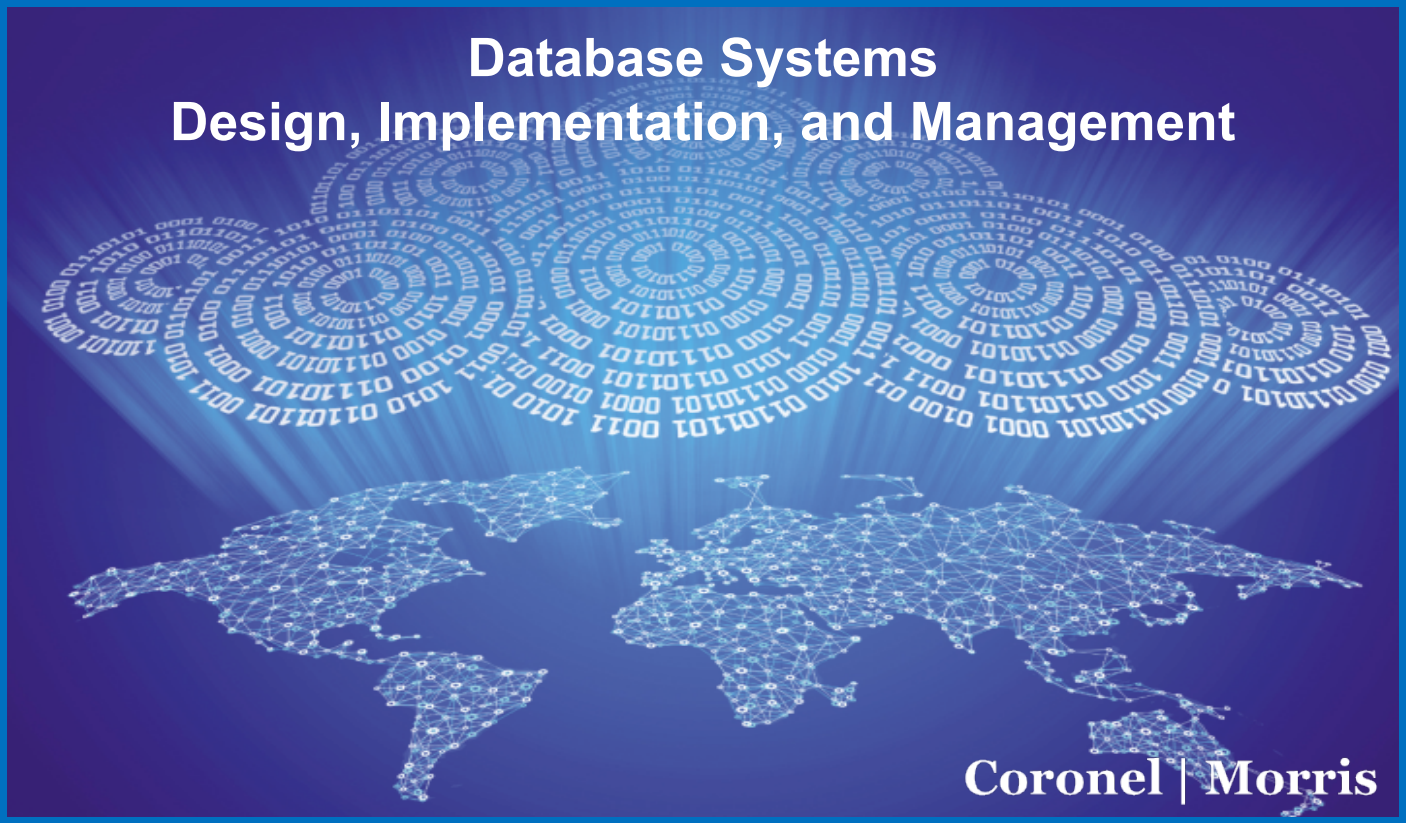
11e




# 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

# Software

- Windows
  - WAMP Server (Windows+Apache+MySQL+PHP)
  - Microsoft Access
- Mac OS
  - MAMP Server (Mac+Apache+MySQL+PHP)
  - Or configure MySQL & any DBM tools by yourself
- Linux (configure ALL the tools by yourself)

**Find the installation video on YouTube by yourself !**

# 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
    - A data model represents data structures and their characteristics, relations, constraints, transformations, and other constructs with the purpose of supporting a specific problem domain.
  - **Model** - Abstraction of a real-world object or event

# Data Model

- An implementation-ready data model should contain at least the following components:
  - A description of the data structure that will store the end-user data.
  - A set of enforceable rules to guarantee the integrity of the data.
  - A data manipulation methodology to support the real-world data transformations.

# 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
  - A person, place, thing or event about which data will be collected and stored
  - **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

# 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

# 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?

    ex: How many classes can one student enroll? Many Classes
  - How many instances of A are related to one instance of B?

    ex: How many students can enroll in one class? Many students

# 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

- Manage large amounts of data for complex manufacturing projects
- Represented by an upside-down tree which contains segments
  - **Segments**: Equivalent of a file system's record type
- Depicts a set of one-to-many (1:M) relationships

## Network Models

- Represent complex data relationships
- Improve database performance and impose a database standard
- Depicts both one-to-many (1:M) and many-to-many (M:N) relationships

# Hierarchical Model

| Advantages | Disadvantages |
| --- | --- |
| ▪ Promotes data sharing<br>▪ Parent/child relationship promotes conceptual simplicity and data integrity<br>▪ Database security is provided and enforced by DBMS<br>▪ Efficient with 1:M relationships | ▪ Requires knowledge of physical data storage characteristics<br>▪ Navigational system requires knowledge of hierarchical path<br>▪ Changes in structure require changes in all application programs<br>▪ Implementation limitations<br>▪ Lack of standards |

# Network Model

| Advantages | Disadvantages |
| --- | --- |
| ▪ Conceptual simplicity<br>▪ Handles more relationship types<br>▪ Data access is flexible<br>▪ Data owner/member relationship promotes data integrity<br>▪ Conformance to standards<br>▪ Includes data definition language (DDL) and data manipulation language (DML) | ▪ System complexity limits efficiency<br>▪ Navigational system yields complex implementation, application development, and management<br>▪ Structural changes require changes in all application programs |

# 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
- SELECT,UPDATE, INSERT

## Schema data definition language (DDL)

- Enables the database administrator to define the schema components
- CREATE, DROP, RENAME, TRUNCATE

# The Relational Model

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

# The Relational Model (continued)

- Based on a relation
  - **Relation** or **table**: Matrix composed of intersecting tuples (rows) and attributes (columns)
    - Related to each other through sharing a common entity characteristic
- Describes a precise set of data manipulation constructs
- 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)**

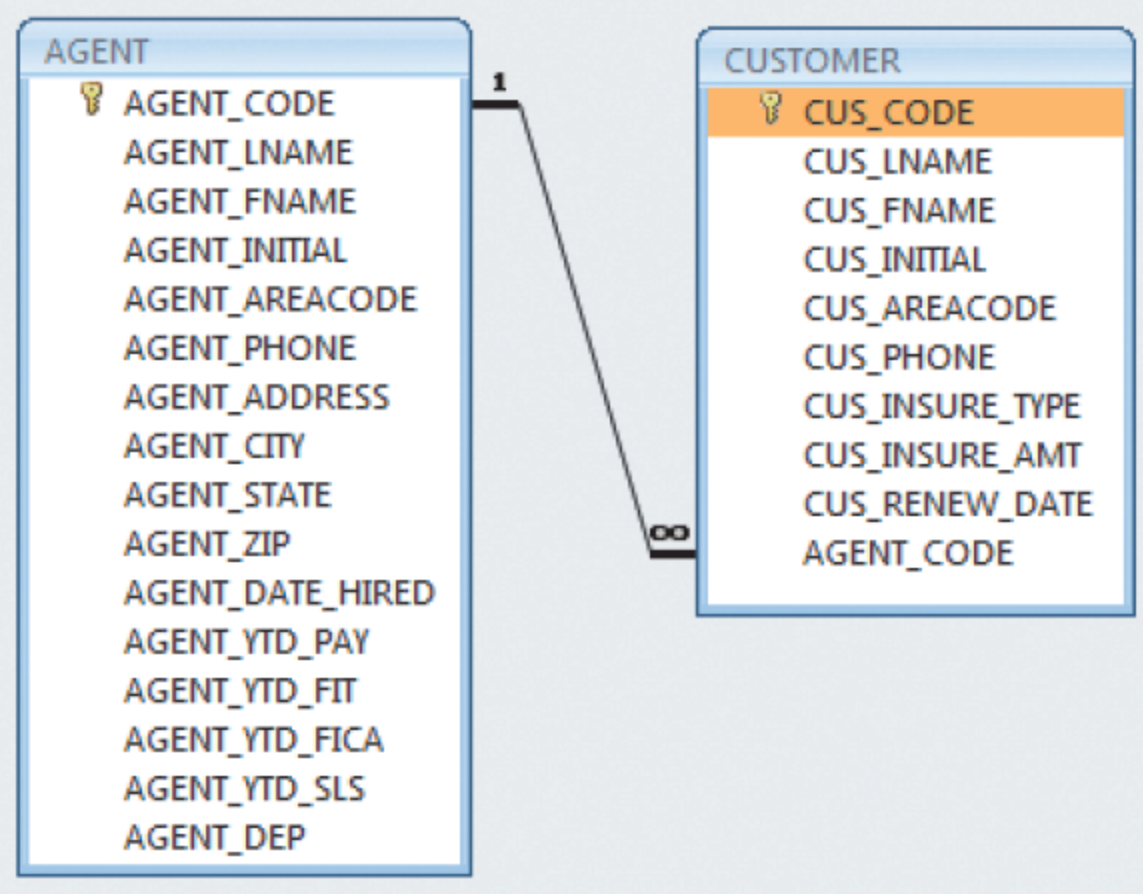
| AGENT_CODE | AGENT_LNAME | AGENT_FNAME | AGENT_INITIAL | AGENT_AREACODE | AGENT_PHONE |
|---|---|---|---|---|---|
| 501 | Alby | Alex | B | 713 | 228-1249 |
| 502 | Hahn | Leah | F | 615 | 882-1244 |
| 503 | Okon | John | T | 615 | 123-5589 |

**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 |

# Figure 2.2 - A Relational Diagram




Cengage Learning © 2015

# Relational Model

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

# 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

# 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

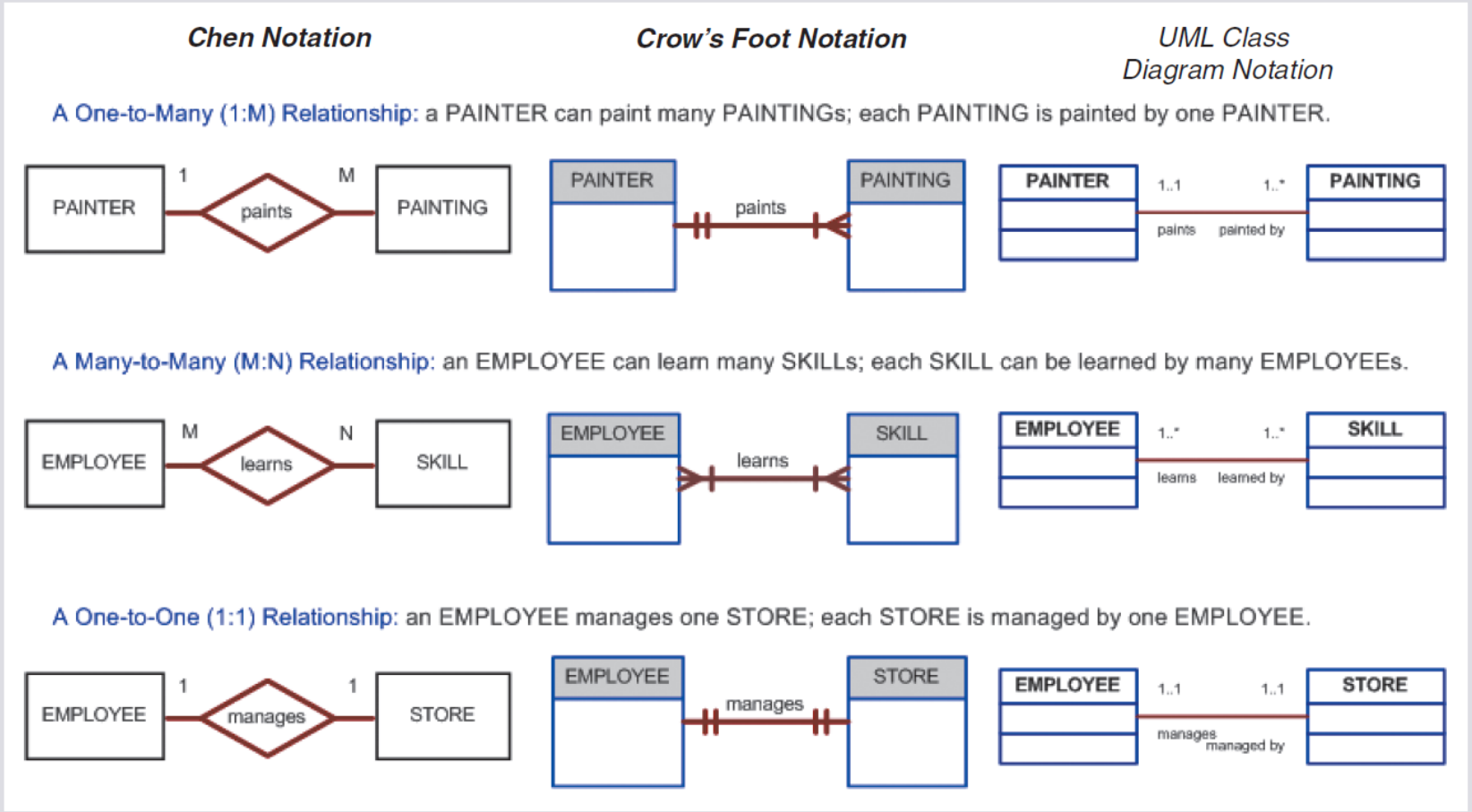
# The Entity Relationship Model

- 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

# Figure 2.3 - The ER Model Notations



# Entity Relationship Model

| Advantages | Disadvantages |
| --- | --- |
| ▪ Visual modeling yields conceptual simplicity<br>▪ Visual representation makes it an effective communication tool<br>▪ Is integrated with the dominant relational model | ▪ Limited constraint representation<br>▪ Limited relationship representation<br>▪ Loss of information content occurs when attributes are removed from entities to avoid crowded displays |

# 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

# Object-Oriented Model

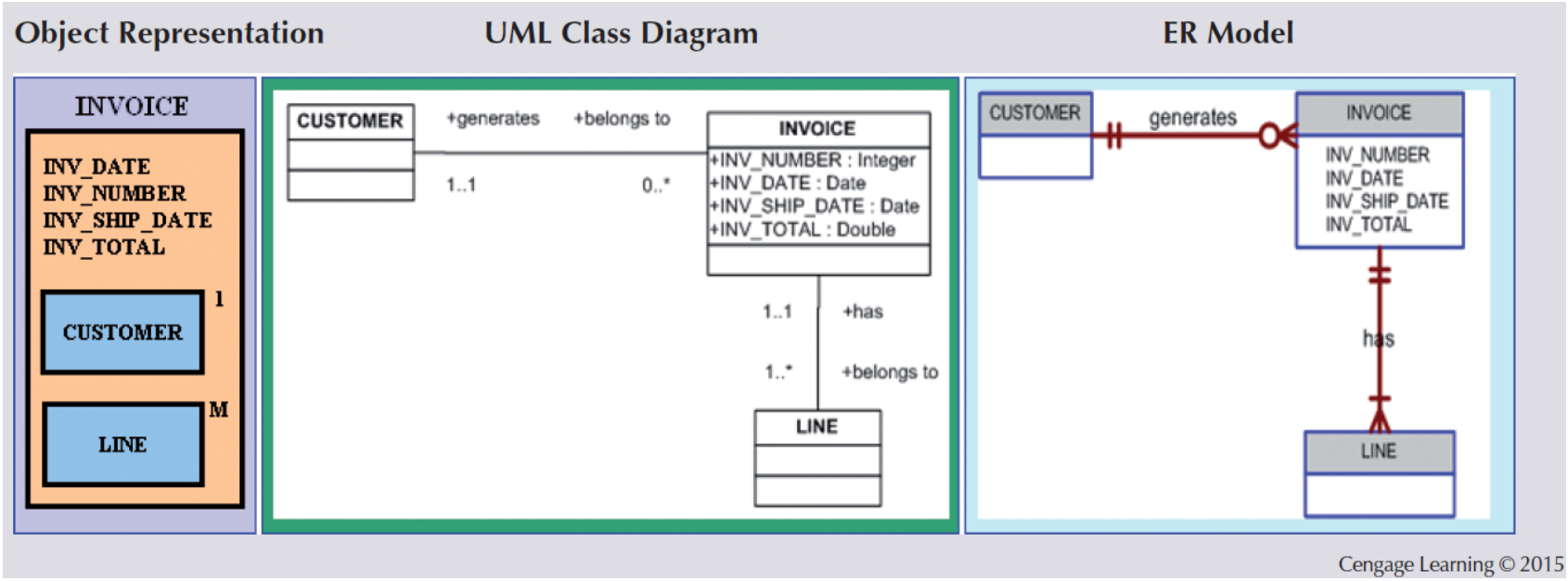
**Advantages**

- 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

# 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

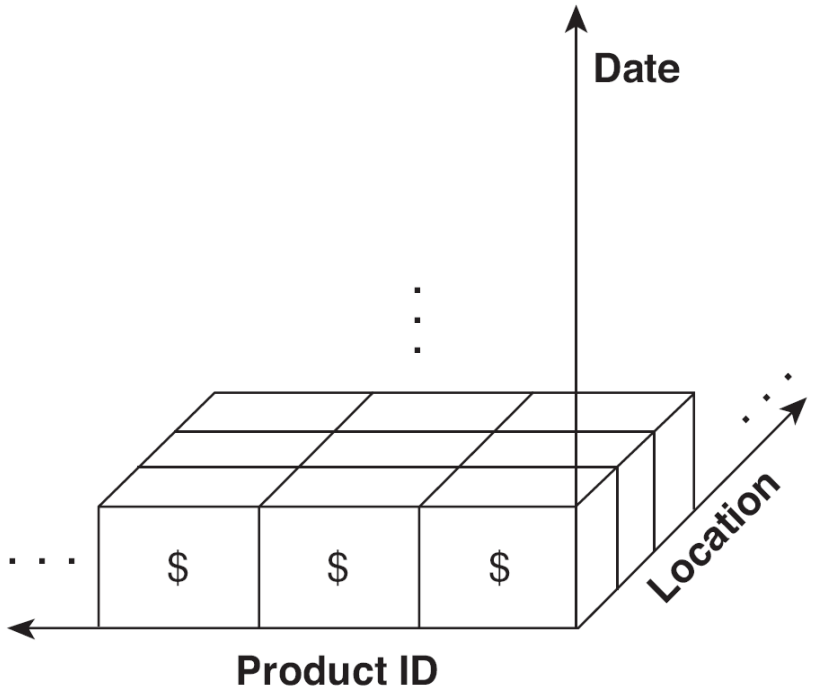
- Volume does not allow the usage of conventional structures
- Expensive
- OLAP tools proved inconsistent dealing with unstructured data

# OLAP

- On-Line Analytical Processing (OLAP) was proposed by E. F. Codd, the father of the relational database.
- Relational databases put data into tables, while OLAP uses a multidimensional array representation.
  - Such representations of data previously existed in statistics and other fields
- There are a number of data analysis and data exploration operations that are easier with such a data representation.

# OLAP

- Consider a data set that records the sales of products at a number of company stores at various dates.
- This data can be represented as a 3 dimensional array
- There are 3 two-dimensional aggregates (3 choose 2 ), 3 one-dimensional aggregates, and 1 zero-dimensional aggregate (the overall total)



# Big Data New Technologies

**Hadoop**

**Hadoop Distributed File System (HDFS)**

**MapReduce**

**NoSQL**

# NoSQL Databases

- We use NoSQL database to……
  - Search for a product on Amazon
  - Send Messages to friends in Facebook
  - Watch a video in YouTube
  - Search for directions in Google Maps

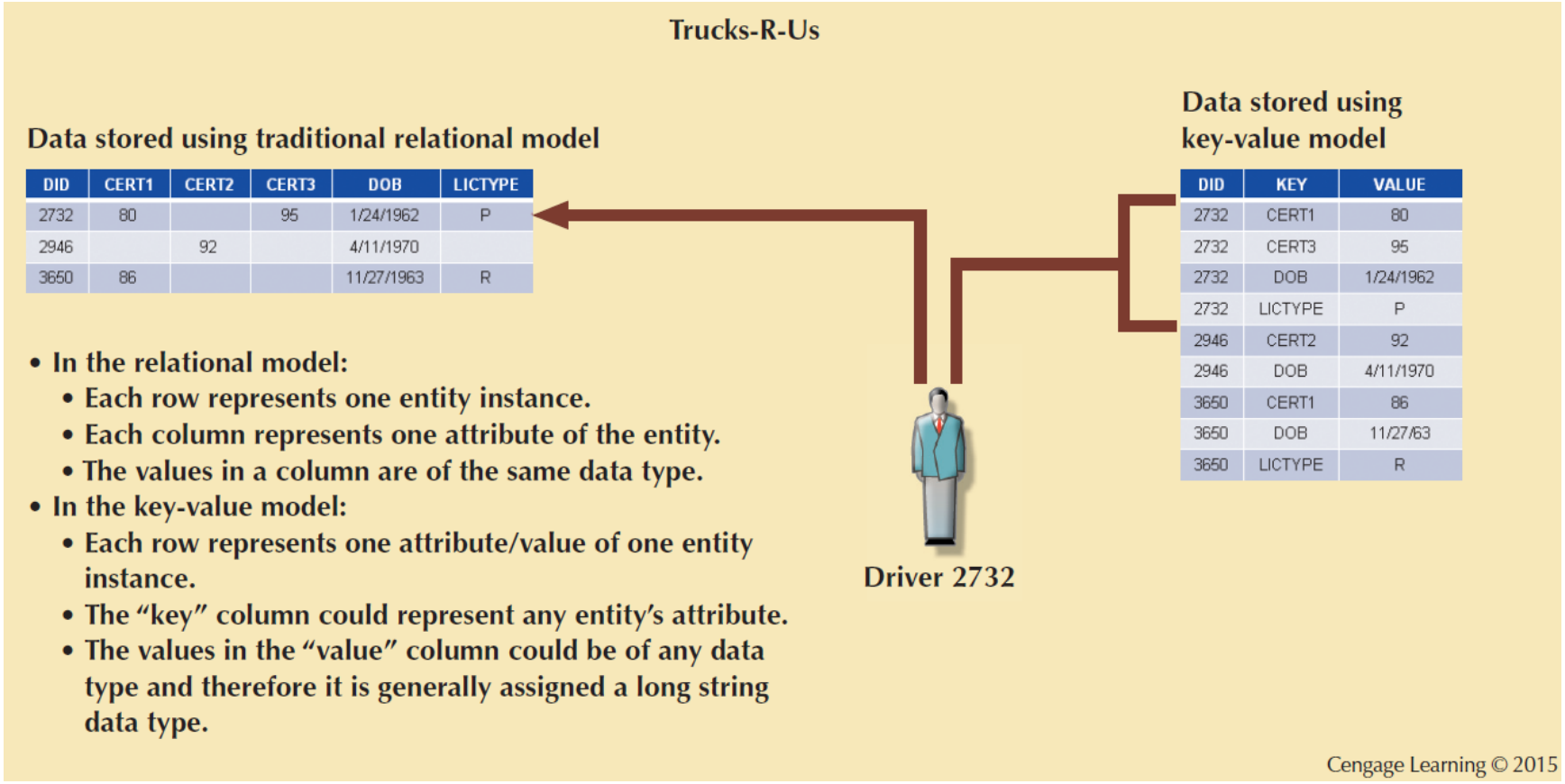
# NoSQL Databases

- 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:
  - Key-value data model is based on a structure composed of two data elements: a key and a value, in which every key has a corresponding value or set of values.

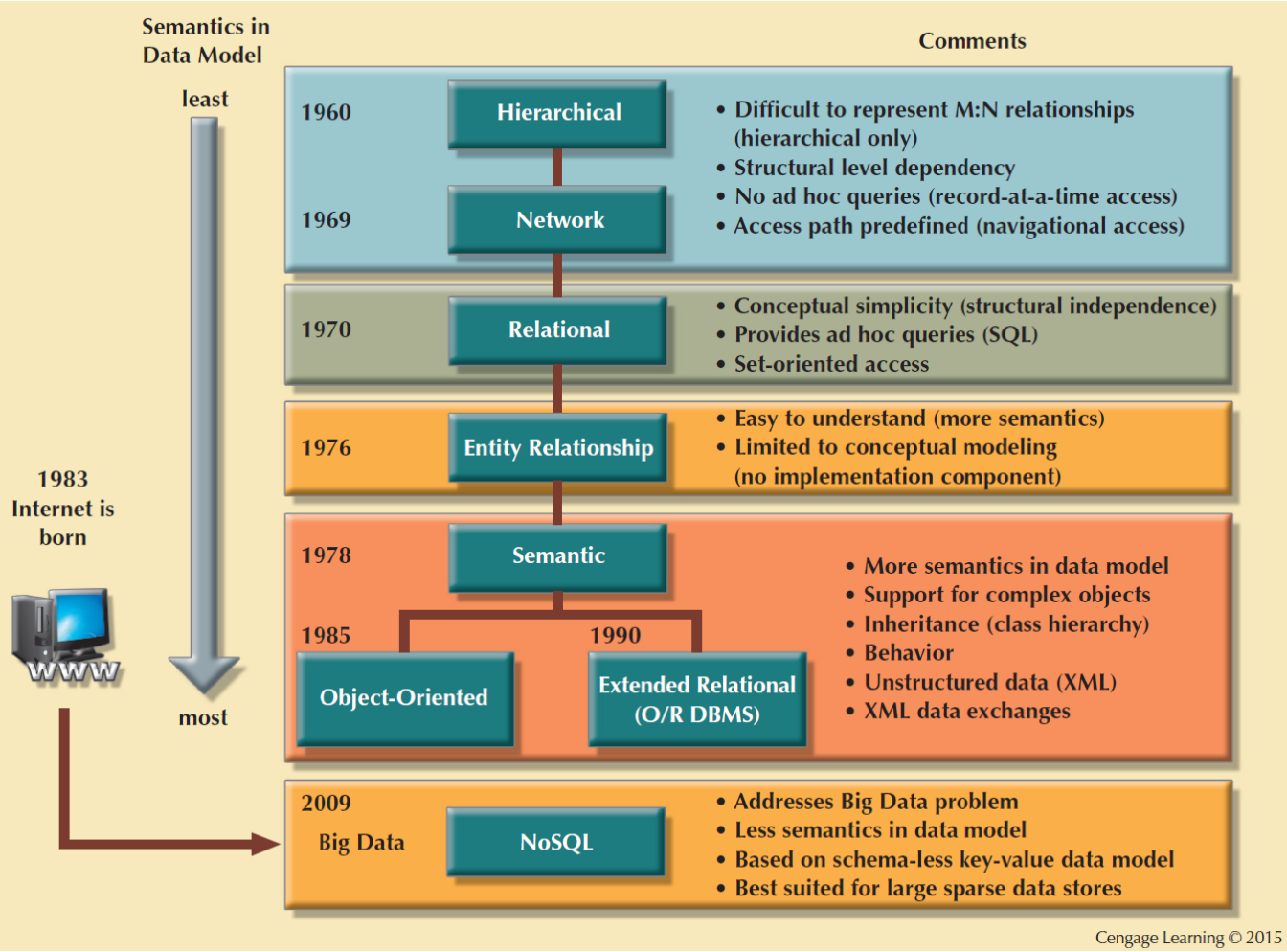
# NoSQL

| Advantages | Disadvantages |
| --- | --- |
| ▪ High scalability, availability, and fault tolerance are provided<br>▪ Uses low-cost commodity hardware<br>▪ Supports Big Data<br>▪ Key-value model improves storage efficiency | ▪ Complex programming is required<br>▪ There is no relationship support<br>▪ There is no transaction integrity support |

# Figure 2.5 - A Simple Key-value Representation



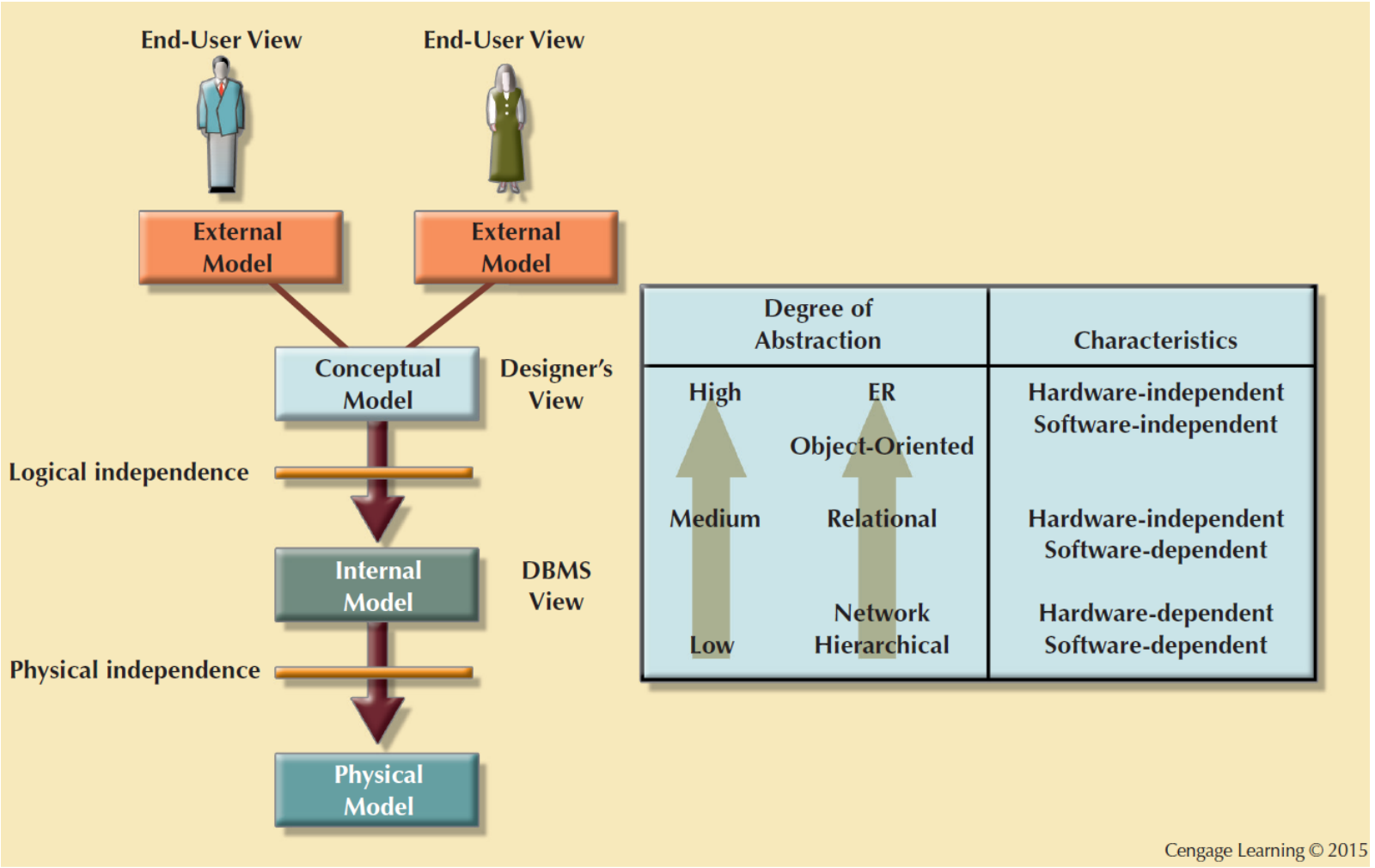
# Figure 2.6 - The Evolution of Data Models



# Advantages and disadvantages

| DATA MODEL | DATA INDEPENDENCE | STRUCTURAL INDEPENDENCE | ADVANTAGES | DISADVANTAGES |
|---|---|---|---|---|
| Hierarchical | Yes | No | 1. It promotes data sharing.<br>2. Parent/child relationship promotes conceptual simplicity.<br>3. Database security is provided and enforced by DBMS.<br>4. Parent/child relationship promotes data integrity.<br>5. It is efficient with 1:M relationships. | 1. Complex implementation requires knowledge of physical data storage characteristics.<br>2. Navigational system yields complex application development, management, and use; requires knowledge of hierarchical path.<br>3. Changes in structure require changes in all application programs.<br>4. There are implementation limitations (no multiparent or M:N relationships).<br>5. There is no data definition or data manipulation language in the DBMS.<br>6. There is a lack of standards. |
| Network | Yes | No | 1. Conceptual simplicity is at least equal to that of the hierarchical model.<br>2. It handles more relationship types, such as M:N and multiparent.<br>3. Data access is more flexible than in hierarchical and file system models.<br>4. Data owner/member relationship promotes data integrity.<br>5. There is conformance to standards.<br>6. It includes data definition language (DDL) and data manipulation language (DML) in DBMS. | 1. System complexity limits efficiency—still a navigational system.<br>2. Navigational system yields complex implementation, application development, and management.<br>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.<br>2. Tabular view substantially improves conceptual simplicity, thereby promoting easier database design, implementation, management, and use.<br>3. Ad hoc query capability is based on SQL.<br>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.<br>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.<br>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.<br>2. Visual representation makes it an effective communication tool.<br>3. It is integrated with the dominant relational model. | 1. There is limited constraint representation.<br>2. There is limited relationship representation.<br>3. There is no data manipulation language.<br>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.<br>2. Visual representation includes semantic content.<br>3. Inheritance promotes data integrity. | 1. Slow development of standards caused vendors to supply their own enhancements, thus eliminating a widely accepted standard.<br>2. It is a complex navigational system.<br>3. There is a steep learning curve.<br>4. High system overhead slows transactions. |
| NoSQL | Yes | Yes | 1. High scalability, availability, and fault tolerance are provided.<br>2. It uses low-cost commodity hardware.<br>3. It supports Big Data.<br>4. Key-value model improves storage efficiency. | 1. Complex programming is required.<br>2. There is no relationship support—only by application code.<br>3. There is no transaction integrity support.<br>4. In terms of data consistency, it provides an eventually consistent model. |

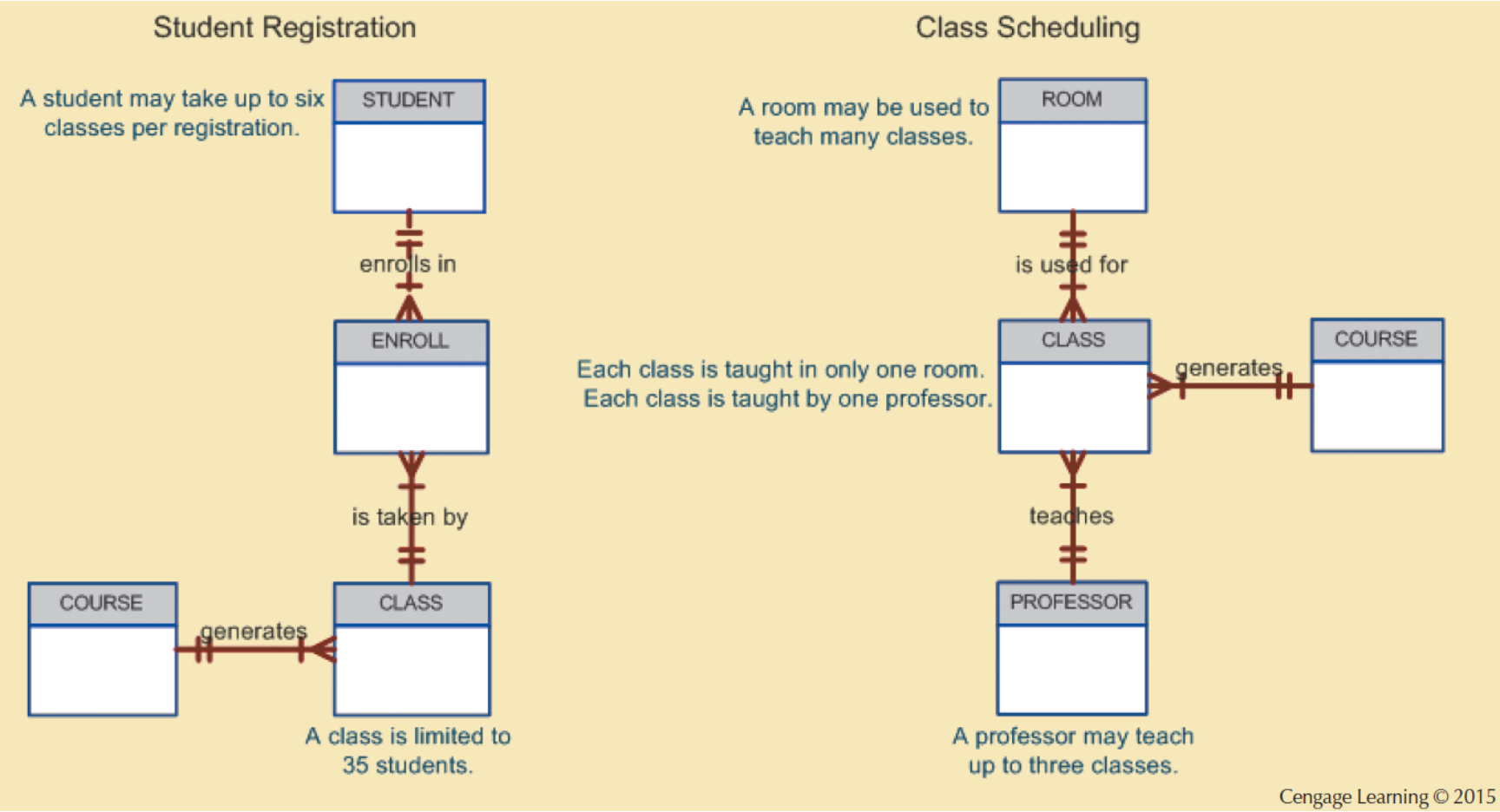
# Figure 2.7 - Data Abstraction Levels



# 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

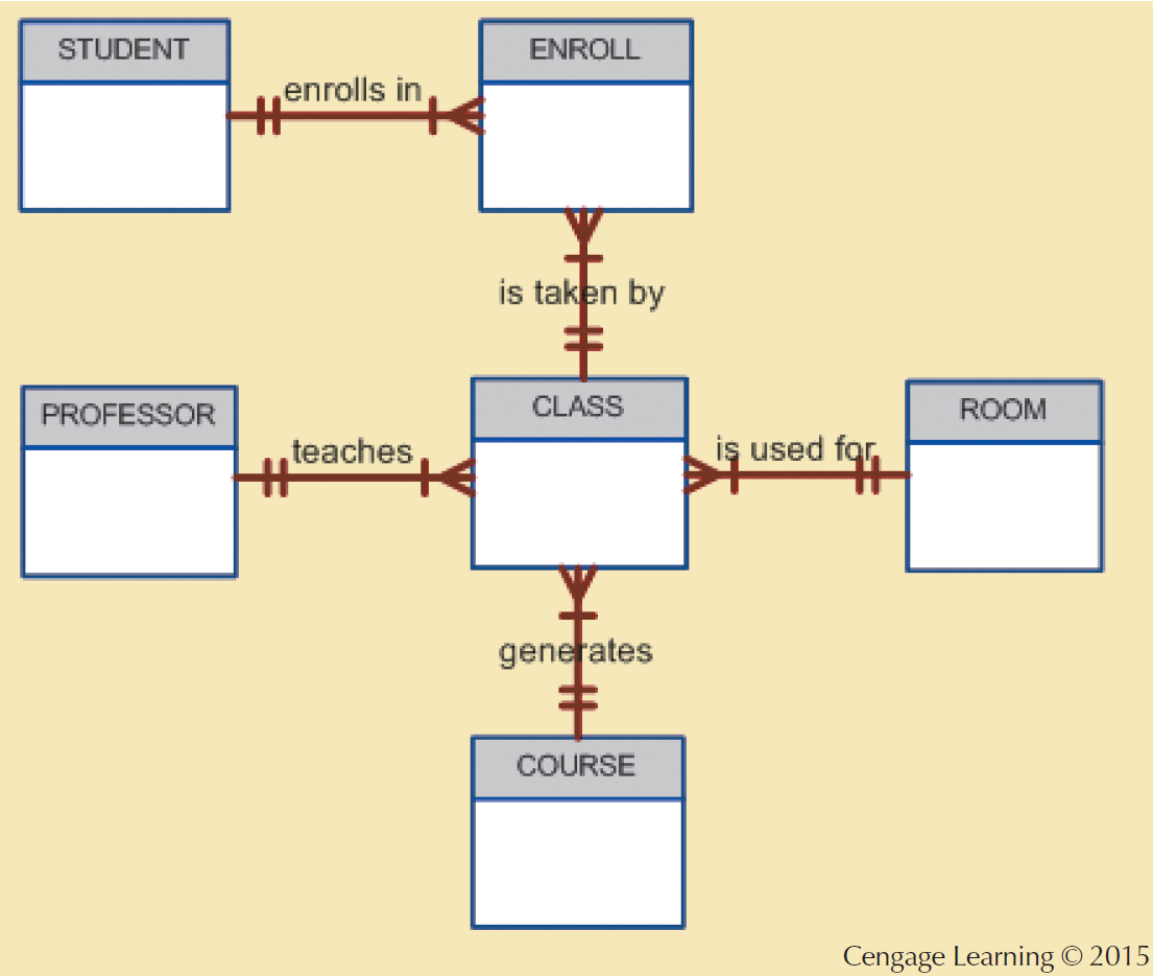
# Figure 2.8 - External Models for Tiny College



# 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(bird's eye)-level view of data environment
- Is software and hardware independent
- **Logical design**: Task of creating a conceptual data model

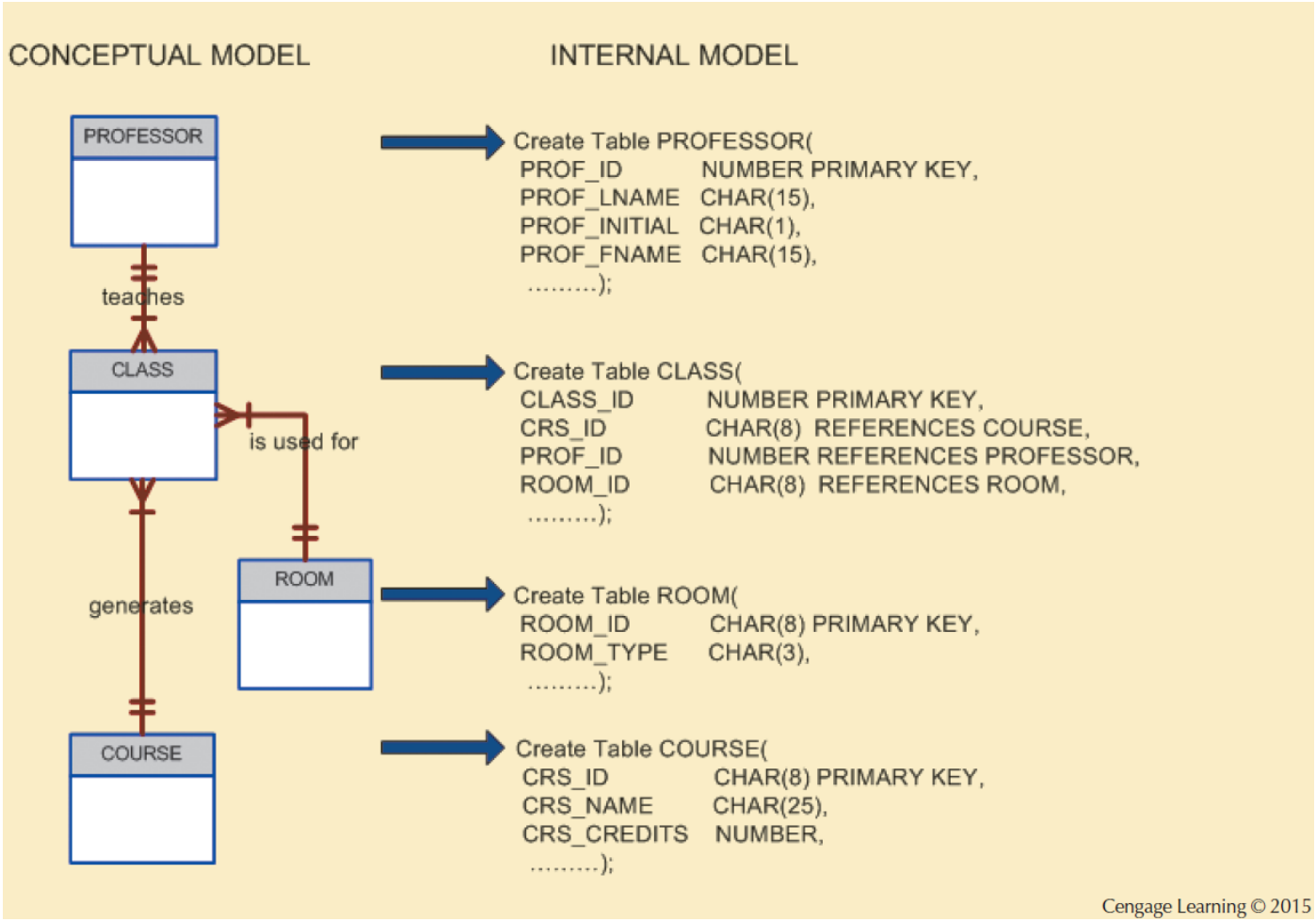
# Figure 2.9 - Conceptual Model for Tiny College



# 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

| 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 |

Cengage Learning © 2015

# 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