

**ĐẠI HỌC QUỐC GIA TP.HCM**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**NHẬP MÔN**  
**CÔNG NGHỆ PHẦN MỀM**

**Giảng viên: TS. Đỗ Thị Thanh Tuyền**

**Email: [tuyendtt@uit.edu.vn](mailto:tuyendtt@uit.edu.vn)**

# NỘI DUNG MÔN HỌC

- Tổng quan về Công nghệ phần mềm
- Xác định và mô hình hóa yêu cầu phần mềm
- **Thiết kế phần mềm: – Thiết kế DỮ LIỆU**
- Cài đặt phần mềm
- Kiểm thử và bảo trì
- Đồ án môn học

# THIẾT KẾ DỮ LIỆU

- 1. Sơ đồ Logic**
- 2. Xác định khóa chính**
- 3. Các kiểu mã hóa**
- 4. Bảng THAMSO**
- 5. Thuật toán thiết kế dữ liệu**
- 6. Các dạng chuẩn**
- 7. Chuẩn hóa dữ liệu**

# Thiết kế dữ liệu

- Mục tiêu của việc thiết kế dữ liệu là nhằm mô tả cách thức tổ chức lưu trữ dữ liệu của phần mềm bên trong máy tính.
- Kết quả của quá trình thiết kế dữ liệu là xây dựng được sơ đồ Logic.
- Khi thiết kế dữ liệu, ta quan tâm đến ba vấn đề sau:
  - Thiết kế dữ liệu với tính đúng đắn;
  - Thiết kế dữ liệu với tính tiến hóa;
  - Thiết kế dữ liệu với yêu cầu hiệu quả về thời gian truy xuất và dung lượng lưu trữ.

# 1. Sơ đồ Logic

- Bao gồm các bảng dữ liệu và mối quan hệ giữa chúng.
- Các ký hiệu:

**Tên Table**

Bảng dữ liệu



Liên kết

# 1. Sơ đồ Logic (tt)

## ■ Các ký hiệu (tt):



- ⇒ Một phần tử của bảng A **xác định duy nhất** một phần tử của bảng B
- ⇒ Ngược lại, một phần tử của bảng B có thể tương ứng với **một hoặc nhiều** phần tử của bảng A
- ⇒ **Mỗi quan hệ giữa A và B là quan hệ 1-n**

Khi đó, **bảng A** sẽ chứa khóa chính của **bảng B**

# 1. Sơ đồ Logic (tt)

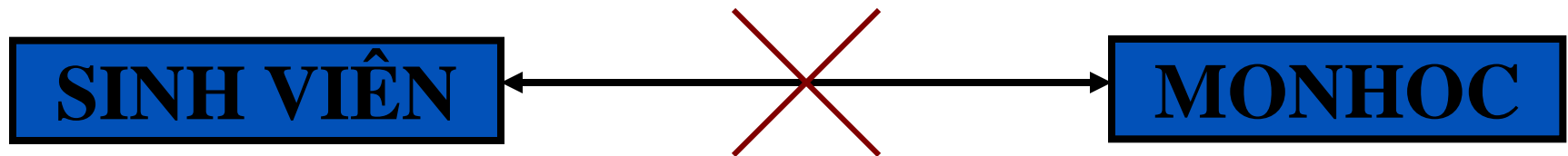
⇒ Nếu quan hệ giữa A và B là quan hệ 1-1 thì có thể gộp hai table A và B lại thành 1 table duy nhất chứa tất cả thuộc tính của A và B

**Lưu ý:** *Quan hệ 1-n không làm được việc này.*

⇒ Nếu quan hệ giữa A và B là quan hệ n-n:

Tách quan hệ này thành 2 quan hệ 1-n bằng cách thêm vào 1 table trung gian chứa khóa chính của A và B

# Ví dụ Quan hệ n-n

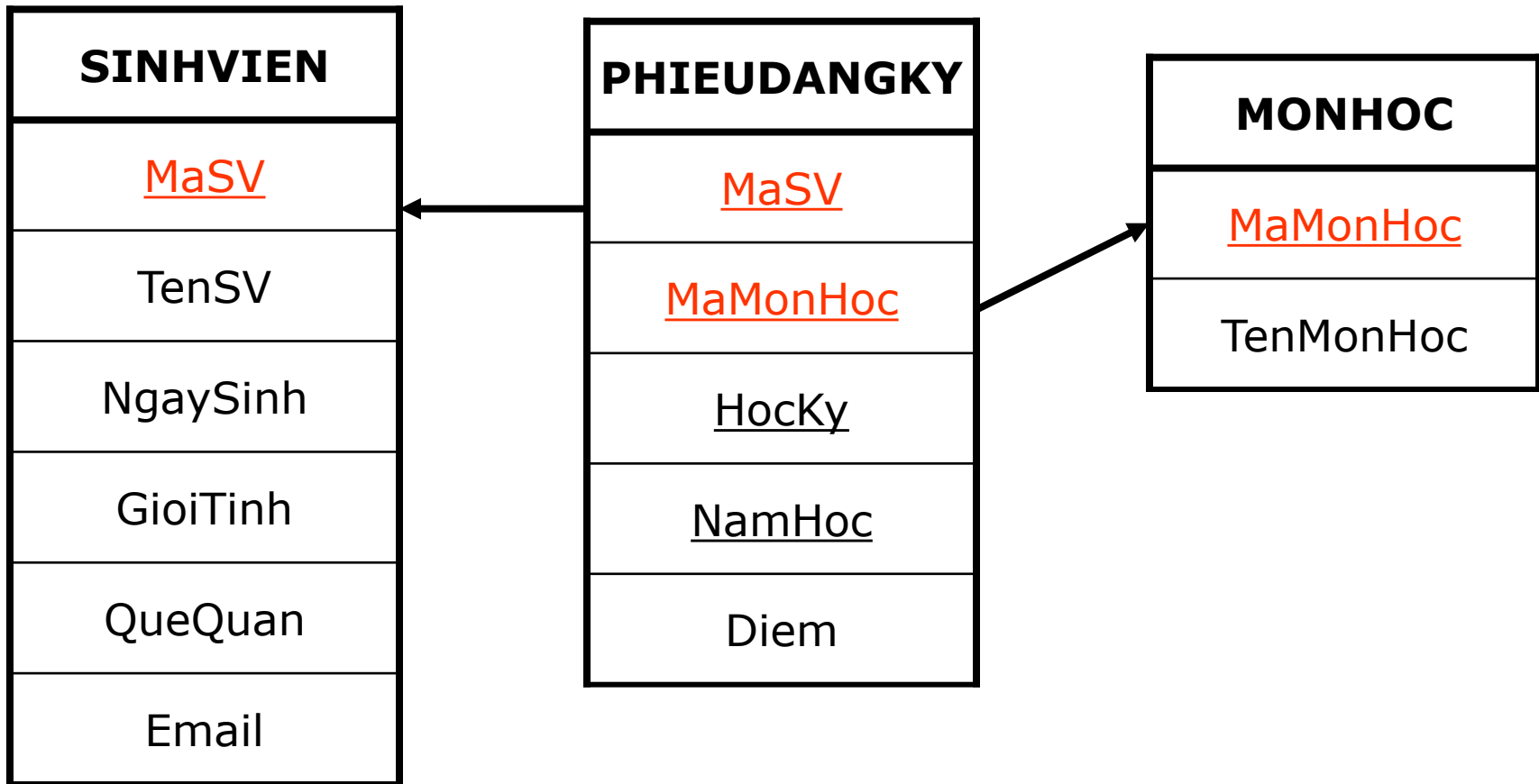


*Chuyển thành:*





# Ví dụ Quan hệ n-n (tt)



# Lưu ý

- **Tên Table:** viết bằng chữ IN **tất cả các ký tự**, không dấu, không có khoảng cách giữa các từ.

Ví dụ: NHANVIEN, KHACHHANG, ...

- **Tên thuộc tính:** viết bằng chữ IN **ký tự đầu** của mỗi từ, không dấu, không có khoảng cách giữa các từ.

Ví dụ: HoTen, NgaySinh, DiaChi, ...

- Đặt tên table/tên thuộc tính xúc tích, cô đọng và nhất quán trong toàn bộ CSDL.

Ví dụ: ~~HOSODAILY~~ -> DAILY

## 2. Xác định Khoá chính

### ■ Tính chất của Khoá chính:

- ✓ Tối thiểu;
- ✓ NOT NULL;
- ✓ Không trùng lặp;
- ✓ Không thay đổi theo thời gian.

### ■ Thuộc tính trừu tượng:

Là thuộc tính không xuất hiện trong thế giới thực, chỉ có trong phần mềm.

Sử dụng thuộc tính trừu tượng để làm khoá chính cho table.

Ví dụ: MaDaiLy, MaLoaiDaiLy, ...

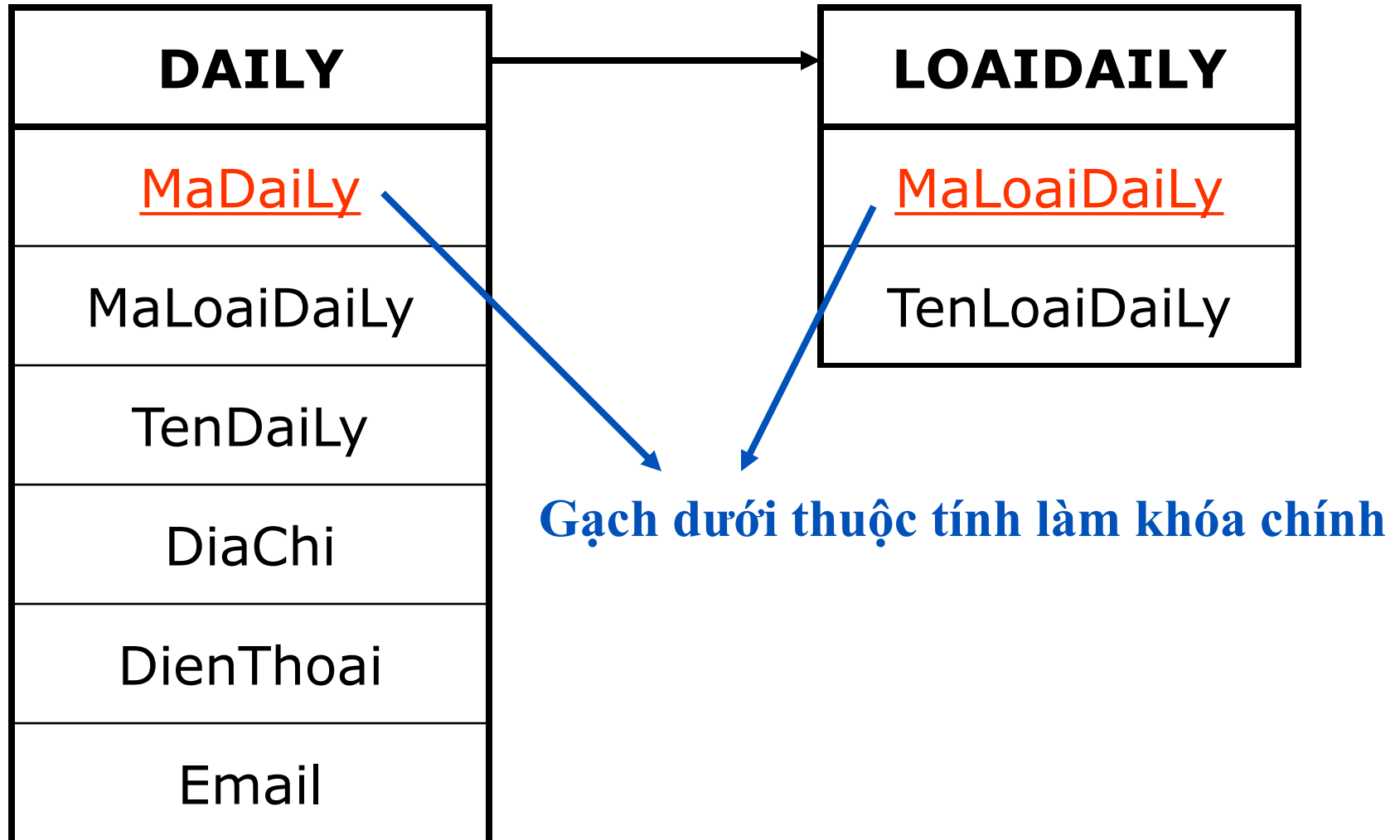
## 2. Xác định Khoá chính (tt)

### ■ Khi nào cần sử dụng thuộc tính trừu tượng?

- ✓ Khi từ danh sách các thuộc tính đã có của table, không chọn được thuộc tính (hoặc tổ hợp thuộc tính) nào thoả các tính chất của khoá chính.
- ✓ Khi khoá chính là một tổ hợp có từ hai thuộc tính trở lên.

*Lưu ý trong trường hợp này vẫn phải kiểm tra dữ liệu trùng trên bộ thuộc tính có thể tham gia làm khoá chính (khi insert hoặc update dữ liệu cho table).*

## 2. Xác định Khoá chính (tt)



## 2. Xác định Khoá chính (tt)

- **Xác định kiểu dữ liệu cho thuộc tính khóa:**
  - ✓ Cân nhắc lựa chọn giữa kiểu số và kiểu chuỗi;
  - ✓ Sử dụng tối ưu chiều dài của mã đồng thời phải xem xét khả năng mở rộng.
- Khi tạo giá trị cho khoá chính, ***không nên dùng lại một mã đã sử dụng*** cho dù đối tượng có mã đó đã bị xóa.

# 3. Các kiểu mã hóa

- **Mã hóa liên tiếp:** 1,2,3...
- **Mã hóa theo lát:** dùng từng lát cho từng nhóm đối tượng, trong mỗi lát thường dùng kiểu mã hóa liên tiếp.
- **Mã hóa phân đoạn:** mã được phân thành nhiều đoạn, mỗi đoạn mang một ý nghĩa riêng.
- **Mã hóa phân cấp:** là mã hóa phân đoạn, mỗi đoạn trở đến một tập hợp các đối tượng và các đối tượng này được phân cấp theo thứ tự từ trái qua phải.
- **Mã hóa diễn nghĩa:** gán một tên ngắn gọn nhưng hiểu được cho từng đối tượng.

Ví dụ: HAN (Hà Nội), HCM (Hồ Chí Minh)...

## 4. Bảng THAMSO

- **Chức năng:** dùng để lưu các giá trị trong các **qui định** mà các giá trị này không liên quan đến bất kỳ đối tượng nào khác trong CSDL.
- Các giá trị này được dùng trong các ***biểu thức tính toán*** hoặc ***kiểm tra***.
- Trên table THAMSO chỉ hỗ trợ cho người dùng cuối 2 thao tác ***select*** và ***update***, không cho phép ***insert*** và ***delete*** sau khi đã hoàn tất việc thiết kế.



## 4. Bảng THAMSO (tt)

### ■ Cấu trúc của bảng THAMSO:

#### ➤ **Cách 1: THAMSO(TenThamSo, GiaTri)**

- Các tham số là các record của table THAMSO
- Qui đổi giá trị của tham số có kiểu Boolean về kiểu số:

True  $\Leftrightarrow$  1; False  $\Leftrightarrow$  0

- Đặt tên các tham số theo qui định về cách đặt tên thuộc tính.

#### ➤ **Cách 2: THAMSO(ThamSo1,ThamSo2,...,ThamSoX)**

- Các tham số là các thuộc tính của table THAMSO
- Mỗi thuộc tính có kiểu dữ liệu riêng, vì vậy không phải qui đổi giá trị của tham số có kiểu Boolean về kiểu số.

# 5. Thuật toán thiết kế dữ liệu

- Thiết kế dữ liệu dựa vào **sơ đồ luồng dữ liệu (ca sử dụng mức 2, Biểu đồ lớp ca sử dụng, Biểu đồ cộng tác) của yêu cầu phần mềm đang xét.**
- Các bước thực hiện:
  - ❖ **Bước 1: Xét yêu cầu phần mềm thứ I**
    - *Thiết kế dữ liệu với tính đúng đắn*
    - *Thiết kế dữ liệu với tính tiến hóa*
  - ❖ **Bước 2: Xét yêu cầu phần mềm thứ II**
  - ...
  - ❖ **Bước n: Xét yêu cầu phần mềm cuối cùng**

## 5. Thuật toán thiết kế dữ liệu (tt)

### ➤ Thiết kế dữ liệu với tính đúng đắn:

- **Biểu mẫu** liên quan: BM<sub>x</sub>
- Sơ đồ luồng dữ liệu/Biểu đồ ca sử dụng, Biểu đồ lớp use case, biểu đồ cộng tác (Ký hiệu biểu đồ - UC<sub>x</sub>)
- Các thuộc tính mới:
- **Thiết kế dữ liệu:** lần lượt bố trí các thuộc tính mới vào các bảng đã có, trong trường hợp không bố trí được thì phải tạo bảng mới để chứa các thuộc tính mới này.
- Các thuộc tính trừu tượng:
- Sơ đồ Logic

## 5. Thuật toán thiết kế dữ liệu (tt)

### ➤ Thiết kế dữ liệu với tính tiến hóa:

- **Qui định** liên quan: QĐ<sub>x</sub>
- Sơ đồ luồng dữ liệu/Biểu đồ ca sử dụng, Biểu đồ lớp use case, biểu đồ cộng tác (Ký hiệu biểu đồ - UC<sub>x</sub>)
- Các thuộc tính mới:
- **Các tham số mới:**
- **Thiết kế dữ liệu:** lần lượt bố trí các thuộc tính mới và các tham số mới vào các bảng đã có, trong trường hợp không bố trí được thì phải tạo bảng mới để chứa các thuộc tính mới và các tham số mới này.
- Các thuộc tính trừu tượng:

# Thiết kế dữ liệu với yêu cầu hiệu quả về thời gian truy xuất và dung lượng lưu trữ

- **Hiệu quả về thời gian truy xuất:**

- + Thêm vào các **thuộc tính tính toán**.

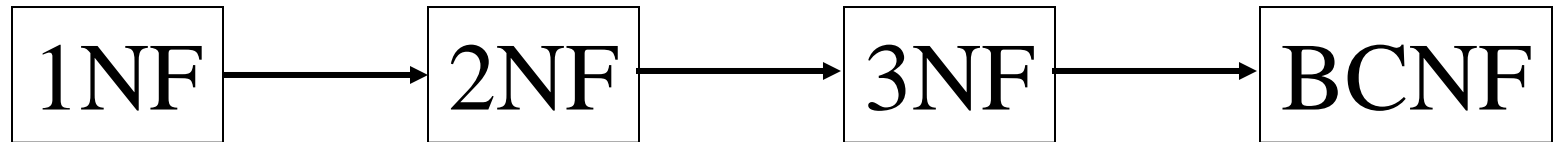
*Lưu ý:* giá trị này phải được **tự động cập nhật** khi có những thay đổi liên quan.

- **Hiệu quả về dung lượng lưu trữ:**

- + Tách bảng có các giá trị cố định được lặp lại nhiều lần thành 2 bảng: một bảng chứa **thông tin tổng quát** và một bảng chứa **thông tin chi tiết**.

- + Thêm đối tượng mới và sử dụng mã của đối tượng thay cho việc lưu trữ thông tin chi tiết về đối tượng.

## 6. Các dạng chuẩn<sup>2</sup>



# Dạng chuẩn 1 (1 Normal Form) – 1NF

Một quan hệ được gọi là ở dạng chuẩn 1 nếu và chỉ nếu toàn bộ miền giá trị của tất cả thuộc tính trong quan hệ đều **chỉ chứa các giá trị nguyên tố**.

# Dạng chuẩn 2 (2 Normal Form) – 2NF

## ■ Một quan hệ ở dạng chuẩn 2 nếu:

+ Thỏa 1NF;

+ Các thuộc tính không khoá **phụ thuộc đầy đủ** vào tập các thuộc tính tham gia làm khoá chính.

## ■ Ví dụ:

$R = (\underline{A}, B, C, D)$

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

$\Rightarrow$

$R_1(\underline{B}, C, D)$

$R_2(\underline{A}, B)$



# Dạng chuẩn 3 (3 Normal Form) – 3NF

## ■ Một quan hệ ở dạng chuẩn 3 nếu:

- + Thoả 2NF;
- + Các thuộc tính không khoá **phụ thuộc trực tiếp** vào khoá chính.

## ■ Ví dụ:

$R = (\underline{A}, \underline{B}, C, D, G, H)$

$F = \{A, B \rightarrow C; A, B \rightarrow D; \textcolor{red}{A, B \rightarrow G, H}; \textcolor{red}{G \rightarrow D, H}\}$

$\Rightarrow$

$R_1(\underline{G}, D, H)$

$R_2(\underline{A}, \underline{B}, C, G)$

# Dạng chuẩn Boyce Codd (Boyce Codd Normal Form) - BCNF

## ■ Một quan hệ ở dạng chuẩn Boyce Codd nếu:

- + Thỏa 3 NF;
- + Không có thuộc tính khoá phụ thuộc vào thuộc tính không khoá.

## ■ Ví dụ:

$R = (\underline{A}, \underline{B}, C, D, G, H)$

$F = \{A, B \rightarrow C; A, B \rightarrow D; A, B \rightarrow G, H; H \rightarrow B\}$

$\Rightarrow$

$R_1(\underline{H}, B)$


$R_2(\underline{A}, \underline{H}, C, D, G)$

# 7. Chuẩn hoá dữ liệu

- Chuẩn hoá dữ liệu là việc đưa quan hệ ban đầu về các dạng chuẩn (1, 2, 3, Boyce Codd).

*Cách thực hiện: tách quan hệ ban đầu thành các quan hệ nhỏ hơn dựa vào các phụ thuộc hàm.*

- Mục đích của chuẩn hoá dữ liệu là **nhằm loại bỏ việc dư thừa dữ liệu.**
- Tuy nhiên chuẩn hoá làm *tăng thời gian truy vấn* do phải thực hiện phép kết giữa các quan hệ (mới tách).
- Dạng chuẩn của cơ sở dữ liệu là dạng chuẩn của *quan hệ có dạng chuẩn thấp nhất* trong CSDL đó.



**Q & A**

# Ôn tập (tt)

- Kết quả của việc thiết kế dữ liệu là gì? Hãy mô tả nó.
- Trình bày các dạng quan hệ giữa table A và table B.
- Trình bày khái niệm và mục đích của việc sử dụng thuộc tính trừu tượng.
- Trình bày chức năng của bảng THAMSO.
- Trình bày thuật toán thiết kế dữ liệu.

# Ôn tập

- Hãy nêu và giải thích các tính chất của khoá chính.
- Trình bày các yêu cầu trong thiết kế dữ liệu.
- Trình bày khái niệm chuẩn hoá dữ liệu.
- Trình bày cách thực hiện chuẩn hoá dữ liệu.
- Trình bày mục đích của việc chuẩn hoá dữ liệu.
- Hãy nêu nhược điểm của việc chuẩn hoá dữ liệu.
- Dạng chuẩn của CSDL là dạng chuẩn của quan hệ có dạng chuẩn thấp nhất hay cao nhất trong CSDL đó?

# Phụ lục: Chuyển đổi từ sơ đồ lớp sang sơ đồ quan hệ

UML	EER
UML là ngôn ngữ mô hình hóa thống nhất chuẩn để thiết kế phần mềm hướng đối tượng	ER được sử dụng trong mô hình thiết kế cơ sở dữ liệu quan hệ
Các thành phần: Biểu đồ lớp trong UML Lớp (Class) Đối tượng (Object) Thuộc tính (Attribute) Miền giá trị (Domain) Miền cấu trúc (Structured Domain) Thao tác (Operation) Kết hợp (Association) Liên kết (Link) Bản số (Multiplicities)  Kết tập (Aggregation)	Mô hình ER (ER Diagram) Kiểu thực thể (Entity Type) Thực thể (Entity) Thuộc tính (Attribute) Miền giá trị (Domain) Thuộc tính phức hợp (Composite Attribute) ~ [Derived Attribute] Kiểu liên kết (Relationship Type) Thể liên kết (Relationship Instance) Tỷ số lực lượng (Cardinality & Participation) Thực thể yếu (Weak entities)

# Phụ lục: Chuyển đổi từ sơ đồ lớp sang sơ đồ quan hệ

## UML

Các kí hiệu:

- UML hiển thị lớp bằng hình chữ nhật có 3 phần:

- ü Phần trên cùng: Tên lớp
- ü Phần giữa: Thuộc tính
- ü Phần dưới cùng: Thao tác

Thuộc tính:

- Thuộc tính phức hợp được mô hình hóa bởi sự lùi vào ở đầu dòng.
- Miền giá trị gán cùng với thuộc tính

## EER

- Kiểu thực thể hiển thị bởi hình chữ nhật

- ü Tên kiểu thực thể
- ü Thuộc tính được hiển thị bởi hình elip
- ü Các thao tác không được hỗ trợ trong mô hình EER

- Thuộc tính phức hợp được mô hình hóa bởi các hình elip.

- Miền giá trị được chỉ rõ riêng rẽ



# Phụ lục: Chuyển đổi từ sơ đồ lớp sang sơ đồ quan hệ

UML	EER
<p>Các mối quan hệ:</p> <ul style="list-style-type: none"><li>- Trong UML gọi là các mối quan hệ giữa các lớp.</li><li>- Lực lượng tham gia vào liên hệ gọi là bản số. Bản số được ghi ở phía đầu đường thẳng thể hiện liên hệ, sát vào lớp là miền áp dụng của nó. Phạm vi số lượng phần tử có thể có trong liên hệ.</li></ul>	<ul style="list-style-type: none"><li>- Trong EER gọi là các mối quan hệ giữa các thực thể.</li><li>- Tỷ số lực lượng cực đại, cực tiểu tham gia vào liên kết.</li></ul>
<p>Bản số và lực lượng:</p> <p>0..*</p> <p>1 hoặc 0..1</p>	<p>0,N</p> <p>0,1</p>
<p>Thuộc tính riêng của quan hệ:</p> <p>Thuộc tính riêng của liên hệ giữa các lớp đối tượng.</p>	<p>Thuộc tính riêng của kiểu liên kết giữa các thực thể</p>