

# Ngôn Ngữ Tân Từ

---

Trong CSDL thực hiện việc mô hình hóa thông tin gồm các sự kiện được liên kết hay biểu diễn một tình trạng thế giới thực. Ngôn ngữ tân từ ứng dụng logic toán để thể hiện truy vấn. Ngôn ngữ tân từ có hai loại, ngôn ngữ tân từ có biến là bộ và ngôn ngữ tân từ có biến là miền giá trị.

## 1. Ngôn ngữ tân từ có biến là bộ

### 1.1. Một số khái niệm

Dạng thức:  $\{t \mid P(t)\}$ , với:

- $t$ : biến bộ, nhận giá trị là một bộ của quan hệ. Khi đó  $t.A$  là giá trị của bộ  $t$  tại thuộc tính  $A$ .
- $P(t)$ : công thức liên quan đến biến bộ  $t$ , phụ thuộc vào giá trị của  $t$  mà  $P(t)$  cho kết quả đúng hay sai.
- Kết quả trả về là tập hợp các bộ  $t$  thỏa  $P(t)$

Ví dụ:

- Tìm tất cả những nhân viên nam

$$\{t \mid NHANVIEN(t) \wedge t.Phai = 'Nam'\}$$

- Tìm tất cả những nhân viên nữ và có lương trên 1500000

$$\{t \mid NHANVIEN(t) \wedge t.Phai = 'Nữ' \wedge t.Luong > 1500000\}$$

Với mỗi biến bộ  $t$ , quan hệ  $R$  mà  $t$  biến thiên trên đó được gọi là quan hệ vùng của biến bộ. Giá trị này được chỉ định bằng điều kiện dạng  $R(t)$ .

### 1.2. Định nghĩa hình thức của phép tính bộ

Dạng tổng quát:  $\{t_1.A_1, t_2.A_2, \dots, t_n.A_n \mid P(t_1, t_2, \dots, t_n, t_{n+1}, \dots, t_{n+m})\}$ , với:

- $t_1, t_2, \dots, t_n, t_{n+1}, \dots, t_{n+m}$  là các biến bộ
- $A_i$  là thuộc tính của quan hệ mà biến bộ  $t_i$  biến thiên
- $P$  là công thức hay điều kiện của phép tính bộ.
- Công thức được hình thành từ các công thức nguyên tố

### Công thức nguyên tố:

Có 3 dạng:

- $R(t_i)$  là công thức nguyên tố, với
  - $R$  là một quan hệ và  $t_i$  là một biến bộ
  - Công thức này xác định vùng của biến bộ  $t_i$  dưới hình thức quan hệ có tên  $R$
  - Ví dụ: DEAN( $t$ ), NHANVIEN( $x$ )
- $t_i.A \theta t_j.B$  là công thức nguyên tố, với:
  - $\theta$  là phép so sánh ( $<, >, =, \leq, \geq, \neq$ )
  - $t_i, t_j$  là các biến bộ
  - $A$  là thuộc tính của quan hệ trên đó bộ  $t_i$  biến thiên,  $B$  là thuộc tính của quan hệ trên đó bộ  $t_j$  biến thiên
  - Ví dụ:  $x.MaPhong = y.Phong$ , với PHONGBAN( $x$ ) và DEAN( $y$ )
- $t_i.A \theta c$  hoặc  $c \theta t_j.B$  là công thức nguyên tố, với:
  - $\theta$  là phép so sánh ( $<, >, =, \leq, \geq, \neq$ )
  - $t_i, t_j$  là các biến bộ
  - $A$  là thuộc tính của quan hệ trên đó bộ  $t_i$  biến thiên,  $B$  là thuộc tính của quan hệ trên đó bộ  $t_j$  biến thiên
  - $c$  là giá trị hằng
  - Ví dụ:  $x.MaPhong = PH01$ ,  $y.Luong > 500000$

Nhận thấy rằng mỗi công thức nguyên tố đều mang giá trị đúng hoặc sai. Với (a),  $R(t)$  đúng nếu  $t$  là một bộ thuộc  $R$ , ngược lại mang giá trị sai; với (b, c) giá trị đúng sai phụ thuộc vào kết quả thay thế giá trị thực sự của bộ vào vị trí biến bộ.

Một công thức (hay điều kiện) được hình thành từ một hay nhiều công thức nguyên tố và được nối với nhau bằng các toán tử và, hoặc, phủ định ( $\wedge, \vee, \neg$ ) và được định nghĩa như sau:

- a. Mọi công thức nguyên tố là công thức
- b. Nếu  $F_1$  và  $F_2$  là các công thức thì  $(F_1 \wedge F_2), (F_1 \vee F_2), \neg F_1, \neg F_2$  là công thức và có giá trị:
  - $(F_1 \wedge F_2)$  chỉ đúng khi cả  $F_1$  và  $F_2$  đều đúng
  - $(F_1 \vee F_2)$  chỉ sai khi cả  $F_1$  và  $F_2$  đều sai
  - $\neg F_1$  đúng nếu  $F_1$  sai,  $\neg F_1$  sai nếu  $F_1$  đúng

### 1.3. Lượng từ tồn tại $\exists$ và với mọi $\forall$

#### Biến tự do và biến kết buộc

Một biến bộ  $t$  là *kết buộc* nếu có kèm lượng từ, nghĩa là nó xuất hiện trong mệnh đề  $\forall t \in S(P(t))$  hay  $\exists t \in S(P(t))$ , ngược lại nó được gọi là *biến tự do*.

Nếu  $F$  là công thức nguyên tố thì mọi biến bộ  $t$  trong  $F$  đều là biến tự do.

Tất cả các biến bộ tự do  $t$  trong  $F$  được xem là biến kết buộc trong công thức  $F' = (\forall t)F$  hay  $F' = (\exists t)F$

Đối với các công thức  $F = (F_1 \wedge F_2), F = (F_1 \vee F_2), F = \neg F_1, F = \neg F_2$ , biến  $t$  là tự do hay kết buộc phụ thuộc vào nó là tự do hay kết buộc trong  $F_1$  và  $F_2$

Biến bộ tự do chỉ ra các bộ mà câu truy vấn trả về, nghĩa là được sử dụng trong vế trái. Ngược lại biến bộ kết buộc thường được sử dụng để thực hiện việc khẳng định các bộ trong CSDL, nghĩa là được sử dụng trong vế phải.

Ví dụ:

- Cho biết những nhân viên thuộc phòng có mã số 5 và có lương  $\geq 500000$

$$\{n \mid NHANVIEN(n) \wedge n.Phong = 5 \wedge n.Luong \geq 500000\}$$

- Cho biết những đề án do phòng ‘Quản Lý’ phụ trách

$$\{d \mid DEAN(d) \wedge (\exists p \in PHONGBAN(d.Phong = p.MaPhong \wedge p.TenPhong = 'QuanLy'))\}$$

- Tìm những nhân viên tham gia trong tất cả các đề án của công ty

$$\{n \mid NHANVIEN(n) \wedge (\forall d \in DEAN(\exists p \in PHANCONG(p.MaDA = d.MaDA \wedge p.MaNV = n.MaNV))))\}$$

Chú ý rằng

- $\forall t(F)$  đúng nếu F đúng với mọi bộ t, sai nếu ít nhất một bộ sai.
- $\exists t(F)$  sai nếu F sai với mọi bộ t, đúng nếu ít nhất một bộ đúng.

## 2. Ngôn ngữ tân từ có biến là miền giá trị

Biến miền giá trị nhận giá trị từ miền giá trị của một thuộc tính.

Dạng thức:  $\{x_1, x_2, \dots, x_n \mid P\}$ , với:

- $x_1, x_2, \dots, x_n$  là danh sách các biến miền giá trị
- P là điều kiện hay công thức theo  $x_1, x_2, \dots, x_n$
- Kết quả câu truy vấn là tập hợp các chọn lựa của các bộ  $x_1, x_2, \dots, x_n$  sao cho, với mọi i, giá trị  $x_i$  được thay thế cho các biến tự do  $x_i$  thì điều kiện đúng. Biến  $x_i$  có thể là hằng số, khi đó tất cả các bộ trong tập hợp đều là hằng số trong vị trí i.

Ví dụ:

- Cho biết mã nhân viên thuộc phòng có mã số 5 và có lương  $\geq 500000$

$$\left\{ Ma, Ho, Ten \mid \exists Phg \exists Lg \left( \begin{array}{l} NHANVIEN(Ma, Ho, Ten, NSinh, DChi, Phai, Lg, MaQL, Phg) \\ \wedge Phg = 5 \wedge Lg \geq 500000 \end{array} \right) \right\}$$

Nhận thấy rằng  $Ma, Ho, Ten$  chính là thuộc tính được yêu cầu;  $Phg, Lg$  là biến thực sự xuất hiện trong một câu điều kiện.

- Tìm họ tên, địa chỉ của nhân viên thuộc phòng ‘Quản Lý’

$$\left\{ Ma, Ho, Ten, DChi \mid \exists Phg \left( \begin{array}{l} NHANVIEN(Ma, Ho, Ten, NSinh, DChi, Phai, Lg, MaQL, Phg) \\ \wedge \exists TenPhg (MaPhg, TenPhg, TrPhg, NgayNC) \wedge TenPhg = 'Quản Lý' \end{array} \right) \right\}$$

## Công thức nguyên tố:

Có 3 dạng:

- a.  $R(x_1, x_2, \dots, x_j)$  là công thức nguyên tố, với
- $R$  là một quan hệ bậc  $j$
  - Mỗi  $x_i, 1 \leq i \leq n$  là một biến miền.
  - Công thức này cho biết  $(x_1, x_2, \dots, x_j)$  là một bộ của quan hệ  $R$ , với  $x_i$  là giá trị thứ  $i$  của bộ.
  - Ví dụ:  $\{x_1, x_2, \dots, x_n \mid R(x_1, x_2) \wedge \dots\}$
- b.  $x_i \theta x_j$  là công thức nguyên tố, với:
- $\theta$  là phép so sánh ( $<, >, =, \leq, \geq, \neq$ )
  - $x_i, x_j$  là các biến miền
  - Ví dụ:  $\text{MaPhg} = \text{Phg}$ .
- c.  $x_i \theta c$  hoặc  $c \theta x_j$  là công thức nguyên tố, với:
- $\theta$  là phép so sánh ( $<, >, =, \leq, \geq, \neq$ )
  - $x_i, x_j$  là các biến miền
  - $c$  là hằng
  - Ví dụ:  $\text{MaPhg} = 1$ .

Một công thức nguyên tố có trị đúng hoặc sai với một tập giá trị cụ thể tương ứng với các biến miền.

Các định nghĩa về công thức dựa trên công thức nguyên tố, các định nghĩa về biến kết buộc và tự do, các lượng từ trong trường hợp phép tính miền cũng tương tự phép tính bộ.

### 3. Bài tập

#### Bài 1:

Hãy viết bằng ngôn ngữ tân từ theo yêu cầu như bài tập 1 trong chương 4.

#### Bài 2:

Hãy viết bằng ngôn ngữ tân từ theo yêu cầu như bài tập 2 trong chương 4.

# ***Ngôn Ngữ Truy Vấn SQL***

---

SQL (Structured Query Language) là ngôn ngữ CSDL quan hệ chuẩn dành cho các CSDL quan hệ thương mại.

Từ những năm 70, SQL được phát triển từ IBM với hệ quản trị CSDL quan hệ SYSTEM-R với ngôn ngữ giao tiếp CSDL là Sequel (Structured English Query Language).

Năm 1986, phiên bản chuẩn được chấp nhận bởi ANSI (American National Standards Institute) gọi là SQL-86 hay SQL1 và năm 1987 được chấp nhận bởi ISO (International Standards Organization). Từ đó các phiên bản đã được đưa ra vào 1992, 1999, 2003.

## **1. Các lệnh hỏi**

Các lệnh hỏi hay còn được gọi là truy vấn rút trích dữ liệu. Lệnh SELECT là lệnh cơ bản để rút trích thông tin từ CSDL.

Chú ý rằng lệnh SELECT không hoàn toàn giống như phép toán chọn trong đại số quan hệ, SQL cho phép một bảng có hai hay nhiều bộ có giá trị giống nhau trên mọi thuộc tính cùng tồn tại

### **1.1. Cú pháp lệnh truy vấn**

Cú pháp lệnh truy vấn:

```
SELECT <danh sách thuộc tính>  
FROM <danh sách các bảng>  
WHERE <điều kiện>  
GROUP BY <các thuộc tính gom nhóm>  
HAVING <điều kiện gom nhóm>  
ORDER BY <danh sách thuộc tính>
```

với:

- **SELECT** <danh sách thuộc tính>: tên các thuộc tính được lấy giá trị
- **FROM** <danh sách các bảng>: tên các bảng cần để xử lý câu truy vấn
- **WHERE** <điều kiện>: biểu thức điều kiện chọn và điều kiện kết các bộ trong các quan hệ được chỉ ra trong mệnh đề FROM
- **GROUP BY** <các thuộc tính gom nhóm>: chỉ ra các thuộc tính gom nhóm
- **HAVING** <điều kiện gom nhóm>: chỉ ra điều kiện để trích chọn các nhóm
- **ORDER BY** <danh sách thuộc tính>: thứ tự hiển thị kết quả câu truy vấn

Trong câu lệnh, **SELECT** và **FROM** là bắt buộc, những thành phần còn lại có thể không có.

## 1.2. Phép chiếu

Sử dụng mệnh đề **SELECT**, kết quả gần giống với phép chiếu của đại số quan hệ, liệt kê các thuộc tính cần hiển thị trong kết quả câu truy vấn.

Ví dụ:

- Tìm mã nhân viên, họ tên tất cả các nhân viên

```
SELECT MaNV, HoNV, TenNV
FROM NHANVIEN
```

- Tìm mã đề án, tên các đề án

```
SELECT MaDA, TenDA
FROM DEAN
```

## 1.3. Phép chọn

Sử dụng mệnh đề **WHERE**, kết quả gần giống với phép chọn của đại số quan hệ, nêu điều kiện liên quan đến thuộc tính của quan hệ xuất hiện trong mệnh đề FROM. Thường sử dụng **AND**, **OR**, **NOT**, **BETWEEN**, các phép toán so sánh.

Ví dụ:

- Tìm mã nhân viên, họ tên tất cả các nhân viên nam

```
SELECT MaNV, HoNV, TenNV
FROM NHANVIEN
```

WHERE Phai='Nam'

#### 1.4. Phép kết

Trong mệnh đề WHERE thường có điều kiện kết nếu như trong mệnh đề FROM có nhiều hơn hai quan hệ.

Ví dụ:

- Tìm mã nhân viên, họ tên tất cả các nhân viên tham gia vào đề án có mã 'DA01'

```
SELECT MaNV, HoNV, TenNV
```

```
FROM NHANVIEN, PHANCONG
```

```
WHERE NHANVIEN.MaNV= PHANCONG.MaNV AND MaDA='DA01'
```

- Tìm mã nhân viên, họ tên tất cả các nhân viên làm việc tại phòng 'Tài chính'

```
SELECT MaNV, HoNV, TenNV
```

```
FROM NHANVIEN, PHONGBAN
```

```
WHERE Phong=MaPhong AND TenPhong = 'Tài chính'
```

#### 1.5. Một số lưu ý

##### Sử dụng \*

Khi cần lấy thông tin về tất cả các cột của bảng thì sử dụng dấu sao (\*) thay vì phải liệt kê mọi tên thuộc tính.

Ví dụ:

```
SELECT *
```

```
FROM NHANVIEN
```

##### Tên bí danh

Khi câu truy vấn cần tham chiếu tới cùng một quan hệ 2 lần thì dùng bí danh cho tên quan hệ. SQL cho phép đổi tên quan hệ và tên thuộc tính được thực hiện bằng AS (có thể dùng hoặc không) theo quy tắc: *Tên cũ [as] tên mới*

Ví dụ:

```
SELECT MaNV, HoNV as Ho, TenNV as Ten
```

```
FROM NHANVIEN
```



WHERE Phai='Nam'

### **Biểu thức trong mệnh đề SELECT**

Tên thuộc tính có thể kèm theo tên bảng nếu cần làm rõ bằng cách thêm dấu chấm (.) trước tên thuộc tính. Ví dụ: NHANVIEN.MaNV.

Câu lệnh SELECT còn cho phép thực hiện tính toán theo công thức dựa trên các cột của bảng.

Ví dụ:

- Tìm mã nhân viên, họ tên và lương nhân viên, trong đó lương được tăng thêm 20% cho mọi nhân viên

```
SELECT MaNV, HoNV, TenNV, Luong*1.2
```

```
FROM NHANVIEN
```

### **DISTINCT**

Như đã trình bày ở trên, SQL cho phép các bộ trùng nhau trong kết quả câu truy vấn. Để loại bỏ các bộ trùng nhau, sử dụng từ khóa DISTINCT sau SELECT.

Ví dụ:

- Cho biết những mức lương hiện có trong công ty

```
SELECT DISTINCT Luong
```

```
FROM NHANVIEN
```

### **Phép so sánh trên chuỗi**

Sử dụng LIKE để so sánh chuỗi, có 2 ký tự đặc biệt sau:

- % (hoặc \*): thay thế cho mọi ký tự bất kỳ
- \_ (hoặc ?) : thay thế cho một ký tự bất kỳ

Ví dụ:

- Tìm tất cả những nhân viên có tên bắt đầu bằng M như Mai, Minh,...

```
SELECT MaNV, HoNV, TenNV
```

```
FROM NHANVIEN
```

```
WHERE TenNV LIKE 'M%'
```

## Thứ tự hiển thị

Mệnh đề ORDER BY sử dụng để sắp xếp các bộ trong kết quả câu truy vấn dựa trên giá trị các thuộc tính. Trong đó:

- Asc: sắp theo thứ tự tăng dần (mặc định)
- Desc: sắp theo thứ tự giảm dần

Ví dụ:

- Tìm tất cả những nhân viên phòng 1 và lương tương ứng, sắp xếp giảm dần theo lương

```
SELECT MaNV, HoNV, TenNV, Luong
FROM NHANVIEN
WHERE Phong = 1
ORDER BY Luong desc
```

## 2. Câu truy vấn lồng

Ở các ví dụ đã trình bày, mệnh đề WHERE bao gồm thuộc tính đơn, các phép so sánh hằng số. Phần này giới thiệu câu truy vấn lồng, cho phép lồng các câu truy vấn lại với nhau. Khi đó câu truy vấn con thông thường được sử dụng để kiểm tra các tập hợp thành viên, so sánh tập hợp, kiểm tra sự tồn tại.

Khi sử dụng truy vấn con trong mệnh đề WHERE của một truy vấn khác, mệnh đề SELECT trong truy vấn con phải phù hợp với số thuộc tính và kiểu dữ liệu của mệnh đề WHERE trong câu truy vấn ngoài.

Truy vấn con trả về giá trị tập hợp trong mệnh đề WHERE có dạng:

- <biểu thức> [NOT] IN (<truy vấn con>)
- <biểu thức> <phép toán so sánh> ANY (<truy vấn con>)
- <biểu thức> <phép toán so sánh> ALL (<truy vấn con>)

Kiểm tra sự tồn tại:

- [NOT] EXISTS (<truy vấn con>)

Các câu truy vấn con trong một mệnh đề WHERE có thể được kết hợp bằng cách sử dụng các phép nối logic

Ví dụ:

- Tìm tất cả những nhân viên nam làm việc trong phòng ‘Tài chính’

```
SELECT MaNV, HoNV, TenNV
FROM NHANVIEN
WHERE Phong IN (SELECT MaPhong as Phong
                FROM PHONGBAN
                WHERE TenPhong = ‘Tài chính’)
AND Phai = ‘Nam’
```

Trong trường hợp điều kiện ở mệnh đề WHERE của câu truy vấn con tham chiếu tới một thuộc tính của quan hệ được khai báo trong truy vấn cha thì hai câu truy vấn được gọi là tương quan. Các tham chiếu tới các quan hệ và các thuộc tính cha xuất hiện thông qua việc sử dụng bí danh.

Ví dụ:

- Tìm tất cả những nhân viên có cùng tên với người thân

```
SELECT nv.MaNV, nv.HoNV, nv.TenNV
FROM NHANVIEN as nv
WHERE nv.MaNV IN (SELECT MaNV
                  FROM THANNHAN
                  WHERE TenTN = nv.TenNV)
```

- Tìm tất cả những nhân viên có lương lớn hơn mọi nhân viên phòng số 5

```
SELECT MaNV, HoNV, TenNV
FROM NHANVIEN
WHERE Luong > ALL (SELECT Luong
                   FROM NHANVIEN
                   WHERE Phong = 5)
```

- Tìm tất cả những nhân viên có lương lớn hơn ít nhất 1 nhân viên phòng số 5

```
SELECT nv1.MaNV, nv1.HoNV, nv1.TenNV
FROM NHANVIEN as nv1, NHANVIEN as nv2
WHERE nv1.Luong > nv2.Luong AND nv2.Phong = 5
```

Cụm từ lớn hơn ít nhất một có thể diễn đạt bằng >SOME. Ví dụ trên được viết lại như sau:

```
SELECT MaNV, HoNV, TenNV
FROM NHANVIEN
WHERE Luong > SOME (SELECT Luong
                     FROM NHANVIEN
                     WHERE Phong = 5)
```

Hàm EXISTS sử dụng để kiểm tra kết quả của câu truy vấn tương quan có rỗng hay không. Khi đó

- EXISTS (r) mang giá trị true nếu có ít nhất một bộ trong r, false nếu ngược lại.
- NOT EXISTS (r) mang giá trị true nếu không có bộ nào trong r, false nếu ngược lại.

Ví dụ

- Cho biết những nhân viên có cùng tên với thân nhân

```
SELECT nv.MaNV, nv.HoNV, nv.TenNV
FROM NHANVIEN nv
WHERE EXISTS (SELECT *
              FROM THANNHAN
              WHERE MaNV = nv.MaNV AND TenTN= nv.TenNV)
```

### Giá trị null

SQL cho phép sử dụng các giá trị null để chỉ ra giá trị của thuộc tính không biết hay không tồn tại.

Chú ý rằng kết quả của điều kiện ở mệnh đề WHERE là false nếu nó liên quan đến null.

Trong SQL sử dụng IS NULL hay IS NOT NULL để kiểm tra các giá trị rỗng. Trong đó:

- Phép so sánh bằng (=) không dùng được
- Các hàm gom nhóm ngoại trừ COUNT() bỏ qua các giá trị null trong tập các dữ liệu đầu vào
- Nếu có điều kiện kết thì các bộ có giá trị null trên thuộc tính kết sẽ không có trong kết quả.

### 3. Hàm kết hợp và gom nhóm

#### Hàm kết hợp

Hàm kết hợp có đầu vào là một tập giá trị và trả về một giá trị đơn

- Avg(): giá trị trung bình
- Min(): giá trị nhỏ nhất
- Max(): giá trị lớn nhất
- Sum(): tính tổng
- Count(): đếm số mẫu tin

Ví dụ

- Cho biết những mức lương trung bình và cao nhất của các nhân viên phòng có mã là 5

```
SELECT AVG(Luong), MAX(Luong)
FROM NHANVIEN
WHERE Phong = 5
```

#### Gom nhóm

Các hàm gom nhóm được áp dụng trên các nhóm bộ cùng thuộc tính trong quan hệ. Mỗi nhóm bộ bao gồm tập hợp các bộ có cùng giá trị trên các thuộc tính gom nhóm. Trong SQL sử dụng cú pháp như sau:

```
SELECT <danhsách thuộc tính[với hàm kết hợp]>
FROM <danhsách các bảng>
WHERE <điều kiện>
GROUP BY <các thuộc tính gom nhóm>
```

HAVING <điều kiện gom nhóm>

- Mệnh đề GROUP BY chỉ ra các thuộc tính gom nhóm, các thuộc tính trong mệnh đề SELET nằm ngoài một hàm kết hợp phải xuất hiện trong mệnh đề GROUP BY
- Mệnh đề HAVING lấy các giá trị của hàm gom nhóm chỉ trên những nhóm nào thỏa điều kiện nhất định. Mệnh đề HAVING chỉ ra điều kiện lọc trên các nhóm, không phải điều kiện lọc trên từng bộ.

Ví dụ

- Cho biết tên phòng và lương trung bình của các nhân viên trong phòng ban lớn hơn 100000.

```
SELECT TenPhong, AVG(Luong)
FROM NHANVIEN, PHONGBAN
WHERE Phong = MaPhong
GROUP BY TenPhong
HAVING AVG(Luong)>100000
```

#### 4. Các lệnh khai báo cấu trúc CSDL

##### Kiểu dữ liệu

Chọn hệ quản trị CSDL SQL Server để minh họa, Ta có một số kiểu dữ liệu cơ bản:

- Số (number)
- Chuỗi (text, char, varchar, nvarchar)
- Ngày tháng (datetime)
- Logic (bit)

##### Tạo bảng

CREATE TABLE <tên bảng>

(

<thuộc tính 1> <kiểu dữ liệu> [not null][unique] [<RBTV thuộc tính>],  
<thuộc tính 2> <kiểu dữ liệu> [not null][unique] [<RBTV thuộc tính>],

```

...
<thuộc tính n> <kiểu dữ liệu> [not null][unique] [<RBTV thuộc tính>],
[<RBTV bảng>]
)

```

Các thuộc tính được xếp theo thứ tự khi tạo bảng. Các ràng buộc cũng có thể được bổ sung sau bằng cách dùng ALTER TABLE

Ví dụ

```

CREATE TABLE PHONGBAN
(
    MaPhong    char(5)      not null,
    TenPhong   nvarchar(30),
    TruongPhong char(5),
    NgayNhanChuc  datetime,
    PRIMARY KEY (MaPhong),
    UNIQUE (TenPhong)
    FOREIGN KEY (TruongPhong) REFERENCES NHANVIEN (MaNV)
)

```

## **RBTV**

Một số RBTV trong khi tạo bảng:

- Not null: không được chứa giá trị null
- Khóa chính: không được chứa giá trị null và được xác định bởi PRIMARY KEY <thuộc tính>
- Khóa ngoại: FOREIGN KEY <thuộc tính> REFERENCE <quan hệ> <thuộc tính>
- Tính duy nhất: có thể chứa giá trị null và được xác định bởi UNIQUE <thuộc tính>
- CHECK <điều kiện>: điều kiện đơn giản, không chứa các câu truy vấn hay tham chiếu tới các quan hệ khác

- Trong SQL server, mỗi RBTV có thể được đặt tên bằng cách sử dụng CONSTRAINT <tên ràng buộc> <kiểu ràng buộc>. Chú ý rằng tên ràng buộc phải duy nhất trong một lược đồ CSDL.

Ví dụ:

```
CREATE TABLE PHONGBAN
(
    MaPhong    char(5)        not null,
    TenPhong   nvarchar(30),
    TruongPhong char(5),
    NgayNhanChuc    datetime,
    CONSTRAINT PK_PHONGBAN PRIMARY KEY (MaPhong),
    CONSTRAINT U_PHONGBAN UNIQUE (TenPhong)
    CONSTRAINT PK_PHONGBAN FOREIGN KEY (TruongPhong)
    REFERENCES NHANVIEN (MaNV)
)
```

### **Xóa bảng**

DROP TABLE <tên bảng> xóa bảng

Ví dụ DROP TABLE PHONGBAN

### **Thay đổi cấu trúc**

Thêm cột

```
ALTER TABLE NHANVIEN ADD ChuyenMon    char(40)
```

Xóa cột

```
ALTER TABLE NHANVIEN DROP ChuyenMon
```

Bổ sung, thay đổi RBTV

```
ALTER TABLE NHANVIEN DROP CONSTRAINT FK_NHANVIEN
```

```
ALTER TABLE NHANVIEN ADD CONSTRAINT FK_NHANVIEN FOREIGN
    KEY (MaNQL) REFERENCES NHANVIEN (MaNV)
```



## 5. Các thao tác cập nhật dữ liệu

### 5.1. Thêm

Có thể thêm một bộ vào bảng bằng cách sử dụng:

```
INSERT INTO <bảng> [<thuộc tính 1>, <thuộc tính 2>, ..., <thuộc tính n>] VALUES  
(<giá trị 1>, <giá trị 2>, ..., <giá trị n>)
```

Chú ý rằng thứ tự giá trị trong VALUES là thứ tự các thuộc tính được chỉ ra trong CREATE TABLE

Ví dụ

```
INSERT INTO PHANCONG VALUES ('NV01', 'DA01', 10 )
```

Có thể thêm nhiều bộ vào bảng bằng cách sử dụng:

```
INSERT INTO <bảng> [<thuộc tính 1>, <thuộc tính 2>, ..., <thuộc tính n>]
```

```
SELECT ... FROM... WHERE...
```

### 5.2. Xóa

Có thể xóa một hay nhiều bộ khỏi bảng bằng cách sử dụng

```
DELETE FROM <bảng>
```

```
[WHERE <điều kiện>]
```

Ví dụ

- Xóa những nhân viên có mức lương dưới 100000

```
DELETE FROM NHANVIEN
```

```
WHERE Luong<100000
```

- Xóa những nhân viên làm việc cho phòng 'Nghiên cứu'

```
DELETE FROM NHANVIEN
```

```
WHERE Phong in (SELECT MaPhong
```

```
FROM PHONGBAN
```

```
WHERE TenPhong = 'Nghiên cứu')
```

Chú ý rằng các bộ trong bảng khác có thể bị xóa do ràng buộc tham chiếu. Để giải quyết vấn đề này có thể không cho xóa hoặc xóa luôn những bộ đang tham chiếu đến.

### 5.3. Cập nhật

Cập nhật các giá trị thuộc tính của một hay nhiều bộ bằng cách sử dụng

UPDATE <bảng>

SET <thuộc tính 1>= <giá trị 1>,

<thuộc tính 2>= <giá trị 2>,

...

<thuộc tính n>= <giá trị n>

[WHERE <điều kiện>]

Các bộ thỏa điều kiện sẽ được cập nhật giá trị cho các thuộc tính. Chú ý rằng các bộ trong bảng khác có thể được cập nhật do ràng buộc tham chiếu. Để giải quyết vấn đề này có thể không cho thay đổi hoặc thay đổi luôn những giá trị tham chiếu đến.

Ví dụ

Tăng thêm 100000 cho các nhân viên phòng ‘Nghiên cứu’

UPDATE NHANVIEN

SET Luong = Luong + 100000

WHERE Phong IN (SELECT MaPhong

FROM PHONGBAN

WHERE TenPhong = ‘Nghiên cứu’)

## 6. Bài tập

Thực hiện các câu truy vấn trong bài tập chương 4.

## Chương 7

# *Phụ thuộc hàm, khóa và ràng buộc toàn vẹn của lược đồ quan hệ*

---

Phụ thuộc hàm (functional dependency) dùng để biểu diễn một cách hình thức các ràng buộc toàn vẹn (RBTV). Phụ thuộc hàm có tầm quan trọng rất lớn trong việc giải quyết các bài toán tìm khóa, phủ tối thiểu và chuẩn hóa cơ sở dữ liệu. Nội dung chương cũng trình bày ràng buộc toàn vẹn (RBTV), các yếu tố liên quan đến ràng buộc toàn vẹn nhằm bảo đảm tính đúng đắn của dữ liệu.

### 1. Phụ thuộc hàm

#### 1.1. Khái niệm phụ thuộc hàm

Xét quan hệ DEAN(MaDA, TenDA, DdiemDA, Phong). Nhận thấy rằng quan hệ DEAN có MaDA là khóa, nghĩa là từ MaDA có thể xác định được tất cả các thông tin về tên đề án, địa điểm thực hiện đề án và phòng ban chủ trì đề án. Có thể phát biểu lại như sau:

- MaDA xác định TenDA hay TenDA phụ thuộc hàm vào MaDA
- MaDA xác định DdiemDA hay DdiemDA phụ thuộc hàm vào MaDA
- MaDA xác định Phong hay Phong phụ thuộc hàm vào MaDA

được ký hiệu:

- $\text{MaDA} \rightarrow \text{TenDA}$
- $\text{MaDA} \rightarrow \text{DdiemDA}$
- $\text{MaDA} \rightarrow \text{Phong}$

Một cách tổng quát, ta có:

#### Định nghĩa phụ thuộc hàm

$Q(A_1, A_2, \dots, A_n)$  là quan hệ;  $Q^+ = \{A_1, A_2, \dots, A_n\}$ ; X, Y là hai tập con của  $Q^+$ ;

$t_1, t_2$  là hai bộ bất kỳ của Q.

Khi đó:  $X \rightarrow Y \Leftrightarrow (t_1.X = t_2.X \Rightarrow t_1.Y = t_2.Y)$

Ta nói  $X$  xác định  $Y$  hay  $Y$  phụ thuộc hàm vào  $X$ .  $X$  được gọi là vế trái phụ thuộc hàm,  $Y$  được gọi là vế phải phụ thuộc hàm.

### Phụ thuộc hàm hiển nhiên

$X \rightarrow Y$  gọi là phụ thuộc hàm hiển nhiên nếu  $Y \subseteq X$ .

### Phụ thuộc hàm nguyên tố

$X \rightarrow Y$  được gọi là phụ thuộc hàm nguyên tố (hoặc nói cách khác  $Y$  được gọi là phụ thuộc đầy đủ vào  $X$ ) nếu  $\forall X' \subsetneq X$  đều không có phụ thuộc hàm  $X' \rightarrow Y$ .

Như vậy các phụ thuộc hàm  $\text{MaDA} \rightarrow \text{TenDA}$ ,  $\text{MaDA} \rightarrow \text{DdiemDA}$ ,  $\text{MaDA} \rightarrow \text{Phong}$  là phụ thuộc hàm nguyên tố.

Xét quan hệ CTIETHOADON (SoHD, MaHang, SoLuong, DonGia, ThanhTien) và các phụ thuộc hàm như sau:

- $\{\text{SoHD}, \text{MaHang}\} \rightarrow \text{SoLuong}$
- $\{\text{SoHD}, \text{MaHang}\} \rightarrow \text{DonGia}$
- $\{\text{SoHD}, \text{MaHang}\} \rightarrow \text{ThanhTien}$

Nhận thấy rằng SoLuong phụ thuộc đầy đủ vào  $\{\text{SoHD}, \text{MaHang}\}$  nhưng DonGia chỉ phụ thuộc vào MaHang (là một thuộc tính khóa) chứ không phụ thuộc đầy đủ vào khóa  $\{\text{SoHD}, \text{MaHang}\}$ . Như vậy, trên một lược đồ quan hệ có thể tồn tại nhiều phụ thuộc hàm.

Tập các phụ thuộc hàm được ký hiệu  $F$

### 1.2. Hệ luật dẫn Amstrong

Gọi  $F$  là tập phụ thuộc hàm đối với lược đồ quan hệ  $R$  định nghĩa trên tập thuộc tính  $U$  và  $X \rightarrow Y$  là một phụ thuộc hàm;  $X, Y \subseteq U$ . Khi đó ta nói rằng  $X \rightarrow Y$  được suy diễn logic từ  $F$  nếu  $R$  thỏa các phụ thuộc hàm của  $F$  thì cũng thỏa  $X \rightarrow Y$  và ký hiệu là  $F \models X \rightarrow Y$

### Bao đóng của tập phụ thuộc hàm

Bao đóng của tập phụ thuộc hàm  $F$ , ký hiệu là  $F^+$  là tập tất cả các phụ thuộc hàm được suy diễn từ  $F$ .

Trong trường hợp  $F = F^+$  thì  $F$  là họ đầy đủ của các phụ thuộc hàm.

## Hệ luật dẫn Amstrong

Amstrong đã đưa ra hệ luật tiên đề còn được gọi là hệ luật dẫn Amstrong sau:

Với  $X, Y, Z, W \subseteq U$ ,  $U$  là tập thuộc tính. Phụ thuộc hàm có các tính chất sau:

(1) Tính phản xạ

Nếu  $Y \subseteq X$  thì  $X \rightarrow Y$

(2) Tính tăng trưởng

Nếu  $X \rightarrow Y$  thì  $XZ \rightarrow YZ$

(3) Tính bắc cầu

Nếu  $X \rightarrow Y$  và  $Y \rightarrow Z$  thì  $X \rightarrow Z$

**Từ hệ tiên đề Amstrong suy ra một số tính chất sau:**

(1) Tính phân rã sau

Nếu  $X \rightarrow YZ$  và  $X \rightarrow Y$  thì  $X \rightarrow Z$

(2) Tính kết hợp

Nếu  $X \rightarrow Y$  và  $X \rightarrow Z$  thì  $X \rightarrow YZ$

(3) Tính tựa bắc cầu

Nếu  $X \rightarrow Y$  và  $YZ \rightarrow W$  thì  $XZ \rightarrow W$

## Ví dụ

Cho lược đồ quan hệ  $R(A, B, C, D, E, G, H)$  và tập phụ thuộc hàm

$F = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A\}$

Chứng tỏ  $AB \rightarrow E$  được suy diễn từ  $F$ .

(1)  $AB \rightarrow C$

(2)  $AB \rightarrow AB$  (tính phản xạ)

(3)  $AB \rightarrow B$  (tính phân rã)

(4)  $B \rightarrow D$

(5)  $AB \rightarrow D$  (tính bắc cầu 3+4)

(6)  $AB \rightarrow CD$  (tính hợp 1+5)

(7)  $CD \rightarrow E$

(8)  $AB \rightarrow E$  (tính bắc cầu 6+7)

### 1.3. Thuật toán tìm bao đóng của tập thuộc tính

#### Bao đóng của tập thuộc tính

Bao đóng của tập thuộc tính  $X$  đối với tập phụ thuộc hàm  $F$ , ký hiệu là  $X_F^+$  là tập tất cả các thuộc tính  $A$  có thể suy dẫn từ  $X$  nhờ tập bao đóng của các phụ thuộc hàm  $F^+$

$$X_F^+ = \{A \in Q^+ \mid X \rightarrow A \in F^+\}$$

#### Thuật toán tìm bao đóng của tập thuộc tính

Input:  $(Q, F), X \subseteq Q^+$

Output:  $X_F^+$

Bước 1: Tính dãy  $X^{(0)}, X^{(1)}, \dots, X^{(i)}$ :

- $X^{(0)} = X$
- $X^{(i+1)} = X^{(i)} \cup Z, \exists (Y \rightarrow Z) \in F (Y \subseteq X^{(i)})$ , loại  $(Y \rightarrow Z)$  ra khỏi  $F$
- Dừng khi  $X^{(i+1)} = X^{(i)}$  hoặc khi  $X^{(i)} = Q^+$

Bước 2: Kết luận  $X_F^+ = X^{(i)}$ :

#### Ví dụ 1

Cho lược đồ quan hệ  $R(A, B, C, D, E, G, H)$  và tập phụ thuộc hàm

$F = \{ \begin{array}{l} f1: B \rightarrow A, \\ f2: DA \rightarrow CE, \\ f3: D \rightarrow H, \\ f4: GH \rightarrow C, \\ f5: AC \rightarrow D \end{array} \}$

Tìm  $AC_F^+$

Bước 1:  $X_0 = AC$

Bước 2: Từ f1 đến f4 không thỏa, f5 thỏa nên  $X_1 = AC \cup D = ACD$

Lặp lại bước 2:

f1 không thỏa, f2 thỏa nên  $X_2 = ACD \cup CE = ACDE$

f3 thỏa nên  $X_3 = ACDE \cup H = ACDEH$

f4 không thỏa, f5 đã thỏa

Lặp lại bước 2: f2, f3 và f5 đã thỏa, f1 và f4 không thỏa. Nên  $X_4 = X_3 = ACDEH$

Vậy  $AC_F^+ = ACDEH$

#### 1.4. Bài toán thành viên

Cho tập thuộc tính Q, tập phụ thuộc hàm F trên Q và một phụ thuộc hàm  $X \rightarrow Y$  trên Q.

Câu hỏi đặt ra rằng  $X \rightarrow Y \in F^+$  hay không?

Giải quyết:  $X \rightarrow Y \in F^+ \Leftrightarrow Y \subseteq X^+$

#### Ví dụ

Cho lược đồ quan hệ R(A, B, C, D, E, G, H) và tập phụ thuộc hàm

$F = \{ \begin{array}{l} f1: B \rightarrow A, \\ f2: DA \rightarrow CE, \\ f3: D \rightarrow H, \\ f4: GH \rightarrow C, \\ f5: AC \rightarrow D \end{array} \}$

Cho biết  $AC \rightarrow E$  có thuộc  $F^+$  không?

Ta có  $AC_F^+ = ACDEH$

Vì  $E \in AC_F^+$  nên  $AC \rightarrow E \in F^+$

#### 1.5. Phủ tối thiểu của một tập phụ thuộc hàm

##### 1.5.1. Hai tập phụ thuộc hàm tương đương

Hai tập phụ thuộc hàm F và G tương đương nếu  $F^+ = G^+$ . Ký hiệu  $F \equiv G$

F được gọi là phủ G nếu  $F^+ \supseteq G^+$

### Thuật toán xác định F và G có tương đương không

Bước 1:

Với mỗi phụ thuộc hàm  $X \rightarrow Y \in F$

Kiểm tra  $X \rightarrow Y$  có là thành viên của G không.

Bước 2:

Với mỗi phụ thuộc hàm  $X \rightarrow Y \in G$

Kiểm tra  $X \rightarrow Y$  có là thành viên của F không.

Bước 3:

Nếu cả bước 1 và 2 đều đúng thì kết luận  $F \equiv G$

### Ví dụ

Cho lược đồ quan hệ R(A, B, C, D, E) và tập phụ thuộc hàm

$F = \{ A \rightarrow BC, A \rightarrow D, CD \rightarrow E \}$

$G = \{ A \rightarrow BCE, A \rightarrow ABD, CD \rightarrow E \}$

Cho biết F và G có tương đương không?

Bước 1:

$$A_G^+ = ABCDE \Rightarrow (A \rightarrow BC \in G^+ \wedge A \rightarrow D \in G^+)$$

Hơn nữa ta có  $(CD \rightarrow E \in F \wedge CD \rightarrow E \in G)$

Vậy mọi phụ thuộc hàm trong F đều là thành viên của G

Bước 2:

$$A_F^+ = ABCDE \Rightarrow (A \rightarrow BCE \in F^+ \wedge A \rightarrow ABD \in F^+)$$

Hơn nữa ta có  $(CD \rightarrow E \in G \wedge CD \rightarrow E \in F)$

Vậy mọi phụ thuộc hàm trong G đều là thành viên của F



Bước 3: Kết luận  $F \equiv G$

### 1.5.2. Phủ tối thiểu của tập phụ thuộc hàm

#### Phụ thuộc hàm có thuộc tính về trái dư thừa

Cho  $F$  là tập các phụ thuộc hàm trên lược đồ quan hệ  $Q$ . Khi đó  $Z \rightarrow Y \in F$  là phụ thuộc hàm có thuộc tính về trái dư thừa nếu tồn tại  $A \in Z$  mà  $F = F - (Z \rightarrow Y) \cup ((Z - A) \rightarrow Y)$

Ngược lại  $Z \rightarrow Y$  là phụ thuộc hàm có thuộc tính về trái không dư thừa hay  $Y$  phụ thuộc đầy đủ vào  $Z$ .  $Z \rightarrow Y$  còn được gọi là phụ thuộc hàm đầy đủ.

Ví dụ

Cho lược đồ quan hệ  $R(A, B, C, D, E)$  và tập phụ thuộc hàm

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

Khi đó  $A \rightarrow B, C \rightarrow D$  là những phụ thuộc hàm đầy đủ.  $BC \rightarrow D$  là phụ thuộc hàm có thuộc tính về trái  $B$  dư thừa.

Vấn đề là tìm các phụ thuộc hàm đầy đủ tương ứng bằng cách loại bỏ các thuộc tính dư thừa.

#### Thuật toán tìm các phụ thuộc hàm đầy đủ tương ứng

Thực hiện với mỗi phụ thuộc hàm  $X \rightarrow Y \in F$

Bước 1:  $X = A_1, A_2, \dots, A_n$  ( $n \geq 2$ , với  $n=1$  thì  $X \rightarrow Y$  là đầy đủ)

Đặt  $Z = X$

Bước 2: Với mỗi  $A_i$ , thực hiện

- $Tam = Z \setminus A_i$
- Nếu  $Tam \rightarrow Y \in F^+$  thì  $Z = Tam$

#### Tập phụ thuộc hàm có về phải một thuộc tính

Mỗi tập phụ thuộc hàm  $F$  đều tương đương với một tập phụ thuộc hàm  $G$  mà về phải của các phụ thuộc hàm thuộc  $G$  chỉ gồm một thuộc tính

Ví dụ:

Cho lược đồ quan hệ  $R(A, B, C, D, E)$  và tập phụ thuộc hàm

$$F = \{ A \rightarrow BC, A \rightarrow D, CD \rightarrow E \}$$

Khi đó  $G = \{ A \rightarrow B, A \rightarrow C, A \rightarrow D, CD \rightarrow E \}$  là tập phụ thuộc hàm có vẻ phải một thuộc tính và  $F \equiv G$

### Tập phụ thuộc hàm không dư thừa

$F$  là tập phụ thuộc hàm không dư thừa

nếu không tồn tại  $F' \subset F$  sao cho  $F' \equiv F$ .

Ngược lại  $F$  được gọi là tập phụ thuộc hàm dư thừa.

### Thuật toán loại những phụ thuộc hàm dư thừa

Với mỗi phụ thuộc hàm  $X \rightarrow Y \in F$

nếu  $X \rightarrow Y$  là thành viên của  $F - \{X \rightarrow Y\}$  thì loại  $X \rightarrow Y$  khỏi  $F$ .

#### 1.5.3. Phủ tối thiểu của tập phụ thuộc hàm

$F$  được gọi là phủ tối thiểu của tập phụ thuộc hàm (hay tập phụ thuộc hàm tối thiểu) nếu thỏa:

- (i)  $F$  là tập phụ thuộc hàm có thuộc tính về trái không dư thừa
- (ii)  $F$  là tập phụ thuộc hàm có vẻ phải một thuộc tính
- (iii)  $F$  là tập phụ thuộc hàm không dư thừa

### Thuật toán tìm phủ tối thiểu của tập phụ thuộc hàm

Bước 1: Loại các thuộc tính có vẻ trái dư thừa của mọi phụ thuộc hàm

Bước 2: Phân rã các phụ thuộc hàm có vẻ phải nhiều thuộc tính thành các phụ thuộc hàm có vẻ phải một thuộc tính

Bước 3: Loại các phụ thuộc hàm dư thừa khỏi  $F$

### Ví dụ

Cho lược đồ quan hệ  $R(A, B, C, D, E, G, H)$  và tập phụ thuộc hàm

$F = \{ B \rightarrow A, DA \rightarrow CE, D \rightarrow H, GH \rightarrow C, AC \rightarrow D \}$

Tìm phủ tối thiểu của  $F$ .

Bước 1:

- Với  $DA \rightarrow CE$ :
  - Giả sử D thừa thì  $A \rightarrow CE \in F^+$ : ta có  $A_F^+ = A$ ,  $CE \notin A_F^+$  nên D không thừa
  - Giả sử A thừa thì  $B \rightarrow CE \in F^+$ : ta có  $D_F^+ = DH$ ,  $CE \notin D_F^+$  nên A không thừa
- Với  $GH \rightarrow C$ :
  - Giả sử G thừa thì  $H \rightarrow C \in F^+$ : ta có  $H_F^+ = H$ ,  $C \notin H_F^+$  nên G không thừa
  - Giả sử H thừa thì  $G \rightarrow C \in F^+$ : ta có  $G_F^+ = G$ ,  $C \notin G_F^+$  nên H không thừa
- Với  $AC \rightarrow D$ :
  - Giả sử A thừa thì  $C \rightarrow D \in F^+$ : ta có  $C_F^+ = C$ ,  $D \notin C_F^+$  nên A không thừa
  - Giả sử C thừa thì  $A \rightarrow D \in F^+$ : ta có  $A_F^+ = A$ ,  $D \notin A_F^+$  nên C không thừa

Vậy mọi phụ thuộc hàm đều là đầy đủ.

Bước 2: Phân rã về phải ta có  $F = \{ B \rightarrow A, DA \rightarrow C, DA \rightarrow E, D \rightarrow H, GH \rightarrow C, AC \rightarrow D \}$

Bước 3: Loại các phụ thuộc hàm dư thừa:

- Với  $B \rightarrow A$ : ta có  $B_{F \setminus \{B \rightarrow A\}}^+ = B$ ,  $A \notin B_{F \setminus \{B \rightarrow A\}}^+$  nên  $B \rightarrow A$  là không thừa
- Với  $DA \rightarrow C$ : ta có  $DA_{F \setminus \{DA \rightarrow C\}}^+ = DAEH$ ,  $C \notin DA_{F \setminus \{DA \rightarrow C\}}^+$  nên  $DA \rightarrow C$  là không thừa
- Với  $DA \rightarrow E$ : ta có  $DA_{F \setminus \{DA \rightarrow E\}}^+ = DACH$ ,  $E \notin DA_{F \setminus \{DA \rightarrow E\}}^+$  nên  $DA \rightarrow E$  là không thừa
- Với  $D \rightarrow H$ : ta có  $D_{F \setminus \{D \rightarrow H\}}^+ = D$ ,  $H \notin D_{F \setminus \{D \rightarrow H\}}^+$  nên  $D \rightarrow H$  là không thừa
- Với  $GH \rightarrow C$ : ta có  $GH_{F \setminus \{GH \rightarrow C\}}^+ = GH$ ,  $C \notin GH_{F \setminus \{GH \rightarrow C\}}^+$  nên  $GH \rightarrow C$  là không thừa
- Với  $AC \rightarrow D$ : ta có  $AC_{F \setminus \{AC \rightarrow D\}}^+ = AC$ ,  $D \notin AC_{F \setminus \{AC \rightarrow D\}}^+$  nên  $AC \rightarrow D$  là không thừa

Vậy phủ tối thiểu của F:  $PTT(F) = \{ B \rightarrow A, DA \rightarrow C, DA \rightarrow E, D \rightarrow H, GH \rightarrow C, AC \rightarrow D \}$

## 2. Khóa

### 2.1. Định nghĩa

Cho lược đồ quan hệ  $Q(A_1, A_2, \dots, A_n)$ ,  $Q^+$  là tập thuộc tính của quan hệ Q, F là tập phụ thuộc hàm trên Q, K là tập con của  $Q^+$ . Khi đó K gọi là một khóa của Q nếu:

$$(i) \quad K_F^+ = Q^+$$

$$(ii) \quad \text{Không tồn tại } K' \subset K \text{ sao cho } K'^+_F = Q^+$$

Thuộc tính A được gọi là thuộc tính khóa nếu  $A \in K$ , trong đó K là khóa của Q. Ngược lại thuộc tính A được gọi là thuộc tính không khóa.

$K'$  được gọi là siêu khóa nếu  $K \subseteq K'$ .

## 2.2. Thuật toán tìm khóa

### 2.2.1. Thuật toán tìm một khóa

Sử dụng đồ thị có hướng để tìm khóa như sau:

Bước 1:

- Mỗi nút của đồ thị là tên một thuộc tính của lược đồ quan hệ R
- Cung nối hai thuộc tính A và B thể hiện phụ thuộc hàm  $A \rightarrow B$
- Thuộc tính chỉ có các mũi tên đi ra (nghĩa là chỉ nằm trong vế trái của phụ thuộc hàm) được gọi là nút gốc
- Thuộc tính chỉ có các mũi tên đi tới (nghĩa là chỉ nằm trong vế phải của phụ thuộc hàm) được gọi là nút lá

Bước 2:

- Xuất phát từ tập các nút gốc (X), dựa trên tập các phụ thuộc hàm F, tìm bao đóng  $X_F^+$ .
- Nếu  $X_F^+ = Q^+$  thì X là khóa, ngược lại bổ sung một thuộc tính không thuộc nút lá vào X rồi thực hiện tìm bao đóng của X. Dừng khi tìm được một khóa của R.

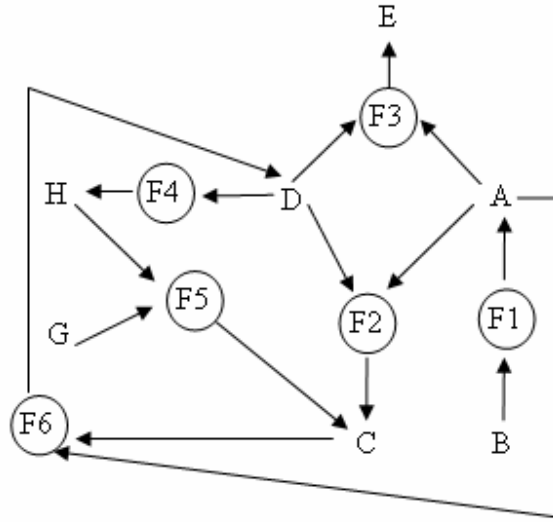
Ví dụ

Cho lược đồ quan hệ R(A, B, C, D, E, G, H) và tập phụ thuộc hàm

$$F = \{ B \rightarrow A, DA \rightarrow CE, D \rightarrow H, GH \rightarrow C, AC \rightarrow D \}$$

Tìm một khóa của R.

Phân rã vế phải ta có  $F = \{ B \rightarrow A, DA \rightarrow C, DA \rightarrow E, D \rightarrow H, GH \rightarrow C, AC \rightarrow D \}$



Nhận thấy từ đồ thị trên, nút B và G là nút gốc. Khóa của R phải chứa thuộc tính B hoặc G, trong ví dụ này chọn B.

$B_F^+ = BA$ , Vì  $B_F^+ \neq Q^+$  nên B không là khóa. Nhận thấy D là thuộc tính ở vế trái của ba phụ thuộc hàm trong F nên bổ sung thuộc tính D vào để xét khóa.

$BD_F^+ = BDACEH$ , vì  $BD_F^+ \neq Q^+$  nên BD không là khóa. Bổ sung thuộc tính G.

$BDG_F^+ = BDGACEH$ , vì  $BDG_F^+ = Q^+$  nên BDG là khóa.

### 2.2.2. Thuật toán tìm tất cả các khóa

Tập thuộc tính nguồn, ký hiệu là N, là tập chứa những thuộc tính chỉ xuất hiện ở vế trái, không nằm bên vế trái và vế phải của mọi phụ thuộc hàm

Tập thuộc tính trung gian, ký hiệu là TG, là tập chứa những thuộc tính vừa xuất hiện ở vế trái, vừa xuất hiện ở vế phải trong các phụ thuộc hàm

#### Thuật toán

Bước 1: Tính tập nguồn N. Nếu  $N_F^+ = Q^+$  thì chỉ có 1 khóa là N, ngược lại qua bước 2

Bước 2: Tính tập trung gian TG. Tính tập tất cả các tập con  $X_i$  của tập TG.

Bước 3: Tìm tập S chứa mọi siêu khóa  $S_i$ :

Với mỗi  $X_i$ , nếu  $(N \cup X_i)_F^+ = Q^+$  thì  $S_i = (N \cup X_i)$

Bước 4: Loại các siêu khóa không tối thiểu:

$\forall S_i, S_j \in S$ , nếu  $S_i \subset S_j$  thì  $S = S - S_j$

**Ví dụ.** Cho lược đồ quan hệ  $R(A, B, C)$  và tập phụ thuộc hàm

$$F = \{ AB \rightarrow C, C \rightarrow A \}$$

Tìm mọi khóa của  $R$ .

Bước 1:  $N = \{B\}$ ,  $N_F^+ = B \neq Q^+$

Bước 2:  $TG = \{AC\}$ , tập các tập con trung gian là  $CTG = \{\emptyset, A, C, AC\}$

Bước 3:

$N$	$X_i$	$N \cup X_i$	$(N \cup X_i)^+$	$(N \cup X_i)^+ = Q^+ ?$
B	$\emptyset$	B	B	Sai
B	A	BA	BAC	Đúng
B	C	BC	BCA	Đúng
B	AC	BAC	BAC	Đúng

Như vậy tập siêu khoá  $S = \{BA, BC, BAC\}$

Bước 4: Loại các siêu khóa không tối thiểu:

Nhận thấy rằng  $BA \subset BAC$  nên loại siêu khóa BAC. Tập các khóa còn lại là  $S = \{BA, BC\}$

### Cải tiến thuật toán

Nhận thấy rằng ở bước 3, khi xét tập một con  $X_i$ , nếu  $(N \cup X_i)^+ = Q^+$ , khi đó  $N \cup X_i$  là một khóa. Trong trường hợp này nếu xét tiếp các tập  $X_j : X_i \subset X_j$ , thì hiển nhiên  $(N \cup X_j)^+ = Q^+$ , và cuối cùng siêu khoá  $N \cup X_j$  cũng sẽ bị loại ở bước 4.

Hơn nữa, bước tính tập con đầu tiên là rỗng cũng không cần tính.

Do đó, khi tính bao đóng của tập thuộc tính, không tính  $X_i = \emptyset$  và nếu  $(N \cup X_i)^+ = Q^+$ , thực hiện loại bỏ các tính toán cho các trường hợp  $X_j : X_i \subset X_j$ . Khi đó kết thúc thuật toán tại bước 3, không cần thực hiện bước 4 vì tất cả các khóa nhận được ở bước 3 đều là tối thiểu.

## 3. Ràng buộc toàn vẹn

### 3.1. Ràng buộc toàn vẹn – các yếu tố của ràng buộc toàn vẹn

Trong CSDL luôn tồn tại nhiều mối liên hệ giữa các thuộc tính, các bộ, các bảng với nhau. Các mối liên hệ này là những điều kiện bất biến mà tất cả các bộ của những quan hệ có liên quan đều phải thỏa mãn tại mọi thời điểm. Những điều kiện bất biến đó được gọi là ràng buộc toàn vẹn.

Ràng buộc toàn vẹn (Integrity constraint) viết tắt tiếng việt là RBTV, là một điều kiện được định nghĩa liên quan đến một hay nhiều quan hệ khác nhau. Các mối liên hệ ràng buộc là những điều kiện bất biến mà mọi thể hiện của quan hệ đều phải thỏa mãn trong mọi thời điểm.

Trong thực tế, RBTV là các quy tắc quản lý trong thế giới thực.

Mục đích của RBTV là bảo đảm tính nhất quán của dữ liệu, bảo đảm rằng dữ liệu luôn biểu diễn đúng ngữ nghĩa trong thực tế.

Ví dụ: với giao dịch chuyển tiền, thực hiện qua các bước sau:

- Tại tài khoản của người gởi: trừ tiền
- Tại tài khoản của người nhận: thêm tiền
- Nếu cả hai việc trên đều thành công thì hoàn tất giao dịch, ngược lại quay lui giao dịch.

Qua ví dụ trên, giao dịch thực hiện trừ tiền và tăng tiền trong tài khoản của người gởi và người nhận, nếu không thì không thực hiện gì cả. Việc thực hiện giao dịch cần phải bảo đảm tính đúng của dữ liệu.

Mỗi một RBTV có các yếu tố sau:

### **Bối cảnh**

Bối cảnh là một hay nhiều quan hệ cần phải sử dụng để kiểm tra RBTV. Hay nói cách khác bối cảnh của RBTV là những quan hệ có khả năng bị vi phạm RBTV khi thực hiện các thao tác cập nhật dữ liệu (thêm, xóa, sửa các bộ)

### **Biểu diễn: điều kiện hay nội dung**

Điều kiện được kiểm tra trên mọi thay đổi của thể hiện của các quan hệ cơ sở. Điều kiện của một RBTV có thể được biểu diễn bằng nhiều cách khác nhau, chẳng hạn như ngôn ngữ tự nhiên, ngôn ngữ hình thức (thuật toán, đại số quan hệ,...).

Khi một RBTV bị vi phạm, có thể xử lý bằng cách thông báo cho NSD biết RBTV đã bị vi phạm như thế nào, hoặc từ chối thực hiện thao tác cập nhật dữ liệu và thông báo cho NSD biết thao tác cập nhật bị từ chối trên các quan hệ nào và tại các bước nào.

## Tầm ảnh hưởng

Trong quá trình phân tích thiết kế CSDL, cần thiết phải lập bảng tầm ảnh hưởng cho RBTV nhằm xác định thời điểm cần kiểm tra RBTV, và khi kiểm tra cần kiểm tra trên quan hệ nào.

Bảng tầm ảnh hưởng của một RBTV có dạng sau:

Tên RBTV	Thêm	Xóa	Sửa
R <sub>1</sub>	+	-	+(A)
R <sub>2</sub>	+	-	+(B)
...	...	...	...
R <sub>n</sub>	-	-	+

Trong đó:

Dấu + thể hiện thao tác có thể gây ra vi phạm RBTV. Trong trường hợp +(A) cho biết thao tác sửa có thể gây vi phạm trên thuộc tính A. Dấu - thể hiện thao tác không vi phạm RBTV.

### 3.2. Các loại ràng buộc toàn vẹn

Trong quá trình phân tích thiết kế CSDL, yêu cầu cần thiết là phải tìm được những RBTV tiềm ẩn trong CSDL. Việc phân loại RBTV cho phép người phân tích tìm kiếm đầy đủ, tránh bỏ sót những RBTV. Các loại RBTV được phân thành hai dạng chính như sau:

- RBTV có bối cảnh là một quan hệ
- RBTV có bối cảnh là nhiều quan hệ

#### 3.2.1. Ràng buộc toàn vẹn có bối cảnh là một quan hệ

RBTV có bối cảnh là một quan hệ được chia thành ba loại: RBTV miền giá trị, RBTV liên bộ và RBTV liên thuộc tính

##### 3.2.1.1. Ràng buộc toàn vẹn miền giá trị

Quy định rõ về miền giá trị của một thuộc tính.

Ví dụ. Thời gian phân công tham gia đề án của một nhân viên không quá 40h/tuần

- Bối cảnh: quan hệ PHANCONG



- Biểu diễn:  $\forall pc \in PHANCONG(pc.ThoiGian \leq 40)$
- Bảng tầm ảnh hưởng:

RB1	Thêm	Xóa	Sửa
PHANCONG	+	-	+(ThoiGian)

Ví dụ. Điểm của môn học phải là thang điểm 10

- Bối cảnh: quan hệ KETQUA (MaMH, MaLop, MaKH, Diem )
- Biểu diễn:  $\forall kq \in KETQUA(kq.Diem \geq 0 \wedge kq.Diem \leq 10)$
- Bảng tầm ảnh hưởng:

RB2	Thêm	Xóa	Sửa
KETQUA	+	-	+(Diem)

### 3.2.1.2. Ràng buộc toàn vẹn liên thuộc tính

Quy định các ràng buộc giữa các thuộc tính khác nhau trong cùng một quan hệ

Ví dụ. Ngày trả sách phải là bằng hoặc sau ngày mượn sách

- Bối cảnh: quan hệ MUONSACH(MaSach, MaDocGia, NgayMuon, NgayHenTra, NgayThucTra)
- Biểu diễn:  $\forall ms \in MUONSACH(ms.NgayMuon \leq ms.NgayHenTra \wedge ms.NgayMuon \leq ms.NgayThucTra)$
- Bảng tầm ảnh hưởng:

RB3	Thêm	Xóa	Sửa
MUONSACH	+	-	+(NgayMuon, NgayHenTra, NgayThucTra)

### 3.2.1.3. Ràng buộc toàn vẹn liên bộ

Quy định sự tồn tại của một hoặc nhiều bộ phụ thuộc vào sự tồn tại của một hoặc nhiều bộ khác trong cùng quan hệ.

**RBTV khóa chính là RBTV liên bộ**

Ví dụ. Mỗi đề án trong công ty có một mã duy nhất để phân biệt với các đề án khác

- Bối cảnh: quan hệ DEAN
- Điều kiện:  $\forall da1, da2 \in DEAN : da1 \neq da2 \Rightarrow (da1.MaDA \neq da2.MaDA)$
- Bảng tầm ảnh hưởng:

RB4	Thêm	Xóa	Sửa
DEAN	+	-	+(MaDA)

### RBTV về số lượng các bộ trong một quan hệ.

Ví dụ. Mỗi sinh viên trong một học kỳ được đăng ký không quá 8 môn học

- Bối cảnh: quan hệ DANGKY(MaSV, MaMH)
- Biểu diễn:  $\forall dk1 \in DANGKY : count(dk2 \in DANGKY \mid dk2.MaSV = dk1.MaSV) \leq 8$ , trong đó count() là hàm đếm số bộ của một quan hệ thỏa điều kiện trong ngoặc ().
- Bảng tầm ảnh hưởng:

RB4	Thêm	Xóa	Sửa
DANGKY	+	-	+(MaSV)

### 3.2.2. Ràng buộc toàn vẹn có bối cảnh là nhiều quan hệ

RBTV có bối cảnh là nhiều quan hệ được chia thành năm loại:

- RBTV tham chiếu
- RBTV liên bộ - liên quan hệ
- RBTV liên thuộc tính - liên quan hệ
- RBTV do thuộc tính tổng hợp
- RBTV do chu trình

#### 3.2.2.1. RBTV tham chiếu

Quy định giá trị xuất hiện của một tập thuộc tính trong một quan hệ phải xuất hiện trong một tập thuộc tính trong một quan hệ khác. RBTV này còn được gọi là RBTV tham chiếu, RBTV phụ thuộc tồn tại hay RBTV khóa ngoại.

### Ví dụ RBTV trên 2 quan hệ

Một nhân viên phải thuộc về một phòng trong công ty, nghĩa là trong quan hệ NHANVIEN, nếu một mã phòng (Phong) mà nhân viên trực thuộc xuất hiện, thì mã phòng này phải xuất hiện trong quan hệ PHONGBAN, cụ thể là thuộc tính (MaPhong). Như vậy:

- Bối cảnh: NHANVIEN, PHONGBAN
- Biểu diễn:  
$$\forall nv \in NHANVIEN ((nv.Phong = NULL) \vee (\exists pb \in PHONGBAN (nv.Phong = pb.MaPhong)))$$
- Bảng tầm ảnh hưởng:

RB5	Thêm	Xóa	Sửa
NHANVIEN	+	-	+(Phong)
PHONGBAN	-	+	+(MaPhong)

### Ví dụ RBTV trên 1 quan hệ

Người quản lý (MaNQL) của một nhân viên cũng phải là một nhân viên trong công ty

- Bối cảnh: NHANVIEN
- Điều kiện:  
$$\forall nv \in NHANVIEN ((nv.MaNQL = NULL) \vee (\exists nv1 \in NHANVIEN (nv.MaNQL = nv1.MaNV)))$$
- Bảng tầm ảnh hưởng:

RB6	Thêm	Xóa	Sửa
NHANVIEN	+	-	+(MaNV, MaNQL)

### Ảnh hưởng của RBTV đối với các thao tác thêm, xóa, sửa dữ liệu

Giả sử  $r_2$  có một khóa ngoại  $\alpha$  tham chiếu đến K trong  $r_1$ , khi đó:  $\pi_\alpha(r_2) \subseteq \pi_K(r_1)$

#### Thêm

Khi thêm một bộ  $t_2$  vào  $r_2$  thì phải bảo đảm tồn tại  $t_1$  trong  $r_1$  sao cho  $t_1[K] = t_2[\alpha]$

## Xóa

Giả sử xóa  $t_1$  khỏi  $r_1$ . Khi đó cần xử lý các bộ trong  $r_2$  tham chiếu tới  $t_1$ , nghĩa là  $s = \sigma_{\alpha=t_1[K]}(r_2)$ . Nếu  $s \neq \emptyset$  thì

- Không thực hiện hành động xóa dữ liệu, hoặc
- Xóa dây chuyền, nghĩa là xóa tất cả các bộ trong  $s$

## Sửa

Trường hợp cập nhật  $t_2$  trong  $r_2$

- Cập nhật  $t_2$  trong  $r_2$ , sửa khóa ngoại  $\alpha$
- Tương tự như trường hợp thêm dữ liệu
- Kiểm tra  $t_2'[\alpha] \in \pi_K(r_1)$

Trường hợp cập nhật  $t_1$  trong  $r_1$

- Cập nhật  $t_1$  trong  $r_1$
- Tương tự như trường hợp xóa dữ liệu
- Kiểm tra  $\sigma_{\alpha=t_1[K]}(r_2) = \emptyset$

### 3.2.2.2. RBTV liên bộ - liên quan hệ

Quy định về từng nhóm các bộ của nhiều quan hệ bối cảnh khác nhau.

Ví dụ Một hóa đơn bán hàng phải có ít nhất một mặt hàng, nghĩa là một chi tiết hóa đơn bán hàng phải có ít nhất một mặt hàng.

- Bối cảnh: HOADON, CTIETHD
- Biểu diễn:  $\forall hd \in HOADON (\exists cthd \in CTIETHD (hd.MaHD = cthd.MaHD))$
- Bảng tầm ảnh hưởng:

RB7	Thêm	Xóa	Sửa
HOADON	+	-	+(MaHD)
CTIETHD	-	+	+(MaHD)

### 3.2.2.3. RBTV liên thuộc tính - liên quan hệ

Quy định về mối liên hệ giữa các thuộc tính trên nhiều quan hệ bối cảnh khác nhau.

Ví dụ Giả sử cho phép thanh toán tiền nhiều lần và thanh toán sau khi mua hàng, khi đó ngày thanh toán tiền theo một hóa đơn mua hàng phải bằng hoặc sau ngày mua hàng.

- Bối cảnh:  $HOADON(\underline{MaHD}, MaKH, NgayHD, TriGia),$   
 $THANHTOAN(\underline{MaHD}, \underline{NgayTToan}, \underline{LanTToan}, SoTienTToan)$

- Biểu diễn:

$$\forall hd \in HOADON (\forall tt \in THANHTOAN (hd.MaHD = tt.MaHD \Rightarrow hd.NgayHD \leq tt.NgayTToan))$$

- Bảng tầm ảnh hưởng:

RB8	Thêm	Xóa	Sửa
HOADON	+	-	+(MaHD, NgayHD)
THANHTOAN	+	-	+(MaHD, NgayTToan)

### 3.2.2.4. RBTV do thuộc tính tổng hợp

Quy định về mối liên hệ giữa các thuộc tính do sự có mặt của thuộc tính tính toán.

Ví dụ Điểm trung bình của sinh viên bằng trung bình của các môn mà sinh viên theo học

- Bối cảnh:  $SINHVIEN(\underline{MaSV}, HoSV, TenSV, Khoa, DTB)$   
 $KETQUA (\underline{MaSV}, \underline{MaMon}, Diem)$

- Biểu diễn:

$$\forall sv \in SINHVIEN (\exists kq \in KETQUA (sv.MaSV = kq.MaSV \Rightarrow sv.DTB = AVG(kq.Diem)))$$

- Bảng tầm ảnh hưởng:

RB9	Thêm	Xóa	Sửa
SINHVIEN	+	-	+(MaSV, DTB)
KETQUA	+	+	+(MaSV, Diem)

### 3.2.2.5. RBTV do chu trình

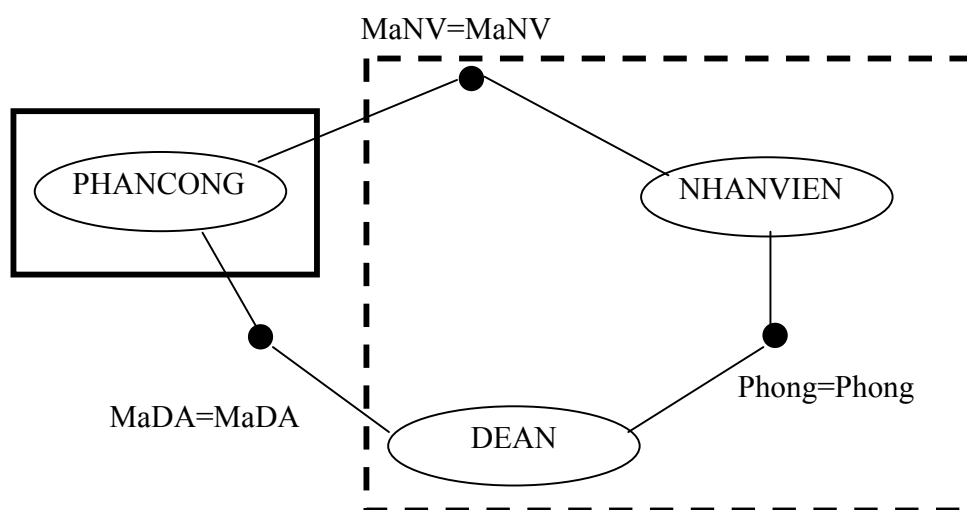
Xảy ra khi có sự hiện diện của chu trình. Để nhận diện chu trình, người ta biểu diễn lược đồ CSDL như sau:

Nút thể hiện lược đồ	
Nút thuộc tính kết	MaNV=MaNV 
Cung nối giữa nút lược đồ và nút thuộc tính kết	

Ví dụ Một nhân viên chỉ được phân công vào các đề án do phòng mình chủ trì

- Bối cảnh: NHANVIEN, DEAN, PHANCONG

Đồ thị thể hiện chu trình như sau:



- Biểu diễn:

$\forall pc \in PHANCONG (\exists nvda \in NV\_DA (nvda.MaNV = pc.MaNV \wedge nvda.MaDA = pc.MaDA))$   
 với:  $NV\_DA \leftarrow NHANVIEN \triangleright \triangleleft_{Phong=MaPhong} DEAN$

- Bảng tầm ảnh hưởng:

RB10	Thêm	Xóa	Sửa
------	------	-----	-----

NHANVIEN	-	+	+(MaNV, Phong)
DEAN	-	+	+(MaDA, Phong)
PHANCONG	+	-	+(MaDA, MaNV)

#### 4. Bài tập

##### Bài tập 1

- Hãy chứng minh 3 tính chất phân rã, kết hợp và tựa bắc cầu.
- Hãy tìm hiểu các tính chất của bao đóng tập thuộc tính, phủ tối thiểu

##### Bài tập 2

Cho lược đồ quan hệ  $R(A, B, C, D, E, G)$  và tập phụ thuộc hàm

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

Tìm  $AB_F^+, CGD_F^+, A_F^+$

##### Bài tập 3

Cho lược đồ quan hệ  $R(A, B, C, D, E, G)$  và tập phụ thuộc hàm

$$F = \{B \rightarrow C, A \rightarrow EG, B \rightarrow A, G \rightarrow E\}$$

Tìm  $AB_F^+, CGD_F^+, A_F^+$

##### Bài tập 4

Cho lược đồ quan hệ  $R(A, B, C, D, E)$  và tập phụ thuộc hàm

$$F = \{A \rightarrow C, BC \rightarrow D, D \rightarrow E, E \rightarrow A\}$$

Tìm  $AB_F^+, BD_F^+, D_F^+$

##### Bài tập 5

Cho lược đồ quan hệ  $R(A, B, C, D, E)$  và tập phụ thuộc hàm

$$F = \{B \rightarrow C, AC \rightarrow D, D \rightarrow G, AG \rightarrow E\}$$

Cho biết  $AC \rightarrow E$  có thuộc  $F^+$  không?

Cho biết  $BD \rightarrow AD$  có thuộc  $F^+$  không?

Với các bài tập 7, 8, 9:

- Tìm một khóa (theo thuật toán tìm một khóa)
- Tìm mọi khóa (theo thuật toán tìm mọi khóa)
- Tìm phủ tối thiểu

#### Bài tập 7

Cho lược đồ quan hệ  $R(A, B, C, D)$  và tập phụ thuộc hàm

$$F = \{AB \rightarrow CD, B \rightarrow C, C \rightarrow D\}$$

#### Bài tập 8

Cho lược đồ quan hệ  $R \{ABCDEFGHIJKLM\}$  và tập phụ thuộc hàm

$$F = \{A \rightarrow B, C \rightarrow D, E \rightarrow F, G \rightarrow AHK, AH \rightarrow G, GLC \rightarrow M\}$$

#### Bài tập 9

Cho lược đồ quan hệ  $R \{ABCDEFGHKL\}$  và tập phụ thuộc hàm

$$F = \{A \rightarrow B, AC \rightarrow D, F \rightarrow G, FK \rightarrow LEH, E \rightarrow FH\}$$

#### Bài tập 10

Với hai bài toán tình huống là quản lý đề án và quản lý ngân hàng, ngoại trừ các ràng buộc khoá chính và khoá ngoại, hãy tìm tất cả các RBTV theo yêu cầu: bối cảnh, biểu diễn, tầm ảnh hưởng. Với những RBTV tìm được, hãy phân theo từng loại RBTV.



## Chương 8

# Dạng chuẩn và chuẩn hóa cơ sở dữ liệu

---

Chương này giới thiệu các dạng chuẩn, phân rã bảo toàn thông tin, bảo toàn phụ thuộc hàm, qua đó cũng trình bày cách phân rã bảo toàn bảo toàn thông tin và bảo toàn phụ thuộc.

### 1. Dạng chuẩn của lược đồ quan hệ

Để dễ dàng trình bày các dạng chuẩn, cần nắm rõ các khái niệm:

#### Thuộc tính khoá:

Cho lược đồ quan hệ  $Q(A_1, A_2, \dots, A_n)$ , thuộc tính  $B$  được gọi là **thuộc tính khoá** nếu  $B$  là một thuộc tính thành phần trong một khoá nào đó của  $Q$ , ngược lại  $B$  được gọi là **thuộc tính không khoá**

Ví dụ:  $R(A, B, C, D)$ ,  $F = \{AB \rightarrow C, B \rightarrow D, BC \rightarrow A\}$

Trong ví dụ trên, lược đồ  $R$  có 2 khoá là  $AB, BC$ . Khi đó  $A, B, C$  là thuộc tính khoá,  $D$  là thuộc tính không khoá.

#### Giá trị nguyên tố

Giá trị nguyên tố là giá trị không phân nhỏ được nữa.

Ví dụ: Giá trị ChiTietMua: “Bánh Orion 1 gói, Kẹo mút 2 cây” không phải là giá trị nguyên tố vì có thể phân thành: tên hàng, số lượng, đơn vị tính.

### 1.1. Dạng chuẩn 1

Lược đồ  $Q$  ở dạng chuẩn 1 nếu mọi thuộc tính đều mang giá trị nguyên tố.

Trong bài toán xét dạng chuẩn, dạng chuẩn thấp nhất là dạng chuẩn 1.

Ví dụ. Cho lược đồ  $HOADON(\underline{MaHD}, MaKH, NgayHD, CtietMua, SoTien)$ , và có thể hiện như sau

MaHD	MaKH	NgàyHD	CtietMua			SoTien
			Tên hàng	Số lượng	ĐVT	
HD01	KH01	15-10-05	Bánh Orion	1	Gói	25.000
			Kẹo mút	2	Cây	2.000
HD02	KH01	18-10-05	Gạo	2	Kg	30.000
HD03	KH02	24-10-05	Đường	1	Kg	15.000
			Bánh AFC	2	Gói	24.000

Nhận thấy rằng CTiếtMua không mang giá trị nguyên tố, do đó HOADON không đạt dạng chuẩn 1.

## 1.2. Dạng chuẩn 2

### 1.2.1. Định nghĩa

Lược đồ Q ở dạng chuẩn 2 nếu thỏa:

- (1) Q đạt dạng chuẩn 1
- (2) Mọi thuộc tính không khóa của Q đều phụ thuộc đầy đủ vào khóa

### 1.2.2. Kiểm tra dạng chuẩn 2

Để kiểm tra dạng chuẩn 2 thực hiện:

Bước 1: Tìm mọi khóa của Q

Bước 2: Với mỗi khóa K, tìm bao đóng của tập tất cả các tập con thực sự  $S_i$  của K

Bước 3: Nếu tồn tại bao đóng  $S_i^+$  chứa thuộc tính không khóa thì Q không đạt dạng chuẩn 2, ngược lại Q đạt dạng chuẩn 2.

Ví dụ 1. Cho Q1 (A, B, C, D),  $F=\{A \rightarrow B, B \rightarrow DC\}$

Lược đồ chỉ có một khóa là A, nên mọi thuộc tính đều phụ thuộc đầy đủ vào khóa. Do vậy Q1 đạt dạng chuẩn 2.

Ví dụ 2. Cho Q2 (A, B, C, D),  $F=\{AB \rightarrow D, C \rightarrow D\}$

Lược đồ có khóa là ABC, ngoài ra còn có  $C \subset ABC$  mà  $C \rightarrow D$ , trong đó D là thuộc tính không khóa (nghĩa là thuộc tính D không phụ thuộc đầy đủ vào khóa). Do vậy Q2 không đạt dạng chuẩn 2.

Ví dụ 3:

Xem ví dụ đơn giản bằng CSDL gồm 2 quan hệ MONHOC, SINHVIEN như sau:

MONHOC (MaMH, TenMH, STC, Loai)

*Tân từ:* Mỗi môn học có mã môn học (MaMH) duy nhất để phân biệt với các môn học khác, có tên môn học (TenMH), số tín chỉ (STC), và loại bắt buộc hay tự chọn (Loai)

MaMH	TenMH	STC	Loai
CT101	Nhập môn tin học	4	BB
CT102	TH kỹ năng máy tính	5	BB
TN311	Xác suất thống kê	3	BB
CT103	Thiết kế Web	4	TC
CT110	Nguyên lý lập trình 2	5	BB

SINHVIEN (MSSV, MaMH, TenSV, DiaChi, Diem)

*Tân từ:* Mỗi sinh viên (MSSV) khi thi một môn học (MaMH) được ghi nhận lại điểm (Diem), ngoài ra còn có thông tin liên quan đến sinh viên như họ tên (TenSV), địa chỉ (DiaChi)

MSSV	MaMH	TenSV	DiaChi	Diem
0310677	CT101	Nguyễn Thị Hoa	11 Nguyễn Công Trứ, Đà Lạt	6
0310678	CT101	Trần Hoàng	20 Bùi Thị Xuân, Đà Lạt	4
0310679	CT101	Lê Thanh Sơn	2 Nhà Chung, Đà Lạt	7
0310677	CT102	Nguyễn Thị Hoa	11 Nguyễn Công Trứ, Đà Lạt	8
0310678	CT102	Trần Hoàng	20 Bùi Thị Xuân, Đà Lạt	8
0310678	TN311	Trần Hoàng	20 Bùi Thị Xuân, Đà Lạt	3
0310679	TN311	Lê Thanh Sơn	2 Nhà Chung, Đà Lạt	6
0310677	CT103	Nguyễn Thị Hoa	11 Nguyễn Công Trứ, Đà Lạt	9
0310678	CT103	Trần Hoàng	20 Bùi Thị Xuân, Đà Lạt	7

0310000	CT110	Nguyễn Ngọc	1 Lê Hồng Phong, Đà Lạt	9
---------	-------	-------------	-------------------------	---

Ở quan hệ SINHVIEN có thể nhận thấy các phụ thuộc:

MSSV, MaMH → Diem

MSSV → TenSV, DiaChi

Vậy quan hệ SINHVIEN không đạt dạng chuẩn 2 vì thuộc tính TenSV, DiaChi là thuộc tính không khoá chỉ phụ thuộc vào MSSV: như vậy không phụ thuộc đầy đủ vào khoá.

Trong quá trình cập nhật và lưu trữ dữ liệu xuất hiện những vấn đề sau:

- Ở quan hệ SINHVIEN, việc lưu trữ thông tin một sinh viên bị lặp lại (tên, địa chỉ)
- Quá trình cập nhật:
  - Sửa đổi: Khi cần sửa đổi địa chỉ của một sinh viên (ví dụ như Nguyễn Thị Hoa) cần phải sửa đổi 3 lần vì trùng lặp thông tin, hơn nữa, khi sửa đổi thông tin của về một sinh viên lại không liên quan đến thông tin về thi cử.
  - Thêm: Nếu chèn thêm một bộ vào quan hệ SINHVIEN mà sinh viên chưa thi môn nào thì không được vì khoá MSSV, MaMH là không đầy đủ. Vấn đề này chỉ được khắc phục khi loại bỏ những thông tin về kết quả thi cử ra khỏi quan hệ.
  - Xóa: Giả sử rằng cần xóa bỏ môn CT110 mà danh sách sinh viên vẫn giữ nguyên. Khi đó xóa bộ {'CT110', 'Nguyên lý lập trình 2', 5, 'BB'} trong quan hệ MONHOC và xóa bộ {'0310000', 'CT110', 'Nguyễn Ngọc', '1 Lê Hồng Phong, Đà Lạt', 9} trong quan hệ SINHVIEN. Khi đó thông tin về sinh viên sẽ bị mất.

Để khắc phục những bất lợi trên, quan hệ SINHVIEN có thể tách thành 2 quan hệ SINHVIEN (MSSV, TenSV, DiaChi) và KETQUATHI (MSSV, MaMH, Diem). Như vậy, 3 quan hệ trên đều đã ở dạng chuẩn thứ 2.

### 1.3. Dạng chuẩn 3

#### 1.3.1. Định nghĩa

##### Định nghĩa 1

Với một lược đồ  $Q$ ;  $X, Y$  là hai tập con của  $Q^+$ ;  $A$  là một thuộc tính. Khi đó  $A$  được gọi là phụ thuộc bắc cầu vào  $X$  nếu thỏa:

$$(1) X \rightarrow Y, Y \rightarrow A$$

$$(2) Y \not\rightarrow X$$

$$(3) A \notin XY$$

Lược đồ  $Q$  ở dạng chuẩn 3 nếu mọi thuộc tính không khóa đều không phụ thuộc bắc cầu vào khóa.

Việc kiểm tra dạng chuẩn 3 theo định nghĩa trên sẽ khó cài đặt. Ta có thể thực hiện kiểm tra dạng chuẩn 3 theo định nghĩa sau:

##### Định nghĩa 2

Lược đồ  $Q$  ở dạng chuẩn 3 nếu mọi phụ thuộc hàm  $X \rightarrow A \in F^+$ , với  $A \notin X$  đều có:

$$(1) X \text{ là siêu khóa, hoặc}$$

$$(2) A \text{ là thuộc tính khóa}$$

#### 1.3.2. Kiểm tra dạng chuẩn 3

Từ định nghĩa 2, để kiểm tra dạng chuẩn 3 thực hiện các bước sau:

Bước 1: Tìm mọi khóa của  $Q$

Bước 2: Phân rã vế phải của mọi phụ thuộc hàm trong  $F$  để tập  $F$  trở thành tập phụ thuộc hàm có vế phải một thuộc tính

Bước 3: Nếu mọi phụ thuộc hàm  $X \rightarrow A \in F$ , mà  $A \notin X$  đều thỏa

$$(1) X \text{ là siêu khóa (vế trái chứa một khóa), hoặc}$$

$$(2) A \text{ là thuộc tính khóa (vế phải là tập con của khóa)}$$

thì  $Q$  đạt dạng chuẩn 3, ngược lại  $Q$  không đạt dạng chuẩn 3.

**Ví dụ.** Cho  $Q(A, B, C, D), F = \{AB \rightarrow D, C \rightarrow D\}$

Bước 1:  $Q$  có một khóa là  $ABC$

Bước 2: Mọi phụ thuộc hàm trong F đều đã có vế phải một thuộc tính.

Bước 3: Với  $AB \rightarrow D$ , nhận thấy rằng  $D \notin ABC$  có

- Vế trái (AB) không phải là siêu khóa.
- Hơn nữa vế phải (D) không là thuộc tính khóa

Vậy Q không đạt dạng chuẩn 3.

## 1.4. Dạng chuẩn BC (Boyce Codd)

### 1.4.1. Định nghĩa

Lược đồ Q ở dạng chuẩn BC nếu mọi phụ thuộc hàm  $X \rightarrow A \in F^+$ , với  $A \notin X$  đều có X là siêu khóa.

### 1.4.2. Kiểm tra dạng chuẩn BC

Từ định nghĩa, để kiểm tra dạng chuẩn BC thực hiện các bước sau:

Bước 1: Tìm mọi khóa của Q

Bước 2: Phân rã vế phải của mọi phụ thuộc hàm trong F để tập F trở thành tập phụ thuộc hàm có vế phải một thuộc tính

Bước 3: Nếu mọi phụ thuộc hàm  $X \rightarrow A \in F$ , mà  $A \notin X$  đều thỏa X là siêu khóa (vế trái chứa một khóa), thì Q đạt dạng chuẩn BC, ngược lại Q không đạt dạng chuẩn BC.

**Ví dụ.** Cho Q (A, B, C, D, E, I),  $F = \{ACD \rightarrow EBI, CE \rightarrow AD\}$

Bước 1: Q có hai khóa là  $\{ACD, CE\}$

Bước 2: Phân rã vế phải của các phụ thuộc hàm trong F, ta có:

$$F = \{ACD \rightarrow E, ACD \rightarrow B, ACD \rightarrow I, CE \rightarrow A, CE \rightarrow D\}$$

Bước 3: Mọi phụ thuộc hàm trong F đều có vế trái là một siêu khóa

Vậy Q đạt dạng chuẩn BC.

## 1.5. Kiểm tra dạng chuẩn

### Kiểm tra dạng chuẩn của lược đồ quan hệ Q

Bước 1: Tìm mọi khóa của Q

Bước 2: Kiểm tra dạng chuẩn BC, nếu đúng thì Q đạt dạng chuẩn BC,

ngược lại qua bước 3.

Bước 3: Kiểm tra dạng chuẩn 3, nếu đúng thì Q đạt dạng chuẩn 3,

ngược lại qua bước 4.

Bước 4: Kiểm tra dạng chuẩn 2, nếu đúng thì Q đạt dạng chuẩn 2,

ngược lại Q đạt dạng chuẩn 1.

### Kiểm tra dạng chuẩn của lược đồ CSDL

Dạng chuẩn của một lược đồ CSDL là dạng chuẩn thấp nhất trong các dạng chuẩn của các lược đồ quan hệ con.

## 2. Phép phân rã

Mục tiêu của việc thiết kế CSDL quan hệ là tạo ra một tập các lược đồ quan hệ cho phép chúng ta lưu trữ thông tin không có những dư thừa không cần thiết và truy tìm thông tin một cách dễ dàng, chính xác. Việc phân rã một lược đồ thành những lược đồ con đều mong muốn đạt được bảo toàn thông tin và bảo toàn phụ thuộc.

### 2.1. Phân rã bảo toàn thông tin

Cho lược đồ quan hệ Q (TenNCC, DiaChiNCC, SanPham, DonGia)

Phân rã Q thành Q1 và Q2 như sau:

Q1 (TenNCC, SanPham, DonGia)

Q2 (TenNCC, DiaChiNCC)

Khi đó ta có các thể hiện sau:

Q	TenNCC	DiaChiNCC	SanPham	DonGia
	Nguyễn Mai	10 Nguyễn Công Trứ	Bánh xốp	10.000
	Nguyễn Mai	10 Nguyễn Công Trứ	Kẹo mè	20.000
	Nguyễn Mai	20 Nguyễn Văn Trỗi	Kẹo mè	20.000

Q1	TenNCC	SanPham	DonGia
	Nguyễn Mai	Bánh xốp	10.000

Nguyễn Mai	Kẹo mè	20.000
------------	--------	--------

Q2

TenNCC	DiaChiNCC
Nguyễn Mai	10 Nguyễn Công Trứ
Nguyễn Mai	20 Nguyễn Văn Trỗi

$Q1 \triangleright \triangleleft Q2$

TenNCC	DiaChiNCC	SanPham	DonGia
Nguyễn Mai	10 Nguyễn Công Trứ	Bánh xốp	10.000
Nguyễn Mai	10 Nguyễn Công Trứ	Kẹo mè	20.000
Nguyễn Mai	20 Nguyễn Văn Trỗi	Bánh xốp	10.000
Nguyễn Mai	20 Nguyễn Văn Trỗi	Kẹo mè	20.000

Như vậy kết quả thể hiện Q và  $Q1 \triangleright \triangleleft Q2$  là khác nhau, khi đó ta nói phép phân rã gọi là không bảo toàn thông tin (mất mát thông tin).

### Định nghĩa

Q là lược đồ quan hệ, Q1, Q2 là hai lược đồ con có:

$$Q_1^+ \cap Q_2^+ = X$$

$$Q_1^+ \cup Q_2^+ = Q^+$$

Khi đó Q được phân rã thành hai lược đồ con Q1, Q2 là phép phân rã bảo toàn thông tin nếu với r là thể hiện bất kỳ của Q ta có:

$$r = r.Q_1 \triangleright \triangleleft r.Q_2$$

(r là kết quả của phép kết tự nhiên của các hình chiếu của nó trên Q1, Q2)

### 2.2. Phân rã bảo toàn phụ thuộc hàm

Một vấn đề cần quan tâm khi phân rã lược đồ Q thành các lược đồ con Qi với tập các Fi tương ứng được tính từ tập phụ thuộc hàm F. Phép phân rã bảo toàn phụ thuộc (giữ lại



phụ thuộc) nếu với  $r_i$  là thể hiện của  $Q_i$  thoả điều kiện:  $r_i$  chỉ thoả những phụ thuộc hàm  $X \rightarrow Y \in F^+$  với  $XY \subseteq Q_i^+$

### Định nghĩa

Gọi  $Q_1, Q_2, \dots, Q_n$  là phân rã của lược đồ quan hệ  $Q$ , tập phụ thuộc hàm  $F$  trên  $Q$ . Hình chiếu của  $F$  trên một tập các thuộc tính  $Q_i^+$  ký hiệu  $\Pi_{Q_i^+}(F)$  là tập các phụ thuộc hàm  $X \rightarrow Y \in F^+$  với  $XY \subseteq Q_i^+$

$$\Pi_{Q_i^+}(F) = F_i^+ = \{X \rightarrow Y \mid X \rightarrow Y \in F^+ \text{ và } XY \subseteq Q_i^+\}$$

Khi đó phân rã là bảo toàn tập phụ thuộc hàm  $F$  nếu

$$F \equiv \cup \Pi_{Q_i^+}(F)$$

## 3. Thiết kế CSDL bằng cách phân rã

### 3.1. Phân rã thành dạng chuẩn BC (hoặc dạng chuẩn 3) bảo toàn thông tin

#### 3.1.1. Thuật toán

Bước 1: Tìm tất cả các khóa của  $Q$

Bước 2: Tìm phụ thuộc hàm  $X \rightarrow Y \in F$  có  $X$  không là siêu khóa và  $Y$  không chứa thuộc tính khóa.

- Nếu tìm thấy thì tách  $Q$  thành  $Q_1$  và  $Q_2$  theo cách:
  - $Q_1 = Q[XY]$ ;  $F_1 \equiv \Pi_{Q_1}(F)$  (tìm bao đóng của tất cả các tập con của  $XY$  để tính  $F_1$ ). Tiếp tục phân rã ( $Q_1, F_1$ ).
  - $Q_2 = Q[Q^+ - Y]$ ;  $F_2 \equiv \Pi_{Q_2}(F)$  (tìm bao đóng của tất cả các tập con của  $(Q^+ - Y)$  để tính  $F_2$ ). Tiếp tục phân rã ( $Q_2, F_2$ ).
- Nếu không tìm thấy thì xét dạng chuẩn  $Q_i$ :
  - Nếu mọi phụ thuộc hàm trong  $F_i$  đều có vế trái là siêu khóa thì  $Q_i$  đạt dạng chuẩn BC
  - Nếu có phụ thuộc hàm trong  $F_i$  có vế trái không là siêu khóa và vế phải là thuộc tính khóa thì  $Q_i$  đạt dạng chuẩn 3

#### 3.1.2. Ví dụ

##### Ví dụ 1

Cho  $Q$  (SDIM),  $F = \{SI \rightarrow D, SD \rightarrow M\}$

Bước 1: Q có một khóa là {SI}

Bước 2: Phụ thuộc hàm  $SD \rightarrow M$  có SD không là siêu khóa nên tách:

$$Q1 = (SDM); F1 = \{SD \rightarrow M\}$$

$$Q2 = (SDI); F2 = \{SI \rightarrow D\}$$

Để tìm tập phụ thuộc hàm F1, F2 cần tính bao đóng của mọi tập con.

*Tìm F1:* với  $Q1 = (SDM)$ : bao đóng của mọi tập con

$$S_F^+ = S$$

$$D_F^+ = D$$

$$M_F^+ = M$$

$$SD_F^+ = SDM$$

$$SM_F^+ = SM$$

$$DM_F^+ = DM$$

$$SDM_F^+ = SDM$$

$$F_1 = \Pi_{Q1}(F) = \{SD \rightarrow M, SD \rightarrow SM, SD \rightarrow DM, SD \rightarrow SDM\}$$

$$\Rightarrow F1 = \{SD \rightarrow M\}$$

*Tìm F2:* với  $Q2 = (SDI)$ : bao đóng của mọi tập con

$$S_F^+ = S$$

$$D_F^+ = D$$

$$I_F^+ = I$$

$$SD_F^+ = SDM$$

$$SI_F^+ = SIDM$$

$$DI_F^+ = DI$$

$$SDI_F^+ = SDIM$$

$$F_2 = \Pi_{Q_2}(F) = \{SI \rightarrow D, SI \rightarrow SD, SI \rightarrow DI, SI \rightarrow SDI\}$$

$$\Rightarrow F_2 = \{SI \rightarrow D\}$$

Bước 3: Mọi phụ thuộc hàm trong F1 và F2 đều có vế trái là một siêu khóa nên Q1 và Q2 đạt dạng chuẩn BC.

## Ví dụ 2

Cho Q (ABCDE),  $F = \{BC \rightarrow A, C \rightarrow D, AE \rightarrow B, B \rightarrow D, B \rightarrow E\}$

Bước 1: 2 khóa là {CB, CAE}

Bước 2: Từ  $C \rightarrow D$  tách thành:

- $Q_1(CD)$ ,  $F_1 = \{C \rightarrow D\}$ , Khóa C. Đạt dạng chuẩn BC.
- $Q_2(CABE)$ ,  $F_2 = \{B \rightarrow E, CB \rightarrow A, AE \rightarrow B\}$

Chi tiết tính F1, F2 như sau:

*Tìm F1:* với  $Q_1 = (CD)$

$$C_F^+ = CD$$

$$D_F^+ = D$$

$$CD_F^+ = CD$$

$$\text{Vậy } F_1 = \{C \rightarrow D\}$$

*Tìm F2:* với  $Q_2(CABE)$ :

$$C_F^+ = CD$$

$$A_F^+ = A$$

$$B_F^+ = BDE$$

$$E_F^+ = E$$

$$CA_F^+ = CAD$$

$$CB_F^+ = CBADE$$

$$CE_F^+ = CED$$

$$AB_F^+ = ABDE$$

$$AE_F^+ = AEBD$$

$$BE_F^+ = BED$$

$$CAB_F^+ = CABDE$$

$$CAE_F^+ = CAEBD$$

$$CBE_F^+ = CBEAD$$

$$ABE_F^+ = ABED$$

$$CABE_F^+ = CABED$$

$$F_2 = \{B \rightarrow E, CB \rightarrow AE, AB \rightarrow E, AE \rightarrow B, CAB \rightarrow E, CAE \rightarrow B, CBE \rightarrow A\}$$

$$\text{Vậy } F_2 = \{B \rightarrow E, CB \rightarrow A, AE \rightarrow B\}$$

Xét dạng chuẩn Q2:

Khoá Q2:  $\{CB, CAE\}$ . Vậy Q<sub>2</sub> đã đạt dạng chuẩn 3.

### 3.1.3. Cải tiến thuật toán

Nhận thấy rằng trong tìm phụ thuộc hàm hình chiếu trên Q<sub>i</sub>, xét tập một con X<sub>i</sub>, nếu  $(X_i)_F^+ = Q^+$ , khi đó nếu tiếp tục xét các tập  $X_j : X_i \subset X_j$ , thì hiển nhiên  $(X_j)_F^+ = Q^+$ , và cuối cùng thì phụ thuộc hàm có vế trái X<sub>j</sub> cũng sẽ bị loại để chỉ chọn phụ thuộc hàm có vế trái X<sub>i</sub>.

Do đó, khi tìm phụ thuộc hàm hình chiếu trên Q<sub>i</sub>, xét tập một con X<sub>i</sub>, nếu  $(X_i)_F^+ = Q^+$ , thực hiện loại bỏ các tính toán cho các trường hợp  $X_j : X_i \subset X_j$ .

**Ví dụ:** với tìm F<sub>2</sub> như ví dụ trên:

$$Q_2(CABE), F = \{BC \rightarrow A, C \rightarrow D, AE \rightarrow B, B \rightarrow D, B \rightarrow E\}$$

$$C_F^+ = CD$$

$$A_F^+ = A$$

$$B_F^+ = BDE$$

$$E_F^+ = E$$

$$CA_F^+ = CAD$$

$$CB_F^+ = CBADE, \text{ loại các tập CAB, CBE, CABE}$$

$$CE_F^+ = CED$$

$$AB_F^+ = ABDE$$

$$AE_F^+ = AEBD$$

$$BE_F^+ = BED$$

$$ABE_F^+ = ABED$$

$$F_2 = \{B \rightarrow E, CB \rightarrow AE, AB \rightarrow E, AE \rightarrow B\}$$

$$\text{Vậy } F_2 = \{B \rightarrow E, CB \rightarrow A, AE \rightarrow B\}$$

### 3.2. Phân rã thành dạng chuẩn 3 vừa bảo toàn thông tin vừa bảo toàn phụ thuộc hàm

Bước 1: Tìm phủ tối thiểu của F.

Bước 2: Loại bỏ tất cả các thuộc tính của Q không liên quan đến một phụ thuộc hàm nào của PTT(F).

Bước 3: Nếu có một phụ thuộc hàm trong PTT(F) liên quan đến mọi thuộc tính của Q thì không thể phân rã. Ngược lại, qua bước 4.

Bước 4: Gom nhóm những phụ thuộc hàm có cùng vế trái. Với mỗi nhóm phụ thuộc hàm có cùng vế trái, tạo thành một lược đồ con.

Bước 5: Kiểm tra các lược đồ con có thỏa dạng chuẩn 3 chưa, nếu chưa thì áp dụng bước 4 để phân rã tiếp.

**Ví dụ.** Cho Q (CTHRSG),  $F = \{C \rightarrow T, HR \rightarrow C, HT \rightarrow R, CS \rightarrow G, HS \rightarrow R\}$

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

Ta có kết quả phân rã Q1(CT), Q2(HRC), Q3(HTR), Q4(CSG), Q5(HRS)

## 4. Bài tập

Bài tập 7, 8, 9 trong chương 7, với yêu cầu:

- Phân rã thành dạng chuẩn BC hoặc dạng chuẩn 3 bảo toàn thông tin

- Phân rã thành dạng chuẩn 3 bảo toàn thông tin và bảo toàn phụ thuộc hàm

---

## ***Tài Liệu Tham Khảo***

1. Bài giảng Cơ sở dữ liệu, Nguyễn Hữu Tân, Đại Học Đà Lạt, 2004.
2. Bài tập cơ sở dữ liệu, Nguyễn Xuân Huy – Lê Hoài Bắc, NXB thống kê, 2003.
3. Giáo trình Cơ sở dữ liệu, Nguyễn Đăng Ty - Đỗ Phúc, Đại Học Quốc gia Tp. Hồ Chí Minh, 2001.
4. Nhập môn Cơ sở dữ liệu quan hệ, Lê Tiến Vương, NXB Khoa học và Kỹ thuật, 1994.
5. David Maier, *The theory of Relational Databases*, Computer Science Press, Rockville, 1983.