DATABASE SEARCH AND REPORTING

Done By: Hibatullah AL Jabri

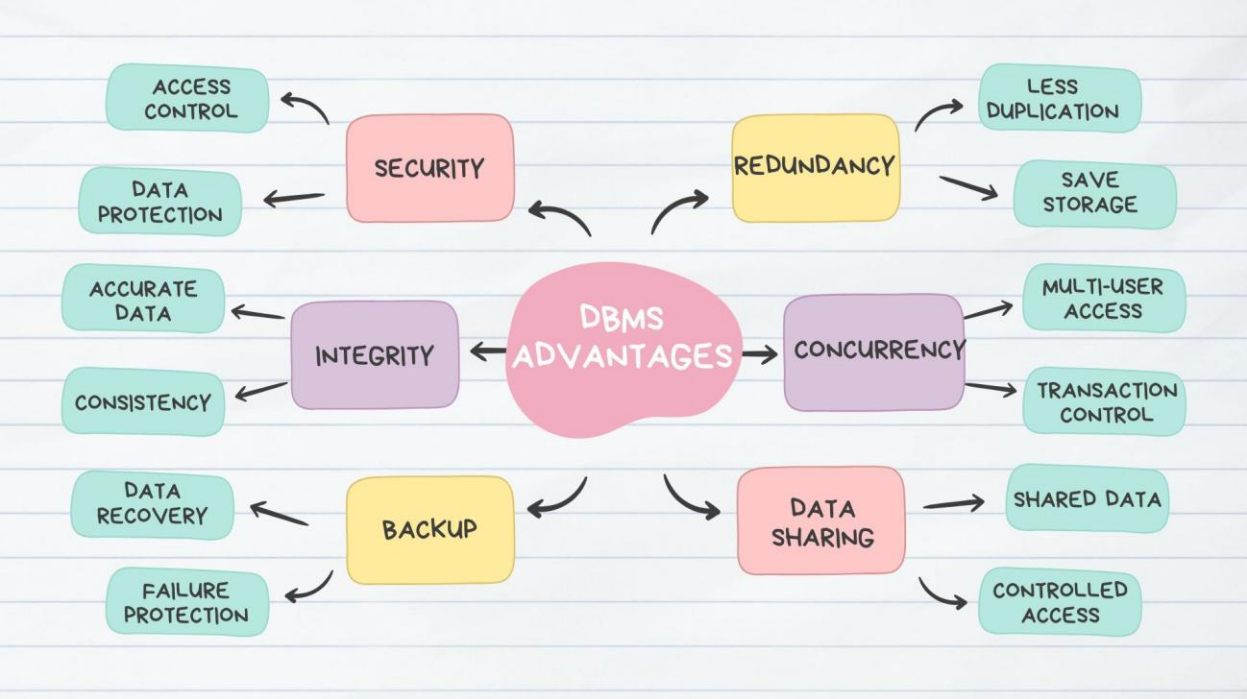
1. Introduction

Databases are a core component of modern software systems. Every full-stack application depends on a database to store, manage, and retrieve data efficiently. Databases help applications handle large amounts of data securely while supporting multiple users at the same time. This report explains database fundamentals, compares flat file systems with relational databases, discusses the advantages of DBMS, explains database roles, describes database types, and highlights the importance of cloud database.

2. Comparison: Flat File Systems vs. Relational Databases

Aspect	Flat File System	Relational Databases
Structure	***Data is stored in single file or table (such as CSV or Excel) with no strict schema or organizational rules.***	***Data is stored in multiple structured tables with predefined schemas (rows and columns).***
Data Redundancy	***High data redundancy because the same data is often repeated in many records.***	***Low data redundancy due to normalization and separation of data into related tables.***
Relationships	***No built-in relationships between data; links must be managed manually.***	***Strong relationship between tables using primary keys and foreign keys.***
Example Usage	***Small datasets, personal records, simple lists, or temporary storge.***	***Business applications such as e-commerce system, banking system, and student management system***
Drawbacks	***Difficult to scale, poor security, high risk of inconsistency, and limited support for multiple users.***	***More complex to design and maintain; requires database management software and SQL knowledge.***

3. DBMS Advantages – Mind Map



4. Roles in a Database System – Explanation

- ***System Analyst***
  The system analyst understands the business problem anr=d translate it into system requirements.

  What they do:
  - *Talk to users and stockholders*
  - *Understand business needs and problems*
  - *Define system requirements*
  - *Decide what data needs to be stored*
  - *Prepare documentation for developers and designers.*
- ***Database Designer***
  Database Designer used to design the structure of the database.

  What they do:
  - *Design database tables and fields.*
  - *Define relationships between tables.*

- *Create Entity Relationship Digrams (ERD).*
- *Choose primary keys and foreign keys.*
- *Ensure the database is well organized and normalized.*

- ***Database Developer***
  Database Developer used for implements the database design using SQL.

  What they do:
  - *Create tables in the database*
  - *Write SQL queries*
  - *Create views, indexes, and stored procedures.*
  - *Optimize database performance*
  - *Ensure data is stored and retrieved correctly*

- ***Database Administrators (DBA)***
  Used for Manages and maintains the database system

  What they do:
  - *Manage database security and user permissions.*
  - *Perform backups and recovery.*
  - *Monitor performance and availability*
  - *Fix database issue and failures*
  - *Ensure the database runs smoothly*

- ***Application Developer***
  The Application Developer builds applications that interact with the database.

  What they do:
  - *Develop front-end and back-end applications*
  - *Connect applications to the database.*
  - *Perform CRUD operations (create, Read, Update, Delete).*
  - *Validate user input.*
  - *Ensure correct data flow between app and database.*

- ***BI (Business Intelligence) Developer***
  The BI Developer analyzes data to support decision-making

  What they do:
  - *Analyze database data*

- ***Create reports and dashboards***
- ***Build data models for analytics***
- ***Help management make data-driven decisions.***
- ***Use BI tools like Power BI or Tableau***

5. Types of Databases

5.1. Relational vs. Non-Relational Databases

Aspect	***Relational Databases***	***Non-Relational Databases (NoSQL)***
Description	Store data in structured tables with rows and columns and use relationships between tables.	Store data in flexible formats and do not rely on traditional table structures.
Data Structure	Highly structured and organized with a fixed schema.	Flexible or schema-less structure.
Query Language	Uses SQL (Structured Query Language).	Uses database-specific query methods (not standard SQL).
Data Integrity	Strong data consistency and integrity.	Weaker consistency compared to relational databases but higher flexibility.
Keys & Relationships	Uses primary keys and foreign keys to define relationships.	Does not rely on traditional keys or table relationships.
Scalability	Scales vertically (by increasing server resources).	Scales horizontally (by adding more servers).
Examples	MySQL, PostgreSQL, SQL Server, Oracle	MongoDB (Document-based), Cassandra (Column-based)
Use Cases	Banking systems, e-commerce websites, student and employee management systems.	Social media platforms, real-time applications, IoT systems, big data applications.
Best Used When	Data structure and relationships are important.	Flexibility and scalability are more important than strict structure.

5.2. Centralized vs. Distributed vs. Cloud Database

Aspect	Centralized Databases	Distributed Databases	Cloud Databases
Description	Store all data in a single server or location.	Store data across multiple servers or locations.	Hosted on cloud platforms and accessed through the internet.
Architecture	Simple and easy to design.	Complex architecture with multiple nodes.	Cloud-based architecture is managed by providers.
Management	Easy to manage and control.	More difficult to manage due to distribution.	Managed by cloud service providers.
Availability	Low availability due to a single point of failure.	High availability and reliability.	Very high availability with built-in redundancy.
Scalability	Limited scalability.	Scales by adding more servers.	Highly scalable on demand.
Maintenance	Requires local maintenance.	Requires skilled administration.	Minimal maintenance by users.
Cost Model	Lower initial cost.	High infrastructure and maintenance cost.	Pay-as-you-use pricing model.
Examples	Local database serves in small offices.	Enterprise banking or telecom systems.	Amazon RDS, Azure SQL Database, Google Cloud Spanner
Use Cases	Small organizations and local office systems.	Large organizations and global systems.	Web and mobile applications, startups, high-availability systems.
Best Used When	The system is small and simple.	High availability and reliability are required.	Scalability and ease of management are priorities.

6. Cloud Storage and Databases

6.1. What is Cloud Storage and How Does It Support Databases?

Cloud storage means storing data on remote serves that are accessed through the internet instead

Cloud storage supports database functionality in several ways:

- ***It stores database files safely on remote servers.***
- ***It allows databases to save backups automatically.***
- ***It helps recover data in case of system failure.***
- ***It supports large amounts of data without needing local hardware.***

6.2. Advantages of Cloud-Based Databases

Databases that run on cloud platforms such as Microsoft Azura, Amazon Web Services, or Google Cloud.

- ***Examples include:***
  - ***Azura SQL Database***
  - ***Amazon RDS***
  - ***Google Cloud Spanner***

- ***Main Advantages:***
  - ***Easy to scale when data grows.***
  - ***Automatic backups and updates***
  - ***High availability and reliability***
  - ***Less hardware and maintenance are needed.***
  - ***Access from anywhere with an internet connection.***
  - ***Pay only for the resources used.***

6.3. Disadvantages and Challenges of Cloud-Based Databases

- ***Dependence on a stable internet connection.***
- ***Ongoing costs can increase with high usage.***
- ***Data security and privacy concerns.***
- ***Limited control over system configuration.***
- ***Possible delays if the cloud server is far from users.***