TẠP CHÍ KHOA HỌC TÍNH TOÁN VÀ ĐIỀU KHIỂN, Tập III (1987), Số 1 (1–7)

CƠ SỞ DỮ LIỆU QUAN HỆ ĐẶC TRƯNG VÀ TRIỂN VỌNG
(TỔNG QUAN)

HỒ THUẦN
Viện khoa học tính toán và điều khiển

I – MỞ ĐẦU

Trước những năm 60, mỗi chương trình ứng dụng đều có riêng một tập dữ liệu trên đó chương trình tiến hành xử lý, tính toán, và cấu trúc của chương trình ứng dụng phụ thuộc khá nhiều vào cấu trúc của tập dữ liệu đó. Tất yếu dẫn tới tình trạng dữ liệu thừa, không nhất quán và không quản lý tập trung được. Công sức giành cho việc bảo trì chương trình cũng rất lớn. Đầu những năm 70, đã hình thành tư tưởng về hệ cơ sở dữ liệu hợp nhất, trong đó hệ quản trị cơ sở dữ liệu (1) (QTCSDL) cung cấp cho mỗi chương trình ứng dụng một khung nhìn (views) riêng về dữ liệu chung, cài đặt các phép toán khác nhau để tìm kiếm và cập nhật dữ liệu, xử lý các tranh chấp giữa các người sử dụng v.v...

Sự cần thiết tổ chức dữ liệu theo một cách thức hoàn toàn xác định và chặt chẽ đã dẫn tới sự phát triển của một số mô hình dữ liệu trong đó các mô hình quan hệ, phân cấp và mạng là những mô hình quen biết nhất.

Trong mô hình phân cấp, dữ liệu được tổ chức dưới dạng các cấu trúc cây đơn giản. Một CSDL phân cấp là một tập hợp các cây như vậy (tức một rừng). Tổ chức này đòi hỏi một số kiểu bản ghi (record) phụ thuộc vào các kiểu khác. Và như vậy một bản ghi cho trước chỉ có ý nghĩa đầy đủ khi được nhìn trong ngữ cảnh của sự phân cấp. Để tránh dư thừa, cần dùng tới kiểu bản ghi lôgic ảo, tức con chỏ tới một kiểu bảo ghi khác. Hơn nữa các phép toán trong mô hình phân cấp thường là hướng thủ tục và ở mức thấp vì người sử dụng phải biết nhiều về tổ chức trong của dữ liệu.

Trong mô hình dữ liệu mạng dữ liệu được biểu diễn bởi các bản ghi liên kết với nhau thông qua các *mối nối* (link) tạo thành một đồ thị có hướng. Các bản ghi được nhóm thành các tập (2) (set), mỗi tập bao gồm một bản ghi chủ (owner record) với không hay nhiều bản ghi thành viên (member record). Để biểu diễn quan hệ nhiều–nhiều phải dùng tới kiểu bản ghi kết nối (connection record type) chứa thông tin chung về hai kiểu bản ghi được móc nối bởi quan hệ nhiều–nhiều.

So với mô hình phân cấp, các phép toán (thao tác) trong mô hình mạng đối xứng hơn. Tuy nhiên chúng vẫn hướng thủ tục và đòi hỏi người dùng quen biết với biểu diễn của dữ liệu ở mức thấp.

---

(1) Hệ QTCSDL là phần mềm cho phép một hay nhiều người sử dụng và/hay sửa đổi dữ liệu của 1 cơ sở dữ liệu.

(2) Tập ở đây khác với ý nghĩa của tập hợp trong toán học. Đây là một khái niệm do Data Base Task Group (DBTG đưa ra.

Sự ra đời của các hệ quản trị CSDL kiểu phân cấp và mạng đã là Một thành công lớn với tư cách là những công cụ điều khiển dữ liệu và giải phóng cho người lập trình ứng dụng khỏi các chi tiết trong việc xử lý các tệp vào, ra v.v... Tuy nhiên việc triển khai các chương trình ứng dụng (trong công nghiệp, thương nghiệp,...) vẫn còn nhiều khó khăn và không có năng suất cao vì những lý do chính sau :

1. Những hệ thống này làm một người lập trình ứng dụng bởi nhiều khái niệm không liên quan tới các nhiệm vụ thao tác và tìm kiếm dữ liệu của họ, buộc phải suy nghĩ và viết mã ở mức thấp không cần thiết của cấu trúc chi tiết (« Tập chủ–thành viên » của DBTG là một thí dụ điển hình !).

2. Các hệ này không có lệnh cho khả năng xử lý tập tức khả năng xử lý nhiều bản ghi một lúc. Kết quả là người lập trình buộc phải suy nghĩ và viết mã dưới dạng các chu trình lặp thường là không cần thiết.

3. Các yêu cầu tương tác trực tiếp của những người dùng ở đầu cuối – đặc biệt các câu hỏi không dự tính trước – chưa được giải quyết thỏa đáng.

Để chứng tỏ các hệ QTCSDL kiểu quan hệ tránh được ba nhược điểm nêu trên như thế nào, ta hãy nhìn lại ba mục tiêu đề ra của việc nghiên cứu đã dẫn tới mô hình quan hệ.

Mục tiêu thứ nhất là cho được một biên rõ nét giữa các khía cạnh lôgic và vật lý của việc quản trị CSDL (bao gồm thiết kế CSDL, tìm kiếm và thao tác dữ liệu). Ta gọi đó là mục tiêu độc lập dữ liệu.

Mục tiêu thứ hai là làm sao cho mô hình đơn giản về mặt cấu trúc sao cho mọi người dùng và lập trình có một cách hiểu chung về dữ liệu và do đó có thể trao đổi với nhau về CSDL. Ta gọi đó là mục tiêu trao đổi.

Mục tiêu thứ ba là đưa vào các khái niệm của ngôn ngữ cấp cao để người dùng có thể biểu thị thao tác (phép toán) trên các mảng thông tin lớn cùng một lúc. Ta gọi đó là mục tiêu xử lý tập.

Ở đây ta bỏ qua mục tiêu khác như tạo một cơ sở lý thuyết vững chắc cho việc tổ chức và quản lý CSDL v.v...

Trong bài này chúng tôi trình bầy ngắn gọn những đặc điểm của mô hình dữ liệu quan hệ, những tiêu chuẩn của một hệ quản trị CSDL quan hệ, một số phương hướng nghiên cứu tiếp tục và triển vọng của các hệ CSDL quan hệ.

II – MÔ HÌNH QUAN HỆ

Có thể nói Edgar F.Codd là người đầu tiên đề xuất mô hình quan hệ cho CSDL với công trình [1] mà ngày nay đã trở thành kinh điển. Sau đó Codd đã có nhiều đóng góp quan trọng phát triển đại số quan hệ, phép tính quan hệ và sự chuẩn hóa các quan hệ. Năm 1981 Codd được tặng giải thưởng Turing của hội máy tính Mỹ. Nhiều định nghĩa và khái niệm dưới đây đã được trình bầy trong [1]. Một nhập môn tuyệt hay về các khái niệm quan hệ và một tổng quan có giá trị về các hệ quản trị CSDL có thể tìm đọc theo thứ tự trong [2] và [3]. Do đó chúng tôi sẽ không trình bầy lại ở đây đầy đủ những khái niệm và định nghĩa đó.

Trong toán học, thuật ngữ *quan hệ* được định nghĩa như sau: Cho trước các tập $D_1, D_2, ..., D_n$ (không nhất thiết phải phân biệt), một *quan hệ* R là một tập các n – bộ trong đó mỗi bộ có phần tử thứ nhất thuộc D_1, phần tử thứ hai thuộc D_2, v.v... Các tập D_i được gọi là các miền. Số n được gọi là bậc của R còn số bộ trong R được gọi là lực lượng của nó.

Để đơn giản, có thể biểu diễn một quan hệ như một bảng trong đó *mỗi hàng biểu diễn một bộ*. Ta đặt tên cho bảng và cho các cột. *Các tên cột của bảng được gọi là các thuộc tính.*

Như vậy mọi thông tin trong một CSDL quan hệ đều được biểu diễn bởi các giá trị (Values) trong các bảng (ngay các tên bảng cũng xuất hiện như các xâu ký tự trong ít nhất một bảng).

Trong biểu diễn bảng của một quan hệ, các tính chất sau đây là rõ ràng (suy ra từ định nghĩa của quan hệ)

a) Không có hai hàng nào giống nhau;

b) Thứ tự các hàng là không quan trọng;

c) Thứ tự các cột là có ý nghĩa;

(Tuy nhiên nếu xét quan hệ như một tập các ánh xạ từ tập thuộc tính vào tập các giá trị trong các miền của thuộc tính thì thứ tự các cột cũng không có ý nghĩa).

Gọi tới dữ liệu theo giá trị thay cho theo vị trí sẽ tăng năng suất công việc của người lập trình cũng như người dùng ở đầu cuối (vị trí các cá thể dữ liệu (dataitems) thường bị thay đổi và không dễ gì một người có thể theo dõi được, đặc biệt nếu dãy chứa nhiều cá thể). Mặt khác, việc người lập trình và người dùng ở đầu cuối đều gọi tới dữ liệu theo một cách như nhau đã là một bước tiến dài đạt tới mục tiêu trao đổi.

Quan hệ bậc n được chọn làm cấu trúc gộp duy nhất (aggregate structure) (3) cho mô hình quan hệ vì với các toán tử thích hợp (chiếu, chọn, kết nối, ...) và một biểu diễn thích hợp (bảng) nó thỏa mãn cả ba mục tiêu đề ra ở trên.

Trong [4] Codd cho rằng một mô hình dữ liệu là sự tổ hợp của ít nhất ba thành phần sau:

1. Một tập các kiểu cấu trúc dữ liệu (các khối xây lắp CSDL).

2. Một tập các phép toán hay qui tắc suy diễn có thể được áp dụng cho mọi giá trị của các kiểu dữ liệu liệt kê trong [1] để tìm kiếm, suy dẫn hay sửa đổi dữ liệu từ bất kỳ phần nào của các cấu trúc đó trong mọi tổ hợp mong muốn.

3. Một tập hợp các qui tắc toàn vẹn tổng quát, xác định hiển hay không tập các trạng thái nhất quán của CSDL hay những thay đổi trạng thái hay cả hai—những qui tắc này là tổng quát theo nghĩa có thể áp dụng cho bất kỳ CSDL nào dùng mô hình đó (đôi khi chúng có thể được biểu thị như các qui tắc xen—cập nhật—loại bỏ).

Mô hình quan hệ là một mô hình dữ liệu theo nghĩa đó và là mô hình đầu tiên được định nghĩa như vậy.

Phần cấu trúc của nó bao gồm các miền, các quan hệ các bậc khác nhau, các thuộc tính, các bộ, các khóa và khóa chính.

Phần thao tác của mô hình quan hệ bao gồm các toán tử đại số (chiếu, chọn, kết nối, v.v...) biến đổi các quan hệ thành quan hệ (và từ đó bảng thành bảng).

Phần toàn vẹn bao gồm các qui tắc toàn vẹn: sự toàn vẹn thực thể và toàn vẹn qui chiếu (4).

---

(3) Phép gộp (aggregation) cho phép nhìn mỗi quan hệ giữa một số đối tượng như một đối tượng gộp.

(4) Có nghĩa các phép xen, sửa, xóa thực hiện trên các quan hệ cơ sở bị ràng buộc bởi hai qui tắc sau:

Qui tắc 1 [Tính toàn vẹn thực thể (entity integrity)]: không có một giá trị khóa chính nào của một quan hệ cơ sở được phép nhận giá trị không (tức chưa xác định) hay có một thành phần không.

Qui tắc 2 [Tính toàn vẹn qui chiếu (referential integrity)]: Giả sử thuộc tính A của một khóa chính kép (nhiều thuộc tính) của quan hệ R được xác định trên một miền sơ cấp D (miền trên đó một khóa chính một thuộc tính của CSDL được xác định). Khi đó, với mỗi giá trị v của A trong R phải tồn tại một quan hệ cơ sở S với khóa chính một thuộc tính (là B) sao cho v xuất hiện như một giá trị của B trong S.

Trong sự phát triển của mô hình quan hệ, luôn có sự phối hợp chặt giữa các khía cạnh cấu trúc, thao tác và toàn vẹn.

Mô hình quan hệ không chỉ đòi hỏi các cấu trúc quan hệ (có thể xem là các bảng) mà còn đòi hỏi một loại xử lý tập đặc biệt gọi là *xử lý quan hệ*. Xử lý quan hệ đòi hỏi xử lý toàn bộ quan hệ như một toán hạng. Mục đích chính của nó là tránh chu trình tạo điều kiện xử dụng có hiệu suất cho người dùng ở đầu cuối và người lập trình ứng dụng.

Toán tử chọn SELECT (cũng gọi RESTRICT) của đại số quan hệ lấy một quan hệ (bảng) làm toán hạng và tạo ra một quan hệ (bảng) mới bao gồm các bộ (hàng) của quan hệ thứ nhất.

Toán tử chiếu PROJECT cũng biến một quan hệ (bảng) thành một quan hệ mới, bao gồm các thuộc tính cột được chọn lọc của quan hệ thứ nhất.

Toán tử kết nối bằngE QUI — JOIN lấy hai quan hệ làm toán hạng và tạo ra mối quan hệ thứ ba bao gồm các hàng của quan hệ thứ nhất ghép với các hàng của quan hệ thứ hai nhưng chỉ ở những cột xác định trong quan hệ thứ nhất và những cột xác định trong quan hệ thứ hai tại đó có các giá trị được sánh.

Nếu chỉ giữ lại một trong những cột giống nhau, toán tử được gọi là NATURAL JOIN kết nối tự nhiên).

Dưới đây thuật ngữ kết nối được dùng để chỉ phép kết nối bằng hay kết nối tự nhiên.

Đại số quan hệ bao gồm các phép chiếu, chọn, kết nối và một số toán tử khác, nhằm dùng làm thước đo sức mạnh của một ngôn ngữ con dữ liệu (5) (data sublanguage). Mục tiêu xử lý tập của mô hình quan hệ nhằm đạt được nhờ một ngôn ngữ con dữ liệu có ít nhất sức mạnh của đại số quan hệ mà *không dùng tới các câu lệnh lặp hay đệ qui*.

Phần lớn sức mạnh dẫn xuất của đại số quan hệ có được chỉ từ các toán tử SELECT, PROJECT và JOIN miễn là toán tử JOIN không bị một hạn chế nào về mặt cài đặt (chẳng hạn các thuộc tính cần sánh phải cùng tên hay phải dựa vào một đường truy nhập thuộc kiểu đã được khai báo trước).

Vậy, ta nói rằng một ngôn ngữ con dữ liệu L có khả năng xử lý quan hệ nếu như các phép biến đổi được xác định bởi các toán tử PROJECT, SELETC và *JOIN không bị hạn chế*, của đại số quan hệ có thể được xác định trong L mà không cần tới các lệnh lặp hay đệ qui.

Một hệ quản trị CSDL được gọi là quan hệ nhất thiết phải có được:

1. Các bảng không có các mối nối dẫn dắt (6) giữa chúng mà người dùng có thể thấy được.

2. Một ngôn ngữ con dữ liệu với ít nhất khả năng xử lý quan hệ tối tiểu đã nói ở trên.

Từ đó suy ra *một hệ quản trị CSDL không có khả năng xử lý quan hệ thì phải xem hệ quản trị CSDL đó là phi quan hệ* (trường hợp chỉ thỏa mãn 1. được gọi là hệ bảng).

Chúng tôi lấy lại ở đây sự so sánh của Codd [5] về các loại quan hệ và bảng khác nhau. (hình 1).

Với mỗi lớp, sự mở rộng phần gạch chéo trong khối S nhằm chỉ mức độ trung thành của các phần tử thuộc lớp với các yêu cầu cấu trúc đối với mô hình quan hệ. Tương tự cho khối M các yêu cầu thao tác và khối I đối với các yêu cầu toàn vẹn.

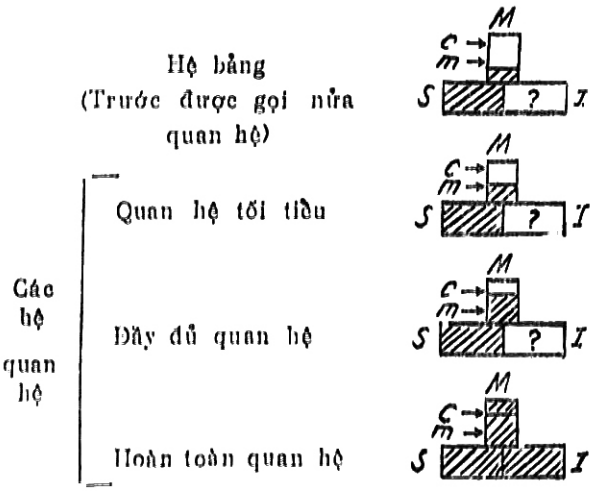
m chỉ khả năng xử lý quan hệ tối tiểu, c chỉ tính đầy đủ quan hệ (khả năng tương ứng với lôgic tân từ cấp một hai trị không có giá trị không (7) (nulls). Khi khối M được lấp đầy, nó chỉ khả năng ứng với đại số quan hệ đầy đủ (lôgic tân từ ba trị với một loại giá trị không). Dấu hỏi trong khối toàn vẹn cho mỗi lớp trừ quan lớp hệ đầy đủ chỉ rằng hiện tại sự đảm bảo toàn vẹn trong các hệ quan hệ là chưa thỏa đáng.

---

(5) Thuật ngữ « ngôn ngữ con dữ liệu » được dùng để chỉ một tập các toán tử CSDL nhằm nhúng vào một ngôn ngữ lập trình chủ, trong khi thuật ngữ « ngôn ngữ hỏi » thường chỉ một ngôn ngữ độc lập mà người dùng ở đầu cuối có thể dùng để tương tác trực tiếp với hệ quản trị CSDL.

(6) Ta gọi quá trình đi lần theo các mối nối (hay tổng quát hơn là các mối quan hệ) là sự lái tàu (navigation).

(7) Để đơn giản giá trị không ở đây hiểu là giá trị xác định nhưng chưa biết (nảy sinh trong các hệ CSDL với thông tin không đầy đủ).



Hình 1 – Phân loại các hệ quản trị CSDL

S = cấu trúc
M = thao tác
I = toàn vẹn
c = tính đầy đủ quan hệ
m = khả năng xử lý quan hệ tối tiểu

Ngoài sự phân loại nêu trên, các hệ QTCSDL còn được phân loại tùy thuộc chúng có hay không một ngôn ngữ con dữ liệu hai mới ; (1) tương tác trực tiếp ở đầu cuối và

(2) nhúng vào chương trình ứng dụng được viết bằng một ngôn ngữ chủ (chẳng hạn Cobol, Fortran, PL/I...)

Vì với khả năng như vậy sẽ cho phép người lập trình ứng dụng gõ rồi ở đầu cuối những câu lệnh CSDL mà anh ta muốn đưa vào trong các chương trình ứng dụng, cho phép trao đổi rộng rãi giữa các người lập trình, phân tích, người dùng ở đầu cuối và nhóm quản trị CSDL, cho phép giảm nhẹ việc học ngôn ngữ cho những người cần làm việc theo cả hai mới.

Ngày nay các hệ System R và INGRES có được kkả năng đó.

Ta gọi các hệ QTCSDL cho khả năng có một ngôn ngữ con hai cách như vậy là có *tính quan hệ đều* (uniform relational property).

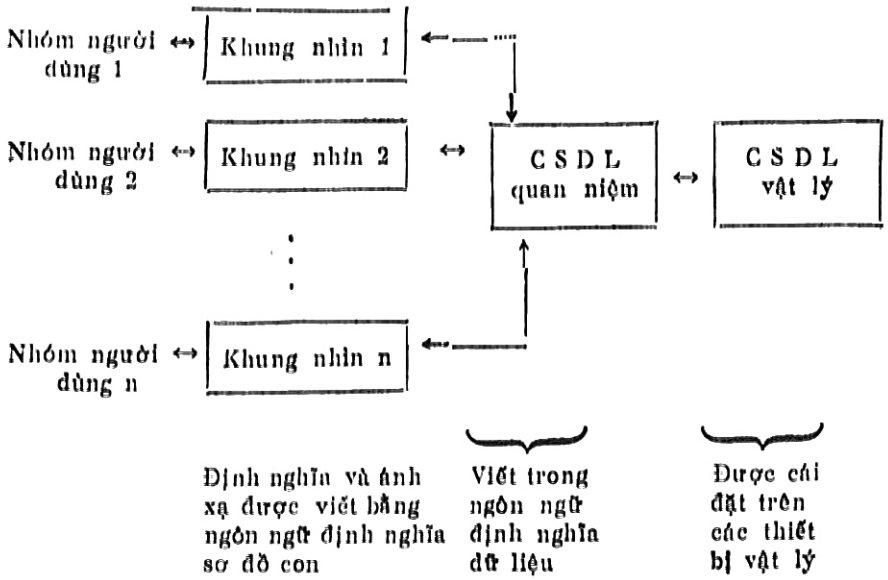
III – CÁC ƯU ĐIỂM CỦA MÔ HÌNH QUAN HỆ

Qua việc trình bày ở trên, ta có thể tóm tắt các ưu điểm của mô hình quan hệ dùng cho quản trị CSDL như sau :

1. **Tính đơn giản.** Điều này là rõ ràng. Thông tin về các đối tượng và các mối quan hệ giữa chúng được biểu diễn với một cấu trúc dữ liệu duy nhất dễ hiểu là các bảng. Người sử dụng phát triển yêu cầu của mình theo nội dung thông tin, không dẫn tới những phức tạp hướng hệ thống.

2. **Tính độc lập dữ liệu.**

Để chứng tỏ mô hình quan hệ cho khả năng thực hiện được tính độc lập dữ liệu, ta nhớ lại các mức độ trừu suất trong một CSDL được cho trong hình 2



Hình 2 — Các mức độ trừu suất trong một hệ CSDL

Dễ thấy là do ở mặt khớp—người sử dụng không có các chi tiết về cấu trúc lưu trữ và chiến lược truy nhập sơ đồ vật lý có thể bị thay đổi bởi người quản trị CSDL mà không ảnh hưởng tới sơ đồ quan niệm cũng như không đòi hỏi định nghĩa lại các sơ đồ con. Vì vậy không cần viết lại các chương trình ứng dụng. Ta gọi đó là sự độc lập dữ liệu vật lý.

Mối quan hệ giữa các khung nhìn và CSDL quan niệm cũng cho một loại độc lập được gọi là độc lập dữ liệu lôgic.

Có thể có những sửa đổi ở mô hình quan niệm mà không ảnh hưởng tới các sơ đồ con hiện có. Trường hợp cần thiết chỉ cần định nghĩa lại ánh xạ từ sơ đồ con vào sơ đồ quan niệm.

3. Tính đối xứng.

Các hệ CSDL trên cơ sở các kết nối giữa các bản ghi đã khiến cho một số câu hỏi dễ hỏi hơn những câu hỏi khác, cụ thể là những câu hỏi có cấu trúc phù hợp với cấu trúc của CSDL. Những câu hỏi có cấu trúc không phù hợp có thể rất khó hỏi.

Tình hình đó không xảy ra với các hệ CSDL quan hệ. Vì mọi thông tin đều được biểu diễn bởi các giá trị dữ liệu do đó không có một dạng ưu tiên nào đối với một câu hỏi ở mặt khớp người dùng. Chú ý là sự đối xứng của mô hình dữ liệu không nhất thiết kéo theo sự đối xứng của các cấu trúc vật lý bên dưới được hệ thống bảo quản.

4. Cơ sở lí thuyết vững chắc.

Mô hình dữ liệu quan hệ dựa trên lý thuyết toán học các quan hệ được phát triển hoàn chỉnh và dựa trên phép tính tân từ cấp một. Cơ sở lý thuyết đó đã cho phép định nghĩa tính đầy đủ quan hệ và cho một cơ sở nghiên cứu chặt chẽ của việc thiết kế tốt CSDL (lý thuyết chuẩn hóa).

Ngoài những ưu điểm chính kể trên, mô hình dữ liệu quan hệ còn có một loạt ưu điểm phụ khác trong đó đáng kể là với mô hình quan hệ ta dễ dàng xác định các ngôn ngữ quan hệ cấp cao, phi thủ tục dễ học và dễ sử dụng. Những ngôn ngữ này còn cho hệ thống linh hoạt tối ưu việc thực hiện một yêu cầu cho trước và thích nghi các cấu trúc dữ liệu lưu trữ với những nhu cầu thay đổi của người dùng.

Ngoài ra mô hình quan hệ còn cho khả năng phân biệt rõ nét giữa ngữ nghĩa dữ liệu và cấu trúc dữ liệu.

Chẳng hạn ngữ nghĩa của một CSDL có thể là khi một bản ghi phân xưởng bị xóa bỏ, thì mọi bản ghi công nhân của phân xưởng cũng bị xóa bỏ theo. Trong mô hình quan hệ những qui tắc ngữ nghĩa như vậy có thể được phát biểu độc lập với cấu trúc CSDL.

Thực tiễn cài đặt và sử dụng các hệ QTCSDL quan hệ cho thấy những hệ này đảm bảo được đầy đủ các chức năng như các hệ QTCSDL phi quan hệ (trong một số trường hợp còn hơn như khai thác được tất cả các quan hệ có ý nghĩa của từ CSDL, cho khả năng xử lý các khung nhìn (views) rất thuận lợi cho người viết trình ứng dụng và người dùng ở đầu cuối v.v...).

Trước đây còn có sự nghi ngại về hoạt trình ([8]) của các hệ quan hệ, cho rằng chậm.

Hoạt trình tốt được xác định bởi hai nhân tố:

1) Hệ phải có khả năng hoạt trình—hướng các cấu trúc dữ liệu vật lý;

2) Các yêu cầu dữ liệu viết trong ngôn ngữ cấp cao phải được dịch thành những dãy mã ở mức thấp ít nhất cũng tốt như người lập trình ứng dụng có thể viết bằng tay.

Thực tiễn cài đặt và làm việc của một số hệ quan hệ đã chứng tỏ hoạt trình của các hệ quan hệ không thua kém gì các hệ phi quan hệ.

IV—KẾT LUẬN

Mô hình quan hệ ngày càng được phát triển và hoàn chỉnh. Nhiều hướng nghiên cứu đang được đẩy mạnh: cài đặt các cơ chế đảm bảo các ràng buộc toàn vẹn, giảm nhẹ các ràng buộc về các khung nhìn kiểu kết nối cần sửa đổi, hoàn thiện các kỹ thuật tối ưu hóa câu hỏi để cải tiến hoạt trình; xây dựng các CSDL cho kỹ thuật và khoa học (khớp với Fortran chẳng hạn), xây dựng các hệ CSDL quan hệ phân tán trên một mạng truyền tin, quản lý và xử lý các giá trị không; đưa các kiểu dữ liệu trừu tượng vào ngôn ngữ CSDL và ngược lại đưa khả năng xử lý quan hệ vào ngôn ngữ lập trình (Pascal/R [6] là một thí dụ tốt), nghiên cứu khả năng xử lý các dữ liệu không thuần nhất (bao gồm hình ảnh, văn bản, ...).

Gần đây, xu thế mới của việc thích nghi các kỹ thuật của trí năng nhân tạo vào môi trường CSDL quan hệ tỏ ra có nhiều triển vọng [7]. Đóng góp chính của lập trình lôgic vào việc nghiên cứu CSDL quan hệ, theo quan điểm của trí năng nhân tạo, là việc đưa vào thông tin suy diễn.

Các hệ CSDL trên cơ sở đó sẽ ngày càng "thông minh» hơn (chẳng hạn có khả năng dùng thông tin về lĩnh vực ứng dụng hiện tại tìm thông tin không được biểu diễn hiển trong CSDL nhưng có thể được suy diễn từ các sự kiện và các tri thức của lĩnh vực ứng dụng; có khả năng giúp, dẫn dắt người dùng trong việc đặc tả câu hỏi, cho những câu trả lời thực tiễn và có ý nghĩa; có mặt khớp—người dùng là ngôn ngữ tự nhiên; v.v...).

Chúng ta hoàn toàn có cơ sở để tin rằng trong một tương lai gần, với những hướng nghiên cứu và phát triển tiếp như đã trình bày ở trên, cộng với sự thâm nhập của các kỹ thuật của trí năng nhân tạo, các hệ QTCSDL quan hệ «thông minh» ngày càng phục vụ đắc lực cho người sử dụng hơn trong việc khai thác có hiệu suất cao và triệt để các cơ sở dữ liệu.

TÀI LIỆU THAM KHẢO

1. Codd E.F., A relational model of data for large shared data banks. Comm. A.C.M. 13,6 (June 1970).

(Có bản dịch tiếng Việt, Tạp chí «Thông tin và tự động hóa chỉ huy, 1985; Hồ Thuần dịch).

2. Date C.J.. An introduction to Data base systems, Second edition 1977. (Có bản dịch tiếng Việt, nhà xuất bản Thống kê 1985, 1986; Hồ Thuần chủ biên).

3. Chamberlin, D.D., Relational DBMS, Computing Surveys March 1976. (Có bản dịch tiếng Việt, tạp chí Thông tin và tự động hóa chỉ huy, 1984; Nguyễn Thúc Hải dịch).

4. Codd. E.F., Data models in data base management.
ACM SIGMOD Record, 11, 2, (Feb. 1981).

5. Codd. E.F., Relational Data base: a practical foundation for productivity, Com, ACM. 25,2 (Feb. 1982).

6. Schmidt. J.W., Some hight level languages constructs for data of type relation. ACM TODS, 2, 3, (Sept. 1977).

7. Bic L. and Gilbert J.P., Learning from AI: New trends in Data base Technogy. Computer, March 1986.

(8) Chúng tôi dịch thuật ngữ «Performance» là hoạt trình với nghĩa trình diễn sự hoạt động.