# NHẬP MÔN CƠ SỞ DỮ LIỆU

# *Chương 1.* CÁC KHÁI NIỆM CƠ BẢN

## 1.1. KHÁI NIỆM CƠ SỞ DỮ LIỆU TRONG TIN HỌC

### 1.1.1. Các mốc lịch sử phát triển Tin học

1936, 1944, 1950, 1954, 1958, 1966, 1968, 1971, 1980,

1990: Tin học —> CNTT

+ Thời gian đầu tiên, khi mới có ngành tin học, nó chỉ có các môn học cơ bản sau: Thuật toán, Lập chương trình máy tính, ngôn ngữ lập trình, …

Chưa có môn học riêng về Cơ sở dữ liệu (CSDL).

**+ *Thực tế đơn giản:*** Dữ liệu chưa nhiều

- Chưa có môn học riêng về CSDL, vì có thể giải quyết bài toán thực tế một cách đơn giản như sau.

Để quản lý học sinh trong một trường học, người ta chỉ cần tạo lập một bảng danh sách các học sinh (gồm các cột: Họ tên, Ngày sinh, Địa chỉ, …), sau đó ghi vào tệp (File).

- Rõ ràng làm như vậy có thể dẫn tới dư thừa dữ liệu, tốn bộ nhớ, hậu quả tiếp theo là tìm kiếm thông tin sẽ chậm hay không chính xác.

+ ***Thực tế phức tạp:*** Dữ liệu rất nhiều

- Dữ liệu ngày một nhiều, nếu ghi nhớ chúng không "ngăn lắp", không theo một "trật tự" nhất định, thì rất khó tìm kiếm thông tin, và tốn bộ nhớ.
- Môn học mới cần có, nhằm hướng dẫn cách thức ghi nhớ dữ liệu và phương pháp khai thác dữ liệu một cách hiệu quả. Đó chính là môn CSDL.

### 1.1.2. Các chuyên ngành trong CNTT

***Cách 1:*** **Hai chuyên ngành:**

Tin học lý thuyết: Thuật toán, CSDL, lập trình, …

Tin học ứng dụng:

Bài toán KH-KT, Bài toán trong công tác quản lý, Bài toán trong hoạt động kinh tế,

***Cách 2:*** **Năm chuyên ngành:**

Khoa học máy tính, Máy tính phần cứng, Mạng máy tính và truyền thông,

Công nghệ phần mềm, Hệ thống thông tin.

### 1.1.3. Khái niệm Dữ liệu, Cơ sở dữ liệu, Hệ Cơ sở dữ liệu

***1/. Khái niệm Dữ liệu***

- Dữ liệu (data) có thể hiểu đơn giản là số liệu như họ tên, địa chỉ, số điện thoại của một học sinh hay một khách hàng, ...
- Dữ liệu phức tạp hơn có thể là hình ảnh, âm thanh, dữ liệu đa phương tiện (Multimedia), …

***2/. Khái niệm Cơ sở dữ liệu***

- Cơ sở dữ liệu (Database: CSDL) có thể hiểu đơn giản là một tập hợp các dữ liệu có liên quan, được lưu trữ trong bộ nhớ theo một cấu trúc nhất định, đã được xác định trước.
- Trong một hệ thống thông tin, CSDL thực chất là một kho chứa dữ liệu.

***Ví dụ***:

Để quản lý học sinh trong một trường học, có 2 cách tạo lập danh sách các học sinh.

+ Cách 1 (Khi chưa có môn học CSDL):

Người ta chỉ cần tạo lập một bảng danh sách các học sinh (gồm các cột: Họ tên, Ngày sinh, Địa chỉ, Ngành học, Lớp học, …), sau đó ghi vào tệp (File).

+ Cách 2 (Khi đã có môn học CSDL):

Người ta không chỉ tạo ra một bảng danh sách các học sinh, mà tạo ra nhiều bảng dữ liệu liên quan, một CSDL có thể có nhiều bảng dữ liệu, ví dụ:

- Một bảng dữ liệu chính gồm các cột: Họ tên, Ngày sinh, địa chỉ, Mã ngành học, Mã lớp học, …
- Một bảng dữ liệu phụ gồm các cột: Mã ngành học, tên ngành học.
- Một bảng dữ liệu phụ gồm các cột: Mã lớp học, tên lớp học.

Với cách thức tạo lập CSDL như trên sẽ tránh dư thừa dữ liệu, tốn ít bộ nhớ, tốc độ tìm kiếm thông tin sẽ nhanh hơn, …

- Trong bảng dữ liệu chính, thay vì phải ghi tên ngành học, hơi dài: tốn bộ nhớ, Người ta chỉ ghi Mã ngành học: tốn ít bộ nhớ, mặt khác tìm kiếm sẽ nhanh hơn !

***3/. Khái niệm Hệ Cơ sở dữ liệu***

+ Hệ Cơ sở dữ liệu (CSDL) bao gồm các thành phần sau:

- Cơ sở dữ liệu các thông tin (Kho thông tin).

- Các chương trình thực hiện quản lý CSDL: Cập nhật và khai thác CSDL. (Quản lý Kho thông tin).

### 1.1.4. Khái niệm Hệ quản trị Cơ sở dữ liệu

**Hệ quản trị Cơ sở dữ liệu**

+ Hệ quản trị Cơ sở dữ liệu (DataBase Management System: DBMS) là một Hệ chương trình trợ giúp quá trình tạo lập Hệ CSDL và quản lý CSDL.

+ Hệ quản trị Cơ sở dữ liệu có ba thành phần chính:

- Bộ công cụ hỗ trợ tạo lập Cơ sở dữ liệu.

- Bộ công cụ hỗ trợ quản lý Cơ sở dữ liệu (cập nhật, khai thác CSDL).

- Ngôn ngữ lập trình để tạo lập các chương trình quản lý CSDL (cập nhật, khai thác).

Ví dụ:

+ Hệ QT CSDL Foxpro gồm có:

- Bộ công cụ hỗ trợ tạo lập Cơ sở dữ liệu.

- Bộ công cụ hỗ trợ quản lý Cơ sở dữ liệu (cập nhật, khai thác CSDL).

- Ngôn ngữ lập trình Foxpro để tạo lập các chương trình quản lý CSDL.

+ Hệ QT CSDL Oracle

+ Hệ QT CSDL SQL Server

**Ví dụ về CSDL**

KHACH_HANG

| MSKH | TÊNKH | TP |
|---|---|---|
| S1 | An | HCM |
| S2 | Hoà | HN |
| S3 | Bình | NT |
| S4 | Trang | NT |

VAN_CHUYEN

| TP | PVC |
|---|---|
| HCM | 01 |
| HN | 02 |
| NT | 03 |

MAT_HANG

| MSMH | TÊNMH | ĐG |
|---|---|---|
| P1 | Táo | 650 |
| P2 | Cam | 500 |
| P3 | Chanh | 450 |

DAT_HANG

| MSKH | MSMH | SL |
|---|---|---|
| S1 | P1 | 300 |
| S1 | P2 | 200 |
| S1 | P3 | 400 |
| S2 | P1 | 100 |
| S2 | P3 | 300 |
| S3 | P2 | 200 |
| S4 | P2 | 210 |

**Định nghĩa CSDL:** chỉ định cấu trúc mỗi "bảng", bao gồm các phần tử dữ liệu và kiểu dữ liệu tương ứng.

**Xây dựng CSDL:** Đưa dữ liệu vào các "bảng" KHACHHANG, VANCHUYEN, MATHANG, DATHANG.

**Xử lý CSDL:** Thực hiện các truy vấn và các phép cập nhật, chẳng hạn: "Khách hàng có tên là An đặt những mặt hàng nào", "Tên những khách hàng đã đặt mặt hàng Cam", "Tính thành tiền"…

### 1.1.5. Khái niệm Hệ thống thông tin

Để xây dựng được một Hệ thống thông tốt, cần phải hiểu rõ cả 5 chuyên ngành trong CNTT

## 1.2. CÁC MÔ HÌNH CƠ SỞ DỮ LIỆU

### 1.2.1. Phân loại tổng quan

***1/. Mô hình CSDL bậc thấp*** **(Mức cụ thể - Mức Vật lý)**

- Mô hình này chỉ quan tâm tới cách thức biểu diễn dữ liệu cụ thể (của CSDL) trong bộ nhớ của máy tính.

Tức là chỉ quan tâm tới việc các dữ liệu của CSDL được lưu trữ trong bộ nhớ của máy tính như thế nào ?

- Mô hình này có ý nghĩa nhiều với các chuyên gia máy tính, nhưng ít có ý nghĩa với người dùng CSDL.

***Ví dụ:***

***2/. Mô hình CSDL bậc cao*** **(Mức Quan niệm - Logic)**

- Mô hình này quan tâm đến các đối tượng được biểu diễn trong CSDL, ít quan tâm tới cách thức biểu diễn dữ liệu cụ thể trong bộ nhớ của máy tính.
- Mô hình này có ý nghĩa nhiều với với người dùng CSDL, nhưng ít có ý nghĩa.với các chuyên gia máy tính

***Ví dụ:***

- Mô hình CSDL dạng quan hệ thực thể (Entity Relationship Model)
- Mô hình CSDL hướng đối tượng (Object Oriented Model)

***3/. Mô hình CSDL thể hiện*** **(Mức Logic - Cụ thể)**

- Mô hình CSDL "thể hiện" nằm giữa hai mô hình trên.
- Mô hình này có ý nghĩa với cả chuyên gia máy tính, và với người dùng CSDL.

***Ví dụ:***

- Mô hình CSDL dạng phân cấp, Mô hình CSDL dạng đồ thị (mạng), Mô hình CSDL dạng quan hệ.

### 1.2.2. Phân loại cụ thể

# *Chương 2.* MÔ HÌNH CƠ SỞ DỮ LIỆU DẠNG QUAN HỆ

## 2.1. CÁC KHÁI NIỆM TRONG MÔ HÌNH CSDL QUAN HỆ

Mô hình CSDL quan hệ được Codd đề nghị năm 1970.

### 2.1.1. Miền, thuộc tính, quan hệ

***1/. Khái niệm Miền:***

+ Miền (domain) là một tập hợp (các giá trị hoặc các đối tượng) D.

Mỗi miền có một tên, mô tả, kiểu dữ liệu và khuôn dạng.

***2/. Quan hệ:***

+ Tích Decac:

Gọi $D_1, D_2, \ldots, D_n$ là **n** miền, Tích Decac của **n** miền trên là $D_1 \times D_2 \times \ldots \times D_n$.

+ Quan hệ là tập con của Tích Decac. Tức là Quan hệ **r** $\subseteq D_1 \times D_2 \times \ldots \times D_n$.

***3/. Bảng:***

+ Bảng là một quan hệ hữu hạn, được biểu diễn thành hàng và cột.

Giá trị trong mỗi cột thuộc về một miền $D_i$ nào đó.

Mỗi hàng là một phần tử của quan hệ **r** .

***Ví dụ***

| *Tên miền* | M_HOTEN | M_SOĐT |
|---|---|---|
| *Mô tả* | Tập các họ tên người VN | Tập các số điện thoại tại VN |
| *Kiểu dữ liệu* | Xâu các ký tự | Xâu các chữ số |
| *Khuôn dạng* | | (ddd)dddddd |

| **HOTEN** | **CMND** | **ĐT_NHA** | **Địa chỉ** | **ĐT_CQ** | **TUOI** |
|---|---|---|---|---|---|
| Lê Chí Phèo | 220877654 | (056)789543 | Hà nội | (08)9876548 | 30 |
| Trần Kim Nở | 345267656 | (088)765890 | Hải phòng | (058)876984 | 25 |
| Lý Bá Kiến | 123123456 | (058)908756 | Hà nội | (058)888888 | 50 |

***4/. Thuộc tính:***

+ Thuộc tính (Attribute) là một lớp dữ liệu mô tả hành vi, tính chất phát sinh trong CSDL, nghĩa là nó chỉ dựa vào tính chất của lớp dữ liệu này.

Mỗi thuộc tính chỉ có các giá trị trong một miền (domain) của thuộc tính.

Một mục dữ liệu (item) trong thuộc tính là một giá trị trong miền thuộc tính này.

Một thuộc tính là dạng kết nối (joined) nếu nó được định nghĩa từ một vài các thuộc tính khác; do đó domain của nó là tập con của tích Đề các các domain của các thuộc tính này.

*Ký hiệu:*

- Gọi c là giá trị của thuộc tính C.

Nếu C được tạo thành từ các thuộc tính $C_1$, $C_2$,...,$C_n$, khi đó ta ký hiệu c.$C_1$ và c($C_1$) chỉ giá trị c đối với thuộc tính $C_1$.

***5/. Lược đồ quan hệ:*** Ký hiệu **R**($A_1$, $A_2$, …, $A_n$)

Là tập thuộc tính **R** = {$A_1$, $A_2$, …, $A_n$}, mỗi thuộc tính $A_i$ có miền giá trị $D_i$.

+ Lược đồ quan hệ để mô tả một đối tượng hoặc một loại quan hệ giữa các đối tượng.

+ Bậc của lược đồ quan hệ là số lượng thuộc tính trong lược đồ quan hệ.

***Ví dụ:***

**GV**(HOTEN, CMND, ĐT_NHA, ĐC, ĐT_CQ, TUOI)

GV là tên lược đồ quan hệ, có bậc là 6.

HOTEN là một thuộc tính, có miền giá trị DOM(TEN) = M_HOTEN.

ĐT_NHA, ĐT_CQ là các thuộc tính, có miền giá trị DOM(ĐT_NHA) = DOM(ĐT_CQ) = M_SĐT (Miền Số ĐT).

***6/. Quan hệ***

+ Một quan hệ (Relation) **r** của lược đồ quan hệ **R**($A_1$, $A_2$, …, $A_n$), ký hiệu là **r**(**R**).

Quan hệ **r** là một tập hữu hạn các bộ (dòng, bản ghi, record) của **R**.

Trong một quan hệ không có hai bộ giống nhau.

+ Một bộ (n-tuple) của **R** là một phần tử của tính Đề các của các domain tương ứng với n thuộc tính của R.

+ Một thực thể (entity) r của R là một bộ của R thoả mãn vị từ ||R||(r)=true.

*Chú ý*: Thực tế một bộ của tích Đề các có thể hay không là một thực thể của quan hệ R.

Ví dụ: Quan hệ **r** của lược đồ quan hệ **GV**

| **HOTEN** | **CMND** | **ĐT_NHA** | **ĐC** | **ĐT_CQ** | **TUOI** |
|---|---|---|---|---|---|
| Lê Chí Phèo | 220877654 | (056)789543 | Hà nội | (08)9876548 | 30 |
| Trần Kim Nở | 345267656 | (088)765890 | Hải phòng | (058)876984 | 25 |
| Lý Bá Kiến | 123123456 | (058)908756 | Hà nội | (058)888888 | 50 |

***Các ký hiệu trong mô hình CSDL quan hệ***

Lược đồ quan hệ R bậc n: **$R(A_1, A_2, \ldots, A_n)$**

Tập thuộc tính của R: **$R = \{A_1, A_2, \ldots, A_n\} = R^+$**

Bộ **t** của quan hệ **r**(**R**): **$t = <v_1, v_2, \ldots, v_n>$**, trong đó **$v_i$** là giá trị của thuộc tính **$A_i$**

**t**[$A_i$] ( t.$A_i$ , **t**($A_i$) ): chỉ giá trị của thuộc tính $A_i$ trên bộ **t**.

**t**[$A_u$, $A_w$, …, $A_z$]: chỉ các giá trị của các thuộc tính $A_u$, $A_w$, …, $A_z$ trên bộ **t**.

### 2.1.2. Khóa của lược đồ quan hệ

**1/. *Siêu khoá:***

Tập thuộc tính khác rỗng SK $\subseteq$ R, được gọi là ***siêu khóa***, nếu

$$\forall \mathbf{r}, \quad \forall \mathbf{t_1}, \mathbf{t_2} \in \mathbf{r}, \quad \mathbf{t_1} \neq \mathbf{t_2} \Rightarrow \quad \mathbf{t_1}[SK] \neq \mathbf{t_2}[SK]$$

Nhận xét: Mỗi lược đồ quan hệ đều có tối thiểu một siêu khóa.

**2/. *Khóa:***

Tập thuộc tính khác rỗng SK $\subseteq$ R, được gọi là ***khoá***, nếu thỏa mãn đồng thời hai điều kiện: (Tóm lại: Khóa là siêu khóa "nhỏ nhất")

+ K là một siêu khóa của lược đồ quan hệ R.

+ $\forall$ K' $\subset$ K, K' $\neq$ K, K' không phải là siêu khoá của R.

***Chú ý:***

- Mọi quan hệ đều có một siêu khóa "tầm thường", đó là tập tất cả các thuộc tính của quan hệ này.

- Khóa là siêu khóa "nhỏ nhất"

Khóa là tập thuộc tính nhỏ nhất, nhờ nó có thể phân biệt các bản ghi với nhau.

Giá trị khóa dùng để nhận biết một bộ trong một quan hệ.

- Khoá được xác định dựa vào ý nghĩa các thuộc tính trong một Lược đồ quan hệ.

- Lược đồ quan hệ có thể có nhiều khoá (gọi là khóa dự tuyển – Candidate key).

Một trong các khóa đó được chỉ định làm khóa chính (primary key) của quan hệ.

Khóa chính thường được chọn là khóa tối thiểu.

Ví dụ: GIẢNG_KHÓA(MÔN, GVIÊN, HKỲ, LỚP, PHÒNG, CA, THỨ)

Tân từ: Mỗi giáo viên (GVIÊN), vào một học kỳ (HKỲ), dạy môn học (MÔN) cho lớp (LỚP), tại phòng (PHÒNG), vào ca giảng (CA) của một thứ trong tuần (THỨ).

$\Rightarrow$ 3 khoá: {HKỲ, PHÒNG, CA, THỨ}, {MÔN, LỚP}, {GVIÊN, HKỲ, CA, THỨ}

+ Khi cài đặt một quan hệ thành một bảng (Table), cần chọn một khoá làm cơ sở để nhận biết các bộ. Khóa được chọn này gọi là ***khóa chính*** (primary key) $\Rightarrow$ các thuộc tính khóa chính phải khác trống (khác null).

Thường chọn khóa có số thuộc tính ít hơn làm khóa chính.

Qui ước: các thuộc tính khóa chính được gạch dưới.

VD: GIẢNG_KHÓA(<u>MÔN</u>, GVIÊN, HKỲ, <u>LỚP</u>, PHÒNG, CA, THỨ)

### 2.1.3. Lược đồ CSDL quan hệ và các ràng buộc toàn ven (RBTV)

***Lược đồ CSDL quan hệ*** = {lược đồ quan hệ} + {Ràng buộc toàn vẹn}

***Thể hiện CSDL quan hệ*** = {Thể hiện quan hệ}

trong đó $r_i$ là thể hiện của $R_i$ thoả mãn các ràng buộc trong tập các ràng buộc toàn vẹn.

***Ràng buộc toàn vẹn (RBTV) trên 1 CSDL quan hệ***

Ràng buộc toàn vẹn (RBTV, integrity constraint): là những qui tắc, điều kiện, ràng buộc cần được thoả mãn cho mọi thể hiện CSDL quan hệ.

***Ràng buộc về khóa*** (key constraint): 2 bộ khác nhau trong cùng một quan hệ *phải có giá trị tại khoá khác nhau.*

***Ràng buộc tham chiếu*** (referential constraint): Một bộ trong một quan hệ, nếu tham chiếu đến một bộ khác trong một quan hệ khác thì *bộ được tham chiếu phải tồn tại trước.*

Ràng buộc tham chiếu còn gọi là ràng buộc khóa ngoại.

Ngoài ra, còn có một số RBTV về ngữ nghĩa khác.

***Khoá ngoại*** (foreign key)

Xét 2 lược đồ quan hệ $R_1$ và $R_2$, FK là 1 tập thuộc tính khác rỗng của $R_1$. FK được gọi là *khóa ngoại* của $R_1$ (tham chiếu tới $R_2$) nếu thỏa mãn 2 điều kiện sau:

Các thuộc tính trong FK phải có cùng miền trị với các thuộc tính khoá chính PK của $R_2$.

Giá trị tại FK của một bộ $t_1 \in R_1$,hoặc bằng giá trị tại PK của một $t_2 \in R_2$, hoặc bằng giá trị trống (null). Trường hợp đầu, ta nói $t_1$ tham chiếu tới bộ $t_2$.

VD: MAMH là khoá ngoại của ĐATHANG tham chiếu đến MATHANG

Chú ý:

Trong 1 lược đồ quan hệ, một thuộc tính có thể vừa tham gia vào khoá chính, vừa tham gia vào khóa ngoại.

Khoá ngoại có thể tham chiếu đến khóa chính của cùng một lược đồ quan hệ.

VD: NHANVIEN(MaNV, HoTen, MaNguoiPhuTrach)

Có thể có nhiều khóa ngoại tham chiếu đến cùng một khóa chính.

Nên khai báo khoá ngoại (ràng buộc tham chiếu) nếu hệ QTCSDL cho phép.

***Ví dụ :*** **CSDL "CÔNG TY"**

NHANVIEN

| **Mã-NV** | Họ tên | Ngày sinh | Địa chỉ | Mã-DV | Lương |
|---|---|---|---|---|---|

ĐƠN_VỊ

| **Mã-DV** | Tên Đơn vị | Trưởng Đơn vị | Địa điểm ĐV |
|---|---|---|---|

DỰ_ÁN

| **Mã-DA** | Tên Dự án | Địa điểm DA |
|---|---|---|

PHÂN_CÔNGVIỆC

| **Mã-NV** | **Mã-DA** | Thời gian làm việc |
|---|---|---|

## 2.2. CÁC PHÉP TÍNH TRONG MÔ HÌNH CSDL QUAN HỆ

### 2.2.1. Các phép toán cập nhật trên một quan hệ

+ Các phép tính cập nhật trên một quan hệ: Xem, Xen, Xoá, Sửa.

Khi sử dụng các phép toán này, cần đảm bảo các ràng buộc toàn vẹn không bị vi phạm.

+ Các phép tính quan hệ (chiếu, chọn).

#### *2.2.1.1. Phép tính cập nhật*

Xem, Xen, Xoá, Sửa.

#### *2.2.1.2. Phép chiếu, phép chọn*

**1/. Phép chiếu**

+ Cho lược đồ quan hệ **R** = {$A_1$, $A_2$, …, $A_n$}, quan hệ **r**, **X** là tập con của R (**X** ⊆ **R**), ta gọi **X** là lược đồ con của **R**.

+ Ta xét quan hệ con của quan hệ **r** chỉ trên tập thuộc tính của **X**, đó là hình chiếu của **r** trên **X**.

Quan hệ **r** chiếu lên **X** là một quan hệ trên lược đồ quan hệ **X** ký hiệu là **r.X**.

Tương tự các phần tử **r.X** được ký hiệu là **t.X** là hình chiếu của **t** lên **X**.

**r.X** = {**t.X**, **t** ∈ **r**}.

Phép chiếu được ký hiệu:

$$\pi_{<ds\_thuộc\_tính>}(<Tên\_quan\_hệ>)$$

Trong đó:

**π**: ký hiệu phép chiếu.

<ds_thuộc_tính>: danh sách các thuộc tính của quan hệ <tên_quan_hệ>

<Tên_quan_hệ>: chỉ quan hệ được chọn.

+ Kết quả thu được từ phép chiếu là một quan hệ, có danh sách thuộc tính như trong <ds_thuộc_tính>, với cùng thứ tự.

***Chú ý***:

+ Nếu <ds_thuộc_tính> chỉ có các thuộc tính không khóa, thì có thể có những bộ trùng lặp sau khi chiếu, phép chiếu ngầm bỏ đi các bộ lặp, do đó kết quả là một quan hệ hợp lệ.

+ Nếu <ds1> ⊆ <ds2> thì $\pi_{<ds1>}(\pi_{<ds2>}(R)) = \pi_{<ds1>}(R)$.

+ Phép chiếu không có tính giao hoán.

***Ví dụ***:

Cho lược đồ quan hệ **R** ={A, B, C}, lược đồ quan hệ con của **R** là X={A, B}

Phép chiếu $\pi_{<ds\_thuộc\_tính>}$**(<Tên_quan_hệ>)** = $\pi_{A,B}$ **(r)** hay **r.X (A B )**:

| **r** (A | B | C ) |
|---|---|---|
| a1 | b1 | c1 |
| a2 | b2 | c1 |
| a2 | b2 | c2 |

| **r.**X (A | B | ) hay $\pi_{A,B}$ (**r**) |
|---|---|---|
| a1 | b1 | |
| a2 | b2 | |

## 2/. Phép chọu

Phép chọn dùng để trích chọn ***1 tập con*** của quan hệ.

Các bộ được trích chọn phải thỏa mãn *điều kiện chọn*.

Phép chọn được ký hiệu:

$$\sigma_{<dk\_chọn>}(<\text{Tên\_quan\_hệ}>)$$

Trong đó:

$\sigma$: ký hiệu phép chọn.

<Tên_quan_hệ>: chỉ quan hệ được chọn.

+ Kết quả thu được từ phép chọn là một quan hệ, có cùng danh sách thuộc tính được chỉ ra trong <Tên_quan_hệ>, nhưng chỉ gồm những bộ thỏa mãn điều kiện chọn.

+ Điều kiện chọn được hình thành từ các mệnh đề có dạng:

<tên_thuộc_tính> <phép_so_sánh> <giá_trị_hằng>

<tên_thuộc_tính> <phép_so_sánh> <tên_thuộc_tính>

<tên_thuộc_tính> là tên thuộc tính của <Tên_quan_hệ>, phép so sánh thường là:

=, ≠, >, ≥, <, ≤.

Các mệnh đề có thể được nối lại nhờ vào các phép ¬, ∧, ∨

***Ví dụ***:

Cho lược đồ quan hệ Sinh viên **R**, tìm sinh viên có ít nhất một điểm < 5, tức là xác định $\sigma_{<diem1<5 \wedge diem2<5>}$**(R)**

| Ten | NS | diem1 | diem2 |
|---|---|---|---|
| Nam | 1978 | 5 | 6 |
| Lan | 1979 | 4 | 6 |
| Hoa | 1978 | 4 | 3 |

| Ten | NS | diem1 | diem2 |
|---|---|---|---|
| Hoa | 1978 | 4 | 3 |

### 2.2.2. Các phép toán cập nhật trên nhiều quan hệ

+ Các phép tính trên nhiều quan hệ (như trên tập hợp): hội, giao, trừ, tích Decac, …

+ Các phép tính trên nhiều quan hệ: kết nối quan hệ, phân tách quan hệ, …)

#### *2.2.2.1. Các phép tính như trên tập hợp*

***+ Khả hợp:*** (Union compatibility)

Hai lược đồ quan hệ $R(A_1, A_2, \ldots, A_n)$ và $S(B_1, B_2, \ldots, B_n)$ đgl ***khả hợp*** nếu cùng bậc n (cùng số thuộc tính) và có $DOM(A_i) = DOM(B_i)$, với $1 \leq i \leq n$.

+ Để thực hiện các phép toán trên nhiều quan hệ, điều kiện các quan hệ phải ***khả hợp***.

VD:

SINHVIEN_LOP1

| **Tên SV** | **Địa chỉ** |
|---|---|
| Trần Kim Nở | Hà nội |
| Lê Chí Phèo | Hải phòng |
| Lý Bá Kiến | Hà nội |

SINHVIEN_LOP2

| **Tên SV** | **Địa chỉ** |
|---|---|
| Trương Văn Cam | Sài gòn |
| Lã Kim Oanh | Hải phòng |
| Vũ Xuân Trường | Thái bình |
| Lê Chí Phèo | Hải phòng |

***1/. Phép hội*** của R và S, ký hiệu là **R ∪ S**, là một quan hệ gồm các bộ thuộc R hoặc thuộc S, hoặc thuộc cả hai quan hệ, các bộ trùng lắp thì loại bỏ.

***2/. Phép giao*** của R và S, ký hiệu là **R ∩ S,** là một quan hệ gồm các bộ thuộc đồng thời R và S.

***3/. Phép trừ*** của R và S, ký hiệu **R – S,** là một quan hệ gồm các bộ thuộc R và không thuộc S.

***Ví dụ***:

SINHVIEN_LOP1 ∪ SINHVIEN_LOP2

| **Tên SV** | **Địa chỉ** |
|---|---|
| Trần Kim Nở | Hà nội |
| Lê Chí Phèo | Hải phòng |
| Lý Bá Kiến | Hà nội |
| Trương Văn Cam | Sài gòn |
| Lã Kim Oanh | Hải phòng |
| Vũ Xuân Trường | Thái bình |

SINHVIEN_LOP1 ∩ SINHVIEN_LOP2

| **Tên SV** | **Địa chỉ** |
|---|---|
| Lê Chí Phèo | Hải phòng |

SINHVIEN_LOP1 - SINHVIEN_LOP2

| **Tên SV** | **Địa chỉ** |
|---|---|
| Trần Kim Nở | Hà nội |
| Lý Bá Kiến | Hà nội |

***Các tính chất:***

Giao hoán: $R \cup S = S \cup R$, $R \cap S = S \cap R$

Kết hợp: $R \cap (S \cap T) = (R \cap S) \cap T$,

$R \cup (S \cup T) = (R \cup S) \cup T$

### *2.2.2.2. Các phép tính kết nối, phân tách quan hệ*

**1/. Tích Decac (Descartes)**

Cho **R**($A_1$, $A_2$, …, $A_n$) và **S**($B_1$, $B_2$, …, $B_m$), tích ***Decac*** giữa hai quan hệ R và S, ký hiệu là **R x S,** là quan hệ có n + m thuộc tính.

**Q**($A_1$, $A_2$, …, $A_n$, $B_1$, $B_2$, …, $B_m$)

Trong đó mỗi bộ của **Q** là tổ hợp giữa 1 bộ trong R và 1 bộ trong S, nếu R có u bộ và S có v bộ thì Q có u . v bộ.

***Ví dụ***:

| **R** | A | B |
|---|---|---|
| | a1 | b1 |
| | a2 | b2 |

| **S** | C | D |
|---|---|---|
| | c1 | d1 |
| | c2 | d2 |
| | c3 | d3 |

| **RxS** | A | B | C | D |
|---|---|---|---|---|
| | a1 | b1 | c1 | d1 |
| | a1 | b1 | c2 | d2 |
| | a1 | b1 | c3 | d3 |
| | a2 | b2 | c1 | d1 |
| | a2 | b2 | c2 | d2 |
| | a2 | b2 | c3 | d3 |

**2/. Phép chia**

Cho lược đồ quan hệ **R** = {$A_1$, .., $A_n$} và **S** là lược đồ con của R (**S** ⊂ R), giả sử **r** và **s** là hai quan hệ trên **R** và **S** tương ứng.

Phép chia của quan hệ r cho s, ký hiệu: **r ÷ s**, kết quả là quan hệ trên lược đồ **R – S** gồm các bộ t, sao cho tồn tại bộ u ∈ s thì bộ <**t, u**> ∈ **r**.

**r ÷ s** = {**t** : ∃ u ∈ **s** và <**t, u**> ∈ **r**}

Chú ý: S phải thực sự là tập con của R, (**S** ⊂ R).

***Ví dụ***:

**r** (A B C D)

a b c d

a b e f

b c e f

e d c d

e d e f

a b d e

**s** (C D )

c d

e f

**r ÷ s** (A B)

a b

e d

b c

**3/. Phép nối tự nhiên (Join)**

Cho hai lược đồ: $R_1$ và $R_2$, $r_1$, $r_2$ là hai quan hệ tương ứng trên $R_1$, $R_2$.

Phép kết nối (tự nhiên) của $r_1$ và $r_2$ ký hiệu: $\mathbf{r_1} |><| \mathbf{r_2}$ là quan hệ trên lược đồ $R_1 \cup R_2$ gồm các phần tử **t** mà chiếu lên $R_1$ là phần tử thuộc $r_1$, còn chiếu lên $R_2$ là phần tử thuộc $r_2$.

$$\mathbf{r_1} |><| \mathbf{r_2} = \{t : \quad t.R_1 \in r_1 \text{ và } t.R_2 \in r_2 \}$$

Trong trường hợp hai tập thuộc tính như nhau thì $r_1 |><| r_2 = r_1 * r_2$.

Trong trường hợp hai tập là tách biệt nhau thì $r_1 |><| r_2 = r_1 \times r_2$.

***Ví dụ***:

| $r_1$ (A | B | C) |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_2$ | $c_2$ |
| $a_2$ | $b_1$ | $c_1$ |

| $r_2$ (C | D ) |
|---|---|
| $c_1$ | $d_1$ |
| $c_2$ | $d_2$ |

| $r_1 |><| r_2$ (A | B | C | D ) |
|---|---|---|---|
| $a_1$ | $b_1$ | $c_1$ | $d_1$ |
| $a_1$ | $b_2$ | $c_2$ | $d_2$ |
| $a_2$ | $b_1$ | $c_1$ | $d_1$ |

**4/. Phép kết nối theo phép tính θ**

Cho r và s là hai quan hệ tương ứng trên hai lược đồ quan hệ R và S rời nhau ($R \cap S = \varnothing$).

Phép kết nối theo phép tính θ của quan hệ r và s, ký hiệu $\mathbf{r}|><|_{i\,\theta\,j}\ \mathbf{s}$ là một quan hệ trên lược đồ $R \cup S$ gồm những bộ thuộc tích Đề các của r và s sao cho thành phần thứ i của quan hệ r thỏa mãn phép toán θ với thành phần thứ j của quan hệ s.

***Ví dụ***: Trong trường hợp này θ là quan hệ <, i=2 và j=1.

| r | A | B | C |
|---|---|---|---|
| | 1 | 2 | 3 |
| | 4 | 5 | 6 |
| | 7 | 8 | 9 |

| s | D | E |
|---|---|---|
| | 3 | 1 |
| | 6 | 2 |

r |><| **C ≤ D** s

| A | B | C | D | E |
|---|---|---|---|---|
| 1 | 2 | 3 | 3 | 1 |
| 1 | 2 | 3 | 6 | 2 |
| 4 | 5 | 6 | 6 | 2 |

**5/. Các phép toán quan hệ bổ sung**

Hầu hết các hệ QT CSDL đều bổ sung thêm một số phép toán sau:

AVERAGE : tính giá trị trung bình

MAX : tính giá trị lớn nhất

MIN : tính giá trị bé nhất

SUM : tính tổng cộng

COUNT : đếm.

Cú pháp: **<các thuộc tính phân nhóm>*F*<d/s hàm> (<quan hệ>)**

***Ví dụ***:

Với mỗi phòng ban, tìm số lượng nhân viên và mục lương trung bình.

$R(SOPHG, SONV, LUONGTB) \leftarrow {}_{PHG}\boldsymbol{F}_{COUNT\ MANV,\ AVERAGE\ LUONG}(NHANVIEN)$

# CHƯƠNG 3. LÝ THUYẾT PHỤ THUỘC HÀM

## 3.1. Các nguyên tắc thiết kế lược đồ quan hệ

Khi chúng ta nhóm các thuộc tính để tạo nên một lược đồ quan hệ, ta giả thiết rằng có một ý nghĩa nào đó gắn với các thuộc tính. Ý nghĩa này, còn gọi là *ngữ nghĩa*, chỉ ra việc hiểu các giá trị thuộc tính lưu giữ trong các bộ của một quan hệ như thế nào. Nói cách khác, các giá trị thuộc tính trong một bộ liên hệ với nhau như thế nào. Nếu việc thiết kế khái niệm được làm một cách cẩn thận, sau đó là một chuyển đổi sang các quan hệ thì hầu hết ngữ nghĩa đã được giải thích và thiết kế kết quả có một ý nghĩa rõ ràng. Nói chung, việc giải thích ngữ nghĩa của quan hệ càng dễ dàng thì việc thiết kế lược đồ quan hệ càng tốt. Một ví dụ về thiết kế lược đồ quan hệ tốt là lược đồ cơ sở dữ liệu "CÔNG TY". Trong lược đồ đó, các thuộc tính đều có ý nghĩa rõ ràng, không có tính mập mờ. Nguyên tắc sau sẽ hỗ trợ cho việc thiết kế lược đồ quan hệ.

Nguyên tắc 1: Thiết kế một lược đồ quan hệ sao cho dễ giải thích ý nghĩa của nó. Đừng tổ hợp các thuộc tính từ nhiều kiểu thực thể và kiểu liên kết vào một quan hệ đơn. Một cách trực quan, nếu một lược đồ quan hệ tương ứng với một kiểu thực thể hoặc một kiểu liên kết thì ý nghĩa trở nên rõ ràng. Ngược lại, một quan hệ tương ứng với một hỗn hợp các thực thể và liên kết thì ý nghĩa trở nên không rõ ràng.

### *3.1.2 Thông tin dư thừa trong các bộ và sự dị thường cập nhật*

Một mục tiêu của thiết kế lược đồ là làm tối thiểu không gian lưu trữ các quan hệ cơ sở. Các thuộc tính được nhóm vào trong các lược đồ quan hệ có một ảnh hưởng đáng kể đến không gian lưu trữ. Nếu cùng một thông tin được lưu giữ nhiều lần trong cơ sở dữ liệu thì ta gọi đó là dư thừa thông tin và điều đó sẽ làm lãng phí không gian nhớ. Ví dụ, giả sử ta có bảng cơ sở sau đây:

HÀNGHÓA_KHO

| **MãsốHH** | **TênHH** | **Mô Tả Hàng** | **Ngàysản xuất** | **MãsốKho** | **TênKho** | **Ghichu** |
|---|---|---|---|---|---|---|
| Mh01 | Ốc vít | Loại 3 phân | 12/02/79 | 5 | Kho số5 | Trữ sản phẩm |

| | | | | | | |
|---|---|---|---|---|---|---|
| Mh02 | Bulong | Loại lớn | 14/02/66 | 5 | Kho số 5 | Trữ sản phẩm |
| Mh03 | Kìm | Khâu bao | 05/08/79 | 4 | Vật liệu | Trữ vật liệu |
| Mh04 | Dao | Loại lớn | 26/06/52 | 4 | Vật liệu | Trữ vật liệu |
| Mh05 | Kéo | Cắt bao | 14/08/73 | 5 | Kho số 5 | Trữ sản phẩm |
| Mh06 | Đinh | 8 phân | 26/03/83 | 5 | Kho số 5 | Trữ sản phẩm |
| Mh07 | Dây gai | Xây dựng | 15/03/80 | 4 | Vật liệu | Trữ vật liệu |
| Mh08 | Găng tay | Công nghiệp | 02/05/47 | 1 | Thiết bị | Các thiết bị điện |

Ở đây có sự dư thừa thông tin. Nếu một kho lưu trữ nhiều sản phẩm thì thông tin về KHO ( Mã số kho, Tên kho, Ghi chú ) được lưu giữ nhiều lần trong bảng. So với việc dùng hai bảng HÀNG HÓA và KHO riêng rẽ không làm lãng phí không gian nhớ.

Ngoài việc lãng phí không gian nhớ nó còn dẫn đến một vấn đề nghiêm trọng là sự dị thường cập nhật. Dị thường cập nhật bao gồm : Dị thường Chèn, dị thường Xoá, dị thường Sửa đổi. Những dị thường cập nhật này sẽ đưa vào cơ sở dữ liệu những thông tin "lạ" và làm cho cơ sở dữ liệu mất tính đúng đắn.

*Dị thường Chèn*: Gây ra khó khăn khi chèn các bộ giá trị vào bảng hoặc dẫn đến vi phạm ràng buộc.

Ví dụ: Để chèn một bộ giá trị cho một *mặt hàng* mới vào bảng HÀNGHÓA_KHO ngoài các thông tin về *hàng hóa*, ta phải đưa vào các thông tin về *kho* mà sản phẩm đó được lưu trữ hoặc các giá trị null (nếu hàng hóa đó không lưu trữ trong kho nào cả). Các thông tin về *kho* phải được đưa vào một cách đúng đắn, phù hợp với các thông tin của *kho* đó trong các bộ khác.Trong lúc đó, với việc sử dụng 2 quan hệ HÀNG HÓA và KHO chúng ta không phải lo lắng gì, vì các thông tin về một *kho* chỉ được lưu trữ một lần.

Rất khó chèn một *kho* mới vào quan hệ HÀNG HÓA_KHO nếu kho đó không có sản phẩm nào lưu trữ. Cách giải quyết duy nhất là điền các giá trị null vào các thuộc tính của hàng hóa. Điều đó làm nảy sinh vấn đề về ràng buộc bởi vì MãsốHH là khóa chính của quan hệ.

*Dị thường Xóa*: Gây ra việc mất thông tin khi xóa.

Ví dụ, khi ta xóa một bộ giá trị trong bảng HÀNG HÓA - KHO. Nếu hàng hóa tương ứng với bộ giá trị đó là sản phẩm cuối cùng lưu trong kho thì phép xóa sẽ kéo theo việc làm mất thông tin về kho.

*Dị thường Sửa đổi*: Gây ra việc sửa đổi hàng loạt khi ta muốn sửa đổi một giá trị trong một bộ nào đó.

Dựa trên các dị thường ở trên, chúng ta có thể phát biểu nguyên tắc sau:

Nguyên tắc 2: Thiết kế các lược đồ quan hệ cơ sở sao cho không sinh ra những dị thường cập nhật trong các quan hệ. Nếu có xuất hiện những dị thường cập nhật thì phải ghi chép lại một cách rõ ràng và phải đảm bảo rằng các chương trình cập nhật dữ liệu sẽ thực hiện một cách đúng đắn.

### *3.1.3. Các giá trị không xác định trong các bộ*

Trong một số thiết kế lược đồ, chúng ta có thể nhóm nhiều thuộc tính với nhau vào một quan hệ "béo". Nếu nhiều thuộc tính không thích hợp cho mọi bộ trong một quan hệ, chúng ta sẽ kết thúc với nhiều giá trị null trong các bộ đó. Điều đó có thể làm tăng không gian ở mức lưu trữ và có thể dẫn đến vấn đề về hiểu ý nghĩa của các thuộc tính. Việc chỉ ra các phép nối ở mức lô gic cũng sẽ gặp khó khăn. Một vấn đề nữa với các giá trị null là các hàm nhóm như COUNT, SUM không áp dụng được đối với chúng. Hơn nữa, các giá trị null có thể nhiều cách giải thích, chẳng hạn như thuộc tính không áp dụng được cho bộ này, giá trị của thuộc tính cho bộ này là không có hoặc giá trị cho thuộc tính là có nhưng vắng mặt. Tóm lại, các giá trị null có nhiều ý nghĩa khác nhau.

Nguyên tắc 3: Tránh càng xa càng tốt việc đặt vào trong các quan hệ cơ sở những thuộc tính mà các giá trị của chúng thường xuyên là null. Nếu không thể tránh được các giá trị null thì phải đảm bảo rằng chúng chỉ áp dụng trong các trường hợp đặc biệt và không áp dụng cho một số lớn các bộ trong quan hệ.

### *3.1.4 Sinh ra các bộ giả*

Nhiều khi chúng ta đưa vào cơ sở dữ liệu những quan hệ không đúng, việc áp dụng các phép toán (nhất là các phép nối) sẽ sinh ra các bộ giá trị không đúng, gọi là các bộ "giả".

Ví dụ, xét hai lược đồ quan hệ:

HH_KHO (Tên, Kho)

HH_SP(Mã sốSP, Mã sốDA, Sốlượng, TênDA, Kho)

| **HH_KHO** | **Tên** | **Kho** |
|---|---|---|
| | Đinh | Kho số 2 |
| | Ốc vít | Kho số 1 |
| | Kéo | Kho số 4 |
| | Dao | Kho số 2 |

| **HH_SP** | **MãsốSP** | **MãsốDA** | **Sốlượng** | **TênDA** | **Kho** |
|---|---|---|---|---|---|
| | SP001 | 1 | 32 | DA01 | Kho số 2 |
| | SP001 | 2 | 7 | DA02 | Kho số 1 |
| | SP016 | 3 | 40 | DA03 | Kho số 4 |
| | SP018 | 1 | 20 | DA01 | Kho số 2 |

Bây giờ ta nối tự nhiên hai quan hệ trên với nhau, ta có quan hệ

| | **MãsốSP** | **MãsốDA** | **Số lượng** | **TênDA** | **Địađiêm** | **Tên** |
|---|---|---|---|---|---|---|
| | SP001 | 1 | 32 | DA01 | Kho số 2 | Đinh |
| * | SP001 | 1 | 32 | DA01 | Kho số 2 | Dao |
| | SP001 | 2 | 7 | DA02 | Kho số 1 | Ốc vít |
| | SP016 | 3 | 40 | DA03 | Kho số 4 | Kéo |
| * | SP018 | 1 | 20 | DA01 | Kho số 2 | Dao |
| | SP018 | 1 | 20 | DA01 | Kho số 2 | Đinh |

Ta thấy các dòng đánh dấu * là các bộ " giả". Đấy là các bộ giá trị không có trên thực tế.

Nguyên tắc 4: Thiết kế các lược đồ quan hệ sao cho chúng có thể được nối với điều kiện bằng trên các thuộc tính là khoá chính hoặc khoá ngoài theo cách đảm bảo không sinh ra các bộ "giả". Đừng có các quan hệ chứa các thuộc tính nối khác với các tổ hợp khoá chính-khoá ngoài. Nếu không tránh được những quan hệ như vậy thì đừng nối chúng trên các thuộc tính đó, bởi vì các phép nối có thể sinh ra các bộ "giả".

**3.2. Các phụ thuộc hàm**

Khái niệm cơ bản nhất trong thiết kế lược đồ quan hệ là khái niệm phụ thuộc hàm. Trong phần này chúng ta sẽ định nghĩa hình thức khái niệm này và cách sử dụng nó để định nghĩa các dạng chuẩn cho các lược đồ quan hệ

***3.2.1. Định nghĩa phụ thuộc hàm (functional dependency - FD)***

Một phụ thuộc hàm là một ràng buộc giữa hai nhóm thuộc tính của một cơ sở dữ liệu. Giả sử rằng lược đồ cơ sở dữ liệu của ta có n thuộc tính $A_1$, $A_2$, …, $A_n$ và hãy nghĩ rằng toàn bộ cơ sở dữ liệu được mô tả bằng một lược đồ quan hệ chung R(U), U ={$A_1$, $A_2$, …., $A_n$}. Giả sử X và Y là hai tập con của R.

*Một phụ thuộc hàm*, ký hiệu là **X**→ **Y**, giữa hai tập thuộc tính X và Y chỉ ra một ràng buộc trên các bộ có thể có tạo nên một trạng thái quan hệ r của R.

Ràng buộc đó là: với hai bộ $\mathbf{t_1}$ và $\mathbf{t_2}$ bất kỳ trong r, nếu có $\mathbf{t_1}$[**X**] = $\mathbf{t_2}$[**X**] thì cũng phải có $\mathbf{t_1}$[**Y**] = $\mathbf{t_2}$[**Y**].

Nếu có X → Y, chúng ta cũng nói rằng X xác định hàm Y hoặc Y phụ thuộc hàm vào X. Tập thuộc tính X được gọi là vế trái của FD, tập thuộc tính Y được gọi là vế phải của FD. Như vậy, X xác định hàm Y trong lược đồ quan hệ R khi và chỉ khi nếu hai bộ của r(R) bằng nhau trên các giá trị của X thì chúng nhất thiết phải bằng nhau trên các giá trị của Y.

Chú ý rằng nếu X → Y thì không thể nói gì về Y → X

Một phụ thuộc hàm là một tính chất ngữ nghĩa của các thuộc tính. Những người thiết kế cơ sở dữ liệu sẽ dùng hiểu biết của họ về ý nghĩa của các thuộc tính của R để chỉ ra các phụ thuộc hàm có thể có trên mọi trạng thái quan hệ của r(R) của R. Khi ngữ nghĩa của hai tập thuộc tính trong R chỉ ra rằng có thể có một phụ thuộc hàm, chúng ta sẽ đặc tả phụ thuộc hàm như một ràng buộc. Các trạng thái quan hệ r(R) thoả mãn các

ràng buộc phụ thuộc hàm được gọi là các trạng thái hợp pháp của R, bởi vì chúng tuân theo các ràng buộc phụ thuộc hàm. Như vậy, việc sử dụng chủ yếu của các phụ thuộc hàm là dùng để mô tả một lược đồ quan hệ R bằng việc chỉ ra các ràng buộc trên các thuộc tính phải thoả mãn ở mọi thời điểm. Một phụ thuộc hàm là một tính chất của lược đồ quan hệ R chứ không phải là tính chất của một trạng thái hợp pháp r của R. Vì vậy, một phụ thuộc hàm không thể được phát hiện một cách tự động từ một trạng thái r mà phải do một người hiểu biết ngữ nghĩa của các thuộc tính xác định một cách rõ ràng. Ví dụ, ta có quan hệ sau

| **DẠY** | **Giáoviên** | **Mônhọc** | **Tàiliệu** |
|---|---|---|---|
| | Hồng Tuyến | Pttk hệ thống | Lý thuyết CSDL q hệ |
| | Hồng Tuyến | Otomat&NNHT | Toán rời rạc |
| | Đặng Hải | Lý thuyết đồ thị | Toán rời rạc |
| | Lê Duy | Toán A3 | Toán cao cấp |

Mới nhìn qua, chúng ta có thể nói có một phụ thuộc hàm Tàiliệu→Mônhọc, tuy nhiên chúng ta không thể khẳng định được vì điều đó chỉ đúng với trạng thái quan hệ này, biết đâu trong trạng thái quan hệ khác có thể có hai môn học khác nhau sử dụng cùng một tài liệu tham khảo, ví dụ trên ta thấy hai môn *Otomat &NNHT* và *lý thuyết đồ thị* sử dụng cùng một tài liệu tham khảo đó là *Toán rời rạc*. Với một trạng thái cụ thể, chúng ta chỉ có thể khẳng định là không có một phụ thuộc hàm giữa nhóm thuộc tính này và nhóm thuộc tính khác. Để làm điều đó chúng ta chỉ cần đưa ra một phản ví dụ. Chẳng hạn, ở trong quan hệ trên chúng ta có thể khẳng định rằng không có phụ thuộc hàm giữa Giáoviên và Mônhọc bằng cách chỉ ra ví dụ là *Hồng Tuyến* dạy hai môn học " Pttk hệ thống" và "Otomat&NNHT" vậy Giáo viên không thể xác định duy nhất Môn học.

Để biểu diễn các phụ thuộc hàm trong một lược đồ quan hệ, chúng ta sử dụng khái niệm sơ đồ phụ thuộc hàm. Mỗi FD được biểu diễn bằng một đường nằm ngang. Các thuộc tính ở vế trái của FD được nối với đường biểu diễn FD bằng các đường thẳng đứng, các thuộc tính ở vế phải của FD được nối với đường biểu diễn FD bằng mũi tên chỉ đến các thuộc tính

**Ví dụ 1**: Ta có lược đồ quan hệ sau:

MUAHANG(Mãhàng, Mãkhách, Tênhàng, Tênkhách, Sốlượng)

Với các phụ thuộc hàm:

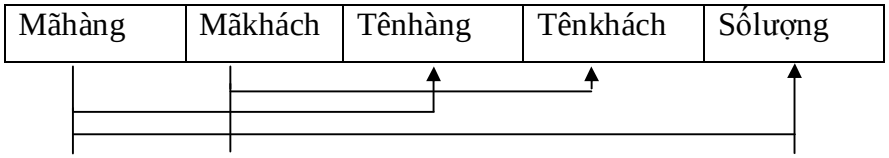
Mãhàng → Tênhàng

Mãkhách → Tênkhách

Mãhàng, Mãkhách → Sốlượng

có sơ đồ phụ thuộc hàm như sau:

MUAHANG




**Ví dụ 2**: quan hệ ĐIEM(MaSV, TenSV, Ngaysinh, MaMH, TenMH, DVHT, Diem) Có phụ thuộc hàm:

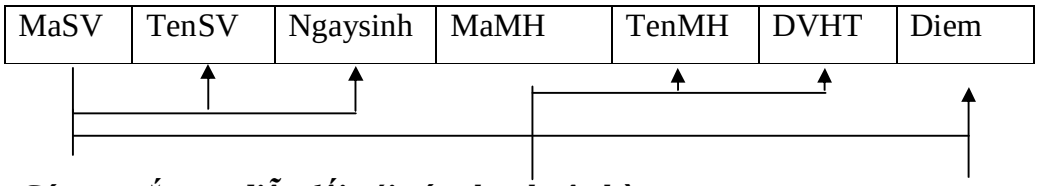
MaSV→TenSV, Ngaysinh

MaMH →TenMH, DVHT

MaSV, MaMH → Diem.

có sơ đồ phụ thuộc hàm như sau:

DIEM




### *3.2.2. Các quy tắc suy diễn đối với các phụ thuộc hàm*

Chúng ta ký hiệu F là tập các phụ thuộc hàm được xác định trên một lược đồ quan hệ R(U). X và Y là hai tập con của U. Một phụ thuộc hàm X → Y, được gọi là suy diễn được từ một tập các phụ thuộc hàm F được xác định trên R nếu X→ Y đúng trong mỗi trạng thái quan hệ r là mở rộng hợp pháp của R; nghĩa là mỗi khi r làm thoả mãn mọi phụ thuộc hàm trong F, X→ Y cũng đúng trong r. Ta sử dụng ký hiệu F |= X→ Y để ký hiệu phụ thuộc hàm X→ Y được suy diễn từ tập các phụ thuộc hàm F. Để xác định một cách suy diễn các phụ thuộc hàm có hệ thống, chúng ta phải phát hiện một tập hợp các quy tắc suy diễn. Tập quy tắc này sẽ được sử dụng để suy diễn các phụ thuộc hàm mới từ một tập các phụ thuộc hàm cho trước. Có 6 quy tắc suy diễn đối với các phụ thuộc hàm:

QT1 (quy tắc phản xạ) : Nếu $X \supset Y$ thì $X \rightarrow Y$

QT2 (quy tắc tăng) : $\{ X \rightarrow Y \} \models XZ \rightarrow YZ$

QT3 (quy tắc bắc cầu) : $\{ X \rightarrow Y, Y \rightarrow Z \} \models X \rightarrow Y$

QT4 (quy tắc chiếu) : $\{ X \rightarrow YZ \} \models X \rightarrow Y$ và $X \rightarrow Z$

QT5 (quy tắc hợp) : $\{ X \rightarrow Y, X \rightarrow Z \} \models X \rightarrow YZ$

QT6 (quy tắc tựa bắc cầu): $\{ X \rightarrow Y, WY \rightarrow Z \} \models WX \rightarrow Z$

Quy tắc phản xạ phát biểu rằng một tập hợp các thuộc tính luôn luôn xác định chính nó hoặc một tập con bất kỳ của nó. Vì QT1 tạo ra các phụ thuộc luôn luôn đúng, những phụ thuộc như vậy được gọi là *tầm thường*. Một cách hình thức, một phụ thuộc hàm $X \rightarrow Y$ là tầm thường nếu $X \supset Y$; ngược lại, nó được gọi là *không tầm thường*. Quy tắc tăng (QT2)nói rằng việc thêm cùng một tập thuộc tính vào cả hai vế của một phụ thuộc hàm sẽ tạo ra một phụ thuộc hàm đúng đắn. Theo QT3, các phụ thuộc hàm là bắc cầu. Quy tắc chiếu (QT4) nói rằng chúng ta có thể bỏ bớt các thuộc tính ra khỏi vế phải của phụ thuộc hàm. Việc áp dụng nhiều lần quy tắc này có thể tách phụ thuộc hàm $X \rightarrow. \{A_1, A_2, \ldots, A_n\}$ thành một tập hợp phụ thuộc hàm $\{ X \rightarrow A_1, X \rightarrow A_2, \ldots, X \rightarrow A_n\}$. Quy tắc hợp (QT5) cho phép chúng ta làm ngược lại; ta có thể gộp các phụ thuộc hàm $\{ X \rightarrow A_1, X \rightarrow A_2, \ldots, X \rightarrow A_n\}$thành một phụ thuộc hàm đơn $X \rightarrow \{A_1, A_2, \ldots, A_n\}$.

Có thể chứng minh các quy tắc suy diễn ở trên một cách trực tiếp hoặc bằng phản chứng dựa trên định nghĩa của phụ thuộc hàm. Để chúng minh phản chứng, ta giả thiết một quy tắc là không đúng và chỉ ra rằng điều đó là không thể. Sau đây là chứng minh các quy tắc.

Quy tắc 1:

Giả sử rằng $X \supset Y$ và hai bộ $t_1$ và $t_2$ trong một thể hiện quan hệ r của R sao cho $t_1[X] = t_2[X]$. Khi đó $t_1[Y] = t_2[Y]$ bởi vì $X \supset Y$; như vậy $X \rightarrow Y$ phải xẩy ra trong r.

Quy tắc 2 (chứng minh phản chứng):

Giả sử rằng $X \rightarrow Y$ đúng trong một thể hiện quan hệ r của R nhưng $XZ \rightarrow YZ$ không đúng. Khi đó phải có hai bộ $t_1$ và $t_2$ trong r sao cho

(1) $t_1[X] = t_2[X]$,

(2) $t_1[Y] = t_2[Y]$,

(3) $t_1[XZ] = t_2[XZ]$ và

(4) $t_1[YZ] \neq t_2[YZ]$. Điều đó là không thể bởi vì từ (1) và (3) chúng ta suy ra

(5) $t_1[Z] = t_2[Z]$, và từ (2) và (5) ta suy ra $t_1[YZ] = t_2[YZ]$, mâu thuẫn với (4).

Quy tắc 3: Giả sử ta có $X \rightarrow Y$ và $Y \rightarrow Z$ . Khi đó với mọi bộ $t_1$ và $t_2$ trong r, $t_1[X] = t_2[X]$ kéo theo $t_1[Y] = t_2[Y]$ (vì $X \rightarrow Y$), và $t_1[Y] = t_2[Y]$ kéo theo $t_1[Z] = t_2[Z]$ vì ($Y \rightarrow Z$). Như vậy, với mọi bộ $t_1$ và $t_2$ trong r, $t_1[X] = t_2[X]$ kéo theo $t_1[Z] = t_2[Z]$ hay là $X \rightarrow Z$.

Chúng ta có thể chứng minh các quy tắc từ QT4 đên QT6 theo phương pháp trên. Tuy nhiên ta có thể lợi dụng các quy tắc đã được chứng minh là đúng để chứng minh chúng. Sau đây ta chứng minh theo cách đấy.

Quy tắc 4:

1. $X \rightarrow YZ$ (cho trước)
2. $YZ \rightarrow Y$ (sử dụng QT1 và $YZ \supset Y$)
3. $X \rightarrow Y$ (sử dụng QT3 trên 1. và 2.)

Quy tắc 5:

1. $X \rightarrow Y$ (cho trước)
2. $X \rightarrow Z$ (cho trước)
3. $X \rightarrow YX$ (sử dụng QT2 trên 1. bằng cách thêm vào cả hai vế X, và XX=X)
4. $YX \rightarrow YZ$ (sử dụng QT2 trên 2. bằng cách thêm vào cả hai vế Y)
5. $X \rightarrow YZ$ (sử dụng QT3 trên 3. và 4.)

Quy tắc 6:

1. $X \rightarrow Y$ (cho trước)
2. $WY \rightarrow Z$ (cho trước)
3. $WX \rightarrow WY$ (sử dụng QT2 trên 1. bằng cách thêm vào c hai vế W)
4. $WX \rightarrow Y$ (sử dụng QT3 trên 3. và 2.)

Từ chứng minh ở trên, chúng ta thấy rằng chỉ cần 3 qui tắc QT1, QT2, QT3 là đủ, các quy tắc sau có thể suy diễn trực tiếp từ 3 qui tắc đó. Các quy tắc từ QT1 đến QT3 được gọi là các *quy tắc suy diễn Amstrong*.

### *3.2.3. Bao đóng của tập phụ thuộc hàm và bao đóng của tập thuộc tính dưới một tập phụ thuộc hàm.*

Thông thường, những người thiết kế cơ sở dữ liệu đầu tiên chỉ ra một tập các phụ thuộc hàm dễ xác định được nhờ ngữ nghĩa của các thuộc tính của R. Sau đó ta sử dụng các quy tắc Amstrong để suy diễn các phụ thuộc hàm bổ sung. Cho trước một tập phụ thuộc hàm F, tập hợp tất cả các phụ thuộc hàm suy ra được từ F bằng cách sử dụng các quy tắc suy diễn được gọi là bao đóng của tập F và được ký hiệu là $F^{+.}$

Ví dụ: $F = \{ X \rightarrow Y; Y \rightarrow T\}$

$F^+ = \{ F \cup \{X \rightarrow T, X \rightarrow YT \} \}$

Một cách có hệ thống, để xác định tất cả các phụ thuộc hàm bổ sung, đầu tiên hãy xác định mỗi tập thuộc tính X xuất hiện ở vế trái của một phụ thuộc hàm nào đấy trong F và sau đó xác định tập hợp tất cả các thuộc tính phụ thuộc hàm vào X. Như vậy, với mỗi tập thuộc tính X, chúng ta xác định tập $X^+$ các thuộc tính phụ thuộc hàm vào X dựa trên F. $X^+$ được gọi là bao đóng của X dưới F và được định nghĩa là:

$X^+ = \{A \in U \mid X \rightarrow A \in F^+ \}$.

Theo định nghĩa $X^+$ chúng ta có bổ đề sau:

**Bổ đề 3. 1:** $X \rightarrow Y$ được suy diễn từ tập phụ thuộc hàm F bằng các quy tắc suy diễn Amstrong khi và chỉ khi $Y \subseteq X^+$.

Thật vậy, giả sử $X \rightarrow Y$ được suy diễn từ tập phụ thuộc hàm F bằng các quy tắc suy diễn Amstrong và $Y = A_1A_2\ldots A_m$ với $A_1, A_2, \ldots A_m$ là các thuộc tính. Như vậy, theo quy tắc chiếu ta có $X \rightarrow A_1$, $X \rightarrow A_2$, …., $X \rightarrow A_m$. Theo định nghĩa $X^+$, $A_i \in X^+$ với i = 1, 2, …, m. Như vậy, $Y \subseteq X^+$.

Ngược lại, giả sử $Y \subseteq X^+$ và $Y = A_1A_2\ldots A_m$. Theo định nghĩa $X^+$ ta có $X \rightarrow Ai$ với i = i, . . m. Theo quy tắc hợp, ta có $X \rightarrow Y$.

Để xác định $X^+$ chúng ta sử dụng thuật toán sau:

**<u>Thuật toán 3. 1</u>** (xác định $X^+$, bao đóng của X dựa trên F)

$X^+ = X$;

Repeat

Old $X^+ = X^+$ ;

với mỗi phụ thuộc hàm $Y \rightarrow Z$ trong F thực hiện

nếu $X^+ \supset Y$ thì $X^+ = X^+ \cup Z$;

until ($X^+$ = Old $X^+$);

**Ví dụ :** Xét lược đồ quan hệ

R={MaSV, TenSV, Ngaysinh, MaMH, TenMH, DVHT, Diem}

Có tập phụ thuộc hàm:

F={MaSV→{TenSV, Ngaysinh}, MaMH →{TenMH, DVHT}, {MaSV, MaMH} → Diem}

Xác định {MaSV, MaMH}$^+$.

Áp dụng thuật toán 1. 1 ta có:

{MaSV}$^+$ = {MaSV, TenSV, NgaySinh}

{MaMH }$^+$ = {MaMH, TênMH, DVHT}

{MaSV, MaMH}$^+$ = {MaSV, TenSV, Ngaysinh, MaMH, TenMH, DVHT, Diem}.

**Định lý 3. 1:** Thuật toán 3.1 tính $X^+$ là đúng.

Giả sử ta có X → Y. Theo thuật toán ở trên, Y sẽ được thêm vào tập $X^+$, như vậy Y ⊆ $X^+$.

Ngược lại, giả sử Y ⊆ $X^+$. Theo cách xây dựng $X^+$ ta có X → $X^+$. Theo quy tắc phản xạ, ta có $X^+$→ Y. Vậy X →Y.

**Định lý 3. 2:** Hệ quy tắc suy diễn Amstrong là đúng và đầy đủ.

Chúng ta đã chứng minh tính đúng đắn của các quy tắc QT1, QT2, QT3 ở trên. Bây giờ ta chỉ cần chứng minh các quy tắc đó là đầy đủ, nghĩa là nếu X→Y không suy diễn lô gic được từ F bằng hệ suy diễn Amstrong thì X→Y không thoả mãn trên quan hệ r(R).

Để làm điều đó, giả sử rằng X→ Y không suy diễn được từ F bằng hệ suy diễn Amstrong, ta sẽ xây dựng một quan hệ r sao cho các phụ thuộc hàm của F là thoả mãn trên r nhưng X →Y không thoả mãn trên r. Quan hệ r được xây dựng như sau: r chỉ gồm hai bộ giá trị $t_1$ và $t_2$, trong đó các thuộc tính trong $t_1$ đều có giá trị 1, trong $t_2$ chỉ có các thuộc tính thuộc $X^+$ là có giá trị 1 còn các thuộc tính còn lại có giá trị 0.

t1 : 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

t2: 1 1 1 1 1 1 1 0 0 0 0 0 0 0

Ta chứng tỏ rằng mọi phụ thuộc hàm của F đều thoả mãn trên r. Thật vậy, giả sử có phụ thuộc hàm W→ V của F không thoả mãn trên r. Như vậy W⊆$X^+$ vì nếu

không sẽ vi phạm tính bằng nhau của W trên hai bộ t1 và t2. Hơn nữa V không thể là tập con của $X^+$ bởi vì nếu V là tập con của $X^+$ thì $W \rightarrow V$ sẽ thoả mãn trên r. Vậy phải có ít nhất một thuộc tính A của V là không thuộc $X^+$. Theo bổ đề 1. 1, nếu $W \subseteq X^+$ thì $X \rightarrow W$. Do $W \rightarrow V$ nên $X \rightarrow V$. Do A là một thuộc tính của V nên $X \rightarrow A$, hay A thuộc $X^+$. Điều đó là vô lý bởi vì A không thuộc $X^+$. Như vậy, mọi phụ thuộc hàm của F là thoả mãn trên r.

Bây giờ ta chứng tỏ rằng $X \rightarrow Y$ không thoả mãn trên r. Thật vậy, giả sử ngược lại $X \rightarrow Y$ thoả mãn trên r. Như vậy cả X và Y đều phải thuộc $X^+$ vì nếu không sẽ vi phạm sự bằng nhau trên các bộ t1 và t2 của X và Y. Nhưng nếu Y thuộc $X^+$ thì $X \rightarrow Y$ sẽ suy diễn được từ F theo bổ đề 3. 1. Điều đó mâu thuẫn với giả thiết $X \rightarrow Y$ không suy diễn được từ F. Vậy $X \rightarrow Y$ không thể thoả mãn trên r. Định lý được chứng minh.

### *3.2.4. Bao đóng và khóa*

Để ý rằng nếu $X^+$ là tập tất cả các thuộc tính của quan hệ thì có nghĩa là X xác định hàm các thuộc tính còn lại, hay nói cách khác X là một siêu khóa. Chúng ta có thể kiểm tra xem một tập thuộc tính X có phải là khóa của một quan hệ bằng cách trước tiên xem $X^+$ có chứa tất cả các thuộc tính của quan hệ hay không sau đó kiểm tra không có một tập con S nào được lập từ X bằng cách loại bỏ một thuộc tính của X thỏa mãn $S^+$ chứa tất cả các thuộc tính của quan hệ (nghĩa là X là siêu khóa tối thiểu). Ví dụ:

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

$F = \{AB \rightarrow F ; A \rightarrow CD; B \rightarrow E\}$

Ta có $\{A, B\}^+ = \{A, B, C, D, E, F\}$, $A^+ = \{A, C, D\}$, $B^+ = \{B, E\}$, vậy AB là khóa của quan hệ.

**Thuật toán 3. 2** Tìm một khóa K của R(U) dựa trên tập F các phụ thuộc hàm.

1) Đặt K := U;

2) Với mỗi thuộc tính A trong K, lặp lại các bước sau:

- tính $(K-A)^+$ đối với F ;
- Nếu $(K-A)^+$ chứa tất cả các thuộc tính trong U thì đặt K := K-{A } ;

Thuật toán 3. 2 cho phép chúng ta xác định được các khoá của một quan hệ xuất phát từ một siêu khoá ban đầu là tập tất cả các thuộc tính của quan hệ. Có thể thấy rằng việc tính khoá như vậy rất mất thời gian bởi vì nếu quan hệ có n thuộc tính thì nó có $2^n$

tập con. Nếu khoá của quan hệ chỉ có ít thuộc tính thì số lần tính các bao đóng để kiểm tra là rất lớn. Trên thực tế, người ta tìm khoá của quan hệ dựa trên nhận xét sau: Nếu quan hệ có khóa thì các thuộc tính khóa của quan hệ phải là các tập con của tập hợp các thuộc tính ở vế trái các phụ thuộc hàm trong F. Vì vậy, để tìm được các khóa nhanh hơn, trước tiên chúng ta tính $L_F$ là hợp của các thuộc tính ở các vế trái của các phụ thuộc hàm trong F, sau đó đi tính bao đóng của tất cả các tập con của $L_F$. Nếu bao đóng của tập con nào chứa tất cả các thuộc tính của R thì tập đó là một siêu khóa. Để kiểm tra nó là một khóa ta thực hiện như bước 2) của thuật toán trên.

***3.2.5. Tính tương đương của các tập phụ thuộc hàm***

Trong phần này chúng ta thảo luận về sự tương đương của hai tập phụ thuộc hàm. Một tập hợp các phụ thuộc hàm E được phủ bởi một tập các phụ thuộc hàm F - hoặc F phủ E - nếu mỗi một phụ thuộc hàm trong E đều ở trong $F^+$, điều đó có nghĩa là mỗi phụ thuộc hàm trong E có thể suy diễn được từ F. Hai tập phụ thuộc hàm E và F là tương đương nếu $E^+ = F^+$. Như vậy tương đương có nghĩa là mỗi phụ thuộc hàm trong E có thể suy diễn được từ F và mỗi phụ thuộc hàm trong F có thể suy diễn được từ E.

Cho hai tập phụ thuộc hàm E và F. Để chứng minh hai tập phụ thuộc hàm này tương đương, ta phải chứng minh các phụ thuộc hàm của E đều suy ra được từ F và ngược lại các phụ thuộc hàm của F đều suy ra được từ E. Để chứng minh phụ thuộc hàm $X \rightarrow Y$ suy ra được từ tập phụ thuộc hàm F chúng ta có thể thực hiện theo hai cách:

a. Áp dụng các quy tắc suy diễn để biến đổi các phụ thuộc hàm trong F cho đến khi nhận được $X \rightarrow Y$.

b. Áp dụng bổ đề 3. 1, tính $X^+$ (bao đóng của tập thuộc tính ở vế trái). Nếu $X^+ \supseteq Y$ thì $X \rightarrow Y$ suy ra được từ F

Ví dụ : Xét hai tập phụ thuộc hàm

$F = \{A \rightarrow C, AC \rightarrow D, E \rightarrow AD, E \rightarrow H\}$

$E = \{A \rightarrow CD, E \rightarrow AH\}$

+ Ta chứng minh hai tập phụ thuộc hàm này là tương đương theo cách a.

*Chứng minh E phủ F:*

$E = \{A \rightarrow CD, E \rightarrow AH\}$

$=\{A \rightarrow C, A \rightarrow D, E \rightarrow A, E \rightarrow H\}$ (áp dụng QT4 – chiếu)

= {A → C, A→ D, E → D, E→ A, E →H} (áp dụng QT3, bắc cầu)

= {A → C, A → D, E → AD, E → H} (áp dụng QT5 – hợp)

= { AC → C, A → D, E → AD, E → H} (áp dụng QT2)

= { AC → D, E → AD, E → H, A→ C} (áp dụng QT3)

*Chứng minh F phủ E:*

F= {A →C, AC → D, E→AD, E →H }

= {A→ C, A → D, E → A, E → D, E → H} (QT4, QT6)

= {A → CD, E → AH} (vì E → A, A → D lên có thể bỏ E → D)

+ Ta chứng minh hai tập phụ thuộc hàm này là tương đương theo cách b.

*Chứng minh F phủ E:*

Tìm bao đóng của các vế trái của các phụ thuộc hàm trong E theo F. Áp dụng thuật toán 3. 1 ở trên, ta có

$\{A\}^+$ = { A, C, D };

$\{E\}^+$ = {E, A, D, H},

ta thấy các bao đóng này chứa các vế phải tương ứng. Từ đó suy ra F phủ E.

C*hứng minh E phủ F* :

Tìm bao đóng của các vế trái của các phụ thuộc hàm trong F theo E. Ta có

$\{A\}^+$ ={A, C, D },

{AC}+ = { A, C, D},

{E}+ = { E, A, H}, ta thấy các bao đóng này chứa các vế phải tương ứng. Từ đó suy ra E phủ F.

Như vậy, E tương đương với F.

### *3.2.6. Các tập phụ thuộc hàm tối thiểu*

Một tập phụ thuộc hàm là *tối thiểu* nếu nó thoả mãn các điều kiện sau đây:

1. Vế phải của các phụ thuộc hàm trong F chỉ có một thuộc tính

2. Chúng ta không thể thay thế bất kỳ một phụ thuộc hàm X →A trong F bằng phụ thuộc hàm Y→A, trong đó Y là tập con đúng của X mà vẫn còn là một tập phụ thuộc hàm tương đương với F.

3. Chúng ta không thể bỏ đi bất kỳ phụ thuộc hàm nào ra khỏi F mà vẫn có một tập phụ thuộc hàm tương đương với F

Chúng ta có thể nghĩ về tập tối thiểu các phụ thuộc hàm như là một tập hợp ở dạng chuẩn không có sự dư thừa. Điều kiện 1 đảm bảo rằng mỗi phụ thuộc hàm là ở dạng chính tắc với một thuộc tính ở vế phải. Điều kiện 2 và 3 đảm bảo rằng không có sự dư thừa trong các phụ thuộc hoặc do có các thuộc tính dư thừa ở vế trái của phụ thuộc, hoặc do có một phụ thuộc có thể được suy diễn từ các phụ thuộc khác ở trong F.

Một *phủ tối thiểu* của một tập phụ thuộc hàm F là một tập tối thiểu các phụ thuộc hàm $F_{min}$ tương đương với F. Thường có rất nhiều các phủ tối thiểu cho một tập các phụ thuộc hàm. Chúng ta luôn luôn có thể tìm được ít nhất là một phủ tối thiểu G cho một tập các phụ thuộc hàm F bất kỳ theo thuật toán 3. 3 sau đây:

**Thuật toán 3. 3** (Tìm phủ tối thiểu G cho F).

1. Đặt G := F;

2. Thay thế mỗi phụ thuộc hàm $X \rightarrow \{A_1, A_2, \ldots, A_n\}$ trong G bằng n phụ thuộc hàm $X \rightarrow A_1$, $X \rightarrow A_2$, …, $X \rightarrow A_n$

3. Với mỗi phụ thuộc hàm $X \rightarrow A$ trong G và với mỗi thuộc tính B là một phần tử của X

nếu $(G - (X \rightarrow A)) \cup ((X - \{B\}) \rightarrow A)$ là tương đương với G

thì thay thế $X \rightarrow A$ bằng $(X-\{B\}) \rightarrow A$ ở trong G

4. Với mỗi phụ thuộc hàm $X \rightarrow A$ còn lại trong G

nếu $(G - \{ X \rightarrow A \})$ là tương đương với G thì loại bỏ $X \rightarrow A$ ra khỏi G.

*Ví dụ áp dụng* : Tìm phủ tối thiểu G cho tập phụ thuộc hàm:

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

Bước 1: $G = \{ A \rightarrow BC, B \rightarrow AC, C \rightarrow AB \}$

Bước 2: $G= \{A \rightarrow B, A \rightarrow C, B \rightarrow A, B \rightarrow C, C \rightarrow A, C \rightarrow B\}$

Bước 3: Do các phụ thuộc hàm trong G đều có vế trái gồm một thuộc tính nên G vẫn giữ nguyên.

Bước 4: Loại bỏ các phụ thuộc hàm thừa:

1) Do $A \rightarrow B$ và $B \rightarrow C$ nên $A \rightarrow C$ là thừa. Do $C \rightarrow B$ và $B \rightarrow A$ nên $C \rightarrow A$ là thừa. Bỏ những phụ thuộc hàm thừa đi, ta có $\{A \rightarrow B, B \rightarrow A, B \rightarrow C, C \rightarrow B\}$ là một phủ tối thiểu

2) Do A→B và B→C nên A→C là thừa. Do có B→C và C→A nên B→A là thừa. Do có C→A và A→B nên C→B là thừa. Bỏ những phụ thuộc hàm thừa đi, ta nhận được một phủ tối thiểu khác là {A→B, B→C, C→A}.

# CHƯƠNG 4. THIẾT KẾ CƠ SỞ DỮ LIỆU QUAN HỆ

Sau khi đã nghiên cứu các phụ thuộc hàm và một số tính chất của chúng, bây giờ chúng ta sẽ sử dụng chúng như thông tin về ngữ nghĩa của các lược đồ quan hệ. Ta giả sử rằng mỗi một quan hệ được cho trước một tập các phụ thuộc hàm và mỗi quan hệ có một khoá chính. Trong phần này chúng ta sẽ nghiên cứu các dạng chuẩn, quá trình chuẩn hoá các lược đồ quan hệ và thiết kế cơ sở dữ liệu quan hệ.

Có hai cách chính để thiết kế cơ sở dữ liệu quan hệ:

Cách thứ nhất là *thiết kế trên-xuống* (top-down design). Đây là cách hay được sử dụng nhất trong thiết kế ứng dụng cơ sở dữ liệu thương mại. Nó bao gồm việc thiết kế một lược đồ quan niệm trong một mô hình dữ liệu bậc cao, chẳng hạn như mô hình E-R, sau đó ánh xạ lược đồ quan niệm vào một tập quan hệ sử dụng các thủ tục ánh xạ. Mỗi một quan hệ được phân tích dựa trên các phụ thuộc hàm và các khóa chính được chỉ định bằng cách áp dụng các thủ tục chuẩn hóa để loại bỏ các phụ thuộc hàm bộ phận và các phụ thuộc hàm bắc cầu. Việc phân tích các phụ thuộc không mong muốn cũng có thể được thực hiện trong quá trình thiết kế quan niệm bằng cách phân tích các phụ thuộc hàm giữa các thuộc tính bên trong các kiểu thực thể và các kiểu liên kết để ngăn ngừa sự cần thiết có sự chuẩn hóa phụ thêm sau khi việc ánh xạ được thực hiện.

Cách thứ hai là *thiết kế dưới-lên* (bottom-up design), một kỹ thuật tiếp cận và nhìn nhận việc thiết kế lược đồ cơ sở dữ liệu quan hệ một cách chặt chẽ trên cơ sở các phụ thuộc hàm được chỉ ra trên các thuộc tính của cơ sở dữ liệu. Sau khi người thiết kế chỉ ra các phụ thuộc, người ta áp dụng một thuật toán chuẩn hóa để tổng hợp các lược đồ quan hệ. Mỗi một lược đồ quan hệ riêng rẽ ở dạng chuẩn 3NF hoặc BCNF hoặc ở dạng chuẩn cao hơn.

Trước tiên, chúng ta trình bày cách tiếp cận thứ hai. Chúng ta sẽ định nghĩa các dạng chuẩn một cách tổng quát, sau đó trình bày các thuật toán chuẩn hóa và các kiểu phụ thuộc khác. Chúng ta cũng sẽ trình bày chi tiết hơn về hai tính chất cần có là tách không mất mát và tách bảo toàn phụ thuộc. Các thuật toán chuẩn hóa thường bắt đầu bằng việc tổng hợp một lược đồ quan hệ rất lớn, gọi là *quan hệ phổ quát*

(universal relation), chứa tất cả các thuộc tính của cơ sở dữ liệu. Sau đó chúng ta thực hiện lặp đi lặp lại việc tách (decomposition) dựa trên các phụ thuộc hàm và các phụ thuộc khác do người thiết kế cơ sở dữ liệu chỉ ra cho đến khi không còn tách được nữa hoặc không muốn tách nữa.

Tiếp theo, chúng ta trình bày về cách tiếp cận thứ nhất để thiết kế cơ sở dữ liệu. Đi từ các lược đồ khái niệm đến lược đồ vật lý.

**4. 1. Nhập môn về chuẩn hoá**

Quá trình chuẩn hoá (do Codd đề nghị 1972) là xét một lược đồ quan hệ và thực hiện một loạt các kiểm tra để xác nhận nó có thoả mãn một dạng chuẩn nào đó hay không. Quá trình này được thực hiện theo phương pháp trên xuống bằng việc đánh giá mỗi quan hệ với tiêu chuẩn của các dạng chuẩn và tách các quan hệ nếu cần. Quá trình này có thể xem như là việc thiết kế quan hệ bằng phân tích. Lúc đầu, Codd đề nghị ba dạng chuẩn gọi là dạng chuẩn 1, dạng chuẩn 2 và dạng chuẩn 3. Một định nghĩa mạnh hơn cho dạng chuẩn 3 gọi là dạng chuẩn Boyce-Codd do Boyce và Codd đề nghị sau đó. Tất cả các dạng chuẩn này dựa trên các phụ thuộc hàm giữa các thuộc tính của một quan hệ. Sau đó, dạng chuẩn 4 (4NF) và dạng chuẩn 5 (5NF) được đề nghị dựa trên các phụ thuộc hàm đa trị và các phụ thuộc hàm nối.

*Chuẩn hoá dữ liệu* có thể được xem như một quá trính phân tích các lược đồ quan hệ cho trước dựa trên các phụ thuộc hàm và các khoá chính của chúng để đạt đến các tính chất mong muốn :

(1) Cực tiểu sự dư thừa và

(2) Cực tiểu các phép cập nhật bất thường.

Các lược đồ quan hệ không thoả mãn các kiểm tra dạng chuẩn sẽ được tách ra thành các lược đồ quan hệ nhỏ hơn thoả mãn các kiểm tra và có các tính chất mong muốn. Như vậy, thủ tục chuẩn hoá cung cấp cho người thiết kế cơ sở dữ liệu:

a. Một cơ cấu hình thức để phân tích các lược đồ quan hệ dựa trên các khoá của nó và các phụ thuộc hàm giữa các thuộc tính của nó.

b. Một loạt các kiểm tra dạng chuẩn có thể thực hiện trên các lược đồ quan hệ riêng rẽ sao cho cơ sở dữ liêu quan hệ có thể được chuẩn hoá đến một mức cần thiết.

Dạng chuẩn của một quan hệ liên quan đến điều kiện dạng chuẩn cao nhất mà nó thoả mãn. Các dạng chuẩn khi được xem xét độc lập với các sự kiện khác không

đảm bảo một thiết kế cơ sở dữ liệu tốt. Nói chung, việc xác minh riêng biệt từng lược đồ quan hệ ở dạng chuẩn này dạng chuẩn nọ là chưa đủ. Tốt hơn là quá trình chuẩn hoá thông qua phép tách phải khẳng định một vài tính chất hỗ trợ mà tất cả các lược đồ quan hệ phải có. Chúng gồm hai tính chất sau:

- Tính chất nối không mất mát (hoặc nối không phụ thêm), đảm bảo rằng vấn đề tạo ra các bộ giả không xuất hiện đối với các lược đồ quan hệ được tạo ra sau khi tách.
- Tính chất bảo toàn phụ thuộc, nó đảm bảo rằng từng phụ thuộc hàm sẽ được biểu hiện trong các quan hệ riêng rẽ nhận được sau khi tách.

Tính chất nối không mất mát là rất quan trọng, phải đạt được bằng mọi cách, còn tính chất bảo toàn phụ thuộc thì cũng rất mong muốn nhưng đôi khi có thể hy sinh.

**4.2. Định nghĩa tổng quát các dạng chuẩn.**

Nói chung, chúng ta muốn thiết kế các lược đồ của chúng ta sao cho chúng không còn các phụ thuộc bộ phận và các phụ thuộc bắc cầu bởi vì các kiểu phụ thuộc này gây ra các sửa đổi bất thường. Trong phần này chúng ta sẽ đưa ra các định nghĩa về các dạng chuẩn tổng quát hơn, có tính đến tất cả các khóa dự tuyển. Cụ thể, *thuộc tính khóa* được định nghĩa lại *là một bộ phận của một khóa dự tuyển*. Các phụ thuộc hàm bộ phận, đầy đủ, bắc cầu bây giờ sẽ được định nghĩa đối với tất cả các khóa dự tuyển của quan hệ.

***4.2.1. Định nghĩa dạng chuẩn 1 (First Normal Form - 1NF)***

Định nghĩa: Một lược đồ quan hệ R là ở dạng chuẩn 1 (1NF) nếu miền giá trị của các thuộc tính của nó chỉ chứa các *giá trị nguyên tử* (đơn, không phân chia được) và giá trị của một thuộc tính bất kỳ trong một bộ giá trị phải là một giá trị đơn thuộc miền giá trị của thuộc tính đó.

Ví dụ: Thực thể: Hóa đơn bán hàng

| SoHD | Ngay | HoTen | DiaChi | MatHang | SoLuong | DonGia |
|---|---|---|---|---|---|---|
| HDB01 | 01/02/2006 | Nguyen A | 100 HQV | Chuot | 01 | 90000 |
| | | | | B.Phim | 01 | 70000 |

- MatHang, SoLuong, DonGia: Là những thuộc tính không đơn trị.
- Khả năng lưu trữ là khó khăn, xử lý phức tạp
- Giải pháp: Tách bảng thành nhiều bảng con

HÓAĐƠN:

| SoHD | Ngay | HoTen | DiaChi |
|---|---|---|---|
| HDB01 | 01/02/2006 | Nguyen A | 100 HQV |

CHITIẾTHÓAĐƠN

| SoHD | MatHang | SoLuong | DonGia |
|---|---|---|---|
| HDB01 | Chuot | 01 | 90000 |
| HDB01 | B.Phim | 01 | 70000 |

### *4.2.2. Định nghĩa dạng chuẩn 2 (Second Normal Form - 2NF )*

*Định nghĩa:* Một lược đồ quan hệ R là ở dạng chuẩn 2 (2NF) nếu mỗi thuộc tính không khóa A trong R không phụ thuộc bộ phận vào một khóa bất kỳ của R.

Ví dụ: Xét lược đồ quan hệ

R={A, B, C, D, E, F}

Với các phụ thuộc hàm:

A $\rightarrow$ BCDEF;

BC $\rightarrow$ ADEF;

B $\rightarrow$F;

D$\rightarrow$E.

Lược đồ trên có hai khóa dự tuyển là A và BC. Ta chọn A làm khóa chính. Do có phụ thuộc hàm B $\rightarrow$ F nên F phụ thuộc bộ phận vào khóa B, C, lược đồ vi phạm chuẩn 2NF. (Chú ý rằng, trong định nghĩa dạng chuẩn dựa trên khóa chính, lược đồ này không vi phạm 2NF).

Ví dụ về CHITIẾTBÁNHÀNG có phụ thuộc hàm {SoHD, MaH} → {SoLuong, DonGia}, {MaH}→ {TenHang, DonVi}. Lược đồ có khóa chính là {SoHD, MaH}. Do có TenHang, Donvi là thuộc tính không khóa phụ thuộc một phần vào thuộc tính khóa.

| SoHD | MaH | TenHang | SoLuong | DonGia | DonVi |
|---|---|---|---|---|---|
| HDB01 | HH01 | B. Phím | 10 | 100000 | Cái |

Tách thành hai bảng: CHITIẾTBÁN và HÀNGHÓA

CHI TIẾT BÁN

| SoHD | MaH | SoLuong | DonGia |
|---|---|---|---|
| HDB01 | HH01 | 10 | 100000 |

HÀNG HÓA

| MaH | TenHang | DonVi |
|---|---|---|
| HH01 | B. Phím | Cái |

***4.2.3. Định nghĩa dạng chuẩn 3 (Third Normal Form - 3NF )***

Định nghĩa: Một lược đồ quan hệ R là ở dạng chuẩn 3 (3NF) nếu khi một phụ thuộc hàm X → A thỏa mãn trong R, thì

1) Hoặc X là một siêu khóa của R
2) Hoặc A là một thuộc tính khóa của R

Ví dụ : Xét lược đồ quan hệ R ở ví dụ trên. Giả sử nó được tách thành hai lược đồ

R1 = { A, B, C, D, E }

R2 = { B, F}.

Do có phụ thuộc hàm D → E trong đó D không phải thuộc tính khóa, E cũng không phải là thuộc tính khóa, nên R1 vi phạm chuẩn 3NF. Do đó R1 sẽ tách thành hai quan hệ sau: R11={A,B,C,D}, R12={D,E}.

***4.2.4. Định nghĩa dạng chuẩn Boyce - Codd (Boyce - Codd Normal Form – BCNF)***

*Định nghĩa:* Một lược đồ quan hệ là ở dạng chuẩn Boyce-Codd (BCNF) nếu khi một phụ thuộc hàm X → A thỏa mãn trong R thì X là một siêu khóa của R.

Ví dụ: Xét lược đồ R = {A, B, C, D} có A là khóa chính và {B, C} là khóa dự tuyển. Nếu có tồn tại một phụ thuộc hàm D $\rightarrow$ B thì lược đồ này vi phạm BCNF vì B là một thuộc tính khóa. (Chú ý rằng trong trường hợp định nghĩa dạng chuẩn dựa trên khóa chính, lược đồ này không vi phạm BCNF).

**4. 3 . Phép tách các lược đồ quan hệ**

*Tách quan hệ:* Các thuật toán thiết kế cơ sở dữ liệu quan hệ được trình bày trong phần này bắt đầu từ một lược đồ quan hệ phổ quát R = {$A_1$, $A_2$, …, $A_n$} chứa tất cả các thuộc tính của cơ sở dữ liệu. Với giả thiết quan hệ phổ quát, tên của mỗi thuộc tính là duy nhất. Tập hợp F các phụ thuộc hàm thỏa mãn trên các thuộc tính của R do những người thiết kế cơ sở dữ liệu chỉ ra sẽ được các thuật toán sử dụng. Sử dụng các phụ thuộc hàm, các thuật toán sẽ tách lược đồ quan hệ phổ quát R thành một tập hợp các lược đồ quan hệ D = {$R_1$, $R_2$, …, $R_m$}, tập hợp đó sẽ là lược đồ cơ sở dữ liệu quan hệ. D được gọi là một phép tách của R. Như vậy:

Phép tách một lược đồ quan hệ R = {$A_1$, $A_2$, …, $A_n$} là thay thế nó bằng một tập các lược đồ con D = {$R_1$, $R_2$, …, $R_m$}, trong đó các $R_i$ với i=1, 2, . . m là các tập con của R và $R_1 \cup R_2 \cup R_3 \cup \ldots \cup R_m = R$.

*4.3.2.1. Phép tách và sự bảo toàn phụ thuộc*

Việc mỗi phụ thuộc hàm X $\rightarrow$ Y trong F hoặc được xuất hiện trực tiếp trong một trong các lược đồ quan hệ $R_i$ trong phép tách D hoặc có thể được suy diễn từ các phụ thuộc hàm có trong $R_i$ là rất có lợi. Ta gọi đó là *điều kiện bảo toàn phụ thuộc.* Chúng ta muốn bảo toàn phụ thuộc bởi vì mỗi phụ thuộc trong F biểu thị một ràng buộc trong cơ sở dữ liệu. Bây giờ chúng ta định nghĩa các khái niệm này một cách hình thức.

Cho trước một tập hợp các phụ thuộc F trên R, *phép chiếu của F trên $R_i$,* ký hiệu là $\pi_{Ri}(F)$ trong đó $R_i$ là một tập con của R, là một tập hợp các phụ thuộc hàm X$\rightarrow$Y trong $F^+$ sao cho các thuộc tính trong X $\cup$ Y đều được chứa trong $R_i$. Như vậy, phép chiếu của F trên mỗi lược đồ quan hệ $R_i$ trong phép tách D là tập hợp các phụ thuộc hàm trong $F^+$, bao đóng của F, sao cho các thuộc tính ở vế trái và vế phải của chúng đều ở trong $R_i$. Ta nói rằng phép tách D = {R1, R2, …, Rm} của R bảo toàn phụ thuộc đối với F nếu hợp của các phép chiếu của F trên mỗi $R_i$ trong D là tương đương với F. Điều đó có nghĩa là

$((\pi_{R1}(F)) \cup (\pi_{R2}(F)) \cup \ldots \cup (\pi_{Rm}(F)))^+ = F^+$

Nếu một phép tách là không bảo toàn phụ thuộc, một vài phụ thuộc sẽ bị mất trong phép tách. Để kiểm tra xem một phụ thuộc hàm $X \rightarrow B$, trong đó X là tập thuộc tính thuộc về $R_i$, B là một thuộc tính thuộc $R_i$ có thỏa mãn trong $R_i$ hay không ta làm như sau: Trước hết tính $X^+$, sau đó với mỗi thuộc tính B sao cho:

1. B là một thuộc tính của $R_i$
2. B là ở trong $X^+$
3. B không ở trong X

Khi đó phụ thuộc hàm $X \rightarrow B$ thỏa mãn trong $R_i$.

Một ví dụ về phép tách không bảo toàn phụ thuộc.

Xét lược đồ quan hệ R = { A, B, C, D},

với các phụ thuộc hàm $A \rightarrow BCD$ ; $BC \rightarrow DA$ ; $D \rightarrow B$;

Lược đồ này có hai khóa dự tuyển là A và BC. Giả sử nó được tách thành

R1 = { D, B}, lược đồ này chứa phụ thuộc hàm $D \rightarrow B$

R2 = { A, C, D }, lược đồ này chứa phụ thuộc hàm $A \rightarrow CD$.

Rõ ràng sau khi tách, phụ thuộc hàm $BC \rightarrow DA$ bị mất.

**Định lý 4. 1**: Luôn luôn tìm được một phép tách bảo toàn phụ thuộc D đối với F sao cho mỗi quan hệ $R_i$ trong D là ở 3NF. Phép tách D đượcc thực hiện theo thuật toán sau đây:

**Thuật toán 4. 1**: Tạo một phép tách bảo toàn phụ thuộc $D = \{R_1, R_2, \ldots, R_m\}$ của một quan hệ phổ quát R dựa trên một tập phụ thuộc hàm F sao cho mỗi $R_i$ trong D là ở 3NF. Thuật toán này chỉ đảm bảo tính chất bảo toàn phụ thuộc, không đảm bảo tính chất nối không mất mát.

Input: Một quan hệ vũ trụ R và một tập phụ thuộc hàm F trên các thuộc tính của R

1) Tìm phủ tối thiểu G của F
2) Với mỗi vế trái X của một phụ thuộc hàm xuất hiện trong G, hãy tạo một lược đồ trong D với các thuộc tính $\{ X \cup \{A_1\} \cup \{A_2\} \cup \ldots \cup \{A_k\}\}$ trong đó $X \rightarrow A_1$, $X \rightarrow A_2$, …, $X \rightarrow A_k$ chỉ là các phụ thuộc hàm trong G với X là vế trái (X là khóa của quan hệ này).

3) Đặt các thuộc tính còn lại (những thuộc tính chưa được đặt vào quan hệ nào) vào một quan hệ đơn để đảm bảo tính chất bảo toàn thuộc tính.

*Ví dụ áp dụng:*

Xét lược đồ : R = { A, B, C, D}, với các phụ thuộc hàm

F = {A → BCD ; BC → DA ; D →B} ;

Lược đồ này có hai khóa dự tuyển là A và BC.

Ta thực hiện thuật toán như sau:

*Bước 1)* Trước tiên ta tìm G là phủ tối thiểu của F. Theo thuật toán tìm phủ tối thiểu, đầu tiên ta làm cho các vế phải trong G chỉ chứa một thuộc tính, ta có:

G = { A → B; A → C; A→ D; BC → D; BC → A; D → B}

Sau đó ta bỏ đi các phụ thuộc hàm thừa (là các phụ thuộc hàm có thể suy diễn được từ các phụ thuộc hàm khác).

Ta thấy A →B là thừa vì có A →D, D→B;

BC → D là thừa vì BC→ A và A → D.

Vậy G còn lại là G = {A → C; A→ D; BC → A; D → B}.

*Bước 2)* Ghép các phụ thuộc hàm có cùng vế trái vào lược đồ con. Lược đồ R sẽ được tách thành: $R_1$(A, C, D); R2(B, C, A); R3(D, B) với các khóa chính được gạch dưới.

**Định lý 4. 2.** Thuật toán tách lược đồ R = {$A_1$, $A_2$, …, $A_n$} thành tập các lược đồ con $R_i$, (i = 1, 2, . . m) ở 3NF và bảo toàn phụ thuộc.

Chứng minh:

Rõ ràng rằng tất cả các phụ thuộc hàm trong G đều được thuật toán bảo toàn bởi vì mỗi phụ thuộc xuất hiện trong một trong các quan hệ của phép tách D. Bởi vì G tương đương với F, tất cả các phụ thuộc của F cũng được bảo toàn hoặc trực tiếp bằng thuật toán hoặc được suy diễn từ những phụ thuộc hàm trong các quan hệ kết quả, như vậy tính chất bảo toàn phụ thuộc được đảm bảo. Do G là phủ tối thiểu nên trong G không có những phụ thuộc hàm bộ phận và phụ thuộc hàm bắc cầu, do vậy các phụ thuộc hàm trong G đều là phụ thuộc hàm trực tiếp, điều đó có nghĩa là các $R_i$ đều ở 3NF

*4.3.2.2 Phép tách không mất mát thông tin (Lossless join)*

Phép tách D phải có một tính chất nữa là tách không mất mát (hoặc tính chất nối không phụ thêm), nó đảm bảo rằng không có các bộ giả được tạo ra khi áp dụng một phép nối tự nhiên vào các quan hệ trong phép tách.

Một cách hình thức, ta nói rằng một phép tách D = {$R_1$, $R_2$, …, $R_m$} của R có tính chất không mất mát thông tin đối với một tập hợp phụ thuộc hàm F trên R nếu với mỗi trạng thái quan hệ r của R thỏa mãn F thì

$$(\pi_{R1}(r) * \pi_{R2}(r) * \ldots\ldots * \pi_{Rm}(r)) = r$$

trong đó * là phép nối tự nhiên của các quan hệ trong D, $\pi_{Ri}(r)$ là phép chiếu của r trên $R_i$.

Nếu một phép tách không có tính chất không mất mát thông tin thì chúng ta có thể nhận được các bộ phụ thêm (các bộ giả) sau khi áp dụng các phép chiếu và nối tự nhiên. Nghĩa của từ mất mát ở đây là mất mát thông tin chứ không phải mất các bộ giá trị. Vì vậy, với tính chất này ta nên gọi chính xác hơn là tính chất nối không phụ thêm.

Chúng ta có thuật toán để kiểm tra một phép tách có tính chất nối không mất mát thông tin hay không như sau:

**Thuật toán 4.2:** Kiểm tra tính chất nối không mất mát

Input: Một quan hệ vũ trụ R($A_1$, $A_2$, …$A_n$), một phép tách D = {$R_1$, $R_2$, …, $R_m$} của R và một tập F các phụ thuộc hàm.

1) Tạo một ma trận S có m hàng, n cột. Mỗi cột của ma trận ứng với một thuộc tính, mỗi hàng ứng với mỗi quan hệ $R_i$

2) Đặt S(i, j) = 1 nếu thuộc tính $A_j$ thuộc về quan hệ $R_i$ và bằng 0 trong trường hợp ngược lại

3) Lặp lại vòng lặp sau đây cho đến khi nào việc thực hiện vòng lặp không làm thay đổi S: Với mỗi phụ thuộc hàm $X \rightarrow Y$ trong F, xác định các hàng trong S có các ký hiệu 1 như nhau trong các cột ứng với các thuộc tính trong X. Nếu có một hàng trong số đó chứa 1 trong các cột ứng với thuộc tính Y thì hãy làm cho các cột tương ứng của các hàng khác cũng chứa 1.

4) Nếu có một hàng chứa toàn ký hiệu "1" thì phép tách có tính chất nối không mất mát; ngược lại phép tách không có tính chất đó.

**Định lý 4.3.** Thuật toán 4.2 là đúng.

*Chứng minh:*

Cho trước một quan hệ R được tách thành một số quan hệ $R_1$, $R_2$, . ., $R_m$. Thuật toán 4.2 bắt đầu bằng việc tạo ra một trạng thái quan hệ r trong ma trận S. Hàng i trong S biểu diễn một bộ $t_i$ (tương ứng với quan hệ $R_i$). Hàng này có các ký hiệu "1" trong các cột tương ứng với các thuộc tính của $R_i$ và các ký hiệu "0" trong các cột còn lại. Như vậy, $t_i[R_i]$ bao gồm các giá trị 1. Sau đó thuật toán biến đổi các hàng của ma trận này (trong vòng lặp của bước 3) sao cho chúng biểu diễn các bộ thỏa mãn tất cả các phụ thuộc hàm trong F. Ở cuối vòng lặp áp dụng các phụ thuộc hàm, hai hàng bất kỳ trong S – chúng biểu diễn hai bộ trong r – có các giá trị giống nhau đối với các thuộc tính của X ở vế trái của phụ thuộc hàm X→ Y trong F sẽ cũng có các giá trị giống nhau đối với các thuộc tính của vế phải Y. Có thể thấy rằng sau khi áp dụng vòng lặp của bước 3, nếu một hàng bất kỳ trong S kết thúc với toàn ký hiệu "1" thì điều đó có nghĩa là $\pi_{R1}(r) * \pi_{R2}(r)*……* \pi_{Rm}(r)) = r$ hay là D có tính chất tách không mất mát đối với F. Mặt khác, nếu không có hàng nào kết thúc bằng tất cả ký hiệu "1" thì D không thỏa mãn tính chất tách không mất mát. Trong trường hợp sau, trạng thái quan hệ r được biểu diễn bằng S ở cuối thuật toán sẽ là một ví dụ về một trạng thái quan hệ r của R thỏa mãn các phụ thuộc trong F nhưng không thỏa mãn điều kiện tách không mất mát. Như vậy, quan hệ này được dùng như một phản ví dụ chứng minh rằng D không có tính chất tách không mất mát đối với F. Chú ý rằng các ký hiệu "1" và "0" không có ý nghĩa đặc biệt gì ở cuối thuật toán.

Ví dụ áp dụng 1:

R = (MãsốNV, TênNV, MãsốDA, TênDA, ĐịađiểmDA, Sốgiờ)

R1= (TênNV, ĐịađiểmDA)

R2 = (MãsốNV, MãsốDA, Sốgiờ, TênDA, ĐịađiểmDA)

F= {MãsốNV→ TênNV, MãsốDA → {TênDA, ĐịađiểmDA}, {MãsốNV, MãsốDA}→ Sốgiờ }

| | MãsốNV | TênNV | Mã sốDA | TênDA | ĐịađiểmDA | Sốgiờ |
|---|---|---|---|---|---|---|
| R1 | | 1 | | | 1 | |
| R2 | 1 | | 1 | 1 | 1 | 1 |

Xét lần lượt phụ thuộc hàm MãsốNV→TênNV, MãsốDA→{TênDA, ĐịađiểmDA}, {MãsốNV, Mã sốDA}→ Sốgiờ. Ta thấy không có trường hợp nào các thuộc tính tương ứng với các vế trái đều có giá trị bằng 1, vì vậy ta không thể làm gì

để biến đổi ma trận. Ma trận không chứa một hàng gồm toàn ký hiệu "1". Phép tách là mất mát.

Ví dụ áp dụng 2

R = (MãsốNV, TênNV, MãsốDA, TênDA, ĐịađiểmDA, Sốgiờ)

R1= (MãsốNV, TênNV)

R2 = { MãsốDA, TênDA, ĐịađiểmDA }

R3 = (MãsốNV, MãsốDA, Sốgiờ)

F= {MãsốNV→ TênNV, MãsốDA → {TênDA, ĐịađiểmDA}, {MãsốNV, Mã sốDA}→ Sốgiờ }

| | MãsốNV | TênNV | Mã sốDA | TênDA | ĐịađiểmDA | Sốgiờ |
|---|---|---|---|---|---|---|
| R1 | 1 | 1 | 0 | 0 | 0 | 0 |
| R2 | 0 | 0 | 1 | 1 | 1 | 0 |
| R3 | 1 | 0 | 1 | 0 | 0 | 1 |

(Giá trị ban đầu của ma trận S)

Xét phụ thuộc hàm MãsốNV→ TênNV. Ta thấy hàng thứ 1 và hàng thứ 3 của S có chứa 1 tại MãsốNV, trong đó hàng thứ 1 chứa 1 tại TênNV nên ta sửa lại giá trị của TênNV ở hàng thứ 3 thành 1.

Xét phụ thuộc hàm MãsốDA →{TênDA, ĐịađiểmDA}, ta thấy hàng thứ 2 và hàng thứ 3 của S chứa 1 tại MãsốDA trong đó hàng thứ hai chứa 1 tại TênDA và ĐịađiểmDA nên ta sửa giá trị của các thuộc tính đó tại hàng thứ 3 thành 1.

| MãsốNV | TênNV | Mã sốDA | TênDA | ĐịađiểmDA | Sốgiờ |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 1 | 1 | 0 1 | 0 1 | 1 |

(Ma trận S sau khi áp dụng hai phụ thuộc hàm đầu tiên dòng cuối cùng chứa toàn ký hiệu "a"). Ma trận chứa một hàng gồm toàn ký hiệu 1. Phép tách này là không mất mát.

**Hình 4. 1.** Thuật toán kiểm tra nối không mất mát

Thuật toán 4.2 cho phép chúng ta kiểm tra xem một phép tách D cụ thể có tuân theo tính chất nối không mất mát hay không. Câu hỏi tiếp theo là liệu có một thuật toán tách một lược đồ quan hệ phổ quát $R = \{ A_1, A_2, \ldots, A_n \}$ bằng một phép tách $D = \{R_1, R_2, \ldots, R_m\}$ sao cho mỗi $R_i$ là ở BCNF và phép tách D có tính chất nối không mất mát đối với F hay không?. Câu trả lời là có. Trước khi trình bày thuật toán, ta xem một số tính chất của các phép tách nối không mất mát nói chung:

Tính chất 1: Một phép tách $D = \{R_1, R_2\}$ của R có tính chất không mất mát thông tin đối với một tập phụ thuộc hàm F trên R khi và chỉ khi

- Hoặc phụ thuộc hàm $((R_1 \cap R_2) \rightarrow (R_1 - R_2))$ ở trong $F^+$
- Hoặc phụ thuộc hàm $((R_1 \cap R_2) \rightarrow (R_2 - R_1))$ ở trong $F^+$

Với tính chất này, chúng ta có thể kiểm tra lại các phép tách chuẩn hóa trong 4.2 và sẽ thấy rằng các phép tách đó là thỏa mãn tính chất nối không mất mát.

Tính chất 2 : Nếu một phép tách $D = \{R_1, R_2, \ldots, R_m\}$ của R có tính chất nối không mất mát đối với một tập phụ thuộc hàm F trên R và nếu một phép tách $D_1 = \{Q_1, Q_2, \ldots, Q_k\}$ của $R_i$ có tính chất nối không mất mát đối với phép chiếu của F trên Ri thì phép tách $D2 = \{ R_1, R_2, \ldots, R_{i-1}, Q_1, Q_2, \ldots, Q_k, R_{i+1}, \ldots, R_m\}$ của R có tính chất nối không mất mát đối với F.

Tính chất này nói rằng nếu một phép tách D đã có tính chất nối không mất mát đối với một tập F và chúng ta tiếp tục tách một trong các quan hệ Ri trong D thành phép tách khác $D_1$ (l = 1, 2, . . k) có tính chất nối không mất mát đối với $\pi_{Ri}(F)$ thì việc thay $R_i$ trong D bằng $D_1$ (l = 1, 2, . . k) cũng tạo ra một phép tách có tính chất nối không mất mát đối với F.

Thuật toán 4.3 sau đây sử dụng hai tính chất trên để tạo ra một phép tách D = $\{R_1, R_2, \ldots, R_m\}$ của một quan hệ vũ trụ R dựa trên một tập các phụ thuộc hàm F sao cho mỗi $R_i$ là BCNF.

**Thuật toán 4.3.** Tách quan hệ thành các quan hệ BCNF với tính chất nối không mất mát

Input: Một quan hệ vũ trụ R và một tập hợp các phụ thuộc hàm F trên các thuộc tính của R

1. Đặt D := {R} ;
2. Khi có một lược đồ quan hệ Q trong D không phải ở BCNF, thực hiện vòng lặp: Với mỗi một lược đồ quan hệ Q trong D không ở BCNF hãy tìm một phụ thuộc hàm $X \rightarrow Y$ trong Q vi phạm BCNF và thay thế Q trong D bằng hai lược đồ quan hệ (Q-Y) và $(X \cup Y)$. Quá trình lặp dừng khi không còn quan hệ nào trong D vi phạm BCNF.

Mỗi lần đi vào vòng lặp trong thuật toán 4.3, chúng ta tách một quan hệ Q không phải BCNF thành hai lược đồ quan hệ. Theo các tính chất 1 và 2, phép tách D có tính chất nối không mất mát. Kết thúc thuật toán, tất cả các quan hệ trong D sẽ ở BCNF.

Trong bước 2 của thuật toán 4.3, cần xác định xem một lược đồ quan hệ Q có ở BCNF hay không. Một phương pháp để làm điều đó là kiểm tra. Với mỗi phụ thuộc hàm $X \rightarrow Y$ trong Q, ta tính $X^+$. Nếu $X^+$ không chứa tất cả các thuộc tính trong Q thì $X \rightarrow Y$ vi phạm BCNF bởi vì X không phải là một siêu khóa.

Một kỹ thuật nữa dựa trên quan sát rằng khi một lược đồ quan hệ Q vi phạm BCNF thì có tồn tại một cặp thuộc tính A, B trong Q sao cho $\{ Q- \{A, B\}\} \rightarrow A$. Bằng việc tính bao đóng $\{Q - \{A, B\}\}^+$ cho mỗi cặp thuộc tính {A, B} của Q và kiểm tra xem bao đóng có chứa A (hoặc B) hay không, chúng ta có thể xác định được Q có ở BCNF hay không.

Ví dụ áp dụng: Xét lược đồ quan hệ

R = { A, B, C, D, E, F)

Với các phụ thuộc hàm:

$A \rightarrow BCDEF$,

BC → ADEF,

B→ F,

D→ E,

D→ B

Lược đồ quan hệ này có hai khóa dự tuyển là A và BC.

Ta có B → F vi phạm BCNF vì B không phải là siêu khóa, R được tách thành

R1(B, F) với phụ thuộc hàm B→ F

R2(A, B, C, D, E) với các phụ thuộc hàm A→ BCDE, BC→ ADE, D→ E, D→ B

Do D→ E vi phạm BCNF (D là một thuộc tính không khóa), R2 được tách thành:

R21(D, E) với phụ thuộc hàm D →E

R22(ABCD) với các phụ thuộc hàm A → BCD, BC→ AD, D→ B

Do D → B vi phạm BCNF (D không phải là thuộc tính khóa), R22 được tách thành

R221(D, B)

R222(A, C, D) với phụ thuộc hàm A→ CD (phụ thuộc hàm BC → AD bị mất)

Tóm lại, ta có phép tách D = {R1, R21, R221, R222}. Phép tách này có tính chất nối không mất thông tin nhưng không bảo toàn phụ thuộc. Nếu chúng ta muốn có một phép tách có tính chất nối không mất mát và bảo toàn phụ thuộc thì ta phải hài lòng với các lược đồ quan hệ ở dạng 3NF. Thuật toán sau đây là cải tiến của thuật toán 4.3, tạo ra một phép tách thỏa mãn :

- Bảo toàn phụ thuộc
- Có tính chất nối không mất mát
- Mỗi lược đồ quan hệ kết quả là ở dạng 3NF.

**Thuật toán 4.4** Thuật toán tổng hợp quan hệ với tính chất bảo toàn phụ thuộc và không mất mát

Input: Một quan hệ vũ trụ R và một tập các phụ thuộc hàm F trên các thuộc tính của R

1) Tìm phủ tối thiểu G cho F

2) Với mỗi vế trái X của một phụ thuộc hàm xuất hiện trong G hãy tạo ra một lược đồ quan hệ trong D với các thuộc tính $\{X \cup \{A_1\} \cup \{A_2\} \cup \ldots \cup \{A_k\}\}$, trong đó $X \rightarrow A_1$, $X \rightarrow A_2$, …, $X \rightarrow A_k$ chỉ là các phụ thuộc hàm ở trong G với X là vế trái (X là khóa của quan hệ này)
3) Nếu không có lược đồ quan hệ nào trong D chứa một khóa của R thì hãy tạo ra thêm một lược đồ quan hệ trong D chứa các thuộc tính tạo nên một khóa của R.

Không phải lúc nào cũng có khả năng tìm được một phép tách thành các lược đồ quan hệ bảo toàn phụ thuộc và mỗi lược đồ trong phép tách là ở BCNF. Các lược đồ quan hệ trong phép tách theo thuật toán ở trên thường là 3NF. Để có các lược đồ BCNF, chúng ta có thể kiểm tra các lược đồ quan hệ 3NF trong phép tách một cách riêng rẽ để xem nó có thỏa mãn BCNF không. Nếu có lược đồ quan hệ $R_i$ không ở BCNF thì ta có thể tách tiếp hoặc để nguyên nó là 3NF.

**4. 4. Các phụ thuộc hàm đa trị và dạng chuẩn 4 (4NF - Fourth Normal Form)**

Trong phần này chúng ta thảo luận khái niệm phụ thuộc hàm đa trị và định nghĩa dạng chuẩn 4. Các phụ thuộc đa trị hệ quả của dạng chuẩn 1 không cho phép một thuộc tính của một bộ có một tập giá trị (nghĩa là các thuộc tính đa trị). Nếu chúng ta có hai hoặc nhiều hơn các thuộc tính đôc lập và đa trị trong cùng một lược đồ quan hệ thì chúng ta phải lặp lại mỗi một giá trị của một trong các thuộc tính với mỗi giá trị của thuộc tính khác để giữ cho trạng thái quan hệ nhất quán và duy trì tính độc lập giữa các thuộc tính. Ràng buộc đó được chỉ ra bằng một phụ thuộc đa trị.

***4.4.1. Định nghĩa phụ thuộc đa trị***

Giả thiết có một lược đồ quan hệ R, X và Y là hai tập con của R. Một phụ thuộc đa trị, ký hiệu là $X \twoheadrightarrow Y$, chỉ ra ràng buộc sau đây trên một trạng thái quan hệ bất kỳ của R: Nếu hai bộ $t_1$ và $t_2$ tồn tại trong R sao cho $t_1[X] = t_2[X]$ thì hai bộ $t_3$ và $t_4$ cũng tồn tại trong R với các tính chất sau:

- $t_3[X] = t_4[X] = t_1[X] = t_2[X]$
- $t_3[Y] = t_1[Y]$ và $t_4[Y] = t_2[Y]$
- $t_3[Z] = t_2[Z]$ và $t_4[Z] = t_1[Z]$ với $Z = (R- (X \cup Y))$

Khi X→→Y thỏa mãn, ta nói rằng X đa xác định Y. Bởi vì tính đối xứng trong định nghĩa, khi X →→ Y thỏa mãn trong R, X→→Z cũng thỏa mãn trong R. Như vậy X→→Y kéo theo X→→Z và vì thế đôi khi nó được viết là X→→Y|Z

Định nghĩa hình thức chỉ ra rằng, cho trước một giá trị cụ thể của X, tập hợp các giá trị của Y được xác định bởi giá trị này của X là được xác định hoàn toàn bởi một mình X và không phụ thuộc vào các giá trị của các thuộc tính còn lại Z của R. Như vậy, mỗi khi hai bộ tồn tại có các giá trị khác nhau của Y nhưng cùng một giá trị X thì các giá trị này của Y phải được lặp lại trong các bộ riêng rẽ với mỗi giá trị khác nhau của Z có mặt với cùng giá trị của X. Điều đó tương ứng một cách không hình thức với Y là một thuộc tính đa trị của các thực thể được biểu diễn bằng các bộ trong R.

Ví dụ về phụ thuộc đa trị:

| NHÂNVIÊN | TênNV | TênDA | TênconNV |
|---|---|---|---|
| | Nam | DA01 | Lan |
| | Nam | DA02 | Hoa |
| | Nam | DA01 | Hoa |
| | Nam | DA02 | Lan |

Trong bảng trên có hai phụ thuộc đa trị là TênNV→→ TênDA, TênNV→→ TênconNV

Một phụ thuộc đa trị X→→Y được gọi phụ thuộc đa trị tầm thường nếu

a) Y là một tập con của X
b) Hoặc X ∪ Y = R

Một phụ thuộc đa trị không thỏa mãn a) hoặc b) được gọi là một phụ thuộc đa trị không tầm thường. Nếu chúng ta có một phụ thuộc đa trị không tầm thường trong một quan hệ, chúng ta có thể phải lặp các giá trị một cách dư thừa trong các bộ. Trong quan hệ NHÂNVIÊN ở ví dụ trên, các giá trị 'DA01', 'DA02' của TênDA được lặp lại với mỗi giá trị của TênconNV (một cách đối xứng, các giá trị 'Lan', 'Hoa' được lặp lại với mỗi giá trị của TênDA). Rõ ràng ta không mong muốn có sự dư thừa đó. Tuy nhiên, lược đồ quan hệ trên là ở BCNF bởi vì không có phụ thuộc hàm nào thỏa mãn trong quan hệ đó. Vì vậy, chúng ta phải định nghĩa một dạng chuẩn thứ tư mạnh hơn BCNF và ngăn cấm các lược đồ quan hệ như quan hệ NHÂNVIÊN.

### *4.4.2. Các quy tắc suy diễn đối với các phụ thuộc hàm và phụ thuộc đa trị*

Các quy tắc từ qt1 đến qt8 sau đây tạo nên một tập hợp đúng đắn và đầy đủ cho việc suy diễn các phụ thuộc hàm và phụ thuộc đa trị từ một tập các phụ thuộc cho trước. Giả thiết rằng tất cả các thuộc tính được chứa trong một lược đồ quan hệ "vũ trụ" R = {$A_1$, $A_2$, …, $A_n$} và X, Y, Z, W là các tập con của R. (FD ký hiệu phụ thuộc hàm, MVD ký hiệu phụ thuộc đa trị)

Qt1) (quy tắc phản xạ cho FD): Nếu $X \supseteq Y$ thì $X \rightarrow Y$

Qt2) (quy tắc tăng cho FD): { $X \rightarrow Y$} |= $XZ \rightarrow YZ$

Qt3) (Quy tắc bắc cầu cho FD): { $X \rightarrow Y$, $Y \rightarrow Z$ } |= $X \rightarrow Z$

Qt4) (quy tắc bù cho MVD): {$X \rightarrow\rightarrow Y$ } |= {$X \rightarrow\rightarrow$ (R-(X ∪ Y))}

Qt5) (quy tắc tăng cho MVD): Nếu $X \rightarrow\rightarrow Y$ và $W \supseteq Z$ thì $WX \rightarrow\rightarrow YZ$

Qt6) (quy tắc bắc cầu cho MVD): { $X \rightarrow\rightarrow Y$, $Y \rightarrow\rightarrow Z$ } |= $X \rightarrow\rightarrow (Z - Y)$

Qt7) (quy tắc tái tạo cho FD và MVD): {$X \rightarrow Y$} |= $X \rightarrow\rightarrow Y$

Qt8) (quy tắc liên hợp cho FD và MVD): Nếu $X \rightarrow\rightarrow Y$ và có tồn tại W với các tính chất

a) $W \cap Y = \square$,

b) $W \rightarrow Z$ và

c) $Y \supseteq Z$

thì $X \rightarrow Z$.

Qt1 đến Qt3 là các quy tắc suy diễn Amstrong đối với các phụ thuộc hàm. Qt4 đến Qt6 là các quy tắc suy diễn chỉ liên quan đến các phụ thuộc đa trị. Qt7 và Qt8 liên kết các phụ thuộc hàm và các phụ thuộc đa trị. Đặc biệt, Qt7 nói rằng một phụ thuộc hàm là một trường hợp đặc biệt của một phụ thuộc đa trị. Điều đó có nghĩa là mỗi phụ thuộc hàm cũng là một phụ thuộc đa trị bởi vì nó thỏa mãn định nghĩa hình thức của phụ thuộc đa trị. Về cơ bản, một phụ thuộc hàm $X \rightarrow Y$ là một phụ thuộc đa trị $X \rightarrow\rightarrow Y$ với một hạn chế phụ rằng có nhiều nhất là một giá trị của Y được kết hợp với mỗi giá trị của X. Cho trước một tập hợp các phụ thuộc hàm và phụ thuộc đa trị chỉ ra trên R = { A1, A2, …, An}, chúng ta có thể sử dụng các quy tắc từ qt1 đến qt8 để suy ra tập hợp đầy đủ các phụ thuộc (hàm và đa trị) $F^+$ đúng trong mọi trạng thái quan hệ r của R thỏa mãn F. Chúng ta lại gọi $F^+$ là bao đóng của F.

### *4.4.3. Dạng chuẩn 4*

*Định nghĩa dạng chuẩn 4:* Một lược đồ quan hệ R là ở dạng chuẩn 4 (4NF) đối với một tập hợp các phụ thuộc F (gồm các phụ thuộc hàm và phụ thuộc đa trị) nếu với mỗi phụ thuộc đa trị không tầm thường $X \rightarrow\rightarrow Y$ trong $F^+$, X là một siêu khóa của R.

Như vậy, một lược đồ quan hệ vi phạm 4NF nếu nó chứa các phụ thuộc hàm đa trị không mong muốn. Ví dụ, lược đồ quan hệ NHÂNVIÊN ở ví dụ trên là vi phạm 4NF bởi vì trong các phụ thuộc hàm đa trị TênNV$\rightarrow\rightarrow$TênDA và TênNV$\rightarrow\rightarrow$ TênconNV, TênNV không phải là một siêu khóa.

Giả sử chúng ta tách bảng NHÂNVIÊN thành hai bảng như sau:

| **NV_DA** | **TênNV** | **TênDA** |
|---|---|---|
| | Nam | DA01 |
| | Nam | DA02 |

| **NV_CON** | **TênNV** | **TênconNV** |
|---|---|---|
| | Nam | Lan |
| | Nam | Hoa |

Hai bảng này là ở 4NF bởi vì các phụ thuộc đa trị TênNV$\rightarrow\rightarrow$TênDA và TênNV $\rightarrow\rightarrow$ TênconNV là các phụ thuộc đa trị tầm thường. Trong hai bảng này không có các phụ thuộc đa trị không tầm thường cũng như không có các phụ thuộc hàm.

### *4.4.4. Tách có tính chất nối không mất mát thành các quan hệ chuẩn 4NF*

Khi chúng ta tách một lược đồ quan hệ R thành $R_1 = (X \cup Y)$ và $R_2 = (R-Y)$ dựa trên phụ thuộc hàm đa trị $X \rightarrow\rightarrow Y$ đúng trong R, phép tách có tính chất nối không mất mát. Đó cũng là điều kiện cần và đủ cho một phép tách một lược đồ thành hai lược đồ có tính chất nối không mất mát. Ta có tính chất sau:

Tính chất 1': Các lược đồ quan hệ $R_1$ và $R_2$ tạo thành một phép tách có tính chất nối không mất mát của R khi và chỉ khi $(R_1 \cap R_2) \rightarrow\rightarrow (R_1 - R_2)$ (hoặc $(R_1 \cap R_2) \rightarrow\rightarrow (R_1 - R_2)$).

Áp dụng tính chất trên chúng ta có thuật toán tạo một phép tách có tính chất nối không mất mát thành các lược đồ quan hệ ở dạng 4NF.

**Thuật toán 4.5** Tách quan hệ thành các quan hệ 4NF với tính chất nối không mất mát.

Input: Một quan hệ vũ trụ R, một tập phụ thuộc hàm và phụ thuộc đa trị F

1. Đặt D := {R} ;
2. Khi có một lược đồ quan hệ Q trong D không ở 4NF, thực hiện

{Chọn một lược đồ quan hệ Q trong D không ở 4NF;

Tìm một phụ thuộc đa trị không tầm thường $X \rightarrow\rightarrow Y$ trong Q vi phạm 4NF;

Thay thế Q trong D bằng hai lược đồ quan hệ $(Q - Y)$ và $(X \cup Y)$; }.

*Ví dụ áp dụng 1:*

Xét lược đồ NHÂNVIÊN(TênNV, TênDA, TênconNV). Ta có phụ thuộc hàm đa trị TênNV→→TênDA trong đó TênNV không phải là một siêu khóa, vậy nó vi phạm 4NF. Ta tách thành NV_DA(TênNV, TênDA), NV_CON(TênNV, TênconNV).

*Ví dụ áp dụng 2:*

Cho quan hệ SẢNXUẤT như sau:

| SẢNXUẤT | Phânxưởng | Nhânviên | Sảnphẩm |
|---|---|---|---|
| | Phân xưởng 1 | Hoàng | Bu lông |
| | Phân xưởng 1 | Yến | Đinh |
| | Phân xưởng 1 | Hoàng | Đinh |
| | Phân xưởng 1 | Yến | Bu lông |
| | Phân xưởng 2 | Minh | Ốc vít |
| | Phân xưởng 2 | Hải | Kìm điện |
| | Phân xưởng 2 | Minh | Kìm điện |
| | Phân xưởng 2 | Hải | Ốc vít |

Trong bảng trên có hai phụ thuộc đa trị là Phânxưởng→→Nhânviên, Phânxưởng→→Sảnphẩm. Quan hệ SẢNXUẤT vi phạm 4NF bởi vì trong các phụ

thuộc hàm đa trị có Phânxưởng không phải là một siêu khóa. Chúng ta tách quan hệ SẢNXUẤT thành hai quan hệ PX_NV(), PX_SP():

| NV_DA | Phânxưởng | Nhânviên |
|---|---|---|
| | Phân xưởng 1 | Hoàng |
| | Phân xưởng 1 | Yến |
| | Phân xưởng 2 | Minh |
| | Phân xưởng 2 | Hải |

| NV_CON | Phânxưởng | Sảnphẩm |
|---|---|---|
| | Phân xưởng 1 | Bu lông |
| | Phân xưởng 1 | Đinh |
| | Phân xưởng 2 | Ốc vít |
| | Phân xưởng 2 | Kìm điện |

**4. 5. Các phụ thuộc nối và dạng chuẩn 5 (Fifth Normal Form - 5NF)**

Như chúng ta đã thấy, các tính chất 1 và tính chất 1' cho điều kiện để một lược đồ quan hệ R được tách thành hai lược đồ quan hệ R1 và R2 và phép tách có tính chất nối không mất mát. Tuy nhiên, trong một số trường hợp, có thể không có phép tách có tính chất nối không mất mát của R thành hai lược đồ quan hệ nhưng có thể có phép tách có tính chất nối không mất mát thành nhiều hơn hai quan hệ. Hơn nữa, có thể không có phụ thuộc hàm nào trong R các chuẩn cho đến BCNF và có thể không có phụ thuộc đa trị nào có trong R vi phạm 4NF. Khi đó chúng ta phải sử dụng đến một phụ thuộc khác gọi là phụ thuộc nối và nếu có phụ thuộc nối thì thực hiện một phép tách đa chiều thành dạng chuẩn 5 (5NF).

Một *phụ thuộc nối*, ký hiệu là JD($R_1$, $R_2$, …, $R_n$) trên lược đồ quan hệ R chỉ ra một ràng buộc trên các trạng thái r của R. Ràng buộc đó tuyên bố rằng mỗi trạng thái hợp pháp r của R phải có phép tách có tính chất nối không mất mát thành R1, R2, ..., Rn. Điều đó nghĩa là:

$*(\pi_{R1}(r), \pi_{R2}(r), \ldots\pi_{Rn}(r)) = r$

Một phụ thuộc nối JD($R_1$, $R_2$, …, $R_n$) là một phụ thuộc nối tầm thường nếu một trong các lược đồ quan hệ $R_i$ ở trong JD($R_1$, $R_2$, …, $R_n$) là bằng R.

*Định nghĩa dạng chuẩn 5:* Một lược đồ quan hệ R là ở dạng chuẩn 5 (5NF) (hoặc dạng chuẩn nối chiếu (PJNF- Project-Join normal form) đối với một tấp F các

phụ thuộc hàm, phụ thuộc đa trị và phụ thuộc nối nếu với mỗi phụ thuộc nối không tầm thường JD($R_1$, $R_2$, …, $R_n$) trong $F^+$, mỗi Ri là một siêu khóa của R.

Ví dụ 1: Xét quan hệ CUNGCẤP gồm toàn các thuộc tính khóa

| CUNGCẤP | **Tênnhàcungcấp** | **Tênhàng** | **TênDựán** |
|---|---|---|---|
| | Ncc1 | Bulong | Dựán1 |
| | Ncc1 | Đaiốc | Dựán2 |
| | Ncc2 | Bulong | Dựán2 |
| | Ncc3 | Đaiốc | Dựán3 |
| | Ncc2 | Đinh | Dựán1 |
| | Ncc2 | Bulong | Dựán1 |
| | Ncc1 | Bulong | Dựán2 |

Giả thiết rằng ràng buộc phụ thêm sau đây luôn đúng: Khi một nhà cung cấp S cung cấp hàng P *và* một dự án J sử dụng hàng P *và* nhà cung cấp S cung cấp ít nhất là một hàng cho dự án J *thì* nhà cung cấp S cũng sẽ cung cấp hàng P cho dự án J. Ràng buộc này chỉ ra một phụ thuộc nối JD($R_1$, $R_2$, $R_3$) giữa ba phép chiếu $R_1$(Tênnhàcungcấp, Tênhàng), $R_2$(Tênnhàcungcấp, Têndựán), $R_3$(Tênhàng, TênDựán) của quan hệ CUNGCẤP. Quan hệ CUNGCẤP được tách thành ba quan hệ $R_1$, $R_2$, $R_3$ ở dạng chuẩn 5. Chú ý rằng nếu ta áp dụng phép nối tự nhiên cho từng đôi quan hệ một thì sẽ sinh ra các bộ giả, nhưng nếu áp dụng phép nối tự nhiên cho cả ba quan hệ thì không sinh ra các bộ giả.

R1

| Tênnhàcungcấp | Tênhàng |
|---|---|
| Ncc1 | Bulong |
| Ncc1 | Đaiốc |
| Ncc2 | Bulong |
| Ncc3 | Đaiốc |
| Ncc2 | Đinh |

R2

| Tênnhàcungcấp | Têndựán |
|---|---|
| Ncc1 | Dựán1 |
| Ncc1 | Dựán2 |
| Ncc2 | Dựán2 |
| Ncc3 | Dựán3 |
| Ncc2 | Dựán1 |

R3

| Tênhàng | Têndựán |
|---|---|
| Bulong | Dựán1 |
| Đaiốc | Dựán2 |
| Bulong | Dựán2 |
| Đaiốc | Dựán3 |
| Đinh | Dựán1 |

Ví dụ 2: Cho quan hệ NGOẠIKHÓA

| NGOẠIKHÓA | Họcviên | Hoạtđộng | Câulạcbộ |
|---|---|---|---|

| | | |
|---|---|---|
| Minh | Khiêu vũ | Clb1 |
| Minh | Chơi ghi ta | Clb2 |
| Ngọc | Đánh trống | Clb1 |
| Ngọc | Hát | Clb3 |
| Hải | Múa | Clb1 |
| Minh | Diễn kịch | Clb3 |
| Hải | Chơi piano | Clb2 |

Quan hệ NGOẠIKHÓA có các ràng buộc sau: Họcviên X có hoạt động Y, hoạt động Y thuộc câu lạc bộ Z. Học viên X muốn tham gia hoạt động Y thì phải đăng ký ở câu lạc bộ Z. Các ràng buộc này chỉ ra một phụ thuộc nối JD(R1, R2, R3) giữa 3 phép chiếu R1(Họcviên, hoạtđộng), R2(Họcviên, câulạcbộ), R3(câulạcbộ, hoạtđộng) của quan hệ NGOẠIKHÓA. Quan hệ NGOẠIKHÓA được tách thành ba quan hệ $R_1$, $R_2$, $R_3$ ở dạng chuẩn 5.

| Họcviên | hoạtđộng |
|---|---|
| Minh | Khiêu vũ |
| Minh | Chơi ghi ta |
| Ngọc | Đánh trống |
| Ngọc | Hát |
| Hải | Múa |
| Minh | Diễn kịch |
| Hải | Chơi piano |

| Họcviên | câulạcbộ |
|---|---|
| Minh | Clb1 |
| Minh | Clb2 |
| Ngọc | Clb1 |
| Ngọc | Clb3 |
| Hải | Clb1 |
| Minh | Clb3 |
| Hải | Clb2 |

| câulạcbộ | hoạtđộng |
|---|---|
| Clb1 | Khiêu vũ |
| Clb2 | Chơi ghi ta |
| Clb1 | Đánh trống |
| Clb3 | Hát |
| Clb1 | Múa |
| Clb3 | Diễn kịch |
| Clb2 | Chơi piano |

Việc phát hiện các phụ thuộc nối trong các cơ sở dữ liệu thực tế với hàng trăm thuộc tính là một điều rất khó khăn. Vì vậy, thực tiễn thiết kế cơ sở dữ liệu hiện nay thường không chú ý đến nó. Nói chung, trong thực tế thiết kế cơ sở dữ liệu, người ta chỉ chuẩn hóa các bảng đến 3NF, BCNF là đủ.

**4.6. Mô hình dữ liệu**

### *4.6.1. Khái niệm*

Mô hình dữ liệu (Data model) là một tập hợp các khái niệm dùng để diễn tả tập hợp dữ liệu và hành động để thao tác lên dữ liệu

Các khái niệm trong mô hình dữ liệu được xây dựng bởi cơ chế trừu tượng hóa và mô tả bằng ngôn ngữ hay biểu diễn đồ họa.

### *4.6.2. Các mô hình dữ liệu thông dụng*

#### *4.6.2.1. Mô hình phân cấp*

Mô hình phân cấp (hierarchy) được xây dựng từ thập kỷ 70. Đó chính là một mạng có nhiều cây. Hình 4.2 ví dụ minh họa mô hình phân cấp.


**Hình 4.2.**. Minh họa mô hình phân cấp

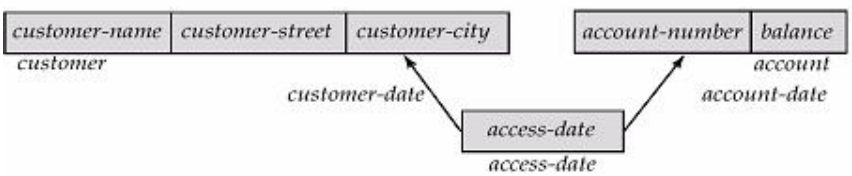
#### *4.6.2.2. Mô hình mạng*

Mô hình mạng cũng được xây dựng từ thập kỷ 70. Trong mô hình này, dữ liệu được trình bày dưới dạng một tập các mẫu tin (record). Các mẫu tin và các thuộc tính được biểu diễn bởi kiểu mẫu tin (record type).

```
type  customer = record          type  account = record
      customer-name: string;           account-number: integer;
      customer-street: string;         balance: integer;
      customer-city: string;
end                              end
```

**Hình 4.3.** Các dạng mẫu tin customer, account.

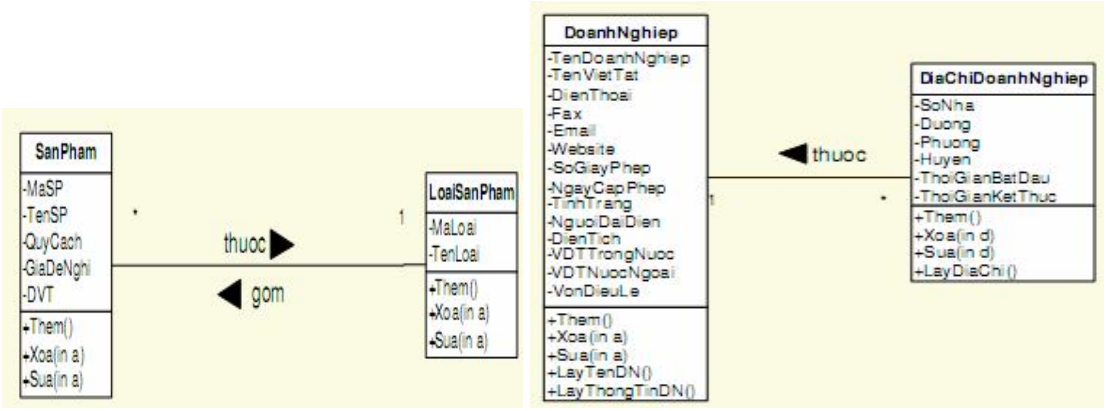
Mối quan hệ giữa các mẫu tin được biểu diễn bằng đường nối (links). Trên đường nối này biểu diễn các mối quan hệ nhiều – nhiều (many – to – many), một – nhiều (one-to-many), một – một (one – to – one).




**Hình 4.4.** Mối quan hệ của mô hình mạng

#### *4.6.2.3. Mô hình hướng đối tượng*

Mô hình hương đối tượng được xây dựng từ giữa thập kỷ 80 – đến nay.




**Hình 4.5.** Ví dụ về mô hình hướng đối tượng

### *4.6.3. Chất lượng của mô hình dữ liệu*

- Tính diễn đạt: cho phép mô tả một khối lượng lớn đa dạng các khái niệm sao cho có thể biểu diễn toàn diện hơn thế giới thực. Do đó, các mô hình dữ liệu phải giàu về các khái niệm và cả tính diễn đạt.

- Tính đơn giản: mô hình dữ liệu phải đơn giản để cho lược đồ xây dựng bằng mô hình sẽ được những người thiết kế và người sử dụng thông hiểu dễ dàng hơn.

- Tính tốt thiểu: Mô hình sẽ có tính tối thiểu nếu mọi khái niệm trình bày trong mô hình có một ý nghĩa phân biệt khi xem xét trong các mối quan hệ đến mọi khía cạnh khác.

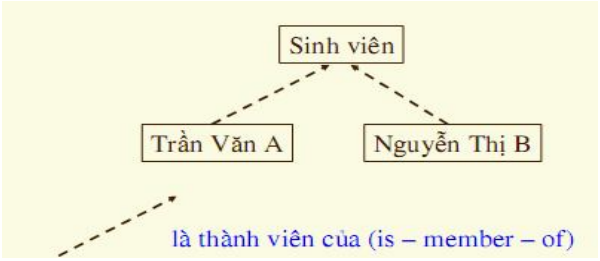
- Tính hình thức: các lược đồ tạo ra bởi các mô hình dữ liệu biểu diễn một đặc tả hình thức dữ liệu. Tính hình thức này đòi hỏi các khái niệm của mô hình sẽ được thể hiện đồng nhất, chính xác và định nghĩa tốt.

- Tính mở rộng: mô hình dữ liệu phải có tính mở cho tương lai.

## 4.7. Mô hình thực thể kết hợp (Entity Relationship – ER)

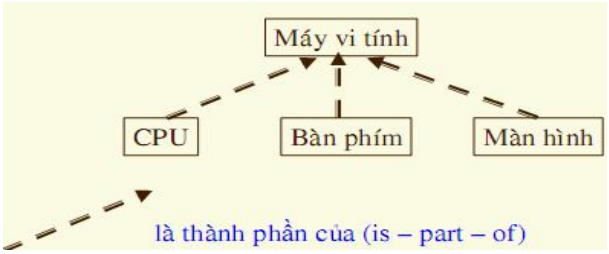
### *4.7.1. Trừu tượng hóa*

- Phân loại:



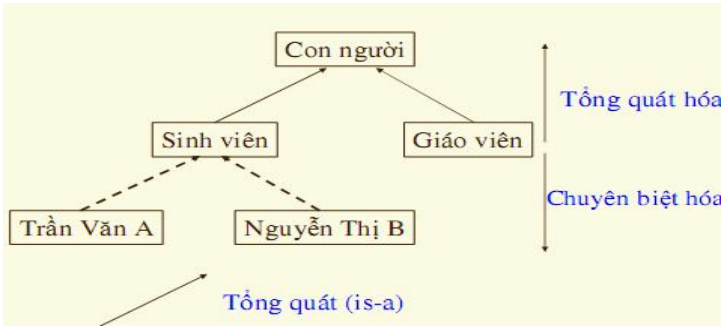

**Hình 4.6.** Mô hình phân loại

- Kết hợp:




**Hình 4.7.** Ví dụ minh họa kết hợp

### *4.7.2. Tổng quát hóa*




**Hình 4.8.** Minh họa tổng quát hóa

### *4.7.3. Mô hình thực thể kết hợp (ER)*

Mô hình thực thể kết hợp:

+ Dễ dùng

+ Hỗ trợ Case tool

+ Dùng để xây dựng mô hình ý niệm

+ Mô hình này ra đời năm 1970 bởi Mr.Chen, tiếp tục được phát triển bởi Teory, Chang và Fry vào năm 1986 và Storey vào năm 1991. Được xem là một chuẩn giúp mô hình hóa dữ liệu.

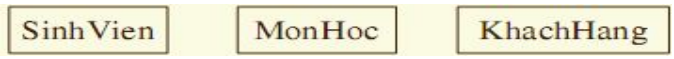
+ Là sự biểu diễn đồ họa các thực thể và các mối liên hệ của chúng trong cấu trúc một database.

*4.7.3.1. Các thành phần chính của mô hình ER*

a) Thực thể: Là một người, nơi chốn, đối tượng, sự kiện hay một khái niệm trong thế giới thực (có thể là khái niệm trừu tượng) và bắt đầu là danh từ

+ Thực thể mạnh: sự tồn tại độc lập với các kiểu thực thể khác. Thực thể mạnh có thể xác định thực thể yếu qua từ khóa: [redacted]

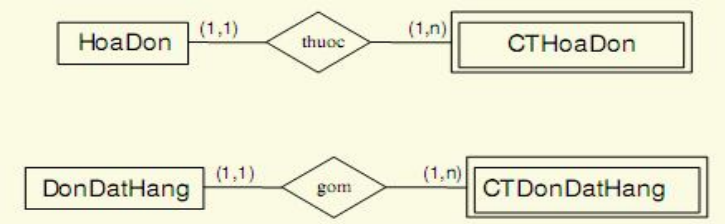
Ví dụ: SinhVien, Monhoc, KhachHang, NhanVien, SanPham, HoaDon



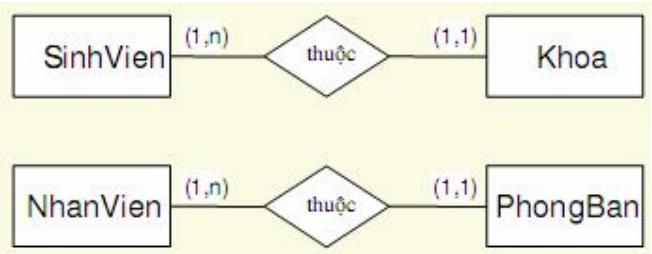

+ Thực thể yếu: Không có đủ các thuộc tính để hình thành nên khóa, tồn tại phụ thuộc vào các thực thể khác. Ký hiệu

- Một tập thực thể yếu có thể có khóa riêng để phân biệt giữa các thực thể yếu có mối liên quan với cùng một thực thể mạnh.
- Khóa của tập thực thể yếu = khóa của tập thuộc tính cha + khóa riêng của tập thực thể yếu

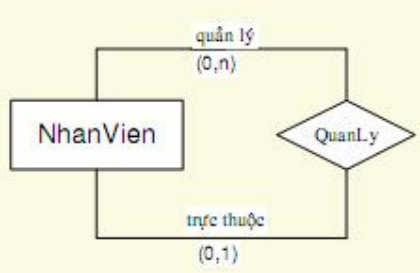
Ví dụ:



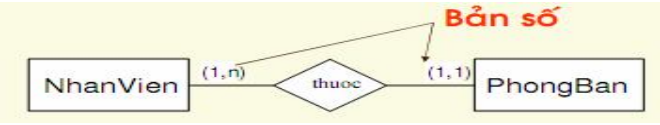

b) Mối kết hợp: Diễn tả mối quan hệ giữa các thực thể. Ký hiệ



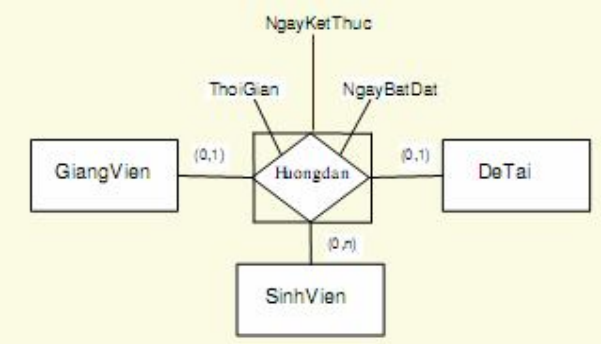

- Mối kết hợp một ngôi (Phản thân)




- Mối kết hợp 2 ngôi (nhị phân)




- Mối kết hợ 3 ngôi (đa phân)

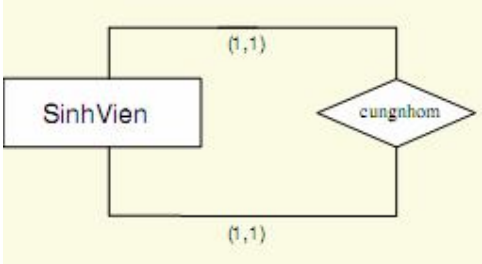



c) Bản số của mối kết hợp

Xét tập quan hệ 2 ngôi giữa hai thực thể A và B:

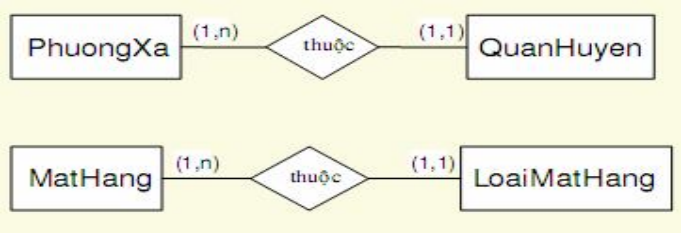
+ Một – một (one-to-one): một thực thể A kết hợp với nhiều nhất một thực thể B và một thực thể B kết hợp với nhiều nhất một thực thể A.

Ví dụ: Một phòng ban có duy nhất 1 trưởng phòng.



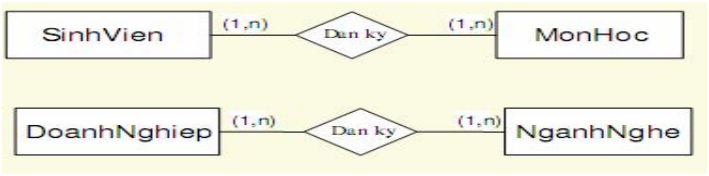

+ Một – nhiều (one to many): Một thực thể A kết hợp với nhiều thực thể trong B và một thực thể B kết hợp với nhiều nhất chỉ một thực thể trong A.

Ví dụ: Một nhân viên chỉ thuộc một phòng ban và một phòng ban có nhiều nhân viên




+ Nhiều – nhiều (many to many): Một thực thể A kết hợp với nhiều thực thể trong B và một thực thể B kết hợp với nhiều thực thể trong A.

Ví dụ: Một sinh viên học nhiều môn học và một môn học có nhiều sinh viên đăng ký học.

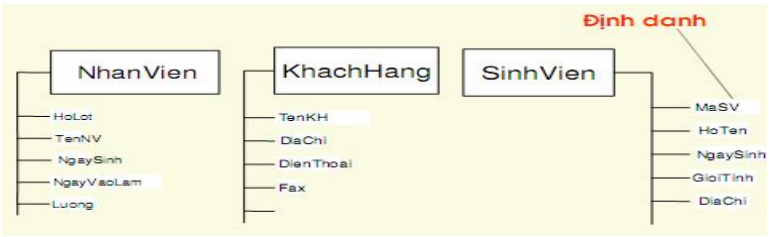


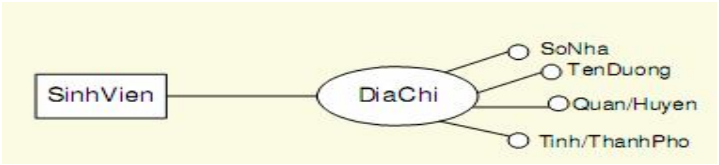

d) Thuộc tính: là đặc tính, tính chất đặc trưng của thực thể hoặc mối kết hợp nhằm để mô tả thực thể hay mối kết hợp.

Ví dụ: thực thể SinhVien có các thuộc tính hoten, tuoi, gioitinh, ngaysinh,…

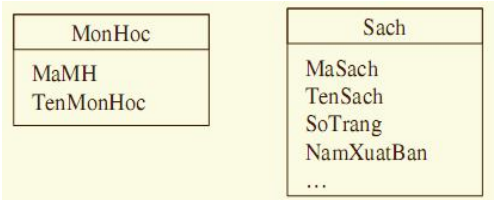
Thuộc tính gồm có các loại:

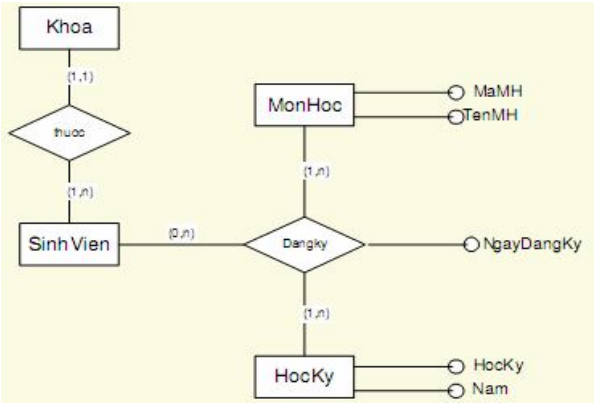
- Thuộc tính đơn trị
- Thuộc tính đa trị
- Thuộc tính kết hợp
- Thuộc tính dẫn xuất

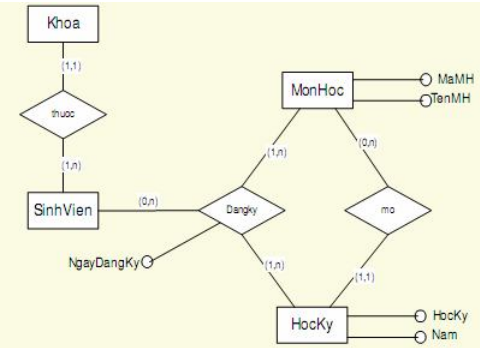


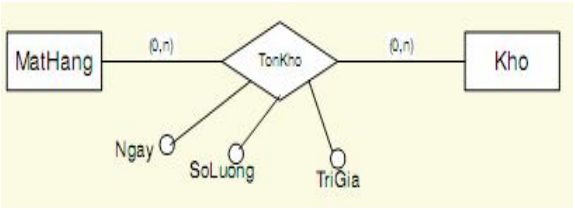




Cách biểu diễn khác





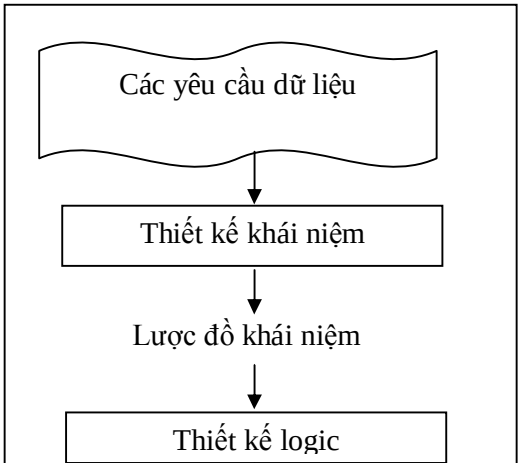

*4.7.3.2. Các bước thiết kế mô hình thực thể kết hợp*

- Nhận dạng các thực thể.
- Nhận dạng các mối quan hệ.
- Găn kết các thuộc tính vào các thực thể và mối kết hợp.
- Xác định các cấu trúc tiêu biểu (có thể có hoặc không).

**4.8. Thiết kế cơ sở dữ liệu**

Thiết kế CSDL đòi hỏi một vài quyết định ở nhiều mức khác nhau. Tính phức tạp của công tác này sẽ được quản lý tốt hơn nếu người ta phân rã bài toán thành các bài toán con và giải các bài toán con một cách độc lập bằng cách dùng phương pháp và kỹ thuật đặc biệt. Thiết kế cơ sở dữ liệu được chia ra các thiết khái niệm, logic và vật lý.

Thiết kế cơ sở dữ liệu được thể hiện như tiếp cận chuyển dữ liệu trong việc phát triển hệ thống thông tin bởi lẽ toàn bộ trọng tâm của quá trình thiết kế là ở dữ liệu và các thuộc tính của nó. Với tiếp cận chuyển dữ liệu, trước hết người ta thiết kế cơ sở dữ liệu, rồi đến các ứng dụng sử dụng cơ sở dữ liệu này. Phương pháp này được phát triển sau những năm 70 cùng với việc đề xuất công nghệ cơ sở dữ liệu.



**Hình 4.9.** Tiếp cận chuyển dữ liệu trong việc thiết kế các hệ thông thông tin

### *4.8.1. Thiết kế khái niệm*

Thiết kế khái niệm bắt đầu từ việc xác định các yêu cầu và kết quả trong lược đồ khái niệm của cơ sở dữ liệu. Lược đồ khái niệm là mô tả mức cao của cấu trúc dữ liệu, độc lập với phần mền quản trị cơ sở dữ liệu cụ thể. Một mô hình khái niệm là ngôn ngữ dùng để mô tả các lược đồ khái niệm. Mục tiêu của thiết kế khái niệm là mô tả nội dung thông tin của cơ sở dữ liệu, chứ không phải là mô tả các cấu trúc lưu trữ do việc quản lý thông tin yêu cầu. Thực tế, thiết kế khái niệm được thực hiện ngay cả khi việc cài đặt, hoàn thiện cuối cùng không sử dụng hệ quản trị cơ sở dữ liệu mà chỉ dùng hệ quản trị tệp và các ngôn ngữ lập trình.

### *4.8.2. Thiết kế logic*

Thiết kế logic bắt đầu từ lược đồ khái niệm và cho ra kết quả là lược đồ logic. Lược đồ logic là mô tả của cấu trúc cơ sở dữ liệu mà hệ quản trị cơ sở dữ liệu xử lý. Một mô hình logic là ngôn ngữ để xác định lược đồ logic; các mô hình logic hay được dùng nhất thuộc về các lớp:

- Mô hình quan hệ,
- Mô hình mạng,
- Mô hình phân cấp,

Mô hình logic phụ thuộc vào lớp của mô hình dữ liệu do hệ quản trị cơ sở dữ liệu dùng, nhưng không phụ thuộc vào một hệ quản trị đặc biệt nào.

Thí dụ: Khi quan tâm đến mô hình quan hệ, người ta tiến hành thiết kế logic theo cùng một cách đối với tất cả các hệ quản trị cơ sở dữ liệu quan hệ bởi vì chúng dùng mô hình quan hệ.

### *4.8.3. Thiết kế vật lý*

Thiết kế vật lý bắt đầu từ lược đồ logic và kết thúc với lược đồ vật lý. Lược đồ vật lý là mô tả cài đặt của cơ sở dữ liệu trên bộ nhớ ngoài: nó mô tả các cấu trúc lưu trữ và các phương pháp truy nhập dùng để truy nhập dữ liệu có hiệu quả. Do đó thiết kế vật lý được sinh ra cho một hệ quản trị cơ sở dữ liệu, và các quyết định trong giai đoạn thiết kế vật lý cũng tác động lại các cấu trúc của lược đồ logic.

Một khi thiết kế cơ sở dữ liệu vật lý đã hoàn tất, các lược đồ vật lý và logic được thể hiện thông qua ngôn ngữ xác định dữ liệu của hệ quản trị cơ sở dữ liệu đích. Các ứng dụng sử dụng cơ sở dữ liệu có thể được hoàn toàn xác định, cài đặt và được thử. Những điều này cho phép một cơ sở dữ liệu dần dần được hình thành.

## KẾT LUẬN

Thiết kế cơ sở dữ liệu là một chủ đề quan trọng trong lĩnh vực cơ sở dữ liệu cả về phương diện lý thuyết lẫn thực hành. Kết quả chính của chuyên đề là: Tìm hiểu và nghiên cứu qua tài liệu để hệ thống các vấn đề sau:

**1/.** Trình bày một số khái niệm cơ bản về CSDL, hệ quản trị CSDL.

**2**/. Khái niệm mô hình quan hệ: thuộc tính, miền dữ liệu, khóa… và các phép toán đại số quan hệ: kết nối, chiếu, chọn…

**3/.** Trình bày một số khái niệm cơ bản khái niệm phụ thuộc hàm trong cơ sở dữ liệu quan hệ.

**4**/. Trình bày các phương pháp chuẩn hóa dữ liệu. Từ đó đưa ra phương pháp thiết kế cơ sở dữ liệu quan hệ.

## TÀI LIỆU THAM KHẢO


[1]. Nguyễn Bá Tường, Lý thuyết cơ sở dữ liệu, HVKTQS, 2000

[2]. Nguyễn Bá Tường, Nhập môn cơ sở dữ liệu phân tán, NXB KHKT, 2004

[3]. Bản dịch của Trần Đức Quang, nguyên lý các hệ cơ sở dữ liệu và cơ sở tri thức, NXB thống kê.

[4]. Nguyễn Bá Tường, Cơ sở dữ liệu lý thuyết và thực hành, NXB khoa học và kỹ thuật - 2001

[5]. Đỗ Trung Tuấn, Lý thuyết cơ sở dữ liệu, NXB khoa học và kỹ thuật - 2000

[6]. Lê tiến Vương, Nhập môn cơ sở dữ liệu quan hệ, NXB thống Kê-2000