King Saud University

College of Engineering

IE – 462: “Industrial Information Systems”

Spring – 2020 (2nd Sem. 1440-41H)

Chapter 3:

Data Modeling and Design – p1 – Introduction

Prepared by: Ahmed M. El-Sherbeeny, PhD

Lesson Overview

- **Introduction** – (p1)
- E-R Diagram – (p2)
- Case Studies – (p3)

Lesson Overview

INTRODUCTION

Introduction

- Remember DFD:
  - Technique shows *how*, *where*, and *when* data are used or changed in an information system
  - But does *not* show *definition*, *structure*, and *relationships* within the data
- Data modeling
  - It's a technique for *organizing* and *documenting* a system's data (aka database modeling)
  - Fills this crucial gap in the system
  - System developers believe this to be most important part of the statement of IS requirements

Introduction

Importance of Data Modeling:

- Necessary to capture characteristics of data in the design of databases, programs
  - e.g. customer name is limited to a specified set of values
  - e.g. product can be in only 1 product line at a time
- Data, not processes, are the most complex aspects of many modern IS
  - e.g. validating data in transaction processing, sales tracking

Introduction

Importance of Data Modeling (cont.):

- Characteristics about data should have common features for the same applications in different organizations
- Structural information about data is essential for automatic program generation
    - e.g. automatic design of a computer screen for entry of customer order

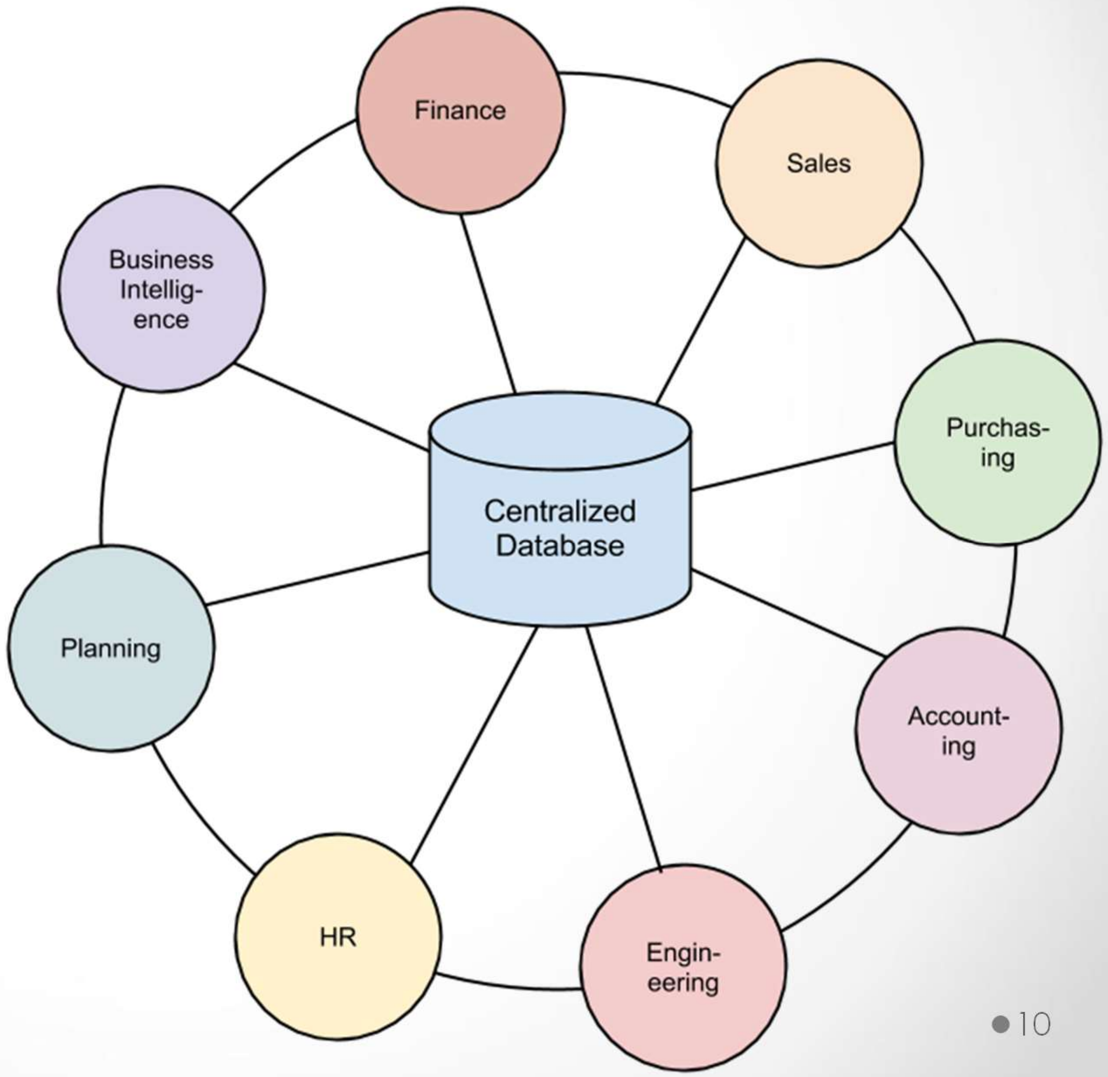
DATABASES AND DBMS

Databases

- A database is a computerized filing cabinet that stores data (i.e. collection of **records**) defined and "filed" by users within the organization
- The database system has both **hardware** and **software** components
- Hardware is the physical storage medium for the data (hard disk, CD, tape, etc.)
- Software is the medium through which the user accesses the physically stored data
    - This software is called the ***DataBase Management System (DBMS)***

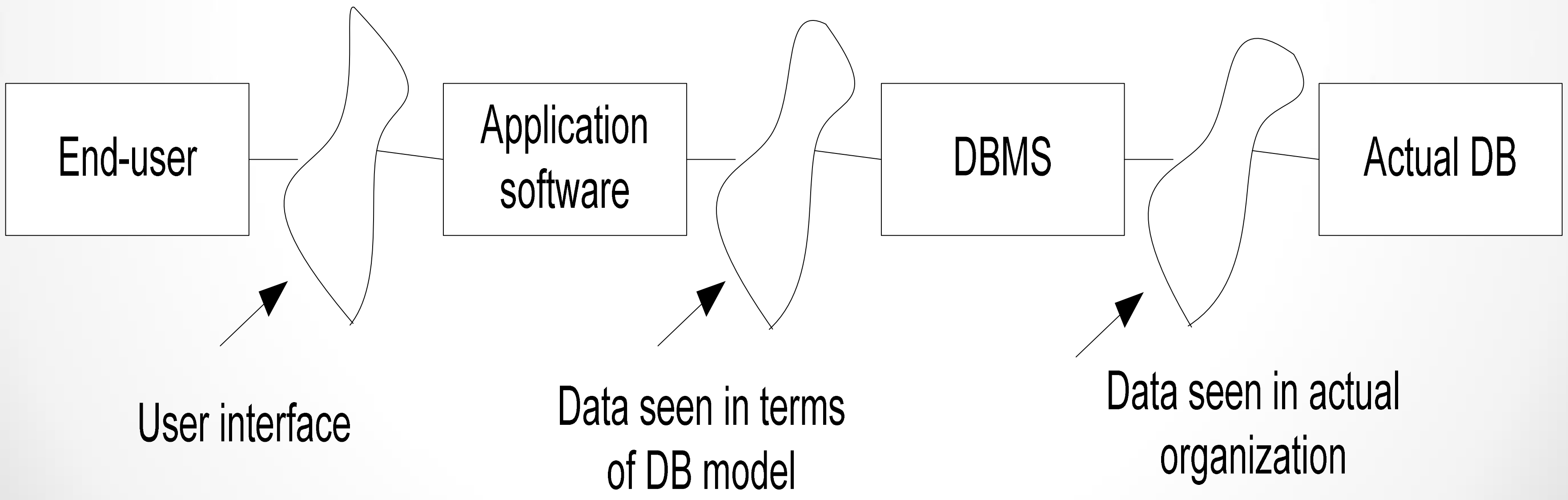
Database and Information Systems

Databases are an essential component of any information system



DBMS

- Database Management Systems (DBMS):
  - Create, control, manage, and provide use of the database (i.e. managing the database *records*)
  - Allows the user to store, retrieve, and update data


End-user
Application software
DBMS
Actual DB
User interface
Data seen in terms of DB model
Data seen in actual organization

Database Classification

We discuss several ways to classify/view databases:

- **Classes** of database systems
- Database **structures**
- Database **architecture**

DBMS Classes

DBMS Classes

There are three classes of database systems with different levels of complexity:

- **Enterprise** databases
- **Workstation** databases
- **Personal** databases

DBMS Classes (cont.)

1. Enterprise Database

- A large database that runs on *one or more servers* and may have several remote client users
- It must be capable of handling a large quantity of transactions and the execution must be in real-time
  - e.g. a transaction involving an ATM debit recorded in seconds
- DBMS like *Oracle* (Oracle Corporation) and *DB2* (IBM) are typically used for these applications

DBMS Classes (cont.)

2. Workstation/Workgroup Database

- Runs on *one server* and distributes information to several client machines running on the same local area network (LAN)
- DBMS must be capable of:
    - o handling multiple clients who are independently generating transactions, thus:
    - o changing the contents of one or more databases running concurrently on the DBMS
- Microsoft's *SQL Server*, which supports client-server architecture, is a popular choice for workgroup applications

DBMS Classes (cont.)

3. Personal Database

- A personal database runs on a single personal computer (PC)
- *Access* DBMS is a good example of a personal database

Database Structures

Database Structures

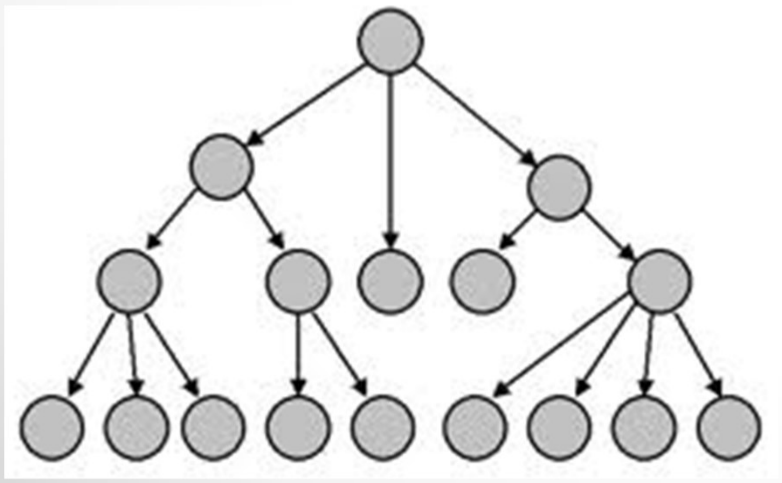
There are 4 types of database system structures:

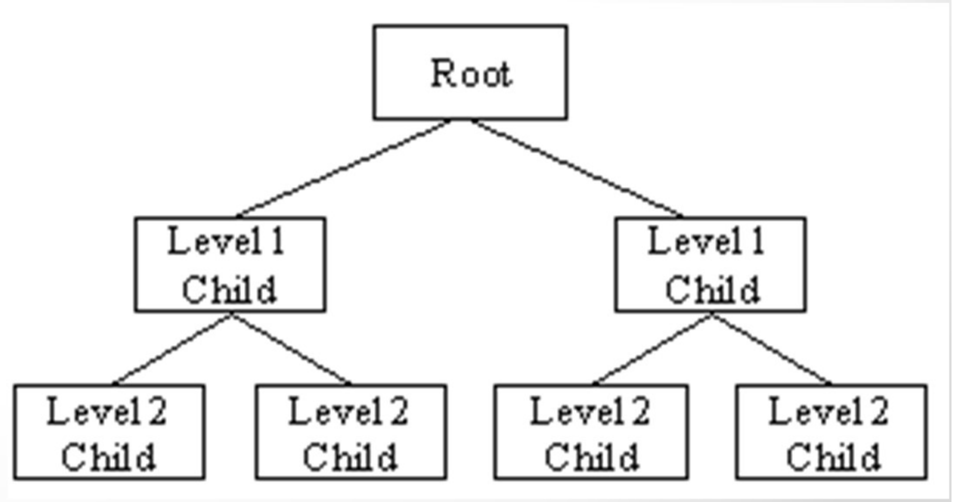
- **Hierarchical**
- **Network**
- **Object-oriented**
- **Relational**

Database Structures (cont.)

1. Hierarchical Database

- Records contain information about parent/child relationships, just like a tree structure
- Data can be accessed and updated rapidly
- Each child in the tree may have only *one parent*, and relationships/linkages between children are not permitted

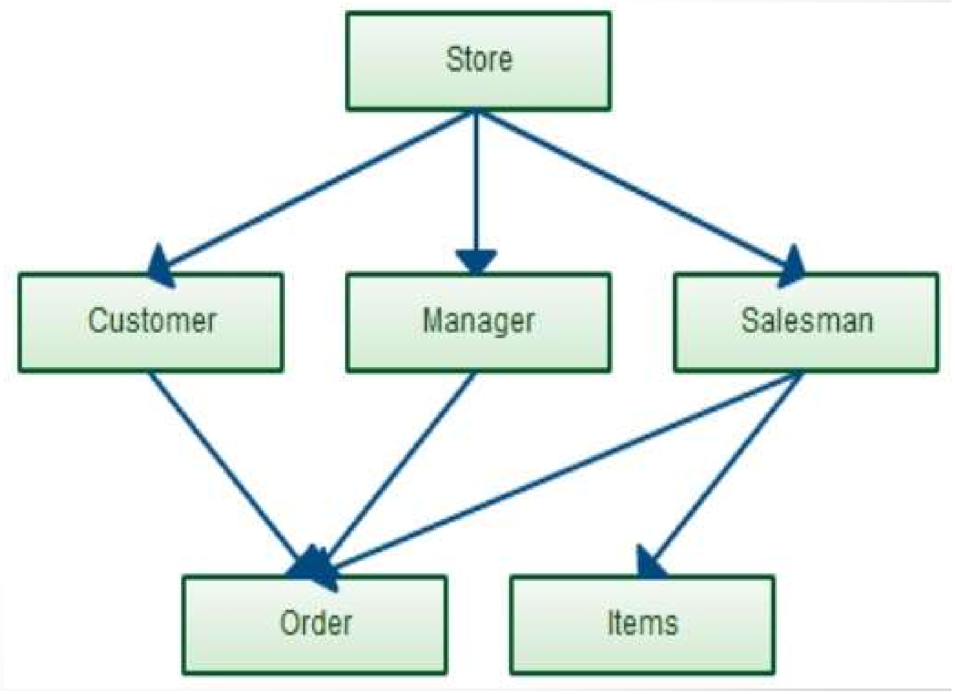




Database Structures (cont.)

2. Network Database

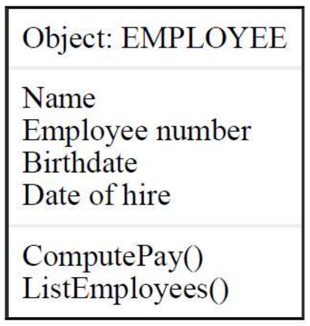
- Network databases are mainly used on large digital computers
- Also has a hierarchical structure (cobweb or interconnected network of records)
- Each child can have more than one parent (i.e. suitable for *many-to-many* relationships in data)



Database Structures (cont.)

3. Object-Oriented Database

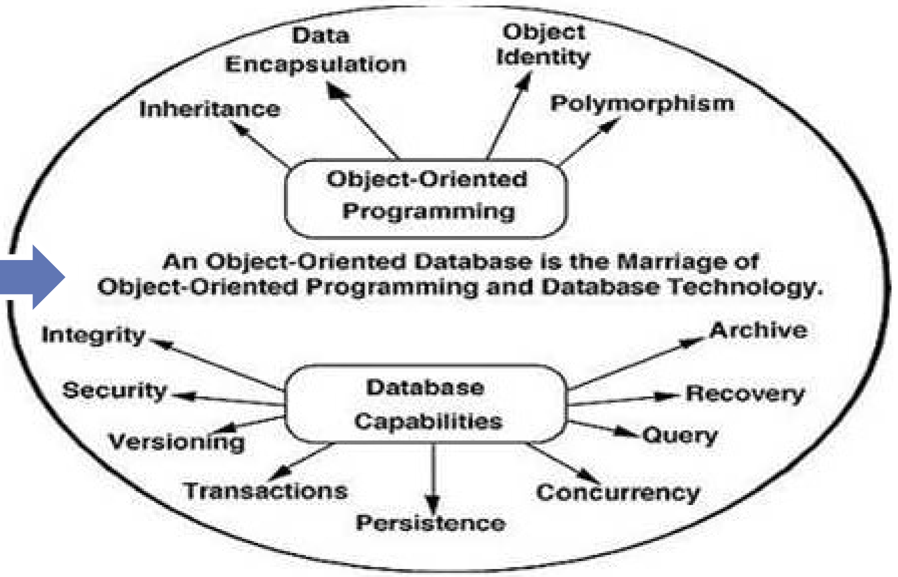
- Information is represented in the form of objects (which are themselves stored in the OO database)
- Each object consists of two elements:
  - Data (e.g. sound, video, text, or graphics)
  - Instructions for what to do with the data
- More expensive to develop, but provide powerful multimedia capability



Object: EMPLOYEE

Name
Employee number
Birthdate
Date of hire

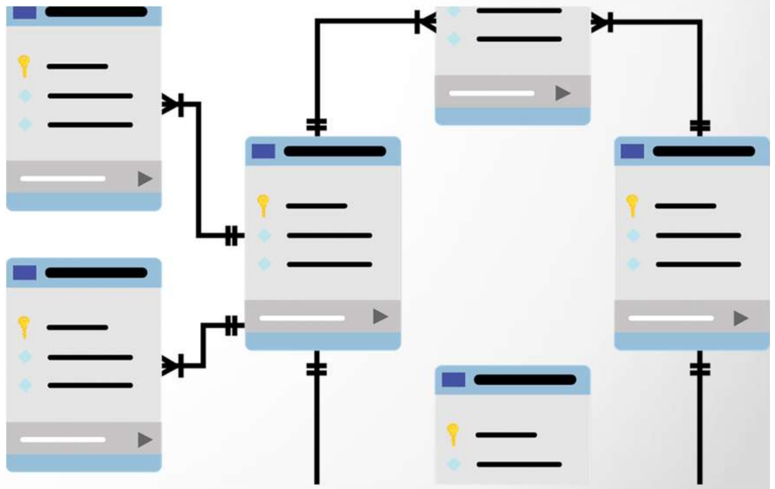
ComputePay()
ListEmployees()



Database Structures (cont.)

4. Relational Database

- Data is stored in tables, each having a *key field* that identifies each row
- This model is more reliable than either the hierarchical or network database structures
- It can be used with little or no training
- It is the foundation of modern DBMS; we will discuss this model in detail in this chapter



DBMS Architecture

DBMS Architecture

- Data representation should be done independent of how data are stored & manipulated in the computer
- General architecture for data representation:
    - Developed in 1975 by ANSI/SPARC (**S**tandards **P**lanning **A**nd **R**equirements **C**ommittee of the **A**merican **N**ational **S**tandards **I**nstitute)
    - 3-level architecture based on 3 views of the data in the database:
        1. **External level**
        2. **Conceptual level**
        3. **Internal level**

DBMS Architecture

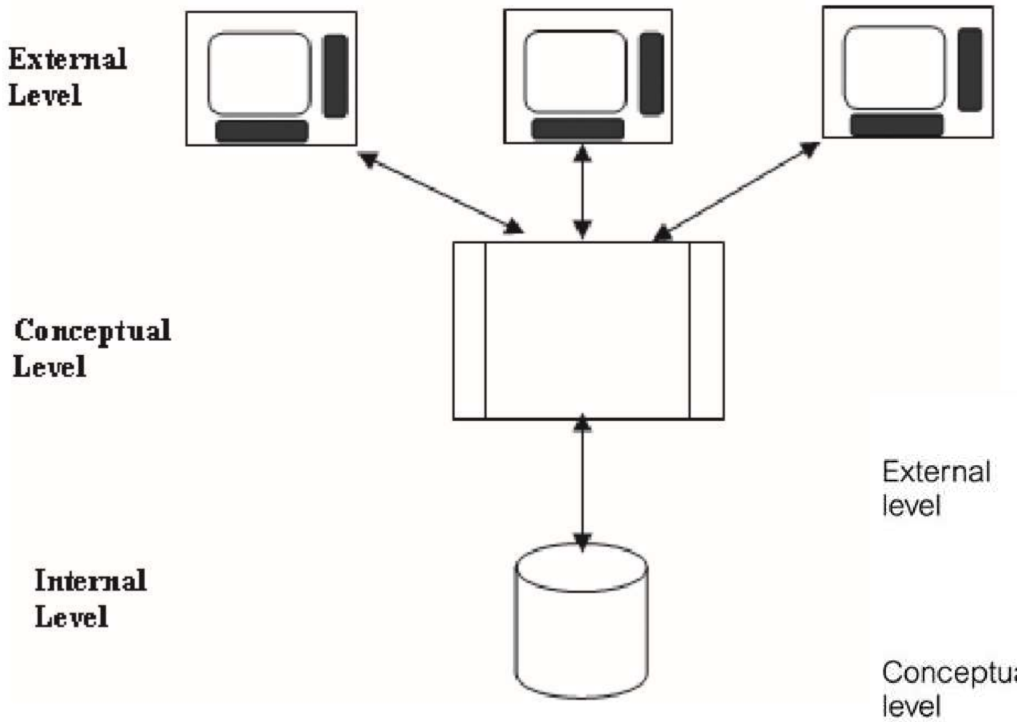
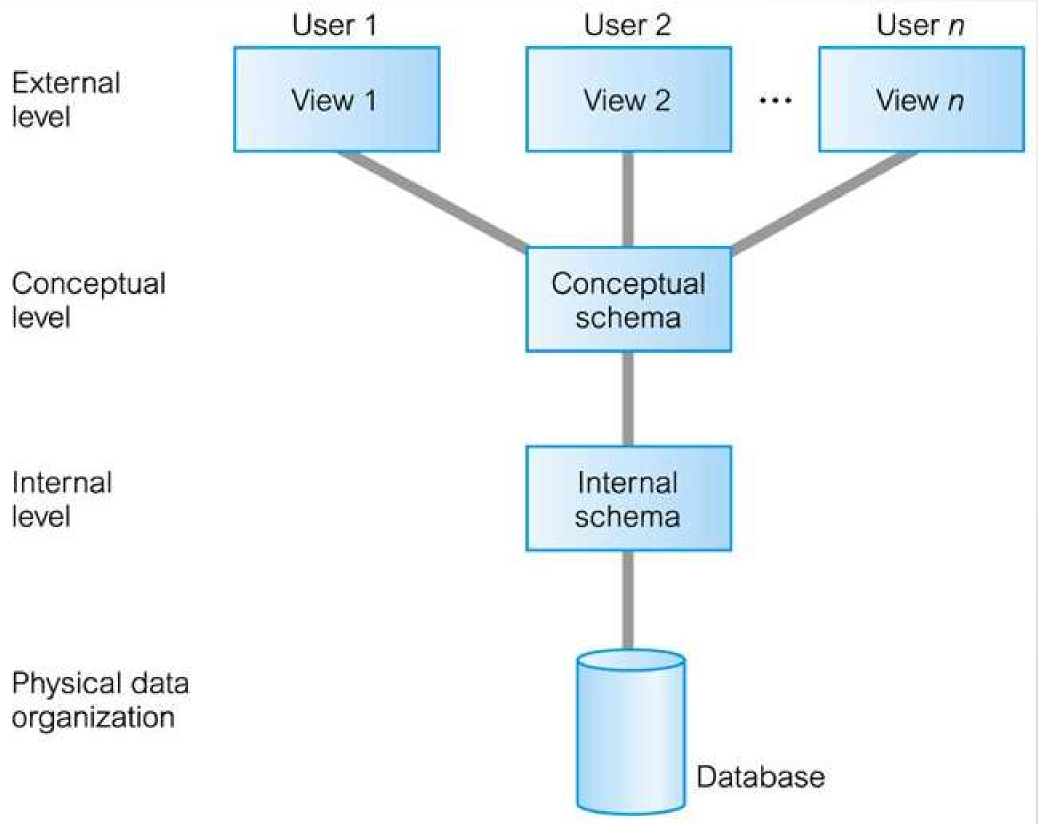


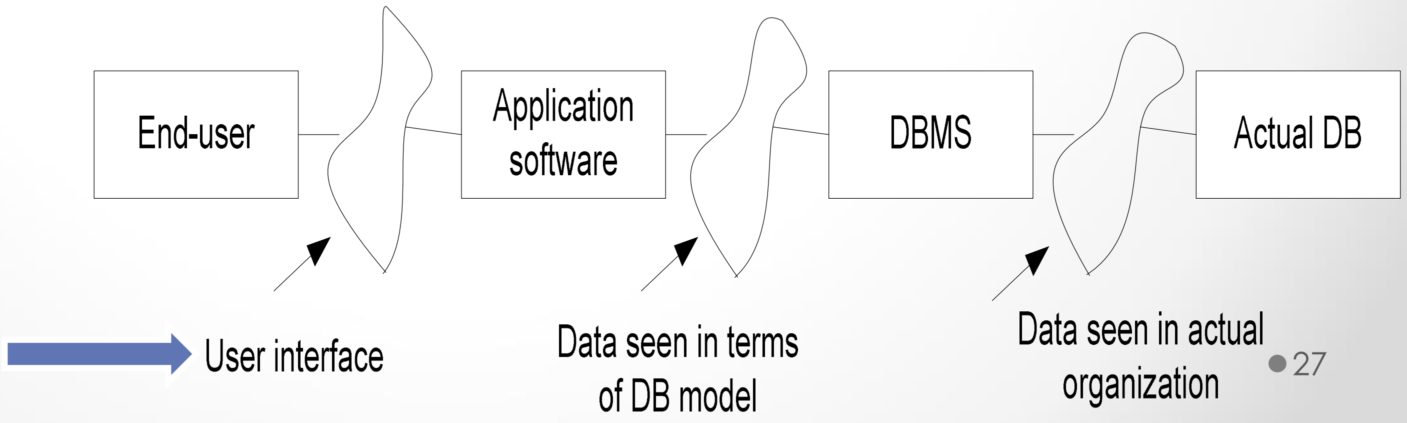
Figure 3.1 ANSI/SPARC three-level architecture.



DBMS Architecture

1. External Level

- External level addresses the way in which different users view the database
- Includes *entities* & *attributes* that the user sees and interacts with
- Implementation at the external level involves user interfaces (“forms”, “reports”, summary statistics, etc.) used to interact with the database


End-user
Application software
DBMS
Actual DB
User interface
Data seen in terms of DB model
Data seen in actual organization

DBMS Architecture

3. Internal Level

- Internal level addresses data structures and the file organization used to store data within the computer
- Properties of internal level:
  - defines *how* data are stored*
  - i.e. includes data compression, data encryption, use of indexes, and other details
  - *dependent* on operating system and physical components of computer system on which the database resides

DBMS Architecture

2. Conceptual Level

- Conceptual level is a holistic (i.e. complete system / "big picture") view of the database
- Properties of conceptual level:
  - defines *entities*, their *attributes*, and their *relationships* (to be discussed in E-R diagram lesson in detail)
  - describes *what* data are stored in the database but not *how* they are stored
  - i.e. it's a *logical description* of the database without saying anything about its implementation
  - independent of specific hardware/software platform

DBMS Architecture

2. Conceptual Level (cont.)

- We focus on this conceptual/logical design of the database, aka design of a data model
- Design of a data model provides representation of:
  - *entities* in the enterprise
  - *attributes* of those entities, and
  - *relationships* that exist among entities

DBMS Architecture

2. Conceptual Level (cont.)

- **Conceptual data model:**
  - a detailed data model
  - captures *overall structure* of organizational data
  - shows rules about the *meaning* and *interrelationships* among data
  - *independent* of any DBMS, implementation considerations, how data is stored in memory
  - usually, a subset of the project development team concentrates on *data modeling* while other team members focus attention on *process modeling*

DBMS Architecture

2. Conceptual Level (cont.)

- **Conceptual data model** (cont.):
  - work of all team members is coordinated and shared through the *project repository*
  - repository is maintained by a common CASE or data modeling software tool
  - process and data model descriptions of system must be consistent/complete since they describe different, but complementary, views of the same IS
  - e.g. names of *data stores* on primitive level DFDs should correspond to names of *data entities* in E-R diagrams

DBMS Architecture

2. Conceptual Level (cont.)

- Conceptual data modeling techniques (originated by Peter Chen in the late 1970s) include:
    - **Entity-relationship** (E-R) modeling; most commonly used technique/format
    - **Class diagramming**; similar format to E-R modeling, used with object-oriented analysis & design methods
    - **Integrated Computer-Aided Manufacturing Definition 1, extended (IDEF1X)**; specifically designed for manufacturing applications

CONCEPTUAL DATA MODELING PROCESS

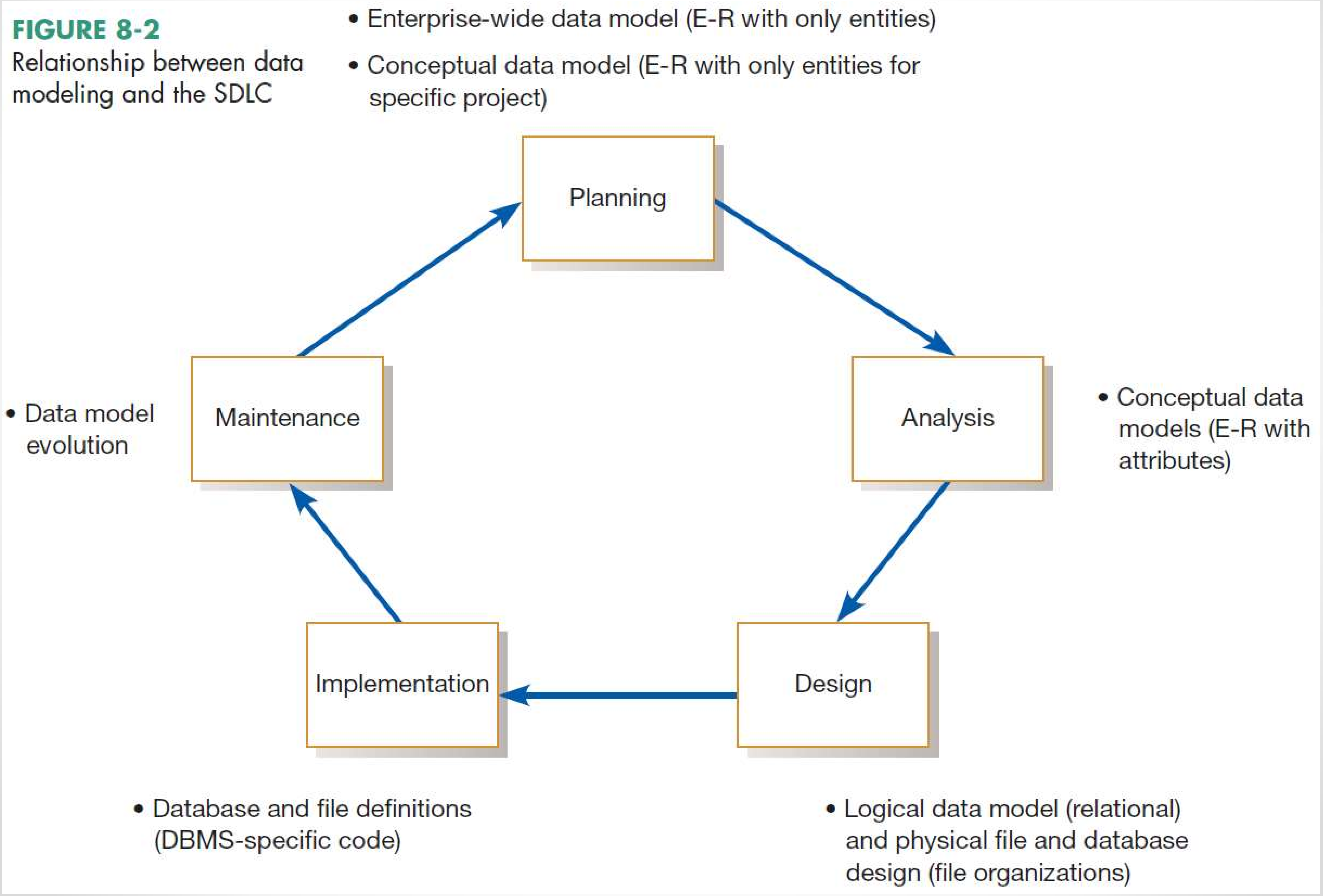
Conceptual Data Modeling Process

Conceptual Data Modeling Steps

- Process begins with developing a conceptual data model for system being replaced (if system already exists)
- Then new conceptual data model is built including all of the data requirements for the new system
- Modeling is iterative process with many checkpoints; uses *rapid development methodologies*
- Conceptual modeling methods are suitable for the *planning* and *analysis phases* of the *development life cycle* (SDLC) (see next slide)

Conceptual Data Modeling Process

FIGURE 8-2
Relationship between data modeling and the SDLC


• Enterprise-wide data model (E-R with only entities)
• Conceptual data model (E-R with only entities for specific project)
Planning
Analysis
• Conceptual data models (E-R with attributes)
Design
• Logical data model (relational) and physical file and database design (file organizations)
Implementation
• Database and file definitions (DBMS-specific code)
Maintenance
• Data model evolution

Conceptual Data Modeling Process

Deliverables

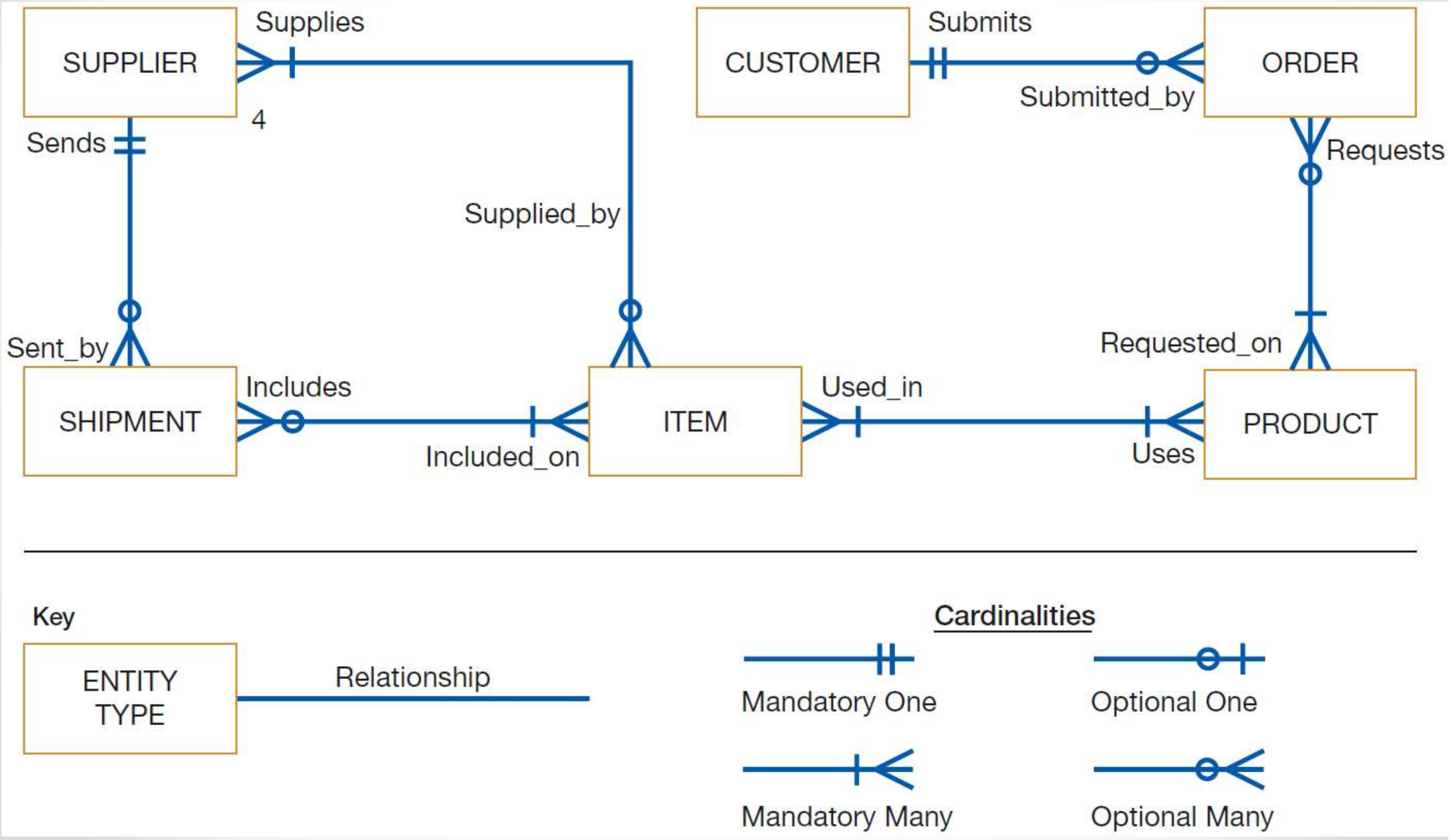
Deliverable 1:

- Primary deliverable from conceptual data modeling step (analysis phase) is an ***E-R diagram***
- E-R diagram shows(next slide):
  - major *categories* of data (rectangles on diagram)
  - *business relationships* between them (lines connecting rectangles)

Conceptual Data Modeling Process

FIGURE 8-3

Sample conceptual data model



Conceptual Data Modeling Process

Deliverables (cont.)

Deliverable 1 (cont.):

- Example from diagram:
  - a **SUPPLIER** *sometimes* Supplies **ITEM**s to company (company wants to keep track of some suppliers *without* designating what they can supply)
  - ITEM is always Supplied by *one to four* SUPPLIERS
- Diagram includes on each line,
  - two names (so that a relationship can be read in either direction) –*test it*!
  - note, some standards include only 1 name/line

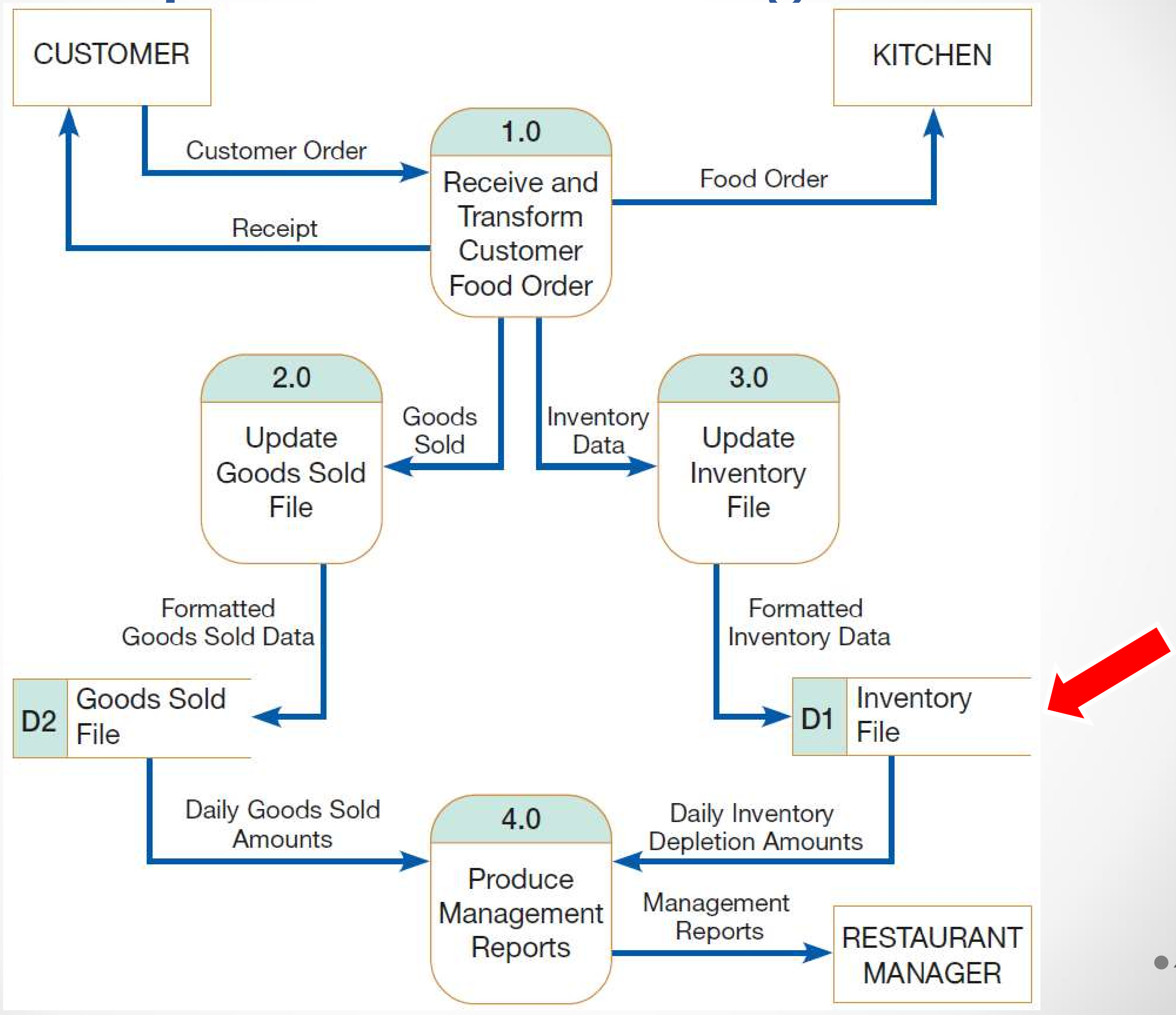
Conceptual Data Modeling Process

Deliverables (cont.)

Deliverable 2:

- Deliverable 2: entries about data objects that will be stored in ***project repository***, or data modeling software
- Repository is mechanism that links *data* and *process models* of an IS (e.g. links between data model & DFD):
  - Data elements included in *data flows* also appear in the *data model*, and vice versa; note, data flows are captured by manual or automated *data stores*
  - Each *data store* in a process model must relate to *business objects* (aka *data entities*); e.g. *Inventory File data* (*Hoosier Burger*) must correspond to ≥ 1 data objects on a data model

Conceptual Data Modeling Process



GATHERING INFORMATION FOR CONCEPTUAL DATA MODELING

Gathering Info. for Conceptual Data Modeling

- During *Joint Application Design* (JAD) sessions/ interviews you must ask specific Q's in order to gain the perspective on data you need for the data model
- These Q's relate to:
  - explaining *what* the organization does
  - rules of *how* work is performed in the organization
  - not how or when data are processed or used to do data modeling
- Ways to gather this information:
  1. **Top-down approach**
  2. **Bottom-up approach**

Gathering Info. for Conceptual Data Modeling

1. Top-Down Approach

- Top-down approach:
  - derives business rules for a data model from proper understanding of nature of business
  - usually used with a purchased data model
- Table 8-1:
  - key Q's to ask system users & business managers
  - help to develop accurate & complete data model, for a particular situation
  - ask these Q's when you begin data modeling project with a purchased data model
  - note, don't use *technical terms* (in bold); instead, frame your Q's in *business terms* for manager

Gathering Info. for Conceptual Data Modeling

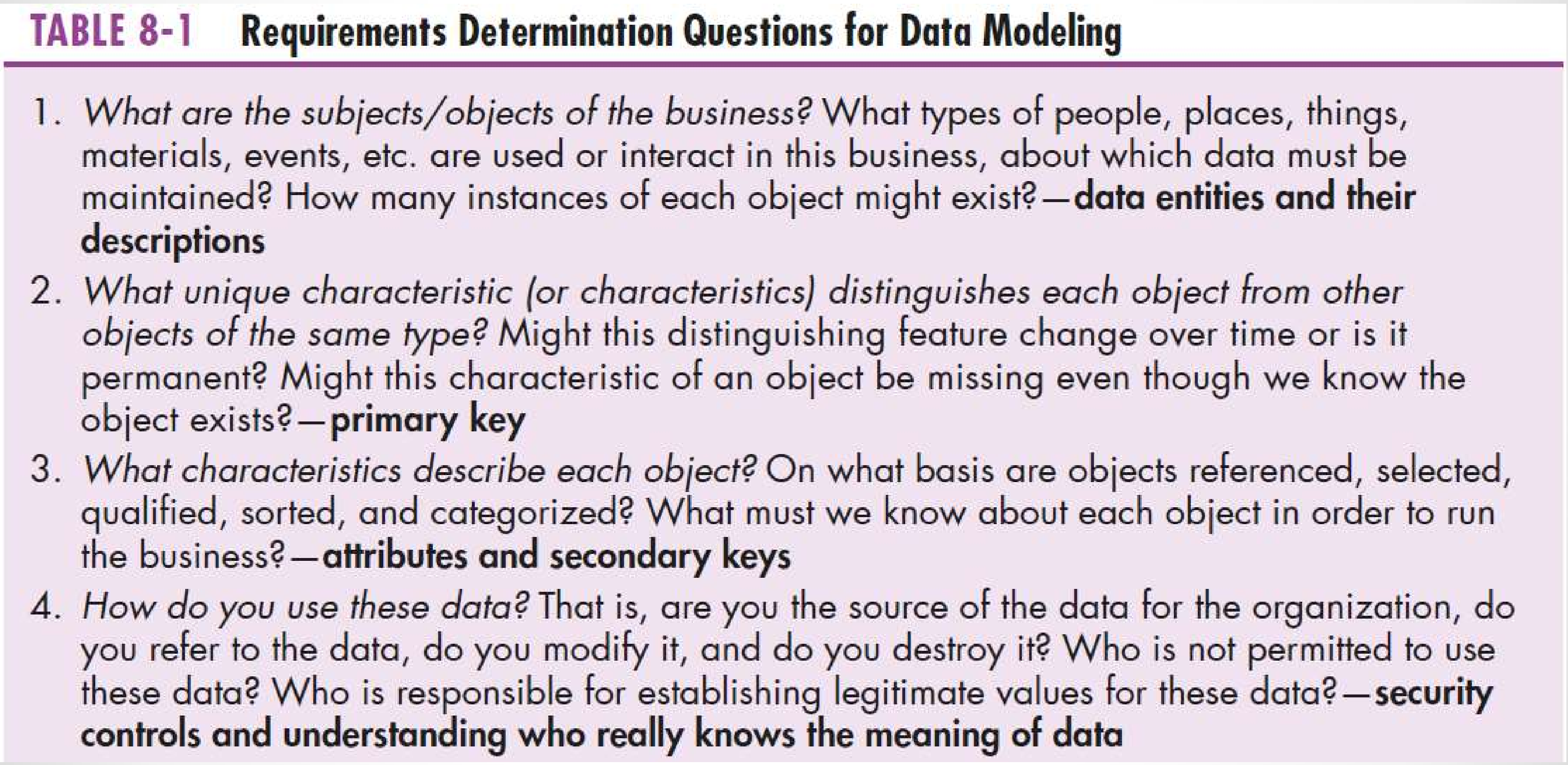


TABLE 8-1 Requirements Determination Questions for Data Modeling

1. *What are the subjects/objects of the business?* What types of people, places, things, materials, events, etc. are used or interact in this business, about which data must be maintained? How many instances of each object might exist?—**data entities and their descriptions**
2. *What unique characteristic (or characteristics) distinguishes each object from other objects of the same type?* Might this distinguishing feature change over time or is it permanent? Might this characteristic of an object be missing even though we know the object exists?—**primary key**
3. *What characteristics describe each object?* On what basis are objects referenced, selected, qualified, sorted, and categorized? What must we know about each object in order to run the business?—**attributes and secondary keys**
4. *How do you use these data?* That is, are you the source of the data for the organization, do you refer to the data, do you modify it, and do you destroy it? Who is not permitted to use these data? Who is responsible for establishing legitimate values for these data?—**security controls and understanding who really knows the meaning of data**

Gathering Info. for Conceptual Data Modeling

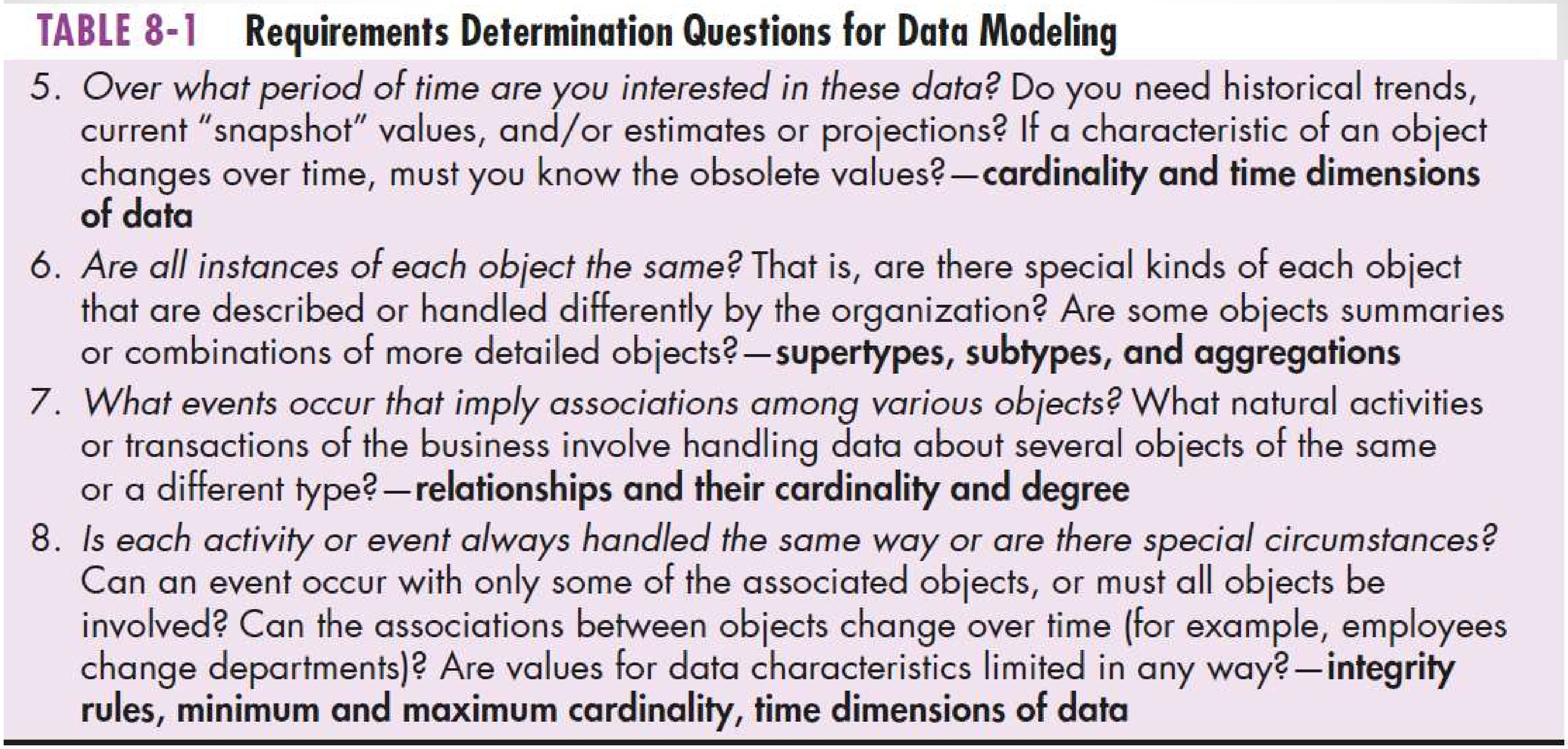


TABLE 8-1 Requirements Determination Questions for Data Modeling

5. *Over what period of time are you interested in these data?* Do you need historical trends, current "snapshot" values, and/or estimates or projections? If a characteristic of an object changes over time, must you know the obsolete values?—**cardinality and time dimensions of data**
6. *Are all instances of each object the same?* That is, are there special kinds of each object that are described or handled differently by the organization? Are some objects summaries or combinations of more detailed objects?—**supertypes, subtypes, and aggregations**
7. *What events occur that imply associations among various objects?* What natural activities or transactions of the business involve handling data about several objects of the same or a different type?—**relationships and their cardinality and degree**
8. *Is each activity or event always handled the same way or are there special circumstances?* Can an event occur with only some of the associated objects, or must all objects be involved? Can the associations between objects change over time (for example, employees change departments)? Are values for data characteristics limited in any way?—**integrity rules, minimum and maximum cardinality, time dimensions of data**

Gathering Info. for Conceptual Data Modeling

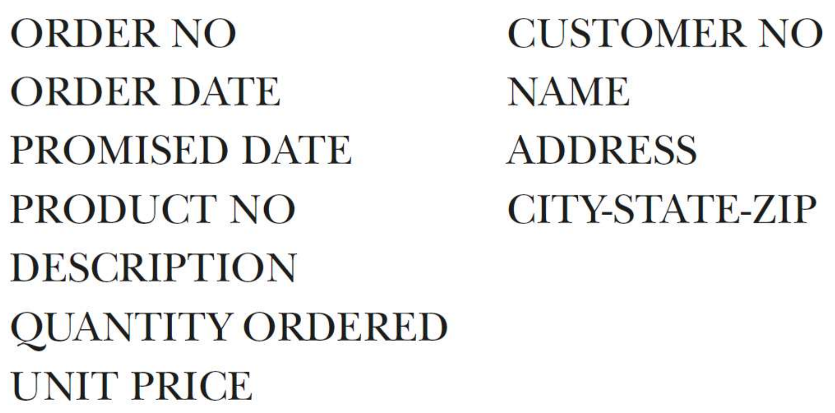
2. Bottom-Up Approach

- You can also gather needed info. for data modeling by reviewing specific *business documents* used in IS:
  - computer displays
  - reports and business forms
- This is usually represented as:
  - data flows on DFDs, and this shows:
  - data processed by the system $\Rightarrow$ i.e. data that must be maintained in system's database

Gathering Info. for Conceptual Data Modeling

2. Bottom-Up Approach (cont.)

- Example: customer order form used at *Pine Valley Furniture* (PVF)
  - following data must be kept in the database:



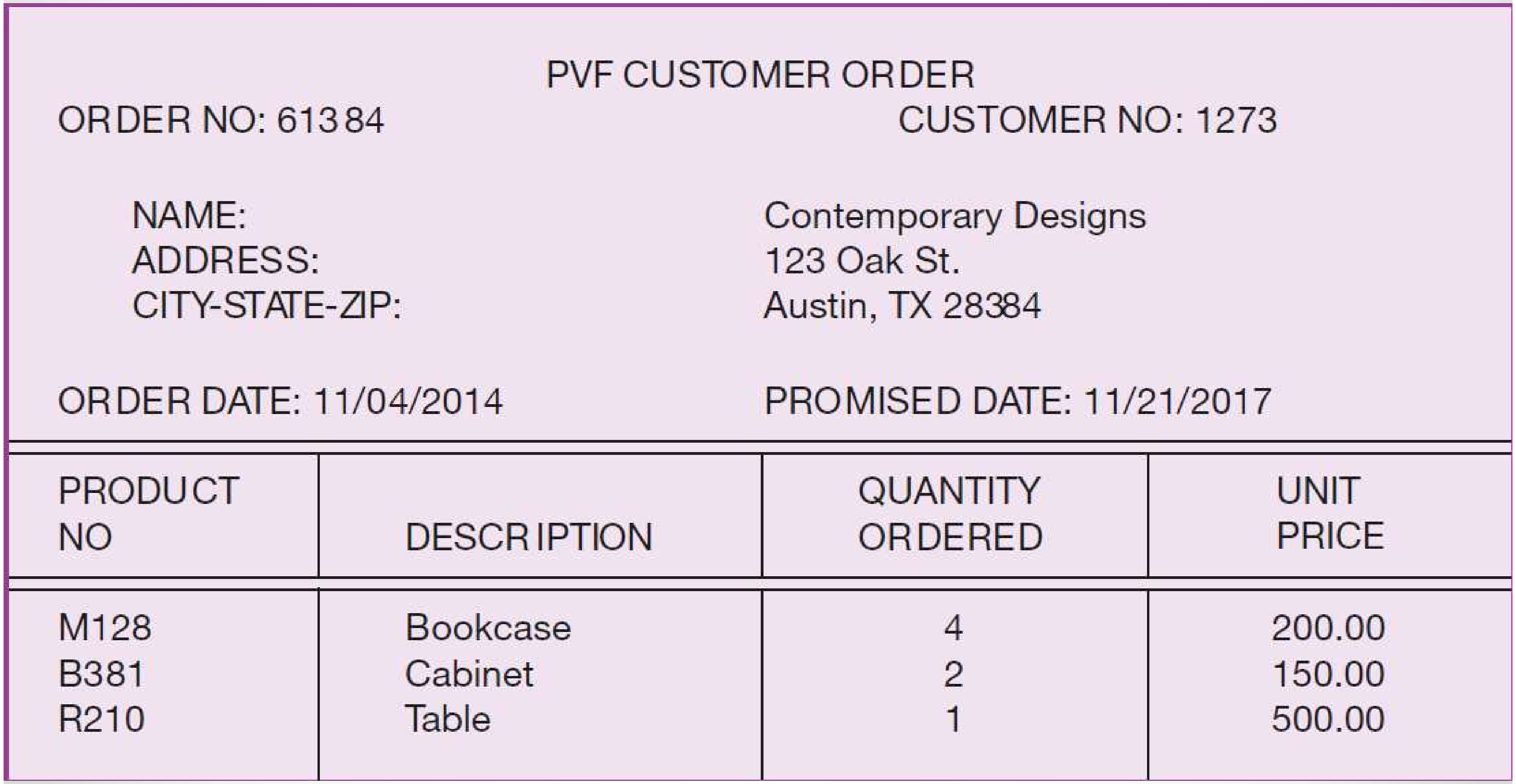
ORDER NO
ORDER DATE
PROMISED DATE
PRODUCT NO
DESCRIPTION
QUANTITY ORDERED
UNIT PRICE

CUSTOMER NO
NAME
ADDRESS
CITY-STATE-ZIP

  - we also see important info. needed for data model:
    - each order is from one customer,
    - order can have multiple line items, 1 per product

Gathering Info. for Conceptual Data Modeling

FIGURE 8-4
Sample customer form



PVF CUSTOMER ORDER

ORDER NO: 61384 CUSTOMER NO: 1273

NAME:	Contemporary Designs
ADDRESS:	123 Oak St.
CITY-STATE-ZIP:	Austin, TX 28384

ORDER DATE: 11/04/2014 PROMISED DATE: 11/21/2017

PRODUCT NO	DESCRIPTION	QUANTITY ORDERED	UNIT PRICE
M128	Bookcase	4	200.00
B381	Cabinet	2	150.00
R210	Table	1	500.00

Videos to Watch

- **Entity Relationship Diagram (ERD) Tutorial - Part 1**
  https://youtu.be/QpdhBUYk7Kk
- **Entity Relationship Diagram (ERD) Tutorial - Part 2**
  https://youtu.be/-CuY5ADwn24
- **Entity-Relationship Diagrams** (another system)
  https://youtu.be/c0_9Y8QAstg
- **Entity Relationship Diagram (ERD) Training Video**
  https://youtu.be/-fQ-bRllhXc

Sources

- "**Chapter 3: Database Modeling and Design**"; Slides by Dr. Sabeur Kosantini (2017)
- "**Types of Database Management Systems**" (2017) by Arjun Panwar, c-sharpcorner.com; Available at: https://www.c/sharpcorner.com/UploadFile/65fc13/types-of-database-management-systems/
- **Modern Systems Analysis and Design**. Joseph S. Valacich and Joey F. George. Pearson. Eighth Ed. 2017. Chapter 8.
- **Design of Industrial Information Systems**. Thomas Boucher, and Ali Yalcin. Academic Press. First Ed. 2006. Chapter 3.