Database Principles: Fundamentals of Design, Implementation, and Management

Tenth Edition

Chapter 3

Data Models

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
- How the major data models evolved

Objectives (cont'd.)

- About emerging alternative data models and the need they fulfill
- How data models can be classified by their level of abstraction

Introduction

- Designers, programmers, and end users see data in different ways
- Different views of same data lead to designs that do not reflect organization's operation
- Data modeling reduces complexities of database design
- Various degrees of data abstraction help reconcile varying views of same data

Data Modeling and Data Models

- Data models
  - Relatively simple representations of complex real-world data structures
    - Often graphical
- Model: an abstraction of a real-world object or event
  - Useful in understanding complexities of the real-world environment
- Data modeling is iterative and progressive

The Importance of Data Models

- Facilitate interaction among the designer, the applications programmer, and the end user
- End users have different views and needs for data
- Data model organizes data for various users
- Data model is an abstraction
  - Cannot draw required data out of the data model

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

- Descriptions of policies, procedures, or principles within a specific organization
  - Apply to any organization that stores and uses data to generate information
- Description of operations to create/enforce actions within an organization's environment
  - Must be in writing and kept up to date
  - Must be easy to understand and widely disseminated
- Describe characteristics of 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

Discovering Business Rules (cont'd.)

- Standardize company's view of data
- 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

Translating Business Rules into Data Model Components

- Nouns translate into entities
- Verbs translate into relationships among entities
- Relationships are bidirectional
- Two 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

- Naming occurs during translation of business rules to data model components
- Names should make the object unique and distinguishable from other objects
- Names should also be descriptive of objects in the environment and be familiar to users
- Proper naming:
  - Facilitates communication between parties
  - Promotes self-documentation

The Evolution of Data Models

TABLE 3.1 Evolution of Major Data Models

GENERATION	TIME	DATA MODEL	EXAMPLES	COMMENTS
First	1960s–1970s	File system	VMS/VSAM	Used mainly on IBM mainframe systems Managed records, not relationships
Second	1970s	Hierarchical and network	IMS, ADABAS, IDS-II	Early database systems Navigational access
Third	Mid-1970s	Relational	DB2 Oracle MS SQL Server MySQL	Conceptual simplicity Entity relationship (ER) modeling and support for relational data modeling
Fourth	Mid-1980s	Object-oriented Object/ relational (O/R)	Versant Objectivity/DB DB2 UDB Oracle 11g	Object/relational supports object data types Star Schema support for data warehousing Web databases become common
Fifth	Mid-1990s	XML Hybrid DBMS	dbXML Tamino DB2 UDB Oracle 11g MS SQL Server	Unstructured data support O/R model supports XML documents Hybrid DBMS adds object front end to relational databases Support large databases (terabyte size)
Emerging Models: NoSQL	Late 2000s to present	Key-value store Column store	SimpleDB (Amazon) BigTable (Google) Cassandra (Apache)	Distributed, highly scalable High performance, fault tolerant Very large storage (petabytes) Suited for sparse data Proprietary API

Hierarchical and Network Models

- The hierarchical model
  - Developed in the 1960s to manage large amounts of data for manufacturing projects
  - Basic logical structure is represented by an upside-down “tree”
  - Structure contains levels or segments

Hierarchical and Network Models (cont'd.)

- Network model
  - Created to represent complex data relationships more effectively than the hierarchical model
  - Improves database performance
  - Imposes a database standard
  - Resembles hierarchical model
    - Record may have more than one parent

Hierarchical and Network Models (cont'd.)

  - Collection of records in 1:M relationships
  - Set composed of two record types:
    - Owner
    - Member
- Network model concepts still used today:
  - Schema
    - Conceptual organization of entire database as viewed by the database administrator
  - Subschema
    - Database portion "seen" by the application programs

Hierarchical and Network Models (cont'd.)

- Data management language (DML)
  - Defines the environment in which data can be managed
- Data definition language (DDL)
  - Enables the administrator to define the schema components

The Relational Model

- Developed by E.F. Codd (IBM) in 1970
- Table (relations)
  - Matrix consisting of row/column intersections
  - Each row in a relation is called a tuple
- Relational models were considered impractical in 1970
- Model was conceptually simple at expense of computer overhead

The Relational Model (cont'd.)

- Relational data management system (RDBMS)
  - Performs same functions provided by hierarchical model
  - Hides complexity from the user
- Relational diagram
  - Representation of entities, attributes, and relationships
- Relational table stores collection of related entities

FIGURE 3.1 Linking relational tables

Table name: AGENT (first six attributes)

Database name: Ch03_InsureCo

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

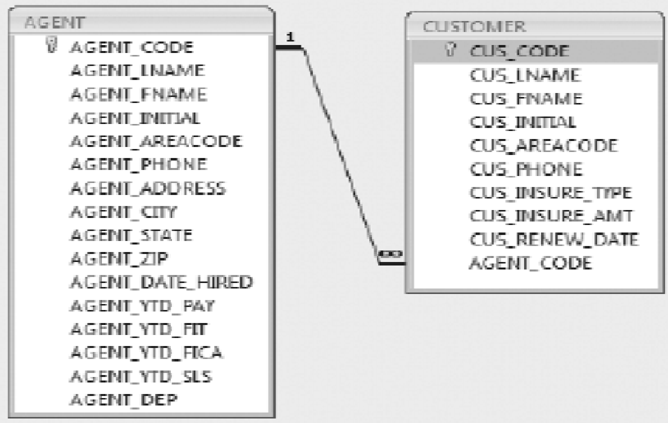
Link through AGENT_CODE

Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_INSURE_TYPE	CUS_INSURE_AMT	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	T1	100.00	05-Apr-2012	502
10011	Dunne	Leona	K	713	894-1238	T1	250.00	16-Jun-2012	501
10012	Smith	Kathy	W	615	894-2285	S2	150.00	29-Jan-2013	502
10013	Olowski	Paul	F	615	894-2180	S1	300.00	14-Oct-2012	502
10014	Orlando	Myron		615	222-1672	T1	100.00	28-Dec-2013	501
10015	O'Brian	Amy	B	713	442-3381	T2	850.00	22-Sep-2012	503
10016	Brown	James	G	615	297-1228	S1	120.00	25-Mar-2013	502
10017	Williams	George		615	290-2556	S1	250.00	17-Jul-2012	503
10018	Farriss	Anne	G	713	382-7185	T2	100.00	03-Dec-2012	501
10019	Smith	Olette	K	615	297-3809	S2	500.00	14-Mar-2013	503

SOURCE: Course Technology/Cengage Learning

FIGURE 3.2 A relational diagram



SOURCE: Course Technology/Cengage Learning

The Relational Model (cont'd.)

- SQL-based relational database application involves three parts:
  - End-user interface
    - Allows end user to interact with the data
  - Set 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

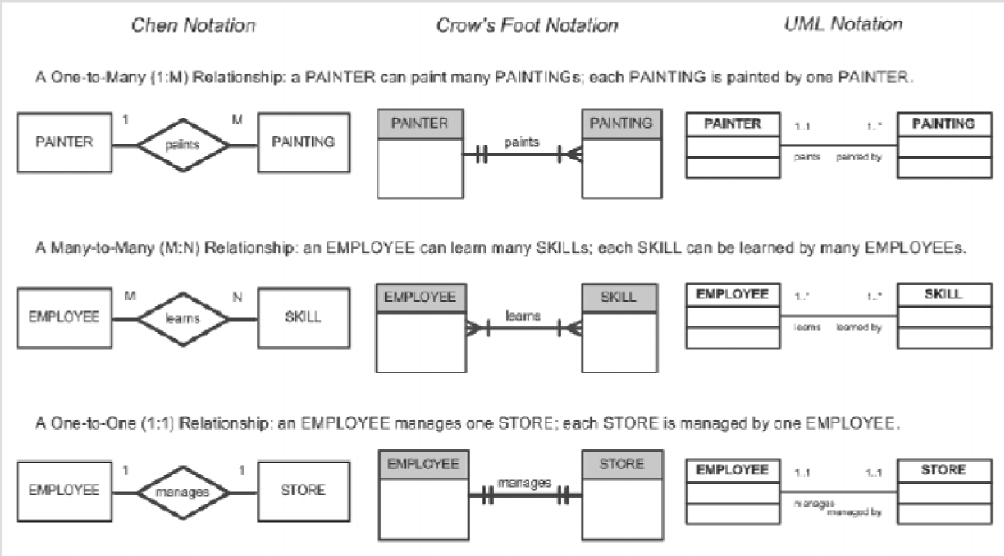
The Entity Relationship Model

- Widely accepted standard for data modeling
- Introduced by Chen in 1976
- Graphical representation of entities and their relationships in a database structure
- Entity relationship diagram (ERD)
  - Uses graphic representations to model database components
  - Entity is mapped to a relational table

The Entity Relationship Model (cont'd.)

- Entity instance (or occurrence) is row in table
- Entity set is collection of like entities
- Connectivity labels types of relationships
- Relationships are expressed using Chen notation
  - Relationships are represented by a diamond
  - Relationship name is written inside the diamond
- Crow's Foot notation used as design standard in this book

FIGURE 3.3 The ER model notations



SOURCE: Course Technology/Cengage Learning

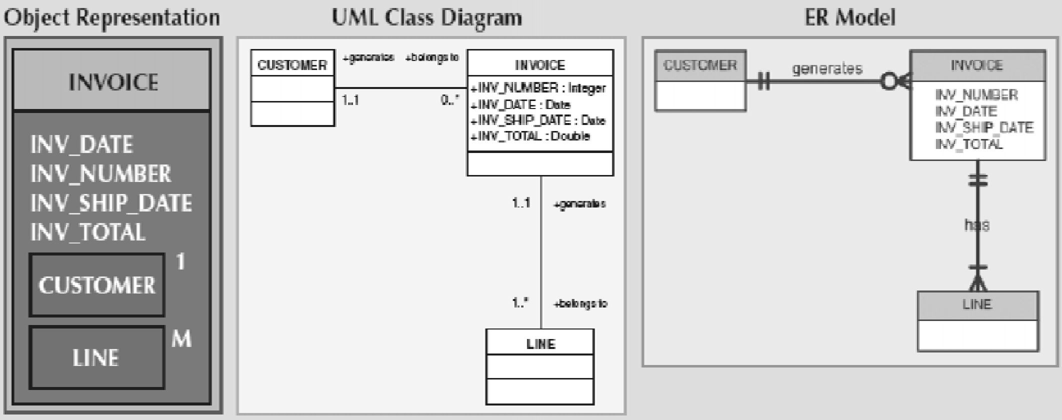
The Object-Oriented (OO) Model

- Data and relationships are contained in a single structure known as an object
- OODM (object-oriented data model) is the basis for OODBMS
  – Semantic data model
- An object:
  – Contains operations
  – Are self-contained: a basic building-block for autonomous structures
  – Is an abstraction of a real-world entity

The Object-Oriented (OO) Model (cont'd.)

- Attributes describe the properties of an object
- Objects that share similar characteristics are grouped in classes
- Classes are organized in a class hierarchy
- Inheritance: object inherits methods and attributes of parent class
- UML based on OO concepts that describe diagrams and symbols
  - Used to graphically model a system

FIGURE 3.4 A comparison of OO, UML, and ER models



SOURCE: Course Technology/Cengage Learning

Object/Relational and XML

- Extended relational data model (ERDM)
  - Semantic data model developed in response to increasing complexity of applications
  - Includes many of OO model's best features
  - Often described as an object/relational database management system (O/RDBMS)
  - Primarily geared to business applications

Object/Relational and XML (cont'd.)

- The Internet revolution created the potential to exchange critical business information
- In this environment, Extensible Markup Language (XML) emerged as the de facto standard
- Current databases support XML
  - XML: the standard protocol for data exchange among systems and Internet services

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition.
May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.

Emerging Data Models: Big Data and NoSQL

- Big Data
  - Find new and better ways to manage large amounts of Web-generated data and derive business insight from it
  - Simultaneously provides high performance and scalability at a reasonable cost
  - Relational approach does not always match the needs of organizations with Big Data challenges

Emerging Data Models: Big Data and NoSQL (cont'd.)

- NoSQL databases
  - Not based on the relational model, hence the name NoSQL
  - Supports distributed database architectures
  - Provides high scalability, high availability, and fault tolerance
  - Supports very large amounts of sparse data
  - Geared toward performance rather than transaction consistency

Emerging Data Models: Big Data and NoSQL (cont'd.)

- Key-value data model
  - Two data elements: key and value
    - Every key has a corresponding value or set of values
- Sparse data
  - Number of attributes is very large
  - Number of actual data instances is low
- Eventual consistency
  - Updates will propagate through system; eventually all data copies will be consistent

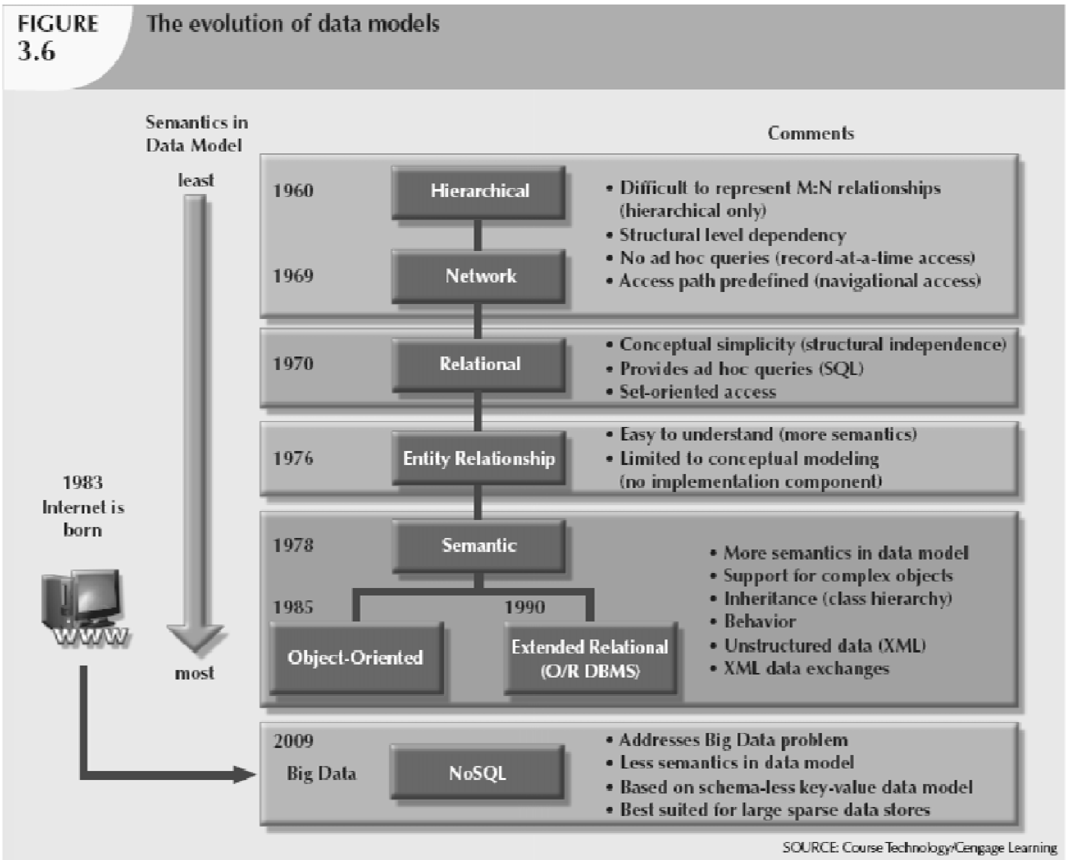


FIGURE 3.6 The evolution of data models

Data Models: A Summary

- Common characteristics:
  - Conceptual simplicity with semantic completeness
  - Represent the real world as closely as possible
  - Real-world transformations must comply with consistency and integrity characteristics
- Each new data model capitalized on the shortcomings of previous models
- Some models better suited for some tasks

TABLE 3.3 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.**

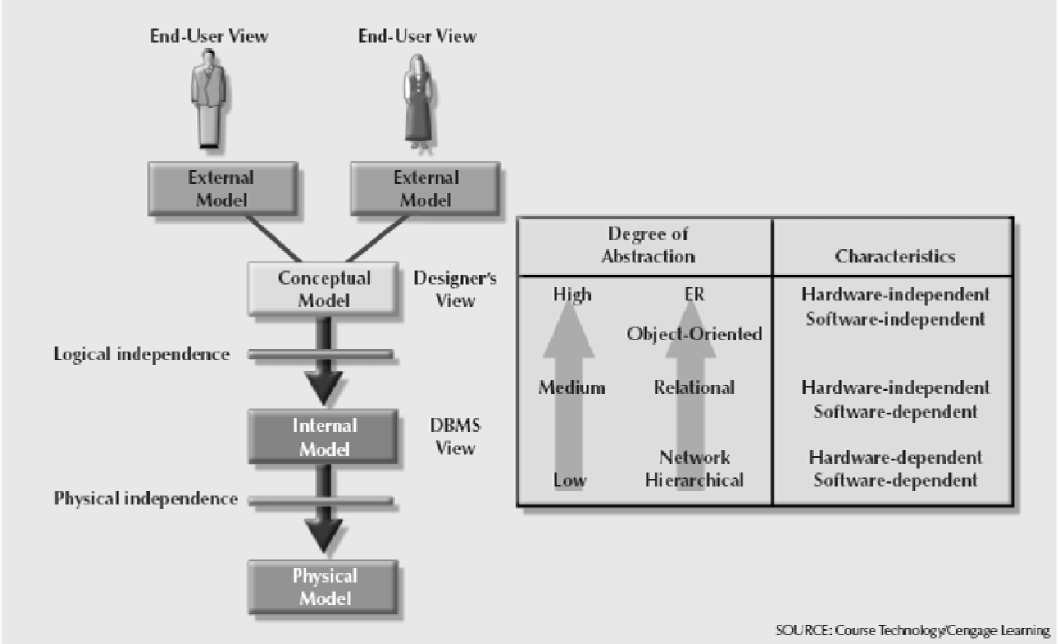
Degrees of Data Abstraction

- Database designer starts with abstracted view, then adds details
- ANSI Standards Planning and Requirements Committee (SPARC)
  - Defined a framework for data modeling based on degrees of data abstraction (1970s):
    - External
    - Conceptual
    - Internal

The External Model

- End users' view of the data environment
- ER diagrams represent external views
- External schema: specific representation of an external view
  - Entities
  - Relationships
  - Processes
  - Constraints

FIGURE 3.7 Data abstraction levels



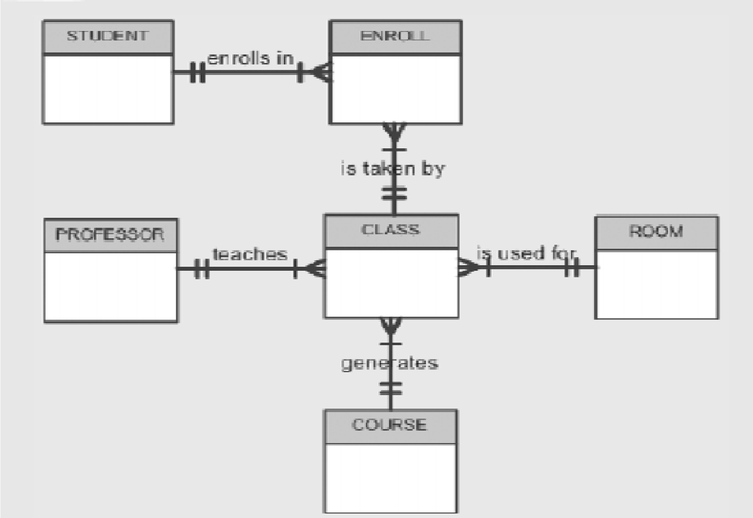
The External Model (cont'd.)

- 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
- Ensures security constraints in database design
- Simplifies application program development

The Conceptual Model

- Represents global view of the entire database
- All external views integrated into single global view: conceptual schema
- ER model most widely used
- ERD graphically represents the conceptual schema

FIGURE 3.9 Conceptual model for Tiny College



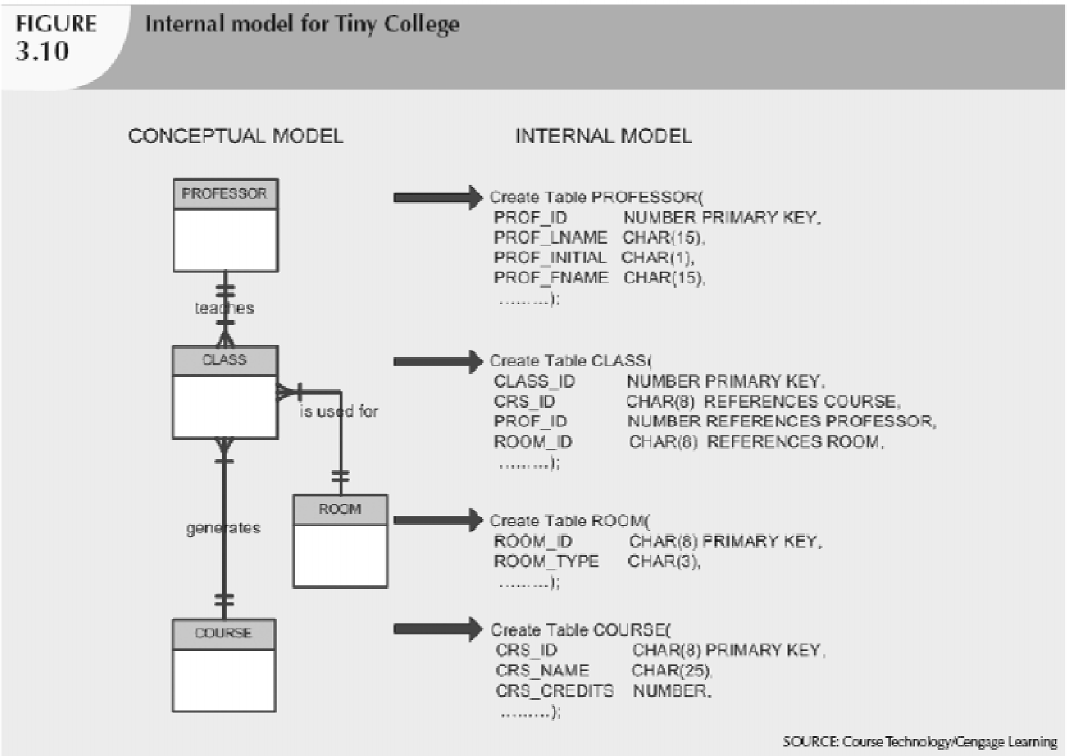
SOURCE: Course Technology/Cengage Learning

The Conceptual Model (cont'd.)

- 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 hardware or software do not affect 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
- Depends on specific database software
  - Change in DBMS software requires internal model be changed
- Logical independence: change internal model without affecting conceptual model



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 3.4 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

Summary

- A data model is an abstraction of a complex real-world data environment
- Basic data modeling components:
  - Entities
  - Attributes
  - Relationships
  - Constraints
- Business rules identify and define basic modeling components

Summary (cont'd.)

- Hierarchical model
  - 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 tool for data modeling
    - Complements relational model

Summary (cont'd.)

- Object-oriented data model: object is basic modeling structure
- Relational model adopted object-oriented extensions: extended relational data model (ERDM)
- OO data models depicted using UML
- Data-modeling requirements are a function of different data views and abstraction levels
  - Three abstraction levels: external, conceptual, and internal

© 2013 Cengage Learning. All Rights Reserved. This edition is intended for use outside of the U.S. only, with content that may be different from the U.S. Edition. May not be scanned, copied, duplicated, or posted to a publicly accessible website, in whole or in part.