

# Phân tích thiết kế Hệ thống

Giảng viên: Nguyễn Bá Ngọc

Hà Nội-2022

# Thiết kế (phần 2)

# Nội dung

- Các nguyên lý thiết kế (SOLID)
- Thiết kế lưu trữ cố định
- Thiết kế tương tác người-máy

# Nội dung

- Các nguyên lý thiết kế (SOLID)
- Thiết kế lưu trữ cố định
- Thiết kế tương tác người-máy

# Nguyên lý trách nhiệm duy nhất

- Nguyên lý trách nhiệm duy nhất / Single Responsibility Principle (SRP)

*Mỗi lớp chỉ nên có 1 lý do duy nhất để thay đổi /  
A class should have only one reason to change*  
[Robert .C Martin]

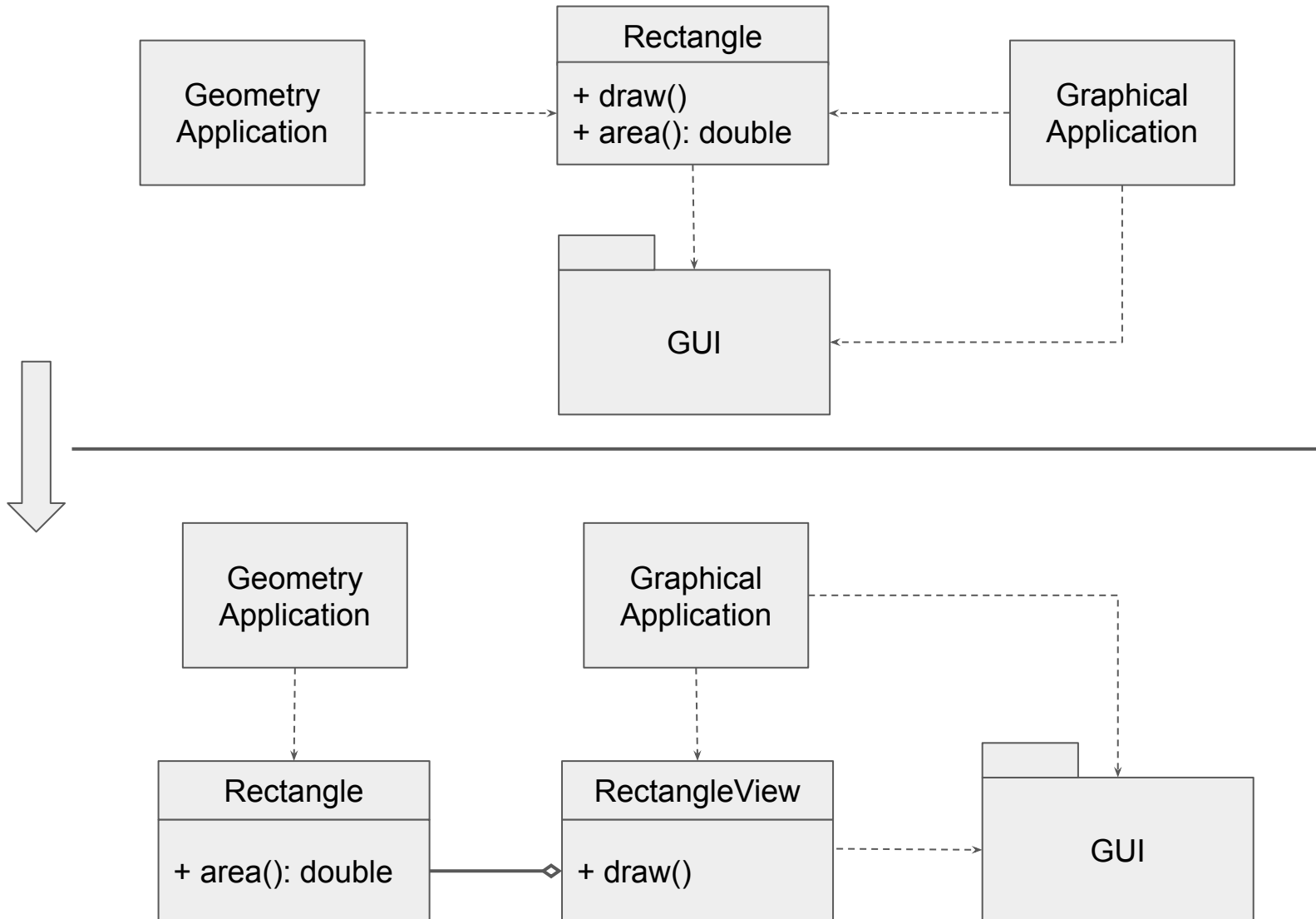
- Trách nhiệm được gán cho lớp
  - (xem thẻ CRC)
- Nếu phát sinh nhu cầu triển khai chức năng mới, hoặc chức năng đang có thay đổi, thì các trách nhiệm của các lớp phải thay đổi.
- Lớp có 1 trách nhiệm sẽ chỉ có 1 lý do để thay đổi.

*Mỗi thành phần chỉ nên có đảm nhận 1 trách nhiệm*

# Trách nhiệm và tính gắn kết

- Lớp có tính thống nhất ở mức cao chỉ triển khai 1 trách nhiệm (hoặc ít trách nhiệm) / Lớp chỉ triển khai 1 trách nhiệm có tính thống nhất ở mức cao.
- Lớp có tính thống nhất ở mức thấp thường vi phạm nguyên lý SRP / Lớp vi phạm SRP thường có tính thống nhất ở mức thấp.
- Ưu điểm: Các lớp 1 trách nhiệm thường nhỏ, dễ tái sử dụng, và dễ hiểu hơn, và ít thay đổi.
- Lưu ý: Phân tách trách nhiệm có thể phức tạp và có hiệu ứng phụ, áp dụng SRP khi thực sự cần thiết, thực sự có thay đổi thường xuyên.

# Ví dụ SRP



# Nguyên lý Mở-Đóng

- Nguyên lý Mở-Đóng / Open-Close Principle (OCP)

*Các thành phần phần mềm cần có khả năng mở rộng mà không thay đổi /  
Software entities should be open for extension, but closed for modifications. [Bertrand Meyer]*

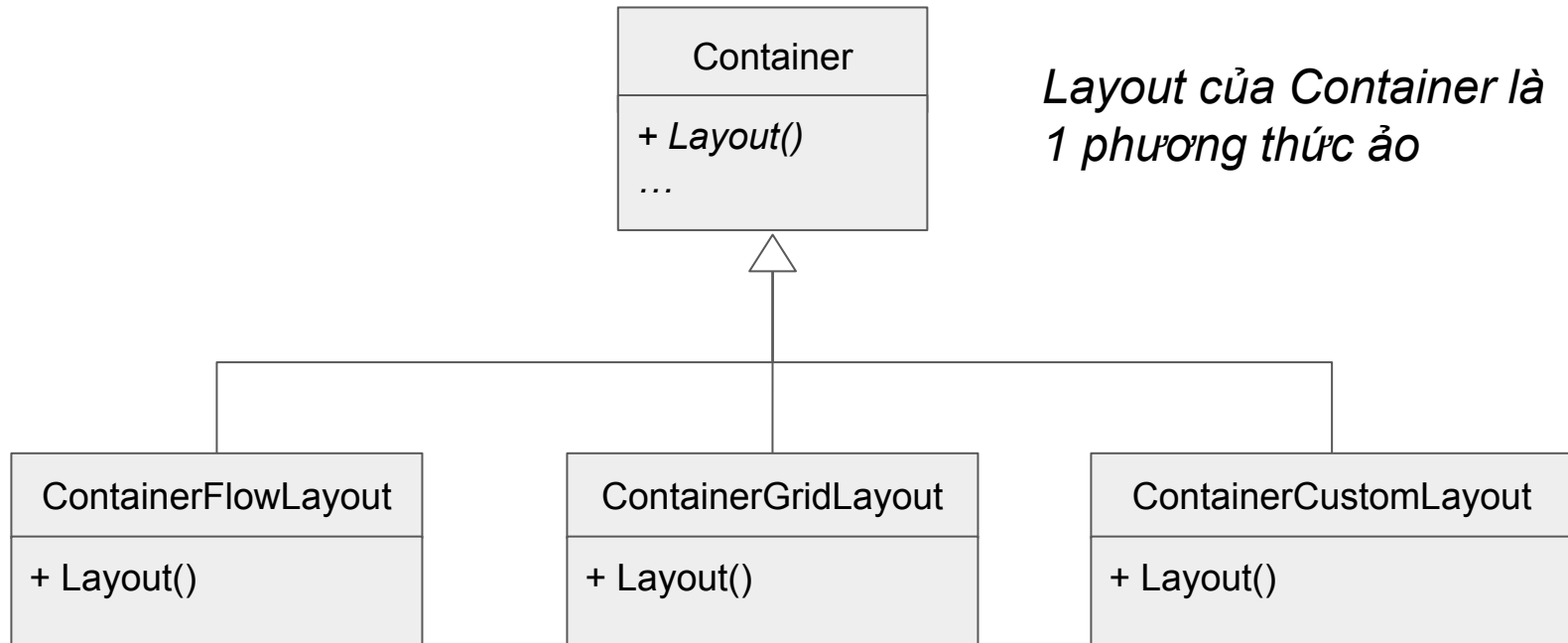
- Mở rộng hành vi của 1 thành phần:
  - Khi yêu cầu ứng dụng thay đổi, chúng ta có thể mở rộng thành phần đã có, thêm vào các hành vi mới.
- Không thay đổi mã nguồn đã có:
  - Mã nguồn đã có được giữ nguyên trong quá trình bổ xung hành vi mới.
  - Đem lại nhiều lợi ích trong quá trình phát triển.



# Trừu tượng hóa là chìa khóa

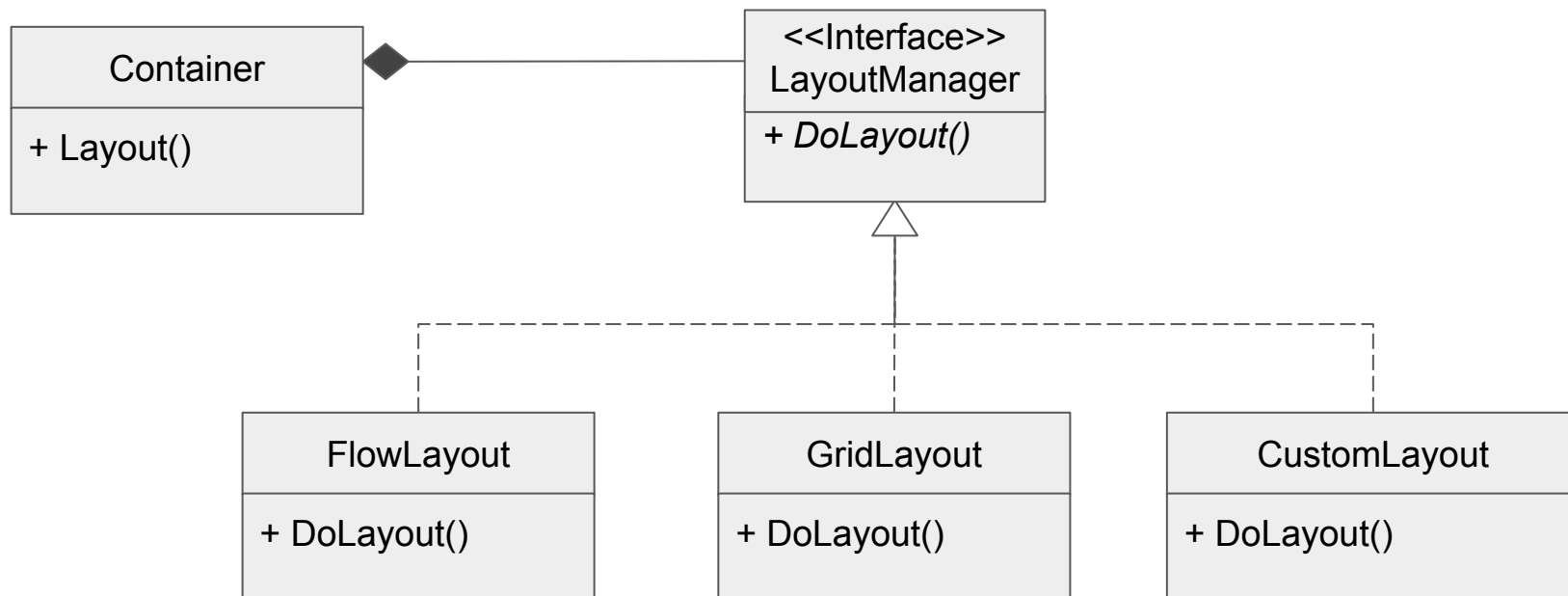
- Để mở rộng một thành phần mà không thay đổi nó, thì phần thay đổi trong triển khai của thành phần đó phải được trừu tượng hóa.
- Lập trình hướng đối tượng: Lớp ảo, giao diện, phương thức ảo, kế thừa, v.v..
- Lập trình hàm: Kiểu dữ liệu khái quát, hàm khái quát, v.v..
- *(Khả năng tránh thay đổi có tính tương đối: Rất khó tuyệt đối tránh thay đổi mã nguồn đã có / Thường vẫn có thể tìm những yêu cầu thay đổi dẫn đến thay đổi mã nguồn).*

# Khái quát hóa dựa trên kế thừa



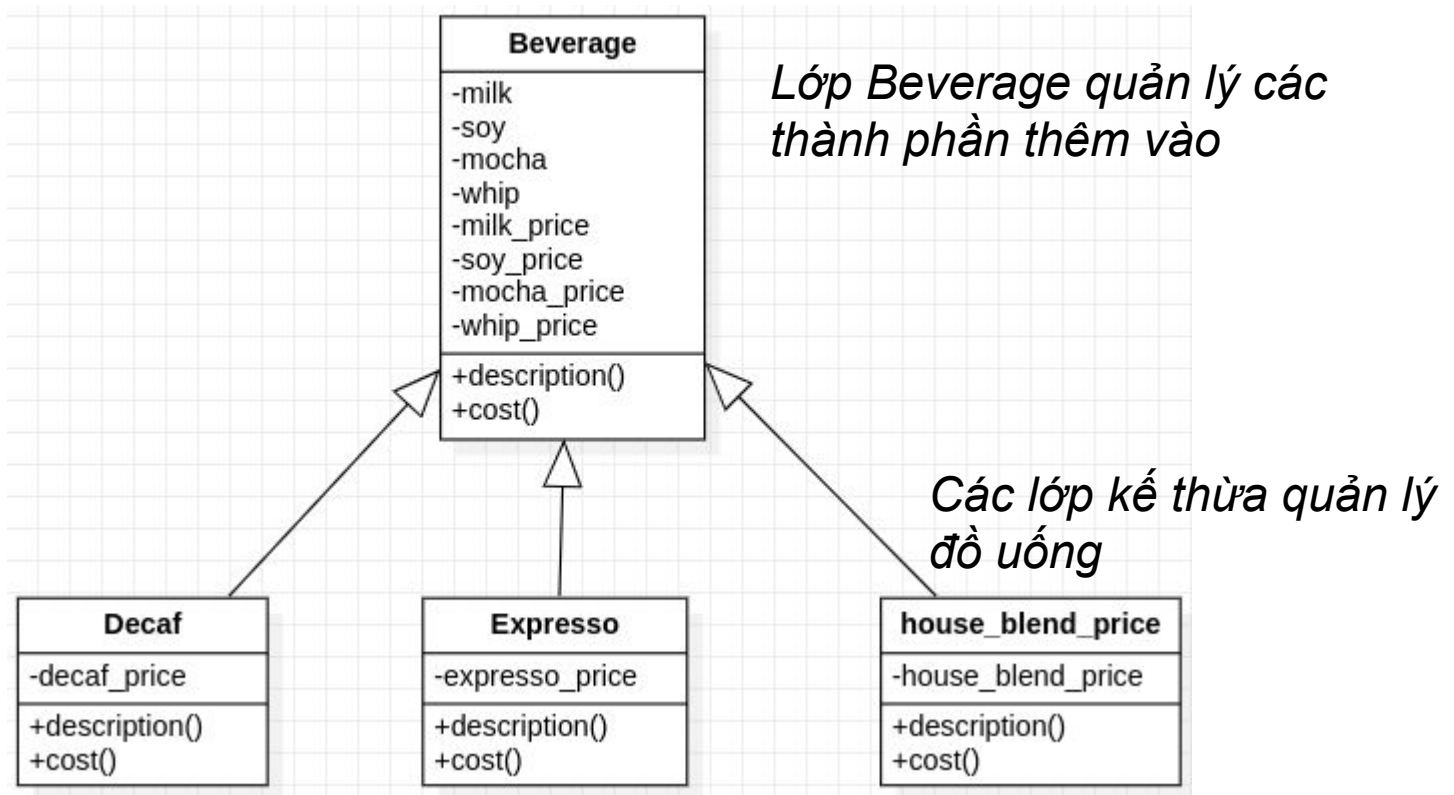
Các lớp con sẽ cung cấp triển khai cụ thể của phương thức Layout

# Khái quát hóa dựa trên tổng hợp



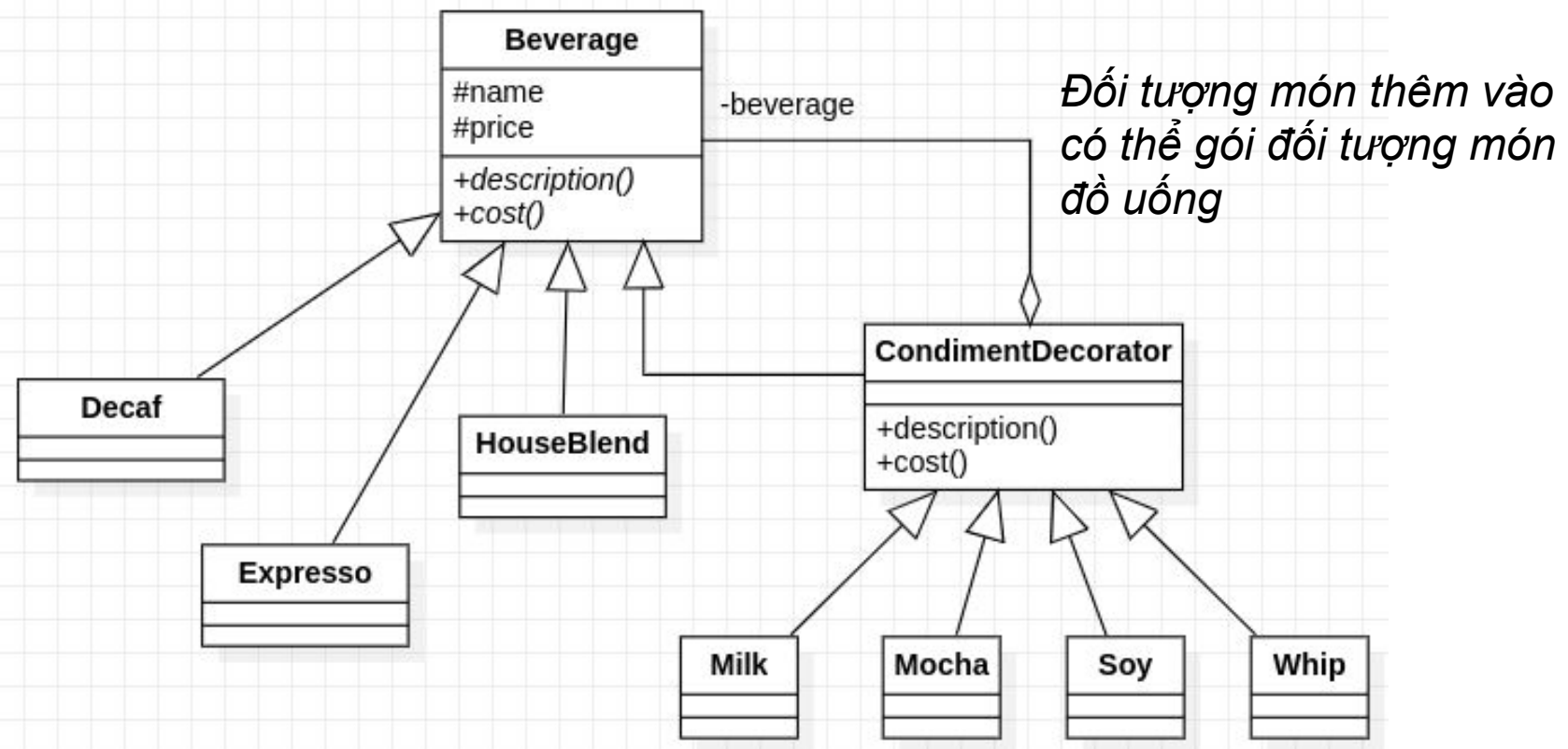
*Chúng ta có thể thay đổi hành vi của Container bằng cách thiết lập 1 đối tượng quản lý bố cục phù hợp (kế thừa LayoutManager)*

# Ví dụ OCP



Thiết kế này vi phạm OCP, để thay đổi giá của thành phần thêm vào hoặc bổ sung món mới chúng ta phải sửa lớp Beverage.

# Ví dụ OCP<sub>(2)</sub>



(Mẫu Tô Điểm / Decorator)

- Để biểu diễn món thêm vào chúng ta có thể gói đối tượng thuộc loại Beverage trong đối tượng thuộc loại CondimentDecorator, để bổ sung món thêm vào chúng ta có thể tạo lớp kế thừa CondimentDecorator.
- Thiết kế này đáp ứng được OCP.

# Nguyên lý khả thay Liskov

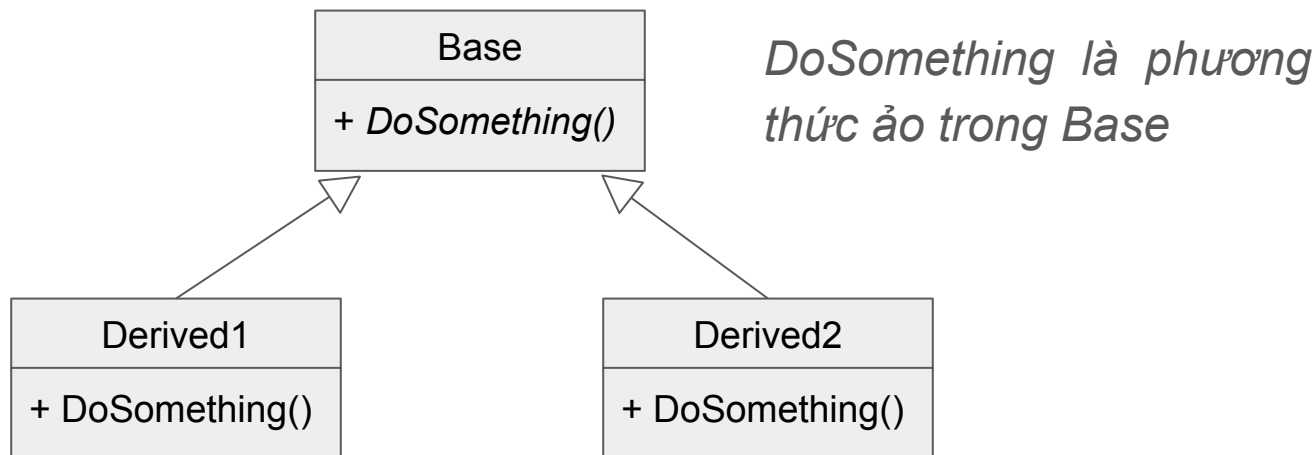
- Nguyên lý khả thay Liskov / Liskov Substitution Principle (LSP)

*Các lớp con phải có khả năng thay thế các lớp cơ sở về mặt hành vi/*

*Subtypes must be behaviorally substitutable for their base types. [Barbara Liskov, 1988]*

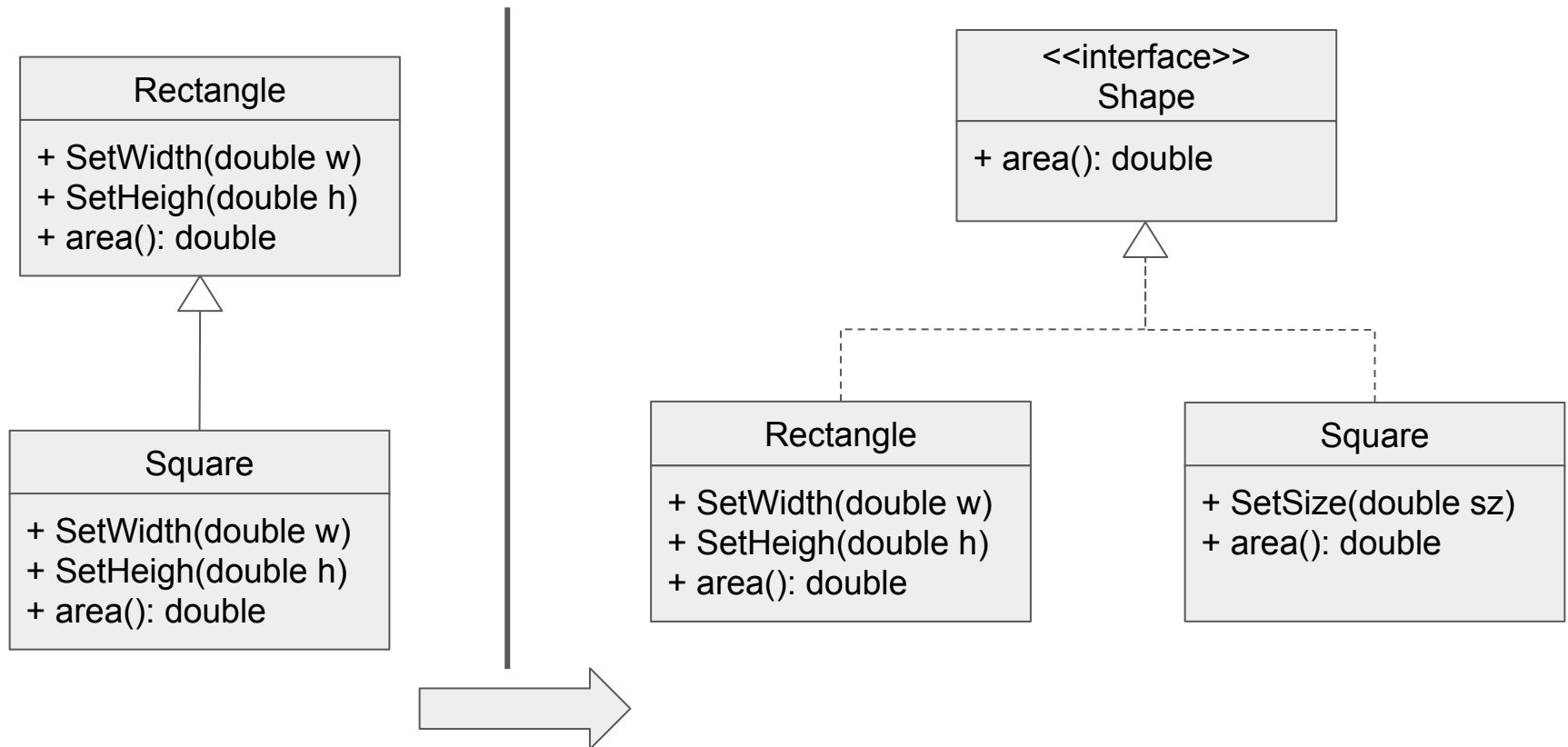
- Giống như dấu hiệu chất lượng thiết kế quan hệ kế thừa
- Hỗ trợ hạn chế các vấn đề có thể phát sinh khi lập trình với các giao diện khái quát.

# Khả thay hành vi



- Ngoài triển khai (định nghĩa lại) phương thức `DoSomething()` trong **Derived1** và **Derived2**, LSP còn ràng buộc về hành vi đối với các đối tượng thuộc các lớp **Derived1** và **Derived2**:
  - Có thể diễn đạt một cách gần đúng là bảo toàn ý nghĩa của lớp cơ sở.

# Ví dụ LSP

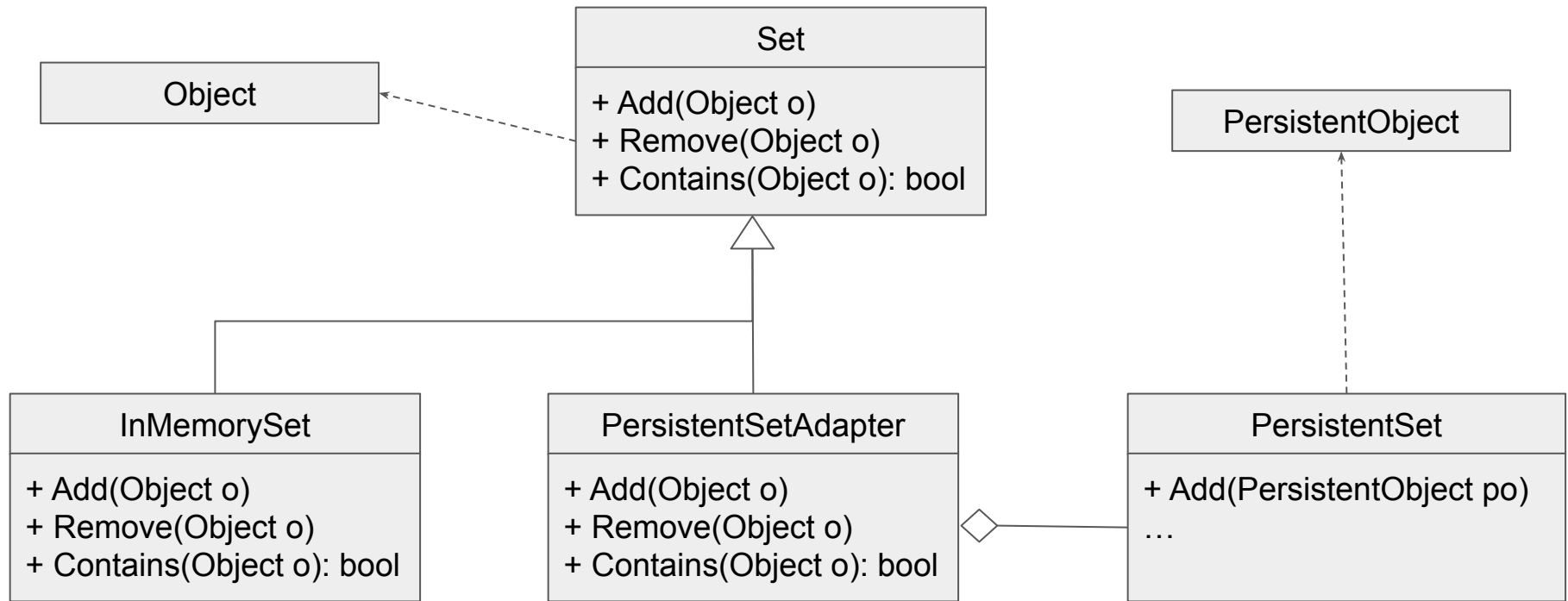


*Không tương thích LSP,  
SetHeigh và SetWidth có ý  
nghĩa khác nhau trong  
Rectangle và Square.*

*Đáp ứng được LSP*



# Ví dụ LSP<sub>(2)</sub>



*Đánh giá thiết kế và đề xuất phương án nếu cần?*

# Nguyên lý phân tách giao diện

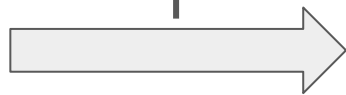
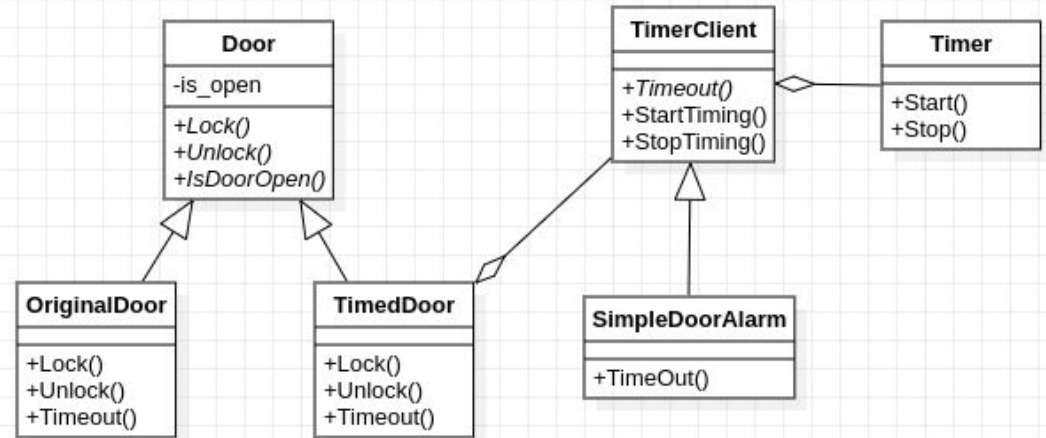
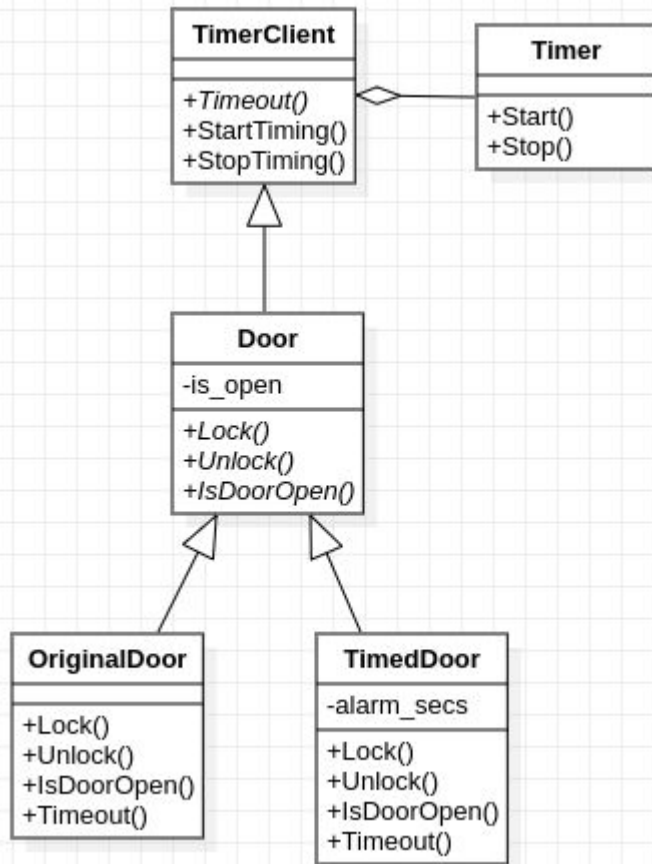
- Nguyên lý phân tách giao diện / Interface Segregation Principle (ISP)

*Không ép phía khách phụ thuộc vào các phương thức không được sử dụng trong 1 giao diện /  
Clients should not be forced to depend on methods that they do not use*

[Robert C. Martin]

- Các giao diện chứa các phương thức không liên quan ép buộc các thành phần sử dụng giao diện đó chịu các thay đổi đáng ra không ảnh hưởng tới chúng.
- Các giao diện bị làm rối cần được phân tách
- Tuy nhiên cần thận trọng tránh làm nát vụn các giao diện

# Ví dụ ISP



# Nguyên lý đảo ngược phụ thuộc

- Nguyên lý đảo ngược phụ thuộc / Dependency Inversion Principle (DIP)

*Các mô-đun bậc cao không nên phụ thuộc vào các mô-đun bậc thấp. Cả 2 nên phụ thuộc vào các cơ chế trừu tượng.*

*Các thành phần trừu tượng không nên phụ thuộc vào các thành phần cụ thể. Các thành phần cụ thể nên phụ thuộc vào các thành phần trừu tượng. /*

*High-level modules should not depend on low-level modules. Both should depend on abstractions.*

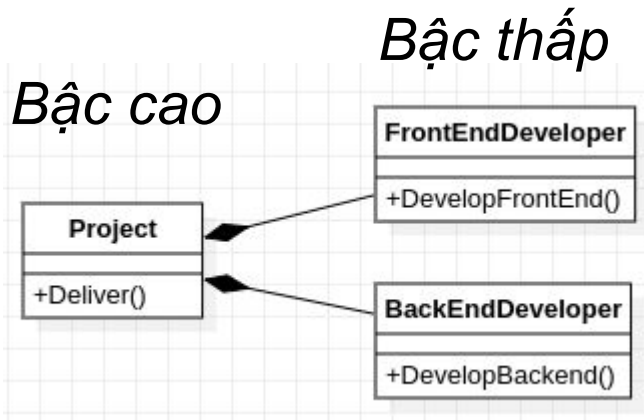
*Abstractions should not depend upon details. Details should depend upon abstractions.*

[Robert C. Martin]

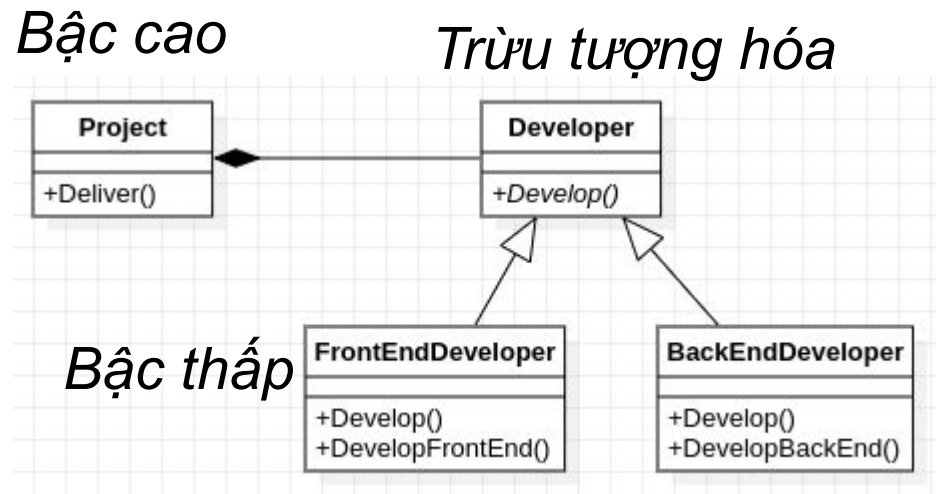
# Nguyên lý đảo ngược phụ thuộc<sub>(2)</sub>

- Phần mềm với thiết kế tốt thường được chia nhỏ thành các mô-đun
  - Mô-đun bậc thấp cung cấp các dịch vụ để triển khai mô-đun bậc cao.
- Phụ thuộc vào các cơ chế trừu tượng giúp nâng cao khả năng tái sử dụng các thành phần.
- Áp dụng DIP giúp nới lỏng các phụ thuộc.

# Ví dụ DIP



Thiết kế này vi phạm DIP với **Project** phụ thuộc trực tiếp vào **FrontEndDeveloper** và **BackEndDeveloper**.



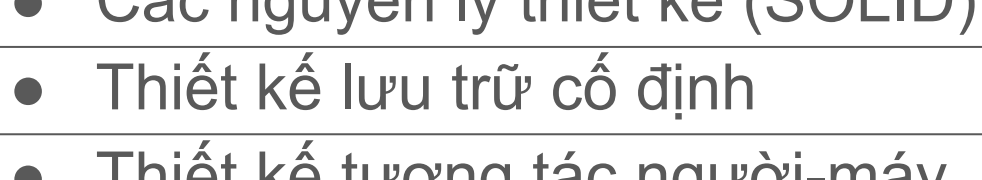
Triển khai cụ thể phụ thuộc vào cơ chế trừu tượng.

Thiết kế này đáp ứng được DIP, chiều phụ thuộc vào mô-đun bậc thấp đã được đảo ngược: Các mô-đun bậc thấp phụ thuộc vào giao diện khái quát được yêu cầu bởi mô-đun bậc cao.

# Chèn phụ thuộc

- Chèn phụ thuộc / Dependency Injection (DI)
- Có thể được sử dụng để nghịch đảo phụ thuộc / Phương pháp nghịch đảo phụ thuộc cụ thể.
- Đối tượng thích hợp được tạo và đưa vào từ bên ngoài.
- Đối tượng phụ thuộc có thể được đưa vào thông qua:
  - Tham số cho hàm tạo (được sử dụng cho những phụ thuộc bắt buộc)
    - Như trong mẫu thiết kế Tô điểm / Decorator
  - Tham số cho phương thức thiết lập phụ thuộc (được sử dụng cho các phụ thuộc không bắt buộc)
    - Như trong mẫu thiết kế Trạng thái / State

# Nội dung

- Các nguyên lý thiết kế (SOLID)
  - Thiết kế lưu trữ cố định
  - Thiết kế tương tác người-máy
- 



# Tổng quan về thiết kế lưu trữ cố định

- Các CSDL (DB) và các Hệ quản trị CSDL (DBMS) là các thành phần quan trọng của 1 hệ thống thông tin.
- Thiết kế CSDL biến đổi biểu đồ lớp của mô hình lĩnh vực thành 1 mô hình CSDL chi tiết cho hệ thống
- Hệ quản trị CSDL được sử dụng để triển khai và tương tác với CSDL

# Các định dạng lưu trữ cố định

- Tự phát triển bắt đầu từ khả năng đọc/ghi tệp cơ bản nhất
- CSDL quan hệ
- CSDL đối tượng-quan hệ
- CSDL hướng đối tượng
- Các định dạng lưu trữ khác
  - Lưu trữ khóa-giá trị / Key-Value data stores
  - Lưu trữ tài liệu / Document data stores
  - CSDL hướng cột / Columnar database
  - CSDL đồ thị / Graph database
  - V.V..

# CSDL quan hệ

- Lựa chọn lưu trữ dữ liệu phổ biến nhất
- CSDL bao gồm 1 tập bảng
  - Trong mỗi bảng khóa chính xác định 1 dòng duy nhất
  - Mỗi quan hệ giữa các bảng được thiết lập bằng khóa ngoại
    - Cơ chế cơ bản để đảm bảo toàn vẹn dữ liệu giữa các bảng, ví dụ: bạn không thể tạo đơn hàng cho 1 khách hàng không tồn tại.
- Ngôn ngữ truy vấn có cấu trúc / Structured Query Language (SQL) được sử dụng để truy cập dữ liệu
  - Quy chuẩn được hỗ trợ bởi nhiều hệ quản trị CSDL
  - Cho phép thao tác với từng bảng và kết hợp nhiều bảng để lấy dữ liệu cần thiết

# CSDL Quan hệ-Đối tượng

- CSDL quan hệ với khả năng lưu trữ đối tượng
- Đối tượng được lưu trữ bằng các kiểu dữ liệu do người dùng tự định nghĩa
  - SQL được mở rộng để xử lý các kiểu dữ liệu phức tạp
  - Hỗ trợ kế thừa ở mức hạn chế

# CSDL hướng đối tượng

- Triển khai các khái niệm hướng đối tượng
- Có thể đặc tả cấu trúc và hành vi của đối tượng
  - Tương thích tốt hơn với các ngôn ngữ lập trình hướng đối tượng
- Sử dụng ngôn ngữ tương tự / được phát triển từ SQL
- Hỗ trợ kế thừa ở mức độ nhất định
- Chiếm thị phần nhỏ trong ứng dụng thực tế

# Các yêu cầu phi chức năng

- Các yêu cầu phi chức năng có ảnh hưởng đến thiết kế lưu trữ cố định
- Yêu cầu vận hành: Ảnh hưởng bởi lựa chọn phần cứng và hệ điều hành
- Yêu cầu hiệu năng: Các vấn đề tốc độ và dung lượng
- Các yêu cầu bảo mật: Quản lý truy cập, mã hóa, và sao lưu
- V.V..

# Ánh xạ đối tượng lĩnh vực ứng dụng

- Với mô hình dữ liệu Hướng đối tượng
  - Mỗi lớp cụ thể có thể có 1 lớp tương ứng trong lưu trữ
- Với mô hình dữ liệu Quan hệ-đối tượng
  - Phụ thuộc vào các tính năng đối tượng được hỗ trợ
- Với mô hình dữ liệu quan hệ
  - Được biểu diễn bằng các bảng

# Ánh xạ tới CSDL Quan hệ-Đối tượng

*Mỗi đối tượng trong ORDBMS có 1 ID duy nhất*

**R1.** Ánh xạ tất cả lớp lĩnh vực cụ thể và lớp lĩnh vực ảo có nhiều lớp con trực tiếp tới bảng ORDBMS.

**R2.** Ánh xạ thuộc tính đơn trị tới cột của bảng.

**R3.** Ánh xạ phương thức và thuộc tính suy diễn tới thủ tục lưu trữ (stored procedure) hoặc mô-đun chương trình.

**R4.** Ánh xạ quan hệ tổng hợp và quan hệ liên kết đơn trị (0..1 hoặc 1..1) tới cột có thể lưu ID đối tượng. Áp dụng cho cả 2 đầu quan hệ.

**R5.** Ánh xạ các thuộc tính đa trị tới cột có thể lưu 1 tập ID của đối tượng.

**R6.** Ánh xạ nhóm thuộc tính lặp sang 1 bảng mới và tạo liên kết 1-nhiều từ bảng gốc sang bảng mới / thêm khóa ngoại vào bảng mới.

**R7.** Ánh xạ các quan hệ tổng hợp và liên kết đa trị (0..\* hoặc 1..\*) tới cột có thể lưu 1 tập ID đối tượng. Thực hiện cho cả 2 đầu quan hệ.



# Ánh xạ tới CSDL quan hệ-đối tượng<sub>(2)</sub>

**R8.** Đối với các quan hệ tổng hợp và liên kết hỗn hợp (một-nhiều), ở phía đơn trị (1..1) hoặc (0..1) của mỗi quan hệ, thêm 1 cột để lưu 1 tập ID của các đối tượng ở phía đa trị. Ở phía đa trị (1..\* hoặc 0..\*), thêm 1 cột để lưu ID của đối tượng thuộc đầu đơn trị.

**R9a.** Thêm cột vào bảng tương ứng với lớp con để chứa ID của đối tượng được lưu trong bảng tương ứng với lớp cha. Cơ số của liên kết mới từ lớp con tới lớp cha phải là 1..1 / 1 đối tượng lớp con xác định đúng 1 đối tượng lớp cha. Thêm cột vào bảng biểu diễn lớp cha để lưu ID đối tượng của lớp con. Nếu lớp cha là cụ thể, thì cơ số từ lớp cha tới lớp con là 0..1 / 1 đối tượng lớp cha *có thể* xác định 1 đối tượng lớp con, nếu ngược lại thì cơ số là 1..1 / xác định đúng 1 đối tượng lớp con. Ràng buộc XOR (OR loại trừ) cần được bổ xung cho các liên kết này. Thực hiện với tất cả các lớp cha / mô phỏng kế thừa. Hoặc

**R9b.** Mở rộng lớp con bằng cách sao chép các thuộc tính của lớp cha xuống các lớp con và loại lớp cha khỏi thiết kế. Hoặc

**R9c.** Làm phẳng cây kế thừa bằng cách sao chép tất cả các thuộc tính trong cây vào lớp gốc, tạo 1 bảng cho cây kế thừa.

# Ánh xạ tới CSDL quan hệ

**R1.** Ánh xạ lớp lĩnh vực cụ thể và lớp lĩnh vực trừu tượng có nhiều lớp con trực tiếp tới bảng RDBMS.

**R2.** Ánh xạ thuộc tính đơn trị tới cột của bảng.

**R3.** Ánh xạ phương thức và thuộc tính suy diễn tới thủ tục lưu trữ hoặc mô-đun chương trình.

**R4.** Ánh xạ các quan hệ tổng hợp và liên kết đơn trị tới 1 cột có thể lưu khóa của bảng liên quan / thêm 1 khóa ngoại vào bảng. Thực hiện cho cả 2 đầu liên kết.

**R5.** Ánh xạ các thuộc tính đa trị và các nhóm lặp tới các bảng mới và tạo quan hệ 1-nhiều từ bảng gốc tới bảng mới / thêm khóa ngoại vào bảng mới.

**R6.** Ánh xạ các quan hệ tổng hợp và liên kết đa trị tới 1 bảng liên kết mới có vai trò kết nối 2 bảng ban đầu. Sao chép khóa chính từ các bảng gốc vào bảng liên kết / thêm các khóa ngoại vào bảng liên kết.

# Ảnh xạ tới CSDL quan hệ<sub>(2)</sub>

**R7.** Đối với các quan hệ tổng hợp và liên kết hỗn hợp, sao chép khóa chính từ phía đơn trị (1..1 hoặc 0..1) của mỗi quan hệ vào 1 cột mới trong bảng của phía đa trị (1..\* hoặc 0..\*) để lưu khóa của bảng liên quan / bổ xung 1 khóa ngoại vào bảng ở phía đa trị của mỗi quan hệ.

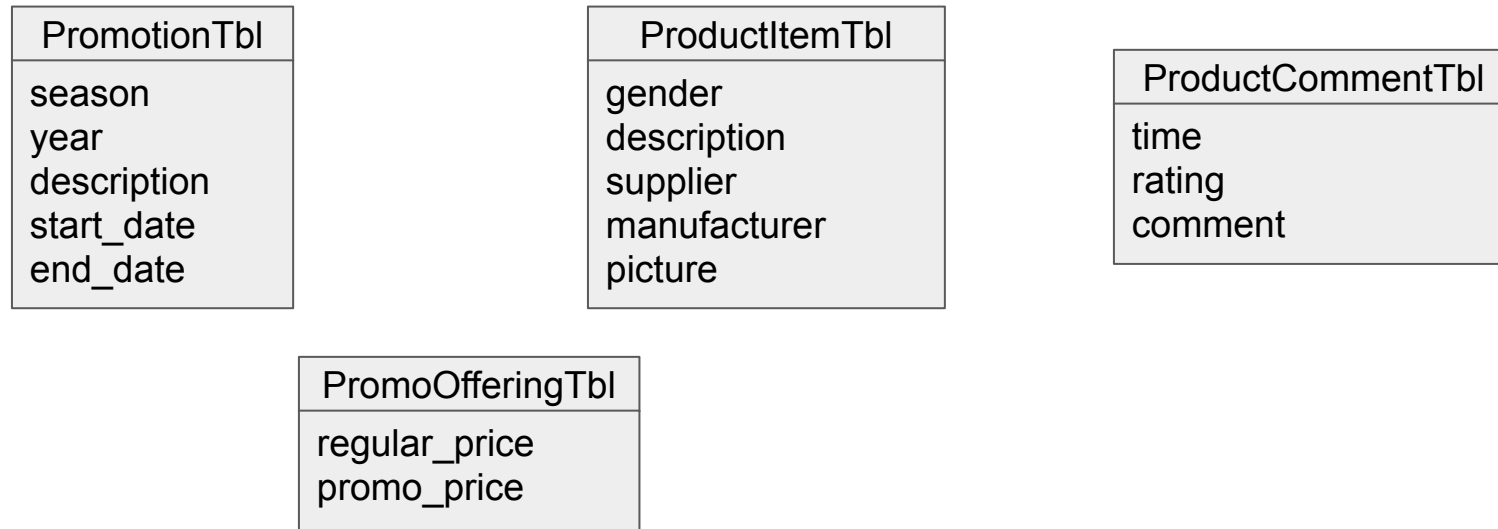
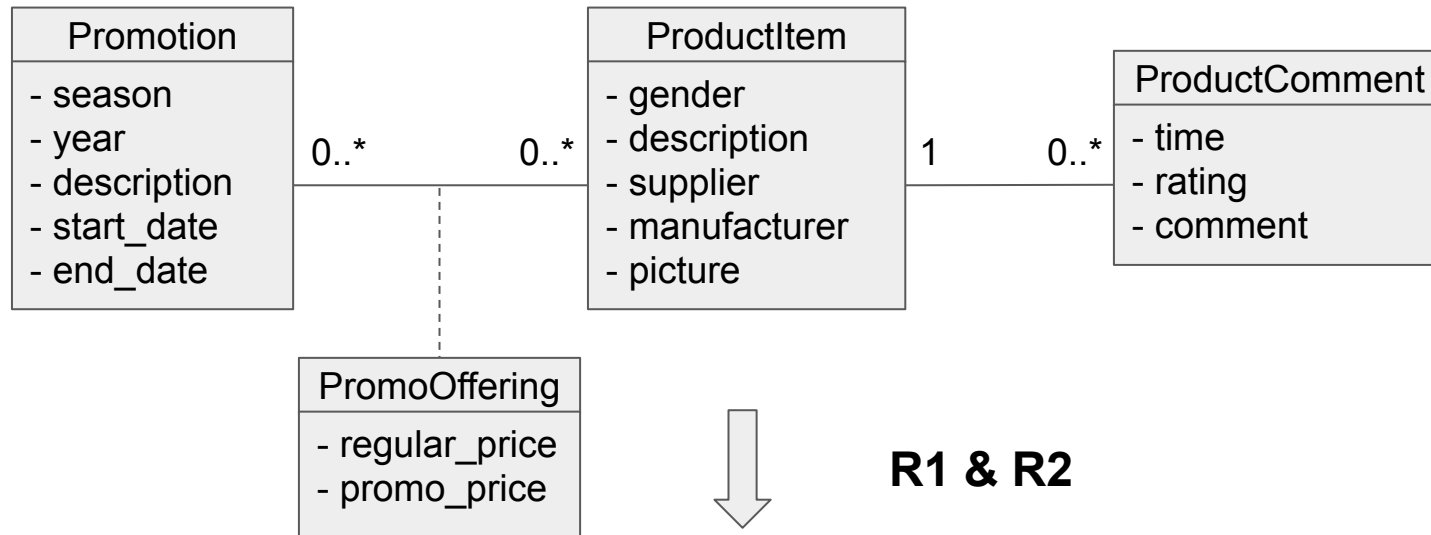
Đối với quan hệ Khái quát hóa/Kế thừa:

**R8a.** Đảm bảo khóa chính của đối tượng thuộc lớp con giống với khóa chính của đối tượng thuộc lớp cha. Cơ sở của liên kết mới này từ lớp con tới lớp cha phải là 1..1. Nếu lớp cha là lớp cụ thể thì cơ sở từ lớp cha tới lớp con là 0..1, nếu ngược lại thì cơ sở là 1..1. Ngoài ra ràng buộc XOR (OR loại trừ) phải được thêm vào giữa các liên kết. Áp dụng quy tắc này cho tất cả các lớp cha / Mô phỏng kế thừa. Hoặc

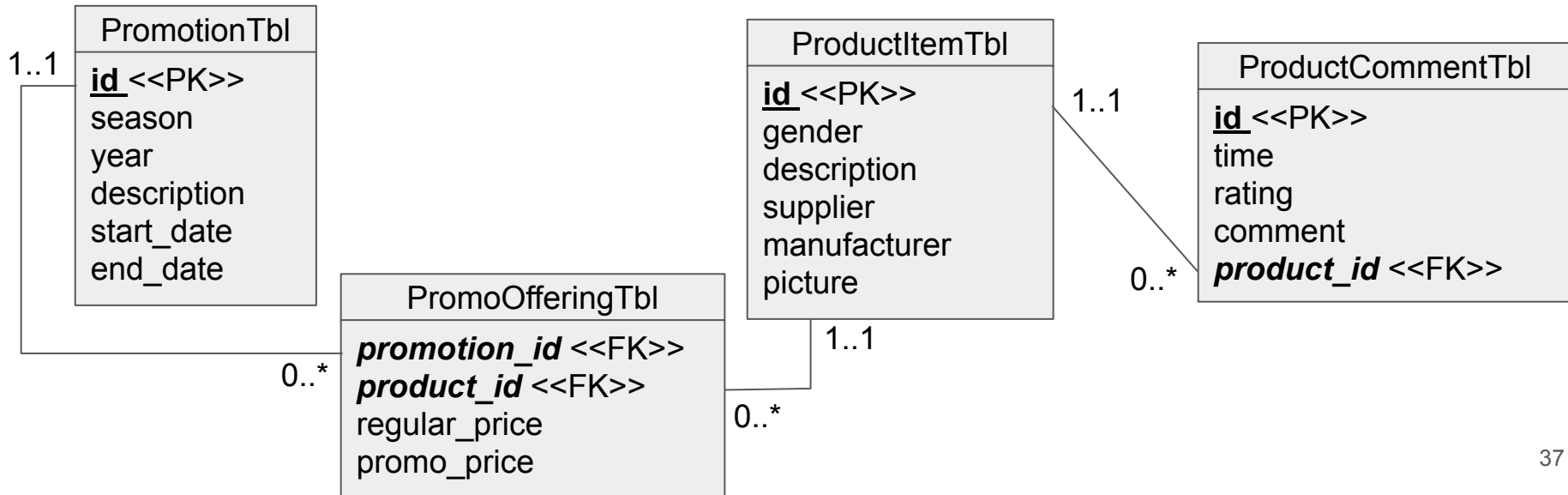
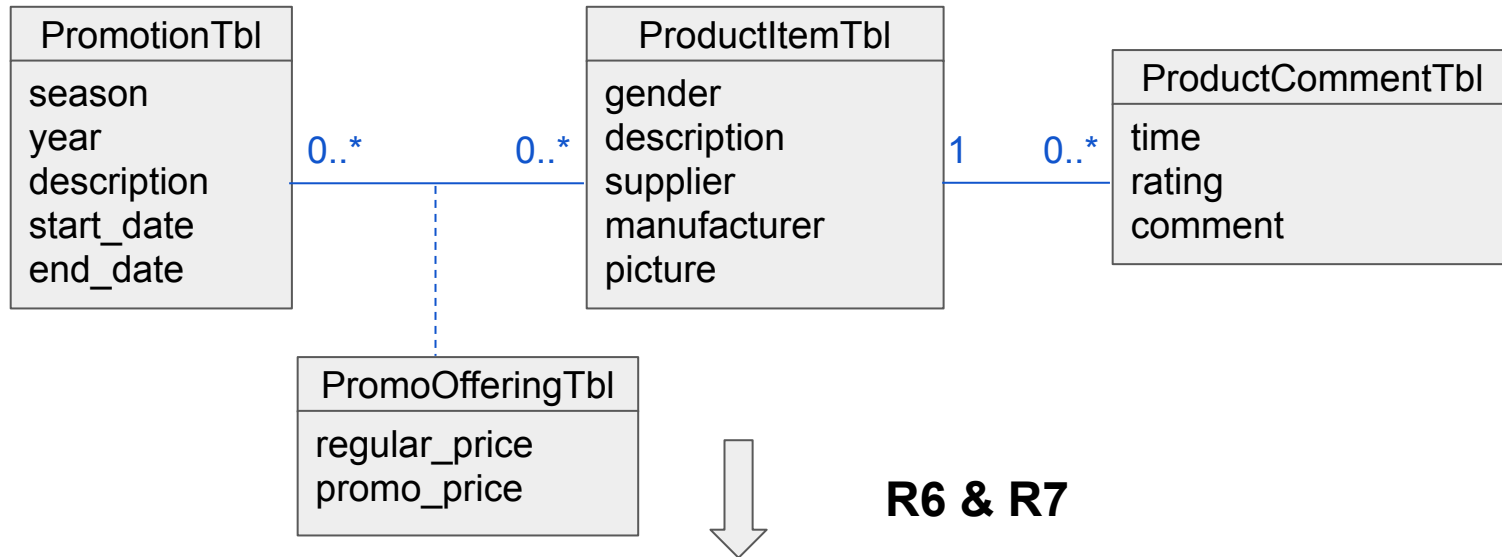
**R8b.** Mở rộng lớp con bằng cách sao chép thuộc tính của lớp cha vào tất cả các lớp con là xóa lớp cha khỏi thiết kế. Hoặc

**R8c.** Làm phẳng cây kế thừa bằng cách sao chép tất cả các thuộc tính trong cây vào lớp gốc, tạo 1 bảng cho cây kế thừa.

# Các quy tắc ánh xạ



# Các quy tắc ánh xạ<sub>(2)</sub>



# Đặc tả bảng

Bảng ProductCommentTbl				
STT	Tên cột	Kiểu dữ liệu	Ràng buộc	Ghi chú
1	id	INT	AUTO_INCREMENT	Mã bình luận
2	time	TIMESTAMP	NOT NULL	Thời điểm bình luận
3	rating	INT	NOT NULL	Điểm bình luận
4	comment	TEXT	NOT NULL	Nội dung bình luận
5	product_id	INT	NOT NULL	Khóa của sản phẩm
Khóa chính		id		
Khóa ngoại		product_id = ProductTbl.id		

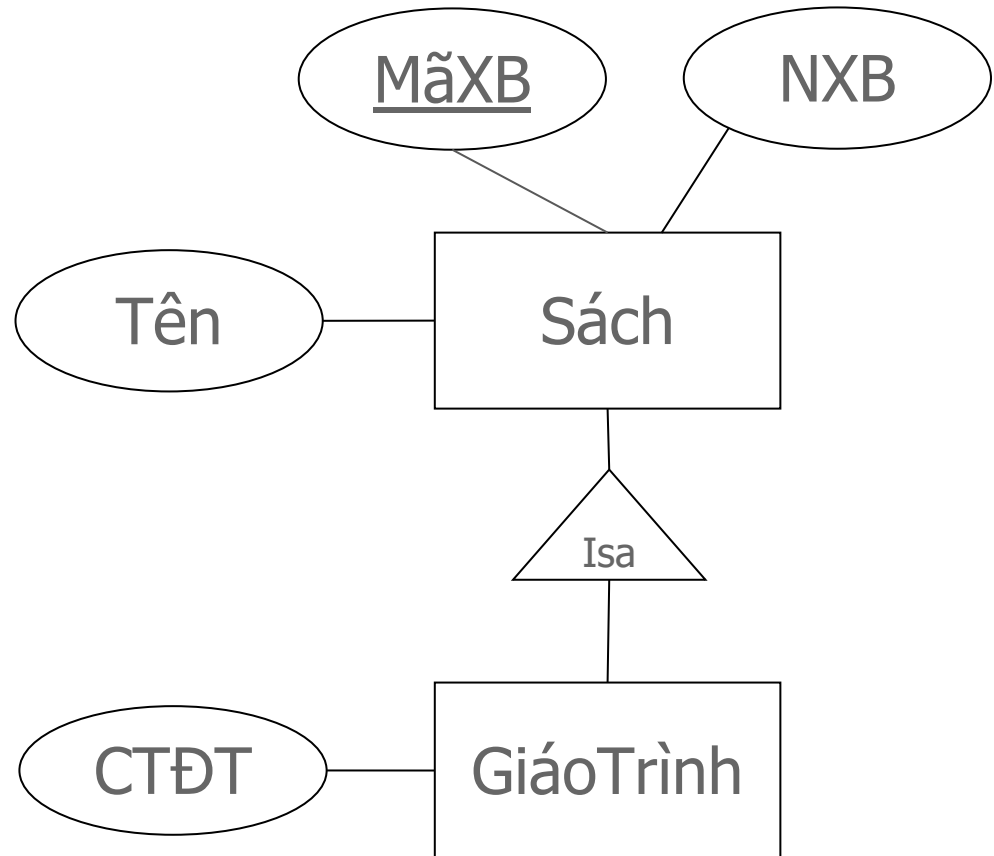
...

# Lưu trữ lớp con với RDBMS

Có nhiều cách lưu khác nhau:

1. *Mô phỏng kế thừa*: Tạo cho mỗi lớp con 1 bảng lưu các thuộc tính riêng, đồng thời kết nối với bảng tương ứng với lớp cha.
2. *Lớp con đầy đủ*: Tạo cho mỗi lớp con 1 bảng lưu đầy đủ các thuộc tính, bao gồm cả thuộc tính riêng và thuộc tính kế thừa từ các lớp cha.
3. *Làm phẳng cây kế thừa*: Tạo 1 bảng cho 1 cây kế thừa với tất cả các thuộc tính trong cây kế thừa.
  - Các đối tượng nhận giá trị **NULL** cho các thuộc tính không thuộc về chúng.
  - ! Dữ liệu có thể ở dạng phi chuẩn

# Ví dụ quan hệ kế thừa





# Ví dụ mô phỏng kế thừa

Sách

Tên	NXB	<b><u>Mã XB</u></b>
LT HĐT	KHKT	K123
Tin ĐC	BKB	B111

Giáo trình

CTĐT	<b><u>Mã XB</u></b>
KHMT	B111

*Lớp con có thể phân tán trên nhiều bảng*

*Thuận tiện xử lý các truy vấn kiểu Tìm tất cả sách (bao gồm cả các giáo trình) được xuất bản bởi 1 nhà xuất bản cụ thể. (tìm kiếm theo thuộc tính của nút gốc trong phạm vi tất cả đối tượng thuộc cây kế thừa)*

# Ví dụ Lớp con đầy đủ

## Sách

Tên	NXB	<u>Mã XB</u>
LT HĐT	KHKT	K123

## Giáo trình

*Không có liên kết giữa bảng của lớp cha và bảng của lớp con*

Tên	NXB	<u>Mã XB</u>	CTĐT
Tin ĐC	BKB	B111	KHMT

*Biểu diễn này thuận tiện để xử lý các truy vấn như “tìm CTĐT của các giáo trình được xuất bản bởi BKB.” (sử dụng cả thuộc tính riêng của lớp con và thuộc tính kế thừa từ lớp cha)*

# Ví dụ Làm phẳng cây kế thừa

<u>Mã số</u>	Phân loại sách
1	Sách
2	Sách Giáo Trình

Tên	NXB	<u>Mã XB</u>	CTĐT	Kiểu
Tin ĐC	BKB	B111	KHMT	1
LT HĐT	KHKT	KT123	<b>NULL</b>	2

*Có thể tiết kiệm dung lượng nếu có ít thuộc tính thường xuyên có giá trị NULL*

*!Lưu ý: Dữ liệu ở dạng phi chuẩn.*

# Ví dụ lưu trữ đối tượng: Quan hệ 1-nhiều

ProductItem		InventoryItem
gender description supplier manufacturer picture	1	0..*
		size color options quantityOnHand averageCost reorderQuantity

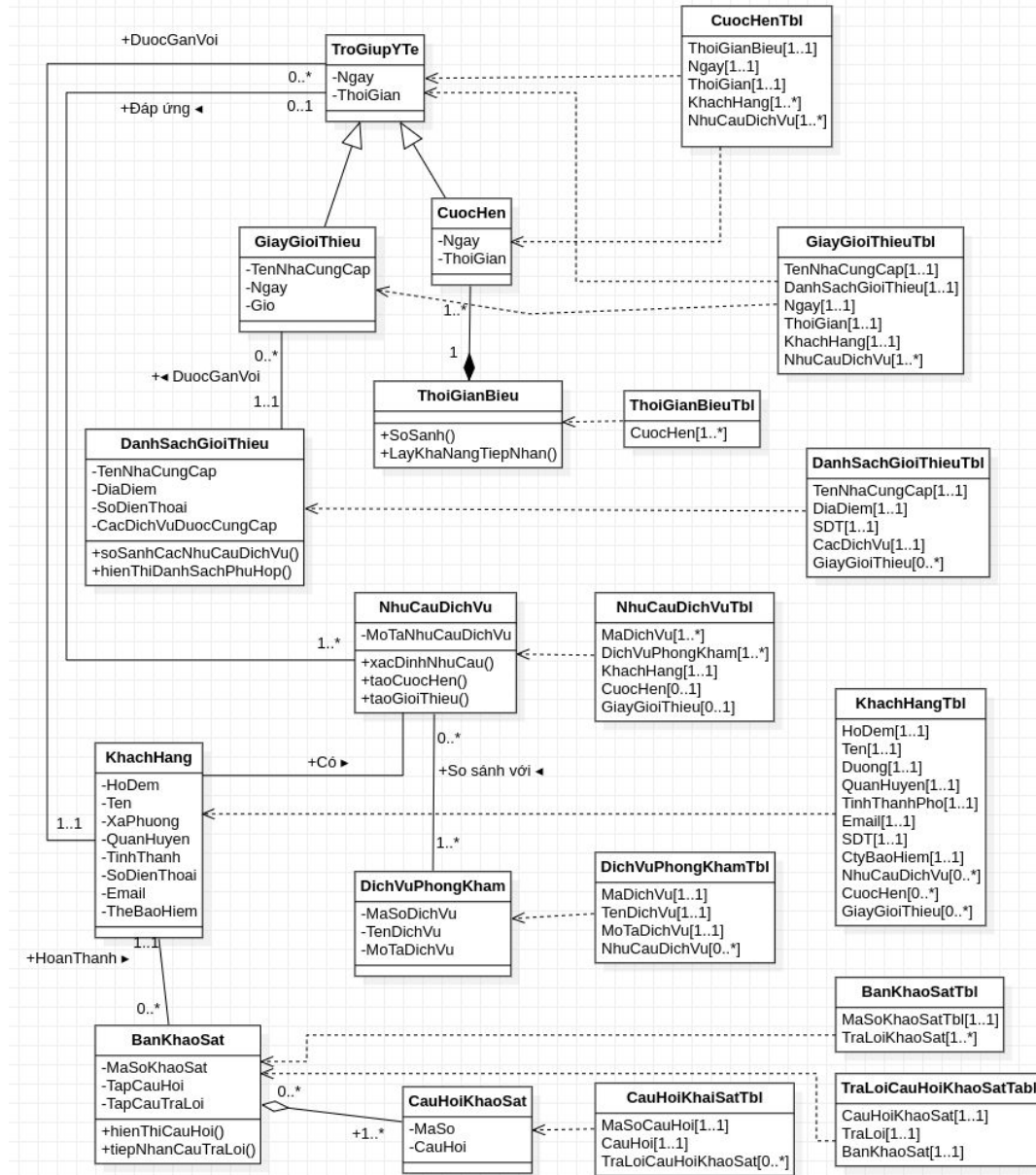
  

ProductItemID	Gender	Description	Supplier	Manufacturer	Picture
10564	Both	Super Alpine Performance Skis	K2	K2	
10766	Man	Extreme Ski Boots	Nordica		
1244	Man	Casual Chino Trousers			
1245	Man	Fleece Crew Sweatshirt			
1246	Man	Fleece Crew Sweatshirt V-Neck			
1247	Man	Fleece Crew Sweatshirt Zippered			
1248	Man	Solid Color Flannel Shirt			
1249	Man	Plaid Flannel Shirt			
1250	Man	Polo Shirt			
1251	Man	Polo Shirt Zippered			

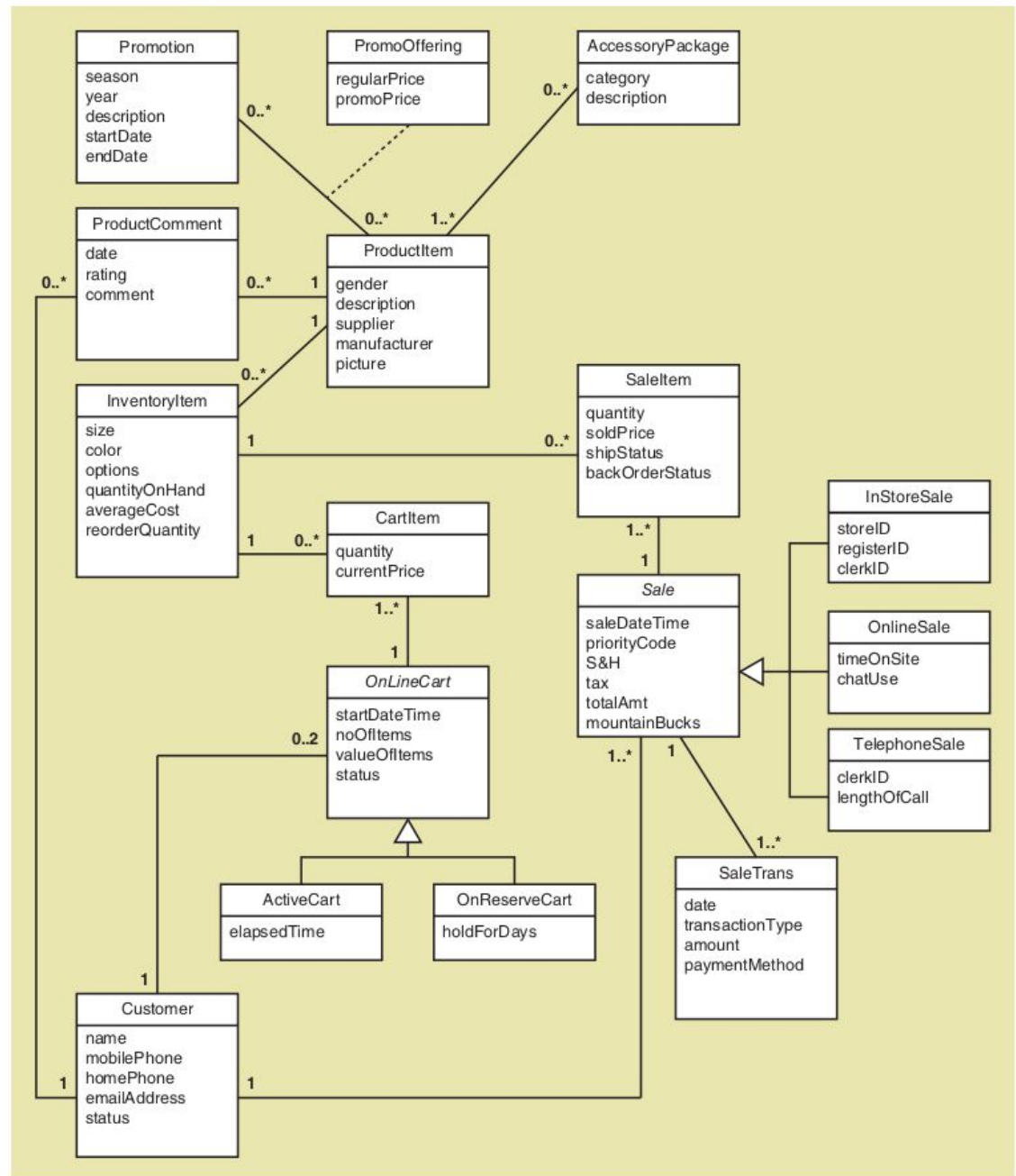
  

InventoryItemID	ProductItemID	Size	Color	Options	QuantityOnHand	Average Cost	ReorderQuantity
86779	1244	30/30	Khaki		45	\$12.75	100
86780	1244	30/30	Slate		10	\$12.75	100
86781	1244	30/30	LightTan		17	\$12.75	100
86782	1244	30/31	Khaki		22	\$12.75	100
86783	1244	30/31	Slate		6	\$12.75	100
86784	1244	30/31	LightTan		31	\$12.75	100
86785	1244	30/32	Khaki		120	\$12.75	100
86786	1244	30/32	Slate		28	\$12.75	100
86787	1244	30/32	LightTan		21	\$12.75	100
86788	1244	30/33	Khaki		7	\$12.75	100

# Ví dụ ánh xạ lớp lĩnh vực tới ORDBMS



# Ví dụ ánh xạ tới RDBMS



# Ví dụ ánh xạ tới RDBMS: Tạo bảng

Áp dụng R1, R2, & R8c: Lớp => Bảng, Thuộc tính => Cột, Làm phẳng cây kế thừa

Bảng	Các cột
AccessoryPackage	Category, Description
CartItem	Quantity, CurrentPrice
Customer	Name, MobilePhone, HomePhone, EmailAddress, Status
InventoryItem	Size, Color, Options, QuantityOnHand, AverageCost, ReorderQuantity
OnlineCart	StartDateTime, NumberOfItems, ValueOfItems, Status, ElapsedTime, HoldForDays
ProductComment	Date, Rating, Comment
ProductItem	Gender, Description, Supplier, Manufacturer, Picture
PromoOffering	RegularPrice, PromoPrice
Promotion	Season, Year, Description, StartDate, EndDate
Sale	SaleDateTime, PriorityCode, ShippingAndHandling, Tax, TotalAmount, MountainBucks, StoreID, RegisterID, ClerkID, TimeOnSite, ChatUse, LengthOfCall
SaleItem	Quantity, SoldPrice, ShipStatus, BackOrderStatus
SaleTransaction	Date, TransactionType, Amount, PaymentMethod

# Ví dụ ánh xạ tới RDBMS: Bổ xung khóa chính

Bảng	Các cột
AccessoryPackage	<b>AccessoryPackageID</b> , Category, Description
CartItem	<b>CartItemID</b> , Quantity, CurrentPrice
Customer	<b>AccountNumber</b> , Name, MobilePhone, HomePhone, EmailAddress, Status
InventoryItem	<b>InventoryItemID</b> , Size, Color, Options, QuantityOnHand, AverageCost, ReorderQuantity
OnlineCart	<b>OnlineCartID</b> , StartDateTime, NumberOfItems, ValueOfItems, Status, ElapsedTime, HoldForDays
ProductComment	<b>ProductCommentID</b> , Date, Rating, Comment
ProductItem	<b>ProductItemID</b> , Gender, Description, Supplier, Manufacturer, Picture
PromoOffering	<b>PromoOfferingID</b> , RegularPrice, PromoPrice
Promotion	<b>PromotionID</b> , Season, Year, Description, StartDate, EndDate
Sale	<b>SaleID</b> , SaleDateTime, PriorityCode, ShippingAndHandling, Tax, TotalAmount, MountainBucks, StoreID, RegisterID, ClerkID, TimeOnSite, ChatUse, LengthOfCall
SaleItem	<b>SaleItemID</b> , Quantity, SoldPrice, ShipStatus, BackOrderStatus
SaleTransaction	<b>SaleTransactionID</b> , Date, TransactionType, Amount, PaymentMethod



# Ví dụ ánh xạ tới RDBMS: Các liên kết

Áp dụng R4, R6, & R7: Bổ xung các khóa ngoại

Bảng	Các cột
Accessory Package	<b>AccessoryPackageID</b> , Category, Description
AccessoryPackageContents	<b>AccessoryPackageID</b> , <b>ProductItemID</b>
CartItem	<b>CartItemID</b> , <i>InventoryItemID</i> , <i>OnlineCartID</i> , Quantity, CurrentPrice
Customer	<b>AccountNumber</b> , Name, MobilePhone, HomePhone, EmailAddress, Status
InventoryItem	<b>InventoryItemID</b> , <i>ProductItemID</i> , Size, Color, Options, QuantityOnHand, AverageCost, ReorderQuantity
OnlineCart	<b>OnlineCartID</b> , <i>CustomerAccountNumber</i> , StartDateTime, NumberOfItems, ValueOfItems, Status, ElapsedTime, HoldForDays
ProductComment	<b>ProductCommentID</b> , <i>ProductItemID</i> , <i>CustomerAccountNumber</i> , Date, Rating, Comment
ProductItem	<b>ProductItemID</b> , Gender, Description, Supplier, Manufacturer, Picture
PromoOffering	<b>PromoOfferingID</b> , <b>PromotionID</b> , <b>ProductItemID</b> , RegularPrice, PromoPrice
Promotion	<b>PromotionID</b> , Season, Year, Description, StartDate, EndDate
Sale	<b>SaleID</b> , <i>CustomerAccountNumber</i> , SaleDateTime, PriorityCode, ShippingAndHandling, Tax, TotalAmount, MountainBucks, StoreID, RegisterID, ClerkID, TimeOnSite, ChatUse, LengthOfCall
SaleItem	<b>SaleItemID</b> , <i>InventoryItemID</i> , <i>SaleID</i> , Quantity, SoldPrice, ShipStatus, BackOrderStatus
SaleTransaction	<b>SaleTransactionID</b> , <i>SaleID</i> , Date, TransactionType, Amount, PaymentMethod

# Ví dụ ánh xạ tới RDBMS: Mô phỏng kế thừa

R8a: Tạo cho mỗi lớp con 1 bảng & thiết lập liên kết với bảng của lớp cha

Bảng	Các cột
Accessory Package	<b>AccessoryPackageID</b> , Category, Description
AccessoryPackageContents	<b>AccessoryPackageID</b> , <b>ProductItemID</b>
CartItem	<b>CartItemID</b> , <i>InventoryItemID</i> , <i>OnlineCartID</i> , Quantity, CurrentPrice
Customer	<b>AccountNumber</b> , Name, MobilePhone, HomePhone, EmailAddress, Status
InventoryItem	<b>InventoryItemID</b> , <i>ProductItemID</i> , Size, Color, Options, QuantityOnHand, AverageCost, ReorderQuantity
OnlineCart	<b>OnlineCartID</b> , <i>CustomerAccountID</i> , StartDateTime, NumberOfItems, ValueOfItems, Status, ElapsedTime, HoldForDays
ActiveCart	<b>OnlineCartID</b> , ElapsedTime
OnReserveCart	<b>OnlineCartID</b> , HoldForDays
ProductComment	<b>ProductCommentID</b> , <i>ProductItemID</i> , <i>CustomerAccountNumber</i> , Date, Rating, Comment
ProductItem	<b>ProductItemID</b> , Gender, Description, Supplier, Manufacturer, Picture
PromoOffering	<b>PromoOfferingID</b> , <b>PromotionID</b> , <b>ProductItemID</b> , RegularPrice, PromoPrice
Promotion	<b>PromotionID</b> , Season, Year, Description, StartDate, EndDate

# Ví dụ ánh xạ tới RDBMS: Mô phỏng kế thừa<sub>(2)</sub>

R8a: Tạo cho mỗi lớp con 1 bảng & thiết lập liên kết với bảng của lớp cha

Bảng	Các cột
Sale	<b>SaleID</b> , <i>CustomerAccountNumber</i> , SaleDateTime, PriorityCode, ShippingAndHandling, Tax, TotalAmount, MountainBucks
InStoreSale	<b>SaleID</b> , StoreID, RegisterID, ClerkID
OnlineSale	<b>SaleID</b> , TimeOnSite, ChatUse
TelephoneSale	<b>SaleID</b> , ClerkID, LengthOfCall
SaleItem	<b>SaleItemID</b> , <i>InventoryItemID</i> , <i>SaleID</i> , Quantity, SoldPrice, ShipStatus, BackOrderStatus
SaleTransaction	<b>SaleTransactionID</b> , <i>SaleID</i> , Date, TransactionType, Amount, PaymentMethod

# Tối ưu hóa CSDL quan hệ

- Tối ưu hóa lưu trữ
  - Chuẩn hóa các bảng
  - Giảm dư thừa dữ liệu và loại bỏ các giá trị NULL
- Tối ưu hóa tốc độ truy cập
  - Giải chuẩn 1 số bảng / giảm nhu cầu kết nối các bảng để truy xuất dữ liệu để giảm thời gian xử lý
  - V.V..

# Chuẩn hóa CSDL quan hệ

- Dữ liệu phi chuẩn: Các luật chuẩn hóa không được đáp ứng

3 mức chuẩn hóa đầu tiên gồm có:

- Dạng chuẩn 1: Không có trường đa giá trị/Thuộc tính chỉ có giá trị nguyên tố
- Dạng chuẩn 2: Thuộc chuẩn 1 và tất cả các thuộc tính không nằm trong khóa đều phụ thuộc vào toàn bộ khóa chính
- Dạng chuẩn 3: Thuộc chuẩn 2 và mọi thuộc tính không khóa chỉ phụ thuộc vào khóa / Không có thuộc tính không nằm trong khóa phụ thuộc vào thuộc tính khác không nằm trong khóa

# Ví dụ chuẩn hóa: Dữ liệu phi chuẩn

SSN	Name	Department	Salary	Dependents
111-22-3333	Mary Smith	Accounting	40,000	John, Alice, Dave
222-33-4444	Jose Pena	Marketing	50,000	---
333-44-5555	Frank Collins	Production	35,000	Jan, Julia

SSN	Name	Department	Salary	Dependent	Dependent	Dependent
111-22-3333	Mary Smith	Accounting	40,000	John	Alice	Dave
222-33-4444	Jose Pena	Marketing	50,000			
333-44-5555	Frank Collins	Production	35,000	Jan	Julia	

# Ví dụ chuẩn hóa: Dữ liệu ở dạng chuẩn

SSN	Name	Department	Salary
111-22-3333	Mary Smith	Accounting	40,000
222-33-4444	Jose Pena	Marketing	50,000
333-44-5555	Frank Collins	Production	35,000

RecordID	SSN	Dependent
1	111-22-3333	John
2	111-22-3333	Alice
3	111-22-3333	Dave
4	333-44-5555	Jan
5	333-44-5555	Julia

# Ví dụ chuẩn hóa<sup>(2)</sup>

## Dữ liệu phi chuẩn

Order
-Order Number : unsigned long
-Date : Date
-Cust ID : unsigned long
-Last Name : String
-First Name : String
-State : String
-Tax Rate : float
-Product 1 Number : unsigned long
-Product 1 Desc. : String
-Product 1 Price : double
-Product 1 Qty. : unsigned long
-Product 2 Number : unsigned long
-Product 2 Desc. : String
-Product 2 Price : double
-Product 2 Qty. : unsigned long
-Product 3 Number : unsigned long
-Product 3 Desc. : String
-Product 3 Price : double
-Product 3 Qty. : unsigned long

Redundant Data

Null Cells

Sample Records:

Order Number	Date	Cust ID	Last Name	First Name	State	Tax Rate	Prod. 1 Number	Prod. 1 Desc.	Prod. 1 Price	Prod. 1 Qty.	Prod. 2 Number	Prod. 2 Desc.	Prod. 2 Price	Prod. 2 Qty.
239	11/23/00	1035	Black	John	MD	0.05	555	Cheese Tray	\$45.00	2				
260	11/24/00	1035	Black	John	MD	0.05	444	Wine Gift Pack	\$60.00	1				
273	11/27/00	1035	Black	John	MD	0.05	222	Bottle Opener	\$12.00	1				
241	11/23/00	1123	Williams	Mary	CA	0.08	444	Wine Gift Pack	\$60.00	2				
262	11/24/00	1123	Williams	Mary	CA	0.08	222	Bottle Opener	\$12.00	2				
287	11/27/00	1123	Williams	Mary	CA	0.08	222	Bottle Opener	\$12.00	2				
290	11/30/00	1123	Williams	Mary	CA	0.08	555	Cheese Tray	\$45.00	3				
234	11/23/00	2242	DeBerry	Ann	DC	0.065	555	Cheese Tray	\$45.00	2				
237	11/23/00	2242	DeBerry	Ann	DC	0.065	111	Wine Guide	\$15.00	1	444	Wine Gift Pack	\$60.00	1
238	11/23/00	2242	DeBerry	Ann	DC	0.065	444	Wine Gift Pack	\$60.00	1				
245	11/24/00	2242	DeBerry	Ann	DC	0.065	222	Bottle Opener	\$12.00	1				
250	11/24/00	2242	DeBerry	Ann	DC	0.065	222	Bottle Opener	\$12.00	1				
252	11/24/00	2242	DeBerry	Ann	DC	0.065	222	Bottle Opener	\$12.00	1	444	Wine Gift Pack	\$60.00	2
253	11/24/00	2242	DeBerry	Ann	DC	0.065	222	Bottle Opener	\$12.00	1	444	Wine Gift Pack	\$60.00	1



# Ví dụ chuẩn hóa: Dạng chuẩn 1

Sample Records:

Order Table							Product Order Table				
Order Number	Date	Cust ID	Last Name	First Name	State	Tax Rate	Order Number	Product Number	Product Desc	Product Price	Product Qty
239	11/23/00	1035	Black	John	MD	0.05	239	555	Cheese Tray	\$45.00	2
260	11/24/00	1035	Black	John	MD	0.05	260	444	Wine Gift Pack	\$60.00	1
273	11/27/00	1035	Black	John	MD	0.05	273	222	Bottle Opener	\$12.00	1
241	11/23/00	1123	Williams	Mary	CA	0.08	241	444	Wine Gift Pack	\$60.00	2
262	11/24/00	1123	Williams	Mary	CA	0.08	262	222	Bottle Opener	\$12.00	2
287	11/27/00	1123	Williams	Mary	CA	0.08	287	222	Bottle Opener	\$12.00	2
290	11/30/00	1123	Williams	Mary	CA	0.08	290	555	Cheese Tray	\$45.00	3
234	11/23/00	2242	DeBerry	Ann	DC	0.065	234	555	Cheese Tray	\$45.00	2
237	11/23/00	2242	DeBerry	Ann	DC	0.065	237	111	Wine Guide	\$15.00	1
238	11/23/00	2242	DeBerry	Ann	DC	0.065	237	444	Wine Gift Pack	\$60.00	1
245	11/24/00	2242	DeBerry	Ann	DC	0.065	238	444	Wine Gift Pack	\$60.00	1
250	11/24/00	2242	DeBerry	Ann	DC	0.065	245	222	Bottle Opener	\$12.00	1
252	11/24/00	2242	DeBerry	Ann	DC	0.065	250	222	Bottle Opener	\$12.00	1
253	11/24/00	2242	DeBerry	Ann	DC	0.065	252	222	Bottle Opener	\$12.00	1
297	11/30/00	2242	DeBerry	Ann	DC	0.065	252	444	Wine Gift Pack	\$60.00	2
243	11/24/00	4254	Bailey	Ryan	MD	0.05	253	222	Bottle Opener	\$12.00	1
246	11/24/00	4254	Bailey	Ryan	MD	0.05	253	444	Wine Gift Pack	\$60.00	1
248	11/24/00	4254	Bailey	Ryan	MD	0.05	297	333	Jams & Jellies	\$20.00	2
235	11/23/00	9500	Chin	April	KS	0.05	243	555	Cheese Tray	\$45.00	2
242	11/23/00	9500	Chin	April	KS	0.05	246	333	Jams & Jellies	\$20.00	3
244	11/24/00	9500	Chin	April	KS	0.05	248	222	Bottle Opener	\$12.00	1
251	11/24/00	9500	Chin	April	KS	0.05	248	333	Jams & Jellies	\$20.00	2
							248	111	Wine Guide	\$15.00	1
							235	222	Bottle Opener	\$12.00	1
							242	333	Jams & Jellies	\$20.00	3
							244	222	Bottle Opener	\$12.00	2
							251	111	Wine Guide	\$15.00	2

Order 237 has 2 products

Order 248 has 3 products

# Ví dụ chuẩn hóa: Dạng chuẩn 2

Customer Table		
Cust ID	Last Name	First Name
1035	Black	John
1123	Williams	Mary
2242	DeBerry	Ann
4254	Bailey	Ryan
9500	Chin	April

Last Name and First Name was moved to the Customer table to eliminate redundancy

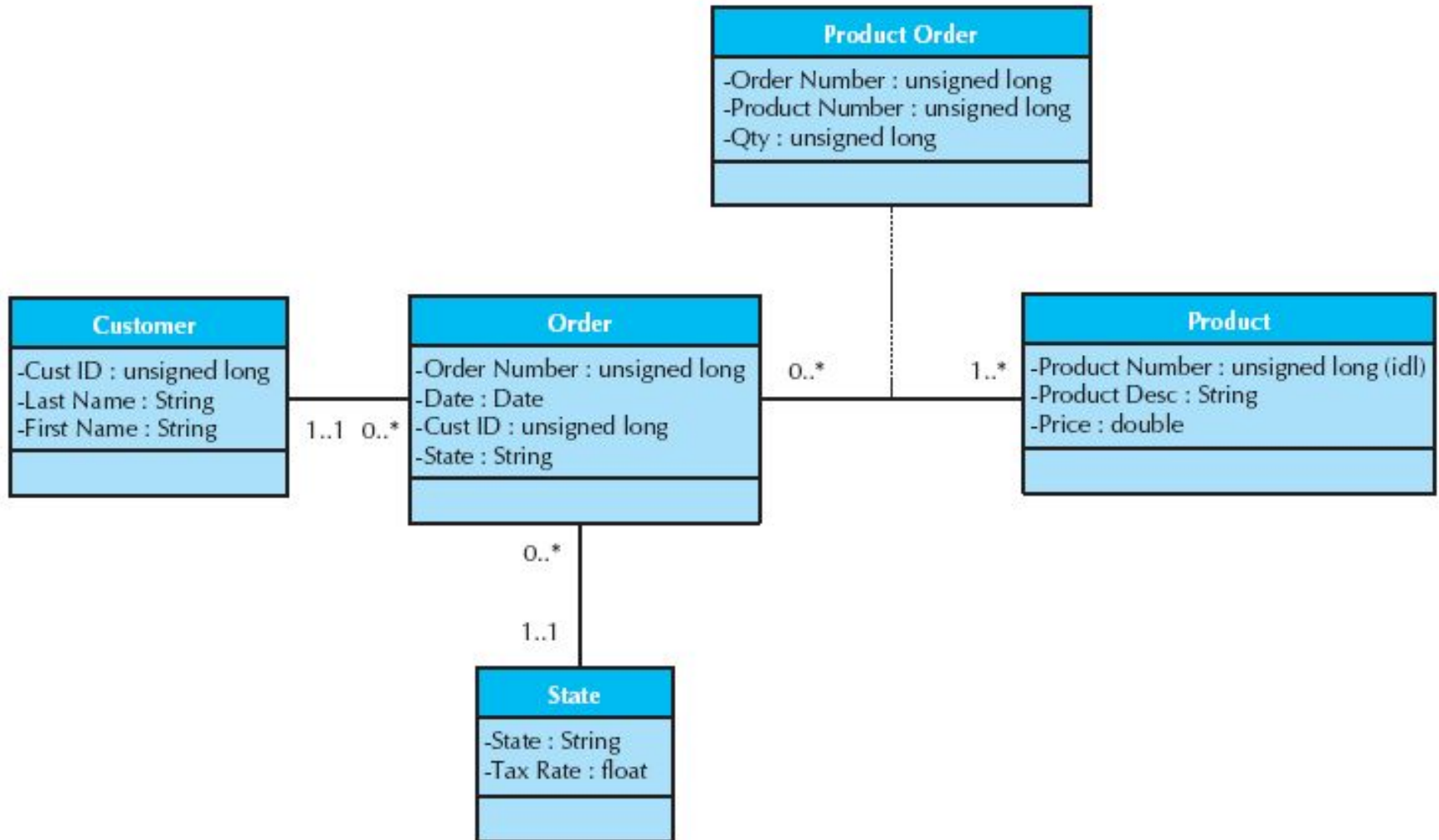
Order Table					Product Order Table				
Order Number	Date	Cust ID	State	Tax Rate	Order Number	Product Number	Product Qty		
239	11/23/00	1035	MD	0.05	239	555	2		
260	11/24/00	1035	MD	0.05	260	444	1		
273	11/27/00	1035	MD	0.05	273	222	1		
241	11/23/00	1123	CA	0.08	241	444	2		
262	11/24/00	1123	CA	0.08	262	222	2		
287	11/27/00	1123	CA	0.08	287	222	2		
290	11/30/00	1123	CA	0.08	290	555	3		
234	11/23/00	2242	DC	0.065	234	555	2		
237	11/23/00	2242	DC	0.065	237	111	1		
238	11/23/00	2242	DC	0.065	237	444	1		
245	11/24/00	2242	DC	0.065	238	444	1		
250	11/24/00	2242	DC	0.065	245	222	1		
252	11/24/00	2242	DC	0.065	250	222	1		
253	11/24/00	2242	DC	0.065	252	222	1		
297	11/30/00	2242	DC	0.065	252	444	2		
243	11/24/00	4254	MD	0.05	253	222	1		
246	11/24/00	4254	MD	0.05	253	444	1		
248	11/24/00	4254	MD	0.05	297	333	2		
235	11/23/00	9500	KS	0.05	243	555	2		
242	11/23/00	9500	KS	0.05	246	333	3		
244	11/24/00	9500	KS	0.05	248	222	1		
251	11/24/00	9500	KS	0.05	248	333	2		
					248	111	1		
					235	222	1		
					242	333	3		
					244	222	2		
					251	111	2		

Product Table		
Product Number	Product Desc	Product Price
111	Wine Guide	\$15.00
222	Bottle Opener	\$12.00
333	Jams & Jellies	\$20.00
444	Wine Gift Pack	\$60.00
555	Cheese Tray	\$45.00

Product Desc and Price was moved to the Product table to eliminate redundancy

tax & state?

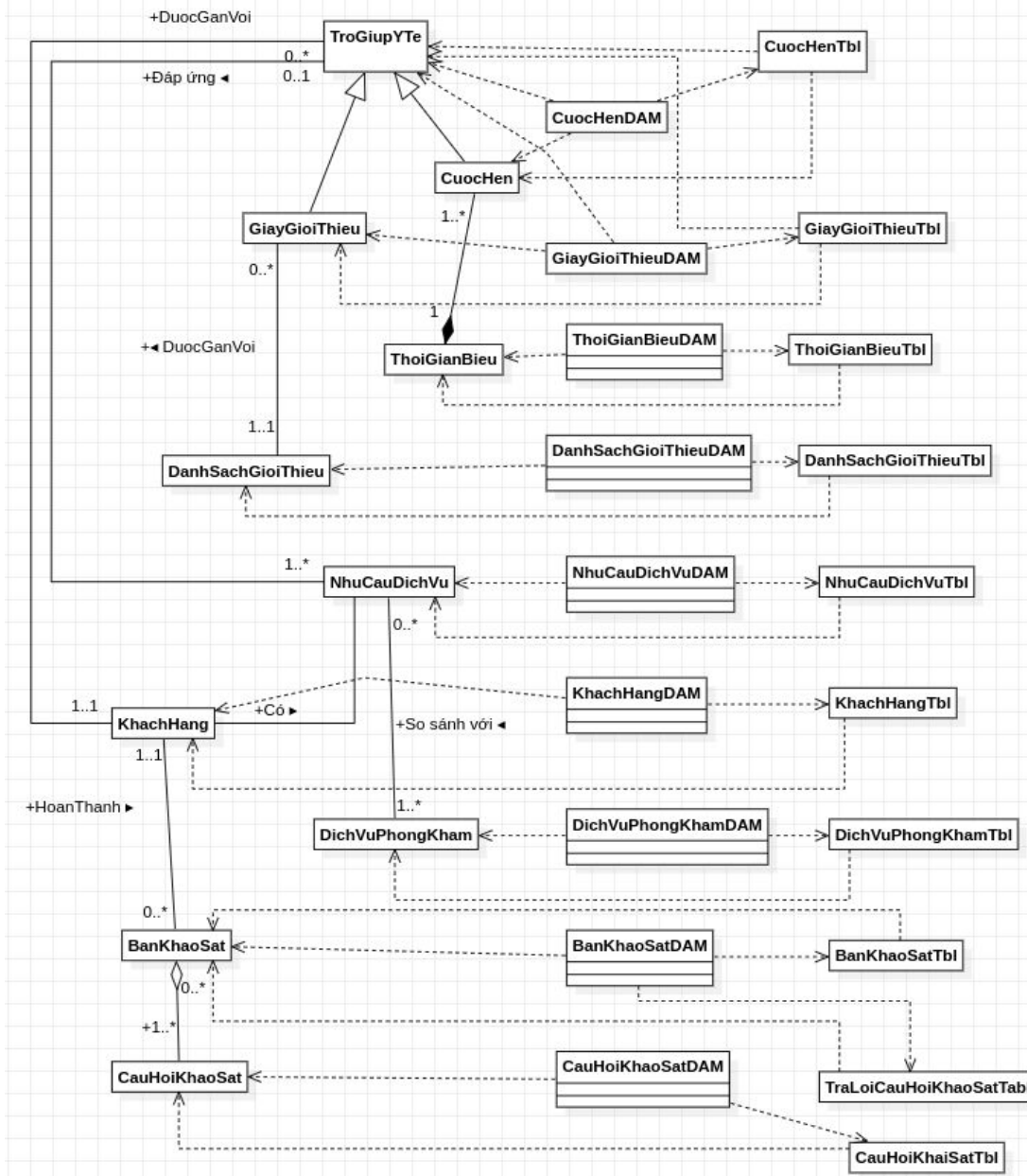
# Ví dụ chuẩn hóa: 3NF



# Các lớp truy cập & quản lý dữ liệu

- Các lớp truy cập và quản lý dữ liệu / Data Access and Management (DAM) Classes
- Giữ vai trò trung gian kết nối các lớp lĩnh vực ứng dụng và cơ sở dữ liệu
- Tạo 1 hoặc nhiều lớp DAM cho mỗi lớp lĩnh vực ứng dụng
  - Do dữ liệu phân tán trên nhiều bảng
  - Phân tách giao diện (ISP)
  - V.V..

# Ví dụ các lớp DAM

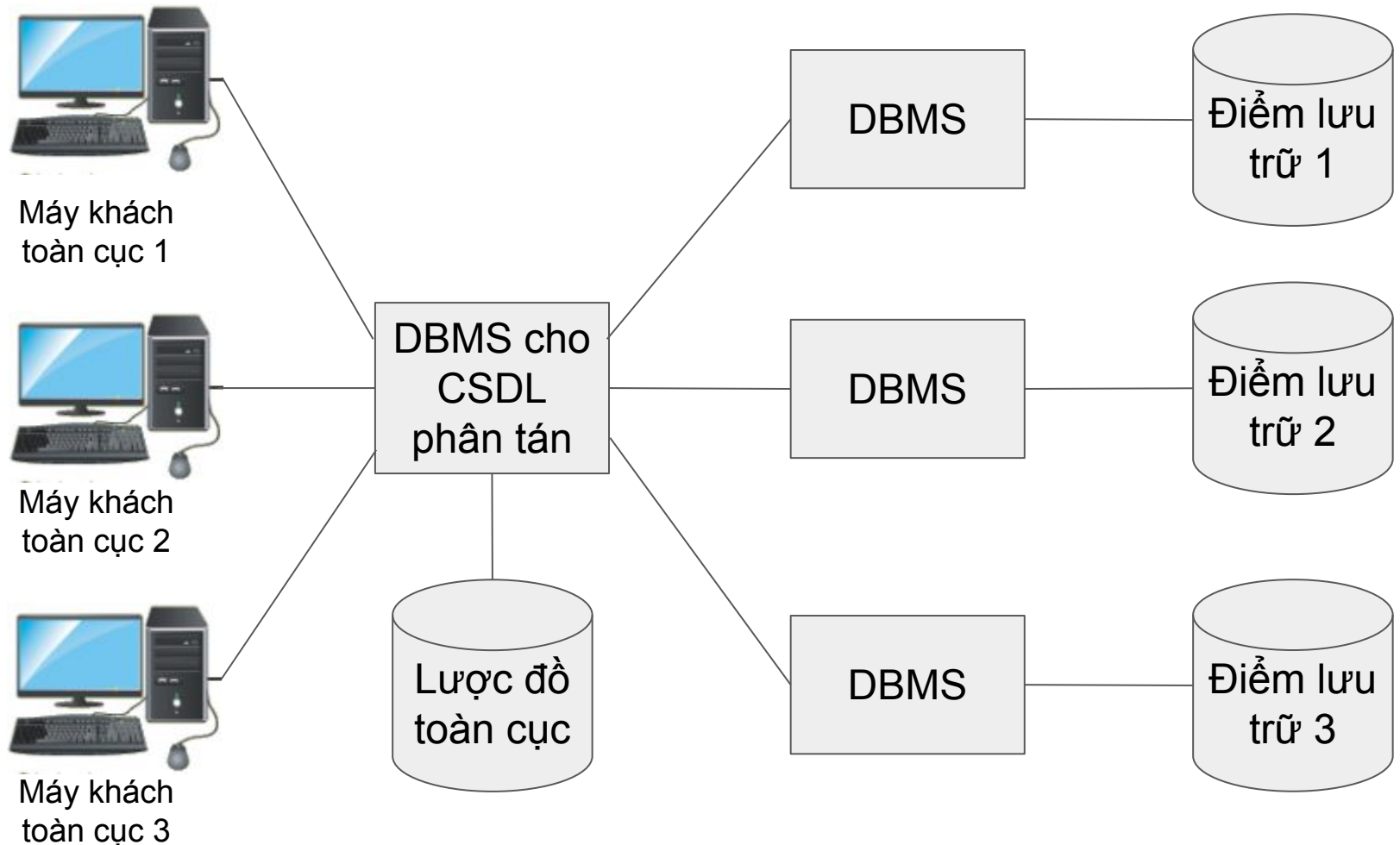


# Kiến trúc CSDL phân tán

- CSDL phân tán đồng nhất / Homogeneous distributed database được lưu ở nhiều nơi với cùng DBMS và 1 lược đồ toàn cục.
- CSDL phân tán không đồng nhất / Heterogeneous distributed database được lưu ở nhiều nơi với các DBMS khác nhau và có thể có lược đồ cục bộ.

# CSDL phân tán đồng nhất

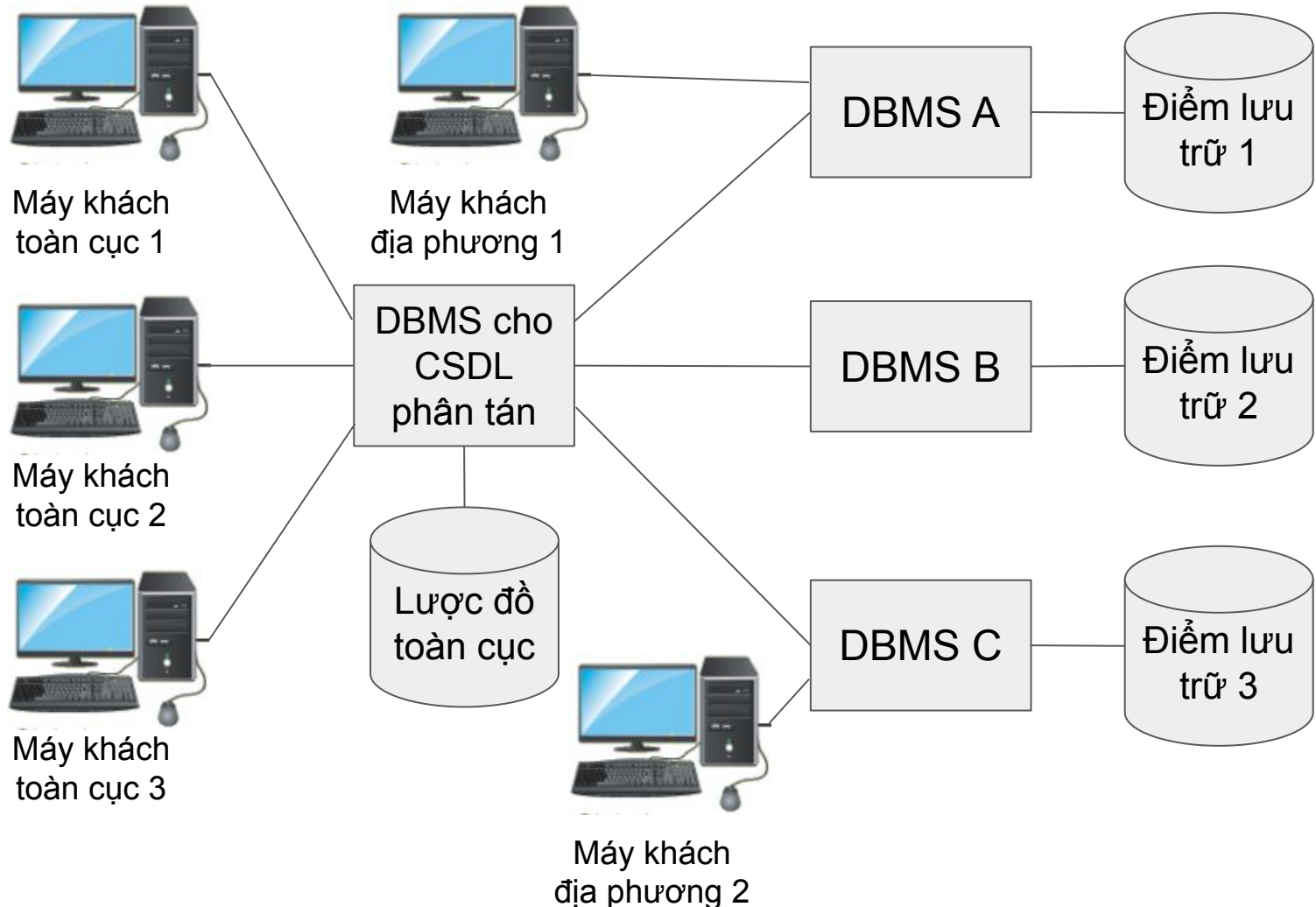
Truy cập thông qua 1 Hệ quản trị CSDL và lược đồ chung





# CSDL phân tán không đồng nhất

Truy cập thông qua các DBMS riêng. Có thể có DBMS và lược đồ chung





# Các cách triển khai

- Lập dữ liệu - Mỗi điểm lưu trữ có 1 bản sao
  - Đồng bộ: Cập nhật tất cả các bản ghi khi thay đổi 1 bản ghi bất kỳ.
- Phân mảnh theo chiều ngang - Mỗi điểm chứa 1 nhóm dòng

AcctNumb	LastName	FirstName	SSN	TypeOfAcct	Balance	DateLastActivity	
01-85562-1	Jones	Bill	878-77-9890	Checking	\$ 7,908.39	5/9/2014	U.S. accounts
01-85444-2	Johnson	Harold	676-44-3433	Checking	\$25,698.33	5/2/2013	
02-45443-2	Williams	Jonathon	343-44-2322	Checking	\$ 3,938.77	4/4/2012	
01-34999-1	Redd	Mary	898-79-3487	Savings	\$12,898.71	12/2/2013	
01-23989-2	Chun	Tun	233-59-6765	Savings	\$ 8,932.67	1/8/2014	Hong Kong accounts
01-87889-4	Gang	Bao	322-48-3545	Checking	\$ 568.33	3/4/2014	
01-32339-2	Jiang	Rui	550-43-5454	Savings	\$35,788.23	7/8/2014	
02-39988-1	Ma	Shuo	343-98-2345	Checking	\$ 1,893.55	8/23/2014	

# Các cách triển khai<sub>(2)</sub>

- Phân mảnh theo chiều dọc - Mỗi điểm lưu 1 nhóm cột

The diagram illustrates vertical partitioning of a table. A large table is divided into two sections by a vertical line. The left section is labeled 'U.S. partition' and the right section is labeled 'Japanese partition'. The 'U.S. partition' contains the first four columns: PartNumber, Description, Manufacturer, and QtyOnHand. The 'Japanese partition' contains the next three columns: SchematicNo, InspectionNo, and QtyOnHand2. Arrows point from the labels to their respective partitions.

PartNumber	Description	Manufacturer	QtyOnHand	SchematicNo	InspectionNo	QtyOnHand2
4568-AC9	Screw assembly	Westco Inc	348	42-596	56	346
7618-IF44	Handle assembly	Japan Tools	276	16-443	43	434
7678-AD22	Door1 assembly	Tokyo Hardware	58	76-454	65	765
4890-XX88	Door2 assembly	Tokyo Hardware	97	78-443	34	446
9890-CD87	Interior module	Open Electronics	454	23-794	67	454
6766-DY65	Interior seal assembly	Sealants Inc	611	56-545	23	2132
8769-DD77	Connection assembly	Open Electronics	546	90-787	22	722
2311-AB28	Crank assembly	Westco Inc	768	33-571	12	121
3432-RB88	Double pulley assembly	Westco Inc	564	90-443	43	342

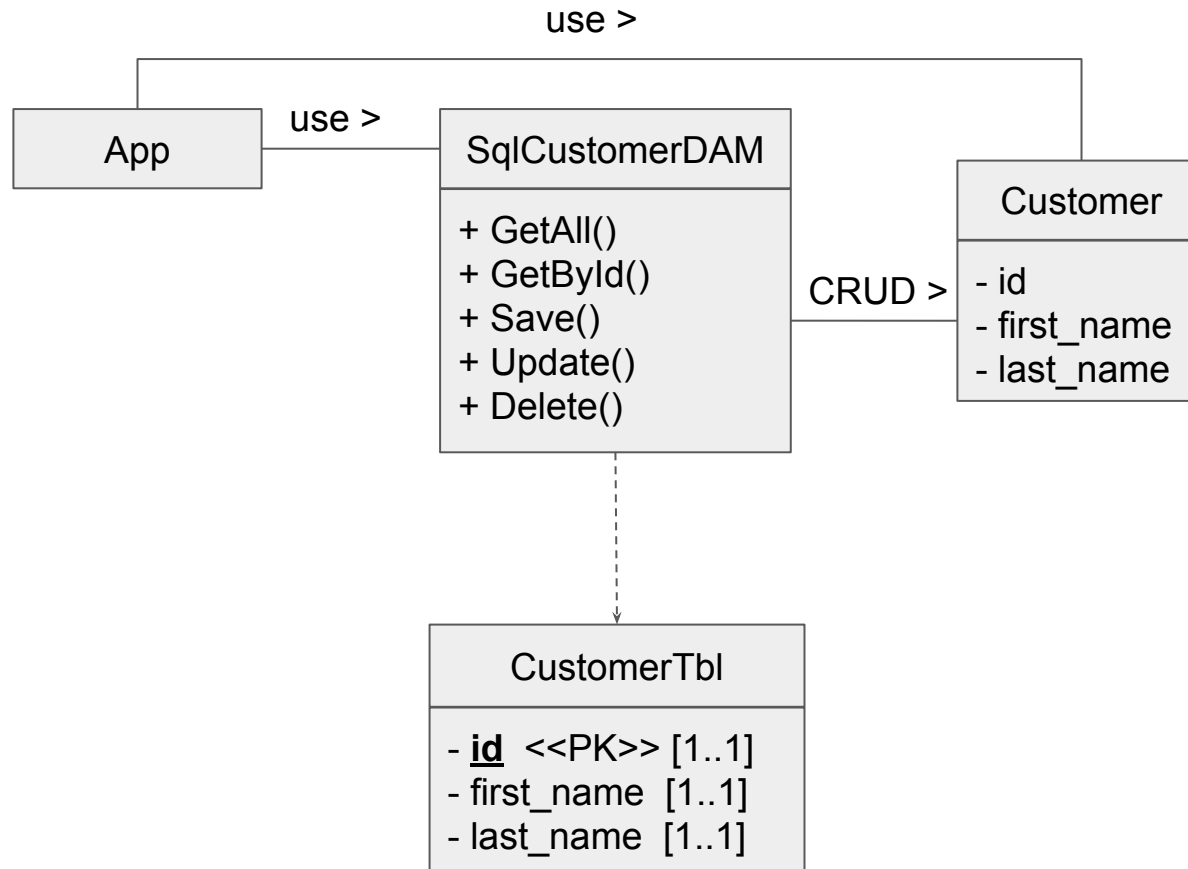
- Kết hợp lập, phân mảnh theo chiều ngang và phân mảnh theo chiều dọc

# Bảo vệ CSDL

- Lịch sử giao dịch: Kỹ thuật lưu lại tất cả các cập nhật
  - Giúp ngăn chặn gian lận
  - Khôi phục sau sự cố
- Kiểm soát xung đột và cập nhật
  - Giao dịch - 1 phần công việc bao gồm nhiều bước, trong đó hoặc tất cả các bước cùng hoàn thành hoặc không chấp nhận kết quả của bước nào.
  - Khóa CSDL - kỹ thuật kiểm soát truy cập (1 phần) CSDL.
  - Khóa dùng chung / khóa đọc - Cho phép nhiều người dùng cùng đọc dữ liệu.
  - Khóa loại trừ / khóa ghi - Chỉ 1 người dùng có thể truy cập phần CSDL bị khóa.

# Mẫu thiết kế DAO

# Sử dụng lớp DAM và RDBMS

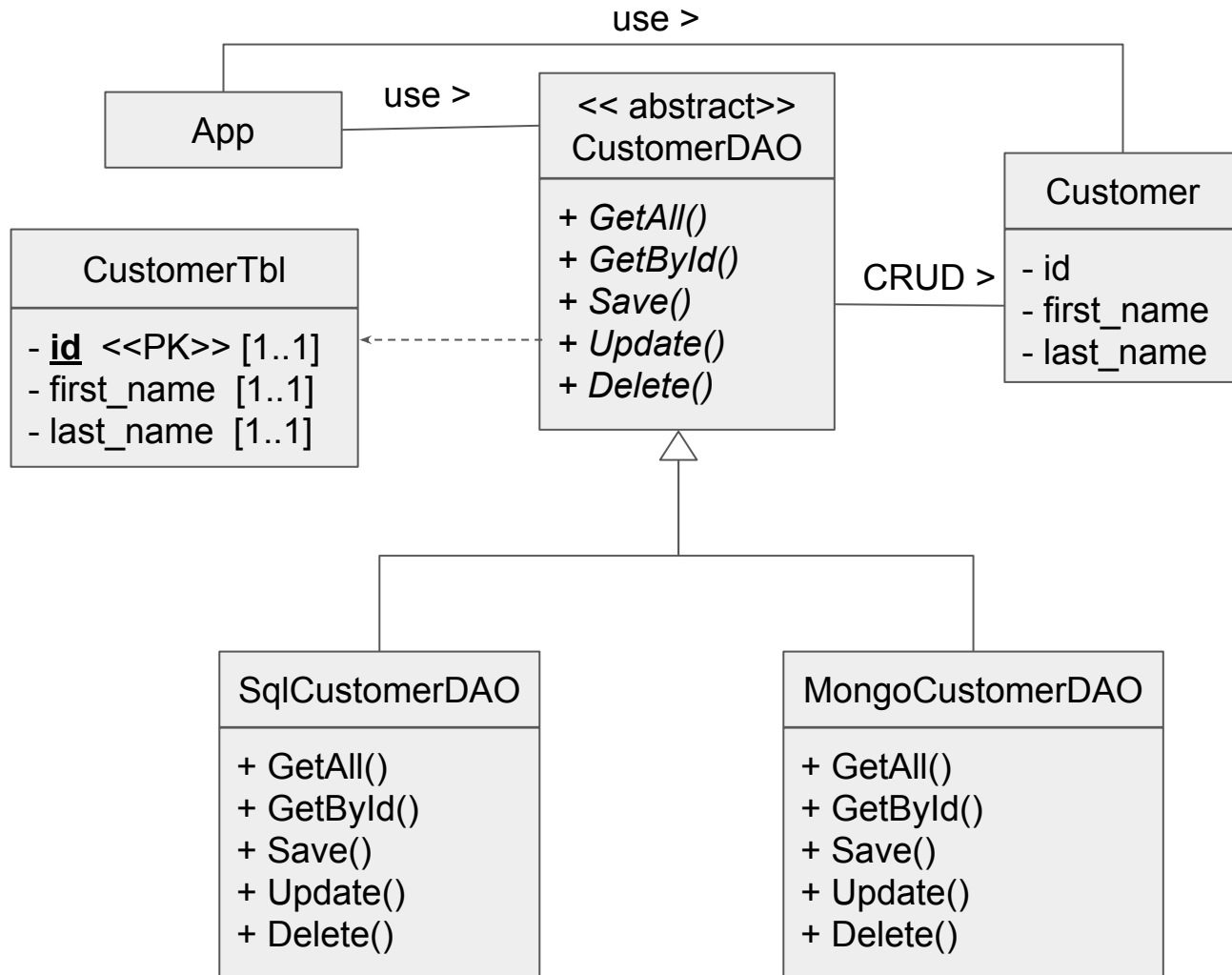


# Đặc tả bảng

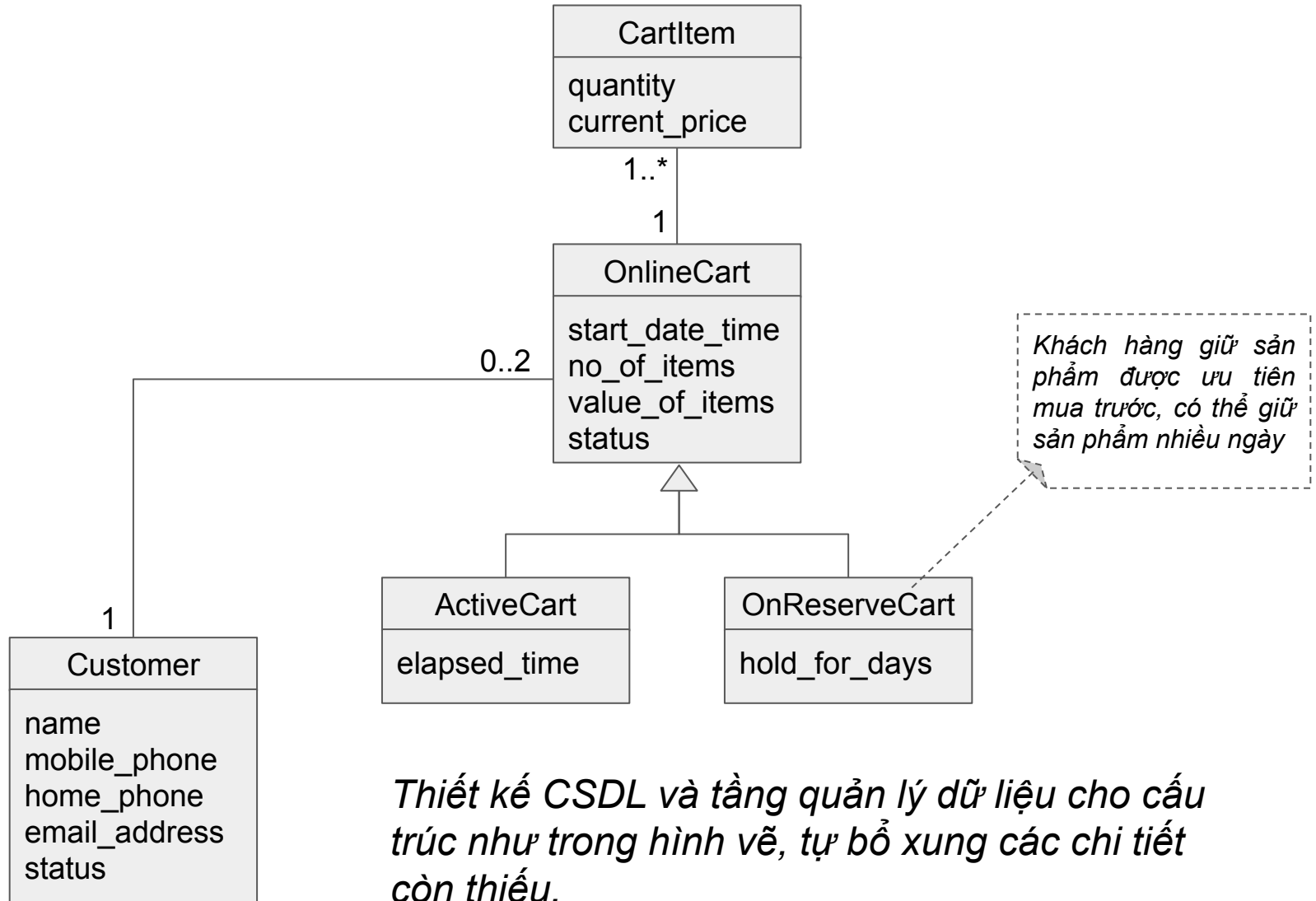
CustomerTbl
- <b>id</b> <<PK>> [1..1] - first_name [1..1] - last_name [1..1]

Bảng CustomerTbl				
STT	Tên cột	Kiểu dữ liệu	Ràng buộc	Ghi chú
1	id	INT	AUTO_INCREMENT	Mã khách hàng
2	first_name	VARCHAR (255)	NOT NULL	Tên
3	last_name	VARCHAR(255)	NOT NULL	Họ và Tên
Khóa chính		id		
Khóa ngoại		Không có		

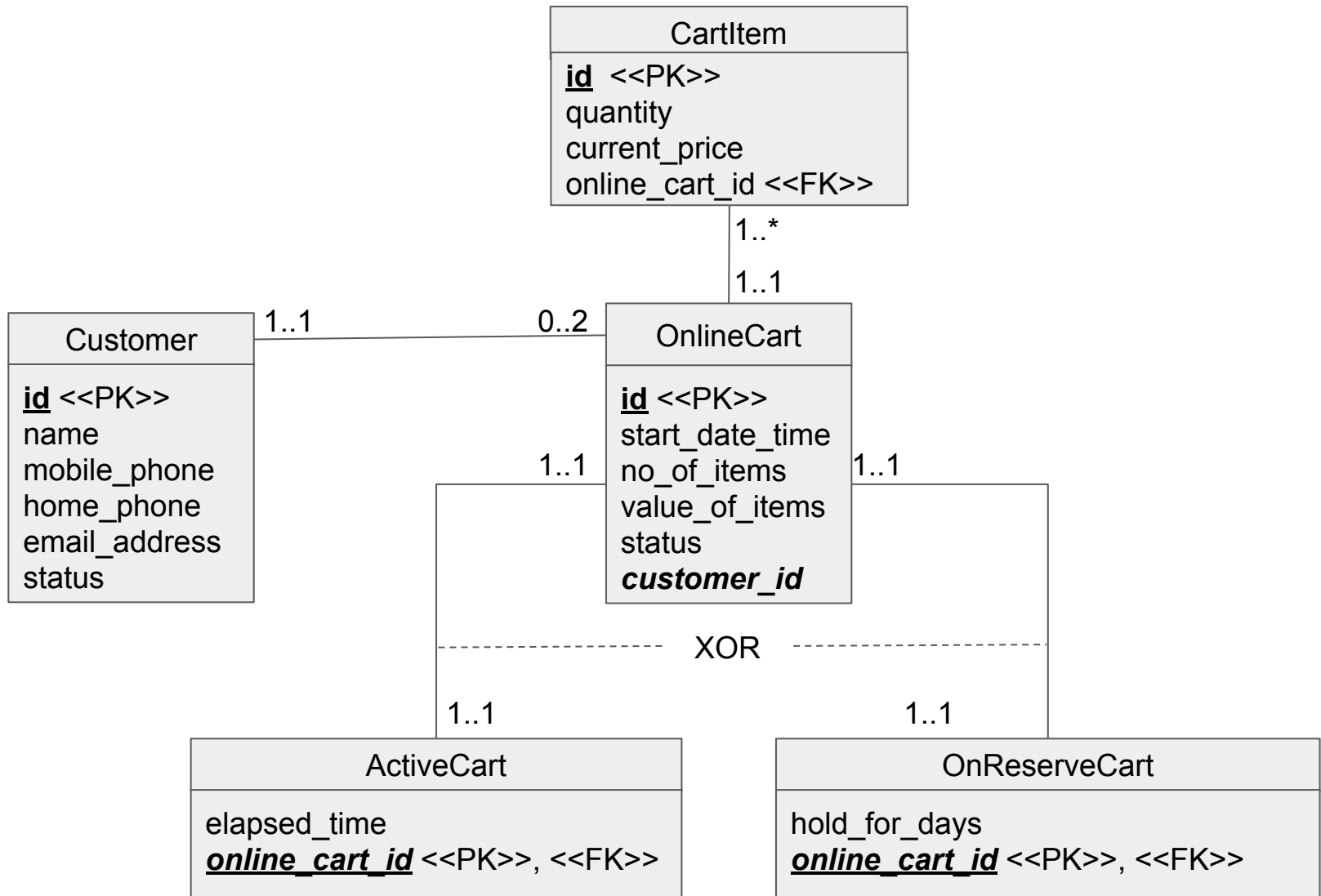
# Mẫu thiết kế DAO



# Bài tập thiết kế tầng quản lý dữ liệu

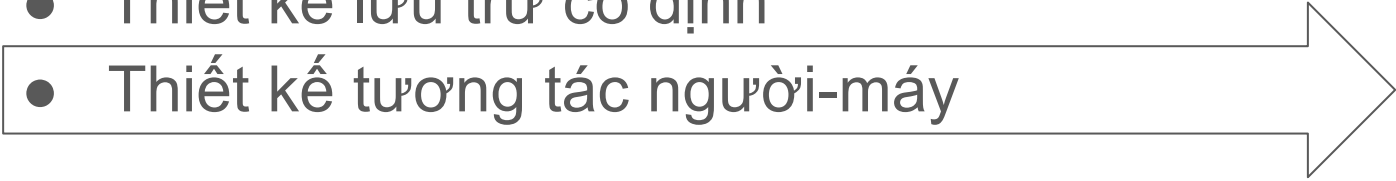






- ...
- Đặc tả các bảng
- Tạo các lớp DAM hoặc DAO, ...

# Nội dung

- Các nguyên lý thiết kế (SOLID)
  - Thiết kế lưu trữ cố định
  - Thiết kế tương tác người-máy
- 

# Tổng quan về thiết kế giao diện

- Thiết kế giao diện theo nghĩa tổng quát xác định cách hệ thống tương tác với các tác nhân (người dùng, hệ thống khác, v.v..)
  - Giao diện hệ thống: Hỗ trợ tương tác máy-máy / hỗ trợ tích hợp hệ thống
  - Giao diện người dùng: Hỗ trợ tương tác người-máy / hỗ trợ người dùng sử dụng hệ thống
    - Người dùng sử dụng và trải nghiệm các dịch vụ được cung cấp bởi hệ thống thông qua giao diện
    - **Chúng ta sẽ tập trung vào nội dung này!**
- Các nguyên lý được áp dụng trong thiết kế giao diện
- Giao diện được thiết kế không tốt có thể làm giảm hiệu quả sử dụng hệ thống.

# Các thành phần hỗ trợ tương tác người-máy

## *Tương tác người-máy / Human-Computer Interaction (HCI)*

Thiết bị: Màn hình thường, màn hình cảm ứng, bàn phím thực, bàn phím ảo, chuột, webcam, mic, v.v..

Các giao diện người dùng (phần mềm): Cửa sổ, biểu mẫu, hộp thoại, hình ảnh, nút bấm, danh sách, bảng, biểu đồ, v.v..



# Các vấn đề thiết kế giao diện

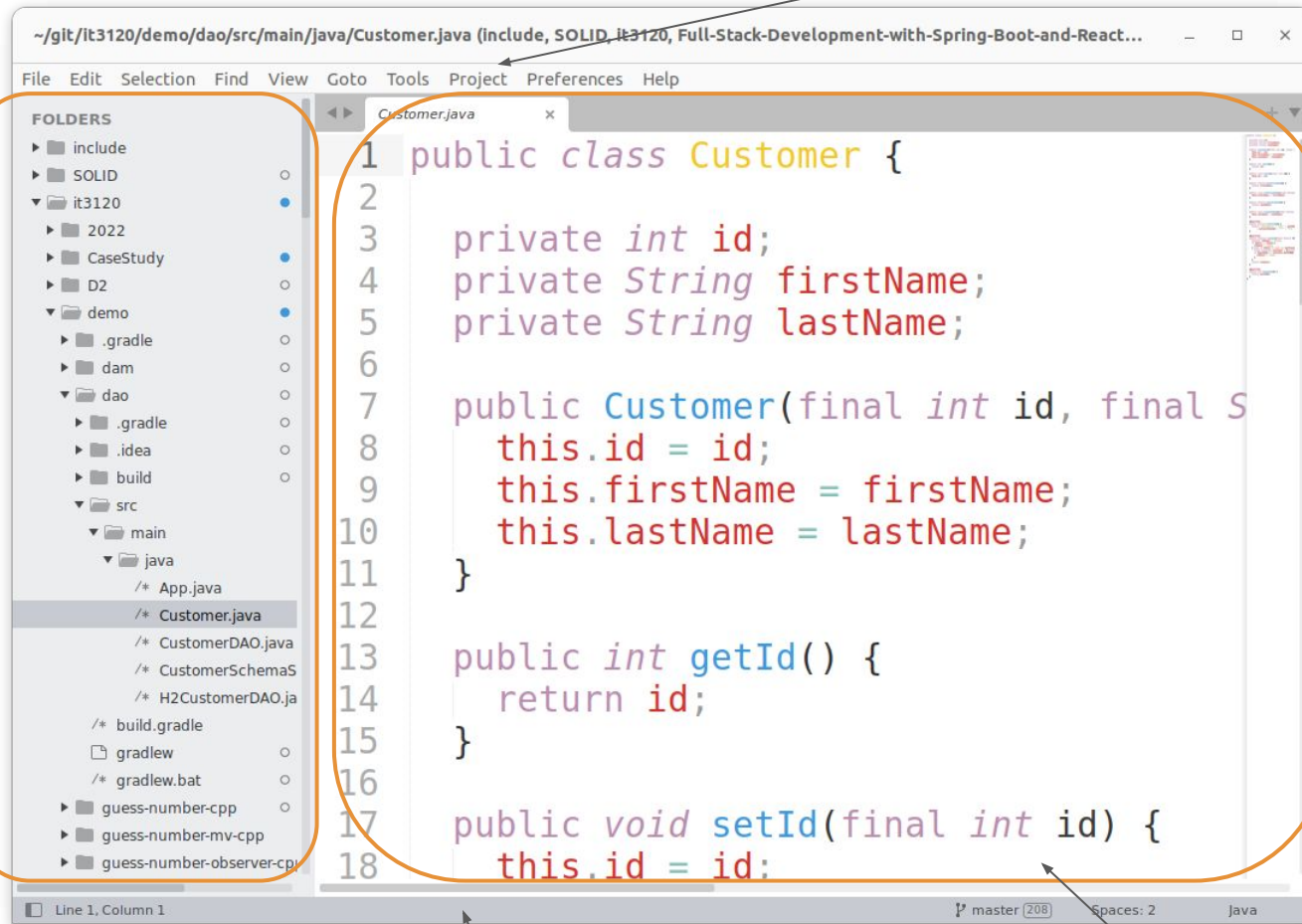
- Bố cục giao diện - Cấu trúc tổng quát, xác định vị trí của các thành phần trên màn hình.
- Nhận biết nội dung - Khả năng giúp người dùng hiểu các thành phần thông tin có trong giao diện
- Mô tả chức năng - Giúp người dùng phán đoán hành vi hệ thống, gợi ý chức năng
- Tính thẩm mỹ - Đẹp đối với người dùng
- Kỹ năng người dùng - Người dùng có kỹ năng khác nhau
- Nhất quán - Tính thống nhất của 1 thứ ở nhiều nơi trong hệ thống
- Hạn chế lỗi thao tác - Người dùng có thể mắc lỗi thao tác
- Tối ưu thao tác - Giúp hoàn thành nhanh công việc

# Bố cục

- Bố cục / Layout: Xác định cách bố trí các thành phần trên màn hình
- Các thành phần được gom lại thành các vùng
- Các vùng có thể tiếp tục được gom thành vùng lớn hơn
- Các vùng được bố trí theo thứ tự tạo thành luồng liên mạch, tự nhiên
  - Thứ tự tự nhiên có thể thay đổi theo đối tượng người dùng, phụ thuộc vào các yếu tố văn hóa
  - Chúng ta đọc từ trái sang phải, từ trên xuống dưới
  - ... tuy nhiên cũng có những nơi đọc từ phải sang trái.

# Ví dụ bố cục giao diện

Các danh mục



Cây thư mục

Thanh trạng thái

Vùng soạn thảo

# Nhận biết nội dung

Giúp người dùng định vị và hiểu các thông tin đang xử lý

- Sử dụng tiêu đề cửa sổ, các nhãn vùng, nhãn tên trường dữ liệu v.v..
- Tách biệt trực quan các vùng
- Các danh mục (menu) ngữ cảnh
- V.v...



# Ví dụ nhận biết nội dung

Các đường dẫn giúp xác định tệp hiện hành

Bản đồ nội dung hỗ trợ định vị

The screenshot shows an IDE window with the title bar: `~/git/it3120/demo/dao/src/main/java/Customer.java (include, SOLID, it3120, Full-Stack-Development-with-Spring-Boot-and-React...`. The interface includes a menu bar (File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, Help), a file explorer on the left, a code editor in the center, and a content map on the right.

Annotations from the text labels point to specific parts of the IDE:

- An arrow from "Các đường dẫn giúp xác định tệp hiện hành" points to the file explorer on the left, which shows the project structure. The path `dao/src/main/java` is highlighted, and `Customer.java` is selected.
- An arrow from the same label points to the tab bar at the top, which shows two tabs: `Customer.java — dam/.../java` and `Customer.java — dao/.../java`.
- An arrow from "Bản đồ nội dung hỗ trợ định vị" points to the content map on the right side of the editor, which displays a hierarchical view of the code structure.

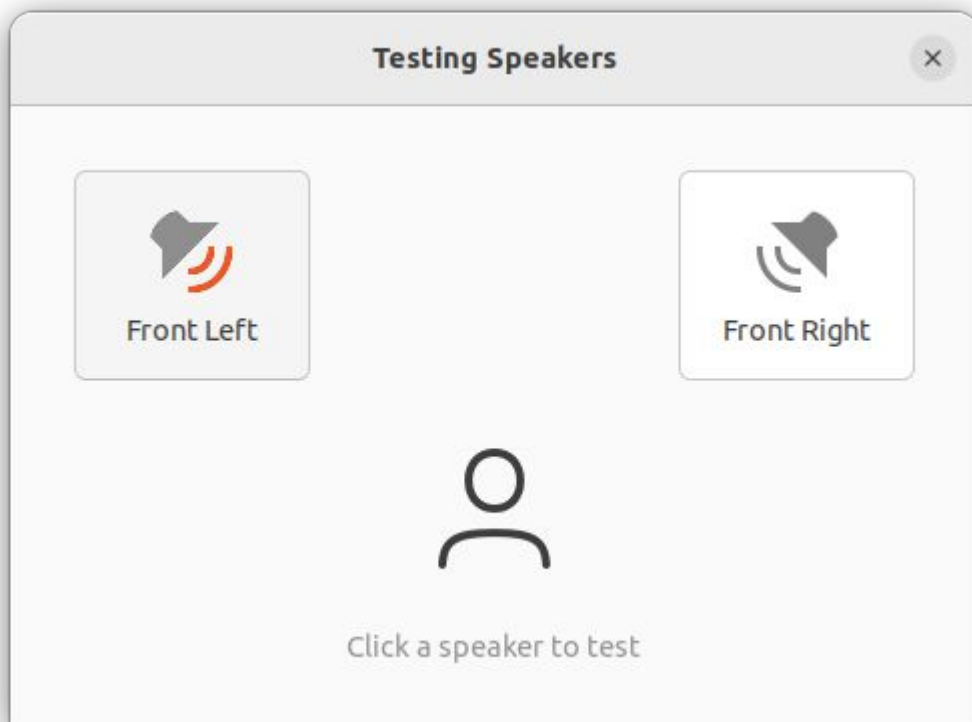
The code editor displays the following Java code:

```
1 public class Customer {
2
3     private int id;
4     private String firstName;
5     private String lastName;
6
7     public Customer(final int id, final S
8         this.id = id;
9         this.firstName = firstName;
10        this.lastName = lastName;
11    }
12
13    public int getId() {
14        return id;
15    }
16
17    public void setId(final int id) {
18        this.id = id;
```

The status bar at the bottom shows: `Line 1, Column 1`, `master (208)`, `Spaces: 2`, and `Java`.

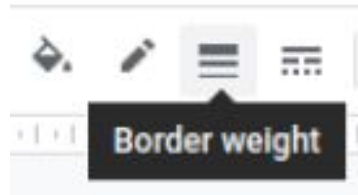
# Mô tả chức năng

- Sử dụng hình ảnh, biểu tượng, sơ đồ v.v.. giúp người dùng phán đoán chức năng và hành vi hệ thống
- Hiển thị trực quan và phản hồi trực quan thao tác người dùng



# Mô tả chức năng: Các gợi ý

- Gợi ý chức năng
  - Cung cấp cho người dùng thông tin về chức năng quan tâm
  - Kích hoạt: Khi giữ chuột trên nút bấm, lần đầu mở cửa sổ mới, v.v...



# Tính thẩm mỹ

- Các giao diện phải hữu dụng, tạo động lực, đồng thời phải có tính thẩm mỹ
- Thiết kế thường càng đơn giản càng tốt
- Các khoảng trắng có vai trò quan trọng để phân tách các phần nội dung
- Mật độ thông tin tỷ lệ với kỹ năng người dùng
  - Người dùng mới thường muốn mật độ mỏng (< 50%)
  - Chuyên gia thường muốn mật độ dày đặc, nhiều thông tin (> 50%)
- Phong cách và khả năng đọc văn bản: Kích thước, kiểu chữ serif (tỉ mỉ, hoa mỹ) hoặc sans serif (đơn giản, thân thiện), cách sử dụng chữ hoa
- Màu sắc và hoa văn
- v.v..

# Kỹ năng người dùng

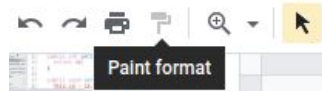
- Dễ học
  - Vấn đề đáng kể đối với người dùng chưa có kinh nghiệm
  - Rất quan trọng đối với hệ thống có nhiều người dùng
- Hiệu quả sử dụng
  - Vấn đề đáng kể đối với người dùng chuyên gia
  - Rất quan trọng trong các hệ thống chuyên dụng
- Tính dễ học có liên quan đến hiệu quả sử dụng
  - Tương hỗ: Dẫn đến những quyết định thiết kế tương tự
  - Mâu thuẫn: Người thiết kế có thể phải lựa chọn ưu tiên người dùng mới hoặc người dùng chuyên gia
    - ... Hoặc có thể hỗ trợ đồng thời nhiều giao diện cơ bản & nâng cao.

# Tính nhất quán

- Phạm vi
  - Trong 1 chương trình
  - Trong 1 nhóm chương trình
  - Giữa các phiên bản của 1 chương trình theo thời gian
  - Giữa các phiên bản của 1 chương trình trên nhiều nền tảng
- Đặc biệt quan trọng để đảm bảo tính đơn giản và dễ học
  - Giúp người dùng phán đoán hành vi
  - Có thể áp dụng kỹ năng học được ở 1 nơi cho những nơi khác
- Các yếu tố cơ bản
  - Những Lô-gic hoạt động chung
  - Điều khiển điều hướng
  - Thuật ngữ
  - Biểu tượng

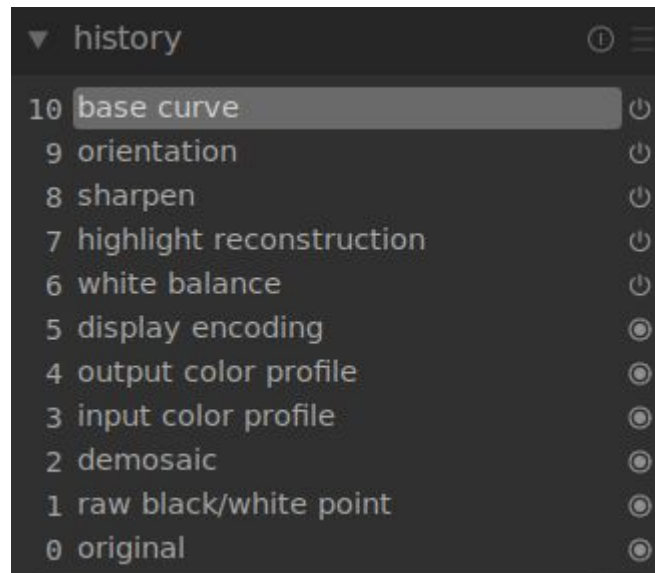
# Hạn chế lỗi thao tác

- Có thể vô hiệu các chức năng khi các điều kiện chưa được đáp ứng



*Cần bôi đen trước khi có thể sao chép định dạng*

- Bảo vệ kết quả
  - Ghi nhớ các thay đổi
  - Hoàn tác / Khôi phục trạng thái trước thao tác lỗi



# Tối ưu thao tác người dùng

- Các giao diện cần được thiết kế để tối ưu thao tác của người dùng
  - Giúp người dùng hoàn thành công việc nhanh nhất, với số thao tác tối thiểu.
  - Ở 1 góc độ nhất định có thể đánh giá độ phức tạp của thao tác người dùng thông qua số lần kích chuột
- Ví dụ quy tắc 3 lần kích chuột
  - Từ màn hình chính của hệ thống người dùng có thể tra cứu bất kỳ thông tin nào với không quá 3 lần kích chuột





# Tiến trình phát triển giao diện người dùng

*Lấy người dùng làm trung tâm, dẫn dắt bởi ca sử dụng, lặp và tăng dần:*

- Nghiên cứu ca sử dụng, các biểu đồ tuần tự, các biểu đồ hoạt động, tập trung vào người dùng và công việc của họ
- Biên soạn các kịch bản sử dụng, mô tả các bước thực hiện. *(Khi thực hiện bước này có thể phát hiện thêm các yêu cầu mới).*
- Thiết kế các giao diện dựa trên tập kịch bản sử dụng: Thiết kế cấu trúc và điều hướng, biên soạn quy chuẩn, tạo nguyên mẫu.

# Tiến trình phát triển giao diện người dùng<sub>(2)</sub>

- Kiểm tra, đánh giá khả năng đáp ứng yêu cầu sử dụng của thiết kế thu được và xác định những thứ cần sửa.
- Lập phát triển phiên bản thiết kế mới cho đến khi đáp ứng được yêu cầu sử dụng, không phát sinh thêm yêu cầu điều chỉnh.

# Biên soạn kịch bản sử dụng

- Kịch bản sử dụng mô tả các bước được thực hiện bởi người dùng để hoàn thành 1 nhiệm vụ / 1 phần công việc của họ
- Mỗi kịch bản sử dụng tương ứng với 1 luồng sự kiện trong 1 ca sử dụng thiết yếu.
  - Tuy nhiên không phải là bản sao 1-1
  - Được mô tả với các chi tiết về các công cụ tác nghiệp.
- Cần ghi chú các kịch bản thông dụng nhất / được sử dụng nhiều nhất để ưu tiên tạo những giao diện phù hợp, dễ dùng cho những tình huống đó.

# Ví dụ kịch bản sử dụng

## **Kịch bản Khách hàng lựa chọn dịch vụ**

1. Khách hàng yêu cầu Lập lịch di động
2. Hệ thống hiển thị các dịch vụ hiện đang được cung cấp
3. Khách hàng lựa chọn dịch vụ
4. Hệ thống hiển thị danh sách thời gian chờ và khả năng tiếp nhận
5. Khách hàng lựa chọn ngày và thời gian cho cuộc hẹn
6. Khách hàng yêu cầu cuộc hẹn theo ngày và giờ
7. Nếu cuộc hẹn được tìm thấy  
    Hệ thống hiển thị thông tin cuộc hẹn  
    Ngược lại  
    Hệ thống lặp các bước 5-7
8. Khách hàng xác nhận cuộc hẹn
9. Hệ thống hiển thị thông điệp xác nhận

# Thiết kế cấu trúc điều hướng

- Cấu trúc điều hướng thiết lập các thành phần giao diện với sự tập trung hướng vào điều khiển điều hướng
  - Các thành phần điều khiển điều hướng giúp người dùng di chuyển giữa các giao diện trong hệ thống
  - Các tương tác để đáp ứng nhu cầu sử dụng
- Đồng thời thông báo về kết quả thực hiện các hành động: Thành công, thất bại, đang được thực hiện, v.v..
- Được biểu diễn bằng biểu đồ điều hướng cửa sổ và các ca sử dụng thực tế
- Các nguyên lý cơ bản:
  - Thiết kế đơn giản
  - Sử dụng 1 trật tự ổn định cho các thành phần (ví dụ, Tập > Tạo mới -- Mở vs. Tập > Mở -- Tạo mới)

# Các hình thức điều khiển điều hướng

- Các nút bấm
- Danh mục / Menus
  - Người dùng được cung cấp 1 danh sách lựa chọn
  - Có nhiều hình thức (ví dụ, thanh menu, menu ngữ cảnh, v.v..)
- Vuốt và chạm (đối với màn hình cảm ứng)
- Ngôn ngữ
  - Ngôn ngữ lệnh - Người dùng sử dụng hệ lệnh của hệ thống
  - Ngôn ngữ tự nhiên - Hệ thống trình diễn ngôn ngữ tự nhiên
- Điều khiển bằng giọng nói
- v.v..

# Các thông báo

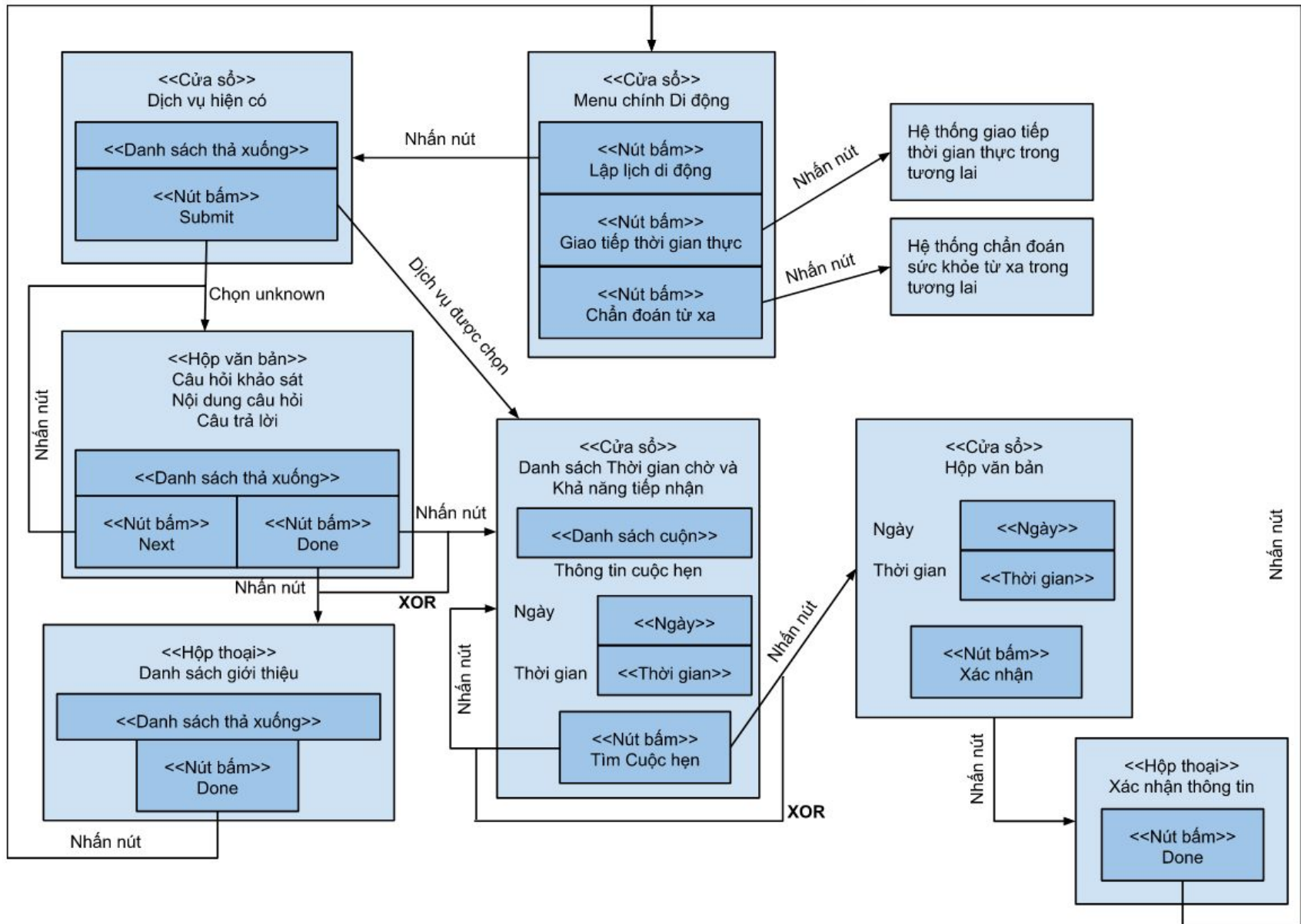
- Thông báo lỗi - Cung cấp thông tin về lỗi phát sinh
  - Ví dụ, Có trường dữ liệu bắt buộc bị để trống
- Thông báo xác nhận - Hạn chế lỗi / Đảm bảo chắc chắn đối với các thao tác quan trọng
  - Ví dụ, Bán có chắc muốn xóa?
- Thông báo kết quả thành công
  - Ví dụ, đơn hàng đã được tạo
- Thông báo trạng thái
  - Ví dụ, đã hoàn thành 90% công việc
- Các thông điệp trợ giúp - Cung cấp thông tin bổ xung để hỗ trợ người dùng thực hiện công việc.
- V.V...

# Biểu đồ điều hướng cửa sổ

- Biểu đồ điều hướng cửa sổ / Windows Navigation Diagram (WND)
- Tương tự biểu đồ máy trạng thái hành vi
- Biểu diễn mối quan hệ giữa các giao diện: Màn hình, biểu mẫu, báo cáo, v.v.. trong hệ thống cần sử dụng để hoàn thành 1 công việc
  - Có thể cần tạo nhiều WND cho 1 hệ thống.
- Biểu diễn cách người dùng di chuyển từ 1 giao diện tới giao diện khác
- Các hộp biểu diễn các thành phần, được phân biệt bằng nhãn loài (stereotype)
- Mũi tên biểu diễn các bước chuyển giao diện



# Ví dụ WND



# Các ca sử dụng thực tế

- Phụ thuộc vào triển khai: Mô tả chi tiết cách sử dụng hệ thống sau khi được triển khai
- Các ca sử dụng thiết yếu được phát triển thành thực tế bằng cách mô tả chi tiết giao diện người dùng thực tế
- Đối với ứng dụng đa nền tảng, ví dụ, máy bàn, máy tính bảng, và điện thoại thông minh, ca sử dụng thực tế cần được phát triển cho từng nền tảng mà ca sử dụng được triển khai.

# Biên soạn quy chuẩn giao diện

- Quy chuẩn giao diện bao gồm các thành phần giao diện cơ bản được sử dụng thống nhất trong toàn hệ thống
- Các thành phần tiêu biểu thường được chuẩn hóa:
  - Khuôn mẫu giao diện: Bố cục tổng quan, Gam màu, lô-gic sắp xếp các cửa sổ, v.v..
  - Biểu diễn đối tượng:
    - Tên của các thứ trong giao diện
    - Thuật ngữ cho các hành động
    - Các biểu tượng giao diện, ví dụ biểu tượng trên các nút bấm, v.v.

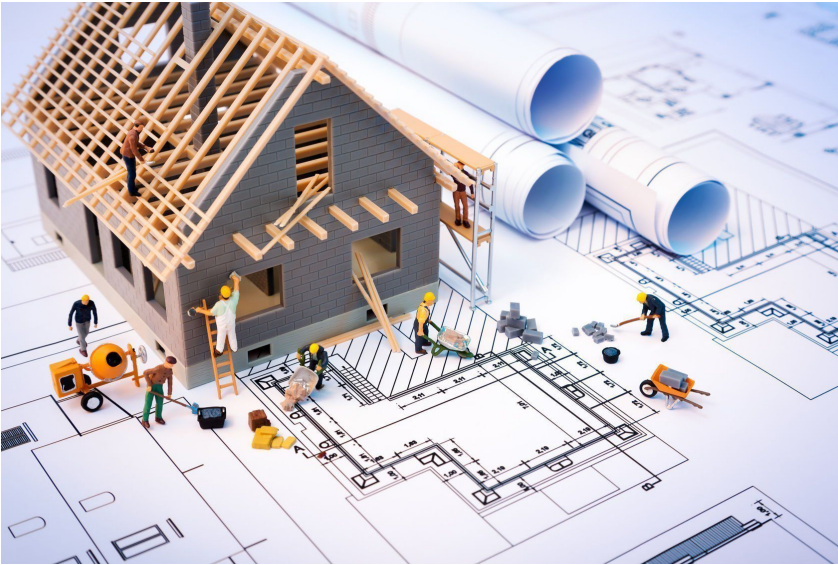


- Các hình tượng tương tác có thể được sử dụng để gợi ý về hành vi, tạo cho người dùng cảm giác như đang tương tác với các thứ trong thế giới thực
  - Ví dụ, giỏ hàng trong hệ thống bán hàng được sử dụng để lưu sản phẩm được chọn vs. Giỏ hàng trong siêu thị

# Các hình tượng trong tương tác người-máy

- Hình tượng thao tác trực tiếp
  - Hình tượng trong đó các đối tượng được hiển thị giống như đối tượng vật lý (bằng hình ảnh / biểu tượng)
- Hình tượng bàn làm việc
  - Màn hình được bố cục thành các vùng, với 1 không gian làm việc rộng ở chính giữa và tập hợp các công cụ được bố trí trên biên.
- Hình tượng tài liệu
  - Hình tượng trong đó dữ liệu được biểu diễn trực quan như các trang giấy hoặc biểu mẫu
- Hình tượng hội thoại
  - Hình tượng trong đó người dùng và máy tính tương tác giống như đang thảo luận thông qua các giao diện.

# Thao tác trực tiếp vs. Thao tác gián tiếp



# Ví dụ hình tượng thao tác trực tiếp

- Giỏ hàng lưu các sản phẩm được chọn



- Kéo tệp vào (biểu tượng) thùng tái chế để xóa



- Các công cụ vẽ



- V.V....

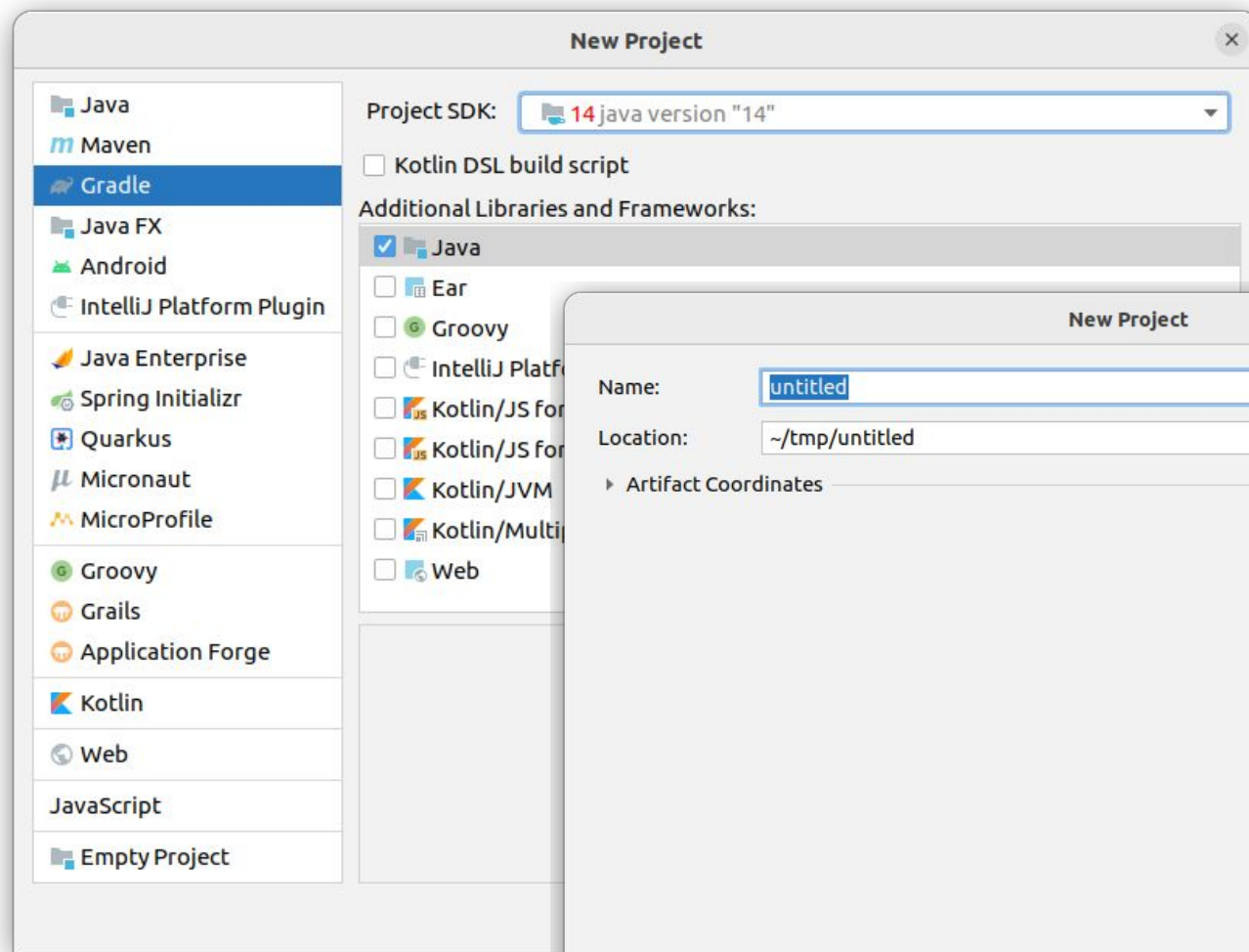


**Hình 9-G. Ảnh xạ lập lĩnh vực sang bản ORDBMS đã cập nhật**

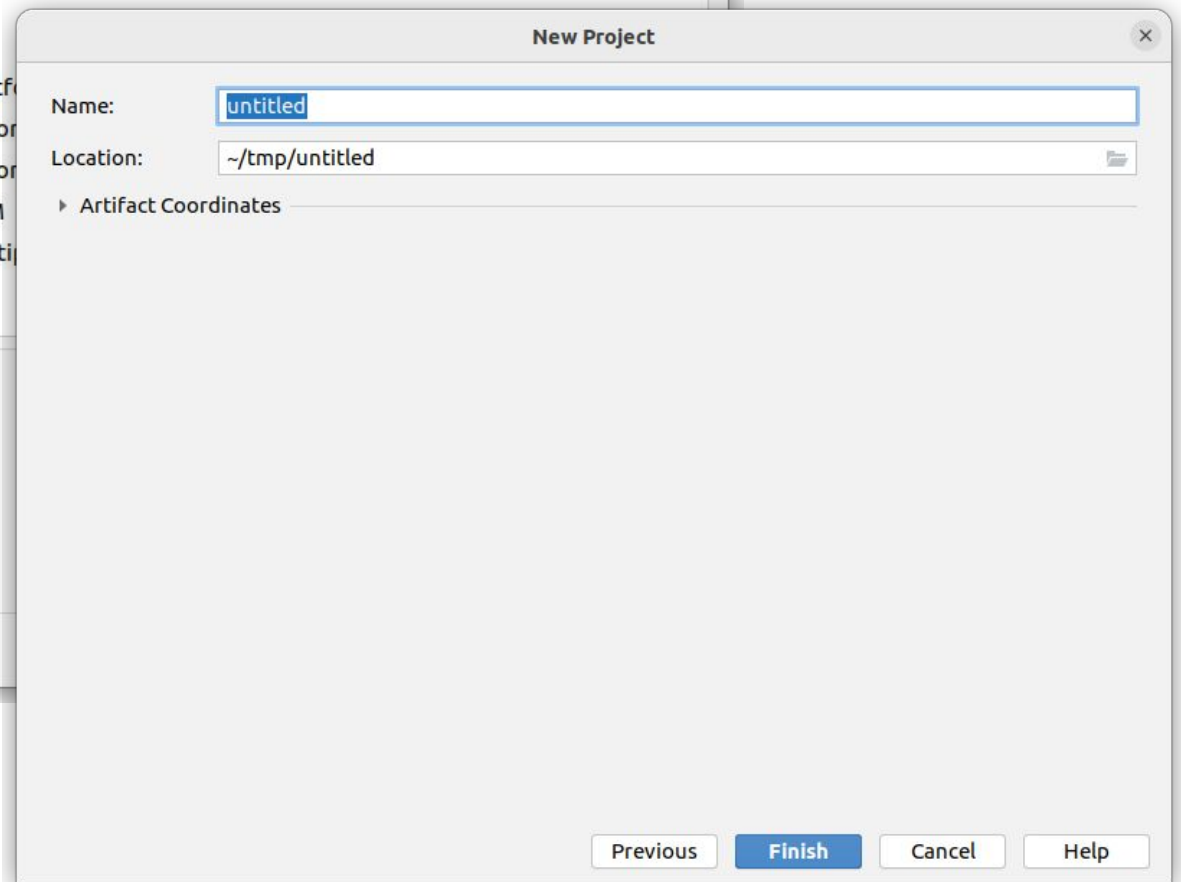
Thứ 2, chỉ ra rằng, với thêm 1 chút công sức, tất cả các bảng ORDBMS đã thiết kế có thể được ánh xạ sang 1 tập các bảng RDBMS. Thực tế là, tập các bảng RDBMS đã được sử dụng trong hệ thống phòng khám sức khỏe hiện có. Tuy nhiên sau khi thảo luận kỹ với Ruby và Ben, Jo đồng ý vì cả Hệ thống bán thuốc theo đơn và Hệ thống phân phối dịch vụ phòng khám sức khỏe tích hợp đều là các sản phẩm chiến lược hướng tới nâng cao lợi thế của Patterson trong thị trường di động, tách riêng ảnh xạ ORDBMS và sử dụng hệ thống mới như sự khởi đầu để chuyển đổi hệ thống cũ từ RDBMS sang ORDBMS là hợp lý. Hơn nữa, Jo đồng ý để đội của cô chuyển đổi các bảng trong RDBMS cũ sang các bảng ORDBMS và chuyển đổi hệ thống cũ để khai thác lợi thế của thay đổi này. Nhưng tạm thời, cô ấy đề nghị nhóm CSDL của cô ấy trước tiên cần tạo 1 tập các bảng ORDBMS tạm thời có thể ánh xạ tới

# Ví dụ hội thoại

Bạn muốn tạo loại dự án nào?



Bạn muốn lưu các tệp ở đâu?





# Tạo nguyên mẫu giao diện

- Các giao diện có thể được:
  - Vẽ / phác thảo bằng các công cụ vẽ thông thường.
  - Tạo bằng các phần mềm chuyên dụng cho phép giả lập các thao tác chuyển giao diện.
  - Tạo bằng ngôn ngữ lập trình như trong pha thực thi:
    - Tuy nhiên chưa có các xử lý thực tế
    - Ví dụ tạo nguyên mẫu trang Web với HTML, CSS, Javascript, và dữ liệu mô phỏng
    - Gắn nhất với giao diện khi triển khai thực tế
- Các giao diện có thể được tập hợp và bố trí theo cấu trúc điều hướng tạo thành các bảng phân cảnh / Storyboard
  - Công cụ hữu ích giúp người dùng hình dung về hoạt động của hệ thống thực tế (trong tương lai)

# Ví dụ phác thảo nguyên mẫu giao diện



# Thiết kế nhập

- Các biểu mẫu được sử dụng để nhập dữ liệu
- Dữ liệu có thể:
  - Có cấu trúc: Ngày, sản phẩm, v.v...
  - Phi cấu trúc: Bình luận, mô tả
- Các nguyên lý cơ bản
  - Nhận dữ liệu từ nguồn (ví dụ Mã vạch vs. RFID)
  - Xử lý trực tuyến vs. xử lý theo gói
  - Tối ưu số thao tác nhập (sử dụng giá trị mặc định cho những giá trị được dùng thường xuyên)

# Các loại đầu vào

- Các hộp nhập dữ liệu:
  - Hộp nhập văn bản
  - Hộp nhập số, có thể hỗ trợ dạng tự động
    - Ví dụ: Nhập số điện thoại
  - Hộp mật khẩu ẩn các ký tự với các dấu \* và không cho cắt dán hoặc sao chép
  - v.v..
- Các hộp lựa chọn
  - Hộp đánh dấu / Checkbox, có thể lựa chọn nhiều mục
  - Hộp lựa chọn loại trừ / Radio box, có thể chọn 1 mục trong 1 nhóm.
  - Hộp danh sách biểu diễn 1 danh sách lựa chọn
  - Thanh kéo / Slider - Có thể di chuyển con trỏ trên 1 thang tương ứng với 1 khoảng giá trị.
  - v.v..

# Kiểm tra đầu vào

- Dữ liệu cần được kiểm tra trước khi được đưa vào hệ thống để đảm bảo tính đúng đắn
- Không tiếp nhận dữ liệu không hợp lệ
- Các kiểm tra tiêu biểu:
  - Định dạng
  - Khoảng giá trị
  - Chữ số kiểm tra - Giúp giảm lỗi nhập số
    - Ví dụ chữ số cuối trong mã vạch
  - Tính nhất quán với các trường dữ liệu liên quan
  - Các ràng buộc trong CSDL

# Thiết kế xuất

- Các báo cáo được tạo từ dữ liệu được xuất bởi hệ thống
- Các nguyên lý cơ bản:
  - Mục đích và tần suất sử dụng ảnh hưởng đến bố cục của báo cáo
  - Quản lý lượng thông tin trong 1 báo cáo - Chỉ cung cấp những gì cần thiết và đặt những thông tin quan trọng nhất ở trên đầu
  - Giảm thiểu độ lệch, đặc biệt trong thiết kế đồ họa (biểu đồ)

# Các loại đầu ra

- Báo cáo chi tiết - Người dùng cần thông tin đầy đủ
- Báo cáo ngắn gọn - Các chi tiết được tổng hợp (ví dụ, tổng, trung bình)
- Báo cáo ngoại lệ
- Tài liệu xoay vòng - Đầu ra quay lại thành đầu vào
- Biểu đồ - Để thuận tiện so sánh trực quan
- Phương tiện xuất báo cáo có thể là điện tử (xem trên màn hình) hoặc bản cứng (in trên giấy)

# Kiểm tra giao diện

- Lý tưởng, đánh giá các giao diện khi hệ thống đang được thiết kế - Trước khi bắt đầu xây dựng nó
  - Giúp phát hiện và giải quyết sớm các vấn đề
  - Người dùng có thể muốn thay đổi giao diện sau khi nhìn thấy nó.
- Huy động tối đa các thành viên tham gia



# Các phương pháp đánh giá giao diện

- Theo kinh nghiệm - Đối chiếu thiết kế với các nguyên lý và quy tắc đã biết
- Thuyết trình & đánh giá - Nhóm thiết kế trình diễn nguyên mẫu với người dùng và giải thích lô-gic hoạt động
- Tương tác - Người dùng thử sử dụng nguyên mẫu và thảo luận trực tiếp với các thành viên đội dự án.
- Kiểm thử kín - Người dùng thử sử dụng nguyên mẫu ngôn ngữ trong phạm vi phòng thí nghiệm.

# Các yêu cầu phi chức năng

*Thiết kế giao diện có thể bị ảnh hưởng bởi các yêu cầu phi chức năng*

- Các yêu cầu vận hành - Lựa chọn nền tảng phần cứng và phần mềm
  - Các công nghệ có thể được sử dụng (Giao diện đồ họa, chuột, v.v..)
- Yêu cầu hiệu năng
  - Thiết bị di động và trình duyệt hạn chế hiệu năng
- Các yêu cầu bảo mật
  - Kiểm soát đăng nhập và có thể mã hóa
  - Mạng không dây dễ bị tấn công
- Các yêu cầu văn hóa
  - Định dạng ngày, sử dụng màu sắc, v.v..

