Database Systems: Design, Implementation, and Management

Lesson 2

Objectives

In this lesson, 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
- How data models can be classified by 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

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

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 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
- 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 2.1 Linking relational tables

Table name: AGENT (first six attributes) **Database name: Ch02_InsureCo**

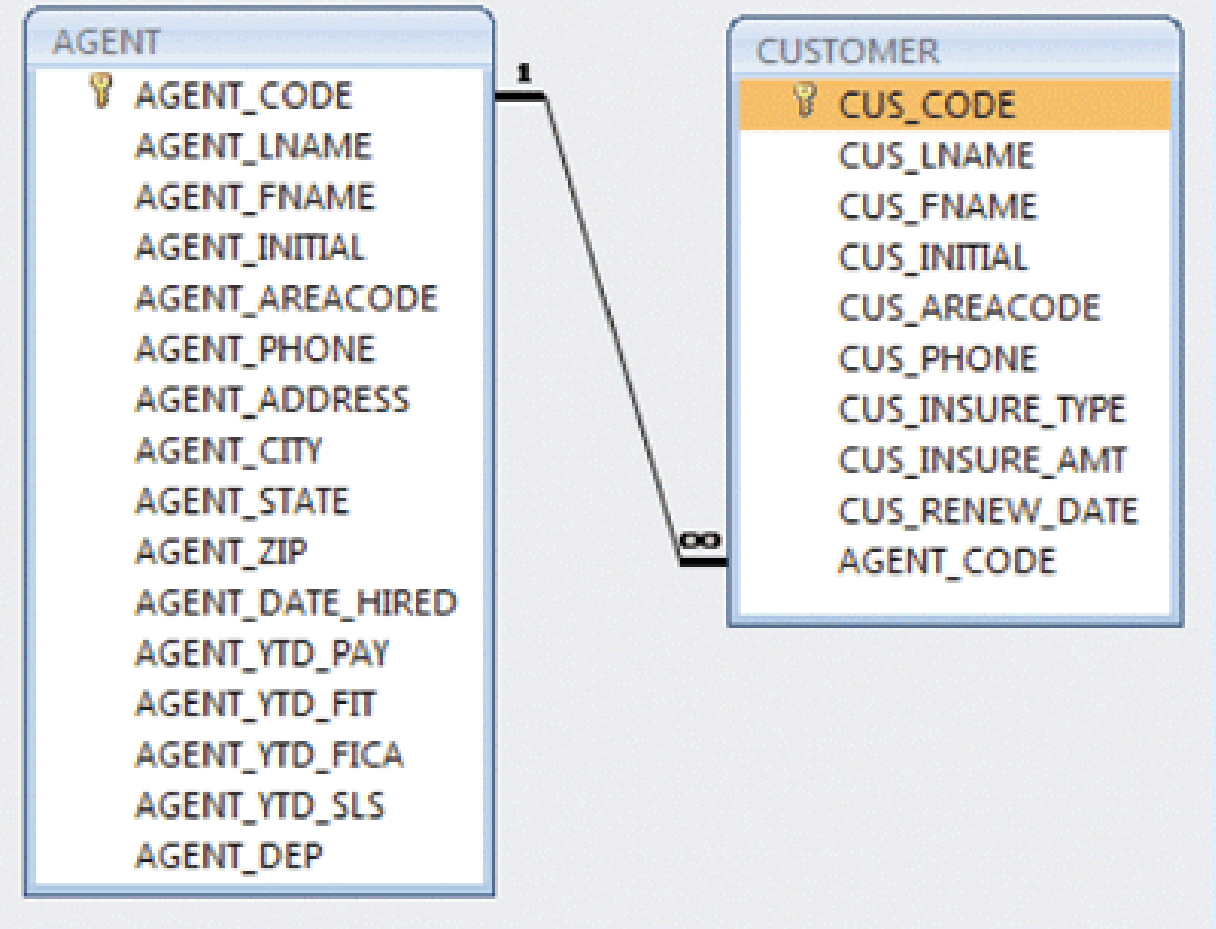
AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
501	Alby	Alex	B	713	228-1249
502	Hahn	Leah	F	615	882-1244
503	Okon	John	T	615	123-5589

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-2010	502
10011	Dunne	Leona	K	713	894-1238	T1	250.00	16-Jun-2010	501
10012	Smith	Kathy	W	615	894-2285	S2	150.00	29-Jan-2011	502
10013	Olowski	Paul	F	615	894-2180	S1	300.00	14-Oct-2010	502
10014	Orlando	Myron		615	222-1672	T1	100.00	28-Dec-2010	501
10015	O'Brian	Amy	B	713	442-3381	T2	850.00	22-Sep-2010	503
10016	Brown	James	G	615	297-1228	S1	120.00	25-Mar-2011	502
10017	Williams	George		615	290-2556	S1	250.00	17-Jul-2010	503
10018	Farriss	Anne	G	713	382-7185	T2	100.00	03-Dec-2010	501
10019	Smith	Olette	K	615	297-3809	S2	500.00	14-Mar-2011	503

FIGURE 2.2 A relational diagram



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

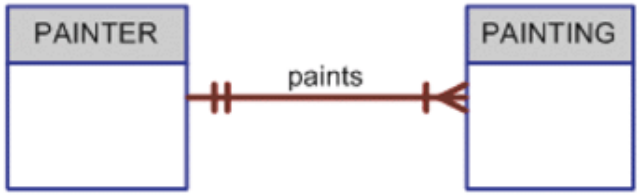
The Entity Relationship Model (cont'd.)

- 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

FIGURE 2.3 The Chen and Crow's Foot notations

Chen Notation *Crow's Foot Notation*

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



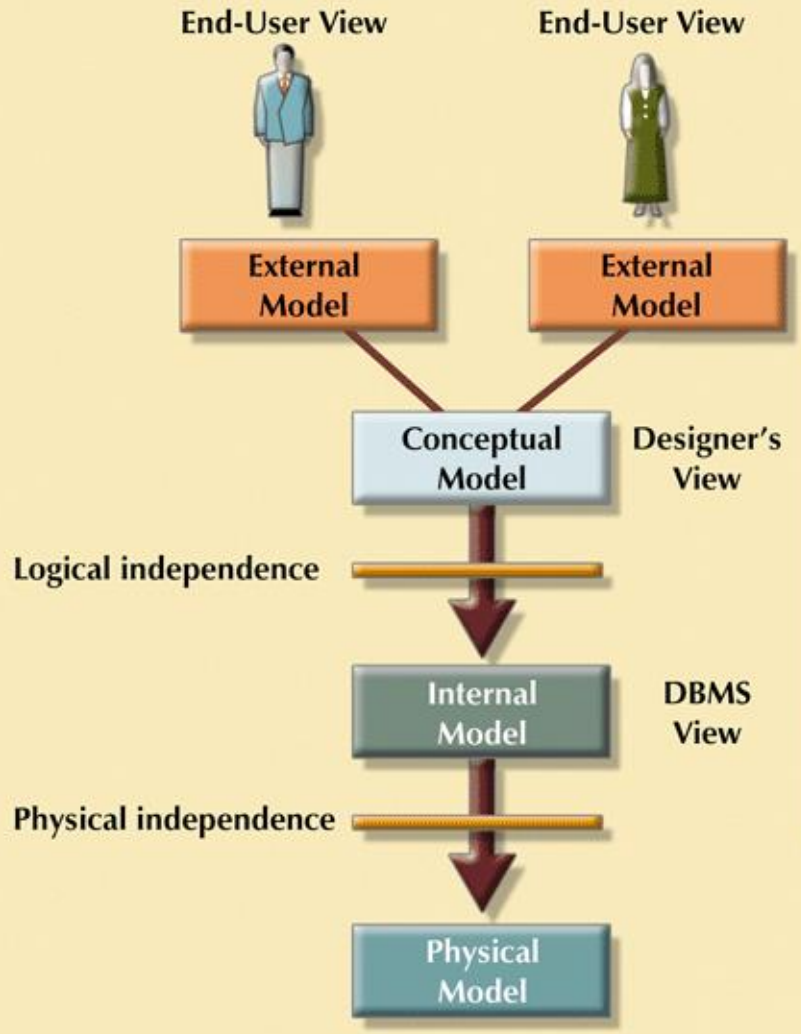
A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.

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

The External Model

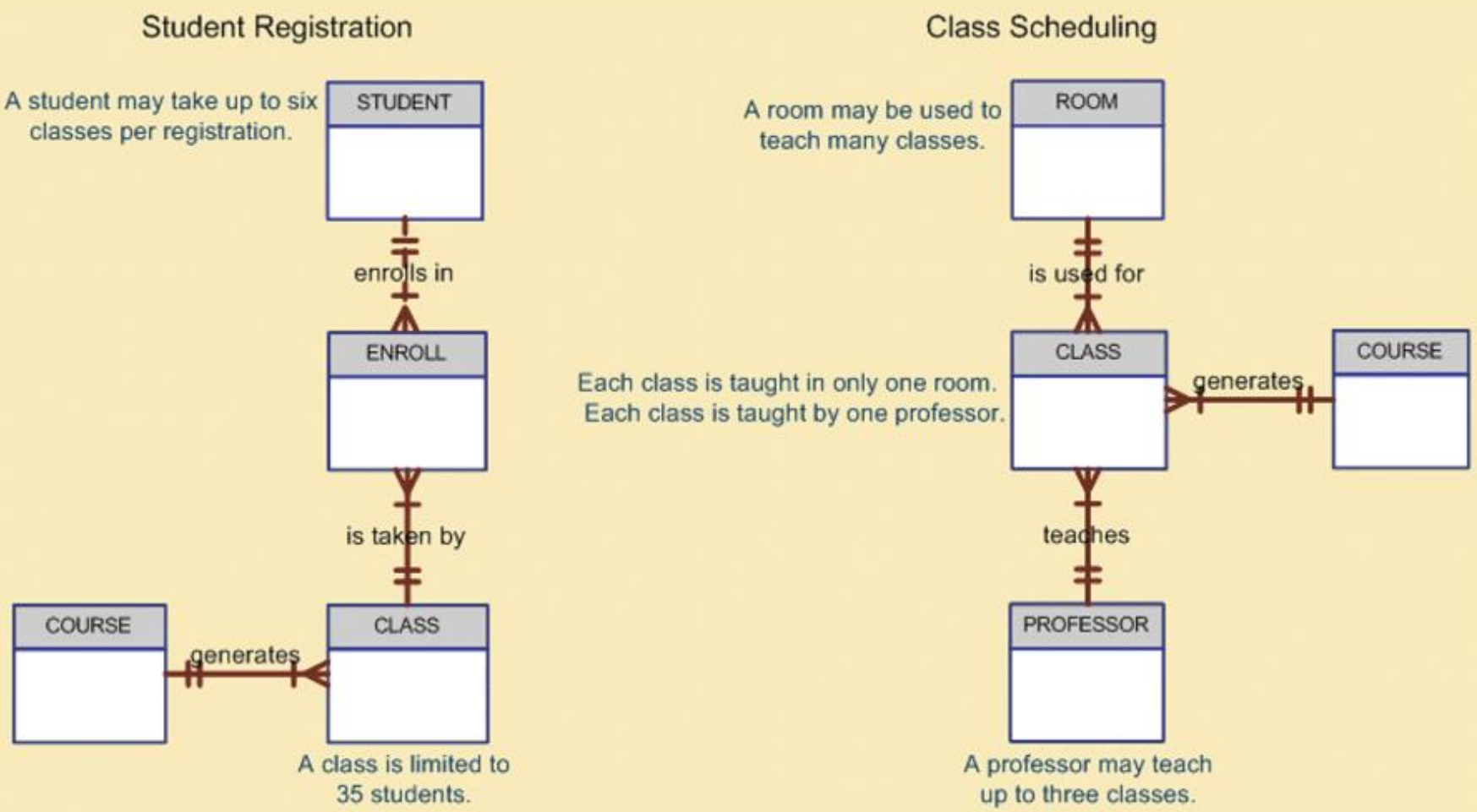
- End users' view of the data environment
- ER diagrams represent external views
- External schema: specific representation of an external view
  - Entities
  - Relationships
  - Processes
  - Constraints

FIGURE 2.6 Data abstraction levels



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

FIGURE 2.7 External models for Tiny College

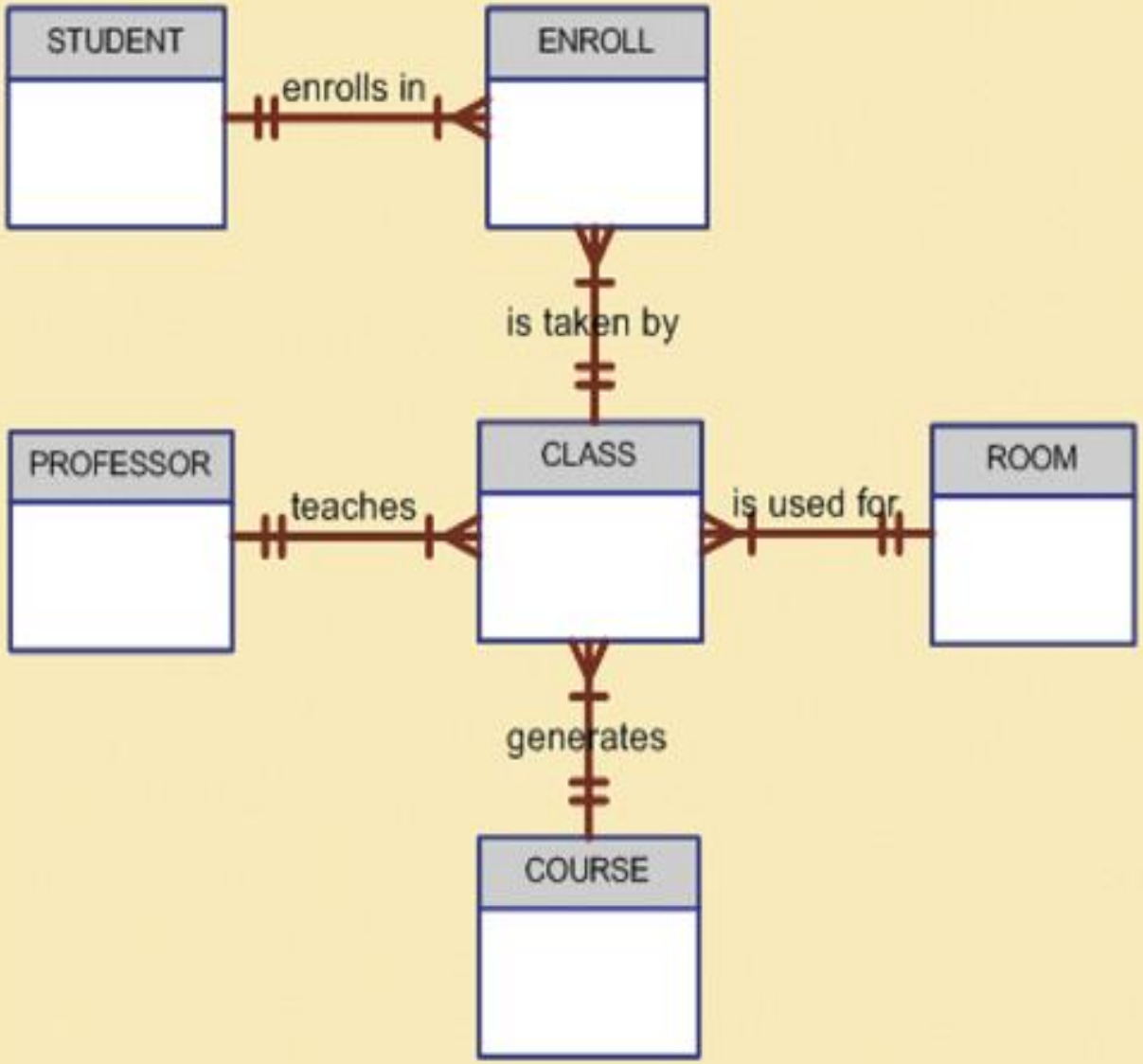


The Conceptual Model

- 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 2.8

Conceptual model for Tiny College



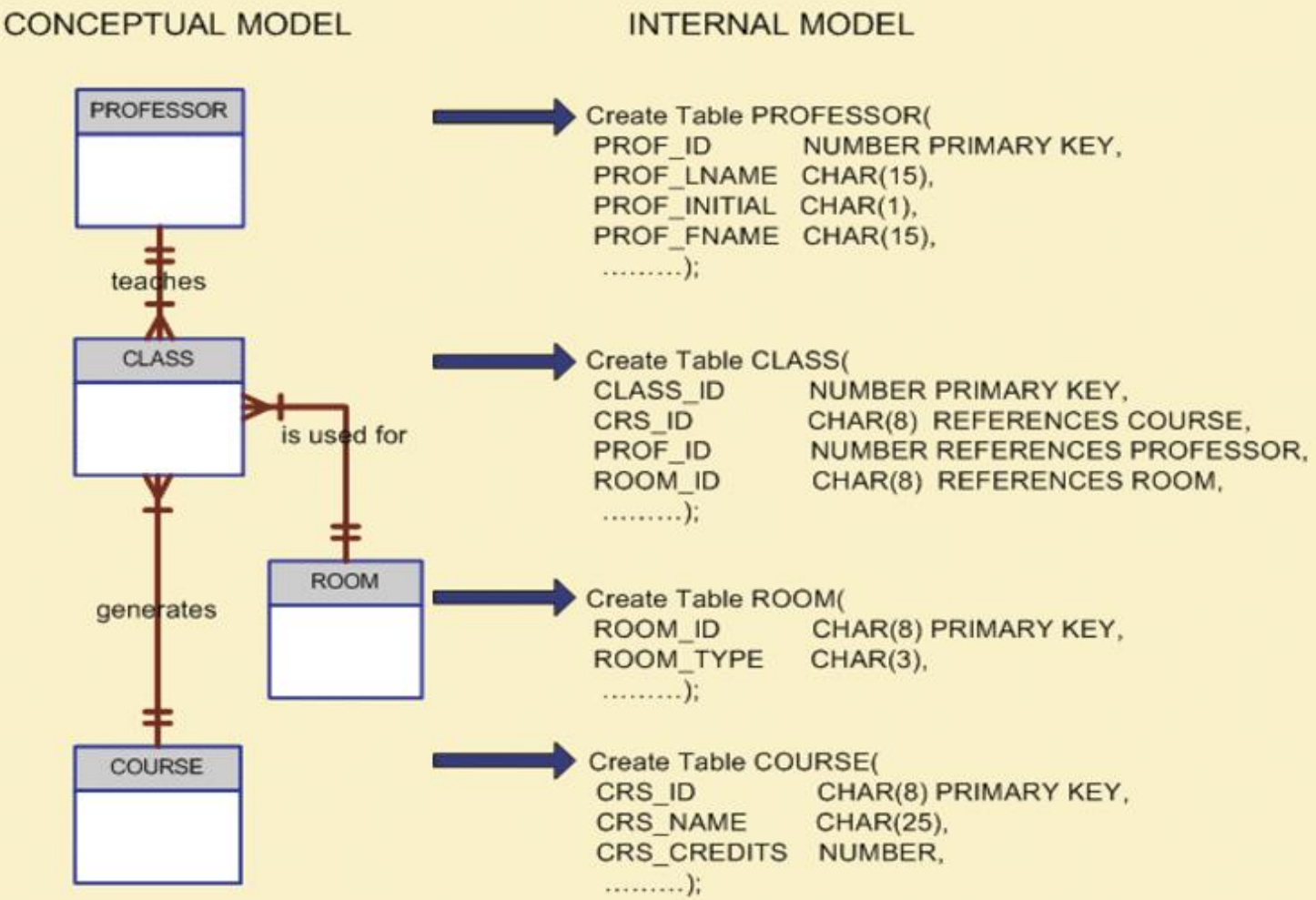
The Conceptual Model (cont'd.)

- 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

FIGURE 2.9 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

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