

# 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

# Nội dung

- Các nguyên lý thiết kế (SOLID)
- Thiết kế lưu trữ cố định

# 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]

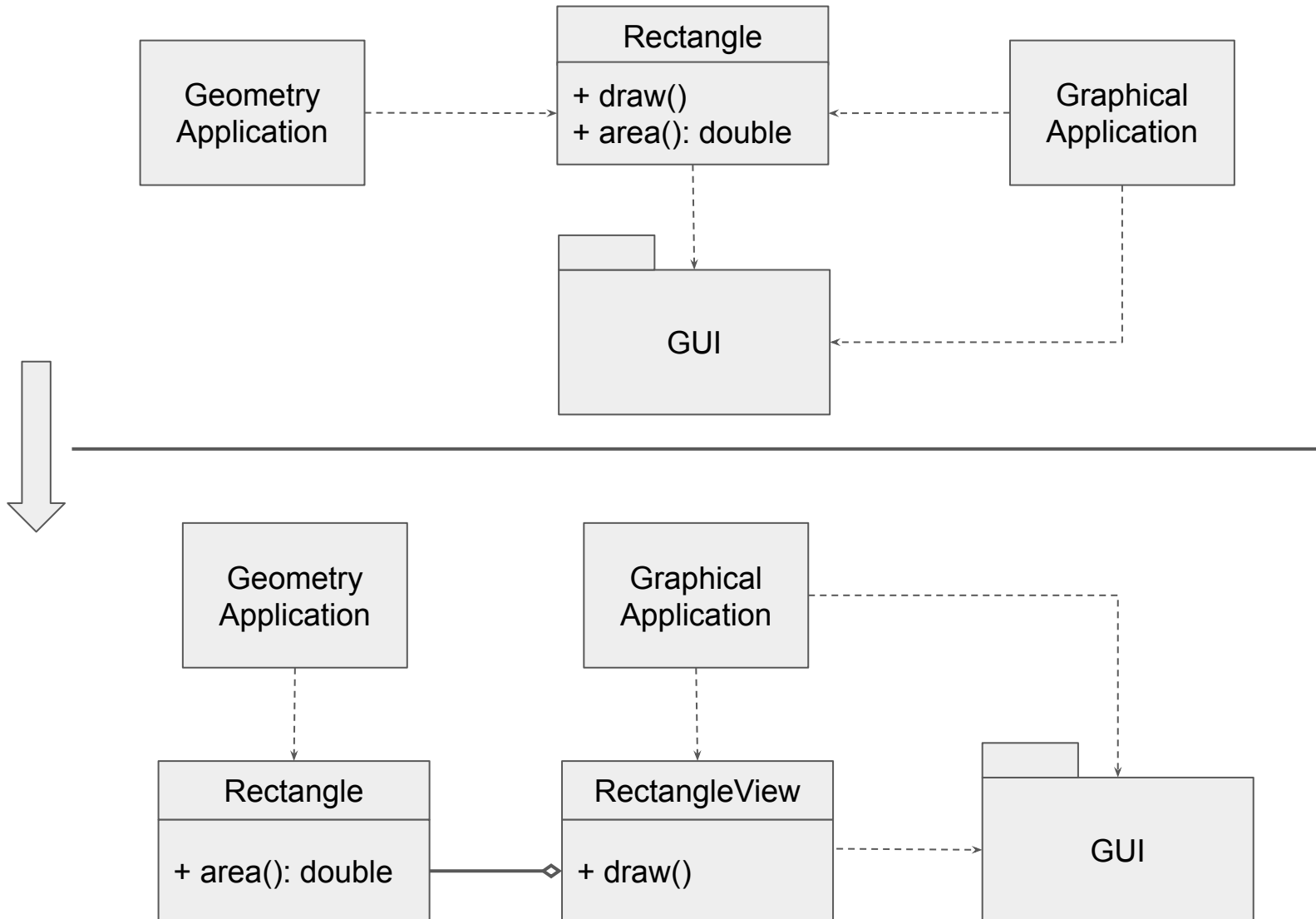
- Trách nhiệm được gán cho lớp
  - (xem thẻ CRC)
- Nếu phát sinh nhu cầu triển khai chức năng mới, hoặc chức năng đang có thay đổi, thì các trách nhiệm của các lớp phải thay đổi.
- Lớp có 1 trách nhiệm sẽ chỉ có 1 lý do để thay đổi.

*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]*

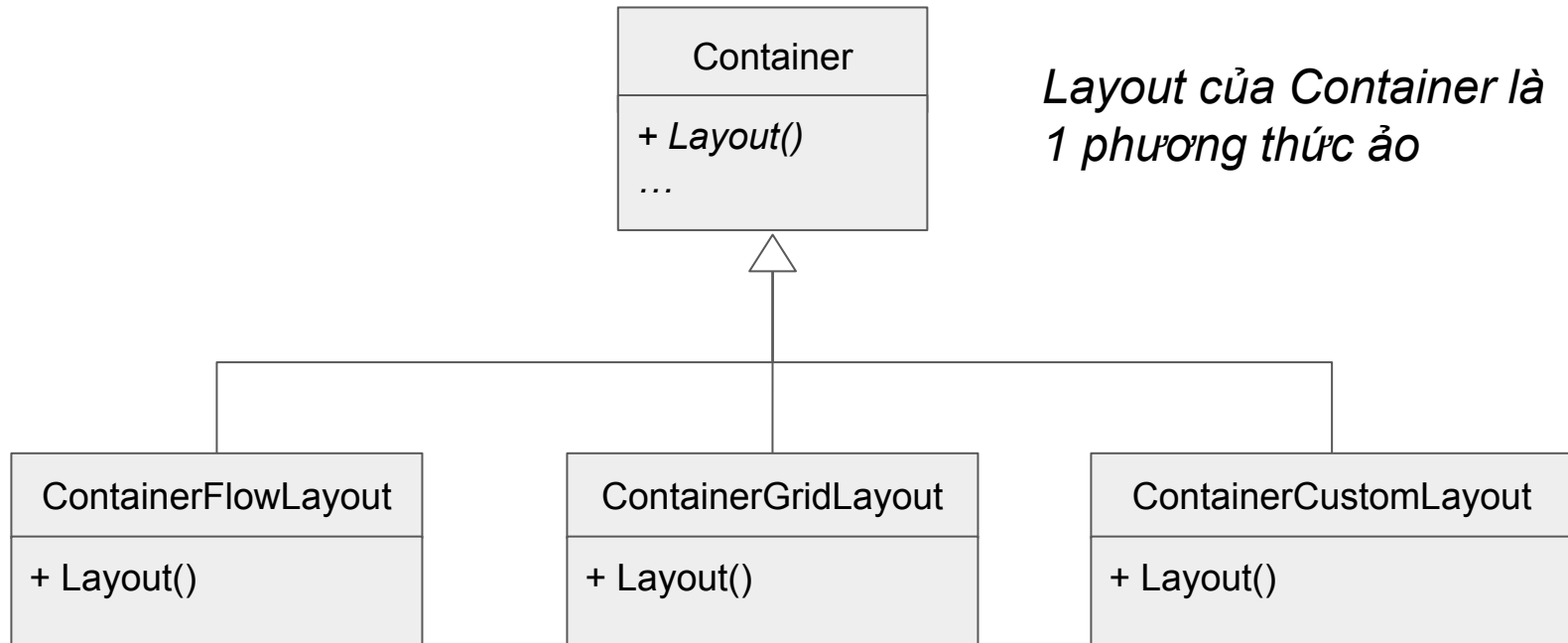
- Mở rộng hành vi của 1 thành phần:
  - Khi yêu cầu ứng dụng thay đổi, chúng ta có thể mở rộng thành phần đã có, thêm vào các hành vi mới.
- Không thay đổi mã nguồn đã có:
  - Mã nguồn đã có được giữ nguyên trong quá trình bổ xung hành vi mới.
  - Đem lại nhiều lợi ích trong quá trình phát triển.



# Trừu tượng hóa là chìa khóa

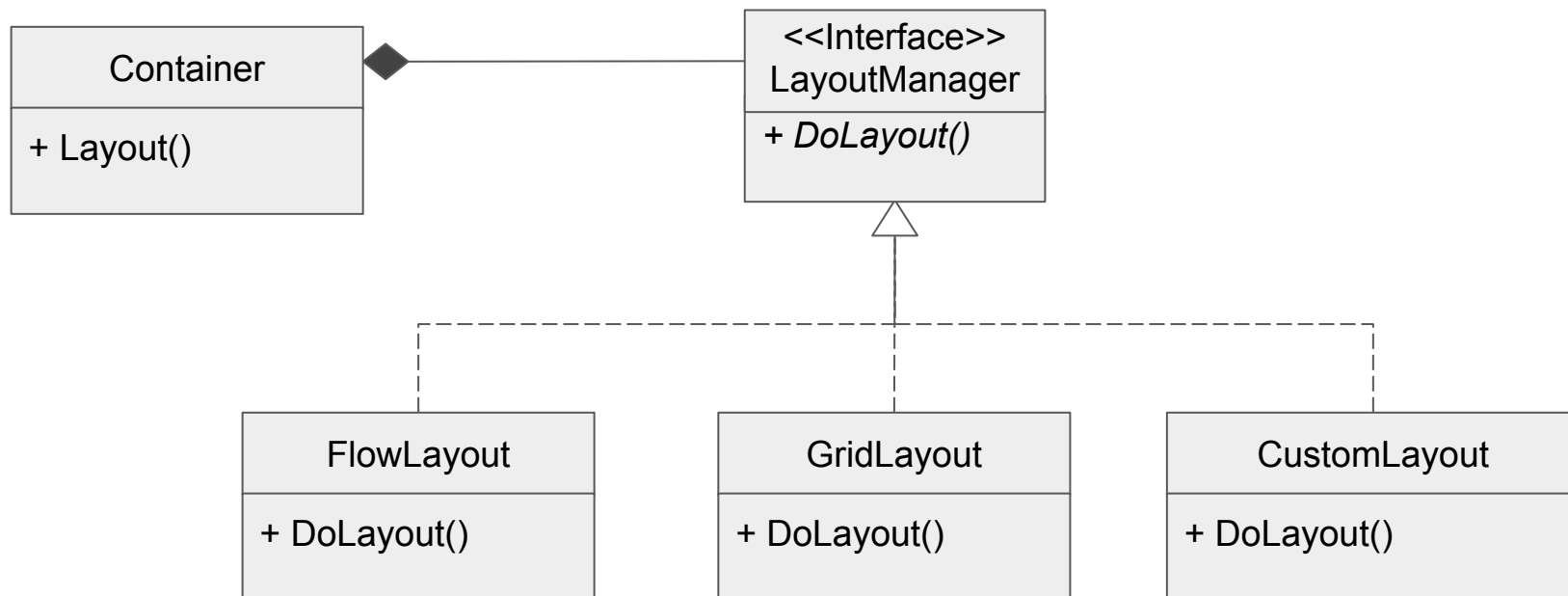
- Để mở rộng một thành phần mà không thay đổi nó, thì phần thay đổi trong triển khai của thành phần đó phải được trừu tượng hóa.
- Lập trình hướng đối tượng: Lớp ảo, giao diện, phương thức ảo, kế thừa, v.v..
- Lập trình hàm: Kiểu dữ liệu khái quát, hàm khái quát, v.v..
- *(Khả năng tránh thay đổi có 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



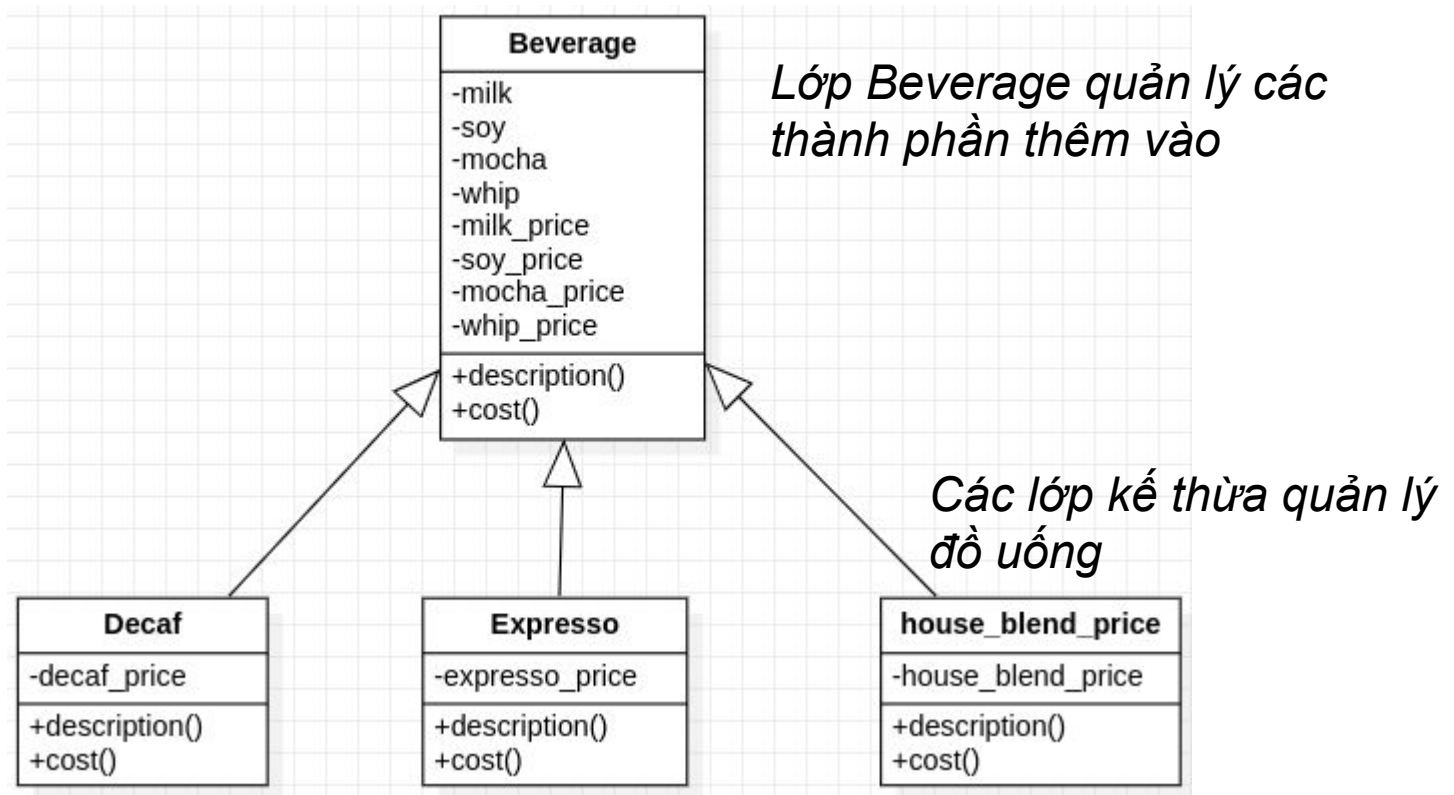
Các lớp con sẽ cung cấp triển khai cụ thể của phương thức Layout

# Khái quát hóa dựa trên tổng hợp



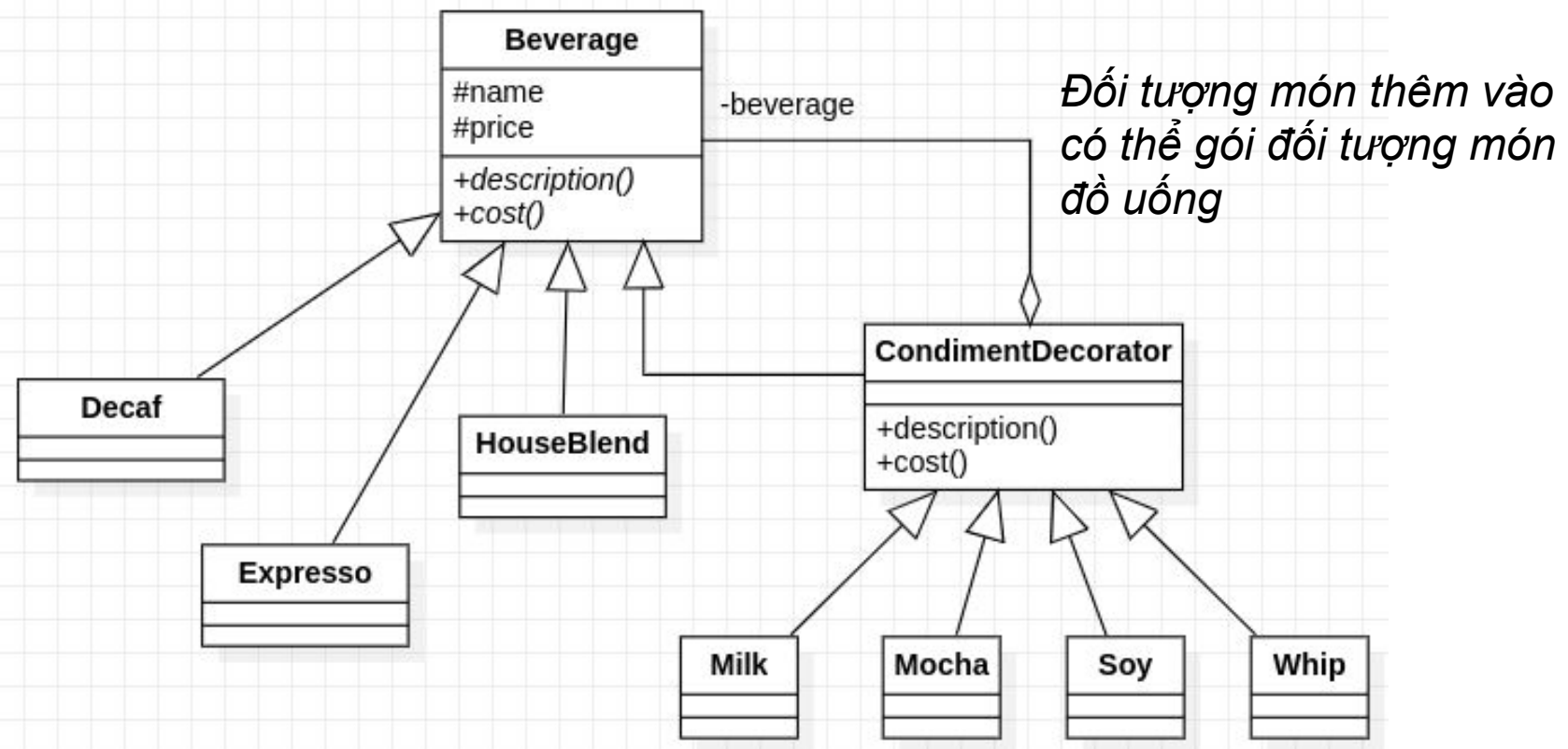
*Chúng ta có thể thay đổi hành vi của Container bằng cách thiết lập 1 đối tượng quản lý bố cục phù hợp (kế thừa LayoutManager)*

# 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

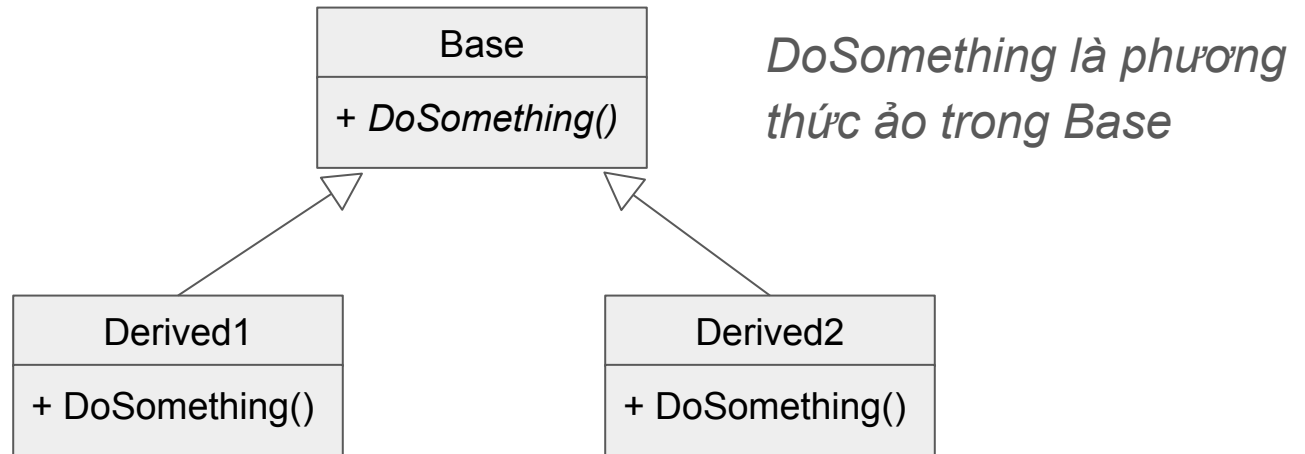
- Nguyên lý khả thay Liskov / Liskov Substitution Principle (LSP)

*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]*

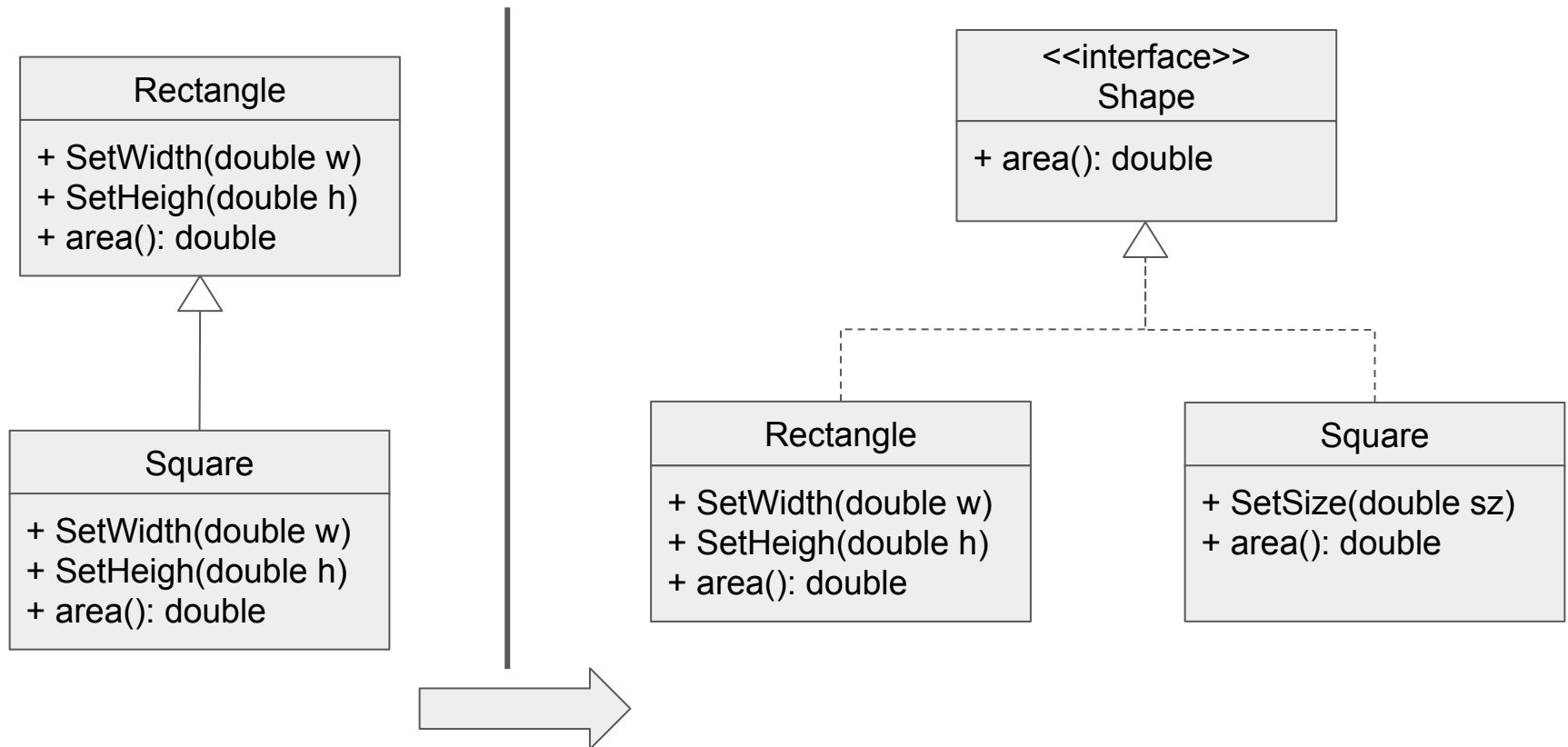
- Giống như dấu hiệu chất lượng thiết kế quan hệ kế thừa
- Hỗ trợ hạn chế các vấn đề có thể phát sinh khi lập trình với các giao diện khái quát.

# 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

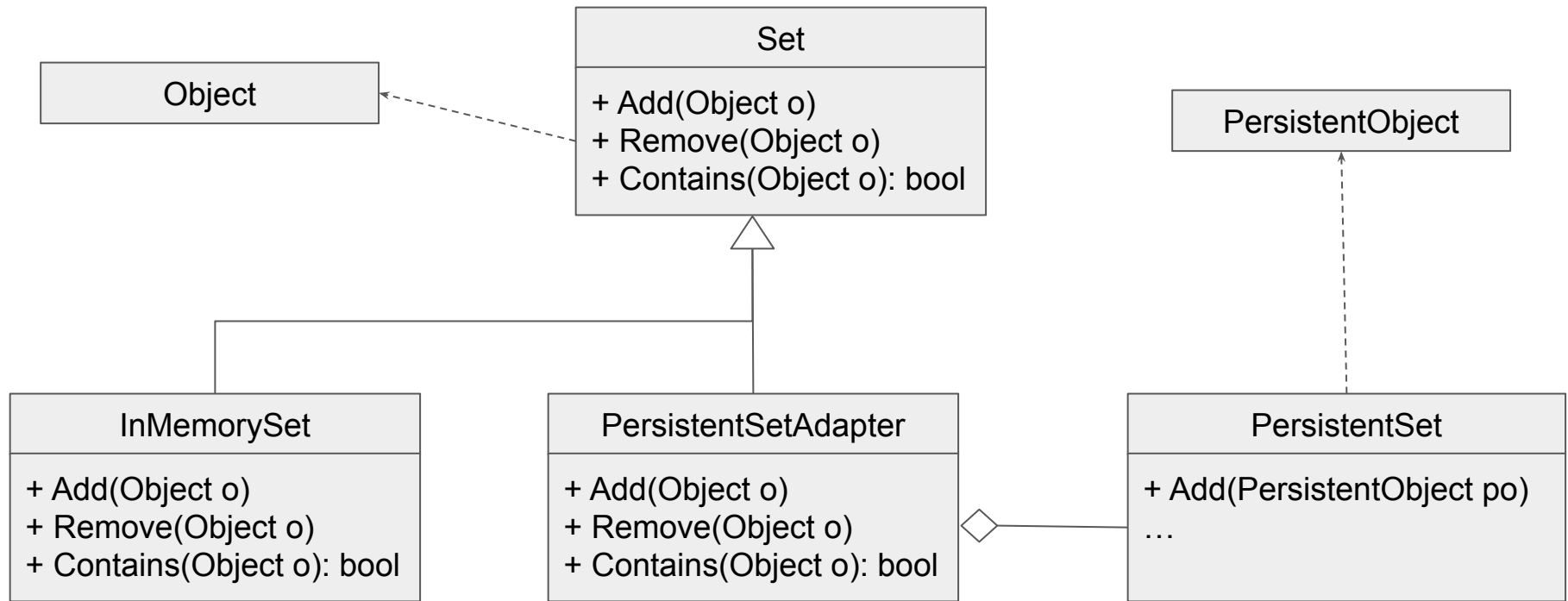


*Không tương thích LSP,  
SetHeigh và SetWidth có ý  
nghĩa khác nhau trong  
Rectangle và Square.*

*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

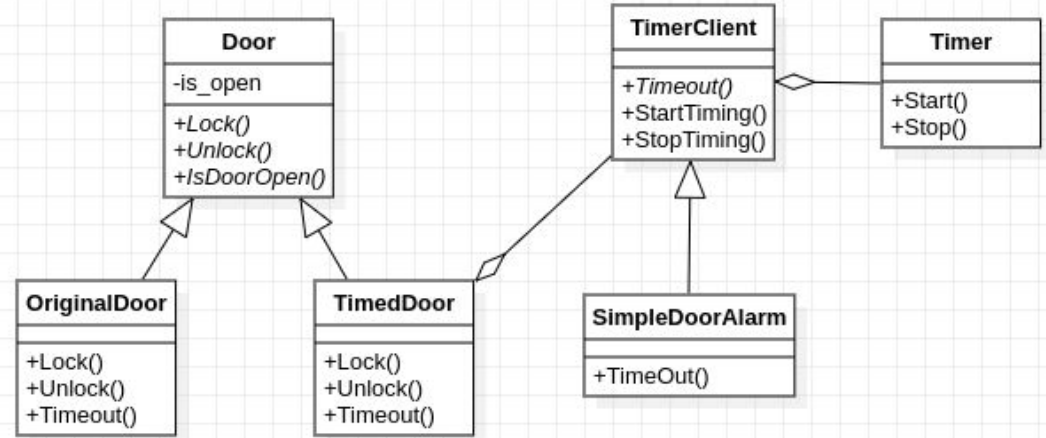
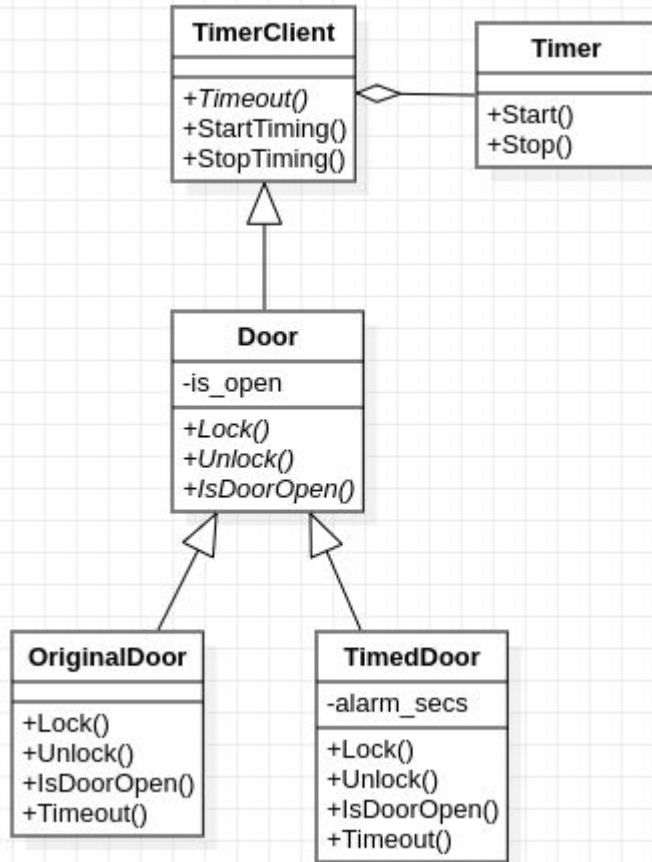
- Nguyên lý phân tách giao diện / Interface Segregation Principle (ISP)

*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.*

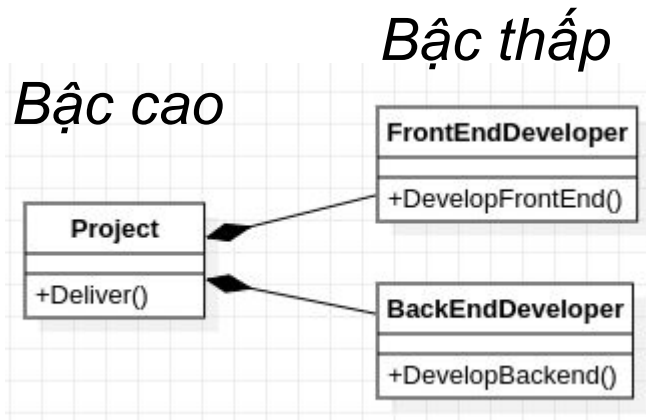
*Abstractions should not depend upon details. Details should depend upon abstractions.*

[Robert C. Martin]

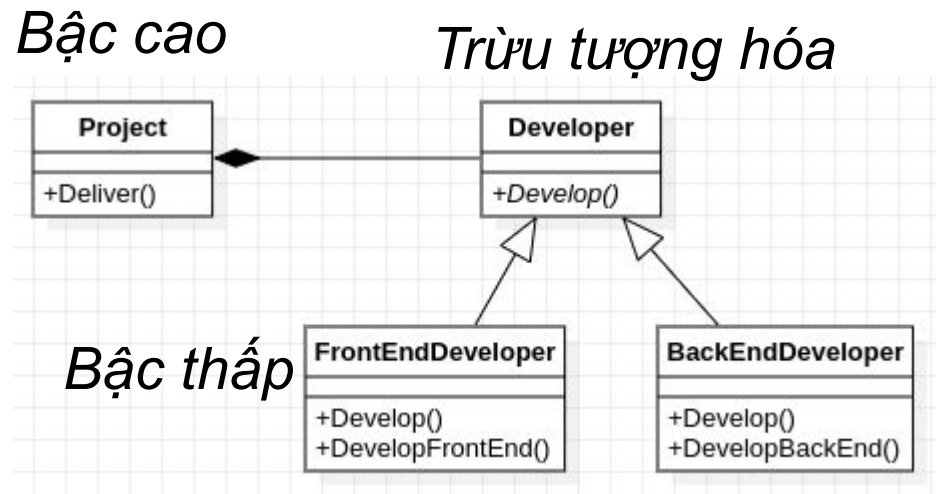
# Nguyên lý đảo ngược phụ thuộc<sub>(2)</sub>

- 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*.



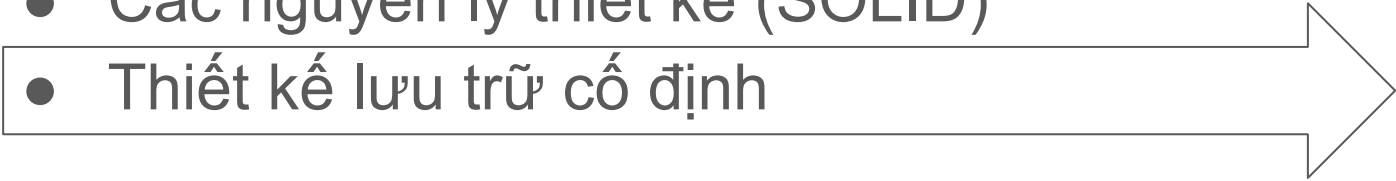
Triển khai cụ thể phụ thuộc vào cơ chế trừu tượng.

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
- 



# 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 cha, 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.

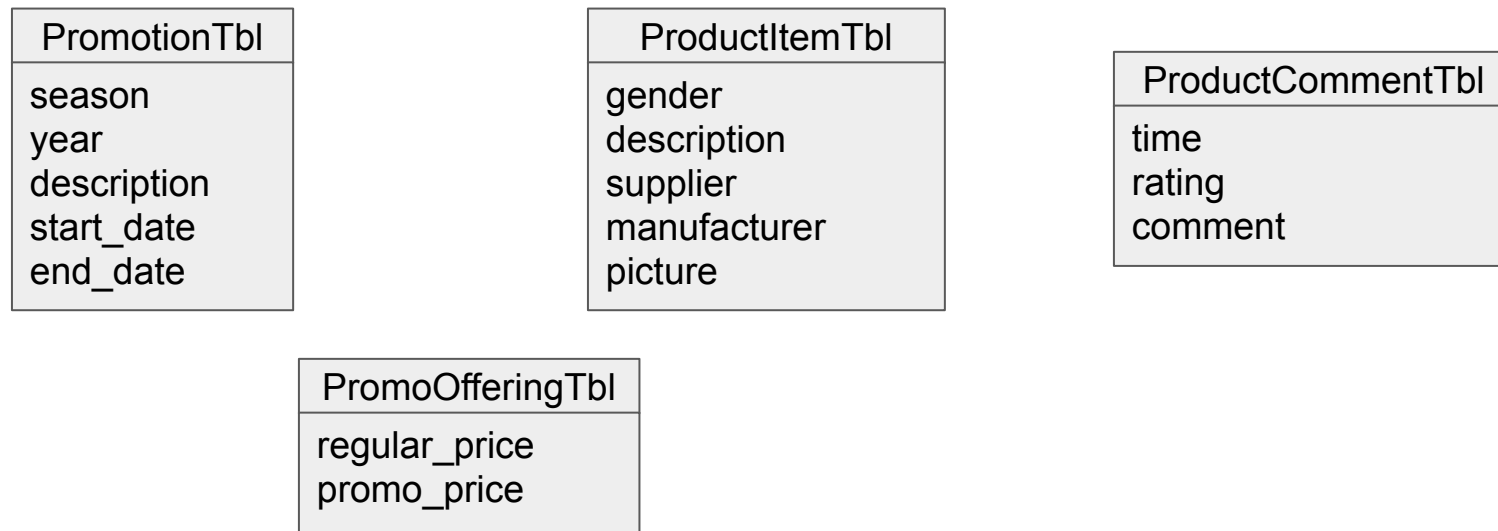
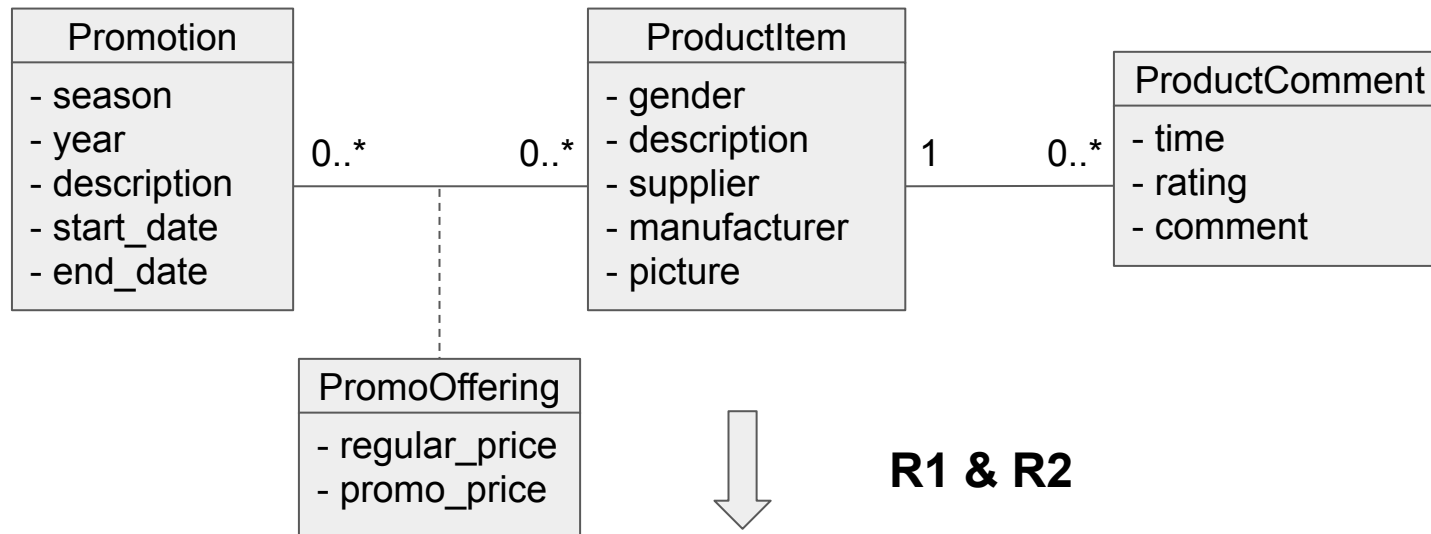
Đối với quan hệ Khái quát hóa/Kế thừa:

**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

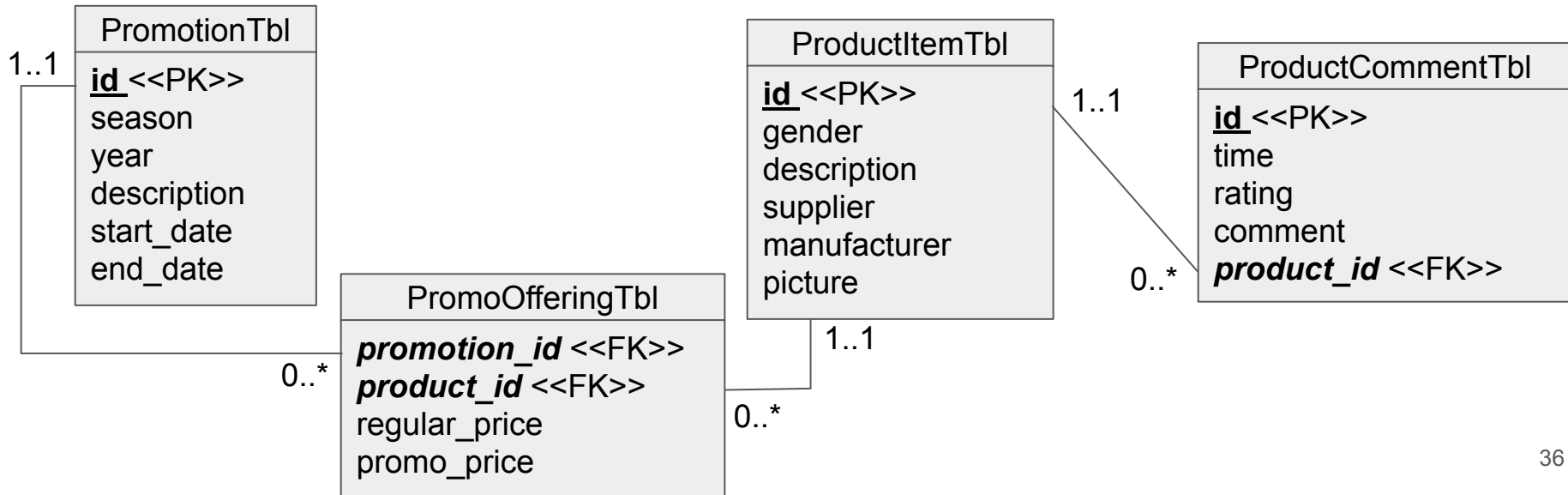
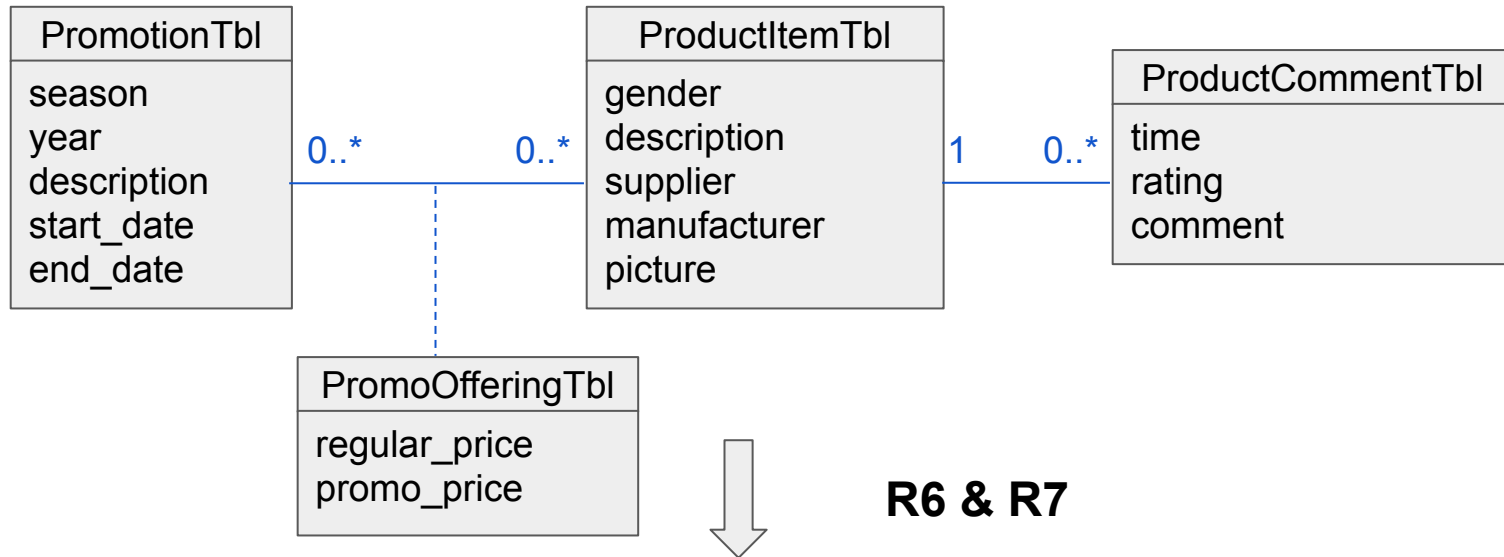
**R8b.** Mở rộng lớp con bằng cách sao chép thuộc tính của lớp cha vào tất cả các lớp con là xóa lớp cha khỏi thiết kế. Hoặc

**R8c.** Làm phẳng cây kế thừa bằng cách sao chép tất cả các thuộc tính trong cây vào lớp cha, 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		

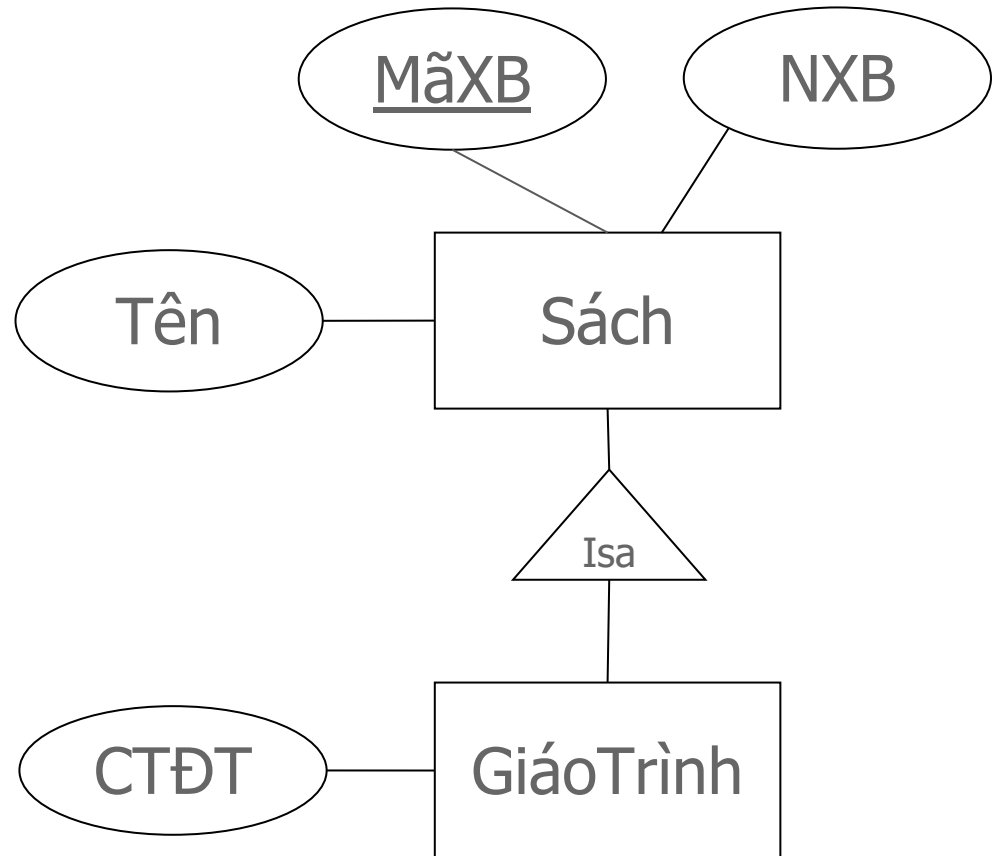
...

# 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	<u>Mã XB</u>
LT HĐT	KHKT	K123
Tin ĐC	BKB	B111

Giáo trình

CTĐT	<u>Mã XB</u>
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

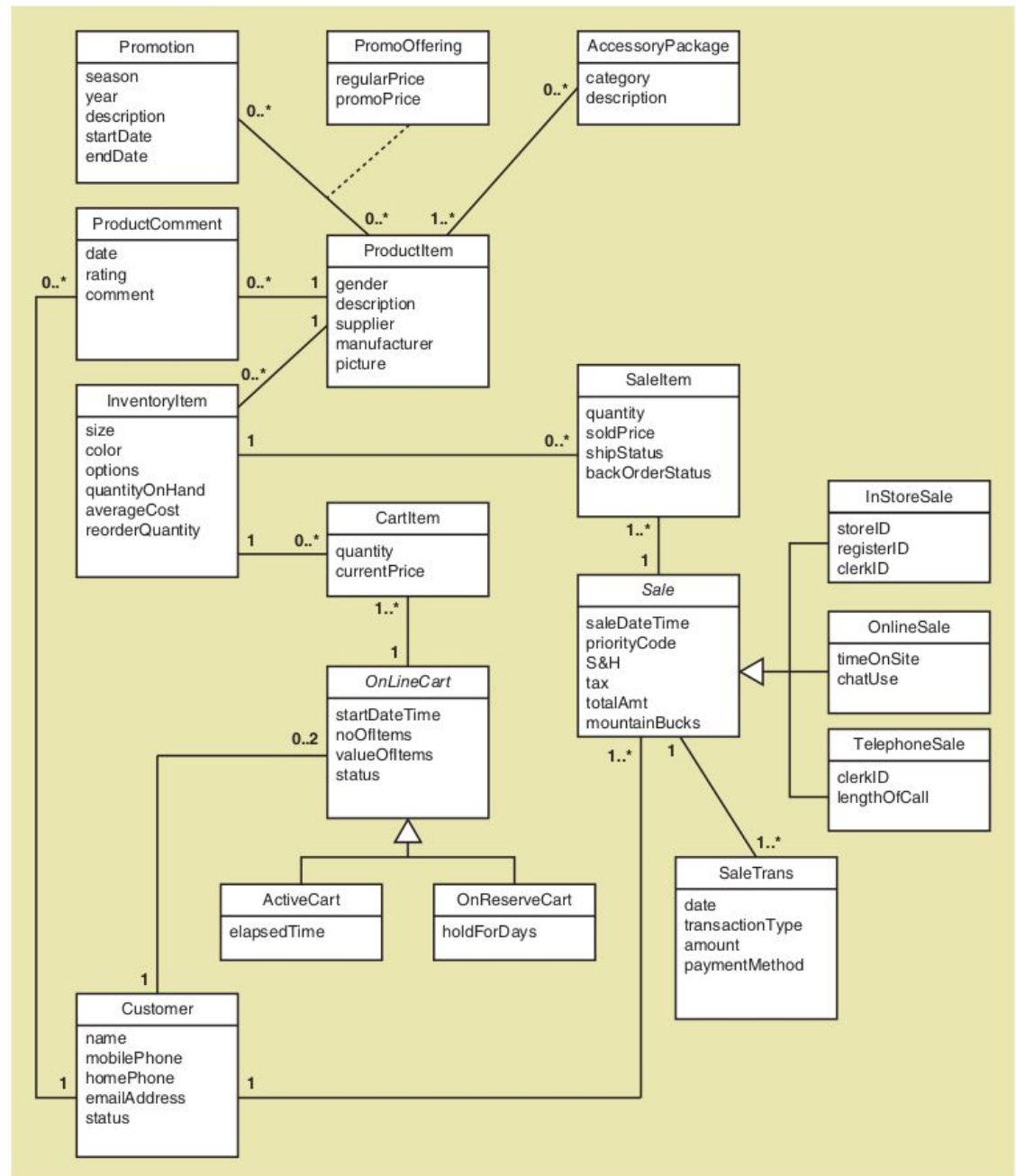
  

ProductItemID	Gender	Description	Supplier	Manufacturer	Picture
10564	Both	Super Alpine Performance Skis	K2	K2	
10766	Man	Extreme Ski Boots	Nordica		
1244	Man	Casual Chino Trousers			
1245	Man	Fleece Crew Sweatshirt			
1246	Man	Fleece Crew Sweatshirt V-Neck			
1247	Man	Fleece Crew Sweatshirt Zippered			
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 tới RDBMS



# Ví dụ ánh xạ tới RDBMS

Table	Attributes
AccessoryPackage	<b>AccessoryPackageID</b> , AccessoryCategory, Description
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
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> , 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

# 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		

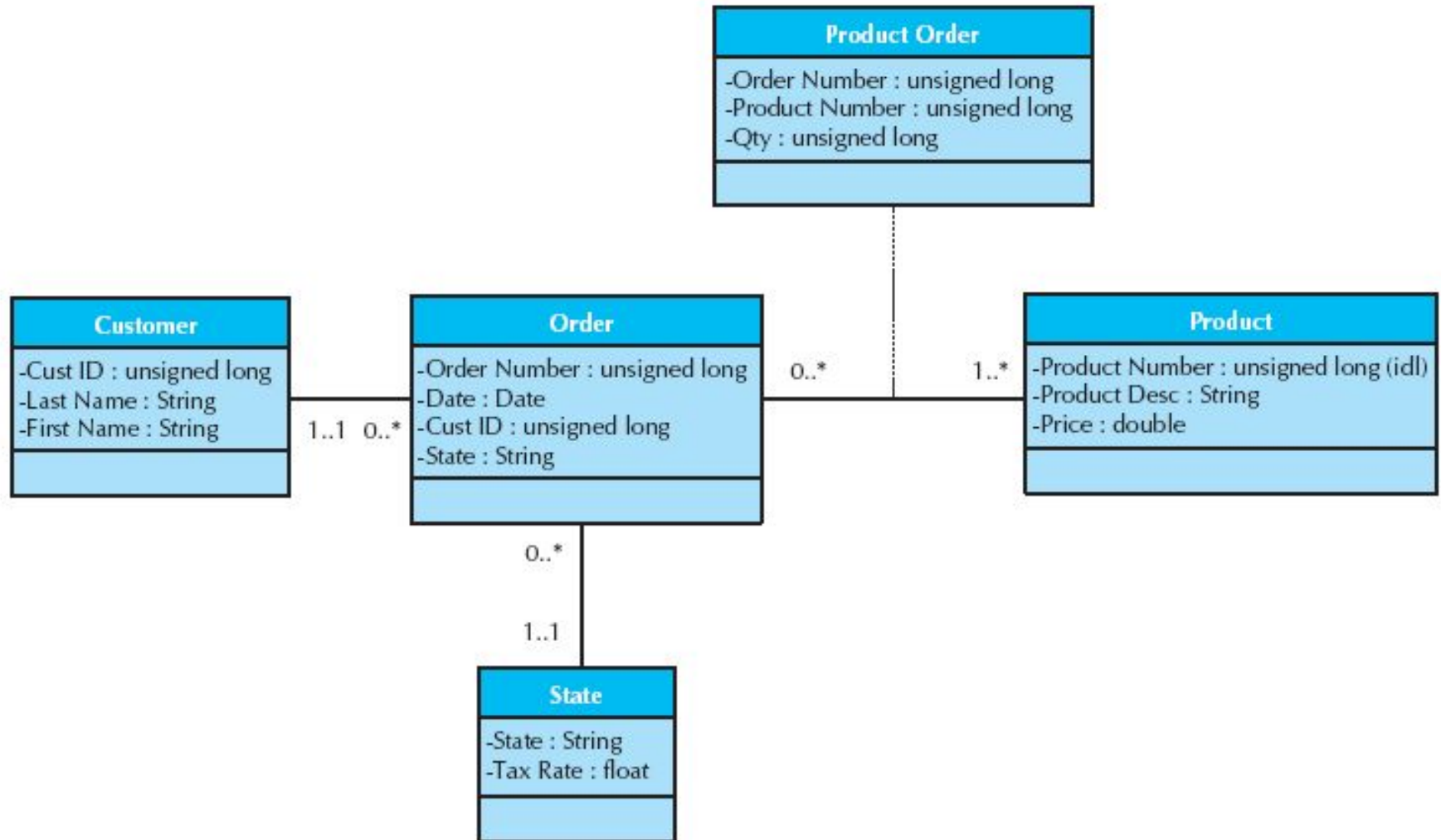
Product Table		
Product Number	Product Desc	Product Price
111	Wine Guide	\$15.00
222	Bottle Opener	\$12.00
333	Jams & Jellies	\$20.00
444	Wine Gift Pack	\$60.00
555	Cheese Tray	\$45.00

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

tax & state?



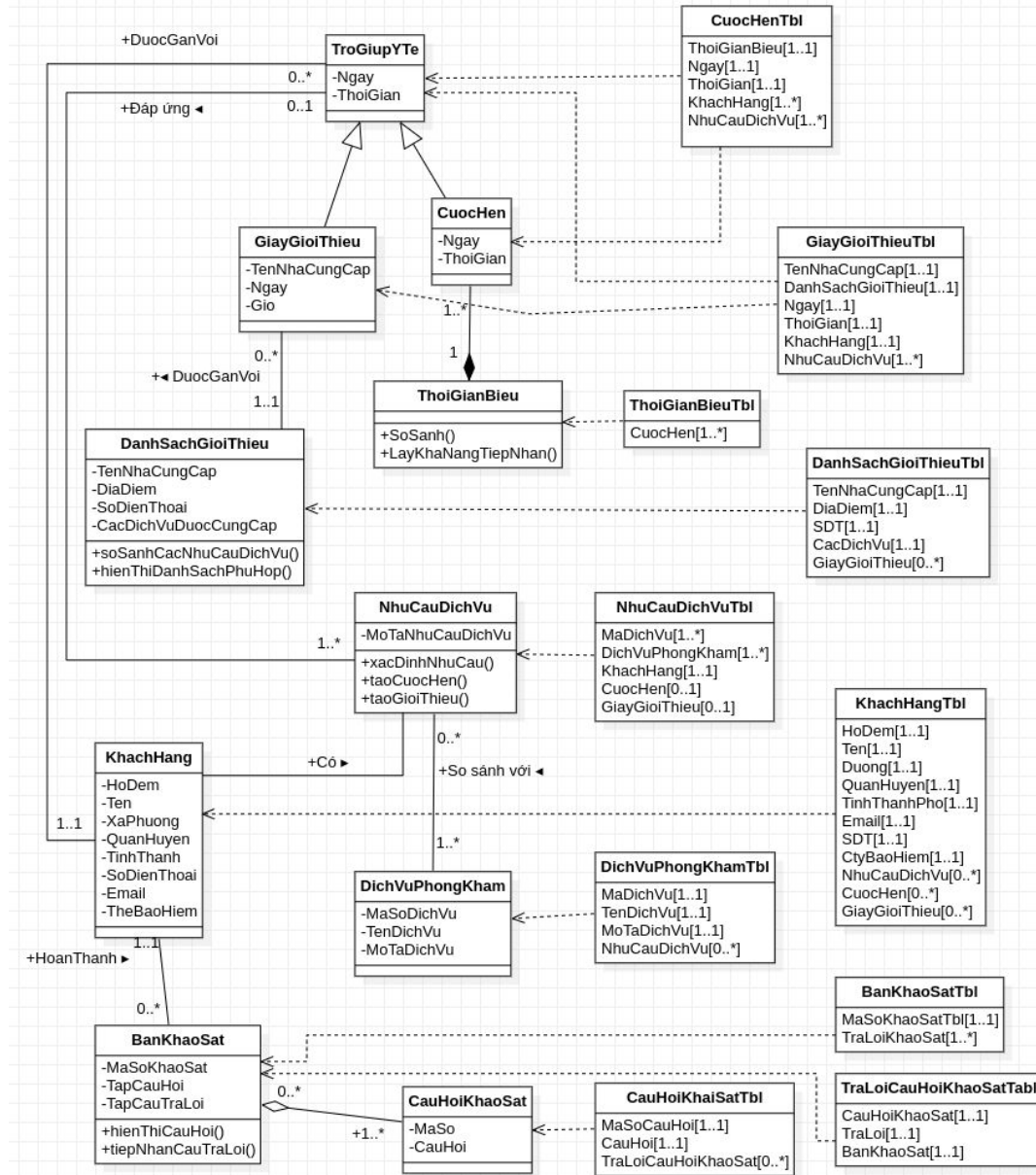
# Ví dụ chuẩn hóa: 3NF



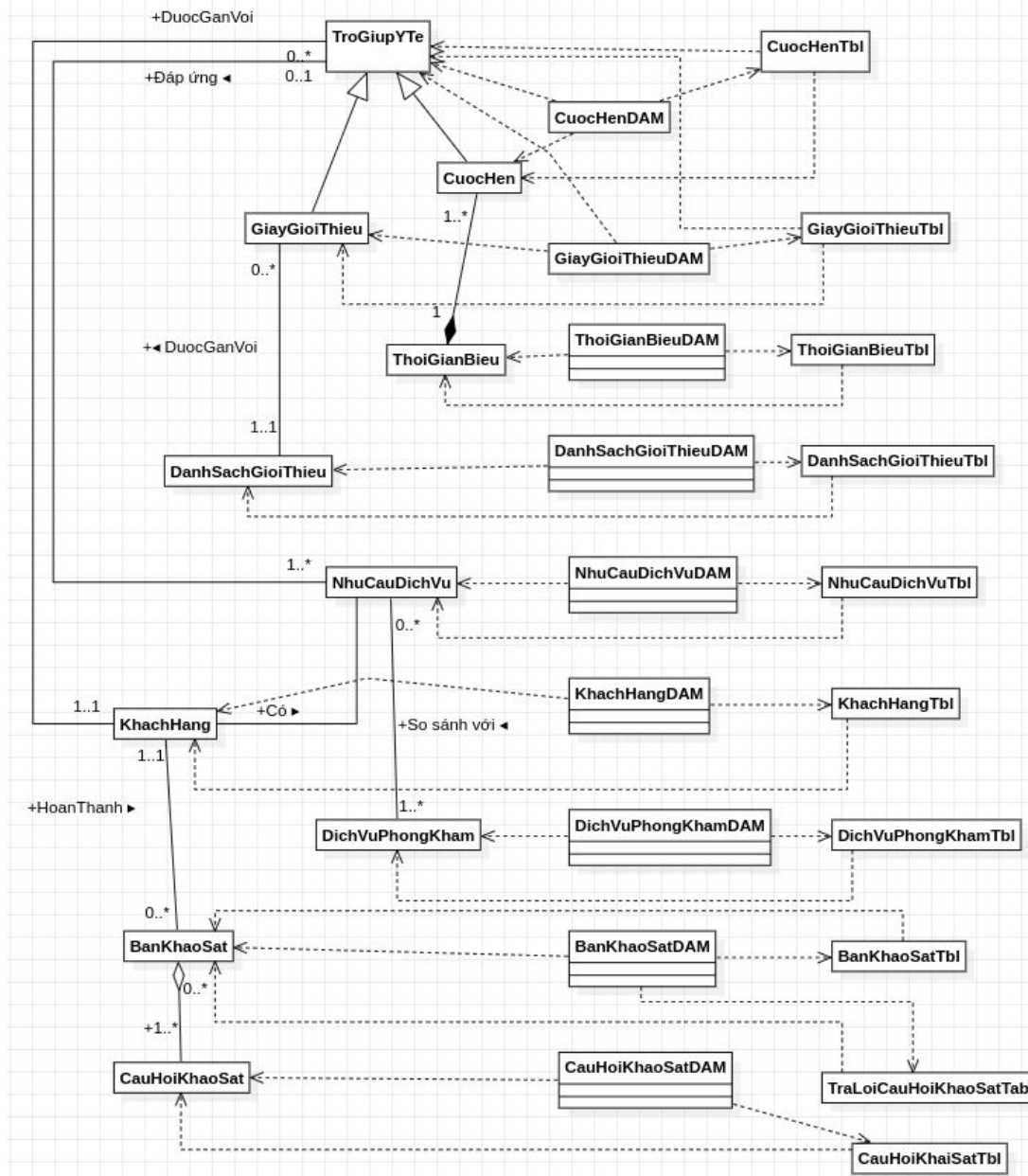
# Các lớp truy cập & quản lý dữ liệu

- Các lớp truy cập và quản lý dữ liệu / Data Access and Management (DAM) Classes
- Các lớp 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 ánh xạ lớp lĩnh vực sang ORDBMS



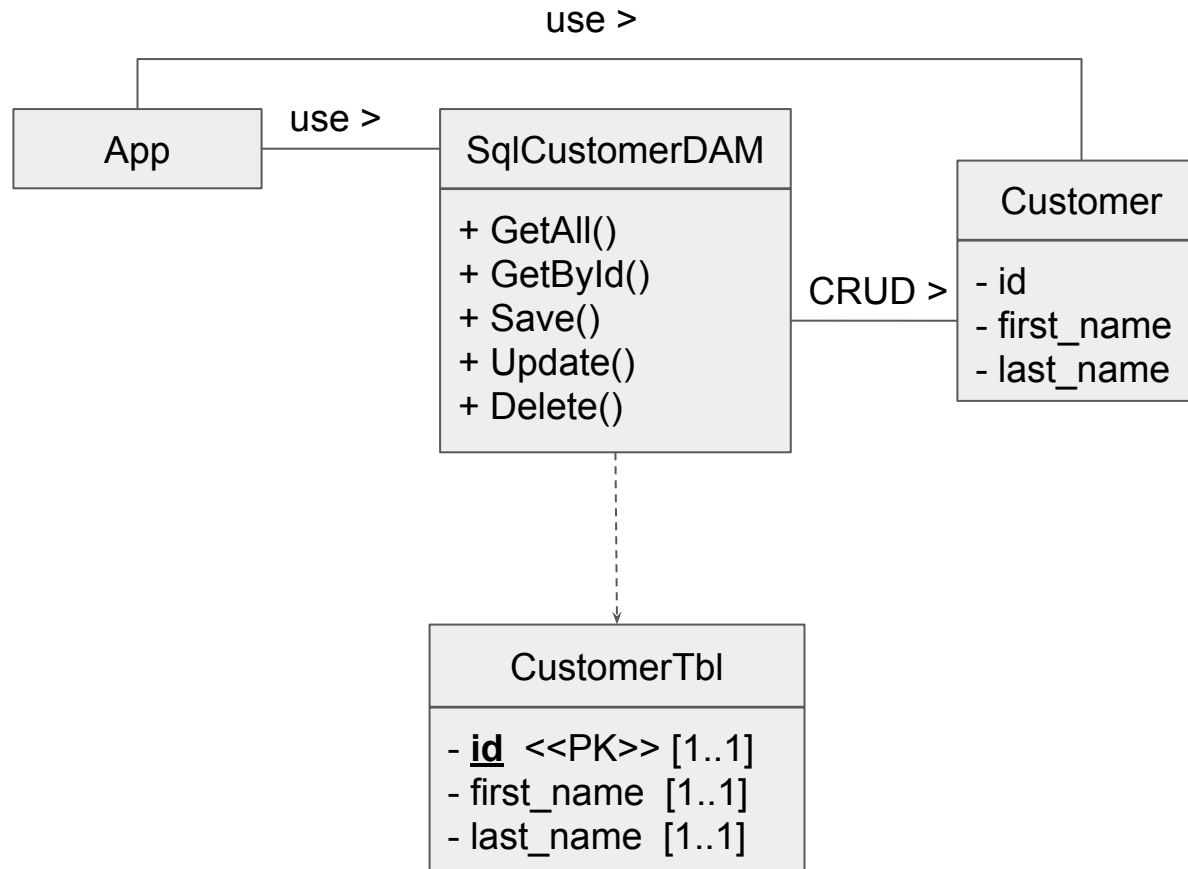
# Ví dụ các lớp DAM



# Mẫu thiết kế DAO



# Sử dụng lớp DAM và RDBMS

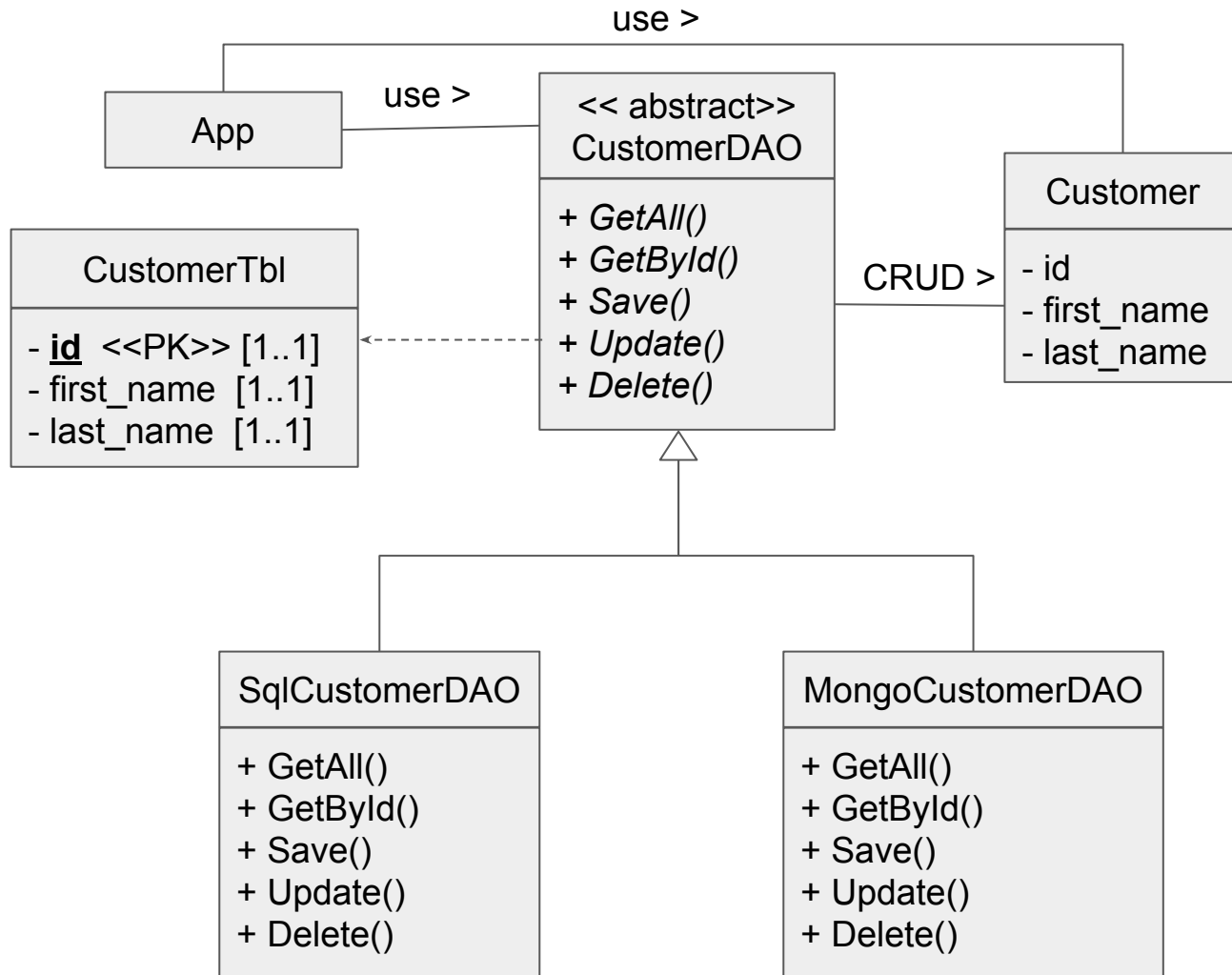


# Đặc tả bảng

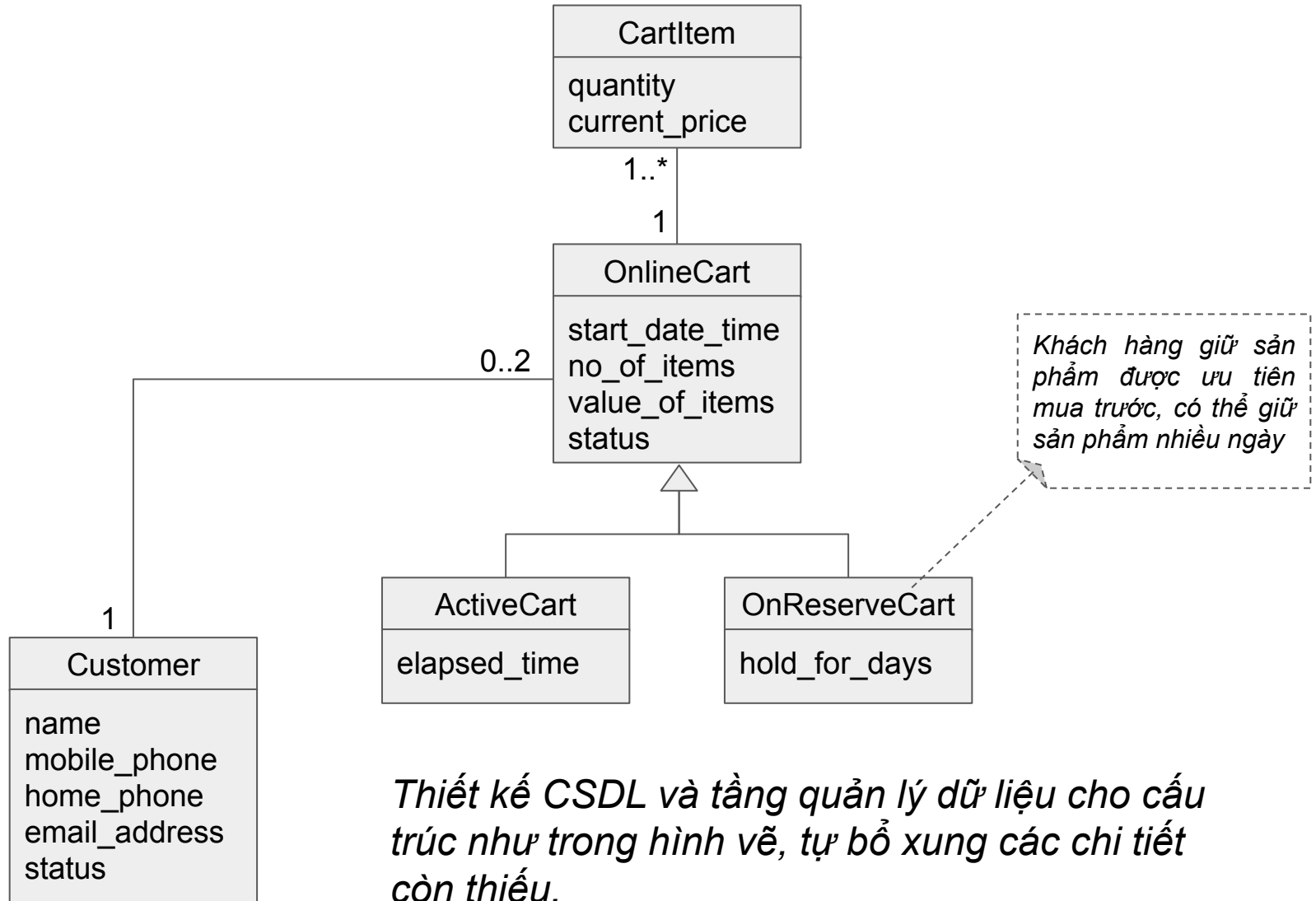
CustomerTbl
- <b>id</b> <<PK>> [1..1] - first_name [1..1] - last_name [1..1]

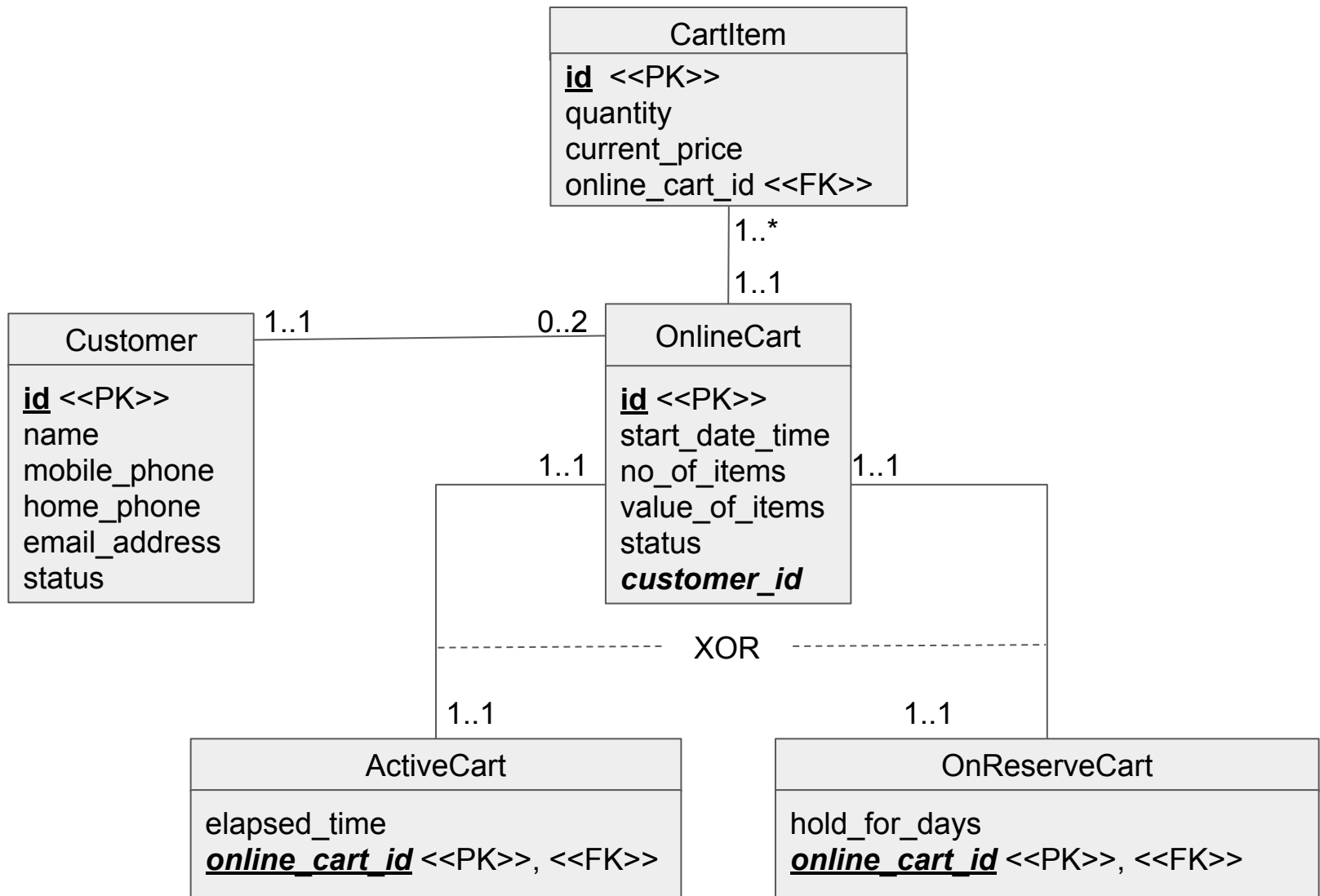
Bảng CustomerTbl				
STT	Tên cột	Kiểu dữ liệu	Ràng buộc	Ghi chú
1	id	INT	AUTO_INCREMENT	Mã khách hàng
2	first_name	VARCHAR (255)	NOT NULL	Tên
3	last_name	VARCHAR(255)	NOT NULL	Họ và Tên
Khóa chính		id		
Khóa ngoại		Không có		

# Mẫu thiết kế DAO



# Bài tập thiết kế tầng quản lý dữ liệu





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

