

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



CƠ SỞ DỮ LIỆU

(Dùng cho sinh viên hệ đào tạo đại học từ xa)

Lưu hành nội bộ

HÀ NỘI - 2006

CƠ SỞ DỮ LIỆU

Biên soạn : Ts. PHẠM THẾ QUẾ

LỜI NÓI ĐẦU

Tài liệu “Cơ sở dữ liệu” là sách hướng dẫn học tập dùng cho sinh viên hệ đào tạo từ xa ngành công nghệ thông tin và ngành kỹ thuật viễn thông.

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 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.

Chương IV 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 V đề cập đến các ngôn ngữ con thao tác dữ liệu: ngôn ngữ con dữ liệu dựa trên đại số quan hệ, ngôn ngữ có cấu trúc SQL truy vấn dữ liệu.

Chương VI đề cập tới một số phương pháp tối ưu hoá câu hỏi, sao cho tăng tốc độ thực hiện và tối ưu hoá bộ nhớ.

Cuối cùng là chương VII, nội dung của chương này trình bày sự cần thiết phải bảo vệ an toàn cơ sở dữ liệu. “An toàn” dữ liệu nghĩa là các hệ cơ sở dữ liệu cần phải được bảo vệ chống truy nhập trái phép sửa đổi hay phá hoại dữ liệu. Các hệ thống dữ liệu cần thiết phải bảo đảm tính toàn vẹn và an toàn dữ liệu.

Tài liệu “Cơ sở dữ liệu” không chỉ đề cập đến những vấn đề cơ sở lý thuyết mà còn trình bày một số kỹ năng cần thiết để thiết kế và cài đặt các hệ cơ sở dữ liệu cụ thể. Hy vọng sẽ có ích cho sinh viên và những người muốn xây dựng các hệ thống tin học ứng dụng phục vụ cho sản xuất, quản lý trong các doanh nghiệp. Có thể còn nhiều thiếu sót trong biên soạn, tôi vẫn mạnh dạn giới thiệu tài liệu này và mong nhận được sự góp ý của bạn đọc.

Tác giả

1 KHÁI NIỆM CƠ BẢN VỀ CÁC HỆ CƠ SỞ DỮ LIỆU

Trong chương này trình bày những khái niệm cơ bản về các hệ cơ sở dữ liệu do E.F Codd đề xuất. Những khái niệm này bao gồm mục tiêu của một hệ cơ sở dữ liệu. Sự cần thiết phải tổ chức dữ liệu dưới dạng cơ sở dữ liệu. Tính độc lập của dữ liệu thể hiện mô hình hình kiến trúc 3 mức. Vì vậy có thể nói cơ sở dữ liệu phản ánh tính trung thực, khách quan của thế giới dữ liệu. Không dư thừa thông tin và cũng không thiếu thông tin. Nội dung của chương bao gồm các phần:

- Cơ sở dữ liệu là gì
- Sự cần thiết của các hệ cơ sở dữ liệu
- Mô hình kiến trúc 3 mức cơ sở dữ liệu
- Mục tiêu của các hệ cơ sở dữ liệu
- Hệ quản trị CSDL & người quản trị CSDL
- Tổ chức lưu trữ dữ liệu
- Các mô hình truy xuất

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 tồn thất thông tin và khi biểu diễn dữ liệu là duy nhất. Dữ liệu được lưu trữ trong bộ nhớ của máy tính không những phải tính đến yếu tố về tối ưu không gian lưu trữ, mà phải đảm bảo tính khách quan, trung thực của dữ liệu hiện thực. Nghĩa là phải đảm bảo tính nhất quán của dữ liệu và giữ được sự toàn vẹn của dữ liệu.

1.2 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 "Cuộc 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. Cuộc 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 Mô hình kiến trúc tổng quát cơ sở dữ liệu 3 mức

Mô hình kiến trúc 3 mức của hệ CSDL gồm: *Mức trong, mức mô hình dữ liệu (Mức quan niệm) và mức ngoài.* Giữa các mức tồn tại các ánh xạ quan niệm trong và ánh xạ quan niệm ngoài. Trung tâm của hệ thống là mức quan niệm, tức là mức mô hình dữ liệu. Ngoài ra còn có khái niệm người sử dụng, hệ quản trị CSDL và người quản trị CSDL.

Người sử dụng: Là những người tại thiết bị đầu cuối truy nhập vào các hệ CSDL theo chế độ trực tuyến hay tương tác bằng các chương trình ứng dụng hay bằng các ngôn ngữ con dữ liệu. Thường là các chuyên viên kỹ thuật tin học, có trình độ thành thạo biết lập trình và biết sử dụng ngôn ngữ con thao tác dữ liệu (SQL Server, Oracle...). Người sử dụng có thể truy nhập toàn bộ hay một phần CSDL mà họ quan tâm, phụ thuộc vào quyền truy nhập của họ. Cách nhìn CSDL của người sử dụng nói chung là triu tượng. Họ nhìn CSDL bằng mô hình ngoài, gọi là *mô hình con dữ liệu*. Chẳng hạn người sử dụng là một nhân viên của phòng kế toán tài chính, chỉ nhìn thấy tập các xuất hiện kiểu bản ghi ngoài về doanh thu, sản lượng trong tháng, không thể nhìn thấy các xuất hiện kiểu bản ghi lưu trữ về các chỉ tiêu kỹ thuật của đường thông, mạng lưới...

Mô hình ngoài: Mô hình ngoài là nội dung thông tin của CSDL dưới cách nhìn của người sử dụng. Là nội dung thông tin của một phần dữ liệu tác nghiệp được một người hoặc một nhóm người sử dụng quan tâm. Nói cách khác, mô hình ngoài mô tả cách nhìn dữ liệu của người sử dụng và mỗi người sử dụng có cách nhìn dữ liệu khác nhau. Nhiều mô hình ngoài khác nhau có thể cùng tồn tại trong một hệ CSD, nghĩa là có nhiều người sử dụng chia sẻ chung cùng một cơ sở dữ liệu. Hơn nữa, có thể mô hình ngoài quan hệ, mô hình ngoài phân cấp hay mô hình ngoài kiểu mạng cũng có thể tồn tại trong một cơ sở dữ liệu. Sơ đồ ngoài không làm “hiện “ mà được nhúng vào trong logic một đơn tác có liên quan.

- Mô hình ngoài gồm nhiều xuất hiện kiểu bản ghi ngoài, nghĩa là mỗi một người sử dụng có một sơ đồ dữ liệu riêng, một khung nhìn dữ liệu riêng. Bản ghi ngoài của người sử dụng có thể khác với bản ghi lưu trữ và bản ghi quan niệm.

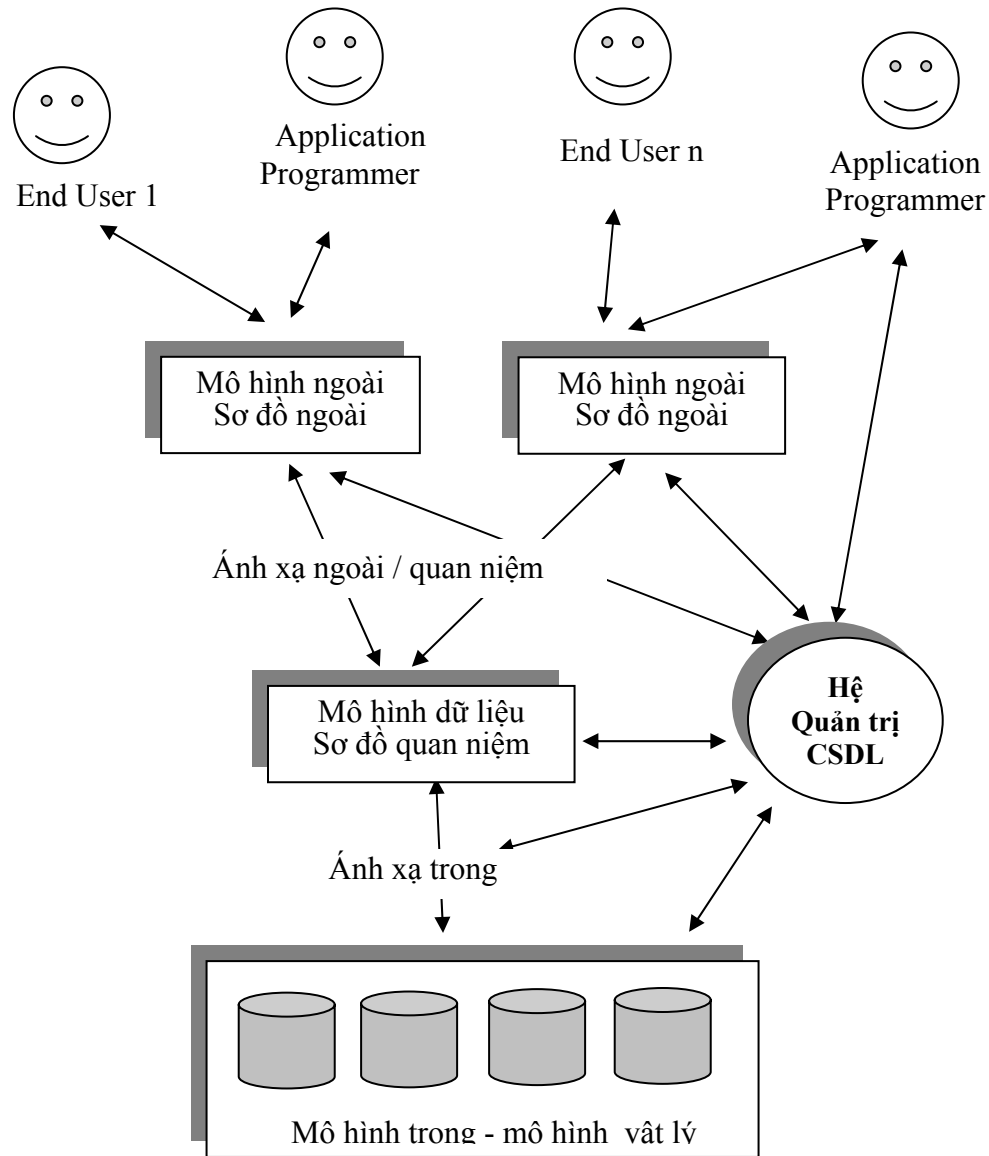
- Mô hình ngoài được xác định bởi một sơ đồ ngoài bao gồm các mô tả về kiểu bản ghi ngoài như tên các trường, kiểu dữ liệu các trường, độ rộng của trường....
- Ngôn ngữ con dữ liệu của người sử dụng thao tác trên các bản ghi ngoài.
- Người sử dụng khác nhau có khung nhìn dữ liệu khác nhau.
- Người sử dụng đầu cuối có thể là các ứng dụng hay thao tác trực tiếp bằng ngôn ngữ thao tác, truy vấn dữ liệu.

Mô hình dữ liệu (mô hình quan niệm): Mô hình quan niệm là cách nhìn dữ liệu một cách tổng quát của người sử dụng. Nghĩa là có rất nhiều cách nhìn dữ liệu ở mô hình ngoài, nhưng chỉ có duy nhất một cách nhìn dữ liệu ở mức quan niệm. Biểu diễn toàn bộ thông tin trong CSDL là duy nhất.

- Mô hình dữ liệu gồm nhiều xuất hiện của nhiều kiểu bản ghi dữ liệu. Ví dụ kiểu xuất hiện bản ghi về nhân sự, kiểu xuất hiện bản ghi về doanh thu, sản lượng, kiểu xuất hiện bản ghi về cuộc đàm thoại...
- Mô hình dữ liệu được xác định bởi một sơ đồ dữ liệu mô tả của nhiều kiểu thực thể, chẳng hạn như mô tả thực thể tuyến cấp, các loại cấp, thầy giáo, học sinh... Sơ đồ dữ liệu bao gồm các định nghĩa về các kiểu bản ghi, đó là các ràng buộc cho quyền và tính toàn vẹn thích hợp. Những ràng buộc này chính là các tính chất của dữ liệu, tính liên kết các thuộc tính cùng một kiểu dữ liệu. Các định nghĩa này không bao hàm về cấu trúc lưu trữ, cũng như về chiến lược truy nhập, chúng chỉ là các định nghĩa về nội dung thông tin, về tính độc lập của dữ liệu trong mô hình quan niệm.
- Sơ đồ quan niệm luôn luôn ổn định, nghĩa là nếu mô tả thêm một kiểu thực thể đặc biệt sát nhập vào sơ đồ dữ liệu, không được làm thay đổi sơ đồ dữ liệu cũ. Nếu sơ đồ dữ liệu không ổn định thì các ứng dụng và mô hình ngoài cũng không ổn định. Sơ đồ dữ liệu chỉ được thay đổi khi có sự điều chỉnh trong thế giới thực, đòi hỏi điều chỉnh lại định nghĩa sao cho nó phản ánh thế giới hiện thực khách quan hơn, chân lý hơn.
- Thiết kế mô hình dữ liệu là giai đoạn quan trọng và quyết định trong việc thiết kế và cài đặt các hệ cơ sở dữ liệu. Quá trình thiết kế không phụ thuộc quá nhiều vào cấu trúc lưu trữ vật lý và chiến lược truy nhập của dữ liệu. Như vậy việc thiết kế sơ đồ dữ liệu phải được tiến hành độc lập với việc thiết kế sơ đồ trong và các sơ đồ ngoài liên kết, vì nếu không việc thiết kế sẽ không ổn định và thường xuyên phải xem xét lại tác động thường xuyên đến nhiều thành phần khác của hệ thống.
- Với cách thiết kế truyền thống hiện nay, người thiết kế chỉ cung cấp một số sơ đồ trong và một tập các sơ đồ ngoài và họ coi đó là sơ đồ dữ liệu, là mô hình dữ liệu. Vì vậy tính không ổn định hệ thống, tính không phù hợp với các ứng dụng này sinh sau một thời gian hoạt động. Mâu thuẫn và dị thường thông tin sẽ xảy ra. Vi phạm tính toàn vẹn của dữ liệu.
- Ngoài các định nghĩa về xuất hiện nhiều kiểu bản ghi quan niệm, sơ đồ dữ liệu còn chứa các định nghĩa về quyền truy nhập của người sử dụng, các thủ tục kiểm tra tính đúng đắn của dữ liệu nhằm bảo đảm tính toàn vẹn của CSDL. Các luồng lưu chuyển thông tin, quy định cách thức sử dụng thông tin..

Chương I: Khái niệm cơ bản về các hệ cơ sở dữ liệu

Như vậy mô hình dữ liệu là cách nhìn toàn bộ nội dung thông tin của CSDL, sơ đồ quan niệm là định nghĩa của cách nhìn ấy. Là bước đi đầu tiên, quan trọng trong việc thiết kế và cài đặt các hệ cơ sở dữ liệu.



Hình 1.1 Kiến trúc hệ cơ sở dữ liệu

Mô hình trong: Mô hình trong là mô hình lưu trữ vật lý dữ liệu. Chỉ có duy nhất một và chỉ một cách biểu diễn CSDL dưới dạng lưu trữ vật lý. Mô hình trong là cách biểu diễn cơ sở dữ liệu trừu tượng ở mức thấp nhất.

- Mô hình trong gồm nhiều xuất hiện của nhiều kiểu bản ghi lưu trữ được xác định bởi một sơ đồ trong. Thông tin biểu diễn trong mô hình trong là duy nhất.
- Sơ đồ trong bao gồm các định nghĩa mô hình trong. Không chỉ xác định các kiểu khác nhau của bản ghi lưu trữ mà còn xác định rõ sự tồn tại của các chỉ dẫn, cách sắp xếp các

bản ghi theo thứ tự nào...Nó xác định dữ liệu lưu trữ và truy nhập như thế nào thông qua các đường dẫn truy nhập tới dữ liệu

Ánh xạ quan niệm trong được xác định giữa mô hình trong và mô hình dữ liệu nhằm bảo đảm tính độc lập của dữ liệu. Nếu cấu trúc lưu trữ của CSDL thay đổi, nghĩa là thay đổi định nghĩa về cấu trúc lưu trữ dữ liệu thì ánh xạ này phải cũng phải thay đổi tương ứng sao cho sơ đồ quan niệm (mô hình dữ liệu) không thay đổi.

Ánh xạ quan niệm-ngoài: Là ánh xạ được xác định tương ứng một-một giữa mô hình ngoài của người sử dụng với mô hình dữ liệu.

1.4 Mục tiêu của các hệ cơ sở dữ liệu

Người sử dụng khi thao tác trên các cơ sở dữ liệu không được làm thay đổi cấu trúc lưu trữ dữ liệu và chiến lược truy nhập tới các hệ cơ sở dữ liệu. Dữ liệu chỉ được biểu diễn, mô tả một cách duy nhất. Cấu trúc lưu trữ dữ liệu và các hệ chương trình ứng dụng trên các hệ CSDL hoàn toàn độc lập với nhau, không phụ thuộc lẫn nhau. Vì vậy bảo đảm tính độc lập dữ liệu là mục tiêu quan trọng của các hệ cơ sở dữ liệu. Có thể định nghĩa tính độc lập dữ liệu là “*Tính bất biến của các hệ ứng dụng đối với sự thay đổi trong cấu trúc lưu trữ và chiến lược truy nhập dữ liệu*”.

Khi thay đổi cấu trúc lưu trữ và các chiến lược truy nhập dữ liệu không kéo theo thay đổi nội dung của các chương trình ứng dụng và ngược lại, khi các chương trình thay đổi cũng không làm ảnh hưởng đến cấu trúc lưu trữ và chiến lược truy nhập của dữ liệu. Tính độc lập của dữ liệu bảo đảm cho việc biểu diễn nội dung thông tin cho các thực thể là duy nhất và bảo đảm tính toàn vẹn và nhất quán dữ liệu trong lưu trữ.

Trong các mô hình dữ liệu như mô hình dữ liệu quan hệ, mô hình dữ liệu phân cấp và mô hình dữ liệu mạng... thì mô hình dữ liệu quan hệ được sử dụng phổ biến và được nhiều người quan tâm nghiên cứu. Vì nó có nhiều ưu điểm cơ bản hơn so với các mô hình dữ liệu khác. Dữ liệu trong mô hình quan hệ được biểu diễn chặt chẽ, logic. Mô tả thế giới hiện thực một cách chính xác, khách quan, phù hợp với cách nhìn và sử dụng của người sử dụng. Vì vậy tính độc lập dữ liệu trong các hệ cơ sở dữ liệu quan hệ cao.

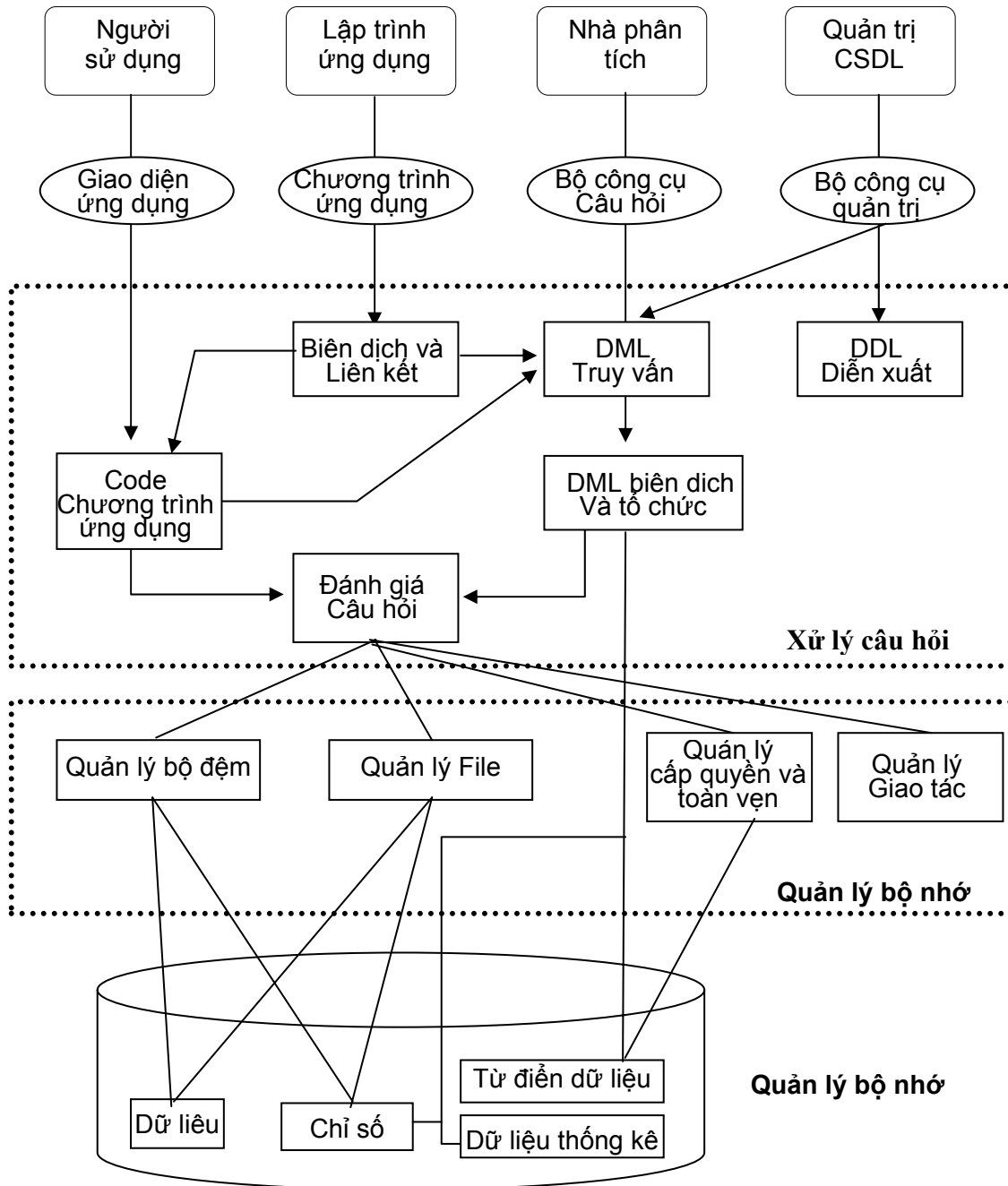
Trong kiến trúc hệ cơ sở dữ liệu (hình 1.1) tính độc lập dữ liệu được thể hiện:

- Có rất nhiều cách nhìn dữ liệu ở mô hình ngoài, người sử dụng khác nhau có cách nhìn dữ liệu khác nhau và các hệ ứng dụng khác nhau có những cách nhìn dữ liệu cũng khác nhau, nhưng chỉ có duy nhất một cách nhìn dữ liệu ở mức quan niệm, biểu diễn toàn bộ nội dung thông tin trong CSDL đó là cách nhìn dữ liệu tổng quát của người sử dụng. Và cũng chỉ có duy nhất một và chỉ một cách biểu diễn CSDL dưới dạng lưu trữ vật lý.
- *Ánh xạ trong* xác định giữa mô hình trong và mô hình dữ liệu, nhằm bảo đảm được tính độc lập của dữ liệu, nghĩa là nếu cấu trúc lưu trữ của CSDL thay đổi, tức là thay đổi định nghĩa về cấu trúc lưu trữ dữ liệu thì ánh xạ này phải cũng phải thay đổi tương ứng sao cho sơ đồ quan niệm (mô hình dữ liệu) không được thay đổi. Tương tự *ánh xạ ngoài* xác định tương ứng giữa một mô hình của người sử dụng nào đó với mô hình dữ liệu. Nó chuyển đổi dạng biểu diễn dữ liệu lưu trữ sang dạng biểu diễn dữ liệu mà các ứng dụng cần đến.

Chương I: Khái niệm cơ bản về các hệ cơ sở dữ liệu

Các hệ ứng dụng khác nhau có nhiều khung nhìn khác nhau với dữ liệu như nhau. Các hệ ứng dụng độc lập với cấu trúc lưu trữ và chiến lược truy nhập. Giữa chúng không có sự ràng buộc lẫn với nhau. Điều này có nghĩa là các hệ ứng dụng hoàn toàn độc lập với bất cứ một cấu trúc lưu trữ và chiến lược truy nhập dữ liệu cụ thể nào. Ngược lại cấu trúc lưu trữ và chiến lược truy nhập dữ liệu không phụ thuộc vào bất kỳ hệ ứng dụng cụ thể nào.

Người quản trị CSDL phải có khả năng đáp ứng với mọi sự thay đổi về cấu trúc lưu trữ và các chiến lược truy nhập mà không cần biết tới có những hệ ứng dụng nào trên CSDL.



Hình 1.2 Sơ đồ kiến trúc hệ thống cơ sở dữ liệu

1.5 Hệ quản trị CSDL & người quản trị CSDL

Hệ quản trị CSDL DBMS (DataBase Management System) là hệ thống phần mềm điều khiển toàn bộ các chiến lược truy nhập và cấu trúc lưu trữ cơ sở dữ liệu. Các chức năng chủ yếu của một hệ quản trị cơ sở dữ liệu: QTCSDL

- Mô tả dữ liệu tạo lập và duy trì sự tồn tại của CSDL
- Cho phép truy xuất vào CSDL theo thẩm quyền đã được cấp
- Cập nhật, chèn thêm, loại bỏ hay sửa đổi dữ liệu mức tệp.
- Đảm bảo an toàn, bảo mật dữ liệu và tính toàn vẹn dữ liệu.
- Tạo cấu trúc dữ liệu tương ứng với mô hình dữ liệu.
- Đảm bảo tính độc lập dữ liệu. Tức là cấu trúc lưu trữ dữ liệu độc lập với các trình ứng dụng dữ liệu.
- Tạo mối liên kết giữa các thực thể.
- Cung cấp các phương tiện sao lưu, phục hồi (backup, recovery).
- Điều khiển tương tranh

Các bước thực hiện của hệ quản trị CSDL có thể tóm tắt như sau:

- Người sử dụng đưa ra yêu cầu truy nhập bằng ngôn ngữ con dữ liệu.
- DBMS sẽ tiếp nhận và phân tích yêu cầu.
- DBMS xem xét sơ đồ ngoài, ánh xạ ngoài, sơ đồ quan niệm, ánh xạ trong,...
- Thực hiện các thao tác trên CSDL lưu trữ.

Các thành phần của một hệ QTCSDL: Một hệ QTCSDL thông thường có các thành phần chính như sau:

- Ngôn ngữ định nghĩa dữ liệu (Data Definition Language).
- Ngôn ngữ thao tác dữ liệu (Data Manipulation Language).
- Ngôn ngữ truy vấn dữ liệu (Query Language).
- Bộ báo cáo (Report Write).
- Bộ đồ họa (Graphics Generator).
- Bộ giao tiếp ngôn ngữ chủ (Host Language Interface).
- Ngôn ngữ thủ tục (Procedure Language)
- Từ điển dữ liệu.
- Bộ phát sinh ứng dụng.

Người quản trị CSDL là một người hay một nhóm người có khả năng chuyên môn cao về công nghệ tin học, có trách nhiệm quản lý và điều khiển toàn bộ hoạt động của các hệ CSDL. Vì vậy người quản trị CSDL cần phải đặt ra các hình thức, quy định cho người sử dụng nhằm ngăn chặn việc truy nhập trái phép vào các hệ CSDL. Người quản trị CSDL có thể cho phép người sử dụng những quyền truy nhập như chỉ được phép đọc, đọc một phần, có thể sửa, bổ sung một phần...

Người quản trị CSDL có một số nhiệm vụ chính:

- Xác định thực thể và nội dung thông tin cần lưu trữ. Xác định sơ đồ quan niệm đáp ứng yêu cầu truy nhập của người sử dụng.

- Quyết định cấu trúc lưu trữ & chiến lược truy nhập: Người quản trị CSDL phải xác định cách thức biểu diễn dữ liệu như mô tả cấu trúc lưu trữ trong, mô tả cấu trúc lưu trữ vật lý. Xác định mô hình dữ liệu, định nghĩa ánh xạ giữa cấu trúc lưu trữ và sơ đồ ngoài..Thực hiện các chiến lược lưu trữ, quản lý hệ thống.
- Người quản trị CSDL phải tạo môi trường giao tiếp giữa người sử dụng với các hệ CSDL, vì sơ đồ ngoài cho người sử dụng là cách nhìn dữ liệu tương ứng với ngôn ngữ con dữ liệu thích hợp, nên người quản trị CSDL phải cung cấp sơ đồ quan niệm, các ánh xạ, và cấu trúc lưu trữ. Kiểm soát thẩm quyền truy nhập của người sử dụng và bảo đảm quyền truy nhập của họ.
- Duy trì các tiêu chuẩn thống nhất về các thủ tục lưu trữ và cấu trúc lưu trữ, biểu diễn thông tin và các chiến lược truy nhập. Kiểm soát và kiểm tra tính đúng đắn của dữ liệu. áp dụng các biện pháp an toàn, an ninh dữ liệu.
- Xác định chiến lược lưu trữ, sao chép, phục hồi...trong các trường hợp hư hỏng do sai sót, hoặc trục trặc kỹ thuật.

1.6 Ràng buộc dữ liệu

Giữa các thực thể dữ liệu tồn tại các mối quan hệ, ràng buộc lẫn nhau. Các ràng buộc này chính là tập các quy tắc, quy định yêu cầu dữ liệu trong cơ sở dữ liệu phải thoả mãn. Mục đích xây dựng các ràng buộc dữ liệu là nhằm bảo đảm tính độc lập và tính toàn vẹn dữ liệu. Dữ liệu lưu trữ trong cơ sở dữ liệu luôn luôn hiện thực khách quan, không thừa thiếu thông tin, không mâu thuẫn thông tin. Các hệ cơ sở dữ liệu cần phải có các cơ chế cho việc mô tả các ràng buộc và quản lý các ràng buộc đã được mô tả.

Có rất nhiều loại ràng buộc. Ràng buộc về kiểu, ràng buộc giải tích, ràng buộc logic... đó là các khái niệm về phụ thuộc hàm, phụ thuộc đa trị, phụ thuộc kết nối.

Ràng buộc kiểu: Loại ràng buộc thấp nhất, mô tả tính chất của các thuộc tính khi tạo lập CSDL Ngoài tên của thuộc tính, thuộc tính đó kiểu gì, chuỗi ký tự, kiểu số, kiểu ngày, kiểu logic...và độ dài là bao nhiêu. Ví dụ thuộc tính “Số điện thoại” là kiểu chuỗi ký tự đúng bằng 7 ký tự trong xâu. Hệ thống sẽ không chấp nhận , nếu nhập vào CSDL một số điện thoại kiểu số hoặc kiểu xâu nhưng chưa đủ hoặc vượt quá 7 ký tự . Phản ứng của hệ thống hoặc là đưa ra thông báo “ Dữ liệu không hợp lệ”, hoặc cắt đi những ký tự thừa.

Ràng buộc giải tích: Là những ràng buộc giữa các thuộc tính được biểu diễn bằng các biểu thức toán học. Ví dụ khi nhập “số lượng” và “đơn giá” của một mặt hàng, hệ thống sẽ tự động tính giá trị của thuộc tính “thành tiền” theo công thức “số lượng” x “đơn giá” = “thành tiền”. Hoặc đánh giá năng lực học tập của một em học sinh, khi nhập giá trị “điểm trung bình” của từng em vào hệ thống , hệ thống tự động đánh giá em đó có năng lực học tập là “kém”, “trung bình”, “khá” hay “giỏi”.

Ràng buộc logic: Mối quan hệ giữa các thuộc tính với nhau không phải là các ràng buộc giải tích, được gọi là phụ thuộc hàm. Thuộc tính Y phụ thuộc hàm vào thuộc tính X , nghĩa là mỗi một giá trị của X xác định giá trị của Y. Ví dụ nếu giá trị của số điện thoại có thể xác

có thể xác định các thông tin về thuê bao có số điện thoại đó. Những ràng buộc logic có thể là ánh xạ một – một hoặc một – nhiều.

1.7 Các mô hình truy xuất dữ liệu

Truy nhập và khai thác các hệ cơ sở dữ liệu trở thành phương thức phổ biến trong các ứng dụng của hệ thống tin học, đặc biệt trên các mạng Internet/Intranet. Chuyển tải thông tin từ các hệ cơ sở dữ liệu lên mạng dưới dạng ngôn ngữ đánh dấu siêu văn bản HTML (HyperText Markup Language) hoặc bằng các ngôn ngữ khác nhằm cung cấp cho dịch vụ World Wide Web đa dạng và phong phú thêm. Truy nhập và khai thác các hệ cơ sở dữ liệu đòi hỏi phải nghiên cứu, giải quyết một số vấn đề về kỹ thuật lập trình mạng, lập trình cơ sở dữ liệu động trên các môi trường Internet và Intranet.

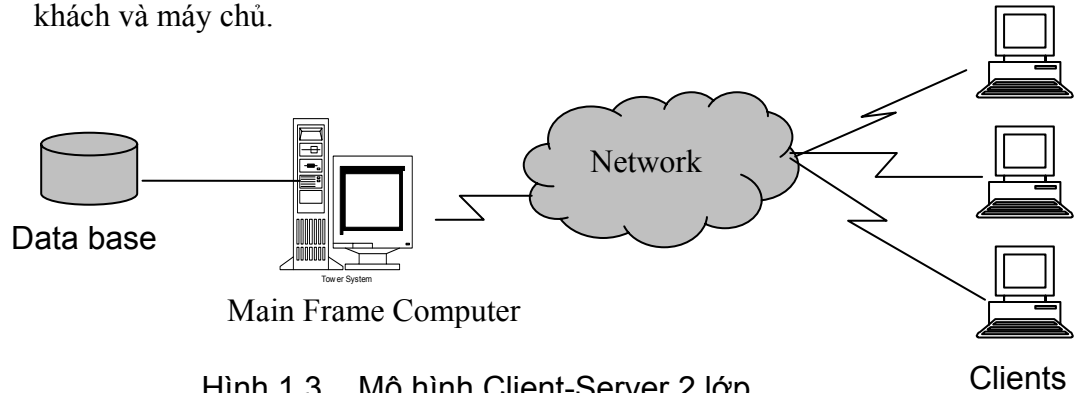
1.7.1 Mô hình cơ sở dữ liệu Client Server

Một trong những mục tiêu khi kết nối các máy tính thành mạng là chia sẻ các tài nguyên thông tin. Một máy chủ cung cấp các loại dịch vụ cho nhiều máy khách thông qua môi trường mạng. Máy chủ và máy khách đều tham gia quá trình xử lý, vì vậy mô hình Client-Server 2 lớp trở nên phổ biến. Các máy khách (Clients) chia sẻ gánh nặng xử lý của máy chủ trung tâm. Khi máy khách thực hiện các ứng dụng, nó gửi yêu cầu về máy chủ được kết nối với cơ sở dữ liệu, máy chủ xử lý và gửi trả lại kết quả về máy khách.

- Trên các máy chủ, thường được cài đặt các hệ cơ sở dữ liệu bao gồm các bảng biểu, các thủ tục lưu trữ... và điều khiển các tiến trình sau:
 - √ Quản lý dữ liệu.
 - √ Bảo mật dữ liệu.
 - √ Thực hiện truy vấn, ràng buộc và các thủ tục lưu trữ.
 - √ Điều khiển lỗi.
- Các tiến trình được thực hiện trên máy khách.
 - √ Tạo giao diện người sử dụng (User Interface).
- Tương tác cơ sở dữ liệu (Database Interaction)
 - √ Cập nhật dữ liệu: thêm, sửa và xoá dữ liệu.
 - √ Điều khiển lỗi.
- Tuy nhiên mô hình Client/Server vẫn còn nhiều bất cập:
 - √ Mô hình Client/Server 2 lớp có hiệu quả cao với các ứng dụng nhỏ và số lượng người sử dụng hạn chế. Khi nhiều máy khách (Clients) kết nối truy nhập vào cơ sở dữ liệu thì năng lực quản lý & xử lý của máy chủ (Server) sẽ bị giảm xuống, tốc độ xử lý chậm.
 - √ Nhiều kết nối dữ liệu phải được duy trì.
 - √ Mã nguồn không có khả năng dùng sử dụng lại. Một ứng dụng tồn tại trong nhiều khối mã nguồn khác nhau được cài đặt trên máy khách. Vì vậy khi có sự

thay đổi mã nguồn người ta cần phải cài đặt lại trên tất cả máy khách, điều này rất khó có thể thực hiện được.

- ✓ Không có lớp trung gian điều khiển sự bảo mật và các giao dịch giữa máy khách và máy chủ.



Hình 1.3 Mô hình Client-Server 2 lớp

1. Trình duyệt Browser gửi yêu cầu cho Web Server.
2. Web Server trả kết quả về cho trình duyệt

1.7.2 Mô hình Client/Server nhiều lớp

Trong mô hình Client/Server 3 lớp (hay nhiều lớp), quá trình xử lý được phân tán trên 3 lớp khác nhau với các chức năng riêng biệt. Vì vậy mô hình này rất thích hợp cho việc tổ chức hệ thống thông tin trên mạng Internet và mạng Intranet hay trong các mạng cục bộ. Phát triển mô hình 3 lớp sẽ khắc phục được một số hạn chế của mô hình 2 lớp. Các hệ cơ sở dữ liệu được cài đặt trên các máy chủ Web Server và có thể được truy nhập không hạn chế các ứng dụng và số lượng người dùng.

- **Lớp khách (Clients):** Chức năng của lớp này là cung cấp dịch vụ trình bày (Presentation Services). Thực hiện việc giao tiếp giữa người sử dụng với lớp giao dịch thông qua trình duyệt Browser hay trình ứng dụng để thao tác và xử lý dữ liệu. Thông thường giao diện người sử dụng được chứa trong các File OCX. Lớp này có thể cài đặt ở dạng ứng dụng Web gồm những trang ASP sử dụng các File OCX. Trong mô hình Internet (Internet model), lớp khách là trình duyệt Internet Explorer hay Netscape.

- **Lớp giao dịch (Business Tier)** cung cấp các dịch vụ quản trị, tổ chức và khai thác CSDL. Các component trước đây được cài đặt trên lớp khách, nay được cài đặt trên lớp giao dịch. Ví dụ, một người sử dụng trên máy khách đặt mua hàng, lớp giao dịch kiểm tra mã hàng còn nữa hay không mới quyết định tiếp tục bán hay không bán. Thành phần của lớp giao dịch trong mô hình Internet là Web Server và COM+/MTS. Công nghệ của Microsoft với Web Server là IIS (Internet Information Services) sử dụng ASP để kết nối Client với COM. Web Server giao tiếp với COM+/MTS component qua COM. COM+/MTS component điều khiển tất cả giao tiếp với lớp dữ liệu nguồn thông qua ODBC hoặc OLE - DB.

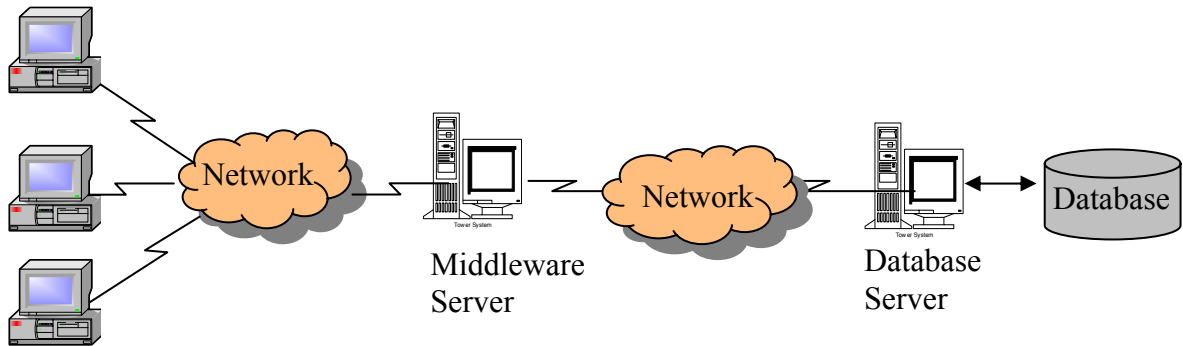
- **Lớp nguồn dữ liệu (Data Source)**

Lớp nguồn dữ liệu cung cấp các dịch vụ tổ chức và lưu trữ các hệ cơ sở dữ liệu quan hệ. Sẵn sàng cung cấp dữ liệu cho lớp giao dịch. Đặc trưng của lớp này là SQL Server quản lý

dữ liệu và các thành phần trong cơ sở dữ liệu như bảng, dữ liệu và các thủ tục lưu trữ. Trong mô hình Internet lớp nguồn dữ liệu quản lý các hoạt động của một lượng lớn Clients. Microsoft phát triển DNA mô hình chung (Framework) xây dựng cho ứng dụng nhiều lớp.

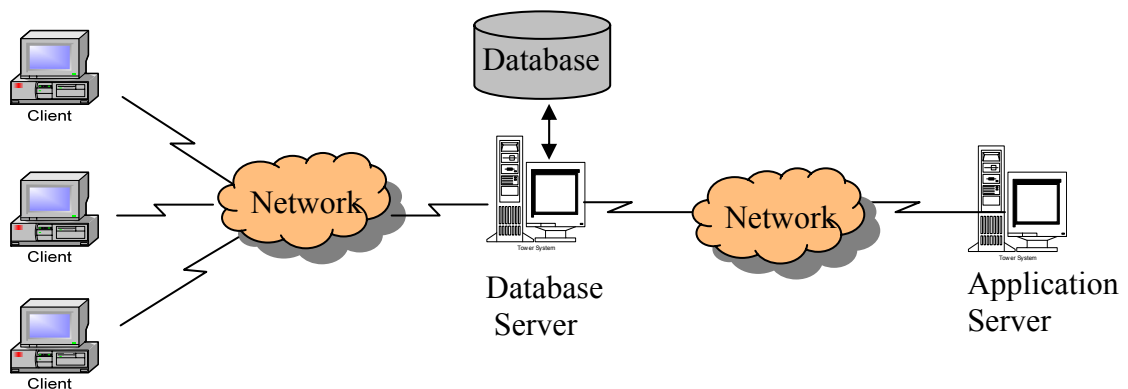
Tóm lại, đặc trưng của mô hình tính toán phân tán là các tài nguyên của mạng như thiết bị phần cứng, chương trình, dữ liệu được phân tán theo địa lý trên mạng. Người sử dụng có thể truy nhập khai thác trực tuyến hoặc và khai thác tương tác dữ liệu từ xa.

Hình 1.4a và 1.4b, mô tả các loại mô hình Client-Server 3 lớp. Hình 1.4a có sử



Hình 1.4a: Mô hình Client-Server 3 lớp có sử dụng Middleware server

dụng Middleware server với mục đích quản lý và điều khiển các tiến trình tương tác của hệ thống. Nghĩa là sử dụng Middleware server để điều khiển và quản lý nhiều kết nối đồng thời truy xuất CSDL của các trạm đầu cuối. Trong cấu trúc loại 3 lớp như hình 1.4b, sử dụng Application server để hỗ trợ cho một số tiến trình ứng dụng cụ thể như truy xuất ghi, cập nhật cơ sở dữ liệu chẳng hạn.

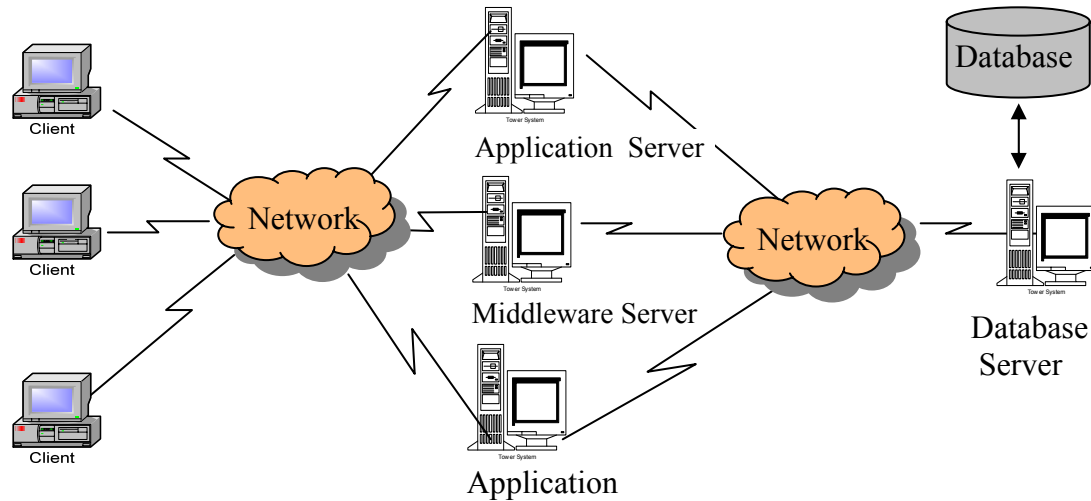


Hình 1.4b: Mô hình Client-Server 3 lớp có sử dụng Application server

1.7.3 Kỹ thuật lập trình cơ sở dữ liệu - Web động

Cơ sở dữ liệu phân tán ngày nay được ứng dụng rộng rãi trên mạng máy tính. Dữ liệu lưu trữ trên các máy tính khác nhau tại các vị trí địa lý khác nhau. Với cách nhìn của người sử dụng là trong suốt và dễ sử dụng. Để đơn giản người ta sao lậ các CSDL thành nhiều bản sao và được cài đặt trên nhiều vị trí khác nhau. Phương pháp này tạo ra độ an toàn cao, đáp ứng được các nhu cầu truy nhập của người sử dụng.

Kỹ thuật kết nối CSDL- Web hỗ trợ cho người sử dụng có tạo ra những trang Web động tùy biến, kết nối tính toán và truy vấn cơ sở dữ liệu từ các máy khách từ xa nhờ các kỹ thuật HTML (HypeTex Markup Language), XML (eXtensible Markup Language) hoặc XSL (eXtensible Style Language). Kỹ thuật kết nối CSDL- Web phổ biến như:



Hình 1.5 Mô hình Client-Server nhiều lớp

- CGI (Common Gateway Interface) hỗ trợ để tạo ra những trang Web tùy biến theo yêu cầu từ máy khách gửi đến.
- ISAPI (Internet Server Application Programming Interface) cho phép lập trình ứng dụng trên máy người sử dụng Web dưới dạng một thư viện liên kết động đơn (Dynamic Link Library) được nạp cùng lúc vào bộ nhớ. Các ứng dụng của ISAPI thực hiện nhanh hơn các ứng dụng của CGI.
- IDC (Internet Database Connector) là một ứng dụng ISAPI, hỗ trợ của IIS (Internet Information Server) truy nhập CSDL qua ODBC kết nối CSDL trên Internet. Với công nghệ ASP (Active Server Pages) cho phép lập trình theo dạng kịch bản tạo những ứng dụng có tính tương tác và hiệu suất cao khi truy xuất cơ sở dữ liệu trên máy chủ.
- JDBC (Java DataBase Connectivity) hỗ trợ lập trình mạng truy xuất CSDL bằng ngôn ngữ Java. Cho phép người sử dụng đầu cuối truy nhập vào các hệ CSDL bằng MS SQL Server, MS Access, Oracle, hay bằng ngôn ngữ truy vấn SQL...

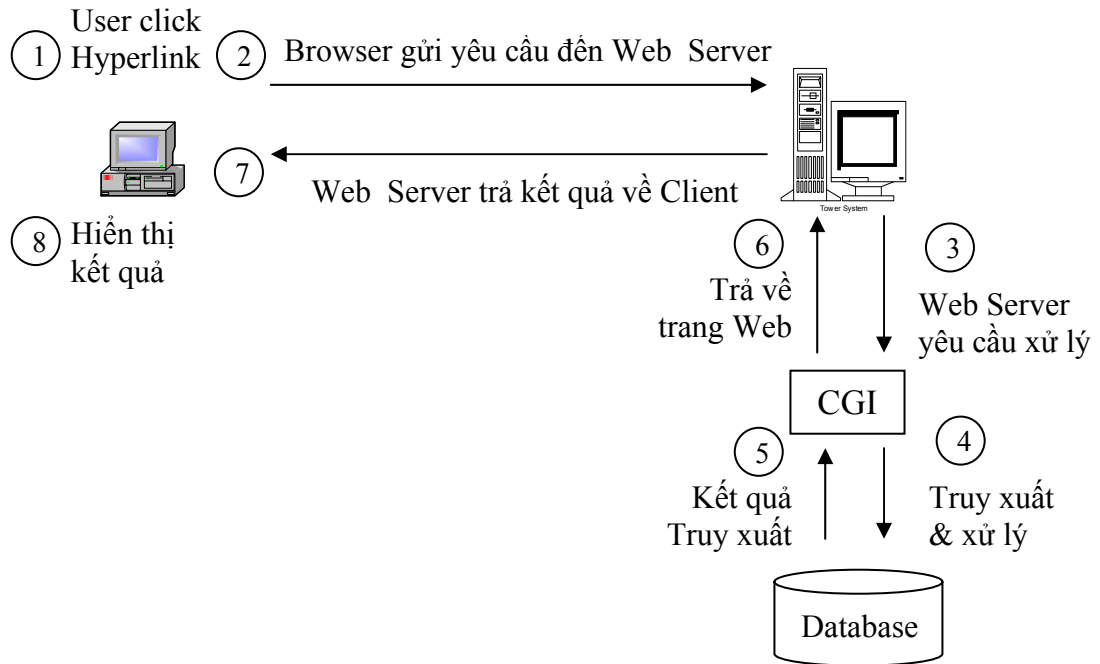
1.7.4 Kiến trúc hệ thống Server (Server System Architecture)

Hệ thống kiến trúc Server bao gồm Server giao tác (Transaction Server) và Data Server.

Giao tác Server: Cũng được gọi là Query Server, cung cấp giao diện cho các trạm khách (Clients) có thể gửi câu hỏi và yêu cầu thực hiện và gửi trả lại kết quả. Yêu cầu có thể biểu diễn bằng ngôn ngữ SQL hoặc bằng giao tiếp trong trình ứng dụng.

- Tiến trình Server (Serverprocess): Tiến trình nhận các yêu cầu từ các máy trạm clients, thực hiện yêu cầu và trả lại kết quả cho các máy trạm. Các yêu cầu của máy trạm được

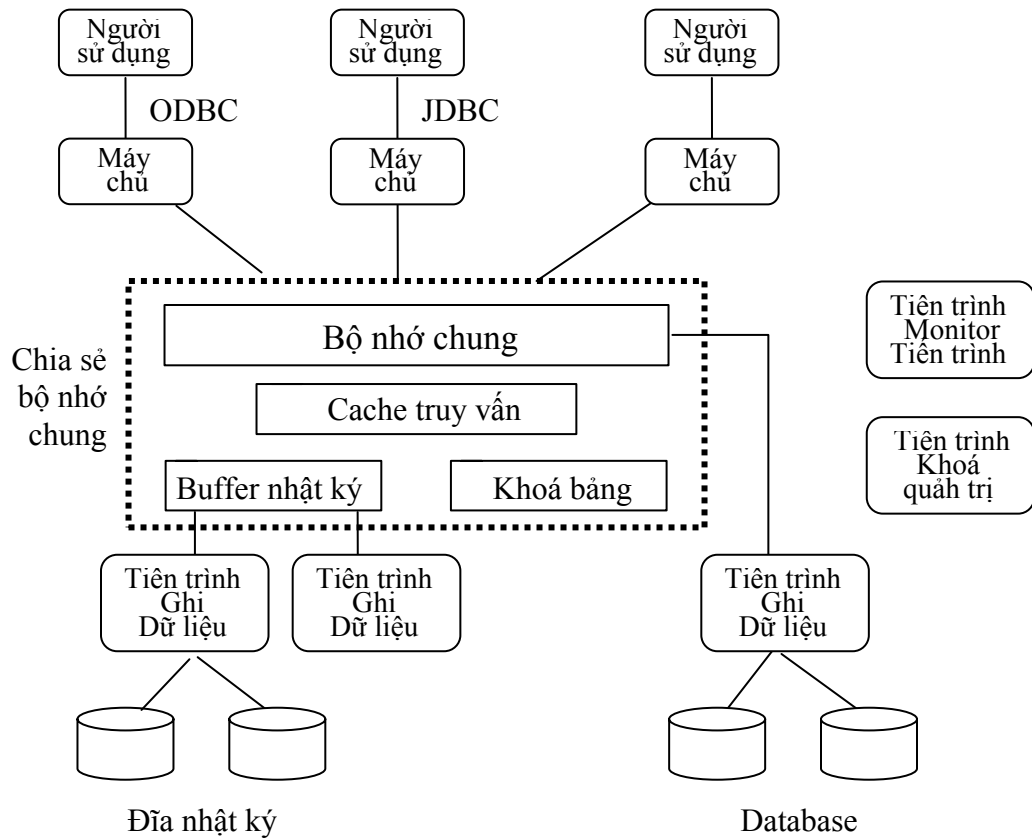
nhận từ giao tiếp người sử dụng hay từ tiến trình người sử dụng được nhúng trong SQL, JDBC hay trong các giao thức tương tự.



Hình 1.6 Cơ chế làm việc của một ứng dụng CGI

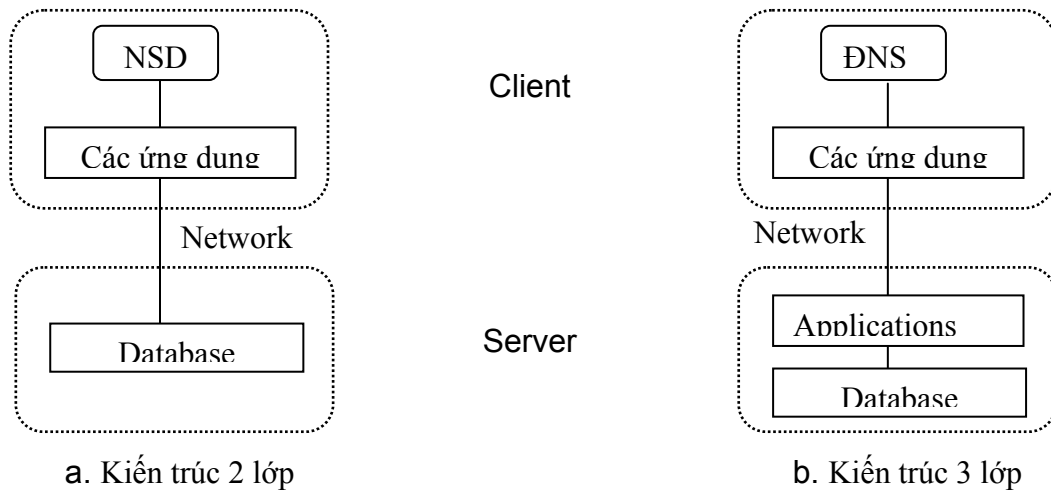
- Lock manager process: Đây là tiến trình khoá cấp quyền truy nhập cơ sở dữ liệu của tiến trình quản trị CSDL cho người sử dụng.
- Database writer process: Tiến trình cho phép đọc dữ liệu vào bộ nhớ phụ và được phép sửa đổi, sau đó được phép cập nhật trở lại CSDL.
- Log writer process: Tiến trình nhật ký theo dõi quá trình đọc, sửa đổi và cập nhật trở lại dữ liệu của tiến trình Database writer process.
- Checkpoint process: Tiến trình kiểm soát định kỳ khuôn dạng trình bày .
- Process monitor process: Đây là tiến trình giám sát các tiến trình khác. nếu phát hiện một tiến trình nào đấy bị lỗi, nó yêu cầu ngắt và thực hiện lại.

Server dữ liệu (Data Server): Cho phép Clients tương tác với Servers bằng các thao tác đọc hay cập nhật dữ liệu. Ví dụ như File Server cung cấp giao diện hệ thống file cho các máy trạm clients có thể thực hiện tạo file, cập nhật dữ liệu, đọc dữ liệu hay xoá dữ liệu



Hình 1.7 Kiến trúc hệ thống Server

1.7.5 Các mô hình kiến trúc ứng dụng



Hình 1.8 Kiến trúc ứng dụng 2 và 3 lớp

Câu hỏi trắc nghiệm

1. Cơ sở dữ liệu là:
 - A. Một bộ sưu tập rất lớn về các loại dữ liệu tác nghiệp
 - B. Một tập File dữ liệu
 - C. Một tập các chương trình ứng dụng và dữ liệu.
 - D. Hệ quản trị cơ sở dữ liệu.
 2. Cơ sở dữ liệu là tài nguyên thông tin... ..
 - A. Truy xuất theo chế độ trực tuyến hay tương tác.
 - B. Chia sẻ chung cho nhiều người sử dụng.
 - C. Cài đặt trên mạng máy tính.
 - D. Trên các thiết bị đầu cuối.
 3. Đối tượng nghiên cứu của CSDL là
 - A. Các thực thể.
 - B. Các mối quan hệ.
 - C. Các thực thể và mối quan hệ giữa các thực thể.
 - D. Các mối liên kết giữa các thực thể. 4. Cơ sở dữ liệu có những ưu điểm:
 - A. Giảm bớt dư thừa dữ liệu trong lưu trữ:
 - B. Tránh không nhất quán trong lưu trữ dữ liệu
 - C. Bảo đảm được tính toàn vẹn của dữ liệu
 - D. Không thể phản ánh thế giới hiện thực dữ liệu,
 5. Mô hình kiến trúc 3 mức của hệ CSDL gồm:
 - A. Mô hình trong, mô hình vật lý
 - B. Mô hình dữ liệu
 - C. Mô hình ngoài, khung nhìn của người sử dụng.
 - D. Các mô hình con dữ liệu
 6. Mô hình dữ liệu là cách nhìn....
 - A. Toàn bộ nội dung thông tin của CSDL
 - B. Khung nhìn của người sử dụng
 - C. Là cách tổ chức lưu trữ
 - D. Là cách biểu diễn cơ sở dữ liệu trừu tượng ở mức thấp nhất.
 7. Mục tiêu của các hệ cơ sở dữ liệu
 - A. Cấu trúc lưu trữ dữ liệu và chiến lược truy nhập tới các hệ cơ sở dữ liệu.
 - B. Dữ liệu chỉ được biểu diễn, mô tả một cách duy nhất.
 - C. Cấu trúc dữ liệu và các hệ chương trình ứng dụng không phụ thuộc lẫn nhau.
 - D. Cấu trúc lưu trữ dữ liệu và các hệ chương trình ứng dụng phụ thuộc lẫn nhau.
 8. Có rất nhiều cách nhìn dữ liệu ở người sử dụng khác nhau có cách nhìn dữ liệu khác nhau và các hệ ứng dụng khác nhau có những cách nhìn dữ liệu cũng khác nhau.
 - A. Mô hình ngoài.
-
-

- B. Mô hình quan niệm.
 - C. Mô hình trong.
 - D. Mô hình dữ liệu
9. Có duy nhất một cách nhìn dữ liệu ở, biểu diễn toàn bộ nội thông tin trong CSDL đó là cách nhìn dữ liệu tổng quát của người sử dụng.
- A. Mô hình ngoài.
 - B. Mô hình quan niệm.
 - C. Mô hình trong.
 - D. Mô hình dữ liệu
10. Có duy nhất một và chỉ một cách biểu diễn CSDL dưới dạng lưu trữ
- A. Logic.
 - B. Vật lý.
 - C. Tổng quát.
 - D. Nội dung thông tin

Câu hỏi & bài tập

1. Cơ sở dữ liệu là gì , hiểu thế nào là một hệ cơ sở dữ liệu tác nghiệp.
2. Sự cần thiết tổ chức lưu trữ dữ liệu theo lý thuyết cơ sở dữ liệu.
3. Cho ví dụ minh họa về giám sát dư thừa dữ liệu trong lưu trữ và không nhất quán dữ liệu trong lưu trữ làm cho dữ liệu mất đi tính toàn vẹn
4. Trình bày tổng quát kiến trúc mô hình hệ cơ sở dữ liệu 3 lớp .
5. Trình bày và phân tích tính ổn định trong mô hình quan niệm.
6. Vai trò & chức năng của ánh xạ quan niệm trong & ánh xạ quan niệm ngoài .
7. Mục tiêu của các hệ cơ sở dữ liệu? Ví dụ minh họa.
8. Chứng minh rằng kiến trúc mô hình cơ sở dữ liệu 3 lớp đảm bảo được tính độc lập dữ liệu và tính ổn định cao.
9. Tại sao nói, mô hình dữ liệu là cách nhìn toàn bộ nội dung thông tin của CSDL, sơ đồ quan niệm là định nghĩa của cách nhìn ấy. Ví dụ minh họa.
10. Hiểu thế nào về khái niệm “tính toàn vẹn dữ liệu” và “tham chiếu toàn vẹn”.
11. Hiểu như thế nào về tính độc lập của dữ liệu.
12. Khái niệm File có gì khác với khái niệm cơ sở dữ liệu, ví dụ minh họa ?.
13. Tổ chức lưu trữ dữ liệu kỹ thuật hàm băm. Cho ví dụ minh họa.
14. Chức năng & vai trò của hệ quản trị CSDL & người quản trị CSDL.
15. Trình bày kỹ thuật tổ chức lưu trữ dữ liệu theo B-cây cân bằng Files.
16. Hiểu thế nào về hệ quản trị CSDL (DataBase Management System DBMS)
17. Người quản trị CSDL
18. Nêu một số nhiệm vụ chính của người quản trị CSDL.
19. Nêu một số thí dụ về ràng buộc kiểu, ràng buộc giải tích và ràng buộc logic.
20. Mô hình cơ sở dữ liệu Client- Server.

2

CÁC MÔ HÌNH CƠ SỞ DỮ LIỆU

Trong chương I đã giới thiệu mục tiêu của các hệ cơ sở dữ liệu là tính độc lập và tính toàn vẹn của dữ liệu. Trong chương này giới thiệu các phương pháp tiếp cận mô hình dữ liệu: mô hình cơ sở dữ liệu mạng, phân cấp, cơ sở dữ liệu quan hệ và mô hình thực thể quan hệ. Cơ sở để so sánh, đánh giá một CSDL tốt dựa vào các mục đích xây dựng hệ thống, cơ sở dữ liệu hướng giá trị hay hướng đối tượng, tính dư thừa và giải quyết tốt mối quan hệ nhiều-nhiều.

- Mô hình dữ liệu
- Mô hình CSDL phân cấp
- Cấu trúc biểu diễn dữ liệu phân cấp
- Mô hình CSDL mạng
- Cách tiếp cận mô hình CSDL quan hệ
- Mô hình thực thể quan hệ

2.1 Mở đầu

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.*

2.2 Mô hình dữ liệu (Data Model)

Một mô hình dữ liệu là một hệ thống hình thức toán học, bao gồm:

- Hệ thống các ký hiệu biểu diễn dữ liệu.
- Tập hợp các phép toán thao tác trên cơ sở dữ liệu.

Đặc trưng của một mô hình dữ liệu:

- Tính ổn định khi thiết kế mô hình dữ liệu.
- Tính đơn giản có nghĩa là dễ hiểu và dễ thao tác.
- Tính dư thừa cần phải kiểm tra kỹ lưỡng.
- Tính đối xứng phải được bảo toàn và
- Có cơ sở lý thuyết vững chắc.

2.2.1 Phân biệt giữa các mô hình dữ liệu

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

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

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

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

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

2.2.2 Các hệ thống CSDL đối tượng và tri thức

Cơ sở dữ liệu hướng đối tượng và hệ quản trị hướng đối tượng (Object Oriented Database management Systems – OO DBMS) mô tả các kiểu dữ liệu được xây dựng bằng phương pháp tạo bản ghi và tạo tập hợp. Các quan hệ được xây dựng từ các bộ bằng thao tác tạo một tập hợp các bản ghi có khuôn dạng thống nhất.

Che dấu dữ liệu (Encapsulation): Nghĩa là khi có yêu cầu truy xuất đến các đối tượng thuộc kiểu đặc biệt, phải qua các thủ tục đã được định nghĩa cho các đối tượng đó. Chẳng hạn định nghĩa *stack* như là một kiểu và định nghĩa các thao tác PUSH, POP áp dụng cho *stack*.

Đặc tính nhận dạng đối tượng (Object Identity) là khả năng phân biệt các đối tượng. Nghĩa là cấu trúc các kiểu cơ bản như nhau. Các kiểu cơ bản là chuỗi ký tự, số.

2.3 Mô hình CSDL phân cấp (Hierarchy Data Model)

2.3.1 Cấu trúc biểu diễn dữ liệu phân cấp

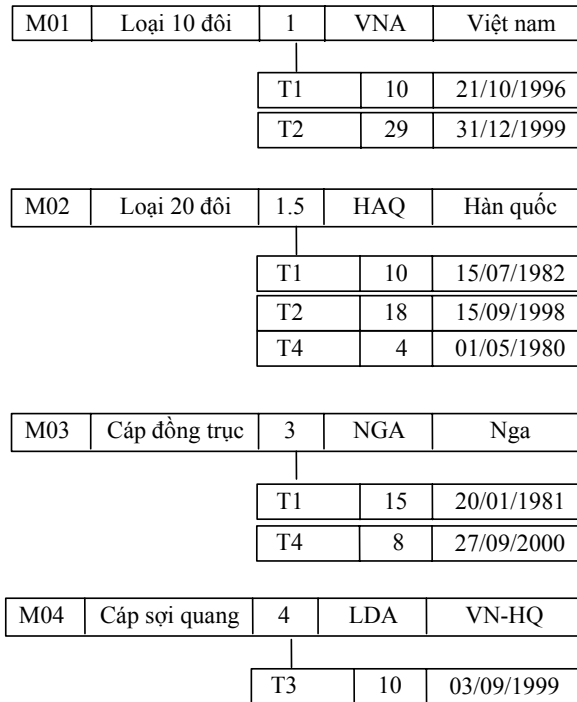
Trong mô hình CSDL phân cấp, dữ liệu được biểu diễn bằng cấu trúc cây. Một CSDL phân cấp là tập các cây (rừng cây). Trong mỗi một cây chỉ chứa một và chỉ một xuất hiện của bản ghi gốc, gọi là bản ghi đỉnh, và dưới nó là tập các xuất hiện của các bản ghi phụ thuộc. Các bản ghi phụ thuộc có thể là tùy ý hoặc không tồn tại. Một bản ghi gốc có thể có một số bất kỳ các bản ghi phụ thuộc và các bản ghi phụ thuộc có thể có một số các bản ghi phụ thuộc mức thấp hơn... Hình 2.1 biểu diễn một mô hình CSDL phân cấp về tuyến cáp và các loại cáp. Dữ liệu được biểu diễn bằng 4 cấu trúc cây đơn giản, trong đó gốc là xuất hiện kiểu bản ghi loại cáp bao gồm các thông tin mã cáp, tên cáp, số lượng, mã nước sản xuất, tên nước sản xuất. Các bản ghi phụ thuộc là kiểu các bản ghi các tuyến cáp có lắp đặt các loại cáp đó, bao gồm các thông tin về số hiệu tuyến cáp, tên gọi tuyến cáp, độ dài và ngày hoàn thành việc xây dựng tuyến cáp đó. Như vậy các bản ghi gốc là các kiểu bản ghi về các loại cáp đã được lắp đặt và các bản ghi phụ thuộc là các bản ghi về thông tin các tuyến cáp.

Theo định nghĩa, không thể có các bản ghi phụ thuộc mà không tồn tại bản ghi gốc, nghĩa là không thể tồn tại các loại cáp mà chưa được lắp đặt trên một tuyến nào cả. Như vậy có thể có thể tồn tại các loại cây vừa có xuất hiện của bản ghi gốc và các bản ghi phụ thuộc, nghĩa là mỗi một loại cáp có thể được lắp đặt trên nhiều tuyến cáp khác nhau và trên một tuyến cáp có thể có nhiều loại cáp khác nhau được lắp đặt. Có loại cây chỉ tồn tại bản ghi gốc thoái hoá, dưới nó không tồn tại bản ghi phụ thuộc, nghĩa là có ít nhất một loại cáp chưa được đưa vào lắp đặt sử dụng. Theo định nghĩa, không tồn tại loại cây chỉ có các bản ghi phụ thuộc mà không có bản gốc, tức là trên mọi tuyến cáp phải có ít nhất một loại cáp. Điều gì sẽ xảy ra khi trên trên một tuyến nào đó có duy nhất một loại cáp bị huỷ bỏ.

2.3.2 Ngôn ngữ thao tác trên CSDL phân cấp

- Biểu diễn phụ thuộc trong mô hình phân cấp: Các đường nối từ bản ghi gốc trở xuống các bản ghi phụ thuộc, hay từ bản ghi cha trở xuống bản ghi con biểu diễn mối quan hệ giữa các bản ghi trong mô hình phân cấp. Ví dụ trong bản ghi Phiếu xuất kho có thể xác định được mã và đơn giá của các loại cáp.
- Chèn thêm: Trong cấu trúc hình cây, có một và chỉ một xuất hiện bản ghi gốc, tức là bắt buộc phải có xuất hiện bản ghi về các loại cáp. Nếu muốn lưu trữ thông tin về một tuyến cáp mới vào CSDL, thì điều này không thể thực hiện được, vì không thể thêm thông tin về các xuất hiện kiểu bản ghi phụ thuộc (thông tin về các tuyến cáp) vào CSDL phân cấp khi chưa có thông tin về các loại cáp.
- Loại bỏ: Trong một số cấu trúc cây dữ liệu, thông tin về một vấn đề nào đấy được chứa trong một xuất hiện kiểu bản ghi phụ thuộc duy nhất thì khi loại bỏ sẽ làm mất thông tin về vấn đề trên. Ví dụ khi loại bỏ loại cáp M04 trên tuyến T03, sẽ kéo theo thông tin về tuyến cáp T03 bị mất và thông tin về cáp mang mã hiệu M04 cũng bị mất luôn. Như vậy có thể

xây ra hiện tượng mất thông tin về các tuyến cáp hoặc thông tin về các loại cáp khi thực hiện các thao tác loại bỏ.



Hình 2.1 Dữ liệu biểu diễn dưới dạng phân cấp

1. Bản ghi gốc: Mã cáp, tên cáp, số lượng, mã nước sản xuất và tên nước sản xuất
2. Bản ghi phụ thuộc: Mã tuyến, độ dài tuyến và ngày lắp đặt cáp

- Sửa đổi: Nếu cần phải sửa đổi một số thông tin trong các xuất hiện kiểu bản ghi phụ thuộc thì phải duyệt toàn bộ mô hình dữ liệu, bằng cách phải dò tìm trong từng xuất hiện. Như vậy khả năng dò tìm không hết có thể xảy ra, sẽ dẫn đến sự xuất hiện mâu thuẫn thông tin và không nhất quán dữ liệu trong lưu trữ. Ví dụ cần thay đổi tên gọi của các tuyến cáp hoặc tên gọi các loại cáp, nếu số các xuất hiện kiểu bản ghi loại này rất lớn, khả năng duyệt sót vẫn có thể xảy ra, nghĩa là mâu thuẫn thông tin, không nhất quán thông tin sẽ xảy ra.
- Các phép tìm kiếm: Các xuất hiện của các bản ghi phụ thuộc chỉ tồn tại khi và chỉ khi tồn tại xuất hiện kiểu bản ghi gốc.

Ví dụ Q1: Tìm số hiệu của các tuyến có lắp đặt cáp MC#="M02". Q2: Tìm số hiệu các loại cáp lắp đặt trên tuyến TC#='T02'.

Bản ghi gốc: MC#='M02' Next: Không còn bản ghi gốc: Thoát
 Không tìm thấy: Thoát Không tìm thấy TC#='T2'
 In TC# Goto Next
 Thoát Tìm thấy: In MC# Goto Next

- Deleted:
- Deleted: P
- Deleted: P
- Deleted:
- Deleted:
- Deleted: S
- Deleted: S
- Deleted: S
- Deleted: S
- Deleted: S
- Deleted: P
- Deleted: Goto Nex
- Deleted:

Các truy vấn thường đối xứng nhau nhưng kết quả của các phép tìm kiếm không đối xứng. Vì vậy trong cấu trúc lưu trữ của mô hình phân cấp rất hạn chế khi thực hiện việc tìm kiếm thông tin. Nếu CSDL phân cấp càng lớn thì tính phức tạp càng cao. Gây nhiều phức tạp cho người sử dụng, nhất là đối với lập trình viên phải mất nhiều thời gian công sức để lập trình và bảo trì, hiệu chỉnh các chương trình.

Tóm lại thông tin tổ chức lưu trữ theo mô hình phân cấp được biểu diễn dữ liệu trong một tập duy nhất theo cấu trúc cây. Trong mỗi một cây, tồn tại một và chỉ duy nhất một xuất hiện kiểu bản ghi gốc và cùng với nó có một tập các xuất hiện kiểu bản ghi phụ thuộc. Khi thao tác trên CSDL phân cấp bằng ngôn ngữ thao tác dữ liệu, có nhiều khả năng xảy ra thừa hoặc thiếu thông tin, mâu thuẫn thông tin dẫn đến sự không nhất quán dữ liệu trong lưu trữ. Tính toàn vẹn của dữ liệu không được đảm bảo. Các câu hỏi hỏi-đáp, tìm kiếm không có tính đối xứng. Tính độc lập của dữ liệu dễ bị vi phạm. Tính ổn định không cao.

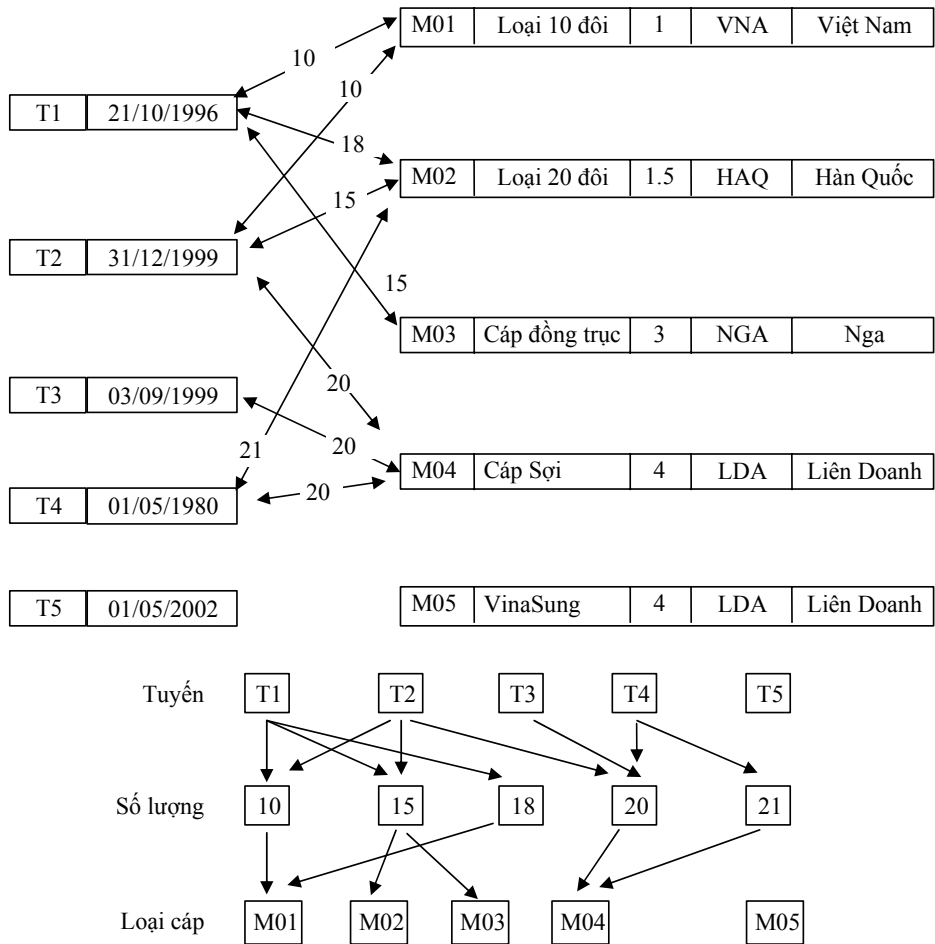
2.4 Mô hình CSDL mạng (Network Data Model)

2.4.1 Cấu trúc biểu diễn dữ liệu mạng

Mô hình dữ liệu mạng là mô hình thực thể quan hệ, trong đó các mối liên kết bị hạn chế trong kiểu một - một và nhiều - một. Trong mô hình CSDL mạng, dữ liệu được biểu diễn trong các bản ghi liên kết với nhau bằng các mối nối liên kết (link) tạo thành một đồ thị có hướng. CSDL mạng có cấu trúc tổng quát hơn so với cấu trúc CSDL phân cấp. Mỗi một xuất hiện của một bản ghi có thể có rất nhiều các xuất hiện kiểu bản ghi trên nó và các xuất hiện kiểu bản ghi dưới nó. Ngoài các kiểu bản ghi biểu diễn dữ liệu còn có kiểu bản ghi các phần tử kết nối, biểu diễn sự kết hợp giữa các biểu diễn dữ liệu. Cho phép mô hình hoá tương ứng nhiều - nhiều. Hình 2.2 biểu diễn một mô hình CSDL mạng về các tuyến cáp và các loại cáp được lắp đặt. Ngoài các xuất hiện kiểu bản ghi tuyến cáp và các xuất hiện kiểu các bản ghi các loại cáp còn có các xuất hiện kiểu các bản ghi về các phần tử kết nối đó là các phần tử số lượng. Các phần tử này kết nối tuyến cáp và các loại cáp là thông tin về tình hình lắp đặt. Sơ đồ T1 → 300 → M01 chỉ ra rằng tuyến T1 có 10 cáp M01, T1 → 18 → M02 có nghĩa là T1 có 18 cáp loại có mã số là M02...

- Mạng chứa hai kiểu thực thể: Trong mỗi một tuyến cáp, có nhiều loại cáp khác nhau với số lượng khác nhau. Mỗi một loại cáp có thể có mặt trong nhiều tuyến cáp khác nhau. Kiểu bản ghi số lượng có chức năng liên kết hai kiểu bản ghi tuyến cáp và các loại cáp. Mỗi một xuất hiện kiểu bản ghi số lượng biểu diễn mối liên kết giữa các bản ghi tuyến cáp với các bản ghi loại cáp. Thiết lập mối liên kết giữa một tuyến cáp và một loại cáp sao cho mỗi xuất hiện tương ứng của số lượng đều có mặt trong xuất hiện các tuyến cáp và loại cáp. Ví dụ tuyến cáp “T1” và loại cáp mã “M01” có mối liên kết với số lượng là 10. “T5” và “M07” không có mối liên kết, điều này có nghĩa là loại cáp “M07” chưa có mặt trong một tuyến nào cả và tuyến “T5” chưa có một loại cáp nào được lắp đặt.
- Mạng chứa hơn hai kiểu thực thể: Liên kết n kiểu thực thể biểu diễn bằng một kiểu bản ghi liên kết n kiểu bản ghi đó với nhau. Mỗi xuất hiện của bản ghi liên kết sẽ là thành viên của đúng một xuất hiện của một trong số n kiểu bản ghi. Như vậy sẽ biểu diễn mối liên kết

n thực thể tương ứng. Ví dụ thêm một kiểu thực thể mới - đơn vị thi công tuyến cáp. Có thể một đơn vị thi công nhiều tuyến cáp và trong một tuyến cáp có nhiều đơn vị thi công. Thêm kiểu thực thể ngày nhập kho (NHAP) của của các loại cáp trước khi đưa vào sử dụng hoặc đưa thêm thông tin về hãng sản xuất....



Hình 2.2 Cách tiếp cận CSDL kiểu mạng

- Mạng chỉ chứa một kiểu thực thể: Tồn tại cơ sở dữ liệu chỉ chứa một thực thể. Ví dụ cơ sở dữ liệu chứa thông tin về phụ tùng và linh kiện, trong đó một linh kiện tự nó có thể là một phụ tùng và tự nó có thể chứa các phụ tùng khác. Như vậy mạng cơ sở dữ liệu chỉ chứa duy nhất một kiểu bản ghi phụ tùng. Và mỗi một phụ tùng lại có thể là tổ hợp của một số chi tiết thành phần, lại vừa có thể là thành phần của các tổ hợp khác. Như vậy cơ sở dữ liệu mạng có hai thực thể, hai kiểu bản ghi nhưng thực chất chỉ là một.

2.4.2 Ngôn ngữ dữ liệu thao tác trên CSDL mạng

- Chèn thêm: Khi thêm các các kiểu bản ghi mới, có thể con trỏ (mối nối) tự trỏ vào nó. Ví dụ, có chèn thêm các thông tin xuất hiện bản ghi về tuyển cấp hoặc xuất hiện kiểu bản ghi về các loại cấp mà chưa có trong CSDL và chưa có các bản ghi kết nối, nghĩa là chưa có thông tin về các loại cấp được lắp đặt, nhưng mô hình CSDL mạng vẫn đảm bảo được tính nhất quán của dữ liệu và tính toàn vẹn của dữ liệu.
- Loại bỏ: Có thể loại bỏ các xuất hiện kiểu bản ghi dữ liệu và xuất hiện kiểu bản ghi kết nối trong mô hình CSDL mạng mà không ảnh hưởng tới tính nhất quán và tính toàn vẹn dữ liệu. Có thể xoá một loại cấp nào đó khi không còn sử dụng. Ví dụ cần xóa bỏ 8 cấp M04 trong T1 bằng cách xoá bỏ các con trỏ giữa T1 và 8; giữa 8 và M04 mà không làm mất thông tin, không xuất hiện dị thường thông tin.
- Sửa đổi: Có thể sửa đổi nội dung dữ liệu mà không cần duyệt qua mô hình và cũng không làm xuất hiện mâu thuẫn dữ liệu.
- Các phép tìm kiếm: Các câu hỏi đối xứng và kết quả của phép tìm kiếm cũng đối xứng với nhau như trong mô hình CSDL quan hệ.

Ví dụ Q1: Tìm số hiệu của các tuyển cấp có lắp đặt cấp MC#="M02".

Q2: Tìm số hiệu của các loại cấp trên tuyển TC#="T02"

Bản kết nối: MC#='M02'
 Không có kết nối: Thoát
 In TC# Thoát

Bản kết nối: TC#='T02'
 Không có kết nối: Thoát
 In MC# Thoát

- Deleted: P
- Deleted: P
- Deleted: S
- Deleted: S
- Deleted:
- Deleted:
- Deleted: S
- Deleted: P
- Deleted:

Mô hình CSDL mạng là mô hình đối xứng, vì vậy các câu hỏi và kết quả các câu hỏi tìm kiếm thường đối xứng với nhau. Khi thực hiện các phép lưu trữ như chèn thêm, loại bỏ hay sửa đổi dữ liệu trong mô hình CSDL mạng vẫn bảo đảm được sự nhất quán của dữ liệu và tính toàn vẹn của dữ liệu. Cách tiếp cận CSDL mạng là phương pháp biểu diễn dữ liệu trong các tệp theo cấu trúc dữ liệu chặt chẽ. Các xuất hiện kiểu bản ghi được kết nối với nhau bằng các xuất hiện kiểu bản ghi liên kết. Khi thao tác các phép cập nhật không xuất hiện các dị thường thông tin. Tuy nhiên cấu trúc dữ liệu rất trong mô hình CSDL mạng quá phức tạp vì quá nhiều liên kết giữa các xuất hiện dữ liệu với nhau bằng các xuất hiện kết nối. Vì vậy việc thiết kế và cài đặt cơ sở dữ liệu mạng thường rất khó khăn, nhất là xây dựng các phép toán thao tác trên nó.

2.5 Cách tiếp cận mô hình CSDL quan hệ.

2.5.1 Khái niệm lý thuyết tập hợp quan hệ

Khái niệm toán học lý thuyết tập hợp là nền tảng xây dựng lý thuyết mô hình dữ liệu quan hệ. Đó là tập con của tích Đề các ((Descartes) của danh sách các miền xác định (Domain). Miền là tập các giá trị, là kiểu của dữ liệu. Chẳng hạn, tập các chuỗi ký tự số có chiều dài chính xác là 7, là miền của của thuộc tính điện thoại. Tích Đề các của các miền A₁, A₂, ..., A_n được biểu diễn:

$$A_1 \times A_2 \times \dots \times A_n = \{ (a_1, a_2, \dots, a_n) \mid a_i \in A_i, i = 1 \div n \}.$$

Quan hệ (Relation) là một tập con của tích Đề các của một hoặc nhiều miền. Sẽ ngầm hiểu quan hệ là hữu hạn. Các phần tử của quan hệ gọi là bộ (a_1, a_2, \dots, a_n) , có n giá trị thành phần, hay được gọi là n _bộ.

Khái niệm tập n _bộ không phải là khái niệm duy nhất đối với lý thuyết cơ sở dữ liệu kiểu quan hệ. Nếu gán tên thuộc tính cho các cột, khi đó thứ tự của các cột là không quan trọng. Vì vậy có thể xem các bộ như là ánh xạ từ các thuộc tính đến tập các giá trị của miền xác định các thuộc tính. Ánh xạ μ được định nghĩa như sau:

$$\mu : \Omega = \{A_1, A_2, \dots, A_n\} \rightarrow \Omega$$

Khi đó quan hệ là tập n _bộ: $\{\mu(A_1), \mu(A_2), \dots, \mu(A_n)\} = \{(a_1, a_2, \dots, a_n)\}$.

Như vậy, "Cơ sở dữ liệu quan hệ" có thể hiểu là dữ liệu được người sử dụng nhìn dưới dạng một quan hệ toán học và các phép toán thao tác dữ liệu được xây dựng trên các cấu trúc quan hệ toán học. Một quan hệ là một bảng và mỗi bảng là một cấu trúc quan hệ toán học. Nói cách khác, cơ sở dữ liệu quan hệ được biểu diễn dưới dạng các bảng, gồm các cột là thuộc tính và các hàng là tập n giá trị của thực thể, đối tượng, có cấu trúc. Cấu trúc dữ liệu quan hệ là mối liên kết giữa các bộ được biểu diễn duy nhất bằng các giá trị dữ liệu trong các cột được rút ra từ miền chung. Giữa các thuộc tính có mối quan hệ ràng buộc phụ thuộc lẫn nhau và các phụ thuộc cũng là các thực thể đặc biệt.

Ví dụ Quan hệ về tuyến cáp – các loại cáp – nhật ký lắp đặt

TC#: Mã tuyến cáp, TC: Tên gọi tuyến cáp, DAI: độ dài tuyến cáp,
 NGHT: Ngày hoàn thành tuyến, MC#: Mã cáp, MC: Tên gọi cáp, GIA: Giá cáp,
 N#: Mã nước sản xuất, NSX: Tên nước sản xuất

a) Thông tin về các loại cáp lắp đặt trên các tuyến cáp:

TC#	MC#	SL	NG
T01	M01	10	21/10/1996
T01	M02	20	15/7/1982
T01	M03	15	20/1/1981
T02	M01	29	31/12/1999
T02	M02	18	15/9/1998
T03	M04	10	3/9/1999
T04	M02	4	1/5/1980
T04	M03	8	27/9/2000

b) Thông tin về các tuyến cáp

TC#	TC	DAI	NGHT
T01	Hà Nội- Hải Phòng	120 Km	1/2/1980
T02	Hà Nội- Quảng Ninh	170 Km	31/12/1985
T03	Hải Phòng-Quảng Ninh	60 Km	30/6/1990
T04	Hà Nội- Hà Tây	15 Km	31/12/1979

c) Thông tin về các loại cáp (danh mục các loại cáp):

MC#	MC	GIA	N#	NSX
M01	Loại 10 đôi	1,0	VNA	Việt Nam
M02	Loại 20 đôi	1,5	HAQ	Hàn Quốc
M03	Cáp đồng trục	3,0	NGA	Nga
M04	Cáp sợi quang	4,0	LDA	Liên Doanh VN-HQ

Hình 2.3 Dữ liệu biểu diễn dưới dạng quan hệ

2.5.2 Ngôn ngữ thao tác dữ liệu kiểu quan hệ.

Ngôn ngữ con dữ liệu kiểu quan hệ là tập các phép toán có các cấu trúc quan hệ thao tác trên các dữ liệu dưới dạng quan hệ. Dữ liệu được biểu diễn một cách duy nhất. Ngôn ngữ con dữ liệu gồm nhóm các các phép toán tìm kiếm và nhóm các phép toán cập nhật, lưu trữ.

- Các phép toán lưu trữ: là các phép chèn thêm, sửa đổi và loại bỏ, là các toán tử được xây dựng trên đại số quan hệ. Các phép toán này có khả năng xử lý tập hợp theo lô, các quan hệ được xem như toán hạng trong các toán tử. Khi thực hiện các phép lưu trữ không có khả năng xuất hiện dị thường thông tin và không làm mất sự nhất quán dữ liệu. Bằng ngôn ngữ con dữ liệu kiểu quan hệ, thao tác trên các quan hệ bảo đảm được tính toàn vẹn dữ liệu.
- Các phép toán tìm kiếm: Kết quả của các phép tìm kiếm dữ liệu cũng là một quan hệ. Vì vậy với cách tiếp cận quan hệ, người ta hoàn toàn có khả năng xây dựng dễ dàng một ngôn ngữ con dữ liệu rất đơn giản nhưng cũng rất mạnh ở mức tập hợp nhằm cung cấp các khả năng thuận tiện cho người sử dụng. Đây là một trong những điểm mạnh của cách tiếp cận quan hệ và ngôn ngữ đại số quan hệ. Các câu hỏi tìm kiếm là đối xứng nhau.

Ví dụ Q1: Tìm số hiệu của các tuyến có lắp đặt M02.

```
SELECT TC#
FROM QLC
WHERE MC#="M02"
```

Q2: Tìm số hiệu của các loại cáp có mặt trong tuyến T02.

```
SELECT MC#
FROM QLC
WHERE TC#="T02"
```

2.5.3 Các phép toán cơ bản của đại số quan hệ.

1. *Phép hợp (Union)*: Hợp của các quan hệ khả hợp là một quan hệ gồm các bộ : $R \cup S := \{ t \mid t \in R \text{ or } t \in S \}$
2. *Phép trừ (Minus)*: Hiệu của hai quan hệ khả hợp là một quan hệ gồm các bộ của quan hệ bị trừ, không thuộc quan hệ trừ.: $R - S := \{ t \mid t \in R \text{ and } t \notin S \}$
3. *Tích Đề các (Descartes)*: của hai quan hệ bao gồm các bộ của quan hệ sao cho thành phần đầu là bộ của quan hệ thứ nhất, các thành phần sau thuộc các thành phần của quan hệ thứ hai. $R \times S := \{ t = \langle r, s \rangle \mid r \in R \text{ and } s \in S \}$
4. *Phép chiếu (Projection)*: Từ quan hệ nguồn, loại bỏ một số một số thuộc tính.

$$\pi_{A_{j_1}, A_{j_2}, \dots, A_{j_k}}(R) := \{ \langle a_{j_1}, a_{j_2}, \dots, a_{j_k} \rangle \mid a_{j_i} \in \text{Dom}(A_{j_i}), j_i = j_1 \div j_k \}$$

5. *Phép chọn (Selection)*: Từ quan hệ nguồn, loại bỏ một số một số bộ thỏa biểu thức logic F.

$$\sigma_F(R) := \{ t \mid t_F = \text{"True"} \}$$

Như vậy mô hình CSDL quan hệ bao gồm các quan hệ toán học, là các tập truyền thống tuân theo các ràng buộc của quan hệ toán học. Ngôn ngữ con thao tác dữ liệu gồm các toán tử có cấu trúc quan hệ toán học thao tác trên các quan hệ.

2.5.4 Ưu điểm cách tiếp cận mô hình CSDL quan hệ.

Tính đơn giản: Mô hình CSDL quan hệ đã mô tả được thế giới hiện thực dữ liệu một cách chính xác, khách quan, phù hợp với cách nhìn thông thường của người sử dụng. Cấu trúc dữ liệu có tính đơn giản, bởi các thông tin về các thực thể và các ràng buộc của các thực thể được biểu diễn duy nhất trong các bảng, trong suốt với người sử dụng.

Tính độc lập dữ liệu: Tính độc lập dữ liệu trong các hệ cơ sở dữ liệu quan hệ cao vì, thể hiện được tính độc lập dữ liệu ở mức vật lý, tức là cấu trúc lưu trữ và chiến lược truy nhập CSDL vật lý có thể thay đổi bởi người quản trị CSDL nhưng không làm thay đổi sơ đồ quan niệm và ngược lại. Tính độc lập dữ liệu logic, ở mức quan niệm, là mối quan hệ giữa khung nhìn của người sử dụng và CSDL quan niệm không phụ thuộc lẫn nhau.

Tính đối xứng: Do cấu trúc biểu diễn dữ liệu trong các hệ CSDL quan hệ, các câu hỏi đối xứng, kết quả của các câu hỏi cũng đối xứng và biểu diễn bằng quan hệ.

Có cơ sở lý thuyết vững chắc: Mô hình CSDL quan hệ được xây dựng trên cơ sở lý thuyết toán học quan hệ chặt chẽ, logic. Ngôn ngữ thao tác dữ liệu là tập các phép toán có cấu trúc quan hệ. Tóm lại các hệ CSDL quan hệ có nền tảng lý thuyết vững, chặt chẽ và logic.

2.6 Mô hình thực thể quan hệ (The Entity Relationship Model)

2.6.1 Khái niệm

Mô hình thực thể quan hệ cho phép mô tả lược đồ khái niệm của một tổ chức, áp dụng cho vòng đời của cơ sở dữ liệu mà không để ý đến tính hiệu quả hoặc thiết kế CSDL vật lý như cách tiếp cận với các mô hình khác. Nó mô tả được thế giới thực gồm những những đối tượng cơ bản được gọi là những thực thể và những quan hệ giữa thực thể. Ảnh xạ và những tác động qua lại của thế giới thực phức tạp được mô tả bằng các khái niệm lược đồ. Nó được phát triển để tạo thuận lợi cho thiết kế cơ sở dữ liệu bằng cách cho phép định rõ một lược đồ phức tạp, đại diện cho toàn bộ cấu trúc logic của cơ sở dữ liệu. Dạng đơn giản của mô hình thực thể-quan hệ được dùng làm cơ sở để trao đổi một cách hiệu quả với người sử dụng đầu cuối về CSDL khái niệm. Từ sơ đồ thực thể quan hệ người ta có thể chuyển thành lược đồ khái niệm mô hình quan hệ. Mô hình thực thể quan hệ có thể được xem như là mô hình hướng đối tượng. Mô hình thực thể - quan hệ cơ bản bao gồm ba lớp đối tượng: thực thể (Entity), mối quan hệ (Relationship), và thuộc tính (Attribute).

2.6.2 Thực thể, mối quan hệ, thuộc tính

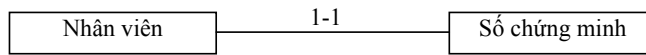
Thực thể (Entity) là những đối tượng dữ liệu cơ bản chứa nội dung các thông tin cần thu thập. Thực thể có thể biểu thị cho người, nơi chốn, sự vật hoặc các biến cố có thông tin đáng chú ý. Một xuất hiện cụ thể của một thực thể được gọi là thể hiện thực thể (Entity)

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.

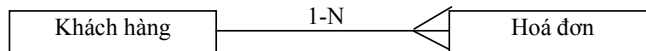
Một nhóm các thực thể cùng chung một số tính chất, một số đặc trưng cơ bản.. tạo ra một tập thực thể (Entity Set). Tập tất cả nhân viên trong một cơ quan, tập các biến trong một chương trình, tập các khái niệm về CSDL.. là những tập các thực thể. Khái niệm tập thực thể là khái niệm ở mức lược đồ.

Mối quan hệ (Relationship) giữa một hay nhiều thực thể trong thế giới thực với nhau, chúng không xuất hiện vật lý hay khái niệm bên ngoài cùng với các phụ thuộc của chúng. Một xuất hiện cụ thể của một mối quan hệ được gọi là thể hiện quan hệ (Relationship Instance). Các mối quan hệ được mô tả theo độ quan hệ (Degree), tính kết nối (Connectivity) và xuất hiện (Existence). Mối quan hệ được chỉ ra bằng tính nối kết giữa các thể hiện thực thể: một-một, một-nhiều và nhiều-nhiều.

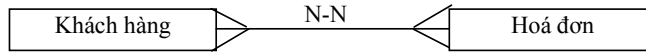
- Quan hệ một \rightarrow một (One to One): Một thực thể trong A được kết hợp với tối đa một thực thể trong B, và một thực thể trong B được kết hợp với tối đa một thực thể trong A.



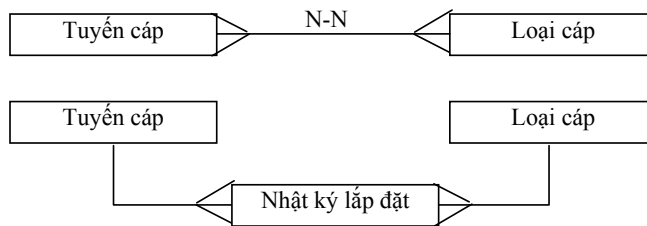
- Quan hệ một \rightarrow nhiều (One to Many). Một thực thể trong A được kết hợp với nhiều thực thể trong B, và một thực thể trong B có thể được kết hợp với tối đa với một thực thể trong A.



- Quan hệ nhiều \rightarrow 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.

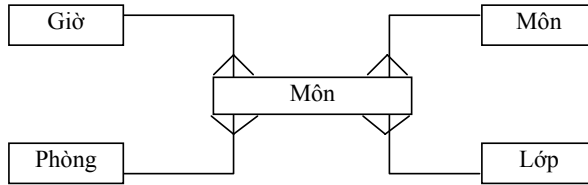


Thuộc tính (Attribute) là các tính chất đặc trưng của thực thể, chỉ ra các chi tiết cần mô tả về thực thể. Một xuất hiện cụ thể của một thuộc tính trong một thực thể hoặc một mối quan hệ được gọi là giá trị thuộc tính (Attribute Value). Thuộc tính của thực thể nhân viên là mã

nhân viên, họ và tên, địa chỉ.... Ký hiệu cho thuộc tính là một hình elip, bên trong là tên thuộc tính, được nối với thực thể.

2.6.3 Bậc của mỗi quan hệ

Bậc của mỗi quan hệ là số lượng các thực thể tham gia trong mỗi quan hệ. Các mối quan hệ hai và ba ngôi thì bậc là 2 và 3. Tổng quát mỗi quan hệ n-ngôi, bậc là n. Một thực thể có thể tham gia vào nhiều mối quan hệ và mỗi mối quan hệ có thể có bậc bất kỳ. Ngoài ra, giữa hai thực thể có thể có nhiều mối quan hệ hai ngôi, và tương tự với n thực thể bất kỳ.



2.6.4 Độ kết nối các mối quan hệ

Độ kết nối (Connectivity) của một mối quan hệ mô tả một ràng buộc trên ánh xạ giữa các xuất hiện thực thể trong mỗi quan hệ. Giá trị của độ kết nối là "một" hoặc "nhiều". Với một mối quan hệ giữa các thực thể "Phòng" và "Nhân viên", độ kết nối "một" cho "Phòng" và "nhiều" cho "Nhân viên" muốn nói rằng có tối đa "một" xuất hiện thực thể "Phòng" được liên kết với "nhiều" xuất hiện của "Nhân viên".

2.6.5 Phân cấp ISA

Nói rằng A *isa* B (A là một B) nếu tập thực thể B là sự tổng quát hóa của tập thực thể A, hoặc A là loại đặc biệt của B. Điều này có nghĩa là A có thể kế thừa các thuộc tính của B. Nói cách khác, mỗi thực thể *a* thuộc A có quan hệ với đúng một phần tử *b* trong B và như vậy *a* và *b* thực sự chỉ là một thực thể. Không cần có phần tử *b* trong B có quan hệ như thế với hai phần tử khác nhau trong A, nhưng có một vài phần tử trong B có thể không có quan hệ gì với mọi phần tử trong A. Các thuộc tính khóa của A thực sự là thuộc tính khóa của tập B và giá trị của những thuộc tính trong A được lấy tương ứng từ các thực thể trong B.

2.6.6 Sơ đồ thực thể quan hệ

Thiết kế CSDL logic có thể thực hiện bằng phương pháp tiếp cận từ trên xuống, từ dưới lên, và tổ hợp cả hai. Cách tiếp cận truyền thống đối với các CSDL quan hệ, là một quá trình từ dưới lên, tổng hợp các phần tử dữ liệu riêng biệt vào các quan hệ được chuẩn hóa sau khi đã phân tích sự phụ thuộc qua lại giữa các thực thể. Việc đưa mô hình quan hệ thực thể vào quá trình thiết kế, là cách tiếp cận từ trên xuống, một hình thái tổ hợp mới trở nên phổ biến nhờ có sử dụng các khái niệm trừu tượng hóa. Một cơ sở dữ liệu thực thể quan hệ có thể biểu diễn toàn bộ cấu trúc logic của nó bằng sơ đồ. Sơ đồ thực thể quan hệ thường đơn giản và có đặc tính rõ ràng nên dễ khái quát hóa.

Một sơ đồ thực thể quan hệ bao gồm các thành phần chính dưới đây:

- Hình chữ nhật biểu diễn các tập thực thể.
- Các hình elip biểu diễn các thuộc tính, chúng được liên kết với tập các thực thể bằng các cạnh vô hướng. Các thuộc tính là khóa sẽ được gạch chân.

Chương I: Khái niệm cơ bản về các hệ cơ sở dữ liệu

- Các hình thoi biểu diễn mối quan hệ, được liên kết với các tập thành viên.
- Những đường kẻ kết nối những thuộc tính với những tập hợp thực thể và giữa những tập hợp thực thể với những tập hợp mối quan hệ.

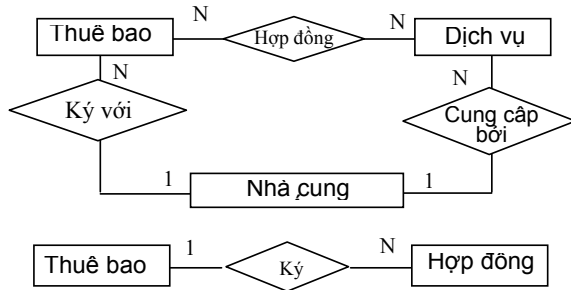
Xem xét sơ đồ thực thể quan hệ trong hình 2.4, bao gồm ba tập hợp thực thể, khách hàng – thuê bao, dịch vụ và nhà cung cấp các dịch vụ được quan hệ thông qua một tập hợp mối quan hệ nhị phân khách hàng- thuê bao được các nhà cung cấp dịch vụ cung cấp những dịch vụ gì. Gồm các bước như sau:

- Bước đầu tiên cần phải xác định các yêu cầu của bài toán và các thành phần của nó. Xác định cụ thể tập các thực thể, tập các thuộc tính và mối quan hệ Bước này đảm bảo tính độc lập và toàn vẹn của dữ liệu.
- Bước thứ 2 là mô hình hóa dữ liệu bằng các sơ đồ. Biểu diễn các thành phần của bài toán bằng các thực thể và tập các thuộc tính.
- Tích hợp các yêu cầu lại bằng sơ đồ. Biểu diễn mối quan hệ giữa các yêu cầu, tức là biểu diễn luồng thông tin lưu chuyển trong hệ thống. Loại bỏ các phụ thuộc dư thừa, thuộc tính dư thừa...
- Chuyển đổi thành các quan hệ. Xác định cấu trúc lưu trữ dưới dạng bảng.

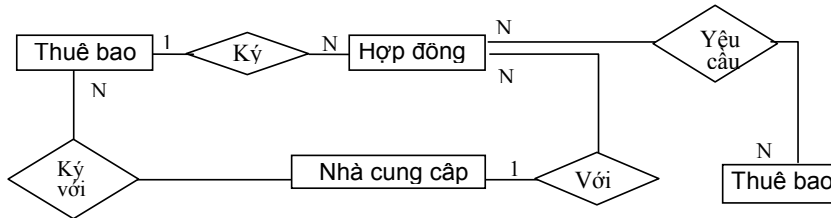
Bước I: Các yêu cầu



Bước II: Mô hình hóa ER



Bước III: Tích hợp các yêu cầu



Bước IV: Chuyển đổi sang quan hệ

THUEBAO

Mã	Họ và tên

DICHVU

MADV	Dịch vụ

NHACCDV

MACC	Tên cơ quan	Địa chỉ

CREATE TABLE THUEBAO

(MA integer,
HOVATEN char (15),
DC char(30),
MACC char(15),
DICHVU integer,
primary key (MA),
foreign key(MACC) references NHACC,
foreign key(MA) referenes DICHVU;

Bước V: Chuẩn hóa dữ liệu (3NF, BCNF, 4NF, 5NF)

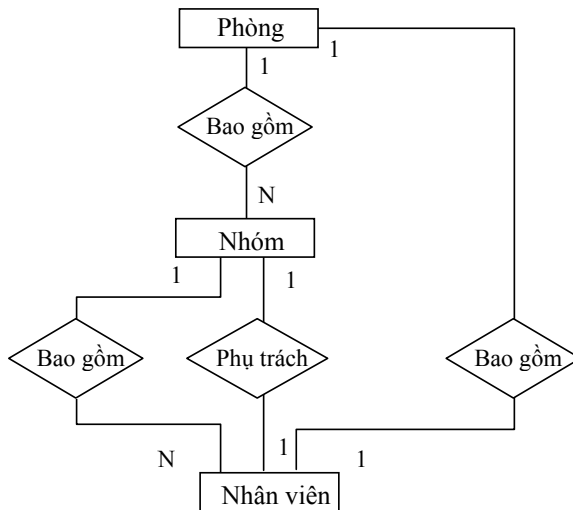
Bước VI : Thiết kế vật lý

Hình 2.4 Các bước thiết kế một mô hình thực thể quan hệ

Ví dụ Mô hình thực thể quan hệ về quản lý nhân sự & quản lý dự án

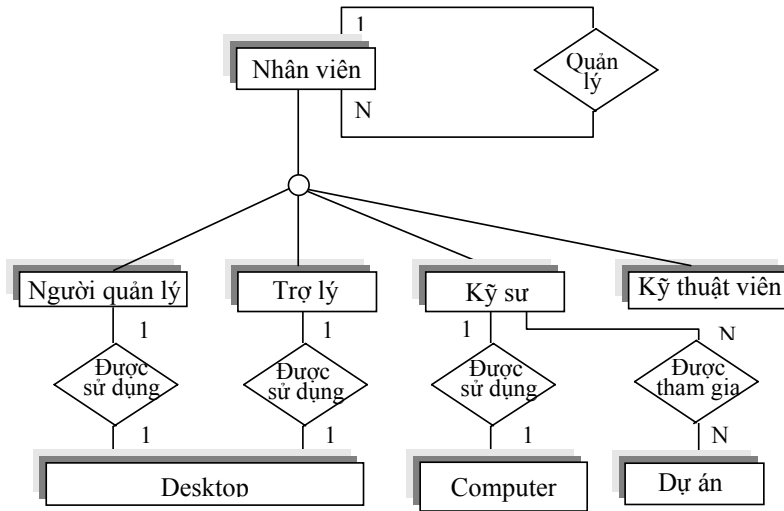
Sau khi khảo sát quản lý nhân sự trong một cơ quan, có các nhận xét như sau:

- Một phòng làm việc gồm nhiều nhóm làm việc và quản lý nhiều nhân viên
- Trong mỗi nhóm bao gồm nhiều nhân viên làm việc và có một người phụ trách.
- Mỗi nhân viên thuộc một nhóm làm việc hoặc thuộc một phòng
- Nhân viên là một người phụ trách.



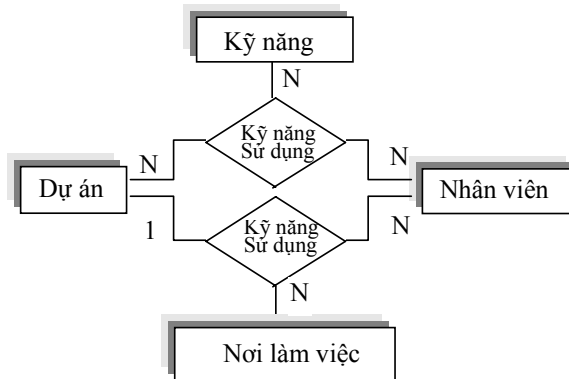
Hình 2.5 Mô hình hóa thực thể theo quan điểm quản lý

- Chức danh nhân viên: Kỹ sư, kỹ thuật viên, trợ lý, người quản lý...
- Kỹ sư được cấp máy tính làm việc, trợ lý và kỹ thuật viên được cấp máy tính để bàn.
- Mỗi một kỹ sư có thể tham gia nhiều dự án khác nhau.



Hình 2.6 Mô hình hóa thực thể quan hệ theo quan điểm nghề nghiệp công việc

- Kỹ năng có thể có nhiều lựa chọn cho một dự án và ngược lại.
- Kỹ năng có thể có nhiều nhân viên tham gia và ngược lại
- Mỗi một dự án có thể có nhiều văn phòng khác nhau.
- Một nhân viên có thể có nhiều nơi làm việc và một nơi coa thể có nhiều nhân viên.



Hình 2.7 Mô hình hóa thực thể quan hệ theo quan điểm nhân sự

Lược đồ toàn cục: Lược đồ toàn cục là lược đồ tích hợp các mô hình đã thiết kế theo quan điểm quản lý, nghề nghiệp và quản lý nhân sự...Lược đồ toàn cục là cơ sở để phát triển về

các quan hệ thành các dạng chuẩn và cấu trúc lưu trữ vật lý. Mỗi một mối quan hệ trong lược đồ toàn cục đều dựa trên một nhận định có thể xác nhận được về dữ liệu thực của cơ quan và sự phân tích các nhận xét để dẫn đến việc biến đổi cấu trúc mô hình ER thực thể quan hệ thành các bảng quan hệ đã được chuẩn hoá.

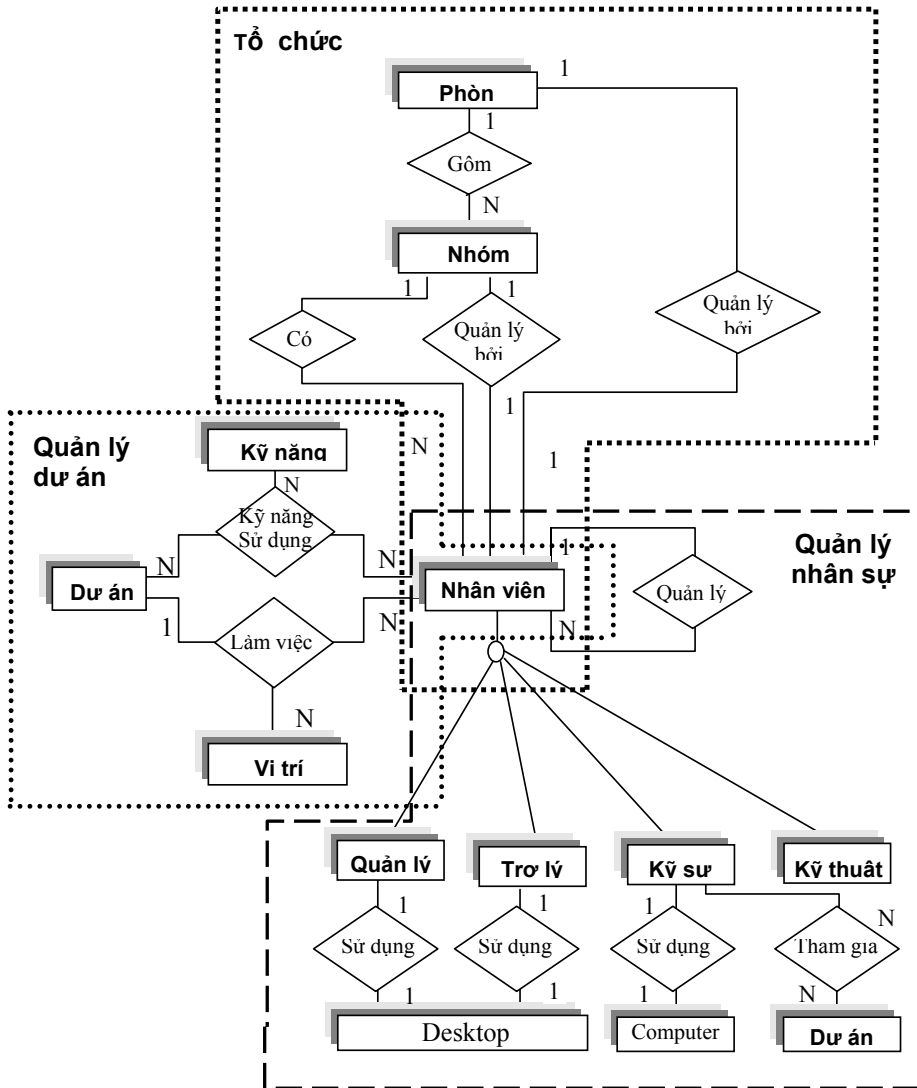
2.6.7 Các bước triển khai mô hình thực thể quan hệ

Phân tích yêu cầu: Phân tích yêu cầu là bước quan trọng của vòng đời CSDL. Người thiết kế phải khảo sát, phỏng vấn ... nhằm xác định CSDL đáp ứng được gì và cần cái. Mục tiêu cơ bản của bước này là:

- Xác định rõ yêu cầu của từng bộ phận, khách quan trung thực. Phân loại thực thể, thuộc tính.
 - Mô tả thông tin về các đối tượng và xác định mối quan hệ giữa các đối tượng cần thiết kế. Nội dung thực thể.
 - Xác định các loại giao dịch trên CSDL. Tương tác giữa các giao dịch.
 - Xác định các ràng buộc toàn vẹn, tính bảo mật để áp đặt lên CSDL.
 - Xác định phần cứng hệ thống, điều kiện cài đặt.
 - Tài liệu khảo sát.
2. *Xác định nội dung, yêu cầu của các thực thể*
- Tổng quát hoá các thực thể và định danh các thuộc tính của các thực thể.
 - Xác định nội dung thông tin của các thực thể, các thuộc tính đa trị,.
 - Đặc tả dữ liệu cần xử lý, mối quan hệ tự nhiên giữa các dữ liệu.
 - Xác định mối quan hệ giữa các thuộc tính.
 - Mô hình hoá thực thể.
3. *Tích hợp các yêu cầu- tích hợp các mô hình thực thể.*
- Lựa chọn chiến lược tích hợp.
 - Xác định các thực thể tương ứng, đồng nghĩa
 - Phát hiện xung đột cấu trúc: kiểu, phụ thuộc dư thừa, xung đột khóa...
 - Tổng quát hóa quá trình tích hợp.
4. *Chuyển đổi thành các bảng quan hệ*
- Sự tiến hóa tự nhiên từ mô hình thực thể quan hệ sang lược đồ quan hệ.
 - Quy tắc biến đổi:
 - √ Một bảng thực thể có nội dung thông tin như thực thể gốc.
 - √ Một bảng chứa khóa ngoại lai của thực thể cha.
 - √ Một bảng chứa khóa ngoại lai của tất cả các thực thể trong mối quan hệ.
 - Các bước biến đổi:
 - √ Biến đổi thực thể.
 - √ Biến đổi các mối quan hệ nhiều- nhiều.
5. *Chuẩn hoá quan hệ*
- Chuyển về dạng chuẩn 1NF
-

- Chuyển về dạng chuẩn 2NF
- Chuyển về dạng chuẩn 3NF
- Chuyển về dạng chuẩn BCNF
- Chuyển về dạng chuẩn 4NF
- Chuyển về dạng chuẩn 5NF

6. Thiết kế vật lý



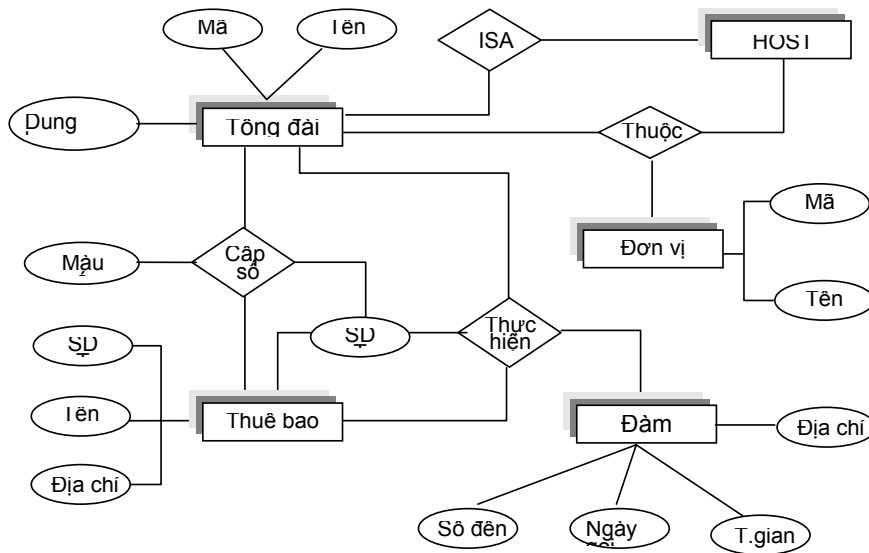
Hình 2.8 Lược đồ toàn cục quản lý nhân sự – dự án

Ví dụ Mô hình thực thể quan hệ cơ sở dữ liệu cước thuê bao điện thoại

- Mạng gồm nhiều tổng đài bao gồm nhiều HOST và các tổng đài vệ tinh.
- Các thuộc tính tổng đài bao gồm mã tổng đài, dung lượng, màu cấp.
- Đơn vị quản lý tổng đài: tên đơn vị, địa chỉ,...
- Ứng với một màu cấp là một số điện thoại cấp cho thuê bao.
- Thuê bao gồm nhiều thông tin: Mã thuê bao, địa chỉ, tên thuê bao...

Mô hình bao gồm tập hợp các thực thể: *khách hàng- thuê bao, cước đàm thoại, tập tổng đài và đơn vị quản lý tổng đài- nhà cung cấp dịch vụ*. Các tập thực thể này được liên kết với nhau bằng một tập hợp mối quan hệ. **nhị phân**: Mỗi một số điện thoại của thuê bao tương ứng một - một mã số đầu dây gồm có các thuộc tính màu cấp,... SĐT (số điện thoại) là thuộc tính của thực thể thuê bao với các thuộc tính khác: họ tên thuê bao, địa chỉ thuê bao. SĐT của thuê bao thực hiện cuộc đàm thoại với số điện thoại gọi đến trong thời gian cụ thể từ giờ bắt đầu, kết thúc...Màu cấp là thuộc tính của tổng đài và tổng đài là tổng đài vệ tinh của một trạm HOST thuộc một đơn vị quản lý nào đó gồm các thuộc tính: tên cơ quan, mã cơ quan, địa chỉ cơ quan.

Tập mối quan hệ tổng đài có thể có thuê bao. Một thuê bao có thể có nhiều tổng đài, nhiều nhà cung cấp dịch vụ và một tổng đài- nhà cung cấp có thể cung cấp cho nhiều thuê bao.

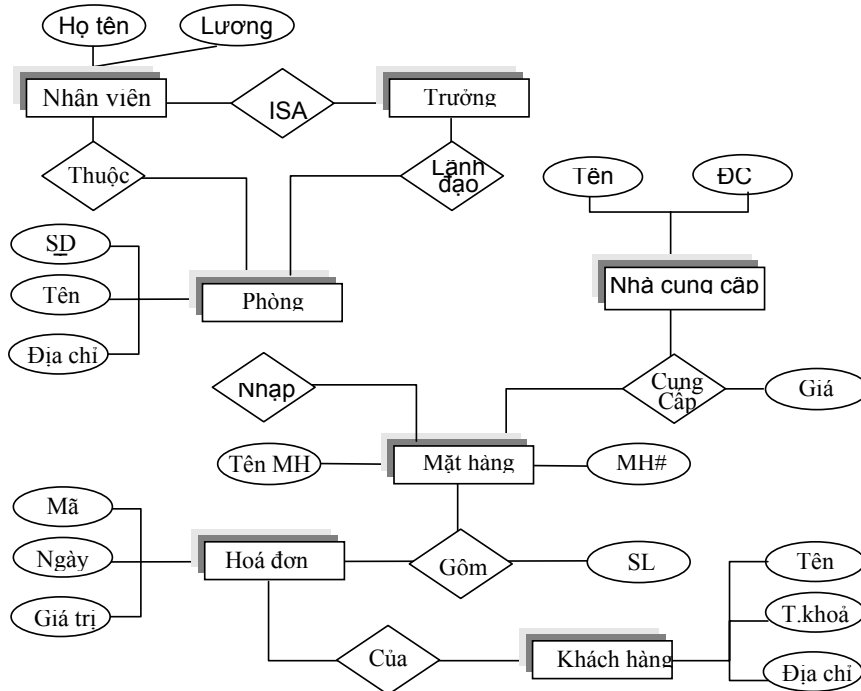


Hình 2.9 Lược đồ toàn cục cước thuê bao điện thoại

Ví dụ Sơ đồ cơ sở dữ liệu siêu thị: Giả sử trong một siêu thị có nhiều gian hàng. Trong mỗi một gian hàng có bán một số mặt hàng và trong từng gian hàng có nhiều nhân viên bán hàng. Trong số nhân viên có một người phụ trách - gian hàng trưởng. Nhân viên và của

hàng trường chỉ làm việc trong một gian hàng. Siêu thị có nhiều nhà cung cấp hàng hoá và có các khách mua hàng. Như vậy:

- Tập thực thể nhà cung cấp (SUPPLIERS) có các thuộc tính như tên nhà cung cấp (SNAME), địa chỉ (SADDR). Mỗi nhà cung cấp có thể cung cấp một mặt hàng (ITEM), có đơn giá (PRICE) của riêng họ.
- Tập thực thể mặt hàng với các thuộc tính mã mặt hàng (ITEM#), và tên mặt hàng (INAME). Một mặt hàng có thể có nhiều nhà cung cấp và nhiều nhà cung cấp có thể cung cấp nhiều mặt hàng khác nhau. Như vậy mối liên kết nhà cung cấp – mặt hàng là mối quan hệ nhiều – nhiều. Một mặt hàng chỉ được bán trong một gian hàng.
- Tập thực thể nhân viên bán hàng (EMPLOYEES) gồm các thuộc tính như họ và tên (NAME), mức lương (SALARY), mã nhân viên (EMP#). Nhân viên có thể là trưởng gian hàng, người quản lý (MANAGE), sử dụng ISA.
- Tập thực thể các hoá đơn mua hàng (ORDERS) gồm các thuộc tính: số hoá đơn (O#), ngày đặt mua (DATE), Mỗi một hoá đơn của khách mua hàng là một hoá đơn của một khách đặt mua và có thể mua nhiều mặt hàng với số lượng tương ứng họ đặt mua (QUANTITY).
- Tập thực thể khách mua hàng (CUSTOMERS) gồm các thuộc tính họ và tên (CNAME), địa chỉ (CADDR), và số dư tài khoản (BALANCE) của họ.



Hình 2.10 Lược đồ toàn cục cơ sở dữ liệu siêu thị

Câu hỏi trắc nghiệm

1. Nghiên cứu mô hình cơ sở dữ liệu dựa trên các yêu cầu...
 - A. Mục tiêu độc lập dữ liệu và trao đổi
 - B. Phải xác định rõ ràng các khía cạnh logic và khía cạnh
 - C. Quản trị cơ sở dữ liệu
 - D. Mục tiêu xử lý tệp
 - E. Mô hình được xây dựng trên cơ sở lý thuyết vững chắc, chặt chẽ.
 2. Đặc trưng của một mô hình dữ liệu.....
 - A. Tính ổn định và tính đơn giản
 - B. Tính dư thừa cần phải kiểm tra .
 - C. Tính đối xứng
 - D. Có cơ sở lý thuyết vững chắc.
 3. Một mô hình CSDL là tốt nhất nếu:
 - A. Cài đặt trong một mô hình dữ liệu với một hệ quản trị cơ sở dữ liệu nào đó.
 - B. Đặc tính nhận dạng hướng đối tượng.
 - C. Tính dư thừa
 - D. Giải quyết mối quan hệ nhiều – nhiều
 4. Cấu trúc CSDL phân cấp biểu diễn dữ liệu bằng...
 - A. Cấu trúc quan hệ.
 - B. Cấu trúc cây
 - C. Cấu trúc bảng
 - D. Cấu trúc mạng
 5. Trong cấu trúc lưu trữ của mô hình phân cấp rất hạn chế khi thực hiện việc
 - A. Tìm kiếm thông tin.
 - B. Xử lý thông tin.
 - C. Tổ chức lưu trữ thông tin
 - D. Tính ổn định thông tin
 6. Mô hình dữ liệu mạng là mô hình
 - A. Thực thể quan hệ
 - B. Một - một và nhiều – một.
 - C. Quan hệ.
 - D. Nhiều - nhiều.
 7. Mô hình CSDL mạng là mô hình
 - A. Đối xứng.
 - B. Không đối xứng
 - C. Không đảm bảo sự nhất quán của dữ liệu và tính toàn vẹn của dữ liệu.
 - D. Các bản ghi được kết nối với nhau bằng các xuất hiện kiểu bản ghi liên kết.
 8. Mô hình CSDL quan hệ là mô hình
 - A. Đối xứng.
 - B. Không đối xứng
-
-

- C. Không đảm bảo sự nhất quán của dữ liệu và tính toàn vẹn của dữ liệu.
D. Các bản ghi được kết nối với nhau bằng các xuất hiện kiểu bản ghi liên kết.
9. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Cơ sở dữ liệu quan hệ có thể hiểu là dữ liệu được người sử dụng nhìn dưới dạng mộtquan hệ toán học và các phép toán thao tác dữ liệu được xây dựng trên các cấu trúc quan hệ toán học.
- A. Quan hệ.
B. Biểu thức đại số.
C. Biểu thức.
D. Phụ thuộc
10. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Ngôn ngữ con dữ liệu kiểu quan hệ là tập các phép toán có cácthao tác trên các dữ liệu dưới dạng quan hệ. Dữ liệu được biểu diễn một cách duy nhất. Ngôn ngữ con dữ liệu gồm nhóm các các phép toán tìm kiếm và nhóm các phép toán cập nhật, lưu trữ.
- A. Cấu trúc quan hệ.
B. Cấu trúc dữ liệu.
C. Biểu thức quan hệ.
D. Quan hệ
11. Hãy so khớp thông tin 2 phần dưới đây với nhau:

Phần 1

- 1 Một quan hệ là một bảng và mỗi bảng.....
2 Cơ sở dữ liệu quan hệ được
3 Cấu trúc dữ liệu quan hệ.....
4 Giữa các thuộc tính có mối quan hệ ràng buộc.....
- a. 1- D; 2- C ; 3 – B ; 4 – A .
b. 1- C; 2- B; 3 – A; 4 – D
c. 1- A; 2- B; 3 – C; 4 – D.
d. 1- D; 2- B; 3 – A; 4 – C.

Phần 2

- A. phụ thuộc lẫn nhau và các phụ thuộc cũng là các thực thể đặc biệt
B. là mối liên kết giữa các bộ được biểu diễn duy nhất bằng các giá trị dữ liệu .
C. biểu diễn dưới dạng các bảng
D. là một cấu trúc quan hệ toán học.
12. Hãy so khớp thông tin 2 phần dưới đây với nhau:

Phần 1

- 1 Kết quả của các phép tìm kiếm trong CSDL quan hệ dữ liệu
2 Các câu hỏi tìm kiếm trong cơ sở dữ liệu
3 Các phép toán lưu trữ trong CSDL quan hệ có khả năng

Phần 2

- A. là đối xứng nhau.
B. cũng là một quan hệ.
C. xây dựng trên đại số quan hệ.
-
-

- 4 Các phép toán lưu trữ trong CSDL quan hệ D. xử lý theo lô. Các quan hệ được xem được như toán hạng. .
- 1- D; 2- C ; 3 – B ; 4 – A .
 - 1- B ; 2- A; 3 – D ; 4 – C
 - 1- A; 2- B; 3 – C; 4 – D.
 - 1- D; 2- B; 3 – A; 4 – C.

Câu hỏi và bài tập

- Giả sử trong nghiệp vụ quản lý phát hành báo chí, thông tin gồm có:
 - Thông tin về khách hàng đặt mua báo (tạp chí) :*
 - Mã khách hàng đặt mua, 5 ký tự, kiểu character.
 - Họ và tên khách hàng, 21 ký tự, kiểu character.
 - Địa chỉ khách hàng, 21 ký tự, kiểu character.
 - Số điện thoại , 7 ký tự, kiểu character.
 - Địa điểm giao nhận báo(tạp chí) hàng ngày, 25 ký tự, kiểu character.
 - Thông tin về các loại báo (tạp chí) bao gồm:*
 - Mã báo khách đặt mua., 3 ký tự, kiểu character.
 - Tên báo (tạp chí), 15 ký tự, kiểu character.
 - Giá báo, 5 số.
 - Kỳ phát hành báo (báo ngày, báo tuần, báo tháng...). 3 ký tự chữ
 - Thông tin về phiếu đặt báo gồm*
 - Ngày khách đặt báo, chí.
 - Mã hiệu khách hàng đặt mua báo.
 - Mã báo khách đặt mua.
 - Số lượng báo, chí khách đặt mua .
 - Thành tiền từng loại báo.
 - a) Hãy phát hoạ mô hình CSDL quan hệ với các dữ liệu trên.
b) Hãy phát hoạ mô hình CSDL mạng .
c) Hãy phát hoạ mô hình CSDL phân cấp.
 - Khi thực hiện các phép lưu trữ (chèn thêm, loại bỏ và sửa đổi) trên các dữ liệu được biểu diễn trong các mô hình CSDL quan hệ, mạng và phân cấp thì dị thường thông tin có xảy ra không ? Vì sao.
 - Chứng tỏ rằng mô hình CSDL mạng với các dữ liệu trên là mô hình dữ liệu phức tạp nhất, vì sao ?
 - Tại sao nói việc xây dựng ngôn ngữ dữ liệu thao tác trên CSDL mạng lại khó khăn, phức tạp hơn nhiều so với mô hình cơ sở dữ liệu quan hệ. Hãy cho thí dụ về tìm kiếm dữ liệu trên cơ sở dữ liệu mạng.
 - Ứng với các mô hình dữ liệu trên, hãy nhận xét kết quả các phép tìm kiếm:
-
-

- a) Thông tin về tất cả khách hàng đặt mua báo NDA (Nhân Dân) trong quý I/2000.
 - b) Thông tin về các loại báo mà khách hàng đặt mua trong quý I/2000.
5. Hãy thực hiện các phép tìm kiếm trên CSDL quan hệ:
- a) Tổng số lượng và tiền các loại báo (tạp chí) khách đặt mua trong tháng năm 1999.
 - b) Thông tin về tất cả khách hàng đặt mua tất cả các loại báo có trong danh mục.
 - c) Lập bảng kê tính tổng tiền các loại báo mà khách đặt mua trong từng tháng?
 - d) Thông tin về các loại báo khách đặt mua có số lượng và tiền cao hơn số lượng và tiền khách đặt mua báo “VHT” văn hoá & thể thao?
5. Hãy sử dụng mô hình thực thể quan hệ để mô tả dữ liệu về Học viên (bao gồm về mô hình tổ chức, các phòng ban, các khoa, quản lý sinh viên quản lý đào tạo, học bổng, các danh hiệu thi đua...).
6. Hãy vẽ sơ đồ mmo hình thực thể quan hệ về mối quan hệ trong gia đình: bố mẹ, vợ chồng, con cái... và chuyển về các mô hình:
- a) Mô hình quan hệ.
 - b) Mô hình phân cấp.
 - c) Mô hình mạng.
6. Giả sử cơ sở dữ liệu quản lý sinh viên gồm các thực thể sinh viên (students), khoá học (courses) và điểm học tập (grade) của sinh viên trong mỗi khoá học. Thực thể sinh viên bao gồm các thuộc tính như họ và tên, địa chỉ, ngày sinh, quê quán... Thông tin về các khoá học bao gồm tên khoá học, khoa nào tổ chức, thời gian, số tiết, nơi học, thầy dạy... Có thể biểu diễn lược đồ quan hệ trên bằng nhiều mô hình khác nhau. Tuy nhiên có thể có một số sơ đồ có một số đặc điểm không mong muốn. Ví dụ như:
- a) Cho trước một SV, không thể xác định những khoá học mà sinh viên đó đã học qua mà không cần phải duyệt nhiều phần của CSDL.
 - b) Không thể xác định những đã đăng ký một lớp học mà không cần phải duyệt nhiều phần của CSDL.
 - c) Không thể xác định điểm của một SV trong một khoá học.
 - d) Tạo dư thừa: khoá học lặp lại nhiều lần, tên lặp lại.....
- Hãy chỉ ra các hạn chế khi chuyển đổi CSDL trên về :
- a) Quan hệ
 - b) Phân cấp
 - c) Mạng

3 CƠ SỞ DỮ LIỆU QUAN HỆ

Edgar F.Codd là người đầu tiên nghiên cứu mô hình cơ sở dữ liệu quan hệ. Codd đã có nhiều đóng góp cho sự phát triển đại số quan hệ, các phép toán quan hệ và sự chuẩn hóa quan hệ. Trong chương này sẽ trình bày những khái niệm cơ bản nhất về lý thuyết cơ sở dữ liệu quan hệ do E.F Codd đề xuất, đó là các khái niệm về quan hệ, về khóa của lược đồ quan hệ. Những khái niệm này có vai trò quan trọng trong việc thiết kế và cài đặt các hệ cơ sở dữ liệu quan hệ và các hệ quản trị cơ sở dữ liệu. Nội dung của chương bao gồm:

- Định nghĩa quan hệ
- Lược đồ quan hệ
- Khóa của lược đồ quan hệ
- Một số tính chất của khoá trong lược đồ quan hệ
- Các thuật toán xác định khoá của lược đồ quan hệ
- Các phép toán trên cơ sở dữ liệu quan hệ

3.1 Quan hệ

3.1.1 Định nghĩa quan hệ

Cho $\Omega := \{A_1, A_2, \dots, A_n\}$ một tập hữu hạn các thuộc tính, khác rỗng và các phần tử không nhất thiết phân biệt nhau. Mỗi một thuộc tính $A_i \in \Omega$, $i=1 \div n$ có một miền giá trị xác định, (Domain), ký hiệu $D(A_i) = \text{Dom}(A_i)$. Ví dụ, tập các ký tự số có chiều dài chính xác là 7, là miền của của thuộc tính số điện thoại. Tích Đề các của các miền $\text{Dom}(A_1), \text{Dom}(A_2), \dots, \text{Dom}(A_n)$ được biểu diễn như sau:

$$\text{Dom}(A_1) \times \text{Dom}(A_2) \times \dots \times \text{Dom}(A_n) := \{ (a_1, a_2, \dots, a_n) \mid a_i \in A_i, i = 1 \div n \}.$$

Nói rằng R là một *quan hệ* trên tập Ω khi và chỉ khi nó là một tập con của tích Đề các của: $D(a_1) \times D(a_2) \times \dots \times D(a_n)$. Ký hiệu $R(\Omega) = R(A_1, A_2, \dots, A_n)$. Tức là

$$R(\Omega) \subseteq D(a_1) \times D(a_2) \times \dots \times D(a_n).$$

Như vậy quan hệ $R(\Omega)$ trên tập Ω là tập các các hàm:

$$R(\Omega) := \{ r \mid r: \Omega \rightarrow \cup D(A), A \in \Omega \}$$

Mỗi một hàm trong quan hệ được gọi là một dòng của quan hệ. Vì vậy trong cách tiếp cận mô hình CSDL quan hệ, người ta thường biểu diễn một quan hệ trên tập các thuộc tính Ω bằng một bảng 2 chiều. Các thuộc tính (Attributes) của quan hệ là các cột A_1, A_2, \dots, A_n , không nhất thiết phân biệt nhau và các hàng của quan hệ được gọi là các bộ (tuple), hay còn được gọi là các bản ghi của quan hệ, ký hiệu là $r \in R(\Omega)$. Giá trị của mỗi một bộ gồm n thành phần giá trị $r = (d_1, d_2, \dots, d_n)$ được rút ra giá trị từ các miền xác định tương ứng, tức là $r(A_1) = d_1 \in D(A_1)$, $r(A_2) = d_2 \in D(A_2)$, \dots , $r(A_n) = d_n \in D(A_n)$, trong đó các ký hiệu $D(A_i) := \text{Dom}(A_i)$ là miền xác định của $A_i \in \Omega$, $i = 1 \div n$.

Có thể định nghĩa quan hệ $R(\Omega)$ cách khác như sau :

$$R(\Omega) = \{(d_1, d_2, \dots, d_n) \mid d_i \in D(A_i), A_i \in \Omega, i=1 \div n \}.$$

	A_1	A_2	A_n
r_1	$r_1(A_1)$	$r_1(A_2)$	$r_1(A_n)$
r_2	$r_2(A_1)$	$r_2(A_2)$	$r_2(A_n)$

r_n	$r_n(A_1)$	$r_n(A_2)$	$r_n(A_n)$

Hình 3.1 Quan hệ biểu diễn thành bảng

3.1.2 Ký hiệu

- **Bậc** của quan hệ: Số thuộc tính của quan hệ $R(\Omega)$.
- **Số n_bộ** của quan hệ gọi là lực lượng của quan hệ. Ký hiệu là $|R|$.
- **n_bộ** : (d_1, d_2, \dots, d_n) là một bản ghi, thường được biểu diễn bằng các chữ La tinh nhỏ, ví dụ $r := (d_1, d_2, \dots, d_n) \in R(\Omega)$.
- X là một tập con các thuộc tính, ký hiệu $X \subseteq \Omega$, khi và chỉ khi $\forall A \in X$, suy ra $A \in \Omega$. Nghĩa là các thuộc tính của X cũng là thuộc tính của Ω . Tổng quát $X \subseteq Y$ khi và chỉ khi với mọi thuộc tính của X cũng là thuộc tính của Y .
- Nếu $\forall X \subseteq \Omega$ và $\forall r \in R(\Omega)$. Khi đó phép chiếu X trên bộ r được hiểu như sau: Nếu $r := (d_1, d_2, \dots, d_n)$ khi đó $r[X] := r(X) := (d_{i_1}, d_{i_2}, \dots, d_{i_k})$, tức là $r[X]$ là các giá trị của bộ r chứa giá trị của thuộc tính X .
- Cho hai bộ $\forall r_1, r_2 \in R(\Omega)$, $r_1 = (d_1, d_2, \dots, d_n)$ và $r_2 = (h_1, h_2, \dots, h_n)$.

Khi đó nói rằng r_1 *trùng* r_2 được hiểu là : $r_1 \equiv r_2 \Leftrightarrow d_i = h_i, i=1 \dots n$.

r_1 *khác* r_2 : $r_1 \neq r_2 \Leftrightarrow \exists i \in \{1, 2, \dots, n\}$ sao cho $d_i \neq h_i$.

Ví dụ 1: Quan hệ về khách hàng đặt mua báo: KHACHHANG (M#,TK,DC,SĐT).

Trong đó: M# ký hiệu là mã khách đặt mua báo, TK là tên khách, DC địa chỉ khách hàng và SĐT là số điện thoại của khách. Miền giá trị của M# là tập hợp (M01, M02, M13, M13), miền giá trị của TK là họ và tên của khách đặt mua báo, Quan hệ KHACHHANG là một bảng 2 chiều, các cột là các thuộc tính M#, TK, DC và SĐT, các hàng của bảng là thông tin về các khách hàng đặt mua báo. Quan hệ là tập các bản ghi gồm các thành phần mã khách, họ tên khách, địa chỉ khách và số điện thoại khách. Ví dụ (M01, Nguyễn Ngọc An, 24 Nguyễn Biều, 822134)

M#	TK	DC	SĐT
M01	Nguyễn Ngọc An	24 Nguyễn Biều	8222134
M02	Hồ Ngọc Hà	174 Tôn Đức Thắng	
M12	Tô Ngọc Long	54 Tràng Thi	5241234
M13	Phan Đăng Cầu	134 Đội Cấn	8345123

Chương I: Khái niệm cơ bản về các hệ cơ sở dữ liệu

Quan hệ BAO (MB#,TBAO,GIA,KY,NXB) là bảng thông tin về các loại báo:

MB# Mã báo, TBAO Tên khách hàng, GIA Giá báo.
KY Kỳ phát hành, NXB Nhà in, xuất bản.

MB#	TBAO	GIA	KY	NXB
A01	Nhân Dân	1.000	NGAY	Nhân Dân
A02	Quân đội Nhân Dân	1.000	NGAY	Quân Đội
B01	Hà Nội Mới	600	NGAY	Tiến Bộ

Quan hệ DAT_MUA là bảng biểu diễn thông tin về các phiếu đặt mua báo, tạp chí của khách hàng: DAT_MUA (M#,MB#,NG,SL)

M# Mã Khách MB# Mã báo,
NG Ngày đặt mua SL Số lượng

M#	MB#	NG	SL
M01	A01	1/2/2000	100
M01	A02	4/2/2000	150
M02	A01	1/3/2000	300
M02	A02	1/4/2000	120
M12	A01	1/6/2000	230
M12	A03	1/5/2000	50
M12	B01	1/4/2000	90
M13	A02	1/4/2000	150

Hình 3.2 Ví dụ các quan hệ quản lý phát hành báo chí

3.2 Phụ thuộc hàm

Phụ thuộc hàm có tầm quan trọng đối với người quản trị cơ sở dữ liệu trong việc thiết kế và cài đặt các mô hình cơ sở dữ liệu quan hệ. Cơ sở lý thuyết về chuẩn hoá dữ liệu dựa trên các khái niệm phụ thuộc hàm và khoá của quan hệ. Phụ thuộc hàm là khái niệm được xây dựng để mô tả các ràng buộc trong cơ sở dữ liệu. Nói rằng mã mặt hàng xác định số lượng, đơn giá, ngày nhập kho... của một mặt hàng. Nói cách khác, mỗi một giá trị của thuộc tính mã mặt hàng xác định duy nhất giá trị của thuộc tính số lượng, đơn giá ... của mặt hàng. Ràng buộc này sẽ từ chối khi chèn thêm thông tin về một mặt hàng mới mà chưa được xác định mã mặt hàng, vì sẽ mâu thuẫn, không nhất quán trong tổ chức lưu trữ dữ liệu.

Có 2 loại phụ thuộc hàm:

1. Ràng buộc giải tích: Giữa một số thuộc tính có sự ràng buộc bằng các biểu thức toán học. Điểm trung bình dưới 5 là học sinh có học lực “kém”, từ 5 đến 6,9 có học lực “trung bình”, từ 7 đến 8,9 học lực “khá” và trên 8 là học lực “giỏi”.
2. Ràng buộc logic là các mối quan hệ giữa các thuộc tính với nhau, nhưng không phải là các ràng buộc giải tích, được gọi là phụ thuộc hàm.

Định nghĩa: Cho R là một quan hệ trên tập Ω và cho X và Y là 2 tập con bất kỳ của Ω . Nói rằng X xác định Y hay Y phụ thuộc hàm vào X , ký hiệu $f: X \rightarrow Y$, khi và chỉ khi nếu 2 bộ bất kỳ r và s của quan hệ R : $(\forall r, s \in R) (r(X) = s(X))$ thì suy ra $r(Y) = s(Y)$, Hay $(\forall r, s \in R) ((\forall a \in X) (r(a) = s(a)))$ thì suy ra $(\forall b \in Y) (r(b) = s(b))$.

Nói cách khác, khi đối số trùng nhau thì hàm có cùng giá trị. Một giá trị của Y được xác định bởi một giá trị của X .

Ký hiệu $F := \{f: L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$ là tập các phụ thuộc hàm trên các thuộc tính Ω .

3.3 Hệ tiên đề các phụ thuộc hàm và các phép suy dẫn logic

Họ đầy đủ các phụ thuộc hàm theo định nghĩa $F := \{f: L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$ chỉ mới thỏa trên một quan hệ $R(\Omega)$. Câu hỏi đặt ra, liệu các phụ thuộc của F có thỏa trong mọi quan hệ trên Ω hay không? Năm 1974 Armstrong đã đưa ra 4 tiên đề đặc trưng cho tập các phụ thuộc hàm của File dữ liệu.

3.3.1 Hệ tiên đề Armstrong cho các phụ thuộc hàm

Cho $\Omega := \{A_1, A_2, \dots, A_n\}$ là tập khác rỗng. Gọi F là tập các phụ thuộc hàm thỏa trên các quan hệ R trên tập các thuộc tính Ω . Ký hiệu $Y := \{(A, B) \mid A, B \subseteq \Omega, A \rightarrow B \in F\}$. Hiển nhiên Y là một họ f . Khi đó nếu $\forall A, B, C, D \subseteq \Omega$:

A1: *Phản xạ:* Nếu với mọi $B \subseteq A \Rightarrow A \rightarrow B$. Quy tắc A1 đưa ra những phụ thuộc không tầm thường, là những phụ thuộc mà vế phải được chứa trong vế trái. Những phụ thuộc loại này luôn luôn đúng trong mọi quan hệ, phụ thuộc vào Ω , không phụ thuộc vào tập các phụ thuộc hàm F .

A2: *Gia tăng:* Nếu $A \rightarrow B \Rightarrow AC \rightarrow B$, $AC \rightarrow BC$. Quy tắc này chỉ ra rằng có thể mở rộng vế trái hoặc cả hai vế phụ thuộc hàm cùng một thuộc tính. Chú ý không cho phép thêm vào vế phải. Trong đó $AC = A \cup C$.

A3: *Bắc cầu:* Nếu $A \rightarrow B$ và $B \rightarrow C$ thì suy ra $A \rightarrow C$. Nếu một thuộc tính xác định thuộc tính thứ hai, và nó xác định thuộc tính thứ ba, khi đó thuộc tính thứ nhất xác định thuộc tính thứ 3.

A4: *Giả bắc cầu:* Nếu $A \rightarrow B$ và $BC \rightarrow Z \Rightarrow AC \rightarrow Z$. Áp dụng A2 và A3 có thể suy ra A4: $A \rightarrow B \Rightarrow AC \rightarrow BC$ (A2), $BC \rightarrow Z \Rightarrow AC \rightarrow Z$ (A3).

A5: *Hợp:* Nếu $A \rightarrow B$ và $A \rightarrow C \Rightarrow A \rightarrow BC$. Áp dụng A2: $A \rightarrow B \Rightarrow AA \rightarrow AB$ và $A \rightarrow C \Rightarrow AB \rightarrow BC$. Áp dụng A3: $AA \rightarrow C$, tức là $A \rightarrow BC$.

A6: *Tách:* Nếu $A \rightarrow BC \Leftrightarrow A \rightarrow B$ và $A \rightarrow C$. Nghĩa là nếu vế phải bao gồm nhiều thuộc tính, khi đó thuộc tính vế trái sẽ xác định các thành phần trong vế phải. Tiên đề được suy dẫn từ các tiên đề A1, A2 và A3 như sau: $A \rightarrow BC \Leftrightarrow AA \rightarrow ABC \Leftrightarrow A \rightarrow ABC$, áp dụng quy tắc A1: $ABC \rightarrow B$ và $ABC \rightarrow C \Leftrightarrow A \rightarrow B$ và $A \rightarrow C$.

3.3.2 Ý nghĩa hệ tiên đề Armstrong

1. Có thể nhận thấy rằng nếu $F := \{L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$ là tập các phụ thuộc hàm thỏa trên quan hệ R thì nó cũng thỏa trên mọi quan hệ trên tập các thuộc tính Ω .

2. Và ngược lại với một tập Y họ f các phụ thuộc hàm, khi đó tồn tại một quan hệ R trên tập các thuộc tính Ω sao cho các phụ thuộc hàm thỏa trên nó. Tức là $F := \{(A,B) \mid A,B \subseteq \Omega \ \& \ A \rightarrow B\}$. Điều này có nghĩa là các hệ tiên đề Armstrong là *đúng và đầy đủ*. Nói cách khác các tiên đề là đặc trưng các họ phụ thuộc hàm.
3. Như vậy phụ thuộc hàm không phụ thuộc vào quan hệ cụ thể. Vì vậy có thể sử dụng các công cụ toán học để làm sáng tỏ cấu trúc logic của mô hình dữ liệu quan hệ.
4. Có nhiều quan hệ khác nhau nhưng họ đầy đủ các phụ thuộc hàm là như nhau.

Ký hiệu:

- $F := \{L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$ tập các phụ thuộc hàm.
- Ký hiệu $s = \langle \Omega, F \rangle$ là lược đồ quan hệ,
Trong đó $\Omega = \{a_1, a_2, \dots, a_n\}$ là tập các thuộc tính và
 $F = \{L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$ tập các phụ thuộc hàm.
- Quan hệ $R \in s := \langle \Omega, F \rangle$ được biểu diễn là một *thể hiện* (Instance) của lược đồ quan hệ $s := \langle \Omega, F \rangle$, là tập tất cả các bộ thỏa tất cả các phụ thuộc hàm F .

3.3.3 Các tính chất của phụ thuộc hàm

1. A1. *Tính phản xạ:* Nếu $B \subseteq A$ khi đó $A \rightarrow B$.
2. A2. *Tính gia tăng:* Nếu $A \rightarrow B$ và $C \subseteq \Omega$ khi đó $AC \rightarrow BC$.
3. A3. *Tính bắc cầu:* Nếu $A \rightarrow B$ và $B \rightarrow C$ khi đó $A \rightarrow C$.
4. A4. *Quy tắc hợp:* Nếu $A \rightarrow B$ và $A \rightarrow C$ khi đó $A \rightarrow BC$.
5. A5. *Quy tắc tách:* Nếu $A \rightarrow B$ và $C \subseteq B$ khi đó $A \rightarrow C$.

3.3.4 Các phép suy dẫn phụ thuộc hàm

Các phụ thuộc hàm có thể được suy dẫn bằng 2 cách:

1. *Suy dẫn theo định nghĩa:* Cho $R(\Omega)$ là một quan hệ trên Ω và tập các phụ thuộc hàm $F := \{f : L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$. Nói rằng phụ thuộc hàm $A \rightarrow B$, $A, B \subseteq \Omega$ được suy dẫn từ tập các phụ thuộc hàm F theo quan hệ R , nếu:
 - a) $A \rightarrow B$ thỏa trên quan hệ R tức là nếu 2 bộ bất kỳ trùng nhau trên A thì cũng trùng nhau trên B .
 - b) Nếu các phần tử của F thỏa trên mọi quan hệ R của lược đồ thì $A \rightarrow B$ cũng thỏa mãn trên các quan hệ đó.
2. *Suy dẫn logic:* Cho lược đồ $s = \langle \Omega, F \rangle$. Nói rằng $A \rightarrow B$ được suy dẫn logic từ F bằng cách áp dụng liên tiếp các tiên đề Armstrong. Tức là, nếu F thỏa trên mọi quan hệ trên lược đồ $s = \langle \Omega, F \rangle$ thì $A \rightarrow B$ cũng thỏa trên các quan hệ ấy. Đây là phép suy dẫn trực tiếp bằng cách áp dụng liên tiếp các hệ tiên đề và các tính chất của phụ thuộc hàm. Như vậy họ các phụ thuộc hàm không phụ thuộc vào một quan hệ cụ thể nào, có thể nhận được bằng cách suy dẫn từ định nghĩa hay suy dẫn theo quan hệ, hoặc suy dẫn logic từ các tiên đề hay theo các tính chất phụ thuộc hàm.

3.3.5 Bao đóng và tính chất bao đóng các phụ thuộc hàm

Định nghĩa bao đóng phụ thuộc hàm: Cho $\Sigma = \langle \Omega, F \rangle$ là một lược đồ quan hệ, trong đó Ω là tập các thuộc tính và F là tập các phụ thuộc hàm, khi đó ký hiệu:

$$F^+ := \{X \rightarrow Y \mid X, Y \subseteq \Omega \text{ và } X \rightarrow Y \text{ được suy dẫn logic từ } F\}$$

được gọi là tập bao đóng (*Closure*) của tập các phụ thuộc hàm. Như vậy bao đóng của tập các phụ thuộc hàm bao gồm các phụ thuộc được suy dẫn bằng cách áp dụng liên tiếp các hệ tiên đề Armstrong. Lực lượng của tập này rất lớn, khó có thể tính toán và liệt kê được. Trong phần sau sẽ làm quen với khái niệm tập tương đương.

Ví dụ 2:

- Cho $F = \{A \rightarrow B, B \rightarrow C, A \rightarrow D, B \rightarrow D\}$.
Khi đó $A \rightarrow C \in F^+$, $B \rightarrow DC \in F^+$ và $A \rightarrow BC \in F^+$.
 - Áp dụng quy tắc bắc cầu, từ $A \rightarrow B, B \rightarrow C$, suy ra $A \rightarrow C \in F^+$.
 - Vì $B \rightarrow C$ và $B \rightarrow D$, suy ra $B \rightarrow DC \in F^+$.
 - Vì $A \rightarrow B$ và $A \rightarrow C \in F^+$, suy ra $A \rightarrow BC \in F^+$.
- Cho $F = \{A \rightarrow B, C \rightarrow X, BX \rightarrow Z\}$. Khi đó $AC \rightarrow Z \in F^+$?
 - Vì $A \rightarrow B \Rightarrow AX \rightarrow BX$.
 - Từ $AX \rightarrow BX$, kết hợp $BX \rightarrow Z$, suy ra $AX \rightarrow Z$.
 - Từ $C \rightarrow X \Rightarrow AC \rightarrow AX$.
 - Áp dụng tính chất bắc cầu, $AC \rightarrow AX$ và $AX \rightarrow Z$ suy ra $AC \rightarrow Z \in F^+$.
- Cho $F = \{A \rightarrow B, C \rightarrow D\}$, $C \subset B$, chứng tỏ rằng $A \rightarrow D \in F^+$?
 - Vì $C \subset B$, áp dụng tính chất phản xạ, suy ra $B \rightarrow C$.
 - Từ $A \rightarrow B$ và $B \rightarrow C$ suy ra $A \rightarrow C$.
 - Từ $A \rightarrow C$ và $C \rightarrow D$ suy ra $A \rightarrow D \in F^+$.

Một số tính chất bao đóng phụ thuộc hàm:

- Tính phản xạ: $F \subseteq F^+$
- Tính đơn điệu: nếu $F \subseteq G \Rightarrow F^+ \subseteq G^+$
- Tính lũy đẳng: $F^+ = F^{++}$

3.3.6 Phụ thuộc hàm đầy đủ và không đầy đủ

Gọi F là tập các phụ thuộc hàm trên tập các thuộc tính Ω . Phụ thuộc $X \rightarrow Y \in F$ được gọi là một phụ thuộc hàm đầy đủ, khi và chỉ khi với mọi tập con thực sự của A : $\forall A' \subset A$ suy ra $A' \rightarrow B \notin F^+$, điều này có nghĩa nghĩa là $A' \rightarrow B$ là phụ thuộc không thể suy dẫn logic từ tập F . Nói cách khác, nếu $X \rightarrow Y \in F$ là một phụ thuộc hàm đầy đủ, khi và chỉ khi tất cả các tập con thực sự của tập các thuộc tính về trái không xác định được các thuộc tính chứa trong vế phải.

Ngược lại, phụ thuộc $X \rightarrow Y \in F$ được gọi là phụ thuộc hàm không đầy đủ khi và chỉ khi tồn tại một tập con thực sự của A: $A' \subset A$, khi đó, phụ thuộc $A' \rightarrow B$ được suy dẫn logic từ F, tức là $A' \rightarrow B \in F^+$.

Ví dụ 3 : Quan hệ quản lý mạng cáp.

Ω (TC#, TTC, MC#, DAI, NG, GT, SL, DV), trong đó

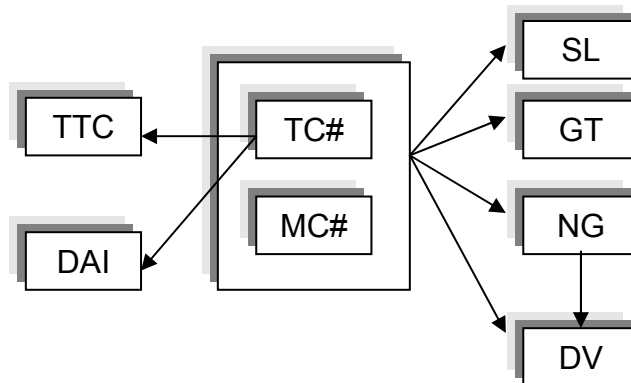
TC #	Mã tuyến cáp,	MC#	Mã cáp
TTC	Tên tuyến cáp,	DAI	Độ dài tuyến cáp
NG	Ngày lắp đặt cáp	DV	Đơn vị lắp đặt.
GT	Giá trị cáp	SL	Số lượng cáp

Khi đó:

- Các phụ thuộc hàm đầy đủ :

$(TC#, MC#) \rightarrow SL$,	$(TC#, MC#) \rightarrow GT$,
$TC# \rightarrow TTC$,	$(TC#, MC#) \rightarrow NG$,
$(C#, MC#) \rightarrow DV$,	$TC# \rightarrow DAI$,
$NG \rightarrow DV$,	$TC# \rightarrow TTC$.
- Các phụ thuộc hàm không đầy đủ :

$\{TC#, MC#\} \rightarrow TTC$,	$\{TC#, MC##\} \rightarrow DAI$.
----------------------------------	-----------------------------------



Hình 3.3 Các phụ thuộc hàm trong quan hệ quản lý mạng cáp

3.4 Bao đóng & thuật toán xác định bao đóng các thuộc tính

3.4.1 Bao đóng tập các thuộc tính (Closure of a set attributes)

Như trong các phần trước đã nghiên cứu, lực lượng của F^+ tập các phụ thuộc hàm được suy dẫn logic từ F bằng cách áp dụng các hệ tiên đề Armstrongs là quá lớn, trong khi có thể lực lượng của F rất nhỏ. Tập F chỉ là tập con của F^+ . Rõ ràng việc tính toán để tạo ra tập đóng F^+ tốn rất khá nhiều thời gian và chi phí quá cao. Ví dụ phụ thuộc hàm mà về phải gồm nhiều thuộc tính $\{X \rightarrow A_2, \dots, X \rightarrow A_n\}$, khi đó các phần tử của F^+ chứa tất cả các phụ thuộc $X \rightarrow Y$, trong đó $Y \subseteq \{A_1 A_2 \dots A_n\}$. Như vậy có thể có 2^n tập Y, số lượng các phụ thuộc của F^+ sẽ rất lớn, không hy vọng có thể liệt kê hết được các phần tử của tập F^+ .

Câu hỏi được đặt ra là, một phụ thuộc hàm bất kỳ cho trước $X \rightarrow Y$ có thuộc vào tập F^+ hay không. Nói cách khác, phụ thuộc hàm $X \rightarrow Y$ có thể được suy dẫn logic từ tập F^+ hay không?. Khái niệm bao đóng tập thuộc tính sẽ giúp trả lời câu hỏi trên.

Định nghĩa: Cho F là tập các phụ thuộc hàm trên tập các thuộc tính Ω và tập con bất kỳ $\forall X \subseteq \Omega$. Khi đó bao đóng của X ứng với tập F được định nghĩa như sau:

Tạo một chuỗi $X_0, X_1, X_2, \dots, X_n, X_{n+1}, X_{n+2}, \dots$ sao cho

$$X_0 := X$$

$$X_1 := X_0 \cup \{B \mid A \rightarrow B \in F^+, A \subseteq X_0, B \notin X_0\}.$$

$$\dots\dots\dots$$

$$X_{j+1} := X_j \cup \{B \mid A \rightarrow B \in F^+, A \subseteq X_j, B \notin X_j\}.$$

$$\dots\dots\dots$$

Hiển nhiên $X_0 \subseteq X_1 \subseteq X_2 \subseteq \dots \subseteq X_n \subseteq X_{n+1} \subseteq X_{n+2} \subseteq \dots$. Vì tập F là một tập hữu hạn, sẽ tồn tại một số k nguyên dương sao cho $X_{k-1} \subseteq X_k = X_{n+1} = X_{n+2} \dots$

Khi đó ký hiệu $X^+ := X_k = X_{n+1}$ và được gọi là bao đóng của thuộc tính X .

3.4.2 Các tính chất bao đóng thuộc tính

Cho $\forall X, Y \subseteq \Omega$

1. $X \subseteq X^+$
2. $X \subseteq Y \Leftrightarrow X^+ \subseteq Y^+$
3. $X^+ = X^{++}$
4. $(X Y)^+ \supseteq X^+ Y^+$
5. $(X Y)^+ = (X^+ Y)^+ = (X Y^+)^+$
6. $X \rightarrow Y \Leftrightarrow Y \subseteq X^+$
7. $X \rightarrow Y \Leftrightarrow Y^+ \subseteq X^+$
8. $X \rightarrow X^+$ và $X^+ \rightarrow X$
9. $X^+ = Y^+ \Leftrightarrow X \rightarrow Y$ và $Y \rightarrow X$

3.4.3 Thuật toán xác định bao đóng

Thuật toán 3.1: Xác định bao đóng tập thuộc tính

Giả sử tập các phụ thuộc hàm $F = \{L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$ thoả trên các quan hệ của lược đồ quan hệ $s = \langle \Omega, F \rangle$. Cho $X \subseteq \Omega$, tính bao đóng X^+ .

Thiết lập chuỗi $T_1, T_2, T_3, T_4, \dots, T_k, \dots$ sao cho

$$T_1 \subseteq T_2 \subseteq T_3 \subseteq T_4 \subseteq \dots \subseteq T_k \subseteq \dots$$

Bước 1: $G := F$.

Bước 2: Gán $T_1 = X$.

Bước 3: Kiểm tra phụ thuộc $A \rightarrow B \in G$.

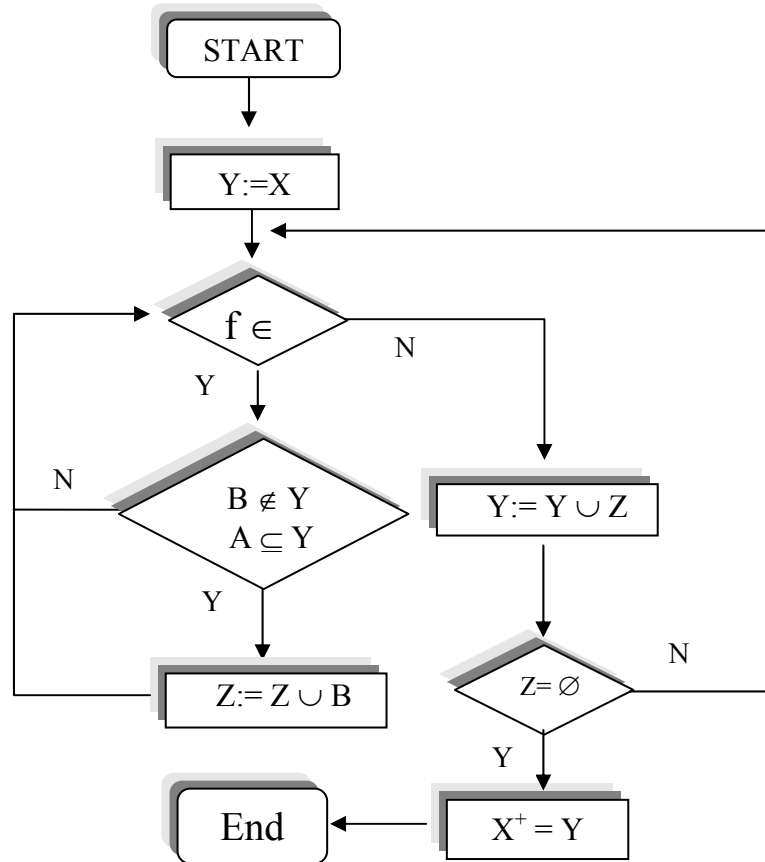
- Nếu $A \subseteq T_i$
 - ✓ Nếu $B \notin T_i$ khi đó $T_i = T_{i-1} \cup B, i = 2, 3, \dots,$
 - ✓ Ngược lại $B \subseteq T_i, T_i$ không thay đổi.

✓ $G := G - \{A \rightarrow B\}$.

✓ Lập lại bước 3.

- Thuật toán dừng kiểm tra nếu $G = \emptyset$ hoặc không tồn tại $A \rightarrow B \in G$, sao cho $A \subseteq T_i$. Nếu $B \subseteq T_i$, khi đó $A \rightarrow B$ là phụ thuộc dư thừa.

Bước 4: Tồn tại chỉ số k sao cho: $T_i = T_{i+1} = T_{i+2} = X^+$.



Hình 3.4 Sơ đồ thuật toán xác định bao đóng thuộc tính

Mô tả bằng ngôn ngữ tự nhiên:

Input : $s = \langle \Omega, F \rangle$ là một lược đồ quan hệ

Ω là tập các thuộc tính .

$F = \{A \rightarrow B \mid A, B \subseteq \Omega\}$ tập các phụ thuộc hàm.

$X \subseteq \Omega$ tập thuộc tính bất kỳ.

Output : Tính X^+ bao đóng thuộc tính X .

Mô tả phương pháp

Begin

$T := X$

$G := F$

Repeat

For each $A \rightarrow B$ in G do
 If ($A \subseteq T$ and $B \notin T$) then $T := T \cup B$
 $G := G - \{A \rightarrow B\}$.
 Until $G = \emptyset$ or không tồn tại $A \rightarrow B \in G$
 $X^+ = T$
 End.

Trong thuật toán xác định bao đóng của tập thuộc tính X ứng với tập phụ thuộc hàm F , tồn tại số k nguyên dương bé nhất sao:

$$X = T_1 \subseteq T_2 \subseteq \dots \subseteq T_{k-2} \subseteq T_{k-1} \subseteq T_k = T_{k+1} = T_{k+2}$$

3.4.4 Ví dụ bao đóng của tập thuộc tính

Ví dụ 4 $\Omega = \{A, B, C, D, E, G\}$

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

Tính X^+ với $X = \{BD\}$

- 1) $G = F, T_1 = BD$
- 2) Xét $AB \rightarrow C \in G: AB \not\subseteq T_1 = BD$.
- 3) Xét $D \rightarrow EG \in G: D \subseteq T_1 = BD, EG \not\subseteq T_1$.
 - $T_2 = T_1 \cup EG = BDEG$.
 - $G = G - \{D \rightarrow EG\} = \{AB \rightarrow C, C \rightarrow A, BE \rightarrow C, BC \rightarrow D,$
 $CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG\}$
- 4) Xét $C \rightarrow A \in G: D \subseteq T_2 = BDEG, A \not\subseteq T_2$.
 - $T_3 = T_2 \cup A = ABDEG$.
 - $G = G - \{C \rightarrow A\} = \{AB \rightarrow C, BE \rightarrow C, BC \rightarrow D,$
 $CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG\}$
- 5) Xét $BE \rightarrow C \in G: BE \subseteq T_3 = ABDEG, C \not\subseteq T_3$.
 - $T_4 = T_3 \cup C = ABCDEG$.
 - $G = G - \{BE \rightarrow C\} = \{AB \rightarrow C, BC \rightarrow D, CG \rightarrow BD$
 $ACD \rightarrow B, CE \rightarrow AG\}$
- 6) Xét $BC \rightarrow D \in G: BC \subseteq T_4 = ABCDEG, D \subseteq T_4$.
 - $T_5 = T_4 = ABCDEG$.
 - $G = G - \{BC \rightarrow D\} = \{AB \rightarrow C, CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG\}$
- 7) Xét $CG \rightarrow BD \in G: CG \subseteq T_5 = ABCDEG, BD \subseteq T_5$.
 - $T_6 = T_5 = ABCDEG$.
 - $G = G - \{CG \rightarrow BD\} = \{AB \rightarrow C, ACD \rightarrow B, CE \rightarrow AG\}$
- 8) Xét $ACD \rightarrow B \in G: ACD \subseteq T_6 = ABCDEG, B \subseteq T_6$.
 - $T_7 = T_6 = ABCDEG$.
 - $G = G - \{ACD \rightarrow B\} = \{AB \rightarrow C, CE \rightarrow AG\}$

- 9) Xét $CE \rightarrow AG \in G : CE \subseteq T_7 = ABCDEG, AG \subseteq T_7$.
- $T_8 = T_7 = ABCDEG$.
 - $G = G - \{CE \rightarrow AG\} = \{AB \rightarrow C\}$
- Như vậy $T_8 = X^+ = ABCDEG$.

Ví dụ 5 Cho tập $F = \{A \rightarrow B, B \rightarrow C, BC \rightarrow D, DA \rightarrow B\}$.

Tính bao đóng $X = \{A\}$ và chứng tỏ rằng $A \rightarrow AD \in F^+$:

- 1) $G = F, T_1 = A$
- 2) Xét $A \rightarrow B \in G : A \subseteq T_1 = A, B \not\subseteq T_1$.
 - $T_2 = T_1 \cup B = AB$.
 - $G = G - \{A \rightarrow B\} = \{B \rightarrow C, BC \rightarrow D, AD \rightarrow B\}$.
- 3) Xét $B \rightarrow C \in G : B \subseteq T_2 = AB, C \not\subseteq T_2$.
 - $T_3 = T_2 \cup C = ABC$.
 - $G = G - \{B \rightarrow C\} = \{BC \rightarrow D, AD \rightarrow B\}$
- 4) Xét $BC \rightarrow D \in G : BC \subseteq T_3 = ABC, D \not\subseteq T_3$.
 - $T_4 = T_3 \cup D = ABCD$.
 - $G = G - \{BC \rightarrow D\} = \{AD \rightarrow B\}$
- 5) Xét $AD \rightarrow B \in G : AD \subseteq T_4 = ABCD, B \subseteq T_4$.
 - $T_5 = T_4 = ABCD$.
 - $G = G - \{AD \rightarrow B\} = \emptyset$

Như vậy $T_5 = X^+ = ABCD$.

Với kết quả này chứng tỏ rằng các thuộc tính đơn hay tổ hợp chứa trong X^+ phụ thuộc hàm vào thuộc tính $X = A$.

Phụ thuộc $A \rightarrow AD \in F^+$: Vì

1. Theo giả thiết $A \rightarrow B$ và $B \rightarrow C$, suy ra $A \rightarrow C$.
2. Theo giả thiết $A \rightarrow B$ và $BC \rightarrow D$ suy ra $AC \rightarrow D$.
3. Theo (1) và (2): $A \rightarrow C$ và $AC \rightarrow D$ suy ra $A \rightarrow D$.
3. $A \rightarrow A$ và $A \rightarrow D$ (3) suy ra $A \rightarrow AD$.

3.4.5 Thuật toán xác định phụ thuộc hàm suy dẫn từ F

Sử dụng thuật toán xác định bao đóng của một tập các thuộc tính để xác định một phụ thuộc hàm bất kỳ có được suy dẫn logic từ tập các phụ thuộc hàm F hay không. Thuật toán sử dụng tính chất $X \rightarrow Y \in F^+ \Leftrightarrow Y \subseteq X^+$. Nghĩa là cần kiểm tra thuộc tính vế phải của phụ thuộc hàm có phải là một thành phần của bao đóng vế trái hay không.

Input : $F = \{A \rightarrow B \mid A, B \subseteq \Omega\}$ tập các phụ thuộc hàm.
 $X \rightarrow Y$

Output : Khẳng định $X \rightarrow Y \in F$

Begin

```

T := X
Repeat
  For each A → B in F do
    If (A ⊆ T and B ∉ T) then T := T ∪ B
    If Y ⊆ T then X → Y ∈ F, EXIT
  Until G = ∅ or không tồn tại A → B ∈ G
  X → Y ∉ F
End.

```

Ví dụ 6: Cho $F = \{A \rightarrow B, C \rightarrow X, BX \rightarrow Z\}$.

Kiểm tra $AC \rightarrow Z$ có thuộc F^+ , hay có được suy dẫn logic từ F^+ ?

I: Tính $(AC)^+$

- 1) $G = F, T_1 = AC$
 - 2) Xét $A \rightarrow B \in G: A \subseteq T_1, B \notin T_1$
 - $T_2 = T_1 \cup B = ABC$.
 - $G = G - \{A \rightarrow B\} = \{C \rightarrow X, BX \rightarrow Z\}$.
 - 3) Xét $C \rightarrow X \in G: C \subseteq T_2, X \notin T_2$
 - $T_3 = T_2 \cup X = ABCX$.
 - $G = G - \{C \rightarrow X\} = \{BX \rightarrow Z\}$.
 - 3) Xét $BX \rightarrow Z \in G: BX \subseteq T_3, Z \notin T_3$
 - $T_4 = T_3 \cup Z = ABCXZ$.
 - $G = \emptyset$
- Như vậy $T_4 = (AC)^+ = ABCXZ$.

II: $AC \rightarrow Z \in F^+ \Leftrightarrow Z \subseteq ABCXZ$.

Ví dụ 7: Cho $F = \{A \rightarrow B, C \rightarrow D\}, C \subseteq B$

Xác định $A \rightarrow D \in F^+$?

I: Tính $(A)^+$

- 1) $C \subseteq B \Leftrightarrow B \rightarrow C$
- 2) $G = F = \{A \rightarrow B, C \rightarrow D, B \rightarrow C\}, T_1 = A$
- 3) Xét $A \rightarrow B \in G: A \subseteq T_1, B \notin T_1$
 - $T_2 = T_1 \cup B = AB$.
 - $G = G - \{A \rightarrow B\} = \{C \rightarrow D, B \rightarrow C\}$.
- 4) Xét $C \rightarrow D \in G: C \not\subseteq T_2$.
- 5) Xét $B \rightarrow C \in G: B \subseteq T_2, C \notin T_2$
 - $T_3 = T_2 \cup C = ABC$.
 - $G = G - \{B \rightarrow C\} = \{C \rightarrow D\}$.
- 6) Xét $C \rightarrow D \in G: C \subseteq T_3, D \notin T_3$.

$$\square T_4 = T_3 \cup D = ABCD.$$

$$\square G = G - \{C \rightarrow D\} = \emptyset.$$

$$\text{Như vậy } T_4 = (A)^+ = ABCD.$$

$$\text{II: } A \rightarrow D \in F^+ \Leftrightarrow D \subseteq ABCD.$$

3.5 Phụ thuộc dư thừa, tập các phụ thuộc tương đương

3.5.1 Tập các phụ thuộc tương đương

Cho 2 tập các phụ thuộc hàm F và G cùng thoả trên một lược đồ $s = \langle \Omega, F \rangle$. Nói rằng F và G tương đương, ký hiệu $F \cong G$, khi và chỉ khi $F^+ \equiv G^+$, tức là các phụ thuộc của F được suy dẫn logic từ G và các phụ thuộc của G được suy dẫn logic từ F .

3.5.2 Phụ thuộc dư thừa

Định nghĩa: Cho $F = \{L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$ là tập các phụ thuộc hàm thoả trên lược đồ quan hệ $s = \langle \Omega, F \rangle$. Phụ thuộc $X \rightarrow Y \in F$ là *phụ thuộc dư thừa*, khi và chỉ khi $X \rightarrow Y$ được suy dẫn logic từ $G := F - \{X \rightarrow Y\}$, ngược lại phụ thuộc $X \rightarrow Y$ được gọi là phụ thuộc *không dư thừa*. Ký hiệu tập G là tập tất cả các phụ thuộc không dư thừa của tập F . Một tập các phụ thuộc hàm cho trước bỏ đi những phụ thuộc dư thừa, tập còn lại sẽ tương đương với tập đầu tiên.

Thuật toán 3.2: Xác định tập các phụ thuộc không dư thừa

Cho $F = \{L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$ thoả trên lược đồ quan hệ $s = \langle \Omega, F \rangle$. Thuật toán cho phép kiểm tra một phụ thuộc bất kỳ $A \rightarrow B$ thuộc F có là phụ thuộc dư thừa hay không.

Bước 1: $G := F - \{A \rightarrow B\}$.

\square Nếu $G \neq \emptyset$ tiếp tục thực hiện bước 2.

\square Ngược lại, nếu $G = \emptyset$ khi đó $A \rightarrow B$ là phụ thuộc không dư thừa.

Bước 2: Gán $T_1 = \{A\}$.

Bước 3: For $X \rightarrow Y \in G$, sao cho $X \subseteq T_1$.

$$T_i = T_{i-1} \cup \{Y\} \quad i = 2, 3, \dots,$$

\square Nếu $B \not\subseteq T_i$, khi đó $G := G - \{X \rightarrow Y\}$.

\square Nếu $G \neq \emptyset$, lập bước 3.

\square Nếu $G = \emptyset$, khi đó $A \rightarrow B$ là phụ thuộc không dư thừa.

\square Nếu $B \subseteq T_i$, khi đó $A \rightarrow B$ là phụ thuộc dư thừa.

Ví dụ 8: Cho tập phụ thuộc hàm $F = \{X \rightarrow YW, XW \rightarrow Z, Z \rightarrow Y, XY \rightarrow Z\}$.

$XY \rightarrow Z$ là phụ thuộc dư thừa của F ?

$$1) G := F - \{XY \rightarrow Z\} = \{X \rightarrow YW, XW \rightarrow Z, Z \rightarrow Y\}.$$

$$2) T_1 = \{XY\}$$

\square Khảo sát: $X \rightarrow YW$: $X \subseteq T_1 = \{XY\}$ và $Z \not\subseteq T_1$

$$\checkmark T_2 = T_1 \cup \{YW\} = \{XY\} \cup \{XYW\}.$$

$$\checkmark G := G - \{X \rightarrow YW\} = \{XW \rightarrow Z, Z \rightarrow Y\}$$

- Khảo sát: $XW \rightarrow Z$: $XW \subseteq T_2 = \{XYW\}$ và $Z \not\subseteq T_2$
 $\checkmark T_3 = T_2 \cup \{YW\} = \{XYW\} \cup \{Z\} = \{XYWZ\}$.
 $\checkmark G := G - \{XW \rightarrow Z\} = \{Z \rightarrow Y\}$

- Khảo sát: $Z \rightarrow Y$: $Z \subseteq T_3 = \{XYWZ\}$ và $Z \subseteq T_3$

Như vậy $XY \rightarrow Z$ là phụ thuộc dư thừa của F .

Có thể kiểm tra bằng suy dẫn logic như sau:

- $X \rightarrow YW \Rightarrow X \rightarrow Y$ và $X \rightarrow W$.
- $X \rightarrow W \Rightarrow X \rightarrow XW$ và $XW \rightarrow Z$ (gt) $\Rightarrow X \rightarrow Z$.
- $X \rightarrow Z \Rightarrow XY \rightarrow Z$.

3.5.3 Thuộc tính dư thừa

Cho tập các phụ thuộc hàm $F = \{L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$. Cho phụ thuộc hàm thuộc F có dạng $A_1A_2 \rightarrow B$. Tổng quát, vế trái của phụ thuộc hàm gồm nhiều thuộc tính. Nói rằng thuộc tính A_1 *dư thừa về trái* khi và chỉ khi $G^+ := F - \{A_1A_2 \rightarrow B\} \cup \{A_2 \rightarrow B\} \cong F^+$. Nói cách khác thuộc tính A_1 trong vế trái của phụ thuộc $A_1A_2 \rightarrow B$ là dư thừa, nếu thay $A_1A_2 \rightarrow B$ bằng $A_2 \rightarrow B$ thì bao đóng F^+ không thay đổi.

Thuật toán 3.3: Loại bỏ các thuộc tính dư thừa về trái

Bước 1: $G := F$

Bước 2: Kiểm tra tất cả phụ thuộc có dạng $A_1A_2A_3\dots A_n \rightarrow B \in G$

Bước 3: Loại bỏ tạm thời $A_i, i=1 \div n$ trong $A_1A_2\dots A_n \rightarrow B$

- Kiểm tra $A_1A_2\dots A_{i-1}A_{i+1}\dots A_n \rightarrow B$ là phần tử của G^+ hay không, bằng cách áp dụng thuật toán xác định bao đóng của tập thuộc tính và áp dụng tính chất $X \rightarrow Y \Leftrightarrow Y \subseteq X^+$.
 - Nếu thỏa, A_i là thuộc tính dư thừa.
 - Ngược lại, tiếp tục thuộc tính tiếp theo A_{i+1} .
- Tiếp tục khảo sát các phụ thuộc có vế trái gồm nhiều thuộc tính.

Ví dụ 9: Cho $F = \{X \rightarrow Z, XY \rightarrow WP, XY \rightarrow ZWQ, XZ \rightarrow R\}$.

Loại bỏ các thuộc tính về trái dư thừa

1) $G := F = \{X \rightarrow Z, XY \rightarrow WP, XY \rightarrow ZWQ, XZ \rightarrow R\}$

2) Khảo sát $XY \rightarrow WP$

- Loại bỏ X : $(Y)^+ = Y$ và $WP \not\subseteq Y$. Suy ra X không dư thừa.
- Loại bỏ Y : $(X)^+ = XZR$ và $WP \not\subseteq XZR$. Suy ra Y không dư thừa.
- Như vậy vế trái của $XY \rightarrow WP$ không chứa thuộc tính dư thừa.

3) Khảo sát $XZ \rightarrow R$

- Loại bỏ X : $(Z)^+ = Z$ và $R \not\subseteq Z$. Suy ra X không dư thừa.
- Loại bỏ Z : $(X)^+ = XZR$ và $R \subseteq XZR$. Suy ra Z dư thừa.
- Như vậy vế trái của $X \rightarrow R \in G$

4) $G := \{X \rightarrow Z, XY \rightarrow WP, XY \rightarrow ZWQ, X \rightarrow R\} \cong F$

3.5.4 Tập các phụ thuộc phủ tối thiểu

Thông thường, tập các phụ thuộc hàm F chứa các phụ thuộc dư thừa và các phụ thuộc có các thuộc tính về trái dư thừa, ảnh hưởng rất nhiều đến hiệu quả khi tách không tổn thất thông tin và xác định khóa của lược đồ quan hệ. Khái niệm phủ tối thiểu các phụ thuộc hàm sẽ hỗ trợ cho các quá trình thực hiện các thuật toán nhanh và chính xác hơn, hiệu quả hơn.

Định nghĩa: Cho F là một tập phụ thuộc hàm của $s = \langle \Omega, F \rangle$. Nói rằng G là tập phụ thuộc phủ tối thiểu (Minimal cover) của F nếu G thỏa các điều kiện sau:

1. Về phải của tất cả phụ thuộc hàm của G chứa duy nhất một thuộc tính (thuộc tính đơn).
Ví dụ $X \rightarrow ABC$ khi đó $X \rightarrow A, X \rightarrow B$ và $X \rightarrow C$.
2. Không tồn tại phụ thuộc sao cho có chứa thuộc tính về trái dư thừa. Nghĩa là không tồn tại $X \rightarrow A \in G$ sao cho với mọi tập con thực sự $Z \subset X$ của X thì phụ thuộc $Z \rightarrow A$ không được suy dẫn logic từ $G - \{X \rightarrow A\} \cup \{Z \rightarrow A\}$.
3. Không tồn tại phụ thuộc dư thừa $X \rightarrow A \in G$ sao cho $G - \{X \rightarrow A\}$ tương đương với G. Vì theo điều kiện (1), trong tất cả các phụ thuộc hàm, về phải không được chứa các thuộc tính dư thừa bằng cách áp dụng các quy tắc tách và hợp các phụ thuộc. Điều kiện (2) đảm bảo không có phụ thuộc hàm nào mà về trái của nó chứa thuộc tính dư thừa, nghĩa là không tồn tại các phụ thuộc hàm không đầy đủ trong tập G. Thuộc tính B dư thừa trong X của phụ thuộc hàm $X \rightarrow A$ khi và chỉ khi $A \in (X - B)^+, B \subseteq X$ và điều kiện (3) đảm bảo cho tập các phụ thuộc G không chứa các phụ thuộc dư thừa, bằng cách xác định bao đóng X^+ ứng với tập các phụ thuộc còn lại trong $G - \{X \rightarrow A\}$.

Có thể khẳng định tập các phụ thuộc hàm F đương đương với tập phụ thuộc hàm G trong đó các về phải không chứa quá một thuộc tính và mỗi một tập các phụ thuộc hàm F có chứa một phủ tối thiểu.

Ví dụ 10 : Cho $F = \{AB \rightarrow C, D \rightarrow EG, C \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG\}$

1. Tách các về phải chỉ chứa một thuộc tính:

$AB \rightarrow C$	$D \rightarrow E$	$CG \rightarrow B$
$C \rightarrow A$	$D \rightarrow G$	$CG \rightarrow D$
$BC \rightarrow D$	$BE \rightarrow C$	$CE \rightarrow A$
$ACD \rightarrow B$	$CE \rightarrow G$	

2. Xóa các phụ thuộc dư thừa:

$CE \rightarrow A$ vì được suy dẫn từ $C \rightarrow A$.
 $CG \rightarrow B$ vì được suy dẫn từ $CG \rightarrow D, C \rightarrow A, ACD \rightarrow B$.

3. Thuộc tính dư thừa về trái:

Thay $ACD \rightarrow B$ bằng $CD \rightarrow B$, vì $C \rightarrow A, ACD \rightarrow B$ có thể suy ra $CD \rightarrow B$.

Kết quả:

$$\begin{array}{lll}
 AB \rightarrow C & D \rightarrow E & CE \rightarrow G \\
 C \rightarrow A & D \rightarrow G & CD \rightarrow B \\
 BC \rightarrow D & BE \rightarrow C & CG \rightarrow D
 \end{array}$$

3.5.5 Thuật toán 3.4 xác định tập phủ tối thiểu

Input : $s = \langle \Omega, F \rangle$ là một lược đồ quan hệ

Ω là tập các thuộc tính .

$$F = \{ A \rightarrow B \mid A, B \subseteq \Omega \}.$$

Output : Xác định tập phủ tối thiểu G từ F

Mô tả phương pháp

1. Tách vế phải của tất cả các phụ thuộc hàm sao cho chỉ chứa duy nhất một thuộc tính. Áp dụng bổ đề 4.2.
2. Loại bỏ những phụ thuộc dư thừa dạng $X \rightarrow A \in F$ ra khỏi tập phụ thuộc F nếu phụ thuộc này được suy dẫn logic từ $F - \{ X \rightarrow A \}$.
3. Xoá các thuộc tính dư thừa trong vế trái của phụ thuộc $X \rightarrow A \in F$ sao cho nếu $Z \subset X$ thì $G = F - \{ X \rightarrow A \} \cup \{ Z \rightarrow A \}$ tương đương với F .

Ví dụ 11: Cho $F = \{ X \rightarrow Z, XY \rightarrow WP, XY \rightarrow ZWQ, XZ \rightarrow R \}$.

Xác định tập phủ tối thiểu từ tập F .

1. Tách các vế phải của các phụ thuộc sao cho chỉ chứa duy nhất một thuộc tính, áp dụng tính chất hợp/tách:
 $G = \{ X \rightarrow Z, XY \rightarrow W, XY \rightarrow P, XY \rightarrow Z, XY \rightarrow W, XY \rightarrow Q, XZ \rightarrow R \}.$
2. Loại bỏ các phụ thuộc dư thừa:
 $G = \{ X \rightarrow Z, XY \rightarrow W, XY \rightarrow P, XY \rightarrow Z, XY \rightarrow Q, X \rightarrow R \}$
3. Loại bỏ các thuộc tính vế trái dư thừa:
 $G = \{ X \rightarrow Z, XY \rightarrow W, XY \rightarrow P, XY \rightarrow Q, X \rightarrow R \}.$

3.6 Khóa của lược đồ quan hệ

Trong một lược đồ quan hệ luôn luôn tồn tại một thuộc tính hay một tập các thuộc tính có khả năng biểu diễn duy nhất các thuộc tính còn lại. Nói cách khác giá trị của một hay nhiều thuộc tính có thể xác định duy nhất giá trị các thuộc tính khác. Số chứng minh thư sẽ xác định được tất cả thông tin về người mang chứng minh thư đó. Ngày gọi, thời gian bắt đầu, thời gian kết thúc, số máy gọi đi, số máy gọi đến xác định duy nhất thông tin về cuộc đàm thoại, số thuê bao điện thoại...Tập thuộc tính có tính chất như trên được gọi là khoá của lược đồ quan hệ. Thông thường trong một lược đồ quan hệ có thể tồn tại nhiều khoá. Trong số đó sẽ chọn một khoá làm khoá chính sao cho đơn giản và không nhập nhằng thông tin.

3.6.1 Định nghĩa

Cho $s = \langle \Omega, F \rangle$ là một lược đồ quan hệ, Ω là tập các thuộc tính khác rỗng và tập các phụ thuộc hàm (tập phủ tối thiểu) $F := \{ A \rightarrow B \mid A, B \subseteq \Omega \}$. Cho tập con bất kỳ $\forall X \subseteq \Omega$. Nói rằng X là *khóa* (Key) của lược đồ quan hệ s khi và chỉ khi:

a) $(X \rightarrow \Omega) \in F^+$

b) Không tồn tại $Z \subset X$ sao cho $(Z \rightarrow \Omega) \in F^+$

Nếu X thỏa điều kiện (a) và không thỏa điều kiện (b) được gọi là *siêu khoá* của lược đồ quan hệ (Supperkey) $s = \langle \Omega, F \rangle$. Điều kiện (a) và (b) khẳng định các thuộc tính không khoá phụ thuộc đầy đủ vào khoá. Từ định nghĩa trên có thể suy ra rằng X là khoá của lược đồ quan hệ khi và chỉ khi $X^+ = \Omega$ và $(X - A)^+ \neq \Omega, \forall A \in X$. Điều này có nghĩa là mỗi một giá trị của khoá xác định duy nhất giá trị của các thuộc tính không khoá. Giá trị khoá khác nhau thì giá trị các bộ có chứa giá trị khoá cũng khác nhau. Nếu loại bỏ một phần thông tin về khoá thì thông tin của các thuộc tính còn lại không thể xác định được. Như vậy khoá là tập các thuộc tính sao cho bao đóng của nó là nhỏ nhất. Nghĩa là nếu thêm hoặc loại bỏ các phần tử của khoá sẽ là dư thừa hay thiếu thông tin.

Có thể định nghĩa khoá trong một quan hệ như sau: nếu 2 bộ bất kỳ trùng nhau trên các thành phần của khoá thì cũng trùng nhau trên các thành phần không khoá và nếu trùng nhau trên một số thành phần của khoá không thể trùng nhau trên các thành phần không khoá.

Giá trị các thành phần của khoá không thể nhận giá trị null hay các giá trị không xác định. Mọi siêu khoá đều chứa ít nhất một khoá. Từ siêu khoá có thể xác định một khoá bằng cách bớt dần các thuộc tính và kiểm tra bao đóng của các thuộc tính còn lại. Các thuộc tính là các phần tử của khoá gọi là các thuộc tính khoá, ngược lại, các thuộc tính không chứa trong khoá gọi là các thuộc tính không khoá.

Ký hiệu K là tập khoá của lược đồ quan hệ $s = \langle \Omega, F \rangle$.

Ví dụ 12: Cho $\Omega = \{A, B, C, D, E, G\}$ và

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

Khi đó tập khoá của lược đồ quan hệ là : $K = \{AB, CG, CD, EB, CE, BC\}$

$$\begin{array}{ll} K_1 = AB & \text{vì } (AB)^+ = ABCDEG \\ K_3 = CG & \text{vì } (CG)^+ = ABCDEG \\ K_5 = CD & \text{vì } (CD)^+ = ABCDEG \end{array} \quad \begin{array}{ll} K_2 = EB & \text{vì } (EB)^+ = ABCDEG \\ K_4 = CE & \text{vì } (CE)^+ = ABCDEG \\ K_6 = BC & \text{vì } (BC)^+ = ABCDEG \end{array}$$

Ví dụ.13: Cho lược đồ quan hệ quản lý mạng cáp, bao gồm tập các thuộc tính (TC#, MC#, MC, GIA, SL, GTR). Trong đó:

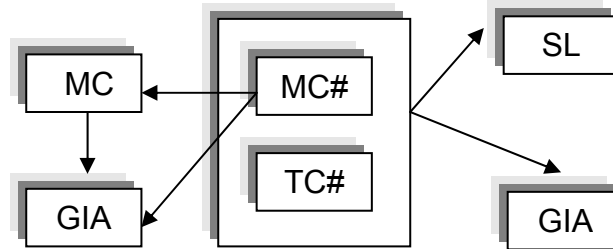
TC#	Mã tuyến cáp	MC#	Mã cáp
MC	Tên gọi cáp	GIA	Đơn giá cáp
SL	Số lượng	GTR	Giá trị

Biết rằng:

- Trong mỗi tuyến cáp TC# có thể lắp đặt nhiều loại cáp khác nhau
- Ứng với một mã cáp MC# chỉ có một và chỉ một tên gọi loại cáp MC.
- Mỗi loại cáp MC# , trong một tuyến cáp TC# xác định một giá trị cáp GTR

Tổ hợp các thuộc tính (MC#, TC#) là khoá của lược đồ quan hệ. Trên một tuyến cáp cụ thể, với một loại cáp cụ thể đã được lắp đặt, sẽ xác định duy nhất các thông tin về các loại cáp

được lắp đặt. Thuộc tính MC# chỉ xác định được các thông tin về loại cáp như tên gọi MC, đơn giá GIA và nước sản xuất loại cáp đó. Nó không thể xác định số lượng SL và giá trị GTR cáp của loại cáp cụ thể, nếu giá trị của TC# không được xác định. Trong trường hợp này có thể hiểu cáp vẫn còn trong các kho, chưa được đưa ra sử dụng. Tương tự, thuộc tính TC# chỉ xác định thông tin về các tuyến cáp như chiều dài tuyến, tên gọi tuyến, cáp chôn hay cáp treo...mà không thể xác định các thông tin về số lượng và giá trị các loại cáp được lắp đặt trên nó. Sơ đồ sau biểu diễn mối quan hệ giữa các thuộc tính khoá và không khoá.



Hình 3.5 Các phụ thuộc hàm trong quan hệ quản lý cáp

3.6.2 Khoá dự bị

Trong một lược đồ quan hệ, nếu tồn tại một thuộc tính hay một tổ hợp các thuộc tính có tính chất xác định như khoá chính, khi đó nói rằng trong lược đồ tồn tại khoá dự bị. Khoá chính có thể tùy chọn trong số các khoá dự bị của lược đồ quan hệ.

Trong quan hệ quản lý mạng cáp, giả sử ứng với một mã cáp MC# chỉ có một và chỉ một tên gọi MC. Nghĩa là ngoài tổ hợp khoá (MC#, TC#), tổ hợp (MC,TC#) cũng có tính chất tương tự như khoá. Và thường người ta chọn tổ hợp (MC#,TC#) làm khoá chính, vì (MC#, TC#) đảm bảo tính nhất quán và đỡ nhập những thông tin hơn so với tổ hợp các thuộc tính tên cáp và mã tuyến cáp (MC,TC#).

Các thuộc tính không khoá là các thuộc tính không có mặt trong các thành phần của khoá.

Tức là: $\Psi := \{A \mid A \notin K \ \& \ K \in K \text{ tập khoá}\}$

3.6.3 Định thuộc (Determinant)

Cho lược đồ quan hệ $s = \langle \Omega, F \rangle$. Gọi $\forall X \subseteq \Omega$ là một định thuộc (Determinant) của lược đồ s khi và chỉ khi tồn tại thuộc tính $Y \subseteq \Omega$ sao cho $X \rightarrow Y \in F^+$ là phụ thuộc hàm đầy đủ. Tức là trong X không tồn tại các thuộc tính dư thừa. (MC#,TC#) là một định thuộc của lược đồ quan hệ quản lý mạng cáp, vì tồn tại $GTR \subseteq \Omega = (NG, MH#, TMH, SL, GIA, NB)$, $(MH#, NG) \rightarrow GTR$ là một phụ thuộc đầy đủ.

3.6.4 Phản khoá (Antikey)

Phản khoá có vai trò quan trọng trong quá trình nghiên cứu cấu trúc logic của họ các phụ thuộc hàm, khoá và các dạng chuẩn của lược đồ quan hệ. Đặc biệt đối với các bài toán tổ hợp trong mô hình dữ liệu quan hệ.

a) Hệ Sperner: Cho Ω là tập các thuộc tính, $P \subseteq P(\Omega) := \{A \mid A \subseteq \Omega\} \subseteq 2^\Omega$ được gọi là hệ Sperner trên Ω nếu với $\forall A, B \in P, A \not\subseteq B (B \not\subseteq A)$. Dễ dàng nhận thấy rằng tập tất cả khoá của lược đồ quan hệ cũng là một hệ Sperner trên tập các thuộc tính Ω .

b) *Phản khóa*: Giả sử $K = \{K_1, K_2, \dots, K_n\}$ là tập các khóa của lược đồ quan hệ. .Hiển nhiên tập K là một hệ Sperner trên tập Ω . Một tập con của Ω được gọi là *phản khoá* của lược đồ quan hệ s , ký hiệu là $K^{-1} \subseteq \Omega$ nếu:

1. $K \not\subseteq K^{-1}$ với $\forall K \in K$
2. $\forall X \subseteq \Omega$ và $K^{-1} \subset X$ suy ra $\exists K \in K$ sao cho $K \subseteq X$.

Từ định nghĩa trên có thể hiểu là phản khóa không chứa khóa và nếu nói rộng phản khóa sẽ chứa khóa. Ký hiệu K^{-1} tập tất cả phản khoá của lược đồ quan hệ .

3.6.5 Lược đồ thu gọn

Định nghĩa: Cho $s = \langle \Omega, F \rangle$ là lược đồ quan hệ, Ω là tập các thuộc tính và F tập các phụ thuộc hàm và K là tập khoá của lược đồ quan hệ. Lược đồ $s_1 := \langle \Omega_1, F_1 \rangle$ được gọi là lược đồ thu gọn: $\Omega_1 = \bigcup_{K \in K} K$, $G := \Omega \setminus \Omega_1$

$$F_1 = \{(L_j \setminus G) \rightarrow (R_j \setminus G) \mid (L_j \rightarrow R_j) \in F\}$$

Một số khẳng định sau

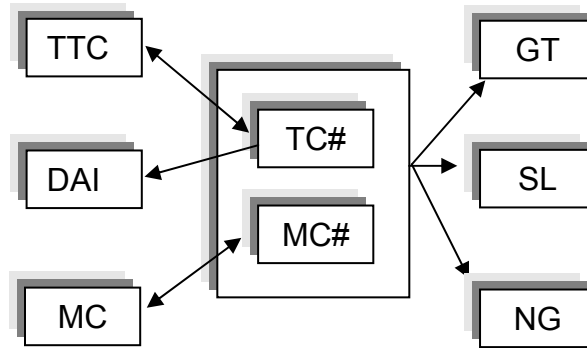
- Nếu K^{-1}_s là tập phản khóa của $s = \langle \Omega, F \rangle$
 Khi đó $K^{-1}_{s_1} = \{K^{-1} \setminus G \mid K^{-1} \in K^{-1}_s\}$ là phản khóa của $s_1 := \langle \Omega_1, F_1 \rangle$.
- Nếu $K^{-1}_{s_1}$ là tập phản khóa của $s_1 := \langle \Omega_1, F_1 \rangle$.
 Khi đó $K^{-1}_s = \{K^{-1} \cup G \mid K^{-1} \in K^{-1}_{s_1}\}$ là tập phản khóa của $\langle \Omega, F \rangle$.
- $$\bigcup_{K \in K} K = \Omega \setminus \bigcap_{K^{-1} \in K^{-1}} K^{-1}$$
- Nhận biết phản khoá)
 Nếu $\forall X \subseteq \Omega$ là phản khoá khi và chỉ khi:
 1. $X^+ = X$
 2. Nếu $\forall Y \subseteq \Omega$ sao cho $X \subset Y$ và $Y^+ = Y$ thì $Y = \Omega$ và $X^+ \neq \Omega$

Ví dụ.14: Trong quan hệ R (TC#, TTC, MC#, MC, DAI, NG, GTR, SL)

TC#	Mã tuyến cáp	TTC	Tên gọi tuyến cáp
MC#	Mã cáp	MC	Tên gọi loại cáp
DAI	Độ dài tuyến cáp	NG	Ngày lắp đặt cáp trên tuyến
GTR	Giá trị một loại cáp trên tuyến cáp	SL	Số lượng cáp trên tuyến

- Mỗi một giá trị của TC# xác định duy nhất một giá trị của tên tuyến cáp TTC và mỗi một giá trị của TTC xác định duy nhất một giá trị của TC.
- Một giá trị của mã cáp MC# xác định duy nhất một giá trị của tên cáp MC, ngược lại, mỗi một giá trị của MC xác định duy nhất một giá trị của MC#.

Khoá dự bị: (TC#, TTC#), (TC, TTC), (TC#, TTC), (TC, TTC#).
 Khoá chính: (TC#, MC#).
 Định thuộc: (TC#, TTC#), (TC#, TTC), (TC, TTC#), (TC, TTC),
 TTC, MC, TC#, MC#.
 Phản khoá: (TC#, TTC, DAI, NG, GT, SL), (MC#, MC, DAI, NG, GT, SL)



Hình 3.6 Lược đồ quan hệ quản lý mạng cáp

3.7 Một số tính chất của khoá trong lược đồ quan hệ

Cho $s = \langle \Omega, F \rangle$ là một lược đồ quan hệ, trong đó Ω là tập các thuộc tính và $F = \{ L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega \}$ là tập các phụ thuộc hàm.

$$\begin{aligned} \text{Ký hiệu} \quad L &= \cup L_j \quad \text{và} \quad R = \cup R_j \\ L_j \rightarrow R_j &\in F \quad \quad L_j \rightarrow R_j \in F \\ H &= \cup_{K \in K} K \quad \text{và} \quad G = \cap_{K \in K} K \end{aligned}$$

- Với $\forall K \in K$, khi đó: $\Omega \setminus R \subseteq K \subseteq (\Omega \setminus R) \cup (L \cap R)$.
- Các khoá của lược đồ quan hệ $s = \langle \Omega, F \rangle$ chỉ khác nhau trên các thuộc tính của $(L \cap R)$.
- Nếu $(L \cap R) = \emptyset$ khi đó $(\Omega \setminus R)$ là khoá duy nhất của quan hệ $s = \langle \Omega, F \rangle$.
- Nếu $(R \setminus L) \neq \emptyset$ thì tồn tại khoá K sao cho $K \neq \Omega$ là khoá không tầm thường
- Nếu $\forall i: (R_i \cap L \neq \emptyset \Rightarrow L_i \cap R = \emptyset)$
 Khi đó $\Omega \setminus R$ là khoá duy nhất $s = \langle \Omega, F \rangle$
- Đặt $L \cap R = \{A_1, A_2, \dots, A_k\} \subseteq \Omega$ và $K_1 := (\Omega \setminus R) \cup (L \cap R)$.

$$K_{(i+1)} := \begin{cases} K_i \setminus A_i & \text{Nếu } (K_i \setminus A_i) \rightarrow A_i \in F^+ \\ K_i & \text{Nếu } (K_i \setminus A_i) \rightarrow A_i \notin F^+ \end{cases}$$

Với $i = 1 \div k$

Khi đó $K_{(k+1)}$ là khoá của lược đồ quan hệ $s = \langle \Omega, F \rangle$.

Ví dụ 15 $\Omega = \{ A, B, H, G, Q, M, N, V, W \}$
 $F := \{ A \rightarrow B, B \rightarrow H, G \rightarrow Q, V \rightarrow W, W \rightarrow V \}$

$$\begin{aligned}
 \text{Khi đó } \quad L &= \{ABGVW\}, \\
 R &= \{BHQWV\}, \\
 L \cap R &= \{BVW\}, \\
 (\Omega \setminus R) &= A G M N \\
 (\Omega \setminus R \subseteq K \subseteq (\Omega \setminus R) \cup (L \cap R) \\
 \{A G M N\} &\subseteq K \subseteq \{A G M N\} \cup \{BVW\}
 \end{aligned}$$

Như vậy khóa lược đồ quan hệ chỉ khác nhau trên các thuộc tính B, V và W

3.8 Các thuật toán xác định khoá của lược đồ quan hệ.

Thuật toán 3.6: Xác định một khoá lược đồ quan hệ

$$\begin{aligned}
 \text{Input: } \quad \Omega &:= \{A_1, A_2, \dots, A_n\} \neq \emptyset \\
 F &:= \{f : L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\} \\
 L &= \bigcup_{L_j \subseteq \Omega} L_j \quad \text{và} \quad R = \bigcup_{R_j \subseteq \Omega} R_j
 \end{aligned}$$

$$L \cap R = \{A_{j1}, A_{j2} \dots A_{jk}\}$$

Output: : Xác định một khoá K của lược đồ quan hệ

Phương pháp: Khoá của lược đồ quan hệ $s = \langle \Omega, F \rangle$ chỉ khác nhau trên các thuộc tính của $L \cap R$, vì vậy chỉ cần kiểm tra và loại dần những phần tử $A \in L \cap R$ sao cho $(X \setminus a)^+ = \Omega$.

- Nếu $(\Omega \setminus R)^+ = \Omega$ suy ra $(\Omega \setminus R)$ là khoá duy nhất của lược đồ quan hệ $s = \langle \Omega, F \rangle$.
- Nếu $(\Omega \setminus R)^+ \neq \Omega$ lược đồ quan hệ có khoá không tầm thường.
- $X = (\Omega \setminus R) \cup (L \cap R)$.
- Kiểm tra $(X \setminus A_{ji})^+$

Begin

$$X = (\Omega \setminus R)$$

If $(\Omega \setminus R)^+ \subset \Omega$ then

Begin

$$X = X \cup (L \cap R)$$

 For each $A \in (L \cap R)$ do

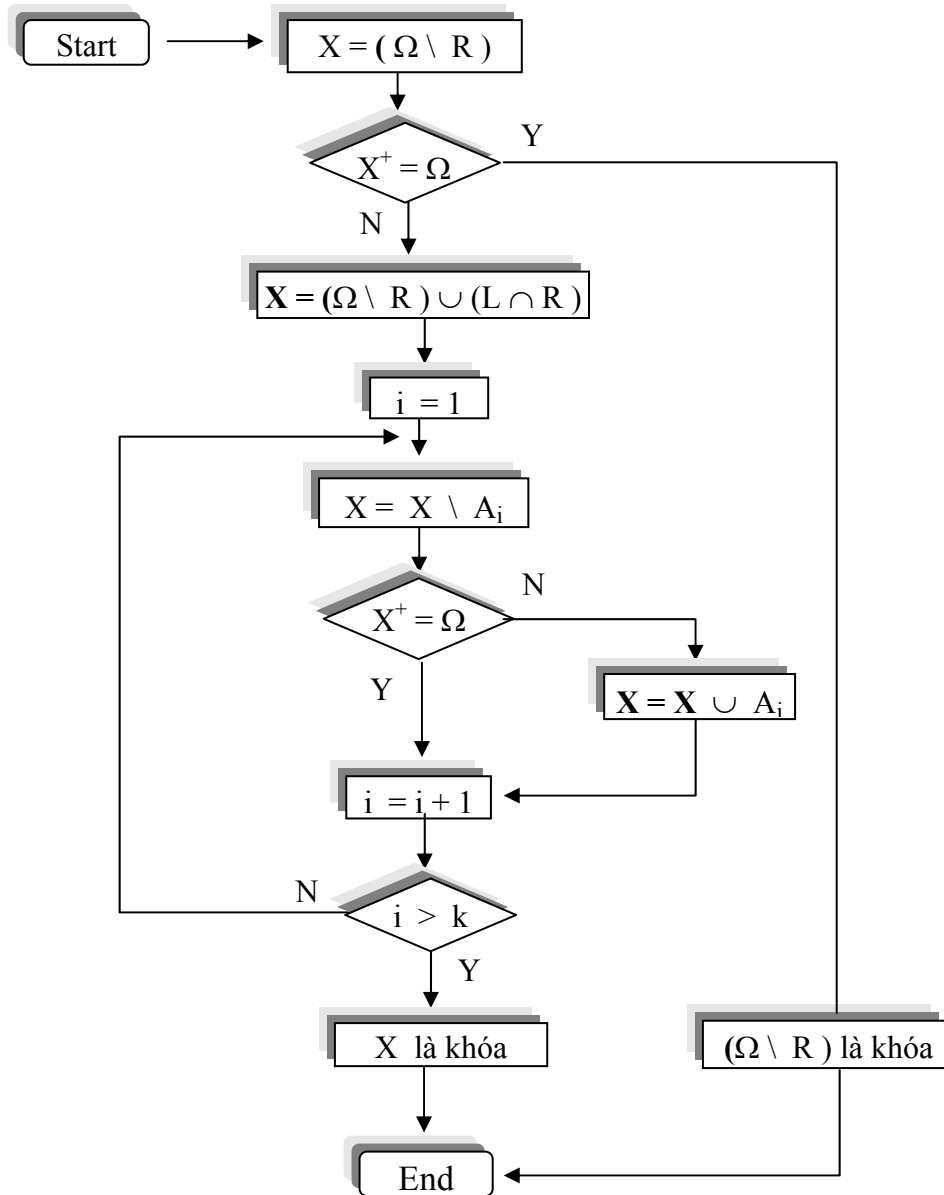
 If $(X - A)^+ = \Omega$ then

$$X = X - A$$

 End

X là khoá của lược đồ quan hệ $s = \langle \Omega, F \rangle$.

End.



Hình 3.7 Sơ đồ thuật toán xác định khóa lược đồ quan hệ

Ví dụ 16: $\Omega = \{A, B, C, D, E, G\}$, $F = \{B \rightarrow C, C \rightarrow B, A \rightarrow GD\}$
 $L = ABC$, $R = BCDG$, $L \cap R = BC$.
 $(\Omega \setminus R) = AE$ không là khóa của $\langle \Omega, F \rangle$
 vì $(\Omega \setminus R)^+ = (AE)^+ = AEDG \neq \Omega$.
 Đặt $X = (\Omega \setminus R) \cup (L \cap R) = ABCE$.

- a) Theo thứ tự $L \cap R = BC$
 Xét $(X - B)^+ = (ACE)^+ = ABCDEG = \Omega \Rightarrow X = ACE$
 $(X - C)^+ = (AE)^+ = ADEG \neq \Omega$.

Suy ra $X = ACE$ là khóa.

b) Theo thứ tự $L \cap R = CB$

Xét $(X - C)^+ = (ABE)^+ = ABCDEG = \Omega \Rightarrow X = ABE$

$(X - B)^+ = (AE)^+ = ADEG \neq \Omega$.

Vậy $X = ABE$ cũng là là khóa thứ 2 của lược đồ quan hệ.

Như vậy khóa của lược đồ quan hệ trên là $K_1 = ABE$ và $K_2 = ABE$.

Thuật toán 3.7: Xác định một khoá trong siêu khóa cho trước

Giả sử K là khóa được chứa trong siêu khóa X cho trước. Hiển nhiên $K \subseteq X$. Mặt khác

$\Omega \setminus R \subseteq K \subseteq (\Omega \setminus R) \cup (L \cap R)$ Suy ra $K \subseteq (\Omega \setminus R) \cup (X \cap (L \cap R))$.

Như vậy nếu cho trước một siêu khóa X , có thể xác định một khóa được chứa trong nó.

Với thuật toán này thời gian kiểm tra chỉ phụ thuộc vào số phần tử của $X \cap (L \cap R)$, nghĩa là thời gian để xác định một khóa trong $Y := (\Omega \setminus R) \cup X \cap (L \cap R)$ nhanh hơn rất nhiều để xác định một khóa trong X .

Input: $\Omega := \{ A_1, A_2, \dots, A_n \} \neq \emptyset$

$F := \{ L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega \}$

$L = \bigcup_{L_j \subseteq \Omega} L_j$ và $R = \bigcup_{R_j \subseteq \Omega} R_j$

$X \cap (L \cap R) = \{ A_1, A_2, \dots, A_k \}$

Output: Xác định một khoá K trong siêu khóa X cho trước của lược đồ quan hệ.

Phương pháp:

- Nếu $X = (\Omega \setminus R)$ suy ra $(\Omega \setminus R)$ là siêu khóa, đồng thời cũng là khoá duy nhất của lược đồ quan hệ $s = \langle \Omega, F \rangle$.
- Đặt $Y := (\Omega \setminus R) \cup X \cap (L \cap R) = \{ A_1, A_2, \dots, A_k \}$.
- $X = X \cap (L \cap R) = \{ A_1, A_2, \dots, A_k \}$.
- Kiểm tra $(X \setminus A_j)^+$

Begin

If $X = (\Omega \setminus R)$ then

Begin

$X = (\Omega \setminus R) \cup X \cap (L \cap R)$

For each $A \in X \cap (L \cap R)$ do

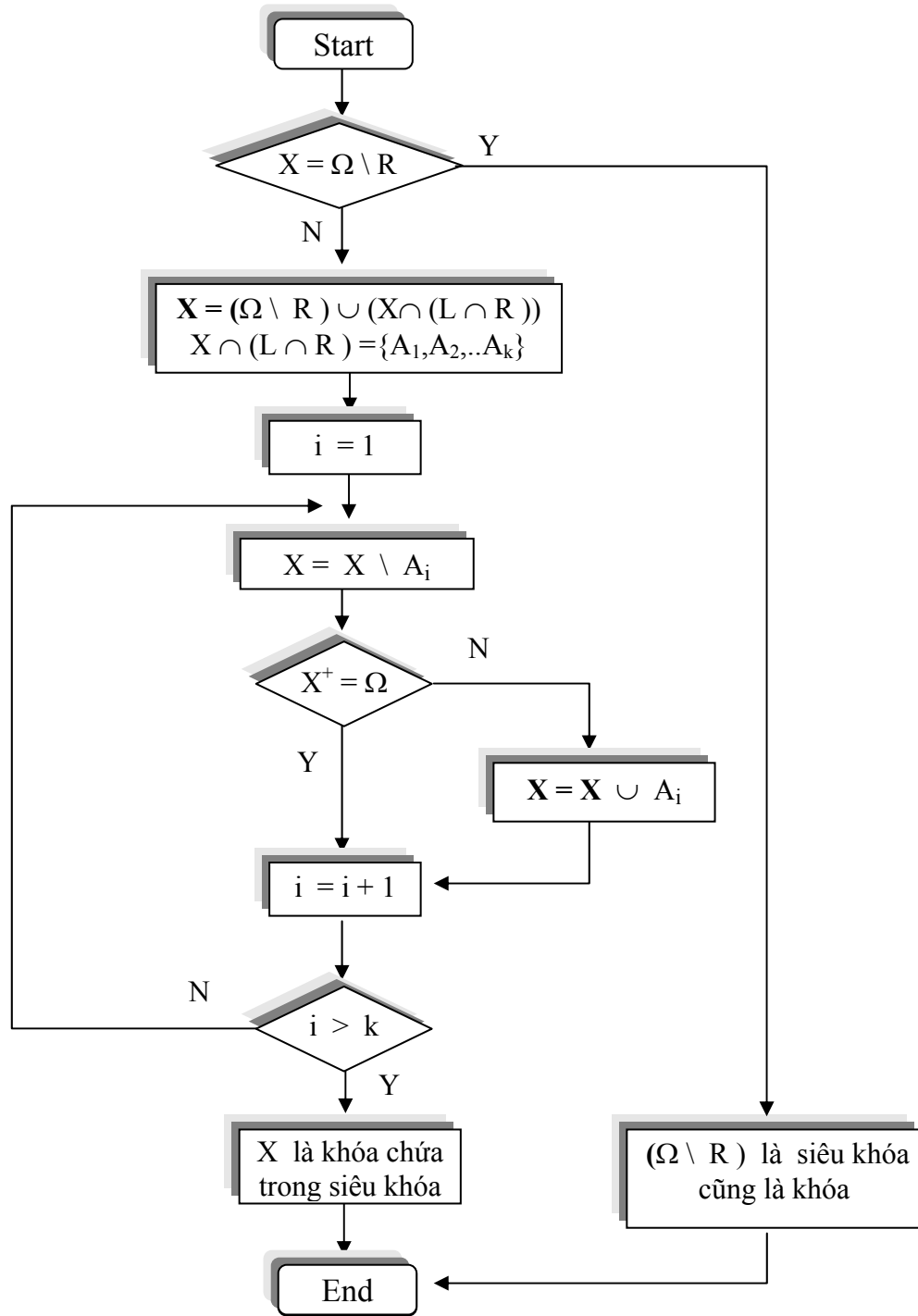
If $(X - A)^+ = \Omega$ then

$X = X - A$

End

X là khóa của lược đồ quan hệ $s = \langle \Omega, F \rangle$.

End.



Hình 3.8 Sơ đồ thuật toán xác định khóa từ siêu khóa cho trước

Vi dụ 16: $\Omega = \{A, B, C, D, E, G\}$
 $F = \{B \rightarrow C, C \rightarrow B, A \rightarrow GD\}$
 $L = ABC, R = BCD, (\Omega \setminus R) = AE, L \cap R = BC$
 Cho trước siêu khóa $X = ABCEG$

Khi đó $X := (\Omega \setminus R) \cup (X \cap (L \cap R)) = AEBC$

Kiểm tra trên tập $X \cap (L \cap R) = BC$

a) Xét $(X - B)^+ = (ACEG)^+ = ABCDEG = \Omega \Rightarrow X = ACE$

$(X - C)^+ = (AE)^+ = ADEG \neq \Omega$, như vậy $X = ACE$ là khóa.

b) Xét $(X - C)^+ = (ABE)^+ = ABCDEG = \Omega \Rightarrow X = ABE$

$(X - B)^+ = (AE)^+ = ADEG \neq \Omega$, như vậy $X = ABE$ là khóa.

Như vậy khóa của lược đồ quan hệ trên là $K_1 = ABE$ và $K_2 = ACE$.

Thuật toán 3.8 đoán nhận khóa

Cho X là tập thuộc tính bất kỳ, đoán nhận X có phải là khóa (siêu khóa không).

Input: $\Omega := \{A_1, A_2, \dots, A_n\} \neq \emptyset$

$F := \{L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$

$L = \bigcup_{L_j \subseteq \Omega} L_j$ và $R = \bigcup_{R_j \subseteq \Omega} R_j$

$X = \{A_1, A_2, \dots, A_k\}$

Output: X là siêu khóa (khóa) của lược đồ quan hệ.

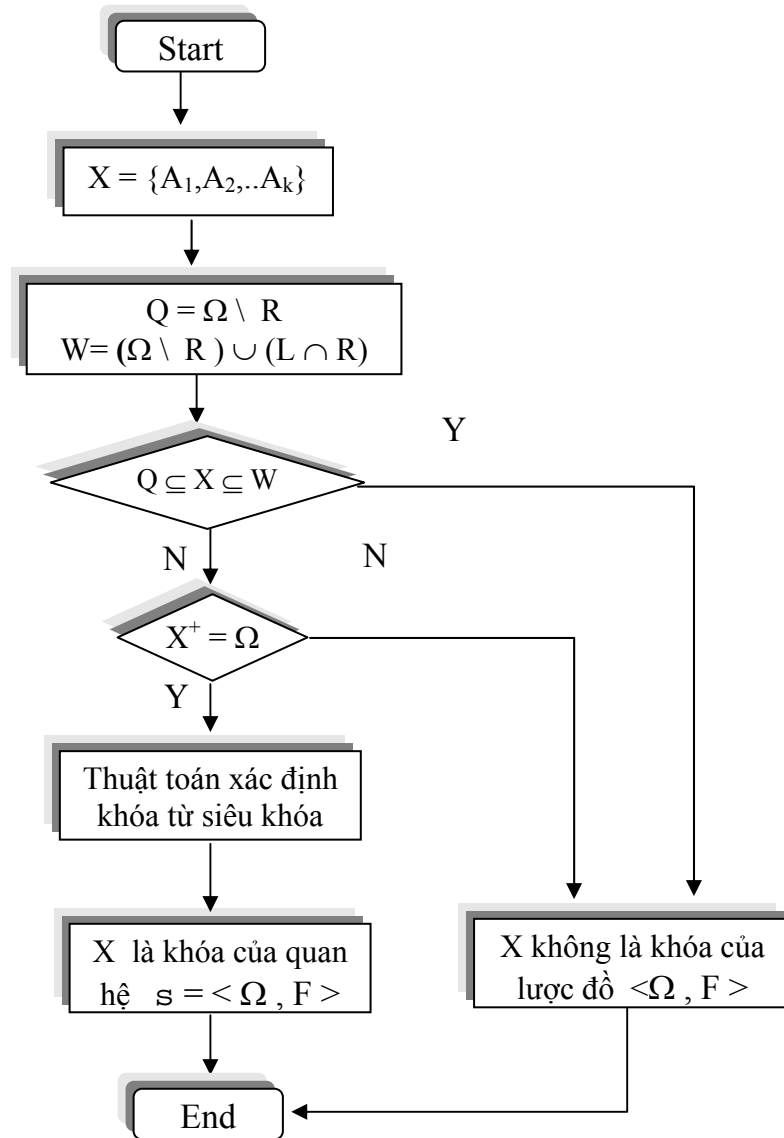
Phương pháp:

- Nếu $(\Omega \setminus R) \subseteq X \subseteq (\Omega \setminus R) \cup (L \cap R)$ khi đó X là siêu khóa.
- Nếu $X^+ \neq \Omega$ khi đó X không là siêu khóa.
- Nếu X là siêu khóa, sử dụng thuật toán xác định khóa từ siêu khóa.

Ví dụ 17: Cho $F = \{B \rightarrow C, C \rightarrow B, A \rightarrow GD\}$.

Xác định $X = ABEG$ có là khoá không ?.

- $L = ABC, R = BCD, (\Omega \setminus R) = AE, L \cap R = BC$.
- Khi đó $AE = (\Omega \setminus R) \subseteq ABEG \not\subseteq (\Omega \setminus R) \cup (L \cap R) = AEBC$.
- $X^+ = (ABEG)^+ = \Omega = \{A, B, C, D, E, G\}$, là một siêu khoá.
- Áp dụng thuật toán xác định khoá từ siêu khoá $X = ABEG$.
- Như vậy khóa của lược đồ quan hệ là $K_1 = ABE$.



Hình 3.9 Sơ đồ thuật toán đoán nhận khóa

Câu hỏi trắc nghiệm

1. R là một quan hệ trên tập Ω khi và chỉ khi:
 - A. $R \subseteq D(a_1) \times D(a_2) \times \dots \times D(a_n)$.
 - B. $R = D(a_1) \times D(a_2) \times \dots \times D(a_n)$.
 - C. $R = D(a_1) \cup D(a_2) \cup \dots \cup D(a_n)$.
 - D. $R \subseteq D(a_1) \cup D(a_2) \cup \dots \cup D(a_n)$.
2. $f: X \rightarrow Y$, khi và chỉ khi nếu
 - A. $(\forall r, s \in R) (r(X) \equiv s(X))$ thì suy ra $r(Y) \equiv s(Y)$,
 - B. $(\forall r, s \in R) ((\forall a \in X) (r(a) \subseteq s(a)))$ suy ra $(\forall b \in Y) (r(b) \subseteq s(b))$.
 - C. Khi đối số trùng nhau thì hàm có nhiều giá trị.
 - D. Một giá trị của Y được xác định bởi nhiều giá trị của X.

3. Quy tắc nào sau đây là quy tắc tách/hợp?
 - A. Nếu $B \subseteq A$ khi đó $A \rightarrow B$.
 - B. Nếu $A \rightarrow B$ và $C \subseteq \Omega$ khi đó $AC \rightarrow BC$.
 - C. Nếu $A \rightarrow B$ và $B \rightarrow C$ khi đó $A \rightarrow C$.
 - D. Nếu $A \rightarrow B$ và $A \rightarrow C$ khi đó $A \rightarrow BC$.
 4. $A \rightarrow B$ được suy dẫn logic từ F bằng cách
 - A. Áp dụng liên tiếp các tiên đề Armstrong.
 - B. Áp dụng các quy tắc tách/hợp.
 - C. Áp dụng các quy tắc phản xạ
 - D. Áp dụng các quy tắc bắc cầu
 5. Các tính chất nào sau đây là tính chất bao đóng thuộc tính:
 - A. $X \subseteq Y \Leftrightarrow X^+ \subseteq Y^+$
 - B. $X \rightarrow Y \Leftrightarrow Y^+ \subseteq X$
 - C. $X \rightarrow Y \Leftrightarrow Y^+ \subseteq X^+$
 - D. $X^+ = Y^+ \Leftrightarrow X \rightarrow Y$ và $Y \rightarrow X$
 6. X là khóa của lược đồ quan hệ khi và chỉ khi
 - A. $X^+ = \Omega$ và $(X - A)^+ \neq \Omega, \forall A \in X$.
 - B. $X^+ = \Omega$ và $(X - A)^+ = \Omega, \forall A \in X$.
 - C. $(X - A)^+ = \Omega, \forall A \in X$.
 - D. $(X - A)^+ \neq \Omega, \forall A \in X$.
 7. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Một quan hệ trên tập các thuộc tính Ω thường biểu diễn bằng Các thuộc tính là các cột A_1, A_2, \dots, A_n , các hàng là các bộ (tuple), hay gọi là các bản ghi.
 - A. Một bảng 2 chiều.
 - B. Biểu thức toán học.
 - C. Phụ thuộc hàm.
 - D. Một bảng n chiều
 8. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Ngôn ngữ con dữ liệu kiểu quan hệ là tập các phép toán có các thao tác trên các dữ liệu dưới dạng quan hệ. Dữ liệu được biểu diễn một cách duy nhất. Ngôn ngữ con dữ liệu gồm nhóm các phép toán tìm kiếm và nhóm các phép toán cập nhật, lưu trữ.
 - A. Cấu trúc quan hệ.
 - B. Cấu trúc dữ liệu.
 - C. Biểu thức quan hệ.
 - D. Quan hệ
 9. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Cho 2 tập các phụ thuộc hàm F và G cùng thoả trên một lược đồ s Nói rằng F và G tương đương, ký hiệu $F \cong G$, khi và chỉ khi.....
-
-

- A. $F^+ \equiv G^+$
B. Các phụ thuộc của F được suy dẫn logic từ G
C. Các phụ thuộc của G được suy dẫn logic từ F
D. $F \equiv G$
10. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Trong một lược đồ quan hệ luôn luôn tồn tại một có khả năng biểu diễn duy nhất các thuộc tính còn lại. Nói cách khác giá trị của một hay nhiều thuộc tính có thể xác định duy nhất giá trị các thuộc tính khác.
- A. Thuộc tính hay một tập các thuộc tính
B. Phụ thuộc hàm.
C. Bao đóng của tập thuộc tính
D. Phụ thuộc dư thừa
11. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Giá trị các thành phần của khoá không thể nhận
- A. Giá trị null .
B. Giá trị xác định.
C. Giá trị của các phụ thuộc hàm.
D. Giá trị của các thuộc tính tương ứng
12. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Mỗi một giá trị của khoá xác định..... của các thuộc tính không khoá. Giá trị khoá khác nhau thì giá trị các bộ có chứa giá trị khoá cũng khác nhau. Nếu loại bỏ một phần thông tin về khoá thì thông tin của các thuộc tính còn lại không thể xác định được.
- A. Đa trị.
B. Giá trị null .
C. Bao đóng
D. Duy nhất.giá trị
13. Hãy so khớp thông tin 2 phần dưới đây với nhau:

Phần 1

- 1 .*Phân xạ*: Nếu với mọi $B \subseteq A$...
2 *Gia tăng*: Nếu $A \rightarrow B$
3 *Bắc cầu*: Nếu $A \rightarrow B$ và $B \rightarrow C$
4 *Hợp và tách*: Nếu $A \rightarrow B$ và $B=CD$
- a. 1- D; 2- C ; 3 – B ; 4 – A .
b. 1- C; 2- B; 3 – A; 4 – D
c. 1- C; 2- D ; 3 – A; 4 – B .
d. 1- D; 2- B; 3 – A; 4 – C.

Phần 2

- A. $\Rightarrow A \rightarrow C$
B. $\Leftrightarrow A \rightarrow B$ và $A \rightarrow C$
C. $\Rightarrow A \rightarrow B$.
D. $\Rightarrow AC \rightarrow B$, $AC \rightarrow BC$.

14. Hãy so khớp thông tin 2 phần dưới đây với nhau:

Phần 1

- 1 Tập các phụ thuộc hàm
- 2 Phụ thuộc hàm
- 3 Các hệ tiên đề Armstrong
- 4 Quan hệ khác nhau nhưng họ đầy đủ

 - a. 1- D; 2- C ; 3 – B ; 4 – A .
 - b. 1- C; 2- B; 3 – A; 4 – D
 - c. 1- A; 2- B; 3 – C; 4 – D.
 - d. 1- C ; 2- D ; 3 – B ; 4 – A .

Phần 2

- A. các phụ thuộc hàm là như nhau
- B. là *đúng và đầy đủ*
- C. thỏa trên quan hệ R thì nó cũng thỏa trên mọi quan hệ trên tập Ω
- D. không phụ thuộc vào quan hệ cụ thể

15. Hãy so khớp thông tin 2 phần dưới đây với nhau:

Phần 1

- 1 $X \rightarrow Y$ là phụ thuộc dư thừa khi và chỉ khi $X \rightarrow Y$
- 2 $X \rightarrow Y$ được gọi là phụ thuộc không dư thừa khi và chỉ khi $X \rightarrow Y$
- 3 Một tập các phụ thuộc hàm bỏ đi những phụ thuộc dư thừa....
- 4 Một tập các phụ thuộc hàm bỏ đi những phụ thuộc bắc cầu.....

 - a. 1- D; 2- C ; 3 – A ; 4 – B .
 - b. 1- C; 2- B; 3 – A; 4 – D
 - c. 1- A; 2- B; 3 – C; 4 – D.
 - d. 1- D; 2- B; 3 – A; 4 – C.

Phần 2

- A. tập còn lại sẽ tương đương với tập đầu tiên.
- B. tập còn lại không tương đương với tập đầu tiên.
- C. không được suy dẫn logic từ $G := F - \{X \rightarrow Y\}$
- D. được suy dẫn logic từ $G := F - \{X \rightarrow Y\}$

Câu hỏi và bài tập

Câu hỏi

1. Định nghĩa quan hệ là gì, sự khác biệt giữa quan hệ với File truyền thống.
2. Hiểu thế nào là trùng nhau, khác nhau của 2 bộ của một quan hệ.
3. Trình bày vai trò của khái niệm phụ thuộc hàm trong lý thuyết cơ sở dữ liệu ?.
4. Định nghĩa phụ thuộc hàm và ví dụ minh họa.
5. Phụ thuộc đối ngẫu, phụ thuộc mạnh, phụ thuộc yếu.
6. Các hệ tiên đề các phụ thuộc hàm và các phép suy dẫn.
7. Chứng minh rằng nếu $F_R := \{L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$ là tập các phụ thuộc hàm thỏa trên một quan hệ R trên Ω thì nó cũng thỏa trên mọi quan hệ trên Ω .

8. Nếu một tập các phụ thuộc hàm F , tồn tại một quan hệ R trên tập các thuộc tính Ω sao F thỏa trên nó. Tức là $Y = F_R := \{ (A,B) \mid A,B \subseteq \Omega \ \& \ A \rightarrow B \}$.
 9. Trình bày và chứng minh một số tính chất của phụ thuộc hàm bằng:
 - a. Suy dẫn theo quan hệ (suy dẫn theo định nghĩa).
 - b. Suy dẫn logic từ F bằng cách áp dụng liên tiếp các tiên đề Armstrong.
 10. Phát biểu định nghĩa bao đóng và tính chất bao đóng các phụ thuộc hàm.
 11. Phát biểu định nghĩa bao đóng tập các thuộc tính (Closure of a set Attributes).
 12. Trình bày và chứng minh một số tính chất bao đóng thuộc tính.
 13. Nêu ý nghĩa của việc nghiên cứu bao đóng tập các thuộc tính.
 14. Thuật toán xác định bao đóng
 15. Lập trình xác định bao đóng của tập thuộc tính bất kỳ.
 16. Thuật toán xác định phụ thuộc hàm suy dẫn từ F .
 17. Nêu ý nghĩa thuật toán xác định phụ thuộc hàm suy dẫn từ F .
 18. Trình bày khái niệm và ý nghĩa phụ thuộc dư thừa.
 19. Tập các phụ thuộc tương đương và ý nghĩa.
 20. Thuật toán xác định tập các phụ thuộc không dư thừa.
 21. Lập trình xác định phụ thuộc dư thừa.
 22. Định nghĩa thuộc tính dư thừa và ý nghĩa khi nghiên cứu thuộc tính dư thừa.
 23. Mối quan hệ giữa thuộc tính dư thừa và phụ thuộc không hoàn toàn.
 24. Thuật toán loại bỏ các thuộc tính dư thừa về trái.
 25. Lập trình loại bỏ các thuộc tính dư thừa về trái.
 26. Định nghĩa tập cực tiểu và phủ cực tiểu.
 27. Chứng minh rằng tập các phụ thuộc hàm F đương đương với tập phụ thuộc hàm G trong đó các vế phải không chứa quá một thuộc tính.
 28. Chứng minh rằng tập các phụ thuộc hàm F có chứa một phủ cực tiểu.
 29. Thuật toán xác định phủ cực tiểu.
 30. Lập trình xác định tập phủ cực tiểu từ tập phụ thuộc hàm F cho trước.
 31. Định nghĩa khóa của lược đồ quan hệ, siêu khóa và vai trò của khái niệm khóa trong lý thuyết cơ sở dữ liệu. Ví dụ minh họa.
 32. Chứng minh rằng mọi siêu khóa bao giờ cũng chứa khóa.
 33. Khái niệm khóa dự bị, định thuộc (Determinant), phản khóa (AntiKey).
 34. Định nghĩa lược đồ thu gọn.
 35. Chứng minh rằng:
 - a. Nếu K^{-1}_S là tập phản khóa của $S = \langle \Omega, F \rangle$. Khi đó $K^{-1}_{S_1} = \{ K^{-1} \setminus G \mid K^{-1} \in K^{-1}_S \}$ là phản khóa của $S_1 := \langle \Omega_1, F_1 \rangle$.
 - b. Nếu $K^{-1}_{S_1}$ là tập phản khóa của $S_1 := \langle \Omega_1, F_1 \rangle$. Khi đó $K^{-1}_S = \{ K^{-1} \cup G \mid K^{-1} \in K^{-1}_{S_1} \}$ là tập phản khóa của $\langle \Omega, F \rangle$.
 36. Chứng minh:
$$\bigcup_{K \in K} K = \Omega \setminus \bigcap_{K^{-1} \in K^{-1}} K^{-1}$$
-

37. Mô tả một cách hình thức thuật toán xác định khoá từ một siêu khoá cho trước.
38. Lập trình xác định khoá từ một siêu khoá cho trước.
39. Thuật toán xác định một khoá lược đồ quan hệ.
40. Lập trình xác định một khoá lược đồ quan hệ.
41. Chứng minh rằng $\forall X \subseteq \Omega$ siêu khoá $\Leftrightarrow X^+ = \Omega$.
42. Chứng minh rằng nếu X là khoá của lược đồ quan hệ khi và chỉ khi $X^+ = \Omega$ và không tồn tại $Y \subset X$ sao cho $Y^+ = \Omega$.
43. Chứng minh rằng rằng giá trị của các thành phần của khoá không thể nhận các giá trị không xác định.
44. Hãy biểu diễn các phụ thuộc hàm bằng sơ đồ và xác định khoá của lược đồ quan hệ. Xác định một số siêu khoá của nó.
45. Cho ví dụ về tập các phụ thuộc hàm, khoá, siêu khoa, định thuộc và phản khoá của lược đồ quan hệ.
46. Cho lược đồ quan hệ $s = \langle \Omega, F \rangle$. Với $\forall K \in K$, khi đó :

$$\Omega \setminus R \subseteq K \subseteq (\Omega \setminus R) \cup (L \cap R).$$
47. Nếu K_1 và K_2 là hai khoá khác nhau của lược đồ quan hệ $s = \langle \Omega, F \rangle$. Khi đó:
 $(K_1 \setminus K_2) \subseteq L \cap R$ và $(K_2 \setminus K_1) \subseteq L \cap R$.
48. Chứng minh rằng: Với $\forall K \in K$, khi đó:
 $(\Omega \setminus R) \cup b(L, R) \subseteq K \subseteq (\Omega \setminus R) \cup (L \cap R) \setminus a(L, R)$
49. Chứng minh rằng: $\forall X \rightarrow Y \in F$ khi đó $X(\Omega \setminus XY)$ là một siêu khoá.
50. Cho $\Omega := \{a_1, a_2, \dots, a_n\}$, hãy xây dựng một lược đồ quan hệ $s = \langle \Omega, F \rangle$ có số khoá tối đa là số phần tử tổ hợp chập $[n/2]$ của n phân tử trên Ω .

Bài tập

1. Chứng minh tính bằng định nghĩa phụ thuộc hàm:
 - a. Nếu $A \rightarrow B$ và $C \subseteq \Omega$ khi đó $AC \rightarrow BC$.
 - b. Nếu $\{A_1, A_2, \dots, A_n\} \rightarrow B$ khi và chỉ khi $A_i \rightarrow B, i=1,2,\dots,n$.
 - c. $X \rightarrow Y \in F^+$ khi và chỉ khi $Y \subseteq X^+$.
2. Cho $F = \{AB \rightarrow E, AG \rightarrow I, E \rightarrow G, GI \rightarrow H\}$.
 Chứng minh rằng $AB \rightarrow GH \in F^+$?
3. Cho $F = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A\}$.
 Chứng minh rằng $AB \rightarrow E \in F^+$ và $AB \rightarrow G \in F^+$?
4. Cho $F = \{XY \rightarrow W, Y \rightarrow Z, WZ \rightarrow P, WP \rightarrow QR, Q \rightarrow X\}$.
 Chứng minh rằng $XY \rightarrow P \in F^+$ sử dụng các hệ tiên đề.
5. Cho $F = \{XY \rightarrow W, Y \rightarrow Z, WZ \rightarrow P, WP \rightarrow QR, Q \rightarrow X\}$.
 Chứng minh rằng $XY \rightarrow P \in F^+$ sử dụng định nghĩa bao đóng $\{XY\}^+$.
6. Cho Ω là tập các thuộc tính và F là tập các phụ thuộc hàm thoả trên Ω .
 Chứng minh $\forall X \subseteq \Omega$ và $\forall A \in X$, thì $X \rightarrow A \in F^+$. Tức là $\forall A \in X \subseteq \Omega$ Suy ra
 $X \rightarrow A \in F^+$.

7. Cho $\Omega = \{A, B, C, D\}$ và $F = \{A \rightarrow B, A \rightarrow C\}$ tập các phụ thuộc hàm. Hãy xác định các phụ thuộc sau phụ thuộc nào được suy dẫn logic từ F :
- a) $A \rightarrow B$ c) $BC \rightarrow A$ e) $AB \rightarrow B$
 b) $C \rightarrow D$ d) $A \rightarrow BC$
8. Cho $\Omega = \{A, B, C, D\}$ và $F = \{A \rightarrow B, BC \rightarrow D\}$ tập các phụ thuộc hàm. Hãy xác định các phụ thuộc sau không được suy dẫn logic từ F :
- a) $C \rightarrow D$ c) $A \rightarrow D$ e) $AD \rightarrow C$
 b) $BC \rightarrow A$ d) $B \rightarrow CD$
9. Cho $F = \{AB \rightarrow E, AG \rightarrow I, BE \rightarrow I, E \rightarrow G, GI \rightarrow H\}$. Chứng minh rằng $AB \rightarrow GH \in F^+$.
10. Cho $F = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A\}$. Chứng minh rằng $AB \rightarrow E \in F^+$ và $AB \rightarrow G \in F^+$.
11. Cho 2 tập các thuộc tính Ω_1 và Ω_2 , $X = \Omega_1 \cap \Omega_2$. Chứng minh rằng nếu quan hệ R trên tập các thuộc tính $\Omega_1 \Omega_2$ thỏa $X \rightarrow \Omega_2$ thì $R = R \bowtie R [\Omega_2]$
12. Cho $\Omega = \{TC\#, TTC, MC\#, DAI, NG, GT, SL, DV\}$ tập các thuộc tính quản lý mạng cáp:
- | | | | |
|------|------------------|-----|------------------|
| TC # | Mã tuyến cáp, | MC# | Mã cáp |
| TTC | Tên tuyến cáp, | DAI | Độ dài tuyến cáp |
| NG | Ngày lắp đặt cáp | DV | Đơn vị lắp đặt |
| GT | Giá trị cáp | SL | Số lượng cáp |
- Hãy xác định tập các phụ thuộc F^+ .
13. Cho $\Omega = \{A, B, C, D, E, G\}$ và $F = \{AB \rightarrow C, D \rightarrow EG, C \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG\}$.
 Tính X^+ với $X = \{AB\}, \{BC\}, \{CD\}, \{ACD\}, \{DEG\}$.
14. Cho $F = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A\}$. Tính bao đóng của X với $X = \{AB\}$ và chứng minh rằng $AB \rightarrow E \in F^+$.
15. Cho $\Omega = \{A, B, C, D, E, G, H\}$ và $F = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow H, G \rightarrow A\}$. Hãy xác định các thuộc tính X sao cho $\{X\}^+ = \Omega$.
16. Cho $\Omega = \{A, B, C, D, E, G\}$ và $F = \{AB \rightarrow C, D \rightarrow EG, C \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG\}$. Hãy xác định các thuộc tính X sao cho $\{X\}^+ = \Omega$.
17. Cho $F = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow H, G \rightarrow A\}$. Hãy xác định các thuộc tính X sao cho $\{X\}^+ = X \subset \Omega$.
18. Cho $F = \{AB \rightarrow E, AG \rightarrow I, BE \rightarrow I, E \rightarrow G, GI \rightarrow H\}$. Hãy xác định các thuộc tính X sao cho $\{X\}^+ = X \subset \Omega$.
19. Cho tập phụ thuộc hàm $F = \{X \rightarrow YW, XW \rightarrow Z, Z \rightarrow Y, XY \rightarrow Z\}$.
 $XY \rightarrow Z$ là phụ thuộc dư thừa của F ?

20. Cho tập phụ thuộc hàm $F = \{ X \rightarrow YW, XW \rightarrow Z, Z \rightarrow Y, XY \rightarrow Z \}$. Hãy xác định các phụ thuộc không dư thừa.
21. Cho $F = \{ AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A \}$. Hãy xác định các phụ thuộc dư thừa.
22. Cho $\Omega = \{ A, B, C, D, E, G \}$ và $F = \{ AB \rightarrow C, D \rightarrow EG, C \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG \}$. Hãy xác định các phụ thuộc dư thừa.
23. Cho $F = \{ X \rightarrow Z, XY \rightarrow WP, XY \rightarrow ZWQ, XZ \rightarrow R \}$. Hãy loại bỏ các thuộc tính dư thừa trong các vế trái.
24. Cho $F = \{ AB \rightarrow C, D \rightarrow EG, C \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG \}$. Hãy xác định các phụ thuộc có vế trái chứa các thuộc tính dư thừa.
25. Cho $F = \{ AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A, BCD \rightarrow D \}$. Hãy xác định các phụ thuộc có vế trái chứa các thuộc tính dư thừa.
26. Cho $F = \{ AB \rightarrow E, AG \rightarrow I, BE \rightarrow I, E \rightarrow G, GI \rightarrow H \}$. Hãy loại bỏ các thuộc tính dư thừa trong các vế trái.
27. Cho F và G là các tập phụ thuộc cùng thoả trên tập Ω . Ký hiệu X_G^+ và X_F^+ là các tập bao đóng ứng với các tập G và F . Chứng minh rằng nếu $F \cong G$ khi đó $X_G^+ = X_F^+$.
28. Cho $F = \{ AB \rightarrow C, D \rightarrow EG, C \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG \}$. Hãy xác định phủ cực tiểu của F .
29. Cho $F = \{ X \rightarrow Z, XY \rightarrow WP, XY \rightarrow ZWQ, XZ \rightarrow R \}$. Hãy xác định phủ cực tiểu của F .
30. Cho $F = \{ AB \rightarrow C, D \rightarrow EG, C \rightarrow A, BE \rightarrow C, BC \rightarrow D, CG \rightarrow BD, ACD \rightarrow B, CE \rightarrow AG \}$. Hãy xác định phủ cực tiểu của F .
31. Cho $F = \{ AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A, BCD \rightarrow D \}$. Hãy xác định phủ cực tiểu của F .
32. Cho $F = \{ AB \rightarrow E, AG \rightarrow I, BE \rightarrow I, E \rightarrow G, GI \rightarrow H \}$. Hãy xác định phủ cực tiểu của F .
33. Cho $\Omega = \{ TC\#, TTC, MC\#, DAI, NG, GT, SL, DV \}$ tập các thuộc tính quản lý mạng cáp:
TC# Mã tuyến cáp, MC# Mã cáp, TTC Tên tuyến cáp.
DAI Độ dài tuyến cáp, NG Ngày lắp đặt cáp, DV Đơn vị lắp đặt
GT Giá trị cáp, SL Số lượng cáp.
Hãy xác định phủ cực tiểu của F .
34. Cho $F = \{ X \rightarrow Z, XY \rightarrow WP, XY \rightarrow ZWQ, XZ \rightarrow R \}$. Xác định tập phủ cực tiểu từ tập F .
35. Cho tập các thuộc tính $\Omega = \{ A, B, C, D, E, G \}$ và tập các phụ thuộc hàm

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

Xác định tập khóa của lược đồ quan hệ trên.

36. Cho lược đồ quan hệ quản lý mạng cáp, gồm tập các thuộc tính (TC#, MC#, MC, GIA, SL, GTR). Xác định tập khoá.
37. Cho tập các thuộc tính $\Omega = \{A, B, C, D, E, H\}$ và tập các phụ thuộc hàm $F := \{A \rightarrow E, C \rightarrow D, E \rightarrow DH\}$. Chứng minh rằng $K = \{A, B, C\}$ là khoá duy nhất của lược đồ quan hệ.
38. Cho tập các thuộc tính $\Omega = \{A, B, C, D\}$ và tập các phụ thuộc hàm $F := \{AB \rightarrow C, D \rightarrow B, C \rightarrow ABD\}$. Xác định tập khóa của lược đồ quan hệ.
39. Cho $s = \langle \Omega, F \rangle$ là một lược đồ quan hệ, trong đó Ω là tập các thuộc tính và F tập các phụ thuộc hàm, K là tập các khoá của lược đồ.

Ký hiệu:

- $M := \{K - a \mid a \in K, K \in K\}$
- $F_N := \{b \notin K, \forall K \in K\}$ (các thuộc tính không khoá).
- $L := \{c^+ \mid c \in M\}$

Chứng minh rằng khi đó các mệnh đề sau là tương đương:

- a) Tập F không chứa các phụ thuộc có vế trái dư thừa.
 - b) $\forall B \in L$ thì $B \cap L = \emptyset$
 - c) $\forall B \in L$ và $b \in F_N$ thì $(B - b)^+ = B - b$.
40. Cho $s = \langle \Omega, F \rangle$ là một lược đồ quan hệ, F_N là tập các thuộc tính không khoá và K là tập các khoá. Đặt $G := \{B - F_N \mid B \in K^{-1}\}$ Chứng minh rằng các thuộc tính không khoá phụ thuộc hoàn toàn vào khoá khi và chỉ khi $\forall C \in G$ thì $C^+ = C$.
41. Chứng minh rằng nếu $X \subseteq \Omega$ là khoá của lược đồ quan hệ $s = \langle \Omega, F \rangle$ khi và chỉ khi $X^+ = \Omega$ và tất cả các tập con thực sự Z của X khi đó $Z^+ \neq \Omega$.
42. Cho tập các thuộc tính quản lý tổng đài- cước thuê bao: TD#, TD, DL, MC#, SDT, NSX, NG, GTR, T, TB, TB#, DC, SDTD, NGG, GB, KT, DD. Trong đó:
- | | | | | | |
|------|-----------------------|-----|----------------|-----|-----------------|
| TD # | Mã tổng đài, | MC# | Màu cáp, | TD | Tên tổng đài. |
| NG | Ngày lắp đặt, | DL | Dung lượng, | VT | Vị trí lắp đặt. |
| GTR | Giá trị tổng đài, | NGG | Ngày gọi, | NSX | Nước sản xuất. |
| TB | Thuê bao, | SDT | Số điện thoại, | TB# | Mã thuê bao |
| SDTD | Số điện thoại đi/đến, | GB | Giờ bắt đầu, | DD | đi/đến |
| KT | Giờ kết thúc | | | | |
- a) Xác định tập khoá quan hệ
 - b) Xác định phủ cực tiểu của tập các phụ thuộc hàm.
 - c) Xác định tập phân khoá.
 - d) Dùng sơ đồ biểu diễn các phụ thuộc hàm.

4 KỸ THUẬT THIẾT KẾ CÁC HỆ CƠ SỞ DỮ LIỆU QUAN HỆ

Chương này sẽ giới thiệu nguyên lý thiết kế và cài đặt các hệ cơ sở dữ liệu quan hệ. Kỹ thuật chuyển đổi một quan hệ chưa chuẩn hóa về một nhóm các quan hệ ở dạng chuẩn 3NF không bị tổn thất thông tin, trong một số trường hợp các phép tách vẫn bảo toàn các phụ thuộc. Nội dung của chương bao gồm:

- Tách một lược đồ quan hệ
- Các dạng chuẩn hóa tiếp dữ liệu.

4.1 Mở đầu

Mục tiêu của lý thuyết CSDL là tính độc lập của dữ liệu. Cấu trúc lưu trữ các hệ cơ sở dữ liệu phản ánh tính hiện thực, khách quan và tính toàn vẹn dữ liệu. Vì vậy trong quá trình chuẩn hoá dữ liệu và tìm kiếm thông tin, cần thiết phải thực hiện các phép tách lược đồ quan hệ chưa chuẩn hoá về tập các lược đồ quan hệ chiếu đã được chuẩn hoá, sao cho quá trình tách không làm tổn thất thông tin (lossless- mất mát thông tin), theo nghĩa các quan hệ gốc được khôi phục chính xác từ phép kết nối tự nhiên của các quan hệ chiếu.

Tách - kết nối các lược đồ quan hệ có làm tổn thất thông tin hay không, có bảo toàn các phụ thuộc hay không đã được nhiều người quan tâm nghiên cứu, giải quyết. A.V. Ho, C. Beeri & J.D. Ullman giới thiệu thuật toán xác định phép kết nối các lược đồ quan hệ không có tổn thất thông tin với giả thiết các phụ thuộc dữ liệu là các phụ thuộc hàm. Các ông cũng đã mở rộng vấn đề này cho các trường hợp phụ thuộc dữ liệu là phụ thuộc đa trị.

4.2 Phép tách – kết nối không tổn thất thông tin

4.2.1 Phép tách

Cho $s = \langle \Omega, F \rangle$ là một lược đồ quan hệ, trong đó $\Omega = \{A_1, A_2, \dots, A_n\}$ là tập các thuộc tính và F là tập các phụ thuộc hàm. Gọi $\varphi[\Omega_1, \Omega_2, \dots, \Omega_p]$ là một phép tách (hay còn gọi là một phân hoạch) của $s = \langle \Omega, F \rangle$, nếu:

- $\Omega_i \subseteq \Omega, i=1 \div p$
- $\Omega = \Omega_1 \cup \dots \cup \Omega_p$
- $F_i := F \mid_{\Omega_i} := \pi_{\Omega_i}(F) := \{X \rightarrow Y \in F, XY \subseteq \Omega_i\}, i = 1 \div p.$
- $s_i := \langle \Omega_i, F_i \rangle = \pi_{\Omega_i}(S), i = 1 \div p.$

Như vậy, nếu $\varphi[\Omega_1, \Omega_2, \dots, \Omega_p]$ là một phép tách của $s = \langle \Omega, F \rangle$, khi đó tập các phụ thuộc $F_i := F \mid_{\Omega_i} = \pi_{\Omega_i}(F)$ được gọi là tập các phụ thuộc chiếu F trên các tập thuộc tính tương ứng Ω_i . Và các lược đồ $s_i = \langle \Omega_i, F_i \rangle = \pi_{\Omega_i}(S)$ gọi là các lược đồ chiếu trên các tập thuộc tính Ω_i với $i = 1 \div p$. Nếu R là một quan hệ trên tập các thuộc tính Ω , khi đó các quan hệ chiếu sẽ là $R_{\Omega_i} := \pi_{\Omega_i}(R), i = 1 \div p$, nghĩa là các quan hệ chiếu $\pi_{\Omega_i}(R)$ chỉ bao gồm các thuộc tính $\Omega_i, i = 1 \div p$.

4.2.2 Tách - kết nối tự nhiên

Phép tách $\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ được gọi là phép tách - kết nối tự nhiên của của lược đồ quan hệ $s = \langle \Omega, F \rangle$, nếu:

- a) $\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ là một phép tách của $s = \langle \Omega, F \rangle$.
- b) Kết quả của phép kết nối tự nhiên của các lược đồ chiếu $\pi_{\Omega_i}(S)$, $i = 1 \div p$, là một lược đồ $m_\varphi(s)$ trên các thuộc tính $\Omega = \Omega_1 \cup \Omega_2 \cup \dots \cup \Omega_p$.

$$m_\varphi(s) := \pi_{\Omega_1}(s) \bowtie \pi_{\Omega_2}(s) \bowtie \dots \bowtie \pi_{\Omega_p}(s) = s_1 \bowtie s_2 \bowtie \dots \bowtie s_p.$$

Nghĩa là với mọi quan hệ $R \in s = \langle \Omega, F \rangle$, khi đó $m_\varphi(R)$ là kết quả phép kết nối tự nhiên của các quan hệ chiếu tương ứng $R_i := R_{\Omega_i} := \pi_{\Omega_i}(R)$, $i = 1 \div p$, được biểu diễn như sau:

$$R \subseteq m_\varphi(R) := \pi_{\Omega_1}(R) \bowtie \pi_{\Omega_2}(R) \bowtie \dots \bowtie \pi_{\Omega_p}(R).$$

Từ định nghĩa trên có thể suy ra, nếu một thể hiện $I \in m_\varphi(s)$ khi đó:

$$\bigtimes_{i=1}^p \pi_{\Omega_i}(I) = \{a_1, a_2, \dots, a_p\} \mid \text{Nếu } A_j \in \Omega_i, \text{ tại vị trí } j \text{ ứng } R_i \text{ nhận giá trị } a_j, \text{ các vị trí khác nhận giá trị khác } a_j, i = 1 \div p \}.$$

4.2.3 Phép tách không tổn thất thông tin

Phép tách $\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ được gọi phép tách không tổn thất thông tin của lược đồ quan hệ $s = \langle \Omega, F \rangle$, nếu $\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ là phép tách - kết nối tự nhiên và:

$$S = m_\varphi(s) := \pi_{\Omega_1}(s) \bowtie \pi_{\Omega_2}(s) \bowtie \dots \bowtie \pi_{\Omega_p}(s) = s_1 \bowtie s_2 \bowtie \dots \bowtie s_p.$$

Nghĩa là với mọi quan hệ $R \in s$, khi đó R được khôi phục chính xác từ phép kết nối tự nhiên của các quan hệ chiếu $R_i = \pi_{\Omega_i}(R)$, $i = 1 \div p$.

$R = \pi_{\Omega_1}(R) \bowtie \pi_{\Omega_2}(R) \bowtie \dots \bowtie \pi_{\Omega_p}(R) := R_1 \bowtie R_2 \bowtie \dots \bowtie R_p$. Thông tin của một quan hệ R bất kỳ có thể nhận được từ các quan hệ chiếu R_i ứng với phép tách φ .

Ví dụ : Tách quan hệ không tổn thất thông tin

Lược đồ quan hệ quản lý phát hành báo chí QLBC gồm các thuộc tính:

$$\Omega = \{MK\#, TK, DC, MB\#, TB, GIA, SL\} \text{ và}$$

$$F = \{MK\# \rightarrow TK, MK\# \rightarrow DC, MB\# \rightarrow TB, MB\# \rightarrow GIA, (MK\#, MB\#) \rightarrow SL\}.$$

- MK#: Mã khách hàng
- TK : Tên khách hàng
- DC : Địa chỉ khách hàng
- MB#: Mã báo, tạp chí
- TB : Tên báo, tạp chí
- GIA: Đơn giá báo, tạp chí
- SL : Số lượng báo, tạp chí khách đặt mua

Trong lược đồ quan hệ QLBC, các thông tin về tên khách (TK), địa chỉ (DC), tên báo (TB) .. lặp lại rất nhiều lần trong các quan hệ thể hiện, đó là nguyên nhân dẫn đến sự xuất hiện các bất thường, nhập nhằng thông tin. Phép tách φ được mô tả dưới đây, sẽ tách lược đồ QLBC thành 3 lược đồ chiếu. Lược đồ $s_3 = \langle \Omega_3, F_3 \rangle$ chỉ cần lưu trữ thông tin về số lượng các loại báo của mỗi một khách hàng đặt mua. Lược đồ quan hệ $s_1 = \langle \Omega_1, F_1 \rangle$ lưu

trữ thông tin về khách đặt mua báo, và tương tự trong lược đồ quan hệ $s_2 = \langle \Omega_2, F_2 \rangle$ lưu trữ thông tin về các loại báo. Có thể kiểm tra phép tách φ không tổn thất thông tin và bảo toàn được các phụ thuộc hàm.

Phép tách $\varphi [\Omega_1, \Omega_2, \Omega_3]$:

- $\Omega_1 = \{M\#, TK, DC\}$, $F_1 = \{MK\# \rightarrow TK, MK\# \rightarrow DC\}$.
- $\Omega_2 = \{MB\#, TB, GIA\}$, $F_2 = \{MB\# \rightarrow TB, MB\# \rightarrow GIA\}$.
- $\Omega_3 = \{M\#, MB\#, SL\}$, $F_3 = \{(MK\#, MB\#) \rightarrow SL\}$.

Như vậy mục tiêu của phép tách lược đồ quan hệ là nhằm loại bỏ các dị thường thông tin khi thực hiện các phép lưu trữ như chèn thêm, loại bỏ hay sửa đổi thông tin trong các quan hệ lưu trữ. Tuy nhiên khi thực hiện phép tách, thông tin của lược đồ quan hệ có bị tổn thất hay không. Nói cách khác nếu kết nối tự nhiên các thành phần lược đồ quan hệ chiếu, liệu thông tin của lược đồ quan hệ gốc có tồn thất thông tin hay không, các phụ thuộc hàm có được bảo toàn hay không?

Ví dụ: Thí dụ sau mô tả phép tách tổn thất thông tin và không tổn thất thông tin:

Cho $\Omega = \{X, Y, Z\}$ tập các thuộc tính và

Giả sử quan hệ gốc: R và các quan hệ tách: R1 và R2

R		
X	Y	Z
x1	y1	z1
x2	y2	z2
x3	y2	z3
x4	y3	z4

⇒

R1	
X	Y
x1	y1
x2	y2
x3	y2
x4	y3

R2	
Y	Z
y1	Z1
y2	Z2
y2	Z3
v3	Z4

a) Phép tách – kết nối tổn thất thông tin:

R		
X	Y	Z
x1	y1	z1
x2	y2	z2
x3	y2	z3
x4	y3	z4

⇒

R1	
X	Y
x1	y1
x2	y2
x3	y2
x4	y3

R2	
Y	Z
y1	Z1
y2	Z2
y2	Z3
y3	Z4

⇒

$R \subseteq R1 \times R2$		
X	Y	Z
x1	y1	z1
x2	y2	z2
x2	y2	z3
x3	y2	z2
x3	y3	z3
x4	y4	z4

b) Phép tách – kết nối không tổn thất thông tin:

R		
X	Y	Z
x1	y1	z1
x2	y2	z2
x3	y2	z3
x4	y3	z4

⇒

R1	
X	Y
x1	y1
x2	y2
x3	y2
x4	y3

R2	
Y	Z
y1	Z1
y2	Z2
y2	Z3
y3	Z4

⇒

$R = R1 \times R2$		
X	Y	Z
x1	y1	z1
x2	y2	z2
x2	y2	z3
x3	y2	z2
x3	y3	z3
x4	y4	z4

Hình 4.1 Tách –kết nối tổn thất thông tin và không tổn thất thông tin

4.3 Thuật toán kiểm tra tách không tổn thất thông tin

Input: $s = \langle \Omega, F \rangle$ là một lược đồ quan hệ.

$\Omega = \{A_1, A_2, \dots, A_n\}$ tập các thuộc tính.

$F = \{f: L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$ tập các phụ thuộc hàm.

$\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ là một phép tách

Output: Một khẳng định phép tách - kết nối φ có tổn thất thông tin hay không.

Phương pháp:

- Tạo một bảng gồm n cột và p dòng. Cột thứ j tương ứng với thuộc tính A_j , hàng thứ i tương ứng với lược đồ quan hệ chiều R_i :

- Cột : A_1, A_2, \dots, A_n
- Hàng: R_1, R_2, \dots, R_p

- Các phần tử của bảng:

$$\alpha(i,j) = \begin{cases} a_i & \text{nếu } A_i \in \Omega_i \\ b_{ij} & \text{nếu } A_i \notin \Omega_i \end{cases}$$

$i = 1 \div n, j = 1 \div p.$

- Áp dụng các phụ thuộc $X \rightarrow Y \in F$ để thay đổi các giá trị của bảng như sau: tìm các hàng giống nhau trên trong các cột thuộc tính của X , trong các cột thuộc tính Y nếu có giá trị là a sẽ thay giá trị các cột trong Y là a_i , nếu không có a_i , thay thế bằng b_{ij} .
- Xét lặp các phụ thuộc trong F cho đến khi không có sự thay đổi trong bảng.
- Việc duyệt bảng bao gồm sắp xếp bảng theo cột có tính các thuộc tính xuất hiện về trái của phụ thuộc hàm. Nếu có k thuộc tính như vậy thì việc thực hiện sắp xếp cần thực hiện trong $n * k$ bước.
- Điền các ký hiệu trong các cột có thuộc tính xuất hiện về phải của phụ thuộc hàm nếu các hàng bằng nhau trên về trái. Công việc này cần $O(k)$ thời gian cho mỗi phụ thuộc. Tổng tất cả độ dài về trái của tất cả các phụ thuộc hàm trong một lần duyệt không quá n , nên toàn bộ thời gian cho một lần duyệt nhiều nhất là $k*n$.
- Khi không còn một ký hiệu nào được làm bằng nhau trong một lần duyệt thì có thể kết thúc việc lặp các bước duyệt vì bảng thu được thoả mọi phụ thuộc.
- Kiểm tra có tồn tại một hàng R_i sao cho giá trị của chứa các ký hiệu a_1, a_2, \dots, a_n hay không. Nếu có, tách - kết nối không tổn thất thông tin. Ngược lại, không tồn tại dòng nào như vậy, nghĩa là các lược đồ quan hệ chiều khi kết nối bị tổn thất thông tin. Điều này có thể suy ra từ định nghĩa của phép tách – kết nối tự nhiên.
- Do đó thời gian tiêu dùng toàn bộ cho thuật toán nhiều nhất là $k*n^2*p$, Nếu $k \leq n$ và $p \leq n$ hiển nhiên thuật toán có thời gian chi phí nhiều nhất là n^4 .

Ví dụ: Cho $\Omega := \{A, B, C, D, E, F\}$ tập các thuộc tính.

Xét phép tách – kết nối $\varphi[\Omega_1, \Omega_2, \Omega_3]$ trong đó:

$$\Omega_1 := \{A, B, D, E\}, \quad \Omega_3 := \{B, C, E, F\},$$

$$\Omega_2 := \{A, C, D, F\} \quad \text{và} \quad F := \{A \rightarrow B, F \rightarrow E\}.$$

Bước 1: Thành lập bảng ban đầu gồm 3 hàng và 6 cột:

	A	B	C	D	E	F
R_1	a_1	a_2	b_{13}	a_4	a_5	b_{16}
R_2	a_1	b_{22}	a_3	a_4	b_{25}	a_6
R_3	b_{31}	a_2	a_3	b_{34}	a_5	a_6

Bước 2: Áp dụng $A \rightarrow B$ suy ra $b_{22} = a_2$

	A	B	C	D	E	F
R_1	a_1	a_2	b_{13}	a_4	a_5	b_{16}
R_2	a_1	a_2	a_3	a_4	b_{25}	a_6
R_3	b_{31}	a_2	a_3	b_{34}	a_5	a_6

Bước 3: Áp dụng $F \rightarrow E$ suy ra $b_{25} = a_5$

	A	B	C	D	E	F
R_1	a_1	a_2	b_{13}	a_4	a_5	b_{16}
R_2	a_1	a_2	a_3	a_4	a_5	a_6
R_3	b_{31}	a_2	a_3	b_{34}	a_5	a_6

Như vậy tồn tại hàng thứ 2 R_2 chứa các ký tự $\{a_1, a_2, a_3, a_4, a_5, a_6\}$. Suy ra phép tách φ có kết nối không tổn thất thông tin.

Ví dụ: Cho $\Omega := \{A, B, C, D, E\}$ tập các thuộc tính.

$$\Omega_1 := \{A, D\}, \quad \Omega_2 := \{A, B\}$$

$$\Omega_3 := \{B, E\}, \quad \Omega_4 := \{C, D, E\}$$

$$\Omega_5 := \{A, E\}$$

$$F := \{A \rightarrow C, B \rightarrow C, C \rightarrow D, E \rightarrow C, CE \rightarrow D\}.$$

Bước 1: Bảng ban đầu gồm 6 hàng và 5 cột:

	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{23}	b_{24}	b_{25}
R_3	b_{31}	a_2	b_{33}	b_{34}	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{53}	b_{53}	b_{54}	a_5

Bước 2: Áp dụng $A \rightarrow C$ và $B \rightarrow C$ suy ra $b_{13} = b_{23} = b_{53} = b_{33}$

	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{13}	b_{24}	b_{25}
R_3	b_{31}	a_2	b_{13}	b_{34}	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{53}	b_{13}	b_{54}	a_5

Bước 3: Áp dụng $C \rightarrow D$ suy ra $a_4 = b_{24} = b_{34} = b_{54}$

	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{13}	a_4	b_{25}
R_3	b_{31}	a_2	b_{13}	a_4	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{53}	b_{13}	a_4	a_5

Bước 4: Áp dụng $DE \rightarrow C$ suy ra $a_3 = b_{13}$

	A	B	C	D	E
R_1	a_1	b_{12}	a_3	a_4	b_{15}
R_2	a_1	a_2	a_3	a_4	b_{25}
R_3	b_{31}	a_2	a_3	a_4	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{53}	a_3	a_4	a_5

Bước 5: Áp dụng $CE \rightarrow A$ suy ra $a_1 = b_{41} = b_{31}$

	A	B	C	D	E
R_1	a_1	b_{12}	a_3	a_4	b_{15}
R_2	a_1	a_2	a_3	a_4	b_{25}
R_3	a_1	a_2	a_3	a_4	a_5
R_4	a_1	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{53}	a_3	a_4	a_5

Như vậy tồn tại hàng thứ 3 R_3 có chứa các ký tự $\{a_1, a_2, a_3, a_4, a_5\}$. Suy ra các phép tách - kết nối không tổn thất thông tin.

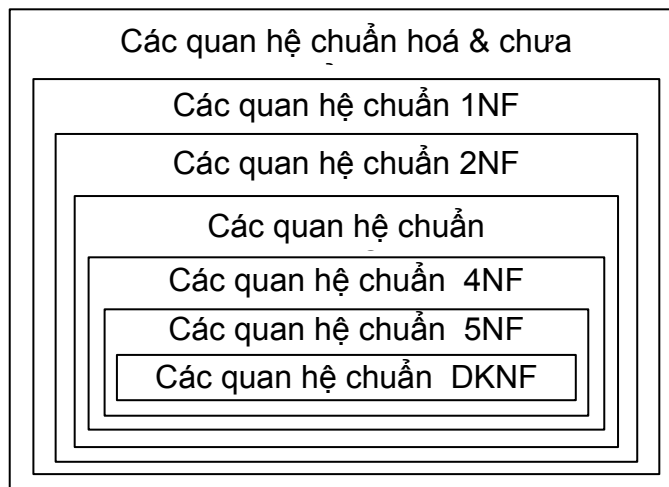
4.4 Các dạng chuẩn của lược đồ quan hệ

Phần này sẽ tìm hiểu và nghiên cứu khái niệm chuẩn hoá dữ liệu quan hệ. Đã có nhiều kết quả trong lĩnh vực chuẩn hoá, song vẫn còn nhiều vấn đề bất lợi khi tìm kiếm thông tin. Vì vậy người ta cũng chỉ mới đề xuất các dạng sao cho tiện lợi, đơn giản hơn trong quá trình thao tác các ngôn ngữ con dữ liệu.

Trong “ Further Normalization of the Data base relational Model” lần đầu tiên E.F. Codd đã đưa ra các khái niệm về chuẩn hoá quan hệ. Ông đã chia các quan hệ thành 3 lớp và gọi chúng là các quan hệ dạng chuẩn thứ nhất, dạng chuẩn thứ hai và dạng chuẩn thứ ba. Trong “ Multivalued Dependencies and a New Normal Form Relational Databaseses“ R. Fagin đã đưa ra khái niệm dạng chuẩn thứ tư, theo ông đó là dạng tiện lợi hơn so với quan hệ dạng chuẩn 3. Mục đích của chương này là minh hoạ những ưu điểm của các quan hệ dạng chuẩn 4 và trình bày phương pháp chuyển đổi một quan hệ chưa là dạng chuẩn 4 về một nhóm tương đương các quan hệ dạng chuẩn 4.

4.4.1 Sự cần thiết chuẩn hoá dữ liệu

Khi thực hiện các phép lưu trữ trên các quan hệ chưa được chuẩn hoá thường xuất hiện các dị thường thông tin. Nghĩa là trong dữ liệu lưu trữ, sự dư thừa, mất dữ liệu, mâu thuẫn hay không nhất quán dữ liệu có thể xảy ra khi cập nhật, bổ sung hay sửa đổi dữ liệu. Dị thường thông tin là nguyên nhân gây cản trở cho việc tìm kiếm, hỏi đáp thông tin. Mục tiêu của chuẩn hoá dữ liệu là triệt tiêu mức cao nhất khả năng xuất hiện các dị thường thông tin khi thực hiện các phép lưu trữ. Có như vậy mục tiêu của các hệ cơ sở mới được bảo đảm. Dữ liệu lưu trữ phản ánh thế giới hiện thực khách quan, đầy đủ hơn và sinh động hơn.



Hình 4.2 Mối quan hệ giữa các lớp dạng chuẩn lược đồ quan hệ

Khi thiết kế và cài đặt các hệ CSDL, chuẩn hoá là quá trình khảo sát danh sách các thuộc tính và áp dụng tập các quy tắc phân tích vào danh sách đó, biến đổi chúng thành nhiều tập nhỏ hơn sao cho:

- *Tối thiểu việc lặp lại.*
- *Tránh dị thường thông tin.*
- *Xác định và giải quyết được sự không rõ ràng, nhập nhằng trong suy diễn.*

Quá trình chuẩn hoá là quá trình tách lược đồ quan hệ về một nhóm tương đương các lược đồ quan hệ chiếu sao cho khi kết nối tự nhiên không làm tổn thất thông tin và bảo toàn được

các phụ thuộc hàm. Cơ sở chuẩn hoá dựa trên các khái niệm về phụ thuộc hàm, phụ thuộc đầy đủ, khoá, các thuộc tính không khoá... Một mô hình được xem là mô hình chuẩn hoá tốt, lý tưởng là mô hình ở đó mỗi một thuộc tính không khoá phụ thuộc hàm vào khoá.

Quan hệ chuẩn hoá là những quan hệ mà mỗi giá trị thuộc tính trong bộ là những thuộc tính nguyên tố, không phân chia ra được. Nói cách khác, mỗi toạ độ của hàng và cột trong quan hệ chỉ có đúng một giá trị chứ không phải là một tập các giá trị.

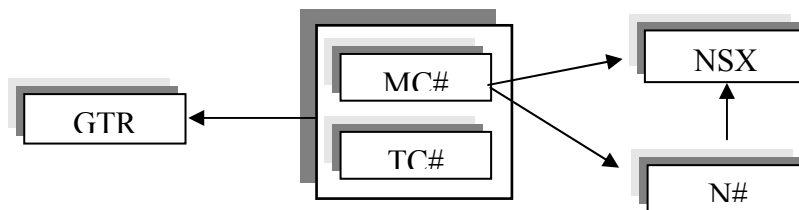
Ví dụ Lược đồ quan hệ QLCAP(TC#,GTR,MC#,N#,NSX)

Trong đó: TC# : Mã tuyến cáp GTR : Giá trị của cáp
 MC# : Mã cáp N# : Mã nước sản xuất
 NSX : Nước sản xuất.

Ngữ nghĩa dữ liệu như sau:

- Trong một tuyến cáp, giá trị của một loại cáp được xác định duy nhất.
- Mỗi mã cáp xác định mã nước sản xuất cáp, và
- Mã nước xác định tên nước sản xuất

$F = \{(TC\#,MC\#) \rightarrow GTR, MC\# \rightarrow NSX, MC\# \rightarrow N\#, N\# \rightarrow NSX\}$ tập các phụ thuộc hàm. Các thuộc tính khoá: (TC#,MC#) và các thuộc tính không khoá: GTR, N#, NSX.



Hình 4.3 Sơ đồ các phụ thuộc hàm trong lược đồ quan hệ QLCAP

TC#	GTR	MC#	N#	NSX
T01	200	C01	HAQ	Hàn Quốc
T01	250	C02	HAQ	Hàn Quốc
T01	220	C03	VTN	Việt Nam
T02	500	C01	HAQ	Hàn Quốc
T02	400	C04	JAN	Nhật Bản
T03	100	C05	RUS	Nga
T04	400	C06	CHN	Trung Quốc
T04	450	C03	VTN	Việt Nam

Hình 4.4 Một thể hiện của lược đồ quan hệ QLCAP

4.4.2 Dạng chuẩn 1 – 1NF (First Normal Form)

Lược đồ quan hệ $s = \langle \Omega, F \rangle$ được gọi là dạng chuẩn 1 – 1NF, khi và chỉ khi các thuộc tính chỉ chứa các giá trị nguyên tố (giá trị nguyên tố là giá trị không thể tách).

Thường các quan hệ chuẩn hoá là các quan hệ dạng chuẩn 1NF. Tuy nhiên, cấu trúc biểu diễn dữ liệu trong các quan hệ dạng 1NF còn nhiều điều bất tiện. Vì vậy khi thao tác thực

Hiện các phép chèn thêm, sửa đổi hay bổ sung cập nhật dữ liệu thường xuất hiện dị thường thông tin, không thể chấp nhận được trong quá trình tìm kiếm.

Theo định nghĩa, quan hệ QLCAP là một quan hệ dạng chuẩn thứ nhất 1NF.

Ví dụ: Xét quan hệ $S(S\#,PRO)$, thấy rằng thuộc tính PRO chứa các giá trị không nguyên tố. Vì vậy quan hệ này không phải là quan hệ dạng chuẩn 1NF.

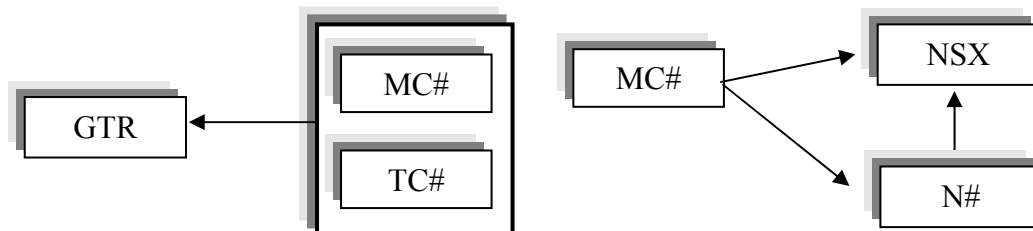
S#	PRO	
	P#	QTY
S1	100	1
	200	1
	300	2
S2	100	2
	200	2
S3	300	3
	100	1

Hình 4.5 Một thí dụ quan hệ không 1NF

4.4.3 Dạng chuẩn 2 – 2NF (Second Normal Form)

Lược đồ quan hệ $s = \langle \Omega, F \rangle$ được gọi là dạng chuẩn 2NF, khi và chỉ khi nó là dạng chuẩn 1NF và các thuộc tính không khoá phụ thuộc hàm đầy đủ vào khoá. Nói cách khác, nếu $s = \langle \Omega, F \rangle$ là dạng chuẩn 1NF và không tồn tại các phụ thuộc hàm $X \rightarrow Y \in F^+$ sao cho X là tập con thực sự của khoá và Y là thuộc tính không khoá.

Lược đồ quan hệ QLCAP không là dạng chuẩn 2NF, vì các thuộc tính không khoá N# và NSX phụ thuộc hàm vào $MC\# \subset \{TC\#, MC\#\}$, tức là MC# phụ thuộc không đầy đủ vào khoá $\{TC\#, MC\#\}$. Với cấu trúc lưu trữ của QLCAP, khi thực hiện các phép chèn thêm, loại bỏ hay sửa đổi dữ liệu sẽ xuất hiện dị thường thông tin. Không thể chèn thêm thông tin về một loại cáp mới khi chưa xác định giá trị của tuyến cáp, vì TC# không thể nhận giá trị không xác định. Tương tự, không thể xoá một tuyến duy nhất có một loại cáp duy nhất, vì như vậy sẽ mất thông tin về tuyến cáp và loại cáp đó. Có thể tách QLCAP thành 2 lược đồ quan hệ TUYEN và CAP_NSX về dạng chuẩn 2NF không tổn thất thông tin. Cấu trúc lưu trữ của TUYEN và CAP_NSX phản ánh thể giới dữ liệu khách quan hơn, trung thực hơn so với cấu trúc của QLCAP.



Hình 4.6 TUYEN

CAP_NSX

Ví dụ Cho $\Omega = \{A, B, C, D, E, G\}$ và

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

Các khoá của lược đồ quan hệ gồm: $K_1 = \{A, B\}$, $K_2 = \{B, E\}$, $K_3 = \{C, G\}$, $K_4 = \{C, E\}$, $K_5 = \{C, D\}$, $K_6 = \{B, C\}$. Như vậy không tồn tại các thuộc tính không khoá, vì vậy lược đồ quan hệ trên dạng chuẩn 2NF.

4.4.4 Dạng chuẩn 3 - 3NF (Third Normal Form)

Lược đồ quan hệ $s = \langle \Omega, F \rangle$ được gọi là dạng chuẩn 3NF, khi và chỉ khi không tồn tại phụ thuộc hàm $X \rightarrow Y \in F^+$ sao cho $X^+ \neq \Omega$, $Y \not\subset X$ và Y là thuộc tính không khoá. Nói cách khác nếu $X \rightarrow Y \in F^+$, $Y \not\subset X$ thì khi đó hoặc X là khoá của lược đồ quan hệ hoặc Y là một thuộc tính của khoá.

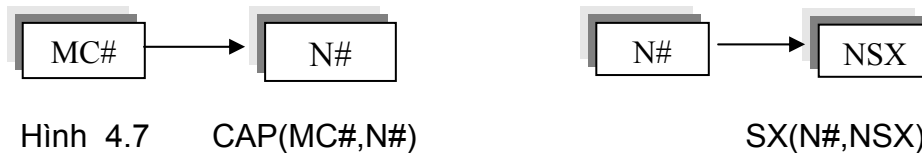
Trong lược đồ 2NF, cấm tất cả các thuộc tính không khoá phụ thuộc vào các tập con thực sự của khoá. Trong dạng chuẩn 3NF, cấm các thuộc tính không khoá phụ thuộc hàm vào tất cả các tập thuộc tính có bao đóng khác Ω .

Các khẳng định sau là tương đương

- a) Không tồn tại phụ thuộc hàm $X \rightarrow Y \in F^+$ sao cho $X^+ \neq \Omega$, $Y \not\subset X$ và Y là thuộc tính không khoá.
- b) Nếu $X \rightarrow Y \in F^+$, $Y \not\subset X$ khi đó hoặc X là khoá của lược đồ quan hệ hoặc Y là một thuộc tính của khoá.
- c) Không tồn tại thuộc tính không khoá phụ thuộc bắc cầu vào khoá.
- d) Tập các thuộc tính không khoá bằng rỗng.

Như vậy một lược đồ quan hệ dạng chuẩn 3NF thì cũng là dạng chuẩn 2NF.

Lược đồ quan hệ TUYEN là lược đồ quan hệ dạng chuẩn 3NF. Vì (MC#,TC#) là các thuộc tính khoá. Lược đồ quan hệ CAP_NSX ở dạng chuẩn 2NF, nhưng không phải là dạng chuẩn 3NF. Vì $N\# \rightarrow NSX$, $N\#$ và NSX là các thuộc tính không khoá. Hiển nhiên có thể kiểm tra được rằng khi chèn thêm dữ liệu, hoặc loại bỏ dữ liệu hoặc sửa đổi giá trị của một số thuộc tính trong một quan hệ bất kỳ của lược đồ quan hệ CAP_NSX sẽ xuất hiện các bất thường khi tìm kiếm, tra cứu thông tin. Có thể tách CAP_NSX thành 2 quan hệ chuẩn 3NF CAP và SX (hình 4.7). Phép tách có kết nối không tồn thất thông tin.



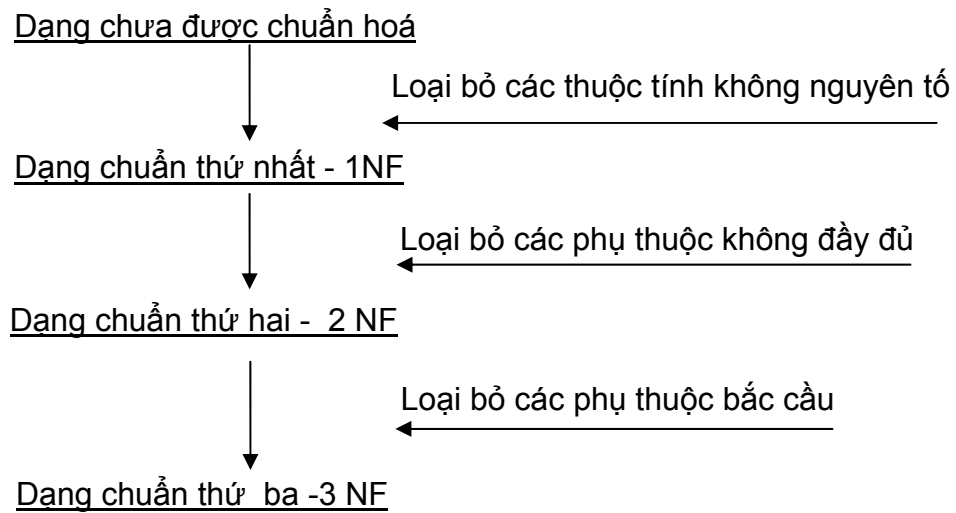
Từ một lược đồ quan hệ 2NF nhưng không phải 3NF được biến đổi thành một họ tương đương các lược đồ quan hệ dạng 3NF. Quá trình chuyển đổi là quá trình khả nghịch, do đó không làm mất thông tin trong quá trình chuyển đổi. Các lược đồ quan hệ kết quả sau biến đổi được gọi là các lược đồ quan hệ chiếu. Nhóm các lược đồ quan hệ chiếu tương đương với lược đồ quan hệ ban đầu theo nghĩa là lược đồ quan hệ ban đầu có thể nhận được bằng cách thực hiện các phép kết nối các lược đồ quan hệ chiếu tương ứng. Như vậy quá trình

biến đổi sẽ không làm mất thông tin. Thông tin của lược đồ quan hệ ban đầu có thể nhận được từ các lược đồ quan hệ chiểu.

4.4.5 Quá trình tách một lược đồ quan hệ 1NF về nhóm tương đương 3NF.

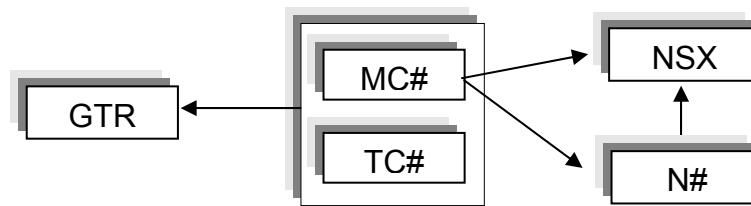
Từ một lược đồ quan hệ dạng chuẩn thứ nhất có thể chuyển đổi về nhóm các lược đồ quan hệ chiểu dạng chuẩn 3NF bằng cách loại bỏ các phụ thuộc không đầy đủ vào khoá và loại bỏ các phụ thuộc bậc cầu vào khoá. Quá trình tách là quá trình loại bỏ đi thường thông tin gây những điều bất tiện khi thao tác lưu trữ dữ liệu. Phép tách các lược đồ quan hệ là phép tách có kết nối không tổn thất thông tin.

Codd đã đưa ra 3 dạng chuẩn của quan hệ và lộ trình chuyển đổi như sau:

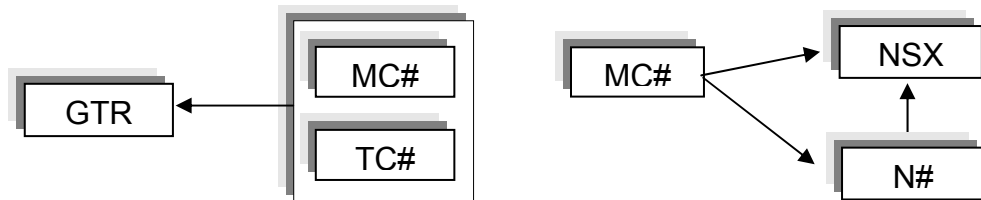


Hình 4.8 Các cấp chuẩn hoá quan hệ

Từ quan hệ QLCAP(TC#, MC#, GTR, N#,NSX)



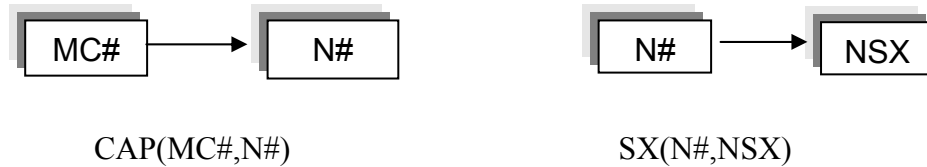
Loại bỏ các phụ thuộc không đầy đủ vào khoá



TUYEN (TC#, MC#, GTR)

CAP_NSX(MC#, N#, NSX)

Loại bỏ các phụ thuộc bậc cầu vào khoá:



Hình 4.8 Quá trình tách QLCAP về các lược đồ quan hệ 3NF

4.5 Dạng chuẩn Boyce Codd

Khái niệm lược đồ quan hệ dạng chuẩn 1NF và dạng chuẩn 2NF là những khái niệm trung gian trên chặng đường đi tới chuẩn 3NF. Tuy nhiên với khái niệm chuẩn 3NF chưa thoả mãn cho các trường hợp lược đồ quan hệ có hơn một khoá, đặc biệt các trường hợp khoá giao nhau. Heath là người đầu tiên, sau đó Boyce Codd đã đưa ra định nghĩa dạng chuẩn Boyce Codd có thể thoả mãn các yêu cầu trên. Khái niệm dạng chuẩn Boyce Codd không dựa vào các khái niệm dạng chuẩn 1NF và 2NF, tức là không dựa vào các khái niệm về phụ thuộc hàm đầy đủ và phụ thuộc bậc cầu.

4.5.1 Định nghĩa dạng chuẩn Boyce Codd

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

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

Có thể định nghĩa cách khác như sau: $s = \langle \Omega, F \rangle$ được gọi là dạng chuẩn Boyce Codd, khi và chỉ khi mọi định thuộc là khoá của lược đồ quan hệ.

Các khẳng định sau là tương đương

- a) Nếu $X \rightarrow Y \in F^+$, $Y \not\subseteq X$ khi đó X là khóa của lược đồ quan hệ.
- b) Nếu $X \rightarrow Y \in F^+$, $Y \not\subseteq X$ khi đó $X^+ = \Omega$.
- c) Mọi định thuộc là khoá của lược đồ quan hệ. (X là định thuộc khi và chỉ khi tồn tại $Y \subseteq \Omega$ sao cho $X \rightarrow Y \in F^+$ là phụ thuộc đầy đủ).

Định nghĩa dạng chuẩn 3NF và BCNF giống nhau trừ mệnh đề “ hoặc Y là một thuộc tính của khóa”. Điều này có nghĩa là nếu lược đồ quan hệ dạng chuẩn BCNF thì cũng là 3NF. Trong định nghĩa 3NF, loại trừ các thuộc tính không khóa phụ thuộc vào các thuộc tính có bao đóng khác Ω , còn trong định nghĩa BCNF thì loại trừ tất cả các thuộc tính phụ thuộc vào các thuộc tính có bao đóng khác Ω .

Các dạng chuẩn lược đồ quan hệ lồng nhau, nghĩa là nếu lược đồ quan hệ dạng chuẩn BCNF thì cũng là dạng chuẩn 3NF, nếu đã là 3NF cũng là dạng chuẩn 2NF và nếu 2NF cũng là dạng chuẩn 1NF. Ký hiệu

$$BCNF \subset 3NF \subset 2NF \subset 1NF.$$

4.5.2 Nhận biết dạng chuẩn 3NF & BCNF

- a) Nhận biết một lược đồ quan hệ là dạng chuẩn 3NF:
 - Xác định tập các thuộc tính không khóa Y.
 - Kiểm tra xem có tồn tại phụ thuộc $X \rightarrow Y \in F^+$, $Y \not\subset X$ và $X^+ \neq \Omega$.
- b) Nhận biết một lược đồ quan hệ là dạng chuẩn BCNF
 - Xác định các tập các thuộc tính X sao cho $X^+ \neq \Omega$.
 - Kiểm tra xem có tồn tại phụ thuộc $X \rightarrow Y \in F^+$, $Y \not\subset X$.

Ví dụ

- a) Cho $\Omega = \{A, B, C, D, E, G, H\}$ và $F = \{C \rightarrow AB, D \rightarrow E, B \rightarrow G\}$.
 - Khóa của lược đồ $K = \{H, C, D\}$.
 - Tập các thuộc tính không khoá: $NK = \{A, B, E, G\}$.
 - Vế phải của các phụ thuộc $C \rightarrow AB, D \rightarrow E$ và $B \rightarrow G$ là các thuộc tính không khoá, và $C^+ \neq \Omega$, $D^+ \neq \Omega$ và $B^+ \neq \Omega$.
 - Lược đồ là dạng chuẩn 3NF
- b) Cho $\Omega = \{A, B, C, D\}$ và $F = \{AB \rightarrow C, C \rightarrow ABD\}$.
 - Tập có bao đóng khác Ω : $X = A, X = B, X = D, X = AD, X = BD$.
 - Không tồn tại $X \rightarrow Y \in F^{++}$, $Y \not\subset X$, suy ra BCNF.
- c) Cho $\Omega = \{A, B, C, D, E, G, H\}$ và $F = \{A \rightarrow BC, D \rightarrow E, H \rightarrow G\}$.
 - Tồn tại $D \rightarrow E \in F^+$, $D^+ \neq \Omega$, suy ra lược đồ không BCNF.

Tồn tại lược đồ quan hệ không là dạng chuẩn 3NF thì cũng không là chuẩn Boyce Codd.

4.6 Thuật toán tách lược đồ quan hệ không tổn thất thông tin

4.6.1 Phép tách về các lược đồ dạng chuẩn 3NF có bảo toàn phụ thuộc

Input: Cho $s = \langle \Omega, F \rangle$ là một lược đồ quan hệ, trong đó
 $\Omega = \{A_1, A_2, \dots, A_n\}$ tập các thuộc tính và
 $F = \{f : L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$ tập các phụ thuộc hàm.
Không mất tính tổng quát, giả sử tập F là phủ cực tiểu.

Output: Một phép tách bảo toàn phụ thuộc sao cho mỗi một lược đồ quan hệ chiếu có dạng chuẩn 3NF tương ứng với tập phụ thuộc chiếu của F trên nó.

Phương pháp:

- Nếu có những thuộc tính của Ω không có mặt trong các vế phải và trái của các phụ thuộc hàm, thì tập các thuộc tính này sẽ tạo ra một lược đồ quan hệ chiếu. Loại bỏ tất cả những thuộc tính này ra khỏi Ω trong quá trình thực hiện thuật toán.
- Nếu tồn tại phụ thuộc $X \rightarrow A \in F$ sao cho vế phải và trái của nó chứa tất cả các thuộc tính của Ω , khi đó kết quả phép tách chính là lược đồ quan hệ s .
- Ngược lại, $\Omega_i = \{XA\}$ là một thành phần của phép tách.

4.6.2 Thuật toán tách bảo toàn phụ thuộc và không tổn thất thông tin về các lược đồ dạng chuẩn 3NF

Input: Cho $s = \langle \Omega, F \rangle$ là một lược đồ quan hệ. Trong đó

$\Omega = \{A_1, A_2, \dots, A_n\}$ tập các thuộc tính và
F phủ cực tiểu các phụ thuộc hàm.
K là khoá của lược đồ quan hệ.

Output: Phép tách $\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ bảo toàn phụ thuộc và không tổn thất thông tin sao cho mỗi một lược đồ quan hệ chiếu có dạng chuẩn 3NF tương ứng với tập phụ thuộc chiếu của F trên nó.

Phương pháp:

1. Xác định các thuộc tính của Ω không có mặt trong các vế phải và trái của các phụ thuộc hàm. Ký hiệu tập các thuộc tính này là A.
 - Tạo ra lược đồ quan hệ chiếu trên tập các thuộc tính A.
 - Loại bỏ tất cả những thuộc tính A ra khỏi Ω trong quá trình thực hiện thuật toán: $\Omega = \Omega - \{A\}$.
2. Nếu tồn tại phụ thuộc $X \rightarrow A \in F$ sao cho vế phải và trái của nó chứa tất cả các thuộc tính của Ω , khi đó kết quả phép tách $\varphi[X, A]$.
3. Với mọi $X \rightarrow A \in F$ trong đó thuộc tính A là thuộc tính đơn (F là phủ cực tiểu). $\Omega_i = \{XA\}$, với $X \rightarrow A, i=1 \div p$.
4. Nếu có một số phụ thuộc cùng vế trái: $X \rightarrow A_1 \in F, X \rightarrow A_2 \in F, \dots, X \rightarrow A_k \in F$ thì có thể hợp lại thành dạng: $\Omega_j = \{XA_1 A_2 \dots A_n\}$, với $X \rightarrow A_i, i=1 \div k$.
5. Nếu các thuộc tính của khoá K không xuất hiện trong các tập Ω_j được tạo bởi các bước trên khi đó một thành của phép tách sẽ được định nghĩa bởi khoá K.
6. Phép tách $\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ bảo toàn phụ thuộc và không tổn thất thông tin. Mỗi một lược đồ quan hệ chiếu có dạng chuẩn 3NF tương ứng với tập phụ thuộc chiếu của F trên nó.

Ví dụ Cho $\Omega = \{X, Y, Z, W, Q\}$ và phủ cực tiểu
 $F = \{X \rightarrow Y, XZ \rightarrow W, YW \rightarrow Q\}$

Xác định một phép tách φ -3NF bảo toàn phụ thuộc và không tổn thất thông tin.

1. XZ là khoá: $(XZ)^+ = \Omega = \{X, Y, Z, W, Q\}$
2. Thực hiện phép tách:
 - Không tồn tại các thuộc tính mà các thuộc tính đó không xuất hiện trong các vế của phụ thuộc hàm.
 - Không tồn tại phụ thuộc hàm chứa các thuộc tính còn lại của Ω .
 - $X \rightarrow Y$: $\Omega_1 = \{XY\}$.
 - $XZ \rightarrow W$: $\Omega_2 = \{X, Z, W\}$.
 - $YW \rightarrow Q$: $\Omega_3 = \{Y, W, Q\}$.
 - Khoá $XZ \subseteq \Omega_2 = \{X, Z, W\}$
3. Vậy phép tách $\varphi [XY, XZW, YWQ]$.
4. Kiểm tra tổn thất thông tin:

	X	Y	Z	W	Q	⇒	X	Y	Z	W	Q	
Ω_1	a	a	b	b	b		Ω_1	a	a	b	b	b
Ω_2	a	b	a	a	b		Ω_2	a	a	a	a	a
Ω_3	b	a	b	a	a		Ω_3	b	a	b	a	a

4.6.3 Thuật toán tách về dạng chuẩn BCNF không tổn thất

Input: $s = \langle \Omega, F \rangle$ lược đồ quan hệ.
 $\Omega = \{A_1, A_2, \dots, A_n\}$ tập các thuộc tính.
 $F = \{L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$ tập các phụ thuộc.

Output: Phép tách $\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ không tổn thất thông tin. Mỗi một lược đồ quan hệ chiếu có dạng chuẩn BCNF tương ứng với tập phụ thuộc chiếu của F trên nó.

Phương pháp: Phương pháp chủ yếu của thuật toán là tách lược đồ $s = \langle \Omega, F \rangle$ thành 2 lược đồ. Chọn bất kỳ $X \rightarrow A \in F^+$ sao cho X không là khoá và $A \notin X$. Khi đó lược đồ có tập các thuộc tính XA sẽ có dạng chuẩn BCNF và phụ thuộc hàm $X \rightarrow A$ sẽ thoả trên nó. Lược đồ thứ 2 có tập các thuộc tính $\Omega \setminus A$. Hiển nhiên, khi kết nối lược đồ có tập thuộc tính $\Omega \setminus A$ với lược đồ có tập thuộc tính XA không tổn thất thông tin. Tiếp tục tách $\Omega \setminus A$ cho đến khi thoả điều kiện (b) của định lý 4.11, cũng là lược đồ có dạng chuẩn BCNF. Theo định lý 4.10, các lược đồ quan hệ chiếu cũng có dạng BCNF và có kết nối không tổn thất thông tin.

$$\begin{aligned} \text{Đặt } \Omega_1 &:= XA, \quad \Omega_2 := \Omega \setminus A \\ \Omega_1 \cap \Omega_2 &= X \\ \Omega_1 \setminus \Omega_2 &= A \end{aligned}$$

$$\text{Suy ra } \Omega_1 \cap \Omega_2 \rightarrow \Omega_1 \setminus \Omega_2.$$

Phép tách $\varphi [\Omega_1, \Omega_2]$ không tổn thất khi và chỉ khi $\Omega_1 \cap \Omega_2 \rightarrow \Omega_1 \setminus \Omega_2$. Tiếp tục cho đến khi tất cả lược đồ ở dạng chuẩn Boyce Codd.

- Sử dụng phương pháp lặp liên tiếp. Mỗi bước đảm bảo không làm tổn thất thông tin.
- Bước đầu áp dụng đối với lược đồ quan hệ $S = \langle \Omega, F \rangle$.
- Giả sử bước thứ i : Lược đồ quan hệ $s_i = \langle \Omega_i, F \mid \Omega_i \rangle$ chưa phải là lược đồ dạng chuẩn Boyce Codd.
- Chọn $X \rightarrow A \in F_i = F \mid \Omega_i$ sao cho X không là khoá và $A \notin X$.
- Đặt $\Omega_{i1} = XA, \quad \Omega_{i2} = \Omega \setminus A$
- Tiếp tục phép lặp.

Ví dụ $\Omega = \{C, T, H, R, S, G\}$, trong đó:

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

Biết rằng:

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

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

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

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

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

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

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

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

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

$$\begin{aligned} S_{221} &= \langle \Omega_{221}, F_{221} \rangle & S_{222} &= \langle \Omega_{222}, F_{222} \rangle \\ \Omega_{221} &= \{ C, \underline{H}, \underline{R} \} & \Omega_{222} &= \{ \underline{H}, \underline{S}, R \} \\ F_{221} &= \{ HR \rightarrow C \} & F_{222} &= \{ HS \rightarrow R \}, \end{aligned}$$

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

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

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

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

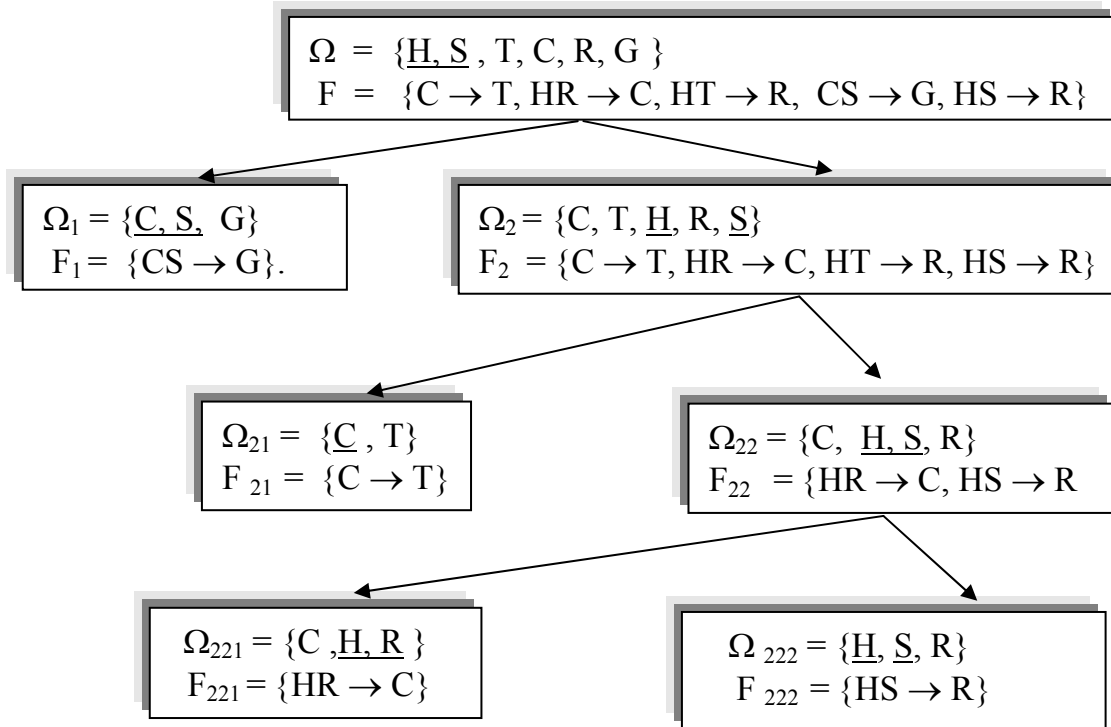
$$\Omega_1 = \{ \underline{C}, \underline{S}, G \}, \quad F_1 = \{ CS \rightarrow G \}$$

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

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

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

Sơ đồ tóm tắt quá trình tách như sau:



Hình 4.9 Quá trình tách- kết nối không mất thông tin

Câu hỏi trắc nghiệm

1. Mục tiêu của lý thuyết CSDL là
 - A. Tính độc lập của dữ liệu
 - B. Tính phụ thuộc dữ liệu.
 - C. Tính đơn giản trong biểu diễn.
 - D. Tính đối xứng trong kết quả..
2. Lược đồ quan hệ $s = \langle \Omega, F \rangle$ được gọi là dạng chuẩn 1 – 1NF, khi và chỉ khi
 - A. Bao đóng các thuộc tính là lớn nhất
 - B. Các thuộc tính khoá phụ thuộc bắc cầu vào khoá.
 - C. Các thuộc tính chỉ chứa các giá trị nguyên tố
 - D. Các phụ thuộc không có vế trái dư thừa
3. Lược đồ quan hệ $s = \langle \Omega, F \rangle$ được gọi là dạng chuẩn 2NF, khi và chỉ khi
 - A. Dạng chuẩn 1NF và tồn tại $X \rightarrow Y \in F +$ sao cho X là tập con của khóa
 - B. Không tồn tại các phụ thuộc dư thừa.
 - C. Các thuộc tính khoá phụ thuộc bắc cầu vào khoá.
 - D. Dạng chuẩn 1NF và các thuộc tính không khoá phụ thuộc đầy đủ vào khoá.

4. Lược đồ quan hệ $s = \langle \Omega, F \rangle$ được gọi là dạng chuẩn 3NF, khi và chỉ khi
 - A. Không tồn tại $X \rightarrow Y \in F^+$ sao cho $X^+ \neq \Omega$, $Y \not\subseteq X$ là thuộc tính không khoá.
 - B. Nếu $X \rightarrow Y \in F^+$, $Y \not\subseteq X$ thì khi đó X
 - C. Y là một thuộc tính của khoá.
 - D. Lược đồ 2NF và các thuộc tính không khoá phụ thuộc khoá.
 5. X là định thuộc khi và chỉ khi
 - A. Tồn tại $Y \subseteq \Omega$ sao cho $X \rightarrow Y \in F^+$ là phụ thuộc đầy đủ
 - B. Tồn tại $Y \subseteq \Omega$ sao cho $X \rightarrow Y \in F^+$ là phụ thuộc không đầy đủ.
 - C. Tồn tại $Y \subseteq \Omega$ sao cho $X \rightarrow Y \in F^+$ là phụ thuộc không dư thừa
 - D. Tồn tại $Y \subseteq \Omega$ sao cho $X \rightarrow Y \in F^+$ là phụ thuộc dư thừa
 6. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Trong quá trình chuẩn hoá dữ liệu và tìm kiếm thông tin phải thực hiện các về tập các lược đồ quan hệ chuẩn hoá, sao cho quá trình tách không làm tổn thất thông tin.
 - A. Phép tách lược đồ quan hệ.
 - B. Phép lưu trữ.
 - C. Phép tìm kiếm thông tin.
 - D. Phép chiếu, chọn và kết nối
 7. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Mục tiêu của phép tách lược đồ quan hệ là nhằm loại bỏ khi thực hiện các phép lưu trữ như chèn thêm, loại bỏ hay sửa đổi thông tin trong trong các quan hệ lưu trữ.
 - A. Dị thường thông tin
 - B. Không toàn vẹn dữ liệu
 - C. Không bảo mật dữ liệu
 - D. Không nhất quán dữ liệu
 8. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Từ một lược đồ quan hệ dạng 1NF có thể chuyển đổi về nhóm các lược đồ quan hệ chiếu dạng chuẩn 3NF bằng cách..... vào khoá và loại bỏ các phụ thuộc bắc cầu vào khoá.
 - A. Loại bỏ các phụ thuộc không đầy đủ.
 - B. Phụ thuộc bắc cầu.
 - C. Phụ thuộc dư thừa.
 - D. Các thuộc tính dư thừa
 9. $\Omega = \{A, B, C, D, E, G, H\}$ và $F = \{C \rightarrow AB, D \rightarrow E, B \rightarrow G\}$.
 - A. Lược đồ chưa được chuẩn hoá
 - B. Lược đồ là dạng chuẩn 3NF
 - C. Lược đồ là dạng chuẩn 2NF
 - D. Lược đồ là dạng chuẩn BCNF
-
-

10. Hãy so khớp thông tin 2 phần dưới đây với nhau:

Phần 1

- 1 Khi thực hiện các phép lưu trữ trên các quan hệ chưa được chuẩn hoá
- 2 Sự dư thừa, mất dữ liệu, mâu thuẫn hay không nhất quán
- 3 Dự thường thông tin là nguyên nhân gây cản trở cho.....
- 4 Mục tiêu của chuẩn hoá dữ liệu là triết tiêu.....

- a. 1- D; 2- C ; 3 – B ; 4 – A .
- b. 1- C; 2- B; 3 – A; 4 – D
- c. 1- A; 2- B; 3 – C; 4 – D.
- d. 1- D; 2- B; 3 – A; 4 – C.

Phần 2

- A. việc tìm kiếm, hỏi đáp thông tin.
- B. dữ liệu có thể xảy ra khi cập nhật, bổ sung hay sửa đổi dữ liệu
- C. khả năng xuất hiện các dự thường thông tin khi thực hiện các phép lưu trữ
- D. thường xuất hiện các dự thường thông tin.

11. Hãy so khớp thông tin 2 phần dưới đây với nhau:

Phần 1

- 1 Các thuộc tính không khoá
- 2 Các thuộc tính khoá
- 3 Nếu $X \rightarrow Y \in F^+$, $Y \not\subset X$ khi đó X.....
- 4 Mọi định thuộc

- a. 1- D; 2- C ; 3 – B ; 4 – A .
- b. 1- C; 2- B; 3 – A; 4 – D
- c. 1- A; 2- B; 3 – C; 4 – D.
- d. 1- D; 2- A; 3 – B; 4 – C.

Phần 2

- A. phụ thuộc hoàn toàn vào tất cả khoá khác
- B. là khoá của lược đồ quan hệ.
- C. là khoá của lược đồ quan hệ.
- D. phụ thuộc hoàn toàn vào khoá.

Câu hỏi và bài tập:

Câu hỏi

1. Sự cần thiết tách các lược đồ quan hệ.
2. Trình bày khái niệm và định nghĩa phép tách.
3. Hiểu như thế nào về các ký hiệu $\pi_{\Omega_i}(R)$ và $F_i = F |_{\Omega_i}$.
4. Định nghĩa phép tách - kết không tổn thất thông tin. Ví dụ minh hoạ.
5. Cho một số thí dụ về phép tách- kết nối tổn thất và không tổn thất thông tin.
6. Trình bày thuật toán kiểm tra tách - kết nối không tổn thất thông tin.
7. Cho $\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ là một phép tách của lược đồ quan hệ $s = \langle \Omega, F \rangle$. Cho một quan hệ bất kỳ $R \in s$ trên Ω . Chứng minh rằng:
 - a) $R \subseteq m_{\varphi}(R)$.
 - b) Nếu $P = m_{\varphi}(R)$ thì $R_i = \pi_{\Omega_i}(P)$, với $i = 1 \div p$.
 - c) $m_{\varphi}(m_{\varphi}(R)) = m_{\varphi}(R)$.

8. Cho $\varphi [\Omega_1, \Omega_2]$ là một phép tách của lược đồ quan hệ $s = \langle \Omega, F \rangle$ trong đó Ω là tập các thuộc tính và F là tập các phụ thuộc hàm. Khi đó phép tách - kết nối φ không tổn thất thông tin khi và chỉ khi:
$$\Omega_1 \cap \Omega_2 \rightarrow \Omega_1 \setminus \Omega_2 \text{ hoặc } \Omega_1 \cap \Omega_2 \rightarrow \Omega_2 \setminus \Omega_1 .$$
9. Trình bày khái niệm và định nghĩa phép chiếu bảo toàn các phụ thuộc.
10. Thuật toán kiểm tra bảo toàn tập phụ thuộc hàm.
11. Cho $\forall Z \subseteq \Omega$, tập F , đặt $Z^+ = Z \cup ((Z \cap \Omega)^+ \cap \Omega)$, khi đó sẽ tồn tại các thuộc tính A sao cho $(Z \cap \Omega) \rightarrow A \in \pi_{\Omega}(F)$.
12. Phát biểu định nghĩa dạng chuẩn thứ nhất – 1NF (First Normal Form)
13. Phát biểu định nghĩa dạng chuẩn thứ hai – 2NF (Second Normal Form)
14. Phát biểu định nghĩa dạng chuẩn thứ 3 - 3NF (Third Normal Form)
15. Các khẳng định sau là tương đương
 - a) Không tồn tại phụ thuộc hàm $X \rightarrow Y \in F^+$ sao cho $X^+ \neq \Omega$, $Y \not\subset X$ và Y là thuộc tính không khóa.
 - b) Nếu $X \rightarrow Y \in F^+$, $Y \not\subset X$ khi đó hoặc X là khóa của lược đồ quan hệ hoặc Y là một thuộc tính của khóa.
 - c) Không tồn tại phụ thuộc bắc cầu vào khóa.
16. Dạng chuẩn 3NF thì cũng là lược đồ quan hệ chuẩn 2NF.
17. Một lược đồ quan hệ 1NF và không tồn tại thuộc tính không khóa phụ thuộc bắc cầu vào khóa, khi đó lược đồ dạng chuẩn 2NF.
18. Trình bày quá trình tách một lược đồ quan hệ 1NF về nhóm 3NF.
19. Trình bày khái niệm và định nghĩa dạng chuẩn Boyce Codd.
20. Các khẳng định sau là tương đương
 - a) Nếu $X \rightarrow Y \in F^+$, $Y \not\subset X$ khi đó X là khóa của lược đồ quan hệ.
 - b) Nếu $X \rightarrow Y \in F^+$, $Y \not\subset X$ khi đó $X^+ = \Omega$.
 - c) Mọi định thuộc là khóa của lược đồ quan hệ. (X là định thuộc khi và chỉ khi tồn tại $Y \subseteq \Omega$ sao cho $X \rightarrow Y \in F^+$ là phụ thuộc đầy đủ).
21. Chứng minh rằng $BCNF \subset 3NF \subset 2NF \subset 1NF$.
22. Nhận biết dạng chuẩn 3NF & BCNF.
23. Cho thí dụ 3NF nhưng không là BCNF.
24. Cho thí dụ không 3NF và không là BCNF.
25. Cho một số thí dụ về các dạng chuẩn lược đồ quan hệ
26. Trình bày thuật toán tách – kết nối lược đồ quan hệ không tổn thất thông tin
27. Phép tách lược đồ quan hệ về nhóm tương đương 3NF.
28. Phép tách lược đồ quan hệ về dạng chuẩn Boyce Codd.
29. Chứng minh rằng:
 - a) Nếu Ω chỉ có 2 thuộc tính, khi đó lược đồ ở dạng chuẩn BCNF.

- b) Nếu $s = \langle \Omega, F \rangle$ không dạng chuẩn BCNF, sẽ tồn tại $A, B \subseteq \Omega$ sao cho $(\Omega \setminus AB) \rightarrow A$ hoặc $(\Omega \setminus AB) \rightarrow B$.
30. Trình bày thuật toán tách – kết nối không tổn thất thông tin từ một lược đồ quan hệ $s = \langle \Omega, F \rangle$ về dạng chuẩn Boyce Codd.

Bài tập

- Cho tập các thuộc tính quản lý tổng đài- cước thuê bao: TD# (Mã tổng đài), TD (Tên tổng đài), DL(Dung lượng), MC#(Màu cáp), SDT(Số điện thoại thuê bao), NSX(Nước sản xuất), NG(Ngày lắp đặt tổng đài), GTR(Giá trị tổng đài), VT(Nơi đặt tổng đài), TB(Tên thuê bao), TB#(Mã thuê bao), DC(Địa chỉ thuê bao), SDTD(Số điện thoại đến/đi), NGG(Ngày thực hiện đàm thoại), GB(Giờ bắt đầu đàm thoại),KT(Giờ kết thúc đàm thoại), DD(Đi/đến).
 - Hãy xác định một phép tách không tổn thất về 3NF có bảo toàn phụ thuộc.
 - Hãy xác định một phép tách – kết nối không tổn thất về BCNF.
- Cho tập các thuộc tính quản lý phát hành báo chí QLBC gồm các thuộc tính MK#(Mã khách), K(Tên khách), DC(Địa chỉ), MB#(Mã báo),TB(Tên báo), GIA(Giá báo), SL(số lượng), KY(kỳ phát hành).
 - Hãy xác định một phép tách – kết nối không tổn thất về 3NF có bảo toàn phụ thuộc.
 - Hãy xác định một phép tách – kết nối không tổn thất về BCNF.
- Giả sử các thuộc tính BC (Bưu cục), BC#(Mã bưu cục), MK(Khách gửi), DC(Địa chỉ khách), NH (Người nhận), DCN(Địa chỉ người nhận), ST(Số tiền), BCN(Bưu cục đến), BCD#(Mã bưu cục đến), NG(ngày gửi) HD(Hoá đơn).
 - Hãy xác định một phép tách – kết nối không tổn thất về 3NF có bảo toàn phụ thuộc.
 - Hãy xác định một phép tách – kết nối không tổn thất về BCNF.
 - Có phép tách – kết nối không tổn thất và đồng thời bảo toàn phụ thuộc về BCNF.
- Cho $\Omega = \{ X, Y, Z, W, Q \}$ và $F = \{XY \rightarrow QW, Z \rightarrow Q, W \rightarrow Z, Q \rightarrow X\}$. $\Omega_1 = \{X, Y, Z\}$. Hãy xác định tập các phụ thuộc chiếu F trên Ω_1 : $F_1 = \pi_{\Omega_1}(F)$.
- Cho $\Omega = \{ X, Y, Z, W \}$ và $F = \{X \rightarrow Y, Z \rightarrow W\}$.
Phép tách $\phi[\Omega_1\{X, Y\}, \Omega_2\{Z, W\}]$. Hãy xác định phép tách bảo toàn phụ thuộc ?.
- Cho $\Omega = \{ A, B, C, D \}$ và $F = \{AB \rightarrow C, A \rightarrow D, BD \rightarrow C\}$.
 - Tìm phủ cực tiểu của F.
 - Hãy xác định một phép tách thành 2 lược đồ dạng 3NF bảo toàn phụ thuộc.
 - Xác định các phụ thuộc chiếu trên các tập thuộc tính trên.
 - Kết quả (b) có phải là tách kết nối tổn thất không. Nếu có sửa lại như thế nào để không tổn thất nhưng vẫn bảo toàn phụ thuộc.
- Cho $\Omega = \{ A, B, C \}$ và $F = \{AB \rightarrow C, A \rightarrow B\}$.

- a) Tìm phủ cực tiểu của F.
- b) Giả sử phủ cực tiểu của F là $G = \{AB \rightarrow C, B \rightarrow C\}$. Chứng tỏ rằng điều giả sử là sai bằng cách xây dựng một lược đồ quan hệ thoả F nhưng vi phạm G.
8. Cho $\Omega = \{A, B, C, D\}$ và $F = \{A \rightarrow B, B \rightarrow C, A \rightarrow D, D \rightarrow C\}$, $\varphi[AB, AC, BD]$.
 - a) Tìm các phụ thuộc chiếu trên mỗi tập tương ứng.
 - b) φ là phép tách – kết nối không tổn thất thông tin.
 - c) φ bảo toàn các phụ thuộc hay không.
9. Cho $\Omega = \{A, B, C, D\}$, $F = \{A \rightarrow C, D \rightarrow C, BD \rightarrow A\}$ và $\varphi[AB, ACD, BCD]$. Chứng minh rằng phép tách – kết nối φ không tổn thất thông tin.
10. Cho $\Omega = \{A, B, C, D\}$ và $F = \{A \rightarrow B, B \rightarrow C, D \rightarrow B\}$
 - a) Nếu $\varphi[ACD, BD]$, hãy xác định các phụ thuộc chiếu trên mỗi tập tương ứng.
 - b) Hãy tiếp tục tách về các lược đồ dạng chuẩn BCNF.
11. Cho F là tập các phụ thuộc hàm có vẻ phải chỉ chứa một thuộc tính.
 - a) Giả sử $X \rightarrow A \in F^+$ nhưng không thoả các điều kiện định nghĩa BCNF. Chứng minh rằng tồn tại một phụ thuộc $Y \rightarrow B \in F$ sao cho không thoả BCNF là khoá (không suy dẫn).
 - b) Như (a) cho 3NF.
12. Cho $\Omega = \{A, B, C, D, E\}$ và $F = \{AB \rightarrow CE, E \rightarrow AB, C \rightarrow D\}$. Lược đồ quan hệ thuộc dạng chuẩn cao nhất.
13. Cho $\Omega = \{A, B, C\}$ và $F = \{A \rightarrow B, B \rightarrow C\}$. Chứng minh rằng lược đồ quan hệ không là 3NF. Nếu thêm $C \rightarrow B$ thì sẽ là 3NF ?
14. Chứng minh rằng nếu một lược đồ 3NF thì cũng là 2NF.
15. Chứng minh rằng lược đồ quan hệ $s = \langle \Omega, F \rangle$ ở dạng 3NF khi và chỉ khi :
 - a) $\forall X \subset \Omega$ thì $X^+ = \Omega$.
 - b) Nếu A là thuộc tính không khoá thì $(X-A)^+ = X-a$
16. Cho $\Omega = \{A, B, C, D, E, F\}$ và $F = \{A \rightarrow B, CD \rightarrow A, BC \rightarrow D, AE \rightarrow F, CE \rightarrow D\}$. Chứng minh rằng phép tách $\varphi[AED, AB, AEF, ACE]$ về BCNF tổn thất thông tin.
17. Xây dựng các thí dụ tách một lược đồ quan hệ chưa chuẩn hóa về nhóm các quan hệ dạng chuẩn 3NF không tổn thất thông tin. Cho ví dụ phép tách về 3NF nhưng khi kết nối tự nhiên tổn thất thông tin ?
18. Xây dựng các thí dụ tách một lược đồ quan hệ chưa chuẩn hóa về nhóm các quan hệ dạng chuẩn BCNF không tổn thất thông tin. Cho ví dụ phép tách về BCNF nhưng khi kết nối tự nhiên tổn thất thông tin ?
19. Xây dựng các thí dụ tách một lược đồ quan hệ chưa chuẩn hóa về nhóm các quan hệ dạng chuẩn 4NF không tổn thất thông tin. Cho ví dụ phép tách về 4NF nhưng khi kết nối tự nhiên tổn thất thông tin ?

4 KỸ THUẬT THIẾT KẾ CÁC HỆ CƠ SỞ DỮ LIỆU QUAN HỆ

Chương này sẽ giới thiệu nguyên lý thiết kế và cài đặt các hệ cơ sở dữ liệu quan hệ. Kỹ thuật chuyển đổi một quan hệ chưa chuẩn hóa về một nhóm các quan hệ ở dạng chuẩn 3NF không bị tổn thất thông tin, trong một số trường hợp các phép tách vẫn bảo toàn các phụ thuộc. Nội dung của chương bao gồm:

- Tách một lược đồ quan hệ
- Các dạng chuẩn hóa tiếp dữ liệu.

4.1 Mở đầu

Mục tiêu của lý thuyết CSDL là tính độc lập của dữ liệu. Cấu trúc lưu trữ các hệ cơ sở dữ liệu phản ánh tính hiện thực, khách quan và tính toàn vẹn dữ liệu. Vì vậy trong quá trình chuẩn hoá dữ liệu và tìm kiếm thông tin, cần thiết phải thực hiện các phép tách lược đồ quan hệ chưa chuẩn hoá về tập các lược đồ quan hệ chiếu đã được chuẩn hoá, sao cho quá trình tách không làm tổn thất thông tin (lossless- mất mát thông tin), theo nghĩa các quan hệ gốc được khôi phục chính xác từ phép kết nối tự nhiên của các quan hệ chiếu.

Tách - kết nối các lược đồ quan hệ có làm tổn thất thông tin hay không, có bảo toàn các phụ thuộc hay không đã được nhiều người quan tâm nghiên cứu, giải quyết. A.V. Ho, C. Beeri & J.D. Ullman giới thiệu thuật toán xác định phép kết nối các lược đồ quan hệ không có tổn thất thông tin với giả thiết các phụ thuộc dữ liệu là các phụ thuộc hàm. Các ông cũng đã mở rộng vấn đề này cho các trường hợp phụ thuộc dữ liệu là phụ thuộc đa trị.

4.2 Phép tách – kết nối không tổn thất thông tin

4.2.1 Phép tách

Cho $s = \langle \Omega, F \rangle$ là một lược đồ quan hệ, trong đó $\Omega = \{A_1, A_2, \dots, A_n\}$ là tập các thuộc tính và F là tập các phụ thuộc hàm. Gọi $\varphi[\Omega_1, \Omega_2, \dots, \Omega_p]$ là một phép tách (hay còn gọi là một phân hoạch) của $s = \langle \Omega, F \rangle$, nếu:

- $\Omega_i \subseteq \Omega, i=1 \div p$
- $\Omega = \Omega_1 \cup \dots \cup \Omega_p$
- $F_i := F \mid_{\Omega_i} := \pi_{\Omega_i}(F) := \{X \rightarrow Y \in F, XY \subseteq \Omega_i\}, i=1 \div p.$
- $s_i := \langle \Omega_i, F_i \rangle = \pi_{\Omega_i}(S), i=1 \div p.$

Như vậy, nếu $\varphi[\Omega_1, \Omega_2, \dots, \Omega_p]$ là một phép tách của $s = \langle \Omega, F \rangle$, khi đó tập các phụ thuộc $F_i := F \mid_{\Omega_i} = \pi_{\Omega_i}(F)$ được gọi là tập các phụ thuộc chiếu F trên các tập thuộc tính tương ứng Ω_i . Và các lược đồ $s_i = \langle \Omega_i, F_i \rangle = \pi_{\Omega_i}(S)$ gọi là các lược đồ chiếu trên các tập thuộc tính Ω_i với $i=1 \div p$. Nếu R là một quan hệ trên tập các thuộc tính Ω , khi đó các quan hệ chiếu sẽ là $R_{\Omega_i} := \pi_{\Omega_i}(R), i=1 \div p$, nghĩa là các quan hệ chiếu $\pi_{\Omega_i}(R)$ chỉ bao gồm các thuộc tính $\Omega_i, i=1 \div p$.

4.2.2 Tách - kết nối tự nhiên

Phép tách $\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ được gọi là phép tách - kết nối tự nhiên của của lược đồ quan hệ $s = \langle \Omega, F \rangle$, nếu:

- a) $\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ là một phép tách của $s = \langle \Omega, F \rangle$.
- b) Kết quả của phép kết nối tự nhiên của các lược đồ chiếu $\pi_{\Omega_i}(S)$, $i = 1 \div p$, là một lược đồ $m_\varphi(s)$ trên các thuộc tính $\Omega = \Omega_1 \cup \Omega_2 \cup \dots \cup \Omega_p$.

$$m_\varphi(s) := \pi_{\Omega_1}(s) \bowtie \pi_{\Omega_2}(s) \bowtie \dots \bowtie \pi_{\Omega_p}(s) = s_1 \bowtie s_2 \bowtie \dots \bowtie s_p.$$

Nghĩa là với mọi quan hệ $R \in s = \langle \Omega, F \rangle$, khi đó $m_\varphi(R)$ là kết quả phép kết nối tự nhiên của các quan hệ chiếu tương ứng $R_i := R_{\Omega_i} := \pi_{\Omega_i}(R)$, $i = 1 \div p$, được biểu diễn như sau:

$$R \subseteq m_\varphi(R) := \pi_{\Omega_1}(R) \bowtie \pi_{\Omega_2}(R) \bowtie \dots \bowtie \pi_{\Omega_p}(R).$$

Từ định nghĩa trên có thể suy ra, nếu một thể hiện $I \in m_\varphi(s)$ khi đó:

$$\bigtimes_{i=1}^p \pi_{\Omega_i}(I) = \{a_1, a_2, \dots, a_p\} \mid \text{Nếu } A_j \in \Omega_i, \text{ tại vị trí } j \text{ ứng } R_i \text{ nhận giá trị } a_j, \text{ các vị trí khác nhận giá trị khác } a_j, i = 1 \div p \}.$$

4.2.3 Phép tách không tổn thất thông tin

Phép tách $\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ được gọi phép tách không tổn thất thông tin của lược đồ quan hệ $s = \langle \Omega, F \rangle$, nếu $\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ là phép tách - kết nối tự nhiên và:

$$S = m_\varphi(s) := \pi_{\Omega_1}(s) \bowtie \pi_{\Omega_2}(s) \bowtie \dots \bowtie \pi_{\Omega_p}(s) = s_1 \bowtie s_2 \bowtie \dots \bowtie s_p.$$

Nghĩa là với mọi quan hệ $R \in s$, khi đó R được khôi phục chính xác từ phép kết nối tự nhiên của các quan hệ chiếu $R_i = \pi_{\Omega_i}(R)$, $i = 1 \div p$.

$R = \pi_{\Omega_1}(R) \bowtie \pi_{\Omega_2}(R) \bowtie \dots \bowtie \pi_{\Omega_p}(R) := R_1 \bowtie R_2 \bowtie \dots \bowtie R_p$. Thông tin của một quan hệ R bất kỳ có thể nhận được từ các quan hệ chiếu R_i ứng với phép tách φ .

Ví dụ : Tách quan hệ không tổn thất thông tin

Lược đồ quan hệ quản lý phát hành báo chí QLBC gồm các thuộc tính:

$$\Omega = \{MK\#, TK, DC, MB\#, TB, GIA, SL\} \text{ và}$$

$$F = \{MK\# \rightarrow TK, MK\# \rightarrow DC, MB\# \rightarrow TB, MB\# \rightarrow GIA, (MK\#, MB\#) \rightarrow SL\}.$$

- MK#: Mã khách hàng
- TK : Tên khách hàng
- DC : Địa chỉ khách hàng
- MB#: Mã báo, tạp chí
- TB : Tên báo, tạp chí
- GIA: Đơn giá báo, tạp chí
- SL : Số lượng báo, tạp chí khách đặt mua

Trong lược đồ quan hệ QLBC, các thông tin về tên khách (TK), địa chỉ (DC), tên báo (TB) .. lặp lại rất nhiều lần trong các quan hệ thể hiện, đó là nguyên nhân dẫn đến sự xuất hiện các bất thường, nhập nhằng thông tin. Phép tách φ được mô tả dưới đây, sẽ tách lược đồ QLBC thành 3 lược đồ chiếu. Lược đồ $s_3 = \langle \Omega_3, F_3 \rangle$ chỉ cần lưu trữ thông tin về số lượng các loại báo của mỗi một khách hàng đặt mua. Lược đồ quan hệ $s_1 = \langle \Omega_1, F_1 \rangle$ lưu

trữ thông tin về khách đặt mua báo, và tương tự trong lược đồ quan hệ $s_2 = \langle \Omega_2, F_2 \rangle$ lưu trữ thông tin về các loại báo. Có thể kiểm tra phép tách φ không tổn thất thông tin và bảo toàn được các phụ thuộc hàm.

Phép tách $\varphi [\Omega_1, \Omega_2, \Omega_3]$:

- $\Omega_1 = \{M\#, TK, DC\}$, $F_1 = \{MK\# \rightarrow TK, MK\# \rightarrow DC\}$.
- $\Omega_2 = \{MB\#, TB, GIA\}$, $F_2 = \{MB\# \rightarrow TB, MB\# \rightarrow GIA\}$.
- $\Omega_3 = \{M\#, MB\#, SL\}$, $F_3 = \{(MK\#, MB\#) \rightarrow SL\}$.

Như vậy mục tiêu của phép tách lược đồ quan hệ là nhằm loại bỏ các dị thường thông tin khi thực hiện các phép lưu trữ như chèn thêm, loại bỏ hay sửa đổi thông tin trong các quan hệ lưu trữ. Tuy nhiên khi thực hiện phép tách, thông tin của lược đồ quan hệ có bị tổn thất hay không. Nói cách khác nếu kết nối tự nhiên các thành phần lược đồ quan hệ chiếu, liệu thông tin của lược đồ quan hệ gốc có tồn thất thông tin hay không, các phụ thuộc hàm có được bảo toàn hay không?

Ví dụ: Thí dụ sau mô tả phép tách tổn thất thông tin và không tổn thất thông tin:

Cho $\Omega = \{X, Y, Z\}$ tập các thuộc tính và

Giả sử quan hệ gốc: R và các quan hệ tách: R1 và R2

R		
X	Y	Z
x1	y1	z1
x2	y2	z2
x3	y2	z3
x4	y3	z4

⇒

R1	
X	Y
x1	y1
x2	y2
x3	y2
x4	y3

R2	
Y	Z
y1	Z1
y2	Z2
y2	Z3
v3	Z4

a) Phép tách – kết nối tổn thất thông tin:

R		
X	Y	Z
x1	y1	z1
x2	y2	z2
x3	y2	z3
x4	y3	z4

⇒

R1	
X	Y
x1	y1
x2	y2
x3	y2
x4	y3

R2	
Y	Z
y1	Z1
y2	Z2
y2	Z3
y3	Z4

⇒

$R \subseteq R1 \times R2$		
X	Y	Z
x1	y1	z1
x2	y2	z2
x2	y2	z3
x3	y2	z2
x3	y3	z3
x4	y4	z4

b) Phép tách – kết nối không tổn thất thông tin:

R		
X	Y	Z
x1	y1	z1
x2	y2	z2
x3	y2	z3
x4	y3	z4

⇒

R1	
X	Y
x1	y1
x2	y2
x3	y2
x4	y3

R2	
Y	Z
y1	Z1
y2	Z2
y2	Z3
y3	Z4

⇒

$R = R1 \times R2$		
X	Y	Z
x1	y1	z1
x2	y2	z2
x2	y2	z3
x3	y2	z2
x3	y3	z3
x4	y4	z4

Hình 4.1 Tách –kết nối tổn thất thông tin và không tổn thất thông tin

4.3 Thuật toán kiểm tra tách không tổn thất thông tin

Input: $s = \langle \Omega, F \rangle$ là một lược đồ quan hệ.

$\Omega = \{A_1, A_2, \dots, A_n\}$ tập các thuộc tính.

$F = \{f: L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$ tập các phụ thuộc hàm.

$\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ là một phép tách

Output: Một khẳng định phép tách - kết nối φ có tổn thất thông tin hay không.

Phương pháp:

- Tạo một bảng gồm n cột và p dòng. Cột thứ j tương ứng với thuộc tính A_j , hàng thứ i tương ứng với lược đồ quan hệ chiều R_i :

- Cột : A_1, A_2, \dots, A_n
- Hàng: R_1, R_2, \dots, R_p

- Các phần tử của bảng:

$$\alpha(i,j) = \begin{cases} a_i & \text{nếu } A_i \in \Omega_i \\ b_{ij} & \text{nếu } A_i \notin \Omega_i \end{cases}$$

$i = 1 \div n, j = 1 \div p.$

- Áp dụng các phụ thuộc $X \rightarrow Y \in F$ để thay đổi các giá trị của bảng như sau: tìm các hàng giống nhau trên trong các cột thuộc tính của X , trong các cột thuộc tính Y nếu có giá trị là a sẽ thay giá trị các cột trong Y là a_i , nếu không có a_i , thay thế bằng b_{ij} .
- Xét lặp các phụ thuộc trong F cho đến khi không có sự thay đổi trong bảng.
- Việc duyệt bảng bao gồm sắp xếp bảng theo cột có tính các thuộc tính xuất hiện về trái của phụ thuộc hàm. Nếu có k thuộc tính như vậy thì việc thực hiện sắp xếp cần thực hiện trong $n * k$ bước.
- Điền các ký hiệu trong các cột có thuộc tính xuất hiện về phải của phụ thuộc hàm nếu các hàng bằng nhau trên về trái. Công việc này cần $O(k)$ thời gian cho mỗi phụ thuộc. Tổng tất cả độ dài về trái của tất cả các phụ thuộc hàm trong một lần duyệt không quá n , nên toàn bộ thời gian cho một lần duyệt nhiều nhất là $k*n$.
- Khi không còn một ký hiệu nào được làm bằng nhau trong một lần duyệt thì có thể kết thúc việc lặp các bước duyệt vì bảng thu được thoả mọi phụ thuộc.
- Kiểm tra có tồn tại một hàng R_i sao cho giá trị của chứa các ký hiệu a_1, a_2, \dots, a_n hay không. Nếu có, tách - kết nối không tổn thất thông tin. Ngược lại, không tồn tại dòng nào như vậy, nghĩa là các lược đồ quan hệ chiều khi kết nối bị tổn thất thông tin. Điều này có thể suy ra từ định nghĩa của phép tách – kết nối tự nhiên.
- Do đó thời gian tiêu dùng toàn bộ cho thuật toán nhiều nhất là $k*n^2*p$, Nếu $k \leq n$ và $p \leq n$ hiển nhiên thuật toán có thời gian chi phí nhiều nhất là n^4 .

Ví dụ: Cho $\Omega := \{A, B, C, D, E, F\}$ tập các thuộc tính.

Xét phép tách – kết nối $\varphi[\Omega_1, \Omega_2, \Omega_3]$ trong đó:

$$\Omega_1 := \{A, B, D, E\}, \quad \Omega_3 := \{B, C, E, F\},$$

$$\Omega_2 := \{A, C, D, F\} \quad \text{và} \quad F := \{A \rightarrow B, F \rightarrow E\}.$$

Bước 1: Thành lập bảng ban đầu gồm 3 hàng và 6 cột:

	A	B	C	D	E	F
R_1	a_1	a_2	b_{13}	a_4	a_5	b_{16}
R_2	a_1	b_{22}	a_3	a_4	b_{25}	a_6
R_3	b_{31}	a_2	a_3	b_{34}	a_5	a_6

Bước 2: Áp dụng $A \rightarrow B$ suy ra $b_{22} = a_2$

	A	B	C	D	E	F
R_1	a_1	a_2	b_{13}	a_4	a_5	b_{16}
R_2	a_1	a_2	a_3	a_4	b_{25}	a_6
R_3	b_{31}	a_2	a_3	b_{34}	a_5	a_6

Bước 3: Áp dụng $F \rightarrow E$ suy ra $b_{25} = a_5$

	A	B	C	D	E	F
R_1	a_1	a_2	b_{13}	a_4	a_5	b_{16}
R_2	a_1	a_2	a_3	a_4	a_5	a_6
R_3	b_{31}	a_2	a_3	b_{34}	a_5	a_6

Như vậy tồn tại hàng thứ 2 R_2 chứa các ký tự $\{a_1, a_2, a_3, a_4, a_5, a_6\}$. Suy ra phép tách φ có kết nối không tổn thất thông tin.

Ví dụ: Cho $\Omega := \{A, B, C, D, E\}$ tập các thuộc tính.

$$\Omega_1 := \{A, D\}, \quad \Omega_2 := \{A, B\}$$

$$\Omega_3 := \{B, E\}, \quad \Omega_4 := \{C, D, E\}$$

$$\Omega_5 := \{A, E\}$$

$$F := \{A \rightarrow C, B \rightarrow C, C \rightarrow D, E \rightarrow C, CE \rightarrow D\}.$$

Bước 1: Bảng ban đầu gồm 6 hàng và 5 cột:

	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{23}	b_{24}	b_{25}
R_3	b_{31}	a_2	b_{33}	b_{34}	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{53}	b_{53}	b_{54}	a_5

Bước 2: Áp dụng $A \rightarrow C$ và $B \rightarrow C$ suy ra $b_{13} = b_{23} = b_{53} = b_{33}$

	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{13}	b_{24}	b_{25}
R_3	b_{31}	a_2	b_{13}	b_{34}	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{53}	b_{13}	b_{54}	a_5

Bước 3: Áp dụng $C \rightarrow D$ suy ra $a_4 = b_{24} = b_{34} = b_{54}$

	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{13}	a_4	b_{25}
R_3	b_{31}	a_2	b_{13}	a_4	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{53}	b_{13}	a_4	a_5

Bước 4: Áp dụng $DE \rightarrow C$ suy ra $a_3 = b_{13}$

	A	B	C	D	E
R_1	a_1	b_{12}	a_3	a_4	b_{15}
R_2	a_1	a_2	a_3	a_4	b_{25}
R_3	b_{31}	a_2	a_3	a_4	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{53}	a_3	a_4	a_5

Bước 5: Áp dụng $CE \rightarrow A$ suy ra $a_1 = b_{41} = b_{31}$

	A	B	C	D	E
R_1	a_1	b_{12}	a_3	a_4	b_{15}
R_2	a_1	a_2	a_3	a_4	b_{25}
R_3	a_1	a_2	a_3	a_4	a_5
R_4	a_1	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{53}	a_3	a_4	a_5

Như vậy tồn tại hàng thứ 3 R_3 có chứa các ký tự $\{a_1, a_2, a_3, a_4, a_5\}$. Suy ra các phép tách - kết nối không tổn thất thông tin.

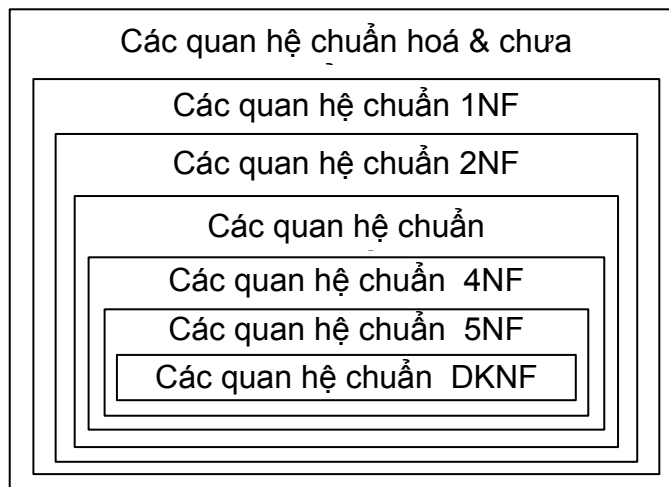
4.4 Các dạng chuẩn của lược đồ quan hệ

Phần này sẽ tìm hiểu và nghiên cứu khái niệm chuẩn hoá dữ liệu quan hệ. Đã có nhiều kết quả trong lĩnh vực chuẩn hoá, song vẫn còn nhiều vấn đề bất lợi khi tìm kiếm thông tin. Vì vậy người ta cũng chỉ mới đề xuất các dạng sao cho tiện lợi, đơn giản hơn trong quá trình thao tác các ngôn ngữ con dữ liệu.

Trong “ Further Normalization of the Data base relational Model” lần đầu tiên E.F. Codd đã đưa ra các khái niệm về chuẩn hoá quan hệ. Ông đã chia các quan hệ thành 3 lớp và gọi chúng là các quan hệ dạng chuẩn thứ nhất, dạng chuẩn thứ hai và dạng chuẩn thứ ba. Trong “ Multivalued Dependencies and a New Normal Form Relational Databaseses” R. Fagin đã đưa ra khái niệm dạng chuẩn thứ tư, theo ông đó là dạng tiện lợi hơn so với quan hệ dạng chuẩn 3. Mục đích của chương này là minh hoạ những ưu điểm của các quan hệ dạng chuẩn 4 và trình bày phương pháp chuyển đổi một quan hệ chưa là dạng chuẩn 4 về một nhóm tương đương các quan hệ dạng chuẩn 4.

4.4.1 Sự cần thiết chuẩn hoá dữ liệu

Khi thực hiện các phép lưu trữ trên các quan hệ chưa được chuẩn hoá thường xuất hiện các dị thường thông tin. Nghĩa là trong dữ liệu lưu trữ, sự dư thừa, mất dữ liệu, mâu thuẫn hay không nhất quán dữ liệu có thể xảy ra khi cập nhật, bổ sung hay sửa đổi dữ liệu. Dị thường thông tin là nguyên nhân gây cản trở cho việc tìm kiếm, hỏi đáp thông tin. Mục tiêu của chuẩn hoá dữ liệu là triệt tiêu mức cao nhất khả năng xuất hiện các dị thường thông tin khi thực hiện các phép lưu trữ. Có như vậy mục tiêu của các hệ cơ sở mới được bảo đảm. Dữ liệu lưu trữ phản ánh thế giới hiện thực khách quan, đầy đủ hơn và sinh động hơn.



Hình 4.2 Mối quan hệ giữa các lớp dạng chuẩn lược đồ quan hệ

Khi thiết kế và cài đặt các hệ CSDL, chuẩn hoá là quá trình khảo sát danh sách các thuộc tính và áp dụng tập các quy tắc phân tích vào danh sách đó, biến đổi chúng thành nhiều tập nhỏ hơn sao cho:

- *Tối thiểu việc lặp lại.*
- *Tránh dị thường thông tin.*
- *Xác định và giải quyết được sự không rõ ràng, nhập nhằng trong suy diễn.*

Quá trình chuẩn hoá là quá trình tách lược đồ quan hệ về một nhóm tương đương các lược đồ quan hệ chiếu sao cho khi kết nối tự nhiên không làm tổn thất thông tin và bảo toàn được

các phụ thuộc hàm. Cơ sở chuẩn hoá dựa trên các khái niệm về phụ thuộc hàm, phụ thuộc đầy đủ, khoá, các thuộc tính không khoá... Một mô hình được xem là mô hình chuẩn hoá tốt, lý tưởng là mô hình ở đó mỗi một thuộc tính không khoá phụ thuộc hàm vào khoá.

Quan hệ chuẩn hoá là những quan hệ mà mỗi giá trị thuộc tính trong bộ là những thuộc tính nguyên tố, không phân chia ra được. Nói cách khác, mỗi toạ độ của hàng và cột trong quan hệ chỉ có đúng một giá trị chứ không phải là một tập các giá trị.

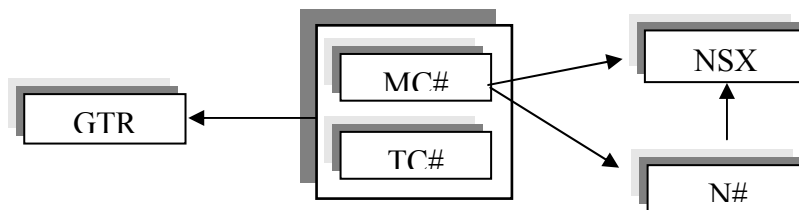
Ví dụ Lược đồ quan hệ QLCAP(TC#,GTR,MC#,N#,NSX)

Trong đó: TC# : Mã tuyến cáp GTR : Giá trị của cáp
 MC# : Mã cáp N# : Mã nước sản xuất
 NSX : Nước sản xuất.

Ngữ nghĩa dữ liệu như sau:

- Trong một tuyến cáp, giá trị của một loại cáp được xác định duy nhất.
- Mỗi mã cáp xác định mã nước sản xuất cáp, và
- Mã nước xác định tên nước sản xuất

$F = \{(TC\#,MC\#) \rightarrow GTR, MC\# \rightarrow NSX, MC\# \rightarrow N\#, N\# \rightarrow NSX\}$ tập các phụ thuộc hàm. Các thuộc tính khoá: (TC#,MC#) và các thuộc tính không khoá: GTR, N#, NSX.



Hình 4.3 Sơ đồ các phụ thuộc hàm trong lược đồ quan hệ QLCAP

TC#	GTR	MC#	N#	NSX
T01	200	C01	HAQ	Hàn Quốc
T01	250	C02	HAQ	Hàn Quốc
T01	220	C03	VTN	Việt Nam
T02	500	C01	HAQ	Hàn Quốc
T02	400	C04	JAN	Nhật Bản
T03	100	C05	RUS	Nga
T04	400	C06	CHN	Trung Quốc
T04	450	C03	VTN	Việt Nam

Hình 4.4 Một thể hiện của lược đồ quan hệ QLCAP

4.4.2 Dạng chuẩn 1 – 1NF (First Normal Form)

Lược đồ quan hệ $s = \langle \Omega, F \rangle$ được gọi là dạng chuẩn 1 – 1NF, khi và chỉ khi các thuộc tính chỉ chứa các giá trị nguyên tố (giá trị nguyên tố là giá trị không thể tách).

Thường các quan hệ chuẩn hoá là các quan hệ dạng chuẩn 1NF. Tuy nhiên, cấu trúc biểu diễn dữ liệu trong các quan hệ dạng 1NF còn nhiều điều bất tiện. Vì vậy khi thao tác thực

Hiện các phép chèn thêm, sửa đổi hay bổ sung cập nhật dữ liệu thường xuất hiện dị thường thông tin, không thể chấp nhận được trong quá trình tìm kiếm.

Theo định nghĩa, quan hệ QLCAP là một quan hệ dạng chuẩn thứ nhất 1NF.

Ví dụ: Xét quan hệ $S(S\#,PRO)$, thấy rằng thuộc tính PRO chứa các giá trị không nguyên tố. Vì vậy quan hệ này không phải là quan hệ dạng chuẩn 1NF.

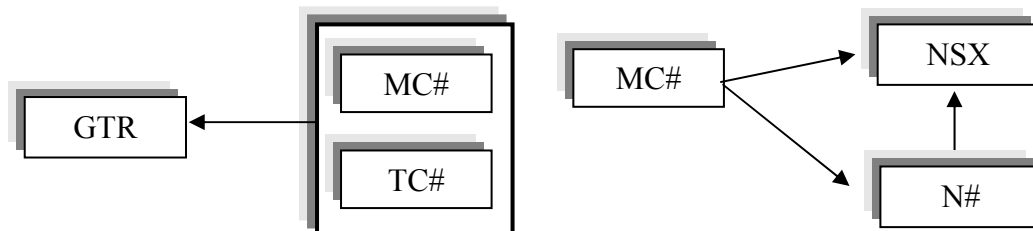
S#	PRO	
	P#	QTY
S1	100	1
	200	1
	300	2
S2	100	2
	200	2
S3	300	3
	100	1

Hình 4.5 Một thí dụ quan hệ không 1NF

4.4.3 Dạng chuẩn 2 – 2NF (Second Normal Form)

Lược đồ quan hệ $s = \langle \Omega, F \rangle$ được gọi là dạng chuẩn 2NF, khi và chỉ khi nó là dạng chuẩn 1NF và các thuộc tính không khoá phụ thuộc hàm đầy đủ vào khoá. Nói cách khác, nếu $s = \langle \Omega, F \rangle$ là dạng chuẩn 1NF và không tồn tại các phụ thuộc hàm $X \rightarrow Y \in F^+$ sao cho X là tập con thực sự của khoá và Y là thuộc tính không khoá.

Lược đồ quan hệ QLCAP không là dạng chuẩn 2NF, vì các thuộc tính không khoá N# và NSX phụ thuộc hàm vào $MC\# \subset \{TC\#, MC\#\}$, tức là MC# phụ thuộc không đầy đủ vào khoá $\{TC\#, MC\#\}$. Với cấu trúc lưu trữ của QLCAP, khi thực hiện các phép chèn thêm, loại bỏ hay sửa đổi dữ liệu sẽ xuất hiện dị thường thông tin. Không thể chèn thêm thông tin về một loại cáp mới khi chưa xác định giá trị của tuyến cáp, vì TC# không thể nhận giá trị không xác định. Tương tự, không thể xoá một tuyến duy nhất có một loại cáp duy nhất, vì như vậy sẽ mất thông tin về tuyến cáp và loại cáp đó. Có thể tách QLCAP thành 2 lược đồ quan hệ TUYEN và CAP_NSX về dạng chuẩn 2NF không tổn thất thông tin. Cấu trúc lưu trữ của TUYEN và CAP_NSX phản ánh thể giới dữ liệu khách quan hơn, trung thực hơn so với cấu trúc của QLCAP.



Hình 4.6 TUYEN

CAP_NSX

Ví dụ Cho $\Omega = \{A, B, C, D, E, G\}$ và

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

Các khoá của lược đồ quan hệ gồm: $K_1 = \{A, B\}$, $K_2 = \{B, E\}$, $K_3 = \{C, G\}$, $K_4 = \{C, E\}$, $K_5 = \{C, D\}$, $K_6 = \{B, C\}$. Như vậy không tồn tại các thuộc tính không khoá, vì vậy lược đồ quan hệ trên dạng chuẩn 2NF.

4.4.4 Dạng chuẩn 3 - 3NF (Third Normal Form)

Lược đồ quan hệ $s = \langle \Omega, F \rangle$ được gọi là dạng chuẩn 3NF, khi và chỉ khi không tồn tại phụ thuộc hàm $X \rightarrow Y \in F^+$ sao cho $X^+ \neq \Omega$, $Y \not\subset X$ và Y là thuộc tính không khoá. Nói cách khác nếu $X \rightarrow Y \in F^+$, $Y \not\subset X$ thì khi đó hoặc X là khoá của lược đồ quan hệ hoặc Y là một thuộc tính của khoá.

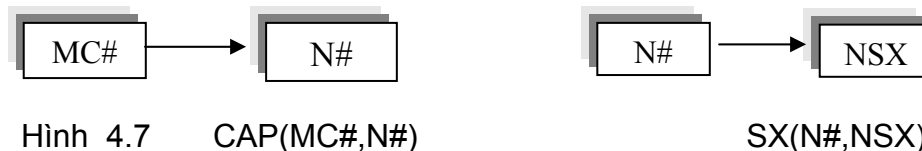
Trong lược đồ 2NF, cấm tất cả các thuộc tính không khoá phụ thuộc vào các tập con thực sự của khoá. Trong dạng chuẩn 3NF, cấm các thuộc tính không khoá phụ thuộc hàm vào tất cả các tập thuộc tính có bao đóng khác Ω .

Các khẳng định sau là tương đương

- Không tồn tại phụ thuộc hàm $X \rightarrow Y \in F^+$ sao cho $X^+ \neq \Omega$, $Y \not\subset X$ và Y là thuộc tính không khoá.
- Nếu $X \rightarrow Y \in F^+$, $Y \not\subset X$ khi đó hoặc X là khoá của lược đồ quan hệ hoặc Y là một thuộc tính của khoá.
- Không tồn tại thuộc tính không khoá phụ thuộc bắc cầu vào khoá.
- Tập các thuộc tính không khoá bằng rỗng.

Như vậy một lược đồ quan hệ dạng chuẩn 3NF thì cũng là dạng chuẩn 2NF.

Lược đồ quan hệ TUYEN là lược đồ quan hệ dạng chuẩn 3NF. Vì (MC#,TC#) là các thuộc tính khoá. Lược đồ quan hệ CAP_NSX ở dạng chuẩn 2NF, nhưng không phải là dạng chuẩn 3NF. Vì $N\# \rightarrow NSX$, $N\#$ và NSX là các thuộc tính không khoá. Hiển nhiên có thể kiểm tra được rằng khi chèn thêm dữ liệu, hoặc loại bỏ dữ liệu hoặc sửa đổi giá trị của một số thuộc tính trong một quan hệ bất kỳ của lược đồ quan hệ CAP_NSX sẽ xuất hiện các bất thường khi tìm kiếm, tra cứu thông tin. Có thể tách CAP_NSX thành 2 quan hệ chuẩn 3NF CAP và SX (hình 4.7). Phép tách có kết nối không tồn thất thông tin.



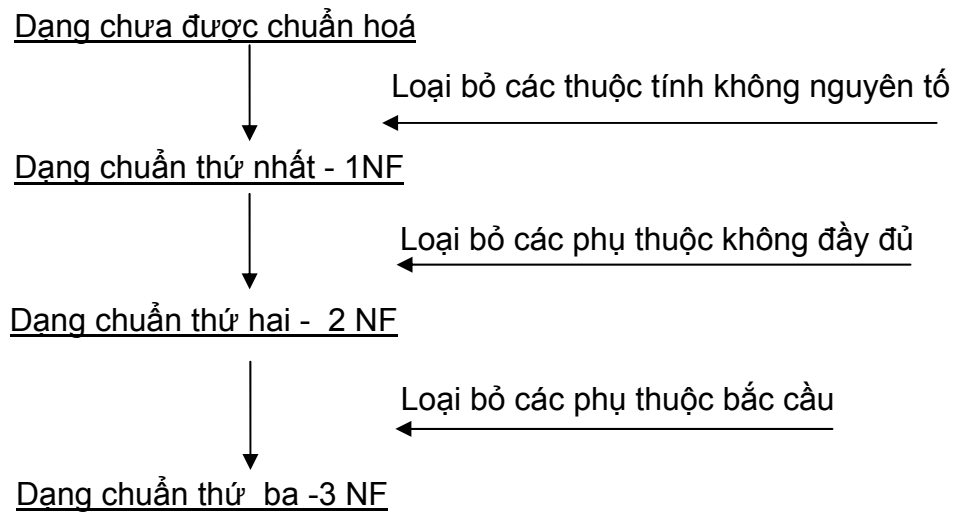
Từ một lược đồ quan hệ 2NF nhưng không phải 3NF được biến đổi thành một họ tương đương các lược đồ quan hệ dạng 3NF. Quá trình chuyển đổi là quá trình khả nghịch, do đó không làm mất thông tin trong quá trình chuyển đổi. Các lược đồ quan hệ kết quả sau biến đổi được gọi là các lược đồ quan hệ chiếu. Nhóm các lược đồ quan hệ chiếu tương đương với lược đồ quan hệ ban đầu theo nghĩa là lược đồ quan hệ ban đầu có thể nhận được bằng cách thực hiện các phép kết nối các lược đồ quan hệ chiếu tương ứng. Như vậy quá trình

biến đổi sẽ không làm mất thông tin. Thông tin của lược đồ quan hệ ban đầu có thể nhận được từ các lược đồ quan hệ chiểu.

4.4.5 Quá trình tách một lược đồ quan hệ 1NF về nhóm tương đương 3NF.

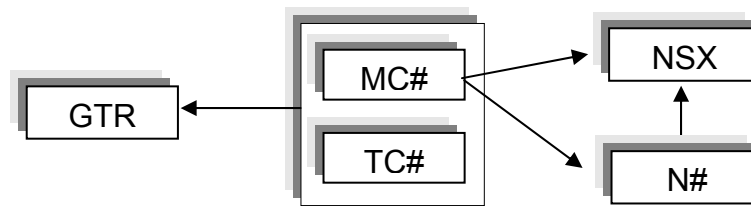
Từ một lược đồ quan hệ dạng chuẩn thứ nhất có thể chuyển đổi về nhóm các lược đồ quan hệ chiểu dạng chuẩn 3NF bằng cách loại bỏ các phụ thuộc không đầy đủ vào khoá và loại bỏ các phụ thuộc bậc cầu vào khoá. Quá trình tách là quá trình loại bỏ đi thường thông tin gây những điều bất tiện khi thao tác lưu trữ dữ liệu. Phép tách các lược đồ quan hệ là phép tách có kết nối không tổn thất thông tin.

Codd đã đưa ra 3 dạng chuẩn của quan hệ và lộ trình chuyển đổi như sau:

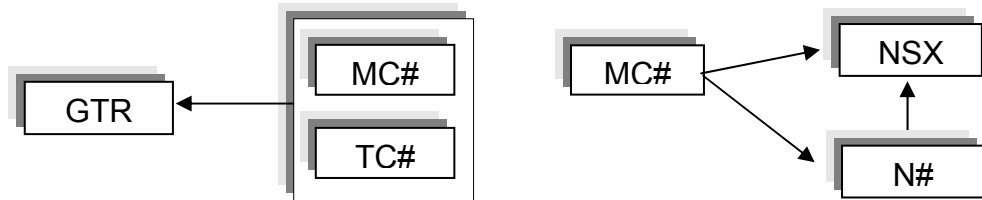


Hình 4.8 Các cấp chuẩn hoá quan hệ

Từ quan hệ QLCAP(TC#, MC#, GTR, N#,NSX)



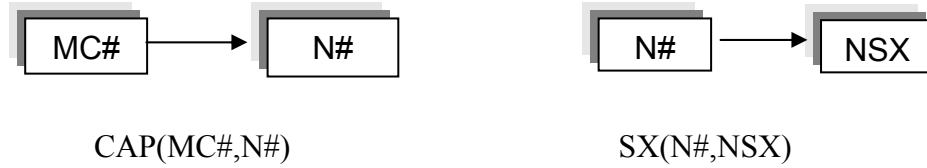
Loại bỏ các phụ thuộc không đầy đủ vào khoá



TUYEN (TC#, MC#, GTR)

CAP_NSX(MC#, N#, NSX)

Loại bỏ các phụ thuộc bắc cầu vào khoá:



Hình 4.9 Quá trình tách QLCAP về các lược đồ quan hệ 3NF

4.5 Dạng chuẩn Boyce Codd

Khái niệm lược đồ quan hệ dạng chuẩn 1NF và dạng chuẩn 2NF là những khái niệm trung gian trên chặng đường đi tới chuẩn 3NF. Tuy nhiên với khái niệm chuẩn 3NF chưa thoả mãn cho các trường hợp lược đồ quan hệ có hơn một khoá, đặc biệt các trường hợp khoá giao nhau. Heath là người đầu tiên, sau đó Boyce Codd đã đưa ra định nghĩa dạng chuẩn Boyce Codd có thể thoả mãn các yêu cầu trên. Khái niệm dạng chuẩn Boyce Codd không dựa vào các khái niệm dạng chuẩn 1NF và 2NF, tức là không dựa vào các khái niệm về phụ thuộc hàm đầy đủ và phụ thuộc bắc cầu.

4.5.1 Định nghĩa dạng chuẩn Boyce Codd

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

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

Có thể định nghĩa cách khác như sau: $s = \langle \Omega, F \rangle$ được gọi là dạng chuẩn Boyce Codd, khi và chỉ khi mọi định thuộc là khoá của lược đồ quan hệ.

Các khẳng định sau là tương đương

- a) Nếu $X \rightarrow Y \in F^+$, $Y \not\subseteq X$ khi đó X là khóa của lược đồ quan hệ.
- b) Nếu $X \rightarrow Y \in F^+$, $Y \not\subseteq X$ khi đó $X^+ = \Omega$.
- c) Mọi định thuộc là khoá của lược đồ quan hệ. (X là định thuộc khi và chỉ khi tồn tại $Y \subseteq \Omega$ sao cho $X \rightarrow Y \in F^+$ là phụ thuộc đầy đủ).

Định nghĩa dạng chuẩn 3NF và BCNF giống nhau trừ mệnh đề “ hoặc Y là một thuộc tính của khóa”. Điều này có nghĩa là nếu lược đồ quan hệ dạng chuẩn BCNF thì cũng là 3NF. Trong định nghĩa 3NF, loại trừ các thuộc tính không khóa phụ thuộc vào các thuộc tính có bao đóng khác Ω , còn trong định nghĩa BCNF thì loại trừ tất cả các thuộc tính phụ thuộc vào các thuộc tính có bao đóng khác Ω .

Các dạng chuẩn lược đồ quan hệ lồng nhau, nghĩa là nếu lược đồ quan hệ dạng chuẩn BCNF thì cũng là dạng chuẩn 3NF, nếu đã là 3NF cũng là dạng chuẩn 2NF và nếu 2NF cũng là dạng chuẩn 1NF. Ký hiệu

$$BCNF \subset 3NF \subset 2NF \subset 1NF.$$

4.5.2 Nhận biết dạng chuẩn 3NF & BCNF

- a) Nhận biết một lược đồ quan hệ là dạng chuẩn 3NF:
 - Xác định tập các thuộc tính không khóa Y.
 - Kiểm tra xem có tồn tại phụ thuộc $X \rightarrow Y \in F^+$, $Y \not\subset X$ và $X^+ \neq \Omega$.
- b) Nhận biết một lược đồ quan hệ là dạng chuẩn BCNF
 - Xác định các tập các thuộc tính X sao cho $X^+ \neq \Omega$.
 - Kiểm tra xem có tồn tại phụ thuộc $X \rightarrow Y \in F^+$, $Y \not\subset X$.

Ví dụ

- a) Cho $\Omega = \{A, B, C, D, E, G, H\}$ và $F = \{C \rightarrow AB, D \rightarrow E, B \rightarrow G\}$.
 - Khóa của lược đồ $K = \{H, C, D\}$.
 - Tập các thuộc tính không khoá: $NK = \{A, B, E, G\}$.
 - Vế phải của các phụ thuộc $C \rightarrow AB, D \rightarrow E$ và $B \rightarrow G$ là các thuộc tính không khoá, và $C^+ \neq \Omega$, $D^+ \neq \Omega$ và $B^+ \neq \Omega$.
 - Lược đồ là dạng chuẩn 3NF
- b) Cho $\Omega = \{A, B, C, D\}$ và $F = \{AB \rightarrow C, C \rightarrow ABD\}$.
 - Tập có bao đóng khác Ω : $X = A, X = B, X = D, X = AD, X = BD$.
 - Không tồn tại $X \rightarrow Y \in F^{++}$, $Y \not\subset X$, suy ra BCNF.
- c) Cho $\Omega = \{A, B, C, D, E, G, H\}$ và $F = \{A \rightarrow BC, D \rightarrow E, H \rightarrow G\}$.
 - Tồn tại $D \rightarrow E \in F^+$, $D^+ \neq \Omega$, suy ra lược đồ không BCNF.

Tồn tại lược đồ quan hệ không là dạng chuẩn 3NF thì cũng không là chuẩn Boyce Codd.

4.6 Thuật toán tách lược đồ quan hệ không tổn thất thông tin

4.6.1 Phép tách về các lược đồ dạng chuẩn 3NF có bảo toàn phụ thuộc

Input: Cho $s = \langle \Omega, F \rangle$ là một lược đồ quan hệ, trong đó
 $\Omega = \{A_1, A_2, \dots, A_n\}$ tập các thuộc tính và
 $F = \{f : L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$ tập các phụ thuộc hàm.
Không mất tính tổng quát, giả sử tập F là phủ cực tiểu.

Output: Một phép tách bảo toàn phụ thuộc sao cho mỗi một lược đồ quan hệ chiếu có dạng chuẩn 3NF tương ứng với tập phụ thuộc chiếu của F trên nó.

Phương pháp:

- Nếu có những thuộc tính của Ω không có mặt trong các vế phải và trái của các phụ thuộc hàm, thì tập các thuộc tính này sẽ tạo ra một lược đồ quan hệ chiếu. Loại bỏ tất cả những thuộc tính này ra khỏi Ω trong quá trình thực hiện thuật toán.
- Nếu tồn tại phụ thuộc $X \rightarrow A \in F$ sao cho vế phải và trái của nó chứa tất cả các thuộc tính của Ω , khi đó kết quả phép tách chính là lược đồ quan hệ s.
- Ngược lại, $\Omega_i = \{XA\}$ là một thành phần của phép tách.

4.6.2 Thuật toán tách bảo toàn phụ thuộc và không tổn thất thông tin về các lược đồ dạng chuẩn 3NF

Input: Cho $s = \langle \Omega, F \rangle$ là một lược đồ quan hệ. Trong đó

$\Omega = \{A_1, A_2, \dots, A_n\}$ tập các thuộc tính và
F phủ cực tiểu các phụ thuộc hàm.
K là khoá của lược đồ quan hệ.

Output: Phép tách $\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ bảo toàn phụ thuộc và không tổn thất thông tin sao cho mỗi một lược đồ quan hệ chiếu có dạng chuẩn 3NF tương ứng với tập phụ thuộc chiếu của F trên nó.

Phương pháp:

1. Xác định các thuộc tính của Ω không có mặt trong các vế phải và trái của các phụ thuộc hàm. Ký hiệu tập các thuộc tính này là A.
 - Tạo ra lược đồ quan hệ chiếu trên tập các thuộc tính A.
 - Loại bỏ tất cả những thuộc tính A ra khỏi Ω trong quá trình thực hiện thuật toán: $\Omega = \Omega - \{A\}$.
2. Nếu tồn tại phụ thuộc $X \rightarrow A \in F$ sao cho vế phải và trái của nó chứa tất cả các thuộc tính của Ω , khi đó kết quả phép tách $\varphi[X, A]$.
3. Với mọi $X \rightarrow A \in F$ trong đó thuộc tính A là thuộc tính đơn (F là phủ cực tiểu). $\Omega_i = \{XA\}$, với $X \rightarrow A, i=1 \div p$.
4. Nếu có một số phụ thuộc cùng vế trái: $X \rightarrow A_1 \in F, X \rightarrow A_2 \in F, \dots, X \rightarrow A_k \in F$ thì có thể hợp lại thành dạng: $\Omega_j = \{XA_1 A_2 \dots A_n\}$, với $X \rightarrow A_i, i=1 \div k$.
5. Nếu các thuộc tính của khoá K không xuất hiện trong các tập Ω_j được tạo bởi các bước trên khi đó một thành của phép tách sẽ được định nghĩa bởi khoá K.
6. Phép tách $\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ bảo toàn phụ thuộc và không tổn thất thông tin. Mỗi một lược đồ quan hệ chiếu có dạng chuẩn 3NF tương ứng với tập phụ thuộc chiếu của F trên nó.

Ví dụ Cho $\Omega = \{X, Y, Z, W, Q\}$ và phủ cực tiểu
 $F = \{X \rightarrow Y, XZ \rightarrow W, YW \rightarrow Q\}$

Xác định một phép tách φ -3NF bảo toàn phụ thuộc và không tổn thất thông tin.

1. XZ là khoá: $(XZ)^+ = \Omega = \{X, Y, Z, W, Q\}$
2. Thực hiện phép tách:
 - Không tồn tại các thuộc tính mà các thuộc tính đó không xuất hiện trong các vế của phụ thuộc hàm.
 - Không tồn tại phụ thuộc hàm chứa các thuộc tính còn lại của Ω .
 - $X \rightarrow Y$: $\Omega_1 = \{XY\}$.
 - $XZ \rightarrow W$: $\Omega_2 = \{X, Z, W\}$.
 - $YW \rightarrow Q$: $\Omega_3 = \{Y, W, Q\}$.
 - Khoá $XZ \subseteq \Omega_2 = \{X, Z, W\}$
3. Vậy phép tách $\varphi [XY, XZW, YWQ]$.
4. Kiểm tra tổn thất thông tin:

	X	Y	Z	W	Q	⇒	X	Y	Z	W	Q	
Ω_1	a	a	b	b	b		Ω_1	a	a	b	b	b
Ω_2	a	b	a	a	b		Ω_2	a	a	a	a	a
Ω_3	b	a	b	a	a		Ω_3	b	a	b	a	a

4.6.3 Thuật toán tách về dạng chuẩn BCNF không tổn thất

Input: $s = \langle \Omega, F \rangle$ lược đồ quan hệ.
 $\Omega = \{A_1, A_2, \dots, A_n\}$ tập các thuộc tính.
 $F = \{L_j \rightarrow R_j \mid L_j, R_j \subseteq \Omega\}$ tập các phụ thuộc.

Output: Phép tách $\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ không tổn thất thông tin. Mỗi một lược đồ quan hệ chiếu có dạng chuẩn BCNF tương ứng với tập phụ thuộc chiếu của F trên nó.

Phương pháp: Phương pháp chủ yếu của thuật toán là tách lược đồ $s = \langle \Omega, F \rangle$ thành 2 lược đồ. Chọn bất kỳ $X \rightarrow A \in F^+$ sao cho X không là khoá và $A \notin X$. Khi đó lược đồ có tập các thuộc tính XA sẽ có dạng chuẩn BCNF và phụ thuộc hàm $X \rightarrow A$ sẽ thoả trên nó. Lược đồ thứ 2 có tập các thuộc tính $\Omega \setminus A$. Hiển nhiên, khi kết nối lược đồ có tập thuộc tính $\Omega \setminus A$ với lược đồ có tập thuộc tính XA không tổn thất thông tin. Tiếp tục tách $\Omega \setminus A$ cho đến khi thoả điều kiện (b) của định lý 4.11, cũng là lược đồ có dạng chuẩn BCNF. Theo định lý 4.10, các lược đồ quan hệ chiếu cũng có dạng BCNF và có kết nối không tổn thất thông tin.

$$\begin{aligned} \text{Đặt } \Omega_1 &:= XA, \quad \Omega_2 := \Omega \setminus A \\ \Omega_1 \cap \Omega_2 &= X \\ \Omega_1 \setminus \Omega_2 &= A \end{aligned}$$

$$\text{Suy ra } \Omega_1 \cap \Omega_2 \rightarrow \Omega_1 \setminus \Omega_2.$$

Phép tách $\varphi [\Omega_1, \Omega_2]$ không tổn thất khi và chỉ khi $\Omega_1 \cap \Omega_2 \rightarrow \Omega_1 \setminus \Omega_2$. Tiếp tục cho đến khi tất cả lược đồ ở dạng chuẩn Boyce Codd.

- Sử dụng phương pháp lặp liên tiếp. Mỗi bước đảm bảo không làm tổn thất thông tin.
- Bước đầu áp dụng đối với lược đồ quan hệ $S = \langle \Omega, F \rangle$.
- Giả sử bước thứ i : Lược đồ quan hệ $s_i = \langle \Omega_i, F \mid \Omega_i \rangle$ chưa phải là lược đồ dạng chuẩn Boyce Codd.
- Chọn $X \rightarrow A \in F_i = F \mid \Omega_i$ sao cho X không là khoá và $A \notin X$.
- Đặt $\Omega_{i1} = XA, \quad \Omega_{i2} = \Omega \setminus A$
- Tiếp tục phép lặp.

Ví dụ $\Omega = \{C, T, H, R, S, G\}$, trong đó:

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

Biết rằng:

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

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

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

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

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

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

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

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

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

$$\begin{aligned} S_{221} &= \langle \Omega_{221}, F_{221} \rangle & S_{222} &= \langle \Omega_{222}, F_{222} \rangle \\ \Omega_{221} &= \{ C, \underline{H}, \underline{R} \} & \Omega_{222} &= \{ \underline{H}, \underline{S}, R \} \\ F_{221} &= \{ HR \rightarrow C \} & F_{222} &= \{ HS \rightarrow R \}, \end{aligned}$$

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

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

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

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

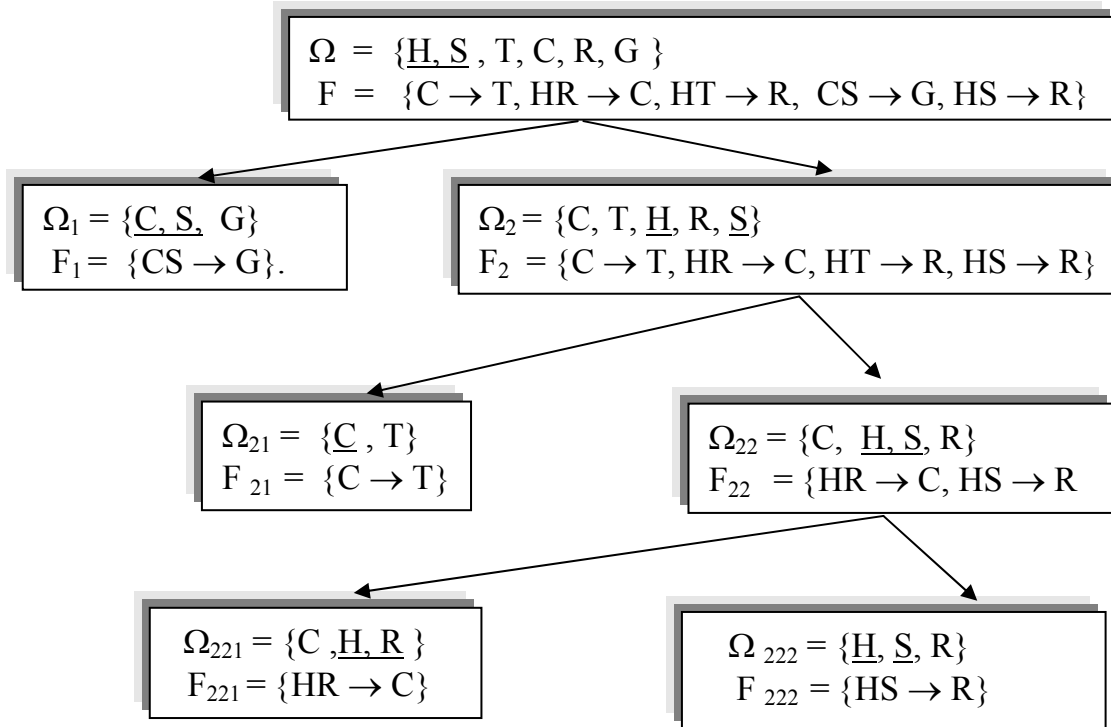
$$\Omega_1 = \{ \underline{C}, \underline{S}, G \}, \quad F_1 = \{ CS \rightarrow G \}$$

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

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

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

Sơ đồ tóm tắt quá trình tách như sau:



Hình 4.10 Quá trình tách- kết nối không mất thông tin

Câu hỏi trắc nghiệm

1. Mục tiêu của lý thuyết CSDL là
 - A. Tính độc lập của dữ liệu
 - B. Tính phụ thuộc dữ liệu.
 - C. Tính đơn giản trong biểu diễn.
 - D. Tính đối xứng trong kết quả..
2. Lược đồ quan hệ $s = \langle \Omega, F \rangle$ được gọi là dạng chuẩn 1 – 1NF, khi và chỉ khi
 - A. Bao đóng các thuộc tính là lớn nhất
 - B. Các thuộc tính khoá phụ thuộc bắc cầu vào khoá.
 - C. Các thuộc tính chỉ chứa các giá trị nguyên tố
 - D. Các phụ thuộc không có vế trái dư thừa
3. Lược đồ quan hệ $s = \langle \Omega, F \rangle$ được gọi là dạng chuẩn 2NF, khi và chỉ khi
 - A. Dạng chuẩn 1NF và tồn tại $X \rightarrow Y \in F +$ sao cho X là tập con của khóa
 - B. Không tồn tại các phụ thuộc dư thừa.
 - C. Các thuộc tính khoá phụ thuộc bắc cầu vào khoá.
 - D. Dạng chuẩn 1NF và các thuộc tính không khoá phụ thuộc đầy đủ vào khoá.

4. Lược đồ quan hệ $s = \langle \Omega, F \rangle$ được gọi là dạng chuẩn 3NF, khi và chỉ khi
 - A. Không tồn tại $X \rightarrow Y \in F^+$ sao cho $X^+ \neq \Omega$, $Y \not\subseteq X$ là thuộc tính không khoá.
 - B. Nếu $X \rightarrow Y \in F^+$, $Y \not\subseteq X$ thì khi đó X
 - C. Y là một thuộc tính của khoá.
 - D. Lược đồ 2NF và các thuộc tính không khoá phụ thuộc khoá.
5. X là định thuộc khi và chỉ khi
 - A. Tồn tại $Y \subseteq \Omega$ sao cho $X \rightarrow Y \in F^+$ là phụ thuộc đầy đủ
 - B. Tồn tại $Y \subseteq \Omega$ sao cho $X \rightarrow Y \in F^+$ là phụ thuộc không đầy đủ .
 - C. Tồn tại $Y \subseteq \Omega$ sao cho $X \rightarrow Y \in F^+$ là phụ thuộc không dư thừa
 - D. Tồn tại $Y \subseteq \Omega$ sao cho $X \rightarrow Y \in F^+$ là phụ thuộc dư thừa
6. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Trong quá trình chuẩn hoá dữ liệu và tìm kiếm thông tin phải thực hiện cácvề tập các lược đồ quan hệ chuẩn hoá, sao cho quá trình tách không làm tổn thất thông tin.
 - A. Phép tách lược đồ quan hệ.
 - B. Phép lưu trữ.
 - C. Phép tìm kiếm thông tin.
 - D. Phép chiếu, chọn và kết nối
7. .Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Mục tiêu của phép tách lược đồ quan hệ là nhằm loại bỏkhi thực hiện các phép lưu trữ như chèn thêm, loại bỏ hay sửa đổi thông tin trong trong các quan hệ lưu trữ.
 - A. Dị thường thông tin
 - B. Không toàn vẹn dữ liệu
 - C. Không bảo mật dữ liệu
 - D. Không nhất quán dữ liệu
8. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Từ một lược đồ quan hệ dạng 1NF có thể chuyển đổi về nhóm các lược đồ quan hệ chiếu dạng chuẩn 3NF bằng cách..... vào khoá và loại bỏ các phụ thuộc bắc cầu vào khoá.
 - A. Loại bỏ các phụ thuộc không đầy đủ.
 - B. Phụ thuộc bắc cầu.
 - C. Phụ thuộc dư thừa.
 - D. Các thuộc tính dư thừa
9. $\Omega = \{A, B, C, D, E, G, H\}$ và $F = \{C \rightarrow AB, D \rightarrow E, B \rightarrow G\}$.
 - A. Lược đồ chưa được chuẩn hoá
 - B. Lược đồ là dạng chuẩn 3NF
 - C. Lược đồ là dạng chuẩn 2NF
 - D. Lược đồ là dạng chuẩn BCNF

10. Hãy so khớp thông tin 2 phần dưới đây với nhau:

Phần 1

- 1 Khi thực hiện các phép lưu trữ trên các quan hệ chưa được chuẩn hoá
- 2 Sự dư thừa, mất dữ liệu, mâu thuẫn hay không nhất quán
- 3 Dự thường thông tin là nguyên nhân gây cản trở cho.....
- 4 Mục tiêu của chuẩn hoá dữ liệu là triết tiêu.....

- a. 1- D; 2- C ; 3 – B ; 4 – A .
- b. 1- C; 2- B; 3 – A; 4 – D
- c. 1- A; 2- B; 3 – C; 4 – D.
- d. 1- D; 2- B; 3 – A; 4 – C.

Phần 2

- A. việc tìm kiếm, hỏi đáp thông tin.
- B. dữ liệu có thể xảy ra khi cập nhật, bổ sung hay sửa đổi dữ liệu
- C. khả năng xuất hiện các dự thường thông tin khi thực hiện các phép lưu trữ
- D. thường xuất hiện các dự thường thông tin.

11. Hãy so khớp thông tin 2 phần dưới đây với nhau:

Phần 1

- 1 Các thuộc tính không khoá
- 2 Các thuộc tính khoá
- 3 Nếu $X \rightarrow Y \in F^+$, $Y \not\subset X$ khi đó X.....
- 4 Mọi định thuộc

- a. 1- D; 2- C ; 3 – B ; 4 – A .
- b. 1- C; 2- B; 3 – A; 4 – D
- c. 1- A; 2- B; 3 – C; 4 – D.
- d. 1- D; 2- A; 3 – B; 4 – C.

Phần 2

- A. phụ thuộc hoàn toàn vào tất cả khoá khác
- B. là khoá của lược đồ quan hệ.
- C. là khoá của lược đồ quan hệ.
- D. phụ thuộc hoàn toàn vào khoá.

Câu hỏi và bài tập:

Câu hỏi

1. Sự cần thiết tách các lược đồ quan hệ.
2. Trình bày khái niệm và định nghĩa phép tách.
3. Hiểu như thế nào về các ký hiệu $\pi_{\Omega_i}(R)$ và $F_i = F |_{\Omega_i}$.
4. Định nghĩa phép tách - kết không tổn thất thông tin. Ví dụ minh hoạ.
5. Cho một số thí dụ về phép tách- kết nối tổn thất và không tổn thất thông tin.
6. Trình bày thuật toán kiểm tra tách - kết nối không tổn thất thông tin.
7. Cho $\varphi [\Omega_1, \Omega_2, \dots, \Omega_p]$ là một phép tách của lược đồ quan hệ $s = \langle \Omega, F \rangle$. Cho một quan hệ bất kỳ $R \in s$ trên Ω . Chứng minh rằng:
 - a) $R \subseteq m_\varphi(R)$.
 - b) Nếu $P = m_\varphi(R)$ thì $R_i = \pi_{\Omega_i}(P)$, với $i = 1 \div p$.
 - c) $m_\varphi(m_\varphi(R)) = m_\varphi(R)$.

8. Cho $\varphi [\Omega_1, \Omega_2]$ là một phép tách của lược đồ quan hệ $s = \langle \Omega, F \rangle$ trong đó Ω là tập các thuộc tính và F là tập các phụ thuộc hàm. Khi đó phép tách - kết nối φ không tổn thất thông tin khi và chỉ khi:
$$\Omega_1 \cap \Omega_2 \rightarrow \Omega_1 \setminus \Omega_2 \text{ hoặc } \Omega_1 \cap \Omega_2 \rightarrow \Omega_2 \setminus \Omega_1 .$$
9. Trình bày khái niệm và định nghĩa phép chiếu bảo toàn các phụ thuộc.
10. Thuật toán kiểm tra bảo toàn tập phụ thuộc hàm.
11. Cho $\forall Z \subseteq \Omega$, tập F , đặt $Z^+ = Z \cup ((Z \cap \Omega)^+ \cap \Omega)$, khi đó sẽ tồn tại các thuộc tính A sao cho $(Z \cap \Omega) \rightarrow A \in \pi_{\Omega}(F)$.
12. Phát biểu định nghĩa dạng chuẩn thứ nhất – 1NF (First Normal Form)
13. Phát biểu định nghĩa dạng chuẩn thứ hai – 2NF (Second Normal Form)
14. Phát biểu định nghĩa dạng chuẩn thứ 3 - 3NF (Third Normal Form)
15. Các khẳng định sau là tương đương
 - a) Không tồn tại phụ thuộc hàm $X \rightarrow Y \in F^+$ sao cho $X^+ \neq \Omega$, $Y \not\subset X$ và Y là thuộc tính không khóa.
 - b) Nếu $X \rightarrow Y \in F^+$, $Y \not\subset X$ khi đó hoặc X là khóa của lược đồ quan hệ hoặc Y là một thuộc tính của khóa.
 - c) Không tồn tại phụ thuộc bắc cầu vào khóa.
16. Dạng chuẩn 3NF thì cũng là lược đồ quan hệ chuẩn 2NF.
17. Một lược đồ quan hệ 1NF và không tồn tại thuộc tính không khóa phụ thuộc bắc cầu vào khóa, khi đó lược đồ dạng chuẩn 2NF.
18. Trình bày quá trình tách một lược đồ quan hệ 1NF về nhóm 3NF.
19. Trình bày khái niệm và định nghĩa dạng chuẩn Boyce Codd.
20. Các khẳng định sau là tương đương
 - a) Nếu $X \rightarrow Y \in F^+$, $Y \not\subset X$ khi đó X là khóa của lược đồ quan hệ.
 - b) Nếu $X \rightarrow Y \in F^+$, $Y \not\subset X$ khi đó $X^+ = \Omega$.
 - c) Mọi định thuộc là khóa của lược đồ quan hệ. (X là định thuộc khi và chỉ khi tồn tại $Y \subseteq \Omega$ sao cho $X \rightarrow Y \in F^+$ là phụ thuộc đầy đủ).
21. Chứng minh rằng $BCNF \subset 3NF \subset 2NF \subset 1NF$.
22. Nhận biết dạng chuẩn 3NF & BCNF.
23. Cho thí dụ 3NF nhưng không là BCNF.
24. Cho thí dụ không 3NF và không là BCNF.
25. Cho một số thí dụ về các dạng chuẩn lược đồ quan hệ
26. Trình bày thuật toán tách – kết nối lược đồ quan hệ không tổn thất thông tin
27. Phép tách lược đồ quan hệ về nhóm tương đương 3NF.
28. Phép tách lược đồ quan hệ về dạng chuẩn Boyce Codd.
29. Chứng minh rằng:
 - a) Nếu Ω chỉ có 2 thuộc tính, khi đó lược đồ ở dạng chuẩn BCNF.

- b) Nếu $s = \langle \Omega, F \rangle$ không dạng chuẩn BCNF, sẽ tồn tại $A, B \subseteq \Omega$ sao cho $(\Omega \setminus AB) \rightarrow A$ hoặc $(\Omega \setminus AB) \rightarrow B$.
30. Trình bày thuật toán tách – kết nối không tổn thất thông tin từ một lược đồ quan hệ $s = \langle \Omega, F \rangle$ về dạng chuẩn Boyce Codd.

Bài tập

- Cho tập các thuộc tính quản lý tổng đài- cước thuê bao: TD# (Mã tổng đài), TD (Tên tổng đài), DL(Dung lượng), MC#(Màu cáp), SDT(Số điện thoại thuê bao), NSX(Nước sản xuất), NG(Ngày lắp đặt tổng đài), GTR(Giá trị tổng đài), VT(Nơi đặt tổng đài), TB(Tên thuê bao), TB#(Mã thuê bao), DC(Địa chỉ thuê bao), SDTD(Số điện thoại đến/đi), NGG(Ngày thực hiện đàm thoại), GB(Giờ bắt đầu đàm thoại),KT(Giờ kết thúc đàm thoại), DD(Đi/đến).
 - Hãy xác định một phép tách không tổn thất về 3NF có bảo toàn phụ thuộc.
 - Hãy xác định một phép tách – kết nối không tổn thất về BCNF.
- Cho tập các thuộc tính quản lý phát hành báo chí QLBC gồm các thuộc tính MK#(Mã khách), K(Tên khách), DC(Địa chỉ), MB#(Mã báo),TB(Tên báo), GIA(Giá báo), SL(số lượng), KY(kỳ phát hành).
 - Hãy xác định một phép tách – kết nối không tổn thất về 3NF có bảo toàn phụ thuộc.
 - Hãy xác định một phép tách – kết nối không tổn thất về BCNF.
- Giả sử các thuộc tính BC (Bưu cục), BC#(Mã bưu cục), MK(Khách gửi), DC(Địa chỉ khách), NH (Người nhận), DCN(Địa chỉ người nhận), ST(Số tiền), BCN(Bưu cục đến), BCD#(Mã bưu cục đến), NG(ngày gửi) HD(Hoá đơn).
 - Hãy xác định một phép tách – kết nối không tổn thất về 3NF có bảo toàn phụ thuộc.
 - Hãy xác định một phép tách – kết nối không tổn thất về BCNF.
 - Có phép tách – kết nối không tổn thất và đồng thời bảo toàn phụ thuộc về BCNF.
- Cho $\Omega = \{ X, Y, Z, W, Q \}$ và $F = \{XY \rightarrow QW, Z \rightarrow Q, W \rightarrow Z, Q \rightarrow X\}$. $\Omega_1 = \{X, Y, Z\}$. Hãy xác định tập các phụ thuộc chiếu F trên Ω_1 : $F_1 = \pi_{\Omega_1}(F)$.
- Cho $\Omega = \{ X, Y, Z, W \}$ và $F = \{X \rightarrow Y, Z \rightarrow W\}$.
Phép tách $\phi[\Omega_1\{X, Y\}, \Omega_2\{Z, W\}]$. Hãy xác định phép tách bảo toàn phụ thuộc ?.
- Cho $\Omega = \{ A, B, C, D \}$ và $F = \{AB \rightarrow C, A \rightarrow D, BD \rightarrow C\}$.
 - Tìm phủ cực tiểu của F.
 - Hãy xác định một phép tách thành 2 lược đồ dạng 3NF bảo toàn phụ thuộc.
 - Xác định các phụ thuộc chiếu trên các tập thuộc tính trên.
 - Kết quả (b) có phải là tách kết nối tổn thất không. Nếu có sửa lại như thế nào để không tổn thất nhưng vẫn bảo toàn phụ thuộc.
- Cho $\Omega = \{ A, B, C \}$ và $F = \{AB \rightarrow C, A \rightarrow B\}$.

- a) Tìm phủ cực tiểu của F.
- b) Giả sử phủ cực tiểu của F là $G = \{AB \rightarrow C, B \rightarrow C\}$. Chứng tỏ rằng điều giả sử là sai bằng cách xây dựng một lược đồ quan hệ thoả F nhưng vi phạm G.
8. Cho $\Omega = \{A, B, C, D\}$ và $F = \{A \rightarrow B, B \rightarrow C, A \rightarrow D, D \rightarrow C\}$, $\varphi[AB, AC, BD]$.
 - a) Tìm các phụ thuộc chiếu trên mỗi tập tương ứng.
 - b) φ là phép tách – kết nối không tổn thất thông tin.
 - c) φ bảo toàn các phụ thuộc hay không.
9. Cho $\Omega = \{A, B, C, D\}$, $F = \{A \rightarrow C, D \rightarrow C, BD \rightarrow A\}$ và $\varphi[AB, ACD, BCD]$. Chứng minh rằng phép tách – kết nối φ không tổn thất thông tin.
10. Cho $\Omega = \{A, B, C, D\}$ và $F = \{A \rightarrow B, B \rightarrow C, D \rightarrow B\}$
 - a) Nếu $\varphi[ACD, BD]$, hãy xác định các phụ thuộc chiếu trên mỗi tập tương ứng.
 - b) Hãy tiếp tục tách về các lược đồ dạng chuẩn BCNF.
11. Cho F là tập các phụ thuộc hàm có vẻ phải chỉ chứa một thuộc tính.
 - a) Giả sử $X \rightarrow A \in F^+$ nhưng không thoả các điều kiện định nghĩa BCNF. Chứng minh rằng tồn tại một phụ thuộc $Y \rightarrow B \in F$ sao cho không thoả BCNF là khoá (không suy dẫn).
 - b) Như (a) cho 3NF.
12. Cho $\Omega = \{A, B, C, D, E\}$ và $F = \{AB \rightarrow CE, E \rightarrow AB, C \rightarrow D\}$. Lược đồ quan hệ thuộc dạng chuẩn cao nhất.
13. Cho $\Omega = \{A, B, C\}$ và $F = \{A \rightarrow B, B \rightarrow C\}$. Chứng minh rằng lược đồ quan hệ không là 3NF. Nếu thêm $C \rightarrow B$ thì sẽ là 3NF ?
14. Chứng minh rằng nếu một lược đồ 3NF thì cũng là 2NF.
15. Chứng minh rằng lược đồ quan hệ $s = \langle \Omega, F \rangle$ ở dạng 3NF khi và chỉ khi :
 - a) $\forall X \subset \Omega$ thì $X^+ = \Omega$.
 - b) Nếu A là thuộc tính không khoá thì $(X-A)^+ = X-a$
16. Cho $\Omega = \{A, B, C, D, E, F\}$ và $F = \{A \rightarrow B, CD \rightarrow A, BC \rightarrow D, AE \rightarrow F, CE \rightarrow D\}$. Chứng minh rằng phép tách $\varphi[AED, AB, AEF, ACE]$ về BCNF tổn thất thông tin.
17. Xây dựng các thí dụ tách một lược đồ quan hệ chưa chuẩn hóa về nhóm các quan hệ dạng chuẩn 3NF không tổn thất thông tin. Cho ví dụ phép tách về 3NF nhưng khi kết nối tự nhiên tổn thất thông tin ?
18. Xây dựng các thí dụ tách một lược đồ quan hệ chưa chuẩn hóa về nhóm các quan hệ dạng chuẩn BCNF không tổn thất thông tin. Cho ví dụ phép tách về BCNF nhưng khi kết nối tự nhiên tổn thất thông tin ?
19. Xây dựng các thí dụ tách một lược đồ quan hệ chưa chuẩn hóa về nhóm các quan hệ dạng chuẩn 4NF không tổn thất thông tin. Cho ví dụ phép tách về 4NF nhưng khi kết nối tự nhiên tổn thất thông tin ?

5 CÁC NGÔN NGỮ THAO TÁC DỮ LIỆU

Chương V sẽ giới thiệu một số ngôn ngữ thao tác dữ liệu, như ngôn ngữ dữ liệu dựa trên đại số quan hệ, ngôn ngữ SQL Ngôn ngữ xử lý dữ liệu được chia thành hai loại khác nhau dựa trên hai chức năng khác nhau. Đó là các chức năng khai báo và chức năng tính toán. Với các trình ứng dụng, dữ liệu thường được mô tả và tính toán khi chương trình thực hiện, còn trong các hệ thống CSDL, dữ liệu được định nghĩa một lần duy nhất. Vì vậy cần có một phương tiện để định nghĩa ngôn ngữ dữ liệu có khả năng mô phỏng các phép tính quan hệ có tính đầy đủ hơn. Nội dung của chương bao gồm:

- Giới thiệu tổng quan về ngôn ngữ thao tác dữ liệu.
- Các phép toán đại số quan hệ
- Đại số quan hệ – ngôn ngữ vấn tin
- Ngôn ngữ vấn tin SQL

5.1 Ngôn ngữ cơ sở dữ liệu

Theo cách nhìn của người sử dụng, ngôn ngữ thao tác dữ liệu là một tập các phép toán dùng để thao tác trên các hệ cơ sở dữ liệu. Như trong các chương II và III đã khảo sát, biểu diễn dữ liệu một cách thống nhất dẫn đến sự thống nhất tương ứng trong các phép toán thao tác, vì thông tin được biểu diễn bằng một và chỉ một cách nên chỉ cần một phép toán cho mỗi một chức năng cơ bản (thêm, bổ sung, loại bỏ...). Ngược với nó, trong nhiều cấu trúc phức tạp, thông tin được biểu diễn bằng nhiều cách. Và vì vậy, cần phải cài đặt nhiều tập các phép toán thao tác khác nhau.

5.1.1 Ngôn ngữ định nghĩa dữ liệu

Lược đồ khái niệm được đặc tả bằng một ngôn ngữ, một phần của hệ quản trị cơ sở dữ liệu, được gọi là ngôn ngữ định nghĩa dữ liệu - DDL (Data Definition Language).

Ngôn ngữ định nghĩa dữ liệu bao gồm một hệ thống ký hiệu mô tả các kiểu thực thể và mối liên hệ giữa chúng theo một mô hình dữ liệu cụ thể. Ngôn ngữ định nghĩa dữ liệu được sử dụng khi thiết kế CSDL hoặc sửa đổi thiết kế CSDL. Nó không sử dụng để truy xuất dữ liệu hoặc sửa đổi dữ liệu. Có những câu lệnh mô tả cấu trúc vật lý theo thuật ngữ trừu tượng. Thiết kế chi tiết CSDL vật lý được thực hiện bởi các thủ tục của hệ quản trị CSDL- DBMS. Khi thực hiện, chúng sẽ biên dịch các câu lệnh ngôn ngữ định nghĩa dữ liệu.

Ví dụ: Về ngôn ngữ định nghĩa dữ liệu SQL:

```
CREATE TABLE QLCAP (TC#:CHAR(3), MC#: CHAR(3), SL: INT,  
DATE: CHAR(6), GTR: INT).  
CREATE INDEX FOR QLCAP ON TC#, MC#;
```

- CREATE TABLE QLCAP mô tả quan hệ với tên là QLCAP và các thuộc tính của nó, đồng thời cấu trúc cài đặt vật lý là các tên trường, kiểu trường là các số nguyên và chuỗi ký tự có chiều dài cố định.

- CREATE INDEX mô tả cần tạo ra một chỉ mục trên số hiệu của tuyến cáp và số hiệu loại cáp như là các thành phần của lược đồ vật lý. Cho phép tìm kiếm các thông tin về các tuyến cáp, về các loại cáp đã được lắp đặt trên các tuyến cáp. Trình biên dịch của DDL có thể chọn một hàm băm có khóa là chuỗi 3 ký tự cho mỗi trường TC# và MC# và nó có thể lưu các bản ghi trong bucket, tùy theo giá trị hàm băm của TC# và MC#.

5.1.2 Ngôn ngữ thao tác dữ liệu

Ngôn ngữ thao tác dữ liệu DML (Data Manipulation Language) hay còn gọi là ngôn ngữ vấn tin (Query Language) dùng để thao tác trên các quan hệ CSDL, bao gồm một số phép toán, các toán hạng là các quan hệ và kết quả của các phép toán cũng là một quan hệ. Ngôn ngữ thao tác dữ liệu gồm 2 nhóm phép toán:

a) Nhóm các phép toán lưu trữ:

- Chèn thêm: Chèn vào CSDL từ vùng làm việc đệm chứa các thông tin về một bản ghi cụ thể.
- Xoá: Xoá một bộ hay xoá một nhóm các bộ.
- Sửa đổi: Sửa đổi giá trị của một số thuộc tính.

b) Nhóm các phép toán tìm kiếm gồm:

- Phép chọn SELECT: Tạo ra một quan hệ mới, các bộ được rút ra một cách duy nhất từ các bộ của quan hệ nguồn thoả mãn một tân từ xác định.
- Phép chiếu PROJECT: Tạo ra một quan hệ mới, các thuộc tính được rút ra, hoặc được biến đổi từ các thuộc tính của quan hệ nguồn, các bộ là các bộ của quan hệ nguồn bỏ đi những bộ trùng lặp.
- Phép kết nối JOIN: Nhằm tạo ra một quan hệ mới, bằng cách nối nhiều quan hệ trên miền thuộc tính chung. Các thuộc tính của quan hệ kết quả là các thuộc tính của các quan hệ thành viên và các bộ là ghép nối các bộ của quan hệ nguồn có cùng chung giá trị thuộc tính chung.

Chương này khảo sát hai lớp ngôn ngữ hỏi đáp cho mô hình quan hệ, đó là:

1. *Ngôn ngữ đại số quan hệ*, trong đó câu hỏi được biểu diễn nhờ áp dụng các phép tính đặc biệt đối với quan hệ.
2. *Ngôn ngữ tính toán tân từ*, trong đó câu hỏi được biểu diễn là một tập các bộ được thoả mãn các tân từ xác định.

5.2 Các phép toán đại số quan hệ

Đại số quan hệ là một trong những ngôn ngữ thao tác dữ liệu, bao gồm các phép toán trên các quan hệ của một cơ sở dữ liệu cho trước. Đó là các phép toán hợp, giao, chiếu và chọn... Tập hợp các phép toán quan hệ tạo nên một cơ chế truy nhập dữ liệu khá linh hoạt và mềm dẻo. Vì vậy người ta thường lấy đại số quan hệ làm đơn vị đo công suất của hệ quản trị cơ sở dữ liệu quan hệ.

5.2.1 Toán hạng của đại số quan hệ

Toán hạng trong đại số quan hệ là các quan hệ hằng hoặc các biến biểu thị cho các quan hệ bậc cố định. Quan hệ R_1 và R_2 là khả hợp khi :

- R_1 và R_2 cùng bậc n .
- Giá trị thuộc tính thứ j của quan hệ này được rút ra từ cùng miền như thuộc tính thứ j của quan hệ kia, $1 \leq j \leq n$.

5.2.2 Phép hợp các quan hệ - UNION

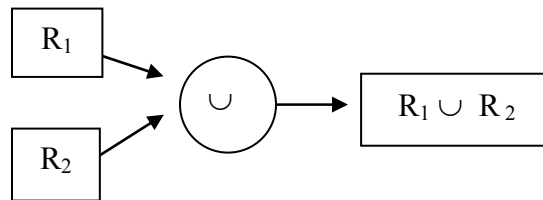
R_1 và R_2 là các quan hệ khả hợp trên cùng tập các thuộc tính của Ω . Khi đó hợp của chúng là một quan hệ trên Ω , các thuộc tính là các thuộc tính của R_1 (hoặc của R_2), các bộ hoặc của R_1 hoặc của R_2 bỏ đi các bộ trùng nhau, chỉ giữ lại những bộ đại diện.

Ký hiệu : $R_1 \cup R_2 = \{t \mid t \in R_1 \text{ hoặc } t \in R_2\}$

UNION (R_1, R_2)

APPEND (R_1, R_2)

$R_1 \cup R_2$



Ví dụ UNION (R_1, R_2)

R_1

M#	N#	NSX
C01	VNA	Việt nam
C02	HQU	Hàn Quốc
C03	JAN	Nhật

R_2

M#	N#	NSX
C04	RUS	Nga
C02	HQU	Hàn Quốc
C05	FRA	Pháp

$R_1 + R_2$

M#	N#	NSX
C01	VNA	Việt nam
C02	HQU	Hàn Quốc
C03	JAN	Nhật
C04	RUS	Nga
C05	FRA	Pháp

Hình 5.1 Mô tả phép hợp đại số quan hệ

5.2.3 Phép giao của các quan hệ - INTERSECT

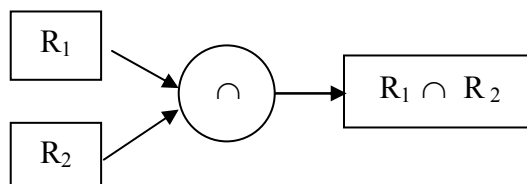
R_1 và R_2 là các quan hệ khả hợp trên cùng tập các thuộc tính của Ω . Khi đó giao của chúng cũng là một quan hệ trên Ω , các thuộc tính của quan hệ kết quả là các thuộc của R_1 hoặc R_2 , các bộ của nó là các bộ có mặt trong R_1 và trong R_2 .

Ký hiệu : $R_1 \cap R_2 = \{t \mid t \in R_1 \text{ And } t \in R_2\}$

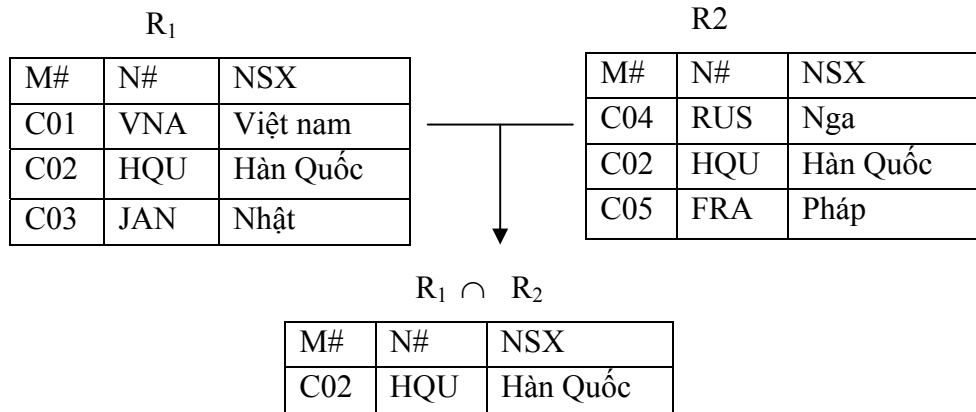
INTERSECT (R_1, R_2)

AND (R_1, R_2)

$R_1 \cap R_2$



Ví dụ INTERSECT (R_1, R_2)



Hình 5.2 Mô tả phép giao đại số quan hệ

5.2.4 Hiệu của các quan hệ - MINUS

R_1 và R_2 là các quan hệ khả hợp trên cùng tập các thuộc tính của Ω . Khi đó hiệu của R_1 và R_2 là quan hệ kết quả KQ trên các thuộc tính Ω . Các thuộc tính của KQ là các thuộc của R_1 hoặc R_2 , các bộ của KQ là các bộ có mặt trong R_1 nhưng không có mặt trong R_2 . (Tương tự hiệu của R_2 và R_1 gồm các bộ có mặt trong R_2 nhưng không có mặt trong R_1)

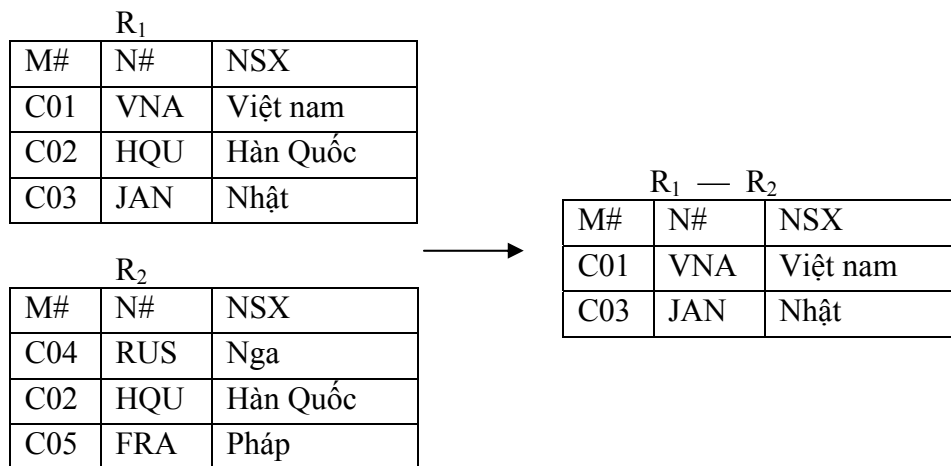
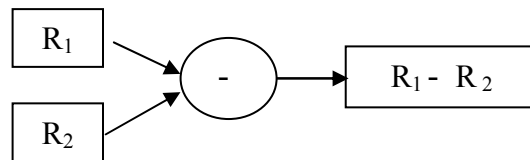
Ký hiệu: $R_1 - R_2 = \{t \mid t \in R_1 \text{ và } t \notin R_2\}$

MINUS (R_1, R_2)

REMOVE (R_1, R_2)

$R_1 - R_2$

Ví dụ: MINUS (R_1, R_2)



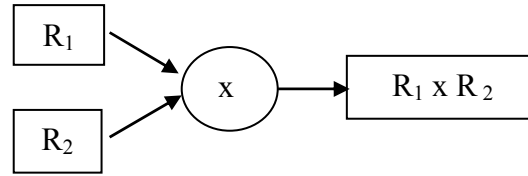
Hình 5.3 Mô tả phép trừ đại số quan hệ

5.2.5 Tích Đề Các - (Cartesian Product)

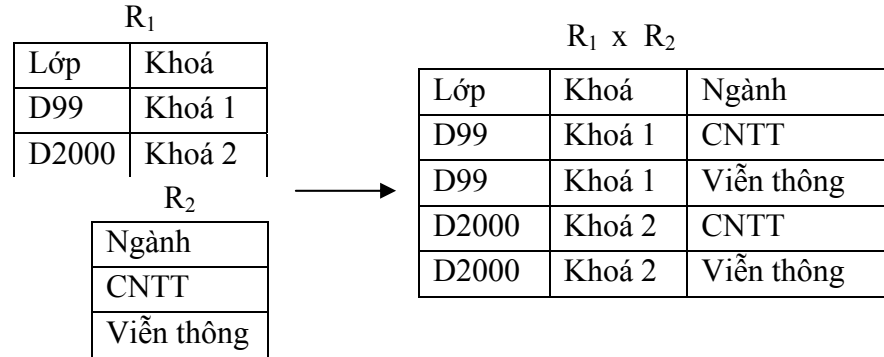
Cho 2 quan hệ $R_1(\Omega)$ và $R_2(\Sigma)$, đặt $\Omega \cap \Sigma = \emptyset$. Tích Đề các của 2 quan hệ $R_1(\Omega)$ và $R_2(\Sigma)$ là một quan hệ P với tập các thuộc tính là $\Omega \Sigma = \Omega \cup \Sigma$ và các bộ của nó thành phần đầu thuộc R_1 và thành phần còn lại thuộc R_2 .

Ký hiệu: $R_1 \times R_2 = P(\Omega \Sigma) := \{t \mid t[\Omega] \in R_1 \ \& \ t[\Sigma] \in R_2\}$

TIMES (R_1, R_2)
 PRODUCT $R_1 \times R_2$
 (R_1, R_2)



Ví dụ PRODUCT $R_1 \times R_2$



Hình 5.4 Mô tả phép tích đại số quan hệ

5.2.6 Phép chọn - Selection

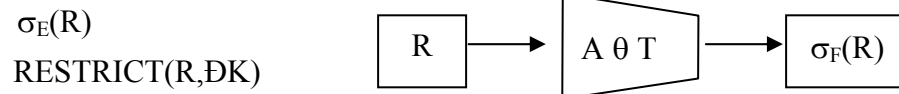
Phép chọn- SELECT tạo ra một quan hệ kết quả. Bậc trùng với bậc quan hệ nguồn và các bộ của nó là các bộ của quan hệ nguồn thỏa một biểu thức logic nào đó.

Định nghĩa: R là một quan hệ trên tập các thuộc tính Ω , cho E là một biểu thức logic phát biểu trên tập thuộc tính Ω . Nói rằng bộ t thỏa mãn biểu thức logic E, ký hiệu t[E] nếu sau khi thay mọi giá trị của thuộc tính A trong E bởi giá trị t[A] tương ứng thì nhận được một mệnh đề đúng, tức là t[A] nhận giá trị “True”.

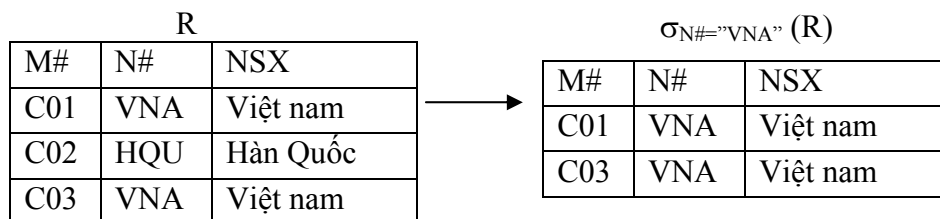
Biểu thức logic được biểu diễn gồm nhiều toán hạng và giữa các toán hạng là các phép so sánh đơn giản, nhận giá trị "đúng" hoặc "sai".

- Toán hạng: là hằng hoặc các thuộc tính quan hệ..
- Các phép so sánh $\theta = \{<, >, \neq, =, \leq, \geq\}$
- Các phép logic: \vee (OR/ hoặc), \wedge (AND /và) và \neg (NOT/phủ định) .

Ký hiệu: $\sigma_E(R) = \{t \mid t \in R \ \& \ t[E] = \text{“True”}\}$



Ví dụ RESTRICT (R, NSX="Việt nam")



Hình 5.5 Mô tả phép chọn đại số quan hệ

5.2.7 Phép chiếu - PROJECT

Từ quan hệ nguồn R bỏ đi một số thuộc tính và/ hoặc sắp xếp lại các thuộc tính còn lại. Các bộ của nó không trùng nhau. Có thể thao tác như sau:

- Loại khỏi quan hệ R tất cả các cột ứng với các thuộc tính trên tập $\Omega \setminus X$.
- Bỏ đi những bộ giống nhau và chỉ giữ lại những bộ đại diện.

Gọi $\pi_{A_1, A_2, \dots, A_k}(R)$ là quan hệ chiếu, khi đó:

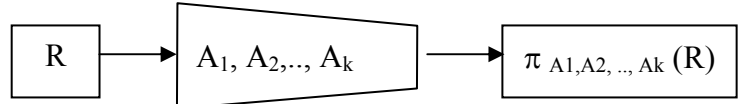
$$\pi_{A_1, A_2, \dots, A_k}(R) = \{t[X] \mid t \in R \text{ và } X = (A_1, A_2, \dots, A_k)\}$$

Ký hiệu :

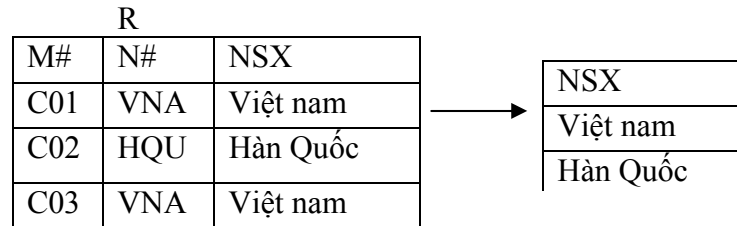
$$\pi_{A_1, A_2, \dots, A_k}(R)$$

$$R[A_1, A_2, \dots, A_k]$$

$$\text{PROJECT}(R, [A_1, A_2, \dots, A_k])$$



Ví dụ PROJECT (R, NSX)



Hình 5.6 Mô tả phép chiếu đại số quan hệ

5.3 Các phép toán đại số bổ sung

5.3.1 Phép chia - DIVISION

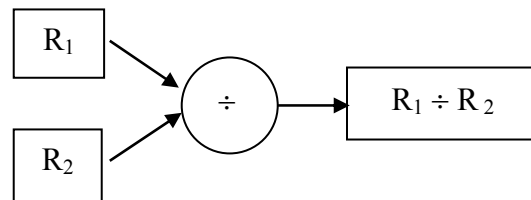
Định nghĩa: Cho 2 quan hệ $R_1(\Omega)$ và $R_2(\Sigma)$, $\Sigma \subset \Omega$, $\Sigma \neq \emptyset$. Thương của phép chia quan hệ $R_1(\Omega)$ cho $R_2(\Sigma)$ là một quan hệ trên tập các thuộc tính $X = \Omega \setminus \Sigma$ được xác định như sau:

$$R_1 \div R_2 := \{t \mid \forall s \in R_2, (t, s) \in R_1\}$$

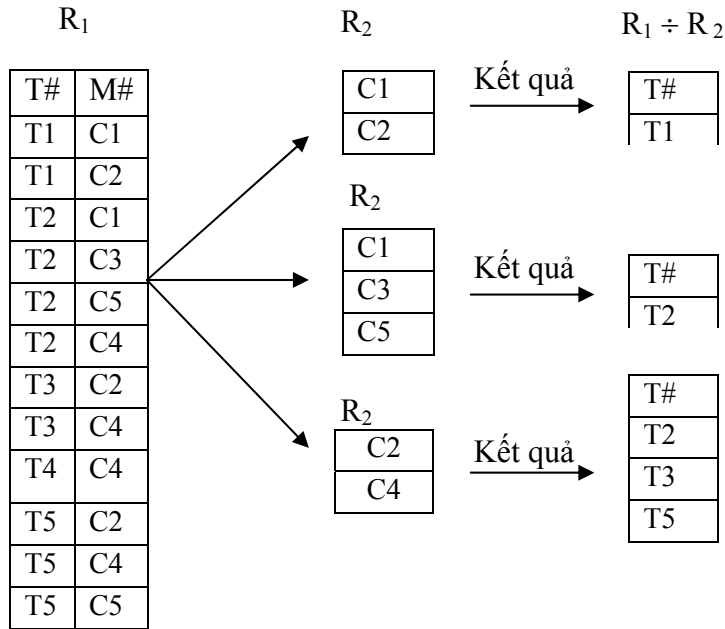
Có thể biểu diễn phép chia bằng các phép toán cơ bản như sau:

- Thực hiện phép chiếu $T = \pi_X(R_1)$
- $V = \pi_X((T \times R_2) - R_1)$.
- $R_1 \div R_2 = T - V = \pi_X(R_1) - \pi_X((\pi_X(R_1) \times R_2) - R_1)$

Ký hiệu: $R_1 \div R_2$
 DIVIDE (R₁, R₂)
 R_1 / R_2



Ví dụ DIVIDE (R₁, R₂)



Hình 5.8 Mô tả phép chia đại số quan hệ

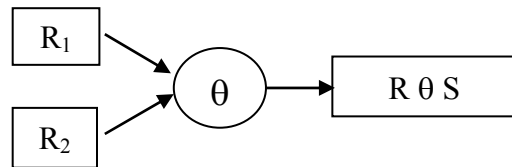
5.3.2 Phép kết nối – JOIN

a) Kết nối bằng nhau và không bằng nhau:

Định nghĩa: Phép kết nối - θ của quan hệ $R(\Omega)$ và $S(\Sigma)$ theo thuộc tính thứ i và j , trong đó $\theta = \{<, >, \neq, \leq, \geq\}$, là quan hệ trên các thuộc tính của $\Omega \Sigma = \Omega \cup \Sigma$ và các bộ là những bộ của tích Descartes của R và S sao cho các thành phần trong thứ i của R quan hệ θ với thành phần thứ j của quan hệ S . Nếu θ là quan hệ “=” gọi là kết nối bằng, ngược lại gọi là kết nối không bằng. Ký hiệu là $R \theta S$.

- Tích Descartes $T = R \times S$.
- $R \theta S$ Thực hiện phép chọn $\sigma_{A_i \theta B_j}(T)$
- $R \theta S = \sigma_{A_i \theta B_j}(R \times S)$

Ký hiệu: $R \theta S$
JOIN (R, S, ĐK)



Ví dụ JOIN (R, S, B < D)

A	B	C
1	2	3
4	5	6
7	8	9

Quan hệ R

D	E
3	1
6	2

Quan hệ S

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2

$R \theta S : B < D$

Hình 5.8 Mô tả phép kết nối đại số quan hệ

b) *Kết nối tự nhiên:*

Định nghĩa: Cho quan hệ $R(\Omega)$ và $S(\Sigma)$. Khi đó kết nối tự nhiên của quan hệ R và S là một quan hệ P trên các thuộc tính $\Omega \Sigma = \Omega \cup \Sigma$, và

$$R \bowtie S = P(\Omega \Sigma) := \{t \in R \theta S \mid t[\Omega \cap \Sigma]_R = t[\Omega \cap \Sigma]_S\}.$$

- Tình Descartes $T = R \times S$.
- Với mọi $t \in T$, $t[\Omega \cap \Sigma] \in R = t[\Omega \cap \Sigma] \in S$
- $R \bowtie S =$ Thực hiện phép chọn $\sigma_{A_i \theta B_j}(T)$
- $R \bowtie S = \pi_X(\sigma_{R.X=S.X}(R \times S))$

Ví dụ

A	B	C
a	b	c
d	b	c
b	b	k
c	a	d

Quan hệ R

B	C	D
b	c	d
b	c	y
a	d	b

Quan hệ S

A	B	C	D
a	b	c	a
a	b	c	y
d	b	c	d
d	b	c	y
c	a	d	b

$R \bowtie S$

Hình 5.9 Phép kết nối tự nhiên

c) *Nửa kết nối:*

Định nghĩa: Cho quan hệ $R(\Omega)$ và $S(\Sigma)$. Khi đó nửa kết nối của quan hệ R và S là phép chiếu trên các thuộc tính Ω trong phép kết nối tự nhiên của R và S :

$$R \ltimes S = \pi_{\Omega}(R \times S)$$

Ví dụ

A	B	C
a	b	c
d	b	c
b	b	k
c	a	d

Quan hệ R

B	C	D
b	c	d
b	c	y
a	d	b

Quan hệ S

A	B	C	D
a	b	c	a
a	b	c	y
d	b	c	d
d	b	c	y
c	a	d	b

$R \ltimes S$

A	B	C
a	b	c
d	b	c
c	a	d

$R \times S$

Hình 5.10 Phép nửa kết nối

5.4 Các tính chất của đại số quan hệ

Kết quả các phép toán cũng là một quan hệ. Tuy nhiên các toán hạng là các quan hệ giao hoán với nhau, nghĩa là các bộ thay đổi thứ tự trong bảng, điều này không quan trọng với cách biểu diễn dữ liệu. Nó vẫn đảm bảo tính độc lập và tính toàn vẹn dữ liệu.

5.4.1 Tính chất giao hoán

Các phép hợp, giao và kết nối trong đại số quan hệ là các phép toán có tính chất giao hoán. Tức là với mọi quan hệ R_1, R_2 thì

$$R_1 \cup R_2 = R_2 \cup R_1$$

$$R_1 \cap R_2 = R_2 \cap R_1$$

$$R_1 \mid \times \mid R_2 = R_2 \mid \times \mid R_1$$

5.4.2 Tính chất kết hợp

Các phép hợp, giao, kết nối và tích Descartes trong đại số quan hệ là các phép toán có tính chất kết hợp. Tức là với mọi quan hệ R_1, R_2, R_3 thì:

$$R_1 \cup R_2) \cup R_3 = R_2 \cup (R_1 \cup R_3)$$

$$(R_1 \cap R_2) \cap R_3 = R_2 \cap (R_1 \cap R_3)$$

$$R_1 \mid \times \mid (R_2 \mid \times \mid R_3) = (R_2 \mid \times \mid R_1) \mid \times \mid R_3$$

$$(R_1 \times R_2) \times R_3 = R_2 \times (R_1 \times R_3)$$

5.4.3 Tính chất tích lũy đẳng

Các phép hợp, giao và kết nối trong đại số quan hệ là các phép toán có tính lũy đẳng. Tức là với mọi quan hệ R trên tập các thuộc tính Ω :

$$R_1 \cup R_1 = R_1$$

$$R_1 \cap R_1 = R_1$$

$$R_1 \mid \times \mid R_1 = R_1$$

5.4.4 Một số tính chất khác

Cho R là quan hệ bất kỳ trên Ω và S là quan hệ bất kỳ trên Σ . Khi đó:

$$R \cap S = R - (R - S)$$

$$R \div S = R[X] - (R[X] * S - R)[X]$$

$$(R \times S) [X] = R[X] \times S[X]$$

$$X = \Omega \setminus \Sigma$$

5.5 Đại số quan hệ – ngôn ngữ vấn tin

Từ các phép toán của đại số quan hệ có thể xây dựng ngôn ngữ vấn tin trên các quan hệ. Một câu hỏi có thể được biểu diễn bởi biểu thức quan hệ hay một cây phép toán quan hệ.

Khảo sát các quan hệ sau:

- Thông tin về tuyến cáp: R_1 (T#, TC, DAI).
- Thông tin về các loại cáp: R_2 (M#, MC, N#, NSX, GIA).
- Thông tin về các loại cáp trên tuyến cáp: R_3 (T#, M#, SL, NG)

Trong đó:

T#	Mã tuyến cáp.	N#	Mã nước sản xuất cáp.
TC	Tên tuyến cáp.	NSX	Nước sản xuất.
DAI	Độ dài tuyến	GIA	Đơn giá cáp.
M#	Mã cáp.	SL	Số lượng.
MC	Tên cáp	NG	Ngày lắp đặt.

5.5.1 Biểu thức quan hệ

Ví dụ: Cho biết tên các tuyến có độ dài trên 100 Km

- Chọn những bộ có DAI > 100 Km từ quan hệ R_1 , cho kết quả S.

- Chiều trên thuộc tính T# và TC từ quan hệ S.
- $KQ(T\#,TC) = \pi_{T\#,TC}(\sigma_{DAI>100}(R1))$
 $S = \text{RESTRICT}(R1, DAI > 100)$
 $KQ = \text{PRODUCT}(S, [T\#,TC])$

Ví dụ : Thông tin về các tuyến có lắp đặt các loại cáp Việt nam

- Thực hiện phép chọn trên R3 thoả M#="VNA", quan hệ kết quả T1.
 $T1 = \sigma_{M\#="VNA"}(R3)$
 $T1 = \text{RESTRICT}(R3, [M\#="VNA"])$
- Kết nối R1 với T1 trên thuộc tính chung T#, quan hệ kết quả T2.
 $T2 = R1 \mid_{>}. R1.T\#=T1.T\# \mid_{<} T1$
 $T2 = \text{JOIN}(R1, T1, R1.T\#=T1.T\#)$
- Kết nối R2 với T2 trên thuộc tính chung M#, quan hệ kết quả T3.
 $T3 = R1 \times_{R2.M\#=T2.M\#} T1$
 $T3 = \text{JOIN}(R2, T2, R2.M\#=T2.M\#)$
- Chiều T3 trên các thuộc tính T#, TC, DAI, quan hệ kết quả KQ
 $KQ = \pi_{T\#,TC,DAI}(T3)$

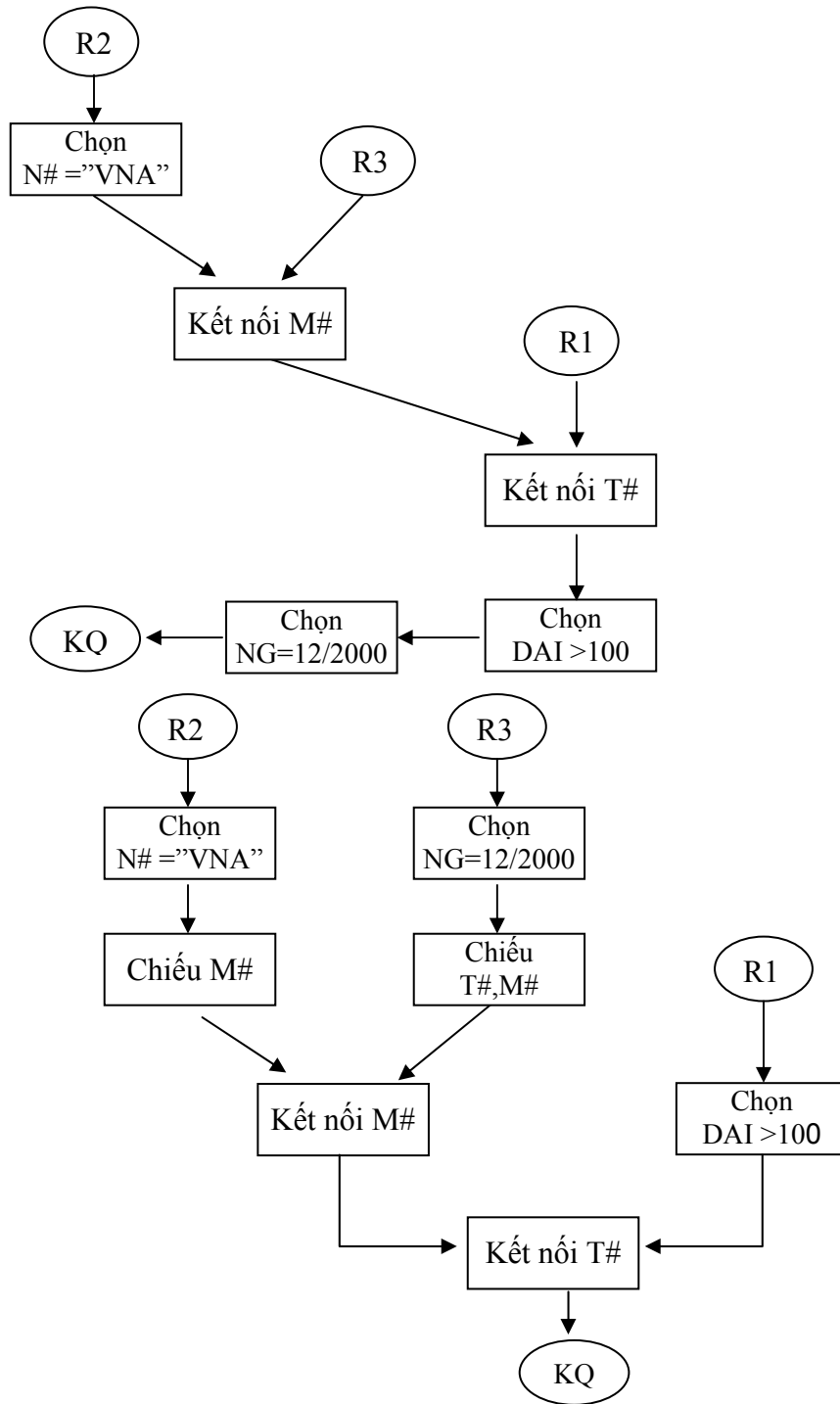
Ví dụ : Thông tin về các tuyến lắp đặt tất cả các loại cáp.

- Thực hiện phép chiếu R3 trên các thuộc tính T#, M#, quan hệ kết quả T1.
 $T1 = \pi_{T\#,M\#}(R3)$
 $T1 = \text{PROJECT}(R3, [T\#,M\#])$
- Thực hiện phép chiếu R2 trên thuộc tính M#, quan hệ kết quả T2.
 $T2 = \pi_{M\#}(R2)$
 $T2 = \text{PROJECT}(R2, [M\#])$
- Thực hiện phép chia T1 cho T2 trên thuộc tính M#, quan hệ thương T3.
 $T3 = R1 \div R2 = \pi_X(T1) - \pi_X((\pi_X(T1) \times T2) - T1)$
 $T3 = \text{DIVIDE}(T1, T2)$
- Kết nối T3 với R1 trên thuộc tính chung T#, quan hệ kết quả T4.
 $T4 = R1 \times_{R1.T\#=T3.T\#} T3$
 $T4 = \text{JOIN}(R1, T3, [R1.T\#=T3.T\#])$
- Chiều T4 trên các thuộc tính T#, TC, DAI, quan hệ kết quả KQ
 $KQ = \pi_{T\#,TC,DAI}(T3)$
 $KQ = \text{PROJECT}(R3, [T\#,TC,DAI])$

5.5.2 Cây đại số quan hệ

Câu hỏi truy vấn có thể biểu diễn bằng các phép toán đại số quan hệ, có thể biểu diễn bằng cây đại số quan hệ.

Ví dụ Các tuyến cáp dài trên 100 km có lắp đặt cáp “Việt Nam” trong tháng 12 năm 2000.



Hình 5.11 Biểu thức đại số biểu diễn bằng cây đại số quan hệ

5.5.3 Cài đặt đại số quan hệ

1. Quy ước:

- *For each t in R [with B] do A endfor* : Lệnh này có nghĩa là thực hiện phép toán A đối với những bộ t thuộc trong quan hệ R thỏa mãn điều kiện B. Phần viết trong [] là tùy chọn.

- *If B then A [else C] endif* : Nếu điều kiện B thoả mãn thực hiện phép toán A, nếu không thực hiện phép toán C.
- *R create (P, X)* : Tạo một quan hệ rỗng P, với tập thuộc tính X cho trước. Lưu ý rằng khi nói đến thuộc tính A là phải hiểu rằng A đã có một miền biến thiên cụ thể nào đó, tức là A đã thuộc một kiểu cho trước.
- *Attrib (R)* : Hàm này trả về tập các thuộc tính quan hệ R.
- *#R* : Hàm này trả về lực lượng (số bộ) của quan hệ R .
- *Add t to P* : Nếu bộ t chưa có trong quan hệ P thì nạp t vào P. Phép toán này tương đương với câu lệnh $P = P \cup \{t\}$.
- $t.X = t[X]$: Phép chiếu X trên bộ t hay sự hạn chế của bộ t trên tập X.
- $t = \langle u,v \rangle$: Tạo một bộ mới t từ các bộ u,v đã có.
- *t in R* : Cho giá trị đúng (“True”) nếu t đã có trong quan hệ R, ngược lại sẽ cho giá trị sai (“False”).

2) *Thuật toán hợp*

Algorithm Union
Format $P = R \cup S$
Input Quan hệ R, S khả hợp
Output $P = \{ t \mid t \in R \text{ OR } t \in S \}$
Begin
 R create(P, Attrib (R)) ;
 For each t in R do
 Add t to P
 Endfor ;
 For each v in S do
 If not (v in P) then
 Add v to P ;
 Endif ;
 Endfor
 Return (P)
End Union.

3) *Thuật toán giao*

Algorithm Intersection
Format $P = R \cap S$
Input Quan hệ R, S khả hợp
Output $P = \{ t \mid t \in R \text{ AND } t \in S \}$
Begin
 R create (P, attrib (R)) ;
 For each t in R do

```
        If  $t$  in S then
            Add  $t$  to P
        Endif
    Endfor;
    Return (P)
End intersection ;
```

4) *Thuật toán trừ*

```
Algorithm          Substraction
Format            P = R - S
Input             Quan hệ R, S khả hợp
Output            P = {  $t \mid t \in R \text{ AND } t \notin S$  }
Begin
    R create (P, attrib (R)) ;
    For each  $t$  in R do
        If not ( $t$  in S) then
            Add  $t$  to P;
        Endif
    Endfor;
    Return (P)
End Subs
```

5) *Thuật toán chọn*

```
Algorithm          Selectoin
Format            P = R(E)
Input             Quan hệ R
                  Biểu thức đại số quan hệ
Output            P = {  $t \mid t \in R \ \& \ t(E) = \text{'True'}$  }
Begin
    R create (P, attrib (R)) ;
    For each  $t$  in R with  $t(E)$  do
        Add  $t$  to P;
    Endfor;
    Return (P)
End Selection :
```

6) *Thuật toán chiếu*

```
Algorithm          Projection
Format            P = R[X]
Input             Tập các thuộc tính X
Output            P = {  $t \mid t.X \ \& \ t \in R$  }
```

```

Begin
  R create (P, X) ;
  For each t in R do
    Add t[X] to P;
  Endfor;
  Return (P)
End Projection

```

7) *Thuật toán kết nối tự nhiên*

```

Algorithm      Join
Format        P = R |><|. S
Input         Quan hệ R, S
Output        P = {<u,v/x> | u ∈ R, v ∈ R
              u.X = v.X, X = Attrb(R) ∩ Attrb(S)}

```

```

Begin
  R create (P, attrib (R) ∪ attrib(S - R) ;
  X = attrib (R) ∩ attrib(S)
  For each u in R do
    For each v in S with u.X = v.X do
      Add <u,v/X> to P;
    Endfor
  Endfor
  Return (P)
Endjoin :

```

8) *Thuật toán phép chia*

Cho 2 quan hệ R và S, thuật toán chia 2 quan hệ $R \div S$ như sau:
 Đặt $M = \text{attrib}(R)$, $N = \text{attrib}(S)$, $X = M - N$. Theo định nghĩa, P sẽ là một quan hệ kết quả trên tập thuộc tính X:

$$P = R \div S := \{ u.X \mid u \in R \ \& \ \forall v \in S \ \text{thì} \ \langle u.X, v \rangle \in R \}$$

$$\Leftrightarrow \{ t \mid \forall s \in S, (t, s) \in R \}$$

```

Algorithm      Division
Format        P = R ÷ S
Input         Quan hệ R, quan hệ S
Output        P = R1 ÷ R2 = πX(R) - πX((πX(R) x S) - R)
Begin

```

```

  X = Attrib (R) – Attrib (S);
  Begin /* Thực hiện phép chiếu πX(R) /
    R create (T1, X) ;
    For each t in R do

```

```
        Add t[X] to T1;
    Endfor;
    Return (T1)
End Projection
Begin /* Thực hiện tích đề các  $\pi_X(R) \times S$  /
    R create (T2, Attrib(R)) ;
    For each t in T1 do
        For each v in S do
            Add <t,v> to T2;
        Endfor;
    Endfor;
    Return (T2)
End Descartes
Begin /* thực hiện phép trừ  $(\pi_X(R) \times S) - R$  /
    R create (T3, Attrib(R)) ;
    For each t in T2 do
        If not (t in R) then
            Add t to T3;
        Endif
    Endfor;
    Return (T3)
End Subs
Begin /* Thực hiện phép chiếu  $\pi_X((\pi_X(R) \times S) - R)$  /
    R create (T4, X) ;
    For each t in T3 do
        Add t[X] to T4;
    Endfor;
    Return (T4)
End Projection
Begin /* thực hiện phép trừ  $\pi_X(R) - \pi_X((\pi_X(R) \times S) - R)$  /
    R create (P, X) ;
    For each t in T1 do
        If not (t in T4) then
            Add t to P;
        Endif
    Endfor;
    Return (P)
End Subs
End Devision.
```

5.6 Ngôn ngữ vấn tin có cấu trúc SQL

5.6.1 Giới thiệu

SQL (Structured Query Language) được xây dựng và phát triển trên cơ sở ngôn ngữ con dữ liệu SEQUEL (Structured English Query Language) và QUERY BY EXAMPLE, là ngôn ngữ hỏi-đáp bằng thí dụ. SQL là một ngôn ngữ được IBM phát triển ở San Jose, với mục đích đề sử dụng trong các hệ CSDL thử nghiệm System R. Ngày nay SQL đã được thương mại hoá và được sử dụng trong rất nhiều hệ thống CSDL trong các lĩnh vực khác nhau.

Ngôn ngữ vấn tin SQL không chỉ là một ngôn ngữ con thao tác dữ liệu độc lập, mà còn là một ngôn ngữ lập trình được nhúng trong các ngôn ngữ lập trình chủ khác. Nó là ngôn ngữ con dữ liệu kiểu quan hệ, được xây dựng trên cơ sở đại số quan hệ và các phép toán quan hệ. Vì vậy SQL có nhiều giao diện khác nhau, được thực hiện qua các thực đơn, cửa sổ, lưới... hoặc có thể lập trình trên các công cụ của nó. SQL trở thành một công cụ truy xuất dữ liệu chuẩn mực và tương đối mạnh, được ứng dụng rộng rãi trong nhiều lĩnh vực tổ chức các hệ thông tin quản lý tác nghiệp.

Cũng như các ngôn ngữ thao tác dữ liệu khác, ngôn ngữ SQL gồm 2 thành phần: ngôn ngữ định nghĩa dữ liệu và ngôn ngữ thao tác dữ liệu. Và có thể sử dụng theo các phương thức tương tác và nhúng vào một ngôn ngữ chủ nào đó.

1. Ngôn ngữ SQL gồm 2 thành phần:

SQL ——— Ngôn ngữ định nghĩa dữ liệu DDL (Data definition language)
 └─── Ngôn ngữ thao tác dữ liệu DML (Data manipulation language)

SQL-DDL ——— Tạo, sửa và xóa cấu trúc quan hệ.
 └─── Bảo mật và quyền truy nhập.

SQL-DML ——— Truy vấn: Các chức năng tìm kiếm.
 └─── Cập nhật: Chức năng thêm, sửa, xoá dữ liệu.

2. Sử dụng SQL trong m.....

SQL ——— SQL trong chế độ tương tác
 └─── SQL trong chế độ nhúng trong ngôn ngữ chủ.

5.6.2 Câu lệnh SELECT

Dạng thông dụng nhất của vấn tin trong SQL là câu lệnh chọn, có cú pháp như sau:

```
SELECT   R1.A1, R2.A2,..., Rn.An
FROM    R1, R2,..., Rn
WHERE   E
GROUP   BY R1.Aj HAVING <điều kiện>
ORDER   BY
```

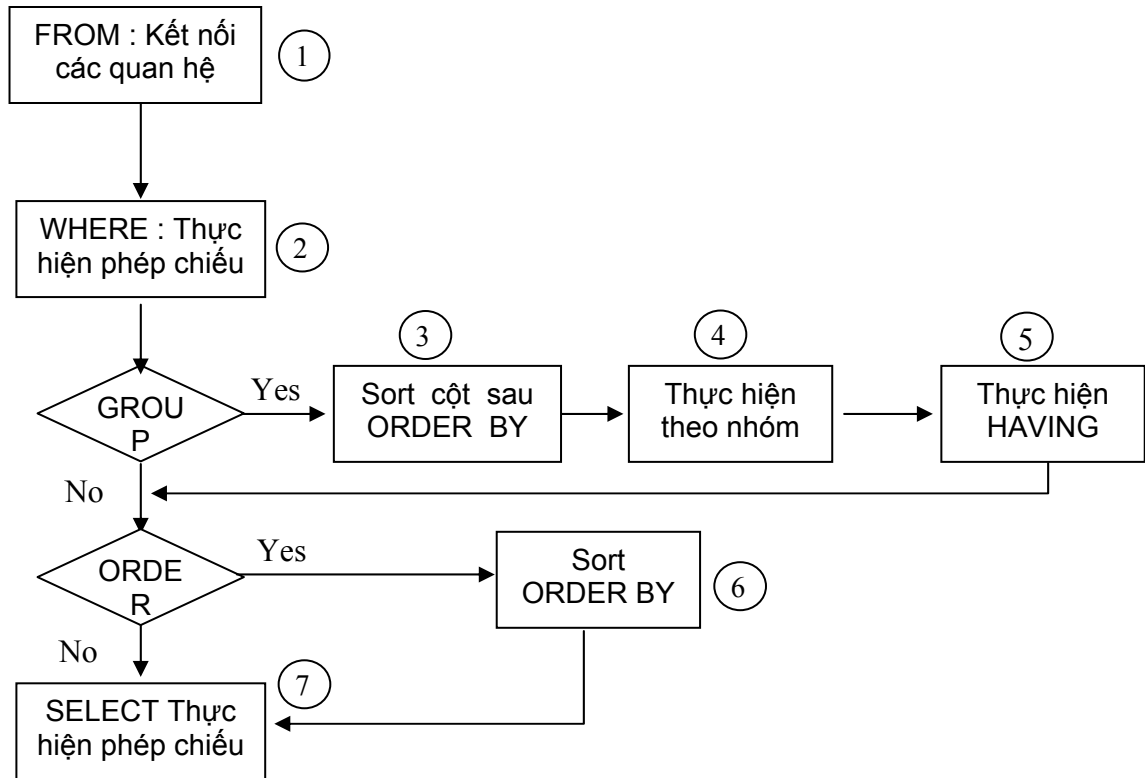
Trong đó R₁, R₂,..., R_n là danh sách các tên quan hệ nguồn và R₁.A₁, R₂.A₂,..., R_n.A_n là danh sách các thành phần được kết xuất, R.A nghĩa là thuộc tính A của quan hệ R. Nếu

trong danh sách sau từ khóa FROM chỉ có một quan hệ có thuộc tính A thì có thể thay A bởi $R.A$ trong danh sách sau SELECT. E là một biểu thức chứa các phép logic AND, OR và NOT và các phép toán so sánh $=, <., >, \neq, \geq, \leq, \dots$

Câu vấn tin SQL ở trên có thể biểu diễn bằng biểu thức đại số quan hệ như sau :

$$\pi_{R_1.A_1, A_2, \dots, R_n.A_n} (\sigma_E (R_1 \times R_2 \times \dots \times R_n)).$$

Nghĩa là lấy tích các quan hệ trong mệnh đề FROM, chọn theo biểu thức E sau WHERE và cuối cùng chiếu trên các thuộc tính sau mệnh đề SELECT.



Hình 5.12 Các bước thực hiện câu lệnh SELECT

1. *Bước thứ nhất:* Thực hiện các phép tích Đề các, hoặc phép kết nối tự nhiên các quan hệ sau FROM: R_1, R_2, \dots, R_n . Nếu lực lượng của $|R_i| = l_i$ và bậc là $n_i, i = 1 \div n$. Khi đó, kết quả của tích sẽ là một quan hệ có $n_1 + n_2 + \dots + n_n$ cột và có $l_1 \times l_2 \times \dots \times l_n$ hàng.
2. *Bước thứ hai:* Thực hiện phép toán chọn trên quan hệ sau bước 1, thỏa biểu thức đại số E sau mệnh đề WHERE.
3. *Bước thứ ba:* Nếu trong câu lệnh có chứa GROUP BY, nghĩa là hệ thống phải phân loại (sort) theo các cột được mô tả sau GROUP BY. Thứ tự phân loại theo chiều từ phải qua trái. Hay nói cách khác, hệ thống sẽ phân hoạch quan hệ kết quả sau bước thứ 2 thành nhiều nhóm tách biệt nhau. Ví dụ GROUP BY ĐC, HT, nghĩa là sắp xếp theo vắn ĐC (địa chỉ), trong ĐC sắp xếp theo vắn HT (họ tên).

4. *Bước thứ tư:* Thực hiện các phép toán gộp nhóm. Kết quả là một quan hệ mới được thiết lập, các hàng là kết quả của các phép gộp nhóm.
5. *Bước thứ năm:* Mệnh đề HAVING xuất hiện nếu GROUP BY đã xuất hiện. Sẽ thực hiện phép chọn trên quan hệ sau bước 4.
6. *Bước sáu:* Nếu có mệnh đề ORDER BY, thực hiện phép phân loại.
7. *Cuối cùng,* bước 7, hệ thống thực hiện phép chiếu sau SELECT.

Ví dụ: “Liệt kê những khách hàng có số dư âm”, được biểu diễn trong SQL như sau :

```
SELECT    TEN_K
FROM      KHAC_HANG.
WHERE     TAI_KHOAN < 0 ;
Kết quả:  TEN_K
           Nga
           Hồng
```

Trong câu lệnh trên chỉ có một quan hệ duy nhất trong mệnh đề FROM, không thể có sự nhầm lẫn các thuộc tính. Vì vậy không cần phải đặt tên quan hệ trước các thuộc tính. Quan hệ kết quả chỉ có một cột là TEN_K. Nếu muốn có một tiêu đề khác cho cột này, có thể đưa ra một bí danh (alias) cho TEN_K bằng cách viết bí danh ngay sau TEN_K có một dấu cách (Blank)

```
SELECT    TEN_K HO_TEN
FROM      KHACH_HANG
WHERE     TAI_KHOAN < 0 ;
Kết quả:  HO_TEN
           Nga
           Hồng
```

Nếu in ra mọi thông tin về những khách hàng có số dư âm, có thể viết như sau:

```
SELECT    TEN_K, DIA_CHI, TAI_KHOAN
FROM      KHACH_HANG
WHERE     TAI_KHOAN < 0 ;
```

hoặc chỉ cần viết :

```
SELECT    *
FROM      KHACH_HANG
WHERE     TAI_KHOAN < 0 ;
```

Ký tự * là cách viết của SQL yêu cầu in tất cả các thuộc tính của quan hệ R. Trong thí dụ này, KHACH_HANG là một quan hệ duy nhất trong mệnh đề FROM, không cần phải xác định các thuộc tính quan hệ đó.

Ví dụ Liệt kê thông tin tất cả nhà cung cấp các mặt hàng mà Nga đặt mua.

Thực hiện kết nối tự nhiên của 3 quan hệ HOA_DON, MH_SL và CUNG_CAP bằng cách dùng những đẳng thức trong mệnh đề WHERE để định nghĩa phép nối.

```
SELECT    DISTINCT TEN_CC
```

```
FROM HOA_DON, CUNG_CAP, MH_SL
WHERE TEN_K = "Nga"
      AND HOA_DON.SHD# = MH_SL.SHD#
      AND CUNG_CAP.MAT_HANG= NG_CAP.MAT_HANG
```

Cũng như hầu hết các ngôn ngữ vấn tin khác, SQL không tự động loại đi các bộ trùng lặp. Vì vậy trong một số trường hợp, phụ thuộc yêu cầu câu hỏi, trong câu vấn tin nên sử dụng từ khóa DISTINCT theo sau SELECT.

5.6.3 Biến quan hệ

Trong một số trường hợp, có nhiều câu vấn tin liên quan đến nhiều bộ trong cùng một quan hệ. Để thực hiện được điều này, cần định nghĩa nhiều biến quan hệ ở mệnh đề FROM và dùng những biến quan hệ này làm bí danh của quan hệ.

Ví dụ: Liệt kê tên và địa chỉ của những khách hàng có số dư nhỏ của Nga.

```
SELECT c1.TEN_K, c1.DIA_CHI
FROM KHACH_HANG c1, KHACH_HANG c2
WHERE c1.TAI_KHOAN < c2.TAI_KHOAN
      AND c2.TEN_K = "Nga";
```

Sau mệnh đề FROM c1 và c2 đều là bí danh của KHACH_HANG, chúng trở thành các bộ chạy trên KHACH_HANG.

5.6.4 Toán tử LIKE

Trong mệnh đề WHERE, có thể bổ sung toán tử LIKE để mô tả điều kiện là một giá trị nào đó phải khớp với một mẫu (pattern).

- Ký hiệu % : Nghĩa là một chuỗi bất kỳ có độ dài tùy ý, chuỗi bắt đầu từ vị trí của %.
- Ký tự gạch thấp _ đại diện cho một ký tự bất kỳ thay thế ký tự “_” .

Ví dụ In ra những mặt hàng có ký tự đầu là “E” và giá từ khoảng 1.00 đến 1.99

```
SELECT ITEM, PRICE
FROM SUPPLIES
WHERE ITEM LIKE "E%" AND PRICE LIKE "1.__";
```

5.6.5 Phép toán tập hợp trong mệnh đề WHERE

Trong phần này tiếp tục nghiên cứu các khả năng rộng hơn so với phép toán chọn trong đại số quan hệ. Trong mệnh đề WHERE có thể sử dụng các phép số học để so sánh, vì thế có thể viết những điều kiện này như sau :

```
WHERE A > B + C -10
```

Một khả năng mở rộng của SQL cho phép sử dụng các tập làm *toán hạng* được định nghĩa bằng câu lệnh SELECT-FROM -WHERE đầy đủ, được lồng trong mệnh đề WHERE và được gọi là *câu vấn tin con (subquery)*.

- Phép toán IN biểu diễn các khái niệm thuộc tập. Ngược lại NOT IN biểu diễn khái niệm không thuộc tập .
- ANY thường sử dụng cho các khái niệm tồn tại. Nếu S là một biểu thức biểu diễn cho một tập, thì điều kiện là $A \theta ANY S$ sẽ tương đương với biểu thức logic $(\exists X)$

$(X \in S \wedge A \theta X)$. Điều này có nghĩa là tồn tại thuộc tính A thuộc tập X sao cho A có quan hệ với X bởi các toán tử so sánh θ .

- Tương tự ALL là với mọi: $A \theta \text{ALL } S \Leftrightarrow (\forall X) (X \in S \text{ suy ra } A \theta X)$. Điều này có nghĩa là với mọi thuộc tính A thuộc tập X thì A có quan hệ với X bởi các toán tử so sánh θ .
- BETWEEN... AND: từ..... đến

Ví dụ In tên của tất cả nhà cung cấp có ít nhất một mặt hàng mà “Nga” đặt.

1. Xác định tập S_1 các đơn đặt hàng của “Nga” từ quan hệ HOA_DON
2. Xác định tập S_2 các mặt hàng trong tập đơn đặt hàng S_1 từ quan hệ MH_SL.
3. Xác định tập S_3 là tên các nhà cung cấp các mặt hàng trong tập S_2 bằng cách sử dụng quan hệ CUNG_CAP.

```
(1) SELECT   TEN_CC
(2) FROM    CUNG_CAP
(3) WHERE   MAT_HANG IN
(4)           (SELECT  MAT_HANG
(5)           FROM    MH_SL
(6)           WHERE   O# IN
(7)           (SELECT  O#
(8)           FROM    HOA_DON
(9)           WHERE   TEN_K = "Nga"))
```

Như vậy:

- Tập S_1 được xác định bởi các câu lệnh từ (7)-(9).
- Tập S_2 được xác định bởi các câu lệnh từ (4)-(6).
- Tập S_3 được xác định bởi các câu lệnh từ (1)-(3).

Ví dụ Liệt kê các mặt hàng có giá lớn hơn mọi giá trong SUPPLIES.

```
SELECT  MAT_HANG, GIA
FROM    CUNG_CAP
WHERE   GIA  $\geq$  ALL
           SELECT  GIA
           FROM    CUNG_CAP)
```

Ví dụ Liệt kê các mặt hàng có giá lớn hơn giá của một mặt hàng của Nga đặt.

```
SELECT  MAT_HANG, GIA
FROM    CUNG_CAP
WHERE   GIA  $\geq$  ANY
           (SELECT  GIA
           FROM    CUNG_CAP
           WHERE   MAT_HANG IN
           (SELECT  MAT_HANG
```

```
FROM CUNG_CAP
WHERE O# IN
(SELECT O#
FROM HOA_DON
WHERE TEN_K ='Nga'))
```

Ví dụ Giả sử chắc chắn rằng “Hoa” chỉ có một đơn đặt hàng, khi đó có thể lấy được mã số đơn đặt hàng đó qua một câu vấn tin con và dùng nó để chọn tất cả các mặt hàng được đặt bởi “Hoa” từ quan hệ MH_SL. Trong trường hợp không biết chắc chắn “Hoa” đặt một đơn đặt hàng, mà có thể từ hai đơn đặt hàng trở lên thì câu vấn tin được trình bày dưới đây sẽ trả lời kết quả sai:

In tên các mặt hàng mà “Hoa” đặt mua (trường hợp chỉ có một đơn đặt).

```
SELECT TEN_K
FROM MH_SL
WHERE O# =
(SELECT SHD#
FROM HOA_DON
WHERE TEN_K = “Hoa”);
```

5.6.6 Các phép toán gộp nhóm

SQL cung cấp năm phép toán gộp nhóm thường dùng AVG, COUNT, SUM, MIN, MAX và các phép toán STDDEV và VARLENCE để tính độ lệch chuẩn và phương sai của một danh sách các số. Một câu lệnh SELECT – FROM – WHERE có thể in ra kết quả khi áp dụng một hay nhiều các phép toán gộp cho các thuộc tính của một quan hệ bằng cách đặt quan hệ này trong mệnh đề FROM và trong mệnh đề SELECT AVG(A), COUNT(A),...trong đó A là một thuộc tính. Nếu trước các thuộc tính A có thêm từ khóa DISTINCT thì sự trùng lặp bị loại bỏ trước khi phép toán được thực hiện.

Ví dụ

- a) *In và tính số dư trung bình trong quan hệ CUSTOMERS..*

```
SELECT AVG(TAI_KHOAN)
FROM KHACH_HANG.
```

- b) *In và tính số nhà cung cấp*

```
SELECT COUNT(DISTINCT TEN_CC) #TEN
FROM CUNG_CAP
#TEN là tên cột mới khi in ra kết quả.
```

- c) *Số các nhà cung cấp “rượu cuốc lủi”*

```
SELECT COUNT(DISTINCT TEN_CC) #TEN_CLUI
FROM CUNG_CAP
WHERE MAT_HANG ='Cuốc lủi';
```

5.6.7 Mệnh đề GROUP BY ... HAVING

Có thể phân hoạch các bộ của một quan hệ thành các nhóm tách biệt nhau và áp dụng các phép toán gộp cho các nhóm. Trong câu lệnh SELECT – FROM – WHERE, mệnh đề GROUP BY nhóm lại bởi một danh sách các thuộc tính của quan hệ cần nhóm và thoả một điều kiện nhóm HAVING:

```
GROUP    BY  A1, A2, ..., Ak  
HAVING   E
```

Phân hoạch quan hệ thành các nhóm sao cho hai bộ cùng trong một nhóm khi và chỉ khi chúng giống nhau ở mọi thuộc tính A₁, A₂, ..., A_k. Đề cho kết quả của câu vấn tin có nghĩa, các thuộc tính A₁, A₂, ..., A_k cũng phải xuất hiện trong mệnh đề SELECT mặc dù chúng có thể có những bí danh để in ra nếu cần.

Ví dụ In ra tất cả các mặt hàng và giá trung bình của từng nhóm

```
SELECT  MAT_HANG, AVG (GIA) AP  
FROM    CUNG_CAP  
GROUP   BY  MAT_HANG
```

Một mệnh đề WHERE có thể theo sau mệnh đề FROM, chỉ cần một tập con của các bộ khi tạo các nhóm. Có thể sắp xếp đề in ra một tập con các nhóm, độc lập với kiểu kiện chọn lọc trong mệnh đề WHERE trước khi xây dựng các nhóm.

Ví dụ In ra tên và giá trung bình các mặt hàng rượu được nhiều nhà cung cấp

```
SELECT  MAT_HANG, AVG (GIA) AP  
FROM    CUNG_CAP  
WHERE   MAT_HANG LIKE “%Rượu%”  
GROUP   BY  MAT_HANG  
HAVING  COUNT(*) > 1 ;
```

5.6.8 Các phép toán lưu trữ

1. Phép chèn:

Chèn thêm các bộ mới vào quan hệ, cấu trúc câu lệnh như sau:

```
INSERT  INTO  R  
VALUES  (a1,a2,...,an)
```

Trong đó R là tên quan hệ và r=(a₁,a₂,...,a_n) là một bộ mới cần chèn.

Ví dụ :

- a) Thêm thông tin “Thủy” cung cấp “Rượu Làng Vân”, nhưng chưa có giá.

```
INSERT  INTO  CUNG_CAP (TEN_CC, MAT_HANG)  
VALUES  (“Thủy”, “Rượu Làng Vân”);
```

- b) Tạo quan hệ NG_CC (TEN_CC, GIA) chứa tên người cung cấp “Rượu cuộc lũ”

```
INSERT  INTO  NG_CC (TEN_CC, GIA)  
SELECT  TEN_CC, GIA
```

```
FROM CUNG_CAP  
WHERE MAT_HANG = "Rượu cuộc lủ" );
```

2. Phép xoá

Cú pháp phép xoá như sau:

```
DELETE FROM R  
WHERE E
```

Trong đó R là tên quan hệ và E là biểu thức đại số. Nghĩa là xoá khỏi quan hệ R tất cả các bộ thoả điều kiện E.

Ví dụ

a) Xoá các đơn đặt hàng về "Rượu cuộc lủ"

```
DELETE FROM HOA_DON  
WHERE O# IN  
( SELECT SHD#  
FROM MH_SL  
WHERE MAT_HANG = "Rượu cuộc lủ" );
```

b) Xoá "Nga" không còn cung cấp "Rượu cuộc lủ" nữa

```
DELETE FROM CUNG_CAP  
WHERE TEN_K = "Nga" AND MAT_HANG = "Rượu cuộc lủ";
```

3. Phép sửa đổi

Cú pháp phép sửa đổi thông tin như sau:

```
UPDATE R  
SET A1=a1, A2=a2, ..., An=an  
WHERE E
```

Trong đó R là tên quan hệ và E là biểu thức đại số. Nghĩa là trong quan hệ R, sửa giá trị các thuộc tính A_i bằng a_i, i=1,2,..., n nếu mỗi bộ thoả điều kiện E.

Ví dụ

a) Hiệu chỉnh giá của "Rượu Hà Nội" do "nhà máy Rượu Hà nội" cung cấp thành 15.000 đồng/chai

```
UPDATE CUNG_CAP  
SET GIA = 15.000  
WHERE TEN_CC = "Nhà máy rượu Hà nội"  
AND MAT_HANG = "Rượu Hà nội";
```

b) Giảm 10% giá "Rượu Hà Nội" do "nhà máy Rượu Hà nội" cung cấp.

```
UPDATE CUNG_CAP  
SET GIA = 0.9 * GIA  
WHERE TEN_CC = "Nhà máy rượu Hà nội"  
AND MAT_HANG = "Rượu Hà nội";
```

5.6.9 Tính đầy đủ của SQL

Phần này xem xét khả năng của ngôn ngữ SQL có thực hiện được các biểu thức đại số quan hệ như các ngôn ngữ thao tác dữ liệu khác hay không. Vì vậy chỉ cần trình bày SQL thực hiện năm phép toán cơ bản của đại số quan hệ.

Giả sử cho các quan hệ $R(A_1, A_2, \dots, A_n)$ và $S(B_1, B_2, \dots, B_m)$. Trong trường hợp lấy hợp, hoặc hiệu của R và S , nếu bậc của R và S không giống nhau, tức là $n \neq m$, có thể đặt lại tên cho các thuộc tính của S , tạo ra một quan hệ mới TAM với các thuộc tính như các thuộc tính của quan hệ R :

```
INSERT INTO TAM
SELECT *
FROM S
```

Không mất tính tổng quát, giả sử $n = m$ và $A_i = B_i, i=1,2,\dots,n$.

- *Phép hợp:* $T = R \cup S$

```
INSERT INTO T
SELECT *
FROM R
INSERT INTO T
SELECT *
FROM S
```

- *Phép trừ:* $T = R - S = (R \cup S) - S$

```
DELETE FROM T
      (SELECT (A1,A2,...,An) IN
      FROM S);
```

- *Phép giao:* $T = R \cap S = R - (R - S) = R - T_2$

```
DELETE FROM T_2
      (SELECT (A1,A2,...,An) IN
      FROM S);
DELETE FROM T
      (SELECT (A1,A2,...,An) IN
      FROM T_2);
```

- *Tích Đề Các:* $T = R \times S$

```
INSERT INTO T
SELECT R.A1,...,R.An, S.B1,...,S.Bn
FROM R,S
```

- *Phép chọn:* $T = \sigma_F(R)$

```
INSERT INTO T
SELECT *
FROM R
```

WHERE E;

- *Phép chiếu* $T = \pi_{A_1, A_2, \dots, A_k}(R)$

INSERT INTO T

SELECT A₁, A₂, ..., A_k

FROM R;

5.6.10 Ngôn ngữ định nghĩa dữ liệu

- *Lệnh tạo, xóa một quan hệ*: Lệnh tạo một quan hệ cho phép mô tả tên bảng và xác định các phần tử của bảng ứng với các cột và các ràng buộc giữa các cột. Cú pháp có dạng tổng quát như sau:

CREATE TABLE <tên bảng> (<Các phần tử của bảng>)

Ví dụ: Tạo cấu trúc quan hệ TUYEN (C#, TC, DAI)

```
CREATE TABLE TUYEN
(C# CHAR(5) NOT NULL
TC CHAR(15),
DAI INT);
```

Ngược với lệnh tạo là lệnh xóa quan hệ: Xóa quan hệ CUNG_CAP ra khỏi CSDL có thể viết: DROP TABLE CUNG_CAP

- *Tạo và xóa chỉ mục* Lệnh tạo chỉ mục: Chỉ mục được sử dụng để làm tăng tốc độ truy xuất dữ liệu. Nếu quan hệ R có chỉ mục trên thuộc tính A thì có thể truy xuất tất cả các bộ có giá a cho trước của thuộc tính A theo thời gian tỷ lệ với số lượng các bộ:

```
CREATE INDEX I
ON R(A);
```

Loại bỏ chỉ mục I ra khỏi quan hệ R nhưng không làm ảnh hưởng đến dữ liệu trong quan hệ R, cú pháp lệnh như sau:

```
DROP INDEX I
```

5.6.11 Khung nhìn View

Trong SQL có nhóm phép toán thực hiện chức năng của ngôn ngữ định nghĩa dữ liệu lược đồ con (subschema DDL), hay là chức năng định nghĩa *khung nhìn* (view) không tồn tại trong bộ nhớ nhưng có thể truy vấn. Cú pháp của lệnh như sau:

```
CREATE VIEW V(A1, A2, ..., An) AS Q;
```

Trong đó V là tên của khung nhìn, A₁, A₂, ..., A_n là các thuộc tính của khung nhìn và Q là câu văn tin của khung nhìn.

Ví dụ

- a) *Tạo khung nhìn gồm các mặt hàng và giá của nhà cung cấp "Nga".*

```
CREATE VIEW NGA_VIEW(MAT_HANG, GIA) AS
SELECT MAT_HANG, GIA
FROM CUNG_CAP R
WHERE TEN_="Nga";
```

b) Tạo khung nhìn khách hàng đặt các đơn hàng .

```
CREATE VIEW DH(TEN_K, NGAY, MAT_HANG,  
                SO_LUONG) AS  
SELECT      TEN_K, NGAY, MAT_HANG, SO_LUONG  
FROM        HOA_DON, MH_SL  
WHERE       HOA_DON.SHD# = MH_SL.SHD# ;
```

Trong câu lệnh trên, biến đổi thuộc tính TEN_K của HOA_DON thành TEN_K trong khung nhìn DH, vì lệnh cho phép xác định các thuộc tính cho khung nhìn một cách tường minh.

Cuối cùng, nếu muốn hủy bỏ khung nhìn, cú pháp như sau:

```
DROP VIEW    V
```

Câu lệnh này không có tác dụng trên CSDL nhưng những ' câu vắn tm trên khung nhìn V không còn được chấp nhận nữa.

5.6.12 Database catalog

Có thể lấy thông tin về CSDL bằng cách sử dụng các câu vắn tin tham chiếu đến các danh mục của cơ sở dữ liệu. Các danh mục có thể được coi như là những quan hệ. Có bốn database catalog (danh mục của CSDL): TABLES, VIEWS, INDEXES, và COLUMNS.

- Một số danh mục của khung nhìn VIEWS:
- VIEWS\$TEXT: Hiển thị chuỗi.
- VEW\$: Ngày tháng tạo khung

.....

Ví dụ

a) Tìm định nghĩa của khung nhìn NGA_VIEW

```
SELECT VIEWS$TEXT  
FROM   VIEWS[NGA_VIEW]
```

Tên của khung nhìn được chỉ ra trong dấu ngoặc vuông [NGA_VIEW]. Kết quả của câu vắn tin trên sẽ là chuỗi:

```
SELECT MAT_HANG, GIA FROM CUNG_CAP WHERE TEN_K ="Nga";
```

b) Xác định ngày tháng tạo khung nhìn NGA_VIEW

```
SELECT VEW$TIME  
FROM   VIEWS[NGA_VIEW]
```

- Để xác định thông tin về các thuộc tính của một quan hệ hoặc một khung nhìn, có thể vắn tin danh mục COLUMNS. Một số các thuộc tính của danh mục COLUMNS:
 - COL\$NAME: tên của thuộc tính.
 - COL\$LD: vị trí của thuộc tính trong danh sách các thuộc tính của quan hệ.
 - COL\$DATATYPE: kiểu dữ liệu của thuộc tính.
 - COL\$LENGTH: số byte hoặc số chữ.số thập phân, tương ứng với kiểu ký tự và kiểu số.
 - COL\$SCALE: số chữ số sau dấu chấm thập phân.

- COL\$NULL: cho biết có được phép sử dụng giá trị null trong cột.

Ví dụ

- a) *Xác định tên, kiểu và vị trí các thuộc tính của khung nhìn NGA_VIEW*

```
SELECT COL$NAME, COL$ID, COL$DATATYPE  
FROM COLUMNS[NGA_VIEW]
```

COL\$NAME	COL\$ID	COL\$DATATYPE
MAT_HANG	1	CHAR
GIA	2	NUMBER

- a) *Xác định ngày tháng tạo ra quan hệ SUPPLIES*

```
SELECT TAB$TIME  
FROM TABLES[SUPPLIES];
```

- Sử dụng danh mục INDEXES có thể vẫn tìm để tìm ra thông tin về những chỉ mục khai báo đối với một quan hệ cho trước. Một số danh mục của INDEXES:
 - IDX\$NAME: tên của chỉ mục.
 - IDX\$COLUMN: thuộc tính được chỉ mục.
 - IDX\$UNIQUE: thuộc tính IDX\$COLUMN có được sử dụng như là một khóa của quan hệ hay không.

Ví dụ Chỉ mục O#_INDEX được tạo như sau

```
SELECT INDEX O#_INDEX  
ON HOA_DON(O#)
```

Khi đó có thể truy vấn chỉ mục O#_INDEX được tạo ra như thế nào:

```
SELECT IDX$NAME, IDX$COLUMN, IDX$UNIQUE  
FROM INDEXES(HOA_DON);
```

IDX\$NAME	IDX\$COLUMN	IDX\$UNIQUE
O#_INDEX	O#	Non Unique

Câu hỏi trắc nghiệm

1. Ngôn ngữ thao tác dữ liệu là
 - A. Một tập các phép toán thao tác trên các hệ cơ sở dữ liệu.
 - B. Biểu diễn dữ liệu một cách thống nhất.
 - C. Sự thống nhất tương ứng trong các phép toán thao tác
 - D. Thông tin được biểu diễn bằng một và chỉ một cách duy nhất
2. Ngôn ngữ định nghĩa dữ liệu - DDL (Data Definition Language).
 - A. Được đặc tả bằng một phần của hệ quản trị cơ sở dữ liệu.
 - B. Được đặc tả bằng một ngôn ngữ dữ liệu
 - C. Được đặc tả bằng một ngôn ngữ, một phần của hệ quản trị cơ sở dữ liệu.
 - D. Được đặc tả bằng cách chương trình ứng dụng

3. Ngôn ngữ thao tác dữ liệu DML (Data Manipulation Language)
 - A. Các toán hạng và kết quả của các phép toán là các quan hệ
 - B. Biểu diễn dữ liệu một cách thống nhất.
 - C. Được đặc tả bằng một ngôn ngữ dữ liệu
 - D. Được đặc tả bằng cách chương trình ứng dụng
 4. Nhóm các phép toán lưu trữ bao gồm
 - A. Chèn thêm , xoá, sửa đổi
 - B. Chèn thêm , xoá,
 - C. Phép chọn SELECT, chèn thêm
 - D. Phép chọn SELECT, phép chiếu PROJECT, phép kết nối JOIN
 5. Nhóm các phép toán tìm kiếm gồm:
 - A. Phép chọn SELECT, phép chiếu PROJECT, phép kết nối JOIN
 - B. Chèn thêm , xoá, sửa đổi
 - C. Chèn thêm , xoá,
 - D. Phép chọn SELECT, chèn thêm
 6. Phép chiếu:
 - A. $\sigma_E(R) = \{t \mid t \in R \ \& \ t[E] = \text{"True"}\}$
 - B. $\pi_{A_1, A_2, \dots, A_k}(R) = \{t[X] \mid t \in R \ \text{và} \ X = (A_1, A_2, \dots, A_k)\}$
 - C. $R_1 \times R_2 = P(\Omega \Sigma) := \{t \mid t[\Omega] \in R_1 \ \& \ t[\Sigma] \in R_2\}$
 - D. $R_1 \cap R_2 = \{t \mid t \in R_1 \ \text{And} \ t \in R_2\}$
 7. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Đại số quan hệ là một trong những ngôn ngữ thao tác dữ liệu, bao gồm các phép toán trên các quan hệ. Đó là các phép toán Tập hợp các phép toán quan hệ tạo nên một cơ chế truy nhập dữ liệu khá linh hoạt và mềm dẻo.
 - A. Hợp, giao, chiếu, chọn và kết nối.
 - B. Chèn thêm , xoá, sửa đổi
 - C. Chọn SELECT, chiếu PROJECT, kết nối JOIN
 - D. Hợp, giao, chiếu và chọn.
 8. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: R1 và R2 là các quan hệ khả hợp trên cùng tập các thuộc tính của Ω . Khi đó hợp của chúng là trên Ω .
 - A. Một quan hệ
 - B. Một phụ thuộc hàm
 - C. Một nhóm các bản ghi
 - D. Là tập con các thuộc tính
 9. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: R1 và R2 là các quan hệ khả hợp trên cùng tập các thuộc tính của Ω . Khi đó giao của chúngtrên Ω ,
 - A. Một quan hệ
 - B. Một phụ thuộc hàm
 - C. Một nhóm các bản ghi
-
-

D. Là tập con các thuộc tính

10. Hãy so khớp thông tin 2 phần dưới đây với nhau:

Phần 1

- 1 Kết quả các phép toán
- 2 Các toán hạng
- 3 Thay đổi thứ tự các bộ trong bảng
- 4 Các phép hợp, giao và kết nối trong đại số quan hệ.....
 - a. 1- D; 2- C; 3 – B; 4 – A .
 - b. 1- C; 2- B; 3 – A; 4 – D
 - c. 1- A; 2- B; 3 – C; 4 – D.
 - d. 1- D; 2- B; 3 – A; 4 – C.

Phần 2

- A vẫn đảm bảo tính độc lập và tính toàn vẹn dữ liệu.
- B. là các quan hệ giao hoán với nhau
- C. là một quan hệ
- D. là các phép toán có tính chất giao hoán

11. Hãy so khớp thông tin 2 phần dưới đây với nhau:

Phần 1

- 1 Các phép hợp, giao, kết nối và tích Đề các trong đại số quan hệ
- 2 Phép chọn- SELECT
- 3 Một câu hỏi có thể được biểu diễn bởi
- 4 Câu hỏi truy vấn có thể biểu diễn bằng các phép toán.....
 - a. 1- D; 2- C; 3 – B; 4 – A .
 - b. 1- C; 2- B; 3 – A; 4 – D
 - c. 1- A; 2- B; 3 – C; 4 – D.
 - d. 1- B; 2- D; 3 – A; 4 – C.

Phần 2

- A biểu thức quan hệ hay một cây phép toán quan hệ
- B. là các phép toán có tính chất kết hợp
- C. đại số quan hệ, có thể biểu diễn bằng cây đại số quan hệ.
- D. tạo ra một quan hệ kết quả

Câu hỏi & bài tập

1. Câu hỏi

1. Trình bày khái niệm ngôn ngữ thao tác dữ liệu.
2. Các phép toán đại số quan hệ.
 - a. Phép hợp các quan hệ – UNION
 - b. Phép giao của các quan hệ – INTERSECT.
 - c. Hiệu của các quan hệ - MINUS.
 - d. Tích Đề Các – (Decalesian Product)
 - e. Phép chọn – Selection.
 - f. Phép chiếu – PROJECT.
3. Các phép toán đại số bổ sung.
 - a. Phép chia - DIVISION.

- b. Phép kết nối – JOIN.
- 4. Các tính chất của đại số quan hệ.
 - a. Tính chất giao hoán, kết hợp.
 - b. Tính chất tích lũy đẳng, một số tính chất khác.
- 5. Đại số quan hệ – ngôn ngữ vấn tin.
- 6. Các thuật toán cài đặt đại số quan hệ.
 - a. Thuật toán hợp , giao.
 - b. Thuật toán trừ.
 - c. Thuật toán chọn, chiếu.
 - d. Thuật toán kết nối tự nhiên.
 - e. Thuật toán phép chia.
- 7. Ngôn ngữ vấn tin SQL.
- 8. Câu lệnh SELECT.
 - a. So sánh mẫu LIKE.
 - b. Phép toán tập hợp trong mệnh đề WHERE.
 - c. Các phép toán gộp nhóm.
 - d. Gộp theo nhóm.
- 9. Các phép toán lưu trữ
 - a. Phép chèn.
 - b. Phép xoá.
 - c. Phép sửa đổi.
- 10. Tính đầy đủ của SQL
- 11. Ngôn ngữ định nghĩa dữ liệu
- 12. Khái niệm ngôn ngữ hướng đối tượng

2. Bài tập

Bài 1: Giả sử trong nghiệp vụ quản lý phát hành báo chí, có các thông tin quản lý tác nghiệp như sau

MK#	: Mã hiệu khách hàng đặt mua báo.	TK	: Họ và tên khách hàng.
DC	: Địa chỉ khách hàng.	SĐT	: Số điện thoại.
B#	: Mã báo khách đặt mua.	TB	: Tên báo (tạp chí).
GIA	: Giá báo.	NG	: Ngày khách đặt báo, chí.
SL	: Số lượng báo, chí khách đặt mua.		

- 1. Tách về các quan hệ có dạng chuẩn cao, không tồn thất thông tin.
- 2. Thông tin về tiền khách đặt mua.
- 3. Có bao nhiêu người mua các loại báo.
- 4. Xác định thông tin về tất cả khách hàng đặt mua báo NDA (Nhân Dân) trong quý I/2000.(trong tất cả 4 quý).
- 5. Thông tin về các loại báo khách hàng đặt mua trong quý I/2000.

6. Tính tổng số lượng và tiền các loại báo(tạp chí) khách đặt mua trong từng quý năm 1999.
7. Tính tổng số lượng và tiền các loại báo(tạp chí) khách đặt mua trong các tháng năm 1999.
8. Hãy xác định thông tin về tất cả khách hàng đặt mua tất cả các loại báo có trong danh mục.
9. Lập bảng kê tính tổng tiền các loại báo mà khách đặt mua trong từng tháng?
10. Lập bảng kê tính tổng tiền các loại báo mà khách đặt mua trong từng quý năm 1999?
11. Thông tin về các loại báo khách đặt mua có số lượng và tiền cao hơn số lượng và tiền khách đặt mua báo “VHT” văn hoá&thể thao?
12. Thống kê số lượng các loại báo & tiền khách đặt mua từng quận.
13. Cho biết tháng nào có số lượng khách đặt mua báo Nhân dân cao hơn tháng 3 hàng năm.
14. Hãy xác định thông tin về các loại báo được đặt mua nhiều nhất
15. Những loại báo nào mà tất cả khách hàng đặt mua.

Bài 2: Thông tin về cuộc đàm thoại:

T# : Mã tổng đài.	TD : Tên gọi tổng đài.
DV: Đơn vị quản lý.	DL : Dung lượng.
B# : Mã thuê bao	TB : Tên thuê bao.
DC: Địa chỉ thuê bao.	SM : Số máy thuê bao.
NG1: Ngày lắp đặt.	DT : Đối tượng thuê bao.
NG2: Ngày gọi.	SĐT : Số máy đến.
BD: Giờ bắt đầu.	KT : Giờ kết thúc

1. Thông tin về các loại tổng đài.
2. Thông tin về các đối tượng khách hàng .
3. Thông tin về các hướng trong tháng.
4. Thông tin về các đơn vị quản lý.
5. Thông tin về các loại tổng đài có cùng dung lượng.
6. Thông tin đối tượng sử dụng điện thoại theo từng Quận.
7. Có bao nhiêu cuộc gọi đi từ một tổng đài.
8. Có bao nhiêu cuộc gọi nội hạt, liên tỉnh, quốc tế.
9. Có bao nhiêu cuộc gọi liên tỉnh , quốc tế bị chệch cước.
10. Sản lượng phút theo từng tổng đài.
11. Sản lượng phút theo từng hướng.
12. Sản lượng phút theo từng tháng.
13. Sản lượng phút theo hướng/ tháng.
14. Sản lượng phút theo tháng/hướng.
15. Sản lượng phút theo đối tượng.

16. Hướng nào thường xuyên (tháng) xảy ra chệch cước.
17. Hướng nào trong năm không có hiện tượng chệch cước.
18. Thống kê tất cả thuê bao có điện thoại trùng nhau.
19. Liệt kê các thuê bao tháng nào cũng sử dụng điện thoại.
20. Liệt kê các thuê bao trong năm không một lần sử dụng gọi đi liên tỉnh, quốc tế.
21. Những thuê bao nào có số cuộc gọi trên 100 cuộc tháng.
22. Thuê bao nào có số phút gọi trên 1000 phút/tháng.
23. Số máy nào hàng tháng đều có các cuộc gọi đến.
24. Số máy nào trong một tháng nhận 100 cuộc.
25. Tính số cuộc gọi đến cho từng thuê bao đến.
26. Tính sản lượng phút tổng đài/quý.
27. Tổng đài nào có số cuộc gọi, sản lượng phút / tháng cao hơn tổng đài “A” do Cty điện thoại quản lý.
28. Địa bàn nào thường có các cuộc gọi cao nhất.
29. Tổng số cuộc gọi qua tổng đài.
30. Tổng số cuộc gọi tháng
31. Tổng số cuộc gọi đối tượng
32. Tổng số cuộc gọi Quý.
33. Số máy nào có mật độ gọi cao.
34. Thống kê các thuê bao hàng tháng sử dụng 5 cuộc gọi.

Bài 3: Quản lý mạng cáp

- TC# : Mã tuyến T : Tên gọi tuyến
DAI : Độ dài tuyến cáp. MC# : Mã cáp.
MC : Tên gọi cáp GIA : Giá cáp
35. SL: Số lượng cáp lắp đặt NSX : Nước sản xuất
 1. Giá trị các loại cáp được lắp đặt.
 2. Giá trị các loại cáp chưa được lắp đặt.
 3. Giá trị các loại cáp được lắp đặt trong từng năm.
 4. Giá trị các loại cáp Việt nam được lắp đặt trên từng tuyến.
 5. Giá trị các loại cáp Việt nam chưa được lắp đặt.
 6. Các loại cáp có tổng giá trị lắp đặt trên 5 tỷ đồng VN.
 7. Các loại cáp có tổng giá trị trên 5 tỷ đồng chưa được lắp đặt.
 8. Tổng giá trị lắp đặt trên từng tuyến.
 9. Loại cáp nào được lắp đặt trên tuyến nhưng trong các kho không còn.
 10. Tuyến nào có tất cả các loại cáp như trong kho có.
 11. Tuyến nào có tổng giá trị lắp đặt lớn hơn tuyến Hà Nội-Quảng ninh.

5 CÁC NGÔN NGỮ THAO TÁC DỮ LIỆU

Chương V sẽ giới thiệu một số ngôn ngữ thao tác dữ liệu, như ngôn ngữ dữ liệu dựa trên đại số quan hệ, ngôn ngữ SQL Ngôn ngữ xử lý dữ liệu được chia thành hai loại khác nhau dựa trên hai chức năng khác nhau. Đó là các chức năng khai báo và chức năng tính toán. Với các trình ứng dụng, dữ liệu thường được mô tả và tính toán khi chương trình thực hiện, còn trong các hệ thống CSDL, dữ liệu được định nghĩa một lần duy nhất. Vì vậy cần có một phương tiện để định nghĩa ngôn ngữ dữ liệu có khả năng mô phỏng các phép tính quan hệ có tính đầy đủ hơn. Nội dung của chương bao gồm:

- Giới thiệu tổng quan về ngôn ngữ thao tác dữ liệu.
- Các phép toán đại số quan hệ
- Đại số quan hệ – ngôn ngữ vấn tin
- Ngôn ngữ vấn tin SQL

5.1 Ngôn ngữ cơ sở dữ liệu

Theo cách nhìn của người sử dụng, ngôn ngữ thao tác dữ liệu là một tập các phép toán dùng để thao tác trên các hệ cơ sở dữ liệu. Như trong các chương II và III đã khảo sát, biểu diễn dữ liệu một cách thống nhất dẫn đến sự thống nhất tương ứng trong các phép toán thao tác, vì thông tin được biểu diễn bằng một và chỉ một cách nên chỉ cần một phép toán cho mỗi một chức năng cơ bản (thêm, bổ sung, loại bỏ...). Ngược với nó, trong nhiều cấu trúc phức tạp, thông tin được biểu diễn bằng nhiều cách. Và vì vậy, cần phải cài đặt nhiều tập các phép toán thao tác khác nhau.

5.1.1 Ngôn ngữ định nghĩa dữ liệu

Lược đồ khái niệm được đặc tả bằng một ngôn ngữ, một phần của hệ quản trị cơ sở dữ liệu, được gọi là ngôn ngữ định nghĩa dữ liệu - DDL (Data Definition Language).

Ngôn ngữ định nghĩa dữ liệu bao gồm một hệ thống ký hiệu mô tả các kiểu thực thể và mối liên hệ giữa chúng theo một mô hình dữ liệu cụ thể. Ngôn ngữ định nghĩa dữ liệu được sử dụng khi thiết kế CSDL hoặc sửa đổi thiết kế CSDL. Nó không sử dụng để truy xuất dữ liệu hoặc sửa đổi dữ liệu. Có những câu lệnh mô tả cấu trúc vật lý theo thuật ngữ trừu tượng. Thiết kế chi tiết CSDL vật lý được thực hiện bởi các thủ tục của hệ quản trị CSDL- DBMS. Khi thực hiện, chúng sẽ biên dịch các câu lệnh ngôn ngữ định nghĩa dữ liệu.

Ví dụ: Về ngôn ngữ định nghĩa dữ liệu SQL:

```
CREATE TABLE QLCAP (TC#:CHAR(3), MC#: CHAR(3), SL: INT,  
DATE: CHAR(6), GTR: INT).  
CREATE INDEX FOR QLCAP ON TC#, MC#;
```

- CREATE TABLE QLCAP mô tả quan hệ với tên là QLCAP và các thuộc tính của nó, đồng thời cấu trúc cài đặt vật lý là các tên trường, kiểu trường là các số nguyên và chuỗi ký tự có chiều dài cố định.

- CREATE INDEX mô tả cần tạo ra một chỉ mục trên số hiệu của tuyến cáp và số hiệu loại cáp như là các thành phần của lược đồ vật lý. Cho phép tìm kiếm các thông tin về các tuyến cáp, về các loại cáp đã được lắp đặt trên các tuyến cáp. Trình biên dịch của DDL có thể chọn một hàm băm có khóa là chuỗi 3 ký tự cho mỗi trường TC# và MC# và nó có thể lưu các bản ghi trong bucket, tùy theo giá trị hàm băm của TC# và MC#.

5.1.2 Ngôn ngữ thao tác dữ liệu

Ngôn ngữ thao tác dữ liệu DML (Data Manipulation Language) hay còn gọi là ngôn ngữ vấn tin (Query Language) dùng để thao tác trên các quan hệ CSDL, bao gồm một số phép toán, các toán hạng là các quan hệ và kết quả của các phép toán cũng là một quan hệ. Ngôn ngữ thao tác dữ liệu gồm 2 nhóm phép toán:

a) Nhóm các phép toán lưu trữ:

- Chèn thêm: Chèn vào CSDL từ vùng làm việc đệm chứa các thông tin về một bản ghi cụ thể.
- Xoá: Xoá một bộ hay xoá một nhóm các bộ.
- Sửa đổi: Sửa đổi giá trị của một số thuộc tính.

b) Nhóm các phép toán tìm kiếm gồm:

- Phép chọn SELECT: Tạo ra một quan hệ mới, các bộ được rút ra một cách duy nhất từ các bộ của quan hệ nguồn thoả mãn một tân từ xác định.
- Phép chiếu PROJECT: Tạo ra một quan hệ mới, các thuộc tính được rút ra, hoặc được biến đổi từ các thuộc tính của quan hệ nguồn, các bộ là các bộ của quan hệ nguồn bỏ đi những bộ trùng lặp.
- Phép kết nối JOIN: Nhằm tạo ra một quan hệ mới, bằng cách nối nhiều quan hệ trên miền thuộc tính chung. Các thuộc tính của quan hệ kết quả là các thuộc tính của các quan hệ thành viên và các bộ là ghép nối các bộ của quan hệ nguồn có cùng chung giá trị thuộc tính chung.

Chương này khảo sát hai lớp ngôn ngữ hỏi đáp cho mô hình quan hệ, đó là:

1. *Ngôn ngữ đại số quan hệ*, trong đó câu hỏi được biểu diễn nhờ áp dụng các phép tính đặc biệt đối với quan hệ.
2. *Ngôn ngữ tính toán tân từ*, trong đó câu hỏi được biểu diễn là một tập các bộ được thoả mãn các tân từ xác định.

5.2 Các phép toán đại số quan hệ

Đại số quan hệ là một trong những ngôn ngữ thao tác dữ liệu, bao gồm các phép toán trên các quan hệ của một cơ sở dữ liệu cho trước. Đó là các phép toán hợp, giao, chiếu và chọn... Tập hợp các phép toán quan hệ tạo nên một cơ chế truy nhập dữ liệu khá linh hoạt và mềm dẻo. Vì vậy người ta thường lấy đại số quan hệ làm đơn vị đo công suất của hệ quản trị cơ sở dữ liệu quan hệ.

5.2.1 Toán hạng của đại số quan hệ

Toán hạng trong đại số quan hệ là các quan hệ hằng hoặc các biến biểu thị cho các quan hệ bậc cố định. Quan hệ R_1 và R_2 là khả hợp khi :

- R_1 và R_2 cùng bậc n .
- Giá trị thuộc tính thứ j của quan hệ này được rút ra từ cùng miền như thuộc tính thứ j của quan hệ kia, $1 \leq j \leq n$.

5.2.2 Phép hợp các quan hệ - UNION

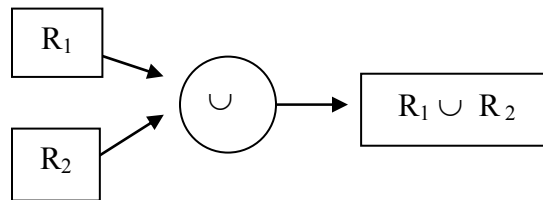
R_1 và R_2 là các quan hệ khả hợp trên cùng tập các thuộc tính của Ω . Khi đó hợp của chúng là một quan hệ trên Ω , các thuộc tính là các thuộc tính của R_1 (hoặc của R_2), các bộ hoặc của R_1 hoặc của R_2 bỏ đi các bộ trùng nhau, chỉ giữ lại những bộ đại diện.

Ký hiệu : $R_1 \cup R_2 = \{t \mid t \in R_1 \text{ hoặc } t \in R_2\}$

UNION (R_1, R_2)

APPEND (R_1, R_2)

$R_1 \cup R_2$



Ví dụ UNION (R_1, R_2)

R_1

M#	N#	NSX
C01	VNA	Việt nam
C02	HQU	Hàn Quốc
C03	JAN	Nhật

R_2

M#	N#	NSX
C04	RUS	Nga
C02	HQU	Hàn Quốc
C05	FRA	Pháp

$R_1 + R_2$

M#	N#	NSX
C01	VNA	Việt nam
C02	HQU	Hàn Quốc
C03	JAN	Nhật
C04	RUS	Nga
C05	FRA	Pháp

Hình 5.1 Mô tả phép hợp đại số quan hệ

5.2.3 Phép giao của các quan hệ - INTERSECT

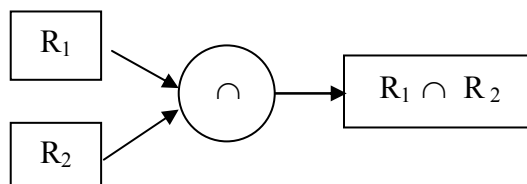
R_1 và R_2 là các quan hệ khả hợp trên cùng tập các thuộc tính của Ω . Khi đó giao của chúng cũng là một quan hệ trên Ω , các thuộc tính của quan hệ kết quả là các thuộc của R_1 hoặc R_2 , các bộ của nó là các bộ có mặt trong R_1 và trong R_2 .

Ký hiệu : $R_1 \cap R_2 = \{t \mid t \in R_1 \text{ And } t \in R_2\}$

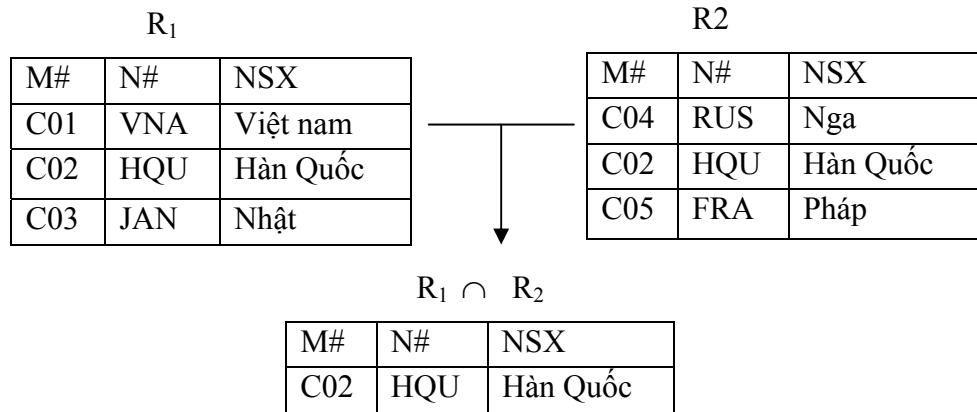
INTERSECT (R_1, R_2)

AND (R_1, R_2)

$R_1 \cap R_2$



Ví dụ INTERSECT (R_1, R_2)



Hình 5.2 Mô tả phép giao đại số quan hệ

5.2.4 Hiệu của các quan hệ - MINUS

R_1 và R_2 là các quan hệ khả hợp trên cùng tập các thuộc tính của Ω . Khi đó hiệu của R_1 và R_2 là quan hệ kết quả KQ trên các thuộc tính Ω . Các thuộc tính của KQ là các thuộc của R_1 hoặc R_2 , các bộ của KQ là các bộ có mặt trong R_1 nhưng không có mặt trong R_2 . (Tương tự hiệu của R_2 và R_1 gồm các bộ có mặt trong R_2 nhưng không có mặt trong R_1)

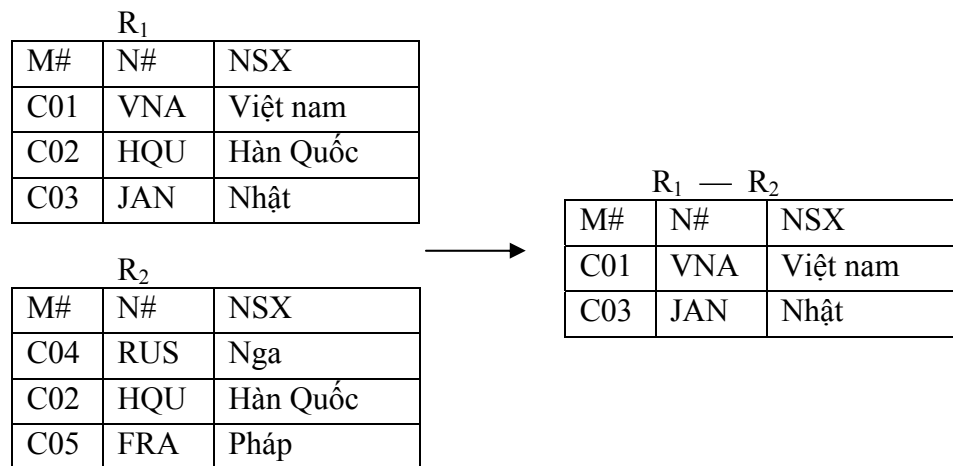
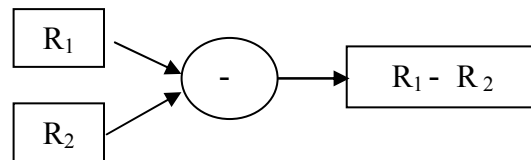
Ký hiệu: $R_1 - R_2 = \{t \mid t \in R_1 \text{ và } t \notin R_2\}$

MINUS (R_1, R_2)

REMOVE (R_1, R_2)

$R_1 - R_2$

Ví dụ: MINUS (R_1, R_2)



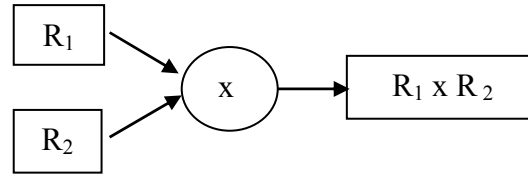
Hình 5.3 Mô tả phép trừ đại số quan hệ

5.2.5 Tích Đề Các - (Cartesian Product)

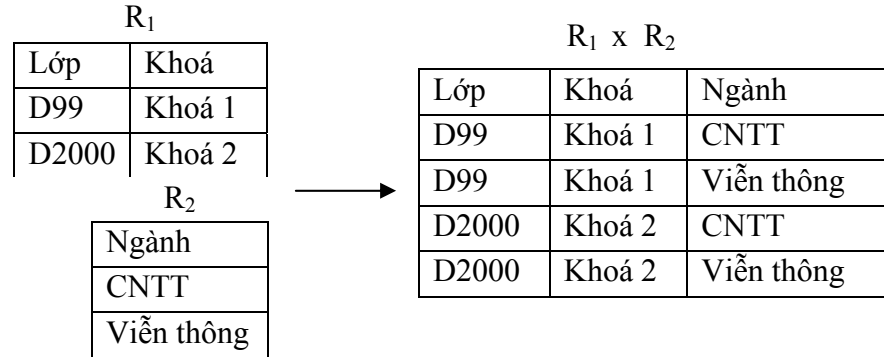
Cho 2 quan hệ $R_1(\Omega)$ và $R_2(\Sigma)$, đặt $\Omega \cap \Sigma = \emptyset$. Tích Đề các của 2 quan hệ $R_1(\Omega)$ và $R_2(\Sigma)$ là một quan hệ P với tập các thuộc tính là $\Omega \Sigma = \Omega \cup \Sigma$ và các bộ của nó thành phần đầu thuộc R_1 và thành phần còn lại thuộc R_2 .

Ký hiệu: $R_1 \times R_2 = P(\Omega\Sigma) := \{t \mid t[\Omega] \in R_1 \ \& \ t[\Sigma] \in R_2\}$

TIMES (R_1, R_2)
 PRODUCT $R_1 \times R_2$
 (R_1, R_2)



Ví dụ PRODUCT $R_1 \times R_2$



Hình 5.4 Mô tả phép tích đại số quan hệ

5.2.6 Phép chọn - Selection

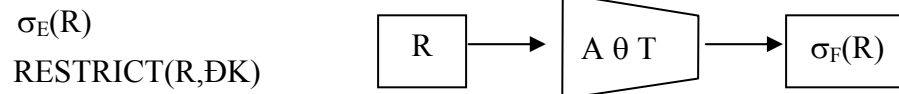
Phép chọn- SELECT tạo ra một quan hệ kết quả. Bậc trùng với bậc quan hệ nguồn và các bộ của nó là các bộ của quan hệ nguồn thỏa một biểu thức logic nào đó.

Định nghĩa: R là một quan hệ trên tập các thuộc tính Ω , cho E là một biểu thức logic phát biểu trên tập thuộc tính Ω . Nói rằng bộ t thỏa mãn biểu thức logic E, ký hiệu $t[E]$ nếu sau khi thay mọi giá trị của thuộc tính A trong E bởi giá trị $t[A]$ tương ứng thì nhận được một mệnh đề đúng, tức là $t[A]$ nhận giá trị “True”.

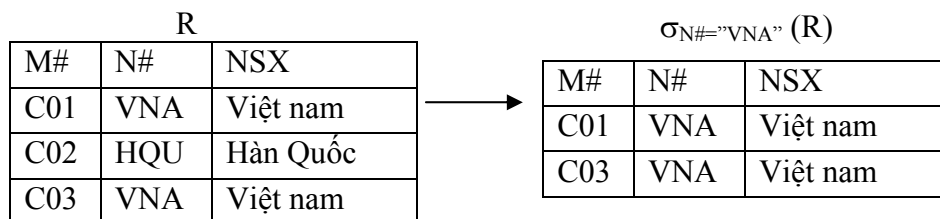
Biểu thức logic được biểu diễn gồm nhiều toán hạng và giữa các toán hạng là các phép so sánh đơn giản, nhận giá trị "đúng" hoặc "sai".

- Toán hạng: là hằng hoặc các thuộc tính quan hệ..
- Các phép so sánh $\theta = \{<, >, \neq, =, \leq, \geq\}$
- Các phép logic: \vee (OR/ hoặc), \wedge (AND /và) và \neg (NOT/phủ định) .

Ký hiệu: $\sigma_E(R) = \{t \mid t \in R \ \& \ t[E] = \text{“True”}\}$



Ví dụ RESTRICT (R, NSX="Việt nam")



Hình 5.5 Mô tả phép chọn đại số quan hệ

5.2.7 Phép chiếu - PROJECT

Từ quan hệ nguồn R bỏ đi một số thuộc tính và/ hoặc sắp xếp lại các thuộc tính còn lại. Các bộ của nó không trùng nhau. Có thể thao tác như sau:

- Loại khỏi quan hệ R tất cả các cột ứng với các thuộc tính trên tập $\Omega \setminus X$.
- Bỏ đi những bộ giống nhau và chỉ giữ lại những bộ đại diện.

Gọi $\pi_{A_1, A_2, \dots, A_k}(R)$ là quan hệ chiếu, khi đó:

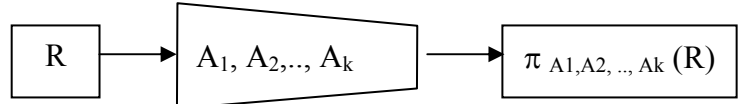
$$\pi_{A_1, A_2, \dots, A_k}(R) = \{t[X] \mid t \in R \text{ và } X = (A_1, A_2, \dots, A_k)\}$$

Ký hiệu :

$$\pi_{A_1, A_2, \dots, A_k}(R)$$

$$R[A_1, A_2, \dots, A_k]$$

$$\text{PROJECT}(R, [A_1, A_2, \dots, A_k])$$



Ví dụ PROJECT (R, NSX)

R		
M#	N#	NSX
C01	VNA	Việt nam
C02	HQU	Hàn Quốc
C03	VNA	Việt nam

NSX
Việt nam
Hàn Quốc

Hình 5.6 Mô tả phép chiếu đại số quan hệ

5.3 Các phép toán đại số bổ sung

5.3.1 Phép chia - DIVISION

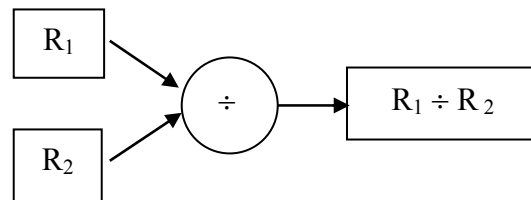
Định nghĩa: Cho 2 quan hệ $R_1(\Omega)$ và $R_2(\Sigma)$, $\Sigma \subset \Omega$, $\Sigma \neq \emptyset$. Thương của phép chia quan hệ $R_1(\Omega)$ cho $R_2(\Sigma)$ là một quan hệ trên tập các thuộc tính $X = \Omega \setminus \Sigma$ được xác định như sau:

$$R_1 \div R_2 := \{t \mid \forall s \in R_2, (t, s) \in R_1\}$$

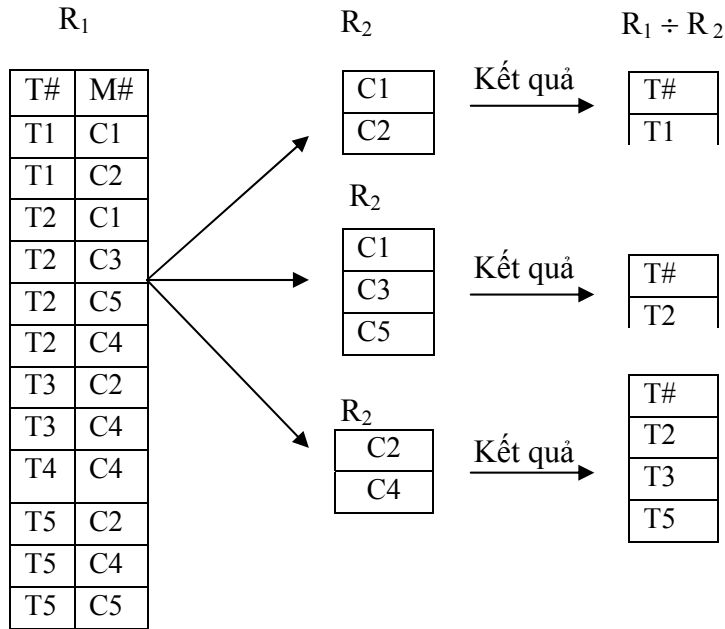
Có thể biểu diễn phép chia bằng các phép toán cơ bản như sau:

- Thực hiện phép chiếu $T = \pi_X(R_1)$
- $V = \pi_X((T \times R_2) - R_1)$.
- $R_1 \div R_2 = T - V = \pi_X(R_1) - \pi_X((\pi_X(R_1) \times R_2) - R_1)$

Ký hiệu: $R_1 \div R_2$
 DIVIDE (R₁, R₂)
 R_1 / R_2



Ví dụ DIVIDE (R₁, R₂)



Hình 5.8 Mô tả phép chia đại số quan hệ

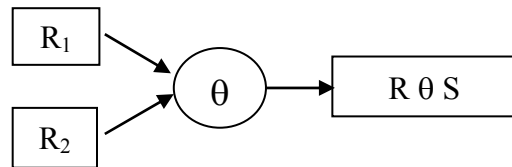
5.3.2 Phép kết nối – JOIN

a) Kết nối bằng nhau và không bằng nhau:

Định nghĩa: Phép kết nối - θ của quan hệ $R(\Omega)$ và $S(\Sigma)$ theo thuộc tính thứ i và j , trong đó $\theta = \{<, >, \neq, \leq, \geq\}$, là quan hệ trên các thuộc tính của $\Omega \Sigma = \Omega \cup \Sigma$ và các bộ là những bộ của tích Descartes của R và S sao cho các thành phần trong thứ i của R quan hệ θ với thành phần thứ j của quan hệ S . Nếu θ là quan hệ “=” gọi là kết nối bằng, ngược lại gọi là kết nối không bằng. Ký hiệu là $R \theta S$.

- Tích Descartes $T = R \times S$.
- $R \theta S$ Thực hiện phép chọn $\sigma_{A_i \theta B_j}(T)$
- $R \theta S = \sigma_{A_i \theta B_j}(R \times S)$

Ký hiệu: $R \theta S$
JOIN (R, S, ĐK)



Ví dụ JOIN (R, S, B < D)

A	B	C
1	2	3
4	5	6
7	8	9

Quan hệ R

D	E
3	1
6	2

Quan hệ S

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2

$R \theta S : B < D$

Hình 5.9 Mô tả phép kết nối đại số quan hệ

b) *Kết nối tự nhiên:*

Định nghĩa: Cho quan hệ $R(\Omega)$ và $S(\Sigma)$. Khi đó kết nối tự nhiên của quan hệ R và S là một quan hệ P trên các thuộc tính $\Omega \Sigma = \Omega \cup \Sigma$, và

$$R \bowtie S = P(\Omega \Sigma) := \{t \in R \theta S \mid t[\Omega \cap \Sigma]_R = t[\Omega \cap \Sigma]_S\}.$$

- Tình Descartes $T = R \times S$.
- Với mọi $t \in T$, $t[\Omega \cap \Sigma] \in R = t[\Omega \cap \Sigma] \in S$
- $R \bowtie S =$ Thực hiện phép chọn $\sigma_{A_i \theta B_j}(T)$
- $R \bowtie S = \pi_X(\sigma_{R.X=S.X}(R \times S))$

Ví dụ

A	B	C
a	b	c
d	b	c
b	b	k
c	a	d

Quan hệ R

B	C	D
b	c	d
b	c	y
a	d	b

Quan hệ S

A	B	C	D
a	b	c	a
a	b	c	y
d	b	c	d
d	b	c	y
c	a	d	b

$R \bowtie S$

Hình 5.10 Phép kết nối tự nhiên

c) *Nửa kết nối:*

Định nghĩa: Cho quan hệ $R(\Omega)$ và $S(\Sigma)$. Khi đó nửa kết nối của quan hệ R và S là phép chiếu trên các thuộc tính Ω trong phép kết nối tự nhiên của R và S :

$$R \ltimes S = \pi_{\Omega}(R \times S)$$

Ví dụ

A	B	C
a	b	c
d	b	c
b	b	k
c	a	d

Quan hệ R

B	C	D
b	c	d
b	c	y
a	d	b

Quan hệ S

A	B	C	D
a	b	c	a
a	b	c	y
d	b	c	d
d	b	c	y
c	a	d	b

$R \ltimes S$

A	B	C
a	b	c
d	b	c
c	a	d

$R \times S$

Hình 5.11 Phép nửa kết nối

5.4 Các tính chất của đại số quan hệ

Kết quả các phép toán cũng là một quan hệ. Tuy nhiên các toán hạng là các quan hệ giao hoán với nhau, nghĩa là các bộ thay đổi thứ tự trong bảng, điều này không quan trọng với cách biểu diễn dữ liệu. Nó vẫn đảm bảo tính độc lập và tính toàn vẹn dữ liệu.

5.4.1 Tính chất giao hoán

Các phép hợp, giao và kết nối trong đại số quan hệ là các phép toán có tính chất giao hoán. Tức là với mọi quan hệ R_1, R_2 thì

$$R_1 \cup R_2 = R_2 \cup R_1$$

$$R_1 \cap R_2 = R_2 \cap R_1$$

$$R_1 \mid \times \mid R_2 = R_2 \mid \times \mid R_1$$

5.4.2 Tính chất kết hợp

Các phép hợp, giao, kết nối và tích Descartes trong đại số quan hệ là các phép toán có tính chất kết hợp. Tức là với mọi quan hệ R_1, R_2, R_3 thì:

$$R_1 \cup R_2) \cup R_3 = R_2 \cup (R_1 \cup R_3)$$

$$(R_1 \cap R_2) \cap R_3 = R_2 \cap (R_1 \cap R_3)$$

$$R_1 \mid \times \mid (R_2 \mid \times \mid R_3) = (R_2 \mid \times \mid R_1) \mid \times \mid R_3$$

$$(R_1 \times R_2) \times R_3 = R_2 \times (R_1 \times R_3)$$

5.4.3 Tính chất tích lũy đẳng

Các phép hợp, giao và kết nối trong đại số quan hệ là các phép toán có tính lũy đẳng. Tức là với mọi quan hệ R trên tập các thuộc tính Ω :

$$R_1 \cup R_1 = R_1$$

$$R_1 \cap R_1 = R_1$$

$$R_1 \mid \times \mid R_1 = R_1$$

5.4.4 Một số tính chất khác

Cho R là quan hệ bất kỳ trên Ω và S là quan hệ bất kỳ trên Σ . Khi đó:

$$R \cap S = R - (R - S)$$

$$R \div S = R[X] - (R[X] * S - R)[X]$$

$$(R \times S) [X] = R[X] \times S[X]$$

$$X = \Omega \setminus \Sigma$$

5.5 Đại số quan hệ – ngôn ngữ vấn tin

Từ các phép toán của đại số quan hệ có thể xây dựng ngôn ngữ vấn tin trên các quan hệ. Một câu hỏi có thể được biểu diễn bởi biểu thức quan hệ hay một cây phép toán quan hệ.

Khảo sát các quan hệ sau:

- Thông tin về tuyến cáp: R_1 (T#, TC, DAI).
- Thông tin về các loại cáp: R_2 (M#, MC, N#, NSX, GIA).
- Thông tin về các loại cáp trên tuyến cáp: R_3 (T#, M#, SL, NG)

Trong đó:

T#	Mã tuyến cáp.	N#	Mã nước sản xuất cáp.
TC	Tên tuyến cáp.	NSX	Nước sản xuất.
DAI	Độ dài tuyến	GIA	Đơn giá cáp.
M#	Mã cáp.	SL	Số lượng.
MC	Tên cáp	NG	Ngày lắp đặt.

5.5.1 Biểu thức quan hệ

Ví dụ: Cho biết tên các tuyến có độ dài trên 100 Km

- Chọn những bộ có DAI > 100 Km từ quan hệ R_1 , cho kết quả S.

- Chiều trên thuộc tính T# và TC từ quan hệ S.
- $KQ(T\#,TC) = \pi_{T\#,TC}(\sigma_{DAI>100}(R1))$
 $S = \text{RESTRICT}(R1, DAI > 100)$
 $KQ = \text{PRODUCT}(S, [T\#,TC])$

Ví dụ : Thông tin về các tuyến có lắp đặt các loại cáp Việt nam

- Thực hiện phép chọn trên R3 thoả M#="VNA", quan hệ kết quả T1.
 $T1 = \sigma_{M\#="VNA"}(R3)$
 $T1 = \text{RESTRICT}(R3, [M\#="VNA"])$
- Kết nối R1 với T1 trên thuộc tính chung T# , quan hệ kết quả T2.
 $T2 = R1 \mid_{>}. R1.T\#=T1.T\# < \mid T1$
 $T2 = \text{JOIN}(R1, T1, R1.T\#=T1.T\#)$
- Kết nối R2 với T2 trên thuộc tính chung M# , quan hệ kết quả T3.
 $T3 = R1 \times_{R2.M\#=T2.M\#} T1$
 $T3 = \text{JOIN}(R2, T2, R2.M\#=T2.M\#)$
- Chiều T3 trên các thuộc tính T#, TC, DAI , quan hệ kết quả KQ
 $KQ = \pi_{T\#,TC,DAI}(T3)$

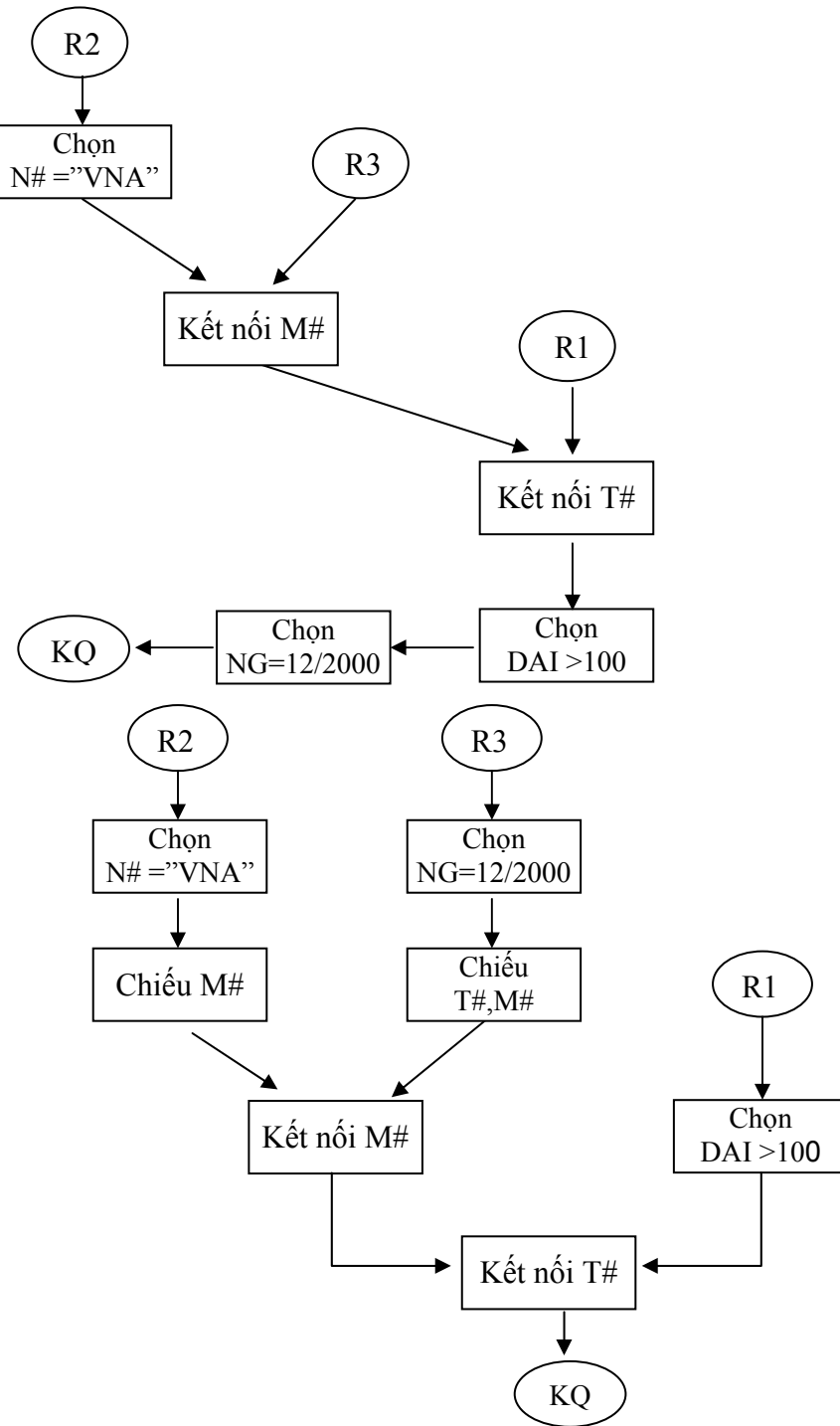
Ví dụ : Thông tin về các tuyến lắp đặt tất cả các loại cáp.

- Thực hiện phép chiếu R3 trên các thuộc tính T#, M#, quan hệ kết quả T1.
 $T1 = \pi_{T\#,M\#}(R3)$
 $T1 = \text{PROJECT}(R3, [T\#,M\#])$
- Thực hiện phép chiếu R2 trên thuộc tính M#, quan hệ kết quả T2.
 $T2 = \pi_{M\#}(R2)$
 $T2 = \text{PROJECT}(R2, [M\#])$
- Thực hiện phép chia T1 cho T2 trên thuộc tính M#, quan hệ thương T3.
 $T3 = R1 \div R2 = \pi_X(T1) - \pi_X((\pi_X(T1) \times T2) - T1)$
 $T3 = \text{DIVIDE}(T1, T2)$
- Kết nối T3 với R1 trên thuộc tính chung T# , quan hệ kết quả T4.
 $T4 = R1 \times_{R1.T\#=T3.T\#} T3$
 $T4 = \text{JOIN}(R1, T3, [R1.T\#=T3.T\#])$
- Chiều T4 trên các thuộc tính T#, TC, DAI , quan hệ kết quả KQ
 $KQ = \pi_{T\#,TC,DAI}(T3)$
 $KQ = \text{PROJECT}(R3, [T\#,TC,DAI])$

5.5.2 Cây đại số quan hệ

Câu hỏi truy vấn có thể biểu diễn bằng các phép toán đại số quan hệ, có thể biểu diễn bằng cây đại số quan hệ.

Ví dụ Các tuyến cáp dài trên 100 km có lắp đặt cáp “Việt Nam” trong tháng 12 năm 2000.



Hình 5.12 Biểu thức đại số biểu diễn bằng cây đại số quan hệ

5.5.3 Cài đặt đại số quan hệ

1. Quy ước:

- *For each t in R [with B] do A endfor* : Lệnh này có nghĩa là thực hiện phép toán A đối với những bộ t thuộc trong quan hệ R thoả mãn điều kiện B. Phần viết trong [] là tùy chọn.

- *If B then A [else C] endif* : Nếu điều kiện B thỏa mãn thực hiện phép toán A, nếu không thực hiện phép toán C.
- *R create (P, X)* : Tạo một quan hệ rỗng P, với tập thuộc tính X cho trước. Lưu ý rằng khi nói đến thuộc tính A là phải hiểu rằng A đã có một miền biến thiên cụ thể nào đó, tức là A đã thuộc một kiểu cho trước.
- *Attrib (R)* : Hàm này trả về tập các thuộc tính quan hệ R.
- *#R* : Hàm này trả về lực lượng (số bộ) của quan hệ R .
- *Add t to P* : Nếu bộ t chưa có trong quan hệ P thì nạp t vào P. Phép toán này tương đương với câu lệnh $P = P \cup \{t\}$.
- $t.X = t[X]$: Phép chiếu X trên bộ t hay sự hạn chế của bộ t trên tập X.
- $t = \langle u,v \rangle$: Tạo một bộ mới t từ các bộ u,v đã có.
- *t in R* : Cho giá trị đúng (“True”) nếu t đã có trong quan hệ R, ngược lại sẽ cho giá trị sai (“False”).

2) *Thuật toán hợp*

Algorithm Union
Format $P = R \cup S$
Input Quan hệ R, S khả hợp
Output $P = \{ t \mid t \in R \text{ OR } t \in S \}$
Begin
 R create(P, Attrib (R)) ;
 For each t in R do
 Add t to P
 Endfor ;
 For each v in S do
 If not (v in P) then
 Add v to P ;
 Endif ;
 Endfor
 Return (P)
End Union.

3) *Thuật toán giao*

Algorithm Intersection
Format $P = R \cap S$
Input Quan hệ R, S khả hợp
Output $P = \{ t \mid t \in R \text{ AND } t \in S \}$
Begin
 R create (P, attrib (R)) ;
 For each t in R do

```
        If  $t$  in S then
            Add  $t$  to P
        Endif
    Endfor;
    Return (P)
End intersection ;
```

4) *Thuật toán trừ*

```
Algorithm          Substraction
Format             $P = R - S$ 
Input             Quan hệ R, S khả hợp
Output             $P = \{ t \mid t \in R \text{ AND } t \notin S \}$ 
Begin
    R create (P, attrib (R)) ;
    For each  $t$  in R do
        If not ( $t$  in S) then
            Add  $t$  to P;
        Endif
    Endfor;
    Return (P)
End Subs
```

5) *Thuật toán chọn*

```
Algorithm          Selectoin
Format             $P = R(E)$ 
Input             Quan hệ R
                  Biểu thức đại số quan hệ
Output             $P = \{ t \mid t \in R \ \& \ t(E) = 'True' \}$ 
Begin
    R create (P, attrib (R)) ;
    For each  $t$  in R with  $t(E)$  do
        Add  $t$  to P;
    Endfor;
    Return (P)
End Selection :
```

6) *Thuật toán chiếu*

```
Algorithm          Projection
Format             $P = R[X]$ 
Input             Tập các thuộc tính X
Output             $P = \{ t \mid t.X \ \& \ t \in R \}$ 
```

```

Begin
  R create (P, X) ;
  For each t in R do
    Add t[X] to P;
  Endfor;
  Return (P)
End Projection

```

7) *Thuật toán kết nối tự nhiên*

```

Algorithm      Join
Format        P = R |><|. S
Input         Quan hệ R, S
Output        P = {<u,v/x> | u ∈ R, v ∈ R
                u.X = v.X, X = Attrb(R) ∩ Attrb(S)}

```

```

Begin
  R create (P, attrib (R) ∪ attrib(S - R) ;
  X = attrib (R) ∩ attrib(S)
  For each u in R do
    For each v in S with u.X = v.X do
      Add <u,v/X> to P;
    Endfor
  Endfor
  Return (P)
Endjoin :

```

8) *Thuật toán phép chia*

Cho 2 quan hệ R và S, thuật toán chia 2 quan hệ $R \div S$ như sau:

Đặt $M = \text{attrib}(R)$, $N = \text{attrib}(S)$, $X = M - N$. Theo định nghĩa, P sẽ là một quan hệ kết quả trên tập thuộc tính X:

$$P = R \div S := \{ u.X \mid u \in R \ \& \ \forall v \in S \ \text{thì} \ \langle u.X, v \rangle \in R \}$$

$$\Leftrightarrow \{ t \mid \forall s \in S, (t, s) \in R \}$$

```

Algorithm      Division
Format        P = R ÷ S
Input         Quan hệ R, quan hệ S
Output        P = R1 ÷ R2 = πX(R) - πX((πX(R) x S) - R)
Begin

```

```

  X = Attrib (R) – Attrib (S);
  Begin /* Thực hiện phép chiếu πX(R) /
    R create (T1, X) ;
    For each t in R do

```

```
        Add t[X] to T1;
    Endfor;
    Return (T1)
End Projection
Begin /* Thực hiện tích đề các  $\pi_X(R) \times S$  /
    R create (T2, Attrib(R)) ;
    For each t in T1 do
        For each v in S do
            Add <t,v> to T2;
        Endfor;
    Endfor;
    Return (T2)
End Descartes
Begin /* thực hiện phép trừ  $(\pi_X(R) \times S) - R$  /
    R create (T3, Attrib(R)) ;
    For each t in T2 do
        If not (t in R) then
            Add t to T3;
        Endif
    Endfor;
    Return (T3)
End Subs
Begin /* Thực hiện phép chiếu  $\pi_X((\pi_X(R) \times S) - R)$  /
    R create (T4, X) ;
    For each t in T3 do
        Add t[X] to T4;
    Endfor;
    Return (T4)
End Projection
Begin /* thực hiện phép trừ  $\pi_X(R) - \pi_X((\pi_X(R) \times S) - R)$  /
    R create (P, X) ;
    For each t in T1 do
        If not (t in T4) then
            Add t to P;
        Endif
    Endfor;
    Return (P)
End Subs
End Devision.
```

5.6 Ngôn ngữ vấn tin có cấu trúc SQL

5.6.1 Giới thiệu

SQL (Structured Query Language) được xây dựng và phát triển trên cơ sở ngôn ngữ con dữ liệu SEQUEL (Structured English Query Language) và QUERY BY EXAMPLE, là ngôn ngữ hỏi-đáp bằng thí dụ. SQL là một ngôn ngữ được IBM phát triển ở San Jose, với mục đích đề sử dụng trong các hệ CSDL thử nghiệm System R. Ngày nay SQL đã được thương mại hoá và được sử dụng trong rất nhiều hệ thống CSDL trong các lĩnh vực khác nhau.

Ngôn ngữ vấn tin SQL không chỉ là một ngôn ngữ con thao tác dữ liệu độc lập, mà còn là một ngôn ngữ lập trình được nhúng trong các ngôn ngữ lập trình chủ khác. Nó là ngôn ngữ con dữ liệu kiểu quan hệ, được xây dựng trên cơ sở đại số quan hệ và các phép toán quan hệ. Vì vậy SQL có nhiều giao diện khác nhau, được thực hiện qua các thực đơn, cửa sổ, lưới... hoặc có thể lập trình trên các công cụ của nó. SQL trở thành một công cụ truy xuất dữ liệu chuẩn mực và tương đối mạnh, được ứng dụng rộng rãi trong nhiều lĩnh vực tổ chức các hệ thông tin quản lý tác nghiệp.

Cũng như các ngôn ngữ thao tác dữ liệu khác, ngôn ngữ SQL gồm 2 thành phần: ngôn ngữ định nghĩa dữ liệu và ngôn ngữ thao tác dữ liệu. Và có thể sử dụng theo các phương thức tương tác và nhúng vào một ngôn ngữ chủ nào đó.

1. Ngôn ngữ SQL gồm 2 thành phần:

SQL ——— Ngôn ngữ định nghĩa dữ liệu DDL (Data definition language)
 └─── Ngôn ngữ thao tác dữ liệu DML (Data manipulation language)

SQL-DDL ——— Tạo, sửa và xóa cấu trúc quan hệ.
 └─── Bảo mật và quyền truy nhập.

SQL-DML ——— Truy vấn: Các chức năng tìm kiếm.
 └─── Cập nhật: Chức năng thêm, sửa, xoá dữ liệu.

2. Sử dụng SQL trong m.....

SQL ——— SQL trong chế độ tương tác
 └─── SQL trong chế độ nhúng trong ngôn ngữ chủ.

5.6.2 Câu lệnh SELECT

Dạng thông dụng nhất của vấn tin trong SQL là câu lệnh chọn, có cú pháp như sau:

```
SELECT   R1.A1, R2.A2,..., Rn.An
FROM    R1, R2,..., Rn
WHERE   E
GROUP   BY R1.Aj HAVING <điều kiện>
ORDER   BY
```

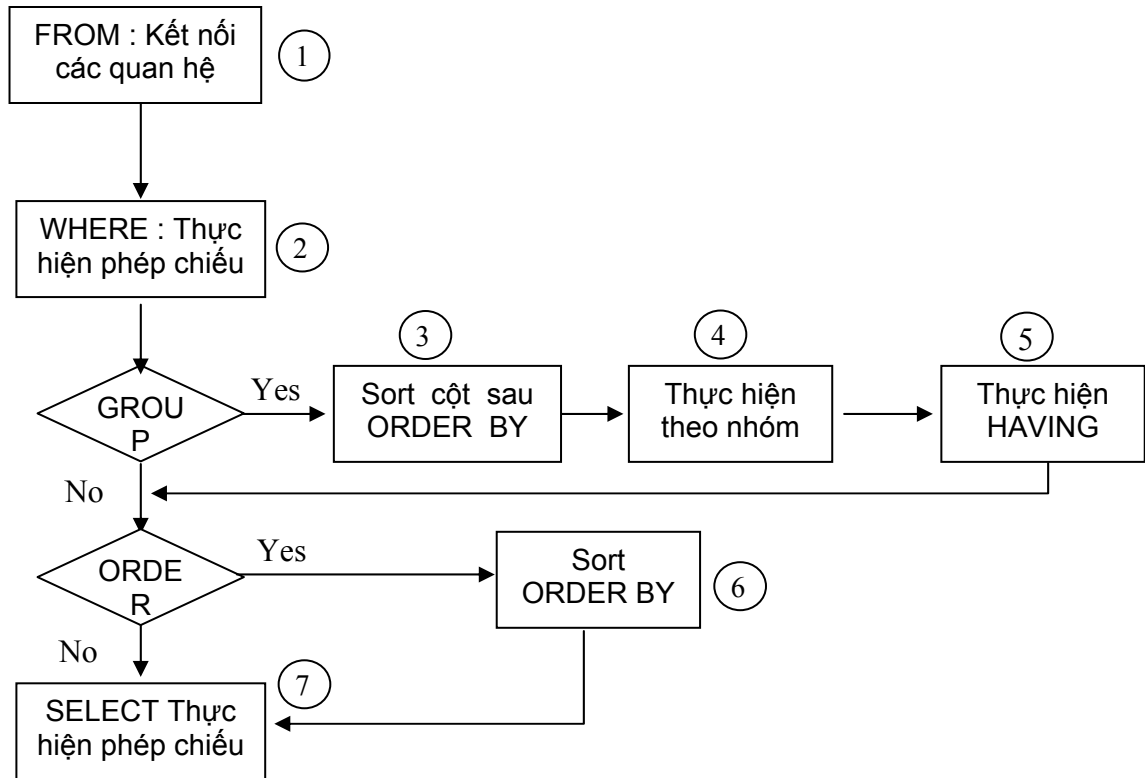
Trong đó R₁, R₂,..., R_n là danh sách các tên quan hệ nguồn và R₁.A₁, R₂.A₂,..., R_n.A_n là danh sách các thành phần được kết xuất, R.A nghĩa là thuộc tính A của quan hệ R. Nếu

trong danh sách sau từ khóa FROM chỉ có một quan hệ có thuộc tính A thì có thể thay A bởi $R.A$ trong danh sách sau SELECT. E là một biểu thức chứa các phép logic AND, OR và NOT và các phép toán so sánh $=, <., >, \neq, \geq, \leq, \dots$

Câu vấn tin SQL ở trên có thể biểu diễn bằng biểu thức đại số quan hệ như sau :

$$\pi_{R_1.A_1, A_2, \dots, R_n.A_n} (\sigma_E (R_1 \times R_2 \times \dots \times R_n)).$$

Nghĩa là lấy tích các quan hệ trong mệnh đề FROM, chọn theo biểu thức E sau WHERE và cuối cùng chiếu trên các thuộc tính sau mệnh đề SELECT.



Hình 5.13 Các bước thực hiện câu lệnh SELECT

1. *Bước thứ nhất:* Thực hiện các phép tích Đề các, hoặc phép kết nối tự nhiên các quan hệ sau FROM: R_1, R_2, \dots, R_n . Nếu lực lượng của $|R_i| = l_i$ và bậc là $n_i, i = 1 \div n$. Khi đó, kết quả của tích sẽ là một quan hệ có $n_1 + n_2 + \dots + n_n$ cột và có $l_1 \times l_2 \times \dots \times l_n$ hàng.
2. *Bước thứ hai:* Thực hiện phép toán chọn trên quan hệ sau bước 1, thỏa biểu thức đại số E sau mệnh đề WHERE.
3. *Bước thứ ba:* Nếu trong câu lệnh có chứa GROUP BY, nghĩa là hệ thống phải phân loại (sort) theo các cột được mô tả sau GROUP BY. Thứ tự phân loại theo chiều từ phải qua trái. Hay nói cách khác, hệ thống sẽ phân hoạch quan hệ kết quả sau bước thứ 2 thành nhiều nhóm tách biệt nhau. Ví dụ GROUP BY ĐC, HT, nghĩa là sắp xếp theo vắn ĐC (địa chỉ), trong ĐC sắp xếp theo vắn HT (họ tên).

4. *Bước thứ tư:* Thực hiện các phép toán gộp nhóm. Kết quả là một quan hệ mới được thiết lập, các hàng là kết quả của các phép gộp nhóm.
5. *Bước thứ năm:* Mệnh đề HAVING xuất hiện nếu GROUP BY đã xuất hiện. Sẽ thực hiện phép chọn trên quan hệ sau bước 4.
6. *Bước sáu:* Nếu có mệnh đề ORDER BY, thực hiện phép phân loại.
7. *Cuối cùng,* bước 7, hệ thống thực hiện phép chiếu sau SELECT.

Ví dụ: “Liệt kê những khách hàng có số dư âm”, được biểu diễn trong SQL như sau :

```
SELECT    TEN_K
FROM      KHAC_HANG.
WHERE     TAI_KHOAN < 0 ;
Kết quả:  TEN_K
           Nga
           Hồng
```

Trong câu lệnh trên chỉ có một quan hệ duy nhất trong mệnh đề FROM, không thể có sự nhầm lẫn các thuộc tính. Vì vậy không cần phải đặt tên quan hệ trước các thuộc tính. Quan hệ kết quả chỉ có một cột là TEN_K. Nếu muốn có một tiêu đề khác cho cột này, có thể đưa ra một bí danh (alias) cho TEN_K bằng cách viết bí danh ngay sau TEN_K có một dấu cách (Blank)

```
SELECT    TEN_K HO_TEN
FROM      KHACH_HANG
WHERE     TAI_KHOAN < 0 ;
Kết quả:  HO_TEN
           Nga
           Hồng
```

Nếu in ra mọi thông tin về những khách hàng có số dư âm, có thể viết như sau:

```
SELECT    TEN_K, DIA_CHI, TAI_KHOAN
FROM      KHACH_HANG
WHERE     TAI_KHOAN < 0 ;
```

hoặc chỉ cần viết :

```
SELECT    *
FROM      KHACH_HANG
WHERE     TAI_KHOAN < 0 ;
```

Ký tự * là cách viết của SQL yêu cầu in tất cả các thuộc tính của quan hệ R. Trong thí dụ này, KHACH_HANG là một quan hệ duy nhất trong mệnh đề FROM, không cần phải xác định các thuộc tính quan hệ đó.

Ví dụ Liệt kê thông tin tất cả nhà cung cấp các mặt hàng mà Nga đặt mua.

Thực hiện kết nối tự nhiên của 3 quan hệ HOA_DON, MH_SL và CUNG_CAP bằng cách dùng những đẳng thức trong mệnh đề WHERE để định nghĩa phép nối.

```
SELECT    DISTINCT TEN_CC
```

```
FROM HOA_DON, CUNG_CAP, MH_SL
WHERE TEN_K = "Nga"
      AND HOA_DON.SHD# = MH_SL.SHD#
      AND CUNG_CAP.MAT_HANG= NG_CAP.MAT_HANG
```

Cũng như hầu hết các ngôn ngữ vấn tin khác, SQL không tự động loại đi các bộ trùng lặp. Vì vậy trong một số trường hợp, phụ thuộc yêu cầu câu hỏi, trong câu vấn tin nên sử dụng từ khóa DISTINCT theo sau SELECT.

5.6.3 Biến quan hệ

Trong một số trường hợp, có nhiều câu vấn tin liên quan đến nhiều bộ trong cùng một quan hệ. Để thực hiện được điều này, cần định nghĩa nhiều biến quan hệ ở mệnh đề FROM và dùng những biến quan hệ này làm bí danh của quan hệ.

Ví dụ: Liệt kê tên và địa chỉ của những khách hàng có số dư nhỏ của Nga.

```
SELECT c1.TEN_K, c1.DIA_CHI
FROM KHACH_HANG c1, KHACH_HANG c2
WHERE c1.TAI_KHOAN < c2.TAI_KHOAN
      AND c2.TEN_K = "Nga";
```

Sau mệnh đề FROM c1 và c2 đều là bí danh của KHACH_HANG, chúng trở thành các bộ chạy trên KHACH_HANG.

5.6.4 Toán tử LIKE

Trong mệnh đề WHERE, có thể bổ sung toán tử LIKE để mô tả điều kiện là một giá trị nào đó phải khớp với một mẫu (pattern).

- Ký hiệu % : Nghĩa là một chuỗi bất kỳ có độ dài tùy ý, chuỗi bắt đầu từ vị trí của %.
- Ký tự gạch thấp _ đại diện cho một ký tự bất kỳ thay thế ký tự “_” .

Ví dụ In ra những mặt hàng có ký tự đầu là “E” và giá từ khoảng 1.00 đến 1.99

```
SELECT ITEM, PRICE
FROM SUPPLIES
WHERE ITEM LIKE "E%" AND PRICE LIKE "1.____";
```

5.6.5 Phép toán tập hợp trong mệnh đề WHERE

Trong phần này tiếp tục nghiên cứu các khả năng rộng hơn so với phép toán chọn trong đại số quan hệ. Trong mệnh đề WHERE có thể sử dụng các phép số học để so sánh, vì thế có thể viết những điều kiện này như sau :

```
WHERE A > B + C -10
```

Một khả năng mở rộng của SQL cho phép sử dụng các tập làm *toán hạng* được định nghĩa bằng câu lệnh SELECT-FROM -WHERE đầy đủ, được lồng trong mệnh đề WHERE và được gọi là *câu vấn tin con (subquery)*.

- Phép toán IN biểu diễn các khái niệm thuộc tập. Ngược lại NOT IN biểu diễn khái niệm không thuộc tập .
- ANY thường sử dụng cho các khái niệm tồn tại. Nếu S là một biểu thức biểu diễn cho một tập, thì điều kiện là $A \theta ANY S$ sẽ tương đương với biểu thức logic $(\exists X)$

$(X \in S \wedge A \theta X)$. Điều này có nghĩa là tồn tại thuộc tính A thuộc tập X sao cho A có quan hệ với X bởi các toán tử so sánh θ .

- Tương tự ALL là với mọi: $A \theta \text{ALL } S \Leftrightarrow (\forall X) (X \in S \text{ suy ra } A \theta X)$. Điều này có nghĩa là với mọi thuộc tính A thuộc tập X thì A có quan hệ với X bởi các toán tử so sánh θ .
- BETWEEN... AND: từ..... đến

Ví dụ In tên của tất cả nhà cung cấp có ít nhất một mặt hàng mà “Nga” đặt.

1. Xác định tập S_1 các đơn đặt hàng của “Nga” từ quan hệ HOA_DON
2. Xác định tập S_2 các mặt hàng trong tập đơn đặt hàng S_1 từ quan hệ MH_SL.
3. Xác định tập S_3 là tên các nhà cung cấp các mặt hàng trong tập S_2 bằng cách sử dụng quan hệ CUNG_CAP.

```
(1) SELECT   TEN_CC
(2) FROM    CUNG_CAP
(3) WHERE   MAT_HANG IN
(4)           (SELECT  MAT_HANG
(5)           FROM    MH_SL
(6)           WHERE   O# IN
(7)           (SELECT  O#
(8)           FROM    HOA_DON
(9)           WHERE   TEN_K = "Nga"))
```

Như vậy:

- Tập S_1 được xác định bởi các câu lệnh từ (7)-(9).
- Tập S_2 được xác định bởi các câu lệnh từ (4)-(6).
- Tập S_3 được xác định bởi các câu lệnh từ (1)-(3).

Ví dụ Liệt kê các mặt hàng có giá lớn hơn mọi giá trong SUPPLIES.

```
SELECT  MAT_HANG, GIA
FROM    CUNG_CAP
WHERE   GIA  $\geq$  ALL
           SELECT  GIA
           FROM    CUNG_CAP)
```

Ví dụ Liệt kê các mặt hàng có giá lớn hơn giá của một mặt hàng của Nga đặt.

```
SELECT  MAT_HANG, GIA
FROM    CUNG_CAP
WHERE   GIA  $\geq$  ANY
           (SELECT  GIA
           FROM    CUNG_CAP
           WHERE   MAT_HANG IN
           (SELECT  MAT_HANG
```

```
FROM CUNG_CAP
WHERE O# IN
(SELECT O#
FROM HOA_DON
WHERE TEN_K ='Nga'))
```

Ví dụ Giả sử chắc chắn rằng “Hoa” chỉ có một đơn đặt hàng, khi đó có thể lấy được mã số đơn đặt hàng đó qua một câu vấn tin con và dùng nó để chọn tất cả các mặt hàng được đặt bởi “Hoa” từ quan hệ MH_SL. Trong trường hợp không biết chắc chắn “Hoa” đặt một đơn đặt hàng, mà có thể từ hai đơn đặt hàng trở lên thì câu vấn tin được trình bày dưới đây sẽ trả lời kết quả sai:

In tên các mặt hàng mà “Hoa” đặt mua (trường hợp chỉ có một đơn đặt).

```
SELECT TEN_K
FROM MH_SL
WHERE O# =
(SELECT SHD#
FROM HOA_DON
WHERE TEN_K = "Hoa");
```

5.6.6 Các phép toán gộp nhóm

SQL cung cấp năm phép toán gộp nhóm thường dùng AVG, COUNT, SUM, MIN, MAX và các phép toán STDDEV và VARLENCE để tính độ lệch chuẩn và phương sai của một danh sách các số. Một câu lệnh SELECT – FROM – WHERE có thể in ra kết quả khi áp dụng một hay nhiều các phép toán gộp cho các thuộc tính của một quan hệ bằng cách đặt quan hệ này trong mệnh đề FROM và trong mệnh đề SELECT AVG(A), COUNT(A),...trong đó A là một thuộc tính. Nếu trước các thuộc tính A có thêm từ khóa DISTINCT thì sự trùng lặp bị loại bỏ trước khi phép toán được thực hiện.

Ví dụ

- a) *In và tính số dư trung bình trong quan hệ CUSTOMERS..*

```
SELECT AVG(TAI_KHOAN)
FROM KHACH_HANG.
```

- b) *In và tính số nhà cung cấp*

```
SELECT COUNT(DISTINCT TEN_CC) #TEN
FROM CUNG_CAP
#TEN là tên cột mới khi in ra kết quả.
```

- c) *Số các nhà cung cấp “rượu cuốc lủi”*

```
SELECT COUNT(DISTINCT TEN_CC) #TEN_CLUI
FROM CUNG_CAP
WHERE MAT_HANG ='Cuốc lủi';
```

5.6.7 Mệnh đề GROUP BY ... HAVING

Có thể phân hoạch các bộ của một quan hệ thành các nhóm tách biệt nhau và áp dụng các phép toán gộp cho các nhóm. Trong câu lệnh SELECT – FROM – WHERE, mệnh đề GROUP BY nhóm lại bởi một danh sách các thuộc tính của quan hệ cần nhóm và thoả một điều kiện nhóm HAVING:

```
GROUP    BY  A1, A2, ..., Ak  
HAVING   E
```

Phân hoạch quan hệ thành các nhóm sao cho hai bộ cùng trong một nhóm khi và chỉ khi chúng giống nhau ở mọi thuộc tính A₁, A₂, ..., A_k. Đề cho kết quả của câu vấn tin có nghĩa, các thuộc tính A₁, A₂, ..., A_k cũng phải xuất hiện trong mệnh đề SELECT mặc dù chúng có thể có những bí danh để in ra nếu cần.

Ví dụ In ra tất cả các mặt hàng và giá trung bình của từng nhóm

```
SELECT  MAT_HANG, AVG (GIA) AP  
FROM    CUNG_CAP  
GROUP   BY  MAT_HANG
```

Một mệnh đề WHERE có thể theo sau mệnh đề FROM, chỉ cần một tập con của các bộ khi tạo các nhóm. Có thể sắp xếp đề in ra một tập con các nhóm, độc lập với kiểu kiện chọn lọc trong mệnh đề WHERE trước khi xây dựng các nhóm.

Ví dụ In ra tên và giá trung bình các mặt hàng rượu được nhiều nhà cung cấp

```
SELECT  MAT_HANG, AVG (GIA) AP  
FROM    CUNG_CAP  
WHERE   MAT_HANG LIKE “%Rượu%”  
GROUP   BY  MAT_HANG  
HAVING  COUNT(*) > 1 ;
```

5.6.8 Các phép toán lưu trữ

1. Phép chèn:

Chèn thêm các bộ mới vào quan hệ, cấu trúc câu lệnh như sau:

```
INSERT  INTO  R  
VALUES  (a1,a2,...,an)
```

Trong đó R là tên quan hệ và r=(a₁,a₂,...,a_n) là một bộ mới cần chèn.

Ví dụ :

- a) Thêm thông tin “Thủy” cung cấp “Rượu Làng Vân”, nhưng chưa có giá.

```
INSERT  INTO  CUNG_CAP (TEN_CC, MAT_HANG)  
VALUES  (“Thủy”, “Rượu Làng Vân”);
```

- b) Tạo quan hệ NG_CC (TEN_CC, GIA) chứa tên người cung cấp “Rượu cuộc lũ”

```
INSERT  INTO  NG_CC (TEN_CC, GIA)  
SELECT  TEN_CC, GIA
```

```
FROM    CUNG_CAP  
WHERE   MAT_HANG = "Rượu cuộc lủ" );
```

2. **Phép xoá**

Cú pháp phép xoá như sau:

```
DELETE FROM  R  
WHERE  E
```

Trong đó R là tên quan hệ và E là biểu thức đại số. Nghĩa là xoá khỏi quan hệ R tất cả các bộ thoả điều kiện E.

Ví dụ

- a) *Xoá các đơn đặt hàng về "Rượu cuộc lủ"*

```
DELETE FROM  HOA_DON  
WHERE  O# IN  
      ( SELECT  SHD#  
        FROM    MH_SL  
        WHERE   MAT_HANG = "Rượu cuộc lủ" );
```

- b) *Xoá "Nga" không còn cung cấp "Rượu cuộc lủ" nữa*

```
DELETE FROM  CUNG_CAP  
WHERE  TEN_K = "Nga" AND MAT_HANG = "Rượu cuộc lủ";
```

3. **Phép sửa đổi**

Cú pháp phép sửa đổi thông tin như sau:

```
UPDATE  R  
SET     A1=a1, A2=a2,...,An=an  
WHERE  E
```

Trong đó R là tên quan hệ và E là biểu thức đại số. Nghĩa là trong quan hệ R, sửa giá trị các thuộc tính A_i bằng a_i, i=1,2,..., n nếu mỗi bộ thoả điều kiện E.

Ví dụ

- a) *Hiệu chỉnh giá của "Rượu Hà Nội" do "nhà máy Rượu Hà nội" cung cấp thành 15.000 đồng/chai*

```
UPDATE  CUNG_CAP  
SET     GIA = 15.000  
WHERE  TEN_CC = "Nhà máy rượu Hà nội"  
        AND MAT_HANG = "Rượu Hà nội";
```

- b) *Giảm 10% giá "Rượu Hà Nội" do "nhà máy Rượu Hà nội" cung cấp.*

```
UPDATE  CUNG_CAP  
SET     GIA = 0.9 * GIA  
WHERE  TEN_CC = "Nhà máy rượu Hà nội"  
        AND MAT_HANG = "Rượu Hà nội";
```

5.6.9 Tính đầy đủ của SQL

Phần này xem xét khả năng của ngôn ngữ SQL có thực hiện được các biểu thức đại số quan hệ như các ngôn ngữ thao tác dữ liệu khác hay không. Vì vậy chỉ cần trình bày SQL thực hiện năm phép toán cơ bản của đại số quan hệ.

Giả sử cho các quan hệ $R(A_1, A_2, \dots, A_n)$ và $S(B_1, B_2, \dots, B_m)$. Trong trường hợp lấy hợp, hoặc hiệu của R và S , nếu bậc của R và S không giống nhau, tức là $n \neq m$, có thể đặt lại tên cho các thuộc tính của S , tạo ra một quan hệ mới TAM với các thuộc tính như các thuộc tính của quan hệ R :

```
INSERT INTO TAM
SELECT *
FROM S
```

Không mất tính tổng quát, giả sử $n = m$ và $A_i = B_i, i=1,2,\dots,n$.

- *Phép hợp:* $T = R \cup S$

```
INSERT INTO T
SELECT *
FROM R
INSERT INTO T
SELECT *
FROM S
```

- *Phép trừ:* $T = R - S = (R \cup S) - S$

```
DELETE FROM T
      (SELECT (A1,A2,...,An) IN
      FROM S);
```

- *Phép giao:* $T = R \cap S = R - (R - S) = R - T_2$

```
DELETE FROM T_2
      (SELECT (A1,A2,...,An) IN
      FROM S);
DELETE FROM T
      (SELECT (A1,A2,...,An) IN
      FROM T_2);
```

- *Tích Đề Các:* $T = R \times S$

```
INSERT INTO T
SELECT R.A1,...,R.An, S.B1,...,S.Bn
FROM R,S
```

- *Phép chọn:* $T = \sigma_F(R)$

```
INSERT INTO T
SELECT *
FROM R
```

WHERE E;

- *Phép chiếu* $T = \pi_{A_1, A_2, \dots, A_k}(R)$

INSERT INTO T

SELECT A₁, A₂, ..., A_k

FROM R;

5.6.10 Ngôn ngữ định nghĩa dữ liệu

- *Lệnh tạo, xóa một quan hệ*: Lệnh tạo một quan hệ cho phép mô tả tên bảng và xác định các phần tử của bảng ứng với các cột và các ràng buộc giữa các cột. Cú pháp có dạng tổng quát như sau:

CREATE TABLE <tên bảng> (<Các phần tử của bảng>)

Ví dụ: Tạo cấu trúc quan hệ TUYEN (C#, TC, DAI)

```
CREATE TABLE TUYEN
(C# CHAR(5) NOT NULL
TC CHAR(15),
DAI INT);
```

Ngược với lệnh tạo là lệnh xóa quan hệ: Xóa quan hệ CUNG_CAP ra khỏi CSDL có thể viết: DROP TABLE CUNG_CAP

- *Tạo và xóa chỉ mục* Lệnh tạo chỉ mục: Chỉ mục được sử dụng để làm tăng tốc độ truy xuất dữ liệu. Nếu quan hệ R có chỉ mục trên thuộc tính A thì có thể truy xuất tất cả các bộ có giá a cho trước của thuộc tính A theo thời gian tỷ lệ với số lượng các bộ:

```
CREATE INDEX I
ON R(A);
```

Loại bỏ chỉ mục I ra khỏi quan hệ R nhưng không làm ảnh hưởng đến dữ liệu trong quan hệ R, cú pháp lệnh như sau:

```
DROP INDEX I
```

5.6.11 Khung nhìn View

Trong SQL có nhóm phép toán thực hiện chức năng của ngôn ngữ định nghĩa dữ liệu lược đồ con (subschema DDL), hay là chức năng định nghĩa *khung nhìn* (view) không tồn tại trong bộ nhớ nhưng có thể truy vấn. Cú pháp của lệnh như sau:

```
CREATE VIEW V(A1, A2, ..., An) AS Q;
```

Trong đó V là tên của khung nhìn, A₁, A₂, ..., A_n là các thuộc tính của khung nhìn và Q là câu văn tin của khung nhìn.

Ví dụ

- a) *Tạo khung nhìn gồm các mặt hàng và giá của nhà cung cấp "Nga".*

```
CREATE VIEW NGA_VIEW(MAT_HANG, GIA) AS
SELECT MAT_HANG, GIA
FROM CUNG_CAP R
WHERE TEN_="Nga";
```

b) Tạo khung nhìn khách hàng đặt các đơn hàng .

```
CREATE VIEW DH(TEN_K, NGAY, MAT_HANG,  
                SO_LUONG) AS  
SELECT      TEN_K, NGAY, MAT_HANG, SO_LUONG  
FROM        HOA_DON, MH_SL  
WHERE       HOA_DON.SHD# = MH_SL.SHD# ;
```

Trong câu lệnh trên, biến đổi thuộc tính TEN_K của HOA_DON thành TEN_K trong khung nhìn DH, vì lệnh cho phép xác định các thuộc tính cho khung nhìn một cách tường minh.

Cuối cùng, nếu muốn hủy bỏ khung nhìn, cú pháp như sau:

```
DROP VIEW    V
```

Câu lệnh này không có tác dụng trên CSDL nhưng những ' câu vắn tm trên khung nhìn V không còn được chấp nhận nữa.

5.6.12 Database catalog

Có thể lấy thông tin về CSDL bằng cách sử dụng các câu vắn tin tham chiếu đến các danh mục của cơ sở dữ liệu. Các danh mục có thể được coi như là những quan hệ. Có bốn database catalog (danh mục của CSDL): TABLES, VIEWS, INDEXES, và COLUMNS.

- Một số danh mục của khung nhìn VIEWS:
- VIEWS\$TEXT: Hiển thị chuỗi.
- VEW\$: Ngày tháng tạo khung

.....

Ví dụ

a) Tìm định nghĩa của khung nhìn NGA_VIEW

```
SELECT VIEWS$TEXT  
FROM   VIEWS[NGA_VIEW]
```

Tên của khung nhìn được chỉ ra trong dấu ngoặc vuông [NGA_VIEW]. Kết quả của câu vắn tin trên sẽ là chuỗi:

```
SELECT MAT_HANG, GIA FROM CUNG_CAP WHERE TEN_K ="Nga";
```

b) Xác định ngày tháng tạo khung nhìn NGA_VIEW

```
SELECT VEW$TIME  
FROM   VIEWS[NGA_VIEW]
```

- Để xác định thông tin về các thuộc tính của một quan hệ hoặc một khung nhìn, có thể vắn tin danh mục COLUMNS. Một số các thuộc tính của danh mục COLUMNS:
 - COL\$NAME: tên của thuộc tính.
 - COL\$LD: vị trí của thuộc tính trong danh sách các thuộc tính của quan hệ.
 - COL\$DATATYPE: kiểu dữ liệu của thuộc tính.
 - COL\$LENGTH: số byte hoặc số chữ.số thập phân, tương ứng với kiểu ký tự và kiểu số.
 - COL\$SCALE: số chữ số sau dấu chấm thập phân.

- COL\$NULL: cho biết có được phép sử dụng giá trị null trong cột.

Ví dụ

- a) *Xác định tên, kiểu và vị trí các thuộc tính của khung nhìn NGA_VIEW*

```
SELECT COL$NAME, COL$ID, COL$DATATYPE  
FROM COLUMNS[NGA_VIEW]
```

COL\$NAME	COL\$ID	COL\$DATATYPE
MAT_HANG	1	CHAR
GIA	2	NUMBER

- a) *Xác định ngày tháng tạo ra quan hệ SUPPLIES*

```
SELECT TAB$TIME  
FROM TABLES[SUPPLIES];
```

- Sử dụng danh mục INDEXES có thể vẫn tìm để tìm ra thông tin về những chỉ mục khai báo đối với một quan hệ cho trước. Một số danh mục của INDEXES:
 - IDX\$NAME: tên của chỉ mục.
 - IDX\$COLUMN: thuộc tính được chỉ mục.
 - IDX\$UNIQUE: thuộc tính IDX\$COLUMN có được sử dụng như là một khóa của quan hệ hay không.

Ví dụ Chỉ mục O#_INDEX được tạo như sau

```
SELECT INDEX O#_INDEX  
ON HOA_DON(O#)
```

Khi đó có thể truy vấn chỉ mục O#_INDEX được tạo ra như thế nào:

```
SELECT IDX$NAME, IDX$COLUMN, IDX$UNIQUE  
FROM INDEXES(HOA_DON);
```

IDX\$NAME	IDX\$COLUMN	IDX\$UNIQUE
O#_INDEX	O#	Non Unique

Câu hỏi trắc nghiệm

1. Ngôn ngữ thao tác dữ liệu là
 - A. Một tập các phép toán thao tác trên các hệ cơ sở dữ liệu.
 - B. Biểu diễn dữ liệu một cách thống nhất.
 - C. Sự thống nhất tương ứng trong các phép toán thao tác
 - D. Thông tin được biểu diễn bằng một và chỉ một cách duy nhất
2. Ngôn ngữ định nghĩa dữ liệu - DDL (Data Definition Language).
 - A. Được đặc tả bằng một phần của hệ quản trị cơ sở dữ liệu.
 - B. Được đặc tả bằng một ngôn ngữ dữ liệu
 - C. Được đặc tả bằng một ngôn ngữ, một phần của hệ quản trị cơ sở dữ liệu.
 - D. Được đặc tả bằng cách chương trình ứng dụng

3. Ngôn ngữ thao tác dữ liệu DML (Data Manipulation Language)
 - A. Các toán hạng và kết quả của các phép toán là các quan hệ
 - B. Biểu diễn dữ liệu một cách thống nhất.
 - C. Được đặc tả bằng một ngôn ngữ dữ liệu
 - D. Được đặc tả bằng cách chương trình ứng dụng
 4. Nhóm các phép toán lưu trữ bao gồm
 - A. Chèn thêm , xoá, sửa đổi
 - B. Chèn thêm , xoá,
 - C. Phép chọn SELECT, chèn thêm
 - D. Phép chọn SELECT, phép chiếu PROJECT, phép kết nối JOIN
 5. Nhóm các phép toán tìm kiếm gồm:
 - A. Phép chọn SELECT, phép chiếu PROJECT, phép kết nối JOIN
 - B. Chèn thêm , xoá, sửa đổi
 - C. Chèn thêm , xoá,
 - D. Phép chọn SELECT, chèn thêm
 6. Phép chiếu:
 - A. $\sigma_E(R) = \{t \mid t \in R \ \& \ t[E] = \text{“True”}\}$
 - B. $\pi_{A_1, A_2, \dots, A_k}(R) = \{t[X] \mid t \in R \ \text{và} \ X = (A_1, A_2, \dots, A_k)\}$
 - C. $R_1 \times R_2 = P(\Omega \Sigma) := \{t \mid t[\Omega] \in R_1 \ \& \ t[\Sigma] \in R_2\}$
 - D. $R_1 \cap R_2 = \{t \mid t \in R_1 \ \text{And} \ t \in R_2\}$
 7. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Đại số quan hệ là một trong những ngôn ngữ thao tác dữ liệu, bao gồm các phép toán trên các quan hệ. Đó là các phép toán Tập hợp các phép toán quan hệ tạo nên một cơ chế truy nhập dữ liệu khá linh hoạt và mềm dẻo.
 - A. Hợp, giao, chiếu, chọn và kết nối.
 - B. Chèn thêm , xoá, sửa đổi
 - C. Chọn SELECT, chiếu PROJECT, kết nối JOIN
 - D. Hợp, giao, chiếu và chọn.
 8. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: R1 và R2 là các quan hệ khả hợp trên cùng tập các thuộc tính của Ω . Khi đó hợp của chúng là trên Ω .
 - A. Một quan hệ
 - B. Một phụ thuộc hàm
 - C. Một nhóm các bản ghi
 - D. Là tập con các thuộc tính
 9. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: R1 và R2 là các quan hệ khả hợp trên cùng tập các thuộc tính của Ω . Khi đó giao của chúngtrên Ω ,
 - A. Một quan hệ
 - B. Một phụ thuộc hàm
 - C. Một nhóm các bản ghi
-
-

D. Là tập con các thuộc tính

10. Hãy so khớp thông tin 2 phần dưới đây với nhau:

Phần 1

- 1 Kết quả các phép toán
- 2 Các toán hạng
- 3 Thay đổi thứ tự các bộ trong bảng
- 4 Các phép hợp, giao và kết nối trong đại số quan hệ.....
 - a. 1- D; 2- C; 3 – B; 4 – A .
 - b. 1- C; 2- B; 3 – A; 4 – D
 - c. 1- A; 2- B; 3 – C; 4 – D.
 - d. 1- D; 2- B; 3 – A; 4 – C.

Phần 2

- A vẫn đảm bảo tính độc lập và tính toàn vẹn dữ liệu.
- B. là các quan hệ giao hoán với nhau
- C. là một quan hệ
- D. là các phép toán có tính chất giao hoán

11. Hãy so khớp thông tin 2 phần dưới đây với nhau:

Phần 1

- 1 Các phép hợp, giao, kết nối và tích Đề các trong đại số quan hệ
- 2 Phép chọn- SELECT
- 3 Một câu hỏi có thể được biểu diễn bởi
- 4 Câu hỏi truy vấn có thể biểu diễn bằng các phép toán.....
 - a. 1- D; 2- C; 3 – B; 4 – A .
 - b. 1- C; 2- B; 3 – A; 4 – D
 - c. 1- A; 2- B; 3 – C; 4 – D.
 - d. 1- B; 2- D; 3 – A; 4 – C.

Phần 2

- A biểu thức quan hệ hay một cây phép toán quan hệ
- B. là các phép toán có tính chất kết hợp
- C. đại số quan hệ, có thể biểu diễn bằng cây đại số quan hệ.
- D. tạo ra một quan hệ kết quả

Câu hỏi & bài tập

1. Câu hỏi

1. Trình bày khái niệm ngôn ngữ thao tác dữ liệu.
2. Các phép toán đại số quan hệ.
 - a. Phép hợp các quan hệ – UNION
 - b. Phép giao của các quan hệ – INTERSECT.
 - c. Hiệu của các quan hệ - MINUS.
 - d. Tích Đề Các – (Decalesian Product)
 - e. Phép chọn – Selection.
 - f. Phép chiếu – PROJECT.
3. Các phép toán đại số bổ sung.
 - a. Phép chia - DIVISION.

- b. Phép kết nối – JOIN.
- 4. Các tính chất của đại số quan hệ.
 - a. Tính chất giao hoán, kết hợp.
 - b. Tính chất tích lũy đẳng, một số tính chất khác.
- 5. Đại số quan hệ – ngôn ngữ vấn tin.
- 6. Các thuật toán cài đặt đại số quan hệ.
 - a. Thuật toán hợp , giao.
 - b. Thuật toán trừ.
 - c. Thuật toán chọn, chiếu.
 - d. Thuật toán kết nối tự nhiên.
 - e. Thuật toán phép chia.
- 7. Ngôn ngữ vấn tin SQL.
- 8. Câu lệnh SELECT.
 - a. So sánh mẫu LIKE.
 - b. Phép toán tập hợp trong mệnh đề WHERE.
 - c. Các phép toán gộp nhóm.
 - d. Gộp theo nhóm.
- 9. Các phép toán lưu trữ
 - a. Phép chèn.
 - b. Phép xoá.
 - c. Phép sửa đổi.
- 10. Tính đầy đủ của SQL
- 11. Ngôn ngữ định nghĩa dữ liệu
- 12. Khái niệm ngôn ngữ hướng đối tượng

2. Bài tập

Bài 1: Giả sử trong nghiệp vụ quản lý phát hành báo chí, có các thông tin quản lý tác nghiệp như sau

MK#	: Mã hiệu khách hàng đặt mua báo.	TK	: Họ và tên khách hàng.
DC	: Địa chỉ khách hàng.	SĐT	: Số điện thoại.
B#	: Mã báo khách đặt mua.	TB	: Tên báo (tạp chí).
GIA	: Giá báo.	NG	: Ngày khách đặt báo, chí.
SL	: Số lượng báo, chí khách đặt mua.		

- 1. Tách về các quan hệ có dạng chuẩn cao, không tồn thất thông tin.
- 2. Thông tin về tiền khách đặt mua.
- 3. Có bao nhiêu người mua các loại báo.
- 4. Xác định thông tin về tất cả khách hàng đặt mua báo NDA (Nhân Dân) trong quý I/2000.(trong tất cả 4 quý).
- 5. Thông tin về các loại báo khách hàng đặt mua trong quý I/2000.

6. Tính tổng số lượng và tiền các loại báo(tạp chí) khách đặt mua trong từng quý năm 1999.
7. Tính tổng số lượng và tiền các loại báo(tạp chí) khách đặt mua trong các tháng năm 1999.
8. Hãy xác định thông tin về tất cả khách hàng đặt mua tất cả các loại báo có trong danh mục.
9. Lập bảng kê tính tổng tiền các loại báo mà khách đặt mua trong từng tháng?
10. Lập bảng kê tính tổng tiền các loại báo mà khách đặt mua trong từng quý năm 1999?
11. Thông tin về các loại báo khách đặt mua có số lượng và tiền cao hơn số lượng và tiền khách đặt mua báo “VHT” văn hoá&thể thao?
12. Thống kê số lượng các loại báo & tiền khách đặt mua từng quận.
13. Cho biết tháng nào có số lượng khách đặt mua báo Nhân dân cao hơn tháng 3 hàng năm.
14. Hãy xác định thông tin về các loại báo được đặt mua nhiều nhất
15. Những loại báo nào mà tất cả khách hàng đặt mua.

Bài 2: Thông tin về cuộc đàm thoại:

T# : Mã tổng đài.	TD : Tên gọi tổng đài.
DV: Đơn vị quản lý.	DL : Dung lượng.
B# : Mã thuê bao	TB : Tên thuê bao.
DC: Địa chỉ thuê bao.	SM : Số máy thuê bao.
NG1: Ngày lắp đặt.	DT : Đối tượng thuê bao.
NG2: Ngày gọi.	SĐT : Số máy đến.
BD: Giờ bắt đầu.	KT : Giờ kết thúc

1. Thông tin về các loại tổng đài.
2. Thông tin về các đối tượng khách hàng .
3. Thông tin về các hướng trong tháng.
4. Thông tin về các đơn vị quản lý.
5. Thông tin về các loại tổng đài có cùng dung lượng.
6. Thông tin đối tượng sử dụng điện thoại theo từng Quận.
7. Có bao nhiêu cuộc gọi đi từ một tổng đài.
8. Có bao nhiêu cuộc gọi nội hạt, liên tỉnh, quốc tế.
9. Có bao nhiêu cuộc gọi liên tỉnh , quốc tế bị chèn cước.
10. Sản lượng phút theo từng tổng đài.
11. Sản lượng phút theo từng hướng.
12. Sản lượng phút theo từng tháng.
13. Sản lượng phút theo hướng/ tháng.
14. Sản lượng phút theo tháng/hướng.
15. Sản lượng phút theo đối tượng.

16. Hướng nào thường xuyên (tháng) xảy ra chệch cước.
17. Hướng nào trong năm không có hiện tượng chệch cước.
18. Thống kê tất cả thuê bao có điện thoại trùng nhau.
19. Liệt kê các thuê bao tháng nào cũng sử dụng điện thoại.
20. Liệt kê các thuê bao trong năm không một lần sử dụng gọi đi liên tỉnh, quốc tế.
21. Những thuê bao nào có số cuộc gọi trên 100 cuộc tháng.
22. Thuê bao nào có số phút gọi trên 1000 phút/tháng.
23. Số máy nào hàng tháng đều có các cuộc gọi đến.
24. Số máy nào trong một tháng nhận 100 cuộc.
25. Tính số cuộc gọi đến cho từng thuê bao đến.
26. Tính sản lượng phút tổng đài/quý.
27. Tổng đài nào có số cuộc gọi, sản lượng phút / tháng cao hơn tổng đài “A” do Cty điện thoại quản lý.
28. Địa bàn nào thường có các cuộc gọi cao nhất.
29. Tổng số cuộc gọi qua tổng đài.
30. Tổng số cuộc gọi tháng
31. Tổng số cuộc gọi đối tượng
32. Tổng số cuộc gọi Quý.
33. Số máy nào có mật độ gọi cao.
34. Thống kê các thuê bao hàng tháng sử dụng 5 cuộc gọi.

Bài 3: Quản lý mạng cáp

- TC# : Mã tuyến T : Tên gọi tuyến
DAI : Độ dài tuyến cáp. MC# : Mã cáp.
MC : Tên gọi cáp GIA : Giá cáp
35. SL: Số lượng cáp lắp đặt NSX : Nước sản xuất
1. Giá trị các loại cáp được lắp đặt.
 2. Giá trị các loại cáp chưa được lắp đặt.
 3. Giá trị các loại cáp được lắp đặt trong từng năm.
 4. Giá trị các loại cáp Việt nam được lắp đặt trên từng tuyến.
 5. Giá trị các loại cáp Việt nam chưa được lắp đặt.
 6. Các loại cáp có tổng giá trị lắp đặt trên 5 tỷ đồng VN.
 7. Các loại cáp có tổng giá trị trên 5 tỷ đồng chưa được lắp đặt.
 8. Tổng giá trị lắp đặt trên từng tuyến.
 9. Loại cáp nào được lắp đặt trên tuyến nhưng trong các kho không còn.
 10. Tuyến nào có tất cả các loại cáp như trong kho có.
 11. Tuyến nào có tổng giá trị lắp đặt lớn hơn tuyến Hà Nội-Quảng ninh.

6 TỐI ƯU HOÁ CÂU HỎI TRUY VẤN

Tối ưu hóa câu hỏi là quá trình lựa chọn phương pháp sao cho khi thực hiện các câu hỏi truy vấn có hiệu quả nhất, có thể đánh giá được các khả năng xử lý câu hỏi từ nhiều chiến lược khác nhau, đặc biệt là cho những câu hỏi phức tạp. Một phương pháp khi thực hiện có chi phí thấp, tức là tối ưu về thời gian truy xuất thông tin và tối ưu về không gian lưu trữ mà vẫn bảo đảm được tính độc lập và toàn vẹn dữ liệu. Nội dung của chương gồm các vấn đề:

- Tổng quan về tối ưu hóa câu hỏi.
- Kỹ thuật tối ưu hoá câu hỏi truy vấn.
- Phương pháp khung nhìn

6.1 Mở đầu

Ngôn ngữ con dữ liệu là một tập các phép toán có cấu trúc quan hệ được dùng để thao tác trên các dữ liệu dưới dạng quan hệ. Kết quả của các phép toán được biểu diễn duy nhất bằng một quan hệ. Các thuộc tính và các bộ của nó được rút ra từ một hay nhiều quan hệ cho trước, vì vậy trước khi thực hiện các câu hỏi cần thiết phải biến đổi hợp lý sao cho chi phí thời gian thực hiện các phép toán là ít nhất. Quá trình biến đổi này được gọi là quá trình tối ưu hoá các câu hỏi truy vấn dữ liệu.

Tối ưu hoá câu hỏi rất có ý nghĩa khi tìm kiếm và truy xuất dữ liệu tại các thiết bị đầu cuối. Nâng cao hiệu suất các phiên làm việc của người sử dụng. Không những tối ưu về thời gian truy xuất thông tin mà là nhân tố quan trọng trong việc bảo đảm tính độc lập của dữ liệu với các chiến lược tìm kiếm, truy xuất dữ liệu. Có nhiều kỹ thuật khác nhau để tối ưu hoá các câu hỏi, trong đó có kỹ thuật phân rã câu hỏi dựa trên ngôn ngữ của đại số quan hệ.

6.2 Tổng quan về tối ưu hóa câu hỏi

Xét cơ sở dữ liệu quản lý mạng cáp gồm các quan hệ sau:

- Thông tin về các tuyến cáp: TUYEN (T#, TC, DAI).
- Thông tin về các loại cáp: CAP (M#, MC, GIA, NSX).
- Thông tin về các loại cáp đã được lắp đặt: L_DAT (T#, M#, SL, NG)

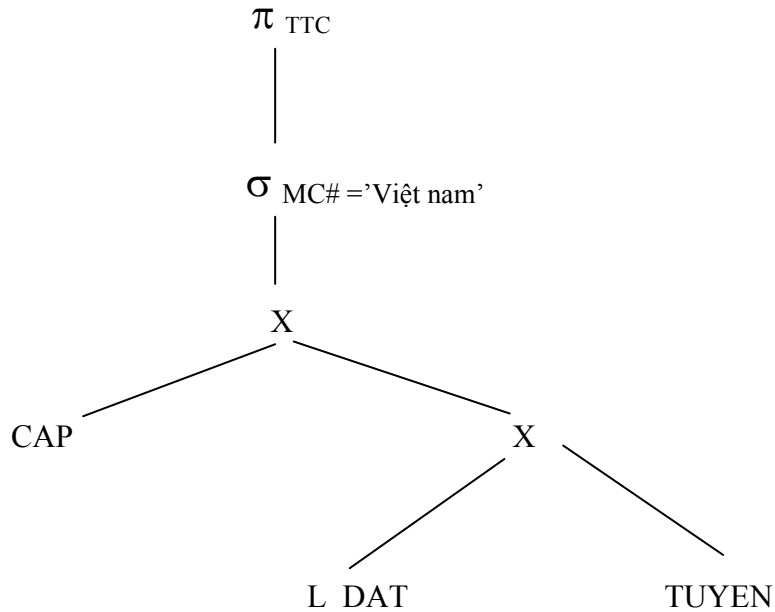
Trong đó:

T#	ký hiệu mã tuyến	TC	Tên gọi tuyến cáp,
DAI	độ dài tuyến cáp	M#	Mã cáp
MC	Tên gọi cáp	GIA	Đơn giá.
NSX	Nước sản xuất	SL	Số lượng cáp lắp đặt.
NG	Ngày lắp đặt cáp.		

Để có được cách nhìn tổng quát của việc tối ưu hoá câu hỏi, xét ví dụ sau: *liệt kê tên tất cả các tuyến cáp có lắp đặt cáp Việt nam.*

Biểu thức đại số quan hệ và cây đại số quan hệ được biểu diễn như sau :

$$\pi_{TTC} (\sigma_{MC\#='Việt\ nam'} (TUYEN \times L_DAT) \times CAP))$$



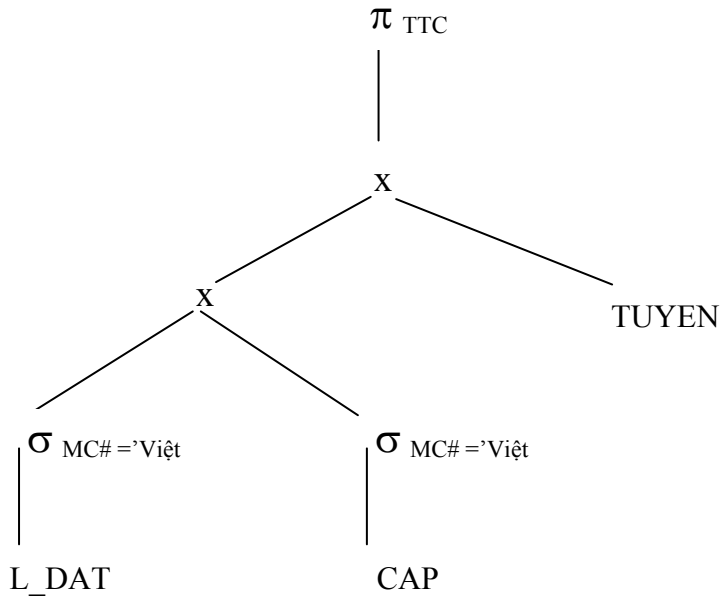
Hình 6.1 Cây đại số quan hệ chưa tối ưu hoá

Biểu thức trên mô tả quá trình thực hiện câu hỏi như sau: Trước hết, biểu thức tạo ra quan hệ TUYEN X L_DAT X CAP trung gian, bằng cách kết nối tự nhiên hoặc thực hiện tích Đề các các quan hệ TUYEN, L_DAT và CAP với một số lượng bản ghi bằng tích số lượng các bản ghi và các thuộc tính bao gồm các thuộc tính của 3 quan hệ trên. Như vậy sau bước này, quan hệ tạm có lực lượng khá lớn và chiều dài của một bản ghi cũng không phải là nhỏ, nếu kích cỡ các quan hệ nguồn lớn. Tuy nhiên kết quả câu hỏi chỉ quan tâm đến thuộc tính tên tuyến cấp TTC và các bộ dữ liệu có tên loại cấp trùng với “Việt nam”. Như vậy sau khi tính tích Đề các, hệ thống sẽ thực hiện các phép chọn và phép chiếu để đưa ra quan hệ kết quả. Trong cách biểu diễn này, số lượng các phép toán thực hiện tương đối lớn, chi phí thời gian quá cao và chi phí lưu trữ tốn kém.

Chú ý rằng, câu hỏi chỉ quan tâm đến các bộ dữ liệu cấp “Việt nam”. Thông tin này được chứa trong các quan hệ L_DAT, và CAP (trong nhật ký lắp đặt là mã các loại cấp Việt nam và trong danh mục là chuỗi “Việt nam”). Như vậy có thể thực hiện các phép chọn trong các quan hệ L_DAT và CAP. Kết quả trung gian các phép chọn sẽ làm giảm đáng kể số lượng các bản ghi, kích cỡ của quan hệ của phép kết nối tự nhiên của 2 thành phần trung gian trên với quan hệ TUYEN. Càng dễ dàng khi thực hiện phép chiếu trên thuộc tính tên tuyến cấp TTC. Có thể biến đổi biểu thức đại số ban đầu tối ưu hơn, bằng cách thực hiện các phép chiếu và chọn, tiếp sau mới thực hiện phép kết nối. Kết quả biến đổi như sau:

$$\pi_{TTC} (\sigma_{MC\#='Việt\ nam'} (L_DAT) \times (\sigma_{MC\#='Việt\ nam'} (CAP) \times TUYEN))$$

Biểu thức này tương đương như biểu thức đại số ban đầu, nhưng nó tạo ra các quan hệ trung gian có kích cỡ nhỏ hơn nhiều so với kích cỡ của TUYEN X L_DAT X CAP.

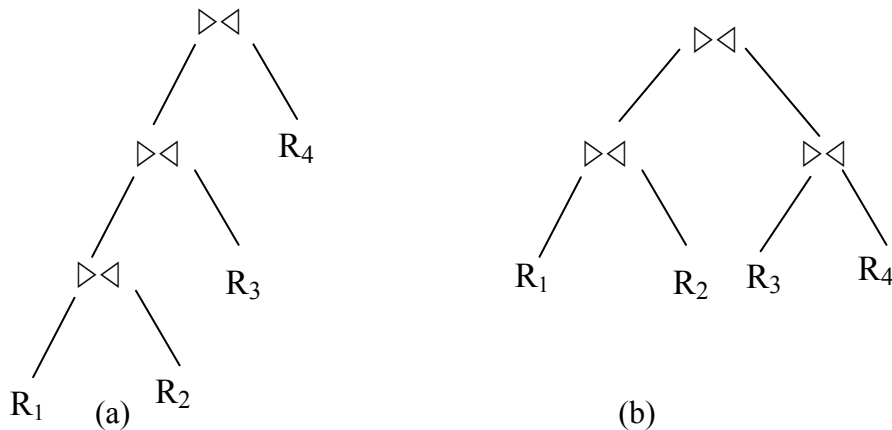


Hình 6.2 Cây đại số được tối ưu hóa

Như vậy, biến đổi một biểu thức đại số quan hệ về một biểu thức đại số quan hệ tương đương tối ưu theo nghĩa sẽ cho cùng một kết quả nhưng với chi phí thời gian thực hiện và sử dụng bộ nhớ ít hơn rất nhiều

6.3 Phương pháp ước lượng cây đại số quan hệ

Có hai cách tiếp cận cơ bản để đánh giá tối ưu hóa biểu thức quan hệ. Thường là kết hợp các yếu tố của cả hai cách tiếp cận trên. Ví dụ các phương án ưu hóa câu hỏi R không quan tâm tới thứ tự kết nối khi các toán hạng bên phải của mỗi phép kết nối là một trong các quan hệ r_1, \dots, r_n , được gọi là *cây kết nối trái theo chiều sâu*. Cây kết nối trái theo chiều sâu rất có chi phí rất nhỏ. Hình 6.3 mô tả sự khác nhau giữa cây nối trái theo chiều sâu và cây không nối trái theo chiều sâu.



Hình 6.3 a) Cây kết nối theo chiều sâu về trái
b) Cây kết nối theo chiều sâu không về trái

6.4 Nguyên tắc tối ưu hoá

- Ưu tiên thực hiện các phép chiếu và chọn, nhằm giới hạn khối lượng dữ liệu trung gian. Giảm chi phí truy nhập bộ nhớ.
- Trước khi phải thực hiện phép tích Đề các, hãy tìm chiến lược truy nhập tốt nhất vào CSDL. Ví dụ như sử dụng các phép sắp xếp, hoặc chọn chỉ số trên thành phần tham gia vào tích Đề các.
- Thực hiện các phép kết nối cân bằng chi phí sẽ rẻ hơn nhiều so với chi phí thực hiện phép tích Đề các.
- Nhóm các phép toán chọn và chiếu liên tiếp thành một phép toán duy nhất.
- Nhóm các phép tích và chiếu liên tiếp thành một phép toán duy nhất. Trong khi thi thực hiện phép tích có thể giới hạn chi phí thực hiện bằng phép chiếu.
- Tìm biểu thức chung trong một biểu thức. Nếu kết quả là một quan hệ không lớn lắm nhưng tần xuất xuất hiện nhiều lần, nên có biểu thức con chung.
- Đánh giá sơ bộ trước khi thực hiện câu hỏi. Số phép toán thực hiện, tổng chi phí thực hiện: thời gian, bộ nhớ ...

Ví dụ Tên các loại cáp sử dụng trên tuyến “T03”

a) Ngôn ngữ đại số quan hệ

Cách 1: $T_1 := L_DAT \bowtie_{L_DAT.M\# = CAP.M\#} CAP$

$T_2 := \sigma_{T\# = "T03"}(T_1)$

$KQ := \pi_{M\#, MC}(T_2)$

$KQ := \pi_{M\#, MC}(\sigma_{T\# = "T03"}(L_DAT \bowtie_{L_DAT.M\# = CAP.M\#} CAP))$

Cách 2: $T_1 := \sigma_{T\# = "T03"}(L_DAT)$

$T_2 := CAP \bowtie_{CAP.M\# = T_1.M\#} T_1$

$KQ := \pi_{M\#, MC}(T_2)$

$KQ := \pi_{M\#, MC}(CAP \bowtie_{CAP.M\# = T_1.M\#}(\sigma_{T\# = "T03"}(L_DAT)))$

b) Biểu diễn bằng ngôn ngữ SQL:

```
CREATE VIEW KQ AS
    SELECT R.M#, R.MC
    FROM CAP R, L_DAT S
    WHERE S.T# = "T03" and R.M# = S.M#
SELECT *
FROM KQ
```

Như vậy với cách biểu diễn trên, câu hỏi đã được phân rã thành 2 câu hỏi con theo chiều hướng ít tốn kém hơn.

6.5 Kỹ thuật tối ưu hoá các biểu thức đại số quan hệ.

6.5.1 Biểu thức quan hệ

Biểu thức quan hệ là một biểu thức mà các toán hạng là các quan hệ trong một CSDL và các phép toán là các phép toán trong đại số quan hệ. Biểu thức E_1 và E_2 tương đương với

nhau, ký hiệu là $E_1 \cong E_2$, nếu chúng biểu diễn cùng một ánh xạ, nghĩa là các quan hệ giống nhau trong biểu thức, thì kết quả cũng giống nhau.

Khi thực hiện các phép toán trong một biểu thức quan hệ, các phép toán một ngôi có thứ tự ưu tiên cao hơn so với phép toán hai ngôi. Có thể xem phép chiếu được ưu tiên hơn so với phép hợp, hoặc phép trừ. Tùy từng hệ cụ thể mà có thể có những quy định về thứ tự ưu tiên. Ví dụ trong một hệ có qui định phép giao có thứ tự ưu tiên trước phép hợp hai quan hệ.

Mỗi biểu thức quan hệ thông thường là một câu hỏi của người sử dụng. Hay nói cách khác, người sử dụng thể hiện nội dung câu hỏi để truy nhập vào CSDL thông qua một biểu thức quan hệ nào đó. Khi một câu hỏi đã được đưa vào hệ thống, thì hệ thống cần thực hiện một số các chức năng tiếp theo: Kiểm tra chức năng của biểu thức E. Nếu biểu thức E không có lỗi, thì thực hiện việc tối ưu hoá quá trình thực hiện E.

Tối ưu hoá là xác định trật tự thực hiện các phép toán trong E nhằm đáp ứng các nhu cầu: cho cùng kết quả cho việc thực hiện theo trật tự ban đầu của biểu thức E, giảm bớt được miền nhớ và thời gian thực hiện. Cuối cùng thực hiện biểu thức đã được tối ưu hoá E.

Khi các quan hệ càng nhỏ thì thời gian chi phí cần thiết để thực hiện các thao tác trên quan hệ càng thấp. Quan hệ nhỏ hiểu theo nghĩa là lực lượng và bậc của quan hệ nhỏ. Đối với các phép toán quan hệ thì phép chọn sẽ làm giảm kích thước chiều dọc, phép chiếu làm giảm kích thước chiều ngang. Các phép toán này thường hay được sử dụng trong các biểu thức quan hệ. Phép chia cũng có tác dụng làm nhỏ quan hệ, nhưng tần số sử dụng không nhiều. Nhận thấy rằng muốn tối ưu hoá một biểu thức quan hệ, chuyển đổi thích hợp trật tự các phép toán quan hệ sao cho các phép chiếu, chọn có thể thực hiện càng sớm càng tốt và thực hiện phép kết nối càng muộn càng tốt.

6.5.2 Biến đổi biểu thức quan hệ

Hai biểu thức đại số quan hệ được gọi là tương đương nếu trên một mô hình dữ liệu bất kỳ, hai biểu thức này tạo ra hai quan hệ có tập các bộ dữ liệu giống nhau. Giống nhau theo nghĩa thứ tự của các bộ không nhất thiết phải tương ứng với nhau, có thể thứ tự khác nhau và thứ tự của các thuộc tính cũng không nhất thiết phải như nhau. Nói một cách khái quát, 2 quan hệ trùng nhau nếu 2 bộ thuộc tính là như nhau và bộ dữ liệu của quan hệ này cũng là bộ dữ liệu của quan hệ kia và ngược lại.

6.5.3 Các quy tắc tương đương

Nói rằng 2 biểu thức tương đương nếu thay thế một biểu thức của dạng thứ nhất bởi một biểu thức của dạng thứ hai, và ngược lại ta có thể thay thế biểu thức của dạng thứ hai bởi biểu thức của dạng thứ nhất thì hai biểu thức cùng tạo ra kết quả giống nhau trên bất kỳ hệ CSDL. Các quy tắc sau được sử dụng để biến đổi các biểu thức tương đương với nhau.

Ký hiệu :

- F_1, F_2, \dots là các điều kiện.
- L_1, L_2, L_3, \dots là tập các thuộc tính.
- E, E_1, E_2, \dots là các biểu thức đại số quan hệ.
- σ : phép chọn.
- Π : là phép chiếu.

Chương I: Khái niệm cơ bản về các hệ cơ sở dữ liệu

- $\triangleright\triangleleft_{F_i}$: phép kết nối có điều kiện.
- $\triangleright\triangleleft$: phép kết nối tự nhiên.
- \times : tích Đề các

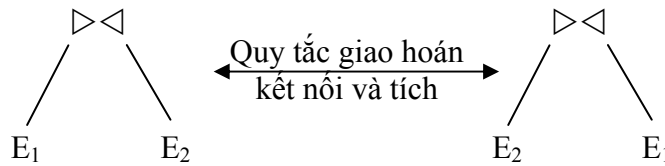
a) Các quy tắc cho các phép kết nối và tích Đề các

1. Quy tắc giao hoán kết nối và tích: Nếu E_1 và E_2 là các biểu thức quan hệ, F là một điều kiện trên các thuộc tính của E_1 và E_2 , thì:

$$E_1 \bowtie_F E_2 = E_2 \bowtie_F E_1$$

$$E_1 \bowtie E_2 = E_2 \bowtie E_1$$

$$E_1 \times E_2 = E_2 \times E_1$$



2. Quy tắc kết hợp phép kết nối và tích: Nếu E_1, E_2 và E_3 là các biểu thức quan hệ, F_1 và F_2 là các điều kiện thì:

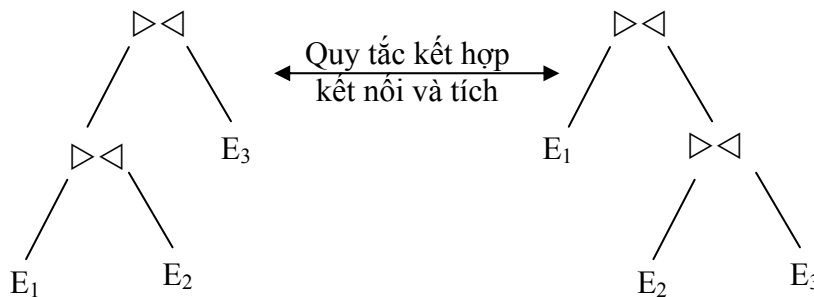
$$(E_1 \bowtie_{F_1} E_2) \bowtie_{F_2} E_3 = E_1 \bowtie_{F_1} (E_2 \bowtie_{F_2} E_3)$$

$$(E_1 \bowtie E_2) \bowtie E_3 = E_1 \bowtie (E_2 \bowtie E_3)$$

$$(E_1 \times E_2) \times E_3 = E_1 \times (E_2 \times E_3)$$

Định nghĩa quan hệ có 2 định nghĩa tương đương. Định nghĩa thứ nhất phát biểu là quan hệ là một tập con của tích Đề các của các thuộc tính, tức là quan hệ là tập các n -bộ. 2 quan hệ trùng nhau là 2 quan hệ có các bộ trùng nhau. Định nghĩa thứ hai nói rằng, quan hệ là một tập ánh xạ từ các thuộc tính vào tập các giá trị. Và 2 quan hệ trùng nhau nếu tập 2 ánh xạ như nhau. Các quy tắc liên quan đến phép kết nối và tích Đề các được sử dụng theo định nghĩa thứ hai, tập các ánh xạ. $E_1 \times E_2 = E_2 \times E_1$

sẽ đúng vì gọi R và S là các quan hệ có các thuộc tính có chứa tương ứng trong các biểu thức E_1, E_2 . Khi đó $R.A$ sẽ được hiểu là thuộc tính của quan hệ R , và $S.A$ cũng sẽ được hiểu là một thuộc tính của S . Gọi μ là một bộ trong $E_1 \times E_2$, khi đó tồn tại một bộ r của R và s là một bộ của S sao cho $\mu[R.A] = r[A]$ và $\mu[S.A] = s[A]$. Tương tự xét $E_2 \times E_1$ sẽ có một bộ t sao cho $t[R.A] = r[A]$ và $t[S.A] = s[A]$. Như vậy μ trùng với t . Suy ra $E_1 \times E_2 \subseteq E_2 \times E_1$. Hiển nhiên $E_1 \times E_2 = E_2 \times E_1$.



b) Các quy tắc cho các phép chọn và phép chiếu

1. *Nhóm các phép chiếu thành một phép chiếu duy nhất:* Nếu E là một biểu thức quan hệ và A_1, A_2, \dots, A_n là các thuộc tính có mặt trong B_1, A_2, \dots, B_k . Khi đó :

$$\pi_{A_1, A_2, \dots, A_n} (\pi_{B_1, A_2, \dots, B_k} (E)) = \pi_{A_1, A_2, \dots, A_n} (E).$$

Nghĩa là thực hiện các phép chiếu liên tiếp trên các thuộc tính của B_1, A_2, \dots, B_k , sau đó quan hệ kết quả lại được chiếu trên các thuộc tính của A_1, A_2, \dots, A_n .

2. *Nhóm các phép chọn thành một chuỗi phép chọn:* Nếu E là một biểu thức quan hệ và một điều kiện $F = F_1 \wedge F_2 \wedge \dots \wedge F_n$, khi đó:

$$\sigma_{F_1 \wedge F_2 \wedge \dots \wedge F_n} (E) = \sigma_{F_1} (\sigma_{F_2} (\dots (\sigma_{F_n} (E)) \dots))$$

3. *Giao hoán các phép chọn*

$$\sigma_{F_1} (\sigma_{F_2} (E)) = \sigma_{F_2} (\sigma_{F_1} (E))$$

- 4 *Giao hoán các phép chiếu và phép:*

Nếu điều kiện F chỉ chứa các thuộc tính A_1, A_2, \dots, A_n , khi đó

$$\pi_{A_1, A_2, \dots, A_n} (\sigma_F (E)) = \sigma_F (\pi_{A_1, \dots, A_n} (E)).$$

Nếu điều kiện F có các thuộc tính B_1, A_2, \dots, B_k không chứa các thuộc tính A_1, A_2, \dots, A_n , khi đó

$$\pi_{A_1, A_2, \dots, A_n} (\sigma_F (E)) = \pi_{A_1, A_2, \dots, A_n} (\sigma_F (\pi_{A_1, \dots, A_n, B_1, \dots, B_k} (E))).$$

- 5 *Giao hoán phép chọn và tích Đề các*

Nếu các thuộc tính có mặt trong điều kiện F là các thuộc tính của E_1 , khi đó:

$$\sigma_F (E_1 \times E_2) = \sigma_F (E_1) \times E_2$$

Nếu điều kiện $F = F_1 \wedge F_2$, F_1 chứa các thuộc tính của E_1 và F_2 chứa các thuộc tính E_2 , khi đó:

$$\sigma_F (E_1 \times E_2) = \sigma_{F_1} (E_1) \times \sigma_{F_2} (E_2)$$

Nếu $F = F_1 \wedge F_2$, F_1 chỉ chứa các thuộc tính của E_1 và F_2 chứa các thuộc tính của E_1 và E_2 , khi đó:

$$\sigma_F (E_1 \times E_2) = \sigma_{F_2} (\sigma_{F_1} (E_1) \times E_2)$$

- 6 *Giao hoán phép chọn và phép hợp*

Nếu biểu thức có dạng $E = E_1 \cup E_2$ và giả sử các thuộc tính của E_1 và E_2 có cùng tên với các thuộc tính của E, F là một điều kiện, khi đó:

$$\sigma_F (E_1 \cup E_2) = \sigma_F (E_1) \cup \sigma_F (E_2)$$

- 7 *Giao hoán phép chọn và phép trừ:*

$$\sigma_F (E_1 - E_2) = \sigma_F (E_1) - \sigma_F (E_2)$$

- 8 *Giao hoán phép chọn và phép kết nối tự nhiên*

Nếu điều kiện F chỉ chứa các thuộc tính chung biểu thức E_1 và E_2 , khi đó:

$$\sigma_F (E_1 \bowtie E_2) = \sigma_F (E_1) \bowtie_F (E_2)$$

Như vậy, phép chọn được đẩy xuống trong 2 nhánh cây biểu thức (expression tree).

Phép chọn làm giảm kích thước của quan hệ kết quả trong cả 2 nhánh.

- 9 *Giao hoán phép chiếu và phép tích Đề các*

Nếu E_1 và E_2 là các biểu thức, gọi A_1, A_2, \dots, A_n là danh sách các thuộc tính và B_1, B_2, \dots, B_k là các thuộc tính của biểu thức E_1 và các thuộc tính còn lại, ký hiệu là C_1, C_2, \dots, C_j là các thuộc tính của E_2 , khi đó:

$$\pi_{A_1, A_2, \dots, A_n}(E_1 \times E_2) = \pi_{B_1, B_2, \dots, B_k}(E_1) \times \pi_{C_1, C_2, \dots, C_j}(E_2)$$

$$\sigma_F(E_1 \bowtie E_2) = \sigma_F(E_1) \bowtie \sigma_F(E_2)$$

10. *Giao hoán phép chiếu và phép hợp*

$$\pi_{A_1, A_2, \dots, A_n}(E_1 \cup E_2) = \pi_{A_1, A_2, \dots, A_n}(E_1) \cup \pi_{A_1, A_2, \dots, A_n}(E_2).$$

11. *Kết hợp phép giao và phép hợp*

$$(E_1 \cup E_2) \cup E_3 = E_1 \cup (E_2 \cup E_3)$$

$$(E_1 \cap E_2) \cap E_3 = E_1 \cap (E_2 \cap E_3)$$

6.5.4 Một số nhận xét khi thực hiện tối ưu hoá câu hỏi

Tối ưu hoá câu hỏi dựa trên chi phí thấp là chủ yếu. Tuy nhiên nhiều khi thực hiện việc tối ưu cũng rất tốn kém. Qua kinh nghiệm, người ta rút ra một số nhận xét sau:

Thực hiện các phép chọn càng sớm càng tốt: Thường các phép chọn chỉ quan tâm đến các bộ thỏa một điều kiện nào đó, như vậy các bước tiếp theo chỉ thực hiện trên một quan hệ kích cỡ nhỏ hơn. Đây là một nhận xét kinh nghiệm, vì không phải lúc nào cũng cho chi phí giảm. Ví dụ biểu thức $\sigma_F(r \triangleright \triangleleft s)$, trong đó F có liên quan đến các thuộc tính của s. Phép chọn này có thể được thực hiện trước khi kết nối. Nếu r là vô cùng nhỏ so với s khi đó thực hiện phép chọn sớm sẽ hiệu quả cao. Phép chiếu cũng như phép chọn, làm giảm độ lớn của các quan hệ. Vì thế bất cứ lúc nào cũng cần tạo ra một quan hệ tạm thời, có lợi khi áp dụng ngay lập tức với bất cứ phép chiếu nào nếu có thể.

Thực hiện các phép chọn sớm hơn các phép chiếu thường tốt hơn, vì phép lựa chọn có khả năng làm giảm bớt độ lớn của các quan hệ và cho phép sử dụng các chỉ số để truy nhập cơ sở dữ liệu.

Sau đây là cách khái quát các bước trong một thuật toán tối ưu hoá kinh nghiệm tiêu biểu.

1. Tách các phép chọn liên kết thành một chuỗi phép chọn riêng lẻ. Bước này dựa trên qui tắc tương đương.
2. Chuyển các phép chọn xuống dưới cây đại số để thực hiện trước. Bước này sử dụng các tính chất giao hoán và tính chất phân phối của các phép chọn được thể hiện trong các qui tắc tương đương. Chẳng hạn, biến đổi $\sigma_\theta(r \triangleright \triangleleft s)$ thành $\sigma_\theta(r) \triangleright \triangleleft s$ hoặc $r \triangleright \triangleleft \sigma_\theta(s)$ mỗi khi có thể. Thực hiện các phép chọn dựa trên giá trị càng sớm càng giảm được chi phí của sự sắp xếp và sự hoà lẫn các kết quả trung gian.
3. Xác định các phép chọn và các phép kết nối sẽ đưa ra các biểu thức quan hệ nhỏ nhất, nghĩa là sẽ đưa ra số bộ dữ liệu ít nhất. Sử dụng tính chất kết hợp của phép toán $\triangleright \triangleleft$ để sắp xếp lại cây cho các quan hệ node-lá với sự lựa chọn thu hẹp này được thực hiện trước. Bước này dựa vào tính chất kết hợp của phép toán nhị phân đã cho của qui tắc tương đương. Lưu ý rằng, phép chọn bị hạn chế khi số các bản ghi không thỏa điều kiện, vì vậy cần phải cân nhắc tính chất của phép chọn hoặc điều kiện kết nối.

4. Có thể thay phép kết nối của kết quả theo tích Đề-các bằng một điều kiện chọn. Kết quả phép tích Đề-các thường tốn kém khi thực hiện vì khi nhân bao gồm một bản ghi cho mỗi kết nối của các bản ghi từ quan hệ A và quan hệ B. Phép chọn này có thể làm giảm đáng kể một số bản ghi trong quá trình thực hiện kết nối. Nó tốn kém hơn nhiều so với phép tích Đề-các.
5. Phá vỡ và chuyển càng nhanh càng tốt danh sách các thuộc tính trong phép chiếu xuống phía dưới cây, tạo ra phép chiếu ở nơi cần. Bước này dựa trên các đặc tính của phép chiếu đã qui định trong các qui tắc tương đương.
6. Nhận dạng các cây con mà các phép toán của nó thực hiện theo đường không rẽ nhánh và thực thi chúng.

Tóm lại các phép chọn sớm sẽ làm giảm bớt số bộ dữ liệu và phép chiếu sớm làm giảm số các thuộc tính.

6.6 Tối ưu hóa bằng khung nhìn (Materialized Views)

Thông thường, quan hệ kết quả được sinh ra bởi các câu hỏi truy vấn sẽ được lưu trữ trong các khung nhìn đã được định nghĩa. Ngược lại, khung nhìn cụ thể hóa là khung nhìn lưu trữ toàn bộ dữ liệu tính toán và kết quả tính toán. Trong nhiều ứng dụng mà các thao tác lặp lại nhiều lần, có thể sử dụng các kết quả của các thao tác trước cho các thao tác sau rất hiệu quả. Ví dụ nếu trước đây có câu hỏi truy vấn về tổng số từng loại cấp lắp đặt trên tuyến:

```
CREATE VIEW SO_LUONG (Loai_cap, SLG) AS
SELECT MC#, SUM(SL)
FROM L_DAT
GROUP BY MC#
```

Và nếu được lưu trữ trong khung nhìn cụ thể hóa thì những câu hỏi về số lượng một số loại cấp cụ thể nào đó sẽ tìm kiếm nhanh chóng, không mất nhiều thời gian.

6.6.1 Phép kết nối

Giả sử $V = R \bowtie S$. Giả sử, chèn thêm vào R một tập I_R . Nếu giá trị cũ của R được biểu thị bằng R^{old} , và giá trị mới của R bằng R^{new} , thì $R^{new} = R^{old} \cup I_R$. Khi đó, giá trị cũ của V^{old} được biểu diễn bởi $R^{old} \bowtie S$, và giá trị mới V^{new} được biểu diễn bởi $R^{new} \bowtie S$. Khi đó $R^{new} \bowtie S$ biểu diễn biểu thức tối ưu như sau:

$$(R^{old} \bowtie S) \cup (I_R \bowtie S). \text{ Tức là } V^{new} = V^{old} \cup (I_R \bowtie S).$$

Như vậy, khi thực hiện phép chèn thêm, nếu sử dụng phương pháp khung nhìn cụ thể hóa, thì việc thực hiện sẽ đơn giản hơn, chỉ cần thêm $I_R \bowtie S$ vào nội dung cũ. Tương tự, giả sử cần xóa một số bộ D_R ra khỏi quan hệ R , với khung nhìn cụ thể hóa, chỉ cần thực hiện $V^{new} = V^{old} - (D_R \bowtie S)$.

6.6.2 Phép chọn và phép chiếu

□ Xem xét phép chọn $V = \sigma_F(R)$:

- Chèn thêm tập các bộ I_R vào quan hệ R : $V^{new} = V^{old} \cup \sigma_F(I_R)$.

- Xóa khỏi R một tập các bộ D_R : $V^{new} = V^{old} - \sigma_F(D_R)$.
- Xem xét phép chọn $V = \Pi_A(R)$: Giả sử quan hệ R có dạng $R=(A,B)$ và chứa hai bộ (a,2) và (a,3). Sau khi thực hiện phép chiếu trên thuộc tính A, quan hệ kết quả $\Pi_A(R)$ chứa bộ đơn (a). Nếu xóa bộ (a,2) ra khỏi R:
 - Xóa trực tiếp từ $\Pi_A(R)$: khi đó quan hệ $\Pi_A(R)$ là quan hệ rỗng.
 - Xóa từ quan hệ R trước khi chiếu, quan hệ $\Pi_A(R)$ sẽ chứa bộ đơn (a).

Như vậy tùy từng trường hợp cụ thể sử dụng phương pháp khung nhìn. Trong trường hợp xóa sau phép chiếu đúng, nếu lực lượng của $\Pi_A(R)$ lớn hơn 1. Ngược lại sẽ cho kết quả sai.

6.6.3 Các phép toán khối

Các phép toán khối, về nguyên tắc cũng như các phép chiếu và chọn. Các phép khối trong SQL là các phép COUNT(), SUM(), AVG(), MIN() và MAX().

- COUNT(): Xem xét khung nhìn cụ thể hóa $V = \text{COUNT}(\text{GROUP BY } A)$, phân hoạch theo thuộc tính A và đếm theo nó: Chèn
 - Nếu không tồn tại, chèn thêm (t.A.1) vào khung nhìn cụ thể hóa.
 - Nếu tồn tại (t.A), thêm 1 vào giá trị đếm của nhóm.
- SUM(): $V = \text{SUM}(\text{GROUP BY } A)$.
- AVG()
- MIN(), MAX().

6.6.4 Các phép toán khác

- Phép giao tập hợp: $V = R \cap S$:
 - Chèn một bộ vào R.
 - Xóa khỏi tập giao nếu nó tồn tại trong đó.
- Các phép toán tập hợp khác: Hợp và hiệu được thực hiện theo kiểu tương tự .

6.6.5 Tối ưu hóa các câu hỏi và các khung nhìn cụ thể hóa

Tối ưu hóa các câu hỏi có thể được thực hiện bằng cách xem các khung nhìn cụ thể hoá như là các quan hệ . Tuy nhiên, sử dụng phương pháp các khung nhìn cụ thể hóa tùy trong các trường hợp cụ thể.

- Biến đổi câu hỏi để sử dụng khung nhìn cụ thể hóa: Giả sử có một khung nhìn cụ thể $V = R \triangleright \triangleleft S$ và một câu hỏi $R \triangleright \triangleleft S \triangleright \triangleleft T$. Khi đó câu hỏi có thể biến đổi và được sử dụng khung nhìn cụ thể như sau: $V \triangleright \triangleleft T$.
- Thay thế khung nhìn cụ thể bằng khung nhìn bình thường. Giả sử có một khung nhìn cụ thể $V = R \triangleright \triangleleft S$ và một câu hỏi của người sử dụng yêu cầu thực hiện: $\sigma_{A=10}(V)$. Giả sử S có một chỉ số trên thuộc tính B và R có một chỉ số trên thuộc tính A. Khi đó phương án tốt nhất cho câu hỏi này có thể thay thế khung nhìn cụ thể V bởi khung nhìn bình thường $R \triangleright \triangleleft S$: $\sigma_{A=10}(R \triangleright \triangleleft S)$. Phép chọn và kết nối có thể được thực hiện một cách hiệu quả bằng cách sử dụng tương ứng các chỉ số trong R.A và S.B . Ngược lại đánh giá phép chọn một cách trực tiếp trên V có thể đòi hỏi sự quét toàn bộ V mà có thể tốn kém hơn nhiều.

6.7 Ví dụ tối ưu hoá các câu hỏi.

Xét các quan hệ sau:

TUYEN (TC#, TC, DAI)

CAP (MC#, MC, GIA, N#, NSX)

L_DAT (TC#, MC#, SL)

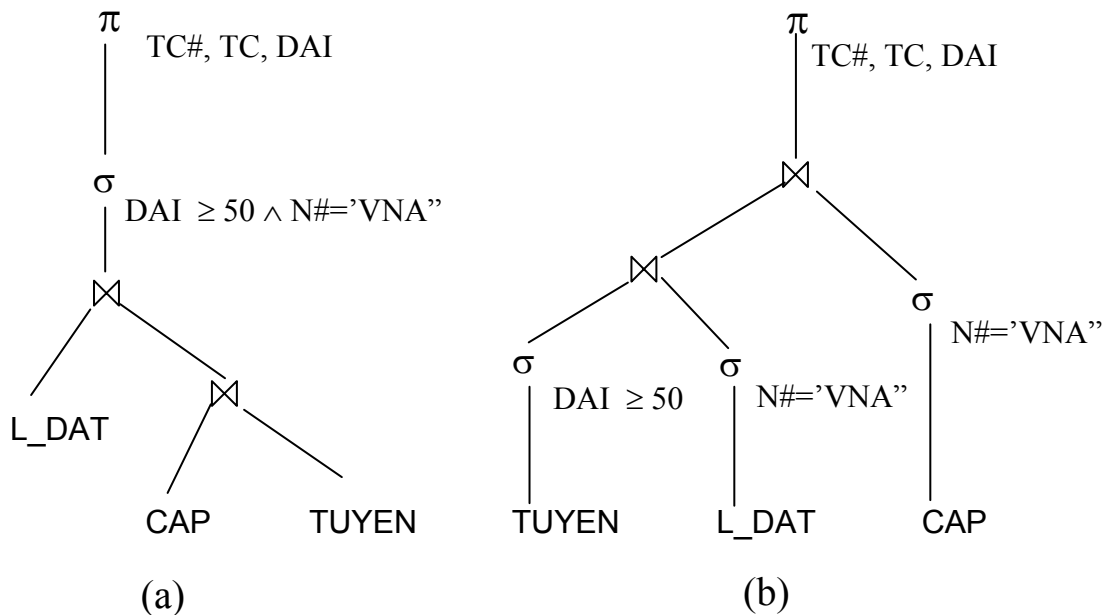
Trong đó

TC#	Mã tuyến cáp	MC#	Mã cáp
TC	Tên tuyến cáp	MC	Tên gọi cáp
DAI	Độ dài tuyến cáp	GIA	Đơn giá loại cáp
N#	Mã nước sản xuất cáp	NSX	Tên nước sản xuất cáp.
SL	Số lượng cáp lắp đặt		

Ngữ nghĩa của dữ liệu như sau:

- Trên một tuyến cáp có nhiều loại cáp khác nhau được lắp đặt.
- Một loại cáp có thể được lắp đặt trên nhiều tuyến.
- Thông tin về tuyến cáp: không tồn tại 2 mã tuyến trùng nhau.
- Không tồn tại mã 2 loại cáp trùng nhau.

Ví dụ Hãy liệt kê tất cả thông tin về các tuyến cáp có độ dài trên 50 km và lắp đặt các loại cáp do Việt nam sản xuất.



Hình 6.4 a) Cây đại số ban đầu chưa tối ưu hóa
b) Cây đại số đã tối ưu hóa

a) Biểu thức đại số quan hệ ban đầu như sau

$$E = \pi_{TC} (\sigma_{NSX='Việt nam' \text{ AND } (DAI>100)} (TUYEN \bowtie CAP \bowtie L_DAT)).$$

Kết nối tự nhiên các quan hệ TUYEN, CAP và L_DAT tạo thành một quan hệ trung gian với lực lượng bằng tích của bậc của 3 quan hệ và các thuộc tính là thuộc tính của 3 quan hệ. Tiếp theo thực hiện phép chọn với điều kiện chỉ quan tâm đến các loại cáp do Việt nam sản xuất được lắp đặt trên các tuyến có chiều dài trên 50 Km.

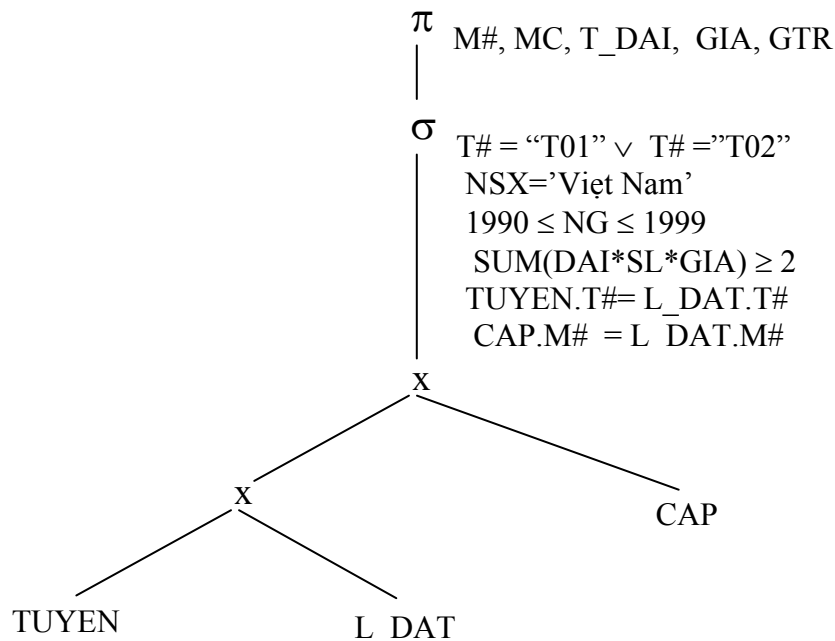
b) *Biểu thức tối ưu*

$$E' = \pi_{TC} ((\sigma_{NSX='Việt\ nam'}(CAP) \bowtie (\sigma_{DAI > 100}(TUYEN)) \bowtie (\sigma_{NSX='Việt\ nam'}(L_DAT)))$$

Đây là biểu thức tương đương với biểu thức ban đầu. Quan hệ trung gian có kích thước nhỏ hơn cũng như về lực lượng. Thời gian thực hiện nhanh hơn, chi phí bộ nhớ ít hơn. Trước tiên thực hiện phép chọn trên quan hệ TUYEN, tạo quan hệ trung gian bao gồm các bộ thỏa chiều dài trên 50 km. Phép chọn trên các quan hệ CAP, tạo ra quan hệ có một bộ duy nhất, nước sản xuất là Việt Nam, và trên quan hệ L_DAT, gồm các bộ thỏa N# = 'VNA'. Khi thực hiện phép kết nối các quan hệ trung gian trên, sẽ có quan hệ kích thước nhỏ hơn và lực lượng bé hơn, chi phí thấp hơn.

Ví dụ Hãy cho biết thông tin về các loại cáp do Việt nam sản xuất được lắp đặt trên các tuyến “T01” (Hà nội- Quảng ninh) và “T02” (Hà nội – Hải phòng) trong các năm 1990 đến 1999 có giá trị trên 2 tỷ đồng Việt nam. Giả sử khung nhìn cần phải tạo có tên là CAP_VN chứa các thông tin về M# mã cáp, MC tên cáp, GIA giá cáp, T_DAI tổng độ dài và giá trị của cùng loại cáp GTR. Để tạo khung CAP_VN cần kết nối các quan hệ TUYEN, CAP và L_DAT: $\pi_S (\sigma_F (TUYEN \times CAP \times L_DAT))$. Trong đó S là danh sách khung nhìn CAP_VN: S = M#, MC, GIA, T_DAI, GTR. Điều kiện chọn F:

$$F = (T\# = "T01" \vee T\# = "T02") \wedge (NSX = 'Việt\ Nam') \wedge (1990 \leq NG \leq 1999) \wedge (SUM(DAI * SL * GIA) \geq 2) \wedge (TUYEN.T\# = L_DAT.T\#) \wedge (CAP.M\# = L_DAT.M\#)$$



Hình 6.5 Cây đại số chưa tối ưu hóa

Tách biểu thức chọn F:

$$F_1 = \{1990 \leq NG \leq 1999\}$$

$$F_2 = \{NSX = 'Việt Nam'\}$$

$$F_3 = \{T\# = 'T01' \vee T\# = 'T02'\}$$

$$F_4 = \{TUYEN.T\# = L_DAT.T\#\}$$

$$F_5 = \{CAP.M\# = L_DAT.M\#\}$$

$$F_6 = \{SUM(DAI * SL * GIA) \geq 2\}$$

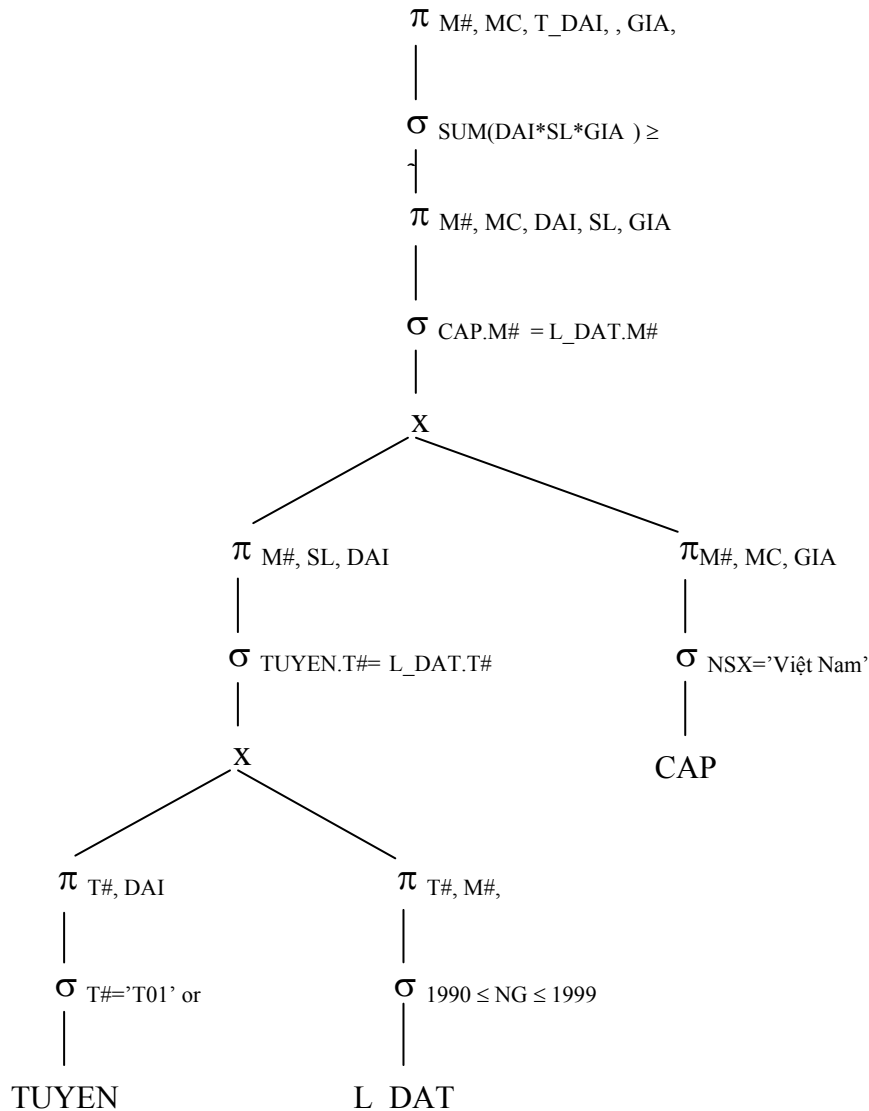
$$T1 = \pi_{T\#,DAI}(\sigma_{F_3}(TUYEN)) = \pi_{T\#,DAI}(\sigma_{T\# = 'T01' \vee T\# = 'T02'}(TUYEN))$$

$$T2 = \pi_{T\#,M\#,SL}(\sigma_{F_1}(L_DAT)) = \pi_{T\#,M\#,SL}(\sigma_{1990 \leq YEAR \leq 1999}(L_DAT))$$

$$T3 = \pi_{M\#,MC,GIA}(\sigma_{F_2}(CAP)) = \pi_{M\#,MC,GIA}(\sigma_{NSX = 'Việt Nam'}(CAP))$$

$$T4 = \sigma_{F_4}((T1 \bowtie_{F_4} T2) \bowtie_{F_5} T3)$$

$$T5 = \pi_{M\#,MC}(T4)$$



Hình 6.6 Cây cú pháp tối ưu hóa biểu thức quan hệ

Biểu diễn bằng ngôn ngữ SQL:

```
CREATE VIEW T1 AS
  SELECT T#, DAI
  FROM TUYEN
  WHERE T#="T01" AND T#="T03"
CREATE VIEW T2 AS
  SELECT T#, M#, SL
  FROM L_DAT
  WHERE YEAR FROM NG ) BETWEEN 1990 AND 1999.
CREAT VIEW T3 AS
  SELECT M#, MC, GIA
  FROM CAP
  WHERE NSX='Việt Nam'
CREAT VIEW T4 AS
  SELECT M#, MC#, DAI*SL*GIA GT
  FROM T1, T2, T3
  WHERE (T1.T#=T2.T# and T3.M#=T2.M# ) and DAI*SL*GIA ≥ 2
SELECT *
FROM T4
```

Câu hỏi trắc nghiệm

- Quá trình tối ưu hoá các câu hỏi truy vấn dữ liệu là quá trình:
 - Cần thiết phải biến đổi các câu hỏi hợp lý
 - Chi phí thời gian thực hiện các phép toán là ít nhất.
 - Cần thiết phải biến đổi các câu hỏi hợp lý sao cho chi phí thời gian thực hiện các phép toán là ít nhất.
 - Kết quả của các phép toán được biểu diễn duy nhất bằng một quan hệ.
- Ý nghĩa của biểu thức $\pi_{TTC} (\sigma_{MC#='Việt nam'} (TUYEN \times L_DAT) \times CAP)$
 - Liệt kê tên tất cả loại cấp Việt nam.
 - Liệt kê tên tất cả các loại cấp Việt nam có lắp đặt trên các tuyến
 - Liệt kê tên tất cả các tuyến có lắp đặt các loại cấp Việt nam.
 - Liệt kê tên tất cả các tuyến cấp
- Hai biểu thức sau đây tương đương với nhau
 - $\pi_{TTC} (\sigma_{MC#='Việt nam'} (L_DAT) \times (\sigma_{MC#='Việt nam'}(CAP) \times TUYEN))$
 - $\pi_{TTC} (\sigma_{MC#='Việt nam'} (TUYEN \times L_DAT) \times CAP)$
 - $\pi_{TTC} (\sigma_{MC#='Việt nam'} (L_DAT) \times \sigma_{MC#='Việt nam'}(CAP) \times TUYEN))$
 - $\pi_{TTC} (\sigma_{MC#='Việt nam'} (CAP \times L_DAT) \times TUYEN))$

4. Biểu thức nào dưới đây biểu diễn câu truy vấn: Tên các loại cáp sử dụng trên tuyến Hà nội- Hải phòng “T03” ?
- A. $\pi_{M\#, MC} (\sigma_{T\#="T03"} (L_DAT \bowtie_{L_DAT.M\#=CAP.M\#} CAP))$
 - B. $\pi_{M\#, MC} (CAP \bowtie_{CAP.M\#=T2.M\#} (\sigma_{T\#="T03"} (L_DAT)))$
 - C. $\sigma_{F1 \wedge F2 \wedge \dots \wedge Fn} (E) = \sigma_{F1} (\sigma_{F2} (\dots (\sigma_{Fn} (R)) \dots))$
 - D. $\pi_{A1, A2, \dots, An} (\pi_{B1, A2, \dots, Bk} (E)) = \pi_{A1, A2, \dots, An} (E)$.
5. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Tối ưu hoá câu hỏi không những tối ưu về thời gian truy xuất thông tin mà là nhân tố quan trọng trong việc bảo đảm.....với các chiến lược tìm kiếm, truy xuất dữ liệu.
- A. Tính khả thi của dữ liệu
 - B. Tính toàn vẹn dữ liệu
 - C. Tính độc lập của dữ liệu
 - D. Tính nhất quán của dữ liệu
6. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Có nhiều kỹ thuật khác nhau để tối ưu hoá các câu hỏi, trong đó có kỹ thuật phân rã câu hỏi dựa trên.....
- A. Ngôn ngữ của đại số quan hệ.
 - B. Ngôn ngữ dữ liệu.
 - C. Biểu thức đại số quan hệ.
 - D. Suy dẫn logic từ các tiên đề Armstrong
7. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Biến đổi một biểu thức đại số quan hệ về một biểu thức đại số quan hệ tương đương theo nghĩa sẽ cho cùng một kết quả nhưng với chi phí thời gian thực hiện và sử dụng bộ nhớ ít hơn rất nhiều
- A. Ngữ nghĩa dữ liệu
 - B. Tối ưu
 - C. Thực hiện
 - D. Tiết kiệm bộ nhớ
8. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Cách tiếp cận cơ bản để đánh giá tối ưu hóa biểu thức quan hệ thường cây kết nối trái theo chiều sâu, là phương án ưu hóa câu hỏi R không quan tâm tới thứ tự kết nối khi các toán hạng bên phải của mỗi phép kết nối . Cây kết nối trái theo chiều sâu rất có.....
- A. Chi phí rất nhỏ.
 - B. Chi phí bằng chi phí biểu thức ban đầu.
 - C. Cho phí thời gian nhỏ.
 - D. Chi phí bộ nhớ nhỏ
9. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Trong biểu thức quan hệ các toán hạng làvà các phép toán là các phép toán trong đại số quan hệ.
- A. Các quan hệ trong một CSDL
 - B. Các thuộc tính.
 - C. Các phụ thuộc hàm.
 - D. Các phần tử

10. Biểu thức E1 và E2 tương đương với nhau ($E1 \equiv E2$), nếu chúng biểu diễn cùng một ánh xạ, nghĩa là giống nhau trong biểu thức, thì kết quả cũng giống nhau.
- A. Các quan hệ.
 - B. Các thuộc tính.
 - C. Các bản ghi.
 - D. Các toán hạng
11. Tối ưu hoá là xác định trật tự thực hiện các phép toán trong E nhằm đáp ứng các nhu cầu cho cùng kết quả cho việc thực hiện theo trật tự ban đầu của biểu thức E, giảm bớt được..... thực hiện.
- A. Bộ nhớ và thời gian.
 - B. Bộ nhớ.
 - C. Thời gian.
 - D. Các phép toán
12. Hãy so khớp thông tin 2 phần dưới đây với nhau:

Phần 1

Phần 2

- | | |
|--|--|
| 1 Ưu tiên thực hiện các phép chiếu và chọn
.... | A. liên tiếp thành một phép toán duy nhất. |
| 2 Trước khi phải thực hiện phép tích Đề các
..... | B. hơn nhiều so với chi phí thực hiện phép tích Đề các |
| 3 Thực hiện các phép kết nối cân bằng chi phí sẽ rẻ..... | C. hãy tìm chiến lược truy nhập tốt nhất vào CSDL |
| 4 Nhóm các phép toán chọn và chiếu..... | D. nhằm giới hạn khối lượng dữ liệu trung gian. Giảm chi phí truy nhập bộ nhớ. |
- a. 1- D; 2- C ; 3 – B ; 4 – A .
 - b. 1- C; 2- B; 3 – A; 4 – D
 - c. 1- A; 2- B; 3 – C; 4 – D.
 - d. 1- D; 2- B; 3 – A; 4 – C.

Câu hỏi và bài tập:

Câu hỏi

Hãy tối ưu hoá các câu hỏi sau:

1. Thông tin về các tuyến có lắp đặt các loại cáp do Việt nam sản xuất có giá trị trên 10 tỷ đồng.
2. Thông tin về các tuyến cáp có lắp đặt các loại cáp sau năm 1999, hoặc có các loại cáp do Việt nam sản xuất.
3. Thông tin về các tuyến cáp xuất phát từ Hà nội, có các loại cáp được lắp đặt sau năm 1999, tổng giá trị tuyến cáp không lớn hơn 3 tỷ đồng.
4. Thông tin về các tuyến cáp có tổng giá trị lớn hơn tổng giá trị các tuyến cáp xuất phát từ Hà nội.

5. Thông tin về các tuyến cáp mà trong đó mỗi một loại chỉ có một loại cáp được lắp đặt, hoặc có ít nhất 5 loại cáp được lắp đặt.
6. Thông tin về các tuyến cáp mà trong đó lắp đặt tất cả các loại cáp có trong danh mục.
7. Tổng giá trị các loại cáp lắp đặt trong các tuyến.
8. Thông tin về các tuyến cáp không sử dụng các loại cáp do Việt nam sản xuất.
9. Thông tin về các tuyến cáp có mã loại cáp không đúng với danh mục (ngoài danh mục).
10. Tổng giá trị các tuyến cáp
11. Thông tin về các tuyến cáp có tổng giá trị nhỏ hơn giá trị của tuyến mang mã hiệu "T15"
12. Thông tin về các tuyến cáp mà trong đó số loại cáp lắp đặt nhiều hơn số loại cáp lắp đặt trong tuyến "T01".
13. Thông tin về các loại cáp do Việt nam sản xuất có giá trị trên 10 tỷ đồng, được sử dụng trong năm 1999.
14. Thông tin về các loại cáp do Việt nam sản xuất hoặc do các liên doanh Việt sản xuất.
15. Thông tin về các loại cáp được lắp đặt trên các tuyến xuất phát từ Hà nội, có giá trị cáp không lớn nhỏ 5 triệu đồng.
16. Thông tin về các loại cáp có chiều dài sử dụng và giá trị của nó lớn hơn chiều dài và giá trị sử dụng của loại cáp mang mã hiệu "M03".
17. Thông tin về các loại cáp được sử dụng trên hơn 10 tuyến.
18. Thông tin về các loại cáp được sử dụng duy nhất trên một tuyến cáp.
19. Thông tin về các loại cáp được sử dụng trên tất cả các tuyến.
20. Thông tin về các loại cáp chưa được sử dụng trên các tuyến.
21. Thông tin về các loại cáp được sử dụng trên 2/3 tuyến cáp.
22. Thông tin về các loại cáp được lắp đặt trên các tuyến xuất phát từ Hà nội, hoặc đến Hà nội có các loại cáp do Việt Nam sản xuất, có giá trị cáp lớn 5 triệu đồng.

Bài tập

1. Giả sử các quan hệ $R_1(A,B,C)$, $R_2(C,D,E)$ và $R_3(E,F)$, và R_1 có 1000 bộ, R_2 có 1500 bộ và R_3 có 750 bộ. Đánh giá kích cỡ của tích Đề các của R_1 , R_2 và R_3 . Hãy đề xuất một chiến lược hiệu quả để tính toán phép kết nối.
2. Giả sử các quan hệ $R_1(A,B,C)$, $R_2(A,D,F)$ và $R_3(B,E,T,Z)$ và R_1 có 3500 bộ, R_2 có 500 bộ và R_3 có 700. Hãy tính chi phí cho các thao tác sau:
 - Thêm 100 bộ vào R_1 .
 - Thêm 20 bộ vào R_2 .
 - Thêm 20 bộ vào R_3 .
 - Biết chi phí của $R_1 \times R_2$ hãy tính chi phí $R_1 \times R_2 \times R_3$
3. Hãy tính chi phí cho các biểu thức quan hệ sau: Thông tin về các loại cáp có giá trị trung bình cao hơn giá trị trung bình của các loại cáp do Việt nam sản xuất được lắp đặt trên từng tuyến.

7 AN TOÀN DỮ LIỆU VÀ TÍNH TOÀN VỆ DỮ LIỆU

Chương này sẽ giới thiệu một cách tổng quát về khái niệm an toàn và toàn vẹn dữ liệu. và một số biện pháp cần thiết để bảo vệ cơ sở dữ liệu. “An toàn” dữ liệu nghĩa là các hệ cơ sở dữ liệu cần phải được bảo vệ chống truy nhập trái phép như sửa đổi hay phá hoại dữ liệu. Tính toàn vẹn dữ liệu theo nghĩa dữ liệu không mâu thuẫn, dị thường thông tin trong các hệ thống khi thực hiện các phép lưu trữ, truy vấn, tìm kiếm. Nội dung bao gồm các vấn đề sau:

- Sự cần thiết phải bảo vệ an toàn cơ sở dữ liệu
- Tính toàn vẹn của dữ liệu
- Ngôn ngữ vấn tin - ngôn ngữ ràng buộc toàn vẹn
- An toàn cơ sở dữ liệu.
- Các mức độ bảo vệ an toàn cơ sở dữ liệu.
- Bảo mật cơ sở dữ liệu

7.1 Sự cần thiết phải bảo vệ an toàn cơ sở dữ liệu

Cơ sở dữ liệu của một cơ quan, một xí nghiệp, của một ngành... thường được cài đặt tập trung hay phân tán trên các máy chủ trên mạng, là tài nguyên thông tin chung cho nhiều người cùng sử dụng. Vì vậy các hệ cơ sở dữ liệu cần phải có cơ chế kiểm soát, quản lý và truy xuất khai thác thông tin sao cho dữ liệu phải được an toàn và toàn vẹn. Thuật ngữ “an toàn” dữ liệu có nghĩa là các hệ cơ sở dữ liệu cần phải được bảo vệ chống truy nhập nhằm sửa đổi hay phá hoại một cách chủ định hay không chủ định. Như vậy các hệ thống cơ sở dữ liệu cần thiết phải quản trị, bảo vệ tập trung, nhằm bảo đảm được tính toàn vẹn và an toàn dữ liệu. Toàn vẹn dữ liệu khác với an toàn dữ liệu, tuy rằng chúng có mối quan hệ mật thiết với nhau. Có thể sử dụng chung một số biện pháp để thực hiện. Có rất nhiều mối nguy hiểm đe dọa đến các hệ thống dữ liệu:

- Cơ sở dữ liệu được cài đặt tập trung hay phân tán trên các vị trí địa lý khác nhau, được khai thác từ các đầu cuối khác nhau theo chế độ Client/Server.
- Nhiều người sử dụng truy nhập và khai thác trên cùng một cơ sở dữ liệu.
- Rất nhiều loại dữ liệu được tải về giữ trên các máy cục bộ để khai thác.
- Truy xuất vào các hệ cơ sở dữ liệu bằng nhiều ngôn ngữ thao tác dữ liệu khác nhau, bằng nhiều hệ ứng dụng khác nhau trên cùng một nội dung thông tin.

Vì vậy có thể xảy ra

- Những sai sót ngoài ý muốn, khi thực hiện thêm, sửa, xoá hay do lỗi khi lập trình.
- Truy nhập trái phép với mục đích xấu: sửa, xoá thông tin hay đánh cắp thông tin...
- Sự cố kỹ thuật như lỗi do các thiết bị, lỗi lập trình...

7.2 Tính toàn vẹn dữ liệu

Vấn đề toàn vẹn dữ liệu là vấn đề đảm bảo dữ liệu trong các hệ thống cơ sở dữ liệu được chính xác tại mọi thời điểm trong chừng mực có thể. Có một giới hạn về phạm vi mà mục tiêu này có thể thực hiện được. Hệ thống cơ sở dữ liệu không thể kiểm tra hết tính đúng đắn

của mỗi một giá trị được đưa vào CSDL. Chẳng hạn, không thể phát hiện giá trị của thuộc tính số điện thoại mới được lắp đặt 8515678, giá trị đúng phải là 8515673 (tất nhiên số điện thoại này trước đó chưa tồn tại trong các quan hệ của cơ sở dữ liệu). Tuy nhiên, nếu nhập vào giá trị 0515673, hệ thống sẽ phát hiện sai và không chấp nhận giá trị này.

Bảo đảm tính toàn vẹn của dữ liệu là một trong các biện pháp bảo vệ dữ liệu, chống lại sự sửa đổi hay phá hoại không chủ định. Có thể có nhiều nguyên nhân khác nhau dẫn đến không được đảm bảo tính toàn vẹn dữ liệu. Có thể trục trặc kỹ thuật, như bộ nhớ bị chập chờn hư hỏng, đường truyền bị lỗi, thiết bị ra/vào không ổn định.. hoặc người sử dụng đầu cuối thao tác sai sót, không đúng quy trình ...hoặc chương trình ứng dụng bị lỗi.. Không đảm bảo tính toàn vẹn dữ liệu có thể xảy ra trong hệ thống coi là ổn định nhất và đảm bảo nhất... Vì vậy điều quan trọng là phải có khả năng phát hiện những tình huống khi chúng xuất hiện và có khả năng khôi phục từ những tình huống đó.

7.2.1 Các ràng buộc toàn vẹn

Giữa các thực thể dữ liệu tồn tại các mối quan hệ, ràng buộc phụ thuộc lẫn nhau. Có 2 loại ràng buộc chủ yếu. Một loại ràng buộc có tính cấu trúc, liên quan đến các đẳng thức giữa các giá trị trong CSDL. Mô tả tính chất của các thuộc tính khi tạo lập CSDL như tên, kiểu, độ dài của thuộc tính. Chẳng hạn thuộc tính “Số điện thoại” là kiểu chuỗi có độ dài đúng bằng 7 ký tự. Hệ thống sẽ không chấp nhận, nếu nhập vào CSDL một giá trị kiểu số hoặc kiểu xâu nhưng chưa đủ hoặc vượt quá 7 ký tự. Ràng buộc biểu diễn giá trị của các thuộc tính bằng đẳng thức toán học, các toán hạng là giá trị của các thuộc tính khác. Biểu hình là những ràng buộc hạn chế một trường chỉ nhận được một số giá trị trong một phạm vi nào đó hoặc diễn tả một mối liên hệ số học giữa các trường khác nhau. Ví dụ, nếu các bản ghi của một hợp đồng thuê bao điện thoại chứa các trường SDT (số điện thoại), NGK (ngày ký hợp đồng), TLD (tiền lắp đặt). Miền xác định giá trị của SDT là một chuỗi 7 ký tự số, không chứa ký tự đặc biệt, ký tự đầu khác ký tự không. Miền xác định của NLD là thời gian ngày tháng năm ký hợp đồng không quá 15 ngày kể từ ngày yêu cầu và miền xác định của TLD là một số thập phân không quá 1.400.000 nghìn đồng. Khi nhập dữ liệu vào CSDL, hệ thống sẽ kiểm tra tính đúng đắn của các giá trị mà khi tạo lập CSDL đã mô tả, nếu vi phạm sẽ không được chấp nhận và thông báo lỗi.

Loại ràng buộc toàn vẹn thứ hai là loại ràng buộc logic giữa các thuộc tính, là các ràng buộc có thể biểu diễn được nếu hệ quản trị cơ sở dữ liệu DBMS cho phép người sử dụng khai báo một tập các trường hoặc thuộc tính tạo ra khóa cho mẫu tin hoặc quan hệ. Trong các hệ thống cơ sở dữ liệu quan hệ, biểu diễn mối quan hệ các thuộc tính là tập các ràng buộc được mô tả bằng khái niệm các phụ thuộc hàm và phụ thuộc đa trị, hay còn được gọi là các phụ thuộc dữ liệu. Theo định nghĩa, mọi khoá của một lược đồ quan hệ đều có tính duy nhất. Không tồn tại 2 bộ có cùng một giá trị trên các thuộc tính khoá. Giá trị của khoá theo quy định không được nhận giá trị null hay giá trị không xác định. Hệ quản trị CSDL có trách nhiệm phải từ chối khi giá trị của khoá nếu vẫn còn chứa các giá trị chưa xác định hoặc đã tồn tại trong cơ sở dữ liệu thông qua các phép thêm hoặc sửa đổi dữ liệu.

Như vậy các ràng buộc toàn vẹn dữ liệu phải bảo đảm tính độc lập và toàn vẹn dữ liệu. Dữ liệu lưu trữ trong các hệ thống cơ sở dữ liệu phải phản ánh hiện thực khách quan trong mọi thời điểm có thể, không dư thừa và mâu thuẫn thông tin.

7.2.2 Ngôn ngữ vấn tin - ngôn ngữ ràng buộc toàn vẹn

Thông thường có nhiều loại ràng buộc toàn vẹn có thể mô tả bằng các ngôn ngữ thao tác dữ liệu DML (Data Manipulation Language). Nói cách khác, có thể sử dụng ngôn ngữ DML làm ngôn ngữ kiểm tra ràng buộc toàn vẹn. Phần này sẽ xem xét về mặt lý thuyết khi sử dụng đại số quan hệ làm ngôn ngữ ràng buộc dữ liệu.

Ví dụ Xét cơ sở dữ liệu quản lý cấp, ràng buộc dữ liệu có thể là: các loại cấp lắp đặt trên tuyến cáp QLCAP, phải là các loại cấp đã có trong quan hệ QLKHO. Nói cách khác, tất cả các cấp lắp đặt trên tuyến phải có xuất xứ từ kho vật tư của công ty. Kết quả vấn tin có thể biểu diễn bằng đại số quan hệ như sau:

$$\pi_{MC\#}(QLCAP) \subseteq \pi_{TENC}(QLKHO)$$

Ví dụ Các cuộc đàm thoại có thời gian âm cấm nhập vào cơ sở dữ liệu:

$$QLCUOC(TB, SDT, SDD, BD, KT) \ \& \ (KT - BD) < 0.$$

Điều này có nghĩa là, cuộc đàm thoại của thuê bao TB từ điện thoại SDT đến điện thoại SDD với thời gian bắt đầu BD, thời gian kết thúc KT và thời gian cuộc đàm thoại $(KT - BD) < 0$, khi đó sẽ dẫn đến mâu thuẫn và nhận giá trị là “False”. Điều này tương đương với khẳng định không tồn tại các cuộc đàm thoại có thời gian âm trong cơ sở dữ liệu QLCUOC.

7.2.3 Kiểm tra các ràng buộc toàn vẹn

Trong ví dụ trên, trước khi chèn thêm thông tin về một loại cáp mới vào quan hệ QLCAP, cần phải kiểm tra xem loại cáp đó đã được xuất kho hay chưa. Tức là phải kiểm tra loại cáp thêm vào CSDL QLCAP là một phần tử của $\pi_{TENC}(QLKHO)$ hay không. Tương tự, trong ví dụ 7.2, trước khi chèn thông tin của một cuộc đàm thoại mới vào cơ sở dữ liệu QLCUOC, cần phải kiểm tra thời gian của cuộc đàm thoại này có âm hay không. (Lưu ý, không phải kiểm tra thời gian của các cuộc đàm thoại đã có trong cơ sở dữ liệu).

7.3 Vấn đề an toàn và quyền truy nhập cơ sở dữ liệu

Dữ liệu lưu trữ trong cơ sở dữ liệu cần phải được bảo vệ để tránh việc truy nhập trái phép và phá hoại có chủ định hay không chủ định khi thực hiện cập nhật, sửa đổi hay bổ sung thông tin trong các cơ sở dữ liệu. Cần phải có biện pháp bảo vệ chống lại việc đưa dữ liệu vào một cách không nhất quán ảnh hưởng nghiêm trọng đến tính toàn vẹn dữ liệu.

7.3.1 Sự vi phạm an toàn cơ sở dữ liệu.

Các dạng truy cập có chủ định bao gồm :

- Không cho phép đọc dữ liệu.
- Không cho phép sửa đổi dữ liệu.
- Không cho phép phá hủy dữ liệu...

Vấn đề an toàn cơ sở dữ liệu đề cập đến việc bảo vệ chống lại sự truy cập có chủ định. Việc bảo vệ tuyệt đối các hệ cơ sở dữ liệu khỏi truy nhập là không thể, nhưng phải có các biện pháp đủ mạnh để ngăn chặn hầu hết truy cập trái phép vào cơ sở dữ liệu.

7.3.2 Các mức độ an toàn cơ sở dữ liệu.

Để bảo vệ cơ sở dữ liệu, phải thực hiện các biện pháp đảm bảo an toàn ở một vài mức bảo vệ như sau:

- *Mức độ an toàn hệ thống cơ sở dữ liệu:* Tùy thuộc vào yêu cầu của người sử dụng mà người quản trị cơ sở dữ liệu cấp phép truy nhập một phần vào cơ sở dữ liệu. Những người sử dụng khác có thể được phép thực hiện các câu hỏi truy vấn, nhưng có thể bị ngăn cấm ý định sửa đổi dữ liệu.
- *Mức độ an toàn hệ thống điều hành:* Mức hệ thống kiểm soát toàn bộ mức điều hành hệ thống. Vấn đề an toàn mức hệ thống điều hành sẽ được đảm bảo bởi mức độ an toàn hệ thống cơ sở dữ liệu. An toàn trong hệ điều hành đã được tiến hành tại nhiều cấp độ từ sắp xếp các mật mã truy cập vào hệ thống cho tới sự cô lập các quá trình đang cùng xử lý trong hệ thống. Tập hệ thống cũng cung cấp một số cấp độ bảo vệ. Sự tham khảo những chú ý trong thư mục là bao quát của những chủ đề này trong các bài học về hệ thống điều hành.
- *An toàn mức độ mạng:* Hầu hết các hệ thống cơ sở dữ liệu đều cho phép truy cập từ xa thông qua các thiết bị đầu cuối. An toàn dữ liệu mức độ mạng là chống ăn cắp thông tin, sao chép thông tin và sửa đổi nội dung thông tin trên đường truyền. Vấn đề an toàn cấp mức mạng đã đạt được nhiều kết quả, ứng dụng phổ biến trên mạng Internet. Danh sách các chú ý trong thư mục đã bao quát nền tảng nguyên lý của vấn đề an toàn mạng.
- *Nhận diện người sử dụng:* Từ định nghĩa an toàn dữ liệu có thể suy ra rằng, hệ quản trị cơ sở dữ liệu DBMS không cho phép người sử dụng được thực hiện một thao tác nào nếu không được phép của người quản trị CSDL. Người quản trị CSDL phải:
 - Xác định cho hệ thống những thao tác mà người sử dụng được phép thực hiện.
 - Cung cấp một phương tiện cho người sử dụng để hệ thống nhận biết họ.

Nói chung người sử dụng đều được trao những quyền khác nhau. Những quyền này có thể bảo đảm quyền đọc một số phần của cơ sở dữ liệu, quyền chèn thêm, xóa hay sửa đổi dữ liệu. Hình thức thông dụng nhất để nhận ra người sử dụng là mật khẩu, và chỉ có hệ thống và người sử dụng biết. Mật khẩu cũng được hệ thống bảo vệ như bảo vệ dữ liệu.

- *Bảo vệ mức vật lý:* Một mô hình bảo vệ đáng tin cậy cũng có khả năng bị tấn công vào cơ sở dữ liệu, từ việc phá được mật khẩu đến việc đánh cắp các thiết bị. Có thể chống đánh cắp khá hiệu quả bằng cách mã hóa, che dấu dữ liệu. Một hệ thống có bảo mật cao cần phải có những phương thức nhận diện khác tốt hơn mật khẩu, như nhận diện từng người sử dụng qua một nhân viên bảo vệ, hoặc kết với các quy định về hành chính...
- *Kiểm tra truy nhập:* Với mỗi người sử dụng hệ thống sẽ quản lý một hồ sơ được phát sinh từ việc các chi tiết về thủ tục xuất trình, xác minh và các chi tiết được quyền thao tác mà người quản trị cơ sở dữ liệu cấp cho người sử dụng. Hệ thống sẽ kiểm tra tính pháp lý của mỗi một thao tác của người sử dụng. Ví dụ yêu cầu được đọc lời đánh giá hàng năm của mỗi một nhân viên, chỉ có thể được phép nếu cơ sở dữ liệu có chứa thông tin quy định rằng người yêu cầu phải là Giám đốc, trưởng, phó phòng tổ chức, chánh văn phòng. Tất cả các đối tượng khác không có trong cơ sở dữ liệu không được phép truy xuất. DBMS sẽ kiểm tra

mỗi một thao tác của người sử dụng xem có vi phạm các ràng buộc an toàn hay không, nếu có sẽ phải huỷ bỏ. Một ràng buộc truy nhập nói chung có liên quan đến một bộ phận của cơ sở dữ liệu. Do đó tồn tại một đặc quyền thích hợp, giả sử là chương trình sẽ kiểm tra mỗi một yêu cầu của người sử dụng. Chương trình sẽ sắp xếp quyền truy nhập theo mức độ phức tạp tăng dần sao cho đạt tới quyết định cuối cùng nhanh nhất có thể.

An ninh ở tất cả các cấp độ phải được duy trì nếu an ninh cơ sở dữ liệu được bảo đảm. Một sự yếu kém ở vấn đề an toàn cấp thấp (cấp độ vật lý hay cấp độ con người) cho phép sự phá vỡ các biện pháp an toàn nghiêm ngặt ở cấp độ cao (cấp độ hệ thống cơ sở dữ liệu).

7.3.3 Những quyền hạn khi sử dụng hệ cơ sở dữ liệu.

Có thể chia quyền hạn truy nhập vào cơ sở dữ liệu như sau

- Đọc một cách hợp pháp: người sử dụng được phép đọc, nhưng không được sửa đổi nội dung dữ liệu.
- Chèn một cách hợp pháp: là cho phép người sử dụng được chèn thêm dữ liệu mới vào cơ sở dữ liệu, nhưng không sửa đổi dữ liệu hiện có.
- Sửa đổi một cách hợp pháp: cho phép người sử dụng được phép sửa đổi nội dung dữ liệu, nhưng không được xoá dữ liệu.
- Xoá một cách hợp pháp: cho phép người sử dụng được phép xoá dữ liệu.
- Cho phép việc tạo và xoá các chỉ số.
- Cho phép việc tạo các mối quan hệ mới.
- Sửa đổi cấu trúc: cho phép chèn thêm, sửa đổi hoặc xoá các thuộc tính trong các quan hệ.
- Bỏ hợp pháp: cho phép xoá các quan hệ.

Một người sử dụng có thể có tất cả các quyền trên, hoặc chỉ có một số quyền hạn nhất định. Thêm vào đó những dạng của sự cho phép truy cập dữ liệu chúng ta có thể ban cho người sử dụng được phép sửa đổi cơ cấu cơ sở dữ liệu. Cho phép bỏ và xoá là khác nhau trong đó xoá hợp pháp là chỉ cho phép xoá bộ dữ liệu. Nếu một người sử dụng xoá tất cả các bộ của một quan hệ, quan hệ đó sẽ vẫn tồn tại nhưng quan hệ đó không còn gì. Nếu một quan hệ bị bỏ nó sẽ không còn tồn tại nữa.

Để minh hoạ bản chất của vấn đề, không mất tính tổng quát, các mệnh đề sau chỉ là một vài ý niệm phạm vi bảo vệ thông tin trong các hệ cơ sở dữ liệu, chỉ ra các mức truy nhập CSDL và trao quyền cho từng lớp người sử dụng:

1. Người sử dụng được phép truy nhập không điều kiện tới toàn bộ cơ sở dữ liệu, với mọi phép toán lưu trữ và truy vấn dữ liệu tr.
2. Người sử dụng không được phép truy nhập tới bất kỳ bộ phận nào của cơ sở dữ liệu, với mọi phép toán.
3. Người sử dụng có thể đọc đúng một nội dung công việc của họ trong cơ sở dữ liệu, nhưng không được phép sửa đổi, bổ sung nó.
4. Người sử dụng có thể đọc đúng một nội dung công việc của họ trong cơ sở dữ liệu, và được phép sửa đổi, bổ sung nó.

5. Người sử dụng có thể đọc và sửa đổi thuộc tính mã nhân viên, họ và tên nhân viên, đơn vị công tác theo định kỳ vào tuần đầu của mỗi tháng.
6. Người sử dụng cấm đọc thuộc tính nhận xét hàng năm, các thuộc tính mức lương và ngày lên lương được đọc và sửa đổi, các thuộc tính khác chỉ được đọc. Công việc chỉ được thực hiện trong khoảng thời gian từ 9 giờ đến 11 giờ trong các ngày của tuần cuối tháng.
7. Người sử dụng có quyền sử dụng các phép toán thống kê cho thuộc tính mức lương để tính mức lương trung bình trong từng đơn vị. Cấm sửa đổi dữ liệu.

7.3.4 Khung nhìn – một cơ chế bảo vệ

Khung nhìn, bằng cách định nghĩa lại cơ sở dữ liệu khái niệm, không chỉ tạo điều kiện thuận lợi khi lập trình ứng dụng và làm tăng tính độc lập dữ liệu logic, mà còn được sử dụng như một cơ chế bảo vệ. Có hai loại khung nhìn. Loại khung nhìn chỉ đọc, không cho phép sửa đổi. Loại khung này gọi là khung chỉ đọc. Trong nhiều trường hợp, người quản trị CSDL cho phép người sử dụng này được đọc dữ liệu, nhưng người khác vừa được đọc, vừa được quyền sửa đổi, bổ sung... Loại khung nhìn thứ hai cho phép đọc và ghi lên các thành phần của khung nhìn. và mọi sửa đổi cho khung nhìn có thể được lưu trong lược đồ khái niệm. SQL đề xuất cho phép đọc/ghi các khung nhìn trong một phạm vi nhất định. Với phương pháp này thiết kế các chương trình ứng dụng linh hoạt hơn loại khung chỉ đọc. Tuy nhiên, khi thao tác cập nhật trên các khung nhìn đọc/ghi thường gây tác động đến một số thành phần của cơ sở dữ liệu không nằm trong khung nhìn. Ví dụ trong một hệ CSDL phân cấp, trong khung nhìn chỉ có kiểu bản ghi gốc, không có bản ghi phụ thuộc. Nếu xóa xuất hiện của kiểu bản ghi này, kéo theo phải xóa các xuất hiện bản ghi phụ thuộc. Đây là một hành động không hợp lệ, vi phạm nguyên tắc không cho người sử dụng được phép xóa một đối tượng mà họ không thấy được trong khung nhìn. Cũng tương tự như trong mô hình mạng, nếu xóa một bản ghi khi không biết các bản ghi khác nằm ngoài khung nhìn nhưng có quan hệ với nó. Và nhiều trường hợp khác tương tự. Vì vậy, tất cả các hệ quản trị cơ sở dữ liệu .DBMS giới hạn quyền cập nhật các khung nhìn trong một số trường hợp cụ thể.

Ví dụ về hoạt động của ngân hàng, một thư ký cần biết tên của tất cả các khách hàng có các khoản vay tại nhiều chi nhánh. Người thư ký này không được phép xem những thông tin về khoản vay đặc biệt mà khách hàng có thể có. Hành động của cô thư ký bị từ chối khi truy nhập trực tiếp tới quan hệ cho vay, nhưng có thể truy nhập bằng khung nhìn *cust-loan* bao gồm các thông tin như: tên của khách hàng và chi nhánh nơi mà khách đó có khoản vay. Khung nhìn này có thể được định nghĩa trong SQL như sau:

```
CREATE VIEW cust-loan AS
  (SELECT branch-name, customer-name
   FROM borrower, loan
   WHERE borrower.loan-number = loan.loan-number)
```

Giả sử rằng cô thư ký đưa ra truy vấn SQL như sau:

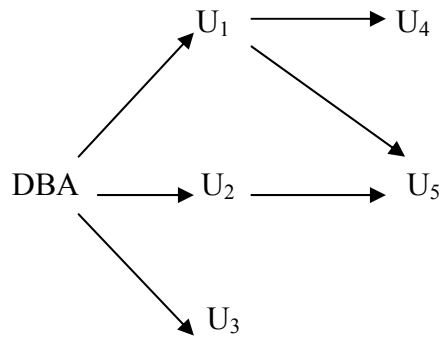
```
SELECT *
```

FROM cust-loan

Như vậy người thư ký được phép xem kết quả của truy vấn trên, tuy nhiên quá trình xử lý truy vấn này sẽ được thực hiện trên các quan hệ BORROWER and LOAN. Vì vậy hệ thống phải kiểm tra các quyền hạn trên truy vấn của thư ký trước khi bắt đầu quá trình xử lý truy vấn. Việc tạo một khung nhìn không phụ thuộc vào các quan hệ nguồn. Một người sử dụng tạo ra một khung nhìn không được nhận tất cả các đặc quyền trên khung nhìn. Ví dụ, người sử dụng không được quyền cập nhật trên khung nhìn nếu không có quyền cập nhật vào quan hệ bằng khung nhìn đã được định nghĩa. Nếu người sử dụng tạo ra một khung nhìn trên những quyền hạn không được phép, thì hệ thống sẽ phủ nhận yêu cầu tạo khung nhìn. Trong ví dụ khung nhìn cust-loan ở trên, người tạo khung nhìn phải có quyền đọc trên cả hai quan hệ BORROWER and LOAN.

7.3.5 Cấp phép các quyền truy nhập

Một người sử dụng được cấp một vài quyền truy nhập cơ sở dữ liệu và các quyền hạn này có thể tham chiếu đến quyền truy nhập của người sử dụng khác. Tuy nhiên người quản trị cơ sở dữ liệu cũng cần phải đặc biệt lưu ý khi các quyền này lưu thông qua giữa nhiều người sử dụng, sao cho các quyền này có thể được thu hồi tại một thời điểm tùy ý.



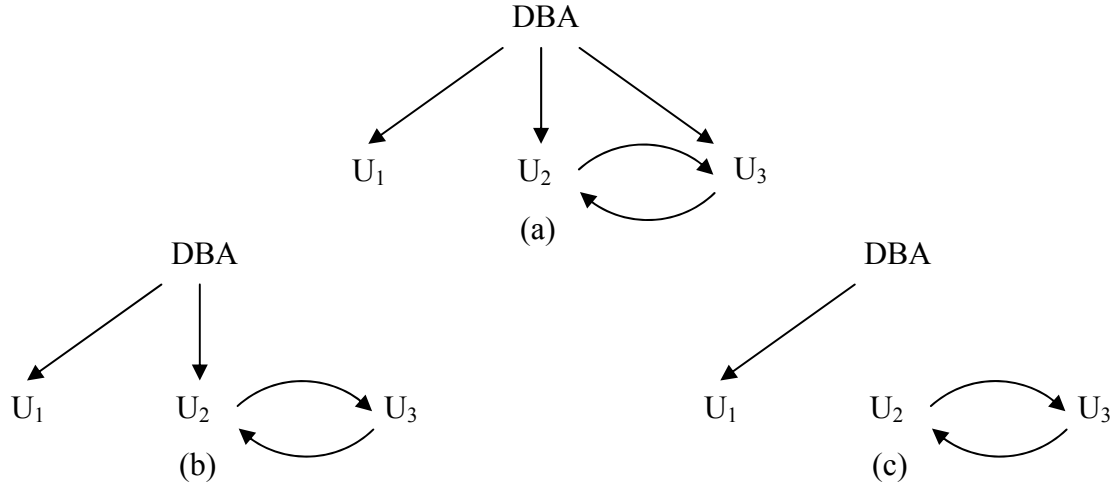
Hình 7.1 Đồ thị cấp quyền truy nhập cơ sở dữ liệu

Vi dụ, giả sử khi khởi tạo, người quản trị cơ sở dữ liệu cấp quyền cập nhật dữ liệu trên quan hệ LOAN của cơ sở dữ liệu ngân hàng cho người sử dụng U_1 , U_2 và U_3 và quyền có thể trong thứ tự thông qua quyền hạn đến các quyền của những người sử dụng khác. Liên thông các quyền từ một người sử dụng này tới người sử dụng khác được mô tả bằng một đồ thị quyền hạn. Đồ thị bao gồm các nút là những người sử dụng và các cạnh $U_i \rightarrow U_j$ nếu người sử dụng U_i cấp quyền cập nhật trên LOAN cho người sử dụng U_j . Góc của đồ thị là người quản trị cơ sở dữ liệu. Trong hình 7.1, người sử dụng U_5 được cấp quyền hạn bởi hai người sử dụng U_1 và U_2 và người sử dụng U_4 được cấp quyền sử dụng chỉ bởi U_1 .

Một người sử dụng có quyền hạn truy nhập vào cơ sở dữ liệu theo một số quyền nào đó khi và chỉ khi (*if and only if*) có một đường đi từ gốc trên đồ thị quyền hạn, tức là liên thông từ nút người quản trị cơ sở dữ liệu tới nút người sử dụng.

Giả sử người quản trị cơ sở dữ liệu quyết định thu hồi các quyền hạn của người sử dụng U_1 . Vì người sử dụng U_4 có quyền hạn đến từ U_1 nên quyền hạn của U_4 cũng sẽ bị thu hồi.

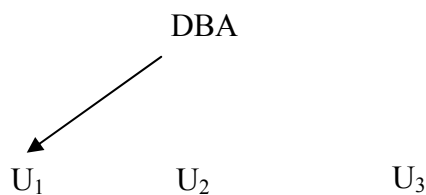
Tuy nhiên, vì U_5 được cấp quyền bởi U_1 và U_2 , vì thế người quản trị cơ sở dữ liệu chỉ thu hồi từ U_1 đến U_5 , không thu hồi quyền cập nhật trên LOAN của U_2 . U_5 vẫn còn quyền cập nhật trên quan hệ LOAN. Nếu người quản trị thu hồi quyền cập nhật của U_2 thì U_5 sẽ mất quyền hạn đó trên quan hệ LOAN.



Hình 7.2 Cố gắng huỷ bỏ những quyền hạn đã bị thu hồi

Có thể xảy ra những trường hợp một cặp người sử dụng “láu cá” có thể cố gắng không chấp nhận các quy tắc thu hồi quyền đã được cấp phát. Giả sử đồ thị cấp phát quyền truy nhập như ở trong hình 7.2a. Ngoài các cạnh xuất phát từ gốc DBA đến U_1 , U_2 và U_3 , giữa U_2 và U_3 còn tồn tại các đường từ U_2 đến U_3 và ngược lại từ U_3 đến U_2 . Điều này có nghĩa là người quản trị cấp phát quyền cho U_1 , U_2 và U_3 , U_2 còn thêm các quyền của U_2 và U_3 còn thêm các quyền của U_2 . Nếu người quản trị cơ sở dữ liệu thu hồi quyền của U_3 giữ lại quyền hạn của U_2 thì quyền truy nhập của U_3 vẫn còn, không bị mất vì đường đi từ gốc đến U_3 liên thông qua U_2 như trong hình 7.2b. Nếu thu hồi đồng thời quyền của cả hai người sử dụng U_3 , U_3 khi đó các quyền của U_3 và U_3 vẫn tồn tại như trong hình 7.2c. Tuy nhiên khi nhà quản trị cơ sở dữ liệu đã xoá bỏ cạnh từ U_3 tới U_2 và từ U_2 tới U_3 thì các quyền sẽ không còn tồn tại trên đường truyền bắt nguồn từ người quản trị cơ sở dữ liệu.

Tuy nhiên, người quản trị cơ sở dữ liệu yêu cầu tất cả các cạnh trong đồ thị cấp quyền truy nhập phải liên thông bắt đầu từ nút gốc, hay bắt đầu từ người quản trị DBA. Như vậy cạnh đi từ U_2 và U_3 và ngược lại sẽ bị xoá, tức là các quyền từ U_2 đến U_3 và ngược lại phải được thu hồi như trong hình 7.3.



Hình 7.3 Đồ thị quyền truy nhập cơ sở dữ liệu

7.3.6 Kiểm tra dấu vết

Nhiều ứng dụng về bảo mật cơ sở dữ liệu cần duy trì một cơ chế kiểm tra dấu vết. Một sự kiểm tra dấu vết là một bản lưu tất cả các thay đổi khi thực hiện các phép lưu trữ như chèn thêm, xoá và sửa đổi thông tin trong cơ sở dữ liệu cùng với những thông tin phát sinh thêm trong quá trình thực hiện. Việc kiểm tra dấu vết sẽ giúp cho việc dò tìm được các nguyên nhân nhanh và chính xác. Ví dụ nếu một tài khoản nào đó được phát hiện không cân đối, người quản trị có thể lần dấu vết của tất cả các cập nhật đã xảy ra trong tài khoản để tìm thấy sự cập nhật không đúng (có thể là gian lận) của những người đã thực hiện việc cập nhật. Tạo ra một sự kiểm tra dấu vết bằng cách định nghĩa các chuỗi phản ứng thích hợp trên các cập nhật quan hệ (sử dụng hệ thống các giá trị đã định nghĩa để nhận biết tên người sử dụng và lần truy nhập). Tuy nhiên nhiều hệ thống cơ sở dữ liệu cung cấp phương pháp tạo sự kiểm tra dấu vết thuận tiện và dễ sử dụng.

7.4 Tính bảo mật trong SQL

7.4.1 Các đặc quyền trong SQL

Các phép toán đặc quyền cơ bản trong SQL bao gồm các phép DELETE, INSERT, SELECT và UPDATE. Quyền của phép chọn tương ứng với quyền đọc và quyền tham chiếu. Ngôn ngữ định nghĩa dữ liệu SQL bao gồm những lệnh cấp phát quyền GRANT và lệnh thu hồi, hủy bỏ các quyền đã cấp phát REVOKE .

1. *Cấp quyền truy xuất một quan hệ cho người sử dụng hay nhóm người sử dụng:* Dạng cơ bản của câu lệnh này là :

GRANT <privilege list > ON <relation name or view name TO <user/role list > .

Privilege list cho phép cấp một số quyền sử dụng trong một lệnh.

- Cấp quyền READ cho nhóm người sử dụng WORLD trên quan hệ R. Người sử dụng chỉ được quyền bao gồm quyền sử dụng quan hệ R trong các câu vấn tin, không được phép sửa đổi, bổ sung.

GRANT READ ON R TO GROUP/ WORLD

- Cấp quyền quyền ưu tiên WRITE: Chỉ có quyền ghi, sửa đổi, cập nhật. Thông tin không được hiển thị:

GRANT WRITE ON R TO GROUP /WORLD

- Cấp quyền được đọc và ghi ALL: Người sử dụng được quyền ghi và đọc bao như quyền chèn, xóa và sửa đổi các bộ cũng như các thao tác tạo chỉ mục các quan hệ hay xóa bỏ chính quan hệ.

GRANT ALL ON R TO GROUP /WORLD

- Người quản trị CSDL có đầy đủ tất cả các quyền. Vì vậy không cần thiết phải mô tả tường minh.
- Cấp quyền SELECT trên quan hệ ACCOUNT cho U₁, U₂ và U₃

GRANT SELECT ON ACCOUNT TO U1, U2, U3.

- Quyền UPDATE mô tả hoặc trên tất cả các thuộc tính hoặc trên một số thuộc tính của quan hệ. Nếu quyền UPDATE trên một số thuộc tính, thì danh sách các thuộc

tính sẽ được mô tả trong dấu ngoặc ngay sau từ khoá UPDATE . Nếu không mô tả các thuộc tính thì quyền cập nhật của người sử dụng sẽ được hiểu trên tất cả những thuộc tính của quan hệ.

GRANT UPDATE (AMOUNT) ON LOAN TO U1, U2, U3

GRANT UPDATE ON LOAN TO U1, U2, U3

- Quyền INSERT: Khi cấp quyền chèn thêm vào quan hệ phải xác định rõ chèn giá trị tương ứng trên những thuộc tính nào vào hệ thống.
- Các quyền tham chiếu REFERENCE được cấp cho người sử dụng trên những thuộc tính cụ thể giống như quyền UPDATE. Cấp quyền tham chiếu cho người sử dụng U1 trên thuộc tính branch-name đến quan hệ branch.

GRANT REFERENCE (branch-name) ON BRANCH TO U1.

2. Quy tắc đặc quyền người sử dụng

- CREATE ROLE TELLER
GRANT SELECT ON ACCOUNT TO TELLER
- GRANT TELLER TO JOHN
CREATE ROLE MANAGER
GRANT TELLER TO MANAGER
GRANT MANAGER TO MARY

Theo mặc định, một người hoặc một nhóm người sử dụng được cấp một đặc quyền, khi đó người hay nhóm này không được quyền cấp đặc quyền này tiếp cho người hay nhóm người sử dụng khác. Nếu muốn cấp một đặc quyền sử dụng và để cho phép người nhận tiếp tục vượt qua đặc quyền tới người sử dụng khác, cần thêm mệnh đề WITH GRANT OPTION tới mệnh lệnh GRANT tương ứng. Ví dụ, giả người sử dụng U1 có quyền thực hiện phép chọn trên quan hệ BRANCH và cho phép U1 có quyền cấp đặc quyền này tới người khác, có thể biểu diễn như sau:

GRANT SELECT ON BRANCH TO U1 WITH GRANT OPTION

3. Thu hồi quyền truy nhập:

Để thu hồi một quyền truy nhập, cấu trúc lệnh tổng quát như sau:

REVOKE <privilege list> ON <relation name or view name>

FROM <user/role list> [restrict | cascade]

- Thu hồi đặc quyền đã cấp phép cuối cùng cho U₁, U₂ và U₃ .

GRANT SELECT ON ACCOUNT TO U1, U2, U3.

REVOKE SELECT ON BRANCH FROM U1, U2, U3

REVOKE REFERENCES (branch-name) ON BRANCH FROM U1

- Lệnh thu hồi một đặc quyền cũng có hiệu lực đối với trường hợp một người hay một nhóm người sử dụng cấp tiếp cho một hay một nhóm người sử dụng khác. Trường hợp này được gọi là *cascading of the revoke*.

REVOKE SELECT ON BRANCH FROM U1, U2, U3 RESTRICT

Nghĩa là, hệ thống sẽ thu hồi quyền SELECT không những của U1, U2, U3, mà còn thu hồi cho những người hay nhóm người sử dụng được U1, U2, U3 cấp tiếp quyền SELECT.

```
REVOKE GRANT OPTION FOR SELECT ON BRANCH FROM U1
REVOKE READ/WRITE/ALL ON R .FROM GROUP/WORLD
```

7.4.2 Hạn chế các quyền hạn trong SQL

Các quyền hạn trong SQL còn một số nhược điểm. Ví dụ, giả sử bạn muốn tất cả các sinh viên có thể nhìn thấy được các điểm của chính họ, nhưng không thấy được điểm của những người khác. Như vậy, quyền hạn phải đặt ở mức các bộ dữ liệu cá nhân, điều này không thực hiện được trong SQL. Hơn nữa, kỹ thuật lập trình Web - cơ sở dữ liệu đã phát triển trong đối mạnh. Các ứng dụng truy xuất vào cơ sở dữ liệu và hiển thị các kết quả trên các thiết bị đầu cuối dưới dạng HTML ngày càng tăng. Vấn đề an toàn, bảo mật thông tin trên các máy chủ Web servers, Data Server trên đường truyền ngày càng khó khăn và phức tạp. Nhiệm vụ của các quyền truy nhập được đặt vào trong các ứng dụng server. Mô hình quyền hạn đầy đủ của SQL nhiều khi phải đi đường vòng. Lợi thế là các quyền hạn được làm nổi bật như là những bộ dữ liệu cá nhân, có thể được thi hành bởi các ứng dụng.

7.4.3 Ví dụ định nghĩa các quyền truy nhập

1. Giả sử, trưởng phòng tổ chức cán bộ của một cơ quan mới được quyền xem và sửa đổi nội dung nhận xét chất lượng làm việc của các nhân viên trong cơ quan.
2. Nhân viên quản lý nhân sự có quyền đọc nhận xét, có quyền xoá hoặc thêm thông tin về nhân sự mới, hoặc cập nhật, bổ sung các thông tin về mã nhân viên, họ và tên, địa chỉ, đơn vị công tác, ngày tháng giữ bậc lương mới.
3. Các nhân viên còn lại của phòng tổ chức cán bộ chỉ có quyền được xem mã nhân viên, họ và tên, địa chỉ nơi ở, đơn vị công tác của nhân viên.

Trưởng phòng có quyền truy xuất vào QLNS, do thoả điều kiện (1), nhưng không thoả điều kiện (2) và (3). Các nhân viên còn lại có thể truy xuất theo điều kiện (2) và (3). Người quản trị CSDL cần tạo ra một khung nhìn, mọi nhân viên của phòng có thể đọc được nội dung của khung này. Khung nhìn được định nghĩa như sau:

```
CREATE VIEW PUBLIC_QLNS (MA, HVT, ĐC, DV) AS
SELECT MA, HVT, ĐC, DV
FROM QLNS.
```

Khi chèn thêm thông tin về một nhân viên mới vào quan hệ QLNS, giá trị của trường nhận xét NHAN_XET sẽ nhận giá trị null. Xoá trong khỏi quan hệ QLNS nhân viên có mã và tên trong khung. Khi sửa đổi, bổ sung cũng tương tự cho các bộ thích hợp.

Để gán quyền truy xuất dữ liệu trên quan hệ QLNS cho một nhóm chuyên viên quản lý nhân sự, trưởng phòng có thể đưa ra câu lệnh sau:

```
GRANT READ ON QLNS TO GROUP;
GRANT READ ON PUBLIC_QLNS TO WORLD;
GRANT WRITE ON PUBLIC_QLNS TO GROUP.
```

7.5 Tóm tắt

- Vấn đề an toàn và bảo toàn dữ liệu không bảo đảm khi quá trình biến đổi dữ liệu không được tôn trọng, phá vỡ tính nhất quán của dữ liệu bởi người sử dụng.
- Những ràng buộc về miền xác định liên quan đến giá trị của các thuộc tính có thể dẫn đến các giá trị rỗng hoặc không xác định.
- Ràng buộc tính tham chiếu toàn vẹn sẽ bị vi phạm khi giá trị xuất hiện tập các thuộc tính cho trước của một quan hệ đồng thời cũng xuất hiện tập các thuộc tính của một quan hệ khác.
- Những ràng buộc miền xác định và những ràng buộc tham chiếu toàn vẹn có thể kiểm tra khá dễ dàng. Việc sử dụng nhiều những ràng buộc phức tạp có thể đưa tới giá trị đáng kể. Có 2 cách để thể biểu diễn các ràng buộc tổng quát.
- Dữ liệu được lưu trữ trong cơ sở dữ liệu cần được bảo vệ khỏi các truy cập trái phép, sự phá hoại hoặc các thay đổi và sự ngẫu nhiên của mâu thuẫn.
- Việc chống lại sự mất mát ngẫu nhiên của dữ liệu dễ hơn là sự chống lại những truy cập trái phép vào cơ sở dữ liệu. Không thể tuyệt đối bảo vệ cơ sở dữ liệu khỏi các truy cập trái phép, nhưng có thể ngăn chặn hầu hết.
- Người sử dụng có thể có rất nhiều quyền truy nhập vào các phần khác nhau của cơ sở dữ liệu. Các quyền này là phương tiện để hệ thống cơ sở dữ liệu có thể chống lại những truy cập không cho phép và nguy hiểm.
- Người sử dụng có quyền có thể chuyển giao quyền truy nhập tới những người sử dụng khác. Tuy nhiên, cũng cần phải cẩn trọng trong việc chuyển những quyền này như thế nào trong số những người sử dụng nếu như có thể chắc chắn rằng sự cho phép này có thể được thu hồi lại tại một thời điểm nào đó trong tương lai.

Câu hỏi trắc nghiệm

1. “An toàn” dữ liệu có nghĩa là cơ sở dữ liệu...
 - A. Cần phải được bảo vệ chống truy nhập trái phép.
 - B. Chống sửa đổi hay phá hoại.
 - C. Chống vi phạm chủ định hay không chủ định.
 - D. Cần thiết phải quản trị, bảo vệ tập trung.
2. Có thể xảy ra không an toàn thông tin
 - A. Những sai sót ngoài ý muốn khi thêm, sửa, xoá.
 - B. Truy nhập trái phép với mục đích xấu
 - C. Sự cố kỹ thuật như lỗi thiết bị, lập trình...
 - D. Cần thiết phải quản trị, bảo vệ tập trung.
3. Để bảo vệ cơ sở dữ liệu, phải thực hiện một số biện pháp an toàn :
 - A. Hệ thống cơ sở dữ liệu:
 - B. Người quản trị cơ sở dữ liệu cấp phép truy nhập.

- C. An toàn hệ thống điều hành
 - D. Mạng
4. Một số biện pháp bảo vệ cơ sở dữ liệu :
- A. Nhận diện người sử dụng:
 - B. Bảo vệ mức vật lý:
 - C. Kiểm tra truy nhập.
 - D. Người quản trị cơ sở dữ liệu cấp phép truy nhập.
5. Giá trị của khoá không được nhận.....hay giá trị không xác định. Hệ quản trị CSDL có trách nhiệm phải từ chối khi giá trị của khoá nếu vẫn còn chứa các giá trị chưa xác định hoặc đã tồn tại trong cơ sở dữ liệu thông qua các phép thêm hoặc sửa đổi dữ liệu.
- A. Giá trị null.
 - B. Giá trị không.
 - C. Số không.
 - D. Rỗng
6. Có thể chia quyền hạn truy nhập vào cơ sở dữ liệu.như sau
- A. Đọc, chèn, sửa đổi, xoá một cách hợp pháp
 - B. Cho phép việc tạo các mối quan hệ mới.
 - C. Sửa đổi cấu trúc, bỏ hợp pháp
 - D. Người quản trị cơ sở dữ liệu cấp phép truy nhập.
7. Hãy chọn từ/cụm từ tương ứng để hoàn thiện khẳng định sau: Cơ sở dữ liệu làcho nhiều người cùng sử dụng. Vì vậy các hệ cơ sở dữ liệu cần phải có cơ chế kiểm soát, quản lý và truy xuất khai thác sao cho dữ liệu phải được an toàn và toàn vẹn.
- A. Tài nguyên thông tin chung.
 - B. Thông tin
 - C. Tài nguyên.
 - D. Dữ liệu
8. Các ràng buộc toàn vẹn dữ liệu phải bảo đảm tính độc lập và Dữ liệu lưu trữ trong các hệ thống cơ sở dữ liệu phải phản ánh hiện thực khách quan trong mọi thời điểm có thể, không dư thừa và mâu thuẫn thông tin.
- A. Toàn vẹn dữ liệu
 - B. Dị thường dữ liệu
 - C. Bảo vệ dữ liệu
 - D. An toàn dữ liệu
9. Hãy so khớp thông tin 2 phần dưới đây với nhau:

Phần 1

1 Vấn đề toàn vẹn dữ liệu là đảm bảo dữ liệu trong

Phần 2

A. các biện pháp bảo vệ dữ liệu, chống lại sự sửa đổi hay phá hoại không chủ định.

2. Bảo đảm tính toàn vẹn của dữ liệu là một trong
3. Có thể có nhiều nguyên nhân....
4. Không đảm bảo tính toàn vẹn dữ liệu.....
- B. dẫn đến không được đảm bảo tính toàn vẹn dữ liệu.
- C. có thể xảy ra trong hệ thống coi là ổn định nhất và đảm bảo nhất
- D. các cơ sở dữ liệu chính xác tại mọi thời điểm.
- a. 1- D; 2- C; 3 – B; 4 – A .
- b. 1- C; 2- B; 3 – A; 4 – D
- c. 1- C; 2- D; 3 – A; 4 – B .
- d. 1- D; 2- A; 3 – B; 4 – C.

10. Hãy so khớp thông tin 2 phần dưới đây với nhau:

Phần 1

1. Xác định cho hệ thống những thao tác.
2. Cung cấp một phương tiện cho.....
3. Hình thức thông dụng nhất....
4. Mật khẩu cũng được hệ thống bảo.....
- a. 1- D; 2- C; 3 – B; 4 – A .
- b. 1- B; 2- C; 3 – D; 4 – A
- c. 1- C; 2- D; 3 – A; 4 – B .
- d. 1- D; 2- A; 3 – B; 4 – C.

Phần 2

- A. vệ như bảo vệ dữ liệu.
- B. người sử dụng được phép thực hiện.
- C. người sử dụng để hệ thống nhận biết
- D. để nhận ra người sử dụng là mật khẩu.

Câu hỏi & bài tập

1. Vì sao phải bảo vệ an toàn các hệ thống cơ sở dữ liệu.
2. Hiểu tính toàn vẹn dữ liệu như thế nào. Ví dụ minh họa.
3. Các ràng buộc toàn vẹn. Ví dụ minh họa.
4. Ngôn ngữ vấn tin - ngôn ngữ ràng buộc toàn vẹn.
5. Kiểm tra các ràng buộc toàn vẹn.
6. Khái niệm phụ thuộc hàm đơn ngôi (unary inclusion dependency).
7. Trình bày kiểm tra ràng buộc tồn tại nhờ các dẫn xuất.
8. Tính bảo mật.
9. Nhận diện người sử dụng.
10. Bảo vệ vật lý.
11. Kiểm tra truy nhập.
12. Khung nhìn – Một cơ chế bảo vệ.
13. Sử dụng ngôn ngữ vấn tin định nghĩa các quyền truy nhập
14. Tính bảo mật trong SQL.

Tài liệu tham khảo

- [1] Codd E.F., “Relational model of data for large share banks”, Comm, ACM. 13,6 (June 1970).
- [2] Date C.J., “ An introduction to data base systems”, Second editon 1977.
- [3] Codd, E.F., “ Data models in data base management”, ACM SIGMOD record,11,2(Feb,1981).
- [4] Aho,A.V., Hocrof, J.E., and Ullman, J.D. “The design of analysic of computer Algorithms” .Addision-Wesley,Reading 1974.
- [5] Beeri, C., and Bernstein “Computational problems related to the design of normal form relation schemes”, ACM on database Systems 4:1, pp 30-39 (1979).
- [6] Beeri, C. “ On the mebership problem for functional and multivaluted dependencies”, ACM trans on database 5:3,pp 241-259 (1980).
- [7] Armstrong , W.W “Dependency structures of data base relationaships”, Proc 1974 EFIP Congress,pp 580-583,Nort Holland, Amsterdam, (1974).
- [8] Hồ Thuần và Lê Văn Bào, “ Khoá của lược đồ quan hệ”, Khoa học kỹ thuật & tự động hoá chỉ huy 2/1983.
- [9] Ullman, J. “Principles of database systems”, Prentice Hall,1980.
- [10] Michanel V. Mannino, “ Database Application Development & Design”, Published by McGaw-Hill /Irwin, New Yor.k, 2001.
- [11] Abram Siberschatz, Henry F.Korth, S.Sudarshan “ Database Systems Concepts”, Published by McGaw-Hill /Irwin, New Yor.k, 2002.

MỤC LỤC

Chương I: KHÁI NIỆM CƠ BẢN VỀ CÁC HỆ CƠ SỞ DỮ LIỆU	
1.1 Cơ sở dữ liệu là gì.....	2
1.2 Sự cần thiết của các hệ cơ sở dữ liệu.....	3
1.3 Mô hình kiến trúc tổng quát cơ sở dữ liệu 3 mức.....	4
1.4 Mục tiêu của các hệ cơ sở dữ liệu.....	7
1.5 Hệ quản trị CSDL & người quản trị CSDL.....	9
1.6 Ràng buộc dữ liệu.....	10
1.7 Các mô hình truy xuất dữ liệu.....	11
1.7.1 Mô hình cơ sở dữ liệu Client Server.....	11
1.7.2 Mô hình Client/Server nhiều lớp.....	12
1.7.3 Kỹ thuật lập trình cơ sở dữ liệu - Web động.....	13
1.7.4 Kiến trúc hệ thống Server (Server System Architecture).....	14
1.7.5 Các mô hình kiến trúc ứng dụng.....	16
Câu hỏi trắc nghiệm.....	17
Câu hỏi & bài tập.....	18
Chương II CÁC MÔ HÌNH CƠ SỞ DỮ LIỆU	
2.1 Mở đầu.....	19
2.2 Mô hình dữ liệu (Data Model).....	19
2.2.1 Phân biệt giữa các mô hình dữ liệu.....	20
2.2.2 Các hệ thống CSDL đối tượng và tri thức.....	20
2.3 Mô hình CSDL phân cấp (Hierarchy Data Model).....	21
2.3.1 Cấu trúc biểu diễn dữ liệu phân cấp.....	21
2.3.2 Ngôn ngữ thao tác trên CSDL phân cấp.....	21
2.4 Mô hình CSDL mạng (Network Data Model).....	23
2.4.1 Cấu trúc biểu diễn dữ liệu mạng.....	23
2.4.2 Ngôn ngữ dữ liệu thao tác trên CSDL mạng.....	25
2.5 Cách tiếp cận mô hình CSDL quan hệ.....	25
2.5.1 Khái niệm lý thuyết tập hợp quan hệ.....	25
2.5.2 Ngôn ngữ thao tác dữ liệu kiểu quan hệ.....	27
2.5.3 Các phép toán cơ bản của đại số quan hệ.....	27
2.5.4 Ưu điểm cách tiếp cận mô hình CSDL quan hệ.....	28
2.6 Mô hình thực thể quan hệ (The Entity Relationship Model).....	28
2.6.1 Khái niệm.....	28
2.6.2 Thực thể, mối quan hệ, thuộc tính.....	28
2.6.3 Bậc của mối quan hệ.....	30

2.6.4	Độ kết nối các mối quan hệ.....	30
2.6.5	Phân cấp ISA.....	30
2.6.6	Sơ đồ thực thể quan hệ.....	30
2.6.7	Các bước triển khai mô hình thực thể quan hệ	34
	Câu hỏi trắc nghiệm.....	39
	Câu hỏi và bài tập	40
Chương III CƠ SỞ DỮ LIỆU QUAN HỆ		
3.1	Quan hệ.....	42
3.1.1	Định nghĩa quan hệ.....	42
3.1.2	Ký hiệu.....	43
3.2	Phụ thuộc hàm.....	44
3.3	Hệ tiên đề các phụ thuộc hàm và các phép suy dẫn logic.....	45
3.3.1	Hệ tiên đề Armstrong cho các phụ thuộc hàm.....	45
3.3.2	Ý nghĩa hệ tiên đề Armstrong.....	45
3.3.3	Các tính chất của phụ thuộc hàm.....	46
3.3.4	Các phép suy dẫn phụ thuộc hàm.....	46
3.3.5	Bao đóng và tính chất bao đóng các phụ thuộc hàm.....	47
3.3.6	Phụ thuộc hàm đầy đủ và không đầy đủ.....	47
3.4	Bao đóng & thuật toán xác định bao đóng các thuộc tính.....	48
3.4.1	Bao đóng tập các thuộc tính (Closure of a set attributes).....	48
3.4.2	Các tính chất bao đóng thuộc tính.....	49
3.4.3	Thuật toán xác định bao đóng.....	49
3.4.4	Ví dụ bao đóng của tập thuộc tính.....	51
3.4.5	Thuật toán xác định phụ thuộc hàm suy dẫn từ F.....	52
3.5	Phụ thuộc dư thừa, tập các phụ thuộc tương đương.....	54
3.5.1	Tập các phụ thuộc tương đương.....	54
3.5.2	Phụ thuộc dư thừa.....	54
3.5.3	Thuộc tính dư thừa.....	55
3.5.4	Tập các phụ thuộc phủ tối thiểu.....	56
3.5.5	Thuật toán 3.4 xác định tập phủ tối thiểu.....	57
3.6	Khoá của lược đồ quan hệ.....	57
3.6.1	Định nghĩa.....	57
3.6.2	Khoá dự bị.....	59
3.6.3	Định thuộc (Determinant).....	59
3.6.4	Phản khoá (Antikey).....	59
3.6.5	Lược đồ thu gọn.....	60
3.7	Một số tính chất của khoá trong lược đồ quan hệ.....	61
3.8	Các thuật toán xác định khoá của lược đồ quan hệ.....	62
	Câu hỏi trắc nghiệm.....	67

Câu hỏi và bài tập	70
Chương IV KỸ THUẬT THIẾT KẾ CÁC HỆ CƠ SỞ DỮ LIỆU QUAN HỆ	
4.1 Mở đầu	76
4.2 Phép tách – kết nối không tổn thất thông tin	76
4.2.1 Phép tách.....	76
4.2.2 Tách - kết nối tự nhiên	77
4.2.3 Phép tách không tổn thất thông tin	77
4.3 Thuật toán kiểm tra tách không tổn thất thông tin.....	79
4.4 Các dạng chuẩn của lược đồ quan hệ.....	81
4.4.1 Sự cần thiết chuẩn hoá dữ liệu	82
4.4.2 Dạng chuẩn 1 – 1NF (First Normal Form)	83
4.4.3 Dạng chuẩn 2 – 2NF (Second Normal Form).....	84
4.4.4 Dạng chuẩn 3 - 3NF (Third Normal Form)	85
4.4.5 Quá trình tách 1NF về nhóm tương đương 3NF.....	86
4.5 Dạng chuẩn Boyce Codd	87
4.5.1 Định nghĩa dạng chuẩn Boyce Codd	87
4.5.2 Nhận biết dạng chuẩn 3NF & BCNF	88
4.6 Thuật toán tách lược đồ quan hệ không tổn thất thông tin	88
4.6.1 Phép tách về các lược đồ dạng chuẩn 3NF có bảo toàn phụ thuộc.....	88
4.6.2 Thuật toán tách bảo toàn phụ thuộc và không tổn thất thông tin về 3NF	89
4.6.3 Thuật toán tách về dạng chuẩn BCNF không tổn thất	90
Câu hỏi trắc nghiệm.....	92
Câu hỏi và bài tập:	94
Chương V CÁC NGÔN NGỮ THAO TÁC DỮ LIỆU	
5.1 Ngôn ngữ cơ sở dữ liệu.....	98
5.1.1 Ngôn ngữ định nghĩa dữ liệu	98
5.1.2 Ngôn ngữ thao tác dữ liệu.....	99
5.2 Các phép toán đại số quan hệ.....	99
5.2.1 Toán hạng của đại số quan hệ	100
5.2.2 Phép hợp các quan hệ - UNION	100
5.2.3 Phép giao của các quan hệ - INTERSECT	100
5.2.4 Hiệu của các quan hệ - MINUS	101
5.2.5 Tích Đề Các – (Cartesian Product).....	101
5.2.6 Phép chọn - Selection.....	102
5.3 Các phép toán đại số bổ sung.....	103
5.3.1 Phép chia - DIVISION.....	103
5.3.2 Phép kết nối – JOIN.....	104
5.4 Các tính chất của đại số quan hệ.....	105
5.4.1 Tính chất giao hoán.....	105

5.4.2	Tính chất kết hợp	106
5.4.3	Tính chất tích lũy đẳng	106
5.4.4	Một số tính chất khác	106
5.5	Đại số quan hệ – ngôn ngữ vấn tin	106
5.5.1	Biểu thức quan hệ	106
5.5.2	Cây đại số quan hệ	107
5.5.3	Cài đặt đại số quan hệ	108
5.6	Ngôn ngữ vấn tin có cấu trúc SQL	113
5.6.1	Giới thiệu	113
5.6.2	Câu lệnh SELECT	113
5.6.3	Biến quan hệ	116
5.6.4	Toán tử LIKE	116
5.6.5	Phép toán tập hợp trong mệnh đề WHERE	116
5.6.6	Các phép toán gộp nhóm	118
5.6.7	Mệnh đề GROUP BY ... HAVING	119
5.6.8	Các phép toán lưu trữ	119
5.6.9	Tính đầy đủ của SQL	121
5.6.10	Ngôn ngữ định nghĩa dữ liệu	122
5.6.11	Khung nhìn View	122
5.6.12	Database catalog	123
	Câu hỏi trắc nghiệm	124
	Câu hỏi & bài tập	126
Chương VI TỐI ƯU HOÁ CÂU HỎI TRUY VẤN		
6.1	Mở đầu	130
6.2	Tổng quan về tối ưu hóa câu hỏi	130
6.3	Phương pháp ước lượng cây đại số quan hệ	132
6.4	Nguyên tắc tối ưu hoá	133
6.5	Kỹ thuật tối ưu hoá các biểu thức đại số quan hệ	133
6.5.1	Biểu thức quan hệ	133
6.5.2	Biến đổi biểu thức quan hệ	134
6.6	Tối ưu hóa bằng khung nhìn (Materialized Views)	138
6.6.1	Phép kết nối	138
6.6.2	Phép chọn và phép chiếu	138
6.6.3	Các phép toán khối	139
6.6.4	Các phép toán khác	139
6.6.5	Tối ưu hóa các câu hỏi và các khung nhìn cụ thể hóa	139
6.7	Ví dụ tối ưu hoá các câu hỏi	140
	Câu hỏi trắc nghiệm	143

Câu hỏi và bài tập:	145
Chương VII AN TOÀN DỮ LIỆU VÀ TÍNH TOÀN VỆ DỮ LIỆU	
7.1 Sự cần thiết phải bảo vệ an toàn cơ sở dữ liệu.....	147
7.2 Tính toàn vẹn dữ liệu.....	147
7.2.1 Các ràng buộc toàn vẹn.....	148
7.2.2 Ngôn ngữ vấn tin - ngôn ngữ ràng buộc toàn vẹn	149
7.2.3 Kiểm tra các ràng buộc toàn vẹn.....	149
7.3 Vấn đề an toàn và quyền truy nhập cơ sở dữ liệu.....	149
7.3.1 Sự vi phạm an toàn cơ sở dữ liệu.....	149
7.3.2 Các mức độ an toàn cơ sở dữ liệu.....	150
7.3.3 Những quyền hạn khi sử dụng hệ cơ sở dữ liệu.....	151
7.3.4 Khung nhìn –một cơ chế bảo vệ	152
7.3.5 Cấp phép các quyền truy nhập.....	153
7.3.6 Kiểm tra dấu vết.....	155
7.4 Tính bảo mật trong SQL	155
7.4.1 Các đặc quyền trong SQL.....	155
7.4.2 Hạn chế các quyền hạn trong SQL	157
7.4.3 Ví dụ định nghĩa các quyền truy nhập	157
7.5 Tóm tắt.....	158
Câu hỏi trắc nghiệm.....	158
Câu hỏi & bài tập.....	160
Mục lục	161
Tài liệu tham khảo.....	166

CƠ SỞ DỮ LIỆU

Mã số: 492CDL351

Chịu trách nhiệm bản thảo

TRUNG TÂM ĐÀO TẠO BƯU CHÍNH VIỄN THÔNG 1

(Tài liệu này được ban hành theo Quyết định số: 827 /QĐ-TTĐT1 ngày 30/10/2006 của Giám đốc Học viện Công nghệ Bưu chính Viễn thông)