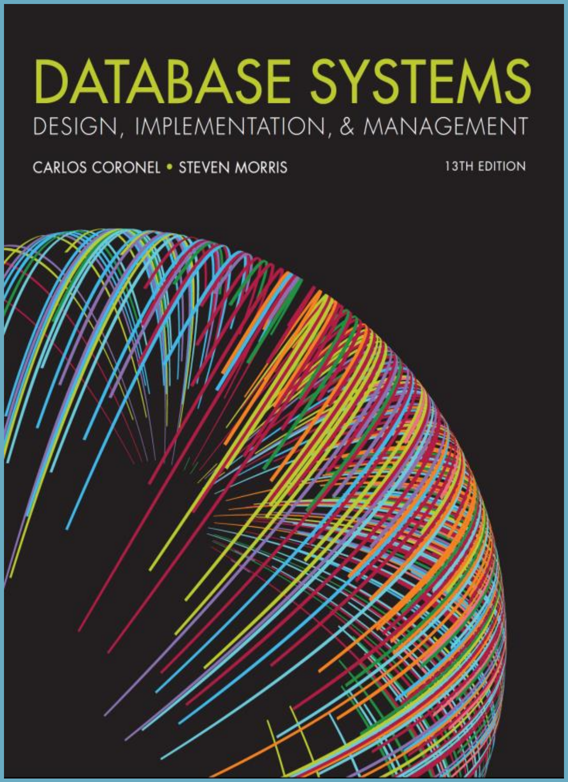


Chapter 2

Data Models

Objectives

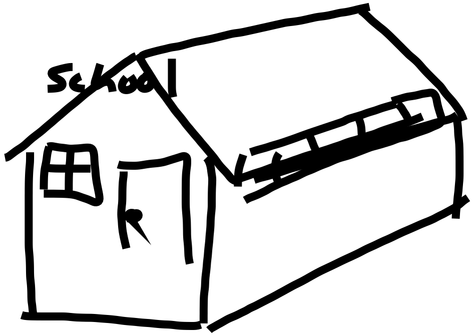
- Describe data modelling and its importance
- Identify and describe data modelling components
- Classify data models by level of abstraction
- Describe business rules and their role in database design

Database Design

What structures will be used to store and manage end-user data?

Data Model

Simple representation of complex, real-world data structures.
An abstraction
Helps to understand the complexities of a real-world object or event



Data modelling is the single most important part of designing a DBMS. It is the very foundation of the system!

The Importance of Data Models

- The importance of data modeling cannot be overstated
  - Facilitates communication
  - Gives various views of the database
  - Organizes data for various users
  - Provides an abstraction for the creation of good a database

Data Model Basic Building Blocks

- **Entity:** person, place, thing, or event about which data will be collected and stored
  - **Attribute:** characteristic of an entity
  - **Relationship:** association among entities
    - One-to-many (1:M OR 1..*)
    - Many-to-many (M:N or *..*)
    - One-to-one (1:1 OR 1..1)
  - **Constraint:** restriction placed on data (Business rules, limitations & Dependencies)
- **Data Integrity**
- Guarantee that data is entered correctly
- Data protection from accidental or intentional change
- **Data Consistency**
- Guarantee that individual data is always the same
  - Changes affect all instances of data
- **Data Redundancy**
- Minimal multiple recurrence of data

CENGAGE

Entity

Any person, place, thing, event about which data are collected.

Each occurrence should be unique

May be physical (customers, products, etc.)
May be abstract (routes, concerts, etc.)

What makes a Student "unique"? SID

What are other College Entities?

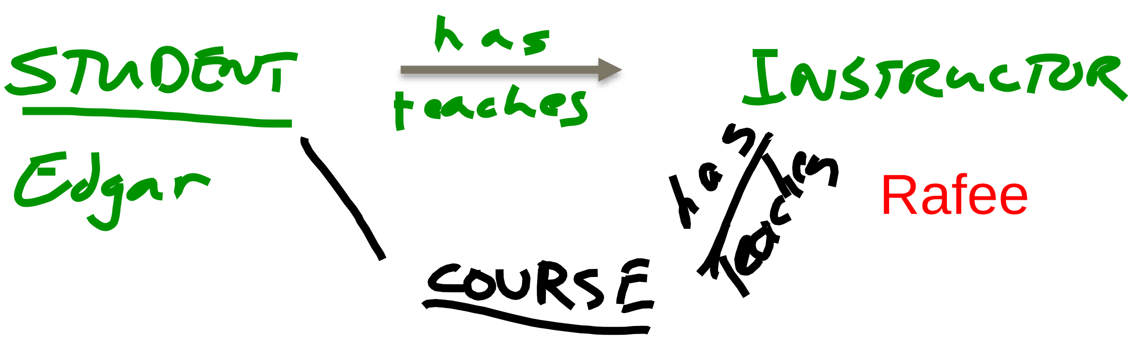
Attribute

A characteristic of an entity.

e.g. STUDENT has the attributes
Student ID
student first name
student last name
student phone
student address
Student grades

Relationship

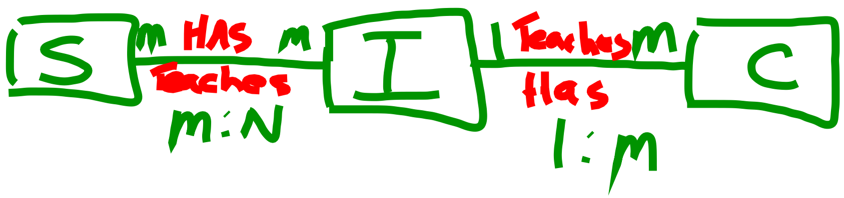
Describes the association among entities.
e.g.
each student has many instructors
each instructor teaches many students
each course has one instructor
each instructor teaches many courses



Relationship

Describes the association among entities.
Three types of relationships
one-to-one (1:1 or 1..1)
one-to-many (1:M or 1..*)
many-to-many (M:N or *..*)

many
0..∞
1..∞
0..M
1..M



Constraint

Restrictions placed on the data
Helps to ensure data integrity
e.g.
GPA range 0.00 - 4.00
Class must have only one teacher

Business Rules

Brief, precise, unambiguous description of policy, procedure, principle
e.g.
a customer may generate many invoices
an invoice is generated by only one customer

Business Rules

- Standardize the view of the data
- Communications tool between users and designers
- nature, role, and scope of data
- business processes
- relationship participation rules

Translating Business Rules into Data Model Components

- Business rules set the stage for the proper identification of entities, attributes, relationships, and constraints
  - 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

⇕

Entity_attribute

STUDENT

student_phone

STUDENT

Student_ID

Section

section_ID

course_ID

Student_ID

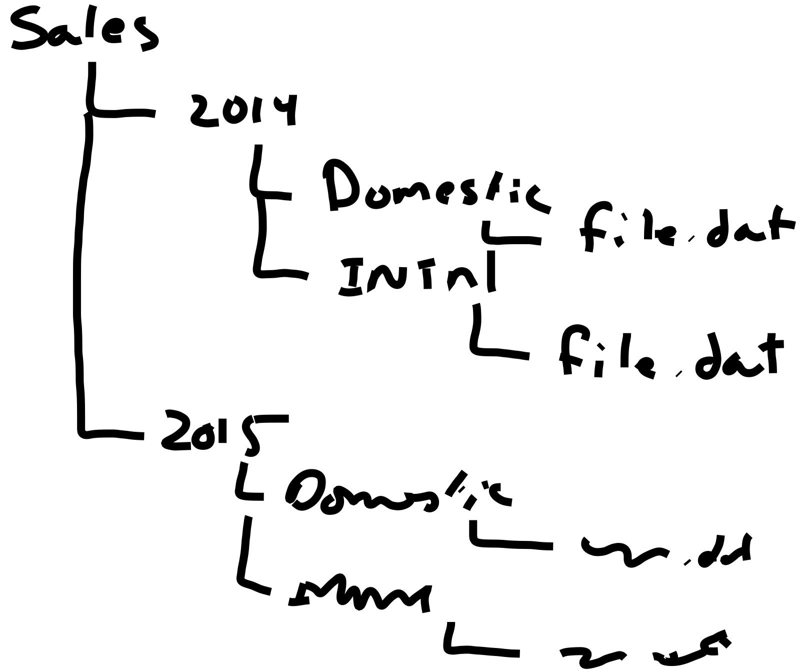
INSTR_ID

Evolution of Data Models

File System
Hierarchical & Network
Relational
Object-oriented & Object/Relational
XML
NoSQL: key-value store, column store

Evolution of Data Models

File System



Evolution of Data Models

Hierarchical Model
Upside down tree
Each level (segment) is a 1:M relationship to the level below it

Segments

CIT1163 C01.15WN Database Rafee
S0001234 Bob
S0002468 Sally
S0003458 Phedd

CIT1156 C01.16WN Prog II Barry
S001234 Bob
A1 B
A2 C
A3 A

Evolution of Data Models

Network Model

Schema - conceptual organization as viewed by the DBA, logical grouping of objects (tables, indexes, views, queries)

Subschema - database as seen by/interaction with applications (“view”)

Data Manipulation Language (DML) - defines the environment, manipulates the data

Data Definition Language (DDL) - enables DBA to define components (structure, types, etc.)

Basically, data is represented as a collection of many-to-one relationships

Evolution of Data Models

Relational Model - Thanks E. F. Codd, 1970

Re**lation** (table) is a matrix of rows and columns ENTITY

Tu**ple** is a row in a relation instance

Relation

STUDENT

student_first	student_last	student_gpa
Joe	Strummer	3.76
Annie	Falcon	3.76

Relationships

STUDENT

	student_first	student_last	student_gpa
	Joe	Strummer	3.76
	Annie	Falcon	3.76

each many

Many each

COURSE

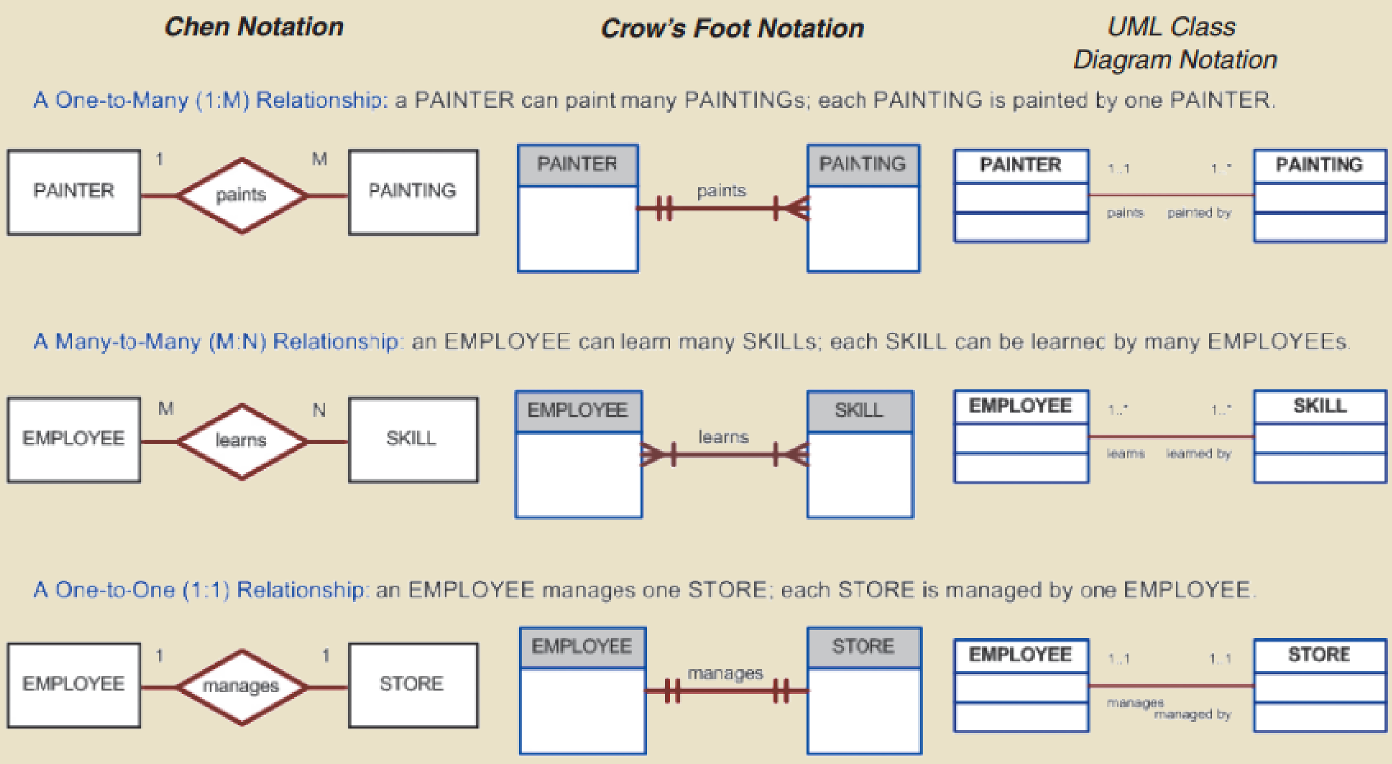
	course_id	course_name	course_instructor
	CIT1163	Database	Rafee
	CIT1156	C++	Robinson

The Entity Relationship Model (1 of 2)

- 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
  - Attributes: describe particular characteristics
  - Connectivity: term used to label the relationship types

The Entity Relationship Model (2 of 2)

FIGURE 2.3 THE ER MODEL NOTATIONS

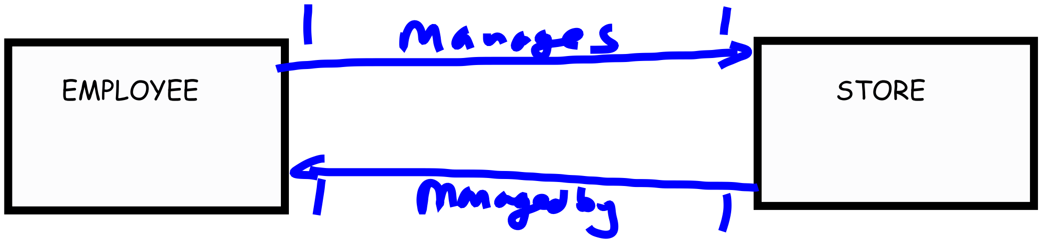


Relationship

Describes the association among entities.
Three types of relationships
one-to-one (1:1 or 1..1)
one-to-many (1:M or 1..*)
many-to-many (M:N or *..*)

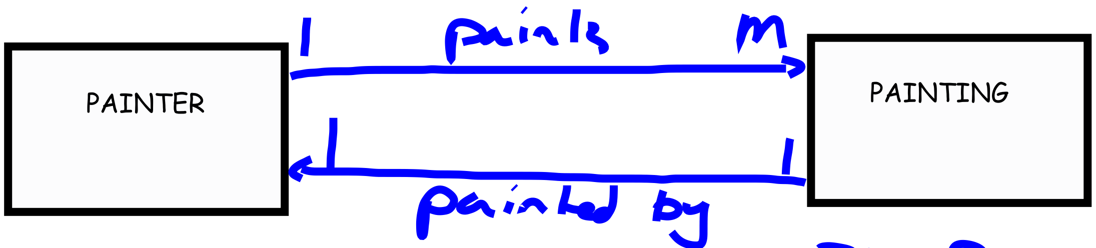
One-to-One

One employee manages one store, each store is managed by one employee



One-to-Many

One painter has (paints) many paintings, each painting is painted by one painter

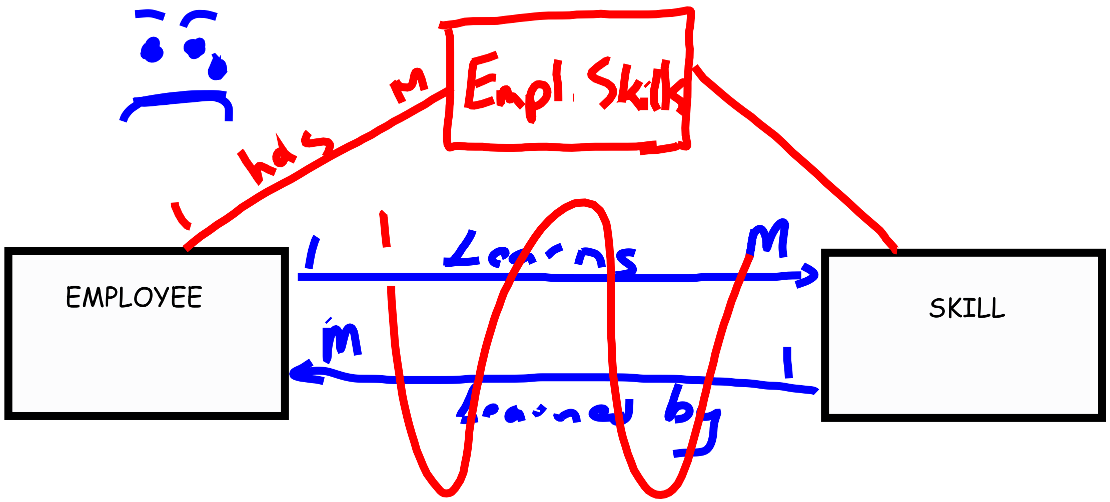


gainesborough
Van gogh
Monet

Blue Boy
Starry Night
Sunflowers
Bouquet of Sunflowers:

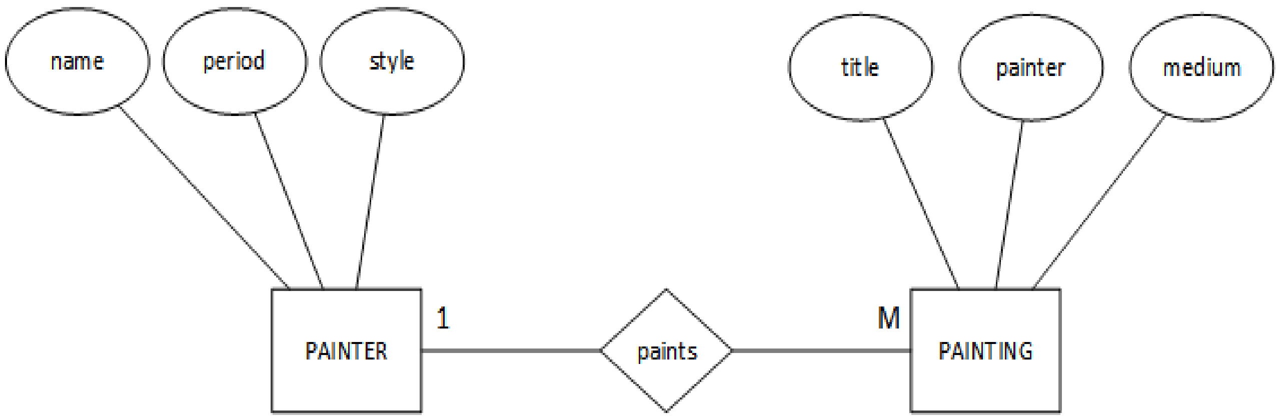
Many-to-Many

An employee can learn many skills, each skill can be learned by many employees



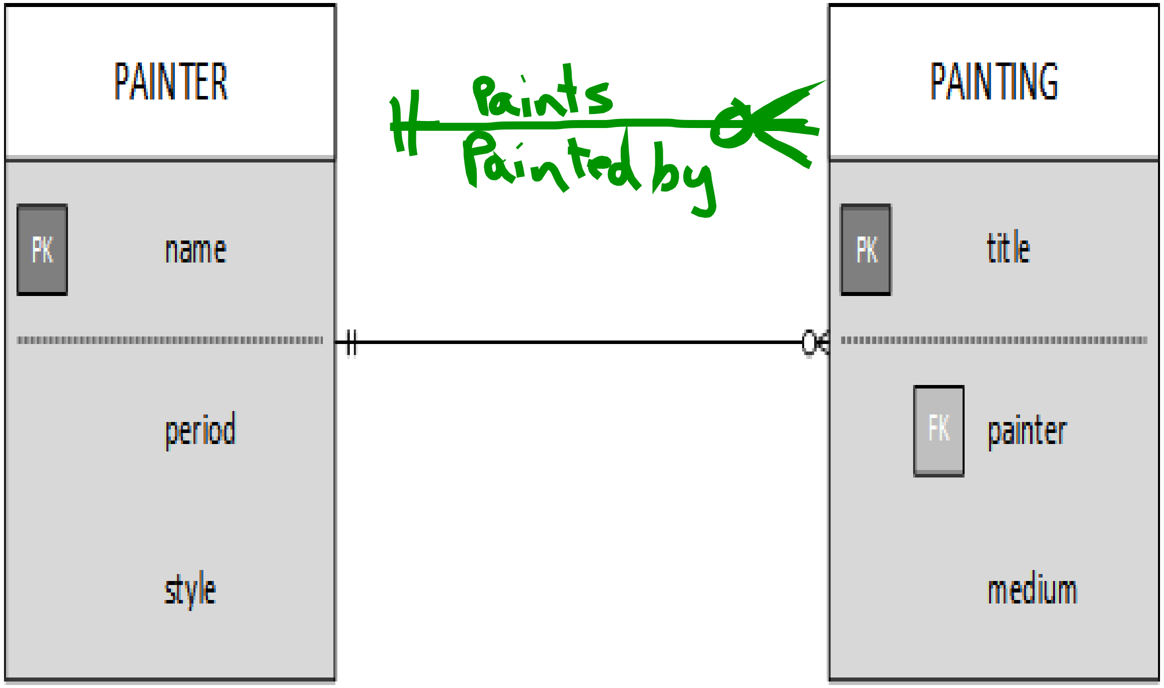
Entity Relationship Diagram

Chen



Entity Relationship Diagram

Crow's Foot

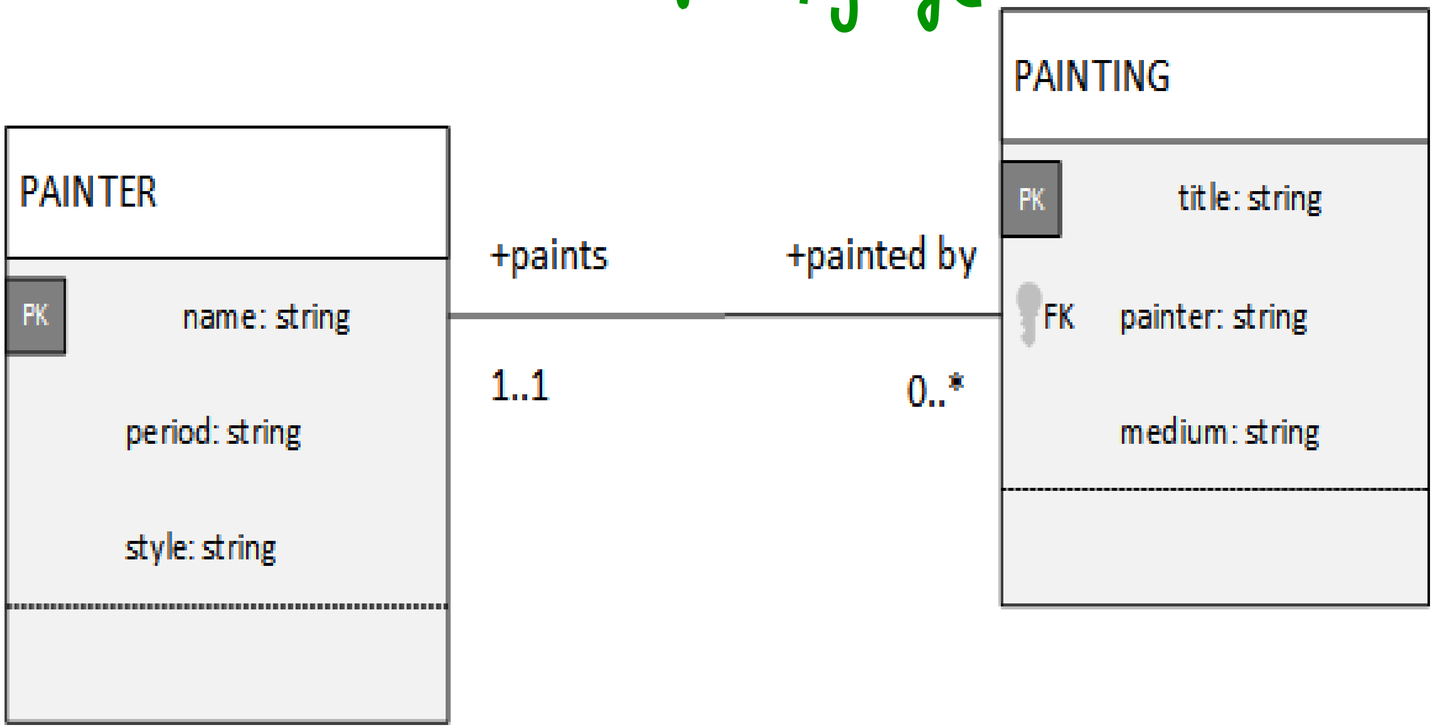


Entity Relationship Diagram

UML

Object-oriented

-unified markup Language

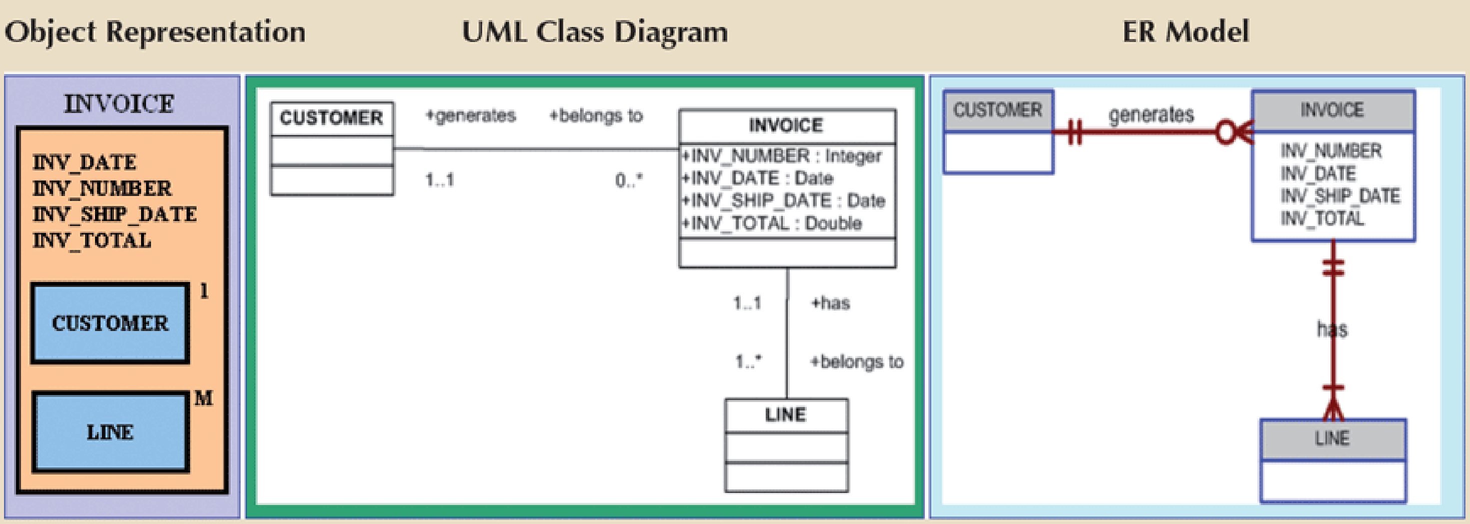


The Object-Oriented Data Model

- Both data and its relationships are contained in a single structure known as an object
- Object: contains data and their relationships with operations that are performed on it
  - Basic building block for autonomous structures
  - Abstraction of real-world entity
- Attribute: describes the properties of an object

The Object-Oriented Data Model

FIGURE 2.4 A COMPARISON OF THE OO, UML, AND ER MODELS



Emerging Data Models: Big Data and NoSQL (1 of 3)

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

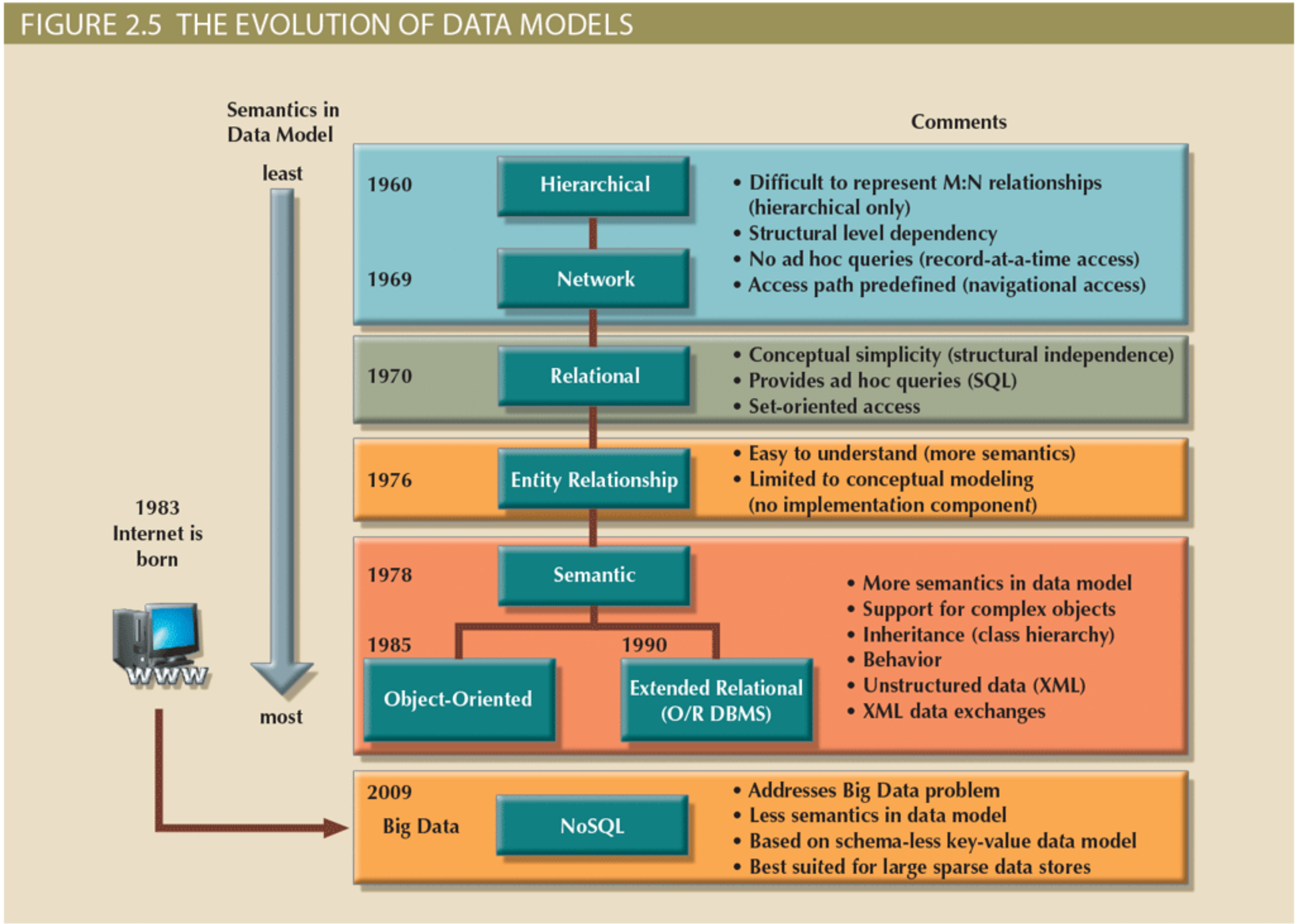
Emerging Data Models: Big Data and NoSQL (2 of 3)

- Challenges of Big Data
  - Volume doesn't allow usage of conventional structures
  - Expensive
- New technologies of Big Data
  - Hadoop
  - Hadoop Distributed File System (HDFS)
  - MapReduce
  - NoSQL

Emerging Data Models: Big Data and NoSQL (3 of 3)

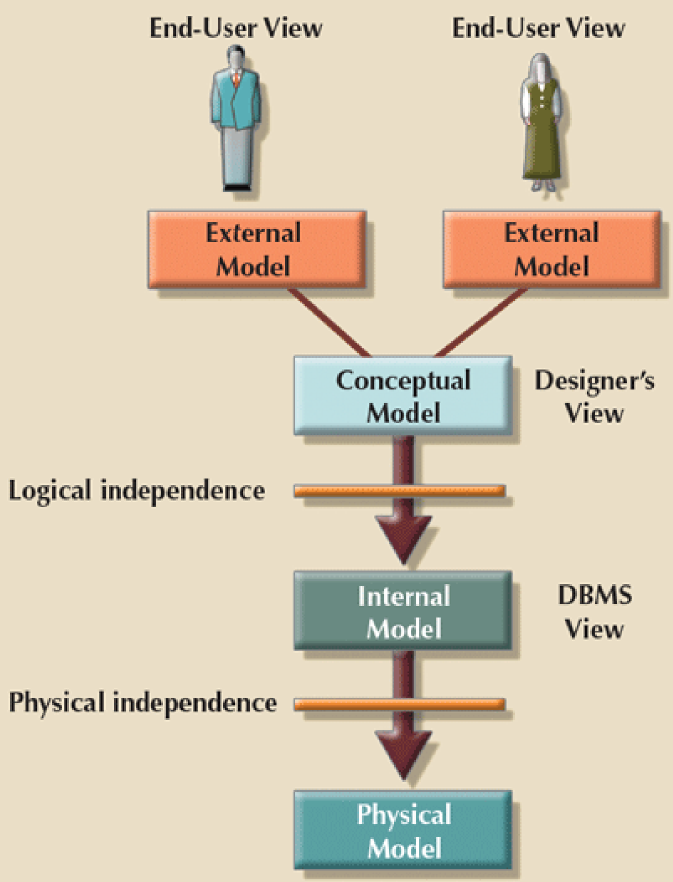
- NoSQL databases
  - Not based on the relational model
  - Support distributed database architectures
  - Provide high scalability, high availability, and fault tolerance
  - Support large amounts of sparse data
  - Geared toward performance rather than transaction consistency
  - Provides a broad umbrella for data storage and manipulation

Data Models: A Summary



Degrees of Data Abstraction

FIGURE 2.6 DATA ABSTRACTION LEVELS



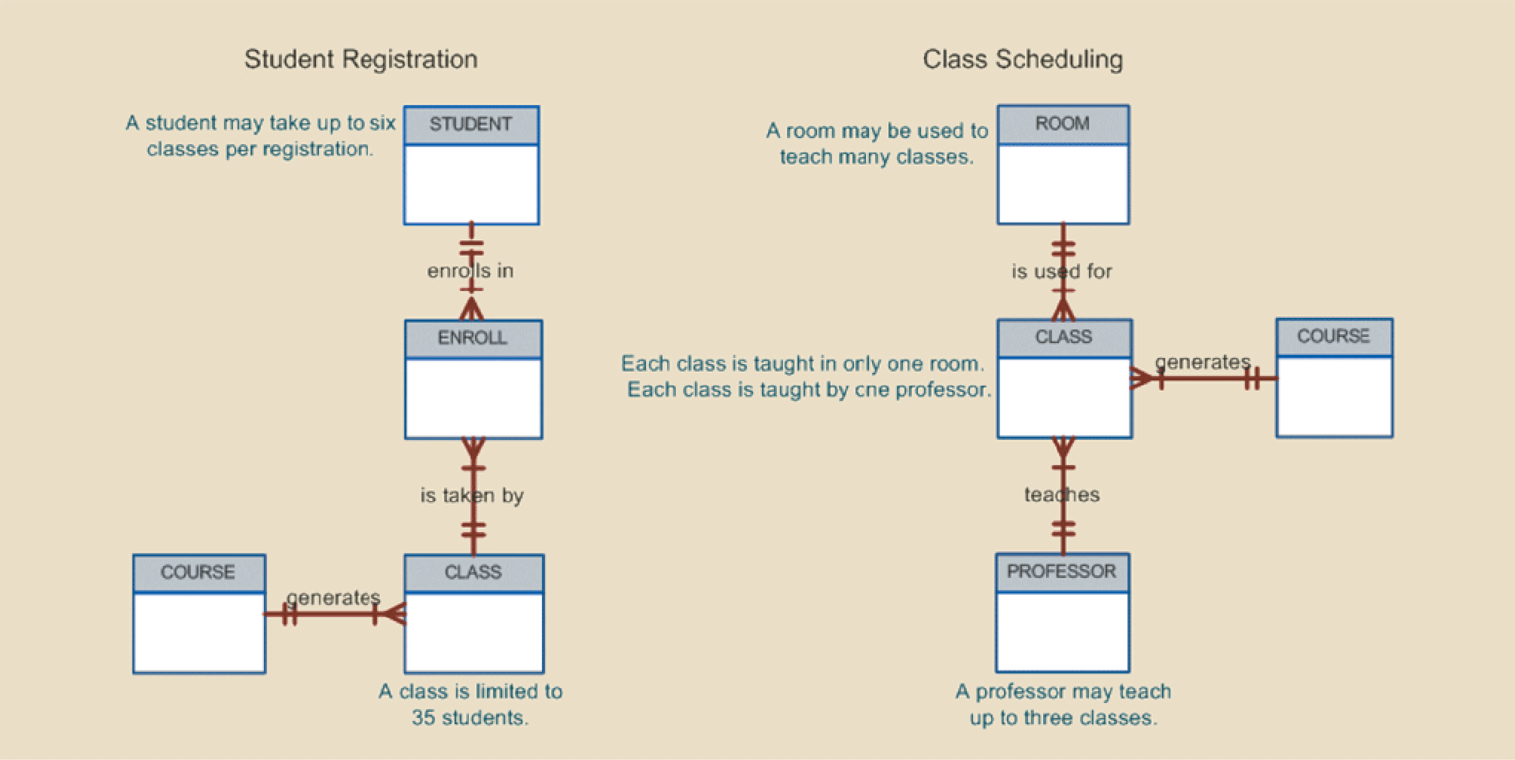
Degree of Abstraction		Characteristics
High	ER Object-Oriented	Hardware-independent Software-independent
Medium	Relational	Hardware-independent Software-dependent
Low	Network Hierarchical	Hardware-dependent Software-dependent

The External Model (1 of 2)

- End users' view of the data environment
  - People who use the application programs to manipulate the data and generate information
- ER diagrams are used to represent the external views
  - External schema: specific representation of an external view

The External Model (2 of 2)

FIGURE 2.7 EXTERNAL MODELS FOR TINY COLLEGE

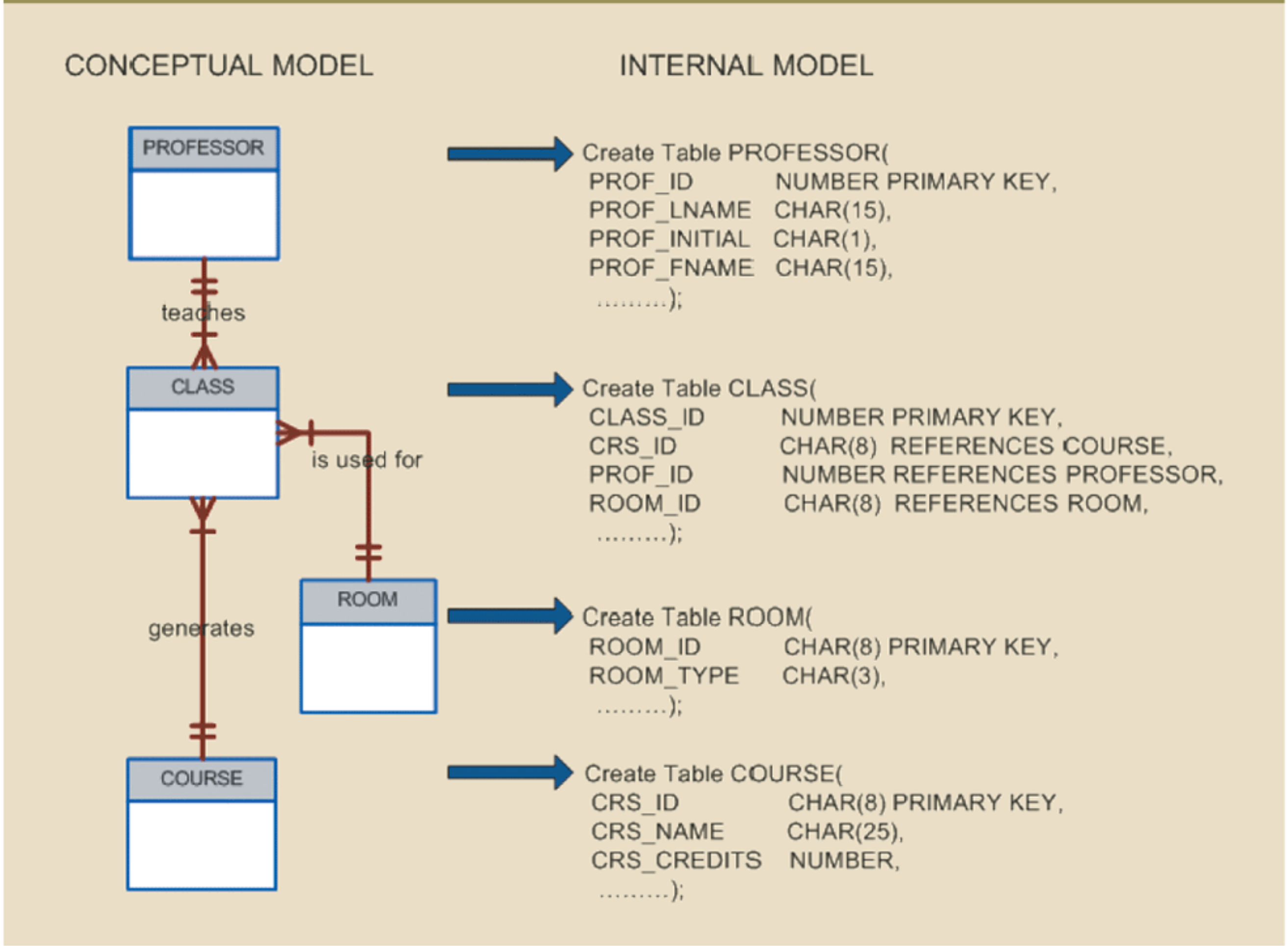


The Conceptual Model (1 of 2)

- Represents a global view of the entire database by the entire organization
- Conceptual model advantages
  - Macro-level view of data environment
  - Software and hardware independent

The Conceptual Model (2 of 2)

FIGURE 2.9 INTERNAL MODEL FOR TINY COLLEGE



CENGAGE

The Internal Model

- Representing database as seen by the DBMS mapping conceptual model to the DBMS
  - Logical independence: changing internal model without affecting the conceptual model
  - Hardware independent: unaffected by the type of computer on which the software is installed

The Physical Model (1 of 2)

- Operates at lowest level of abstraction
  - Describes the way data are saved on storage media such as magnetic, solid state, or optical media
- Requires the definition of physical storage and data access methods
  - Software and hardware dependent
- Physical independence: changes in physical model do not affect internal model

The Physical Model (2 of 2)

Table 2.4 Levels of Data Abstraction			
Model	Degree of Abstraction	Focus	Independent of
External	High	End-user views	Hardware and software
Conceptual	Medium-High	Global view of data (database model independent)	Hardware and software
Internal	Medium-Low	Specific database model	Hardware
Physical	Low	Storage and access methods	Neither hardware nor software

CENGAGE

Summary

- A data model is an abstraction of a complex real-world data environment
- There are many types of data models (e.g., hierarchical, network, relational, object-oriented, extended relational data model, etc.)
- Data-modeling requirements are a function of different data views (global versus local) and the level of data abstraction