



# **Giáo trình Cơ Sở Dữ Liệu**

**Biên tập bởi:**

Vien CNTT – DHQG Hanoi

# **Giáo trình Cơ Sở Dữ Liệu**

**Biên tập bởi:**

Vien CNTT – DHQG Hanoi

**Các tác giả:**

unknown

Phiên bản trực tuyến:

<http://voer.edu.vn/c/09e95dc9>

# MỤC LỤC

1. Chương I. Cơ sở dữ liệu và Mô hình dữ liệu quan hệ
    - 1.1. Hệ thống cơ sở dữ liệu
    - 1.2. Các mô hình dữ liệu
    - 1.3. Mô hình dữ liệu quan hệ - Các khái niệm cơ bản
  2. Chương II. Phụ Thuộc Hàm
    - 2.1. Một số định nghĩa
    - 2.2. Hệ tiên đề cho phụ thuộc hàm
  3. Chương III. Chuẩn Hóa Dữ Liệu
    - 3.1. Các dạng chuẩn hóa dữ liệu
    - 3.2. Các thuật toán phân rã
  4. Chương IV. Sơ đồ thực thể liên kết(ERD)
    - 4.1. Sơ đồ thực thể liên kết(ERD)
  5. Chương V. Ngôn Ngữ SQL(Structured Query Language)
    - 5.1. Giới thiệu
    - 5.2. Các lệnh và các mệnh đề cơ bản trong SQL
    - 5.3. An toàn dữ liệu
  6. Bài Tập Thực Hành
    - 6.1. Bài tập thực hành
- Tham gia đóng góp

# Chương I. Cơ sở dữ liệu và Mô hình dữ liệu quan hệ

## Hệ thống cơ sở dữ liệu

### Định nghĩa CSDL

Cơ sở dữ liệu là một tập hợp các bảng dữ liệu có quan hệ với nhau sao cho cấu trúc của chúng cũng như các mối quan hệ bên trong giữa chúng là tách biệt với chương trình ứng dụng bên ngoài, đồng thời nhiều người dùng khác nhau cũng như nhiều ứng dụng khác nhau có thể cùng khai thác và cùng chia sẻ một cách có chọn lọc lúc cần.

*Ví dụ:*

Hệ thống bán vé máy bay của một hãng hàng không:

Mã chuyến bay	Loại máy bay	Sân bay đi	Sân bay đến	Ngày bay	Giờ bay	Giờ đến
VN272	ART72	SG	HN	30/06/02	7:20	08:05P
VN372	ART72	HN	HUE	30/06/02	8:45	09:40P
VNA32	A320A	HUE	SG	30/06/02	12:05	13:35P
VN472	ART72	HN	SG	30/06/02	12:25	13:25P
VNB77	BOE77	SG	HUE	30/06/02	14:05	15:45P
VNB67	BOE67	SG	DN	30/06/02	14:05	15:45P
VNT06	TU106	DN	HN	30/06/02	16:25	17:35P
...	...	...	...	...	...	...

Để lựa chọn chuyến bay, khách hàng có thể tham khảo lịch bay của hãng, thông tin về chuyến bay được lập theo bảng trên.

Mỗi chuyến bay là một mối quan hệ giữa các thuộc tính: mã chuyến bay, loại máy bay, sân bay đi, sân bay đến, ngày bay, giờ bay, giờ đến.

Bảng trên được gọi là bảng dữ liệu, tùy từng đối tượng mà khai thác dữ liệu trong bảng trên theo mục đích của mình chẳng hạn:

- Mỗi khách hàng muốn bay từ Hà Nội vào Sài Gòn, lúc đó anh ta chỉ quan tâm tới các dòng chứa thông tin về các chuyến bay từ Hà Nội tới Sài Gòn.
- Trong khi đó đối với một nhà quản lý, như tổng giám đốc có thể ông ta chỉ cần biết tới số lượng chuyến bay thực hiện trong ngày bằng cách đếm số dòng trên bảng.

Điều này có nghĩa là các dữ liệu của bảng trên độc lập với các xử lý tác động lên chúng, và được tổ chức thành một bảng gồm các cột và các hàng, mỗi cột được gọi là một thuộc tính (attribute) hay một trường, mỗi hàng (dòng) được gọi là một thể hiện (instance) một bộ (tuple) hay một bản ghi (record) của bảng dữ liệu.

Các thao tác thường áp dụng lên bảng dữ liệu là:

- Thêm một bản ghi
- Xoá, sửa một bản ghi.

## **Mục đích của CSDL**

Tích hợp (intergration)

- Tập trung và quản lý các dữ liệu rời rạc.
- Dữ liệu không tồn tại riêng lẻ mà có sự phụ thuộc lẫn nhau (vd: dữ liệu nhân viên có quan hệ chặt chẽ với dữ liệu của phòng ban và gia đình)
- CSDL không chỉ quản lý dữ liệu mà còn quản lý các quan hệ của dữ liệu

Chia sẻ( sharing)

- Nhiều user cùng sử dụng đồng thời
- Dữ liệu có thể được sử dụng cho nhiều mục đích, nhiều ứng dụng

## **Các thành phần của hệ thống**

### **CSDL hợp nhất**

CSDL phải thoả mãn hai yêu cầu sau:

- Không dư thừa dữ liệu(trên thực tế là dư thừa ít nhất)
- Sử dụng dùng chung

Người sử dụng

- Là người có nhu cầu truy nhập CSDL để thực hiện một thao tác nào đó
- Người sử dụng cuối (End-User): Là những người truy nhập vào CSDL từ một terminal, muốn tìm kiếm tra cứu thông tin.

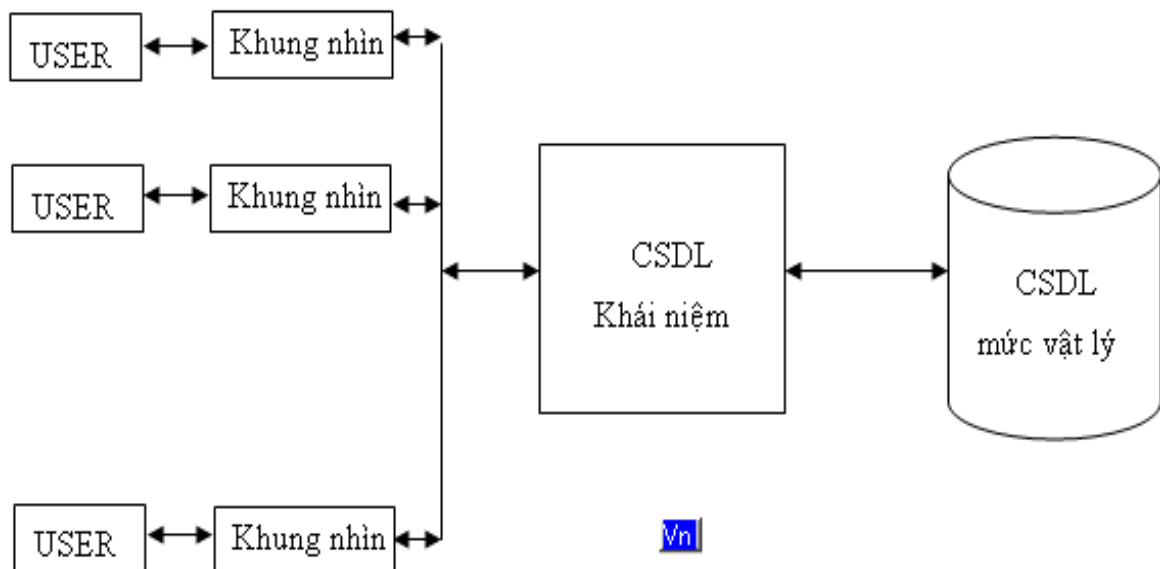
- Người viết chương trình ứng dụng: Những người này ngoài những thao tác trên còn cần đến một ngôn ngữ lập trình (NNLT).
- Người quản trị CSDL: Là người có nhiệm vụ điều khiển toàn bộ hệ CSDL, là người có quyền cao nhất
- Hệ quản trị cơ sở dữ liệu: Đây chính là phần mềm của hệ CSDL

Phần cứng:

Là các thiết bị vật lý được sử dụng để lưu trữ dữ liệu

## Các khái niệm cơ bản của hệ thống CSDL

### *Kiến trúc một hệ cơ sở dữ liệu*



### *Các mức trừu tượng*

#### *Mức vật lý(Physical level)*

- Tập hợp các tập tin, các chỉ mục, những cấu trúc lưu trữ khác dùng để truy xuất
- Tồn tại trong các thiết bị lưu trữ như đĩa từ
- Được quản trị bởi phần mềm quản trị CSDL

#### *Mức khái niệm(Conceptual level)*

- Trừu tượng hóa thế giới thực

- CSDL khái niệm được thiết kế như một thể thống nhất, bao gồm tất cả các dữ liệu được dùng bởi một tổ chức
- DBMS cho phép gộp các tập tin lại và đọc chúng theo phương cách riêng - phương cách được mô tả bằng CSDL khái niệm

### ***Mức khung nhìn(View level)***

- Là cách nhìn, là quan niệm của từng người sử dụng đối với CSDL mức khái niệm. Sự khác nhau giữa khung nhìn và mức khái niệm thực chất là không lớn

### **Tính độc lập dữ liệu, chia sẻ dữ liệu**

Hai loại độc lập dữ liệu trong CSDL:

- Độc lập vật lý: Thay đổi tổ chức của cơ sở dữ liệu vật lý có thể thay đổi hiệu quả tính toán của chương trình, nhưng không đòi hỏi phải thay đổi lại chương trình. Nói một cách khác, mức quan niệm phải đảm bảo được các truy nhập đến CSDL từ phía người sử dụng
- Độc lập logic: sự thay đổi, thêm bớt thông tin về các thực thể ở mức quan niệm không đòi hỏi thay đổi các khung nhìn của NSD dẫn tới không cần thay đổi chương trình ứng dụng

Tính chia sẻ dữ liệu: Vì độc lập với chương trình ứng dụng nên nhiều chương trình ứng dụng cùng sử dụng một cơ sở dữ liệu.

Với hai tính chất trên, ta có thể trừu tượng hoá dữ liệu ở mức cao. Sự trừu tượng hoá không những tăng cường hiệu quả quản lý mà còn giúp tư duy tốt về CSDL.

# Các mô hình dữ liệu

## Khái niệm mô hình dữ liệu

Mô hình dữ liệu là một khuôn dạng của dữ liệu cho phép người dùng nhìn thấy dữ liệu dưới cấu trúc thuật ngữ để diễn tả mà ta gọi là lược đồ(scheme). Nó cho ta biết cấu trúc của cơ sở dữ liệu, bao gồm hai thành phần

- Hệ thống ký hiệu để mô tả dữ liệu
- Tập hợp các phép toán thao tác trên dữ liệu đó

## Mô hình phân cấp (Hierarchical model)

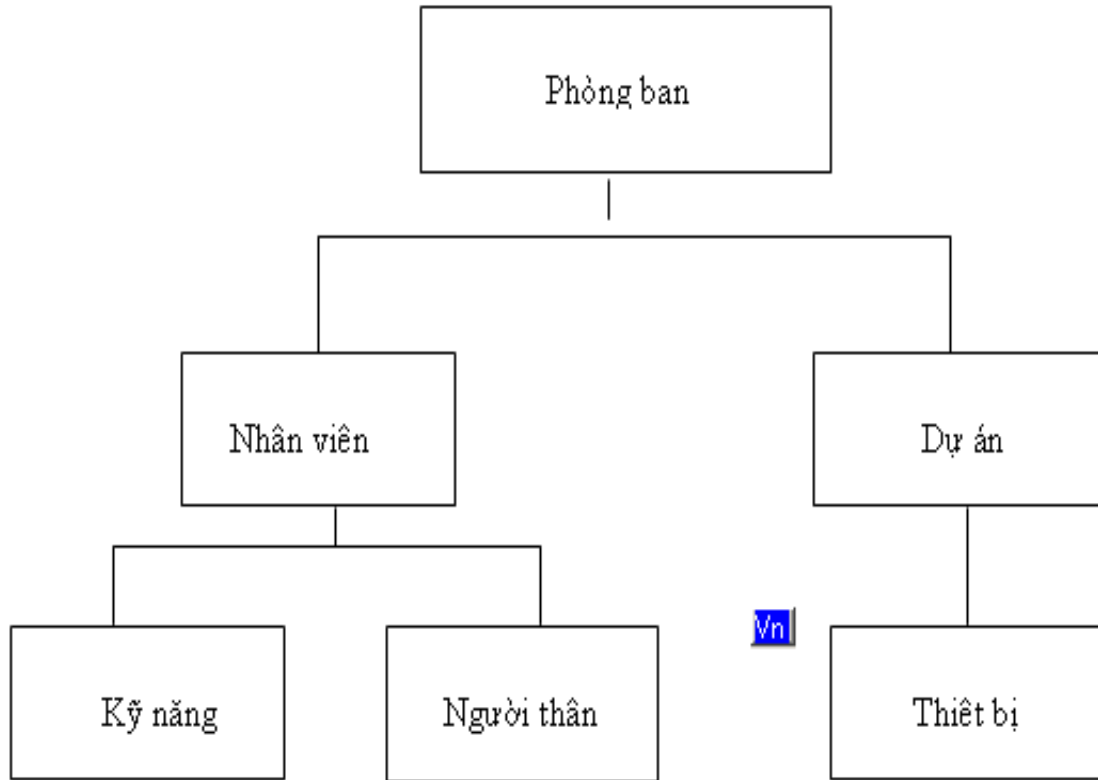
Mô hình phân cấp được đưa ra vào những năm 60, trong mô hình này dữ liệu được tổ chức thành cấu trúc cây, các nút (node) là tập các thực thể, các cạnh là các mối quan hệ giữa hai nút theo mối quan hệ nhất định, cứng nhắc. Hay nói cách khác:

- Là mô hình dữ liệu trong đó các bản ghi được sắp xếp theo cấu trúc top-down(tree).
- Một con chỉ có một cha -> chỉ có một đường truy nhập tới dữ liệu đó trước.
- Tập dữ liệu được tổ chức theo cấu trúc của mô hình dữ liệu phân cấp gọi là CSDL phân cấp

Vi dụ:

Dưới đây là một ví dụ về mô hình phân cấp quản lý nhân sự của một công ty



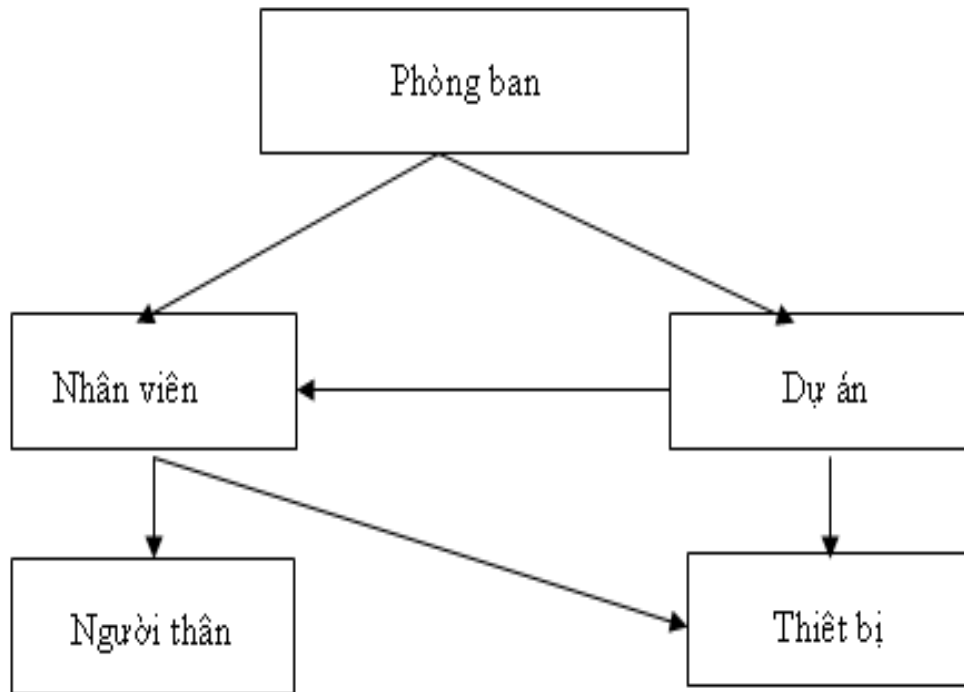


### Mô hình mạng(Network model)

Mô hình mạng được đưa vào cuối những năm 60. Trong mô hình này dữ liệu được tổ chức thành một đồ thị có hướng, trong đó các đỉnh là các thực thể, các cung là quan hệ giữa hai đỉnh, một kiểu bản ghi có thể liên kết với nhiều kiểu bản ghi khác.

Một con có thể có nhiều cha -> có nhiều đường truy nhập đến một dữ liệu cho trước tập dữ liệu được tổ chức theo cấu trúc của mô hình dữ liệu mạng gọi là CSDL mạng

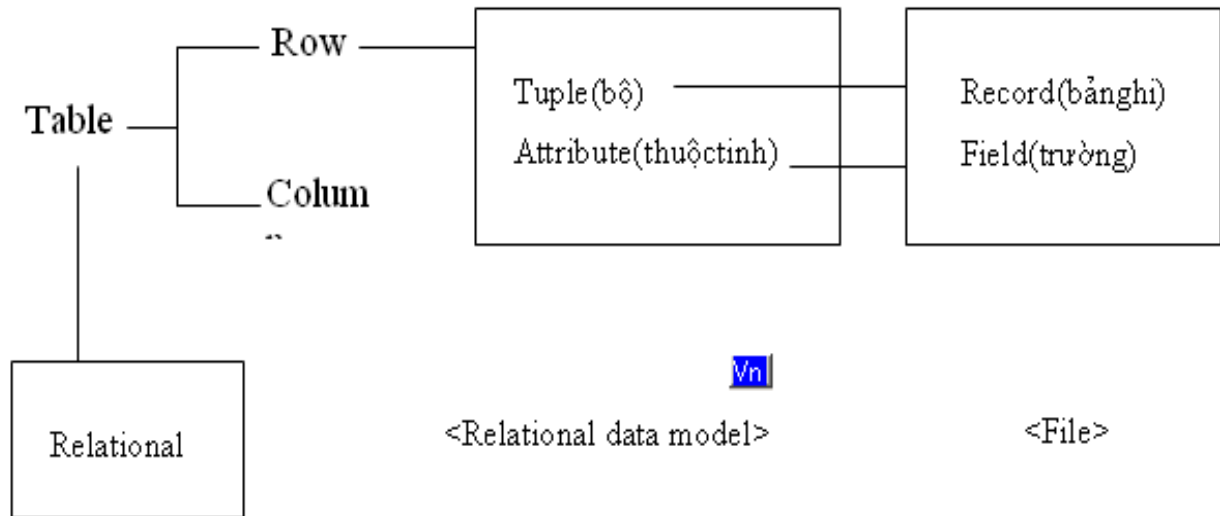
Ví dụ



### **Mô hình dữ liệu quan hệ(Relational model)**

Mô hình này được E.F Codd đưa vào đầu những năm 70, mô hình này dựa trên lý thuyết tập hợp và đại số quan hệ (*chương sau*). Vì tính chất chặt chẽ của toán học về lý thuyết tập hợp nên mô hình này đã mô tả dữ liệu một cách rõ ràng, mềm dẻo và là mô hình thông dụng nhất hiện nay. Hầu hết các hệ QTCSDL đều tổ chức dữ liệu theo mô hình dữ liệu quan hệ. Trong đó dữ liệu được tổ chức dưới dạng bảng các phép toán thao tác trên dữ liệu dựa trên lý thuyết tập hợp của toán học. Tập dữ liệu được tổ chức theo cấu trúc của mô hình dữ liệu quan hệ gọi là CSDL quan hệ.

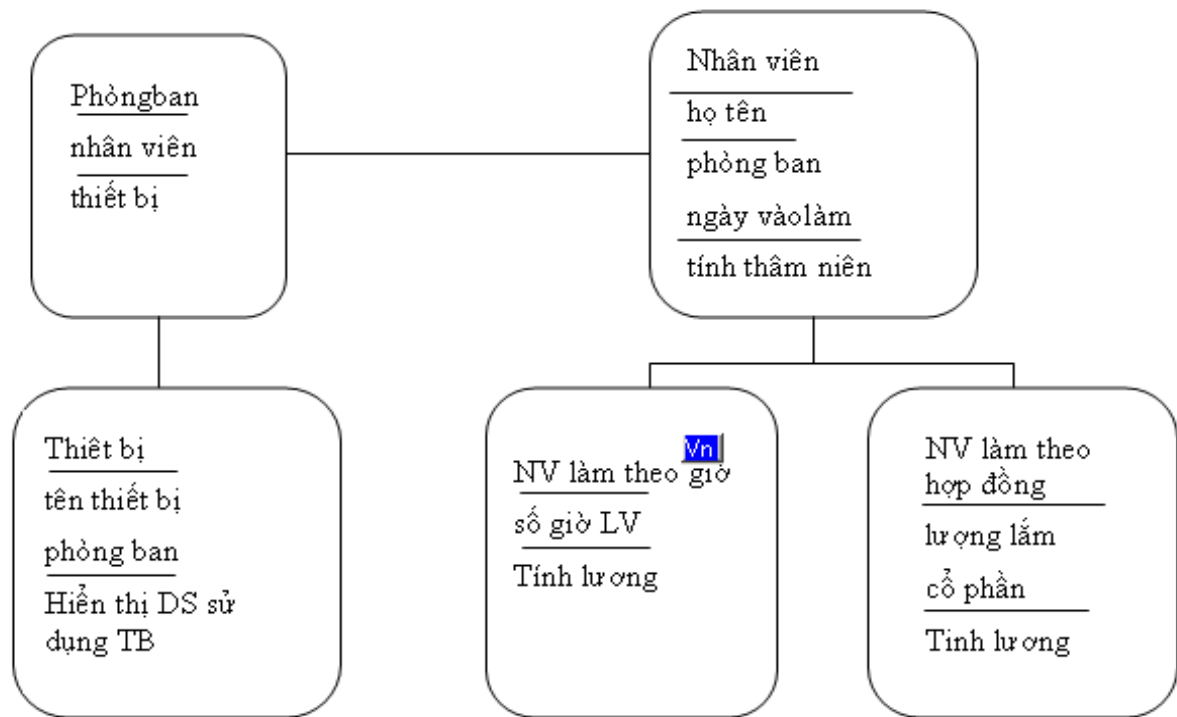
Dữ liệu bảng được thể hiện như sau:



### Mô hình dữ liệu hướng đối tượng (Object Oriented model)

- Là mô hình dữ liệu trong đó các thuộc tính dữ liệu và các phương thức thao tác trên các thuộc tính đó đều được đóng gói trong các cấu trúc gọi là đối tượng.
- Tập dữ liệu được tổ chức theo cấu trúc của mô hình dữ liệu hướng đối tượng gọi là CSDL hướng đối tượng.

*Ví dụ*



Chú ý:

Mô hình dữ liệu hướng đối tượng là mô hình của tương lai đang được phát triển và ngày càng hoàn thiện hơn

# Mô hình dữ liệu quan hệ - Các khái niệm cơ bản

## Các khái niệm và các phép toán về tập hợp

- Tập hợp là khái niệm đầu tiên của toán học, không định nghĩa. Ví dụ: Tập hợp các số nguyên; Tập hợp các sinh viên trong một lớp; Tập hợp các nghiệm của một phương trình..
- Tập hợp được tạo thành từ các phần tử của nó, phần tử cũng là một khái niệm không định nghĩa. Người ta thường dùng các chữ cái viết hoa để kí hiệu cho tập hợp, chữ cái viết thường để chỉ một phần tử của tập hợp.

Các phép toán về tập hợp.

Cho hai tập hợp A và B, khi đó ta có các phép toán trên hai tập hợp như sau:

*Phép giao:* Giao của hai tập hợp A và B là một tập hợp, ký hiệu  $A \cap B$  gồm những phần tử vừa thuộc A vừa thuộc B

$$A \cap B = \{x \mid x \in A \text{ và } x \in B\}$$

*Phép hội:* Hội của hai tập hợp A và B là một tập hợp, ký hiệu  $A \cup B$  gồm những phần tử thuộc A hay thuộc B.

$$A \cup B = \{x \mid x \in A \text{ và } x \in B\}$$

*Phép hiệu:* Hiệu của hai tập hợp A và B là một tập hợp, ký hiệu  $A \setminus B$ , gồm những phần tử thuộc A và không thuộc B

$$A \setminus B = \{x \mid x \in A \text{ và } x \notin B\}$$

*Ví dụ*

$$A = \{1,2,3,4,5\}; B = \{2,4,6,8\}$$

$$A \cup B = \{1,2,3,4,5,6,8\}; A \cap B = \{2,4\}; A \setminus B = \{1,3,5\}$$

*Phép tích đề các:* Tích đề các của hai tập A và B là một tập hợp, ký hiệu  $A \times B$ , được định nghĩa như sau:

$$A \times B = \{(x,y) \mid x \in A \text{ và } y \in B\}$$

Ví dụ:  $A = \{1,2,3\}; B = \{a,b\}$

$A \times B = \{(1,a),(1,b),(2,a),(2,b),(3,a),(3,b)\}$

Ví dụ :  $A = \{1,2,3\}; B = \{a,b\}; C = \{\alpha,\beta\}$  lúc đó:

$A \times B \times C = \{(1,a,\alpha),(1,b,\beta),(2,a,\alpha),(2,b,\beta),(3,a,\alpha),(3,b,\beta)\}$

## Mô hình dữ liệu quan hệ và các khái niệm cơ bản

### Định nghĩa mô hình dữ liệu quan hệ

Mở đầu

Mô hình dữ liệu quan hệ là một mô hình được sử dụng rộng rãi trong đời sống xã hội của mọi tổ chức, cơ quan, xí nghiệp, doanh nghiệp, nơi nào cần quản lý và xử lý thông tin. Ta xét một vài ví dụ minh họa

Ví dụ 1: Xét hồ sơ cán bộ của 1 cơ quan

TT	Mã Số	Tên	Năm Sinh	TĐộ	Quê	GT	Lương
1	01	Huy	1945	Đại Học	T.Bình	Nam	3000000
2	02	Hoàng	1965	Cao Học	H.Nội	Nam	5000000
3	03	Minh	1970	Trung Học	H.Nội	Nữ	2000000

Ví dụ 2: Xét sổ theo dõi khách của một khách sạn

MK	Đến	Đi	MãP	Số người	Tiền
101	1/10/98	5/10/98	301	2	40000
102	5/10/98	20/11/98	302	2	20000

Trong các vấn đề trên tuy quản lý các mảng thông tin khác nhau nhưng cả 2 đều có chung một đặc thù là dữ liệu để mô tả dưới dạng bảng, *mỗi bảng có một dòng đầu tiên gọi là dòng thuộc tính*. Mỗi thuộc tính có một miền giá trị của nó

Ví dụ3 :

- Các thuộc tính là (MK, Đến, Đi.....)
- Miền giá trị của thuộc tính năm sinh là: 1900....1986

- Trong mỗi ví dụ ở trên mỗi bảng đều có một số phần tử như bản hồ sơ nhân sự có 3 phần tử, bảng theo dõi khách sạn có hai phần tử, mỗi 1 phần tử trong bảng gọi là một bộ (*1 bản ghi*).

Các dữ liệu được lưu dưới dạng bảng như vậy được gọi là mô hình CSDL quan hệ.

Sau đây chúng ta sẽ định nghĩa chính xác mô hình cơ sở dữ liệu quan hệ

### **Định nghĩa quan hệ dưới dạng hình thức**

Trong mô hình cơ sở dữ liệu, mỗi quan hệ là một bảng.

### **Định nghĩa quan hệ dưới dạng toán học**

Gọi  $U = \{A_1, A_2, \dots, A_n\}$  là tập hữu hạn các thuộc tính, mỗi thuộc tính  $A_i$  ( $i=1..n$ ) có miền giá trị tương ứng là  $Dom(A_i)$ . Người ta gọi  $r$  là quan hệ trên tập thuộc tính  $U$  nếu  $r$  là tập con của tích Đề-các của  $n$  miền  $Dom(A_i)$ :

$$r \subseteq Dom(A_1) \times Dom(A_2) \times \dots \times Dom(A_n)$$

### **Lược đồ quan hệ**

Một lược đồ quan hệ (relational scheme) là một cặp có thứ tự:

$$R = \langle U, F \rangle$$

Trong đó  $U$  là tập hữu hạn các thuộc tính của quan hệ và  $F$  là tập các ràng buộc của quan hệ (tập phụ thuộc hàm). Ở đây một ràng buộc trên tập các thuộc tính  $\{A_1, A_2, \dots, A_n\}$  được hiểu là một tính chất trên tập tất cả các quan hệ xác định trên tập thuộc tính này.

### **Đại số quan hệ**

Ngôn ngữ đại số quan hệ là cơ sở quan trọng của một ngôn ngữ bậc cao được sử dụng để thao tác trên các quan hệ. Ngôn ngữ này bao gồm hai nhóm phép toán:

- Các phép toán tập hợp (phép giao, phép trừ, phép hợp, và tích Đề-các).
- Các phép toán đặc biệt trên quan hệ (phép chọn, phép chiếu, phép kết nối và phép chia).

*Định nghĩa:* Hai quan hệ gọi là khả hợp nếu chúng có số thuộc tính bằng nhau và thuộc tính thứ  $i$  của quan hệ này có miền giá trị bằng miền giá trị thuộc tính thứ  $i$  của quan hệ kia.

Giả thiết  $r$  là quan hệ xác định trên tập thuộc tính  $U = \{A_1, A_2, \dots, A_N\}$ , với  $r$  là tập hữu hạn các bộ. Khi đó ta có các phép toán trên quan hệ  $r$  như sau:

### Phép hợp:

Hợp của hai quan hệ  $r_1$  và  $r_2$  khả hợp, ký hiệu  $r_1 \cup r_2$  là tập tất cả các bộ thuộc  $r_1$  hoặc  $r_2$  hoặc thuộc cả hai quan hệ  $r_1$  và  $r_2$ .

Biểu diễn phép hợp có dạng:

$$r_1 \cup r_2 = \{t \mid t \in r_1 \text{ hoặc } t \in r_2 \text{ hoặc } t \in r_1 \text{ và } t \in r_2\}$$

*Ví dụ:*

$r_1$	( A B C)	$r_2$	( A B C)	$r_1 \cup r_2$	A	B	C
	$a_1$	$b_1$	$c_1$		$a_1$	$b_1$	$c_1$
	$a_2$	$b_1$	$c_2$		$a_1$	$b_1$	$c_2$
	$a_2$	$b_2$	$c_1$		$a_2$	$b_2$	$c_1$
		$a_2$	$b_2$		$a_2$	$b_2$	$c_2$

### Phép giao:

Giao của hai quan hệ  $r_1$  và  $r_2$  khả hợp, ký hiệu  $r_1 \cap r_2$  là tập tất cả các bộ thuộc cả  $r_1$  và  $r_2$ .

Biểu diễn hình thức:

$$r_1 \cap r_2 = \{t \mid t \in r_1 \text{ và } t \in r_2\}$$

*Ví dụ:* Với  $r_1$  và  $r_2$  là hai quan hệ ở ví dụ trên, khi đó ta có:

$r_1 \cap r_2$	(A	B	C)
	$a_1$	$b_1$	$c_1$

### Phép trừ:

Hiệu của hai quan hệ  $r_1$  và  $r_2$  ký hiệu  $r_1 - r_2$  là tập tất cả các bộ thuộc  $r_1$  không thuộc  $r_2$ .



$$r_1 - r_2 = \begin{pmatrix} A & B & C \\ a_2 & b_1 & c_2 \\ a_2 & b_2 & c_1 \end{pmatrix}$$

### Tích Đề-các:

Gọi  $r$  là quan hệ xác định trên tập thuộc tính  $\{A_1, A_2, \dots, A_n\}$  và  $s$  là quan hệ xác định trên tập thuộc tính  $\{B_1, B_2, \dots, B_m\}$ . Tích Đề-các của  $r$  và  $s$  là tập  $(n + m)$  bộ sao cho  $n$  thành phần đầu có dạng một bộ thuộc  $r$  và  $m$  thành phần sau có dạng của một bộ thuộc  $s$ .

$r \times s = \{t \mid t \text{ có dạng } (a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m)\}.$  trong đó:  $(a_1, a_2, \dots, a_n) \in r$  và  $(b_1, b_2, \dots, b_m) \in s\}$

*Ví dụ:*

$$r \begin{pmatrix} A & B & C \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{pmatrix} \quad s \begin{pmatrix} D & E & F \\ d & e & f \\ d' & e' & f' \end{pmatrix} \quad r \times s = \begin{pmatrix} A & B & C & D & E & F \\ a_1 & b_1 & c_1 & d & e & f \\ a_1 & b_1 & c_1 & d' & e' & f' \\ a_2 & b_2 & c_2 & d & e & f \\ a_2 & b_2 & c_2 & d' & e' & f' \end{pmatrix}$$

### Phép chiếu:

Gọi  $X$  là tập con của tập thuộc tính  $R = (A_1, A_2, \dots, A_n)$ . Phép chiếu trên tập  $X$  của quan hệ  $r$ , ký hiệu là  $\pi_X(r)$  và được định nghĩa như sau:

$$\pi_X(r) = \{t[X] \mid t \in r\}$$

- *Ví dụ:*  $R = \{A, B, C, D\}$ ,  $X = \{A, B\}$ ,  $Y = \{A, C\}$

$r$ ( A B C D)	$\Pi_x(r)=$ A B	$\Pi_x(r)=$ A C
a <sub>1</sub> b <sub>1</sub> c <sub>1</sub> d <sub>1</sub>	a <sub>1</sub> b <sub>1</sub>	a <sub>1</sub> c <sub>1</sub>
a <sub>1</sub> b <sub>1</sub> c <sub>1</sub> d <sub>2</sub>	a <sub>2</sub> a <sub>2</sub>	a <sub>2</sub> c <sub>2</sub>
a <sub>2</sub> b <sub>2</sub> c <sub>2</sub> d <sub>2</sub>		a <sub>2</sub> c <sub>3</sub>
a <sub>2</sub> b <sub>2</sub> c <sub>3</sub> d <sub>3</sub>		

### Phép chọn:

Phép chọn là phép tính để xây dựng một tập con các bộ của quan hệ đã cho thoả mãn biểu thức  $q$  xác định.

Có thể diễn đạt như sau: Cho  $r$  là một quan hệ trên lược đồ quan hệ, một phép chọn trên  $r$  thoả mãn điều kiện  $q$  là một tập hợp được định nghĩa và ký hiệu như sau.

$$\sigma_q(r) = \{t / t \in r \mid q(t) = \text{true}\}$$

Biểu thức  $q$  được diễn tả bằng một tổ hợp Boolean của các toán hạng, mỗi toán hạng là một phép so sánh đơn giản giữa 2 biến là hai thuộc tính hoặc giữa một biến là một thuộc tính và một hằng, cho giá trị đúng sai đối với mỗi bộ đã kiểm tra. Các phép so sánh trong biểu thức  $q$  là  $<$ ,  $=$ ,  $>=$ ,  $<=$ ,  $>$ ,  $\neq$ . Các phép toán logic là  $\neg$  (không),  $\vee$  (hoặc),  $\wedge$  (và).

$$R = \{A, B, C, D\}, q = \{A = a_1\}$$

$r$ ( A B C D)	$\sigma_{A=a_1}(r)=$ A B C B
a <sub>1</sub> b <sub>1</sub> c <sub>1</sub> d <sub>1</sub>	a <sub>1</sub> b <sub>1</sub> c <sub>1</sub> d <sub>1</sub>
a <sub>1</sub> b <sub>1</sub> c <sub>1</sub> d <sub>2</sub>	a <sub>2</sub> a <sub>2</sub> c <sub>1</sub> d <sub>2</sub>
a <sub>2</sub> b <sub>2</sub> c <sub>2</sub> d <sub>2</sub>	
a <sub>2</sub> b <sub>2</sub> c <sub>3</sub> d <sub>3</sub>	

### Phép nối tự nhiên:

Cho hai lược đồ quan hệ  $R_1(U_1)$ ,  $R_2(U_2)$ . Gọi  $S = U_1 \bowtie U_2$  và  $U = U_1 \bowtie U_2$ . Phép kết nối tự nhiên trên hai quan hệ  $r_1$ ,  $r_2$  là một quan hệ  $r$  trên  $U$  được ký hiệu và định nghĩa:

$$r_1 * r_2 = \{t(U) \mid \exists t_1 \in r_1 \text{ và } \exists t_2 \in r_2 \text{ và } t(U_1) = t_1, t(U_2) = t_2\}$$

Ví dụ:

$r$	( A B C )		$r$	( B C D )		$r_1 * r_2 =$	( A B C D )
	a <sub>1</sub> b <sub>1</sub> c <sub>1</sub>			b <sub>1</sub> c <sub>1</sub> d <sub>1</sub>			a <sub>1</sub> b <sub>1</sub> c <sub>1</sub> d <sub>1</sub>
	a <sub>1</sub> b <sub>2</sub> c <sub>1</sub>			b <sub>1</sub> c <sub>2</sub> d <sub>1</sub>			a <sub>1</sub> b <sub>2</sub> c <sub>1</sub> d <sub>1</sub>
	a <sub>2</sub> b <sub>1</sub> c <sub>2</sub>			b <sub>1</sub> c <sub>1</sub> d <sub>1</sub>			a <sub>2</sub> b <sub>1</sub> c <sub>2</sub> d <sub>2</sub>
	a <sub>2</sub> b <sub>3</sub> c <sub>1</sub>			b <sub>1</sub> c <sub>1</sub> d <sub>1</sub>			a <sub>3</sub> b <sub>3</sub> c <sub>1</sub> d <sub>1</sub>
	a <sub>3</sub> b <sub>3</sub> c <sub>2</sub>			b <sub>1</sub> c <sub>1</sub> d <sub>1</sub>			

### Phép trừ

Hiệu của hai quan hệ  $r$  và  $s$  khả hợp, ký hiệu là  $r - s$ , là tập các bộ thuộc  $r$  nhưng không thuộc  $s$ .

$$r - s = \{t \mid t \in r \text{ và } t \notin s\}$$

Ví dụ:

$r$	( A B C )		$s$	( A B C )		$r - s =$	( A B C )
	a <sub>1</sub> b <sub>1</sub> c <sub>1</sub>			a <sub>1</sub> b <sub>1</sub> c <sub>1</sub>			a <sub>2</sub> b <sub>1</sub> c <sub>2</sub>
	a <sub>2</sub> b <sub>1</sub> c <sub>2</sub>			a <sub>1</sub> a <sub>1</sub> a <sub>1</sub>			a <sub>2</sub> b <sub>2</sub> c <sub>1</sub>
	a <sub>2</sub> b <sub>2</sub> c <sub>1</sub>			a <sub>2</sub> b <sub>2</sub> c <sub>2</sub>			

# Chương II. Phụ Thuộc Hàm

## Một số định nghĩa

### Định nghĩa phụ thuộc hàm:

**Định nghĩa:** Cho một tập hữu hạn các thuộc tính  $U = (A_1, A_2, \dots, A_n)$ ,  $r$  là một quan hệ,  $X$  và  $Y$  là hai tập con của  $U$ . Khi đó:  $X \rightarrow Y$  (đọc là  $X$  xác định  $Y$  hay  $Y$  phụ thuộc hàm vào  $X$ ) nếu với mọi bộ  $t_1, t_2 \in r$  mà  $t_1[X] = t_2[X]$  thì  $t_1[Y] = t_2[Y]$ .

Ký hiệu phụ thuộc hàm là FD

Ví dụ:

$r$	(A	B	C	D)
	0	0	0	0
	1	1	1	1
	0	1	1	1

Đối với quan hệ này, ta thấy ngay  $A \rightarrow C$  là một phụ thuộc hàm bởi vì mọi cặp  $t$  và  $t'$  của  $r$ , nếu  $t[X]=t'[X]$  thì  $t[Y]=t'[Y]$ .

### Định lý

Xét quan hệ  $r$  xác định trên tập thuộc tính  $U$  và  $X, Y \subset U$ , lúc đó:

- $X \rightarrow Y$  là phụ thuộc hàm trên  $r$  khi và chỉ khi  $X$  là khoá của quan hệ  $r(XY)$ .
- $X \rightarrow Y$  là phụ thuộc hàm đầy đủ trên  $r$  khi và chỉ khi  $X$  là khoá tối thiểu của quan hệ  $r(XY)$ .

Chú ý:

Cần chú ý rằng chỉ xét các phụ thuộc hàm thoả mãn cho mọi quan hệ trên lược đồ tương ứng của nó. Không thể xem xét một phụ thuộc hàm thoả một quan hệ  $r$  đặc biệt của lược đồ  $R$  rồi sau đó qui nạp rằng phụ thuộc đó thoả trên  $R$ .

# Hệ tiên đề cho phụ thuộc hàm

## Hệ tiên đề Armstrong

Gọi  $F$  là tập tất cả các phụ thuộc hàm đối với lược đồ quan hệ  $r(U)$  và  $X \rightarrow Y$  là một phụ thuộc hàm với  $X, Y \subseteq U$ , ta nói rằng  $X \rightarrow Y$  được *suy diễn logic* từ  $F$  nếu quan hệ trên  $r(U)$  đều thỏa mãn các phụ thuộc hàm của  $F$  thì cũng thỏa  $X \rightarrow Y$ . Sau đây là tập quy tắc của hệ tiên đề được Armstrong đề xuất vào năm 1974, được gọi là *hệ tiên đề Armstrong*.

### Hệ tiên đề Armstrong

Gọi  $R(U)$  là lược đồ quan hệ với  $U = (A_1, A_2, \dots, A_n)$  là tập các thuộc tính: giả sử  $X, Y, Z \subseteq U$ , hệ tiên đề Armstrong bao gồm:

- *Tính phản xạ*: Nếu  $Y \subseteq X$  thì  $X \rightarrow Y$
- *Tính tăng trưởng*: Nếu  $Z \subseteq U, X \rightarrow Y$  thì  $ZX \rightarrow ZY$ . Trong đó  $ZX = Z \cup X$
- *Tính bắc cầu*: Nếu  $X \rightarrow Y$  và  $Y \rightarrow Z$  thì  $X \rightarrow Z$ .

Ví dụ:

Cho  $AB \rightarrow C, C \rightarrow A$ , chứng minh  $BC \rightarrow ABC$

(1)  $C \rightarrow A$  (theo giả thiết)

(2)  $BC \rightarrow AB$  (áp dụng luật tăng trưởng tăng (1) lên B)

(3)  $AB \rightarrow C$  (theo giả thiết)

(4)  $AB \rightarrow ABC$  (tăng (3) AB)

(5)  $BC \rightarrow ABC$  (bắc cầu (1), (2)).

### Bổ đề 1:

Hệ tiên đề Armstrong là đầy đủ, có nghĩa là nếu  $F$  là tập phụ thuộc hàm đúng trên quan hệ  $r$  và  $f: X \rightarrow Y$  là một phụ thuộc hàm được suy diễn từ  $F$  nhờ hệ tiên đề Armstrong thì  $f$  đúng trên  $r$

### Bổ đề 2:

*Tính hợp*: nếu  $X \rightarrow Y$  và  $X \rightarrow Z$  thì  $X \rightarrow YZ$ .

*Tính tỵrabắc cầu* : Nếu  $X \rightarrow Y$  và  $WY \rightarrow Z$  thì  $XW \rightarrow Z$ .

*Tính tách*: Nếu  $X \rightarrow Y$  và  $Z \subseteq Y$  thì  $X \rightarrow Z$ .

### **Bao đóng của tập phụ thuộc hàm F (F<sup>+</sup>)**

Bao đóng của F được kí hiệu là  $F^+$ , là tập tất cả các phụ thuộc hàm được suy diễn logic nhờ *tiên đề Armstron*, 3 *bổ đề trên* từ F, nếu  $F = F^+$  thì F là họ đầy đủ của các phụ thuộc hàm. Bao đóng của tập thuộc tính (X<sup>+</sup>)

$X \rightarrow Y$  là một phụ thuộc hàm suy ra từ F. Khi đó  $X^+$  được gọi là bao đóng của tập thuộc tính X nếu  $X^+$  là tập tất cả các thuộc tính U được suy dẫn bắt đầu từ tập X.

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

Thuật toán tìm bao đóng của tập thuộc tính:

INPUT: X, F, U

OUTPUT: X<sup>+</sup>

S := X

WHILE có (Z → Y) thuộc F với  $Z \subset S$  và  $Y \notin S$

DO S := SY

ENDWHILE

X<sup>+</sup> = S

*Phương pháp*: Tính liên tiếp các tập thuộc tính  $X_0, X_1, X_2, \dots, X_i$

BC0: Đặt  $X_0 = X$

BC1: Nếu tồn tại phụ thuộc hàm  $f: Y \rightarrow Z \in F$  sao cho  $Y \in X_0, Z \in U$  thì  $X_1 = X_0 \cup Z$

BC2: Tương tự.

Until  $X_i = X_{i+1}$ .

Kết luận  $X^+ = X_i$ .

Ví dụ:

Cho lược đồ quan hệ  $R(U)=(ABCDEM)$ , với  $U$  là tập thuộc tính, tập phụ thuộc hàm

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

Tính bao đóng của  $(AB)^+$

BC0: Đặt  $X_0 = AB$

BC1: Tồn tại  $A \rightarrow E$  thuộc  $F$ , mà  $A \in X_0, E \in U$  vậy  $X_1 = X_0 \cup E = ABE$

BC2: Tồn tại  $AB \rightarrow C$  thuộc  $F$ , mà  $AB \in X_1, C \in U$  vậy  $X_2 = X_1 \cup C = ABCE$

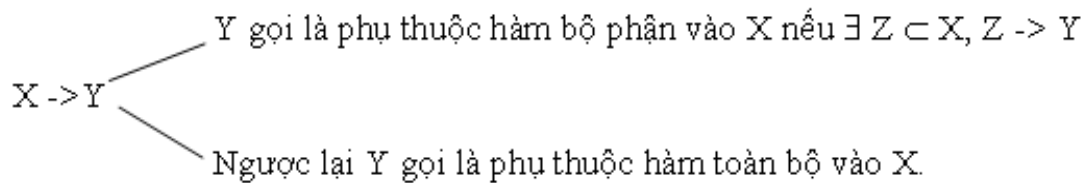
BC3: Tồn tại  $C \rightarrow D$  thuộc  $F$ , mà  $C \in X_2, D \in U$  vậy  $X_3 = X_2 \cup D = ABCDE$

BC4: Tồn tại  $BC \rightarrow EM$  thuộc  $F$ , mà  $BC \in X_3, EM \in U$  vậy  $X_4 = X_3 \cup M = ABCDEM$

BC5: Xét toàn bộ các phụ thuộc hàm thuộc  $F$ , thì  $X_5 = X_4$

Vậy kết luận  $(AB)^+ = X_4 = ABCDEF$ .

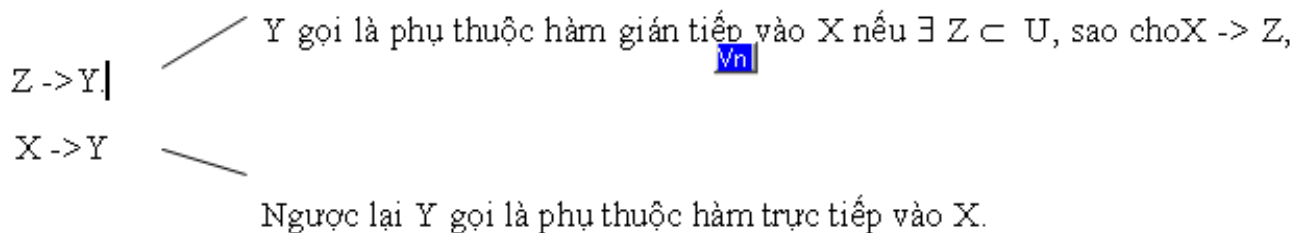
**Định nghĩa: Phụ thuộc hàm bộ phận và phụ thuộc hàm toàn bộ:**



Ví dụ:

$F=\{AB \rightarrow C, C \rightarrow D, A \rightarrow C, BC \rightarrow EM\} \Rightarrow C$  phụ thuộc hàm bộ phận vào  $AB$ .

**Định nghĩa: Phụ thuộc hàm gián tiếp và phụ thuộc hàm trực tiếp:**



Ví dụ:

$F = \{AB \rightarrow C, C \rightarrow D, A \rightarrow C, BC \rightarrow EM, AB \rightarrow D\} \Rightarrow D$  phụ thuộc hàm gián tiếp vào AB.

### Định nghĩa hai tập phụ thuộc hàm tương đương:

Hai tập phụ thuộc hàm F và G được gọi là tương đương với nhau nếu mọi phụ thuộc hàm của G đều suy ra từ F và ngược lại mọi phụ thuộc hàm từ F cũng suy ra được từ G. Hoặc cũng có thể nói G là phủ của F và F cũng là phủ của G.

Chú ý:

Mỗi tập phụ thuộc hàm F tương đương với một tập phụ thuộc G trong đó các vế phải không có quá một thuộc tính

Mỗi tập phụ thuộc hàm F đều tương đương với một tập F' tối thiểu

### Định nghĩa phủ cực tiểu (tối thiểu)

Cho tập thuộc tính U và tập phụ thuộc hàm F. Người ta nói F là tối thiểu khi và chỉ khi:

- Vế phải của mỗi phụ thuộc hàm trong F chỉ có một thuộc tính độc nhất.
- $\nexists X \rightarrow A \in F$  tập  $F - \{X \rightarrow A\}$  tương đương với F (loại bỏ các phụ thuộc dư thừa).
- $\nexists X \rightarrow A \in F, Z \subset X \mid F - \{X \rightarrow A\} \cup \{Z \rightarrow A\}$  tương đương với F (loại bỏ thuộc tính dư thừa về trái)

Thuật toán: tìm phủ tối thiểu của tập phụ thuộc hàm F, với tập thuộc tính U:

INPUT: F, U

OUTPUT: G (G là phủ tối thiểu của F).

BC1:  $G = \emptyset$ . Tách tất cả các phụ thuộc hàm f của F thành phụ thuộc hàm mà vế phải chỉ có một thuộc tính:

FOR  $\forall f \in F, f = X \rightarrow DO G = G \cup \{X \rightarrow A, A \in Y\}$

BC2: Loại bỏ những phụ thuộc hàm không đầy đủ:

WHILE  $\exists Z \subset X, Z \neq X, G \cong G \setminus \{f\} \cup \{Z \rightarrow A\}$



DO  $G = G \setminus \{f\} \cup \{Z - A\}$

BC3: Loại bỏ những phụ thuộc hàm dư thừa:

FOR  $\forall f \in G$  DO

IF  $G \setminus \{f\} \cong$  THEN  $G = G \setminus \{f\}$

BC4: RETURN (G) .

Ta có thể diễn giải lại thuật giải như sau:

BC1: Tách tất cả các phụ thuộc hàm của F thành phụ thuộc hàm mà về phải chỉ có một thuộc

tính, Ví dụ:

$AB \rightarrow CD$  được tách thành  $AB \rightarrow C, AB \rightarrow D$  (luật tách)

BC2: Loại bỏ những phụ thuộc hàm không đầy đủ. Khi loại bỏ, ta phân biệt hai loại phụ thuộc

hàm không đầy đủ sau:

*Loại1:* Phụ thuộc hàm mà về phải là tập con của về trái ( loại  $AB \rightarrow B$ )

*Loại2:* Hai phụ thuộc hàm có về phải giống nhau, nếu về trái của phụ thuộc hàm này chứa về trái của phụ thuộc hàm kia thì ta loại ra khỏi F.

*Ví dụ:* Nếu có  $ABC \rightarrow D$  và  $BC \rightarrow D$  thì ta loại  $ABC \rightarrow D$  khỏi F.

BC3: Loại bỏ những phụ thuộc hàm dư thừa:

Giả sử hai phụ thuộc hàm có về phải giống nhau:  $f1 = X \rightarrow A$  và  $f2 = Y \rightarrow A$ , lúc đó nếu có  $A \in X^+_{F \setminus \{f1\}}$  thì loại  $f1$  ra khỏi F:

*Ví dụ:*

Cho tập phụ thuộc hàm  $F = \{A \rightarrow C, AB \rightarrow C, C \rightarrow DI, CD \rightarrow I, EC \rightarrow AB, EI \rightarrow C\}$ , tìm phủ không dư thừa của F

BC1: Tách các phụ thuộc hàm sao cho về phải chỉ có một thuộc tính duy nhất.

$C \rightarrow D \quad A \rightarrow C$

$C \rightarrow I \quad AB \rightarrow C$

$EC \rightarrow A \quad CD \rightarrow I$

$EC \rightarrow B \quad EI \rightarrow C$

BC2: Loại bỏ những phụ thuộc hàm không đầy đủ

Loại1: Giữ nguyên

Loại2: Loại  $AB \rightarrow C, CD \rightarrow I$

Sau bước 2:  $F = \{C \rightarrow D, C \rightarrow I, EC \rightarrow A, EC \rightarrow B, EI \rightarrow C, A \rightarrow C\}$

BC3: Loại bỏ những phụ thuộc hàm dư thừa

$EI \rightarrow C \quad C \notin (EI)^+(F \setminus EI \rightarrow C)$

$A \rightarrow C \quad C \notin (A)^+(F \setminus A \rightarrow C)$

Do vậy ta giữ cả hai phụ thuộc hàm này. Vậy  $F' = \{C \rightarrow D, C \rightarrow I, EC \rightarrow A, EC \rightarrow B, EI \rightarrow C, A \rightarrow C\}$ .

## **Khoá của quan hệ**

### **Khoá của tập thực thể**

Là một hoặc một tập các thuộc tính xác định duy nhất một thực thể trong một tập thực thể.

### **Khoá của quan hệ**

Tập các thuộc tính  $X$  của một quan hệ  $R$  là một khoá nếu. Không tồn tại 2 bộ khác nhau nhưng tất cả các phần tử của  $X$  đều giống nhau, và không có tập con thực sự nào của  $X$  có tính chất này.

**Định nghĩa:** Cho lược đồ quan hệ  $R(A_1, A_2, \dots, A_n)$  và tập phụ thuộc hàm  $F, X \subseteq A_1, A_2, \dots, A_n$ . Ta nói  $X$  là một khoá của  $R$  khi và chỉ khi  $X \rightarrow A_1, A_2, \dots, A_n \in F^+$  (tất cả các thuộc tính phụ thuộc vào tập thuộc tính  $X$ ),  $\exists Y \subset X \mid X \rightarrow A_1 A_2 \dots A_n \in F^+$ .

Sau đây là một số định nghĩa về khoá của quan hệ:

- Khoá dự tuyển (candidate key): là khoá của quan hệ. Một quan hệ có thể có nhiều khoá dự tuyển.
- Khoá chính (Primary key): là khoá dự tuyển được chọn làm khoá chính của quan hệ. Khoá chính không thể rỗng ( NOT NULL).
- Siêu khoá (Super key): một khoá gọi là siêu khoá nếu như ta bỏ đi một hay nhiều thuộc tính bất kỳ, thì không đảm bảo phần còn lại là khoá.
- Khoá ngoại (Foreign key): (khoá liên kết ) là một hoặc một tập thuộc tính trong quan hệ R1 nhưng là khoá chính trong quan hệ R2.

### Thuộc tính khoá, thuộc tính không khoá

Một thuộc tính trong lược đồ quan hệ R(U) được gọi là thuộc tính khoá nếu nó là một thành phần của một khoá nào đó của R. Ngược lại người ta gọi là thuộc tính không khoá (thuộc tính thứ cấp).

*Ví dụ:* Cho lược đồ quan hệ R = (ABCD) và tập phụ thuộc hàm

$F = \{ A \rightarrow C, AB \rightarrow DC \}$ , khoá là  $\{AB\}$ . Khi đó thuộc tính A, B gọi là thuộc tính khoá, còn thuộc tính D, C gọi là thuộc tính không khoá.

-Thuật toán tìm tất cả các khoá của một lược đồ quan hệ

Bước 1: Tạo tập thuộc tính nguồn TN, tập thuộc tính trung gian TG

Bước 2: if  $TG = \emptyset$  then lược đồ quan hệ chỉ có một khoá K

$K = TN$

Kết thúc

Ngược lại

Qua bước 3

Bước 3: Tìm tất cả các tập con  $X_i$  của tập trung gian TG

Bước 4; Tìm các siêu khoá  $S_i$  bằng cách :

$\forall X_i$

if  $(TN \cup X_i)^+ = Q^+$  then

$$S_i = TN \cup X_i$$

Bước 5: Tìm khoá bằng cách loại bỏ các siêu khoá không tối tiểu

$$\forall S_i, S_j \in S$$

if  $S_i \subset S_j$  then Loại  $S_j$  ra khỏi tập siêu khoá S

S còn lại chính là tập khoá cần tìm.

Ví dụ:

Tìm tất cả các khoá của lược đồ quan hệ sau và tập phụ thuộc hàm như sau:

Áp dụng thuật toán trên ta có lời giải như sau:

$$TN = \{S\}; TG = \{C, Z\}$$

Gọi  $X_i$  là các tập con TG:

$X_i$	$TN \cup X_i$	$TN \cup X_i$	Siêu khoá	khóá
$\emptyset$	S	S		
C	SC	$Q^+$	SC	SC
Z	SZ	$Q^+$	SZ	SZ
CZ	SCZ	$Q^+$	SCZ	

**Kết quả quan hệ trên có hai khoá là :  $\{D, C\}$  và  $\{S, Z\}$**

### Tách một quan hệ

Như ta đã biết mô hình quan hệ do Cood đề suất năm 1970, có những ưu điểm vượt trội so với các mô hình trước đó:

- *Đơn giản*: Các dữ liệu được biểu diễn dưới một dạng duy nhất, là các bảng giá trị, khá tự nhiên và dễ hiểu đối với mọi người sử dụng
- *Chặt chẽ*: Các khái niệm được hình thức cao, cho phép sử dụng các công cụ toán học, có thuật toán

- *Trừu tượng hoá cao*: Mô hình chỉ dừng ở mức quan niệm, nghĩa là độc lập với mức vật lý, với sự cài đặt, với các thiết bị. Nhờ đó làm tăng thêm tính độc lập của dữ liệu và chương trình.
- *Cung cấp các ngôn ngữ truy nhập dữ liệu ở mức cao* ( như SQL...) nhờ đó dễ sử dụng và trở thành chuẩn.

Tuy vậy, khi thiết kế một cơ sở dữ liệu quan hệ thường phải chọn các lược đồ quan hệ. Việc chọn tập các lược đồ này có thể tốt hơn hay xấu hơn tập các lược đồ khác dựa trên một số tiêu chuẩn nào đó. Trọng tâm của việc thiết kế các lược đồ cơ sở dữ liệu là ta tổ chức bao nhiêu lược đồ và mỗi lược đồ có những thuộc tính nào để bảo đảm các tính chất sau:

- *Không trùng lặp dữ liệu*: Trong một quan hệ, giá trị của một thuộc tính nào đó chiếm dụng lượng bộ nhớ lớn không được lặp lại nhiều lần
- *Nhất quán dữ liệu*: Trong một lược đồ quan hệ xác định được nhiều phụ thuộc hàm, tất cả các quan hệ xác định trên lược đồ quan hệ phải thoả các phụ thuộc hàm trên lược đồ ấy.
- *Không gây dị thường khi thêm bộ, xoá bộ*.

Vậy để tạo một cơ sở dữ liệu tốt hơn, nghĩa là không trùng lặp thông tin, nhất quán dữ liệu, ta phải tách một lược đồ quan hệ thành nhiều lược đồ con.

Ví dụ:

Khảo sát về quan hệ cung cấp: cung cấp(tên, địa chỉ, mặt hàng, giá)

*Dư thừa dữ liệu*: Dễ dàng thấy rằng mỗi khi xuất hiện tên nhà cung cấp thì địa chỉ của ông ta lại lặp lại trong quan hệ.

*Không nhất quán dữ liệu*: là hệ quả của việc dư thừa dữ liệu khi sửa đổi địa chỉ của nhà cung cấp ở một bộ nào đó còn các bộ khác vẫn dữ nguyên, khi đó xảy ra một nhà cung cấp lại không có địa chỉ duy nhất.

*Dị thường khi thêm bộ*: một nhà cung cấp khi chưa cung cấp một mặt hàng nào cả, khi đó không thể đưa địa chỉ, tên nhà cung cấp là một bản ghi vào quan hệ vì rằng sẽ phải đưa giá trị vào vị trí của thuộc tính mặt hàng.

*Dị thường khi xoá bộ*: là vấn đề ngược lại của dị thường khi thêm bộ. Không thể xoá tất cả các mặt hàng được cung cấp bởi một nhà cung cấp, vì mặt hàng đó có thể được nhiều người cùng cung cấp.

## **Định nghĩa phép tách lược đồ quan hệ**

Phép tách một lược đồ quan hệ  $U = A_1, A_2, \dots, A_n$  là việc thay thế lược đồ quan hệ  $U$  bằng một tập lược đồ con:  $U_1, U_2, \dots, U_n$  trong đó  $U_i \subseteq U, i = 1..n$ .

$$U = U_1 \cup U_2 \cup \dots \cup U_n \text{ và } U_i \neq U_j \text{ với } i \neq j$$

## **Phép tách bảo toàn thông tin**

Cho lược đồ quan hệ  $R$  và  $F$  là tập phụ thuộc hàm xác định trên  $r$ .

Phép tách lược đồ  $R$  thành các lược đồ con  $R_1, R_2, \dots, R_n$ , dựa trên tập  $F$  gọi là phép tách bảo toàn thông tin nếu: Với mọi quan hệ  $r$  trên  $R$  ta đều có  $r$  là phép kết nối tự nhiên của các phép chiếu của  $r$  lên các  $R_i$ :

$$\forall r(r) \Rightarrow r = \Pi_{R_1}(r) * \Pi_{R_2}(r) * \dots * \Pi_{R_n}(r)$$

## **Thuật toán kiểm tra phép tách bảo toàn thông tin**

INPUT: Lược đồ quan hệ  $R = \{A_1, A_2, \dots, A_n\}$ , tập phụ thuộc hàm  $F$  và phép tách  $\rho = (R_1, \dots, R_k)$

OUTPUT: Kết luận phép tách  $\rho$  có phải là mất mát thông tin không?

Ta có thể diễn giải lại thuật giải như sau:

BC1: Dựng bảng  $S$  gồm  $n$  cột và  $k$  hàng, cột  $j$  ứng với thuộc tính  $A_j$ , hàng  $i$  ứng với lược đồ  $R_i$ .

BC2: Ở vị trí hàng  $i$  và cột  $j$ , đặt  $a_{ij}$  nếu  $A_j$  thuộc  $R_i$ . Nếu không đặt  $b_{ij}$ .

BC3: Xét mỗi phụ thuộc hàm  $X \rightarrow Y$  trong  $F$  cho đến khi bảng  $S$  không còn thay đổi:

Label: Với mỗi phụ thuộc hàm  $X \rightarrow Y$  trong  $F$ , nếu trong bảng  $S$  có chứa 2 dòng  $u, v$  mà:

$$u[X] = v[X] = a_{ij}$$

thì sửa các giá trị tại cột  $Y$  như sau:

+ Nếu  $u[Y] = v[Y] = a_{ij}$  thì không sửa

+ Nếu  $u[Y] = v[Y] = b_{ij}$  thì không sửa



## Phép tách bảo toàn phụ thuộc

### Tính bảo toàn phụ thuộc

- Lược đồ quan hệ R, tập phụ thuộc F,  $\rho = \{R_1, R_2, \dots, R_k\}$ .
- Phụ thuộc hình chiếu: Hình chiếu của F trên lược đồ  $R_i$  là tập các phụ thuộc  $X \rightarrow Y \in F^+$  sao cho  $XY \subseteq Z$ , kí hiệu:  $\Pi_{R_i}(F)$ .
- Phân rã  $\rho$  bảo toàn phụ thuộc hàm F nếu hợp các phụ thuộc hình chiếu  $\Pi_{R_i}(F)$  với  $i = 1 \dots k$  khẳng định logic tất cả các phụ thuộc hàm trong F.

*ví dụ:*

Cho lược đồ quan hệ R(ABC), và  $F = \{A \rightarrow B, A \rightarrow C, B \rightarrow C\}$ , phân rã thành  $R_1(A,B)$  và  $R_2(BC)$ .

Ta có tập phụ thuộc hình chiếu như sau

(1)  $A \rightarrow B$ , khi chiếu trên  $R_1(AB)$

(2)  $B \rightarrow C$ , khi chiếu trên  $R_2(BC)$

Từ (1) và (2)  $\Rightarrow A \rightarrow C$  (tính chất bắc cầu). Vậy phân rã trên là bảo toàn phụ thuộc.

### Kiểm tra tính bảo toàn phụ thuộc.

INPUT: Lược đồ quan hệ  $R = A_1, A_2, \dots, A_n$ , tập phụ thuộc hàm F, phân rã  $\rho = \{R_1, R_2, \dots, R_k\}$

OUTPUT: Khẳng định  $\rho$  có phải là phân rã bảo toàn phụ thuộc.

Phương pháp:

- $\{X \rightarrow Y\} \notin \{\text{phụ thuộc hình chiếu của F lên các lược đồ}\}$
- Cần kiểm tra xem  $\{\text{phụ thuộc hình chiếu của F lên các lược đồ}\} \Rightarrow \{X \rightarrow Y\}$
- Nếu có thì khẳng định  $\rho$  bảo toàn F.

Thuật toán kiểm tra  $X \rightarrow Y$  có được suy dẫn từ tập phụ thuộc hình chiếu:

BEGIN

Z := X;

WHILE vẫn còn những thay đổi với Z DO



FOR I:=1 TO K DO Z:= Z U ((Z ∩ R<sub>i</sub>) bao đóng được lấy ứng với F)  
 END.

*Kết luận:* Nếu Y là tập con của Z thì X → Y được suy ra từ tập phụ thuộc hình chiếu và ρ bảo toàn F.

*Ví dụ:*

Cho R(ABCD) và ρ = {ab, bc, cd}, F = {A → B, B → C, C → D, D → A}

Ta có tập phụ thuộc hình chiếu là: A → B, B → C, C → D, còn D → A Không thuộc tập phụ thuộc hình chiếu. Cần kiểm tra xem D → A có được suy dẫn từ tập phụ thuộc hình chiếu không.

Áp dụng thuật toán :

$$Z = \{D\}$$

Với R<sub>i</sub> = AB;

$$Z = \{D\} \cap ((\{D\} \cap \{AB\})^+ \cap \{AB\}) = \{D\}$$

Tương tự với R<sub>i</sub> = BC cũng không làm thay đổi Z.

Với R<sub>i</sub> = CD:

$$Z = \{D\} \cup ((\{D\} \cap \{CD\})^+ \cap \{CD\})$$

$$= \{D\} \cup (\{D\})^+ \cap \{CD\}$$

$$= \{D\} \cup (\{ABCD\}) \cap \{CD\}$$

$$= \{CD\}$$

Với R<sub>i</sub> = BC áp dụng cho Z = {CD} cho Z = {BCD}

Với R<sub>i</sub> = AB áp dụng cho Z = {BCD} cho Z = {ABCD}

Z không thay đổi nhiều, và Z có chứa A nên ρ bảo toàn F.

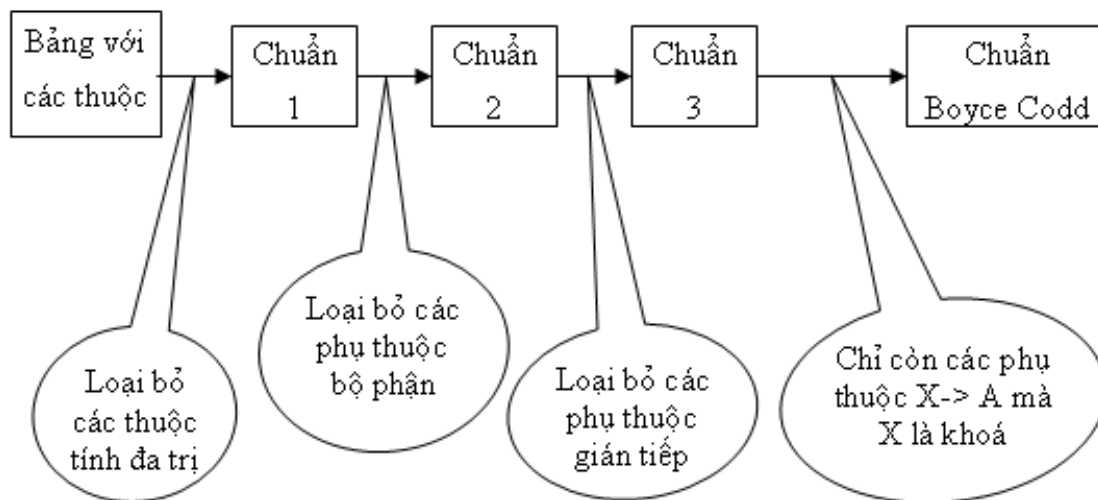
# Chương III. Chuẩn Hóa Dữ Liệu

## Các dạng chuẩn hóa dữ liệu

- Chuẩn hoá là quá trình tách bảng (phân rã) thành các bảng nhỏ hơn dựa vào các phụ thuộc hàm.
- Các dạng chuẩn là các chỉ dẫn để thiết kế các bảng trong CSDL.
- Mục đích của chuẩn hoá là loại bỏ các dư thừa dữ liệu và các lỗi khi thao tác dư thừa và các lỗi khi thao tác dữ liệu (Insert, Delete, Update).
- Nhưng chuẩn hoá làm tăng thời gian truy vấn

Các dạng chuẩn hoá

(Normal Form)



### Dạng chuẩn 1 – 1NF (First Normal Form)

#### Định nghĩa:

Một bảng (quan hệ) được gọi là ở dạng chuẩn 1NF nếu và chỉ nếu toàn bộ các miền giá trị của các cột có mặt trong bảng (quan hệ) đều chỉ chứa các giá trị nguyên tử (nguyên tố)

#### Ví dụ:

Một bảng (quan hệ) chưa ở 1NF:

MASV	HOTEN	DIACHI	MAMON	TENMON	DIEM
A01	Lê Na	12 Thái Hà	M01M02	CSDLAnh	89
A02	Trần An	56 Mã Mây	M01	CSDL	8
A03	Hà Nam	24 Cầu Gỗ	M01M02M03	CSDLAnhToán 1	689

### Cách chuyển sang dạng 1NF:

Điền đủ dữ liệu vào các cột khác. Như sau:

MASV	HOTEN	DIACHI	MAMON	TENMON	TENGV	PHONG	DIEM
A01	Lê Na	12 Thái Hà	M01	CSDL	Mai	P401	8
A01	Lê Na	12 Thái Hà	M02	Anh	Hương	P405	9
A02	Trần An	56 Mã Mây	M01	CSDL	Mai	P401	8
A03	Hà Nam	24 Cầu Gỗ	M01	CSDL	Mai	P401	6
A03	Hà Nam	24 Cầu Gỗ	M02	Anh	Hương	P405	8
A03	Hà Nam	24 Cầu Gỗ	M03	Toán 1	Hoa	P406	9

### Dạng chuẩn 2 – 2NF

#### Định nghĩa

Một quan hệ ở dạng chuẩn 2NF nếu quan hệ đó:

+ Là 1NF

+ Các thuộc tính không khoá phải phụ thuộc hàm đầy đủ vào khoá chính

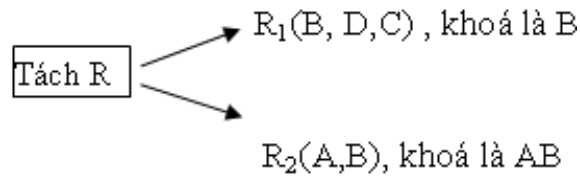
#### Ví dụ

- **Ví dụ1:** Cho quan hệ  $R = (ABCD)$ , khoá là AB và tập phụ thuộc hàm  $F = \{AB \rightarrow C, AB \rightarrow D\}$  là quan hệ đạt chuẩn 2NF.

- **Ví dụ2:** Cho quan hệ  $R = (ABCD)$ , khoá là AB và tập phụ thuộc hàm

$F = \{AB \rightarrow C, AB \rightarrow D, B \rightarrow DC\}$  là quan hệ không đạt chuẩn 2NF vì có phụ thuộc hàm

B → DC là phụ thuộc hàm bộ phận (phụ thuộc hàm không đầy đủ) vào khoá. Khi đó ta đưa về dạng chuẩn 2NF như sau:



### Nhận xét

Một quan hệ ở dạng chuẩn 2NF nếu thoả mãn 1 trong các điều kiện sau:

- Khoá chính chỉ gồm một thuộc tính
- Bảng không có các thuộc tính không khoá
- Tất cả các thuộc tính không khoá phụ thuộc hoàn toàn vào tập các thuộc tính khoá chính

Dạng chuẩn 3 – 3NF

Định nghĩa

Một quan hệ ở dạng chuẩn 3NF nếu quan hệ đó:

+ Là 2NF

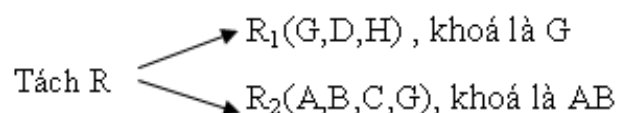
+ Các thuộc tính không khoá phải phụ thuộc trực tiếp vào khoá chính

Ví dụ

- **Ví dụ1:** Cho quan hệ R = (ABCDGH, khoá là AB và tập phụ thuộc hàm F = {AB → C, AB → D, AB → GH} là quan hệ đạt chuẩn 3NF.

- **Ví dụ2:** Cho quan hệ R = (ABCDGH), khoá là AB và tập phụ thuộc hàm

F = {AB → C, AB → D, AB → GH, G → DH} là quan hệ không đạt chuẩn 3NF vì có phụ thuộc hàm G → DH là phụ thuộc hàm gián tiếp vào khoá. Khi đó ta đưa về dạng chuẩn 3NF như sau:



## Dạng chuẩn BCNF (Boyce Codd Normal Form)

Định nghĩa

Một quan hệ ở dạng chuẩn BCNF nếu quan hệ đó:

+ Là 3NF

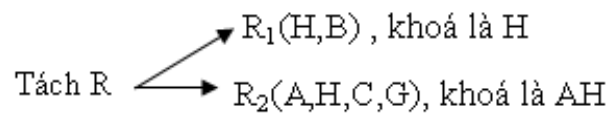
+ Không có thuộc tính khoá mà phụ thuộc hàm vào thuộc tính không khoá.

Ví dụ

- **Ví dụ1:** Cho quan hệ  $R = (ABCDGH)$ , khoá là  $AB$  và tập phụ thuộc hàm  $F = \{AB \rightarrow C, AB \rightarrow D, AB \rightarrow GH\}$  là quan hệ đạt chuẩn BCNF.

- **Ví dụ2:** Cho quan hệ  $R = (ABCDGH)$ , khoá là  $AB$  và tập phụ thuộc hàm

$F = \{AB \rightarrow C, AB \rightarrow D, AB \rightarrow GH, H \rightarrow B\}$  là quan hệ không đạt chuẩn BCNF vì có thuộc tính khoá  $B$  phụ thuộc hàm vào thuộc tính không khoá  $H$ . Khi đó ta đưa về dạng chuẩn BCNF như sau:



# Các thuật toán phân rã

## Phân rã nội không mất thành 3NF

### Thuật toán:

INPUT: Lược đồ quan hệ R, tập phụ thuộc hàm F.

OUTPUT: Một phân rã bảo toàn thông tin của R sao cho mỗi lược đồ quan hệ đều có dạng 3NF.

### Phương pháp:

BC1: Tìm phủ cực tiểu của tập phụ thuộc hàm F.

BC2: Loại bỏ tất cả các thuộc tính của R nếu các thuộc tính đó không liên quan tới một phụ thuộc hàm nào trong F, hoặc về trái hoặc về phải.

BC3: Nếu có phụ thuộc hàm nào của F mà liên quan tới tất cả các thuộc tính của R thì kết quả ra chính là R.

BC4: Ngoài ra, phép tách R đưa ra các lược đồ con gồm các thuộc tính XA cho phụ thuộc hàm  $(X \rightarrow A) \in F$ , tuy nhiên nếu  $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n$  thì thay thế tập thuộc tính  $XA_1A_2 \dots A_n$  cho  $XA_i (i = 1 \dots n)$ .

### Chú ý:

Tại mỗi bước kiểm tra lược đồ R, nếu mỗi thuộc tính không khoá không phụ thuộc gián tiếp vào khoá chính, khi đó R sẽ ở 3NF, ngược lại cần áp dụng BC3.

### Ví dụ:

Cho lược đồ quan hệ R(CTHRSC) với tập phụ thuộc hàm tối thiểu  $F = \{C \rightarrow T, HR \rightarrow C, CS \rightarrow G, HS \rightarrow R, HT \rightarrow R\}$ .

### Áp dụng thuật toán:

BC1: Ta đã có phủ cực tiểu là F (*giả thiết*)

BC2: Giữ nguyên.

BC3: Giữ nguyên.

BC4:  $R_1(CT), R_2(HRC), R_3(CSG), R_4(HSR), R_5(HTR)$

### **Phân rã nối không mất thành BCNF.**

Thuật toán:

INPUT: Lược đồ quan hệ R, tập phụ thuộc hàm F.

OUTPUT: Một phân rã của R có nối không mất, sao cho mỗi lược đồ quan hệ trong phân rã có dạng BCNF.

*Phương pháp:* Phân rã lược đồ R thành hai lược đồ.

Lược đồ 1: Có tập các thuộc tính XA, có dạng BCNF và phụ thuộc  $X \rightarrow A$  đúng.

Lược đồ 2:  $R - A$ .

Lặp lại phân rã trên với  $R - A$  ở vị trí của R cho đến khi không thể phân rã được nữa hoặc lược đồ chỉ còn 2 thuộc tính.

### **Bổ đề:**

Mỗi lược đồ có 2 thuộc tính đều có dạng BCNF.

Nếu R không có dạng BCNF thì ta có thể tìm được các thuộc tính A và B trong R sao cho  $(R - AB) \rightarrow A$  (có thể  $(R - AB) \rightarrow B$  cũng đúng). Điều ngược lại có thể không đúng.

### **Chương trình :**

```
Chương trình chính: BEGIN: Z := R; REPEAT Phân rã Z thành
Z - A và XA /* gọi thủ tục con */ Thêm XA và phân rã; Z :=
Z - A ; UNTIL Không thể phân rã Z Thêm Z vào phân rã. END.
Thủ tục phân rã: BEGIN IF Z không chứa AB sao cho  $A \hat{=} (Z - AB)^+$  THEN RETURN Z có dạng BCNF và không phân rã được ELSE
BEGIN Tìm một cặp A và B; Y:=Z; WHILE Y chứa A và B sao
cho  $(Y - AB) \rightarrow A$  DO Y := Y - B; RETURN phân rã Z-A và Y
/* Y là XA trong CT chính*/ END; END;
```

Ví dụ:

Cho lược đồ quan hệ  $R(U) = CTHRSG$  và tập phụ thuộc hàm  $F = \{C \rightarrow T, HR \rightarrow C, CS \rightarrow G, HS \rightarrow R\}$

Áp dụng thuật toán:

- $Y = CTHRSR, A = C, B = T, C \hat{=} (HRSR)^+ \Rightarrow Y = CHRSG$
- $Y = CHRSG, A = R, B = C, R \hat{=} (HSG)^+ \Rightarrow Y = HRSR$
- $Y = HRSR, A = R, B = G, R \hat{=} (HS)^+ \Rightarrow Y = HRS$
- Không phân rã được nữa, HRS là một lược đồ trong phân rã.
- $Z = CTHRSR - R = CTHSG$
- Tiếp tục với  $Y = CTHSG$
- $Y = CTHSG, A = T, B = H, T \hat{=} (CSG)^+ \Rightarrow Y = CTSG$
- $Y = CTSG, A = T, B = S, T \hat{=} (CS)^+ \Rightarrow Y = CTG$
- $Y = CTG, A = T, B = G, T \hat{=} (C)^+ \Rightarrow Y = CT$
- CT là một lược đồ trong phân rã.
- $Z = CSHG$
- Tiếp tục với  $Y = CSHG$
- ....
- Cuối cùng được phân rã (HRS, CT, CSG, CHS)

## Chuẩn hoá nhờ phép tổng hợp

Trong phần 5.2 chúng ta đã xem xét quá trình chuẩn hoá một lược đồ quan hệ thành 3NF nhờ phép tách không mất mát thông tin đối với tập các phụ thuộc hàm. Trong phần này sẽ trình bày quá trình chuẩn hoá nhờ phép tách tổng hợp. Điều khác nhau cơ bản ở phép tách tổng hợp so với phép tách bảo toàn thông tin là thông tin ban đầu gồm tập các thuộc tính và tập các phụ thuộc hàm, còn trong phương pháp tách bảo toàn thông tin, thông tin ban đầu là một lược đồ rất cụ thể. Qua phép tổng hợp kết quả cũng cho một tập các lược đồ ở dạng chuẩn 3NF. Để là rõ chúng ta sẽ nói đến một số khai niệm.

*Phụ thuộc hàm dư thừa:* Cho lược đồ quan hệ R và tập phụ thuộc hàm F. Một phụ thuộc hàm

$f: X \rightarrow Y \in F$  gọi là dư thừa nếu  $F^+ = (F - \{X \rightarrow Y\})^+$

*Phủ không dư thừa:* Cho lược đồ quan hệ R và tập phụ thuộc hàm F. G là phủ không dư thừa

của F nếu  $G^+ = F^+$  và G không chứa phụ thuộc hàm nào dư thừa.

*Thuộc tính dư thừa:* Một thuộc tính thuộc về trái một phụ thuộc hàm được gọi là dư thừa nếu loại bỏ nó không làm thay đổi bao đóng của tập các phụ thuộc hàm, tức là:

$f: X \rightarrow Y \in F, A \in X, A$  là dư thừa nếu  $\{X - A\} \rightarrow Y \in F$



Thuật toán

INPUT: Tập thuộc tính  $U = \{A_1, A_2, \dots, A_n\}$ , tập phụ thuộc hàm  $F$ .

OUTPUT: Tập các lược đồ quan hệ ở 3NF.

Phương pháp

BC1: Tìm phủ không dư thừa  $G$  của  $F$  (loại bỏ các phụ thuộc hàm không dư thừa của  $F$ )

BC2: Phân chia tập phụ thuộc hàm  $G$  thành các nhóm sao cho các phụ thuộc hàm trong một nhóm có cùng vế phải.

BC3: Mỗi cặp nhóm, ví dụ  $G_1, G_2$  có vế trái là  $X$  và  $Y$  mà tồn tại song ánh  $(X \leftrightarrow Y) \in G^+$  (tức là  $X \rightarrow Y$  và  $Y \rightarrow X$ ) thì hoà hai nhóm lại với nhau. Với  $A \in Y$  (nếu  $X - A \in H$ ) thì loại nó khỏi  $H$ . Tương tự đối với các phụ thuộc hàm còn lại.

BC4: Ở mỗi nhóm đạt được cấu trúc thoả mãn ba bước trên thì ta gộp lại là một lược đồ quan hệ, trong đó mỗi tập thuộc tính xuất hiện ở vế trái của phụ thuộc hàm thì là một khoá của lược đồ quan hệ. Các khoá tìm được gọi là khoá được tổng hợp.

*Ví dụ:*

Cho tập các thuộc tính  $(A, B_1, B_2, C_1, C_2, D, E, I_1, I_2, I_3, J)$ , tập các phụ thuộc hàm

$F = \{$

$A \rightarrow B_1 B_2 C_1 C_2 D E I_1 I_2 I_3 J$

$B_1 B_2 C_1 \rightarrow A C_2 D E I_1 I_2 I_3 J$

$B_1 B_2 C_2 \rightarrow A C_1 D E I_1 I_2 I_3 J$

$E \rightarrow I_1 I_2 I_3, C_1 D \rightarrow J, C_2 D \rightarrow J$

$I_1 I_2 \rightarrow I_3, I_2 I_3 \rightarrow I_1, I_1 I_3 \rightarrow I_2$

BC1: Rõ ràng  $F$  là tối thiểu nhưng dư thừa. Áp dụng thuật toán loại bỏ các thuộc tính dư thừa ta thu được tập phụ thuộc hàm  $G$  là phủ không dư thừa của

$F, G = \{$

$A \rightarrow B_1 B_2 C_1 C_2 D E, C_1 D \rightarrow J$

$B_1B_2C_1 \rightarrow A \quad C_2D \rightarrow J$

$B_1B_2C_2 \rightarrow A \quad I_1I_2 \rightarrow I_3$

$E \rightarrow I_1I_2 \quad I_2I_3 \rightarrow I_1$

$I_1I_3 \rightarrow I_2 \}$

BC2 + BC3: Sau khi hoà các nhóm phụ thuộc hàm ta được

$H = \{ (A, B_1B_2C_1, B_1B_2C_2) \rightarrow DE \quad (E) \rightarrow I_1I_2 \quad C_1D \rightarrow J \quad C_2D \rightarrow J \quad (I_1I_2, I_2I_3, I_1I_3) \}$

BC4: Thiết lập các lược đồ

$R_1 = AB_1B_2C_1C_2DE$  với khoá  $K_1 = \{A, B_1B_2C_1, B_1B_2C_2\}$

$R_2 = EI_1I_2$  với khoá  $K_2 = \{E\}$

$R_3 = C_1DJ$  với khoá  $K_2 = \{C_1D\}$

$R_4 = C_2DJ$  với khoá  $K_2 = \{C_2D\}$

$R_5 = I_1I_2I_3$  với khoá  $K_2 = \{I_1I_2, I_2I_3, I_1I_3\}$

# Chương IV. Sơ đồ thực thể liên kết(ERD)

## Sơ đồ thực thể liên kết(ERD)

### Thực thể

Thực thể (Entity) là đối tượng mà ta cần quan tâm trong công tác quản lý. Một đối tượng có thể là:

Rất cụ thể:

- Nhân viên của một cơ quan là đối tượng cần quản lý.
- Tờ hoá đơn là một đối cần quản lý.
- ...

Hoặc trừu tượng, chẳng hạn:

- Khoa tin học
- Ngành toán ứng dụng.
- ...

a) Tiêu chuẩn xác định thực thể

- Có ích cho quản lý.
- Phân biệt được giữa các thực thể với nhau.

b) Kiểu thực thể:

Là tập hợp các đối tượng cùng loại hình thành một kiểu thực thể, nói khác đi kiểu thực thể chính là những thực thể cùng được mô tả bằng những đặc trưng giống nhau.

*Ví dụ:* Một nhân viên là một thực thể, tập hợp các nhân viên của cùng một hệ thống tạo thành một kiểu thực thể.

c) *Thuộc tính của thực thể:* Thuộc tính được hiểu là dữ liệu, là tính chất để dùng để mô tả một đặc trưng của thực thể.

*Ví dụ:* Thực thể Sinh Viên có các thuộc tính: Họ Tên, Que Quan, MaSV...

d) *Thuộc tính của liên kết:* Là các tính chất chung của các quan hệ trong một tập thực thể. thuộc tính của quan hệ phụ thuộc vào tất cả các tập thực thể tham gia vào quan hệ.

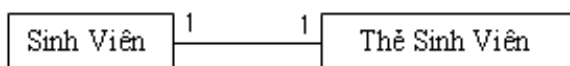
e) *Chú ý*: Chú ý thực thể có thể là đa trị, đơn trị hoặc phức hợp

- *thuộc tính phức hợp*: giá trị của thuộc tính có thể chia nhỏ thành các phần có nghĩa . VD: thuộc tính họ và tên, thuộc tính địa chỉ...
- *thuộc tính đa trị*: giá trị của thuộc tính là những thành phần thuộc cùng một loại. VD: thuộc tính ngoại ngữ

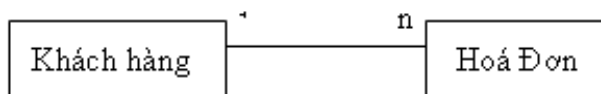
## Liên kết

Một liên kết là một sự ghép nối giữa hai hay nhiều thực thể. Phản ánh sự liên hệ giữa các thực thể. Có các kiểu liên kết sau.

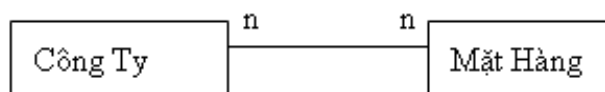
*Quan hệ 1-1* (đọc là liên kết một một): Hai thực thể A và B có mối quan hệ 1-1 nếu một thực thể kiểu A tương ứng với một thực thể kiểu B và ngược lại:



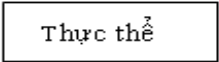
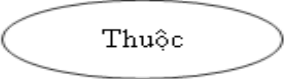

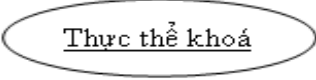
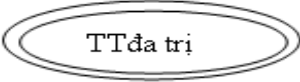
*Quan hệ 1-n* (đọc là liên kết một nhiều): Hai thực thể A và B có mối Quan hệ 1- n nếu một thực thể kiểu A tương ứng với nhiều thực thể kiểu B và một thực thể của B chỉ tương ứng với một thực thể kiểu A:



*Quan hệ n-n* (đọc là liên kết nhiều nhiều): Hai thực thể A và B có mối Quan hệ n-n nếu một thực thể kiểu A tương ứng với nhiều thực thể kiểu B và ngược lại”



## Hệ thống kí hiệu

 Thực thể	Thực thể được biểu diễn thành hình chữ nhật trong đó có ghi tên thực thể
 Thuộc	Thuộc tính được biểu diễn thành hình Elip trong đó có ghi tên thuộc tính
 Quan hệ	Quan hệ
 Thực thể khoá	Khoá của thuộc tính
 TT đa trị	TT đa trị

## Chuyển mô hình thực thể liên kết thành các lược đồ quan hệ

Tập thực thể:

- Tên của tập thực thể -> tên của lược đồ quan hệ.
- Các thuộc tính của tập thực thể -> thuộc tính của lược đồ quan hệ.

Tập quan hệ:

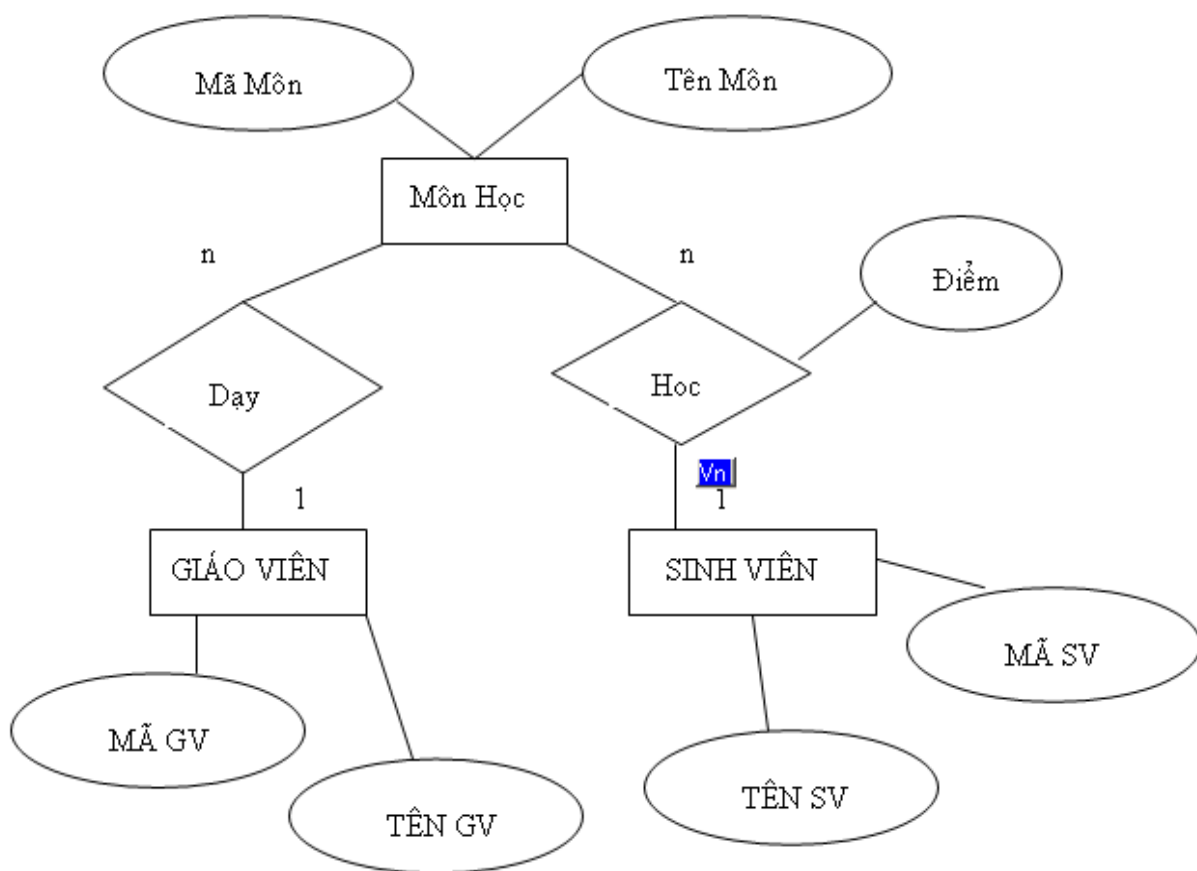
- Tên của tập quan hệ -> tên của lược đồ quan hệ
- Khoá của các tập thực thể tham ra vào quan hệ và các thuộc tính của riêng quan hệ -> thuộc tính của lược đồ quan hệ.

Quan hệ có khoá chung:

- Khi hai quan hệ có chung khoá dự tuyển -> thay hai lược đồ với các thuộc tính kết hợp.

Ví dụ:

Chuyển mô hình ER sau thành lược đồ quan hệ.



Chuyển thành lược đồ quan hệ như sau:

MÔN HỌC (# Mã môn, Tên Môn)

GIAOVIÊN (# Mã GV, Họ Tên)

SV (#Mã SV, Tên SV)

DẠY (Mã GV, # Mã môn )

HỌC (#Mã SV, # Mã môn, Điểm)

Lược đồ DẠY và MÔN HỌC có cùng khoá => kết hợp thành lược đồ

MÔN HỌC ((# Mã môn, Tên Môn, Mã GV) .

## **Các bước thiết kế CSDL**

Giai đoạn tìm hiểu nhu cầu là nhằm làm rõ hệ thống CSDL sẽ được lập ra phải đáp ứng các nhu cầu gì của người dùng-các nhu cầu trước mắt và tương lai, nhu cầu tường minh và tiềm ẩn.

Giai đoạn phân tích là nhằm đi sâu vào bản chất và chi tiết của hệ thống cho thấy hệ thống là phải thực hiện những việc gì và các dữ liệu mà nó đề cập là những dữ liệu nào, có cấu trúc ra sao.

Giai đoạn thiết kế là nhằm đưa ra các quyết định về cài đặt hệ thống, để sao cho hệ thống thoả mãn được các yêu cầu mà giai đoạn phân tích đã đưa ra, đồng thời lại thích ứng với các điều kiện ràng buộc trong thực tế.

Giai đoạn cài đặt bao gồm hai công việc chính, là lập trình và kiểm định, nói theo thuật ngữ của ngành xây dựng, thì đó là giai đoạn thi công nhằm chuyển các kết quả phân tích và thiết kế trên giấy, thành một hệ thống chạy được.

Giai đoạn khai thác và bảo dưỡng là giai đoạn đưa hệ thống vào sử dụng, đồng thời thực hiện các chỉnh sửa, khi phát hiện thấy hệ thống còn có chỗ chưa thích hợp.

# Chương V. Ngôn Ngữ SQL(Structured Query Language)

## Giới thiệu

SQL là ngôn ngữ được phát triển từ ngôn ngữ SEQUEL – 2, được thử nghiệm và cài đặt tại trung tâm nghiên cứu của hãng IBM ở San Jose, California cho các hệ thống Quản trị cơ sở dữ liệu lớn.

SQL là ngôn ngữ vấn tin (ngôn ngữ hỏi), mỗi câu hỏi được thể hiện bằng một câu lệnh. Phép toán cơ bản trong SQL được miêu tả như là một khối SELECT...FROM...WHERE.

Các thuật ngữ trong cơ sở dữ liệu quan hệ như bảng, thuộc tính, bộ,... được thay thế bằng các thuật ngữ như bảng(table), cột(column), bản ghi(record) hoặc hàng(row) để phù hợp với ý nghĩa của hệ phần mềm này.



# Các lệnh và các mệnh đề cơ bản trong SQL

## Lệnh CREATE TABLE

- Cú pháp: CREATE TABLE Tên\_bảng

Tên\_cột Loại\_dữ\_liệu [Not Null]),

Primary Key( Tên khoá chính ),

Foreign Key( Tên khoá ngoài),...);

Trong đó:

+ *Tên\_bảng*: là chuỗi ký tự không chứa các ký tự trống và không trùng với các từ khoá

+ *Tên\_cột*: là chuỗi ký tự bất kỳ không chứa ký tự trống, tên cột trong một bảng là duy nhất, thứ tự các cột không quan trọng

+ *Loại\_dữ\_liệu*: gồm một số loại dữ liệu sau:

- *integer*: số nguyên từ -2147483648 đến 2147483647
- *smallinteger*: số nguyên từ -32768 đến 32767
- *decimal(n, p)*: số thập phân với độ dài tối đa là n kể cả p chữ số phần thập phân (không tính dấu chấm thập phân). Từ khoá Number trong SQL được dùng trong dạng dữ liệu này.
- *Float*: số dấu phẩy động
- *Char(n)*: chuỗi ký tự có độ dài cố định n, (n<=255)
- *Varchar(n)*: chuỗi ký tự có độ dài biến đổi, độ dài chuỗi có thể từ 0 đến n và được xác định tại thời điểm đưa dữ liệu vào lưu trữ.
- *Longvarchar*: chuỗi ký tự có độ dài không cố định. Độ dài này có thể từ 4Kbs đến 32 Kbs
- *Date*: dữ liệu dạng ngày tháng

- Ví dụ:

+ Tạo bảng sinh viên như sau:

```
CREATE TABLE sinhvien
```

```
(Hodem Varchar(20) Not Null,
```

Ten Varchar(15) Not Null,  
Nsinh Date,  
MaSV Varchar(5) Not Null,  
Que Varchar(25),  
Hocluc Decimal(4,2),  
PRIMARY KEY (MaSV));

+ Tạo bảng đề tài như sau:

```
CREATE TABLE detai  
(MaDT Varchar(10) Not Null,  
TenDT Varchar(30) Not Null,  
ChuNhiem Varchar(25),  
Kinhphi Decimal(10,2),  
KetQua Decimal(4,2),  
PRIMARY KEY (MaDT));
```

+ Tạo bảng sinh viên đề tài như sau:

```
CREATE TABLE sv_detai  
(MaSV Varchar(5) Not Null,  
MaDT Varchar(10) Not Null,  
NoiTT Varchar(30) Not Null,  
KM Decimal(10,2),  
PRIMARY KEY (MaDT, MaSV),  
FOREIGN KEY (MaDT) REFERENCES detai(MaDT),
```

FOREIGN KEY (MaSV) REFERENCES sinhvien(MaSV));

## **Lệnh INSERT INTO**

- Cú pháp:

```
INSERT INTO Tên_bảng
```

```
VALUES(gia trị1, giá trị 2, ..., giá trị n);
```

trong đó: giá trị1, giá trị 2, ..., giá trị n là những giá trị để chèn vào các cột tương ứng từ cột 1 đến cột n của một bản ghi trong bảng

- Chức năng:

Chèn một bản ghi vào bảng

- Ví dụ:

Chèn vào bảng sinh viên một bản ghi như sau:

```
INSERT INTO sinhvien
```

```
VALUES("Nguyễn Hồng", "Son", "2/12/1976", "A420", "Hà Nội", 9.5);
```

## **Lệnh DELETE**

- Cú pháp:

```
DELETE FROM Tên_bảng
```

```
WHERE <Điều kiện xoá >;
```

- Chức năng:

Xoá các bản ghi trong bảng thoả mãn điều kiện xoá

- Ví dụ:

Xoá những sinh viên có học lực <5 trong bảng sinh viên

```
DELETE FROM sinhvien
```

```
WHERE (HocLuc<5);
```

## **Lệnh UPDATE**

- *Cú pháp:*

UPDATE Tên\_bảng

SET Giá\_trị\_mới

WHERE <Điều kiện sửa đổi>;

- *Chức năng:*

Sửa đổi giá trị các trường của các bản ghi trong bảng

- Ví dụ:

Sửa học lực của sinh viên có MaSV = “B401” lên 8

UPDATE sinhvien

SET HocLuc = 8

WHERE MaSV = “B401”;

## **Lệnh SELECT**

SELECT là một lệnh hỏi dữ liệu cơ bản trong SQL. Có rất nhiều mệnh đề con tùy chọn trong câu lệnh SELECT, vì vậy chúng ta sẽ làm quen lần lượt từng bước một.

### **Cấu trúc SELECT – FROM – WHERE**

Cấu trúc SELECT – FROM – WHERE là cấu trúc đơn giản nhất của SQL.

- *Cú pháp:*

SELECT <Danh sách các cột>

FROM <Danh sách bảng>

WHERE <Điều kiện>;

Trong đó:

+ <Danh sách các cột>: Là danh sách các cột hoặc biểu thức của các cột được đưa vào kết quả truy vấn

+ <Danh sách bảng>: Là danh sách các bảng mà từ đó các cột được lấy ra

+ <Điều kiện>: Là một biểu thức logic xác định các bản ghi thoả mãn điều kiện của câu lệnh.

- Ví dụ 1: Hiển thị họ đệm, tên của bảng sinh viên:

```
SELECT Hodem, Ten
```

```
FROM sinhvien;
```

- Ví dụ 2: Hiển thị họ đệm, tên, học lực của những sinh viên có học lực  $\geq 8$  trong bảng sinh viên:

```
SELECT Hodem, Ten, HocLuc
```

```
FROM sinhvien
```

```
WHERE HocLuc  $\geq$  8;
```

## **Truy vấn sử dụng các hàm MAX, MIN, AVG, SUM, COUNT**

### ***Hàm MAX***

- Chức năng:

Cho giá trị lớn nhất trong cột

- Ví dụ:

Hiển thị học lực cao nhất trong danh sách sinhvien

```
SELECT Max(HocLuc) AS DiemCaoNhat
```

```
FROM sinhvien;
```

( AS để đặt tên cho cột Max(HocLuc))

### ***Hàm MIN***

- Chức năng:

Cho giá trị nhỏ nhất trong cột

- Ví dụ:

Hiển thị học lực nhỏ nhất trong danh sách sinh viên

```
SELECT Min(HocLuc) AS DiemThapNhat
```

```
FROM sinhvien;
```

### ***Hàm AVG***

- Chức năng: Cho giá trị trung bình cộng trong cột

- Ví dụ: Hiển thị học lực trung bình của cột học lực trong danh sách sinh viên

```
SELECT AVG(HocLuc) AS DiemTB
```

```
FROM sinhvien;
```

### ***Hàm SUM***

- Chức năng: Cho tổng giá trị trong cột

- Ví dụ: Hiển thị tổng học lực của cột học lực trong danh sách sinh viên

```
SELECT SUM(HocLuc) TongHL
```

```
FROM sinhvien;
```

### ***Hàm COUNT***

- Chức năng: Cho biết số phần tử ( hàng) trong cột

- Ví dụ: Đếm số bản ghi (hàng) của cột học lực trong danh sách sinh viên

```
SELECT COUNT(HocLuc) AS SoSinhVien
```

```
FROM sinhvien;
```

**Truy vấn sử dụng các phép AND, OR, IN, BETWEEN, NOT, ALL**

+ AND: Phép và logic

+ OR: Phép hoặc logic

+ IN: Là phần tử của...

+ BETWEEN: Là phần tử giữa các phần tử ...

+ NOT: Phép phủ định

+ ALL: Là tất cả những phần tử ...

- Ví dụ 1: Đưa ra danh sách những sinh viên có điểm  $\geq 9$  và có quê = “Hà Nội”

```
SELECT *
```

```
FROM sinhvien
```

```
WHERE (diem $\geq$ 9) AND ( que = “Hà Nội”);
```

- Ví dụ 2: Đưa ra danh sách những sinh viên có quê = “Thái Bình” hoặc “Thái Nguyên”

```
SELECT *
```

```
FROM sinhvien
```

```
WHERE (que = “Thái Bình”) OR (que = “Thái Nguyên”);
```

- Ví dụ 3: Đưa ra danh sách những nhân viên có kết quả là 8,9,10

```
SELECT *
```

```
FROM sv_dtai
```

```
WHERE KetQua IN (8, 9, 10);
```

- Ví dụ 4: Đưa ra danh sách những sinh viên có kết quả nằm trong khoảng [8,10]

```
SELECT *
```

```
FROM sv_dtai
```

```
WHERE KetQua BETWEEN 8 and 10;
```

- Ví dụ 5: Đưa ra danh sách những sinh viên có quê không phải là “Hà Nội”

```
SELECT *  
FROM sinhvien  
WHERE que NOT(SELECT que  
FROM sinhvien  
WHERE (que = "Hà Nội"));
```

- Ví dụ 6: Hiển thị tất cả kết quả của sinh viên

```
SELECT ALL KetQua  
FROM sv_detai;
```

**Truy vấn thay đổi tên cột, tên bảng và hiển thị các cột từ nhiều bảng khác nhau,**

- Muốn hiển thị các cột từ nhiều bảng khác nhau thì trong câu lệnh SELECT chúng ta phải làm như sau:

```
SELECT < Danh sách Tên_bang.Tên_cột>  
FROM <Danh sách Tên_bảng>  
WHERE <Điều kiện nối bảng>;
```

- Có thể đặt tên các cột trong kết quả các truy vấn bằng cách đặt tên mới vào sau cột được chọn ngăn cách bởi từ khoá AS, tương tự ta có thể đặt tên mới cho các bảng

Ví dụ: Hiển thị danh sách sinh viên bao gồm họ đệm, tên, kết quả từ bảng sinhvien và bảng sv\_dt:

```
SELECT sinhvien.HoDem AS Ho, sinhvien.Ten AS Ten, sv_dt.KetQua AS KQ  
FROM sinhvien AS sv, sv_dt AS sd  
WHERE (sv.MaSV = sd.MaSV);
```

**Truy vấn sử dụng lượng từ DISTINCT/ ALL**

\* Để tránh tình trạng đưa ra các bộ (hàng/bản ghi) trùng lặp trong các kết quả truy vấn thì SQL có lượng từ DISTINCT.



Ví dụ: Hiển thị các mã đề tài được sinh viên đăng ký trong bảng đề tài

```
SELECT DISTINCT MaDT
```

```
FROM sinhvien;
```

\* Để hiển thị tất cả các hàng (lấy cả các hàng có giá trị trùng nhau ) ta dùng lượng từ ALL

Ví dụ: Hiển thị tất cả các MaDT mà bảng sv\_dtai có

```
SELECT ALL MaDT
```

```
FROM sv_dtai;
```

***Chú ý: Ngầm định (nếu không viết Distinct/All) thì máy hiểu là All***

### **Truy vấn sử dụng mệnh đề GROUP BY**

- Để hiển thị các bản ghi theo nhóm ta dùng mệnh đề GROUP BY

- Ví dụ: Hiển thị bảng đề tài theo nhóm mã đề tài

```
SELECT MaDT
```

```
FROM detai
```

```
GROUP BY MaDT;
```

### **Truy vấn có sử dụng mệnh đề HAVING**

- Mệnh đề HAVING thường được sử dụng cùng mệnh đề GROUP BY. Sau HAVING là biểu thức điều kiện. Biểu thức điều kiện này không tác động vào toàn bảng được chỉ ra ở mệnh đề FROM mà chỉ tác động lần lượt từng nhóm các bản ghi đã chỉ ra tại mệnh đề GROUP BY.

- Ví dụ: Đếm xem có bao nhiêu đề tài đã được sinh viên đăng ký tham gia

```
SELECT MaDT
```

```
FROM detai
```

```
GROUP BY MaDT
```

HAVING COUNT(\*);

### **Truy vấn có sử dụng mệnh đề ORDER BY**

- Mệnh đề ORDER BY được dùng để sắp xếp dữ liệu trong bảng theo chiều tăng hoặc giảm (ASC hoặc DESC) của một cột nào đó.
- Mệnh đề ORDER BY nếu đứng sau GROUP BY thì miền tác động của sắp xếp là trong từng nhóm của cột được chỉ ra trong GROUP BY.
- Ví dụ: Sắp xếp bảng sinhvien theo chiều giảm dần của cột học lực

```
SELECT *
```

```
FROM sinhvien
```

```
ORDER BY hl DESC;
```

### **Truy vấn lồng nhau**

- Trong lệnh SELECT có thể được lồng nhiều mức
- Ví dụ 1: Hiển thị sinh viên có học lực cao nhất

```
SELECT *
```

```
FROM sinhvien
```

```
WHERE hl = (SELECT MAX(hl)
```

```
FROM sinhvien);
```

- Ví dụ 2: Hiển thị họcđem, tên của những sinh viên có kết quả  $\geq 9$  và có tên đề tài là Access

```
SELECT hodem,ten
```

```
FROM sinhvien
```

```
WHERE MaSV IN (SELECT MaSV
```

```
FROM sv_dtai
```

```
WHERE (kq  $\geq$  9) AND
```

```
( MaDT IN ( SELECT MaDT  
FROM detai  
WHERE tendt = "Access")));
```

## An toàn dữ liệu

Ngôn ngữ SQL cho phép người sử dụng kiểm tra dữ liệu của mình khi cập nhật và tuyên bố quyền truy nhập tới cơ sở dữ liệu để đảm bảo cho tính nhất quán và toàn vẹn dữ liệu. Đặc biệt trong trường hợp có nhiều người cùng sử dụng hệ thống, nhất là cùng truy nhập tới cùng một tệp (bảng) của CSDL.

Vì vậy cần phải có những biện pháp phòng ngừa để ngăn ngừa các nguy cơ

- Vô tình sử dụng sai
- Sự cố trong quá trình xử lý giao dịch
- Dị thường gây ra bởi truy cập đồng thời vào CSDL
- Dị thường gây ra bởi sự phân tán của dữ liệu trên một số máy tính
- Cố tình sử dụng sai
- Đọc dữ liệu một cách trái phép ( đánh cắp thông tin)
- Sửa đổi dữ liệu trái phép
- Phá hoại dữ liệu trong CSDL
- ...

Bảo vệ dữ liệu trên hai phương diện:

- An toàn dữ liệu: Chỉ việc bảo vệ CSDL tránh khỏi những hiện tượng cố tình sử dụng sai dữ liệu
- Toàn vẹn dữ liệu: Chỉ việc tránh khỏi những hiện tượng vô tình hay cố ý làm mất tính nhất quán của dữ liệu.

Sau đây là một số biện pháp:

Biện pháp : Sự cấp quyền

Nói một cách cụ thể người sử dụng có thể có một số quyền truy cập sau:

Quyền đọc: được phép đọc dữ liệu trong CSDL

Quyền chèn thêm dữ liệu: Được phép chèn thêm dữ liệu mới vào trong CSDL có sẵn nhưng không được thay đổi bất kỳ dữ liệu có sẵn nào.

Quyền cập nhật: Được phép sửa đổi dữ liệu nhưng không được xoá dữ liệu

Quyền xoá: Được phép xoá dữ liệu trong CSLD

Ngoài quyền truy cập, người sử dụng còn có thể được phép sửa đổi lược đồ CSDL:

Quyền tạo chỉ mục: Được phép tạo các chỉ mục

Quyền quản lý tài nguyên: Được phép tạo các quan hệ mới

Quyền thay đổi: Được phép thêm hoặc xoá các thuộc tính trong quan hệ

Quyền loại bỏ: Loại bỏ một quan hệ

Biện pháp 2: Quyền tạo và sử dụng khung nhìn (Views)

Khung nhìn là một phương thức cho phép:

+ Che dấu những dữ liệu mà người sử dụng cụ thể nào đó không cần thiết phải “nhìn” thấy

+ Làm đơn giản hoá việc sử dụng hệ thống

+ Làm tăng cường an toàn dữ liệu

+ ...

Biện pháp 3: Sự cấp đặc quyền luân chuyển dữ liệu

Một người sử dụng có thể chuyển quyền cho những người sử dụng khác và cần phải kiểm tra cẩn thận những đặc quyền này.

Sơ đồ chuyển quyền:

Những câu lệnh cấp và thu hồi quyền trong SQL:

- Lệnh cấp quyền: Việc tuyên bố và kiểm tra quyền truy nhập CSDL được thực hiện qua mệnh đề GRANT. Cú pháp như sau:

```
GRANT danh_sách_quyền ON đối_tượng TO danh_sách_người_sử_dụng [WITH GRANT OPTION];
```

trong đó:

+ *danh\_sách\_quyền\_truy\_nhập*: trong SQL bao gồm: read, select, write, insert, update, delete và run.

+ *đối\_tượng*: là tên bảng hoặc tên khung hoặc tên chương trình

+ *danh\_sách\_người\_sử\_dụng*: là tên người hoặc một nhóm người

+ Từ khoá WITH GRANT OPTION đảm bảo cho người sử dụng có thể tiếp tục trao quyền cho người khác khi cần.

Ví dụ: Trao quyền đọc bảng S cho cô Hồng và khi cần thì cô Hồng có thể trao quyền cho người khác

```
GRANT read ON S TO Hong GRANT OPTION ;
```

- Lệnh thu hồi quyền: REVOKE, cú pháp như sau:

```
REVOKE danh_sách_quyền ON tên_bảng/tên_khung_nhìn/  
tên_chương_trình/ FROM danh_sách_người_sử_dụng;
```

*Chú ý:* Việc huỷ bỏ quyền của một người sử dụng kéo theo việc huỷ bỏ quyền của những người sử dụng được uỷ quyền.

Biện pháp 4: Mã hoá dữ liệu

# Bài Tập Thực Hành

## Bài tập thực hành

### CHƯƠNG I: ĐẠI SỐ QUAN HỆ

**Bài 1:** Cho ba quan hệ sau:

R1(	A	B	C)	R2(	D	E)	R3(	B	C	D)
	a	b	c		c	e		b	c	d
	d	b	c		c	f		b	c	e
	b	b	f		d	f		a	d	b
	c	a	d							

Thực hiện các phép tính quan hệ sau:

1.  $R1 \times R2$
2.  $R1 * R2$
3.  $\delta_D = d(R1 * R3)$
4.  $R1 * R2 * R3$
5.  $\pi_{BC} (R1 * R2 * R3)$

**Bài 2:** Cho hai quan hệ sau:

R1(	A	B)	R2(	D	E)
	a	c		e	a
	b	c		b	c
	e	f		e	e

Thực hiện các phép tính quan hệ sau:

1.  $R1 \times R2$
2.  $R1 * R2$
3.  $\pi_{BE} (R1 * R2)$

## CHƯƠNG 2: PHỤ THUỘC HÀM

**Bài 1:** Cho quan hệ r dưới đây:

r{	A	B	C	D	E
	a1	b1	c1	d1	e1
	a2	b2	c2	d2	e1
	a2	b1	c3	d3	e1
	a2	b1	c4	d3	e1
	a3	b2	c5	d1	e1

Tìm những phụ thuộc hàm thoả r

**Bài 2:** Phát biểu hệ tiên đề Armstrong cùng các luật suy dẫn

**Bài 3:** Cho lược đồ quan hệ R và tập các phụ thuộc hàm

$$F = \{ AB \rightarrow E, AG \rightarrow I, BE \rightarrow I, E \rightarrow G, GI \rightarrow H \}$$

Chứng minh rằng:  $AB \rightarrow GH$

**Bài 4:** Cho lược đồ quan hệ R và tập các phụ thuộc hàm

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

Chứng minh rằng:  $AB \rightarrow E, AB \rightarrow G$

**Bài 5:** Nêu thuật toán tìm bao đóng của một tập thuộc tính.

**Bài 6:** Cho lược đồ quan hệ  $R = (ABCDEFGHI)$  và tập phụ thuộc hàm F xác định trên R

$$F = \{ A \rightarrow D, AB \rightarrow DE, CE \rightarrow G, E \rightarrow H \}$$

Tính bao đóng:  $(AB)^+$

**Bài 7:** Cho lược đồ quan hệ  $R = (ABCDEF)$  và tập phụ thuộc hàm F xác định trên R

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

Tính bao đóng:



a)  $(AB)^+$

b)  $(BD)^+$

**Bài 8:** Phát biểu thuật toán tìm khoá của một quan hệ

**Bài 9:** Cho lược đồ quan hệ  $R = ( ABCDEG )$  và tập phụ thuộc hàm  $F$

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

Hãy tìm khoá của lược đồ trên.

**Bài 10:** Cho lược đồ quan hệ  $R = ( ABCDEF )$

$F = \{ AB \rightarrow C, C \rightarrow B, ABD \rightarrow E, F \rightarrow A \}$

- Hãy tìm một khoá của lược đồ quan hệ trên
- Tập  $ABC$  có phải là khoá của lược đồ trên không? Vì sao?
- Tập  $BC$  có phải là khoá của lược đồ trên không? Vì sao?
- Lược đồ trên còn khoá nào nữa không?

### CHƯƠNG 3: CHUẨN HOÁ

**Bài 1:** Nêu định nghĩa lược đồ quan hệ ở dạng chuẩn 1NF, 2NF, 3NF, BCNF

**Bài 2:** Cho lược đồ quan hệ  $R = ( \underline{A}BCD )$  và tập phụ thuộc hàm  $F$

$F = \{ A \rightarrow B, A \rightarrow BC, A \rightarrow BD \}$

Lược đồ trên có ở dạng chuẩn 2NF? Vì sao? Nếu chưa hãy chuẩn hoá lược đồ về 2NF

**Bài 3:** Cho lược đồ quan hệ  $R = ( \underline{A}BCD )$  và tập phụ thuộc hàm  $F$

$F = \{ A \rightarrow D, AB \rightarrow DC \}$

Lược đồ trên có ở dạng chuẩn 2NF? Vì sao? Nếu chưa hãy chuẩn hoá lược đồ về 2NF

**Bài 4:** Cho lược đồ quan hệ  $R = ( \underline{ABCDEF} )$  và tập phụ thuộc hàm  $F$

$F = \{ AB \rightarrow C, AB \rightarrow D, AB \rightarrow EF \}$

Lược đồ trên có ở dạng chuẩn 3NF? Vì sao? Nếu chưa hãy chuẩn hoá lược đồ về 3NF

**Bài 5:** Cho lược đồ quan hệ  $R = ( \underline{ABCDEF} )$  và tập phụ thuộc hàm  $F$

$$F = \{ AB \rightarrow C, AB \rightarrow D, E \rightarrow F \}$$

Lược đồ trên có ở dạng chuẩn 3NF? Vì sao? Nếu chưa hãy chuẩn hoá lược đồ về 3NF

**Bài 6:** Cho lược đồ quan hệ  $R = (\underline{A}BCDEFGH)$  và tập phụ thuộc hàm  $F$

$$F = \{ AB \rightarrow C, AB \rightarrow D, AB \rightarrow EF, AB \rightarrow GH \}$$

Lược đồ trên có ở dạng chuẩn BCNF? Vì sao? Nếu chưa hãy chuẩn hoá lược đồ về BCNF

**Bài 7:** Cho lược đồ quan hệ  $R = (\underline{A}BCDEFGH)$  và tập phụ thuộc hàm  $F$

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

Lược đồ trên có ở dạng chuẩn BCNF? Vì sao? Nếu chưa hãy chuẩn hoá lược đồ về BCNF

**Bài 8:** Kiểm tra tính kết nối không mất mát thông tin của

$$R = ABCDE \text{ thành: } R_1 = AD, R_2 = AB, R_3 = BE, R_4 = CDE, R_5 = AE$$

Với tập phụ thuộc hàm

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

**Bài 9:** Kiểm tra tính kết nối không mất mát thông tin của

$$R = ABCD \text{ thành: } R_1 = AB, R_2 = ACD$$

Với tập phụ thuộc hàm

$$F = \{ A \rightarrow B, AC \rightarrow D \}$$

**Bài 10:** Kiểm tra tính kết nối không mất mát thông tin của

$$R = ABCDEI \text{ thành: } R_1 = AD, R_2 = AB, R_3 = BE, R_4 = CDE$$

Với tập phụ thuộc hàm

$$F = \{ A \rightarrow I, B \rightarrow C, C \rightarrow D, DE \rightarrow C, CE \rightarrow A \}$$

Bài 11: Cho biết dạng chuẩn của các lược đồ quan hệ sau

1.  $Q(ABCDEG); F = \{A \rightarrow BC, C \rightarrow DE, E \rightarrow G\}$
2.  $Q(ABCDEFGH); F = \{C \rightarrow AB, D \rightarrow E, B \rightarrow G\}$
3.  $Q(ABCDEFGH); F = \{A \rightarrow BC, D \rightarrow E, H \rightarrow G\}$
4.  $Q(ABCDEG); F = \{AB \rightarrow C, C \rightarrow B, ABD \rightarrow E, G \rightarrow A\}$
5.  $Q(ABCDEFGHI); F = \{AC \rightarrow B, BI \rightarrow ACD, ABC \rightarrow D, H \rightarrow I, ACE \rightarrow BCG, CG \rightarrow AE\}$

Bài 12: Kiểm tra bảo toàn thông tin

$Q(ABCDE); R1(DA); R2(AB); R3(BE); R4(CDE); R5(AE)$

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

Bài 13: Cho lược đồ quan hệ  $Q(ABCD)$  và tập phụ thuộc hàm  $F = \{A \rightarrow B, B \rightarrow C, A \rightarrow D, D \rightarrow C\}$

Và một lược đồ CSDL như sau:  $C = \{Q1(AB), A2(AC), Q3(BD)\}$

1. C có bảo toàn thông tin đối với F
2. C có bảo toàn phụ thuộc hàm

Bài 14: Cho lược đồ CSDL

Kehoach(NGAY, GIO, PHONG, MONHOC, GIAOVIEN)

$F = \{NGAY, GIO, PHONG \rightarrow MONHOC$

$MONHOC, NGAY \rightarrow GIAOVIEN$

$MONHOC \rightarrow GIAOVIEN\}$

1. Xác định dạng chuẩn cao nhất của Kehoach
2. Nếu Kehoach chưa đạt dạng chuẩn 3, hãy phân rã Kehoach thành lược đồ CSDL dạng chuẩn 3 vừa bảo toàn phụ thuộc hàm vừa bảo toàn thông tin.
3. Nếu Kehoach chưa đạt dạng chuẩn BC, hãy phân rã Kehoach thành lược đồ CSDL dạng BC

Bài 15: Cho lược đồ quan hệ  $Q(ABCD)$  và tập phụ thuộc hàm F

$F = \{A \rightarrow B, B \rightarrow C, D \rightarrow B\}$   $C = \{Q1(ACD), Q2(BD)\}$

1. Xác định  $F_i$  (những phụ thuộc hàm F được bao đóng trong  $Q_i$ )
2. Lược đồ CSDL C có đạt dạng chuẩn BC. Nếu không có thể phân rã  $Q_i$  của C để biến C thành dạng chuẩn BC

Bài 16 Giả sử ta có lược đồ quan hệ Q(CDEGHK) và tập phụ thuộc hàm F như sau

$F = \{CK \rightarrow H, C \rightarrow D, E \rightarrow C, E \rightarrow G, CK \rightarrow E\}$

1. từ tập F hãy chứng minh  $EK \rightarrow DH$
2. tìm tất cả các khoá của Q
3. Xác định dạng chuẩn của Q
4. Hãy tìm cách phân rã Q thành một lược đồ CSDL đạt dạng chuẩn BC. Tìm tập phụ thuộc hàm và khoá cho mỗi lược đồ quan hệ con.

Bài 17 Cho lược đồ quan hệ Q(SIDM)

$F = \{f_1: SI \rightarrow DM, f_2: SD \rightarrow M, f_3: D \rightarrow M\}$

1. Tìm bao đóng của  $D^+, SD^+, SI^+$
2. Tìm tất cả các khoá của Q
3. Tìm phủ tối thiểu của F
4. Xác định dạng chuẩn của Q
5. Nếu Q chưa đạt dạng chuẩn 3, hãy phân rã Q thành lược đồ CSDL dạng chuẩn 3 vừa bảo toàn phụ thuộc hàm vừa bảo toàn thông tin
6. Nếu Q chưa đạt dạng chuẩn BCNF, hãy phân rã Q thành lược đồ CSDL dạng BCNF
7. Kiểm tra phép tách Q thành các lược đồ con (SID, SIM) có bảo toàn phụ thuộc hàm

Bài 18 Cho lược đồ quan hệ

$R(W, A, Z, Y, Q, P)$

$R_1(A, Z)$

$R_2(W, Y, Q, P)$

$R_3(Y, Q, P, A)$

$F = \{W \rightarrow AYQP, A \rightarrow Z, YQP \rightarrow A\}$

Hãy kiểm tra tính kết nối không mất thông tin.

Bài 19 Cho lược đồ quan hệ Q(môn, Giảng Viên, Giờ giảng, Phòng, Sinh Viên, Hạng) với

$F = \{M \rightarrow GV; G, P \rightarrow M; G, GV \rightarrow P; M, SV \rightarrow H; G, SV \rightarrow P\}$

$C = \{Q1(M, G, P); Q2(M, GV); Q3(M, SV, H)\}$

Kiểm tra xem lược đồ cơ sở dữ liệu sau đây có bảo toàn thông tin đối với F

Bài 20 Kiểm tra dạng chuẩn

1.  $Q(A, B, C, D) F = \{CA \rightarrow D; A \rightarrow B\}$
2.  $Q(S, D, I, M) F = \{SI \rightarrow D; SD \rightarrow M\}$
3.  $Q(N, G, P, M, GV) F = \{N, G, P \rightarrow GV\}$
4.  $Q(S, N, D, T, X) F = \{S \rightarrow N; S \rightarrow D; S \rightarrow T; S \rightarrow X\}$

Bài 20 Phân rã lược đồ thành dạng 3NF vừa bảo toàn phụ thuộc hàm vừa bảo toàn thông tin

1.  $Q(A, B, C) F = \{A \rightarrow B, A \rightarrow C, B \rightarrow A, C \rightarrow A, B \rightarrow C\}$
2.  $Q(A, B, C, D) F = \{AB \rightarrow C, C \rightarrow B\}$

## CHƯƠNG 5: SQL

**Bài 1:** Cho CSDL *Thực tập* gồm ba quan hệ như sau:

SV(MSV, HT, NS, QUE, HL)

DT(MDT, TDT, CN, KP)

SD(MSV, MDT, NTT, KM, KQ)

Trong đó:

SV: sinh viên	TDT: tên đề tài
MSV: mã sinh viên	CN: chủ nhiệm
HT: họ tên	KP: kinh phí
NS: ngày sinh	SD: sinh viên, đề tài
QUE: quê	NTT: Nơi thực tập
HL: học lực	KM: Kilomet ( Khoảng cách từ trường đến nơi thực
DT: đề tài	tập)
MDT: mã đề tài	KQ: kết quả

Hãy thực hiện các câu hỏi sử dụng câu lệnh của SQL:

1. Đưa ra danh sách sinh viên có tuổi <18 và có HL >8.5
2. Cho biết thông tin về các đề tài được cấp kinh phí trên 10 triệu đồng
3. Đưa ra danh sách sinh viên < 18 tuổi, HL và KQ đều > 8.5
4. Cho biết danh sách các chủ nhiệm đề tài có các sinh viên quê ở Hà Nội tham gia
5. Đưa ra danh sách các sinh viên học giỏi hơn các sinh viên Hà Nội
6. Cho biết điểm trung bình của các sinh viên Hà Nội
7. Cho biết tổng số đoạn đường thực tập theo đề tài số 5
8. Cho biết tổng số sinh viên đi thực tập
9. Đưa ra danh sách các tỉnh và số sinh viên quê ở tỉnh đó, nhóm theo QUE
10. Đưa ra danh sách các sinh viên sinh trước năm 1980 và có quê ở Hải Phòng
11. Cho biết danh sách các sinh viên có nơi thực tập tại quê nhà

**Bài 2:** Cho CSDL Nhanvien\_Duan như sau:

Nhanvien(MaNV, Hoten, Ngaysinh, MaPhong)

Phong(MaPhong, TenPhong, Diadiem, SoDT)

Duan(MaDA, TenDA, Ngansach)

Thamgia(MaNV, MaDA, SogioThamgia)

Biểu diễn các câu truy vấn sau bằng ngôn ngữ SQL:

1. Đưa ra danh sách Họ tên, Ngày sinh của các nhân viên tham gia dự án có tên là “Đào tạo từ xa” hoặc tham gia dự án có tên là “112”.
2. Đưa ra danh sách Tên phòng, Địa điểm của phòng có nhân viên mã số “NV-101” làm việc
3. Cho biết danh sách Họ tên, Ngày sinh, Mã phòng của các nhân viên tham gia tất cả các dự án

**Bài 3:** Cho CSDL gồm các quan hệ sau:

Nhanvien(MaNV, Hoten, Diachi, Ngaysinh)

DuA

n(MaDA, TenDA, Chudautu, Ngansach)

ThamGia(MaNV, MaDA, SogioLamviec)

Hãy biểu diễn các câu hỏi sau bằng SQL:

1. Đưa ra danh sách MaNV, Hoten, Ngaysinh của những nhân viên làm cho dự án có chủ đầu tư là “VINACO”.
2. Đưa ra danh sách bao gồm TenDA, Chudautu, của các dự án có ngân sách khoảng từ 10.000.000 đến 25.000.000

**Bài 4:** Cho CSDL bao gồm các quan hệ sau:

CB(MaCB, TenCB, Tel, Namsinh, Luong, MaPhong)

Phong( MaPhong, TenPhong, TruongPhong)

TĐVH(MaCb, TrinhDo, Thoigian)

Hãy biểu diễn các câu truy vấn sau bằng ngôn ngữ SQL:

1. Đưa ra danh sách cán bộ có trình độ “Tiến sĩ” của phòng “kỹ thuật”
2. Thay đổi số điện thoại cho cán bộ có mã là “CB01” với số điện thoại mới là 7717674
3. Đưa ra danh sách cán bộ có trình độ đại học và có lương > 2.000.000

**Bài 5:** Cho CSDL gồm các quan hệ:

SV(MSV, Hoten, Malop, Diachi)

LOP(Malop, Tenlop, SoSV)

CBGD(MaCBDG, TenCB, MonDay, MaLop, Sotiet)

Hãy biểu diễn các truy vấn sau bằng ngôn ngữ SQL:

### **Bài tập làm trên máy**

Bài thực hành số 1:

Sử dụng câu lệnh CREATE TABLE của SQL để tạo bảng:

Mở Access tạo một CSDL mới: Quanlysinhvien

CREATE TABLE KHOA CREATE TABLE GIAOVIEN

(Makhoa Varchar(3) Not null, (Magv varchar(4) Not null,

Tenkhoa Varchar(20) Not null, Tengv Varchar(40) Not null,

PRIMARY KEY(Makhoa)); Ngaysinh Date,

Trinhdo Varchar(10),

PRIMARY KEY (Mamon, Magv));

CREATE TABLE SINHVIEN CREATE TABLE MON

(Masv Varchar(4) Not null, (Mamon Varchar(3) Not null,

Hodem Varchar(15) Not null, Tenmon Varchar(20) Not null,

Ten Varchar(7) Not null, Magv Varchar(4) Not null,

Que Varchar(50),

Gioitinh Varchar(3), CREATE TABLE BANGDIEM

Ngaysinh Date, (Masv Varchar(4) Not null,

Hocbong IN, Mamon Varchar(3) Not null,

Makhoa Varchar(3) Not null, Lanthi bye Not null,

PRIMARY KEY(Masv)); Diem Real Not null,

PRIMARY KEY (Masv, Mamon, Lanthi));

Bài thực hành số 2:

Sử dụng câu lệnh INSERT INTO để nhập giá trị vào bảng :

Ví dụ nhập dữ liệu cho bảng môn:

INSERT INTO KHOA VALUES(“KHO1”,”Toan tin”)

KHOA	
Makhoa	Tenkhoa
KH1	Toán tin
KH2	Tiếng Anh
KH3	Luật



KH4	Công nghệ thông tin
-----	---------------------

GIAOVIEN			
Magv	Tengv	Namsinh	Trinhdo
GV01	Lê Mai	11/12/1976	Đại học
GV02	Hà Châu	03/12/1955	Thạc sỹ
GV03	Trần Tuấn	05/07/1972	Đại học
GV04	QuỳnhAnh	16/05/1980	Đại học
GV05	HoangHoa	06/04/1973	Thạc sỹ
GV06	ThanhToan	12/12/1947	Tiến sỹ

MON		
Mamon	Tenmon	Magv
MO1	Tin Văn Phòng	MG01
MO2	Lập trình Pascal	MG01
MO3	Cow sở dữ liệu	MG01
MO4	Toán rời tạc	MG01
MO5	Tiếng Anh	MG01
MO6	Điện tử số	MG01

SINHVIEN							
Masv	Hodem	Ten	Que	Giotinh	Ngaysinh	Hocbong	Khoa
SV01	Trần	Toàn	Hà Nội	Nam	12/12/80		KH01
SV01	Le	Na	Hà Nội	Nữ	24/03/84	100	KH02
SV01	Phan	Ngoc	Thai Bình	Nam	16/05/84		KH01
SV01	Thai	Trung	Hà Nội	Nam	11/09/83	200	KH01
SV01	Nguyễn	Huy	Ninh Bình	Nữ	25/12/81		KH01
SV01	Nguyễn	Minh	Thanh Hoá	Nữ	31/04/87		KH02
SV01	Hoàng	Dương	Huế	Nam	05/04/87	500	KH02

SV0	Hoàng	Việt	Hà Nội	Nam	25/10/81		KH01
VS0	Quynh	Anh	Ninh Bình	Nữ	06/05/82		KH02

BANGDIEM			
Masv	Mamon	Lanthi	Diem
SV01	M01	1	6
SV01	M02	1	3
SV01	M03	2	5
SV02	M04	1	8
SV02	M05	1	9
SV03	M06	1	2
SV03	M01	1	4
SV04	M03	1	6
SV04	M05	1	7
SV05	M02	2	8
SV05	M05	1	5
SV06	M05	1	6
SV06	M01	1	9
SV06	M03	1	9
SV07	M05	2	10
SV07	M02	2	7

Bài thực hành số 3:

Tạo các Query thực hiện các câu hỏi sau:

1. Hiện thị điểm cao nhất trong danh sách bảng điểm
2. Hiện thị điểm thấp nhất trong danh sách bảng điểm
3. Hiện thị điểm cao nhất theo từng môn
4. Hiện thị điểm thấp nhất theo từng môn
5. Tính điểm trung bình của các sinh viên theo từng môn học
6. Tính điểm trung bình của các môn học cho từng sinh viên

7. Tìm tên của những sinh viên có lần thi bằng 1
8. Hiển thị họ tên của những sinh viên không phải thi lại môn nào( thi lại khi điểm < 5)
9. Hiển thị họ tên sinh viên, quê, ngày sinh của sinh viên có điểm cao nhất theo danh sách bảng điểm
10. Hiển thị họ tên, quê, giới tính của sinh viên học khoa “toán tin” và điểm trung bình các môn >5
11. Hiển thị họ tên sinh viên học khoa là “Luật” hoặc “Tiếng Anh” có lần thi là 2 và có điểm là 3, 4, 5
12. Hiển thị các sinh viên có điểm thấp hơn điểm trung bình của bảng điểm
13. Tìm tên của các môn và sinh viên có tên “Huy” có điểm >5
14. Tìm tên của các môn mà sinh viên có tên “Huy” có lần thi lại là 2 và điểm > 5
15. Đếm số lần thi của từng sinh viên
16. Hiển thị tên và số lần thi của từng sinh viên
17. Hiển thị họ tên, lần thi và điểm cao nhất trong các lần thi theo môn của từng sinh viên
18. Hiển thị họ tên sinh viên, môn thi, điểm cao nhất của môn , tổng số lần thi

Bài thực hành số 4:

1. Hiển thị tên, điểm cao nhất theo từng môn
2. Hiển thị Masv, Điểm trung bình của các sinh viên theo môn
3. Hiển thị danh sách Họ đệm, Tên sinh viên, tên môn và điểm của sinh viên
4. Hiển thị Masv, Họ đệm, Tên , Điểm trung bình của sinh viên theo môn
5. Tìm tên sinh viên có lần thi là 1 và môn học là môn “Tin văn phòng”
6. Tìm tên sinh viên có lần thi là 1 và môn học là môn “Toán rời rạc” và Điểm  $\geq 5$
7. Hiển thị Masv, họ đệm, tên sinh viên phải thi lại ít nhất 1 môn

## Tham gia đóng góp

Tài liệu: Giáo trình Cơ Sở Dữ Liệu

Biên tập bởi: Viện CNTT – DHQG Hanoi

URL: <http://voer.edu.vn/c/09e95dc9>

Giấy phép: <http://creativecommons.org/licenses/by/3.0/>

Module: Hệ thống cơ sở dữ liệu

Các tác giả: unknown

URL: <http://www.voer.edu.vn/m/d2694ff6>

Giấy phép: <http://creativecommons.org/licenses/by/3.0/>

Module: Các mô hình dữ liệu

Các tác giả: unknown

URL: <http://www.voer.edu.vn/m/0b246dbc>

Giấy phép: <http://creativecommons.org/licenses/by/3.0/>

Module: Mô hình dữ liệu quan hệ - Các khái niệm cơ bản

Các tác giả: unknown

URL: <http://www.voer.edu.vn/m/54558900>

Giấy phép: <http://creativecommons.org/licenses/by/3.0/>

Module: Một số định nghĩa

Các tác giả: unknown

URL: <http://www.voer.edu.vn/m/f5371efc>

Giấy phép: <http://creativecommons.org/licenses/by/3.0/>

Module: Hệ tiên đề cho phụ thuộc hàm

Các tác giả: unknown

URL: <http://www.voer.edu.vn/m/bec9256d>

Giấy phép: <http://creativecommons.org/licenses/by/3.0/>

Module: Các dạng chuẩn hóa dữ liệu

Các tác giả: unknown

URL: <http://www.voer.edu.vn/m/3ff7a7b7>

Giấy phép: <http://creativecommons.org/licenses/by/3.0/>

Module: Các thuật toán phân rã

Các tác giả: unknown

URL: <http://www.voer.edu.vn/m/8ef96164>

Giấy phép: <http://creativecommons.org/licenses/by/3.0/>

Module: Sơ đồ thực thể liên kết(ERD)

Các tác giả: unknown

URL: <http://www.voer.edu.vn/m/e2336b82>

Giấy phép: <http://creativecommons.org/licenses/by/3.0/>

Module: Giới thiệu

Các tác giả: unknown

URL: <http://www.voer.edu.vn/m/e0e803c3>

Giấy phép: <http://creativecommons.org/licenses/by/3.0/>

Module: Các lệnh và các mệnh đề cơ bản trong SQL

Các tác giả: unknown

URL: <http://www.voer.edu.vn/m/b75e99ed>

Giấy phép: <http://creativecommons.org/licenses/by/3.0/>

Module: An toàn dữ liệu

Các tác giả: unknown

URL: <http://www.voer.edu.vn/m/04587620>

Giấy phép: <http://creativecommons.org/licenses/by/3.0/>

Module: Bài tập thực hành

Các tác giả: unknown

URL: <http://www.voer.edu.vn/m/92226908>

Giấy phép: <http://creativecommons.org/licenses/by/3.0/>

## **Chương trình Thư viện Học liệu Mở Việt Nam**

Chương trình Thư viện Học liệu Mở Việt Nam (Vietnam Open Educational Resources – VOER) được hỗ trợ bởi Quỹ Việt Nam. Mục tiêu của chương trình là xây dựng kho Tài nguyên giáo dục Mở miễn phí của người Việt và cho người Việt, có nội dung phong phú. Các nội dung đều tuân thủ Giấy phép Creative Commons Attribution (CC-by) 4.0 do đó các nội dung đều có thể được sử dụng, tái sử dụng và truy nhập miễn phí trước hết trong môi trường giảng dạy, học tập và nghiên cứu sau đó cho toàn xã hội.

Với sự hỗ trợ của Quỹ Việt Nam, Thư viện Học liệu Mở Việt Nam (VOER) đã trở thành một cổng thông tin chính cho các sinh viên và giảng viên trong và ngoài Việt Nam. Mỗi ngày có hàng chục nghìn lượt truy cập VOER ([www.voer.edu.vn](http://www.voer.edu.vn)) để nghiên cứu, học tập và tải tài liệu giảng dạy về. Với hàng chục nghìn module kiến thức từ hàng nghìn tác giả khác nhau đóng góp, Thư Viện Học liệu Mở Việt Nam là một kho tàng tài liệu khổng lồ, nội dung phong phú phục vụ cho tất cả các nhu cầu học tập, nghiên cứu của độc giả.

Nguồn tài liệu mở phong phú có trên VOER có được là do sự chia sẻ tự nguyện của các tác giả trong và ngoài nước. Quá trình chia sẻ tài liệu trên VOER trở lên dễ dàng như đếm 1, 2, 3 nhờ vào sức mạnh của nền tảng Hanoi Spring.

Hanoi Spring là một nền tảng công nghệ tiên tiến được thiết kế cho phép công chúng dễ dàng chia sẻ tài liệu giảng dạy, học tập cũng như chủ động phát triển chương trình giảng dạy dựa trên khái niệm về học liệu mở (OCW) và tài nguyên giáo dục mở (OER). Khái niệm chia sẻ tri thức có tính cách mạng đã được khởi xướng và phát triển tiên phong bởi Đại học MIT và Đại học Rice Hoa Kỳ trong vòng một thập kỷ qua. Kể từ đó, phong trào Tài nguyên Giáo dục Mở đã phát triển nhanh chóng, được UNESCO hỗ trợ và được chấp nhận như một chương trình chính thức ở nhiều nước trên thế giới.