NHẬP MÔN
CÔNG NGHỆ PHẦN MỀM

Giảng viên: Đỗ Thị Thanh Tuyền

Email: dothithanhtuyen@gmail.com

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

- **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 mặt truy suất và lưu trữ

Sơ đồ Logic

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

Bảng dữ liệu

Liên kết

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

⇨ Bảng A chứa thuộc tính khóa của bảng B (là khóa ngoại của bảng A và là khóa chính của bảng B).

Sơ đồ Logic (tt)

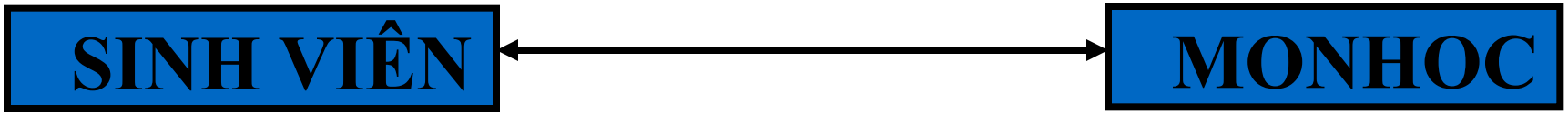
⇨ Nếu quan hệ giữa A và B là quan hệ 1-1 thì có thể gộp hai table A và B lại thành 1 table duy nhất chứa tất cả thuộc tính của A và B.

Quan hệ 1-n không làm được việc này.

⇨ **Nếu quan hệ giữa A và B là quan hệ n-n:**

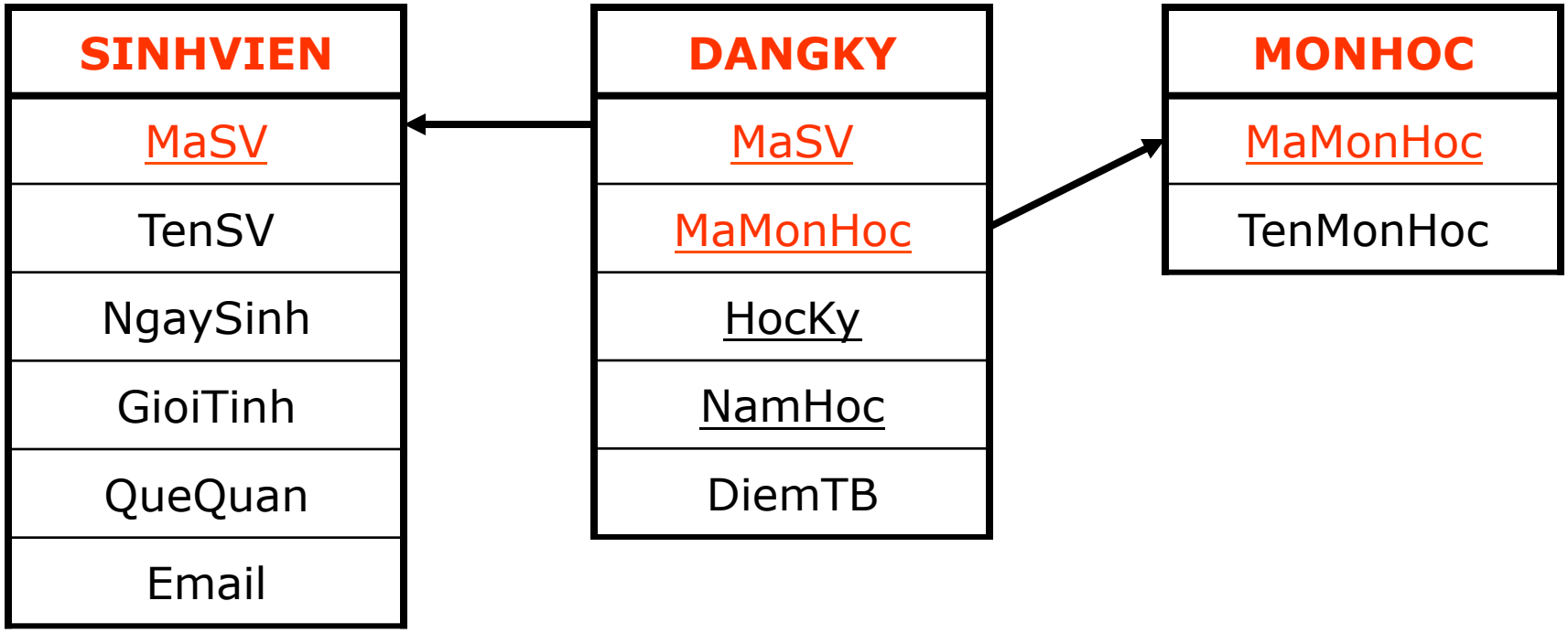
Tách quan hệ này thành 2 quan hệ 1-n bằng cách thêm vào 1 table trung gian chứa khóa chính của A và B.

Ví dụ Quan hệ n-n



Chuyển thành:

Ví dụ Quan hệ n-n (tt)



Xác định Khoá chính

- **3 Tính chất của Khoá chính:**
  - ✓ Tối thiểu;
  - ✓ Không trùng lắp (bao gồm NOT NULL);
  - ✓ 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.

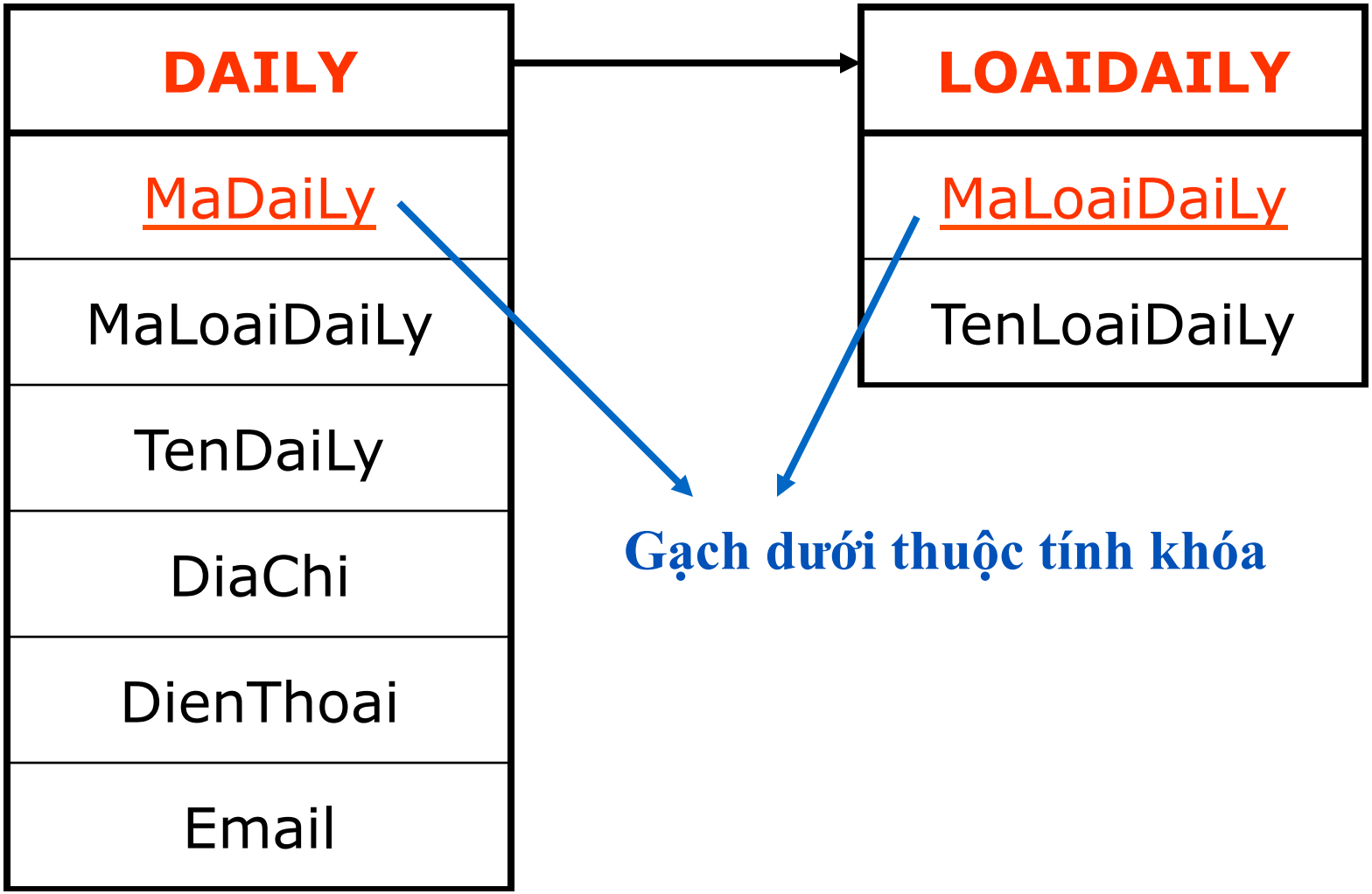
  Ví dụ: MaDaiLy, MaLoaiDaiLy…

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 khóa chính.
  - ✓ Khi khoá chính là một tổ hợp có từ hai thuộc tính trở lên.

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.

Xác định Khoá chính (tt)



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.**

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 trỏ 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)…

Bảng THAMSO

- **Chức năng:** dùng để lưu các giá trị trong các qui định mà các giá trị này không liên quan đến các đối tượng dữ liệu khác.
- **Các giá trị này dùng trong các biểu thức tính toán hoặc kiểm tra.**
- **Trên table THAMSO chỉ có các thao tác select và update, không có insert và delete sau khi đã hoàn tất việc thiết kế.**

Bảng THAMSO (tt)

- **Cấu trúc của bảng THAMSO:**

  C1: **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 ⇔ 1; False ⇔ 0
  - Đặt tên các tham số theo qui định về cách đặt tên của thuộc tính.

  C2: **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.

Lưu ý

- **Tên Table: viết bằng chữ IN HOA, không dấu, không có khoảng cách giữa các từ.**

  Ví dụ: **NHANVIEN, KHACHHANG…**

- **Tên thuộc tính: viết hoa các 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 của table súc tích, cô đọng và nhất quán trong toàn bộ CSDL.**

  Ví dụ: ~~**HOSODAILY**~~ **-> DAILY**

Thuật toán thiết kế dữ liệu

- **Thiết kế dữ liệu dựa vào sơ đồ luồng dữ liệu 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 thứ n (cuối cùng)**

Thuật toán thiết kế dữ liệu (tt)

➢ ***Thiết kế dữ liệu với tính đúng đắn:***

- **Biểu mẫu liên quan:** BMx
- **Sơ đồ luồng dữ liệu:** SĐx
- **Các thuộc tính mới:**
- **Thiết kế dữ liệu:** 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**

Thuật toán thiết kế dữ liệu (tt)

- ***Thiết kế dữ liệu với tính tiến hóa:***
  - **Qui định liên quan:** QĐx
  - **Sơ đồ luồng dữ liệu về việc thay đổi qui định:** SĐy
  - **Các thuộc tính mới:**
  - **Các tham số mới:**
  - **Thiết kế dữ liệu:** 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**

Thiết kế dữ liệu với yêu cầu hiệu quả về mặt truy suất và lưu trữ

- **Hiệu quả về mặt truy suất:**

  + Thêm vào các thuộc tính tính toán.

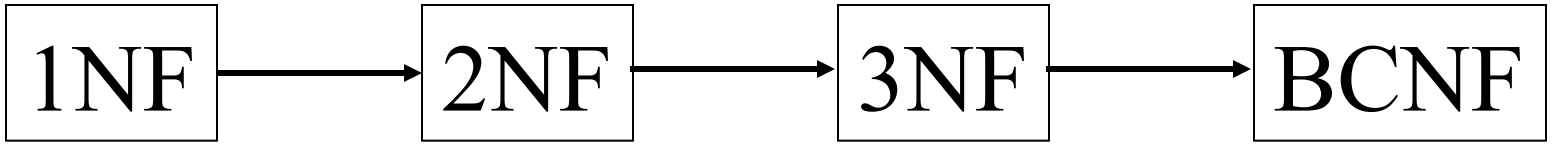
  + Lưu ý: giá trị này phải được tự động cập nhật khi có những thay đổi liên quan.

- **Hiệu quả về mặt 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.

Các dạng chuẩn



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 các 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:**
  + Thoả 1NF;
  + Các thuộc tính không khoá phụ thuộc đầy đủ vào tập các thuộc tính tham gia làm khoá chính.
- **Ví dụ:**

R = (A,B,C,D)

F = {A,B -> C, A,B -> D, B -> D,C}

=>

R1(B,C,D)

R2(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 = (A,B,C,D,G,H)

F = {A,B -> C; A,B -> D; A,B -> G,H; G -> D,H}

=>

R1(G,D,H)

R2(A,B,C,G)

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

- **Một quan hệ ở dạng chuẩn Boyce Codd nếu:**
  + Thoả 3 NF;
  + Không có thuộc tính khoá phụ thuộc vào thuộc tính không khoá.
- **Ví dụ:**

R = (A,B,C,D,G,H)

F = {A,B -> C; A,B -> D; A,B -> G,H; H -> B}

=>

R1(H,B)

R2(A,H,C,D,G)

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