Tạp chí Tin học và Điều khiển học, T. 17, S. 3 (2001), 77–86

XÂY DỰNG MỘT THUẬT TOÁN CHO CHUẨN HÓA QUAN HỆ VỀ DẠNG CHUẨN 3 DỰA TRÊN DỮ LIỆU

ĐẶNG XUÂN HỒNG

Abstract. The third normal form (3NF) which was introduced by E. F. Code is an important normal form for relation in the relation database. In this paper, we put forward a method for normalizing a relation (that is a data file) from the first normal form to the third normal form.

Tóm tắt. Dạng chuẩn 3 đóng vai trò quan trọng trong mô hình dữ liệu quan hệ. Trong bài báo này, chúng tôi đề xuất phương pháp chuẩn hóa một quan hệ, mà thực chất là một tệp dữ liệu, từ dạng chuẩn 1 về dạng chuẩn 3.

1. MỞ ĐẦU

Mô hình dữ liệu quan hệ được E. F. Codd đề xuất là mô hình dữ liệu phổ biến nhất hiện nay. Hạt nhân chính của mô hình này là quan hệ mà thực chất là một tệp dữ liệu (đôi khi gọi là bảng). Một vấn đề quan trọng là việc chuẩn hóa các tệp dữ liệu. Thực chất của vấn đề này là việc chuyển một tệp dữ liệu bất kỳ về tệp dữ liệu ở những dạng chuẩn quen thuộc như dạng chuẩn 2 (2NF) và dạng chuẩn 3 (3NF). Mục tiêu chính của bài báo này là đề xuất một phương pháp chuẩn hóa một tệp dữ liệu về dạng 3NF.

2. MỘT SỐ ĐỊNH NGHĨA

Trong phần này, chúng tôi trình bày những định nghĩa cơ bản nhất cần dùng cho việc trình bày bài báo (có thể đọc trong [1-8]).

Định nghĩa 2.1. (quan hệ) Một quan hệ r xác định trên tập hữu hạn và không rỗng các thuộc tính $\Omega = \{a_1, a_2, \ldots, a_n\}$ là một tập hợp m bộ có dạng:

$$h_j = (A_1, A_2, \ldots, A_n), \quad j = 1, \ldots, m,$$

$A_1 \in \mathrm{Dom}\,(a_1)$, $A_2 \in \mathrm{Dom}\,(a_2), \ldots, A_n \in \mathrm{Dom}\,(a_n)$, trong đó $\mathrm{Dom}\,(a_i)$ là miền giá trị của thuộc tính a_i. Nói cách khác một quan hệ r trên tập thuộc tính $\Omega = \{a_1, a_2, \ldots, a_n\}$ là một tập con của tích Descarts:

$$\mathrm{Dom}\,(a_1) \times \mathrm{Dom}\,(a_2) \times \ldots \times \mathrm{Dom}\,(a_n).$$

Đương nhiên quan hệ r có thể bị thay đổi theo thời gian do việc thực hiện các phép toán cập nhật trên các bộ của quan hệ r (thêm vào, loại bỏ, sửa đổi).

Trong khi đó, ngữ nghĩa của một bộ thuộc r là bất biến và điều đó liên quan tới mô tả cấu trúc của tập các bộ mà ta gọi là lược đồ quan hệ.

Định nghĩa 2.2. (phụ thuộc hàm trên quan hệ) Cho $\Omega = \{a_1, a_2, \ldots, a_n\}$ là tập hữu hạn và không rỗng các thuộc tính, $r = \{h_1, h_2, \ldots, h_m\}$ là một quan hệ trên Ω và A, B là các tập con của Ω $(A < B \subseteq \Omega)$.

Khi đó, chúng ta nói A xác định hàm cho B hay B phụ thuộc hàm vào A trong r (ký hiệu là: $A \xrightarrow[r]{f} B$) nếu:

$$\forall h_i, h_j \in r)((\forall a \in A)(h_i(a) = h_j(a)) \Rightarrow (\forall b \in B)(h_i(b) = h_j(b))).$$

Đặt $F_r = \left\{(A,B) : A,B \subseteq \Omega,\ A \xrightarrow[r]{f} B\right\}$. Khi đó F_r được gọi là họ đầy đủ các phụ thuộc hàm của quan hệ r.

Định nghĩa 2.3. (Hệ tiên đề Armstrong) Giả sử Ω là tập hữu hạn và không rỗng các thuộc tính và kí hiệu: $P(\Omega)$ là tập các tập con của Ω.

Cho $Y \subseteq P(\Omega) \times P(\Omega)$. Chúng ta nói Y là một họ f trên Ω nếu đối với mọi $A,B,C,D \subseteq \Omega$:
(1) $(A,A) \in Y$,
(2) $(A,B) \in Y, (B,C) \in Y$ thì $(A,C) \in Y$,
(3) $(A,B) \in Y, A \subseteq C, D \subseteq B$ thì $(C,D) \in Y$,
(4) $(A,B) \in Y, (C,D) \in Y$ thì $(A \cup C, B \cup D) \in Y$.

Rõ ràng, F_r là một họ f trên Ω.

Định nghĩa 2.4. (hệ bằng nhau) Giả sử r là quan hệ trên Ω, đặt:

$$E_r = \{E_{ij} : 1 \leq i < j \leq |r|\} \text{ trong đó } E_{ij} = \{a \in \Omega : h_i(a) = h_j(a)\}.$$

Khi đó E_r được gọi là hệ bằng nhau của r.

$$M_r = \{A \in P(\Omega) : \text{tồn tại } E_{ij} = A, \text{ và } \nexists E_{pq} : A \subset E_{pq}\}.$$

Khi đó M_r được gọi là hệ bằng nhau cực đại của r.

Định nghĩa 2.5. (bao đóng của một tập thuộc tính trên một quan hệ) Giả sử r là quan hệ trên tập các thuộc tính Ω, và $A \subseteq \Omega$.

Ký hiệu $A_r^+ = \left\{a : A \xrightarrow[r]{f} \{a\}\right\}$. Khi đó A_r^+ được gọi là bao đóng của A trên r.

Định nghĩa 2.6. (khóa của quan hệ) Giả sử r là một quan hệ, và $A \subseteq \Omega$. Khi đó A là một khóa của r nếu:

$$A \xrightarrow[r]{f} \Omega.$$

Gọi A là một khóa tối tiểu của r nếu:
i) A là một khóa của r,
ii) bất kỳ một tập con thực sự của A không là khóa của r.

Chú ý: Nếu chỉ thỏa mãn điều kiện i) thì A đôi khi còn được gọi là siêu khóa.

Ký hiệu K_r tương ứng là tập tất cả các khóa tối tiểu của r.

Định nghĩa 2.7. (thuộc tính cơ bản, thuộc tính thứ cấp) Giả sử r là một quan hệ trên Ω và K_r là tập tất cả các khóa tối tiểu của r. Khi đó nói a là thuộc tính cơ bản của r nếu tồn tại một khóa tối tiểu K $(K \in K_r)$ để a là một phần tử của K.

Nếu a không thỏa mãn tính chất trên thì a là thuộc tính thứ cấp.

Nhận xét: Thuộc tính cơ bản và thuộc tính thứ cấp đóng vai trò quan trọng trong việc chuẩn hóa các quan hệ.

3. LÝ THUYẾT CHUẨN HÓA

3.1. Dạng chuẩn 1 (1NF)

Định nghĩa 3.1. Quan hệ r được gọi là ở dạng chuẩn 1 nếu các phần tử của nó $h_i(a_j)$ đều là các giá trị sơ cấp. Một giá trị sơ cấp được hiểu là giá trị không thể chia nhỏ được nữa.

3.2. Dạng chuẩn 2 (2NF)

Định nghĩa 3.2. Quan hệ r được gọi là ở dạng chuẩn 2 nếu:
- r đã ở dạng chuẩn 1.
- Với mọi khóa tối tiểu K không tồn tại phụ thuộc hàm $A \to \{a\} \in F_r$ với $A \subset K$, $A \neq K$ và a là thuộc tính thứ cấp.

Định lý 3.1. *Cho r là một quan hệ Ω. Khi đó r ở 2NF khi và chỉ khi:*

- *r là 1NF.*
- *Mỗi thuộc tính thứ cấp của r đều phụ thuộc hàm đầy đủ vào mọi khóa tối tiểu.*

(Phụ thuộc hàm $A \to B$ được gọi là phụ thuộc hàm đầy đủ nếu không tồn tại tập hợp $A' \subset A$ sao cho $A' \to B$).

3.3. Dạng chuẩn 3 (3NF)

Định nghĩa 3.3. Quan hệ r được gọi là ở dạng chuẩn 3 nếu:

- r đã ở dạng chuẩn 2NF.
- $A \to \{a\} \notin F_r$ đối với những tập thuộc tính A mà:

$$A_r^+ \neq \Omega,\ a \notin A,\ a \notin \cup K.$$

Định nghĩa này có thể giải thích như sau:

Với mọi thuộc tính thứ cấp a và với mọi khóa tối tiểu K không tồn tại tập thuộc tính A sao cho $K \to A$, $A \to \{a\}$.

Định lý 3.2. *Giả sử r là một quan hệ trên Ω. Khi ấy r ở dạng 3NF nếu và chỉ nếu:*

- *r đã ở 2NF.*
- *Không có thuộc tính thứ cấp nào của r phụ thuộc hàm bắc cầu vào một khóa tối tiểu.*

(Phụ thuộc hàm $A \to C$ được gọi là bắc cầu nếu tồn tại tập thuộc tính B ($B \neq A$, $B \neq C$) mà $A \to B$ và $B \to C$. Trong trường hợp ngược lại C được gọi là phụ thuộc hàm trực tiếp vào A).

Định lý 3.3. *Giả sử r là một quan hệ trên Ω. Khi ấy r là 3NF nếu và chỉ nếu với mọi $A : A^+ = A$, $a \in A$ và a thuộc tính thứ cấp thì $\{A - a\}_r^+ = \{A - a\}$.*

4. CHUẨN HÓA TỆP DỮ LIỆU

Khi thiết kế các cơ sở dữ liệu quan hệ, người ta thường tìm cách loại bỏ được các dị thường khi thao tác với các tệp dữ liệu trong cơ sở dữ liệu. Điều này chỉ được hạn chế khi người ta chuẩn hóa được các tệp dữ liệu về dạng chuẩn 3. Tuy nhiên, với lý thuyết chuẩn hóa được trình bày trong phần trên thì công việc chuẩn hóa sẽ đòi hỏi thời gian và công sức khá lớn, do đó để có thể áp dụng được lý thuyết vào thực tế, trong phần này chúng tôi tìm cách đơn giản hóa các yêu cầu trên mà vẫn có thể chuẩn hóa được các tệp dữ liệu.

Dạng chuẩn 1

Ta nói rằng một quan hệ là dạng chuẩn 1 nếu tất cả các giá trị các thuộc tính của nó là sơ cấp.

Dạng chuẩn 2

Một quan hệ là 1NF được xem là dạng chuẩn 2 nếu tất cả các phụ thuộc hàm giữa khóa được chọn làm khóa chính và các thuộc tính khác của nó đều là đầy đủ.

Ta nhận thấy rằng định nghĩa dạng chuẩn 2 trong Phần 3 chặt hơn vì điều kiện phụ thuộc hoàn toàn liên quan đến mọi khóa tối tiểu, chứ không chỉ liên quan đến một khóa tối tiểu được chọn làm khóa chính.

Để kiểm tra xem một quan hệ nào có ở dạng chuẩn 2 hay không, thì việc đầu tiên cần phải thực hiện là tìm ra khóa của quan hệ. Như đã trình bày ở các phần trước, một quan hệ có thể có nhiều hơn một khóa, do đó việc chọn ra một khóa phù hợp với ý nghĩa thực tế là khá khó khăn. Hơn nữa bản thân dữ liệu trong quan hệ cũng phải lột tả được bản chất sự phụ thuộc lẫn nhau giữa các thuộc tính trong quan hệ.

Ta phát hiện ra quan hệ không ở dạng chuẩn 2 khi trong quan hệ đưa vào tồn tại một tập thuộc tính Y chỉ phụ thuộc hàm vào một bộ phận K' của khóa K, ($K' \subset K$). Khi ấy, ta phải tách các thuộc tính trong $K' \cup Y$ thành một quan hệ mới và các thuộc tính nằm trong $\Omega - Y$ thành một quan

hệ khác. K khi đó sẽ là khóa của quan hệ $\Omega - Y$, còn K' sẽ là khóa của các thuộc tính trong quan hệ mới $K' \cup Y$.

Dạng chuẩn 3

Một quan hệ đã là 2NF được xem là ở dạng chuẩn 3 nếu tất cả các phụ thuộc hàm giữa khóa được chọn làm khóa chính và các thuộc tính khác của nó đều là trực tiếp.

Nhận xét: Một quan hệ có nhiều khóa nhận dạng không thể thỏa mãn dạng chuẩn 3. Mặt khác, định nghĩa 3NF trong Phần 3 chặt hơn vì điều kiện phụ thuộc đầy đủ và phụ thuộc trực tiếp liên quan đến mọi khóa tối tiểu, chứ không chỉ liên quan đến một khóa tối tiểu được chọn làm khóa chính.

Ta phát hiện ra một quan hệ đã ở dạng chuẩn 2 mà không ở dạng chuẩn 3 khi trong quan hệ đưa vào tồn tại các phụ thuộc hàm bắc cầu vào khóa chính có dạng: $K \to X$, $X \to Y$, trong đó $X \not\subseteq K$, $Y \not\subseteq X$.

Nếu tìm thấy phụ thuộc hàm bắc cầu như trên thì tách quan hệ hiện thời thành hai quan hệ $X \cup Y$ và $\Omega - Y$.

Kiểm tra với các quan hệ con xem đã ở dạng chuẩn 3 hay chưa, nếu chưa lại thực hiện tách tiếp như trên.

5. THUẬT TOÁN CHUẨN HÓA

Nội dung chính của phần này là thiết kế một thuật toán chuẩn hóa một quan hệ về dạng chuẩn 3 mà thực chất là tách quan hệ này thành các quan hệ con ở dạng chuẩn 3 mà không làm mất mát thông tin.

Vào: Một quan hệ r bất kỳ (ta giả thiết các trường đã chứa trị sơ cấp, tức đã ở dạng chuẩn 1).

Ra: Các quan hệ ở dạng chuẩn 3 và r nhận được bằng việc kết nối tự nhiên các quan hệ này.

Phương pháp:

Bước 1: Quan hệ cần được chuẩn hóa.

Bước 2: Tính hệ bằng nhau E_r theo công thức:
$E_r = \{E_{ij} : 1 \le i < j \le |r|\}$ trong đó $E_{ij} = \{a \in \Omega : h_i(a) = h_j(a)\}$.

Bước 3: Tính hệ bằng nhau cực đại:
$M_r = \{A \in P(\Omega) : \exists E_{ij} = A, \text{ và } \nexists E_{pq} : A \subset E_{pq}\}$.

Bước 4: Tìm khóa chính: Lần lượt tính $K_0, K_1, \ldots, K_n$ (với n là số thuộc tính của quan hệ r) để tìm ra khóa K.

Chú ý: ở bước này, nếu quan hệ đưa vào không điển hình (tức bản chất số liệu không lột tả được các phụ thuộc dữ liệu giữa các thuộc tính) thì khóa chính tìm được có thể không chính xác.

Bước 5: Kiểm tra và tách quan hệ về dạng chuẩn 2:
Nếu số thuộc tính của khóa K là một ($|K| = 1$) thì ta có thể kết luận quan hệ đã ở dạng chuẩn 2 và chuyển sang bước 7.
Ngược lại, ta thực hiện công việc sau:
Tính $F_n = \Omega - K$ (Ω là tập các thuộc tính của quan hệ đưa vào). Ta gọi F_n là tập các thuộc tính không khóa.
Với mỗi $a_i \in F_n$ ta kiểm tra như sau:
Đặt $T = K$
Với mỗi $b_j \in T$ ta kiểm tra xem phụ thuộc hàm:
$\{T - \{b_j\}\} \to \{a_i\}$ có thỏa mãn không, bằng cách tính bao đóng của tập $\{T - \{b_j\}\}$, sau đó xét xem $\{a_i\}$ có thuộc tập bao đóng này không.
Nếu thuộc ta gán $T = T - \{b\}$.
Ngược lại ta giữ nguyên T.

Cứ tiếp tục như vậy cho đến khi ta duyệt hết các phần tử của tập T. Nếu $|T| < |K|$ $(T \subset K)$ thì sẽ tồn tại một phụ thuộc bộ phận giữa thuộc tính a_i và T. Do đó quan hệ chưa ở 2NF và phải được phân tách như sau:

Quan hệ đang xét sẽ bị loại bỏ đi thuộc tính a_i, tập K vẫn là khóa của quan hệ này. Và ta thêm một quan hệ mới với tập thuộc tính là $T \cup \{a_i\}$ và T sẽ là khóa của quan hệ này.

Sau khi duyệt hết các $a_i \in F_n$ ta chuyển sang bước 6.

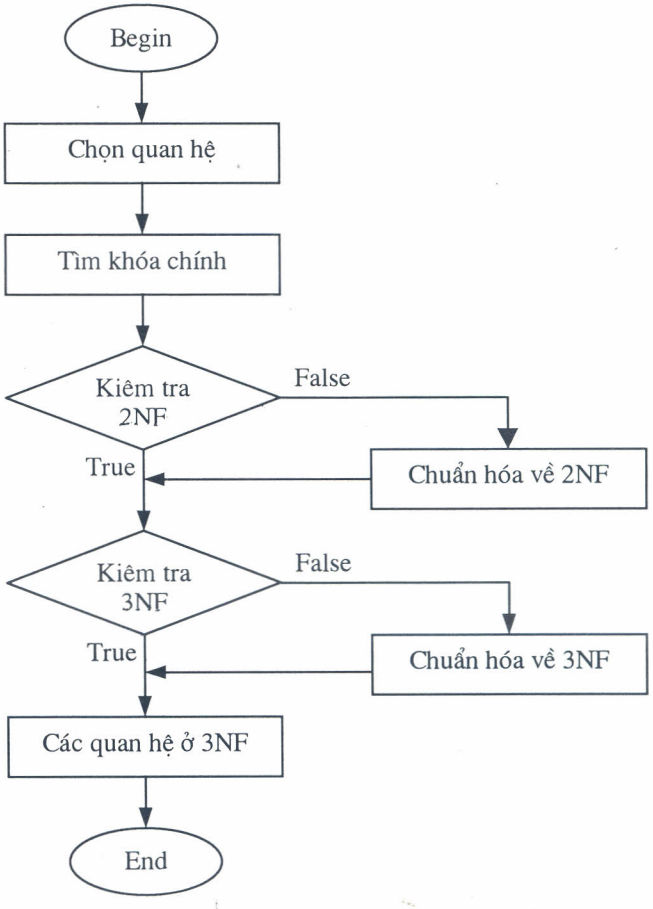
Bước 6: Gộp các quan hệ có cùng chung khóa: Đối với tập các quan hệ bộ phận đã được tách ở bước trên, nếu trong số chúng có một số quan hệ có chung một khóa, ta giả thiết đó là khóa T, thì ta tiến hành gộp các quan hệ này thành một quan hệ chung. Khóa của quan hệ gộp này sẽ là khóa T và tập thuộc tính của quan hệ này sẽ là hợp của tập tất cả các thuộc tính của các quan hệ con thành phần (có chung khóa T).

Bước 7: Đối với từng quan hệ bộ phận kiểm tra xem có ở dạng chuẩn 3 không, nếu mọi quan hệ đã ở dạng chuẩn 3 thì chuyển sang bước 8, nếu không tiến hành phân rã. Cụ thể:

Với mỗi quan hệ r_i thực hiện:

Với mỗi $a \in F_{ni}$ (F_{ni} là tập các thuộc tính không khóa của quan hệ i hiện thời) tính: bao đóng $\{a\}^+_{ri}$.

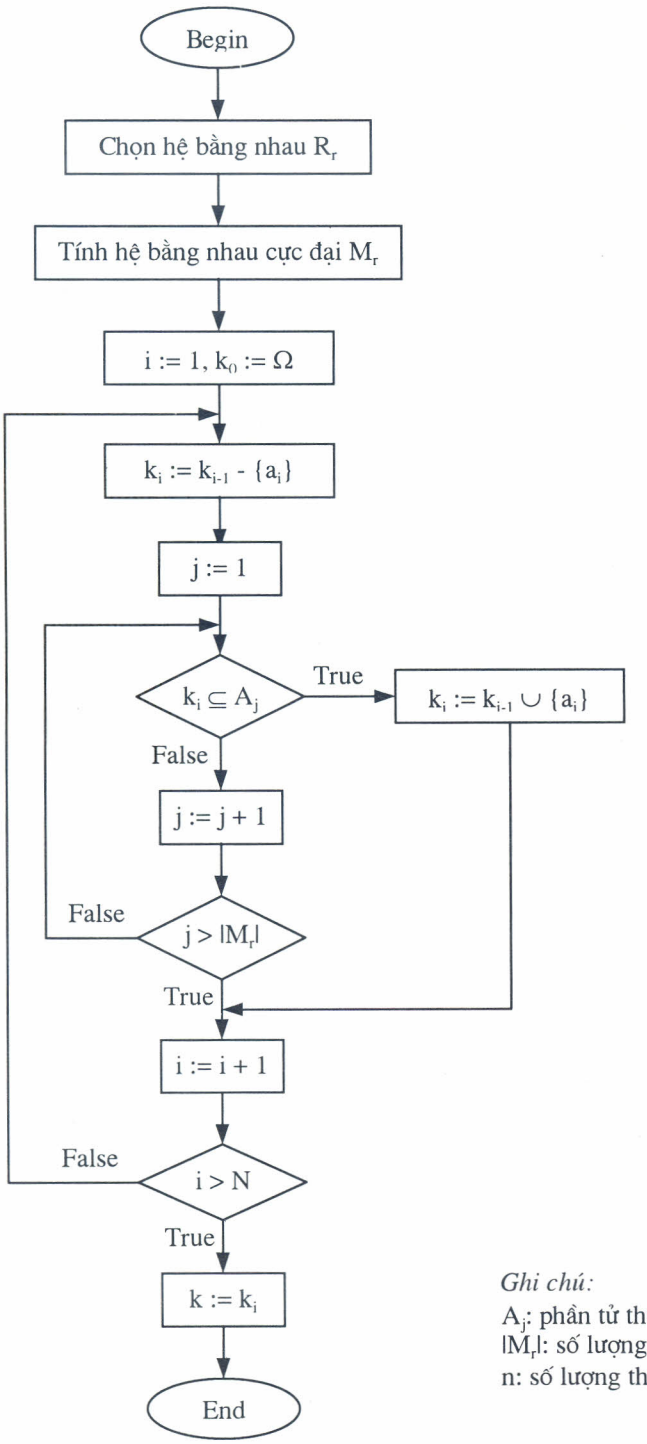
Nếu $\{a\}^+_{ri} - (K_i \cup \{a\}) = B \neq \emptyset$ có nghĩa là tồn tại một phụ thuộc bắc cầu giữa K_i và B, khi đó ta phải tách $\{a\} \cup B$ thành một quan hệ mới, còn quan hệ r_i hiện thời sẽ loại bỏ bớt các thuộc tính thuộc B.



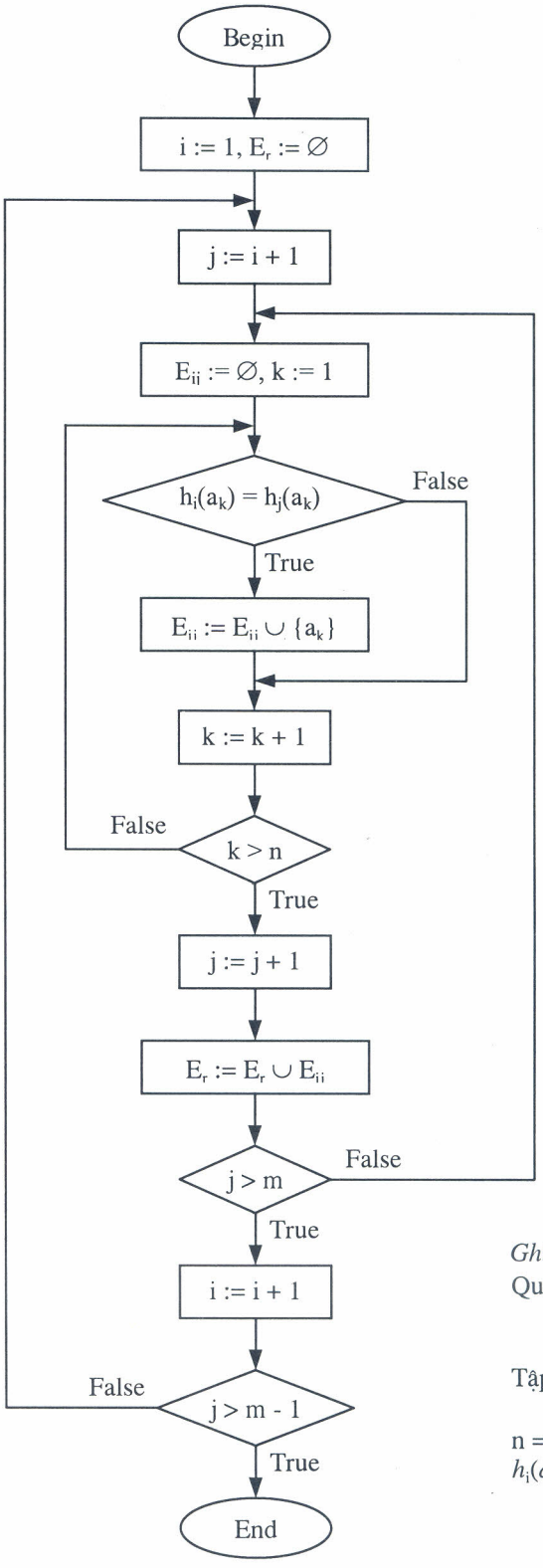
Sơ đồ 1. Sơ đồ thuật toán tổng quát

Bước 8: Hiển thị các quan hệ đã ở dạng chuẩn 3. Không khó khăn, có thể thấy rằng quan hệ ban đầu r nhận được bằng phép kết nối tự nhiên các quan hệ con ở dạng chuẩn 3.

Ta xây dựng các thuật toán như các sơ đồ 1 - 3.

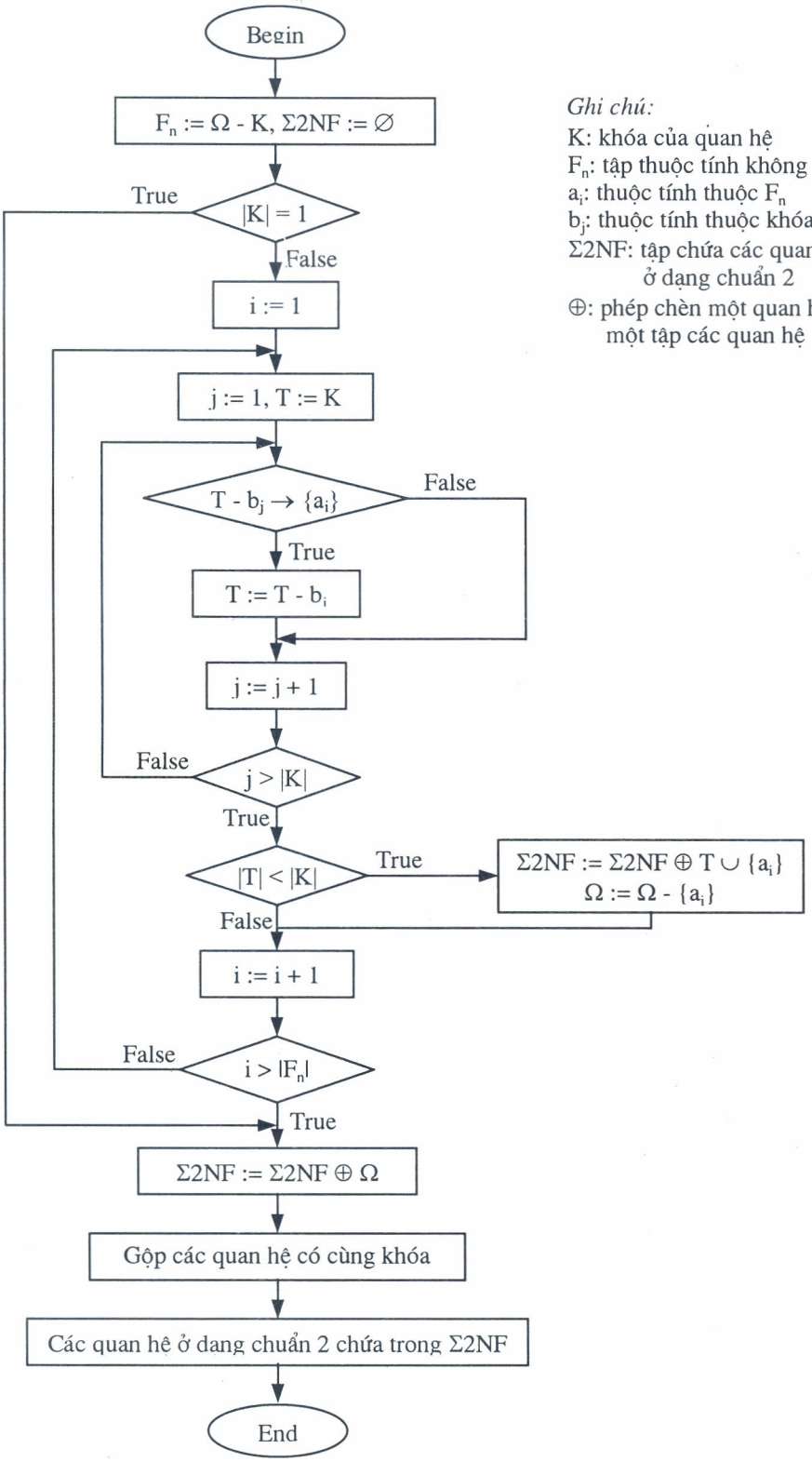


Sơ đồ 2. Sơ đồ thuật toán tìm khóa của quan hệ



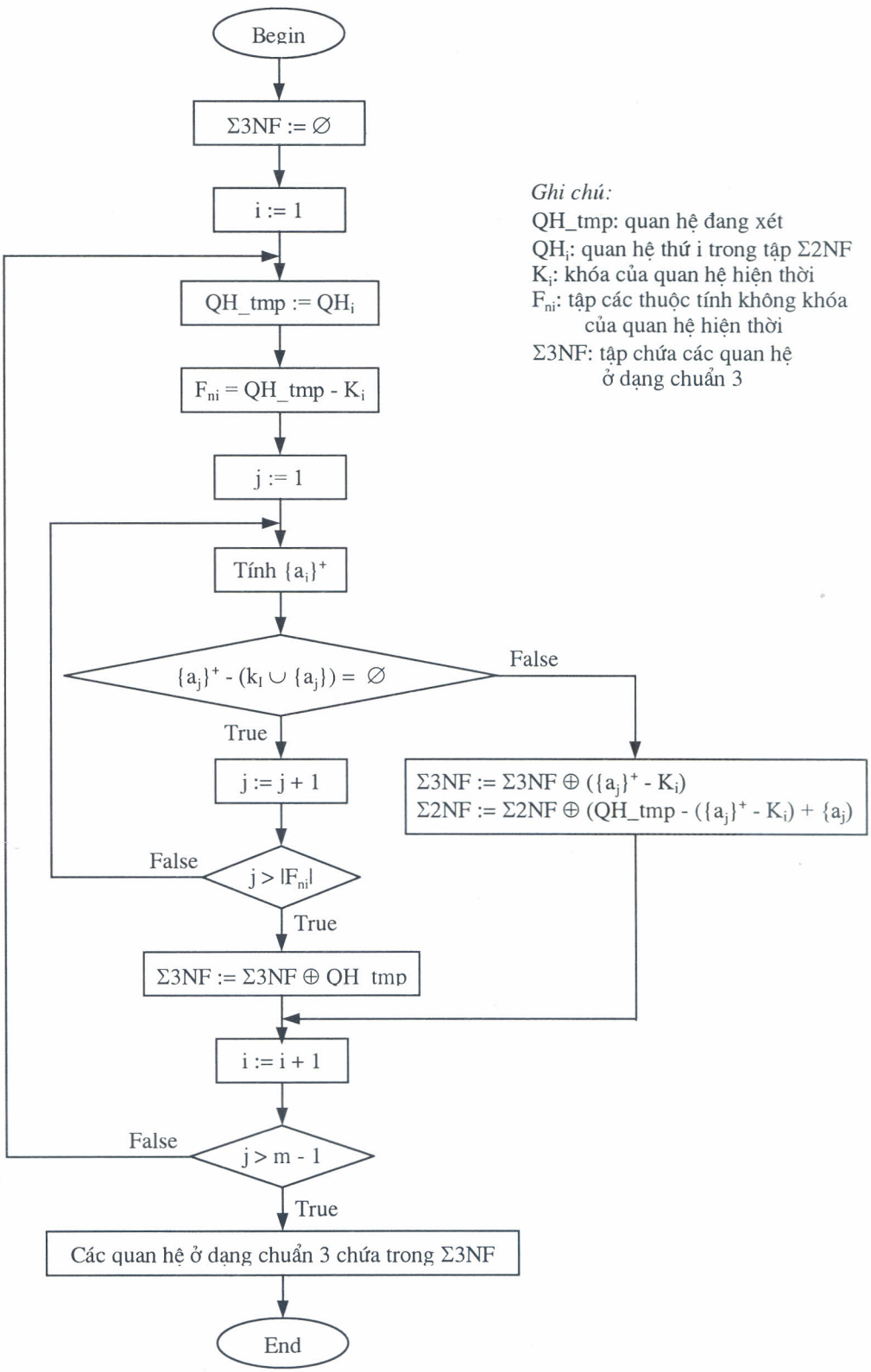
Ghi chú:
Quan hệ *r* là tập *m* bộ có dạng:
$h_i = \{A_1, A_2, ..., A_n\}$
$j = 1, ..., m$
Tập thuộc tính $\Omega = a_1, a_2, ..., a_n\}$
$i = 1, ..., n$
$n = |\Omega|$: số thuộc tính của r
$h_i(a_k)$ là giá trị tại thuộc tính *k* của bộ thứ i ($i = 1, ..., m, k = 1, ..., n$)

Sơ đồ 3. Sơ đồ thuật toán tính hệ bằng nhau E_r



Ghi chú:
K: khóa của quan hệ
F_n: tập thuộc tính không khóa
a_i: thuộc tính thuộc F_n
b_j: thuộc tính thuộc khóa K
Σ2NF: tập chứa các quan hệ ở dạng chuẩn 2
⊕: phép chèn một quan hệ vào một tập các quan hệ

Sơ đồ 4. Sơ đồ thuật toán chuẩn hóa về dạng chuẩn 2



Sơ đồ 5. Sơ đồ thuật toán chuẩn hóa về dạng chuẩn 3

Lời cám ơn:

Tác giả xin chân thành cảm ơn PGS. TS. Vũ Đức Thi đã đóng góp những ý kiến quý báu trong quá trình hoàn thành bài báo này.

TÀI LIỆU THAM KHẢO

[1] Armstrong W. W., *Dependency Structures of Database Relationships*, Information Processing 74, Holland Publ. Co., **74** (1974) 580–583.

[2] Beeri C., Bernstein P. A., Computational problems related to the design of normal form relational schemas, *ACM Trans. on Database Syst.* **4** (1979) 30–59.

[3] Beeri C., Dowd M., Fagin R., Staman R., On the structure of Armstrong relations for functional dependencies, *J. ACM* **31** (1984) 30–46.

[4] Demetrovies J., Katona G. O. H., A survey of some combinatorial results concerning functional depencies in database relations, *Annals of Mathematics and Artificial Intelligence* **7** (1993) 63–82.

[5] Demetrovies J., Thi V. D., Algorithm for generating Armstrong relations and inferring functional dependencies in the relational datamodel, *Computers and Mathematics with Applications* **26** (4) (1993) 43–55.

[6] Demetrovies J., Thi V. D., Armstrong relation, Functional dependencies and Strong Dependencies, *Computer and Artificial Intelligence* **14** (1995) 279–298.

[7] Demetrovies J., Thi V. D., Some results about normal forms for functional dependencies in the relational data model, *Discrete Applied Mathematics* **69** (1996) 61–74.

[8] Demetrovies J., Thi V. D., Describing candidate keys by hypergraphs, *Computer and Artificial Intelligence* **18** (1999) 191–207.

Nhận bài ngày 3 tháng 2 năm 2001
Nhận bài sau khi sửa ngày 10 tháng 4 năm 2001

Viện Công nghệ thông tin