**Chapter 1**

Table of Contents

[Introduction 2](#_Toc49497238)

[File Processing Systems 3](#_Toc49497239)

[View of Data 5](#_Toc49497240)

[Database Languages 6](#_Toc49497241)

## Introduction

A database management system (DBMS) is a collection of interrelated data and a set of programs to access that data. For a database system, it is essential that:

* Data can be stored and retrieved in a convenient and efficient manner
* Safety of the stored data is ensured
* If data is shared amongst several users, anomalous results are avoided

Among all the system components (hardware, software and data), data is the most important.

## File Processing Systems

Before the existence of Database Management Systems, data was stored and managed in files. This caused a large number of problems.

* **Repetition of data** - For example, separate files for student results and student fees would both have to hold the student ID.
* **Disagreement between data** – Changes made to data in one file may not be reflected in other files.
* **Difficult to access data** – Accessing any information, especially with criteria set, would mean accessing a number of files and comparing the text between them. Consider trying to retrieve the list of students from two departments who are from the same country. We would need to retrieve the names of all students from that country from one file and then compare that data with the text in the files containing the names of students in each of those departments.
* **Data isolation** – This refers to how files may be stored in different formats such as .txt and .docx. We would need to write separate code to deal with each file format.
* **Inability to put constraints** – Every system needs to be able to satisfy certain consistency constraints. For examples, students need to have an ID. This sort of constraint cannot be set using files. However, this problem exists to a certain extent in a database management system as well. The constraints are enforced by adding the appropriate code in the applications. However, if constraints are changed, then the code needs to be edited again, which can be difficult, especially if the data is from multiple locations.
* **Atomicity problems** – There are certain actions that have to be atomic, that is, they must occur in their entirety, or not at all. Consider a fund transfer from one account to another. We must first deduct the balance from one account, and then add the balance to another. If at any point a problem occurs, the entire process has to be cancelled and everything must be brought back to their original state. When we try to implement this with files, it is entirely possible that the balance is deducted from one account, but a problem occurs before it can be added to the other account, thus showing both accounts an incorrect balance.
* **Con-current access anomalies** – File processing systems would be unable to deal with two users modifying the data at the same time. For example, if two users tried to withdraw money from the same account at the same time, the balance change should reflect the total change from both transactions to both users. A locking mechanism is needed to make this possible, which is not available in a file processing system.
* **Security problems** – Specific data within a file cannot be restricted from certain users. If a user is given access to a file, they must be given access to everything it contains. Thus, security of the data becomes an issue.

## View of Data

A database system is a collection of interrelated data and a set of programs that allows users to access and modify this data. A major purpose of this is to provide users with an abstract view of the data, i.e. the system hides certain details of how the data is stored and maintained in order to make the process less complex for users.

* **Physical Level**: This is the lowest level of abstraction and describes how the data is actually stored. It describes complex low-level data structures in detail.
* **Logical Level**: The next level describes what data is stored and the relationships between this data.
* **View Level**: The highest level of abstraction is used mainly for reporting or viewing data and only describes part of the entire database. The View Level may provide multiple views based on which user is viewing the data and what their access level is.

## Database Languages

A database system consists of two languages, the Data-Definition Language (DDL) and the Data Manipulation Language (DML). In practice, these are not two separate languages, but instead two parts of a single database language, such as the Structured Query Language (SQL).

DDL is used to specify the database schema, which is the overall design of the database. It is also used to specify additional properties like domain constraints (the range of acceptable values), referential integrity (ensures references are valid), assertions (conditions that *must* be satisfied) and authorization (who can view or edit which data). DDL is just like any other programming language, taking an input and generating an output. The output of the DDL is placed in a data dictionary, which contains metadata (data about data).

DML is used to express database queries and updates, i.e. it allows users to access and manipulate the data using the appropriate data model. DML is mainly concerned with the retrieval, insertion, deletion and modification of data. It has three operations, INSERT, UPDATE and DELETE.