

# THIẾT KẾ LƯU TRỮ DỮ LIỆU

Th.s Nguyễn Tấn Phương

Trường Đại học Bà Rịa- Vũng Tàu

Khoa Công nghệ thông tin

# Thiết kế lưu trữ dữ liệu

- ❖ Có những đối tượng cần phải lưu trữ lại một cách lâu dài (trên bộ nhớ ngoài), chứ không thể để mất đi cùng với sự kết thúc chương trình
- ❖ Đó là các đối tượng cần lưu trữ
  - Thường thì đó là các đối tượng thực thể
  - Hiếm khi là các đối tượng biên (giao diện) hay đối tượng điều khiển

# Thiết kế lưu trữ dữ liệu

1. Chọn lựa cách lưu trữ dữ liệu Việc mô hình hoá hệ thống bằng các lớp và đối tượng tạo nhiều thuận lợi cho việc lưu trữ dữ liệu. Tuy nhiên ngày nay có ba cách lưu trữ có thể dùng:
  - ❖ Các hệ thống tệp (tập tin): Đó là phương tiện lưu trữ đơn giản nhất :
    - Tuy cho phép đọc và viết các đối tượng, nhưng không thích hợp cho các truy vấn dữ liệu phức tạp
    - Ít được dùng với các hệ thống lớn, có tầm quan trọng

# Thiết kế lưu trữ dữ liệu

- ❖ Các cơ sở dữ liệu quan hệ (RDBMS): Cho phép quản lý dữ liệu và tìm kiếm thông tin với các câu truy vấn phức tạp
  - Tồn tại nhiều hệ quản trị CSDL thích ứng theo các yêu cầu khác nhau về khối lượng, sự phân tán, và hệ điều hành
  - Đây là cách lưu trữ dữ liệu được dùng phổ biến nhất
- ❖ Các cơ sở dữ liệu hướng đối tượng (OODBMS): Cho phép lưu trữ và quản lý các đối tượng một cách trực tiếp
  - Nhờ vậy mà có thể nói khâu thiết kế cho việc lưu trữ dữ liệu hầu như chẳng còn việc gì phải làm
  - Tuy nhiên các hệ quản trị CSDL đối tượng còn chưa chiếm được thị phần lớn trên thực tế
  - Phần tiếp theo đây, ta đề cập việc thiết kế CSDL với (giả sử) sự chọn lựa một hệ quản trị CSDL quan hệ

# Thiết kế lưu trữ dữ liệu

## 2. Ánh xạ các lớp sang bảng

- ❖ Mỗi lớp thực thể tương ứng với một bảng trong mô hình quan hệ:
  - Mỗi thuộc tính của lớp tương ứng với một cột của bảng,
  - Mỗi cá thể (đối tượng) của lớp tương ứng với một dòng (một bộ-n) của bảng, trong đó OID của đối tượng đóng vai trò là khoá chính (primary key) trong bảng.
- ❖ Có một số thuộc tính của lớp có thể có kiểu phức tạp (các cấu trúc dữ liệu) và không tương ứng được với các kiểu của SQL. Vậy phải có sự biến đổi các kiểu phức tạp đó:
  - Hoặc là bằng nhiều cột, mỗi cột tương ứng với một trường trong cấu trúc dữ liệu.
  - Hoặc là bằng một bảng riêng biệt, liên hệ với bảng chính bằng một khoá ngoài, cho phép kết nối các đối tượng với các giá trị của thuộc tính phức tạp.

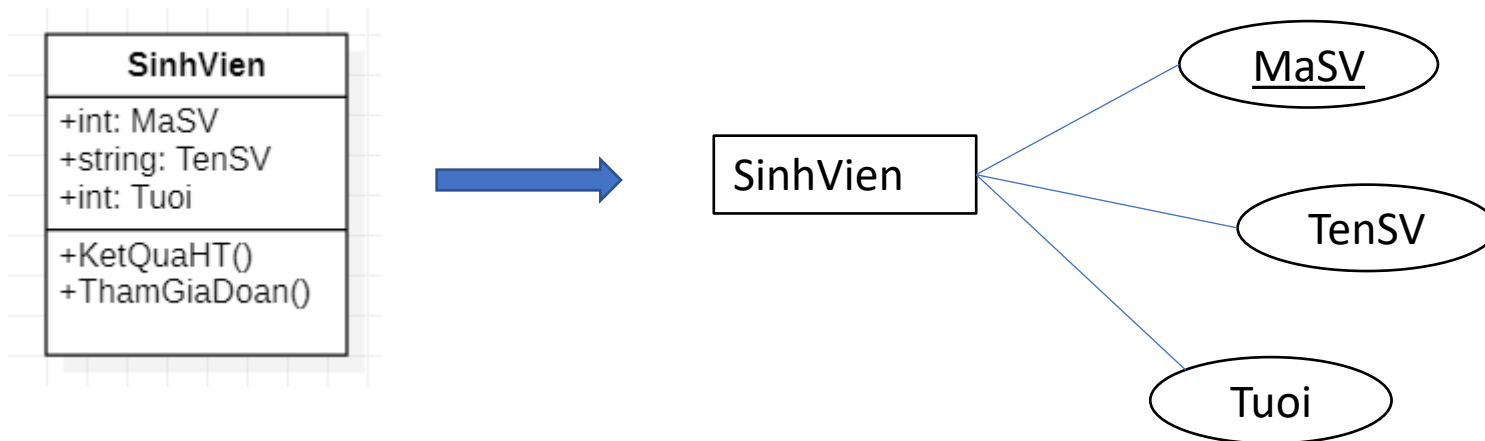
# Thiết kế lưu trữ dữ liệu

## ❖ Sự tương đương giữa lớp và bảng

Mô hình đối tượng	Mô hình quan hệ
Lớp	Bảng
Thuộc tính có kiểu đơn	Cột
Thuộc tính có kiểu phức tạp	Các cột hay khoá ngoài
Đối tượng	Bộ-n
OID	Khoá chính
Liên kết và Kết nối	Khoá ngoài hay bảng
Thừa kế	Khoá chính đồng nhất trên nhiều bảng

# Thiết kế lưu trữ dữ liệu

- Chuyển đổi một lớp thành một kiểu thực thể: Với mỗi lớp trong biểu đồ lớp, ta tạo ra một kiểu thực thể tương ứng. Tên của kiểu thực thể được lấy trực tiếp từ tên lớp. Các thuộc tính của lớp gốc được chuyển thành các thuộc tính của kiểu (tập) thực thể. Thuộc tính định danh OID đóng vai trò là thuộc tính khoá.



# Thiết kế lưu trữ dữ liệu

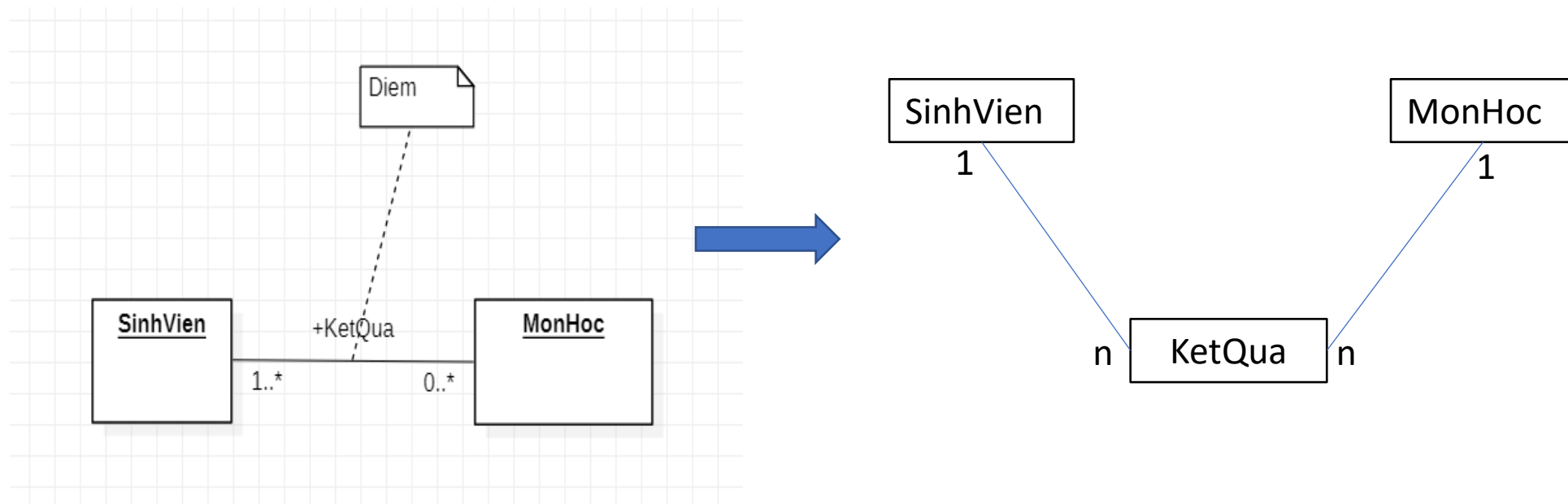
- ❖ Ánh xạ từ lớp sang bảng lại không nên hiểu máy móc là ánh xạ 1-1, vì ánh xạ 1-1 thường dẫn tới các hiện tượng không tốt như sau:
  - Quá nhiều bảng và quá nhiều kết nối phải thực hiện: Thông thường thì biểu đồ lớp có thể chứa rất nhiều lớp, nếu cứ ánh xạ 1-1 thành các bảng, thì ta sẽ có rất nhiều bảng. Mà khi đã có nhiều bảng, thì để tìm kiếm thông tin trong CSDL, ta lại phải áp dụng nhiều phép kết nối (join), ảnh hưởng nặng nề tới tốc độ truy cập. Vì vậy nên cố gắng gom các bảng thường đi liền với nhau trong tìm kiếm thành các bảng lớn.



# Thiết kế lưu trữ dữ liệu

- Thiếu các bảng: Trong thiết kế các liên kết giữa các lớp, nếu giữa hai lớp có một liên kết nhiều-nhiều mà liên kết đó lại có thuộc tính, thì phải định nghĩa một lớp liên kết chứa thuộc tính đó. Chẳng hạn giữa Sinh viên và Môn học có một liên kết nhiều-nhiều ("Sinh viên hoàn thành môn học"); liên kết này lại có thuộc tính (như là năm học, kết quả thi), do đó trong mô hình đối tượng, phải thêm một lớp liên kết là lớp Kết quả học. Chuyển sang mô hình quan hệ, thì ngoài hai bảng Sinh viên và Môn học, ta cũng có bảng Kết quả học.

# Thiết kế lưu trữ dữ liệu



# Thiết kế lưu trữ dữ liệu

- Mất khả năng thừa kế: Mô hình quan hệ không sẵn sàng hỗ trợ cho sự thừa kế. Vì vậy một ánh xạ đơn giản các lớp sang các bảng, mà không có các xử lý đặc biệt (xem ánh xạ liên kết) thì không còn giữ được mối liên quan khái quát hoá và sự thừa kế.
- Suy giảm hiệu năng: Một ánh xạ 1-1 thường chuyển biểu đồ lớp thành mô hình quan hệ ở dạng chuẩn 3. Tuy nhiên có nhiều ứng dụng hướng tới một số báo cáo nhất định (reporting functionality), đòi hỏi phải hạ chuẩn đối với dữ liệu (chẳng hạn phải lặp lại cùng một dữ liệu ở nhiều bảng, hay sáp nhập một số bảng) nhằm tạo khả năng truy cập thuận lợi nhất cho những mục đích định sẵn (vd: tạo báo cáo thống kê) của ứng dụng đó.

# Thiết kế lưu trữ dữ liệu

## 3. Ánh xạ các liên kết

- ❖ Trong phần trên, ta đã đề cập việc thiết kế chi tiết các liên kết, vốn không có sẵn trong các ngôn ngữ lập trình hướng đối tượng. Nay ta ánh xạ các liên kết giữa các lớp trường cứu vào các bảng trong mô hình quan hệ, ta cũng có điều tương tự, nghĩa là phải chuyển các liên kết thành các khoá ngoài hay bảng liên kết. Sự khác biệt chỉ là ở chỗ các thuộc tính trong các bảng của mô hình quan hệ đều phải là thuộc tính đơn.
  - Với liên kết một-một: Lập một bảng cho mỗi lớp (bảng A, B). Khoá chính của mỗi bảng cũng là khoá ngoài đối với bảng kia. Thực ra lập thành hai bảng là chỉ thực sự cần thiết khi liên kết là một ánh xạ vào (1 đến 0..1), còn khi liên kết là một đối một, thì cách tốt nhất là chỉ lập một bảng.

# Thiết kế lưu trữ dữ liệu

- Với liên kết một-nhiều: Lập một bảng cho từng lớp (bảng A, B). Khoá chính của bảng A (đầu Một) là khoá ngoài trong bảng B (đầu Nhiều).
- Với liên kết nhiều-nhiều: Lập một bảng cho từng lớp (bảng A, B). Lập thêm một bảng kết nối, hay còn gọi là bảng giao (bảng C). Khoá chính của mỗi bảng A, B được định nghĩa là khoá ngoài trong bảng C. Khoá chính của C có thể là một cột riêng song cũng có thể là khoá bội hợp thành từ hai khoá ngoài.

# Thiết kế lưu trữ dữ liệu

- Với kết tập: Trong mô hình quan hệ thì kết tập cũng chỉ được mô hình hoá như một liên kết bình thường. Tạo thành hai bảng, vì các đối tượng bị kết tập vẫn có thể tồn tại một cách độc lập.
- Với các liên kết đệ quy: Đó là loại liên kết giữa một lớp với chính nó, cho ta các kết nối giữa các cặp đối tượng của lớp. Liên kết đệ quy được cài đặt bằng cách thêm vào một cột, mà giá trị là các giá trị khoá chính của lớp đó, xem như là một khoá ngoài.

# Thiết kế lưu trữ dữ liệu

## 4. Ánh xạ mối liên quan khái quát hoá

Hầu hết các ngôn ngữ lập trình hướng đối tượng hỗ trợ cho mối liên quan này giữa các lớp (sự thừa kế), thì mô hình quan hệ không hỗ trợ cho nó. Có nhiều cách để chuyển đổi mối liên quan này vào mô hình quan hệ:

- ❖ Lập một bảng cho mỗi lớp (lớp trên cũng như lớp dưới). Mối liên quan giữa các bảng được giải quyết theo hai cách:
  - Dùng một OID chung cho mọi bảng trong cùng một phả hệ thừa kế. Như vậy để truy cập dữ liệu của một đối tượng phải thực hiện phép kết nối (join) giữa lớp trên và lớp dưới (cha và một con).
  - Lập một khung nhìn SQL (view) cho mỗi cặp lớp trên/lớp dưới. Như vậy số bảng tăng thêm, và thực chất cũng phải dùng phép kết nối khi truy cập dữ liệu.
- Cả hai cách làm này đều cho phép tăng thêm các lớp con trong tương lai, mà không làm xáo trộn các lớp cũ.

# Thiết kế lưu trữ dữ liệu

- ❖ Chỉ lập một bảng (ứng với lớp trên), mọi thuộc tính ở các lớp dưới đều dồn về bảng trên. Như vậy số bảng là tối thiểu (=1), nhưng số cột tăng thêm, và với mỗi cá thể, có những cột là không dùng tới. Vậy chỉ thích hợp khi số các thuộc tính trong các lớp dưới là ít. Không thuận lợi cho việc thêm mới các lớp dưới.
- ❖ Lập một bảng cho mỗi lớp dưới, lặp lại các thuộc tính của lớp trên vào bảng của mỗi lớp dưới (hạ chuẩn). Như vậy số bảng rút bớt (không có bảng cho lớp cha), và việc tăng thêm lớp con trong tương lai không gây ảnh hưởng, nhưng việc điều chỉnh lớp cha buộc phải điều chỉnh lại các bảng. Thích hợp khi số các thuộc tính trong lớp cha là ít.