

## Cơ sở dữ liệu

Nguyễn Hồng Phương

[phuongnh-fit@mail.hut.edu.vn](mailto:phuongnh-fit@mail.hut.edu.vn)

Bộ môn Hệ thống thông tin  
Viện Công nghệ thông tin và Truyền thông  
Đại học Bách Khoa Hà Nội

1

## Thông tin chung

- Giảng viên
  - Nguyễn Hồng Phương
  - Bộ môn Hệ thống thông tin, khoa CNTT, phòng 325 – C1
  - Email: [phuongnh-fit@mail.hut.edu.vn](mailto:phuongnh-fit@mail.hut.edu.vn)
- Giờ tiếp sinh viên tại Bộ môn:
  - Chiều thứ hai hàng tuần.
  - Ngoài ra, xin liên hệ trước.



2

## Tổng quan về môn học

- **Mục tiêu:** Sau khi học xong môn học này, sinh viên ngành công nghệ thông tin có thể:
  - Chỉ ra nguyên lý của hệ cơ sở dữ liệu (CSDL).
  - Thiết kế và xây dựng một hệ CSDL.
- Khối lượng: 4 tc, trong 15 tuần

3

## Đánh giá môn học

- Dự lớp đầy đủ, tích cực xây dựng bài
- Kiểm tra giữa kỳ
- Kiểm tra cuối kỳ



4

## Tài liệu học tập



- Bài giảng trên lớp
- Sách tham khảo:
  - Nguyễn Kim Anh, *Nguyên lý của các hệ cơ sở dữ liệu*, NXB Đại học Quốc gia, Hà Nội, 2004.
  - Tô Văn Nam, *Giáo trình Cơ sở dữ liệu*, NXB Giáo dục, 2006.
  - Nguyễn Thiên Bằng, Phương Lan, *Giáo trình SQL Server 2000*, 2004.
  - Nguyễn Ngọc Minh, Hoàng Đức Hải, Trần Tiến Dũng, *Tự học Microsoft SQL Server 2000 trong 21 ngày*, NXB Lao động-Xã hội, 2002.
  - J.D.Ullman, *A First Course in Database Systems*, Prentice-Hall, 1997.
  - J.D.Ullman, *Principles of Database and Knowledge-Base Systems*, vol.1, Computer Science Press, 1988.
  - Các tài liệu khác...

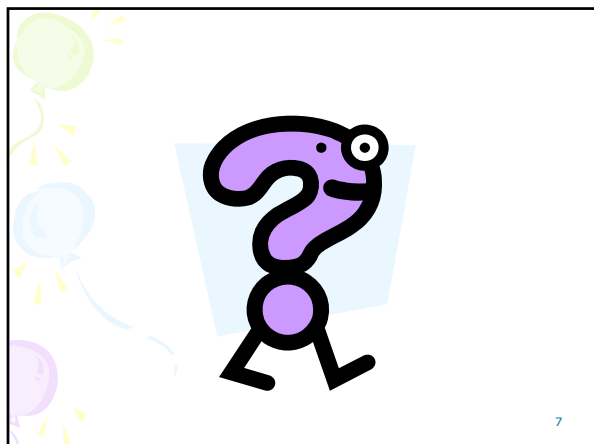
5

## Nội dung môn học



- Chương 1: Đại cương về các hệ CSDL.
- Chương 2: Các mô hình dữ liệu.
- Chương 3: Ngôn ngữ định nghĩa và thao tác dữ liệu đối với mô hình quan hệ.
- Chương 4: Lý thuyết thiết kế CSDL quan hệ.
- Chương 5: Tối ưu hóa câu truy vấn
- Chương 6: An toàn và toàn vẹn dữ liệu.

6



## Danh ngôn Hồ Chí Minh

"Trời có bốn mùa: Xuân, Hạ, Thu, Đông;  
Đất có bốn phương: Đông, Tây, Nam, Bắc;  
Người có bốn đức: **Cần, Kiệm, Liêm, Chính**;  
Thiếu một mùa không thể thành Trời;  
Thiếu một phương không thể thành Đất;  
Thiếu một đức không thể thành **Người**."

## Chương 1 Đại cương về các hệ cơ sở dữ liệu

Nguyễn Hồng Phương  
[phuongnh-fit@mail.hut.edu.vn](mailto:phuongnh-fit@mail.hut.edu.vn)

Bộ môn Hệ thống thông tin  
Viện Công nghệ thông tin và Truyền thông  
Đại học Bách Khoa Hà Nội

1

## Nội dung chương này

- 1.1 Các hệ thống xử lý tệp truyền thống và những hạn chế của nó.
- 1.2 Các hệ CSDL: khái niệm, khả năng, kiến trúc, người dùng của một hệ quản trị CSDL.
- 1.3 Sự phân loại các hệ CSDL.



2

### 1.1 Các hệ thống xử lý tệp truyền thống

- Bước khởi đầu của quá trình tin học hóa doanh nghiệp.
- Tập trung vào nhu cầu xử lý dữ liệu của các phòng riêng lẻ trong tổ chức mà không xem xét tổng thể tổ chức này.
- Viết một chương trình mới đối với mỗi ứng dụng đơn lẻ, không có kế hoạch, không có mô hình hướng đến sự tăng trưởng.



3

### Các hệ thống xử lý tệp truyền thống

- Mỗi chương trình ứng dụng định nghĩa và quản lý các tệp dữ liệu của riêng nó.
- Trước khi xuất hiện các phần mềm hệ quản trị CSDL, trong quá khứ các hệ thống trên cơ sở tệp đã được tạo lập để xử lý một số lượng lớn dữ liệu.



4

### Hạn chế của các hệ thống xử lý tệp truyền thống

- Dư thừa và không nhất quán dữ liệu
- Khó khăn trong truy nhập dữ liệu
- Cô lập và hạn chế chia sẻ dữ liệu
- Các vấn đề về an toàn và toàn vẹn
- Các vấn đề về độ tin cậy
- Sự phụ thuộc dữ liệu của các chương trình ứng dụng



5

## Nội dung chương này

- 1.1 Các hệ thống xử lý tệp truyền thống và những hạn chế của nó.
- 1.2 Các hệ CSDL: khái niệm, khả năng, kiến trúc, người dùng của một hệ quản trị CSDL.
- 1.3 Sự phân loại các hệ CSDL.



6

## 1.2 Các hệ cơ sở dữ liệu

- CSDL (database) là gì ?
- Tại sao phải sử dụng CSDL ?
- Tại sao phải tìm hiểu về các hệ CSDL (database systems) ?

7

## Ví dụ về quản lý đào tạo

- Thông tin cần quan tâm
  - Khóa học, lớp học, sinh viên, giáo viên, môn học,...
  - Thông tin về sinh viên: thông tin cá nhân, thông tin học tập,...
  - Thông tin về môn học: khối lượng, giáo viên, lịch học,...
- Cần **lưu trữ** những thông tin đa dạng

→ **Cơ sở dữ liệu**



8

## Ví dụ: khai thác thông tin

- Sinh viên
  - Các môn học của Khoa Công nghệ thông tin ?
  - Điểm thi môn “Hệ cơ sở dữ liệu” ?
- Giáo viên
  - Danh sách sinh viên lớp Tin2-K49 ?
  - Thời khóa biểu của lớp Tin2-K49 ?
- Giáo vụ
  - Danh sách sinh viên K47 tốt nghiệp loại giỏi ?...

→ **Phần mềm ứng dụng**



9

## “Hình dung” về xây dựng một CSDL

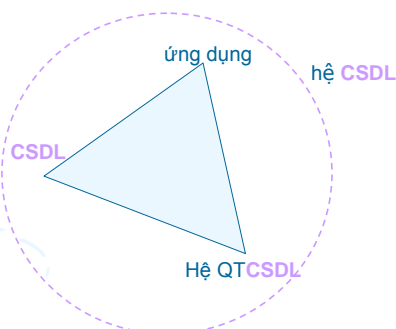
- Yêu cầu
  - Lưu trữ thông tin **cần thiết** một cách **chính xác**
  - Truy xuất thông tin **hiệu quả**
- Thực hiện
  - Xác định yêu cầu nghiệp vụ
  - Xác định thông tin cần lưu trữ
  - Xác định cách thức lưu trữ
- Cần công cụ trợ giúp xây dựng một CSDL

→ **Phần mềm quản trị CSDL**



10

## Các khái niệm cơ bản



11

## Cơ sở dữ liệu (database)

- Là một tập hợp các dữ liệu
  - Biểu diễn một vài khía cạnh nào đó của thế giới thực
  - Có liên hệ logic thống nhất
  - Được thiết kế và bao gồm những dữ liệu phục vụ một mục đích nào đó.
- Là một bộ sưu tập các dữ liệu tác nghiệp được lưu trữ lại và được các hệ ứng dụng của một xí nghiệp cụ thể nào đó sử dụng.

12

## Hệ quản trị cơ sở dữ liệu (Database Management System-DBMS)

- Là một hệ thống phần mềm cho phép
  - Định nghĩa, tạo lập: xác định kiểu, cấu trúc, ràng buộc dữ liệu, lưu trữ dữ liệu trên các thiết bị nhớ.
  - Thao tác: truy vấn, cập nhật, kết xuất,... các CSDL cho các ứng dụng khác nhau
- Ví dụ: MS SQL Server, DB2, MS Access, Oracle, FoxPro,...



13

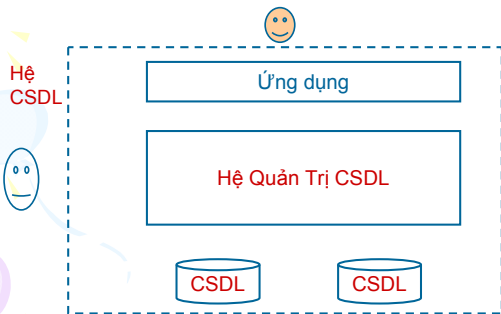
## Hệ cơ sở dữ liệu

- Là một hệ thống gồm 4 thành phần
  - Hệ quản trị CSDL
  - Phần cứng
  - CSDL và phần mềm ứng dụng
  - Những người sử dụng
- Ví dụ: Hệ quản lý đào tạo, hệ quản lý nhân sự, hệ quản lý kinh doanh,...



14

## Hệ CSDL



15

## Các tính năng của hệ quản trị CSDL

- Quản lý dữ liệu tồn tại lâu dài
  - Định nghĩa dữ liệu
  - Quản lý lưu trữ
- Truy xuất dữ liệu một cách hiệu quả
  - Biểu diễn các thao tác dữ liệu
  - Xử lý câu hỏi
  - Quản trị giao dịch



16

## Các tính năng của hệ quản trị CSDL

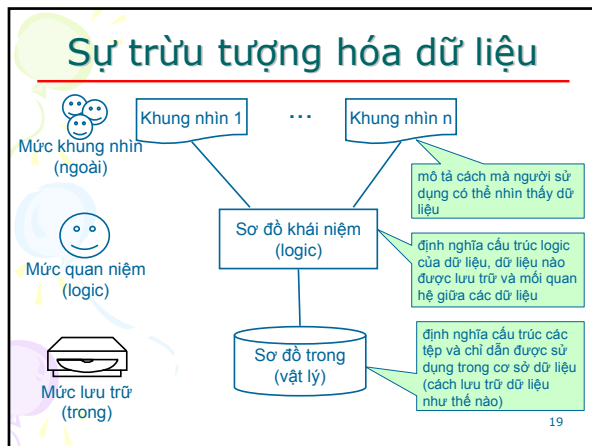
- Hỗ trợ ít nhất một mô hình dữ liệu
- Đảm bảo tính độc lập dữ liệu
- Hỗ trợ các ngôn ngữ cấp cao nhất định cho phép người sử dụng định nghĩa cấu trúc của dữ liệu, truy nhập và thao tác dữ liệu
- Điều khiển truy nhập
- Phục hồi dữ liệu

17

## Các ngôn ngữ

- Ngôn ngữ định nghĩa dữ liệu (**D**ata **D**efinition **L**anguage - DDL)
  - Cấu trúc dữ liệu
  - Mối liên hệ giữa các dữ liệu và quy tắc, ràng buộc áp đặt lên dữ liệu
- Ngôn ngữ thao tác dữ liệu (**D**ata **M**anipulation **L**anguage - DML)
  - Tìm kiếm, thêm, xóa, sửa dữ liệu trong CSDL
- Ngôn ngữ điều khiển dữ liệu (**D**ata **C**ontrol **L**anguage - DCL)
  - Thay đổi cấu trúc của các bảng dữ liệu
  - Khai báo bảo mật thông tin
  - Quyền hạn của người dùng trong khai thác CSDL

18



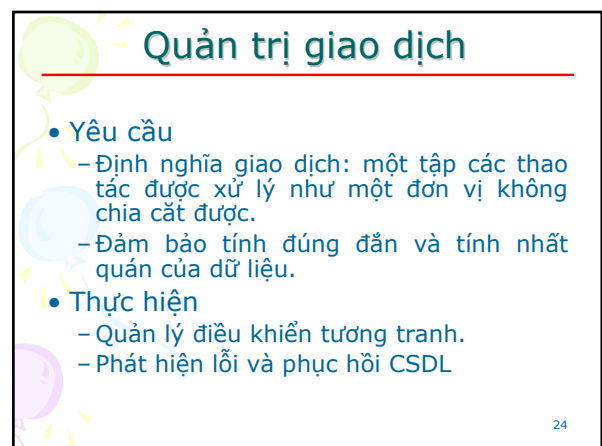
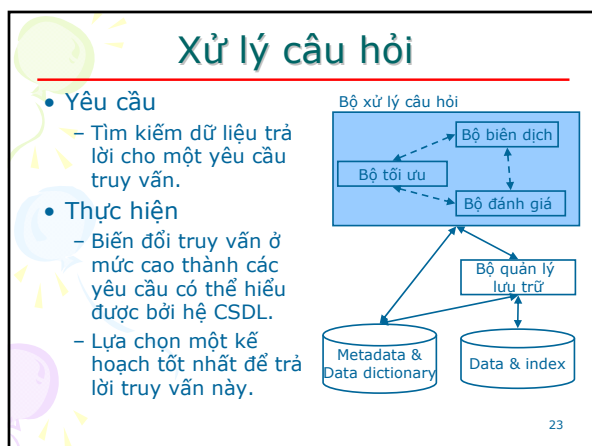
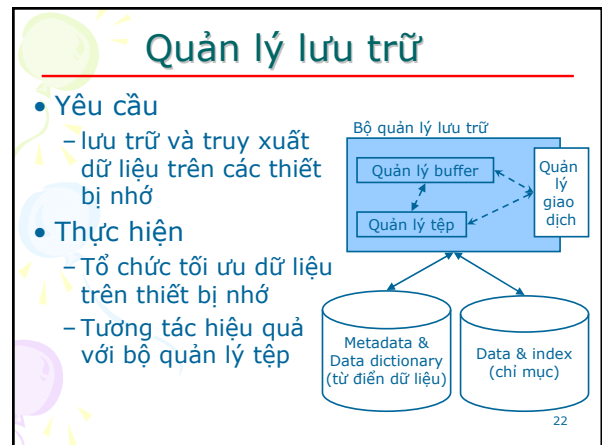
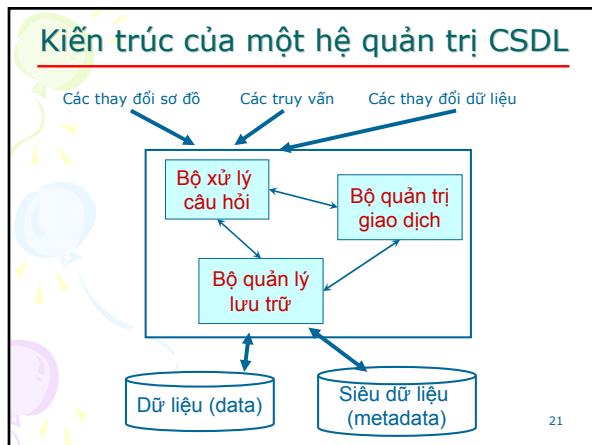
## Tương ứng 3 mức với ngôn ngữ Pascal

```

Type khách_hang = record
    ten:string;
    ngay_sinh:string;
    địa_chi:string;
end;
    
```

- Mức vật lý: một bản ghi khách\_hang được mô tả như một khối nhớ, chương trình dịch che dấu các chi tiết mức này đối với người lập trình.
- Mức logic: mỗi bản ghi được mô tả bởi một định nghĩa kiểu, người lập trình sử dụng ngôn ngữ lập trình làm việc tại mức trừu tượng này.
- Mức khung nhìn: người sử dụng máy tính thấy một tập các chương trình ứng dụng, che dấu những chi tiết về các kiểu dữ liệu

20



## Người dùng

- **Người thiết kế và cài đặt hệ QTCSDL:** chịu trách nhiệm thiết kế và cài đặt các module của hệ QTCSDL và các giao diện dưới hình thức các gói phần mềm
- **Người phát triển công cụ:** chịu trách nhiệm thiết kế và cài đặt các gói phần mềm hỗ trợ cho việc thiết kế, sử dụng cũng như tăng cường hiệu năng của các hệ CSDL.

25

## Người dùng

- **Người phân tích hệ thống và phát triển ứng dụng:** chịu trách nhiệm xác định yêu cầu của người dùng cuối, xác định các giao dịch cần thiết để đáp ứng các yêu cầu người dùng. Người lập trình ứng dụng cài đặt những yêu cầu này trong chương trình, kiểm thử, gỡ rối, lập tài liệu cho chương trình
- **Người thiết kế CSDL:** chịu trách nhiệm xác định dữ liệu lưu trữ trong CSDL và cấu trúc biểu diễn và lưu trữ những dữ liệu này

26

## Người dùng

- **Người sử dụng cuối:** là người khai thác các hệ CSDL
- **Người quản trị CSDL:** chịu trách nhiệm cho phép truy nhập CSDL, điều phối và kiểm tra sử dụng CSDL, quản lý tài nguyên phần cứng và phần mềm khi cần thiết
- **Người bảo trì hệ thống:** là những người quản trị hệ thống chịu trách nhiệm việc hoạt động và bảo trì môi trường (phần cứng và phần mềm) cho hệ CSDL

27

## Nội dung chương này

- 1.1 Các hệ thống xử lý tệp truyền thống và những hạn chế của nó.
- 1.2 Các hệ CSDL: khái niệm, khả năng, kiến trúc, người dùng của một hệ quản trị CSDL.
- 1.3 Sự phân loại các hệ CSDL.



28

## 1.3 Phân loại các hệ CSDL

- Mô hình dữ liệu
  - Mạng vs. phân cấp vs. quan hệ vs. hướng đối tượng vs. ...
- Số người sử dụng
  - Một người dùng vs. nhiều người dùng
- Tính phân tán của CSDL
  - Tập trung vs. Phân tán
- Tính thống nhất của dữ liệu
  - Đồng nhất vs. Không đồng nhất
- ...

29

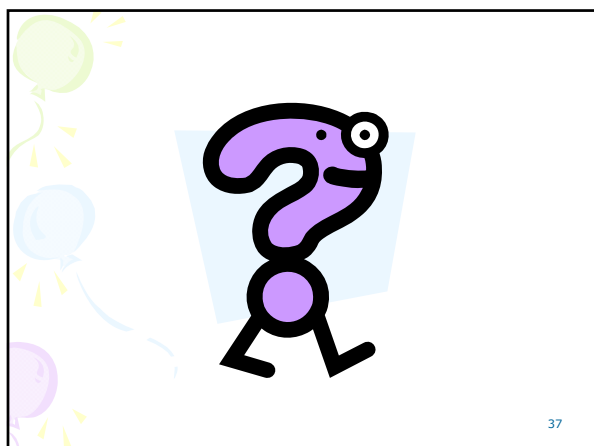
## Các hệ CSDL tập trung

- Hệ CSDL cá nhân: một người sử dụng đơn lẻ vừa thiết kế, tạo lập CSDL, cập nhật, bảo trì dữ liệu, lập và hiển thị báo cáo. đảm nhiệm vai trò: người quản trị CSDL, người viết chương trình ứng dụng, end-user.
- Hệ CSDL trung tâm: dữ liệu được lưu trữ trên một máy tính trung tâm.
- Hệ CSDL khách-chủ:
  - Các máy tính trung tâm lớn đặt so với các máy nhỏ và máy trạm.
  - Các ứng dụng máy khách truy nhập dữ liệu được quản lý bởi máy chủ.

30







## Lời hay ý đẹp

---

**Điều chúng ta biết chỉ là một giọt nước,  
điều không biết mênh mông như đại dương**

Einstein

38

# Các mô hình dữ liệu

Nguyễn Hồng Phương

[phuongnh-fit@mail.hut.edu.vn](mailto:phuongnh-fit@mail.hut.edu.vn)

Bộ môn Hệ thống thông tin  
Viện Công nghệ thông tin và Truyền thông  
Đại học Bách Khoa Hà Nội

1

## Nội dung

- Tổng quan về mô hình dữ liệu
- Mô hình phân cấp
- Mô hình mạng
- Mô hình quan hệ
- Mô hình thực thể liên kết
- Mô hình hướng đối tượng
- Đánh giá, bài tập

2

## Tổng quan về mô hình dữ liệu

- Mô hình dữ liệu [Codd, 1980] gồm:
  - Một tập hợp các cấu trúc của dữ liệu
  - Một tập hợp các phép toán để thao tác với các dữ liệu
  - Một tập hợp các ràng buộc về dữ liệu
- Mô hình dữ liệu là một tập hợp các khái niệm dùng để mô tả:
  - Dữ liệu
  - Ngữ nghĩa của dữ liệu
  - Các mối quan hệ trong dữ liệu
  - Các ràng buộc dữ liệu

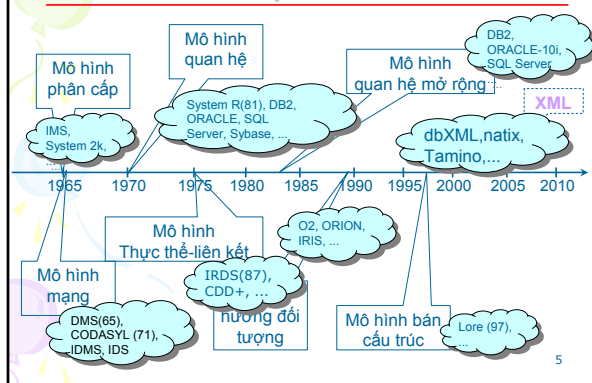
3

## Tổng quan...(tiếp)

- Nhiều mô hình còn bao gồm cả một tập các phép toán để thao tác các dữ liệu
- Mô hình thuộc dạng ngữ nghĩa: tập trung về ngữ nghĩa của dữ liệu như mô hình thực thể liên kết, sử dụng để hỗ trợ người dùng có cái nhìn khái quát về dữ liệu
- Mô hình thuộc dạng khái niệm: tập trung vào cách thức tổ chức dữ liệu tại mức khái niệm như mô hình mạng, mô hình liên kết, mô hình quan hệ, độc lập với DBMS và hệ thống phần cứng để cài đặt cơ sở dữ liệu

4

## Vài nét về lịch sử



5

## Một vài mô hình dữ liệu

- Mô hình phân cấp
- Mô hình mạng
- Mô hình quan hệ
- Mô hình thực thể liên kết
- Mô hình hướng đối tượng
- Mô hình bán cấu trúc
- Mô hình dữ liệu của XML

6

## Đặt vấn đề

- Đặc điểm của các mô hình dữ liệu?
- Sự khác nhau giữa các mô hình dữ liệu?
- Các mô hình dữ liệu phổ biến ngày nay

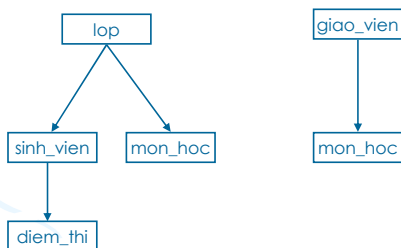
7

## Mô hình dữ liệu phân cấp (Hierarchical data model)

- Ra đời những năm 60-65
- Biểu diễn bằng cây
  - Quan hệ cha-con
  - Mỗi nút có 1 cha duy nhất
  - 1 CSDL = 1 tập các cây = 1 rừng
- Các khái niệm cơ bản
  - Bản ghi
  - Mốc nối
  - Các phép toán: GET, GET UNIQUE, GET NEXT, GET NEXT WITHIN PARENT,...

8

## Mô hình dữ liệu phân cấp – Ví dụ



9

## Mô hình dữ liệu phân cấp

- Ưu điểm
  - Dễ xây dựng và thao tác
  - Tương thích với các lĩnh vực tổ chức phân cấp
  - Ngôn ngữ thao tác đơn giản: duyệt cây.
- Nhược điểm:
  - Sự lặp lại của các kiểu bản ghi → dữ liệu dư thừa và không nhất quán.
    - Giải pháp: bản ghi ảo
  - Hạn chế trong biểu diễn ngữ nghĩa của các mốc nối giữa các bản ghi (chỉ cho phép quan hệ 1-n)

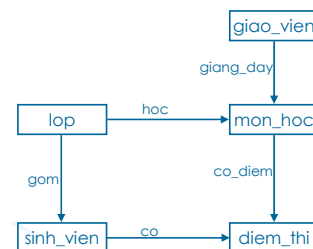
10

## Mô hình dữ liệu mạng (Network data model)

- Sự ra đời
  - Sử dụng phổ biến từ những năm 60, được định nghĩa lại vào năm 71
- Biểu diễn bằng đồ thị có hướng
- Các khái niệm cơ bản
  - Tập bản ghi (record)
    - Kiểu bản ghi (record type)
    - Các trường (field)
  - Mốc nối
    - Tên của mốc nối
    - Chủ (owner) – thành viên (member): theo hướng của mốc nối
    - Kiểu mốc nối: 1-1, 1-n, đệ quy
  - Các phép toán
    - Duyệt: FIND, FIND member, FIND owner, FIND NEXT
    - Thủ tục: GET

11

## Mô hình dữ liệu mạng – Ví dụ



12

## Mô hình dữ liệu mạng

- Ưu điểm
  - Đơn giản
  - Có thể biểu diễn các ngữ nghĩa đa dạng với kiểu bản ghi và kiểu móc nối
  - Truy vấn thông qua phép duyệt đồ thị (navigation)
- Nhược điểm:
  - Số lượng các con trỏ lớn
  - Hạn chế trong biểu diễn ngữ nghĩa của các móc nối giữa các bản ghi

13

## Mô hình dữ liệu quan hệ

- Sự ra đời: vào năm 1970 [Codd, 1970]
- Dữ liệu được biểu diễn dưới dạng bảng
- Là mô hình dữ liệu khái niệm phổ biến cho đến tận thời điểm hiện tại
- Dựa trên lý thuyết toán học, đồng thời cũng gần với cấu trúc tệp và cấu trúc dữ liệu nên có hai loại thuật ngữ liên quan:
  - Thuật ngữ toán học: quan hệ, bộ, thuộc tính
  - Thuật ngữ hướng dữ liệu: bảng, bản ghi, trường

14

### Ví dụ mô hình dữ liệu quan hệ

MON_HOC		
maMH	tenmon	soHT
CNTT01	Nhập môn CSDL	4
CNTT02	Truyền DL và mạng	4
CNTT03	Phân tích và thiết kế hệ thống	4
HTTT01	Quản lý dự án	3

#### LOP

malop	lop	khoa	GVCN	loptruong
IT4	Tin 4	CNTT	Ng. V. Anh	Trần T. Bình
IT5	Tin 5	CNTT	Lê A. Văn	Ng. Đ. Trung
IT6	Tin 6	CNTT	Ng. T. Thảo	Trần M. Quế
IT7	Tin 7	CNTT	Ng. V. Quý	Ng. T. Phương

#### SINH VIEN

maSV	tenSV	ngaysinh	gt	diachi	malop
SV0011	Trần T. Bình	1/4/1981	0	21 T. Q. B	IT4
SV0025	Ng. Đ. Trung	3/2/1980	1	56 Đ. C. V	IT5
SV0067	Trần M. Quế	26/3/1982	0	45 H. B. T	IT6
SV0034	Ng. T. Phương	29/2/1980	0	86 L. T. N	IT7

## Mô hình dữ liệu quan hệ

- Các khái niệm cơ bản
  - Thuộc tính, miền thuộc tính
  - Quan hệ
  - Khóa

16

## Mô hình dữ liệu quan hệ

- **Thuộc tính** (~trường): là các đặc tính của một đối tượng
- Mỗi thuộc tính được xác định trên một miền giá trị nhất định gọi là **miền thuộc tính**
- Ví dụ:
  - Sinh viên (MãSV, TênSV, Năm sinh, Giới tính, Địa chỉ)
  - $\text{dom}(\text{MãSV}) = \{\text{char}(5)\}$
  - $\text{dom}(\text{TênSV}) = \{\text{char}(30)\}$
  - $\text{dom}(\text{Năm sinh}) = \{\text{date}\}$
  - $\text{dom}(\text{Giới tính}) = \{0, 1\}$
  - $\text{dom}(\text{Địa chỉ}) = \{\text{char}(50)\}$

17

## Mô hình dữ liệu quan hệ

- **Quan hệ** (~bảng): Cho  $n$  miền giá trị  $D_1, D_2, \dots, D_n$  không nhất thiết phân biệt,  $r$  là một quan hệ trên  $n$  miền giá trị đó nếu  $r$  là một tập các  $n$ -bộ  $(d_1, d_2, \dots, d_n)$  sao cho  $d_i \in D_i$
- Một **quan hệ** có thể được biểu diễn dưới dạng 1 **bảng** trong đó 1 **dòng** trong bảng tương đương với 1 **bộ**, một **cột** trong bảng tương đương với 1 **thuộc tính** của quan hệ
- **Bậc** của 1 quan hệ là số các thuộc tính trong quan hệ
- **Lực lượng** của 1 quan hệ là số các bộ trong quan hệ

18

## Mô hình dữ liệu quan hệ

- **Định nghĩa (tiếp):** Cho  $U = \{A_1, A_2, \dots, A_n\}$  là một tập hữu hạn các thuộc tính trong đó  $\text{dom}(A_i) = D_i$ ,  $r$  là quan hệ trên tập thuộc tính  $U$  ký hiệu là  $r(U)$  nếu:

$$r \subseteq D_1 \times D_2 \times \dots \times D_n$$

- $U$  được gọi là sơ đồ quan hệ (lược đồ quan hệ)

19

## Mô hình dữ liệu quan hệ

- Định nghĩa **khoá** của quan hệ  $r$  trên tập thuộc tính  $U = \{A_1, A_2, \dots, A_n\}$  là một tập  $K \subseteq U$  sao cho với bất kỳ 2 bộ  $t_1, t_2$  thuộc  $r$  đều tồn tại một thuộc tính  $A$  thuộc  $K$  mà  $t_1[A] \neq t_2[A]$
- Một quan hệ có thể có nhiều khoá
- Nếu  $K$  là khoá của  $r$  thì mọi  $K'$  sao cho  $K \subseteq K'$  đều là khoá của  $r$ .  $K'$  được gọi là **siêu khoá** của  $r$

20

## Mô hình dữ liệu quan hệ

- **Định nghĩa:**  $K$  là **khoá tối thiểu** của  $r$  nếu  $K$  là một khoá của  $r$  và bất kỳ tập con thực sự nào của  $K$  đều không phải là khoá của  $r$
- **Định nghĩa:** Một tập con  $K \subseteq U$  được gọi là **khoá ngoài** của quan hệ  $r(U)$  tham chiếu đến một quan hệ  $r'$  nếu  $K$  là khoá chính của  $r'$

21

## Mô hình dữ liệu quan hệ Ví dụ:

- Quan hệ: SinhViên(MãSV, TênSV, NămSinh, GiớiTinh, Lớp)

SV001	Nguyễn Văn An	1982	1	Tin 7
SV002	Nguyễn Văn An	1985	1	HTTT
SV003	Lê Văn Cường	1981	1	HTTT
SV004	Nguyễn Thùy Linh	1981	0	BK65
- Siêu khoá: {MãSV, HọTên};
- Khoá tối thiểu: {MãSV}; {HọTên, NămSinh}
- Khoá ngoài: TênLớp nếu coi nó là khoá chính của quan hệ Lớp

22

## Mô hình dữ liệu quan hệ - nhận xét

- Ưu điểm
  - Dựa trên lý thuyết tập hợp
  - Khả năng tối ưu hoá các xử lý phong phú
- Nhược điểm
  - Hạn chế trong biểu diễn ngữ nghĩa
  - Cấu trúc dữ liệu không linh hoạt

23

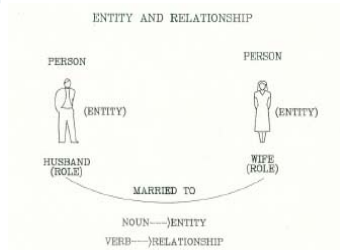
## Mô hình thực thể liên kết (Entity-Relationship data model)

- Cho phép mô tả các dữ liệu có liên quan trong một xí nghiệp trong thế giới thực dưới dạng các đối tượng và các mối quan hệ của chúng.
- Được sử dụng cho bước đầu thiết kế CSDL, làm nền tảng để ánh xạ sang một mô hình khái niệm nào đó mà Hệ quản trị CSDL sẽ sử dụng
- Trong mô hình thực thể liên kết, CSDL được mô hình hóa như là:
  - Một tập hợp các thực thể
  - Liên hệ giữa các thực thể này

24

## Mô hình thực thể liên kết Các khái niệm cơ bản

- Thực thể, tập thực thể
- Thuộc tính
- Khoá
- Liên kết, tập liên kết



25

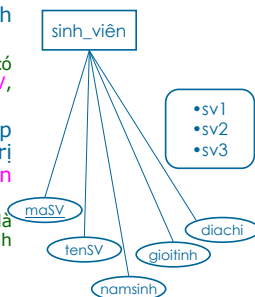
## Mô hình thực thể liên kết...

- **Thực thể**: một đối tượng trong thế giới thực, tồn tại độc lập và phân biệt được với các đối tượng khác
- **Tập thực thể**: một tập hợp các thực thể có tính chất giống nhau
- Ví dụ:
  - Thực thể: một sinh viên, một lớp
  - Tập thực thể: toàn thể sinh viên của 1 lớp, toàn thể các lớp của 1 khoa

26

## Mô hình thực thể liên kết...

- **Thuộc tính** là đặc tính của một tập thực thể
  - Tập thực thể **SinhViên** có các thuộc tính như: **TênSV**, **NămSinh**,...
- Mỗi thực thể trong tập thực thể có một giá trị đặc tính nằm trong **miền giá trị của thuộc tính**
  - Sinh viên 1 có: Họ tên là Nguyễn Hải Anh, Năm sinh 1980

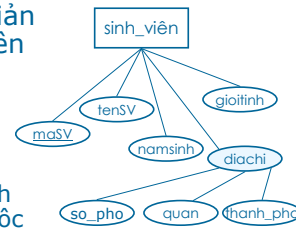


27

## Mô hình thực thể liên kết...

### Kiểu thuộc tính

- Thuộc tính đơn giản (thuộc tính nguyên tố)
  - có kiểu dữ liệu nguyên tố
- Thuộc tính phức
  - có kiểu phức, định nghĩa bởi các thuộc tính khác

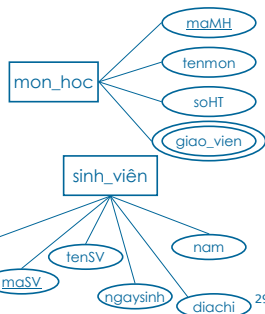


28

## Mô hình thực thể liên kết...

### Kiểu thuộc tính

- Thuộc tính đa giá trị
  - tương ứng với mỗi thực thể, có thể nhận nhiều giá trị
- Thuộc tính suy diễn
  - có thể tính toán được từ (các) thuộc tính khác



29

## Mô hình thực thể liên kết...

### Khoá

- Một hay một tập thuộc tính mà giá trị của chúng có thể xác định duy nhất một thực thể trong tập thực thể
  - Tập thực thể SinhViên có thể dùng MãSV làm khoá
- Khoá gồm nhiều thuộc tính thì gọi là **khoá phức**
- Một tập thực thể có thể có nhiều khoá nhưng chỉ một trong số các khoá được chọn làm **khoá chính**
- Trong sơ đồ ER, thuộc tính nào được chọn làm khoá chính sẽ được **gạch chân**

30

## Mô hình thực thể liên kết...

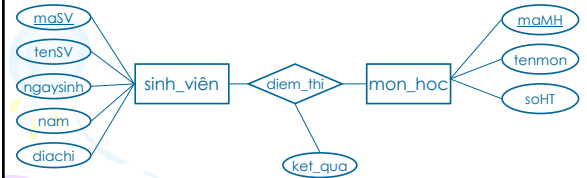
### Liên kết - Tập liên kết

- Một liên kết là một mối liên hệ có nghĩa giữa nhiều thực thể
  - Cho một thực thể SinhViên1 và LớpA, liên kết ThànhViên chỉ ra rằng SinhViên1 là 1 thành viên của LớpA
- Tập liên kết là một tập hợp các liên kết cùng kiểu
  - Giữa tập thực thể SinhViên và Lớp có 1 tập liên kết ThànhViên, chỉ ra rằng mỗi sinh viên đều là thành viên của 1 lớp nào đó
- Một liên kết có thể có thuộc tính

31

## Mô hình thực thể liên kết...

### Liên kết - Tập liên kết - Ví dụ:



32

## Mô hình thực thể liên kết...

### Ràng buộc của kết nối

- 1-1:** Liên kết 1 thực thể của một tập thực thể với nhiều nhất 1 thực thể của tập thực thể khác
- 1-n:** Liên kết 1 thực thể của một tập thực thể với nhiều thực thể của tập thực thể khác
- n-n:** Liên kết 1 thực thể của một tập thực thể với nhiều thực thể của tập thực thể khác và ngược lại
- đệ quy:** Liên kết giữa các thực thể cùng kiểu

33

## Cách lập sơ đồ thực thể - liên kết

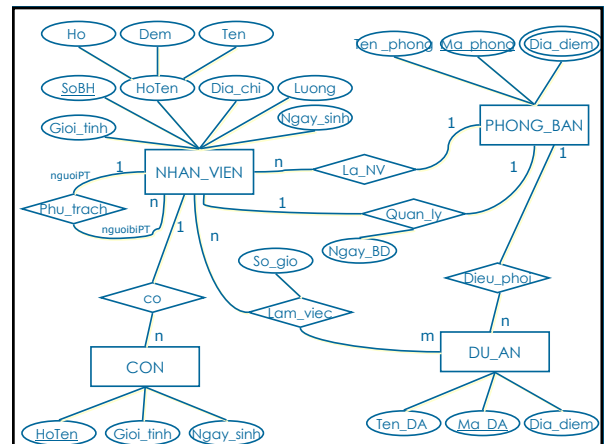
- Bước 1: Xác định các thực thể
- Bước 2: Xác định các liên kết giữa các thực thể
  - Bậc của liên kết
  - Ràng buộc (1-1, 1-n, n-n, đệ quy)

34

## Bài tập: Vẽ sơ đồ ER

- Bài toán: phân tích và thiết kế 1 CSDL gồm các thông tin trong 1 công ty (nhân viên, phòng ban, dự án)
  - Công ty được tổ chức bởi các phòng ban. Mỗi phòng ban có 1 tên duy nhất, 1 số duy nhất và 1 người quản lý (thời điểm bắt đầu công tác quản lý của người này cũng được lưu lại trong CSDL). Mỗi phòng ban có thể có nhiều trụ sở làm việc khác nhau.
  - Mỗi phòng ban điều phối một số dự án. Mỗi dự án có 1 tên và 1 mã số duy nhất, thực hiện tại một địa điểm duy nhất.
  - Các thông tin về nhân viên cần được quan tâm gồm: tên, số bảo hiểm, địa chỉ, lương, giới tính, ngày sinh. Mỗi nhân viên làm việc tại một phòng ban nhưng có thể tham gia nhiều dự án khác nhau. Những dự án này có thể được điều phối bởi các phòng ban khác nhau. Thông tin về số giờ làm việc trong từng dự án (theo tuần) cũng như người quản lý trực tiếp của các nhân viên cũng được lưu trữ.
  - Thông tin về con cái của từng nhân viên: tên, giới tính, ngày sinh.

35



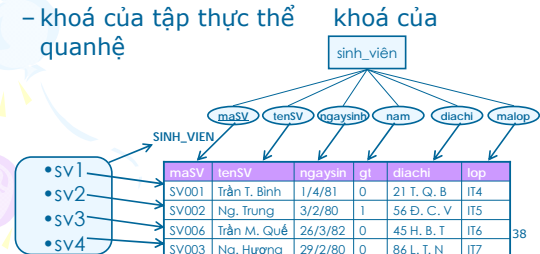
## Biến đổi sơ đồ thực thể liên kết sang sơ đồ quan hệ

- Biến đổi tập các thực thể
- Biến đổi các liên kết
- Các khoá của các sơ đồ quan hệ
- Các sơ đồ quan hệ với khoá chung

37

## Biến đổi các tập thực thể

- **Bước 1:** 1 tập thực thể → 1 quan hệ
  - thuộc tính thuộc tính (trường)
  - 1 thực thể 1 bộ
  - khoá của tập thực thể khoá của quan hệ



## Biến đổi các tập thực thể

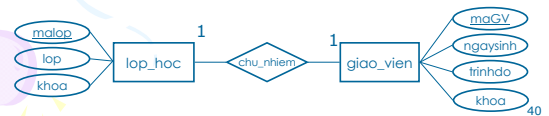
- **Bước 2:** 1 tập thực thể xác định từ tập thực thể khác (E) qua 1 liên kết → 1 quan hệ chứa khoá của E:  
LOPTRUONG(maSV)



39

## Biến đổi các liên kết

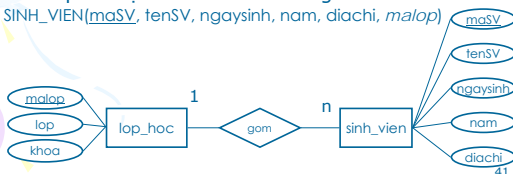
- **Bước 3:** Liên kết 1-1
  - Thêm 1 quan hệ mới xác định bởi các thuộc tính nằm trong khoá của các thực thể có liên quan  
CHU\_NHIEM\_LOP(malop, maGV)
  - hoặc
  - Dùng khoá ngoài  
LOP\_HOC(malop, lop, khoa, maGV)



40

## Biến đổi các liên kết (tiếp)

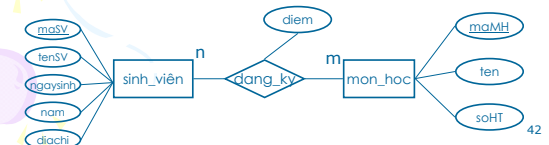
- **Bước 4:** Liên kết 1-n
  - Thêm 1 quan hệ mới xác định bởi các thuộc tính nằm trong khoá của các thực thể có liên quan  
SINHVIEN\_LOP(malop, maSV)
  - hoặc
  - Dùng khoá ngoài: thêm khoá chính của quan hệ bên 1 vào quan hệ bên n làm khoá ngoài  
SINH\_VIEN(maSV, tenSV, ngaysinh, nam, diachi, malop)



41

## Biến đổi các liên kết (tiếp)

- **Bước 5:** Liên kết n-n
  - Thêm 1 quan hệ mới xác định bởi các thuộc tính nằm trong khoá của các thực thể có liên quan và các thuộc tính của liên kết  
DANG\_KY(maSV, maMH, diem)



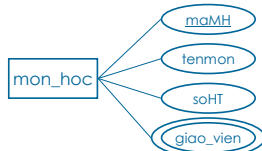
42



## Thuộc tính đa trị

- **Bước 6:** Với mỗi thuộc tính đa trị
- Thêm 1 quan hệ mới xác định bởi thuộc tính đa trị và khoá của tập thực thể tương ứng

MH\_GV(maMH, giao\_vien)



43

## Mô hình dữ liệu hướng đối tượng (Object-oriented data model)

- Sự ra đời
  - Khoảng đầu những năm 90
- Biểu diễn: sơ đồ lớp
- Các khái niệm cơ bản
  - **Đối tượng:** một đối tượng trong thế giới thực, được xác định bởi một định danh duy nhất
  - **Thuộc tính:** biểu diễn một đặc tính của đối tượng,
  - **Phương thức:** thao tác được thực hiện trên đối tượng.
    - Tất cả các truy nhập vào thuộc tính của đối tượng đều phải được thực hiện thông qua các phương thức này.
  - **Lớp:** một cách thức để khai báo một tập các đối tượng có chung một tập thuộc tính và phương thức

44

## Mô hình dữ liệu hướng đối tượng Ví dụ:

```

class sinh_vien {
    string maSV;
    string tenSV;
    date ngaysinh;
    boolean nam;
    string diachi;
    string lop;

    string ten();
    string ngay_sinh();
    string dia_chi();
    string lop();
    void gan_DC(string DC_moi);
    void gan_lop(string lop);
}
  
```

45

## Mô hình dữ liệu hướng đối tượng Nhận xét:

- Ưu điểm
  - Cho phép định nghĩa kiểu đối tượng phức tạp
  - Tính chất: bao đóng (*encapsulation*), kế thừa (*heritage*), đa hình (*polymorphism*)
- Nhược điểm
  - Cấu trúc lưu trữ phức tạp và có thể sử dụng nhiều con trỏ
  - Khả năng tối ưu hoá các xử lý bị hạn chế trong nhiều trường hợp

46

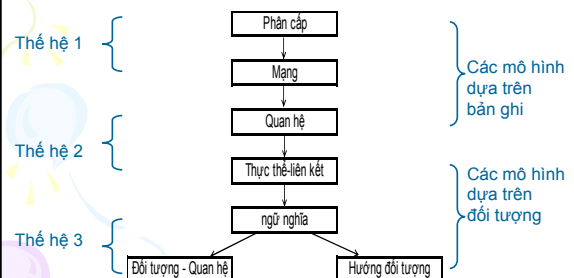
## So sánh và đánh giá

Nhắc lại: Mô hình dữ liệu là một tập hợp các khái niệm dùng để mô tả cấu trúc của một CSDL

	Mô hình mạng	Mô hình phân cấp	Mô hình quan hệ	Mô hình TT-LK	Mô hình HDT
<b>biểu diễn ngữ nghĩa DL</b>	hạn chế	hạn chế	<i>tương đối đa dạng</i>	đa dạng	đa dạng
<b>lưu trữ DL</b>	s/d nhiều con trỏ	dữ liệu lặp lại	đễ dàng và hiệu quả	khó lưu trữ	<i>cấu trúc phức tạp</i>
<b>khả năng truy vấn</b>	đơn giản	đơn giản	đa dạng		đa dạng
<b>hiệu quả của truy vấn</b>	ít khả năng tối ưu	ít khả năng tối ưu	tối ưu hoá tốt	không được xem xét (không hiệu quả)	<i>không h/q khi s/d nhiều con trỏ</i>

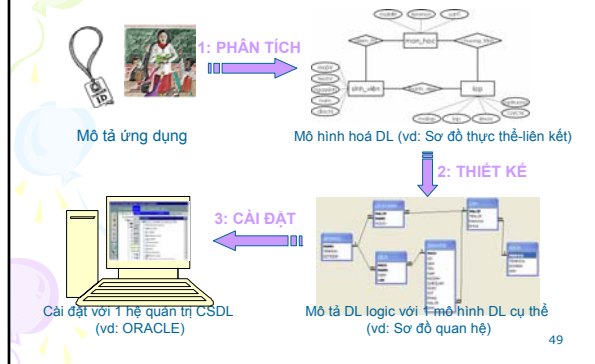
47

## Phân loại các mô hình



48

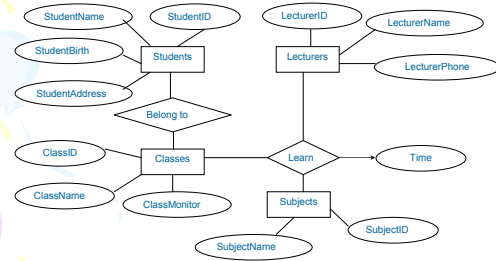
## Các bước xây dựng một hệ CSDL



49

## Bài tập

- Cho sơ đồ thực thể liên kết bên dưới, hãy biến đổi sang mô hình quan hệ:



50

## Lời giải

- Biến đổi các tập thực thể và các quan hệ thành các bảng:
  - Lecturers(LecturerID, LecturerName, LecturerPhone)
  - Students(StudentID, StudentName, StudentBirth, StudentAddress)
  - Classes(ClassID, ClassName, ClassMonitor)
  - Subjects(SubjectID, SubjectName)
  - Belongto(StudentID, ClassID)
  - Learn(LecturerID, ClassID, SubjectID, Time)

51

## Lời giải (tiếp)

- Cải tiến thiết kế: Bảng Students và bảng Belongto có cùng khóa (StudentID), ta nên kết hợp chúng lại:
  - Students'(StudentID, StudentName, StudentBirth, StudentAddress, ClassID)

52

## Lời hay ý đẹp

**Trong 10 lần thành công thì có tới 9 lần thành công nhờ sự hăng hái và niềm tin trong công việc**

Teewilson

53

# Ngôn ngữ định nghĩa và thao tác dữ liệu đối với mô hình quan hệ

Nguyễn Hồng Phương

[phuongngh-fit@mail.hut.edu.vn](mailto:phuongngh-fit@mail.hut.edu.vn)

Bộ môn Hệ thống thông tin  
Viện Công nghệ thông tin và Truyền thông  
Đại học Bách Khoa Hà Nội

1

## Nội dung

- Các cách tiếp cận đối với thiết kế ngôn ngữ của CSDL quan hệ
  - Giới thiệu một số ngôn ngữ và phân loại
    - So sánh và đánh giá
- Một số ngôn ngữ dữ liệu mức cao
  - QBE (Query By Example)
  - SQL (Structured Query Language)
- Kết luận

2

## CSDL ví dụ 1

Student			Takes		Enrol	
Id	Name	Suburb	SID	SNO	SID	Course
1108	Robert	Kew	1108	21	3936	101
3936	Glen	Bundoora	1108	23	1108	113
8507	Norman	Bundoora	8507	23	8507	101
8452	Mary	Balwyn	8507	29		

Course			Subject		
No	Name	Dept	No	Name	Dept
113	BCS	CSCE	21	Systems	CSCE
101	MCS	CSCE	23	Database	CSCE
			29	VB	CSCE
			18	Algebra	Maths

3

## CSDL ví dụ 2

Supplier				SupplyProduct		
SID	SNAME	SIZE	CITY	SID	PID	QUANTITY
S1	Dustin	100	London	S1	P1	500
S2	Rusty	70	Paris	S1	P2	400
S3	Lubber	120	London	S1	P4	100
S4	M&M	60	NewYork	S2	P3	250
S5	MBI	1000	NewOrlean	S2	P4	50
S6	Panda	150	London	S3	P1	300

Product		
PID	PNAME	COLOR
P1	Screw	red
P2	Screw	green
P3	Nut	red
P4	Bolt	blue
P5	Plier	green
P6	Scissors	blue

4

## Đặt vấn đề: các câu hỏi

- Tìm tên của các sinh viên nào sống ở Bundoora
  - Tìm các bộ của bảng Student có Suburb = Bundoora
  - Đưa ra các giá trị của thuộc tính Name của các bộ này

Student		
Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

5

## Câu hỏi (tiếp)

- Tìm các sinh viên đăng ký khoá học có mã số 113
  - Tìm các giá trị SID trong bảng Enrol có Course tương ứng là 113
  - Đưa ra các bộ của bảng Student có SID trong các giá trị tìm thấy ở trên

Student		
Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Enrol	
SID	Course
3936	101
1108	113
8507	101

Course		
No	Name	Dept
113	BCS	CSCE
101	MCS	CSCE

6

## Phân loại các ngôn ngữ truy vấn

- Ngôn ngữ đại số
  - 1 câu hỏi = 1 tập các phép toán trên các quan hệ
  - Được biểu diễn bởi một biểu thức đại số (quan hệ)
- Ngôn ngữ tính toán vị từ
  - 1 câu hỏi = 1 mô tả của các bộ mong muốn
  - Được đặc tả bởi một vị từ mà các bộ phải thỏa mãn
  - Phân biệt 2 lớp:
    - ngôn ngữ tính toán vị từ biến bộ
    - ngôn ngữ tính toán vị từ biến miền

7

## Ngôn ngữ đại số quan hệ

8

## Tổng quan

- Gồm các phép toán tương ứng với các thao tác trên các quan hệ
- Mỗi phép toán
  - Đầu vào: một hay nhiều quan hệ
  - Đầu ra: một quan hệ
- Biểu thức đại số quan hệ = chuỗi các phép toán
- Kết quả thực hiện một biểu thức đại số là một quan hệ
- Được cài đặt trong phần lớn các hệ CSDL hiện nay

9

## Phân loại các phép toán đại số quan hệ

- Phép toán quan hệ
  - Phép chiếu (*projection*)
  - Phép chọn (*selection*)
  - Phép kết nối (*join*)
  - Phép chia (*division*)
- Phép toán tập hợp
  - Phép hợp (*union*)
  - Phép giao (*intersection*)
  - Phép trừ (*difference*)
  - Phép tích đề-các (*cartesian product*)

10

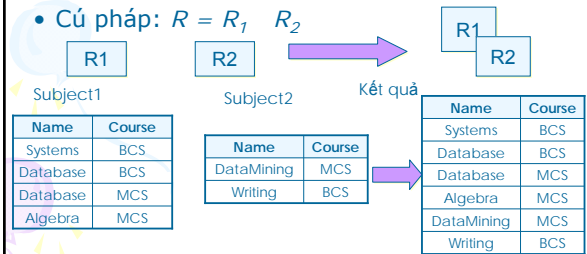
## Phép toán tập hợp

- Định nghĩa: Quan hệ khả hợp
  - 2 quan hệ  $r$  và  $s$  được gọi là khả hợp nếu chúng được xác định trên cùng 1 miền giá trị
  - $r$  xác định trên  $D_1 \times D_2 \times \dots \times D_n$
  - $s$  xác định trên  $D'_1 \times D'_2 \times \dots \times D'_m$
  - $\rightarrow D_i = D'_i$  và  $n=m$

11

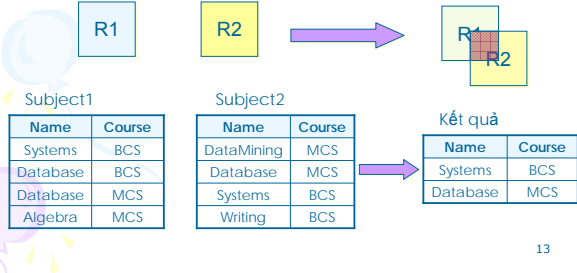
## Phép hợp

- Đ/n: gồm các bộ thuộc ít nhất 1 trong 2 quan hệ đầu vào
- 2 quan hệ đầu vào phải là khả hợp
- Cú pháp:  $R = R_1 \cup R_2$



## Phép giao

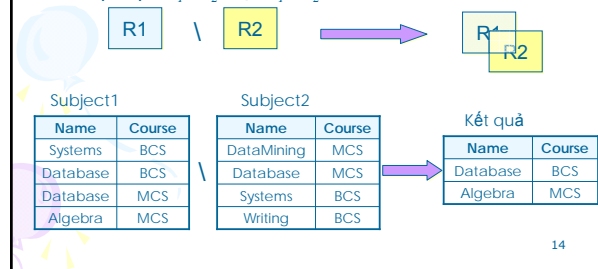
- Đ/n: gồm các bộ thuộc cả hai quan hệ đầu vào
- Cú pháp:  $R_1 \cap R_2$



13

## Phép trừ

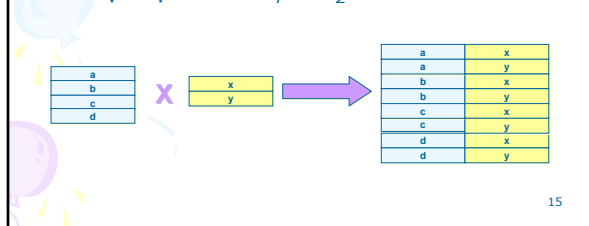
- Đ/n: gồm các bộ thuộc quan hệ thứ nhất nhưng không thuộc quan hệ thứ hai
- 2 quan hệ phải là khả hợp
- Cú pháp:  $R_1 \setminus R_2$  hoặc  $R_1 - R_2$



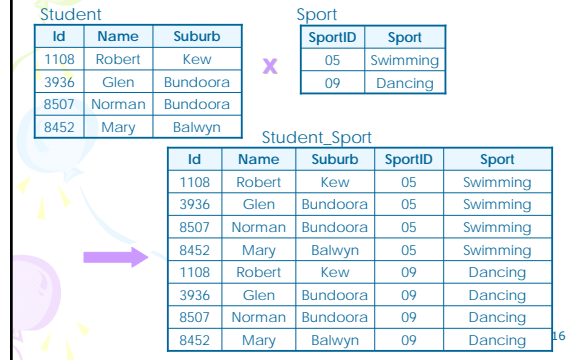
14

## Phép tích Đề-các

- Đ/n: là kết nối giữa từng bộ của quan hệ thứ nhất với mỗi bộ của quan hệ thứ hai
- Cú pháp:  $R = R_1 \times R_2$



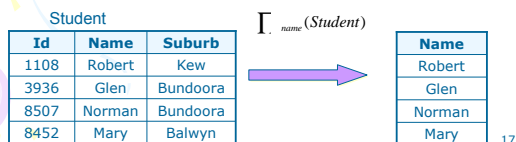
## Ví dụ phép tích Đề-các



## Phép chiếu

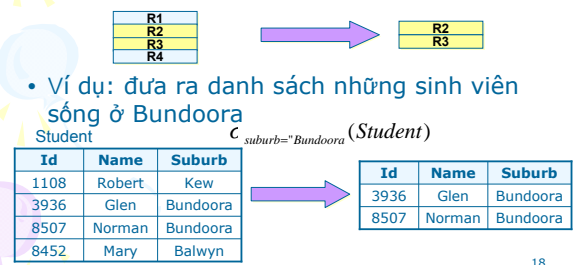
- Đ/n: Lựa chọn một số thuộc tính từ một quan hệ.
- Cú pháp:  $\pi_{A_1, A_2, \dots}(R)$

❖ Ví dụ: đưa ra danh sách tên của tất cả các sinh viên



## Phép chọn

- Đ/n: Lựa chọn các bộ trong một quan hệ thỏa mãn điều kiện cho trước.
- Cú pháp:  $\sigma_{\langle \text{condition} \rangle}(R)$



## Phép chọn - Điều kiện ?

- Điều kiện chọn còn gọi là biểu thức chọn.
- Biểu thức chọn F: một tổ hợp logic của các toán hạng. Mỗi toán hạng là một phép so sánh đơn giản giữa 2 biến là hai thuộc tính hoặc giữa 1 biến là 1 thuộc tính và 1 giá trị hằng.
  - Các phép so sánh trong F:  $=, <, >, \leq, \geq$
  - Các phép toán logic trong F:  $\wedge, \vee, \neg$

19

## Ví dụ: chọn và chiếu

- Đưa ra tên của các sinh viên sống ở Bundoora

$$\pi_{name}(\sigma_{suburb='Bundoora'} Student)$$

Student				
Id	Name	Suburb		
1108	Robert	Kew		
3936	Glen	Bundoora		
8507	Norman	Bundoora		
8452	Mary	Balwyn		

Name
Glen
Norman

20

## Phép kết nối (join) 2 quan hệ r và s

- Khái niệm ghép bộ:  $u = (a_1, \dots, a_n); v = (b_1, \dots, b_m)$   
 $(u, v) = (a_1, \dots, a_n, b_1, \dots, b_m)$
- Phép kết nối 2 quan hệ thực chất là phép ghép các cặp bộ của 2 quan hệ thỏa mãn 1 điều kiện nào đó trên chúng.
- Biểu thức kết nối là phép hội của các toán hạng, mỗi toán hạng là 1 phép so sánh đơn giản giữa 1 thuộc tính của quan hệ r và 1 thuộc tính của quan hệ s.
- Cú pháp:  $R_1 \bowtie_{<join\_condition>} R_2$

a	r		
b	r	x	
c	v		

r	x		
s	v		
t	z		

a	r	x	
b	r	x	

21

## Phép kết nối - Ví dụ:

- Đưa ra danh sách các sinh viên và khoá học

$$Student \bowtie_{Id=SID} Enrol$$

Student				
Id	Name	Suburb		
1108	Robert	Kew		
3936	Glen	Bundoora		
8507	Norman	Bundoora		
8452	Mary	Balwyn		

Enrol			
SID	Course		
3936	101		
1108	113		
8507	101		

Kết quả				
SID	Id	Name	Suburb	Course
1108	1108	Robert	Kew	113
3936	3936	Glen	Bundoora	101
8507	8507	Norman	Bundoora	101

22

## Phép kết nối bằng-kết nối tự nhiên

- Định nghĩa: Nếu phép so sánh trong điều kiện kết nối là phép so sánh bằng thì kết nối gọi là **kết nối bằng**.
- Định nghĩa: Phép kết nối bằng trên các thuộc tính cùng tên của 2 quan hệ và sau khi kết nối 1 thuộc tính trong 1 cặp thuộc tính trùng tên đó sẽ bị loại khỏi quan hệ kết quả thì phép kết nối gọi là **kết nối tự nhiên**.
- Cú pháp phép kết nối tự nhiên:  $R_1 * R_2$

23

## Phép kết nối tự nhiên - Ví dụ:

Takes			
SID	SNO		
1108	21		
1108	23		
8507	23		
8507	29		

Enrol			
SID	Course		
3936	101		
1108	113		
8507	101		

SID	SNO	Course
1108	21	113
1108	23	113
8507	23	101
8507	29	101

24

## Ví dụ: chọn, chiếu, kết nối

- Đưa ra tên của các sinh viên sống ở Bundoora và mã khoá học mà sinh viên đó đăng ký:

$\pi_{name, Course} (\sigma_{suburb='Bundoora'} (Student \bowtie_{id=SID} Enrol))$

Student	ID	Name	Suburb
	1108	Robert	Kew
	3936	Glen	Bundoora
	8507	Norman	Bundoora
	8452	Mary	Balwyn

Enrol	SID	Course
	3936	101
	1108	113
	8507	101

Kết quả	Name	Course
	Glen	101
	Norman	101

25

## Phép kết nối ngoài

- Phép kết nối ngoài trái

a	r	r	x	a	r	x
b	r	s	y	b	r	x
c	v	t	z	c	v	null

- Phép kết nối ngoài phải

a	r	r	x	a	r	x
b	r	s	y	b	r	x
c	v	t	z	null	s	y
				null	t	z

26

## Phép kết nối ngoài - Ví dụ:

- Đưa ra danh sách các sinh viên và mã khoá học mà sinh viên đó đăng ký nếu có

Student	ID	Name	Suburb
	1108	Robert	Kew
	3936	Glen	Bundoora
	8507	Norman	Bundoora
	8452	Mary	Balwyn

Enrol	SID	Course
	3936	101
	1108	113
	8507	101

Kết quả	ID	Name	Suburb	Course
	1108	Robert	Kew	113
	3936	Glen	Bundoora	101
	8507	Norman	Bundoora	101
	8452	Mary	Balwyn	null

27

## Phép chia

- Định nghĩa: Phép chia giữa 1 quan hệ  $r$  bậc  $n$  và quan hệ  $s$  bậc  $m$  ( $m < n$ ) với sơ đồ quan hệ của  $s$  là tập con của sơ đồ quan hệ của  $r$  là một tập các  $(n-m)$ -bộ sao cho khi ghép mọi bộ thuộc  $s$  với  $t$  thì ta đều có một bộ thuộc  $r$

- Cú pháp:  $R = R_1 : R_2$

$$r \div s = \{ t \mid \forall v \in s \Rightarrow (t, v) \in r \}$$

28

## Phép chia (tiếp)

a	x	:	x	$\rightarrow$	a
a	y		z		
a	z				
b	x				
c	y				

- Ví dụ: Đưa ra môn học được dạy ở tất cả các khoá học

Subject	Name	Course
	Systems	BCS
	Database	BCS
	Database	MCS
	Algebra	MCS

Course	Course
	BCS
	MCS

Kết quả	Name
	Database

29

## Luyện tập

- Phép hợp (Union)

$$r \cup s = \{ t \mid t \in r \vee t \in s \}$$

Ví dụ:

r	(A B C)	s	(A B C)	$r \cup s$	(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>1</sub> b <sub>1</sub> c <sub>1</sub>
	a <sub>1</sub> b <sub>1</sub> c <sub>2</sub>		a <sub>1</sub> b <sub>2</sub> c <sub>1</sub>		a <sub>1</sub> b <sub>1</sub> c <sub>2</sub>
	a <sub>1</sub> b <sub>2</sub> c <sub>2</sub>		a <sub>1</sub> b <sub>2</sub> c <sub>2</sub>		a <sub>1</sub> b <sub>2</sub> c <sub>1</sub>
	a <sub>2</sub> b <sub>2</sub> c <sub>2</sub>				a <sub>1</sub> b <sub>2</sub> c <sub>2</sub>
	a <sub>3</sub> b <sub>2</sub> c <sub>2</sub>				a <sub>2</sub> b <sub>2</sub> c <sub>2</sub>
					a <sub>3</sub> b <sub>2</sub> c <sub>2</sub>

30

## Luyện tập

### • Phép giao (intersection)

$$r \cap s = \{ t \mid t \in r \wedge t \in s \}$$

Ví dụ:

r	(A	B	C)	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>1</sub>	b <sub>1</sub>	c <sub>2</sub>		a <sub>1</sub>	b <sub>2</sub>	c <sub>1</sub>
	a <sub>1</sub>	b <sub>2</sub>	c <sub>2</sub>		a <sub>1</sub>	b <sub>2</sub>	c <sub>2</sub>
	a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>				
	a <sub>3</sub>	b <sub>2</sub>	c <sub>2</sub>				

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

31

## Luyện tập

### • Phép trừ (minus)

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

Ví dụ:

r	(A	B	C)	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>1</sub>	b <sub>1</sub>	c <sub>2</sub>		a <sub>1</sub>	b <sub>2</sub>	c <sub>1</sub>
	a <sub>1</sub>	b <sub>2</sub>	c <sub>2</sub>		a <sub>1</sub>	b <sub>2</sub>	c <sub>2</sub>
	a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>				
	a <sub>3</sub>	b <sub>2</sub>	c <sub>2</sub>				

r - s = t	(A	B	C)
	a <sub>1</sub>	b <sub>1</sub>	c <sub>2</sub>
	a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>
	a <sub>3</sub>	b <sub>2</sub>	c <sub>2</sub>

32

## Luyện tập

### • Phép tích Đề - Các (Cartesian Product)

$$r \times s = \{ t \mid t = (a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m) \wedge (a_1, a_2, \dots, a_n) \in r \wedge (b_1, b_2, \dots, b_m) \in s \}$$

Ví dụ:

r	(A	B	C)	s	(D	E)	r × s = p	(A	B	C	D	E)
	a <sub>1</sub>	b <sub>1</sub>	1		1	e <sub>1</sub>		a <sub>1</sub>	b <sub>1</sub>	1	1	e <sub>1</sub>
	a <sub>1</sub>	b <sub>1</sub>	2		2	e <sub>2</sub>		a <sub>1</sub>	b <sub>1</sub>	1	2	e <sub>2</sub>
	a <sub>1</sub>	b <sub>2</sub>	3		3	e <sub>3</sub>		a <sub>1</sub>	b <sub>1</sub>	1	3	e <sub>3</sub>
	a <sub>2</sub>	b <sub>2</sub>	2					a <sub>2</sub>	b <sub>2</sub>	2	1	e <sub>1</sub>
	a <sub>2</sub>	b <sub>2</sub>	2					a <sub>2</sub>	b <sub>2</sub>	2	2	e <sub>2</sub>
	a <sub>2</sub>	b <sub>2</sub>	2					a <sub>2</sub>	b <sub>2</sub>	2	3	e <sub>3</sub>
	a <sub>3</sub>	b <sub>3</sub>	3					a <sub>3</sub>	b <sub>3</sub>	3	1	e <sub>1</sub>
	a <sub>3</sub>	b <sub>3</sub>	3					a <sub>3</sub>	b <sub>3</sub>	3	2	e <sub>2</sub>
	a <sub>3</sub>	b <sub>3</sub>	3					a <sub>3</sub>	b <sub>3</sub>	3	3	e <sub>3</sub>

33

## Luyện tập

### • Phép chiếu (Projection)

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

Ví dụ:

$$X = \{ A, B \}; Y = \{ C \}$$

r	(A	B	C)	$\Pi_X(r) = s_1$	(A	B)	$\Pi_Y(r) = s_2$	(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>1</sub>	b <sub>1</sub>	c <sub>2</sub>		a <sub>1</sub>	b <sub>2</sub>		c <sub>1</sub>
	a <sub>1</sub>	b <sub>2</sub>	c <sub>2</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>		a <sub>3</sub>	b <sub>2</sub>		c <sub>1</sub>
	a <sub>3</sub>	b <sub>2</sub>	c <sub>2</sub>					c <sub>1</sub>

34

## Luyện tập

### • Phép chọn (Selection)

$$\sigma_F(r) = \{ t \mid t \in r \wedge F(t) = \text{đúng} \}$$

Ví dụ:

r	(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>2</sub>
	a <sub>1</sub>	b <sub>2</sub>	c <sub>2</sub>
	a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>
	a <sub>3</sub>	b <sub>2</sub>	c <sub>2</sub>

$\sigma_{A=a1}(r) = r_1$	(A	B	C)	$\sigma_{A=a1 \wedge C=c2}(r) = r_2$	(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>2</sub>
	a <sub>1</sub>	b <sub>2</sub>	c <sub>2</sub>		a <sub>1</sub>	b <sub>2</sub>	c <sub>2</sub>
	a <sub>1</sub>	b <sub>1</sub>	c <sub>2</sub>				

35

## Luyện tập

### • Phép kết nối (join)

$$r \bowtie s = \{ t \mid t = (u, v) \wedge u \in r \wedge v \in s \wedge F(t) = \text{đúng} \}$$

Ví dụ:

r	(A	B	C)	s	(D	E)	$r \bowtie_F s = k$	(A	B	C	D	E)
	a <sub>1</sub>	b <sub>1</sub>	1		1	e <sub>1</sub>		a <sub>1</sub>	b <sub>1</sub>	1	1	e <sub>1</sub>
	a <sub>1</sub>	b <sub>1</sub>	2		2	e <sub>2</sub>		a <sub>1</sub>	b <sub>1</sub>	1	2	e <sub>2</sub>
	a <sub>1</sub>	b <sub>1</sub>	3		3	e <sub>3</sub>		a <sub>1</sub>	b <sub>1</sub>	1	3	e <sub>3</sub>
	a <sub>2</sub>	b <sub>2</sub>	2		2	e <sub>2</sub>		a <sub>2</sub>	b <sub>2</sub>	2	2	e <sub>2</sub>
	a <sub>2</sub>	b <sub>2</sub>	3		3	e <sub>3</sub>		a <sub>2</sub>	b <sub>2</sub>	2	3	e <sub>3</sub>
	a <sub>3</sub>	b <sub>3</sub>	3		3	e <sub>3</sub>		a <sub>3</sub>	b <sub>3</sub>	3	3	e <sub>3</sub>

$r \bowtie_{F'} s = k'$	(A	B	C	D	E)
	a <sub>1</sub>	b <sub>1</sub>	1	1	e <sub>1</sub>
	a <sub>2</sub>	b <sub>2</sub>	2	2	e <sub>2</sub>
	a <sub>3</sub>	b <sub>3</sub>	3	3	e <sub>3</sub>

36



## Luyện tập

### • Kết nối tự nhiên (natural join)

$$r(U) \bowtie s(V) = \{ t[U \cup V] \mid t[U] \in r \wedge t[V] \in s \}$$

p	(A	B	C)	q	(C	D)	p*q = z	(A	B	C	D)
a <sub>1</sub>	b <sub>1</sub>	1		1	d <sub>1</sub>		a <sub>1</sub>	b <sub>1</sub>	1		d <sub>1</sub>
a <sub>2</sub>	b <sub>1</sub>	1		1	d <sub>2</sub>		a <sub>1</sub>	b <sub>1</sub>	1		d <sub>2</sub>
a <sub>3</sub>	b <sub>2</sub>	2		2	d <sub>2</sub>		a <sub>2</sub>	b <sub>1</sub>	1		d <sub>1</sub>
							a <sub>2</sub>	b <sub>1</sub>	1		d <sub>2</sub>
							a <sub>3</sub>	b <sub>2</sub>	2		d <sub>2</sub>

37

## Luyện tập

### • Phép chia (Division)

$$r \div s = \{ t \mid \forall v \in s \Rightarrow (t, v) \in r \}$$

Ví dụ:

p	(A B C D E)	s	(D E)	p+s = q	(A B C)
a <sub>1</sub>	b <sub>1</sub> 1 1 e <sub>1</sub>	1 e <sub>1</sub>		a <sub>1</sub>	b <sub>1</sub> 1
a <sub>1</sub>	b <sub>1</sub> 1 2 e <sub>2</sub>	2 e <sub>2</sub>		a <sub>2</sub>	b <sub>2</sub> 2
a <sub>1</sub>	b <sub>1</sub> 1 3 e <sub>3</sub>	3 e <sub>3</sub>		a <sub>3</sub>	b <sub>3</sub> 3
a <sub>2</sub>	b <sub>2</sub> 2 1 e <sub>1</sub>				
a <sub>2</sub>	b <sub>2</sub> 2 2 e <sub>2</sub>				
a <sub>2</sub>	b <sub>2</sub> 2 3 e <sub>3</sub>				
a <sub>3</sub>	b <sub>3</sub> 3 1 e <sub>1</sub>				
a <sub>3</sub>	b <sub>3</sub> 3 2 e <sub>2</sub>				
a <sub>3</sub>	b <sub>3</sub> 3 3 e <sub>3</sub>				

38

## Bài tập

- Cho CSDL gồm 3 quan hệ sau: S(Các hãng cung ứng), P (các mặt hàng), SP(các sự cung ứng).

S(S# SNAME STATUS CITY)	SP(S# P# QTY)
S1 Smith 20 London	S1 P1 300
S2 Jones 10 Paris	S1 P2 200
S3 Black 30 Paris	S1 P3 400
	S2 P1 300
	S2 P2 400
	S3 P2 200

P(P# PNAME COLOR WEIGHT CITY)
P1 Nut red 12 London
P2 Bolt green 17 Paris
P3 Screw blue 17 Rom
P4 Screw red 14 London

39

## Yêu cầu của bài tập

- Biểu diễn các truy vấn sau bằng đại số quan hệ:
  - Đưa ra danh sách các mặt hàng màu đỏ
  - Cho biết S# của các hãng cung ứng mặt hàng 'P1' hoặc 'P2'
  - Liệt kê S# của các hãng cung ứng cả hai mặt hàng 'P1' và 'P2'
  - Đưa ra S# của các hãng cung ứng ít nhất một mặt hàng màu đỏ
  - Đưa ra S# của các hãng cung ứng tất cả các mặt hàng.

40

## Lời giải của bài tập

- Đưa ra danh sách các mặt hàng màu đỏ:
 
$$\sigma_{\text{COLOR} = \text{'red'}}(P)$$
- Cho biết S# của các hãng cung ứng mặt hàng 'P1' hoặc 'P2':
 
$$\Pi_{S\#}(\sigma_{P\# = \text{'P1'}} \vee P\# = \text{'P2'}}(SP))$$
- Liệt kê S# của các hãng cung ứng cả hai mặt hàng 'P1' và 'P2':
 
$$\Pi_{S\#}(\sigma_{P\# = \text{'P1'}}(SP)) \cap \Pi_{S\#}(\sigma_{P\# = \text{'P2'}}(SP))$$
- Đưa ra S# của các hãng cung ứng ít nhất một mặt hàng màu đỏ:
 
$$\Pi_{S\#}(SP * \sigma_{\text{COLOR} = \text{'red'}}(P))$$
- Đưa ra S# của các hãng cung ứng tất cả các mặt hàng:
 
$$\Pi_{S\#, P\#}(SP) \div \Pi_{P\#}(P)$$

41

## Bài tập về nhà

- Cho các quan hệ sau:

Supplier				SupplyProduct		
sid	sname	size	city	sid	pid	quantity
S1	Dustin	100	London	S1	P1	500
S2	Rusty	70	Paris	S1	P2	400
S3	Lubber	120	London	S1	P3	100
				S2	P2	200
				S3	P4	100
				S2	P3	155

Product		
pid	pname	colour
P1	Screw	red
P2	Screw	green
P3	Nut	red
P4	Bolt	blue

42

## Bài tập về nhà

- Biểu diễn các truy vấn sau bằng biểu thức đại số quan hệ:
  - Đưa ra {sid,sname,size,city} của các Supplier có trụ sở tại London
  - Đưa ra {pname} của tất cả các mặt hàng
  - Đưa ra {sid} của các Supplier cung cấp mặt hàng P1 hoặc P2
  - Đưa ra {sname} của các Supplier cung cấp mặt hàng P3
  - Đưa ra {sname} của các hãng cung ứng ít nhất một mặt hàng màu đỏ

43

## Bài tập về nhà

- Đưa ra {sid} của các hãng cung ứng tất cả các mặt hàng màu đỏ
- Đưa ra {sname} của các hãng cung ứng ít nhất một mặt hàng màu đỏ hoặc màu xanh
- Đưa ra {sname} của các hãng cung ứng ít nhất 1 mặt hàng màu đỏ và một mặt hàng màu xanh
- Đưa ra {sid} của các hãng không cung ứng mặt hàng nào

44



45

## Ngôn ngữ QBE

46

## QBE (Query-By-Example)

- Là một ngôn ngữ truy vấn dữ liệu
- Các câu truy vấn được thiết lập bởi một giao diện đồ họa
- Phù hợp với các câu truy vấn đơn giản, tham chiếu đến ít bảng
- Một số sản phẩm: IBM (IBM Query Management Facility), Paradox, MS. Access, ...

47

## Truy vấn trên một quan hệ

- P.~ Print

Student	ID	Name	Suburb
		P_x	Bundoora

- Biểu thức đại số quan hệ tương đương

$\sigma_{suburb="Bundoora"}(Student)$

48

## Truy vấn trên một quan hệ (tiếp)

- Lựa chọn tất cả các cột

Student	ID	Name	Suburb
P.			Bundoora

- Sắp xếp

Student	ID	Name	Suburb
		P.AO(1)	P.AO(2)

- AO: sắp xếp tăng dần
- DO: sắp xếp giảm dần

49

## Truy vấn trên nhiều quan hệ

- Đưa ra tên của các sinh viên có đăng ký ít nhất một khoá học

Student	ID	Name	Suburb
	_id	P._name	

Enrol	SID	Course
	_id	

- Đưa ra tên các sinh viên không đăng ký một khoá học nào

Student	ID	Name	Suburb
	_id	P._name	

Enrol	SID	Course
	_id	

50

## Các tính toán tập hợp

- Các phép toán: AVG, COUNT, MAX, MIN, SUM

- Ví dụ: đưa ra tên các thành phố và số lượng sinh viên đến từ thành phố đó

Student	ID	Name	Suburb
	_id		G.P.

- G. - Grouping

51

## Hộp điều kiện

- Được sử dụng để biểu diễn
  - Điều kiện trên nhiều hơn 1 thuộc tính
  - Điều kiện trên các trường tính toán tập hợp
- Ví dụ: đưa ra danh sách các thành phố có nhiều hơn 5 sinh viên

Student	ID	Name	Suburb
	_id		G.P.

Condition
COUNT._id > 5

52

## Các thao tác thay đổi dữ liệu

- Xóa

Student	ID	Name	Suburb
D.	1108		

- Thêm

Student	ID	Name	Suburb
I.	1179	David	Evry

- Sửa

Student	ID	Name	Suburb
	1179		U.Paris

53

## Tính đầy đủ của QBE

- Có thể biểu diễn cả 5 phép toán đại số cơ sở ( , , , \, x)

54

## Định nghĩa dữ liệu trong QBE

- sử dụng cùng qui cách và giao diện đồ họa như đối với truy vấn.

I.Student	I.	ID	Name	Suburb
KEY	I.	Y	N	N
TYPE	I.	CHAR(5)	CHAR(30)	CHAR(30)
DOMAIN	I.	Sid	SName	Surb
INVERSION	I.	Y	N	N

55

## Định nghĩa dữ liệu trong QBE (tiếp)

- Các khung nhìn

I.View V	I.	ID	Name	Course
	I.	_id	_name	_course

Student	ID	Name	Suburb	Enrol	SID	Course
	_id	_name			_id	_course

56

## Ngôn ngữ SQL

57

## SQL (Structured Query Language)

- 1975: SEQUEL
  - System-R
- 1976: SEQUEL2
- 1978/79: SQL
  - System-R
- 1986: chuẩn SQL-86
- 1989: chuẩn SQL-89
- 1992: chuẩn SQL-92
- 1996: chuẩn SQL-96



ORACLE

58

## Các thành phần của SQL

- Ngôn ngữ định nghĩa dữ liệu (*Data Definition Language*)
  - Cấu trúc các bảng CSDL
  - Các mối liên hệ của dữ liệu
  - Quy tắc, ràng buộc áp đặt lên dữ liệu
- Ngôn ngữ thao tác dữ liệu (*Data Manipulation Language*)
  - Thêm, xoá, sửa dữ liệu trong CSDL
  - Truy vấn dữ liệu
- Ngôn ngữ điều khiển dữ liệu (*Data Control Language*)
  - Khai báo bảo mật thông tin
  - Quyền hạn của người dùng trong khai thác CSDL

59

## Ngôn ngữ định nghĩa dữ liệu

- Các thông tin được định nghĩa bao gồm
  - Sơ đồ quan hệ
  - Kiểu dữ liệu hay miền giá trị của mỗi thuộc tính
  - Các ràng buộc toàn vẹn
  - Các chỉ số đối với mỗi bảng
  - Thông tin an toàn và ủy quyền đối với mỗi bảng
  - Cấu trúc lưu trữ vật lý của mỗi bảng trên đĩa
- Được biểu diễn bởi các lệnh định nghĩa dữ liệu

60

## Quy ước đặt tên và kiểu dữ liệu

- Quy ước đặt tên
  - 32 ký tự: chữ cái, số, dấu \_
- Kiểu dữ liệu (SQL-92)
  - CHAR(n)
  - VARCHAR(n)
  - Int
  - Smallint
  - Numeric(p,d)
  - Real, double
  - float(n)
  - Date
  - time

61

## Cú pháp

### • Tạo bảng

```
CREATE TABLE tab(  
    col1 type1(size1)[NOT NULL], ...,  
    col2 type2(size2)[NOT NULL], ...,  
    .....  
    [CONSTRAINT <constraint name> <constraint  
    type> clause]  
    ...  
);
```

### • Xoá bảng

```
DROP TABLE tab
```

62

## Tạo bảng - Ví dụ:

```
CREATE TABLE Supplier(  
    sid varchar(4) NOT NULL,  
    sname varchar(30) NOT NULL,  
    size smallint,  
    city varchar(20),  
    CONSTRAINT KhoachinhS primary key(sid)  
);
```

63

## Tạo bảng - Ví dụ (tiếp)

```
CREATE TABLE SupplyProduct(  
    sid varchar(4) NOT NULL,  
    pid varchar(4) NOT NULL,  
    quantity smallint,  
    primary key(sid,pid),  
    foreign key(sid) references Supplier(sid),  
    foreign key(pid) references Product(pid),  
    check(quantity > 0)  
);
```

64

## Kiểu ràng buộc

- Ràng buộc toàn vẹn (RBTV) về giá trị miền

```
CONSTRAINT <name>
```

```
CHECK <condition>
```

- RBTV về khoá ngoại hay phụ thuộc tồn tại

```
CONSTRAINT <name> FOREIGN KEY (fk1,fk2,...)  
REFERENCES tab(k1,k2);
```

65

## Thêm/xoá/sửa cột của các bảng

### • Thêm

```
ALTER TABLE <tên bảng>  
ADD COLUMN <tên cột> <kiểu dữ liệu> [NOT NULL];
```

### • Xoá

```
ALTER TABLE <tên bảng>  
DROP COLUMN <tên cột>;
```

### • Sửa

```
ALTER TABLE <tên bảng>  
CHANGE COLUMN <tên cột> <kiểu dữ liệu mới>;
```

66

## Ví dụ:

- ALTER TABLE SupplyProduct ADD COLUMN price real NOT NULL;
- ALTER TABLE SupplyProduct DROP COLUMN price;
- ALTER TABLE Supplier CHANGE COLUMN sname TO varchar(20);

67

## Thêm/xóa các ràng buộc

### • Thêm

ALTER TABLE <tên bảng>  
ADD CONSTRAINT <tên ràng buộc>  
<kiểu ràng buộc>

### • Xóa

ALTER TABLE <tên bảng>  
DROP CONSTRAINT <tên ràng buộc>

68

## Ngôn ngữ truy vấn dữ liệu

### • Cú pháp câu lệnh SQL:

SELECT [DISTINCT] <DS> | \* | <Biểu thức> | <Hàm TV>  
FROM <DS bảng>  
[WHERE <Điều kiện tìm kiếm>]  
[GROUP BY <DS cột> [HAVING <Điều kiện>]]  
[ORDER BY <Danh sách cột> [ASC | DESC]]  
[UNION | INTERSECT | MINUS <Câu truy vấn khác>]

69

## Truy vấn không điều kiện trên một bảng

### • Tìm thông tin từ các cột của bảng

> SELECT ColumnName, ColumnName, ...  
FROM TableName;  
> SELECT \*  
FROM TableName;

### • Ví dụ

SELECT Name  
FROM Student;

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

→

Name
Robert
Glen
Norman
Mary

70

## Truy vấn không điều kiện trên một bảng Một số ví dụ khác:

- Đưa ra tên của các mặt hàng  
SELECT pname FROM Product;
- Đưa ra tên khác nhau của các mặt hàng  
SELECT DISTINCT pname  
FROM Product;
- Đưa ra toàn bộ thông tin về các hãng cung ứng  
SELECT \* FROM Supplier;
- Đưa ra mã số hãng cung ứng, mã mặt hàng được cung ứng và 10 lần số lượng mặt hàng đã được cung ứng  
SELECT sid, pid, quantity\*10  
FROM SupplyProduct;

71

## Truy vấn có điều kiện trên 1 bảng

### • Chọn các bản ghi (dòng)

SELECT ColumnName, ColumnName, ...  
FROM TableName  
WHERE condition\_expression;

### • Ví dụ

SELECT \*  
FROM Student  
WHERE suburb="Bundoora";

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

→

Id	Name	Suburb
3936	Glen	Bundoora
8507	Norman	Bundoora

72

### Truy vấn có điều kiện trên 1 bảng Một số ví dụ khác:

- Đưa ra tên của các hãng cung ứng có trụ sở tại London  

```
SELECT sname FROM Supplier  
WHERE city = 'London';
```
- Đưa ra mã số và tên của các hãng cung ứng nằm ở London và có số nhân viên lớn hơn 75  

```
SELECT sid, sname FROM Supplier  
WHERE city = 'London' AND size > 75;
```

73

### Biểu diễn điều kiện lựa chọn

- Các phép toán quan hệ: =, !=, <, >, <=, >=
- Các phép toán logic: NOT, AND, OR
- Phép toán phạm vi: BETWEEN, IN, LIKE
  - Kiểu dữ liệu số
    - attr BETWEEN val1 AND val2 ( (attr>=val1) and (attr<=val2) )
    - attr IN (val1, val2, ...) ( (attr=val1) or (attr=val2) or ... )
  - Kiểu dữ liệu chuỗi
    - LIKE: sử dụng đối sánh mẫu chuỗi với các ký tự % hoặc \_ (thay thế cho 1 ký tự bất kỳ), \* hay % (thay thế cho 1 chuỗi ký tự bất kỳ)

74

### Biểu diễn điều kiện lựa chọn - Ví dụ:

- Đưa ra thông tin của các hãng cung ứng có số nhân viên trong khoảng từ 100 đến 150  

```
SELECT * FROM Supplier  
WHERE size BETWEEN 100 AND 150;
```
- Đưa ra mã số của hãng cung ứng mặt hàng P1 hoặc P2
  - Cách 1:  

```
SELECT sid FROM SupplierProduct  
WHERE pid = 'P1' OR pid = 'P2';
```
  - Cách 2:  

```
SELECT sid FROM SupplierProduct  
WHERE pid IN ('P1', 'P2');
```

75

### Biểu diễn điều kiện lựa chọn - Ví dụ (tiếp)

- Đưa ra thông tin của hãng sản xuất có trụ sở đặt tại thành phố bắt đầu bằng chữ New  

```
SELECT * FROM SUPPLIER  
WHERE city LIKE 'New%';
```

76

### Truy vấn có sử dụng phép toán đổi tên

- SQL cho phép đổi tên các bảng và các cột trong một câu truy vấn (sau mệnh đề SELECT và FROM) sử dụng cấu trúc:
- <tên cũ> AS <tên mới>
  - Đưa ra tên và số nhân viên của các hãng cung ứng ở Paris  

```
SELECT sname AS HangOParis, size AS SoNhanVien  
FROM Supplier  
WHERE city = 'Paris';
```
  - Đưa ra tên và số nhân viên của các hãng cung ứng ở Paris  

```
SELECT SID, Stud.Name as SName,  
Sub.Name as Subject  
FROM Student as Stud,Takes,  
Subject as Sub  
WHERE (Id=SID) and (SNO = No)
```

77

### Truy vấn phức tạp trên nhiều bảng

- Điều kiện kết nối  

```
SELECT T1.C1,T1.C2,T2.C1,T2.C4, ...  
FROM T1, T2  
WHERE condition_expression
```
- Ví dụ: đưa ra danh sách mã sinh viên (Id), tên sinh viên (Name), thành phố (Suburb), mã khóa học (Course) mà các sinh viên đã đăng ký  

```
SELECT Id, Name, Suburb,Course  
FROM Student,Enrol  
WHERE Id=SID
```

78

## Truy vấn phức tạp trên nhiều bảng Một số ví dụ khác:

- Đưa ra tên của hãng có cung ứng mặt hàng P1  

```
SELECT sname
FROM Supplier S, SupplyProduct SP
WHERE S.sid = SP.sid AND SP.pid = 'P1';
```
- Đưa ra tên và mã số của hãng cung ứng ít nhất một mặt hàng màu đỏ  

```
SELECT sname, sid
FROM Supplier S, SupplyProduct SP, Product P
WHERE S.sid = SP.sid AND P.pid = SP.pid AND
P.colour = 'red';
```

79

## Loại trừ các bản ghi trùng nhau

- Từ khoá **DISTINCT**

```
SELECT DISTINCT <bt1>, <bt2>, ...
FROM <bang1>, <bang2>, ...
```

- Ví dụ: đưa ra danh sách tên các khoa (dept) tương ứng với các khoá học (Course). Mỗi giá trị chỉ hiện thị một lần

```
SELECT DISTINCT Dept
FROM Course
```

80

## Tìm kiếm có sắp xếp

- Sắp xếp các bản ghi kết quả theo một thứ tự cho trước

```
SELECT <bt1>, <bt2>, ...
FROM <bang1>, <bang2>, ...
[WHERE <điều kiện chọn>]
ORDER BY <tt1> | <biểu thức số 1> [ASC | DESC]
```

- Ví dụ: đưa ra danh sách tên các sinh viên theo thứ tự tăng dần

```
SELECT Name
FROM Student
ORDER BY Name ASC
```

81

## Phân nhóm các bản ghi kết quả

- Phân nhóm các bản ghi kết quả theo giá trị của 1 hoặc nhiều thuộc tính

```
SELECT <bt1>, <bt2>, ...
FROM <bang1>, <bang2>, ...
[WHERE <điều kiện chọn>]
[GROUP BY <tt1>, <tt2>, ...]
```

- Cột được chỉ ra trong mệnh đề GroupBy được sử dụng làm cơ sở để chia nhóm. Cột này cũng bắt buộc phải được chỉ ra trong mệnh đề Select
- Ví dụ đưa ra tên các sinh viên nhóm theo thành phố của sinh viên đó

```
SELECT Suburb, Name
FROM Student
GROUP BY Suburb
```

```
SELECT Suburb,
Count(Id)
FROM Student
GROUP BY Suburb
```

82

## Điều kiện hiển thị các bản ghi kết quả

- Lựa chọn các bản ghi kết quả để hiển thị

```
SELECT <bt1>, <bt2>, ...
FROM <bang1>, <bang2>, ...
[WHERE <điều kiện chọn>]
HAVING <điều kiện in kết quả>
```

- Ví dụ: đưa ra tên các thành phố có nhiều hơn 3 sinh viên

```
SELECT Suburb, COUNT(ID)
FROM Student
GROUP BY Suburb
HAVING COUNT(ID) > 3
```

83

## Các phép toán tập hợp: UNION, MINUS, INTERSECT

- Ví dụ: đưa ra danh sách tên các môn học không có sinh viên nào tham dự

```
SELECT DISTINCT Subject.Name
FROM Subject
```

MINUS

```
SELECT DISTINCT Subject.Name
FROM Student, Takes, Subject
WHERE Student.Id = Takes.SID and Takes.SNO = Subject.No
```

- Tìm sid của hãng cung ứng đồng thời 2 mặt hàng P1 và P2

```
SELECT sid FROM SupplyProduct WHERE pid = 'P1'
```

INTERSECT

```
SELECT sid FROM SupplyProduct WHERE pid = 'P2'
```

- Tìm mã số của hãng không cung ứng mặt hàng nào

```
SELECT sid FROM Supplier
MINUS
SELECT sid FROM SupplyProduct
```

84



## Các câu truy vấn lồng nhau

- Là trường hợp các câu truy vấn (con) được viết lồng nhau
- Thường được sử dụng để
  - Kiểm tra thành viên tập hợp (**IN**, **NOT IN**)
  - So sánh tập hợp (**>ALL**, **>=ALL**, **<ALL**, **<=ALL**, **=ALL**, **NOT IN**, **SOME**, )
    - vd: **SELECT \* FROM Supplier WHERE SIZE >= ALL(SELECT SIZE FROM Supplier);**
  - Kiểm tra các bảng rỗng (**EXISTS** hoặc **NOT EXISTS**)
- Các truy vấn con lồng nhau thông qua mệnh đề **WHERE**

85

## Các câu truy vấn lồng nhau (tiếp)

- Kiểm tra thành viên tập hợp với **IN** và **NOT IN**:
  - Đưa ra mã số của các hãng cung ứng đồng thời 2 mặt hàng P1 và P2:  
**SELECT DISTINCT sid FROM SupplyProduct WHERE pid = 'P1' AND sid IN (SELECT sid FROM SupplyProduct SP2 WHERE SP2.pid = 'P2');**
  - Đưa ra sid của các hãng không cung ứng mặt hàng P3:  
**SELECT sid FROM SupplyProduct WHERE sid NOT IN (SELECT sid From SupplyProduct SP2 WHERE SP2.pid = 'P3');**

86

## Các câu truy vấn lồng nhau (tiếp)

- So sánh tập hợp: Sử dụng các phép toán **<**, **>**, **≥**, **≤**, **=**, **≠** kèm với các mệnh đề **ANY** và **ALL**
  - Đưa ra tên của các hãng có số nhân viên đồng nhất:  
**SELECT sname FROM Supplier WHERE size ≥ ALL(SELECT size FROM Supplier)**
  - Đưa ra sid của hãng cung ứng một mặt hàng với số lượng bằng ít nhất 1 trong số lượng các mặt hàng được cung ứng bởi S2  
**SELECT sid FROM SupplyProduct WHERE sid ≠ 'S2' AND quantity = ANY(SELECT quantity FROM SupplyProduct SP2 WHERE SP2.sid = 'S2');**

87

## Các câu truy vấn lồng nhau (tiếp)

- Kiểm tra tập hợp rỗng với **EXISTS** và **NOT EXISTS**
  - **EXISTS**(câu truy vấn con): nhận giá trị đúng khi câu truy vấn con cho ra kết quả là một quan hệ khác rỗng
  - **NOT EXISTS**(câu truy vấn con): nhận giá trị đúng khi câu truy vấn con cho ra kết quả là một quan hệ rỗng

88

## Các câu truy vấn lồng nhau (tiếp)

- Đưa ra thông tin của các nhà cung cấp đã cung ứng ít nhất một mặt hàng  
**SELECT \* FROM Supplier S WHERE EXISTS (SELECT sid FROM SupplyProduct SP WHERE S.sid = SP.sid);**
- Đưa ra thông tin của các nhà cung cấp không cung ứng mặt hàng nào  
**SELECT \* FROM Supplier S WHERE NOT EXISTS (SELECT \* FROM SupplyProduct SP WHERE S.sid = SP.sid);**

89

## Các hàm thư viện

- Hàm tính toán trên nhóm các bản ghi
  - **MAX/MIN**
  - **SUM**
  - **AVG**
  - **COUNT**
- Hàm tính toán trên bản ghi
  - Hàm toán học: **ABS**, **SQRT**, **LOG**, **EXP**, **SIGN**, **ROUND**
  - Hàm xử lý chuỗi ký tự: **LEN**, **LEFT**, **RIGHT**, **MID**
  - Hàm xử lý thời gian: **DATE**, **DAY**, **MONTH**, **YEAR**, **HOURL**, **MINUTE**, **SECOND**
  - Hàm chuyển đổi kiểu giá trị: **FORMAT**

90

## Một số ví dụ với các hàm thư viện

- Có bao nhiêu mặt hàng khác nhau được cung ứng  

```
SELECT COUNT(DISTINCT pid)
FROM SupplyProduct;
```
- Có tổng cộng bao nhiêu nhân viên làm cho các hãng ở Paris  

```
SELECT SUM(size) FROM Supplier
WHERE city = 'Paris';
```
- Đưa ra số lượng mặt hàng trung bình mà hãng S1 cung ứng  

```
SELECT AVG(quantity)
FROM SupplyProduct
WHERE sid = 'S1';
```

91

## Một số truy vấn phức tạp

- Đưa ra tên của hãng S1 và tổng số mặt hàng mà hãng đó cung ứng  

```
SELECT sname, SUM(quantity)
FROM Supplier S, SupplyProduct SP
WHERE S.sid = SP.sid AND S.sid = 'S1'
GROUP BY sname;
```
- Đưa ra mã số các hãng cung ứng và số lượng trung bình các mặt hàng được cung ứng bởi từng hãng  

```
SELECT sid, AVG(quantity) FROM SupplyProduct
GROUP BY sid;
```
- Đưa ra mã số các hãng cung ứng mà số lượng mặt hàng trung bình được cung cấp bởi hãng đó là trong khoảng từ 75 đến 100  

```
SELECT sid, AVG(quantity) FROM SupplyProduct
GROUP BY sid HAVING AVG(quantity) BETWEEN 75 AND 100
```

92

## Các câu lệnh cập nhật dữ liệu

- Thêm  

```
>INSERT INTO table[(col1,col2,...)]
VALUES (exp1,exp2,...)
>INSERT INTO table[(col1,col2,...)]
SELECT col1,col2, ...
FROM tab1, tab2, ...
WHERE <dieu_kien>
```
- Ví dụ  

```
>INSERT INTO Student(id, Name, Suburb)
VALUES ('1179','David','Evr')
```

93

## Các câu lệnh cập nhật dữ liệu

- Xóa dữ liệu:  

```
DELETE FROM <Tên bảng>
WHERE <Điều kiện xóa>;
```
- Ví dụ:  

```
DELETE FROM SupplyProduct
WHERE sid = 'S4';
DELETE FROM Student
WHERE Suburb = "Bundoora";
```

94

## Các câu lệnh cập nhật dữ liệu

- Sửa đổi dữ liệu:  

```
UPDATE <tên bảng> SET (<Tên cột> = Giá trị mới , ...)
[WHERE <Điều kiện sửa đổi>;]
```
- Ví dụ:  

```
UPDATE Supplier SET city = 'Milan'
WHERE sid = 'S1';
```

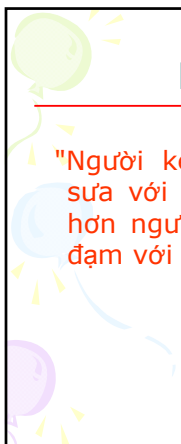
  - Tất cả các mặt hàng được cung cấp với số lượng nhỏ hơn 100 đều tăng số lượng lên 1.5 lần  

```
UPDATE SupplyProduct SET quantity = quantity * 1.5
WHERE quantity < 100;
```

95



96



## Lời hay ý đẹp

"Người kém thông minh nhưng say sưa với công việc, tiến mạnh và xa hơn người cực thông minh mà lãnh đạm với công việc".

J. Deval

97

# Lý thuyết thiết kế cơ sở dữ liệu quan hệ

Nguyễn Hồng Phương

[phuongnh-fit@mail.hut.edu.vn](mailto:phuongnh-fit@mail.hut.edu.vn)

Bộ môn Hệ thống thông tin  
Viện Công nghệ thông tin và Truyền thông  
Đại học Bách Khoa Hà Nội

1

## Nội dung

- Tổng quan về thiết kế CSDLQH
- Phụ thuộc hàm
- Phép tách các sơ đồ quan hệ (SDQH)
- Các dạng chuẩn đối với các SDQH

2

## Tổng quan về thiết kế CSDLQH

- Vấn đề của một sơ đồ quan hệ được thiết kế chưa tốt:

Giả sử ta cần một cơ sở dữ liệu lưu trữ thông tin về các hãng cung ứng. Sơ đồ quan hệ được thiết kế trong đó tất cả các thuộc tính cần thiết được lưu trong đúng 1 quan hệ:

Suppliers(sid, sname, city, numofemps, product, quantity)

sid	sname	city	NOE	product	quantity
S1	Smith	London	100	Screw	50
S1	Smith	London	100	Nut	100
S2	J&J	Paris	124	Screw	78
S3	Blake	Tokyo	75	Bolt	100

## Các vấn đề đối với CSDL VD

- **Dư thừa dữ liệu:** Hãng nào cung ứng nhiều hơn 1 mặt hàng thì thông tin của hãng đó sẽ bị lặp lại trong bảng (VD S1), mặt hàng được cung ứng bởi nhiều hãng cũng bị lặp lại (VD Screw)
- **Di thường dữ liệu khi thêm:** Nếu có một hãng chứa cung cấp mặt hàng nào, vậy giá trị cho thuộc tính product và quantity trong bộ dữ liệu mới được thêm vào sẽ không được xác định
- **Di thường dữ liệu khi xóa:** Nếu một hãng chỉ cung cấp 1 mặt hàng, nếu ta muốn xóa thông tin về sự cung cấp này thì ta sẽ mất thông tin về hãng cung cấp
- **Di thường dữ liệu khi sửa đổi:** Do thông tin bị lặp lại nên việc sửa đổi 1 bộ dữ liệu có thể dẫn đến việc không nhất quán trong dữ liệu về một hãng nếu sơ sót không sửa đổi trên toàn bộ các bộ giá trị liên quan đến hãng đó

4

## Đề xuất giải pháp

- Nếu sơ đồ trên được thay thế bằng 2 sơ đồ quan hệ
  - *Supp(sid, sname, city, numofemps)*
  - *Supply(sid, product, quantity)*

Thì tất cả các vấn đề nêu ở trên sẽ được loại bỏ. Tuy nhiên khi tìm kiếm kết nối 2 bảng chứ không chỉ là chọn và chiếu trên 1 bảng như ở cách thiết kế trước

5

## Mục đích của chuẩn hoá

- Xác định được 1 tập các lược đồ quan hệ cho phép tìm kiếm thông tin một cách dễ dàng, **đồng thời** tránh được dư thừa dữ liệu
- Hướng tiếp cận: Một trong những kỹ thuật được sử dụng là Tách các lược đồ quan hệ có vấn đề thành những lược đồ quan hệ chuẩn hơn. Phụ thuộc hàm có thể được sử dụng để nhận biết các lược đồ chưa chuẩn và đề xuất hướng cải tiến

6

## Phụ thuộc hàm

- Định nghĩa: Cho  $R(U)$  là một sơ đồ quan hệ với  $U$  là tập thuộc tính  $\{A_1, A_2, \dots, A_n\}$ .  $X, Y$  là tập con của  $U$ . Nói rằng  $X$  xác định  $Y$  hay  $Y$  là phụ thuộc hàm vào  $X$  ( $X \rightarrow Y$ ) nếu với 1 quan hệ  $r$  xác định trên  $R(U)$  và 2 bộ bất kỳ  $t_1, t_2$  thuộc  $r$  mà  $t_1[X] = t_2[X]$  thì ta có  $t_1[Y] = t_2[Y]$

7

## Ví dụ

- Ví dụ 1:

A	B	C
a1	b1	c1
a2	b2	c2
a3	b1	c1
a4	b3	c2

- $A \rightarrow B, A \rightarrow C, B \rightarrow C$
- Ví dụ 2: trong cơ sở dữ liệu mẫu dùng trong chương 3, ta có bảng  $S$ , với mỗi giá trị của  $sid$  đều tồn tại một giá trị tương ứng cho  $sname$ ,  $city$  và  $status$ . Do đó ta có  $sid \rightarrow sname$ ,  $sid \rightarrow city$ ,  $sid \rightarrow status$

8

## Hệ tiên đề Amstrong đối với phụ thuộc hàm

Cho

- $R(U)$  là 1 sơ đồ quan hệ,  $U$  là tập các thuộc tính.
- $X, Y, Z, W \subseteq U$   
(Ký hiệu:  $XY = X \cup Y$ )

- Phản xạ (reflexivity)**  
Nếu  $Y \subseteq X$  thì  $X \rightarrow Y$
- Tăng trưởng (augmentation)**  
Nếu  $X \rightarrow Y$  thì  $XZ \rightarrow YZ$
- Bắc cầu (transitivity)**  
Nếu  $X \rightarrow Y, Y \rightarrow Z$  thì  $X \rightarrow Z$

9

## Hệ quả của hệ tiên đề Amstrong

- Luật hợp (union)**  
Nếu  $X \rightarrow Y, X \rightarrow Z$  thì  $X \rightarrow YZ$ .
- Luật tựa bắc cầu (pseudotransitivity)**  
Nếu  $X \rightarrow Y, WY \rightarrow Z$  thì  $XW \rightarrow Z$ .
- Luật tách (decomposition)**  
Nếu  $X \rightarrow Y, Z \subseteq Y$  thì  $X \rightarrow Z$

10

## Ví dụ

- Ví dụ 1:  
Cho tập phụ thuộc hàm  $\{AB \rightarrow C, C \rightarrow A\}$   
Chứng minh:  $BC \rightarrow ABC$

$C \rightarrow A$	$BC \rightarrow AB$
$AB \rightarrow C$	$AB \rightarrow ABC$
$BC \rightarrow AB, AB \rightarrow ABC$	$BC \rightarrow ABC$

- Ví dụ 2:  
Cho lược đồ quan hệ  $R(ABEIJGH)$  và tập phụ thuộc hàm  $F = \{AB \rightarrow E, AG \rightarrow J, BE \rightarrow I, E \rightarrow G, GI \rightarrow H\}$   
Chứng minh:  $AB \rightarrow GH$

11

## Bao đóng của một tập phụ thuộc hàm

- Định nghĩa:** Cho  $F$  là một tập phụ thuộc hàm. Bao đóng của  $F$  ký hiệu là  $F^+$  là tập lớn nhất chứa các phụ thuộc hàm có thể được suy ra từ các phụ thuộc hàm trong  $F$
- Bao đóng của một tập phụ thuộc hàm có thể rất lớn, sẽ chi phí rất tốn kém cho việc tìm kiếm bao đóng của 1 tập phụ thuộc hàm. Do đó để thuận tiện cho việc kiểm tra xem một phụ thuộc hàm có được suy diễn từ một tập phụ thuộc hàm có sẵn không, người ta có thể sử dụng Bao đóng của 1 tập thuộc tính

12

### Bao đóng của một tập các thuộc tính đối với tập các phụ thuộc hàm

- Định nghĩa: Cho một lược đồ quan hệ  $R(U)$ ,  $F$  là một tập phụ thuộc hàm trên  $U$ .  $X$  là tập con của  $U$ . Bao đóng của tập thuộc tính  $X$  ký hiệu là  $X^+$  là tập tất cả các thuộc tính được xác định hàm bởi  $X$  thông qua tập  $F$   
 $X^+ = \{A \in U \mid X \twoheadrightarrow A \text{ theo } F\}$
- Ta có thể thấy là định nghĩa về bao đóng của một tập thuộc tính dựa trên bao đóng của tập phụ thuộc hàm. Trên thực tế, người ta đưa ra một thuật toán để giúp xác định bao đóng của một tập thuộc tính dễ dàng hơn

13

### Thuật toán 1: Tìm bao đóng của một tập thuộc tính đối với tập phụ thuộc hàm

- Vào:** Tập hữu hạn các thuộc tính  $U$ , tập các phụ thuộc hàm  $F$  trên  $U$
- Ra:**  $X^+$
- Thuật toán**
  - $B^0$   $X^0 = X$
  - $B^i$  Tính  $X^i$  từ  $X^{i-1}$   
 Nếu  $Y \twoheadrightarrow Z$  theo  $F$  và  $Y \subseteq X^{i-1}$  và  $A \in Z$  và  $A \notin X^{i-1}$   
 thì  $X^i = X^{i-1} \cup A$   
 ngược lại,  $X^i = X^{i-1}$   
 Nếu  $X^i = X^{i-1}$   
 thì lặp lại  $B^i$   
 ngược lại, chuyển  $B^n$
  - $B^n$   $X^+ = X^n$

14

### Ví dụ

- Cho  $R(U)$ ,  $U = \{A, B, C, D, E, F\}$   
 $F = \{AB \rightarrow C, BC \rightarrow AD, D \rightarrow E, CF \rightarrow B\}$   
 Tính  $(AB)^+$
- Thực hiện:
  - Bước 0:  $X^0 = AB$
  - Bước 1:  $X^1 = ABC$  (do  $AB \rightarrow C$ )
  - Bước 2:  $X^2 = ABCD$  (do  $BC \rightarrow AD$ )
  - Bước 3:  $X^3 = ABCDE$  (do  $D \rightarrow E$ )
  - Bước 4:  $X^4 = ABCDE$

15

### Bổ đề

- $X \twoheadrightarrow Y$  được suy diễn từ hệ tiên đề Armstrong khi và chỉ khi  $Y \subseteq X^+$
- Chứng minh:
  - Giả sử  $Y = A_1 \dots A_n$ , với  $A_1, \dots, A_n$  là các thuộc tính và  $Y \subseteq X^+$
  - Từ định nghĩa  $X^+$  ta có  $X \twoheadrightarrow A_i$ . Áp dụng tiên đề Armstrong cho mọi  $i$ , suy ra  $X \twoheadrightarrow Y$  nhờ luật hợp.
  - Ngược lại, giả sử có  $X \twoheadrightarrow Y$ , áp dụng hệ tiên đề Armstrong cho mỗi  $i$ , ta có  $X \twoheadrightarrow A_i$ ,  $A_i \in Y$  nhờ luật tách. Từ đó suy ra  $Y \subseteq X^+$

16

### Khoá

- Định nghĩa: Cho lược đồ quan hệ  $R(U)$ ,  $F$  là một tập các phụ thuộc hàm xác định trên  $U$ .  $K$  là một tập con của  $U$ ,  $K$  được gọi là khoá tối thiểu của  $R$  nếu như
  - $K \rightarrow U$  là một phụ thuộc hàm trong  $F^+$
  - Với mọi tập con thực sự  $K'$  của  $K$  thì  $K' \not\rightarrow U$  không thuộc  $F^+$
- Với những gì ta đã đề cập trong phần bao đóng ở trên, ta có thể nói, để thỏa mãn là một khoá tối thiểu thì  $K^+ = U$  và  $K$  là tập thuộc tính nhỏ nhất có tính chất như vậy

17

### Thuật toán 2: Tìm khoá tối thiểu

- Vào:**  $U = \{A_1, A_2, \dots, A_n\}$ ,  $F$
- Ra:** khoá tối thiểu  $K$  xác định được trên  $U$  và  $F$
- Thuật toán**
  - $B^0$   $K^0 = U$
  - $B^i$  Nếu  $(K^{i-1} \setminus \{A_i\}) \rightarrow U$   
 thì  $K^i = K^{i-1} \setminus \{A_i\}$   
 ngược lại,  $K^i = K^{i-1}$   
 Nếu  $K^i = K^{i-1}$   
 thì  $K = K^i$
  - $B^{n+1}$   $K = K^n$

18

## Ví dụ

- Cho  $U = \{A, B, C, D, E\}$
- $F = \{AB \rightarrow C, AC \rightarrow B, BC \rightarrow DE\}$ . Tìm một khoá tối thiểu của một quan hệ  $r$  xác định trên  $U$  và  $F$
- Thực hiện
- $B0: K^0 = U = ABCDE$
- $B1$ : Kiểm tra xem có tồn tại phụ thuộc hàm  $(K0 \setminus \{A\}) \rightarrow U$  ( $BCDE \rightarrow U$ ) hay không. Ta cần phải sử dụng thuật toán 1 để kiểm tra điều kiện tương đương là  $(BCDE)^+ \subseteq U$  hay không.  $(BCDE)^+ = BCDE$ , khác  $U$ . Vậy  $K^1 = K^0 = ABCDE$
- $B2$ : Tương tự, thử loại bỏ  $B$  ra khỏi  $K^1$  ta có  $(ACDE)^+ = ABCDE = U$ . Vậy  $K^2 = K^1 \setminus \{B\} = ACDE$
- $B3: K^3 = ACDE$
- $B4: K^4 = ACE$
- $B5: K^5 = AC$
- Vậy  $AC$  là một khoá tối thiểu mà ta cần tìm

19

## Nhận xét về phụ thuộc hàm

- Từ một tập các phụ thuộc hàm có thể suy diễn ra các phụ thuộc hàm khác
- Trong một tập phụ thuộc hàm cho sẵn có thể có các phụ thuộc hàm bị coi là dư thừa
- Làm thế nào để có được một tập phụ thuộc hàm tốt?

20

## Tập phụ thuộc hàm tương đương

- Định nghĩa: Tập phụ thuộc hàm  $F$  là **phủ** của tập phụ thuộc hàm  $G$  hay  $G$  là **phủ** của  $F$  hay  $F$  và  $G$  **tương đương** nếu  $F^+ = G^+$ .  
– Ký hiệu là  $F \equiv G$
- Kiểm tra tính tương đương của 2 tập phụ thuộc hàm
- $B.1$ . Với mỗi phụ thuộc hàm  $Y \rightarrow Z$  trong  $F$ ,  $Z \rightarrow Y^+$  (trên  $G$ ) thì  $Y \rightarrow Z$  thuộc  $G^+$   
Nếu với phụ thuộc hàm  $f \in F$ ,  $f \in G^+$  thì  $F \subseteq G^+$
- $B.2$ . Tương tự, nếu phụ thuộc hàm  $g \in G$ ,  $g \in F^+$  thì  $G \subseteq F^+$
- $B.3$ . Nếu  $F \subseteq G^+$  và  $G \subseteq F^+$  thì  $F \equiv G$

21

## Ví dụ

- Cho lược đồ quan hệ  $R(U)$  với  $U = \{A, B, C, D, E, F\}$   
 $F = \{AB \rightarrow C, D \rightarrow EF, C \rightarrow BD\}$   
 $G = \{AC \rightarrow B, D \rightarrow EF, B \rightarrow CD\}$   
Hỏi  $F$  và  $G$  có phải là 2 tập pth tương đương hay không?
- Thực hiện:  
Đối với các phụ thuộc hàm trong  $F$   
–  $f_1 = AB \rightarrow C$ .  $AB^+$  (đối với  $G$ ) =  $ABCDEF = U$ . Vậy  $f_1$  thuộc  $G^+$   
–  $f_2 = D \rightarrow EF$  thuộc  $G$  nên chắc chắn thuộc  $G^+$   
–  $f_3 = C \rightarrow BD$ .  $C^+$  (đối với  $G$ ) =  $C$  không chứa  $BD$ . Vậy  $f_3$  không thuộc  $G^+$   
– Kết luận  $F$  không tương đương với  $G$

22

## Tập phụ thuộc hàm không dư thừa

- Đ/N: Tập phụ thuộc hàm  $F$  là **không dư thừa** nếu không  $X \rightarrow Y \in F$  sao cho  $F \setminus \{X \rightarrow Y\} \neq F$ .
- Thuật toán 3: Tìm phủ không dư thừa của 1 tập phụ thuộc hàm  
– Vào: Tập thuộc tính  $U$ ,  $F = \{L_i \rightarrow R_i: i = 1..n\}$   
– Ra: Phủ không dư thừa  $F'$  của  $F$
- Thuật toán
- $B^0: F^0 = F$
- $B^i$ : Nếu  $F^{i-1} \setminus \{L_i \rightarrow R_i\} \neq F^{i-1}$  thì  $F^i = F^{i-1} \setminus \{L_i \rightarrow R_i\}$   
ngược lại,  $F^i = F^{i-1}$
- $B^{n+1}: F' = F^n$

23

## Phủ tối thiểu của 1 tập phụ thuộc hàm

- Đ/N:  $F_c$  được gọi là **phủ tối thiểu của 1 tập phụ thuộc hàm  $F$**  nếu thỏa mãn 3 điều kiện sau:
- Đk1**: Với  $f \in F_c$ ,  $f$  có dạng  $X \rightarrow A$ , trong đó  $A$  là 1 thuộc tính
- Đk2**: Với  $f = X \rightarrow Y \in F_c$ ,  $A \in X$  ( $A$  là 1 thuộc tính):  
 $(F_c \setminus f) \cup \{X \setminus A \rightarrow Y\} \neq F_c$
- Đk3**:  $\nexists X \rightarrow A \in F_c: F_c \setminus \{X \rightarrow A\} \neq F_c$

24

## Thuật toán 4: Tìm phủ tối thiểu của một tập phụ thuộc hàm

- Vào:** Tập thuộc tính  $U$ ,  $F = \{L_i \rightarrow R_i; i = 1..n\}$
- Ra:** phủ tối thiểu  $F_c$  của tập phụ thuộc hàm  $F$
- Thuật toán**
  - B.1. Biến đổi  $F$  về dạng  $F_1 = \{L_i \rightarrow A_j\}$**   
trong đó  $A_j$  là 1 thuộc tính bất kỳ thuộc  $U$  (thỏa mãn đk1)
  - B.2. Loại bỏ thuộc tính thừa trong vế trái của các phụ thuộc hàm**  
Lần lượt giảm ước từng thuộc tính trong vế trái của từng phụ thuộc hàm trong  $F_1$  thu được  $F_1'$ . Nếu  $F_1' \neq F_1$  thì loại bỏ thuộc tính đang xét  
Khi không có sự giảm ước nào xảy ra nữa ta thu được  $F_2$  thỏa mãn đk2
  - B.3. Loại bỏ phụ thuộc hàm dư thừa**  
Lần lượt kiểm tra từng phụ thuộc hàm  $f$ . Nếu  $F_2 \setminus f \models F_2$  thì loại bỏ  $f$   
Khi không có phụ thuộc hàm nào có thể loại bỏ thì thu được  $F_3$  thỏa mãn đk3
  - B.4.  $F_c = F_3$**

25

## Ví dụ 1

- $U = \{A, B, C\}$   
 $F = \{A \rightarrow BC, B \rightarrow C, A \rightarrow B, AB \rightarrow C\}$ . Tìm phủ tối thiểu của  $F$ ?
- $F_1 = \{A \rightarrow B, A \rightarrow C, B \rightarrow C, AB \rightarrow C\}$
- Xét các pth trong  $F_1$  mà vế trái có nhiều hơn 1 thuộc tính  $AB \rightarrow C$ . Giảm ước  $A$  thì ta còn  $B \rightarrow C$  có trong  $F_1$ , vậy  $A$  là thuộc tính thừa. Tương tự ta cũng tìm được  $B$  là thừa, vậy loại bỏ luôn  $AB \rightarrow C$  khỏi  $F_1$ .  $F_2 = \{A \rightarrow B, A \rightarrow C, B \rightarrow C\}$
- Bỏ pth thừa:  $A \rightarrow C$  là thừa. Vậy  $F_c = \{A \rightarrow B, B \rightarrow C\}$

26

## Ví dụ 2

- Tìm phủ tối thiểu của tập phụ thuộc hàm  
 $F = \{A \rightarrow B, ABCD \rightarrow E, EF \rightarrow G, ACDF \rightarrow EG\}$
- $F_1 = \{A \rightarrow B, ABCD \rightarrow E, EF \rightarrow G, ACDF \rightarrow E, ACDF \rightarrow G\}$
- Loại bỏ thuộc tính thừa** trong 3 phụ thuộc hàm  $ABCD \rightarrow E, ACDF \rightarrow E$  và  $ACDF \rightarrow G$   
Xét  $ABCD \rightarrow E$ : Giả sử giảm ước  $A$ , ta còn  $BCD \rightarrow E$ , kiểm tra  $BCD \rightarrow E$  có được suy ra từ  $F_1$  không, ta tính  $(BCD)^+$  (đối với  $F_1$ ).  $(BCD)^+ = BCD$ , không chứa  $E$ , vậy  $A$  không phải là thuộc tính thừa trong pth đang xét.  $B$  là thừa vì từ  $F_1$  ta có  $A \rightarrow B$  dẫn đến  $(ACD)^+ = ABCDE$  có chứa  $E$   
Làm tương tự ta thấy không có thuộc tính nào là thừa nữa.  $F_2 = \{A \rightarrow B, ACD \rightarrow E, EF \rightarrow G, ACDF \rightarrow E, ACDF \rightarrow G\}$

27

## Ví dụ 2 (tiếp)

- Loại bỏ pth thừa trong  $F_2$ :** Lần lượt thử loại bỏ 1 pth ra khỏi  $F_2$ , nếu tập pth thu được sau khi loại bỏ vẫn tương đương với  $F_2$  thì pth vừa loại là thừa  
 $A \rightarrow B$  không thừa vì nếu loại pth này khỏi  $F_2$  thì từ tập phụ thuộc hàm còn lại  $A$  không chứa  $B$   
Tương tự,  $ACD \rightarrow E, EF \rightarrow G$  không thừa  
 $ACDF \rightarrow E$  là phụ thuộc hàm thừa vì nếu loại bỏ pth này, trong tập pth vẫn còn lại  $ACD \rightarrow E$ , theo tiên đề tăng trưởng ta sẽ suy ra được  $ACDF \rightarrow E$   
 $ACDF \rightarrow G$  là thừa vì nếu loại bỏ pth này, trong tập pth còn lại vẫn có  $ACD \rightarrow E$  và  $EF \rightarrow G$ , do đó ta vẫn có  $(ACDF)^+ = ACDEFG$  có chứa  $G$
- Vậy  $F_c = \{A \rightarrow B, ACD \rightarrow E, EF \rightarrow G\}$

28

## Phép tách các Sơ đồ quan hệ

- Mục đích**
  - Thay thế một sơ đồ quan hệ  $R(A_1, A_2, \dots, A_n)$  bằng một tập các sơ đồ con  $\{R_1, R_2, \dots, R_k\}$  trong đó  $R_i \subseteq R$  và  $R = R_1 \cup R_2 \cup \dots \cup R_k$
- Yêu cầu của phép tách**
  - Bảo toàn thuộc tính, ràng buộc
  - Bảo toàn dữ liệu

29

## Phép tách không mất mát thông tin

- Đ/N: Cho lược đồ quan hệ  $R(U)$  phép tách  $R$  thành các sơ đồ con  $\{R_1, R_2, \dots, R_k\}$  được gọi là **phép tách không mất mát thông tin** đ/v một tập phụ thuộc hàm  $F$  nếu với mọi quan hệ  $r$  xác định trên  $R$  thỏa mãn  $F$  thì:  
$$r = R_1(r) \bowtie R_2(r) \bowtie \dots \bowtie R_k(r)$$
- Ví dụ: Phép tách mất mát thông tin**  
 $Supplier(sid, sname, city, NOE, pid, pname, colour, quantity)$   
 $\rightarrow S1(sid, sname, city, NOE)$  và  
 $SP1(pid, pname, colour, quantity)$
- Ví dụ: Phép tách không mất mát thông tin**  
 $\rightarrow S1(sid, sname, city, NOE)$  và  
 $SP2(sid, pid, pname, colour, quantity)$

30



## Định lý tách đôi

- Cho lược đồ quan hệ  $R(U)$ , tập pth  $F$ , phép tách  $R$  thành  $R_1(U_1)$ ,  $R_2(U_2)$  là một phép tách không mất mát thông tin nếu 1 trong 2 phụ thuộc hàm sau là thỏa mãn trên  $F^+$ :

$$U_1 \cap U_2 \rightarrow U_1 - U_2$$

$$U_1 \cap U_2 \rightarrow U_2 - U_1$$

- Hệ quả: Cho lược đồ quan hệ  $R(U)$  và phụ thuộc hàm  $X \rightarrow Y$  thỏa mãn trên  $R(U)$ . Phép tách  $R$  thành 2 lược đồ con  $R_1(U_1)$ ,  $R_2(U_2)$  là một phép tách không mất mát thông tin với:

$$U_1 = XY$$

$$U_2 = XZ$$

$$Z = U \setminus XY$$

31

## Thuật toán 5: Kiểm tra tính không mất mát thông tin của 1 phép tách

- Vào:**  $R(A_1, A_2, \dots, A_n)$ ,  $F$ , phép tách  $\{R_1, R_2, \dots, R_k\}$
- Ra:** phép tách là mất mát thông tin hay không

### Thuật toán

**B.1.** Thiết lập một bảng  $k$  hàng,  $n$  cột  
Nếu  $A_i$  là thuộc tính của  $R_i$  thì điền  $a_{ij}$  vào ô  $(i, j)$ .  
Nếu không thì điền  $b_{ij}$ .

**B.i.** Xét  $f = X \rightarrow Y$   $F$

Nếu 2 hàng  $t_1, t_2$  thuộc bảng :  $t_1[X] = t_2[X]$  thì đồng nhất

$t_1[Y] = t_2[Y]$ , ưu tiên về giá trị  $a$

Lập cho tới khi không thể thay đổi được giá trị nào trong bảng

**B.n.** Nếu bảng có 1 hàng gồm các kí hiệu  $a_1, a_2, \dots, a_n$  thì phép tách là không mất mát thông tin ngược lại, phép tách không bảo toàn thông tin

32

## Ví dụ

- $R = ABCD$  được tách thành  $R_1=AB$ ,  $R_2=BD$ ,  $R_3=ABC$ ,  $R_4=BCD$ .  $F = \{A \rightarrow C, B \rightarrow C, CD \rightarrow B, C \rightarrow D\}$

- B.1:** Tạo bảng gồm 4 hàng, 4 cột

	A	B	C	D
R1	a1	a2	b31	b41
R2	b12	a2	b32	a4
R3	a1	a2	a3	b43
R4	b14	a2	a3	a4

33

## Ví dụ (tiếp)

- B.2 & 3:**
- Từ  $A \rightarrow C$ , ta có

	A	B	C	D
R1	a1	a2	a3	b41
R2	b12	a2	b32	a4
R3	a1	a2	a3	b43
R4	b14	a2	a3	a4

- Từ  $B \rightarrow C$ , ta có

	A	B	C	D
R1	a1	a2	a3	b41
R2	b12	a2	a3	a4
R3	a1	a2	a3	b43
R4	b14	a2	a3	a4

34

## Ví dụ (tiếp)

- Từ  $C \rightarrow D$ , ta có

	A	B	C	D
R1	a1	a2	a3	a4
R2	b12	a2	a3	a4
R3	a1	a2	a3	a4
R4	b14	a2	a3	a4

- Vậy ta có 2 hàng có toàn các giá trị  $a$ . Chứng tỏ phép tách đã cho là không mất mát thông tin

35

## Phép tách bảo toàn tập phụ thuộc hàm

- Hình chiếu của tập phụ thuộc hàm**

Cho sơ đồ quan hệ  $R$ , tập phụ thuộc hàm  $F$ , phép tách  $\{R_1, R_2, \dots, R_k\}$  của  $R$  trên  $F$ .

Hình chiếu  $F_i$  của  $F$  trên  $R_i$  là tập tất cả  $X \rightarrow Y$   $F^+$ :

$$XY \ R_i$$

- Phép tách sơ đồ quan hệ  $R$  thành  $\{R_1, R_2, \dots, R_k\}$  là một **phép tách bảo toàn tập phụ thuộc hàm  $F$**  nếu

$$(F_1 \ F_2 \ \dots \ F_k)^+ = F^+$$

hay hợp của tất cả các phụ thuộc hàm trong các hình chiếu của  $F$  lên các sơ đồ con sẽ suy diễn ra các phụ thuộc hàm trong  $F$ .

36

## Ví dụ

- Ví dụ 1:**  $R = \{A, B, C\}$ ,  $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$  được tách thành  $R_1 = AB$ ,  $R_2 = BC$ . Phép tách này có phải là bảo toàn tập phụ thuộc hàm không?
- Ví dụ 2:**  $R = \{A, B, C\}$ ,  $F = \{AB \rightarrow C, C \rightarrow B\}$  được tách thành  $R_1 = AB$ ,  $R_2 = BC$ . Phép tách này có bảo toàn tập pth không, có mất mát thông tin không?
- Ví dụ 3:**  $R = \{A, B, C, D\}$ ,  $F = \{A \rightarrow B, C \rightarrow D\}$  được tách thành  $R_1 = AB$ ,  $R_2 = CD$ . Phép tách này có bảo toàn tập pth không, có mất mát thông tin không?
- Vậy một phép tách có bảo toàn tập phụ thuộc hàm thì không đảm bảo là nó sẽ không mất mát thông tin và ngược lại

37

## Các dạng chuẩn đối với SDQH

- Quay lại vấn đề thiết kế cơ sở dữ liệu quan hệ, câu hỏi mà chúng ta đặt ra trong quá trình này là Có cần thiết phải tinh chỉnh thiết kế nữa hay không, thực sự thiết kế mà chúng ta có được đã là tốt hay chưa. Để giúp trả lời câu hỏi này, người ta đưa ra các định nghĩa về các dạng chuẩn. Có một vài dạng chuẩn đã được xem xét, khi một quan hệ thuộc vào một dạng chuẩn nào đó thì ta có thể coi như là một số các vấn đề về dư thừa dữ liệu hay dị thường dữ liệu đã được ngăn ngừa hay tối thiểu hóa
- Các dạng chuẩn mà chúng ta quan tâm
  - Dạng chuẩn 1 (1NF)
  - Dạng chuẩn 2 (2NF)
  - Dạng chuẩn 3 (3NF)
  - Dạng chuẩn Boye-Code (BCNF)

38

## Dạng chuẩn 1 (1NF)

- Định nghĩa: Một sơ đồ quan hệ  $R$  được gọi là ở dạng chuẩn 1 nếu tất cả các miền giá trị của các thuộc tính trong  $R$  đều chỉ chứa giá trị nguyên tố
  - Giá trị nguyên tố là giá trị mà không thể chia nhỏ ra được nữa
- Một quan hệ  $r$  xác định trên sơ đồ quan hệ ở dạng chuẩn 1 thì quan hệ đấy là ở dạng chuẩn 1
- Ví dụ: Quan hệ không ở dạng chuẩn 1 và quan hệ sau khi chuẩn hóa về dạng chuẩn 1

sname	city	product	
		name	price
Blake	London	Nut	100
		Bolt	120
Smith	Paris	Screw	75

39

## Dạng chuẩn 2 (2NF)

- Định nghĩa: Một sơ đồ quan hệ  $R$  được coi là ở dạng chuẩn 2 nếu
  - Sơ đồ quan hệ này ở 1NF
  - Tất cả các thuộc tính không khoá đều phụ thuộc hàm đầy đủ vào khoá chính (Lưu ý,  $A$  là một thuộc tính khoá nếu  $A$  thuộc một khoá tối thiểu nào đó của  $R$ . Ngược lại  $A$  là thuộc tính không khoá)

40

## Phụ thuộc hàm đầy đủ

- Định nghĩa: Cho lược đồ quan hệ  $R(U)$ ,  $F$  là tập phụ thuộc hàm trên  $R$ .  $X, Y \subseteq U$ .  $Y$  được gọi là phụ thuộc đầy đủ vào  $X$  nếu:
  - $X \rightarrow Y$  thuộc  $F^+$
  - $\neg \exists X' \subseteq X: X' \rightarrow Y \in F^+$
- Các phụ thuộc hàm không đầy đủ còn gọi là phụ thuộc bộ phận

41

## Ví dụ

- $Sales(sid, sname, city, item, price)$
- $F = \{sid \rightarrow (sname, city), (sid, item) \rightarrow price\}$
- Khoá chính  $(sid, item)$ , ta có  $sname, city$  không phụ thuộc hàm đầy đủ vào khoá chính  $\Rightarrow$  Quan hệ  $Sales$  không thuộc 2NF
- $S(sid, sname, city)$  và  $Sales(sid, item, price)$  là quan hệ thuộc 2NF

42

### Dạng chuẩn 3 (tiếp)

- Định nghĩa: Một sơ đồ quan hệ R được coi là ở dạng chuẩn 3 nếu
  - Sơ đồ quan hệ này ở 2NF
  - Mọi thuộc tính không khoá đều không phụ thuộc bắc cầu vào khoá chính

43

### Phụ thuộc bắc cầu

- Định nghĩa: Cho lược đồ quan hệ R(U). F là tập phụ thuộc hàm trên R(U).  $X, Y, Z \subseteq U$ . Ta nói Z là phụ thuộc bắc cầu vào X nếu ta có  $X \rightarrow Y$ ,  $Y \rightarrow Z$  thuộc F+. Ngược lại, ta nói Z không phụ thuộc bắc cầu vào X

44

### Ví dụ

- Ví dụ 1: Trong ví dụ tách về dạng chuẩn 2 ta có: S (sid, sname, city) và Sales(sid, item, price). Xét quan hệ S, pth  $sid \rightarrow sname, city$  tồn tại trên S, sid là khoá chính, các thuộc tính không khoá sname, city đều phụ thuộc trực tiếp vào sid. S thuộc 3NF. Tương tự ta có Sales cũng thuộc 3NF
- Ví dụ 2:
  - ItemInfo(item, price, discount). F = {item  $\rightarrow$  price, price  $\rightarrow$  discount}. Khoá chính là item, thuộc tính không khoá discount phụ thuộc bắc cầu vào khoá chính item. Vậy quan hệ này không ở 3NF.
  - ItemInfo(item, price) và Discount(price, discount) thuộc 3NF.

45

### Dạng chuẩn Boye-Codd

- Định nghĩa: Một sơ đồ quan hệ R(U) với một tập phụ thuộc hàm F được gọi là ở dạng chuẩn Boye-Codd (BCNF) nếu với  $X \rightarrow A$  F+ thì
  - A là thuộc tính xuất hiện trong X *hoặc*
  - X chứa một khoá của quan hệ R.
- Ví dụ
  - R = {A, B, C}; F = {AB  $\rightarrow$  C, C  $\rightarrow$  B}.
  - R không phải ở BCNF vì C  $\rightarrow$  B, C không phải là khoá
- Chú ý:
  - Một quan hệ thuộc 3NF thì chưa chắc đã thuộc BCNF. Nhưng một quan hệ thuộc BCNF thì thuộc 3NF

46

### Tách bảo toàn tập phụ thuộc hàm về 3NF

- **Vào:** R(U), F (giả thiết F là phủ tối thiểu)
- **Ra:** Phép tách bảo toàn tập phụ thuộc hàm về 3NF
- **Thuật toán**
  - B1.** Với các  $A_i \subseteq U$ ,  $A_i \subseteq F$  thì loại  $A_i$  khỏi R và lập 1 quan hệ mới cho các  $A_i$
  - B2.** Nếu  $f \in F$ , f chứa tất cả các thuộc tính của R (đã bỏ các  $A_i$  ở bước trên) thì kết quả là R
  - B3.** Ngược lại, với mỗi  $X \rightarrow A \in F$ , xác định một quan hệ  $R_i(XA)$ .  
Nếu  $X \rightarrow A_i$ ,  $X \rightarrow A_j$  thì tạo một quan hệ chung  $R'(XA_iA_j)$

47

### Ví dụ

- Cho R = {A, B, C, D, E, F, G}
- F = {A  $\rightarrow$  B, ACD  $\rightarrow$  E, EF  $\rightarrow$  G} (đã tối thiểu)
- Xác định phép tách bảo toàn tập phụ thuộc hàm về 3NF
    - B1.** Không lập được quan hệ nào mới.
    - B2.** !  $f \in F$ : f chứa tất cả các thuộc tính của R
    - B3.** A  $\rightarrow$  B  $\Rightarrow$  R1(AB)  
ACD  $\rightarrow$  E  $\Rightarrow$  R2(ACDE)  
EF  $\rightarrow$  G  $\Rightarrow$  R3(EFG)

48

## Tách không mất mát thông tin và bảo toàn tập phụ thuộc hàm về 3NF

- Yêu cầu:
  - Bảo toàn tập phụ thuộc hàm (như thuật toán trên)
  - Đảm bảo là có một lược đồ con chứa khoá của lược đồ được tách
- Các bước tiến hành
  - Tìm một khoá tối thiểu của lược đồ quan hệ R đã cho
  - Tách lược đồ quan hệ R theo phép tách bảo toàn tập phụ thuộc hàm.
  - Nếu 1 trong các sơ đồ con có chứa khoá tối thiểu thì kết quả của B2 là kết quả cuối cùng  
Ngược lại, thêm vào kết quả đó một sơ đồ quan hệ được tạo bởi khoá tối thiểu tìm được ở 1

49

## Ví dụ

- Cho  $R(U)$  trong đó  $U = \{A, B, C, D, E, F, G\}$ .  $F = \{A \rightarrow B, ACD \rightarrow E, EF \rightarrow G\}$
- Tìm một khoá tối thiểu của R:
  - $K^0 = ABCDEFG$
  - $K^1 = K^0$  do nếu loại A thì BCDEFG  $\rightarrow U$  không thuộc  $F^+$
  - $K^2 = K^1 \setminus \{B\} = ACDEFG$  do  $ACDEFG \rightarrow U$  thuộc  $F^+$
  - $K^3 = K^2$  do nếu loại C thì ADEFG  $\rightarrow U$  không thuộc  $F^+$
  - $K^4 = K^3$  do nếu loại D thì ACEFG  $\rightarrow U$  không thuộc  $F^+$
  - $K^5 = K^4 \setminus \{E\} = ACDFG$  do  $ACDFG \rightarrow U$  thuộc  $F^+$
  - $K^6 = K^5$  do nếu loại F thì ACDG  $\rightarrow U$  không thuộc  $F^+$
  - $K^7 = K^6 \setminus \{G\} = ACDF$  do  $ACDF \rightarrow U$  thuộc  $F^+$
- Vậy khoá tối thiểu cần tìm là ACDF

50

## Ví dụ (tiếp)

- Dùng kết quả của ví dụ ở phần tách bảo toàn tập phụ thuộc hàm ta có một phép tách R thành 3 sơ đồ con  $R_1 = AB$ ,  $R_2 = ACDE$ ,  $R_3 = EFG$
- Do khoá ACDF không nằm trong bất kỳ một sơ đồ con nào trong 3 sơ đồ con trên, ta lập một sơ đồ con mới  $R_4 = ACDF$
- Kết quả cuối cùng ta có phép tách R thành 4 sơ đồ con  $\{R_1, R_2, R_3, R_4\}$  là một phép tách không mất mát thông tin và bảo toàn tập phụ thuộc hàm

51

## Tách không mất mát thông tin về BCNF

- Vào:** Sơ đồ quan hệ R, tập phụ thuộc hàm F.
- Ra:** phép tách không mất mát thông tin bao gồm một tập các sơ đồ con ở BCNF với các phụ thuộc hàm là hình chiếu của F lên sơ đồ đó.
- Cách tiến hành
  - $KQ = \{R\}$ ,
  - Với mỗi S  $KQ$ , S không ở BCNF, xét  $X \twoheadrightarrow A$  S, với điều kiện X không chứa khoá của S và A  $X$ . Thay thế S bởi  $S_1, S_2$  với  $S_1 = A \twoheadrightarrow X$ ,  $S_2 = S \setminus A$ .
  - Lặp (B2) cho đến khi S  $KQ$  đều ở BCNF  
KQ gồm các sơ đồ con của phép tách yêu cầu

52

## Kết luận

- Tầm quan trọng của thiết kế CSDL
  - ảnh hưởng đến chất lượng dữ liệu lưu trữ
  - Hiệu quả của việc khai thác dữ liệu
- Mục đích của thiết kế CSDL:
  - Tránh dư thừa dữ liệu
  - Tránh dị thường dữ liệu khi thêm/xoá/sửa đổi
  - Hiệu quả trong tìm kiếm
- Đưa về các dạng chuẩn
  - 2NF: giảm bớt sự dư thừa để tránh các dị thường khi cập nhật
  - 3NF: tránh các dị thường khi thêm/xoá

53



54



## Lời hay ý đẹp

---

"Nếu anh thấy một gia đình hạnh phúc,  
anh nên tin rằng ở trong gia đình có  
một người đàn bà biết quên mình."

(René Bazin)

55

## Tối ưu hóa câu truy vấn

Nguyễn Hồng Phương

[phuongnh-fit@mail.hut.edu.vn](mailto:phuongnh-fit@mail.hut.edu.vn)

Bộ môn Hệ thống thông tin  
Viện Công nghệ thông tin và Truyền thông  
Đại học Bách Khoa Hà Nội

1

## Nội dung

- Tổng quan về xử lý truy vấn
- Tối ưu hóa các biểu thức đại số quan hệ

NHP

2

## Tổng quan về xử lý truy vấn

- Xử lý một truy vấn bao gồm 3 bước chính:
  - Phân tích và Biên dịch câu truy vấn: Trong bước này, hệ thống phải dịch câu truy vấn từ dạng ngôn ngữ bậc cao thành một ngôn ngữ biểu diễn dữ liệu bên trong để máy tính có thể thao tác trên đó. Một biểu diễn bên trong thích hợp và hỗ trợ cho bước tối ưu hóa tiếp theo là biểu diễn bằng ngôn ngữ đại số quan hệ

NHP

3

## Tổng quan về xử lý truy vấn (tiếp)

- Tối ưu hóa câu truy vấn: Mục tiêu của bước tối ưu hóa là chọn ra một kế hoạch thực hiện câu truy vấn có chi phí thấp nhất.
  - Để thực hiện được điều này, trước tiên ta cần biến đổi 1 biểu thức ĐSQH đầu vào thành một biểu thức ĐSQH tương đương nhưng có thể xử lý được 1 cách hiệu quả và ít tốn kém hơn. Bước con đầu tiên này được gọi là tối ưu hóa đại số.
  - Tiếp theo đó, ta cần phải đặc tả các thuật toán đặc biệt tiến hành thực thi các phép toán, chọn 1 chỉ dẫn cụ thể nào đó để sử dụng.
  - Các dữ liệu thống kê về CSDL sẽ giúp ta trong quá trình xem xét và lựa chọn. Ví dụ như:

NHP

4

## Tổng quan về xử lý truy vấn (tiếp)

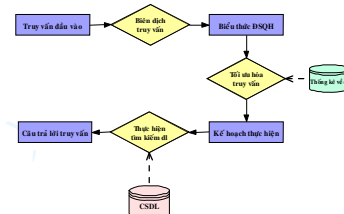
- Số bộ trong quan hệ
- Kích thước của một bộ
- Số khối (block) chứa các bộ của quan hệ
- Số bộ của quan hệ mà một khối có thể chứa
- Các thông tin về cơ chế truy nhập, chỉ dẫn trên quan hệ
- Chi phí cho việc thực hiện một truy vấn được đo bởi chi phí sử dụng tài nguyên như việc truy cập đĩa, thời gian CPU dùng để thực hiện một truy vấn.
- Trong chương này, chúng ta sẽ tập trung vào việc đánh giá các biểu thức đại số quan hệ chứ không đi vào chi tiết việc tính toán chi phí cho việc thực hiện đánh giá một truy vấn.

NHP

5

## Tổng quan về xử lý truy vấn (tiếp)

- Thực hiện đánh giá truy vấn: Từ một kế hoạch thực hiện có được do Trình tối ưu hóa cung cấp, hệ thống sẽ tiến hành thực hiện các thao tác trên dữ liệu trong CSDL và đưa ra câu trả lời cho truy vấn đó.



NHP

6

## Đánh giá biểu thức ĐSQH

- Sau bước phân tích và biên dịch, ta có một truy vấn được biểu diễn bằng một biểu thức đại số quan hệ bao gồm nhiều phép toán và tác động lên nhiều quan hệ khác nhau. Ta sẽ phải tiến hành đánh giá biểu thức này. Có 2 hướng tiếp cận để thực thi quá trình đánh giá biểu thức ĐSQH:

- Vật chất hóa (Materialize)
- Đường ống (Pipeline)

NHP

7

## Đánh giá biểu thức ĐSQH (tiếp)

- Vật chất hóa:** Trong cách tiếp cận này thì ta lần lượt đánh giá các phép toán theo một thứ tự thích hợp. Kết quả của việc đánh giá mỗi phép toán sẽ được lưu trong một quan hệ trung gian tạm thời để sử dụng làm đầu vào cho các phép toán tiếp theo.
- Điểm bất lợi của cách tiếp cận này là việc cần thiết phải xây dựng các quan hệ trung gian tạm thời nhất là khi các quan hệ này thường phải được ghi ra đĩa (trừ khi chúng có kích thước rất nhỏ). Mà việc đọc và ghi ra đĩa có chi phí khá lớn.

NHP

8

## Đánh giá biểu thức ĐSQH (tiếp)

- Đường ống:** Chúng ta có thể cải thiện hiệu quả đánh giá truy vấn bằng cách làm giảm bớt số lượng các quan hệ trung gian tạm thời được tạo ra. Điều này có thể đạt được nhờ việc kết hợp một vài phép toán quan hệ vào một đường ống của các phép toán. Trong đường ống thì kết quả của một phép toán được chuyển trực tiếp cho phép toán tiếp theo mà không cần phải lưu lại trong quan hệ trung gian.
- Rõ ràng, cách tiếp cận thứ hai sẽ hạn chế được nhược điểm của cách tiếp cận đầu tiên, nhưng có những trường hợp, ta bắt buộc phải vật chất hóa chứ không dùng đường ống được.

NHP

9

## Đánh giá biểu thức ĐSQH (tiếp)

- Ví dụ: Chúng ta có một biểu thức đại số quan hệ gồm 2 phép toán: kết nối và chiếu.
- Trong cách tiếp cận vật chất hóa, xuất phát từ phép toán ở mức thấp nhất là phép kết nối tự nhiên, kết quả của phép kết nối này sẽ được lưu trong một quan hệ trung gian. Sau đó, đọc từ quan hệ trung gian này để tiến hành chiếu lấy kết quả mong muốn.
- Trong cách tiếp cận đường ống, khi một bộ được sinh ra trong phép kết nối 2 quan hệ, bộ này sẽ được chuyển trực tiếp đến phép chiếu để xử lý và kết quả được ghi vào quan hệ đầu ra. Quan hệ kết quả sẽ được tạo lập một cách trực tiếp.

NHP

10

## Tối ưu hóa các biểu thức ĐSQH

- Mục tiêu là tổ chức lại trình tự thực hiện các phép toán trong biểu thức để giảm chi phí thực hiện đánh giá biểu thức đó.
- Trong quá trình tối ưu hóa, ta biểu diễn một biểu thức ĐSQH dưới dạng một cây toán tử. Trong cây thì các nút lá là các quan hệ có mặt trong biểu thức, các nút trong là các phép toán trong biểu thức
- Ví dụ: Đưa ra tên hãng cung ứng mặt hàng có mã là 'P1':  
`Select sname From S, SP Where S.sid = SP.sid And pid = 'P1'`
- Biểu thức ĐSQH tương ứng là?
- Cây toán tử tương ứng là?

NHP

11

## Các chiến lược tối ưu tổng quát

- Đẩy phép chọn và phép chiếu xuống thực hiện sớm nhất có thể: vì hai phép toán này giúp làm giảm kích thước của quan hệ trước khi thực hiện các phép toán 2 ngôi
- Nhóm dãy các phép chọn và chiếu: Sử dụng chiến lược này nếu như có một dãy các phép chọn hoặc dãy các phép chiếu trên cùng một quan hệ
- Kết hợp phép chọn và tích Đề các thành phép kết nối: Nếu kết quả của một phép tích Đề các là đối số của 1 phép chọn có điều kiện chọn là phép so sánh giữa các thuộc tính trên 2 quan hệ tham gia tích Đề các thì ta nên kết hợp 2 phép toán thành phép kết nối.
- Tìm các biểu thức con chung trong biểu thức đại số quan hệ để đánh giá chỉ một lần

NHP

12



### Các chiến lược tối ưu tổng quát (tiếp)

5. Xác định các phép toán có thể được đưa vào đường ống và thực hiện đánh giá chúng theo đường ống
6. Xử lý các tệp dữ liệu trước khi tiến hành tính toán: Tạo lập chỉ dẫn hay sắp xếp tệp dữ liệu có thể góp phần làm giảm chi phí của các phép tính trung gian
7. Ước lượng chi phí và lựa chọn thứ tự thực hiện: Đối với mỗi câu truy vấn có thể có nhiều cách khác nhau để thực hiện, với việc ước lượng chi phí (số phép tính, tài nguyên sử dụng, dung tích bộ nhớ, thời gian thực hiện ...) ta có thể chọn cách đánh giá biểu thức ĐSQH có chi phí nhỏ nhất.

NHP

13

### Các phép biến đổi tương đương biểu thức ĐSQH

- Hai biểu thức ĐSQH  $E_1$  và  $E_2$  là tương đương nếu chúng cho cùng một kết quả khi áp dụng trên cùng một tập các quan hệ
- Trong phần này, ta có các ký hiệu dạng sau:
  - $E_1, E_2, E_3, \dots$  là các biểu thức đại số quan hệ
  - $F_1, F_2, F_3, \dots$  là các điều kiện chọn hoặc là các điều kiện kết nối
  - $X_1, X_2, \dots, Y, Z, U_1, U_2, \dots$  là các tập thuộc tính

NHP

14

### Các phép biến đổi tương đương biểu thức ĐSQH (tiếp)

1. Quy tắc kết hợp của phép tích Đề các và kết nối

$$(E_1 \times E_2) \times E_3 \equiv E_1 \times (E_2 \times E_3)$$

$$(E_1 * E_2) * E_3 \equiv E_1 * (E_2 * E_3)$$

$$(E_1 \bowtie_{F_1} E_2) \bowtie_{F_2} E_3 \equiv E_1 \bowtie_{F_1} (E_2 \bowtie_{F_2} E_3)$$

- Quy tắc này sử dụng cho chiến lược số 7. Thứ tự thực hiện các phép kết nối hay tích Đề các là rất quan trọng vì kích thước của quan hệ trung gian có thể rất lớn nếu không cân nhắc kỹ. Lựa chọn thứ tự thực hiện các phép toán này thì tùy thuộc vào kích thước của các quan hệ tham gia phép toán và cả ngữ nghĩa của quan hệ (mối liên hệ)

NHP

15

### Các phép biến đổi tương đương biểu thức ĐSQH (tiếp)

- VD:  $S * SP * P$  có thể được thực hiện theo 3 thứ tự như sau

$$1) (S * SP) * P$$

$$2) (S * P) * SP$$

$$3) S * (SP * P)$$

Xét theo ngữ nghĩa  $S, P$  không kết nối được nên (1) và (3) là tốt hơn (2). Xét về kích thước thì (3) tốt hơn (1) vì  $S$  có 4 thuộc tính còn  $P$  có 3 thuộc tính, tuy nhiên, cũng còn tùy thuộc vào lực lượng của 2 quan hệ  $S$  và  $P$  nữa

NHP

16

### Các phép biến đổi tương đương biểu thức ĐSQH (tiếp)

2. Quy tắc giao hoán trong phép tích Đề các và kết nối

$$E_1 \times E_2 \equiv E_2 \times E_1$$

$$E_1 * E_2 \equiv E_2 * E_1$$

$$E_1 \bowtie_F E_2 \equiv E_2 \bowtie_F E_1$$

3. Quy tắc đối với dãy các phép chiếu

$$\Pi_{X_1}(\Pi_{X_2} \dots \Pi_{X_n}(E) \dots) \equiv \Pi_{X_1}(E)$$

$$X_1 \subseteq X_2 \subseteq \dots \subseteq X_n$$

4. Quy tắc đối với dãy các phép chọn

$$c_{F_1}(c_{F_2} \dots c_{F_n}(E) \dots) \equiv c_{F_1 \wedge F_2 \wedge \dots \wedge F_n}(E)$$

NHP

17

### Các phép biến đổi tương đương biểu thức ĐSQH (tiếp)

5. Quy tắc giao hoán phép chọn và phép chiếu

$$\Gamma_X(c_F(E)) \equiv c_F(\Gamma_X(E))$$

Quy tắc này áp dụng khi  $F$  là điều kiện xác định được trên tập thuộc tính  $X$ . Tổng quát hơn ta có:

$$\Gamma_X(c_F(E)) \equiv \Gamma_X(c_F(\Gamma_{XY}(E)))$$

NHP

18



## Các phép biến đổi tương đương biểu thức ĐSQH (tiếp)

### 6. Quy tắc đối với phép chọn và phép tích Đề các

- Ta ký hiệu:
  - $E_1(U_1)$  có nghĩa là biểu thức  $E_1$  xác định trên tập thuộc tính  $U_1$
  - $F_1(U_1)$  có nghĩa là điều kiện chọn  $F_1$  xác định trên tập thuộc tính  $U_1$
  - Quy tắc biến đổi liên quan đến phép chọn và tích Đề các được phát biểu như sau:
- $c_F(E_1(U_1) \times E_2(U_2))$  tương đương với:
  - $c_{F_1}(E_1) \times E_2$  trong trường hợp  $F = F_1(U_1)$
  - $c_{F_1}(E_1) \times c_{F_2}(E_2)$  trong trường hợp  $F = F_1(U_1) \wedge F_2(U_2)$
  - $c_{F_2}(c_{F_1}(E_1) \times E_2)$  trong trường hợp  $F = F_1(U_1)$

NHP

19

## Các phép biến đổi tương đương biểu thức ĐSQH (tiếp)

### 7. Quy tắc đối với phép chọn và phép hợp:

$$c_F(E_1 \cup E_2) \equiv c_F(E_1) \cup c_F(E_2)$$

### 8. Quy tắc đối với phép chọn và phép trừ:

$$c_F(E_1 - E_2) \equiv c_F(E_1) - c_F(E_2)$$

NHP

20

## Các phép biến đổi tương đương biểu thức ĐSQH (tiếp)

### 9. Quy tắc đối với phép chiếu và tích Đề các:

$$\Gamma_X(E_1(U_1) \times E_2(U_2)) \equiv \Gamma_Y(E_1) \times \Gamma_Z(E_2)$$

$$X = YZ, Y \subset U_1, Z \subset U_2$$

### 10. Quy tắc đối với phép chiếu và phép hợp:

$$\Gamma_X(E_1 \cup E_2) \equiv \Gamma_X(E_1) \cup \Gamma_X(E_2)$$

NHP

21

## Ví dụ

- Tìm tên hãng cung ứng ít nhất 1 mặt hàng màu đỏ hoặc màu xanh  
`SELECT sname FROM S, P, SP`  
`WHERE S.sid = SP.sid AND P.pid = SP.pid`  
`AND (colour = 'Red' OR colour = 'Green');`
- Biểu thức đại số quan hệ tương đương với câu truy vấn trên là:

$$\Gamma_{sname}(c_{S.sid=SP.sid \wedge P.pid=SP.pid \wedge (colour='Red' \vee colour='Green')}(S \times SP \times P))$$

NHP

22



NHP

23

## Lời hay ý đẹp

"Phẩm cách chân chính của con người là ở trong cách họ sống chứ không phải ở cái họ có"

Blackie

NHP

24

## An toàn và toàn vẹn dữ liệu

Nguyễn Hồng Phương

[phuongnh-fit@mail.hut.edu.vn](mailto:phuongnh-fit@mail.hut.edu.vn)

Bộ môn Hệ thống thông tin  
Viện Công nghệ thông tin và Truyền thông  
Đại học Bách Khoa Hà Nội

1

## Nội dung

- An toàn dữ liệu
  - Xác minh người sử dụng
  - Kiểm tra quyền truy nhập của người sử dụng
  - Các câu lệnh an toàn dữ liệu trong SQL
- Toàn vẹn dữ liệu
  - Các ràng buộc toàn vẹn trong SQL
  - Điều khiển tương tranh

2

## An toàn dữ liệu

- **Định nghĩa:** Tính an toàn dữ liệu là sự bảo vệ dữ liệu trong cơ sở dữ liệu chống lại những truy nhập, sửa đổi hay phá hủy bất hợp pháp.
- Người sử dụng hợp pháp là những người sử dụng được cấp phép, được ủy quyền. Ngược lại là những người sử dụng bất hợp pháp.
- Để đảm bảo tính an toàn cho cơ sở dữ liệu, chúng ta cần có một cơ chế để quản lý người dùng cho hợp lý.
- Những nhóm người dùng khác nhau trong hệ CSDL có quyền sử dụng khác nhau đối với các dữ liệu trong CSDL.

3

## Các quyền truy nhập của người sử dụng

- Quyền đọc dữ liệu: được phép đọc một phần hay toàn bộ dữ liệu trong CSDL
- Quyền cập nhật dữ liệu: được phép sửa đổi một số giá trị nhưng không được xóa dữ liệu trong CSDL
- Quyền xóa dữ liệu: được phép xóa dữ liệu trong CSDL
- Quyền bổ sung dữ liệu: được phép thêm dữ liệu mới vào trong CSDL nhưng không được phép thay đổi dữ liệu
- Quyền tạo chỉ dẫn trên các quan hệ trong CSDL
- Quyền thay đổi sơ đồ cơ sở dữ liệu: thêm hay xóa các thuộc tính của các quan hệ trong CSDL
- Quyền loại bỏ quan hệ trong CSDL
- Quyền quản lý tài nguyên: được phép thêm các quan hệ mới vào CSDL

4

## Trách nhiệm của người quản trị hệ thống

- Để có thể phân biệt được người sử dụng trong hệ CSDL, người quản trị hệ thống phải có trách nhiệm:
  - Xác định các quyền cụ thể mà mỗi người sử dụng hay một nhóm người sử dụng được phép thực hiện, xác định vai trò và trách nhiệm của mỗi người sử dụng. Điều này được gọi chung là Phân quyền người sử dụng
  - Cung cấp một phương tiện cho người sử dụng để hệ thống có thể nhận biết được người sử dụng đó hay còn gọi là Xác minh người sử dụng

5

## Xác minh người sử dụng

- Để xác minh được người sử dụng, người ta có thể dùng các kỹ thuật sau:
  - Kỹ thuật dùng tài khoản và mật khẩu, mật khẩu cũng được bảo vệ bởi hệ thống một cách kỹ càng.
  - Kỹ thuật sử dụng các hàm kiểm tra cho người sử dụng: Hệ thống đưa cho người sử dụng một số ngẫu nhiên  $x$ , người sử dụng dùng một hàm  $F$  tính nhăm kết quả và đưa kết quả  $y = F(x)$  vào hệ thống. Trong lúc đó, hệ thống cũng tính toán và so sánh kết quả với  $y$ . Người sử dụng hợp pháp là người biết hàm biến đổi  $F$  và đưa vào giá trị  $y$  đúng.
  - Kỹ thuật dùng thẻ điện tử, thẻ thông minh.
  - Kỹ thuật sử dụng nhận dạng tiếng nói, vân tay v.v.

6

### Kiểm tra quyền truy nhập của người sử dụng

- Mỗi người sử dụng sẽ có một bộ hồ sơ do người quản trị thiết lập và được hệ thống quản lý, trong hồ sơ đó sẽ có chi tiết về các thao tác người sử dụng được phép thực hiện:
  - Phân quyền người sử dụng: Người quản trị hệ thống phải có trách nhiệm xác định khung nhìn để kiểm soát xem mỗi người sử dụng chỉ được truy nhập phần dữ liệu nào trong CSDL và có được các quyền nào trong số các quyền đọc, thêm, xóa, sửa đổi.
  - Xác định và kiểm soát sự lưu chuyển dữ liệu: Hệ thống phải bảo trì danh sách các quyền một cách chặt chẽ vì người sử dụng có thể được quyền lan truyền các quyền cho người sử dụng khác.

7

### Các câu lệnh an toàn dữ liệu trong SQL

- Câu lệnh tạo khung nhìn
- Câu lệnh phân quyền cho người sử dụng
- Câu lệnh thu hồi quyền của người sử dụng

8

### Câu lệnh tạo khung nhìn

- CREATE VIEW <Tên khung nhìn> [(d/s cột)] AS <Câu truy vấn>
- Danh sách các cột trong khung nhìn là phần không bắt buộc. Trong trường hợp người sử dụng muốn đặt tên khác cho các cột xuất hiện trong khung nhìn thì người sử dụng có thể chỉ ra tên các cột, dữ liệu trên cột thì tương ứng với các cột trong mệnh đề Select của câu truy vấn.

9

### Ví dụ câu lệnh tạo khung nhìn

- Cho cơ sở dữ liệu gồm 2 quan hệ:  
Nhânviên(Id,Hotên,ĐC,Lương,NămBD,Đánhgiá,PhòngCT)  
Phòng(PId, Tên, ĐC, Điệnthoại, Trưởngphòng)

- Câu lệnh tạo khung nhìn cho một nhân viên của phòng Khoa Học có thể được định nghĩa như sau:

```
CREATE VIEW NVKH(HotênNhânviên, Đíachiliênlạc) AS
SELECT Hotên,Đíachi FROM Nhânviên
WHERE PhòngCT IN
      (SELECT PId FROM Phòng WHERE Tên ='Khoa Học')
```

10

### Câu lệnh phân quyền cho NSD

- GRANT <D/s thao tác> ON <Đối tượng> TO <D/s người dùng> [WITH GRANT OPTION]
- <D/s thao tác>: có thể bao gồm 1 hay nhiều thao tác được liệt kê dưới đây:
  - Insert: chèn dữ liệu vào trong CSDL có sẵn nhưng không được thay đổi bất kỳ mục dữ liệu nào trong CSDL
  - Update: sửa đổi dữ liệu nhưng không được xóa dữ liệu
  - Delete: xóa dữ liệu trong CSDL
  - Select : tìm kiếm
  - Create: tạo lập các quan hệ mới
  - Alter: Thay đổi cấu trúc của quan hệ
  - Drop: Loại bỏ quan hệ
  - Read/Write: Đọc và Ghi

11

### Câu lệnh phân quyền cho NSD (tiếp)

- <Đối tượng>: bảng hoặc khung nhìn
- <D/s người dùng>: Một người hay một nhóm hay một danh sách người sử dụng. Từ khóa public được dùng thay thế cho mọi người sử dụng
- [With Grant Option] Nếu dùng từ khóa này trong câu lệnh phân quyền thì người dùng xuất hiện trong <D/s người dùng> có quyền được lan truyền các quyền vừa được tuyên bố cho những người dùng khác

12

### Ví dụ câu lệnh phân quyền cho NSD

- Trao quyền đọc, ghi, tìm kiếm, sửa đổi dữ liệu cho nhân viên tên Hoa của phòng Khoa học trên khung nhìn vừa tạo lập trong phần trước

```
GRANT read, write, select, update ON NVKH TO Hoa;
```

- Trao quyền cho trưởng phòng Khoa học – ông HungNC

```
GRANT read, write, select, update, delete ON NVKH TO HungNC WITH GRANT OPTION;
```

13

### Câu lệnh thu hồi quyền của NSD

- REVOKE <D/s thao tác> ON <Đối tượng> FROM <D/s người dùng> [RESTRICT/CASCADE]
- <D/s thao tác>, <Đối tượng>, <D/s người dùng> giống như đối với câu lệnh GRANT.
- Phần [RESTRICT/CASCADE] là chỉ ra cơ chế thu hồi với các quyền đã được người dùng trong <D/s người dùng> lan truyền

14

### Câu lệnh thu hồi quyền của NSD (tiếp)

- Nếu Restrict thì có nghĩa là chỉ hủy bỏ quyền của những người có trong danh sách, quyền đã được lan truyền cho người khác không bị thu hồi.
- Nếu dùng Cascade thì hủy bỏ quyền của người trong <D/s người dùng>, đồng thời kéo theo hủy bỏ quyền mà người dùng đó đã lan truyền cho những người khác.
- Ví dụ:

```
REVOKE update,delete ON NVKH FROM HungNC CASCADE
```

15

### Toàn vẹn dữ liệu

- Định nghĩa: Tính toàn vẹn dữ liệu là sự bảo vệ dữ liệu trong CSDL chống lại những sự sửa đổi, phá hủy vô căn cứ để đảm bảo tính đúng đắn và chính xác của dữ liệu.
- Các thao tác có thể ảnh hưởng đến tính đúng đắn của CSDL là thêm, xóa, sửa đổi.
- Để đảm bảo tính toàn vẹn dữ liệu, cần phải chỉ ra và duy trì những ràng buộc toàn vẹn liên kết với mỗi quan hệ. Các ràng buộc toàn vẹn cung cấp 1 phương tiện để đảm bảo rằng các thao tác được thực hiện bởi những người sử dụng hợp pháp không làm mất đi tính đúng đắn của CSDL.
- Trong hệ thống đa người dùng, để đảm bảo được toàn vẹn dữ liệu, hệ thống còn phải có được một trình điều khiển tương tranh để tránh đụng độ giữa các thao tác được đưa ra bởi những người sử dụng khác nhau tại cùng một thời điểm

16

### Các ràng buộc toàn vẹn trong SQL

- Các ràng buộc về khóa chính, khóa ngoài, kiểm tra trên miền sử dụng Check chúng ta đã đề cập đến khi nói về câu lệnh tạo bảng trong CSDL.
- Các khẳng định: Là một vị từ biểu thị một điều kiện mà CSDL phải luôn luôn thỏa mãn. Các khẳng định được tạo ra bằng câu lệnh:

```
CREATE ASSERTION <Tên khẳng định>  
CHECK <Vị từ>
```

17

### Ví dụ về khẳng định

- Số lượng mặt hàng được cung cấp bởi các hãng cổ số nhân viên < 50 phải nhỏ hơn 100:

```
CREATE ASSERTION KĐSốlượng CHECK  
NOT EXISTS  
(SELECT * FROM S WHERE numofemps  
< 50 AND sid IN  
(SELECT sid FROM SP WHERE quantity  
>= 100))
```

18

### Các ràng buộc toàn vẹn trong SQL (tiếp)

- Các kích hoạt: Là một thao tác được thực hiện một cách tự động khi có một thay đổi đối với CSDL. Kích hoạt là các cơ chế có ích để bảo động hoặc thực hiện những nhiệm vụ được định sẵn khi các điều kiện nhất định được thỏa mãn.
- Kích hoạt có thể được định nghĩa để hủy bỏ, hoặc kiểm tra và thực hiện một số các sự kiện do đó nó có thể được coi là một biện pháp để đảm bảo toàn vẹn dữ liệu.

19

### Ví dụ về kích hoạt

- Nhân viên(ID,Họ tên,Lương,Địa chỉ,Người quản lý)
- Một nhân viên bao giờ cũng có lương ít hơn lương người trưởng phòng, điều kiện này phải được kiểm tra khi thêm bộ dữ liệu.

```
DEFINE TRIGGER ThemNV ON INSERT Nhânviên
IF Nhânviên.Lương > (SELECT E.Lương FROM
Nhânviên AS E WHERE E.ID =
Nhânviên.Ngườiquảnlý)
THEN ABORT;
```

20

### Điều khiển tương tranh

- Trong hệ CSDL đa người dùng, hệ thống cần đưa ra giải pháp chống đụng độ giữa các giao dịch (một dãy các thao tác) được đưa ra bởi những người dùng khác nhau để tránh việc một đối tượng dữ liệu nào đó bị làm mất tính đúng đắn trong quá trình cập nhật.

21

### Các kỹ thuật điều khiển tương tranh

- **Kỹ thuật dùng khóa:** Khi một giao dịch cần dữ liệu nào thì xin hệ điều hành một khóa trên phần dữ liệu đó, các giao dịch khác phải đợi đến khi giải phóng khóa mới được sử dụng phần dữ liệu đó. Có thể người ta sử dụng các loại khóa khác nhau ví dụ như khóa đọc – cho phép nhiều giao dịch đọc cùng 1 lúc, khóa ghi – chỉ 1 giao dịch có được tại một thời điểm.
- **Kỹ thuật gán nhãn thời gian:** Mỗi giao dịch được gán một nhãn T theo thời gian, giao dịch nào cần được ưu tiên thì được gán nhãn thời gian nhỏ hơn và được thực hiện trước. Kỹ thuật này giúp đưa yêu cầu đồng thời về thực hiện tuần tự.

22



23

### Lời hay ý đẹp

"Khi nói sự thật bạn sẽ không phải nhớ mình vừa nói gì, mà bạn cũng không bao giờ quên những gì mình vừa nói"

S.Raybum

24