AI Empowered Youth Program

The 6-months training program - Codeline

week 10

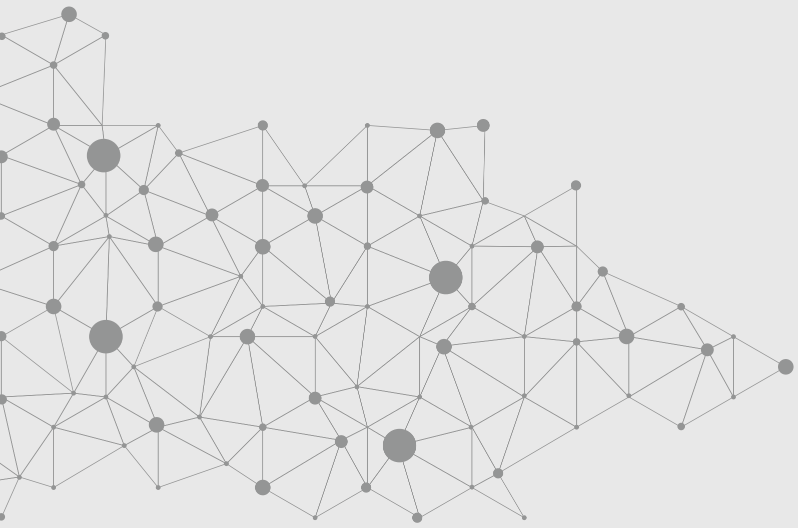
Monday-16-June.2025

Database Report

Prepared by: Nusiba Muslim Juma AL nabhani
instructor: Fatma Almamari

Overview

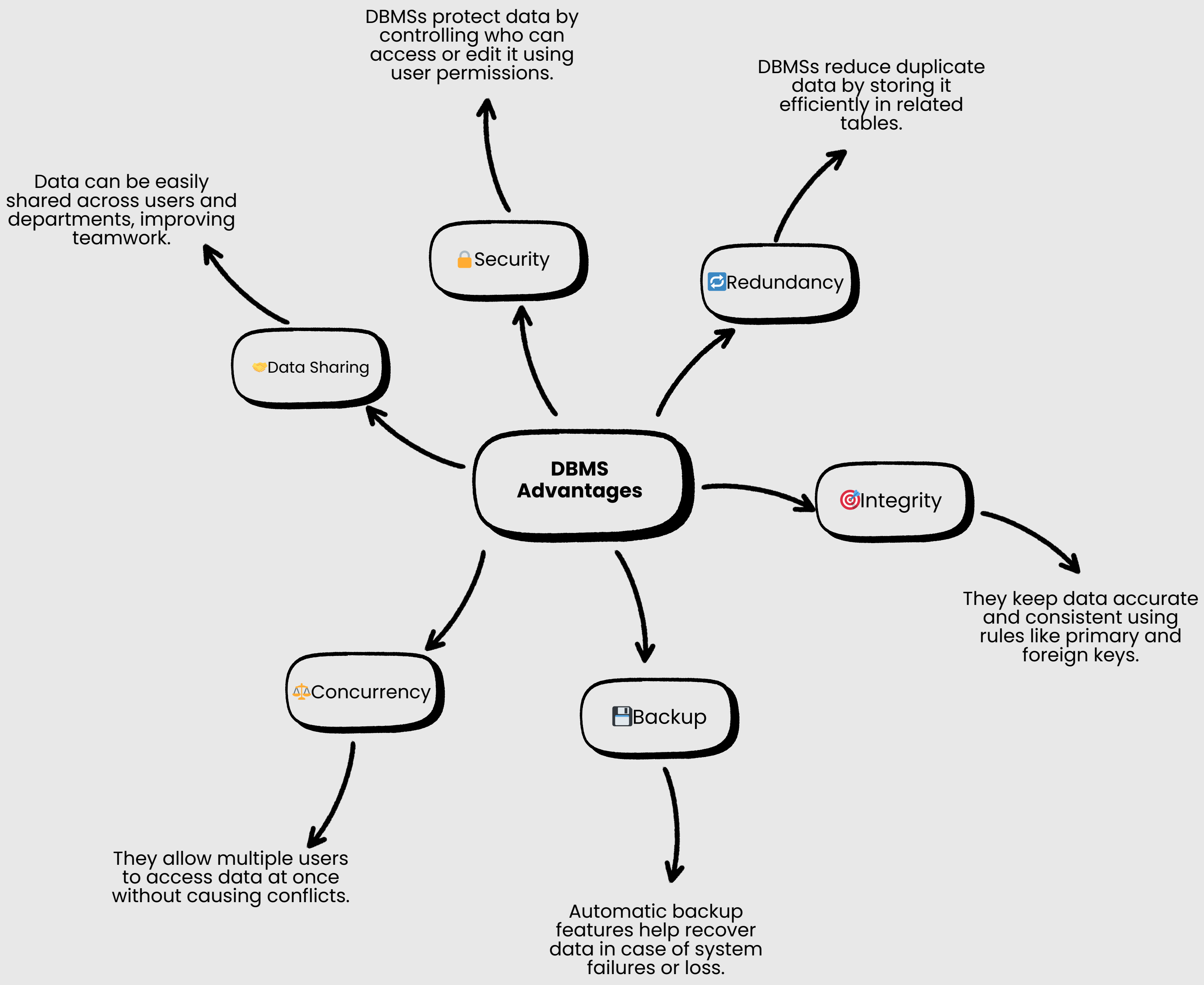
This report provides a concise overview of modern database systems, comparing flat file systems with relational databases and highlighting the benefits of DBMS like security, integrity, backup, and data sharing. It outlines key roles in database projects, including analysts, designers, developers, and administrators. The report covers relational and non-relational databases, as well as centralized, distributed, and cloud-based architectures, with real-world examples. It also explains the link between cloud storage and databases, discussing advantages of cloud services like Amazon RDS and Google Cloud Spanner, along with challenges such as security and vendor lock-in. Overall, it offers a clear foundation for understanding databases in today's data-driven world.



. Comparison Assignment: Flat File Systems vs. Relational Databases

Category	Flat File Systems	Relational Databases (RDBMS)
Structure	Simple text or binary files (like CSV or TXT). No built-in structure beyond what you manually create.	Data is stored in structured tables with defined columns and rows. Each table follows a schema.
Data Redundancy	High – same data may be repeated in multiple files because there's no central reference.	Low – data is split into related tables, and duplication is reduced using keys.
Relationships	Not supported by default. You have to manually manage links between pieces of data.	Built-in support using foreign keys and table relationships (one-to-many, etc.).
Example Usage	Simple applications, config files, spreadsheets, small data storage tasks.	Websites, enterprise software, banking systems, inventory management, CRMs, etc.
Drawbacks	Difficult to manage large or related data. No search features. Poor data integrity.	Requires setup, technical knowledge (like SQL), and may be too complex for small tasks.

. DBMS Advantages – Mind Map


DBMS Advantages
🔒Security
DBMSs protect data by controlling who can access or edit it using user permissions.
🔁Redundancy
DBMSs reduce duplicate data by storing it efficiently in related tables.
🤝Data Sharing
Data can be easily shared across users and departments, improving teamwork.
🎯Integrity
They keep data accurate and consistent using rules like primary and foreign keys.
⚖️Concurrency
They allow multiple users to access data at once without causing conflicts.
💾Backup
Automatic backup features help recover data in case of system failures or loss.

. Roles in a Database System

Effective database projects require collaboration among several specialized roles. Each role contributes to the design, development, and maintenance of a reliable and efficient database system. Below are the primary roles and their responsibilities:

1. **System Analyst:**

   The System Analyst plays a critical role in understanding business needs and translating them into technical requirements. They work closely with stakeholders to define what the system must achieve, analyze existing processes, and help guide the overall direction of the database solution.

2. **Database Designer:**

   The Database Designer is responsible for creating the logical and physical structure of the database. They develop data models, define tables and relationships, and ensure that the design supports data integrity, performance, and scalability. Their work lays the foundation for how data is organized and accessed.

3. **Database Developer:**

   Database Developers build and implement the database according to the designer's blueprint. They write SQL scripts, create stored procedures, and optimize queries. Their focus is on creating a database that functions efficiently and supports the required operations and applications.

4. **Database Administrator (DBA):**

   The DBA is in charge of maintaining and managing the database environment. This includes tasks like installing database software, configuring servers, performing backups, tuning performance, and enforcing security. They ensure the database runs smoothly and is available, secure, and recoverable.

5. Application Developer:

Application Developers create software applications that interact with the database. They design and build user interfaces, write application logic, and connect the front-end to the back-end database. Their goal is to provide users with seamless access to the data they need.

6. BI (Business Intelligence) Developer:

A BI Developer focuses on turning raw data into meaningful insights. They create reports, dashboards, and data visualizations. They use tools like Power BI or Tableau and work closely with the database to extract and analyze data for business decision-making.

. Types of Databases

1. Relational vs. Non-Relational Databases :

Relational Databases (RDBMS):

Relational databases store data in structured tables with rows and columns, using SQL (Structured Query Language) for querying and managing data. They support strong data integrity and relationships using keys.

- Examples: MySQL, PostgreSQL, Oracle, Microsoft SQL Server
- Use Case: Ideal for applications requiring structured data, complex queries, and consistency, such as financial systems, HR systems, or inventory management.

. Non-Relational Databases (NoSQL):

Non-relational databases store data in flexible formats like documents, key-value pairs, wide-columns, or graphs. They are schema-less, scalable, and handle unstructured or semi-structured data well.

- Examples:
    - MongoDB (Document-based) – stores JSON-like documents.
    - Cassandra (Wide-column) – excels at handling large volumes of distributed data.

- Use Case: Best for big data, real-time analytics, content management, IoT, or applications with rapidly changing data structures (e.g., e-commerce product catalogs, social media).

2. Centralized vs. Distributed vs. Cloud Databases :

. Centralized Databases
A single database located on one central server. All users access the database from this single point.

- Use Case: Small to medium businesses or internal systems where high availability and scalability are not critical, such as a local school or small clinic management system.

. Distributed Databases
Data is stored across multiple physical locations (servers or nodes), but appears as one logical database to users. Offers high availability and fault tolerance.

- Use Case: Large-scale enterprise systems, global e-commerce platforms, or any application requiring regional data replication and low-latency access (e.g., Netflix, Amazon).

. Cloud Databases
Hosted on cloud platforms like AWS, Azure, or Google Cloud. Offers scalability, flexibility, and managed services like backups, security, and updates.

- Use Case: Startups, SaaS platforms, or any business that needs to scale quickly and cost-effectively (e.g., web apps, mobile backends, CRM systems).

. Cloud Storage and Databases

Cloud storage is the practice of keeping data on distant servers that may be accessed online as opposed to on local hardware. It enables online data management, access, and storage, usually over a network of dispersed servers. By offering the scalability and infrastructure needed to host and administer databases in the cloud, it facilitates database functioning. Cloud storage makes it possible to implement databases as a service, also referred to as Database-as-a-Service (DBaaS), in place of setting up and maintaining database software on on-premise servers.

Advantages of Cloud-Based Databases

- Scalability: Cloud databases can scale up or down easily to meet demand, without needing new hardware.
- Cost Efficiency: Pay-as-you-go pricing means businesses only pay for what they use, reducing capital expenditure.
- High Availability and Disaster Recovery: Built-in redundancy and automatic backups help ensure uptime and quick recovery after failures.
- Automatic Updates and Maintenance: Providers handle updates, patching, and routine maintenance, reducing the need for in-house DBAs.
- Global Accessibility: Cloud databases are accessible from anywhere, supporting distributed teams and applications.

Examples:

- Azure SQL Database
- Amazon RDS
- Google Cloud Spanner

Disadvantages or Challenges of Cloud-Based Databases

- Internet Dependency: Access to cloud databases requires a stable internet connection, which can be a limitation in some regions.

- Security and Compliance Concerns: Storing data offsite may raise compliance and data privacy issues, especially for sensitive industries.

- Limited Customization: Managed services might restrict access to certain configurations or advanced tuning.

- Vendor Lock-In: Switching cloud providers can be complex and costly due to differences in services and platforms.

- Performance Variability: Shared infrastructure may occasionally lead to unpredictable performance, especially during peak usage.