Disciplina de Gestão de Dados e de Bases de Dados

Ano lectivo 2013/2014

The Database Environment

Parts of this presentation were taken from the backing material of the book

Modern Database Management, 11/E Edition, 2013

Jeffrey A. Hoffer, V. Ramesh, Heikki Topi

Objectives

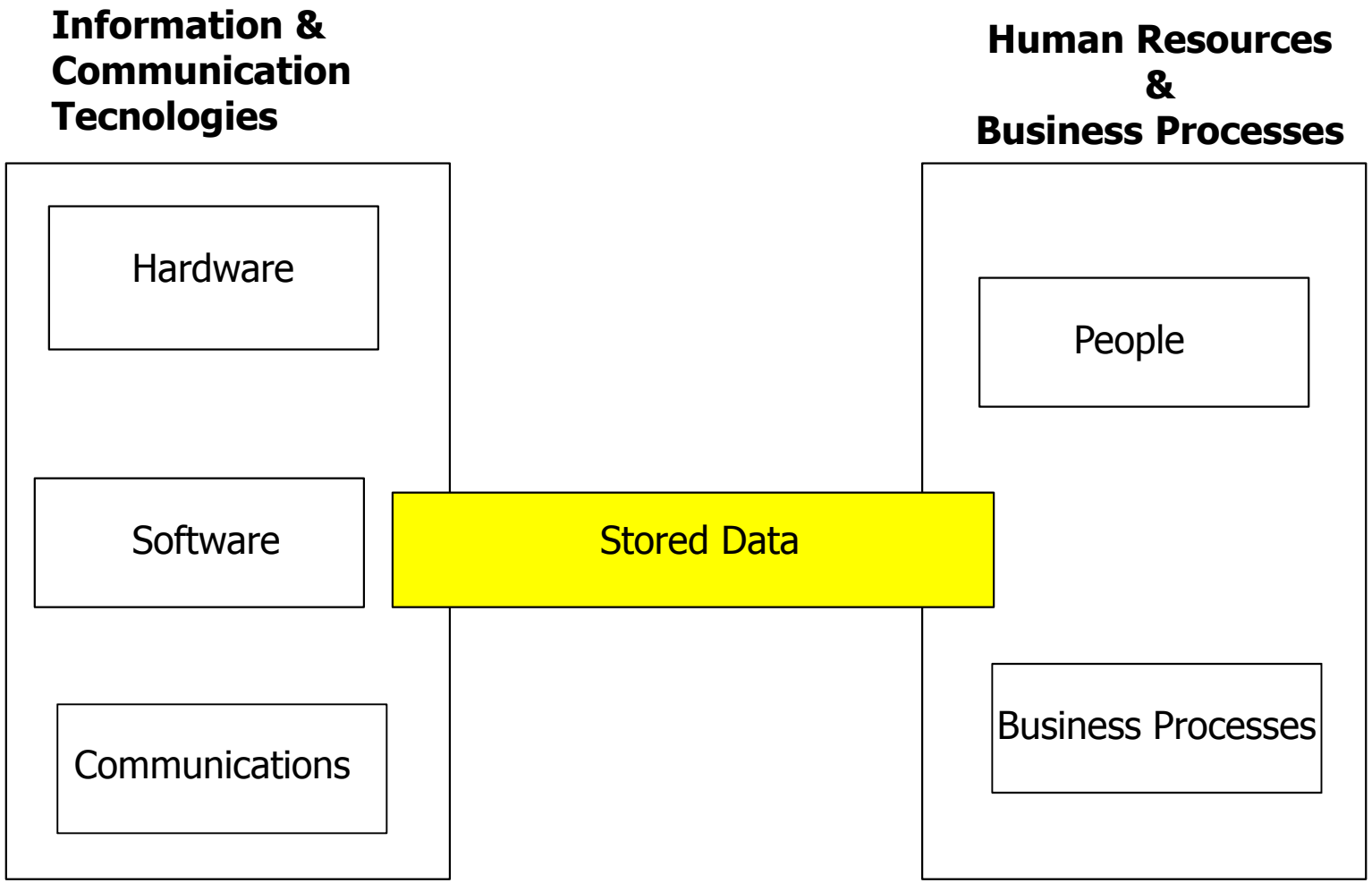
- ✓ Definition of concepts
- ✓ Explain growth and importance of databases
- ✓ Name limitations of conventional file processing
- ✓ Classify databases
- ✓ Explain advantages of databases
- ✓ List components of database environment
- ✓ Describe evolution of database management systems

Information System Definition

An information system (IS) is a socio-technical system, the purpose of which is to process data and provide information to support the operations, management and governance of an organization

Adapted from Robert Nickerson (2009)
http://online.sfsu.edu/~rnick/mannheim/lecturerev.pdf

Information system components model



Adapted from Robert Nickerson (2009)
http://online.sfsu.edu/~rnick/mannheim/lecturerev.pdf

Definitions (1/2)

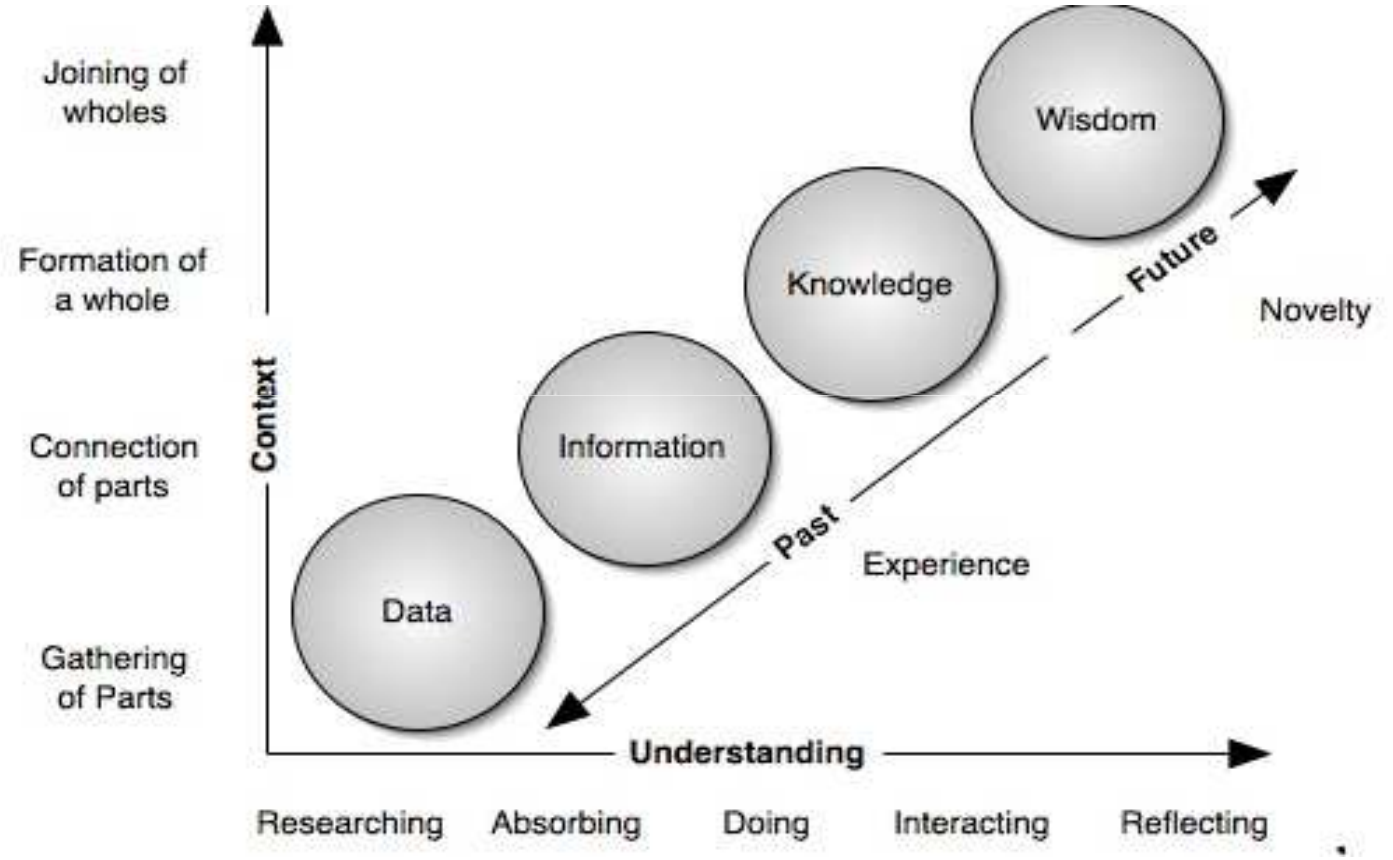
- **Database: organized collection of logically related data**
- **Data: stored representations of objects and events that have meaning and importance in the user's environment**
  - Structured: numbers, text, dates
  - Unstructured: images, video, documents
- **Information: data that have been processed in such a way as to increase the knowledge of the person who uses the data**

Definitions (2/2)

- **Metadata:** data that describes the properties and context of other data
- **Knowledge** is information that changes something or somebody -- either by becoming grounds for actions, or by making an individual (or an institution) capable of different or more effective action

in Drucker (2003)

Data, Information, Knowledge and Wisdom



Clark, D. (n.d.). *Understanding and Performance.* Visto em 18-07-2009, em http://www.skagitwatershed.org/~donclark/performance/understanding.html

Data in context

Class Roster

Course: MGT 500 Business Policy

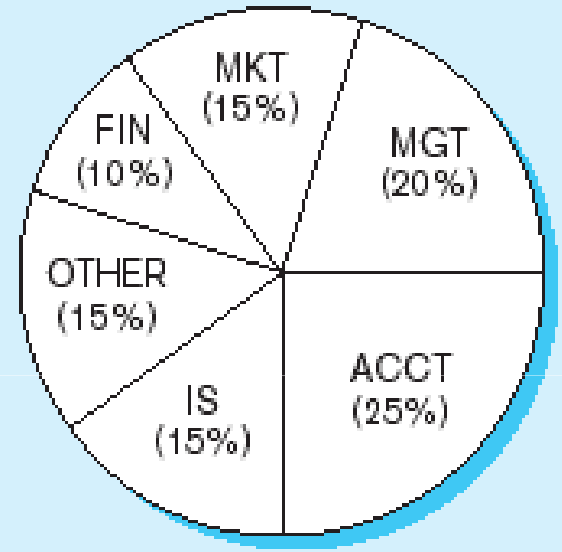
Semester: Spring 200X

Section: 2

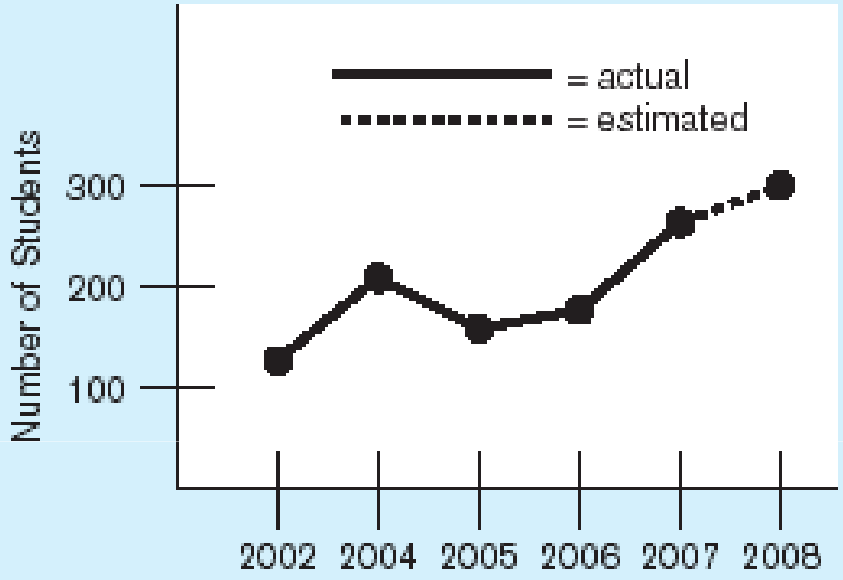
Name	ID	Major	GPA
Baker, Kenneth D.	324917628	MGT	2.9
Doyle, Joan E.	476193248	MKT	3.4
Finkle, Clive R.	548429344	PRM	2.8
Lewis, John C.	551742186	MGT	3.7
McFerran, Debra R.	409723145	IS	2.9
Sisneros, Michael	392416582	ACCT	3.3

Context helps users understand data

Summarized data



Percent Enrollment by Major (200X)



Enrollment Projections

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

Metadata

Table 1-1 Example Metadata for Class Roster

Data Item		*Value*				
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

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

Disadvantages of File Processing

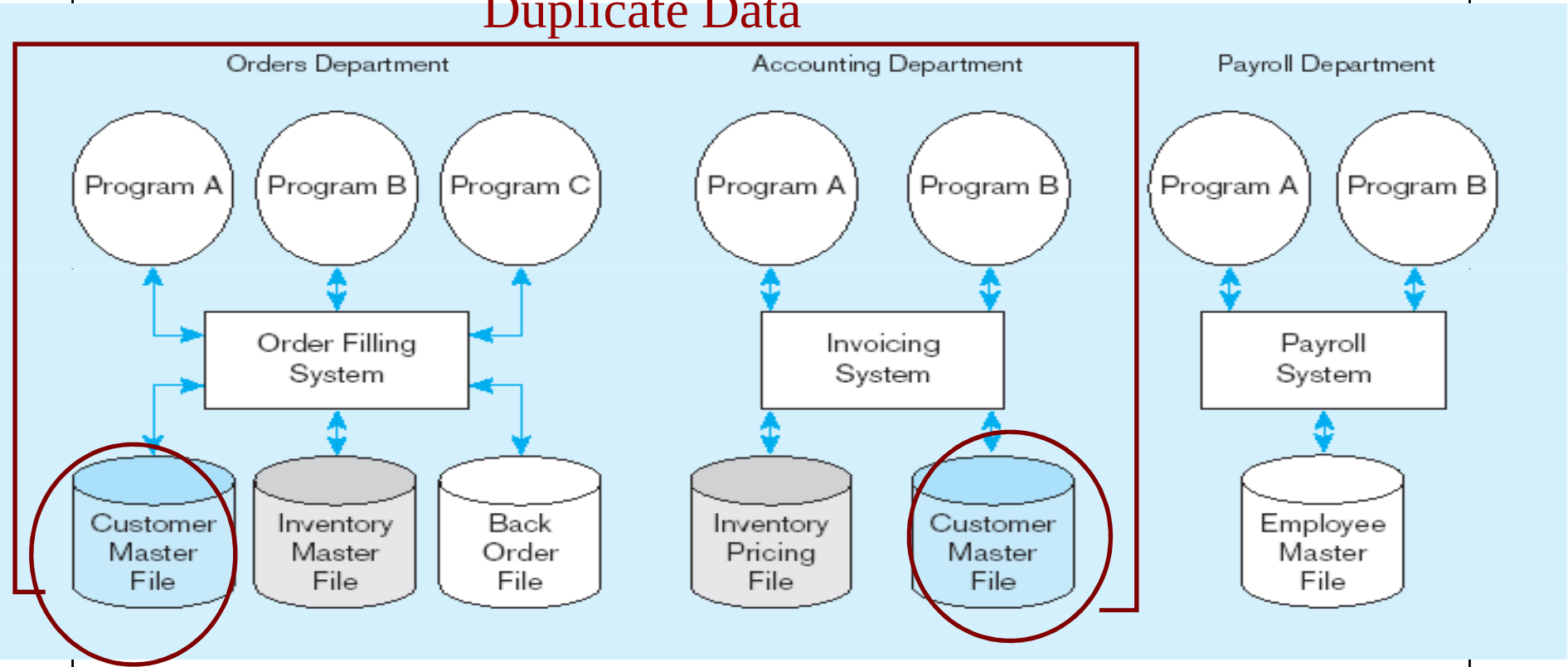
- **Program-Data Dependence**
  - All programs maintain metadata for each file they use
- **Data Redundancy**
  - Different systems/programs have separate copies of the same data
- **Limited Data Sharing**
  - No centralized control of data
- **Lengthy Development Times**
  - Programmers must design 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

File processing systems

Duplicate Data



Problems with Data Redundancy

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

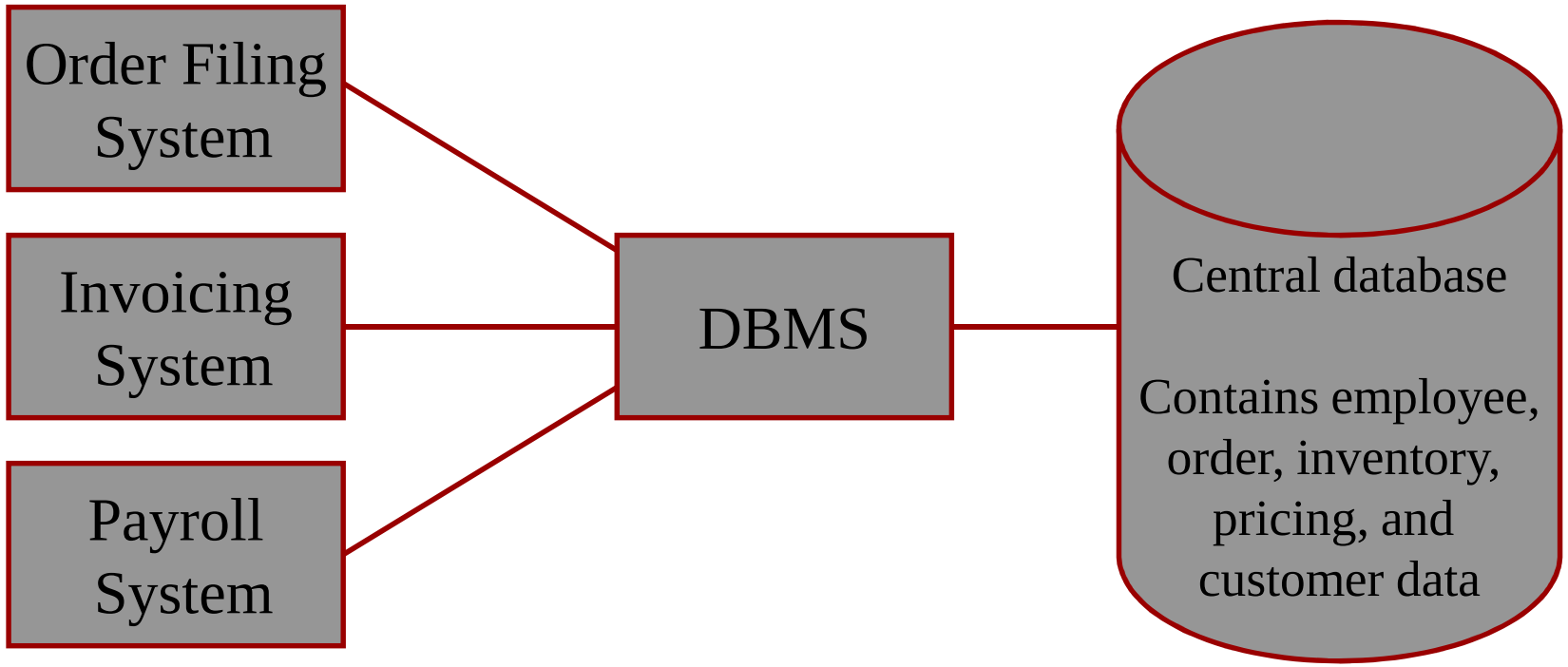
SOLUTION:
The DATABASE Approach

- Central repository of shared data
- Data is managed by a controlling agent (DBMS)
- 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

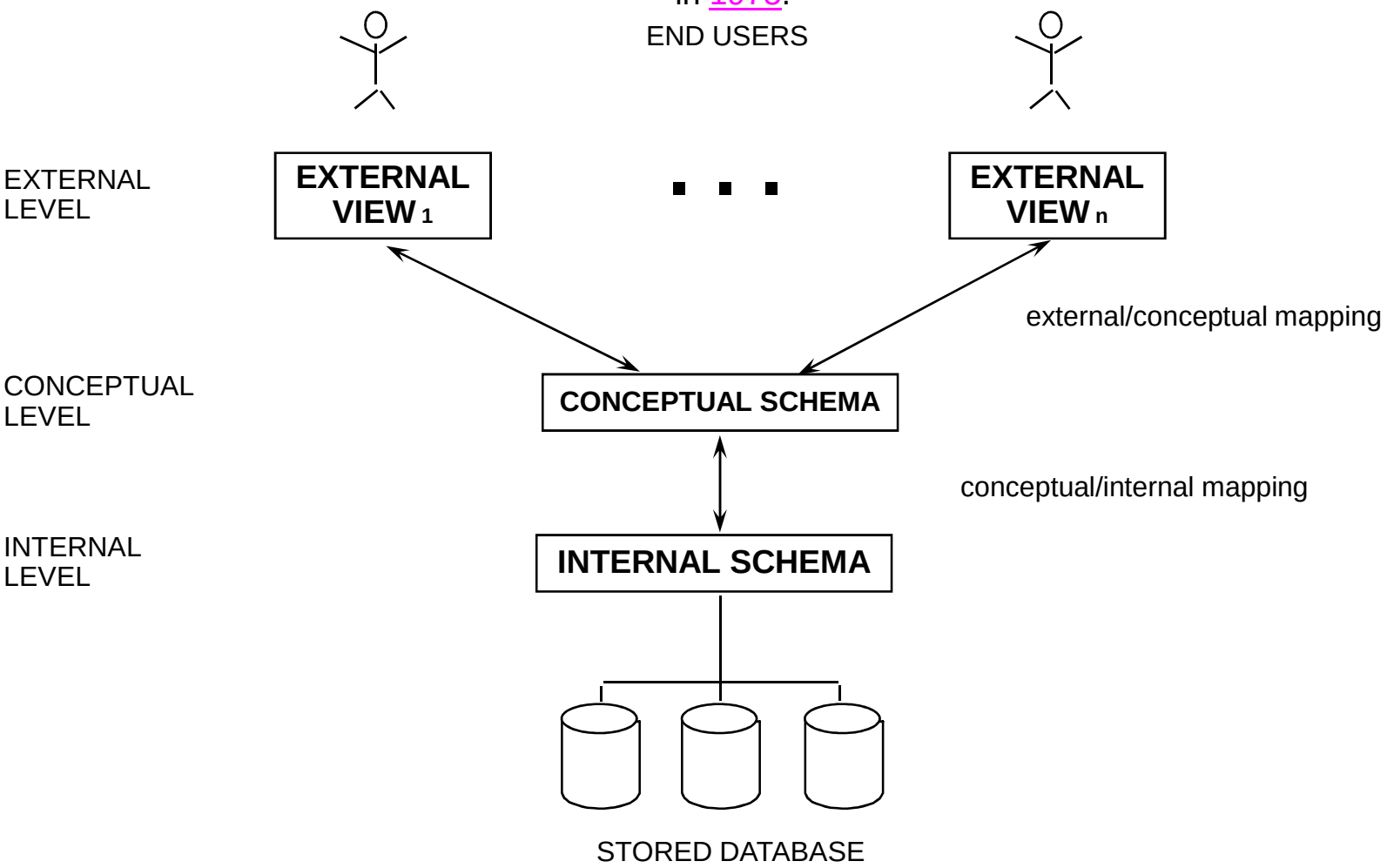
Advantages of the Database Approach

- 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

Program-data independence

ANSI-SPARC Architecture

American National Standards Institute, Standards Planning And Requirements Committee) is an abstract design standard for a Database Management System (DBMS), first proposed in 1975.



Program-Data independence (1/2)

- **Physical data independence**

  Changes to the physical level (how the data is stored, whether in arrays or linked lists etc.) must not require a change to an application based on the structure

- **Logical data independence**

  Changes to the logical level (tables, columns, rows, and so on) must not require a change to an application based on the structure

  Logical data independence is more difficult to achieve than physical data independence

Program-Data independence (2/2)

- **Integrity independence**

  Integrity constraints must be specified separately from application programs and stored in the catalog. It must be possible to change such constraints as and when appropriate without unnecessarily affecting existing applications

- **Distribution independence**

  The distribution of portions of the database to various locations should be invisible to users of the database. Existing applications should continue to operate successfully :

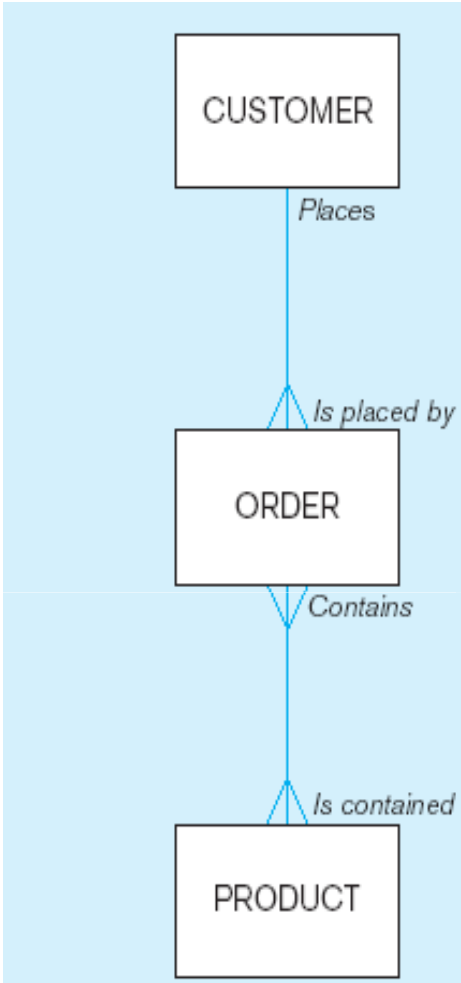
  1. When a distributed version of the DBMS is first introduced; and

  2. When existing distributed data are redistributed around the system

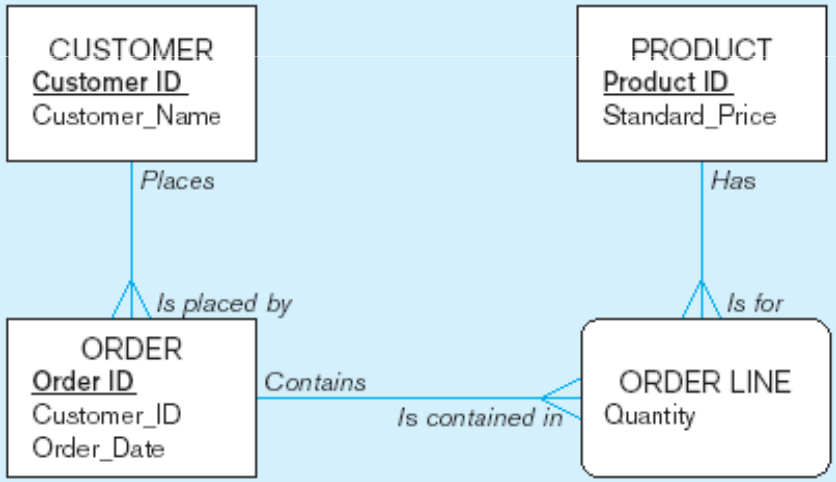
Elements of the Database Approach

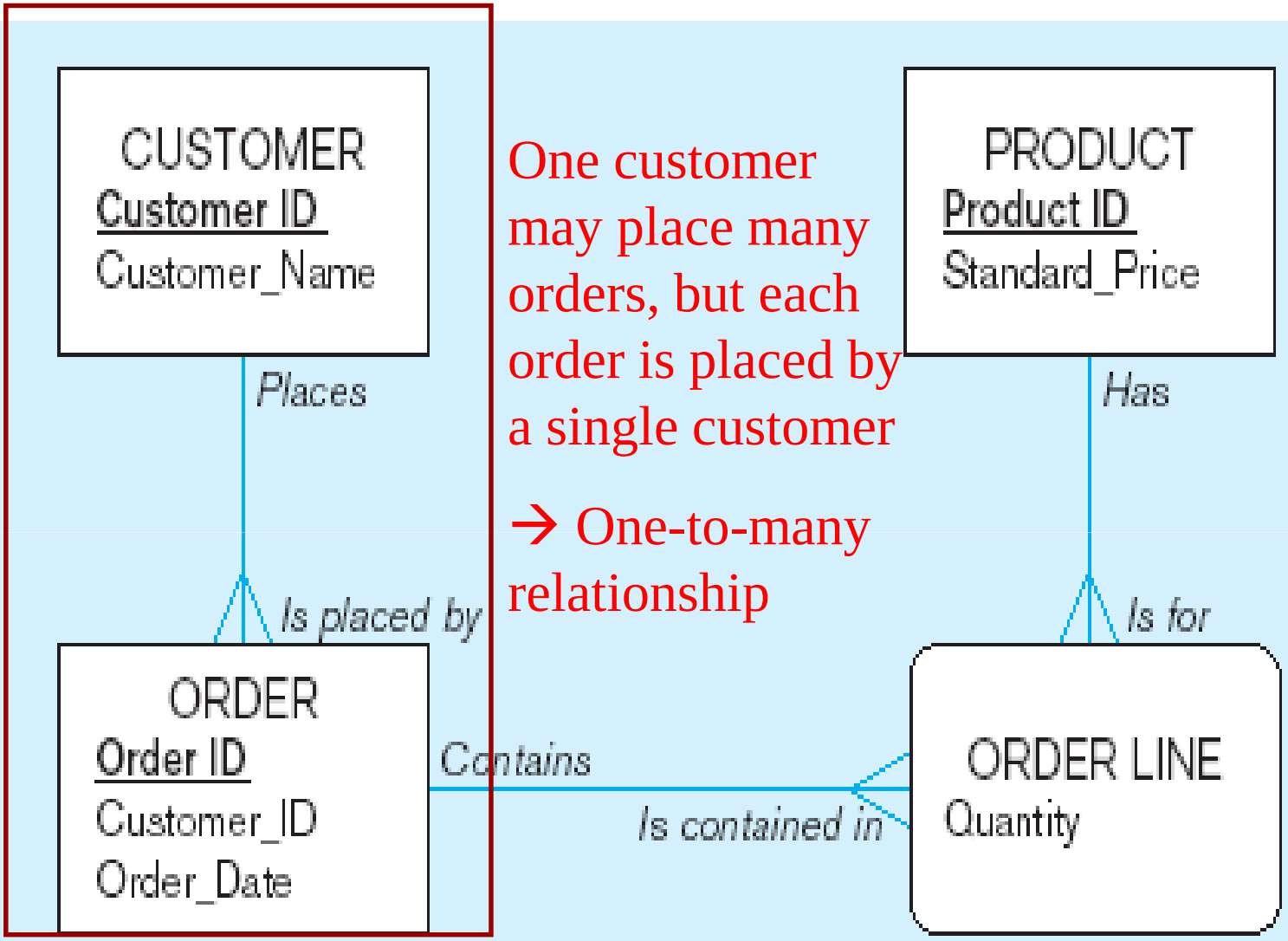
- **Data models**
  - Graphical system capturing nature and relationship of data
  - **Enterprise Data Model**–high-level entities and relationships for the organization
  - **Project Data Model**–more detailed view, matching data structure in database or data warehouse
- **Relational Databases**
  - Database technology involving tables (relations) representing entities and primary/foreign keys representing relationships
- **Use of Internet Technology**
  - Networks and telecommunications, distributed databases, client-server, and 3-tier architectures
- **Database Applications**
  - Application programs used to perform database activities (create, read, update, and delete) for database users

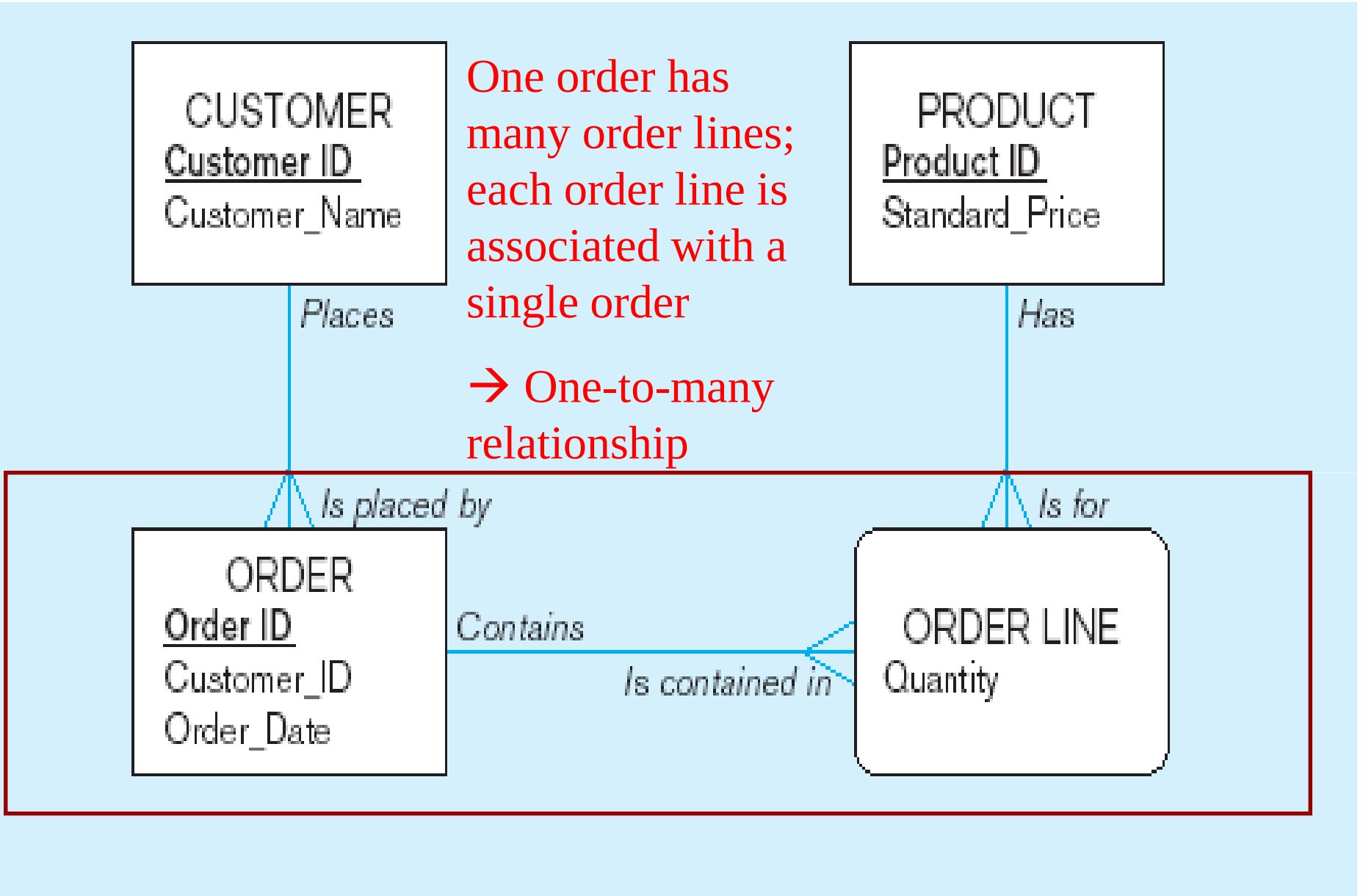
Segment of an Enterprise Data Model

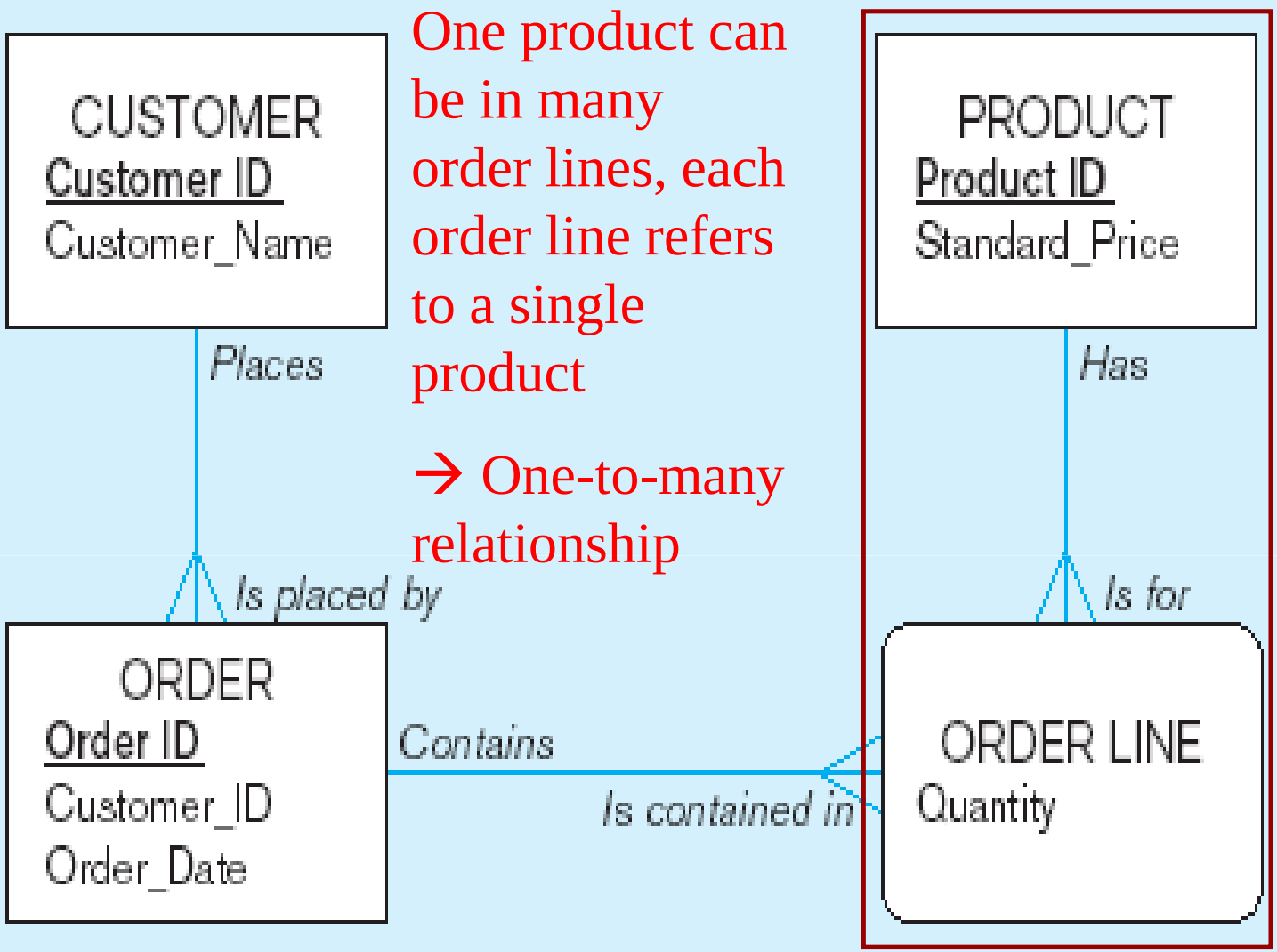


Segment of a Project-Level Data Model



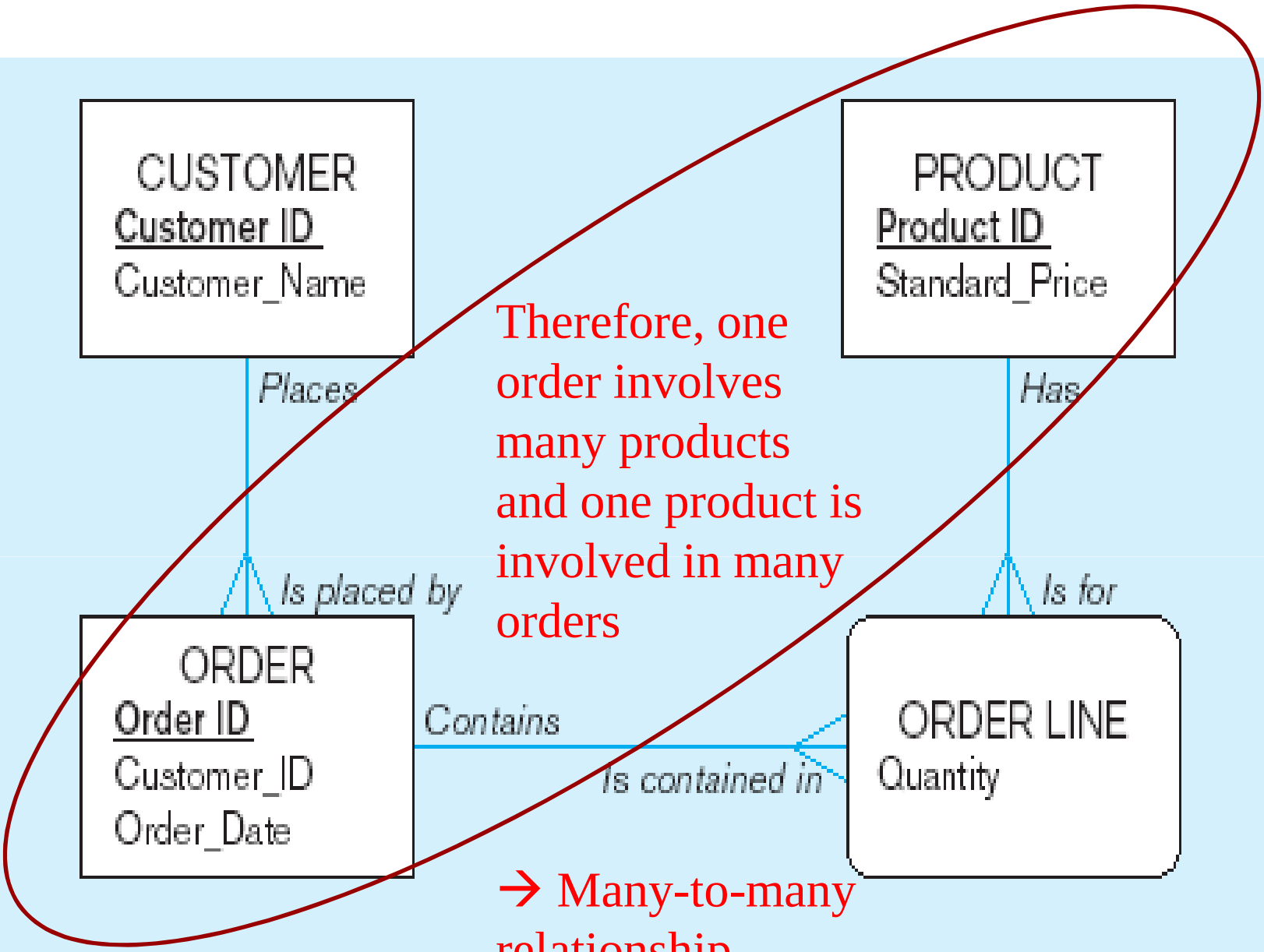

CUSTOMER
Customer ID
Customer_Name
Places
Is placed by
ORDER
Order ID
Customer_ID
Order_Date
Contains
Is contained in
ORDER LINE
Quantity
PRODUCT
Product ID
Standard_Price
Has
Is for
One customer may place many orders, but each order is placed by a single customer
→ One-to-many relationship


CUSTOMER
Customer ID
Customer_Name
Places
One order has many order lines; each order line is associated with a single order
→ One-to-many relationship
PRODUCT
Product ID
Standard_Price
Has
Is placed by
Is for
ORDER
Order ID
Customer_ID
Order_Date
Contains
Is contained in
ORDER LINE
Quantity

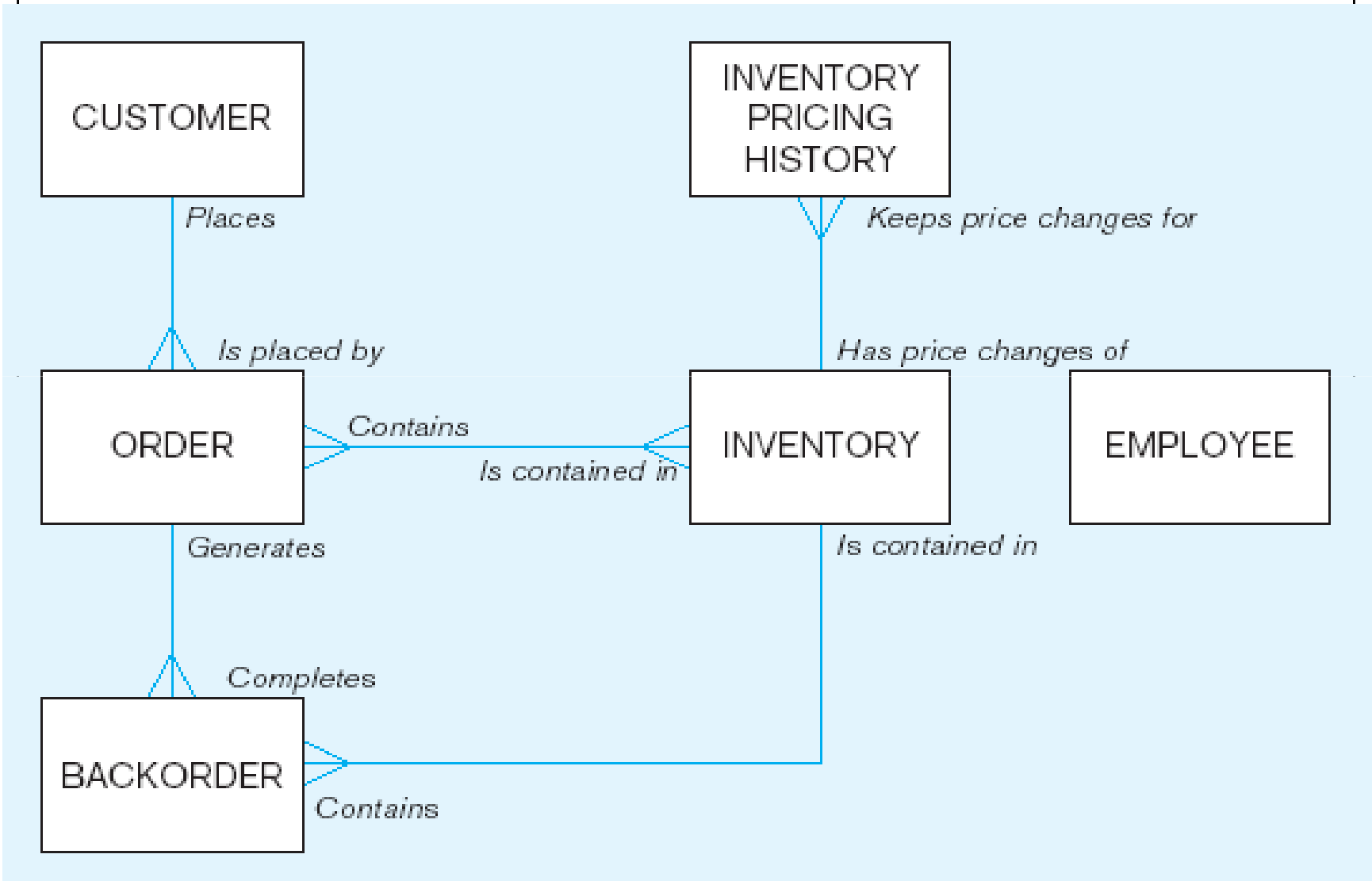


One product can be in many order lines, each order line refers to a single product

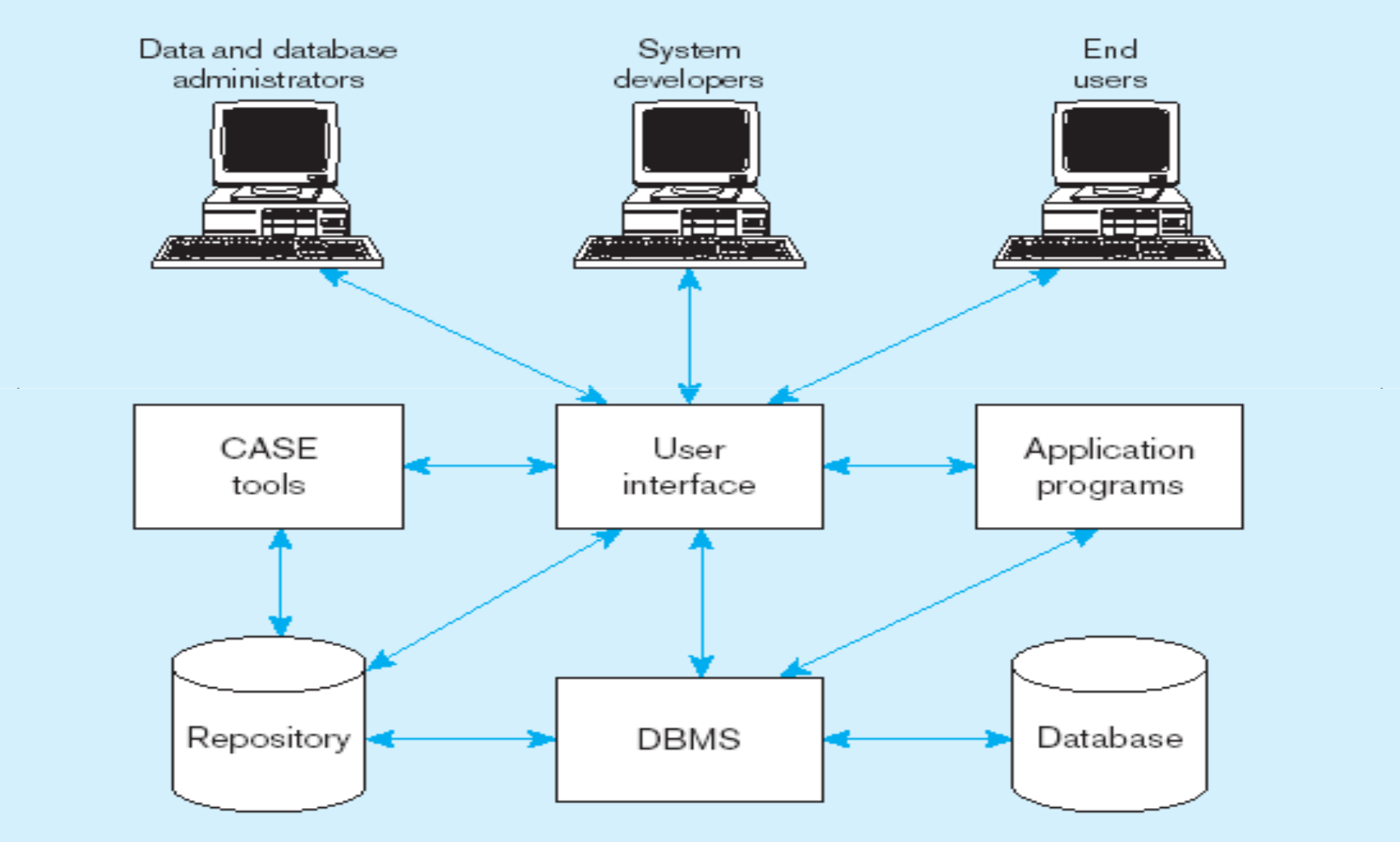
→ One-to-many relationship


CUSTOMER
Customer ID
Customer_Name
PRODUCT
Product ID
Standard_Price
Places
Has
Is placed by
Is for
ORDER
Order ID
Customer_ID
Order_Date
Contains
Is contained in
ORDER LINE
Quantity
Therefore, one order involves many products and one product is involved in many orders
→ Many-to-many relationship

Enterprise data model



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
- **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

Commercial and Open Source DBMS

- Oracle
- DB2
- SQL Server
- Informix
- Sybase
- Ingres
- MySQL
- Postgres
- Access
- …..

Commercial Repositories

- **CA Repository for Distributed Systems r2.3 – Computer Associates**
- **Rochade - ASG**
- **Metacenter 3.8 - Data Advantage Group**
- **MetaMatrix Data Services Platform – Red Hat**
- **The InfoLibrarian Framework – Info Librarian Corporation**
- **…..**

Enterprise Data Model

- First step in the database development process
- Specifies scope and general content
- Overall picture of organizational data at high level of abstraction
- Entity-relationship diagram
- Descriptions of entity types
- Relationships between entities
- Business rules

Example business function-to-data entity matrix

Business Functions \ Data Entity Types	Customer	Product	Raw Material	Order	Work Center	Work Order	Invoice	Equipment	Employee
Business Planning	X	X						X	X
Product Development		X	X		X			X	
Materials Management		X	X	X	X	X		X	
Order Fulfillment	X	X	X	X	X	X	X	X	X
Order Shipment	X	X		X	X		X		X
Sales Summarization	X	X		X			X		X
Production Operations		X	X	X	X	X		X	X
Finance and Accounting	X	X	X	X	X		X	X	X

X = data entity is used within business function

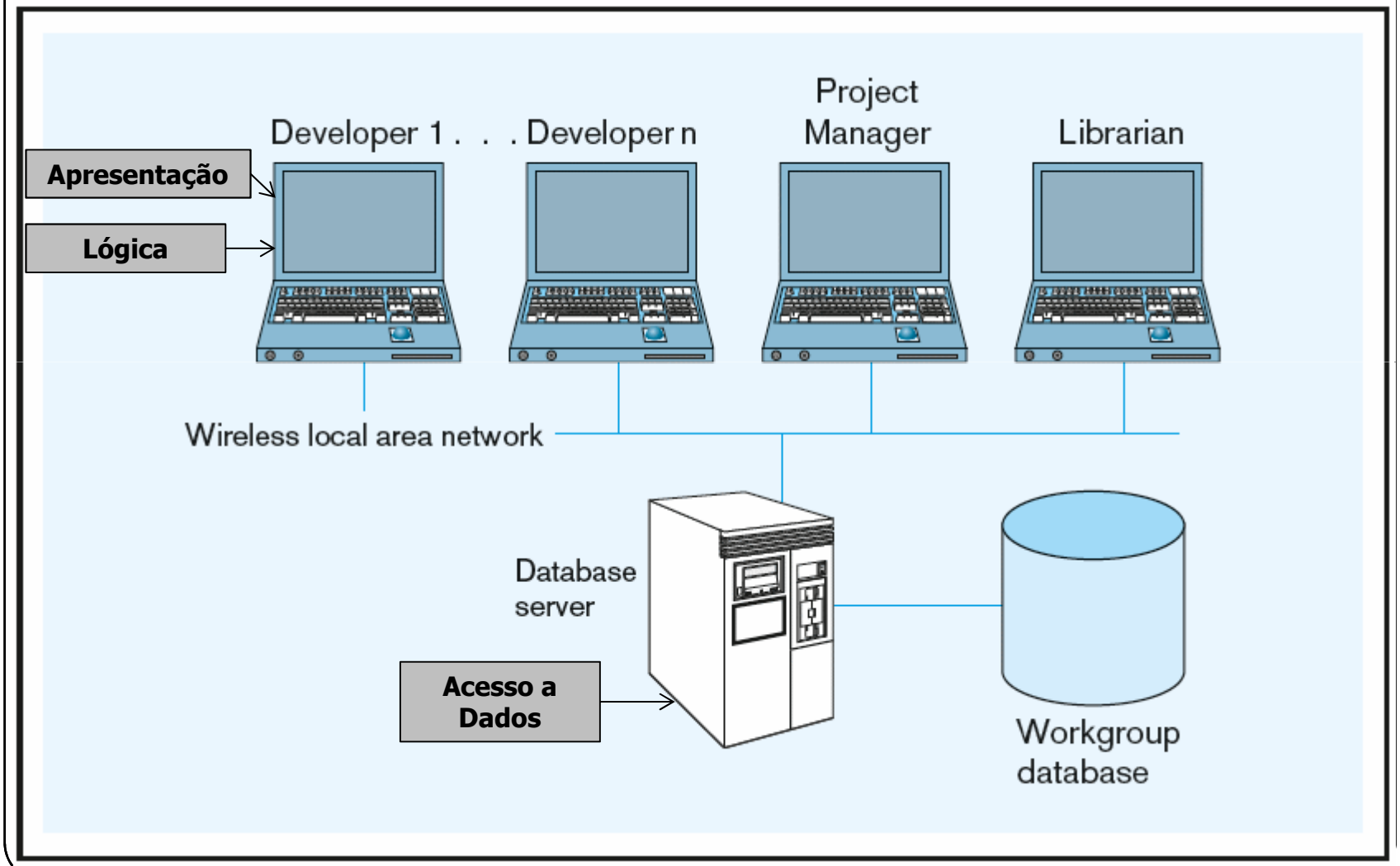
The Range of Database Applications

- **Personal databases**
- **Two-tier Client/Server databases**
- **Multitier Client/Server databases**
- **Enterprise applications**
  - Enterprise resource planning (ERP) systems
  - Data warehousing implementations

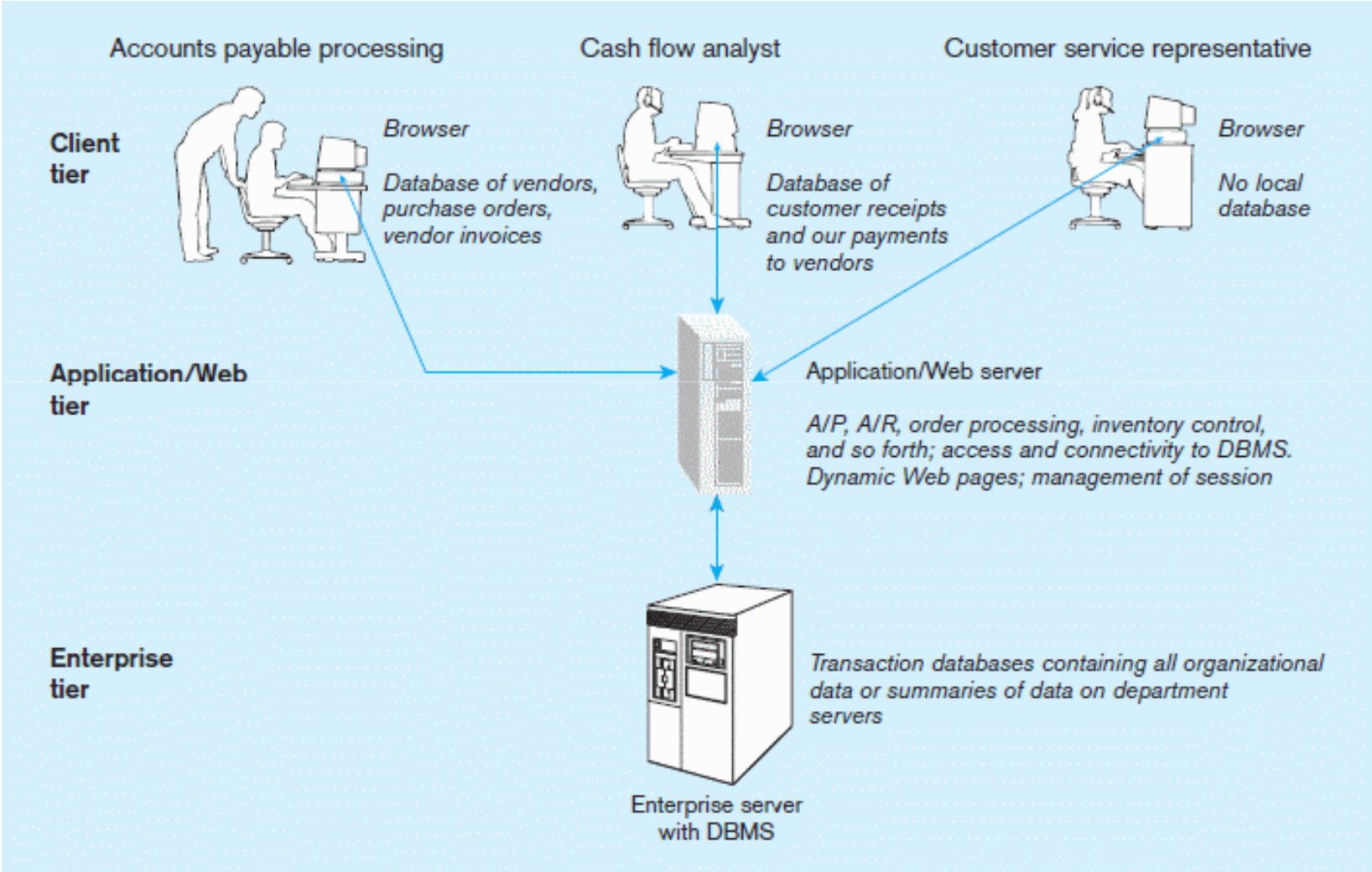
TABLE 1-5 Summary of Database Applications

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

Two-tier database with local area network



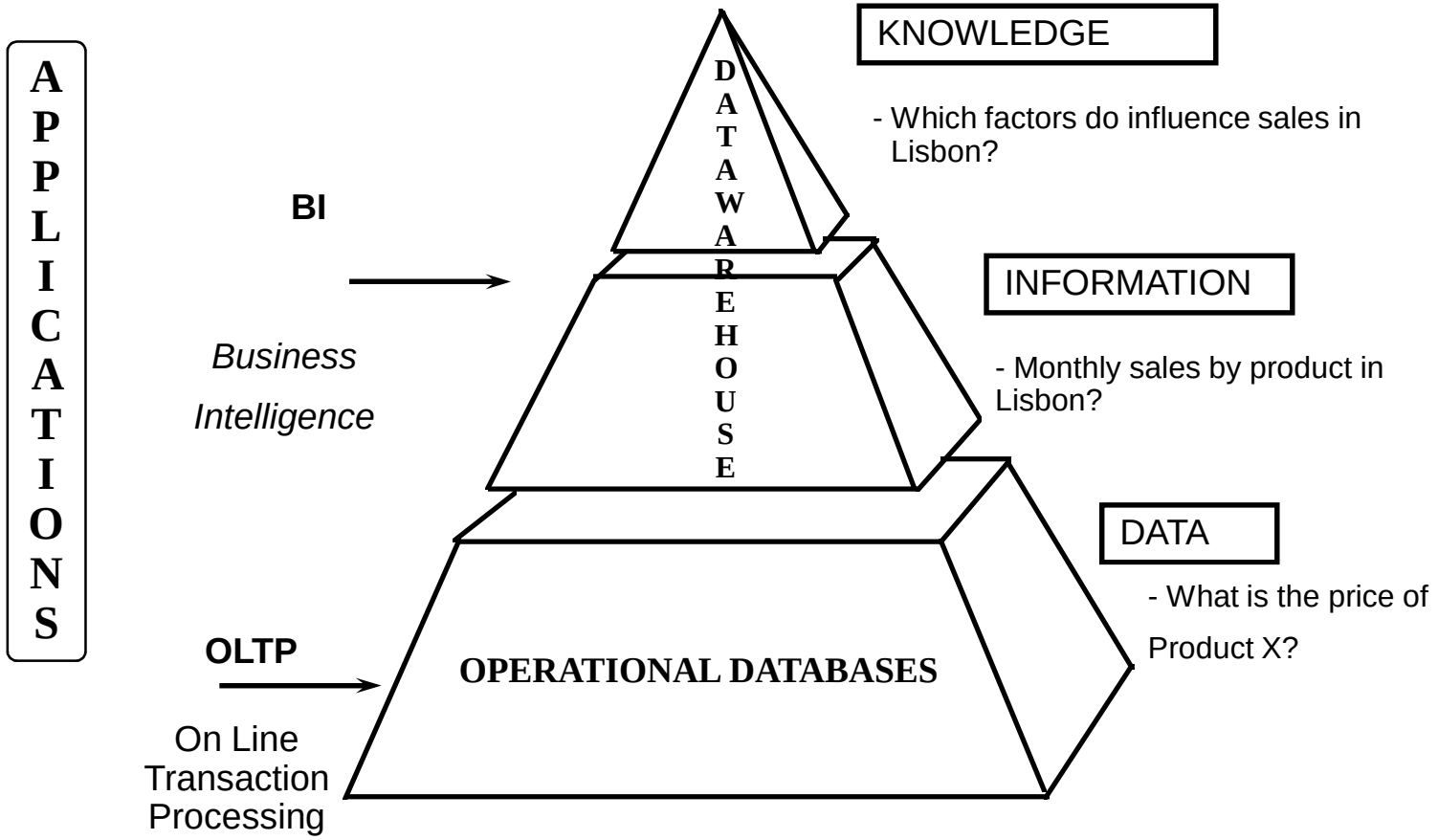
Three-tiered client/server database architecture



Enterprise Database Applications

- **Enterprise Resource Planning (ERP)**
  - Integrate all enterprise functions (manufacturing, finance, sales, marketing, inventory, accounting, human resources)
- **Data Warehouse**
  - Integrated decision support system derived from various operational databases

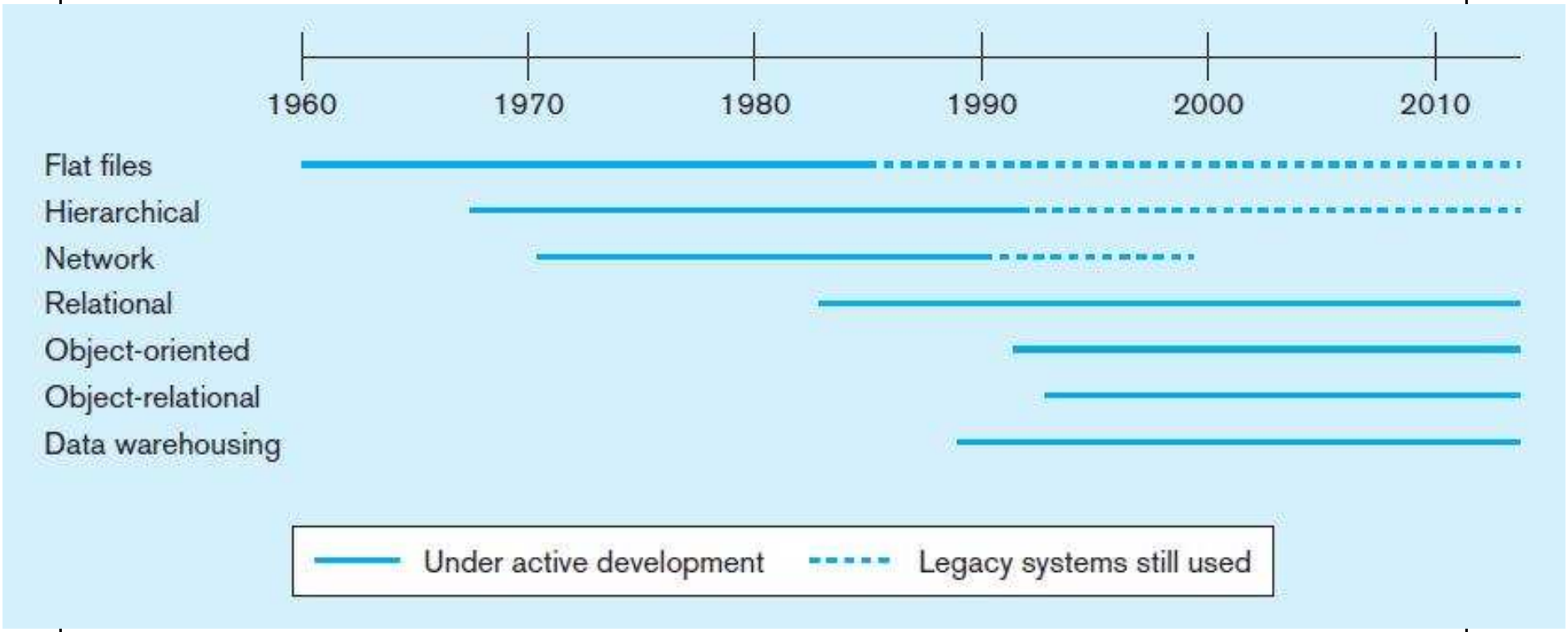
Classification of Databases According to the type of use



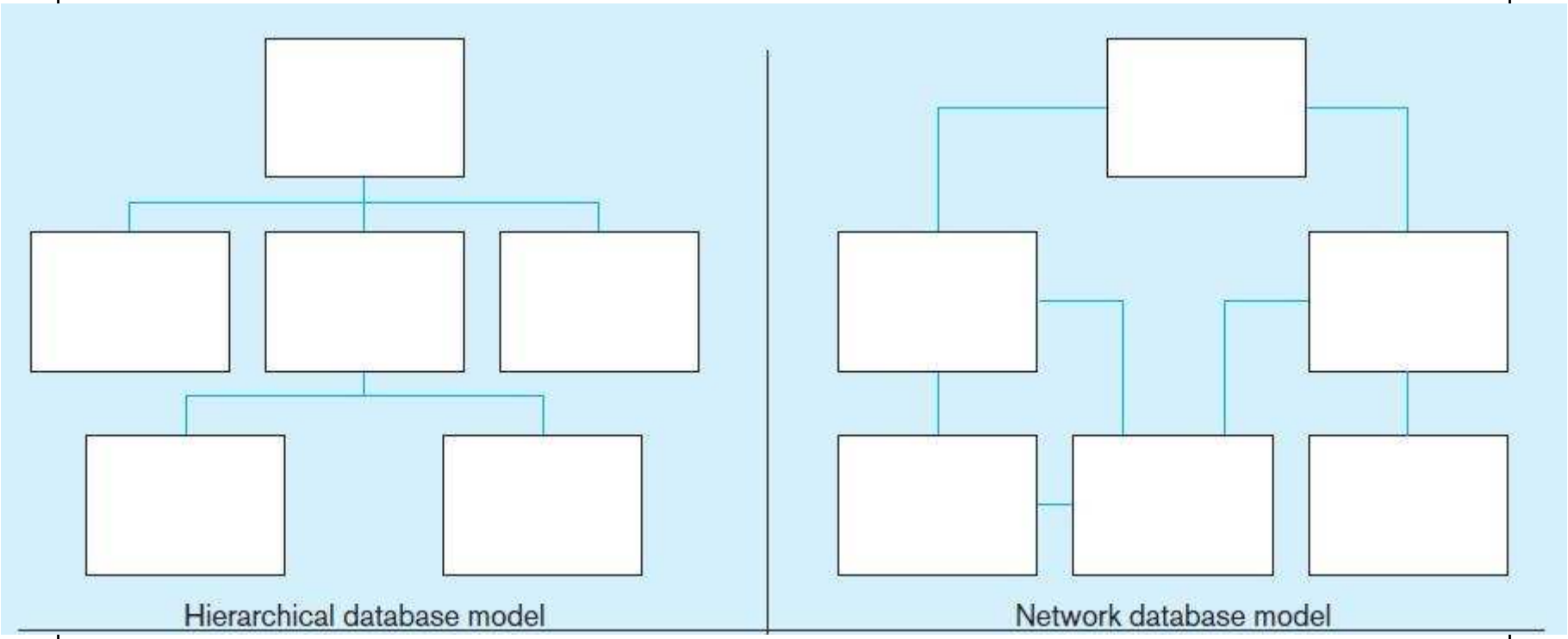
Evolution of Database Systems

- **Driven by four main objectives:**
  - Need for program-data independence ➔ reduced maintenance
  - Desire to manage more complex data types and structures
  - Ease of data access for less technical personnel
  - Need for more powerful decision support platforms

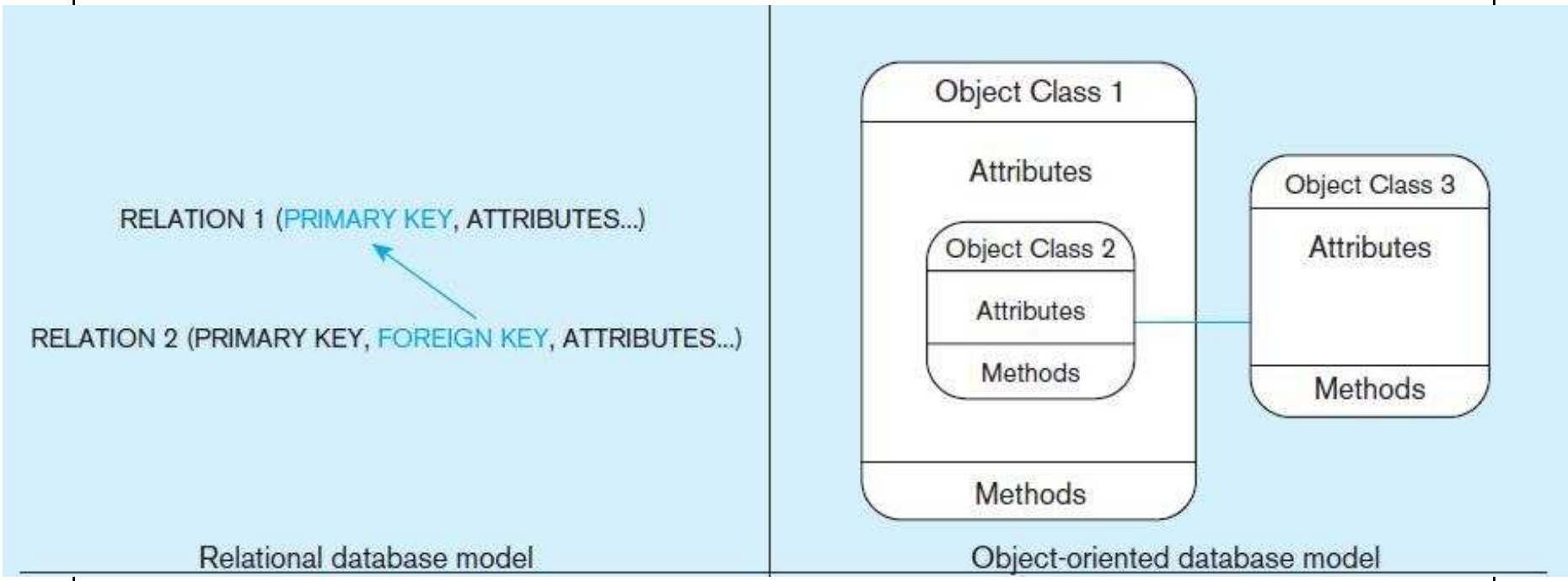
Evolution of DBMS



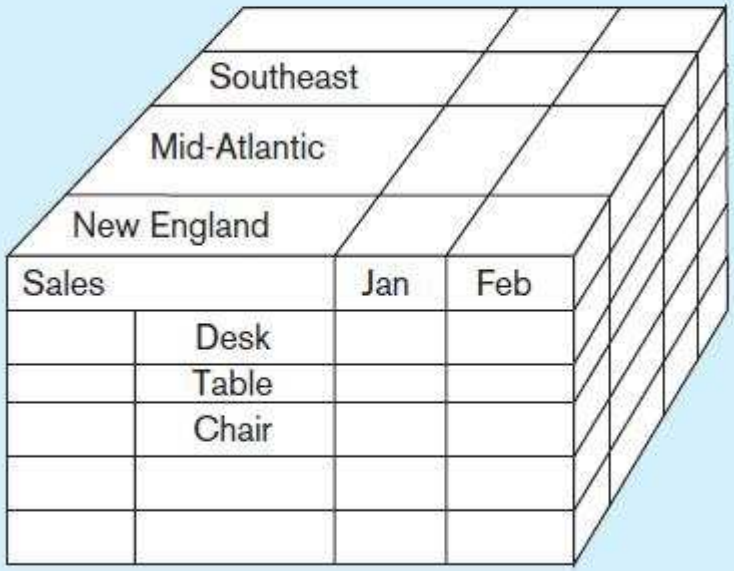
Database architectures



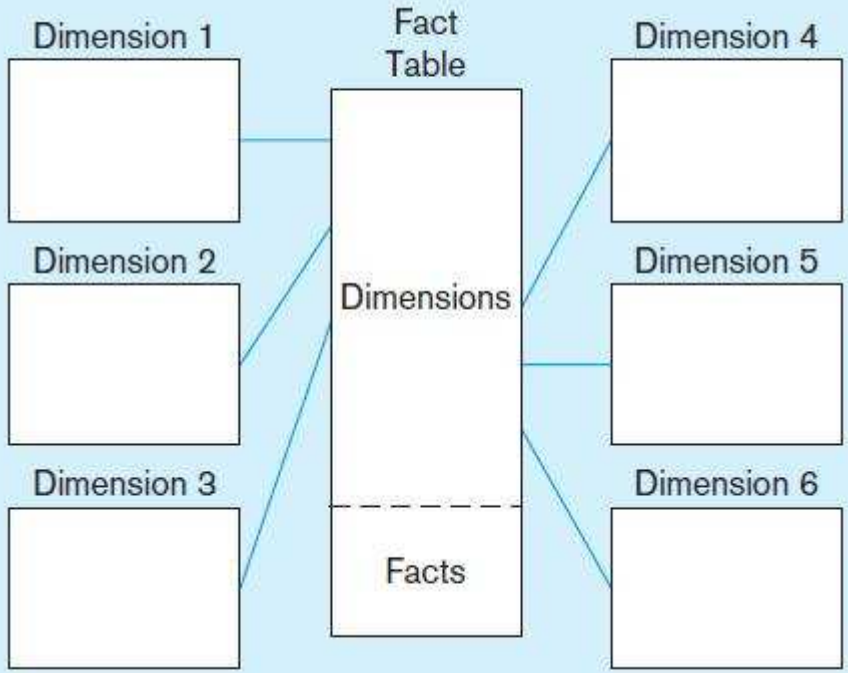
Database architectures (cont.)



Database architectures (cont.)



Multidimensional database model – multidimensional cube view



Multidimensional database model – star-schema view