Data Models

Sonia Tomer
GITAM, Kablana
Objectives

• Why data models are important
• About the basic data-modeling building blocks
• How the major data models evolved
• How data models can be classified by level of abstraction
The Importance of Data Models

• Data models
  – Relatively simple representations, usually graphical, of complex real-world data structures
  – Facilitate interaction among the designer, the applications programmer, and the end user
The Importance of Data Models (continued)

• End-users have different views and needs for data
• Data model organizes data for various users
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
The Evolution of Data Models (continued)

- Hierarchical
- Network
- Relational
- Entity relationship
- Object oriented (OO)
The Hierarchical Model

• Developed in the 1960s to manage large amounts of data for complex manufacturing projects
• Basic logical structure is represented by an upside-down “tree”
The Hierarchical Model (continued)
The Hierarchical Model (continued)

• The hierarchical structure contains levels, or segments

• Depicts a set of one-to-many (1:M) relationships between a parent and its children segments
  – Each parent can have many children
  – each child has only one parent
The Hierarchical Model (continued)

• Advantages
  – Many of the hierarchical data model’s features formed the foundation for current data models
  – Its database application advantages are replicated, albeit in a different form, in current database environments
  – Generated a large installed (mainframe) base, created a pool of programmers who developed numerous tried-and-true business applications
The Hierarchical Model (continued)

• Disadvantages
  – Complex to implement
  – Difficult to manage
  – Lacks structural independence
  – Implementation limitations
  – Lack of standards
The Network Model

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

• Conference on Data Systems Languages (CODASYL)

• Database Task Group (DBTG)
The Network Model (continued)

- **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)**
  - Defines the environment in which data can be managed
The Network Model (continued)

- **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**
  - Works with the data in the database
The Network Model (continued)

• 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
The Network Model (continued)
The Network Model (continued)

- Disadvantages
  - Too cumbersome
  - The lack of ad hoc query capability put heavy pressure on programmers
  - Any structural change in the database could produce havoc in all application programs that drew data from the database
  - Many database old-timers can recall the interminable information delays
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 (continued)

• Relational Database Management System (RDBMS)

• Performs same basic functions provided by hierarchical and network DBMS systems, in addition to a host of other functions

• Most important advantage of the RDBMS is its ability to hide the complexities of the relational model from the user
The Relational Model (continued)

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

• Relational diagram
  – Representation of relational database’s entities, attributes within those entities, and relationships between those entities
The Relational Model (continued)

• 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
The Relational Model (continued)

**FIGURE 2.3** Linking relational tables

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

<table>
<thead>
<tr>
<th>AGENT_CODE</th>
<th>AGENT_LNAME</th>
<th>AGENT_FNAME</th>
<th>AGENT_INITIAL</th>
<th>AGENT_AREACODE</th>
<th>AGENT_PHONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>501</td>
<td>Alby</td>
<td>Alex</td>
<td>B</td>
<td>713</td>
<td>228-1249</td>
</tr>
<tr>
<td>502</td>
<td>Hahn</td>
<td>Leah</td>
<td>F</td>
<td>615</td>
<td>802-1244</td>
</tr>
<tr>
<td>503</td>
<td>Okon</td>
<td>John</td>
<td>T</td>
<td>615</td>
<td>123-5589</td>
</tr>
</tbody>
</table>

**Link through AGENT_CODE**

**Table name:** CUSTOMER

<table>
<thead>
<tr>
<th>CUS_CODE</th>
<th>CUS_LNAME</th>
<th>CUS_FNAME</th>
<th>CUS_INITIAL</th>
<th>CUS_AREACODE</th>
<th>CUS_PHONE</th>
<th>CUS_RENEW_DATE</th>
<th>AGENT_CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10010</td>
<td>Rames</td>
<td>Alfred</td>
<td>A</td>
<td>615</td>
<td>844-2573</td>
<td>05-Apr-2006</td>
<td>502</td>
</tr>
<tr>
<td>10011</td>
<td>Dunne</td>
<td>Leonia</td>
<td>K</td>
<td>713</td>
<td>894-1238</td>
<td>16-Jun-2005</td>
<td>501</td>
</tr>
<tr>
<td>10012</td>
<td>Smith</td>
<td>Kathy</td>
<td>V</td>
<td>615</td>
<td>894-2285</td>
<td>29-Jan-2007</td>
<td>502</td>
</tr>
<tr>
<td>10013</td>
<td>Olowski</td>
<td>Paul</td>
<td>F</td>
<td>615</td>
<td>894-2180</td>
<td>14-Oct-2006</td>
<td>502</td>
</tr>
<tr>
<td>10014</td>
<td>Orlando</td>
<td>Myron</td>
<td>B</td>
<td>615</td>
<td>222-1672</td>
<td>28-Dec-2005</td>
<td>501</td>
</tr>
<tr>
<td>10015</td>
<td>O’Brien</td>
<td>Amy</td>
<td>B</td>
<td>713</td>
<td>442-3381</td>
<td>22-Sep-2006</td>
<td>503</td>
</tr>
<tr>
<td>10016</td>
<td>Brown</td>
<td>James</td>
<td>G</td>
<td>615</td>
<td>297-1228</td>
<td>25-Mar-2006</td>
<td>502</td>
</tr>
<tr>
<td>10017</td>
<td>Williams</td>
<td>George</td>
<td>G</td>
<td>615</td>
<td>290-2556</td>
<td>17-Jul-2005</td>
<td>503</td>
</tr>
<tr>
<td>10018</td>
<td>Farriss</td>
<td>Anne</td>
<td>G</td>
<td>713</td>
<td>382-7185</td>
<td>03-Dec-2006</td>
<td>501</td>
</tr>
<tr>
<td>10019</td>
<td>Smith</td>
<td>Olette</td>
<td>K</td>
<td>615</td>
<td>297-3809</td>
<td>14-Mar-2006</td>
<td>503</td>
</tr>
</tbody>
</table>
The Relational Model (continued)
The Relational Model (continued)

• Rise to dominance due in part to its powerful and flexible query language
• Structured Query Language (SQL) allows the user to specify what must be done without specifying how it must be done
• SQL-based relational database application involves:
  – User interface
  – A set of tables stored in the database
  – SQL engine
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
The Entity Relationship Model (continued)

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 (continued)

**FIGURE 2.6** 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 Object Oriented Model

- Modeled both data and their relationships in a single structure known as an object
- Object-oriented data model (OODM) is the basis for the object-oriented database management system (OODBMS)
- OODM is said to be a semantic data model
The Object Oriented Model (continued)

• Object 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 all operations

• Object becomes basic building block for autonomous structures
The Object Oriented Model (continued)

• Object is an 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
The Object Oriented Model (continued)

FIGURE 2.7 A comparison of the OO model and the ER model

OO data model

INVOICE

INV_DATE
INV_NUMBER
INV_SHIP_DATE
INV_TOTAL

CUSTOMER

LINE

ER model

CUSTOMER
generates

INVOICE

INV_NUMBER
INV_DATE
INV_SHIP_DATE
INV_TOTAL

LINE
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
Database Models and the Internet

- Internet drastically changed role and scope of database market
- OODM and ERDM-O/RDM have taken a backseat to development of databases that interface with Internet
- Dominance of Web has resulted in growing need to manage unstructured information
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 comply with consistency and integrity characteristics of any data model
Data Models: A Summary (continued)

**FIGURE 2.8** The development of data models

<table>
<thead>
<tr>
<th>Semantics in Data Model</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hierarchical</strong></td>
<td>• Difficult to represent M:N relationships (hierarchical only)</td>
</tr>
<tr>
<td></td>
<td>• Structural level dependency</td>
</tr>
<tr>
<td></td>
<td>• No ad hoc queries (record-at-a-time access)</td>
</tr>
<tr>
<td></td>
<td>• Access path predefined (navigational access)</td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td>• Conceptual simplicity (structural independence)</td>
</tr>
<tr>
<td></td>
<td>• Provides ad hoc queries (SQL)</td>
</tr>
<tr>
<td></td>
<td>• Set-oriented access</td>
</tr>
<tr>
<td><strong>Relational</strong></td>
<td>• Easy to understand (more semantics)</td>
</tr>
<tr>
<td></td>
<td>• Limited to conceptual modeling (no implementation component)</td>
</tr>
<tr>
<td><strong>Entity Relationship</strong></td>
<td>• More semantics in data model</td>
</tr>
<tr>
<td></td>
<td>• Support for complex objects</td>
</tr>
<tr>
<td></td>
<td>• Inheritance (class hierarchy)</td>
</tr>
<tr>
<td></td>
<td>• Behavior</td>
</tr>
<tr>
<td></td>
<td>• Unstructured data (XML)</td>
</tr>
</tbody>
</table>

The figure illustrates the development of data models from least to most semantics, demonstrating the evolution of data modeling techniques and their associated features and limitations.
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 (ANSI) Standards Planning and Requirements Committee (SPARC)
  – Defined a framework for data modeling based on degrees of data abstraction (1970s):
    • External
    • Conceptual
    • Internal
Degrees of Data Abstraction (continued)
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
The External Model (continued)

• Advantages:
  – 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
  – Creation of external models helps to ensure security constraints in the database design
  – Simplifies application program development
The External Model (continued)

FIGURE 2.10 External models for Tiny College

Student Registration

A student may take up to six classes per registration.

- STUDENT
- ENROLL
- COURSE

is taken by

enrolls in

Each class is limited to 35 students.

Class Scheduling

A room may be used to teach many classes.

- ROOM
- CLASS
- PROFESSOR
- COURSE

is used for
teaches
generates

generates

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

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

- Represents global view of the entire database
- Representation of data as viewed by the entire organization
- Basis for identification and high-level description of main data objects, avoiding details
- Most widely used conceptual model is the entity relationship (ER) model
The Conceptual Model (continued)

![Conceptual Model for Tiny College](image-url)
The Conceptual Model (continued)

• 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 hardware or 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
- Maps the conceptual model to the DBMS
- Internal schema depicts a specific representation of an internal model
The Internal Model (continued)

FIGURE 2.12  An internal model for Tiny College

CONCEPTUAL MODEL

INTERNAL MODEL

Create Table PROFESSOR(
  PROF_ID      NUMBER PRIMARY KEY,
  PROF_LNAME   CHAR(15),
  PROF_INITIAL CHAR(1),
  PROF_FNAME   CHAR(15),
  ..........);

Create Table CLASS(
  CLASS_ID     NUMBER PRIMARY KEY,
  CRS_ID       CHAR(8) REFERENCES COURSE,
  PROF_ID      NUMBER REFERENCES PROFESSOR,
  ROOM_ID      CHAR(8) REFERENCES ROOM,
  ..........);

Create Table ROOM(
  ROOM_ID      CHAR(8) PRIMARY KEY,
  ROOM_TYPE    CHAR(3),
  ..........);

Create Table COURSE(
  CRS_ID       CHAR(8) PRIMARY KEY,
  CRS_NAME     CHAR(25),
  CRS_CREDITS  NUMBER,
  ..........);
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>
<thead>
<tr>
<th>MODEL</th>
<th>DEGREE OF ABSTRACTION</th>
<th>FOCUS</th>
<th>INDEPENDENT OF</th>
</tr>
</thead>
<tbody>
<tr>
<td>External</td>
<td>High</td>
<td>End-user views</td>
<td>Hardware and software</td>
</tr>
<tr>
<td>Conceptual</td>
<td></td>
<td>Global view of data</td>
<td>Hardware and software</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(independent of database model)</td>
<td></td>
</tr>
<tr>
<td>Internal</td>
<td>Low</td>
<td>Specific database model</td>
<td>Hardware</td>
</tr>
<tr>
<td>Physical</td>
<td></td>
<td>Storage and access methods</td>
<td>Neither hardware nor software</td>
</tr>
</tbody>
</table>
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
Summary (continued)

• Hierarchical model
  – Depicts a 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 popular graphical tool for data modeling that complements the relational model
Summary (continued)

• Object is basic modeling structure of object oriented data model

• The relational model has adopted many object-oriented extensions to become the extended relational data model (ERDM)

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