The Database Report

Done by Mohammed

Comparing Flat File Systems and Relational Database Systems.

Feature	*Flat File System*	*Relational Database System*
The Structure	Simple with text and binary files	Uses Tables with rows and columns to store data
Data Redundancy	High risk of duplicated files	Low since normalized data reduces duplication
Relations	None since they're structured in folders	Built in using foreign keys
Usages	CSV Files , .log files	MySQL, Oracle
Disadvantages	Poor Scalability and no integrity checks	Complex Setup and requires maintenance

Database Advantages : Mind Map

Roles in a Database System

Role	Responsibilities
System Analyst	Collects requirements, determines the system scope, engages with stakeholders
Database Designer	Creates ERDs and defines schema
Database Developer	Implements schema, writes SQL queries
Database Administrator	Manages DB performance, backups, security, assigns user roles
Application Developer	Integrates DB with applications, handles API/database calls
BI Developer	Designs reports and dashboards

Types of Databases

Relational vs. Non-Relational:

- Relational: Structured, SQL-based (e.g., MySQL, PostgreSQL)
- Non-Relational: Flexible schema, document/key-value (e.g., MongoDB, Cassandra)

Type	Description	Use Case
Centralized	Single location, easier control	Small businesses, local apps
Distributed	Multiple nodes, high availability	Global apps, real-time systems
Cloud	Hosted by providers, scalable	SaaS platforms, mobile apps

Cloud Storage and Database

Consider cloud storage as a distant file cabinet where you can open a lock no matter the location-where you can lodge insider so you can use it any time it is required. And since nobody wants to work with nuts and bolts, these services tidily bundle up all you need to operate a database: the database engine itself, routine backing up and even the facility to upscale or downscale element with minimal bother.

What is the point of using it? The first ones are scalability, availability and cost-efficiency. The physical location of the data in the cloud means that it can be easy to scale up or scale down the resources at any given time, and that the provider will undertake regular maintenance.

Downsides? Vendor lock-in seller lock-in, once you are a customer that is tied to a certain provider, moving can be a complicated, costly process and there is the issue of latency and data privacy.

The best-known are Azure SQL, Amazon RDS, and Google Cloud Spanner. Glip-diagram, in a moment, how a cloud DB usually arranges itself.

End of Report

GitHub : moodyminji