Chapter 2
Data Models

Learning Objectives

- In this chapter, you will learn:
  - About data modeling and why data models are important
  - About the basic data-modeling building blocks
  - What business rules are and how they influence database design

Learning Objectives

- In this chapter, you will learn:
  - How the major data models evolved
  - About emerging alternative data models and the need they fulfill
  - How data models can be classified by their level of abstraction

Data Modeling and Data Models

- **Data modeling**: Iterative and progressive process of creating a specific data model for a determined problem domain
- **Data models**: Simple representations of complex real-world data structures
  - Useful for supporting a specific problem domain
- **Model** - Abstraction of a real-world object or event

Importance of Data Models

- Are a communication tool
- Give an overall view of the database
- Organize data for various users
- Are an abstraction for the creation of good database

Data Model Basic Building Blocks

- **Entity**: Unique and distinct object used to collect and store data
  - **Attribute**: Characteristic of an entity
- **Relationship**: Describes an association among entities
  - **One-to-many (1:M)**
  - **Many-to-many (M:N or M:M)**
  - **One-to-one (1:1)**
- **Constraint**: Set of rules to ensure data integrity

Business Rules

Brief, precise, and unambiguous description of a policy, procedure, or principle

Enable defining the basic building blocks

Describe main and distinguishing characteristics of the data

Sources of Business Rules

Company managers

Policy makers

Department managers

Written documentation

Direct interviews with end users

Reasons for Identifying and Documenting Business Rules

- Help standardize company's view of data
- Communications tool between users and designers
- Allow designer to:
  - Understand the nature, role, scope of data, and business processes
  - Develop appropriate relationship participation rules and constraints
  - Create an accurate data model

Translating Business Rules into Data Model Components

- Nouns translate into entities
- Verbs translate into relationships among entities
- Relationships are bidirectional
- Questions to identify the relationship type
  - How many instances of B are related to one instance of A?
  - How many instances of A are related to one instance of B?

Naming Conventions

- Entity names - Required to:
  - Be descriptive of the objects in the business environment
  - Use terminology that is familiar to the users
- Attribute name - Required to be descriptive of the data represented by the attribute
- Proper naming:
  - Facilitates communication between parties
  - Promotes self-documentation

Hierarchical and Network Models

Hierarchical Models	Network Models
▪ Developed to manage large amounts of data for complex manufacturing projects ▪ Represented by an upside-down tree which contains **segments** (equivalent of a file system's record type) ▪ Depicts a set of one-to-many (1:M) relationships	▪ Created to represent complex data relationships effectively ▪ Improved database performance and imposed a database standard ▪ Allows a record to have more than one parent ▪ Depicts both one-to-many (1:M) and many-to-many (M:N) relationships

Standard Database Concepts

Schema

- Conceptual organization of the entire database as viewed by the database administrator

Subschema

- Portion of the database seen by the application programs that produce the desired information from the data within the database

Standard Database Concepts

Data manipulation language (DML)

- Environment in which data can be managed and is used to work with the data in the database

Schema data definition language (DDL)

- Enables the database administrator to define the schema components

The Relational Model

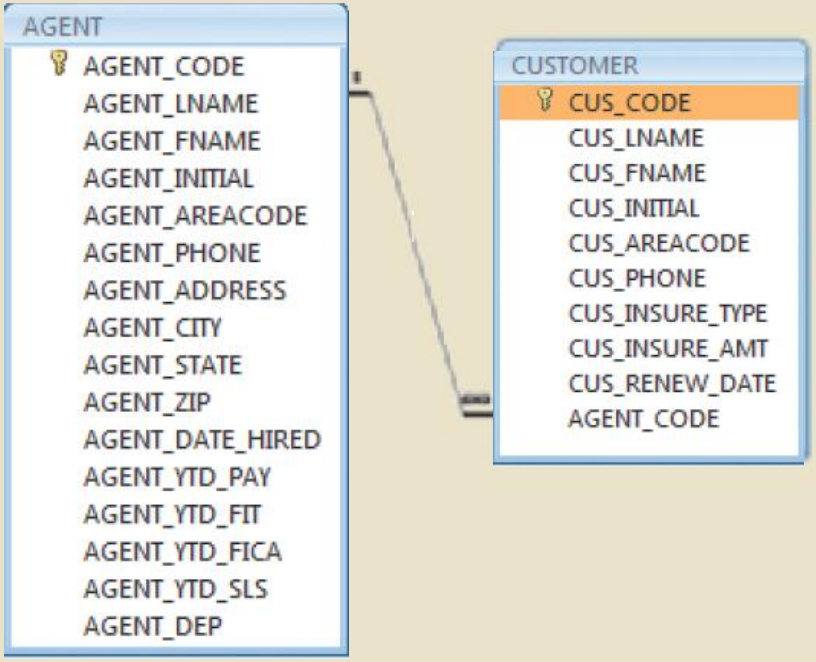
- Produced an automatic transmission database that replaced standard transmission databases
- Based on a relation
  - **Relation** or **table**: Matrix composed of intersecting tuple and attribute
    - **Tuple**: Rows
    - **Attribute**: Columns
- Describes a precise set of data manipulation constructs

Relational Database Management System (RDBMS)

- Performs basic functions provided by the hierarchical and network DBMS systems
- Makes the relational data model easier to understand and implement
- Hides the complexities of the relational model from the user

Figure 2.2 - A Relational Diagram

FIGURE 2.2 A RELATIONAL DIAGRAM



SQL-Based Relational Database Application

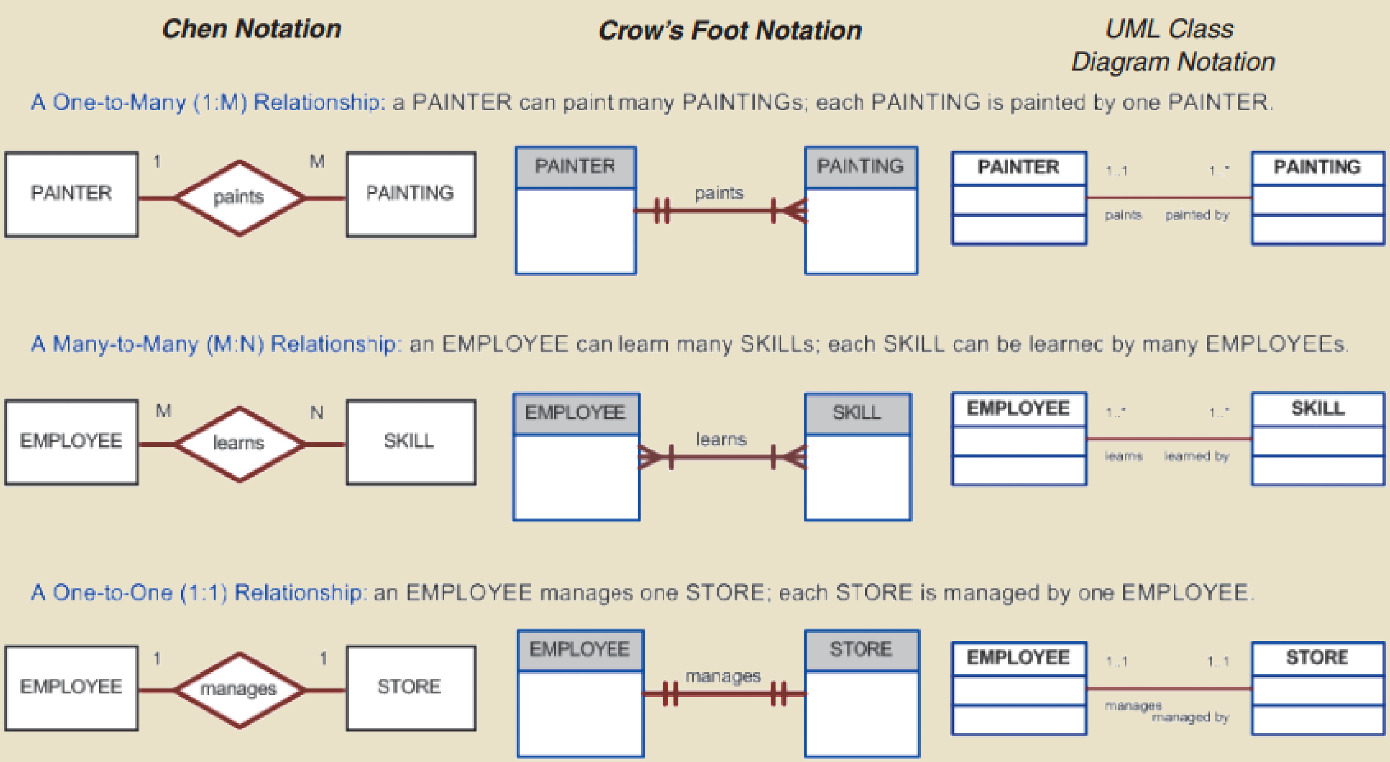
- End-user interface
  - Allows end user to interact with the data
- Collection of tables stored in the database
  - Each table is independent from another
  - Rows in different tables are related based on common values in common attributes
- SQL engine
  - Executes all queries

The Entity Relationship Model

- Graphical representation of entities and their relationships in a database structure
- **Entity relationship diagram (ERD)**
  - Uses graphic representations to model database components
- **Entity instance or entity occurrence**
  - Rows in the relational table
- **Connectivity**: Term used to label the relationship types

Figure 2.3 - The ER Model Notations

FIGURE 2.3 THE ER MODEL NOTATIONS



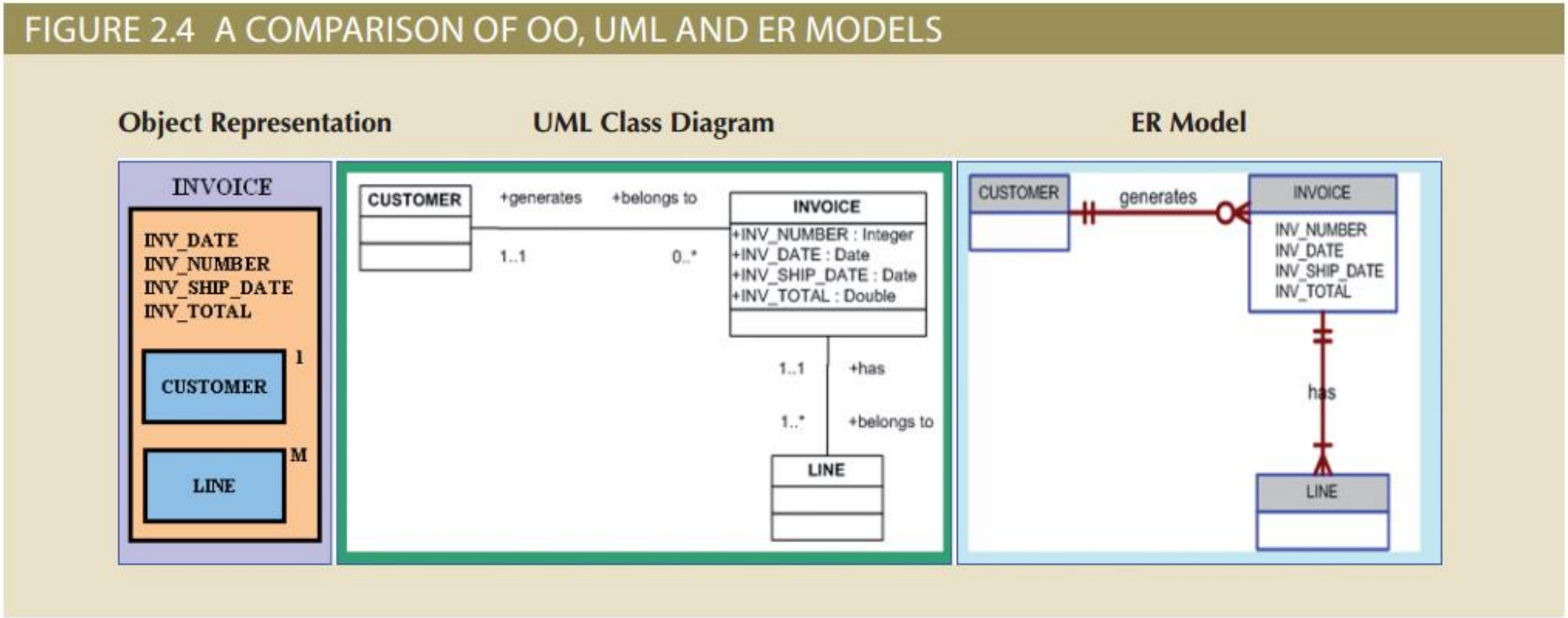
The Object-Oriented Data Model (OODM) or Semantic Data Model

- **Object-oriented database management system(OODBMS)**
  - Based on OODM
- **Object**: Contains data and their relationships with operations that are performed on it
  - Basic building block for autonomous structures
  - Abstraction of real-world entity
- Attributes - Describe the properties of an object

The Object-Oriented Data Model (OODM)

- **Class**: Collection of similar objects with shared structure and behavior organized in a class hierarchy
  - **Class hierarchy**: Resembles an upside-down tree in which each class has only one parent
- **Inheritance**: Object inherits methods and attributes of parent class
- **Unified Modeling Language (UML)**
  - Describes sets of diagrams and symbols to graphically model a system

Figure 2.4 - A Comparison of OO, UML and ER Models



Object/Relational and XML

- **Extended relational data model (ERDM)**
  - Supports OO features and complex data representation
  - **Object/Relational Database Management System (O/R DBMS)**
    - Based on ERDM, focuses on better data management
- **Extensible Markup Language (XML)**
  - Manages unstructured data for efficient and effective exchange of all data types

Big Data

- Aims to:
  - Find new and better ways to manage large amounts of web and sensor-generated data
  - Provide high performance and scalability at a reasonable cost
- Characteristics
  - Volume
  - Velocity
  - Variety

Big Data Challenges

- Volume does not allow the usage of conventional structures
- Expensive
- OLAP tools proved inconsistent dealing with unstructured data

Big Data New Technologies

Hadoop	**Hadoop Distributed File System (HDFS)**
MapReduce	**NoSQL**

NoSQL Databases

- Not based on the relational model
- Support distributed database architectures
- Provide high scalability, high availability, and fault tolerance
- Support large amounts of sparse data
- Geared toward performance rather than transaction consistency
- Store data in key-value stores

Figure 2.5 - A Simple Key-Value Representation

FIGURE 2.5 A SIMPLE KEY-VALUE REPRESENTATION

Trucks-R-Us

Data stored using traditional relational model

DID	CERT1	CERT2	CERT3	DOB	LICTYPE
2732	80		95	1/24/1962	P
2946		92		4/11/1970	
3650	86			11/27/1963	R

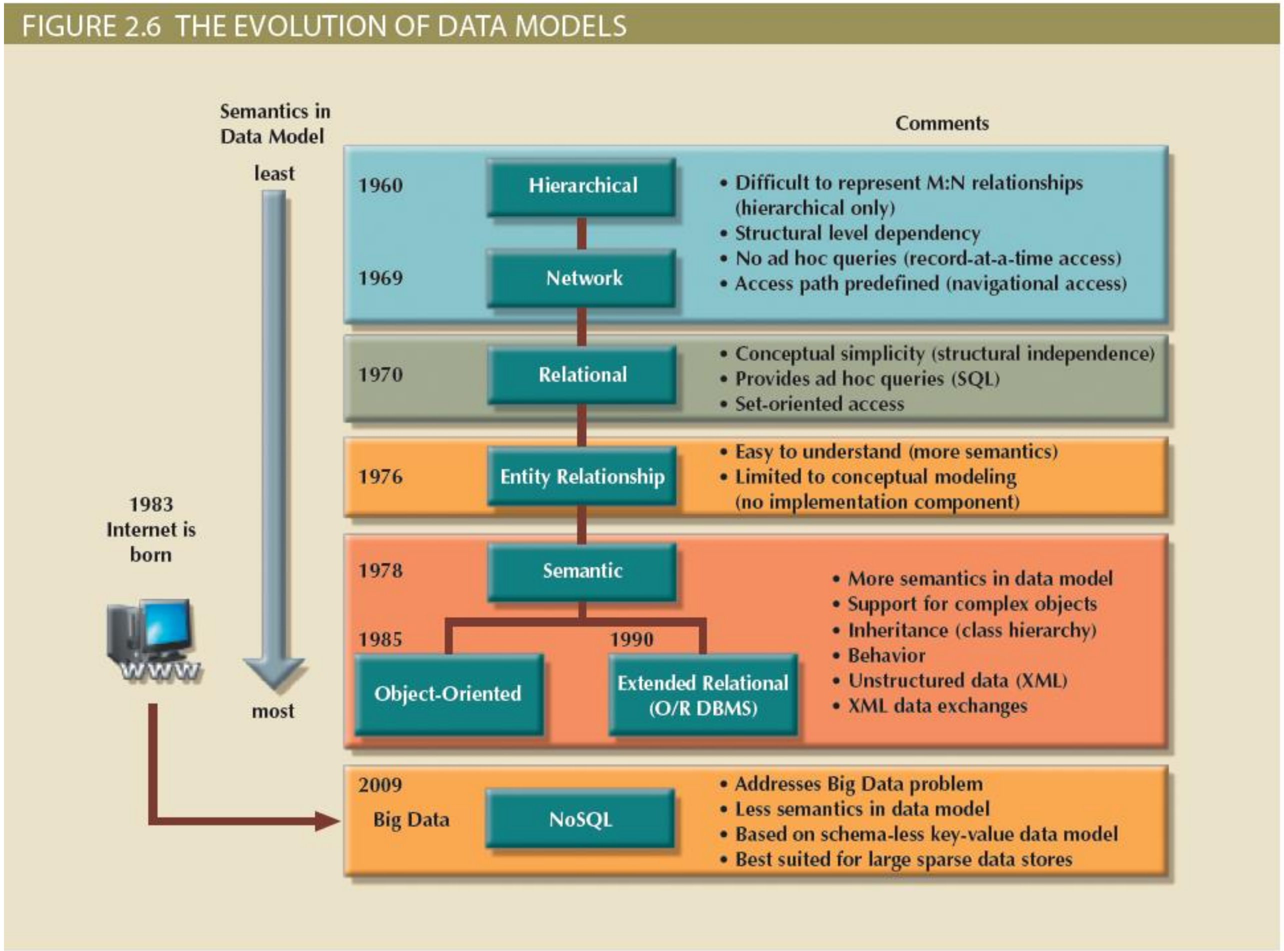
Data stored using key-value model

DID	KEY	VALUE
2732	CERT1	80
2732	CERT3	95
2732	DOB	1/24/1962
2732	LICTYPE	P
2946	CERT2	92
2946	DOB	4/11/1970
3650	CERT1	86
3650	DOB	11/27/63
3650	LICTYPE	R

Driver 2732

- **In the relational model:**
  - **Each row represents one entity instance.**
  - **Each column represents one attribute of the entity.**
  - **The values in a column are of the same data type.**
- **In the key-value model:**
  - **Each row represents one attribute/value of one entity instance.**
  - **The “key” column could represent any entity’s attribute.**
  - **The values in the “value” column could be of any data type and therefore it is generally assigned a long string data type.**

Figure 2.6 - The Evolution of Data Models



Hierarchical Model

Advantages	Disadvantages
- Promotes data sharing - Parent/child relationship promotes conceptual simplicity and data integrity - Database security is provided and enforced by DBMS - Efficient with 1:M relationships	- Requires knowledge of physical data storage characteristics - Navigational system requires knowledge of hierarchical path - Changes in structure require changes in all application programs - Implementation limitations - No data definition - Lack of standards

Network Model

Advantages	Disadvantages

Advantages

- Conceptual simplicity
- Handles more relationship types
- Data access is flexible
- Data owner/member relationship promotes data integrity
- Conformance to standards
- Includes data definition language (DDL) and data manipulation language (DML)

Disadvantages

- System complexity limits efficiency
- Navigational system yields complex implementation, application development, and management
- Structural changes require changes in all application programs

Relational Model

Advantages	Disadvantages
Structural independence is promoted using independent tables Tabular view improves conceptual simplicity Ad hoc query capability is based on SQL Isolates the end user from physical-level details Improves implementation and management simplicity	Requires substantial hardware and system software overhead Conceptual simplicity gives untrained people the tools to use a good system poorly May promote information problems

Advantages

- Structural independence is promoted using independent tables
- Tabular view improves conceptual simplicity
- Ad hoc query capability is based on SQL
- Isolates the end user from physical-level details
- Improves implementation and management simplicity

Disadvantages

- Requires substantial hardware and system software overhead
- Conceptual simplicity gives untrained people the tools to use a good system poorly
- May promote information problems

Entity Relationship Model

Advantages	Disadvantages
Visual modeling yields conceptual simplicity	Limited constraint representation
Visual representation makes it an effective communication tool	Limited relationship representation
Is integrated with the dominant relational model	No data manipulation language
	Loss of information content occurs when attributes are removed from entities to avoid crowded displays

Object-Oriented Model

Advantages	Disadvantages
Semantic content is added; Visual representation includes semantic content; Inheritance promotes data integrity	Slow development of standards caused vendors to supply their own enhancements (Compromised widely accepted standard); Complex navigational system; Learning curve is steep; High system overhead slows transactions

Advantages

- Semantic content is added
- Visual representation includes semantic content
- Inheritance promotes data integrity

Disadvantages

- Slow development of standards caused vendors to supply their own enhancements
  - Compromised widely accepted standard
- Complex navigational system
- Learning curve is steep
- High system overhead slows transactions

NoSQL

Advantages	Disadvantages
High scalability, availability, and fault tolerance are provided	Complex programming is required
Uses low-cost commodity hardware	There is no relationship support
Supports Big Data	There is no transaction integrity support
Key-value model improves storage efficiency	In terms of data consistency, it provides an eventually consistent model

Table 2.3 - Data Model Basic Terminology Comparison

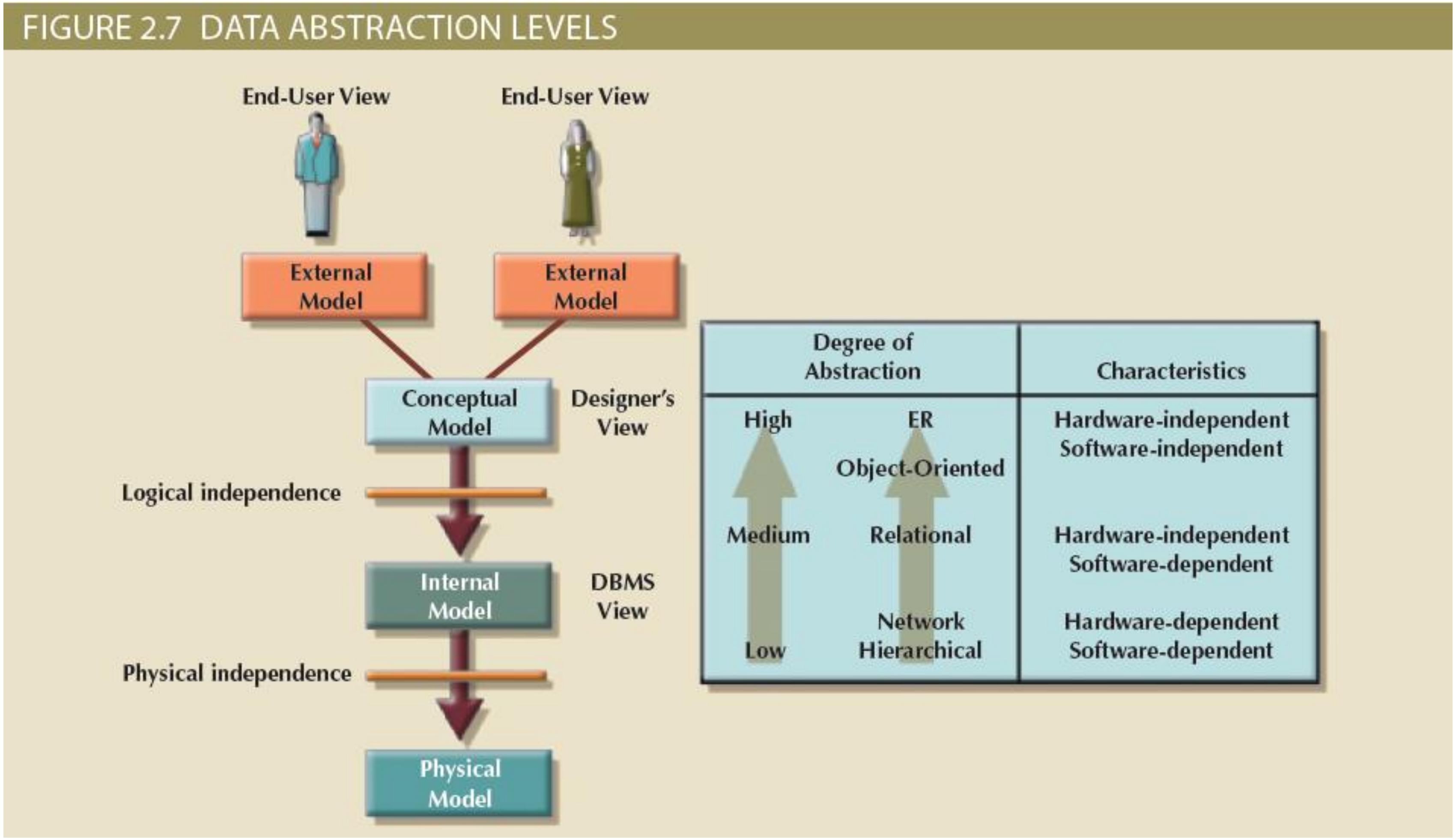
TABLE 2.3

DATA MODEL BASIC TERMINOLOGY COMPARISON

REAL WORLD	EXAMPLE	FILE PROCESSING	HIERARCHICAL MODEL	NETWORK MODEL	RELATIONAL MODEL	ER MODEL	OO MODEL
A group of vendors	Vendor file cabinet	File	Segment type	Record type	Table	Entity set	Class
A single vendor	Global supplies	Record	Segment occurrence	Current record	Row (tuple)	Entity occurrence	Object instance
The contact name	Johnny Ventura	Field	Segment field	Record field	Table attribute	Entity attribute	Object attribute
The vendor identifier	G12987	Index	Sequence field	Record key	Key	Entity identifier	Object identifier

Note: For additional information about the terms used in this table, consult the corresponding chapters and online appendixes that accompany this book. For example, if you want to know more about the OO model, refer to Appendix G, Object-Oriented Databases.

Figure 2.7 - Data Abstraction Levels

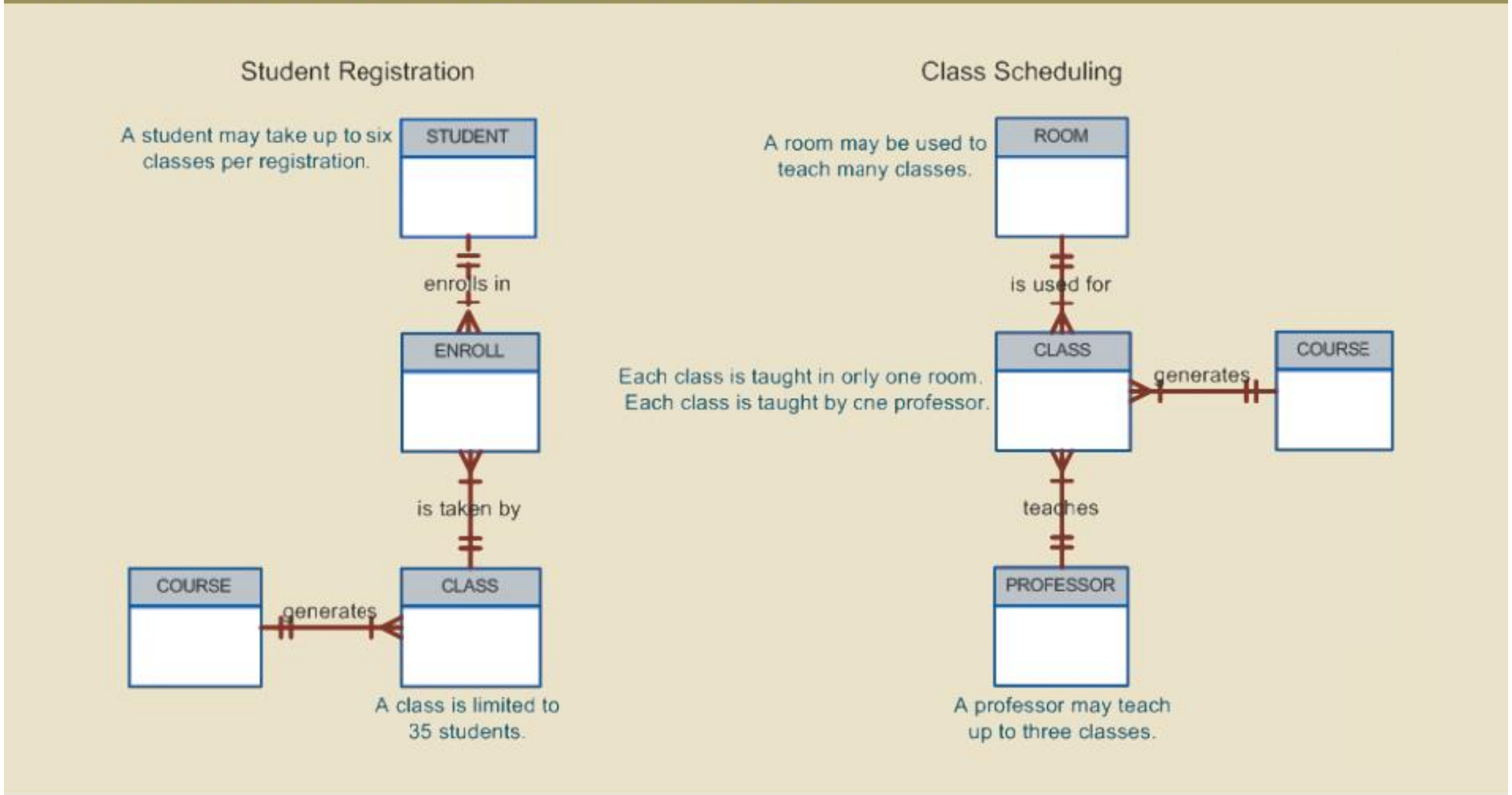


The External Model

- End users' view of the data environment
- ER diagrams are used to represent the external views
- **External schema**: Specific representation of an external view

Figure 2.8 - External Models For Tiny College

FIGURE 2.8 EXTERNAL MODELS FOR TINY COLLEGE

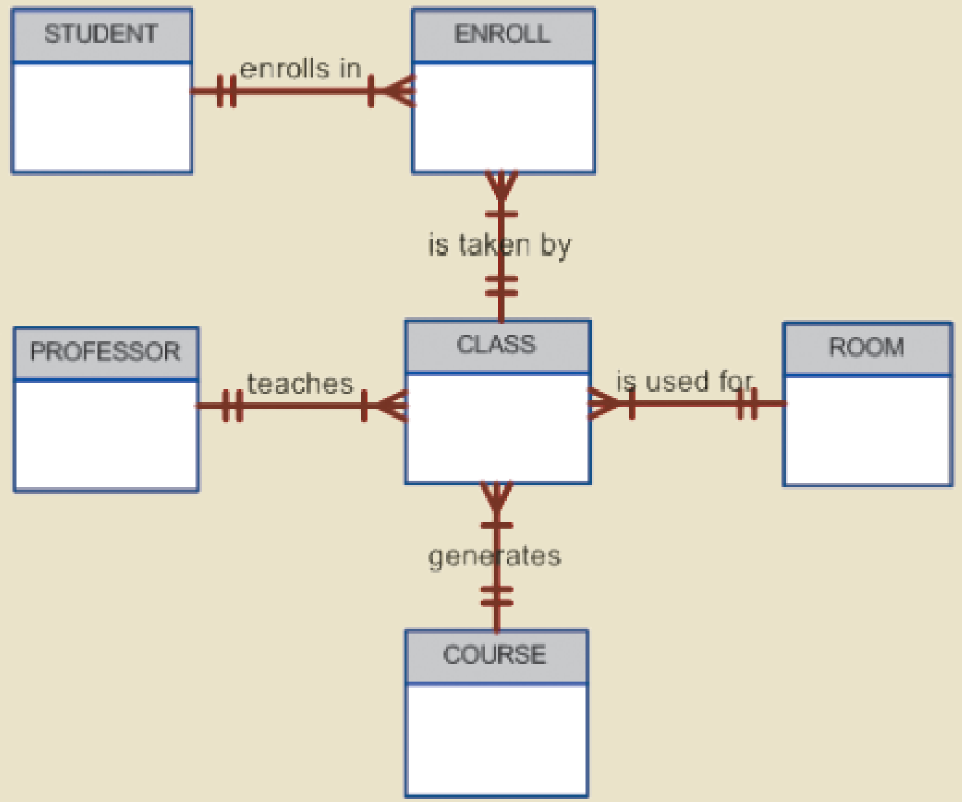


The Conceptual Model

- Represents a global view of the entire database by the entire organization
- **Conceptual schema**: Basis for the identification and high-level description of the main data objects
- Has a macro-level view of data environment
- Is software and hardware independent
- **Logical design**: Task of creating a conceptual data model

Figure 2.9 - Conceptual Model For Tiny College

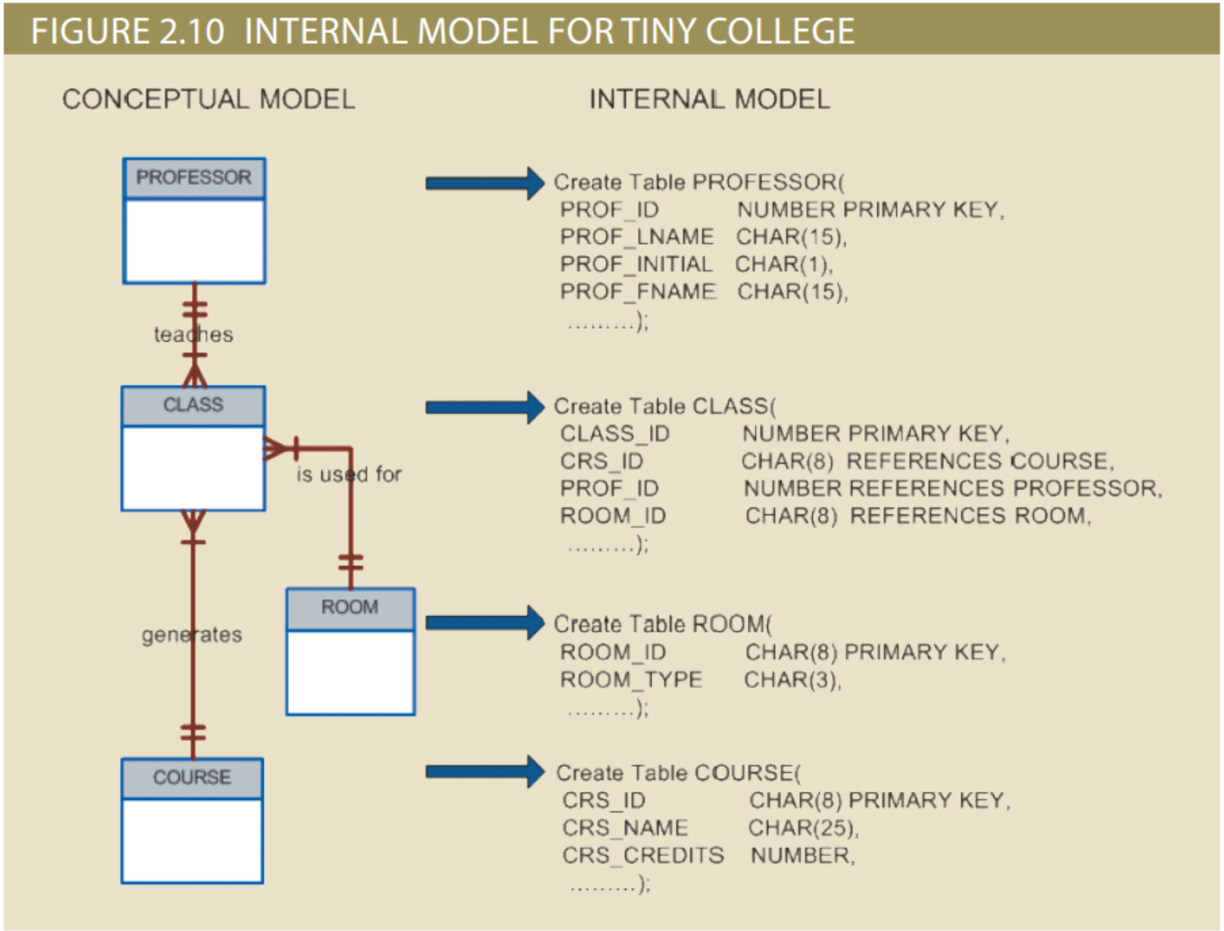
FIGURE 2.9 CONCEPTUAL MODEL FOR TINY COLLEGE



The Internal Model

- Representing database as seen by the DBMS mapping conceptual model to the DBMS
- **Internal schema**: Specific representation of an internal model
  - Uses the database constructs supported by the chosen database
- Is software dependent and hardware independent
- **Logical independence**: Changing internal model without affecting the conceptual model

Figure 2.10 - Internal Model for Tiny College



The Physical Model

- Operates at lowest level of abstraction
- Describes the way data are saved on storage media such as disks or tapes
- Requires the definition of physical storage and data access methods
- Relational model aimed at logical level
  - Does not require physical-level details
- **Physical independence**: Changes in physical model do not affect internal model

Table 2.4 - Levels of Data Abstraction

TABLE 2.4

LEVELS OF DATA ABSTRACTION

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