

PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN

(ĐH-CNTT)

HÀ THỊ THANH NGÀ
Bộ môn Hệ thống thông tin

Tài liệu tham khảo

- Joseph S. Valacich, Joey F. George; **Modern System Analysis and Design 8th edition**; 2017; Pearson
- Alan Dennis, Barbara Haley Wixom, David Tegarden; **Systems Analysis and Design An Object-Oriented Approach with UML 5th Edition**; 2015; Wiley
- Elmasri, Navathe; **Fundamentals of Database Systems-6th**; 2011; Addison Wesley
- Nguyễn Đức Thuần; **Bài giảng Cơ sở dữ liệu**; Khoa CNTT, ĐH Nha Trang
- Lê Thị Bích Hằng; **Bài giảng PTTK HĐT UML**; 2017; ĐH Nha Trang
- Huỳnh Văn Đức, Hoàng Đức Hải; **Giáo trình nhập môn UML**; 2003; NXB LĐ XH
- Phạm Nguyễn Cương, Nguyễn Trần Minh Thư, Hồ Quốc Bảo; **Giáo trình PTTKHTTT theo hướng đối tượng**, 2016, NXB Khoa học và Kỹ thuật

Chương 1. Nền tảng phát triển các hệ thống

- 1.1 Môi trường phát triển các hệ thống
- 1.2 Các nguồn phần mềm

1.1. Môi trường phát triển các hệ thống

- **Mục tiêu:**

- Định nghĩa phân tích và thiết kế HTTT
- Mô tả chu trình phát triển HTTT (System Development Life Cycle-SDLC)
- Giải thích các công cụ hỗ trợ công nghệ phần mềm (Computer Aided Software Engineering-CASE)
- Mô tả các Phương pháp Agile và eXtreme Programming

Giới thiệu

- Phân tích và thiết kế HTTT là một quy trình tổ chức phức tạp (theo từng bước).
- Được dùng để phát triển và duy trì các hệ thống thông tin dựa trên máy tính
- Được một đội nghiệp vụ và các chuyên gia hệ thống sử dụng.

Giới thiệu (tt)

- Quy trình Công nghệ Phần mềm (*Software Engineering Process*) là một quy trình được dùng để tạo một HTTT.
- Bao gồm:
 - Các phương pháp luận (Methodologies)
 - Một trình tự những cách tiếp cận từng bước giúp phát triển HTTT
 - Các kỹ thuật (Techniques)
 - Những quy trình mà các phân tích viên tuân theo để đảm bảo việc phân tích và thiết kế thông suốt, đầy đủ và toàn diện
 - Các công cụ (Tools)
 - Các chương trình máy tính trợ giúp việc áp dụng các kỹ thuật

Giới thiệu (tt)

- Các phương pháp luận (Methodologies)
 - Phương pháp luận phát triển hệ thống (**System Development Methodology**) là một quy trình tiêu chuẩn được theo trong một tổ chức để thực hiện tất cả các bước cần thiết để phân tích, thiết kế, thực thi và duy trì các HTTT
 - Hầu hết các PP luận phôi hợp vài kỹ thuật phát triển.

Giới thiệu (tt)

- Các kỹ thuật (Techniques)
 - Tập hợp các hướng dẫn giúp phân tích viên hoàn thành một hoạt động hay tác vụ phát triển HT
 - Việc học các kỹ thuật là bí quyết để trở nên thành thạo trong một lĩnh vực

Strategic planning techniques
Project management techniques
User interviewing techniques
Data-modeling techniques
Relational database design techniques
Structured programming technique
Software-testing techniques
Process modeling techniques
Domain modeling techniques
Use case modeling techniques
Object-oriented programming techniques
Architectural design techniques
User-interface design techniques

Giới thiệu (tt)

• Các công cụ (Tools)

- Thường là các ứng dụng phần mềm dễ sử dụng và có ích cho các kỹ thuật (techniques) và chính xác theo các hướng dẫn của phương pháp luận phát triển tổng thể.
- Để hiệu quả, các kỹ thuật và các công cụ phải cùng thống nhất với phương pháp luận phát triển các HT của tổ chức.
- Phạm vi từ những chương trình vẽ đơn giản đến các công cụ CASE phức tạp hay các phần mềm quản lý dự án

Project management application

Drawing/graphics application

Word processor/text editor

Visual modeling tool

Integrated development environment (IDE)

Database management application

Reverse-engineering tool

Code generator tool

Giới thiệu (tt)

- **Phần mềm ứng dụng**

- Phần mềm máy tính được thiết kế để hỗ trợ các quy trình hoặc chức năng thuộc tổ chức.

- **Phân tích viên hệ thống**

- Vai trò tổ chức có phần lớn trách nhiệm phân tích và thiết kế các HTTT
- Các phân tích viên hệ thống thực hiện phân tích và thiết kế dựa vào:
 - Sự hiểu biết về mục tiêu, cấu trúc và các quy trình của tổ chức
 - Kiến thức để khai thác kỹ thuật thông tin thuận lợi

Vai trò của nhà phân tích trong việc phát triển hệ thống

- Nghiên cứu những khó khăn và nhu cầu của một tổ chức
- Quyết định cách tiếp cận tốt nhất để cải thiện tổ chức qua việc sử dụng:
 - Con người
 - Các phương pháp
 - Công nghệ thông tin
- Giúp những người dùng HT và các nhà quản lý xác định được những yêu cầu của họ cho những HTTT mới hoặc nâng cấp

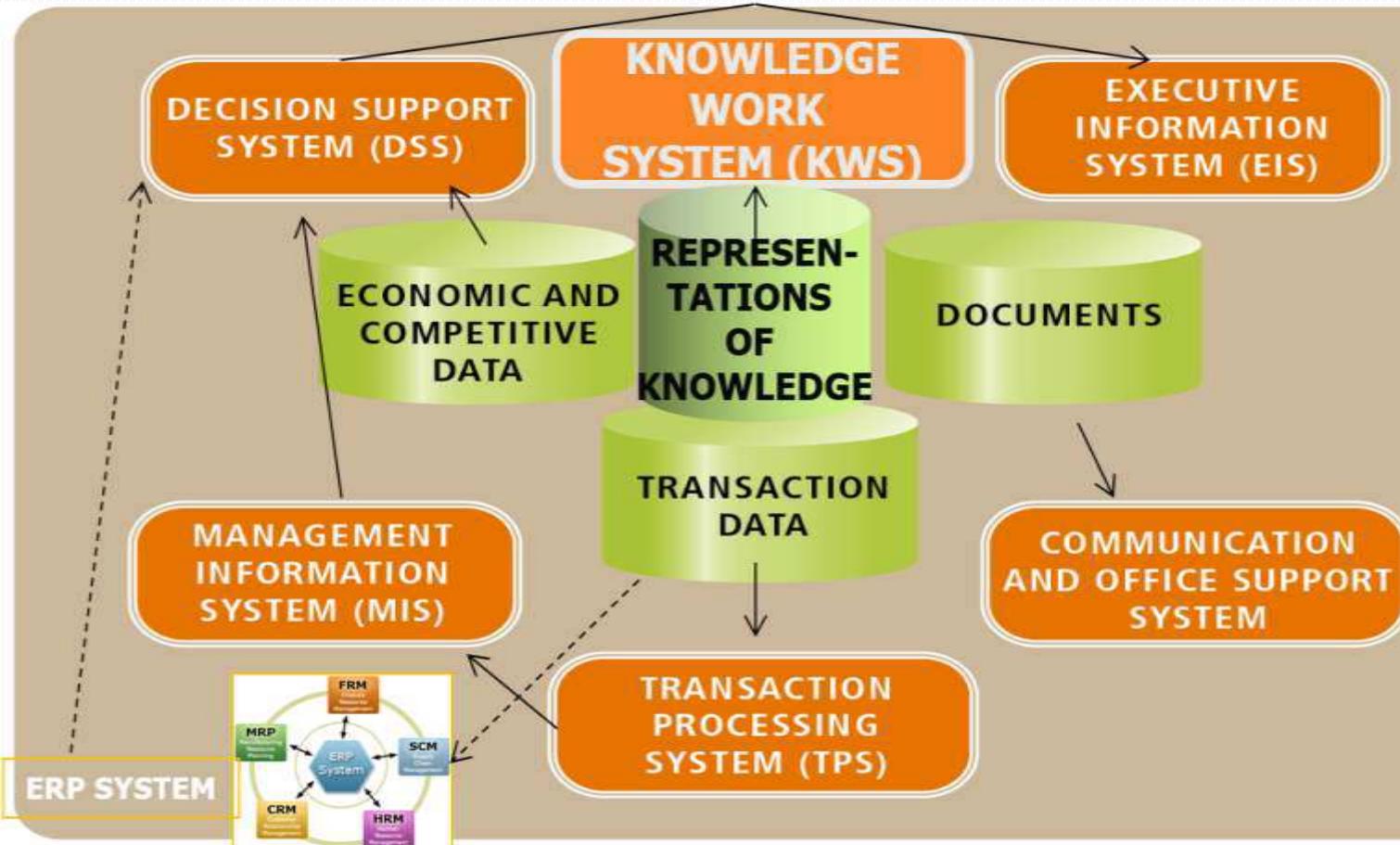
Vai trò của nhà phân tích trong việc phát triển hệ thống

- Để thành công trong vai trò một người phân tích hệ thống, cần phát triển bốn loại kỹ năng:
 - Phân tích (analytical),
 - Kỹ thuật (technical),
 - Quản lý (managerial)
 - Quan hệ cá nhân/Giao tiếp (interpersonal)

Phân tích và thiết kế các hệ thống: Các khái niệm chính

- Hệ thống: chuyển dữ liệu thành thông tin và bao gồm:
 - Phần cứng và phần mềm hệ thống
 - Tài liệu hướng dẫn và tài liệu đào tạo
 - Những vai trò công việc liên kết với hệ thống
 - Những điều khiển ngăn chặn gian lận hoặc trộm cắp
 - Những người sử dụng phần mềm để thực hiện công việc của mình.

Các loại hệ thống thông tin



- Mỗi loại HTTT có thể được đặt trong bất kỳ bộ phận nào của tổ chức (mua hàng, sản xuất, marketing/bán hàng, quản lý nguồn nhân lực,...). Ngoại trừ các HTT bao gồm nhiều hơn một bộ phận

Chu trình phát triển hệ thống (System Development Life Cycle - SDLC)

- Phương pháp truyền thống được dùng để phát triển, duy trì và thay thế các HTTT
- Các pha trong SDLC:
 - Hoạch định (Planning)
 - Phân tích (Analysis)
 - Thiết kế (Design)
 - Thực thi (Implementation)
 - Bảo trì (Maintenance).

Các pha của chu trình phát triển hệ thống (SDLC)

- **Hoạch định** – pha đầu tiên của SDLC trong đó tổng số nhu cầu HTTT của 1 tổ chức được nhận diện, phân tích, được ưu tiên, và được sắp xếp
- **Phân tích** – pha thứ hai của SDLC trong đó các yêu cầu của hệ thống được tìm hiểu và được cấu trúc.

Các pha của chu trình phát triển hệ thống (SDLC)

- **Thiết kế HT** – pha thứ ba của SDLC trong đó mô tả của giải pháp được đề xuất được chuyển thành các đặc tả hệ thống logic và sau đó là vật lý.
- **Thiết kế logic** – một phần pha thiết kế của SDLC trong đó tất cả các đặc tính chức năng của HT được lựa chọn để phát triển trong pha phân tích được mô tả độc lập với bất kỳ nền tảng máy tính nào.
- **Thiết kế vật lý** – một phần pha thiết kế của SDLC trong đó các đặc tả logic của hệ thống từ thiết kế logic được chuyển đổi thành các chi tiết kỹ thuật cụ thể từ đó mọi việc xây dựng hệ thống và lập trình có thể được hoàn thành.

Các pha của chu trình phát triển hệ thống (SDLC)

- **Thực thi** – pha thứ tư của SDLC trong đó HTTT được viết mã lệnh, được kiểm thử, được cài đặt và được hỗ trợ trong tổ chức.
- **Bảo trì** – một HTTT được chỉnh sửa và được cải tiến một cách hệ thống.

Chu trình phát triển hệ thống (SDLC)

- Có 2 cách tiếp cận chung cho SDLC
 - Cách tiếp cận dự đoán (Predictive Approach)
 - Cách tiếp cận thích nghi (Adaptive Approach)

Chu trình phát triển hệ thống (SDLC)

- **Cách tiếp cận dự đoán (Predictive Approach)**
 - Mô hình thác nước (Waterfall model)
 - Giả sử dự án có thể được lập kế hoạch trước và từ đó HTTT có thể được phát triển theo kế hoạch
 - Những yêu cầu được hiểu rõ và /hoặc rủi ro kỹ thuật thấp.
- **Cách tiếp cận thích nghi (Adaptive Approach)**
 - Mô hình lặp (Iterative model)
 - Giả sử dự án phải linh hoạt hơn và thích nghi với những trường hợp cần thay đổi khi dự án đang tiến hành.
 - Những yêu cầu và những trường hợp cần đến là không chắc chắn và/hoặc rủi ro kỹ thuật cao.

Những cách tiếp cận khác nhau để cải tiến sự phát triển

- **Prototyping** - Mẫu thử
- **CASE Tools** (*Computer Aided Software Engineering Tools*) -
Hệ thống các công cụ được sử dụng để thiết kế và phát triển
các phần mềm với sự trợ giúp của máy tính
- **JAD** (*Joint Application Design*) - Thiết kế ứng dụng ghép nối
- **RAD** (Rapid Application Development) - Phát triển UD nhanh
- **Agile Methodology** - Phương pháp linh hoạt

Những cách tiếp cận khác nhau để cải tiến sự phát triển (tt.)

- **Prototyping-Mẫu thử**

- Xây dựng một phiên bản làm việc thu nhỏ của hệ thống
- Thuận lợi:
 - Liên quan đến những người dùng trong thiết kế
 - Nắm bắt những yêu cầu theo hình thức cụ thể

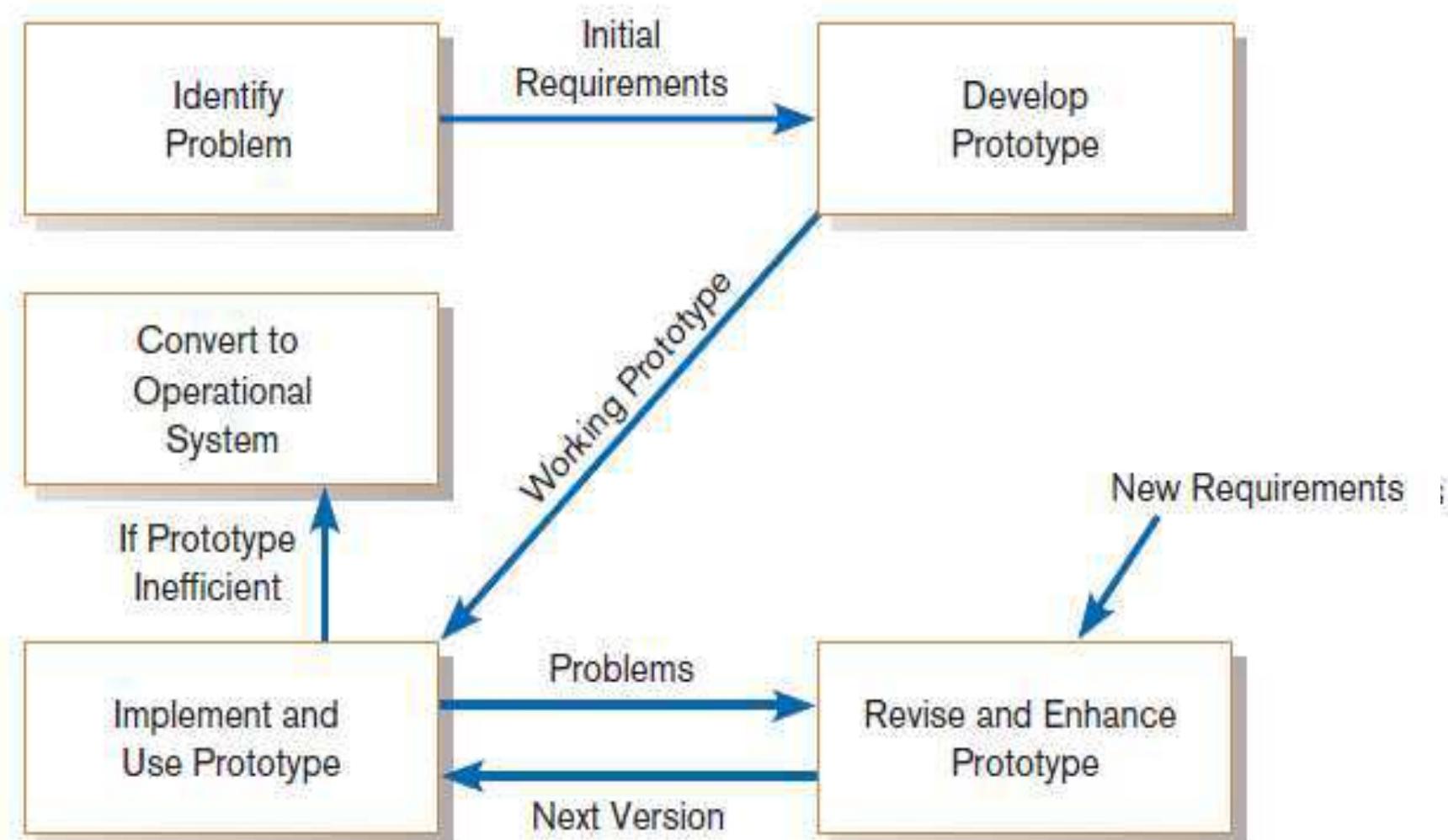
Những cách tiếp cận khác nhau để cải tiến sự phát triển (tt.)

• Mẫu thử (Prototyping)

FIGURE 1-12

The prototyping method.

Source: Adapted from J. D. Naumann and A. M. Jenkins, "Prototyping: The New Paradigm for Systems Development," MIS Quarterly 6, no. 3 (1982): 29-44.



Những cách tiếp cận khác nhau để cải tiến sự phát triển (tt.)

- **CASE Tools** (Computer Aided Software Engineering Tools)

- Các công cụ lập biểu đồ cho phép biểu diễn đồ họa được sử dụng bởi các nhà phân tích hệ thống để phát triển các HTTT
- Các màn hình máy tính và bộ tạo sinh báo cáo giúp lập mẫu thử các HT “trông thấy và cảm nhận” như thế nào
- Các công cụ phân tích tự động kiểm tra sự thống nhất trong các biểu đồ, biểu mẫu và các báo cáo.
- Một kho trung tâm cung cấp việc lưu trữ được tích hợp các biểu đồ, các báo cáo, và các đặc tả QLDA.
- Các bộ tạo sinh tài liệu tiêu chuẩn hóa các tài liệu kỹ thuật và tài liệu hướng dẫn người dùng.
- Các bộ tạo sinh mã lệnh cho phép việc tạo tự động mã lệnh các chương trình và CSDL trực tiếp từ các tài liệu thiết kế, các biểu đồ, các biểu mẫu và các báo cáo.

Những cách tiếp cận khác nhau để cải tiến sự phát triển (tt)

- Các loại CASE tools phổ biến
 - Các công cụ lập biểu đồ (Diagramming tools) cho phép biểu diễn đồ họa. Vd: *PowerDesigner, Microsoft Visio, ArgoUML, StarUML, Visual Paradigm*
 - Công cụ lập mô hình quy trình (Process Modeling Tools).
Vd: EPF Composer
 - Công cụ quản lý dự án (Project Management Tools).
Vd: Creative Pro Office, Trac Project, Basecamp, Microsoft Project,.....
 - Bộ tạo sinh tài liệu (Documentation Tools).
Vd: Doxygen, DrExplain, Adobe RoboHelp

Những cách tiếp cận khác nhau để cải tiến sự phát triển (tt)

- Các loại CASE tools phổ biến
 - Các công cụ phân tích (Analysis Tools) tự động kiểm tra sự thống nhất trong các biểu đồ, biểu mẫu và báo cáo.
Vd: CaseComplete (phân tích các yêu cầu), Visible Analyst
 - Các công cụ thiết kế (Design Tools). Vd: Animated Software Design
 - Các bộ tạo sinh mã cho phép tự động tạo các chương trình và mã csdl trực tiếp từ các tài liệu thiết kế, biểu đồ, biểu mẫu và báo cáo.
Vd: Microsoft Visual Studio
 - Các công cụ lập trình (Programming Tools).
Vd: Eclipse

Những cách tiếp cận khác nhau để cải tiến sự phát triển (tt)

• JAD (Joint Application Design)

- Vào cuối những năm 1970s, bộ phận phát triển các hệ thống của IBM đã phát triển một quy trình mới để tập hợp các yêu cầu HTTT và xem xét các thiết kế hệ thống → Thiết kế ứng dụng liên kết.
- Một quy trình có cấu trúc gồm những người dùng, các nhà quản lý, các nhà phân tích làm việc cùng nhau vài ngày trong một loạt các cuộc họp chuyên sâu để đặc tả hoặc xem xét các yêu cầu hệ thống.

Những cách tiếp cận khác nhau để cải tiến sự phát triển (tt)

- **RAD (Rapid Application Development)**

- PP phát triển các HT được tạo nhằm giảm triệt để thời gian cần dùng để thiết kế và thực thi các HTTT
- Mẫu thử, CASE và JAD là những công cụ chính hỗ trợ RAD

Agile Methodologies

- Được tạo động lực qua việc nhận diện việc phát triển phần mềm có khả năng thay đổi, không thể dự đoán và động
- Ba nguyên tắc chính:
 - Thích nghi hơn dự đoán
 - Nhấn mạnh con người hơn các vai trò
 - Các quy trình tự thích nghi

Agile Methodologies

- **Các PP phát triển PM agile nổi tiếng gồm**

- Agile Software Development
- **eXtreme Programming (XP)**
- Feature Driven Development (FDD)
 - Tương tự XP nhưng tập trung nhiều vào tài liệu và quy trình
- Adaptive Software Development (ASD)
- Dynamic System Development Method (DSDM) - *Phổ biến ở Châu Âu*
- **Unified Process (UP) or Rational Unified Process (RUP)**
- **Scrum**
-

eXtreme Programming

- Các chu trình phát triển ngắn, tăng trưởng
- Kiểm thử tự động
- Các đội lập trình 2 người
- Viết mã lệnh, kiểm thử, lắng nghe, thiết kế
- Viết mã lệnh và kiểm thử thao tác cùng nhau
- Thuận lợi:
 - Giao tiếp giữa các nhà phát triển
 - Năng suất mức độ cao
 - Mã lệnh chất lượng cao

PP mô hình hóa khi phân tích và thiết kế các HTTT

- Các PP mô hình hóa chia theo hai xu hướng:
 - Mô hình hóa **hướng chức năng** (từ 1970, với Youdon, Constantine, DeMarco,...)
 - Mô hình hóa **hướng đối tượng** (từ 1990, với Booch, Rumbaugh, Jacobson, Yourdon & Coad,...)

PP mô hình hóa khi phân tích và thiết kế các HTTT

- **Cách tiếp cận có cấu trúc (Structured Approach)**
 - Cách tiếp cận ban đầu. Thừa nhận một HT là một tập hợp các quy trình tương tác với dữ liệu
 - Phân tích có cấu trúc, thiết kế có cấu trúc và lập trình có cấu trúc
- **Cách tiếp cận hướng đối tượng (Object-oriented approach)**
 - Cách tiếp cận gần đây. Thừa nhận một HT là một tập hợp các đối tượng tương tác để hoàn thành các tác vụ
 - OO analysis, OO design, và OO programming

Điểm khác nhau giữa SAD và OOSAD

Những đặc điểm	Structured Analysis and Design	Object-Oriented System Analysis and Design
Phương pháp luận	SDLC	Lắp lại/tăng trưởng
Tập trung	Các tiến trình (Processes)	Các đối tượng (Objects)
Rủi ro	Thấp	Cao
Sử dụng lại	Thấp	Cao
Tính hoàn thiện	Hoàn thiện và mở rộng	Đang phát triển
Phù hợp với	Những dự án được định nghĩa rõ ràng với những yêu cầu người dùng ổn định	Những dự án lớn nhiều rủi ro với những thay đổi về các yêu cầu người dùng.

1.2 Các nguồn phần mềm

- Mục tiêu
 - Hiểu outsourcing
 - Mô tả 6 loại nguồn phần mềm khác nhau
 - Thảo luận việc làm thế nào đánh giá phần mềm off-the-shelf
 - Giải thích việc tái sử dụng và vai trò của nó trong việc phát triển phần mềm

Outsourcing (Sử dụng nguồn lực bên ngoài)

- Việc thực hiện chuyển một số hoặc tất cả trách nhiệm về các hoạt động và ứng dụng HTTT của một tổ chức cho công ty khác bên ngoài
- Vd:
 - Công ty outsourcing phát triển và chạy các ứng dụng trên các máy tính của họ (ứng dụng Payroll)
 - Công ty outsourcing chạy các ứng dụng tại nơi có các máy tính của bạn
- Outsourcing nên được xem xét trong khi phân tích các HT

Outsourcing (Sử dụng nguồn lực bên ngoài)

- Lý do để sử dụng nguồn lực bên ngoài
 - Chi phí - Sự hiệu quả
 - Giảm thời gian tiếp cận thị trường
 - Tăng hiệu suất xử lý
 - Khắc phục những khó khăn của tổ chức trong đơn vị HTTT
 - Phát triển HT là một hoạt động không cốt lõi của tổ chức
- Outsourcing nên được xem xét khi phân tích các HT

Các nguồn phần mềm

- Các công ty Dịch vụ CNTT
- Các nhà sản xuất PM đóng gói
- Các nhà cung cấp phần mềm giải pháp toàn tổ chức
 - Enterprise Resource Planning (ERP)
- Các nhà cung cấp điện toán đám mây
- Các giải pháp PM mã nguồn mở
- Các nhà phát triển tại chỗ

Các nguồn phần mềm

- **Các công ty Dịch vụ CNTT**
- Được dùng khi
 - Tổ chức thiếu các nguồn lực để phát triển tại chỗ
 - Giải pháp off-the-shelf phù hợp không có sẵn
- Giúp các công ty phát triển các HTTT đặt làm để sử dụng nội bộ
- Phát triển, làm chủ (host) và chạy các ứng dụng
- Có thể cung cấp chuyên môn nghiệp vụ
- Các công ty dịch vụ CNTT thuê các tư vấn về kinh nghiệm nghiệp vụ cụ thể.

Các nguồn phần mềm (tt)

- **Các nhà sản xuất Phần mềm đóng gói**

- Sản xuất các hệ thống pre-packed hoặc off-the-shelf
- Các sản phẩm phạm vi từ các gói liên quan nhiều lĩnh vực đến các gói đặc trưng theo ngành
- Hai loại phần mềm
 - Turnkey – chìa khóa trao tay: không thể được chỉnh sửa để thỏa mãn những nhu cầu người dùng cụ thể
 - Non-turnkey: có thể được chỉnh sửa bởi nhà cung cấp để thỏa mãn tốt hơn nhu cầu của người dùng
- Phần mềm Pre-packed là phần mềm off-the-shelf, turnkey (ie. Không thể tùy chỉnh)
- Phần mềm Off-the-shelf có thể thỏa mãn đến 70% nhu cầu của một tổ chức.

Các nguồn phần mềm (tt)

- **Các Phần mềm giải pháp cho doanh nghiệp**

- Còn gọi là các hệ thống Hoạch định nguồn lực tổ chức (Enterprise Resource Planning (ERP) Systems)
- Gồm một loạt các module được tích hợp
 - Mỗi module hỗ trợ chức năng nghiệp vụ truyền thống đơn lẻ
 - Kế toán
 - Phân phối
 - Sản xuất
 - Nguồn nhân lực
 - Được tích hợp để tập trung vào các tiến trình nghiệp vụ hơn là các lĩnh vực nghiệp vụ chức năng.
- Cho phép một tổ chức tích hợp tất cả các phần của một quy trình nghiệp vụ vào một HTTT thống nhất
 - Vd. Tất cả các khía cạnh của một giao dịch đặt hàng xuất hiện liền mạch trong HTTT đơn

Các nguồn phần mềm (tt)

- Lợi ích **Các Phần mềm giải pháp cho doanh nghiệp**
 - Kho đơn cho tất cả khía cạnh của một quy trình nghiệp vụ
 - Đảm bảo dữ liệu thống nhất và chính xác hơn
 - Ít bảo trì
 - SỰ linh hoạt của các module
 - Các module có thể được thêm vào khi cần thiết
 - Các module được tích hợp ngay vào hệ thống đang có.
- **Bất lợi Các Phần mềm giải pháp cho doanh nghiệp**
 - SỰ phức tạp khi thực thi → thời gian thực thi mở rộng
 - Độ tin cậy dựa vào các nhà tư vấn
 - Các tổ chức thường phải thay đổi cách họ thực hiện nghiệp vụ để sử dụng các hệ thống
 - Các nhà phân phối ERP: SAP AG, Oracle Corp

Các nguồn phần mềm (tt)

- **Điện toán đám mây (Cloud Computing)**

- Trả chi phí cho việc sử dụng (phần mềm như một dịch vụ): tự động thuê phần mềm từ các nhà cung cấp thứ 3 (chạy trên các server từ xa = đám mây)
- Trả chi phí cho dịch vụ: Các ứng dụng được mua, được cài đặt và bảo trì bởi nhà cung cấp dịch vụ
- Trả chi phí cho nền tảng (phần cứng như một dịch vụ): tự động thuê nền tảng và quản lý trung tâm dữ liệu ảo của riêng bạn
- Các ứng dụng được truy cập qua Internet (VPN -Virtual Private Network)
- Những người dùng trả tiền mỗi lần dùng hoặc cấp quyền hàng tháng

Các nguồn phần mềm (tt)

- **Điện toán đám mây (Cloud Computing) (tt)**

- Lý do chọn lựa Đám mây
 - Giải phóng đội ngũ IT nội bộ
 - Truy cập nhanh hơn đến ứng dụng so với phát triển nội bộ
 - Chi phí thấp hơn phát triển nội bộ
- Các mối quan tâm
 - Độ tin cậy
 - An toàn thông tin
 - Tuân thủ các quy định của chính phủ

Các nguồn phần mềm (tt)

- **Phần mềm Open-Source**

- Có sẵn tự do
- Gồm source code
- Được duy trì bởi một cộng đồng những người quan tâm
- Vd: ???
- Làm thế nào kiếm tiền từ Open-Source?
 - Cung cấp bảo trì/dịch vụ
 - Bán một phiên bản đặc trưng hơn của PM miễn phí

Các nguồn phần mềm (tt)

• **Phát triển In-House**

- Nếu chuyên gia phát triển HT có khả năng với nền được chọn đã có tại chỗ, thì một vài hoặc tất cả HT có thể được phát triển bởi chính đối ngũ của tổ chức
- Các giải pháp lai liên quan một số thành phần được mua và một số thành phần tại chỗ đang có

Bảng 2-2. So sánh sáu nguồn thành phần phần mềm

Các nhà sản xuất	Khi nào chọn loại này của tổ chức phần mềm	Những yêu cầu đối với nhân viên nội bộ
Các công ty Dịch vụ CNTT	Khi tác vụ yêu cầu hỗ trợ tùy chỉnh và các hệ thống không thể được xây dựng nội bộ hoặc hệ thống cần phải có nguồn gốc	Nhân viên nội bộ có thể cần thiết, tuy thuộc vào ứng dụng
Các nhà sản xuất PM đóng gói (packaged software producers)	Khi tác vụ được hỗ trợ là chung chung	Một số nhân viên hệ thống thông tin và người sử dụng để xác định các yêu cầu và đánh giá các gói phần mềm
Các giải pháp toàn tổ chức (enterprise-wide solutions)	Đối với các hệ thống hoàn chỉnh hướng liên chức năng	Một số nhân viên nội bộ cần thiết nhưng chủ yếu cần tư vấn
Điện toán đám mây (cloud computings)	Để truy cập nhanh vào một ứng dụng; khi nào tác vụ được hỗ trợ là chung chung	Vài nhân viên; giải phóng nhân viên cho công việc CNTT khác
Phần mềm mã nguồn mở (Open source software)	Khi tác vụ được hỗ trợ là chung chung nhưng chi phí là một vấn đề	Một số nhân viên hệ thống thông tin và người sử dụng để xác định các yêu cầu và đánh giá gói phần mềm.
Các nhà phát triển tại chỗ (In-house developers)	Khi nguồn tài nguyên và nhân viên có sẵn và các hệ thống có thể được xây dựng từ đầu	Nhân viên nội bộ cần thiết nhưng kích thước nhân viên có thể thay đổi

Việc lựa chọn phần mềm bán đại trà (Off-the-Shelf Software)

- Tiêu chuẩn
 - Chi phí
 - In-house vs purchase
 - Tính năng
 - Hỗ trợ từ nhà phân phối
 - Khả năng tồn tại của nhà phân phối
 - Sự linh hoạt
 - Tài liệu hướng dẫn
 - Thời gian phản hồi
 - Sự dễ dàng cài đặt

Phê chuẩn thông tin phần mềm đã mua

- Sử dụng các nguồn thông tin đa dạng
 - Tập hợp thông tin từ nhà cung cấp
 - Tài liệu hướng dẫn phần mềm
 - Tài liệu tiếp thị kỹ thuật
 - Bảng câu hỏi

Phê chuẩn thông tin phần mềm đã mua (tt)

- **Hồ sơ mời thầu (Request for Proposal - RFP)**
 - Là một tài liệu được cung cấp cho các nhà cung cấp để yêu cầu họ đề xuất phần cứng và phần mềm HT sẽ thỏa mãn những yêu cầu của một HT mới.
 - SỰ VẬN HÀNH PM QUA MỘT LOẠT KIỂM THỬ
 - Phản hồi từ những người dùng khác của sản phẩm từ nhà cung cấp
 - Các dịch vụ kiểm thử phần mềm độc lập

Tái sử dụng (Reuse)

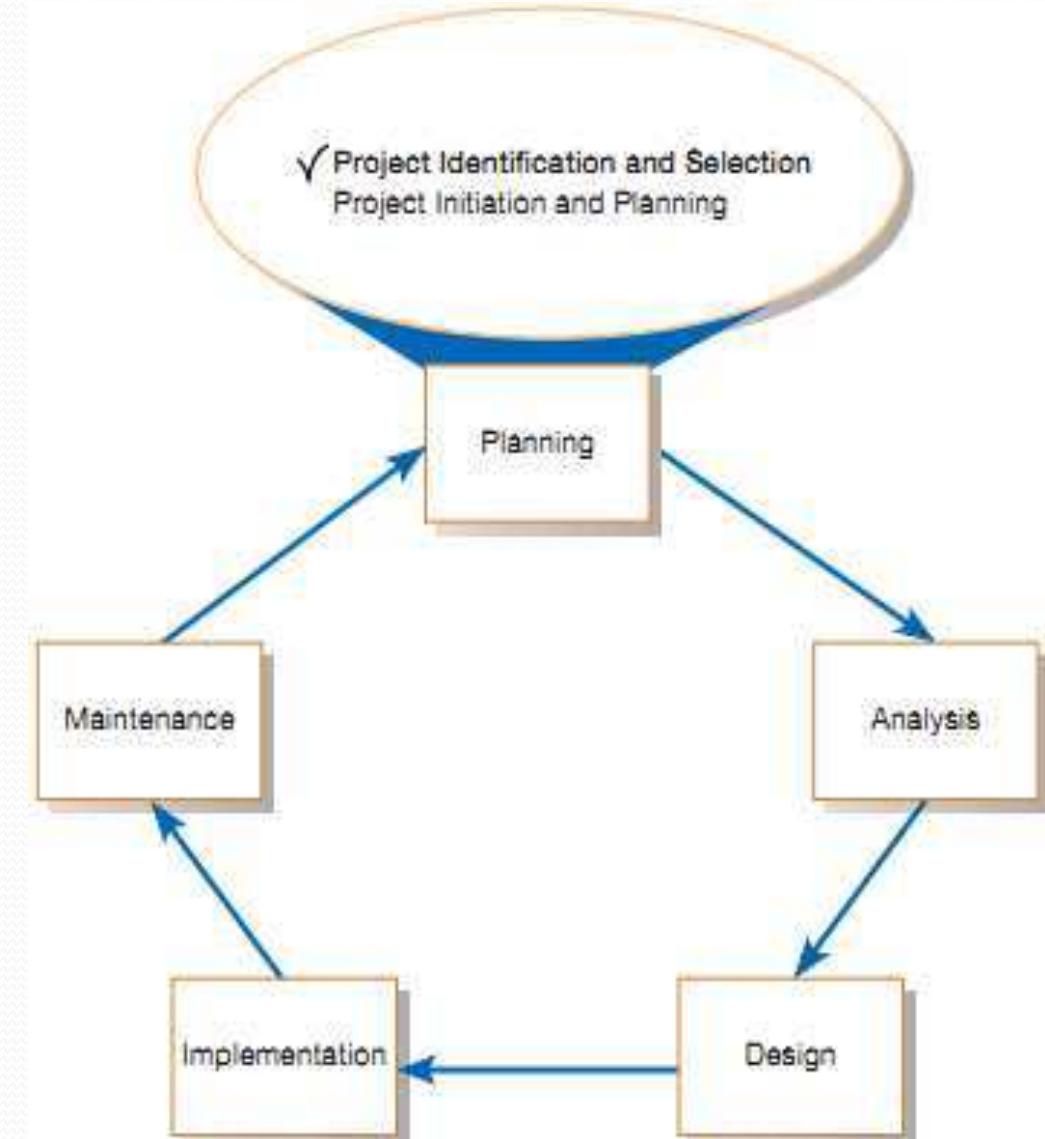
- Việc sử dụng các nguồn tài nguyên phần mềm được viết trước đó vào những ứng dụng mới
- Hầu hết thường được áp dụng cho việc phát triển hướng đối tượng và hướng thành phần
- Việc tái sử dụng các object classes
 - Tăng hiệu suất
 - Giảm mật độ sai sót
 - Giảm việc làm lại
- Kế hoạch tái sử dụng phải phù hợp với các mục tiêu kinh doanh chiến lược của các công ty

Tái sử dụng

- 4 cách tiếp cận
 - **Ad hoc Reuse:** các cá nhân tự do tìm hoặc phát triển những tài sản có thể dùng lại được
 - **Facilitated Reuse:** các nhà phát triển được khuyến khích thực hành việc dùng lại
 - **Managed Reuse:** việc phát triển, chia sẻ, và chấp nhận những tài sản có thể dùng lại được bị bắt buộc
 - **Designed Reuse:** những tài sản bị bắt buộc sử dụng lại khi chúng được thiết kế cho các ứng dụng riêng biệt

Chương 2. Hoạch định

- Nhận diện và lựa chọn các dự án phát triển hệ thống
- Khởi động và hoạch định các dự án phát triển hệ thống



Nhận diện và lựa chọn dự án

- Gồm 3 bước chính :

- 1. Nhận diện các dự án phát triển tiềm năng
- 2. Phân loại và xếp hạng các dự án phát triển HTTT
- 3. Lựa chọn các dự án phát triển HTTT

Việc nhận diện và lựa chọn các dự án (tt)

- **1. Nhận diện các dự án tiềm năng** từ các bên liên quan: việc nhận diện từ mỗi nhóm liên quan dự án chia sẻ viễn cảnh và động lực cho quyết định HTTT
 - Quản lý cấp cao
 - Ban chỉ đạo
 - Các phòng ban người dùng
 - Nhóm phát triển hoặc đội ngũ IS cao cấp

Việc nhận diện và lựa chọn các dự án

• **1. Nhận diện các dự án tiềm năng**

- Các dự án được nhận diện bởi một sự kết hợp tiếp cận nhóm từ trên xuống và nhóm từ dưới lên
 - Nhóm nhận diện từ trên xuống
 - Quản lý cấp cao hoặc ban lãnh đạo
 - Tập trung vào nhu cầu tổng quát của tổ chức
 - Nhóm nhận diện từ dưới lên
 - Các đơn vị kinh doanh hoặc nhóm CNTT
 - Không phản ánh các mục tiêu tổng thể của tổ chức

Việc nhận diện và lựa chọn các dự án (tt)

Bảng 4-1. Những đặc điểm chung của Các phương pháp thay thế cho việc nhận diện các Hệ thống thông tin và ra Các quyết định lựa chọn

Nguồn dự án	Chi phí	Thời gian thực hiện	Độ phức tạp	Quy mô hệ thống	Tiêu điểm
Ban quản lý cấp cao	Cao nhất	Dài nhất	Cao nhất	Lớn nhất	Chiến lược
Ban chỉ đạo	Cao	Dài	Cao	Lớn	Liên chức năng
Bộ phận người dùng	Thấp	Ngắn	Thấp	Nhỏ	Bộ phận
Nhóm phát triển	Thấp đến cao	Ngắn đến dài	Thấp đến cao	Nhỏ đến lớn	Tích hợp với các hệ thống hiện có

Việc nhận diện và lựa chọn các dự án (tt)

Bảng 4-2. Các tiêu chuẩn đánh giá có thể khi phân loại và xếp hạng các dự án

Tiêu chuẩn đánh giá	Mô tả
Phân tích chuỗi giá trị	Mức độ mà các hoạt động tăng thêm giá trị và chi phí khi phát triển các sản phẩm và/hoặc các dịch vụ; Các dự án hệ thống thông tin cung cấp những lợi ích tổng thể lớn nhất sẽ được ưu tiên hơn những dự án có ít lợi ích hơn
Định hướng chiến lược	Mức độ mà dự án được xem như giúp tổ chức đạt được mục tiêu chiến lược và mục đích dài hạn
Lợi ích tiềm năng	Mức độ mà dự án được xem như cải thiện lợi nhuận, dịch vụ khách hàng, vv, và thời gian thực hiện của những lợi ích này
Tình sẵn có của nguồn lực	Số tiền và loại tài nguyên mà dự án yêu cầu và sự sẵn có của chúng
Quy mô/Thời gian thực hiện dự án	Số lượng cá nhân và khoảng thời gian cần thiết để hoàn thành dự án
Khó khăn/những rủi ro kỹ thuật	Độ khó kỹ thuật để hoàn thành dự án thành công trong những ràng buộc thời gian và nguồn lực nhất định.

Việc nhận diện và lựa chọn các dự án (tt)

• **3. Chọn lựa các dự án phát triển IS**

- Dựa trên các nhân tố khác nhau:
 - Nhu cầu được nhận biết từ tổ chức
 - Các hệ thống đang tồn tại và các dự án đang triển khai
 - Tính sẵn có của nguồn tài nguyên
 - Các tiêu chuẩn đánh giá
 - Các điều kiện kinh doanh hiện tại
 - Ý định của những người ra quyết định
- Cả DA ngắn hạn và dài hạn được xem xét
- Phù hợp để đạt những mục tiêu kinh doanh đã lựa chọn
- Một hoạt động rất quan trọng và tiếp diễn

Việc nhận diện và lựa chọn các dự án (tt)

- Một phương pháp dùng để quyết định lựa chọn giữa các dự án khác nhau hoặc các thiết kế thay thế:
 - Với mỗi yêu cầu hoặc ràng buộc:
Score = weight X rating
 - Mỗi thay thế: tổng điểm tính qua các yêu cầu/ràng buộc.
 - Lựa chọn phương án có điểm số cao nhất

Việc nhận diện và lựa chọn các dự án (tt)

- Những sản phẩm bàn giao và kết quả
 - Những sản phẩm bàn giao ban đầu của pha này là một bản danh mục các dự án phát triển IS cụ thể
 - Ràng buộc/cam kết từng bước
 - Đánh giá lại dự án liên tục sau mỗi pha

Việc nhận diện và lựa chọn các dự án (tt)

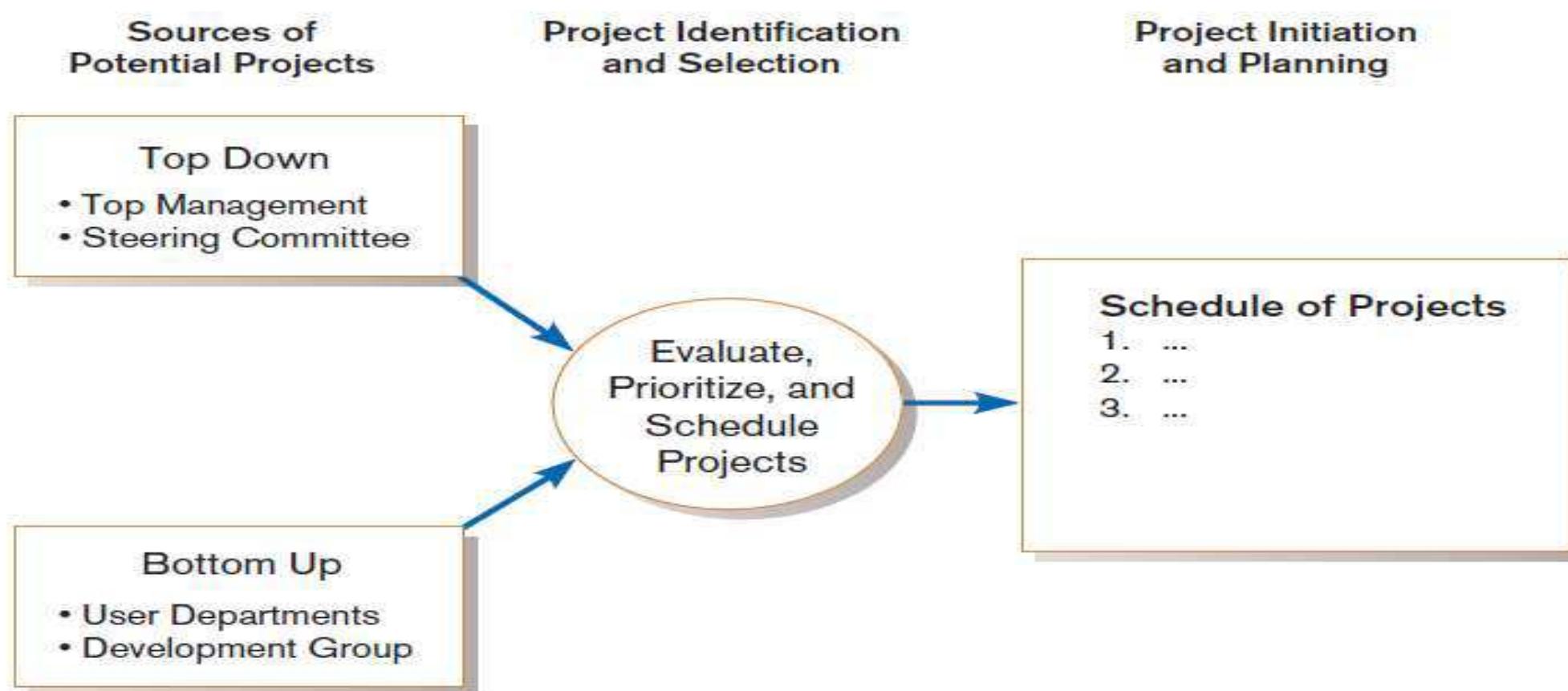


Figure 4-5

Các dự án phát triển HTTT từ những đánh giá mức top-down và bottom-up

Hoạch định các HTTT và tập đoàn

- Để đạt lợi ích từ cách tiếp cận dựa trên việc hoạch định để nhận diện và lựa chọn các dự án, một tổ chức phải:
 - Phân tích thông tin của tổ chức cần
 - Lập kế hoạch các dự án của tổ chức cẩn thận.

Các lý do về tầm quan trọng của việc hoạch định được cải thiện

- Tăng chi phí các HTTT (40% chi phí thuộc tổ chức)
- Thiếu các hệ thống và ứng dụng xuyên tổ chức
- Các HT không cải thiện các vấn đề chiến lược quan trọng
- Quá dư thừa dữ liệu, thiếu chất lượng dữ liệu
- Chi phí bảo trì HT cao
- Nhiều tính năng mong muốn từ ứng dụng

Hoạch định chiến lược tổ chức (tt)

- **Bảng trình bày sứ mệnh:** trình bày sứ mệnh hay nghiệp vụ của một công ty



Figure 4-7
Mission statement (Pine Valley Furniture)

Hoạch định chiến lược tổ chức (tt)

- **Bảng trình bày mục tiêu:** một loạt các phát biểu thể hiện các mục tiêu định tính và định lượng của tổ chức để đạt được vị trí mong muốn trong tương lai.

FIGURE 4-8
Statement of Corporate Objectives
(Pine Valley Furniture)



Hoạch định các HTTT

- Một phương pháp tổ chức tốt việc đánh giá các nhu cầu thông tin của một tổ chức và định nghĩa các hệ thống, các CSDL, và các kỹ thuật sẽ thỏa mãn tốt nhất những yêu cầu đó.
- ISP phải được thực hiện cùng với sự mệnh, các mục tiêu và chiến lược cạnh tranh của tổ chức.

Hoạch định các HTTT (tt)

- **Hoạch định Top-down** nỗ lực để đạt một hiểu biết rộng về những nhu cầu HTTT của toàn tổ chức và cung cấp:
 - Quan điểm rộng.
 - Tích hợp được cải thiện.
 - Hỗ trợ việc quản lý được cải thiện.
 - Hiểu tốt hơn.
- **Hoạch định Bottom-up** nhận diện các DA phát triển HTTT dựa trên việc giải quyết các vấn đề nghiệp vụ vận hành hoặc nhận ra thuận lợi của những cơ hội cụ thể.
 - Có thể nhanh hơn và giảm chi phí, do đó có thể lợi ích trong những tình huống cụ thể.

Hoạch định các HTTT (tt)

FUNCTIONS:	DATA ENTITIES:	INFORMATION SYSTEMS:
<ul style="list-style-type: none">• business planning• product development• marketing and sales• production operations• finance and accounting• human resources...	<ul style="list-style-type: none">• customer• product• vendor• raw material• order• invoice• equipment <p>...</p>	<ul style="list-style-type: none">• payroll processing• accounts payable• accounts receivable• time card processing• inventory management...

FIGURE 4-11

Information systems planning information (Pine Valley Furniture)

Các thành phần của Kế hoạch HTTT

- Sứ mệnh, mục tiêu và chiến lược thuộc tổ chức
 - Mô tả ngắn gọn về sứ mệnh, mục tiêu và chiến lược của tổ chức.
- Kho thông tin
 - Tóm tắt các quy trình, chức năng, thực thể dữ liệu, và nhu cầu thông tin của tổ chức.
- Sứ mệnh và mục tiêu của HTTT
 - HTTT vai trò chính trong tổ chức để chuyển đổi tổ chức từ tình trạng hiện tại sang tình huống tương lai.
- Những ràng buộc khi phát triển HTTT
 - Những hạn chế do kỹ thuật và mức độ tài chính, kỹ thuật và nguồn nhân lực hiện tại .

Các thành phần của Kế hoạch HTTT (tt)

- Nhu cầu về các HT và chiến lược HTTT
 - Tóm tắt nhu cầu các HTTT tổng quan trong công ty và những chiến lược dài hạn (2-5 năm) để đáp ứng yêu cầu.
- Kế hoạch Ngắn hạn
 - Kho chi tiết của các DA và HT hiện tại và kế hoạch chi tiết cho năm hiện tại
- Các kết luận
 - Những sự kiện chưa biết nhưng có thể ảnh hưởng kế hoạch, những yếu tố thay đổi nghiệp vụ hiện tại và tác động của chúng lên kế hoạch.

Khởi động và hoạch định các dự án phát triển HT

- Khởi động DA tập trung vào các hoạt động được thiết kế để trợ giúp việc tổ chức một đội để thực hiện lập kế hoạch DA
 - Lập Đội khởi đầu DA
 - Thiết lập một Mối quan hệ với Khách hàng
 - Lập kế hoạch khởi đầu DA
 - Thiết lập các Thủ tục quản lý
 - Thiết lập môi trường Quản lý DA và tài liệu hướng dẫn DA
 - Phát triển Tôn chỉ DA (Project Charter)

Khởi động và hoạch định các dự án phát triển hệ thống

- Hoạt động chính của hoạch định DA là quá trình định nghĩa các hoạt động rõ ràng, riêng biệt và công việc cần thiết để hoàn thành mỗi hoạt động trong 1 DA đơn.
- Các mục tiêu của quá trình hoạch định DA là phát triển:
 - **Kế hoạch dự án gốc (Baseline Project Plan-BPP)**
 - Tài liệu bên trong (chứa ước tính tốt nhất phạm vi, lợi ích, chi phí, rủi ro và những yêu cầu tài nguyên của một dự án) mục đích chính để hướng dẫn đội phát triển
 - **Bảng trình bày phạm vi dự án (Project Scope Statement)**
 - Được chuẩn bị cho những bên liên quan bên trong và bên ngoài
 - Cung cấp một tổng quan dự án mức cao

Khởi động và hoạch định các dự án phát triển hệ thống

• **Các yếu tố của Lập kế hoạch DA**

- Mô tả phạm vi, những lựa chọn, tính khả thi của DA
- Chia DA thành các tác vụ
- Ước tính những yêu cầu nguồn tài nguyên và tạo kế hoạch nguồn tài nguyên
- Phát triển lịch trình sơ bộ (preliminary schedule)
- Phát triển kế hoạch truyền thông
-

Khởi động và hoạch định các dự án phát triển hệ thống

• **Các yếu tố của Lập kế hoạch DA**

-
- Xác định các tiêu chuẩn và các thủ tục
- Nhận diện và đánh giá rủi ro
- Tạo ngân sách sơ bộ
- Phát triển 1 bảng trình bày công việc
- Thành lập kế hoạch DA gốc (baseline project plan)

Đánh giá tính khả thi của dự án

- 6 danh mục
 - Kinh tế
 - Vận hành/ Hoạt động
 - Kỹ thuật
 - Lịch trình
 - Pháp luật và hợp đồng
 - Chính trị

Đánh giá tính khả thi của dự án

FIGURE 5-2
Yêu cầu dịch vụ hệ thống cho (System Service Request – SSR) cho Customer Tracking System (Pine Valley Furniture)

Công ty nội thất Pine Valley			
Yêu cầu dịch vụ hệ thống			
NGƯỜI YÊU CẦU	Juanita Lopez	NGÀY	2/11/2015
BỘ PHẬN	Bộ phận mua hàng, hỗ trợ sản xuất		
Địa Chỉ	Trụ sở 1-322		
Liên Hệ	Tel: 4-3267 FAX: 4-3270 email: jlopez@pvf.com		
Loại yêu cầu	MỨC ĐỘ KHẨN CẤP		
<input checked="" type="checkbox"/> Hệ thống mới	[] Ngay lập tức – Hoạt động bị ảnh hưởng hoặc thời cơ bị bỏ lỡ		
<input type="checkbox"/> Nâng cấp hệ thống	[] Vẫn để tồn tại, nhưng có thể có cách giải quyết gián tiếp		
<input type="checkbox"/> Sửa lỗi hệ thống	[x] Thiệt hại có thể chấp nhận được cho đến khi có hệ thống mới		

NÊU VẤN ĐỀ

Tăng trưởng doanh số bán hàng của PVF đã tạo ra khối lượng công việc lớn hơn cho đơn vị hỗ trợ sản xuất thuộc bộ phận mua hàng. Ngoài ra, việc tập trung hơn vào dịch vụ khách hàng đã làm giảm thời gian sản xuất, từ đó tăng thêm sức ép lên hoạt động mua hàng.Thêm vào đó, các biện pháp cắt giảm chi phí buộc bộ phận mua hàng phải chịu nhiều sức ép hơn khi thương lượng điều khoản với các nhà phân phối, cải thiện thời gian vận chuyển hàng và giảm đầu tư vào kho hàng. Sự hỗ trợ khiêm tốn của hệ thống hiện tại cho công việc mua hàng và sản xuất không đáp ứng được những điều kiện kinh doanh mới. Hiện tại nếu không có dữ liệu, chúng ta không thể tổng kết thông tin, không thể theo dõi hóa đơn của nhà cung cấp một cách thỏa đáng và việc mua hàng cũng không được hỗ trợ tốt. PVF đang tập trung quá nhiều vào nguyên liệu thô và không đáp ứng tốt các nhu cầu về sản xuất.

YÊU CẦU DỊCH VỤ

Tôi yêu cầu có một phân tích đầy đủ về hoạt động hiện tại với mục tiêu thiết kế và xây dựng một hệ thống thông tin hoàn toàn mới. Hệ thống này sẽ xử lý tất cả các giao dịch mua hàng, hỗ trợ hiển thị và báo cáo nhưng thông tin mua hàng quan trọng và hỗ trợ nhân viên trong hoạt động mua hàng.

NGƯỜI NHẬN	Chris Martin (Tel: 4-6204 FAX: 4-6200 email: cmartin@pvf.com)
BẢO TRỢ	Sall Divario Giám đốc, Bộ phận mua hàng

PHẦN DÀNH CHO BAN ĐÁNH GIÁ ƯU TIÊN HỆ THỐNG

Chấp nhận yêu cầu

Giao cho _____

Ngày bắt đầu _____

Yêu cầu sửa đổi

Đề nghị bộ phận gửi yêu cầu tự phát triển

Từ chối vì lý do: _____

Đánh giá tính khả thi kinh tế

- **Đánh giá tính khả thi kinh tế:** một tiến trình nhận diện những chi phí và lợi nhuận tài chính liên kết với một dự án phát triển
 - Thường liên quan đến 1 phân tích chi phí-lợi ích/lợi nhuận
 - Dự án được xem xét sau mỗi pha SDLC theo thứ tự để quyết định liệu có tiếp tục, chuyển hướng, hoặc dừng một DA
 - Xác định những lợi ích
 - **Những lợi ích hữu hình**
 - **Những lợi ích vô hình**

Đánh giá tính khả thi kinh tế

• **Những lợi ích hữu hình (Tangible benefits)**

- Có thể dễ dàng đo lường bằng tiền và có sự chắc chắn
- Hầu hết những lợi ích hữu hình sẽ phù hợp trong những mục sau:
 - Giảm và tránh chi phí
 - Giảm lỗi
 - Tốc độ các hoạt động tăng
 - Cải thiện hoạch định và kiểm soát việc quản lý
 - Mở những thị trường mới và tăng cơ hội bán hàng

Đánh giá tính khả thi kinh tế

• **Những lợi ích vô hình (Intangible benefits)**

- Những lợi ích đến từ việc tạo một IS điều mà không thể đo lường dễ dàng bằng tiền hoặc với điều chắc chắn
 - Có thể có những lợi ích thuộc tổ chức trực tiếp như cải thiện tinh thần nhân viên
 - Có thể có những quan hệ mật thiết thuộc xã hội rộng hơn như giảm việc tạo hoặc việc sử dụng nguồn lực phí phạm
- vd
 - Tính linh hoạt của tổ chức tăng
 - Tinh thần nhân viên tăng
 - Sự cần thiết cạnh tranh
 - Thông tin kịp thời hơn
 - Đẩy mạnh sự học hỏi và hiểu biết thuộc tổ chức

Đánh giá tính khả thi kinh tế (tt)

- Xác định các chi phí DA gồm
 - **Chi phí hữu hình (Tangible cost)**
 - **Chi phí vô hình (Intangible cost)**
 - **Các chi phí 1 lần (One-time cost)**
 - **Chi phí định kỳ (recurring cost)**

Đánh giá tính khả thi khác

- **Khả thi vận hành**

- Việc đánh giá làm thế nào một hệ thống đã được đề xuất giải quyết những vấn đề kinh doanh hoặc tận dụng cơ hội

- **Khả thi kỹ thuật**

- Việc đánh giá khả năng phát triển của tổ chức để xây dựng một hệ thống được đề xuất.

Đánh giá tính khả thi khác

• **Khả thi kỹ thuật**

- Những hậu quả tiềm năng khi không đánh giá và quản lý rủi ro có thể là:
 - Thất bại để đạt những lợi nhuận mong đợi từ DA
 - Các ước tính chi phí DA không chính xác
 - Các ước tính thời gian DA không chính xác
 - Thất bại trong việc đạt các mức hiệu suất HT thỏa đáng
 - Thất bại khi tích hợp thỏa đáng HT mới với phần cứng, phần mềm, hoặc các thủ tục tổ chức

Các nhân tố rủi ro DA

- Quy mô DA (Project size)
 - Quy mô đội, các phòng ban thuộc tổ chức, khoảng thời gian thực hiện dự án, nỗ lực lập trình Team size, organizational departments, project duration, programming effort
- Cấu trúc DA (Project structure)
 - HT mới vs. nâng cấp, kết quả những thay đổi thuộc tổ chức, cam kết quản lý, nhận thức người dùng
- Nhóm phát triển (Development group)
 - Thân thiện với nền tảng, phần mềm, phương pháp phát triển, lĩnh vực của ứng dụng, phát triển các HT tương tự.
- Nhóm người dùng (User group)
 - Quen thuộc với quy trình phát triển HTTT, lĩnh vực ứng dụng, sử dụng những HT tương tự.

Đánh giá tính khả thi khác

- **Bốn quy tắc chung cho việc đánh giá rủi ro kỹ thuật**
 - *Các DA lớn hơn rủi ro hơn các DA nhỏ*
 - *Một HT mà các yêu cầu đạt được dễ dàng và được cấu trúc ở mức cao sẽ ít rủi ro hơn 1 HT mà các yêu cầu lộn xộn, được cấu trúc yếu, được định nghĩa yếu hoặc lệ thuộc vào sự đánh giá của một cá nhân.*
 - *Sự phát triển của một HT thuê thường được áp dụng hoặc kỹ thuật tiêu chuẩn sẽ ít rủi ro hơn*
 - *Một DA ít rủi ro khi nhóm người dùng quen thuộc với quá trình phát triển HT và lĩnh vực ứng dụng*

Đánh giá tính khả thi khác (tt)

- **Khả thi lịch trình**

- Việc đánh giá khung thời gian và những ngày hoàn thành DA có thỏa mãn những hạn cuối của tổ chức (ứng với những ràng buộc của tổ chức)

- **Khả thi pháp luật và hợp đồng**

- Việc đánh giá sự phân nhánh pháp luật và hợp đồng của hệ thống mới

- **Khả thi chính trị**

- Làm thế nào các bên liên quan chính xem xét HT được đề xuất

Xây dựng kế hoạch dự án gốc

- **Những mục tiêu**

- Đảm bảo rằng khách hàng và nhóm phát triển có một hiểu biết đầy đủ về các yêu cầu và hệ thống được đề xuất
- Cung cấp việc đảm bảo tổ chức có một ý tưởng rõ ràng về phạm vi, lợi nhuận và khoảng thời gian của dự án

- **4 phần của một kế hoạch dự án gốc**

- Giới thiệu
- Mô tả hệ thống
- Đánh giá tính khả thi
- Những vấn đề quản lý

Xem xét kế hoạch dự án gốc

- Những mục tiêu
 - Đảm bảo sự phù hợp với các tiêu chuẩn tổ chức
 - Tất cả các bên đồng ý tiếp tục dự án

Các ứng dụng Thương mại điện tử và Internet cơ bản (tt)

Bảng 4-7. Những điều không biết phải được giải quyết khi thiết kế và xây dựng các ứng dụng Internet

Người sử dụng	Mỗi quan tâm: Ai là người dùng? Ví dụ: Người dùng ở đâu? Mức độ chuyên môn và kinh nghiệm của người sử dụng là gì? Kỳ vọng của người dùng là gì?
Tốc độ kết nối	Mỗi quan tâm: Tốc độ kết nối và thông tin nào có thể được hiển thị hiệu quả? Ví dụ: modem, modem cáp, vệ tinh, băng thông rộng, di động
Truy cập / Phương pháp	Mỗi quan tâm: Cách truy cập Internet là gì? Ví dụ: trình duyệt Web, máy tính bảng (iPad), điện thoại thông minh, truyền hình trực tuyến

PVF - System Planning and Selection

- WebStore của Pine Valley Furniture
- Khởi tạo và Hoạch định Hệ thống TMĐT của PVF
- Kiểm thử từ đầu đến cuối Dự án WebStore

WebStore của Pine Valley Furniture

Nghiên cứu thị trường đã tìm ra cơ hội tốt cho mua đồ nội thất trực tuyến, đặc biệt là trong các lĩnh vực:

- Mua đồ nội thất cho công ty
- Mua đồ nội thất trong nhà-văn phòng
- Mua đồ nội thất dành cho sinh viên

Khởi tạo và Hoạch định Hệ thống TMĐT của PVF

Bảng 4-8. Các chi phí Hệ thống dựa trên Web

Phân loại chi phí	Ví dụ
Các chi phí nền tảng	Dịch vụ lưu trữ web Máy chủ web Phần mềm máy chủ Phần mềm plug-in Máy chủ tường lửa Router Kết nối Internet
Nội dung và dịch vụ	Thiết kế và phát triển sáng tạo Phi thiết kế đang thực hiện Quản lý dự án Web Quản lý trang web kỹ thuật Nhân viên nội dung Nhân viên đồ họa Nhân viên hỗ trợ Quỹ tăng cường trang web Lệ phí cấp giấy phép bên ngoài nội dung Lập trình, tư vấn và nghiên cứu Đào tạo và du lịch
Tiếp thị	Thư trực tiếp Khởi động và quan hệ công chúng In quảng cáo Liên kết phải trả tiền đến các trang web khác Khuyến mãi Nhân viên tiếp thị Nhân viên bán hàng quảng cáo

Kiểm thử từ đầu đến cuối Dự án WebStore

Bảng 4-9. Những lợi ích và chi phí Dự án WebStore của PVF

Những lợi ích hữu hình	Những lợi ích vô hình
Chi phí chìm mỗi giao dịch thấp hơn	Đầu tiên với thị trường
Lắp lại nghiệp vụ	Nền tảng cho hệ thống thông tin đầy đủ dựa trên Web
Chi phí hữu hình (một-lần)	Sự đơn giản đối với các khách hàng
Phi thiết lập dịch vụ Internet	Chi phí vô hình
Phản ứng	Không có sự tương tác mặt đối mặt
Chi phí phát triển	Không phải tất cả khách hàng sử dụng Internet
Nhập dữ liệu	
Chi phí hữu hình (định kỳ)	
Phi dịch vụ máy chủ dịch vụ Internet	
Phản hồi	
Hỗ trợ	
Bảo trì	
Bán hàng giảm đi qua các kênh truyền thống	

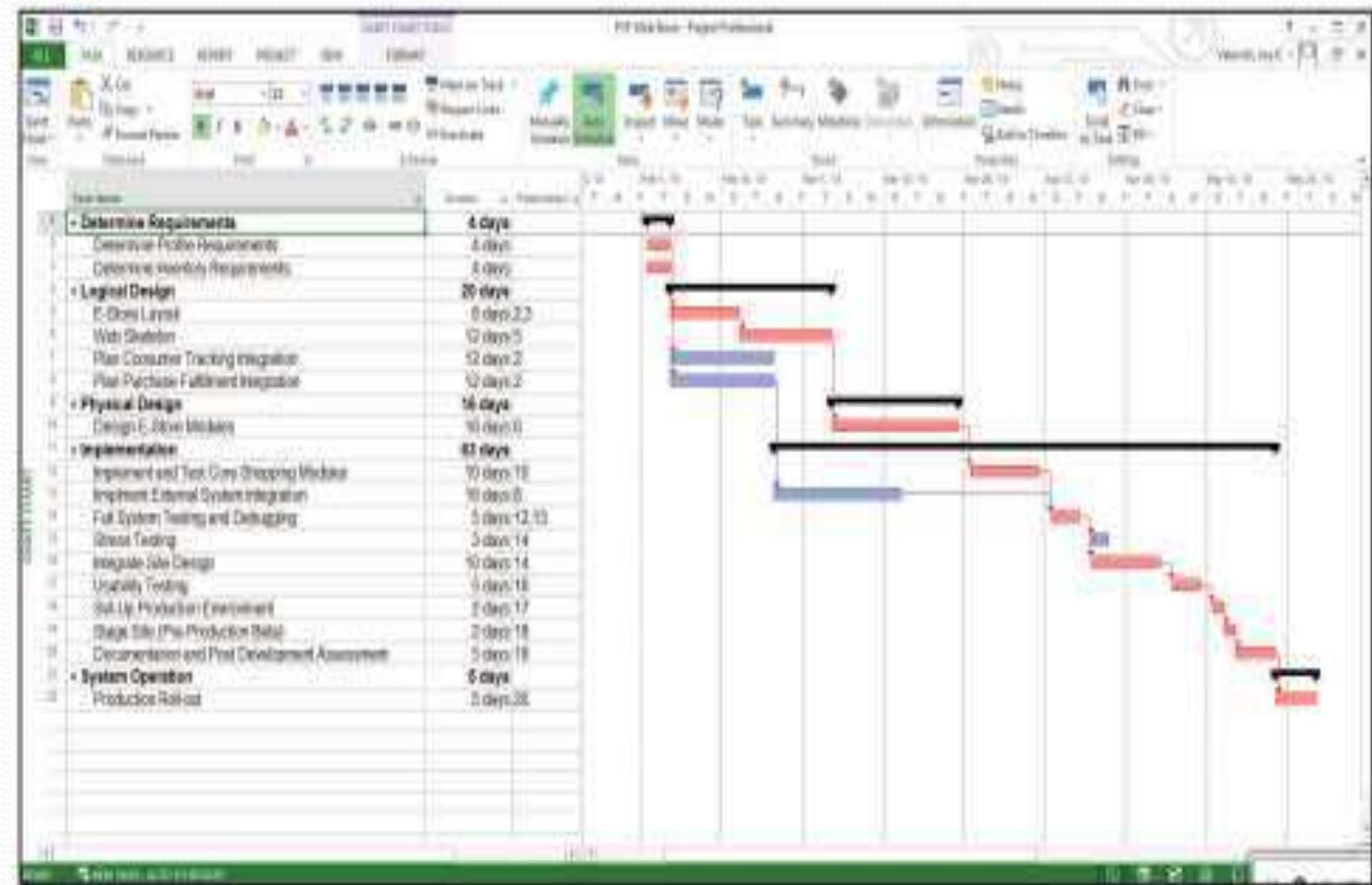
Kiểm thử từ đầu đến cuối Dự án WebStore

Bảng 4-10. Những quan tâm về tính khả thi WebStore PVF

Mỗi quan tâm tính khả thi	Mô tả
Vận hành	Cửa hàng trực tuyến mở 24/7/365; trả lại / hỗ trợ khách hàng
Kỹ thuật	Thiết lập kỹ năng mới để phát triển, bảo trì và vận hành
Lịch trình	Phải cởi mở cho nghiệp vụ theo Q3
Pháp lý	Gian lận thẻ tín dụng
Chính trị	Kênh phân phối truyền thống đánh mất nghiệp vụ

Kiểm thử từ đầu đến cuối Dự án WebStore

Hình 4-19. Biểu đồ Gantt biểu diễn lịch trình cho dự án WebStore



Chương 3. Phân tích các hệ thống

- 3.1 Việc xác định các yêu cầu hệ thống
- 3.2 Cấu trúc các yêu cầu hệ thống:
 - Phân tích có cấu trúc
 - FDD- *Functional Decomposition Diagram* / BFD- *Business Function Diagram*
 - DFD- *Data Flow Diagram*
 - ERD- *Entity Relationship Diagram*
 - Phân tích hướng đối tượng
 - Use case, Activity Diagram, Sequence Diagram
 - Class Diagram

Thực hiện việc xác định những yêu cầu

- Tập hợp thông tin về những gì HT nên làm từ nhiều nguồn
 - Những người dùng
 - Các báo cáo
 - Các biểu mẫu
 - Các thủ tục

Thực hiện việc xác định những yêu cầu (tt)

- **Các loại sản phẩm bàn giao**

- Thông tin được tập hợp từ những người dùng
- Những tài liệu và tập tin đang có
- Thông tin dựa trên máy tính
- Hiểu rõ các thành phần tổ chức
 - Mục tiêu kinh doanh
 - Nhu cầu thông tin
 - Các quy tắc xử lý dữ liệu
 - Những sự kiện chính

Loại sản phẩm bàn giao	Những sản phẩm cụ thể
Thông tin được tập hợp từ những cuộc nói chuyện với những người dùng	<p>Những bản chép lại cuộc phỏng vấn</p> <p>Những ghi chú từ những buổi quan sát</p> <p>Những phiếu họp</p>
Những tài liệu và tập tin hiện có	<p>Sứ mệnh kinh doanh và bản tuyên bố chiến lược</p> <p>Mẫu các bảng biểu và báo cáo nghiệp vụ, và các hiển thị máy tính</p> <p>Cẩm nang trình tự thao tác</p> <p>Những mô tả chi tiết công việc</p> <p>Sách huấn luyện</p> <p>Các sơ đồ luồng và tài liệu của những HT đang tồn tại</p> <p>Các báo cáo của chuyên viên</p>
Thông tin dựa máy tính	<p>Các kết quả từ các phiên họp JAD</p> <p>Các báo cáo và nội dung kho CASE của HT đang tồn tại</p> <p>Các màn hình hiển thị và báo cáo từ các mẫu thử HT</p>

Những PP truyền thông cho việc xác định các yêu cầu

PP truyền thông	Các hoạt động liên quan
Phỏng vấn riêng lẻ	Các buổi phỏng vấn riêng lẻ cung cấp thông tin về hoạt động và vấn đề của HT hiện tại và nhu cầu của các HT trong các hoạt động tổ chức tương lai
Các buổi quan sát các nhân viên	Quan sát các nhân viên tại những thời điểm được lựa chọn để nhìn thấy làm thế nào dữ liệu được điều khiển và thông tin gì mọi người cần đến cho công việc của họ
Các tài liệu nghiệp vụ	Nghiên cứu các tài liệu nghiệp vụ để khám phá những vấn đề, chính sách, nguyên tắc và phương hướng được báo cáo cũng như như những ví dụ cụ thể của việc sử dụng dữ liệu và thông tin trong tổ chức

Những PP truyền thống cho việc xác định các yêu cầu (tt)

- **Phỏng vấn và Lắng nghe**

- Một trong những cách chính để các phân tích viên tập hợp thông tin về 1 DA các HTTT
- Một hướng dẫn phỏng vấn (interview guide) là một tài liệu để phát triển, hoạch định và thực hiện một phỏng vấn.

Interview Outline	
Interviewee: Name of person being interviewed	Interviewer: Name of person leading interview
Location/Medium: Office, conference room, or phone number	Appointment Date: Start Time: End Time:
Objectives: What data to collect On what to gain agreement What areas to explore	Reminders: Background/experience of interviewee Known opinions of interviewee
Agenda: Introduction Background on Project Overview of Interview Topics to Be Covered Permission to Record Topic 1 Questions Topic 2 Questions ... Summary of Major Points Questions from Interviewee Closing	Approximate Times: 1 minute 2 minutes 1 minute 5 minutes 7 minutes 2 minutes 5 minutes 1 minute
General Observations: Interviewee seemed busy probably need to call in a few days for follow-up questions because he gave only short answers. PC was turned off — probably not a regular PC user.	
Unresolved Issues, Topics Not Covered: He needs to look up sales figures from 1999. He raised the issue of how to handle returned goods, but we did not have time to discuss.	
Interviewee:	Date:
Questions:	Notes:
When to ask question, if conditional Question: 1 Have you used the current sales tracking system? If so, how often?	Answer: Yes, I ask for a report on my product line weekly. Observations: Seemed anxious — may be overestimating usage frequency.
If yes, go to Question 2	
Question: 2 What do you like least about the system?	Answer: Sales are shown in units not dollars. Observations: System can show sales in dollars, but user does not know this.

Những PP truyền thống cho việc xác định các yêu cầu (tt)

- **Phỏng vấn và Lắng nghe**

- Tập hợp các sự kiện, ý kiến, và suy đoán
- Quan sát ngôn ngữ cơ thể và những xúc cảm
- *Nguyên tắc (Xem chi tiết Bảng 5-3)*
- Các câu hỏi phỏng vấn:
 - **Câu hỏi đóng (Closed-Ended)**
 - **Câu hỏi mở (Open-Ended)**

Những PP truyền thống cho việc xác định các yêu cầu (tt)

- **Phỏng vấn và Lắng nghe (tt)**

- **Câu hỏi đóng (Closed-Ended)**

- Người trả lời được yêu cầu chọn từ một tập những câu trả lời đã xác định
- Thực hiện tốt khi những câu trả lời phổ biến đối với những câu hỏi được biết
- Không yêu cầu thời gian dài và có thể bao quát một lượng chủ đề lớn hơn.
- Vd
 - Có bao nhiêu đặt hàng nhận qua điện thoại mỗi ngày?
 - Các KH đặt hàng như thế nào?
 - Những thông tin thêm nào bạn muốn HT mới cung cấp?

Những PP truyền thống cho việc xác định các yêu cầu (tt)

- **Phỏng vấn và Lắng nghe** (tt)

- **Câu hỏi mở (Open-Ended)**

- Không có những câu trả lời được xác định trước
- Được dùng để thăm dò cho những câu trả lời bất ngờ
- Vd.
 - Bạn nghĩ gì về HT hiện tại?
 - Những khó khăn nào bạn đối mặt hàng ngày?
 - Tại sao
 - Bạn có thể đưa ra một ví dụ không?
 - Bạn có thể giải thích chi tiết hơn không?

Những PP truyền thống cho việc xác định các yêu cầu (tt)

Bảng 5-3

Các nguyên tắc	Những gì được đòi hỏi
Kế hoạch buổi phỏng vấn	Chuẩn bị người được phỏng vấn bằng cách hẹn gặp và giải thích mục đích buổi phỏng vấn. Chuẩn bị 1 ds kiểm tra, 1 đề tài thảo luận và các câu hỏi
Trung tính	Tránh hỏi những câu hỏi dẫn đến câu trả lời theo ý muốn
Nghe và ghi chú	Tập trung hoàn toàn vào người được phỏng vấn và ghi chú hoặc ghi âm lại buổi phỏng vấn (nếu được cho phép)
Xem xét lại các ghi chú	Xem xét lại bản ghi chú trong vòng 48 giờ của buổi gặp mặt. nếu bạn khám phá các câu hỏi tiếp theo hoặc cần thông tin thêm thì liên lạc người được phỏng vấn
Tìm kiếm những quan điểm đa dạng	Phỏng vấn phạm vi nhiều người, gồm những người dùng và những nhà quản lý tiệm năn

Các nhóm phỏng vấn

- **Những hạn chế của các cuộc phỏng vấn cá nhân:**
 - Những mâu thuẫn và không thống nhất giữa những người được phỏng vấn.
 - Những thảo luận tiếp theo đòi hỏi nhiều thời gian.
 - Những cuộc phỏng vấn mới có thể tiết lộ những câu hỏi mới cần phỏng vấn thêm với những người đã thực hiện trước đó.

Các nhóm phỏng vấn (tt)

- Phỏng vấn vài người chính cùng nhau:
 - Thuận lợi
 - Hiệu quả hơn về thời gian
 - Có thể nghe sự đồng thuận hoặc mâu thuẫn cùng lúc
 - Cơ hội cho sự hợp lực
 - Bất lợi
 - Khó khăn hơn lập lịch phỏng vấn cá nhân

Kỹ thuật nhóm danh nghĩa (*Nominal Group Technique - NGT*)



- Một quy trình tạo điều kiện hỗ trợ việc tạo ý tưởng bởi các nhóm
- Quy trình
 - Các thành viên đến cùng nhau như một nhóm, nhưng ban đầu làm việc tách riêng.
 - Mỗi người viết các ý kiến.
 - Người điều hành đọc lớn các ý kiến, và chúng được viết lên bảng đen hoặc bảng flipchart.
 - Nhóm thảo luận mở để làm rõ các ý kiến.
 - Các ý kiến được xếp ưu tiên, nối kết, lựa chọn, giản lược.
- Được dùng để bổ sung các cuộc họp nhóm hoặc như một phần của nỗ lực JAD.

Những PP truyền thống cho việc xác định các yêu cầu (tt)

- **Quan sát trực tiếp những người dùng**

- Xem những người dùng thực hiện những công việc của họ.
- Được dùng để thu về những đánh giá trực tiếp và khách quan hơn từ sự tương tác nhân viên với những HTTT
- Có thể gây cho mọi người thay đổi những hành vi hoạt động bình thường
- Đòi hỏi nhiều thời gian và thời gian hạn chế để quan sát

- **Phân tích các thủ tục và Những tài liệu khác**

- Xem xét những tài liệu nghiệp vụ đang có
- Có thể đưa ra một xem xét lịch sử và chính thức cho các yêu cầu HT

Những PP truyền thống cho việc xác định các yêu cầu (tt)

- **Phân tích các thủ tục và Những tài liệu khác**

- **Những loại thông tin được phát hiện**

- Những vấn đề với hệ thống hiện tại
- Cơ hội để đáp ứng nhu cầu mới
- Định hướng thuộc tổ chức
- Chức danh và tên của những nhân vật chính
- Các giá trị của tổ chức hoặc những cá nhân
- Các tình huống xử lý thông tin đặc biệt
- Các lý do cho việc thiết kế HT hiện tại
- Dữ liệu, các quy tắc xử lý dữ liệu

Những PP truyền thống cho việc xác định các yêu cầu (tt)

- **Phân tích các thủ tục và Những tài liệu khác**

- 4 loại tài liệu hữu ích cho xác định các yêu cầu:
 - **Những thủ tục công việc được ghi chép lại**
 - **Biểu mẫu nghiệp vụ**
 - **Báo cáo**
 - **Mô tả HTTT hiện tại**

Những PP truyền thống cho việc xác định các yêu cầu (tt)

- **Phân tích các thủ tục và Những tài liệu khác**

- **Những thủ tục công việc được ghi chép lại**

- Đối với một cá nhân hoặc nhóm công việc
- Mô tả một công việc hoặc một tác vụ cụ thể được thực hiện như thế nào
- Gồm dữ liệu và thông tin đã dùng và tạo trong quy trình

- **Những vấn đề tiềm ẩn với các tài liệu thủ tục**

- Có thể liên quan những kết quả đạt được trùng nhau
- Có thể có những thủ tục thắt lạc
- Có thể là quá hạn
- Có thể đạt những thông tin mâu thuẫn qua các cuộc phỏng vấn.

Những PP truyền thống cho việc xác định các yêu cầu (tt)

- **Phân tích các thủ tục và Những tài liệu khác**
- **Những HT chính thức:** cách chính thức một HT làm việc như đã mô tả trong tài liệu tổ chức (ie. Thủ tục công việc)
- **Những HT không chính thức:** cách một HT làm việc thực sự (ie. Các cuộc phỏng vấn, quan sát)

Những PP truyền thống cho việc xác định các yêu cầu (tt)

- **Phân tích các thủ tục và Những tài liệu khác**

- **Biểu mẫu nghiệp vụ**

- Được sử dụng cho tất cả các loại chức năng nghiệp vụ
- Chỉ ra rõ ràng luồng dữ liệu vào hoặc ra của một HT và dữ liệu cần thiết cho HT đối với chức năng
- Đưa ra thông tin quan trọng về bản chất của tổ chức

- **Báo cáo**

- Kết quả đầu ra (output) chính của HT hiện tại
- Cho phép phân tích viên làm việc ngược từ báo cáo đến dữ liệu cần thiết để tạo ra nó.

- **Mô tả của HTTT hiện tại**

- Làm thế nào những HTTT được thiết kế và chúng hoạt động như thế nào

TABLE 6-4 Comparison of Observation and Document Analysis

Đặc điểm	Quan sát	Phân tích tài liệu
Giàu thông tin	Cao (nhiều kênh)	Thấp (bị động) và cũ
Thời gian cần thiết	Có thể được bao quát	Thấp đến vừa phải
Chi phí	Có thể cao	Thấp đến vừa phải
Cơ hội theo và phát hiện sự thật	Tốt: cơ hội phát hiện sự thật và lọc các câu hỏi trong hoặc sau khi quan sát	Bị hạn chế: phát hiện sự thật chỉ khi người tạo ra bản gốc vẫn còn
Bí mật	Người bị quan sát biết thì có thể thay đổi hành vi khi được quan sát	Phụ thuộc vào bản chất của tài liệu, không thay đổi khi được đọc
Sự tham gia của đối tượng	Sự tham gia của người được quan sát phụ thuộc vào việc liệu họ có biết mình đang bị quan sát hay không	Không, không có điều ràng buộc rõ ràng
Đối tượng tiềm năng	Số lượng đối tượng và thời gian bị hạn chế	Khả năng chênh lệch theo tài liệu được giữ hoặc bởi vì tài liệu không được tạo ra cho mục đích này.

Những PP hiện đại cho việc xác định các yêu cầu

- **Joint Application Design-JAD**
- **Group Support Systems**
- **CASE tools**
- **System Prototypes**

Những PP hiện đại cho việc xác định các yêu cầu

- **Joint Application Design-JAD**

- Tập hợp những người dùng chính, những nhà quản lý, và các nhà phân tích hệ thống ngồi cùng nhau.
- Mục đích: tập hợp những yêu cầu hệ thống đồng thời từ những người chính
- Được tiến hành bên ngoài

Joint Application Design-JAD

- **Thành phần tham gia**

- Người lãnh đạo phiên họp
- Những người dùng
- Các nhà quản lý
- Nhà tài trợ
- Các nhà phân tích hệ thống
- Người sao chép bản thảo
- Bộ phận IS

- **Kết quả cuối cùng**

- Tài liệu chi tiết HT đang có
- Những đặc tính của HT đề xuất

Joint Application Design-JAD

- CASE Tools trong các phiên JAD
 - Các công cụ Upper CASE được sử dụng
 - Cho phép phân tích viên nhập các mô hình HT trực tiếp vào CASE trong suốt phiên JAD
 - Những thiết kế màn hình và mẫu thử có thể được làm trong JAD và được biểu diễn cho những người dùng

Những PP hiện đại cho việc xác định các yêu cầu

- **Prototyping**

- Quy trình lặp đi lặp lại
- Phiên bản làm việc thô sơ của HT được xây dựng
- Thay thế hoặc tăng cường SDLC
- Phát triển những đặc tả cụ thể hơn cho hệ thống cuối cùng
- Nhanh chóng chuyển các yêu cầu sang phiên bản làm việc của hệ thống
- Cho phép người dùng nhìn thấy những yêu cầu được chuyển sang hệ thống → sẽ yêu cầu chỉnh sửa hoặc sẽ tạo ra những yêu cầu thêm vào.

PP căn bản cho việc xác định các yêu cầu HT

Tái cơ cấu quy trình nghiệp vụ (Business Process Reengineering-BPR)

- Tìm kiếm và thực hiện thay đổi triệt để trong các quy trình nghiệp vụ để đạt được những cải tiến đột phát trong các sản phẩm và dịch vụ
- Các mục tiêu
 - Nhận ra luồng dữ liệu hoàn chỉnh trong những phần chính của một tổ chức
 - Bỏ qua các bước không cần thiết
 - Kết hợp các bước
 - Trở nên nhanh nhạy hơn với thay đổi tương lai

Tái cơ cấu quy trình nghiệp vụ (Business Process Reengineering-BPR)

- Nhận ra các tiến trình để tái cấu trúc
 - Các tiến trình nghiệp vụ chính
 - Tập hợp các hoạt động được cấu trúc, được cân nhắc để tạo ra đầu ra cụ thể cho một khách hàng hoặc thị trường đặc thù
 - Tập trung vào các khách hàng và kết quả
 - Những kỹ thuật giống nhau được dùng khi xác định các yêu cầu
 - Nhận ra những hoạt động cụ thể có thể được cải tiến qua BPR

Các kỹ thuật đột phá - Disruptive Technologies

- Công nghệ thông tin phải được áp dụng để cải tiến hoàn toàn triệt để các tiến trình nghiệp vụ
- Các kỹ thuật đột phá *Bảng hình bên*
 - Những kỹ thuật cho phép phá vỡ những nguyên tắc nghiệp vụ lâu đời cản trở các tổ chức tạo ra những thay đổi nghiệp vụ triệt để.

Luật	Kỹ thuật đột phá
Thông tin có thể xuất hiện chỉ trong 1 nơi tại 1 thời điểm	Các CSDL phân tán cho phép chia sẻ thông tin
Chỉ các chuyên gia có thể thực hiện công việc phức tạp	Các HT chuyên gia có thể trợ giúp khi không có chuyên gia
Các nghiệp vụ phải chọn giữa sự tập trung và sự phân quyền	Các mạng viễn thông nâng cao có thể hỗ trợ các cấu trúc tổ chức động
Các nhà quản lý phải ra tất cả các quyết định	Các công cụ hỗ trợ quyết định có thể trợ giúp khi không có các nhà quản lý
Các nhân viên theo lĩnh vực cần các văn phòng nơi họ có thể nhận, lưu trữ, khôi phục, và chuyển thông tin	Truyền thông dữ liệu không dây và các máy tính xách tay cung cấp 1 văn phòng ảo cho các nhân viên
Liên lạc tốt nhất với 1 người mua tiêm nǎng là liên lạc cá nhân	Các kỹ thuật liên lạc tương tác cho phép khả năng gửi tin nhắn phức tạp
Bạn phải tìm ra các vật ở đâu	Kỹ thuật nhận dạng tự động và dò tìm cho biết các vật ở đâu
Các kế hoạch phải sửa lại thường xuyên	Tính toán năng suất-cao có thể cung cấp sự cập nhật theo thời gian thực.

Các ƯD TMĐT: Xác định các yêu cầu

- **Xác định các yêu cầu HT cho WebStore của PVF gồm:**

- Đặc điểm bối cảnh và điều hướng trang web
- WebStore và khả năng HT quản lý trang web
- Thông tin khách hàng và kho hàng
- Sự phát triển mẫu thử Website
- Các yêu cầu ứng dụng điện thoại thông minh

Pine Valley Funiture WebStore: Xác định các yêu cầu HT

• **Đặc điểm bối cảnh và điều hướng trang web**

- Một số trang web bán lẻ được yêu cầu truy cập bởi những người tham gia trong phiên họp JAD
 - www.amazon.com, www.landsend.com, www.sony.com, và www.pier1.com
- Những người tham gia xác định và thảo luận về các tính năng mong muốn WebStore sở hữu
- Các kết quả của hoạt động này được tóm tắt trong

Bảng 6-8

BẢNG 6-8 Đặc tính điều hướng và bố cục mong muốn của WebStore

Khung nhìn và Thiết kế

Vị trí menu điều hướng và logo vẫn phù hợp trong toàn bộ trang web (điều này cho phép người dùng vẫn quen thuộc khi sử dụng trang web và giảm thiểu số người quên trang web).

Đồ họa phải nhẹ để cho phép hiển thị trang nhanh.

Văn bản nên được sử dụng đồ họa bắt cứ khi nào có thể.

Điều hướng

Có thể truy cập bất kỳ phần nào của cửa hàng qua menu điều hướng.

Người dùng luôn nhận thức được phần nào họ đang ở tại đó.

Pine Valley Funiture WebStore: Xác định các yêu cầu HT

- **WebStore và khả năng HT quản lý trang web**
 - Cấu trúc của cả hệ thống WebStore và hệ thống Quản lý trang Web (Bảng 6-9) sẽ được trao cho bộ phận hệ thống thông tin làm cơ sở để tiếp tục các hoạt động phân tích và thiết kế.

BẢNG 6-9 Cấu trúc HT của WebStore và các HT Quản lý trang web

<i>Hệ thống WebStore</i>	<i>Hệ thống quản lý trang web</i>
Trang chính	Người quản lý hồ sơ người dùng
Dòng sản phẩm (danh mục)	Người quản lý việc duy trì đơn hàng
<ul style="list-style-type: none"> - Bàn làm việc - Ghế - Bàn ăn - Tủ hồ sơ 	Người quản lý nội dung (danh mục)
Giò hàng	Các báo cáo
Thanh toán	Tổng số lượt truy cập
Hồ sơ tài khoản	Lượt xem trang thường xuyên nhất
Trạng thái đơn đặt hàng / lịch sử	Người dùng / thời gian trong ngày
Nhận xét của khách hàng	Người dùng / ngày trong tuần
Thông tin công ty	Người mua sắm không mua hàng (đã sử dụng giò hàng - không thanh toán)
Phản hồi	Phân tích phản hồi
Thông tin liên lạc	

Pine Valley Funiture WebStore: Xác định các yêu cầu HT

- **Thông tin khách hàng và kho hàng**

- WebStore sẽ được thiết kế để hỗ trợ việc mua đồ gỗ của ba loại khách hàng khác nhau:
 - Khách hàng doanh nghiệp
 - Khách hàng gia đình – văn phòng
 - Khách hàng sinh viên
- Bảng 6-10 tóm tắt thông tin này cho từng loại khách hàng đã được xác định trong phiên JAD

BẢNG 6-10 Thông tin Khách hàng và Kho hàng cho WebStore

Khách hàng doanh nghiệp	Khách hàng Gia đình-Văn phòng	Khách hàng Sinh viên	Thông tin Kho hàng
Tên công ty	Tên	Tên	SKU
Địa chỉ công ty	Kinh doanh về (tên công ty)	Trường học	Tên
Điện thoại công ty	Địa chỉ nhà	Địa chỉ	Mô tả
Số fax Công ty	Điện thoại	Điện thoại	Kích thước sản phẩm hoàn thiện
Phương thức chuyên hàng ưa thích	Số fax E-mail	E-mail	Trọng lượng sản phẩm đã hoàn thiện
Tên người mua			Tài liệu sẵn có
Điện thoại người mua			Màu sẵn có
E-mail người mua			Giá bán
			Thời gian chờ hàng

Pine Valley Funiture WebStore: Xác định các yêu cầu HT

• **Sự phát triển mẫu thử Website**

- Những thành viên JAD đã đồng ý rằng sau khi hoàn thành các hoạt động phân tích và thiết kế, việc thực thi hệ thống nên tiến triển trong ba giai đoạn chính để những thay đổi yêu cầu có thể được xác định và thực hiện dễ dàng hơn.
- Bảng 6-11 tóm tắt các giai đoạn và các chức năng kết hợp ở mỗi một giai đoạn

BẢNG 6-11 Các giai đoạn Thực thi HT của WebStore

Giai đoạn 1 (Chức năng cơ bản)

Định vị danh mục đơn giản; hai sản phẩm cho mỗi phân – tập tính phân giới hạn 25 mẫu người dùng

Giao dịch thẻ tín dụng già lập

Chức năng giờ hàng đầy đủ

Giai đoạn 2 (Xem và Cảm nhận)

Tập thuộc tính sản phẩm đầy đủ và phương tiện (hình ảnh, video) - thường được gọi là "sản phẩm"

Danh mục dữ liệu "

Bố trí trang web đầy đủ

Tích hợp mô phỏng với thực hiện mua hàng và hệ thống theo dõi khách hàng

Giai đoạn 3 (Giai đoạn / Chuẩn bị sản xuất)

Tích hợp hoàn chỉnh với hệ thống hoàn tất việc mua hàng và hệ thống theo dõi khách hàng

Tích hợp đầy đủ việc xử lý thẻ tín dụng

Danh mục sản phẩm đầy đủ

Pine Valley Funiture WebStore: Xác định các yêu cầu HT

- **Các yêu cầu ứng dụng điện thoại thông minh**

- Các thành viên JAD đã xác định các yêu cầu cho một ứng dụng điện thoại thông minh gắn với WebStore
- Chức năng chính của ứng dụng (Bảng 5-10).
- Xây dựng kính tăng cường thực tế ảo vào ứng dụng

Bảng 5-10. Các yêu cầu của ứng dụng WebStore trên điện thoại thông minh

Ứng dụng nên cho phép người dùng:

1. Duyệt qua sản phẩm theo màu sắc, giá cả và chủng loại
2. Xem danh mục trực tuyến
3. Tải xuống các danh mục (cho Skydrive, Google Drive, v.v.) để xem ngoại tuyến
4. Xác định vị trí cửa hàng PVF gần nhất của họ
5. Trực quan đồ nội thất với tăng cường thực tế ảo, cho phép người dùng lựa chọn một mặt hàng nội thất từ danh mục và đặt nó trong một căn phòng của ngôi nhà của họ bằng cách sử dụng một máy ảnh điện thoại thông minh

Phân tích các hệ thống

• 3.2 Cấu trúc các yêu cầu HT: Mô hình hóa quy trình

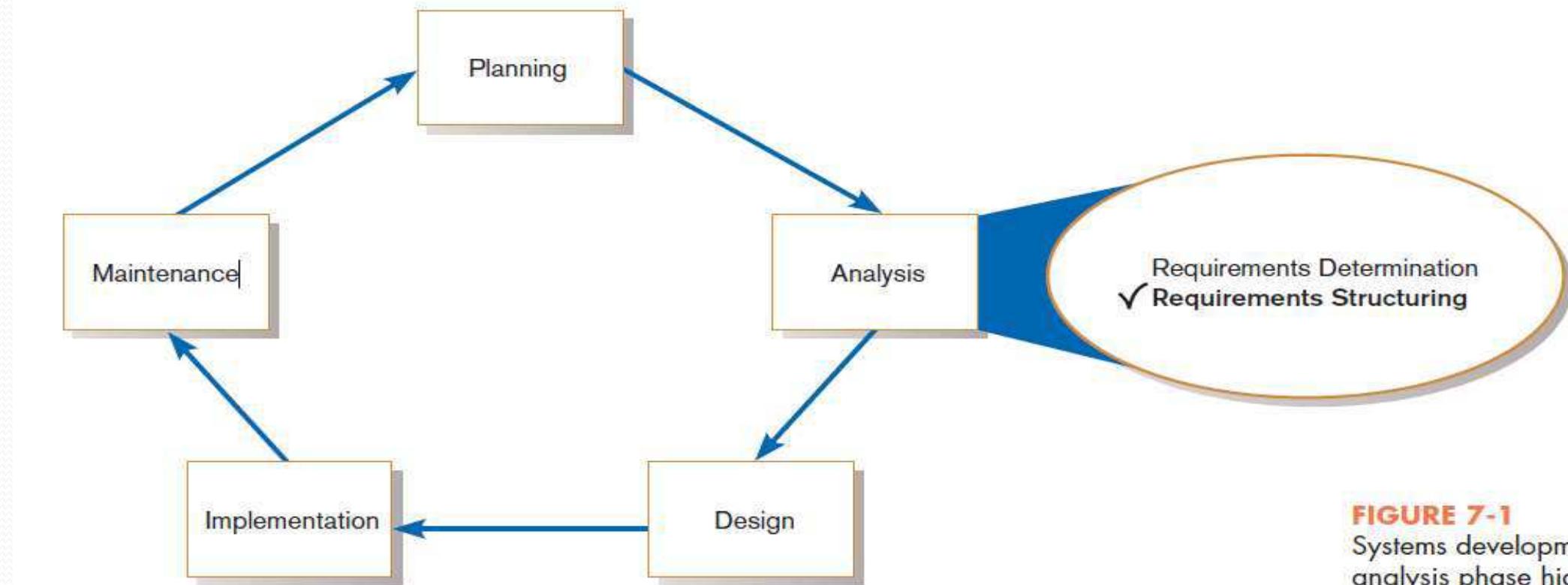


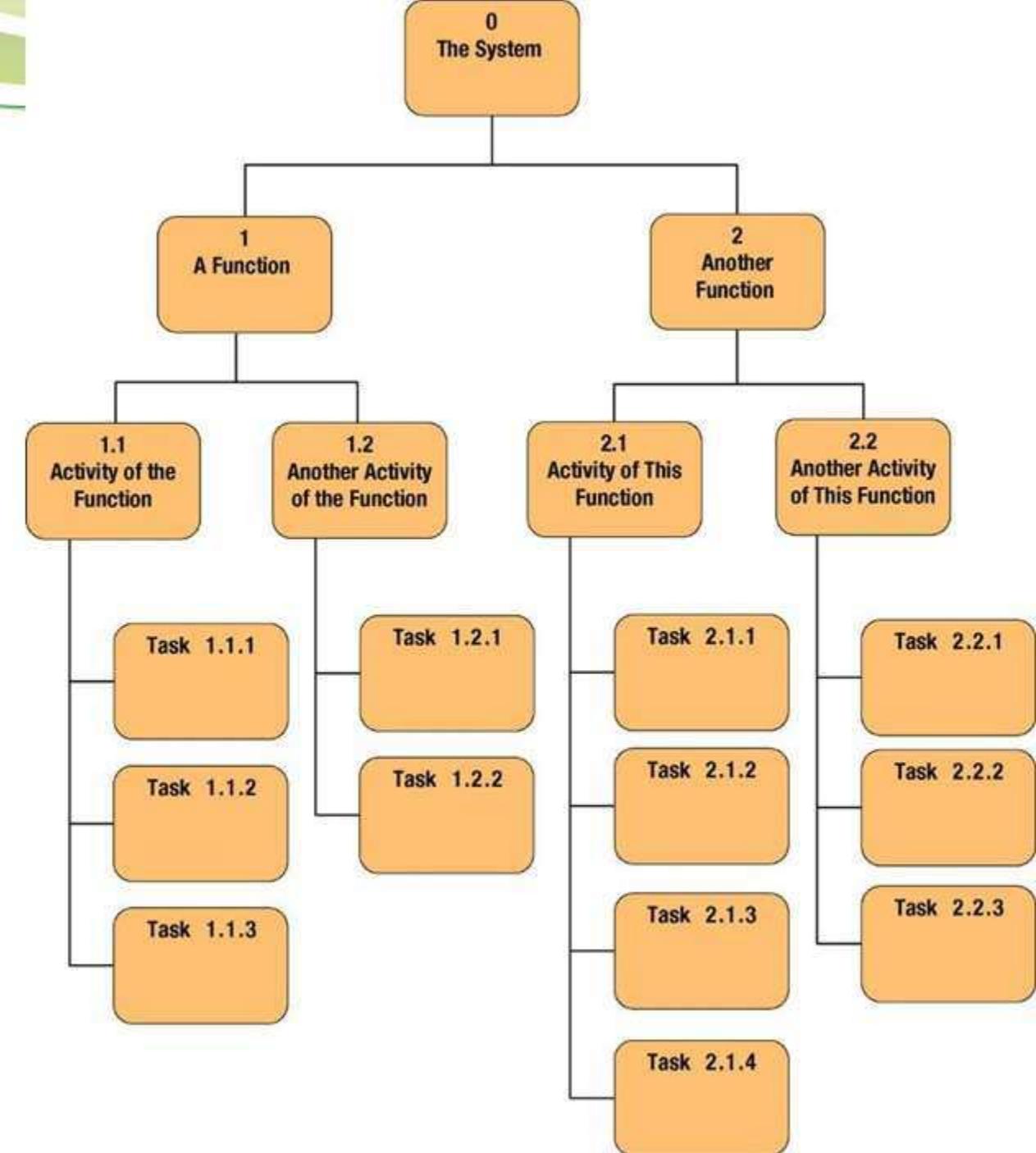
FIGURE 7-1
Systems development analysis phase highlight

Các sơ đồ phân rã chức năng (Functional Decomposition Diagrams - FDD)

- Một sơ đồ phân rã chức năng (*FDD*) hoặc đồ thị phân cấp (*hierarchy chart*) biểu diễn sự phân rã chức năng của một HT
(hoặc Sơ đồ chức năng nghiệp vụ - Business Function Diagram-BFD)
- Cho phép xác định phạm vi các chức năng hay miền cần nghiên cứu của tổ chức
- Cho thấy vị trí của mỗi chức năng trong toàn HT, giúp phát hiện các chức năng thiếu, thừa
- Mỗi chức năng được phân rã phải là một bộ phận thực sự tham gia thực hiện chức năng đã phân rã ra nó
- Ở mỗi mức, các chức năng cùng mức sắp xếp trên cùng một hàng. Mức cuối cùng có thể sắp xếp theo hàng dọc.

Các sơ đồ phân rã chức năng (FDDs)

- Các sơ đồ phân rã chức năng có thể được xây dựng bằng cách tiếp cận từ trên xuống (top down) hoặc từ dưới lên (bottom up)
- Sự phân rã các chức năng và các tiến trình tương ứng việc phân mực của các sơ đồ luồng dữ liệu (Data Flow Diagram)



Các sơ đồ phân rã chức năng (FDDs)

- **Đặt tên chức năng**

- Mỗi chức năng có một tên duy nhất
- Tên chức năng = **Động từ + Bổ ngữ**

- **Mô tả chi tiết chức năng ở mức cuối**

- Tên chức năng
- Các sự kiện kích hoạt
- Quy trình thực hiện
- Dữ liệu vào, ra
- Công thức tính toán sử dụng (Nếu có)
- Quy tắc nghiệp vụ cần tuân thủ

Sơ đồ phân rã chức năng-FDD

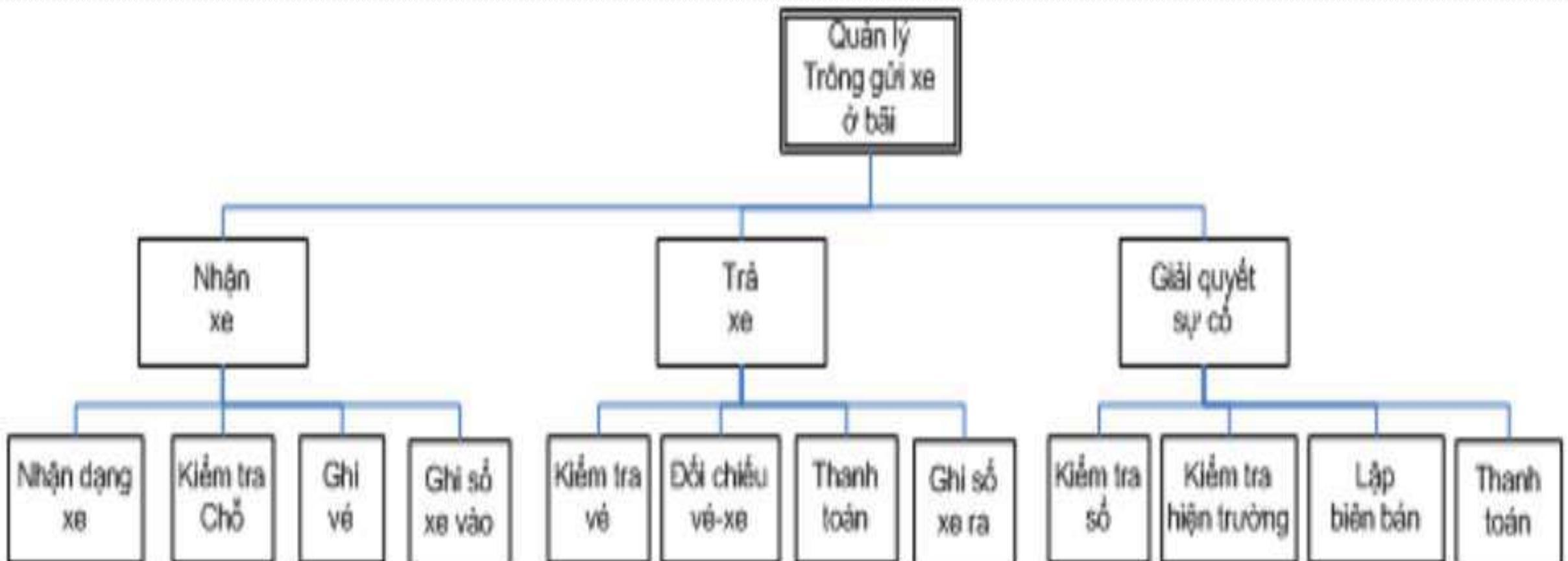
Cách tiếp cận từ dưới lên

- **Ví dụ: Xây dựng mô hình nghiệp vụ cho bài toán**
- Khi khách lấy xe, người coi xe kiểm tra vé xem vé là thật hay giả, đổi chiểu vé với xe. Nếu vé giả hay không đúng xe thì không cho nhận xe. Ngược lại thì viết phiếu thanh toán và thu tiền của khách, đồng thời ghi các thông tin cần thiết vào sổ xe ra
- Khi khách đến báo cáo có sự cố thì kiểm tra xe trong sổ xe vào và sổ xe ra để xác minh xe có gửi hay không và đã lấy ra chưa. Nếu không đúng như vậy thì không giải quyết. Trong trường hợp ngược lại tiến hành kiểm tra xe ở hiện trường. Nếu đúng như sự việc xảy ra thì tiến hành lập biên bản giải quyết và trong trường hợp cần thiết thì viết phiếu chi bồi thường cho khách

Sơ đồ phân rã chức năng-FDD

Cách tiếp cận từ dưới lên

- Ví dụ: Mô hình phân rã chức năng quản lý trông gửi xe



Lập mô hình quy trình

- Có nhiều cách khác nhau để nắm bắt quy trình nghiệp vụ với các sơ đồ.
- Một dạng phổ biến của một mô hình quy trình là một sơ đồ luồng dữ liệu (Data Flow Diagram -DFD)
- Các công cụ khác đã được phát triển cho việc lập mô hình quy trình:
 - Sơ đồ ca sử dụng (use case diagram) và Sơ đồ hoạt động (activity diagram).
 - Hệ thống ký hiệu mô hình hóa tiến trình nghiệp vụ (Business Process Modeling Notation-BPMN)

Lập mô hình quy trình

- **Các sơ đồ luồng dữ liệu (Data-flow Diagrams-DFD)**
 - Minh họa đồ họa các quy trình nắm bắt, thao tác, lưu trữ và phân phối dữ liệu giữa một HT và môi trường của HT và giữa các thành phần của HT.
- **Lập mô hình quy trình của 1 HT**
 - Sử dụng thông tin được tập hợp khi xác định các yêu cầu
 - Lập mô hình các quy trình và các cấu trúc dữ liệu

Lập mô hình quy trình (tt)

• **Những sản phẩm và kết quả**

- Tập hợp các sơ đồ luồng dữ liệu mạch lạc và có liên quan với nhau
- Sơ đồ luồng dữ liệu ngữ cảnh (Context DFD)
 - Phạm vi của hệ thống
- DFDs của hệ thống hiện tại
 - Cho phép các nhà phân tích hiểu HT hiện tại
- DFDs của hệ thống logic mới
 - Công nghệ độc lập
 - Biểu diễn các luồng dữ liệu, cấu trúc và những yêu cầu chức năng của HT mới.
- Mô tả đầy đủ chi tiết mỗi thành phần DFD

Lập mô hình quy trình (tt)

- Cơ chế lập sơ đồ luồng dữ liệu
 - 4 ký hiệu được dùng
 - Dòng-luồng dữ liệu (Data Flow)
 - Kho dữ liệu (Data Store)
 - Tiết trình (Process)
 - Nguồn/đích (Source/Sink)

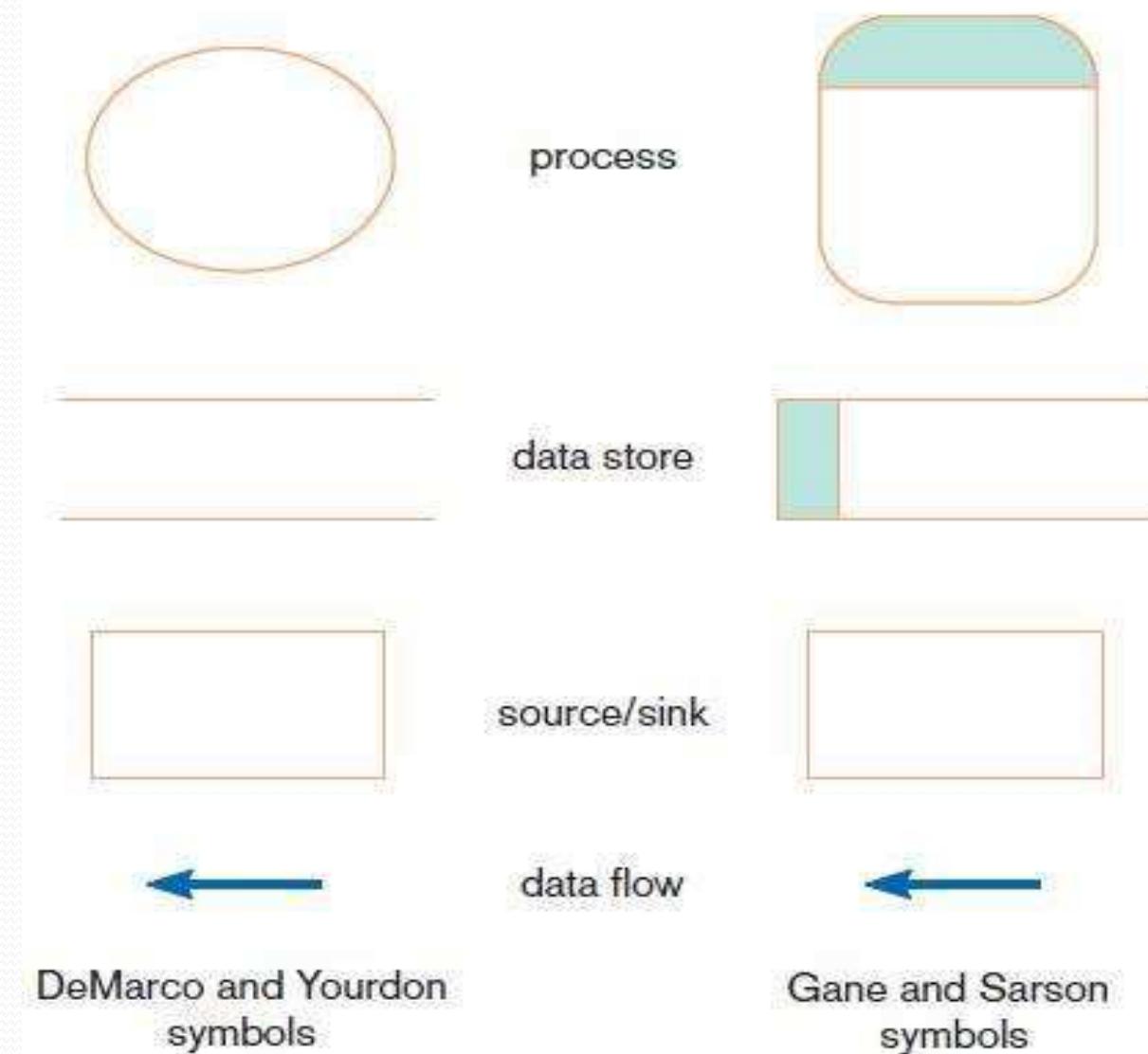


FIGURE 7-2
Comparison of DeMarco and Yourdon
and Gane and Sarson DFD symbol sets

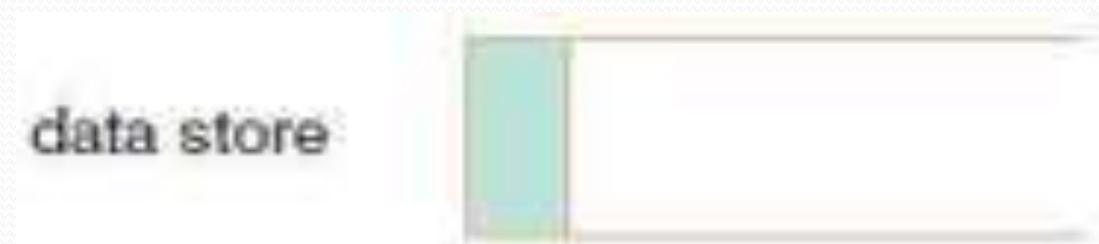
Cơ chế lập sơ đồ luồng dữ liệu

- Luồng dữ liệu (Data Flow)
 - Mô tả luồng dữ liệu
 - Chọn một tên ý nghĩa để đặt cho dữ liệu
 - Vd: biểu diễn dữ liệu trong 1 đơn đặt hàng của khách, các kết quả của 1 truy vấn đến 1 csdl, các nội dung của 1 báo cáo được in ấn,....



Cơ chế lập sơ đồ luồng dữ liệu (tt)

- Kho dữ liệu (Data Store)
 - Mô tả nơi chứa dữ liệu (bên trong HT)
 - Có thể biểu diễn dữ liệu ở
 - Thư mục tập tin
 - Tập tin dựa trên máy tính
 - Sổ sách
 - Nhãn gồm tên của kho cũng như con số
 - Vd: Một kho dữ liệu chứa dữ liệu về khách hàng, sinh viên, hóa đơn khách hàng hoặc các đơn hàng gửi nhà cung cấp



Cơ chế lập sơ đồ luồng dữ liệu (tt)

- Tiến trình (Process - còn gọi là Quy Trình, Xử lý)
 - Mô tả công việc hoặc các hành động được thực hiện trên dữ liệu khi chúng được chuyển đổi, lưu trữ hoặc phân phối (bên trong HT)
 - Số tiến trình cũng như tên được ghi lại



Cơ chế lập sơ đồ luồng dữ liệu (tt)

- Nguồn/Đích (Source/Sink) – Tác nhân ngoài (External Entity)
 - Mô tả nơi phát (nguồn) hoặc/và nơi nhận (đích) của dữ liệu (bên ngoài HT)
 - Thỉnh thoảng chỉ một thực thể bên ngoài (hoặc tác nhân ngoài)



Các định nghĩa sơ đồ luồng dữ liệu

- **Sơ đồ ngữ cảnh (Context Diagram)**

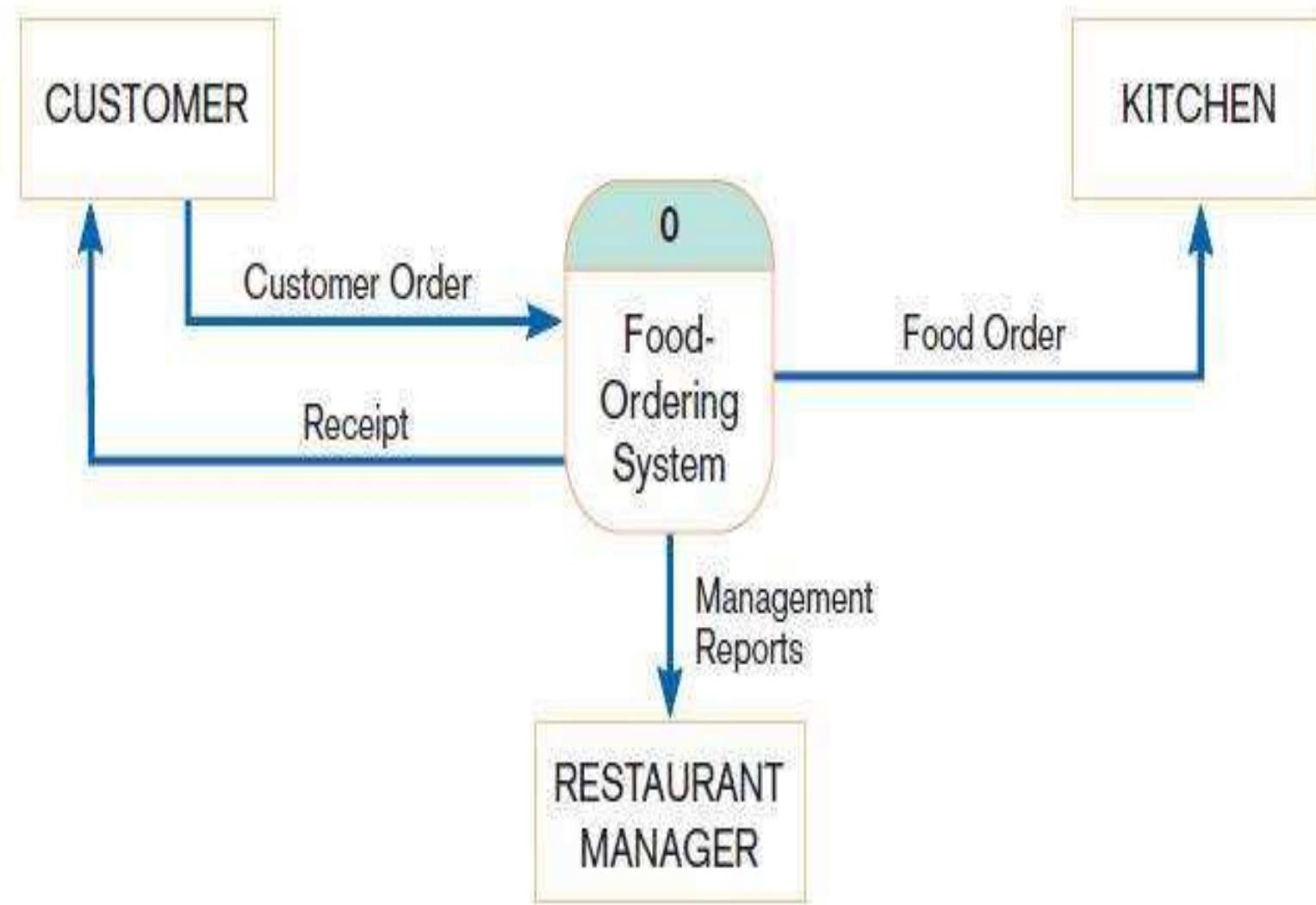
- Một sơ đồ luồng dữ liệu của phạm vi một hệ thống tổ chức biểu diễn các đường biên hệ thống, các thực thể bên ngoài tương tác với hệ thống và những luồng thông tin chính giữa các thực thể và hệ thống

- **Sơ đồ Mức-0 (Level-0 Diagram) hay mức định**

- Một sơ đồ luồng dữ liệu biểu diễn các tiến trình chính, các luồng dữ liệu, và các kho dữ liệu tại một mức cao hơn.
- Các tiến trình được gán nhãn 1.0, 2.0,..

Phát triển DFDs: Một ví dụ

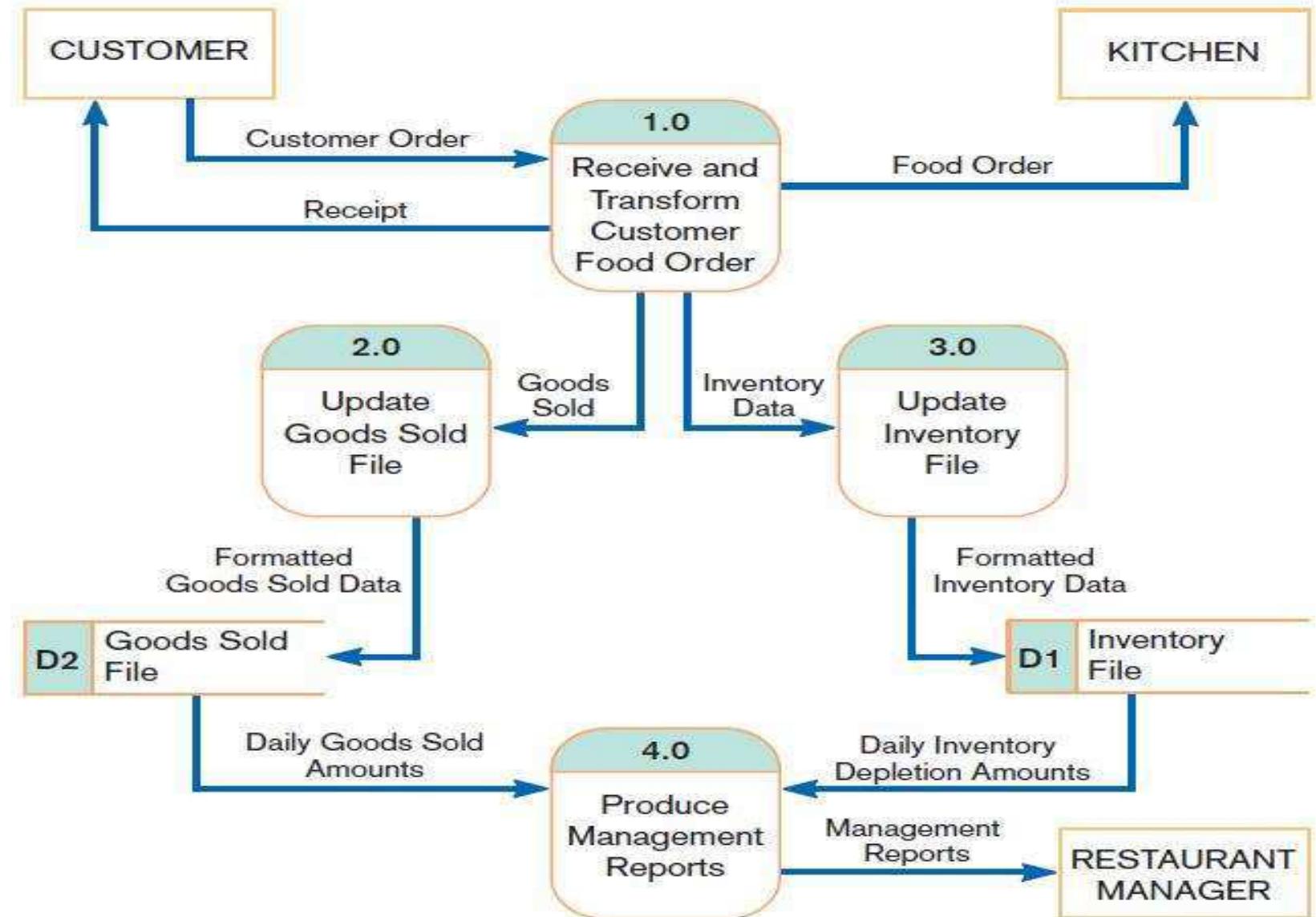
- *Sơ đồ ngữ cảnh* của Hệ thống đặt hàng Thức ăn tự động của Hoosier Burger
- Sơ đồ ngữ cảnh không chứa các kho dữ liệu



Phát triển DFDs: Một ví dụ (tt)

- Bước tiếp theo là mở rộng sơ đồ ngữ cảnh để biểu diễn sự phân tích các tiến trình

- Figure 7-5
Level-0 DFD of Hoosier Burger's food-ordering system



Các luật lập sơ đồ luồng dữ liệu

- Các luật cơ bản áp dụng cho tất cả DFDs:
 - Các đầu vào cho một tiến trình luôn luôn khác với các đầu ra
 - Các đối tượng luôn luôn có một tên duy nhất
 - Để giữ cho sơ đồ thông suốt, bạn có thể lặp lại các kho dữ liệu và các luồng dữ liệu trên một sơ đồ

Các luật lập sơ đồ luồng dữ liệu (tt)

- Tiết trình
 - A. Không có tiết trình chỉ có các đầu vào.
 - B. Không có tiết trình chỉ có các đầu ra (hỗn đen)
 - C. Một tiết trình là một nhãn cụm động từ
- Kho dữ liệu
 - D. Dữ liệu không thể được di chuyển từ 1 kho này đến 1 kho khác
 - E. Dữ liệu không thể di chuyển từ một nguồn bên ngoài đến 1 kho dữ liệu
 - F. Dữ liệu không thể di chuyển trực tiếp từ một kho dữ liệu đến một đích dữ liệu (data sink)
 - G. Kho dữ liệu là một nhãn cụm danh từ
- Nguồn/Đích
 - H. Dữ liệu không thể di chuyển trực tiếp từ 1 nguồn đến 1 đích
 - I. Một source/sink là một nhãn cụm danh từ

Các luật lập sơ đồ luồng dữ liệu (tt)

- Luồng dữ liệu
 - J. Một luồng dữ liệu chỉ có một hướng luồng giữa các ký hiệu
 - K. Một ký hiệu ngã ba (folk) nghĩa là chính xác cùng dữ liệu đi từ 1 vị trí chung đến 2 hoặc nhiều tiến trình, các kho dữ liệu, hoặc các nguồn/dịch
 - L. Một điểm nối (joint) nghĩa là chính xác cùng dữ liệu đến từ 2 hay nhiều tiến trình, kho dữ liệu hoặc nguồn/dịch khác nhau bất kỳ đến một vị trí chung
 - M. Một luồng dữ liệu không thể trực tiếp quay lại cùng tiến trình nó đã rời đi
 - N. Một luồng dữ liệu đến một kho dữ liệu nghĩa là cập nhật
 - O. Một luồng dữ liệu từ một kho dữ liệu nghĩa là khôi phục hoặc sử dụng
 - P. Một luồng dữ liệu là một nhãn cụm danh từ

Phân rã DFDs (Decomposition)

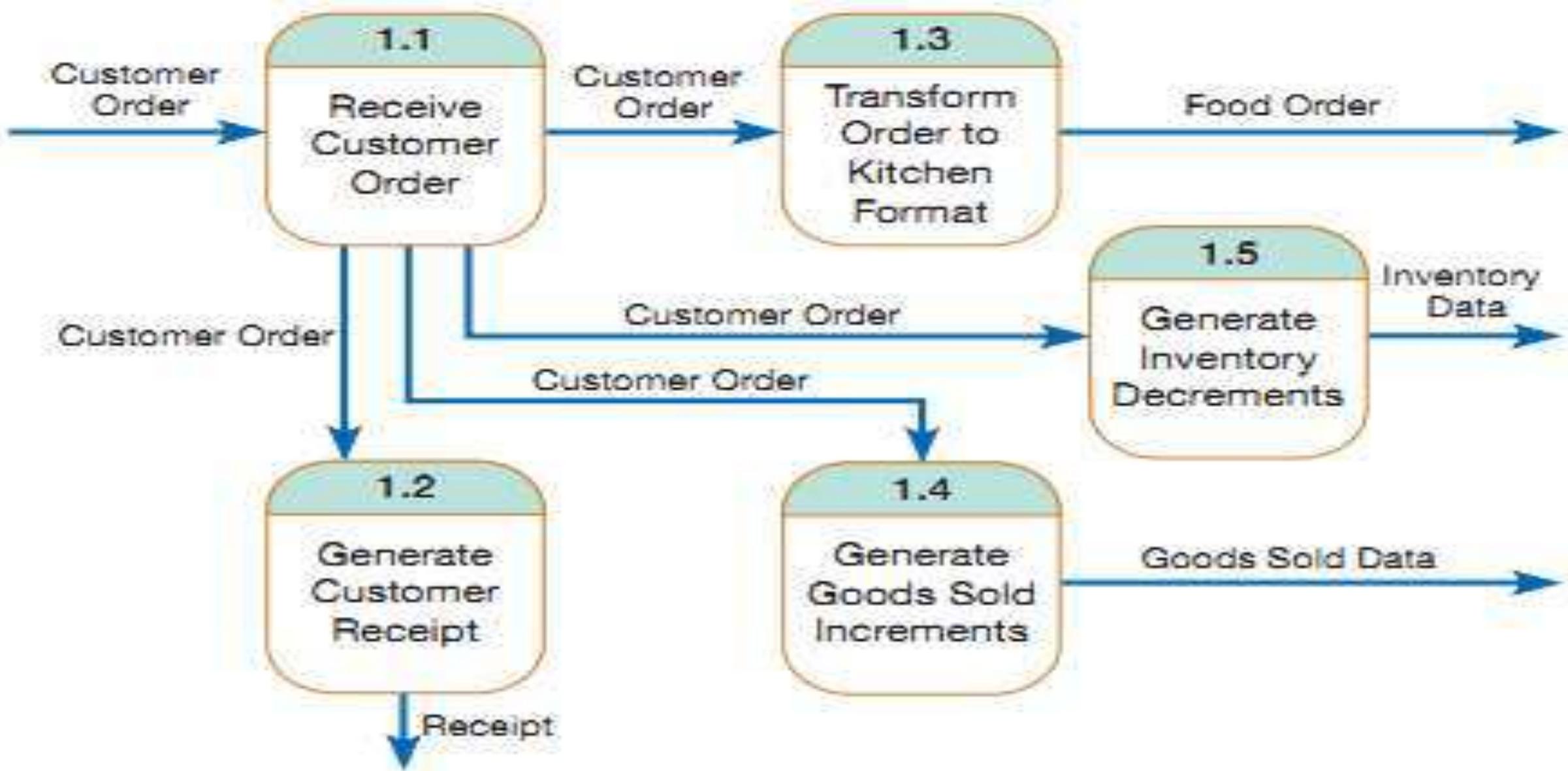
- **Phân rã chức năng**

- Từ một hệ thống đơn thành nhiều tiến trình thành phần
- Thủ tục lặp lại
- Mức thấp nhất được gọi là một DFD nguyên thủy

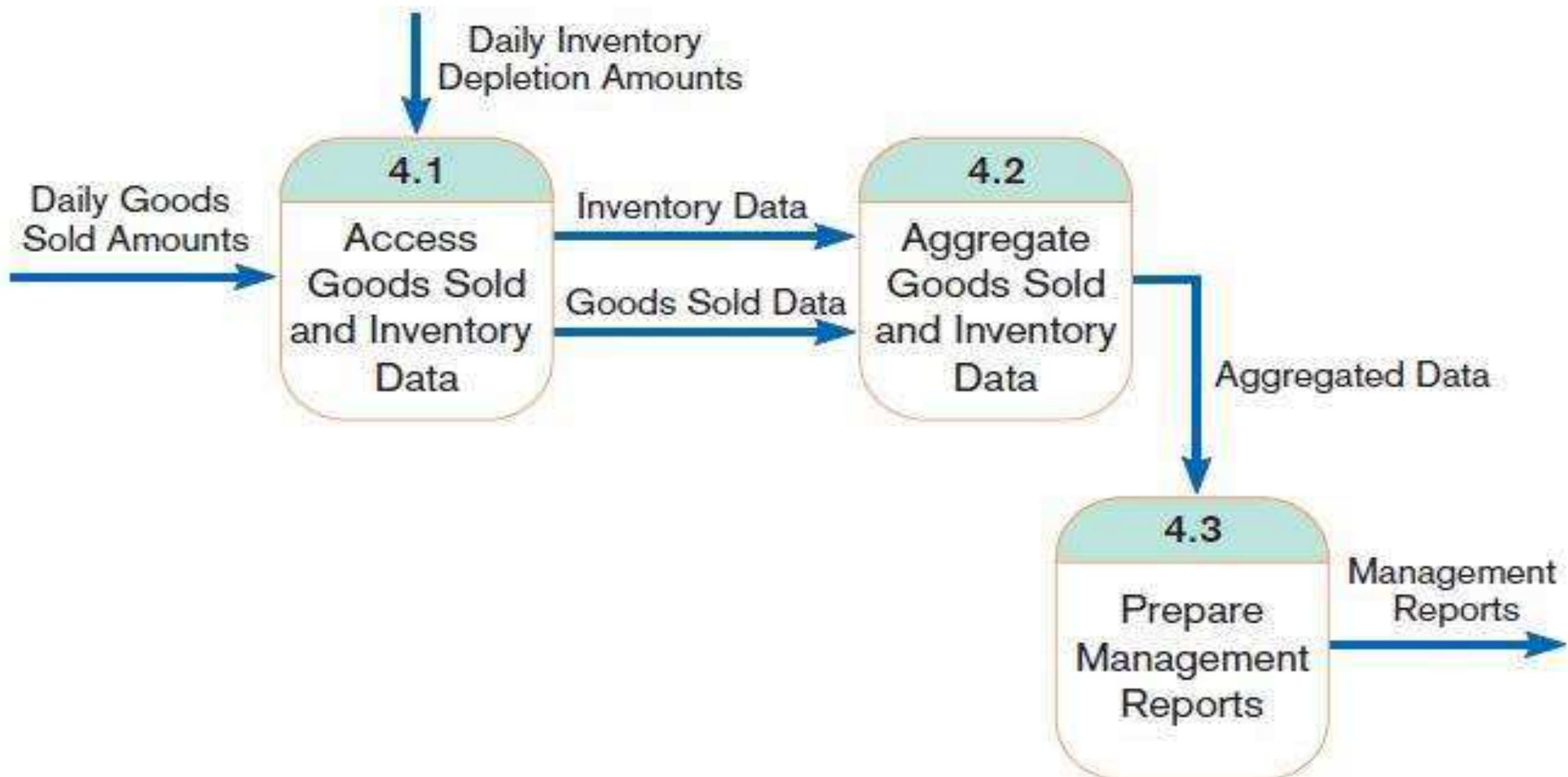
- **Các sơ đồ Mức-n**

- Một DFD là kết quả của n phân rã được lồng nhau của một loạt các tiến trình con từ một tiến trình trên một sơ đồ mức 0.

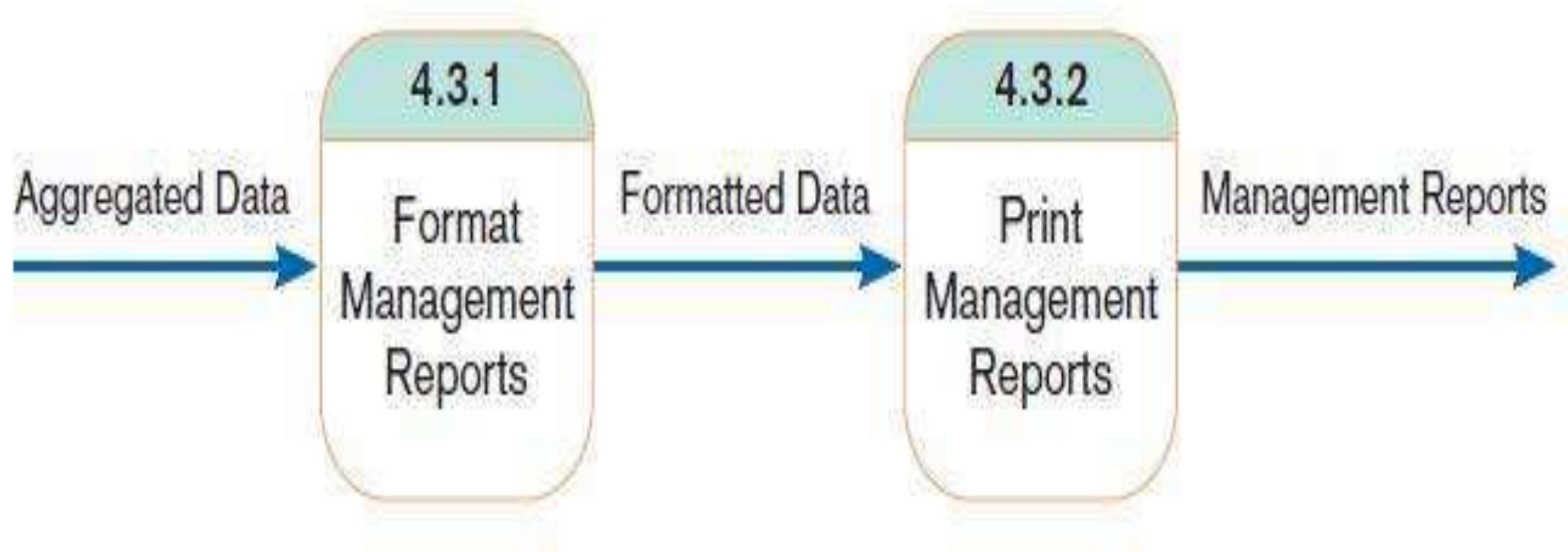
Phân rã DFDs (Decomposition) (tt)



Phân rã DFDs (Decomposition) (tt)



Phân rã DFDs (Decomposition) (tt)

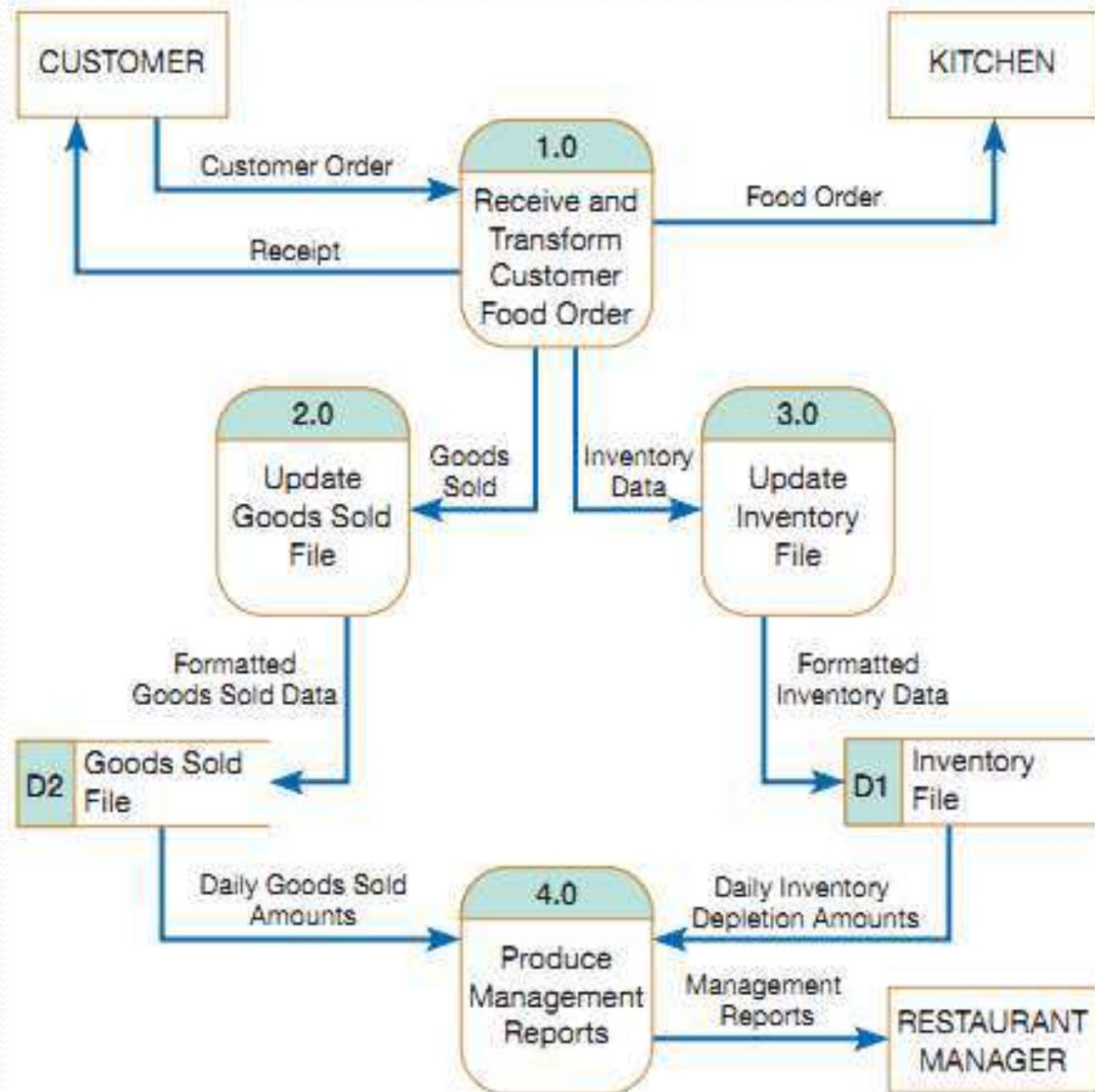


Cân bằng DFDs (tt)

Vd (tt):

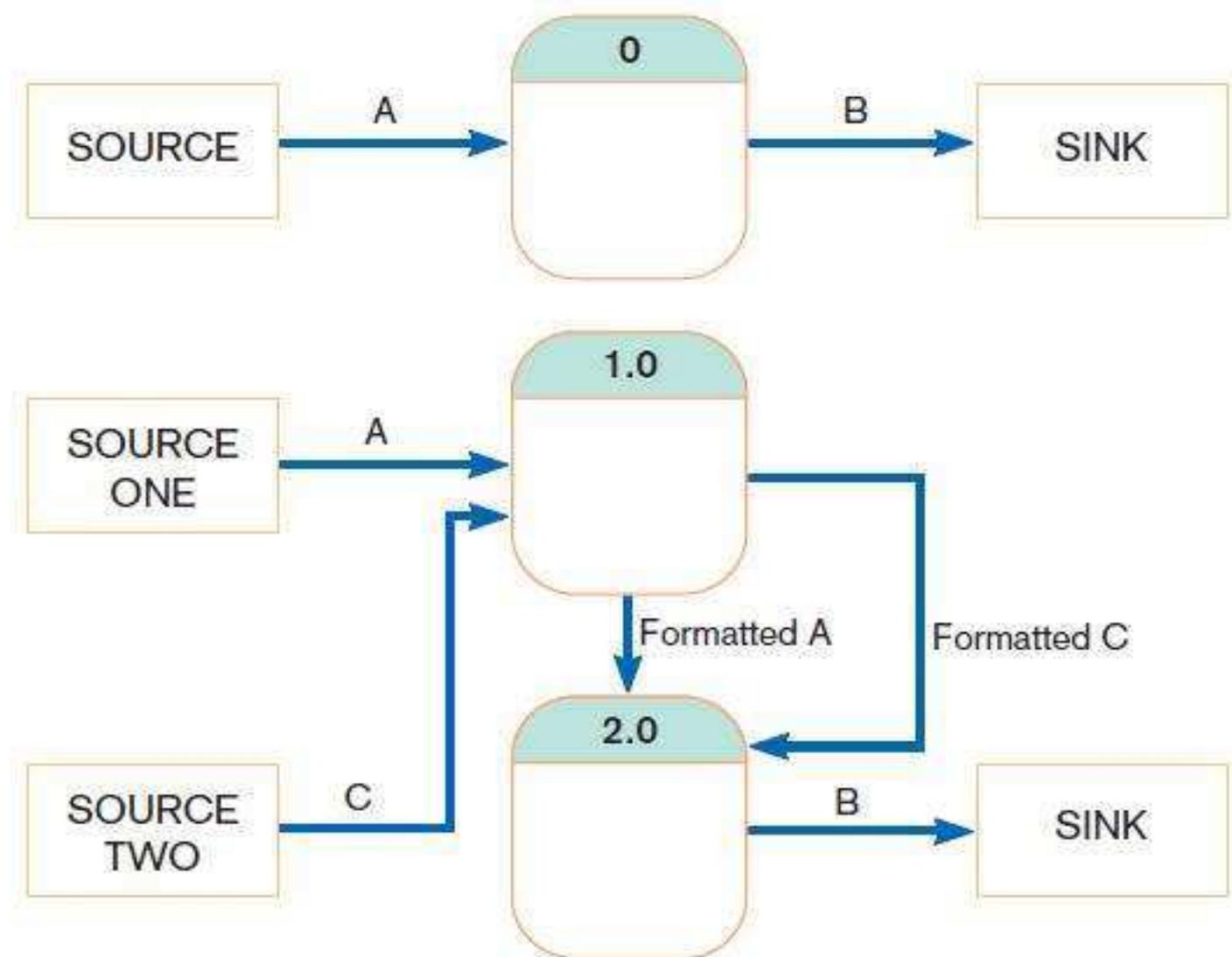
- Chú ý hình bên:

- Kiểm tra sự cân bằng sơ đồ ngũ cảnh và DFD mức 0



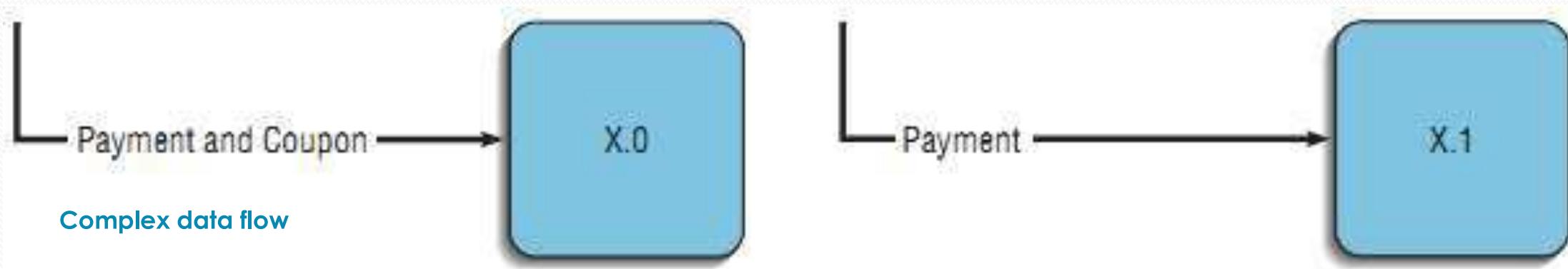
Cân bằng DFDs - 1 vd không cân bằng

- Trong sơ đồ ngữ cảnh, chúng ta có 1 đầu vào hệ thống là A và 1 đầu ra B
- Kiểm tra sự cân bằng các DFDs ??



Cân bằng DFDs

- Chúng ta có thể chia 1 luồng dữ liệu thành những luồng dữ liệu riêng rẽ trên 1 sơ đồ mức thấp hơn



A



B

Disaggregated data flows

Cân bằng DFDs-4 luật nâng cao

- Q. Một luồng dữ liệu ghép lại trên một mức có thể được tách thành các luồng dữ liệu thành phần ở mức tiếp theo, nhưng không có dữ liệu mới được thêm vào, và tất cả dữ liệu trong luồng ghép lại phải được coi như cho một hoặc nhiều luồng con.
- R. Đầu vào một quy trình phải đủ để tạo các đầu ra (gồm dữ liệu được đặt trong các kho dữ liệu) từ quy trình. Vì vậy tất cả đầu ra có thể được tạo, và tất cả dữ liệu trong đầu vào di chuyển hoặc đến một tiến trình khác hoặc đến một kho dữ liệu bên ngoài quy trình hoặc trên một DFD mức chi tiết hơn biểu diễn một sự phân rã của quy trình đó.

Cân bằng DFDs-4 luật nâng cao thêm (tt)

- S. Tại mức DFD thấp nhất, các luồng dữ liệu mới có thể được thêm vào để biểu diễn dữ liệu được chuyển theo các điều kiện ngoại lệ; những luồng dữ liệu này thường biểu diễn các thông điệp lỗi hoặc những thông báo xác nhận. Vd,
 - với khách hàng không biết, bạn có muốn tạo một khách hàng mới không?
 - “Bạn có muốn xóa bản ghi này không?”
- Để tránh các đường luồng dữ liệu chéo nhau, bạn có thể lặp lại kho dữ liệu hoặc các nguồn/đích trên một DFD. Sử dụng một ký hiệu bổ sung như một đường đôi vào giữa đường đọc đứng của ký hiệu kho dữ liệu, hoặc một đường chéo trong góc hình vuông của một nguồn/đích, để chỉ một ký hiệu lặp lại.

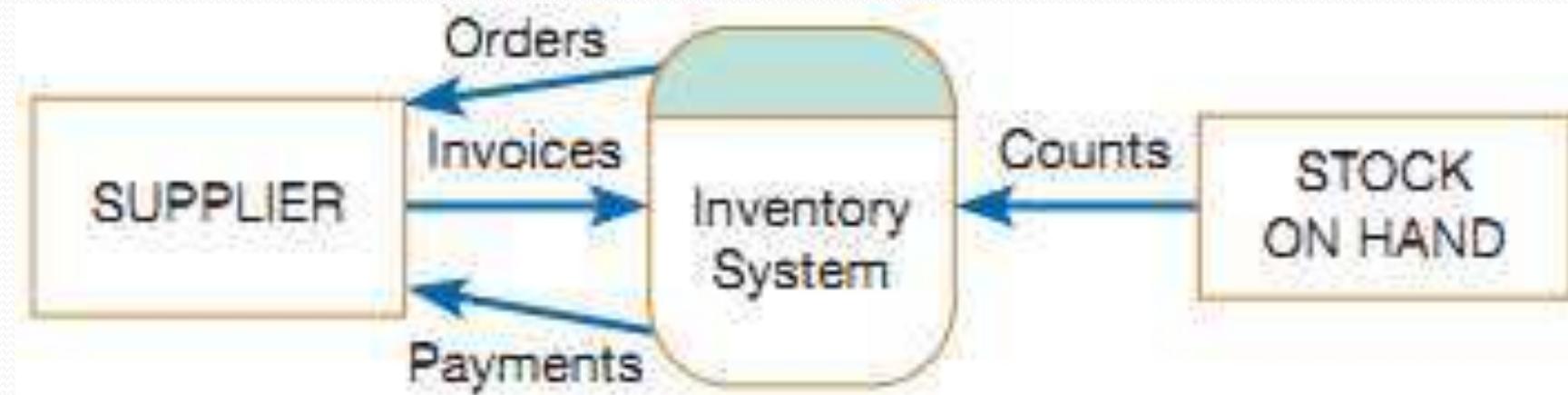
VD DFD

- Vd từ Hoosier Burger: HT đặt thức ăn phát sinh ra 2 loại dữ liệu – hàng hóa đã bán và kho hàng.
- Danh sách các bước liên quan HT kiểm soát kho hàng của Bob Mellankamp.

1. Thỏa mãn các xe chở sản phẩm đến người tiêu dùng (delivery trucks) trước khi mở cửa nhà hàng
2. Dỡ hàng (unload) và lưu trữ việc giao hàng
3. Ghi các hóa đơn (log invoices) và hồ sơ trong ứng tập tin.
4. Thêm bằng tay số lượng đã nhận vào kho
5. Sau khi đóng cửa, in báo cáo hàng tồn kho
6. Kiểm kê số lượng hàng tồn kho
7. So sánh tổng số báo cáo hàng tồn kho với tổng số kiểm kê kho.
8. So sánh tổng số kiểm kê kho với số lượng đặt hàng nhỏ nhất; nếu số lượng nhỏ hơn thì đặt hàng; nếu không thì không làm gì cả.
9. Thanh toán hóa đơn đến hạn và lưu trữ lại khi đã thanh toán xong.

VD DFD

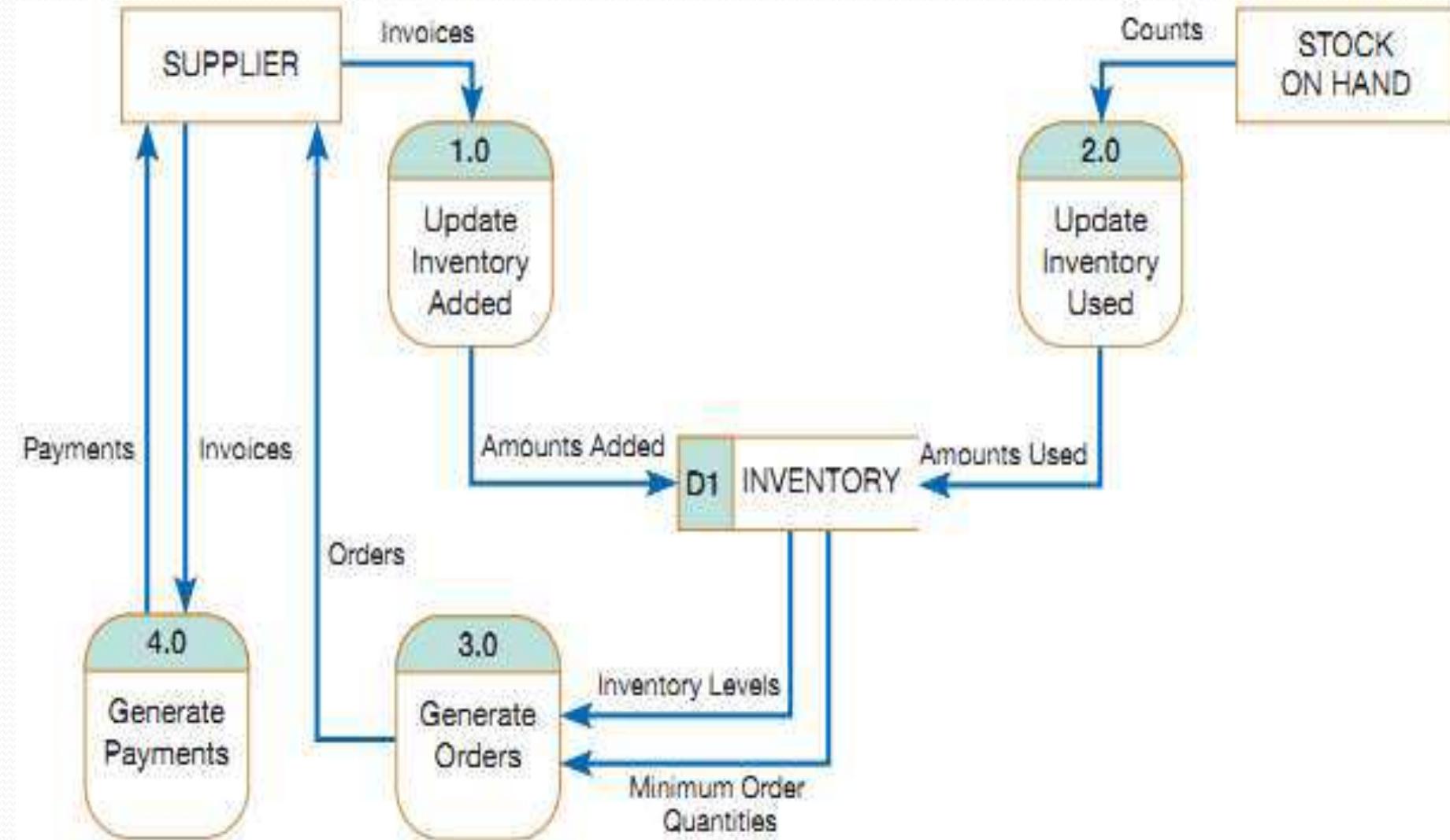
- Hoosier Burger



- Figure 7-14
 - (a) Context diagram for Hoosier Burger's inventory control system

VD DFD

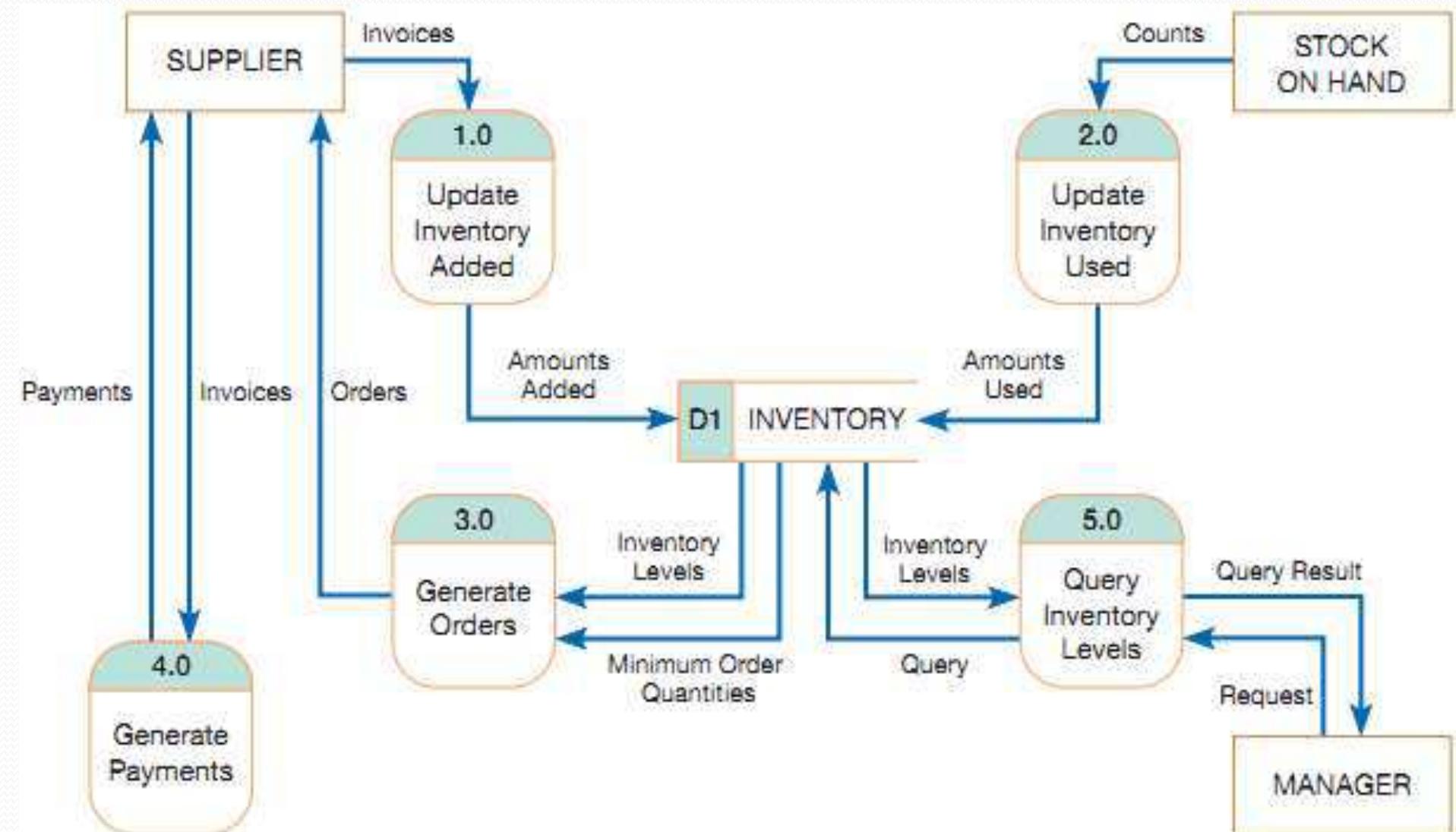
- Hoosier Burger



- Figure 7-14
(b) Level-0 DFD for Hoosier Burger's inventory control system

VD DFD

- Hoosier Burger



- Figure 7-15
Revised level-0 DFD for Hoosier
Burger's inventory control system

Nguyên tắc vẽ DFDs

- 1. Tính đầy đủ (Completeness)
 - DFD phải gồm tất cả các thành phần cần thiết cho hệ thống
 - Mỗi thành phần phải được mô tả đầy đủ trong từ điển dự án hoặc kho CASE
- 2. Tính thống nhất (Consistency)
 - Phần mở rộng đối với thông tin được chứa trên 1 mức của một tập các DFDs lồng nhau cũng được tính đến trên những mức khác.

Nguyên tắc vẽ DFDs

- 3. Lưu ý sự phối hợp thời gian (timing)
 - Thời gian không được biểu diễn rõ trên các DFDs
 - Tốt nhất để vẽ các DFDs nếu như hệ thống chưa bao giờ được bắt đầu và sẽ không bao giờ dừng.
- 4. Sự phát triển lặp lại (iterative development)
 - Nhà phân tích mong muốn việc vẽ lại các DFD vài lần trước khi có được bản gần giống nhất so với hệ thống đang được mô hình hóa

Nguyên tắc vẽ DFDs

- 5. Việc vẽ các DFD nguyên thủy
 - Mức phân rã logic thấp nhất
 - Quyết định phải được thực hiện khi dùng phân rã
- **Các luật dùng phân rã:**
 - Khi mỗi tiến trình được giảm nhỏ thành 1 quyết định, tính toán hoặc thao tác csdl đơn
 - Khi mỗi kho dữ liệu biểu diễn dữ liệu về một thực thể đơn
 - Khi người dùng hệ thống không quan tâm xem xét mức chi tiết hơn nữa...

Nguyên tắc vẽ DFDs

- **Các luật dùng phân rã (tt):**

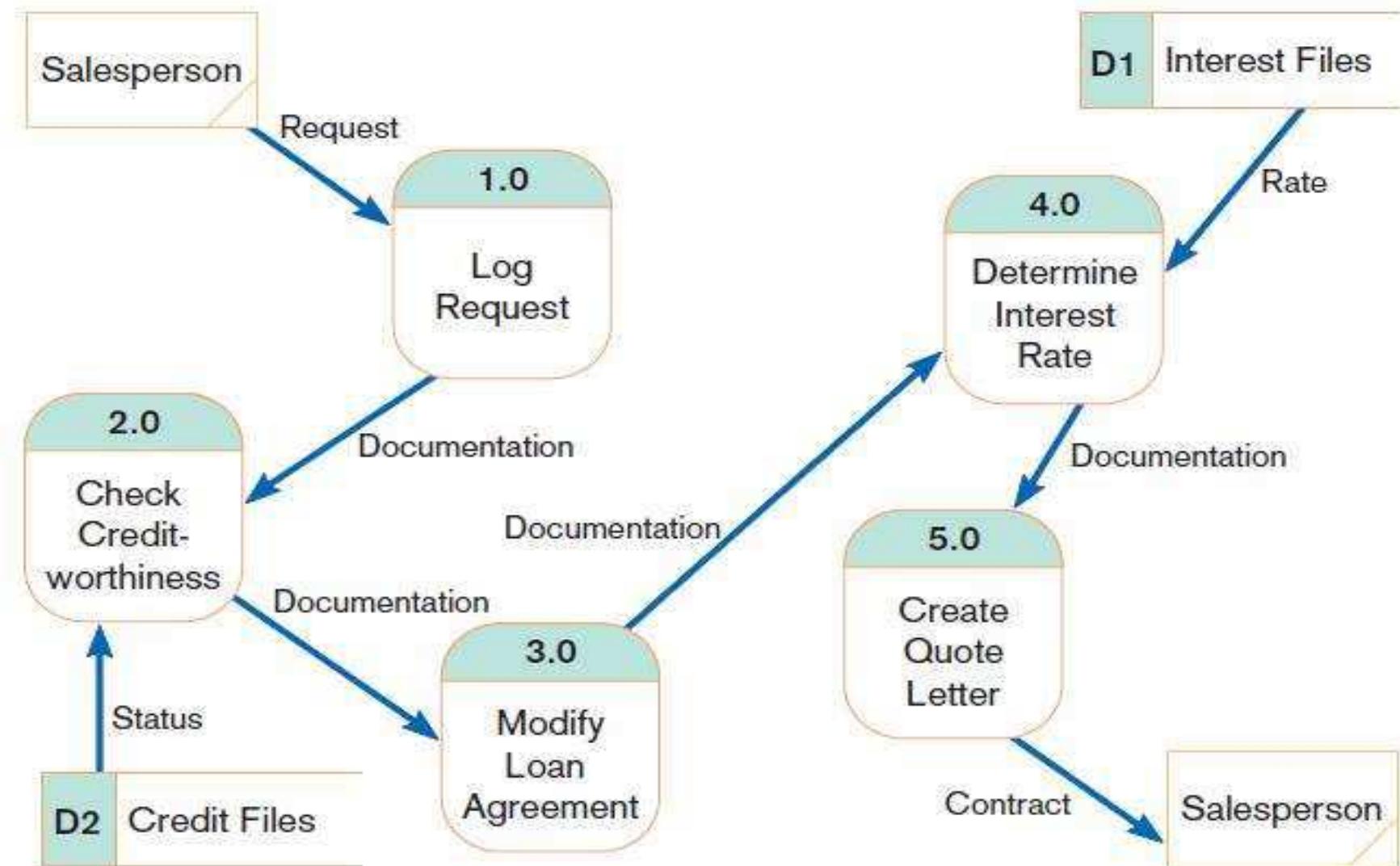
- Khi mỗi luồng dữ liệu không cần được chia nhỏ hơn để biểu diễn rằng dữ liệu được xử lý theo những cách khác
- Khi đã biểu diễn mỗi biểu mẫu hoặc giao tác nghiệp vụ, màn hình trực tuyến và báo cáo như một luồng dữ liệu đơn
- Khi có một tiến trình tách rời được biểu diễn cho mỗi lựa chọn trên tất cả tùy chọn menu mức-thấp nhất.

Sử dụng DFDs như các Công cụ Phân tích

- Phân tích lỗ hổng (Gap Analysis)
 - Tiến trình khám phá sự không thống nhất giữa 2 hay nhiều tập hợp sơ đồ luồng dữ liệu hoặc sự không thống nhất trong một DFD đơn
 - Sự không hiệu quả trong 1 HT có thể thường được nhận ra qua các DFDs

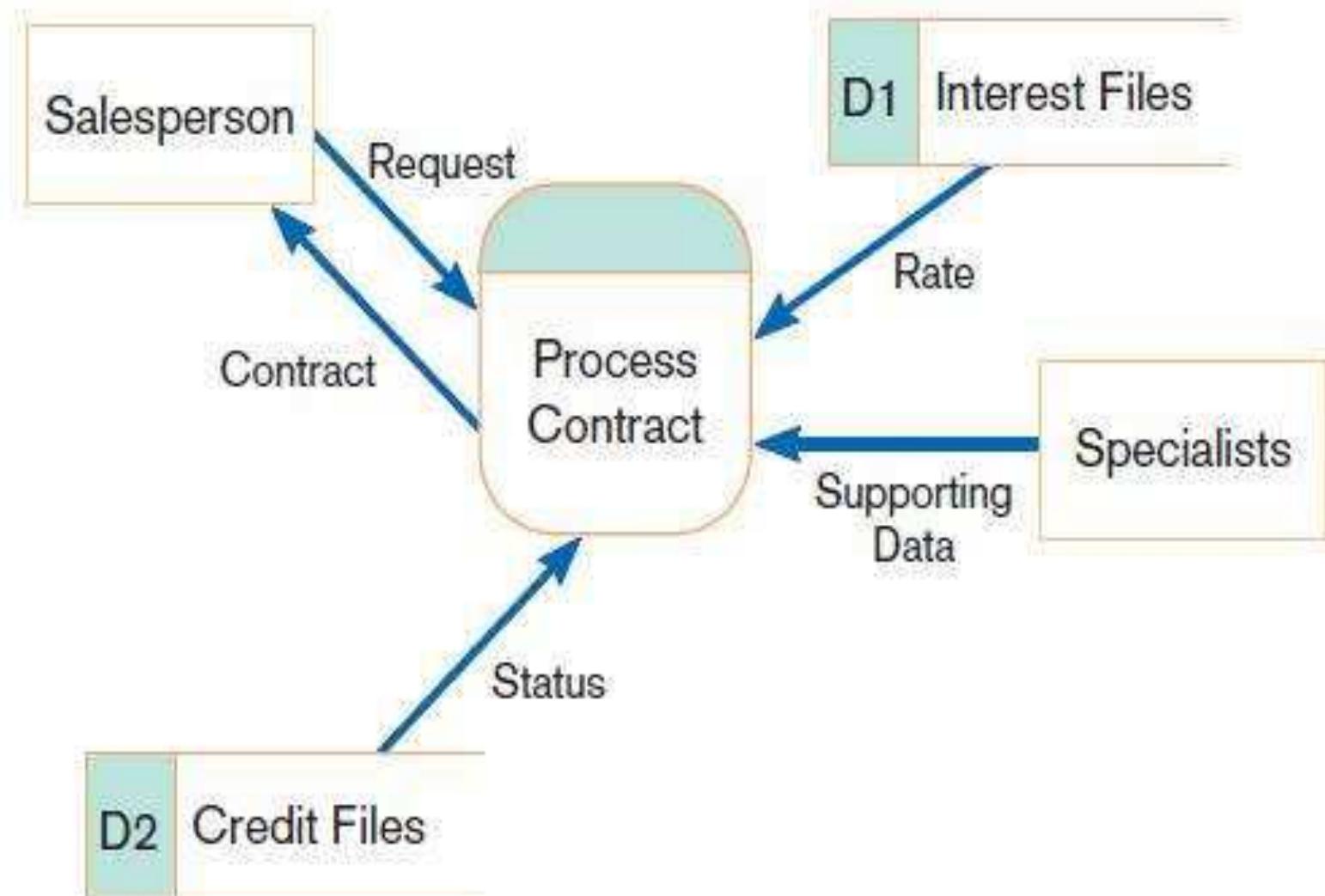
Sử dụng DFDs trong Tái cơ cấu Quy trình Nghiệp vụ

- Vd: IBM Credit
- Tiến trình
Credit approval
đòi hỏi 6 ngày
trước khi Tái cơ
cấu quy trình
nghiệp vụ.



Sử dụng DFDs trong Tái cơ cấu Quy trình Nghiệp vụ

- Sau khi Tái cơ cấu Quy trình nghiệp vụ, IBM đã có thể xử lý 100 lần số giao dịch trong cùng lượng thời gian



Lập mô hình Logic (còn gọi là Đặc tả quy trình)

- Nhằm biểu diễn nội dung luận lý (logic) của các quy trình. Vì các sơ đồ luồng dữ liệu không biểu diễn logic bên trong các quy trình
- Mô hình hóa logic liên quan việc biểu diễn cấu trúc và chức năng bên trong của các quy trình được mô tả trên 1 DFD
- Các hình thức đặc tả
 - Mã giả (Pseudo code)
 - Lưu đồ thuật giải (Flow charts)
 - Bảng Quyết định (Decision tables)
 - Cây quyết định (Decision tree)

Mô hình logic với Bảng Quyết định

- Một ma trận biểu diễn trình tự logic của một quyết định
- Đặc tả những điều kiện có thể và những hành động kết quả
- Được sử dụng tốt nhất cho quyết định logic phức tạp
- Gồm 3 phần:
 - Các gốc điều kiện (condition stubs)
 - Liệt kê điều kiện liên quan đến quyết định
 - Các gốc hành động (action stubs)
 - Những hành động cho một tập điều kiện được đưa ra
 - Các luật (rules)
 - Đặc tả những hành động nào (giá trị hành động) được theo cho một tập những điều kiện được đưa ra (giá trị điều kiện)

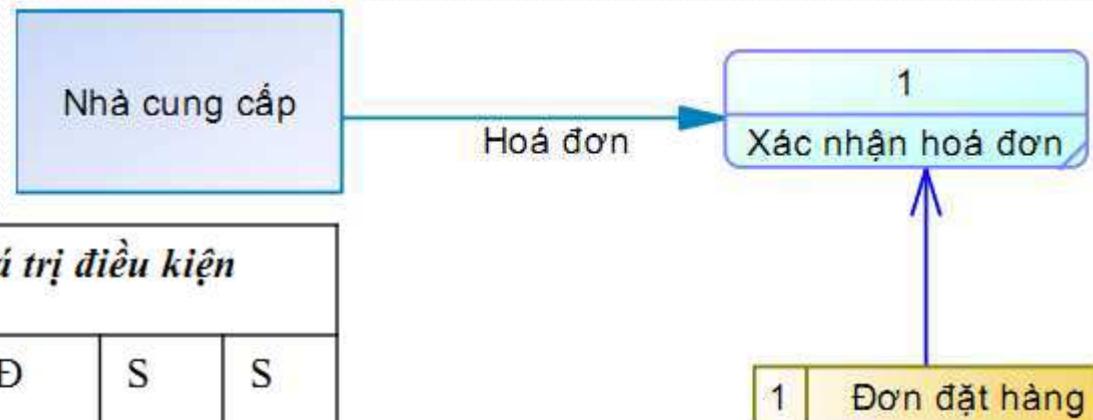
Mô hình logic với Bảng Quyết định (tt)

- Những điều kiện không quan trọng
- Thủ tục tiêu chuẩn cho việc tạo các bảng quyết định
 - Tên các điều kiện và giá trị mỗi điều kiện có thể có
 - Tên tất cả các hành động có thể xuất hiện
 - Danh sách tất cả các luật có thể
 - Định nghĩa các hành động cho mỗi luật (H1)
 - Đơn giản hóa bảng quyết định (H2)

Ví dụ: Bảng quyết định

- Bảng quyết định

Một phần mô hình tiến trình



<i>Điều kiện</i>	<i>Giá trị điều kiện</i>			
Có đơn đặt hàng tương ứng với hóa đơn không?	Đ	Đ	S	S
Hóa đơn có đúng số lượng và đơn giá không?	Đ	S	Đ	S
<i>Hành động</i>	<i>Giá trị hành động</i>			
Từ chối hóa đơn do không có đơn đặt hàng			X	X
Từ chối hóa đơn do không đúng số lượng và đơn giá		X		X
Thanh toán hóa đơn	X			

Mô hình logic với cây quyết định

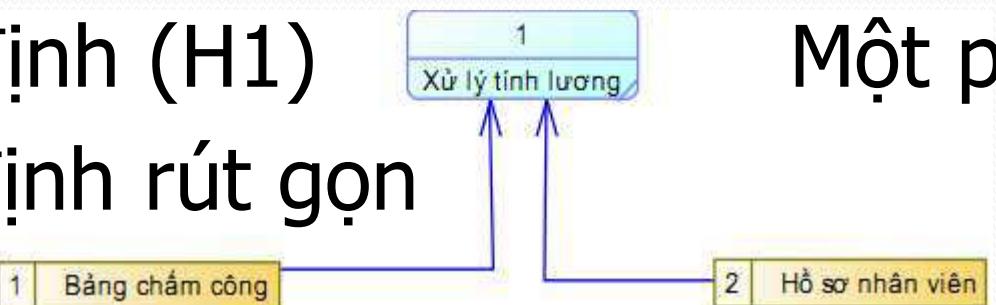
- ❖ Cây quyết định - Ví dụ: *Xử lý xác nhận hóa đơn*



Mô hình logic với Bảng Quyết định (tt) –H1-H2

- Bảng quyết định (H1)

Bảng quyết định rút gọn



Điều kiện	Giá trị điều kiện					
	1	2	3	4	5	6
Loại nhân viên	C	H	C	H	C	H
Số giờ làm việc	<40	<40	40	40	>40	>40
Hành động		Giá trị hành động				
Trả lương tháng	X		X		X	
Tính lương giờ		X		X		X
Tính lương ngoài giờ						X
Phát sinh báo cáo ngày nghỉ		X				

Một phần mô hình tiến trình

Điều kiện	Giá trị điều kiện			
Hành động	Giá trị hành động			
	1	2	3	4
	C	H	H	H

Điều kiện	Giá trị điều kiện			
Hành động	Giá trị hành động			
Loại nhân viên	1	2	3	4
Số giờ làm việc	-	<40	40	>40
Hành động		Giá trị hành động		
Trả lương tháng	X			
Tính lương giờ		X	X	X
Tính lương ngoài giờ				X
Phát sinh báo cáo ngày nghỉ		X		

Mô hình logic với Bảng Quyết định (tt) –H1-H2

Conditions/ Courses of Action	Rules											
	1	2	3	4	5	6	7	8	9	10	11	12
Type of item	P	N	P	N	P	N	P	N	P	N	P	N
Time of week	D	D	W	W	D	D	W	W	D	D	W	W
Season of year	A	A	A	A	S	S	S	S	H	H	H	H
Standing daily order	X			X				X				
Standing weekend order			X			X				X		
Minimum order quantity	X		X		X		X		X		X	
Holiday reduction							X		X			
Summer reduction				X		X						

Điều kiện	Giá trị điều kiện
Hành động	Giá trị hành động

Conditions/ Courses of Action	Rules						
	1	2	3	4	5	6	7
Type of item	P	P	P	P	P	P	N
Time of week	D	W	D	W	D	W	-
Season of year	A	A	S	S	H	H	-
Standing daily order	X		X		X		
Standing weekend order		X		X		X	
Minimum order quantity							X
Holiday reduction						X	X
Summer reduction			X	X			

Mô hình hóa Quy trình cho Ứng dụng thương mại điện tử

Mô hình quy trình cho WebStore của PVF

- Xem xét kết quả của các phiên làm việc JAD, tập trung vào định nghĩa cấu trúc HT WebStore
- **Xác định 6 tiền trình làm nền tảng cho mô hình DFD mức 0**
- Hệ thống WebStore nên trao đổi thông tin với các HT có sẵn thay vì lưu trữ thông tin dư thừa:
 - Purchasing Fullfillment System (HT theo dõi các đơn đặt hàng),
 - Customer Tracking System (HT theo dõi KH)

Bảng 6-4. Cấu trúc Hệ thống của WebStore và Các Quy trình Mức 0 tương ứng

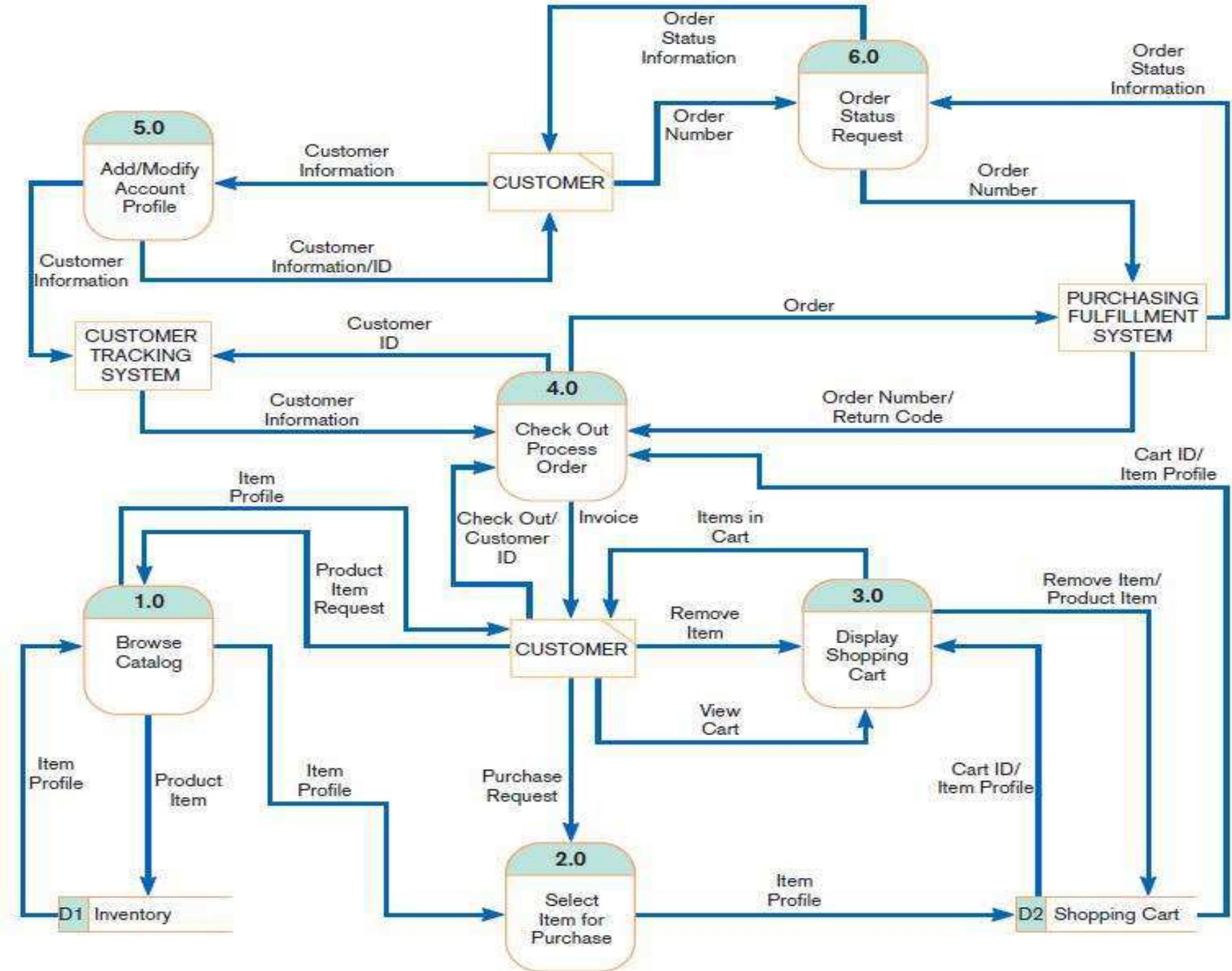
Hệ thống WebStore	Các quy trình
Trang chính	Màn hình thông tin (phụ / không có quy trình)
Dòng sản phẩm (Catalog)	
• Bàn làm việc	1.0 Duyệt Danh mục hàng
• Ghế	2.0 Chọn hàng để mua
• Bàn ăn	3.0 Hiển thị Giỏ hàng
• Tủ hồ sơ	4.0 Thanh toán / Xử lý đơn đặt hàng
Giỏ hàng	5.0 Thêm / Sửa đổi Hồ sơ Tài khoản
Kiểm tra	6.0 Yêu cầu Tình trạng Đơn đặt hàng
Hồ sơ tài khoản	Màn hình thông tin (phụ / không có quy trình)
Trạng thái đơn đặt hàng / lịch sử	
Bình luận của khách hàng	
Thông tin công ty	
Phản hồi	
Thông tin liên lạc	

Mô hình quy trình cho WebStore của PVF

- Khi khách hàng mở tài khoản, thông tin sẽ được truyền từ WebStore sang Customer Tracking System.
- Khi một đơn hàng được đặt hay yêu cầu thông tin của đơn hàng trước đó, thông tin sẽ được lưu vào hay lấy ra từ Purchasing Fulfillment System.
- Hệ thống cần truy xuất thông tin từ 2 nguồn:
 - Inventory (để tạo ra catalog sản phẩm online),
 - Shopping Cart (lưu trữ mặt hàng khách muốn mua) → thì một CSDL tạm thời cần được tạo ra. Một khi giao tác được hoàn thành, thì dữ liệu giỏ hàng có thể được xóa

Mô hình quy trình cho WebStore của PVF

- Mô hình hóa quy trình cho các dự án TMĐT không khác với những dự án khác
- Xem ví dụ Pine Valley Furniture

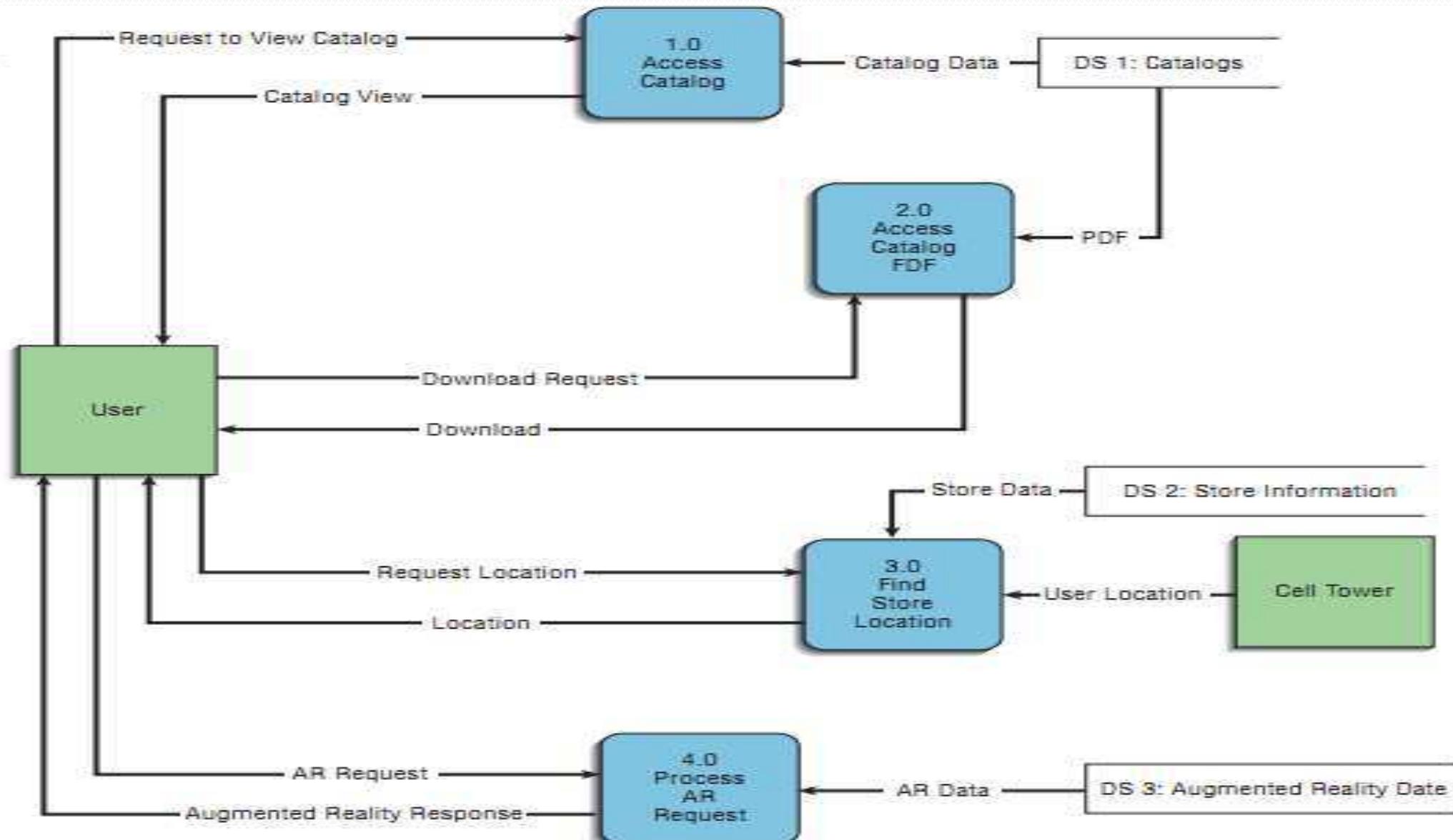


Mô hình quy trình cho WebStore của PVF

DFD cho ứng dụng điện thoại thông minh WebStore

FIGURE 6-20

Level-0 DFD for the
WebStore smartphone app.



3.2. Phân tích và thiết kế hướng đối tượng

- Biểu đồ

- **Use case**: nắm bắt các yêu cầu
- **Sequence**: Thể hiện thời gian tương tác giữa các đối tượng
- **Collaboration**: Làm nổi bật các quan hệ giữa các đối tượng và liên kết giữa chúng
- **Class**: thể hiện cấu trúc tinh của HT
- **State**: các trạng thái của đối tượng
- **Activity**: các hoạt động song song và các tiến trình kinh doanh
- **Component**: cấu trúc vật lí của các thành phần chính
- **Deployment**: phân bố vật lí của HT

3.2.A Ca sử dụng (Use Cases)

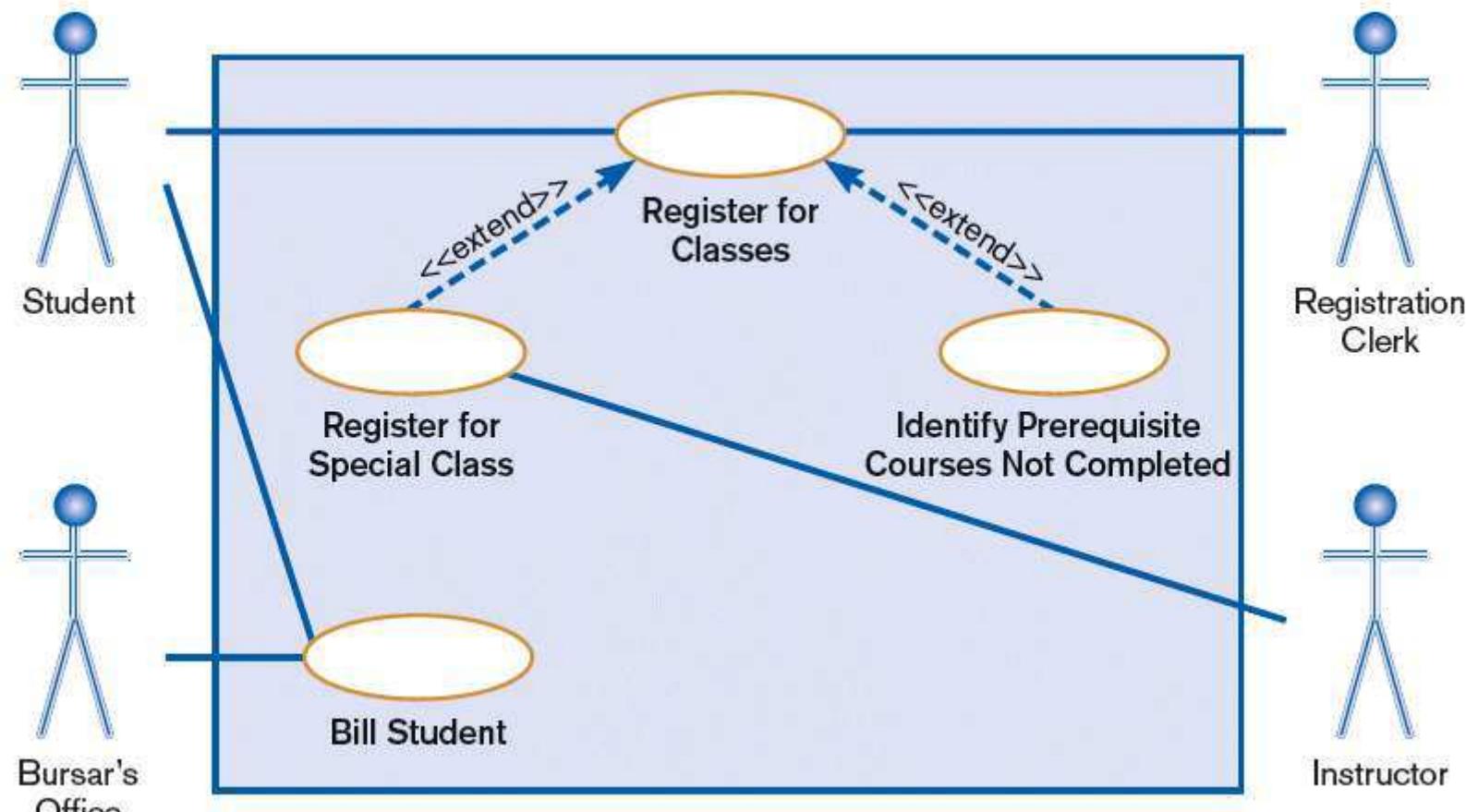
- **Mục tiêu**

- Giải thích các ca/tình huống sử dụng và các biểu đồ ca sử dụng và làm thế nào chúng có thể được dùng để lập mô hình chức năng hệ thống.
- Trình bày các hướng cơ bản làm thế nào để tạo các ca sử dụng được viết lại (written use cases).
- Thảo luận lập mô hình quy trình với các ca sử dụng cho ứng dụng TMĐT.

Ca/Tình huống sử dụng (Use Cases)

- Một **use case** là một mô tả về chức năng hoặc cách hoạt động cụ thể (hành vi) của một hệ thống theo những điều kiện khác nhau khi hệ thống phản hồi các yêu cầu từ những người dùng.
- Một **tác nhân (actor)** là một thực thể ngoài (external entity) tương tác với hệ thống.

Ca/Tình huống sử dụng (Use Cases) (tt)



HÌNH 7-26

Một biểu đồ use case của một hệ thống đăng ký học phần tại trường đại học

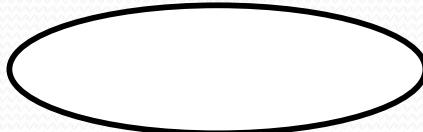
Ca/Tình huống sử dụng (Use Cases) (tt)

- Hầu hết các tác nhân biểu diễn các vai trò người dùng, nhưng các tác nhân cũng có thể là các hệ thống bên ngoài.
- Một tác nhân (actor) là một vai trò, không phải một người dùng cụ thể; một người dùng có thể đóng nhiều vai trò, và một tác nhân có thể đại diện nhiều người dùng.
- Một mô hình use case gồm các tác nhân và các ca sử dụng.

Các biểu đồ ca/tình huống sử dụng (Use Cases diagrams)

- **Biểu đồ ca sử dụng (Use case diagram):** một hình ảnh biểu diễn hành vi hệ thống cùng với các tác nhân chính tương tác với hệ thống.
 - Mô hình UC được mô tả bởi một hay nhiều biểu đồ UC
 - Là công cụ mạnh giúp thu thập yêu cầu chức năng HT
 - UCD chỉ ra quan hệ giữa UC và tác nhân và giữa UC với nhau
- **Ca sử dụng trừu tượng (Abstract use case)** là khi một use case được bắt đầu bởi một use case khác.
- Một use case mô tả chức năng hoàn chỉnh.

Định nghĩa và Kí hiệu



<<include>>

Ca sử dụng (Use Case)

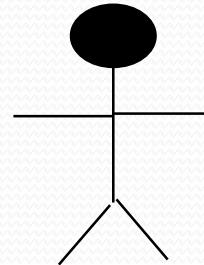
Tác nhân (Actor)

Đường biên (Boundary)

Kết nối (Connection)

Mỗi quan hệ bao gồm
(Include relationship)

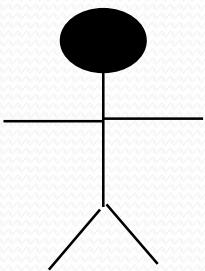
Mỗi quan hệ mở rộng
(Extend relationship)



<<extend>>

Định nghĩa và Kí hiệu (tt)

- **Tác nhân (Actor)** là một vai trò, không phải một cá nhân.
 - Liên quan với chức năng của hệ thống ở mức cơ bản nào đó.
 - Được biểu diễn bằng những hình que (stick figures)
- **Tác nhân (Actor)** có thể là
 - Nhóm người dùng
 - Phần cứng ngoại vi
 - Phần mềm khác



Định nghĩa và Kí hiệu (tt)

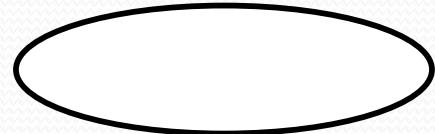
- Nhận diện **tác nhân (Actor)**: Trả lời một số câu hỏi
 - Ai là người sử dụng chức năng chính của HT?
 - Ai cần sự hỗ trợ từ HT để thực hiện công việc thường nhật của họ?
 - Ai phải thực hiện công việc bảo dưỡng, quản trị và giữ cho HT hoạt động?
 - HT sẽ kiểm soát thiết bị phần cứng nào?
 - HT đang xây dựng cần tương tác với những HT khác hay không ?
 - Ai hoặc vật thể nào quan tâm đến hay chịu ảnh hưởng bởi kết quả mà HT phần mềm tạo ra?

Định nghĩa và Kí hiệu (tt)

- **Một mô tả tóm tắt của một use case**

- Trả lời 3 câu hỏi
 - Ai? → một nhóm người dùng
 - Làm gì? → hành vi nào đó từ HT
 - Và tại sao? → có được giá trị nào đó

Định nghĩa và Kí hiệu (tt)

- **Ca sử dụng (Use case)** biểu diễn một chức năng HT đơn.
 - Được biểu diễn bằng một hình elip. 
- Trong thực hành, use case được cài đặt thành một vài loại chương trình máy tính và các actor sẽ dùng những chương trình này bằng cách nhập thông tin vào, nhận thông tin ra.
- Một actor có thể được kết hợp với một hoặc nhiều use case, và một use case có thể được kết hợp với một hoặc nhiều actor

Định nghĩa và Kí hiệu (tt)

- Tìm kiếm **Ca sử dụng (Use case)**: trả lời một số câu hỏi
 - Actor yêu cầu chức năng gì của HT?
 - Actor cần phải đọc, tạo, xoá, sửa đổi hoặc lưu trữ thông tin nào đó của HT không?
 - Actor cần thiết phải được cảnh báo về những sự kiện trong HT, hay actor cần phải báo hiệu cho HT về vấn đề nào đó không?
 - HT có thể hỗ trợ một số công việc thường nhật của actor nào đó hay không?
- Một số câu hỏi khác cần chú ý:
 - HT cần dữ liệu input/output nào? Dữ liệu đó đến từ đâu?
 - Những khó khăn nào liên quan đến hiện thực của HT hiện tại (vd HT quản lý bằng giấy tờ nên được thay thế bằng HT quản lý trên máy tính)?

Định nghĩa và Kí hiệu (tt)

- **Đường biên hệ thống (System boundary)** chứa tất cả các ca sử dụng có liên quan.
 - Một đường biên là đường chia cắt giữa hệ thống và môi trường của hệ thống.
 - Các use case nằm trong đường biên.
 - Các tác nhân nằm ngoài đường biên.
 - Được biểu diễn bằng một hình hộp.

Định nghĩa và Kí hiệu (tt)

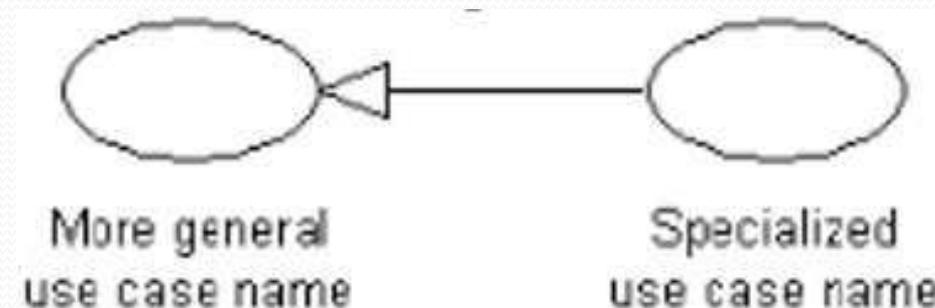
- Các mối kết hợp (association) và quan hệ (relationship) trong 1 biểu đồ use case
 - Quan hệ kết nối (Connection) hoặc mối kết hợp giao tiếp (communicates-association) giữa actor và use case (*hoặc quan hệ kết hợp - Association*)
 - Kết hợp generalization giữa các actor
 - Kết hợp generalization (tổng quát hóa) giữa các use case
 - Quan hệ extend (mở rộng) giữa các use case
 - Quan hệ include (bao gồm) giữa các use case

Định nghĩa và Kí hiệu (tt)

- **Kết nối (Connection)** là một mối kết hợp giữa một tác nhân và một ca sử dụng (use case).
 - Biểu diễn một mối quan hệ thông thường.
 - Kết nối không biểu diễn luồng dữ liệu
 - Các tác nhân được kết nối với các ca sử dụng qua các đường nối. _____
 - Các Use case được kết nối với nhau qua các mũi tên.

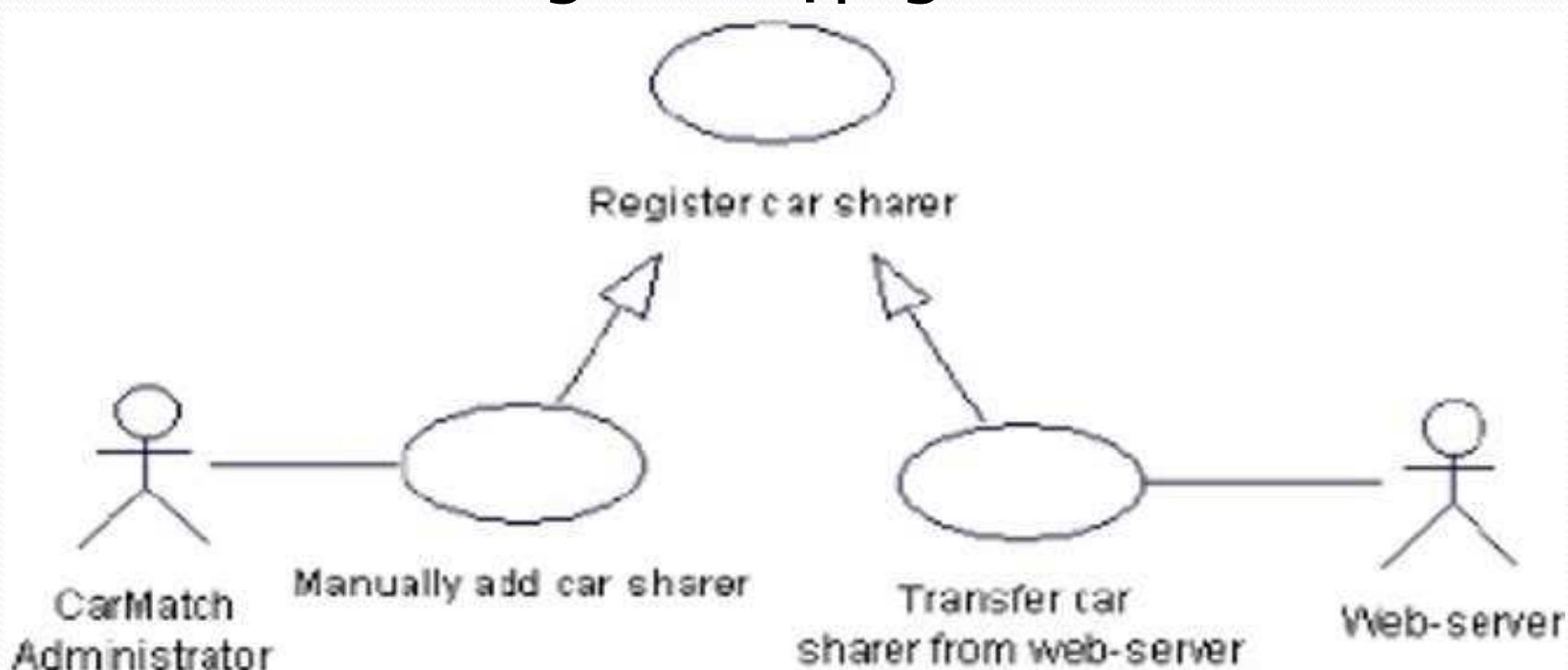
Định nghĩa và Kí hiệu (tt)

- Kết hợp generalization (tổng quát hóa) giữa các use case
 - Khi 2 use case có cùng một chức năng nhưng cách hoạt động lại không hoàn toàn giống nhau
 - Kết hợp generalization được vẽ bằng một biểu tượng hình tam giác trên đường nối hướng đến use case tổng quát



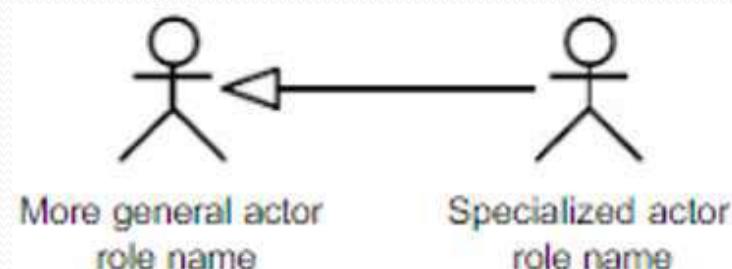
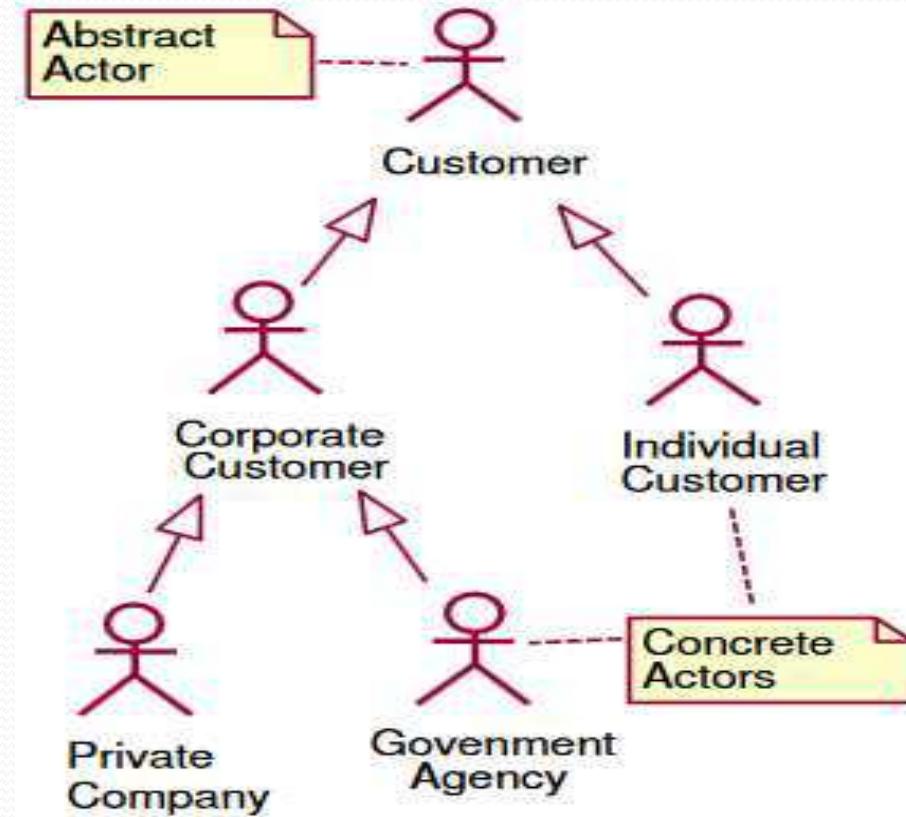
Định nghĩa và Kí hiệu (tt)

- Kết hợp generalization (tổng quát hóa) giữa các use case
 - VD: Các use case trong kết hợp generalization



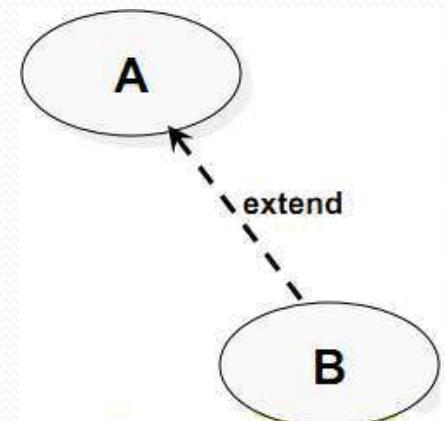
Định nghĩa và Kí hiệu (tt)

- Kết hợp generalization giữa các actor
 - Khi actor chuyên biệt là một trường hợp đặc biệt của actor tổng quát
- Kí hiệu kết hợp generalization giữa các actor



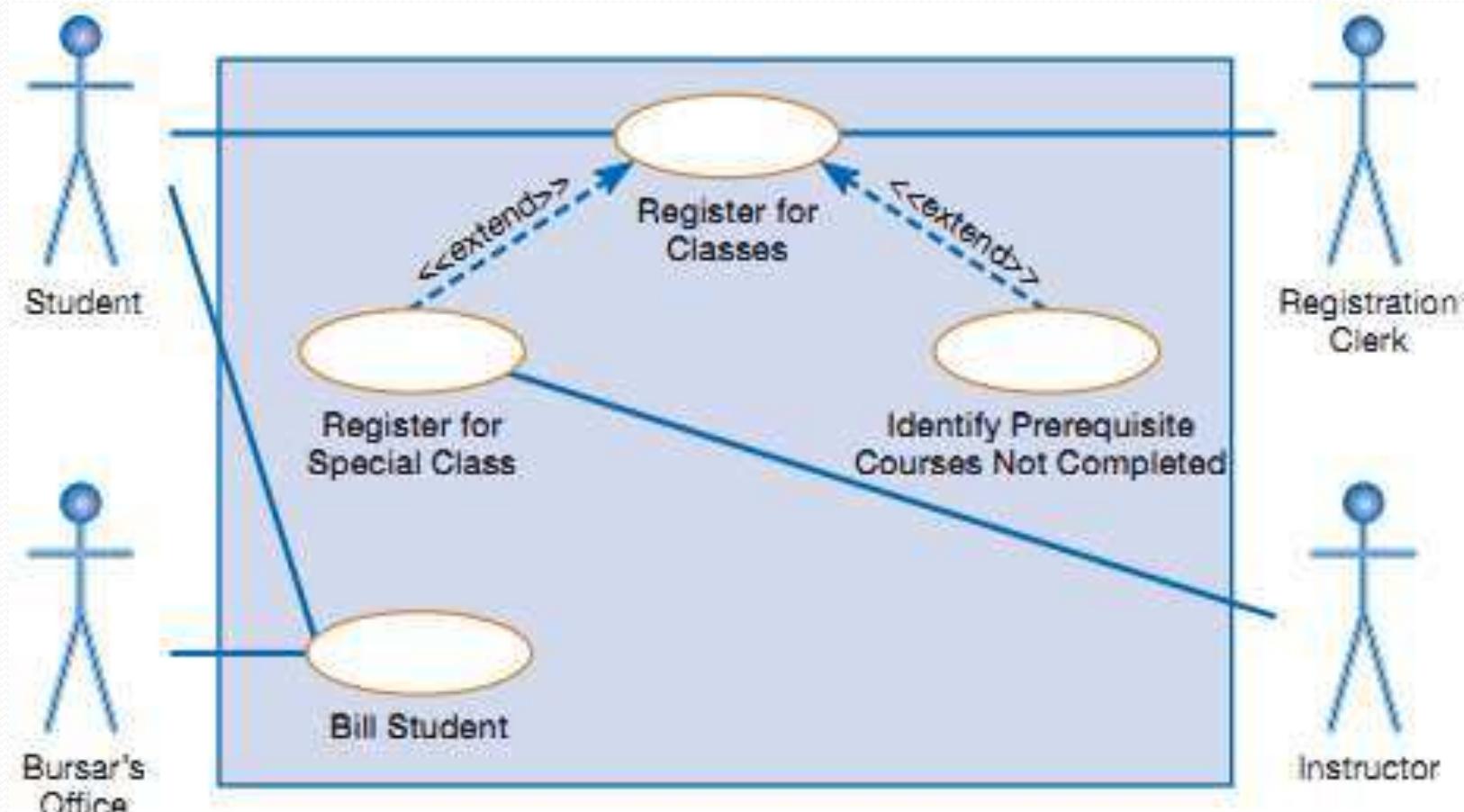
Định nghĩa và Kí hiệu (tt)

- *Mối quan hệ mở rộng (Extend relationship)* là một mối kết hợp giữa hai ca sử dụng nơi mà một ca sử dụng thêm những hành vi hoặc hành động mới cho ca sử dụng khác.
 - Mở rộng một ca sử dụng bằng cách thêm hành vi hoặc những hành động mới.
 - Ca sử dụng chuyên biệt (Specialized use case) mở rộng ca sử dụng chung (general use case).



Định nghĩa và Kí hiệu (tt)

- *Mỗi quan hệ mở rộng (Extend relationship)*
- *VD*

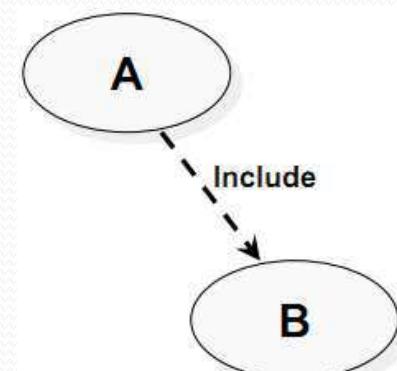


Hình 7-26

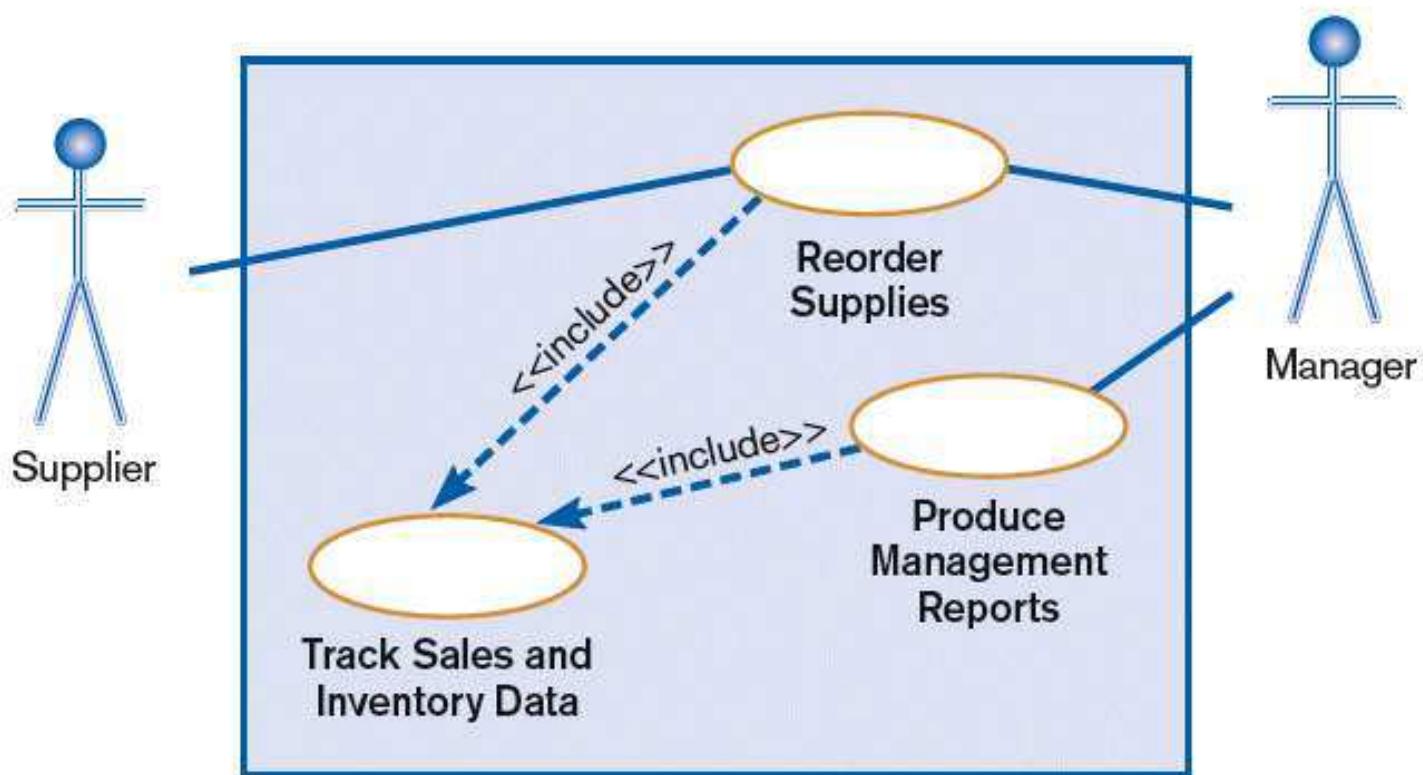
Một biểu đồ use case về HT đăng ký lớp học của trường ĐH

Định nghĩa và Kí hiệu (tt)

- *Mối quan hệ bao gồm (Include relationship)* là một mối kết hợp giữa hai ca sử dụng nơi mà một ca sử dụng dùng đến chức năng được chứa trong ca sử dụng khác.
 - Biểu diễn một use case được dùng (được cầu khẩn - invoked) bởi một use case khác
 - Liên kết với các giá trị mục tiêu chung (general purpose functions), được dùng bởi nhiều use case khác.



Định nghĩa và Kí hiệu (tt)

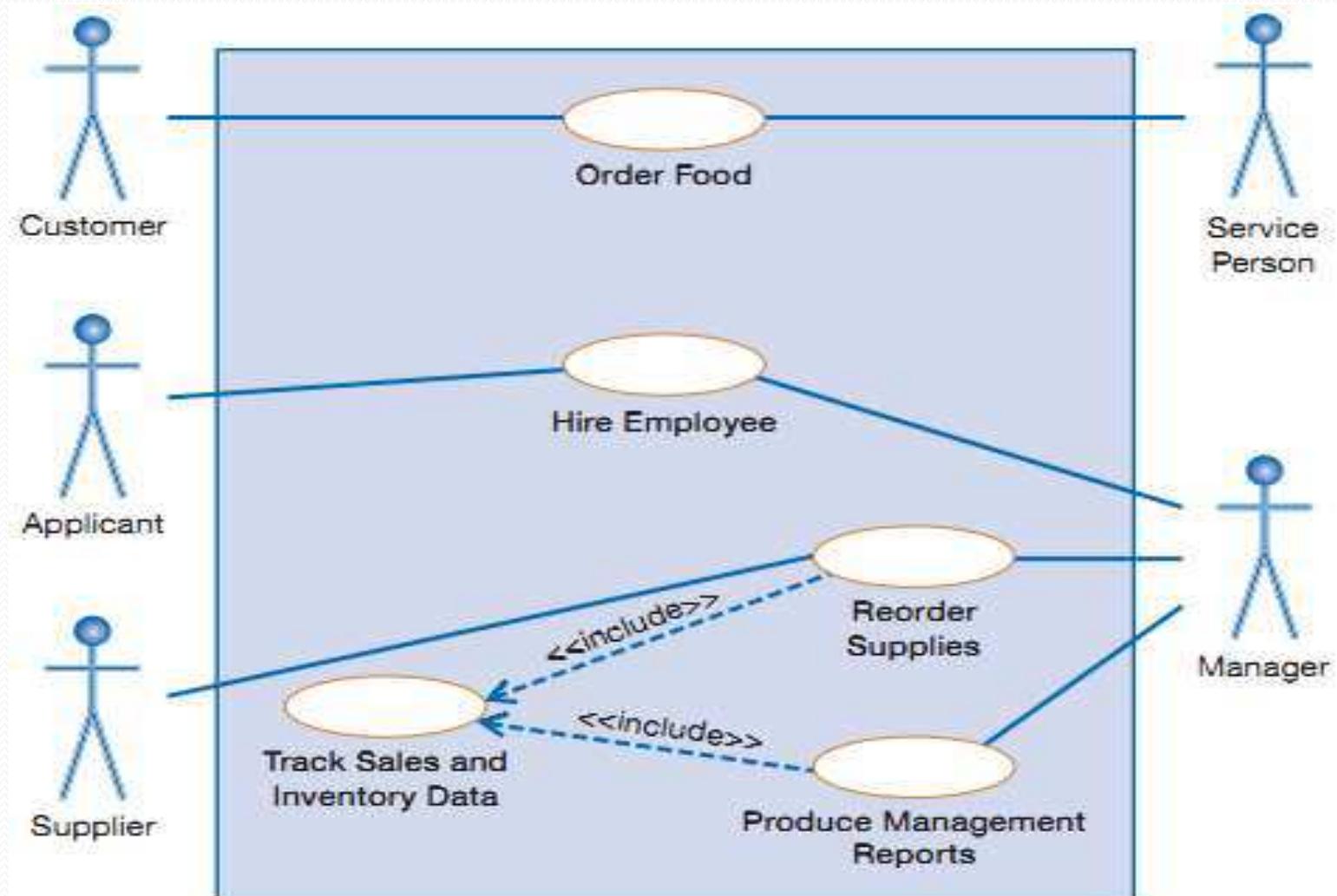


Hình 7-27

Một biểu đồ use case biểu diễn mối quan hệ bao gồm (include relationship)

Định nghĩa và Kí hiệu (tt)

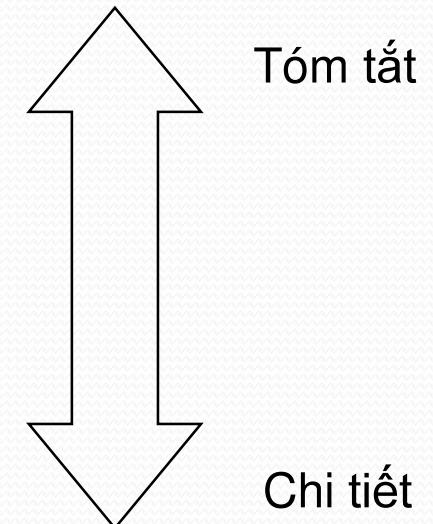
- *Biểu đồ use case cho Hoosier Burger*



Hình 7-28
Một biểu đồ use case của Hoosier Burger

Mức của Ca sử dụng

- **Mức (Level)** phải làm với mức chi tiết mà ca sử dụng đang được mô tả. Mức có thể dao động từ cao đến thấp, ở đó mức cao là tổng quát và trừu tượng, và thấp là chi tiết.
- 5 mức được đề nghị (Cockburn)
 1. White – như nhìn từ các đám mây
 2. Kite – “khung nhìn không trung”
 3. Blue – khung nhìn mực nước biển
 4. Fish – dưới mực nước biển
 5. Black – đáy biển



Mức của Ca sử dụng

- 5 mức được đề nghị (Cockburn)
 - White – Kite – Blue – Fish –Black

FIGURE 7-30

Use case levels and detail when moving from top to bottom

(Source: George, Joey F.; Batra, Dinesh; Valacich, Joseph S.; Hoffer, Jeffrey A., *Object-Oriented Systems Analysis and Design*, 2nd Ed., ©2007, pp. 174, 168, 175, 172, 176, 177. Reprinted and Electronically reproduced by permission of Pearson Education, Inc., New York, NY

	Buy parts to build cars
	Buy parts to build Escorts
	Order Escort parts from suppliers
	Choose supplier for part
	Encrypt data for secure transmission

Các ca sử dụng được viết lại (Written Use Cases) (hoặc *Viết bộ mô tả use case*)

- Tài liệu chứa những đặc tả chi tiết cho một use case
- Các nội dung có thể được viết như các văn bản đơn giản hoặc theo một định dạng cụ thể.
- Mô tả từng bước những gì sẽ xuất hiện trong một use case mong muốn.
- Có 3 pp thông dụng để viết bộ mô tả use case:
 - Đoạn văn mô tả chuỗi hành động mà use case thực hiện
 - Liệt kê 2 cột hành động của actor và trả lời hệ thống
 - Sử dụng tài liệu mẫu có chứa các phần ý nghĩa khác nhau của use case muốn ghi vào (*còn gọi là kịch bản use case*).

Các ca sử dụng được viết lại (Written Use Cases) (hoặc *Viết bộ mô tả use case*)

- PP Đoạn văn mô tả chuỗi hành động mà use case thực hiện
 - Bộ mô tả use case đơn giản cho **Register car sharer**
 - *Người dùng sẽ nhập tên, địa chỉ và số điện thoại của tiệm năn
car sharer. Với mỗi quãng đường mà người muốn chia sẻ, địa chỉ
bắt đầu, địa chỉ đến, thời gian bắt đầu và thời gian kết thúc của
một quãng đường khi được đăng nhập vào*

Các ca sử dụng được viết lại (Written Use Cases) (hoặc Viết bộ mô tả use case)

- PP Liệt kê 2 cột hành động của actor và trả lời hệ thống

- Hình mô tả use case cho *Register car sharer* dùng cột

	Tác nhân	Hệ thống
1	Người dùng nhập tên và địa chỉ của người chia sẻ xe.	Hệ thống xác nhận địa chỉ khớp với cơ sở dữ liệu địa lý (<i>geographical database</i>).
2	Người dùng nhập số điện thoại của người chia sẻ xe.	Hệ thống yêu cầu cung cấp các thông tin chi tiết của lô trình.
3	Người dùng nhập thời gian bắt đầu và địa chỉ xuất phát.	Hệ thống xác nhận địa chỉ xuất phát khớp với cơ sở dữ liệu địa lý.
4	Người dùng nhập thời gian kết thúc và địa chỉ đích.	Hệ thống xác nhận địa chỉ đích khớp với cơ sở dữ liệu địa lý.
5	Người dùng hoặc nhập lô trình khác (quay lại bước 3) hoặc lưu và thoát.	Hệ thống hỏi người dùng có muốn nhập vào lô trình mới hay không.
		Hệ thống hoặc nhắc nhập thêm lô trình (quay lại bước 3) hoặc lưu người chia sẻ xe, thông tin chi tiết của lô trình và thoát khỏi use case.

● PP viết kịch bản của ca sử dụng

Use Case Title: Browse catalog
Primary Actor: Customer
Level: Kite (summary)
Stakeholders: Customer, credit bureau
Precondition: Customer accesses the hotel website
Minimal Guarantee: Rollback of any uncompleted transaction
Success Guarantees: Reservation held with one night's deposit
Trigger: Customer accesses hotel homepage
Main Success Scenario:
1. Customer searches for hotel location and room availability for desired time period. 2. Customer makes reservation for desired room for desired time period. 3. Customer holds reservation by authorizing a deposit for one night's stay. 4. Credit bureau verifies that customer has necessary credit for deposit. 5. Customer requests confirmation of reservations.
Extensions:
1a. Hotel property search function is not available. 1a1. Customer quits site
1b. Specific hotel room not available for desired time period. 1b1. Customer quits site. 1b2. Customer searches for different hotel for desired time period. 1b3. Customer searches for same hotel for different time period
2a. Making reservation transaction is interrupted. 2a1. Transaction rolled back. Customer starts again. 2a2. Transaction rolled back. Customer quits site.
3a. Holding reservation transaction is interrupted. 3a1. Transaction rolled back. Customer starts again. 3a2. Transaction rolled back. Customer quits site.
4a. Credit bureau cannot verify that customer has necessary credit. 4a1. Customer notified of issue. Transaction rolled back. Customer quits site. 4a2. Customer notified of issue. Transaction rolled back. Customer begins reservation process again with different credit card.
5a. Confirmation of transaction is interrupted. 5a1. Customer seeks other means of confirmation. 5a2. Customer quits site.

FIGURE 7-32

Kite Level Written Use Case for Making a Hotel Room Reservation

(Source: George, Joey F.; Barna, Dinesh; Valacich, Joseph S.; Hoffer, Jeffrey A., *Object-Oriented Systems Analysis and Design*, 2nd Ed., ©2007, pp. 174, 168, 173, 172, 176, 177. Reprinted and Electronically reproduced by permission of Pearson Education, Inc.)

Figure 7-29

Một mẫu cho các ca sử dụng
được viết lại

(Source: Cockburn, Alistair, *Writing Effective Use Cases*, 1st ed., © 2001. Reprinted and Electronically reproduced by permission of Pearson Education, Inc. Upper Saddle River, New Jersey.)



Định dạng mẫu cho Ca sử dụng được viết lại

- **Title** – tên mô tả, khớp với tên trong biểu đồ use case
- **Primary actor** – luôn là một vai trò người dùng
- **Stakeholders** – bất kỳ nhóm hoặc cá nhân có cùng mối quan tâm về chức năng của ca sử dụng.
- **Precondition** – các điều kiện phải được thỏa mãn để thực thi use case
- **Minimal guarantee** – các điều ra có thể được mong đợi nếu hành động cần thiết chung thất bại (service attempt failed).
- **Success guarantee** – các điều ra có thể được mong đợi nếu hành động thành công.
- **Trigger** – một sự kiện hoặc hành động để bắt đầu use case
- **Main success scenario** – mô tả trình tự những tương tác giữa tác nhân và use case trong khi thực thi use case.
- **Extensions** – mô tả chi tiết cách thức lỗi được xử lý

Ví dụ

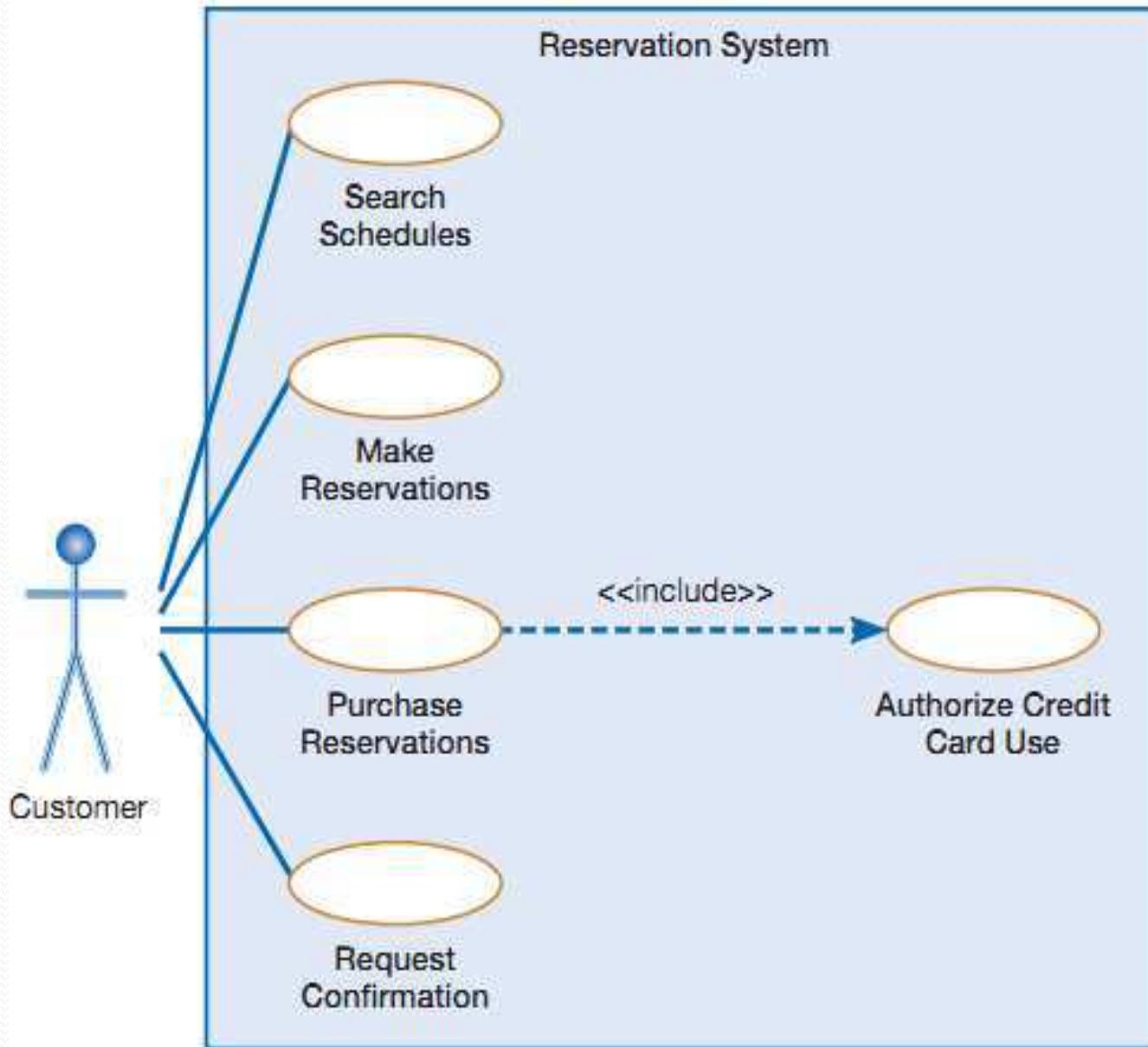


FIGURE 7-31

A use case diagram for a reservation system

(Source: George, Joey F.; Batra, Dinesh; Valacich, Joseph S.; Hoffer, Jeffrey A., *Object-Oriented Systems Analysis and Design*, 2nd Ed., ©2007, pp. 174, 168, 175, 172, 176, 177. Reprinted and Electronically reproduced by permission of Pearson Education, Inc., New York, NY)

Use Case Title: Browse catalog
Primary Actor: Customer
Level: Kite (summary)
Stakeholders: Customer, credit bureau
Precondition: Customer accesses the hotel website
Minimal Guarantee: Rollback of any uncompleted transaction
Success Guarantees: Reservation held with one night's deposit
Trigger: Customer accesses hotel homepage
Main Success Scenario:
1. Customer searches for hotel location and room availability for desired time period.
2. Customer makes reservation for desired room for desired time period.
3. Customer holds reservation by authorizing a deposit for one night's stay.
4. Credit bureau verifies that customer has necessary credit for deposit.
5. Customer requests confirmation of reservations.
Extensions:
1a. Hotel property search function is not available.
1a1. Customer quits site
1b. Specific hotel room not available for desired time period.
1b1. Customer quits site.
1b2. Customer searches for different hotel for desired time period.
1b3. Customer searches for same hotel for different time period
2a. Making reservation transaction is interrupted.
2a1. Transaction rolled back. Customer starts again.
2a2. Transaction rolled back. Customer quits site.
3a. Holding reservation transaction is interrupted.
3a1. Transaction rolled back. Customer starts again.
3a2. Transaction rolled back. Customer quits site.
4a. Credit bureau cannot verify that customer has necessary credit.
4a1. Customer notified of issue. Transaction rolled back. Customer quits site.
4a2. Customer notified of issue. Transaction rolled back. Customer begins reservation process again with different credit card.
5a. Confirmation of transaction is interrupted.
5a1. Customer seeks other means of confirmation.
5a2. Customer quits site.

FIGURE 7-32

Kite Level Written Use Case for Making a Hotel Room Reservation

(Source: George, Joey F.; Barre, Dinesh; Valacich, Joseph S.; Hoffer, Jeffrey A., *Object-Oriented System Analysis and Design*, 2nd Ed., ©2007, pp. 174, 168, 175, 172, 176, 177. Reprinted and Electronically reproduced by permission of Pearson Education, Inc., New York, NY.)

Ứng dụng TMĐT: Lập mô hình quy trình WebStore PVF

TABLE 7-5 System Structure of the WebStore and Corresponding Functions

WebStore System	Functions
□ Main Page <ul style="list-style-type: none">• Product Line (catalog)<ul style="list-style-type: none">✓ Desks✓ Chairs✓ Tables✓ File Cabinets• Shopping Cart• Checkout• Account Profile• Order Status/History• Customer Comments	Browse Catalog Place Order Place Order Maintain Account Check Order
□ Company Information	
□ Feedback	
□ Contact Information	

Ứng dụng thương mại điện tử: Biểu đồ Use Case WebStore

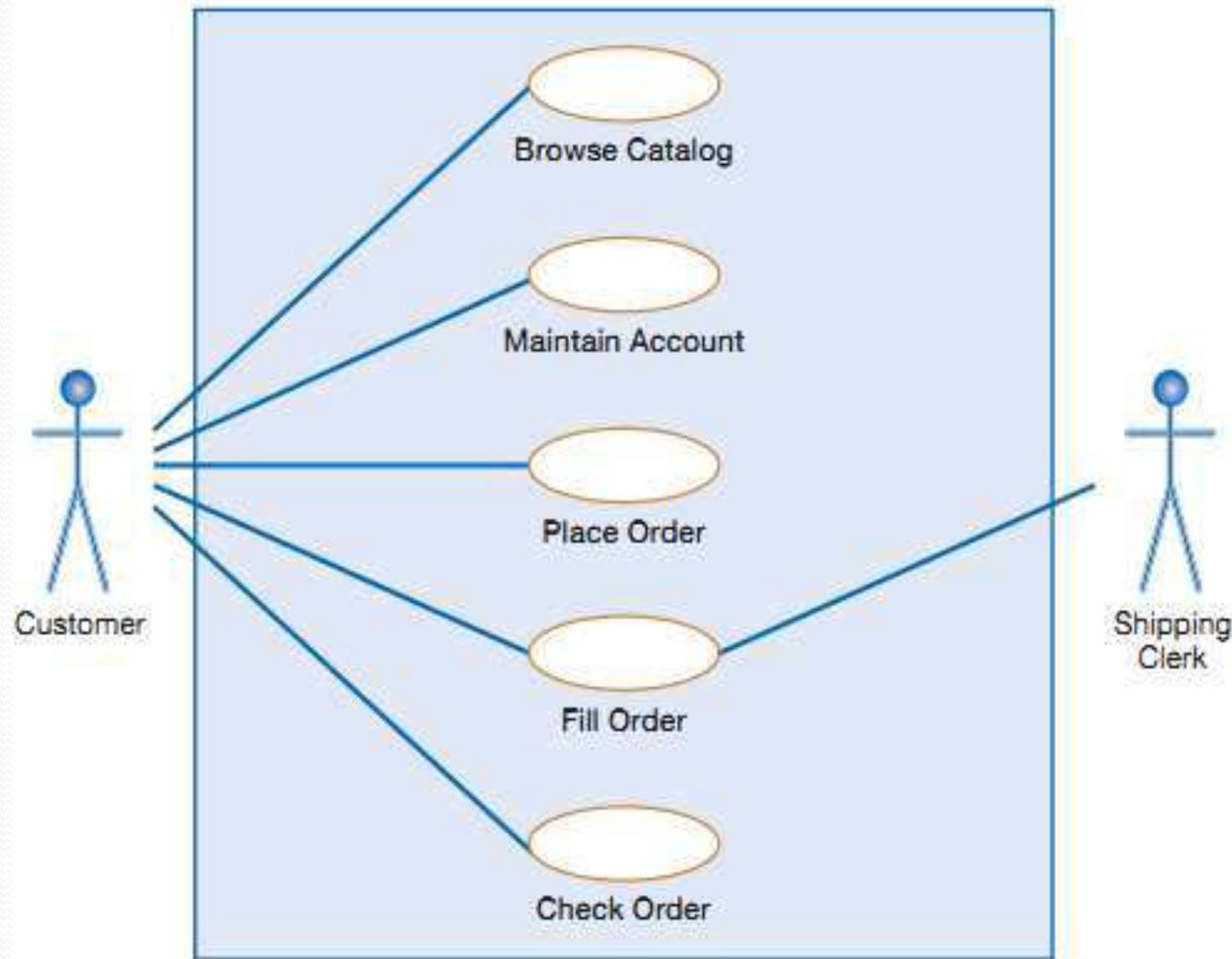


FIGURE 7-33

WebStore use case diagram
Source: George, Joey F.; Batra, Dinesh; Valacich, Joseph S.; Hoffer, Jeffrey A., *Object-Oriented Systems Analysis and Design*, 2nd Ed., ©2007, pp. 174, 168, 175, 172, 176, 177. Reprinted and Electronically reproduced by permission of Pearson Education, Inc., New York, NY.

Ứng dụng thương mại điện tử: Written Use Cases

Hình 7-34

Use case được viết lại mức diều (kite level) của Jim Woo về việc mua sản phẩm tại WebStore PVF.

((Source: George, Joey F.; Batra, Dinesh; Valacich, Joseph S.; Hoffer, Jeffrey A., *Object-Oriented Systems Analysis and Design*, 2nd Ed., ©2007, pp. 174, 168, 175, 172, 176, 177. Reprinted and Electronically reproduced by permission of Pearson Education, Inc., New York, NY.)

Use Case Title: Buying a PVF Product at WebStore
Primary Actor: Customer
Level: Kite (summary)
Stakeholders: Customer, shipping clerk
Precondition: Customer accesses the WebStore website
Minimal Guarantee: Rollback of any uncompleted transaction
Success Guarantees: Order filled
Trigger: Customer accesses WebStore homepage
Main Success Scenario:
1. Customer browses catalog. 2. Customer places order for desired product(s). 3. Shipping clerk fills order. 4. Customer checks status of order.
Extensions:
1a. Catalog is not available. 1a1. Customer quits site. 1a2. Customer takes action to gain access to catalog.
2a. Order transaction is interrupted. 2a1. Transaction rolled back. Customer starts again. 2a2. Transaction rolled back. Customer quits site.
3a. Item is out of stock. 3a1. Shipping clerk notifies customer. Customer waits for stock to be replenished. 3a2. Shipping clerk notifies customer. Customer cancels order.
4a. Order status is not available. 4a1. Customer quits site. 4a2. Customer takes action to gain access to order status.

Ứng dụng TMĐT: Written Use Cases

Hình 7-35

Mẫu đã hoàn thành của Jim Woo cho ca sử dụng Browse catalog của WebStore PVF.

(Source: George, Joey F.; Batra, Dinesh; Valacich, Joseph S.; Hoffer, Jeffrey A., Object-Oriented Systems Analysis and Design, 2nd Ed., ©2007, pp.174, 168, 175, 172, 176, 177. Reprinted and Electronically reproduced by permission of Pearson Education, Inc., New York, NY.

Use Case Title: Browse catalog
Primary Actor: Customer
Level: Sea level (user goal)
Stakeholders: Customer
Precondition: Customer must be online with Web access
Minimal Guarantee: Rollback of any uncompleted transaction; system logs progress until failure
Success Guarantees: Files customer desires load correctly
Trigger: Customer accesses WebStore homepage
Main Success Scenario:
1. Cookie created on customer hard drive. 2. Customer selects category of item to view from list (e.g., home, office, patio). 3. Customer selects subcategory of item to view from list (e.g., home is subdivided into kitchen, dining room, bedroom, living room, den, etc.). 4. Customer selects specific item from list in subcategory to view (e.g., TV stand in den). 5. Customer selects specific item from list of products (e.g., Smith & Wesson TV stand). 6. Customer clicks on thumbnail photo of item to get regular-sized photo to view. 7. Customer selects "Product Specifications" to get detailed information on product. 8. Customer uses Web browser "Back" button to go back to see other products or other rooms or other types of furniture. 9. Customer selects from choices on menu bar to go elsewhere, either "Other Types of Furniture," "WebStore Home," or "PVF Home."
Extensions:
1.a. Cookie cannot be created. <ul style="list-style-type: none">1.a.1. Message created indicates to customer that browsing is not possible because his or her Web browser does not allow for the creation of cookies.1.a.2. Customer either adjusts the browser's cookie settings and tries again or leaves the site.
6.a. Full-sized photo does not load. <ul style="list-style-type: none">6.a.1. Customer gets a broken-link symbol.6.a.2. Customer hits the refresh button and the photo loads successfully.6.a.3. Customer hits the refresh button and the photo does not load successfully; customer leaves the site.
2-7.a. The requested Web page does not load or cannot be found. <ul style="list-style-type: none">2-7.a.1. Customer gets a "page not found" error page in browser.2-7.a.2. Customer hits the refresh button and the requested page loads successfully.2-7.a.3. Customer hits the refresh button and the requested page does not load successfully; customer leaves the site.

Phân tích và thiết kế hướng đối tượng

- Lập mô hình quy trình: Biểu đồ hoạt động (Activity Diagrams)
 - Mục tiêu
 - Hiểu làm thế nào để biểu diễn logic hệ thống với các sơ đồ hoạt động (activity diagrams)

Lập mô hình quy trình: Biểu đồ hoạt động (Activity Diagrams)

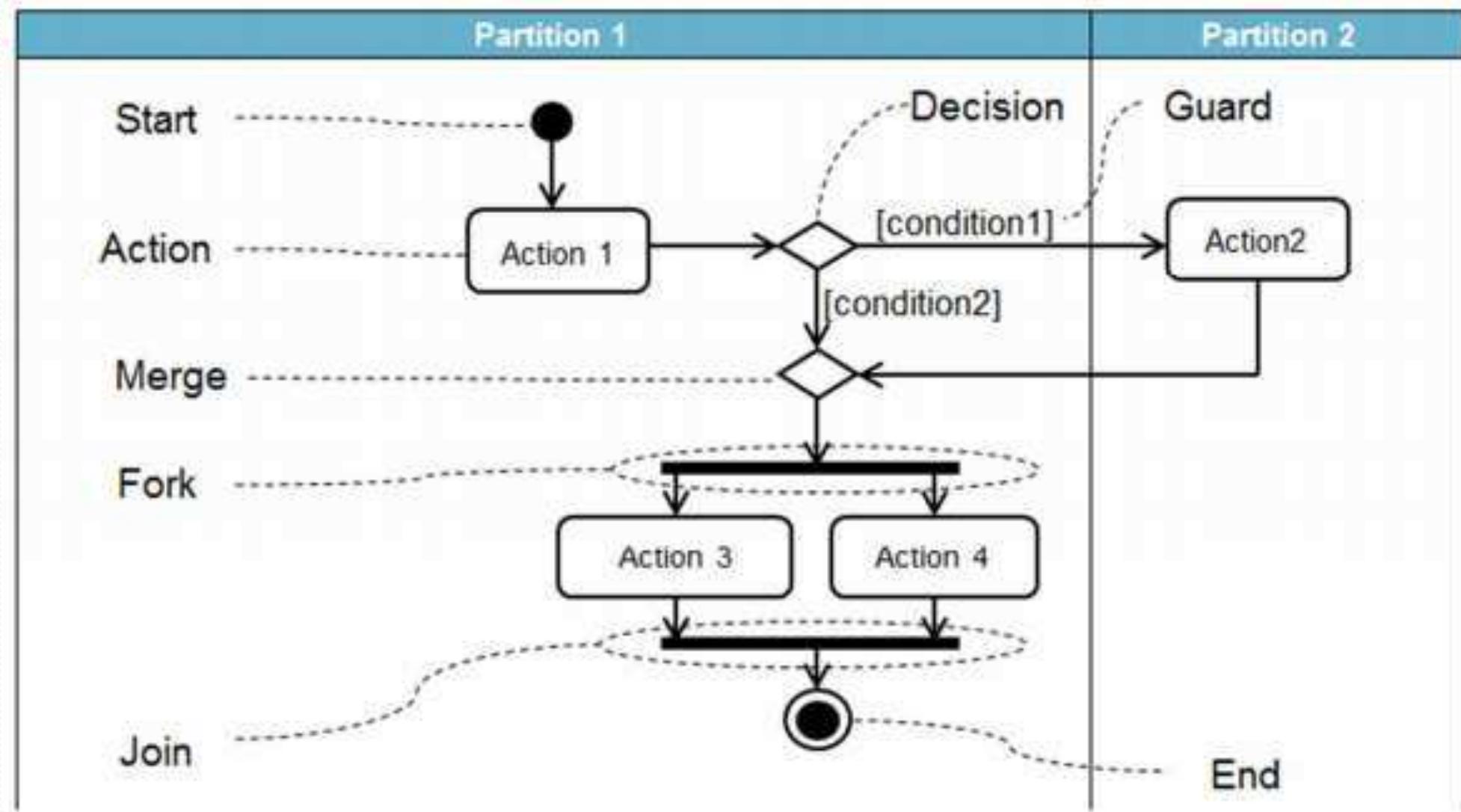
- Biểu diễn logic điều kiện cho chuỗi các hoạt động hệ thống cần thiết để hoàn thành một quy trình nghiệp vụ.
- Biểu diễn rõ ràng các hành vi song song và thay thế.
- Có thể được dùng để biểu diễn logic một use case.

Sử dụng Biểu đồ Hoạt động để:

- Biểu diễn luồng điều khiển theo hoạt động.
- Giúp phân tích use case để hiểu hành động gì cần diễn ra.
- Giúp xác định các phần mở rộng trong một use case.
- Lập mô hình luồng công việc (work flow) và các quy trình nghiệp vụ.
- Lập mô hình các bước tuần tự và song hành theo một quy trình tính toán.

Lập mô hình quy trình: Biểu đồ hoạt động (Activity Diagrams)

- Các kí hiệu của biểu đồ hoạt động :



Lập mô hình quy trình: Biểu đồ hoạt động (Activity Diagrams)

- Ký hiệu cơ bản của biểu đồ hoạt động

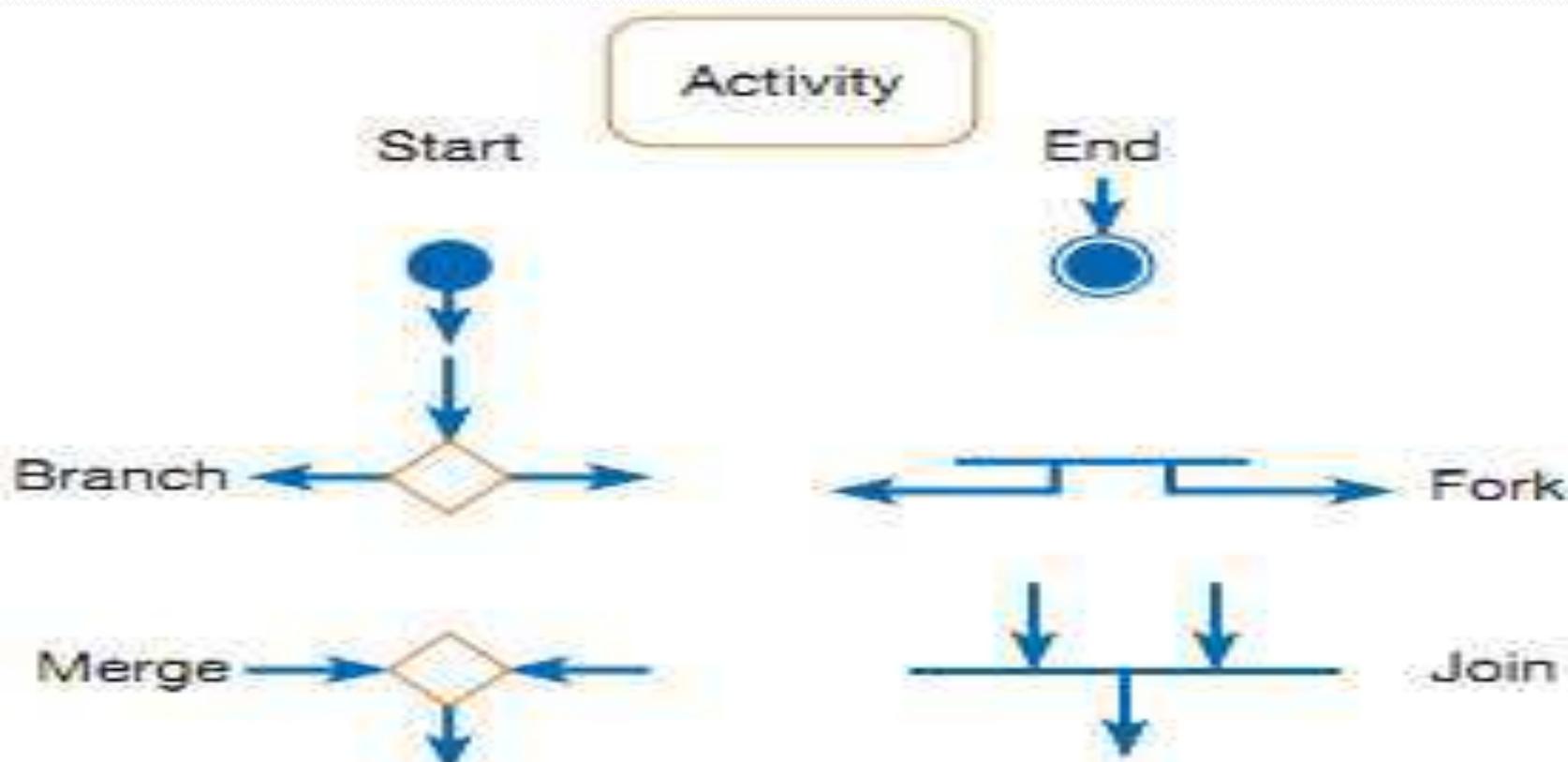


FIGURE 7-36

Basic notation for activity diagrams.

Lập mô hình quy trình: Biểu đồ hoạt động (Activity Diagrams)

- Các yếu tố của biểu đồ hoạt động:

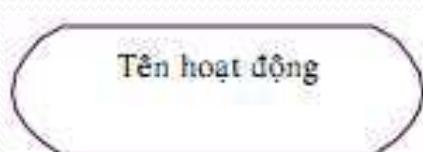
- **Nút bắt đầu (Start)** là điểm bắt đầu luồng hoặc quy trình. Trong một biểu đồ, chỉ có một trạng thái bắt đầu.



- **Nút kết thúc (End)** là điểm kết thúc luồng hoặc quy trình. Trong một luồng có thể có nhiều trạng thái kết thúc, chúng cho biết có các tiến trình khác theo sau



- **Hoạt động (Activity)**: một hành vi (behavior) mà một đối tượng thực hiện trong một trạng thái cụ thể.



Lập mô hình quy trình: Các biểu đồ hoạt động (tt.)

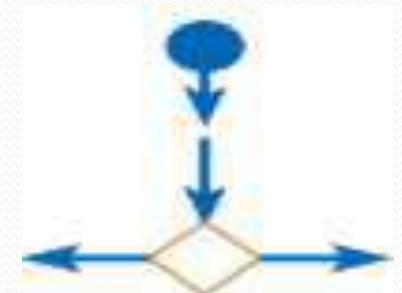
- Các yếu tố của biểu đồ hoạt động:

- **Nút quyết định (decision point)**

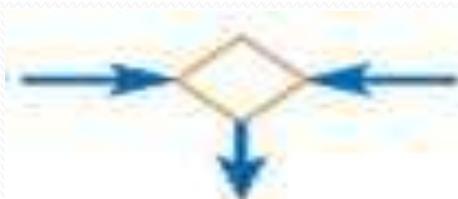
- Ký hiệu **Nút quyết định (decision point)** có thể có một hoặc nhiều sự thay đổi đi vào và một hoặc nhiều sự thay đổi đi ra



- **Nhánh (Branch):** một biểu tượng hình thoi chứa một điều kiện mà các kết quả của nó chứa các sự chuyển tiếp (transitions) đến các đường hoạt động (paths of activities) khác nhau.



- **Hợp nhất (Merge):** một biểu tượng vòng (circular symbol) nơi các đường khác nhau hội tụ (different paths converge)



Lập mô hình quy trình: Các biểu đồ hoạt động (tt.)

- Các yếu tố của biểu đồ hoạt động:
 - **Thanh đồng bộ hóa (synchronization bar):** để kết hợp nhiều luồng hoạt động **tiếp nối (join)** hoặc để **rẽ nhánh (fork)** cho những luồng có khả năng thực hiện song song.



Lập mô hình quy trình: Các biểu đồ hoạt động (tt.)

- Các yếu tố của biểu đồ hoạt động:
 - **Rẽ nhánh (Fork)**: phần bắt đầu các hoạt động song song.
 - **Tiếp nối (Join)**: phần kết thúc của các hoạt động song song.
 - **Một trạng thái (state)** là điểm mà các sự kiện cần đạt tới trước khi hoạt động tiếp tục.
 - *Trạng thái là các hoạt động (Activity) thể hiện sự thi hành của các thao tác.*
 - *Kí hiệu giống hoạt động*

Lập mô hình quy trình: Biểu đồ hoạt động (Activity Diagrams)

- Các yếu tố của biểu đồ hoạt động:

- **Sự chuyển tiếp (Transition):** là việc chuyển đổi giữa các hoạt động hoặc trạng thái.

- Một transition là sự di chuyển từ:

- một hoạt động này sang một hoạt động khác

- thay đổi từ trạng thái này sang trạng thái khác

- hoặc chuyển tiếp từ một trạng thái sang một hoạt động hoặc ngược lại

- Transition được biểu diễn bằng một mũi tên

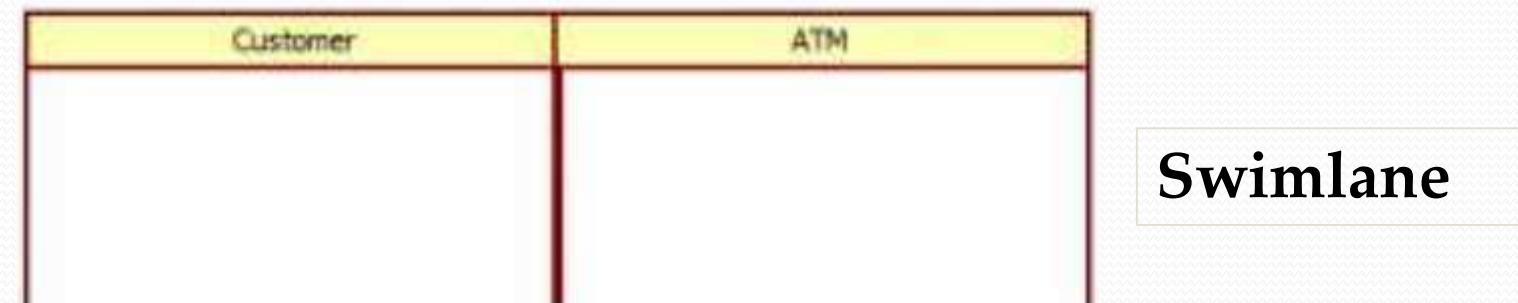


Một chuyển tiếp từ một hoạt động này sang một hoạt động khác

Lập mô hình quy trình: Các biểu đồ hoạt động (tt.)

- Các yếu tố của biểu đồ hoạt động:

- **Tuyến công việc/Đường phân dòng nghiệp vụ (Swimlanes):** các cột đại diện các đơn vị tổ chức khác nhau của hệ thống → dùng để xác định đối tượng nào tham gia hoạt động nào trong một quy trình.



Customer thì Insert Card còn ATM Machine thì Show Balance

Lập mô hình quy trình: Các biểu đồ hoạt động (tt.)

- Hình 7-37 chỉ ra một sơ đồ hoạt động đơn giản minh họa logic điều kiện.

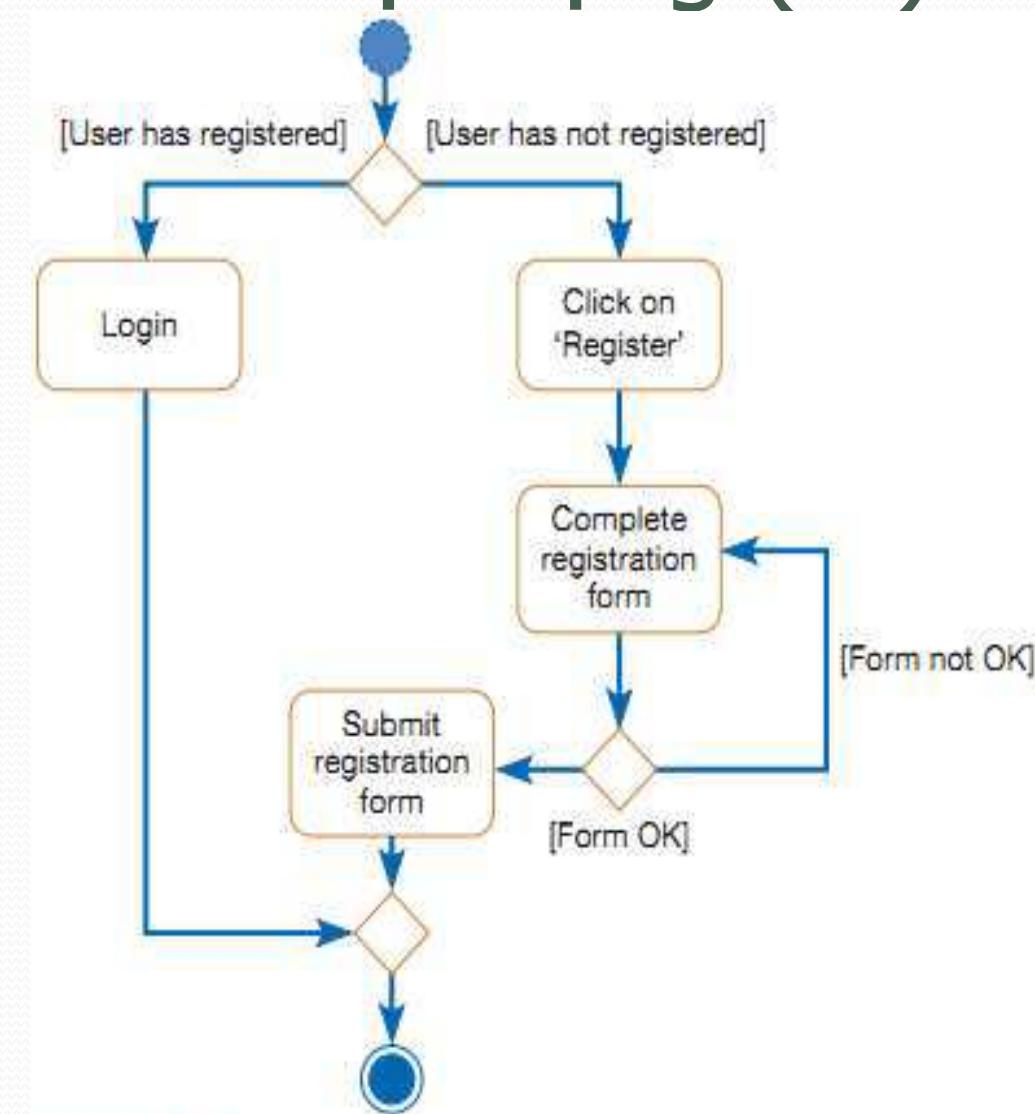


FIGURE 7-37

Simple activity diagram showing conditional logic.

Lập mô hình quy trình: Các biểu đồ hoạt động (tt.)

- Hình 7-38: Sơ đồ hoạt động cho việc đặt bánh pizza

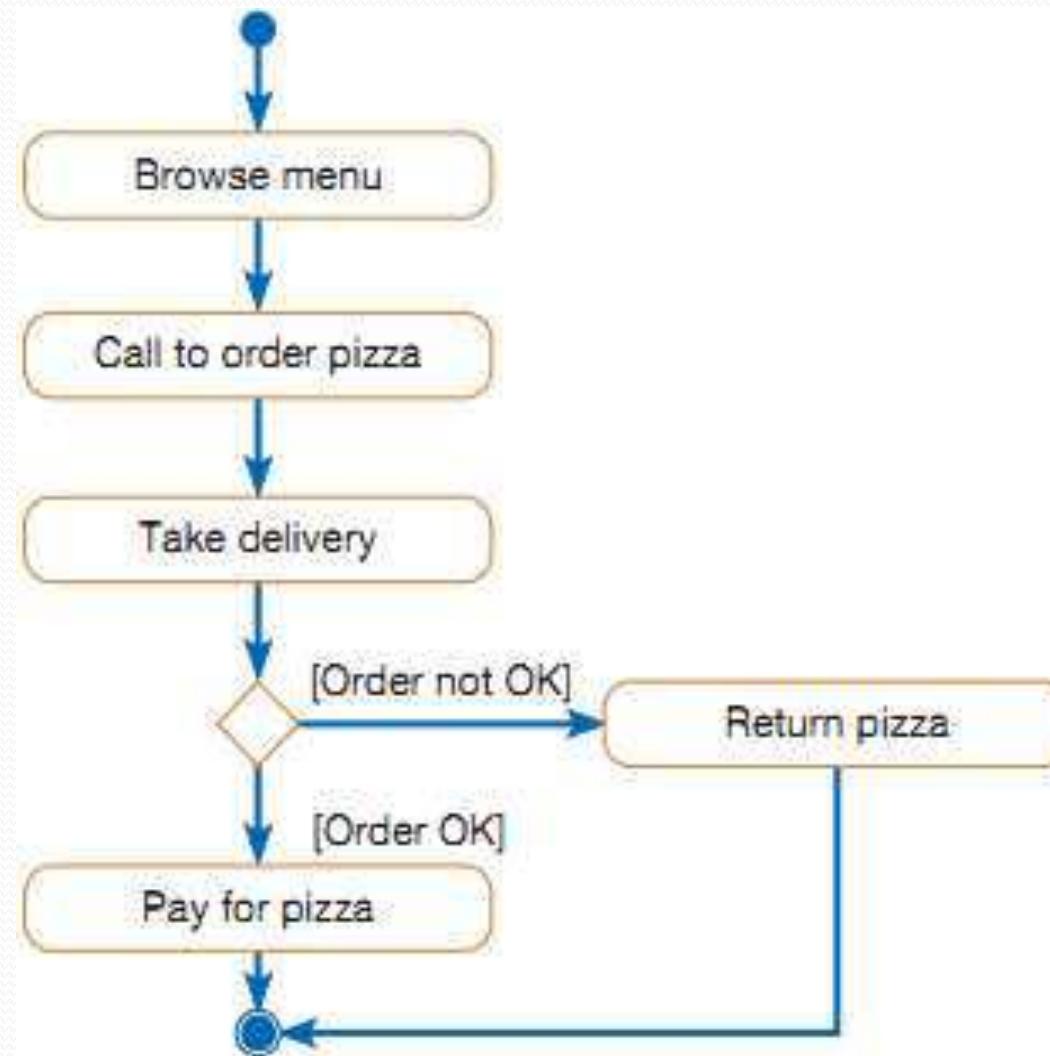
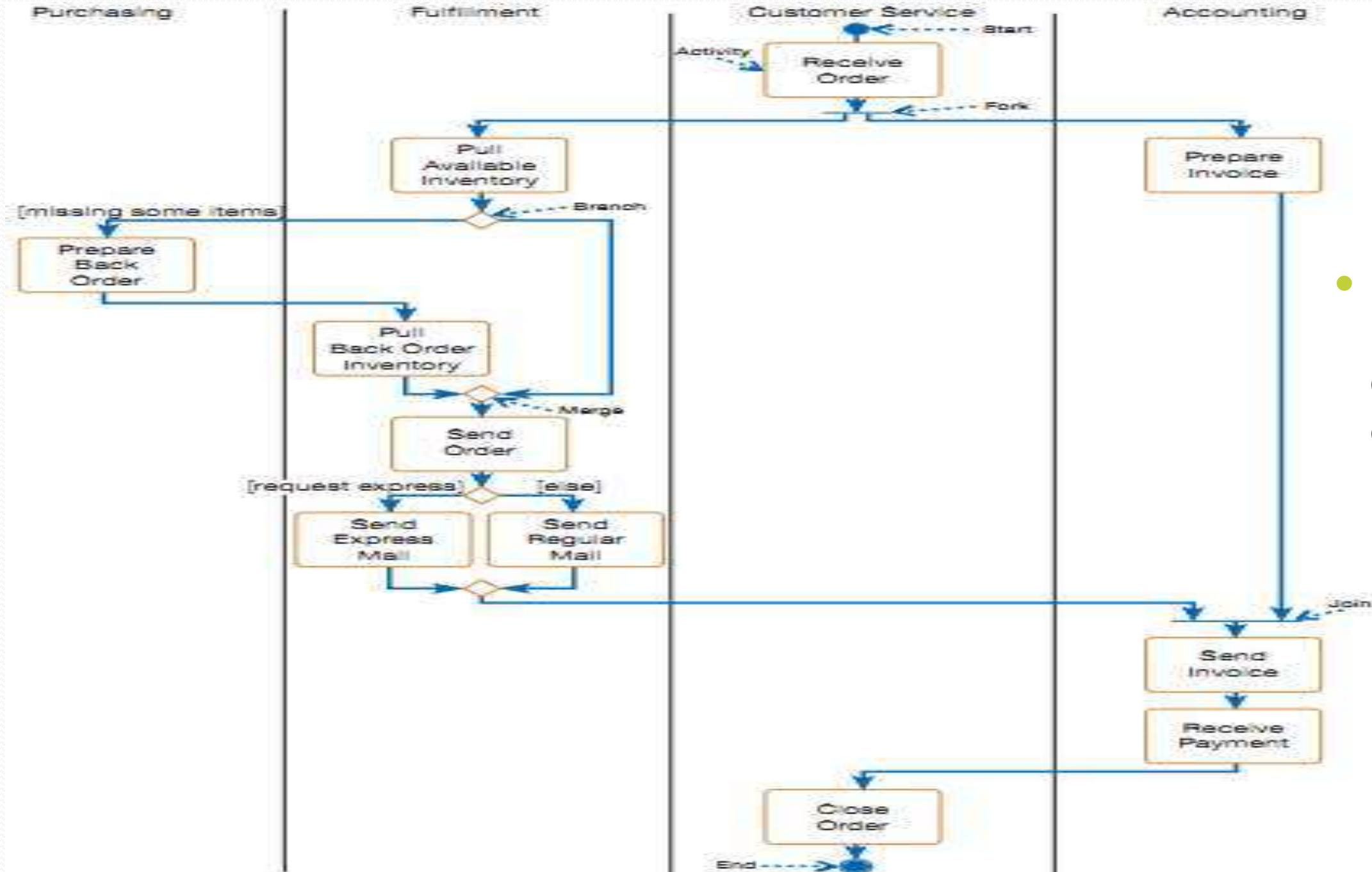


FIGURE 7-38

A simple activity diagram for ordering pizza.



• **Hình 7-39**
Biểu đồ hoạt động của một quy trình đặt hàng của khách

Phân tích và thiết kế hướng đối tượng

- Lập mô hình động: Biểu đồ trình tự (Sequence Diagrams)
 - Mục tiêu
 - Hiểu cách biểu diễn logic hệ thống với các biểu đồ trình tự

Lập mô hình động: Biểu đồ trình tự (Sequence Diagrams)

- **Sequence diagram:** mô tả các tương tác giữa các đối tượng trong một khoảng thời gian nhất định. Gồm:
 - Mô tả tương tác giữa actor và các đối tượng hệ thống.
 - Mô tả sự tương tác giữa các đối tượng theo trình tự thời gian
 - Thường được dùng để biểu diễn các bước thực hiện trong một kịch bản (Scenario) của một use-case

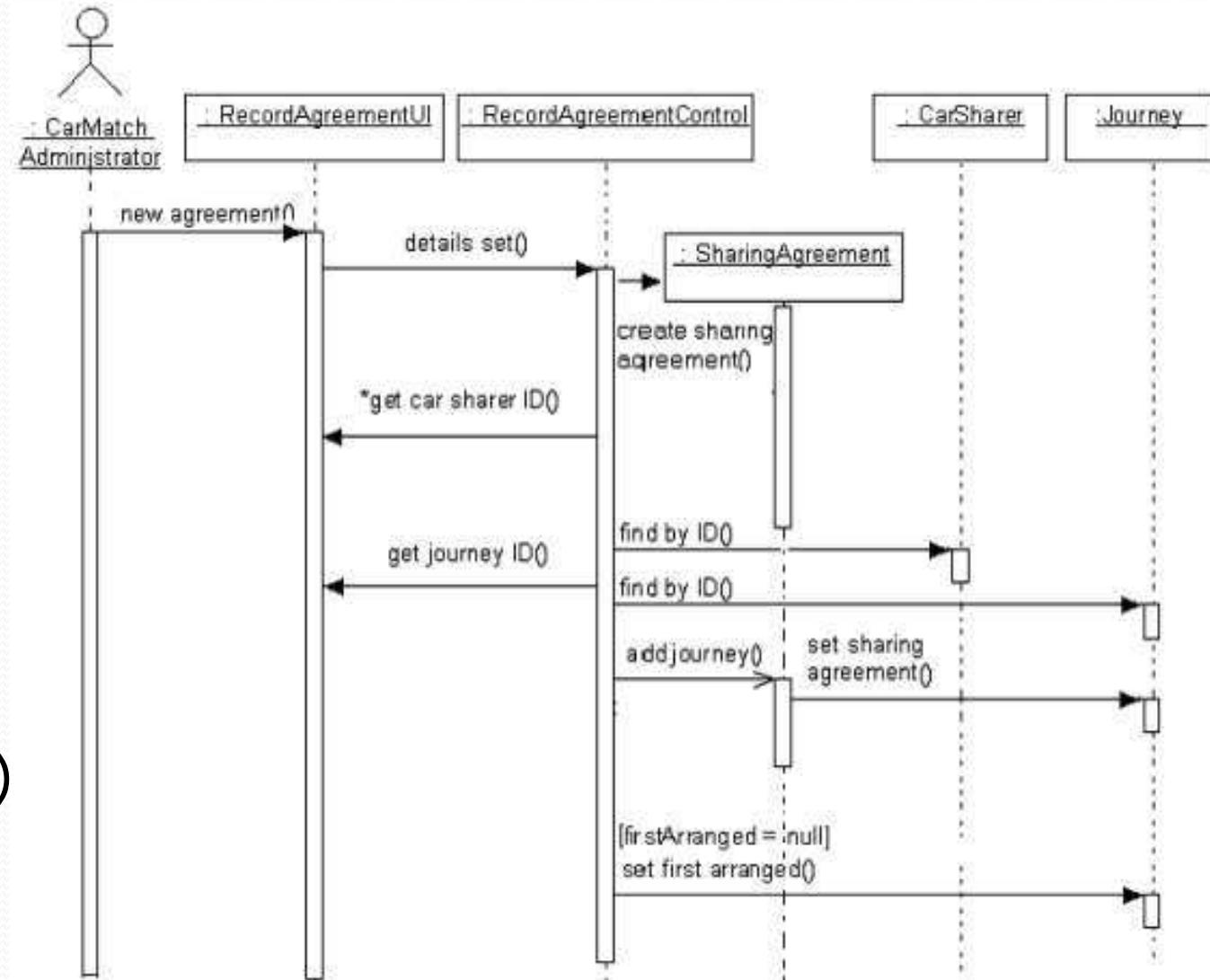
Lập mô hình động: Biểu đồ trình tự (Sequence Diagrams)

- **Sequence diagram:** mô tả các tương tác giữa các đối tượng trong một khoảng thời gian nhất định
 - Có thể được trình diễn theo một mẫu chung (generic form) hoặc theo một mẫu dạng thể hiện (instance form).
 - **Mẫu chung** biểu diễn tất cả các trình tự có thể của các tương tác – các trình tự tương ứng tất cả các kịch bản của một use case.
 - **Mẫu dạng thể hiện** biểu diễn trình tự cho một kịch bản duy nhất.

Lập mô hình động: Biểu đồ trình tự (Sequence Diagrams) (tt.)

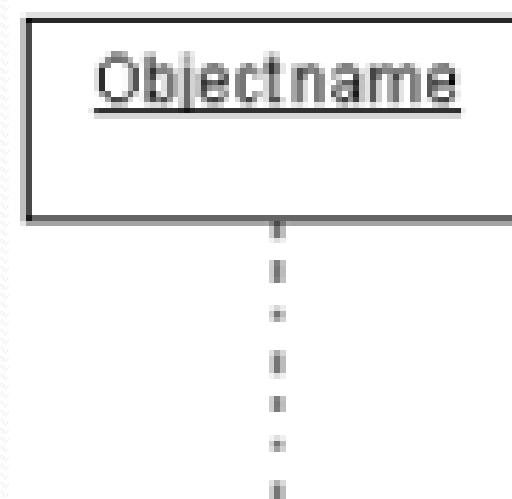
- Biểu đồ trình tự có 2 trục

- Trục dọc chỉ thời gian (đường đời/đường sinh tồn - **Lifeline**) → *đường đứt nét*
- Trục ngang biểu diễn tập hợp các đối tượng (**Objects**)



Lập mô hình động: Biểu đồ trình tự (Sequence Diagrams) (tt.)

- Các yếu tố của một biểu đồ trình tự
 - **Các đối tượng** (Objects): được biểu diễn bởi các hộp tại đỉnh của biểu đồ
 - **Đường đời** (Lifeline): thời gian khi một đối tượng tồn tại



Lập mô hình động: Biểu đồ trình tự (Sequence Diagrams) (tt.)

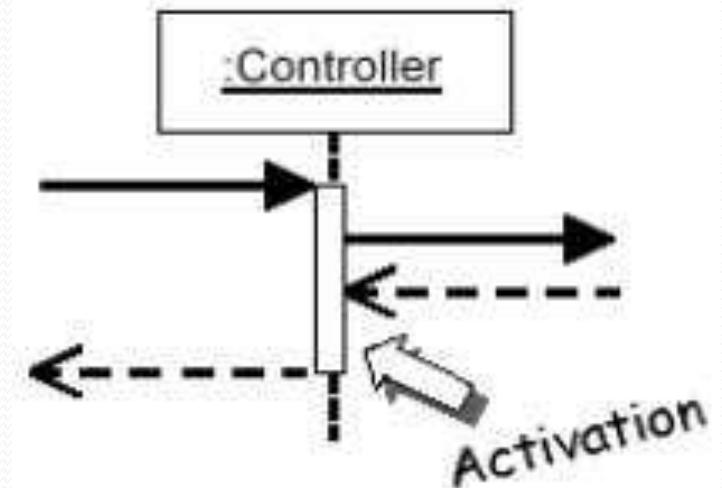
- Các yếu tố của một biểu đồ trình tự
 - Các qui ước về **tên của các thể hiện đối tượng**

Cú pháp	Giải thích	TênĐốiTượng	TênĐốiTượng: Lớp	:Lớp	<<StereoType>> TênĐốiTượng
<u>o</u>	Một đối tượng tên o.				
<u>o:C</u>	Đối tượng o của lớp C.				
<u>:C</u>	Đối tượng vô danh của lớp C.				
<u>/R</u>	Đối tượng vô danh đóng vai trò R.				
<u>/R:C</u>	Đối tượng vô danh của lớp C đóng vai trò R.				
<u>o/R</u>	Đối tượng o đóng vai trò R.				
<u>o/R:C</u>	Đối tượng o của lớp C đóng vai trò R.				

Cú pháp tên của thể hiện

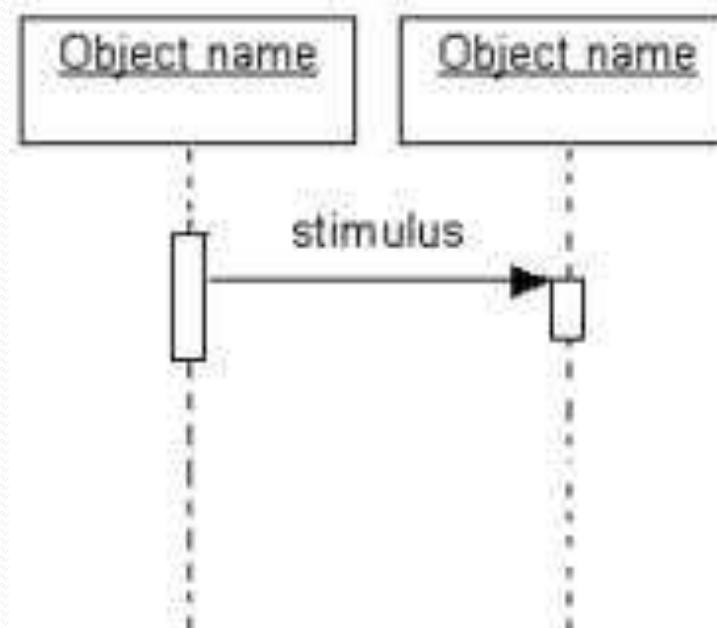
Lập mô hình động: Biểu đồ trình tự (Sequence Diagrams) (tt.)

- Các yếu tố của một biểu đồ trình tự
 - **Kích hoạt (Activation)**: khoảng thời gian một đối tượng thực hiện một thao tác (operation)
 - **Vùng kích hoạt-Activation** được biểu diễn bằng một *hình chữ nhật nhỏ đặt trên đường đời của đối tượng*



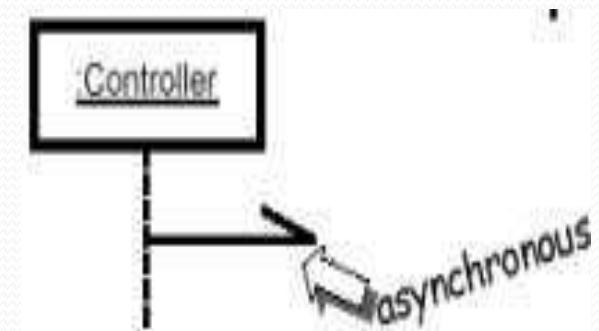
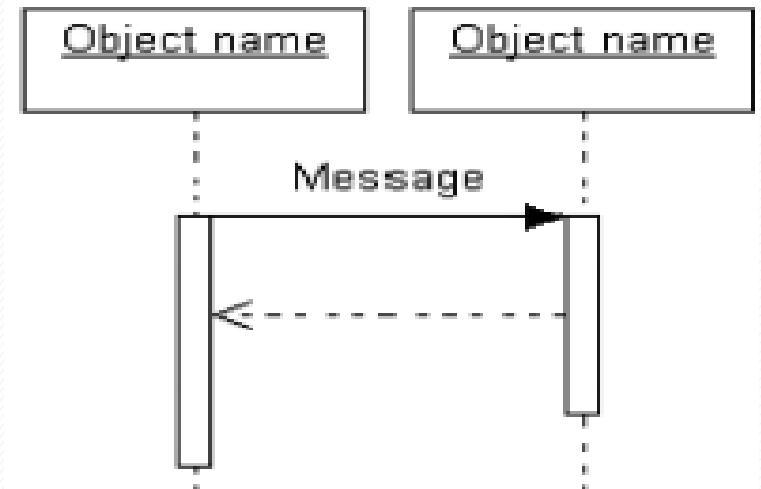
Lập mô hình động: Biểu đồ trình tự (Sequence Diagrams) (tt.)

- Các yếu tố của một biểu đồ trình tự
 - **Thông điệp (Messages):** phương tiện mà các đối tượng giao tiếp với nhau (nhằm yêu cầu thực hiện một hành động, xử lý).
 - Được mô tả bằng **một mũi tên** từ đối tượng gửi đến đối tượng nhận



Lập mô hình động: Biểu đồ trình tự (Sequence Diagrams) (tt.)

- **Thông điệp đồng bộ (Synchronous message)**: một loại thông điệp nơi mà người gọi phải chờ đợi tương nhận hoàn thành việc thực hiện thao tác được gọi trước khi chính nó có thể tiếp tục thực hiện.
- **Thông điệp đơn giản (Simple message)**: một thông điệp chuyển kiểm soát từ người gửi đến người nhận mà không mô tả chi tiết về việc giao tiếp.
- **Thông điệp không đồng bộ (Asynchronous message)**: một thông điệp nơi mà người gửi không phải đợi người nhận để xử lý thông điệp.



Lập mô hình động: Biểu đồ trình tự (Sequence Diagrams) (tt.)

• **Thiết kế một ca sử dụng với một biểu đồ trình tự**

Vẽ một biểu đồ trình tự cho một thể hiện của ca sử dụng Đăng ký lớp học (Class registration), một biểu đồ trong đó khóa học có các điều kiện tiên quyết. Một mô tả về kịch bản này được cung cấp dưới đây.

1. Nhân viên Ghi danh (Registration Clerk) mở cửa sổ đăng ký và nhập thông tin đăng ký (sinh viên và lớp học).
2. Kiểm tra xem lớp học có mở hay không.
3. Nếu lớp học mở, kiểm tra xem khóa học có bất kỳ môn học tiên quyết nào không.
4. Nếu khóa học có môn học tiên quyết, sau đó kiểm tra xem sinh viên đã tham gia tất cả những môn học tiên quyết chưa.
5. Nếu học sinh đã tham gia những môn học tiên quyết thì đăng ký sinh viên vào lớp và tăng số lượng trong lớp học thêm một.
6. Kiểm tra xem lớp đã đầy chưa; nếu không/có lẽ, dừng làm gì (if not, do nothing).
7. Hiển thị đăng ký đã được xác nhận trong cửa sổ đăng ký.

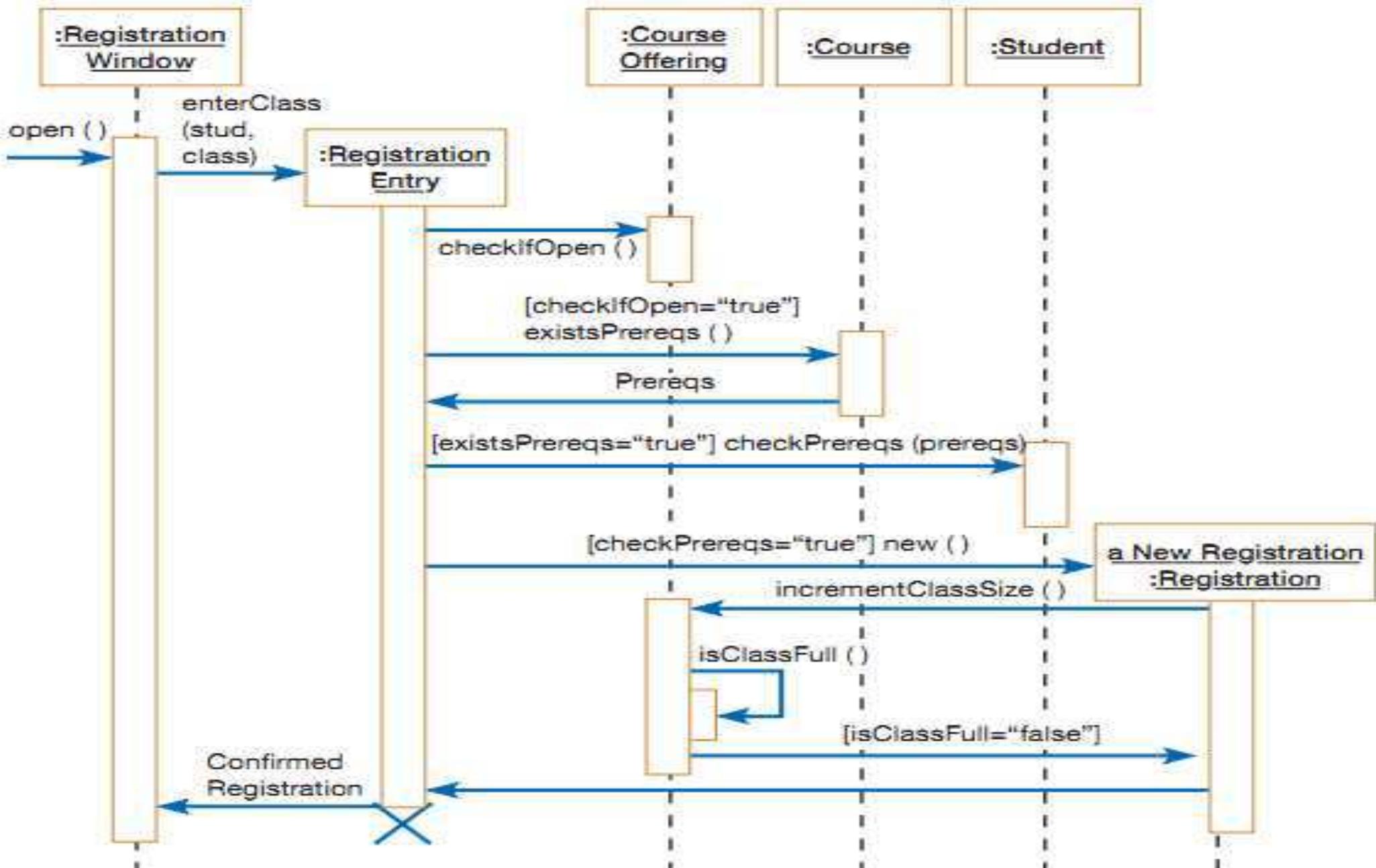


FIGURE 7-40
Sequence diagram for a class registration scenario with prerequisites

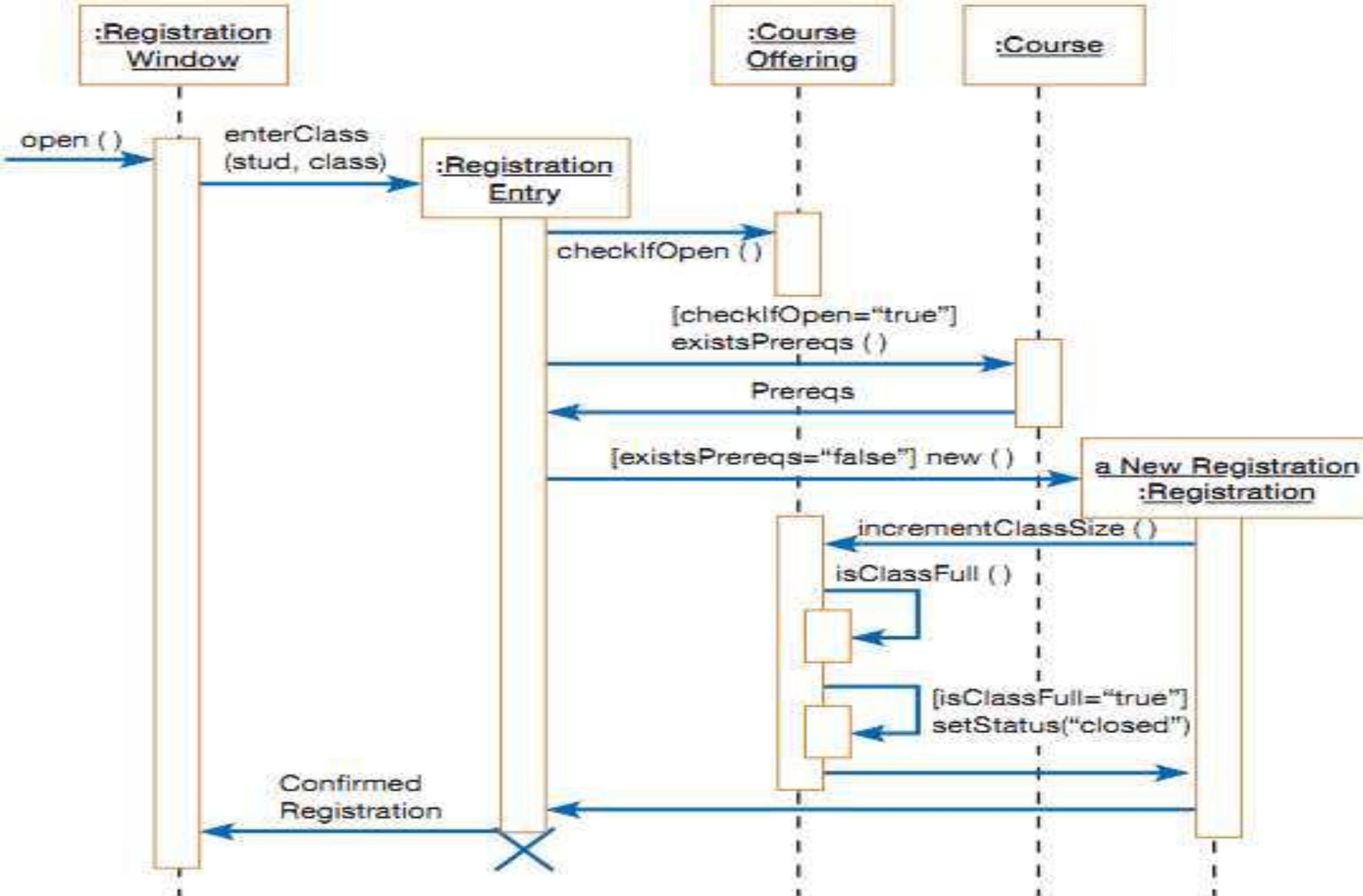


FIGURE 7-41
Sequence diagram for a class registration scenario without prerequisites

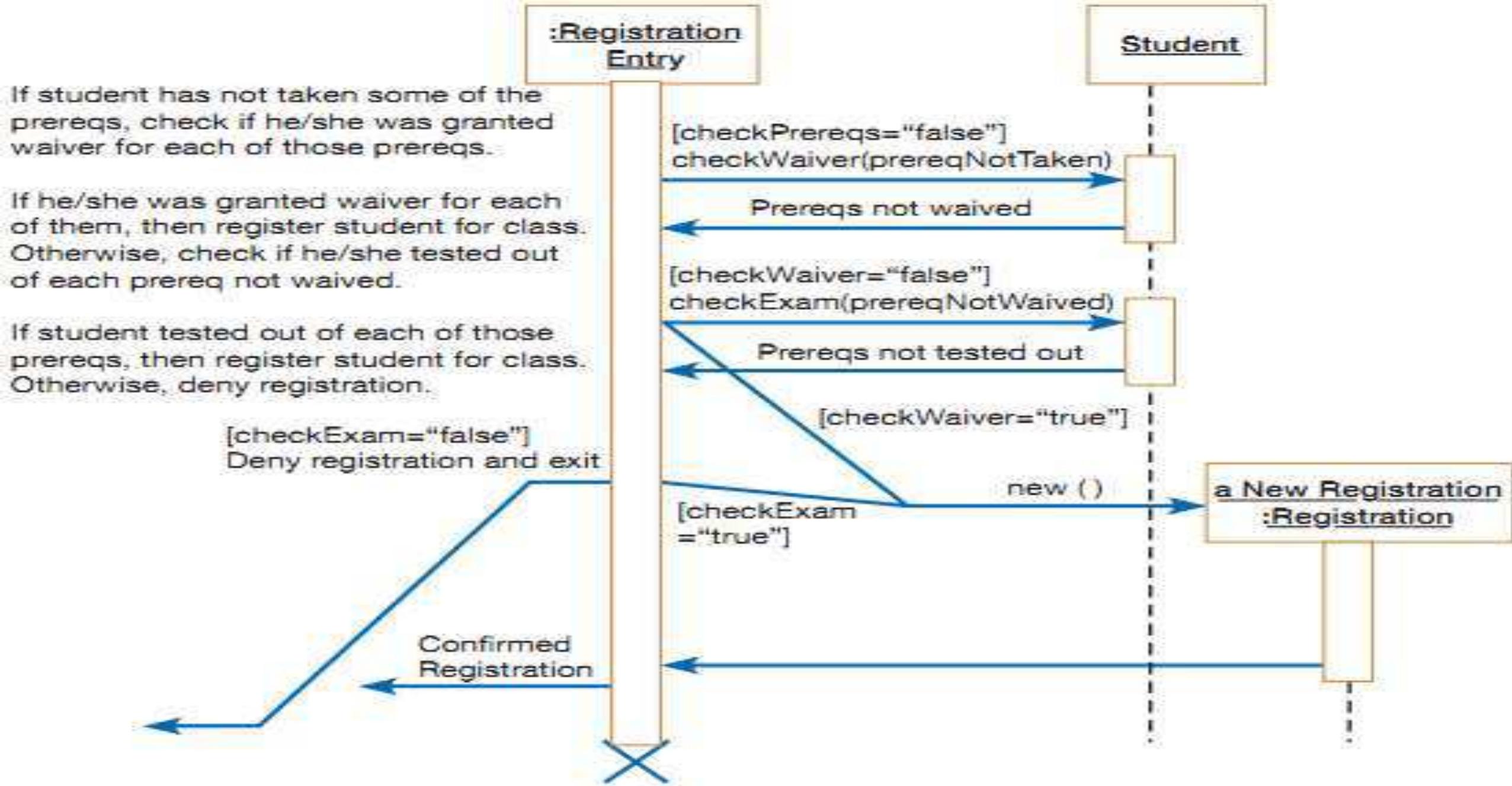


FIGURE 7-42

A generic sequence diagram for the Prereq courses not completed use case

Một biểu đồ trình tự cho Hoosier Burger

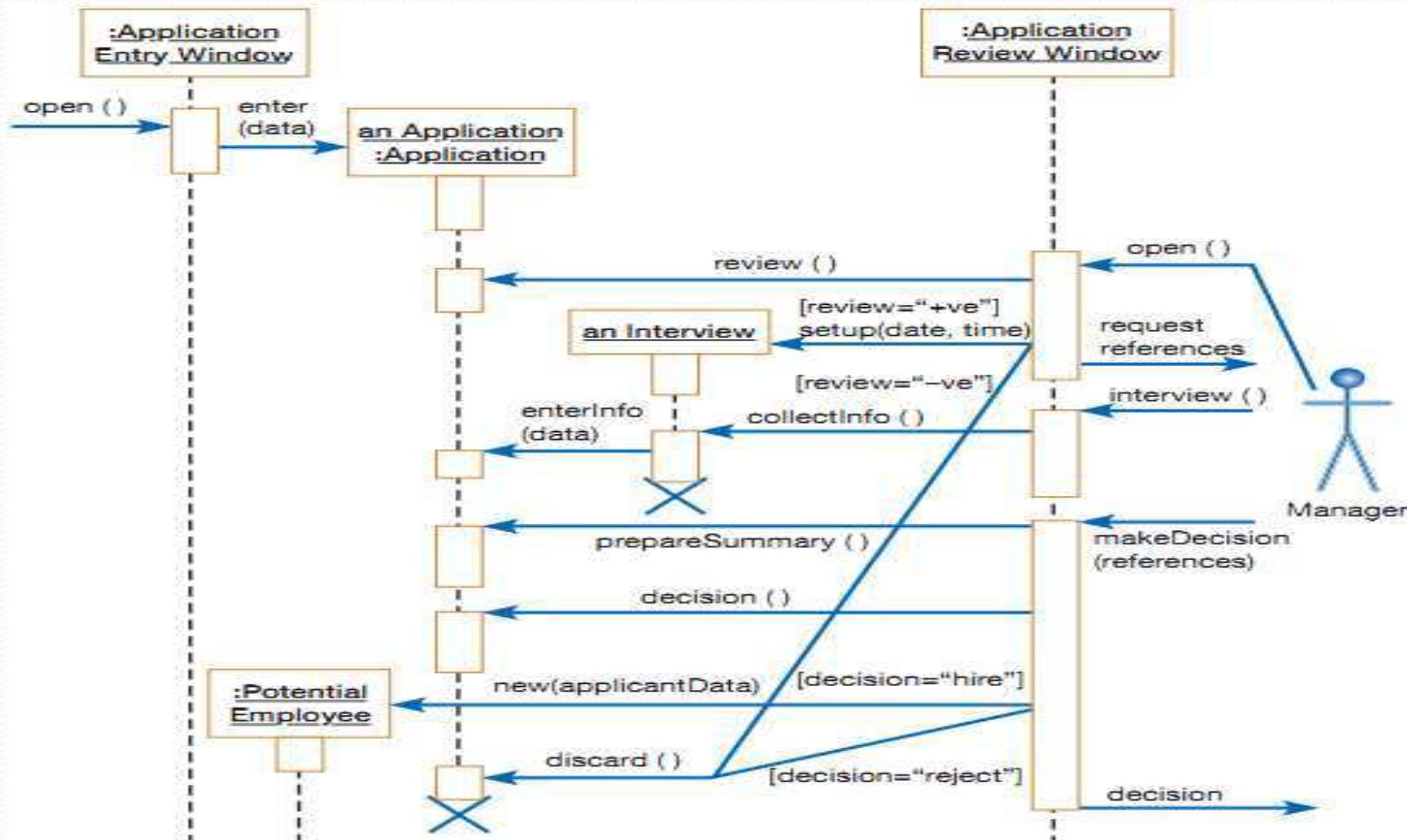
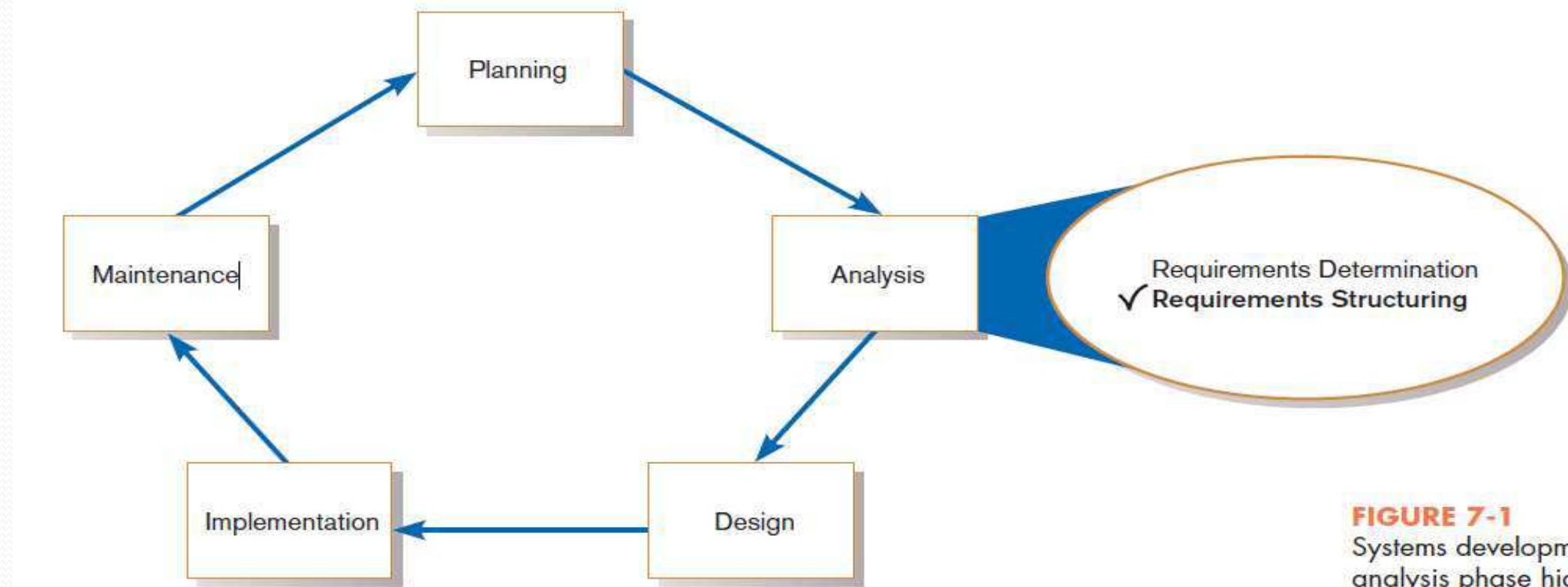


FIGURE 7-43
Sequence diagram for Hoosier Burger's
Hire employee use case

Phân tích các hệ thống

• 3.2 Cấu trúc các yêu cầu HT: Mô hình hóa dữ liệu



Lập mô hình quan niệm dữ liệu

- **Lập mô hình quan niệm dữ liệu:** một mô hình chi tiết nắm bắt cấu trúc tổng quan của dữ liệu trong một tổ chức
 - Xem xét cẩn thận nghiệp vụ của bất kỳ HQTCSDL nào hoặc xem xét việc thực hiện theo kế hoạch khác.

Quy trình Lập mô hình dữ liệu mức quan niệm

- Phát triển một mô hình dữ liệu cho HT hiện tại.
- Phát triển một mô hình dữ liệu mức quan niệm mới gồm tất cả các yêu cầu của HT mới.
- Trong giai đoạn thiết kế, mô hình quan niệm dữ liệu được chuyển sang thiết kế vật lý.
- Kho dự án liên kết tất cả các bước thiết kế và lập mô hình dữ liệu được thực hiện trong chu trình phát triển hệ thống (SDLC).

Các sản phẩm bàn giao và kết quả

- Sơ đồ thực thể - kết hợp (Entity-relationship diagram – ERD) hoặc sơ đồ lớp UML
 - Các thực thể (hoặc các lớp) – các danh mục dữ liệu
 - Các mối quan hệ - relationships (hoặc các mối quan hệ kết hợp - associations)
- Tập hợp các đầu vào của những đối tượng dữ liệu được lưu trữ trong từ điển kho dự án (repository project dictionary), hoặc phần mềm lập mô hình dữ liệu
 - Kho liên kết dữ liệu, tiến trình, và các mô hình logic của một HTTT.
 - Các yếu tố dữ liệu chứa trong DFD phải xuất hiện trong mô hình dữ liệu và ngược lại.
 - Mỗi kho dữ liệu trong một mô hình tiến trình phải liên quan đến các đối tượng nghiệp vụ được biểu diễn trong mô hình dữ liệu.

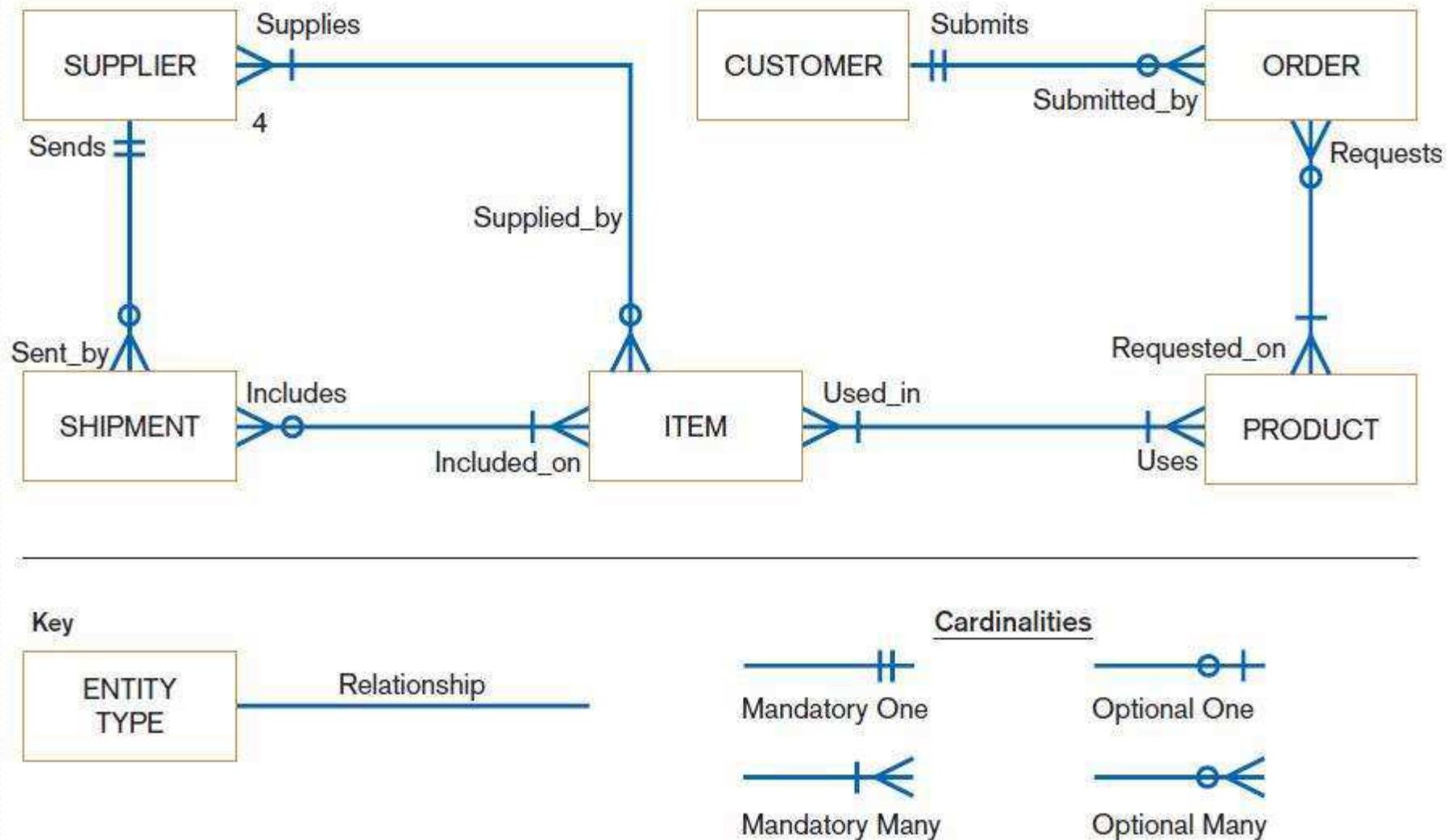


FIGURE 8-3
Sample conceptual data model

Tập hợp thông tin cho lập mô hình quan niệm dữ liệu

- Hai quan điểm
 - Top-down
 - Mô hình dữ liệu được bắt nguồn từ một sự hiểu biết sâu sắc về nghiệp vụ.
 - Bottom-up
 - Mô hình dữ liệu được bắt nguồn bằng việc xem xét những đặc tả và các tài liệu nghiệp vụ

Danh mục các Câu hỏi	Các câu hỏi để hỏi Những người dùng Hệ thống và các Nhà quản lý Nghiệp vụ
1. Các thực thể dữ liệu và mô tả của chúng	Các chủ đề/dối tượng của nghiệp vụ là gì? Những loại người, địa điểm, vật và nguyên liệu được sử dụng hoặc tương tác trong nghiệp vụ này về dữ liệu nào phải được duy trì? Bao nhiêu cá thể của mỗi đối tượng có thể tồn tại?
2. Khó khăn viên	Những đặc điểm duy nhất nào phân biệt từng đối tượng với các đối tượng khác cùng loại? Có thể có bất kỳ tính năng phân biệt nào như vậy thay đổi theo thời gian hoặc vĩnh viễn không? Liệu đặc tính này của một đối tượng có thể bị thiếu cho dù chúng ta biết đối tượng tồn tại hay không?
3. Các thuộc tính và các khóa phụ	Đặc điểm nào mô tả từng đối tượng? Các đối tượng được tham chiếu, lựa chọn, đủ điều kiện, được sắp xếp và phân loại dựa trên cơ sở nào? Chúng ta phải biết gì về mỗi đối tượng để điều hành kinh doanh?
4. Những kiểm soát an ninh và sự hiểu biết về người thực sự biết ý nghĩa của dữ liệu	Làm thế nào để bạn sử dụng những dữ liệu này? Nghĩa là, với nguồn dữ liệu cho tổ chức, bạn có tham chiếu các dữ liệu, bạn có sửa đổi chúng, và bạn có tiêu hủy chúng không? Ai không được phép sử dụng những dữ liệu này? Ai chịu trách nhiệm thiết lập các giá trị hợp pháp cho những dữ liệu này?
5. Bản số và các chiều thời gian của dữ liệu	Trong khoảng thời gian nào bạn quan tâm đến những dữ liệu này? Bạn có cần các xu hướng lịch sử, các giá trị "chụp nhanh" hiện tại và/hoặc các ước tính hoặc dự báo? Nếu một đặc điểm của một đối tượng thay đổi theo thời gian, bạn có phải biết những giá trị lỗi thời không?
6. Mối quan hệ và bản số và các bậc của chúng	Những sự kiện nào xảy ra mà ngũ ý kết hợp giữa các đối tượng khác nhau? Những giao dịch hoặc hoạt động tự nhiên nào của doanh nghiệp liên quan đến việc xử lý dữ liệu về một số đối tượng cùng loại hoặc khác nhau?
7. Các quy tắc về tính ràng buộc toàn vẹn, bản số lớn nhất và nhỏ nhất, chiều thời gian của dữ liệu	Mỗi hoạt động hoặc sự kiện có phải luôn luôn được xử lý theo cùng một cách, hay có tính hướng đặc biệt nào? Có thể nào một sự kiện xảy ra chỉ với một số đối tượng liên quan, hay phải có tất cả các đối tượng tham gia? Có thể nào có những liên kết giữa các đối tượng thay đổi theo thời gian không (ví dụ: các nhân viên thay đổi phòng ban)? Liệu các giá trị cho các đặc tính dữ liệu có bị giới hạn dưới bất kỳ hình thức nào không?

Một ứng dụng CSDL mẫu

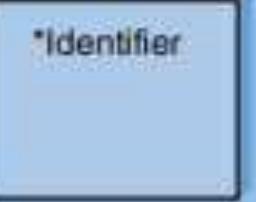
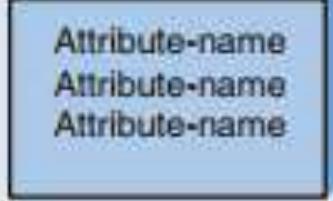
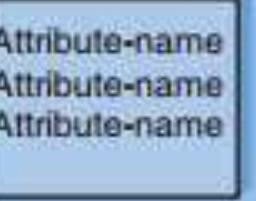
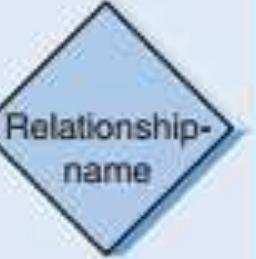
- CSDL Company quản lý các nhân viên-NV (employees), các phòng ban-PB (departments), và các dự án-DA (projects) của công ty.
 - Công ty được tổ chức thành các PB. Mỗi PB có 1 tên duy nhất, và 1 mã số phòng duy nhất và một NV cụ thể quản lý phòng. Quản lý ngày bắt đầu (start date) khi NV đó bắt đầu quản lý phòng. Một phòng có thể có vài vị trí (locations)
 - PB kiểm soát một số DA, mỗi DA có 1 tên duy nhất, 1 số DA duy nhất và 1 vị trí đơn.
 - Chúng ta lưu trữ tên mỗi NV (name), số an ninh xã hội (Social Security number), địa chỉ (address), lương (salary), giới tính (sex-gender), và ngày sinh (birthday). Mỗi NV được gán 1 phòng, nhưng có thể làm việc trên vài DA và các DA không nhất thiết được kiểm soát bởi cùng phòng. Quản lý số giờ mỗi tuần 1 NV làm việc trên mỗi DA. Cũng theo dõi người giám sát (supervisor) trực tiếp của mỗi NV (là NV khác)
 - Quản lý thân nhân (dependents) của mỗi NV cho mục đích bảo hiểm. Theo dõi tên, giới tính, ngày sinh và quan hệ của thân nhân với NV

Giới thiệu Lập mô hình Entity-Relationship

- **Mô hình dữ liệu Thực thể - Kết hợp (E-R model):** một mô tả chi tiết, logic các thực thể, các mối kết hợp và các yếu tố dữ liệu đối với một tổ chức hoặc một lĩnh vực nghiệp vụ
- **Sơ đồ Thực thể - kết hợp (E-R diagram):** một biểu diễn đồ họa của một mô hình dữ liệu E-R.

Giới thiệu Lập mô hình Entity-Relationship (tt)

- Mô hình E-R được thể hiện gồm:
 - Các **thực thể dữ liệu** (Data entities) trong môi trường nghiệp vụ.
 - Các **mối quan hệ** (Relationships) hoặc các **mối quan hệ kết hợp** (associations) giữa các thực thể.
 - Các **thuộc tính** (Attributes) hoặc các **tính chất** (properties) của các thực thể và các mối quan hệ của chúng.

	IDEF1X	Chen	Crow's Foot
<p>An ENTITY</p> <ul style="list-style-type: none"> ✓ is a person, place, or thing. ✓ has a singular name spelled in all capital letters. ✓ has an identifier. ✓ should contain more than one instance of data. 	<p>ENTITY-NAME</p> 	<p>ENTITY-NAME</p> 	<p>ENTITY-NAME</p> 
<p>An ATTRIBUTE</p> <ul style="list-style-type: none"> ✓ is a property of an entity. ✓ should be used by at least one business process. ✓ is broken down to its most useful level of detail. 	<p>ENTITY-NAME</p> 	<p>Attribute-name</p> 	<p>ENTITY-NAME</p> 
<p>A RELATIONSHIP</p> <ul style="list-style-type: none"> ✓ shows the association between two entities. ✓ has a parent entity and a child entity. ✓ is described with a verb phrase. ✓ has cardinality (1 : 1, 1 : N, or M : N). ✓ has modality (null, not null). ✓ is dependent or independent. 	<p>Relationship-name</p> 	<p>Relationship-name</p> 	<p>Relationship-name</p> 

Mô hình ER với các hệ thống kí hiệu:

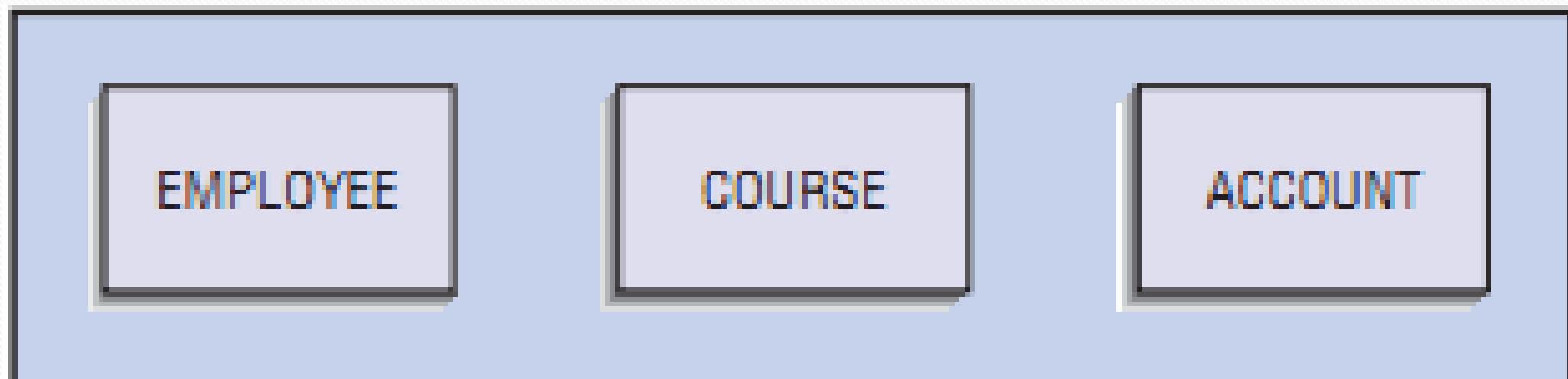
1. IDEF1X
2. Chen
3. Crow's Foot

Lập mô hình Entity-Relationship (E-R)

Những thuật ngữ chính

- **Thực thể (Entity)**

- Những gì tổ chức mong muốn để duy trì dữ liệu trong môi trường người dùng như một người, nơi chốn, đối tượng, sự kiện hoặc khái niệm
- Được biểu diễn bằng hình chữ nhật



Lập mô hình Entity-Relationship (E-R)

Những thuật ngữ chính

- **Tập thực thể (Entity Type)**

- Một tập hợp các thực thể chia sẻ những thuộc tính hoặc những đặc điểm chung.

- **Cá thể/Bản thể (Entity Instance)**

- Một thể hiện đơn (single occurrence) của một tập thực thể.

Entity Types

Strong

Weak

Associative

Attributes

ENTITY NAME

Identifier

Partial identifier

Optional

[Derived]

{Multivalued}

Composite(, ,)

(kí hiệu Crow's foot)

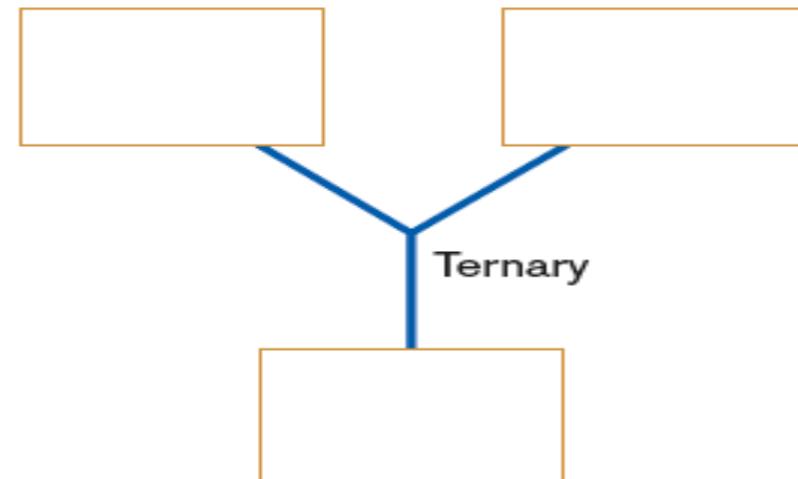
Relationship Degrees



Unary



Binary



Ternary

Relationship Cardinality



Mandatory One



Mandatory Many



Optional One



Optional Many

FIGURE 8-5 Basic E-R notation

Đặt tên và Định nghĩa Tập Thực thể (Entity Types)

- Tên một tập thực thể nên:
 - *Một danh từ số ít.*
 - Vd: CUSTOMER, STUDENT, AUTOMOBILE
 - *Chi tiết và cụ thể với tổ chức.*
 - Vd: PURCHASE ORDER, CUSTOMER ORDER
 - *Ngắn gọn và rõ ràng.*
 - Vd: trong 1 CSDL trường ĐH việc sử dụng tên REGISTRATION cho sự kiện đăng ký 1 SV đối với 1 lớp học
- *Tập thực thể sự kiện nên được đặt tên theo kết quả của sự kiện, không theo hoạt động hoặc quy trình của sự kiện.*

Đặt tên và Định nghĩa Tập Thực thể (Entity Types) (tt)

- Định nghĩa tập thực thể:
 - Gồm 1 phát biểu cái gì là (những) đặc tính duy nhất cho mỗi cá thể.
 - Làm rõ các cá thể gồm những gì và không gồm những gì trong tập thực thể.
 - Thường có một mô tả khi nào một cá thể của tập thực thể được tạo hoặc được xóa.
- Với một số tập thực thể, định nghĩa phải đặc tả cụ thể:
 - *Khi nào một cá thể phải thay đổi thành một cá thể của một tập thực thể khác.*
 - *Những giá trị nào được duy trì cho các cá thể.*

Lập mô hình Entity-Relationship (E-R) (tt)

Những thuật ngữ chính

- **Thuộc tính (Attribute)**

- Một thuộc tính hoặc đặc điểm của 1 thực thể được đặt tên
- Một đặc tính (property) hoặc đặc điểm của một thực thể có liên quan đến tổ chức (các mối quan hệ có thể cũng có thuộc tính)

Đặt tên và định nghĩa Thuộc tính

- Tên một thuộc tính là một danh từ và nên duy nhất.
 - Vd: Customer_ID, Age, Product_Minimum_Price
- Để có một tên thuộc tính duy nhất và rõ ràng, mỗi tên thuộc tính nên theo một định dạng tiêu chuẩn.
 - Vd: Student_GPA vs. GPA_of_Student
- *Những thuộc tính tương tự của những tập thực thể khác nhau nên sử dụng những tên gọi phân biệt.*
 - Vd: Faculty_Residence_City_Name và Student_Residence_City_Name

Đặt tên và định nghĩa Thuộc tính (tt)

- Định nghĩa thuộc tính:
 - *Cho biết thuộc tính là gì và có thể kèm theo tại sao nó quan trọng.*
 - Nên làm rõ nó gồm có gì và không gồm có gì trong giá trị của thuộc tính.
 - Chứa bất kỳ bí danh (*aliases*) hoặc tên gọi thay thế khác.
 - Cho biết nguồn các giá trị đối với thuộc tính.

Đặt tên và định nghĩa Thuộc tính (tt)

- Định nghĩa một thuộc tính nên chỉ ra:
 - *Liệu một giá trị cho thuộc tính là bắt buộc hay tùy chọn.*
 - *Liệu một giá trị cho thuộc tính có thể thay đổi không.*
 - Bất kỳ mối quan hệ của thuộc tính với các thuộc tính khác.

Lập mô hình Entity-Relationship (E-R) (tt)

Những thuật ngữ chính

- **Các khóa ứng viên và định danh** (*Candidate Keys and Identifiers*)

- Mỗi tập thực thể phải có một thuộc tính hoặc tập thuộc tính để phân biệt mỗi cá thể trong cùng tập thực thể.
- **Khóa ứng viên**
 - Thuộc tính (hoặc sự kết hợp các thuộc tính) mà xác định duy nhất mỗi cá thể của một tập thực thể.

Lập mô hình Entity-Relationship (E-R) (tt)

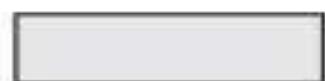
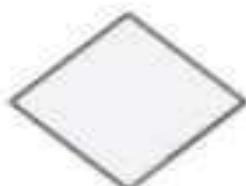
Những thuật ngữ chính

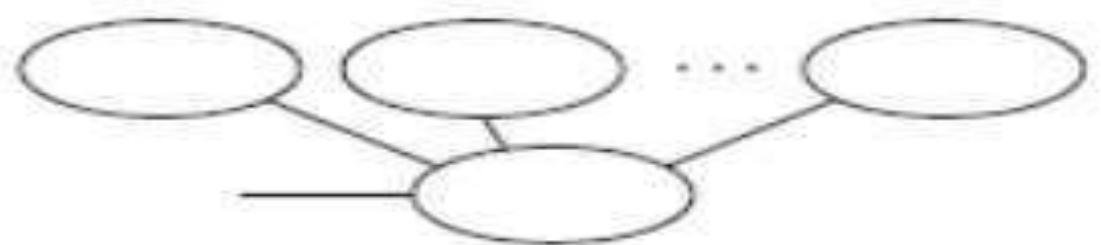
- **Định danh**

- Một khóa ứng viên đã được chọn như đặc tính xác định duy nhất cho một tập thực thể
- Các luật lựa chọn cho 1 định danh
 - 1. Chọn 1 khóa ứng viên mà sẽ không thay đổi giá trị của nó
 - 2. Chọn 1 khóa ứng viên mà sẽ không bao giờ là null
 - 3. Tránh sử dụng các khóa thông minh
 - 4. Xem xét thay thế các khóa lớn nhiều thuộc tính bằng các khóa một thuộc tính

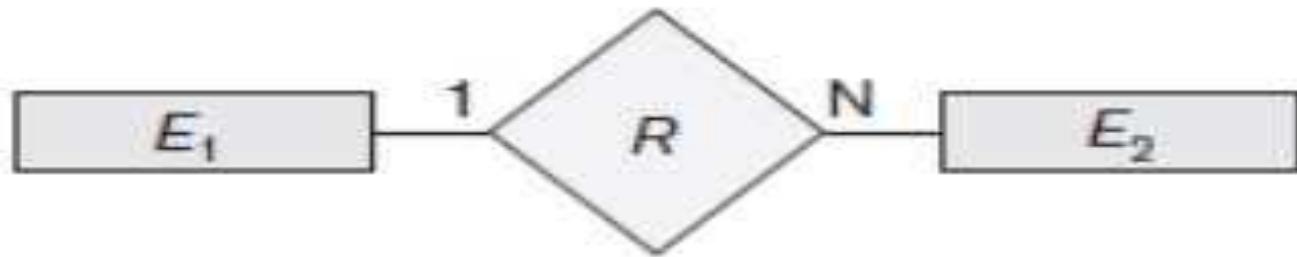
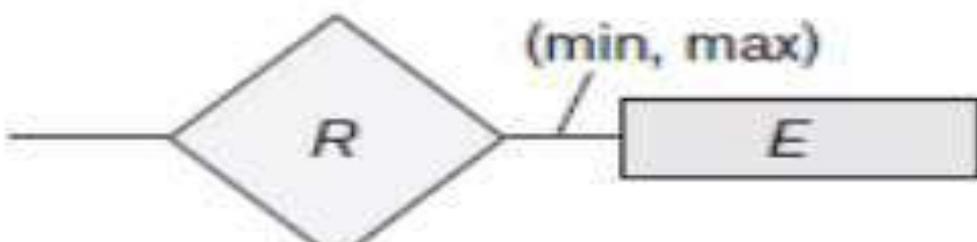
Symbol**Meaning****Figure 7.14**

Summary of the notation for ER diagrams.

**Entity****Weak Entity****Relationship****Identifying Relationship****Attribute****Key Attribute****Multivalued Attribute**

Symbol**Meaning****Figure 7.14**

Common symbols of the notation

Composite Attribute**Derived Attribute****Total Participation of E_2 in R** **Cardinality Ratio 1:N for $E_1:E_2$ in R** **Structural Constraint (min, max)
on Participation of E in R**

Thực thể mạnh (Strong Entity)

- Tồn tại độc lập với các tập thực thể khác (vd EMPLOYEE)
- Luôn có 1 đặc tính duy nhất (identifier-định danh) - một thuộc tính hoặc sự kết hợp các thuộc tính mà phân biệt duy nhất mỗi xuất hiện của đặc tính đó
- Là chủ nhân (owner) nếu có 1 tập thực thể yếu khác dựa vào nó (còn gọi là Identifying owner)

Weak Entity Types – Các loại thực thể yếu

- Không có các thuộc tính khóa của chính chúng
 - Được xác định bằng mối liên quan đến các thực thể cụ thể từ tập thực thể khác (vd DEPENDENT).
- **Identifying relationship** – Mỗi quan hệ định danh/Liên kết định danh
 - Liên quan 1 tập thực thể yếu đến chính chủ nhân của nó.
- Luôn luôn có 1 ràng buộc tham gia toàn phần (ít nhất 1, phụ thuộc sự tồn tại)
- Vd1, 1 thực thể DRIVER_LICENSE không thể tồn tại trừ phi nó liên quan đến 1 thực thể PERSON.

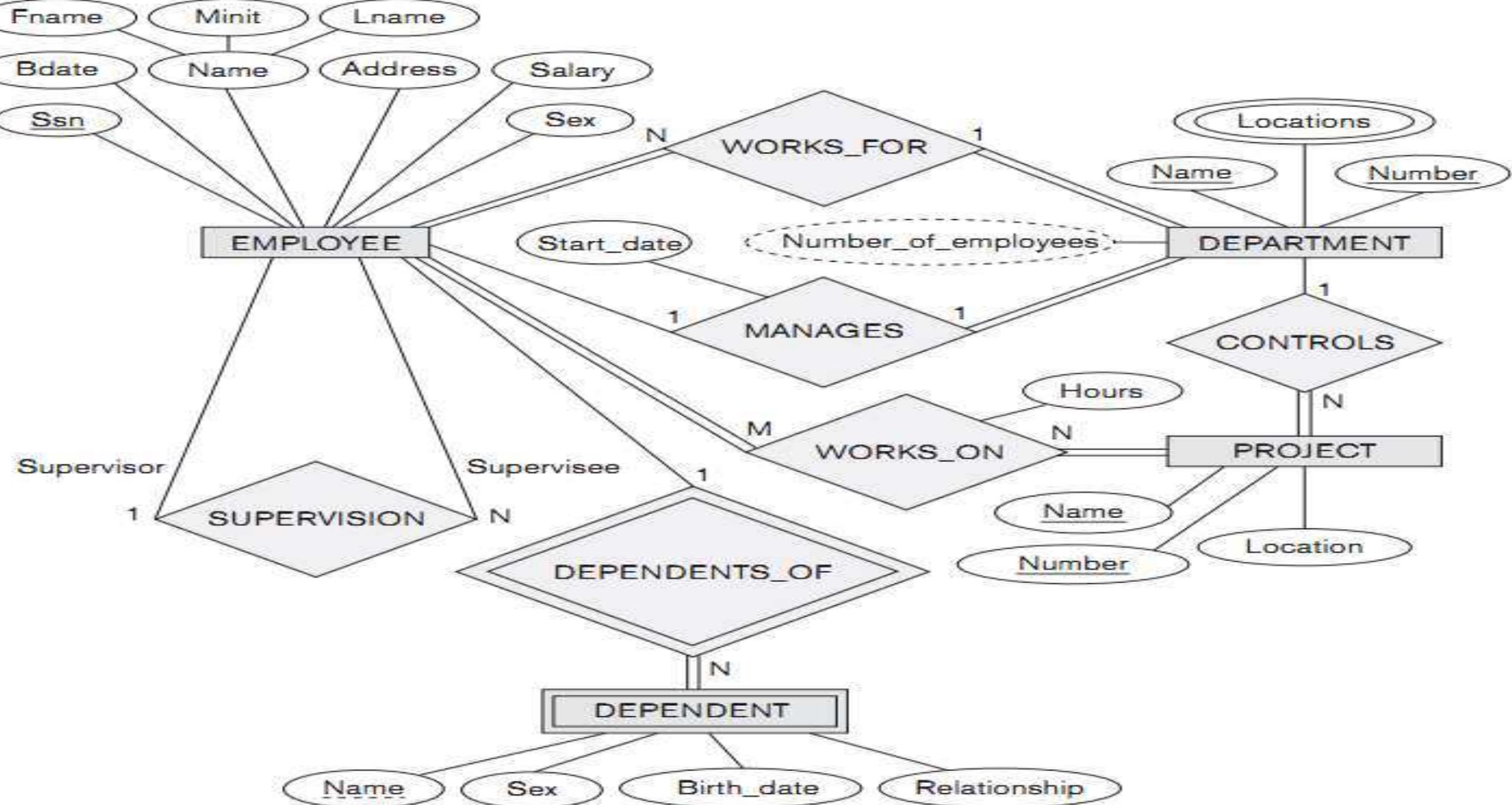
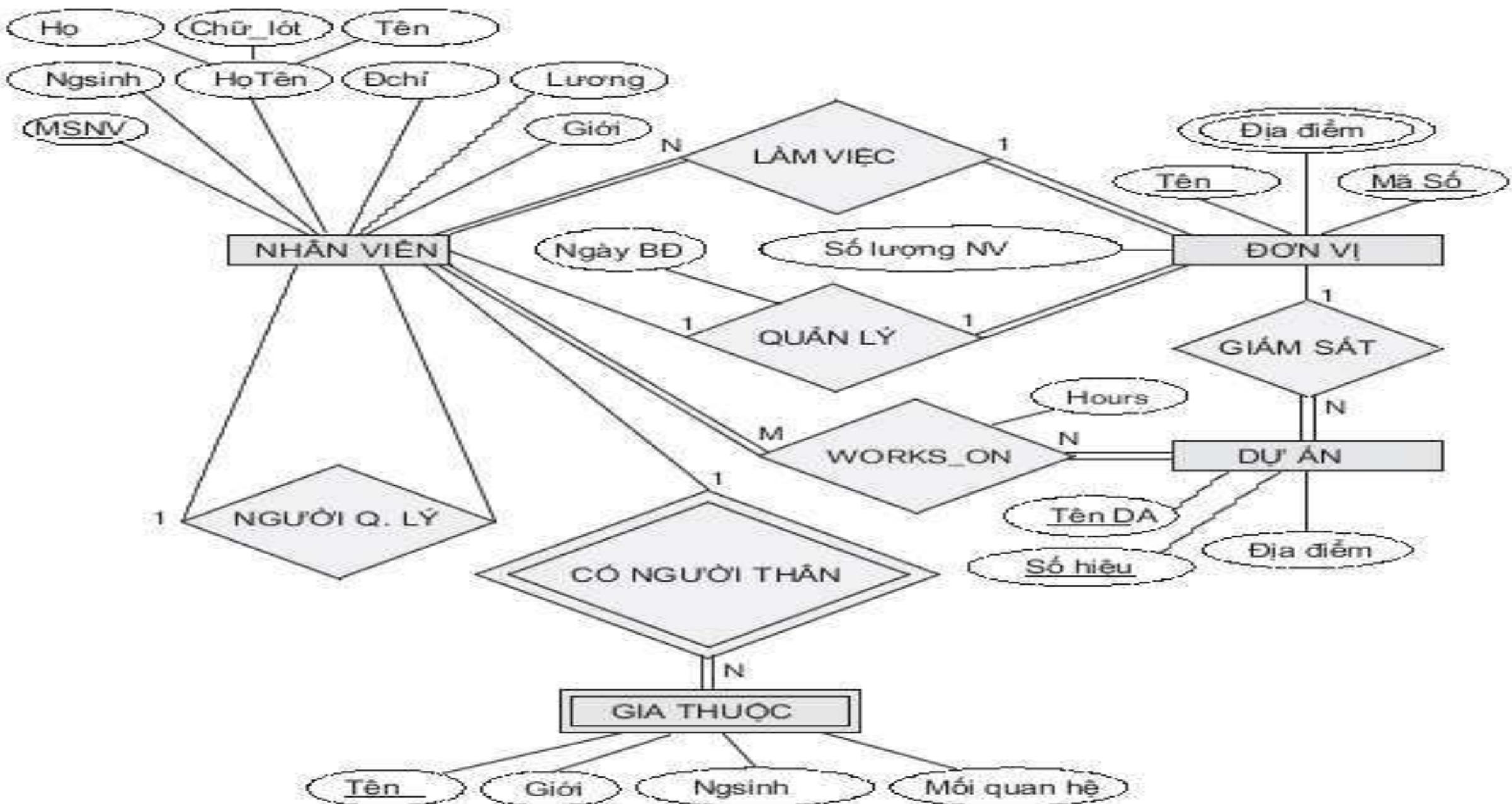


Figure 7.2

An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

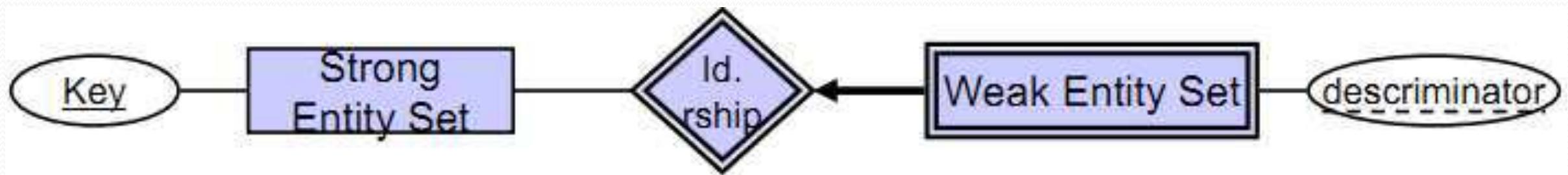
(kí hiệu Chen)



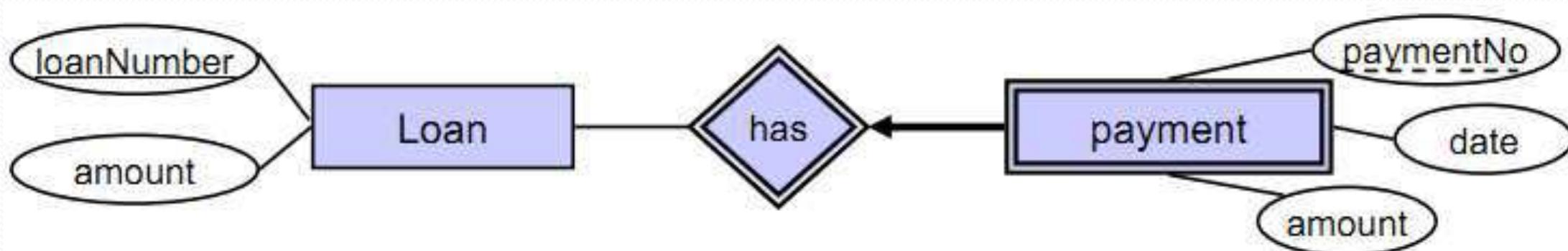
Weak Entity Types

(kí hiệu Chen)

- Kí hiệu: desriminator (or partial key - khóa thành phần)



- Ví dụ

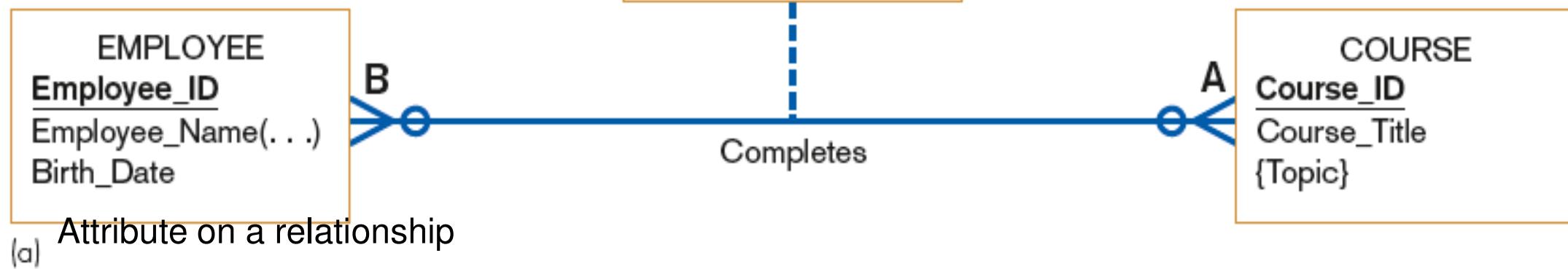


- paymentNo: là khóa thành phần của payment
- Khóa chính cho payment là (loanNumber, paymentNo)

Thực thể kết hợp (Associative Entity)

- Một tập thực thể mà liên kết các cá thể của một hoặc nhiều tập thực thể và chứa các thuộc tính riêng biệt đối với mỗi quan hệ giữa những cá thể đó.
- Người lập mô hình dữ liệu lựa chọn lập mô hình mỗi quan hệ như một tập thực thể.

FIGURE 8-15 An associative entity



Thực thể kết hợp (Associative Entity)

- Điều kiện để chuyển đổi một mối quan hệ thành một thực thể kết hợp (Associative Entity):
 - Là quan hệ thể hiện mối quan hệ M:N
 - Có ý nghĩa độc lập với những người dùng cuối, có thể được nhận diện với một định danh thuộc tính đơn.
 - Có một hoặc nhiều thuộc tính ngoài định danh
 - Tham gia độc lập vào một hoặc nhiều mối quan hệ của các thực thể liên quan trong mỗi quan hệ được liên kết

Lập mô hình Entity-Relationship (E-R) (tt)

- Các loại thuộc tính (Types of Attributes)
 - **Required attribute** (thuộc tính bắt buộc): một thuộc tính phải có giá trị cho mỗi cá thể
 - **Optional attribute** (thuộc tính tùy chọn): một thuộc tính có thể không có một giá trị cho mỗi cá thể.
 - **Composite vs simple (atomic) attributes** (thuộc tính phức hợp với thuộc tính đơn)
 - **Single-valued vs multivalued** attributes (thuộc tính đơn trị với thuộc tính đa trị)
 - **Stored vs derived** attributes (thuộc tính được lưu trữ với thuộc tính được suy dẫn)
 - **NULL values**
 - **Complex** attributes (thuộc tính phức tạp)

Lập mô hình Entity-Relationship (E-R) (tt)

- *Composite vs simple* (atomic) attributes - thuộc tính phức hợp vs thuộc tính đơn
 - **Composite attributes** có thể được chia thành những phần con nhỏ hơn để biểu diễn thêm nhiều thuộc tính cơ bản với những ý nghĩa độc lập.
 - Vd: thuộc tính địa chỉ (address) của thực thể NV (Employee) có thể được chia nhỏ thành Street_address, City, State, Zip với các giá trị '2311 Kirby', 'Houston', 'Texas', '77001'.
 - Composite attributes có thể hình thành một hệ thống cấp bậc. Giá trị của một composite attribute là dãy ghép các giá trị của các thuộc tính thành phần đơn của nó.
 - Các thuộc tính không thể chia nhỏ được gọi là các thuộc tính đơn, hoặc đơn trị (**simple (atomic) attributes**)

Lập mô hình Entity-Relationship (E-R) (tt)

- *Composite vs simple (atomic) attributes*

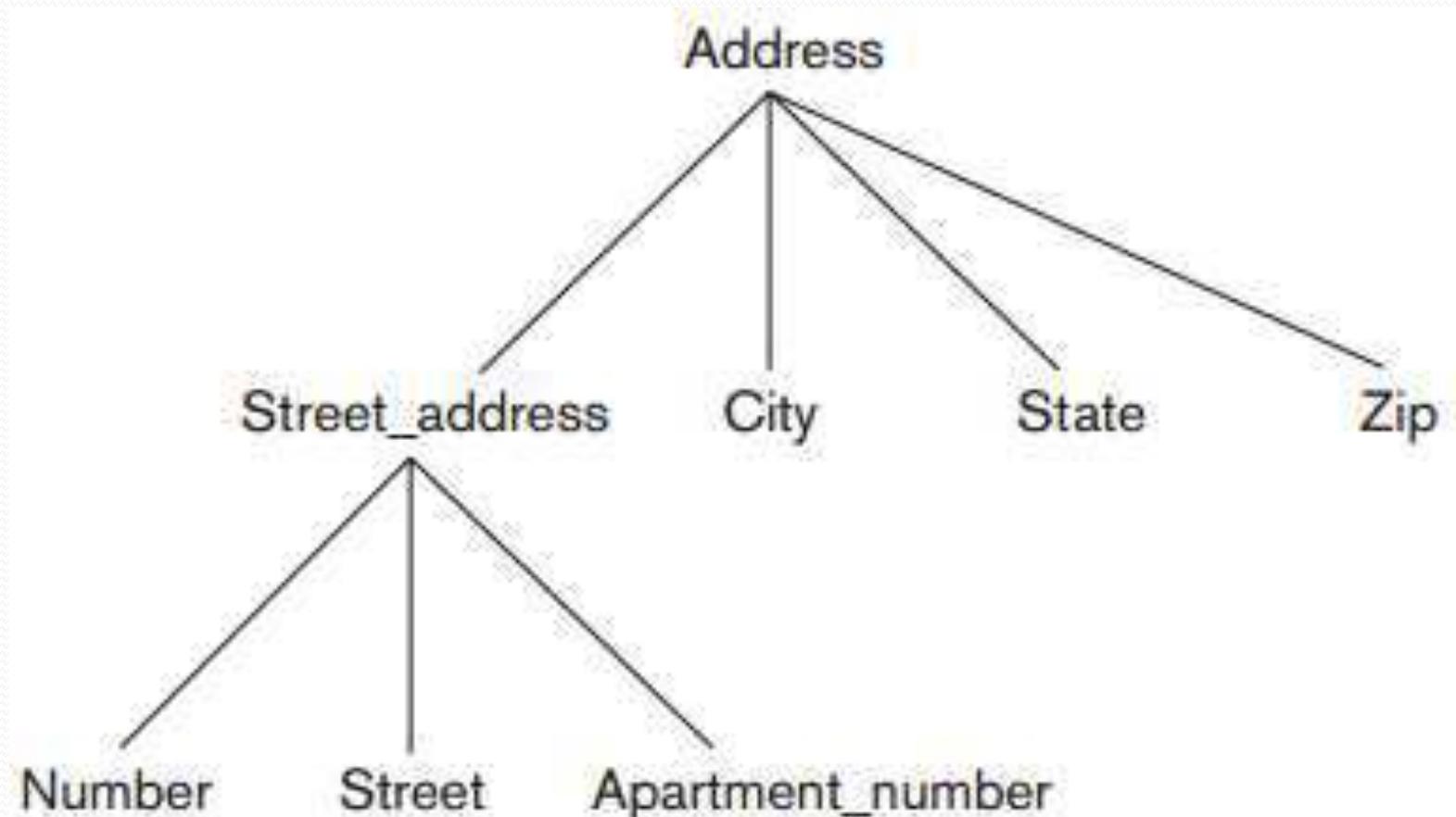


Figure 7.4

A hierarchy of composite attributes.

Lập mô hình Entity-Relationship (E-R) (tt)

- **Single-valued** vs **multivalued** attributes - Thuộc tính đơn trị vs thuộc tính đa trị
 - **Single valued** attributes. Là những thuộc tính có 1 giá trị đơn cho 1 thực thể cụ thể.
 - Vd, Ten_KhoaHoc (Course_Name)
 - **Multivalued** attributes. Một thuộc tính có thể nhận nhiều hơn một giá trị cho mỗi cá thể
 - Vd, 1 thuộc tính Màu sắc (Colors) cho 1 xe hơi, hoặc thuộc tính trình độ Đại học (College_degrees) của 1 người.

Lập mô hình Entity-Relationship (E-R) (tt)

- **Multivalued attributes.** Một thuộc tính có thể nhận nhiều hơn một giá trị cho mỗi cá thể.
- **Nhóm lặp (Repeating group):** một tập hợp 2 hoặc nhiều thuộc tính có quan hệ logic lặp lại cùng nhau. (có thể tách Nhóm lặp thành **Thực thể yếu**).

EMPLOYEE
Employee_ID
Employee_Name
Payroll_Address
{Skill}

(a) Multivalued attribute skill

EMPLOYEE
Employee_ID
{Dep_Name,
Dep_Age,
Dep_Relation}

(b) Repeating group of dependent data

EMPLOYEE
Employee_ID

DEPENDENT
Dep_Name
Dep_Age
Dep_Relation

(c) Weak entity for dependent data

FIGURE 8-8
Multivalued attributes
and repeating groups

Lập mô hình Entity-Relationship (E-R) (tt)

- Thuộc tính đa trị (Multivalued Attribute)

- Được biểu diễn trên sơ đồ E-R theo 2 cách:

- Cách 1: Một thuộc tính

(Kí hiệu Chen)

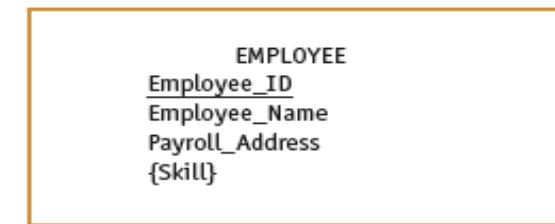
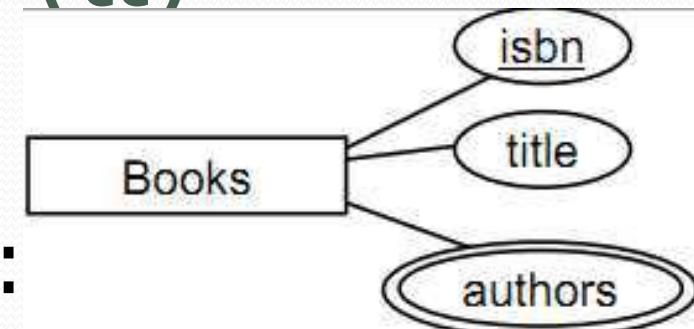


- Hình elip đường đôn

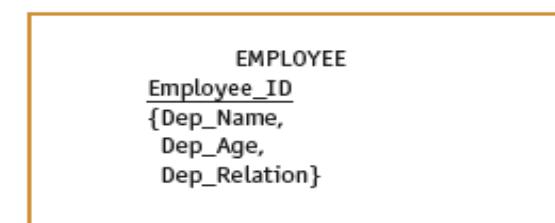
(Kí hiệu Crow's foot)

- Đặt trong dấu ngoặc nhọn

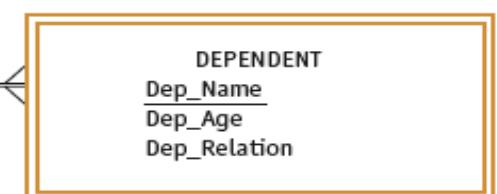
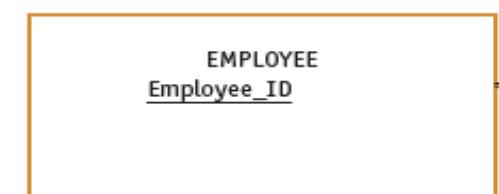
- Cách 2: Thực thể yếu



(a) Multivalued attribute skill



(b) Repeating group of dependent data



(c) Weak entity for dependent data

Lập mô hình Entity-Relationship (E-R) (tt)

- **Stored vs Derived Attributes**- thuộc tính được lưu trữ với thuộc tính được suy dẫn
 - Trong một số TH, 2 hay nhiều giá trị thuộc tính có liên quan
 - vd các thuộc tính **Age** và **Birth_date** của một người. Với 1 thực thể person cụ thể, giá trị **Age** có thể được xác định từ ngày hiện tại và giá trị **Birth_date** của person. Thuộc tính **Age** được gọi là 1 **derived attribute** và được suy dẫn từ thuộc tính **Birth_date** là 1 **stored attribute**
 - Một số giá trị thuộc tính có thể được suy dẫn từ những thực thể có liên quan;
 - vd, 1 thuộc tính **Number_of_employees** của 1 thực thể DEPARTMENT có thể được suy dẫn bằng cách đếm số nhân viên làm việc cho department đó.

Entities and Attributes (tt)

• **NULL Attributes**

- Trong một số TH, 1 thực thể cụ thể không thể có 1 giá trị có thể áp dụng cho 1 thuộc tính. Cho nên giá trị đặc biệt gọi là NULL được tạo ra.
 - Vd, thuộc tính **Apartment_number** của một địa chỉ (address) chỉ áp dụng cho các địa chỉ trong các tòa nhà chung cư và không cho các nơi cư trú khác như nhà đơn lẻ.
 - Tương tự 1 thuộc tính **College_degrees** chỉ áp dụng cho những người có bằng đại học.
 - NULL cũng có thể được dùng nếu ta không biết giá trị của một thuộc tính cho 1 thực thể cụ thể.

Entities and Attributes (tt)

- **Complex Attributes** - Thuộc tính phức tạp
 - Chú ý rằng, nhìn chung các thuộc tính composite và multivalued có thể được lồng nhau tùy ý.
 - Chúng ta có thể biểu diễn việc lồng nhau tùy ý bằng cách nhóm các thành phần của 1 thuộc tính composite giữa các dấu (), và tách biệt các thành phần bằng dấu phẩy, và bằng cách hiển thị các thuộc tính multivalued giữa các dấu {}.Những thuộc tính như vậy gọi là **complex attributes**

Entities and Attributes (tt)

- **Complex Attributes**

- Vd, nếu 1 person có thể có nhiều hơn 1 nơi cư trú và mỗi nơi cư trú có thể có 1 địa chỉ đơn và nhiều số điện thoại, một thuộc tính Address_phone cho 1 person có thể biểu diễn trong Hình dưới. Cả Phone và Address là các thuộc tính composite

(Address_phone((Phone(Area_code,Phone_number)),Address(Street_address
(Number,Street,Apartment_number),City,State,Zip)))

Figure 7.5
A complex attribute:
Address_phone.

Lập mô hình Entity-Relationship (E-R) (tt)

Những thuật ngữ chính

- **Mối quan hệ (Relationship)** *Đặt tên và Định nghĩa các Mối quan hệ*

- Một liên kết/mối quan hệ kết hợp giữa các cá thể của một hoặc nhiều tập thực thể có liên quan đến tổ chức.
- Tên một mối quan hệ là một cụm động từ
- Một định nghĩa mối quan hệ:
 - Giải thích hành động gì được thực hiện và có thể tại sao nó quan trọng.
 - Cho ví dụ để làm rõ hành động.

Lập mô hình Entity-Relationship (E-R) (tt)

- Kiểu/Tập quan hệ (**Relationship Type** hoặc **Relationship Set**)
 - Là một sự kết hợp giữa các thực thể trong các tập thực thể tham gia vào mỗi quan hệ đó.
 - Thể hiện của quan hệ (relationship instances): là mỗi liên kết/quan hệ kết hợp giữa các thể hiện của tập thực thể
- Kiểu quan hệ và thể hiện của quan hệ như là tập thực thể và thực thể
- Giữa 2 tập thực thể có thể tồn tại nhiều hơn một liên kết/mỗi quan hệ
 - Vd: mỗi liên kết MANAGES và WORKS_FOR là những mối lk riêng biệt giữa 2 tập thực thể EMPLOYEE và DEPARTMENT

(a)



(b) Employee :

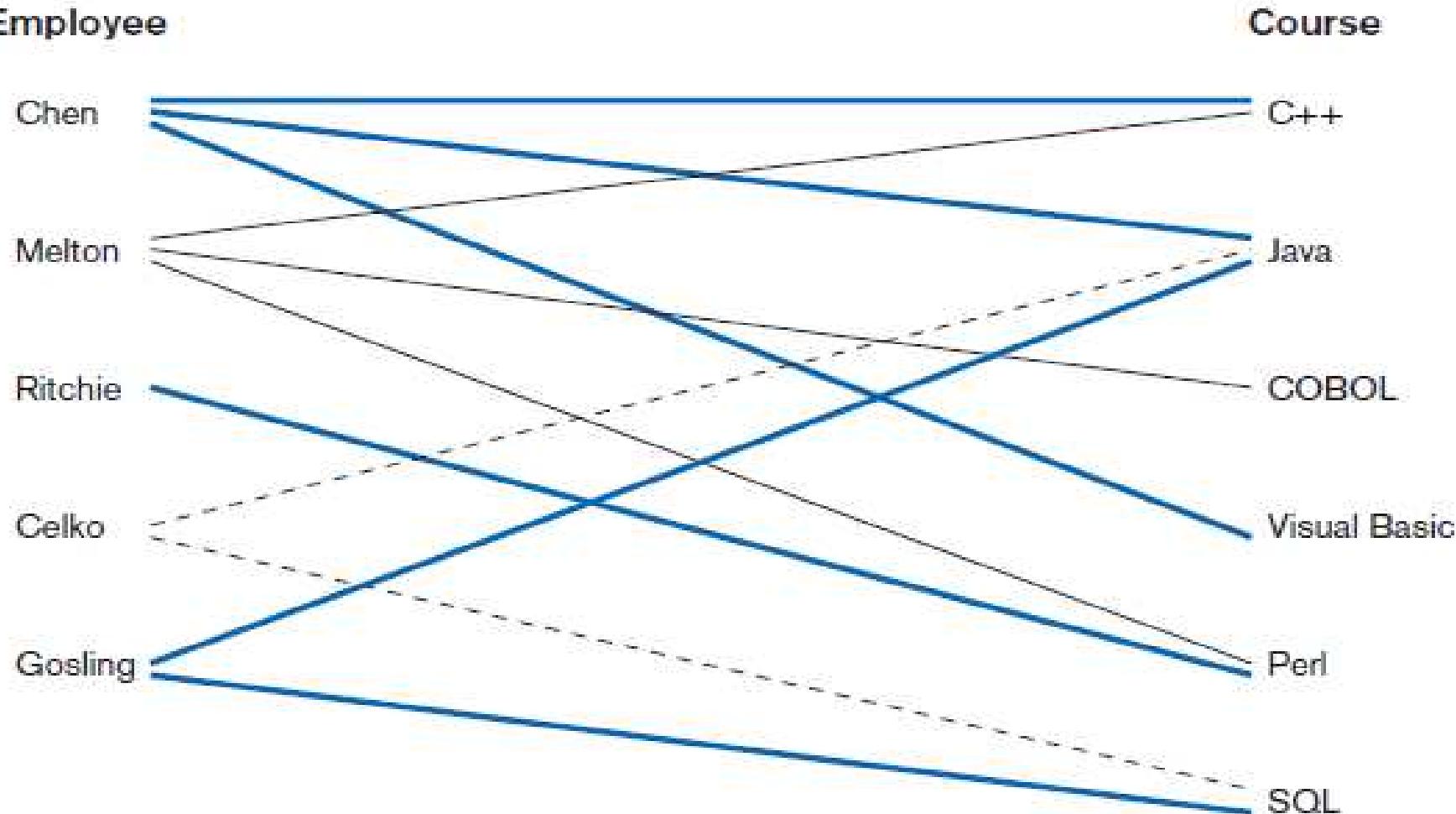


Figure 8-10
Relationship type and instances
(a) Relationship type (Completes)
(b) Relationship instances

Bậc của mối quan hệ

- **Bậc (Degree)**
 - Số tập thực thể tham gia vào một mối quan hệ
- Xét 3 trường hợp
 - **Bậc 1 (Unary)** *Unary Relationship (Recursive-Đệ qui)*
 - 1 quan hệ giữa các cá thể của một tập thực thể
 - **Bậc 2 (Binary)**
 - 1 quan hệ giữa các cá thể của 2 tập thực thể
 - **Bậc 3 (Ternary)**
 - 1 quan hệ đồng thời giữa các cá thể của 3 tập thực thể
 - Không giống như 3 quan hệ bậc 2

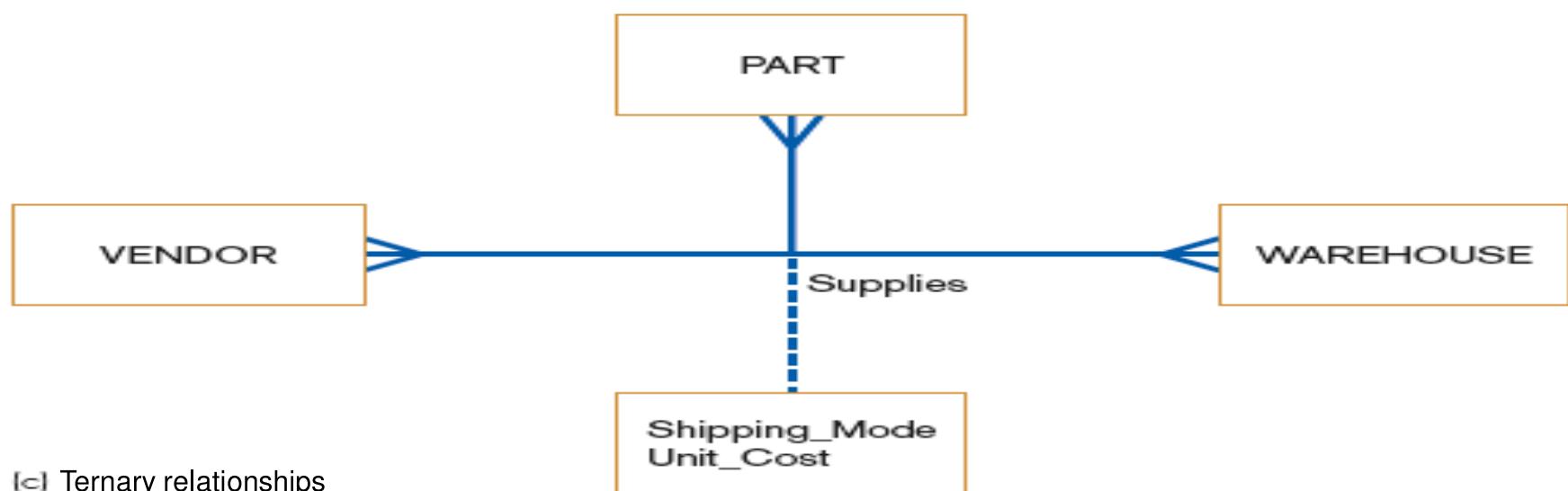
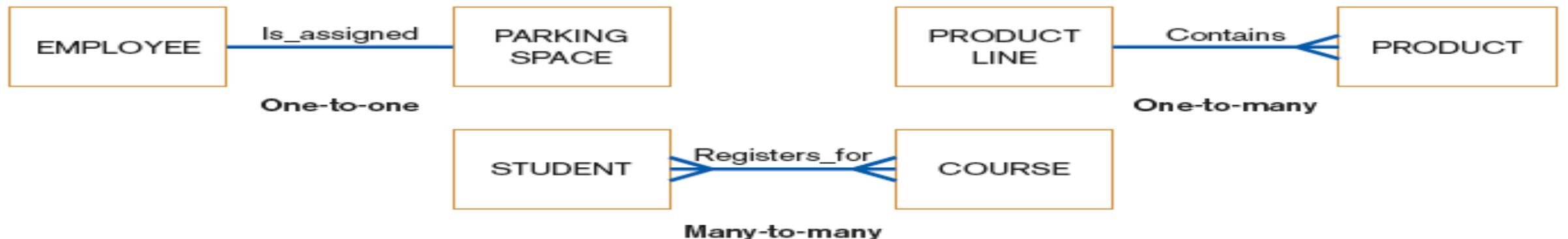


FIGURE 8-11
Examples of relationships
of different degrees

Mối quan hệ Bậc 3

(Ternary relationships)

- Chọn giữa các mối quan hệ bậc 2 và bậc 3

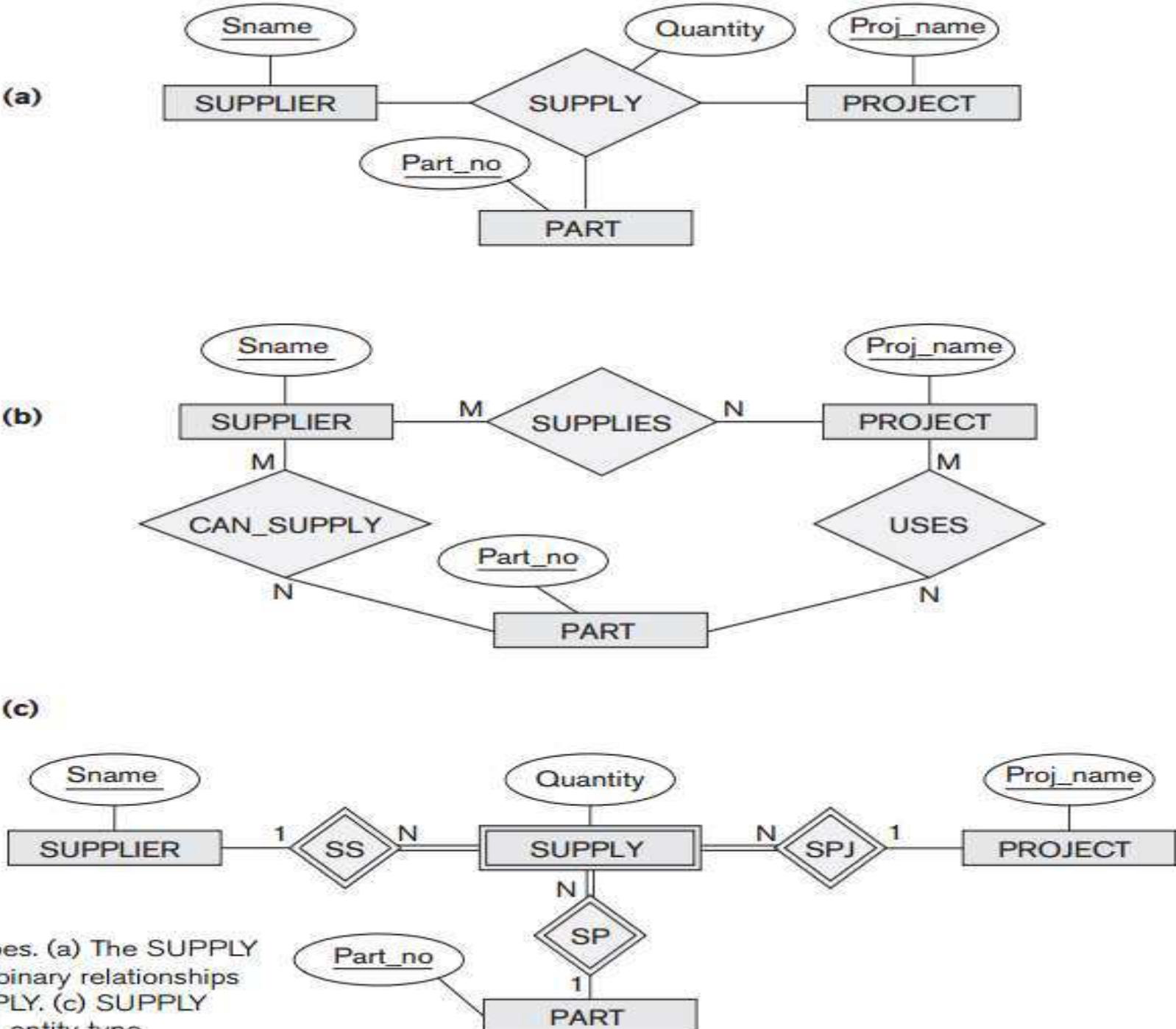
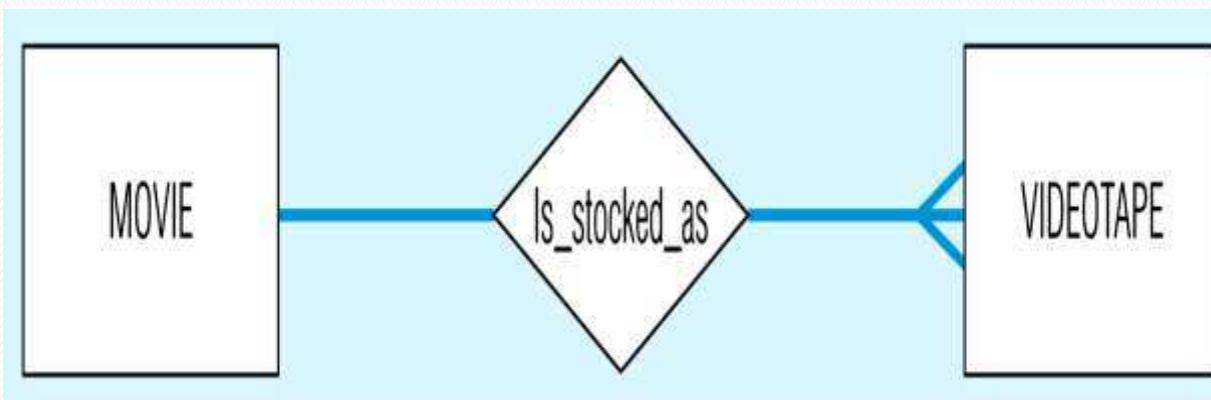


Figure 7.17
Ternary relationship types. (a) The SUPPLY relationship. (b) Three binary relationships not equivalent to SUPPLY. (c) SUPPLY represented as a weak entity type.

Các ràng buộc bản số (Cardinality constraints)

Hay ràng buộc số lượng phần tử

- **Bản số (Cardinality):** Số cá thể của thực thể B có thể hoặc phải được liên kết với mỗi cá thể của thực thể A
- Vd. 1 cửa hàng video chứa nhiều hơn 1 VIDEOTAPE cho mỗi MOVIE, đây là một mối quan hệ 'one-to-many' relationship



Các ràng buộc bản số (Cardinality constraints)

Hay ràng buộc số lượng phần tử trong kí hiệu Crow's foot

- **bản số nhỏ nhất** (Minimum Cardinality) của 1 mối quan hệ
 - Số cá thể nhỏ nhất của 1 thực thể B có thể được liên kết với mỗi cá thể của 1 thực thể A
- **bản số lớn nhất** (Maximum Cardinality) của 1 mối quan hệ
 - Số cá thể lớn nhất của 1 thực thể B có thể được liên kết với mỗi cá thể của thực thể A
- **bản số bắt buộc vs. bản số tùy chọn** (Mandatory vs. Optional Cardinalities)
 - Đặc tả liệu một cá thể phải tồn tại hoặc có thể không tồn tại trong mối quan hệ.
- **bản số 1 bắt buộc** (Mandatory one)
 - Mỗi quan hệ là 2 hướng (bi-directional), do đó cũng có kí hiệu bản số cạnh thực thể MOVIE
 - CHÚ Ý khi cả số lớn nhất và nhỏ nhất là 1 thì gọi là bản số 1 bắt buộc (vd, mỗi VIDEOTAPE của 1 MOVIE phải là 1 bản copy của chính xác 1 movie)

FIGURE 8-14 Examples of cardinality constraints

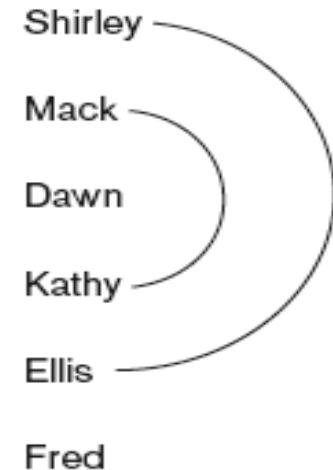
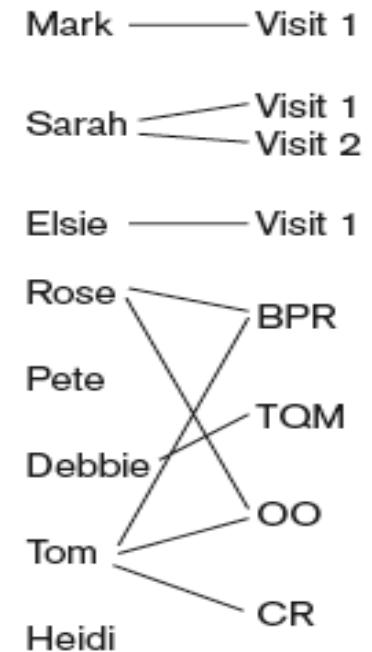
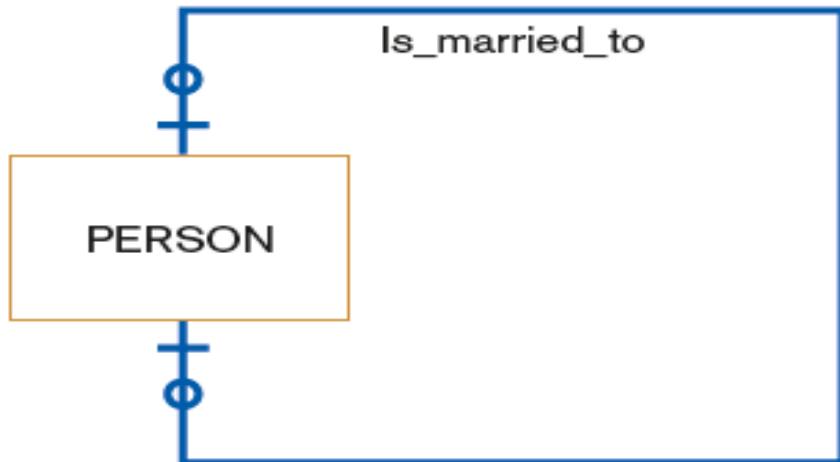
(a) Mandatory cardinalities



(b) One optional, one mandatory cardinality



(c) Optional cardinalities



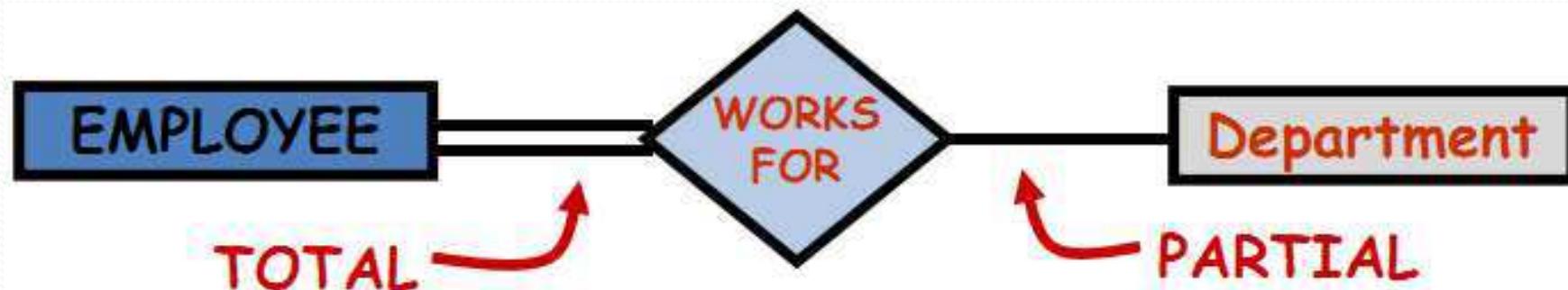
Ràng buộc lượng thành viên tham gia (*Participation constraint*) (cách biểu diễn bản số trong kí hiệu Chen)

- Ràng buộc này đặc tả số thể hiện mỗi quan hệ nhỏ nhất mà mỗi thực thể có thể tham gia vào, và thỉnh thoảng được gọi là ràng buộc số lượng thành viên nhỏ nhất
- Các loại: toàn phần (total) và riêng phần (partial)

Ràng buộc lượng thành viên tham gia (*Participation constraint*)

- Vd1

- Mỗi Employee phải làm việc cho một Department đơn
- 1 Department có thể có vài Employees làm việc cho nó
- Với lượng thành viên tham gia của các employee là Toàn phần (total)
 - Áp dụng cho Mỗi Employee
- Với lượng thành viên tham gia của các department là Riêng phần (partial)
 - DEPT mới thành lập không có employee nào



Một ứng dụng CSDL mẫu

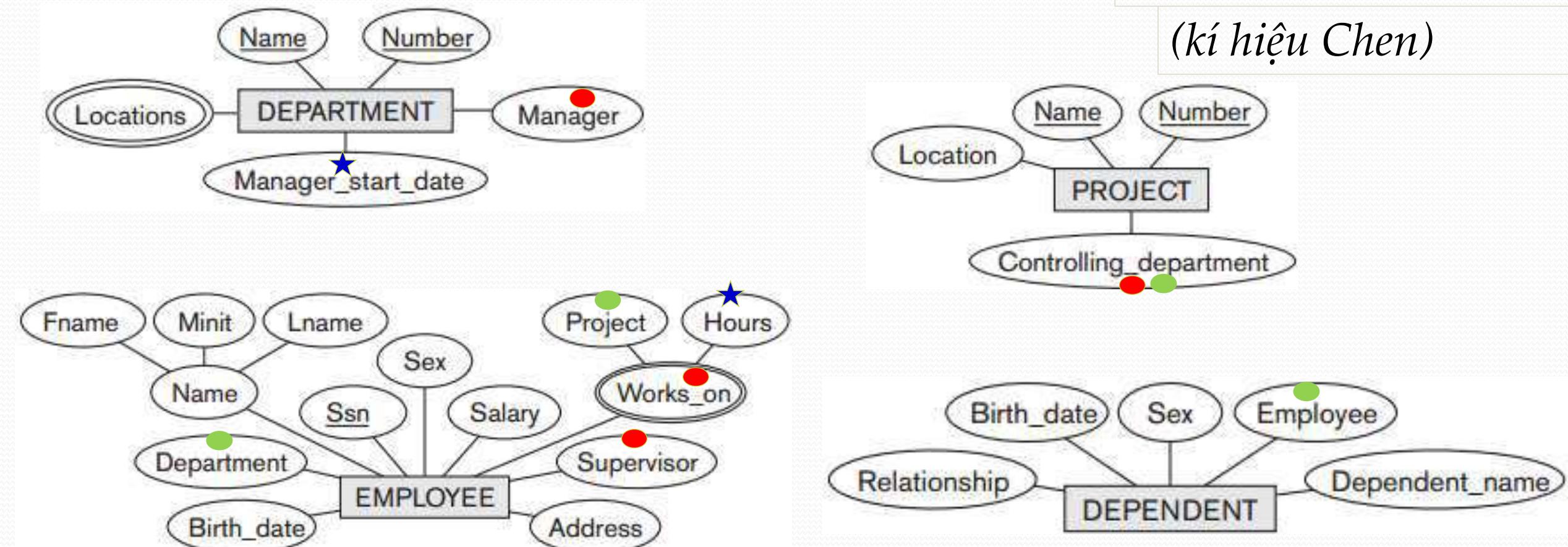
- CSDL Company quản lý các nhân viên-NV (employees), các phòng ban-PB (departments), và các dự án-DA (projects) của công ty.
 - Công ty được tổ chức thành các PB. Mỗi PB có 1 tên duy nhất, và 1 mã số phòng duy nhất và một NV cụ thể quản lý phòng. Quản lý *ngày bắt đầu (start date)* khi NV đó bắt đầu quản lý phòng. Một phòng có thể có vài vị trí (locations)
 - PB kiểm soát một số DA, mỗi DA có 1 tên duy nhất, 1 mã số DA duy nhất và 1 vị trí.
 - Chúng ta lưu trữ tên mỗi NV (name), số an ninh xã hội (Social Security number), địa chỉ (address), lương (salary), giới tính (sex-gender), và ngày sinh (birthday). Mỗi NV làm việc cho 1 phòng, nhưng có thể làm việc trên vài DA và các DA không nhất thiết được kiểm soát bởi cùng phòng. Quản lý số giờ mỗi tuần 1 NV làm việc trên mỗi DA. Cũng theo dõi người giám sát (supervisor) trực tiếp của mỗi NV (là NV khác)
 - Quản lý thân nhân (dependents) của mỗi NV cho mục đích bảo hiểm. Theo dõi tên, giới tính, ngày sinh và quan hệ của thân nhân với NV

Thiết kế Quan niệm ban đầu của CSDL COMPANY

Chúng ta có thể xác định 4 loại thực thể tương ứng 4 mục trong đặc tả

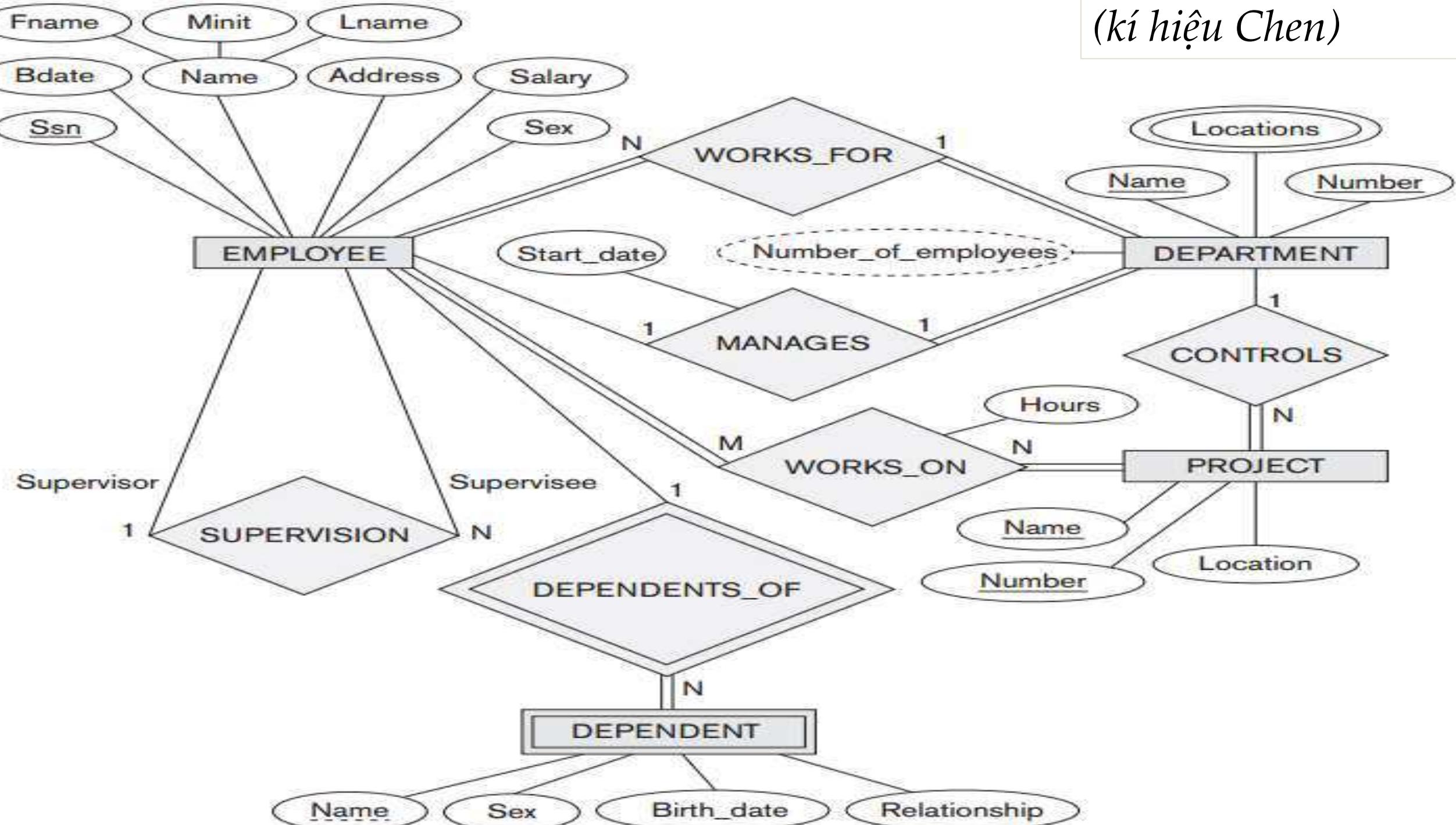
Figure 7.8
Preliminary design of entity types
for the COMPANY database.
Some of the shown attributes will
be refined into relationships.

(kí hiệu Chen)



Tập quan hệ (Relationship Types), Các vai trò, và các Ràng buộc về cấu trúc (Structural Constraints)

- Khi nào 1 thuộc tính của 1 tập thực thể để cập đến tập thực thể khác thì có mối quan hệ tồn tại. Trong Hình 7-8 của COMPANY database sample có vài *mối quan hệ ngầm/hàm ý (implicit relationships)* giữa các tập thực thể như:
 - thuộc tính **Manager** của DEPARTMENT để cập đến 1 NV quản lý department;
 - thuộc tính **Controlling_department** của PROJECT để cập đến department kiểm soát dự án;
 - Thuộc tính **Supervisor** của EMPLOYEE để cập đến nhân viên khác
 - Thuộc tính **Department** của EMPLOYEE để cập đến department nơi nhân viên làm việc,....



Thuộc tính của Kiểu mối quan hệ (*Relationship Types*)

- Vd để ghi lại *số giờ mỗi tuần mà 1 NV làm việc cho 1 DA* cụ thể, ta có thể đưa thuộc tính **Hours** vào kiểu mối quan hệ WORKS_ON (Hình 7.13); hoặc đưa *ngày mà 1 manager bắt đầu quản lý 1 department* qua 1 thuộc tính **Start_date** cho kiểu mối quan hệ MANAGES
- Các thuộc tính của các loại mối quan hệ **1:1** hoặc **1:N** có thể được chuyển vào một trong số các loại thực thể tham gia.
 - Vd1, thuộc tính **Start_date** cho quan hệ MANAGES có thể là 1 thuộc tính của EMPLOYEE hoặc DEPARTMENT

Thuộc tính của Kiểu mối quan hệ (*Relationship Types*)

- VỚI 1 LOẠI MỐI QUAN HỆ **1:N**
 - *Thuộc tính mối quan hệ có thể được chuyển vào chỉ trên tập thực thể bên N của quan hệ*
 - Vd2, Hình 7.9, nếu quan hệ WORKS_FOR cũng có 1 thuộc tính **Start_date** chỉ khi nào 1 NV bắt đầu làm việc cho 1 department, thì thuộc tính này có thể được đặt như 1 thuộc tính của EMPLOYEE.
- VỚI CÁC LOẠI MỐI QUAN HỆ **M:N**
 - *Một vài thuộc tính có thể được xác định bằng cách kết hợp của các thực thể tham gia trong 1 thể hiện quan hệ chứ không theo thực thể đơn.*
 - Phải được đặc tả như **các thuộc tính quan hệ**.
 - Vd3, thuộc tính **Hours** của quan hệ M:N WORKS_ON (Hình 7.13), số giờ mỗi tuần 1 NV hiện tại làm việc trên 1 DA được xác định bởi 1 sự kết hợp 2 thực thể NV-DA chứ không tách biệt theo thực thể nào

Điều chỉnh thiết kế ER cho csdl COMPANY

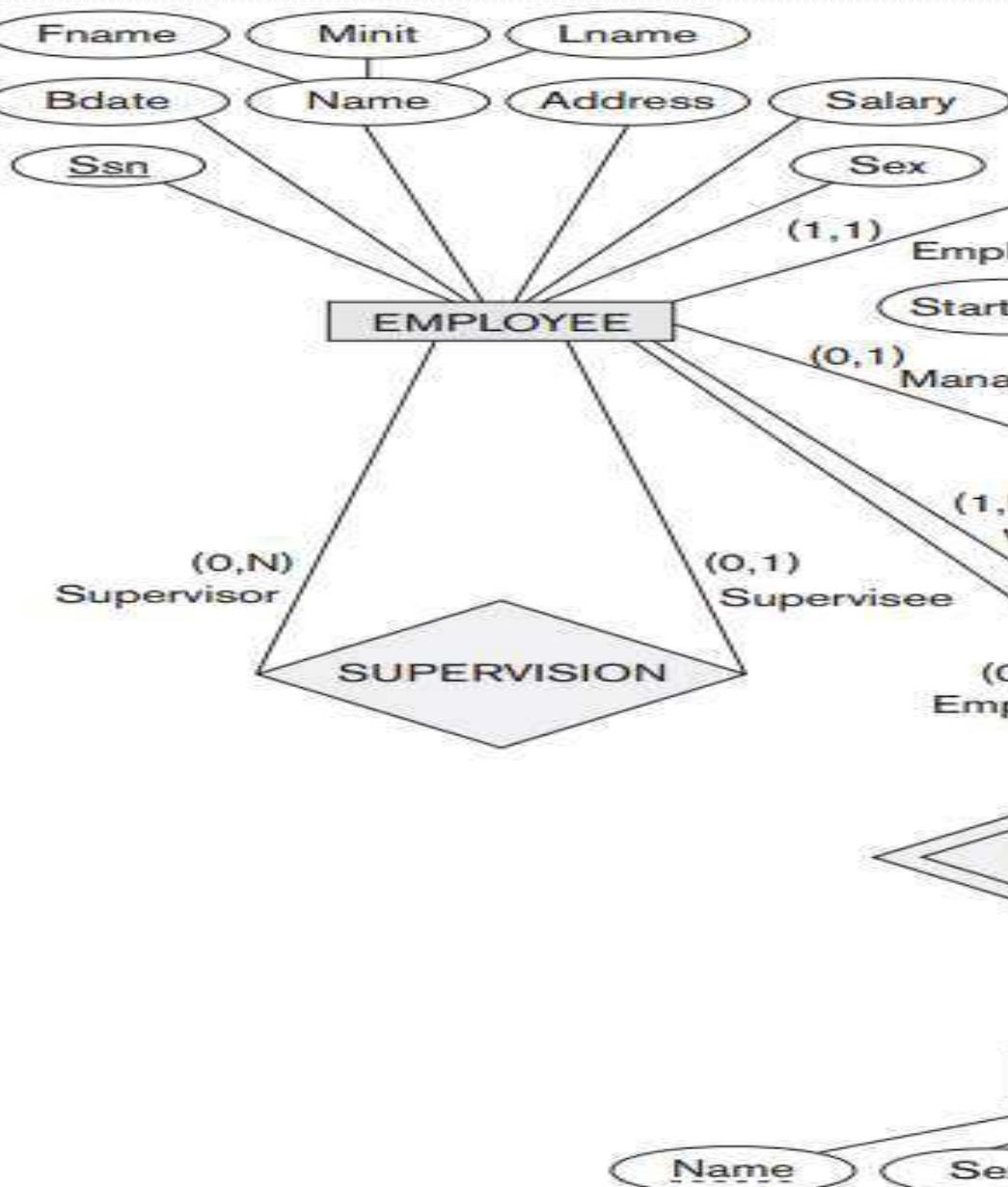
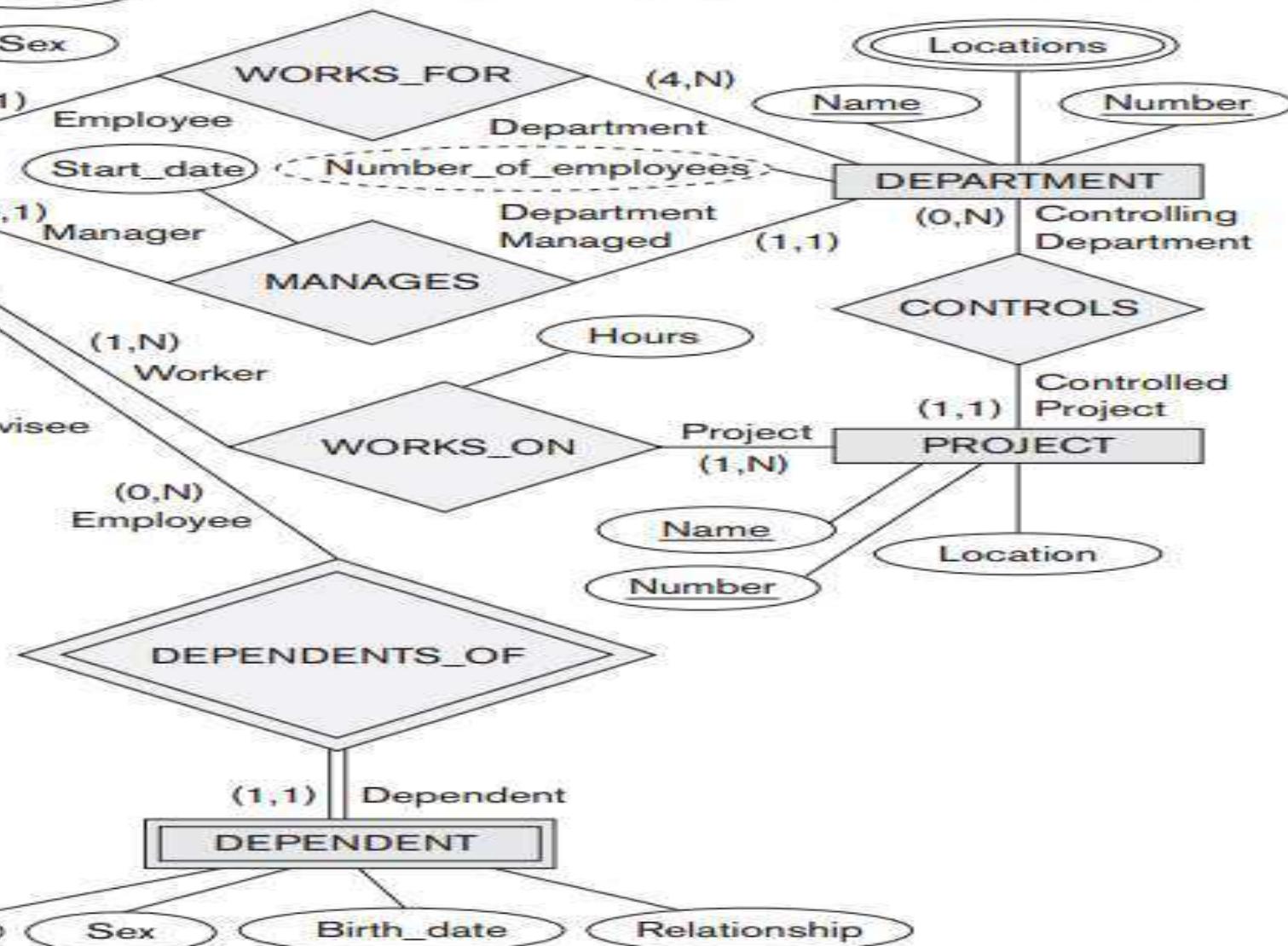


Figure 7.15

ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.



PVF WebStore: Conceptual Data Modeling

- Bước 1: Xác định thực thể
 - B1.1: Xem xét thông tin về khách hàng và kho hàng đã có từ các phiên họp JAD (xem bảng 7-2)
 - B1.2: Nghiên cứu biểu đồ DFD mức 0
 - Hai kho dữ liệu → Thực thể kho hàng và giỏ hàng
 - Xem xét các luồng dữ liệu
 - Kết quả B1.1 và B1.2: tìm ra 5 nhóm thông tin cần quan tâm
 - Khách hàng, kho hàng, đơn hàng, giỏ hàng, người dùng/thông điệp

BẢNG 7-2: Thông tin khách hàng và kho hàng cho WebStore.

Khách hàng doanh nghiệp	Khách hàng văn phòng- gia đình	Khách hàng sinh viên	Thông tin kho hàng
Tên công ty	Số điện thoại	Tên	SKU
Địa chỉ công ty	Địa chỉ email	Kinh doanh	Tên
		Trường như tên công ty	
Số điện thoại	Địa chỉ	Địa chỉ	Mô tả
Số fax	Điện thoại	Điện thoại	Kích thước sản phẩm cuối cùng
Phương thức giao hàng ưa thích	Fax	E-mail	Trọng lượng sản phẩm cuối cùng
Tên người mua	E-mail		Vật liệu có sẵn
Điện thoại người mua			Màu sắc có sẵn
Email người mua			Giá
			Thời gian chờ hàng

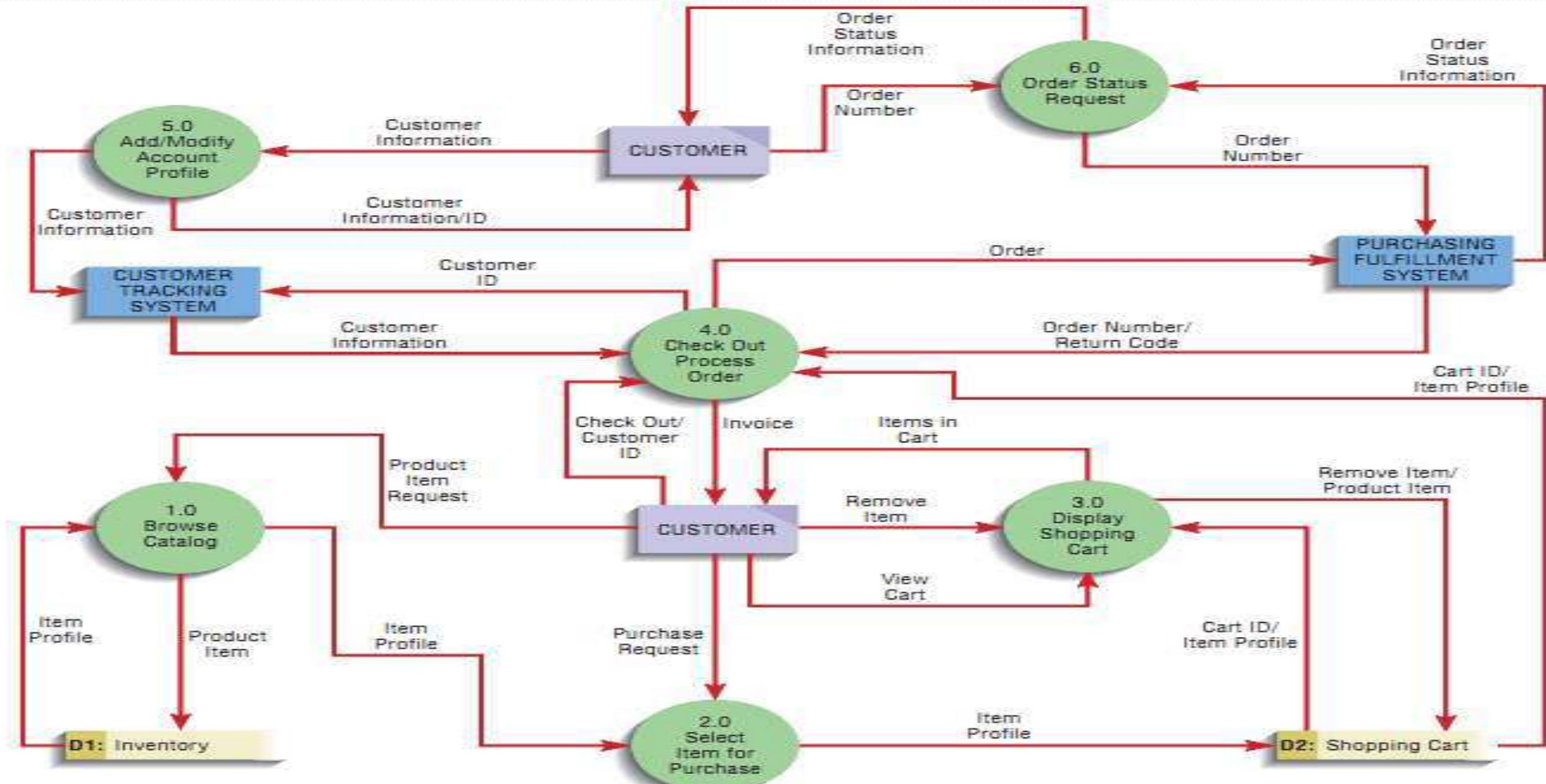


FIGURE 7-12
Level-0 data-flow diagram for the WebStore.

PVF WebStore: Conceptual Data Modeling

- Bước 1: Xác định thực thể (t.t.)

.....

- B1.3: Nghiên cứu các nhóm thông tin trong DFD
 - Xem bảng 7-3: Thuộc tính của mỗi nhóm thông tin
 - Xem bảng 7-4: Dịch chuyển của mỗi nhóm thông tin
- B1.4: Xác định 4 tập thực thể
 - Khách hàng (Customer)
 - Kho hàng (Inventory)
 - Đơn hàng (Order)
 - Giỏ hàng (Shopping cart)

- Bảng 7-3: Nhóm dữ liệu, luồng dữ liệu và các mô tả luồng dữ liệu trong sơ đồ DFD của WebStore

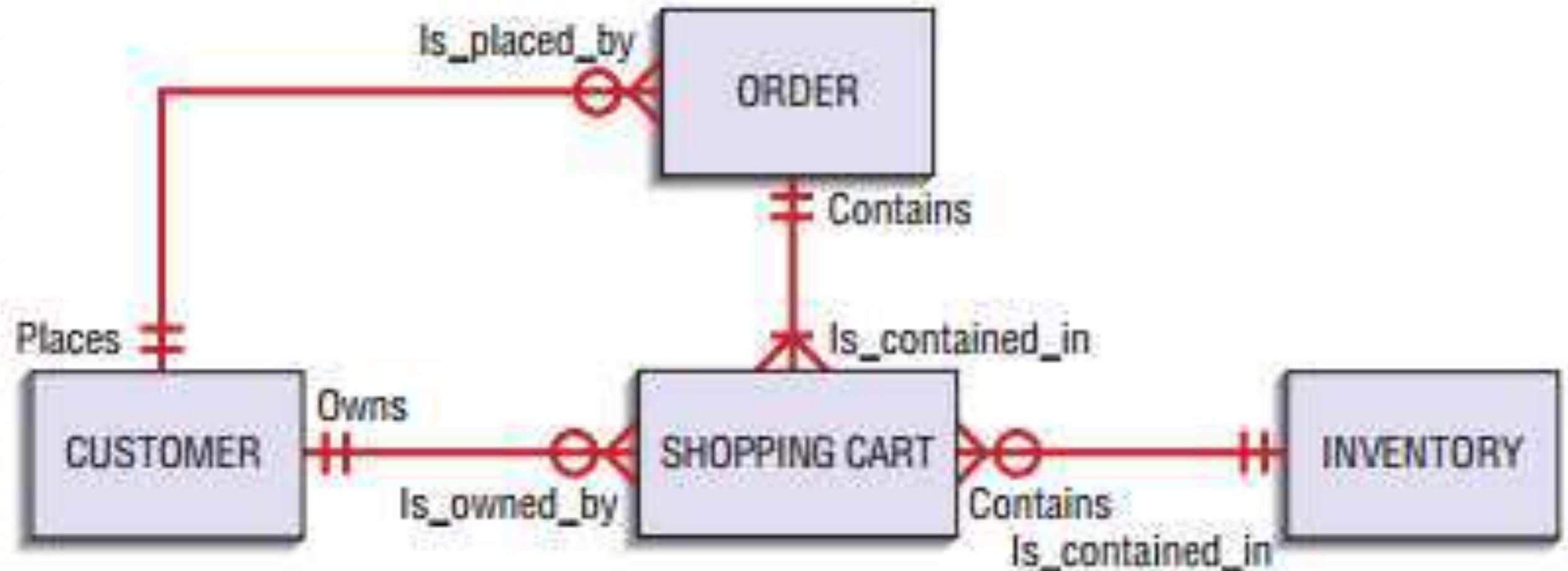
Nhóm dữ liệu	Luồng dữ liệu	Mô tả
Nhóm khách hàng		
Mã khách hàng		Định danh duy nhất được dùng để xác định mỗi khách hàng (được sinh bởi hệ thống Theo dõi khách hàng)
Thông tin khách hàng		Thông tin chi tiết về khách hàng (được lưu trong hệ thống Theo dõi khách hàng)
Nhóm kho hàng		
Thông tin sản phẩm		Định danh duy nhất dùng để xác định thông tin về mỗi sản phẩm (lưu trong cơ sở dữ liệu Kho hàng)
Thông tin mặt hàng		Thông tin chi tiết về sản phẩm (lưu trong cơ sở dữ liệu Kho hàng)
Nhóm đơn hàng		
Số hiệu hóa đơn		Định danh duy nhất dùng để xác định một đơn hàng (được sinh bởi hệ thống Đặt hàng toàn diện)
Đơn hàng		Thông tin chi tiết về đơn hàng (lưu trong hệ thống Đặt hàng toàn diện)
Mã trả vé		Mã duy nhất thể hiện kết quả xử lý từ hệ thống Đặt hàng toàn diện (được sinh và lưu trữ trong hệ thống Đặt hàng toàn diện)
Hóa đơn		Thông tin tổng kết chi tiết về đơn hàng (được tạo từ các thông tin về đơn hàng lưu trong hệ thống Đặt hàng toàn diện)
Thông tin trạng thái đơn hàng		Các thông tin chi tiết về trạng thái đơn hàng (được sinh ra/lưu trữ bởi hệ thống Đặt hàng toàn diện)
Giỏ hàng		
Mã giỏ hàng		Định danh duy nhất cho giỏ hàng
Người dùng tạm thời/Thông điệp hệ thống		
Yêu cầu thông tin sản phẩm		Yêu cầu xem thông tin về một mặt hàng trong danh mục
Yêu cầu mua hàng		Yêu cầu thêm một mặt hàng vào giỏ hàng
Xem giỏ hàng		Yêu cầu xem thông tin các mặt hàng hiện có trong giỏ hàng
Mặt hàng trong giỏ hàng		Thông tin về các mặt hàng có trong giỏ hàng
Xóa mặt hàng		Yêu cầu xóa một mặt hàng ra khỏi giỏ hàng
Thanh toán		Yêu cầu thanh toán và xử lý đơn hàng

● **Bảng 7-4:**
Nhóm dữ liệu,
luồng dữ liệu
và nguồn/dịch
của các luồng
dữ liệu bên
trong DFD của
WebStore

Nhóm dữ liệu	
Luồng dữ liệu	Từ/tới
Nhóm khách hàng	
Mã khách hàng	Từ Khách hàng tới Quy trình 4.0 Từ Quy trình 4.0 tới hệ thống Theo dõi Khách hàng Từ quy trình 5.0 tới Khách hàng
Thông tin khách hàng	Từ Khách hàng tới Quy trình 5.0 Từ Quy trình 5.0 tới Khách hàng Từ Quy trình 5.0 tới hệ thống Theo dõi Khách hàng Từ hệ thống Theo dõi Khách hàng tới Quy trình 4.0
Nhóm Kho hàng	
Thông tin sản phẩm	Từ Quy trình 1.0 tới Kho dữ liệu D1 Từ Quy trình 3.0 tới Kho dữ liệu D2
Thông tin mặt hàng	Từ Kho dữ liệu D1 tới Quy trình 1.0 Từ Kho dữ liệu D1 tới Quy trình 2.0 Từ Quy trình 2.0 tới Kho dữ liệu D2 Từ Kho dữ liệu D2 tới Quy trình 3.0 Từ Kho dữ liệu D2 tới Quy trình 4.0
Giỏ hàng	
Mã giỏ hàng	Từ Kho dữ liệu D2 tới Quy trình 3.0 Từ Kho dữ liệu D2 tới Quy trình 4.0
Người dùng tạm thời/Thông điệp hệ thống	
Yêu cầu thông tin mặt hàng	Từ Khách hàng tới Quy trình 1.0
Yêu cầu mua hàng	Từ Khách hàng tới Quy trình 2.0
Xem giỏ hàng	Từ Khách hàng tới Quy trình 3.0
Những mặt hàng trong giỏ hàng	Từ Quy trình 3.0 tới Khách hàng
Xóa mặt hàng	Từ Khách hàng tới Quy trình 3.0 Từ Quy trình 3.0 tới Kho dữ liệu D2
Thanh toán	Từ Khách hàng tới Quy trình 4.0

PVF WebStore: Conceptual Data Modeling

- Bước 2: Xác định tương quan giữa các thực thể (mỗi quan hệ)
 - Mỗi khách hàng có thể sở hữu không-tới-nhiều đối tượng giỏ hàng; mỗi đối tượng giỏ hàng được sở hữu bởi một-và-chỉ một Khách hàng
 - Mỗi đối tượng giỏ hàng có chứa một-và-chỉ-một Mặt hàng trong kho; mỗi Mặt hàng trong kho được chứa trong không-tới-nhiều đối tượng giỏ hàng.
 - Mỗi khách hàng có thể đặt không-tới-nhiều Đơn đặt hàng; mỗi Đơn đặt hàng được đặt bởi một-và-chỉ-một Khách hàng.
 - Mỗi Đơn hàng có chứa một-tới-nhiều đối tượng Giỏ hàng; mỗi đối tượng Giỏ hàng được chứa trong một-và-chỉ-một Đơn hàng



Hệ thống kiểm soát kho mới của Hoosier Burger

- Thay thế cho HT đang có
- Hình 7-15: Những yêu cầu HT và ràng buộc được xếp hạng cho Hệ thống kho của Hoosier Burger

CÁC YÊU CẦU HỆ THỐNG <i>(Theo thứ tự giảm dần)</i>	CÁC RÀNG BUỘC HỆ THỐNG <i>(Theo thứ tự giảm dần)</i>
<ol style="list-style-type: none">1. Phải dễ dàng nhập việc gửi chờ hàng (shipments) vào hệ thống nhanh chóng khi được nhận.2. Hệ thống phải tự động xác định có nên và khi nào một đặt hàng mới nên được thực hiện.3. Việc quản lý có thể xác định tại bất kỳ thời điểm nào về khoảng chừng (approximately) các mức hàng tồn kho cho bất kỳ sản phẩm nào trong kho.	<ol style="list-style-type: none">1. Phát triển hệ thống không quá chi phí \$50,0002. Phản ứng mới không thể nhiều hơn \$50,0003. Hệ thống mới phải vận hành trong vòng 6 tháng từ khi bắt đầu hợp đồng4. Những cần thiết về huấn luyện phải ít nhất (nghĩa là, hệ thống mới phải dễ sử dụng.)

Hệ thống kiểm soát kho mới của Hoosier Burger (tt)

- *Hình 7-16: Các bước của HT hiện tại của Hoosier Burger*
 - 1. Thỏa mãn các xe chở sản phẩm đến người tiêu dùng (delivery trucks) trước khi mở cửa nhà hàng
 - 2. Dỡ hàng (unload) và lưu trữ việc giao hàng
 - 3. Ghi các hóa đơn (log invoices) và hồ sơ tương ứng tập tin.
 - 4. Thêm bằng tay số lượng đã nhận vào kho
 - 5. Sau khi đóng cửa, in báo cáo hàng tồn kho
 - 6. Kiểm kê số lượng hàng tồn kho
 - 7. So sánh tổng số báo cáo hàng tồn kho với tổng số kiểm kê kho.
 - 8. So sánh tổng số kiểm kê kho với số lượng đặt hàng nhỏ nhất; nếu số lượng nhỏ hơn thì đặt hàng; nếu không thì không làm gì cả.
 - 9. Thanh toán hóa đơn đến hạn và lưu trữ lại khi đã thanh toán xong.
- Khi đề xuất những giải pháp thay thế, các yêu cầu và ràng buộc phải được xem xét

Hệ thống kiểm soát kho mới của Hoosier Burger (tt)

- Hình 7-18: liệt kê 3 giải pháp: A là dễ xuất low-end, C là high-end, và B là mid-range*

TIÊU CHUẨN	LỰA CHỌN A	LỰA CHỌN B	LỰA CHỌN C
Các yêu cầu			
1. Dễ dàng nhập dữ liệu vận chuyển mới theo thời gian thực	Có	Có	Có
2. Các quyết định đặt mua lại tự động (automatic reorder)	Cho một số mặt hàng	Cho tất cả mặt hàng	Cho tất cả mặt hàng
3. Dữ liệu thời gian thực theo các mức hàng tồn kho	Không có sẵn	Có sẵn chỉ một số mặt hàng	Có sẵn đầy đủ
Các ràng buộc			
1. Chi phí phát triển	\$25000	\$50000	\$65000
2. Chi phí phần cứng	\$25000	\$50000	\$50000
3. Thời gian hoạt động	3 tháng	6 tháng	9 tháng
4. Dễ huấn luyện	Một tuần huấn luyện	Hai tuần huấn luyện	Một tuần huấn luyện

Hệ thống kiểm soát kho mới của Hoosier Burger (tt)

- Lựa chọn giải pháp thay thế nhiều khả năng nhất
 - Cách tiếp cận trọng số** có thể được dùng để so sánh 3 giải pháp thay thế
 - Hình minh họa một cách tiếp cận trọng số cho Hoosier Burger*

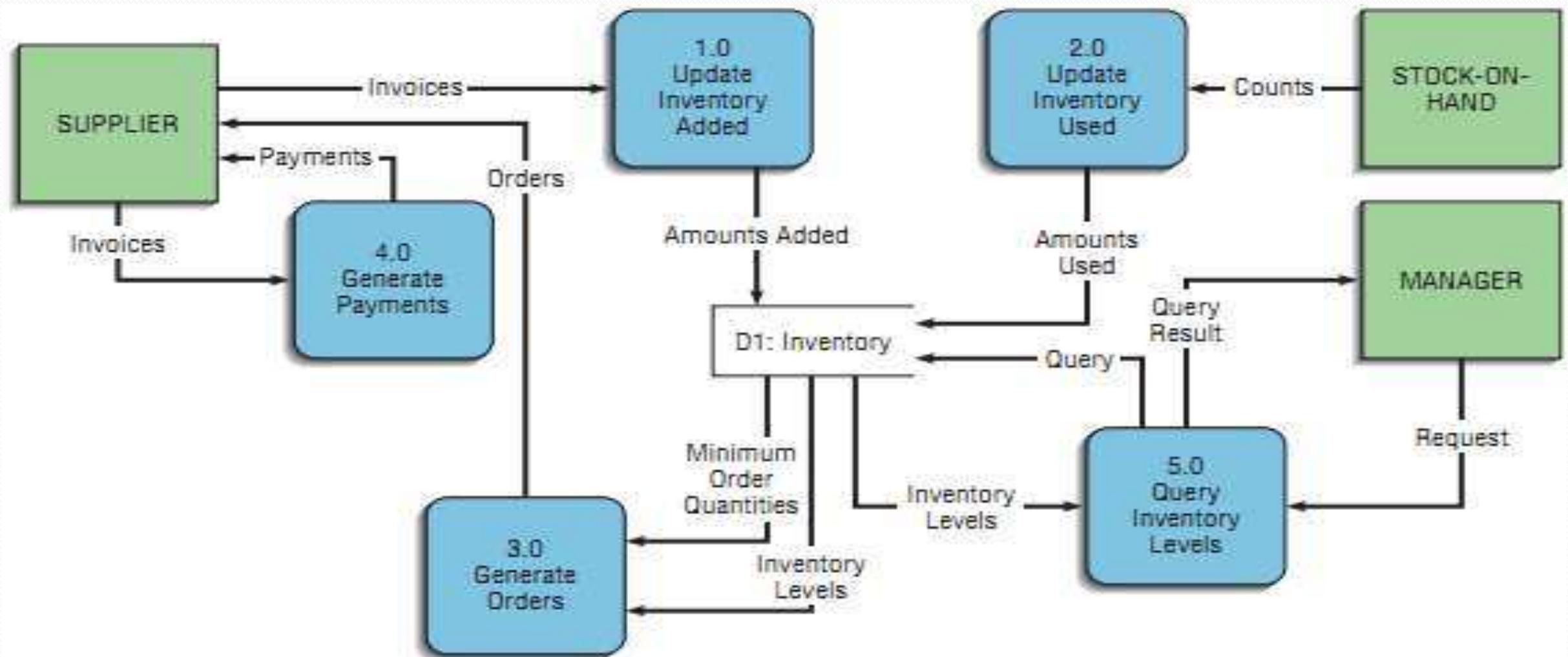
Tiêu chí	Trọng số	Phương án A		Phương án B		Phương án C	
		Xếp hạng	Điểm số	Xếp hạng	Điểm số	Xếp hạng	Điểm số
Các yêu cầu							
Nhập dữ liệu đầu vào theo thời gian thực	18	5	90	5	90	5	90
Đặt mua hàng lại tự động	18	3	54	5	90	5	90
Truy vấn dữ liệu thời gian thực	14	1	14	3	42	5	70
	50		158		222		250
Các ràng buộc							
Các chi phí phát triển	20	5	100	4	80	3	60
Các chi phí phần cứng	15	5	75	4	60	4	60
Thời gian để vận hành	10	5	50	4	40	3	30
Dễ dàng đào tạo	5	5	25	3	15	5	25
	50		250		195		175
Tổng số	100		408		417		425

Hình 7-19. Cách tiếp cận được đánh trọng số để so sánh ba hệ thống lựa chọn thay thế cho hệ thống kho hàng của Hoosier Burger.

Hoosier Burger: Conceptal Data Modeling

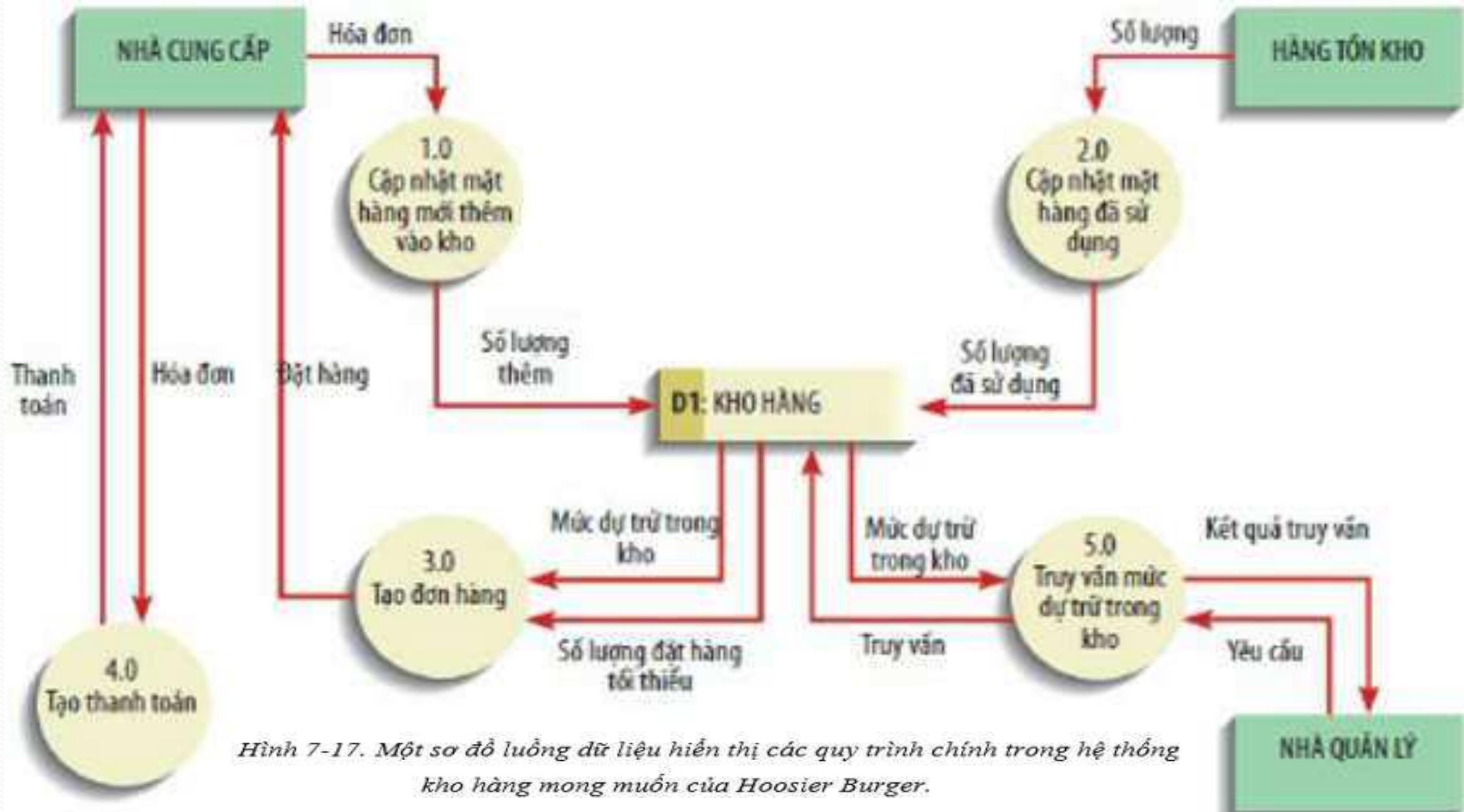
- **Chức năng:** theo dõi mức dự trữ nguyên liệu chế biến trong kho, đặt hàng và thanh toán cho nhà cung cấp
- **Mô tả:**
 - Khi NCC chuyển hàng đến, số lượng hàng trong hóa đơn sẽ được cập nhật vào kho (1.0)
 - Khi nguyên liệu được lấy chế biến, lượng hàng trong kho sẽ trừ đi lượng nguyên liệu đó (2.0)
 - Từ trạng thái hàng trong kho thực tế và quy định về mức lưu trữ an toàn của kho, ứng dụng tạo hóa đơn mua nguyên liệu (3.0) và thực hiện thanh toán với nhà cung cấp (4.0)
 - Quản lý có thể theo dõi báo cáo về thông tin hàng trong kho (5.0)

Hoosier Burger: DFD



Hình 7-17. Một sơ đồ luồng dữ liệu hiển thị các quy trình chính trong hệ thống kho hàng mong muốn của Hoosier Burger.

Hoosier Burger: DFD mức 0 (Fig 7.17)

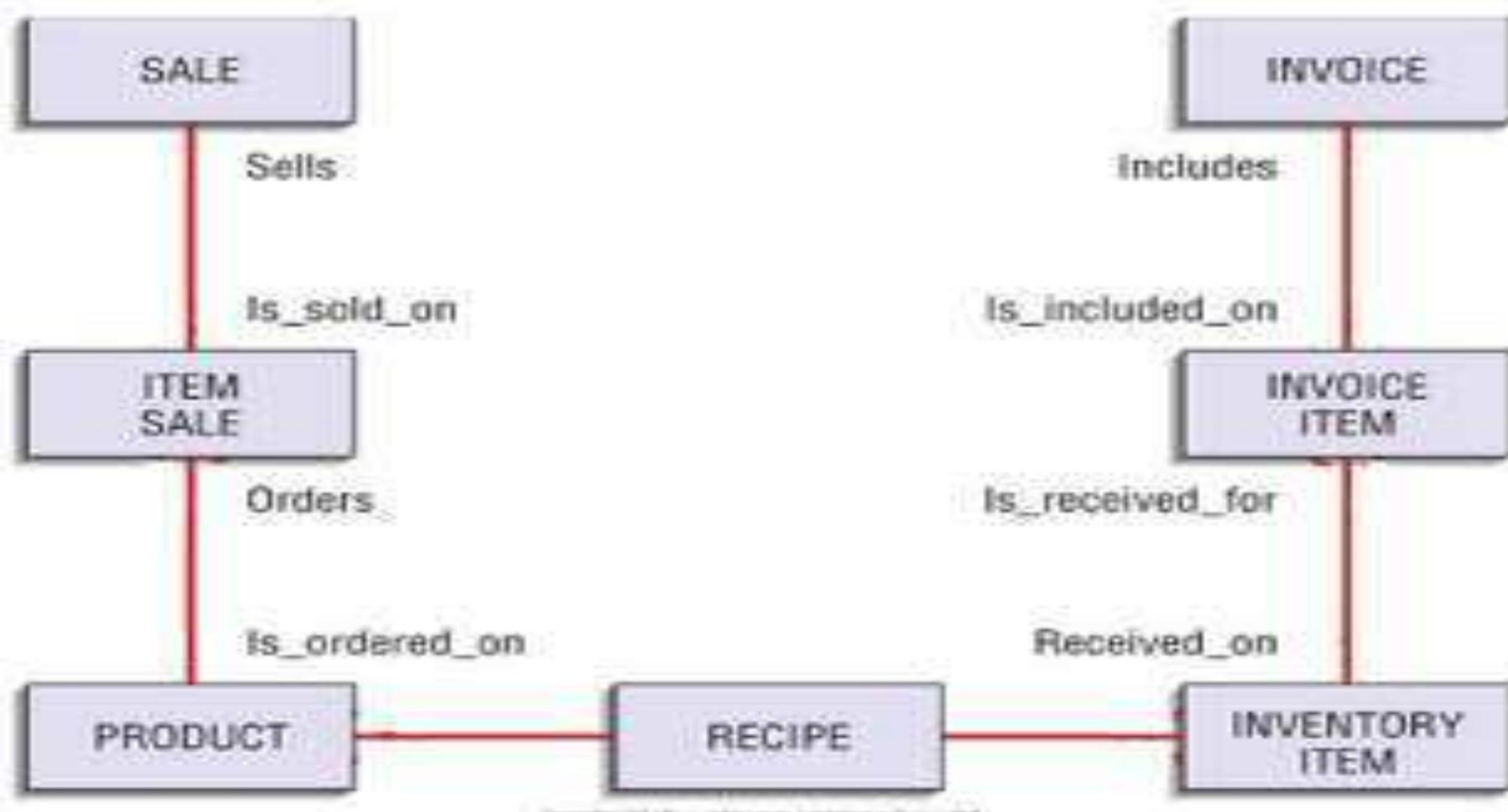


Hoosier Burger: Xây dựng ERD- Phân tích và tìm thực thể

- Inventory Levels: mức dự trữ trong kho cộng thêm số lượng nguyên liệu mới nhập về, nguyên liệu này được tính toán thông qua hóa đơn (INVOICE) mà cửa hàng nhận từ nhà cung cấp
- Mỗi hóa đơn có nhiều mặt hàng (INVOICE ITEM), mỗi mặt hàng trên hóa đơn, tương ứng với mặt hàng trong kho (INVENTORY ITEM)
- Khi khách hàng thực hiện giao dịch mua hàng (SALE), mỗi giao dịch sẽ có nhiều món ăn (SALE ITEM), trong trường hợp này, mỗi món ăn thuộc một sản phẩm (PRODUCT) của cửa hàng
- Để tạo được 1 sản phẩm cần có công thức nấu ăn (RECIPE) Mỗi sản phẩm sẽ tương ứng với lượng nguyên liệu tương ứng. Khi khách hàng mua món ăn, lượng nguyên liệu tương ứng sẽ được trừ khỏi kho.

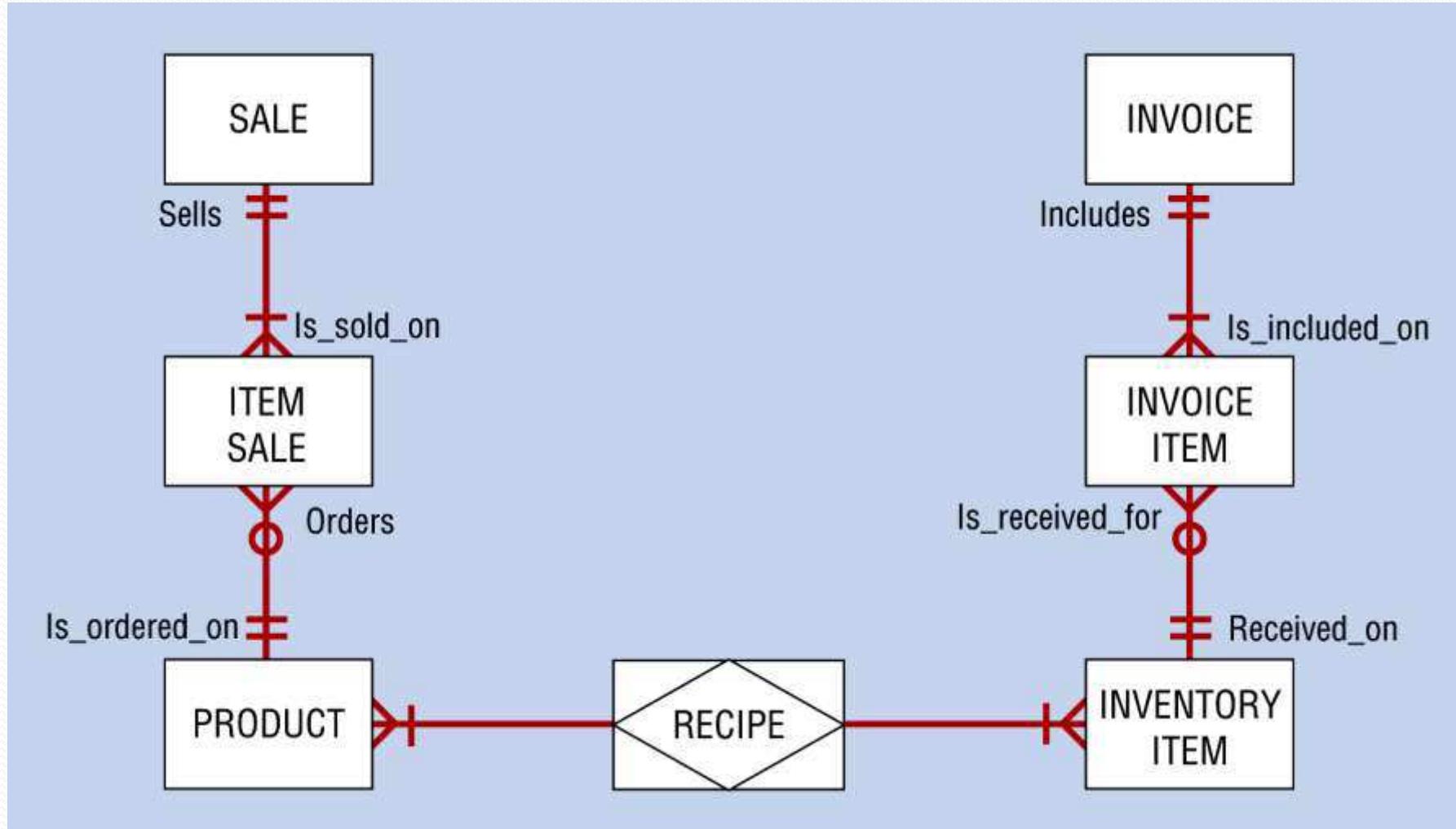
Hoosier Burger: ERD

- Phân tích và tìm các thực thể, mối quan hệ

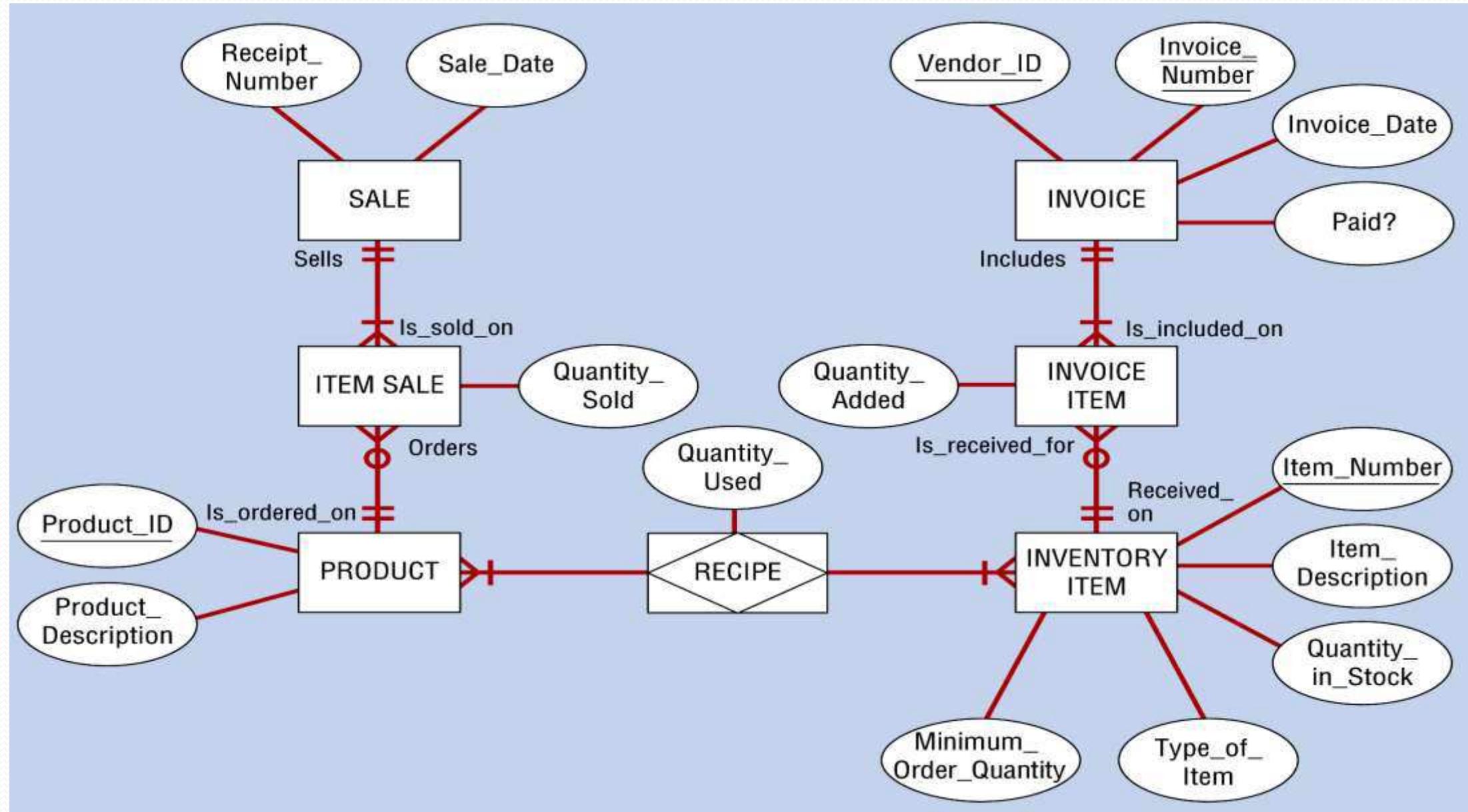


Hoosier Burger: ERD ban đầu

- Xác định bản số trong các mối quan hệ

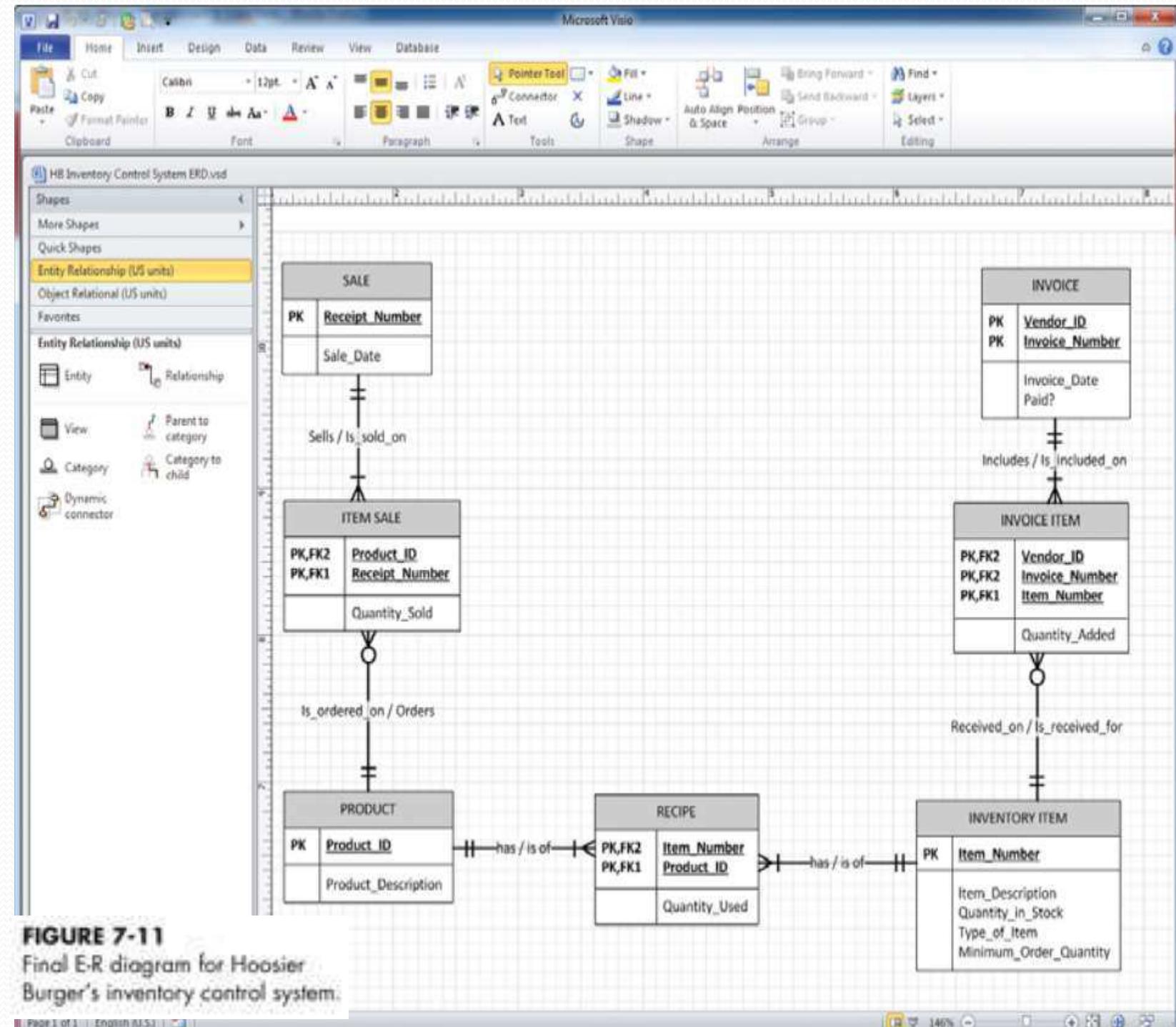


Hoosier Burger: ERD đã chỉnh sửa



Hoosier Burger: ERD

- Xác định thuộc tính của mỗi quan hệ
 - Khảo sát luồng dữ liệu
 - Khảo sát bảng logic
 - Vd: Khảo sát luồng dữ liệu của quy trình 2.0
 - Khi KH thực hiện mua hàng, số lượng hàng đã bán được cập nhật vào csdl
 - Do đó trong thực thể SALE ITEM có thuộc tính Quantity_Sold (số lượng hàng đã bán)



Bài tập- Vẽ ERD

- Vd:
 - Một công ty thương mại Y chuyên kinh doanh các mặt hàng điện tử. Công ty nhập các mặt hàng từ các nhà cung cấp khác nhau. Chi tiết về các mặt hàng gồm có: mã hàng (duy nhất), tên hàng và các mô tả mặt hàng.
 - Công ty cũng cần lưu giữ thông tin về các nhà cung cấp như tên, địa chỉ, điện thoại, fax. Mỗi nhà cung cấp có một mã duy nhất. Mỗi nhà cung cấp có thể cung cấp nhiều mặt hàng nhưng mỗi mặt hàng chỉ được cung cấp từ một nhà cung cấp.
 - Các mặt hàng được lưu giữ trong các kho. Mỗi kho hàng có một diện tích khác nhau và chỉ chứa một loại mặt hàng.
 - Công ty có nhiều cửa hàng đại lý để bán các mặt hàng. Hàng được cung cấp cho các cửa hàng thông qua các phiếu xuất. Thông tin trên mỗi phiếu xuất cần có mã số cửa hàng nhận hàng, ngày xuất, thông tin về các mặt hàng được xuất như tên hàng, số lượng, đơn giá, thành tiền.
- Yêu cầu: Vẽ mô hình Entity-Relationship HT bán hàng cho công ty Y
 - 1. Xác định các thực thể, định danh thực thể và các thuộc tính mô tả
 - 2. Xác định liên kết giữa các thực thể
 - 3. Vẽ mô hình thực thể liên kết

Enhanced Entity Relationship Model (EER Model)

- EER model
 - Được tạo ra để thiết kế các lược đồ csdl chính xác hơn (database schemas)
 - Phản ánh các thuộc tính dữ liệu và các ràng buộc chính xác hơn
 - Các yêu cầu phức tạp hơn các ứng dụng truyền thống

Các lớp con, các siêu lớp, và Tính kế thừa

- Mô hình EER gồm tất cả các khái niệm lập mô hình của mô hình ER
- Hơn nữa EER còn bao gồm:
 - Các lớp con (Subclasses) và các siêu lớp/lớp cha (superclasses)
 - Chuyên biệt hóa (Specialization) và tổng quát hóa (generalization)
 - Khái niệm category hoặc kiểu Union (union type)
 - Tính kế thừa Thuộc tính (Attribute) và mối quan hệ/liên kết (relationship)
- Các sơ đồ EER
 - Kỹ thuật sơ đồ giúp hiển thị những khái niệm này trong 1 lược đồ EER

Các lớp con, các siêu lớp, và Tính kế thừa

- Tập con (Subtype) hoặc lớp con (subclass) của 1 tập thực thể
 - Các nhóm con của các thực thể có ý nghĩa
 - Được biểu diễn rõ ràng vì sự quan trọng với ứng dụng csdl

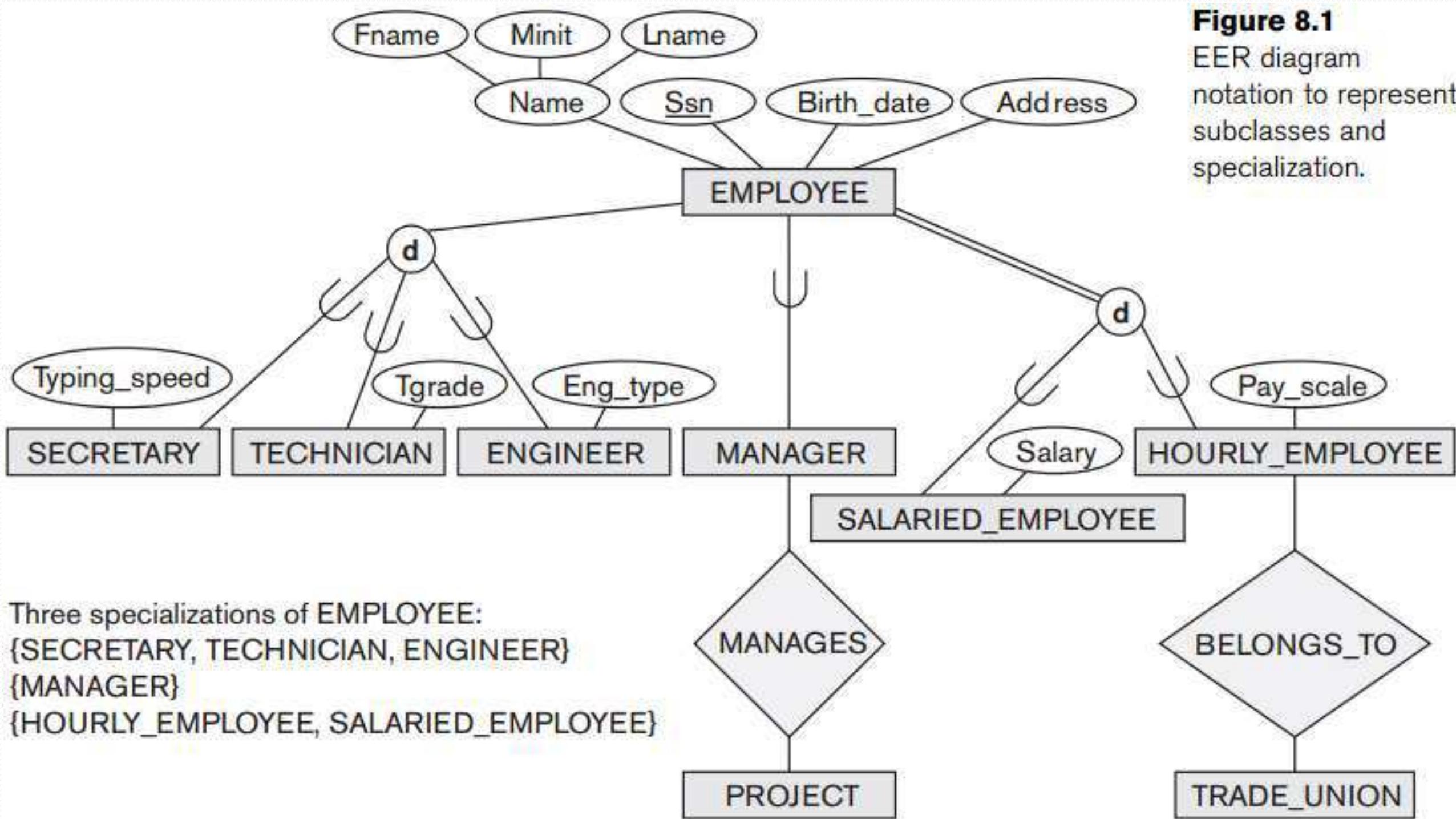
Các lớp con, các siêu lớp, và Tính kế thừa

- Các thuật ngữ cho mối quan hệ giữa 1 siêu lớp với mỗi lớp con bất kỳ của nó
 - Siêu lớp/ lớp con
 - Kiểu cha/kiểu con
 - Mối quan hệ lớp/lớp con (thường được gọi là một mối quan hệ IS-A)
- Kiểu kế thừa (type inheritance)
 - Thực thể lớp con thừa kế tất cả các thuộc tính và mối quan hệ của siêu lớp.

Các lớp con, các siêu lớp, và Tính kế thừa

- Vd: EMPLOYEE có thể được nhóm nhỏ hơn thành SECRETARY, ENGINEER, MANAGER, TECHNICIAN, SALARIED_EMPLOYEE, HOURLY_EMPLOYEE,...
 - Tập hợp các thực thể trong mỗi nhóm con (subgroup) là 1 tập con (subset) của tập thực thể EMPLOYEE
 - Mỗi tập được gọi là 1 lớp con của EMPLOYEE
 - EMPLOYEE là siêu lớp của mỗi lớp con này.

Figure 8.1
EER diagram
notation to represent
subclasses and
specialization.



Các lớp con, các siêu lớp, và Tính kế thừa

- Một thành viên lớp con cũng như thực thể trong siêu lớp, những trong một vai trò riêng biệt cụ thể.
- Một thực thể không thể tồn tại trong csdl đơn thuần là 1 thành viên của 1 lớp con; mà nó cũng phải là 1 thành viên của siêu lớp
- Một thành viên của 1 siêu lớp có thể được đính kèm tùy chọn như là 1 thành viên của một lớp con nào đó của nó.
- Không cần thiết mỗi thực thể trong 1 siêu lớp phải là 1 thành viên của lớp con nào đó
- Mỗi quan hệ siêu lớp/lớp con là 1:1

Các lớp con, các siêu lớp, và Tính kế thừa

- Thực thể lớp con kế thừa tất cả thuộc tính và mối quan hệ của siêu lớp
- Chú ý rằng một lớp con với những thuộc tính và mối quan hệ cụ thể của chính nó (hoặc cục bộ) mà nó kế thừa từ siêu lớp, có thể được xem như 1 tập thực thể độc lập.

Chuyên biệt hóa và Tổng quát hóa

- Chuyên biệt hóa
 - Tiến trình định nghĩa 1 tập các lớp con của một tập thực thể, được gọi là siêu lớp
 - Được định nghĩa dựa trên cơ sở đặc điểm phân biệt nào đó của các thực thể trong siêu lớp
 - Có thể có vài chuyên biệt hóa của cùng tập thực thể được dựa trên những đặc điểm phân biệt khác nhau.
- Lớp con có thể sở hữu
 - Các thuộc tính đặc trưng (thuộc tính địa phương/cục bộ)
 - Các kiểu mối quan hệ đặc trưng

Chuyên biệt hóa và Tổng quát hóa (tt)

- 2 lý do chính để gắn các mối quan hệ lớp/lớp con và chuyên biệt hóa trong 1 mô hình dữ liệu:
 - 1. Một số thuộc tính nào đó có thể áp dụng cho một vài thực thể chứ không phải tất cả các thực thể của siêu lớp
 - 2. Một vài loại mối quan hệ có thể được tham gia chỉ bởi các thực thể là các thành viên của lớp con.

Tổng quát hóa

- Tiến trình ngược của trừu tượng
- Tổng hợp vài tập thực thể có một số đặc tính chung thành thành 1 siêu lớp đơn
 - Những tập thực thể gốc là những lớp con đặc biệt
- Tổng quát hóa
 - Tiến trình định nghĩa 1 tập thực thể được tổng hợp từ những tập thực thể được đưa ra.

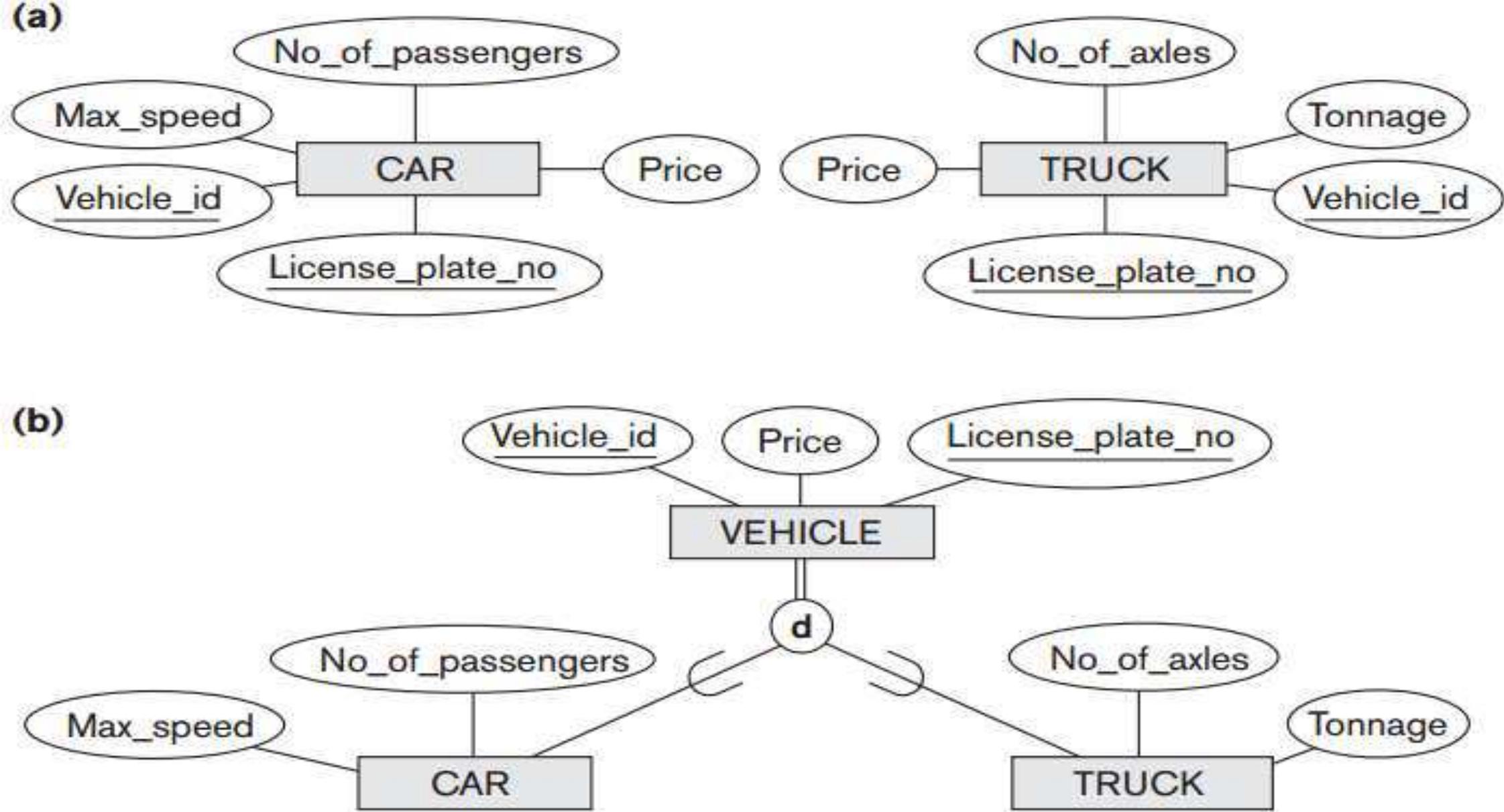


Figure 8.3

Generalization. (a) Two entity types, CAR and TRUCK. (b) Generalizing CAR and TRUCK into the superclass VEHICLE.

Chuyên biệt hóa và Tổng quát hóa

- Lập mô hình dữ liệu với Chuyên biệt hóa và Tổng quát hóa
 - Một siêu lớp hoặc lớp con biểu diễn 1 tập hợp các thực thể
 - Được vẽ bằng hình chữ nhật trong các sơ đồ EER (như các tập thực thể)
 - Thỉnh thoảng, tất cả tập thực thể đơn giản được gọi các lớp, cho dù chúng là các tập thực thể, các siêu lớp, hoặc các lớp con.

Những ràng buộc và đặc điểm của những thứ tự phân cấp chuyên biệt hóa và Tổng quát hóa

- Những ràng buộc áp dụng cho 1 chuyên biệt hóa đơn hoặc 1 tổng quát hóa đơn
- Những sự khác nhau giữa các lưới (lattices) và thứ tự phân cấp (hierarchies) chuyên biệt hóa/tổng quát hóa

Những ràng buộc trên Chuyên biệt hóa và Tổng quát hóa

- Có thể là vài hoặc 1 lớp con
- Xác định tập con thực thể:
 - Các lớp con vị ngũ-được định nghĩa (hoặc điều kiện-được định nghĩa)
 - Chuyên biệt hóa Thuộc tính-được định nghĩa
 - Người dùng-được định nghĩa

Những ràng buộc trên Chuyên biệt hóa và Tổng quát hóa

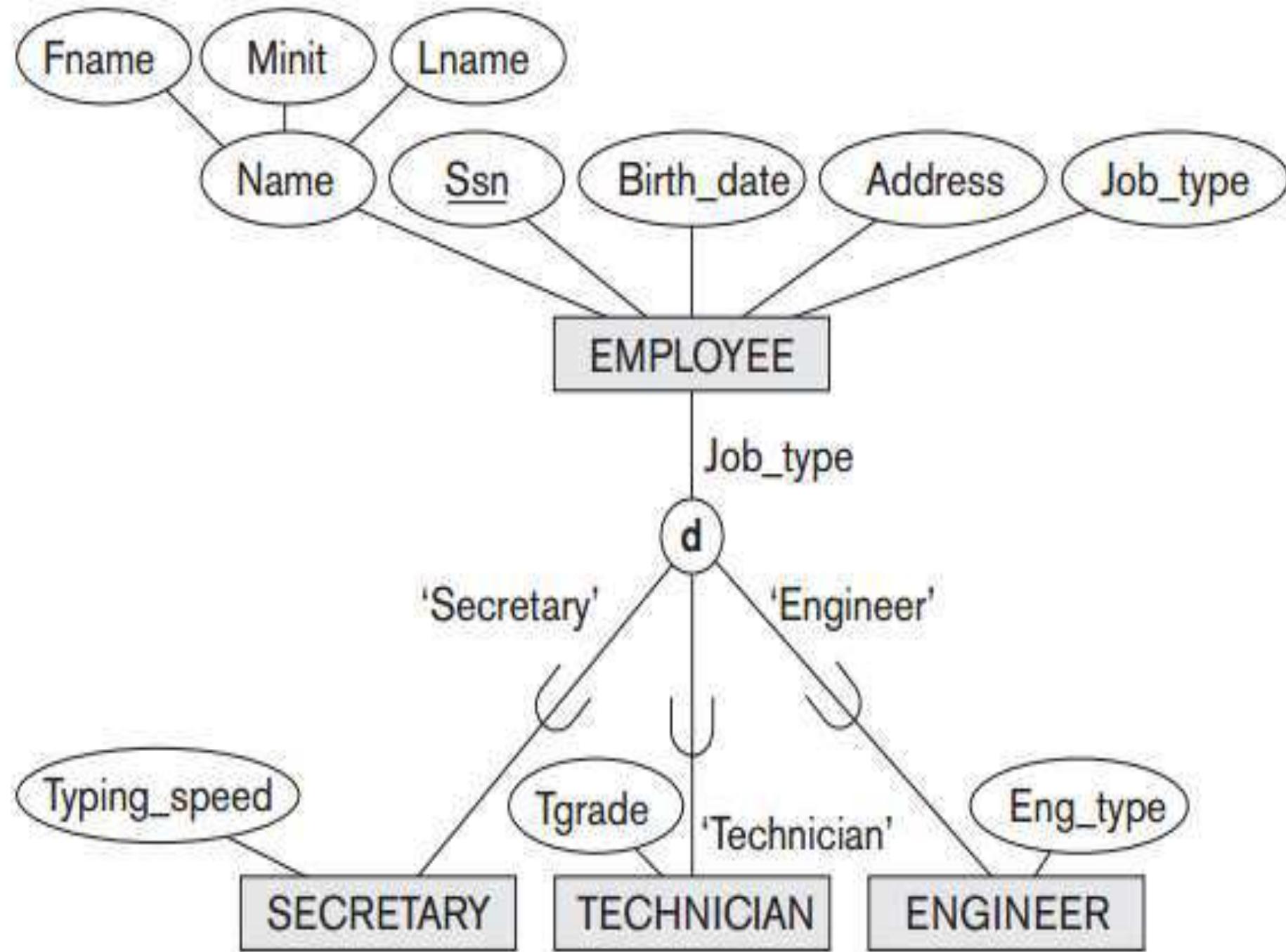
- Nếu chúng ta có thể xác định chính xác những thực thể mà sẽ trở thành các thành viên của mỗi lớp con bởi 1 điều kiện, thì các lớp con được gọi là các lớp con **vị ngữ-định nghĩa** (hoặc **điều kiện-định nghĩa**)
 - Điều kiện là 1 ràng buộc xác định các thành viên lớp con
 - Hiển thị 1 lớp con **vị ngữ-định nghĩa** bằng cách viết điều kiện **vị ngữ** bên cạnh đường gắn lớp con với siêu lớp của nó.

Những ràng buộc trên Chuyên biệt hóa và Tổng quát hóa (tt)

- Nếu tất cả lớp con trong 1 chuyên biệt hóa có điều kiện tư cách thành viên trên cùng thuộc tính của siêu lớp, thì chuyên biệt hóa được gọi là 1 chuyên biệt hóa thuộc tính định nghĩa
 - Thuộc tính được gọi thuộc tính định nghĩa của chuyên biệt hóa
 - Vd: JobType là thuộc tính định nghĩa của chuyên biệt hóa {SECRETARY, TECHNICIAN, ENGINEER} của EMPLOYEE

Figure 8.4

EER diagram notation for an attribute-defined specialization on Job_type.



Những ràng buộc trên Chuyên biệt hóa và Tổng quát hóa (tt)

- Nếu không có điều kiện xác định tư cách thành viên thì lớp con được gọi là người dùng-định nghĩa
 - Tư cách thành viên trong 1 lớp con được xác định bởi những người dùng csdl bằng cách áp dụng 1 thao tác để thêm 1 thực thể vào lớp con
 - Tư cách thành viên trong 1 lớp con được đặc tả riêng lẻ cho mỗi thực thể trong siêu lớp bởi người dùng.

Những ràng buộc trong Chuyên biệt hóa và Tổng quát hóa (tt)

- Hai điều kiện cơ bản áp dụng cho 1 chuyên biệt hóa/tổng quát hóa: các ràng buộc tính phân ly và tính đầy đủ.
- Ràng buộc về tính phân ly (Disjointness constraint)
 - Có thể phân ly (disjoint) hoặc trùng lặp (overlap)
- Ràng buộc về tính đầy đủ (completeness constraint) hoặc tính toàn phần (totalness)
 - Có thể là toàn phần (mỗi thực thể trong siêu lớp phải là một thành viên của không ít hơn 1 lớp con trong specialization) hoặc riêng phần (cho phép 1 thực thể không phụ thuộc vào bất kỳ lớp con nào)
- Những ràng buộc về tính phân ly và tính đầy đủ là độc lập

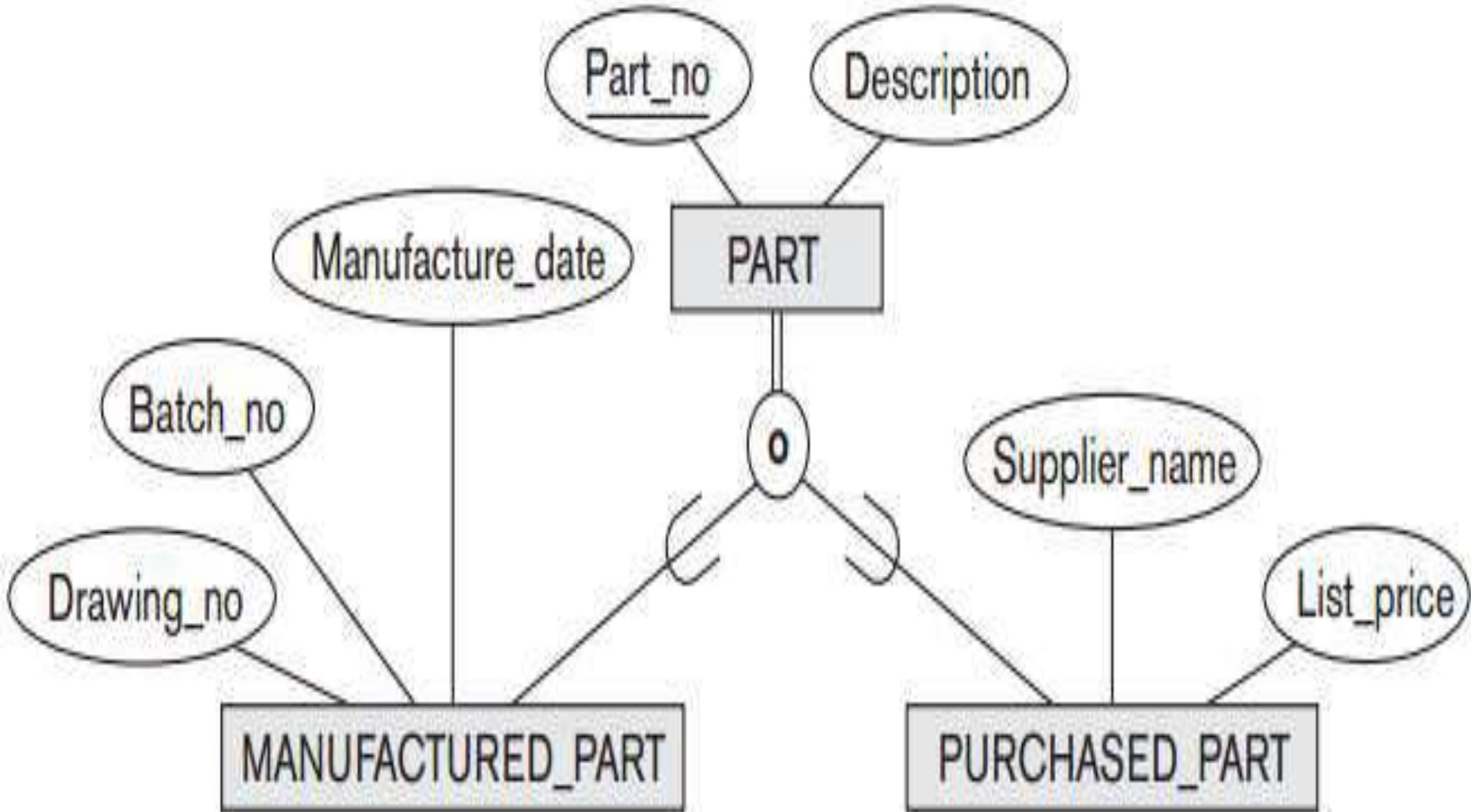
Những ràng buộc trong Chuyên biệt hóa và Tổng quát hóa (tt)

- Ràng buộc về tính phân ly (Disjointness constraint)
 - Đặc tả các lớp con của chuyên biệt hóa phải được **phân ly** (*1 thể hiện của thực thể của lớp cha có thể là 1 thể hiện của chỉ 1 thực thể con trong các lớp con của chuyên biệt hóa*)
 - Được đặc tả bằng **d** trong sơ đồ EER
 - Nếu không phân ly thì **chồng lấp**;
(nghĩa là thể hiện của thực thể cha có thể là 1 thể hiện của nhiều thực thể con trong các lớp con của chuyên biệt hóa)
 - Được đặc tả bằng **o** trong sơ đồ EER

Những ràng buộc trong Chuyên biệt hóa và Tổng quát hóa (tt)

- Ràng buộc về tính đầy đủ (completeness constraint) hoặc tính toàn phần (totalness)
 - **Toàn phần** đặc tả rằng mỗi thể hiện của thực thể cha trong siêu lớp phải là 1 thể hiện của một trong những thực thể con nào đó trong chuyên biệt hóa/tổng quát hóa
 - được biểu diễn bằng **đường đôi**
 - **Riêng phần** cho phép 1 thể hiện của thực thể cha có thể không là thể hiện của thực thể con nào cả (*ie. không phụ thuộc vào bất kỳ lớp con nào*)
 - được biểu diễn bằng 1 **đường đơn**.

Figure 8.5
EER diagram notation
for an overlapping
(nondisjoint)
specialization.



Những ràng buộc trong Chuyên biệt hóa và Tổng quát hóa (tt)

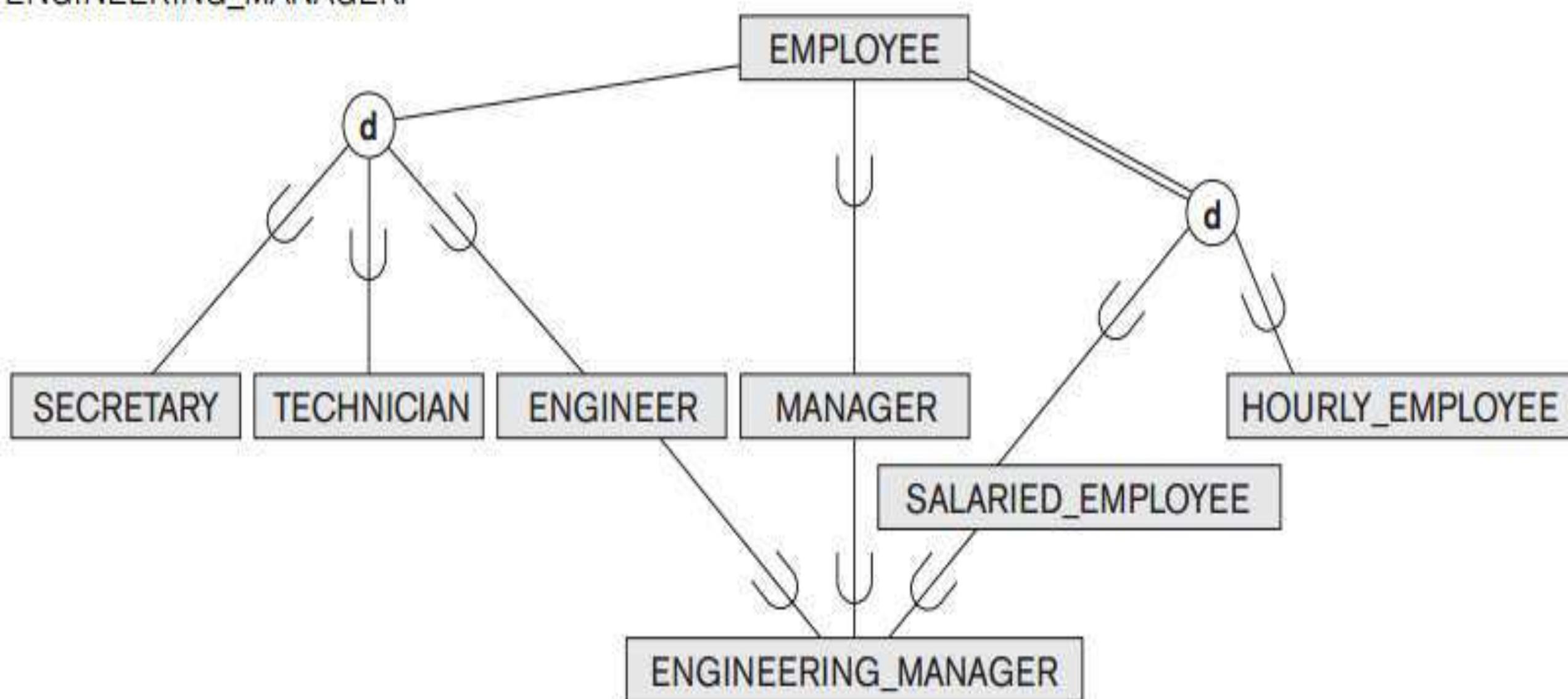
- Những ràng buộc về tính phân ly và tính đầy đủ là độc lập. Vì vậy chúng ta có 4 loại chuyên biệt hóa/tổng quát hóa:
 - Phân ly, toàn phần
 - Phân ly, riêng phần
 - Chồng lấp, toàn phần
 - Chồng lấp, riêng phần
- Chú ý: tổng quát hóa thường toàn phần vì siêu lớp được xuất phát từ các lớp con

Những thứ tự phân cấp và lưới Chuyên biệt hóa và Tổng quát hóa

- Một lớp con có thể chính nó có các lớp con tiếp theo được đặc tả trên nó, hình thành 1 thứ tự phân cấp hoặc 1 lưới.
- Thứ tự phân cấp chuyên biệt hóa
 - Mỗi lớp con tham gia như 1 lớp con chỉ trong 1 mối quan hệ lớp/lớp con
 - Dẫn đến 1 cấu trúc cây hoặc thứ tự phân cấp nghiêm ngặt (strict hierarchy)
 - Tính kế thừa đơn
- Lưới chuyên biệt hóa (specialization lattice)
 - Lớp con có thể là 1 lớp con trong nhiều hơn 1 mối quan hệ lớp/lớp con
 - Tính đa kế thừa

Figure 8.6

A specialization lattice with shared subclass
ENGINEERING_MANAGER.



Những thứ tự phân cấp và lưới Chuyên biệt hóa và Tổng quát hóa (tt)

- Trong 1 lưới hoặc thứ tự phân cấp, 1 lớp con kế thừa các thuộc tính không chỉ của siêu lớp trực tiếp của nó, mà còn của tất cả các siêu lớp trước đó (predecessor superclasses)
- Một lớp con với nhiều hơn 1 siêu lớp được gọi là 1 lớp con được chia sẻ (shared subclass)
- Trong trường hợp đa kế thừa, nếu thuộc tính (hoặc mối quan hệ) bắt nguồn trong cùng siêu lớp được kế thừa nhiều hơn 1 lần qua những đường dẫn khác nhau trong lưới
 - Được đính kèm chỉ 1 lần trong lớp con được chia sẻ.

VD là 1 phần của 1 lược đồ quan niệm cho 1 csdl UNIVERSITY.

Yêu cầu cho 1 phần csdl UNIVERSITY như sau:

- CSDL theo dõi 3 loại persons: employees, alumni, và students. 1 người có thể phụ thuộc vào 1, 2 hoặc 3 loại này. Mỗi người có 1 name, SSN, sex, address, và birth date.
- Mỗi employee có 1 salary, và có 3 loại employee: faculty, staff, và student assistants. Mỗi employee phụ thuộc chính xác 1 trong những loại này với mỗi alumnus, 1 bảng ghi của 1 degree hoặc nhiều degree mà anh/chị có tại đại học, gồm name of the degree, year granted, và major department. Mỗi student có 1 major department.
- Mỗi faculty có 1 rank (cấp bậc), trong khi mỗi staff member có một staff position. Student assistants được phân lớp tiếp theo hoặc là research assistant hoặc teaching assistants, và phân trăm thời gian mà họ làm việc được ghi nhận trong csdl. Research assistants có research project được lưu trữ, trong khi teaching assistant có khóa học hiện tại họ làm việc.
- Students được phân lớp tiếp theo hoặc là graduate hoặc undergraduate với những thuộc tính cụ thể chương trình degree (bằng cấp như M.S., Ph.D., M.B.A,...) cho các graduate students và class (freshman – sv năm 1, sophomore-sv năm 2,...) cho undergraduates

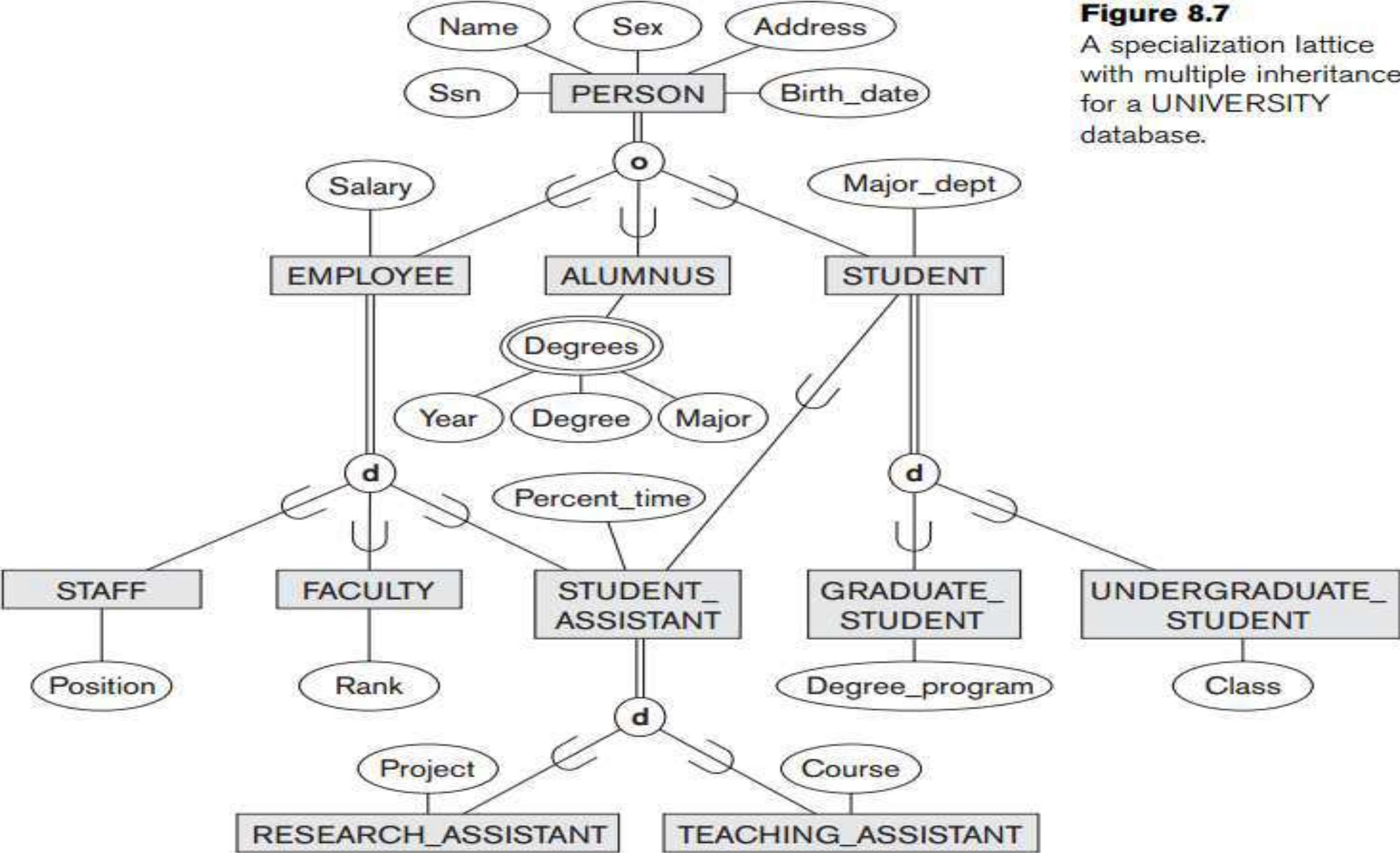


Figure 8.7

A specialization lattice with multiple inheritance for a UNIVERSITY database.

Dùng Chuyên biệt hóa và Tổng quát hóa trong việc điều chỉnh các lược đồ Quan niệm

- Tiến trình chuyên biệt hóa
 - Bắt đầu với tập thực thể sau đó định nghĩa các lớp con bằng chuyên biệt hóa lần lượt
 - Tiến trình điều chỉnh quan niệm từ trên-xuống
- Tổng hợp khái niệm từ dưới-lên
 - Liên quan việc tổng quát hóa hơn chuyên biệt hóa.

Lập mô hình của các kiểu UNION bằng cách dùng Categories

- kiểu Union (Union type) hoặc 1 category
 - Biểu diễn 1 mối quan hệ siêu lớp/lớp con đơn với nhiều hơn 1 siêu lớp
 - Lớp con biểu diễn 1 tập hợp các đối tượng đó là 1 tập con của UNION các tập thực thể riêng biệt (distinct entity types)
 - Kế thừa thuộc tính làm việc chọn lọc hơn
 - Category có thể là toàn phần hoặc riêng phần
- Một số phương pháp lập mô hình không có các kiểu union

Categories

- Ví dụ: csdl cho việc đăng ký phương tiện (vehicle registration), chủ phương tiện có thể là 1 người (person), 1 nhà băng (bank) hoặc 1 công ty.
 - Category (subclass) OWNER là 1 tập hợp con của việc hợp 3 lớp con COMPANY, BANK, và PERSON
 - Một thành viên category phải tồn tại trong ít nhất 1 trong số các siêu lớp của nó.
- Chú ý: khác với shared subclass, category là 1 tập con giao nhau (intersection) của các siêu lớp của nó (thành viên shared subclass phải tồn tại trong tất cả siêu lớp của nó)

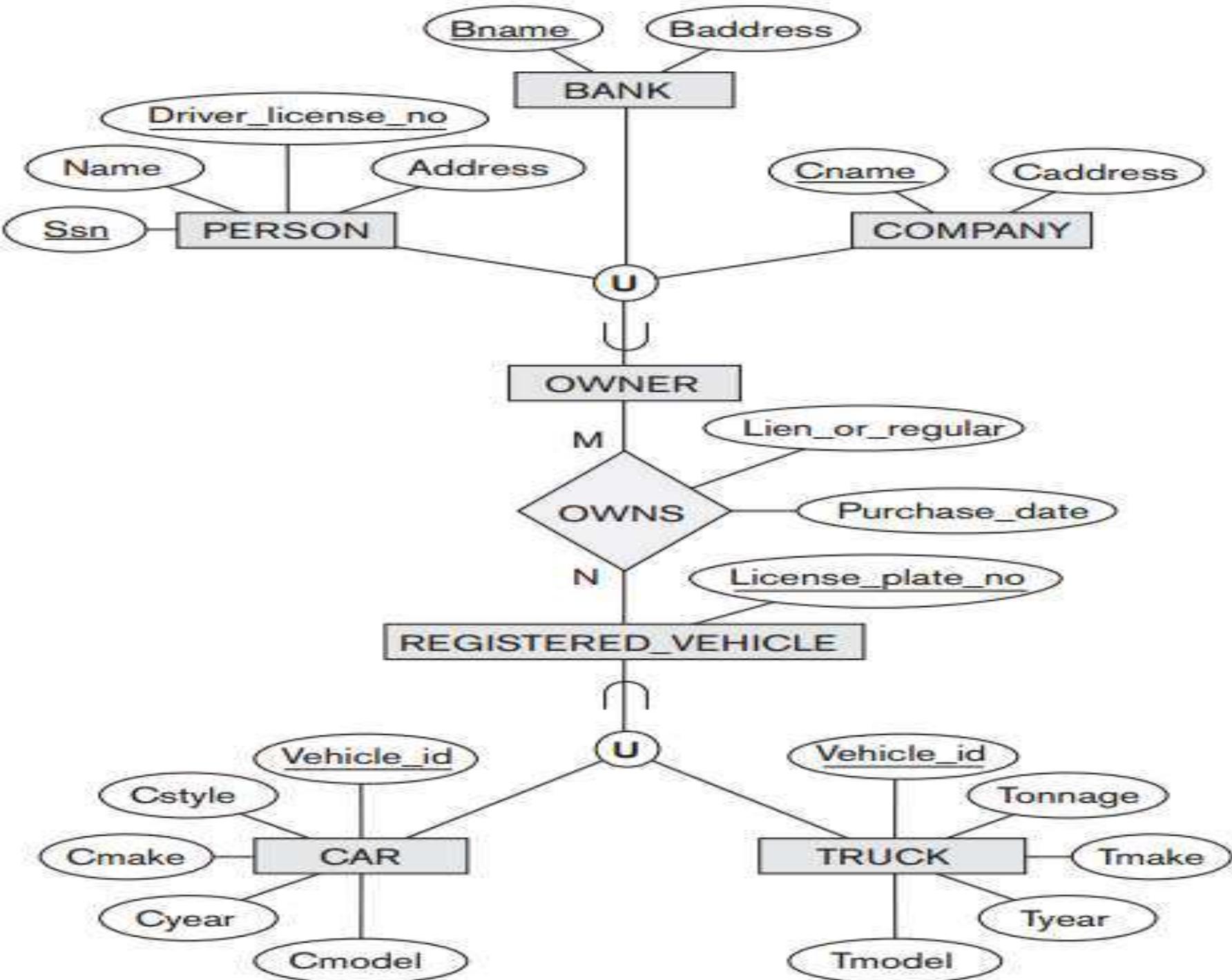


Figure 8.8
Two categories (union types): **OWNER** and **REGISTERED_VEHICLE**.

1 ví dụ lược đồ EER UNIVERSITY, các lựa chọn thiết kế,
và các định nghĩa chính thức.

- Ví dụ CSDL UNIVERSITY

- CSDL UNIVERSITY

- Các sinh viên (students) và chuyên ngành (majors) của họ
 - Bảng điểm (transcripts), và đăng ký (registration)
 - Đăng ký khóa học của trường đại học

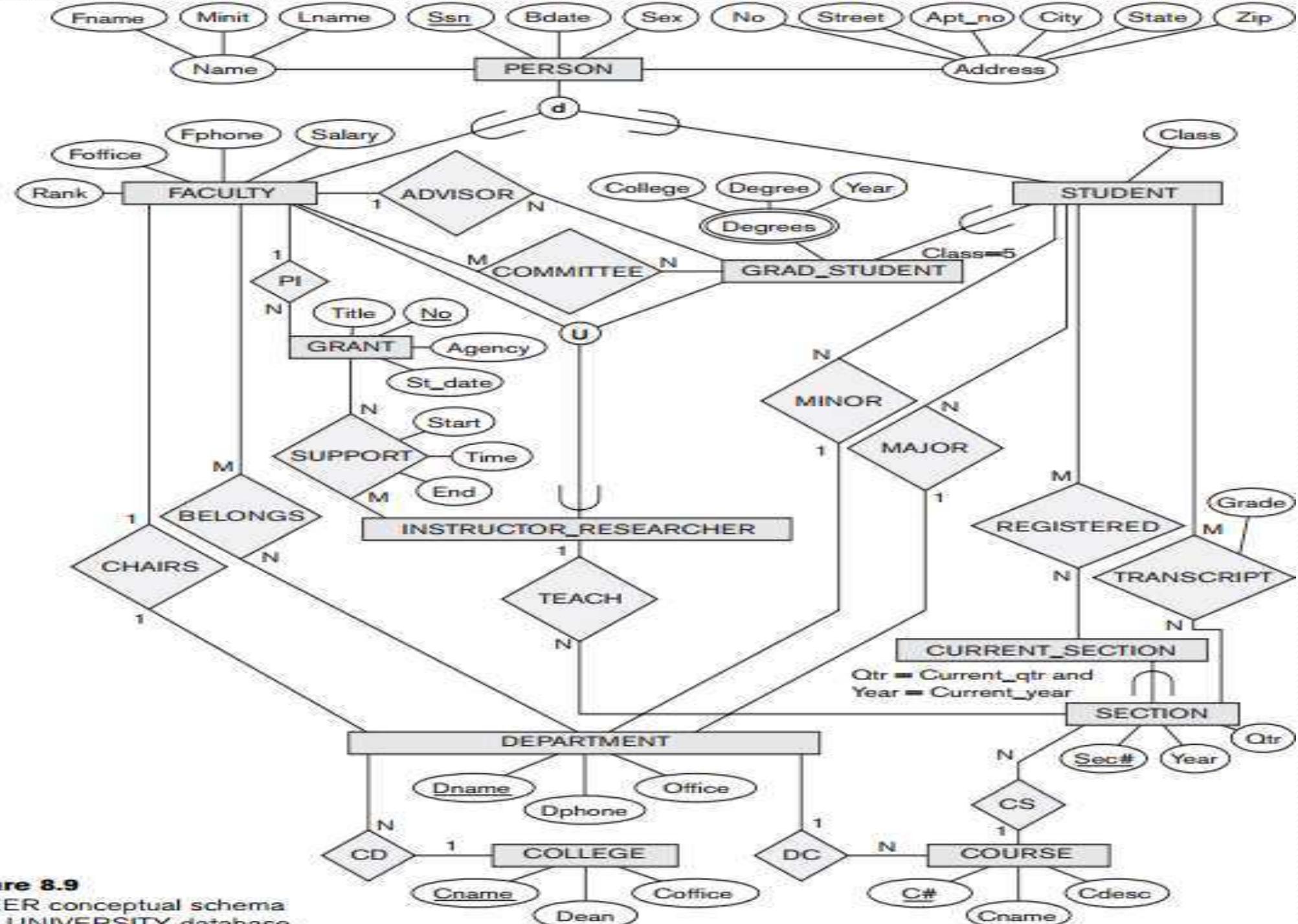


Figure 8.9
An EER conceptual schema
for a UNIVERSITY database.

Các lựa chọn thiết kế cho Chuyên biệt hóa/Tổng quát hóa

- Nhiều chuyên biệt hóa và các lớp con có thể được định nghĩa để làm mô hình quan niệm chính xác
- Nếu lớp con có vài thuộc tính cụ thể và không có các mối quan hệ cụ thể
 - Có thể được trộn thành siêu lớp
- Nếu tất cả các lớp con của 1 chuyên biệt hóa/tổng quát hóa có vài thuộc tính cụ thể và không có các mối quan hệ cụ thể
 - Có thể được trộn thành siêu lớp
 - Thay thế bằng 1 hoặc nhiều thuộc tính *type đặc* tả lớp con hoặc các lớp con mà mỗi thực thể phụ thuộc.
- Kiểu Union và categories nói chung nên tránh
- Lựa chọn các ràng buộc phân ly/chồng lấp và toàn phần/riêng phần trên chuyên biệt hóa/tổng quát hóa
 - Được dẫn dắt bởi các luật trong thế giới nhỏ được lập mô hình

Các định nghĩa chính thức cho các Khái niệm Mô hình EER

- Lớp C (Class C)

- Tập hợp các thực thể
- Gồm bất kỳ các cấu trúc lược đồ EER của các thực thể nhóm
- Có thể là tập thực thể, lớp con, siêu lớp, hoặc category
- Chú ý: định nghĩa *kiểu mối quan hệ* trong ER/EER nên có “tập thực thể” được thay thế bằng ‘class’ để cho phép các mối quan hệ giữa các lớp nói chung.

Các định nghĩa chính thức cho các Khái niệm Mô hình EER

- Lớp con S (Subclass S)

- Kế thừa tất cả các thuộc tính và mối quan hệ của một lớp C
- Tập hợp các thực thể phải luôn là 1 tập con của tập các thực thể của lớp C khác: $S \subseteq C$
- C được gọi là siêu lớp của S
- Một mối quan hệ siêu lớp/lớp con tồn tại giữa S và C

Các định nghĩa chính thức cho các Khái niệm Mô hình EER

- Chuyên biệt hóa Z (Specialization Z): $Z=\{S_1, S_2, \dots, S_n\}$ là một tập các lớp con có cùng siêu lớp G;
 - G/S_i là 1 mối quan hệ siêu lớp/lớp con, $i=1..n$
 - G được gọi là một tổng quát hóa của các lớp con $\{S_1, S_2, \dots, S_n\}$
 - Z là toàn phần nếu:
 - $S_1 \cup S_2 \cup \dots \cup S_n = G$;
 - Nếu không thì, Z là riêng phần
 - Z là phân ly nếu:
 - $S_i \cap S_j$ tập hợp rỗng khi $i \neq j$;
 - Nếu không thì, Z là chồng lấp

Các định nghĩa chính thức cho các Khái niệm Mô hình EER

- Tổng quát hóa (Generalization)
 - Kiểu thực thể được tổng quát hóa hoặc siêu lớp
- Lớp con S của C là vị ngữ-được định nghĩa nếu vị ngữ (điều kiện) p trên các thuộc tính của C được dùng để đặc tả tư cách thành viên trong S; ie $S=C[p]$, trong đó $C[p]$ là tập hợp các thực thể trong C thỏa mãn điều kiện p
- Một lớp con không được định nghĩa bởi 1 vị ngữ được gọi là người dùng – được định nghĩa

Các định nghĩa chính thức cho các Khái niệm Mô hình EER

- Category hoặc kiểu UNION T
 - Là lớp mà 1 tập hợp con của union các định nghĩa các siêu lớp D1, D2, ...Dn, n>1: $T \subseteq (D1 \cup D2 \cup \dots \cup Dn)$
 - Có thể có 1 vị ngữ pi trên các thuộc tính của Di để đặc tả các thực thể của Di là thành viên của T.
 - Nếu 1 vị ngữ được đặc tả trên mỗi Di: $T = (D1[p1] \cup D2[p2] \cup \dots \cup Dn[pn])$
- Kiểu mỗi quan hệ (Relationship type)
 - Bất kỳ lớp nào có thể tham gia vào một mỗi quan hệ

Phân tích và thiết kế hướng đối tượng

- Lập mô hình đối tượng – Các biểu đồ lớp
(Object Modeling – Class Diagrams)

Giới thiệu Objects và Classes

- **Đối tượng (Object)**: một thực thể với một vai trò được xác định rõ trong một miền ứng dụng; có trạng thái, hành vi và đặc điểm nhận diện.
- **Trạng thái (State)**: gồm các đặc tính (thuộc tính và các mối quan hệ) của một đối tượng và các giá trị của những đặc tính đó.
- **Hành vi (Behavior)**: biểu diễn làm thế nào một đối tượng hành động và phản ứng lại.
- **Định danh (Identity)**: tính duy nhất - không có hai đối tượng giống nhau.

Giới thiệu Objects và Classes (tt.)

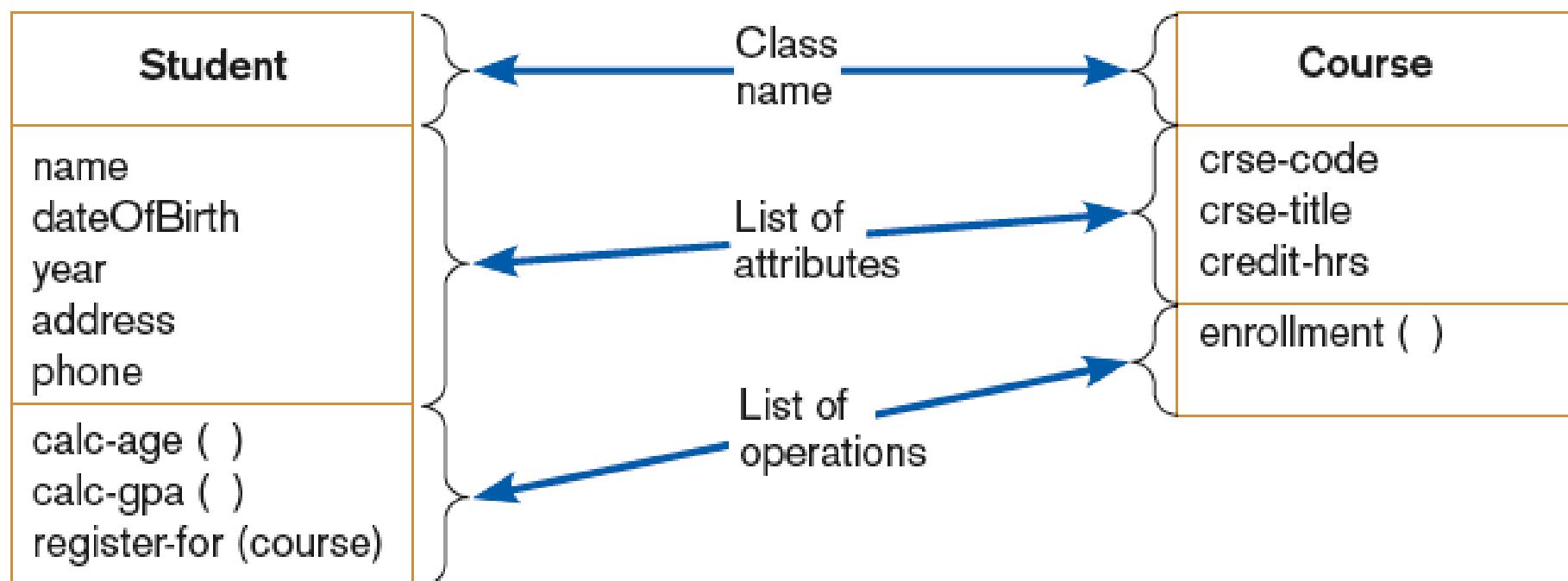
- **Lớp đối tượng/lớp (Object class/class)**: một nhóm logic các đối tượng giống nhau (hoặc tương tự) về thuộc tính, các mối quan hệ - và các hành vi (các phương thức)
- **Biểu đồ Lớp (Class diagram)**: một biểu đồ biểu diễn cấu trúc tĩnh của các lớp đối tượng, cấu trúc bên trong của các chúng, và các mối quan hệ chúng tham gia
- Các lớp UML tương tự các thực thể E-R.

Giới thiệu Objects và Classes (tt.)

- **Phương thức/Thao tác (Operation):** một hàm hoặc một dịch vụ được cung cấp qua tất cả các thể hiện của một lớp để đề cập hành vi của một đối tượng bằng cách truyền một thông điệp
- **Nguyên tắc bao đóng/bao gói (Encapsulation):** kỹ thuật ẩn các chi tiết thực thi bên trong của một đối tượng từ khung nhìn bên ngoài.

Giới thiệu Objects và Classes (tt.)

Hình 8-26 Biểu đồ lớp UML biểu diễn hai lớp



Các loại thao tác

- **Constructor:** một thao tác tạo một thể hiện mới của một lớp
- **Query:** một thao tác truy cập trạng thái của một đối tượng nhưng không thay đổi trạng thái
- **Update Operation:** một thao tác thay đổi trạng thái của một đối tượng.
- **Class Scope Operation:** một thao tác áp dụng cho một lớp hơn là cho một thể hiện đối tượng.

Giới thiệu Mối quan hệ

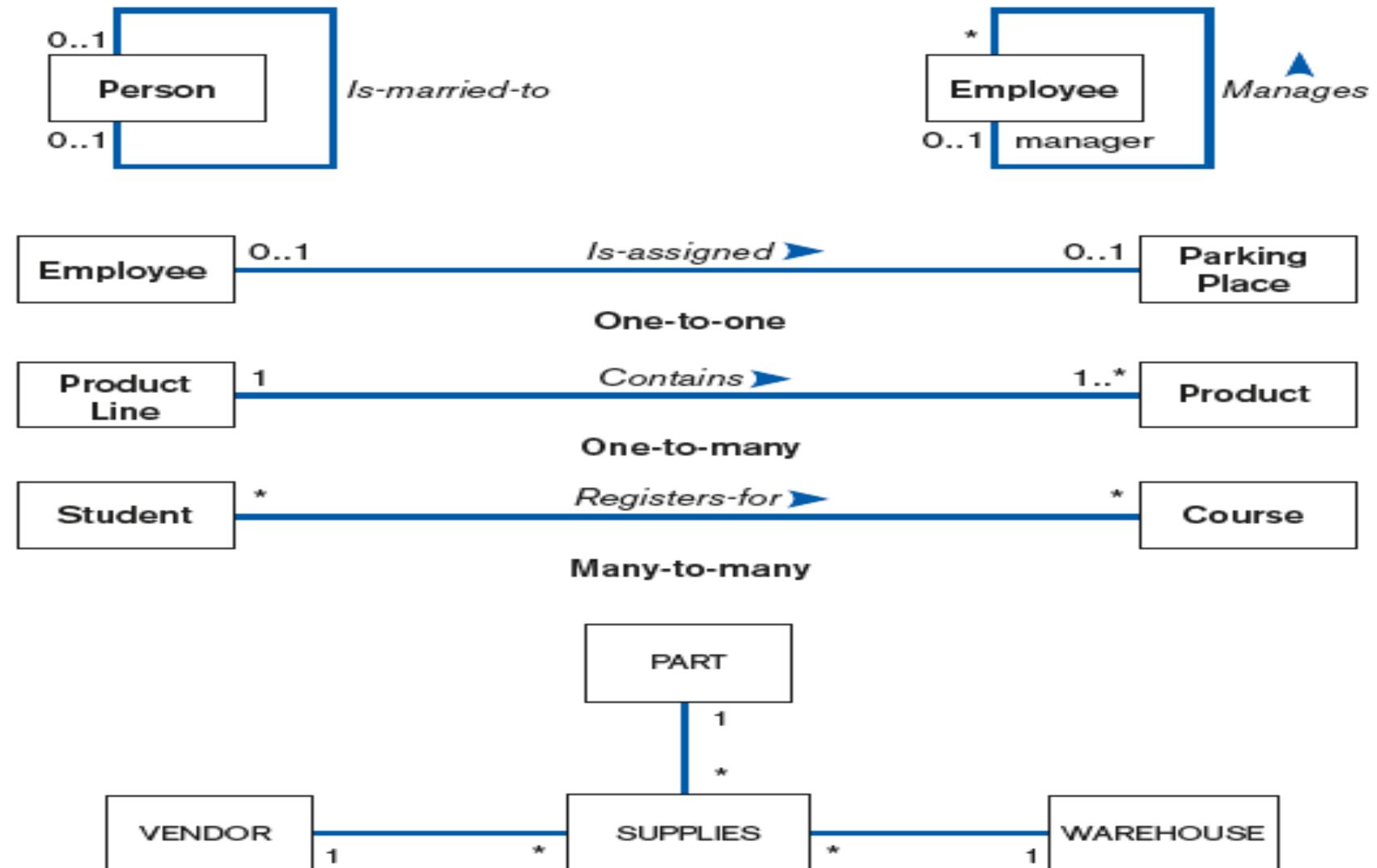
- **Mối quan hệ kết nối (Link)**: một link là một mối quan hệ giữa hai đối tượng.
- **Mối quan hệ kết hợp (Association)**: một mối quan hệ giữa các thể hiện của các lớp đối tượng.
- **Vai trò mối quan hệ kết hợp (Association role)**: tên gọi được đưa vào cuối của một mối quan hệ kết tập nơi liên kết với một lớp
- **Ràng buộc của kết hợp** → tên kết hợp (Association) hay tên vai trò (Association role)

Giới thiệu Mối quan hệ Associations(tt.)

- **Bản số (Multiplicity):** chỉ rõ có bao nhiêu đối tượng tham gia vào một mối quan hệ:
 - 1..1 được rút gọn là 1
 - Nếu không xác định được bản số thì mặc định là 1
 - 0..10 có nghĩa nhỏ nhất là 0 và lớn nhất là 10
 - 1, 2 có nghĩa có thể là 1 hoặc 2
 - 0..* được rút gọn là * có nghĩa bất kỳ giá trị nào

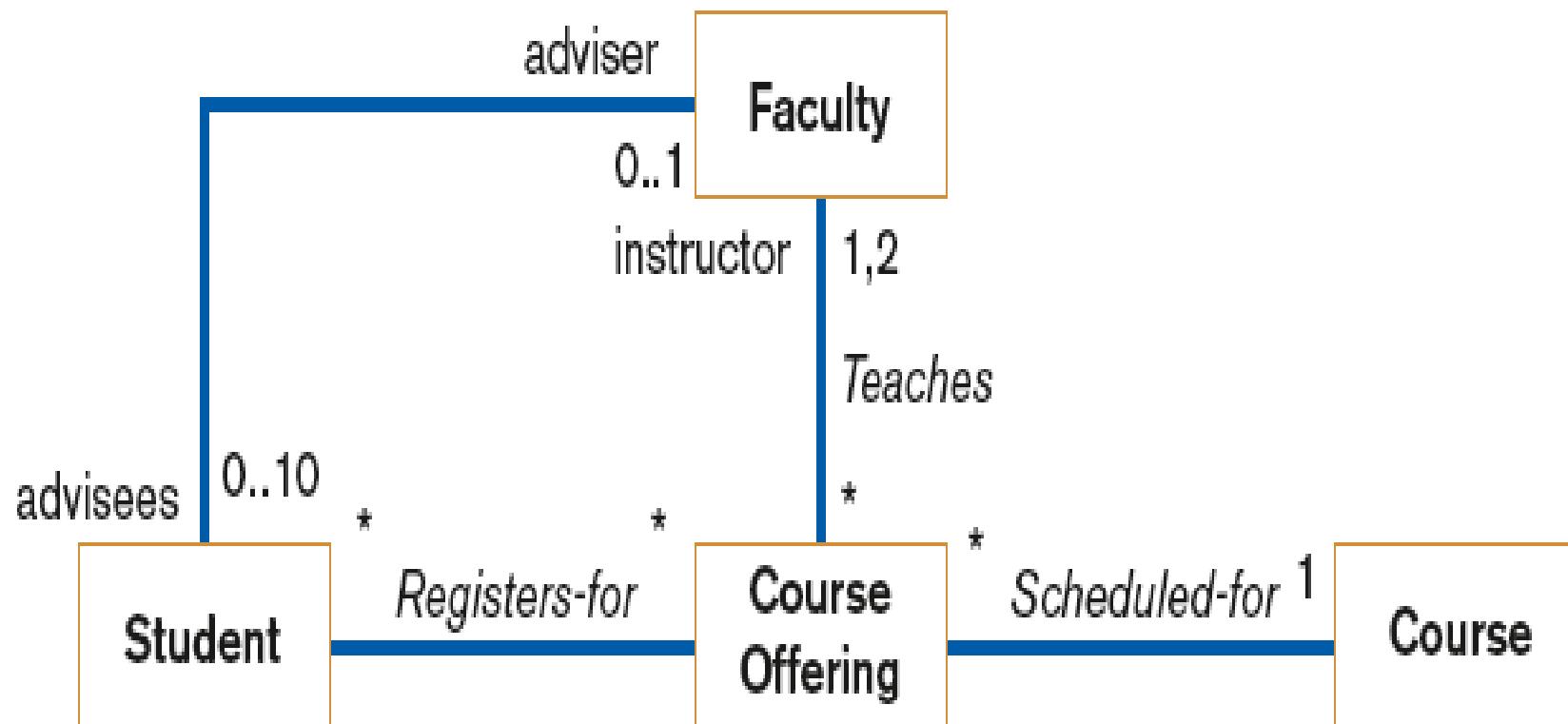
Giới thiệu Mối quan hệ Kết hợp - Associations(tt.)

Figure 8-27 Ví dụ về các mối quan hệ liên kết ở các bậc (degrees) khác nhau



Giới thiệu Mối quan hệ Kết hợp - Associations(tt.)

Hình 8-28 Các ví dụ của các mối quan hệ liên kết bậc 2 (binary associations)

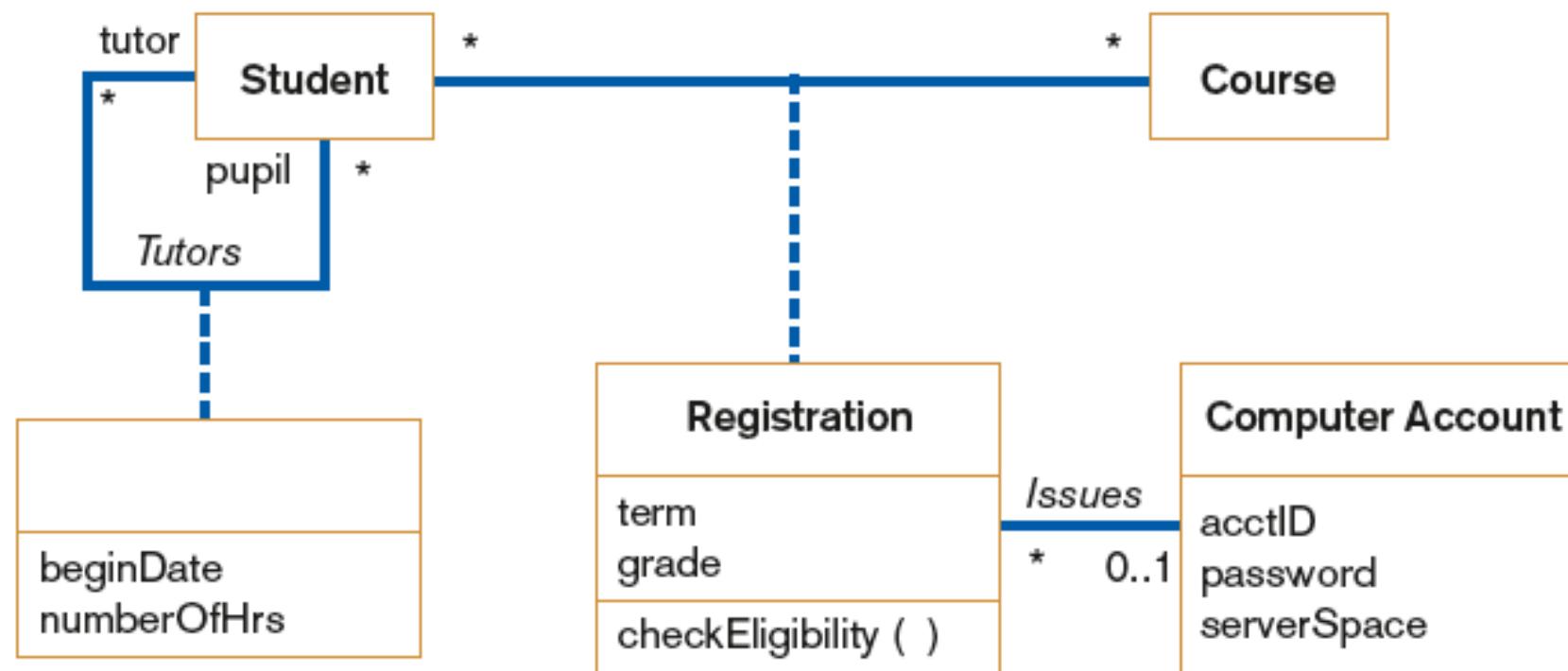


Giới thiệu Lớp liên kết - Associative Classes (tt.)

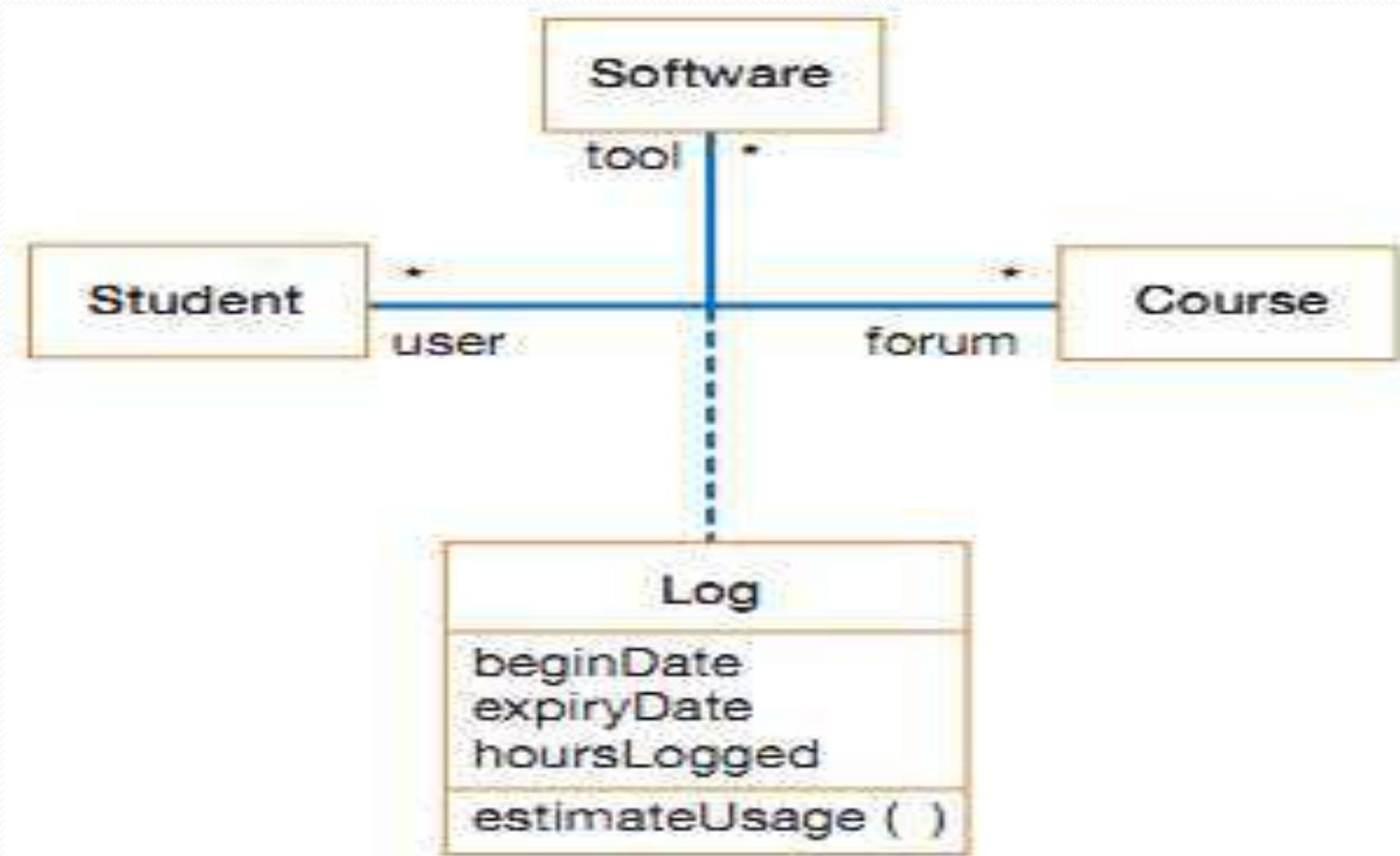
- **Lớp liên kết (Associative class):** một mối quan hệ kết tập có các thuộc tính hoặc phương thức của chính nó tham gia vào các mối quan hệ với các lớp khác.
- Các lớp liên kết UML tương tự các thực thể kết hợp E-R.
- Mỗi quan hệ Tổng quát hóa (*Generalization*) và kế thừa được thực hiện qua siêu lớp/các lớp con (*superclass/subclasses*) trong UML và qua các mối quan hệ siêu kiểu/kiểu con (*supertypes/subtypes*) trong E-R.

Giới thiệu Lớp liên kết - Associative Classes (tt.)

Hình 8-29 Biểu đồ lớp biểu diễn các lớp kết tập



Giới thiệu Lớp liên kết - Associative Classes (tt.)



Hình 8-30 Mối quan hệ
bậc 3 với các lớp kết tập

Giới thiệu các định khuôn cho các thuộc tính

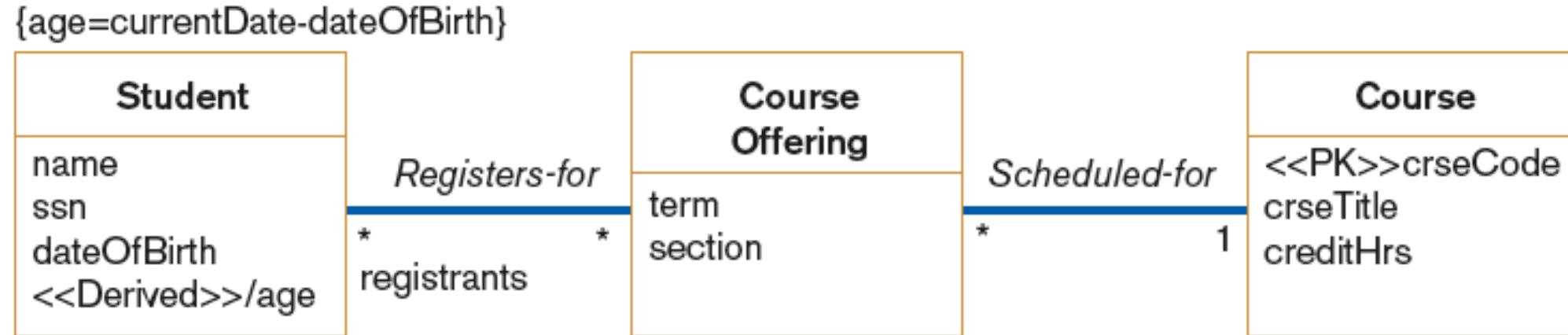


Figure 8-31 Các định khuôn (Stereotypes)

Giới thiệu quan hệ Tổng quát hóa - Generalization

- **Lớp trừu tượng (Abstract class)**: một lớp không có các thể hiện trực tiếp nhưng con cháu của nó (whose descendants) có thể có những thể hiện trực tiếp (direct instances).
- **Lớp cụ thể (Concrete class)**: một lớp có thể có các thể hiện trực tiếp.

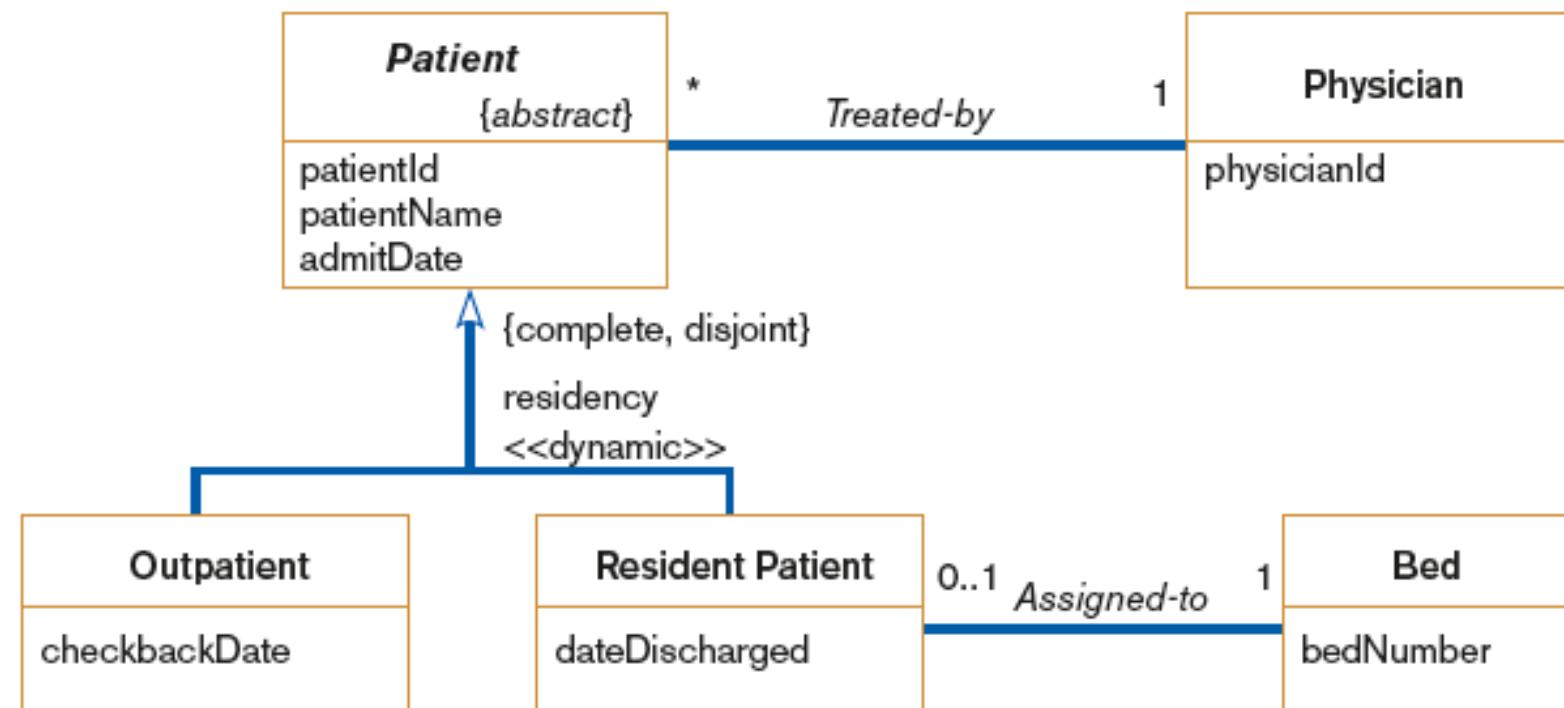
Giới thiệu quan hệ Tổng quát hóa – Generalization (tt.)

• Các từ khóa UML:

- *Chồng lắp (Overlapping)*: một hậu duệ (descendant) có thể từ nhiều hơn một trong các lớp con.
- *Phân ly (Disjoint)*: một hậu duệ (descendant) không thể là nhiều hơn một trong các lớp con.
- *Đầy đủ (Complete)*: tất cả lớp con đều đã được xác định (specified)
- *Không đầy đủ (Incomplete)*: một vài lớp con đã được xác định, nhưng danh sách đã biết là không đầy đủ.

Giới thiệu quan hệ Tổng quát hóa – Generalization (tt.)

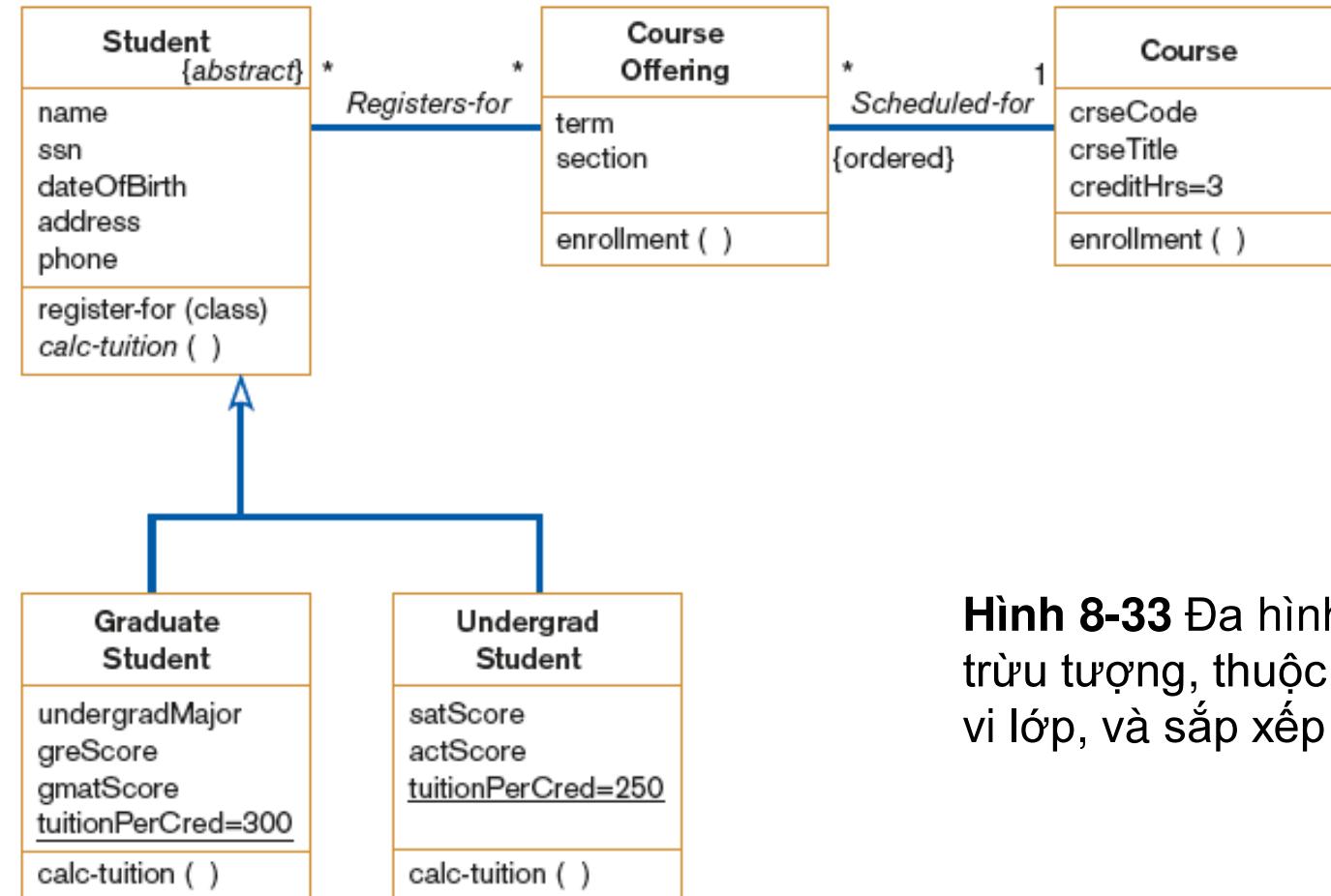
Figure 8-32 Ví dụ về tổng quát hóa, kế thừa và các ràng buộc



Giới thiệu quan hệ Tổng quát hóa – Generalization (tt.)

- **Thuộc tính phạm vi lớp (Class-scope attribute):** một thuộc tính của một lớp đặc tả một giá trị chung cho một lớp thực thể hơn là một giá trị cụ thể cho một thể hiện.

Giới thiệu quan hệ Tổng quát hóa – Generalization (tt.)



Hình 8-33 Đa hình, thao tác
trừu tượng, thuộc tính phạm
vi lớp, và sắp xếp

Giới thiệu quan hệ Tổng quát hóa – Generalization (tt.)

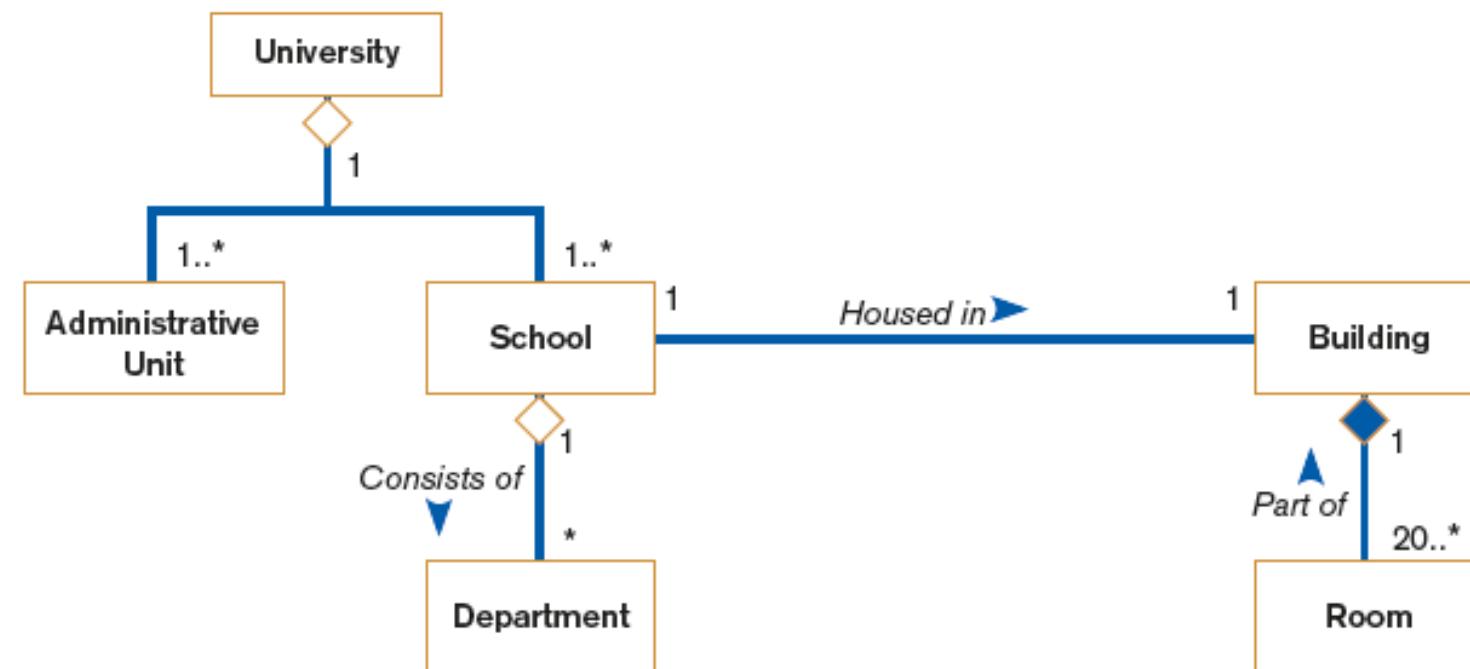
- **Thao tác trừu tượng (Abstract operation):** định nghĩa biểu mẫu (form) hoặc giao thức (protocol) của thao tác, chứ không phải sự thực thi
- **Phương thức (Method):** sự thực thi của thao tác
- **Đa hình (Polymorphism):** thao tác giống nhau có thể áp dụng cho hai hoặc nhiều lớp theo những cách khác nhau.

Giới thiệu quan hệ kết tập - Aggregation

- **Mối quan hệ Kết tập (Aggregation):** một mối quan hệ một phần giữa một đối tượng thành phần và một đối tượng gộp
 - Được biểu diễn bằng hình thoi rỗng
- **Mối quan hệ Kết cấu (Composition):** một đối tượng bộ phận phụ thuộc vào chỉ một đối tượng toàn thể và tồn tại và mất đi với đối tượng toàn thể đó.
 - Được biểu diễn bằng hình thoi đặc

Quan hệ kết tập và quan hệ kết cấu

Figure 8-34 Mối quan hệ kết tập và kết cấu



Một ví dụ của Lập mô hình quan niệm tại Hoosier Burger

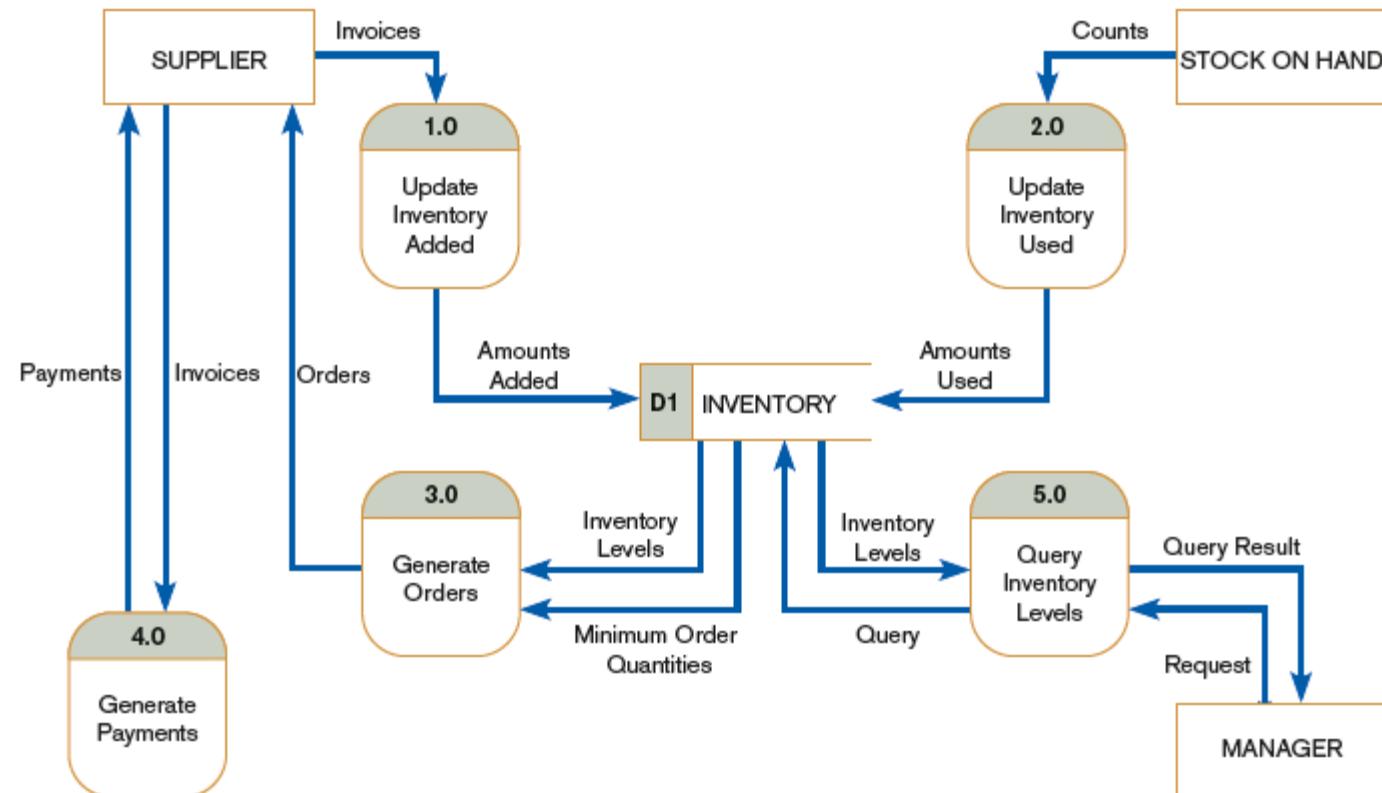


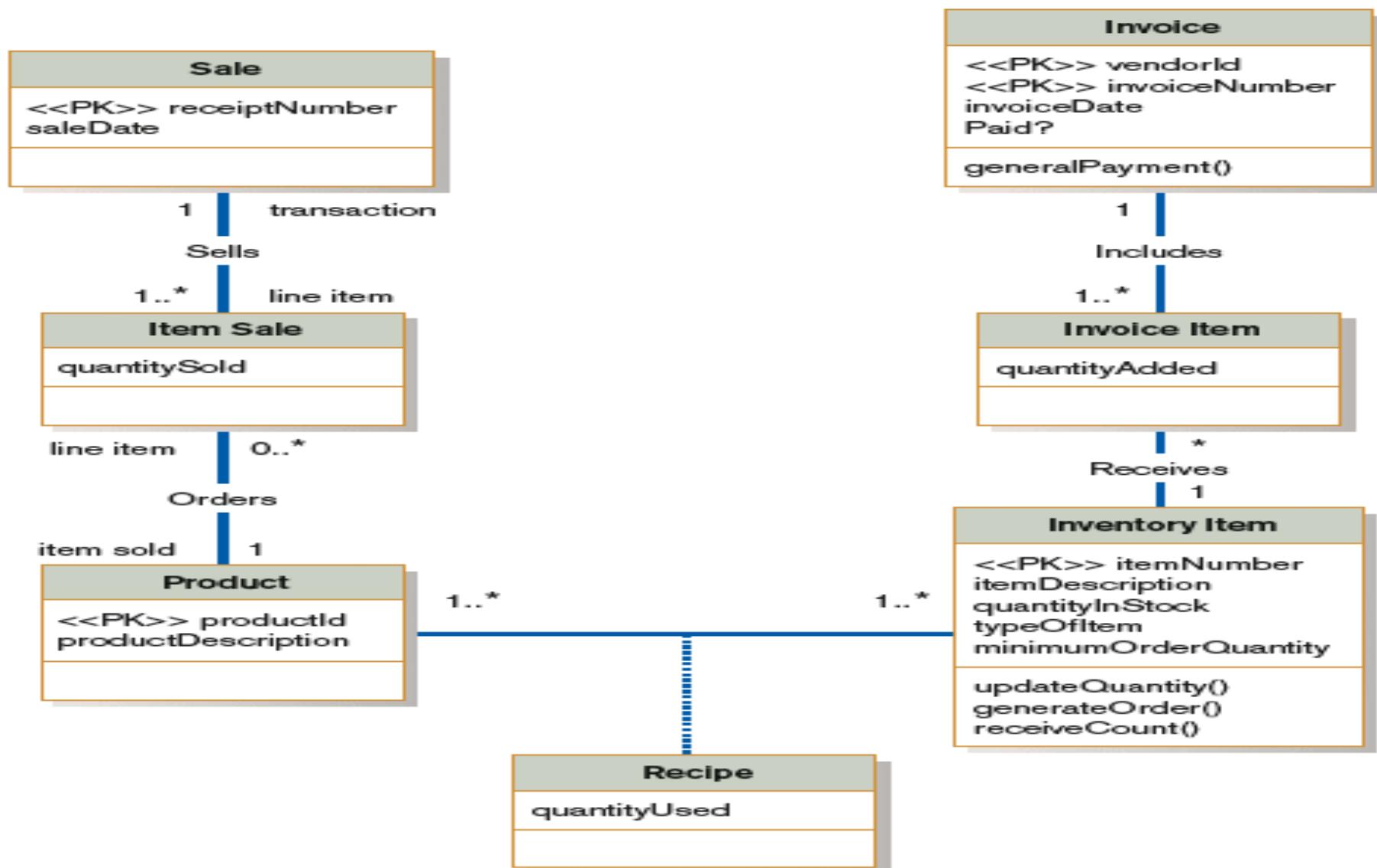
Figure 8-35 Sơ đồ luồng dữ liệu mức 0 của HT kiểm soát kho mới mức logic của Hoosier Burger

Một ví dụ của Lập mô hình quan niệm tại Hoosier Burger (tt.)

Conditions/ Courses of Action	Rules						
	1	2	3	4	5	6	7
Type of item	P	P	P	P	P	P	N
Time of week	D	W	D	W	D	W	-
Season of year	A	A	S	S	H	H	-
Standing daily order	X		X		X		
Standing weekend order		X		X		X	
Minimum order quantity							X
Holiday reduction					X	X	
Summer reduction			X	X			

Hình 8-36 Bảng quyết định đã tối giản cho việc sắp xếp lại kho hàng của Hoosier Burger

Một ví dụ của Lập mô hình quan niệm tại Hoosier Burger (tt.)



Hình 8-38 Biểu đồ lớp cuối cùng cho HT kiểm soát kho hàng của Hoosier Burger

Chương 4. Thiết kế các hệ thống

- 4.1 Thiết kế cơ sở dữ liệu
- 4.2 Thiết kế giao diện người dùng

Thiết kế CSDL

- Thiết kế tệp và CSDL xuất hiện theo 2 bước:
 - Phát triển một mô hình CSDL logic mô tả dữ liệu bằng cách dùng hệ thống kí hiệu tương ứng với một tổ chức dữ liệu được dùng bởi một HQTCSDL
 - Mô hình CSDL quan hệ
 - Đưa ra luật đặc tả kỹ thuật cho các tệp máy tính và CSDL để lưu trữ dữ liệu.
 - Thiết kế CSDL vật lý cung cấp cung cấp đặc tả kỹ thuật
- Thiết kế CSDL logic và vật lý song song với các bước thiết kế HT khác.

Quy trình thiết kế CSDL

- Bốn bước chính trong thiết kế và lập mô hình CSDL mức logic
 - 1. Phát triển một mô hình dữ liệu mức logic cho mỗi giao diện người dùng đã biết đối với ứng dụng bằng cách sử dụng các quy tắc chuẩn hóa
 - 2. Kết hợp các yêu cầu dữ liệu đã chuẩn hóa từ tất cả giao diện người dùng thành một mô hình CSDL logic hợp nhất (tích hợp khung nhìn).
 - 3. Chuyển mô hình dữ liệu ERD mức quan niệm cho ứng dụng thành các yêu cầu dữ liệu đã được chuẩn hóa.
 - 4. So sánh thiết kế CSDL logic hợp nhất với mô hình E-R đã chuyển đổi và tạo ra một mô hình CSDL logic cuối cùng cho ứng dụng

Quy trình thiết kế CSDL (tt)

- Những quyết định chính về thiết kế CSDL mức vật lý gồm:
 - Chọn một định dạng lưu trữ cho mỗi thuộc tính từ mô hình CSDL mức logic
 - Nhóm các thuộc tính từ mô hình CSDL mức logic thành các bản ghi vật lý.
 - Sắp xếp các bản ghi liên quan đến nhau vào bộ nhớ thứ cấp để các bản ghi có thể lưu trữ, truy cập và cập nhật nhanh chóng
 - Chọn lựa phương tiện và cấu trúc cho việc lưu trữ dữ liệu để truy cập hiệu quả

Các sản phẩm bàn giao và kết quả

- Thiết kế csdl logic
 - Phải tính đến mỗi yếu tố dữ liệu trên 1 đầu vào hoặc đầu ra hệ thống-form hoặc report-và trên mô hình E-R.
 - Các quan hệ được chuẩn hóa là sản phẩm bàn giao chính.
- Các quan hệ đã chuẩn hóa là những sản phẩm bàn giao chính
- Các kết quả thiết kế csdl vật lý
 - Chuyển các quan hệ thành các bảng CSDL
 - Các lập trình viên và phân tích viên CSDL viết mã lệnh những định nghĩa của CSDL
 - Được viết theo Structured Query Language (SQL)

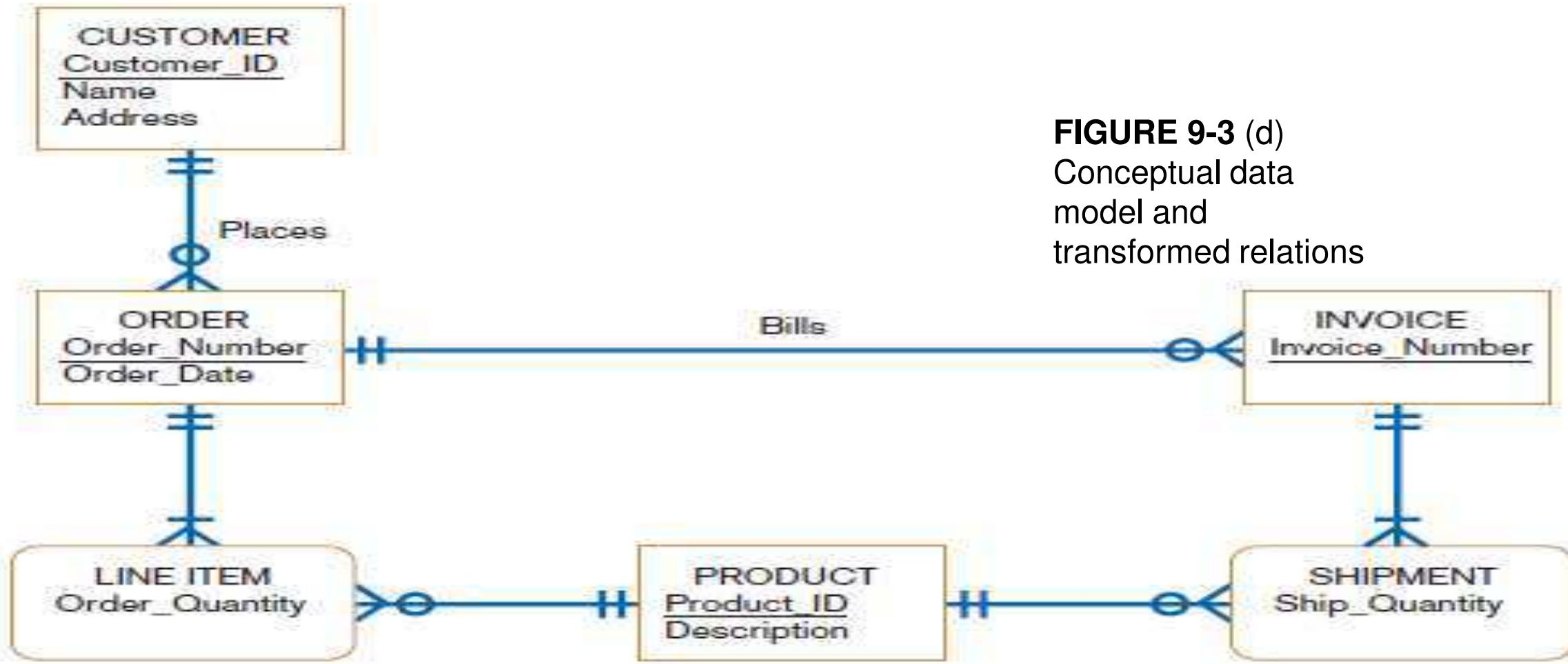


FIGURE 9-3 (d)
Conceptual data
model and
transformed relations

Relations:

```

CUSTOMER(Customer_ID,Name,Address)
PRODUCT(Product_ID,Description)
ORDER(Order_Number,Customer_ID,Order_Date)
LINE_ITEM(Order_Number,Product_ID,Order_Quantity)
INVOICE(Invoice_Number,Order_Number)
SHIPMENT(Invoice_Number,Product_ID,Ship_Quantity)

```

Mô hình CSDL quan hệ

- Nhắc lại kiến thức học phần **CƠ SỞ DỮ LIỆU**

Tài liệu tham khảo:

Nguyễn Đức Thuần, *Bài giảng Cơ sở dữ liệu*, Đại học Nha Trang

Mô hình CSDL quan hệ

- Nhắc lại:
 - Mô hình CSDL quan hệ
 - Quan hệ
 - Khóa chính
 - Chuẩn hóa (Dạng chuẩn 1-2-3, Phụ thuộc hàm)
 - Khóa ngoại và toàn vẹn tham chiếu

Dạng chuẩn 2 (2NF)

- Chuyển quan hệ thành dạng chuẩn 2 → Ví dụ
 - Quan hệ EMPLOYEE2 (1NF nhưng không là 2NF)
 - EMPLOYEE2(Emp_ID, Name, Dept, Salary, Course, Date_Completed)

EMPLOYEE2

Emp_ID	Name	Dept	Salary	Course	Date_Completed
100	Margaret Simpson	Marketing	42,000	SPSS	6/19/2012
100	Margaret Simpson	Marketing	42,000	Surveys	10/7/2012
140	Alan Beeton	Accounting	39,000	Tax Acc	12/8/2012
110	Chris Lucero	Info Systems	41,500	SPSS	1/12/2012
110	Chris Lucero	Info Systems	41,500	C++	4/22/2012
190	Lorenzo Davis	Finance	38,000	Investments	5/7/2012
150	Susan Martin	Marketing	38,500	SPSS	6/19/2012
150	Susan Martin	Marketing	38,500	TQM	8/12/2012

- Phụ thuộc hàm trong quan hệ này là

- $\text{Emp_ID} \rightarrow \text{Name, Dept, Salary}$
- $\text{Emp_ID, Course} \rightarrow \text{Date_Completed}$

- Chuyển về dạng chuẩn 2

- EMPLOYEE1(Emp_ID, Name, Dept, Salary)
- EMP COURSE(Emp_ID, Course, Date_Completed)

EMPLOYEE1

Emp_ID	Name	Dept	Salary
100	Margaret Simpson	Marketing	42,000
140	Allen Beeton	Accounting	39,000
110	Chris Lucero	Info Systems	41,500
190	Lorenzo Davis	Finance	38,000
150	Susan Martin	Marketing	38,500

EMP COURSE

Emp_ID	Course	Date_Completed
100	SPSS	6/19/2012
100	Surveys	10/7/2012
140	Tax Acc	12/8/2012
110	SPSS	1/22/2012
110	C++	4/22/2012
190	Investments	5/7/2012
150	SPSS	6/19/2012
150	TQM	8/12/2012

Dạng chuẩn 3 (3NF)

- Chuyển về dạng chuẩn 3 (3NF) → Ví dụ
 - quan hệ
 - SALES (Customer_ID, Customer_Name, Salesperson, Region) → Thỏa mãn chuẩn 2
 - Phụ thuộc hàm
 - Customer_ID → Customer_Name, Salesperson, Region
 - Salesperson → Region
- Chuyển về dạng chuẩn 3
 - SALES1(Customer_ID, Customer_Name, Salesperson)
 - SPERSON(Salesperson, Region)

Customer_ID	Customer_Name	Salesperson	Region
8023	Anderson	Smith	South
9167	Bancroft	Hicks	West
7924	Hobbs	Smith	South
6837	Tucker	Hernandez	East
8596	Eckersley	Hicks	West
7018	Arnold	Faulb	North

SALES1

Customer_ID	Customer_Name	Salesperson
8023	Anderson	Smith
9167	Bancroft	Hicks
7924	Hobbs	Smith
6837	Tucker	Hernandez
8596	Eckersley	Hicks
7018	Arnold	Faulb

SPERSON

Salesperson	Region
Smith	South
Hicks	West
Hernandez	East
Faulb	North

Ví dụ khóa ngoại và tham chiếu toàn vẹn

SALES1

<u>Customer_ID</u>	<u>Customer_Name</u>	<u>Salesperson</u>
8023	Anderson	Smith
9167	Bancroft	Hicks
7924	Hobbs	Smith
6837	Tucker	Hernandez
8596	Eckersley	Hicks
7018	Arnold	Faulb

SPERSON

<u>Salesperson</u>	<u>Region</u>
Smith	South
Hicks	West
Hernandez	East
Faulb	North

B

- Khóa ngoại thiết lập kết hợp một-nhiều giữa SPERSON (1) và SALES1 (nhiều)
- Không có giá trị nào của SalesPerson trong SALES1 mà nó không tồn tại trong SPERSON (tham chiếu toàn vẹn)

XÂY DỰNG CSDL MỨC LOGIC

- Bước 1: Phát triển mô hình dữ liệu logic cho mỗi GUI
- Bước 2: Xây dựng mô hình dữ liệu logic hợp nhất
- Bước 3: Chuyển ERD thành mô hình dữ liệu logic
- Bước 4: So sánh và hợp nhất

B1: Phát triển mô hình dữ liệu logic cho mỗi GUI

Màn hình này biểu diễn khách hàng với tổng lượng hàng mua lớn nhất của 1 sản phẩm cụ thể trong 1 khoảng thời gian nào đó.

Các quan hệ:

CUSTOMER(Customer_ID, Name)

ORDER(Order_Number, Customer_ID, Order_Date)

PRODUCT(Product_ID)

LINE ITEM(Order_Number, Product_ID, Order_Quantity)

A

HIGHEST VOLUME CUSTOMER	
ENTER PRODUCT ID.:	M128
START DATE:	11/01/2012
END DATE:	12/31/2012

CUSTOMER ID.:	1256
NAME:	Commonwealth Builder
VOLUME:	30

B1: Phát triển mô hình dữ liệu logic cho mỗi GUI

Báo cáo này biểu diễn số lượng mỗi sản phẩm đã được đặt hàng ít hơn lượng được chuyển hàng trong ngày cụ thể

Các quan hệ:

PRODUCT(Product_ID)

LINE ITEM(Product_ID,Order_Number,Order_Quantity)

ORDER(Order_Number,Order_Date)

SHIPMENT(Product_ID,Invoice_Number,Ship_Quantity)

INVOICE(Invoice_Number,Invoice_Date,Order_Number)

PAGE 1

BACKLOG SUMMARY REPORT
11/30/2012

<u>PRODUCT ID</u>	<u>BACKLOG QUANTITY</u>
B381	0
B975	0
B985	6
E125	30
⋮	⋮
M128	2

B2: Xây dựng mô hình dữ liệu logic hợp nhất

CUSTOMER(Customer_ID,Name)

PRODUCT(Product_ID)

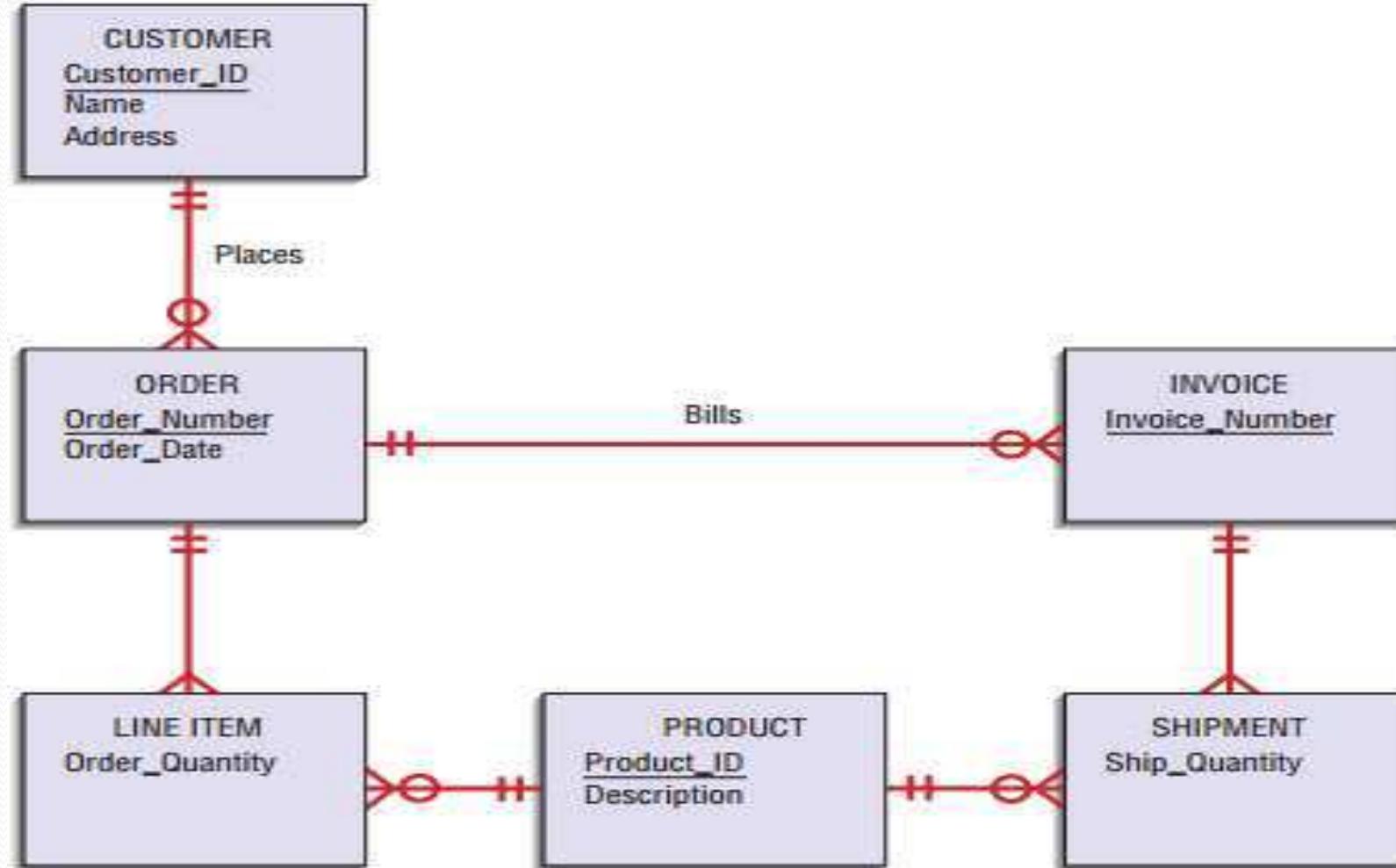
INVOICE(Invoice_Number,Invoice_Date,Order_Number)

ORDER(Order_Number,Customer_ID,Order_Date)

LINE ITEM(Order_Number,Product_ID,Order_Quantity)

SHIPMENT(Product_ID,Invoice_Number,Ship_Quantity)

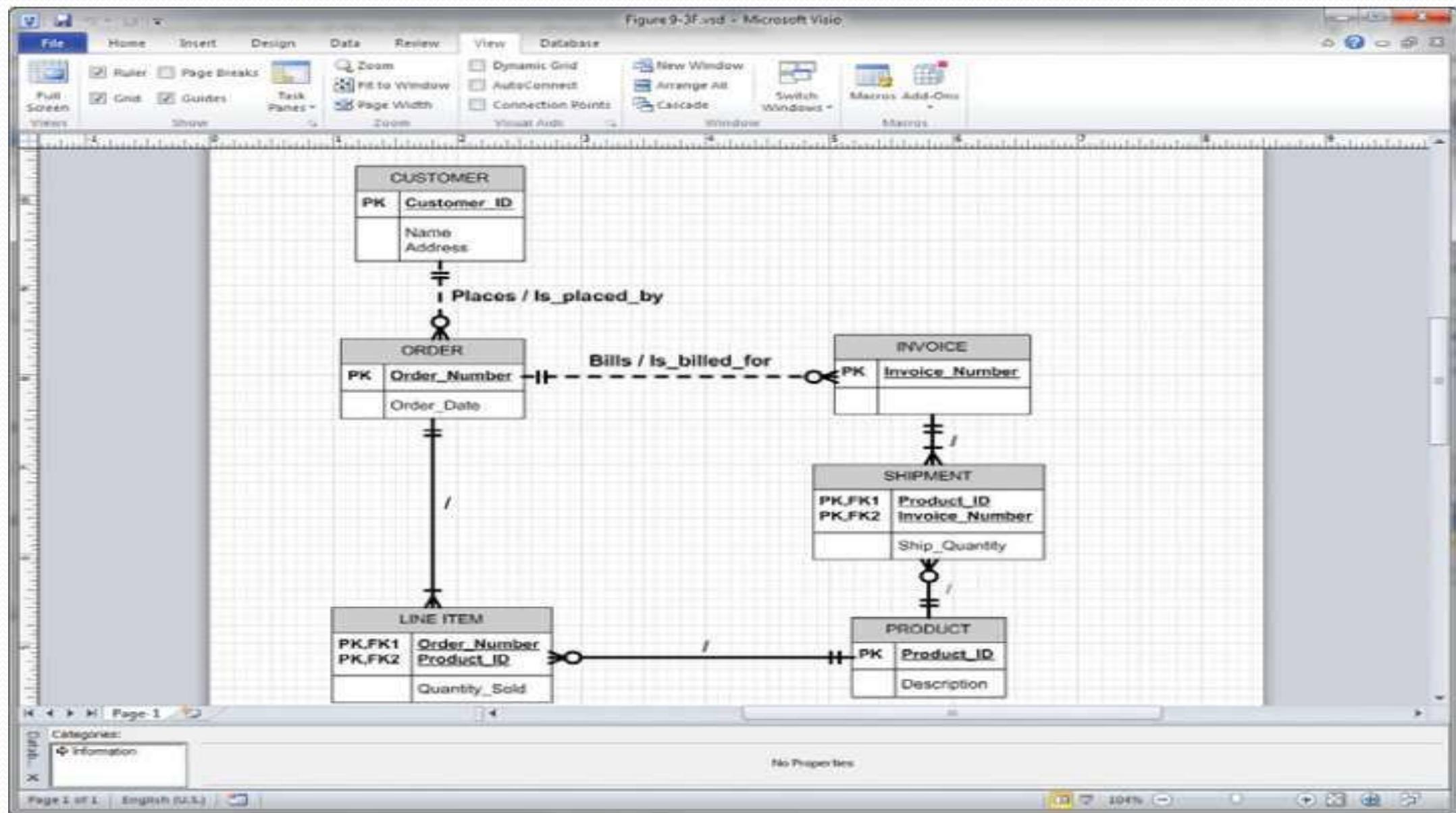
B3: Chuyển ERD thành mô hình dữ liệu logic



E

CUSTOMER(Customer_ID, Name, Address)
PRODUCT(Product_ID, Description)
ORDER(Order_Number, Customer_ID, Order_Date)
LINE ITEM(Order_Number, Product_ID, Order_Quantity)
INVOICE(Invoice_Number, Order_Number, Invoice_Date)
SHIPMENT(Invoice_Number, Product_ID, Ship_Quantity)

B4: So sánh và hợp nhất



Chuyển ERD thành các quan hệ (*hay mô hình DL logic*)

- Có ích để chuyển mô hình quan niệm dữ liệu thành 1 tập các quan hệ đã được chuẩn hóa
- Các bước
 - I. Ánh xạ ER- & EER- thành Quan hệ
 - Biểu diễn thực thể
 - Biểu diễn liên kết
 - Biểu diễn thuộc tính đa trị
 - II. Chuẩn hóa các quan hệ
 - III. Hợp nhất các quan hệ

I. Ánh xạ ER- & EER- thành Quan hệ

- ER-
 - B1: ánh xạ các tập thực thể thông thường
 - B2: ánh xạ các tập thực thể yếu
 - B3: ánh xạ các kiểu mối quan hệ nhị phân 1:1
 - B4: ánh xạ các kiểu mối quan hệ nhị phân 1:N
 - B5: ánh xạ các kiểu mối quan hệ nhị phân M:N
 - B6: ánh xạ các thuộc tính đa trị
 - B7: ánh xạ các kiểu mối quan hệ bậc n (>2)
- EER-
 - B8: các tùy chọn cho việc ánh xạ chuyên biệt hóa hoặc tổng quát hóa
 - B9: ánh xạ kiểu Union (Categories)

I. Ánh xạ ER thành Quan hệ

- Ánh xạ các tập thực thể thông thường (mạnh)
 - Thực thể → Quan hệ
 - Tất cả các thuộc tính của thực thể → các thuộc tính của quan hệ
 - Định danh của tập thực thể → khóa chính của quan hệ
 - Khóa chính phải thỏa mãn 2 điều kiện sau
 - A. Giá trị của khóa phải xác định duy nhất mỗi dòng trong quan hệ
 - B. Khóa không nên dư thừa
 - Nhãn của tập thực thể được chuyển thành tên quan hệ.

I. Ánh xạ ER thành Quan hệ (tt)

- Chuyển các tập thực thể thông thường (mạnh)

A

CUSTOMER

<u>Customer_ID</u>	Name	Address	City_State_Zip	Discount
1273	Contemporary Designs	123 Oak St.	Austin, TX 28384	5%
6390	Casual Corner	18 Hoosier Dr.	Bloomington, IN 45821	3%

FIGURE 9-10
Transforming an entity type to a relation:
(A) E-R diagram;
(B) Relation.

I. Ánh xạ ER thành Quan hệ (tt)

- Thực thể yếu (weak entity)
 - Với mỗi tập thực thể yếu W trong sơ đồ ER với tập thực thể chủ nhân E, tạo 1 quan hệ R và gồm tất cả các thuộc tính đơn của W là các thuộc tính của R, và các thuộc tính khóa chính của các quan hệ tương ứng các tập thực thể chủ nhân;
 - Khóa chính của R là sự kết hợp của các khóa chính chủ nhân E và khóa thành phần của tập thực thể yếu W, nếu có.

I. Ánh xạ ER thành Quan hệ (tt)

- Thực thể yếu (weak entity)
 - Vd, tạo quan hệ DEPENDENT tương ứng tập thực thể yếu DEPENDENT.

DEPENDENT				
Essn	Dependent_name	Sex	Bdate	Relationship

I. Ánh xạ ER thành Quan hệ (tt)

• **Liên kết bậc 2 (2 ngôi) 1:N**

- Thủ tục biểu diễn các liên kết phụ thuộc vào bậc của liên kết – bậc 1, bậc 2, bậc 3 – và bản số của mỗi quan hệ
- Liên kết bậc 2 (1:N) được biểu diễn bằng cách thêm thuộc tính khóa chính (hoặc các thuộc tính) của thực thể ở liên kết bên 1 thành khóa ngoại trong quan hệ của liên kết bên nhiều (N).

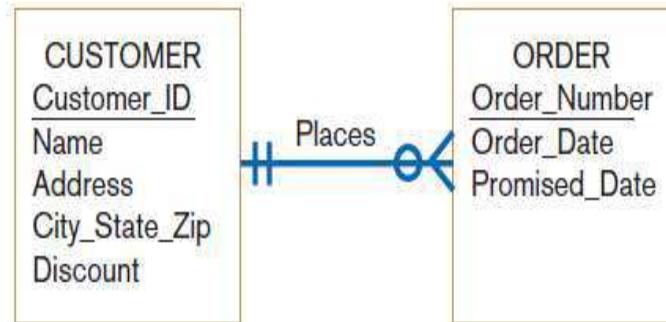


FIGURE 9-11
Representing a 1:N relationship
(a) E-R diagram

CUSTOMER

<u>Customer_ID</u>	Name	Address	City_State_ZIP	Discount
1273	Contemporary Designs	123 Oak St.	Austin, TX 28384	5%
6390	Casual Corner	18 Hoosier Dr.	Bloomington, IN 45821	3%

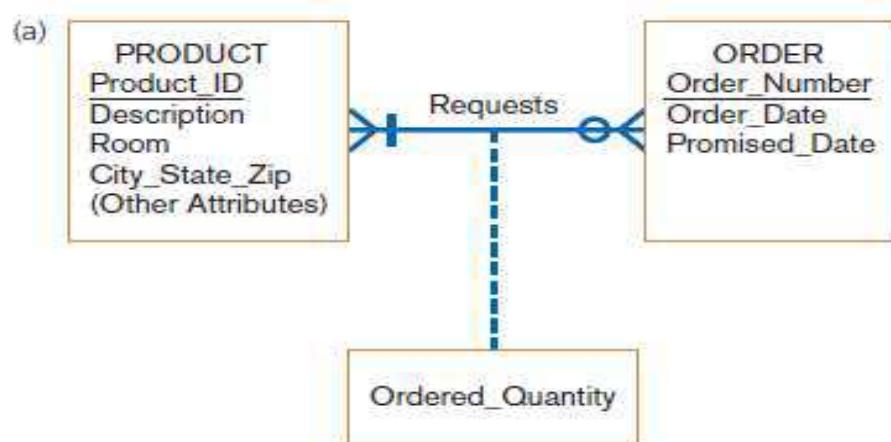
ORDER

<u>Order_Number</u>	<u>Order_Date</u>	<u>Promised_Date</u>	<u>Customer_ID</u>
57194	3/15/1X	3/28/1X	6390
63725	3/17/1X	4/01/1X	1273
80149	3/14/1X	3/24/1X	6390

(b) Relations

I. Ánh xạ ER thành Quan hệ (tt)

- **Liên kết bậc 2 hoặc bậc 1 (1:1)** được biểu diễn bởi bất kỳ lựa chọn nào sau đây:
 - Thêm khóa chính của A như một khóa ngoại của B
 - Thêm khóa chính của B như một khóa ngoại của A
 - Cả 2 cách trên.
- **Liên kết bậc 2 hoặc bậc cao hơn (M:N)**
 - Tạo quan hệ khác và chứa các khóa chính của tất cả quan hệ như khóa chính của quan hệ mới.



(b) ORDER

<u>Order_Number</u>	<u>Order_Date</u>	<u>Promised_Date</u>
61384	2/17/2014	3/01/2014
62009	2/13/2014	2/27/2014
62807	2/15/2014	3/01/2014

ORDER LINE

<u>Order_Number</u>	<u>Product_ID</u>	<u>Quantity_Ordered</u>
61384	M128	2
61384	A261	1

PRODUCT

<u>Product_ID</u>	<u>Description</u>	<u>Room</u>	<u>(Other Attributes)</u>
M128	Bookcase	Study	-
A261	Wall unit	Family	-
R149	Cabinet	Study	-

FIGURE 9-12

Representing an M:N relationship

(a) E-R diagram

(b) Relations

I. Ánh xạ ER thành Quan hệ (tt)

- Liên kết bậc 2 hoặc bậc cao hơn (M-N)
 - Đôi khi quan hệ mới được tạo từ một liên kết M:N chứa thêm **một khóa chính** khác 2 khóa chính của 2 quan hệ có liên quan

- Vd



- Date là một phần của khóa cho quan hệ mới SHIPMENT như sau:

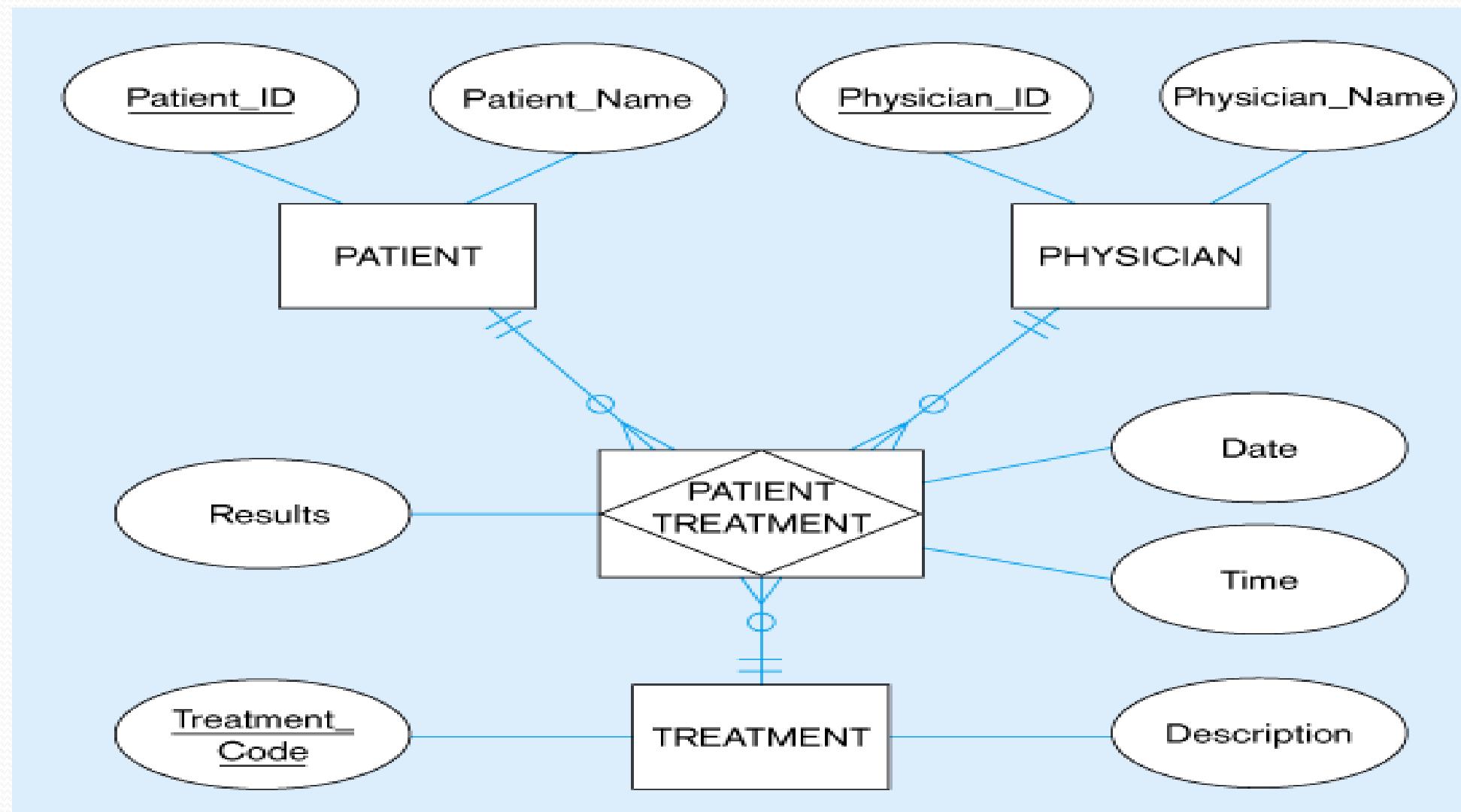
SHIPMENT(Customer_ID, Vendor_ID, Date, Amount)

- Nếu mỗi shipment có một khóa tách rời là một mã số shipment (Shipment_Number) thì Date trở thành một thuộc tính không khóa và Customer_ID và Vendor_ID trở thành các khóa ngoại, như sau:

SHIPMENT(Shipment_Number, Customer_ID, Vendor_ID, Date, Amount)

I. Ánh xạ ER thành Quan hệ (tt)

- Vd: ánh xạ các kiểu mối quan hệ bậc n (>2)



I. Ánh xạ ER thành Quan hệ (tt)

- Vd: ánh xạ các kiểu mối quan hệ bậc n (>2)

PATIENT

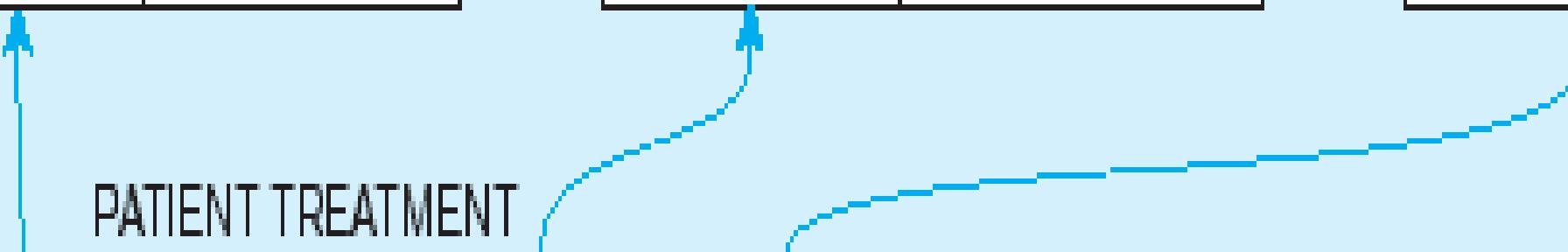
Patient_ID	Patient_Name

PHYSICIAN

Physician_ID	Physician_Name

TREATMENT

Treatment_Code	Description



Patient_ID	Physician_ID	Treatment_Code	Treatment_Date	Treatment_Time	Results

Vì bộ khóa chính gồm 5 thuộc tính này khá cồng kềnh
→ tốt hơn nên tạo một khóa thay thế là Treatment_ID

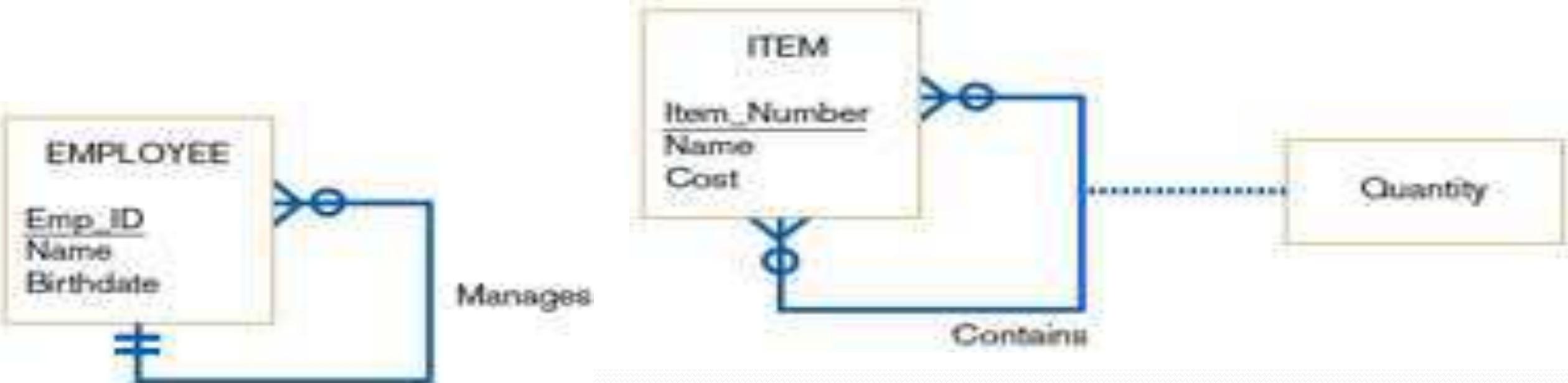
I. Ánh xạ ER thành Quan hệ (tt)

- **Liên kết bậc 1 (1:N)**

- Liên kết giữa các cá thể của 1 tập thực thể đơn
- Sử dụng 1 khóa ngoại đệ quy
 - 1 khóa ngoại trong 1 quan hệ mà nó tham chiếu đến các giá trị của khóa chính trong cùng quan hệ đó.

- **Các liên kết bậc 1 (M:N)**

- Tạo quan hệ tách rời để biểu diễn liên kết M:N
- Khóa chính của quan hệ mới là một tập hợp 2 thuộc tính mà chúng lấy các giá trị từ cùng một khóa chính.
- Bất kỳ thuộc tính kết hợp với liên kết được đưa vào quan hệ mới này như một thuộc tính không khóa.



Hình 9-13
Hai liên kết bậc 1

EMPLOYEE(Emp_ID, Name, Birthdate, Manager_ID)

(a) EMPLOYEE with
Managers
relationship (1:N)

ITEM(Item_Number, Name, Cost)

ITEM-BILL(Item_Number, Component_Number, Quantity)

(b) Bill-of-materials
structure (M:N)

Cấu trúc E-R

Thực thể thông thường (mạnh)

Thực thể yếu

Liên kết bậc 2 hoặc bậc 1 (1:1)

Liên kết bậc 2 (1:N)

Thực thể kết hợp hoặc liên kết bậc 1 hoặc bậc 2 (M:N)

Thực thể kết hợp với các khóa thêm vào hoặc liên kết bậc 2 hoặc bậc 1 (M:N)

Thực thể kết hợp với khóa của chính nó hoặc liên kết bậc 2 hoặc bậc 1 (M:N)

Biểu diễn quan hệ

Tạo 1 quan hệ với khóa chính và các thuộc tính không khóa

Tạo một quan hệ với một khóa chính phức hợp (gồm khóa chính của thực thể mà thực thể yếu phụ thuộc) và các thuộc tính không khóa.

Đặt khóa chính của mỗi thực thể trong quan hệ vào thực thể còn lại hoặc đặt khóa chính vào cả hai thực thể

Đặt khóa chính của thực thể bên một của mỗi quan hệ vào làm một khóa ngoại trong quan hệ của thực thể bên nhiều (N)

Tạo một quan hệ với một khóa chính phức hợp gồm các khóa chính của các thực thể liên quan, cộng thêm bất kỳ các thuộc tính không khóa nào của liên kết hoặc thực thể kết hợp

Tạo một quan hệ với một khóa chính phức hợp gồm các khóa chính của các thực thể liên quan và các thuộc tính khóa chính thêm vào liên kết hoặc thực thể kết hợp, cộng thêm bất kỳ thuộc tính không khóa của liên kết hoặc thực thể kết hợp

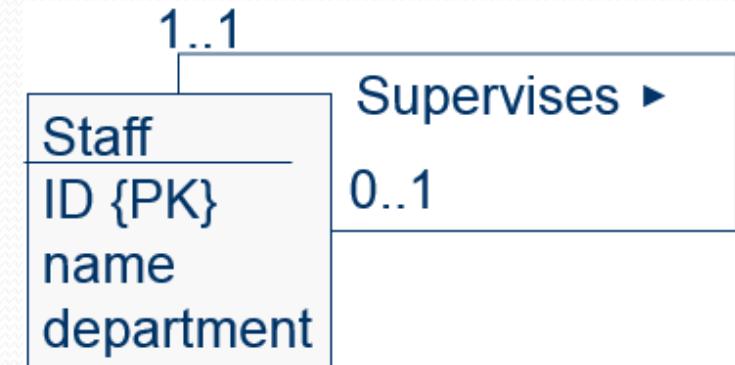
Tạo một quan hệ với khóa chính của liên kết hoặc thực thể kết hợp, cộng thêm bất kỳ thuộc tính không khóa nào của liên kết hoặc thực thể kết hợp và các khóa chính của các thực thể liên quan (như các thuộc tính không khóa)

I. Ánh xạ ER thành Quan hệ (tt)

- Chuyển đổi các mối quan hệ bậc 1-đệ quy 1:1
 - Mỗi quan hệ giữa các thể hiện khác nhau của cùng thực thể
 - Tên của các thuộc tính khóa cần được thay đổi để phản ánh vai trò của mỗi thực thể trong mối quan hệ
 - Tư cách thành viên tham gia bắt buộc cả 2 bên (**đã trình bày**)
 - Giữ quan hệ đơn
 - Thêm thuộc tính mới – “bản sao” của PK làm FK
 - **Tùy chọn cả 2 bên**
 - Tạo thêm 1 quan hệ mới với 2 bản sao chép của PK
 - Chúng là PK phức hợp và tách biệt như các FK để lk lại thực thể
 - **Tùy chọn 1 bên**
 - Theo 1 trong 2 pp trên

I. Ánh xạ ER thành Quan hệ (tt)

- Chuyển đổi các mối quan hệ bậc 1-đệ quy 1:1
 - Vd: mỗi thành viên staff có 1 supervisor và có thể giám sát 1 thành viên staff → tùy chọn 1 bên



C1. tạo 1 quan hệ

Staff(ID, supervisorID*, name, department)

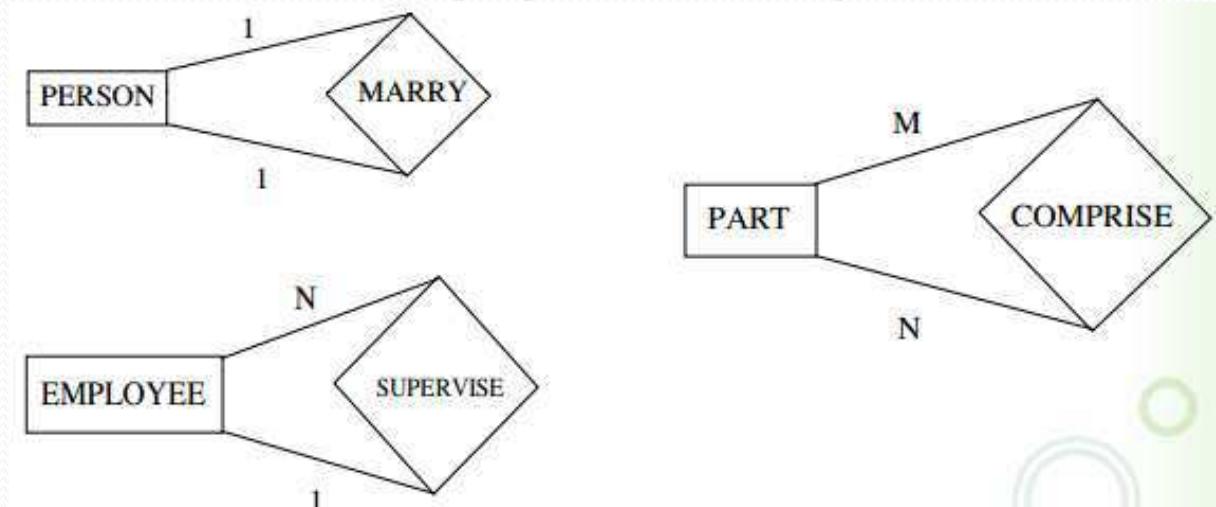
C2. tạo 2 quan hệ đơn

Staff(ID, name, department)

Supervision(staffID*, supervisorID*)

I. Ánh xạ ER thành Quan hệ (tt)

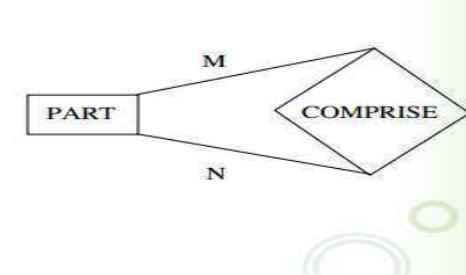
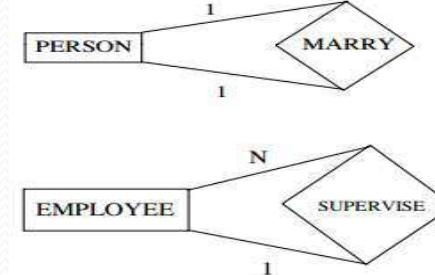
- Chuyển đổi các mối quan hệ bậc 1-đệ quy/đối hợp



- Vd1: Mỗi quan hệ 1:1, trong đó các tư cách thành viên cho cả 2 thực thể là tùy chọn

PERSON (ID, NAME, ADDRESS,...)

MARRY (HUSBAND-ID, WIFE-ID, DATE_OF_MARRIAGE)



I. Ánh xạ ER thành Quan hệ (tt)

- Chuyển đổi các mối quan hệ bậc 1-đệ quy/đối hợp

Vd2: Mối quan hệ bậc 1 (1:N)

- Nếu mối quan hệ là bắt buộc hoặc hầu như bắt buộc (**đã trình bày**):

EMPLOYEE (ID, ENAME,..., **SUPERVISOR_ID**)

- Nếu mối quan hệ là tùy chọn

EMPLOYEE (ID, ENAME,...)

SUPERVISE (ID, START_DATE,..., **SUPERVISOR_ID**)

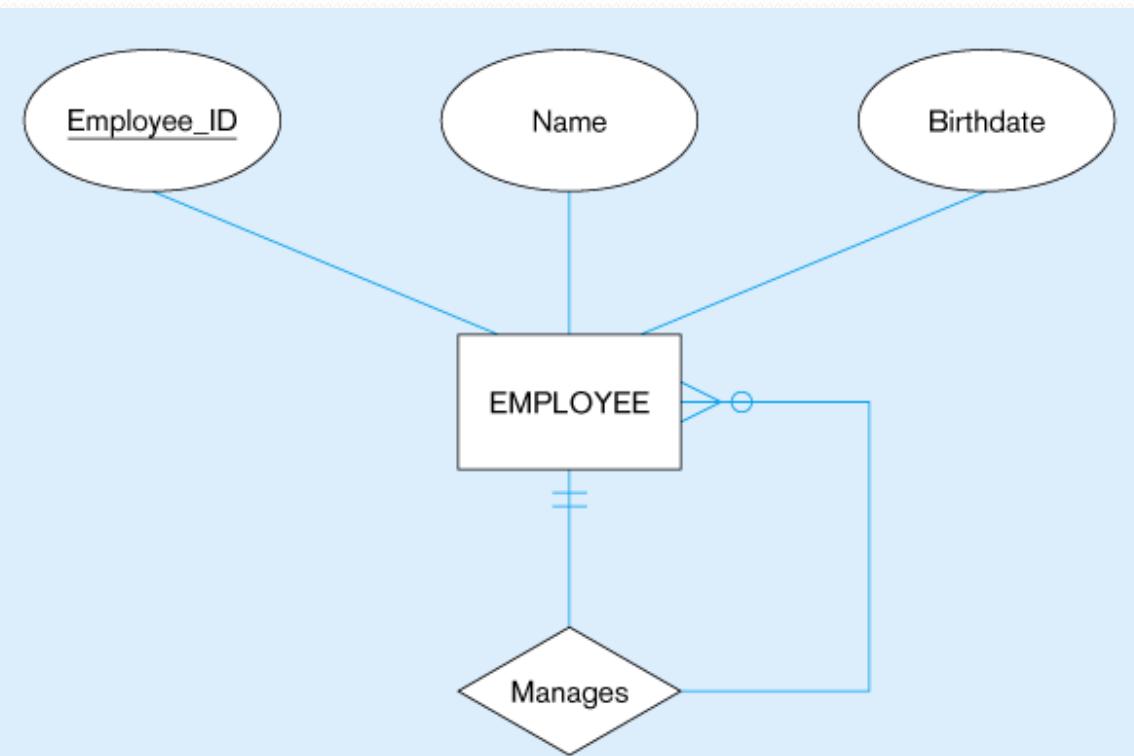
Vd3: Mối quan hệ bậc 1 (M:N)

PART(**PNUMBER**, DESCRIPTION,...)

COMPRISE (MAJOR-PNUMBER, MINOR-PNUMBER, QUANTITY)

I. Ánh xạ ER thành Quan hệ (tt)

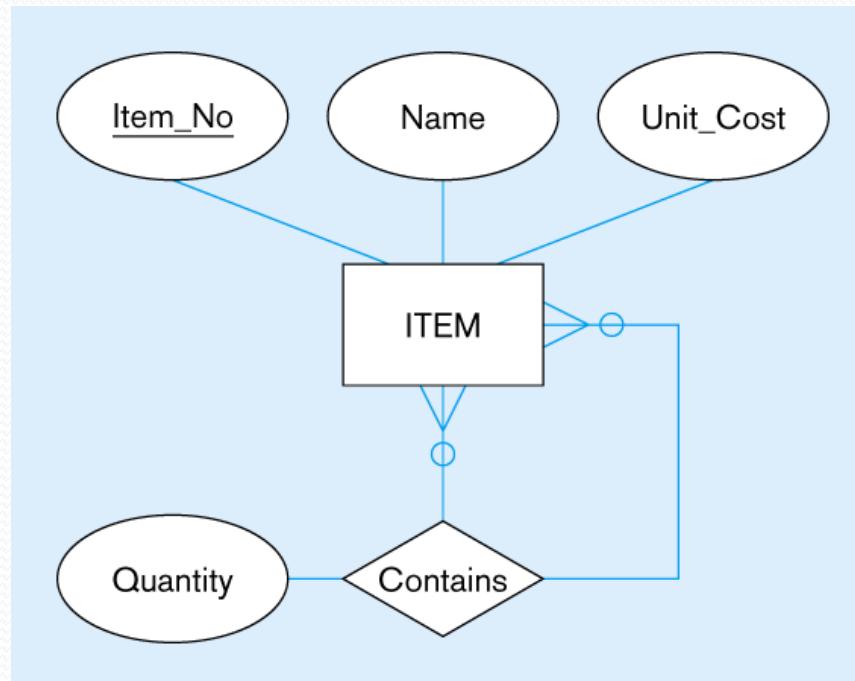
- Chuyển đổi các mối quan hệ bậc 1-đệ quy/đối hợp
- Vd4:** EMPLOYEE entity with Manages relationship



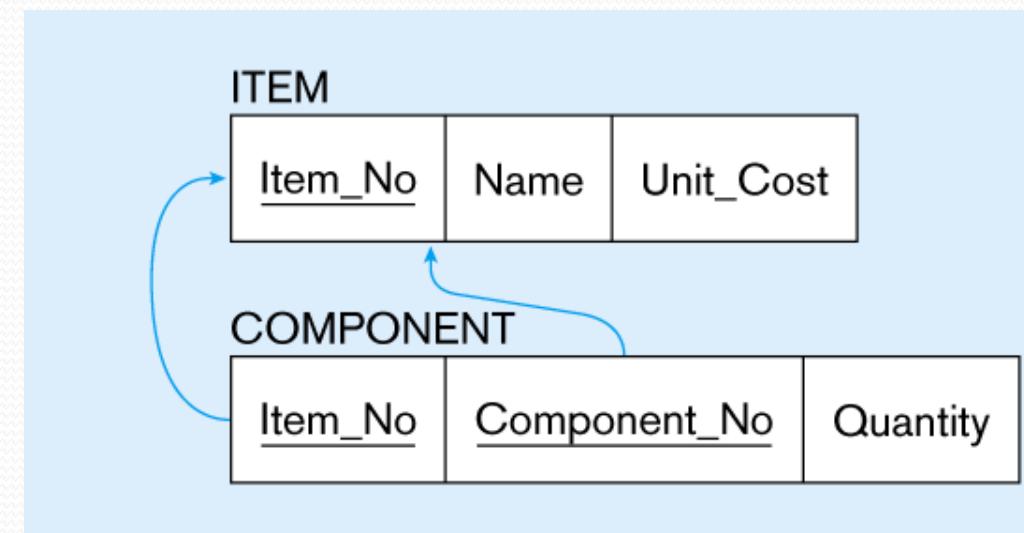
EMPLOYEE			
<u>Employee_ID</u>	Name	Birthdate	<u>Manager_ID</u>

I. Ánh xạ ER thành Quan hệ (tt)

- Chuyển đổi các mối quan hệ bậc 1-đệ quy/đối hợp
 - Vd5:** Bill-of-materials relationships (M:N)



Tạo 2 quan hệ mới:



I. Ánh xạ ER thành Quan hệ (tt)

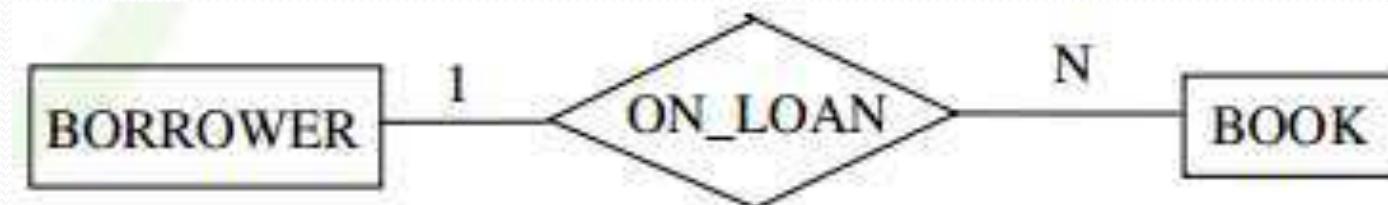
- B3: ánh xạ các kiểu mối quan hệ nhị phân 1:1
 - B4: ánh xạ các kiểu mối quan hệ nhị phân 1:N
 - B5: ánh xạ các kiểu mối quan hệ nhị phân M:N
(đã trình bày khi mối quan hệ bắt buộc cả 2 bên)
-
- Sự chuyển đổi các mối quan hệ nhị phân phụ thuộc vào chức năng của mỗi quan hệ và lượng thành viên tham gia của tập thực thể
→ bổ sung TH mối quan hệ nhị phân (bậc 2) tùy chọn

I. Ánh xạ ER thành Quan hệ (tt)

- Lớp tư cách thành viên **tùy chọn**
 - Nếu tập thực thể E2 là 1 thành viên tùy chọn của mỗi quan hệ N:1 với tập thực thể E1 (ie, E2 tại bên N của mỗi quan hệ) thì sau đó mỗi quan hệ thường được biểu diễn bởi 1 quan hệ mới chứa các thuộc tính khóa chính của E1 và E2, cùng với bất kỳ thuộc tính nào của mỗi quan hệ.
 - Khóa của tập thực thể tại bên N (ie. E2) sẽ trở thành khóa của quan hệ mới

I. Ánh xạ ER thành Quan hệ (tt)

- Vd Lớp tư cách thành viên **tùy chọn** cho mỗi quan hệ 1:N



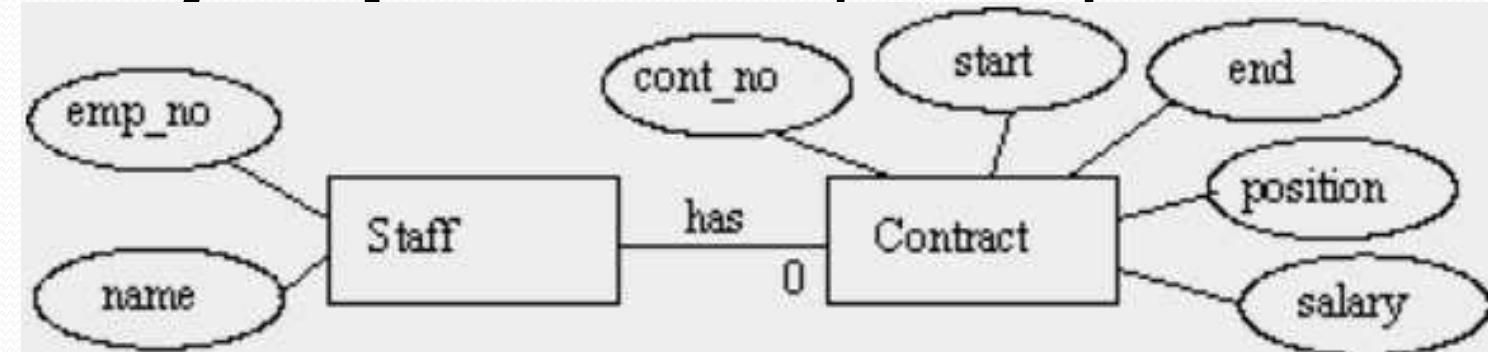
- Một biểu diễn có thể của mối quan hệ
BORROWER (BNUMBER, NAME, ADDRESS,...)
BOOK (ISBN, TITLE, ..., **BNUMBER**)
- Một lựa chọn tốt hơn
BORROWER (BNUMBER, NAME, ADDRESS,...)
BOOK (ISBN, TITLE, ...)
ON_LOAN (ISBN, BNUMBER)

I. Ánh xạ ER thành Quan hệ (tt)

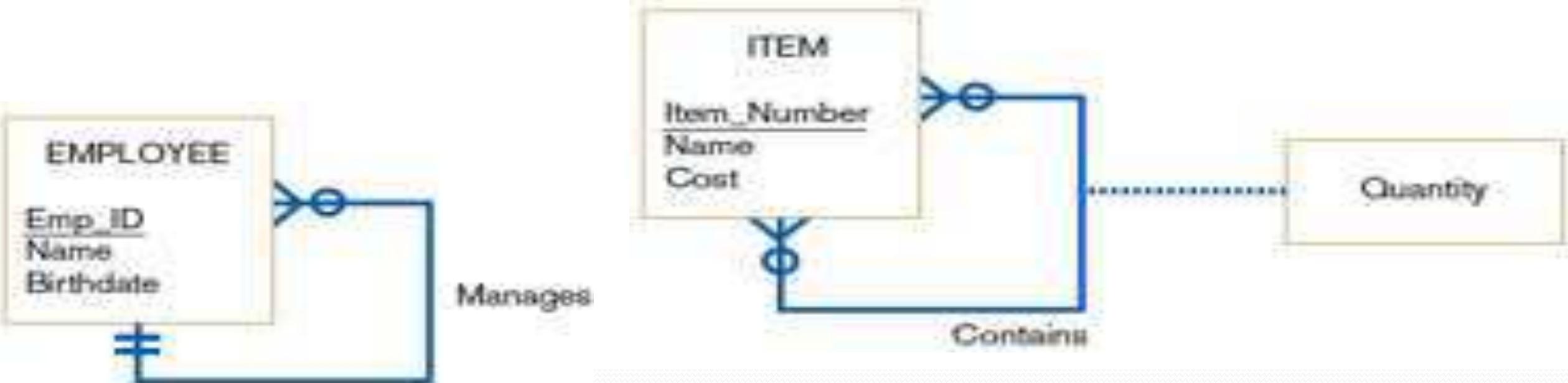
- Lớp tư cách thành viên **tùy chọn**
 - C1. Nếu các tập thực thể trong 1 mối quan hệ 1:1 có tư cách thành viên tùy chọn (1 bên hoặc cả 2 bên), thì 1 quan hệ mới được tạo chứa các thuộc tính khóa của cả 2 tập thực thể, cùng với bất kỳ thuộc tính nào của quan hệ. Các thuộc tính khóa của cả 2 tập thực thể sẽ là khóa của quan hệ mới.
 - C2. Tạo thành 2 quan hệ. Thêm PK của bên tùy chọn vào bên bắt buộc làm FK, thêm các thuộc tính của mỗi quan hệ vào bên bắt buộc (*bảng số min max - kí hiệu Chen*)

I. Ánh xạ ER thành Quan hệ (tt)

- Vd Lớp tư cách thành viên **tùy chọn** cho mỗi quan hệ 1:1



- Kí hiệu Chen $\text{Staff} (0,1) ----- (1,1) \text{Contract}$
- Dùng cách 2 tạo thành 2 quan hệ
Staff(emp_no, name)
Contract(cont_no, start, end, position, salary, *emp_no*)



EMPLOYEE(Emp_ID, Name, Birthdate, Manager_ID)

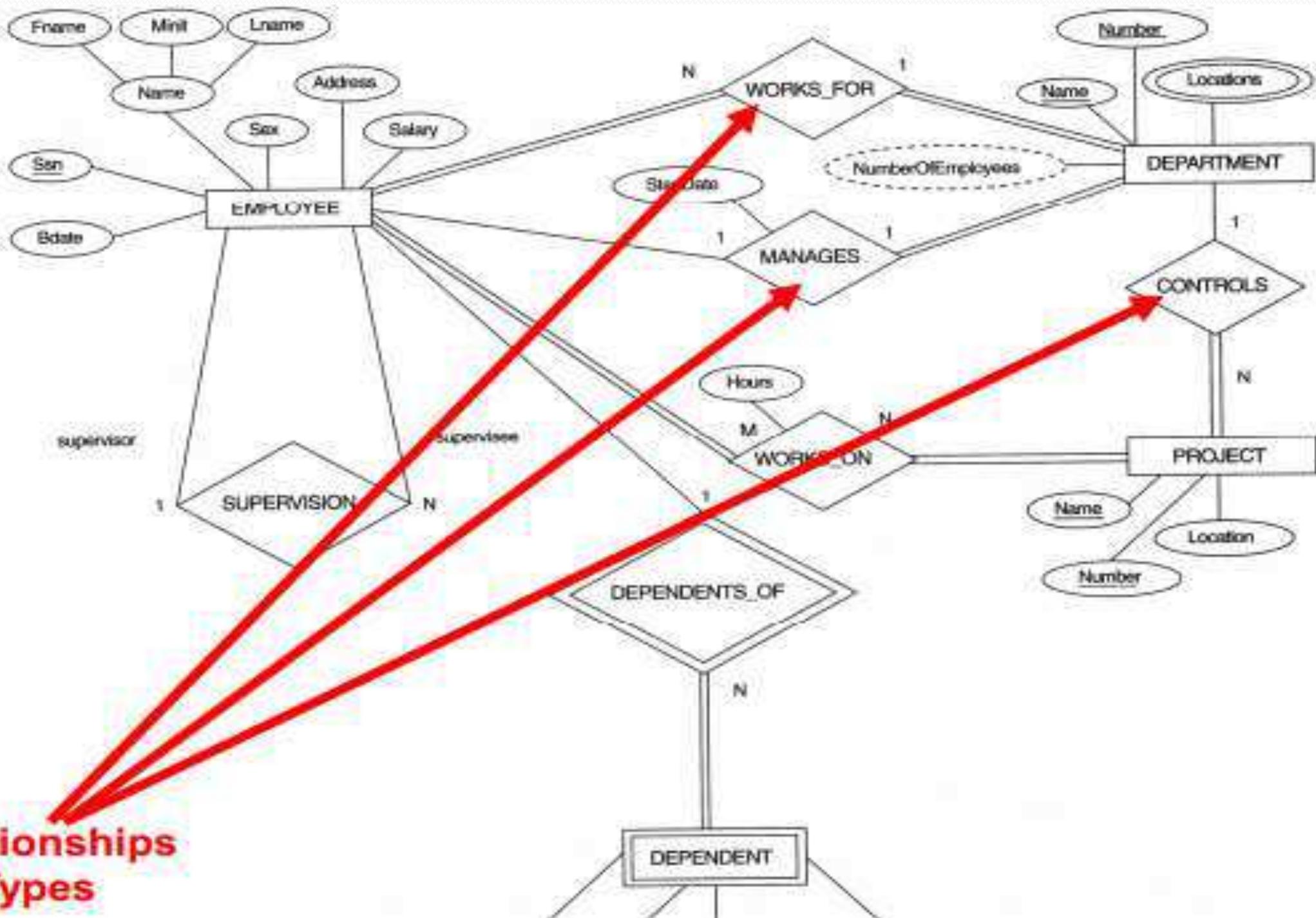
ITEM(Item_Number, Name, Cost)

ITEM-BILL(Item_Number, Component_Number, Quantity)

Hình 9-13
Hai liên kết bậc 1

(a) EMPLOYEE with
Managers
relationship (1:N)

(b) Bill-of-materials
structure (M:N)



**Relationships
Types**

EMPLOYEE

FNAME	MINIT	LNAME	<u>SSN</u>	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
-------	-------	-------	------------	-------	---------	-----	--------	----------	-----

DEPARTMENT

DNAME	<u>DNUMBER</u>	MGRSSN	MGRSTARTDATE
-------	----------------	--------	--------------

DEPT_LOCATIONS

<u>DNUMBER</u>	DLOCATION
----------------	-----------

PROJECT

PNAME	<u>PNUMBER</u>	PLOCATION	DNUM
-------	----------------	-----------	------

WORKS_ON

ESSN	PNO	HOURS
------	-----	-------

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
------	----------------	-----	-------	--------------

The ERD for the COMPANY database



EMPLOYEE

FNAME	MINIT	LNAME	<u>SSN</u>	BDATE	ADDRESS	SEX	SALARY	<u>SUPERSSN</u>	DNO
-------	-------	-------	------------	-------	---------	-----	--------	-----------------	-----

DEPARTMENT

DNAME	<u>DNUMBER</u>	MGRSSN	MGRSTARTDATE
-------	----------------	--------	--------------

DEPT_LOCATIONS

<u>DNUMBER</u>	DLOCATION
----------------	-----------

PROJECT

PNAME	<u>PNUMBER</u>	PLOCATION	DNUM
-------	----------------	-----------	------

WORKS_ON

<u>ESSN</u>	PNO	HOURS
-------------	-----	-------

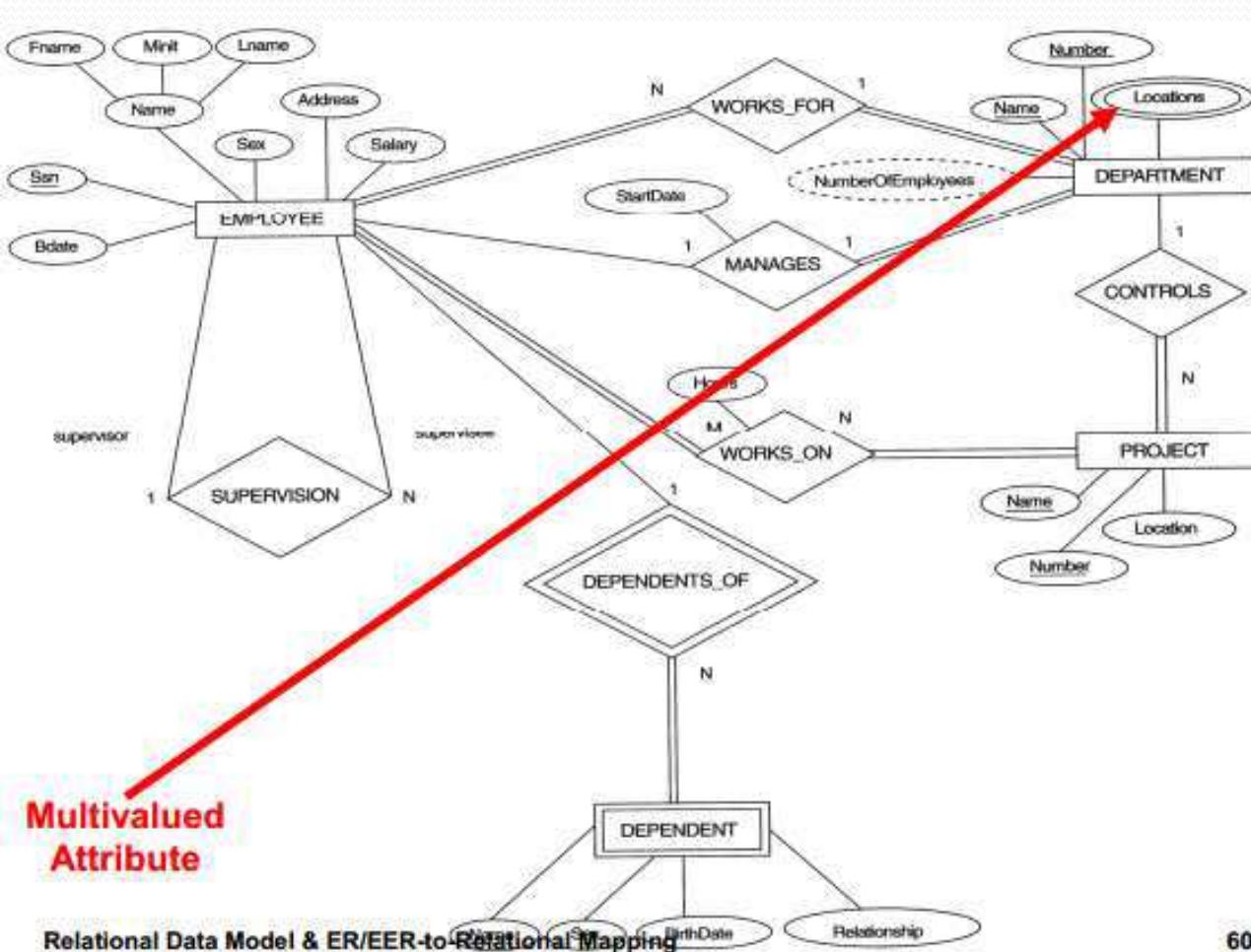
DEPENDENT

<u>ESSN</u>	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
-------------	----------------	-----	-------	--------------

???

I. Ánh xạ ER thành Quan hệ (tt)

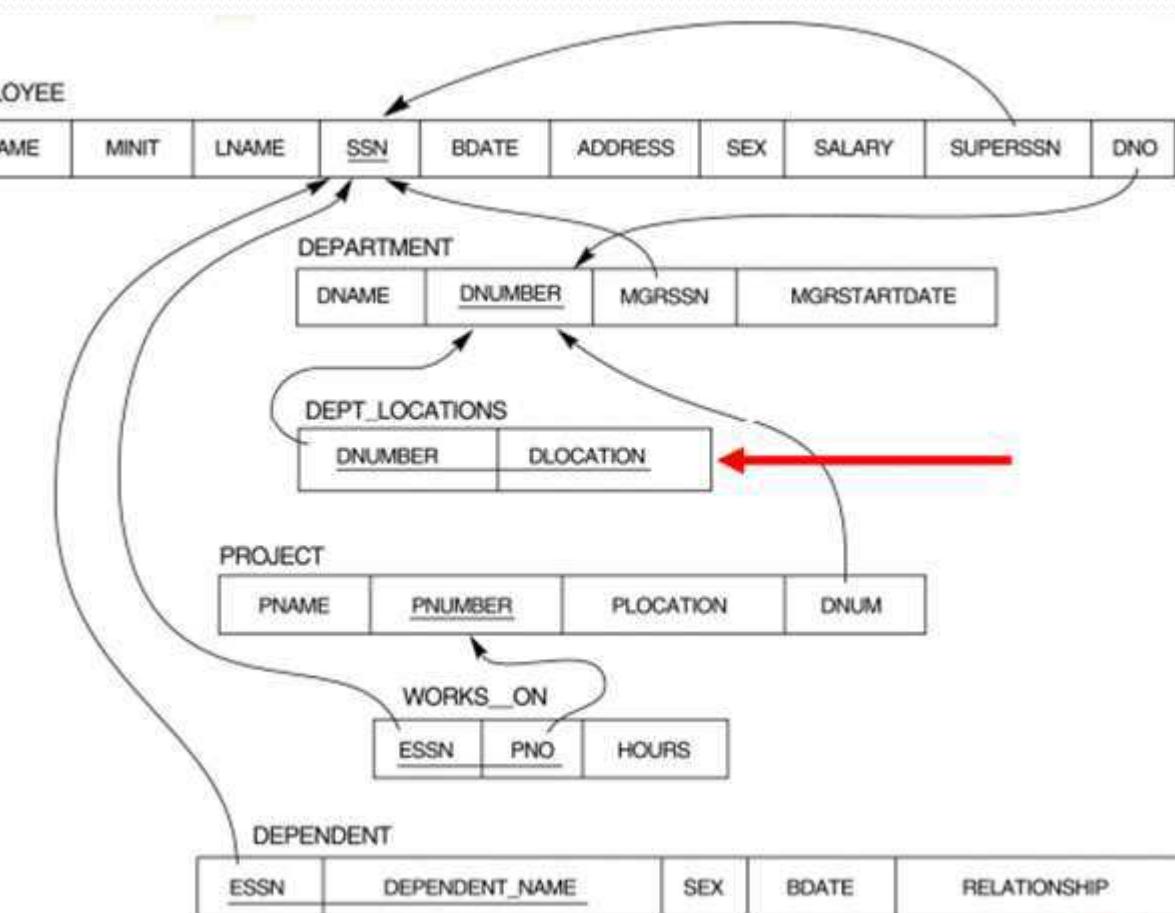
- Ánh xạ các thuộc tính đa trị
 - Với mỗi thuộc tính đa trị A, tạo 1 quan hệ R mới. Quan hệ R này sẽ gồm 1 thuộc tính A, cộng với thuộc tính khóa chính K của quan hệ biểu diễn tập thực thể hoặc kiểu mỗi quan hệ có A như 1 thuộc tính (K như 1 khóa ngoại trong R)
 - Khóa chính của R là sự kết hợp của A và K. Nếu thuộc tính đa trị là phức hợp, ta đính kèm các thành phần đơn giản của nó.
 - Vd, quan hệ DEPT_LOCATIONS được tạo. Thuộc tính DLOCATION biểu diễn thuộc tính đa trị LOCATION của DEPARTMENT, trong khi khóa ngoại DNUMBER biểu diễn khóa chính của quan hệ DEPARTMENT. Khóa chính của R là sự kết hợp của {DNUMBER, DLOCATION}



Multivalued Attribute

Relational Data Model & ER/EER-to-Relational Mapping

60



Ánh xạ ER thành Quan hệ

- Ánh xạ các kiểu mối quan hệ bậc n (>2)
 - Với mỗi kiểu quan hệ bậc n R, tạo 1 mối quan hệ mới S để biểu diễn R
 - Gồm các PK của các quan hệ làm FK trong S biểu diễn các tập thực thể tham gia.
 - Cũng chứa các thuộc tính đơn của kiểu mối quan hệ bậc n (hoặc các thành phần đơn của các thuộc tính phức hợp) làm các thuộc tính của S
 - Vd kiểu mối quan hệ SUPPLY trong hình bên. Nó có thể được ánh xạ thành quan hệ SUPPLY trong Idqh, nơi khóa chính là sự kết hợp của 3 khóa ngoại {SNAME, PARTNO, PROJNAME}

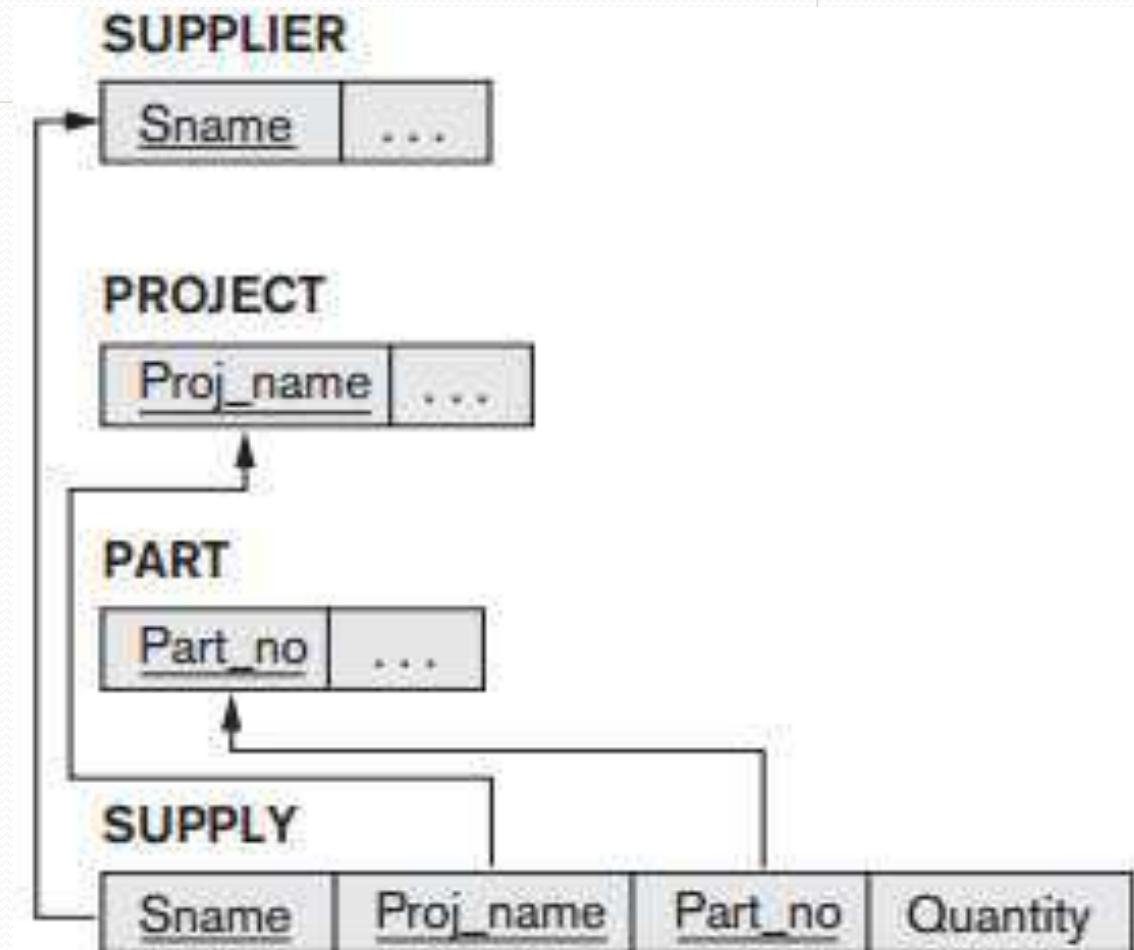
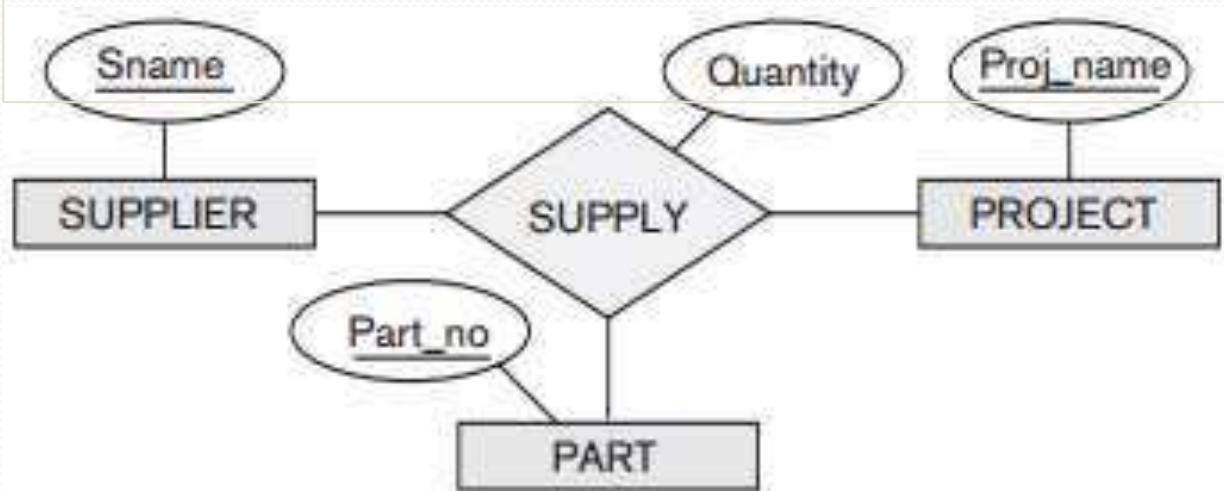
Ánh xạ ER thành Quan hệ

- **Ánh xạ các kiểu mối quan hệ bậc n (>2)**

- Khóa chính mặc định của S gồm các thuộc tính PK của các tập thực thể tham gia (thỉnh thoảng một số thuộc tính thêm vào được cần đến để tạo 1 PK duy nhất)
- Chú ý, nếu ràng buộc bản số của bất kỳ tập thực thể E nào tham gia trong R là 1, thì PK của S không gồm thuộc tính FK tham chiếu quan hệ E' tương ứng E.

Ánh xạ ER thành Quan hệ

Vd: ánh xạ kiểu mối quan hệ bậc n (>2) SUPPLY.



I. Ánh xạ EER- thành Quan hệ

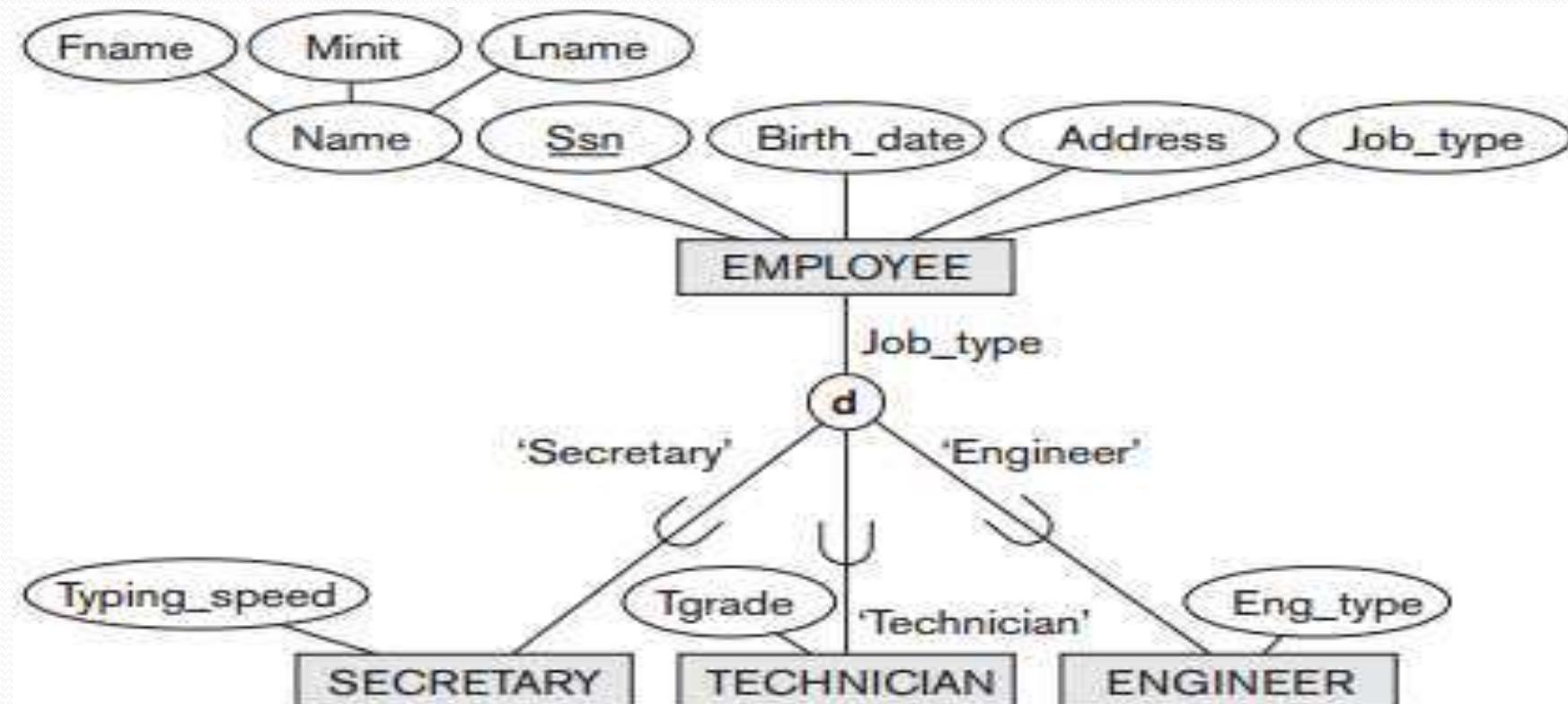
- B8: Các tùy chọn cho việc ánh xạ chuyên biệt hóa hoặc tổng quát hóa
 - Chuyển mỗi chuyên biệt hóa với m lớp con $\{S_1, S_2, \dots, S_m\}$ và siêu lớp C (được tổng quát hóa), nơi các thuộc tính của C là $\{k, a_1, \dots, a_n\}$ và k là khóa chính, thành các lđqh bằng cách dùng 1 trong 4 tùy chọn sau:
 - Tùy chọn 8A: đa quan hệ - siêu lớp và các lớp con
 - Tùy chọn 8B: đa quan hệ - chỉ các quan hệ lớp con
 - Tùy chọn 8C: quan hệ đơn với 1 kiểu thuộc tính
 - Tùy chọn 8D: quan hệ đơn với đa kiểu thuộc tính

I. Ánh xạ EER- thành Quan hệ (tt)

- B8: Các tùy chọn cho việc ánh xạ chuyên biệt hóa hoặc tổng quát hóa
 - Tùy chọn 8A: đa quan hệ - siêu lớp và các lớp con
 - Tạo 1 quan hệ L cho siêu lớp C với các thuộc tính $\text{Attrs}(L)=\{k,a_1,\dots,a_n\}$ và $\text{PK}(L)=k$.
 - Tạo 1 quan hệ L_i cho mỗi lớp con S_i , $1 \leq i \leq m$, với $\text{Attrs}(L_i)=\{k\} \cup \{\text{các thuộc tính của } S_i\}$ và $\text{PK}(L_i)=k$.
 - Tùy chọn 8A thực hiện cho bất kỳ chuyên biệt hóa nào (toute phần hoặc riêng phần, phân ly hoặc chồng lấp)
 - Tùy chọn 8B: đa quan hệ - chỉ các quan hệ lớp con
 - Tạo 1 quan hệ L_i cho mỗi lớp con S_i , $1 \leq i \leq m$, với $\text{Attrs}(L_i)=\{\text{các thuộc tính của } S_i\} \cup \{k,a_1,\dots,a_n\}$ và $\text{PK}(L_i)=k$
 - Tùy chọn 8B chỉ thực hiện cho 1 chuyên biệt hóa mà các lớp con của nó là toàn phần

I. Ánh xạ EER- thành Quan hệ (tt)

- Vd: 8A

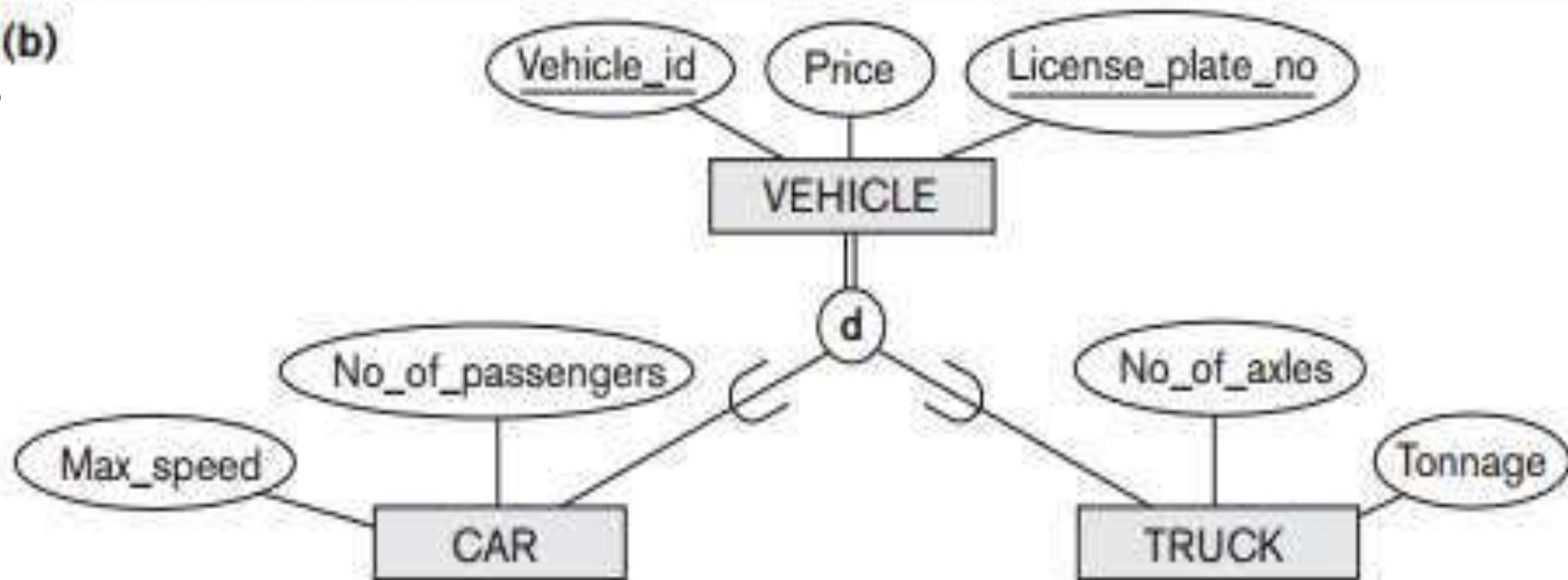


(a) **EMPLOYEE**



I. Ánh xạ EER- thành Quan hệ (tt)

- Vd: 8B (b)



(b) CAR

Vehicle_id	License_plate_no	Price	Max_speed	No_of_passengers
------------	------------------	-------	-----------	------------------

TRUCK

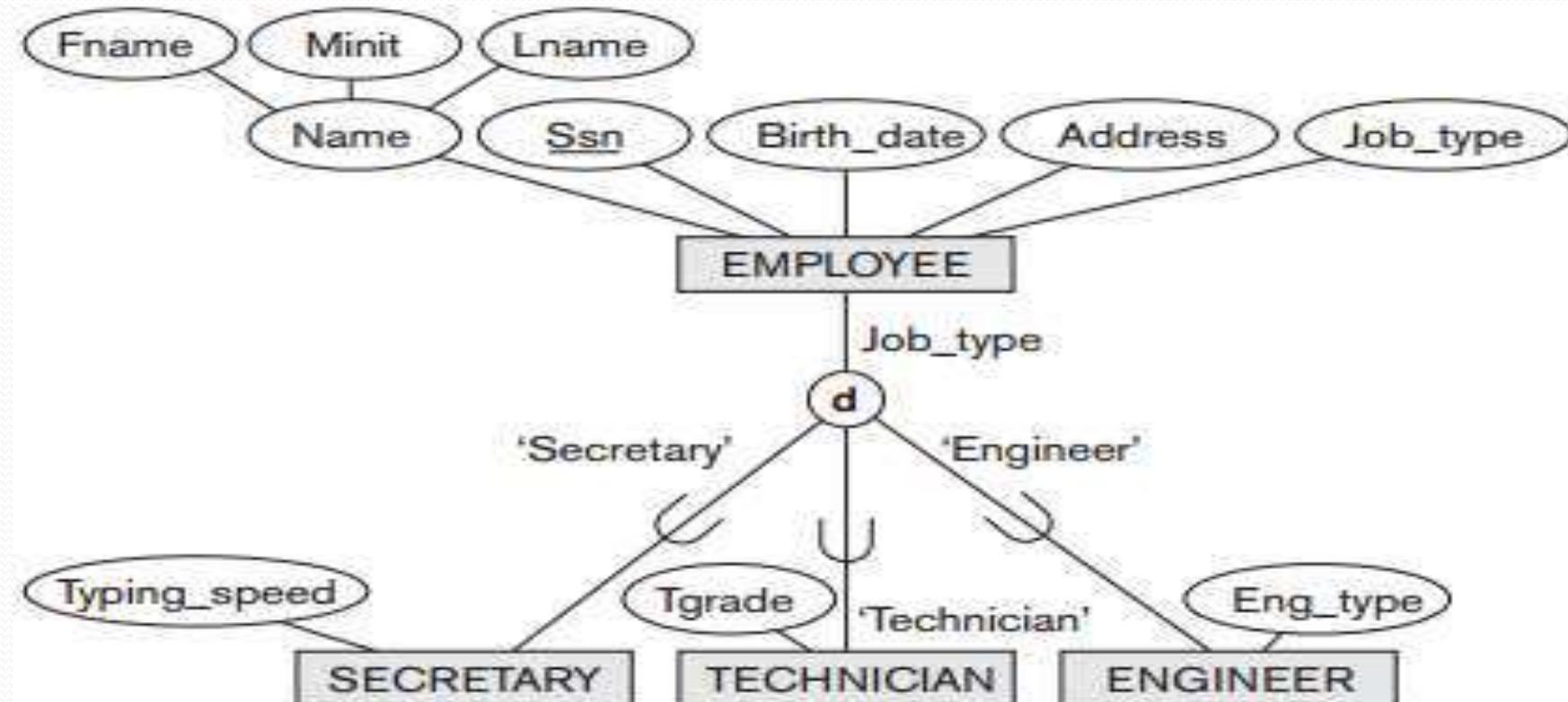
Vehicle_id	License_plate_no	Price	No_of_axles	Tonnage
------------	------------------	-------	-------------	---------

I. Ánh xạ EER- thành Quan hệ (tt)

- B8: Các tùy chọn cho việc ánh xạ chuyên biệt hóa hoặc tổng quát hóa
 - Tùy chọn 8C: quan hệ đơn với 1 kiểu thuộc tính
 - Tạo 1 quan hệ đơn L với $\text{Attrs}(L) = \{k, a_1, \dots, a_n\} \cup \{\text{các thuộc tính của } S_1\} \cup \dots \cup \{\text{các thuộc tính của } S_m\} \cup \{t\}$ và $\text{PK}(L) = k$. Thuộc tính t được gọi là 1 thuộc tính kiểu-type (hoặc phân biệt) mà giá trị của nó chỉ ra lớp con mà mỗi bản ghi phụ thuộc, nếu có.
 - Tùy chọn 8C chỉ thực hiện cho 1 chuyên biệt hóa mà các lớp con là phân ly, và có tiềm năng tạo nhiều giá trị NULL nếu nhiều thuộc tính đặc trưng/riêng biệt tồn tại trong các lớp con
 - Tùy chọn 8D: quan hệ đơn với đa kiểu thuộc tính
 - Tạo 1 lđqh L với $\text{Attrs}(L) = \{k, a_1, \dots, a_n\} \cup \{\text{các thuộc tính của } S_1\} \cup \dots \cup \{\text{các thuộc tính của } S_m\} \cup \{t_1, t_2, \dots, t_m\}$ và $\text{PK}(L) = k$. Mỗi t_i , $1 \leq i \leq m$, là 1 thuộc tính kiểu Boolean chỉ ra liệu 1 bản ghi có phụ thuộc vào lớp con S_i không
 - Tùy chọn 8D được dùng cho 1 chuyên biệt hóa mà các lớp con là chồng lấp (những cũng sẽ thực hiện cho 1 chuyên biệt hóa phân ly)

I. Ánh xạ EER- thành Quan hệ (tt)

- Vd: 8C

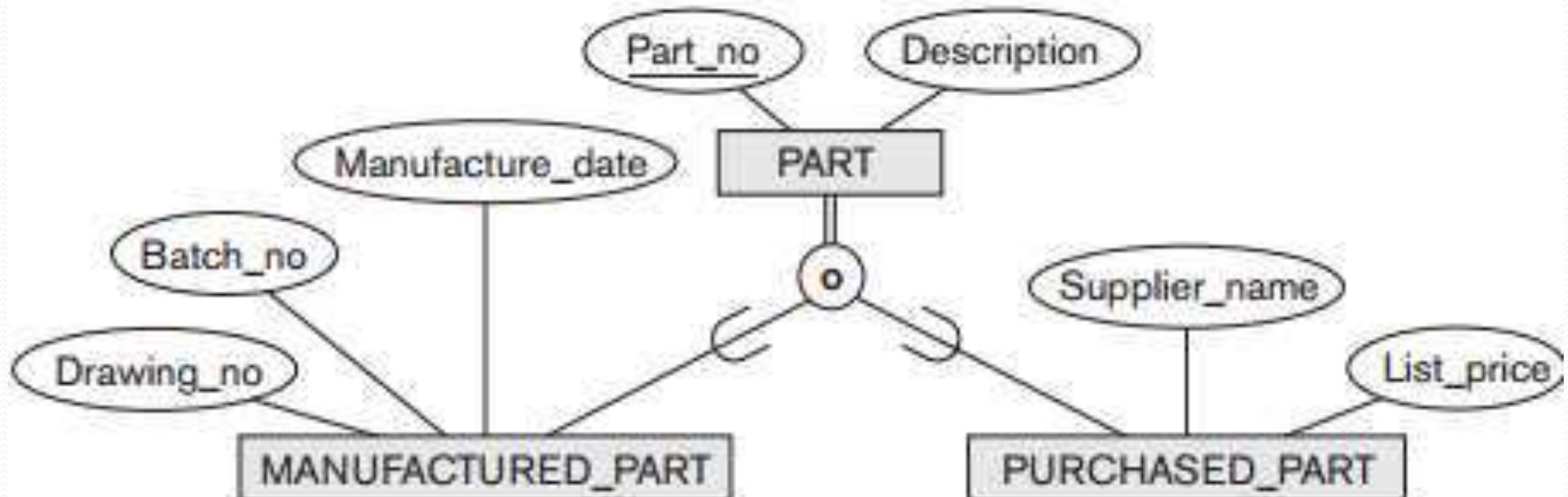


(c) **EMPLOYEE**

<u>Ssn</u>	Fname	Minit	Lname	Birth_date	Address	Job_type	Typing_speed	Tgrade	Eng_type
------------	-------	-------	-------	------------	---------	----------	--------------	--------	----------

I. Ánh xạ EER- thành Quan hệ (tt)

- Vd: 8D



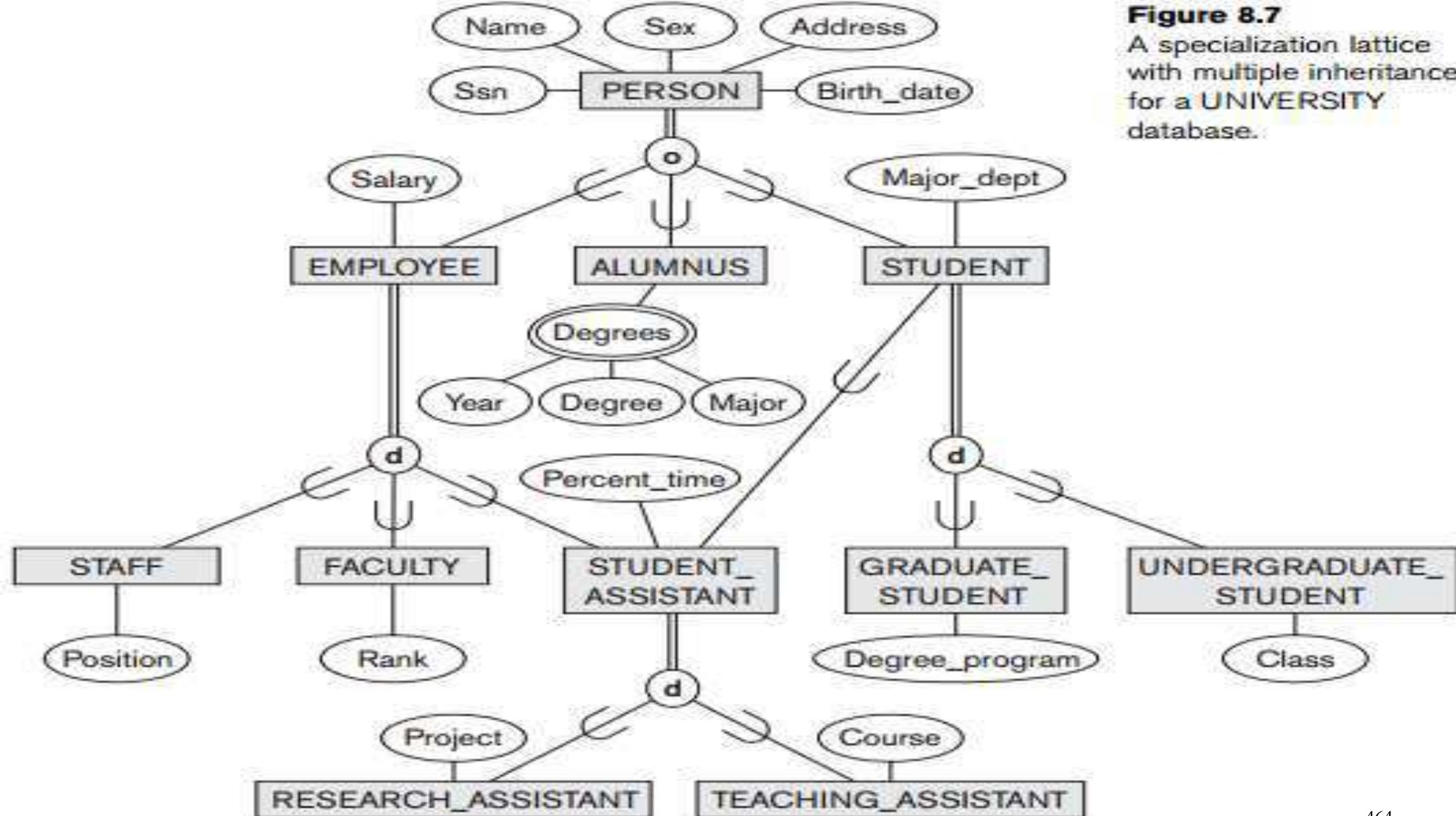
(d) PART

Part_no	Description	Mflag	Drawing_no	Manufacture_date	Batch_no	Pflag	Supplier_name	List_price
---------	-------------	-------	------------	------------------	----------	-------	---------------	------------

I. Ánh xạ EER- thành Quan hệ (tt)

- **Ánh xạ các Shared Subclass (Multiple Inheritance)**
 - Một shared subclass, như STUDENT_ASSISTANT, là 1 lớp con của vài lớp, chỉ sự đa kế thừa. Tất cả những lớp này phải có cùng thuộc tính khóa; nếu không thì shared class sẽ được lập mô hình như 1 category (union type).
 - Chúng ta có thể áp dụng bất kỳ tùy chọn nào trong B8 cho 1 shared subclass, chịu những hạn chế đã nói trong B8 của giải thuật ánh xạ.
 - Vd, tùy chọn 8C và 8D được dùng cho shared class STUDENT_ASSISTANT.

Figure 8.7
 A specialization lattice with multiple inheritance for a UNIVERSITY database.



PERSON

<u>Ssn</u>	Name	Birth_date	Sex	Address
------------	------	------------	-----	---------

EMPLOYEE

<u>Ssn</u>	Salary	Employee_type	Position	Rank	Percent_time	Ra_flag	Ta_flag	Project	Course
------------	--------	---------------	----------	------	--------------	---------	---------	---------	--------

ALUMNUS

<u>Ssn</u>	<u>Ssn</u>	Year	Degree	Major
------------	------------	------	--------	-------

ALUMNUS_DEGREES**STUDENT**

<u>Ssn</u>	Major_dept	Grad_flag	Undergrad_flag	Degree_program	Class	Student_assist_flag
------------	------------	-----------	----------------	----------------	-------	---------------------

Figure 9.6

Mapping the EER specialization lattice in Figure 8.8 using multiple options.

I. Ánh xạ EER- thành Quan hệ (tt)

- B9: **Ánh xạ các Union Type (Categories)**
 - Để ánh xạ 1 category mà xác định siêu lớp có các khóa khác nhau, cách thông thường để đặc tả 1 thuộc tính khóa mới là dùng khóa thay thế (surrogate key), khi tạo 1 quan hệ tương ứng category.
 - Trong vd, chúng ta tạo 1 quan hệ OWNER tương ứng category OWNER và gồm bất kỳ thuộc tính nào của category trong quan hệ này. PK của quan hệ OWNER là khóa thay thế OwnerId

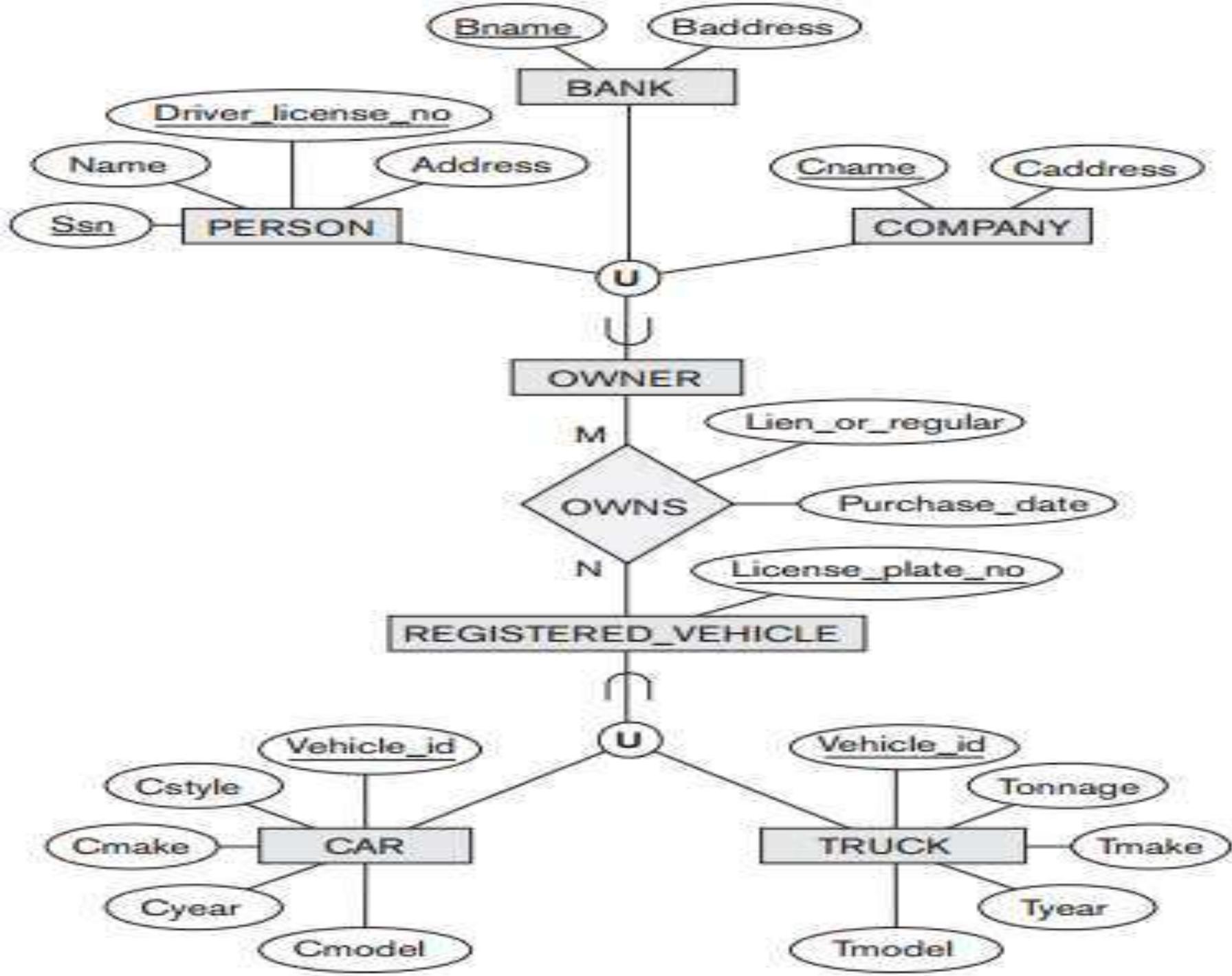


Figure 8.8
Two categories (union types): **OWNER** and **REGISTERED_VEHICLE**.

PERSON

Ssn	Driver_license_no	Name	Address	Owner_id
-----	-------------------	------	---------	----------

BANK

Bname	Baddress	Owner_id
-------	----------	----------

COMPANY

Cname	Caddress	Owner_id
-------	----------	----------

OWNER

Owner_id

REGISTERED_VEHICLE

Vehicle_id	License_plate_number
------------	----------------------

CAR

Vehicle_id	Cstyle	Cmake	Cmodel	Cyear
------------	--------	-------	--------	-------

TRUCK

Vehicle_id	Tmake	Tmodel	Tonnage	Tyear
------------	-------	--------	---------	-------

OWNS

Owner_id	Vehicle_id	Purchase_date	Lien_or_regular
----------	------------	---------------	-----------------

II. Chuẩn hóa các quan hệ

- Đưa các quan hệ về dạng chuẩn 3
 - Loại bỏ việc dư thừa dữ liệu

III. Hợp nhất các quan hệ

- Mục đích là bỏ các quan hệ dư thừa
- Bước cuối cùng trong thiết kế CSDL logic
- Trước khi thiết kế tệp vật lý và CSDL
- Các vấn đề tích hợp khung nhìn
 - Đồng nghĩa (khác tên cùng nghĩa)
 - Đồng âm (cùng tên khác nghĩa)
 - Phụ thuộc giữa các thuộc tính không khóa

Các vấn đề tích hợp khung nhìn

- Phải hiểu nghĩa của dữ liệu và được chuẩn bị để giải quyết bất kỳ vấn đề nào phát sinh trong tiến trình.
- **Đồng nghĩa:** hai tên gọi khác nhau được dùng cho cùng thuộc tính (khác tên cùng nghĩa)
 - Khi trộn, nhận sự đồng thuận từ những người dung về một tên đơn, tiêu chuẩn.
- **Đồng âm:** một thuộc tính đơn được dùng cho hai hoặc nhiều thuộc tính khác nhau (cùng tên khác nghĩa)
 - Giải quyết bằng cách tạo 1 tên mới
- **Phụ thuộc giữa các thuộc tính không khóa:** những phụ thuộc có thể được tạo như một kết quả của tích hợp khung nhìn
 - Để giải quyết, quan hệ mới phải được chuẩn hóa.

Các vấn đề tích hợp khung nhìn (tt)

- Vd: Minh họa việc hợp nhất quan hệ
 - giả sử lập mô hình 1 giao diện người dùng hoặc chuyển 1 sơ đồ ER thành quan hệ 3NF như sau:

EMPLOYEE1(Emp_ID, Name, Address, Phone)

- Lập mô hình 1 giao diện người dùng thứ 2 có thể đưa đến quan hệ sau EMPLOYEE2(Emp_ID, Name, Address, Jobcode, Number_of_Years)
- Vì 2 quan hệ này có cùng PK (Emp_ID) và mô tả cùng thực thể, →chúng nên được trộn thành 1 quan hệ như sau

EMPLOYEE(Emp_ID, Name, Address, Phone, Jobcode, Number_of_Years)

Các vấn đề tích hợp khung nhìn (tt)

- VD: Xem xét các vấn đề tích hợp
 - Từ đồng nghĩa (Synonyms): Student_ID và Matriculation

STUDENT1(Student_ID, Name)

STUDENT2(Matriculation_Number, Name, Address)



STUDENT(SSN, Name, Address)

- Từ đồng âm (Homonyms): Address của STUDENT1 chỉ 1 địa chỉ gia đình → thay bằng Permanent_Address, còn ở STUDENT2 thì chỉ địa chỉ campus (địa chỉ cư trú tại nơi học) của sv → thay bằng Campus_Address

STUDENT1(Student_ID, Name, Address)

STUDENT2(Student_ID, Name, Phone_Number, Address)



STUDENT(Student_ID, Name, Phone_Number, Campus_Address, Permanent_Address)

Các vần đề tích hợp khung nhìn (tt)

- VD: Xem xét các vần đề tích hợp

Dependencies between Nonkeys When two 3NF relations are merged to form a single relation, dependencies between nonkeys may result. For example, consider the following two relations:

STUDENT1(Student_ID, Major)
STUDENT2(Student_ID, Adviser)

Because STUDENT1 and STUDENT2 have the same primary key, the two relations may be merged:

STUDENT(Student_ID, Major, Adviser)

However, suppose that each major has exactly one adviser. In this case, Adviser is functionally dependent on Major:

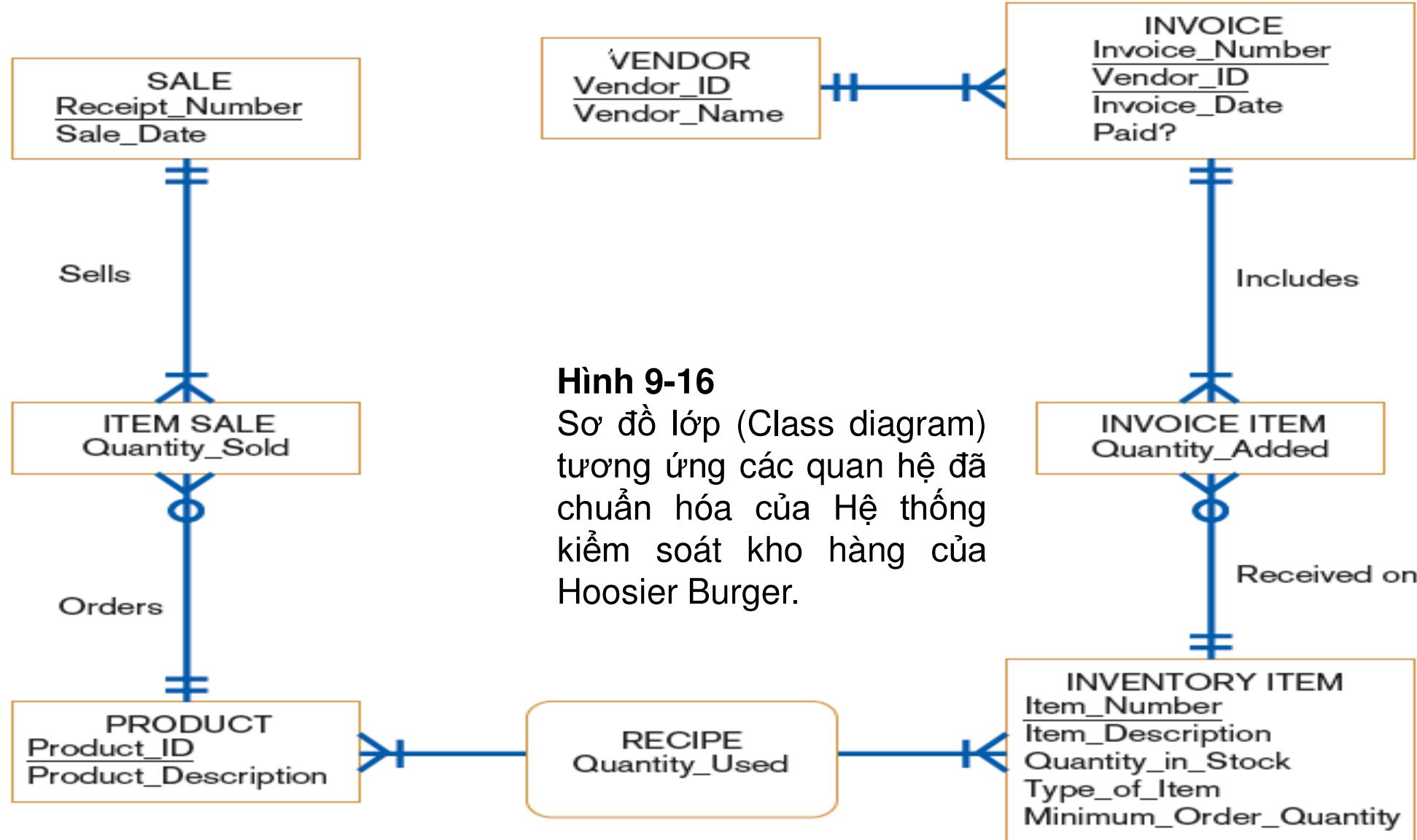
Major →→ Adviser

If the previous dependency exists, then STUDENT is in 2NF but not 3NF, because it contains a functional dependency between nonkeys. The analyst can create 3NF relations by creating two relations with Major as a foreign key in STUDENT:

STUDENT(Student_ID, Major)
MAJOR ADVISER(Major, Adviser)

Các vấn đề tích hợp khung nhìn (tt)

- **Lớp/Lớp con** – các liên kết có thể ẩn trong khung nhìn người dùng hoặc các quan hệ
 - Giải quyết bằng cách tạo 1 tên mới



Hình 9-16

Sơ đồ lớp (Class diagram) tương ứng các quan hệ đã chuẩn hóa của Hệ thống kiểm soát kho hàng của Hoosier Burger.

Các quan hệ của Hoosier Burger

SALE(Receipt Number,Sale Date)

PRODUCT(Product ID,Product Description)

INVOICE(Vendor ID,Invoice Number,Invoice Date,Paid?)

INVENTORY ITEM(Item Number,Item Description,Quantity_in_Stock,
Minimum_Order_Quantity,Type_of_Item)

ITEM SALE(Receipt Number,Product ID,Quantity_Sold)

INVOICE ITEM(Vendor ID,Invoice Number,Item Number,Quantity_Added)

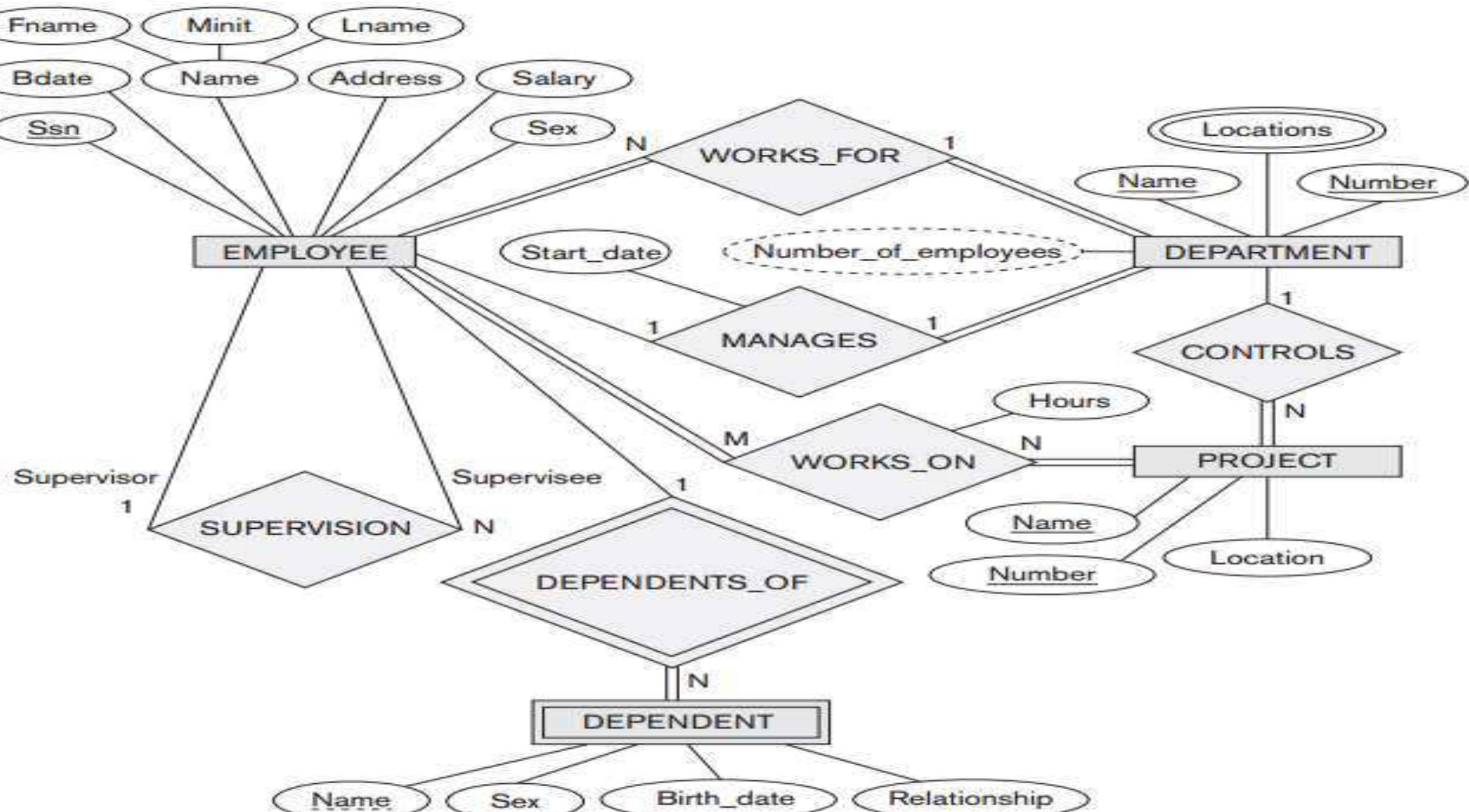
RECIPE(Product ID,Item Number,Quantity_Used)

VENDOR(Vendor ID,Vendor Name)

Cân bằng ERDs với DFDs

- Mô hình tiến trình tập trung vào các tiến trình của HT tổ chức gồm 2 thành phần dữ liệu
 - Luồng dữ liệu (chứa các yếu tố dữ liệu)
 - Kho dữ liệu
- Những thành phần dữ liệu của DFD cần tương ứng với các kho dữ liệu (ie. các thực thể) của ERD và các yếu tố dữ liệu chứa các luồng dữ liệu (ie. các thuộc tính) được vẽ trên mô hình dữ liệu.
- Một công cụ hữu ích để mô tả rõ ràng mối tương quan giữa tiến trình và các mô hình dữ liệu là **CRUD matrix**.

The ER conceptual schema diagram for the COMPANY database.



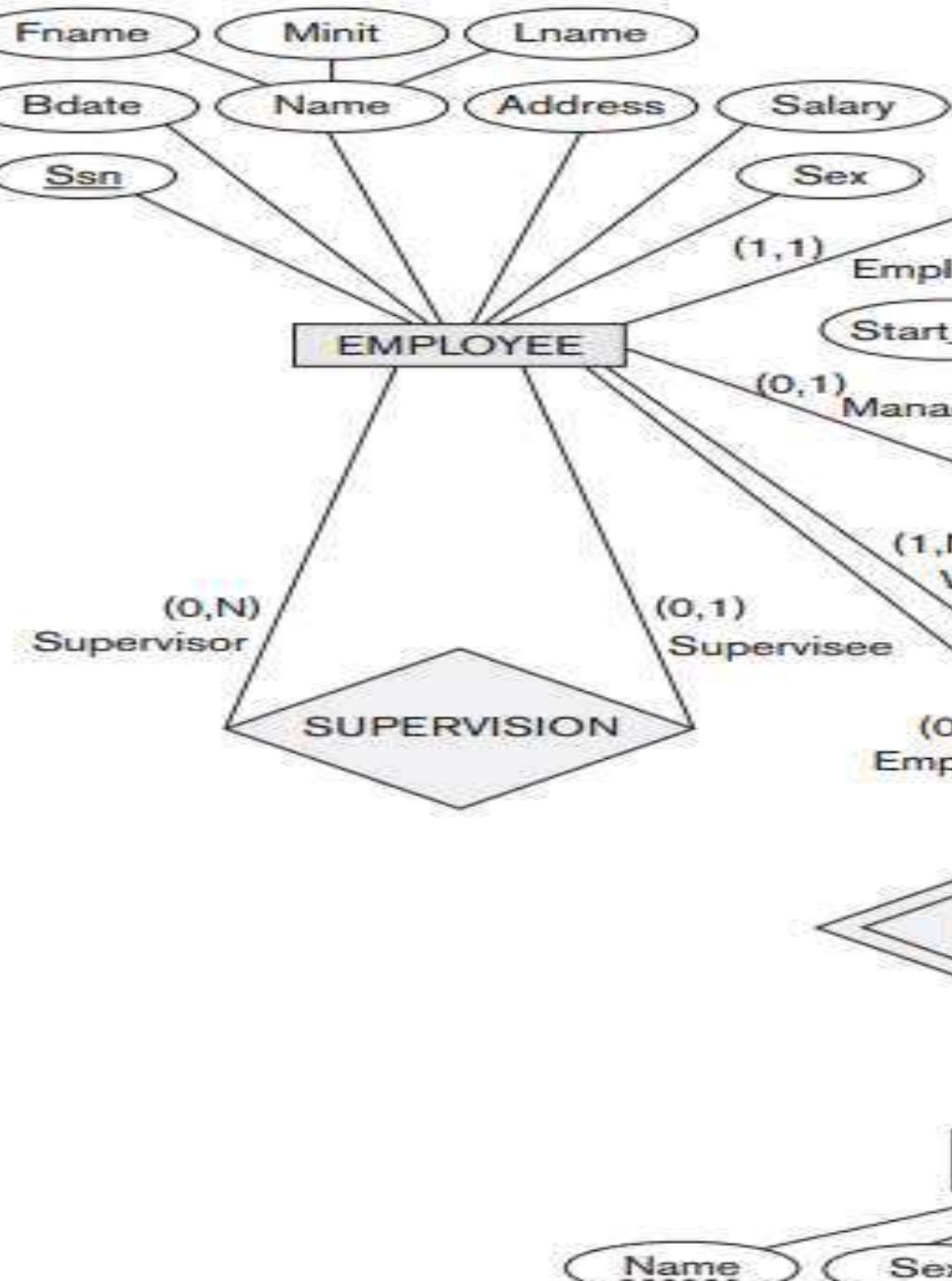


Figure 7.15
ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.

THIẾT KẾ FILE VÀ CSDL MỨC VẬT LÝ - Thông tin yêu cầu

- Để thiết kế file và CSDL mức vật lý cần có những thông tin sau
 - Các quan hệ được chuẩn hóa, bao gồm cả các ước tính về lượng dữ liệu của chúng
 - Định nghĩa của mỗi thuộc tính
 - Mô tả khi nào và trong trường hợp nào dữ liệu được dùng, được nhập, được khôi phục/truy xuất, bị xóa và được cập nhật (gồm cả tần suất thực hiện)
 - Sự mong đợi hoặc những yêu cầu cho thời gian phản hồi và tính toàn vẹn dữ liệu
 - Mô tả về những kỹ thuật/công nghệ được dùng để triển khai các tập tin và csdl

Thiết kế các trường

- **Trường (Field)**

- Đơn vị nhỏ nhất của dữ liệu ứng dụng được nhận ra bởi phần mềm hệ thống
- Mỗi thuộc tính từ mỗi quan hệ sẽ được biểu diễn như 1 hoặc nhiều trường

- **Lựa chọn loại/kiểu dữ liệu (Data Type) nhằm thỏa mãn**

- Tối thiểu hóa không gian lưu trữ
- Biểu diễn tất cả giá trị có thể cho trường
- Cải tiến tính toàn vẹn dữ liệu cho trường
- Hỗ trợ tất cả các thao tác dữ liệu được mong muốn cho trường

- **Các trường Calculated**

- Một trường có thể được bắt nguồn từ các trường csdl khác

Ràng buộc dữ liệu

- Một CSDL có thể gồm một hoặc nhiều ràng buộc, ràng buộc có thể trên một bảng, trên nhiều bảng
- Toàn vẹn dữ liệu chia thành 4 loại
 - **Toàn vẹn thực thể (Entry integrity)** hay ràng buộc khóa chính
 - **Toàn vẹn theo miền (Domain integrity)**
 - Là loại toàn vẹn ứng với các cột dữ liệu trong một phạm vi nào đó. Ví dụ kiểu dữ liệu, ràng buộc theo khóa check
 - **Toàn vẹn dang tham chiếu (Referential integrity)** hay ràng buộc kiểu mối quan hệ
 - khóa ngoại sẽ là khóa tham chiếu của khóa chính.
 - **Toàn vẹn do người dùng định nghĩa (User-defined integrity)**
 - Là toàn vẹn do người dùng định nghĩa, quy định dữ liệu nhập vào theo quy cách, giá trị được kiểm soát chặt chẽ, toàn vẹn kiểu này cũng có thể xây dựng trên cơ sở các toàn vẹn trước

Kiểm soát tính toàn vẹn dữ liệu

- Default Value
 - 1 giá trị của 1 trường sẽ thừa nhận trừ khi 1 giá trị rõ ràng được nhập cho trường đó.
- Input Mask
 - 1 mẫu đoạn mã giới hạn độ dài và các giá trị có thể có cho mỗi vị trí của 1 trường
- Range Control
 - Giới hạn phạm vi các giá trị có thể được nhập vào trường
- Referential Integrity
 - Một ràng buộc tính toàn vẹn đặc tả giá trị (hoặc tồn tại) của 1 thuộc tính trong 1 quan hệ phụ thuộc vào giá trị (hoặc tồn tại) của cùng thuộc tính trong quan hệ khác
- Null Value
 - Một giá trị đặc biệt của trường, khác với 0, để trống hoặc giá trị nào khác, trường này chỉ giá trị cho trường còn thiếu hoặc chưa biết.

Thiết kế các bảng vật lý

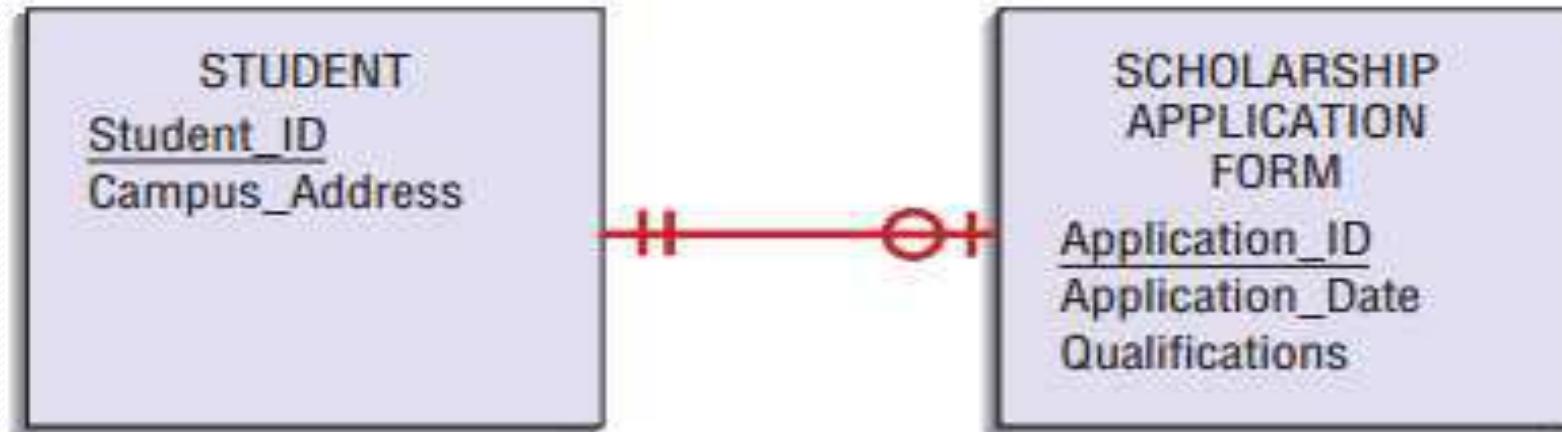
- CSDL quan hệ là 1 tập các bảng có quan hệ với nhau
- **Bảng vật lý**
 - Một tập hợp các dòng và cột được đặt tên nhằm đặc tả các trường trong mỗi dòng của bảng
- **Phi chuẩn hóa (Denormalization)**
 - Tiến trình chia hoặc kết hợp các quan hệ đã chuẩn hóa thành các bảng vật lý dựa trên sự giống nhau về cấu trúc sử dụng của các dòng và các trường dữ liệu.
 - Phi chuẩn hóa tối ưu các hoạt động xử lý dữ liệu nào đó gây bất lợi cho các hoạt động khác

Thiết kế các bảng vật lý (tt)

- Có thể thực hiện các dạng phi chuẩn hóa khác nhau (Phá vỡ chuẩn) bằng cách kết hợp dữ liệu từ vài bảng đã chuẩn hóa
- 3 tình huống chung nơi mà phi chuẩn hóa có thể được dùng:
 - 1. 2 thực thể với một liên kết 1:1
 - 2. một liên kết nhiều-nhiều với các thuộc tính không khóa
 - 3. dữ liệu tham chiếu

Ví dụ Phi chuẩn hóa

Hai thực thể với một quan hệ 1:1



Normalized relations:

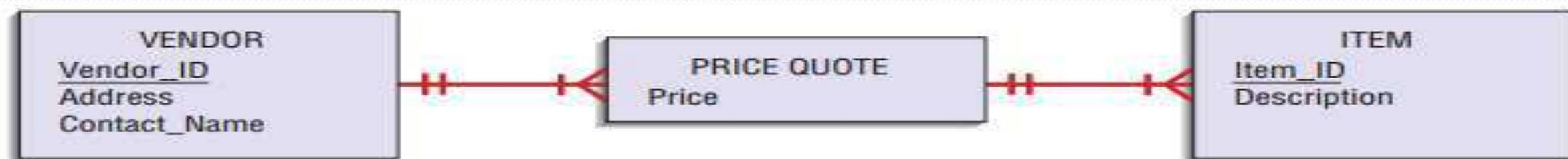
STUDENT(Student_ID, Campus_Address, Application_ID)

APPLICATION(Application_ID, Application_Date, Qualifications, Student_ID)

Denormalized relation:

STUDENT(Student_ID, Campus_Address, Application_Date, Qualifications) and Application_Date and Qualifications may be null

(Note: We assume Application_ID is not necessary when all fields are stored in one record, but this field can be included if it is required application data.)



Normalized relations:

VENDOR(Vendor_ID, Address, Contact_Name)
ITEM(Item_ID, Description)
PRICE QUOTE(Vendor_ID, Item_ID, Price)

Denormalized relations:

VENDOR(Vendor_ID, Address, Contact_Name)
ITEM-QUOTE(Vendor_ID, Item_ID, Description, Price)

B



Normalized relations:

STORAGE(Instr_ID, Where_Store, Container_Type)
ITEM(Item_ID, Description, Instr_ID)

Denormalized relation:

ITEM(Item_ID, Description, Where_Store, Container_Type)

C

Thiết kế các bảng vật lý (tt)

- Sắp xếp các dòng của bảng
 - File vật lý
 - Một tập hợp các dòng của bảng được chọn lưu trữ trong 1 vùng liền kề nhau của bộ nhớ thứ cấp
 - Mỗi bảng có thể là một file vật lý hoặc toàn bộ CSDL có thể là 1 file, phụ thuộc vào HQTCSDL được sử dụng

Thiết kế CSDL vật lý cho Hoosier Burger

- Các quyết định sau nên được thực hiện:

- Quyết định tạo một hoặc nhiều trường cho mỗi thuộc tính và xác định một loại dữ liệu cho mỗi trường.
- Với mỗi trường, quyết định liệu có dùng kiểu calculated; có cần được lập mã hoặc nén; phải có một hình hoặc giá trị mặc định; hoặc phải có một phạm vi, ràng buộc toàn vẹn, hoặc những kiểm soát giá trị null.
- Với mỗi quan hệ, quyết định liệu nó có nên được phi chuẩn hóa để đạt được hiệu quả xử lý mong muốn.
- Chọn 1 tổ chức file cho mỗi file vật lý.
- Chọn lựa các kiểm soát phù hợp cho mỗi file và CSDL.

Hoosier Burger

- Vd: Thiết kế bảng vật lý INVOICE

Field	Physical Design Parameter				
	Data Type and Size	Format and Input Mask	Default Value	Validation Rule	Required, Zero Length
Vendor_ID	Number	Fixed with 0 decimals, 9999	N/A	< 0	Required, not 0 length
Invoice_Number	Text, 10	1199-999999	N/A	N/A	Required, not 0 length
Invoice_Date	Date/Time	Medium date	= Date()	> #1/1/2000	Not required
Paid?	Yes/No	N/A	False	N/A	Required
Vendor_Name	Text, 30	N/A	N/A	N/A	Required, may be 0 length

Ứng dụng TMĐT: Thiết kế CSDL

- Thiết kế CSDL cho WebStore của PVF
 - Xem xét mô hình quan niệm (E-R diagram).
 - Kiểm tra những danh sách các thuộc tính cho mỗi thực thể.
 - Hoàn thành thiết kế CSDL
 - Chia sẻ thắt thông tin thiết kế với đội dự án để chuyển thành một CSDL hoạt động trong khi thực thi.

Chương 4. Thiết kế các hệ thống

- 4.2 Thiết kế giao diện người dùng
 - Biểu mẫu và Báo cáo
 - Hộp thoại và Giao diện
 - Hệ thống phân tán và Internet

Thiết kế Forms và Reports (tt)

- Các đầu vào và đầu ra HT được tạo ra vào cuối pha phân tích
 - Bề ngoài chính xác không cần thiết được định nghĩa trong pha phân tích
- Forms và reports có liên quan trọng vẹn với các sơ đồ DFD và E-R

Các khái niệm chính

- Biểu mẫu (Form)
 - Một tài liệu nghiệp vụ chứa một số dữ liệu được định nghĩa trước và có thể chứa một số vùng mà dữ liệu thêm được đưa vào
 - Một bản thể của một form thường dựa chủ yếu trên dữ liệu một bản ghi trong csdl
- Báo cáo (Report)
 - Một tài liệu nghiệp vụ chỉ chứa dữ liệu đã có sẵn thông tin
 - Một tài liệu bị động chỉ được dùng cho việc đọc và xem dữ liệu
 - Chủ yếu chứa dữ liệu từ nhiều bản ghi csdl hoặc các giao tác không liên quan nhau.

Quy trình thiết kế Forms và Reports

- Là một hoạt động chú trọng đến người sử dụng (User-focused)
- Theo một cách tiếp cận Mẫu thử
- Xác định các yêu cầu:
 - Ai sẽ dùng form hoặc report?
 - Mục đích của form và report là gì?
 - Khi nào form hoặc report được cần và sử dụng?
 - Form hoặc report cần được cung cấp và được dùng ở đâu?
 - Bao nhiêu người cần dùng hoặc xem form hoặc report?



Quy trình thiết kế Forms và Reports (tt)

- Mẫu thử (Prototyping)
 - Mẫu thử đầu tiên được thiết kế từ các yêu cầu
 - Người dùng xem xét thiết kế mẫu thử và có thể chấp nhận thiết kế hoặc yêu cầu thay đổi
 - Nếu có yêu cầu thay đổi, thì chu kỳ tinh chỉnh-đánh giá được lặp lại đến khi thiết kế được chấp nhận

Sản phẩm bàn giao và kết quả

- Những đặc tả thiết kế là những sản phẩm bàn giao chính và là đầu vào của pha thực thi HT
- Đặc tả thiết kế chứa 3 phần
 - 1. tổng quan sơ lược/khai quát (narrative overview): Đặc trưng người sử dụng, nhiệm vụ, hệ thống & các yếu tố môi trường
 - 2. thiết kế mẫu: hình cảnh của biểu mẫu (từ trang mã lệnh hoặc công cụ phát triển xây dựng biểu mẫu)
 - 3. kiểm thử và đánh giá khả năng sử dụng: thử nghiệm đo lường / kết quả sử dụng (thống nhất, hiệu suất, chính xác...)

Những nguyên tắc định dạng chung cho Forms và Reports

• Các nguyên tắc định dạng chung

Nguyên tắc	Mô tả
Dùng các tiêu đề có ý nghĩa	Các tiêu đề rõ ràng và cụ thể mô tả nội dung và việc dùng form hoặc report Chỉnh sửa ngày hoặc mã để phân biệt 1 form hoặc report từ những phiên bản trước đó. Ngày hiện tại xác định khi nào form hoặc report được tạo ra. Ngày hợp lệ xác định ngày nào (hoặc thời gian) dữ liệu trong form hoặc report là chính xác
Chứa thông tin có ý nghĩa	Chi thông tin cần thiết được hiển thị Thông tin được cung cấp theo 1 cách thông thường mà không sửa đổi
Cân bằng bố cục	Thông tin được cân bằng trong màn hình hoặc trang Không gian và đường biên hợp lý Tất cả dữ liệu và các trường đầu vào được gắn nhãn rõ ràng
Thiết kế 1 hệ thống định hướng dễ dàng	Chi rõ làm thế nào để tiến và lùi Chi rõ bạn đang ở đâu Chú ý người dùng trang cuối của 1 chuỗi nhiều trang

Những nguyên tắc định dạng chung cho Forms và Reports

• **Làm nổi bật (Highlighting) thông tin**

- Chỉ sử dụng để thu hút hoặc tập trung người dùng vào thông tin mong muốn
- Thông tin thường được làm nổi bật khi:
 - Thông báo cho người dùng lỗi khi nhập hoặc xử lý dữ liệu
 - Cảnh báo người dùng
 - Hướng sự chú ý đến các từ khóa, mệnh lệnh, thông báo
- Làm nhấp nháy, sử dụng các hiệu ứng âm thanh chỉ nên được dùng để làm nổi bật thông tin quan trọng cần sự chú ý ngay của người dùng
- Các PP nên được lựa chọn thống nhất và được dùng dựa trên mức độ quan trọng của thông tin được nhấn mạnh

Bảng 8-3: Các phương pháp làm nổi bật

Nhấp nháy và âm thanh nghe được

Sự khác biệt về màu sắc

Sự khác biệt cường độ

Sự khác biệt về kích thước

Sự khác biệt font chữ

Quay ngược video

Đóng khung

Gạch chân

Tất cả chữ in hoa

Vị trí bù của thông tin không chuẩn

Những nguyên tắc định dạng chung cho Forms và Reports (tt)

- **Hiển thị văn bản**

- Kết hợp chữ hoa, chữ thường và sử dụng dấu chấm câu thích hợp
- Sử dụng khoảng trắng gấp đôi nếu không gian cho phép. Nếu không thay thế một dòng khoảng trắng giữa các đoạn
- Văn bản canh trái đều và để lề phải không đều
- Không nối các từ giữa các dòng
- Sử dụng từ viết tắt (abbreviations) và những chữ đầu của một nhóm từ (acronyms)

Những nguyên tắc định dạng chung cho Forms và Reports (tt)

- **Hiển thị các bảng và danh sách (Tables and Lists)**
 - Các nhãn (Labels)
 - Tất cả các cột và dòng chứa là các nhãn có ý nghĩa
 - Các nhãn nên được tách biệt khỏi thông tin khác bằng cách làm nổi bật
 - Hiển thị lại các nhãn khi dữ liệu mở rộng quá 1 màn hình hoặc 1 trang đơn

Những nguyên tắc định dạng chung cho Forms và Reports (tt)

- **Hiển thị các bảng và danh sách (Tables and Lists) (tt)**
 - **Định dạng các cột, dòng, và văn bản**
 - Sắp xếp theo một trật tự có ý nghĩa
 - Thay 1 dòng trống cách 5 dòng một trong các cột dài
 - Thông tin giống nhau được hiển thị trong nhiều cột nên được xếp theo chiều dọc
 - Các cột nên có ít nhất 2 khoảng trống giữa chúng
 - Cho phép khoảng trống trắng trên các reports được in cho người dùng để viết ghi chú
 - Sử dụng 1 kiểu chữ đơn, trừ khi nhấn mạnh
 - Sử dụng kiểu font chữ giống nhau trong các màn hình và báo cáo
 - Tránh những font chữ quá kiểu cách

Những nguyên tắc định dạng chung cho Forms và Reports (tt)

- **Hiển thị các bảng và danh sách (Tables and Lists) (tt)**
 - **Định dạng dữ liệu số, văn bản, và chữ số**
 - Dữ liệu số canh phải-đầu và canh các cột theo dấu thập phân hay những dấu phân cách khác
 - Dữ liệu văn bản canh trái-đầu. Sử dụng độ dài dòng ngắn, thường 30-40 ký tự trên một dòng
 - Phá những dãy dữ liệu chữ số dài thành những nhóm nhỏ từ 3 đến 4 ký tự mỗi nhóm

Ứng dụng TMĐT: Việc thiết kế Forms và Reports cho WebStore của Pine Valley Furniture

- Những nguyên tắc chung cho việc triển khai nhanh các trang Web Internet đã tồn.
- Ba giải pháp có thể dùng để giải quyết vấn đề:
 - Có thể thiết kế những trang thỏa mãn có thể dùng mà không cần kinh nghiệm UI.
 - Huấn luyện nhiều người hơn trong việc thiết kế Web tốt.
 - Sống chung với những trang đã thiết kế tồi khó sử dụng.

Thiết kế Forms và Reports tại Pine Valley Furniture

- PVF đã thiết lập các nguyên tắc sau:
 - Sử dụng đồ họa nhẹ.
 - Thiết lập các luật tích hợp forms và dữ liệu.
 - Sử dụng HTML dựa template.

Chương 4. Thiết kế các hệ thống

- 4.2 Thiết kế giao diện người dùng
 - Biểu mẫu và Báo cáo
 - **Hộp thoại và Giao diện**
 - Hệ thống phân tán và Internet

Thiết kế các giao diện và hộp thoại (Interfaces and Dialogues)

- Tập trung vào việc cung cấp thông tin cho người dùng và thu nhận thông tin từ phía người dùng
- Các hộp thoại tương tự một cuộc hội thoại giữa 2 người
- Thiết kế giao diện và hộp thoại liên quan đến cách thức con người và máy tính trao đổi thông tin
- Một giao diện người-máy tốt cung cấp một cấu trúc thống nhất cho việc tìm kiếm, hiển thị và kích hoạt những thành phần khác nhau của 1 HT

Quy trình thiết kế các giao diện và hộp thoại

- Hoạt động tập trung người dùng (hay lấy người dùng làm trung tâm)
- Thực hiện song song với thiết kế Form và Report
- Sử dụng phương pháp Mẫu thử
 - Tập hợp thông tin
 - Xây dựng mẫu thử
 - Đánh giá khả năng sử dụng
 - Tạo hiệu chỉnh
- Sản phẩm bàn giao
 - Những đặc tả thiết kế
 - Tổng quan mô tả
 - Thiết kế mẫu
 - Kiểm thử và đánh giá khả năng sử dụng / tính tiện dụng

Design Specification

1. Narrative Overview
 - a. Interface/Dialogue Name
 - b. User Characteristics
 - c. Task Characteristics
 - d. System Characteristics
 - e. Environmental Characteristics
2. Interface/Dialogue Designs
 - a. Form/Report Designs
 - b. Dialogue Sequence Diagram(s) and Narrative Description
3. Testing and Usability Assessment
 - a. Testing Objectives
 - b. Testing Procedures
 - c. Testing Results
 - i) Time to Learn
 - ii) Speed of Performance
 - iii) Rate of Errors
 - iv) Retention over Time
 - v) User Satisfaction and Other Perceptions

Hình 11-2
Phác thảo đặc tả cho thiết kế
giao diện và hộp thoại

Thiết kế các giao diện (Interfaces)

- Thiết kế bối cục
- Thiết kế trường nhập liệu
- Kiểm soát dữ liệu nhập vào
- Cung cấp phản hồi
- Cung cấp trợ giúp

Thiết kế bố cục (Layouts)

- Các định dạng tiêu chuẩn giống các biểu mẫu và các báo cáo dựa trên giấy nên được sử dụng
- Định hướng màn hình trên các màn hình nhập liệu nên trình bày từ trái sang phải, trên xuống dưới như định dạng giấy

Thiết kế bố cục (Layouts)

- Sự linh hoạt và thống nhất là những mục tiêu thiết kế chính
 - Người dùng có thể dễ dàng di chuyển giữa các trường
 - Dữ liệu chưa được ghi cho đến khi được sự xác nhận của người dùng
 - Mỗi khóa hoặc lệnh chỉ tương ứng với một chức năng duy nhất

Thiết kế trường nhập dữ liệu (Data Entry)

- Đầu vào (entry): không yêu cầu nhập liệu các trường có sẵn hoặc có thể được tính toán
- Mặc định (Defaults): luôn cung cấp các giá trị mặc định khi phù hợp
- Units: chỉ ra kiểu đơn vị / loại đơn vị cho dữ liệu nhập vào
- Replacement: thay thế kí tự khi phù hợp
- Captioning: luôn có tiêu đề cho các trường nhập liệu
- Format: cung cấp các ví dụ định dạng
- Justify: tự động canh chỉnh các đầu vào dữ liệu
- Help: cung cấp trợ giúp ngữ cảnh-nhạy khi phù hợp

Kiểm soát nhập dữ liệu (Data Input)

- Kiểm soát dữ liệu nhập vào để giảm lỗi gây ra khi nhập dữ liệu
- Vai trò của nhà phân tích HT là đoán trước các lỗi người dùng và những tính năng thiết kế đưa vào các giao diện của HT để tránh, phát hiện và sửa lỗi nhập liệu

Kiểm soát nhập dữ liệu (Data Input)

- Bảng 8-8 mô tả các tùy chọn thiết kế cho việc nhập văn bản.

Options

Line caption

Drop caption

Boxed caption

Delimited characters

Check-off boxes

Example

Phone Number []

[]

Phone Number

Phone Number

111 111 1111

Phone Number

Method of payment (check one)

Check

Cash

Credit card: Type

Các lỗi thường gặp khi nhập liệu

- Nối dữ liệu
 - Thêm ký tự bổ sung vào một trường
- Cắt bớt dữ liệu
 - Làm mất ký tự trong trường
- Ghi lại dữ liệu
 - Nhập dữ liệu không hợp lệ vào trường
- Chuyển vị dữ liệu
 - Đảo ngược chuỗi của một hoặc nhiều ký tự trong trường

Data Error

Appending

Truncating

Transcribing

Transposing

Description

Adding additional characters to a field

Lossing characters from a field

Entering invalid data into a field

Reversing the sequence of one or more characters in a field

Các kỹ thuật kiểm tra lỗi nhập liệu

- Phân nhóm
- Kết hợp
 - Kiểm tra kết hợp giá trị của nhiều trường
- Giá trị mong muốn
- Thiếu sót dữ liệu
- Định dạng
- Phạm vi
- Hợp lý
- Số ký tự
- Giá trị

Kiểm tra tính hợp lệ	Mô tả
Lớp hoặc thành phần	Kiểm tra để đảm bảo rằng dữ liệu có kiểu thích hợp (ví dụ: tất cả các số, tất cả các chữ cái, chữ số)
Kết hợp	Kiểm tra để thấy rằng các kết hợp giá trị của hai hoặc nhiều trường dữ liệu là thích hợp hoặc có ý nghĩa (ví dụ: số lượng bán có hợp lý hay không với loại sản phẩm được đưa ra?)
Các giá trị được mong đợi	Kiểm tra xem dữ liệu có phải là cái mong đợi hay không (ví dụ: phù hợp với tên khách hàng hiện tại, số tiền thanh toán, v.v.)
Mất dữ liệu	Kiểm tra sự tồn tại của các mục dữ liệu trong tất cả các trường của một bản ghi (ví dụ: có một trường số lượng trên mỗi mục hàng của một đơn đặt hàng của khách hàng không?)
Ảnh / Mẫu	Kiểm tra để đảm bảo dữ liệu phù hợp với định dạng chuẩn (ví dụ: dấu gạch nối ở đúng nơi cho số ID một sinh viên?)
Phạm vi	Kiểm tra để đảm bảo dữ liệu nằm trong phạm vi giá trị phù hợp (ví dụ: điểm trung bình của điểm số của một học sinh từ 0 đến 4.07)
Tính hợp lý	Kiểm tra để đảm bảo dữ liệu hợp lý cho tình huống (ví dụ: tỷ lệ trả lương cho một loại nhân viên cụ thể)
Chữ số tự kiểm tra	Kỹ thuật mà các kí tự thêm vào được lấy bằng cách dùng một công thức chuẩn (xem hình 8-15), được thêm vào một trường số trước khi truyền và được kiểm tra sau khi truyền
Kích thước	Kiểm tra khi quá ít hoặc quá nhiều ký tự (ví dụ: số an sinh xã hội chính xác có chín chữ số không?)
Các giá trị	Kiểm tra để đảm bảo rằng các giá trị đến từ một bộ các giá trị chuẩn (ví dụ: mã trạng thái hai chữ cái)

Cung cấp phản hồi (Feedback)

- 1. Thông tin tình trạng
 - Thông báo người dùng những gì đang diễn ra trong HT
 - Hiển thị thông tin tình trạng đặc biệt quan trọng nếu hoạt động diễn ra lâu hơn 1 hoặc 2 giây
- 2. Những lời nhắc gợi ý
 - Càng rõ ràng càng tốt
- 3. Các thông báo lỗi và cảnh báo
 - Các thông điệp nêu cụ thể và không có các mã lỗi và biệt ngữ
 - Người dùng nên được hướng dẫn đi đến 1 kết quả hơn là khiển trách
 - Sử dụng các thuật ngữ thân thiện với người dùng
 - Thống nhất trong định dạng và bố trí sắp xếp các thông điệp

Cung cấp trợ giúp (Help)

- Đặt chính bạn vào vị trí của người dùng khi thiết kế trợ giúp
- Các nguyên tắc
 - Đơn giản
 - Các thông điệp trợ giúp nên ngắn và trọng tâm
 - Tổ chức
 - Thông tin trong các thông điệp trợ giúp nên dễ dàng được hiểu bởi những người dùng
 - Trực quan
 - Cung cấp rõ ràng các ví dụ cho người dùng để họ biết cách thực hiện một thao tác là có ích.

Cung cấp trợ giúp (Help) (tt)

- Trợ giúp ngữ cảnh-nhạy
 - Cho phép người dùng nhận trợ giúp theo trường cụ thể
- Những người dùng nên luôn được quay về nơi họ đã ở khi yêu cầu trợ giúp
- Một số môi trường lập trình cung cấp công cụ để thiết kế tài liệu trợ giúp
 - SDK (Software Development Kit)

Thiết kế các hộp thoại (Dialogues)

- **Hộp thoại**
 - Một chuỗi hành động trong đó hiển thị thông tin cho người dùng và nhận thông tin từ người dùng
- Nguyên tắc thiết kế chính là sự thống nhất trong trình tự các hành động, tổ hợp phím và thuật ngữ
- Tiến trình 3 bước:
 - 1. Thiết kế trình tự hộp thoại
 - 2. Xây dựng một mẫu thử
 - 3. Đánh giá khả năng sử dụng



B1. Thiết kế trình tự hộp thoại

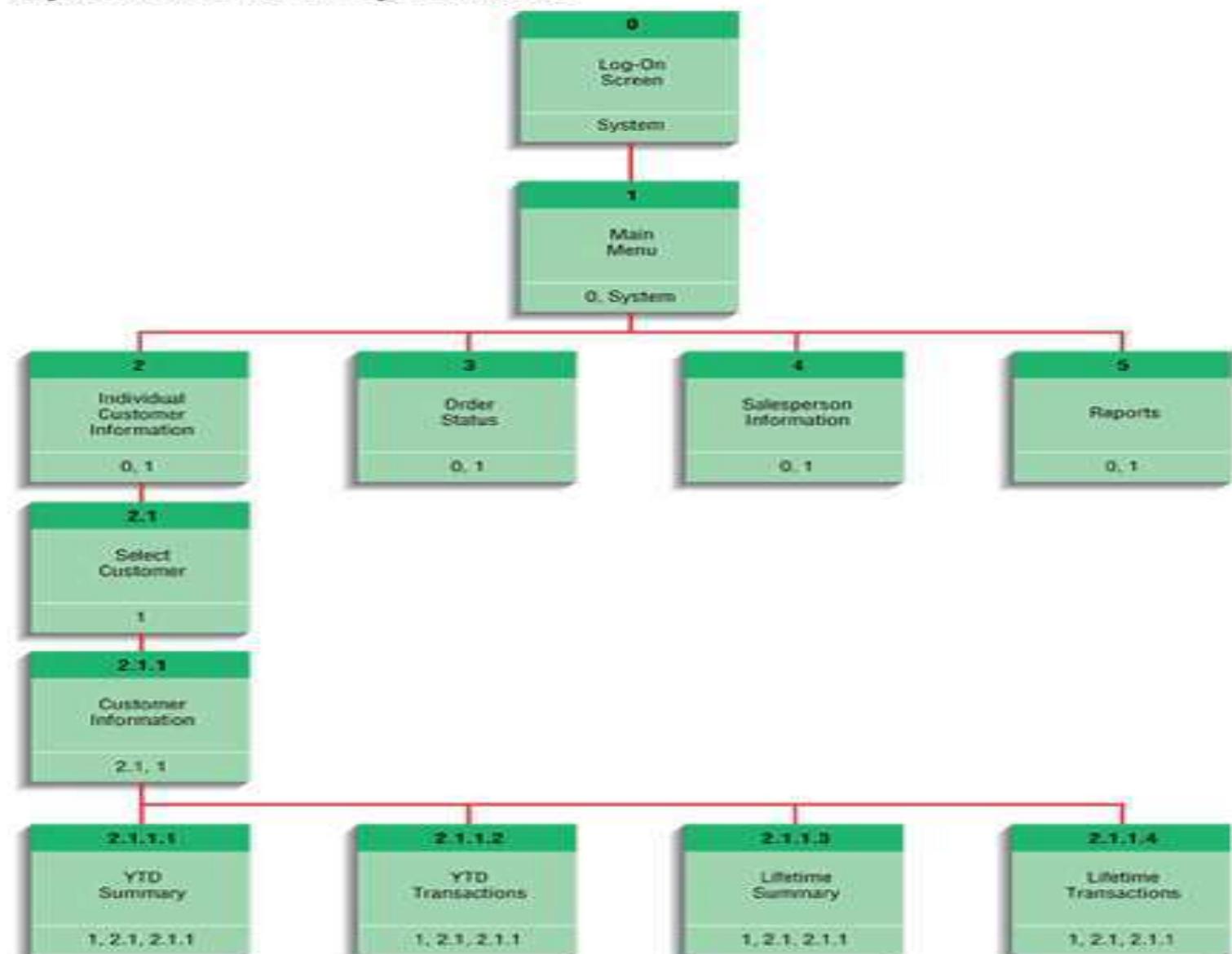
- Định nghĩa trình tự
- Có một sự hiểu biết rõ về người dùng, tác vụ, kỹ thuật và các đặc điểm của môi trường
- Sơ đồ hộp thoại
 - Một pp chính thức cho việc thiết kế và biểu diễn các hộp thoại người-máy bằng cách sử dụng các hộp và các sơ đồ đường
 - Gồm một hộp có 3 phần
 - 1. Phần đầu: số tham chiếu màn hình duy nhất được dùng bởi các màn hình khác cho việc tham chiếu hộp thoại
 - 2. Phần thân: chứa tên hoặc mô tả của màn hình
 - 3. Phần cuối: chứa các số tham chiếu màn hình có thể được truy cập từ màn hình hiện tại

B2,3. Xây dựng các Mẫu thử và Đánh giá khả năng sử dụng

- Các hoạt động của bước này thường tùy chọn
- Tác vụ được đơn giản bằng cách sử dụng mô thường thiết kế đồ họa để xây dựng nguyên mẫu nhanh chóng

- *Visual Basic.NET*
- *Các công cụ Prototypers*
- *Các công cụ Demo Builders*

Figure 8.20 Dialogue Diagram for the Customer Information System at Pine Valley Furniture



Ứng dụng TMĐT: Thiết kế Giao diện và Hộp thoại cho WebStore của Pine Valley Furniture

- Hoạt động thiết kế trung tâm và quan trọng
- Nơi nào khách hàng tương tác với công ty
 - Cẩn thận phải thể hiện trong thiết kế!
- Tiến trình thiết kế mẫu thử là thích hợp nhất để thiết kế giao diện người dùng.
- Một số nguyên tắc thiết kế chung đã xuất hiện.

Chương 4. Thiết kế các hệ thống

- 4.2 Thiết kế giao diện người dùng
 - Biểu mẫu và Báo cáo
 - Hộp thoại và Giao diện
 - Hệ thống phân tán và Internet (Đọc tài liệu)