
08K7A2	Lê Ngọc Dương
08K7A3	Trần Thị Hồng

Chú ý: Ta có thể sử dụng các bí danh cho các bảng hay khung nhìn trong câu lệnh truy vấn với cú pháp sau: <Tên_bảng> <Tên_bí_danh>.

Ví dụ 4.6: Câu lệnh sau gán bí danh là SV cho bảng SinhVien

```
SELECT * FROM SinhVien SV
```

c) Mệnh đề điều kiện **WHERE**

Mệnh đề WHERE trong câu lệnh SELECT được sử dụng nhằm xác định các điều kiện đối với việc truy xuất dữ liệu. Sau mệnh đề WHERE là một biểu thức logic và những dòng dữ liệu nào thoả mãn điều kiện được chỉ định mới được hiển thị trong kết quả truy vấn.

Ví dụ 4.12: Câu lệnh dưới đây hiển thị mã số của các sinh viên đã thực tập đề tài có mã 'DT02'

```
SELECT MaSV
```

```
FROM SV_DeTai
```

```
WHERE MaDT='DT02'
```

Kết quả:

MaSV
08K7A1
08K7A2

Trong mệnh đề WHERE thường sử dụng

Các toán tử kết hợp điều kiện (AND, OR)

Các toán tử so sánh.

Toán tử phạm vi và toán tử tập hợp

Các giá trị NULL

S1: Các toán tử so sánh

Toán tử	Ý nghĩa
= (Equals)	Ngang bằng
> (Greater Than)	Lớn hơn
< (Less Than)	Nhỏ hơn
>= (Greater Than or Equal To)	Lớn hơn hoặc bằng
<= (Less Than or Equal To)	Nhỏ hơn hoặc bằng
<> (Not Equal To)	Không bằng
!= (Not Equal To)	Không bằng
!< (Not Less Than)	Không nhỏ hơn
!> (Not Greater Than)	Không lớn hơn

S2: Toán tử phạm vi (Range Operator):

[NOT] BETWEEN a AND b

Toán tử này dùng để kiểm tra xem giá trị dữ liệu nằm trong (ngoài) một khoảng nào đó, ta sử dụng toán tử [NOT] BETWEEN như sau:

Cách sử dụng	Ý nghĩa
giá_trị BETWEEN a AND b	$a \leq \text{giá_trị} \leq b$
giá_trị NOT BETWEEN a AND b	$(\text{giá_trị} < a) \text{ AND } (\text{giá_trị} > b)$

Câu lệnh dưới đây cho biết danh sách các đề tài có kinh phí nằm trong khoảng từ 2,5 đến 3 triệu đồng

```
SELECT *
```

```
FROM DeTai
```

```
WHERE KP Between 2500000 And 3000000
```

Kết quả:

MaDT	TenDT	GVHD	KP
DT02	Xây dựng thư viện điện tử	Trần Văn Lâm	2500000
DT03	Truy vấn dữ liệu Multimedia	Ngô Hải Long	3000000

Câu lệnh dưới đây cho biết danh sách các đề tài có kinh phí không nằm trong khoảng từ 2,5 đến 3 triệu đồng

```
SELECT *
```

```
FROM DeTai
```

```
WHERE KP NOT Between 2500000 And 3000000
```

Kết quả:

MaDT	TenDT	GVHD	KP
DT01	Pháp triển ứng dụng WEB	Nguyễn Hồng An	2000000

S3: Toán tử tập hợp

IN và NOT IN

Toán tử IN được sử dụng khi ta cần chỉ định điều kiện tìm kiếm dữ liệu cho câu SELECT là một danh sách các giá trị. Sau IN (hoặc NOT IN) có thể là một danh sách các giá trị hoặc là một câu lệnh SELECT khác.

Ví dụ 4.12: Để biết danh sách các đề tài có kinh phí bằng 2 hoặc 3 triệu đồng thay vì sử dụng câu lệnh:

```
SELECT * FROM DeTai
WHERE KP =2000000 OR KP=3000000
```

Ta có thể sử dụng câu lệnh

```
SELECT * FROM DeTai WHERE KP IN (2000000,3000000)
```

Kết quả:

MaDT	TenDT	GVHD	KP
DT01	Pháp triển ứng dụng WEB	Nguyễn Hồng An	2000000
DT03	Truy vấn dữ liệu Multimedia	Ngô Hải Long	3000000

S4: Toán tử LIKE và các ký tự đại diện

Toán tử LIKE (hoặc NOT LIKE) sử dụng trong câu lệnh SELECT nhằm mô tả khuôn dạng của dữ liệu cần tìm kiếm. Chúng thường kết hợp với các ký tự đại diện sau đây:

Dấu phần trăm (%): Chỉ một chuỗi các ký tự bất kỳ.

Dấu gạch dưới (_): Chỉ một ký tự bất kỳ

Ví dụ 4.13: Cho biết danh sách các sinh viên có họ tên bắt đầu là ký tự 'L'

```
SELECT * FROM SinhVien
WHERE HoTenSV Like 'L%'
```

Kết quả:

MaSV	HotenSV	NS	GT	Diachi	MaLop
08K7A2	Lê Ngọc Dương	01/09/1989	1	Bắc Giang	L01

Ví dụ 4.14: Cho biết mã, tên, địa chỉ của các sinh viên có họ tên kết thúc bằng chữ 'ng'

```
SELECT * FROM SinhVien  
WHERE HoTenSV Like '%ng'
```

Kết quả:

MaSV	HotenSV	Diachi
08K7A1	Nguyễn Văn Dũng	Hà Nội
08K7A2	Lê Ngọc Dương	Bắc Giang
08K7A3	Trần Thị Hồng	Bắc Kạn

S5: Giá trị NULL

Trong mệnh đề WHERE, để kiểm tra giá trị của một cột có giá trị NULL hay không ta sử dụng cách viết:

WHERE tên_cột IS NULL hoặc WHERE tên_cột IS NOT NULL

d) Sắp xếp kết quả truy vấn

Mặc định các dòng dữ liệu trong kết quả của câu truy vấn tuân theo thứ tự của chúng trong bảng dữ liệu hoặc được sắp xếp theo chỉ mục (nếu trên bảng có chỉ mục). Trong trường hợp muốn dữ liệu được sắp xếp theo chiều tăng hoặc giảm của giá trị của một hoặc nhiều trường, ta sử dụng thêm mệnh đề ORDER BY trong câu lệnh SELECT.

Sau ORDER BY là danh sách các cột cần sắp xếp (tối đa là 16 cột). Dữ liệu được sắp xếp có thể theo chiều tăng (ASC) hoặc giảm (DESC), mặc định là sắp xếp theo chiều tăng dần.

Ví dụ 4.15: Câu lệnh dưới đây hiển thị danh sách các đề tài và sắp xếp theo chiều giảm dần kinh phí.

```
SELECT * FROM DeTai
ORDER BY KP DESC
```

Kết quả:

MaDT	TenDT	GVHD	KP
DT03	Truy vấn dữ liệu Multimedia	Ngô Hải Long	3000000
DT02	Xây dựng thư viện điện tử	Trần Văn Lâm	2500000
DT01	Pháp triển ứng dụng WEB	Nguyễn Hồng An	2000000

Chú ý: Nếu sau ORDER BY có nhiều cột thì việc sắp xếp dữ liệu sẽ được ưu tiên theo chiều từ trái qua phải.

Ví dụ 4.16: Liệt kê danh sách sinh viên và sắp xếp theo tên sinh viên theo Alphabet (tăng dần), nếu trùng tên thì sắp theo giới tính.

```
SELECT * FROM SinhVien
ORDER BY HoTenSV, GT
```

e) Phép kết nối

Khi cần thực hiện một yêu cầu truy vấn dữ liệu từ hai hay nhiều bảng, ta phải sử dụng đến phép kết nối

Để thực hiện được một phép nối, cần phải xác định được những yếu tố sau:

- Những cột nào cần hiển thị trong kết quả truy vấn.
- Những bảng nào có tham gia vào truy vấn.
- Điều kiện để thực hiện phép nối giữa các bảng dữ liệu là gì?

Trong các yếu tố kể trên, việc xác định chính xác điều kiện để thực hiện phép nối giữa các bảng đóng vai trò quan trọng nhất. Trong đa số các trường hợp, điều kiện của phép nối được xác định nhờ vào mối quan hệ giữa các bảng cần phải truy xuất dữ liệu. Thông thường, đó là điều kiện bằng nhau giữa khoá chính và khoá ngoài của hai bảng có quan hệ với nhau.

Ví dụ 4.17: Câu lệnh dưới đây hiển thị danh sách các sinh viên với các thông tin: Mã sinh viên, họ tên, mã lớp và tên lớp


```
SELECT MaSV, HoTenSV, LopHoc.MaLop, TenLop
FROM SinhVien , LopHoc
WHERE SinhVien.MaLop = LopHoc.MaLop
```

Kết quả:

MaSV	HotenSV	MaLop	TenLop
08K7A1	Nguyễn Văn Dũng	L01	K7A-CNTT
08K7A2	Lê Ngọc Dương	L01	K7A-CNTT
08K7A3	Trần Thị Hồng	L02	K7B-CNTT

Trong câu lệnh trên, các bảng tham gia vào truy vấn bao gồm: SinhVien và LopHoc. Điều kiện để thực hiện phép kết nối giữa hai bảng là điều kiện sau: SinhVien.MaLop = LopHoc.MaLop

Chú ý: - Tên của một số cột nào đó trong các bảng có tham gia vào truy vấn. Nếu tên cột trong các bảng trùng tên nhau thì tên cột phải được viết dưới dạng: **Tên_bảng.tên_cột**

- Dấu sao (*) được sử dụng trong danh sách chọn khi cần hiển thị tất cả các cột của các bảng tham gia truy vấn.

- Trong trường hợp cần hiển thị tất cả các cột của một bảng nào đó, ta sử dụng cách viết: **tên_bảng.***

Ví dụ 4.18 : Liệt kê danh sách các sinh viên đã tham gia thực tập để tại có mã số là 'DT02'

```
SELECT SinhVien.*
FROM SinhVien , SV_DeTai
WHERE SinhVien.MaSV = SV_DeTai.MaSV AND MaDT='DT02'
```

Kết quả

MaSV	HotenSV	NS	GT	Diachi	MaLop
08K7A1	Nguyễn Văn Dũng	09/11/1989	1	Hà Nội	L01
08K7A2	Lê Ngọc Dương	01/09/1989	1	Bắc Giang	L01

f) Thống kê dữ liệu với **GROUP BY**

Mệnh đề GROUP BY sử dụng trong câu lệnh SELECT nhằm phân hoạch các dòng dữ liệu trong bảng thành các nhóm dữ liệu và trên mỗi nhóm dữ liệu thực hiện tính toán các giá trị thống kê như tính tổng, tính giá trị trung bình...

Các hàm nhóm được sử dụng để tính giá trị thống kê cho toàn bảng hoặc trên mỗi nhóm dữ liệu. Chúng có thể được sử dụng như là các cột trong danh sách chọn của câu lệnh SELECT hoặc xuất hiện trong mệnh đề HAVING, nhưng không được phép xuất hiện trong mệnh đề WHERE

SQL cung cấp các hàm nhóm dưới đây:

Hàm nhóm	Chức năng
SUM(Tên_thuộc_tính biểu_thức)	Tính tổng các giá trị
AVG(Tên_thuộc_tính biểu_thức)	Tính trung bình của các giá trị
COUNT(Tên_thuộc_tính biểu_thức)	Đếm số các giá trị trong biểu thức
COUNT(*)	Đếm số các dòng được chọn
MAX(Tên_thuộc_tính biểu_thức)	Tính giá trị lớn nhất
MIN(Tên_thuộc_tính biểu_thức)	Tính giá trị nhỏ nhất

Trong đó:

Hàm SUM, AVG chỉ làm việc với các biểu thức số

Hàm SUM, AVG, COUNT, MIN và MAX bỏ qua các giá trị NULL khi tính toán.

Hàm COUNT(*) không bỏ qua các giá trị NULL

S1: Thống kê trên toàn bộ dữ liệu

Khi cần tính toán giá trị thống kê trên toàn bộ dữ liệu, ta sử dụng các hàm nhóm trong danh sách chọn của câu lệnh SELECT. Trong trường hợp này, trong danh sách chọn không được sử dụng bất kỳ một tên cột hay biểu thức nào ngoài các hàm nhóm.

Ví dụ 4.19 : Để tính trung bình kinh phí của tất cả các đề tài ta sử dụng câu lệnh như sau:

```
SELECT AVG(KP) AS TBKP FROM DeTai
```

Kết quả:

TBKP
2500000

S2: Thống kê dữ liệu trên các nhóm

Trong trường hợp cần thực hiện tính toán các giá trị thống kê trên các nhóm dữ liệu, ta sử dụng mệnh đề GROUP BY để phân hoạch dữ liệu thành các nhóm riêng biệt. Các hàm nhóm được sử dụng sẽ thực hiện thao tác tính toán trên mỗi nhóm và cho biết giá trị thống kê theo từng nhóm dữ liệu.

Ví dụ 4.20: Câu lệnh dưới đây cho biết sĩ số sinh viên của mỗi lớp

```
SELECT LopHoc.MaLop, TenLop, COUNT(MaSV) AS SiSo
FROM LopHoc, SinhVien
WHERE LopHoc.MaLop = SinhVien. MaLop
GROUP BY LopHoc.MaLop, TenLop
```

Kết quả:

MaLop	TenLop	SiSo
L01	K7A-CNTT	2
L02	K7B-CNTT	1

Chú ý:

Biểu thức nào điều khiển việc phân nhóm dữ liệu thì các biểu thức đó phải được liệt kê sau mệnh đề GROUP BY.

Trong trường hợp danh sách chọn của câu lệnh SELECT có các hàm nhóm và những biểu thức hoặc thuộc tính không phải là đối số của các hàm nhóm thì những biểu thức hoặc những thuộc tính này phải được liệt kê đầy đủ trong mệnh đề GROUP BY, nếu không câu lệnh sẽ không hợp lệ.

Ví dụ 4.21: Dưới đây là một câu lệnh sai do thiếu trường TenLop sau mệnh đề GROUP BY

```
SELECT LopHoc.MaLop, TenLop, COUNT(MaSV) AS SiSo
FROM LopHoc, SinhVien
WHERE LopHoc.MaLop = SinhVien. MaLop
```

GROUP BY LopHoc.MaLop

g) Mệnh đề điều kiện nhóm **HAVING**

Mệnh đề HAVING là mệnh đề điều kiện tác động trên các nhóm dữ liệu. Mệnh đề HAVING luôn sử dụng kết hợp với mệnh đề GROUP BY.

Một điểm khác biệt giữa HAVING và WHERE là trong điều kiện của WHERE không được phép sử dụng các hàm nhóm trong khi HAVING lại cho phép sử dụng các hàm nhóm trong điều kiện của mình.

Ví dụ 4.22: Cho biết các lớp có số ≥ 50 học sinh

```
SELECT    LopHoc.MaLop, TenLop, COUNT(*) AS SiSo
FROM      LopHoc, SinhVien
WHERE     LopHoc.MaLop = SinhVien. MaLop
GROUP BY LopHoc.MaLop, TenLop
HAVING    COUNT(*) >= 50
```

h) Truy vấn con (Subquery)

Truy vấn con là một câu lệnh SELECT được lồng vào bên trong một câu lệnh SELECT, INSERT, UPDATE, DELETE hoặc bên trong một truy vấn con khác. Loại truy vấn này được sử dụng để biểu diễn cho những truy vấn trong đó điều kiện của câu truy vấn dữ liệu này cần phải sử dụng đến kết quả của một truy vấn khác.

Ví dụ 4.23 :Đưa ra danh sách các sinh viên đã thực hiện đề tài có mã là 'DT02'

```
SELECT SV.*
FROM SinhVien SV, SV_DeTai TT
WHERE SV.MaSV=TT.MaSV AND MaDT='DT02'
```

Hoặc có thể viết như sau:

```
SELECT *
FROM SinhVien
WHERE MaSV IN (SELECT MaSV
                FROM SV_DeTai
                WHERE MaDT = 'DT02')
```

Ví dụ 4.24: Cho biết các sinh viên đã thực tập đề tài có tên đề tài là 'Truy vấn dữ liệu Multimedia'

```
SELECT *
```

```

FROM SinhVien
WHERE MaSV IN (SELECT MaSV
               FROM SV_DeTai
               WHERE MaDT=(SELECT MaDT
                           FROM DeTai
                           WHERE
                               TenDT= 'Truy vấn dữ liệu
Multimedia'))

```

Ví dụ 4.25: Câu lệnh sau đây cho biết danh sách các đề tài chưa có sinh viên thực tập.

```

SELECT *
FROM DeTai
WHERE MaDT NOT IN (SELECT DISTINCT MaDT
                   FROM SV_DeTai)

```

Ví dụ 4.26: Cho biết danh sách các đề tài có kinh phí lớn nhất?

```

SELECT *
FROM DeTai
WHERE KP = (SELECT Max(KP) FROM SV_DeTai)

```

4.2.3.2 Nhập dữ liệu.

Dữ liệu trong các bảng được thể hiện dưới các dòng (bản ghi). Để nhập thêm các dòng dữ liệu vào một bảng, ta sử dụng câu lệnh INSERT với cú pháp sau:

```

INSERT INTO <tên_bảng>[(<danh_sách_tên_cột>)]
VALUES (<danh_sách_các_giá_trị>)

```

Trong câu lệnh INSERT, danh_sách_tên_cột ngay sau tên bảng không cần thiết phải chỉ định. Trong trường hợp này, thứ tự các giá trị trong danh sách trị phải bằng số lượng các trường của bảng cần bổ sung dữ liệu cũng như phải tuân theo đúng thứ tự của các trường như khi bảng được định nghĩa.

Ví dụ 4.27: Câu lệnh dưới đây sẽ bổ sung thêm một lớp học vào bảng lớp

```

INSERT INTO LopHoc
VALUES ('L03', 'K7C-KT')

```

Ngược lại <danh_sách_tên_cột> được chỉ định sau tên bảng thì <danh_sách_các_giá_trị> phải tương ứng với thứ tự của các cột có trong danh sách.

Ví dụ:

```
INSERT INTO LopHoc(TenLop)
VALUES ('K7D-ĐTVT', 'L04')
```

Ghi chú: Trong trường hợp chỉ nhập giá trị cho một số cột trong bảng, ta phải chỉ định danh sách các cột cần nhập dữ liệu ngay sau tên bảng. Khi đó các cột không được nhập dữ liệu sẽ nhận giá trị mặc định (nếu có) hoặc nhận giá trị NULL (nếu cột cho phép nhận giá trị NULL). Nếu một cột không có giá trị mặc định và không chấp nhận giá trị NULL mà không được nhập dữ liệu, câu lệnh sẽ bị lỗi.

4.2.3.3. Cập nhật dữ liệu

Câu lệnh UPDATE trong SQL được sử dụng để cập nhật dữ liệu trong các bảng. Câu lệnh này có cú pháp như sau:

```
UPDATE <tên_bảng>
SET <tên_cột_1>=<biểu_thức_1> [, ..., <tên_cột_k> = <biểu_thức_k>]
[WHERE <điều_kiện>]
```

Sau UPDATE là tên của bảng cần cập nhật dữ liệu. Một câu lệnh UPDATE có thể cập nhật dữ liệu cho nhiều cột bằng cách chỉ định danh sách tên cột và biểu thức tương ứng sau từ khoá SET. Mệnh đề WHERE trong câu lệnh UPDATE thường được sử dụng để chỉ định các dòng dữ liệu chịu tác động của câu lệnh (nếu không chỉ định, phạm vi tác động của câu lệnh là toàn bộ các dòng trong bảng)

Ví dụ 4.28: Cập nhật lại địa chỉ của sinh viên có mã là 08K7A1 thành Thái Nguyên

```
UPDATE SinhVien
SET DiaChi = 'Thái Nguyên' WHERE MaSV ='08K7A1'.
```

Ví dụ 4.29: Cập nhật lại kết quả thực tập của các sinh viên đã thực tập đề tài có tên đề tài là 'Truy vấn dữ liệu Multimedia'.

```
UPDATE SV_DeTai
SET KQ = 10 WHERE MaDT=(SELECT MaDT
FROM DeTai
WHERE
TenDT='Truy vấn dữ liệu Multimedia').
```

4.2.3.4 Xoá dữ liệu

Để xoá dữ liệu trong một bảng, ta sử dụng câu lệnh DELETE . Cú pháp của câu lệnh như sau:

```
DELETE FROM <tên_bảng>  
[WHERE <điều_kiện>]
```

Trong đó, tên của bảng cần xoá dữ liệu được chỉ định sau DELETE FROM. Mệnh đề WHERE trong câu lệnh được sử dụng để chỉ định điều kiện đối với các dòng dữ liệu cần xoá. Nếu câu lệnh DELETE không có mệnh đề WHERE thì toàn bộ các dòng trong bảng đều bị xoá. Mệnh đề FROM chỉ định danh sách các bảng có dữ liệu liên quan đến việc xoá dữ liệu.

+Ví dụ 4.30: Câu lệnh sau sẽ xoá khỏi bảng SinhVien những sinh viên có địa chỉ ở Hà nội

```
DELETE FROM SinhVien  
WHERE DiaChi = 'Hà nội'
```

Ví dụ 4.31: Xoá khỏi bảng sinh viên những sinh viên chưa từng tham gia bất kỳ một đề tài thực tập nào?

```
DELETE FROM SinhVien  
WHERE MaSV NOT IN (SELECT DISTINCT MaSV  
FROM SV_DeTai )
```

+ Xoá tất cả các thông tin về tình hình thực tập của sinh viên

```
DELETE FROM SV_DeTai
```

BÀI TẬP CÂU HỎI

1. Cho CSDL gồm 2 quan hệ

CC(MSNCC,TEN_CC,DCCC) và MH(MSNCC,MSMH)

Trong đó:

MSNCC: Mã số người cung cấp.

TEN_CC: Tên người cung cấp

DCCC: Địa chỉ cung cấp

MSMH: Mã số mặt hàng

Hãy cho mỗi quan hệ 5 bộ dữ liệu

Hãy biểu diễn các yêu cầu sau đây bằng ngôn ngữ SQL.

a. Tìm mã số người đã cung cấp

Q1: ít nhất một mặt hàng

Q2: không cung cấp mặt hàng nào

Q3: cung cấp mặt hàng có MSMH là 15

Q4: cung cấp ít nhất một mặt hàng nhưng không có mặt hàng có mã số là 15

b. Mặt hàng có mã số là 12, 13, 15 được cung cấp bởi các nhà cung cấp địa chỉ nào?

c. Lập danh sách gồm các cột MSNCC, TEN_CC,MSMH) từ cơ sở dữ liệu trên

2. Cho CSDL gồm các quan hệ sau:

DAIHOC(TENTRUONG,HIEUTRUONG,DIACHI)

KHOA(TENTRUONG,MSKHOA,TENKHOA,SOSINHVIENT)

SINHVIENT(TENTRUONG,MSKHOA,MSSV,TENSV,DIACHISV)

trong đó:

SOSINHVIENT: số lượng là sinh viên

MSKHOA: mã số khoa

TENSV: tên sinh viên

DIACHISV: địa chỉ của sinh viên (hiểu là quê quán)

Hãy cho mỗi quan hệ 5 bộ dữ liệu

Biểu diễn các câu hỏi sau đây bằng ngôn ngữ SQL

a/ Trường Đại học nào có khoa TINHOC

b/ Tổng số sinh viên học ở tất cả các trường đại học.

c/ Sinh viên nào học tại quê nhà (giả sử lấy tên tỉnh, thành phố)

d/ Khoa nào của trường có số sinh viên cao nhất?

e/ Cho biết tên hiệu trưởng của các trường có khoa TINHOC

3. Cho CSDL với các quan hệ:

NHANVIEN(MSNV,TENNHANVIEN,MSCOQUAN,CONGVIEC,
THUTRUONG,LUONG)

COQUAN(MSCOQUAN,TENCOQUAN,DIACHI)

Hãy cho mỗi quan hệ 5 bộ dữ liệu

Biểu diễn bằng ngôn ngữ SQL, và đại số quan hệ yêu cầu sau đây:

Q1: Tìm tên những nhân viên ở cơ quan có mã số là 50

Q2: Tìm mã số tất cả các cơ quan từ quan hệ NHANVIEN

Q3: Tìm tên các nhân viên cơ quan có mã số là 15, 20, 25

Q4: Tìm tên những người làm việc ở Đồ Sơn

4. Cho cơ sở dữ liệu quản lý dự án gồm các bảng dữ liệu sau:

+ Bảng NHANVIEN chứa danh sách các nhân viên gồm các thuộc tính sau:

Tên Thuộc tính	Giải thích
MaNV	Mã nhân viên
Hoten	HỌ tên nhân viên
Ngaysinh	Ngày sinh
GT	Giới tính

+ Bảng DU_AN chứa thông tin về các dự án gồm có các thuộc tính sau:

Tên Thuộc tính	Giải thích
MaDA	Mã dự án
TenDA	Tên dự án
NganSach	Ngân sách

+ Bảng THAMGIA ghi danh sách sinh viên đăng ký tham gia dự án

Tên Thuộc tính	Giải thích
MaDA	Mã dự án
MaNV	Mã nhân viên
TGBD	Thời gian bắt đầu
TGKT	Thời gian kết thúc

Biểu diễn bằng ngôn ngữ SQL, và đại số quan hệ các yêu cầu sau đây:

a/ Đưa ra danh sách các nhân viên ?

b/ Đưa ra danh sách các nhân viên có giới tính bằng 1 ?

- c/ Đưa ra danh sách các dự án có ngân sách lớn nhất ?
- d/ Cho biết mỗi nhân viên đã tham gia tổng số bao nhiêu dự án ?
- e/ Cho biết mỗi dự án có tổng số bao nhiêu nhân viên tham gia ?
- f/ Cho biết mã và tên của các dự án có số tổng số nhân viên tham gia ≥ 10 người ?
- g/ Cho biết mã và tên của các nhân viên đã tham gia dự án có tên dự án là ‘Dự án nước sạch nông thôn’ ?
- h/ Đưa ra mã và tên các dự án mà nhân viên có mã là NV01 đã tham gia ?
- i/ Đưa ra danh sách các nhân viên chưa tham gia bất kỳ dự án nào ?
- k/ Cho biết thông tin về các nhân viên đã tham gia ít nhất một dự án ?
- m/ Đưa ra danh sách các dự án có sự sắp xếp giảm dần theo ngân sách ?
5. Cho cơ sở dữ liệu gồm các bảng dữ liệu sau:

+ Bảng NSX (nước sản xuất)

Tên Thuộc tính	Giải thích
MaNSX	Mã nhà sản xuất
TenNSX	Tên nhà sản xuất

+ Bảng SANPHAM (sản phẩm)

Tên Thuộc tính	Giải thích
MaSP	Mã sản phẩm
TenSP	Tên sản phẩm
DVT	Đơn vị tính
NgaySX	Ngày sản xuất
SoLuong	Số lượng
ChungLoai	Chủng loại
MaNSX	Mã nhà sản xuất

Biểu diễn bằng ngôn ngữ SQL, và đại số quan hệ các yêu cầu sau đây:

- a) Cho biết mã và tên của nhà sản xuất đã sản xuất sản phẩm có tên là ‘Máy lọc nước’?
- b) Cho biết mã, tên, số lượng, ngày sản xuất của các sản phẩm do nhà sản xuất có mã là ‘N01’ đã sản xuất?
- c) Hãy tổng hợp thông tin về từng loại sản phẩm của mỗi nhà sản xuất (gồm các thuộc tính sau: MaSP, TenSP, DVT, TongSoLuong)?
- d) Cho biết mỗi nhà sản xuất đã sản xuất tổng số bao nhiêu loại sản phẩm?

e) Cho biết danh sách các sản phẩm do nhà sản xuất có tên là 'Panasonic' đã sản xuất?

6. Cho cơ sở dữ liệu quản lý điểm gồm các bảng sau:

+ Bảng LopHoc gồm các thuộc tính

Tên Thuộc tính	Giải thích
MaLop	Mã lớp
TenLop	Tên lớp

+Bảng SinhVien gồm các thuộc tính

Tên Thuộc tính	Giải thích
MaSV	Mã sinh viên
HoTen	Họ và tên sinh viên
NS	Ngày sinh
GT	Giới tính
DC	Địa chỉ
MaLop	Mã lớp

+ Bảng MonHoc gồm các thuộc tính

Tên Thuộc tính	Giải thích
MaMon	Mã môn
TenMon	Tên môn
TC	Số tín chỉ

+Bảng Diem gồm các thuộc tính

Tên Thuộc tính	Giải thích
MaSV	Mã sinh viên
MaMon	Mã môn
Ky	Kỳ thi
DiemLan1	Điểm thi lần 1
DiemLan2	Điểm thi lần 2

Hãy biểu diễn các yêu cầu sau đây bằng ngôn ngữ SQL:

a) Cho biết danh sách các sinh viên có giới tính bằng 1 ?

b) Cho biết danh sách các sinh viên có địa chỉ ở Thái Nguyên?

c) Cho biết mã và tên và điểm thi lần 1 của các sinh viên đã học môn học có mã là MH01?

d) Cho biết mã và tên và điểm thi lần 1 của các sinh viên đã học môn học có tên môn là Cơ sở dữ liệu?

- e) Cho biết danh sách các sinh viên phải thi lại môn học có tên môn là Cơ sở dữ liệu?
- f) Cho biết sinh viên có mã là SV01 phải thi lại những môn học nào?
- g) Cho biết điểm cao thi cao nhất của môn học Hệ quản trị cơ sở dữ liệu là bao nhiêu?
- h) Cho biết những sinh viên đạt điểm thi cao nhất của môn Cơ sở dữ liệu?

Chương 5

TỐI ƯU HOÁ CÂU HỎI

Nói chung các ngôn ngữ bậc cao, đòi hỏi thực hiện trong máy tính đều rất tốn kém thời gian. Do vậy trước khi thực hiện các câu hỏi thuộc các ngôn ngữ đó cần thiết phải biến đổi hợp lý để giảm thời gian tính toán. Việc làm đó tạm thời gọi là "tối ưu hoá".

Trong chương này chủ yếu trình bày một vài phương pháp tối ưu hoá các biểu thức quan hệ, đặc biệt là xử lý biểu thức có liên quan tới phép kết nối và phép tích ĐỀ - các. Sau đó sẽ trình bày chi tiết một phương pháp tối ưu hoá một lớp phổ cập của các biểu thức quan hệ.

5.1 Các chiến lược tối ưu

1- Thực hiện phép chọn sớm nhất như có thể

Biến đổi câu hỏi để đưa phép chọn vào thực hiện trước nhằm làm giảm bớt kích cỡ của kết quả trung gian và do vậy chi phí phải trả giá cho việc truy nhập bộ nhớ thứ cấp cũng như lưu trữ của bộ nhớ chính sẽ nhỏ đi.

2-Tổ hợp những phép chọn với phép tích ĐỀ-các thành phép kết nối

Như đã biết, phép kết nối, đặc biệt là phép kết nối bằng được thực hiện "rẻ" hơn là thực hiện các phép tích ĐỀ-các trên cùng các quan hệ. Nếu kết quả của tích ĐỀ-các $R \times S$ là đối số của phép chọn và phép chọn liên quan tới các phép so sánh giữa các thuộc tính của R và S thì rõ ràng phép tích ĐỀ-các là phép kết nối.

3-Tổ hợp dãy các phép tính một ngôi thành một

Một dãy các phép một ngôi như phép chọn hoặc phép chiếu mà kết quả của chúng phụ thuộc vào các bộ của một quan hệ độc lập thì có thể nhóm các phép đó lại.

4-Tìm các biểu thức con chung trong một biểu thức

$$F = F_1 \times F_2 \cup F_1 \times F_3 \Rightarrow F = F_1 \times (F_2 \cup F_3)$$

5-Xử lý độc lập các tập trước khi xử lý chung CSDL

Có thể sắp xếp và thiết lập các tập chỉ số cho từng quan hệ độc lập trước khi xử lý

6- Lựa chọn thứ tự thực hiện các phép toán.

Một khi cần chọn trình tự thực hiện các phép tính trong biểu thức hoặc chọn một trong hai đối số của một phép hai ngôi cần tính toán xem chi phí thực hiện các phép tính đó (thường là số phép tính, thời gian, dung tích bộ nhớ theo một tỷ lệ giữa kích cỡ các quan hệ...). Từ đó sẽ có được các chi phí phải trả cho các cách khác nhau để thực hiện các câu hỏi

5.2 Các phép biến đổi tương đương

1-Phép giao hoán

Nếu R_1 và R_2 là hai quan hệ, F là điều kiện trên các thuộc tính của R_1 và R_2 thì

$$R_1 \begin{array}{c} \diagup \\ F \\ \diagdown \end{array} R_2 \equiv R_2 \begin{array}{c} \diagdown \\ F \\ \diagup \end{array} R_1 ; R_1 * R_2 \equiv R_2 * R_1 ; R_1 \times R_2 \equiv R_2 \times R_1$$

2- Phép kết hợp

Nếu R_1, R_2 và R_3 là các quan hệ, F_1 và F_2 là các biểu thức điều kiện thì

$$(R_1 \times R_2) \times R_3 \equiv R_1 \times (R_2 \times R_3);$$

$$(R_1 \begin{array}{c} \diagup \\ F_1 \\ \diagdown \end{array} R_2) \begin{array}{c} \diagdown \\ F_2 \\ \diagup \end{array} R_3 \equiv R_1 \begin{array}{c} \diagup \\ F_1 \\ \diagdown \end{array} R_2 \begin{array}{c} \diagdown \\ F_2 \\ \diagup \end{array} R_3$$

3- Xử lý dãy các phép toán chọn

$$\sigma_{F_1}(\sigma_{F_2}(\sigma_{F_3} \dots (\sigma_{F_n}(I)))) \equiv \sigma_{F_1 \wedge F_2 \wedge F_3 \wedge \dots \wedge F_n}(I)$$

4- Xử lý dãy các phép toán chiếu

$$\text{Nếu có: } A_1 A_2 \dots A_n \subseteq B_1 B_2 \dots B_n \text{ Thì } \Pi_{A_1 A_2 \dots A_n}(\Pi_{B_1 B_2 \dots B_n}(R)) \equiv \Pi_{A_1 A_2 \dots A_n}(R)$$

5- Giao hoán phép chọn và phép chiếu

$$\sigma_F(\prod_{A_1 A_2 \dots A_n} (R)) \equiv \prod_{A_1 A_2 \dots A_n} (\sigma_F(R))$$

6-Giao hoán giữa phép chọn và phép tích ĐỀ -các

-Nếu các điều kiện chọn chỉ liên quan đến R1

$$\sigma_F(R_1 \times R_2) \equiv \sigma_{F_1}(R_1) \times (R_2)$$

-Nếu $F = F_1 \wedge F_2$ trong đó F_1 chỉ liên quan đến R_1 ; F_2 chỉ liên quan đến R_2 .

$$\sigma_F(R_1 \times R_2) \equiv \sigma_{F_1}(R_1) \times \sigma_{F_2}(R_2)$$

-Nếu F_1 chỉ liên quan đến R_1 , F_2 liên quan đến cả R_1 và R_2

$$\sigma_F(R_1 \times R_2) \equiv \sigma_{F_2}(\sigma_{F_1}(R_1) \times R_2)$$

7- Giao hoán giữa phép chọn và một phép hợp

$$\sigma_F(R_1 \cup R_2) \equiv \sigma_F(R_1) \cup \sigma_F(R_2)$$

8- Giao hoán giữa một phép chọn và một phép trừ

$$\sigma_F(R_1 - R_2) \equiv \sigma_F(R_1) - \sigma_F(R_2)$$

9- Giao hoán giữa một phép chiếu và phép tích ĐỀ các

Nếu các thuộc tính $A_1, A_2, \dots, A_n \in R_1$; $B_1, B_2, \dots, B_n \in R_2$

Ta có: $\prod_{A_1 A_2 \dots A_n, B_1 B_2 \dots B_n} (R_1 \times R_2) \equiv \prod_{A_1 A_2 \dots A_n} (R_1) \times \prod_{B_1 B_2 \dots B_n} (R_2)$

10-Giao hoán giữa một phép chiếu và một phép hợp

$$\prod_{A_1 A_2 \dots A_n} (R_1 \cup R_2) \equiv \prod_{A_1 A_2 \dots A_n} (R_1) \cup \prod_{A_1 A_2 \dots A_n} (R_2)$$

Ví dụ: Cho cơ sở dữ liệu gồm các bảng dữ liệu sau

Bảng Công Ty (CONGTy) gồm các thuộc tính: Mã công ty (MaCongTy), Tên công ty (TenCongTy), Ngân sách (NganSach), Địa chỉ (DiaChi).

Bảng Hàng Hoá (HANGHOA) gồm các thuộc tính: Mã hàng (MaHang), Tên hàng (TenHang), Màu sắc (Mau), Đơn vị tính (DonViTinh).

Bảng Cung Cấp hàng (CUNGCAP) gồm các thuộc tính: MaCongTy, MaHang, Số lượng (SoLuong), Đơn giá (DonGia).

Yêu cầu: Cho biết tên công ty cung cấp mặt hàng màu đỏ

1/ Dùng đại số quan hệ

$$\prod_{TenCongTy} (\sigma_{Mau = "đỏ"} (\sigma_{CongTy.MaCongTy = CungCap.MaCongTy}$$

$(\sigma_{\text{CungCap.MaHang=HangHoa.MaHang}}(\text{CONGTY x HANGHOA x CUNGCAP})))$

-Đẩy phép chọn lên trước

$\Pi_{\text{TenCongTy}}(\sigma_{\text{Mau="đỏ"}}(\sigma_{\text{CongTy.MaCongTy = CungCap.MaCongTy}}(\text{CongTy x } \sigma_{\text{CungCap.MaHang=HangHoa.MaHang}}(\text{HangHoa x CungCap}))))$

-Chuyển phép chọn và phép tích Đề các thành phép kết nối

$\Pi_{\text{TenCongTy}}(\sigma_{\text{Mau="đỏ"}}(\sigma_{\text{CongTy.MaCongTy = CungCap.MaCongTy}}(\text{CongTy x (HangHoa * CungCap)})))$

MaHang

-Chuyển phép chọn và phép tích Đề các thành phép kết nối

$\Pi_{\text{TenCongTy}}(\sigma_{\text{Mau="đỏ"}}(\text{CongTy * (HangHoa * CungCap)}))$
MaCongTy MaHang

-Đẩy phép chọn lên trước

$\Pi_{\text{TenCongTy}}(\text{CongTy * } (\sigma_{\text{Mau="đỏ"}}\text{HangHoa}) * \text{CungCap}))$
MaHang MaCongTy

2/ Dùng SQL

-Không tối ưu:

```
SELECT TenCongTy
FROM Congty, HangHoa, CungCap
WHERE (CongTy.MaCongTy = CungCap.MaCongTy) AND
      (CungCap.MaHang = HangHoa.MaHang) AND (HangHoa. Mau = "ĐỎ")
```

- Tối ưu:

```
SELECT TenCongTy
FROM CongTy
WHERE MaCongTy IN ( SELECT MaCongTy
                    FROM CungCap
                    WHERE MaHang IN ( SELECT MaHang
                                      FROM HangHoa
                                      WHERE Mau = "ĐỎ" ))
```