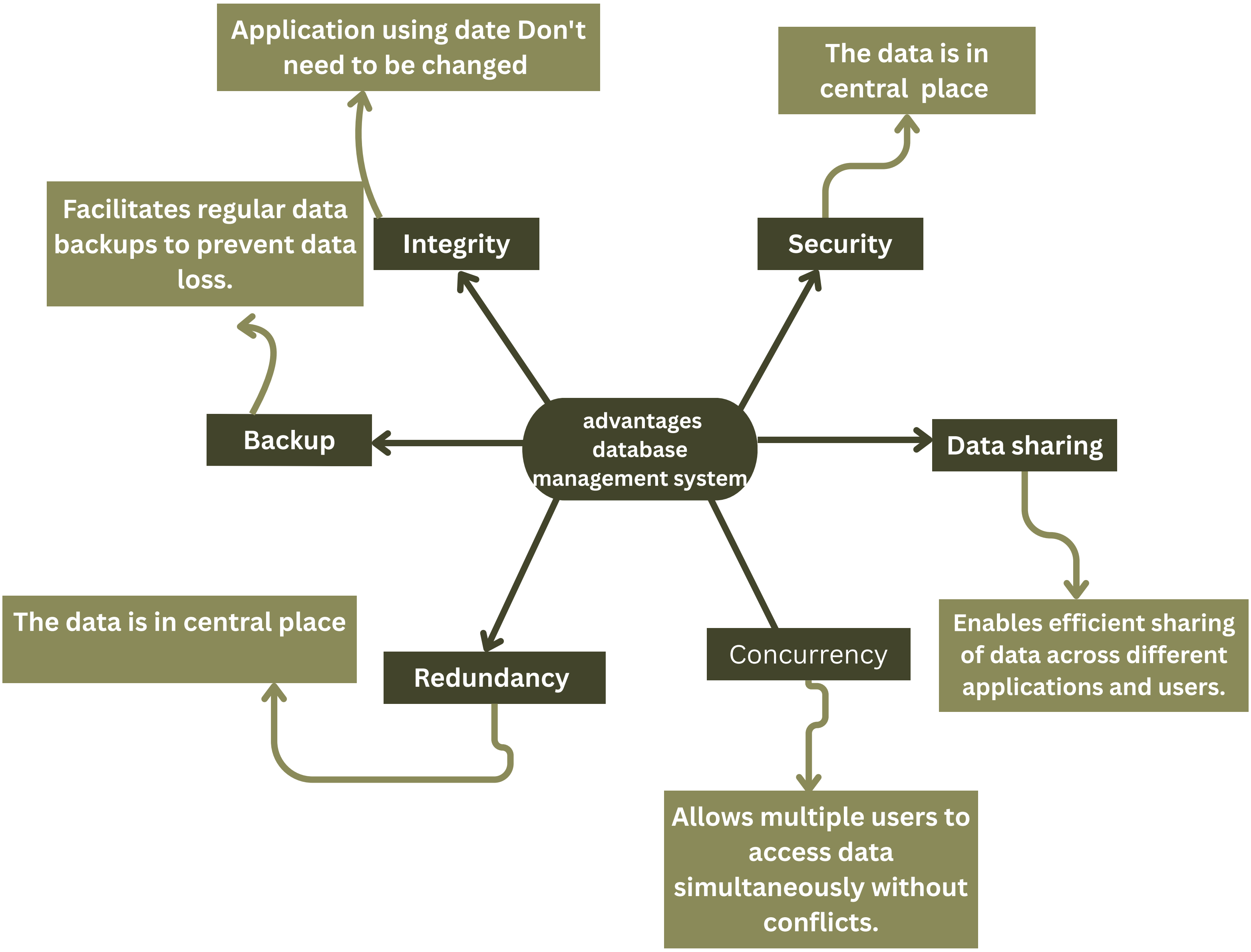
Database Search and Reporting Task

Flat File Systems VS Relational Databases

	Flat File Systems	**Relational Databases**
Structure	Data is stored in a single table or file	Data is organized into multiple tables with rows and columns.
Data Redundancy	High data redundancy as the same data might be repeated in multiple files.	Reduced data redundancy through normalization.
Relationships	No inherent support for relationships between data.	Built-in support for relationships using **foreign keys.**
Example usage	log files	customer relationship management
Drawbacks	limited scalability . performance issues . high risk of data .	More complex. Requires more resources .

database management system



advantages database management system

Integrity

Application using date Don't need to be changed

Security

The data is in central place

Backup

Facilitates regular data backups to prevent data loss.

Data sharing

Enables efficient sharing of data across different applications and users.

Redundancy

The data is in central place

Concurrency

Allows multiple users to access data simultaneously without conflicts.

Roles in a Database System:

System Analyst

technology employee who analyzes, maintains, improves and designs information systems for an organization

Database Designer

responsible for defining the detailed database design, including tables, indexes, views, constraints, triggers, stored procedures, and other database-specific constructs needed to store, retrieve, and delete persistent objects

Database Developer

a software engineer that designs, builds, tests and maintains databases. Database developers create relational database management systems (RDBMS), which store and manage data.

DBA (Admin)

is responsible for maintaining, securing, and operating databases and also ensures that data is correctly stored and retrieved

Application Developer

is a professional who designs, creates, deploys and updates programs for a particular device, the web or a specific operating system.

BI Developer

business intelligence (BI) developer uses data analytics and technology to share valuable business information with decision-makers in their company.

Types of Databases:

	Relational	Non-Relational
Data Structure	Table-based structure	Various Data Models
Stored Items	The stored items have relationships with each other	The stored items can be structured, unstructured or semi-structured data.
performance	strong data consistency and integrity	Faster performance for specific use case
Scalability	Limited	High Scalability
Examples	MySQL, SQL Server, Oracle Database	MongoDB, Cassandra, DocumentDB

Cloud Storage and Databases

Cloud Storage and Databases

is a service that allows data to be stored, managed, and accessed over the internet. It provides scalable and flexible storage solutions without the need for physical hardware. Examples include services like Google Drive, Amazon S3, and Microsoft OneDrive.

Advantages:

- Scalability: Easily scale up or down based on demand.
- Cost-Effective: Pay-as-you-go pricing models reduce upfront costs.
- High Availability: Built-in redundancy and failover mechanisms ensure uptime.
- Managed Services: Providers handle maintenance, updates, and backups.
- Accessibility: Access data from anywhere with an internet connection.

Disadvantages:

- Latency: Potential for higher latency compared to on-premises databases.
- Security Concerns: Data security and privacy issues need to be managed.
- Dependency on Internet: Requires a stable internet connection.
- Vendor Lock-In: Switching providers can be complex and costly.
- Compliance: Ensuring compliance with data regulations can be challenging

Database Engines and Languages

Database Engine is the core service for storing, processing, and securing data. It provides the mechanisms for creating, reading, updating, and deleting data in a database. The engine ensures data integrity, manages transactions, and optimizes query performance.

Advantages:

- **Scalability: Easily scale up or down based on demand.**
- **Cost-Effective: Pay-as-you-go pricing models reduce upfront costs.**
- **High Availability: Built-in redundancy and failover mechanisms ensure uptime.**
- **Managed Services: Providers handle maintenance, updates, and backups.**
- **Accessibility: Access data from anywhere with an internet connection.**

Disadvantages:

- **Latency: Potential for higher latency compared to on-premises databases.**
- **Security Concerns: Data security and privacy issues need to be managed.**
- **Dependency on Internet: Requires a stable internet connection.**
- **Vendor Lock-In: Switching providers can be complex and costly.**
- **Compliance: Ensuring compliance with data regulations can be challenging**

Can We Transfer a Database Between Engines?

it is possible to migrate a database from SQL Server to MySQL or from Oracle to PostgreSQL. There are tools and methods available to facilitate these migrations

SQL Server to MySQL
Tools like MySQL Workbench's Migration Wizard can help migrate databases from SQL Server to MySQL.

Oracle to PostgreSQL
Tools such as Ora2Pg, AWS Database Migration Service (DMS), and pgLoader can assist in migrating databases from Oracle to PostgreSQL

Logical vs. Physical Schema

Logical Schema :

logical schema defines what the database will contain—its structure, including tables, columns, relationships, and constraints, but not how it's physically implemented. It is platform-independent and often created during the design phase to model business requirements.

physical schema :

physical schema defines how the data is stored in the database. It includes details specific to a database system, such as storage format, indexing, partitioning, performance tuning, and actual SQL DDL (Data Definition Language).

	Logical Schema	**physical schema**
Level of Abstraction	High	low
Platform Specific	No	yes (depends on DBMS)
Focus	Entities and relationships	Storage, indexing, performance
Used By	Database designers, analysts	DBAs, system architects
Language	ER Diagrams, UML	SQL

Why Is It Important to Understand Both?

- **Logical schema** helps in aligning the database design with business needs and understanding data relationships.
- **Physical schema** is essential for performance, security, and efficient storage.
- Understanding both ensures that database design is both functional and optimized, allowing seamless communication between analysts, developers, and DBAs.

Logical Schema

Entity: Student
Attributes:
- StudentID (Primary Key)
- Name
- DateOfBirth
- Major

Relationships:
- Enrolls in Courses (many-to-many)

physical schema

```
CREATE TABLE Student (
    StudentID INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    DateOfBirth DATE,
    Major VARCHAR(50)
);

CREATE INDEX idx_major ON
Student(Major);
```