II. Data Models

Data Modeling and Data Models

- **Model**: Abstraction of a real-world object or event
- **Data modeling**: Iterative and progressive process of creating a specific data model for a specific problem domain
- **Data models**: Simple representations of complex real-world data structures
  - Useful for supporting a specific problem domain

Importance of Data Models

- As a communication tool
  - Present an overall view of the database
  - Organize subsets of data for various users
- Abstraction for the creation of a well-designed database
  - Check and re-check structure, constraints, etc.
- Make sure everyone is on same page

Data Model Basic Building Blocks

- **Entity**: Person, place, thing, event from the real world about which data will be collected
  - Each occurrence is unique and distinct
- **Entity set**: Collection of entities defining an object type
  - E.g., CUSTOMER is an entity set; “Kathy Smith” is an entity occurrence of the CUSTOMER type
  - Entity / Entity set often used synonymously
- **Attribute**: Characteristic of an entity
  - E.g., C_NAME, C_PHONE, C_ADDRESS, C_ZIP

Data Model Basic Building Blocks

- **Relationship**: Describes an association among entities
  - One-to-many (1:M)
  - Many-to-many (M:N or M:M)
  - One-to-one (1:1)
- **Constraint**: Set of rules to ensure data integrity
  - “A student’s GPA must be between 0.00 and 4.00”
  - Relationship rules are called cardinality constraints
  - Also modality constraints, which we’ll see later

Entity Set (and its attributes)

- Salesperson = a type of entity
Attributes

- Types of attributes
  - Simple/composite
    - E.g., LastName vs. Address
  - Single-valued/multi-valued
    - E.g., StudentID vs. Parents
  - Stored/derived
    - E.g., CreditHours vs. GPA

Relationships

- Relationship: association between entities
- Relationship type: set of relationships between entity sets
  - Same concept as “entities” and “entity sets”
  - Terms also often used interchangeably
- Different kinds:
  - Binary relationships
  - Unary relationships
  - Ternary relationships

Binary Relationships

- Simplest kind of relationship
- Relationship between two entity sets
- A salesperson “sells” products or products are sold by salespersons

One-to-One Binary Relationship

- 1-1
  - A single occurrence of one entity type can be associated with a single occurrence of the other entity type and vice versa.

One-to-Many Binary Relationship

- 1-M
  - “many” = the maximum number of occurrences that can be involved, means a number that can be 1, 2, 3, ... n

Many-to-Many Binary Relationship

- M-M
  - “many” can be either an exact number or have a known maximum
Unary Relationships

- Associate occurrences of an entity type with other occurrences of the same entity type.
- Cardinality:
  - One-to-One Unary Relationship
  - One-to-Many Unary Relationship
  - Many-to-Many Unary Relationship

Ternary Relationship

- Involves three different entity types.

Modality Constraints

- Participation constraint
  - The minimum number of entity occurrences that can be involved in a relationship.
- Must vs. May
  - “A student may be enrolled in multiple sections”
  - “A section must have one professor”

Business Rules

- Brief, precise, and unambiguous description of a policy, procedure, or principle
- Describe main and distinguishing characteristics of the data
- Enable defining the basic building blocks of the data model by allowing the designer to:
  - Understand the nature, role, scope of data, and business processes
  - Develop appropriate relationship participation rules and constraints
  - Create an accurate data model

Consider a generic corporate database

- Department
  - Name, number, manager (employee), start date of manager
  - Projects controlled by department
    - Name, number, description
  - Employees
    - Name, SSN, address, salary, gender, birthday, age, phone
    - Assigned to a department, possibly multiple projects
  - Dependents of employees
- NOTE: entity sets, attributes, relationship types, constraints

Where does the DB designer get these building blocks from?
Sources of Business Rules

- Company managers
- Policy makers
- Department managers

Written documentation
Direct interviews with end users

Translating Business Rules into Data Model Components

- 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 names - Required to:
  - Be descriptive of the objects in the business environment
  - Use terminology that is familiar to the users
- Attribute names - Required to be descriptive of the data represented by the attribute
- Proper naming:
  - Facilitates communication between parties
  - Promotes self-documentation

The Evolution of Data Models

Standard Database Concepts

- Schema
  - Conceptual organization of the entire database as viewed by the database administrator
- Subschema
  - Portion of the database seen by the application programs that produce the desired information from the data within the database

Standard Database Concepts

- Data manipulation language (DML)
  - Environment in which data can be managed and is used to work with the data in the database
- Data definition language (DDL)
  - Enables the database administrator to define the schema components
The Network Model
- Pioneered standard concepts from previous slides
- Separate files are linked via physical links (link fields)

![Physical Links to Relate Files in the Network Model](image)

The Relational Model
- Separate files are linked via logical links (relationships) in the data model

![Logical Links to Relate Files in the Relational Model](image)

Relational Model
**Advantages**
- Ad hoc query capability based on SQL
- Isolates the end user from physical-level details
- Improves implementation and management simplicity

**Disadvantages**
- Requires substantial hardware and system software overhead
- Conceptual simplicity gives untrained people the tools to use a good system poorly

| More flexible | Less efficient |

![A Relational Diagram](image)

Entity-Relationship (ER) Model
- A design technique using graphical representations to model the building blocks
- Used in conjunction with the relational model
- Diagrams entities sets (with attributes) and the relationship between the entity sets
- Recall previous definitions
- **ER diagrams (ERDs)** are the *deliverables* of the design phase of information system development
- There are many variations of ER diagrams in use

![ER Model Notations](image)
Hybrid Notation

Entity Relationship Model

Advantages
- Visual modeling yields conceptual simplicity
- Visual representation makes it an effective communication tool
- Is integrated with the dominant relational model

Disadvantages
- No data manipulation language
- Loss of information content occurs when attributes are removed from entities to avoid crowded displays
- Limited constraint representation
- Limited relationship representation

ER diagram for Company

The Object-Oriented Data Model (OODM) or Semantic Data Model

Object: Contains data and relationships with operations that are performed on it
- Basic building block for autonomous structures
- Abstraction of real-world entity
- Attributes - Describe the properties of an object

The Object-Oriented Data Model (OODM)

- Class: Collection of similar objects with shared structure and behavior organized in a class hierarchy
  - Class hierarchy: Resembles an upside-down tree in which each class has only one parent
  - Inheritance: Object inherits methods and attributes of parent class
- Unified Modeling Language (UML)
  - Describes sets of diagrams and symbols to graphically model a system

Object-Oriented Model

Advantages
- Semantic content is added
- Visual representation includes semantic content
- Inheritance promotes data integrity

Disadvantages
- Slow development of standards caused vendors to supply their own enhancements
- Compromised widely accepted standard
- Complex navigational system
- Learning curve is steep
- High system overhead slows transactions
Comparison of OO, UML, and ER Models

Object/Relational and XML

- **Extended relational data model (ERDM)**
  - Supports OO features and complex data representation
  - Represented by Extended ER Diagrams
- **Extensible Markup Language (XML)**
  - Manages unstructured data for efficient and effective exchange of all data types

Extended ER diagram

- Specialization, generalization, inheritance

Multiple subclass hierarchies

Big Data

- Aims to:
  - Find new and better ways to manage large amounts of web and sensor-generated data
  - Provide high performance and scalability at a reasonable cost
- Characteristics
  - Volume
    - Doesn’t allow usage of conventional structures
  - Velocity
    - Almost impossible to maintain data integrity
  - Variety
    - OLAP tools inconsistent in dealing with unstructured data

Big Data New Technologies

- Hadoop
- Hadoop Distributed File System (HDFS)
- MapReduce
- NoSQL
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
- Store data in key-value stores

NoSQL

**Advantages**

- High scalability, availability, and fault tolerance are provided
- Uses low-cost commodity hardware
- Supports Big Data
- Key-value model improves storage efficiency

**Disadvantages**

- Complex programming is required
- There is no relationship support
- There is no transaction integrity support
- In terms of data consistency, it provides an eventually consistent model

Simple Key-value Representation

<table>
<thead>
<tr>
<th>Data Abstraction Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Model</td>
</tr>
<tr>
<td>External Model</td>
</tr>
<tr>
<td>Logical Independence</td>
</tr>
<tr>
<td>Physical Independence</td>
</tr>
</tbody>
</table>

The External Model

- End users’ view of the data environment
- ER diagrams are used to represent the external views
- **External schema**: Specific representation of an external view

External Models for Tiny College

- **Student Registration**
  - A student may take up to six classes per semester.
- **Class Scheduling**
  - A course may be taught by more than one professor.
  - Each class to be taught by one professor.
The Conceptual Model
- Represents a global view of the entire database by the entire organization
  - **Conceptual schema**: Basis for the identification and high-level description of the main data objects
  - Is software and hardware independent
- **Logical design**: Task of creating a conceptual data model
- ER diagrams also used to represent the conceptual model

The Internal Model
- Representing database as seen by the DBMS
  - Involves mapping conceptual model to the DBMS
- **Internal schema**: Specific representation of an internal model
  - Uses the database constructs supported by the chosen database
  - Software dependent and hardware independent
  - Makes a difference if you use Oracle, or MySQL, etc., but not what it’s running on

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

Levels of Data Abstraction

---

Cengage Learning © 2015