Database

Leena Alhabsi

FLAT FILE SYSTEMS AND RELATIONAL DATABASES

	Flat File Systems	Relational Databases
Structure	Simple, single-table or file format; no hierarchy.	Organized into multiple tables with defined schema and data types.
Data Redundancy	High redundancy — same data may be repeated across records.	Low redundancy due to normalization and data linking.
Relationships	No built-in relationships between data.	Supports relationships using primary and foreign keys.
Example usage	Config files, logs, CSV exports, or simple contact lists.	E-commerce sites, banking systems, inventory management.
Drawbacks	- Harder to manage as data grows. - No querying support. - Poor data integrity.	- More complex to set up and manage. - Requires DBMS software.

DBMS ADVANTAGES

SECURITY

- **Description:** A DBMS provides user access control. Only authorized users can access or modify the data.

REDUNDANCY

Description: Minimizes unnecessary duplication of data using table relationships and normalization.

INTEGRITY

Description: Ensures the accuracy and consistency of data across the database using rules and constraints.

CONCURRENCY

Description: Allows multiple users to access and modify data at the same time without conflicts.

BACKUP

Description: Most DBMSs have automatic or scheduled backup and restore options.

DATA SHARING

Description: Authorized users from different locations or departments can access the same data.

ROLES IN A DATABASE SYSTEM

SYSTEM ANALYST

Role:
Acts as a bridge between business needs and technical solutions.
Responsibilities:

- Analyzes business requirements and translates them into system specifications.
- Works with stakeholders to understand what the system should do.
- Collaborates with database designers and developers to ensure the database meets business goals.

DATABASE DESIGNER

Role:
Designs the structure of the database.
Responsibilities:

- Creates the data model (conceptual, logical, physical).
- Defines tables, relationships, keys (primary/foreign), and constraints.
- Ensures the database is efficient, normalized, and scalable.

ROLES IN A DATABASE SYSTEM

DATABASE DEVELOPER

Role: Builds and implements the database structure and logic.
Responsibilities:

- Writes SQL scripts, stored procedures, triggers, and functions.
- Implements the design created by the database designer.
- Works on optimizing queries and ensuring the database runs efficiently.

DBA (ADMIN)

Role: Manages and maintains the database system.
Responsibilities:

- Installs and configures the DBMS.
- Handles backup, recovery, security, user access, and performance tuning.
- Ensures availability, reliability, and integrity of the data.

ROLES IN A DATABASE SYSTEM

APPLICATION DEVELOPER

Role: Develops software applications that interact with the database.

Responsibilities:

- Creates front-end or backend applications that connect to the database.
- Uses programming languages like Java, Python, C#, etc.
- Works closely with database developers to handle data input/output smoothly.

BI DEVELOPER

Role: Turns raw data into meaningful insights.

Responsibilities:

- Designs and builds dashboards, reports, and data visualizations.
- Uses tools like Power BI, Tableau, or SQL for data analysis.
- Works with data warehouses and helps businesses make data-driven decisions.

RELATIONAL VS NON-RELATIONAL DATABASES

	Relational	Non-Relational
Description	Store data in tables with rows and columns. Data is organized with relationships using primary and foreign keys	Store data in formats like documents, key-value pairs, graphs, or wide-columns — not based on tables.
Example	MySQL, PostgreSQL, Oracle, SQL Server.	• MongoDB (Document-based) • Cassandra (Wide-column store) • Redis (Key-value store) • Neo4j (Graph database)
Use Case Example	Banking systems, HR systems, inventory management — where structured data and relationships are critical.	Social media platforms, real-time analytics, IoT apps, content management — where flexibility and speed are key.

CENTRALIZED VS DISTRIBUTED VS CLOUD DATABASES

	Description	Use Case
Centralized Database	All data is stored in a single central location, typically on one server	Small companies, local apps, or legacy systems with limited access points
Distributed Database	Data is spread across multiple physical locations or servers, but appears as one database to users.	Large-scale systems like multinational e-commerce platforms, global enterprise systems — where speed and fault tolerance are needed.
Cloud Database	Databases hosted on cloud services like AWS, Azure, or Google Cloud. Can be relational or non-relational.	SaaS applications, scalable web apps, startups, and organizations seeking high availability and low maintenance

Cloud Storage and Databases

What is Cloud Storage?

Cloud storage refers to storing data on remote servers hosted on the internet (cloud) instead of on local hardware. The data is maintained, managed, and backed up by a third-party cloud provider (e.g., AWS, Google Cloud, Microsoft Azure).

how does it relate to databases?

Cloud storage provides the foundation for cloud-based databases. Instead of hosting a database on a local server, you can:

- Deploy the database in the cloud
- Use platforms like Amazon RDS, Azure SQL, or Google Cloud Spanner
- Access and manage the database anytime, anywhere via the internet

Advantages & Disadvantages

- Scalability

Easily scale up/down resources based on demand

- Accessibility

Access from anywhere with internet

- Automatic Backups & Updates

Managed services take care of patches and backups

- Cost-Effective

Pay-as-you-go; no need for expensive on-prem hardware

- Internet Dependency

No access if there's no stable internet connection

- Security Concerns

Sensitive data stored offsite; must rely on provider security

- Compliance Issues

Some industries require data to stay on-premise (e.g., healthcare, banking)