Chapter 2

Data Models

Database Systems:
Design, Implementation, and Management,
Seventh Edition, Rob and Coronel

In this chapter, you will learn:

- Why data models are important
- About the basic data-modeling building blocks
- What business rules are and how they influence database design
- How the major data models evolved
- How data models can be classified by level of abstraction

The Importance of Data Models

- Data models
  - Relatively simple representations, usually graphical, of complex real-world data structures
  - Facilitate interaction among the designer, the applications programmer, and the end user

The Importance of Data Models (continued)

- End-users have different views and needs for data
- Data model organizes data for various users

Data Model Basic Building Blocks

- Entity - anything about which data are to be collected and stored
- Attribute - a characteristic of an entity
- Relationship - describes an association among entities
  - One-to-many (1:M) relationship
  - Many-to-many (M:N or M:M) relationship
  - One-to-one (1:1) relationship
- Constraint - a restriction placed on the data

Business Rules

- Brief, precise, and unambiguous descriptions of a policies, procedures, or principles within a specific organization
- Apply to any organization that stores and uses data to generate information
- Description of operations that help to create and enforce actions within that organization's environment

Business Rules (continued)

- Must be rendered in writing
- Must be kept up to date
- Sometimes are external to the organization
- Must be easy to understand and widely disseminated
- Describe characteristics of the data as viewed by the company

Discovering Business Rules

Sources of Business Rules:

- Company managers
- Policy makers
- Department managers
- Written documentation
  - Procedures
  - Standards
  - Operations manuals
- Direct interviews with end users

Translating Business Rules into Data Model Components

- Standardize company's view of data
- Constitute a communications tool between users and designers
- Allow designer to understand the nature, role, and scope of data
- Allow designer to understand business processes
- Allow designer to develop appropriate relationship participation rules and constraints
- Promote creation of an accurate data model

Discovering Business Rules (continued)

- Generally, nouns translate into entities
- Verbs translate into relationships among entities
- Relationships are bi-directional

The Evolution of Data Models

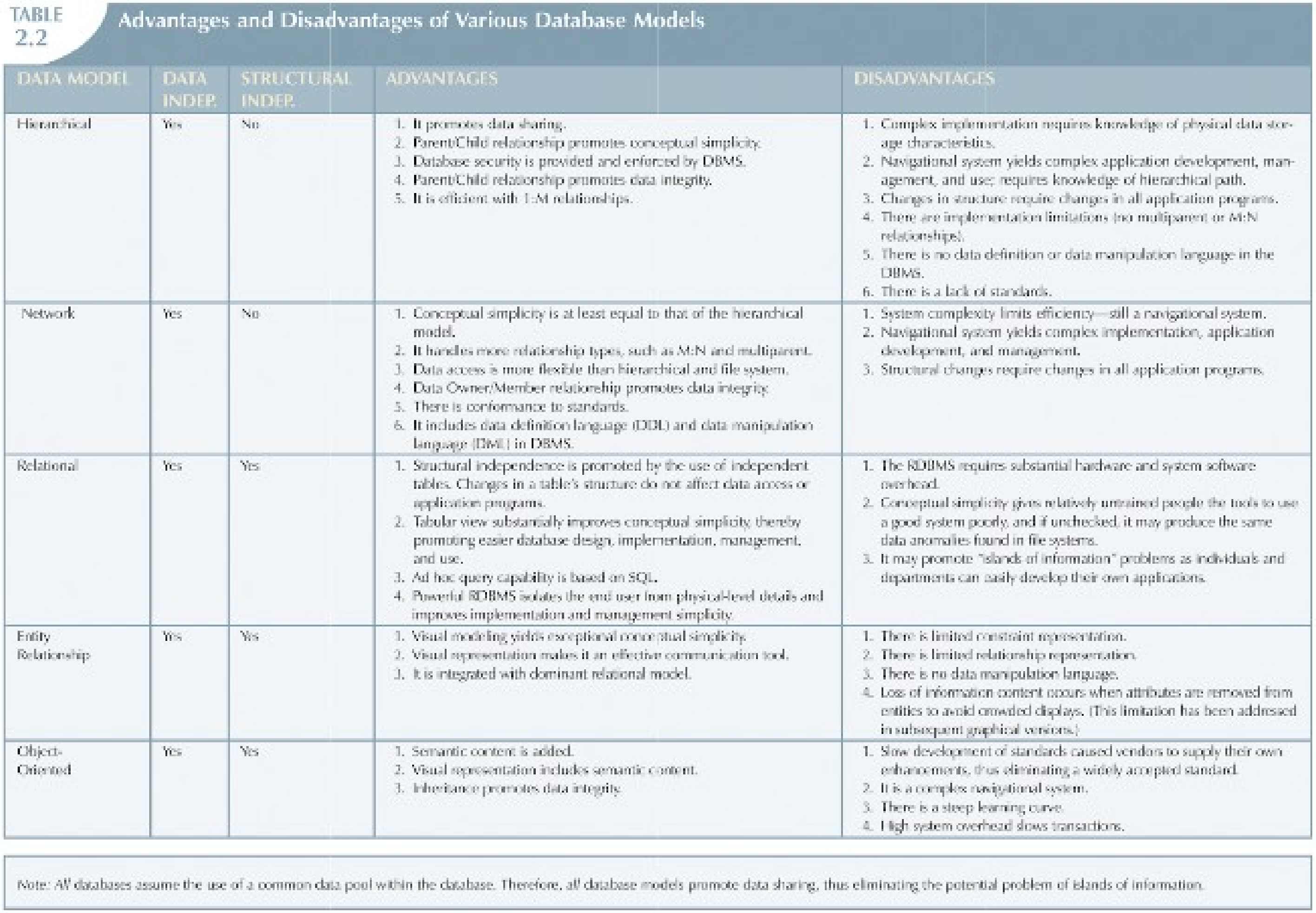


TABLE 2.2 Advantages and Disadvantages of Various Database Models

DATA MODEL	DATA INDEP.	STRUCTURAL INDEP.	ADVANTAGES	DISADVANTAGES
Hierarchical	Yes	No	1. It promotes data sharing. 2. Parent/Child relationship promotes conceptual simplicity. 3. Database security is provided and enforced by DBMS. 4. Parent/Child relationship promotes data integrity. 5. It is efficient with 1:M relationships.	1. Complex implementation requires knowledge of physical data storage characteristics. 2. Navigational system yields complex application development, management, and use; requires knowledge of hierarchical path. 3. Changes in structure require changes in all application programs. 4. There are implementation limitations (no multiparent or M:N relationships). 5. There is no data definition or data manipulation language in the DBMS. 6. There is a lack of standards.
Network	Yes	No	1. Conceptual simplicity is at least equal to that of the hierarchical model. 2. It handles more relationship types, such as M:N and multiparent. 3. Data access is more flexible than hierarchical and file system. 4. Data Owner/Member relationship promotes data integrity. 5. There is conformance to standards. 6. It includes data definition language (DDL) and data manipulation language (DML) in DBMS.	1. System complexity limits efficiency—still a navigational system. 2. Navigational system yields complex implementation, application development, and management. 3. Structural changes require changes in all application programs.
Relational	Yes	Yes	1. Structural independence is promoted by the use of independent tables. Changes in a table's structure do not affect data access or application programs. 2. Tabular view substantially improves conceptual simplicity, thereby promoting easier database design, implementation, management, and use. 3. Ad hoc query capability is based on SQL. 4. Powerful RDBMS isolates the end user from physical-level details and improves implementation and management simplicity.	1. The RDBMS requires substantial hardware and system software overhead. 2. Conceptual simplicity gives relatively untrained people the tools to use a good system poorly, and if unchecked, it may produce the same data anomalies found in file systems. 3. It may promote "islands of information" problems as individuals and departments can easily develop their own applications.
Entity Relationship	Yes	Yes	1. Visual modeling yields exceptional conceptual simplicity. 2. Visual representation makes it an effective communication tool. 3. It is integrated with dominant relational model.	1. There is limited constraint representation. 2. There is limited relationship representation. 3. There is no data manipulation language. 4. Loss of information content occurs when attributes are removed from entities to avoid crowded displays. (This limitation has been addressed in subsequent graphical versions.)
Object-Oriented	Yes	Yes	1. Semantic content is added. 2. Visual representation includes semantic content. 3. Inheritance promotes data integrity.	1. Slow development of standards caused vendors to supply their own enhancements, thus eliminating a widely accepted standard. 2. It is a complex navigational system. 3. There is a steep learning curve. 4. High system overhead slows transactions.

Note: *All* databases assume the use of a common data pool within the database. Therefore, *all* database models promote data sharing, thus eliminating the potential problem of islands of information.

The Evolution of Data Models (continued)

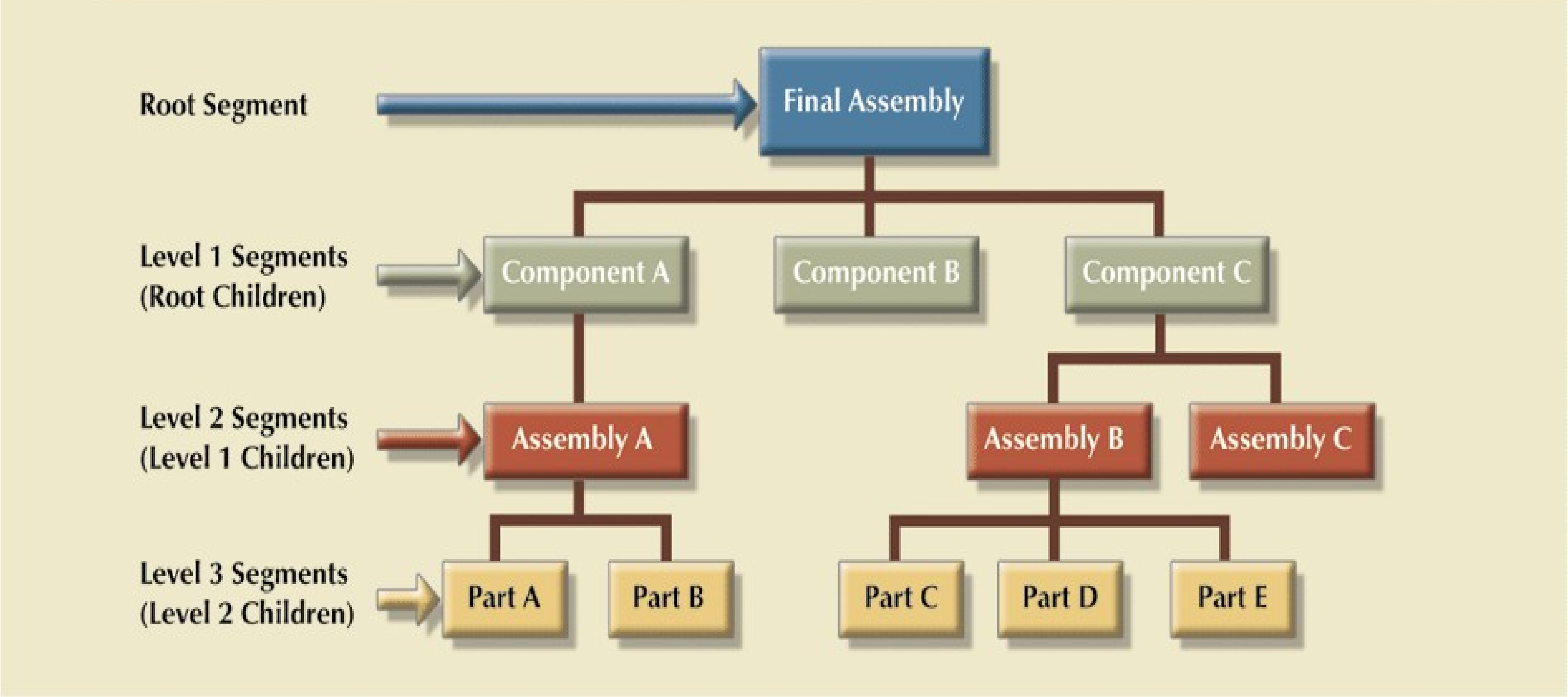
- Hierarchical
- Network
- Relational
- Entity relationship
- Object oriented (OO)

The Hierarchical Model

- Developed in the 1960s to manage large amounts of data for complex manufacturing projects
- Basic logical structure is represented by an upside-down "tree"

The Hierarchical Model (continued)

FIGURE 2.1 A hierarchical structure



The Hierarchical Model (continued)

- The hierarchical structure contains levels, or segments
- Depicts a set of one-to-many (1:M) relationships between a parent and its children segments
  - Each parent can have many children
  - each child has only one parent

The Hierarchical Model (continued)

- Advantages
  - Many of the hierarchical data model’s features formed the foundation for current data models
  - Its database application advantages are replicated, albeit in a different form, in current database environments
  - Generated a large installed (mainframe) base, created a pool of programmers who developed numerous tried-and-true business applications

The Hierarchical Model (continued)

- Disadvantages
  - Complex to implement
  - Difficult to manage
  - Lacks structural independence
  - Implementation limitations
  - Lack of standards

The Network Model

- Created to
  - Represent complex data relationships more effectively
  - Improve database performance
  - Impose a database standard
- Conference on Data Systems Languages (CODASYL)
- Database Task Group (DBTG)

The Network Model (continued)

- Schema
  - Conceptual organization of entire database as viewed by the database administrator
- Subschema
  - Defines database portion “seen” by the application programs that actually produce the desired information from data contained within the database
- Data Management Language (DML)
  - Defines the environment in which data can be managed

The Network Model (continued)

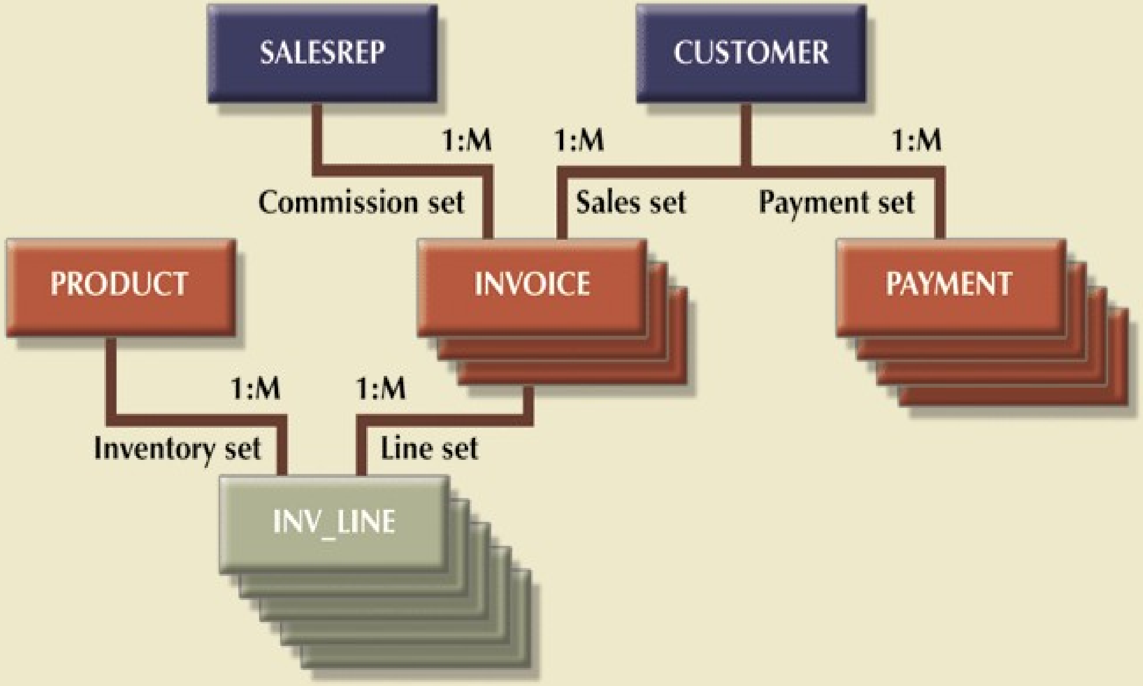
- Schema Data Definition Language (DDL)
  - Enables database administrator to define schema components
- Subschema DDL
  - Allows application programs to define database components that will be used
- DML
  - Works with the data in the database

The Network Model (continued)

- Resembles hierarchical model
- Collection of records in 1:M relationships
- Set
  - Relationship
  - Composed of at least two record types
    - Owner
      - Equivalent to the hierarchical model’s parent
    - Member
      - Equivalent to the hierarchical model’s child

The Network Model (continued)

FIGURE 2.2 A network data model



The Network Model (continued)

- Disadvantages
  - – Too cumbersome
  - – The lack of ad hoc query capability put heavy pressure on programmers
  - – Any structural change in the database could produce havoc in all application programs that drew data from the database
  - – Many database old-timers can recall the interminable information delays

The Relational Model

- Developed by Codd (IBM) in 1970
- Considered ingenious but impractical in 1970
- Conceptually simple
- Computers lacked power to implement the relational model
- Today, microcomputers can run sophisticated relational database software

The Relational Model (continued)

- Relational Database Management System (RDBMS)
- Performs same basic functions provided by hierarchical and network DBMS systems, in addition to a host of other functions
- Most important advantage of the RDBMS is its ability to hide the complexities of the relational model from the user

The Relational Model (continued)

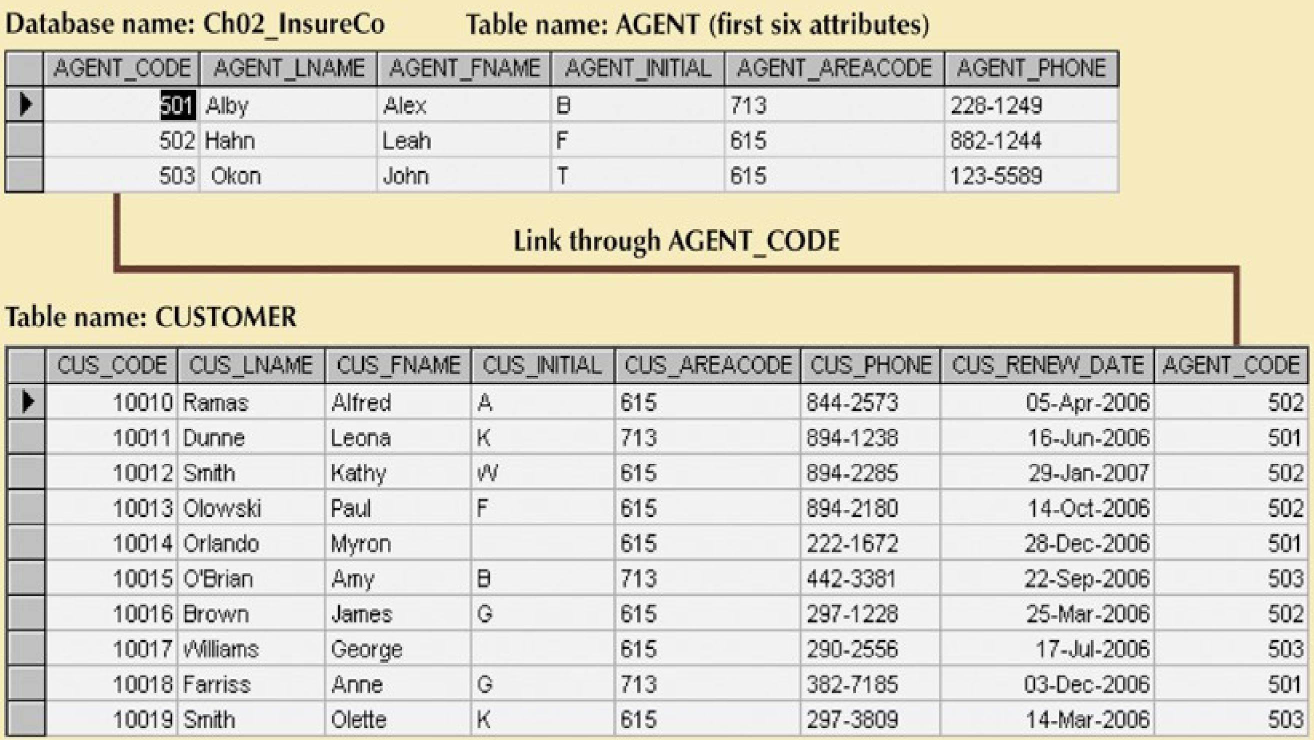
- Table (relations)
  - Matrix consisting of a series of row/column intersections
  - Related to each other through sharing a common entity characteristic
- Relational diagram
  - Representation of relational database’s entities, attributes within those entities, and relationships between those entities

The Relational Model (continued)

- Relational Table
  - Stores a collection of related entities
    - Resembles a file
- Relational table is purely logical structure
  - How data are physically stored in the database is of no concern to the user or the designer
  - This property became the source of a real database revolution

The Relational Model (continued)

FIGURE 2.3 Linking relational tables



Database name: Ch02_InsureCo

Table name: AGENT (first six attributes)

AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
501	Alby	Alex	B	713	228-1249
502	Hahn	Leah	F	615	882-1244
503	Okon	John	T	615	123-5589

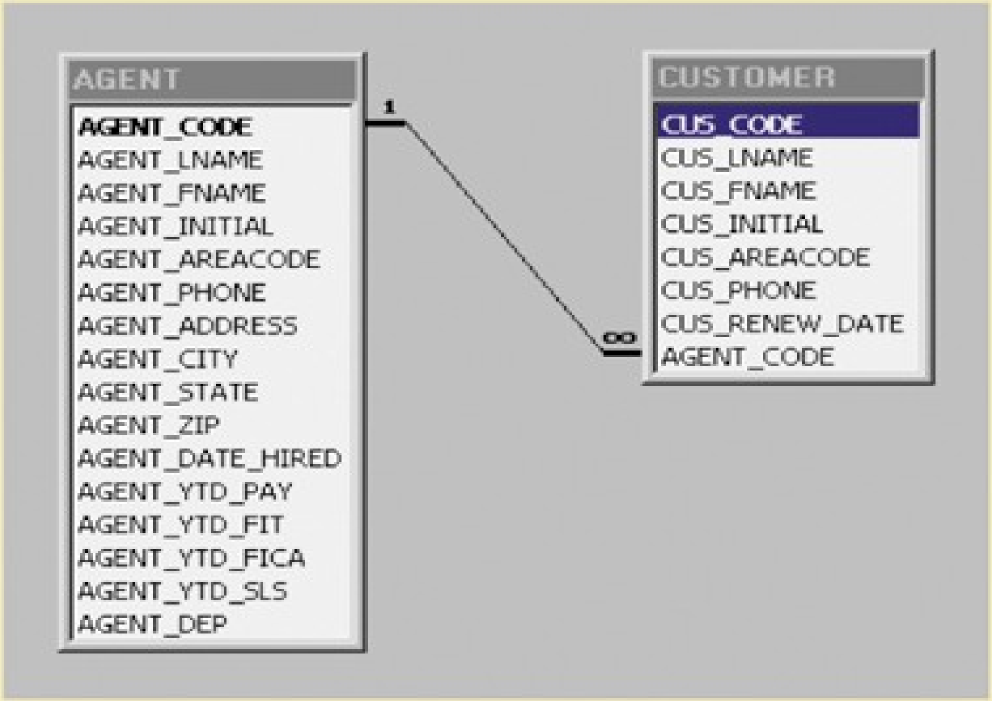
Link through AGENT_CODE

Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	05-Apr-2006	502
10011	Dunne	Leona	K	713	894-1238	16-Jun-2006	501
10012	Smith	Kathy	W	615	894-2285	29-Jan-2007	502
10013	Olowski	Paul	F	615	894-2180	14-Oct-2006	502
10014	Orlando	Myron		615	222-1672	28-Dec-2006	501
10015	O'Brian	Amy	B	713	442-3381	22-Sep-2006	503
10016	Brown	James	G	615	297-1228	25-Mar-2006	502
10017	Williams	George		615	290-2556	17-Jul-2006	503
10018	Farriss	Anne	G	713	382-7185	03-Dec-2006	501
10019	Smith	Olette	K	615	297-3809	14-Mar-2006	503

The Relational Model (continued)

FIGURE 2.4 A relational diagram



AGENT

AGENT_CODE
AGENT_LNAME
AGENT_FNAME
AGENT_INITIAL
AGENT_AREACODE
AGENT_PHONE
AGENT_ADDRESS
AGENT_CITY
AGENT_STATE
AGENT_ZIP
AGENT_DATE_HIRED
AGENT_YTD_PAY
AGENT_YTD_FIT
AGENT_YTD_FICA
AGENT_YTD_SLS
AGENT_DEP

CUSTOMER

CUS_CODE
CUS_LNAME
CUS_FNAME
CUS_INITIAL
CUS_AREACODE
CUS_PHONE
CUS_RENEW_DATE
AGENT_CODE

The Relational Model (continued)

- Rise to dominance due in part to its powerful and flexible query language
- Structured Query Language (SQL) allows the user to specify what must be done without specifying how it must be done
- SQL-based relational database application involves:
  - User interface
  - A set of tables stored in the database
  - SQL engine

The Entity Relationship Model

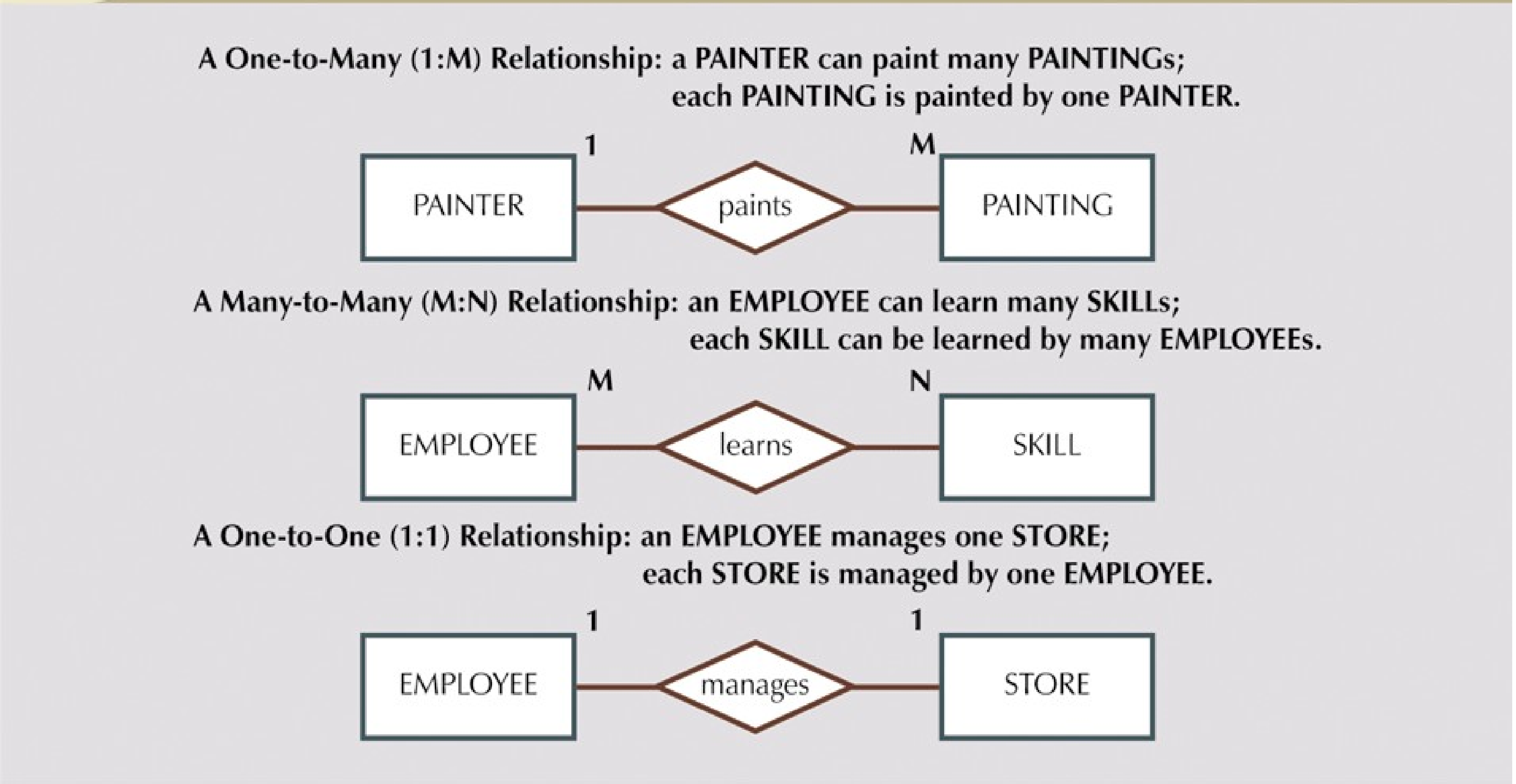
- Widely accepted and adapted graphical tool for data modeling
- Introduced by Chen in 1976
- Graphical representation of entities and their relationships in a database structure

The Entity Relationship Model (continued)

- Entity relationship diagram (ERD)
  - Uses graphic representations to model database components
  - Entity is mapped to a relational table
- Entity instance (or occurrence) is row in table
- Entity set is collection of like entities
- Connectivity labels types of relationships
  - Diamond connected to related entities through a relationship line

The Entity Relationship Model (continued)

FIGURE 2.5 The basic Chen ERD



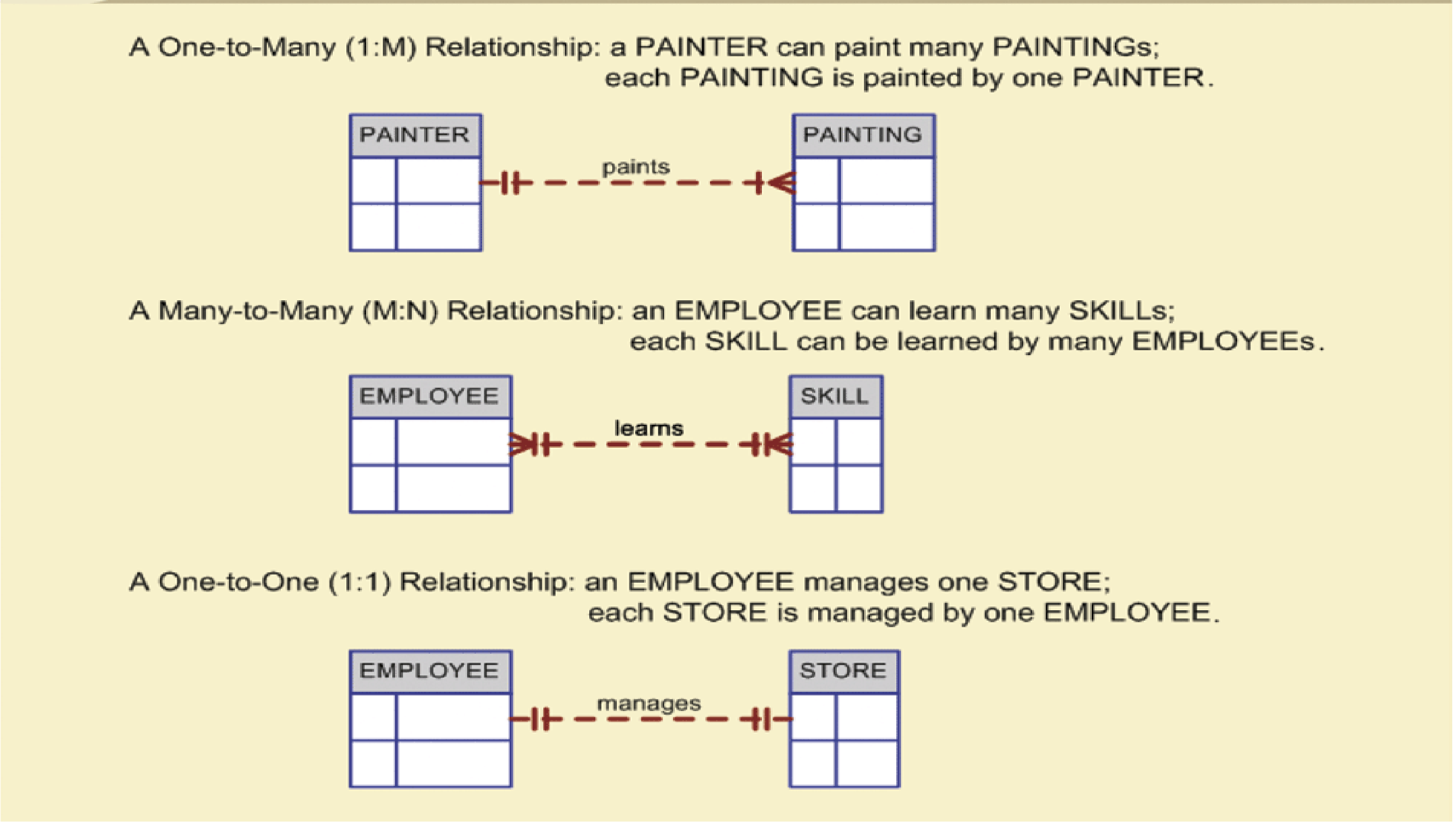
A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.

A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.

A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.

The Entity Relationship Model (continued)

FIGURE 2.6 The basic Crow's foot ERD



A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.

A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.

A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.

The Object Oriented Model

- Modeled both data and their relationships in a single structure known as an object
- Object-oriented data model (OODM) is the basis for the object-oriented database management system (OODBMS)
- OODM is said to be a semantic data model

The Object Oriented Model (continued)

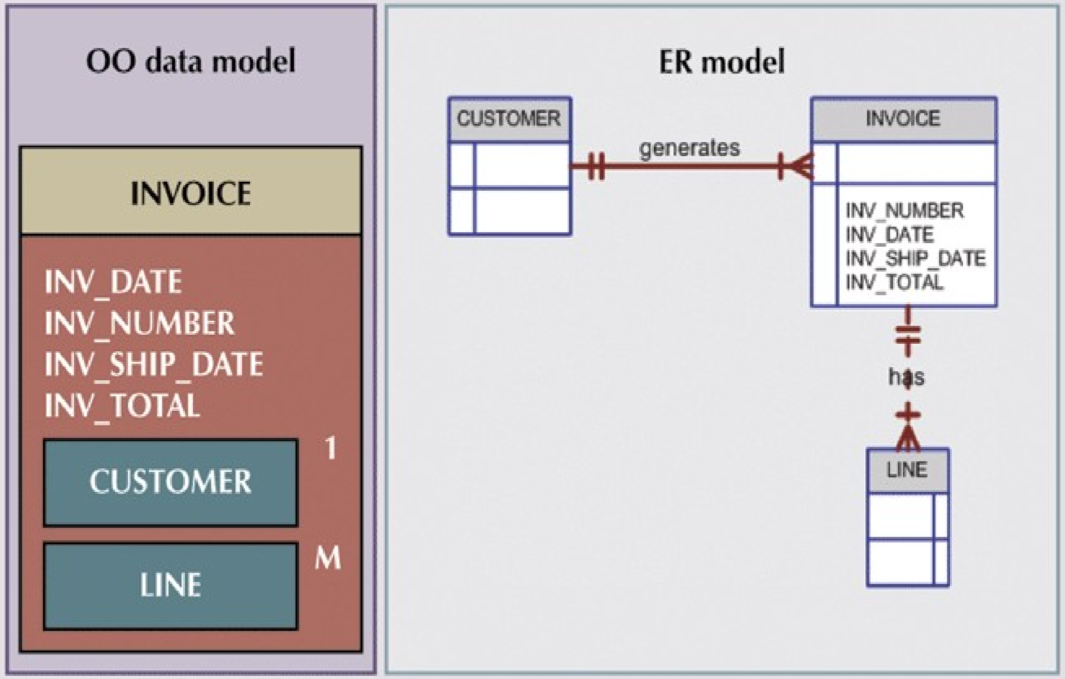
- Object described by its factual content
  - Like relational model's entity
- Includes information about relationships between facts within object, and relationships with other objects
  - Unlike relational model's entity
- Subsequent OODM development allowed an object to also contain all operations
- Object becomes basic building block for autonomous structures

The Object Oriented Model (continued)

- Object is an abstraction of a real-world entity
- Attributes describe the properties of an object
- Objects that share similar characteristics are grouped in classes
- Classes are organized in a class hierarchy
- Inheritance is the ability of an object within the class hierarchy to inherit the attributes and methods of classes above it

The Object Oriented Model (continued)

FIGURE 2.7 A comparison of the OO model and the ER model



Other Models

- Extended Relational Data Model (ERDM)
  - Semantic data model developed in response to increasing complexity of applications
  - DBMS based on the ERDM often described as an object/relational database management system (O/RDBMS)
  - Primarily geared to business applications

Database Models and the Internet

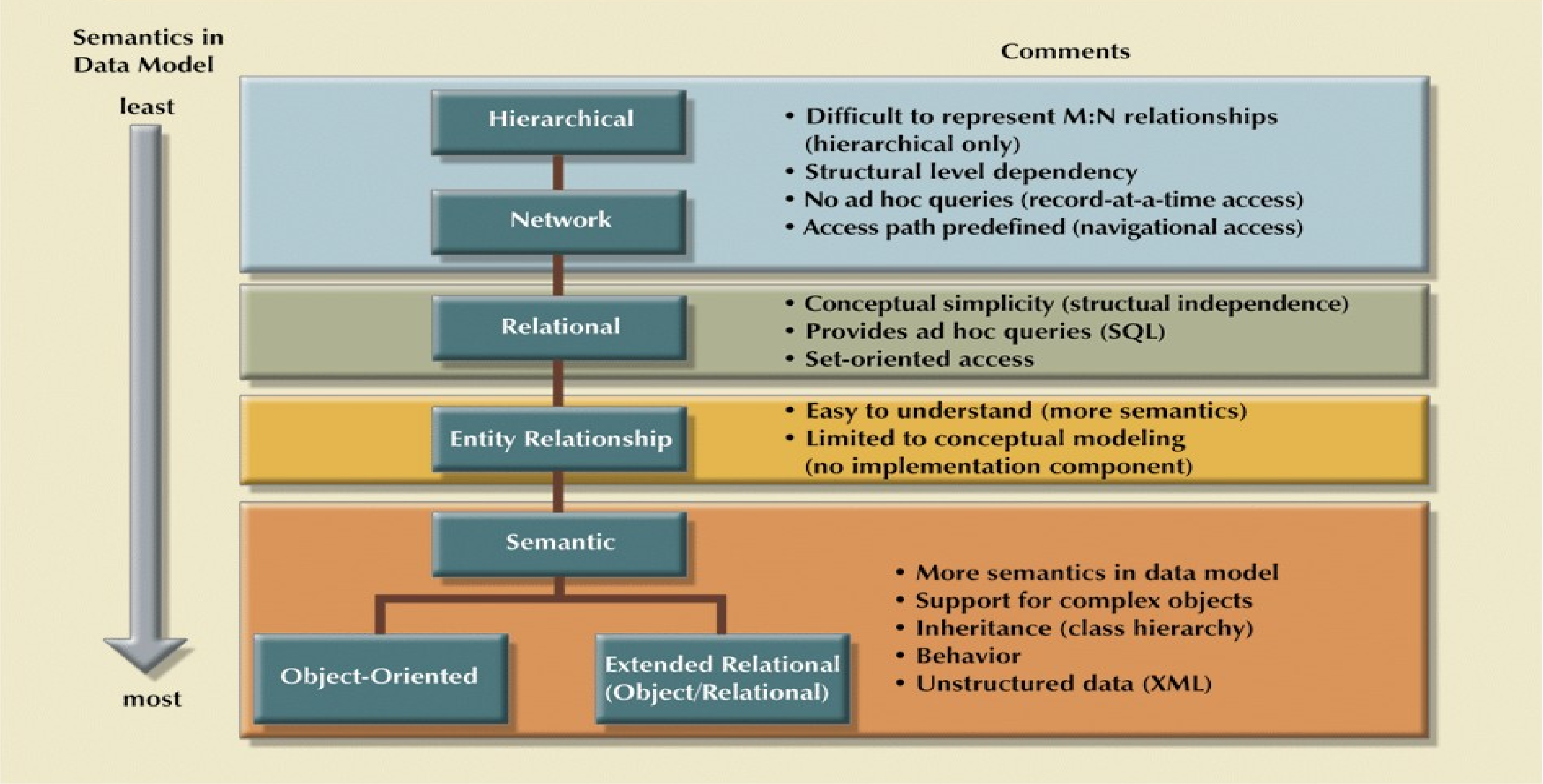
- Internet drastically changed role and scope of database market
- OODM and ERDM-O/RDM have taken a backseat to development of databases that interface with Internet
- Dominance of Web has resulted in growing need to manage unstructured information

Data Models: A Summary

- Each new data model capitalized on the shortcomings of previous models
- Common characteristics:
  - Conceptual simplicity without compromising the semantic completeness of the database
  - Represent the real world as closely as possible
  - Representation of real-world transformations (behavior) must comply with consistency and integrity characteristics of any data model

Data Models: A Summary (continued)

FIGURE 2.8 The development of data models



- Difficult to represent M:N relationships (hierarchical only)
- Structural level dependency
- No ad hoc queries (record-at-a-time access)
- Access path predefined (navigational access)

- Conceptual simplicity (structual independence)
- Provides ad hoc queries (SQL)
- Set-oriented access

- Easy to understand (more semantics)
- Limited to conceptual modeling (no implementation component)

- More semantics in data model
- Support for complex objects
- Inheritance (class hierarchy)
- Behavior
- Unstructured data (XML)

Degrees of Data Abstraction

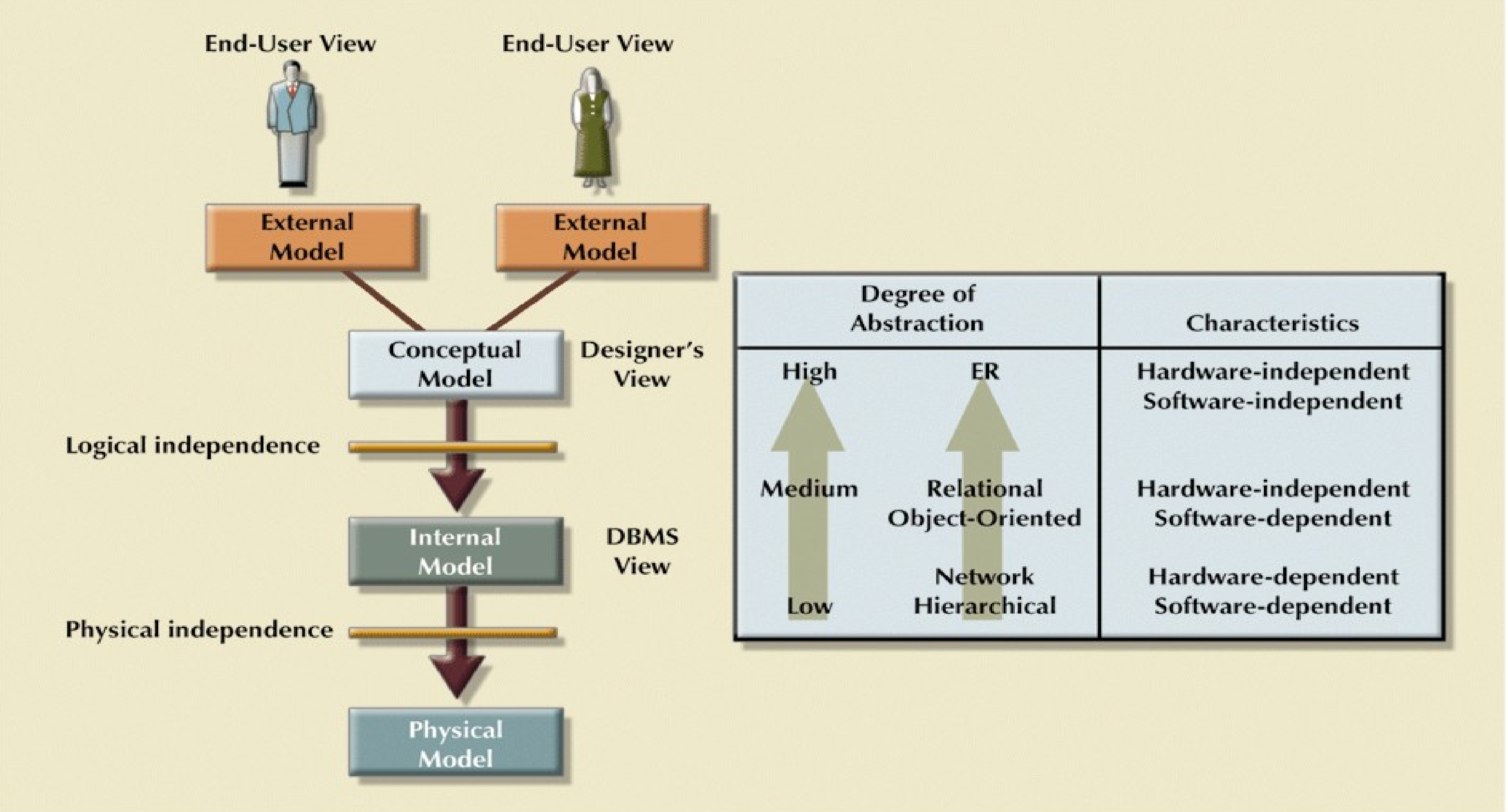
- Way of classifying data models
- Many processes begin at high level of abstraction and proceed to an ever-increasing level of detail
- Designing a usable database follows the same basic process

Degrees of Data Abstraction (continued)

- American National Standards Institute (ANSI) Standards Planning and Requirements Committee (SPARC)
  - Defined a framework for data modeling based on degrees of data abstraction(1970s):
    - External
    - Conceptual
    - Internal

Degrees of Data Abstraction (continued)

FIGURE 2.9 Data abstraction levels



Degree of Abstraction		Characteristics
High	ER	Hardware-independent Software-independent
Medium	Relational Object-Oriented	Hardware-independent Software-dependent
Low	Network Hierarchical	Hardware-dependent Software-dependent

The External Model

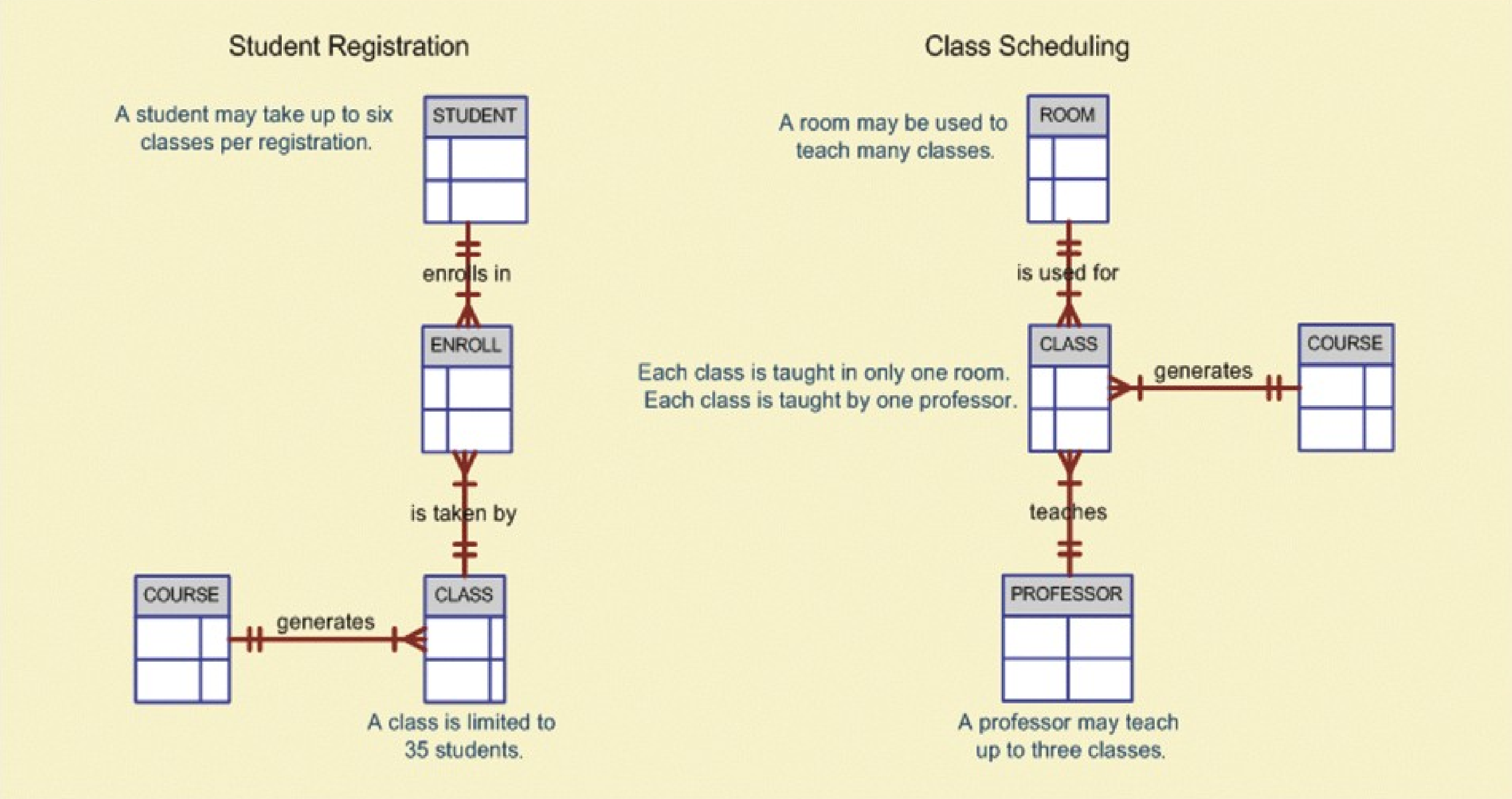
- End users’ view of the data environment
- Requires that the modeler subdivide set of requirements and constraints into functional modules that can be examined within the framework of their external models

The External Model (continued)

- Advantages:
  - Easy to identify specific data required to support each business unit’s operations
  - Facilitates designer’s job by providing feedback about the model’s adequacy
  - Creation of external models helps to ensure security constraints in the database design
  - Simplifies application program development

The External Model (continued)

FIGURE 2.10 External models for Tiny College

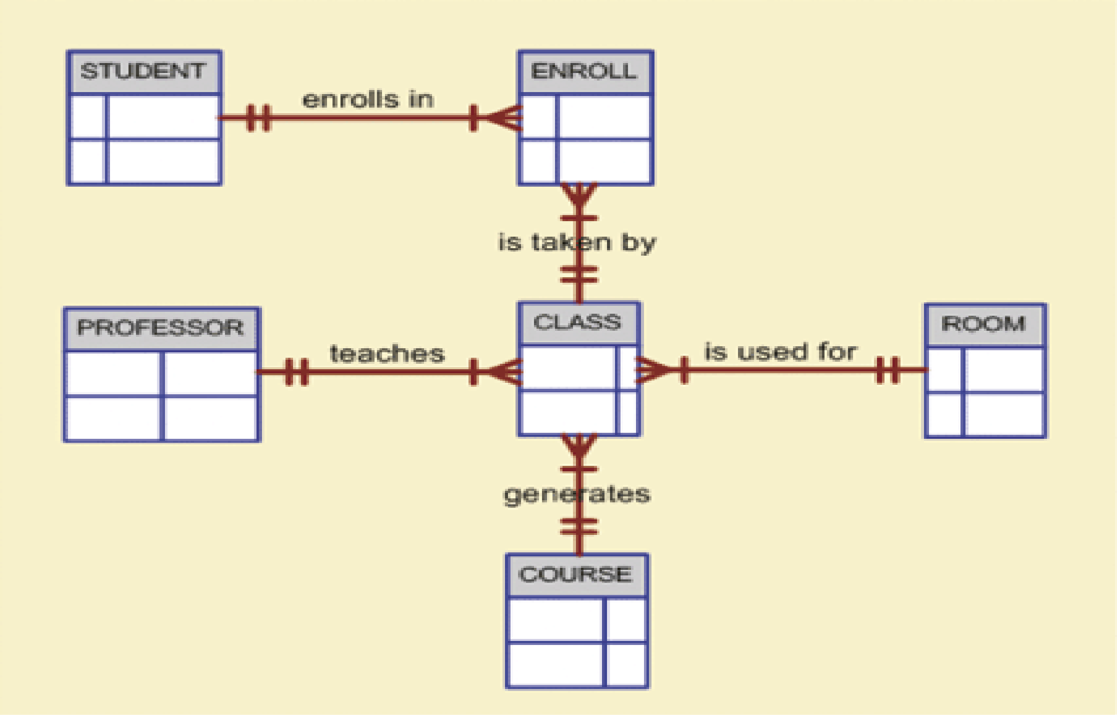

Student Registration
A student may take up to six classes per registration.
STUDENT
enrolls in
ENROLL
is taken by
CLASS
COURSE
generates
A class is limited to 35 students.
Class Scheduling
A room may be used to teach many classes.
ROOM
is used for
CLASS
Each class is taught in only one room.
Each class is taught by one professor.
generates
COURSE
teaches
PROFESSOR
A professor may teach up to three classes.

The Conceptual Model

- Represents global view of the entire database
- Representation of data as viewed by the entire organization
- Basis for identification and high-level description of main data objects, avoiding details
- Most widely used conceptual model is the entity relationship (ER) model

The Conceptual Model (continued)

FIGURE 2.11 Conceptual model for Tiny College



The Conceptual Model (continued)

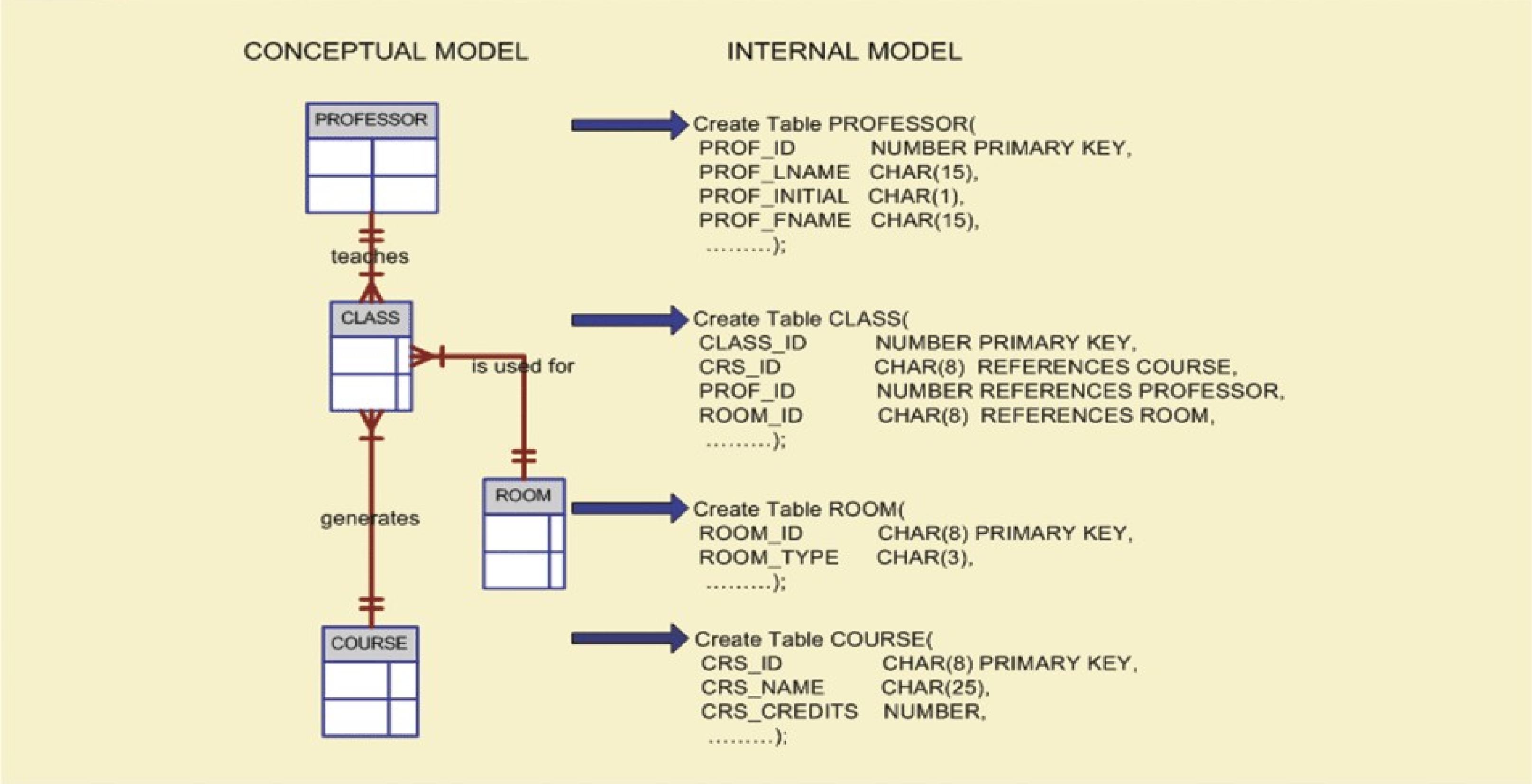
- Provides a relatively easily understood macro level view of data environment
- Independent of both software and hardware
  - Does not depend on the DBMS software used to implement the model
  - Does not depend on the hardware used in the implementation of the model
  - Changes in either hardware or DBMS software have no effect on the database design at the conceptual level

The Internal Model

- Representation of the database as “seen” by the DBMS
- Maps the conceptual model to the DBMS
- Internal schema depicts a specific representation of an internal model

The Internal Model (continued)

FIGURE 2.12 An internal model for Tiny College



```
Create Table PROFESSOR(
 PROF_ID          NUMBER PRIMARY KEY,
 PROF_LNAME       CHAR(15),
 PROF_INITIAL     CHAR(1),
 PROF_FNAME       CHAR(15),
  .........);

Create Table CLASS(
 CLASS_ID         NUMBER PRIMARY KEY,
 CRS_ID           CHAR(8)  REFERENCES COURSE,
 PROF_ID          NUMBER REFERENCES PROFESSOR,
 ROOM_ID          CHAR(8)  REFERENCES ROOM,
  .........);

Create Table ROOM(
 ROOM_ID          CHAR(8) PRIMARY KEY,
 ROOM_TYPE        CHAR(3),
  .........);

Create Table COURSE(
 CRS_ID           CHAR(8) PRIMARY KEY,
 CRS_NAME         CHAR(25),
 CRS_CREDITS      NUMBER,
  .........);
```

The Physical Model

- Operates at lowest level of abstraction, describing the way data are saved on storage media such as disks or tapes
- Software and hardware dependent
- Requires that database designers have a detailed knowledge of the hardware and software used to implement database design

The Physical Model (continued)

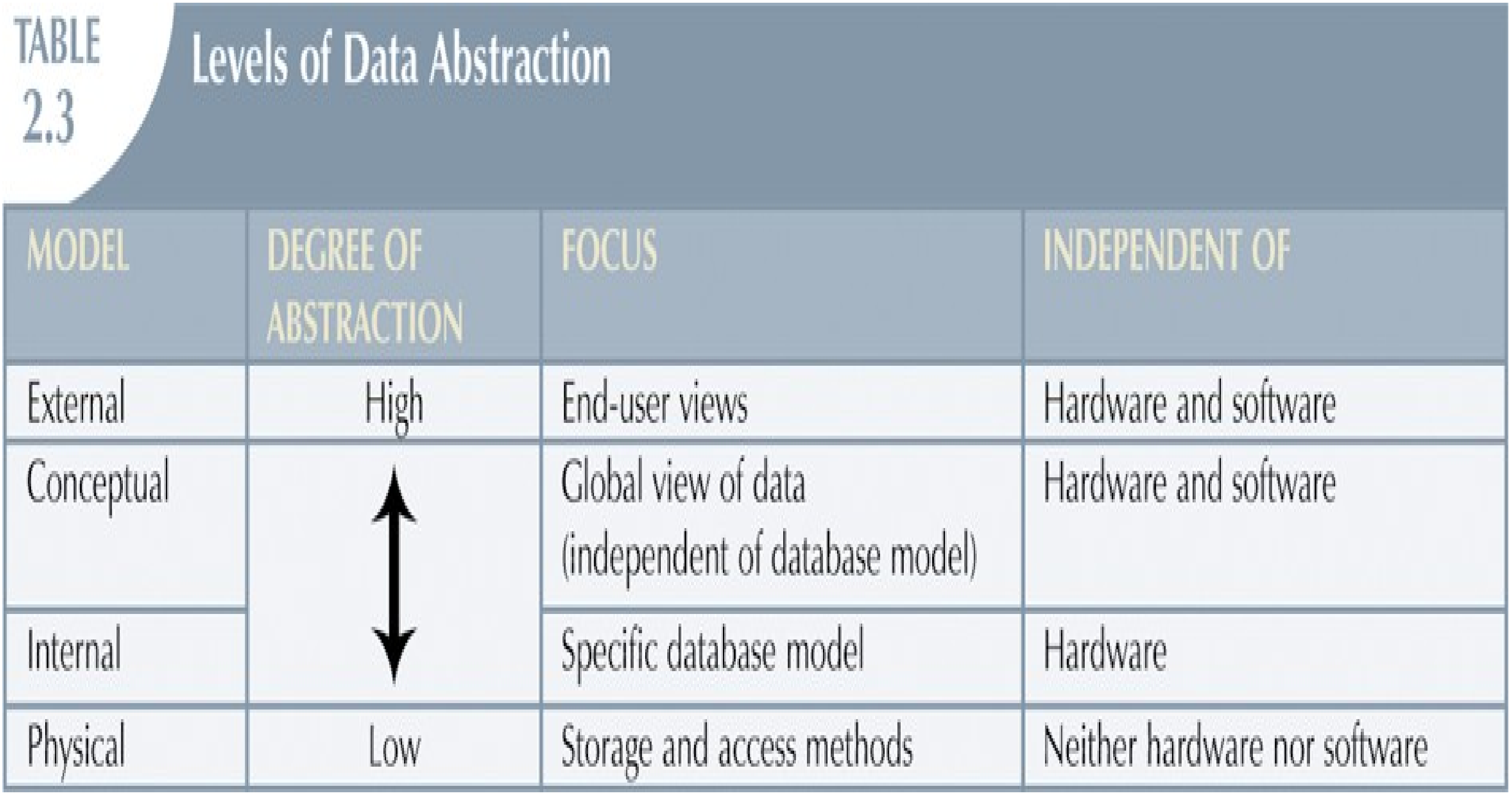


TABLE 2.3 Levels of Data Abstraction

MODEL	DEGREE OF ABSTRACTION	FOCUS	INDEPENDENT OF
External	High	End-user views	Hardware and software
Conceptual	↕	Global view of data (independent of database model)	Hardware and software
Internal		Specific database model	Hardware
Physical	Low	Storage and access methods	Neither hardware nor software

Summary

- A data model is a (relatively) simple abstraction of a complex real-world data environment
- Basic data modeling components are:
  - Entities
  - Attributes
  - Relationships
  - Constraints

Summary (continued)

- Hierarchical model
  - Depicts a set of one-to-many (1:M) relationships between a parent and its children segments
- Network data model
  - Uses sets to represent 1:M relationships between record types
- Relational model
  - Current database implementation standard
  - ER model is a popular graphical tool for data modeling that complements the relational model

Summary (continued)

- Object is basic modeling structure of object oriented data model
- The relational model has adopted many object-oriented extensions to become the extended relational data model (ERDM)
- Data modeling requirements are a function of different data views (global vs. local) and level of data abstraction