

Các mô hình dữ liệu

Nguyễn Hồng Phương

phuongnh@soict.hust.edu.vn

<http://users.soict.hust.edu.vn/phuongnh>

<https://www.facebook.com/phuongnhbk>

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



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



Tổng quan về mô hình dữ liệu

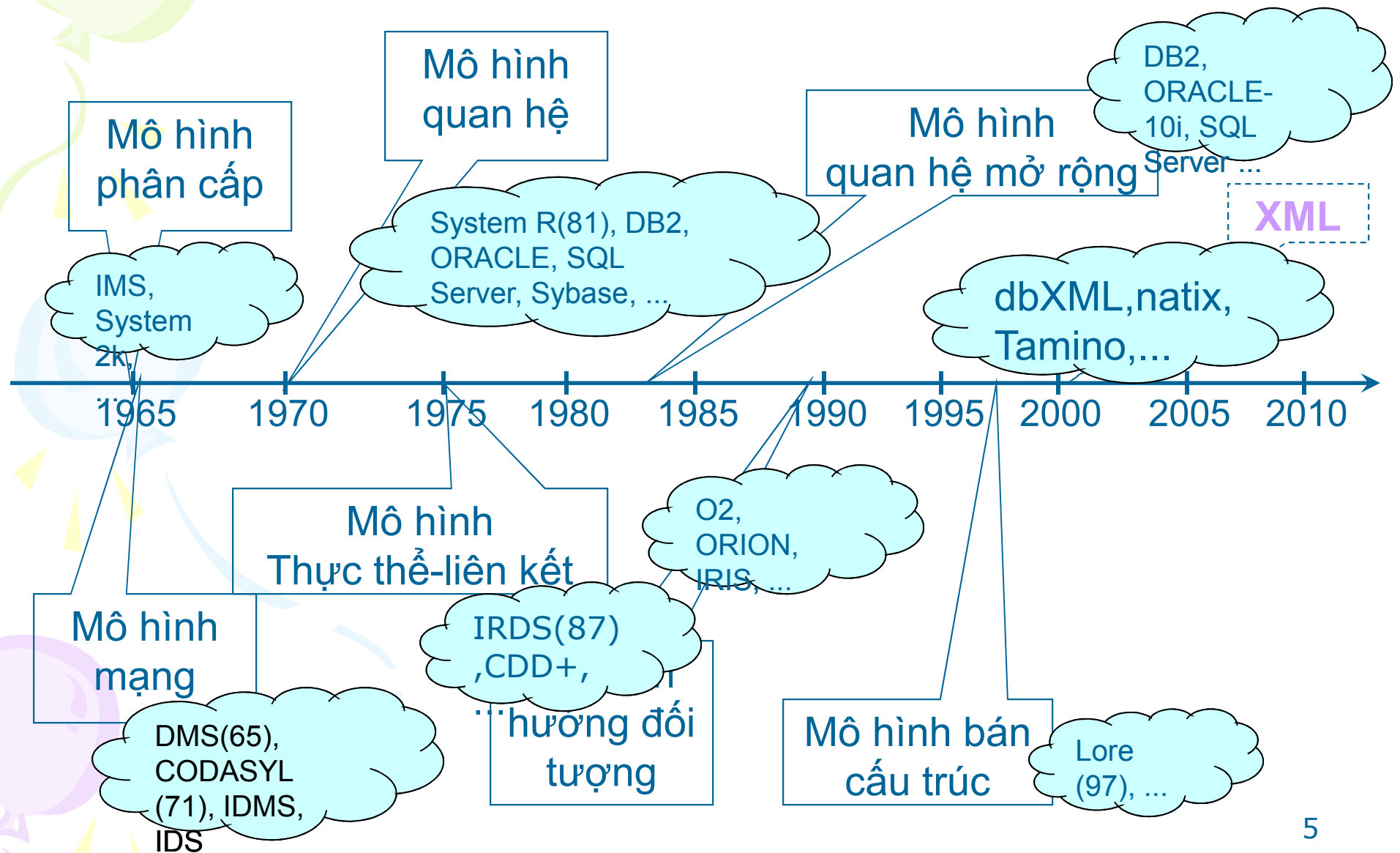
- Mô hình dữ liệu [Codd, 1980] gồm:
 - Một tập hợp các cấu trúc của dữ liệu
 - Một tập hợp các phép toán để thao tác với các dữ liệu
 - Một tập hợp các ràng buộc về dữ liệu
- Mô hình dữ liệu là một tập hợp các khái niệm dùng để mô tả:
 - Dữ liệu
 - Ngữ nghĩa của dữ liệu
 - Các mối quan hệ trong dữ liệu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



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

Vài nét về lịch sử





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



Đặt vấn đề

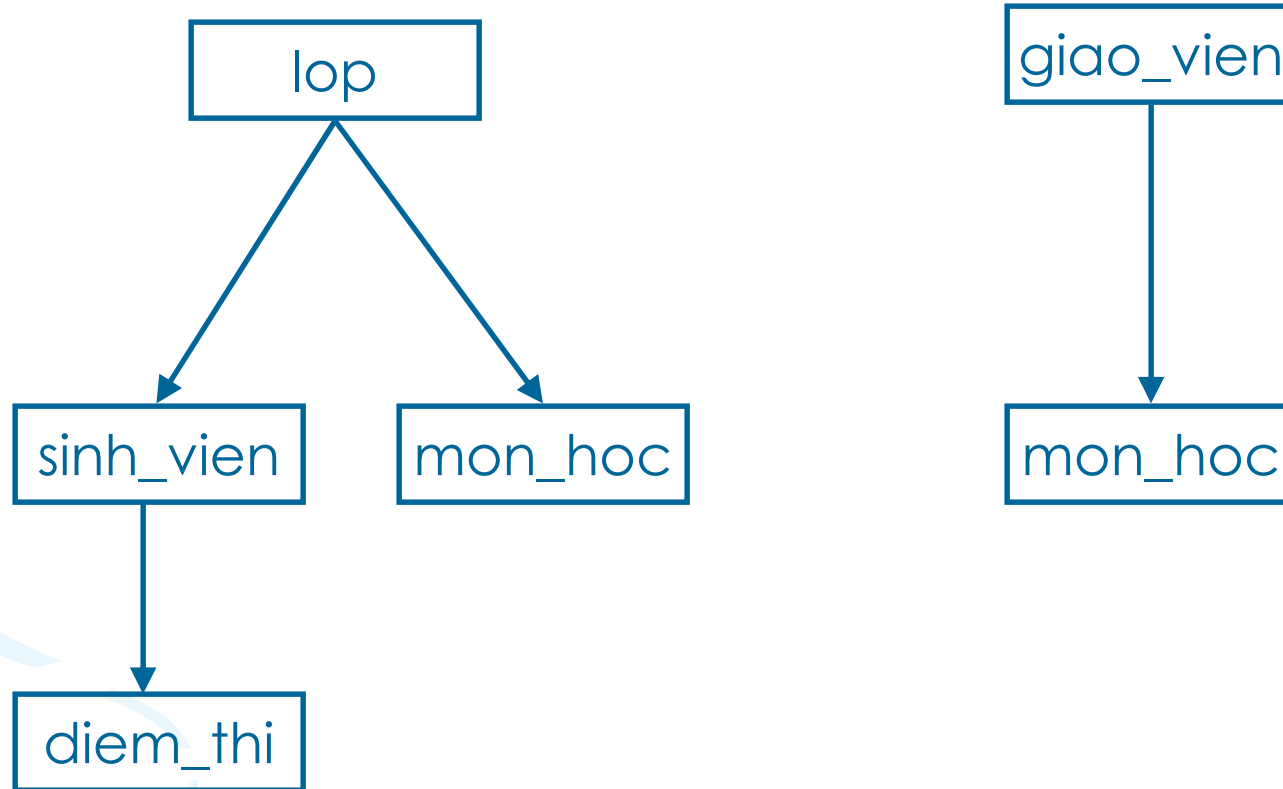
- Đặc điểm của các mô hình dữ liệu?
- Sự khác nhau giữa các mô hình dữ liệu?
- Các mô hình dữ liệu phổ biến ngày nay



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
 - Móc nối
 - Các phép toán: GET, GET UNIQUE, GET NEXT, GET NEXT WITHIN PARENT,...

Mô hình dữ liệu phân cấp – Ví dụ



Mô hình dữ liệu phân cấp

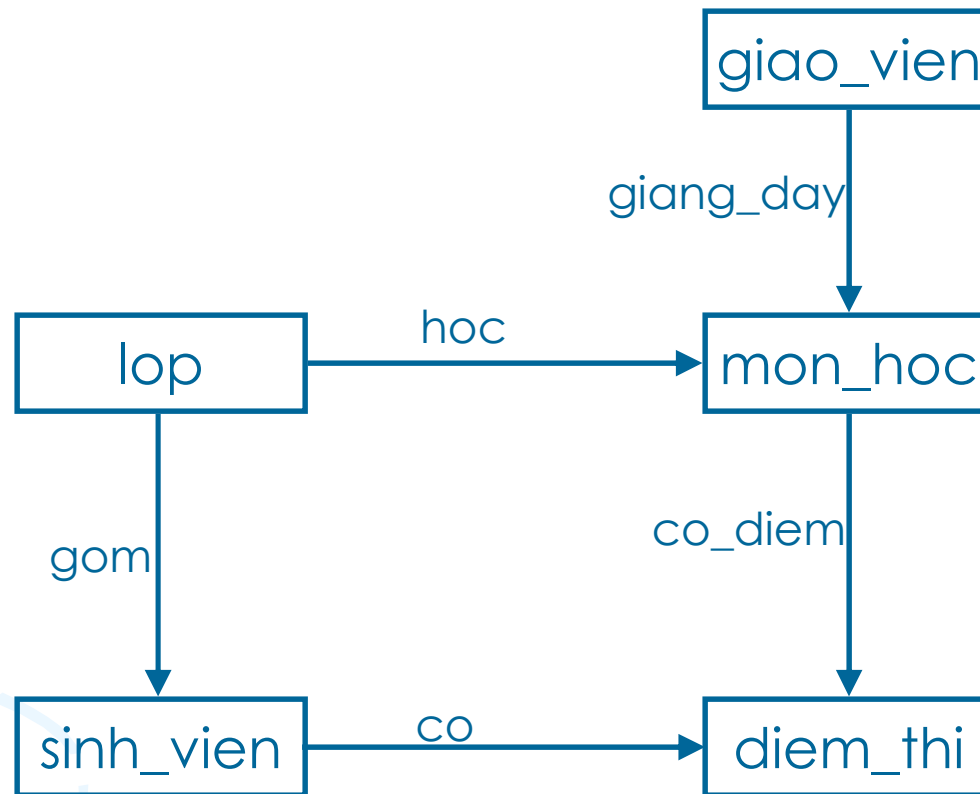
- Ưu điểm
 - Dễ xây dựng và thao tác
 - Tương thích với các lĩnh vực tổ chức phân cấp
 - Ngôn ngữ thao tác đơn giản: duyệt cây.
- Nhược điểm:
 - Sự lặp lại của các kiểu bản ghi → dữ liệu dư thừa và không nhất quán.
 - Giải pháp: bản ghi ảo
 - Hạn chế trong biểu diễn ngữ nghĩa của các mối nối giữa các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



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)
 - Móc nối
 - Tên của móc nối
 - Chủ (owner) – thành viên (member): theo hướng của móc nối
 - Kiểu móc nối: 1-1, 1-n, đệ quy
 - Các phép toán
 - Duyệt: FIND, FIND member, FIND owner, FIND NEXT
 - Thủ tục: GET

Mô hình dữ liệu mạng – Ví dụ





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



Mô hình dữ liệu quan hệ

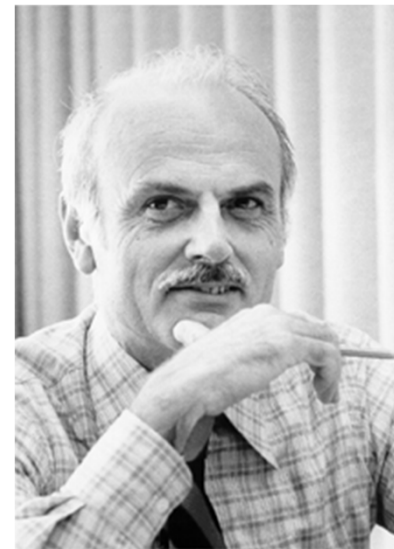
- Sự ra đời: vào năm 1970[Codd, 1970]
- Dữ liệu được biểu diễn dưới dạng bảng
- Là mô hình dữ liệu khái niệm phổ biến cho đến tận thời điểm hiện tại
- Dựa trên lý thuyết toán học, đồng thời cũng gần với cấu trúc tệp và cấu trúc dữ liệu nên có hai loại thuật ngữ liên quan:
 - Thuật ngữ toán học: quan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

Edgar Frank "Ted" Codd

- (19/08/1923 – 18/04/2003)
- Nhà khoa học máy tính người Anh, làm việc cho IBM.
- Cha đẻ của mô hình dữ liệu quan hệ
- Nhận giải thưởng Turing, năm 1981

<https://history.computer.org/pioneers/codd.html>

https://en.wikipedia.org/wiki/Edgar_F._Codd





Codd's 12 Rules

- Rule zero
- Rule 1: Information rule
- Rule 2: Guaranteed Access
- Rule 3: Systematic treatment of NULL
- Rule 4: Active Online Catalog
- Rule 5: Powerful and Well-Structured Language
- Rule 6: View Updation Rule
- Rule 7: Relational Level Operation
- Rule 8: Physical Data Independence
- Rule 9: Logical Data Independence
- Rule 10: Integrity Independence
- Rule 11: Distribution Independence
- Rule 12: Nonsubversion Rule

Ví dụ mô hình dữ liệu quan hệ

MON_HOC

maHP	tenHP	soTC
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

Mô hình dữ liệu quan hệ

- **Thuộc tính** (\sim 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:
 - Sinhviên (MãSV, TênSV, Nămsinh, GiớiTính, ĐịaChỉ)
 - $\text{dom}(\text{MãSV}) = \{\text{char}(5)\}$
 - $\text{dom}(\text{TênSV}) = \{\text{char}(30)\}$
 - $\text{dom}(\text{Nămsinh}) = \{\text{date}\}$
 - $\text{dom}(\text{GiớiTính}) = \{0, 1\}$
 - $\text{dom}(\text{ĐịaChỉ}) = \{\text{char}(50)\}$

Mô hình dữ liệu quan hệ

- **Quan hệ** (\sim 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

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:

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

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

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

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'

Mô hình dữ liệu quan hệ

Ví dụ:

- Quan hệ: SinhViên, Lớp
- Siêu khoá: {MãSV, HọTên};
- Khoá tối thiểu: {MãSV}; {HọTên, NămSinh}
- Khoá ngoài: TênLớp nếu coi nó là khoá chính của quan hệ Lớp

MãSV	HọTên	NămSinh	GiớiTính	TênLớp
001	Nguyễn Văn An	1999	1	KHMT1
002	Nguyễn Văn An	1998	1	KHMT2
003	Lê Văn Cường	1997	1	KHMT2
004	Nguyễn Thùy Linh	1997	0	KTMT1

Tên Lớp	SĩSố
KHMT1	60
KHMT2	65
KTMT1	58



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



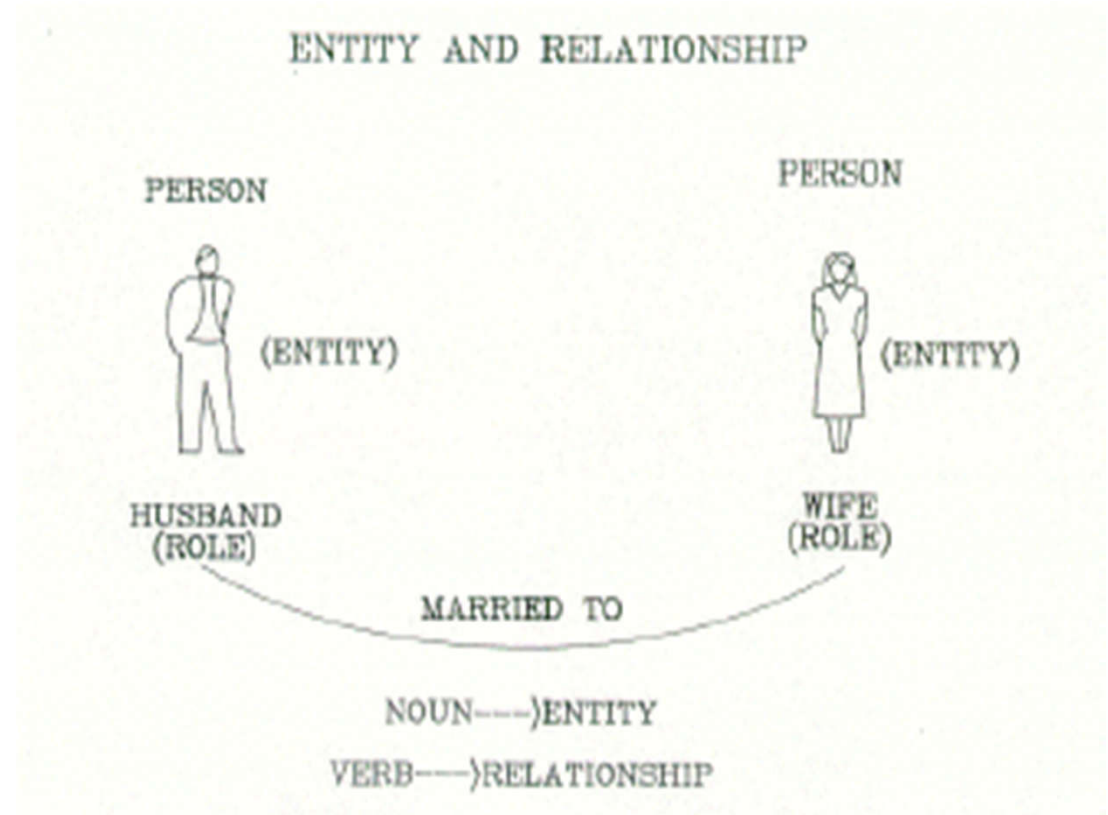
Mô hình thực thể liên kết (*Entity-Relationship data model*)

- 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

Mô hình thực thể liên kết

Các khái niệm cơ bản

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



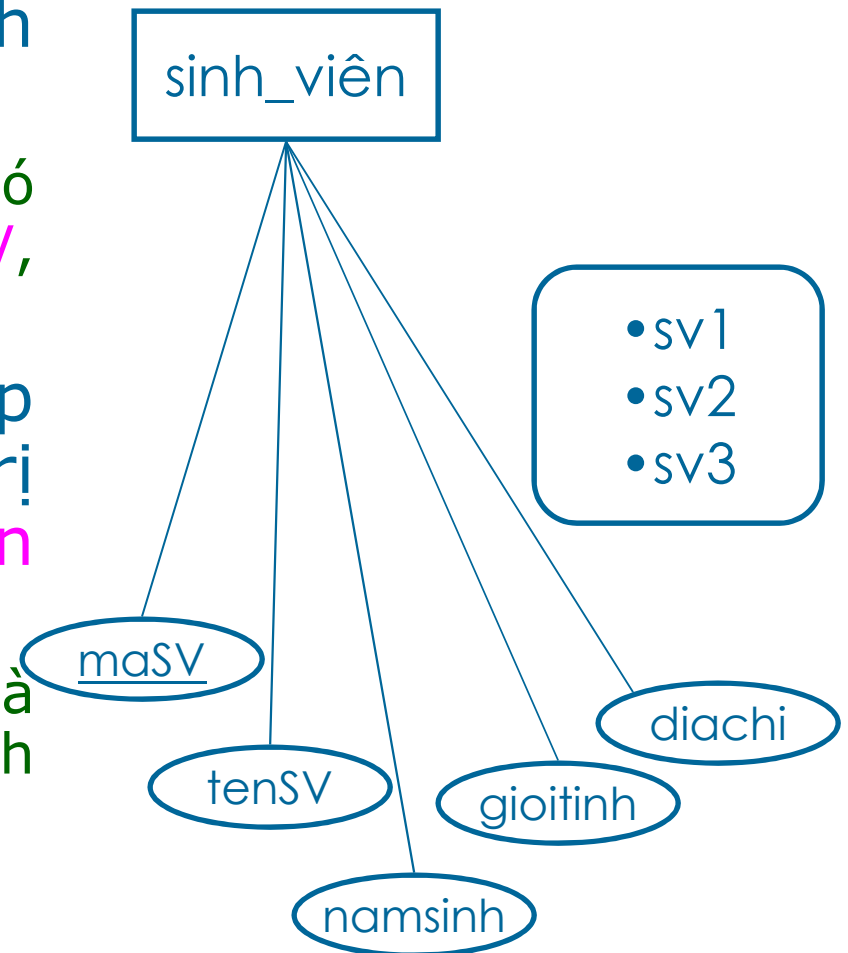


Mô hình thực thể liên kết...

- **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

Mô hình thực thể liên kết...

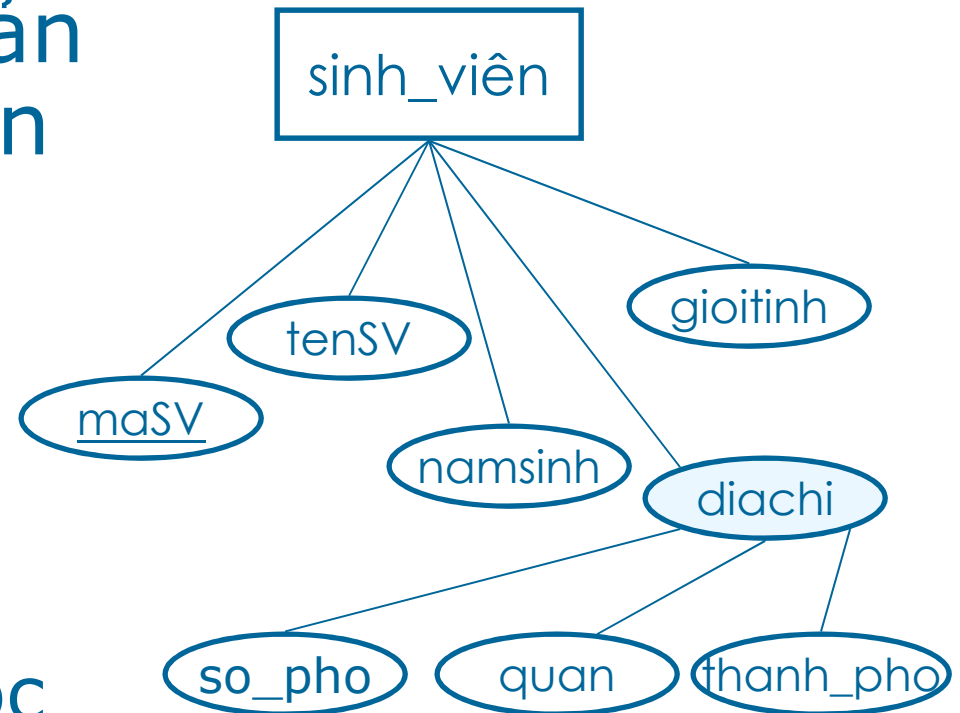
- **Thuộc tính** là đặc tính của một tập thực thể
 - Tập thực thể **SinhViên** có các thuộc tính như: **TênSV**, **NămSinh**,...
- Mỗi thực thể trong tập thực thể có một giá trị đặc tính nằm trong **miền giá trị của thuộc tính**
 - Sinh viên 1 có: Họ tên là Nguyễn Hải Anh, Năm sinh 1980



Mô hình thực thể liên kết...

Kiểu thuộc tính

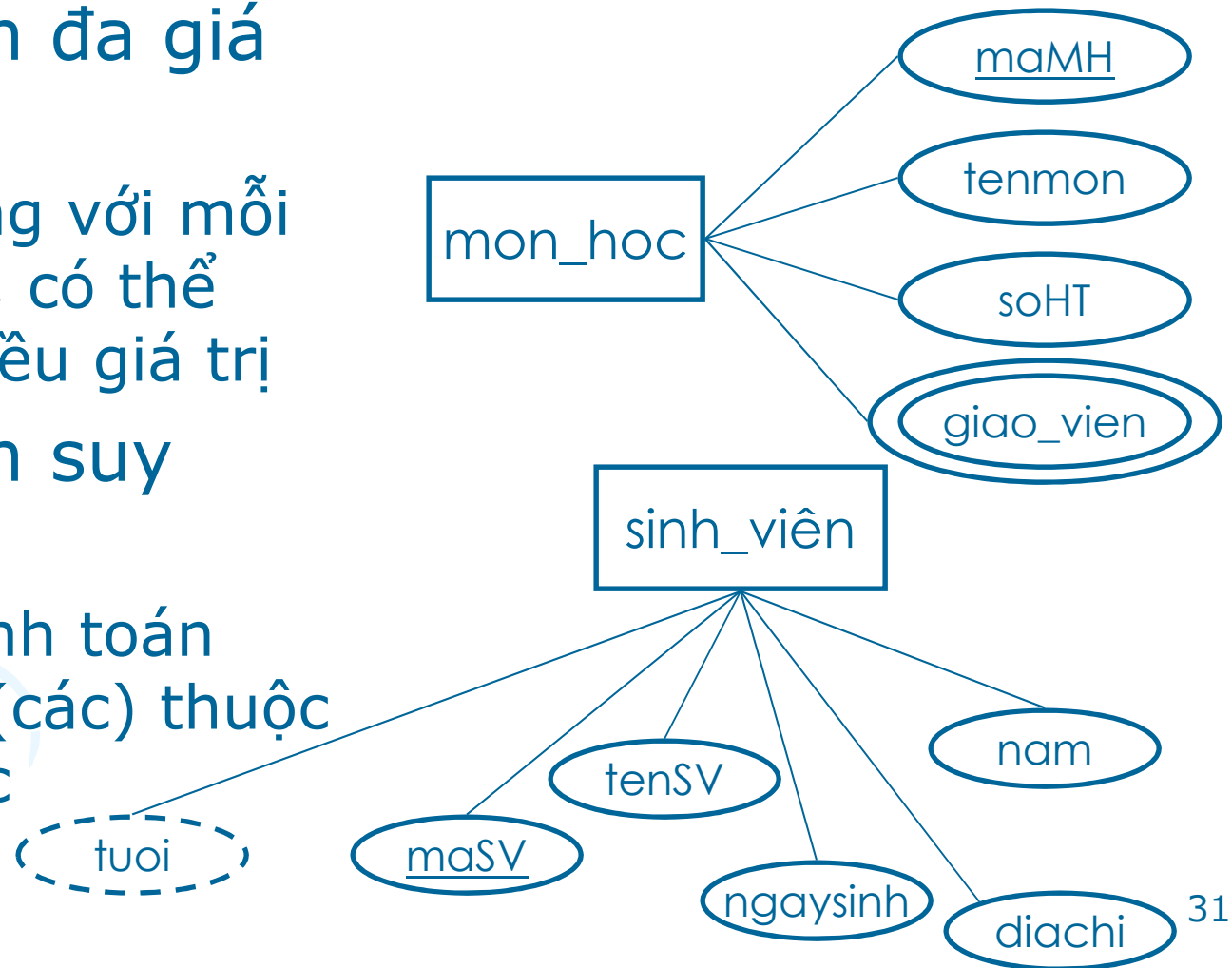
- Thuộc tính đơn giản (thuộc tính nguyên tố)
 - có kiểu dữ liệu nguyên tố
- Thuộc tính phức
 - có kiểu phức, định nghĩa bởi các thuộc tính khác



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



Mô hình thực thể liên kết...

Khóa

- 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 khóa
- Khóa gồm nhiều thuộc tính thì gọi là **khóa phức**
- Một tập thực thể có thể có nhiều khóa nhưng chỉ một trong số các khóa được chọn làm **khóa chính**
- Trong sơ đồ ER, thuộc tính nào được chọn làm khóa chính sẽ được **gạch chân**

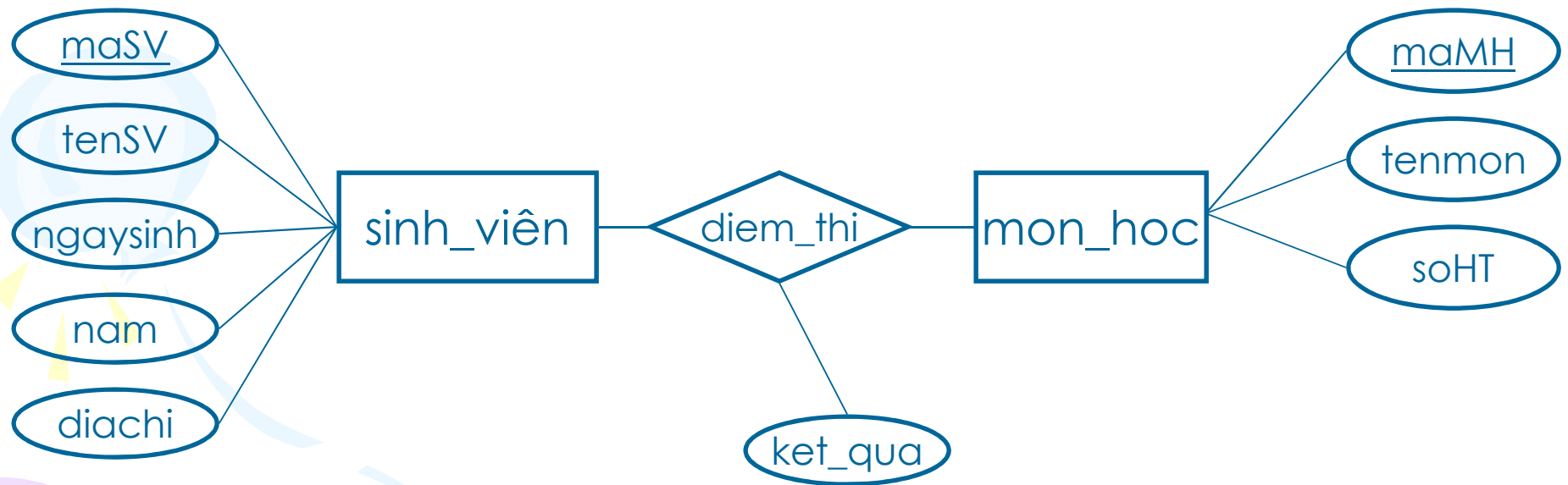
Mô hình thực thể liên kết...

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

Mô hình thực thể liên kết...

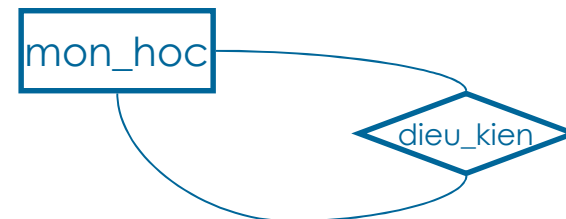
Liên kết - Tập liên kết - Ví dụ:



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





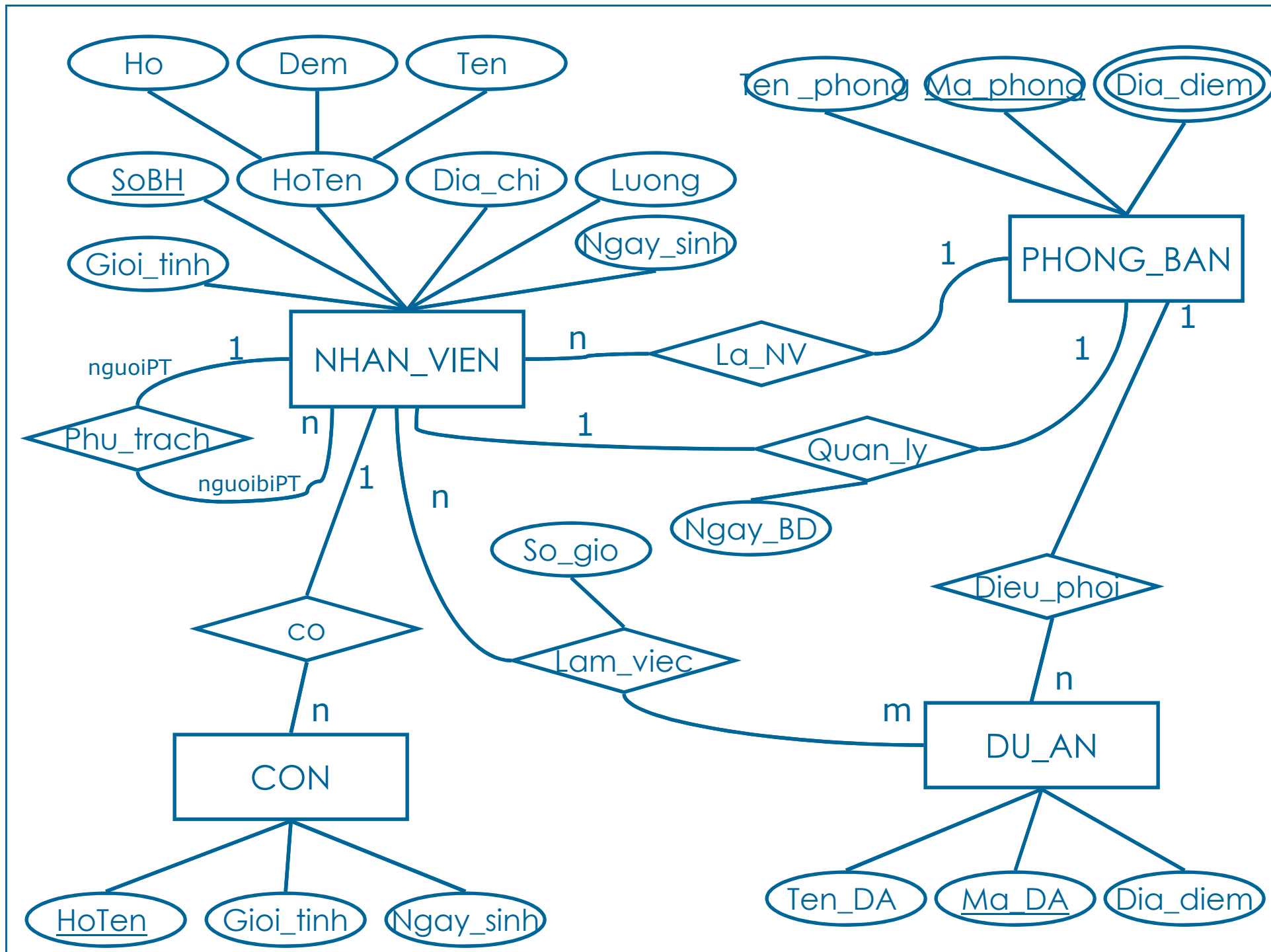
Cách lập sơ đồ thực thể - liên kết

- 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)



Bài tập: Vẽ sơ đồ ER

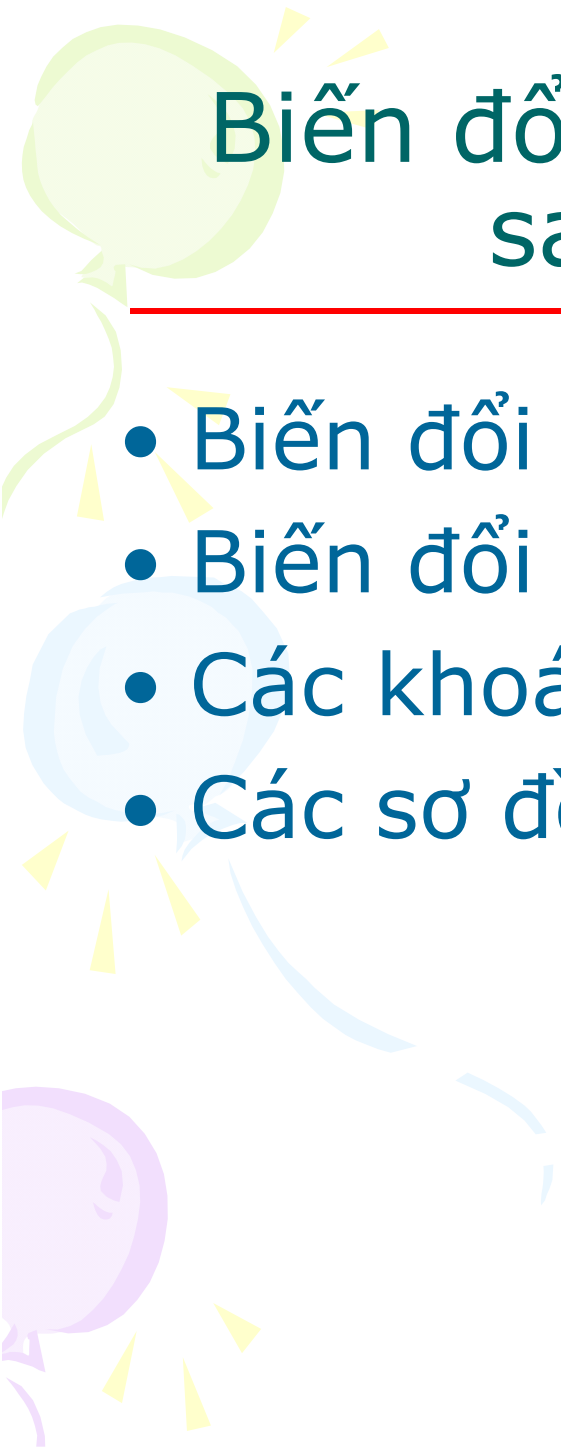
- Bài toán: phân tích và thiết kế 1 CSDL gồm các thông tin trong 1 công ty (nhân viên, phòng ban, dự án)
 - Công ty được tổ chức bởi các phòng ban. Mỗi phòng ban có 1 tên duy nhất, 1 số duy nhất và 1 người quản lý (thời điểm bắt đầu công tác quản lý của người này cũng được lưu lại trong CSDL). Mỗi phòng ban có thể có nhiều trụ sở làm việc khác nhau
 - Mỗi phòng 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





Các công cụ online giúp vẽ ERD

- Lucidchart
 - <https://www.lucidchart.com>
- Drawio
 - <https://www.draw.io/>
- Một số mô hình ERD tham khảo
 - http://www.databaseanswers.org/data_models/index.htm

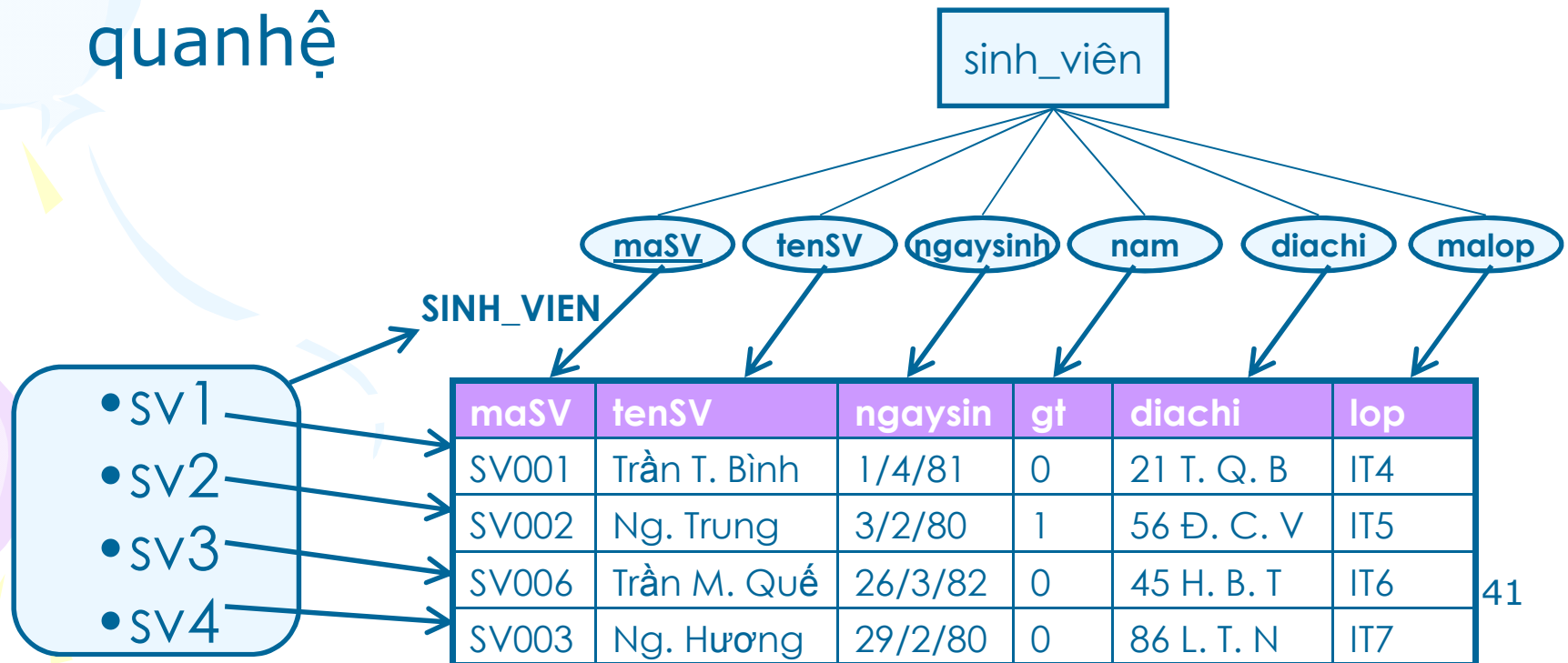


Biến đổi sơ đồ thực thể liên kết sang sơ đồ quan hệ

- 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

Biến đổi các tập thực thể

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



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)



Biến đổi các liên kết (tiếp)

- **Bước 4:** Liên kết 1-n

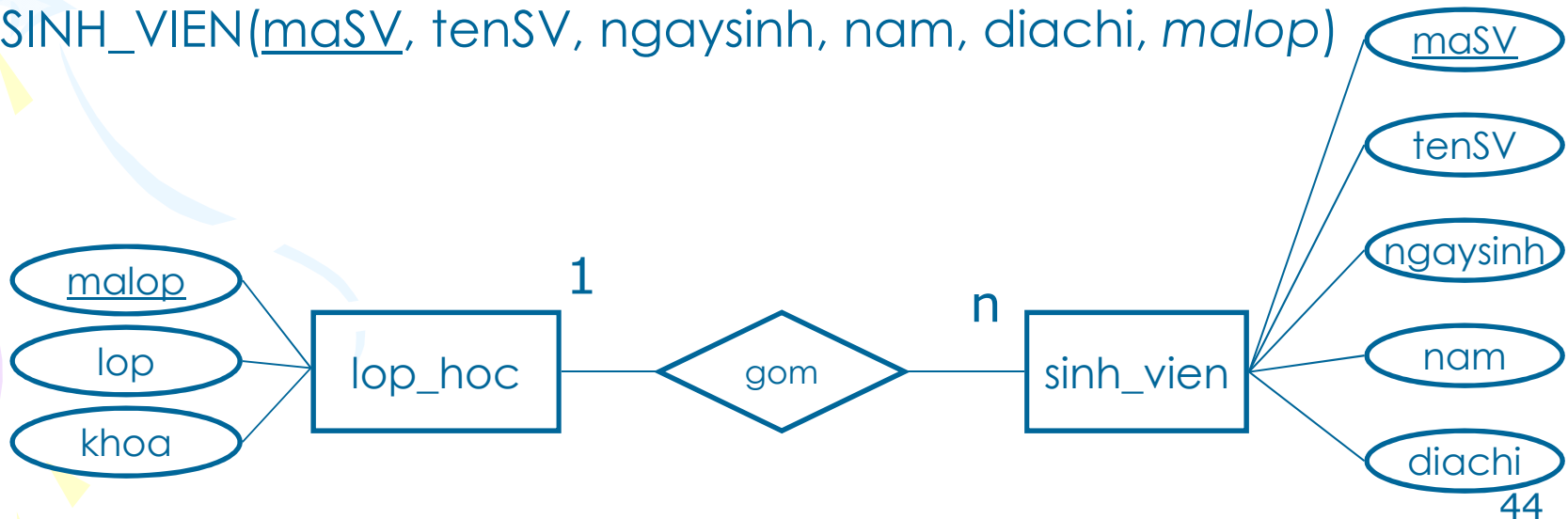
- Thêm 1 quan hệ mới xác định bởi các thuộc tính nằm trong khoá của các thực thể có liên quan

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)

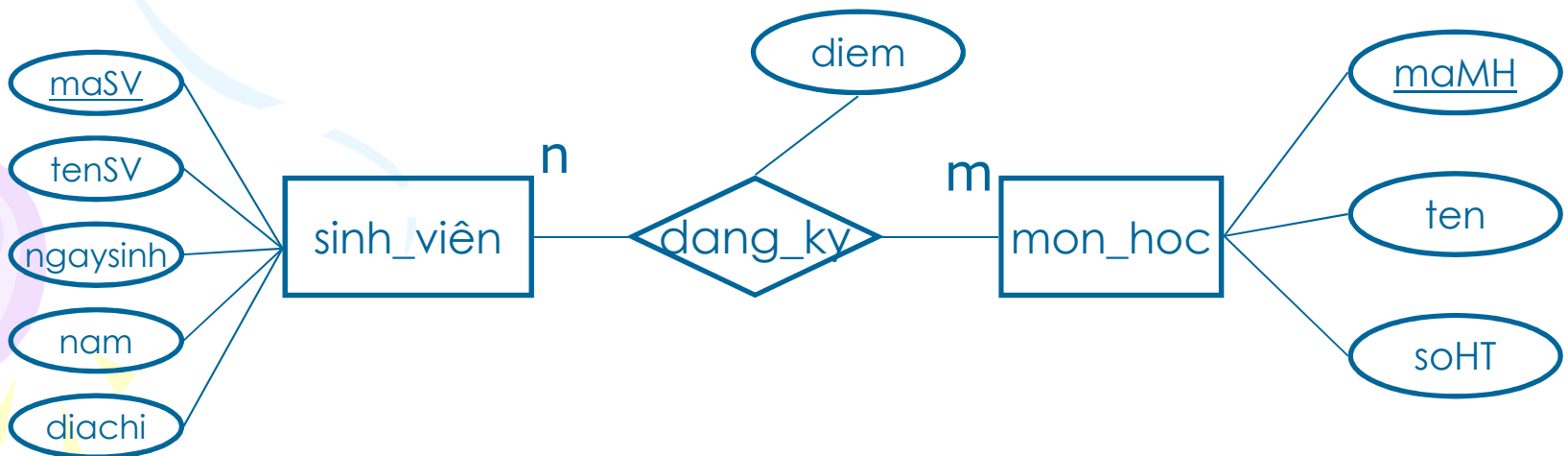


Biến đổi các liên kết (tiếp)

- **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

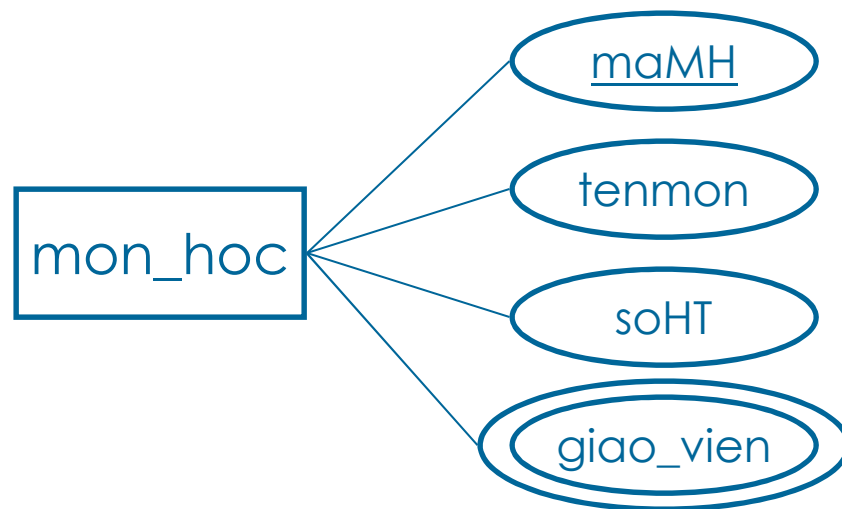
DANG_KY(maSV, maMH, diem)



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)





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



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);  
}
```




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

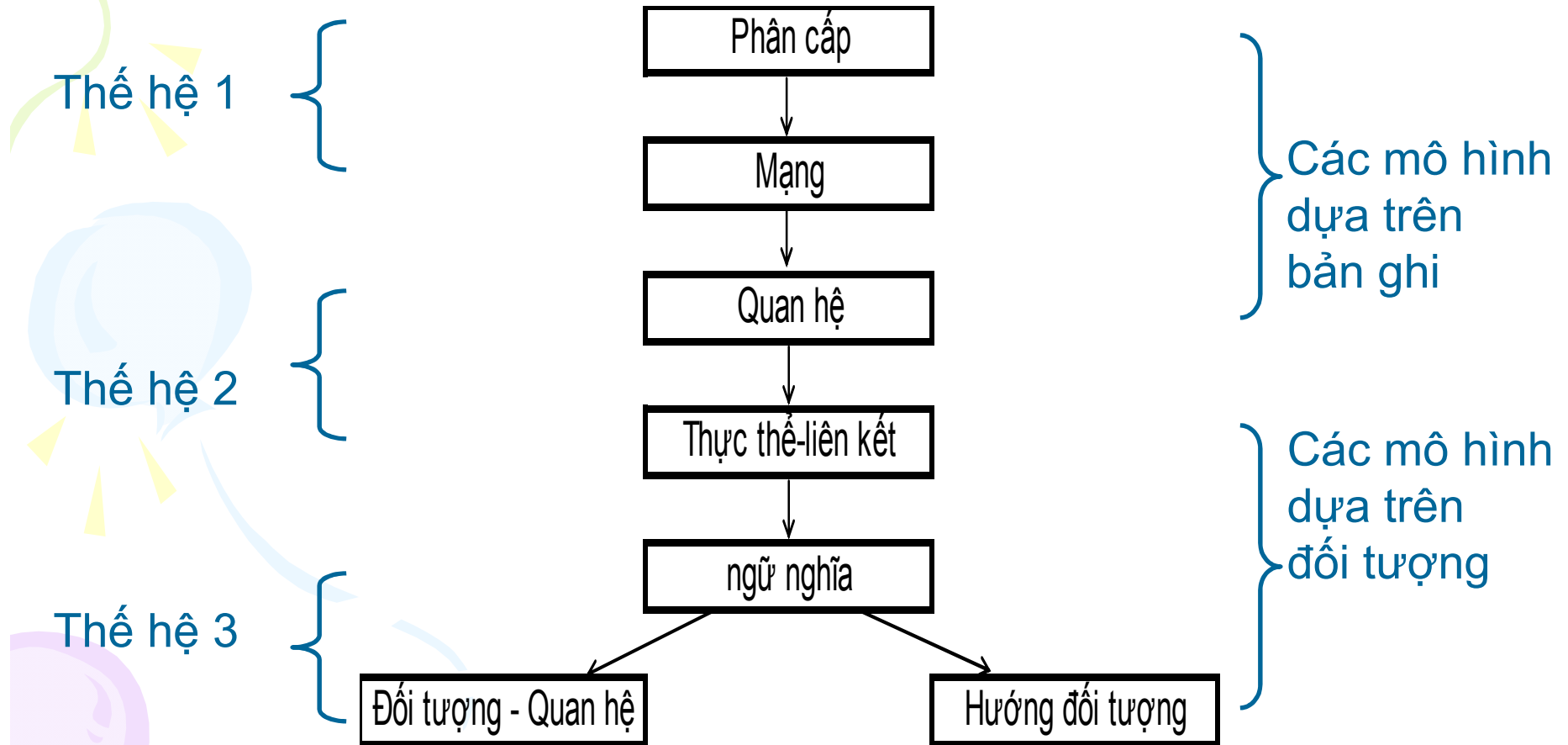


So sánh và đánh giá

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
biểu diễn ngữ nghĩa DL	hạn chế	hạn chế	<i>tương đối đa dạng</i>	đa dạng	đa dạng
lưu trữ DL	s/d nhiều con trỏ	dữ liệu lặp lại	dễ dàng và hiệu quả	khó lưu trữ	<i>cấu trúc phức tạp</i>
khả năng truy vấn	đơn giản	đơn giản	đa dạng		đa dạng
hiệu quả của truy vấn	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>

Phân loại các mô hình

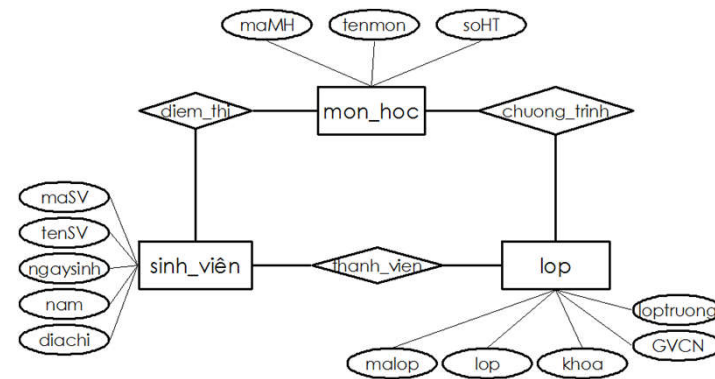
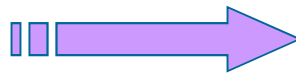


Các bước xây dựng một hệ CSDL



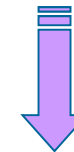
Mô tả ứng dụng

1: PHÂN TÍCH

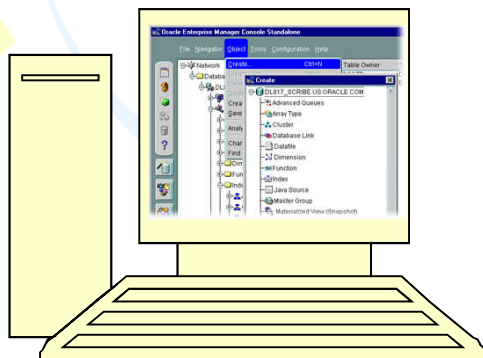
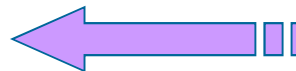


Mô hình hoá DL (vd: Sơ đồ thực thể-liên kết)

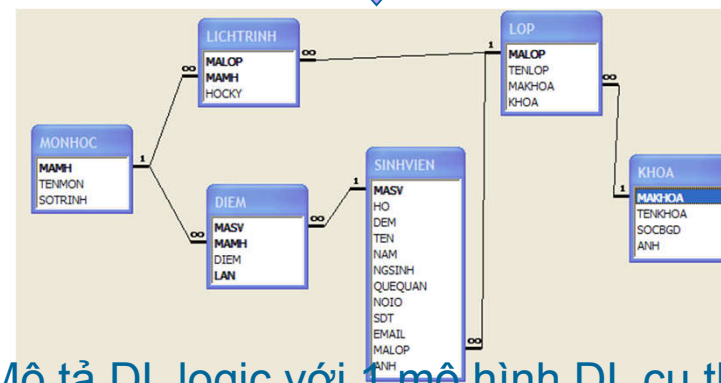
2: THIẾT KẾ



3: CÀI ĐẶT



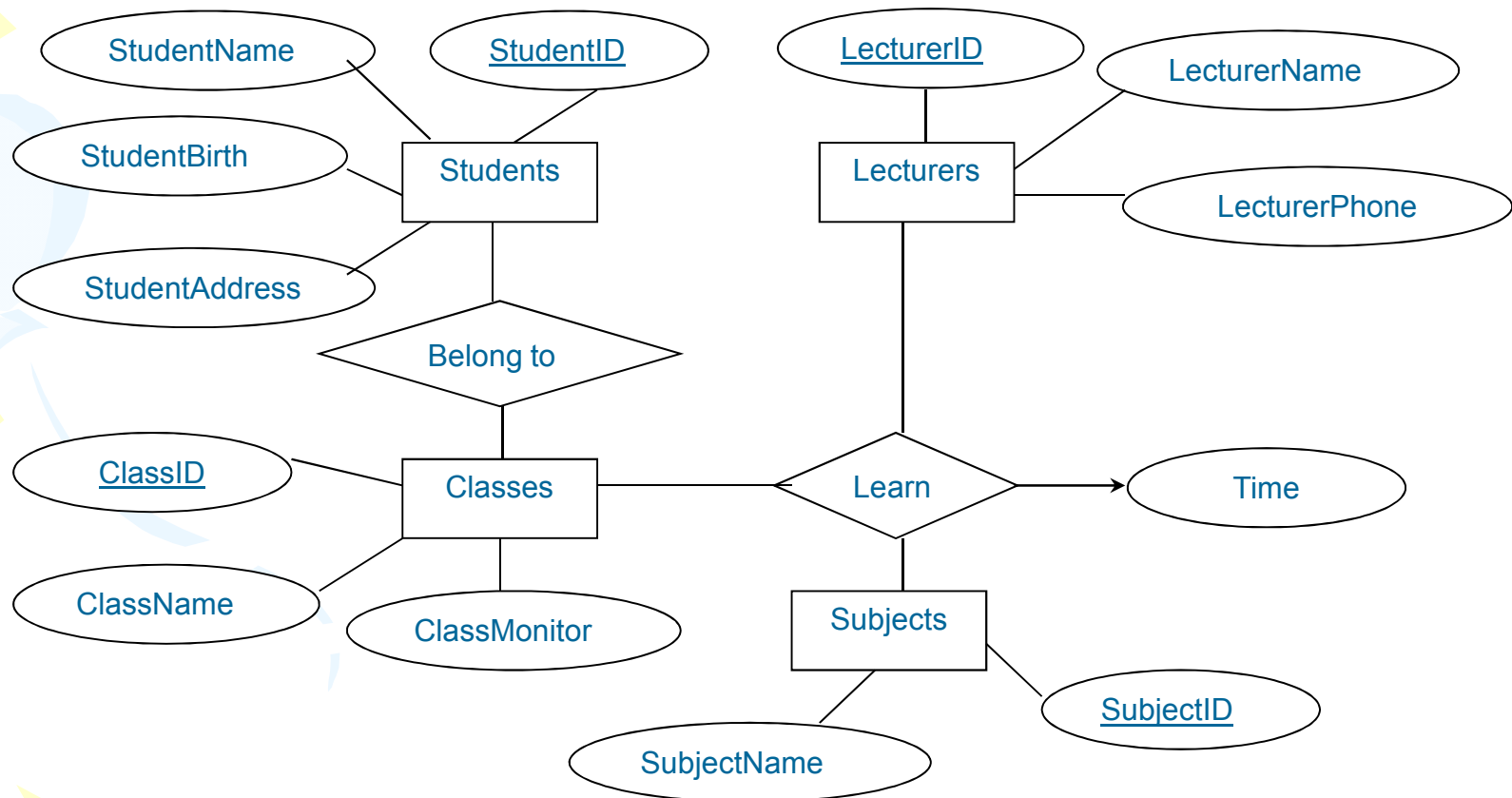
Cài đặt với 1 hệ quản trị CSDL
(vd: ORACLE)



Mô tả DL logic với 1 mô hình DL cụ thể
(vd: Sơ đồ quan hệ)

Bài tập

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





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)



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)



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