Module 1 - Data, Database and Database Management System

What is Data?

Data means **raw facts or figures** — things we collect or observe in the real world, but they don't have meaning by themselves.
It could be:
- Numbers (like 23, 5.6, 100)
- Words (like "John", "Blue", "Apple")
- Dates (like 30-06-2025)
- Symbols (like @, #, %, etc.)

Important: Data alone doesn't tell us much — it becomes useful only when it is **organized** and given **context**. That's where **databases** come in.

Example:

Imagine a **school** wants to keep track of its students.
Here's some raw data they might collect:
- "John"
- "10"
- "A+"
- "Mathematics"

By themselves, these are just **data points**. We don't know what they mean yet.
But if we organize them like this:

Student Name	Grade	Marks	Subject
John	10	A+	Mathematics

This **organized data** now gives us **information**.

Difference Between Data and Information?

Term	Meaning
Data	Raw facts and figures (unprocessed)
Information	Organized and processed data that gives meaning or value

Why Is Data Important in Databases?

In a **database**, we store a **lot of data** in an organized way so that:
- It's **easy to find** later
- It can be **updated**
- It can be **analyzed** to make decisions

For example, a school database might have **data** about:
- Students (name, age, grade, subjects)
- Teachers

- Fees
- Attendance

All of this data is stored in **tables**, like in Excel, but more powerful.

Example:

Unorganized Data (Raw):

- 42
- Apple
- 1.5 kg
- ₹120

Organized in Context:

Item	Quantity	Weight	Price
Apple	6	1.5 kg	₹120

Now we know: someone bought **6 apples weighing 1.5 kg for ₹120** — that's **information**, made from data.

What is a Database?

A **Database** is a **collection of organized data** that can be **easily accessed, managed, and updated.**

Imagine a **digital cupboard** where you store information in a neat and organized way so that you can find and use it whenever you want.

Why do we need a Database?

In real life, we deal with **huge amounts of data** — about students, customers, employees, products, etc.

If we just store this data in paper files or random Excel sheets, it becomes **hard to manage** — we need a better way to **store, search, update, and analyze** it. That's where a **database** comes in — it stores data in **structured form.**

Real-Life Example: A School Database

Imagine a school wants to keep records of students.
Instead of writing everything on paper, they create a **student database** with information like:

Student_ID	Name	Class	Age	Marks
101	Rahul	10	15	88
102	Priya	9	14	92

This is a **database table** — it organizes **related data** into **rows and columns.**

- Each **row** = one student (record)
- Each **column** = one type of information (field)

So, a **database** is like an **organized digital notebook** for storing large sets of data in an efficient way.

What is a Database Management System (DBMS)?

A **DBMS** is a **software** that helps us **create, manage, and interact with a database.**
Think of it as a **manager or helper** that takes care of storing, retrieving, and modifying the data inside a database.

Why do we need a DBMS?
Without a DBMS, it would be very difficult to:
- Add new data
- Delete old data
- Search for specific data
- Keep data safe and consistent

A DBMS automates all these tasks and makes sure the database works smoothly and securely.

Popular DBMS Software Examples:

DBMS Name	Description
MySQL	Open-source, widely used in web systems
Oracle DB	Powerful, used in large enterprise systems
Microsoft SQL Server	Used in many business applications
PostgreSQL	Open-source, known for stability and power
SQLite	Lightweight, used in mobile apps

How Database and DBMS Work Together

Component	Role
Database	Where data is stored (like files in a cabinet)
DBMS	The software that helps you create, read, update, and delete that data (like a smart assistant)

Example: Managing Student Data with DBMS

Let's go back to the **school** example.
- The **Database** contains student info (tables, records).
- The **DBMS** allows you to:
  - Add a new student: INSERT INTO Students VALUES (103, 'Amit', 10, 15, 90);
  - View students of class 10: SELECT * FROM Students WHERE Class = 10;
  - Update marks: UPDATE Students SET Marks = 95 WHERE Student_ID = 101;
  - Delete a student: DELETE FROM Students WHERE Student_ID = 102;

You don't need to manually search files or rows — the DBMS does it all in seconds using **SQL (Structured Query Language)**.

Features of a Good DBMS:

1. **Data Storage** – Efficiently stores large volumes of data.
2. **Data Retrieval** – Quickly finds and displays data.
3. **Data Manipulation** – Easily insert, update, or delete data.
4. **Data Security** – Controls who can access or change the data.
5. **Backup and Recovery** – Saves data and restores it if lost.
6. **Data Integrity** – Makes sure data is accurate and consistent.

Traditional File System vs DBMS

Feature	Traditional File System	DBMS
Data Redundancy	High	Low
Data Security	Poor	Strong security & access control
Data Access Speed	Slow for large data	Fast and optimized
Multi-user Support	Limited	Yes
Data Relationships	Hard to manage	Easily handled via tables