Database Search & Reporting

By: Osama Al Zakwani

>Table of Contents_:

>Research Objective_:

The objective of this report is to develop research and analytical skills by exploring key database concepts and presenting the findings in a clear and structured manner.

>Flat File Systems vs. Relational Databases: A Comparative Study_:

First, let's define each one of them then we will compare based on our understanding.

What is Flat File Systems Database:

It is a simple database system where the data is stored in a local or shared file and the data stored in text format.

What is Relational Database:

It is a structured system way of storing the data and we can create a relationship between the two or more structured (Table) data.

For further distinction between Flat File Systems and Relational Databases, refer to the table below:
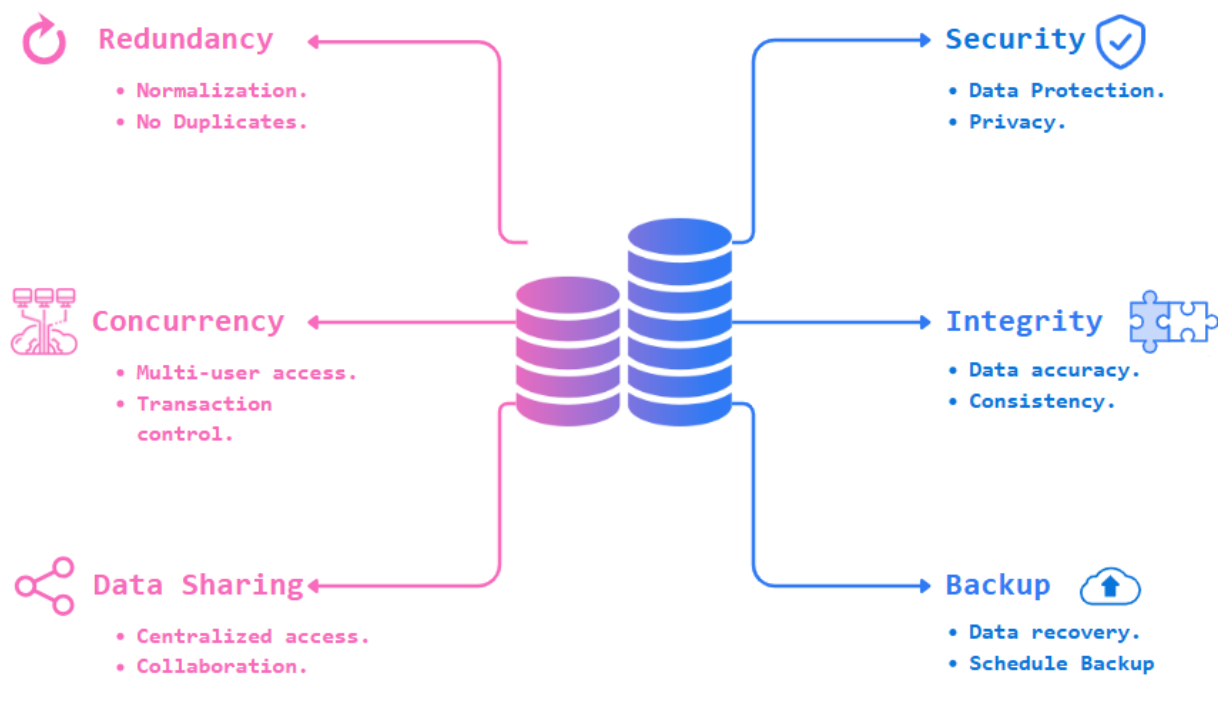
Aspect	Flat File Systems Database	Relational Database
Structure	Every record is stored in one line, and they are separated with commas or taps.	Structured in tables that contain rows and columns.
Data Redundancy	High redundancy (More duplicates)	Low redundancy (Low duplicates)
Relationships	No relationship	There is relationship because of using primary and foreign keys
Example Usage	Used for simple and not too much data	Used for complex and much data.
Drawbacks	Not scalable and difficult to maintain.	More planning for structure and it's difficult to maintain if there is a huge amount of data.

>DBMS Advantages_:

It stands for **Database Management System** and its software used to manage the database like:

1. Create data.
2. Read data.
3. Update data.
4. Delete data.

The following map used to illustrate the advantages of using **Database Management System** (DBMS):

Redundancy
- Normalization.
- No Duplicates.

Security
- Data Protection.
- Privacy.

Concurrency
- Multi-user access.
- Transaction control.

Integrity
- Data accuracy.
- Consistency.

Data Sharing
- Centralized access.
- Collaboration.

Backup
- Data recovery.
- Schedule Backup

>Roles in a Database System_:

These are the key people involved in designing, building, and managing a database. Each role has a specific task to keep the database system working effectively.

Role 1:

System Analyst:

- **Name:** Ahmed Khalid.
- **Role:** Understand what the business (Client) requires.

Role 2:

Database Designer:

- **Name:** Majid Mohammed.
- **Role:** Design the data structure and how the data will be organized in the database.

Role 3:

Database Developer:

- **Name:** Rahma Saif.
- **Role:** Use of programing language (e.g. SQL) to build the database.

Role 4:

Database Administrator:

- **Name:** Nasser Issa.
- **Role:** oversees database security, performance, backups, and user access.

Role 5:

Application Developer:

- **Name:** Malak Rashid.
- **Role:** Apps and website developer that use the terms UI/UX and who brings the database into life.

Role 6:

BI (Business Intelligence) Developer:

- **Name:** Sultan Nabhan.
- **Role:** Turning the raw data into visuals so the decision will be made based on logic.

>Types of Databases_:

Relational database:

This is a type of database that saves and structures the data in structured tables that contain rows and columns and supports SQL for querying as well. We can make the relationship between the tables by using primary and foreign keys.

Examples:

1. MySQL.
2. Oracle.
3. PostgreSQL.
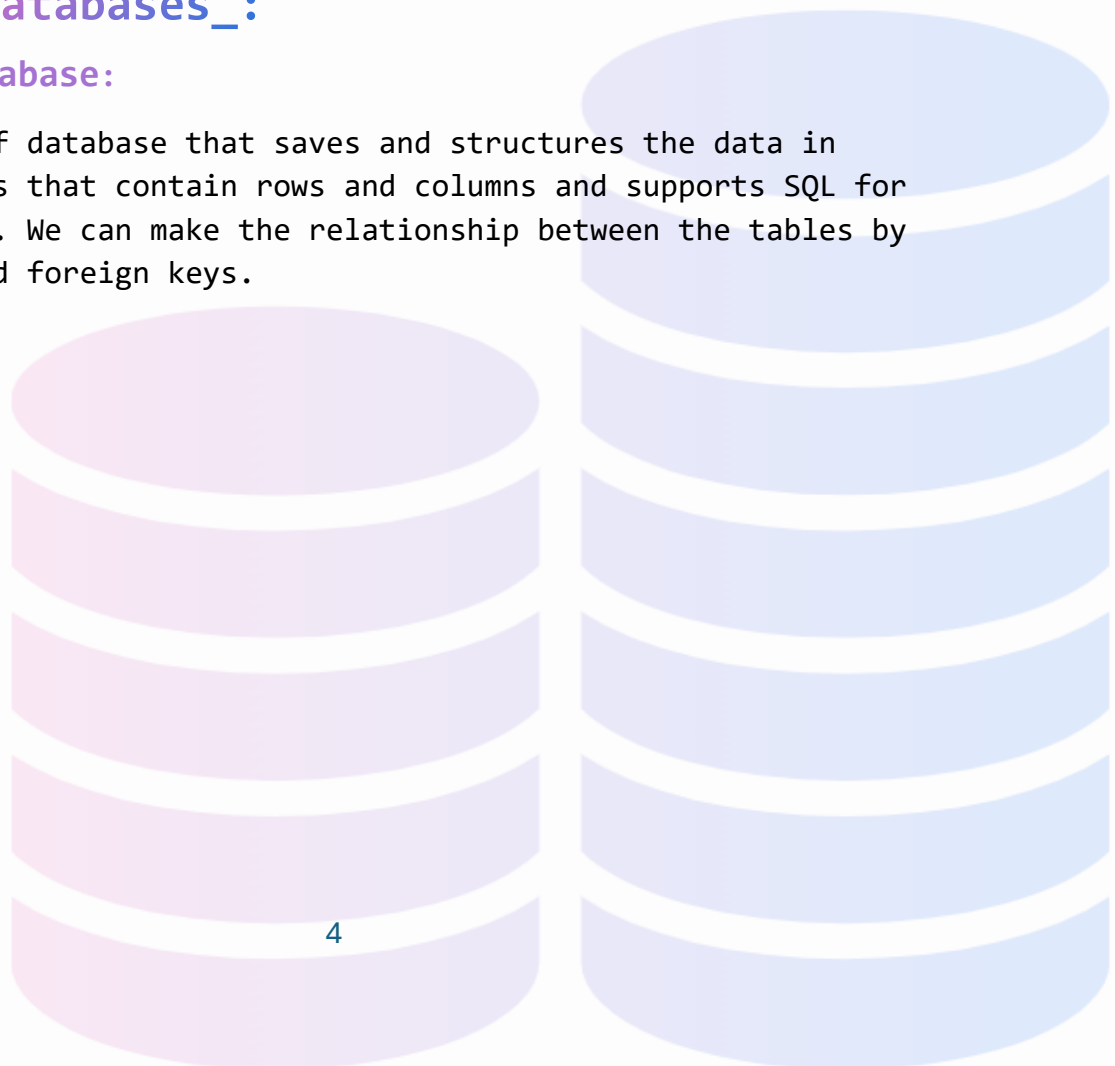
Non-Relational database:

keeps the data in formats such as wide-column, document, graph, and key-value.

Examples:

1. MongoDB.
2. Cassandra.
3. Apache HBase.

Centralized database:

All the data is kept on one server or location.

Examples:

1. Microsoft Access.
2. MySQL.

Use Cases:

1. Local apps.
2. Small businesses.
3. Startups.

Distributed database:

Data is kept on several servers and this type of database will increase the speed of operations.

Examples:

1. Apache Cassandra.
2. Google Spanner.

Use Cases:

1. International apps.
2. Systems with high availability.

Cloud database:

It is a type of database which located in the cloud and its hosted-on cloud platforms rather than in physical locations.

Examples:

1. Microsoft Azure.
2. Google Cloud Platform.
3. Oracle Cloud Database Technologies.
4. Amazon Web Services.

Use Cases:

1. Software as a service application (SaaS)
2. Mobile and web Applications.
3. IoT applications.

>Cloud Storage and Databases_:

What is Cloud Storage? How Does It Support Databases?

Instead of storing data on local computers, cloud storage enables it to be stored remotely on servers connected to the internet. Files, logs, backups, and even structured data are frequently stored there.

Cloud storage facilitates database operation by offering the infrastructure required for dependable, secure, and scalable data storage and retrieval. It makes it possible:
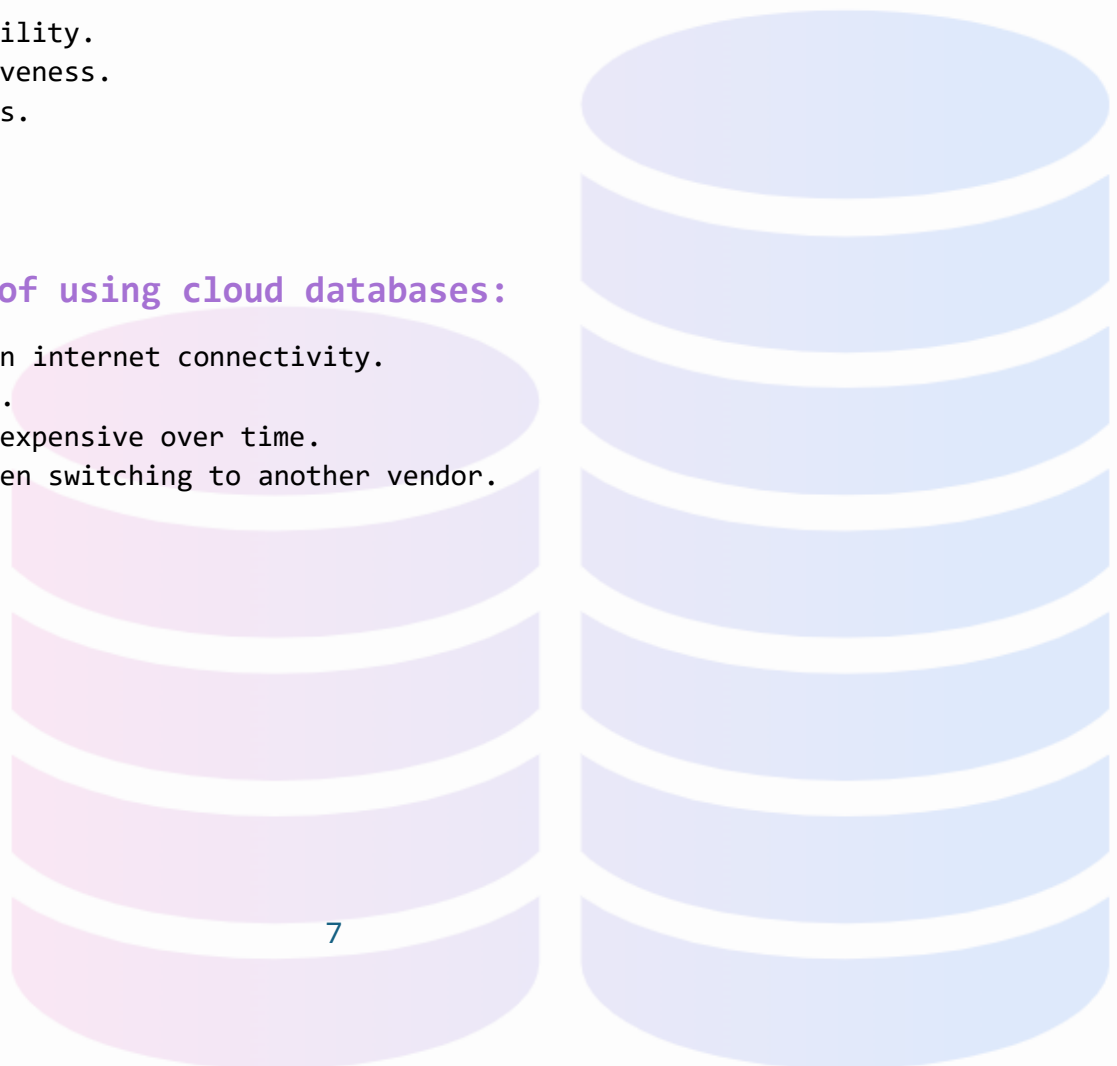
1. Database files are hosted on cloud servers.
2. Automated replicating and backup to safeguard data.
3. The ability to scale and be highly available without requiring physical hardware.
4. Databases can be accessed remotely from any location.

Advantages of using cloud databases:

1. High availability.
2. Cost effectiveness.
3. Online access.
4. Scalability

Disadvantages of using cloud databases:

1. It depends on internet connectivity.
2. Easy hacking.
3. Coming more expensive over time.
4. High cost when switching to another vendor.

>References_:

https://www.mongodb.com/nosql-explained

https://www.oracle.com/database/what-is-database

https://appmaster.io/blog/real-world-examples-of-relational-databases

https://www.pluralsight.com/resources/blog/software-development/relational-vs-non-relational-databases#:~:text=NoSQL%20or%20non%2Drelational%20databases,best%20for%20Rapid%20Application%20Development

https://www.ibm.com/topics/database

https://aws.amazon.com/rds