

CÀI ĐẶT CÁC THUẬT TOÁN BẢO ĐẢM TÍNH NHẤT QUÁN DỮ LIỆU CHO MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ

PHẠM QUANG TRUNG & NGUYỄN XUÂN HUY

Viện Công nghệ Thông tin

Summary. Data integrity preservation is an important requirement in designing databases satisfied to build normal forms of relational schemes. All algorithms that translate a form of schema into a normal form require to define keys of the schema. Therefore a key finding problem is very important for designing database. There have been many solutions for this problem. In this paper, we present a solution that base on the concept of translation of relational schemes, which translates the prime schema into the class of schemes that made key finding problem simpler. Basis factors of relational schema are an amount of attributes, of functional dependencies. That has concern with a completeness of closure, key, and cover finding problem... The purpose of the translation to simplify the prime schema by rejecting some attributes, which are not belong to or are belong to all keys of prime schema is the result after adding with a set of attributes belong DABASCHE that is a solution of implementation of that algorithm. We present some important functions of the module DABASCHE for designing relational databases and it's basic data structures in the language that simulates the Pascal programming language.

I. MỞ ĐẦU

Việc đảm bảo tính nhất quán của dữ liệu là một yêu cầu quan trọng trong thiết kế cơ sở dữ liệu, đáp ứng yêu cầu xây dựng các lược đồ quan hệ (LĐQH) ở dạng chuẩn. Hầu hết các thuật toán đưa LĐQH về dạng chuẩn đều đòi hỏi phải xác định khoá của lược đồ. Do đó bài toán tìm tập hợp các khoá của một LĐQH để có thể xác định khoá chỉ định tối ưu nhất là có ý nghĩa. Đã có nhiều cách giải bài toán tìm khoá. Phương pháp biến đổi một LĐQH đưa về lớp lược đồ có cách giải bài toán tìm khoá đơn giản hơn được trình bày trong bài này. Những yếu tố cơ bản của LĐQH liên quan đến độ phức tạp của việc xác định bao đóng, tìm khoá, phủ tối thiểu ... là số các thuộc tính, và số lượng các phụ thuộc hàm. Với ý tưởng làm đơn giản hoá lược đồ ban đầu bằng cách loại bớt các thuộc tính không mang thông tin về khoá và cả các thuộc tính nằm trong mọi khoá, việc xác định các tập hợp khoá của lược đồ nhận được sẽ thuận lợi hơn, và với kết quả lý thuyết đã được chứng minh, tập khoá của LĐQH ban đầu chính là kết quả của phép phối hợp tập khoá đã nhận được với tập giao của các lược đồ ban đầu. Cùng với các cơ sở lý thuyết của phương pháp này, bài báo trình bày một giải pháp cài đặt nhằm cung cấp công cụ cho những người thiết kế cơ sở dữ liệu vận dụng phép biến đổi

lược đồ quan hệ để xác định tập hợp khóa của LĐQH, đó là module DABASCHE, bao gồm giới thiệu các chức năng chính và thông qua ngôn ngữ tựa Pascal thể hiện các cấu trúc dữ liệu cơ bản của module DABASCHE.

Cho $U = \{a_1, \dots, a_n\}$ là một tập hữu hạn khác trống, các ký hiệu a_i , $i = 1, \dots, n$, gọi là các thuộc tính. Với mỗi i , tập khác trống $\text{dom}(a_i)$ gọi là miền xác định của a_i . Kí hiệu tập hội với mọi i : $D = \cup_{i=1}^n \text{dom}(a_i)$. Một quan hệ R trên tập thuộc tính U là tập hữu hạn các ánh xạ $\{r_1, \dots, r_n\}$ từ tập U vào tập D sao cho với mọi $a_i \in U$ ta luôn có: $r(a_i) \in \text{dom}(a_i)$.

Các ánh xạ $r_i \in R$ được gọi là các bộ (tuples) của quan hệ R .

Các ký hiệu. Theo truyền thống trình bày của lý thuyết quan hệ, trong bài này sử dụng các ký hiệu sau:

- Quan hệ R trên tập thuộc tính U được kí hiệu là $R(U)$.
- Hợp của hai tập thuộc tính X, Y sẽ được viết là $X \cup Y$.
- Với quan hệ R trên tập thuộc tính U , r là một bộ của R , X là tập con của U thì giá trị của r trên X được viết là $r(X)$.

Cho tập hợp U và M, N là họ các tập con của U , tập $Z \subseteq U$. Ta định nghĩa phép toán \oplus như sau:

$$Z \oplus N = \{ZY \mid Y \in N\},$$

$$M \oplus N = \{XY \mid X \in M, Y \in N\}.$$

Cho $R(U)$ và hai tập thuộc tính $X, Y \subseteq U$. Ta nói rằng tập thuộc tính Y phụ thuộc hàm vào tập thuộc tính X và viết $X \rightarrow Y$ nếu: với mọi bộ r_1, r_2 trong quan hệ R từ: $r_1.X = r_2.X \Rightarrow r_1.Y = r_2.Y$.

Cho F là tập các phụ thuộc hàm đã biết trên quan hệ R , và cho $X \rightarrow Y$ là một phụ thuộc hàm. Ta nói $X \rightarrow Y$ là suy diễn được từ F nếu với mỗi quan hệ $R(U)$ thoả các phụ thuộc hàm của F thì cũng thoả phụ thuộc hàm $X \rightarrow Y$, kí hiệu $F \Rightarrow X \rightarrow Y$.

Định nghĩa. F^+ là bao đóng của tập phụ thuộc hàm là tập chứa tất cả phụ thuộc hàm có thể suy diễn được từ F : $F^+ = \{X \rightarrow Y \mid F \Rightarrow X \rightarrow Y\}$.

Lược đồ quan hệ $\omega = (U, F)$ là một cặp trong đó U là một tập hữu hạn các thuộc tính và F là tập hữu hạn các phụ thuộc hàm trên U . Cho lược đồ quan hệ (LĐQH) $\omega = (U, F)$ và tập $X \subseteq U$. Bao đóng của tập X dưới ω , kí hiệu X^+ là tập lớn nhất các thuộc tính $Y \subseteq U$ sao cho: $F \Rightarrow X \rightarrow Y$. (Kí hiệu X^+ được sử dụng nếu không có sự nhầm lẫn).

Việc kiểm chứng $F \Rightarrow X \rightarrow Y$ được đảm bảo qua bổ đề sau đây:

Bổ đề. $F \Rightarrow X \rightarrow Y \iff Y \subseteq X^+$.

Cho lược đồ quan hệ $\omega = (U, F)$, một tập con X của U là siêu khoá của lược đồ ω nếu $X \rightarrow U \in F^+$. Một tập con $K \subseteq U$ là một khoá của ω nếu K là một siêu khoá của ω và không tồn tại $K' \subset K$ nào sao cho $K' \rightarrow U \in F^+$.

Thuật toán tìm khoá cơ bản dựa vào ý tưởng xuất phát từ một siêu khoá, tiến hành loại dần các thuộc tính cho đến khi nhận được tập nhỏ nhất.

II. PHÉP BIẾN ĐỔI LỢC ĐỒ QUAN HỆ

Cho một LDQH $\omega = (U, F)$ và cho $Z \subseteq U$, đối với mỗi đối tượng E , (E có thể là tập các thuộc tính, tập các phụ thuộc hàm hoặc chính là lược đồ quan hệ), ta xây dựng đối tượng mới ký hiệu là $E \setminus Z$ bằng cách loại bỏ khỏi E các phần tử trong Z như sau:

- Trường hợp E là tập thuộc tính, ta định nghĩa $E \setminus Z$ là: đối với mỗi phụ thuộc hàm $f: X \rightarrow Y$ trong E thì $f \setminus Z: Z \setminus Z = (U \setminus Z, F \setminus Z)$.

Cho lược đồ quan hệ $\omega = (U, F)$, ký hiệu:

- Tập tất cả các khoá của ω được ký hiệu là K_ω .

- $\forall f \in F$, kí hiệu LEFT(f) và RIGHT(f) là vế trái và vế phải của f .

- Định nghĩa một số tập hợp như sau:

$I_\omega = \bigcap_{k \in K_\omega} K$ tập giao của tất cả các khoá.

$N_\omega = U \setminus \bigcup_{k \in K_\omega} K$ tập các thuộc tính không thuộc vào khoá nào / tập các thuộc tính không nguyên tố).

Định nghĩa phép biến đổi LDQH.

Một lược đồ $\omega' = (U', F')$ là kết quả của *phép biến đổi LDQH* $\omega = (U, F)$ trên Z nếu $Z \subseteq U$ và $\omega' = \omega \setminus Z$ nghĩa là $U' = U \setminus Z$ và $F' = F \setminus Z$.

Sau khi biến đổi, tập phụ thuộc hàm của ω' có thể chứa những phụ thuộc hàm dạng sau đây

$$- \Phi \rightarrow \Phi, \quad (1)$$

$$- C \rightarrow \Phi \quad (C \subseteq U \setminus Z; C \# \Phi), \quad (2)$$

$$- \Phi \rightarrow C \quad (C \subseteq U \setminus Z; C \# \Phi). \quad (3)$$

Theo định nghĩa của bao đóng, các phụ thuộc hàm dạng (1) và (2) không bổ xung các phần tử mới trong quá trình suy diễn, do đó về thực chất chúng không được sử dụng trong bất kỳ dãy suy diễn để tìm bao đóng nào, việc bỏ chúng đi sẽ không làm thay đổi các tập khoá $K_{\omega'}$ của lược đồ ω' . Vì vậy, chúng ta ngầm định rằng sau khi thực hiện phép biến đổi trong tập các phụ thuộc hàm của lược đồ nhận được sẽ không chứa các phụ thuộc hàm dạng (1) và (2).

Định lý cơ bản của phép biến đổi. Cho lược đồ quan hệ $\omega = (U, F)$ và cho $Z \subseteq U$. Nếu $\omega' = \omega \setminus Z = (U \setminus Z, F \setminus Z)$, thì $\forall X \subseteq U \setminus Z$ ta có: $(ZX)_{\omega'}^+ = Z(X)_{\omega}^+$. **Định lý.** Cho $\omega = (U, F)$ là một lược đồ quan hệ. Ta có $I_{\omega} = U \setminus \bigcup_{f \in F} (RIGHT(f) \setminus LEFT(f))$.

Bổ đề. Nếu M là siêu khoá của ω thì $M \setminus X$ là siêu khoá của $\alpha = \omega \setminus X$; ngược lại nếu N là siêu khoá của α thì NX là siêu khoá của ω .

Định lý. Cho $\omega = (U, F)$ là một lược đồ quan hệ và X là tập con của U . Cho $\alpha = \omega \setminus X$ thì:

- 1) $K_{\omega} = K_{\alpha} \iff X \subseteq N_{\omega}$.
- 1) $K_{\omega} = X \oplus K_{\alpha} \iff X \subseteq I_{\omega}$.

Hệ quả. Cho một lược đồ quan hệ $\omega = (U, F)$ và cho

$$U' = \bigcup_{f \in F} RIGHT(f) \setminus \bigcup_{f \in F} LEFT(f),$$

thì

- 1) $U' \subseteq N_{\omega}$.
- 2) Nếu $X \subseteq I_{\omega} \& Y \subseteq N_{\omega}$ thì $(XY)_{\omega}^+ \setminus X \subseteq N_{\omega}$.

Ứng dụng phép biến đổi trong việc tìm khoá. Cho lược đồ quan hệ $\omega = (U, F)$, việc tìm khoá của ω là bài toán NP đầy đủ [8], do đó chúng ta tiến hành biến đổi ω về một lớp lược đồ đặc biệt mà tập khoá của nó có thể được nhận biết dễ dàng hơn. Phương pháp giải quyết là tìm tập X lớn nhất $\subseteq I_{\omega}$ và tập Y lớn nhất $\subseteq N_{\omega}$, và tính $Z = X((XY)_{\omega}^+ \setminus X) = (XY)_{\omega}^+$, vận dụng phép biến đổi lược đồ quan hệ để chuyển lược đồ ω thành $\alpha = \omega \setminus Z$. So với lược đồ α có số lượng thuộc tính ít hơn, và số lượng phụ thuộc hàm cũng có thể ít hơn, do đó việc xác định khoá của α sẽ đơn giản hơn việc xác định khoá của ω . Khi đã có khoá của α ta dễ dàng khôi phục lại được khoá của ω bằng phép hợp khoá của α với X .

Thuật toán chuyển dịch lược đồ quan hệ

Cho lược đồ quan hệ $\omega = (U, F)$, với

$$U = \{a_1, \dots, a_n\},$$

$$F = \{LEFT(f) \rightarrow RIGHT(f) \mid f \in F\}.$$

Bước 1. Tìm tập X lớn nhất $\subseteq I_{\omega}$, theo định lý đã nêu thì I_{ω} dễ tính được, nên ta sẽ lấy $X = I_{\omega} = U \setminus \bigcup_{f \in F} (RIGHT(f) \setminus LEFT(f))$.

Bước 2. Tìm tập Y lớn nhất $\subseteq N_{\omega}$, theo hệ quả đã nêu thì ta có:

$U' = \bigcup_{f \in F} (RIGHT(f) \setminus LEFT(f)) \subseteq N_{\omega}$, và do chưa có công thức hiển để tính được N_{ω} nên ta lấy $Y = U'$.

Bước 3. Tính $Z = X((XY)_\omega^+ \setminus X) = (XY)_\omega^+$

Bước 4. Tiến hành phép biến đổi lược đồ quan hệ để chuyển lược đồ ω thành lược đồ $\alpha = \omega \setminus Z$.

Thí dụ. Cho lược đồ quan hệ $\omega = (U, F)$, với $U = abcdfghik$, và

$$\begin{aligned} F = \{ & ach \rightarrow b, \\ & bh \rightarrow acd, \\ & abci \rightarrow dk, \\ & adei \rightarrow bcg, \\ & cgi \rightarrow aek, \\ & h \rightarrow bc\}, \end{aligned}$$

Thực hiện giải thuật nêu trên ta có:

1) Tính tập X :

$$X = I_\omega = U \setminus \cup_{f \in F} (RIGHT(f) \setminus (LEFT(f))) = abcdeghik \setminus abcdegk = hi.$$

2) Tính

$$Y = U' = \cup_{f \in F} (RIGHT(f) \setminus (LEFT(f))) = abcdegk \setminus abcdeghi = k.$$

3) Tính

$$Z = X((XY)_\omega^+ \setminus X) = (XY)_\omega^+ = (I_\omega U')_\omega^+ = (hik)_\omega^+ = abcdhik.$$

4) Lập lược đồ quan hệ chuyển dịch: $\alpha = \omega \setminus Z$

$$\alpha = (U_{\alpha'}, F_\alpha), \text{ với } U_\alpha = U \setminus Z = abcdeghik \setminus abcdhik = eg,$$

$$F_\alpha = F \setminus Z = \{\Phi \rightarrow \Phi, \text{ loại}$$

$$\Phi \rightarrow \Phi, \text{ loại}$$

$$\Phi \rightarrow \Phi, \text{ loại}$$

$$e \rightarrow g,$$

$$g \rightarrow e,$$

$$\Phi \rightarrow \Phi, \text{ loại}$$

Vậy $\alpha = (U_{\alpha'}, F_\alpha)$ với $U_\alpha = eg$, và $F_\alpha = \{e \rightarrow g, g \rightarrow e\}$. Lược đồ α dễ nhận thấy là chỉ có hai khoá là e và g , tức là $\mathcal{K} = \{e, g\}$. Và ta có: $\mathcal{K}_\omega = I_\omega \oplus \mathcal{K}_\alpha = hi \oplus \{e, g\} = \{ehi, ghi\}$. Như vậy lược đồ ω đã cho có hai khoá $\{ehi\}$ và $\{ghi\}$.

III. THIẾT KẾ VÀ CÀI ĐẶT MODULE DABASCHE

Phần này trình bày phương pháp thiết kế và cài đặt module DABASCHE nhằm thể hiện các khái niệm lý thuyết đã trình bày trong các phần trước, đồng thời cung cấp một công cụ hữu hiệu cho những người thiết kế cơ sở dữ liệu.

3.1. Chức năng chính của module

1.1. Chức năng khai báo một lược đồ quan hệ.

1.1.1. Cập nhật các thuộc tính của lược đồ.

1.1.2. Cập nhật các phụ thuộc hàm của lược đồ.

1.2. Chức năng xác định bao đóng của một tập thuộc tính.

Cho phép người sử dụng tìm bao đóng của một tập thuộc tính nào đó trong lược đồ đã khai báo.

1.3. Chức năng xác định một khoá của lược đồ xuất phát từ một siêu khoá.

Chức năng này thể hiện thuật toán tìm khoá cơ bản dựa vào ý tưởng xuất phát từ một siêu khoá.

1.4. Chức năng thực hiện phép biến đổi lược đồ quan hệ theo một tập thuộc tính cho trước: Đây là một chức năng thuận tiện phục vụ cho việc biến đổi linh hoạt lược đồ đã cho theo một tập thuộc tính nào đó.

1.5. Chức năng thực hiện phép biến đổi lược đồ quan hệ.

Chức năng này thể hiện lý thuyết về phép biến đổi lược đồ quan hệ đã trình bày trong phần II.

3.2. Cấu trúc dữ liệu

Các đối tượng dữ liệu được chương trình xử lý là các thuộc tính, các phụ thuộc hàm, do đó tương ứng có hai dạng cấu trúc dữ liệu sau đây:

2.1. Cấu trúc dữ liệu lưu các thuộc tính.

Để lưu các thuộc tính chúng ta xây dựng một "cây tìm kiếm" có dạng sau:

```
TreeU = ^Tree1;
```

```
Tree1 = Record
```

```
  Name : String[30]; { tên thuộc tính }
```

```
  B : Byte ; { mã thuộc tính }
```

```
  Ord : Byte; { số thứ tự của thuộc tính }
```

```
  Left, Right : TreeU; { con trỏ trỏ đến nút của cây }
```

```
End;
```

2.2. Cấu trúc dữ liệu lưu các phụ thuộc hàm.

Cấu trúc danh sách liên kết kép để lưu các phụ thuộc hàm được xây dựng như sau:

```

SetByte = Set of byte;
Node = ^ Node1;
Node1 = Record
    L,R      : SetByte; { lưu về trái, về phải của phụ thuộc hàm }
    Pred, next : Node;

```

2.3. Một số đối tượng đặc biệt.

Vận dụng phương pháp lập trình hướng đối tượng, bộ chương trình đã xây dựng một số object phục vụ việc cập nhật các thuộc tính, các phụ thuộc hàm, cũng như tiến hành các xử lý khác rất hiệu quả và thuận tiện.

Sau đây là một đối tượng cơ bản App2 hậu bối từ kiểu App1 đã xây dựng trong chương trình:

```

App2 = Object(App1)

Constructor      Init;
Procedure        HandleEvent(Var Event: TEvent);Virtual;
Procedure        InitMenuBar;Virtual;
Procedure        UpdateA(Goc: TreeU; Attr: String[30];
                      Var Tap: SetByte; Them_Bot: Byte);
Procedure        UpdateFD;
Procedure        UpdateX(Var Tap: SetBByte);
Procedure        ViewFD(GocUa: TreeU; DauList: Node;
                      NameFile: String[8]; Var OFd, NFd: Byte;
                      Var WinCount: Integer);
Procedure        ViewSFD(GocUa: TreeU; DList: NodeG;
                      NameFile: String[8]; Tit: String;
                      Var OFd, NFd: Byte;
                      Var WinCount: Integer);
Procedure        View1Set(GocUa: TreeU; Tap: SetByte;
                      NameFile: String[8]; Tit1, Tit2: String;
                      Var OFd, NFd: Byte;
                      Var WinCount: Integer);
Procedure        ViewTran(GocUa: TreeU; Tap: SetByte;Dau: NodeG;
                      NameFile: String[8]; Tit1, Tit2, Tit3: String;
                      Var WinCount: Integer);
Procedure        InitStatusLine; Virtual;

```

End;

Các phương thức đầu Update: cập nhật các thuộc tính, các phụ thuộc hàm.

Các phương thức đầu View: hiển thị các kết quả xử lý.

IV. KẾT LUẬN

Phương pháp xử dụng phép biến đổi các lược đồ quan hệ để xác định khoá như đã giới thiệu trong các phần nêu trên là một phương pháp hiệu quả để xác định khoá của lược đồ. Và module DaBasche là một bộ chương trình cài đặt thuận tiện cho người thiết kế cơ sở dữ liệu.

TÀI LIỆU THAM KHẢO

1. Amstrong W.W., Dependency structures of database relationships (Information processing 74), North Holland Pub. Co. Amsterdam 1974.
2. Berri C., Berstein P.A., Computational problems related to the design of normal form relational schemes, ACM Translations on data base systems, v. 4 n. 1 1980.
3. Codd E.F., A Relational Model of Data for Large Shared Data Banks CACM 13 : 6 June 1970, 377-387.
4. Demetrovics J., Nguyen Xuan Huy, *Closed sets and translations of relation schemes*, Computer math. applic. v. 21, n. 1 1991, 13-23.
5. Horowitz E., Sahni S., Fundamentals of computer algorithms, Computer science press INC, Computer software engineering series, 1978.
6. Lê Văn Bào, Nguyễn Xuân Huy, Hồ Thuần, Demetrovics J., Lược đồ cân bằng và khoá của lược đồ quan hệ, Số đặc biệt về hai đề tài Nhà nước 48.05.08
7. Maier D. The theory of Relational Databases. Computer Science Press, Rockville, Md., 1983.
8. Niklaus w. Cấu trúc dữ liệu + Giải thuật = Chương trình, Nxb Thống kê, Hà nội, 1982 (Sách dịch).
9. Phạm Quang Trung, Một số vấn đề lý thuyết và lập trình cài đặt đảm bảo tính toàn vẹn của dữ liệu trong cơ sở dữ liệu quan hệ. (Luận văn Thạc sỹ tin học), Trường Đại học Bách khoa Hà nội, 1993.
10. Ullman J. D. Principles of Database Systems. 2nd. Computer Science Press. Potomac, Md, 1982.