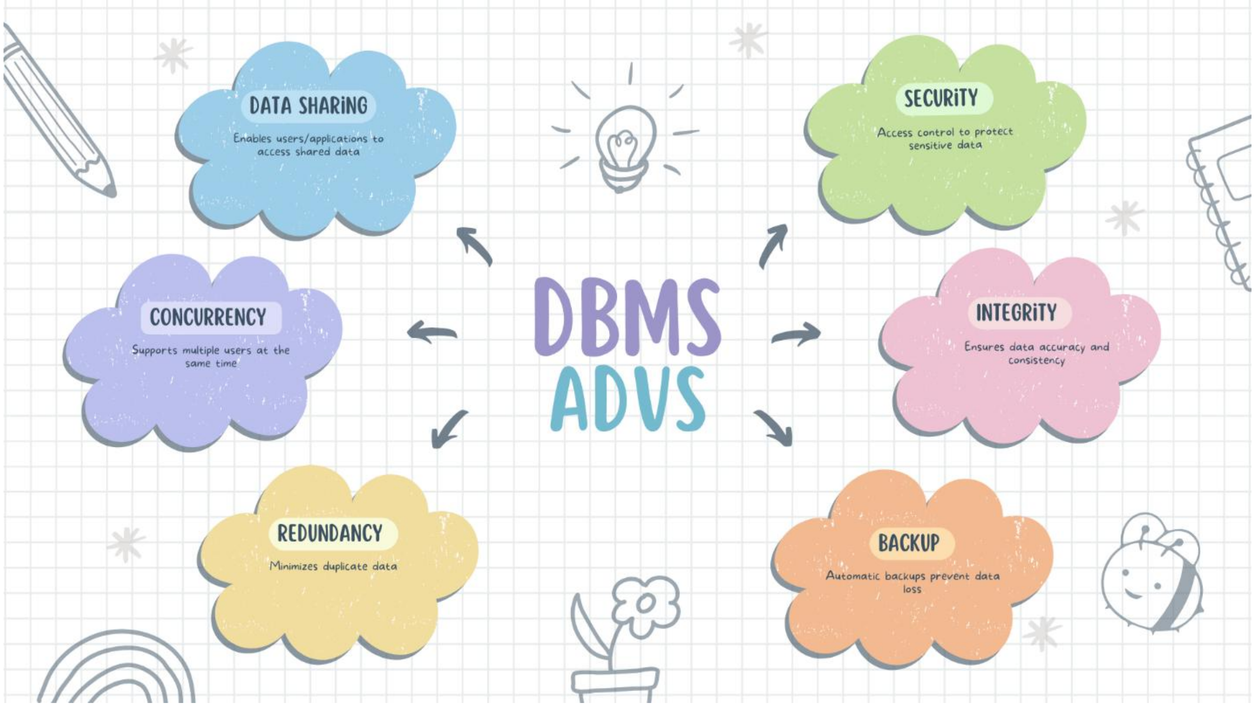
Intro to Database

1. Comparison Assignment: Flat File Systems vs. Relational Databases

Feature	**Flat File System**	**Relational Database**
Structure	Stores data in plain text files	Uses tables with rows & columns
Data Redundancy	High data is often repeated	Low normalization reduces redundancy
Relationships	No built-in support	Can define relationships via keys
Example Usage	CSV files, Excel	MySQL, PostgreSQL, Oracle DB
Drawbacks	Hard to maintain, not scalable	Requires setup and knowledge

2. DBMS Advantages – Mind Map


DATA SHARiNG
Enables users/applications to access shared data
SECURiTY
Access control to protect sensitive data
CONCURRENCY
Supports multiple users at the same time
DBMS
ADVS
INTEGRiTY
Ensures data accuracy and consistency
REDUNDANCY
Minimizes duplicate data
BACKUP
Automatic backups prevent data loss

3. Roles in a Database System

There are different roles and responsibilities in a Database System. Here are the roles and what each person has to do in each role:

1. System Analyst

- Works closely with the client or end users to understand the business needs.
- Defines what the database system should do (requirements).
- Acts as a bridge between users and the technical team.

2. Database Designer

- Designs the database structure (schemas, tables, relationships).
- Focuses on how the data is logically organized.
- Uses tools like ER diagrams to plan the layout of the database.

3. Database Developer

- Builds the database using SQL and other tools.
- Writes queries, procedures, and scripts to manage data.
- Works closely with the designer to turn the design into a working system.

4. Database Administrator (DBA)

- Manages and maintains the database after it's built.
- Handles user access, performance tuning, backups, and security.
- Ensures data is safe, available, and consistent.

5. Application Developer

- Creates applications that interact with the database (like a web app or mobile app).
- Uses programming languages (e.g., Python, Java, PHP) to build user-facing tools.
- Ensures the app sends and receives data correctly.

6. BI (Business Intelligence) Developer

- Analyzes data from the database to support decision-making.
- Builds dashboards and reports (using tools like Power BI or Tableau).
- Helps businesses find patterns, trends, and insights from their data.

4. Types of Databases

➢ Relational vs. Non-Relational Databases

Feature	**Relational Database**	Non-Relational Database
Structure	Tables (rows & columns)	Documents, key-value, graphs, etc.
Schema	Fixed and defined	Flexible or schema-less
Best For	Structured data, clear relations	Unstructured or semi-structured data
Examples	MySQL, PostgreSQL, Oracle	MongoDB, Cassandra
Use Cases	ERP, Banking, HR Systems	Social media, IoT, Big Data analytics

➢ Centralized vs. Distributed vs. Cloud Databases

- **Centralized Database**
  - All data stored in one physical location (single server)
  - Easy to manage and secure but becomes a bottleneck

  - Use Case: Small office systems, early-stage applications

- **Distributed Database**
  - Data spread across multiple locations or servers
  - Improves speed and fault tolerance
  - Use Case: Global apps like Uber or Facebook (faster regional access)

- **Cloud Database**
  - **Hosted and managed in the cloud**
  - **Highly scalable, available from anywhere**
  - **Examples: Amazon RDS, Azure SQL, Google Cloud Firestore**
  - **Use Case: SaaS platforms, startups, enterprise systems**

5. Cloud Storage and Databases

➢ What is Cloud Storage?

- Cloud storage allows data to be saved on remote servers, accessed via the internet.
- It replaces the need to store data on physical local drives.
- Examples: Google Drive, Dropbox, AWS S3.

➢ How Does It Support Databases?

- Cloud platforms provide Database-as-a-Service (DBaaS) — letting developers run, manage, and scale databases online.
- Instead of setting up servers, you just create a database in the cloud.
- Cloud storage provides backups, redundancy, and 24/7 access.

➢ Advantages of Cloud-Based Databases

- **Scalability**: Easily adjust storage and performance as data grows.
- **High Availability**: Cloud providers offer uptime guarantees (99.9%+).
- **Cost Efficiency**: Pay for what you use — no physical infrastructure.
- **Accessibility**: Access from anywhere with an internet connection.
- **Managed Services**: Providers handle maintenance, updates, and security.

Examples**:**

- Amazon RDS (Relational Database Service)
- Azure SQL Database
- Google Cloud Spanner

➢ Disadvantages or Challenges

- **Security Risks**: Data is stored online — needs strong encryption and access control.
- **Latency**: Slower access than local servers in some cases.
- **Vendor Lock-in**: Switching providers later can be hard.
- **Cost Over Time:** Monthly usage fees can add up.

References

Tutorialspoint. (n.d.). DBMS – Overview. https://www.tutorialspoint.com/dbms/dbms_overview.htm

MongoDB. (n.d.). NoSQL Explained. https://www.mongodb.com/nosql-explained

IBM. (n.d.). What is a Cloud Database? https://www.ibm.com/cloud/learn/cloud-databases

Amazon Web Services. (n.d.). Amazon RDS. https://aws.amazon.com/rds/

Google Cloud. (n.d.). Cloud Spanner. https://cloud.google.com/spanner