ITCS 3160
DATA BASE DESIGN AND IMPLEMENTATION
JING YANG
2010 FALL

Outline

- Introduction
- An Example
- Characteristics of the Database Approach
- Actors on the Scene
- Workers behind the Scene
- Advantages of Using the DBMS Approach
- A Brief History of Database Applications
- When Not to Use a DBMS

Fundamentals of Database Systems Elmasri and Navathe
Question for fun:

- Did you do anything related to a database in recent two days? If so, what was it?

Overview

- **Traditional database applications**
  - Store textual or numeric information

- **Multimedia databases**
  - Store images, audio clips, and video streams digitally

- **Geographic information systems (GIS)**
  - Store and analyze maps, weather data, and satellite images
Overview (cont'd.)

- **Data warehouses and online analytical processing (OLAP) systems**
  - Extract and analyze useful business information from very large databases
  - Support decision making

- **Real-time and active database technology**
  - Control industrial and manufacturing processes

What is a database

- **A database** is a collection of related data
  - **Data** - Known facts that can be recorded and that have implicit meaning

- **Properties of a database:**
  - Represents some aspect of the real world, called **Miniworld** or **universe of discourse (UoD)**
  - Logically coherent collection of data with inherent meaning
  - Built for a specific purpose

- **Example:** Amazon.com (over 2 terabytes)
What is a database management system (DBMS)

- A DBMS is a collection of programs that enables users to create and maintain a database

Four main functions provided by a DBMS:

- **Defining a database**
  - Specify the data types, structures, and constraints of the data to be stored
  - **Meta-data**
    - Database definition or descriptive information
    - Stored by the DBMS in the form of a database catalog or dictionary

- **Constructing a database**
  - Store data on storage medium controlled by the DBMS

- **Manipulating a database**
  - Query and update the database miniworld
    - **Application program** accesses database by sending queries to DBMS
    - **Query** causes some data to be retrieved
  - Generate reports

- **Sharing a database**
  - Allow multiple users and programs to access the database simultaneously
What is a database management system (DBMS) (cont'd. 2)

Other functions:
- **Protection** includes:
  - System protection
  - Security protection
- **Maintain** the database system
  - Allow the system to evolve as requirements change over time

A database system is the database and DBMS software together

Figure 1.1
A simplified database system environment.
An example

- UNIVERSITY database
  - Information concerning students, courses, and grades in a university environment

- Data records
  - STUDENT
  - COURSE
  - SECTION
  - GRADE_REPORT
  - PREREQUISITE

Fundamentals of Database Systems
Elmasri and Navathe

---

### STUDENT

<table>
<thead>
<tr>
<th>Name</th>
<th>Student_number</th>
<th>Class</th>
<th>Major</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>17</td>
<td>1</td>
<td>CS</td>
</tr>
<tr>
<td>Brown</td>
<td>18</td>
<td>2</td>
<td>CS</td>
</tr>
</tbody>
</table>

### COURSE

<table>
<thead>
<tr>
<th>Course_name</th>
<th>Course_number</th>
<th>Credit</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intro to Computer Science</td>
<td>CSI310</td>
<td>4</td>
<td>CS</td>
</tr>
<tr>
<td>Data Structures</td>
<td>CSE330</td>
<td>4</td>
<td>CS</td>
</tr>
<tr>
<td>Discrete Mathematics</td>
<td>MATH2410</td>
<td>3</td>
<td>MATH</td>
</tr>
<tr>
<td>Database</td>
<td>CSE380</td>
<td>2</td>
<td>CS</td>
</tr>
</tbody>
</table>

### SECTION

<table>
<thead>
<tr>
<th>Section_identifier</th>
<th>Course_number</th>
<th>Semester</th>
<th>Year</th>
<th>Instructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CSE110</td>
<td>Fall</td>
<td>07</td>
<td>King</td>
</tr>
<tr>
<td>2</td>
<td>CSE120</td>
<td>Fall</td>
<td>07</td>
<td>Andrus</td>
</tr>
<tr>
<td>3</td>
<td>MATH2410</td>
<td>Fall</td>
<td>06</td>
<td>Chang</td>
</tr>
<tr>
<td>4</td>
<td>CSE120</td>
<td>Fall</td>
<td>08</td>
<td>Andrus</td>
</tr>
<tr>
<td>5</td>
<td>CSE330</td>
<td>Fall</td>
<td>08</td>
<td>Stone</td>
</tr>
</tbody>
</table>

### GRADE_REPORT

<table>
<thead>
<tr>
<th>Student_number</th>
<th>Section_identifier</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>112</td>
<td>B</td>
</tr>
<tr>
<td>17</td>
<td>119</td>
<td>C</td>
</tr>
<tr>
<td>8</td>
<td>85</td>
<td>A</td>
</tr>
<tr>
<td>9</td>
<td>92</td>
<td>A</td>
</tr>
<tr>
<td>8</td>
<td>102</td>
<td>B</td>
</tr>
<tr>
<td>8</td>
<td>130</td>
<td>A</td>
</tr>
</tbody>
</table>

### PREREQUISITE

<table>
<thead>
<tr>
<th>Course_number</th>
<th>Prerequisite_number</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSE330</td>
<td>CSE320</td>
</tr>
<tr>
<td>CSE330</td>
<td>MATH2410</td>
</tr>
<tr>
<td>CSE320</td>
<td>CSE1310</td>
</tr>
</tbody>
</table>

Figure 1.2
A database that stores student and course information.
Define University database

- Specify structure of records of each file by specifying data type for each data element
  - String of alphabetic characters
  - Integer
  - Etc.

Construct University database

- Store data to represent each student, course, section, grade report, and prerequisite as a record in appropriate file
- Notice the records in the files may be related: Relationships among the records
Manipulate University database

- Queries and updates
- Examples of queries:
  - Retrieve the transcript
  - List the names of students who took the section of the ‘Database’ course offered in fall 2008 and their grades in that section
  - List the prerequisites of the ‘Database’ course

Manipulate University database (cont’d.)

- Examples of updates:
  - Change the class of ‘Smith’ to sophomore
  - Create a new section for the ‘Database’ course for this semester
  - Enter a grade of ‘A’ for ‘Smith’ in the ‘Database’ section of last semester
Phases for designing a database

- Requirements specification and analysis
- Conceptual design
- Logical design
- Physical design

 Characteristics of database approach

- Traditional file processing
  - Each user defines and implements the files needed for a specific software application
- Database approach
  - Single repository maintains data that is defined once and then accessed by various users
Characteristics of database approach (cont'd.)

- Main characteristics of database approach
  - Self-describing nature of a database system
  - Insulation between programs and data, and data abstraction
  - Support of multiple views of the data
  - Sharing of data and multiuser transaction processing

Self-describing nature of a database system

- Database system contains complete definition of structure and constraints
- **Meta-data**: information stored in the DBMS catalog
  - Describes structure of the database
- Catalog used by:
  - DBMS software
  - Database users who need information about database structure
Insulation between programs and data

- **Program-data independence**
  - Structure of data files is stored in DBMS catalog separately from access programs

- **Program-operation independence**
  - **Operations** specified in two parts:
    - Interface includes operation name and data types of its arguments
    - Implementation can be changed without affecting the interface
Data abstraction

- **Data abstraction**
  - Allows program-data independence and program-operation independence

- **Conceptual representation of data**
  - Does not include details of how data is stored or how operations are implemented

- **Data model**
  - Type of data abstraction used to provide conceptual representation

Support of multiple views of data

- **View**
  - Subset of the database
  - Contains *virtual data* derived from the database files but is not explicitly stored

- **Multiuser DBMS**
  - Users have a variety of distinct applications
  - Must provide facilities for defining multiple views
Sharing of data and multiuser transaction processing

- Allow multiple users to access the database at the same time
- **Concurrency control software**
  - Ensure that several users trying to update the same data do so in a controlled manner
    - Result of the updates is correct
- **Online transaction processing (OLTP) application**
Sharing of Data and Multiuser Transaction Processing (cont’d.)

- **Transaction**
  - Central to many database applications
  - Executing program or process that includes one or more database access, such as reading or updating
- **Isolation** property
  - Each transaction appears to execute in isolation from other transactions
- **Atomicity** property
  - Either all the database operations in a transaction are executed or none are

Actors on the scene

- **Database administrators (DBA)** are responsible for:
  - Authorizing access to the database
  - Coordinating and monitoring its use
  - Acquiring software and hardware resources
- **Database designers** are responsible for:
  - Identifying the data to be stored
  - Choosing appropriate structures to represent and store this data
Actors on the scene (cont'd.)

- **End users**
  - People whose jobs require access to the database
  - Types
    - Casual end users
    - Naive or parametric end users
    - Sophisticated end users
    - Standalone users

Actors on the scene (cont'd.)

- **System analysts**
  - Determine requirements of end users

- **Application programmers**
  - Implement these specifications as programs
Workers behind the Scene

- **DBMS system designers and implementers**
  - Design and implement the DBMS modules and interfaces as a software package

- **Tool developers**
  - Design and implement tools

- **Operators and maintenance personnel**
  - Responsible for running and maintenance of hardware and software environment for database system

Advantages of using DBMS approach

- **Controlling redundancy**
  - **Data normalization**
    - Store each logical data item in only one place
    - Ensure consistency and save storage
  - **Denormalization**
    - Sometimes necessary to use controlled redundancy to improve the performance of queries
Advantages of using DBMS approach (cont'd.)

- Restricting unauthorized access
  - Security and authorization subsystem
  - Privileged software

Advantages of using DBMS approach (cont'd.)

- Providing **persistent** storage for program objects
  - Complex object in C++ can be stored permanently in an object-oriented DBMS

- **Impedance mismatch problem**
  - Object-oriented database systems typically offer data structure compatibility
Advantages of using DBMS approach (cont'd.)

- Providing storage structures and search techniques for efficient query processing
  - **Indexes**
    - Auxiliary files to speed up disk search
    - Based on modified tree data structure or hash data structures
  - **Buffering and caching**
  - **Query processing and optimization**

Advantages of using DBMS approach (cont'd.)

- Providing backup and recovery
  - **Backup and recovery subsystem** of the DBMS is responsible for recovery
- Providing multiple user interfaces
  - **Graphical user interfaces (GUIs)**
- Representing complex relationships among data
  - May include numerous varieties of data that are interrelated in many ways
Advantages of using DBMS approach (cont'd.)

- Enforcing **integrity constraints**
  - **Referential integrity** constraint
    - Every section record must be related to a course record
  - **Key or uniqueness** constraint
    - Every course record must have a unique value for Course_number
  - **Business rules**
  - **Inherent rules** of the data model

Advantages of using DBMS approach (cont'd.)

- Permitting inferencing and actions using rules
  - **Deductive database systems**
    - Provide capabilities for defining deduction rules
    - Inferencing new information from the stored database facts
  - **Trigger**
    - Rule activated by updates to the table
  - **Stored procedures**
    - More involved procedures to enforce rules
    - Invoked when certain conditions are met
Advantages of using DBMS approach (cont'd.)

- Additional implications of using the database approach
  - Reduced application development time
  - Flexibility
  - Availability of up-to-date information
  - Economies of scale

A brief history of database applications

- Early database applications using hierarchical and network systems
  - Large numbers of records of similar structure
  - Problem: intermixing of conceptual relationships with physical storage and placement of records on disk
- Providing data abstraction and application flexibility with relational databases
  - Separates physical storage of data from its conceptual representation
  - Provides a mathematical foundation for data representation and querying
A brief history of database applications (cont’d.)

- Object-oriented applications and the need for more complex databases
  - Used in specialized applications: engineering design, multimedia publishing, and manufacturing systems
- Interchanging data on the Web for e-commerce using XML
  - Extended markup language (XML) primary standard for interchanging data among various types of databases and Web pages

- Extending database capabilities for new applications
  - Extensions to better support specialized requirements for applications
  - Scientific applications, images, videos, video clips, data mining applications, spatial applications, time series applications
  - Enterprise resource planning (ERP)
    - Integrated computer-based system used to manage internal and external resources
    - Facilitate information flow between all business functions inside an organization and manage connections to outside stakeholders
    - Consolidate all business operations into a uniform and enterprise-wide system environment
  - Customer relationship management (CRM)
    - Goal: new, existing, and former customizers
    - Organize, automate, and synchronize business processes—sales activities, marketing, customer service, and technical support
A brief history of database applications (cont'd.)

- Databases versus information retrieval
  - Information retrieval (IR)
    - Deals with books, manuscripts, and various forms of library-based articles
    - Searching, categorization, topic modeling, etc.
    - Use database and IR techniques together to retrieve information on the Web

---

When not to use a DBMS

- More desirable to use regular files for:
  - Simple, well-defined database applications not expected to change at all
  - Stringent, real-time requirements that may not be met because of DBMS overhead
  - Embedded systems with limited storage capacity
  - No multiple-user access to data
Summary

- **Database**
  - Collection of related data (recorded facts)
- **DBMS**
  - Generalized software package for implementing and maintaining a computerized database
- **Several categories of database users**
- **Database applications have evolved**
  - Current trends: IR, Web

After class exercises

- 1.1, 1.2, 1.3, 1.9, 1.12, 1.14
- Reading: Chapter 1