Database Systems

Dr./ Ahmed Osman Mahmoud

Lecture # 1

Reference

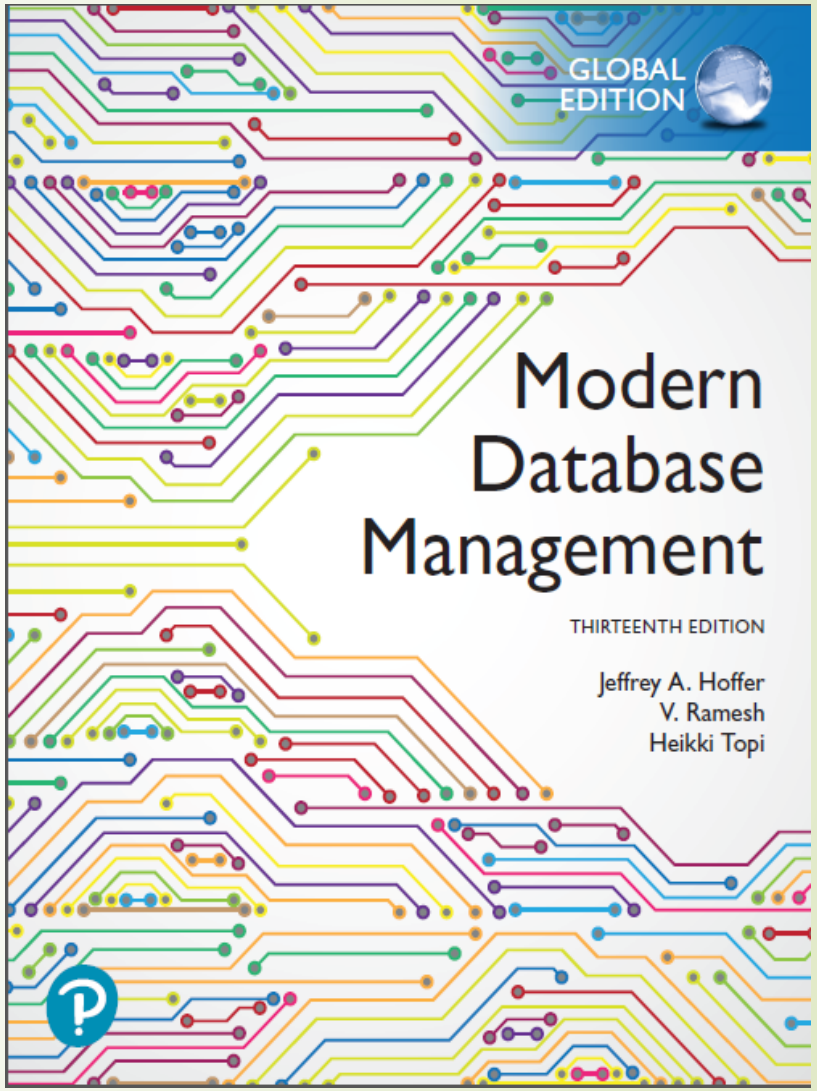
Textbook

MODERN DATABASE MANAGEMENT

Jeffrey A. Hoffer
University of Dayton

V. Ramesh
Indiana University

Heikki Topi
Bentley University



Chapter 1:
The Database Environment and Development Process

Objective

- Define terms
- Name limitations of conventional file processing
- Explain advantages of databases
- Identify costs and risks of databases
- List components of database environment
- Identify categories of database applications
- Describe database system development life cycle
- Explain prototyping and agile development approaches
- Explain roles of individuals
- Explain the three-schema architecture for databases

Definitions

- **Database**: organized collection of logically related data
  - A database may be of any size and complexity.
  - For example, a salesperson may maintain a small database of customer contacts—consisting of a few megabytes of data—on her laptop computer.
  - A large corporation may build a large database consisting of several terabytes of data (a *terabyte* is a trillion bytes) on a large mainframe computer that is used for decision support applications.
  - Very large data warehouses contain more than a petabyte of data. (A *petabyte* is a quadrillion bytes.)

Definitions

- **Data**: referred to facts concerning objects and events that could be recorded and stored on computer media.
  - For example, in a salesperson's database, the data would include facts such as customer name, address, and telephone number. This type of data is called structured data.
  - The most important structured data types are numeric, character, and dates. Structured data are stored in tabular form.
  - For example, the salesperson's database might include a photo image of the customer contact. It might also include a sound recording or video clip about the most recent product. This type of data is referred to as unstructured data, or as multimedia data.
  - Structured and unstructured data are often combined in the same database to create a true multimedia environment.

Definitions

- **Information:** is data that have been processed in such a way that the knowledge of the person who uses the data is increased.
  - For example, consider the following list of facts: These facts satisfy our definition of data, but most people would agree that the data are useless in their present form. Even if you guessed that this is a list of people's names paired with their Social Security numbers; the data remain useless because you would have no idea what the entries mean.

Baker, Kenneth D.	324917628
Doyle, Joan E.	476193248
Finkle, Clive R.	548429344
Lewis, John C.	551742186
McFerran, Debra R.	409723145

Definitions

- what happens when you see the same data in a context, as shown in Figure.

Class Roster

Course: MGT 500 Business Policy Semester: Spring 2018

Section: 2

Name	ID	Major	GPA
Baker, Kenneth D.	324917628	MGT	2.9
Doyle, Joan E.	476193248	MKT	3.4
Finkle, Clive R.	548429344	PRM	2.8
Lewis, John C.	551742186	MGT	3.7
McFerran, Debra R.	409723145	IS	2.9
Sisneros, Michael	392416582	ACCT	3.3

FIGURE 1-1 Converting data to information

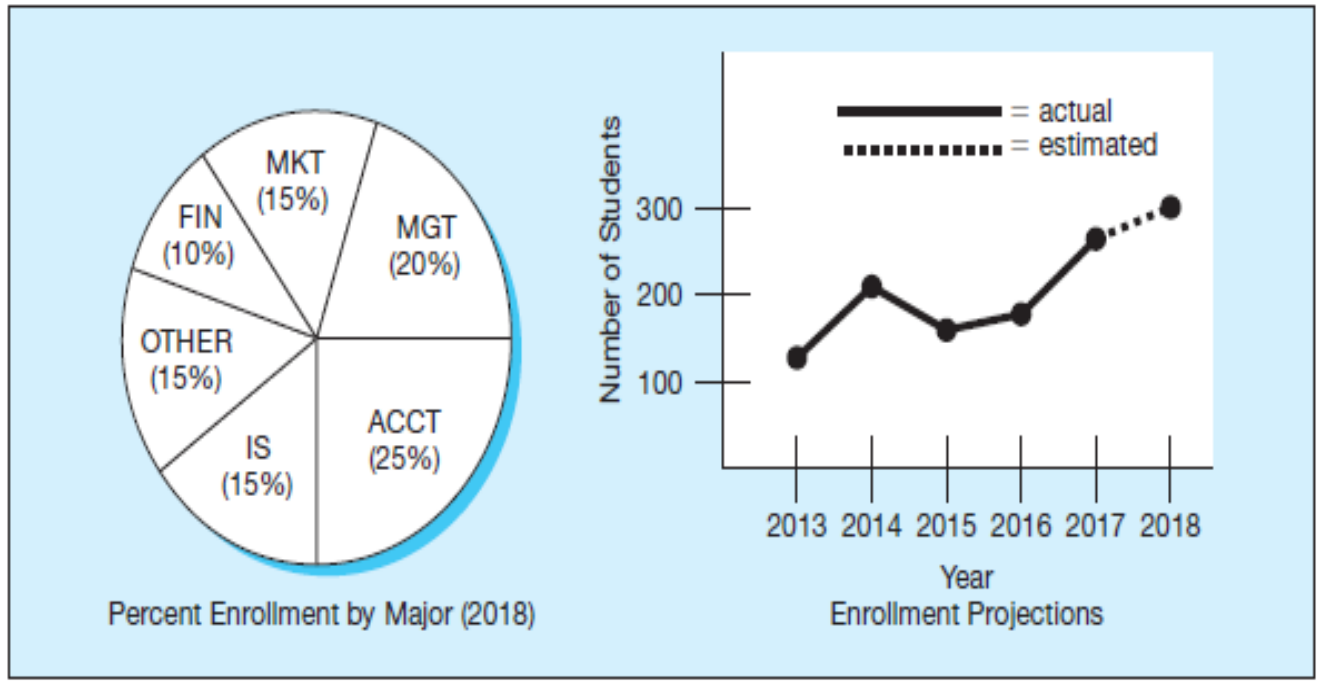
(a) Data in context

Definitions

- what happens when you see the same data in a context, as shown in Figure.

FIGURE 1-1 (*continued*)

(b) Summarized data



Graphical displays turn data into useful information that managers can use for decision making and interpretation

Definitions

- **Metadata:** are data that describe the properties or characteristics of end-user data and the context of that data.

Descriptions of the properties or characteristics of the data, including data types, field sizes, allowable values, and data context

TABLE 1-1 Example Metadata for Class Roster

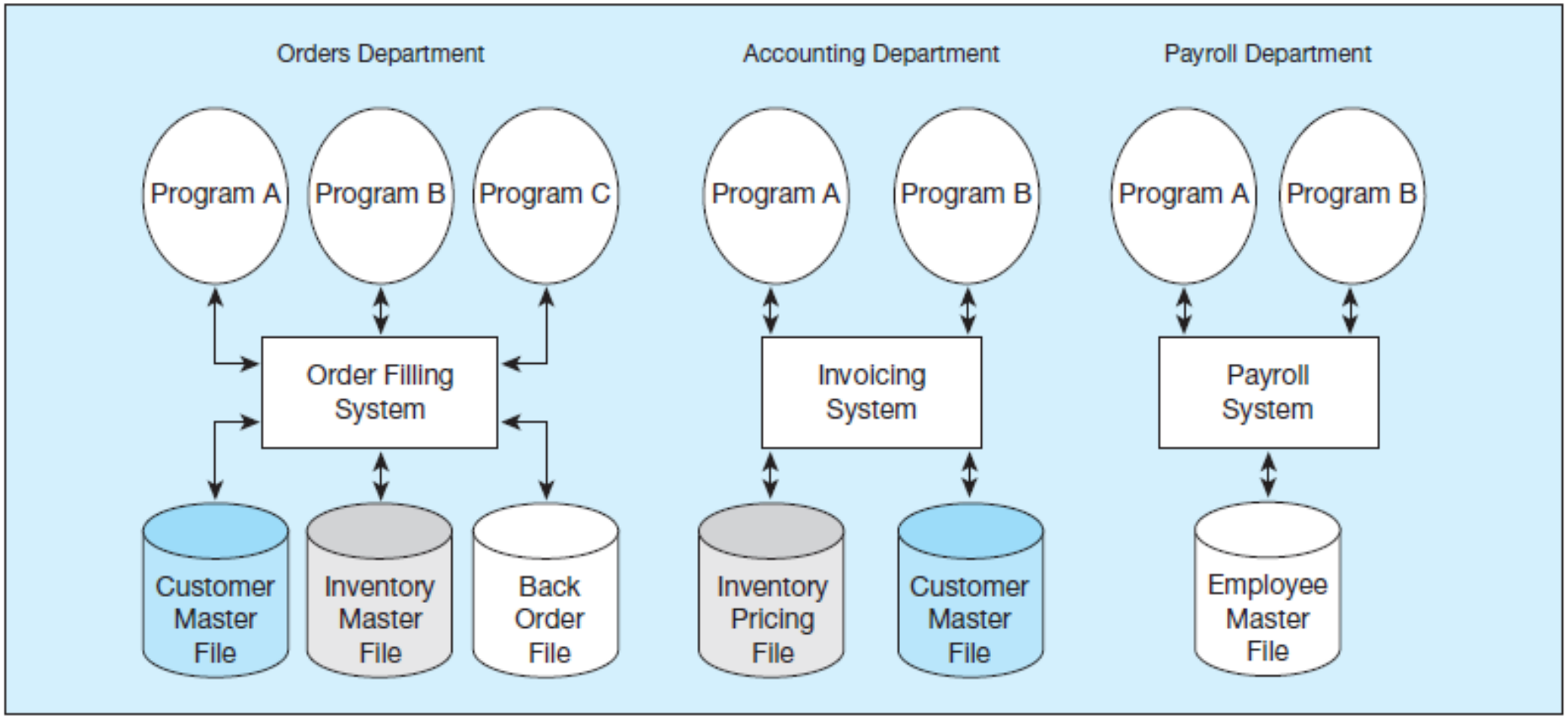
Data Item	Metadata					
Name	**Type**	**Length**	**Min**	**Max**	**Description**	**Source**
Course	Alphanumeric	30			Course ID and name	Academic Unit
Section	Integer	1	1	9	Section number	Registrar
Semester	Alphanumeric	10			Semester and year	Registrar
Name	Alphanumeric	30			Student name	Student IS
ID	Integer	9			Student ID (SSN)	Student IS
Major	Alphanumeric	4			Student major	Student IS
GPA	Decimal	3	0.0	4.0	Student grade point average	Academic Unit

TRADITIONAL FILE PROCESSING SYSTEMS

- When computer-based data processing was first available, there were no databases.
- To be useful for business applications, computers had to store, manipulate, and retrieve large files of data.
- As business applications became more complex, it became evident that traditional file processing systems had a number of shortcomings and limitations.

TRADITIONAL FILE PROCESSING SYSTEMS

FIGURE 1-2 Old file processing systems at Pine Valley Furniture Company



Disadvantages of File Processing Systems

- Program-Data Dependence
  - All programs maintain metadata for each file they use
- Duplication of Data
  - Different systems/programs have separate copies of the same data
- Limited Data Sharing
  - No centralized control of data
- Lengthy Development Times
  - Programmers must design their own file formats
- Excessive Program Maintenance
  - 80% of information systems budget

Problems with Data Dependency

- Each application programmer must maintain his/her own data
- Each application program needs to include code for the metadata of each file
- Each application program must have its own processing routines for reading, inserting, updating, and deleting data
- Lack of coordination and central control
- Non-standard file formats

Problems with Data Redundancy

- Waste of space to have duplicate data
- Causes more maintenance headaches
- The biggest problem:
  - **Data changes in one file could cause inconsistencies**
  - Compromises in ***data integrity***

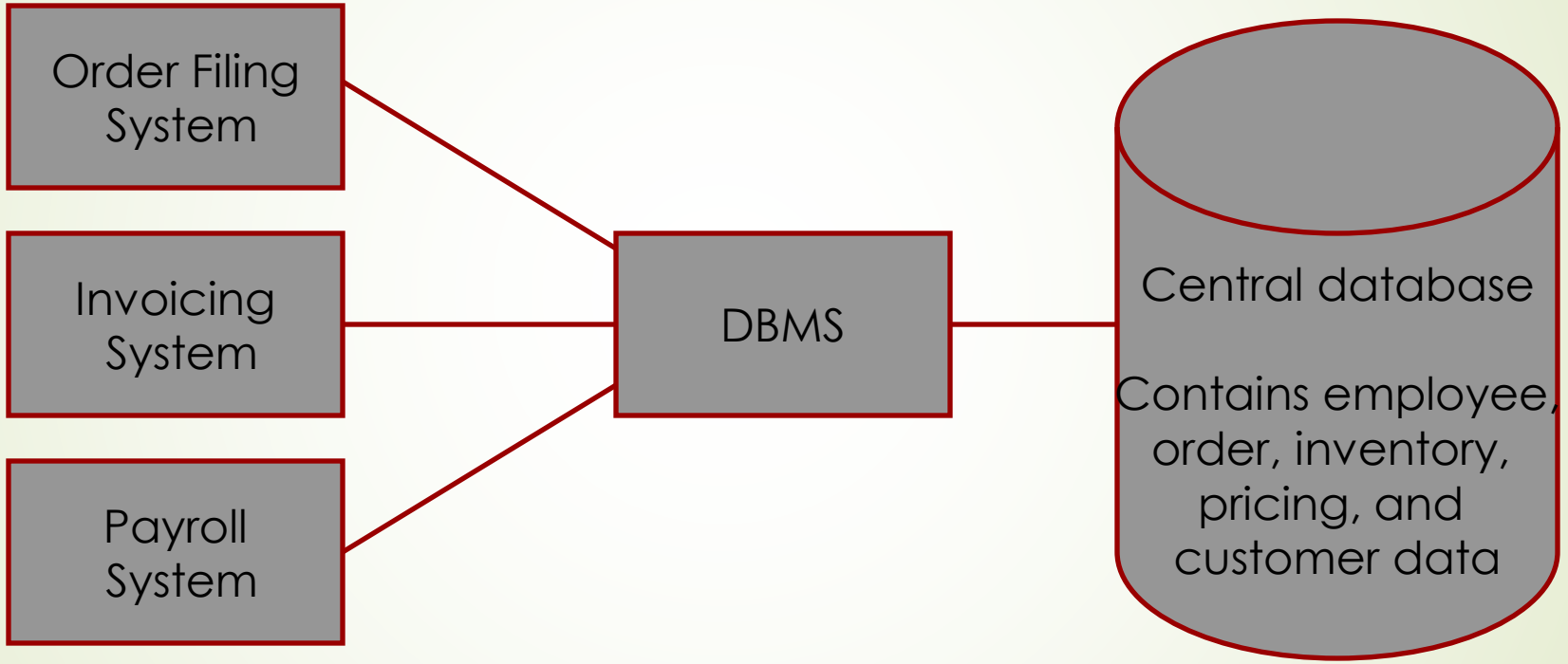
SOLUTION:
The DATABASE Approach

- Central repository of shared data
- Data is managed by a controlling agent
- Stored in a standardized, convenient form

Requires a Database Management System (DBMS)

Database Management System

- A software system that is used to create, maintain, and provide controlled access to user databases



DBMS manages data resources like an operating system manages hardware resources

Advantages of the Database Approach

- Program-data independence
- Planned data redundancy
- Improved data consistency
- Improved data sharing
- Increased application development productivity
- Enforcement of standards
- Improved data quality
- Improved data accessibility and responsiveness
- Reduced program maintenance
- Improved decision support

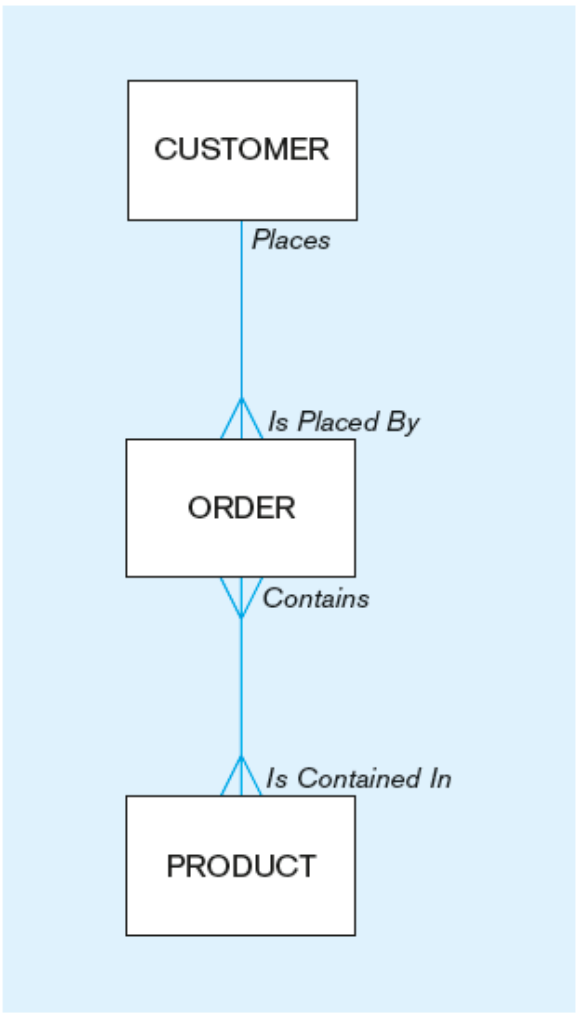
Elements of the Database Approach

- **Data models :**Graphical system capturing nature and relationship of data
  - Enterprise Data Model–high-level entities and relationships for the organization
  - Project Data Model–more detailed view, matching data structure in database or data warehouse.
- **Entities**
  - Noun form describing a person, place, object, event, or concept
  - Composed of attributes

Elements of the Database Approach

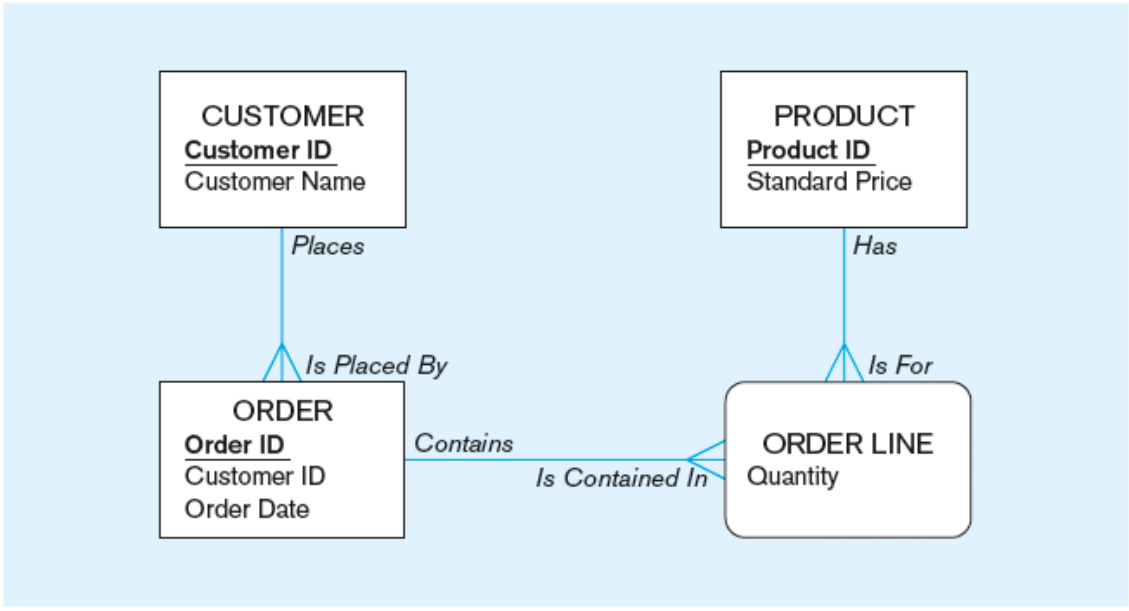
- **Relationships**
  - Between entities
  - Usually one-to-many (1:M) or many-to-many (M:N)
- **Relational Databases**
  - Database technology involving tables (relations) representing entities and primary/foreign keys representing relationships.

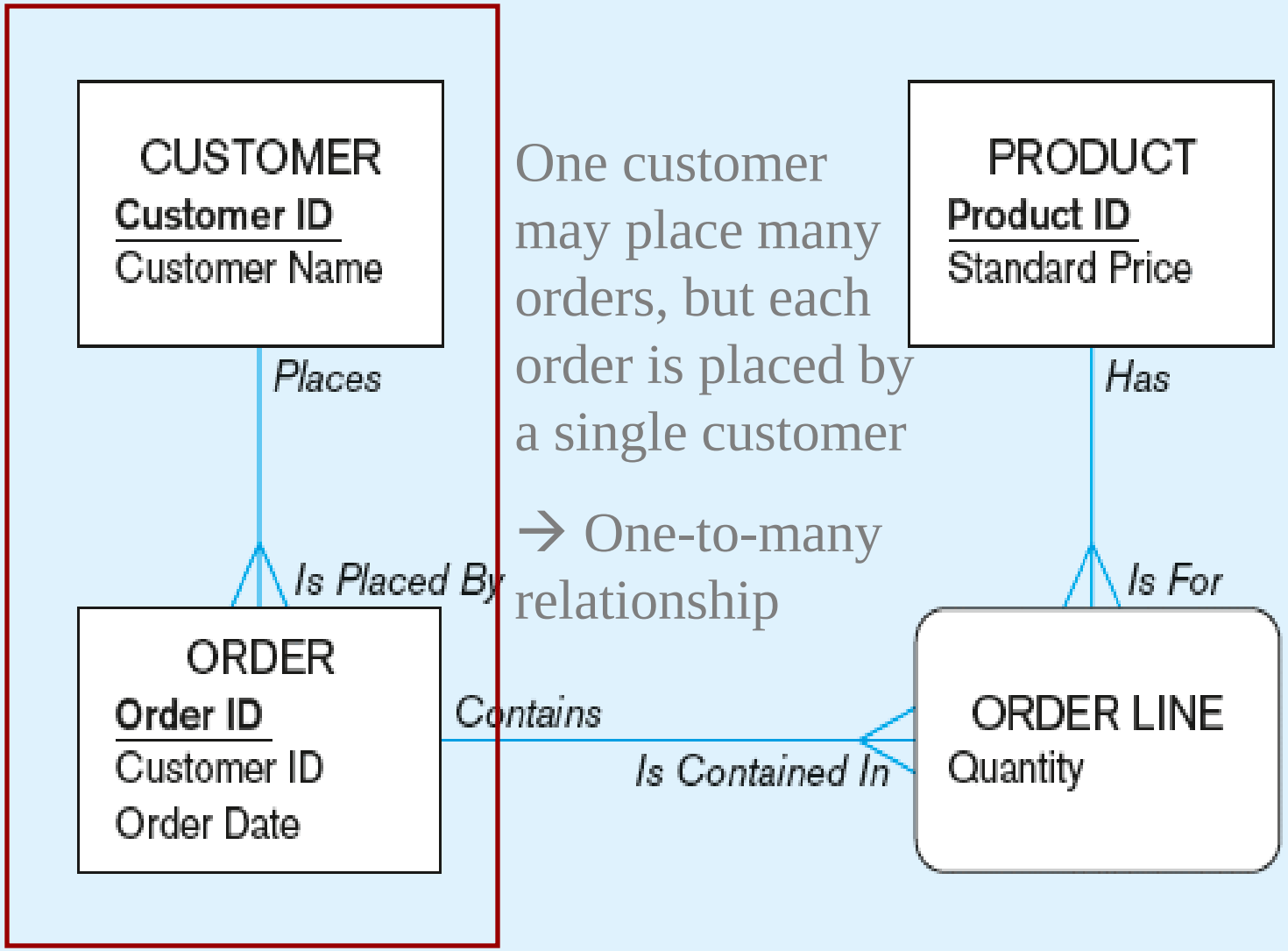
Figure 1-3 Comparison of enterprise and project level data models



Segment of an enterprise data model

Segment of a project-level data model



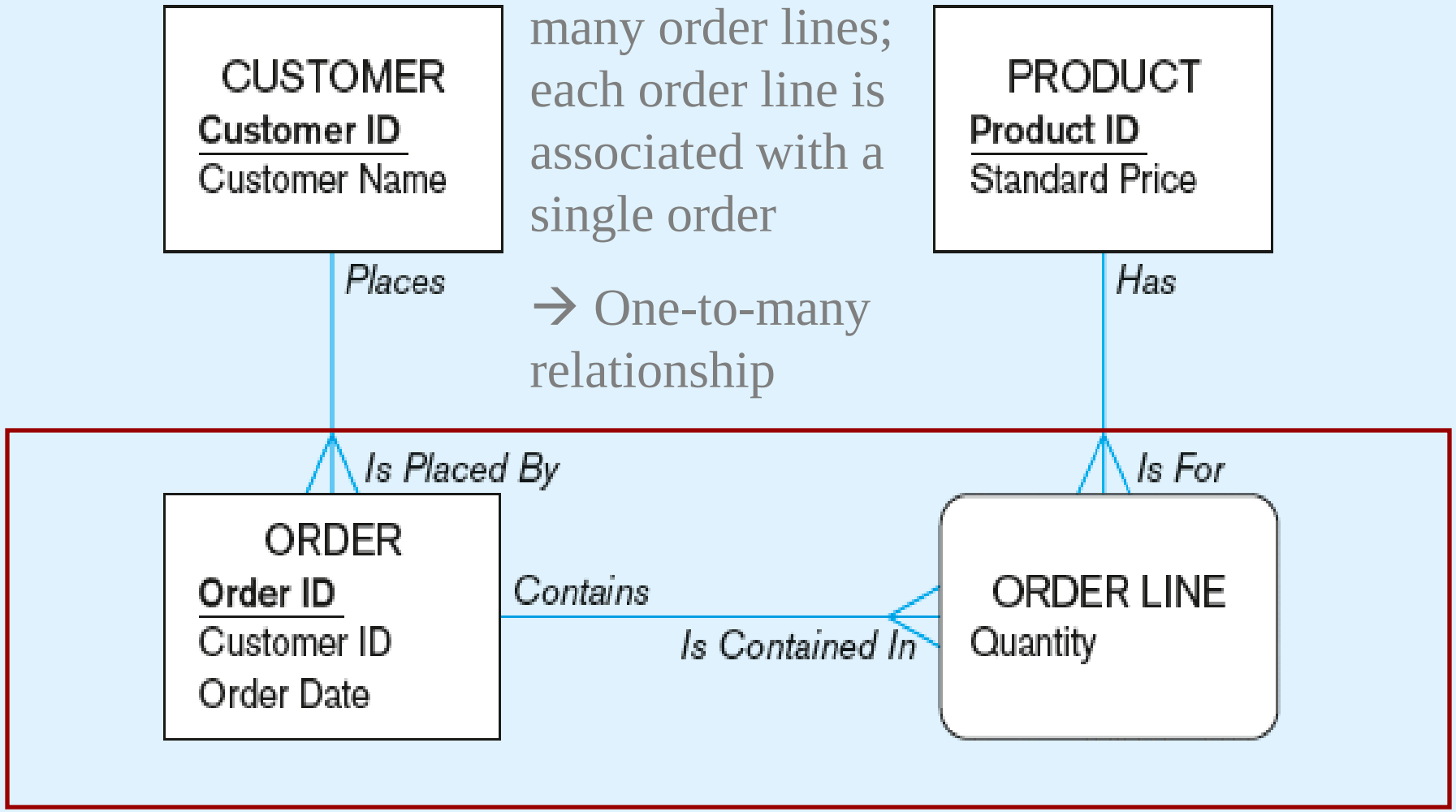


One customer may place many orders, but each order is placed by a single customer

→ One-to-many relationship

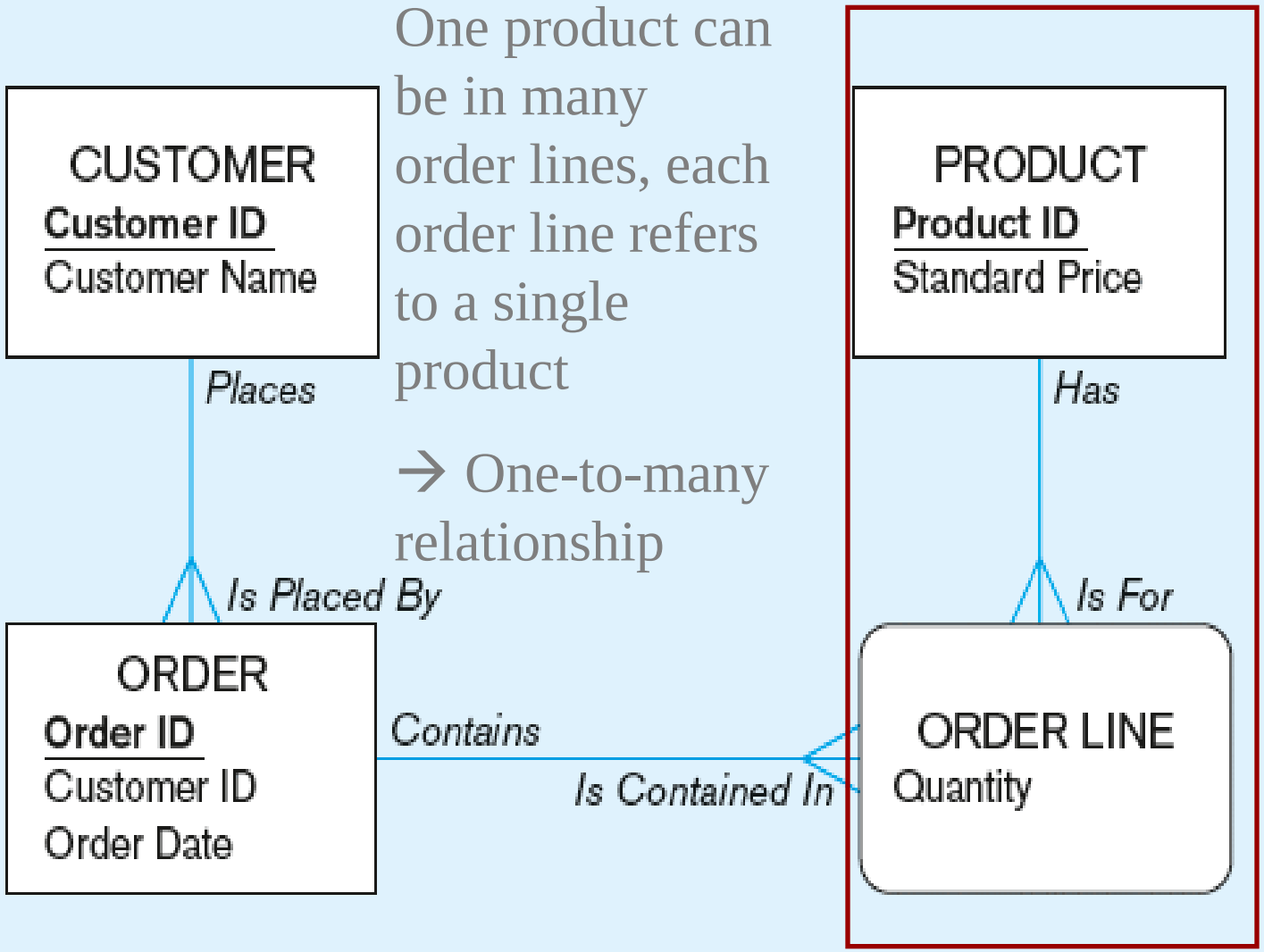
One order has many order lines; each order line is associated with a single order

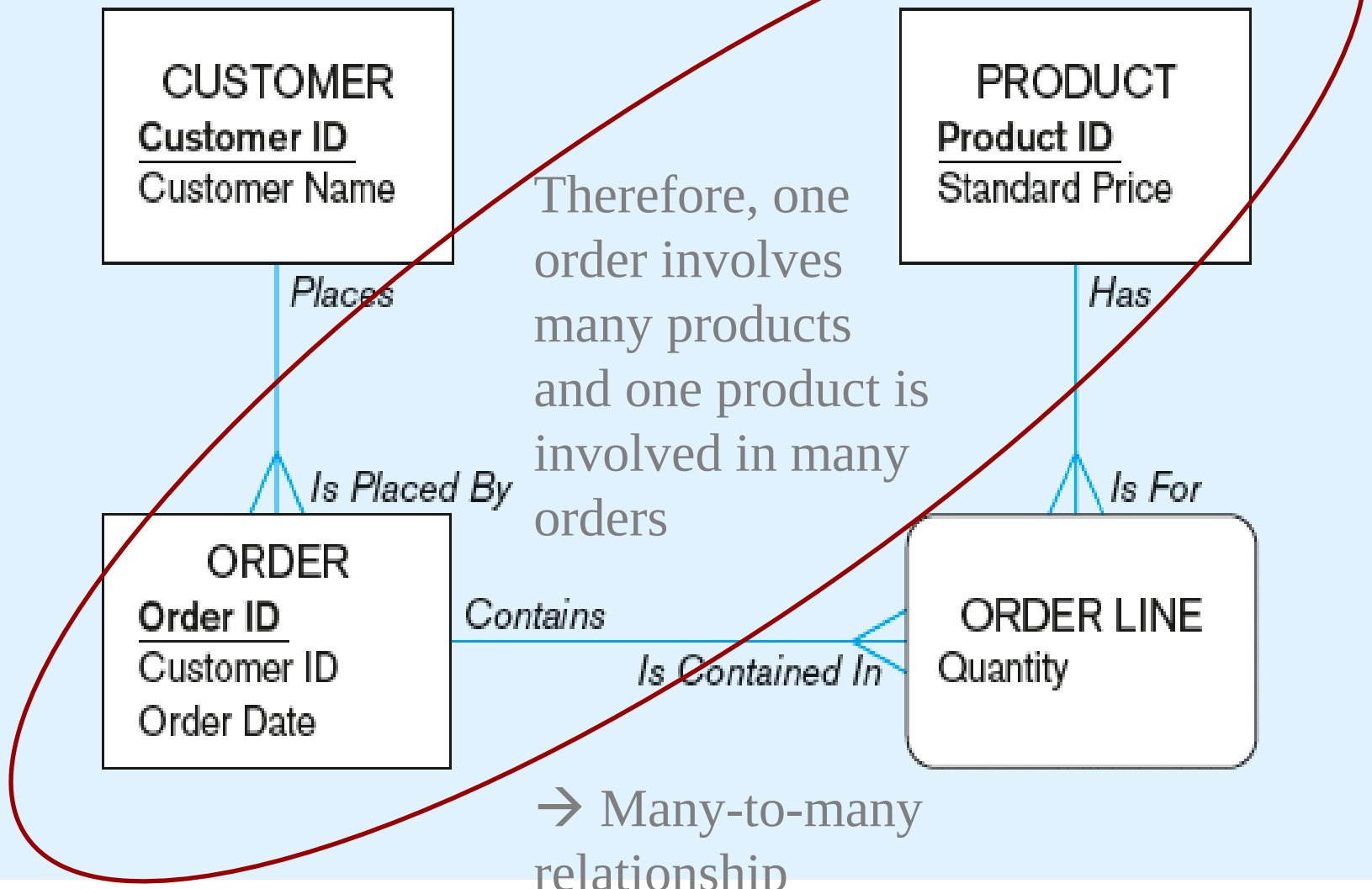
→ One-to-many relationship



One product can be in many order lines, each order line refers to a single product

→ One-to-many relationship





Therefore, one order involves many products and one product is involved in many orders

→ Many-to-many relationship

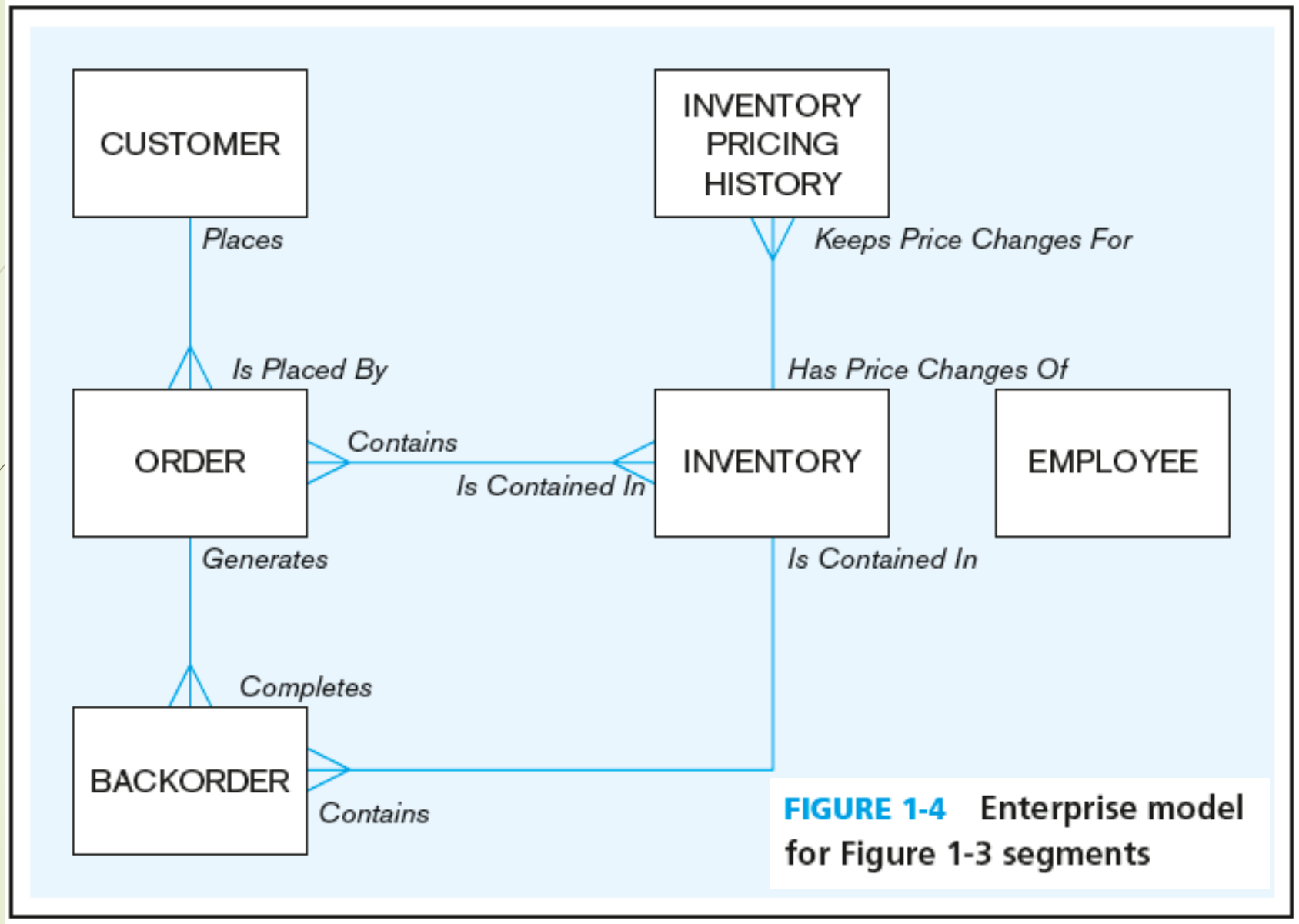
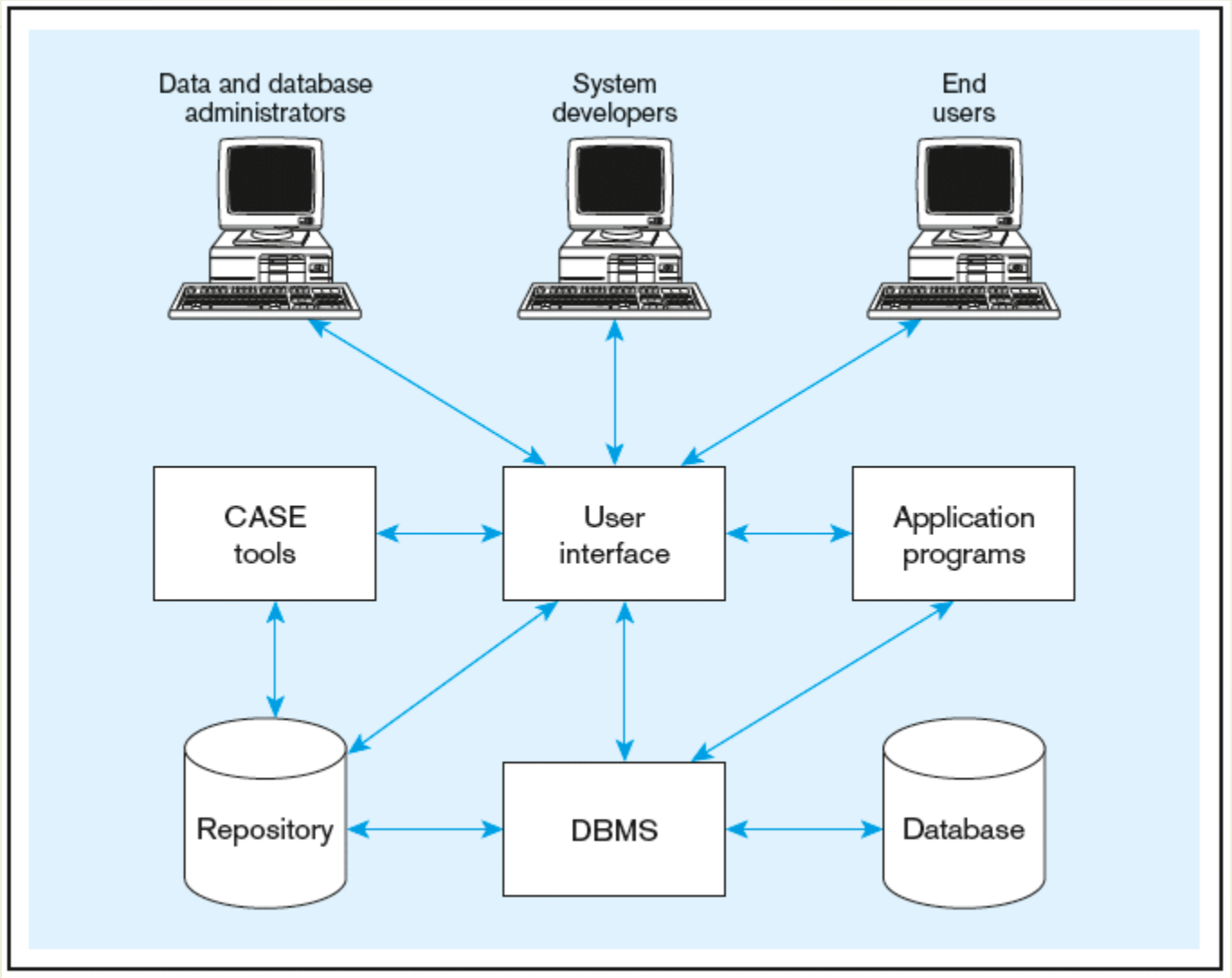


FIGURE 1-4 Enterprise model for Figure 1-3 segments

Figure 1-5 Components of the Database Environment



Components of the Database Environment

- **CASE Tools**–computer-aided software engineering
- **Repository**–centralized storehouse of metadata
- **Database Management System (DBMS)** –software for managing the database
- **Database**–storehouse of the data
- **Application Programs**–software using the data
- **User Interface**–text and graphical displays to users
- **Data/Database Administrators**–personnel responsible for maintaining the database
- **System Developers**–personnel responsible for designing databases and software
- **End Users**–people who use the applications and databases

The Range of Database Applications

Type of Database / Application	Typical Number of Users	Typical Size of Database
Personal	1	Megabytes
Two-tier	5–100	Megabytes–gigabytes
Three-tier	100–1000	Gigabytes
Enterprise resource planning	>100	Gigabytes–terabytes
Data warehousing	>100	Terabytes–petabytes

The Range of Database Applications

- **Personal Databases**

- **Description**:
  These are small-scale databases designed for individual use. They are typically managed on a personal computer or mobile device and cater to single-user applications.
- **Typical Number of Users**:
  Only one user at a time.
- **Typical Size of Database**:
  The size is usually small, ranging in **megabytes**, as the data being stored is personal or specific to a single user.
- **Examples**:
  - Microsoft Access databases for personal projects.
  - Simple SQLite databases used in mobile apps.
  - Personal expense tracking or address book applications.

The Range of Database Applications

2. Two-Tier Databases

- **Description**:
  Two-tier databases involve a client-server architecture, where the client interacts directly with the database server. This setup is common in small businesses or localized applications.
- **Typical Number of Users**:
  Supports **5 to 100 users**, making it suitable for small teams or departments.
- **Typical Size of Database**:
  Ranges from **megabytes to gigabytes**, depending on the complexity and volume of data.
- **Examples**:
  - A small business managing inventory using a MySQL database.
  - Localized applications like customer support tools or departmental software.

The Range of Database Applications

3. Three-Tier Databases

- **Description**:
  These databases use a more complex architecture with three layers:
  - **Client layer** (user interface).
  - **Application layer** (business logic and processing).
  - **Database layer** (data storage and retrieval).
    This structure supports larger and more scalable applications.
- **Typical Number of Users**:
  Serves **100 to 1,000 users** simultaneously, often in medium to large organizations.
- **Typical Size of Database**:
  Data is stored in **gigabytes**, suitable for managing more complex and extensive datasets.
- **Examples**:
  - E-commerce platforms like Amazon.
  - Online banking systems.
  - Content management systems for websites (e.g., WordPress with large-scale data).

The Range of Database Applications

4. Enterprise Resource Planning (ERP) Systems

- **Description**:
  ERP databases integrate all core business functions (e.g., finance, HR, supply chain) into a unified system, allowing large organizations to manage their operations efficiently.
- **Typical Number of Users**:
  Typically used by **more than 100 users** across different departments in an organization.
- **Typical Size of Database**:
  Data stored ranges from **gigabytes to terabytes**, as ERP systems handle massive volumes of transactional and operational data.
- **Examples**:
  - SAP ERP systems used in multinational corporations.
  - Oracle ERP for business processes like payroll and inventory management.

The Range of Database Applications

5. Data Warehousing

- **Description**:
  Data warehouses are specialized databases used for storing and analyzing large volumes of historical and aggregated data. They support decision-making and business intelligence tasks.
- **Typical Number of Users**:
  Serves **more than 100 users**, including analysts and decision-makers.
- **Typical Size of Database**:
  Extremely large, ranging from **terabytes to petabytes**, as they store data collected over long periods.
- **Examples**:
  - Amazon Redshift for large-scale data analytics.
  - Google BigQuery for querying massive datasets.
  - Enterprise data lakes for predictive analytics