Chapter 2

Data Model

Database Systems:
Design, Implementation, and Management,
Sixth Edition, Rob and Coronel
In this chapter, you will learn:

• Why data models are important
• About the basic data-modeling building blocks
• What business rules are and how they affect database design
• How the major data models evolved, and their advantages and disadvantages
• How data models can be classified by level of abstraction
The Importance of Data Models

• Data model
  – Relatively simple representation, usually graphical, of complex real-world data structures
  – Communications tool to facilitate interaction among the designer, the applications programmer, and the end user

• Good database design uses an appropriate data model as its foundation
Importance of Data Modeling

- End-users have different views and needs for data
- Data model organizes data for various users
Data Model Basic Building Blocks

• Entity is anything about which data are to be collected and stored
• Attribute is a characteristic of an entity
• Relationship describes an association among (two or more) entities
  – One-to-many (1:M) relationship
  – Many-to-many (M:N or M:M) relationship
  – One-to-one (1:1) relationship
Business Rules

• Brief, precise, and unambiguous description of a policy, procedure, or principle within a specific organization’s environment

• Apply to any organization that stores and uses data to generate information

• Description of operations that help to create and enforce actions within that organization’s environment
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
  – Procedures
  – Standards
  – Operations manuals
• Direct interviews with end users
Importance of Business Rules

• Promote creation of an accurate data model
• Standardize company’s view of data
• Constitute a 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
The Evolution of Data Models

- Hierarchical
- Network
- Relational
- Entity relationship
- Object oriented
The Hierarchical Model—Evolution

• GUAM (Generalized Update Access Method)
  – Based on the recognition that the many smaller parts would come together as components of still larger components

• Information Management System (IMS)
  – World’s leading mainframe hierarchical database system in the 1970s and early 1980s
The Hierarchical Model—Characteristics

• Basic concepts form the basis for subsequent database development

• Limitations lead to a different way of looking at database design

• Basic concepts show up in current data models

• Best understood by examining manufacturing process
A Hierarchical Structure

FIGURE 2.1 A HIERARCHICAL STRUCTURE
Hierarchical Structure—Characteristics

• Each parent can have many children
• Each child has only one parent
• Tree is defined by path that traces parent segments to child segments, beginning from the left

• Hierarchical path
  – Ordered sequencing of segments tracing hierarchical structure

• Preorder traversal or hierarchic sequence
  – “Left-list” path
The Hierarchical Model

• Advantages
  – Conceptual simplicity
  – Database security
  – Data independence
  – Database integrity
  – Efficiency
The Hierarchical Model (continued)

• Disadvantages
  – Complex implementation
  – Difficult to manage
  – Lacks structural independence
  – Complex applications programming and use
  – Implementation limitations
  – Lack of standards
Child with Multiple Parents

FIGURE 2.2 CHILD WITH MULTIPLE PARENTS
The Network Model

• Created to
  – Represent complex data relationships more effectively
  – Improve database performance
  – Impose a database standard

• Conference on Data Systems Languages (CODASYL)

• American National Standards Institute (ANSI)

• Database Task Group (DBTG)
Crucial Database Components

• Schema
  – Conceptual organization of entire database as viewed by the database administrator

• Subschema
  – Defines database portion “seen” by the application programs that actually produce the desired information from data contained within the database

• Data Management Language (DML)
  – Define data characteristics and data structure in order to manipulate the data
Data Management Language Components

• Schema Data Definition Language (DDL)
  – Enables database administrator to define schema components

• Subschema DDL
  – Allows application programs to define database components that will be used

• DML
  – Manipulates database contents
Network Model—Basic Structure

• Resembles hierarchical model
• Collection of records in 1:M relationships
• Set
  – Relationship
  – Composed of at least two record types
    • Owner
      – Equivalent to the hierarchical model’s parent
    • Member
      – Equivalent to the hierarchical model’s child
A Network Data Model

FIGURE 2.3 A NETWORK DATA MODEL

- **SALESREP**
  - 1:M with **CUSTOMER** (Commission set)
  - 1:M with **PRODUCT** (Inventory set)
  - 1:M with **INV_LINE**

- **CUSTOMER**
  - 1:M with **INVOICE** (Sales set)
  - 1:M with **PAYMENT** (Payment set)

- **PRODUCT**
  - 1:M with **INVOICE** (Line set)
  - 1:M with **INV_LINE**
The Network Data Model

• Advantages
  – Conceptual simplicity
  – Handles more relationship types
  – Data access flexibility
  – Promotes database integrity
  – Data independence
  – Conformance to standards
The Network Data Model (continued)

• Disadvantages
  – System complexity
  – Lack of structural independence
The Relational Model

• Developed by Codd (IBM) in 1970
• Considered ingenious but impractical in 1970
• Conceptually simple
• Computers lacked power to implement the relational model
• Today, microcomputers can run sophisticated relational database software
The Relational Model—Basic Structure

• Relational Database Management System (RDBMS)

• Performs same basic functions provided by hierarchical and network DBMS systems, plus other functions

• Most important advantage of the RDBMS is its ability to let the user/designer operate in a human logical environment
The Relational Model—Basic Structure (continued)

• Table (relations)
  – Matrix consisting of a series of row/column intersections
  – Related to each other by sharing a common entity characteristic

• Relational schema
  – Visual representation of relational database’s entities, attributes within those entities, and relationships between those entities
Relational Table

• Stores a collection of related entities
  – Resembles a file

• 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
A Relational Schema

FIGURE 2.5 A RELATIONAL SCHEMA
Linking Relational Tables

**Figure 2.4 Linking Relational Tables**

<table>
<thead>
<tr>
<th>Database name: Ch02_InsureCo</th>
<th>Table name: AGENT (first six attributes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENT_CODE</td>
<td>AGENT_LNAME</td>
</tr>
<tr>
<td>501</td>
<td>Alby</td>
</tr>
<tr>
<td>502</td>
<td>Hahn</td>
</tr>
<tr>
<td>503</td>
<td>Okon</td>
</tr>
</tbody>
</table>

Link through AGENT_CODE

<table>
<thead>
<tr>
<th>Table name: CUSTOMER</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUS_CODE</td>
</tr>
<tr>
<td>10010</td>
</tr>
<tr>
<td>10011</td>
</tr>
<tr>
<td>10012</td>
</tr>
<tr>
<td>10013</td>
</tr>
<tr>
<td>10014</td>
</tr>
<tr>
<td>10015</td>
</tr>
<tr>
<td>10016</td>
</tr>
<tr>
<td>10017</td>
</tr>
<tr>
<td>10018</td>
</tr>
<tr>
<td>10019</td>
</tr>
</tbody>
</table>
The Relational Model

• Advantages
  – Structural independence
  – Improved conceptual simplicity
  – Easier database design, implementation, management, and use
  – Ad hoc query capability
  – Powerful database management system
The Relational Model (continued)

• Disadvantages
  – Substantial hardware and system software overhead
  – Can facilitate poor design and implementation
  – May promote “islands of information” problems
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—Basic Structure

• 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
Relationships: The Basic Chen ERD

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
FIGURE 2.7  RELATIONSHIPS: THE BASIC CROW’S FOOT ERD

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 Entity Relationship Model

• Advantages
  – Exceptional conceptual simplicity
  – Visual representation
  – Effective communication tool
  – Integrated with the relational data model
The Entity Relationship Model (continued)

• Disadvantages
  – Limited constraint representation
  – Limited relationship representation
  – No data manipulation language
  – Loss of information content
The Object Oriented Model

- Semantic data model (SDM) developed by Hammer and McLeod in 1981
- Modeled both data and their relationships in a single structure known as an object
- Basis of object oriented data model (OODM)
- OODM becomes the basis for the object oriented database management system (OODBMS)
The Object Oriented Model (continued)

• Object is described by its factual content
  – Like relational model’s entity
• Includes information about relationships between facts within object and relationships with other objects
  – Unlike relational model’s entity
• Subsequent OODM development allowed an object to also contain operations
• Object becomes basic building block for autonomous structures
Developments that Boosted OODM’s Popularity

- Growing costs put a premium on code reusability
- Complex data types and system requirements became difficult to manage with a traditional RDBMS
- Became possible to support increasingly sophisticated transaction & information requirements
- Ever-increasing computing power made it possible to support the large computing overhead required
Object Oriented Data Model—
Basic Structure

- Object: abstraction of a real-world entity
- Attributes describe the properties of an object
- Objects that share similar characteristics are grouped in classes
- Classes are organized in a class hierarchy
- Inheritance is the ability of an object within the class hierarchy to inherit the attributes and methods of classes above it
A Comparison of the OO Model and the ER Model

FIGURE 2.8 A COMPARISON OF THE OO MODEL AND THE ER MODEL
The Object Oriented Model

• Advantages
  – Adds semantic content
  – Visual presentation includes semantic content
  – Database integrity
  – Both structural and data independence
The Object Oriented Model (continued)

- Disadvantages
  - Slow pace of OODM standards development
  - Complex navigational data access
  - Steep learning curve
  - High system overhead slows transactions
  - Lack of market penetration
Other Models

- Extended Relational Data Model (ERDM)
  - Semantic data model developed in response to increasing complexity of applications
  - DBMS based on the ERDM often described as an object/relational database management system (O/RDBMS)
  - Primarily geared to business applications
Other Models (continued)

• Date’s objections to ERDM label
  – Given proper support for domains, relational data models are quite capable of handling complex data
    • Therefore, capability that is supposedly being extended is already there
  – O/RDM label is not accurate because the relational data model’s domain is not an object model structure
Data Models: A Summary

• Each new data model capitalized on the shortcomings of previous models
• Common characteristics:
  – Conceptual simplicity without compromising the semantic completeness of the database
  – Represent the real world as closely as possible
  – Representation of real-world transformations (behavior) must be in compliance with consistency and integrity characteristics of any data model
The Development of Data Models

**Figure 2.9** The Development of Data Models

<table>
<thead>
<tr>
<th>Semantics in Data Model</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchical</td>
<td>• Difficult to represent M:N relationships (Hierarchical only)</td>
</tr>
<tr>
<td></td>
<td>• Physical level dependency</td>
</tr>
<tr>
<td></td>
<td>• No ad hoc queries</td>
</tr>
<tr>
<td></td>
<td>• Access path predefined (navigational access)</td>
</tr>
<tr>
<td>Network</td>
<td></td>
</tr>
<tr>
<td>Relational</td>
<td>• Provides ad hoc queries</td>
</tr>
<tr>
<td></td>
<td>• Set-oriented access</td>
</tr>
<tr>
<td></td>
<td>• Weak semantic content</td>
</tr>
<tr>
<td>Entity Relationship</td>
<td>• Easy to understand</td>
</tr>
<tr>
<td></td>
<td>• Incorporates more semantics</td>
</tr>
<tr>
<td>Semantic</td>
<td></td>
</tr>
<tr>
<td>Object-Oriented</td>
<td>• More semantics in data model</td>
</tr>
<tr>
<td></td>
<td>• Support for complex objects</td>
</tr>
<tr>
<td></td>
<td>• Inheritance</td>
</tr>
<tr>
<td></td>
<td>• Behavior</td>
</tr>
<tr>
<td>Extended Relational (Object/Relational)</td>
<td></td>
</tr>
</tbody>
</table>
Database Models and the Internet

- Characteristics of successful “Internet age” databases
  - Flexible, efficient, and secure Internet access that is easily used, developed, and supported
  - Support for complex data types and relationships
  - Seamless interfacing with multiple data sources and structures
Database Models and the Internet (continued)

– Relative conceptual simplicity to make database design and implementation less cumbersome

– An abundance of available database design, implementation, and application development tools

– A powerful DBMS graphical user interface (GUI) to help make the DBA’s job easier
Degrees of Data Abstraction

• Way of classifying data models

• Many processes begin at high level of abstraction and proceed to an ever-increasing level of detail

• Designing a usable database follows the same basic process
Degrees of Data Abstraction (continued)

• American National Standards Institute/Standards Planning and Requirements Committee (ANSI/SPARC)
  – Classified data models according to their degree of abstraction (1970s):
    • Conceptual
    • External
    • Internal
Data Abstraction Levels

**FIGURE 2.10 DATA ABSTRACTION LEVELS**

- **Conceptual Model**
  - Degree of Abstraction: High
  - Characteristics:
    - Hardware-independent
    - Software-independent
  - DBMS: Hierarchical & Network

- **Internal Model**
  - Degree of Abstraction: Medium
  - Characteristics:
    - Hardware-independent
    - Software-dependent

- **Physical Model**
  - Degree of Abstraction: Low
  - Characteristics:
    - Hardware-dependent
    - Software-dependent

- **External Model**
The Conceptual Model

• Represents global view of the database

• Enterprise-wide representation of data as viewed by high-level managers

• Basis for identification and description of main data objects, avoiding details

• Most widely used conceptual model is the entity relationship (ER) model
A Conceptual Model for Tiny College

FIGURE 2.12 A CONCEPTUAL MODEL FOR TINY COLLEGE
Advantages of Conceptual Model

• 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 either the hardware or the DBMS software have no effect on the database design at the conceptual level
The Internal Model

• Representation of the database as “seen” by the DBMS

• Adapts the conceptual model to the DBMS

• Software dependent

• Hardware independent
The External Model

- End users’ view of the data environment
- Requires that the modeler subdivide set of requirements and constraints into functional modules that can be examined within the framework of their external models
- Good design should:
  - Consider such relationships between views
  - Provide programmers with a set of restrictions that govern common entities
A Division of an Internal Model into External Models

**Figure 2.13** A Division of an Internal Model into External Models

- **COURSE**
  - generates
  - 1
- **STUDENT**
  - enrolls in
  - 1
- **PROFESSOR**
  - teaches
  - M
  - is used for
  - 1
  - ROOM
  - M
- **CLASS**
  - is found in
  - M
- **ENROLL**
  - M

---

Advantages of External Models

• Use of database subsets makes application program development much simpler
  – Facilitates designer’s task by making it easier to identify specific data required to support each business unit’s operations
  – Provides feedback about the conceptual model’s adequacy

• Creation of external models helps to ensure security constraints in the database design
The External Model

- DBMS dependent
- Hardware independent
The External Models for Tiny College

**Figure 2.14: The External Models for Tiny College**

**Student Registration**
- **STUDENT**
  - 1
  - enrolls in
  - M
  - ENROLL
  - M
  - is found in
  - 1
  - CLASS

  A student may take up to six classes per registration.

  A class is limited to 35 students.

**Class Scheduling**
- **ROOM**
  - 1
  - is used for
  - M
  - CLASS
  - M
  - teaches
  - 1
  - PROFESSOR

  A room may be used to teach many classes.

  Each class is taught in only one room. Each class is taught by one professor.

  A professor may teach up to three classes.
The Physical Model

• Operates at lowest level of abstraction, describing the way data are saved on storage media such as disks or tapes

• Software and hardware dependent

• Requires that database designers have a detailed knowledge of the hardware and software used to implement database design
# Levels of Data Abstraction

## Table 2.2: Levels of Data Abstraction

<table>
<thead>
<tr>
<th>MODEL</th>
<th>DEGREE OF ABSTRACTION</th>
<th>DATA MODEL</th>
<th>FOCUS</th>
<th>INDEPENDENT OF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptual</td>
<td>High</td>
<td>Entity</td>
<td>Global</td>
<td>Hardware and software</td>
</tr>
<tr>
<td>External</td>
<td></td>
<td>ER components</td>
<td>Subset</td>
<td>Hardware</td>
</tr>
<tr>
<td>Internal</td>
<td></td>
<td>Relational and others</td>
<td>Global</td>
<td>Hardware</td>
</tr>
<tr>
<td>Physical</td>
<td>Low</td>
<td>Physical storage methods</td>
<td>N/A</td>
<td>Neither hardware nor software</td>
</tr>
</tbody>
</table>
Summary

• A good DBMS will perform poorly with a poorly designed database

• A data model is a (relatively) simple abstraction of a complex real-world data-gathering environment

• Basic data modeling components are:
  – Entities
  – Attributes
  – Relationships
Summary (continued)

• Hierarchical model
  – Based on a tree structure composed of a root segment, parent segments, and child segments
  – Depicts a set of one-to-many (1:M) relationships between a parent and its children
  – Does not include ad hoc querying capability
Summary (continued)

• Network model attempts to deal with many of the hierarchical model’s limitations

• Relational model:
  – Current database implementation standard
  – Much simpler than hierarchical or network design

• Object is basic modeling structure of object oriented model

• Data modeling requirements are a function of different data views (global vs. local) and level of data abstraction