Unit-01

Database System Introduction

Data

- → Data is nothing but facts and statistics stored or free flowing over a network, generally it's raw and unprocessed. For example: When you visit any website, they might store you IP address, that is data, in return they might add a cookie in your browser, marking you that you visited the website, that is data, your name, its data, your age, it's data.

Information

Data becomes information when it is processed, turning it into something meaningful.

Database

- → A database is an organized collection of structured information, or data, typically stored electronically in a computer system.
- → Database Management System is basically a software that manages the collection of related data.
- → It is used for storing data and retrieving the data effectively when it is needed.
- → It also provides proper security measures for protecting the data from unauthorized access.
- → In Database Management System the data can be fetched by SQL queries and relational algebra.
- → It also provides mechanisms for data recovery and data backup.

DBMS

- → A database is a systematic collection of data. They support electronic storage and manipulation of data. Databases make data management easy.
- → Let us discuss a database example: An online telephone directory uses a database to store data of people, phone numbers, and other contact details. Your electricity service provider uses a database to manage billing, client-related issues, handle fault data, etc.
- → Let us also consider Facebook. It needs to store, manipulate, and present data related to members, their friends, member activities, messages, advertisements, and a lot more. We can provide a countless number of examples for the usage of databases.

DBMS Example: MySql, Oracle, SQL Server, IBM DB2, PostgreSQL, Amazon SimpleDB (cloud based) etc.

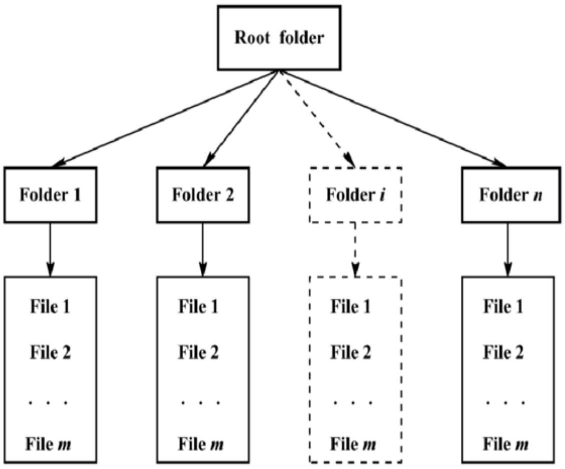
Database benefits:

- → reduce the amount of time you spend managing data.
- → analyze data in a variety of ways.
- → promote a disciplined approach to data management.
- → turn disparate information into a valuable resource.
- → improve the quality and consistency of information.

→ And many more…….

File System

→ File system is basically a way of arranging the files in a storage medium like hard disk.
→ File system organizes the files and helps in retrieval of files when they are required.
→ File systems consists of different files which are grouped into directories.
→ The directories further contain other folders and files.
→ File system performs basic operations like management, file naming, giving access rules etc.



Drawbacks of File system

- Data redundancy and inconsistency

→ duplication of information in different files

- Difficulty in accessing data

→ Need to write a new program to carry out each new task

- Data isolation

→ Multiple files and formats

- Atomicity of updates

→ Failures may leave database in an inconsistent state with partial updates carried out
→ Example: Transfer of funds from one account to another should either complete or not happen at all

- Concurrent access by multiple users

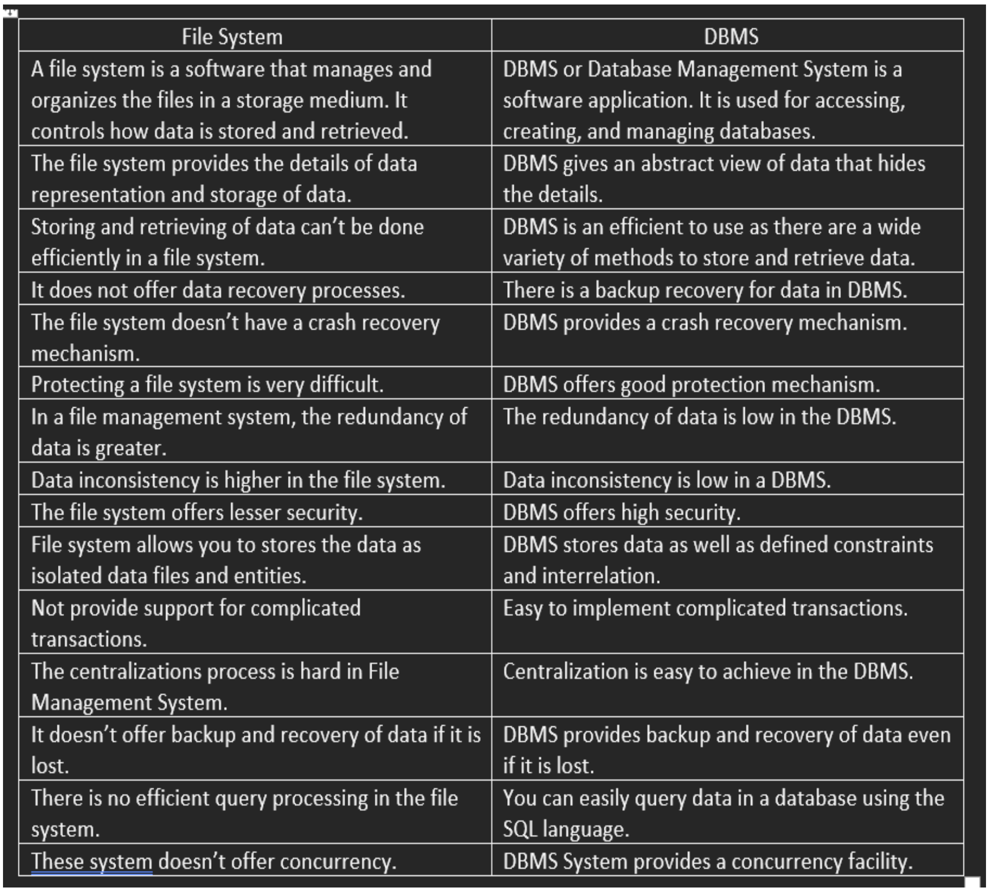
→ Concurrent access needed for performance
→ Uncontrolled concurrent accesses can lead to inconsistencies

→ Example: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time

- Security problems

→ Hard to provide user access to some, but not all, data.

Difference between File System and DBMS

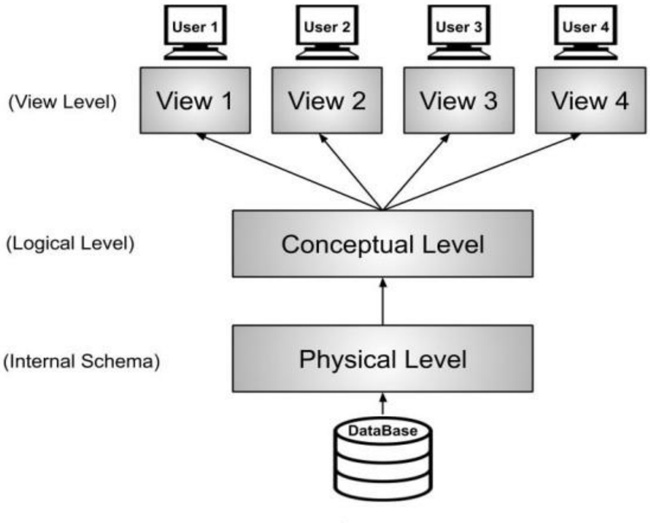


File System	DBMS
A file system is a software that manages and organizes the files in a storage medium. It controls how data is stored and retrieved.	DBMS or Database Management System is a software application. It is used for accessing, creating, and managing databases.
The file system provides the details of data representation and storage of data.	DBMS gives an abstract view of data that hides the details.
Storing and retrieving of data can't be done efficiently in a file system.	DBMS is an efficient to use as there are a wide variety of methods to store and retrieve data.
It does not offer data recovery processes.	There is a backup recovery for data in DBMS.
The file system doesn't have a crash recovery mechanism.	DBMS provides a crash recovery mechanism.
Protecting a file system is very difficult.	DBMS offers good protection mechanism.
In a file management system, the redundancy of data is greater.	The redundancy of data is low in the DBMS.
Data inconsistency is higher in the file system.	Data inconsistency is low in a DBMS.
The file system offers lesser security.	DBMS offers high security.
File system allows you to stores the data as isolated data files and entities.	DBMS stores data as well as defined constraints and interrelation.
Not provide support for complicated transactions.	Easy to implement complicated transactions.
The centralizations process is hard in File Management System.	Centralization is easy to achieve in the DBMS.
It doesn't offer backup and recovery of data if it is lost.	DBMS provides backup and recovery of data even if it is lost.
There is no efficient query processing in the file system.	You can easily query data in a database using the SQL language.
These system doesn't offer concurrency.	DBMS System provides a concurrency facility.

View Of Database

1. Data Abstraction
2. Instance and Schema
3. Data Models

Three level view of abstraction



Levels of Data Abstraction

Data Abstraction:

→ Abstraction means System hides certain detail of how the data are stored and maintained.

a. Physical level

→ It is the lowest level of abstraction for DBMS which defines how the data is actually stored, it defines data-structures to store data and access methods used by the database.

→ Actually, it is decided by developers or database application programmers how to store the data in the database.

b. Logical level:

→ Logical level is the intermediate level or next higher level.

→ It describes what data is stored in the database and what relationship exists among those data.

→ It tries to describe the entire or whole data because it describes what tables to be created and what are the links among those tables that are created.

→ Manages and views database schema Logical level is used by developers or database administrators (DBA).

c. External Level:

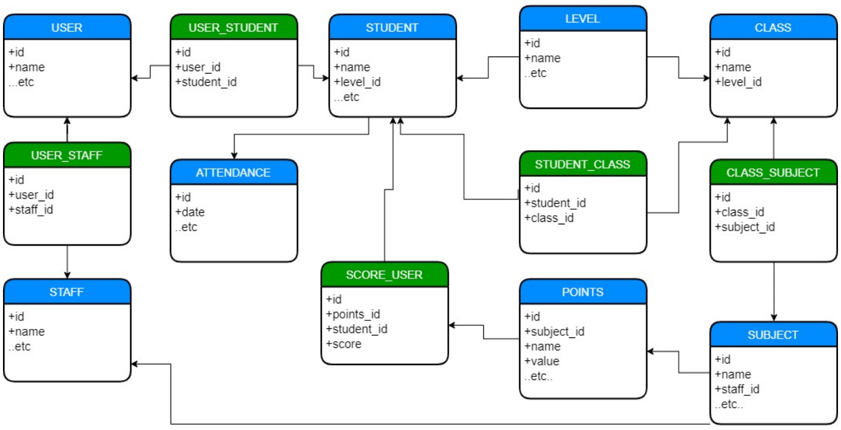
→ It is the highest level. In view level, there are different levels of views and every view only defines a part of the entire data.

→ It also simplifies interaction with the user and it provides many views or multiple views of the same database.

Instance and Schema

- → The data which is stored in the database at a particular moment of time is called an instance of the database.
- → The overall design of a database is called schema.
- → A database schema is the skeleton structure of the database. It represents the logical view of the entire database.
- → A schema contains schema objects like table, foreign key, primary key, views, columns, data types, stored procedure, etc.
- → A database schema can be represented by using the visual diagram. That diagram shows the database objects and their relationship with each other.
- → A database schema is designed by the database designers to help programmers whose software will interact with the database. The process of database creation is called data modeling.

Database Schema

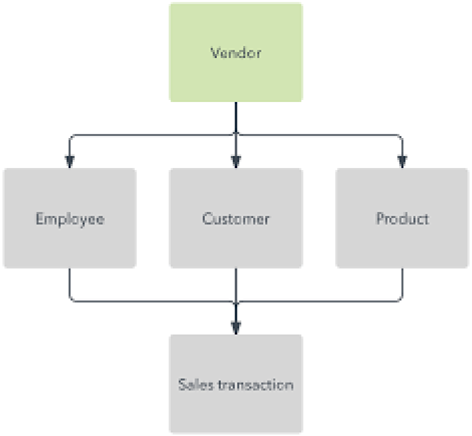


Database model

→ A database model shows the logical structure of a database, including the relationships and constraints that determine how data can be stored and accessed.

→ Individual database models are designed based on the rules and concepts of whichever broader data model the designers adopt.

→ Most data models can be represented by an accompanying database diagram.



Database Users

→ Database users are the ones who really use and take the benefits of the database. There will be different types of users depending on their needs and way of accessing the database.

1. Application Programmers

✓ They are the developers who interact with the database by means of DML queries. These DML queries are written in the application programs like C, C++, JAVA, Pascal, etc. These queries are converted into object code to communicate with the database. For example, writing a C program to generate the report of employees who are working in a particular department will involve a query to fetch the data from the database. It will include an embedded SQL query in the C Program.

2. Sophisticated Users

✓ They are database developers, who write SQL queries to select/insert/delete/update data. They do not use any application or programs to request the database. They directly interact with the database by means of a query language like SQL. These users will be scientists, engineers, analysts who thoroughly study SQL and DBMS to apply the concepts in their requirements. In short, we can say this category includes designers and developers of DBMS and SQL.

3. Specialized Users

- ✓ These are also sophisticated users, but they write special database application programs. They are the developers who develop the complex programs to the requirement.

4. **Stand-alone Users**

- ✓ These users will have a stand-alone database for their personal use. These kinds of the database will have readymade database packages which will have menus and graphical interfaces.

5. **Native Users**

- ✓ these are the users who use the existing application to interact with the database. For example, online library system, ticket booking systems, ATMs etc which has existing application and users use them to interact with the database to fulfill their requests.

Database Administrators:

The life cycle of a database starts from designing, implementing to the administration of it. A database for any kind of requirement needs to be designed perfectly so that it should work without any issues. Once all the design is complete, it needs to be installed. Once this step is complete, users start using the database. The database grows as the data grows in the database. When the database becomes huge, its performance comes down. Also accessing the data from the database becomes a challenge. There will be unused memory in the database, making the memory inevitably huge. This administration and maintenance of the database are taken care of by the database Administrator – DBA. A DBA has many responsibilities. A good-performing database is in the hands of DBA.

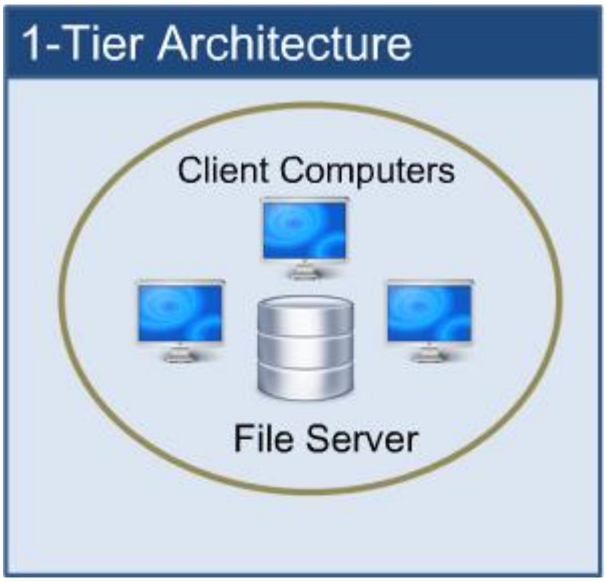
Database Architecture

- → Database architecture uses programming languages to design a particular type of software for businesses or organizations. Database architecture focuses on the design, development, implementation and maintenance of computer programs that store and organize information for businesses, agencies and institutions. A database architect develops and implements software to meet the needs of users.

- → The design of a DBMS depends on its architecture. It can be centralized or decentralized or hierarchical. The architecture of a DBMS can be seen as either single tier or multi-tier. The tiers are classified as follows:

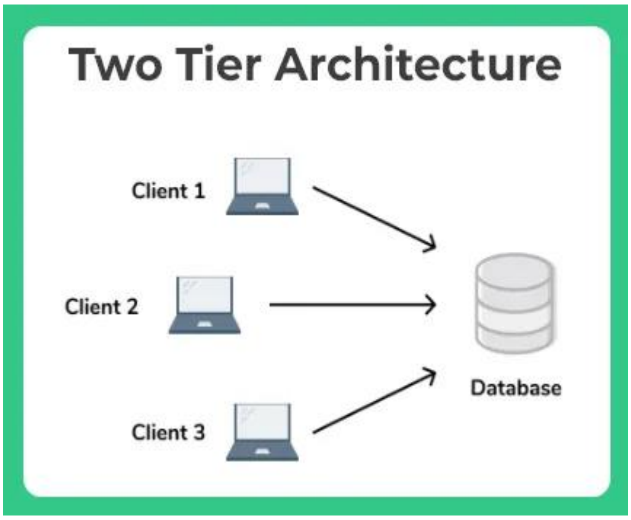
1. **One-tier architecture:**

- → involves putting all of the required components for a software application or technology on a single server or platform.
- → Basically, a one-tier architecture keeps all of the elements of an application, including the interface, Middleware and back-end data, in one place. Developers see these types of systems as the simplest and most direct way.



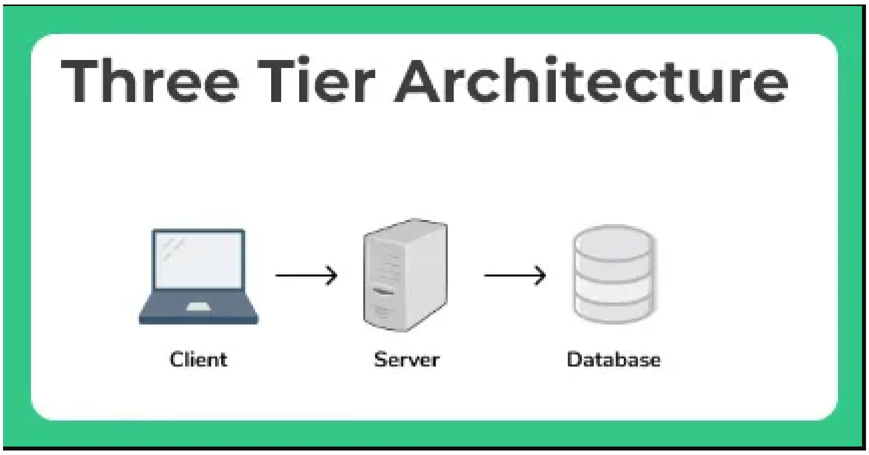
2. **Two-tier architecture:**

→ The two-tier is based on Client Server architecture. The two-tier architecture is like client server application. The direct communication takes place between client and server. There is no intermediate between client and server.



3. **Three-tier architecture:**

→ A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is the most widely used architecture to design a DBMS.



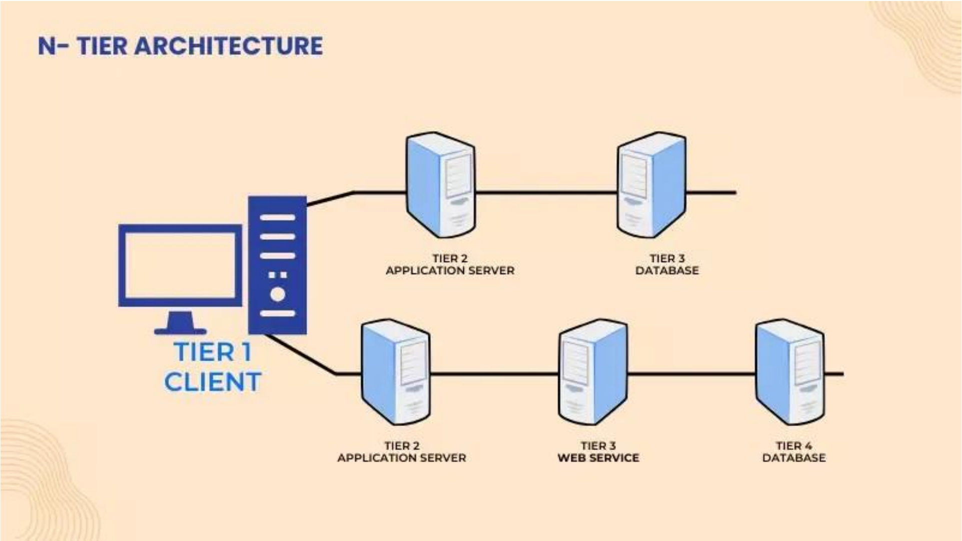
→ This architecture has different usages with different applications. It can be used in web applications and distributed applications. The strength in particular is when using this architecture over distributed systems.

- ♦ **Database (Data) Tier** − At this tier, the database resides along with its query processing languages. We also have the relations that define the data and their constraints at this level.
- ♦ **Application (Middle) Tier** − At this tier reside the application server and the programs that access the database. For a user, this application tier presents an abstracted view of the database. End-users are unaware of any existence of the database beyond the application. At the other end, the database tier is not aware of any other user beyond the application tier. Hence, the application layer sits in the middle and acts as a mediator between the end-user and the database.
- ♦ **User (Presentation) Tier** − End-users operate on this tier and they know nothing about any existence of the database beyond this layer. At this layer, multiple views of the database can be provided by the application. All views are generated by applications that reside in the application tier.

4. N-tier architecture

→ N-tier architecture would involve dividing an application into three different tiers. These would be the:

(a) logic tier,
(b) the presentation tier, and
(c) the data tier.



N- TIER ARCHITECTURE
TIER 1
CLIENT
TIER 2
APPLICATION SERVER
TIER 3
DATABASE
TIER 2
APPLICATION SERVER
TIER 3
WEB SERVICE
TIER 4
DATABASE