

Title:

Database Course Documentation

A Research and Reporting Assignment on Database Systems

Submitted by:

Israa Yousuf AlWahaibi

Instructor Name: Ms. Fatma AlMamari

Date: 31st July 2025

Table of Contents

Objective	3
Section 1: Flat File Systems vs. Relational Databases	3
Section 2: DBMS Advantages – Mind Map	4
Section 3: Roles in a Database System	4
Section 4: Types of Databases	8
Section 5: Cloud Storage and Databases.....	9
Conclusion	10

Table of Figures

Figure 1: DBMS Advantages – Mind Map	4
--	---

Table of Tables

Table 1: Flat File Systems vs. Relational Databases	3
Table 2: Relational vs. Non-Relational Databases	8
Table 3: Centralized, Distributed, and Cloud Databases	8

Objective

To develop research and analytical reporting skills by exploring key database concepts and presenting them in a well-documented format.

Section 1: Flat File Systems vs. Relational Databases

Table 1: Flat File Systems vs. Relational Databases

FEATURE	FLAT FILE SYSTEM	RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)
STRUCTURE	Data stored in plain text files such as .txt or .csv	Data organized in structured tables with defined schemas
DATA REDUNDANCY	High – no mechanism to eliminate repeated data	Low – normalization minimizes redundancy
RELATIONSHIPS	No support for relationships between data	Supports complex relationships using primary and foreign keys
EXAMPLE USAGE	Simple data logs, configuration files, spreadsheet archives	Banking systems, school databases, eCommerce platforms
DRAWBACKS	Prone to errors, difficult to update or query efficiently	Requires DBMS setup, more complex but scalable

Section 2: DBMS Advantages – Mind Map

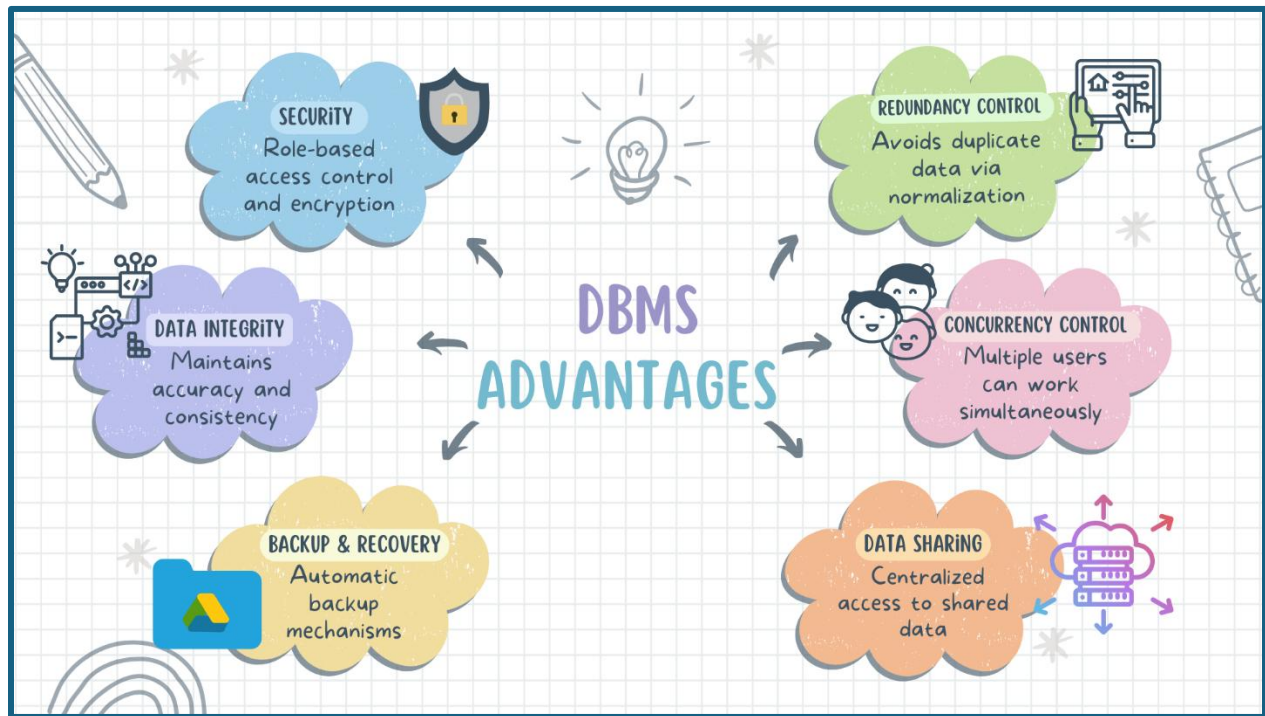


Figure 1: DBMS Advantages – Mind Map

Section 3: Roles in a Database System

In a database project, several professionals with distinct roles collaborate to ensure the successful design, development, deployment, and maintenance of the system. Below is an explanation of the key roles and their typical responsibilities:

System Analyst

Role Overview:

The system analyst acts as a bridge between end-users, stakeholders, and the technical team.

Responsibilities:

- Gathers and analyzes user requirements.
- Studies the existing system and proposes improvements.

- Prepares technical and functional specifications.
- Ensures the database system aligns with business goals.
- Communicates between clients and developers.

Skills Required:

- Business process analysis
- Communication and documentation
- Basic understanding of databases and system design

Database Designer

Role Overview:

The database designer focuses on the logical and physical design of the database structure.

Responsibilities:

- Creates the data model and Entity Relationship Diagrams (ERDs).
- Defines tables, relationships, indexes, and constraints.
- Normalizes data to avoid redundancy.
- Designs for performance, integrity, and scalability.

Skills Required:

- Data modeling (ERD tools like Lucidchart)
- Knowledge of relational design principles
- SQL and normalization techniques

Database Developer

Role Overview:

The database developer implements the design into a working system using SQL and procedural extensions.

Responsibilities:

- Develops stored procedures, views, triggers, and functions.
- Writes optimized SQL queries for CRUD operations.

- Works closely with frontend/backend developers.
- Implements business logic inside the database.

Skills Required:

- SQL, PL/SQL, or T-SQL
- Database performance tuning
- Version control and scripting

Database Administrator (DBA)

Role Overview:

The DBA is responsible for the installation, configuration, maintenance, and security of the database environment.

Responsibilities:

- Installs and upgrades the DBMS.
- Manages user access and security.
- Performs regular backups and recovery testing.
- Monitors performance and applies tuning strategies.
- Ensures data integrity and availability.

Skills Required:

- DBMS tools (e.g., Oracle, SQL Server, MySQL)
- Security management
- Backup and disaster recovery planning

Application Developer

Role Overview:

The application developer builds the front-end or middleware applications that interact with the database.

Responsibilities:

- Creates user interfaces for data input/output.

- Connects apps to databases using APIs or ORM tools.
- Validates and processes user data.
- Implements error handling and session management.

Skills Required:

- Programming languages (e.g., Java, Python, C#, PHP)
- API integration (REST, GraphQL)
- Frontend/backend frameworks (e.g., React, Django, .NET)

Business Intelligence (BI) Developer

Role Overview:

The BI developer extracts useful insights from data and presents them in dashboards or reports.

Responsibilities:

- Designs and builds dashboards and visual reports.
- Writes queries to aggregate and filter data.
- Works with tools like Power BI, Tableau, or SSRS.
- Helps stakeholders make data-driven decisions.

Skills Required:

- SQL and data warehousing
- BI tools and reporting platforms
- Data visualization and presentation

Section 4: Types of Databases

1. Relational vs. Non-Relational Databases

Table 2: Relational vs. Non-Relational Databases

<i>Feature</i>	<i>Relational DB (SQL)</i>	<i>Non-Relational DB (NoSQL)</i>
Structure	Tables with rows and columns	Documents, Key-Value pairs, Graphs, Columns
Schema	Fixed, predefined	Flexible or schema-less
Examples	MySQL, PostgreSQL, Oracle	MongoDB, Cassandra, Redis, Firebase
Use Cases	Banking, HRM, ERP	Social media, IoT, eCommerce, Big Data apps

2. Centralized, Distributed, and Cloud Databases

Table 3: Centralized, Distributed, and Cloud Databases

<i>Type</i>	<i>Description</i>	<i>Use Case Example</i>
Centralized	Single location, controlled access	Small business system
Distributed	Multiple databases across different locations, synchronized updates	Multinational company data synchronization
Cloud-Based	Hosted and managed on cloud platforms (DBaaS)	SaaS applications, scalable mobile/web apps

Section 5: Cloud Storage and Databases

What is Cloud Storage?

Cloud storage stores data over the internet on remote servers. In a database context, it is used to store:

- Backup files
- Live databases (via DBaaS platforms)
- Configuration and scaling data

Cloud-Based Databases

Examples:

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

Advantages:

- **Scalability** – Automatic resource scaling
- **Cost-efficient** – Pay-as-you-go model
- **Availability** – High uptime and disaster recovery
- **Maintenance-free** – Automatic updates and security patches

Disadvantages:

- **Vendor lock-in** – Difficult to migrate between cloud providers
- **Latency issues** – Dependent on network quality
- **Security & Compliance** – Extra measures needed for data privacy
- **Downtime risks** – Cloud outages can affect access

Conclusion

This report explored key database concepts, including differences between flat file and relational databases, benefits of DBMS, and various database roles. It also compared different types of databases and introduced cloud-based storage. Through this assignment, I gained a better understanding of how databases function and their importance in managing data efficiently in modern systems.