

Chương 5. RÀNG BUỘC TOÀN VỆ

Mục đích

Trình bày ràng buộc toàn vẹn và toàn vẹn dữ liệu.

Yêu cầu

Hiểu được ràng buộc toàn vẹn, ý nghĩa của ràng buộc toàn vẹn.

Nắm các yếu tố của ràng buộc toàn vẹn và phân loại ràng buộc toàn vẹn.

Trong chương này chúng ta trình bày một vấn đề quan trọng trong phân tích - thiết kế và khai thác cơ sở dữ liệu: Ràng buộc toàn vẹn và kiểm tra ràng buộc toàn vẹn. Ràng buộc toàn vẹn cần được quan tâm đúng mức nhằm tăng tính ngữ nghĩa của cơ sở dữ liệu, thể hiện rõ mối liên hệ giữa các tập thực thể trong mô hình dữ liệu. Đồng thời khi phân tích rõ các ràng buộc toàn vẹn ta đã chỉ ra các yêu cầu quản lý dữ liệu của bài toán thực tế đặt ra, cũng như nhằm giảm thiểu các hậu quả nghiêm trọng khi lưu trữ và xử lý dữ liệu.

5.1. Định nghĩa ràng buộc toàn vẹn

Trong một CSDL luôn luôn có nhiều mối liên hệ, nhiều sự ràng buộc qua lại giữa các thuộc tính, giữa các bộ với nhau. Các mối liên hệ ràng buộc này là những điều kiện bất biến mà tất cả các bộ của những quan hệ có liên quan trong CSDL đều phải thỏa mãn ở bất kỳ thời điểm nào. Những điều kiện bất biến đó được gọi là “ràng buộc toàn vẹn”.

Trong thực tế, ràng buộc toàn vẹn thường là các qui tắc quản lý được áp đặt lên trên các đối tượng của thế giới thực. Việc xác định rõ ràng và đầy đủ các ràng buộc toàn vẹn trên cơ sở dữ liệu không những nâng cao tính ngữ nghĩa của cơ sở dữ liệu mà còn là những ràng buộc, yêu cầu để phân tích thiết kế hệ thống thông tin. Nhằm bảo đảm tính kết dính của các thành phần cấu tạo nên CSDL, tính nhất quán của dữ liệu và làm cho CSDL luôn biểu diễn đúng ngữ nghĩa thực tế.

Ví dụ: Cho CSDL quản lý sinh viên, gồm các lược đồ quan hệ sau.

1. Sinh viên (MASV, HOSV, TENSU, NGSINH, MAKH, HBONG)
2. Kết quả (MASV, MAMH, LANTHI, DIEM).
3. Khoa (MAKH, TENKH, DIACHI).

Trong CSDL này, chúng ta có một số ràng buộc toàn vẹn sau:

R₁: Mỗi sinh viên có một mã số (MASV) riêng biệt không trùng với bất kì với một sinh viên nào khác.

R₂: Mỗi sinh viên phải đăng kí vào một khoa của một trường nhất định.

R₃: Mỗi sinh viên chỉ được thi tối hai lần cho một môn học...

Công việc kiểm tra ràng buộc toàn vẹn có thể tiến hành vào một trong các thời điểm sau:

+ Kiểm tra ngay sau khi thực hiện một thao tác cập nhập CSDL (thêm, sửa, huỷ,...). Thao tác cập nhập được xem là hợp lệ nếu như nó không vi phạm bất cứ một ràng buộc toàn vẹn nào (nghĩa là không làm cho CSDL rơi vào tình trạng mất tính kết dính).

+ Kiểm tra định kì hay đột xuất. Đối với những trường hợp vi phạm ràng buộc toàn vẹn hệ thống sẽ có xử lí ngầm hoặc yêu cầu người sử dụng xử lí những sai sót một cách tường minh.

5.2. Các yếu tố của ràng buộc toàn vẹn

Một ràng buộc toàn vẹn có 3 yếu tố:

- + Nội dung.
- + Bối cảnh.
- + Tầm ảnh hưởng.

5.2.1. Nội dung

Nội dung của ràng buộc toàn vẹn có thể được biểu diễn bằng: ngôn ngữ tự nhiên, thuật giải (mã giả, ngôn ngữ tựa lập trình), đại số quan hệ, phụ thuộc hàm...

Ví dụ 1

Với ràng buộc biểu diễn bằng ngôn ngữ tự nhiên:

“Mức lương của một người nhân viên không được vượt quá lương của trưởng phòng”, có biểu diễn bằng ngôn ngữ hình thức như sau:

$\forall t \in \text{NHANVIEN (}$

$\exists u \in \text{PHONGBAN (} \exists v \in \text{NHANVIEN (}$

$u.\text{TRPHG} = v.\text{MANV} \wedge$

$u.\text{MAPHG} = t.\text{PHG} \wedge$

$$t.LUONG \leq v.LUONG)))$$

Ví dụ 2. Trong lược đồ CSDL $T_{sinhviên}$.

$$R_1: \forall SV1 \in T_{sinhviên}, \forall SV2 \in T_{sinhviên}.$$

$$SV1 \triangleleft SV2 \Rightarrow SV1.MASV \triangleleft SV2.MASV.$$

$$R_2: T_{sinhviên}[MAKH] \subseteq T_{Khoa}[MAKH].$$

$$R_3: \forall SV \in T_{kếtquả}.$$

$Card(\{k \in T_{kếtquả} \mid K.MASV = SV.MASV\}) \leq 2.$ ($Card(A)$ = lực lượng của tập A).

Việc biểu diễn nội dung của ràng buộc liên quan đến vấn đề đặc tả, xin xem thêm các sách về đặc tả.

5.2.2. Bối cảnh

Bối cảnh của ràng buộc toàn vẹn là những quan hệ mà ràng buộc toàn vẹn có hiệu lực, nói cách khác đó là những quan hệ cần sử dụng để kiểm tra ràng buộc toàn vẹn. Bối cảnh của ràng buộc toàn vẹn có thể là một hoặc nhiều quan hệ.

Ví dụ Bối cảnh của ràng buộc toàn vẹn trong R_3 chỉ là một quan hệ $T_{kếtquả}$ (mỗi sinh viên (MASV) thì tối đa hai lần (LANTHI) thì hai thuộc tính MASV, LANTHI $\in T_{kếtquả}$). Nhưng ở R_2 , bối cảnh của ràng buộc toàn vẹn phải là hai quan hệ (muốn biết sinh viên có MASV thuộc khoa nào thì $T_{sinhviên}[MAKH] = T_{khoa}[MAKH]$ từ đó tìm ra tên khoa).

5.2.3. Tầm ảnh hưởng

Trong quá trình thiết kế CSDL cần chỉ ra các ảnh hưởng của các thao tác trên dữ liệu có ảnh hưởng như thế nào đến các ràng buộc toàn vẹn, các ràng buộc nào bị vi phạm. Bảng tầm ảnh hưởng là một trong số các cách để thể hiện việc ấy.

Ví dụ. Ràng buộc R_i với các thao tác thêm, sửa, xóa và bối cảnh là các quan hệ $T_{Qi1}, T_{Qi2}, \dots, T_{Qik}$ ta thành lập bảng tầm ảnh hưởng sau

R_i	Thêm	Sửa	Xóa
T_{Qi1}	+	+	-
T_{Qi2}	-	-	+
.....			
T_{Qik}	+	-	-

Trong đó: dấu "+" được hiểu là cần kiểm tra ràng buộc toàn vẹn R_i .
 dấu "-" được hiểu là không cần kiểm tra ràng buộc toàn vẹn R_i .

Từ bảng tầm ảnh hưởng tổng quát ta lập bảng tầm ảnh hưởng cho các quan hệ R_1, R_2, R_3 trong ví dụ 1.

R_i	Thêm	Sửa	Xoá
$T_{sinh\ viên}$	+	-(*)	-

(*) Qui ước là không được phép sửa đổi giá trị thuộc tính MASV.

R_i	Thêm	Sửa	Xoá
$T_{sinh\ viên}$	+	+	-
T_{Khoa}	-	-(*)	+

(*) Qui ước là không được sửa đổi giá trị thuộc tính MAKH.

R_i	Thêm	Sửa	Xoá
$T_{kết\ quả}$	+	-(*)	-

(*) Qui ước là không được sửa đổi giá trị thuộc tính MASV, MAMH, LANTHI.

Ta có “bảng tầm ảnh hưởng tổng hợp” cho CSDL T như sau.

	$T_{sinh\ viên}$			T_{Khoa}			$T_{kết\ quả}$		
	Thêm	Sửa	Xoá	Thêm	Sửa	Xoá	Thêm	Sửa	Xoá
R_1	+	-	-						
R_2	+	+	-	-	-	+			
R_3							+	-	-

5.3. Phân loại ràng buộc toàn vẹn

Trước hết phải khẳng định rằng không thể xác định được hết tất cả các RBTV trên một CSDL, do đó việc phân loại dưới đây chỉ nhằm đưa ra một cơ sở để nhận diện các RBTV có thể trên CSDL.

Ràng buộc toàn vẹn có thể chia làm hai loại chính:

- + Ràng buộc toàn vẹn có bối cảnh là một quan hệ.
- + Ràng buộc toàn vẹn có bối cảnh gồm nhiều quan hệ.

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

5.3.1.1. Ràng buộc toàn vẹn về miền giá trị

Những ràng buộc về miền giá trị của các thuộc tính xuất phát từ cơ sở thực tế như về kiểu, độ rộng, miền giá trị của thuộc tính. Những ràng buộc này đảm bảo rằng giá trị được lưu giữ trong 1 cột phải nằm trong một miền trị hợp lệ được xác định trước. Trong SQL ràng buộc miền trị được hiện thực bằng ràng buộc CHECK.

Ví dụ 1: Trong lược đồ quan hệ $T_{\text{kết quả}}$, ta có: $\text{MGT}[\text{DIEM}] = [0..10]$.

Ngoài ra, Phòng đào tạo có thể quy định thêm một số ràng buộc khác là điểm thi có độ chính xác đến $0,5^d$. Khi đó ta có $\forall kq \in T_{\text{kết quả}}. ((kq.DIEM * 4) \bmod 2) = 0$.

Ví dụ 2: Trên quan hệ $\text{NHANVIEN}(\text{MANV}, \text{TENN}, \text{LUONG}, \text{TAMUNG}, \text{CONLAI})$. Ràng buộc về miền giá trị là: $\text{TAMUNG} \leq \text{LUONG}$.

Ví dụ 3: Trong lược đồ quan hệ $\text{CONGNHAN}(\text{MACN}, \text{MUCLG}, \text{NGXEPLG})$. Khi một công nhân thay đổi mức lương, mức lương đó có thể tăng hoặc giảm nhưng rõ ràng thuộc tính NGXEPLG bắt buộc phải tăng theo thời gian. Nếu gọi S là phép sửa đổi lương(NGXEPLG và MUCLG) của các công nhân.

$T_{\text{công nhân}}$ là một quan hệ trên lược đồ quan hệ CONGNHAN.

$S: T_{\text{công nhân}} \rightarrow T_{\text{công nhân}}$

Ta có $\forall cn \in T_{\text{công nhân}}. cn.NGXEPLG \leq S(cn).NGXEPLG$

5.3.1.2. Ràng buộc liên thuộc tính

Ràng buộc liên thuộc tính là ràng buộc giữa các thuộc tính trong một lược đồ quan hệ.

Ví dụ 1: Cho lược đồ quan hệ HOADON, ta có một ràng buộc toàn vẹn liên thuộc tính như sau:

“Hàng hoá chỉ được xuất kho sau khi đã được lập hoá đơn”

$\forall hd \in T_{\text{HOADON}}$

$hd.NGAYHD \leq hd.NGAYXUAT \quad (*) \quad (\text{thứ tự thời gian}).$

Nếu giá trị của một thuộc tính A (thuộc tính dẫn xuất) được tính toán từ các thuộc tính khác trong cùng một lược đồ quan hệ với A thì ta cũng có một ràng

buộc liên thuộc tính nhưng trong quá trình lưu trữ chúng ta có thể loại bỏ thuộc tính A ra khỏi lược đồ quan hệ.

Ví dụ 2: Trong lược đồ quan hệ CTIETHD nếu bổ sung thêm thuộc tính THANHTIEN thì ta sẽ có một ràng buộc toàn vẹn liên thuộc tính.

$$\forall \text{cthd} \in T_{\text{CTIETHD}}$$

$$\text{cthd.THANHTIEN} = \text{cthd.SLBAN} * \text{cthd.GIABAN}.$$

nhưng nếu thiết kế lược đồ quan hệ như trên sẽ làm tăng chi phí lưu trữ và chi phí kiểm tra ràng buộc toàn vẹn một cách vô ích vì thế nên loại bỏ thuộc tính trên.

5.3.1.3. Ràng buộc toàn vẹn liên bộ (ràng buộc thực thể)

Sự tồn tại của một hay nhiều bộ phụ thuộc vào sự tồn tại của một hay nhiều bộ khác trong cùng quan hệ. Các trường hợp đặc biệt là ràng buộc khóa chính (PRIMARY KEY) và ràng buộc duy nhất (UNIQUE). Đây là loại ràng buộc toàn vẹn rất phổ biến, nó có mặt trong mọi lược đồ quan hệ của CSDL và thường được các hệ quản trị CSDL tự động kiểm tra.

Ví dụ: Với r là một quan hệ của Khách ta có ràng buộc toàn vẹn sau

$$R_1: \quad \forall t_1, t_2 \in r, t_1. \text{MAKH} \neq t_2. \text{MAKH}$$

R ₁	Thêm	Sửa	Xóa
R	+	+	-

5.3.2. Ràng buộc toàn vẹn có bối cảnh gồm nhiều mối quan hệ

5.3.2.1. Ràng buộc toàn vẹn về phụ thuộc tồn tại (ràng buộc tham chiếu)

Giá trị xuất hiện tại các thuộc tính trong một quan hệ nào đó phải tham chiếu đến giá trị khóa chính của một quan hệ khác cho trước. Một trường hợp đặc biệt là ràng buộc khóa ngoại. RBTV tham chiếu còn gọi là phụ thuộc tồn tại thường có bối cảnh là hai quan hệ, nhưng có trường hợp suy biến thành một quan hệ.

Ví dụ 1:

Giả sử có cơ sở dữ liệu gồm các quan hệ sau:

Sinh viên: (1)

Mã số sinh viên	Tên sinh viên	Mã khoa
101	Lê Anh	M01

102	Nguyễn Hoài	P01
103	Vũ Tuấn	G01

Khoa (2)

Mã khoa	Tên khoa
M01	Information
P01	History
G01	Mathematics
F01	Fine Arts

Ở Mã số sinh viên, Tên sinh viên và mã khoa của sinh viên là khoá ngoại chỉ đến bảng Khoa chứa mã khoa và tên khoa. Bảng (2) là bảng được truy xuất qua khoá ngoại được xem như là bảng “cha”, bảng (1) tham chiếu đến bảng khác qua khoá ngoại được gọi là bảng “phụ thuộc”.

Nếu chúng ta muốn thêm một sinh viên vào bảng (1).

104	Nguyễn Tuấn Long	C01
-----	------------------	-----

Chúng ta có một khoá ngoại C01 không tham chiếu đến một khoá nào trong bảng (2). Chúng ta đã vi phạm ràng buộc toàn vẹn qua phép chèn trong bảng phụ thuộc.

Nếu chúng ta xoá một trường trong bảng “cha” chẳng hạn:

M01	Information
-----	-------------

Một lần nữa ta có khoá ngoại M01 trong bảng (1) nhưng không còn tham chiếu đến một khoá nào trong bảng “cha” như vậy chúng ta đã vi phạm ràng buộc tồn tại.

Ví dụ 2: Trong quan hệ $T_{\text{kết quả}}$. Sự tồn tại của bộ KQ = (MASV1, MON1, LANTHI, DIEM1) $\in T_{\text{kết quả}}$ là hoàn toàn phụ thuộc vào sự tồn tại của bộ sv $\in T_{\text{sinh viên}}$ sao cho: sv.MASV = MASV1, Ví dụ: sv = (MASV1, “Tran Hue”, “Chi”, False, 01/04/79, “B1 KTX Đống Đa”, “Tin học”, “DHSP”).

Nếu không tồn tại một bộ sv $\in T_{\text{sinh viên}}$ Sao cho: sv.MASV = MASV1 thì bộ kq = (MASV1, MON1, LANTHI, DIEM1) không được phép tồn tại trong $T_{\text{kết quả}}$.

Ví dụ 3: Trong quan hệ $T_{\text{sinh viên}}$ của một tình trạng của cơ sở dữ liệu T , sự tồn tại của bộ $sv = ("6760", "MASV1", "Tran Hue", "Chi", \text{False}, 01/04/79, "B1 KTX Đồng Đa", "Tin học", "DHSP") \in T_{\text{sinh viên}}$ là hoàn toàn phụ thuộc vào sự tồn tại của bộ không thuộc $T_{\text{khoa}} : k = ("Tin Học", "khoa Tin Học")$.

Nếu không tồn tại một bộ $k \in T_{\text{khoa}}$ sao cho $k.MAKH = "Tin học"$ thì bộ sv như trên không được phép tồn tại trong $T_{\text{sinh viên}}$.

Đối với bảng “cha” phép chèn không bị hạn chế, tuy nhiên khi một hàng bị xoá ra khỏi bảng “cha”, quy tắc cập nhật đồng thời (CASCADE) buộc hệ QTCSDL phải xoá tất cả các hàng trong bảng phụ thuộc có tham chiếu đến hàng bị xoá trong bảng cha, còn quy tắc SET NULL đặt khoá ngoại tương ứng trong bảng phụ thuộc thành NULL và một quy tắc khác (RESTRICT) cấm xoá các hàng trong bảng cha.

Đối với bảng phụ thuộc, phép xoá không bị hạn chế, nhưng khi một hàng đã được chèn vào bảng phụ thuộc thì tất cả các giá trị trong khoá ngoại của hàng đó phải là giá trị NULL hoặc phải có trong bảng cha tương ứng và khi cập nhật bảng phụ thuộc giá trị cập nhật phải là giá trị NULL hoặc phải có trong một hàng của bảng “cha” tương ứng.

5.3.2.2. Ràng buộc toàn vẹn về liên bộ liên quan hệ

Ràng buộc này có tác dụng đối với từng nhóm các bộ của nhiều quan hệ khác nhau.

Ví dụ:

HOADON(SOHD, MAKH, NGAYHD)

CTHD(SOHD, MAHH, DGIA, SLG)

Với RBTV “mỗi hóa đơn phải có ít nhất một chi tiết hóa đơn”

5.3.2.3. Ràng buộc toàn vẹn liên thuộc tính - liên quan hệ

Là mối liên hệ giữa các thuộc tính trong nhiều lược đồ quan hệ.

Ví dụ: Xét hai quan hệ sau : $T_{\text{đặt hàng}}, T_{\text{hoá đơn}}$

$\forall dh \in T_{\text{đặt hàng}}$

$\forall hd \in T_{\text{hoá đơn}}: hd.SOHD = dh.SODH,$

$dh.NGAYDH \leq hd.NGAYHD$

Cuối $\forall dh$

5.3.2.4. Ràng buộc toàn vẹn về thuộc tính tổng hợp

Thuộc tính A của lược đồ quan hệ Q được tính toán giá trị từ các thuộc tính của các lược đồ quan hệ khác.

Ví dụ: Tiền nợ của khách hàng bằng Σ giá trị của các hoá đơn bán cho A - Σ tiền thu của A.

$$\forall kh \in T_{Khách}.$$

$$kh.congno = \Sigma hd.trigia_hd - \Sigma pt.SOTIEN.$$

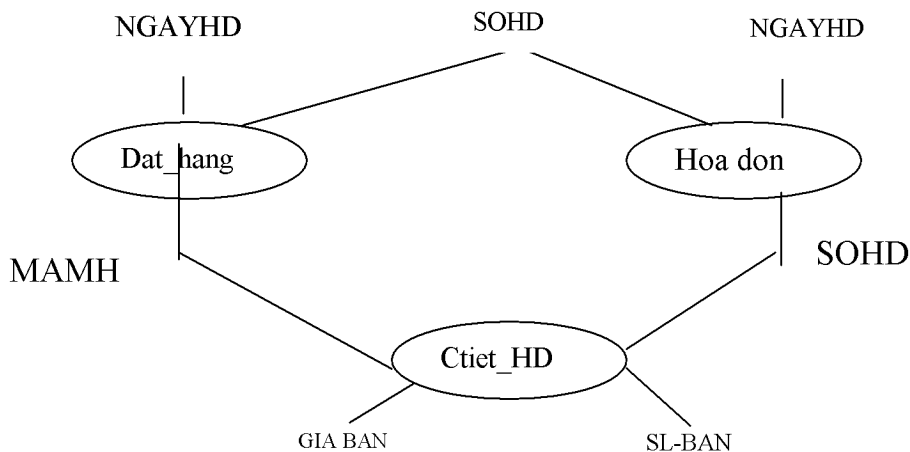
$$hd \in H_{kh} \quad ; \quad pt \in P_{kh}$$

Trong đó $H_{kh} = (T_{hoá\ đơn} \bowtie T_{đặt\ hàng}) : (MAKH = kh.MAKH)$ và $P_{kh} = T_{phiếu\ thu} : (MAKH = kh.MAKH)$

5.3.2.5. Ràng buộc toàn vẹn có chu trình trong đồ thị biểu diễn của lược đồ cơ sở dữ liệu

Một lược đồ cơ sở dữ liệu có thể biểu diễn bằng một đồ thị vô hướng. Trong đồ thị này có hai loại nút: nút thuộc tính và nút lược đồ quan hệ. Một cung vô hướng trong đồ thị nối một nút thuộc tính A với một nút lược đồ quan hệ Q có nghĩa: $A \in Q^+$.

Ví dụ: Xét một phần của đồ thị biểu diễn lược đồ cơ sở dữ liệu T gồm các lược đồ quan hệ *Đặt_hang*, *Hoa_don* và *Ctiet_hd*.



Trong trường hợp trên, ta thấy đồ thị biểu diễn có chứa một chu trình giữa 3 nút *Đặt_hang*, *Hoa_don* và *Ctiet_hd*. Lúc này xảy ra một ràng buộc toàn vẹn giữa các thuộc tính *MAMH*, *SOHD* trong cả ba quan hệ và hiện tượng chờ đợi để xác

định thuộc tính, chờ đợi để phụ thuộc tồn tại. Vì vậy mà trong một số hệ quản trị CSDL như FOXPRO chẳng hạn không cho phép đặt quan hệ vòng A với B, B với C và C với A.

5.4. Cài đặt ràng buộc toàn vẹn với SQL

Trong các hệ QTCSDL ngày nay đều có các kỹ thuật để thể hiện các RBTV, trong SQL Server các RBTV được cài đặt bởi các ràng buộc Primary key, Foreign key, Check constraint và các thủ tục Assertion, Trigger và Transaction.

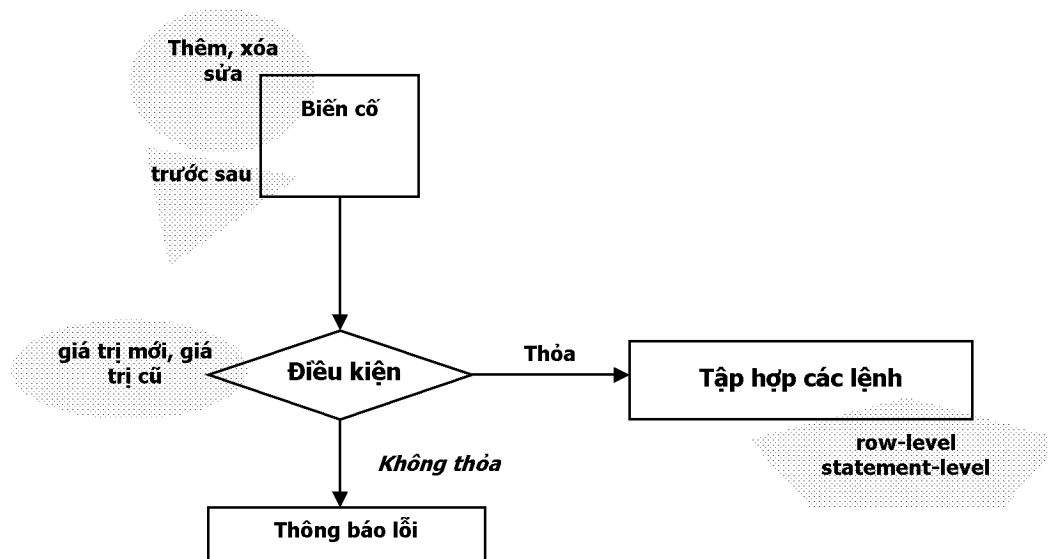
Ở đây trình bày sơ lược về Assertion, Trigger và Transaction, chi tiết về chúng có thể xem thêm trong các tài liệu chuyên khảo về SQL Server.

- Assertion là một biểu thức SQL luôn mang giá trị TRUE tại mọi thời điểm và người sử dụng cần cho biết cái gì phải đúng. Các câu lệnh liên quan trong SQL là:

```
CREATE      ASSERTION      <Tên_assertion>      CHECK
(<Điều_kiện>)
```

Và DROP ASSERTION <Tên_assertion>

- Trigger là tập hợp các lệnh được thực hiện tự động khi xuất hiện một biến cố nào đó (tựa như một ngắt (interup) trong hệ điều hành). Một trigger hoạt động theo sơ đồ sau:



Các câu lệnh liên quan trigger trong SQL là:

```
CREATE TRIGGER <Tên_trigger>
AFTER|BEFORE INSERT|UPDATE|DELETE ON <Tên_bảng>
REFERENCING
    NEW ROW|TABLE AS <Tên_1>
    OLD ROW|TABLE AS <Tên_2>
FOR EACH ROW | FOR EACH STATEMENT
WHEN (<Điều_kiện>)
    <Tập_lệnh_SQL>
và DROP TRIGGER <Tên_trigger>
```

- Transaction là tập các lệnh thực hiện một xử lý nào đó trong một ứng dụng CSDL, sao cho:

- Hoặc là tất cả các lệnh đều được thực hiện thành công
- Hoặc là không có lệnh nào được thực hiện

Ví dụ 1 Quá trình xử lý chuyển tiền trong ngân hàng

Giao tác Chuyển tiền

Giảm tiền trong tài khoản người gửi

Tăng tiền trong tài khoản người nhận

Nếu tất cả đều thành công thì hoàn tất giao tác

Ngược lại quay lui giao tác

Cuối giao tác.

Một giao tác phải đảm bảo:

Tính nguyên tử (atomicity)

Tính nhất quán của CSDL (consistency).

Các RBTV phải đảm bảo không bị vi phạm trong khi thực hiện giao tác, trước và sau khi thực hiện giao tác.

Ví dụ 2 Với RBTV mỗi hóa đơn phải có ít nhất một chi tiết hóa đơn.

Giao tác Thêm_hóa_đơn

Thêm HOADON

Thêm chi tiết thứ 1 vào CTHD

Thêm chi tiết thứ 2 vào CTHD

...

Nếu có một thao tác thêm thất bại thì

Quay lui giao tác

Ngược lại

Hoàn tất giao tác

Cuối nếu

Cuối giao tác.

Các DBMS thương mại cung cấp cách thức lưu trữ các hàm hay thủ tục nhằm thực hiện các giao tác. Các thủ tục này được lưu trữ trong lược đồ CSDL và được sử dụng trong các câu lệnh. Trong SQL cung cấp kỹ thuật ***Stored Procedure*** có cú pháp tổng quát như sau:

```
CREATE PROCEDURE <Tên_thủ_tục> <DS_tham_số>
AS
    Khai báo biến cục bộ
    Thân chương trình
GO
EXEC <Tên_thủ_tục> <DS_ tham_số>
```

Ví dụ 3 Với RBTV “mỗi trận đấu là cuộc thi đấu của đúng 2 đội”, ta có giao tác thể hiện việc thêm trận đấu như sau:

Giao tác Thêm_trận_đấu(t, s)

Thêm t vào THIDAU

Thêm s vào THIDAU

Nếu có một thao tác thất bại thì

Quay lui giao tác

Ngược lại

Hoàn tất giao tác

Cuối nếu

Cuối giao tác.

Và thủ tục tương ứng trong SQL sẽ là:

```
CREATE PROCEDURE Thêm_trận_đấu
    t THIDAU, s THIDAU
```

```

AS
begin tran
    Thêm t vào THIDAU
    If @@error<>0 rollback tran

    Thêm s vào THIDAU
    If @@error<>0      rollback tran
commit tran
GO
EXEC Thêm_trận_đấu x, y.

```

Tóm tắt chương 5

* Trong một CSDL luôn luôn có nhiều mối liên hệ, nhiều sự ràng buộc qua lại giữa các thuộc tính, giữa các bộ với nhau. Các mối liên hệ ràng buộc này là những điều kiện bất biến mà tất cả các bộ của những quan hệ có liên quan trong CSDL đều phải thỏa mãn ở bất kỳ thời điểm nào. Những điều kiện bất biến đó được gọi là “ràng buộc toàn vẹn”.

* Việc xác định rõ ràng và đầy đủ các ràng buộc toàn vẹn trên cơ sở dữ liệu không những nâng cao tính ngữ nghĩa của cơ sở dữ liệu mà còn là những ràng buộc, yêu cầu để phân tích thiết kế hệ thống thông tin. Nhằm bảo đảm tính kết dính của các thành phần cấu tạo nên CSDL, tính nhất quán của dữ liệu và làm cho CSDL luôn biểu diễn đúng ngữ nghĩa thực tế.

* Một ràng buộc toàn vẹn có 3 yếu tố: Nội dung, bối cảnh, tầm ảnh hưởng.

* Trước hết phải khẳng định rằng không thể xác định được hết tất cả các RBTV trên một CSDL, do đó việc phân loại trên đây chỉ nhằm đưa ra một cơ sở để nhận diện các RBTV có thể trên CSDL.

* Trong các hệ QTCSDL ngày nay đều có các kỹ thuật để thể hiện các RBTV, trong SQL Server các RBTV được cài đặt bởi các ràng buộc Primary key, Foreign key, Check constraint và các thủ tục Assertion, Trigger và Transaction.

Câu hỏi ôn tập và bài tập chương 5

1. Việc tổ chức kỳ thi tốt nghiệp của một khoa như sau:

Mỗi thí sinh có một Mã số sinh viên duy nhất (MASV), mỗi MASV xác định được các thông tin: họ và tên (HOTEN), ngày sinh (NGAYSINH), nơi sinh, nữ,phái, dân tộc.

Mỗi lớp có một mã lớp (MALOP) duy nhất , mỗi mã lớp xác định các thông tin: tên lớp (TENLOP), mỗi lớp chỉ thuộc sự quản lý của một khoa nào đó. Mỗi khoa có một mã khoa duy nhất (MAKHOA), mỗi mã khoa xác định tên khoa (TENKHOA).

Mỗi thí sinh đều phải dự thi tốt nghiệp ba môn. Mỗi môn thi có một mã môn thi (MAMT) duy nhất, mỗi mã môn thi xác định các thông tin: tên môn thi (TENMT), thời gian làm bài – được tính bằng phút (PHUT), ngày thi (NGAYTHI), buổi thi (BUOITHI), môn thi này là môn lý thuyết hay thực hành (LYTHUYET). Chú ý rằng, nếu một môn học được cho thi ở nhiều hệ thì được đặt MAMT khác nhau (chẳng hạn cả trung cấp và cao đẳng ngành công nghệ thông tin đều thi môn Cơ Sở Dữ Liệu), để diễn tả điều này, mỗi mã môn học cần phải được ghi chú (GHICHU) để cho biết môn thi đó dành cho khối nào trung cấp, hay cao đẳng). Mỗi thí sinh ứng với một môn thi có một điểm thi (DIEMTHI) duy nhất, điểm thi được chấm theo thang điểm 10 và có lấy điểm lẻ đến 0.5. Một thí sinh được coi là đậu tốt nghiệp nếu điểm thi của tất cả các môn của thí sinh đó đều lớn hơn hoặc bằng 5.

Trong một phòng thi có thể có thí sinh của nhiều lớp. Trong một kỳ thi, mỗi thí sinh có thể thi tại những phòng thi (PHONGTHI) khác nhau, chẳng hạn một thí sinh thi tốt nghiệp ba môn là Cơ sở dữ liệu, Lập trình C và Visual Basic thi môn Cơ Sở Dữ Liệu và Lập Trình C thi tại phòng A3.4, còn môn thực hành Visual Basic thi tại phòng máy BII6

Qua phân tích sơ bộ trên, ta có thể lập một lược đồ cơ sở dữ liệu như sau:

THISINH(MASV,HOTEN,NGAYSINH,MALOP)

LOP(MALOP,TENLOP)

MONTHI(MAMT,TENMT,LYTHUYET,PHUT,NGAYTHI,BUOITHI,GHICHU)

KETQUA(MASV,MAMT,DIEMTHI)

a. Tìm khoá cho mỗi lược đồ quan hệ trên.

b. Hãy phát biểu các ràng buộc toàn có trong cơ sở dữ liệu trên.

2. Bài toán quản lý điểm của sinh viên được phát biểu sơ bộ như sau:

Mỗi sinh viên cần quản lý các thông tin như: họ và tên (HOTENSV), ngày tháng năm sinh (NGAYSINH), giới tính (NU), nơi sinh (NOISINH), hộ khẩu thường trú (TINH). Mỗi sinh viên được cấp một mã số sinh viên duy nhất (MASV) để phân biệt với mọi sinh viên khác của trường, mỗi sinh viên chỉ thuộc về một lớp nào đó.

Mỗi lớp học có một mã số lớp (MALOP) duy nhất để phân biệt với tất cả các lớp học khác trong trường: có một tên gọi (TENLOP) của lớp, mỗi lớp chỉ thuộc về một khoa.

Mỗi khoa có một tên gọi (TENKHOA) và một mã số duy nhất (MAKHOA) để phân biệt với các khoa khác.

Mỗi môn học có một tên gọi (TENMH) cụ thể, được học trong một số đơn vị học trình (DONVIHT) và ứng với môn học là một mã số duy nhất (MAMH) để phân biệt với các môn học khác.

Mỗi giảng viên cần quản lý các thông tin: họ và tên (HOTENGV), cấp học vị (HOCVI), thuộc một chuyên ngành (CHUYENNGANH) và được gán cho một mã số duy nhất gọi là mã giảng viên (MAGV) để phân biệt với các giảng viên khác. Mỗi giảng viên có thể dạy nhiều môn ở nhiều khoa, nhưng chỉ thuộc về sự quản lý hành chính của một khoa.

Mỗi sinh viên với một môn học được phép thi tối đa 3 lần, mỗi lần thi (LANTHI), điểm thi (DIEMTHI).

Mỗi môn học ở mỗi lớp học chỉ phân công cho một giảng viên dạy (tất nhiên là một giảng viên thì có thể dạy nhiều môn ở một lớp).

Với bài toán trên thì các loại thực thể cần quản lý như: Sinh viên, Môn học, Khoa, Lớp, Giảng viên. Và các lược đồ quan hệ được chuyển từ mô hình thực thể quan hệ của bài toán là:

Sinhvien(MASV, HOTENSV, NU, NGAYSINH, NOISINH, TINH, MALOP)

Lop(MALOP, TENLOP, MAKHOA)

Khoa(MAKHOA, TENKHOA)

Monhoc(MAMH, TENMH, DONVIHT)

Giangvien(MAGV, HOTENGV, HOCVI, CHUYENNGANH, MAKHOA)

Ketqua(MASV, MAMH, LANTHI, DIEMTHI)

Phancong(MALOP, MAMH, MAGV)

Hãy phát biểu các ràng buộc toàn vẹn có trong lược đồ cơ sở dữ liệu trên.

3. Cho lược đồ cơ sở dữ liệu ở bài tập 1. Thực hiện các yêu cầu sau bằng ngôn ngữ SQL

a. Lập bảng điểm môn thi có mã môn thi là “CSDL02” cho tất các thí sinh có mã lớp là “CDTH2A”. danh sách cần MASV, HOTEN, NGAYSINH, DIEMTHI và được sắp xếp tăng dần theo MASV.

b. Hãy thống kê xem mỗi môn thi có bao nhiêu thí sinh có điểm thi lớn hơn hay bằng 5? Danh sách cần: MAMT, TENMT, GHICHU, SOLUONG trong đó số lượng (SOLUONG) là thuộc tính tự đặt.

c. Lập danh sách những thí sinh đậu tốt nghiệp (theo tiêu chuẩn đã phân tích ở trên), danh sách cần: MASV, HOTEN, NGAYSINH, DIEMTONG, trong đó DIEMTONG bằng tổng điểm thi của 3 môn thi, DIEMTONG là thuộc tính tự đặt.

d. Nếu cần mở rộng bài toán theo hai hướng; Thứ nhất là quản lý kỳ thi tốt nghiệp cho tất cả các khoa trong toàn trường, Thứ hai là quản lý thông tin về phòng thi (PHONGTHI) của mỗi thí sinh, thì lược đồ cơ sở dữ liệu trên cần phải được điều chỉnh như thế nào ?

e. Hãy phát biểu các ràng buộc toàn vẹn có trong lược đồ cơ sở dữ liệu trên.

CHƯƠNG 6. PHỤ THUỘC HÀM VÀ KHÓA

Mục đích

- Trình bày một số vấn đề nảy sinh trong thiết kế cơ sở dữ liệu quan hệ.
- Các khái niệm phụ thuộc hàm, các bài toán về khóa.

Yêu cầu

- *Nắm được và vận dụng một số kỹ năng trong các bài toán về phụ thuộc hàm, khóa trong lý thuyết thiết kế cơ sở dữ liệu quan hệ.*
- *Hiểu và khái quát được các ứng dụng của các bài toán về phụ thuộc hàm, phụ thuộc hàm và khóa trong các lĩnh vực của cơ sở dữ liệu.*
- *Hiểu và vận dụng được ý nghĩa của các bài toán nói trên trong thực tiễn.*

Trong một số tài liệu khác, thường trình bày các vấn đề liên quan đến phụ thuộc hàm, khóa, phân tách và chuẩn hoá... thành lý thuyết thiết kế cơ sở dữ liệu quan hệ. Nhưng nếu làm như vậy thì chương này sẽ quá nặng nề. Ở đây chúng tôi chia các vấn đề về lý thuyết thiết kế cơ sở dữ liệu quan hệ nói trên thành 3 chương, chương 6. Phụ thuộc hàm và Khóa, chương 7. Lý thuyết phân tách và chương 8. Chuẩn hoá, để vấn đề được rõ ràng hơn. Các kiến thức tổng quan về một số loại ràng buộc mở rộng trên cơ sở dữ liệu quan hệ như phụ thuộc đa trị, phụ thuộc kết nối được đưa vào chương 9. Tách riêng vấn đề phân tách và chuẩn hoá, theo chúng tôi sẽ giúp ta có nhìn tốt trong khi nghiên cứu cơ sở dữ liệu phân tán.

Chương này sẽ trình bày phụ thuộc hàm như một công cụ toán học trợ giúp cho việc biểu đạt ngữ nghĩa dữ liệu và đảm bảo tính nhất quán của dữ liệu trong cơ sở dữ liệu. Các tính chất của phụ thuộc hàm và các hệ tiên đề cho phụ thuộc hàm được mô tả đầy đủ, trong đó hệ tiên đề Armstrong được sử dụng nhiều hơn cả. Một trong những khái niệm quan trọng của phụ thuộc hàm là bao đóng của tập thuộc tính và các tính chất cơ bản của phép toán lấy bao đóng cũng được trình bày. Tiếp theo giới thiệu các loại phủ phụ thuộc hàm quan trọng nhất đóng vai trò thu gọn các tập phụ thuộc hàm, tạo thuận tiện cho tối ưu hóa các thao tác ngữ nghĩa. Cuối phần này là các vấn đề về khóa của lược đồ, bài toán tìm mọi khóa cũng được đặt ra.

6.1. Các vấn đề thường gặp trong thiết kế cơ sở dữ liệu quan hệ

Để hình dung được những vấn đề thường xảy ra trong thiết kế cơ sở dữ liệu, ta xét một ví dụ như sau:

Ví dụ: Xét lược đồ quan hệ SINHVIEN(HOTEN, NGAYSINH, LOP, MONHOC, DIEM) được sử dụng để biểu diễn thông tin về các sinh viên và điểm thi các môn học của sinh viên. Ta xét quan hệ là thể hiện của lược đồ này như sau:

HOTEN	NGAYSINH	LOP	MONHOC	DIEM
Nam	23/05/1989	Tin SP20	Pascal	6
Châu	11/06/1988	Tin SP20	Pascal	9
Hoa	15/07/1989	Tin SP20	Đại số	10
Nam	23/05/1989	Tin SP20	Đại số	9
Hoa	15/07/1989	Tin SP20	Pascal	5

Trên quan hệ này, ta có thể nhận thấy một số vấn đề tồn tại như sau:

Thông tin về họ tên, ngày sinh và tên lớp của mỗi sinh viên sẽ phải lưu trữ lặp đi lặp lại nhiều lần nếu sinh viên đó có nhiều điểm ứng với nhiều môn học. (Chẳng hạn như ở quan hệ trên, thông tin về sinh viên có tên là *Nam* phải lưu trữ lặp lại hai lần). Điều này gây nên sự *dư thừa về dữ liệu* trong cơ sở dữ liệu. Sự dư thừa này lại dẫn đến các dị thường về dữ liệu như sau:

+ *Dị thường khi cập nhật dữ liệu:* Giả sử ta cần thay đổi thông tin liên quan đến sinh viên có tên là *Nam*. Điều này đòi hỏi phải cập nhật lại tất cả các bộ dữ liệu có liên quan đến sinh viên này. Tuy nhiên, do thông tin về sinh viên này phải lưu trữ lặp đi lặp lại nhiều lần nên có thể còn "sót" một số bản ghi không được cập nhật và dẫn đến sự thiếu nhất quán đối với dữ liệu.

+ *Dị thường khi bổ sung dữ liệu:* Với việc tổ chức lưu trữ thông tin như trên, ta lại không thể bổ sung thêm một bộ dữ liệu lưu trữ thông tin về một sinh viên nào đó nếu như sinh viên đó chưa có điểm thi ít nhất một môn học.

+ *Dị thường khi xóa dữ liệu:* Mỗi khi ta cần xóa toàn bộ điểm thi của một sinh viên nào đó sẽ dẫn đến xóa cả những thông tin về sinh viên đó ra khỏi cơ sở dữ liệu trong khi thực sự đây không phải là điều mà ta mong muốn.

Những vấn đề tồn tại ở trên sẽ không còn nếu ta sử dụng hai lược đồ SINHVIEN(HOTEN, NGAYSINH, TENLOP) và DIEMTHI(HOTEN,

MONHOC, DIEM) thay cho lược đồ đề cập ở trên. Khi đó, thay vì một quan hệ như ở trên, ta lại có hai quan hệ:

HOTEN	NGAYSINH	TENLOP
Nam	23/05/1989	Tin SP20
Châu	11/06/1988	Tin SP20
Hoa	15/07/1989	Tin SP20

HOTEN	MONHOC	DIEM
Nam	Pascal	6
Châu	Pascal	9
Hoa	Đại số	10
Nam	Đại số	9
Hoa	Pascal	5

Tuy nhiên, trong trường hợp này lại nảy sinh những vấn đề khác như mỗi khi muốn biết các thông tin và điểm thi các môn học của sinh viên, ta phải thực hiện phép nối tự nhiên đối với hai quan hệ trên. Tức là phải thực hiện một phép toán phức tạp và tốn kém về chi phí. Một câu hỏi khác được đặt ra là liệu khi thực hiện phép nối như vậy ta có thể có được thông tin chính xác như ban đầu hay không? Nói cách khác, liệu có tốt không khi phải tách các lược đồ quan hệ thành những lược đồ "nhỏ" hơn?

Trong một CSDL quan hệ, các quan hệ được sử dụng để mô hình hoá thế giới thực. Mỗi một quan hệ được sử dụng để tổ chức các thông tin về một tập các thực thể và/hoặc mối quan hệ giữa các tập thực thể. Tức là, mỗi một bộ trong quan hệ diễn tả thông tin về một thực thể và/hoặc mối quan hệ giữa các thực thể thông qua những giá trị cụ thể của các thuộc tính. Tuy nhiên, không phải bất kỳ một tập các bộ dữ liệu nào cũng là một thể hiện đúng của quan hệ.

Ta xét một số ví dụ sau:

Ví dụ: Xét lược đồ quan hệ HOCSINH(HOTEN, TUOI, DIACHI) và tập các bộ dữ liệu như sau:

HOTEN	TUOI	DIACHI
Nam	19	Huế
Hoa	15	Đà Nẵng
Anh	150	Hà Nội

Ta có thể nhận thấy bộ dữ liệu (Anh,150,Hà Nội) không thể là một thể hiện đúng bởi sẽ là không hợp lý nếu có một học sinh có tuổi là 150.

Ví dụ: Xét lược đồ quan hệ XEMAY(SOXE, LOAIXE, CHUXE). Khi đó tập các bộ dữ liệu dưới đây cũng không thể là một thể hiện đúng

SOXE	LOAIXE	CHUXE
75 F1 3103	Dream II	Châu
75 F3 2543	Wave	Tùng
75 F1 3103	Viva 110	Châu

Do thực sự không thể xảy ra trường hợp xe có biển số 75F1 3103 lại thuộc vào hai loại xe khác nhau.

Như vậy, ta có thể nhận thấy trên các quan hệ phải có những ràng buộc đối với giá trị của các bộ dữ liệu nhằm đảm bảo được tính hợp lệ của dữ liệu. Những ràng buộc này được phân làm hai loại:

- Những ràng buộc qui định miền giá trị cho các thành phần của các bộ dữ liệu. (Chẳng hạn bắt buộc tuổi của học sinh phải nằm trong khoảng từ 6 đến 20).

- Những ràng buộc liên quan đến tính bằng nhau giữa các giá trị thuộc tính của các bộ dữ liệu (Chẳng hạn ràng buộc qui định nếu hai bộ dữ liệu nào đó trong quan hệ XEMAY ở trên có cùng số xe thì cũng phải có cùng loại xe và chủ sở hữu). Ràng buộc dữ liệu dạng này được gọi là các *phụ thuộc hàm* và được bàn luận đến trong chương này.

6.2. Phụ thuộc hàm

Định nghĩa 6.1.

Cho tập thuộc tính U . Một *phụ thuộc hàm* (PTH) trên U là công thức dạng $f: X \rightarrow Y; X, Y \subseteq U$.

Nếu $f: X \rightarrow Y$ là một phụ thuộc hàm trên U thì ta nói tập thuộc tính Y *phụ thuộc* vào tập thuộc tính X , hoặc tập thuộc tính X *xác định hàm* tập thuộc tính Y . X là *vế trái* và Y là *vế phải* của PTH. Ta cũng dùng hai toán tử $Left(f)$ và $Right(f)$ để lấy vế trái và vế phải của PTH f .

Cho quan hệ $R(U)$ và một PTH $f: X \rightarrow Y$ trên U . Ta nói quan hệ R *thỏa PTH* f và viết $R(f)$, nếu hai bộ tùy ý trong R giống nhau trên X thì chúng cũng giống nhau trên Y ,

$$R(X \rightarrow Y) \Leftrightarrow (\forall u, v \in R): (u.X = v.X) \Rightarrow (u.Y = v.Y)$$