Summary on Chapter 1

Databases and Database Users

1. Introduction

1.1 Types of Database Applications

- Traditional Applications:
  - Numeric and Textual Databases in Business Applications
- More Recent Applications:
  - Multimedia Databases (images, videos, voice, etc.)
  - Geographic Information Systems (GIS)
  - Data Warehouses
  - Real-time and Active Databases
  - Many other applications such as Document oriented applications

- Initial part of book focuses on traditional applications
- A number of recent applications are described later in the book (for example, Chapters 26, 27, 28, 29)

1.2 Basic Definitions

- **Database:**
  - A collection of related data.
- **Data:**
  - Known facts that can be recorded and have an implicit meaning.
- **Mini-world (domain of discourse or universe of discourse (UoD)):**
  - Some part of the real world about which data is stored in a database. For example, student registration, grades and transcripts at a university.
- **Database Management System (DBMS):**
  - A software package/system to facilitate the creation and maintenance of a computerized database.
- **Database System:**
  - The DBMS software together with the data itself. Sometimes, the application programs and interfaces are also included.

Simplified database system environment (see Figure 1.1)
1.3 Typical DBMS Functionality

- **Define** a particular database in terms of its data types, structures, and constraints
- **Construct** or Load the initial database contents on a secondary storage medium (typically hard disk)
- **Manipulating** the database:
  - Retrieval: Querying, generating reports
  - Modification: Insertions, deletions and updates to its content
  - Accessing/changing the database through Web applications
- **Processing** and **Sharing** by a set of concurrent users and application programs – yet, keeping all data valid and consistent

- **Other features**:
  - Protection or Security measures to prevent unauthorized access
  - “Active” processing to take internal actions on data
  - Presentation and Visualization of data
  - Maintaining the database and associated programs over the lifetime of the database application
    - Called database, software, and system life-cycle maintenance
2. An Example
A Database of UNIVERSITY Application

- **Mini-world (UoD) for the example:**
  - Part of a UNIVERSITY environment.
- **Some mini-world entities:**
  - STUDENTs
  - COURSEs
  - SECTIONs (of COURSEs)
  - (academic) DEPARTMENTs
  - INSTRUCTORs
- **Some mini-world relationships:**
  - SECTIONs are of specific COURSEs
  - STUDENTs take SECTIONs
  - COURSEs have prerequisite COURSEs
  - INSTRUCTORs teach SECTIONs
  - COURSEs are offered by DEPARTMENTs
  - STUDENTs major in DEPARTMENTs
- Note: The above entities and relationships are typically expressed in a conceptual data model, such as the ENTITY-RELATIONSHIP (ER) data model (see Chapters 7, 8)

<table>
<thead>
<tr>
<th>STUDENT</th>
<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>
<td></td>
</tr>
<tr>
<td>Brown</td>
<td>8</td>
<td>2</td>
<td>CS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE</th>
<th>Course_name</th>
<th>Course_number</th>
<th>Credit_hours</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intro to Computer Science</td>
<td>CS1310</td>
<td>4</td>
<td>CS</td>
<td></td>
</tr>
<tr>
<td>Data Structures</td>
<td>CS3320</td>
<td>4</td>
<td>CS</td>
<td></td>
</tr>
<tr>
<td>Discrete Mathematics</td>
<td>MATH2410</td>
<td>3</td>
<td>MATH</td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td>CS3380</td>
<td>3</td>
<td>CS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SECTION</th>
<th>Section_identifier</th>
<th>Course_number</th>
<th>Semester</th>
<th>Year</th>
<th>Instructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>85</td>
<td>MATH2410</td>
<td>Fall</td>
<td>07</td>
<td>King</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>CS1310</td>
<td>Fall</td>
<td>07</td>
<td>Anderson</td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>CS3320</td>
<td>Spring</td>
<td>08</td>
<td>Knuth</td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>MATH2410</td>
<td>Fall</td>
<td>08</td>
<td>Chang</td>
<td></td>
</tr>
<tr>
<td>119</td>
<td>CS1310</td>
<td>Fall</td>
<td>08</td>
<td>Anderson</td>
<td></td>
</tr>
<tr>
<td>135</td>
<td>CS3380</td>
<td>Fall</td>
<td>08</td>
<td>Stone</td>
<td></td>
</tr>
</tbody>
</table>
3. Main Characteristics of the Database Approach

3.1 Self-describing nature of a database system

- A DBMS catalog stores the description of a particular database (e.g. data structures, types, and constraints)
- The description is called meta-data.
- This allows the DBMS software to work with different database applications (university, bank, airlines, etc.)

3.2 Insulation between programs and data

- Called program-data independence.
- Allows changing data structures and data storage organization without having to change the DBMS access programs.
- Accomplished through data abstraction
- A data model is used to hide storage details and present the users with a conceptual view of the database.
- Programs refer to the data model constructs rather than data storage details

3.3 Support of multiple views of the data

- Each user may see a different view of the database, which describes only the data of interest to that user.

3.4 Sharing of data and multi-user transaction processing

- Allowing a set of user transactions to access and update the database concurrently (at the same time).
- Concurrency control within the DBMS guarantees that each transaction is correctly executed or aborted
- Recovery subsystem ensures each completed transaction has its effect permanently recorded in the database
OLTP (Online Transaction Processing) is a major part of database applications (allows hundreds of concurrent transactions to execute per second)

<table>
<thead>
<tr>
<th>Relation_name</th>
<th>No_of_columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>STUDENT</td>
<td>4</td>
</tr>
<tr>
<td>COURSE</td>
<td>4</td>
</tr>
<tr>
<td>SECTION</td>
<td>5</td>
</tr>
<tr>
<td>GRADE_REPORT</td>
<td>3</td>
</tr>
<tr>
<td>PREREQUISITE</td>
<td>2</td>
</tr>
</tbody>
</table>

**RELATIONS**

4. **Actors on the Scene**

Those who actually use and control the database contents and those who design, develop and maintain database applications (called “Actors on the Scene”).

- Database administrators:
  - Responsible for authorizing/controlling access to the database; coordinating and monitoring its use; acquiring software and hardware resources; and monitoring efficiency of operations.

- Database Designers:
  - Responsible for defining database structure, constraints, and transactions; communicate with users to understand their needs.

- End Users
  - Casual end users: occasionally access the DB, they may need different information each time. Use application, or browser, or SQL. Mid or high-level managers.
  - Naïve or parametric end users: constantly querying and updating the DB, using standard types of queries and updates (called canned transactions).
    - (E.g.,) Bank tellers check account balances and post withdrawals and deposits. Reservation agents for airlines, hotels, rent cars, etc., check availability for a given request and make reservations.
  - Sophisticated end users: engineers, scientists, business analysts and others implement their own applications based on well understood DBMS facilities.
  - Standalone users: maintain personal DBs by using ready-made program packages that provide easy to use User Interface.
5. Workers behind the scene

- Those who design and develop the DBMS software and related tools, and the computer systems operators.
  - DBMS system designers and implementers
  - Tool developers
  - Operators and maintenance personnel

6. Advantages of Using the Database Approach

- Controlling redundancy in data storage and in development and maintenance efforts.
  - Sharing of data among multiple users.
- Restricting unauthorized access to data.
- Providing persistent storage for program Objects
- Providing Storage Structures (e.g. indexes) for efficient Query Processing
- Providing backup and recovery services.
- Providing multiple interfaces to different classes of users.
- Representing complex relationships among data.
- Enforcing integrity constraints on the database.
- Drawing inferences and actions from the stored data using deductive and active rules
- Allowing multiple “views” of the same data (see next slide, Figure 1.5 from textbook)

**TRANSCRIPT**

<table>
<thead>
<tr>
<th>Student_name</th>
<th>Course_number</th>
<th>Grade</th>
<th>Semester</th>
<th>Year</th>
<th>Section_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>CS1310</td>
<td>C</td>
<td>Fall</td>
<td>08</td>
<td>119</td>
</tr>
<tr>
<td></td>
<td>MATH2410</td>
<td>B</td>
<td>Fall</td>
<td>08</td>
<td>112</td>
</tr>
<tr>
<td>Brown</td>
<td>MATH2410</td>
<td>A</td>
<td>Fall</td>
<td>07</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>CS1310</td>
<td>A</td>
<td>Fall</td>
<td>07</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>CS3320</td>
<td>B</td>
<td>Spring</td>
<td>08</td>
<td>102</td>
</tr>
<tr>
<td></td>
<td>CS3380</td>
<td>A</td>
<td>Fall</td>
<td>08</td>
<td>135</td>
</tr>
</tbody>
</table>

**COURSE_PREREQUISITES**

<table>
<thead>
<tr>
<th>Course_name</th>
<th>Course_number</th>
<th>Prerequisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>CS3380</td>
<td>CS3320</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MATH2410</td>
</tr>
<tr>
<td>Data Structures</td>
<td>CS3320</td>
<td>CS1310</td>
</tr>
</tbody>
</table>

Figure 1.5
Two views derived from the database in Figure 1.2. (a) The TRANSCRIPT view.
(b) The COURSE_PREREQUISITES view.

- Potential for enforcing standards:
  - Crucial for the success of database applications in large organizations.
  - Standards refer to data item names, display formats, screens, report structures, meta-data (description of data), Web page layouts, etc.
- Reduced application development time:
  - The time needed to add each new application is reduced.
- Flexibility to change data storage structures:
Storage structures may evolve to improve performance, or because of new requirements.

- Availability of up-to-date information:
  - Extremely important for on-line transaction systems such as airline, hotel, car reservations.

- Economies of scale:
  - Wasteful overlap of resources and personnel can be avoided by consolidating data and applications across departments.

### 7. A Brief History of Database Applications

- **Early Database Applications:**
  - The Hierarchical and Network Models were introduced in mid 1960s and dominated during the seventies.
  - Some worldwide database processing still occurs using these models; particularly, the hierarchical model.

- **Relational Model based Systems:**
  - Relational model was introduced in 1970, and heavily researched and experimented with at IBM Research and several universities.
  - Relational DBMS Products emerged in the early 1980s and now dominate the market.

- **Object-oriented and emerging applications:**
  - Object Databases (ODBs) were introduced in late 1980s and early 1990s to cater to the need of complex data and applications, and the proliferation of object-oriented programming languages.
    - Their use has not taken off much.
  - Many relational DBMSs have incorporated object database concepts, leading to a new category called *object-relational* databases (ORDBs) (see Ch. 11)
  - *Extended relational* systems add further capabilities (e.g. for multimedia data, XML, spatial, and other data types)

- **Data on the Web and E-commerce Applications:**
  - *Static* Web pages often specified in HTML (Hypertext markup language) with links among pages.
  - *Dynamic* Web pages have portions of their content extracted from databases, and allow user interaction with databases by typing in form boxes.
    - Java EE6 based applications (+ Java Script/Ajax)
    - .Net application framework
    - Script programming language such as PHP and Ruby

- **Extending Database Capabilities**
  - New functionality is being added to DBMSs in the following areas:
  - Scientific Applications
  - XML (eXtensible Markup Language)
  - Image Storage and Management
  - Audio and Video Data Management
o Data Warehousing and Data Mining
o Spatial Data Management and Geographic Information Systems
o Time Series and Historical Data Management
o Collecting and fusing data from distributed sensors
o The above led to new research and development in incorporating new data types, complex data structures, new operations/query languages, and new storage and indexing schemes (see Chapter 26).

8. When Not to Use a DBMS

- Main inhibitors (costs) of using a DBMS:
  - High initial investment and possible need for additional hardware.
  - Overhead for providing generality, security, concurrency control, recovery, and other functions.

- When a DBMS may be unnecessary:
  - If the database and applications are simple, well defined, and not expected to change.
  - If there are stringent real-time requirements that may not be met because of DBMS overhead.
  - If access to data by multiple users is not required.