
# Database and its Applications

## Data Models and Mathematical Foundations

**Pooja T S**

Computer Applications

# Database and its Applications

## Data Models and Mathematical Foundations

## Introduction to Databases and Applications

**Pooja T S**

Computer Applications

# Database and its Applications

## What is Data?

- Data is a collection of raw facts, figures, or observations stored electronically.
- Characteristics:
  - Raw and unprocessed
  - Uninterpreted until context is added
  - Exists in various forms: text, numbers, images, multimedia
- Example:
  - Student ID: 10234
  - Name: Rahul
  - Marks: 88
- Data vs Information:
  - Data = raw facts (e.g., “88”)
  - Information = processed data with meaning (e.g., “Rahul scored 88 in Mathematics”)

# Database and its Applications

## Introduction to Databases

- Definition of Database
  - A structured collection of data, stored and managed electronically.
  - Supports efficient storage, retrieval, and manipulation of information.
  - Example: Student records in a university.
- Key Features:
  - Organized
  - Accessible
  - Manageable
- Real-life Analogy: A library with categorized books $\rightarrow$ a database with structured data.

# Introduction to Databases

Figure: Databases can be stored and accessed through the cloud

# Database and its Applications

## Key Terms in DBMS

- Important Definitions:
  - **Database** – Organized collection of structured data.
  - **DBMS** – Software to manage and operate databases.
  - **Schema** – Blueprint/structure of the database.
  - **Instance** – Actual data at a specific point in time.
  - **Tuple (Record)** – A row in a table.
  - **Attribute (Field)** – A column in a table.
  - **Primary Key** – Unique identifier for a record.
  - **Foreign Key** – Attribute linking two tables.
  - **Normalization** – Process to reduce redundancy.
  - **ACID Properties** – Atomicity, Consistency, Isolation, Durability.

# Database and its Applications

## Applications of Databases

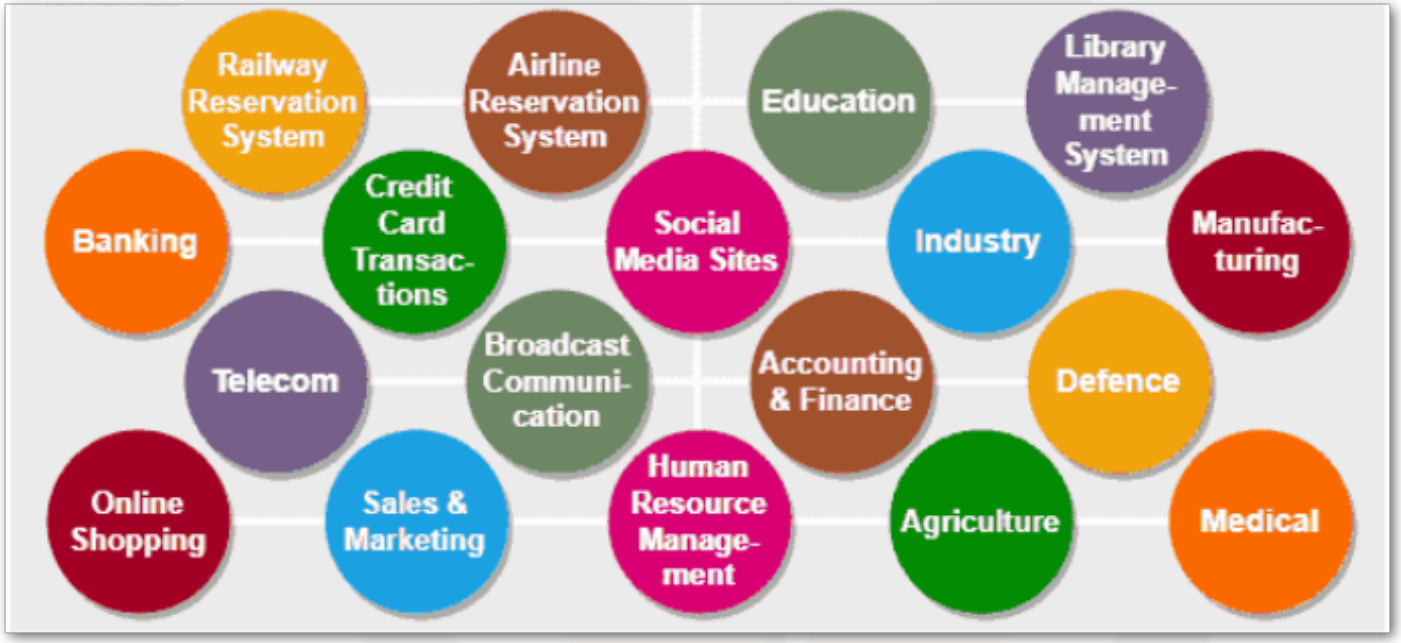



Figure: Real World Applications

# Database and its Applications

## Why Do We Need Databases?

- Problems with Traditional File Systems:
  - Data redundancy
  - Inconsistency
  - Difficulty in retrieval
  - Lack of security
- How Databases Solve These Issues:
  - Centralized control and consistency
  - Efficient querying
  - Stronger security

# Database and its Applications

## Why Use Databases?

- Data Integrity – consistency across operations
- Security – controlled access
- Scalability – handles large datasets
- Concurrency – supports multiple users simultaneously
- Data Independence
- Reduced Redundancy
- Efficient Querying

# Database and its Applications

## Database Components

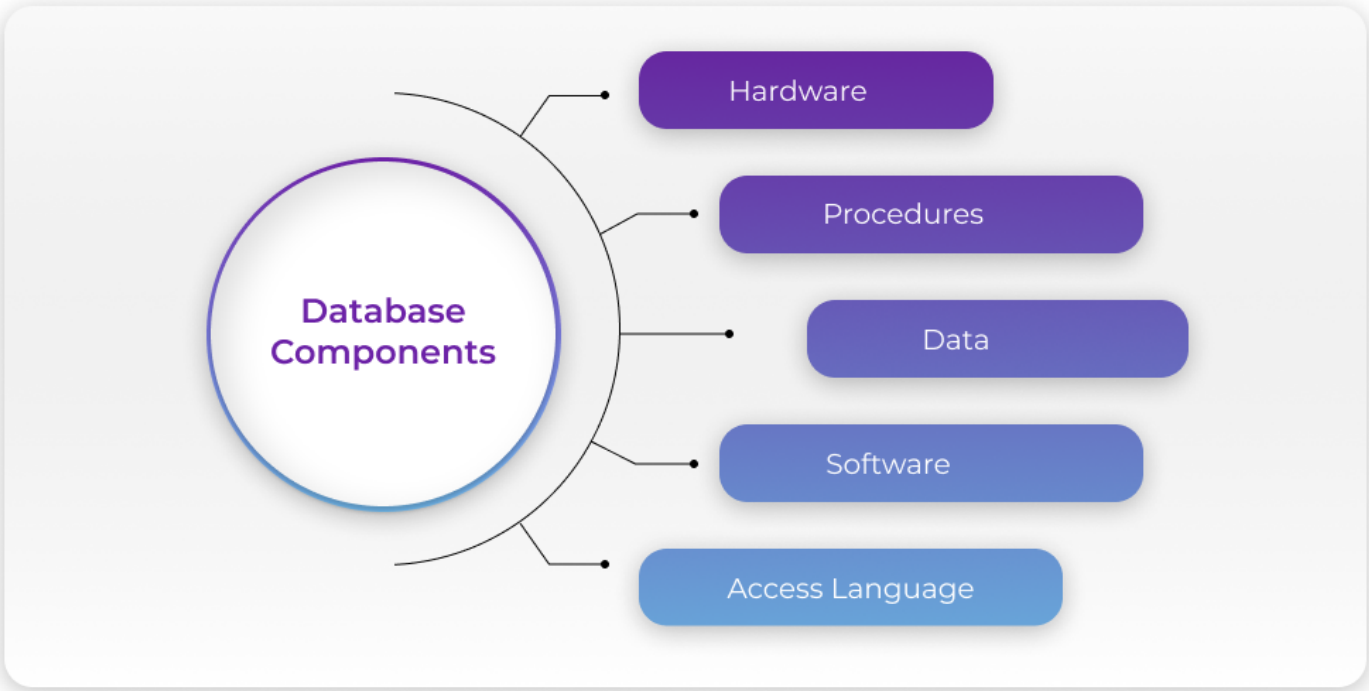



Figure: Components of a Database

# Database and its Applications

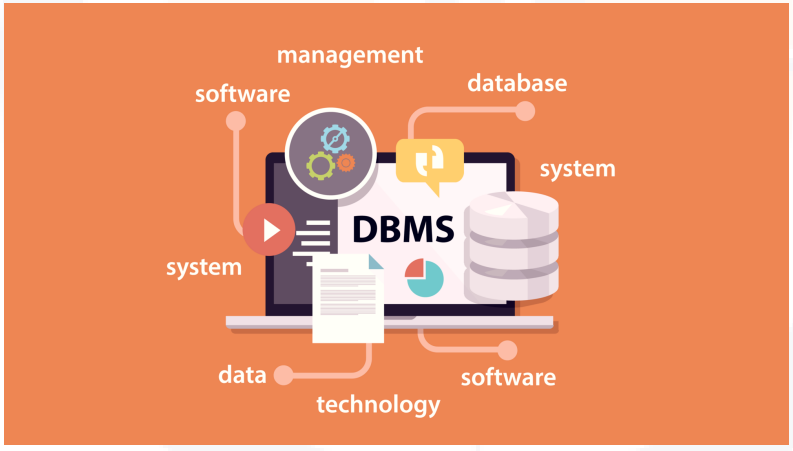
## Database Components

- **Hardware** – Physical devices where the database runs. Example: Servers, storage devices, hard disks.
- **Software** – Programs that manage the database. Example: DBMS like MySQL, Oracle.
- **Data** – The actual information stored in the database. Example: Student records, employee details, sales transactions.
- **Access Language** – The language used to interact with the database. Example: SQL (Structured Query Language).
- **Procedures** – Rules and methods to use/manage the database. Example: Backup procedures, security policies, user guidelines.

# Database and its Applications

## Database Management System (DBMS)

- Software that manages databases
- Provides interface between users/applications and the database
- Ensures data independence and abstraction



# Database and its Applications

## Metadata in DBMS

- Definition: Metadata is “data about data.”
- Purpose:
  - Describes the structure of the database.
  - Defines tables, attributes, data types, and constraints.
  - Includes indexing, storage details, and relationships.
- Stored in: **Data Dictionary** or **System Catalog**.
- Example:
  - Table: `Students`
  - Metadata: {Name: VARCHAR(50), ID: INT PRIMARY KEY, Marks: INT}

# Database and its Applications

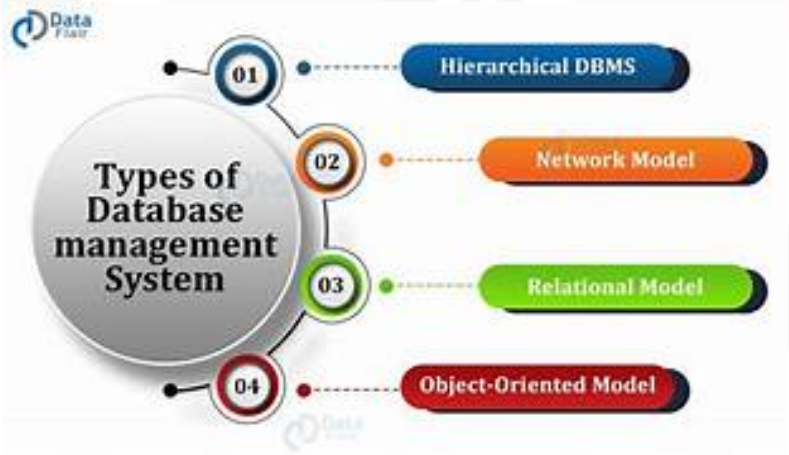
## Database State / Instance

- The data stored in the database at a **particular moment in time**.
- Characteristics:
  - Represents a snapshot of the database.
  - Can change frequently as updates, insertions, and deletions occur.
  - Must always conform to the database schema.
- Difference:
  - **Schema** – The structure (remains mostly unchanged).
  - **Instance/State** – The actual content (changes over time).
- Example:
  - Schema: `Students(id, name, marks)`
  - Instance: {(101, 'Rahul', 88), (102, 'Anita', 92)}

# Database and its Applications

## Types of DBMS

- Relational DBMS (RDBMS) – based on tables (e.g., PostgreSQL, MySQL)
- NoSQL DBMS – flexible, schema-less (e.g., MongoDB)
- Hierarchical/Network DBMS – legacy models



# Database and its Applications

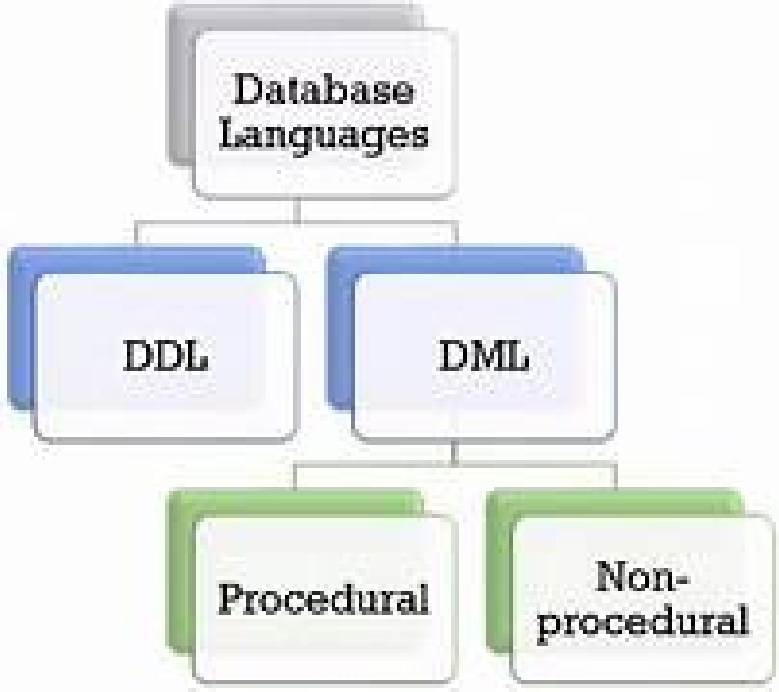
## Database Users

- **End Users**
  - People who use applications to interact with the database.
  - Example: Students checking results, customers shopping online.
- **Application Programmers**
  - Write software that uses the database.
  - Example: Developers creating apps for banking or e-commerce.
- **Database Administrators (DBA)**
  - Manage the database system, ensure security, backup, and performance.
  - Example: IT staff handling university or hospital databases.
- **System Analysts / Designers**
  - Design the database structure and define how data should be stored and accessed.
  - Example: Planning how student data is organized.

# Database and its Applications

## Database Languages

- Data Definition Language (DDL) – schema creation
- Data Manipulation Language (DML) – insert, update, delete
- Data Query Language (DQL) – SELECT queries



# Database and its Applications

## Databases in Modern Applications

- Cloud Databases – scalable, managed, cost-effective (e.g., AWS RDS, Google Firestore, MongoDB Atlas)
- Big Data Integration – works with Hadoop/Spark to handle volume, velocity, and variety of data
- AI/ML-ready Databases – support vector search, predictive analytics, and recommendations (e.g., Pinecone, Oracle AI features)
- Multi-Model Databases – support relational, document, graph, and key-value models in one system
- Edge Databases – enable real-time processing close to IoT devices
- Serverless Databases – auto-scale and charge only for usage

# Database and its Applications

## Summary

- Databases store and manage structured data
- DBMS provides tools for secure and efficient use
- Different models serve different applications
- Databases are critical in nearly all modern domains

# Database and its Applications

## Food for thought!!!!

- Think of an application you use daily (e.g., Zomato, Instagram, Banking App). Identify:
  - What kind of data is stored?
  - How is it organized?


# Thank You

**Pooja T S**

**Assistant Professor**

**Department of Computer Applications**

**poojats@pes.edu**

**080-26721983 Extn: 233**