

Chương 5. CƠ SỞ DỮ LIỆU

- 5.1. Định nghĩa và thuật ngữ
- 5.2. Các loại cơ sở dữ liệu
- 5.3. Hệ quản trị cơ sở dữ liệu
- 5.4. Cơ sở dữ liệu quan hệ
- 5.5. Sự phát triển của các hệ cơ sở dữ liệu

5.1. Định nghĩa và thuật ngữ

► *Dữ liệu* (*data*): sự biểu diễn của các đối tượng và sự kiện được ghi nhận và được lưu trữ trên các phương tiện của máy tính.

Dữ liệu có cấu trúc: số, ngày, chuỗi ký tự, ...

Dữ liệu không có cấu trúc: hình ảnh, âm thanh, đoạn phim, ...

Các đối tượng biểu diễn dữ liệu có thể được tìm thấy ở đâu ?

Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & more - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.amazon.com/

Google Search

amazon.com

Hello, Ruoming Jin. We have [recommendations](#) for you. (Not Ruoming?)

FREE 2-Day Shipping. No Minimum Purchase

Ruoming's Amazon.com Today's Deals Gifts & Wish Lists Gift Cards Your Account Help

Search All Departments GO Cart Your Lists

Shop All Departments

- Books
- Movies, Music & Games
- Digital Downloads
- Computers & Office
- Electronics
- Home & Garden
- Grocery, Health & Beauty
- Toys, Kids & Baby
- Apparel, Shoes & Jewelry
- Sports & Outdoors
- Tools, Auto & Industrial

100% More Books, Newspapers, and Blogs

We've doubled our selection since launch to over 200,000 titles—all available wirelessly in under 60 seconds:

- New York Times bestsellers are \$9.99, unless otherwise marked
- Expanded newspaper selection, including *The New York Times*, *Wall Street Journal*, and *Financial Times*
- More magazines and blogs, from *Newsweek* to *Time* to *Gawker*

amazonkindle Learn more

Amazon Daily BLOG 5 posts since yesterday Posts for Ruoming

Treat Yourself

LOOK INSIDE! Combinatorial Optimization

From the reviews: What a book! Actually it is three books which come... [Read more](#)

\$135.00 **\$107.51**

See more in your Wish List

Check This Out

- APRIL 15** Tax Central Find tax software, books, and more
- 2009** Cast Your Vote See our Top 10 New Year's Resolutions. Which one's for you?
- Holiday Returns Visit our self-service Online Returns Center.
- For the Photography Fan Zoom in on top

Frequently Bought with Items in Your Wish List

LOOK INSIDE! Theory of Linear and Integer Programming Paperback by Alexander

LOOK INSIDE! Combinatorial Optimization... Paperback by Christos H.

Approximation Algorithms Hardcover by Vijay V. Vazirani

Get Your Motor Running Explore Our New Motorcycle & ATV Store

Shop now at Amazon.com

Done

start C:\Documents and... Microsoft PowerPo... Inbox for jin@cs.k... Amazon.com: Onli... EN 8:46 PM

Google Maps - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://maps.google.com/

Google Jim Gray Search

W Jim Gray (computer scientist) - Wi... HIV protein breaks biological clock... Google Maps

Web Images Maps News Shopping Gmail more Sign in | Help

Google Maps Find businesses, addresses and places of interest. [Learn more.](#) Search Maps Show search options

Get Directions My Maps

New! Add driving directions to your website.

Set default location

Browse popular maps

Follow the Sundance Film Festival

See more maps

Put your business on Google Maps

Display your ads on Google Maps

Print Send Link

Traffic More... Map Satellite Terrain

500 mi 500 km

©2009 Google - Map data ©2009 Tele Atlas, LeadDog Consulting - [Terms of Use](#)

Find: interg Next Previous Highlight all Match case

Done

start C:\Docu... Inbox fo... Google ... C++ Pr... lecture1 Downloads EN 9:42 PM

Entrez cross-database search - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.ncbi.nlm.nih.gov/gquery/gquery.fcgi?tool=toolbar

Google home database Search Bookmarks PageRank Check AutoLink AutoFill Send to Genome Settings

Entrez cross-database search HIV protein breaks biological clock...

NCBI Entrez, The Life Sciences Search Engine

HOME SEARCH SITE MAP PubMed All Databases Human Genome GenBank Map Viewer BLAST

Search across databases GO Clear Help

Welcome to the Entrez cross-database search page

PubMed: biomedical literature citations and abstracts	Books: online books
PubMed Central: free, full text journal articles	OMIM: online Mendelian Inheritance in Man
Site Search: NCBI web and FTP sites	OMIA: online Mendelian Inheritance in Animals
Nucleotide: Core subset of nucleotide sequence records	dbGaP: genotype and phenotype
EST: Expressed Sequence Tag records	UniGene: gene-oriented clusters of transcript sequences
GSS: Genome Survey Sequence records	CDD: conserved protein domain database
Protein: sequence database	3D Domains: domains from Entrez Structure
Genome: whole genome sequences	UniSTS: markers and mapping data
Structure: three-dimensional macromolecular structures	PopSet: population study data sets
Taxonomy: organisms in GenBank	GEO Profiles: expression and molecular abundance

Done

start C:\Documen... Microsoft Po... Inbox for jin... Entrez cross... C++ Progra... EN 9:13 PM

Cơ sở dữ liệu

- ❖ **Cơ sở dữ liệu** (*database*) tập hợp có tổ chức các dữ liệu có liên quan luận lý với nhau.
- ▶ **Có tổ chức** (*organized*): user dễ dàng lưu trữ, thao tác và truy xuất dữ liệu.
- ▶ **Có liên quan luận lý** (*logically related*): dữ liệu mô tả một lãnh vực mà nhóm user quan tâm và được dùng để trả lời các câu hỏi liên quan đến lãnh vực này.

Market Shares

From 2004

www.computerworld.com

- IBM: 35% market with \$2.5BN in sales
- Oracle: 33% market with \$2.3BN in sales
- Microsoft: 19% market with \$1.3BN in sales

Các ứng dụng quản lý dữ liệu đầu tiên

- Hệ thống đặt vé máy bay (*Airline Reservation Systems*)
 - single passenger reservations; Information about flights and airports; Information about ticket prices and tickets restrictions.
- Hệ thống hoạt động ngân hàng (*Banking Systems*)
 - accounts, customers, loans, mortgages, balances, etc. Failures are not tolerable. Concurrent access must be provided
- Hồ sơ công ty (*Corporate Records*)
 - sales, accounts, bill of materials records, employee and their dependents

Các ứng dụng quản lý dữ liệu hiện nay

- Client – Server architecture
 - DBMS serves as a server and client queries are sent to servers
 - Where to locate servers
- Multimedia Applications
- Multidatabase Applications
- Data Warehouses
- It's everywhere now !!!

Thông tin

❖ **Thông tin** (*information*) là dữ liệu đã được xử lý để làm tăng sự hiểu biết của user.

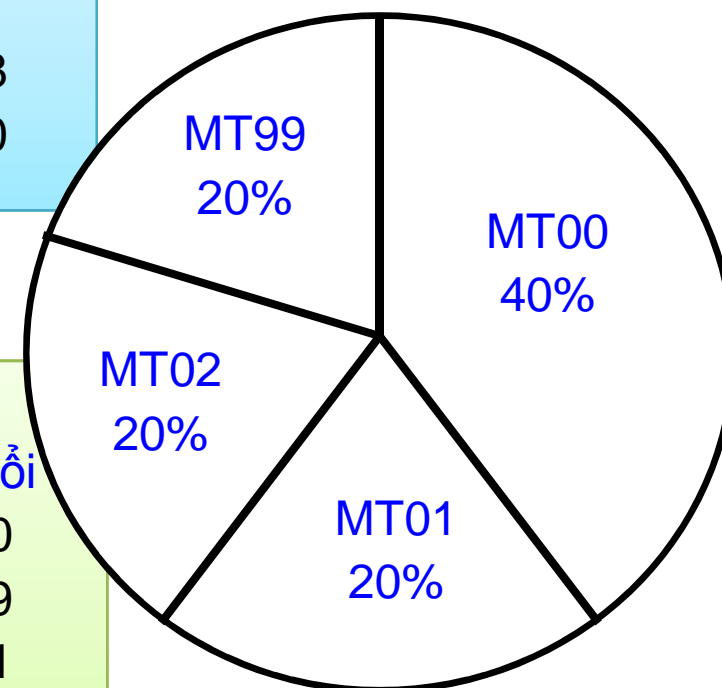
- ▶ Dữ liệu trong ngữ cảnh.
- ▶ Dữ liệu được tổng hợp / xử lý.

Mẫu dữ liệu ví dụ

Dữ liệu

50010273	Nguyễn Trung Tiến	MT00	20
50100298	Lê Việt Hùng	MT01	19
59900012	Trần Hùng Việt	MT99	21
50200542	Hồ Xuân Hương	MT02	18
50000075	Bùi Đức Duy	MT00	20

*Thông tin: dữ liệu
được tổng hợp /
xử lý*



Thông tin: dữ liệu trong ngữ cảnh

Mã sinh viên	Họ và tên sinh viên	Lớp	Tuổi
50010273	Nguyễn Trung Tiến	MT00	20
50100298	Lê Việt Hùng	MT01	19
59900012	Trần Hùng Việt	MT99	21
50200542	Hồ Xuân Hương	MT02	18
50000075	Bùi Đức Duy	MT00	20

Siêu dữ liệu

❖ Siêu dữ liệu (*metadata*)

- ▶ là dữ liệu dùng để mô tả các tính chất / đặc tính của dữ liệu khác (dữ liệu về dữ liệu).
- ▶ Các đặc tính: định nghĩa dữ liệu, cấu trúc dữ liệu, qui tắc / ràng buộc.

Siêu dữ liệu cho Sinh_viên

Data Item		Value			
Name	Type	Length	Min	Max	Description
MaSV	Character	8			Ma sinh vien
Hoten	Character	30			Ho ten sinh vien
Lop	Character	3			Lop
Tuoi	Number	2	17	25	Tuoi

Hệ thống xử lý tập tin

❖ Hệ thống xử lý tập tin (*file processing system*)

- ▶ là tập hợp các chương trình dùng để lưu trữ, thao tác và truy xuất các tập tin dữ liệu có kích thước lớn.
- ▶ Các tập tin dữ liệu được lưu trữ trong các thư mục (*folder*).

Thành phần của hệ thống xử lý tập tin

❖ Các thành phần của hệ thống xử lý tập tin

▶ Phần cứng: các máy tính.

▶ Phần mềm:

Hệ điều hành

Các tiện ích

Các tập tin

Các chương trình quản lý tập tin

Các chương trình ứng dụng tạo các báo cáo từ các dữ liệu được lưu trữ trong các tập tin.

▶ Con người: người quản lý, chuyên gia, người lập trình, người sử dụng cuối cùng.

▶ Các thủ tục: các lệnh và các qui tắc chi phối việc thiết kế và sử dụng các thành phần của phần mềm.

▶ Dữ liệu: tập hợp các sự kiện.

Quản lý dữ liệu của hệ thống tập tin

❖ Quản lý dữ liệu của hệ thống tập tin

▶ Ngôn ngữ lập trình: 3GL (*third-Generation Language*).

Làm gì? Làm như thế nào?

Các ngôn ngữ:

- ▶ COBOL (*COmmon Business-Oriented Language*)
- ▶ BASIC (*Beginner's All-purpose Symbolic Instruction Code*)
- ▶ FORTRAN (*FORmula TRANslation*)

▶ Các chương trình xử lý tập tin

- ▶ Tạo cấu trúc tập tin.
- ▶ Thêm dữ liệu vào tập tin.
- ▶ Xóa dữ liệu của tập tin
- ▶ Sửa dữ liệu của tập tin.
- ▶ Liệt kê dữ liệu của tập tin.

Nhược điểm hệ thống xử lý thông tin

- ▶ Phụ thuộc dữ liệu – chương trình (*Program-Data Dependence*)
Chương trình ứng dụng cần duy trì siêu dữ liệu (phần mô tả) của các tập tin mà chúng sử dụng.
- ▶ Dư thừa dữ liệu / Trùng lặp dữ liệu (*Data Redundancy / Duplication of Data*)
Các hệ thống / chương trình khác nhau có các bản dữ liệu riêng biệt của cùng dữ liệu.
- ▶ Hạn chế việc dùng chung dữ liệu
Mỗi ứng dụng có các tập tin riêng biệt, ít sử dụng chung dữ liệu với các ứng dụng khác.
- ▶ Thời gian phát triển lâu
Người lập trình phải thiết kế các dạng tập tin dữ liệu riêng và viết cách truy xuất tập tin cho mỗi ứng dụng mới.
- ▶ Chi phí bảo trì chương trình cao
Các nhược điểm nêu trên làm cho việc bảo trì chương trình gặp nhiều khó khăn, thường chiếm khoảng 80% ngân sách phát triển HTTT (*Information System*).

Phụ thuộc dữ liệu

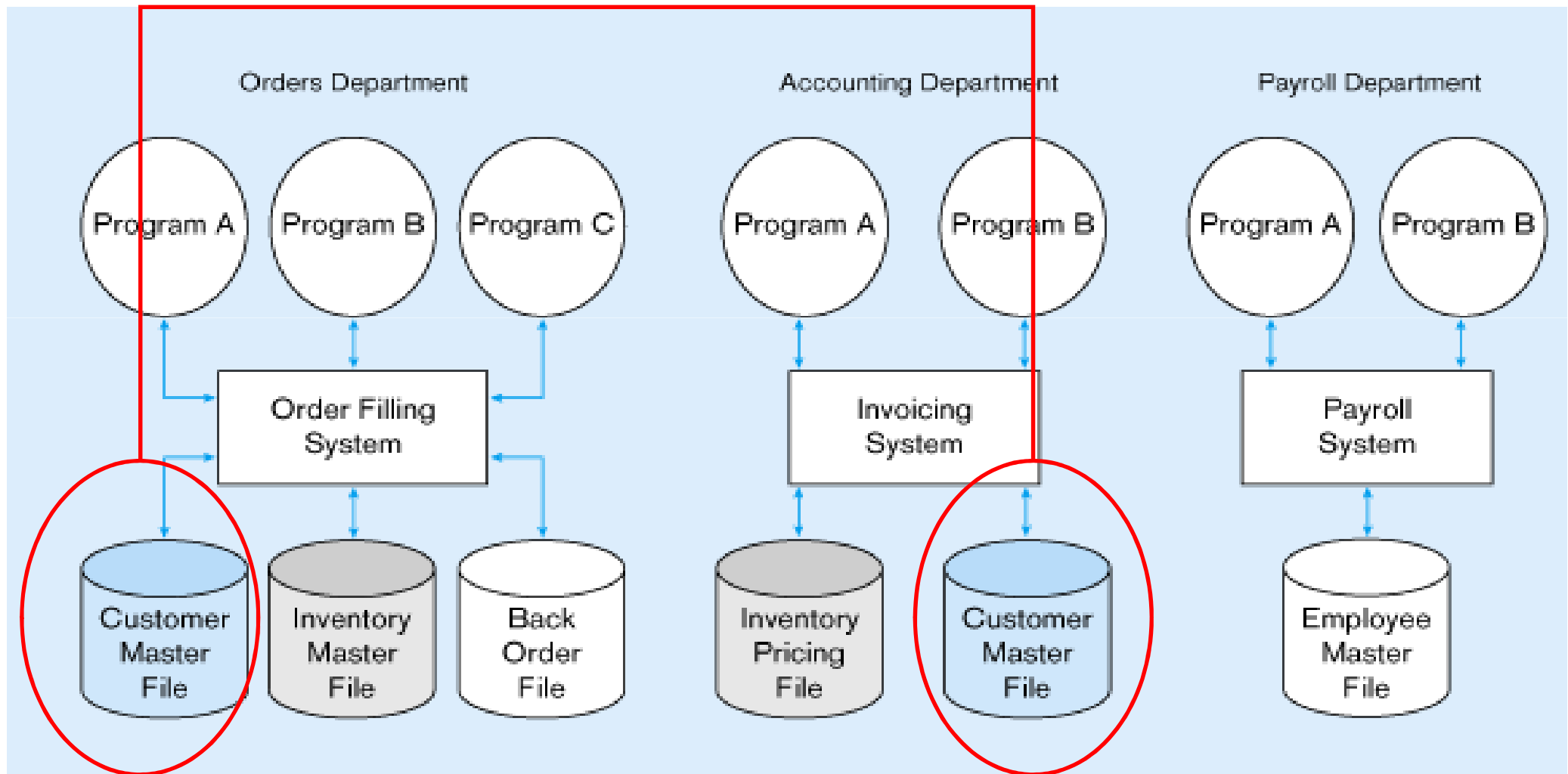
- ❖ Mỗi người lập trình phải duy trì dữ liệu riêng biệt.
- ❖ Mỗi chương trình ứng dụng phải có mã lệnh cho siêu dữ liệu của mỗi tập tin.
- ❖ Mỗi chương trình ứng dụng phải có các chương trình con xử lý để đọc, thêm, sửa và xóa dữ liệu.
- ❖ Không có các điều khiển chung và phối hợp.
- ❖ Các dạng thức tập tin không có cùng chuẩn.

Dư thừa dữ liệu

- ❖ Tốn vùng nhớ để lưu trữ dữ liệu dư thừa.
- ❖ Gây ra các vấn đề khó về bảo trì dữ liệu.
- ❖ Vấn đề chính:
 - ▶ Việc cập nhật dữ liệu của một tập tin có thể dẫn đến các mâu thuẫn dữ liệu.
 - ▶ Vi phạm tính toàn vẹn dữ liệu.

Minh họa dư thừa dữ liệu

Duplicate Data



Cách tiếp cận cơ sở dữ liệu

- ❖ Kho dữ liệu trung tâm chứa các dữ liệu dùng chung.
- ❖ Dữ liệu được quản lý bởi một đơn vị điều khiển (*controlling agent*).
- ❖ Dữ liệu được lưu trữ theo một dạng thức chuẩn và thích hợp.
- ❖ Cần phải có một hệ quản trị cơ sở dữ liệu.

Ưu điểm cách tiếp cận cơ sở dữ liệu

- ▶ Độc lập dữ liệu – chương trình (*data - program independence*).

DBMS chứa siêu dữ liệu (*metadata*), do đó các ứng dụng không cần quan tâm đến các dạng thức của dữ liệu.

DBMS quản lý các truy vấn và cập nhật dữ liệu, do đó ứng dụng không cần xử lý việc truy xuất dữ liệu.

- ▶ Giảm tối thiểu sự dư thừa dữ liệu (*data redundancy*).
- ▶ Nâng cao tính nhất quán (*data consistency*) / toàn vẹn dữ liệu (*data integrity*).

Ưu điểm cách tiếp cận cơ sở dữ liệu

- ▶ Nâng cao việc dùng chung dữ liệu (*data sharing*).
Những người sử dụng khác nhau có những cái nhìn khác nhau về dữ liệu.
- ▶ Tăng hiệu suất phát triển ứng dụng.
- ▶ Tuân thủ các tiêu chuẩn.
Tất cả các truy xuất dữ liệu đều được thực hiện theo cùng một cách.
- ▶ Nâng cao chất lượng của dữ liệu.
Các ràng buộc (*constraint*), các quy tắc hợp lệ của dữ liệu (*data validation rule*).

Ưu điểm cách tiếp cận cơ sở dữ liệu

- ▶ Nâng cao tính truy xuất và tính đáp ứng của dữ liệu.
Sử dụng ngôn ngữ truy vấn dữ liệu chuẩn (*SQL - Structured Query Language*).
- ▶ Giảm chi phí bảo trì chương trình.
- ▶ Bảo mật (*security*).
- ▶ Chép lưu (*backup*) và phục hồi (*recovery*).
- ▶ Điều khiển tương tranh (*concurrency control*).

Chi phí và rủi ro của cách tiếp cận DB

▶ Chi phí ban đầu

Chi phí cài đặt và quản lý

Chi phí chuyển đổi (*conversion cost*)

▶ Chi phí vận hành

Cần nhân viên mới có chuyên môn.

Cần phải chép lưu và phục hồi.

▶ Mâu thuẫn về mặt tổ chức

Rất khó thay đổi các thói quen cũ.

5.2. Các loại cơ sở dữ liệu

- ❖ **CSDL cá nhân**
 - ▶ *personal database*
 - ▶ CSDL riêng.
- ❖ **CSDL nhóm làm việc**
 - ▶ *workgroup database*
 - ▶ Mạng cục bộ (ít hơn 25 người sử dụng)
- ❖ **CSDL phòng ban**
 - ▶ *department database*
 - ▶ Mạng cục bộ (từ 25 đến 100 người sử dụng)
- ❖ **CSDL xí nghiệp**
 - ▶ *enterprise database*
 - ▶ Mạng diện rộng (hàng trăm hoặc hàng ngàn người sử dụng)

Các loại cơ sở dữ liệu

<i>Type of Database</i>	<i>Typical Number of Users</i>	<i>Typical Architecture</i>	<i>Typical Size of Database</i>
Personal	1	Desktop/laptop computer, PDA	Megabytes
Workgroup	5–25	Client/server (two-tier)	Megabytes–gigabytes
Department	25–100	Client/server (three-tier)	Gigabytes
Enterprise	>100	Client/server (distributed or parallel server)	Gigabytes–terabytes
Internet	>1000	Web server and application servers	Megabytes–gigabytes

5.3. Hệ quản trị cơ sở dữ liệu

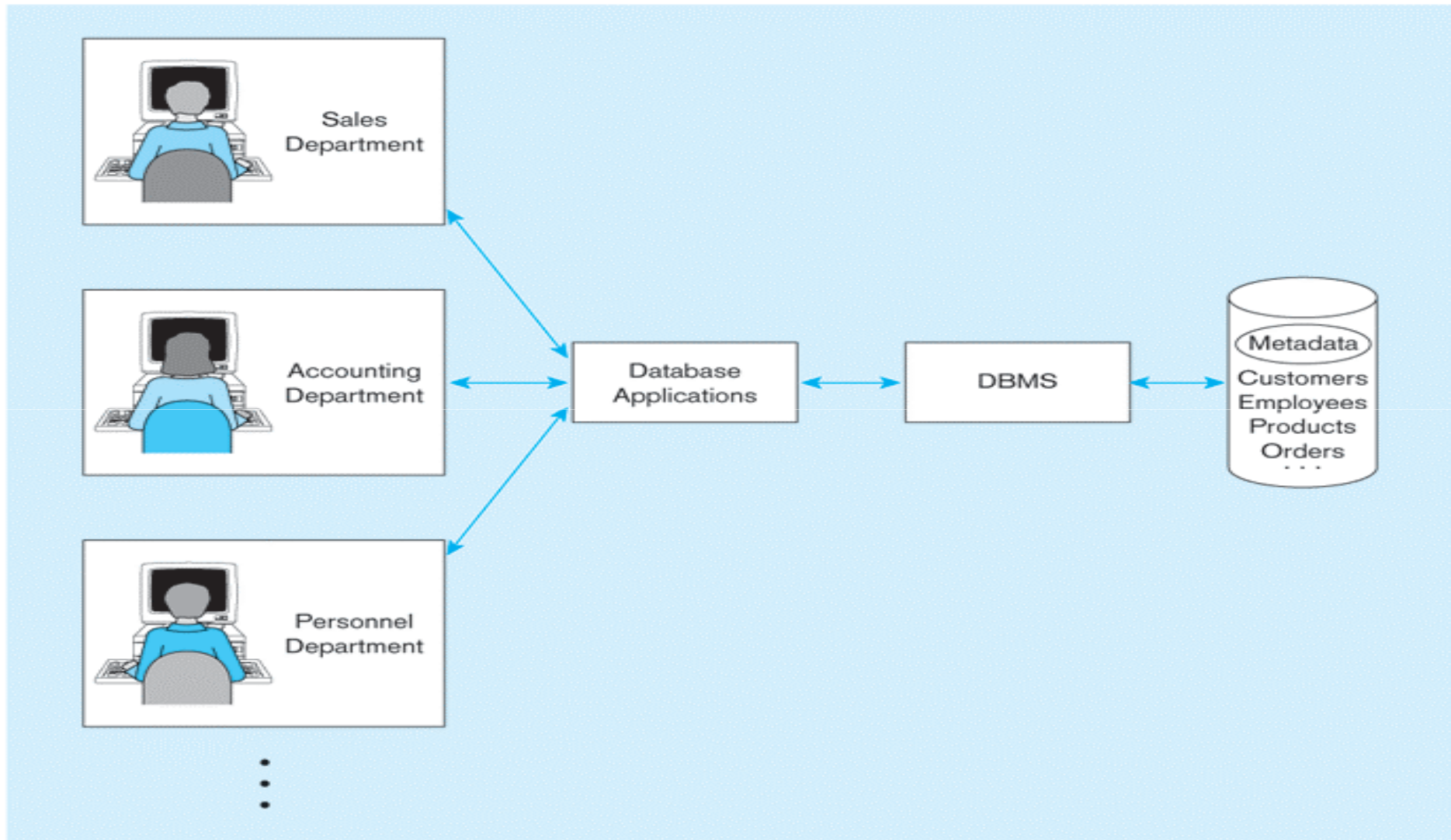
❖ Hệ quản trị CSDL

- ▶ DBMS – *DataBase Management System*
- ▶ *Hệ quản trị CSDL* là tập hợp các chương trình dùng để quản lý cấu trúc và dữ liệu của CSDL và điều khiển truy xuất dữ liệu trong CSDL.
- ▶ Cho phép người sử dụng định nghĩa, tạo lập và bảo trì CSDL và cung cấp các truy xuất dữ liệu.

Các ứng dụng của DBMSs

- DB2 (IBM), SQL Server (MS), Oracle, Sybase
- MySQL, PostgreSQL, ...

Hệ quản trị cơ sở dữ liệu



Các chức năng của DBMS

- ▶ Lưu trữ, truy xuất và cập nhật dữ liệu
 - Ngôn ngữ định nghĩa dữ liệu (*DDL - Data Definition Language*)
 - Ngôn ngữ thao tác dữ liệu (*DML - Data Manipulation Language*).
- ▶ Quản lý giao tác (*transaction management*).
- ▶ Điều khiển tương tranh (*concurrency control*)
- ▶ Chép lưu và phục hồi dữ liệu.
- ▶ Bảo mật dữ liệu
 - Ngôn ngữ điều khiển dữ liệu (*DCL - Data Control Language*).
- ▶ Hỗ trợ truyền thông dữ liệu.
- ▶ Duy trì tính toàn vẹn / nhất quán dữ liệu.
- ▶ Cung cấp các tiện ích.

5.4. Cơ sở dữ liệu quan hệ

- Quan hệ/ Khóa/ Thực thể
- Mô hình liên kết thực thể
- Lược đồ cơ sở dữ liệu
- Ngôn ngữ truy vấn
- Ràng buộc toàn vẹn

- **Cơ sở dữ liệu quan hệ** (*relational database*) bao gồm các bảng (quan hệ) biểu diễn các thực thể và các khóa chính / khóa ngoại biểu diễn các mối liên kết.

Quá trình thiết kế CSDL gồm 3 bước:

- **Phân tích yêu cầu** (*Requirements Analysis*): Khám phá những thông tin cần phải được lưu trữ, làm thế nào các thông tin được lưu trữ sẽ được sử dụng, vv
- **Thiết kế khái niệm** (*Conceptual Database Design*) Mô tả cấp cao của dữ liệu được lưu trữ (Mô hình liên kết thực thể - *ER model*)
- **Thiết kế cơ sở dữ liệu mức logic** (*Logical Database Design*) Dịch *ER model* thành một lược đồ cơ sở dữ liệu quan hệ (mô tả các bảng)

Ví dụ

- ✓ Đối với các diễn viên và đạo diễn, lưu tên của họ, địa chỉ, và sinh nhật
- ✓ Đối với các diễn viên, lưu trữ một bức ảnh
- ✓ Đối với các bộ phim, lưu trữ tên, năm sản xuất và kiểu (phim hài, phim kinh dị, vv)
- ✓ Có thể xác định đạo diễn và diễn viên trong mỗi bộ phim.
- ✓ Yêu cầu mỗi bộ phim có một đạo diễn.
- ✓ Có khả năng lưu trữ lương của nam diễn viên cho từng bộ phim

Ví dụ (tt)

- Một diễn viên có thể nhận được giải thưởng
 - lưu trữ thông tin về những người đã nhận được giải thưởng và bộ phim được trao giải, cùng với tên của các giải thưởng trong năm.
- Lưu trữ tên và số điện thoại của tổ chức đã trao giải thưởng.
 - Hai tổ chức khác nhau có thể cho một giải thưởng có cùng tên.
 - Một tổ chức duy nhất không đưa ra nhiều hơn một giải thưởng với một tên cụ thể trong mỗi năm.

Quan hệ

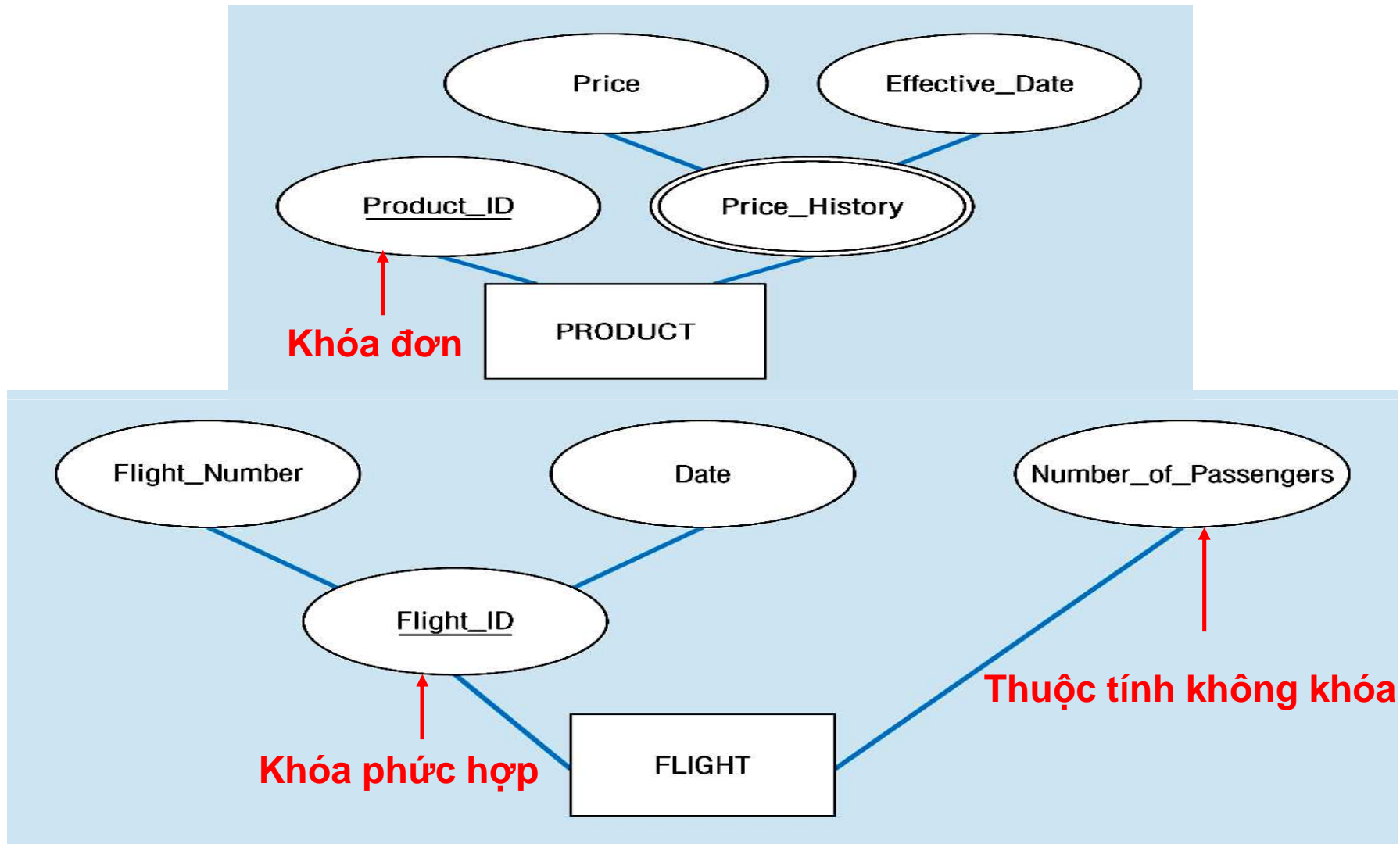
- ❖ **Quan hệ** (*relation*) là một bảng dữ liệu hai chiều bao gồm nhiều hàng (mẫu tin) và nhiều cột (thuộc tính hoặc vùng tin).
 - ▶ Mỗi hàng là duy nhất: không thể có hai hàng có cùng các giá trị ở tất cả vùng tin.
 - ▶ Thứ tự của các hàng là không quan trọng.
 - ▶ Thứ tự của các cột là không quan trọng.
 - ▶ Không phải mọi bảng đều là quan hệ.

Snum	Name	City
S1	Nguyễn Trung Tiến	SF
S2	Trần Thị Yến	LA
S3	Nguyễn Văn An	SF

Khóa

❖ Khóa (*key*)

- ▶ *Khóa quan hệ* là một tập nhỏ nhất các thuộc tính dùng để xác định duy nhất một hàng.
- ▶ Một khóa chỉ có một thuộc tính được gọi là *khóa đơn* (*simple key*).
- ▶ Một khóa có nhiều thuộc tính được gọi là *khóa phức hợp* (*composite key*).
- ▶ Khóa thường được sử dụng làm *chỉ mục* (*index*) của bảng dữ liệu để làm tăng tốc độ xử lý câu truy vấn.



- ▶ Một quan hệ phải có ít nhất một khóa và có thể có nhiều khóa.
- ▶ Các thuộc tính thuộc một khóa được gọi là *thuộc tính khóa* (*prime attribute*), các thuộc tính còn lại trong lược đồ quan hệ được gọi là các *thuộc tính không khóa* (*nonprime attribute*).
- ▶ Các thuộc tính khóa được gạch dưới.
- ▶ Các thuộc tính khóa không được có giá trị rỗng (*null value*).

Các loại khóa

- ▶ Tất cả các khóa của một quan hệ được gọi là *khóa dự tuyển* (*candidate key*).
- ▶ Một trong các khóa dự tuyển được chọn làm khóa tiêu biểu, khóa này được gọi là *khóa chính* (*primary key*).
- ▶ Một quan hệ chỉ có một khóa chính và có thể có nhiều khóa dự tuyển.
- ▶ Trong một quan hệ, một hoặc nhiều thuộc tính được gọi là *khóa ngoại* (*foreign key*) nếu chúng là khóa chính của một quan hệ khác.

Thực thể?

❖ Thể hiện thực thể (*entity instance*)

- ▶ *Thể hiện thực thể* là người, vị trí, đối tượng, sự kiện, khái niệm (thường tương ứng với một hàng của bảng).
- ▶ *Thực thể nên là*
 - đối tượng có nhiều thể hiện trong CSDL.
 - đối tượng có nhiều thuộc tính.
 - đối tượng cần được mô hình hóa.
- ▶ *Thực thể không nên là*
 - người sử dụng của hệ CSDL.
 - kết xuất của hệ CSDL (ví dụ bản báo cáo).
- ▶ Đặc điểm của thực thể là *tính phân biệt* (*distinctness*): có thể phân biệt giữa thực thể này với thực thể khác.

Kiểu và thuộc tính thực thể

❖ Kiểu thực thể (*entity type*)

- ▶ *Kiểu thực thể* là tập hợp các thực thể thuộc cùng một loại (thường tương ứng với một bảng).
- ▶ Được biểu diễn bằng *hình chữ nhật*.

❖ Thuộc tính (*attribute*)

- ▶ *Thuộc tính* là một đặc tính / tính chất của một kiểu thực thể (thường tương ứng với một vùng tin trong một bảng).
- ▶ Được biểu diễn bằng *hình bầu dục*.

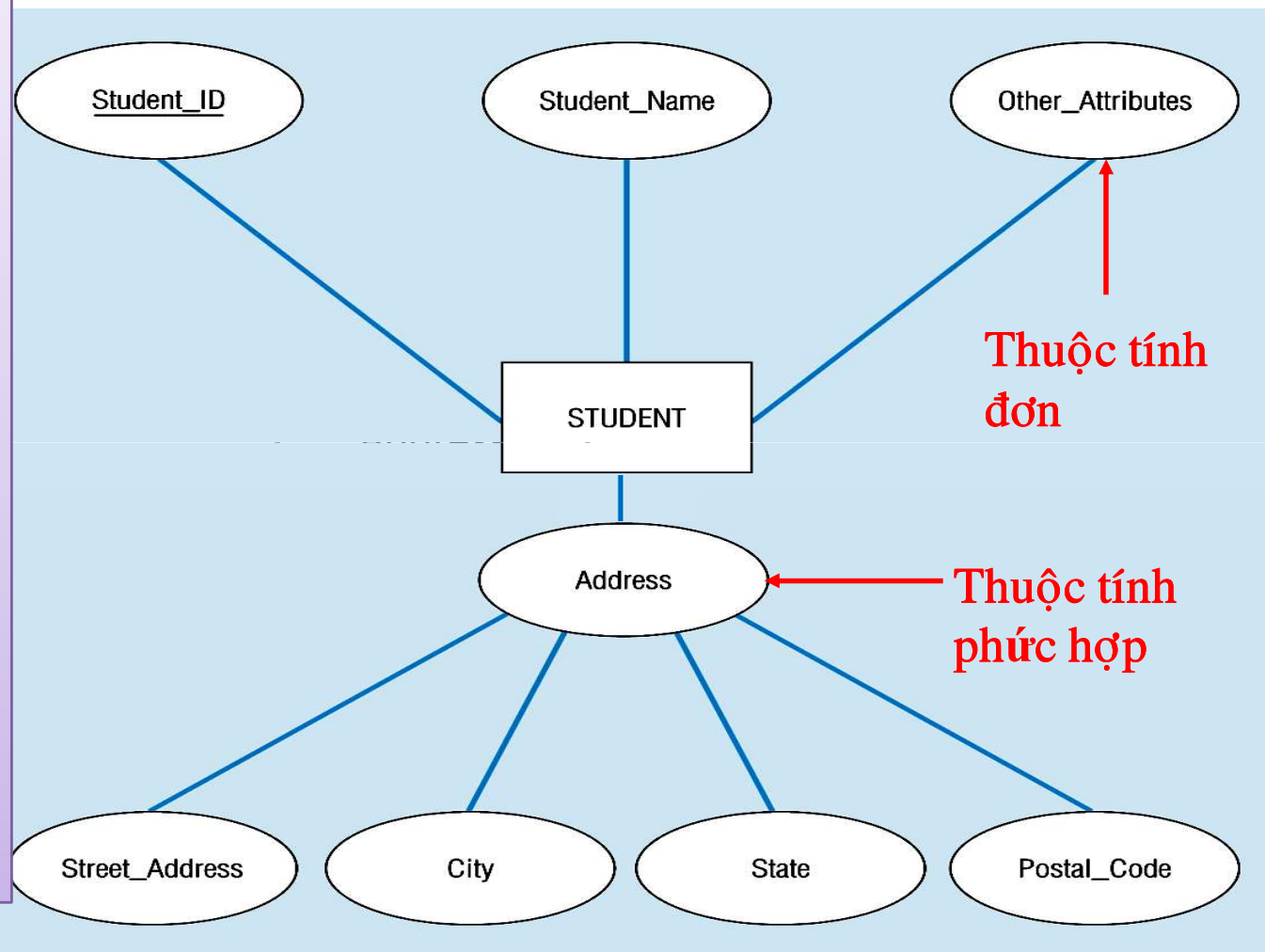
Các loại thuộc tính

❖ Các loại thuộc tính

- ▶ Thuộc tính bắt buộc và thuộc tính tùy chọn.
- ▶ Thuộc tính đơn và thuộc tính phức hợp.
- ▶ Thuộc tính đơn trị và thuộc tính đa trị.
- ▶ Thuộc tính chứa và thuộc tính dẫn xuất.
- ▶ Thuộc tính khóa và thuộc tính không khóa.

Phân loại thuộc tính

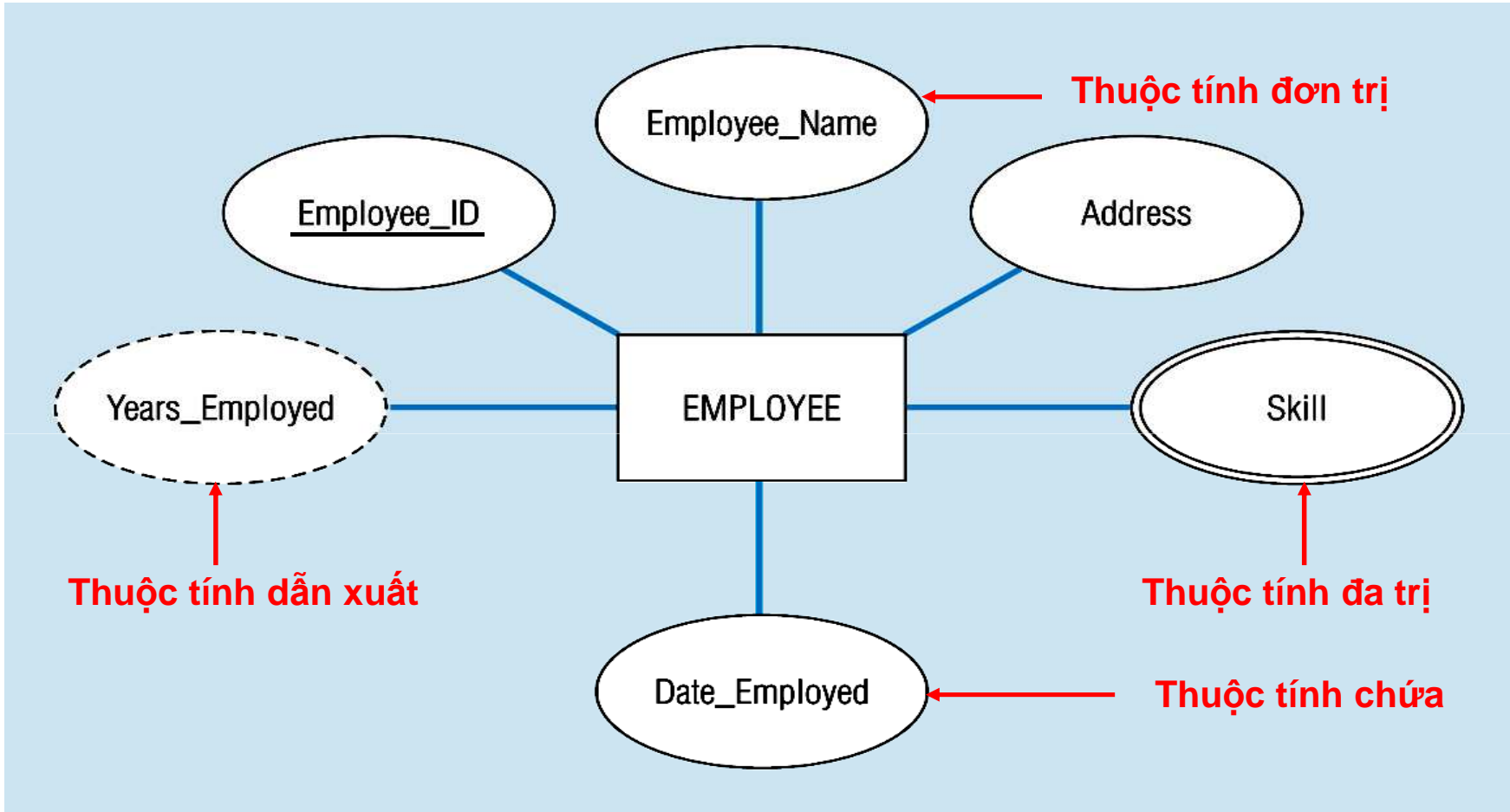
- ▶ **Thuộc tính đơn** (*simple attribute*) là thuộc tính không bị phân rã thành nhiều thuộc tính khác.
- ▶ **Thuộc tính phức hợp** (*composite attribute*) là thuộc tính bị phân rã thành nhiều thuộc tính khác.



Phân loại thuộc tính

- ▶ *Thuộc tính đơn trị* (*single-valued attribute*) là thuộc tính chỉ chứa một giá trị.
- ▶ *Thuộc tính đa trị* (*multivalued attribute*) là thuộc tính chứa nhiều giá trị khác nhau thuộc một miền trị, được biểu diễn bằng *hình bầu dục nét đôi*.

- ▶ *Thuộc tính chứa* (*stored attribute*) là thuộc tính mà giá trị của nó không được suy dẫn từ các thuộc tính khác.
- ▶ *Thuộc tính dẫn xuất* (*derived attribute*) là thuộc tính mà giá trị của nó được suy dẫn từ các thuộc tính khác, được biểu diễn bằng *hình bầu dục nét đứt*.



Mô hình liên kết thực thể[?]

❖ Mô hình liên kết thực thể (ER - *entity-relationship model*)

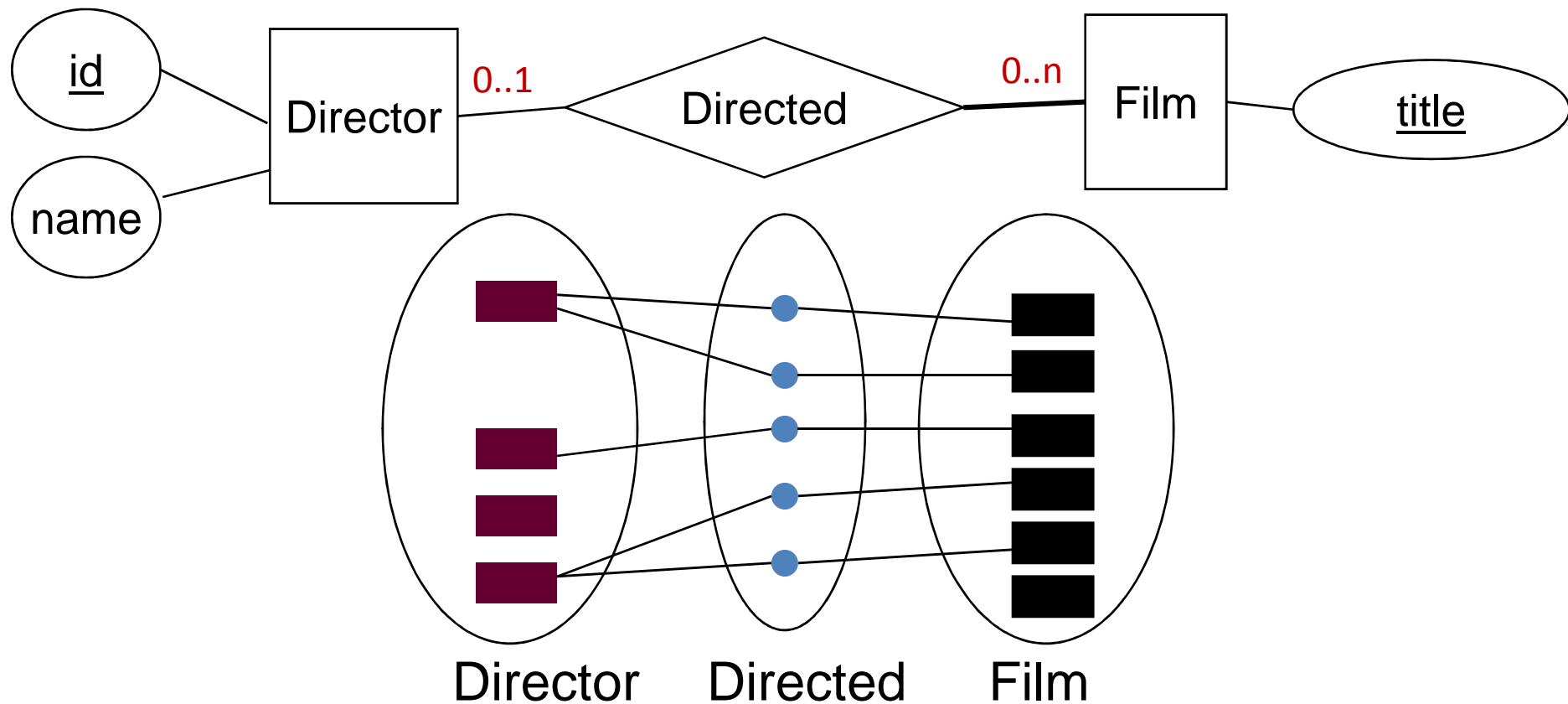
- ▶ Mô hình liên kết thực thể là cách tiếp cận chính để mô hình hóa dữ liệu ý niệm (*conceptual data modeling*).
- ▶ Mô hình ER là công cụ giao tiếp giữa người thiết kế CSDL và người sử dụng cuối cùng để xây dựng CSDL trong giai đoạn phân tích.
- ▶ Mô hình ER được dùng để xây dựng mô hình dữ liệu ý niệm (*conceptual data model*) nhằm biểu diễn cấu trúc và các ràng buộc của CSDL.

❖ Mô hình liên kết thực thể bao gồm

- ▶ Thực thể, thuộc tính, mối liên kết giữa các thực thể

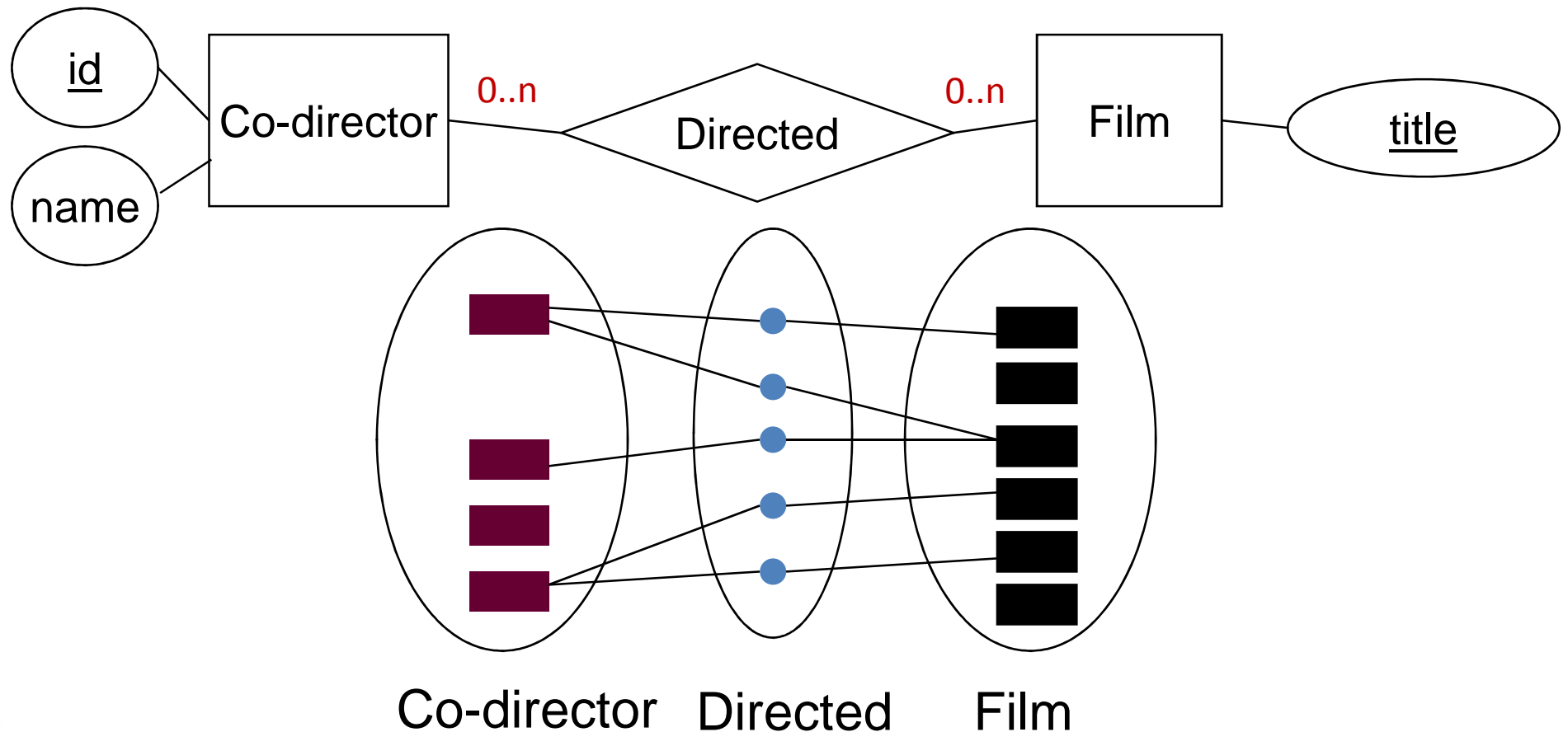
Liên kết một-nhiều

- ✓ Một bộ phim được đạo diễn bởi nhiều nhất một đạo diễn
- ✓ Một đạo diễn có thể hiện thực một số lượng phim bất kỳ



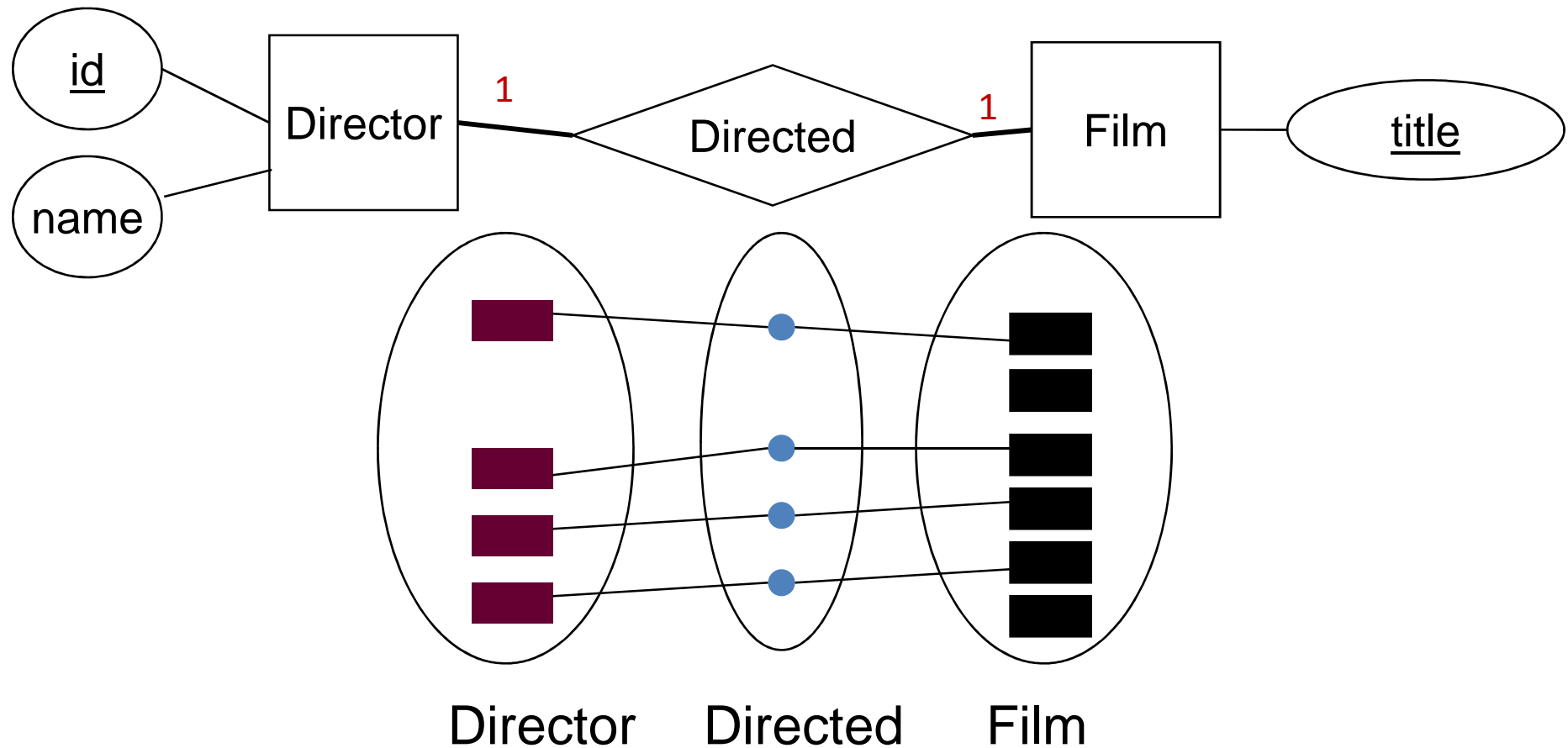
Liên kết nhiều-nhiều

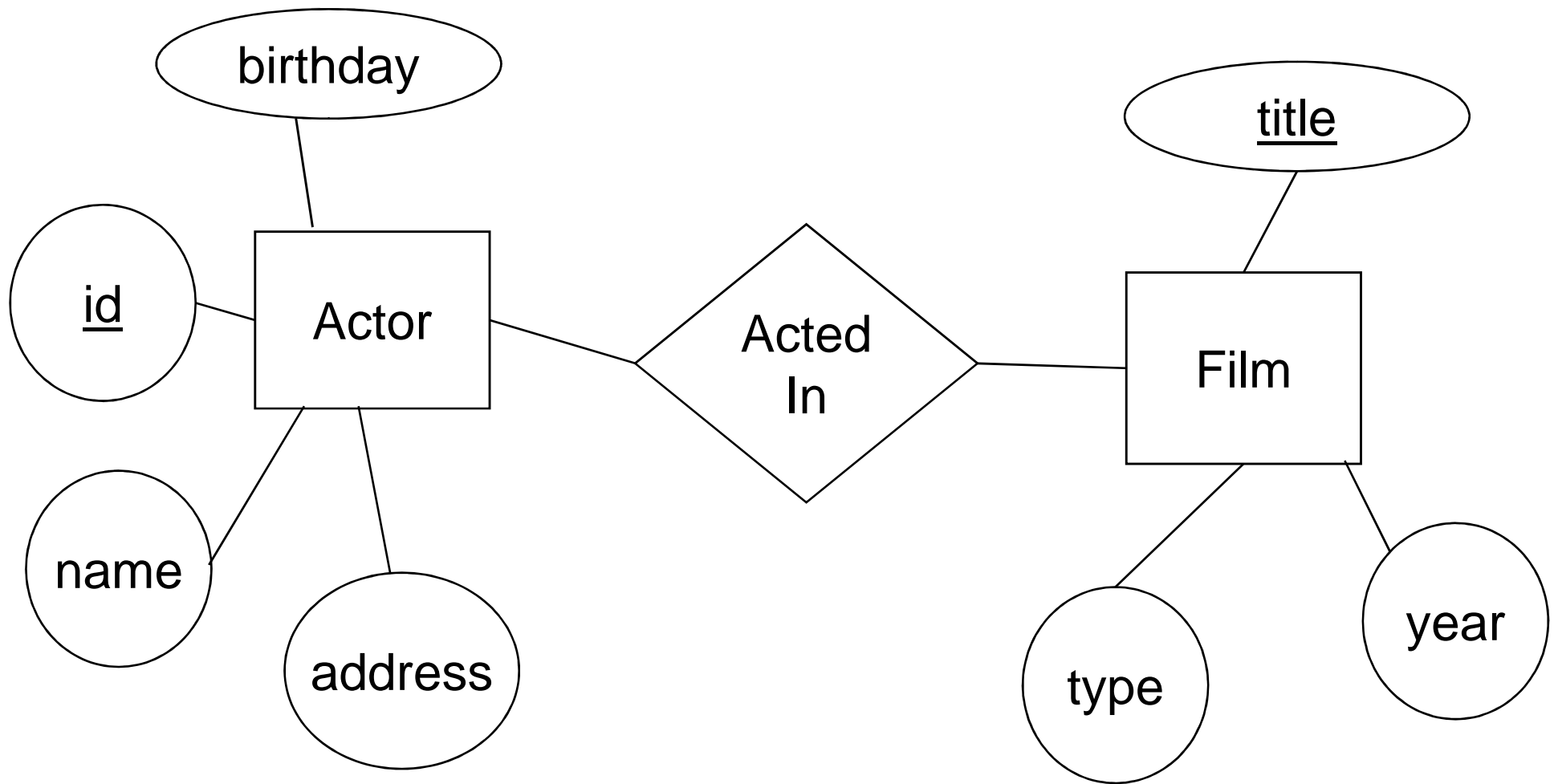
- ✓ Một bộ phim được chỉ đạo bởi một số phó đạo diễn
- ✓ Một phó đạo diễn có thể chỉ đạo nhiều bộ phim

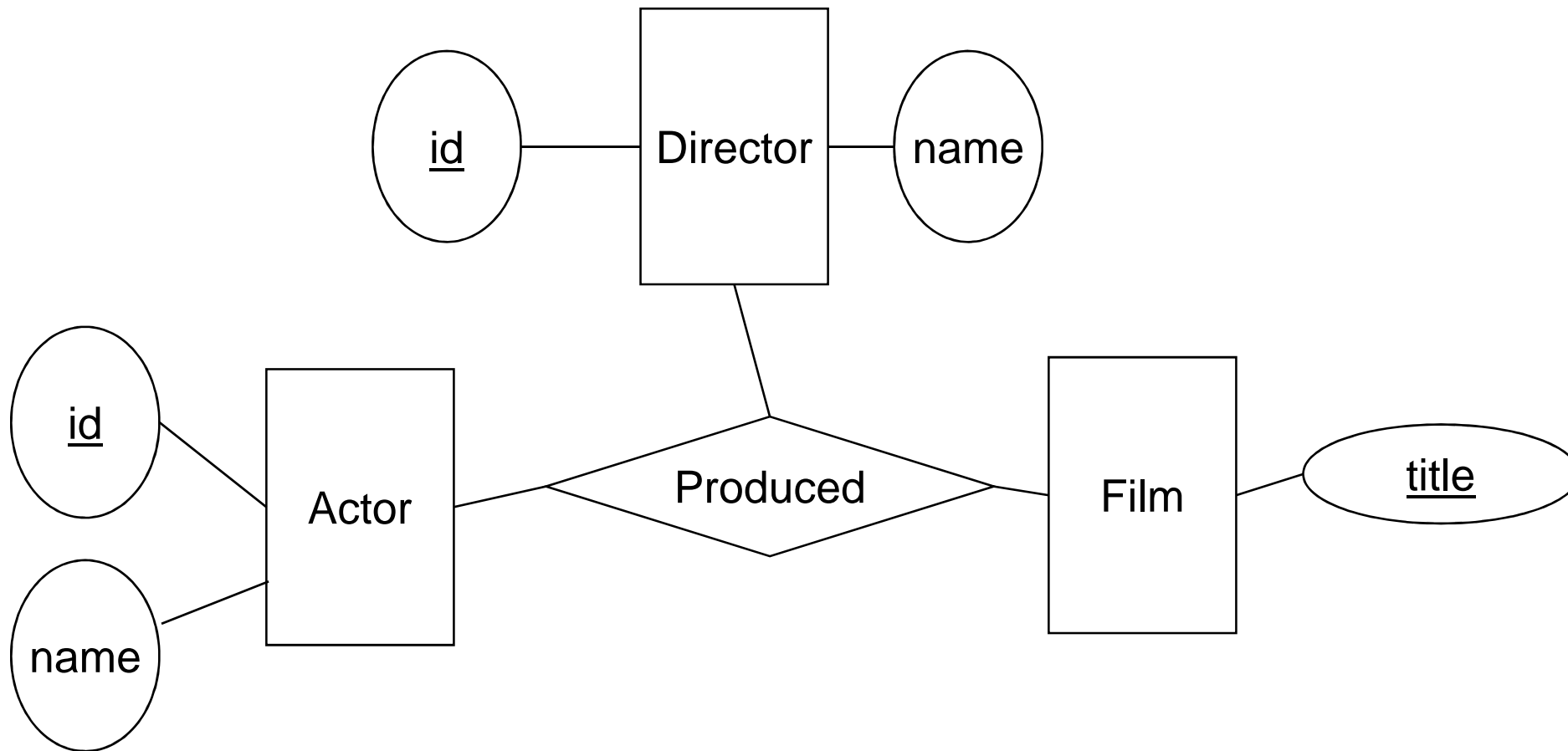


Liên kết một-một

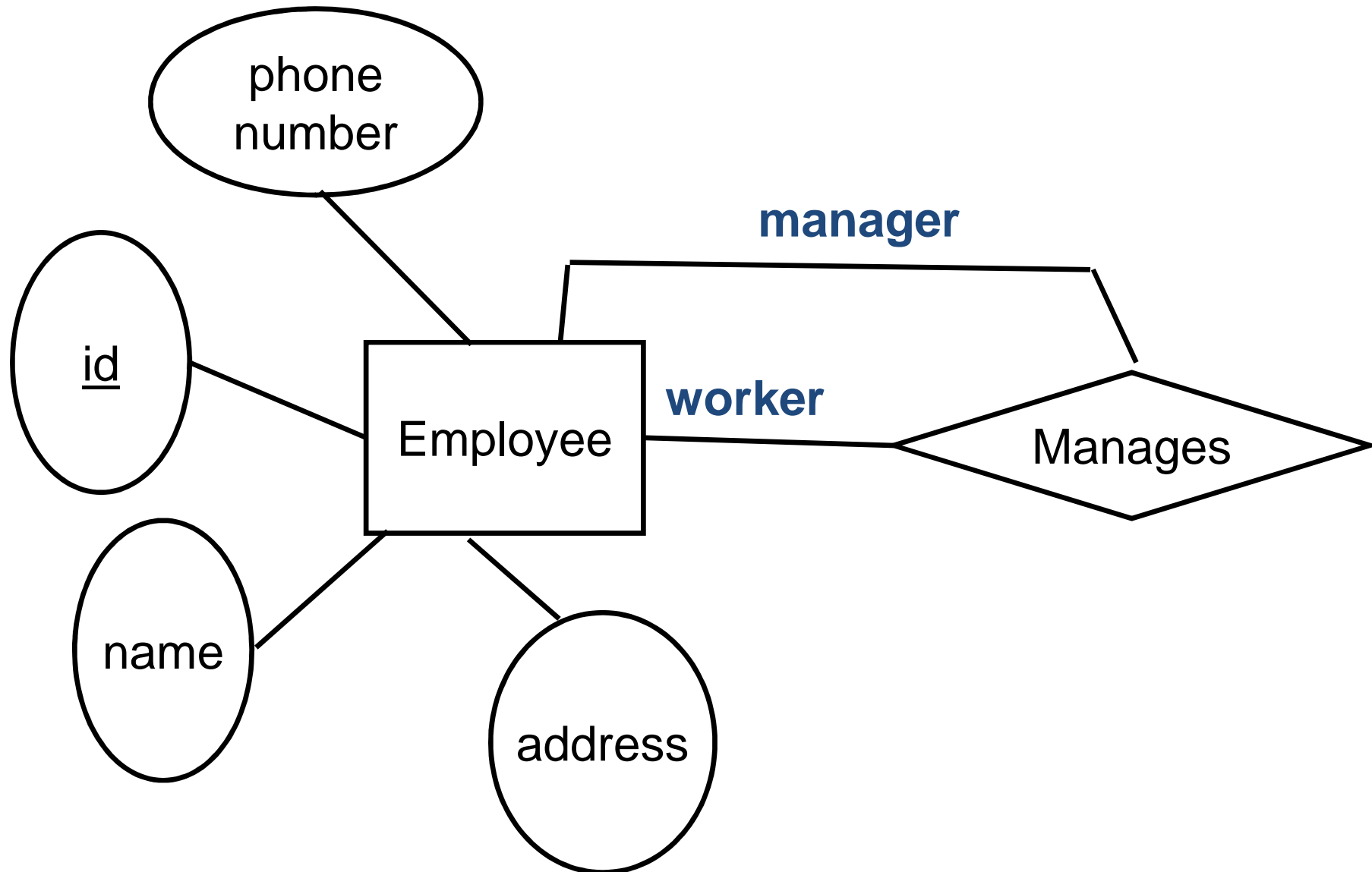
- ✓ Một bộ phim được chỉ đạo bởi nhiều nhất một đạo diễn
- ✓ Một đạo diễn có thể chỉ đạo nhiều nhất một bộ phim







Đặt tên cho quan hệ



Lược đồ cơ sở dữ liệu

❖ Lược đồ cơ sở dữ liệu (*database schema*)

- ▶ là một tập hợp các lược đồ quan hệ.
- ▶ Trong một lược đồ cơ sở dữ liệu, các tên lược đồ quan hệ là duy nhất.

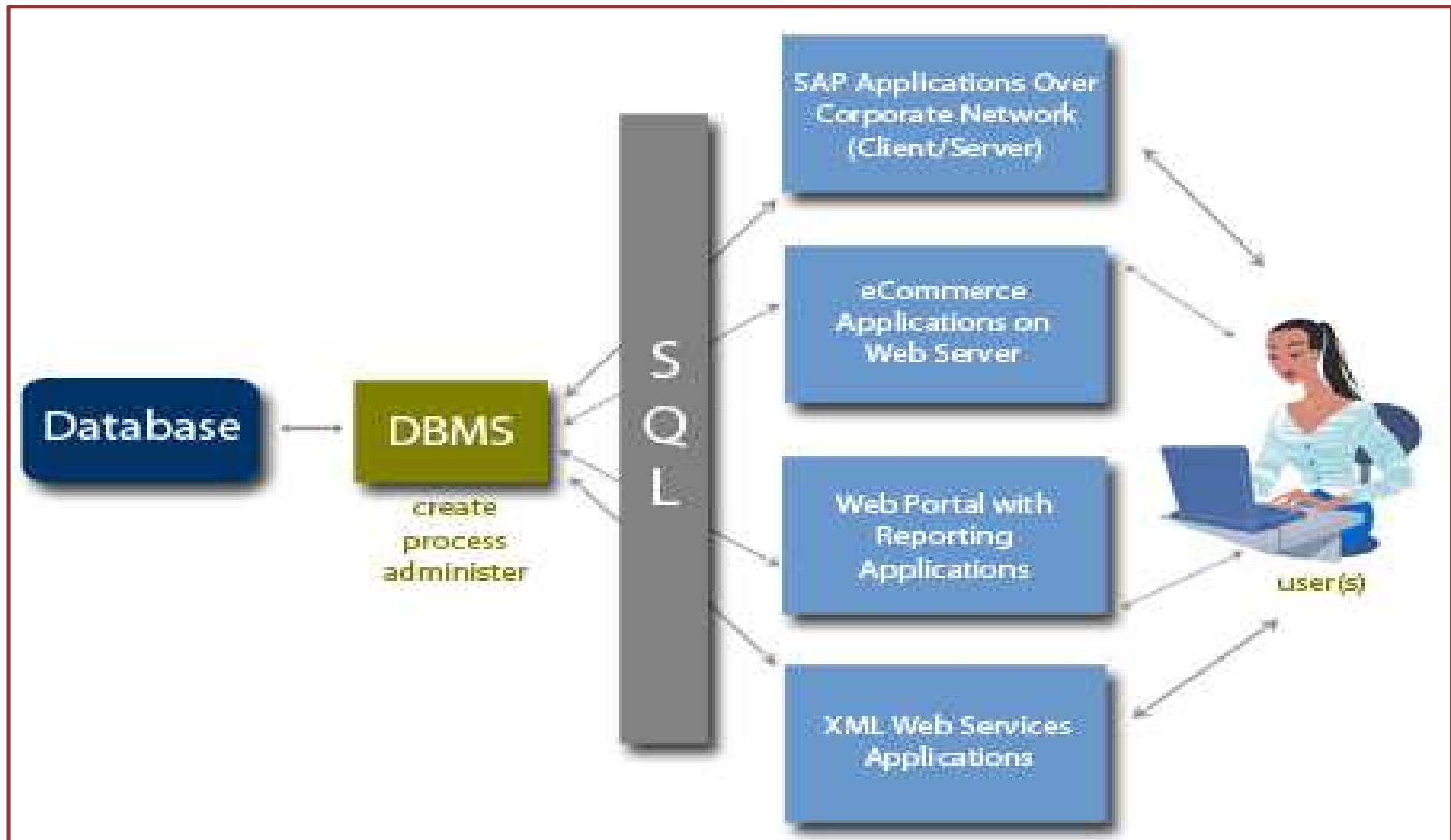
Emp (*Empnum*, *Name*, *Sal*, *Tax*, *Mgrnum*, *Deptnum*)

Dept (*Deptnum*, *Name*, *Area*, *Mgrnum*)

Supplier (*Snum*, *Name*, *City*)

Supply (*Snum*, *Pnum*, *Deptnum*, *Quan*)

SQL – Ngôn ngữ truy vấn có cấu trúc



Source: <http://systemsvie.net/infosystems/dbms/dbms.html>

Ngôn ngữ SQL

- ❖ *Ngôn ngữ truy vấn có cấu trúc* (SQL - *Structured Query Language*) là một ngôn ngữ chuẩn được dùng để tạo lập và truy vấn các cơ sở dữ liệu quan hệ.
 - ❖ SQL là một ngôn ngữ chuẩn cho các hệ quản trị CSDL quan hệ (RDBMS - *Relational DBMS*).
-
- ❖ Ngôn ngữ SQL là một *ngôn ngữ tựa tiếng Anh* (*English-like language*), sử dụng các từ như *select*, *insert*, *delete*, *update* trong tập lệnh.
 - ❖ Ngôn ngữ SQL là một *ngôn ngữ phi thủ tục* (*nonprocedural language*).
 - ▶ Chỉ ra các thông tin *gì* cần thiết (*what*).
 - ▶ *Không* cần phải chỉ ra cách thực hiện *như thế nào* (*how*) để có được các thông tin này.
 - ❖ SQL xử lý các tập hợp mẫu tin (bảng) hơn là mỗi lần một mẫu tin đơn lẻ.

Đặc điểm của ngôn ngữ SQL

- ❖ Nhiều loại người có thể sử dụng *SQL*: người quản trị CSDL (**DBA**), người lập trình ứng dụng, người quản lý, người sử dụng cuối cùng (*end user*).
- ❖ *SQL* cung cấp nhiều lệnh cho nhiều công việc khác nhau:
 - ▶ Truy vấn dữ liệu.
 - ▶ Thêm vào, cập nhật và xóa bỏ các hàng của bảng.
 - ▶ Tạo lập, thay đổi và xóa bỏ các đối tượng CSDL.
 - ▶ Điều khiển truy xuất cơ sở dữ liệu và các đối tượng CSDL.
 - ▶ Bảo đảm tính nhất quán của CSDL.

Một số kiểu dữ liệu

❖ Kiểu chuỗi

- ▶ *CHAR*(n) – kiểu chuỗi có chiều dài cố định gồm n ký tự (chiều dài tối đa 2000 bytes).
- ▶ *VARCHAR2*(n) – kiểu chuỗi có chiều dài thay đổi gồm n ký tự (chiều dài tối đa 4000 bytes).
- ▶ *LONG* – kiểu chuỗi có chiều dài thay đổi, tối đa 4GB (chỉ có một cột trong một bảng).

❖ Kiểu số

- ▶ *NUMBER*(p, q) – kiểu số có p ký số và q số lẻ.
- ▶ *INTEGER*(p) – kiểu số nguyên có p ký số.

❖ Kiểu ngày giờ

- ▶ *DATE* – kiểu ngày giờ có chiều dài cố định theo dạng **dd-mm-yy**.

Ngôn ngữ SQL

❖ Ngôn ngữ định nghĩa dữ liệu

- ▶ DDL - *Data Definition Language*
- ▶ Các lệnh dùng để định nghĩa CSDL: tạo lập (*create*), thay đổi (*alter*) và hủy bỏ (*drop*) các đối tượng dữ liệu, thiết lập các ràng buộc.

❖ Ngôn ngữ thao tác dữ liệu

- ▶ DML - *Data Manipulation Language*
- ▶ Các lệnh dùng để bảo trì và truy vấn CSDL: thêm (*insert*), sửa (*update*), xóa (*delete*) dữ liệu của bảng, truy vấn (*select*).

❖ Ngôn ngữ điều khiển dữ liệu

- ▶ DCL - *Data Control Language*
- ▶ Các lệnh dùng để điều khiển CSDL: quản trị các quyền (*grant*, *revoke*).

Ngôn ngữ định nghĩa dữ liệu

- ❖ Tạo cấu trúc của một bảng.

```
CREATE TABLE <table name>  
  
(<column_definition>, ...  
  
[<table_constraint_definition>]);
```

- ❖ Hủy bỏ một bảng.

```
DROP TABLE <table name>  
  
[CASCADE CONSTRAINTS];
```

```
DROP TABLE CUSTOMER;
```

```
CREATE TABLE CUSTOMER  
(CUST_ID NUMBER(11,0) NOT NULL,  
NAME VARCHAR(25) NOT NULL,  
ADDRESS VARCHAR(30),  
CITY VARCHAR(20),  
CONSTRAINT PK_CUSTOMER  
PRIMARY KEY (CUST_ID));
```

Ngôn ngữ thao tác dữ liệu

❖ Mệnh đề của lệnh *SELECT*

- ▶ *SELECT*: liệt kê các cột (các biểu thức) của kết quả.
- ▶ *FROM*: các bảng hoặc các khung nhìn chứa dữ liệu cần thiết cho truy vấn.
- ▶ *WHERE*: điều kiện xử lý các hàng để tạo ra kết quả.
- ▶ *GROUP BY*: gom nhóm các hàng.
- ▶ *HAVING*: điều kiện xử lý các nhóm.
- ▶ *ORDER BY*: sắp thứ tự kết quả.

```
SELECT CUST_ID, NAME, ADDRESS  
FROM CUSTOMER  
WHERE CITY = 'HCM';
```

Liệt kê mã, tên và địa chỉ của các khách hàng thuộc thành phố 'HCM'

Ví dụ về lệnh SELECT

```
SELECT *  
FROM Order;
```

```
SELECT OrderID, OrderDate, CustomerID  
FROM Order;
```

```
SELECT DISTINCT OrderDate AS "Date of Order"  
FROM Order;
```

```
SELECT Order_ID AS Identifier, OrderDate AS Date  
FROM Order;
```

Ví dụ về lệnh SELECT

```
SELECT ProductID, StandardPrice
FROM Product
WHERE StandardPrice < 275;
```

```
SELECT Cust.CustomerName AS Name, CustomerAddress
FROM Customer Cust
WHERE CustomerName = 'Home Furnishings';
```

```
SELECT ProductID, StandardPrice
FROM Product
WHERE StandardPrice BETWEEN 100 AND 200;
```

```
SELECT CustomerName, District, City
FROM Customer
WHERE City IN ('Long An', 'Tien Giang', 'My Tho');
```

Ví dụ về lệnh SELECT

```
SELECT ProductDescription, ProductFinish, StandardPrice
FROM Product
WHERE (ProductDescription LIKE '%Desk'
      OR ProductDescription LIKE 'S%')
      AND StandardPrice > 300;
```

```
SELECT ProductID, ProductFinish, StandardPrice
FROM Product
WHERE ProductDescription IS NULL;
```

```
SELECT COUNT(*)
FROM Order
WHERE OrderID = 1004;
```

```
SELECT District, COUNT(District)
FROM Customer
WHERE District IN ('Q1', 'Q2', 'Q3', 'Q9')
GROUP BY District
HAVING COUNT(District) >= 10
ORDER BY District DESC;
```

Thêm dữ liệu vào một bảng

- Cú pháp thêm một hàng:

```
INSERT INTO <table name>  
[(<list of columns>)]  
VALUES (<list of expressions>);
```

```
INSERT INTO Customer  
VALUES (001, 'Contemporary Casuals',  
'1355 S. Himes Blvd.', 'Gainesville', 'FL',  
32601);
```

```
INSERT INTO Product (ProductID,  
ProductDescription, ProductFinish,  
StandardPrice, ProductOnHand)  
VALUES (1, 'End Table', 'Cherry', 175, 8);
```

- Cú pháp thêm nhiều hàng:

```
INSERT INTO <table name>  
[(<list of columns>)]  
SELECT statement;
```

```
INSERT INTO CA_Customer  
SELECT *  
FROM Customer  
WHERE State = 'CA';
```

Cập nhật dữ liệu

Sử dụng lệnh UPDATE:

UPDATE <table name> [<alias>]

SET <column1> = {<expression>, <subquery>}

[, <column2> = {<expression>, <subquery>} ...]

[**WHERE** <row conditions>];

```
UPDATE Product
```

```
SET UnitPrice = 775
```

```
WHERE ProductID = 7;
```

Xóa các hàng trong một bảng

Sử dụng lệnh DELETE:

```
DELETE [FROM] <table name>
```

```
[WHERE <row conditions>];
```

Xóa một số hàng của bảng Customer

```
DELETE FROM Customer
```

```
WHERE State = 'HI';
```

Xóa tất cả các hàng của bảng Customer

```
DELETE FROM Customer;
```


Ngôn ngữ điều khiển dữ liệu

Cho phép người sử dụng **Tien** và **Truc** được phép xem và thêm dữ liệu vào bảng **Customer**

```
GRANT SELECT, INSERT  
ON CUSTOMER  
TO TIEN, TRUC;
```

Xóa quyền thêm dữ liệu vào bảng **Customer** của **Truc**

```
REVOKE INSERT  
ON CUSTOMER  
FROM TRUC;
```

Ràng buộc toàn vẹn

- ❖ Ràng buộc toàn vẹn (*integrity constraint*)
 - ▶ *Ràng buộc toàn vẹn* là một qui tắc mà tất cả các dữ liệu trong CSDL phải thỏa mãn qui tắc này.
- ❖ Ràng buộc miền trị (*domain constraint*)
 - ▶ Các giá trị cho phép của một thuộc tính.
- ❖ Toàn vẹn thực thể (*entity integrity*)
 - ▶ Thuộc tính khóa chính không có giá trị rỗng (*null value*).

Ví dụ về ràng buộc miền trị

<i>Attribute</i>	<i>Domain Name</i>	<i>Description</i>	<i>Domain</i>
Customer_ID	Customer_IDs	Set of all possible customer IDs	character: size 5
Customer_Name	Customer_Names	Set of all possible customer names	character: size 25
Customer_Address	Customer_Addresses	Set of all possible customer addresses	character: size 30
City	Cities	Set of all possible cities	character: size 20
State	States	Set of all possible states	character: size 2
Postal_Code	Postal_Codes	Set of all possible postal zip codes	character: size 10
Order_ID	Order_IDs	Set of all possible order IDs	character: size 5
Order_Date	Order_Dates	Set of all possible order dates	date format mm/dd/yy
Product_ID	Product_IDs	Set of all possible product IDs	character: size 5
Product_Description	Product_Descriptions	Set of all possible product descriptions	character size 25
Product_Finish	Product_Finishes	Set of all possible product finishes	character: size 15
Standard_Price	Unit_Prices	Set of all possible unit prices	monetary: 6 digits
Product_Line_ID	Product_Line_IDs	Set of all possible product line IDs	integer: 3 digits
Ordered_Quantity	Quantities	Set of all possible ordered quantities	integer: 3 digits

Định nghĩa miền trị cho các thuộc tính

Ràng buộc toàn vẹn tham chiếu

❖ *Referential integrity constraint*

- ▶ là một qui tắc mà tất cả các giá trị của khóa ngoại (nếu khác *null*) của một bảng phải có trong các giá trị của khóa chính trong bảng liên quan.

▶ Qui tắc xóa các hàng dữ liệu

- **Hạn chế** (*restrict*): không cho phép xóa các hàng bên phía cha (*parent*) nếu tồn tại các hàng liên quan bên phía phụ thuộc (*dependent*).
- **Tầng** (*cascade*): tự động xóa các hàng bên phía phụ thuộc tương ứng với các hàng bên phía cha.
- **Gán null** (*set-to-null*): gán *null* cho khóa ngoại của các hàng bên phía phụ thuộc tương ứng với các hàng bên phía cha.

Ví dụ về ràng buộc toàn vẹn tham chiếu

CUSTOMER

<u>Customer_ID</u>	Customer_Name	Address	City	State	Zip
--------------------	---------------	---------	------	-------	-----

ORDER

<u>Order_ID</u>	Order_Date	<u>Customer_ID</u>
-----------------	------------	--------------------

ORDER LINE

<u>Order_ID</u>	<u>Product_ID</u>	Quantity
-----------------	-------------------	----------

PRODUCT

<u>Product_ID</u>	Product_Description	Product_Finish	Standard_Price	On_Hand
-------------------	---------------------	----------------	----------------	---------

Ràng buộc toàn vẹn tham chiếu được biểu diễn bằng những mũi tên từ phía phụ thuộc đến phía cha tương ứng

Chuẩn hóa dữ liệu

❖ *data normalization*

- ▶ là một quá trình thuận nghịch từng bước để thay thế tập hợp các quan hệ cho trước thành các quan hệ có *cấu trúc đơn giản hơn* và *chuẩn hơn*.
- ▶ nhằm để cải tiến một thiết kế CSDL *thỏa mãn các ràng buộc toàn vẹn* và *tránh dữ liệu bị lặp lại không cần thiết*.

❖ Mục đích của chuẩn hóa dữ liệu

- ▶ Loại bỏ các bất thường (*anomaly*) của một quan hệ để có được các quan hệ *có cấu trúc tốt hơn, nhỏ hơn*.
- ▶ *Quan hệ có cấu trúc tốt (well-structured relation)*

Có sự dư thừa dữ liệu là tối thiểu.

Cho phép người sử dụng thêm vào, cập nhật và xóa bỏ dữ liệu mà không gây ra sự mâu thuẫn dữ liệu.

Phụ thuộc hàm

❖ *Functional Dependency*

- ▶ Cho R là một quan hệ, X và Y là hai tập thuộc tính của R .
- ▶ Chúng ta nói “ X xác định hàm Y ” hoặc “ Y phụ thuộc hàm vào X ”, ký hiệu là $X \rightarrow Y$ và được gọi là *phụ thuộc hàm* nếu với mỗi giá trị của X trong R chỉ tương ứng với một giá trị của Y .
- ▶ Khóa của một quan hệ xác định hàm các thuộc tính không khóa của quan hệ này.

Các loại phụ thuộc hàm

❖ Phụ thuộc hàm riêng phần

▶ *partial functional dependency*

▶ $X \rightarrow A$ được gọi là *phụ thuộc hàm riêng phần* nếu tồn tại $Y \subset X$ để cho $Y \rightarrow A$.

❖ Phụ thuộc hàm đầy đủ

▶ *full functional dependency*

▶ $X \rightarrow A$ được gọi là *phụ thuộc hàm đầy đủ* nếu không tồn tại $Y \subset X$ để cho $Y \rightarrow A$.

❖ Phụ thuộc bắc cầu

▶ *transitive dependency*

▶ $X \rightarrow A$ được gọi là *phụ thuộc bắc cầu* nếu tồn tại Y để cho $X \rightarrow Y, Y \rightarrow A, Y \not\rightarrow X$ và $A \notin XY$.

Ví dụ về phụ thuộc hàm

R

Mãsv	Họ tên	Mã lớp	Tên lớp	Môn học	Điểm
S1	Tiến	L1	MT01	M1	9
S1	Tiến	L1	MT01	M2	7
S1	Tiến	L1	MT01	M3	8
S2	Trúc	L1	MT01	M1	9
S2	Trúc	L1	MT01	M2	8
S3	Hiền	L2	MT02	M1	5

Khóa của R: {Mãsv, Môn học}

Các phụ thuộc hàm:

Mãsv \rightarrow {Họ tên, Mã lớp}

Mã lớp \rightarrow Tên lớp

{Mãsv, Môn học} \rightarrow Điểm

{Mãsv, Môn học} \rightarrow Họ tên là phụ thuộc hàm riêng phần

{Mãsv, Môn học} \rightarrow Điểm là phụ thuộc hàm đầy đủ

Mãsv \rightarrow Tên lớp là phụ thuộc bắc cầu

Dạng chuẩn 1²

❖ Định nghĩa

Quan hệ R ở *dạng chuẩn 1* (1NF - *First Normal Form*) nếu mọi thuộc tính của R đều chứa các *giá trị nguyên tố* (*atomic value*), giá trị này không là *một danh sách các giá trị* hoặc *các giá trị phức hợp* (*composite value*).

❖ Các thuộc tính của quan hệ R

- ▶ Không là thuộc tính đa trị (*multivalued attribute*).
- ▶ Không là thuộc tính phức hợp (*composite attribute*).

R

Mãsv	Họtên	Mãlớp	Tênlớp	Điểmthi	
S1	Tiến	L1	MT01	M1	9
S1	Tiến	L1	MT01	M2	7
S1	Tiến	L1	MT01	M3	8
S2	Trúc	L1	MT01	M1	9
S2	Trúc	L1	MT01	M2	8
S3	Hiền	L2	MT02	M1	5

Quan hệ R không ở dạng chuẩn 1 vì thuộc tính *Điểmthi* là thuộc tính phức hợp.

R

Mãsv	Họ tên	Mã lớp	Tên lớp	Môn học	Điểm
S1	Tiến	L1	MT01	M1	9
S1	Tiến	L1	MT01	M2	7
S1	Tiến	L1	MT01	M3	8
S2	Trúc	L1	MT01	M1	9
S2	Trúc	L1	MT01	M2	8
S3	Hiền	L2	MT02	M1	5

Quan hệ R ở **1NF** vì các thuộc tính của R không là **thuộc tính đa trị**, không là **thuộc tính phức hợp**.

Các bất thường của quan hệ ở 1NF

▶ Thêm vào

Không thể thêm thông tin của sinh viên mới có mã là *S4*, tên là *Thành*, thuộc lớp có mã là *L1* nếu sinh viên này chưa đăng ký học môn học nào cả.

▶ Cập nhật

Sửa tên của sinh viên có tên là *Tiến* với tên mới là *Thành* sẽ phải sửa tất cả các hàng của sinh viên này.

▶ Xóa bỏ

Xóa thông tin sinh viên *S3* đăng ký môn học *M1* sẽ làm mất thông tin của sinh viên này.

▶ Nguyên nhân

Tồn tại thuộc tính không khóa phụ thuộc hàm riêng phần vào khóa.

Dạng chuẩn 2

- Quan hệ R ở *dạng chuẩn 2* (2NF - *Second Normal Form*) nếu R ở dạng chuẩn 1 và mọi thuộc tính không khóa đều *phụ thuộc hàm đầy đủ* vào mọi khóa của R .

R

Mãsv	Họ tên	Mã lớp	Tên lớp	Môn học	Điểm
S1	Tiến	L1	MT01	M1	9
S1	Tiến	L1	MT01	M2	7
S1	Tiến	L1	MT01	M3	8
S2	Trúc	L1	MT01	M1	9
S2	Trúc	L1	MT01	M2	8
S3	Hiền	L2	MT02	M1	5

Khóa của R

{Mãsv, Môn học}

Các phụ thuộc hàm:

Mãsv \rightarrow {Họ tên, Mã lớp}

Mã lớp \rightarrow Tên lớp

{Mãsv, Môn học} \rightarrow Điểm

Lược đồ quan hệ R không ở 2NF vì thuộc tính không khóa *Họ tên* phụ thuộc hàm riêng phần vào khóa {Mãsv, Môn học}.

R_1

Mãsv	Họ tên	Mã lớp	Tên lớp
S1	Tiến	L1	MT01
S2	Trúc	L1	MT01
S3	Hiền	L2	MT02

Khóa của R_1 : Mãsv

 R_2

Mãsv	Môn học	Điểm
S1	M1	9
S1	M2	7
S1	M3	8
S2	M1	9
S2	M2	8
S3	M1	5

Khóa của R_2 : {Mãsv, Môn học}

Lược đồ quan hệ R_1 và R_2 đều ở 2NF vì các thuộc tính không khóa đều phụ thuộc hàm đầy đủ vào khóa.

Các bất thường của quan hệ ở 2NF

- ▶ Thêm vào

Không thể thêm thông tin của lớp $L3$ có tên là $MT03$ nếu chưa có sinh viên nào học lớp này.

- ▶ Cập nhật

Sửa tên của lớp có mã $L1$ với tên mới là MT_1 sẽ phải sửa tất cả các hàng của lớp này.

- ▶ Xóa bỏ

Xóa thông tin của sinh viên có mã $S3$ sẽ làm mất thông tin của lớp $L2$.

- ▶ Nguyên nhân

Tồn tại thuộc tính không khóa **phụ thuộc bắc cầu** vào khóa.

Dạng chuẩn 3

- Quan hệ R ở *dạng chuẩn 3* (3NF- *Third Normal Form*) nếu R ở dạng chuẩn 1 và mọi thuộc tính không khóa đều **không phụ thuộc bắc cầu** vào một khóa của R .

R_1

Mãsv	Họ tên	Mã lớp	Tên lớp
S1	Tiến	L1	MT01
S2	Trúc	L1	MT01
S3	Hiền	L2	MT02

Lược đồ quan hệ R_1 không ở 3NF vì thuộc tính không khóa *Tên lớp* phụ thuộc bắc cầu vào khóa *Mãsv*.

$Mãsv \rightarrow Mã lớp$

$Mã lớp \rightarrow Tên lớp$

$Mã lớp \not\rightarrow Mãsv$

$Tên lớp \notin \{Mãsv, Mã lớp\}$

R_{11}

Mã lớp	Tên lớp
L1	MT01
L2	MT02

Khóa của R_{11} : Mã lớp

R_{12}

Mã sv	Họ tên	Mã lớp
S1	Tiến	L1
S2	Trúc	L1
S3	Hiền	L2

Khóa của R_{12} : Mã sv

Lược đồ quan hệ R_{11} và R_{12} đều ở **3NF** vì các thuộc tính không khóa đều không phụ thuộc bắc cầu vào khóa.

Sự phát triển các hệ DB

- ❖ Hệ thống tập tin (*flat file*): 1960 - 1980
- ❖ Hệ CSDL phân cấp (*hierarchical*): 1970 - 1990
- ❖ Hệ CSDL mạng (*network*): 1970 - 1990
- ❖ Hệ CSDL quan hệ (*relational*): 1980 - nay
- ❖ Hệ CSDL hướng đối tượng (*object-oriented*): 1990 - nay
- ❖ Hệ CSDL đối tượng - quan hệ (*object-relational*): 1990 - nay
- ❖ Kho dữ liệu (*data warehouse*): 1980 - nay
- ❖ *Web-enabled*: 1990 - nay