Database Management Systems

1. A **database management system** (DBMS), or simply a **database system** (DBS), consists of
   - A collection of interrelated and persistent data (usually referred to as the **database** (DB)).
   - A set of application programs used to access, update and manage that data (which form the data management system (MS)).

2. The goal of a DBMS is to provide an environment that is both **convenient** and **efficient** to use in
   - Retrieving information from the database.
   - Storing information into the database.

3. Databases are usually designed to manage **large** bodies of information. This involves
   - Definition of structures for information storage (data modeling).
   - Provision of mechanisms for the manipulation of information (file and systems structure, query processing).
   - Providing for the safety of information in the database (crash recovery and security).
   - Concurrency control if the system is shared by users.

Purpose of Database Systems

- Difficulty in accessing data
  - May have to write a new application program to satisfy an unusual request.
  - E.g. find all customers with the same postal code.
  - Could generate this data manually, but a long job...
- Data isolation
  - Data in different files.
  - Data in different formats.
  - Difficult to write new application programs.
- Multiple users
  - Want concurrency for faster response time.
  - Need protection for concurrent updates.
  - E.g. two customers withdrawing funds from the same account at the same time - account has $500 in it, and they withdraw $100 and $50. The result could be $350, $400 or $450 if no protection.
- Security problems
• Every user of the system should be able to access only the data they are permitted to see.
• E.g. payroll people only handle employee records, and cannot see customer accounts; tellers only access account data and cannot see payroll data.
• Difficult to enforce this with application programs.
  - Integrity problems
    • Data may be required to satisfy constraints.
    • E.g. no account balance below $25.00.
    • Again, difficult to enforce or to change constraints with the file

Data Abstraction

1. The major purpose of a database system is to provide users with an abstract view of the system.

   The system hides certain details of how data is stored and created and maintained

   Complexity should be hidden from database users.

2. There are several levels of abstraction:
   1. Physical Level:
      • How the data are stored.
      • E.g. index, B-tree, hashing.
      • Lowest level of abstraction.
      • Complex low-level structures described in detail.
   2. Conceptual Level:
      • Next highest level of abstraction.
      • Describes what data are stored.
      • Describes the relationships among data.
      • Database administrator level.
   3. View Level:
      • Highest level.
      • Describes part of the database for a particular group of users.
      • Can be many different views of a database.
      • E.g. tellers in a bank get a view of customer accounts, but not of payroll data.
Data Models

1. **Data models** are a collection of conceptual tools for describing data, data relationships, data semantics and data constraints. There are three different groups:
   1. Object-based Logical Models.
   2. Record-based Logical Models.

We’ll look at them in more detail now.

The Relational Model

- Data and relationships are represented by a collection of **tables**.
- Each **table** has a number of columns with unique names, e.g., *customer, account*.
- Figure 1.3 shows a sample relational database.

```
<table>
<thead>
<tr>
<th>name</th>
<th>street</th>
<th>city</th>
<th>number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowry</td>
<td>Maple</td>
<td>Queens</td>
<td>900</td>
</tr>
<tr>
<td>Shier</td>
<td>North</td>
<td>Bronx</td>
<td>555</td>
</tr>
<tr>
<td>Shier</td>
<td>North</td>
<td>Bronx</td>
<td>617</td>
</tr>
<tr>
<td>Hodges</td>
<td>Sidhill</td>
<td>Brooklyn</td>
<td>801</td>
</tr>
</tbody>
</table>
```

The Network Model

- Data are represented by collections of records.
- Relationships among data are represented by links.
- Organization is that of an **arbitrary graph**.
- Figure 1.4 shows a sample network database that is the equivalent of the relational database of Figure 1.3.

```
Lowery         Maple     Queens
    v
   /|
  /  |
Shier          North    Bronx  
    v
   /|
  /  |
Shier          North    Bronx  
    v
   /|
  /  |
Hodges        Sidhill   Brooklyn
```

**Figure 1.4:** A sample network database

The Hierarchical Model

- Similar to the network model.
- Organization of the records is as a collection of **trees**, rather than arbitrary graphs.
- Figure 1.5 shows a sample hierarchical database that is the equivalent of the relational database of Figure 1.3.
Figure 1.5: A sample hierarchical database

Database Administrator

1. The **database administrator** is a **person** having central control over data and programs accessing that data. Duties of the database administrator include:
   - **Scheme definition:** the creation of the original database scheme. This involves writing a set of definitions in a DDL (data storage and definition language), compiled by the DDL compiler into a set of tables stored in the data dictionary.
   - **Storage structure and access method definition:** writing a set of definitions translated by the data storage and definition language compiler
   - **Scheme and physical organization modification:** writing a set of definitions used by the DDL compiler to generate modifications to appropriate internal system tables (e.g. data dictionary). This is done rarely, but sometimes the database scheme or physical organization must be modified.
   - **Granting of authorization for data access:** granting different types of authorization for data access to various users
   - **Integrity constraint specification:** generating integrity constraints. These are consulted by the database manager module whenever updates occur.

Overall System Structure

1. Database systems are partitioned into modules for different functions. Some functions (e.g. file systems) may be provided by the operating system.
2. Components include:
   - **File manager** manages allocation of disk space and data structures used to represent information on disk.
   - **Database manager:** The interface between low-level data and application programs and queries.
   - **Query processor** translates statements in a query language into low-level instructions the database manager understands. (May also attempt to find an equivalent but more efficient form.)
- **DML precompiler** converts DML statements embedded in an application program to normal procedure calls in a host language. The precompiler interacts with the query processor.
- **DDL compiler** converts DDL statements to a set of tables containing metadata stored in a data dictionary.

In addition, several data structures are required for physical system implementation:

- **Data files**: store the database itself.
- **Data dictionary**: stores information about the structure of the database. It is used heavily. Great emphasis should be placed on developing a good design and efficient implementation of the dictionary.
- **Indices**: provide fast access to data items holding particular values.

**Database security** concerns the use of a broad range of information security controls to protect databases (potentially including the data, the database applications or stored functions, the database systems, the database servers and the associated network links) against compromises of their confidentiality, integrity and availability. It involves various types or categories of controls, such as technical, procedural/administrative and physical. **Database security** is a specialist topic within the broader realms of computer security, information security and risk management.

Security risks to database systems include, for example:

- Unauthorized or unintended activity or misuse by authorized database users, database administrators, or network/systems managers, or by unauthorized users or hackers (e.g. inappropriate access to sensitive data, metadata or functions within databases, or inappropriate changes to the database programs, structures or security configurations);
- Malware infections causing incidents such as unauthorized access, leakage or disclosure of personal or proprietary data, deletion of or damage to the data or programs, interruption or denial of authorized access to the database, attacks on other systems and the unanticipated failure of database services;
- Overloads, performance constraints and capacity issues resulting in the inability of authorized users to use databases as intended;
- Physical damage to database servers caused by computer room fires or floods, overheating, lightning, accidental liquid spills, static discharge, electronic breakdowns/equipment failures and obsolescence;
- Design flaws and programming bugs in databases and the associated programs and systems, creating various security vulnerabilities (e.g. unauthorized privilege escalation), data loss/corruption, performance degradation etc.;
- Data corruption and/or loss caused by the entry of invalid data or commands, mistakes in database or system administration processes, sabotage/criminal damage etc.
Many layers and types of information security control are appropriate to databases, including:

- Access control
- Auditing
- Authentication
- Encryption
- Integrity controls
- Backups
- Application security
- Database Security applying Statistical Method

**Introduction to MS Access 2007**

A database is a collection of information that’s related. Access allows you to manage your information in one database file. Within Access there are four major areas: Tables, Queries, Forms and Reports

- Tables store your data in your database
- Queries ask questions about information stored in your tables
- Forms allow you to view data stored in your tables
- Reports allow you to print data based on queries/tables that you have created

**Understanding the Views**

There are 2 basic views when you work in a table: Design View and Datasheet View. Design View is used to set the data types, insert or delete fields, and set the Primary key. Datasheet View is used to enter the data for the records. By default, Access places you in Datasheet View.

**Setting a Primary Key**

The Primary Key is the unique identifier for each record in a table. Access will not allow duplicate entries in a Primary Key field. By default, Access sets the first field in the table as the Primary Key field. An example of a Primary Key would be your Social Security Number. This is something unique about you and should not be duplicated.

**Queries**

You use Queries to view, change, and analyze data in different ways. You can also use them as a source of records for forms and reports.

To Create a Query:
1) Click the Create tab on the Ribbon
2) Click Query Design icon
3) Double-click Create Query in Design View
4) Select the table that you would like to base your Query on
5) Click Add
6) Close the Show Table window

The table(s) will now be displayed in the upper part of the Query Design Screen by boxes containing the tables’ fields.

7) Double click on the field names in the field list window which you would like to include in the Query

Creating a Form Using the Forms Wizard

A form is a database object that is used to enter or display data in a database.

To Create a Form Using the Wizard:

8) Navigate to the table you want to base the form on
9) Click Create on the Ribbon
10) Click Forms Reports

Reports can be based on tables or queries and can be made with the Report Wizard.

To Create a Report Using the Report Wizard:

1) Click the Create tab on the Ribbon
2) Click the Report Wizard icon
3) Select the table or query upon which the report will be based
4) Select the fields that you want to include on the report by double clicking on them
5) Click Next
6) If you would like to add grouping to your report, select the field you wish to group by double clicking on it (Example: City)
7) Click Next
8) Select a style for the report
9) Click Next
10) Type a title for the report
11) Click Finish