# Database Documentation

Contents

[Database Documentation 1](#_Toc205110423)

[1. **Comparison Assignment: Flat File Systems vs. Relational Databases** 2](#_Toc205110424)

[**2.** **DBMS Advantages – Mind Map** 4](#_Toc205110425)

[**1.** **Roles in a Database System** 5](#_Toc205110426)

[4. Types of Databases 6](#_Toc205110427)

[5. Cloud Storage and Databases 7](#_Toc205110428)

Rihal Summer Internship

## **Comparison Assignment: Flat File Systems vs. Relational Databases**

A flat file is usually a table with rows of information, whereas a relational database contains many tables with rows and columns that recognize relationships between different pieces of information. Keys are used for linking related data in different tables. Flat files are usually used for simpler applications with less data than relational databases.

|  |  |  |
| --- | --- | --- |
| Feature | Flat File system | Relational Database |
| Structure | Data stored in plain text or CSV. | Many tables with rows and columns (schema-based) |
| Data Redundancy | High chance of duplicate data because flat files lack normalization. | Redundancy is minimized through normalization. Related data is stored in separate tables and linked via keys. |
| Relationships | Relationships between different files are not supported by the system. Any linking must be done manually through code. | Supports complex relationships using **primary keys** (unique identifiers for rows) and **foreign keys** (links to other tables) |
| Example Usage | Used for **simple data storage** where advanced features like querying, relationships, and multi-user access are not needed.  -CSV files for Excel or Google Sheets – System log files | Used in **complex, data-intensive applications like:**  -Bank account systems,  -E-commerce product and order databases. |
| Drawbacks | **Not scalable**: Hard to manage as data grows. | **Requires software**: Needs a Database Management System |

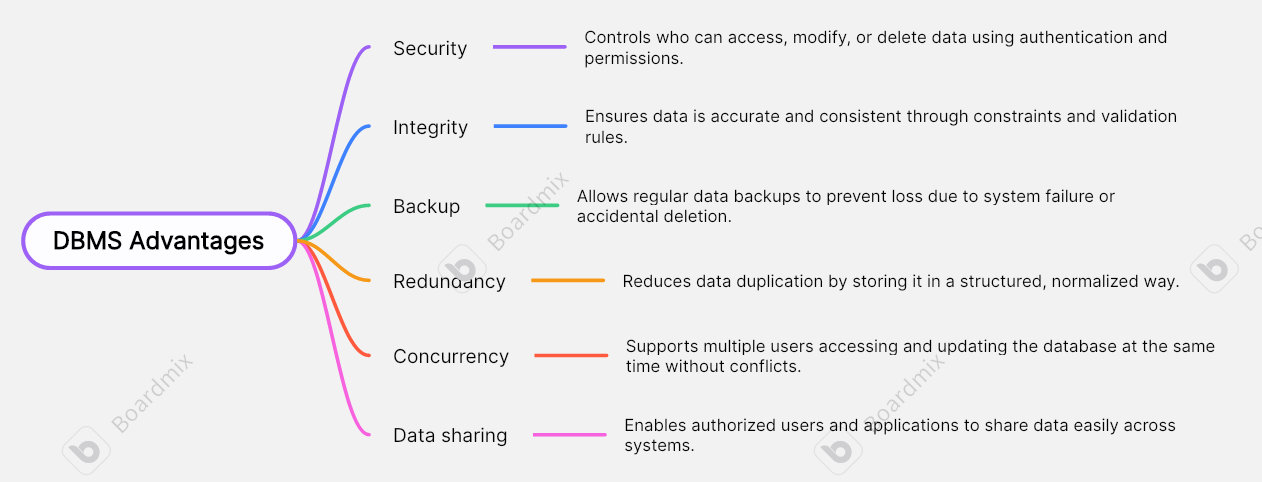
More Drawbacks of the Flat file system:

* No relationships: Can’t link data across files easily.
* High redundancy: Data is often duplicated, increasing errors.
* **limited querying**: No advanced tools to search or filter data.

More Drawbacks of the Relational Database:

* **Learning curve**: Users need to understand SQL and database concepts.
* **May be overkill**: Too heavy for very small, one-off tasks.
* **Performance tuning needed**: For large-scale systems, indexing and optimization are necessary

## **DBMS Advantages – Mind Map**



## **Roles in a Database System**

In this part I’ll Explain the following key roles and what each person typically does in a database project:

**• System Analyst:**

A system analyst studies business processes and identifies what data the system needs to support those processes. They act as a bridge between business users and the technical team by gathering requirements and defining system specifications.

• **Database Designer:**

The database designer creates the data model, defining how data will be stored, organized, and related within the system. They design schemas, tables, keys, and relationships to ensure efficiency, scalability, and data integrity.

• **Database Developer**:

Database developers write the actual code for the database system, including stored procedures, triggers, and queries. They focus on implementing business logic within the database and optimizing performance.

• **Database Administrator (DBA):**

responsible for installing, configuring, maintaining, and securing the database. They manage user access, perform backups, monitor performance, and ensure data availability and integrity.

• **Application Developer:**

They create software that interacts with the database through user interfaces or APIs. They ensure that data flows smoothly between the application frontend/backend and the underlying database.

• **BI (Business Intelligence) Developer:**

BI developers design reports, dashboards, and data visualizations to help organizations make data-driven decisions. They extract and transform data into useful insights using tools like SQL, Power BI, or Tableau.

## Types of Databases

Relational vs. Non-Relational Databases :

* **Relational Databases (RDBMS)**  
  Store data in structured tables with predefined schemas. They use **SQL** for queries and support relationships through primary and foreign keys.  
  **Example:** MySQL, PostgreSQL, Oracle
* **Non-Relational Databases (NoSQL)**  
  Store data in flexible formats like documents, key-value pairs, graphs, or wide-columns. Ideal for handling **large-scale, unstructured, or semi-structured data**.  
  **Examples:**
  + **MongoDB** – Document-oriented, stores data in JSON-like format
  + **Cassandra** – Wide-column store, highly scalable and fault-tolerant

Centralized vs. Distributed vs. Cloud Databases:

* **Centralized Database**  
  All data is stored on a single central server. Easier to manage but prone to a **single point of failure**.  
  **Use Case:** Small companies or local office applications
* **Distributed Database**  
  Data is stored across multiple physical locations or servers. Offers better **scalability, availability, and fault tolerance**.  
  **Use Case:** Multinational organizations, content delivery networks
* **Cloud Database**  
  Hosted on cloud platforms (e.g., AWS, Azure, Google Cloud) with **on-demand scalability** and managed infrastructure. Accessible from anywhere.  
  **Use Case:** Web apps, SaaS platforms, eCommerce sites

## Cloud Storage and Databases

What is Cloud Storage, and How Does It Support Database Functionality?

**Cloud storage** is a service that allows users to save and access data over the internet, instead of relying on local servers or physical drives. In the context of databases, cloud storage serves as the underlying infrastructure where data is stored, backed up, and made accessible to database systems. It enables **cloud-based databases** to scale automatically, support global access, and ensure high availability without needing physical hardware.

Cloud platforms like **Microsoft Azure**, and **Google Cloud** offer managed database services.

**✅** Advantages of Using Cloud-Based Databases

* **High Availability**: Built-in failover and replication reduce downtime and ensure data is always accessible.
* **Automatic Backups**: Regular and secure backups are managed automatically by the provider.
* **Reduced Maintenance**: Cloud providers handle software updates, patches, and infrastructure.
* **Global Access**: Data and applications can be accessed from anywhere with an internet connection.
* **Cost Efficiency**: Pay-as-you-go pricing means you only pay for what you use, reducing hardware costs.

**⚠️** Disadvantages or Challenges of Cloud-Based Databases

* **Latency**: Data access speed can be affected by internet quality or server location.
* **Security Concerns**: Storing sensitive data off-site can raise privacy or compliance issues if not managed correctly.
* **Limited Customization**: Users may not have full control over server configurations or performance tuning.
* **Ongoing Costs**: Long-term operational costs can become high if not carefully managed.