

GIÁO TRÌNH

CƠ SỞ DỮ LIỆU

Mục lục

Mục lục.....	2
LỜI NÓI ĐẦU.....	6
Chương I.....	7
Các khái niệm cơ bản.....	7
I. DỮ LIỆU (DATA), TẬP DỮ LIỆU, CSDL & CÁC TÍNH CHẤT.....	7
I.1 Dữ liệu (Data):.....	7
I.2. File dữ liệu:.....	8
I.3. Cơ sở dữ liệu (Database):.....	8
I.4 Các vấn đề cần quan tâm khi nghiên cứu CSDL:.....	8
II. THỰC THỂ, THUỘC TÍNH. MỐI LIÊN HỆ GIỮA CÁC THỰC THỂ:.....	8
II.1.Thực thể (Entity):.....	8
II.3 Ràng buộc (Constraint):.....	9
II.4 Khóa (key) của một thực thể:.....	11
II.5 Phân cấp ISA :.....	11
III. KIẾN TRÚC CỦA MỘT HỆ CSDL.....	11
IV. CÁC MÔ HÌNH DỮ LIỆU.....	12
IV.1 Mô hình dữ liệu:.....	12
V. KIẾN TRÚC CƠ SỞ DỮ LIỆU:.....	20
V.1.Cơ sở dữ liệu tập trung:(Centralized database).....	20
V.2.Cơ sở dữ liệu phân tán: (Distributed database).....	21
VI. CÁC NGÔN NGỮ CSDL :.....	21
VII. HỆ QUẢN TRỊ CSDL, CHỨC NĂNG CỦA HỆ QUẢN TRỊ CSDL.....	22
VII.2. Những khả năng của hệ quản trị CSDL :.....	25
VII.3. Lịch sử các hệ quản trị CSDL :.....	25
VII.4. Các chức năng của hệ CSDL.....	26
VII.5. Chuyển hóa dữ liệu :.....	26
VII.6 Điều khiển tính toàn vẹn của dữ liệu :.....	26
VII.7. Quản lý các giao tác và an toàn dữ liệu :.....	27
VIII. NHỮNG KHÓ KHĂN KHI XÂY DỰNG CSDL:.....	27
Bài tập.....	28
Chương II.....	30
Mô hình dữ liệu quan hệ.....	30
I. TOÁN HỌC HÓA MÔ HÌNH QUAN HỆ :.....	30
I.1 Miền (Domain):.....	30
I.2 Quan hệ (Relation):.....	30
I.3 Lược đồ quan hệ (Relation Scheme):.....	31
I.4 Lược đồ cơ sở dữ liệu và Cơ sở dữ liệu:.....	31

II. KHÓA (Keys).....	32
II.1 Khoá của một quan hệ:.....	32
II.2 Khoá của một lược đồ quan hệ:.....	32
II.3 Khoá ngoại (Foreign key).....	32
III. RÀNG BƯỚC TOÀN VỆN.....	33
III.1 Định nghĩa 1:.....	33
III.2 Định nghĩa 2:.....	33
III.3 Ràng buộc toàn vẹn trên 1 quan hệ.....	33
IV. Các phép toán quan hệ : (Relational Operators).....	34
IV.1 Các phép toán logic (Boolean Operations).....	34
IV.2 Phép chọn (The Select Operator).....	34
IV.3 Phép chiếu (Projection).....	35
IV.4 Tích Descaster :(Cartesian Product).....	36
IV.5 Phép dán bộ.....	36
IV.6 Phép nối tự nhiên (Natural Join).....	36
IV.7 Phép nối nửa (Semi Join).....	37
IV.8 Phép chia (Division).....	37
Bài tập.....	38
Chương III.....	40
Lý thuyết thiết kế CSDI quan hệ.....	40
I. PHỤ THUỘC HÀM (Functional Dependencies) :.....	41
I.1 Định nghĩa : Phụ thuộc hàm.....	41
I.2 Định nghĩa: Bao đóng tập phụ thuộc hàm.....	41
I.3 Định nghĩa : Bao đóng tập thuộc tính.....	41
I.4 Bổ đề 1:.....	42
I.5 Định nghĩa : suy dẫn theo quan hệ.....	42
I.6 Bổ đề 2:.....	42
I.7 Định lý: Hệ tiên đề Armstrong là đúng.....	42
I.6 Tính đầy đủ của hệ tiên đề Armstrong :.....	43
I.7 Tính bao đóng của tập các thuộc tính đối với một tập phụ thuộc hàm :..	44
I.8 Tính đúng của thuật toán tìm bao đóng tập thuộc tính.....	44
II. PHỦ CÁC PHỤ THUỘC HÀM: (Covers for functional Dependencies)....	45
II.1. Định nghĩa : Hai tập phụ thuộc hàm tương đương.....	45
II.2. Định lý : Điều kiện 2 tập phụ thuộc hàm tương đương.....	45
II.3. Các thuật toán liên quan kiểm tra sự tương đương của hai phụ thuộc hàm:.....	45
III.4. Bổ đề 4:.....	46
III.5 Định nghĩa : Tập phụ thuộc hàm cực tiểu.....	46
III.6 Định nghĩa : Phủ cực tiểu của 1 tập phụ thuộc hàm.....	46
III.7 Định lý : Tồn tại phủ tối thiểu của 1 tập phụ thuộc hàm.....	46

III.8 Thuật toán tìm phủ tối thiểu: (Bernstein, 1976).....	47
IV. KHÓA – THUẬT TOÁN XÁC ĐỊNH KHÓA.....	49
IV.1 Thuật toán : Tìm khóa tối thiểu của 1 sơ đồ quan hệ.....	49
IV.2 Định lý: Thuật toán xác định khoá tối thiểu của sơ đồ quan hệ IV.1 là đúng.....	50
IV.3 Thuật toán : Tìm khóa tối thiểu của 1 quan hệ r.....	50
IV.4 Một số tính chất của khoá và siêu khoá:.....	51
IV.5 Bổ đề 5:.....	51
IV.6 Bổ đề 6:.....	52
IV.7 Định lý (ĐK cần) [Hồ Thuận & Lê Văn Bào, 1985].....	52
IV.8 Hệ quả:.....	53
IV.9 Thuật toán cải biên để tìm khoá của một sơ đồ quan hệ:.....	53
IV.10 Định lý : [Lucchesi & Osborn – 1979].....	53
V. PHÂN RÃ THÔNG TIN (Decomposition):.....	54
V.1. Phân rã không tổn thất thông tin (Lossless joint decomposition).....	54
V.2 Thuật toán kiểm tra phân rã tổn thất thông tin:.....	55
V.3 Định lý: Phép tách bảo toàn.....	55
V.4 Phân rã bảo toàn phụ thuộc hàm.....	56
VI. CHUẨN HÓA LƯỢC ĐỒ QUAN HỆ:.....	58
VI.1 Ba dạng chuẩn đầu tiên:.....	58
VI.2 Phân rã thành dạng chuẩn 3 (3NF) :.....	61
VI.3 Chuẩn Boyce-Codd :.....	62
VI.4 Phép tách lược đồ quan hệ thành BCNF (T.Son Fischer – 1982):.....	63
VII. Phụ thuộc đa trị (Multivalued Dependency) :.....	64
VIII. Dạng chuẩn bốn (Fourth Normal Form- 4NF).....	69
Bài tập.....	70
Chương IV.....	74
Ngôn ngữ SQL-Tối ưu hóa câu hỏi.....	74
I. NGÔN NGỮ SQL.....	74
IV.1 Câu lệnh tạo dữ liệu.....	75
V.2 Thao tác trên bảng dữ liệu:.....	76
IV.3 Hỏi đáp dữ liệu.....	77
II. TỐI ƯU HOÁ CÂU HỎI.....	81
Bài tập.....	83
Tài liệu tham khảo.....	84

LỜI NÓI ĐẦU

Cơ sở dữ liệu (*Database*) là một trong những lĩnh vực quan trọng của Công Nghệ Thông Tin. Hiện nay, đã có một số tài liệu tham khảo chuyên ngành được biên soạn bằng tiếng Việt. Tuy nhiên, các tài liệu này hoặc được lược dịch, hoặc được viết theo một chủ đích nhất định, điều này làm cho người mới tiếp cận cảm thấy khó khăn do sự logic của các vấn đề bị phá vỡ.

Với mong muốn được đóng góp một bài giảng tương đối khá đầy đủ cho sinh viên ngành Công Nghệ Thông Tin của ĐH Nha Trang nói riêng, và cho những ai quan tâm. Chúng tôi mạnh dạn biên soạn giáo trình này. Những kiến thức trong giáo trình là sự đúc kết chủ yếu từ hai tài liệu của hai tác giả *Ullman, D.Maier*. Hệ thống bài tập được chúng tôi sưu tập từ các tài liệu của các trường Đ.H trong và ngoài nước, đặc biệt là các đề thi tuyển cao học, nghiên cứu sinh trong những năm gần đây.

Ước muốn thì nhiều, nhưng khả năng còn có hạn ! Có thể do tác giả chưa am tường hay nhìn nhận vấn đề chưa đúng. Rất mong những ai quan tâm nếu thấy những hạn chế của giáo trình xin gửi đến tác giả qua Email: ngducthuan@gmail.com

Xin chân thành cảm ơn!

NGUYỄN ĐỨC THUẦN
Bộ Môn Hệ Thống Thông Tin
Khoa Công Nghệ Thông Tin – Đại Học Nha Trang

Chương I

Các khái niệm cơ bản

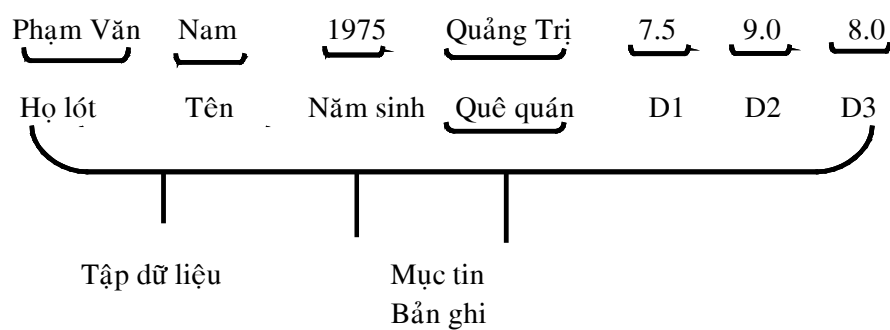
I. DỮ LIỆU (DATA), TẬP DỮ LIỆU, CSDL & CÁC TÍNH CHẤT

I.1 Dữ liệu (Data):

Dữ liệu là tập các sự kiện thô, chúng được tổ chức ở các dạng logic. Thành phần nhỏ nhất của dữ liệu được “thừa nhận” bởi máy tính là các ký tự đơn, (như chữ A, số 5, hay 1 vài ký tự như /, #, %..). Một ký tự đơn được lưu trữ bởi 1 byte.

-Mục tin ¹(Data Item): là đơn vị bé nhất của dữ liệu có một tên.

Ví dụ 1: Ta có thông tin của một học sinh gồm :



Hình 1 Các mục tin của 1 bản ghi

- Tập dữ liệu (Data Aggregate): tập hợp các mục tin xếp tuần tự trong dữ liệu và giữ một tên. Nó thường có hai dạng : vector hay mảng, và nhóm lặp.

¹ Trường, cột, thuộc tính

Ví dụ 2: Trong thông tin mô tả ở hình 1
Họ lót và tên là một tập dữ liệu
D1, D2, D3 là một nhóm lặp chỉ các điểm số.

- **Bản ghi**²(*Record*): tập các dữ liệu gộp lớn và các mục tin xếp kế tiếp nhau của thông tin về một đối tượng, tạo thành đơn vị trao đổi giữa cơ sở dữ liệu và các ứng dụng.

I.2. File dữ liệu:

tập hợp các bản ghi có cùng một cấu trúc xác định, thường được lưu trữ thành từng file.

I.3. Cơ sở dữ liệu (*Database*):

một tập hợp dữ liệu được tổ chức có cấu trúc, được lưu trữ trên những thiết bị trữ tin thỏa mãn một cách đồng thời và có chọn lọc cho nhiều người dùng khác nhau và cho những mục đích khác nhau.

I.4 Các vấn đề cần quan tâm khi nghiên cứu CSDL:

- . Giảm bớt sự dư thừa dữ liệu
- . Tránh được (chừng mực nào đó) tính không nhất quán trong dữ liệu lưu trữ.
- . Dữ liệu có thể được chia sẻ cho nhiều người dùng.
- . Đảm bảo được sự tuân thủ các tiêu chuẩn thống nhất.
- . Có thể áp dụng được các biện pháp an toàn dữ liệu.
- . Giữ được sự toàn vẹn dữ liệu.
- . Điều khiển tương tranh.
- . Phục hồi dữ liệu.

II. THỰC THỂ, THUỘC TÍNH. MỐI LIÊN HỆ GIỮA CÁC THỰC THỂ:

II.1. Thực thể (*Entity*):

là một hình ảnh tượng trưng cho một đối tượng cụ thể hay một khái niệm trừu tượng nhưng có mặt trong thế giới thực.

Ví dụ 3: Sinh viên, Môn học, Giáo viên, Khoa, .. là các thực thể chúng ta cần khảo sát trong bài toán tuyển sinh.

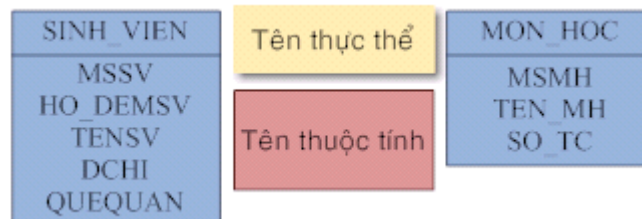
Dự án, Sản phẩm, hóa đơn .. là các thực thể trong bài toán quản lý kinh doanh.

² Dòng, bộ

II.2 Thuộc tính (Attribute):

Đó là các đặc trưng của thực thể mà chúng ta cần mô tả, và lưu trữ để xử lý. Để phân biệt các thực thể người ta dựa vào các thuộc tính.

Ví dụ 4: Thực thể Sinh viên, Môn học

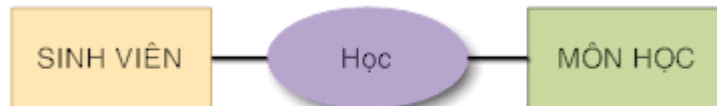


Giá trị các thuộc tính của một thực thể cho phép diễn tả một trường hợp cụ thể của thực thể, gọi là một thể hiện (instance) của thực thể đó.

Ví dụ 5: (49D1234, Trần Văn, Minh, 215 Trần Hưng Đạo, Hà Nam)

II.3 Ràng buộc (Constraint):

Đó là các mối quan hệ logic giữa các thực thể. Kích thước của một mối quan hệ là số thực thể tham gia vào mối quan hệ, và có thể là một số nguyên bất kỳ. Tuy vậy trong thực tế người ta luôn tránh sử dụng các mối quan hệ kích thước lớn 3.



Bản số của mối quan hệ : (cardinary) với R là một mối quan hệ, và E là một thực thể cấu thành của R. Ứng với thực thể E, ta có thể xác định được:

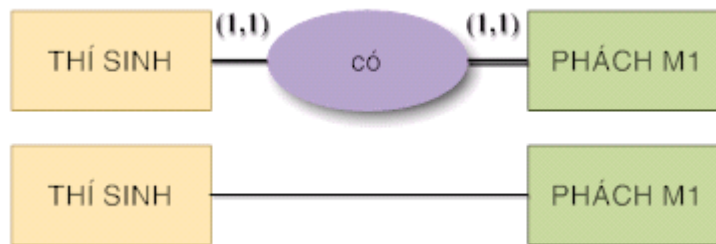
- X là tần suất tối thiểu các thể hiện ứng với E mà R có thể có trong thực tế.
- Y là tần suất tối đa các thể hiện ứng với E mà R có thể có trong thực tế.
- X có thể mang giá trị 0 hay 1.
- Y có thể mang giá trị 1 hay một số nguyên $N > 1$.

Cặp số (X,Y) được định nghĩa là bản số của (E,R) và có thể lấy các giá trị sau: (0,1), (1,1), (0,N) hay (1,N), với $N > 1$.



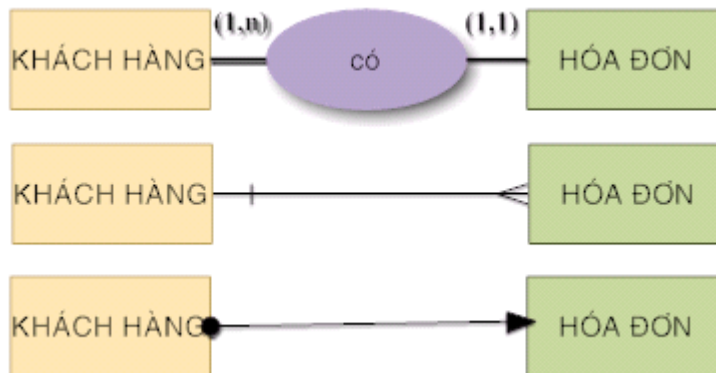
Đối với quan hệ nhị nguyên R liên kết giữa 2 thực thể A và B, ta phân thành 3 loại quan hệ cơ bản sau:

Quan hệ 1-1 (một - một) : mỗi thể hiện của thực thể A được kết hợp với 0 hay 1 thể hiện của B và ngược lại.



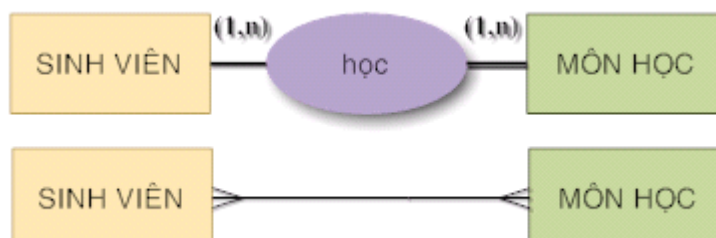
Ví dụ 6: Mỗi thí sinh ứng với mỗi môn thi chỉ có một số phách và ngược lại.

Quan hệ 1-N (một - nhiều) : mỗi thể hiện của thực thể A được kết hợp với 0 hay 1 hay nhiều thể hiện của B và mỗi thể hiện của B được kết hợp duy nhất với một thể hiện của A. Đây là một quan hệ thông dụng và đơn giản nhất.



Ví dụ 7: Một khách hàng có thể có nhiều hóa đơn.
Mỗi hóa đơn chỉ có thể mang tên một khách hàng.

Quan hệ N-P (nhiều - nhiều): mỗi thể hiện của thực thể A được kết hợp với 0 hay 1 hay nhiều thể hiện của B và ngược lại, mỗi thể hiện của thực thể B được kết hợp với 0 hay 1 hay nhiều thể hiện của A.



Ví dụ 8: Một Sinh viên có thể theo học nhiều môn.
Mỗi Môn học có nhiều sinh viên theo học.

Thông thường quan hệ R: N-P chứa các thuộc tính.



II.4 Khóa (key) của một thực thể:

là một thuộc tính hoặc một tập thuộc tính bé nhất dùng để xác định một cách duy nhất mỗi thể hiện trong tập thực thể.

Ví dụ 8: MSSV là khóa của thực thể sinh viên

II.5 Phân cấp ISA :

chúng ta nói rằng thực thể A isa thực thể B nếu tập thực thể B là sự tổng quát hóa của tập thực thể A.

Ví dụ 9: Gọi A là tập các nhân viên phục vụ trong trường X
B là tập cán bộ CNV của trường X thì A **isa** B

III. KIẾN TRÚC CỦA MỘT HỆ CSDL

Kiến trúc của 1 hệ CSDL được ANSI-SPARC (*American National Institute/ System Planning And Requirements Committee*) đề xuất lược đồ 3 mức (*3-tier schema*) vào năm 1978, và được chấp nhận rộng rãi.

Mức biểu diễn ngoài (*External level*) hay gọi là lược đồ ngoài (*External scheme*). Đây là mức đặc tả dữ liệu theo quan niệm của người dùng.

Mức biểu diễn trong (*Internal level*) hay lược đồ vật lý (*physical scheme*) : Đặc tả dữ liệu được lưu trữ phù hợp với thiết bị trữ tin (đĩa từ, băng từ... hoặc tổ chức lưu trữ mã hệ điều hành).

Mức biểu diễn quan niệm (*conceptional scheme*) là quá trình diễn đạt thế giới thực bằng ngôn ngữ định nghĩa dữ liệu.

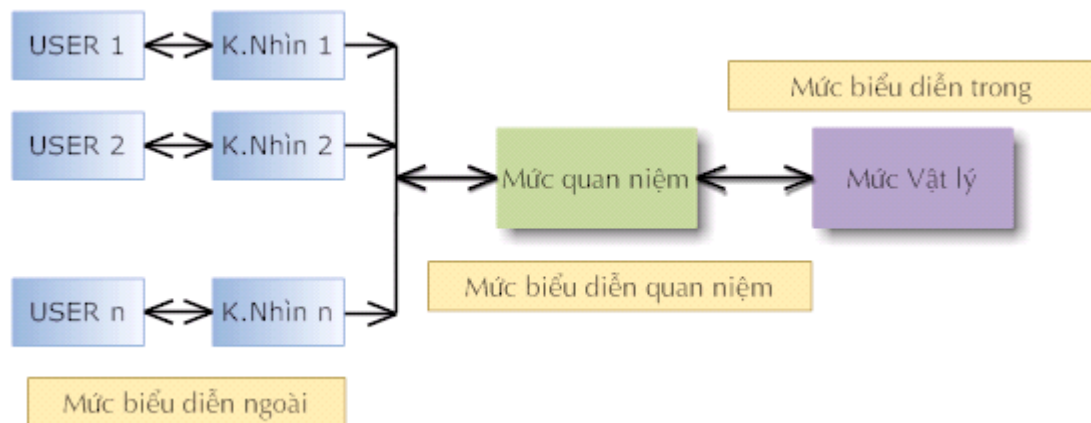
Ví dụ 10:

Ở mức vật lý : Các tập tin lưu trữ hồ sơ nhân viên được lưu trữ lên đĩa từ dưới dạng mã nhị phân.

Ở mức quan niệm : Hồ sơ của 1 cán bộ bao gồm các thuộc tính.

Họ tên, ngày sinh, địa chỉ, mức lương, trình độ, lý lịch.

Ở mức k nhìn : Giả sử đối với người sử dụng cần thống kê cán bộ công nhân viên có tuổi lớn hơn 35, thì sẽ có mức chọn (k nhìn) về tuổi của các cán bộ công nhân viên về tuổi bằng năm hiện thời – năm sinh > 35.



Mô hình kiến trúc của một hệ CSDL

IV. CÁC MÔ HÌNH DỮ LIỆU

IV.1 Mô hình dữ liệu:

Mô hình dữ liệu là một hệ hình thức toán học gồm có 2 phần :

Một hệ thống ký hiệu để mô tả dữ liệu.

Một tập hợp các phép toán thao tác trên dữ liệu đã cho. Đến hiện nay chưa có mô hình dữ liệu nào là tốt nhất cho các hệ thống CSDL.

Các mô hình dữ liệu phổ biến hiện nay :

- **Mô hình thực thể liên hệ** (Entity_Relationship Model)
- **Mô hình dữ liệu quan hệ** (Relational Data Model)
- **Mô hình dữ liệu mạng** (Network data Model)
- **Mô hình dữ liệu phân cấp** (Hierarchical Data Model)
- **Mô hình hướng đối tượng** (Object Orient Model)

Mô hình thực thể_liên hệ (Entity_Relationship Model) ER

Mục đích của mô hình thực thể_liên hệ (ER) là cho phép mô tả lược đồ khái niệm của một tổ chức mà không chú ý đến tính hiệu quả hoặc thiết kế CSDL vật lý. Vì vậy nó không có hệ thống ký hiệu cho các phép toán trên dữ liệu (do đó có tác giả đề nghị không xem chúng là một mô hình dữ liệu).

Thông thường người ta thừa nhận rằng sơ đồ thực thể liên hệ (Entity_Relationship diagram) của một mô hình thực thể liên hệ sẽ được chuyển thành lược đồ của một mô hình khác (ví dụ mô hình dữ liệu quan hệ).

Mô hình ER được Peter Chen đưa ra năm 1976, được sử dụng rộng rãi năm 1988 và ANSI chọn làm mô hình chuẩn cho các hệ thống tự điển tài nguyên thông tin IRDS.

Mô hình ER thể hiện qua sơ đồ thực thể_liên hệ (*Entity – Relationship diagram*) bao gồm các thành phần:

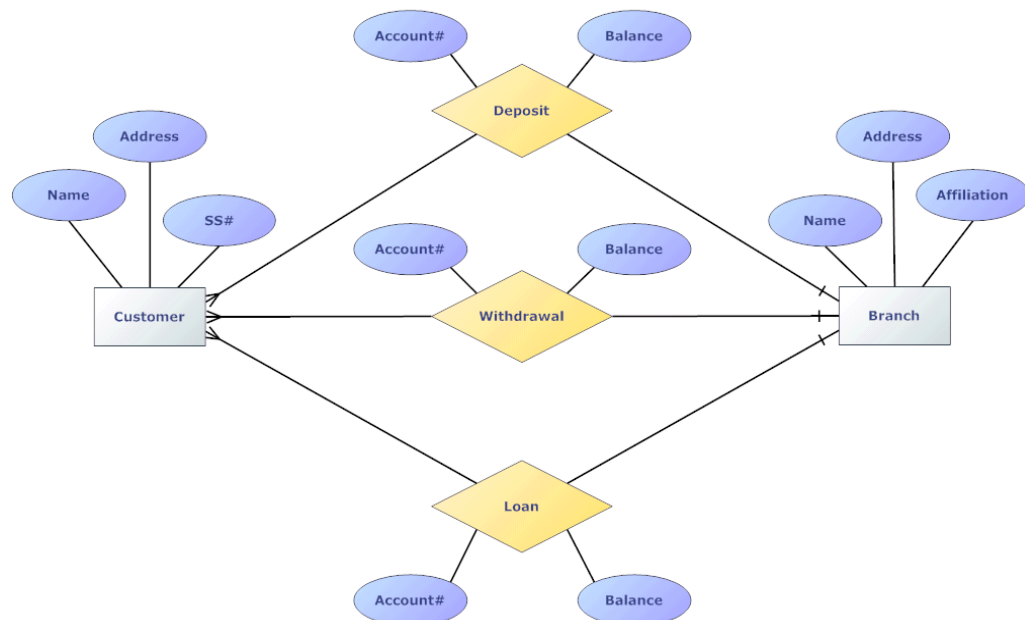
Các hình chữ nhật biểu diễn các tập thực thể

Các hình ellip biểu diễn thuộc tính, hoặc thuộc tính được liệt kê dưới thực thể

Các hình thoi biểu diễn mối quan hệ, người ta cũng có thể biểu diễn bằng hình ellip nếu thuộc tính đã được liệt kê dưới thực thể

Ví dụ 10:

ERD - Banking Transaction



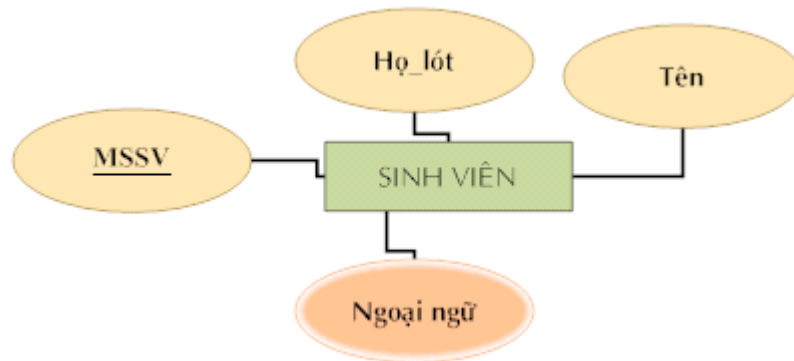
Ưu nhược điểm của mô hình thực thể – liên hệ:

Mô hình hóa thế giới thực dễ dàng, đơn giản, dễ theo dõi. Tuy vậy cùng với thực thể có thể được mô tả bằng nhiều cách khác nhau, do nhận thức chủ quan của mỗi người thiết kế mô hình.

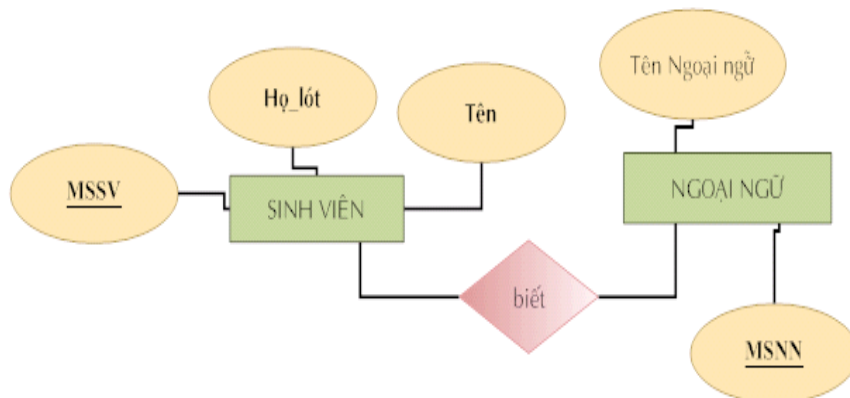
Khi biểu diễn các thực thể có các thuộc tính bội (1 thể hiện của một thực thể có thể ứng với thuộc tính có nhiều giá trị), sẽ gây nhiều khó khăn cho việc chuyển đổi thành mô hình dữ liệu để cài đặt. Vì vậy người ta thường xử lý thuộc tính bội bằng cách thêm kiểu thực thể gồm các thuộc tính bội liên quan và quan hệ với thực thể gốc.

Ví dụ 11 : Xét thực thể **SINHVIEN** có thuộc tính bội **NGOAINGU**:
SINHVIEN (**MSSV**, **HOLOT**, **TEN**, **NGOAINGU**)

Để khử thuộc tính bội NGOAINGU, ta thêm kiểu thực thể NGOAINGU với thuộc tính TENNN, quan hệ giữa thực thể SINHVIEN và NGOAINGU là quan hệ biết.



Thực thể Sinh viên có thuộc tính Ngoại ngữ là đa trị



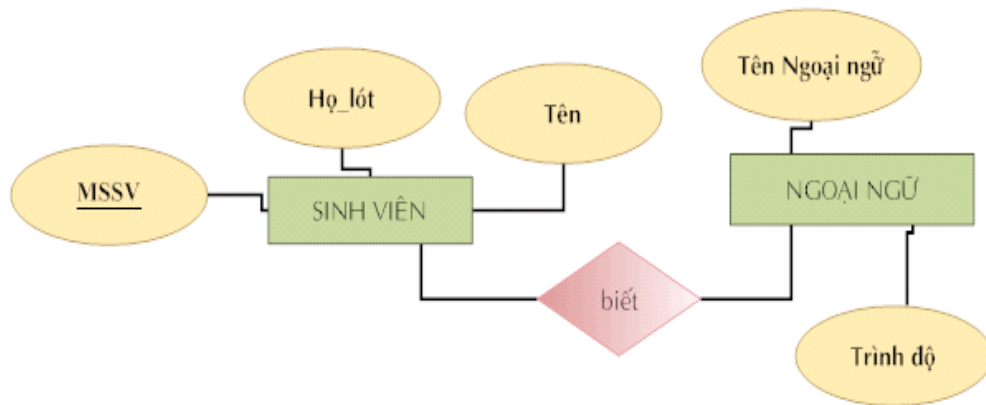
Sau khi khử tính đa trị của thuộc tính Ngoại ngữ

Nhóm lặp: là một tập thuộc tính bội có quan hệ logic với nhau. Để khử các nhóm lặp ta cũng có thể thêm thực thể gồm các thuộc tính bộ liên quan và quan hệ với thực thể gốc.

Ví dụ 12: Xét thực thể SINHVIEN có thuộc tính bội NGOAINGU, NGOAINGU có thể có các trình độ TRINHDO khác nhau.

SINHVIEN (MSSV, HOLOT, TEN, NGOAINGU, TRINHDO)

Một sinh viên có thể biết nhiều ngoại ngữ (Anh, Pháp, Đức, ..) với các trình độ khác nhau (A,B,C,..), Vậy NGOAINGU và TRINHDO là các thuộc tính bộ liên quan với nhau. Để khử thuộc tính bội NGOAINGU và TRINHDO, ta thêm thực thể NGOAINGU với các thuộc tính TENN, TRINHDO và quan hệ **biết**



Dữ liệu phụ thuộc thời gian

Có những loại dữ liệu thay đổi theo thời gian, chẳng hạn như LUONG, NGAYQĐ (lên lương), GIACA (hàng hoá)..., những thuộc tính biểu diễn dữ liệu đó cũng là thuộc tính bội, và cũng thuộc dạng nhóm thuộc tính lặp, ta cũng có thể khử thuộc tính bội, và nhóm lặp như trường hợp trên.

Mô hình dữ liệu quan hệ (*Relational Data Model*)

Mô hình này được Codd E.F đề xuất vào năm 1970 tại công ty IBM ở San Jose, Mỹ. Hiện nay, mô hình này chiếm 80% các mô hình CSDL thương mại. Hầu như các phần mềm quản trị CSDL phổ biến ở nước ta như : dBASE, FoxBASE, FoxPRO, ACCESS, hay ORACLE cũng sử dụng mô hình quan hệ.

Mô hình này dựa trên cơ sở khái niệm lý thuyết tập hợp các quan hệ, tức là tập k_bộ (k cố định) tổ chức dưới dạng bảng.

Tính ưu điểm của mô hình này là nhờ tính chặt chẽ của toán học về thuyết tập hợp, nên mô hình này đã mô tả dữ liệu một cách rõ ràng uyển chuyển, thể hiện qua các tính chất sau :

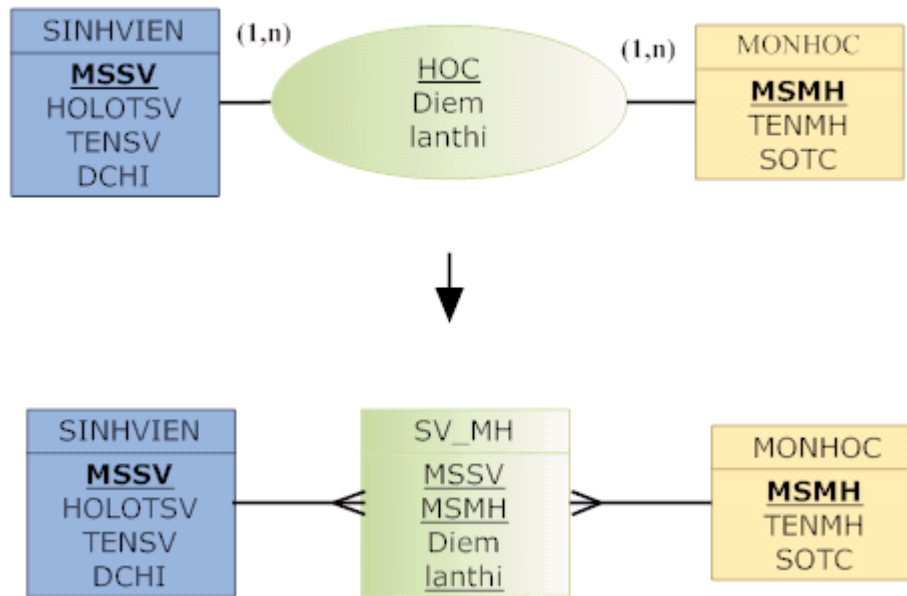
- . Thể hiện dữ liệu ở dạng bảng dữ liệu
- . Có các phép toán trên các quan hệ
- . Sử dụng phép toán vị từ trên các đối tượng.

Mô hình này sẽ được khảo sát cụ thể ở chương tiếp theo.

Qui tắc chuyển đổi từ mô hình thực thể liên hệ sang mô hình dữ liệu quan hệ

- * Mỗi thực thể hình thành một quan hệ tương ứng.
- * Nếu giữa 2 thực thể có mối quan hệ có bản số (1,n)-(1,n) thì hình thành thêm 1 quan hệ mới chứa ít nhất là khoá của các thực thể thành phần.
- * Nếu giữa 2 thực thể có mối liên hệ có bản số (1,1)-(1,n) thì không hình thành thêm quan hệ mới, mà quan hệ ứng thực thể phía (1,1) nhận thêm thuộc tính là khoá của thực thể phía (1,n).

Ví dụ 13: Chuyển đổi mối kết hợp (1,n)-(1,n)



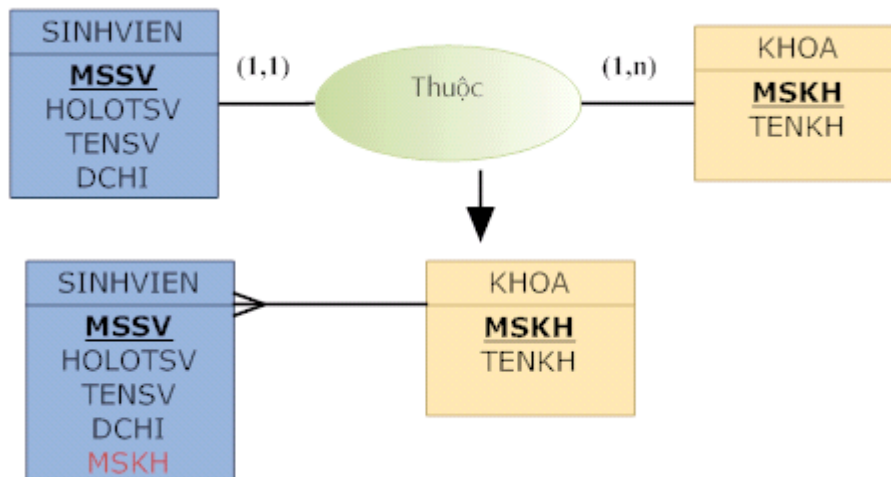
Mô hình dữ liệu quan hệ tương ứng:

SINHVIEN(MSSV, HOLOTSV, TENS, DCHI)

MONHOC(MSMH, TENMH, SOTC)

SV_MH(MSSV, MSMH, Diem, lan thi)

Ví dụ 14: Chuyển đổi mối kết hợp (1,1) – (1,n)



Mô hình dữ liệu quan hệ tương ứng:

KHOA(MSKH, TENKH)

SINHVIEN (MSSV, HOLOTSV, TENS, DCHI, **MSKH**)

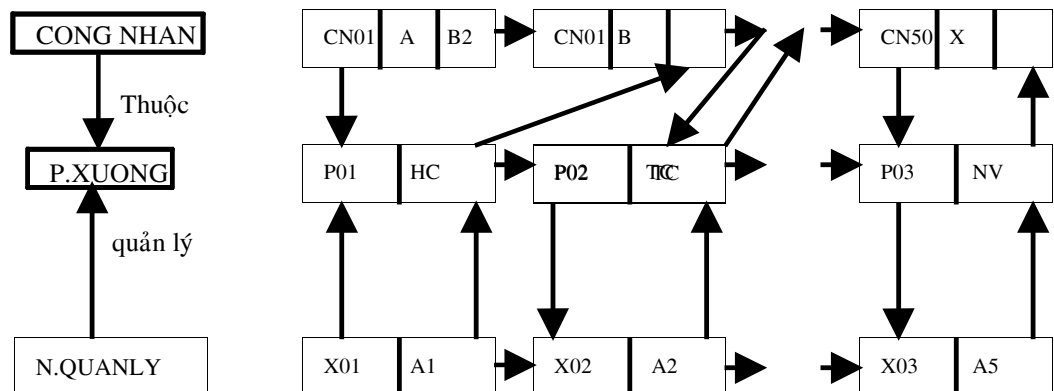
Mô hình dữ liệu mạng (*Network Data model*)

Mô hình CSDL mạng được trình bày bởi nhóm DBTG (Data Base Task Group) mã Codasyl vào năm 1971. Sau này vào năm 1978 nó được phát triển dựa vào các kết quả nghiên cứu. Nhiều hệ thống tin học dùng mô hình

mạng như CII, UNIVAX. Mô hình mạng, và mô hình quan hệ có thể chuyển đổi với nhau.

Một mô hình mạng chính là, mô hình thực thể_liên hệ, trong đó các mối liên hệ bị hạn chế là kiểu nhị phân và một-nhiều. Chính vì điều này cho phép chúng ta dùng mô hình đồ thị vô hướng biểu diễn dữ liệu. Ở mức biểu diễn quan niệm, mô hình mạng có cấu trúc đa danh sách.

Ví dụ 15: a. Mô hình mạng thể hiện mối quan hệ : Công nhân, Phân Xưởng và Người quản lý



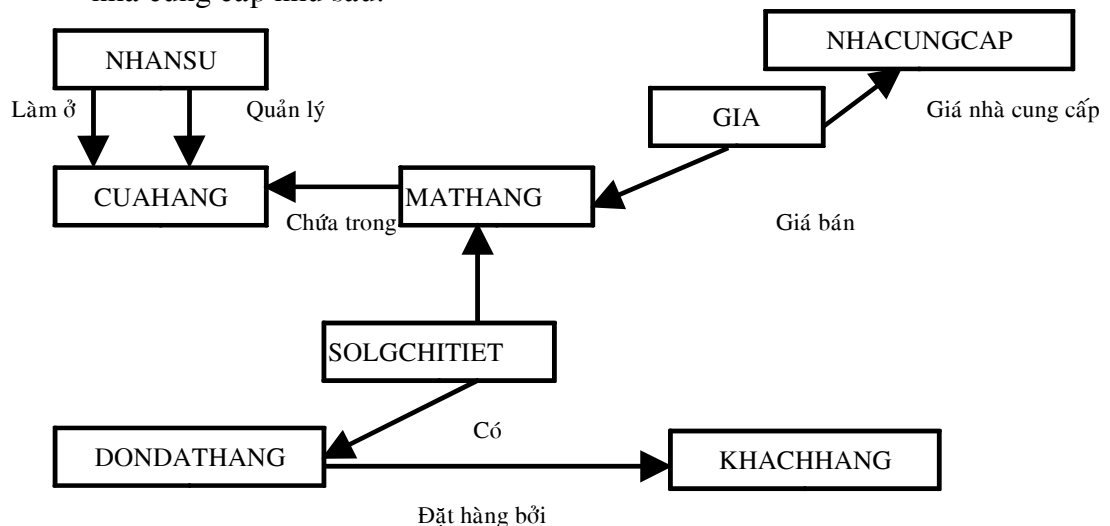
Hướng của vectơ biểu diễn mối quan hệ đi từ thực thể ứng với cặp bản số (1,1) đến thực thể ứng với bản số (1,n)

Các bản ghi tương ứng của các thực thể được tổ chức theo danh sách liên kết

Mô hình mạng có cấu trúc đa danh sách

Đồ thị biểu diễn là 1 đồ thị có hướng phức tạp.

b. Mô hình mạng cho bài toán quản lý một siêu thị về nhân sự, khách hàng, nhà cung cấp như sau:



Chuyển đổi từ 1 mô hình mô hình thực thể liên hệ thành mô hình mạng

* Chuyển các mối liên hệ có số ngôi (số thực thể tham gia vào mỗi liên hệ) > 2 thành các mối liên hệ có số ngôi là 2 (bằng cách thêm vào thực thể mới)

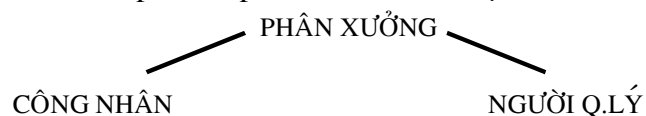
- * Chuyển các mối quan hệ 1-n, n-n bằng cách hình thành thêm quan hệ như chuyển đổi sang mô hình dữ liệu quan hệ.
- * Mỗi thực thể ứng với 1 đỉnh của đồ thị, các cung có hướng từ phía thực thể (1,1) đến phía (1,n).

Mô hình dữ liệu phân cấp (Hierarchy data model)

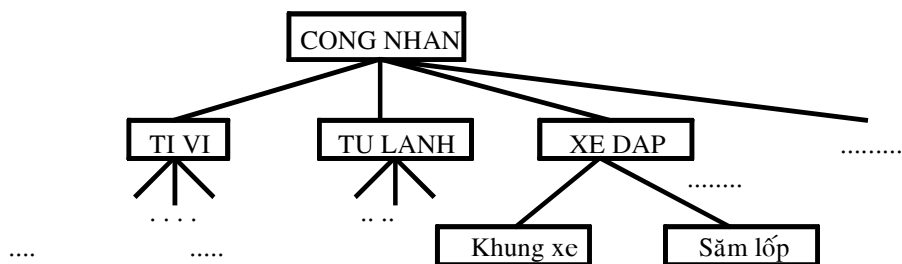
Một phân cấp (*Hierarchy*) : chính là một mạng có nhiều cây, nghĩa là tập các cây hay gọi là rừng (*forest*), trong đó tất cả các liên kết được biểu diễn bởi các nút đường nối từ nút ở ngọn vectơ đến các nút ở gốc vectơ.

Ví dụ 16 :

Một mô hình CSDL phân cấp thể hiện sơ đồ thực thể biểu diễn ở ví dụ 15.a



Mô hình CSDL phân cấp thể hiện sơ đồ thực thể biểu diễn mối quan hệ giữa người sản xuất hàng hóa và phụ kiện liên quan đến hàng hóa.



Việc tìm kiếm và xử lý trên mô hình dữ liệu phân cấp thường mất nhiều thời gian, và khó. Mô hình này được IBM sử dụng ngôn ngữ DL/1 vào năm 1975 để xử lý. Mô hình mạng và mô hình phân cấp thường được gắn liền với ngôn ngữ xử lý dữ liệu của hệ thống điều hành. Chính vì vậy 2 mô hình này còn được gọi là mô hình truy nhập dữ liệu.

Qui tắc chuyển đổi từ mô hình mạng sang mô hình phân cấp

Input: - Mô hình mạng, mỗi thực thể tương ứng với 1 số thứ tự

Output: - Các mô hình phân cấp tương ứng

Procedure BUILD(n);

Begin

Đánh dấu chọn cho nút n;

For mỗi đường nối từ một nút m nào đó đến n do

Begin

đặt m là con của n;

If m chưa đánh dấu chọn then BUILD(m);

```

End;
End;
/* Chương trình chính */
Begin
    Đánh dấu “huỷ chọn” cho tất cả các nút;
    While vẫn còn nút có dấu “huỷ chọn” do
        Begin
            Lấy ra 1 nút k chưa đánh dấu chọn;
            /* Nên chọn nút n có nhiều đường nối đến và không có đường nối
đi cá nút có dấu huỷ chọn */
            BUIILD(k)
        End.

```

Mô hình hướng đối tượng (Object Oriented model) :

Đây là một mô hình CSDL sử dụng phương pháp định hướng đối tượng. Phương pháp này được đề xuất vào những năm 1980 đặc trưng bởi 2 đặc tính :

- Tính Trừu tượng hóa
- Tính Kế thừa

Trừu tượng hóa dữ liệu là cách thức biểu diễn một nhóm các đối tượng có liên quan bằng một lớp có những đặc tính chung nhất của chúng và bỏ đi những tính chất cá biệt.

Kế thừa là quá trình mà trong đó các đối tượng của một lớp có thể yêu cầu sử dụng một số tính chất của những đối tượng ở những lớp khác. Khái niệm kế thừa hỗ trợ cho việc tạo ra các cấu trúc phân cấp các lớp.

Cấu trúc của 1 đối tượng gồm:

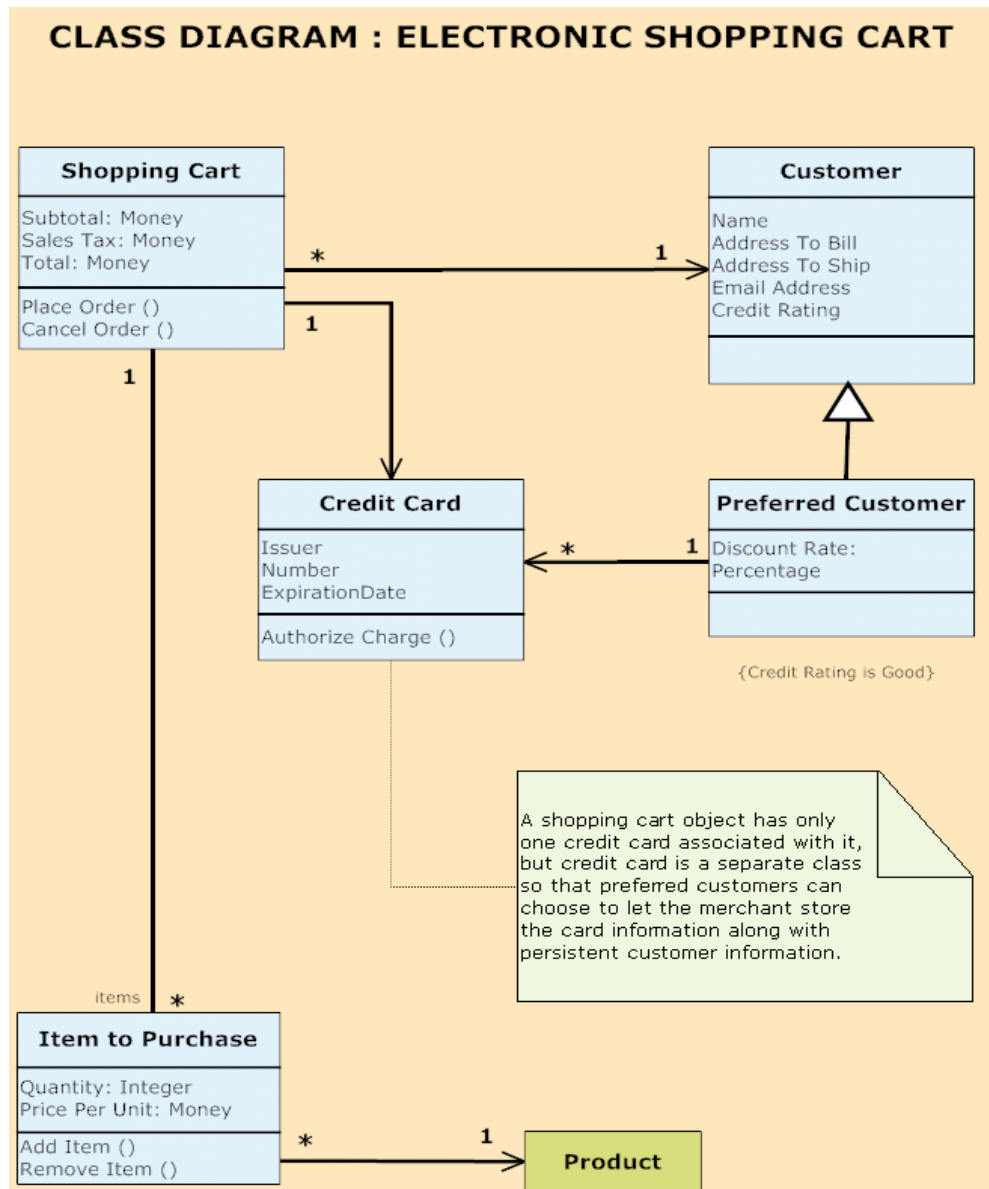
Property (Thuộc tính) : quy định các đặc trưng của nó.

Method (Phương thức): quy định các hoạt động của đối tượng.

Event (sự kiện) : Xử lý các sự kiện trên hệ thống tác động lên nó.

Mô hình hướng đối tượng thường được xử lý bởi các ngôn ngữ cấp cao, như C, C++. Đây là mô hình đang được quan tâm.

Ví dụ 17 : Một mô hình hướng đối tượng thể hiện sự trừu tượng hóa một hệ thống Cửa hàng điện tử



V. KIẾN TRÚC CƠ SỞ DỮ LIỆU:

V.1. Cơ sở dữ liệu tập trung: (Centralized database)

Đặc trưng của kiến trúc này là dữ liệu tập trung tại 1 thiết bị lưu trữ.

CSDL trên máy tính cá nhân (Personal computer database)

CSDL được lưu trữ trong thiết bị nhớ của 1 máy tính cá nhân, thường để giải quyết các bài toán nhỏ, phục vụ cho 1 người dùng.

- CSDL máy tính trung tâm (Central computer database)

Dữ liệu lưu ở máy tính trung tâm. Người dùng từ các địa điểm xa truy cập CSDL qua các terminal và đường truyền. Tùy theo quy mô CSDL mà máy trung tâm là các cấu hình lớn bé khác nhau.

CSDL mạng kiểu File server (LAN file server database): Trên mạng cục bộ có 1 hay nhiều máy chủ (File Server). File Server là thiết bị điều khiển thao

tác các File và được chia sẻ bởi các máy trạm nối trên mạng. Các máy trạm có quyền sử dụng CSDL khi chạy các chương trình ứng dụng. Như vậy, với 1 CSDL nhưng có nhiều bản copy cùng chạy 1 lúc, mỗi bản gán cho một máy trạm. Đặc trưng của cấu trúc này là toàn bộ việc thao tác dữ liệu được thực hiện ở máy trạm chứ không phải ở máy chủ. Thiết kế CSDL này có các hạn chế: lưu lượng trên đường truyền lớn làm giảm tốc độ xử lý, trong khi hệ thống không sử dụng hết khả năng của máy chủ, máy trạm phải có cấu hình phù hợp với trình ứng dụng. Chương trình ứng dụng phải đảm bảo ràng buộc toàn vẹn, là cho việc lập trình khó khăn hơn.

CSDL mạng kiểu khách/chủ (LAN Client/Server database): Đây là kiến trúc khắc phục hạn chế của kiểu File server, trong đó công việc được phân chia hợp lý giữa máy trạm(client) và máy chủ (database server). Máy trạm chịu trách nhiệm về giao diện người dùng, kể cả biểu diễn dữ liệu, còn máy chủ chịu trách nhiệm lưu trữ và xử lý dữ liệu, bao gồm tính bảo mật, toàn vẹn, an toàn dữ liệu. Hầu hết các hệ quản trị CSDL hiện nay đều hỗ trợ kiến trúc này.

V.2.Cơ sở dữ liệu phân tán: (*Distributed database*)

Trong kiến trúc này dữ liệu được lưu trữ vật lý tại nhiều điểm khác nhau. Có 2 loại kiến trúc phân tán.

Cơ sở dữ liệu đồng bộ (*Homogeneous database*)

CSDL ở các địa điểm khác nhau được xây dựng theo cùng 1 công nghệ, hoặc ít nhất chúng tương thích với nhau. Điều này làm cho việc chia sẻ dữ liệu được dễ dàng hơn.

Cơ sở dữ liệu dị bộ (*Heterogeneous database*)

CSDL ở các điểm không tương thích với nhau. Để chia sẻ dữ liệu cần phải có hệ thống chuyển đổi dữ liệu gọi là Communication server để làm tương thích dữ liệu giữa các CSDL khác nhau.

VI. CÁC NGÔN NGỮ CSDL :

Trong các ngôn ngữ lập trình thông thường, tất cả các khai báo và các câu lệnh khả thi đều là thành phần của ngôn ngữ. Trong thế giới CSDL thường có 2 chức năng riêng lẽ là khai báo và tính toán.

Ngôn ngữ định nghĩa dữ liệu (*Data Definition Language-DDL*) : là hệ thống ký hiệu để mô tả các kiểu thực thể và mối liên hệ giữa chúng theo một mô hình dữ liệu cụ thể nào đó.

Ví dụ 18 : Có thể định nghĩa một quan hệ mô tả các chuyến bay thực hiện bởi một hãng hàng không với định nghĩa dữ liệu như sau :

```
Create table Flights (Number : int, Date : char(6)),  
                      Seats : int, From : char(3), to : char(6)),  
Create index For Flights on Number
```

Đoạn mã này là một thí dụ về ngôn ngữ định nghĩa dữ liệu SQL. Ngôn ngữ định nghĩa dữ liệu được sử dụng khi thiết kế CSDL. Ngôn ngữ định nghĩa dữ liệu có những câu lệnh để mô tả cấu hình vật lý (physical layout) theo các thuật ngữ trừu tượng.

Ngôn ngữ thao tác dữ liệu : (*Data Manipulation Language- DML*)

Các thao tác trên CSDL đòi hỏi phải có một ngôn ngữ đặc biệt gọi là ngôn ngữ thao tác dữ liệu (Data Manipulation Language- DML) hay còn gọi là ngôn ngữ vấn tin(Query Language).

Các thao tác dữ liệu thường là : cập nhật (update), bổ sung (append), chèn (insertion), xóa (delete), sửa (modify), tìm kiếm (Find)

VII. HỆ QUẢN TRỊ CSDL, CHỨC NĂNG CỦA HỆ QUẢN TRỊ CSDL

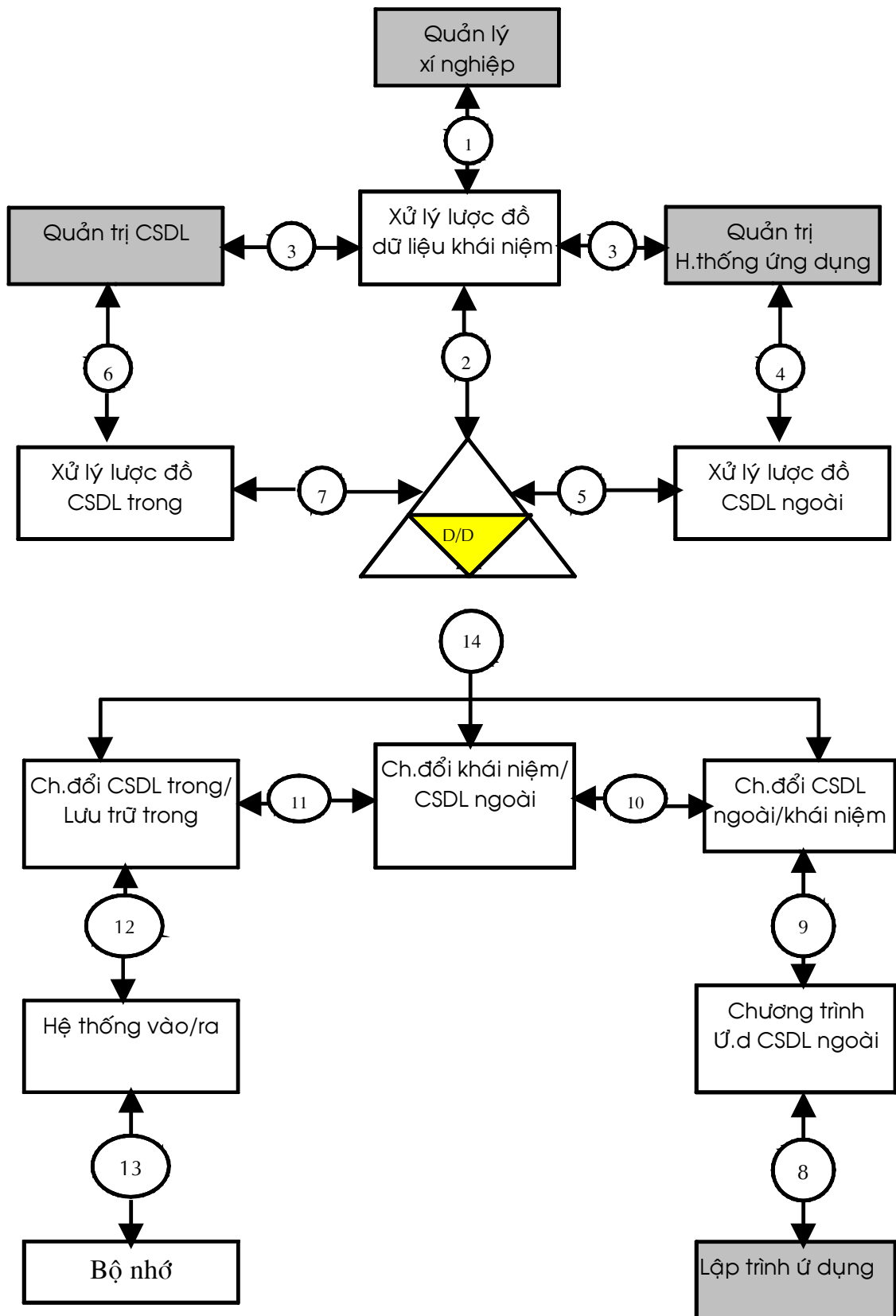
Hệ quản trị CSDL (Database Management system _DBMS) là một hệ thống phần mềm cho phép người sử dụng định nghĩa, xây dựng và bảo trì một CSDL và cho phép truy cập có điều khiển đến dữ liệu.

VII.1 Kiến trúc 3 mức ANSI/X3/SPARC

Ứng với kiến trúc 3 mức của 1 hệ CSDL nhóm ANSI/X3/SPARC đề nghị dùng 3 mức lược đồ trong kiến trúc hệ quản trị CSDL là mức ngoài, mức khái niệm và mức trong. Kiến trúc này cho phép mô tả luân chuyển dữ liệu trong CSDL. Mô hình đó được thể hiện như sau:

Trong kiến trúc trên, các hình chữ nhật không màu biểu diễn các chức năng xử lý, các hình chữ nhật màu sẫm biểu thị chức năng quản lý. Các mũi tên chỉ các dòng dữ liệu, dòng lệnh, dòng chương trình và dòng mô tả, và các số trên mũi tên biểu diễn các giao diện.

Hình tam giác biểu diễn từ điển/thư mục dữ liệu (data dictionary/directory – D/D) hay metadatabase (siêu CSDL) chứa các định nghĩa lược đồ và ánh xạ. Nó cũng có thể chứa các số liệu thống kê về việc sử dụng CSDL, các thông tin kiểm soát truy xuất,.. Từ điển/thư mục dữ liệu có vai trò trung tâm, vừa để xử lý các lược đồ khác nhau, vừa cung cấp các ánh xạ giữa chúng.



Kiến trúc 3 mức ANSI/X3/SPARC

Trong kiến trúc này 3 quản trị viên giữ các vai trò quan trọng.

Quản trị viên xí nghiệp (Enterprise administrator) là người chịu trách nhiệm chính trong việc xây dựng và sử dụng hệ thống thông tin trong đơn vị, có nhiệm vụ chuẩn bị nội dung để xây dựng lược đồ khái niệm.

Quản trị viên CSDL (Database Administrator-DBA) là người có quyền hạn và trách nhiệm cao nhất về tài nguyên thông tin của cơ quan, chịu trách nhiệm định nghĩa các lược đồ dữ liệu, có quyền thiết kế, hiệu chỉnh cấu trúc CSDL.

Quản trị viên ứng dụng (Application administrator) chịu trách nhiệm định nghĩa các lược đồ ngoài cho các ứng dụng. Họ không có quyền sửa đổi cấu trúc CSDL.

Ngoài 3 loại quản trị viên có chức năng quản lý, còn có các loại người dùng khác là lập trình viên ứng dụng (application programmer), lập trình viên hệ thống (System programmer) và người sử dụng đầu cuối (End user)

Các giao diện giữa các chức năng bao gồm :

1. Ngôn ngữ mô tả dữ liệu khái niệm cho phép quản trị viên xí nghiệp xác định lược đồ khái niệm.
2. Ngôn ngữ mô tả dữ liệu khái niệm, biên dịch lược đồ khái niệm và đưa nó vào từ điển dữ liệu.
3. Ngôn ngữ mô tả dữ liệu khái niệm biên dịch, cho phép quản trị viên ứng dụng và quản trị viên CSDL tham chiếu đến lược đồ khái niệm để xác định các phép xử lý tương ứng.
4. Ngôn ngữ mô tả dữ liệu ngoài, cho phép người dùng xác định lược đồ ngoài và các luật tương ứng giữa lược đồ ngoài và lược đồ khái niệm.
5. Ngôn ngữ mô tả dữ liệu ngoài, biên dịch lược đồ ngoài và đưa nó vào từ điển dữ liệu.
6. Ngôn ngữ mô tả dữ liệu trong, cho phép người dùng xác định lược đồ trong và các luật tương ứng giữa lược đồ trong và lược đồ khái niệm.
7. Ngôn ngữ mô tả dữ liệu trong, biên dịch lược đồ trong và đưa nó vào từ điển dữ liệu.
8. Ngôn ngữ xử lý dữ liệu, cho phép lập trình viên ứng dụng thao tác với dữ liệu thông qua lược đồ ngoài.
9. Ngôn ngữ xử lý dữ liệu ngoài, chuyển hoá dữ liệu ngoài sang dữ liệu khái niệm tương ứng.
10. Ngôn ngữ xử lý dữ liệu khái niệm, chuyển hoá dữ liệu khái niệm sang dữ liệu trong tương ứng.
11. Ngôn ngữ xử lý dữ liệu trong, chuyển hoá dữ liệu trong sang dữ liệu trong tương ứng.
12. Ngôn ngữ lưu trữ dữ liệu, ứng với giao diện của hệ thống lưu trữ dữ liệu.
13. Giao diện bộ nhớ phụ, cho phép thực hiện các tác vụ vào/ra trên thiết bị lưu trữ.

14. Giao diện truy nhập từ điển, cho phép chuyển hoá dữ liệu khái niệm và dữ liệu trong.

VII.2. Những khả năng của hệ quản trị CSDL :

Có hai đặc tính để phân biệt DBMS với các hệ thống lập trình khác, đó là :

- 1) Khả năng quản lý những dữ liệu cố định
- 2) Khả năng truy xuất có hiệu quả một số lượng lớn dữ liệu

Ngoài hai đặc tính cơ bản trên. Một số khả năng khác rất thường gặp trong các DBMS trên thị trường . Đó là :

- a- Hỗ trợ ít nhất một mô hình dữ liệu (*Data Model*) khi đó người sử dụng có thể xem được dữ liệu.
- b- Hỗ trợ một số ngôn ngữ bậc cao cho phép người sử dụng định nghĩa các cấu trúc dữ liệu, truy xuất dữ liệu và các thao tác dữ liệu.
- c- Quản lý của các giao tác (*transaction*) , nghĩa là cho phép nhiều người sử dụng truy xuất đồng thời và chính xác đến một CSDL.
- d- Điều khiển quá trình truy xuất, là khả năng giới hạn các quá trình truy xuất dữ liệu của những người không được phép và khả năng kiểm tra độ tin cậy của dữ liệu.
- e- Có đặc tính tự thích ứng, là khả năng phục hồi lại dữ liệu do sự cố của hệ thống mà không làm mất dữ liệu.

Chú ý : Hệ quản trị CSDL khác với hệ quản lý các file hay các file CSDL bởi vì nó cho phép mô tả dữ liệu theo cách không phụ thuộc vào người sử dụng, không phụ thuộc vào yêu cầu tìm kiếm và trao đổi thông tin. Hệ quản trị CSDL còn có các công cụ quản lý thông tin.

VII.3. Lịch sử các hệ quản trị CSDL :

Quá trình phát triển các hệ CSDL có thể xem bắt đầu từ năm 1968. Các mốc lịch sử của hệ quản trị CSDL :

- Những năm 1960 gắn liền với các sản phẩm đầu tiên của hệ quản trị file; xuất hiện bộ nhớ ngoài như bộ nhớ lý tưởng. Bộ nhớ này cho phép dùng chung, dễ dàng sử dụng, cho phép đánh địa chỉ trực tiếp và có dung lượng lớn, các hệ thống dùng ngôn ngữ lập trình để xử lý dữ liệu.
- Giữa những năm 60, thế hệ đầu của hệ CSDL đánh dấu bằng sự phân rã, mô tả những dữ liệu của chương trình ứng dụng và ngôn ngữ truy nhập bên trong. Bằng các lệnh phi thủ tục, người ta có thể truy nhập dữ liệu thay vì đi theo cấu trúc lưu trữ vật lý của các dữ liệu. Đại diện của các hệ thống này là CODASYL và IMS. Chúng dựa trên mô hình truy nhập, tức các mô hình sử dụng nhiều chức năng xử lý dữ liệu của hệ thống điều hành của máy tính có tính tới việc tối ưu phương pháp phân phối bộ nhớ phụ.
- Những năm 70, có thế hệ thứ hai của hệ quản trị CSDL với mô hình quan hệ. Đây là mô hình giúp đơn giản hóa việc truy nhập dữ liệu của người sử dụng bên ngoài. Nó sử dụng khái niệm lý thuyết tập hợp nên việc truy nhập dễ dàng, góp phần tối ưu việc khai thác dữ liệu.

Đầu năm 1980 bắt đầu có những hệ thống quản trị cụ thể loại này các hệ thống tiêu biểu như : ORACLE, INGRES, SYBASE, INFORMIX, DBZ và RDB. Các hệ thống này bước đầu có kiến trúc phân tán dựa trên môi trường khách/chủ (Client/server).

-Thế hệ 3 : Các hệ quản trị CSDL được phát triển vào những năm 80. Chúng dùng các mô hình dữ liệu phong phú và kiến trúc phân tán hơn so với các hệ thống trước. Kiến trúc này cho phép người dùng liên hệ với nhau tốt hơn. Thế hệ 3 có thể kể ra gồm:

- Mô hình hướng đối tượng
- Mô hình suy diễn dữ liệu
- Cơ sở dữ liệu phân tán

Một số hệ quản trị CSDL điển hình như GEMSTORE của Servio Corporation, OBJECT/DB của Objectivity, OBJECT STORE của Object Design, OPEN ODB của Hewlett-Packard, OBJECT/DB của DEC....

VII.4. Các chức năng của hệ CSDL.

Mô tả dữ liệu : Công việc mô tả dữ liệu tập trung vào các thuộc tính của tập các đối tượng được mô hình hóa trong CSDL, chứ không mô tả đối tượng cụ thể. Các đối tượng được mô tả cụ thể nhờ các chương trình ứng dụng. Việc mô tả dữ liệu độc lập với việc xử lý dữ liệu.

Tìm kiếm dữ liệu : Trên CSDL đã có, người sử dụng có thể tìm một đối tượng hoặc chỉ một phần dữ liệu. Công cụ tìm kiếm thường là ngôn ngữ hỏi CSDL. Điều kiện hỏi (*Query Qualification*) là các biểu thức logic diễn tả các yêu cầu mà đối tượng cần tìm phải thỏa mãn.

Cập nhật dữ liệu : Dữ liệu luôn phải có sự bổ sung, thay đổi nhằm phục vụ cho các chương trình ứng dụng. Thuật ngữ cập nhật được dùng theo nghĩa rộng, bao hàm các phép xử lý sau :

- Thêm dữ liệu
- Bớt hay xóa dữ liệu
- Sửa đổi hay cập nhật dữ liệu

VII.5. Chuyển hóa dữ liệu :

Các thể hiện của dữ liệu ở các mức lược đồ khác nhau là không như nhau. Khi người sử dụng chuyển quan tâm sang mức khác thì các dữ liệu cần phải chuyển sang dạng phù hợp khác.

VII.6 Điều khiển tính toàn vẹn của dữ liệu :

Tính toàn vẹn dữ liệu trong CSDL biểu hiện theo một trong ba nội dung sau:

- Điều kiện duy nhất của khóa dữ liệu : Mỗi bảng ghi dữ liệu hay mọi thực thể đều có các tên tương ứng duy nhất dùng làm khóa.
- Điều kiện tham chiếu : Trong CSDL có một số liên kết được mô tả và phải tôn trọng.

Ví dụ 19: Khi quản lý tuổi của sinh viên, thì trong CSDL phải có một thuộc tính liên quan đến ngày sinh.

- Điều kiện về miền giá trị : Các thuộc tính của các thực thể mang một số giá trị. Các giá trị này không thể tùy tiện, mà bị hạn chế.

Ví dụ 20: Điểm số một môn học của sinh viên không thể bé hơn 0 hoặc lớn hơn 10 (Nếu quản lý theo thang điểm 10).

VII.7. Quản lý các giao tác và an toàn dữ liệu :

Xử lý các dữ liệu dẫn đến việc quản lý các giao tác xử lý dữ liệu. Do có nhiều truy xuất dữ liệu đồng thời dẫn đến sự tương tranh. Việc đảm bảo sự tương hợp dữ liệu cho các truy xuất đồng thời này chính là thực hiện điều kiện toàn vẹn dữ liệu. Việc xử lý tốt quá trình tương tranh cũng đảm bảo không mất dữ liệu, không gây ùn tắc trong quá trình xử lý dữ liệu. Đó chính là sự đảm bảo an toàn dữ liệu.

VIII. NHỮNG KHÓ KHĂN KHI XÂY DỰNG CSDL:

Xây dựng 1 CSDL thường gặp 1 số nguy cơ tiềm ẩn và chi phí gia tăng so với các phương pháp truyền thống.

Chi phí nhân lực: Thông thường, các tổ chức quyết định xây dựng 1 CSDL hoặc mua hệ thống quản trị CSDL cần phải thuê hoặc đào tạo nhân lực để thiết kế CSDL, khai thác phần mềm CSDL mới, phát triển chương trình, quản trị hệ thống... Đôi khi chi phí này vượt quá lợi ích mà việc tổ chức và khai thác CSDL đem lại.

Chi phí an toàn dữ liệu: Sự giảm thiểu dư thừa dữ liệu kéo theo nguy cơ mất dữ liệu, nếu một hệ thống nào đó của hệ thống bị trục trặc. Để đảm bảo an toàn dữ liệu, cần trang bị hệ thống sao lưu để phục hồi dữ liệu hỏng.

Tương tranh dữ liệu: Sự truy cập đồng thời dữ liệu, chia sẻ bởi nhiều chương trình ứng dụng có thể dẫn đến 1 số bất cập: không đồng bộ dữ liệu, tranh chấp dữ liệu dẫn đến tắt nghẽn (khóa chết).

Mâu thuẫn về mặt tổ chức: CSDL dùng chung đòi hỏi sự quản trị và trách nhiệm về dữ liệu. Những mâu thuẫn trong việc thiết kế CSDL, mã hoá, phân quyền thường hay xảy ra và không dễ giải quyết.

Người quản trị CSDL cần lưu ý CSDL cũng như 1 phần mềm, chu kỳ sống của CSDL gồm các giai đoạn:

Lập kế hoạch CSDL (*database planning*)

Khảo sát, phân tích (*study and analysis*)

Thiết kế CSDL (*database design*)

Cài đặt CSDL (*database implementation*)

Bảo trì CSDL (*Post-implementation*)

Việc thiết kế CSDL có thể có 2 cách tiếp cận:

Thiết kế hướng quá trình:

P.tích yêu cầu → Thiết kế quá trình → Th.kế dữ liệu → cài đặt

Thiết kế hướng dữ liệu:

P.tích yêu cầu → Thiết kế CSDL → Th.kế quá trình → cài đặt

Bài tập

1. Hãy mô tả dữ liệu của một tổ chức mà bạn am tường bằng các mô hình CSDL đã khảo sát. (Ví dụ dữ liệu của một trường học hoặc một công ty kinh doanh)
2. CSDL Beer Drinkers (Bia và khách uống bia) chứa các thông tin về khách uống bia (drinkers), loại bia (Beers) và quán bia (Bars) nhằm cho biết các thông tin :

- i/ Loại bia ưa thích của mỗi khách
- ii/ Các khách uống bia thường xuyên của các quán
- iii/ Các loại bia có ở mỗi quán

Hãy biểu diễn lược đồ CSDL này trong các mô hình :

- a. Mô hình quan hệ
 - b. Mô hình mạng
 - c. Mô hình phân cấp
 - d. Mô hình đối tượng
3. Tìm hiểu ý nghĩa của mô hình dữ liệu logic (tài liệu [7])
 4. Tìm hiểu tổ chức dữ liệu vật lý (tài liệu [7])
 5. Tìm hiểu ý nghĩa của CSDL suy diễn (tài liệu [7],[5])
 6. * Giả sử chúng ta cần xây dựng 1 CSDL về các **sinh viên** (*Student*), các **khoá học** (*Course*) và **điểm** (*Grade*) của các sinh viên cho một khoá học. đối với mỗi sinh viên, chúng ta cần biết **tên** (*name*), **địa chỉ** (*Address*); với khoá học cần có **tên khoá học** và **tên Khoa** (*Departement*) tổ chức khoá học này. Chúng ta có thể biểu diễn CSDL này bằng các mô hình khác nhau và có thể có nhiều lựa chọn trong mỗi mô hình. Một số lược đồ có thể có những đặc tính không mong muốn, chẳng hạn:
 - A) Cho trước 1 sinh viên, chúng ta không thể xác định những khoá học đã được học bởi sinh viên này mà không phải duyệt qua rất nhiều phần của CSDL.
 - B) Không thể xác định được những sinh viên đã đăng ký một khoá học đã cho mà không phải duyệt qua rất nhiều phần của CSDL.
 - C) Không thể xác định được điểm của 1 sinh viên nhận được trong 1 khoá học đã cho.
 - D) Tạo ra dư thừa, chẳng hạn như phải lặp lại *Khoá học-Khoa* trong các khoá học; lặp lại *sinh viên-khoá học-điểm* hoặc *tên-địa chỉ* cho các sinh viên.

Dưới đây là các lược đồ được đề nghị. Đối với mỗi lược đồ, bạn hãy chỉ ra tập các hạn chế {A, B, C, D} trong chúng

- a) Lược đồ quan hệ

(COURSE, DEPT, STUDENT, ADDR, GRADE)

với các chỉ mục trên STUDENT và COURSE cho phép chúng ta tìm ra các bộ của một sinh viên đã cho, hoặc một khoá học đã cho mà không cần xét các bộ khác.

- b) Các lược đồ quan hệ

(COURSE, DEPT, GRADE) và (COURSE, STUDENT, ADDR)

với mỗi chỉ mục trên mỗi quan hệ.

- c) Lược đồ mạng với những kiểu mẫu tin logic COURSE(NAME, DEPT) cho biết tên khoá học và khoa; kiểu mẫu tin SAG(NAME, ADDR, GRADE) cho biết tên sinh viên, địa chỉ và điểm.

Mạng này có đường nối CSG từ SAG đến COURSE với ý nghĩa là một mẫu tin COURSE là chủ nhân của 1 tập mẫu tin SAG(s,a,g), là mẫu tin biểu thị cho mỗi sinh viên s có khoá học này, a là địa chỉ của sinh viên s, và g là điểm của sinh viên trong khoá học.

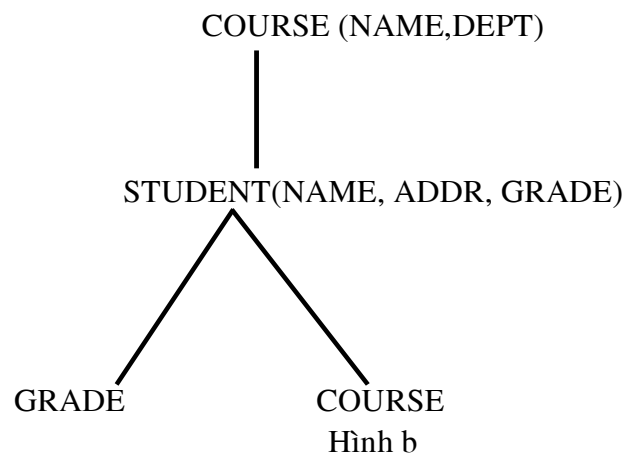
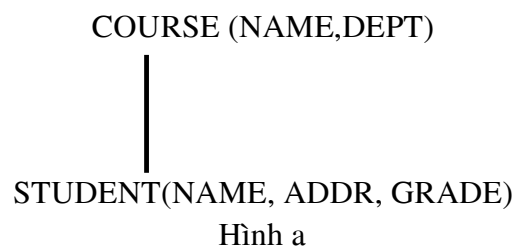
- d) Lược đồ phân cấp hình 6.1.a
 e) Lược đồ phân cấp hình 6.1.b
 f) Lược đồ mô hình hướng đối tượng, có một đối tượng thuộc kiểu SETOF (Ctype) biểu thị cho các khoá học, một đối tượng thuộc kiểu SETOF (Stype) biểu thị cho các sinh viên. Những kiểu này được định nghĩa như sau:

Ctype = RECORDOF(name:string; student:SETOF(Stype))

Stype = RECORDOF(name:string; transcript:Ttype)

Ttype = SETOF (RECORDOF (Course:Ctype, grade:string)).

Hình 6.1



Chương II

Mô hình dữ liệu quan hệ

I. TOÁN HỌC HÓA MÔ HÌNH QUAN HỆ :

Khái niệm toán học làm nền tảng cho mô hình quan hệ là các quan hệ theo lý thuyết tập hợp. Trong chương này ta xét đến các khái niệm cơ sở sau

I.1 Miền (*Domain*):

Miền là một tập các giá trị

Ví dụ 1 :

- Miền giá trị về tuổi của một sinh viên là $[17 \dots 30]$
- Miền giá trị về điểm số mã sinh viên thang điểm mười là:
 $\{x \in \mathbb{R}: 0 \leq x \leq 10\}$
- Tích Descartes của các miền $D_1, D_2, D_3, \dots, D_n$ được kí hiệu:
 $D_1 \times D_2 \times \dots \times D_n$ là tập tất cả các bộ (V_1, V_2, \dots, V_n) trong đó $V_i \in D_i$

Ví dụ 2: $D_1 = \{0, 1\}$, $D_2 = \{a, b, c\}$ thì

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

I.2 Quan hệ (*Relation*):

Quan hệ:

Cho tập hữu hạn các thuộc tính $U = \{A_1, A_2, \dots, A_n\}$ khác rỗng ($n \geq 1$). A_i có miền trị khác rỗng là $\text{dom}(A_i)$, $i=1, 2, \dots, n$. Đặt $D = \bigcup_{i=1}^n \text{dom}(A_i)$, một quan hệ r xác định trên U , ký hiệu $r(U)$, là một tập các ánh xạ $t: U \rightarrow D$ sao cho với mỗi $t(A_i) \in \text{dom}(A_i)$ (*Kí hiệu $t(A_i)$ là thành phần cột i của bộ t*). Mỗi ánh xạ được gọi là 1 bộ (*Tuples*) của quan hệ r .

Trong một số tài liệu người ta cũng định nghĩa quan hệ là một tập con của tích Descartes của một hay nhiều miền. Về mặt toán học, mỗi quan hệ có thể là vô hạn. Khi đề cập đến CSDL, chúng ta ngầm hiểu là chỉ khảo sát các quan hệ hữu hạn. Chú ý rằng, người ta không quan tâm đến thứ tự các thuộc tính trong 1 quan hệ.

Có thể xem **quan hệ** như một **bảng** (Table), trong đó mỗi **hàng** (Row) là một bộ, và mỗi **cột** (Column) tương ứng là một thành phần. Cột của một quan hệ gọi là **thuộc tính** (Attribute).

I.3 Lược đồ quan hệ (Relation Scheme):

Lược đồ quan hệ: Tập các thuộc tính của một quan hệ.

Nếu ký hiệu A_1, A_2, \dots, A_k là các thuộc tính của một quan hệ r thì lược đồ quan hệ tương ứng với r là $R = \{A_1, A_2, \dots, A_k\}$

Đôi khi để ngắn gọn người ta dùng $R = A_1A_2\dots A_k$ hoặc $R(A_1A_2\dots A_k)$

Ví dụ 3 Xét một quan hệ là tập các bản ghi thông tin công nhân

Họ và tên	Ph.Xưởng	Năm_sinh	Lương
Lê văn Tèo	Hàn	1962	810
Nguyễn Mỹ Thủy	Phay	1958	750
Trần Lê Văn	Phay	1968	710

Ta có lược đồ quan hệ CN như sau :

CN (Họ và tên, Ph.Xưởng, Năm_sinh, Lương)

Một cách hình thức ta đã định nghĩa quan hệ như sau :

Gọi $R = \{A_1, A_2, \dots, A_n\}$ là tập hữu hạn các thuộc tính, mỗi thuộc tính A_i , với $i=1, n$ có miền giá trị tương ứng là $\text{dom}(A_i)$. Một quan hệ trên tập thuộc tính $R = \{A_1, A_2, \dots, A_n\}$ là tập con của tích Descartes :

$$r \subseteq \text{dom}(A_1) \times \dots \times \text{dom}(A_n)$$

khi đó ký hiệu $r(R)$ hoặc $r(A_1\dots A_n)$.

I.4 Lược đồ cơ sở dữ liệu và Cơ sở dữ liệu:

Lược đồ cơ sở dữ liệu: Cho U là một tập các thuộc tính. Một lược đồ cơ sở dữ liệu D trên U là một họ các lược đồ quan hệ $\{R_1, R_2, \dots, R_p\}$ sao cho :

$$\bigcup_{i=1}^p R_i = U; \text{ và } R_i \neq R_j \text{ với } i \neq j$$

Một **cơ sở dữ liệu** d trên lược đồ CSDL $D = \{R_1, R_2, \dots, R_p\}$, là một họ các quan hệ $d = \{r_1, r_2, \dots, r_p\}$, sao cho r_i là một quan hệ trên R_i .

II. KHÓA (Keys)

II.1 Khoá của một quan hệ:

Khoá của một quan hệ r trên lược đồ quan hệ R là tập con $K=\{B_1, B_2, \dots, B_m\}$ của R thỏa mãn những tính sau :

1) $\forall t_1, t_2 \in r, \exists B \in K$ sao cho $t_1(B) \neq t_2(B)$ ⁴

2) $\forall K' \subsetneq K, K'$ không có tính chất 1. ($K' \neq \emptyset$)

Nếu K chỉ thỏa tính chất 1, thì K được gọi là siêu khoá (Super key)

Ví dụ 4: Xét quan hệ FLIGHTS (Lịch bay) như sau :

NUMBER	FROM	TO	DEPARTS	ARRIVES
83	JFK	TOKYO	11:30a	1:43p
84	LONDON	JFK	3:00p	5:55p
109	JFK	LOS ANGELES	9:50p	2:52p
213	JFK	BOSTON	11:43a	12:45p

Chúng ta dễ thấy $\{NUMBER\}, \{FROM, TO\}$ là hai khóa.

Nhận xét :

- Một quan hệ có thể có nhiều khóa.
- Để chỉ khóa của một quan hệ, người ta thường gạch dưới các thuộc tính hình thành khóa. Ví dụ : $R(\underline{A} \underline{B} C D E)$

Ví dụ 5 : với quan hệ FLIGHTS của ví dụ 4

$\{NUMBER\}, \{NUMBER, FROM\}$ là các siêu khóa

II.2 Khoá của một lược đồ quan hệ:

Nếu $K \in R$, K là khóa của $\forall r(R)$ thì K được gọi là một khóa của lược đồ quan hệ R . ⁵

II.3 Khoá ngoại (Foreign key)

Cho 1 lược đồ CSDL D trên $U = \bigcup_{i=1}^n R_i$, $R_i \in U$ và $K \in R_i$, K được gọi là khoá ngoại của R_i nếu và chỉ nếu tồn tại 1 $R_j \subseteq U, i \neq j$ sao cho K là khóa của R_j .

Ví dụ 6:

⁴ (1) Không có hai bộ nào có cùng giá trị trên mọi thuộc tính thuộc K

(2) K là tập thuộc tính nhỏ nhất phân biệt được hai bản ghi

⁵ Chú ý: Khóa của 1 quan hệ $r(R)$ chưa chắc là một khóa của lược đồ quan hệ R

Mã_PX	Tên_PX
H01	Hàn 01
TC	Tổ chức
HC	Hành chính

Mã_NV	Tên_NV	Ch_Môn	Mã_PX
A123	Trần	Điện	H01
B101	Anh	Ch.viên	HC
C112	Minh	Ch.viên	TC

Mã_PX là khoá ngoại của lược đồ quan hệ NHANVIEN

III. RÀNG BUỘC TOÀN VỆN

III.1 Định nghĩa 1:

Một ràng buộc toàn vẹn (RBTV) trong một CSDL là một quy luật bất biến mà tất cả các quan hệ trong CSDL ấy phải tuân theo.

Một CSDL có thể có nhiều RBTV, mỗi ràng buộc toàn vẹn liên quan đến một số quan hệ của CSDL. Tập RBTV này do người thiết kế CSDL đặt ra khi thiết kế hệ thống hay do quy định của hệ quản trị CSDL.

III.2 Định nghĩa 2:

1. Các quan hệ mà trên đó ràng buộc toàn vẹn được định nghĩa gọi là **bối cảnh** của ràng buộc toàn vẹn.
2. Tất cả các thao tác Thêm, Xóa, Sửa trên 1 quan hệ bối cảnh của 1 ràng buộc toàn vẹn mà có nguy cơ vi phạm ràng buộc toàn vẹn được gọi là **tầm ảnh hưởng** của RBTV.

III.3 Ràng buộc toàn vẹn trên 1 quan hệ

1. Ràng buộc miền giá trị

Cho $r(R)$ là 1 quan hệ của CSDL \mathcal{D}

- a. Ràng buộc khoá: K là khoá của R thì: $\forall t \in r$ ta có $t.K$ là duy nhất.
- b. Ràng buộc giá trị: $A \in R$ thì: $t \in r$ ta có $t.A \in \text{Dom}(A)$.

Nghĩa là giá trị của mỗi bộ tại một thuộc tính phải thuộc một miền nào đó được xác định trước.

2. **Ràng buộc liên thuộc tính:** Trong $r(R)$ có thể tồn tại 2 thuộc tính A, B mà giá trị mỗi bộ tại 2 thuộc tính có liên quan nhau theo một quy tắc nào đó.

Ví dụ 7: Cho lược đồ NHANVIEN(MSNV, Hoten, Nsinh, Ngay_vaoCQ)

Nhân viên nào vào cơ quan phải ít nhất 18 tuổi nên ta phải có:

$\forall t \in \text{NHANVIEN}: t.\text{YEAR}(\text{Nsinh}) > t.\text{YEAR}(\text{Ngay_vaoCQ}) - 19$

3. Ràng buộc toàn vẹn trên nhiều lược đồ quan hệ:

Ràng buộc khoá ngoại: Một lược đồ R có K1 là khoá ngoại, nghĩa là K1 là một khoá của lược đồ Ri nào đó. Khi đó K thoả:

$t \in r(R), \exists t1 \in r1(R1) : t.K1 = t1.K1$

IV. Các phép toán quan hệ : (*Relational Operators*)

IV.1 Các phép toán logic (*Boolean Operations*)

Cho hai quan hệ trên cùng một lược đồ $r(R)$, $s(R)$ có thể định nghĩa :

- **Phép hợp (Union)** : của $r(R)$ và $s(R)$, kí hiệu $r \cup s$
- **Phép giao (Intersection)** : của $r(R)$ và $s(R)$, kí hiệu $r \cap s$
- **Phép hiệu (Set difference)** : của $r(R)$ và $s(R)$ kí hiệu $r - s$

Xác định :

$$r \cup s = \{t : t \in r \vee t \in s\}$$

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

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

- Đặt $\text{dom}(R)$ là tập tất cả các bộ xác định trên các thuộc tính của R

- Phần bù (Complement) của quan hệ $r(R)$ là $\bar{r} = \text{dom}(R) - r$

Ví dụ 7:

A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d2
a2	b2	c2	d2

r

A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d2
a3	b2	c2	d2
a4	b5	c6	d2
a3	b3	c3	d3

s

$r \cup s$

A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d2
a2	b2	c2	d2
a3	b2	c2	d2
a4	b5	c6	d2
a3	b3	c3	d3

$r \cap s$

A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d2

$r - s$

IV.2 Phép chọn (*The Select Operator*)

Cho 1 quan hệ $r(R)$, phép chọn trên quan hệ r đã cho, thỏa mãn một biểu thức F kí hiệu $\delta_F(r)$ xác định như sau:

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

F là một công thức gồm có:

- i/ Các toán hạng, hằng, hoặc số hiệu các thành phần (thành phần i được ký hiệu là \$i\$)
- ii/ Các phép so sánh số học $<$, $=$, $>$, \leq , \geq , \neq , và
- iii/ Các toán tử logic \wedge (and), \vee (or), \neg (not)

Ví dụ 8: Xét quan hệ FLIGHTS ở ví dụ 4

$\delta_{\text{FROM} = \text{'JFK'}}(\text{FLIGHTS})$

NUMBER	FROM	TO	DEPARTS	ARRIVES
83	JFK	TOKYO	11:30a	1:43p
109	JFK	LOS ANGELES	9:50p	2:52p
213	JFK	BOSTON	11:43a	12:45p

Chú ý: Ta có thể chứng minh được các kết quả :

$$1) \delta_{A=a}(\delta_{B=b}(r)) = \delta_{B=b}(\delta_{A=a}(r))$$

$$2) \delta_{A=a}(r \gamma s) = \delta_{A=a}(r) \gamma \delta_{A=a}(s)$$

Ở đây : $\gamma = \cap, \cup$ hay $-$, r, s là các quan hệ trên cùng một lược đồ.

IV.3 Phép chiếu (Projection)

Giả sử $r(R)$, với $R = A_1 A_2 \dots A_n$, $X \subset R$, phép chiếu⁶ của r trên X ký hiệu $\Pi_X(r)$ được xác định :

$$\Pi_X(r) = r'(X) = \{t(X) \mid t \in r\}$$

Ví dụ 9: $R = ABCD$, $X = ACD$, $Y = ABC$

A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d1
a2	b3	c2	d2

$r(R)$

A	C	D
a1	c1	d1
a2	c2	d2

$\Pi_X(r)$

A	B	C
a1	b1	c1
a1	b2	c1
a2	b3	c2

$\Pi_Y(r)$

Chú ý: Ta có thể chứng minh các kết quả $\Pi_X(r)$

1) Nếu $R = A_1 A_2 \dots A_k$, $r(R)$, $X_1 \subseteq X_2 \subseteq \dots \subseteq X_m \subset R$ thì

$$\Pi_{X_1}(r) (\Pi_{X_2}(r) (\dots (\Pi_{X_m}(r) (r)) \dots)) = \Pi_{X_1}(r)$$

2) Nếu $A \in X$, $X \subseteq R$, $r(R)$ thì $\Pi_X(\delta_{A=a}(r)) = \delta_{A=a}(\Pi_X(r))$

IV.4 Tích Descaster : (Cartesian Product)

Xét $r(R)$, $s(K)$. Tích Descaster của r và s ký hiệu $r \times s$ được xác định :

⁶ Phép chiếu trên 1 quan hệ thực chất là loại bỏ đi một số thuộc tính và giữ lại những thuộc tính còn lại của quan hệ đó.

$$r \times s = \{t / t \text{ có dạng } (a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_k)\}$$

Ví dụ 10:

	A	B	D		A	C	E
	a1	b1	d1		a1	c1	e1
	a2	b2	d2		a1	c2	e1
					a2	c3	e2
	r (ABD)				s (ACE)		
rxs	A	B	D	A	C	E	
	a1	b1	d1	a1	c1	e1	
	a1	b1	d1	a1	c2	e1	
	a1	b1	d1	a2	c3	e2	
	a2	b2	d2	a1	c1	e1	
	a2	b2	d2	a1	c2	e1	
	a2	b2	d2	a2	c3	e2	

IV.5 Phép dán bộ

Xét $r(R)$, $s(S)$, $u \in r$, $v \in s$ ta ký hiệu $u*v$ là phép dán bộ. $U*v$ cho ta bộ t trên tập thuộc tính RS thỏa điều kiện: $t.R = u$ và $t.V = v$.

Với mỗi bộ u trong quan hệ $r(R)$ và với mỗi quan hệ $s(S)$ ta ký hiệu $u*S$ là phép dán bộ u với quan hệ S . $u*S$ cho ta quan hệ

$$l(RS) = \{u*v: v \in S\}$$

IV.6 Phép nối tự nhiên (Natural Join)

Phép kết nối tự nhiên của 2 quan hệ $r(R)$, $s(S)$, ký hiệu $r*s$, cho ta quan hệ chứa các bộ được dán từ các bộ u của quan hệ r với mỗi bộ v của quan hệ s sao cho các trị trên miền thuộc tính chung (nếu có của 2 bộ này giống nhau).

Nếu hai quan hệ $r(R)$, $s(S)$ có chung một số thuộc tính. Kết nối tự nhiên của r và s ký hiệu : $r \bowtie s$ được xác định như sau :

Nếu $r(A_1 A_2 \dots A_k)$, $s(B_1 B_2 \dots B_m)$

Đặt :

$$\{c_1, c_2, \dots, c_e\} = \{A_1, A_2, \dots, A_k\} \cap \{B_1, B_2, \dots, B_m\}, R \cup S = \{i_1, i_2, \dots, i_t\}$$

$$r \bowtie s = \Pi_{i_1, i_2, \dots, i_t} \delta_{r c_1 = s c_1, r c_2 = s c_2, \dots, r c_e = s c_e}(r \times s)$$

Ví dụ 11:

	A	B	D		A	C	E
	a1	b1	d1		a1	c1	e1
	a2	b2	d2		a1	c2	e1
					a2	c3	e2
	r(ABD)				s(ACE)		
	A	B	D	C	E	$r \bowtie s$	
	a1	b1	d1	c1	e1		
	a1	b1	d1	c2	e1		
	a2	b2	d2	c3	e2		

Chú ý : Nếu $\{A_1, A_2, \dots, A_k\} \cap \{B_1, B_2, \dots, B_m\} = \emptyset \Rightarrow r \bowtie s = r \times s$

IV.7 Phép nối nửa (Semi Join)

Phép nối nửa của $r(R)$ và $s(S)$, ký hiệu là $r \bowtie s$ được xác định

$$r \bowtie s = \Pi_R(r \bowtie s)$$

Chú ý : $r \bowtie s \neq s \bowtie r$

Ví dụ 12 :

Với r, s xác định ở ví dụ 11 ta có $r \bowtie s$

A	B	D
a1	b1	d1
a2	b2	d2

IV.8 Phép chia (Division)

Xét 2 quan hệ $r(R)$ và $s(S)$, với $S \subseteq R$. đặt $R' = R - S$ phép chia r cho s

Kí hiệu : $r \div s$ là quan hệ

$$r'(R') = \{t(R') : \forall ts \in s, \exists tr \in r \text{ mà: } tr(R') = t \text{ và } tr(s) = ts\}$$

Ví dụ 13 :

$r(R)$			$s(S)$	
A	B	D	B	D
a1	b1	d1	b1	d1
a1	b2	d2	b2	d2
a1	b2	d3	b2	d3
a1	b2	d4		
a2	b2	d2		
a3	b1	d1		
a3	b2	d2		
a3	b2	d3		
a4	b1	d1		

$r \div s$

A
a1
a3



Bài tập

1. Các quan hệ hằng (*Constant Relations*)

- Có thể hợp hai khóa thành 1 khóa ?
- Giao 2 siêu khóa có phải là 1 khóa ?

2*. Cho lược đồ quan hệ $R (A_1 A_2 \dots A_n)$

- Có thể có tối đa bao nhiêu khóa của R ?
- Có thể có tối đa bao nhiêu siêu khóa trên R ?

3. Cho

$r (A \ B)$

a b

c b

d e

$s (B \ C)$

b c

c a

b d

Hãy tính :

a/ $r \cup s$

b/ $r - s$ (bỏ qua tên thuộc tính c tính trong kết quả hợp và hiệu)

c/ $r \bowtie s$

d/ $\Pi_A (r)$

e/ $\delta_{A=c} (r \times s)$

f/ $s \bowtie r$

g/ $s \div \{b, c\}$

h/ $r \bowtie s$ (dùng $<$ theo thứ tự chữ cái)

$B < C$

4. Cho các quan hệ $r(ABC)$ và $s(BCD)$, $a \in \text{dom}(B)$. Những biểu thức nào sau đây là hợp lệ ?

a/ $r \cup s$

b/ $\Pi_B (r) - \Pi_B (s)$

c/ $\delta_{B=b} (r)$

d/ $\delta_{A=a, B=b} (s)$

e/ $r \bowtie s$

f/ $\Pi_A (r) \bowtie \Pi_P (s)$

5. Cho $r (R)$ và $s (S)$ là các quan hệ với $A \in R$. Chứng minh

$$\delta_{A=a} (r \bowtie s) = \delta_{A=a} (r) \bowtie s$$

6**. Cho quan hệ $q(RS)$. Tìm điều kiện đủ để :

$$q = \Pi_R (q) \bowtie \Pi_S (q)$$

Điều kiện cần ?

7** . Cho các quan hệ $r(R)$ và $s(S)$, với $R \supseteq S$ và đặt $R' = R - S$. Chứng minh đẳng thức :

$$a/ r \div s = \Pi_{R'}(r) - \Pi_{R'}((\Pi_{R'}(r) \bowtie s) - r)$$

$$b/ r \div s = \bigcap_{t \in s} \Pi_{R'}(\delta_{S=t}(r))$$

8. Cho X là một tập con của R , $A \in R$, $A \notin X$ và r là 1 quan hệ trên R . Tìm để :

$$\Pi_X(\delta_{A=a}(r)) = \delta_{A=a}(\Pi_X(r))$$

9. Cho $r(R)$ là quan hệ, $A \in R$ và $a, a' \in \text{dom}(A)$. Chứng minh :

$$[\delta_{A=a, A=a'}(r) = \emptyset \text{ hay } \delta_{A=a, A=a'}(r) = \delta_{A=a}(r)]$$

$$\text{Ký hiệu : } \delta_{A=a, A=a'}(r) = \delta_{A=a}(\delta_{A=a'}(r))$$

10. Cho quan hệ $r(R)$, $s(S)$ và $q = r \bowtie s$. Định nghĩa $r' = \Pi_R(q)$ và $s' = \Pi_S(q)$.

$$\text{Chứng minh : } q = r' \bowtie s'$$

11. Cho $r(R)$ và $s(S)$ là các quan hệ, ở đây $R \cap S = \emptyset$.

$$\text{Chứng minh : } (r \bowtie s) \div s = r$$

12. Cho r là 1 quan hệ trên lược đồ R và đặt s và s' là các quan hệ trên lược đồ S , ở đây $R \supseteq S$. Chứng minh rằng : nếu $s \subseteq s'$ thì $r \div s \supseteq r \div s'$

Chứng minh nếu điều kiện ngược là sai.

13. Cho CSDL :

$S(S\#, SNAME, STATUS, CITY)$

$P(P\#, PNAME, COLOR, WEIGHT, CITY)$

$SP(S\#, P\#, QTY)$

trong đó:

S : thể hiện thực thể hãng cung cấp

P : thể hiện thực thể mặt hàng

SP : thể hiện thực thể trung gian của S, P

Hãy biểu diễn các câu hỏi sau bằng các biểu thức đại số quan hệ:

- Liệt kê tất cả các mặt hàng màu đỏ
- Liệt kê $S\#$ tất cả các hãng cung ứng có cung cấp ít nhất 1 mặt hàng màu đỏ
- Liệt kê $S\#$ tất cả các hãng cung ứng tham gia cung cấp tất cả mặt hàng màu đỏ.
- Liệt kê $S\#$ tất cả các hãng cung ứng mặt hàng có tên P1
- Liệt kê $S\#$ tất cả các hãng cung ứng mặt hàng P1 và P2.
- Liệt kê $S\#$ tất cả các hãng không cung ứng mặt hàng nào.
- Liệt kê $S\#$ tất cả các hãng cung ứng tất cả các mặt hàng.

Làm lại các câu trên với yêu cầu hiển thị tên các nhà cung ứng.



Chương III

Lý thuyết thiết kế CSDL quan hệ

Khi thiết kế một CSDL, yếu tố quan trọng nhất là việc chọn lược đồ quan hệ. Trong mô hình quan hệ, lược đồ quan hệ được xây dựng hoàn chỉnh dựa vào cơ sở toán học lý thuyết tập hợp, quan hệ, chặt chẽ nên tạo được một lược đồ CSDL phù hợp cho đa số các ứng dụng. Trọng tâm của việc thiết kế các lược đồ CSDL là ý tưởng về phụ thuộc dữ liệu (*Data Dependency*) tức là các mức ràng buộc có thể giữa các giá trị hiện hữu của các lược đồ. Ví dụ thuộc tính này xác định duy nhất thuộc tính kia như SBD (số báo danh) xác định duy nhất (Tên, ngày sinh, địa phương của một học sinh).

Các vấn đề cần quan tâm khi thiết kế một CSDL đó là :

- Dư thừa dữ liệu (*Redundancy*)
- Không nhất quán (*Inconsistency*)
- Dị thường khi thêm bộ (*Insertion anomalies*)
- Dị thường khi xóa bộ (*Deletion anomalies*)

Ví dụ 1 : Xét một lược đồ quan hệ

BANHANG(Ngày, MSMH, Tên hàng, Đơn giá, Số lượng).

Ngày	MSMH	Tên hàng	Đơn giá	Số lượng
201107	TV01	Ti vi sony	2900	20
211107	TL02	Sanyo	1500	25
251107	TV01	Ti vi sony	2900	30
261107	RD01	Radio1	10	20
261107	VB01	Lò Viba	100	25
261107	TV01	Ti vi sony	2900	30
281107	TL01	Tủ lạnh LG	1500	15

- a) Dư thừa dữ liệu : Tên hàng được lập lại khi mỗi mặt hàng cùng Mã hàng được cung cấp.
- b) Không nhất quán : Khi cập nhật giả sử có sự thay đổi tên hàng tương ứng với một Mã hàng ta thay Mã hàng không có tên duy nhất.

c) Dị thường khi thêm bộ: khi có một mặt hàng mới, nhưng chưa biết Đơn giá, số lượng. Ta có thể đặt các giá trị rỗng ở những thành phần này, nhưng khi ta có số liệu, có thể những giá trị rỗng này không được loại bỏ.

d) Dị thường khi xóa: Khi xóa đơn giá cùng một mặt hàng có thể làm sai lệch doanh thu, vì đơn giá của cùng một mặt hàng ở các ngày khác nhau là khác nhau.

Để giải quyết điều đó cần xác định sự liên quan các thuộc tính, phân rã thành những quan hệ nhỏ hơn.

Khái niệm phụ thuộc hàm (*Functional Dependencies*) được đưa ra nhằm xác định mối quan hệ giữa các thuộc tính.

I. PHỤ THUỘC HÀM (*Functional Dependencies*) :

I.1 Định nghĩa : Phụ thuộc hàm

Cho r là một quan hệ trên lược đồ R , X và Y là các tập con của R .

Nói rằng X xác định hàm Y hoặc Y phụ thuộc hàm ký hiệu $X \rightarrow Y$.

Một quan hệ $r(R)$ thỏa phụ thuộc hàm $X \rightarrow Y$ nếu :

$$\forall t_1, t_2 \in r(R) : t_1(X) = t_2(X) \Rightarrow t_1(Y) = t_2(Y)$$

Chú ý : - Phụ thuộc hàm : $X \rightarrow \emptyset$ thỏa mãn với bất kỳ $r(R)$

Phụ thuộc hàm $\emptyset \rightarrow Y$ thỏa mãn với các $r(R)$, mà giá trị của thuộc Y là duy nhất.

I.2 Định nghĩa: Bao đóng tập phụ thuộc hàm

Cho F là tập các phụ thuộc hàm xác định trên lược đồ quan hệ R , bao đóng của tập phụ thuộc hàm F ký hiệu F^+ là tập phụ thuộc hàm bé nhất chứa F sao cho thỏa hệ tiên đề Armstrong:

a. Tính phản xạ: $Y \subseteq X \Rightarrow X \rightarrow Y \in F^+$

b. Tính tăng trưởng: $\forall X, Y, Z \subseteq R: X \rightarrow Y \Rightarrow XZ \rightarrow YZ \in F^+$

c. Tính bắc cầu: $\forall X, Y, Z \subseteq R: X \rightarrow Y, Y \rightarrow Z \Rightarrow X \rightarrow Z$

$$F^+ = \{ X \rightarrow Y \mid F \models X \rightarrow Y \}$$

Kí hiệu $F \models X \rightarrow Y$ nói rằng F suy dẫn logic $X \rightarrow Y$ hay $X \rightarrow Y$ được suy dẫn bởi hệ tiên đề Armstrong từ F .

Dễ thấy : $F \subseteq F^+$

I.3 Định nghĩa : Bao đóng tập thuộc tính

Cho tập thuộc tính R , F là tập phụ thuộc hàm xác định trên R . $X \subseteq R$ Bao đóng của tập thuộc tính X theo F ký hiệu :

$$X^+ = \{ A \in R \mid F \models X \rightarrow A \}$$

Dễ thấy : $X \subseteq X^+$

I.4 Bổ đề 1:

$X \rightarrow Y$ được suy ra từ tập phụ thuộc F đã cho bằng cách sử dụng hệ tiên đề Armstrong $\leftrightarrow Y \subseteq X^+$.

Chứng minh: (\rightarrow) Ta có $X \rightarrow Y \in F^+$, giả sử $Y = A_1 A_2 A_3 \dots A_n$,
vì $A_i \in Y \Rightarrow Y \rightarrow A_i \in F^+$ (tính phản xạ) $\Rightarrow X \rightarrow A_i \in F^+$ (tính bắc cầu),

$$\forall i=1..n \Rightarrow \bigcup_{i=1}^n A_i \subseteq X^+ \Rightarrow Y \subseteq X^+$$

(\leftarrow) Xét $Y \subseteq X^+$, nếu $Y = A_1 A_2 \dots A_n \Rightarrow A_i \in X^+$
 $\Rightarrow X \rightarrow A_i \in F^+, \forall i=1..n$

$$\Rightarrow X \rightarrow \bigcup_{i=1}^n A_i \in F^+$$

$$\text{Do: } X \rightarrow A_i \Rightarrow XX \rightarrow XA_i \leftrightarrow X \rightarrow XA_i$$

$$X \rightarrow A_{i+1} \Rightarrow XA_i \rightarrow A_i A_{i+1} \Rightarrow X \rightarrow A_i A_{i+1} \in F$$

$$\text{Tương tự ta có: } X \rightarrow A_1 A_2 \dots A_n \in F^+ \Rightarrow X \rightarrow Y \in F^+$$

Chú ý: $X \rightarrow Y \in F^+ \leftrightarrow F \models X \rightarrow Y$

Người ta nói $X \rightarrow Y$ được suy dẫn logic từ F

Hay $X \rightarrow Y$ được suy dẫn bởi hệ tiên đề Armstrong từ F

I.5 Định nghĩa : suy dẫn theo quan hệ

Cho R là tập các thuộc tính, F là tập phụ thuộc hàm xác định trên U . Ta nói rằng phụ thuộc hàm f theo quan hệ từ F nếu:

$$\forall r(U) : r \text{ thỏa } F \leftrightarrow r \text{ thỏa } f$$

$$\text{Ký hiệu : } F \models f$$

F^* là tập tất cả các các phụ thuộc hàm được suy dẫn theo quan hệ từ F .

$$F^* = \{f \mid F \models f\}$$

I.6 Bổ đề 2:

Hệ tiên đề Armstrong là đúng đắn nghĩa là $X \rightarrow Y$ được suy ra từ hệ tiên đề Armstrong thì $X \rightarrow Y$ đúng trong mọi quan hệ mà mọi phụ thuộc hàm của F đúng.

I.7 Định lý: Hệ tiên đề Armstrong là đúng

Chứng minh: ta chỉ cần chứng minh 3 tiên đề đúng :

a1) $\forall Y \subseteq X \subseteq R$ thì $X \rightarrow Y$: hiển nhiên

$$(\forall r(R) \text{ thỏa } F : \forall t, t' \in r : t(X) = t'(X) \Rightarrow t(Y) = t'(Y) \text{ do } Y \subseteq X)$$

a2) $X \rightarrow Y, \forall Z \subseteq R \Rightarrow XZ \rightarrow YZ$

$$\forall r(R) \text{ thỏa } F : \forall t, t' \in r \text{ theo giả thiết ta có :}$$

$$t(X) = t'(X) \Rightarrow t(Y) = t'(Y) \quad (1)$$

$$\text{nếu } t(XZ) = t'(XZ) \Rightarrow t(X).t(Z) = t'(X).t'(Z)$$

$$t(Y).t(Z) \neq t'(Y).t'(Z) \Rightarrow t(Y) \neq t'(Y) \quad (2)$$

(1), (2) : mâu thuẫn, vô lý. Vậy : $t(XZ) = t'(XZ) \Rightarrow t(YZ) = t'(YZ)$

a3) $X \rightarrow Y, Y \rightarrow Z$ thì $X \rightarrow Z$

$\forall r(R)$ thỏa $F : t, t' \in r :$

$$t(X) = t'(X) \Rightarrow t(Y) = t'(Y) \quad \text{do } X \rightarrow Y$$

$$t(Y) = t'(Y) \Rightarrow t(Z) = t'(Z) \quad \text{do } Y \rightarrow Z$$

$$\text{Vậy } t(X) = t'(X) \Rightarrow t(Z) = t'(Z) \quad \text{đpcm.}$$

I.5 . Hệ quả từ hệ tiên đề Armstrong :

Dễ dàng chứng minh

Bổ đề 3 :

b1) Qui tắc hợp : (*The union rule*) : $\{ X \rightarrow Y, X \rightarrow Z \} \models X \rightarrow YZ$

b2) Qui tắc giả bắc cầu : (*The pseudotransitivity rule*)

$$\{ X \rightarrow Y, WY \rightarrow Z \} \models WX \rightarrow Z.$$

b3) Qui tắc phân rã (*The decomposition rule*)

$$X \rightarrow Y, Z \subseteq Y \text{ thì } X \rightarrow Z$$

I.6 Tính đầy đủ của hệ tiên đề Armstrong :

Chứng minh tính đầy đủ (*Completeness*) của hệ tiên đề Armstrong có nghĩa là chứng minh : $F^* \subseteq F^+$.

$$\Leftrightarrow (X \rightarrow Y \notin F^+ \Rightarrow X \rightarrow Y \notin F^*) \quad (\text{Phản chứng})$$

Chứng minh :

Giả sử $X \rightarrow Y \notin F^+$

Xét quan hệ r gồm hai bộ như sau :

$$\begin{array}{cc} 11 \dots 1 & 11 \dots 1 \\ 11 \dots 1 & 00 \dots 0 \end{array}$$

Các thuộc tính
thuộc X^+

Các thuộc tính
còn lại

Ta thấy tất cả các phụ thuộc hàm trên F đều thỏa r : giả sử đảo lại :

$\exists U \rightarrow V \in F$ nhưng không thỏa $r \Rightarrow U \subseteq X^+, V \not\subseteq X^+ \Rightarrow \exists A \subseteq V,$

$A \in V \cap (R - X^+) \Rightarrow X \rightarrow A \in F^+$ vô lý.

Nếu $X \rightarrow Y \in F^* \Rightarrow X \rightarrow Y$ thỏa $r \Rightarrow Y \subseteq X^+ \Rightarrow X \rightarrow Y \in F^+$ vô lý

Vậy : $F^* \subseteq F^+$.

Chú ý: $F^+ \equiv F^*$. Khái niệm suy dẫn logic và suy dẫn quan hệ là đồng nhất.

I.7 Tính bao đóng của tập các thuộc tính đối với một tập phụ thuộc hàm :

Input : Cho U tập hữu hạn các thuộc tính, $X \in U$, F tập các phụ thuộc hàm trên U .

Output : X^+ (bao đóng của X đối với F)

Method:

$$B1) X_0 = X$$

$$B2) X_{i+1} = X_i \cup A \text{ sao cho } \exists (Y \rightarrow Z) \in F, A \in Z, Y \subseteq X_i$$

Do $X = X_0 \subseteq X_1 \subseteq \dots \subseteq U$, U hữu hạn nên $\exists i, X_i = X_{i+1}$ khi đó $X^+ = X_i$

Ví dụ 2 : Cho $F = \{A \rightarrow D, AB \rightarrow E, BI \rightarrow E, CD \rightarrow I, E \rightarrow C\}$

Tìm $X^+ = (AE)^+$

$$B_0 : X_0 = AE$$

$$B_1 : X_1 = AEDC$$

$$B_2 : X_2 = AEDCI$$

$$B_3 : X_3 = AEDCI$$

$$B_4 : X_4 = AEDCI \text{ Vậy } X^+ = AEDCI$$

I.8 Tính đúng của thuật toán tìm bao đóng tập thuộc tính

Chứng minh:

1. Thuật toán dừng do $|U| < \infty, |F| < \infty$

2. Gọi i_0 là số nguyên nhỏ nhất : $X_{i_0} = X_{i_0+1} = X_{i_0+2} = \dots$

Ta có : $X = X_0 \subset X_1 \subset X_2 \subset \dots \subset X_{i_0}$

Ta chứng minh : $X_{i_0} = X^+ \leftrightarrow X_{i_0} \subseteq X^+ \text{ (1) và } X^+ \subseteq X_{i_0} \text{ (2)}$

(1): Hiển nhiên

(2): Giả sử $\exists A \in X^+ \Rightarrow A \notin X_{i_0}$

Xây dựng quan hệ r gồm hai bộ như sau :

$$\begin{array}{cc} \underbrace{11 \dots 1}_{\text{Thuộc } X_{i_0}} & \underbrace{11 \dots 1}_{\text{còn lại}} \\ \underbrace{11 \dots 1}_{\text{Thuộc } X_{i_0}} & \underbrace{00 \dots 0}_{\text{còn lại}} \end{array}$$

Các thuộc tính Thuộc X_{i_0} Các thuộc tính còn lại

Ta có r thỏa $F \Rightarrow X \rightarrow A \in F^+$ (do tính đủ của hệ tiên đề Armstrong)

$\Rightarrow A \in X_{i_0}$ vô lý. Nói khác hơn $X^+ \subseteq X_{i_0}$.

Qui ước: Một lược đồ quan hệ R , trên đó xác định 1 tập phụ thuộc hàm F được gọi là 1 sơ đồ quan hệ $s = \langle R, F \rangle$

II. PHỦ CÁC PHỤ THUỘC HÀM: (*Covers for functional Dependencies*)

II.1. Định nghĩa : Hai tập phụ thuộc hàm tương đương

Hai tập phụ thuộc hàm F và G trên lược đồ R là tương đương, ký hiệu $F \equiv G$, nếu $F^+ = G^+$. Nếu $F \equiv G$ thì F gọi là phủ G .

F suy dẫn G , ký hiệu $F \models G$ nếu $\forall X \rightarrow Y \in G$ thì $F \models X \rightarrow Y$

II.2. Định lý : Điều kiện 2 tập phụ thuộc hàm tương đương

Cho hai tập phụ thuộc hàm F và G trên lược đồ R , $F \equiv G$ khi và chỉ khi $F \models G$ và $G \models F$.

Chứng minh : Trước hết ta chứng minh : $F^+ = (F^+)^+$

Theo định nghĩa :

$$F^+ = \{X \rightarrow Y \mid F \models X \rightarrow Y\}$$

$$(F^+)^+ = \{X \rightarrow Y \mid F^+ \models X \rightarrow Y\}$$

Ta chứng minh :

$F^+ \subseteq (F^+)^+$: hiển nhiên từ định nghĩa

$$(F^+)^+ \subseteq F^+$$

$$(\forall X \rightarrow Y \in (F^+)^+ \Leftrightarrow F^+ \models X \rightarrow Y) \Rightarrow F \models X \rightarrow Y \Rightarrow (X \rightarrow Y) \in F^+$$

(Vì $\forall r$ thỏa F^+ thì r cũng thỏa $X \rightarrow Y \Rightarrow \forall r$ thỏa F thì r cũng thỏa $X \rightarrow Y$ do $F \subseteq F^+$)

Chứng minh : $F \equiv G \Leftrightarrow (F \models G \text{ và } G \models F)$

$$F \models G \Rightarrow G \subseteq F^+ \Rightarrow G^+ \subseteq (F^+)^+ = F^+ \quad (1)$$

$$G \models F \Rightarrow F \subseteq G^+ \Rightarrow F^+ \subseteq (G^+)^+ = G^+ \quad (2)$$

$$(1), (2) \Leftrightarrow (G^+ \subseteq F^+) \wedge (F^+ \subseteq G^+) \Leftrightarrow F^+ = G^+$$

II.3. Các thuật toán liên quan kiểm tra sự tương đương của hai phụ thuộc hàm:

c1 : Gọi thuật toán I.7 là CLOSURE (X, F)

c2 : Thuật toán kiểm tra $F \models X \rightarrow Y$? là

MEMBER ($F, X \rightarrow Y$)

begin

if $Y \subset \text{CLOSURE}(X, F)$ then return (true) else

return (false)

end

c3 : Thuật toán kiểm tra $F \models G$?

DERIVES (F, G)

begin

$v := \text{true};$

```

for FD  $X \rightarrow Y \in G$  do
     $v := v$  and MEMBER (  $F, X \rightarrow Y$  ) ;
    return( $v$ )
end;
c4 : Kiểm tra sự tương của hai tập phụ thuộc hàm  $F, G$ 
EQUIV    ( $F, G$ )
begin
     $v :=$  DERIVERS ( $F, G$ ) and DERIVERS ( $G, F$ );
    return ( $v$ );
end;

```

III.4. Bổ đề 4:

Mỗi tập phụ thuộc hàm F đều tương với tập phụ thuộc phụ thuộc hàm G trong đó các vế phải phải không có quá 1 thuộc tính.

Chứng minh : Gọi G là tập phụ thuộc $X \rightarrow A$ sao cho tồn tại 1 phụ thuộc $X \rightarrow Y$ thuộc F , mà $A \in Y$. Người ta nói rằng $X \rightarrow A$ được suy ra từ $X \rightarrow Y$ bằng qui tắc phân rã.

Vậy $G \subseteq F^+$. Ta cũng có $F \subseteq G^+$ vì nếu $Y = A_1 A_2 \dots A_n$ thì $X \rightarrow Y$ được suy ra từ $X \rightarrow A_1, \dots, X \rightarrow A_n$ nhờ luật hợp. Vì vậy $F \equiv G$.

III.5 Định nghĩa : Tập phụ thuộc hàm cực tiểu

Một tập hợp phụ thuộc hàm F gọi là cực tiểu nếu :

- 1) Vế phải của mỗi phụ thuộc trong F chỉ có 1 thuộc tính độc nhất.
- 2) Không tồn tại bất kỳ một $X \rightarrow A$ nào trong F mà tập

$$(F - \{X \rightarrow A\})^+ = F^+$$

- 3) Không tồn tại 1 phụ thuộc hàm $X \rightarrow A$ thuộc F và 1 tập con Z của X :

$$F^+ = (F - \{X \rightarrow A\} \cup \{Z \rightarrow A\})^+$$

Ý nghĩa:

- Điều kiện 2 để đảm bảo không có phụ thuộc hàm nào trên F là dư thừa.
- Điều kiện 3 đảm bảo không có thuộc tính nào ở vế trái là dư thừa.
- Điều kiện 1 đảm bảo không có thuộc tính nào ở vế phải là dư thừa.

III.6 Định nghĩa : Phủ cực tiểu của 1 tập phụ thuộc hàm

Cho F là tập phụ thuộc hàm, G là tập phụ thuộc hàm, G là cực tiểu và $G \equiv F$ thì G được gọi là 1 phủ cực tiểu (*Minimal cover*) của F .

III.7 Định lý : Tồn tại phủ tối thiểu của 1 tập phụ thuộc hàm

Mỗi tập phụ thuộc hàm F đều có 1 phủ cực tiểu

Chứng minh : – Nhờ bổ đề 4 $\Rightarrow \exists F' \equiv F$, mà mỗi phụ thuộc hàm của F' vế phải chỉ có 1 thuộc tính.

Từ F' nếu bất kỳ $X \rightarrow A \in F$ mà $(F' - \{X \rightarrow A\})^+ = F^+$.

Ta chọn $F'' = F^+ - \{X \rightarrow A\}$

Dễ thấy F'' thỏa 1,2 và $F \equiv F''$.

Phép loại bỏ thứ 3 cũng tạo ra 1 tập phụ thuộc hàm tương đương với F .
(đpcm)

III.8 Thuật toán tìm phủ tối thiểu: (Bernstein, 1976)

Input : Cho lược đồ quan hệ R , F là tập phụ thuộc hàm xác định trên R

Output: Một phủ tối thiểu của F

Method:

B1: Tách các phụ thuộc hàm có vế phải là 1 thuộc tính theo qui tắc phân rã

For $X \rightarrow A \in F$ do

IF $A = A_1 A_2 \dots A_n$ then

$F := F \setminus \{X \rightarrow A\} \cup \{X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n\}$

B2: Loại bỏ thuộc tính dư thừa ở vế trái

for $A_1 A_2 \dots A_k \rightarrow B \in F$ do

if $EQUI((F \setminus \{A_1 A_2 \dots A_k \rightarrow B\} \cup \{A_1 A_2 \dots A_{i-1} A_{i+1} \dots A_k \rightarrow B\}), F)$

then $F := F \setminus \{A_1 A_2 \dots A_k \rightarrow B\} \cup \{A_1 A_2 \dots A_{i-1} A_{i+1} \dots A_k \rightarrow B\}$

B3: Loại bỏ phụ thuộc hàm dư thừa:

if $EQUI((F \setminus \{X \rightarrow A\}), F)$ then $F := F \setminus \{X \rightarrow A\}$

Nhận xét:

1. Đối với B2: Loại bỏ thuộc tính dư thừa ở vế trái, có nghĩa là phải loại bỏ các thuộc tính $X \setminus Z$, nếu phụ thuộc hàm $X \rightarrow A$ thuộc F và $Z \subset X$ mà :

$$F^+ = (F - \{X \rightarrow A\} \cup \{Z \rightarrow A\})^+$$

Thực ra ta không cần tiến hành:

for $A_1 A_2 \dots A_k \rightarrow B \in F$ do

if $EQUI((F \setminus \{A_1 A_2 \dots A_k \rightarrow B\} \cup \{A_1 A_2 \dots A_{i-1} A_{i+1} \dots A_k \rightarrow B\}), F)$

then $F := F \setminus \{A_1 A_2 \dots A_k \rightarrow B\} \cup \{A_1 A_2 \dots A_{i-1} A_{i+1} \dots A_k \rightarrow B\}$

Vì theo định lý II.2 : $F^+ = (F - \{X \rightarrow A\} \cup \{Z \rightarrow A\})^+ \Leftrightarrow$

a. $F \models (F - \{X \rightarrow A\} \cup \{Z \rightarrow A\})$ và

b. $(F - \{X \rightarrow A\} \cup \{Z \rightarrow A\}) \models F$

Mà b. là hiển nhiên vì :

$Z \subset X \Rightarrow X \rightarrow Z$ (Tính phản xạ) (1); $Z \rightarrow A$ (ghiết) (2) .

Từ (1)(2), theo tính bắc cầu ta có : $X \rightarrow A \in (F - \{X \rightarrow A\} \cup \{Z \rightarrow A\})^+$

Hay $(F - \{X \rightarrow A\} \cup \{Z \rightarrow A\}) \models F$.

Kiểm tra tính chất a. cũng chính là kiểm tra: $F \models \{Z \rightarrow A\}$, vì $(F - \{X \rightarrow A\} \cup \{Z \rightarrow A\})$ sai khác F phụ thuộc hàm $\{Z \rightarrow A\}$, có nghĩa là xét xem:

$$A \in Z_F^+ ?$$

Vì vậy, để kiểm tra loại bỏ thuộc tính dư thừa ở vế trái ta thực hiện

for $A_1 A_2 \dots A_k \rightarrow B \in F$ do

if $MEMBER(F, A_1 A_2 \dots A_{i-1} A_{i+1} \dots A_k \rightarrow B)$

then $F := F \setminus \{A_1 A_2 \dots A_k \rightarrow B\} \cup \{A_1 A_2 \dots A_{i-1} A_{i+1} \dots A_k \rightarrow B\}$

Đối với B3: Loại bỏ phụ thuộc hàm dư thừa, nghĩa là cần loại bỏ các phụ thuộc hàm: $X \rightarrow A$ nào trong F mà tập $(F - \{X \rightarrow A\})^+ = F^+$

Tương tự như trên, ta không cần tiến hành kiểm tra loại bỏ:

if EQUI $((F \setminus \{X \rightarrow A\}), F)$ then $F := F \setminus \{X \rightarrow A\}$

Vì : $(F - \{X \rightarrow A\})^+ = F^+ \Leftrightarrow$

a. $F \models F - \{X \rightarrow A\}$ và

b. $F - \{X \rightarrow A\} \models F$

Mà a. là hiển nhiên vì $F - \{X \rightarrow A\} \subset F$. Vì vậy để kiểm tra loại bỏ phụ thuộc hàm dư thừa, ta chỉ cần kiểm tra b. Mà kiểm tra $F - \{X \rightarrow A\} \models F$ cũng chính là kiểm tra $F - \{X \rightarrow A\} \models X \rightarrow A$, vì F sai khác $F - \{X \rightarrow A\}$ chỉ phụ thuộc hàm $X \rightarrow A$. Có nghĩa là B3 được tiến hành:

if MEMBER $((F \setminus \{X \rightarrow A\}), X \rightarrow A)$ then $F := F \setminus \{X \rightarrow A\}$

Nếu tiến hành giải thuật trên bằng cách loại bỏ phụ thuộc hàm dư thừa (B3), rồi mới tiến hành loại bỏ thuộc tính dư thừa (B2), thì kết quả thu được chưa chắc là phủ tối thiểu của tập phụ thuộc hàm đã cho.

Ví dụ 3: Xét lược đồ quan hệ $R=ABC$, F là tập phụ thuộc hàm xác định trên R , $F = \{AC \rightarrow B, C \rightarrow A, B \rightarrow A\}$.

Nếu tiến hành theo thứ tự:

B1: (Phân rã các phụ thuộc hàm)

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

B3: (Loại bỏ phụ thuộc hàm dư thừa)

$AC \rightarrow B$: Do $AC^+_{F \setminus \{AC \rightarrow B\}} = \{AC \supseteq B \Rightarrow AC \rightarrow B$ không dư

$C \rightarrow A$: Do $C^+_{F \setminus \{C \rightarrow A\}} = \{C \supseteq A \Rightarrow C \rightarrow A$ không dư

$B \rightarrow A$: Do $B^+_{F \setminus \{B \rightarrow A\}} = \{B \supseteq A \Rightarrow B \rightarrow A$ không dư

B2: Loại bỏ thuộc tính dư thừa ở vế trái

$AC \rightarrow B$: $A^+_F \not\models A \supseteq B$

$C^+_F = CAB \supseteq B \Rightarrow A$ dư (loại)

Vậy theo 3 bước trên ta có kết quả : $F = \{C \rightarrow B, C \rightarrow A, B \rightarrow A\}$

Nhưng dễ thấy F chưa phải là 1 phủ tối thiểu vì $C \rightarrow A$ là dư thừa.

Ví dụ 4: Cho lược đồ quan hệ $R = ABCDEFG$ trên đó có xác định tập phụ thuộc hàm :

$F = \{BF \rightarrow AG, BE \rightarrow ABC, CD \rightarrow AE, E \rightarrow BC, AF \rightarrow B\}$

Tìm 1 phủ tối thiểu của F .

B1: Xây dựng tập phụ thuộc hàm G như sau:

(Bằng cách tách vế phải các phụ thuộc hàm đã cho)

$BF \rightarrow A \quad BF \rightarrow G \quad BE \rightarrow A \quad BE \rightarrow B \quad BE \rightarrow C \quad CD \rightarrow A \quad CD \rightarrow E$

$E \rightarrow B \quad E \rightarrow C \quad AF \rightarrow B$

B2: (Loại bỏ thuộc tính dư thừa ở vế trái):

$BF \rightarrow A: F^+ = F \Rightarrow A \notin F^+$

$B^+ = B \Rightarrow A \notin B^+$

$BF \rightarrow G: F^+ = F \Rightarrow G \notin F^+$

$B^+ = B \Rightarrow G \notin B^+$

$BE \rightarrow A: E^+ = EBCA \Rightarrow A \in E^+ \Rightarrow B$ dư (loại)

$BE \rightarrow B: E^+ = EABC \Rightarrow B \in F^+ \Rightarrow B$ dư (loại)

$BE \rightarrow C: E^+ = EABC \Rightarrow C \in F^+ \Rightarrow B$ dư (loại)

$CD \rightarrow A: D^+ = D \Rightarrow A \notin D^+$

$C^+ = C \Rightarrow A \notin C^+$

$CD \rightarrow E: D^+ = D \Rightarrow E \notin D^+$

$C^+ = C \Rightarrow E \notin C^+$

$AF \rightarrow B: F^+ = F \Rightarrow B \notin D^+$

$A^+ = A \Rightarrow B \notin C^+$

B3: (Loại bỏ phụ thuộc hàm dư thừa)

$BF \rightarrow A: BF^+_{G \setminus \{BF \rightarrow A\}} = BFG \Rightarrow A \notin BF^+_{G \setminus \{BF \rightarrow A\}}$

$BF \rightarrow G: BF^+_{G \setminus \{BF \rightarrow G\}} = BFA \Rightarrow G \notin BF^+_{G \setminus \{BF \rightarrow G\}}$

$E \rightarrow A: E^+_{G \setminus \{E \rightarrow A\}} = EBC \Rightarrow A \notin E^+_{G \setminus \{E \rightarrow A\}}$

$CD \rightarrow A: CD^+_{G \setminus \{CD \rightarrow A\}} = CDEAB \Rightarrow A \in CD^+_{G \setminus \{CD \rightarrow A\}}$

$\Rightarrow CD \rightarrow A$ dư (loại)

$CD \rightarrow E: CD^+_{G \setminus \{CD \rightarrow E\}} = CD \Rightarrow E \notin CD^+_{G \setminus \{CD \rightarrow E\}}$

$E \rightarrow B: E^+_{G \setminus \{E \rightarrow B\}} = EAC \Rightarrow B \notin E^+_{G \setminus \{E \rightarrow B\}}$

$E \rightarrow C: E^+_{G \setminus \{E \rightarrow C\}} = EAB \Rightarrow C \notin E^+_{G \setminus \{E \rightarrow C\}}$

$AF \rightarrow B: AF^+_{G \setminus \{AF \rightarrow B\}} = AF \Rightarrow B \notin AF^+_{G \setminus \{AF \rightarrow B\}}$

Vậy 1 phủ tối thiểu của F là :

$G = \{ BF \rightarrow A, BF \rightarrow G, E \rightarrow A, E \rightarrow B, E \rightarrow C, CD \rightarrow E, BF \rightarrow AF \rightarrow B \}$

Chú ý: Nếu thay đổi thứ tự xét các phụ thuộc hàm trong các bước có thể sinh ra các phủ tối thiểu khác nhau.

IV. KHÓA – THUẬT TOÁN XÁC ĐỊNH KHÓA

Cho sơ đồ quan hệ $S = \langle R, F \rangle$, $K \subseteq R$ được gọi là 1 khóa của S nếu:

1. K là siêu khóa $\Leftrightarrow K^+ = R$ (Chứng minh như bài tập)

2. K^+ là tập bé nhất thỏa 1.

IV.1 Thuật toán : Tìm khóa tối tiểu của 1 sơ đồ quan hệ

Input : F tập phụ thuộc hàm xác định trên lược đồ quan hệ $R = A_1A_2...A_n$

Output : K là 1 khóa tối tiểu của lược đồ quan hệ R

Method:

B1 : $K_0 = R = A_1 A_2 \dots A_n$
 B2 : $i := 1$
 B3 : $K_i = K_{i-1}$ nếu $(K_{i-1} \setminus \{A_i\})^+ \neq R$ hoặc
 $K_i = K_{i-1} \setminus \{A_i\}$, ngược lại.
 B4 : $i := i + 1$
 B5 : $i := n \rightarrow K_n$ là khóa \rightarrow dừng
 B6 : $i < n \rightarrow B3$.

Ví dụ 5: Xét $R = A_1 A_2 A_3 A_4$
 $F = \{ A_1, A_2 \rightarrow A_4, A_3 \rightarrow A_2 \}$

Theo thuật toán trên ta có:

$K_0 = A_1 A_2 A_3 A_4 = R$
 $K_1 = A_1 A_2 A_3 A_4$ vì $(A_1 A_2 A_3)^+_F \neq R$
 $K_2 = A_1 A_3 A_4$ vì $(A_1 A_3 A_4)^+_F = R$
 $K_3 = K_2$
 $K_4 = A_1 A_3$ vì $(A_1 A_3)^+_F = R$
 Vậy khóa là $K = A_1 A_3$

Chú ý : Nếu thay đổi thứ tự xét các thuộc tính trong thuật toán trên có thể dẫn đến nhiều khoá khác nhau.

IV.2 Định lý: Thuật toán xác định khoá tối thiểu của sơ đồ quan hệ IV.1 là đúng.

Chứng minh: Xét lược đồ quan hệ R , tập phụ thuộc hàm F xác định trên R ,
 Giả sử có dạng: $F = \{ L_i \rightarrow R_i, i = 1..n \}$ Đặt:

$$L = \bigcup_1^n L_i, R^* = \bigcup_1^n R_i$$

Theo thuật toán : $K^+ = R \Rightarrow K$ là 1 siêu khoá.

Cũng theo thuật toán, ta thấy nếu $K \subset (R \setminus R^*) \cup A_{ik1} A_{ik2} \dots A_{ikn}$

với : $A_{ik1} A_{ik2} \dots A_{ikn}$ là tập bé nhất thuộc : $L \cap R^*$ thoả : $K^+ = R$.

Nếu ở $\exists A_{ikm}$ sao cho:

$$A_{ik1} A_{ik2} \dots A_{ikm-1} A_{ikm+1} \dots A_{ikn} = Z$$

mà $[(R \setminus R^*) \cup Z]^+ = R$, thì ở bước k_m , A_{ikm} đã bị loại.

Do đó, K là tập siêu khoá bé nhất. (đpcm)

IV.3 Thuật toán : Tìm khóa tối thiểu của 1 quan hệ r

Input : Cho quan hệ $r = \{h_1, h_2, \dots, h_m\}$ trên xác định trên lược đồ quan hệ
 $R = A_1 A_2, \dots, A_n$

Output : K là 1 khóa tối thiểu của r

Method:

B1 : Tính $E_r = \{E_{ij} : 1 \leq i < j \leq m\}$

Với $E_{ij} = \{A_i \in R, h_i(A_i) = h_j(A_j)\}$

B2 : Tính $M_r = \{A, \text{ có } A_j \in E_r : A = A_j \text{ và không có } A_i \in E_r \text{ mà } A \subset A_i, A \neq A_i\}$

B3 : Lần lượt tính các tập thuộc tính $K_0, K_1, K_2, \dots, K_n$ theo qui tắc :

$$K_0 = R = A_1 A_2 \dots A_n$$

$$K_i = \begin{cases} K_{i-1} - \{A_i\}, & \text{nếu } \exists A \in M_r \text{ sao cho } K_{i-1} - \{A_i\} \subseteq A \\ K_{i-1}, & \text{ngược lại.} \end{cases}$$

B4 : $K = K_n$

K là khóa tối thiểu.

Ví dụ 7: Cho quan hệ r

A	B	C	D	E
1	1	0	3	0
0	1	0	1	1
0	0	1	1	0
2	0	2	0	1

$E_{12} = BC, E_{13} = E, E_{14} = 0, E_{23} = AD, E_{24} = E, E_{34} = B,$

$M_r = \{BC, E, AD\}$

$K_1 = BCDE, K_2 = CDE, K_3 = DE, K_4 = K_3, K_5 = K_4$

Vậy DE là một khóa tối thiểu của r

Chú ý : Nếu ta thay đổi thứ tự của $R = \{E, D, C, B, A\}$

$K_1 = DCBA, K_2 = CBA, K_3 = BA, K_4 = AB$ và $A \subset \{A_1\}$

$K_4 - A = B$ là một phần tử của $M_r \Rightarrow AB$ là một khóa tối thiểu của r .

IV.4 Một số tính chất của khoá và siêu khoá:

Không mất tính tổng quát ta có thể giả sử như sau:

Xét sơ đồ quan hệ $S = \langle R, F \rangle$, tập phụ thuộc hàm F

$F = \{L_i \rightarrow R_i, i=1, k\}$ thỏa :

$$L_i \cap R_i = \emptyset$$

$$L_i \neq R_i, i \neq j$$

$$\text{Đặt : } \mathcal{R} = \bigcup_1^k R_i \quad \mathcal{L} = \bigcup_1^k L_i$$

IV.5 Bổ đề 5:

$$X \rightarrow Y \in F^+, A \notin \mathcal{L} \Rightarrow X \setminus \{A\} \rightarrow Y \setminus \{A\} \in F^+$$

Chứng minh: Ta có: $X \rightarrow Y \in F \Rightarrow Y \subset X^+ \Rightarrow Y \setminus \{A\} \subset X^+ \setminus \{A\}$

Mà :

$$\text{hoặc } (X \setminus \{A\})^+ = X^+, \text{ hoặc } (X \setminus \{A\})^+ = X^+ \setminus \{A\}$$

(Chứng minh quy nạp theo thuật toán tìm bao đóng)
 $\Rightarrow Y \setminus \{A\} \subset X^+ \setminus \{A\} \Rightarrow X \setminus \{A\} \rightarrow Y \setminus \{A\} \in F^+$ (đ.p.c.m)

IV.6 Bổ đề 6:

$X \subset R, A \in X$. Nếu $X \setminus \{A\} \rightarrow A \in F^+ \Rightarrow X$ không là khoá

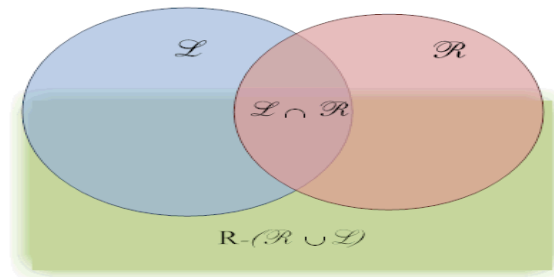
Chứng minh: Nếu có: $X \setminus \{A\} \rightarrow X \setminus \{A\}$ và $X \setminus \{A\} \rightarrow A$, theo qui tắc hợp ta có: $X \setminus \{A\} \rightarrow X \in F^+$.

Nếu X là khoá $\Rightarrow X \setminus \{A\}$ là siêu khoá, và $X \setminus \{A\} \subset X$: vô lý (đpcm)

IV.7 Định lý (ĐK cần) [Hồ Thuận & Lê Văn Bào, 1985]

Cho sơ đồ quan hệ $s = \langle R, F \rangle$, $X \subset R$. Nếu X là khoá thì :

$$(R \setminus \mathcal{R}) \subseteq X \subseteq (R \setminus \mathcal{R}) \cup (\mathcal{L} \cap \mathcal{R})$$



Chứng minh:

$$(R \setminus \mathcal{R}) \subseteq X:$$

Theo thuật toán tìm bao đóng:

$$X^+ \subseteq X \cup \mathcal{R} \subseteq R$$

vì X là khoá :

$$R = X^+ = X \cup \mathcal{R} \subseteq R \Rightarrow X \cup \mathcal{R} = R$$

$$(R \setminus \mathcal{R}) \subseteq X \text{ (đpcm)}$$

$$X \subseteq (R \setminus \mathcal{R}) \cup (\mathcal{L} \cap \mathcal{R})$$

$$\text{Ta có : } X \subseteq R = (R \setminus \mathcal{R}) \cup (\mathcal{L} \cap \mathcal{R}) \cup (\mathcal{R} \setminus \mathcal{L})$$

Vậy để có điều phải chứng minh, ta đi chứng minh:

$$X \cap (\mathcal{R} \setminus \mathcal{L}) = \emptyset$$

(Chứng minh bằng phản chứng)

$$\text{Giả sử } X \cap (\mathcal{R} \setminus \mathcal{L}) \neq \emptyset \Rightarrow \exists A \in R: A \in X \cap (\mathcal{R} \setminus \mathcal{L})$$

$$\text{vì } X \text{ là khoá : } X \rightarrow R \in F^+, A \notin \mathcal{L}, A \in \mathcal{R}$$

$$\text{theo bổ đề 5 : } X \setminus A \rightarrow R \setminus A \in F^+$$

Do tính phản xạ : $R \setminus A \rightarrow \mathcal{L}$, mà $\mathcal{L} \rightarrow \mathcal{R}$, nên theo luật bắc cầu:

$$R \setminus A \rightarrow \mathcal{R} \text{ (chú ý } \mathcal{R} \rightarrow A, \text{ vì } A \in \mathcal{R})$$

Vì vậy : $X \setminus A \rightarrow A \in F^+$, theo bổ đề 6 $\Rightarrow X$ không là khoá (vô lý)

Vậy: $X \cap (\mathcal{R} \setminus \mathcal{L}) = \emptyset$, nói khác hơn :

$$X \subseteq (R \setminus \mathcal{R}) \cup (\mathcal{L} \cap \mathcal{R})$$

IV.8 Hệ quả:

a. Nếu $(\mathcal{L} \cap \mathcal{R}) = \emptyset$. Lược đồ có 1 khoá duy nhất $K = R \setminus \mathcal{R}$

Ví dụ 8: (Lucchesi & Osborn)

Cho $R = \{A_1, A_2, \dots, A_{10}\}$, $F = \{A_1 \rightarrow A_2, A_3 \rightarrow A_4, \dots, A_9 \rightarrow A_{10}\}$

$\mathcal{L} = A_1 A_3 \dots A_9$, $\mathcal{R} = A_2 A_4 \dots A_{10}$

$(\mathcal{L} \cap \mathcal{R}) = \emptyset \Rightarrow R \setminus \mathcal{R} = \mathcal{L} = A_1 A_3 \dots A_9$ là khoá duy nhất (!)

b. Nếu ký hiệu $\bigcap_i K_i = G$, K_i là 1 khoá của lược đồ quan hệ:

$$R \setminus \mathcal{R} \subseteq G$$

Chú ý : $(R \setminus \mathcal{R}) \cup (\mathcal{L} \cap \mathcal{R})$ chưa chắc là khoá (!)

IV.9 Thuật toán cải biên để tìm khoá của một sơ đồ quan hệ:

Input : $S = \langle R, F \rangle$

Output : K là một khoá của R

Method:

Tương tự thuật toán tìm khoá trước, thay bước B_0 bởi:

$$B_0' = K_0 = (R \setminus \mathcal{R}) \cup (\mathcal{L} \cap \mathcal{R})$$

Chú ý : Ta có thể sử dụng điều kiện cần trên để giải các bài toán:

Xác định 1 khoá từ 1 siêu khoá.

Kiểm tra tập thuộc tính có phải là khoá không.

IV.10 Định lý : [Lucchesi & Osborn – 1979]

Cho sơ đồ quan hệ $S = \langle R, F \rangle$, \mathcal{H} là một tập khác rỗng các khoá của S . Điều kiện cần và đủ để: $2^R \setminus \mathcal{H}$ chứa ít nhất 1 khoá của S là :

$$\exists L_{i0} \rightarrow R_{i0} \in F, \exists K \in \mathcal{H} : L_{i0} \cup (K \setminus R_{i0}) \text{ không chứa phần tử nào của } \mathcal{H} \quad (*)$$

(2^R tập các tập con của R)

Chứng minh:

a. Điều kiện đủ:

Ta có : $L_{i0} \rightarrow R_{i0} \in F$

$$K \setminus R_{i0} \rightarrow K \setminus L_{i0}$$

$$\Rightarrow L_{i0} \cup (K \setminus R_{i0}) \rightarrow (K \setminus R_{i0}) \cup R_{i0} \supseteq K$$

$\Rightarrow L_{i0} \cup (K \setminus R_{i0})$ là 1 siêu khoá chứa ít nhất 1 khoá của R (theo giả thiết *)

$\Rightarrow L_{i0} \cup (K \setminus R_{i0})$ chứa ít nhất 1 khoá không thuộc \mathcal{H} (khoá thuộc $2^R \setminus \mathcal{H}$)

b. Điều kiện cần:

Gọi K' là 1 khoá của họ $2^R \setminus \mathcal{H}$ (đương nhiên K' không chứa khoá nào của \mathcal{H}). Gọi K'' là tập lớn nhất (tối đại) có tính chất: $K'' \supset K'$,

K'' không chứa phần tử nào của \mathcal{H} . (K'' là tồn tại, xấu nhất $K'' = K$)

Hiển nhiên $K'' \subset R$ và $K'' \neq R$

Do K'' là siêu khoá $\Rightarrow (K'')^+ = R$. Theo thuật toán tìm bao đóng:

$\exists L_{i0} \rightarrow R_{i0} : L_{i0} \subset K'', R_{i0} \not\subset K'' \Rightarrow K'' \cup R_{i0}$ chứa ít nhất 1 phần tử $K \in \mathcal{H}$ (do K'' là tối đại). Có nghĩa: $K \subset K'' \cup R_{i0}$ hay $K \setminus R_{i0} \subseteq K''$.

Ta cũng có: $L_{i0} \subset K''$. Do đó $L_{i0} \cup (K \setminus R_{i0}) \subset K''$, theo giả sử trên:

K'' không chứa phần tử nào của $\mathcal{H} \Rightarrow L_{i0} \cup (K \setminus R_{i0})$ không chứa phần tử nào của \mathcal{H} . (đpcm)

V. PHÂN RÃ THÔNG TIN (*Decomposition*):

Phân rã một lược đồ quan hệ $R = A_1 A_2 \dots A_n$ là việc thay lược đồ quan hệ R bằng một tập các lược đồ R_1, R_2, \dots, R_n trong đó: $R_i \subseteq R$, $i = 1..m$ và

$$R = \bigcup_{i=1}^m R_i$$

Ký hiệu $R = \rho(R_1, R_2, \dots, R_m)$

V.1. Phân rã không tổn thất thông tin (*Lossless joint decomposition*)

Phân rã 1 lược đồ quan hệ R thành R_1, R_2, \dots, R_n gọi là không tổn thất nếu:

$$\forall r(R): r = \Pi_{R_1}(r) \bowtie \Pi_{R_2}(r) \bowtie \dots \Pi_{R_n}(r)$$

Ví dụ 6: Xét lược đồ quan hệ $R = ABC$ và quan hệ $r(R)$

R	A	B	C
	a1	b1	c1
	a1	b2	c1

$\Pi_{AC} r(R)$

$\Pi_{BC} r(R)$

R1	A	C
	a1	c1
	a2	c1

R2	B	C
	b1	c1
	b2	c1

R	A	B	C
	a1	b1	c1
	a1	b2	c1
	a2	b1	c1
	a2	b2	c1

$$\Pi_{AC} r(R) \bowtie \Pi_{BC} r(R) \neq r(R)$$

V.2 Thuật toán kiểm tra phân rã tổn thất thông tin:

Input: Lược đồ quan hệ $R = A_1A_2...A_n$ và tập phụ thuộc hàm F

$$R = \rho (R_1, R_2, \dots, R_m)$$

Output: ρ là tập hợp phân rã có tổn thất hay không

Method:

Xây dựng một bảng n cột và k hàng, cột thứ j ứng với thuộc tính A_j , hàng i tương ứng với lược đồ quan hệ R_i ở vị trí (i,j) , đặt ký hiệu a_j nếu A_j thuộc R_i . nếu không đặt b_{ij} vào vị trí đó.

Xét lặp các phụ thuộc hàm thuộc F cho đến khi không còn thay đổi: nếu có $X \rightarrow A$: Những hàng có giá trị X bằng nhau, thì giá trị a tương ứng ta chuyển đổi bằng nhau. Nếu một trong hai ký hiệu là A_j thì chuyển ký hiệu kia thành a_j . Nếu cả hai ký hiệu là b_x, b_z thì có thể chuyển về 1 trong 2 giá trị tùy ý.

Kết thúc quá trình lặp nếu có 1 hàng toàn bộ giá trị a thì phân rã này không tổn thất. Ngược lại phân rã có tổn thất

Tính đúng đắn của giải thuật đã được chứng minh trong [7].

Ví dụ 7: Xét $R=ABCDE$, $R_1=AD$, $R_2=AB$, $R_3=BE$, $R_4=CDE$, $R_5=AE$

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

Từ bảng

A	B	C	D	E
a1	b12	b13	a4	b15
a1	a2	b23	b24	b25
b31	a2	b33	b34	a5
b41	b42	a3	a4	a5
a1	b52	b53	b54	a5



A	B	C	D	E
a1	b12	a3	a4	b15
a1	a2	a3	a4	b25
a1	a2	a3	a4	a5
a1	b42	a3	a4	a5
a1	b52	a3	a4	a5

Vậy phép kết nối không tổn thất

V.3 Định lý: Phép tách bảo toàn

Nếu $\rho=(R_1, R_2)$ là một phép tách của R và F là tập phụ thuộc hàm thì phép tách không mất mát thông tin đối với F khi và chỉ khi $R_1 \cap R_2 \rightarrow R_1 \setminus R_2$ hoặc $R_1 \cap R_2 \rightarrow R_2 \setminus R_1$ (Chứng minh như bài tập)

V.4 Phân rã bảo toàn phụ thuộc hàm

Định nghĩa

Xét lược đồ quan hệ R , $\rho = (R_1, R_2, \dots, R_k)$ là một phân rã của R , F là tập phụ thuộc hàm xác định trên R . Gọi

$$\Pi_Z(F) = \{ X \rightarrow Y \in F^+ \mid XY \subseteq Z \}$$

Phép phân rã ρ gọi là bảo toàn các phụ thuộc hàm nếu :

$$F \models \bigcup_{i=1}^k \Pi_{R_i}(F)$$

Ví dụ 8 : Xét lược đồ quan hệ $R = CSZ$, F là tập phụ thuộc hàm xác định trên R

$$F = \{ CS \rightarrow Z, Z \rightarrow C \}, R = \rho(SZ, CZ)$$

$$\Pi_{SZ} F = \{ Z \rightarrow Z, SZ \rightarrow S, SZ \rightarrow Z, SZ \rightarrow SZ \}$$

(Các phụ thuộc hàm tầm thường trên SZ).

$$\Pi_{CZ} F = \{ Z \rightarrow C, C \rightarrow C, Z \rightarrow Z, CZ \rightarrow C, CZ \rightarrow Z \}$$

$$\text{Vì } \Pi_{SZ}(F) \cup \Pi_{CZ}(F) \not\models F$$

Vậy ρ không bảo toàn các phụ thuộc.

Chú ý

1. Ta luôn có

$$F \models \bigcup_{i=1}^k \Pi_{R_i}(F)$$

ρ là phép phân rã bảo toàn phụ thuộc hàm thì ta có:

$$F^+ = \left(\bigcup_{i=1}^k \Pi_{R_i}(F) \right)^+$$

3. Một phép phân rã có thể **không tổn thất thông tin** nhưng không bảo toàn các phụ thuộc.

Ví dụ 9 : Với ví dụ 8 :

$$\text{Do } (SZ \cap CZ) = Z \rightarrow (CZ - SZ)$$

nên ρ là bảo toàn thông tin (nhưng ρ không bảo toàn các phụ thuộc hàm)

Thuật toán kiểm tra bảo toàn phụ thuộc hàm:(Beeri & Huneyman 1981)

Input : Cho lược đồ quan hệ R , F là tập phụ thuộc hàm xác định trên R .

$$R = \rho(R_1, R_2, \dots, R_k)$$

Output: Khẳng định ρ có bảo toàn F hay không

Method: Gọi $G = \bigcup_{i=1}^k \Pi_{R_i}(F)$

Theo nhận xét 1, ta chỉ cần kiểm tra $G \models F$? để nhận biết phép phân rã có bảo toàn phụ thuộc hay không \leftrightarrow kiểm tra $\forall X \rightarrow Y \in F, G \models X \rightarrow Y$?

\leftrightarrow kiểm tra $\forall X \rightarrow Y \in F : Y \subset X_G^+$?

Tuy vậy, ta không cần biết tường minh G , để tính X_G^+ ta cần tính X^+ bằng các phụ thuộc hàm ứng với các hình chiếu của F lên các R_i .

Cụ thể, để tính X^+ theo G ta thực hiện như sau:

$$Z := X$$

While Z còn thay đổi do

For $i := 1$ to k do

$$Z = Z \cup ((Z \cap R_i)_F^+ \cap R_i)$$

{ Bao đóng được lấy ứng với F }

Nếu $(Z \supset Y \Rightarrow X \rightarrow Y \in G^+) \Rightarrow$ sự bảo toàn của ρ .

Ví dụ 10: Xét lược đồ quan hệ $R = ABCD$ với các phép phân rã $\rho = \{AB, BC, CD\}$ tập các phụ thuộc hàm

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

Ta có :

$$\Pi_{AB} F \supseteq \{A \rightarrow B, B \rightarrow A\}$$

$$\Pi_{BC} F \supseteq \{B \rightarrow C, C \rightarrow B\}$$

$$\Pi_{CD} F \supseteq \{C \rightarrow D, D \rightarrow C\}$$

Do đó :

$$G = \Pi_{AB} F \cup \Pi_{BC} F \cup \Pi_{CD} F \supseteq \{A \rightarrow B, B \rightarrow A, B \rightarrow C, C \rightarrow B, C \rightarrow D, D \rightarrow C\}$$

Vì vậy để xét $G \models F$?, ta chỉ cần xét D_G^+

Theo thuật toán trên, xét $Z = \{D\}$

Với $R_1 = AB$

$$Z = \{D\} \cup \{D \cap AB\}_F^+ \cap AB = \{D\}$$

Với $R_2 = BC$

$$Z = \{D\} \cup \{D \cap BC\}_F^+ \cap BC = \{D\}$$

Với $R_3 = CD$

$$Z = \{D\} \cup \{D \cap CD\}_F^+ \cap CD = \{D, C\}$$

- Tiến hành cho lần lặp tiếp theo với $Z = \{C, D\}$

Với $R_1 = AB$

$$Z = \{C, D\} \cup \{\{C, D\} \cap AB\}_F^+ \cap AB = \{C, D\}$$

Với $R_2 = BC$

$$Z = \{C, D\} \cup \{\{C, D\} \cap BC\}_F^+ \cap BC = \{C, D, B\}$$

Với $R_3 = CD$

$$Z = \{C, D, B\} \cup \{\{C, D, B\} \cap CD\}_F^+ \cap CD = \{C, D, B\}$$

Tiến hành cho lần lặp tiếp với $Z = \{B, C, D\}$ (tương tự như trên)

Ta có :

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

Vì $Z \ni A \Rightarrow G \models F$

\Rightarrow Phép phân rã ρ bảo toàn các phụ thuộc hàm trên F .

Định lý : Tính đúng của thuật toán kiểm tra phân rã bảo toàn phụ thuộc hàm

Thuật toán trên xác định chính xác $X \rightarrow Y \in F$ có thuộc G^+ hay không.

Chứng minh Định lý đúng $\leftrightarrow X_G^+ = Z$

Chứng minh : Định lý đúng tương đương với chứng minh 2 điều sau:

$$1. X_G^+ \subset Z : X \rightarrow Y \in G^+ (\Rightarrow Y \subset X_G^+) \Rightarrow Y \subset Z$$

$$2. Z \subset X_G^+ : Y \subset Z \Rightarrow X \rightarrow Y \in G^+ (\Leftrightarrow Y \subset X_G^+)$$

Mà:

$$2) Y \subset Z \Rightarrow \exists f \in G : \forall_p (f) \supset Y \Rightarrow X \rightarrow Y \in G^+ \Rightarrow Y \subset X_G^+$$

1) $X \rightarrow Y \in G^+ \Rightarrow Y \subset X_G^+ \Rightarrow$ Các thuộc tính của Y sẽ được đưa vào X_G^+ bằng các phụ thuộc hàm của $G \Rightarrow Y \subset Z$ (theo thuật toán)
 $\Rightarrow Y \subset Z$. Từ (1) (2) ta có đpcm.

VI. CHUẨN HÓA LƯỢC ĐỒ QUAN HỆ:

-Trong một lược đồ quan hệ R trên tập thuộc tính $U=\{A_1,...,A_n\}$ thuộc tính

$A \in U$ gọi là thuộc tính khóa (*prime*) nếu A là thành phần thuộc một khóa nào đó của R , ngược lại A được gọi là thuộc tính không khóa (*nonprime*)

- Hai thuộc tính là độc lập lẫn nhau nếu giữa chúng không có phụ thuộc hàm.

VI.1 Ba dạng chuẩn đầu tiên:

Codd năm 1972 đã đề ra các dạng chuẩn CSDL. Nếu một CSDL được chuẩn hóa chúng ta sẽ tránh được việc dư thừa dữ liệu và tăng tốc độ các phép toán xử lý quan hệ.

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

Định nghĩa: Một lược đồ quan hệ R được gọi là ở dạng chuẩn 1 (**1NF**) nếu và chỉ nếu toàn bộ các miền có mặt trong R đều chỉ chứa các giá trị nguyên tố (atom)

Một thuộc tính là nguyên tố nếu không là:

- Đa trị (là một danh sách (list) hay tập các giá trị hoặc hợp các giá trị khác)
- Phân rã được thành các thuộc tính nhỏ hơn
- Có thể được kết xuất từ các thuộc tính khác.

Chú ý: Một thuộc tính được xem là nguyên tố hay không còn phụ thuộc yêu cầu và ngữ cảnh của bài toán.

Ví dụ 11: Xét quan hệ:

MSSV	HOTEN	NGAYSINH	NGOANGU	D1	D2	DTB
47TH123	Trần minh Văn	12/07/80	Anh Phap Nga	8.0	10.0	9.0
47TH124	Lê Thị Nguyệt	05/06/81	Phap	9.0	7.0	8.0
47TH125	Lê Anh	28/08/81	Anh, Phap	6.0	8.0	7.0
47TH126	Trịnh Minh Ngọc	12/01/81	Phap	6.0	8.0	7.0

Thuộc tính HOTEN không nguyên tố do có thể tách thành HOLOT (Họ lót) và TEN (Tên).

Thuộc tính NGOANGU không nguyên tố đa trị

Thuộc tính DTB không nguyên tố do có thể kết xuất từ 2 thuộc tính D1, D2 nên quan hệ HS không ở 1NF.

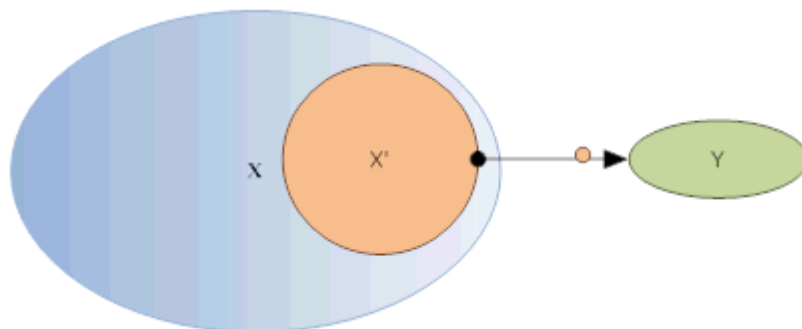
Thuộc tính NGAYSINH không tách thành NGÀY, THANG, NAM do các hệ quản trị CSDL đã mô tả kiểu dữ liệu Datetime để lưu trữ như một đơn nguyên.

Ta có thể chuẩn hóa quan hệ trên thành 1NF như sau:

MSSV	HOLOT	TEN	NGAYSINH	NGOANGU	D1	D2
47TH123	Trần Minh	Văn	12/07/80	Anh	8.0	10.0
47TH123	Trần Minh	Văn	12/07/80	Pháp	8.0	10.0
47TH123	Trần Minh	Văn	12/07/80	Nga	8.0	10.0
47TH124	Lê Thị	Nguyệt	05/06/81	Phap	9.0	7.0
47TH125	Lê	Anh	28/08/81	Anh	6.0	8.0
47TH125	Lê	Anh	28/08/81	Pháp	6.0	8.0
47TH126	Trịnh Minh	Ngọc	12/01/81	Phap	6.0	8.0

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

Định nghĩa: Cho tập phụ thuộc hàm F , phụ thuộc hàm $X \rightarrow Y \in F^+$, Y được gọi là phụ thuộc bộ phận vào X nếu $\exists X' \subset^7 X, X' \rightarrow Y \in F^+$ ngược lại Y phụ thuộc đầy đủ vào X .



Định nghĩa: Một sơ đồ quan hệ $s = \{R, F\}$ được gọi là ở dạng chuẩn hai (2NF), nếu nó ở dạng chuẩn 1 và nếu mỗi thuộc tính không khóa của R phụ thuộc hàm đầy đủ vào khóa⁸.

⁷ \subset : Ký hiệu tập con thực sự

⁸ Phụ thuộc đầy đủ vào tất cả các khóa

Ví dụ 12 : Xét quan hệ

$R(\text{CHUYEN BAY}, \text{NGAY BAY}, \text{CONG PHI CONG})$

$F = (\{\text{CHUYEN BAY}, \text{NGAY BAY}\} \rightarrow \text{PHI CONG},$

$\text{CHUYEN BAY} \rightarrow \text{PHI CONG})$

Để thấy: $\{\text{CHUYENBAY}, \text{NGAYBAY}\}$ là khóa chính

CONG chỉ phụ thuộc vào CHUYENBAY nên s không ở dạng chuẩn 2.

Chú ý: Nếu lược đồ không thỏa chuẩn 2 dễ dẫn đến trùng lặp dữ liệu.

Ví dụ 13: Xét 1 quan hệ r trên $S = \langle R, F \rangle$ xác định như sau:

SOHD	MSKH	MSMH	TENMH	SOLG	DONGIA
12345	A123	TV01	Ti Vi Sony -M27	20	1000
12345	A123	TL01	Tủ lạnh LG-M13	10	750
12346	A124	TV01	Ti Vi Sony -M27	15	1000
12347	A123	TL01	Tủ lạnh LG-M13	20	750

Ta thấy khóa của sơ đồ quan hệ trên là $K = \{\text{SOHD}, \text{MSMH}\}$

và: $\text{SOHD} \rightarrow \text{MSKH}, \text{MSMH} \rightarrow \text{TENMH}, \text{DONGIA}$

nên sơ đồ quan hệ trên không ở 2NF. Có thể tách lược đồ quan hệ trên thành các lược đồ con 2NF bảo toàn thông tin⁹ ứng với các quan hệ như sau:

MSMH	TENMH	DONGIA
TV01	Ti Vi Sony -M27	1000
TL01	Tủ lạnh LG-M13	750
TV01	Ti Vi Sony -M27	1000
TL01	Tủ lạnh LG-M13	750

SOHD	MSMH	SOLG
12345	TV01	20
12345	TL01	10
12346	TV01	15
12347	TL01	20

SOHD	MSKH
12345	A123
12345	A123
12346	A124
12347	A123

Sự tách này làm giảm sự trùng lặp thông tin.

c. Dạng chuẩn 3: (Third Normal Form – 3NF)

Định nghĩa : Cho sơ đồ quan hệ $s = \{R, F\}$, một tập thuộc tính $X \subseteq R$, $A \in R$, A được gọi là phụ thuộc bất cầu vào X trên s, nếu tồn tại một tập con $Y \subseteq R$ mà trên F^+ : $X \rightarrow Y, Y \not\rightarrow X, Y \rightarrow A$ và $A \notin XY$

Ví dụ 14: Xét $R = \text{ch}(\text{CHUYENBAY}, \text{MA_PHICONG}, \text{TEN})$

$F = \{\text{CHUYENBAY} \rightarrow \text{MA_PHICONG}, \text{TEN}$

$\text{MA_PHICONG} \rightarrow \text{TEN}, \text{TEN} \rightarrow \text{MA_PHICONG}\}$

TEN là phụ thuộc bất cầu vào CHUYENBAY vì :

⁹ Thuật toán tách một lược đồ con thành 2NF bảo toàn thông tin, xem như bài tập

CHUYENBAY \rightarrow MA_PHICONG,
 MA_PHICONG $\not\rightarrow$ CHUYENBAY,
 MA_PHICONG \rightarrow TEN.

Định nghĩa 1: Một sơ đồ quan hệ $s=\{R,F\}$ là ở dạng chuẩn 3 (3NF) nếu :

- s là ở dạng 1NF (1)
- Mọi thuộc tính không khóa không phụ thuộc bất cầu vào khóa.

Định lý : Bất kỳ lược đồ quan hệ R nào là 3NF với tập phụ thuộc hàm F đã cho thì cũng là 2NF với tập phụ thuộc đó.

Chứng minh : Chúng ta chỉ ra rằng 1 phụ thuộc bộ phận vào khóa sẽ dẫn đến 1 phụ thuộc bất cầu vào khóa. (*chứng minh phản chứng*).

Giả sử thuộc tính không khóa A thuộc R là phụ thuộc bộ phận vào khóa $K \subseteq R$, thì có một tập con $K' \subsetneq K$ mà :

$F \models K' \rightarrow A, K' \not\rightarrow K$ do K' không là một khóa trên R, $A \notin K$.

Vậy $K \rightarrow K', K' \not\rightarrow K, K' \rightarrow A$: A phụ thuộc bắc cầu vào K (đpcm)

Có thể chứng minh định nghĩa 1 tương đương với hai định nghĩa sau:

Định nghĩa 2: Một sơ đồ quan hệ $s=\{R,F\}$ là ở dạng chuẩn 3 (3NF) nếu :

- s là ở dạng 2NF (1)
- Mọi thuộc tính không khóa không phụ thuộc bất cầu vào khóa.

Định nghĩa 3: Một sơ đồ quan hệ $s=\{R,F\}$ là ở dạng chuẩn 3 (3NF) nếu :

- $\forall X \rightarrow Y \in F \Rightarrow$ hoặc X là siêu khóa, hoặc Y là thuộc tính khóa¹⁰.

VI.2 Phân rã thành dạng chuẩn 3 (3NF) :

Có thể phân rã một sơ đồ quan hệ $s=\{R,F\}$ thành các sơ đồ con thỏa 3NF mà không tổn thất thông tin. Thuật toán phân rã này do Bernstein đưa ra năm 1976.

Không mất tính tổng quát có thể giả sử F là 1 phủ tối thiểu, các thuộc tính của lược đồ quan hệ đều ít nhất có mặt trong một phụ thuộc hàm nào đó. (*Những thuộc tính Ai không tham gia phụ thuộc hàm nào có thể tách thành 1 lược đồ quan hệ $R_X(X,A_i)$ với X là khóa*).

Thuật toán phân rã thành dạng chuẩn 3 (3NF) :

Input : Lược đồ quan hệ R, tập phụ thuộc hàm F (F là phủ tối thiểu, R gồm các thuộc tính độc lập)

Output : $R = \rho(R_1 R_2, \dots, R_m)$, K là một khoá của $s = \langle R, F \rangle$
 R_i là 3NF, $i = 1, m$

¹⁰ Không mất tính tổng quát có thể giả sử trong F, mỗi phụ thuộc hàm về phải chỉ có 1 thuộc tính

ρ là phép tách không tổn thất thông tin và bảo toàn phụ thuộc hàm.

Method :

B1 : Nếu có 1 phụ thuộc hàm nào đó của F mà liên quan đến tất cả các thuộc tính của R thì kết quả chính là $R \Rightarrow$ dừng

B2 : $FD := F$;

Trong khi $FD \neq \emptyset$ thực hiện

– Nếu có $X \rightarrow A \in FD$ tìm tất cả các phụ thuộc hàm có vế trái là X (ví dụ : $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n$). Thì tạo thành lược đồ con : $R_x(X, A_1, A_2, \dots, A_n)$

– Loại bỏ các phụ thuộc hàm $X \rightarrow A_i, i = \overline{1, n}$ khỏi FD

$FD := FD \setminus \{ X \rightarrow A_i, i = \overline{1, n} \}$

B3 : Gọi R_F là tập các thuộc tính không tham gia các phụ thuộc hàm trong F , xây dựng lược đồ : $R_K(K, R_F)$ (K là một khoá của $S = \langle R, F \rangle$)

$\{R_{x1}(), R_{x2}(), \dots, R_{xm}(), R_K(K, R_F), K\}$ là phân rã R cần tìm.

Tính đúng của giải thuật :

- Do F là phủ cực tiểu nên không tồn tại các phụ thuộc hàm

$X \rightarrow A_i, X \rightarrow A_j, A_i \rightarrow A_j$, nên trong các $R_x()$ là 3NF

- Gọi R_F là tập các thuộc tính không tham gia các phụ thuộc hàm trong F , xây dựng lược đồ : $R_K(K, R_F)$ (K là một khoá tối thiểu của $S = \langle R, F \rangle$), $R_K(K, R_F)$ là 3NF

- Phép tách không tổn thất có thể chứng minh dễ dàng bằng phương pháp lập bảng (Chứng minh như bài tập).

- Theo cách xây dựng các thành phần phân rã, dễ thấy tập các phụ thuộc hình chiếu là chứa F , nên phép phân rã là bảo toàn tập phụ thuộc hàm.

Ví dụ 14 : Xét $R = ABSC$

$F = \{AB \rightarrow S, S \rightarrow B, S \rightarrow C\}$

Theo thuật toán

$R = \rho(R_1 = ABS, R_2 = SBC, R_3 = AB)$

VI.3 Chuẩn Boyce-Codd :

Trong thực tế có những CSDL đã thỏa mãn chuẩn 3 nhưng vẫn còn dư thừa dữ liệu do sự trùng lặp.

Ví dụ 15 : Xét quan hệ $r(R)$, với $R = ABS$ có tập phụ thuộc hàm

$F = \{AB \rightarrow S, S \rightarrow B\}$

A	B	S
a1	b1	s1
a2	b1	s1
a3	b2	s2
a3	b1	s1
a2	b2	s2

Một trong những nguyên nhân dẫn đến điều này là do lược đồ quan hệ có nhiều khóa, mỗi khóa có nhiều thuộc tính và có thuộc tính giao nhau. Để khắc phục điều này Boyce và Codd đã đưa ra dạng chuẩn BCNF (Boyce Codd normal Form) năm 1974.

Định nghĩa : Lược đồ quan hệ R với tập các phụ thuộc hàm được cho F gọi là dạng chuẩn Boyce_Codd (BCNF) nếu $\forall X \rightarrow A \in F, A \notin X$ thì X là 1 **siêu khóa** của R đối với F ($X^+ = R$).

Ví dụ 16: Dễ thấy với giả thiết ở ví dụ 15, R không là BCNF nhưng là 3NF.

Định lý : Nếu 1 lược đồ quan hệ R với tập phụ thuộc hàm F là BCNF thì nó là 3NF.

Chứng minh : Giả sử rằng lược đồ quan hệ R là ở BCNF nhưng không ở 3NF. Như vậy sẽ tồn tại 1 phụ thuộc hàm bắc cầu vào khóa $X \rightarrow Y, Y \rightarrow A$ trong đó X là khóa của R, $A \notin X$ hoặc $A \notin Y$ và $Y \rightarrow X \notin F^+$. Do đó Y không phải một siêu khóa của R. Nhưng $A \notin Y$, và $Y \rightarrow A$ là 1 phụ thuộc hàm trên R, nên Y phải là 1 siêu khóa.

VI.4 Phép tách lược đồ quan hệ thành BCNF (T.Son Fischer – 1982):

Input : Lược đồ quan hệ R, và tập phụ thuộc hàm F

Output : $R = \rho (R_1 R_2, \dots R_k)$

ρ : không mất mát thông tin

$\forall i = \overline{1, k}, R_i$ là BCNF

Method :

B1 : $\rho = R$

B2 : Nếu S là một lược đồ thuộc ρ , S chưa **BNCF**, chọn $X \rightarrow A$ là phụ thuộc hàm chứa các thuộc tính thuộc S, trong đó $A \notin X$, X không chứa khóa của S thì, tách S thành :

$S_1 = S - A$, và $S_2 = XA$

(Dễ thấy phép tách S thành S_1, S_2 là không mất mát thông tin, do định lý V. 3).

Quá trình tiếp tục cho đến khi tất cả là **BNCF**

Ví dụ 17 : Xét lược đồ quan hệ $R = CTHRS$

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

Thực hiện các phép tách sau:

$C \rightarrow T : (\underline{CT}, \text{CHRSRG})$

$HR \rightarrow C : (\underline{CT}, \underline{HRC}, \text{HRSRG})$

$HT \rightarrow R : (\underline{CT}, \underline{HRC}, \text{HRSRG})$

$CS \rightarrow G : (\underline{CT}, \underline{HRC}, \text{HRSRG})$

$HR \rightarrow R : (\underline{CT}, \underline{HRC}, \underline{HSR}, \underline{HSG})$

Chú ý : Việc sử dụng thuật toán trên, khi tiến hành còn phải phụ thuộc vào yêu cầu của bài toán, sự tách chỉ có ý nghĩa khi phần lớn các lược đồ quan hệ con có thể phục vụ cho một số yêu cầu nhất định (xem trong [7]: các vấn đề nảy sinh trong các phân rã BCNF tùy ý)

VII. Phụ thuộc đa trị (Multivalued Dependency) :

Khái niệm phụ thuộc hàm trong trường hợp tổng quát không thể vét hết các loại phụ thuộc hàm tồn tại trong quan hệ. Ví dụ mỗi cặp vợ chồng (tên cha, tên mẹ) thì không phải xác định duy nhất tên 1 đứa con, mà có thể có nhiều đứa con. Mỗi quan hệ đó gọi là phụ thuộc đa trị.

Định nghĩa: Gọi R là 1 lược đồ quan hệ. X và Y là 2 tập con của R, $Z = R - XY$. Một quan hệ r(R) thỏa phụ thuộc đa trị (MVD) $X \twoheadrightarrow Y$.

Nếu $\forall t_1, t_2 \in r, t_1(X) = t_2(X)$, thì tồn tại $t_3 \in r$ sao cho :

$$t_3(X) = t_1(X), t_3(Y) = t_1(Y), t_3(Z) = t_2(Z).$$

Do tính đối xứng của t_1, t_2 trong định nghĩa, có thể suy ra tồn tại 1 bộ t_4 trong r mà $t_4(X) = t_1(X)$, $t_4(Y) = t_2(Y)$ và $t_4(Z) = t_1(Z)$.

Ví dụ 18 : a. Xét 1 quan hệ :

SV	LOP	MASO	SOTHICH
t1	Tin38	TH101	Bơi
t2	Tin38	TH198	Võ
t3	Tin38	TH101	Võ
t4	Tin38	TH198	Bơi

X Y Z

Ta có : $X \twoheadrightarrow Y$, và $X \twoheadrightarrow Z$

Chú ý : Ta có thể chứng minh kết quả sau trực tiếp từ định nghĩa :

Nếu quan hệ r trên lược đồ R thỏa phụ thuộc đa trị $X \twoheadrightarrow Y$.

Nếu $X' \subseteq X$, thì $X \twoheadrightarrow YX'$

b. Xét 1 quan hệ r

Ta thấy r thỏa các phụ thuộc đa trị : $AB \twoheadrightarrow BC$, $AB \twoheadrightarrow C$.

Chú ý :

X và Y có thể là tập rỗng, nếu $Y = \emptyset$ thì $X \rightarrow\rightarrow \emptyset$ đúng mọi quan hệ.
 Nếu $X = \emptyset$ thì $\emptyset \rightarrow\rightarrow Y$ thỏa trên mọi quan hệ $r \Leftrightarrow$ tập giá trị trên tập thuộc tính còn lại trong quan hệ. Gọi $r[YZ]$ là 1 quan hệ, $Y \cap Z = \emptyset$.
 Khi đó $\emptyset \rightarrow\rightarrow Y$ thỏa trên $r \Leftrightarrow r = r[Y] \times r[Z]$.

Các tính chất của phụ thuộc đa trị :

Mối quan hệ giữa phụ thuộc đa trị và phân rã không tổn thất thông tin

Định lý : Cho r là 1 quan hệ trên lược đồ R , và X, Y, Z là các tập con của R , $Z = R - (XY)$. Quan hệ r thỏa phụ thuộc đa trị $X \rightarrow\rightarrow Y$ khi và chỉ khi r phân rã không tổn thất thông tin trên lược đồ quan hệ $R_1 = XY$ và $R_2 = XZ$.

Chứng minh : Ta có $X \rightarrow\rightarrow Y$.

Đặt $r_1 = \Pi_{R_1}(r)$, $r_2 = \Pi_{R_2}(r)$. Gọi t là một bộ thuộc $r_1 \bowtie r_2$ thì phải có một bộ $t_1 \in r_1$; và 1 bộ $t_2 \in r_2$ mà :

$$t(X) = t_1(X) = t_2(X),$$

$$t(X) = t_1(Y)$$

$$t(Z) = t_2(Z)$$

Do r_1 và r_2 là chiếu của r , nên phải tồn tại các bộ t_1' và t_2' thuộc r sao cho :
 $t_1(XY) = t_1'(XY)$ và $t_2(XZ) = t_2'(XZ)$. Phụ thuộc đa trị $X \rightarrow\rightarrow Y$ dẫn đến t phải thuộc r , và r phải chứa bộ t_3 , với $t_3(X) = t_1'(X)$, $t_3(Y) = t_1'(Y)$ và $t_3(Z) = t_2'(Z)$ là một biểu diễn của t .

Ngược lại, giả sử r là tách không mất thông tin trên R_1 và R_2 . Đặt t_1 và t_2 là các bộ trên r sao cho : $t_1(X) = t_2(X)$, gọi r_1, r_2 lần lượt là tách của r trên R_1, R_2 . Quan hệ r chứa một bộ $t_1' = t_1(XY)$ và quan hệ $t_2' = t_2(XZ)$. Vì $r = r_1 \bowtie r_2$, nên r phải chứa 1 bộ t sao cho $t[XY] = t_1[XY]$ và $t[XZ] = t_2[XZ]$. Bộ t là kết quả của phép nối t_1' và t_2' . Vậy có 3 bộ $t, t_1, t_2 \in r$ thỏa định nghĩa phụ thuộc đa trị: $X \rightarrow\rightarrow Y$.

Chú ý : Định lý trên cho chúng ta 1 phương pháp để kiểm tra 1 quan hệ $r(R)$ thỏa phụ thuộc đa trị $X \rightarrow\rightarrow Y$?, bằng cách thực hiện chiếu r trên XY và $X(R - XY)$, kết nối 2 hình chiếu này và kiểm tra xem kết quả bằng r ?.

- X và Y có thể là tập rỗng, nếu $Y = \emptyset$ thì $X \rightarrow\rightarrow Y$ đúng với mọi quan hệ.
- Nếu $X = \emptyset$ thì $\emptyset \rightarrow\rightarrow Y$ thỏa trên mọi quan hệ $r \Leftrightarrow$ tập giá trị trên tập thuộc tính Y là độc lập các giá trị của những thuộc tính còn lại trong quan hệ, có nghĩa là : nếu gọi $r[YZ]$ là 1 quan hệ, $Y \cap Z = \emptyset$. Khi đó $\emptyset \rightarrow\rightarrow Y$ thỏa trên $r \Leftrightarrow r = r[Y] \times r[X]$.

Phụ thuộc đa trị và phụ thuộc hàm :

Hệ quả : Cho r là 1 quan hệ trên lược đồ R và X, Y là các tập con của R . Nếu r thỏa phụ thuộc hàm $X \rightarrow Y$, thì r thỏa phụ thuộc đa trị $X \rightarrow\rightarrow Y$.

Chứng minh : Từ kết quả: Nếu $\rho(R_1, R_2)$ là phép tách của R và F là tập phụ thuộc hàm trên R, ρ không mất mát thông tin $\Leftrightarrow R_1 \cap R_2 \rightarrow R_1 \setminus R_2$ hoặc $R_1 \cap R_2 \rightarrow R_2 \setminus R_1$

Ta có : $X \rightarrow Y$ suy ra r kết nối không tổn thất thông tin trên XY và $X(R \setminus (XY))$ kết quả này suy ra $X \rightarrow Y$ theo định lý đã có.

Định lý : Cho F là tập phụ thuộc hàm trên R. Gọi X, Y và Z là các tập con của R, với $Z = R - (XY)$. Đặt X^+ là bao đóng của X trên F .

Nếu $Y \not\subseteq X^+$ và $Z \not\subseteq X^+$ thì tồn tại 1 quan hệ r(R) thỏa F nhưng không thỏa phụ thuộc đa trị $X \twoheadrightarrow Y$.

Chứng minh :

Vì $Y \not\subseteq X^+ \Rightarrow X \rightarrow Y \notin F^+$ và $Z \not\subseteq X^+ \Rightarrow X \rightarrow Z \notin F^+$
 $\Rightarrow X \not\rightarrow XY \setminus XZ = Y$, $X \not\rightarrow XZ \rightarrow XY = Z$
 $\Rightarrow XY \cap XZ \not\rightarrow XY \setminus XZ$, $XY \cap XZ \not\rightarrow XZ \rightarrow XY$
 $\Rightarrow R = \rho(XY, XZ)$ không bảo toàn $\Rightarrow X \not\rightarrow Y$ (đpcm)

Ví dụ 19 : Cho $R = ABCDEI$, và $F = \{A \rightarrow BC, C \rightarrow D\}$, thì $F \models A \twoheadrightarrow BCD$ và $A \twoheadrightarrow C$, nhưng $F \not\models A \twoheadrightarrow DE$.

Hệ tiên đề cho phụ thuộc hàm và phụ thuộc đa trị :

Hệ tiên đề này được Beeri, Fagin và Howard đề xuất vào năm 1977. Tuy vậy tập hệ tiên đề này vẫn được gọi là hệ tiên đề Armstrong .

Cho r là 1 quan hệ trên R và W, X, Y, Z là các tập con của R.

M₁ : (Tính phản xạ) Nếu $Y \subseteq X \subseteq R$ thì $X \rightarrow Y$

M₂ : (Tính tăng trưởng) Nếu $X \rightarrow Y$ và $Z \subseteq R \Rightarrow XZ \rightarrow YZ$

M₃ : (Bắc cầu) Nếu $X \rightarrow Y, Y \rightarrow Z$ thì $X \rightarrow Z$

Các luật cho phụ thuộc đa trị :

M₄ : (Luật bù) $X \twoheadrightarrow Y$ thì $X \twoheadrightarrow R - X - Y$

M₅ : (Luật tăng trưởng) Nếu $X \twoheadrightarrow Y$ và $V \subseteq W$ thì $WX \twoheadrightarrow VY$

M₆ : (Luật bắc cầu) Nếu $X \twoheadrightarrow Y$ và $Y \twoheadrightarrow Z$ thì $X \twoheadrightarrow Z - Y$

Các luật phối hợp giữa FD và MVD

M₇ : Nếu $X \rightarrow Y$ thì $X \twoheadrightarrow Y$

M₈ : Nếu $X \twoheadrightarrow Y, Z \subseteq Y$ và W là tập con phân biệt với Y (tức là $W \cap Y = \emptyset$) và $W \rightarrow Z$ thì $X \rightarrow Z$.

Định lý : (Beeri, Fagin, Howard [1997])

Hệ tiên đề từ $M_1 \div M_8$ là đúng & đầy đủ cho các tập phụ thuộc hàm và phụ thuộc đa trị. Nghĩa là nếu D là tập phụ thuộc hàm và phụ thuộc đa trị, một phụ thuộc hàm hay một phụ thuộc đa trị được suy diễn logic từ D thì cũng thỏa D^+ . (Để thấy $D \models D^+$)

Các luật suy diễn bổ sung cho phụ thuộc đa trị :

Đối với các phụ thuộc đa trị có thêm một số luật suy dẫn :

M_9 : (Luật hợp)

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

M_{10} : (Luật bắc cầu)

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

M_{11} : (Luật tựa bắc cầu)

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

M_{12} : (Luật tách)

Nếu $X \twoheadrightarrow Y$ và $X \twoheadrightarrow Z$ thì $X \twoheadrightarrow Y \cap Z, X \twoheadrightarrow Y - Z, X \twoheadrightarrow Z - Y$

Nhận xét :

- Chúng ta cần chú ý qui tắc phân rã của phụ thuộc đa trị không mạnh bằng quy tắc tương ứng của phụ thuộc hàm. Qui tắc của phụ thuộc hàm cho chúng ta suy ra trực tiếp từ $X \rightarrow Y$ rằng $X \rightarrow A$ đúng với mọi phụ thuộc A trong Y .
- Quy tắc của phụ thuộc đa trị chỉ cho phép ta kết luận $X \twoheadrightarrow A$ từ $X \twoheadrightarrow Y$ nếu chúng ta tìm được Z sao cho $X \twoheadrightarrow Z$ và hoặc $Z \cap Y = A$ hoặc $Y - Z = A$.
- Cũng từ tập các phụ thuộc đa trị chỉ suy dẫn được phụ thuộc hàm tầm thường tương ứng ($X \rightarrow Y, Y \subseteq X$)

Tập cơ sở phân rã tối thiểu

Định nghĩa : Cho một tập $S = \{S_1, S_2, \dots, S_p\}$, ở đây $U = S_1 \cup S_2 \cup \dots \cup S_p$, tập cơ sở phân rã tối thiểu của S (*minimal disjoint set basis of S*) kí hiệu **mdsb** (S) là các phân bộ (*partition*) của U là $\{T_1, T_2, \dots, T_k\}$ mà:

1. $\forall i, S_i$ là hợp của một số T_j nào đó
2. Không có phân bộ nào của U có ít phần tử hơn có tính chất 1.

Chú ý : **mdsb**(S) là duy nhất.

Ví dụ 20 : Xét $S = \{ABCD, CDE, AE\}$. Chúng ta có $U = ABCDE$ và **mdsb** (S) = $\{A, B, CD, E\}$.

Nhận xét : -Cho F là tập các phụ thuộc đa trị trên R , và $X \subseteq R$. Định nghĩa G như sau :

$$G = \{ Y \mid F \models X \twoheadrightarrow Y \}$$

Chúng ta thấy rằng **mdsb** (G) là tập con của G . Nếu có một tập Y_1 thuộc G mà Y_1 chứa các thuộc tính sao cho có một số thuộc tính Y_2 , và một số thuộc tính khác không thuộc $Y_2 (\subseteq G)$, thì theo luật M_{12} , có $Y_3 = Y_1 - Y_2$, và $Y_4 = Y_1 \cap Y_2$ thuộc G . **Mdsb**(G) chứa các tập không rỗng này nếu chúng không chứa 1 tập con thực sự khác của G . Chú ý rằng nếu $X = A_1 A_2 \dots A_n$ thì A_1, A_2, \dots, A_n thuộc **mdsb**(G)

Định nghĩa : Cho F, X, G được định nghĩa như trên. Cơ sở phụ thuộc (*dependency basis*) của X với tập phụ thuộc đa trị đã cho F là **mdsb**(G) được

kí hiệu là $DEP(X)$. Nếu $X = A_1A_2...A_n$ và $DEP(X) = \{ A_1, A_2,..., A_n, Y_1, Y_2, ..., Y_m \}$, Chúng ta viết $X \rightarrow\rightarrow Y_1 | Y_2 | ... | Y_m$

Ví dụ 21 : Cho $F = \{ A \rightarrow\rightarrow BC, DE \rightarrow\rightarrow C \}$ là 1 tập phụ thuộc đa trị trên ABCDE. Nếu $X = A$, thì $G = \{ A, BC, DE, C, BDE, B, BCDE, CDE \}$ và $DEP(A) = mdsb(G) = \{ A, B, C, DE \}$

Nhận xét :

Chúng ta có thể phát hiện tất cả các phụ thuộc đa trị suy dẫn từ F với X là vế trái từ $DEP(X)$ với nhận xét :

$F \models X \rightarrow\rightarrow Y \Leftrightarrow Y$ là hợp một số tập phần tử thuộc $DEP(X)$

Theo M_9 , Nếu $Y_1, Y_2, ..., Y_k \subseteq DEP(X)$, thì $F \models X \rightarrow\rightarrow Y_1 Y_2 ... Y_k$

Thuật toán thành viên (*Membership algorithm*)

Input : Cho F là tập các phụ thuộc đa trị trên lược đồ quan hệ R

Output : $F \models X \rightarrow\rightarrow Y$?

Method:

B1 : Tìm $DEP(X)$ từ tập phụ thuộc đa trị F

B2 : Nếu $Y = \cup_{A_i \in DEP(X)} A_i$ thì $F \models X \rightarrow\rightarrow Y$

Ngược lại $F \not\models X \rightarrow\rightarrow Y$.

Vậy điểm mấu chốt của bài toán thành viên của tập phụ thuộc đa trị là thuật toán xác định $DEP(X)$.

Thuật toán : Tìm cơ sở phụ thuộc của tập thuộc tính X

Input : Cho lược đồ quan hệ R , F là tập phụ thuộc đa trị xác định trên R

X là tập thuộc tính, $X \subseteq R$

Output : $DEP(X)$

Method :

B0 : $G = \emptyset$

B1 : Tìm G là tập tất cả Y mà phụ thuộc đa trị $X \rightarrow\rightarrow Y$ được suy bởi F :

- Nếu có $X' \rightarrow\rightarrow Y' \in F, X' \subseteq X \Rightarrow G = G \cup \{Y', R \setminus (X'Y')\}$ (do M_4)
- Nếu $A \in X \Rightarrow G = G \cup \{A\}$ (do M_1)
- $X \rightarrow\rightarrow Y, Y \rightarrow\rightarrow Z \Rightarrow X \rightarrow Z - Y$ (do M_6)

B2 : Đặt $DEP(X) = G$

B3 : Lặp lại quy trình tinh chế G sao cho các phần tử của G là các tập rời nhau :

Nếu $Y_1, Y_2 \in G, Y_1 \cap Y_2 \neq \emptyset$ thay Y_1, Y_2 bởi $Y_1 - Y_2, Y_2 - Y_1, Y_1 \cap Y_2$

B4 : Lặp đến khi nào không còn biến đổi được G theo quy tắc :

Với mỗi phụ thuộc đa trị : $V \rightarrow\rightarrow W \in F$, và $Y \subseteq G$ sao cho $Y \cap W \neq \emptyset$, thay Y bởi $Y \cap W$ và $Y \setminus W$ trong G

B5 : $DEP(X) = G = mdsb(G)$.

Kết nối không mất mát thông tin đối với tập phụ thuộc hàm phụ thuộc đa trị :

Thuật toán kiểm tra phép tách 1 lược đồ quan hệ R thành R_1, \dots, R_n theo tập phụ thuộc hàm và phụ thuộc đa trị F

Input : Cho R lược đồ quan hệ, F tập phụ thuộc hàm & phụ thuộc đa trị, $\rho(R_1, R_2, \dots, R_n)$

Output : Khẳng định $\rho(R_1, R_2, \dots, R_n)$ tổn thất thông tin hay không

Method :

B1 : Thiết lập bảng như thuật toán kiểm tra phép phân rã bảo toàn thông tin

B2 : - Nếu là FD (phụ thuộc hàm) thao tác theo thuật toán như thuật toán kiểm tra phép phân rã bảo toàn thông tin của phụ thuộc hàm. Nếu có MVD (phụ thuộc đa trị) : $X \twoheadrightarrow Y$ có 2 hàng t_1, t_2 mà $t_1(X) = t_2(X)$ thì thêm 1 hàng u với $u[X] = t_1[X] = t_2[X]$, $u[Y] = t_1[Y]$ và $u[R - XY] = t_2[R - XY]$ (Nếu u chưa có trong bảng)

B3 : Thực hiện phép biến đổi dựa theo b2 cho đến lúc nào không còn biến đổi được nữa.

Nếu bảng có 1 giá trị toàn a thì ρ không mất mát thông tin, ngược lại kết nối là mất mát thông tin.

Định lý : Cho R là 1 lược đồ quan hệ, $\rho = (R_1, R_2)$ là một phép tách của R , F là tập các phụ thuộc hàm và phụ thuộc đa trị trên R , ρ là phép tách không mất mát thông tin khi và chỉ khi :

$$R_1 \cap R_2 \twoheadrightarrow R_1 - R_2 \text{ hoặc } R_1 \cap R_2 \twoheadrightarrow R_2 - R_1.$$

Chứng minh : ρ là phép tách không mất mát thông tin \Leftrightarrow với r thỏa F và 2 bộ $t, s \in r$, $\exists u \in r$ sao cho $u[R_1] = t[R_1]$ và $u[R_2] = s[R_2]$. Nhưng $\exists u \Leftrightarrow t[R_1 \cap R_2] = s[R_1 \cap R_2] \Leftrightarrow R_1 \cap R_2 \twoheadrightarrow R_1 - R_2 \text{ hoặc } R_1 \cap R_2 \twoheadrightarrow R_2 - R_1$.

VIII. Dạng chuẩn bốn (Fourth Normal Form- 4NF)

Định nghĩa : Gọi R là 1 lược đồ quan hệ, F là tập hợp các phụ thuộc hàm & phụ thuộc đa trị đối với R . R gọi là thuộc dạng 4NF nếu mỗi phụ thuộc đa trị $X \twoheadrightarrow Y \in F^+$ trong đó $Y \not\subset X$ và XY không chứa tất cả thuộc tính của R thì X chứa 1 khóa của R .

Định lý : Nếu 1 lược đồ quan hệ R thỏa 4NF với tập phụ thuộc hàm và phụ thuộc đa trị F , thì R thỏa BCNF với tập phụ thuộc suy dẫn từ F .

Chứng minh : Giả sử R không thuộc BCNF : $X \rightarrow A$, $A \notin X$, X không là siêu khóa; Nếu $XA = R \Rightarrow X$ chứa 1 khóa $\Rightarrow X$ siêu khóa, Vậy $XA \neq R$.

Do $X \rightarrow A$, $A \notin X$, $XA \neq R \Rightarrow X \twoheadrightarrow A$, $A \notin X$, $XA \neq R \Rightarrow X$ siêu khóa mâu thuẫn 4NF. (đpcm)



Bài tập

1. Khảo sát tính đúng của các luật suy dẫn trên quan hệ $r(R)$ với W, X, Y , và Z là các tập con của R :

- $(X \rightarrow Y \text{ và } Z \rightarrow W) \Rightarrow XZ \rightarrow YW$
- $(ZX \rightarrow Y \text{ và } Z \rightarrow X) \Rightarrow (Z \rightarrow Y)$
- $(X \rightarrow Y) \text{ và } (Y \rightarrow Z) \Rightarrow (X \rightarrow YZ)$
- $(X \rightarrow Y, W \rightarrow Z \text{ và } Y \supseteq W) \Rightarrow (X \rightarrow Z)$

2. Chứng minh với mọi lược đồ quan hệ R , và tập phụ thuộc hàm F xác định trên R , $\forall X, Y \subseteq R$ thì:

$$(XY)^+ = (X^+Y)^+ = (XY^+)^+$$

3. Chứng minh với mọi lược đồ quan hệ R , và tập phụ thuộc hàm F xác định trên R , và K là một khoá thì:

$$Z^+ \cap (KVZ) = \emptyset, \forall Z \subseteq K$$

4. Cho lược đồ quan hệ R và tập phụ thuộc hàm:

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

Chứng minh : Nếu quan hệ r xác định trên R thoả F thì r cũng thoả : $AB \rightarrow GH$

5. Cho lược đồ quan hệ R và tập phụ thuộc hàm:

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

Chứng minh nếu r thoả F thì r cũng thoả : $AB \rightarrow E$ và $AB \rightarrow G$.

6. Cho lược đồ quan hệ R và tập phụ thuộc hàm F xác định trên R :

$$F = \{A \rightarrow D, AB \rightarrow DE, CE \rightarrow G, E \rightarrow H\}$$

Tính $(AB)^+$.

7. Cho quan hệ r dưới đây:

R	A	B	C	D	E
	a_1	b_1	c_1	d_1	e_1
r	a_1	b_2	c_2	d_2	e_1
	a_2	b_1	c_3	d_2	e_1
	a_2	b_2	c_4	d_3	e_1
	a_3	b_2	c_5	d_1	e_1

Những phụ thuộc hàm nào sau đây thoả r :

$$A \rightarrow D, AB \rightarrow D, C \rightarrow BDE, E \rightarrow A, A \rightarrow E ?$$

8. Gọi $F = \{ AB \rightarrow C, A \rightarrow B \}$

- Tìm 1 phủ cực tiểu của F
- Khi câu (a) được cho trong một kỳ thi học kỳ tại 1 trường đại học, hơn 1 nửa số sinh viên trả lời :

$$G = \{A \rightarrow B, B \rightarrow C\}$$

Chứng tỏ câu trả lời này sai bằng cách đưa ra 1 quan hệ thoả F nhưng vi phạm G .

9. Tìm : 1. Phủ cực tiểu của các tập phụ thuộc hàm cho ở bài tập 5, 6, 7.

2. Khoá của các lược đồ quan hệ R đã cho ở bài tập 5, 6, 7.

10. Tìm khoá của quan hệ cho ở bài tập 8.

11. Cho lược đồ quan hệ $R = ABCDEFG$, trên đó có xác định một tập phụ thuộc hàm \mathcal{F} xác định như sau:

$\mathcal{F} = \{ AB \rightarrow FD, FG \rightarrow AC, BC \rightarrow FE, BD \rightarrow GA, DF \rightarrow ABG, G \rightarrow AD \}$

1. Chứng minh : $FBG \rightarrow BC \in \mathcal{F}^+$

2. Tìm một khóa của lược đồ R

3. Hãy kiểm tra phân rã ρ sau của lược đồ quan hệ R có bảo toàn thông tin không ?

$$\rho(R) = \{ R_1, R_2, R_3, R_4 \}$$

Với : $R_1 = ACD, R_2 = BDF, R_3 = GAE, R_4 = DFC$

4. Tìm 1 phủ cực tiểu của \mathcal{F} .

5. Tách lược đồ quan hệ R thành các lược đồ con 3NF bảo toàn thông tin.

6. Tách lược đồ quan hệ R thành các lược đồ con BCNF bảo toàn thông tin

12. Cho đồ quan hệ $R=ABCDEF$ và tập phụ thuộc hàm F xác định trên R :

$F = \{ AC \rightarrow EF, F \rightarrow ED, BE \rightarrow AF, D \rightarrow AE \}$

a. Chứng minh rằng : $BD \rightarrow AF \in F^+$

b. Tìm phủ cực tiểu của F .

c. Dùng thuật toán đã biết tìm 1 khóa của R .

d. Hãy kiểm tra phân rã ρ sau của lược đồ quan hệ R có bảo toàn thông tin không ?

$$\rho(R) = \{ R_1, R_2, R_3, R_4 \}$$

Với : $R_1 = ACD, R_2 = BDF, R_3 = AE, R_4 = DFC$.

e. Tách lược đồ quan hệ R thành các lược đồ con 3NF bảo toàn thông tin.

f. Tách lược đồ quan hệ R thành các lược đồ con BCNF bảo toàn thông tin

13. Cho lược đồ quan hệ R , và tập phụ thuộc hàm xác định trên R . Chứng minh thuật toán tìm phủ tối thiểu sau là **không chính xác**:

- B1: Tách tất cả các phụ thuộc hàm của F thành tập phụ thuộc hàm mà vế phải chỉ 1 thuộc tính. (Ví dụ $AB \rightarrow CD$ được tách thành $AB \rightarrow C, AB \rightarrow D$)

- B2: Loại bỏ những phụ thuộc không đầy đủ :

Loại 1 : Phụ thuộc hàm có vế phải là tập con của vế trái (ví dụ loại $AB \rightarrow B$)

Loại 2: Hai phụ thuộc hàm có vế phải giống nhau, nếu vế trái của phụ thuộc hàm này chứa vế trái của phụ thuộc hàm kia thì loại ra khỏi F (ví dụ có $ABC \rightarrow D, BC \rightarrow D$ thì loại $ABC \rightarrow D$ khỏi F)

- B3: Loại bỏ các phụ thuộc hàm dư thừa:

Nếu $X \rightarrow A$ mà $A \in X \rightarrow F \setminus \{X \rightarrow A\}$

14. Hãy đưa ra một phản ví dụ để chỉ ra rằng thuật toán tìm khoá sau đây là **không chính xác**:

Xem mỗi thuộc tính thuộc lược đồ quan hệ là đỉnh của đồ thị định hướng. Một cung đi từ đỉnh X đến đỉnh Y , nếu $\exists A \rightarrow B \in F$ mà $X \in A, Y \in B$. Gọi :

- Đỉnh gốc là đỉnh xuất phát của 1 cung (*không có cung đến*)

- Đỉnh ngọn là đỉnh đến của 1 cung (*không có cung đi*)

- Đỉnh trung gian là đỉnh vừa cung đi, và cung đến.

Khoá của một lược đồ quan hệ R , ứng với tập phụ thuộc hàm F được xác định:

Các đỉnh gốc là các thuộc tính khoá.

Các đỉnh ngọn không là thuộc tính khoá.

Nếu K là tập các đỉnh gốc:

Nếu $K_F^+ = R$ thì K là khoá

Ngược lại ta thêm lần lượt vào K một số thuộc tính là đỉnh trung gian cho đến khi $K_F^+ = R$.

15. Dựa vào định lý của Lucchesi & Osborn viết thuật toán tìm tất cả các khoá của 1 sơ đồ quan hệ $S = \langle R, F \rangle$ cho trước. Đánh giá độ phức tạp của giải thuật trên.

16. Hãy đề xuất thuật toán kiểm tra xem một lược đồ quan hệ 2 NF ?

17. Hãy đề xuất thuật toán kiểm tra xem một lược đồ quan hệ 3 NF ?

18*. Giả sử $s = \langle R, F \rangle$ là sơ đồ quan hệ, K là họ khóa của W . Đặt :

$$M = \{A - a \mid a \in A, A \in K\}$$

$$F_n = \{B \mid B \text{ thuộc tính thứ cấp}\}$$

$$L = \{C+ \mid C \in M\}$$

Chứng minh 3 mệnh đề sau tương đương:

1. s là 2NF

$$2. \forall B \in L \Rightarrow B \cap F_n = \emptyset$$

$$3. \forall B \in L \text{ và } a \in F_n \Rightarrow (B - a)^+ = B - a$$

19* Chứng minh nếu K là 1 khóa bất kỳ của $s = \langle R, F \rangle$ thì $Z^+ \cap (K - Z) = \emptyset, \forall Z \subseteq K$

20* Cho $s = \langle R, F \rangle, F_n = \{B \mid B \text{ thuộc tính thứ cấp}\}$. Chứng minh: s ở 3NF khi và chỉ khi

$$\forall B \in K^{-1}, a \in F_n \text{ thì } (B - a)^+ = B - a$$

21* Cho $s = \langle R, F \rangle, F_n = \{B \mid B \text{ thuộc tính thứ cấp}\}, K$ là tập khóa. Đặt $G = \{B - F_n \mid B \in K^{-1}\}$

Chứng minh s ở 2NF khi và chỉ khi $\forall C \in G : C^+ = C$.

22. Xét xem sơ đồ quan hệ sau thuộc dạng chuẩn nào: 3NF? 2NF?

$$s = \langle R, F \rangle, R = ABCDEGHIM$$

$$F = \{CB \rightarrow GH, DE \rightarrow IMH, CI \rightarrow CBDH, H \rightarrow I\}$$

23. a. Khẳng định tính đúng hoặc bác bỏ mệnh đề:

Nếu hai lược đồ quan hệ $p = (U, F)$ và $q = (U, G)$ có cùng tập khóa thì chúng có cùng dạng chuẩn.

b. Hỏi dạng chuẩn cao nhất của lược đồ sau:

$$s = (U, F) \text{ với } U = ABCD; F = \{A \rightarrow B, B \rightarrow C, C \rightarrow A, B \rightarrow D\}$$

Tìm tất cả các khóa của lược đồ quan hệ

24. Cho lược đồ quan hệ $R = CIDBKFGML$, trên đó có xác định 1 tập phụ thuộc hàm

$PTH = \{C \rightarrow IDBK, D \rightarrow B, K \rightarrow F\}$. Xét xem với giả thiết đã cho lược đồ quan hệ R có dạng chuẩn cao nhất là dạng chuẩn nào ? Vì sao?

25. Cho lược đồ quan hệ $s = (U, F); U = ABCDE$, tập phụ thuộc hàm

$$F = \{ABC \rightarrow E, ABC \rightarrow E, C \rightarrow D, D \rightarrow C, CD \rightarrow E\};$$

a. Tìm tất cả các khóa của s ?

b. Cho biết dạng chuẩn cao nhất của lược đồ quan hệ s ?

c. Có thể loại bỏ một phụ thuộc hàm của F để s là 2NF?

26. Cho lược đồ quan hệ $s = (U, F);$ Với $U = ABCDEGH$, tập phụ thuộc hàm

$$F = \{AB \rightarrow CD, CE \rightarrow BH, DGH \rightarrow AC, G \rightarrow A, G \rightarrow B\}$$

a. Tìm 1 phủ tối thiểu của F .

b. Tìm 1 phép phân rã U thành các lược đồ con 3NF bảo toàn thông tin.

c. Kiểm tra phép phân rã $U = \rho(ABC, DEG, AEGH, CD)$ có bảo toàn thông tin?

27. Cho lược đồ quan hệ $s = (U, F); F = \{L_i \rightarrow R_i \mid L_i \cap R_i \neq \emptyset, i = 1..n\}$; Kí hiệu:

$$\mathcal{L} = \cup L_i, i = 1..n; \quad \mathcal{R} = \cup R_i, i = 1..n$$

Khẳng định hay bác bỏ các mệnh đề sau:

1. Nếu $\emptyset \neq (U \setminus \mathcal{R})^+ \neq U$ và $|\mathcal{L} \cap \mathcal{R}| = 2$ thì lược đồ quan hệ s chỉ có 2 khóa.

2. Nếu s ở 3NF và $\mathcal{R} \setminus \mathcal{L} = \emptyset$ thì s ở BCNF.

28. Cho lược đồ quan hệ R trên tập thuộc tính $U = \{ABCDE\}$

Chứng minh rằng nếu quan hệ r xác định trên U thỏa các phụ thuộc đa trị:

$$F = \{A \twoheadrightarrow BC, DE \twoheadrightarrow C\} \text{ thì } r \text{ cũng thỏa } AD \twoheadrightarrow BE$$

29. Chứng minh tính đúng của các luật sau đây:

a. $X \twoheadrightarrow X$ cho mọi X

- b. Nếu $X \twoheadrightarrow Y$ thì $XZ \twoheadrightarrow Y$ với $Z \subseteq U$.
30. Tìm 1 ví dụ để chỉ ra rằng : $X \twoheadrightarrow Y \not\Rightarrow X \rightarrow Y$
31. Cho $X, Y, Z, W \subseteq U$
 Chứng minh rằng nếu quan hệ r trên $R(U)$ thoả các phụ thuộc đa trị
 $X \twoheadrightarrow Y, Z \subseteq W$ thì r cũng thoả $XW \twoheadrightarrow YZ$.
32. Cho lược đồ $R(ABCDEI)$ và $F = \{A \twoheadrightarrow BCD, B \twoheadrightarrow AC, C \rightarrow D\}$. Tìm dạng chuẩn 4.
33. Cho lược đồ $R(BOISQD)$ và tập phụ thuộc hàm
 $F = \{S \twoheadrightarrow D, I \rightarrow B, IS \rightarrow Q, B \rightarrow Q\}$
 Tìm dạng chuẩn 4 qua phép tách.
34. Cho lược đồ $R(ABCDEI)$ và tập phụ thuộc hàm:
 $F = \{A \twoheadrightarrow BCD, C \rightarrow D, B \twoheadrightarrow AC\}$
 Tìm dạng chuẩn 4.
35. Chứng minh rằng nếu quan hệ $r(R)$ thoả các phụ thuộc đa trị:
 $X \twoheadrightarrow Y_1, X \twoheadrightarrow Y_2, \dots, X \twoheadrightarrow Y_k$
 trong đó $R = XY_1 \dots Y_k$ thì R có thể tách thành các lược đồ:
 XY_1, XY_2, \dots, XY_k không mất mát thông tin.



Chương IV

Ngôn ngữ SQL-Tối ưu hóa câu hỏi

I. NGÔN NGỮ SQL

SQL viết tắt của Structured Query language là ngôn ngữ hỏi của công ty IBM SanJoe năm 1981. Phiên bản gốc của ngôn ngữ này là ngôn ngữ SEQUEL được giới thiệu năm 1976.

SQL là 1 ngôn ngữ phi thủ tục, chuẩn mực và điển hình. Để xây dựng các ứng dụng phức tạp, SQL được nhúng vào các ngôn ngữ thế hệ thứ 3 như : COBOL, PL/1. Hiện nay rất nhiều sản phẩm phần mềm thương mại có cài đặt SQL như Oracle, Ingres, DB2,... so sánh với các ngôn ngữ thế hệ thứ 4, SQL chưa có chức năng đồ họa và phương tiện hỗ trợ quyết định.

Tên gọi SQL = Structured Query Language, nghĩa là ngôn ngữ truy vấn (thông tin) có cấu trúc.

Tuy nhiên, không chỉ có Query mà SQL còn thực hiện nhiều chức năng khác như bổ sung thêm dữ liệu, sửa đổi cập nhật thông tin, hủy thông tin, liên kết lưu trữ thông tin.

Vậy phải gọi là :

Structured Add, Modify, Delete, Join, Store, Trigger & Query Language.

Khả năng của SQL gồm :

1. Tạo dữ liệu, hỏi dữ liệu, cập nhật dữ liệu
2. Sắp xếp dữ liệu
3. Trình bày dữ liệu

Một số khả năng, tiện ích về cơ sở dữ liệu quan hệ

- Tạo sinh báo cáo
- Chuyển đổi ngôn ngữ
- Chức năng trợ giúp

Chú ý : Các thuật ngữ trong CSDL quan hệ như quan hệ, thuộc tính, bộ ... trong phần trình bày sau được thay thế bằng các thuật ngữ như bảng (table), cột (column), bản ghi (record) hoặc hàng (row) nhằm trực quan.

SQL trong chương trình ứng dụng.

Ba kiểu tương tác với ứng dụng

- Moduln : các thủ tục làm sẵn, thông qua các tham số, gọi và trả kết quả về.
- Nhúng trực tiếp : các câu lệnh SQL trong chương trình.
- Gọi trực tiếp : tương tác.

Nhúng là hình thức làm việc tĩnh – static : câu lệnh được biên dịch ra, không thay đổi được khi chạy.

Hiện nay Dynamic SQL : cho phép chuẩn bị câu lệnh SQL trong khi chạy. Các thành phần của câu lệnh SQL được ghép nối lại trong thời gian chạy, cho phép mức linh hoạt rất cao trong khai thác CSDL.

Gọi trực tiếp quá trình xử lý tương tự như câu lệnh DOS.

IV.1 Câu lệnh tạo dữ liệu

a. Tạo bảng

Cú pháp : **CREATE TABLE** <tên quan hệ>(<tên thuộc tính> <kiểu dữ liệu> [not null][, <tên thuộc tính> <kiểu dữ liệu> [not null][,...]])

Trong đó :

- **Tên quan hệ** : là 1 chuỗi ký tự bất kỳ không có ký hiệu trống, không trùng với các từ khóa
- **Tên thuộc tính** : là chuỗi ký tự bất kỳ không có ký hiệu trống.(thứ tự của các thuộc tính không là quan trọng).
- **Kiểu dữ liệu** : mô tả kiểu dữ liệu của thuộc tính, gồm có:

Số nguyên:

Integer:(-2147483648->2147483647)

Smallinteger : (-32768 -> 32767)

Số thực :

Decimal(n,p): số thập phân có độ dài tối đa là n ký số có p chữ số thập phân (không tính dấu chấm thập phân).

Float : số thực dấu chấm động

Ký tự :

Char(n) : chuỗi ký tự có độ dài cố định n, n<=255

Varchar(n) : chuỗi ký tự có độ dài biến đổi từ 0 đến n.

Longvarchar : xâu ký tự có độ dài không cố định

(độ dài này từ 4Kbyte đến 32 Kbytes)

Ngày tháng :

Date : dữ liệu ngày tháng (yy-mm-dd)

Trong mô tả các thuộc tính trên, NOT NULL được dùng với nghĩa thuộc tính có mô tả này nhất thiết phải có dữ liệu.

Tác dụng : Tạo các quan hệ (bảng) theo cấu trúc được chỉ ra.

Ví dụ 1:

1. Create Table CONGTY(S# char(5) not null, tên char(25) not null, Status integer, CITY char(40))
2. Create Table NHANVIEN(NA# char(4) not null, Hoten Varchar(30), Ngsinh date, baclg decimal(5,2))
3. Xét 3 lược đồ quan hệ

NHACUNGCAP(Mã số, tên, tình trạng, địa chỉ)

MATHANG(số hiệu, tên mặt hàng, màu sắc, trọng lượng, thành phố sản xuất mặt hàng)

MATHANGCCAP(Mã số nhà cung cấp, số hiệu mặt hàng, số lượng hàng gửi đi, ngày gửi, đơn giá)

Các bảng tương ứng được tạo lập :

CREATE TABLE S(S# integer not null, Sname varchar(8) not null, Status integer not null, city varchar(30) not null)

CREATE TABLE P(P# integer not null, Pname varchar(10) not null, weight integer not null, Colour varchar(10) not null, city varchar(30) not null)

CREATE TABLE SP(S# integer not null, P# integer not null, QTY integer, Sdate date not null, Price Decimal(8,2)).

b. Đặt bí danh :

Cú pháp : CREATE SYNONYM <bí danh> FOR <tên quan hệ>

Tác dụng : để đặt bí danh (để ngắn gọn) cho tên bảng sử dụng câu lệnh:

Ví dụ 2 : Create synonym H For CONGTY

c. Xóa dữ liệu :

Cú pháp : DROP TABLE <tên quan hệ>

Tác dụng : để xóa bảng sử dụng câu lệnh :

Ví dụ 3: DROP TABLE CONGTY

V.2 Thao tác trên bảng dữ liệu:

a. Bổ sung :

Cú pháp : INSERT INTO <tên quan hệ> [(<ds thuộc tính>)] [VALUES (bộ giá trị)][<câu hỏi>]

Tác dụng : chèn 1 bản ghi vào quan hệ có tên được chỉ ra

Ví dụ 4 : INSERT INTO CONGTY(S#, Ten, STATUS, CITY) VALUES ('S4', 'John', Null, Null)

b. Xóa dữ liệu :

Cú pháp : DELETE [FROM] <tên quan hệ>WHERE <mệnh đề logic>

Tác dụng : xóa các bản ghi thỏa mệnh đề logic được chỉ ra

Ví dụ 5 : DELETE CONGTU WHERE S# = 'S1'

c. Sửa đổi dữ liệu :

Cú pháp : UPDATE <tên quan hệ> SET <tên thuộc tính><biểu thức>[,...]
WHERE < mệnh đề logic>

Tác dụng : Thay giá trị của thuộc tính bởi giá trị của biểu thức tương ứng của các bản ghi thỏa điều kiện được chỉ ra :

Ví dụ 6: UPDATE CONGTU SET CITY = ' London' WHERE
S# = 'S1' CITY = 'PARIS'

d. Thay đổi cấu trúc của bảng :

Cú Pháp : ALTER TABLE <tên quan hệ> ADD <tên thuộc tính> <k. dữ liệu>

Tác dụng : Bổ sung 1 thuộc tính vào quan hệ.

Ví dụ 7: ALTER TABLE NHANVIEN ADD Diachi Varchar (45)

IV.3 Hỏi đáp dữ liệu**a. Câu lệnh Select**

Cú pháp :

SELECT [*[DISTINCT]][<danh sách thuộc tính>
FROM <danh sách bảng>
[WHERE <mệnh đề điều kiện>]
[GROUP BY <danh sách thuộc tính>]
[HAVING <mệnh đề điều kiện>]
[ORDER BY <tên cột>|<biểu thức>|<số thứ tự cột> [ASC \ DESC]]

Trong đó các mệnh đề WHERE có dạng :

WHERE [NOT] <mệnh đề logic>
WHERE [NOT] <tên thuộc tính> [NOT] LIKE <chuỗi ký tự/ mẫu>
WHERE [NOT] <bthức> [NOT] BETWEEN [bthức1> and <bthức2>
WHERE [NOT] <bthức> [NOT] IN (<dsách> | <câu hỏi>)
WHERE [NOT] EXISTS (câu hỏi)
WHERE [NOT] <mđề logic> ANY | ALL (câu hỏi)
WHERE [NOT] bthức AND | OR [NOT] bthức2

Có thể kết nối nhiều kết quả truy vấn bằng cách dùng các phép toán tập hợp:

UNION: hợp, MINUS: hiệu, INTERSECT : giao, CONTAINS: chứa dạng:

Select .. From .. Where .. Group by ..order
UNION / MINUS/ INTERSECT/ CONTAINS
Select .. From .. Where .. Group by ..order

Tác dụng :

Truy vấn không điều kiện :

Với **SELECT *** : Chọn tất cả các cột thuộc bảng được chỉ ra .

SELECT DISTINCT : Chọn các giá trị phân biệt của thuộc tính được chỉ ra.

SELECT <ds thuộc tính> : Chọn các thuộc tính có tên được chỉ ra.

FROM <ds bảng> : Danh sách bảng chứa các thuộc tính.

Ví dụ 8: Hiển thị danh sách nhân viên

```
SELECT *
FROM NHANVIEN
```

Hiển thị danh sách họ tên nhân viên

```
SELECT hoten
FROM NHANVIEN
```

Chú ý : Kết quả câu lệnh trên không tương đương với biểu thức quan hệ :

$\Pi_{\text{hoten}}(\text{NHANVIEN})$

Vì câu lệnh trên hiển thị họ tên của các nhân viên có thể trùng nhau.

Câu lệnh tương đương với biểu thức quan hệ trên

```
SELECT DISTINCT hoten
FROM NHANVIEN
```

Truy vấn có điều kiện :

Sử dụng **WHERE <bthức logic>**

Ví dụ 9 : Hiển thị danh sách họ tên nhân viên có bậc lương = 2.4

```
SELECT DISTINCT hoten
FROM NHANVIEN
WHERE baclg = 2.4
```

Biểu thức đại số quan hệ tương đương :

$\Pi_{\text{hoten}}(\sigma_{\text{baclg} = 2.4}(\text{NHANVIEN}))$

Các tham số chọn của câu lệnh và hàm mẫu :

- **ORDER BY <tên cột> | <bthức> | <số thứ tự cột> [ASC \ DESC]**

Sắp xếp thứ tự theo <tên cột> | <bthức> | <số thứ tự cột> theo thứ tự tăng dần (**mặc nhiên**).

ASC : Xếp theo thứ tự tăng dần

DESC : Xếp theo thứ tự giảm dần

- **GROUP BY <danh sách thuộc tính>**

Phân nhóm theo danh sách thuộc tính.

• **HAVING <mđề đk>** : Thường được sử dụng cùng mệnh đề GROUP BY. Mệnh đề điều kiện không tác động vào toàn bảng được chỉ ra ở mệnh đề FROM, mà chỉ tác động lần lượt từng nhóm các bảng ghi đã chỉ ra ở mệnh đề GROUP BY.

- **Hàm mẫu :** SQL có các hàm mẫu sau :

COUNT (<tên cột>) : Cho số phần tử thuộc cột được chỉ ra.

COUNT (*) : Cho số bản ghi thỏa yêu cầu tìm kiếm mà không cần quan tâm đến bất kỳ cột nào.

SUM (<tên cột>) : Cho tổng giá trị các cột có tên được chỉ ra (có dữ liệu là kiểu số)

AVG (<tên cột>) : Cho giá trị trung bình của cột có tên được chỉ ra.

MIN (<tên cột>) : Cho giá trị bé nhất của cột có tên được chỉ ra.
 MAX (<tên cột>) : Cho giá trị lớn nhất của cột có tên được chỉ ra.
 SET (<tên cột>) : Tập giá trị của cột có tên được chỉ ra.

Ví dụ 10:

a. Cho biết số lần mặt hàng P2 đã được cung cấp

```
SELECT COUNT (*)
FROM SP
WHERE P# = 'P2'
```

b. Tính giá trị trung bình trọng lượng các mặt hàng

```
SELECT AVG(WEIGHT)
FROM P
```

c. Hãy đưa ra danh sách nhà cung ứng mặt hàng P1 :

```
SELECT S.SNAME
FROM S
WHERE S.S# IN
      (SELECT SP.S#
       FROM SP
       WHERE SP.P# = 'P1')
```

d. Hãy đưa ra danh sách số hiệu hãng cung ứng tất cả các mặt hàng
 (Biểu thức đại số quan hệ : $\Pi_{S\#, P\#} (SP) \div \Pi_{P\#} (P)$).

```
SELECT S#
FROM SP
GROUP BY S# CONTAINS
      (SELECT P#
       FROM P)
```

e. Đưa ra tất cả mặt hàng có tên bắt đầu là BOT

```
SELECT *
FROM S
WHERE SNAME LIKE 'BOT %'
```

Chú ý : *Ký hiệu % : Thay thế cho chuỗi con bất kỳ.
 ?, - : Thay cho 1 ký tự.*

f. Đưa ra những mặt hàng bán trước ngày 00-08-25 là 15 ngày

```
SELECT P#
FROM SP
WHERE '00-08-25' - sdate = 15
Đưa ra tên mặt hàng có đơn giá từ 1000 đến 2000
SELECT PNAME
FROM P, SP
WHERE P.P# = SP.P# and PRICE BETWEEN 1000 and 2000
```

h. Đưa ra mã hiệu của những nhà cung ứng không cung ứng mặt hàng nào

```
SELECT S#
FROM S
WHERE S# NOT IN
SELECT S#
FROM SP
```

hoặc :

```
SELECT S#
FROM S
MINUS /* hiệu */
SELECT S#
FROM SP
```

$(\Pi_{S\#}(SP) - \Pi_{S\#}(P))$

i. Đưa ra danh sách số hiệu nhà cung cấp cả mặt hàng P1 và P2

```
SELECT S#
FROM SP
WHERE P# = 'P1'
INTERSECT /* giao */
SELECT S#
FROM SP
WHERE P# = 'P2'
```

$(\Pi_{S\#}(\delta_{P\#='P1'}(SP)) \cap \Pi_{S\#}(\delta_{P\#='P2'}(SP)))$

k. Đưa ra tên các mặt hàng màu xanh và sắp xếp theo thứ tự giảm dần của mã số mặt hàng

```
SELECT PNAME, P#
FROM P
WHERE COLOUR = 'Blue'
ORDER BY P# ASC
```

Chú ý :

l. Đưa ra những nhà cung cấp đã cung cấp ít nhất 1 mặt hàng nào đó

```
SELECT *
FROM S
WHERE EXISTS
( SELECT *
FROM SP
WHERE SP.S# = S.S# )
```

n. Đưa ra tên mặt hàng có mã số mặt hàng của 1 mặt hàng nào đó mà hãng S1 đã bán

```
SELECT PNAME
FROM P
WHERE P# = ANY SELECT P#
FROM SP
```


WHERE S# = 'S1'

o. Đưa ra mã số những nhà cung cấp có số lượng của 1 mặt hàng nào đó lớn hơn bằng số lượng cung cấp mỗi lần của tất cả các hãng

```
SELECT S#
FROM SP
WHERE QTY >= ALL
      SELECT QTY
      FROM SP
```

hoặc :

```
SELECT S#
FROM SP
WHERE QTY = (SELECT MAX (QTY)
             FROM SP)
```

II. TỐI ƯU HOÁ CÂU HỎI

Khi thực hiện các biểu thức đại số quan hệ trên máy, đôi khi gặp khó khăn:

- Thiếu bộ nhớ: do phép kết nối,
- Có kết quả lớn : do tích Descartes..
- Thời gian tính toán lớn: Do phải xử lý nhiều bản ghi vì vậy, nên biến đổi các biểu thức đại số quan hệ cần tính thành các biểu thức đại số quan hệ tương đương khác phục được những nhược điểm
- trên .

a. Các chiến lược tối ưu:

- Thực hiện các phép chọn, phép chiếu thực hiện sớm nhất có thể: giảm bớt phần tử cần xử lý
- Nhóm dãy các phép toán 1 ngôi (*phép chọn, phép chiếu*)
- Tổ hợp những phép chọn xác định bởi phép tích Descartes

$$\delta F(r \times s) = r \bowtie s$$
- Tìm các biểu thức con chung trong 1 biểu thức (tính toán trước nhất 1 lần)
- Xử lý các file trước khi tính toán
- Sắp xếp chỉ mục hoặc sắp xếp trước khi tính toán (phép kết nối tự nhiên được nhanh hơn khi các thành phần được sắp)
- Lựa chọn thứ tự thực hiện sẽ phù hợp về thời gian và dung lượng

b. Các phép biến đổi tương đương của đại số quan hệ:

1. Tính kết hợp của tích Descartes, và phép kết nối:

$$(R_1 \times R_2) \times R_3 = R_1 \times (R_2 \times R_3)$$

$$(R_1 \bowtie R_2) \bowtie R_3 = R_1 \bowtie (R_2 \bowtie R_3)$$

2. Tính gộp mã phép chọn:

$$\delta_{F1}(\delta_{F2}(\dots(\delta_{Fn}(R))\dots)) = \delta_{F1} \wedge \delta_{F2} \wedge \dots \wedge \delta_{Fn}(R)$$

3. Tính tổ hợp của phép chiếu:

$$\Pi_{x1}(\Pi_{x2}(\dots(\Pi_{xn}(R))\dots)) = \Pi_{x1}(R)$$

Nếu $X_n \supset X_{n-1} \supset \dots \supset X_1$

4. Tính giao hoán giữa phép chọn và phép chiếu:

$$\delta_F(\Pi_x(R)) = \Pi_x(\delta_F(R))$$

5. Tính giao hoán giữa phép chọn và phép tích Descartes

a. $\delta_F(R_1 \times R_2) = \delta_F(R_1) \times R_2$

Nếu F chỉ liên quan đến R_1

b. $\delta_F(R_1 \times R_2) = \delta_{F_1}(R_1) \times \delta_{F_2}(R_2)$

Nếu $F = R_1 \wedge R_2$, F_1 chỉ liên quan đến R_1 , F_2 chỉ liên quan đến R_2

c. $\delta_F(R_1 \times R_2) = \delta_{F_2}(\delta_{F_1}(R_1) \times R_2)$

Nếu $F = F_1 \wedge F_2$, F_1 chỉ liên quan đến R_1 , F_2 liên quan cả R_1 và R_2

6. Tính giao hoán phép chọn và một phép hợp

$$\delta_F(R_1 \cup R_2) = \delta_F(R_1) \cup \delta_F(R_2)$$

7. Tính giao hoán 1 phép chọn và 1 phép hiệu

$$\delta_F(R_1 - R_2) = \delta_F(R_1) - \delta_F(R_2)$$

8. Tính giao hoán giữa phép chọn và phép tích Descartes

$$\Pi_x(R_1(Y) \times R_2(Z)) = \Pi_{x \cap Y}(R_1) \times \Pi_{x \cap Z}(R_2)$$

9. Tính giao hoán giữa 1 phép chiếu và 1 phép hợp

$$\Pi_x(R_1 \cup R_2) = \Pi_x(R_1) \cup \Pi_x(R_2)$$

Ví dụ 11 : Liệt kê mã số của các nhà cung ứng, cấp ít nhất 1 mặt hàng màu đỏ

$$\Pi_{S\#}(\delta_{SP.P\# = P.P\#}(\delta_{P.color = 'Red'}(P \bowtie SP)))$$

Áp dụng 2: Ta có biểu thức tương đương

$$\Pi_{S\#}(\delta_{SP.P\# = P.P\# \wedge P.color = 'Red'}(P \bowtie SP))$$

Áp dụng 5: ta có biểu thức tương đương

$$\Pi_{S\#}(\delta_{SP.P\# = P.P\#}(\delta_{P.color = 'Red'}(P) \bowtie SP)) \Leftrightarrow \Pi_{S\#}(\delta_{P.color = 'Red'}(P) \bowtie SP)$$



Bài tập

1. Sử dụng cơ sở dữ liệu :

S (S#, SNAME, STATUS, CITY)

P (P#, PNAME, COLOR, WEIGHT, CITY)

SP (S#, P#, QTY, SDATE, PRICE)

Hãy thể hiện các câu hỏi sau bằng SQL và biểu thức đại số quan hệ tương ứng.

- Đưa ra danh sách các mặt hàng màu đỏ
- Đưa ra danh sách số hiệu các nhà cung cấp cung ứng "ít nhất" 1 mặt hàng màu đỏ.
- Đưa ra danh sách số hiệu các nhà cung cấp tham gia cung ứng tất cả các mặt hàng màu đỏ.
- Đưa ra danh sách số hiệu các nhà cung cấp cung ứng tất cả các mặt hàng.
- Hãy đưa ra và danh sách tên tất cả các nhà cung ứng tất cả các mặt hàng.
- Đưa ra tên tất cả các nhà cung ứng không cung ứng 1 mặt hàng nào.

2. Cho CSDL gồm 2 quan hệ CC (MSCC, TEN_CC, DCCC) và MH (MSCC, MSMH).

Trong đó:

MSCC : mã số người cung cấp

TEN_CC : tên người cung cấp

DCCC : địa chỉ cung cấp

MSMH : mã số mặt hàng

Hãy biểu diễn các yêu cầu sau bằng đại số quan hệ và câu lệnh SQL tương ứng:

- a) Tìm mã số người cung cấp đã cung cấp

Q1: ít nhất một mặt hàng

Q2: Không cung cấp mặt hàng nào

Q3: Cung cấp một mặt hàng có MSMH là 15

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

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

3. Cho CSDL gồm các quan hệ sau:

DAIHOC (TENTRUONG, HIEUTRUONG, DIACHI)

KHOA (TENTRUONG, MSKHOA, TENKHOA, SOSINHVIENT)

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

trong đó:

SOSINHVIENT : số lượng sinh viên

MSKHOA : mã số Khoa

TENSIV : tên sinh viên

DIACHISV : Địa chỉ của sinh viên (Quê quán)

Hãy biểu diễn các yêu cầu sau bằng đại số quan hệ và câu lệnh SQL tương ứng:

- Trường đại học nào có khoa TINHOC
- Tổng số sinh viên học tin học ở tất cả các trường đại học
- Sinh viên nào học tại quê nhà
- Khoa nào của trường có số sinh viên cao nhất.

4. Hãy tối ưu hoá các câu hỏi truy vấn đã thực hiện ở các bài tập 1, 2, 3.

5. Cho 3 lược đồ quan hệ như sau:

$q(ABD)$, $r(BDF)$, và $s(FG)$

Tối ưu hoá các câu hỏi sau:

- a. $\delta_{D=d} (\Pi_{BDF} (q \bowtie r) - (\Pi_{BDF} (r \bowtie s)))$
- b. $(\Pi_{AD} (q \bowtie r \bowtie \delta_{G=g}(s)))$
- c. $(\Pi_B (\delta_{A=a}((\delta_{D=d} (q) \bowtie (\Pi_{BDF} (r \bowtie s))))))$

6. Cho các lược đồ quan hệ q, r, s tương ứng trên tập các thuộc tính AB, BC , và CD

Tối ưu hoá các câu hỏi sau:

- a. $E1 = \delta_{(B \leq C) \wedge (C=4) \wedge (D < A)} (q \bowtie r \bowtie s)$
- b. $E2 = (q \bowtie r) - (\Pi_{ABC} (q \bowtie \delta_{D=d} (r \bowtie s)))$



Tài liệu tham khảo

- [1]- Hồ Thuần, Contribution to the Theory of Relational Databases- *Bản in Roneo*- Luận án Tiến sĩ
- [2]- Vũ Đức Thi, *Cơ sở dữ liệu: kiến thức & thực hành*, NXB Thống Hê, Hà Nội 1997
- [3]- Đỗ Trung Tuấn, *Cơ sở dữ liệu*, NXB Giáo dục, Hà Nội, 1997
- [4]- Lê Tiến Vương, *Nhập môn CSDL Quan hệ*, NXB Thống Hê, Hà Nội 1999
- [5]- C.J Date, *Nhập môn các quan hệ cơ sở dữ liệu*, Dịch giả: Hồ Thuần, Nguyễn Quang Vinh, Nguyễn Xuân Huy, NXB Thống Hê, Hà Nội, Tập I (1985) Tập II(1986)
- [6]- Nguyễn Xuân Huy, Lê Hoài Bắc, *Bài Tập Cơ sở Dữ liệu*, NXB Thống Hê, Hà Nội, 2003
- [7]- Ullman J.D, *Principles of database and knowledgn base system*- Computer Science Press, 1986, (Có bản dịch tiếng Việt của Trần Đức Quang)
- [8]- Maier D, *The Theory of Relational Database*, Computer Science Press, 1983

