

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

Biên soạn: Trần Thanh

CƠ SỞ DỮ LIỆU

(Dùng cho sinh viên hệ đại học)

Thái Nguyên 2020

LỜI NÓI ĐẦU

Tài liệu “Cơ sở dữ liệu” là sách hướng dẫn học tập dùng cho sinh viên đại học ngành kỹ thuật máy tính và sự phạm kỹ thuật tin.

Nội dung của tài liệu bao gồm:

Chương I. Giới thiệu sự cần thiết tổ chức dữ liệu theo mô hình hệ cơ sở dữ liệu, mục tiêu và tính độc lập của dữ liệu. Kiến trúc mô hình tổng quát 3 lớp và tính ổn định trong mô hình quan niệm. Các mô hình truy xuất thông dụng hiện nay.

Chương II. Giới thiệu tổng quát về các mô hình cơ sở dữ liệu kinh điển và truyền thống. Đó là các khái niệm cơ bản về mô hình dữ liệu mạng, mô hình phân cấp, mô hình quan hệ và mô hình thực thể – liên hệ. Một mô hình CSDL phải có khả năng biểu diễn thực thể và liên kết giữa các thực thể, là cách nhìn và cách biểu diễn các liên kết của người sử dụng.

Chương III. Đề cập đến các ngôn ngữ con thao tác dữ liệu: ngôn ngữ con dữ liệu dựa trên đại số quan hệ.

Chương IV. Đề cập đến các ngôn ngữ con thao tác dữ liệu: Ngôn ngữ có cấu trúc SQL truy vấn dữ liệu.

Chương V. Trình bày những khái niệm cơ bản về lý thuyết cơ sở dữ liệu quan hệ do E.F Codd đề xuất. Mô hình cơ sở dữ liệu quan hệ được xây dựng trên cơ sở toán học chặt chẽ và logic, có tính độc lập dữ liệu cao và có cấu trúc biểu diễn đơn giản. Giới thiệu quá trình chuẩn hoá dữ liệu, là quá trình tách không tổn thất thông tin các quan hệ chưa chuẩn hoá về các quan hệ có dạng chuẩn 3NF và Boye –Codd theo nghĩa các quan hệ gốc được khôi phục chính xác từ các phép kết nối tự nhiên các quan hệ chiếu, với giả thiết các phụ thuộc dữ liệu là các phụ thuộc hàm.

CHƯƠNG I.

KHÁI NIỆM CƠ BẢN VỀ CÁC HỆ CƠ SỞ DỮ LIỆU

(Số tiết lý thuyết 4 tiết)

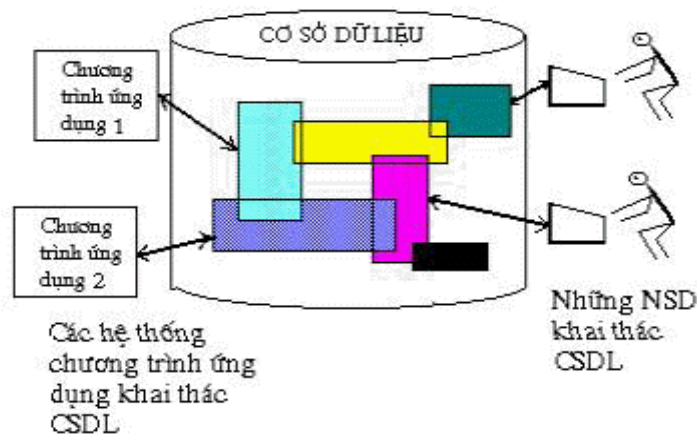
1.1. CƠ SỞ DỮ LIỆU LÀ GÌ?

Cơ sở dữ liệu là một bộ sưu tập rất lớn về các loại dữ liệu tác nghiệp, bao gồm các loại dữ liệu âm thanh, tiếng nói, chữ viết, văn bản, đồ hoạ, hình ảnh tĩnh hay hình ảnh động....được mã hoá dưới dạng các chuỗi bit và được lưu trữ dưới dạng File dữ liệu trong các bộ nhớ của máy tính. Cấu trúc lưu trữ dữ liệu tuân theo các quy tắc dựa trên lý thuyết toán học. Cơ sở dữ liệu phản ánh trung thực thế giới dữ liệu hiện thực khách quan.

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

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

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



Hình 1.2.1 Sơ đồ tổng quát về một cơ sở dữ liệu

1.2 SỰ CẦN THIẾT CỦA CÁC HỆ CƠ SỞ DỮ LIỆU

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

Giảm bớt dư thừa dữ liệu trong lưu trữ: Trong các ứng dụng lập trình truyền thống, phương pháp tổ chức lưu trữ dữ liệu vừa tốn kém, lãng phí bộ nhớ và các thiết bị lưu trữ, vừa dư thừa thông tin lưu trữ. Nhiều chương trình ứng dụng khác nhau cùng xử lý trên các dữ liệu như nhau, dẫn đến sự dư thừa đáng kể về dữ liệu.

Ví dụ trong các bài toán nghiệp vụ quản lý "Cước thuê bao điện thoại" và "Doanh thu & sản lượng", tương ứng với mỗi một chương trình là một hay nhiều tệp dữ liệu được lưu trữ riêng biệt, độc lập với nhau. Trong cả 2 chương trình cùng xử lý một số thuộc tính của một cuộc đàm thoại như "số máy gọi đi", "số máy gọi đến", "hướng cuộc gọi", "thời gian bắt đầu" và "thời gian kết thúc"Nhiều thuộc tính được mô tả và lưu trữ nhiều lần độc lập với nhau. Nếu tổ chức lưu trữ theo lý thuyết CSDL thì có thể hợp nhất các tệp lưu trữ của các bài toán trên, các chương trình ứng dụng có thể cùng chia sẻ tài nguyên trên cùng một hệ CSDL.

Tổ chức lưu trữ dữ liệu theo lý thuyết CSDL sẽ tránh được sự không nhất quán trong lưu trữ dữ liệu và bảo đảm được tính toàn vẹn của dữ liệu: Nếu một thuộc tính được mô tả trong nhiều tệp dữ liệu khác nhau và lặp lại nhiều lần trong các bản ghi, khi thực hiện việc cập nhật, sửa đổi, bổ sung sẽ không sửa hết nội dung các mục đó. Nếu dữ liệu càng nhiều thì sự sai sót khi cập nhật, bổ sung càng lớn. Khả năng xuất hiện mâu thuẫn, không nhất quán thông tin càng nhiều, dẫn đến không nhất quán dữ liệu trong lưu trữ. Tất yếu kéo theo sự dị thường thông tin, thừa, thiếu và mâu thuẫn thông tin. Thông thường, trong một thực thể, giữa các thuộc tính có mối quan hệ ràng buộc lẫn nhau, tác động ảnh hưởng lẫn nhau. Cước của một cuộc đàm thoại phụ thuộc vào khoảng cách và thời gian cuộc gọi, tức là phụ thuộc hàm vào các thuộc tính máy gọi đi, máy gọi đến, thời

gian bắt đầu và thời gian kết thúc cuộc gọi. Các trình ứng dụng khác nhau cùng xử lý cuộc đàm thoại trên các thực thể lưu trữ tương ứng khác nhau chưa hẳn cho cùng một kết quả về sản lượng phút và doanh thu. Điều này lý giải tại sao trong một doanh nghiệp, cùng xử lý trên các chỉ tiêu quản lý mà số liệu báo cáo của các phòng ban, các công ty con lại cho các kết quả khác nhau, thậm chí còn trái ngược nhau. Như vậy, có thể khẳng định, nếu dữ liệu không tổ chức theo lý thuyết cơ sở dữ liệu, tất yếu không thể phản ánh thế giới hiện thực dữ liệu, không phản ánh đúng bản chất vận động của dữ liệu.

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

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

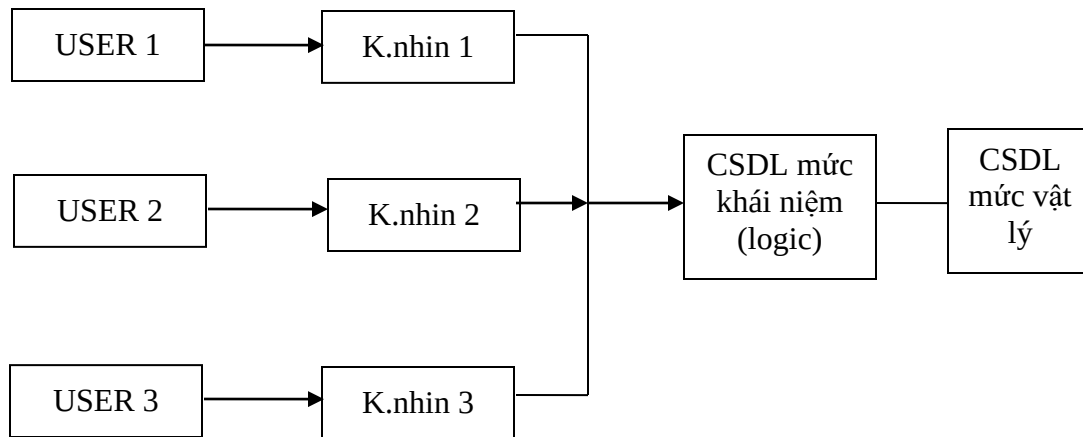
Tổ chức dữ liệu theo lý thuyết cơ sở dữ liệu sẽ thống nhất các tiêu chuẩn, thủ tục và các biện pháp bảo vệ, an toàn dữ liệu: Các hệ CSDL sẽ được quản lý tập trung bởi một người hay một nhóm người quản trị CSDL, bằng các hệ quản trị CSDL. Người quản trị CSDL có thể áp dụng thống nhất các tiêu chuẩn, quy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

1.3. TÍNH ĐỘC LẬP DỮ LIỆU

Tính độc lập của dữ liệu là tính chất quan trọng nhất của cơ sở dữ liệu, với 1 hệ quản trị cơ sở dữ liệu thì tính độc lập dữ liệu phải được đảm bảo.

- **Lược đồ vật lý** có thể thay đổi do người quản trị CSDL mà không cần thay đổi lược đồ con. Việc tổ chức lại CSDL vật lý có thể làm thay đổi hiệu quả tính toán của các chương trình ứng dụng nhưng không đòi hỏi phải viết lại chương trình đó. Tính độc lập này gọi là độc lập dữ liệu mức vật lý.
- **Độc lập dữ liệu logic:** Khi sử dụng một CSDL, có thể cần thiết phải thay đổi lược đồ khái niệm như thêm thông tin về các loại khác nhau của thực thể đang tồn tại trong CSDL. Việc thay đổi lược đồ khái niệm không làm ảnh hưởng tới các lược đồ con đang tồn tại, do đó không cần thiết phải thay đổi các chương trình ứng dụng.

Định nghĩa tính độc lập dữ liệu là sự bất biến của các chương trình ứng dụng đối với các thay đổi cấu trúc lưu trữ và chiến lược truy nhập điều này có nghĩa là khi có sự thay đổi về cấu trúc thì không phải viết lại chương trình.



Theo hình 1.1 từ khung nhìn tới CSDL mức khái niệm và CSDL mức vật lý cho thấy hai mức *độc lập tư liệu*.

1.4. CÁC TIÊU CHUẨN CỦA CSDL VÀ HQTCSDL

1.4.1. Các tiêu chuẩn của CSDL (đảm bảo 5 tiêu chuẩn)

- Biểu diễn tốt thế giới thực tức là có khả năng cung cấp 1 hình ảnh trung thực của thực tại trong mọi lúc.
- Không dư thừa thông tin tức là có sự lặp lại thông tin trong CSDL.
- Tính độc lập của các chương trình đối với dữ liệu.
- Tính an toàn và bí mật của dữ liệu (điều này quan trọng khi một dữ liệu có nhiều người cũng sử dụng).
- Hiệu suất sử dụng tức là các ứng dụng phải có hiệu suất cao đối với cơ sở dữ liệu giống như sử dụng các nguồn thông tin truyền thống.

1.4.2. Hệ quản trị cơ sở dữ liệu (SQL Server, Oracle.....)

Các tính năng của HQTCSDL(có 4 tính năng)

- Các thao tác cơ sở: bổ sung, sửa, xoá, tìm kiếm...
- Đảm bảo tính độc lập của dữ liệu.
- Cung cấp các thủ tục cho phép sao lưu và phục hồi.
- Cung cấp các thủ tục điều khiển sự cạnh tranh.

Hệ cơ sở dữ liệu là hệ đa người dùng: gồm 4 thành phần

- ① Cơ sở dữ liệu: Dữ liệu trong CSDL là hợp nhất và được sử dụng chung, các dữ liệu này phải được kiểm soát sao cho tính dư thừa là tối thiểu.

② Người sử dụng (UESR) gồm có:

- Người sử dụng cuối (END - USER): Là những người sử dụng thiết bị đầu cuối bằng bàn phím.
- Người viết chương trình ứng dụng: Là những người viết những câu lệnh để truy cập cơ sở dữ liệu.
- Người quản trị cơ sở dữ liệu: là những người có nhiệm vụ điều khiển toàn bộ hệ CSDL.

③ Phần mềm: Là chương trình để có thể xử lý, thay đổi dữ liệu trong cơ sở dữ liệu (hệ quản trị cơ sở dữ liệu).

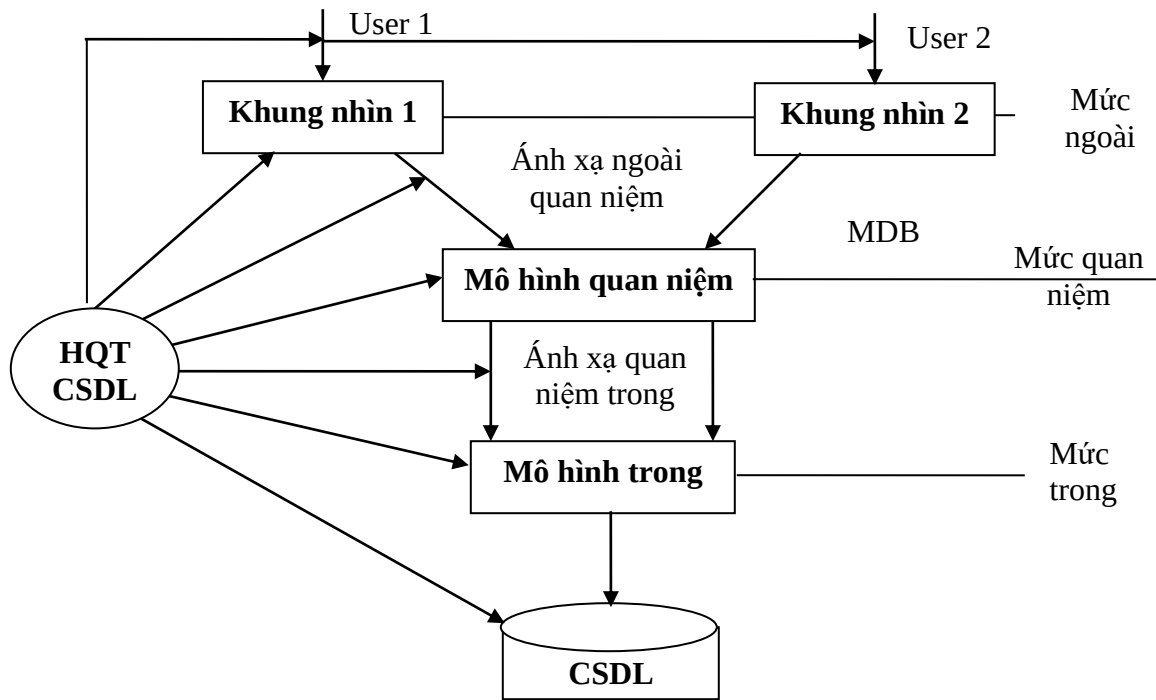
④ Phần cứng: Gồm những thiết bị nhớ thứ cấp được sử dụng để lưu trữ cơ sở dữ liệu.

1.5. KIẾN TRÚC CỦA HỆ CSDL

Kiến trúc chuẩn của một hệ CSDL gồm 3 mức: Mức trong (còn gọi là mức vật lý - *Physical*), mức quan niệm (*Conception* hay *Logical*) và mức ngoài.

- Trong đó khung nhìn gọi là mức ngoài là nơi người sử dụng nhìn vào CSDL gồm tập tất cả các dữ liệu mà người sử dụng được phép truy nhập cùng với các thao tác được sử dụng. Làm việc tại mức này có các nhà chuyên môn, các kỹ sư tin học và những người sử dụng không chuyên.
- Mô hình quan niệm ứng với mức quan niệm là tập các dữ liệu được bỏ dờ ít nhiều dưới dạng trừu tượng vì vậy phải có ánh xạ ngoài quan niệm để chỉ sự tương ứng giữa khung nhìn và mô hình quan niệm. Tại mức này sẽ giải quyết cho câu hỏi CSDL cần phải lưu giữ bao nhiêu loại dữ liệu? đó là những dữ liệu gì? Mối quan hệ giữa các loại dữ liệu này như thế nào?
- Mô hình trong là tập các dữ liệu được bỏ dờ dưới dạng gần với lưu trữ thực của nó trong các thiết bị nhớ thứ cấp. Đây là mức lưu trữ CSDL, tại mức này vấn đề cần giải quyết là dữ liệu gì và được lưu trữ như thế nào? ở đâu (đĩa từ, băng từ, track, sector ... nào)? Cần các chỉ mục gì? Việc truy xuất là tuần tự (*Sequential Access*) hay ngẫu nhiên (*Random Access*) đối với từng loại dữ liệu.
- Ánh xạ quan niệm trong để chỉ sự tương ứng giữa mô hình quan niệm và mô hình trong.
- MDB_người quản trị có các nhiệm vụ sau:
 - Là người quyết định nội dung thông tin của CSDL.
 - Là người quyết định cấu trúc lưu trữ và chiến lược truy nhập.
 - Cùng với sự giúp đỡ của người sử dụng mô tả các khung nhìn.
 - Là người đưa ra các phương pháp sao chép phục hồi.

- Là người đưa ra cơ chế điều khiển cạnh tranh.
- Là người kiểm soát thẩm quyền của người sử dụng và kiểm tra tính đúng đắn của dữ liệu.



Kiến trúc tổng quát của một cơ sở dữ liệu

1.6. PHÂN LOẠI CÁC HỆ CƠ SỞ DỮ LIỆU

Với hệ cơ sở dữ liệu tập chung chia làm 3 loại:

- Loại 1: Hệ cơ sở dữ liệu một người dùng (Terminal computer database) trong hệ này chỉ có một máy tính duy nhất chúng vừa là đầu vào vừa là đầu cuối thích ứng với những cơ sở dữ liệu nhỏ không phức tạp.
- Loại 2: Cơ sở dữ liệu đa người dùng (Center computer database) CSDL được đặt ở các máy tính trung tâm các thiết bị đầu cuối được nối với máy tính trung tâm, người sử dụng truy nhập CSDL từ các thiết bị đầu cuối.

Hệ cơ sở dữ liệu bao gồm 4 thành phần:

- CSDL là nơi hợp nhất lưu trữ toàn bộ dữ liệu xí nghiệp cho phép nhiều người dùng chung và tính dư thừa dữ liệu được kiểm soát tối đa.
- Người sử dụng (được chia làm 3 lớp), những người dùng cuối là người sử dụng truy nhập vào hệ từ các thiết bị đầu cuối với các CSDL và các thao tác được phép. Người thứ hai là người viết chương trình ứng dụng. Người

thứ 3 là người quản trị CSDL có thể một hoặc một nhóm là những người có quyền cao nhất và điều khiển toàn bộ hệ.

- Phần mềm của hệ: là các chương trình quản trị CSDL được viết bằng một hệ quản trị CSDL nào đó.
- Phần cứng của hệ: bao gồm các thiết bị nhớ thứ cấp dùng để lưu trữ CSDL của hệ.

 Loại 3: CSDL khách phục vụ (Client/ Server Database) CSDL được đặt tại máy phục vụ (server) các máy tính khách (client) được nối với máy phục vụ thông qua mạng Lan, mọi thao tác của người phục vụ đều được thực hiện trên các máy tính khách, chỉ khi cần truy xuất CSDL đặt tại máy chủ thì mới làm việc với máy chủ do đó máy chủ không yêu cầu phải mạnh.

CHƯƠNG II.

CÁC MÔ HÌNH DỮ LIỆU

(Số tiết lý thuyết 3 tiết)

Sự cần thiết tổ chức lưu trữ dữ liệu theo một cách thức xác định và chặt chẽ đã dẫn đến sự phát triển các mô hình dữ liệu. Từ những mô hình mạng, mô hình phân cấp và mô hình dữ liệu quan hệ là những mô hình cơ sở dữ liệu kinh điển, truyền thống cho đến các mô hình cơ sở dữ liệu phân tán, cơ sở dữ liệu hướng đối tượng...là những mô hình dữ liệu hiện đại được áp dụng nhiều trên thị trường hiện nay. Đối tượng nghiên cứu các hệ CSDL là các thực thể và các mối liên kết giữa các thực thể. Một mô hình CSDL phải có khả năng biểu diễn thực thể và liên kết giữa các thực thể. Các liên kết là một dạng đặc biệt của thực thể. Các cách tiếp cận CSDL là các cách nhìn và các cách biểu diễn liên kết của người sử dụng.

Nghiên cứu mô hình cơ sở dữ liệu dựa trên các yêu cầu sau:

1. *Mục tiêu độc lập dữ liệu:* Phải xác định rõ ràng các khía cạnh logic và khía cạnh vật lý của việc quản trị cơ sở dữ liệu, bao gồm việc thiết kế các hệ cơ sở dữ liệu, các thao tác và tìm kiếm dữ liệu bằng các công cụ ngôn ngữ con dữ liệu.
2. *Mục tiêu trao đổi:* Mô hình dữ liệu đơn giản về cấu trúc, sao cho người sử dụng có cách nhìn trong suốt khi truy nhập vào các hệ cơ sở dữ liệu và có khả năng trao đổi với nhau về cơ sở dữ liệu.
3. *Mục tiêu xử lý tập:* Người sử dụng có thể sử dụng ngôn ngữ bậc cao để biểu diễn các phép toán trên các mảng thông tin, kỹ thuật xử lý theo lô (batch), mà không phải xử lý tuần tự theo từng bản ghi.
4. *Mô hình được xây dựng trên cơ sở lý thuyết vững chắc, chặt chẽ*

2.1. CÁC MÔ HÌNH DỮ LIỆU

2.1.1. Mô hình dữ liệu quan hệ

Mô hình được xác định dựa trên khái niệm lý thuyết tập hợp các quan hệ tức là tập các k bộ với k cố định (mỗi một bộ là một dòng hoặc một bản ghi), dữ liệu được biểu diễn dưới dạng bảng như sau:

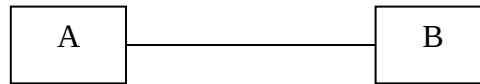
- Thực thể biến đổi thành bảng, tên thực thể thành tên các bảng.
- Các thuộc tính của thực thể biến đổi thành các cột trong bảng (thuộc tính là 1 thông tin nhỏ về vấn đề gì đó).

Ví dụ: Học sinh A (tên, noisinh, ngaysinh, đối tượng, ...)

tên	}	Là thuộc tính
noisinh		
ngaysinh		
đoituong		

- Với mỗi thực thể phải xác định khoá của bảng.

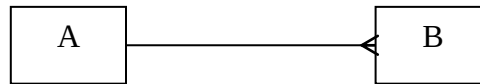
- Phải xác định các kiểu liên kết giữa các thực thể (có 3 kiểu liên kết: một-một, một - nhiều, nhiều - nhiều).
- Thiết lập liên kết
 - + Liên kết 1-1



Thiết lập bằng cách lấy khóa của đầu này đặt vào đầu kia.

$A(K_A, \dots, K_B), B(K_B, \dots)$

- + Liên kết 1 - nhiều



Thiết lập bằng cách lấy khóa của đầu 1 đặt vào đầu nhiều.

$A(K_A, \dots), B(K_B, K_A)$

- + Liên kết nhiều -n nhiều



Liên kết nhiều - nhiều có thể được xem là trường hợp đặc biệt của hai liên kết một nhiều.

$A(K_A, \dots), B(K_B, \dots), AB(K_A, K_B, \dots)$

Ví dụ: Có CSDL KHOA gồm 2 thực thể: thực thể khoa và thực thể sinh viên.

Thực thể khoa có 3 thuộc tính: Makhoa, Tenkhoa, Diadiem

Thực thể sinhvien gồm 4 thuộc tính: Masv, tensv, đchi, Makhoa.

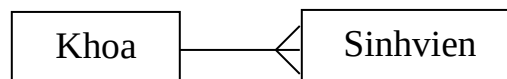
Khoa

<u>Makhoa</u>	Tenkhoa	Diadiem
KDT	Điện Tử	A4
KCB	Cơ bản	A2
KD	Điện	A3
KC	Cơ	A1

Sinhvien

<u>MaSV</u>	<i>Ten sv</i>	<i>Điachi</i>	<i>Makhoa</i>
SV01	Trần Quân	Thái Nguyên	KDT
SV02	Hoàng Thị Xuân	Hà Nội	KC
SV03	Đặng Ngọc Chiến	Quảng Ninh	KCB
SV04	Trần Nam	Sơn La	KDT
SV05	Nguyễn Thị Cao	Thái Nguyên	KDT
SV06	Đỗ Hữu Ngọc	Yên Bái	KCB
SV07	Hoàng Thao	Quảng Ninh	KD

Vậy liên kết: Quan hệ Sinhvien và Khoa là quan hệ một nhiều (vì một khoa có thể có nhiều sinh viên).

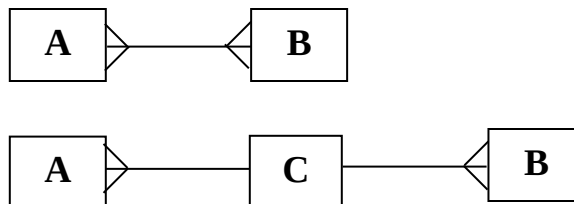


Mô hình trong ví dụ trên là mô hình dữ liệu quan hệ sử dụng để biểu diễn dữ liệu và sự liên kết giữa hai thực thể Khoa và Sinhvien, đó là liên kết một nhiều với một đầu là Khoa một đầu là sinhvien, khóa của đầu một là (Makhoa) được đặt vào đầu nhiều để làm khóa kết nối.

2.1.2. Mô hình dữ liệu mạng

Đây là mô hình dựa trên khái niệm lý thuyết tập hợp của các quan hệ nó là sự phát triển của mô hình dữ liệu quan hệ với các nguyên tắc sau:

- Dữ liệu được biểu diễn dưới dạng một đồ thị có hướng phức tạp, mô hình chỉ biểu diễn các liên kết một - nhiều, các liên kết một - một và nhiều - nhiều khi đưa vào mô hình được quy đổi như sau: Liên kết một - một được coi là dạng đặc biệt của liên kết một - nhiều được tổ hợp lại để tạo thành các liên kết một nhiều, với liên kết nhiều - nhiều thì được thêm vào một thực thể trung gian để phát triển thành các liên kết một - nhiều.

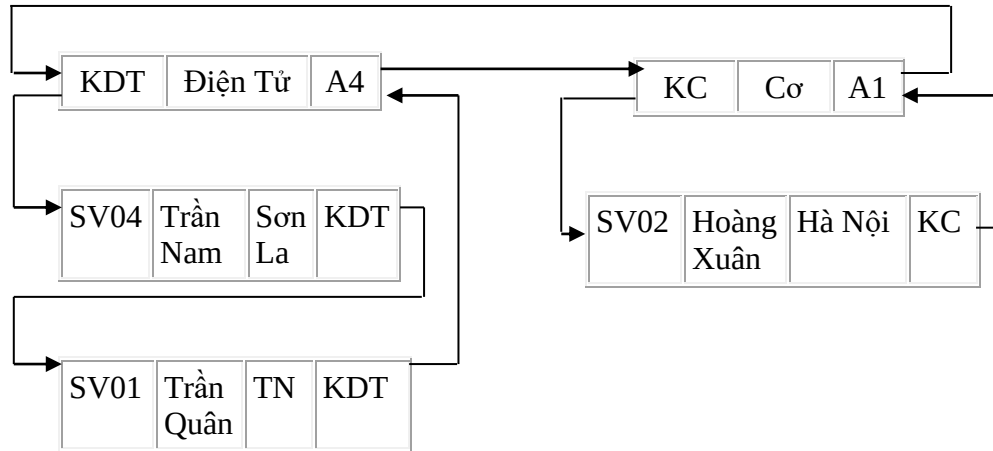


- Thực thể được biến đổi thành các kiểu bản ghi.

- Thuộc tính biến đổi thành các trường của bản ghi.

Một liên kết một nhiều là sự phân cấp mà một tập được biến đổi sang kiểu bản ghi chủ ở đầu một và kiểu bản ghi thành viên ở đầu nhiều.

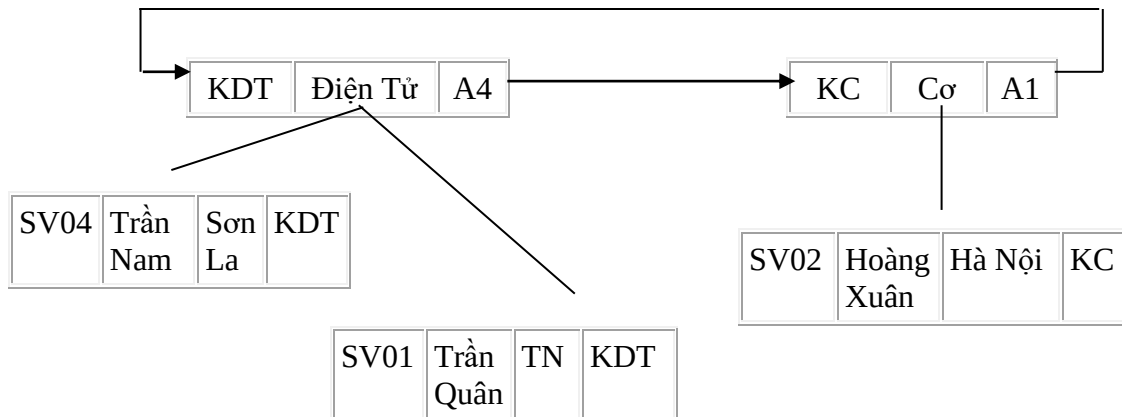
Ví dụ: Mô tả mô hình dữ liệu mạng

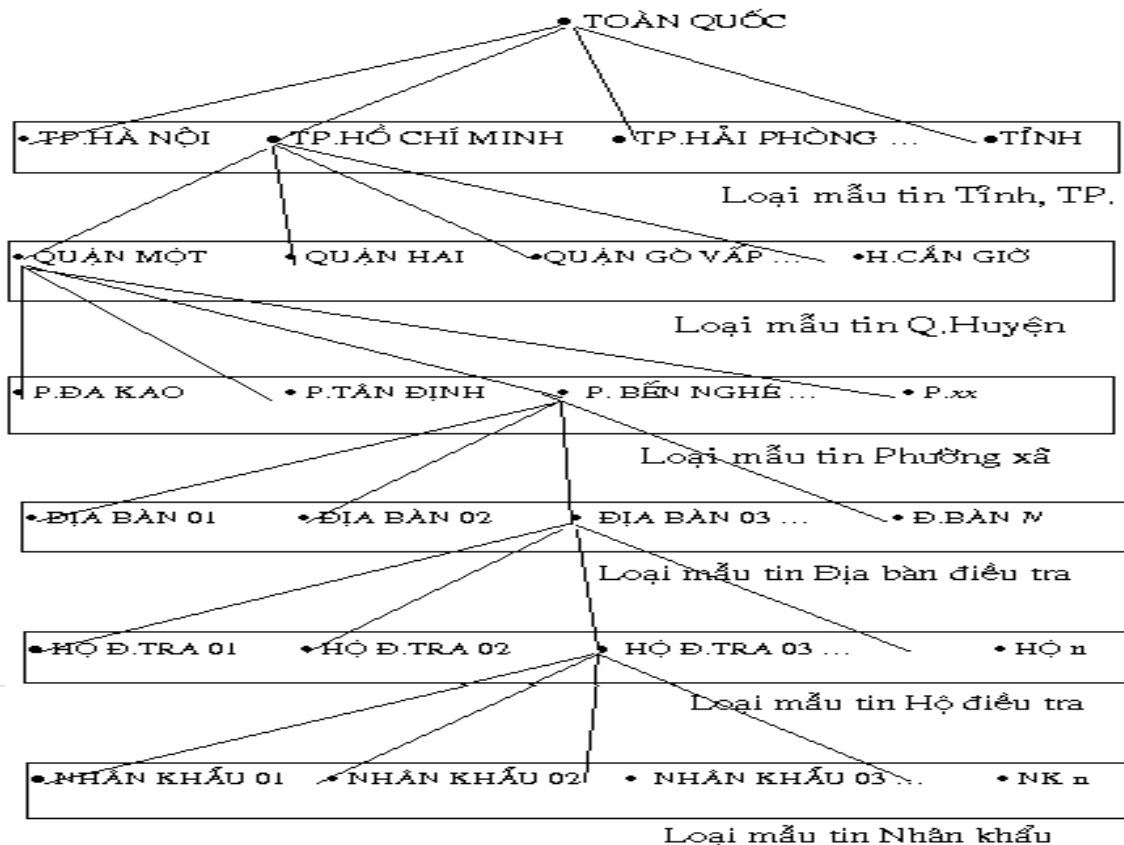


2.1.3. Mô hình dữ liệu phân cấp

Đây là dạng đặc biệt của mô hình dữ liệu mạng với hạn chế là mỗi kiểu bản ghi chỉ là bản ghi thành viên của nhiều nhất một tập và mô hình được bố trí theo dạng cây (thư mục) và giữa các cấp cha con trên dưới đều có những liên hệ nhất định.

Ví dụ: Mô tả cây phân cấp của mô hình dữ liệu đối với CSDL điều tra sinh viên trong các khoa.





2.2. SƠ ĐỒ THỰC THỂ - LIÊN KẾT

Là kết quả của quá trình phân tích thiết kế hệ thống. Sơ đồ thực thể liên kết gồm 3 thành phần:

2.2.1. Thực thể

Thực thể (Entity) là những đối tượng dữ liệu cơ bản chứa nội dung các thông tin cần thu thập. Thực thể có thể biểu thị cho người, nơi chốn, sự vật hoặc các biến cố có thông tin đáng chú ý. Một xuất hiện cụ thể của một thực thể được gọi là thể hiện thực thể (Entity Instance).

Ví dụ nhân viên, phòng ban, kỹ năng, vị trí, thủ tục, chương trình...là các thực thể.

Tên thực thể được viết bên trong hình chữ nhật.

Ví dụ: CSDL thư viện có các thực thể là: Sách, Độc giả.

Tên của thực thể bao giờ cũng là 1 danh từ.

Cách biểu diễn:

Sách

Độc giả

2.2.2. Thuộc tính

Thuộc tính (Attribute) là các tính chất đặc trưng của thực thể, chỉ ra các chi tiết cần mô tả về thực thể. Mỗi thực thể được xác định bởi một tập các thuộc tính. Một xuất hiện cụ thể của một thuộc tính trong một thực thể hoặc một mối quan hệ được gọi là giá trị thuộc tính (Attribute Value).

Ví dụ: Đối với các thực thể trong CSDL thư viện:

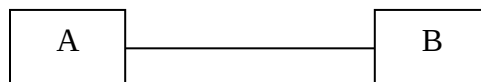


2.2.3. Liên kết thực thể

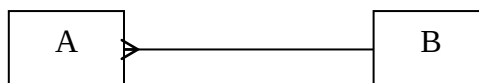
Trong sơ đồ thực thể liên kết có đường nối giữa 2 thực thể

Cách biểu diễn liên kết:

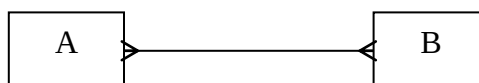
Quan hệ một → một (One to One): Một thực thể trong A được kết hợp với tối đa một thực thể trong B, và một thực thể trong B được kết hợp với tối đa một thực thể trong A.



Quan hệ một → nhiều (One to Many): Một thực thể trong A được kết hợp với nhiều thực thể trong B, và một thực thể trong B có thể được kết hợp với tối đa với một thực thể trong A.

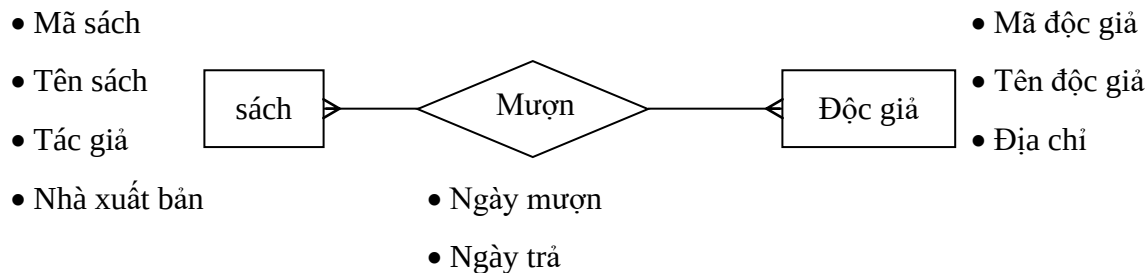


Quan hệ nhiều → nhiều (Many to Many): Nhiều thực thể trong A được kết hợp với nhiều thực thể trong B, và nhiều thực thể trong B được kết hợp với nhiều thực thể trong A.



Trong lược đồ quan hệ, cấu trúc dữ liệu quan hệ nhiều - nhiều sẽ được thực thể hoá dưới dạng một - nhiều.

Ví dụ :



Trong cơ sở dữ liệu, dữ liệu tác nghiệp bao gồm thực thể và liên kết thực thể. Cơ sở dữ liệu phải có khả năng biểu diễn hai đối tượng này. Căn cứ vào cách biểu diễn hai đối tượng đó người ta chia thành các mô hình dữ liệu khác nhau.

CHƯƠNG 3 MÔ HÌNH DỮ LIỆU QUAN HỆ

(Số tiết lý thuyết 5 tiết)

3.1. CÁC KHÁI NIỆM CƠ BẢN

3.1.1. Thực thể

a. Thực thể

🔹 Định nghĩa: Thực thể của thế giới thực hoặc đối tượng cụ thể mà ta có thể lưu trữ thông tin thao tác và dễ dàng kiểm tra sự tồn tại của nó hoặc một đối tượng trừu tượng tức là người ta không thể không thể thao tác được trên đó như thừa nhận trên đó sự tồn tại đúng đắn ngoài sự vật chất hoá.

🔹 Các đặc trưng của thực thể kết hợp: Các đặc trưng của thực thể sử dụng để phân biệt, để xác định giá trị của thực thể đó là những yếu tố cho ta hiểu biết về thực thể.

Ví dụ: Một học sinh gồm có các đặc trưng: Masv, Ngaysinh, Gioitinh, hoten.

Các đặc trưng của liên kết là các yếu tố để có thể hiểu rõ về một sự kết hợp.

→ Lớp K4th có 75 học sinh: có 60 là đặc trưng liên kết giữa hai thực thể lớp K4 và học sinh.

Tuy nhiên không phải tất cả các đặc trưng của các thực thể hoặc các liên kết đều đem lại lợi ích cho và được lưu trữ trong cơ sở dữ liệu vì vậy cần phải có sự sàng lọc trên các đặc trưng chỉ lưu trữ lại trong cơ sở dữ liệu, những đặc trưng có ích.

🔹 Thực thể thiết kế: Một thực thể thiết kế là một hình ảnh trong hệ thống thông tin của một thực thể của thế giới thực.

Một thực thể thiết kế là một phần tử của không gian CSDL.

Một thực thể của thế giới thực tồn tại một cách độc lập với lợi ích mà người ta mang đến cho nó.

Kiểu thực thể là tập hợp các thực thể thiết kế có cùng các đặc trưng chung hoặc tựa tựa như nhau.

Ví dụ: Trong thư viện có những quyển sách khác nhau → người ta tập hợp những quyển sách tựa tựa nhau thành kiểu thực thể.

Tính chất của kiểu thực thể là các đặc trưng giống nhau của thực thể được giữ lại trong mô hình và chúng được gọi là thuộc tính.

3.1.2. Thuộc tính (Attribute)

🔹 Định nghĩa: Một thuộc tính biểu thị một lô thông tin nhỏ nhất có thể sử dụng một cách tự do và có ý nghĩa độc lập với các lô khác. Thuộc tính diễn tả bản chất của thực thể và nó đại diện cho một phần tử cơ sở của thực thể.

Một thuộc tính được biểu diễn bằng một cặp giá trị:

Thuộc tính = {tên thuộc tính, miền giá trị}

Ví dụ: Đối tượng KHOA (tương ứng với loại thực thể KHOA trong mô hình thực thể kết hợp) có các thuộc tính Mã-khoa, Tên-khoa, Địa-diện.
Loại thực thể SINH-VIÊN có một số thuộc tính Mã-khoa, Mã-sinh-viên, Tên-sinh-viên, Địa-chi.

Chú ý:

- Tên thuộc tính là duy nhất cho mỗi thuộc tính và là một danh từ.
- Miền giá trị là một tập các giá trị mà có thể rút ra các giá trị thực của thuộc tính và ký hiệu DOM (DOMAIN).

Miền giá trị có thể là chữ số, ngày, ký tự, logic.

Ví dụ: Gọi $D_1, D_2, \dots, D_n$ là n miền. Trong đó $D_1, D_2, \dots, D_n$ có thể được biểu diễn dưới dạng chuỗi ký tự, chuỗi số hay chuỗi chữ và số.

Hay: Giới tính có hai miền giá trị NAM & NU.

Số trong lớp có miền giá trị $\{10 \div 150\}$

 Các loại thuộc tính (Gồm 2 loại thuộc tính)

- Thuộc tính tính toán: Khi một giá trị của thuộc tính là giá trị nhận được từ các giá trị của các thuộc tính khác.

Ví dụ: Lương là thuộc tính tính toán = lương cơ bản * hệ số lương.

- Thuộc tính ổn định: Là thuộc tính đặc trưng của sự kiện không thể hiện lại và một khi đã gán cho nó một giá trị, giá trị đó là không thể thay đổi.

Ví dụ: Một người chỉ có duy nhất một số chứng minh thư nhân dân.

3.1.3. Quan hệ (Relation)

a. Định nghĩa:

Quan hệ là một tập con các tích Đề-Các của một hay nhiều miền, vì vậy mỗi quan hệ có thể là hữu hạn hoặc vô hạn.

Miền là tập các giá trị có thể là số nguyên, ký tự, xâu ký tự...nếu gọi $D_1, D_2, \dots, D_n$ là các miền khác nhau thì khi tích Đề-các của $D_1 \times D_2 \times \dots \times D_n$ là tập của n bộ $(v_1, v_2, \dots, v_n)$ sao cho $v_i \in D_i; i = 1 \div n$.

Ví dụ: Có hai miền $D_1(0,1), D_2(a,b,c)$ khi đó:

$D_1 \times D_2 = \{(0,a), (0,b), (0,c), (1,a), (1,b), (1,c)\}$

Với $(0,a), (0,b), (0,c), (1,a), \dots$ là các quan hệ.

 Trong cơ sở dữ liệu: Một quan hệ được biểu diễn dưới dạng một bảng, mỗi thuộc tính biểu diễn bằng một cột, mỗi giá trị của thuộc tính được ghi trong một cột tương ứng. Một bộ giá trị quan hệ được biểu diễn bằng một dòng.

 Định nghĩa hình thức: Gọi $R = \{A_1, A_2, \dots, A_n\}$ là tập hữu hạn các thuộc tính $A_i (i = 1, \dots, n)$ có miền giá trị tương ứng là $dom(A_i)$.

Quan hệ trên các tập thuộc tính $R = (A_1, \dots, A_n)$ là tập con các tích Đề-các của các miền.

$r \subseteq Dom(A_1) \times \dots \times Dom(A_n)$

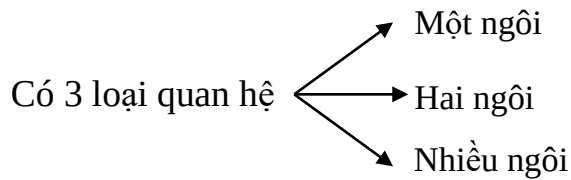
Quan hệ có thể viết lại: $r(R)$ hoặc $r(A_1, \dots, A_n)$.

Ví dụ: Ta có quan hệ Sinhvien (Masv, Tensv, Namsinh, Quequan).
Quan hệ này được biểu diễn như sau:

<u>Masv</u>	Tensv	Ngaysinh	Quequan
SV01	A	1/01/1988	TN
SV02			

bộ

b. Các kiểu quan hệ:



- Quan hệ một ngôi: là quan hệ chỉ chứa một thuộc tính.
Ví dụ: $R = \{Masv\}$
 $R = \{SV01\}$
- Quan hệ hai ngôi: là quan hệ chứa hai thuộc tính.
Ví dụ: $R = \{Masv, Tensv\}$
 $R = \{SV01, A\}$
- Quan hệ nhiều ngôi: là quan hệ chứa nhiều hơn hai thuộc tính.
Ví dụ: $R = \{Masv, Tensv, ngaysinh, quequan\}$
 $R = \{SV01, A, 1988, TN\}$

3.1.4. Khoá

Khoá (Key) của lược đồ quan hệ R định nghĩa: Cho quan hệ r xác định trên tập thuộc tính R, ta có $k \in R$ là tập con, k là khoá của r nếu với $\forall t_1, t_2 \in r$ trong đó $t_1 \neq t_2$ thì sẽ $\exists A \in k$ sao cho $t_1(A) \neq t_2(A)$ và đương nhiên dẫn theo $t_1(k) \neq t_2(k)$, có nghĩa là cứ với hai bộ bất kỳ khác nhau thì giá trị trên k của hai bộ phải khác nhau hay giá trị trên k của một bộ là phân biệt với tất cả các giá trị trên k của tất cả các bộ khác (k là giá trị để xác định duy nhất một bộ) khi đó k được gọi là khoá của quan hệ.

Để có thể định nghĩa khoá một cách tốt nhất ta lưu ý: Nếu k là khoá của quan hệ r thì với mọi k' chứa k ($\forall k' \subseteq k$), k' cũng là khoá của r.

🔄 **Khóa tối thiểu:** k là khoá của r và với mọi k' nhỏ hơn k ($k' < k$) hay k' nằm trong k thì k' không là khoá của r khi đó nói k là khóa tối thiểu của r.

🔄 **Khóa chính:** Một quan hệ có thể có rất nhiều khóa tối thiểu, người ta sẽ chọn một trong các khóa tối thiểu làm khóa chính, số còn lại là khóa dự bị.

🔄 **Khóa ngoài:** Cho một quan hệ r xác định trên tập thuộc tính R , có k là khóa ngoài của r nếu như k không phải là khóa chính của quan hệ r nhưng lại là khóa chính của quan hệ s (khóa ngoài còn được gọi là khóa liên kết hay khóa kết nối).

Ví dụ: LOP (Malop, Tenlop, Hoitruong)

SINHVIEN (Masv, Malop, Hoten, Ngaysinh)

(Hai quan hệ trên có quan hệ với nhau bởi Malop hoặc Malop là khóa ngoài hay khóa kết nối).

3.2.CÁC PHÉP TOÁN CỦA ĐẠI SỐ QUAN HỆ

3.2.1. Quan hệ khả hợp

Cho r và s là hai quan hệ, chúng ta nói rằng r và s là khả hợp nếu chúng được xác định trên cùng miền giá trị.

Giả sử miền giá trị của r là: $D_1x_1...x_nD_n$ và miền giá trị của s là: $D'_1x'_1...D'_m$ thì r và s là khả hợp nếu $D_1x_1...x_nD_n = D'_1x'_1...D'_m$, có nghĩa là $n = m$ và $D_i = D'_i$ (với $i = 1...n$).

Khi các cột của bảng biểu diễn quan hệ được đặt tên, chúng có thể định nghĩa tính khả hợp đối với hai quan hệ như sau:

🔄 **Định nghĩa:** Cho hai quan hệ r và s , r và s được gọi là khả hợp nếu chúng được xác định trên cùng tập thuộc tính và các thuộc tính có cùng tên thì có cùng miền giá trị.

3.2.2. Các phép toán

a. Phép hợp (Union)

Phép hợp của hai quan hệ khả hợp r và s (k/h: $r \cup s$), là tập tất cả các bộ thuộc r hoặc thuộc s hoặc thuộc cả hai quan hệ.

Biểu diễn hình thức của phép hợp có dạng:

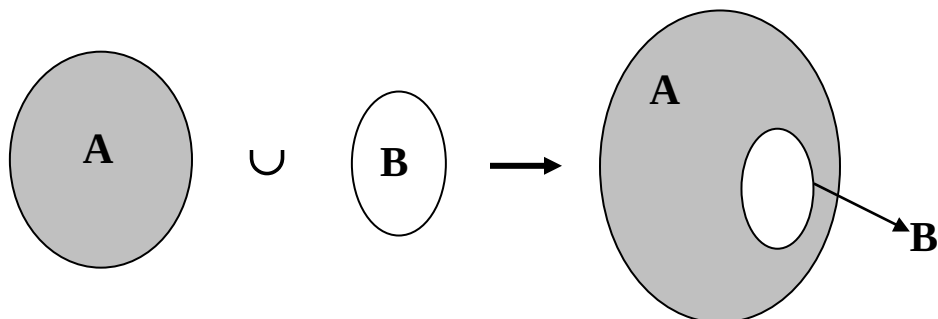
$$r \cup s = \{t : t \in r \text{ hoặc } t \in s \text{ hoặc } t \in r \& s\}$$

Ví dụ 1: Cho hai quan hệ r và s là khả hợp, chúng ta tính r hợp s .

r			s			$r \cup s$		
MASV	MAMH	DIEM	MASV	MAMH	DIEM	SV01	CSDL	5.0
SV01	CSDL	5.0	SV02	CSDL	2.0	SV02	CTDL	2.0
SV02	CTDL	2.0	SV04	CTDL	9.0	SV03	MANG	8.0
SV03	MANG	8.0	SV01	MANG	8.0	SV04	CNPM	7.0
SV04	CNPM	7.0				SV04	CTDL	9.0
						SV01	MANG	8.0

Ví dụ 2 : Cho hai quan hệ r và s là khả hợp, chúng ta tính r hợp s

r(A B C)	s(A B C)	$r \cup s = (A B C)$
a1 b1 c1	a1 b1 c1	a1 b1 c1
a2 b1 c1	a2 b2 c1	a2 b1 c1
a2 b2 c2		a2 b2 c2
		a2 b2 c1



b. Phép giao (Intersection)

Phép giao của hai quan hệ khả hợp r và s ($k/h: r \cap s$) là tập tất cả các bộ thuộc thuộc cả hai quan hệ r và s.

Biểu diễn hình thức của phép giao có dạng:

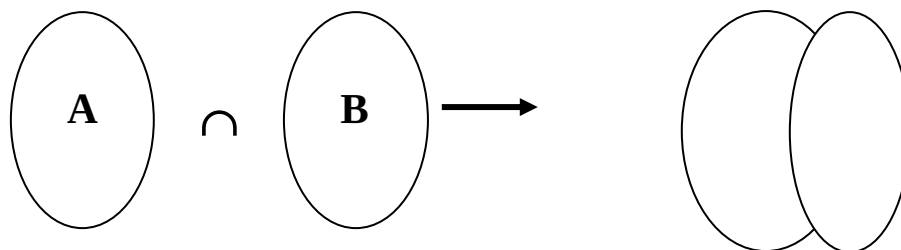
$$r \cap s = \{t : t \in r \ \& \ s\}$$

Ví dụ 1 : Cho hai quan hệ r và s là khả hợp, chúng ta tính r giao s.

r			s			r ∩ s		
MASV	MAMH	DIEM	MASV	MAMH	DIEM	MASV	MAMH	DIEM
SV01	CSDL	5.0	SV02	CSDL	2.0	SV02	CSDL	2.0
SV02	CTDL	2.0	SV04	CTDL	9.0			
SV03	MANG	8.0	SV01	MANG	8.0			
SV04	CNPM	7.0						

Ví dụ 2 : Cho hai quan hệ r và s là khả hợp, chúng ta tính r giao s.

R(A B C)	s(A B C)	$r \cap s = (A B C)$
a1 b1 c1	a1 b1 c1	a1 b1 c1
a2 b1 c1	a2 b2 c1	
a2 b2 c2		



c. Phép trừ (minus)

Phép trừ của hai quan hệ khả hợp r và s ($k/h: r - s$) là tập tất cả các bộ thuộc quan hệ r nhưng không thuộc quan hệ s .

$$r-s = \{t: t \in r \wedge t \notin s\}$$

Ví dụ 1 : Cho hai quan hệ r và s là khả hợp, chúng ta tính $r - s$

r			s			r - s		
MASV	MAMH	DIEM	MASV	MAMH	DIEM	MASV	MAMH	DIEM
SV01	CSDL	5.0	SV02	CTDL	2.0	SV01	CSDL	5.0
SV02	CTDL	2.0	SV04	CTDL	9.0	SV03	MANG	8.0
SV03	MANG	8.0	SV01	MANG	8.0	SV04	CNPM	7.0
SV04	CNPM	7.0						

Ví dụ 2 : Cho hai quan hệ r và s là khả hợp, chúng ta tính $r - s$

R(A B C)	s(A B C)	r-s=(A B C)
a1 b1 c1	a1 b1 c1	a2 b1 c1
a2 b1 c1	a2 b2 c1	a2 b2 c2
a2 b2 c2		

d. Phép tích Đề - các (Cartesian product)

Gọi r là quan hệ xác định trên tập thuộc tính $\{A_1, A_2, \dots, A_n\}$ và s là quan hệ xác định trên tập thuộc tính $\{B_1, B_2, \dots, B_m\}$. Tích đề- các của hai quan hệ r và s , ký hiệu là $r \times s$, là tập tất cả các $(n + m)$ bộ, với n phần đầu thuộc quan hệ r và m phần sau thuộc quan hệ s .

$$r = \{A_1, A_2, \dots, A_n\}$$

$$s = \{B_1, B_2, \dots, B_m\}$$

$$r \times s = n + m$$

Ví dụ 1 : Cho hai quan hệ r và s , chúng ta tính $r \times s$

r 1

MASV	MAMH	DIEM
SV01	CSDL	5.0
SV02	CTDL	2.0
SV03	MANG	8.0

r2

MAMH	TENMH
CSDL	Cơ sở dữ liệu
CTDL	Cấu trúc dữ liệu

r1*r2

MASV	MAMH	DIEM	MAMH	TENMH
SV01	CSDL	5.0	CSDL	Cơ sở dữ liệu
SV01	CSDL	5.0	CTDL	Cấu trúc dữ liệu
SV02	CTDL	2.0	CSDL	Cơ sở dữ liệu
SV02	CTDL	2.0	CTDL	Cấu trúc dữ liệu
SV03	MANG	8.0	CSDL	Cơ sở dữ liệu
SV03	MANG	8.0	CTDL	Cấu trúc dữ liệu

Ví dụ 2 : Cho hai quan hệ r và s, chúng ta tính r x s

r	(A	B	C)
	a ₁	b ₁	c ₁
	a ₂	b ₂	c ₂

s	(D	E	F)
	d	e	f
	d'	e'	f

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 ₂	b ₂	c ₂	d	e	f
	a ₂	b ₂	c ₂	d'	e'	f

3.2.2. Các phép toán quan hệ

a. Phép chiếu

Phép chiếu trên một quan hệ thực chất là phép toán loại bỏ đi một số thuộc tính chỉ giữ lại một số thuộc tính còn lại của quan hệ.

Định nghĩa:

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

Biểu diễn hình thức của phép toán này là:

$$\Pi_X(r) = \{t[X] / t \in r\}$$

Chú ý: Phép chiếu chính là phép trích rút dữ liệu theo cột (Chiều dọc)

Ví dụ 1 : $R = \{A, B, C, D\}$; $X = \{A, B\}$; $Y = \{A, C\}$

r	(A	B	C	D)
	a ₁	b ₁	c ₁	d ₁
	a ₁	b ₁	c ₁	d ₂
	a ₂	b ₂	c ₂	d ₂
	a ₂	b ₂	c ₃	d ₃

$\Pi_X(r) =$	(A	B)
	a ₁	b ₁
	a ₂	b ₂

$\Pi_Y(r) =$	(A	C)
	a ₁	c ₁
	a ₂	c ₂
	a ₂	c ₃

Ví dụ 2:

r			
	MASV	MAMH	DIEMTHI
	SV01	CSDL	5.0
	SV02	CTDL	2.0
	SV03	MANG	8.0

$\Pi_X(r) = \{MaMH\}$

MAMH
CSDL
CTDL
MANG

b. Phép chọn

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

Định nghĩa:

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

Hình thức hóa phép chọn trên quan hệ r được định nghĩa như sau:

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

Chú ý: $F(t)$ ở đây phải được hiểu là giá trị của thuộc tính xuất hiện trong biểu thức F của bộ t thỏa mãn điều kiện của F .

Phép chọn là phép trích rút dữ liệu theo chiều ngang.

Các phép toán so sánh trong biểu thức F là $<$, $=$, $>$, $>=$, $<=$ và $\neq$;

Các phép toán logic là $\wedge$ (và), $\vee$ (hoặc) và $\neg$ (không).
 Ví dụ 1: $R = \{A, B, C, D\}$;

r	(A	B	C	D)
	a ₁	b ₁	c ₁	d ₁
	a ₁	b ₁	c ₁	d ₂
	a ₂	b ₂	c ₂	d ₂
	a ₂	b ₂	c ₃	d ₃

Bước chọn ký hiệu là σ_F : Nếu $\sigma_{A=a_1}(r)$ thì ta có

r	(A	B	C	D)
	a ₁	b ₁	c ₁	d ₁
	a ₁	b ₁	c ₁	d ₂

Nếu $\sigma_{A = a_1 \wedge D = d_2}(r)$ thì:

r	(A	B	C	D)
	a ₁	b ₁	c ₁	d ₂

Ví dụ 2 :

r

$\sigma_{\text{Điểm thi} \geq 5}(r)$

MASV	MAMH	DIEMTHI
SV01	CSDL	5.0
SV02	CTDL	2.0
SV03	MANG	8.0

MASV	MAMH	DIEMTHI
SV01	CSDL	5.0
SV03	MANG	8.0

c. Phép kết nối:

Cho 2 lược đồ quan hệ r và s , quan hệ r xác định trên tập thuộc tính $\{A_1, \dots, A_n\}$ và quan hệ s xác định trên tập thuộc tính $\{B_1, \dots, B_m\}$. A_i và B_j lần lượt là các thuộc tính của r và s sao cho $MGT(A_i) = MGT(B_j)$, θ là phép so sánh trên MGT . Ta có biểu thức so sánh: $A \theta B$

Phép kết nối 2 quan hệ thực chất là phép ghép các cặp bộ của 2 quan hệ thỏa mãn một điều kiện nào đó trên chúng. Điều kiện đó là điều kiện kết nối hay biểu thức kết nối.

Phép kết nối của quan hệ r với quan hệ s với biểu thức kết nối F được định nghĩa như sau:

$$r \bowtie s = \{ (t \cap u) \mid t \in r; u \in s \text{ và } t[A] \theta u[B] \}.$$

$$A \theta B$$

Trong đó $t \cap u$ là 2 bộ thuộc 2 quan hệ và được xếp cạnh nhau trong trường hợp phép so sánh θ là phép so sánh $=$ thì phép kết nối được gọi là kết nối $=$. Nếu trong 2 quan hệ có 2 thuộc tính cùng tên và chúng được kết nối với nhau bằng phép kết nối $=$ thì một trong 2 thuộc tính sẽ bị loại bỏ và phép kết nối được gọi là kết nối tự nhiên, phép kết nối tự nhiên ký hiệu “ $*$ ”. Phép kết nối tự nhiên là phép kết nối được sử dụng nhiều nhất trong thực tế.

Kết nối tự nhiên của 2 quan hệ có thể được định nghĩa như sau:

$$r(ABC) * S(CDE) = \{ t[ABCDE] \mid t[ABC] \in r \text{ và } t[CDE] \in s \}.$$

Ví dụ 1:

r	(A	B	C)
	a ₁	1	1
	a ₂	2	1
	a ₂	2	2

s	(C	D	E)
	1	d ₁	e ₁
	2	d ₂	e ₂
	3	d ₃	e ₃

$r \bowtie s =$ $B \geq C$	(A	B	C	C	D	E)
	a ₁	1	1	1	d ₁	e ₁
	a ₂	2	1	1	d ₁	e ₁
	a ₂	2	1	2	d ₂	e ₂
	a ₁	2	2	1	d ₁	e ₁
	a ₁	2	2	2	d ₂	e ₂

Kết nối tự nhiên:

$r(BAC) * S(CDE) =$	(A	B	C	D	E)
	a_1	1	1	d_1	e_1
	a_2	2	1	d_1	e_1
	a_1	2	2	d_2	e_2

Ví dụ 2: Với $A_i = B_j = MAMH$

r1

MASV	MAMH	DIEMTHI
SV01	CSDL	5.0
SV02	CTDL	2.0
SV03	MANG	8.0

r2

MAMH	TENMH
CSDL	Cơ sở dữ liệu
CTDL	Cấu trúc dữ liệu

$r3 = r1 \bowtie_{MAMH} r2$

MASV	MAMH	DIEMTHI	TENMH
SV01	CSDL	5.0	Cơ sở dữ liệu
SV02	CTDL	2.0	Cấu trúc dữ liệu

d. Phép chia:

Định nghĩa: Cho r là một quan hệ r ngôi xác định trên tập thuộc tính U và s là một quan hệ m ngôi xác định trên tập thuộc tính V với $n > m$ và $s \neq \emptyset$.

Phép chia quan hệ r cho quan hệ s , ký hiệu là c , là tập tất cả các bộ t sao cho với mọi bộ $v \in s$ thì t ghép với v thuộc r .

Hình thức hóa phép chia được định nghĩa như sau:

$$r \div s = \{t \mid \forall v \in s \Rightarrow (t, v) \in r\}$$

Ví dụ1:

r	(A	B	C	D)
	a	b	c	d
	a	b	e	f
	b	c	e	f
	e	d	c	d
	e	d	e	f
	a	b	d	e

S	(C	D)
	c	d
	e	f

r ÷ s =	(A	B)
	a	b
	e	d

3.2.3. Một số ví dụ về tìm kiếm bằng đại số quan hệ

Ví dụ: Cả 3 quan hệ:

S (S#, SNAME, STATUS, CITY): C, c h·ng cung øng,

P (P#, PNAME, COLOR, WSEIGHT, CITY): C, c mÆt h·ng,

SP (S#, P#, QTY): C, c mÆt h·ng ®· cung cÆp.

- T×m sè hiÖu cña nh÷ng h·ng ®· cung øng mÆt h·ng P2:

$$\Pi_{S\#}(\sigma_{P\# = 'P2'}(SP))$$

- T×m sè hiÖu cña nh÷ng h·ng cung øng Ýt nhÊt l· mét mÆt h·ng m·u

®á:

$$\Pi_{S\#}(\sigma_{COLOR = 'RED'}(P * SP))$$

hoÆc

$$\Pi_{S\#}((\sigma_{COLOR = 'RED'}(P)) * SP)$$

CHƯƠNG 4

NGÔN NGỮ SQL VÀ CÁC HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

(18 tiết lý thuyết)

4.1. TỔNG QUAN VỀ NGÔN NGỮ SQL

SQL (viết tắt của *Structured Query Language*) là hệ thống ngôn ngữ đặc biệt để mô tả dữ liệu và thao tác dữ liệu trong cơ sở dữ liệu. Thông qua SQL ta có thể thực hiện các thao tác trên cơ sở dữ liệu như: Định nghĩa dữ liệu, thao tác dữ liệu, điều khiển truy cập dữ liệu và quản lý vận hành dữ liệu... SQL là thành phần quan trọng và không thể thiếu trong hệ quản trị CSDL quan hệ.

SQL là ngôn ngữ truy vấn dựa trên các khái niệm về quan hệ, các lệnh SQL dùng để trích xuất dữ liệu từ một hay nhiều quan hệ, kết quả của một lệnh SQL là một quan hệ.

SQL ra đời nhằm mô tả dữ liệu cho các cơ sở dữ liệu theo mô hình quan hệ, trong một cơ sở dữ liệu sẽ có nhiều quan hệ, dữ liệu đặc trưng và lưu trữ trong các bảng. Mỗi một bảng là một tập hợp bao gồm các dòng và cột; mỗi một dòng là một bản ghi và mỗi một cột tương ứng với một trường. Các bảng trong cơ sở dữ liệu sẽ có mối quan hệ với nhau, các mối quan hệ đặc biệt được thể hiện thông qua khóa chính và khóa ngoại của bảng, khóa chính của bảng là một hoặc nhiều cột của nó, trên duy nhất trong bảng và do đó giá trị, trên của nó là x, c Định duy nhất một dòng dữ liệu trong bảng. Khóa ngoại là một tập hợp một hoặc nhiều cột của nó, trên đặc biệt tổ hợp khóa chính của các bảng khác.

Như vậy, các thao tác của SQL là ngôn ngữ hoàn thiện các mô tả dữ liệu trong cơ sở dữ liệu và là một phần không thể thiếu trong các hệ quản trị CSDL. Bản thân SQL không phải là một hệ quản trị, mà là một công cụ để xây dựng và vận hành cơ sở dữ liệu. Vai trò của ngôn ngữ là giao tiếp giữa người sử dụng và hệ quản trị CSDL.

4.1.1. Bảng ký tự và tập lệnh của SQL

♦ Bảng ký tự

Ngôn ngữ SQL đã bảng ký tự riêng như: các ngôn ngữ khác và nên nhớ x@y dùng trên các ký tự của bảng bao gồm:

- Các chữ cái la tinh: A, B, C,..., Z, a, b, c,..., z
- Các chữ số: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Các ký hiệu đặc biệt: +, -, *, /, <, >, {, }, [,], %, &, #...
- Ký tự gạch dưới '_' và ký tự khoảng trắng ' ' (space)

♦ Các tập lệnh của SQL

Các tập lệnh của SQL nên chia thành 3 nhóm:

- Nhóm ngôn ngữ dữ liệu:

- + Tạo ra cấu trúc bảng.
- + Sửa đổi cấu trúc bảng.
- + Tạo CSDL.
- + Xo, bảng.

- Quản lý dữ liệu:

- + Xem dữ liệu.
- + Sửa đổi dữ liệu.
- + Chèn dữ liệu.
- + Xo, dữ liệu.

- Tìm kiếm dữ liệu: Tìm kiếm và trả về thông tin cần thiết về dữ liệu theo yêu cầu của người dùng.

4.1.2. Cách viết lệnh trong SQL

SQL chuẩn gồm khoảng 40 câu lệnh, các câu lệnh của SQL nên nhớ nên chú ý đến các từ lệnh, lưu ý một số khác cho biết các chức năng của câu lệnh như: (SELECT, DELETE, INSERT,...) sau từ lệnh lưu ý các mệnh đề của câu

KQ: Kĩt qu¶ thùc tĒp theo    t i    ch n

Trong vÝ dō trªn mçi mét dñng trong b¶ng SINHVIEN t–ng øng vói mét sinh viªn vµ mçi mét cét biÓu diÔn cho mét tÝnh chÊt cña thùc thÓ (ch¼ng h¹n cét NS cña sinh viªn biÓu diÔn cho ngµy sinh cña sinh viªn ®-íc l-u tr÷ trong b¶ng).

Minh hãa cho ta thÊy ®-íc 3 b¶ng trong CSDL:

MSSV	HODEM	TEN	NS	QUE	HL
MS001	Phuong	Anh	1983	Thai Nguyen	7,2
MS002	Nguyen Thi	Hanh	1983	Thai Nguyen	7,0
MS003	Tran	Nam	1984	Thai Nguyen	6.9
MS004	Tran	Quan	1983	Hue	7.4
MS005	Khanh	Chi	1983	Ha Noi	6.5
MS006	Nguyen Thi	Sinh	1984	Ha Nam	7.6
MS007	Le Minh	Tan	1985	Thai Nguyen	7.3
MS008	Cam Duc	Nam	1985	Son La	5.9
MS009	Le Son	thao	1980	Ha Tay	5.5

MSDT	TDT	CN	KP
DT001	Quan ly kinh doanh	Nguyen Tuan Dung	2
DT002	Quan ly tai chinh	Nguyen Thi Bien	1.5
DT003	Quan ly nhan su	Vu Thi Thuy Thao	3
DT004	Quan ly tien von	Nguyen Chi Hieu	1.8
DT005	Quan ly dien	Trinh Thi Ha	1
DT006	Quan ly nuoc	Do Duy Cop	2.5
DT007	Quan ly thu vien	Nguyen Tuan Dung	2
DT008	Quan ly sach	Nguyen Thi Bien	1.5
DT009	Quan tri mang	Vu Thi Thuy Thao	3
DT010	Thiet ke mang Lan	Nguyen Chi Hieu	1.8
DT011	Quan ly thue bao DD	Trinh Thi Ha	1
DT012	Quan ly KD may tinh	Do Duy Cop	2.5

MSSV	MSDT	NTT	NGAYTT	KM	KQ
MS001	DT001	Thai Nguyen	17/03/2006	5	9.5
MS002	DT002	Thai Nguyen	22/3/2006	7	10
MS003	DT003	Lang Son	17/3/2006	150	7.5

MS004	DT004	Hue	22/3/2006	300	8
MS005	DT005	Ha Noi	25/3/2006	80	8.5
MS006	DT006	TuyenQuang	25/3/2006	80	10
MS007	DT007	Yen Bai	25/3/2006	170	7.5
MS008	DT008	Lao Cai	17/3/2002	190	8
MS009	DT009	Ha Noi	17/3/2006	80	8.5

4.2. Ng«n ng÷ ®Þnh nghÜa d÷ liÖu

Ng«n ng÷ ®Þnh nghÜa d÷ liÖu bao g¸m c,c c©u lÖnh cho phÐp m« t¶ c,c ®òi t-ìng.

4.2.1. T¹o c¬ s¸ d÷ liÖu

T¹o c¬ s¸ d÷ liÖu trªn c,c b¶ng lù viÖc tæ chøc vù l-u tr÷. Mçi mét b¶ng lù mét t¸p hùp bao g¸m c,c dßng vù c,c c¸t; mçi mét dßng lù mét b¶n ghi vù mçi mét c¸t t-ìng øng vù mét tr-êng, t¸p c,c tªn c¸t cùng vù kiÓu d÷ liÖu vù c,c tÝnh ch¸t kh,c t¹o nªn c¸u tróc b¶ng, t¸p c,c dßng trong b¶ng chÝnh lù d÷ liÖu c¸a b¶ng. C,c b¶ng trong mét CSDL c¸ mòi quan hÖ vù nhau, c,c mòi quan hÖ ®-ìc biÓu diÔn th«ng qua kh¸a chÝnh vù kh¸a ngoù c¸a c,c b¶ng.

Ta c¸ thÓ s¸ d¸ng c,c hÖ qu¶n trÞ c¬ s¸ d÷ liÖu ®Ó t¹o vù qu¶n lý c,c c¬ s¸ d÷ liÖu nh- Microsoft Access, SQL Server...

Có ph,p: **CREAT database<tªn c¬ s¸ d÷ liÖu>**

VÝ d¸: **CREAT Database QLSV;**

4.2.2. T¹o b¶ng

B¶ng lù ®òi t-ìng ®-ìc s¸ d¸ng ®Ó tæ chøc vù l-u tr÷ d÷ liÖu, mét CSDL c¸ thÓ bao g¸m nhiÒu b¶ng, mçi b¶ng ®-ìc x,c ®Þnh duy nh¸t bëi mét tªn. Mét

bảng bao gồm mét tệp c,c dùng vụ c,c cét: mỗi mét dùng trong bảng biểu diễn cho mét thùc thố.

Câu lệnh **CREATE TABLE** xác số dùng xác định nghĩa mét bảng dữ liệu mới trong cơ sở dữ liệu.

- Bảng mới xác tạo ra số dùng với môc đích g× cả vai trò nh- thố nư.
- Cấu trúc của bảng bao gồm nh÷ng tr÷ng nư, mỗi mét tr÷ng cả ý nghĩa nh- thố nư trong việc biểu diễn dữ liệu, kiốu dữ liệu của tr÷ng lư g× tr÷ng xác cả cho phép nh÷n gi, tr÷ null hay kh÷ng.

Có ph,p:

CREATE TABLE tⁿ_bảng (tⁿ_cét loⁱ_d÷_liệu [not null],...)

Trong xác:

Tⁿ_bảng: lư x©u ký từ bắt kù kh÷ng cả ký hiệu tr÷ng vụ kh÷ng tr÷ng với c,c tổ khã. Xác dùng xác x,c xác duy nhất mỗi bảng trong CSDL, tⁿ bảng kh÷ng v÷t qu, 128 ký từ.

Tⁿ_cét: lư x©u ký từ bắt kù kh÷ng cả ký hiệu tr÷ng vụ lư duy nhất trong bảng.

Loⁱ_d÷_liệu: Gồm cả:

INTEGER: sè nguyên

SMALLINTEGER: sè nguyên tồ – 32768 ã 32767

DECIMAL (n,p): sè thập phân với xác định tòi xác lư n trong xác cả p ch÷ sè thập phân, kh÷ng kố đều ch÷m thập phân.

FLOAT: sè đều ph÷y xác.

CHAR(n): kiốu x©u ký từ cả xác định cè xác n, n ≤ 255.

VARCHAR(n): kiốu x©u ký từ cả xác định kh÷ng cè xác.

DATE: kiốu dữ liệu ngày th,ng.

Not null: với cột thuộc tính not null bắt buộc phải điền dữ liệu, chỗ để nhập định nghĩa {null= 0}.

VÝ dụ: Trong CSDL THUCTAP gồm cả 3 quan hệ, ta sẽ thiết lập các bảng sau:

Các lĩnh vực này định nghĩa bảng *SINHVIEN* với các trường MSSV (mã số sinh viên), HT (họ tên), NS (ngày sinh), QUE (quê), HL (học lực). Trong các lĩnh vực bảng *SINH VIEN* trường MSSV, HT, HL không thể null (tức là bắt buộc phải điền dữ liệu, trường NS, QUE sẽ không, trường null nếu ta không nhập gì, trường cho trống).

Tạo bảng SV:

```
CREATE TABLE SINHVIEN
(MSSV INTEGER NOT NULL,
HT VARCHAR (20) NOT NULL,
NS DATE,
QUE VARCHAR (15),
HL DECIMAL (8,2) NOT NULL)
```

Tạo bảng DETAI:

```
CREATE TABLE DETAI
(MSDT INTEGER NOT NULL,
TDT VARCHAR (50) NOT NULL,
CN VARCHAR (20) NOT NULL,
KP INTEGER NOT NULL);
```

Tạo bảng SD:

```
CREATE TABLE SD
(MSSV INTEGER NOT NULL
```

MSDT INTEGER NOT NULL
 NTT VARCHAR (20) NOT NULL
 KM INTEGER NOT NULL
 KQ VARCHAR (5) NOT NULL);

Khi ®· thùc hiÖnh t¹o b¶ng ta s¶ thùc hiÖnh c,c c©u lÖnh ®Ó bæ sung d÷ liÖu cho b¶ng. Cã 3 c, ch biÓu diÖn:

C©u lÖnh cã c¸c ph, p nh- sau:

INSERT INTO ten_bang (danh_s, ch_ten_cot)
VALUES (cac_gia_trÞ)

VÝ dõ : Ta bæ sung d÷ liÖu cho b¶ng SINHVIEN

NÕu vÞ trÝ c,c cét trong b¶ng lµ cè ®Þnh ta cã thÓ viÖt:

INSERT INTO SINHVIEN
 VALUES (10, ‘Nguyen thi Hoa’, 1981, ‘HaNoi’, ‘Kha’);
 INSERT INTO SINHVIEN (MSSV, HODEM, TEN)
 VALUES (11, ‘Tran thi Lan’, ‘Trung Binh’).
 INSERT INTO SINHVIEN (MSSV, HODEM, TEN, QUE, HL)
 VALUES (12, ‘Nguyen hoai Nam’, ‘Nam Dinh’, ‘Kha’);

Ta s¶ cã d÷ liÖu trong b¶ng SINHVIEN nh- sau:

MSSV	HODEM	TEN	NS	QUE	HL
MS001	Phuong	Anh	1983	Thai Nguyen	7,2
MS002	Nguyen Thi	Hanh	1983	Thai Nguyen	7,0
MS003	Tran	Nam	1984	Thai Nguyen	6.9
MS004	Tran	Quan	1983	Hue	7.4
MS005	Khanh	Chi	1983	Ha Noi	6.5
MS006	Nguyen Thi	Sinh	1984	Ha Nam	7.6
MS007	Le Minh	Tan	1985	Thai Nguyen	7.3
MS008	Cam Duc	Nam	1985	Son La	5.9
MS009	Le Son	thao	1980	Ha Tay	5.5
MS0010	Nguyen thi	Hoa	1981	HaNoi	7.3

MS0011	Tran thi	Lan	Null	null	5.8
MS0012	Nguyen hoai	Nam	null	Nam Dinh	7.0

4.2.3. Sửa lại cấu trúc bảng

Một bảng sau khi đã được phân nghĩa bằng câu lệnh **CREATE TABLE** cả có thể sửa lại thông qua câu lệnh **ALTER TABLE**. Câu lệnh này cho phép ta thực hiện các thao tác sau:

- Bào sung một cột vào bảng.

ALTER TABLE t^an_bảng

ADD COLUMN T^an_cột

- Xóa một cột khỏi bảng.

ALTER TABLE t^an_bảng

DROP COLUMN t^an_cột

- Thay lại phân nghĩa của một cột trong bảng.

ALTER TABLE t^an_bảng

MODIFY COLUMN t^an_cột kiểu_dữ_liệu [null/ not null]

VÝ dƠ: Dưa vào ví dƠ CSDL THUCTAP ta đã tạo lại bảng SINHVIEN vào bảng DETAI.

```
CREATE TABLE SINHVIEN
(MSSV VARCHAR (5) NOT NULL,
```

```
HT VARCHAR (20) NOT NULL,
NS DATE NULL,
QUE VARCHAR (15) NULL,
HL DECIMAL (8,2) NOT NULL);
```

```
CREATE TABLE DETAI
(MSDT VARCHAR (5) NOT NULL,
TDT VARCHAR (50) NOT NULL,
CN VARCHAR (20) NOT NULL,
KP INTEGER NOT NULL);
```

Chúng ta cần thay đổi cấu trúc của bảng như sau:

- Thêm cột DIENTHOAI có kiểu dữ liệu là text 20 ký tự vào bảng ở quan hệ SINHVIEN.

```
ALTER TABLE SINHVIEN
ADD COLUMN DIENTHOAI text(20);
```

- Sửa kiểu dữ liệu của cột DIENTHOAI.

```
ALTER TABLE SINHVIEN
MODIFY COLUMN DIENTHOAI double;
```

- Sửa định nghĩa của kiểu dữ liệu cho cột HL trong bảng SV cho phép cột này chấp nhận giá trị null.

```
ALTER TABLE SINHVIEN
MODIFY COLUMN HL DECIMAL (8,2) NULL;
```

- Xóa cột DIENTHOAI từ bảng và thêm vào ở trên.

```
ALTER TABLE SINHVIEN
DROP COLUMN DIENTHOAI;
```

- Ta xóa cột KP trong bảng ở tại:

```
ALTER TABLE DETAI
```

DROP COLUMN KP;

4.2.4. Lệnh xóa bảng

Khi mét bảng không cần thiết ta cần xóa bỏ nó ra khỏi CSDL bằng câu lệnh DROP TABLE. Câu lệnh cần có ph, p nh- sau:

DROP TABLE tên_bảng

Trong các hồ qu, n tr, sẽ d÷ li, u, khi xóa mét bảng bằng lệnh *DROP TABLE*, ta không th, kh, i ph, c l, i bảng cùng nh- d÷ li, u của nó ta cần ph, i th, n tr, ng khi số dòng câu lệnh n, y.

Bảng cần @-ic ch, ra trong m, n @-i sẽ @-ic xóa khỏi CSDL.

V, d, : DROP TABLE SD;

(Khi @, bảng SD sẽ b, xóa khỏi CSDL THUCTAP).

4.3. Qu, n tr, d÷ li, u

Qu, n tr, d÷ li, u cần nh, l, ta truy xu, t d÷ li, u v, i câu lệnh SELECT, câu lệnh SELECT @-ic đ, ng @-i truy xu, t d÷ li, u t, c, c đ, ng v, i c, c cét của mét hay nhi, u bảng, khung nh, n. Câu lệnh n, y cần th, đ, ng @-i th, c hi, n ph, p ch, n (t, c l, truy xu, t mét t, p con c, c đ, ng trong mét hay nhi, u bảng), ph, p chi, u (t, c l, truy xu, t mét t, p con c, c cét trong mét hay nhi, u bảng) v, i ph, p n, i (t, c l, li, n k, t c, c đ, ng trong hai hay nhi, u bảng @-i truy xu, t d÷ li, u) ngo, i ra câu lệnh n, y cần cung c, p kh, n, ng th, c hi, n c, c thao t, c truy v, n v, i th, ng k, d÷ li, u ph, c t, p kh, c.

Câu tr, c cần lệnh truy v, n SQL

Câu tr, c cần sẽ cần mét bi, u th, c SQL bao g, m 3 m, n @-i: SELECT, FROM v, i WHERE.

- M, n @-i SELECT t, -ng ơng v, i ph, p chi, u của @-i sẽ quan h, . nó @-ic số dòng @-i li, t k, c, c cét mong mu, n trong k, t qu, của mét truy v, n.

- M, n @-i FROM t, -ng ơng v, i ph, p t, ych @-i c, c của @-i sẽ quan h, , nó li, t k, c, c bảng cần tra c, u @-i @-i, nh gi, k, t qu, của bi, u th, c.

- Mệnh đề WHERE dùng với vế tổ chức của câu sẽ quan hệ, nhằm bao gồm một vế tổ chức theo các cột của bảng xuất hiện trong mệnh đề FROM.

Số mệnh đề khác nhau của ngôn ngữ này, mệnh đề SELECT - FROM - WHERE sẽ được bổ sung thêm các mệnh đề GROUP BY, HAVING, ORDER BY.

Có phép chung của câu lệnh SELECT đã nêu:

```
SELECT [* DISTINCT] danh_s, ch_chiỐu
FROM danh_s, ch_t^n_bảng
[WHERE biỐu_thøc_điều kiện]
[GROUP BY danh_s, ch_t^n_cột]
[HAVING biỐu_thøc_điều kiện]
[ORDER BY {t^n_s^p_xếp}]
[ASC/DESC]
```

Điều cần lưu ý câu lệnh này lấy các thành phần trong câu lệnh SELECT nêu được số dòng phải tuân theo đúng thứ tự - trong có phép. Nếu không câu lệnh sẽ xem như - lấy không tập hợp.

4.3.1. Tập kiểm tra điều kiện

Các thao tác với tập kiểm tra của SQL dựa trên phép, như x¹ lấy các bảng và sẽ biểu diễn về mặt có phép lấy một mệnh đề:

SELECT – FROM – WHERE

Trong đó:

Select: x, c biểu diễn nội dung các cột cần đưa ra kết quả.

From: X, c biểu diễn các bảng cần lấy thông tin ra.

Where: X, c biểu diễn các bảng ghi thừa nhận điều kiện cần lọc.

- Mệnh đề SELECT dùng với phép chiếu của câu sẽ quan hệ. nhằm được số dòng có liệt kê các cột mong muốn trong kết quả của một truy vấn.

- Mệnh đề FROM t-đng đng vớí phđp tÝch ®Ò c,c cũa ®¹i sè quan hÖ, nã liÖt k^a c,c b¶ng cÇn tra cøu ®Ó ®,nh gi, kÖt qu¶ cũa biÓu thøc.

- Mệnh đề WHERE t-đng đng vớí vÞ tã chän cũa ®¹i sè quan hÖ, nã bao gôm mét vÞ tã kĐo theo c,c cét cũa b¶ng xuÊt hiÖn trong mệnh đề FROM.

§Ó më réng kh¶ nã cũa ng«n ng÷ nÿy, khèi SELECT - FROM - WHERE sã ®-íc bæ sung thªm c,c mệnh đề GROUP BY, HAVING, ORDER BY.

4.3.1.1. T×m kiÖm ®-n gi¶n

a. T×m kiÖm kh«ng ®iÖu kiÖn

T×m kiÖm ®-n gi¶n cũa nghÜa lÿ chØ lÿm quen vớí nh÷ng c©u hái liªn quan tíi mét b¶ng. Trong mệnh đề Select cũa danh s, ch chiÖu. Danh s, ch nÿy x,c ®Þnh tªn c,c cét cÇn cũa trong b¶ng kÖt qu¶.

Có ph,p cũa c©u lÖnh nÿy cũa d¹ng:

SELECT [DISTINCT] danh_s, ch_chiÖu*

FROM danh_s, ch_tªn_b¶ng

C©u lÖnh SELECT ®-íc sã dõng ®Ó t,c ®éng lªn c,c b¶ng d÷ liÖu, kÖt qu¶ cũa c©u lÖnh cõng ®-íc hiÖn thÞ d-íi d¹ng b¶ng, tøc lÿ mét tËp hîp c,c dßng, c,c cét.

VÝ dô: KÖt qu¶ cũa c©u lÖnh sau ®©y sã cho ta biÖt vÒ m· sè ®Ò tÿi, tªn ®Ò tÿi, chñ nhiÖm ®Ò tÿi cũa c,c ®Ò tÿi hiÖn cũa.

SELECT MSDT, TDT, CN

FROM DETAI;

C©u lÖnh ®-íc thÓ hiÖn trªn ®¹i sè quan hÖ nh- sau:

$\prod_{\text{MSDT, TDT, CN}}(\text{DETAI})$

Ta cũa b¶ng kÖt qu¶:

MSDT	TDT	CN
------	-----	----

DT001	Quan ly kinh doanh	Nguyen Tuan Dung
DT002	Quan ly tai chinh	Nguyen Thi Bien
DT003	Quan ly nhan su	Vu Thi Thuy Thao
DT004	Quan ly tien von	Nguyen Chi Hieu
DT005	Quan ly dien	Trinh Thi Ha
DT006	Quan ly nuoc	Do Duy Cop
DT007	Quan ly thu vien	Nguyen Tuan Dung
DT008	Quan ly sach	Nguyen Thi Bien
DT009	Quan tri mang	Vu Thi Thuy Thao
DT010	Thiet ke mang Lan	Nguyen Chi Hieu
DT011	Quan ly thue bao DD	Trinh Thi Ha
DT012	Quan ly KD may tinh	Do Duy Cop

♦ *C₃ch chặn c©u lÖnh trong SELECT.*

Chặn tÊt c¶ c,c cét trong b¶ng: Khi c¶n hiÖn th¶ tÊt c¶ c,c tr-êng trong b¶ng, sô dung ký tù * trong danh s, ch chặn thay v× liÖt k^a danh s, ch tÊt c¶ c,c cét. Trong tr-êng hÏp nuy, c,c cét ®-íc hiÖn th¶ trong kÖt qu¶ truy vÊn s¶ tu©n theo thø tù mụ chóng ®· ®-íc t¶o ra khi b¶ng ®-íc ®¶nh nghÜa.

VÝ dô: C©u lÖnh sau ®©y s¶ hiÖn th¶ danh s, ch ®Ò tui trong quan hÖ.

SELECT *

FROM DETAI;

KÖt qu¶ thu ®-íc:

MSDT	TDT	CN	KP
DT001	Quan ly kinh doanh	Nguyen Tuan Dung	2
DT002	Quan ly tai chinh	Nguyen Thi Bien	1.5
DT003	Quan ly nhan su	Vu Thi Thuy Thao	3
DT004	Quan ly tien von	Nguyen Chi Hieu	1.8
DT005	Quan ly dien	Trinh Thi Ha	1
DT006	Quan ly nuoc	Do Duy Cop	2.5
DT007	Quan ly thu vien	Nguyen Tuan Dung	2

DT008	Quan ly sach	Nguyen Thi Bien	1.5
DT009	Quan tri mang	Vu Thi Thuy Thao	3
DT010	Thiet ke mang Lan	Nguyen Chi Hieu	1.8
DT011	Quan ly thue bao DD	Trinh Thi Ha	1
DT012	Quan ly KD may tinh	Do Duy Cop	2.5

Trong tr-êng hîp cÇn chØ ®Þnh cô thÓ c,c cét cÇn hiÕn thÞ trong kÕt qu¶ truy vÊn, ta chØ ®Þnh danh s,ch c,c t¹n cét trong danh s,ch chän. Thø tù cña c,c cét kÕt qu¶ tu©n theo thø tù cña c,c tr-êng trong danh s,ch chän.

VÝ dõ: C©u lÖnh sau sÏ cho biÕt m· ®Ò t¹i, m· sinh viªn, n-i thùc tÛp cña b¶ng SD.

SELECT MSDT, MSSV, NTT

FROM SD;

C©u lÖnh ®-îc thÓ hiÕn trªn ®-i sè quan hÖ nh- sau:

$\Pi_{\text{MSDT, MSSV, NTT}}(\text{SD})$

C©u lÖnh SELECT dïng ®Ó t,c ®éng l¹n c,c b¶ng d÷ liÖu vµ kÕt qu¶ hiÕn thÞ d-íi dïng b¶ng g¸m c,c dïng vµ c,c cét m· ®Ò t¹i, m· sinh viªn, n-i thùc tÛp c¸ nh- sau:

MSDT	MSSV	NTT
DT001	MS001	Thai Nguyen
DT002	MS002	Thai Nguyen
DT003	MS003	Lang Son
DT004	MS004	Hue
DT005	MS005	Ha Noi
DT006	MS006	TuyenQuang
DT007	MS007	Yen Bai
DT008	MS008	Lao Cai
DT009	MS009	Ha Noi

Trong kết quả truy vấn cả thảy xuất hiện các dòng khác nhau, đó là bất cứ các dòng này, ta chèn thêm từ khóa DISTINCT ngay sau từ khóa SELECT

Ví dụ: Cho danh sách các tỉnh thành Việt Nam như sau:

Cho hai câu lệnh dưới đây:

```
SELECT DISTINCT NTT
```

```
FROM SD;
```

```
Và SELECT NTT
```

```
FROM SD;
```

Câu lệnh trên sẽ cho kết quả như sau:

$$\Pi_{NTT}(SD)$$

Kết quả là như sau:

NTT
Thai Nguyen
Thai Nguyen
Thai Nguyen
Hue
Ha Noi
Ha Noi
Yen Bai
Lao Cai
Ha Noi
Lang Son
Lang Son
TuyenQuang

TuyenQuang

NTT
Thai Nguyen
Lang Son
Hue
Hue
TuyenQuang
Yen Bai
Lao Cai
Ha Noi



✦ Mệnh đề FROM trong câu lệnh SELECT để chỉ danh sách các bảng tham gia truy vấn. Sau FROM là danh sách tên của các bảng tham gia truy vấn.

Ví dụ: Câu lệnh sau sẽ hiển thị danh sách sinh viên trong cơ sở dữ liệu THUCTAP.

SELECT *

FROM SINHVIEN;

Đều ‘*’ câu lệnh trên biểu thị danh sách các thuộc tính của bảng SINHVIEN. Bảng kết quả sẽ hiển thị:

MSSV	HODEM	TEN	NS	QUE	HL
MS001	Phuong	Anh	1983	Thai Nguyen	7,2
MS002	Nguyen Thi	Hanh	1983	Thai Nguyen	7,0
MS003	Tran	Nam	1984	Thai Nguyen	6.9
MS004	Tran	Quan	1983	Hue	7.4
MS005	Khanh	Chi	1983	Ha Noi	6.5
MS006	Nguyen Thi	Sinh	1984	Ha Nam	7.6
MS007	Le Minh	Tan	1985	Thai Nguyen	7.3
MS008	Cam Duc	Nam	1985	Son La	5.9
MS009	Le Son	Thao	1980	Ha Tay	5.5

MS09	Nguyen thi	Hoa	1981	HaNoi	7.3
MS010	Tran thi	Lan	1981	HaNoi	5.9
MS011	Nguyen Hoai	Nam	1981	HaNoi	5.5

b. $T \times m$ kiểm với $\text{@i@ou ki@n @-n gi@n}$

Ch@ sø d@ng kh@i select-from-where

Có ph,p c@i c@u l@nh n@y c@ d@ng:

SELECT [*** DISTINCT**] danh_s,ch_chi@u
FROM danh_s,ch_t@n_b@ng
[WHERE bi@u_th@c_@i@ou ki@n]

VÝ d@: @-a ra danh s,ch nh÷ng sinh vi@n t@n l@ TAN

SELECT *
FROM SINHVIEN
WHERE TEN= 'TAN';

C@u l@nh @-ic th@ hi@n tr@n @-i s@ quan h@ nh- sau:

$\sigma_{HT='TAN'}(SINHVIEN)$

M@nh @Ò **WHERE** trong c@u l@nh **SELECT** @-ic sø d@ng nh»m x,c @Đnh c,c @i@ou ki@n @@i v@i vi@c truy xuÊt d÷ li@u. Sau m@nh @Ò **WHERE** l@ mét bi@c th@c logic v@ ch@ nh÷ng đ@ng d÷ li@u n@o th@a m·n @i@ou ki@n.

L-u ý: N@u truy vÊn @-ic th@c hi@n tr@n nhi@u b@ng, kh@ng nh×n c@ c,c tr-êng tr@ng t@n nh@u th× t@n c@i c,c tr-êng n@y n@u xuÊt hi@n trong danh s,ch ch@n ph@i @-ic vi@c d-@i d@ng:

$T@n_b@ng. T@n_tr-êng$

VÝ d@: C@u l@nh sau s@ cho ta bi@c m· @Ò t@i, t@n @Ò t@i, n-i th@c t@p @-ic truy suÊt d÷ li@u t@ b@ng DETAI, SD

SELECT DT.MSDT, TDT, NTT
FROM DETAI, SD

WHERE DT.MSDT = SD. MSDT;

C©u lÖnh ®-íc thÓ hiÖn trªn ®¹i sè quan hÖ nh- sau:

$\Pi_{\text{MSDT, TDT, NTT}}(\text{SD} * \text{DETAI})$

MÖnh ®Ò WHERE: trong mÖnh ®Ò WHERE th-êng sô dông:

- C,c to,n tö kÖt hÖp ®iÖu kiÖn (AND, OR).
- C,c to,n tö so s,nh (=, >, <, >=, <=, <>, !>, !<).
- KiÓm tra giú h¹n cªa d÷ liÖu (BETWEEN/ NOT BETWEEN).
- Danh s,ch (IN/ NOT IN).
- To,n tö LIKE
- C,c to,n tö kÖt hÖp ®iÖu kiÖn (AND, OR).

VÝ dõ: Danh s,ch c,c sinh viªn tr¶ (d-úi 22 tuæi) vµ (HL> = 7.5).

SELECT *

FROM SINHVIEN

WHERE 2006- NS < 22 and HL >= 7.5;

(ë c©u lÖnh trªn chóng ta truy suÊt d÷ liÖu tõ b¶ng SINHVIEN tho¶ m·n ®iÖu kiÖn c,c sinh viªn d-úi 18 tuæi vµ hãc lúc lín h-n 7.5).

B¶ng kÖt qu¶ ®-íc hiÖn th¶:

MSSV	HODEM	TEN	NS	QUE	HL
MS006	Nguyen Thi	Sinh	1984	Ha Nam	7.6

C©u lÖnh ®-íc thÓ hiÖn trªn ®¹i sè quan hÖ nh- sau:

$\sigma_{2006 - NS < 22 \wedge HL > = 7.5}(\text{SINHVIEN})$

4.3.1.2. T×m kiÖm cã ®iÖu kiÖn

Có ph,p cªa c©u lÖnh nuy cã d¹ng:

SELECT [DISTINCT] danh_s,ch_chiÖu*

```

FROM danh_s,ch_t^n_b¶ng
[WHERE biÓu_thøc_®iÒu kiÖn]
[GROUP BY danh_s,ch_t^n_cét]
[HAVING biÓu_thøc_®iÒu_kiÖn]

[ORDER BY {cét_s¾p_xÕp}

[ASC/DESC]]

```

Trong ®ã:

GROUP BY: chia th¶nh quan hÖ c,c nh¶m, mçi nh¶m øng víi mét sè hiÖu, sau ®ã c,c thao t,c s¶ ®-íc thùc hiÖn trªn c,c nh¶m.

HAVING: l¶ d¶ng ®Æc biÖt c¶a WHERE luªn di sau GROUP BY, sau Having l¶ mét biÓu thøc ®iÒu kiÖn, mÖnh ®Ò n¶y khªng t,c ®éng v¶o to¶n b¶ng ®-íc chØ ra ë mÖnh ®Ò FROM m¶ chØ t,c ®éng l¶n l-ít v¶o c,c nh¶m b¶n ghi chØ ra ë mÖnh ®Ò GROUP BY

ORDER BY DESC/ASC: l¶ mÖnh ®Ò t×m kiÖm c¶ s¾p xÕp, ®Ó s¾p xÕp theo chiÒu gi¶m hoÆc tªng.

VÝ dõ: Thªng tin vÒ c,c ®Ò t¶i ®-íc cÊp kinh phÝ trªn 10 triÖu

```

SELECT *
FROM DETAI
WHERE KP > 10;

```

C©u lÖnh ®-íc thÓ hiÖn trªn ®¶i sè quan hÖ nh- sau:

```

σKP > 10(DETAI)

```

► LÖp danh s,ch ®Ò t¶i c¶ m· sè l¶ 001, 002, 005, 007 v¶ kinh phÝ ®-íc s¾p xÕp gi¶m d¶n:

```

SELECT MSDT, TDT, KP
FROM DETAI

```


WHERE MSDT IN ('DT001', 'DT002', 'DT005', 'DT007')
ORDER BY KP DESC;

Bảng kết quả trả về hiển thị:

MSDT	TDT	KP
DT005	Quan ly dien	1
DT002	Quan ly tai chinh	1.5
DT001	Quan ly kinh doanh	2
DT007	Quan ly thu vien	2

a, Tìm kiếm các ký tự

Việc tìm kiếm các ký tự ngoại việc xử lý các SQL cần cho phép xử lý các ký tự khi cần khi cần sử dụng các ký tự “%” để thay thế cho một ký tự, các ký tự “_” để thay thế cho một ký tự. Kết hợp với các ký tự LIKE.

A%B : ký tự bắt đầu ký tự B

%A: ký tự kết thúc ký tự A.

A_B: Ký tự giữa 2 ký tự A và B.

A_: Ký tự giữa hai ký tự A và B.

Ví dụ: Cho biết tên các sinh viên bắt đầu ký tự T

```
SELECT *
FROM SINHVIEN
WHERE TEN LIKE 'T%';
```

Kết quả thu về:

Hodem	Ten
Le Minh	Tan
Le Son	Thao

- Cho biết họ tên của các sinh viên có họ bắt đầu bằng 'NGUYEN'

```
SELECT HODEM, TEN
FROM SINHVIEN
WHERE HODEM LIKE 'NGUYEN%';
```

Ta có bảng kết quả:

Hodem	Ten
Nguyen Thi	Hanh
Nguyen Thi	Sinh

- Cho biết họ tên của các sinh viên có họ bắt đầu bằng 'TRAN' và tên có chữ cái cuối cùng là N.

```
SELECT HODEM, TEN FROM SINHVIEN
WHERE HT LIKE 'TRAN%' and HT LIKE '%N';
```

Kết quả thu được của câu lệnh:

Hodem	Ten
TRAN	NAM

b. Xử lý ngày tháng

Ngoài những phép tính trong thành ngữ SQL còn có thể xử lý dữ liệu ngày tháng.

Ví dụ: Tìm những đơn tại cả ngày thực tiếp trước 27/03/2006 là 10 ngày

```
SELECT TDT
FROM DETAI,SD
WHERE '27/03/2006' - NGAYTT = 10 and DT.MSDT = SD.MSDT;
```

Câu lệnh trên hiển thị các liên hệ sau:

$\Pi_{TDT}(\sigma_{'27/03/2006' - NGAYTT = 10}(DETAI * SD))$

c. $T \times m$ kiểm tra số đông IN và BETWEEN

♦ Số kiểm tra dữ liệu nằm trong hay ngoài một khoảng nào đó ta số đông to, nhỏ BETWEEN và NOT BETWEEN như sau:

Cách số đông ý nghĩa
BETWEEN a AND b $a \leq gi, tr \leq b$
NOT BETWEEN a AND b $(gi, tr < a)$ và $(gi, tr > b)$

Ví dụ: $T \times m$ nhân hàng ở tại cả kinh phí từ 1.5 đến 3 triệu

```
SELECT TDT
FROM DETAI
WHERE KP BETWEEN 1.5 and 3;
```

Câu lệnh để tìm kiếm nhân viên sẽ quan hệ như sau:

$\Pi_{TDT}(\sigma_{KP \geq 1.5 \wedge KP \leq 3}(DETAI))$

♦ Tìm kiếm IN để số đông khi ta cần chọn ra những nhân viên kiểm tra dữ liệu cho câu lệnh SELECT lấy một danh sách các giá trị. Sau IN (hoặc NOT IN) cả thôi lấy danh sách các giá trị hoặc một câu lệnh SELECT khác.

Ví dụ: Số biết các danh sách các ở tại cả mức sẽ ở tại mức 2, 4 hoặc 5 thay vì số đông câu lệnh.

```
SELECT *
FROM DETAI
WHERE MSDT = 'DT002' or MSDT = 'DT004' or MSDT = 'DT005';
```

► Ta cần thôi số đông câu lệnh

```
SELECT *
FROM DETAI
WHERE MSDT IN ('DT002', 'DT004', 'DT005');
```

Câu lệnh để tìm kiếm nhân viên sẽ quan hệ như sau:

$\sigma_{MSDT=002 \vee MSDT=004 \vee MSDT=005}(DETAI)$

VÝ dō: Cho biÕt danh s, ch c, c sinh viªn kh«ng cª MSSV lµ 007.

```
SELECT *  
FROM SINHVIEN  
WHERE MSSV NOT IN('MS007');
```

C©u lÖnh ®-íc thÓ hiÖn trªn ®¹i sè quan hÖ nh- sau:

$\sigma_{MSSV < 007} (SINHVIEN)$

4.3.2. T×m kiÖm vÖi ,nh x¹ l¶ng

§Ó t×m kiÖm trªn nhiÖu b¶ng qua viÖc sø dông ,nh x¹ l¶ng hoÆc phÐp kÖt nèi. ,nh x¹ l¶ng cª nghÖa lµ hai khøi SELECT l¶ng nhau, phÐp l¶ng cª cª thÓ ®-íc l¶ng nhiÖu m¸c. Lo¹i truy vÊn nµy ®-íc sø dông ®Ó biÓu diÖn cho nh÷ng truy vÊn trong ®ã ®iÖu kiÖn truy vÊn d÷ liÖu ph¶i sø dông ®Õn kÖt qu¶ cªa mét truy vÊn kh, c.

Có ph, p:

```
SELECT [* DISTINCT] danh_s, ch_chiÖu  
FROM danh_s, ch_tªn_b¶ng  
[WHERE biÓu_thøc_®iÖu_kiÖn]  
[GROUPE BY danh_s, ch_tªn_cét]  
[HAVING biÓu_thøc_®iÖu_kiÖn]
```

Chó ý: Mét truy vÊn con ph¶i ®-íc viÕt trong cÆp dÊu ngoÆc vµ th-êng cª kÖt qu¶ lµ mét cét.

♦ Khi cÇn thùc hiÖn kiÓm tra gi, trÞ cªa mét biÓu thøc cª xuÊt hiÖn (kh«ng xuÊt hiÖn) trong tÕp c, c gi, trÞ cªa truy vÊn con hay kh«ng ta cª thÓ sø dông to, n tö IN (NOT IN) nh- sau:

```
WHERE biÓu_thøc IN (NOT IN)  
( truy_v©n_con)
```

VÝ dō: Cho biÕt danh s, ch sinh viªn cª m· sè lµ 001, 007, 008.

```
SELECT *
FROM SV
WHERE MSSV IN (001, 007, 009);
```

Câu lệnh sẽ thực hiện trên các bảng sau:

$\sigma_{MSSV = 001 \vee MSSV = 007 \vee MSSV = 009}(\text{SINHVIEN})$

Ví dụ: Danh sách các sinh viên (đ-đ 18 tuổi) học lực giỏi (HL > 8.5 và KQ > 8.5)

```
SELECT HT
FROM SINHVIEN
WHERE 2006 - NS < 18 and HL > 8.5 and MSSV in
(SELECT MSSV
FROM SD
WHERE KQ > 8.5);
```

Câu lệnh sẽ thực hiện trên các bảng sau:

$\Pi_{HT}(\sigma_{2006 - NS < 18 \wedge HL > 8.5 \wedge KQ > 8.5}(\text{SINHVIEN} * \text{SD}))$

► Danh sách các thí sinh (đ-đ 18 tuổi) học lực giỏi (HL > 8.5 và KQ > 8.5) và Hạng Nhất tham gia:

```
SELECT CN
FROM DETAI
WHERE MSDT IN (SELECT MSDT FROM SD
WHERE MSSV not in
(SELECT MSSV FROM SV
WHERE QUE = 'HANOI'));
```

Câu lệnh sẽ thực hiện trên các bảng sau:

$\Pi_{CN}(\text{DETAI} * (\text{SD} * (\sigma_{QUE \neq 'HANOI'}(\text{SINHVIEN}))))$

♦ Kõt qu¶ truy vÊn con cã thÓ ®-íc sø dông ®Ó thùc hiÖn phÐp to,n sè hãc víi mét biÓu thøc truy vÊn cha, trong tr-êng hîp nuy truy vÊn ®-íc sø dông d-íi d¹ng:

*WHERE biÓu_ thøc_phÐp_to,n_sè_hãc [ANY/ALL]
(truy_v©n_con)*

Trong ®ã c,c phÐp to,n sè hãc cã thÓ sø dông bao g¸m =, <>, >, <, >=, <= vµ truy vÊn con cã kÕt qu¶ bao g¸m ®óng mét cét. NÕu truy vÊn con tr¶ vÒ nhiÒu h-n mét gi, trÞ th× viÖc sø dông phÐp so s,nh nh- tr¹n sã kh«ng hîp lÖ. NÕu sau c,c phÐp to,n ph¶i sø dông th¹m ALL hoÆc ANY. L-êng tõ ALL ®-íc sø dông khi cÇn so s,nh gi, trÞ cña biÓu thøc víi tÊt c¶ gi, trÞ tr¶ vÒ trong kÕt qu¶ cña truy vÊn con. Ng-íc l¹i, phÐp so s,nh víi l-êng tõ ANY cã kÕt qu¶ ®óng khi chØ cÇn mét kÕt qu¶ bÊt kú nµo ®ã trong kÕt qu¶ cña truy vÊn con tháa m·n ®iÒu kiÖn.

VÝ dõ: C©u lÖnh d-íi ®©y cho biÕt c,c ®Ò tui cã sè kinh phÝ lín h-n hoÆc b»ng sè kinh phÝ cña ®Ò tui cã m· lụ 7.

SELECT *
FROM DETAI
WHERE KP > = (SELECT KP FROM DT
WHERE MSĐT = 7);

C©u lÖnh ®-íc thÓ hiÖn tr¹n ®¹i sè quan hÖ nh- sau:

$\sigma_{KP \geq \pi_{KP}(\sigma_{MSDT=7}(DETAI))}(DETAI)$

► §-a ra danh s, ch c,c sinh vi¹n hãc gi¸i h-n c,c sinh vi¹n ẽ Hµ Núi.

SELECT HODEM, TEN
FROM SINHVIEN
WHERE HL > ALL (SELECT HL
FROM SV

WHERE QUE = 'HANOI');

Hay cả thÓ viÖt theo c₁ch kh₁c:

SELECT HODEM, TEN

FROM SINHVIEN

WHERE HL> (SELECT MAX (HL)

FROM SINHVIEN

WHERE QUE = 'HANOI');

✦ Ngoµi viÖc sö dông ,nh x¹ lảng ®Ó t×m kiÖm d÷ liÖu tã hai hay nhiÖu b¶ng chóng ta còng cã thÓ sö dông phÐp kÖt nèi. Mét c©u lÖnh nèi kÖt hîp c₁c dßng d÷ liÖu trong c₁c b¶ng kh₁c nhau l'i theo mét hoÆc nhiÖu ®iÖu kiÖn nµo ®ã hiÖn thÞ trong kÖt qu¶ truy vÊn.

Trong phÐp kÖt nèi, c₁c cét tham gia kÖt nèi ph¶i cã miÒn trÞ lµ s₁nh ®-íc vói nhau. T^n cét cña c₁c b¶ng kh₁c nhau cã thÓ viÖt t-êng minh qua t^n b¶ng.

◆ **VÝ dõ:** Cho th«ng tin vÒ c₁c ®Ò tûi cã sinh vi^n thùc tËp

SELECT DISTINCT *

FROM DETAI, SD

WHERE (DT.MSDT =SD.MSDT);

C©u lÖnh ®-íc thÓ hiÖn tr^n ®i sè quan hÖ nh- sau:

DETAI*SD

► Cho danh s₁ch sinh vi^n thùc tËp t'i quª nhµ

SELECT HODEM, TEN

FROM SD, SINHVIEN

WHERE (SV.MSSV = SD.MSSV) and (SV.QUE = SD.NTT);

C©u lÖnh ®-íc thÓ hiÖn tr^n ®i sè quan hÖ nh- sau:

$\Pi_{HODEM, TEN}(\sigma_{NTT = QUE} (SINHVIEN*SD))$

4.3.3. T×m kiÖm vói mÖnh ®Ò GROUP BY

MÖnh ®Ò GROUP BY: chia quan hÖ th×nh c,c nhãm mçi nhãm øng vói mét sè hiÖu, sau ®ã c,c thao t,c sÏ ®-íc thùc hiÖn trªn c,c nhãm, thùc hiÖn trªn c,c dñng c,c b¶n ghi vµ mÉu tin cũ d÷ liÖu giềng nhau sÏ ®-íc xÕp vµo th×nh mét nhãm.

◆ VÝ dō: LiÖt kª danh s, ch gãm MSSV, HODEM, TEN, HL, KQ cũa ®Ò tñi cũ m· sè lụ 5, theo thø tù KQ.

```
SELECT MSSV, HODEM, TEN, HL, KQ
FROM SINHVIEN,SD
WHERE (SD.MSDT='DT005') and (SINHVIEN.MSSV = SD.MSSV)
GROUP BY SD.KQ;
```

► §-a ra tªn nh÷ng sinh viªn thùc tẾp tªi quª nhµ, nhãm theo que.

```
SELECT HODEM, TEN
FROM SINHVIEN,SD
WHERE (SD.NTT = SV.QUE) and (SINHVIEN.MSSV = SD.MSSV)
GROUP BY SINHVIEN.QUE;
```

✦ T×m kiÖm cũ sō dōng mÖnh ®Ò HAVING:

HAVING lụ dñng ®Æc biÖt cũa where lụ«n ®i sau GROUP BY, sau HAVING lụ mét biÓu thøc ®iÖu kiÖn, biÓu thøc ®iÖu kiÖn nµy kh«ng t,c ®éng vµo toµn b¶nng ®-íc chØ ra ẽ mÖnh ®Ò FROM mµ chØ t,c ®éng lÇn l-ít tō nhãm c,c b¶n ghi ®· chØ ra tªi mÖnh ®Ò GROUP BY. Th-êng th× HAVING ®i vói mét sè hµm th- viÖn.

COUNT - cho sè l-îng phÇn tō cũa cét

SUM - cho tæng c,c trÞ trong cét

MIN - cho gi, trÞ nhá nhÊt trong cét

MAX - cho gi, trÞ lín nhÊt trong cét

AVG - cho gi, trÞ trung b×nh céng cũa cét

VÝ dō: §-a ra danh s, ch @Ò tui cũ Ýt nhÊt lư 3 sinh vi^n @`ng ký

```
SELECT SD.MSDT, TDT, CN, NTT, KP
FROM DETAI, SD
WHERE (DT.MSDT = SD.MSDT)
GROUP BY SD.MSDT
HAVING COUNT (SD.MSDT) ≥ 3;
```

♦ T×m kiÕm cũ s¼p xÕp

§Ó t×m kiÕm cũ s¼p xÕp ta dùng mÕnh @Ò ORDER BY DESC/ASC-s¼p gi¶m/t`ng. Sau mÕnh @Ò BY lư t^n cét rã @Õn chiÒu s¼p xÕp t`ng hoÆc gi¶m. Cũ thÓ s¼p xÕp nhiÒu cét vư nÕu kh«ng chØ ra chiÒu s¼p xÕp th× hÕ thèng ngÇm @¶nh lư ASC.

MÕnh @Ò ORDER BY nÕu @øng sau GROUP BY th× miÒn t,c @éng cũ s¼p xÕp lư trong tōng nhãm cũ cét @-íc chØ ra trong GROUP BY. NÕu cét chØ s¼p xÕp cũ mÕnh @Ò SELECT th× trong mÕnh @Ò ORDER BY cũ thÓ chØ cÇn chØ ra thø tũ cũ cét @ã trong dang s, ch chiÕu lư @ñ.

VÝ dō: LÈp danh s, ch sinh vi^n cũ hãc lúc @-íc s¼p xÕp gi¶m dÇn

```
SELECT MSSV, HODEM, TEN, HL
FROM SINHVIEN
ORDER BY HL DESC;
```

MSSV	HODEM	TEN	HL
MS01	Nguyen thi	Hoa	7.5
MS03	Nguyen hoai	Nam	7.3
MS02	Tran thi	Lan	6.8

- T×m t^an nh÷ng sinh vi^an cã MSDT lụ 5 vự s³4p xÕp theo thø từ t^ìng dÇn cña kÕt qu¶ thùc tËp.

```
SELECT TEN
FROM SINHVIEN, SD
WHERE MSDT = 'DT005' AND (SINHVIEN.MSSV=SD.MSSV)
ORDER BY KQ ASC;
```

4.3.4. T×m kiÖm sø dông c,c phÐp to,n tËp hùp

VÝ dƠ: Cho biÕt m· cña c,c ®Ò tụi kh«ng cã sinh vi^an nưo tham gia:

```
SELECT MSDT
FROM DETAI
MINUS
SELECT DISTINCT MSDT
FROM SD;
```

C©u lÖnh ®-íc thÓ hiÖn trªn ®i sè quan hÖ nh- sau:

$$\Pi_{\text{MSDT}}(\text{DT}) - \Pi_{\text{MSDT}}(\text{SD})$$

4.3.5. T×m kiÖm vói danh s,ch ®Ých trªn nhiÒu b¶ng

T×m kiÖm vói danh s,ch ®Ých trªn nhiÒu b¶ng là c¬ sè ®Ó thùc hiÖn c,c yªu cÇu truy vÊn liªn quan ®Õn nhiÒu b¶ng, c©u lÖnh thùc hiÖn lÊy c,c dñng d÷ liÖu trong c,c b¶ng tham gia truy vÊn, so s,nh gi, trÞ cña c,c dñng nựy trªn mét hoÆc nhiÒu cét ®-íc chØ ®Þnh trong ®iÒu kiÖn nòi vự kÕt hùp c,c dñng tháa m·n ®iÒu kiÖn thựnh nh÷ng dñng trong kÕt qu¶ truy vÊn.

♦ T×m kiÖm vói danh s,ch ®Ých trªn nhiÒu b¶ng b¾t ®Çu b»ng tữ khãa SELECT, c,c cét ®-íc chØ ®Þnh sau tữ khãa lụ c,c cét hiÖn thÞ trong kÕt qu¶ truy vÊn. ViÖc sø dông tªn c,c cét trong danh s,ch chãn cã thÓ lụ:

- Tªn cña mét sè cét nưo ®ã trong c,c b¶ng tham gia vựo truy vÊn, nÕu tªn cét trong c,c b¶ng trõng tªn nhau th× tªn cét ph¶i ®-íc viÖt d-i d¹ng

Ten_bang. Ten_cét

- DÊu * @-íc sô dông trong danh s, ch chän khi cÇn hiÓn thÞ tÊt c¶ c, c cét cña b¶ng tham gia truy vÊn.

- ✦ TiÕp theo lụ mÖnh @Ò FROM, sau mÖnh @Ò FROM cña c©u lÖnh kÕt nèi lụ danh s, ch trªn c, c b¶ng tham gia truy vÊn. NÕu ta sô dông dÊu * trong danh s, ch chän th× thø tù cña b¶ng liÕt kª sau FROM sÏ ¶nh h-ëng @Ön thø tù c, c cét @-íc hiÓn thÞ trong kÕt qu¶ truy vÊn.

- ✦ Khi hai hay nhiÒu b¶ng @-íc nèi víi nhau ta ph¶i chØ @Þnh @iÒu kiÖn @Ó thùc hiÖn phÐp nèi ngay sau mÖnh @Ò WHERE. §Òu kiÖn nèi @-íc biÓu diÔn d-íi d¹ng mét biÓu thøc logic so s, nh gi, trÞ d÷ liÖu gi÷a c, c cét cña c, c b¶ng tham gia truy vÊn.

VÝ dõ: C©u lÖnh sau sÏ hiÓn thÞ danh s, ch sinh viªn @Ò tui víi th«ng tin: MSSV, MSDT, TEN, TDT, NTT.

SELECT SD.MSSV, MSDT, TEN, TDT, NTT

FROM SINHVIEN, DETAI, SD

WHERE (SV.MSSV = SD.MSSV) and (SD.MSDT = DT.MSDT);

(Trong c©u lÖnh trªn c, c b¶ng tham gia vµo truy vÊn SV, SD, DT. §iÒu kiÖn @Ó thùc hiÖn thùc hiÖn phÐp nèi gi÷a c, c b¶ng g¸m hai @iÒu kiÖn.

SINHVIEN.MSSV = SD.MSSV

vµ DETAI.MSDT = SD.MSDT

§iÒu kiÖn t×m kiÕm danh s, ch trªn nhiÒu b¶ng @-íc x, c @Þnh @-íc trªn mèi quan hÖ gi÷a c, c b¶ng trªn c¬ sÏ d÷ liÖu).

C©u lÖnh @-íc thÓ hiÖn trªn @i sè quan hÖ nh- sau:

$\Pi_{MSSV,MSDT,TEN,TDT,NTT}((SV*SD)DT)$

4.4. C, c hµm th- viÖn

Trong SQL g¸m c¸ 6 hµm th- viÖn:

COUNT - cho sè l-íng phÇn tö cña cét

SUM - cho tæng c,c trÞ trong cét

MIN - cho gi, trÞ nhá nhÊt trong cét

MAX - cho gi, trÞ lín nhÊt trong cét

AVG - cho gi, trÞ trung b×nh céng cña cét

♦ Hµm count:

§-íc sô dông ®Ó ®Öm sè dßng cña mét cét, mét b¶ng. Nh-ng khi hµm Count cũ ®èi sè lµ "*" cũ nghÜa lµ ®Öm sè b¶n ghi tháa m·n yªu cÇu t×m kiÖm mµ kh«ng quan t©m tíi bÊt cú mét cét nµo.

VÝ dõ: H·y ®Öm danh s, ch c,c tØnh vµ sè sinh viªn quª ã tØnh ®ã

```
SELECT COUNT.QUE
```

```
FROM SINHVIEN;
```

► H·y ®Öm c,c MSDT trong b¶ng DT

```
SELECT COUNT MSDT
```

```
FROM DETAI;
```

NÕu nh- cÇu lÖnh ®-íc viÖt d-úi dßng nh- sau nã sÏ ®Öm tÊt c¶ c,c b¶n ghi tháa m·n yªu cÇu t×m kiÖm.

►

```
SELECT COUNT ' *'
```

```
FROM DETAI;
```

♦ Hµm SUM:

TÝnh tæng c,c gi, trÞ sè trong cét víi tªn cét ®· cho

VÝ dõ TÝnh tæng sè sinh viªn ®i thùc tËp

```
SELECT SUM (MSSV)
```

```
FROM SD;
```

► Tæng sè ®o¹n ®-êng thùc tËp cña ®Ò tµi 5

```
SELECT SUM (KM)
```

```
FROM SD
```

```
WHERE MSDT = 'DT005';
```

♦ Hàm AVG:

Hàm tính trung bình cộng AVG = AVERAGE.

Tính trung bình cộng của một bé sè.

Ví dụ: Tính điểm trung bình của các sinh viên ở Hà Nội

```
SELECT AVG (HL)
FROM SINHVIEN
WHERE QUE = 'HANOI';
```

♦ Hàm MIN, MAX:

Lấy các hàm ra giá trị lớn nhất và nhỏ nhất của các bé sè.

Ví dụ: Cho biết điểm thi lớn nhất của các sinh viên thi tại HUE.

```
SELECT MAX (2006 - NS),
FROM SINHVIEN
WHERE QUE = 'HUE';
```

► Cho biết điểm thi nhỏ nhất của các sinh viên thi tại HÀ NỘI.

```
SELECT MIN (2006 - NS).
FROM SINHVIEN
WHERE QUE = 'HUE';
```

► Cho biết kết quả thi tốt nghiệp cao nhất và thấp nhất của những sinh viên của NTT lấy ở quê nhà.

```
SELECT MAX (KQ)
FROM SD, SINHVIEN
WHERE (SD.MSSV = SINHVIEN .MSSV) and
      (SD. NTT = SV. QUE);
SELECT MIN (KQ)
FROM SD, SINHVIEN
WHERE (SD.MSSV = SINHVIEN .MSSV) and
      (SD. NTT = SV. QUE);
```

♦ SET: lụ hụm th- viÖn chØ tỄp

CHƯƠNG 5

LÝ THUYẾT THIẾT KẾ CƠ SỞ DỮ LIỆU QUAN HỆ

(Số tiết lý thuyết 12 tiết)

5.1. Phụ thuộc hàm

ThiÖt kÖ CSDL quan hÖ tÖc x, c ®Þnh CSDL ®ã cũ nh÷ng quan hÖ $g \times$?
Quan hÖ gãm t³n quan hÖ vµ ®-íc x, c ®Þnh bëi 1 tỄp c, c thuéc tÝnh vµ c, c bé.

ThiÖt kÖ mét $c \rightarrow$ sẽ d÷ liÖu nh»m:

- Cho phĐp qu¶n lý d÷ liÖu mét c, ch tỄp trung.

- T³o mét $c \rightarrow$ sẽ d÷ liÖu híp nhÊt, kh«ng d- thõa, ®¶m b¶o tÝnh nhÊt qu, n
cho d÷ liÖu.

- §¶m b¶o tÝnh an toµn cho d÷ liÖu.

- §¶m b¶o tÝnh ®éc lỄp cũa d÷ liÖu.

Khi thiÖt kÖ cÇn l-u ý ®Ön c, c vÊn ®Ò sữ n¶y sinh nh- sau:

- D- thõa d÷ liÖu.

- Kh«ng nhÊt qu, n (dÞ th-êng xuÊt hiÖn khi sõa d÷ liÖu).

- DÞ th-êng khi th³m bé.

- DÞ th-êng khi xo, bé.

Do vỄy khi thiÖt kÖ mét CSDL cũ thÓ ph©n chia quan hÖ thụn nh÷ng
quan hÖ kh, c nhau nh»m tr, nh nh÷ng vÊn ®Ò n³u tr³n nh»m ®¹t ®-íc mét l-íc
®ã CSDL (tỄp c, c l-íc ®ã quan hÖ) tèt h-n.

5.1.1. Khái niệm phụ thuộc hàm.

Kh, i niÖm vÒ phö thuéc hụm lụ mét quan niÖm cũ tÇm quan tr¾ng rÊt lín ®èi
vói viÖc thiÖt kÖ m« h×nh d÷ liÖu.

§Þnh nghÜa: Cho $R(U)$ lụ mét l-íc ®ã quan hÖ vói $U = \{A_1, \dots, A_n\}$ lụ tỄp
thuéc tÝnh. X vµ Y lụ hai tỄp con cũa U .

Những X, Y thuộc hàm Y hoặc Y phụ thuộc hàm vào X ($X \rightarrow Y$) nếu và chỉ nếu quan hệ $R(U)$ sao cho bất kỳ hai tuple $t_1, t_2 \in R$ mà

$$t_1[X] = t_2[X] \text{ thì } t_1[Y] = t_2[Y]$$

Phụ thuộc hàm ký hiệu là FD.

5.1.2. Một số ví dụ

Cho quan hệ phản công sau:

Phản công (PHICONG, MAYBAY, NGÀYKH, GIOKH)

ở đây quan hệ phản công diễn tả phi công lái máy bay nào, máy bay khởi hành vào thời gian nào. Không phải sự phối hợp bất kỳ giữa phi công, máy bay, giờ khởi hành nào cũng đều được chấp nhận mà chúng có các điều kiện ràng buộc quy định sau:

- + Mỗi máy bay có một giờ khởi hành duy nhất
- + Nếu biết phi công, biết giờ khởi hành thì biết máy bay do phi công ấy lái
- + Nếu biết máy bay, biết ngày khởi hành thì biết phi công chuyến bay ấy.

Các ràng buộc này là các ví dụ về phụ thuộc hàm và được phát biểu lại như sau:

- + MAYBAY xác định GIOKH (maybay phụ thuộc hàm vào giờ KH)
- + {PHICONG, NGÀYKH, GIOKH} xác định MAYBAY
- + {MAYBAY, GIOKH} xác định PHICONG

và được ký hiệu như sau:

- + {MAYBAY} $\rightarrow$ GIOKH
- + {PHICONG, NGÀYKH, GIOKH} $\rightarrow$ MAYBAY
- + {MAYBAY, GIOKH} $\rightarrow$ PHICONG

5.2. Tiên đề đối với phụ thuộc hàm

Giả F là tập tất cả các phụ thuộc hàm trên với lược đồ quan hệ $R(U)$ mà $X \rightarrow Y$ là một phụ thuộc hàm, $X, Y \subseteq U$. Những $X \rightarrow Y$ suy diễn logic từ F nếu và chỉ nếu tồn tại $R(U)$ thỏa các phụ thuộc hàm của F mà không thỏa $X \rightarrow Y$.

Chứng minh $F = \{ A \rightarrow B, B \rightarrow C \} \vdash A \rightarrow C$ suy ra từ F . Gọi F^+ là bao đóng của F , tức là tập tất cả các công thức hàm ý suy diễn logic từ F . Nếu $F = F^+$ thì F là tập các công thức hàm ý.

Để kiểm tra tính đúng đắn của một tập các công thức suy diễn logic giữa các công thức hàm ý cần thiết phải tính các tập F^+ từ F . Do vậy phải xác định các công thức thuộc tập F^+ . Tập các quy tắc của tập F^+ là các quy tắc Armstrong và các quy tắc suy diễn logic từ tập F^+ là các quy tắc suy diễn logic từ tập F^+ .

Gọi $R(U)$ là tập các công thức suy diễn logic với $U = \{A_1, \dots, A_n\}$ là tập các thuộc tính. $X, Y, Z, W \subseteq U$. Tập các quy tắc Armstrong bao gồm:

A1 (phản xạ): Nếu $Y \subseteq X$ thì $X \rightarrow Y$

A2 (tính bắc cầu): Nếu $Z \subseteq U$ và $X \rightarrow Y$ thì $XZ \rightarrow YZ$,

trong đó ký hiệu XZ là tập hợp của 2 tập X và Z thay cho ký hiệu $X \cup Z$.

A3 (bổ sung): Nếu $X \rightarrow Y$ và $Y \rightarrow Z$ thì $X \rightarrow Z$

Ví dụ: Cho $AB \rightarrow C, C \rightarrow A$

Cần chứng minh rằng $BC \rightarrow ABC$

Ta chứng minh như sau:

1. $C \rightarrow A$ (giả thiết)

2. $BC \rightarrow BA$ (luật tính bắc cầu của (1) thêm B)

3. $AB \rightarrow C$ (giả thiết)

4. $AB \rightarrow ABC$ (tính bắc cầu (3) thêm AB)

5. $BC \rightarrow ABC$ (bổ sung từ (2) và (4))

Tổ hợp các quy tắc Armstrong suy ra một số luật sau đây:

① Luật hấp thụ: Nếu $X \rightarrow Y$ và $X \rightarrow Z$ thì $X \rightarrow YZ$

② Luật tiêu chuẩn bổ sung: Nếu $X \rightarrow Y$ và $WY \rightarrow Z$ thì $XW \rightarrow Z$

③ Luật t, ch; Nếu $X \rightarrow Y$ và $Z \subseteq Y$ thì $X \rightarrow Z$

5.3. Phép tách một quan hệ

Phép tính mệnh đề liên quan đến $R(u)$ với $U = \{A_1, \dots, A_n\}$ là việc thay thế mệnh đề liên quan đến R bằng các tập mệnh đề $\{R_1, \dots, R_k\}$, trong đó $R_i \subseteq R$, $i = 1, 2, \dots, k$ và

$$R = R_1 \cup R_2 \cup \dots \cup R_k$$

Trong các l-íc R_i kh«ng   i hái ph  i l  ph n bi t.

Môc ti^au c^hĩa ph^háp t^hị ch^hĩ y^hõu l^u lo^li bá c^h c^h đ^h th^h-êng đ^h li^hõu.

Ví dụ 1:

NCC (TenNCC, Dchi, Cluong, Gia)

Ten NCC $\rightarrow$ Dchi

Ten NCC, Cluong $\rightarrow$ Gia

Với lược đồ trên ta có thể thay thế bởi lược đồ

NCC 1 (Ten NCC, Dchi)

NCC2 (Ten NCC, Cluong, Gia)

5.3.1. Phép tách không mất mát thông tin

Phép tách một quan hệ mà không làm mất mát thông tin được ký hiệu là ρ

Định lý: Cho quan hệ $R(u)$ với $\rho = (r_1(u_1), r_2(u_2))$ là phép tách 2 không mất mát thông tin của r nếu $U_1 \cap U_2 \rightarrow U_1 \setminus U_2$ hoặc $U_1 \cap U_2 \rightarrow U_2 \setminus U_1$.

Đồng thời khi $r_1 * r_2 \equiv r$ khi đó phép tách dgl phép tách 2 không mất mát thông tin.

Vậy muốn tách một quan hệ thành m quan hệ không mất mát thông tin thì phải tiến hành $m-1$ phép tách 2 không mất mát thông tin.

5.3.2. Kiểm tra phép tách không mất mát thông tin

Bài toán: Cho lược đồ quan hệ $r(u)$, tập các phụ thuộc hàm f và phép tách quan hệ ρ .

Yêu cầu: Hãy kiểm tra phép tách ρ có mất mát thông tin không.

Phương pháp thực hiện: Từ quan hệ lập một bảng n cột và k hàng. Cột ứng với thuộc tính A_j , hàng ứng với lược đồ r_i với $(i = 1, k)$. Tại hàng i , cột j điền ký hiệu a_{ij} nếu $A_j \in R_i$, Ngược lại điền b_{ij} .

Xét các phụ thuộc hàm: Nếu $X \rightarrow Y \in F$ nếu có giá trị bằng nhau trên thuộc tính X (cột x) thì làm bằng các giá trị của chúng trên thuộc tính Y (cột y). Khi làm bằng nếu trên cột Y có giá trị là A_j thì làm bằng ký hiệu là A_j còn lại làm bằng với một trong các b_{ij} .

Tiếp tục thực hiện như trên cho tới khi không thực hiện được nữa, kể cả việc sử dụng các phụ thuộc hàm đã sử dụng rồi.

Xem xét bảng kết quả nếu có một hàng bao gồm toàn các ký hiệu aj thì kết luận phép tách không mất mát thông tin.

Ví dụ: Xét quan hệ nguyên-cung cấp

NCC(TenNCC, Dchi, Chatluong, Gia)

®-ít t, ch lụm hai quan hệ

NCC1(TenNCC, Dchi) vµ NCC2(TenNCC, Chatluong, Gia)

với c, c phân thức lụm:

TenNCC → Dchi

TenNCC, Chatluong → Gia.

Bảng ban ®-ít thiốt lếp nh- sau:

TenNCC	Dchi	Chatluong	Gia
a ₁	a ₂	b ₁₃	b ₁₄
a ₁	b ₂₂	a ₃	a ₄

, p đồng phân thức lụm TenNCC → Diachi cho hai lụm của bảng. Hai lụm b»ng nhau trên cột TenNCC (®-òu b»ng a₁) nên ở cột Dchi chóng ®-ít lụm b»ng vµ b»ng a₂.

Bảng kết quả lụm:

TenNCC	Dchi	Chatluong	Gia
a ₁	a ₂	b ₁₃	b ₁₄
a ₁	a ₂	a ₃	a ₄

Bảng kết quả cả đồng thời hai cả c, c gì, trê toàn lụ a, do ®-ã phđp t, ch trªn lụ phđp t, ch kh«ng mất mát thông tin.

5.4. Quá trình chuẩn hóa

Trong lý thuyết ban ®-Çu Codd ®-a ra 3 d¹ng chuẩn của quan hệ:

- D¹ng chuẩn thø nhÊt 1NF.
- D¹ng chuẩn thø hai 2NF.
- D¹ng chuẩn thø ba 3NF.

Sau khi Boye nghiên cứu l-ic ở quan hệ với tập các thuộc hàm của nó ra một định chuẩn nữa gọi là chuẩn Boye - Codd về giải quyết định chuẩn BCNF.

Định nghĩa:

Cho l-ic ở quan hệ $R(U)$ trên tập thuộc tính $U = \{A_1, \dots, A_k\}$. X và Y là hai tập thuộc tính khác nhau $X \subseteq U$ và $Y \subseteq U$.

Y là thuộc hàm của X nếu Y thuộc hàm vào X nhưng không thuộc hàm vào bất kỳ một tập hợp con thực sự nào của X .

5.4.1. Định chuẩn một (1NF):

Định nghĩa: Một l-ic ở quan hệ R về giải quyết là định chuẩn một (1NF) nếu và chỉ nếu toàn bộ các miền của các thuộc tính trong R chỉ chứa các giá trị nguyên tử (giá trị không thể chia nhỏ).

Định nghĩa này cho thấy rằng bất kỳ quan hệ chuẩn hóa nào cũng là 1NF.

5.4.2. Định chuẩn hai:

Định nghĩa: L-ic ở quan hệ R là định chuẩn hai nếu và chỉ nếu mỗi thuộc tính khác nhau của R là thuộc hàm của một hoặc nhiều thuộc tính khác nhau.

Ví dụ:

Cho CSDL gồm 2 quan hệ: THI và SINHVIEN

THI	(MONTHI	GIAOVIEN)
	3	A
	4	B
	5	C

SINHVIEN	(MONTHI	MSSV	TEN	TUOI	DIACHI	DIEM)
	3	11	LAN	20	X	8
	3	12	HA	21	Y	6
	4	11	LAN	209	X	7

4	12	HA	21	Y	6
5	11	LAN	20	X	7
5	13	TU	22	Z	2

Trong quan hồ trªn MONTHI lµ kho, cªn quan hồ THI vµ MONTHI, MSSV (m· sè sinh viªn) lµ kho, cªn quan hồ SINHVIEN.

ë quan hồ thø hai MONTHI, MSSV, DIEM x,c ®Þnh kÕt qu¶ thi cªn sinh viªn cßn MSSV, TEN, TUOI, DIACHI x,c ®Þnh c,c ®òi t-ìng lµ sinh viªn.

Trong qu, tr×nh cËp nhÊt vµ l-u tr÷ d÷ liÖu xuÊt hiÖn nh÷ng vÊn ®Ò sau:

- ë quan hồ SINHVIEN viÖc l-u tr÷ c,c th«ng tin vÒ 1 sinh viªn (vÝ dõ nh- LAN) ph¶i lÆp l¹i 3 lÇn ®Þa chØ vµ tuæi. Nh- vËy lµ d- thõa d÷ liÖu.

- Qu, tr×nh cËp nhÊt d÷ liÖu g©y nªn nh÷ng c¶n trë nh-:

+ Khi thay ®æi th«ng tin vÒ mét sinh viªn cÇn ph¶i thay ®æi nhiÖu lÇn, tèn kÐm vÒ thêi gian, dÔ sai sãt. Trong khi ®ã sõa ®æi th«ng tin vÒ mét sinh viªn l¹i kh«ng cã liªn quan g× ®Õn th«ng tin vÒ thi cõ.

§Ó x,c ®Þnh nh÷ng th«ng tin vÒ mét sinh viªn, chØ cÇn MSSV lµ x,c ®Þnh ®-ìc duy nhÊt th«ng tin vÒ hã.

+ Trong quan hồ SINHVIEN chØ chõa nh÷ng sinh viªn ®· thi. Muèn chënh mét sinh viªn vµo quan hồ th× kh«ng ®-ìc v× kh«ng rã sinh viªn ®ã ®· thi hay ch-a; kho, MONTHI, MSSV lµ kh«ng ®Çy ®ñ. §iÖu nµy chØ kh¼c phõc ®-ìc khi lo¹i bá nh÷ng th«ng tin vÒ thi cõ ra khái quan hồ.

+ Gi¶ sõ cÇn huû bá m«n thi thø 5 mµ danh s, ch sinh viªn vÊn gi÷ nguyªn. Khi ®ã ë quan hồ THI xo, bé (5,c), cßn ë quan hồ SINHVIEN nÕu xo, m«n thi thø 5 th× th«ng tin vÒ sinh viªn TU sã bÞ mÊt.

§Ó kh¼c phõc nh÷ng c¶n trë trªn, quan hồ SINHVIEN cã thÓ t, ch th¶nh hai quan hồ SINHVIEN(MSSV, TEN, TUOI, DIACHI) vµ quan hồ THIXONG (MONTHI, MSSV, DIEM).

Nh- vËy CSDL lóc nµy th¶nh ba quan hồ ®· ë d¹ng chuÈn thø 2.

5.4.3. D¹ng chuÈn thø ba:

§Ó t×m hiÖu vÒ d¹ng chuÈn thø ba cªn quan hồ, tr-íc hÕt ph¶i hiÖu vÒ kh,i niÖm phõ thuéc b¼c cÇu.

§Định nghĩa: Cho một lược đồ quan hệ $R(U)$; X là một tập con c, c thuộc tính $X \subseteq U$, A là một thuộc tính thuộc U . A lược đồ giải là một thuộc tính $b \rightarrow c$ của c vào X nếu trên R tồn tại một tập con Y của R sao cho

$$X \rightarrow Y, Y \rightarrow A \text{ nhưng } Y \not\rightarrow X \text{ với } A \notin XY.$$

§Định nghĩa: Lược đồ quan hệ R là đúng chuẩn thứ 3 (3NF) nếu và chỉ là 2NF và mọi thuộc tính không khóa, của R là không thuộc hàm bất kỳ của c vào c hoặc, chính.

Ví dụ:

Cho lược đồ quan hệ R (SAIP) với c, c thuộc hàm $SI \rightarrow P$ và $S \rightarrow A$.

R là không đúng 3NF, không đúng 2NF. Giả sử $X = SI, Y = S$. A là thuộc tính không khóa, và chỉ có một khóa, là SI . Vì $X \rightarrow Y$ và $Y \rightarrow A$, nhưng lại cả $Y \rightarrow X$ tức là $S \rightarrow SI$ không thỏa. Trong trường hợp này $X \rightarrow Y$ và $Y \rightarrow A$ không thỏa trên R mà là không thuộc tính cho. Tức là $X \rightarrow Y, Y \rightarrow A$ suy ra tập con c, c thuộc hàm.

Như vậy A là một thuộc tính bất kỳ của c vào c hoặc, chính SI .

Ví dụ:

Cho lược đồ quan hệ R (CSZ) với c, c thuộc hàm $CS \rightarrow Z, Z \rightarrow C$.

Trong lược đồ này mọi thuộc tính đều là thuộc tính không khóa. Do vậy R là đúng 3NF.

Ví dụ:

Cho lược đồ R (SIDM) và c, c thuộc hàm $SI \rightarrow D, SD \rightarrow M$.

Trong quan hệ này chỉ có một khóa, chính là SI . Như vậy quan hệ này là đúng chuẩn 2NF nhưng không phải là đúng chuẩn 3NF.

5.4.4. Định nghĩa Boye - Codd (BCNF):

Lược đồ quan hệ R với tập c, c thuộc hàm lược đồ giải là đúng chuẩn Boye - Codd nếu $X \rightarrow A$ thỏa trên $R, A \notin X$ thì X là một khóa, của R .

Ví dụ: Cho lược đồ quan hệ R (CSZ) với c, c thuộc hàm $CS \rightarrow Z, Z \rightarrow C$. Ta thấy R không đúng BCNF mà là đúng 3NF vì $Z \rightarrow C$ nhưng không phải là khóa, của R .

Tổ ví dụ này thấy rằng một lược đồ quan hệ chỉ thỏa cả 3NF nhưng không phải BCNF. Do vậy mọi lược đồ quan hệ đúng BCNF là đúng 3NF.

§Định lý: Nếu một lược đồ quan hệ liên quan với tập thuộc hàm F là BCNF thì nó cũng là 3NF.

5.5. Thuật toán chuẩn hóa

5.5.1. Thuật toán 1.

a. Tập bao đóng của một tập thuộc hàm

§Định nghĩa: Cho X là tập con các thuộc hàm của U. Bao đóng của X là tập nhỏ nhất các thuộc hàm thỏa mãn suy ra từ X, ký hiệu X^+ .

+ Thuật toán tập bao đóng của 1 tập thuộc hàm:

Input: Tập U hữu hạn, tập các thuộc hàm F trên U và $X \subseteq U$

Output: X^+

Phương pháp: Tính liên tiếp tập $X_0, X_1, \dots$ theo các bước:

- Bước 1: $X_0 = X$
- Bước ...
- Bước i: $X_{i+1} = X_i \cup A$ sao cho $\exists (Y \rightarrow Z) \in F, A \in Z, Y \subseteq X_i$

Với $X = X_0 \subseteq \dots \subseteq U$, mà U là hữu hạn nên sẽ tồn tại một chø sẽ ổn định mà $X_i = X_{i+1}$ khi đó $X^+ = X_i$.

Ví dụ: Cho $U\{A B C D E G\}$

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

$X = BD$, Tính X^+ ?

Bước 1: Xét $X_0 = BD$

- Kiểm tra thuộc hàm nào thỏa mãn chứa B, D, BD của $D \rightarrow EG$

Ta có $X_1 = X_0 \cup EG = BDEG$

- Kiểm tra thuộc hàm nào thỏa mãn chứa X_1 của $D \rightarrow EG, BE \rightarrow C$

Ta có $X_2 = X_1 \cup C = BCDEG$

- Kiểm tra thuộc hàm nào thỏa mãn chứa X_2 của $C \rightarrow A$

Ta có $X_3 = X_4 = ABCDEG$

Vậy cuối cùng $X^+ = (BD)^+ = ABCDEG = U$.

b. Phân tích tập các thuộc hàm:

§Định nghĩa: Gọi F và G là c,c tập của c,c pho thức hạm. Nãi F và G là t- $\neg$ ng @- $\neg$ ng nũ $F^+ = G^+$, hay cãn gãi là F phĩ G và G phĩ F .

§Ó kiÓm tra ta đĩng thuỄt to,n 1 trªn: lỄy mėt pho thức hạm

$Y \rightarrow Z$ thức F , kiÓm tra xem $Y \rightarrow Z$ cũ thức G^+ kh«ng. TÝnh Y^+ , kiÓm tra xem $Z \subseteq Y^+$ kh«ng? Nũ tãn tªi 1 pho thức hạm $Y \rightarrow Z$ thức F mụ kh«ng pho thức G^+ th× ch¼c ch¼n $F^+ \neq G^+$. Nũ mçi pho thức hạm thức F còng thức G^+ th× mçi pho thức hạm $V \rightarrow W$ thức F^+ còng thức G^+ . §iỜu ng-ĩc lªi ta lụm t- $\neg$ ng tũ.

Vĩi mėt tỄp pho thức hạm F th× F là tỄp pho thức hạm kh«ng d- thõa nũ nã tho¶ m·n @iỜu kiỐn là kh«ng tãn tªi pho thức hạm $X \rightarrow Y \in F$ tho¶ m·n @iỜu kiỐn $(F - \{X \rightarrow Y\})^+ = F^+$.

5.5.2. ThuỄt to,n 2: ThuỄt to,n t×m phĩ tòi thiÓu.

§Định nghĩa 1: Cho 2 tỄp pho thức hạm F và G . Ta nãi r»ng F và G là t- $\neg$ ng @- $\neg$ ng nũ $F^+ = G^+$. (F phĩ G và G phĩ F).

§Định nghĩa 2: Cho F là 1 tỄp pho thức hạm. Khi @ã nãi F là tỄp pho thức hạm kh«ng d- thõa nũ kh«ng tãn tªi $X \rightarrow Y \in F / (F - \{X \rightarrow Y\})^+ = F^+$.

§Định nghĩa 3: cho F là tỄp c,c pho thức hạm, nãi r»ng F là tỄp pho thức hạm tòi thiÓu nũ nã tho¶ m·n 3 tÝnh chỄt:

① Mãi pho thức hạm trong F @Ờu cũ đªng là vỄ ph¶i cũ pho thức hạm chØ cũ 1 thức tÝnh.

$$F = \{X_i \rightarrow A_i / i = 1, m\}$$

② F là tỄp c,c pho thức hạm kh«ng d- thõa

$$\text{Kh«ng tãn tªi mėt tỄp } X \rightarrow Y \in F \text{ mụ } (F \setminus \{X \rightarrow Y\}) \sim F$$

③ VỜ trªi cũ mãi pho thức hạm trong tỄp F kh«ng d- thõa 1 thức tÝnh nũo.

$$\forall X \rightarrow Y \in F, Z \subset X \text{ mụ } (F \setminus \{X \rightarrow Y\} \cup \{Z \rightarrow Y\}) \sim F$$

ThuỄt to,n:

Input: U, F

Output: F' là phĩ tòi thiÓu cũ F

Ph- $\neg$ ng ph,p: TuỢn tũ theo c,c b-ĩc

- B-ĩc 1: Vĩi $X \rightarrow Y \in F$ mụ $Y = A_1, A_2, \dots, A_m$

ta sẽ biến $X \rightarrow Y$ sang : $X \rightarrow A_1$

$X \rightarrow A_2$ X x,c bnh m phô thuc hụm

...

$X \rightarrow A_m$

Kõt qu b-íc 1 cho ra $F^1 \sim F$

• B-íc 2: Tõ b-íc 1 ta cũ $F_1 \sim F$ (Output)

Gi sũ $F^1 = \{X_i \rightarrow A_i, \forall i = 1, m\}$

- B-íc 0: $F^0 = F^1$

...

- B-íc i: $F^i = F^{i-1} - \{X_i \rightarrow A_i\}$ nõu $(F^{i-1} - \{X_i \rightarrow A_i\}) \sim F^{i-1}$

$F^i = F^{i-1}$ nõu ng-íc l*i*

TÝnh õn b-íc m vµ $F^2 = F^m$

$F^2 \sim F^1$

• B-íc 3: $F^2 = \{X_i \rightarrow A_i, i = 1, \overline{k}\}$ (k $\in$ m)

$\forall i = 1, \overline{k}: X_i = A_{i1} \dots A_{ij}$

- B-íc 0: $X^0 = X_i$

...

- B-íc j: $X^j = X^{j-1} - \{A_{ij}\}$ nõu $F - \{X^{j-1} \rightarrow A_i\} \cup \{X^{j-1}\} -$

$-\{A_{ij} \rightarrow A_i\} \sim F$

$X^j = X^{j-1}$ nõu ng-íc l*i*

Cuèi cũng $X_j = X^n$ th× Output cũ $F_3 \sim F_2$

Bæ Ò: $X \rightarrow Y \in F^+$ nõu vµ chØ nõu $Y \subseteq X^+$.

5.5.3. ThuËt to,n 3: KiÓm tra tÝnh t-õng Ò-õng cũa 2 tËp phô thuc hụm

Input: cũ F, G lụ 2 tËp phô thuc hụm

$F = \{X_i \rightarrow Y_i, \forall i = 1, m\}, G = \{X_j \rightarrow Y_j, \forall j = 1, n\}$

Output: KiÓm tra $F \sim G$?

Ph-õng ph,p:

• B-íc 1: Vii mãi $i = 1, m$ kiÓm tra $X_i \rightarrow Y_i$ cũ thuc G^+ ? nõu thuc th×
 $\sim$ tËp con $Y_i \subseteq X_i^+$, suy ra: $F^+ \subseteq G^+$

• B-íc 2: Vii $j = 1, n$ kiÓm tra $X_j \rightarrow Y_j$ cũ thuc $F^+ ?$ t- $\neg$ ng tù $G^+ \subseteq F^+ ?$ nÕu 2 ®iÒu trªn lụ ®óng th× ta cũ $F \sim G$ cũn nÕu mét trong hai ®iÒu trªn kh«ng ®óng th× $F \neq G$.

5.5.4. ThuËt to,n 4: T×m 1 kho, tòi thiÓu

Input: TÛp thuc tÝnh $U = \{A_1, A_2, \dots, A_n\}$

TÛp phô thuc hụm F

Output: t×m k lụ kho, tòi thiÓu

Ph- $\neg$ ng ph,p:

• B-íc 1: §Æt $k^0 = U \dots$

• B-íc i: tÝnh $k^i = k^{i-1} - \{A_i\}$ nÕu $X^{i-1} - \{A_i\} \rightarrow U$

NÕu ng-íc lªi th× $k^i = k^{i-1}$

Cuòi cũng ®Æt $k = k^n$ khi tÝnh ®Õn thuc tÝnh n vù x,c ®õnh k lụ kho, tòi thiÓu.

5.5.5. ThuËt to,n 5: ThuËt to,n kiÓm tra d'ng chuÈn 2:

ĐÇu vùo: L-íc ®ả quan hÖ R vù tÛp phô thuc hụm F

ĐÇu ra: Kh½ng ®õnh R ®t chuÈn 2 hay kh«ng ®t chuÈn 2

B-íc 1: T×m tÛt cũ c,c khĩa cũ R

B-íc 2: Vii mçi khĩa K, T×m bao ®ãng cũ tÛt cũ c,c tÛp con thÛt sù S^+ cũ K.

B-íc 3: NÕu cũ bao ®ãng S^+ chøa thuc tÝnh kh«ng khĩa th× R kh«ng ®t chuÈn 2, ng-íc lªi th× R ®t chuÈn 2.

5.5.6. ThuËt to,n 6: ThuËt to,n kiÓm tra d'ng chuÈn 3:

ĐÇu vùo: Líc ®ả quan hÖ R vù tÛp phô thuc hụm F

ĐÇu ra: Kh½ng ®õnh R ®t chuÈn 3 hay kh«ng ®t chuÈn 3

B-íc 1: T×m tÛt cũ c,c khĩa cũ R

B-íc 2: Tõ F t'ò lÛp mét phô thuc hụm t- $\neg$ ng ®- $\neg$ ng F1tt, cũ vÕ phỉ 1 thuc tÝnh.

B-íc 3: NÕu mĩ phô thuc hụm $X \rightarrow A \in F1tt$ vii $A \notin X$ ®Òu cũ X lụ si'u khĩa hoÆc A lụ thuc tÝnh khĩa th× R ®t chuÈn 3, ng-íc lªi R kh«ng ®t chuÈn 3.

5.5.7. Thuật toán 7: Thuật toán kiểm tra dạng chuẩn BC

Đầu vào: Lược đồ quan hệ U và tập phụ thuộc hàm F

Đầu ra: Khẳng định U đạt chuẩn BC hay không

Bước 1: Tìm tất cả các khóa của U

Bước 2: Từ F tạo một PTHF1tt có vẻ phải 1 thuộc tính

Bước 3: Nếu mọi PTH $X \rightarrow A \in F1tt$ với $A \notin X$ đều có X

là siêu khóa thì U đạt chuẩn BC, ngược lại U không đạt chuẩn BC

5.5.8. Thuật toán 8 Thuật toán kiểm tra dạng chuẩn của một lược đồ quan hệ

Đầu vào: Lược đồ quan hệ Q và tập phụ thuộc hàm F

Đầu ra: Khẳng định Q đạt chuẩn gì?

Bước 1: Tìm tất cả các khóa của Q

Bước 2: Kiểm tra dạng chuẩn BC, nếu đúng thì Q đạt chuẩn BC, kết thúc thuật toán.

Ngược lại sang bước 3.

Bước 3: Kiểm tra dạng chuẩn 3, nếu đúng thì Q đạt chuẩn 3, kết thúc thuật toán.

Ngược lại sang bước 4.

Bước 4: Kiểm tra dạng chuẩn 2, nếu đúng thì Q đạt chuẩn 2, kết thúc thuật toán.

Ngược lại kết luận Q đạt chuẩn 1NF.

5.5.9: Thuật toán 9: Chuẩn hóa lược đồ quan hệ 3(3NF)

Input: Quan hệ r(U)

Tập các phụ thuộc hàm F

Output: Tập $\rho = r_1(U_1), \dots, r_m(U_m)$ với $r_i(U_i)$ là 3NF

Phân hoạch ρ :

- Bước 1: Dùng thuật toán 2 để tìm phân hoạch tối thiểu của F (F'), dùng F' làm

Input của bước tiếp theo

- Bước 2: Trong bước này cần 3 bước con:

① Nếu tồn tại thuộc tính $A \in U$ mà không xuất hiện phụ thuộc hàm nào của F (không có trong vế trái và vế phải) thì ta thêm tập tất cả các A vào ρ để $U_0 = \{A\}$, nếu không thêm A thì $U_0 = \emptyset$

② Nếu tồn tại một phôi thức hàm trong F' : $X \rightarrow A \in F'$ mà $XA = U - U_0$ thì $\rho = (r_0(U_0), r_1(U - U_0)) \rightarrow$ thuật toán kết thúc. Nếu ngược lại chuyển sang bước tiếp theo.

③ Với mỗi $X_i \rightarrow A_{i_1}, X_i \rightarrow A_{i_2}, \dots, X_i \rightarrow A_{i_l}$

phải xây dựng tập $U_i = X_i A_{i_1}, \dots, X_i A_{i_l}$ và cuối cùng cần:

$$\rho = (r_0(U_0), r_1(U_i))$$

• Bước 3: Dùng thuật toán 4 để tìm kho, kết quả thi

Kết quả:

$$\rho = \rho_c - r_0(U_0) \cup r_0(k) \text{ khi } U_0 = \emptyset \text{ thì}$$

$$\rho = \rho_c \cup r_0(k)$$

TÀI LIỆU THAM KHẢO

- [1]. Nhập môn cơ sở dữ liệu quan hệ - Lê Tiến Vương - NXB Khoa học Kỹ thuật – 2003.
- [2]. Elmasri – Navathe, Fundamentals_of_Database_Systems, 2010.
- [3]. Nguyên lý các hệ cơ sở dữ liệu - Nguyễn Kim Anh - NXB Khoa học Kỹ thuật – 2005.
- [4]. Cơ sở dữ liệu - Đỗ Trung Tuấn – 2000.
- [5]. Lê Xuân Huy, Lê Hoài Bắc, Bài tập cơ sở dữ liệu, NXB Thống kê, 2003
- [6] Codd E.F., “Relational model of data for large share banks”, Comm, ACM. 13,6 (June 1970).
- [7] Date C.J., “ An introduction to data base systems”, Second edition 1977.
- [8] Codd, E.F., “ Data models in data base management”, ACM SIGMOD record, 11,2 (Feb, 1981).