A database management system (or DBMS) is essentially nothing more than a computerized data-keeping system.

A database is **an organized collection of structured information, or data, typically stored electronically in a computer system**. A database is usually controlled by a database management system (DBMS).

## Centralized Database

It is the type of database that stores data at a centralized database system. It comforts the users to access the stored data from different locations through several applications. An example of a Centralized database can be Central Library that carries a central database of each library in a college/university.

### Advantages of Centralized Database

* It has decreased the risk of data management, i.e., manipulation of data will not affect the core data.
* Data consistency is maintained as it manages data in a central repository.
* It provides better data quality, which enables organizations to establish data standards.
* It is less costly because fewer vendors are required to handle the data sets.

Disadvantages of Centralized Database

* The size of the centralized database is large, which increases the response time for fetching the data.
* It is not easy to update such an extensive database system.
* If any server failure occurs, entire data will be lost, which could be a huge loss

**Some key features of a DBMS include:**

1. Data modeling: A DBMS provides tools for creating and modifying data models, which define the structure and relationships of the data in a database.
2. Data storage and retrieval: A DBMS is responsible for storing and retrieving data from the database, and can provide various methods for searching and querying the data.
3. Concurrency control: A DBMS provides mechanisms for controlling concurrent access to the database, to ensure that multiple users can access the data without conflicting with each other.
4. Data integrity and security: A DBMS provides tools for enforcing data integrity and security constraints, such as constraints on the values of data and access controls that restrict who can access the data.
5. Backup and recovery: A DBMS provides mechanisms for backing up and recovering the data in the event of a system failure.
6. DBMS can be classified into two types: Relational Database Management System (RDBMS) and Non-Relational Database Management System (NoSQL or Non-SQL)
7. RDBMS: Data is organized in the form of tables and each table has a set of rows and columns. The data is related to each other through primary and foreign keys.
8. NoSQL: Data is organized in the form of key-value pairs, document, graph, or column-based. These are designed to handle large-scale, high-performance scenarios.

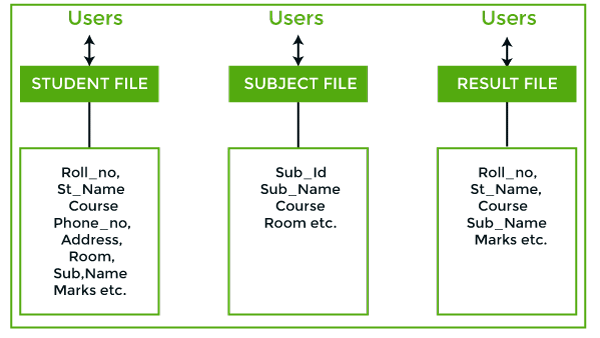
Benefits of the Database Approach

* The data can be shared
* Redundancy can be reduced.
* Inconsistency can be avoided (to some extent).
* Transaction support can be provided.
* Integrity can be maintained.
* Security’ can be enforced.
* Conflicting requirements can be balanced.
* Standards can be enforced

# Difference between File System and DBMS

## File System Approach

File based systems were an early attempt to computerize the manual system. It is also called a traditional based approach in which a decentralized approach was taken where each department stored and controlled its own data with the help of a data processing specialist. The main role of a data processing specialist was to create the necessary computer file structures, and also manage the data within structures and design some application programs that create reports based on file data.



**In the above figure:**

Consider an example of a student's file system. The student file will contain information regarding the student (i.e. roll no, student name, course etc.). Similarly, we have a subject file that contains information about the subject and the result file which contains the information regarding the result.

Some fields are duplicated in more than one file, which leads to data redundancy. So to overcome this problem, we need to create a centralized system, i.e. DBMS approach.

## DBMS:

A database approach is a well-organized collection of data that are related in a meaningful way which can be accessed by different users but stored only once in a system. The various operations performed by the DBMS system are: Insertion, deletion, selection, sorting etc.

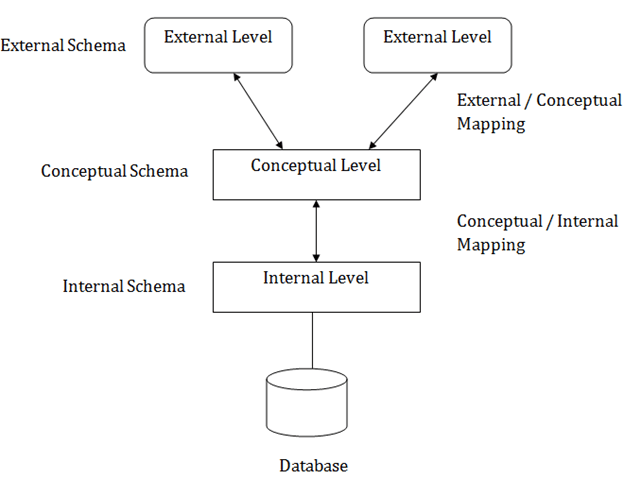
## DBMS vs. File System

**In the above figure,**

In the above figure, duplication of data is reduced due to centralization of data.

**There are the following differences between DBMS and File systems:**

|  |  |  |
| --- | --- | --- |
| **Basis** | **DBMS Approach** | **File System Approach** |
| **Meaning** | DBMS is a collection of data. In DBMS, the user is not required to write the procedures. | The file system is a collection of data. In this system, the user has to write the procedures for managing the database. |
| **Sharing of data** | Due to the centralized approach, data sharing is easy. | Data is distributed in many files, and it may be of different formats, so it isn't easy to share data. |
| **Data Abstraction** | DBMS gives an abstract view of data that hides the details. | The file system provides the detail of the data representation and storage of data. |
| **Security and Protection** | DBMS provides a good protection mechanism. | It isn't easy to protect a file under the file system. |
| **Recovery Mechanism** | DBMS provides a crash recovery mechanism, i.e., DBMS protects the user from system failure. | The file system doesn't have a crash mechanism, i.e., if the system crashes while entering some data, then the content of the file will be lost. |
| **Manipulation Techniques** | DBMS contains a wide variety of sophisticated techniques to store and retrieve the data. | The file system can't efficiently store and retrieve the data. |
| **Concurrency Problems** | DBMS takes care of Concurrent access of data using some form of locking. | In the File system, concurrent access has many problems like redirecting the file while deleting some information or updating some information. |
| **Where to use** | Database approach used in large systems which interrelate many files. | File system approach used in large systems which interrelate many files. |
| **Cost** | The database system is expensive to design. | The file system approach is cheaper to design. |
| **Data Redundancy and Inconsistency** | Due to the centralization of the database, the problems of data redundancy and inconsistency are controlled. | In this, the files and application programs are created by different programmers so that there exists a lot of duplication of data which may lead to inconsistency. |
| **Structure** | The database structure is complex to design. | The file system approach has a simple structure. |
| **Data Independence** | In this system, Data Independence exists, and it can be of two types.   * Logical Data Independence * Physical Data Independence | In the File system approach, there exists no Data Independence. |
| **Integrity Constraints** | Integrity Constraints are easy to apply. | Integrity Constraints are difficult to implement in file system. |
| **Data Models** | In the database approach, 3 types of data models exist:   * Hierarchal data models * Network data models * Relational data models | In the file system approach, there is no concept of data models exists. |
| **Flexibility** | Changes are often a necessity to the content of the data stored in any system, and these changes are more easily with a database approach. | The flexibility of the system is less as compared to the DBMS approach. |
| **Examples** | Oracle, SQL Server, Sybase etc. | Cobol, C++ etc. |



**In the above diagram:**

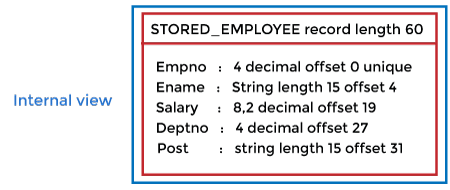
* It shows the DBMS architecture.
* Mapping is used to transform the request and response between various database levels of architecture.
* Mapping is not good for small DBMS because it takes more time.
* In External / Conceptual mapping, it is necessary to transform the request from external level to conceptual schema.
* In Conceptual / Internal mapping, DBMS transform the request from the conceptual to internal level.

## Objectives of Three schema Architecture

The main objective of three level architecture is to enable multiple users to access the same data with a personalized view while storing the underlying data only once. Thus it separates the user's view from the physical structure of the database. This separation is desirable for the following reasons:

* Different users need different views of the same data.
* The approach in which a particular user needs to see the data may change over time.
* The users of the database should not worry about the physical implementation and internal workings of the database such as data compression and encryption techniques, hashing, optimization of the internal structures etc.
* All users should be able to access the same data according to their requirements.
* DBA should be able to change the conceptual structure of the database without affecting the user's
* Internal structure of the database should be unaffected by changes to physical aspects of the storage.

1. Internal Level



* The internal level has an internal schema which describes the physical storage structure of the database.
* The internal schema is also known as a physical schema.
* It uses the physical data model. It is used to define that how the data will be stored in a block.
* The physical level is used to describe complex low-level data structures in detail.

The internal level is generally is concerned with the following activities:

* Storage space allocations.

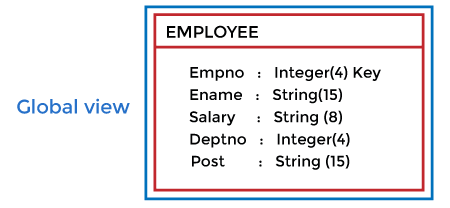
**For Example:** B-Trees, Hashing etc.

* Access paths.

**For Example:** Specification of primary and secondary keys, indexes, pointers and sequencing.

* Data compression and encryption techniques.
* Optimization of internal structures.
* Representation of stored fields.

 Conceptual Level



* The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level.
* The conceptual schema describes the structure of the whole database.
* The conceptual level describes what data are to be stored in the database and also describes what relationship exists among those data.
* In the conceptual level, internal details such as an implementation of the data structure are hidden.
* Programmers and database administrators work at this level.

3. External Level

DBMS Three schema Architecture

* At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database.
* An external schema is also known as view schema.
* Each view schema describes the database part that a particular user group is interested and hides the remaining database from that user group.
* The view schema describes the end user interaction with database systems.

## Mapping between Views

The three levels of DBMS architecture don't exist independently of each other. There must be correspondence between the three levels i.e. how they actually correspond with each other. DBMS is responsible for correspondence between the three types of schema. This correspondence is called Mapping.

**There are basically two types of mapping in the database architecture:**

* Conceptual/ Internal Mapping
* External / Conceptual Mapping

**Conceptual/ Internal Mapping**

The Conceptual/ Internal Mapping lies between the conceptual level and the internal level. Its role is to define the correspondence between the records and fields of the conceptual level and files and data structures of the internal level.

**External/ Conceptual Mapping**

The external/Conceptual Mapping lies between the external level and the Conceptual level. Its role is to define the correspondence between a particular external and the conceptual view.

## DATA INDEPENDENCE

* Data independence can be explained using the three-schema architecture.
* Data independence refers characteristic of being able to modify the schema at one level of the database system without altering the schema at the next higher level.

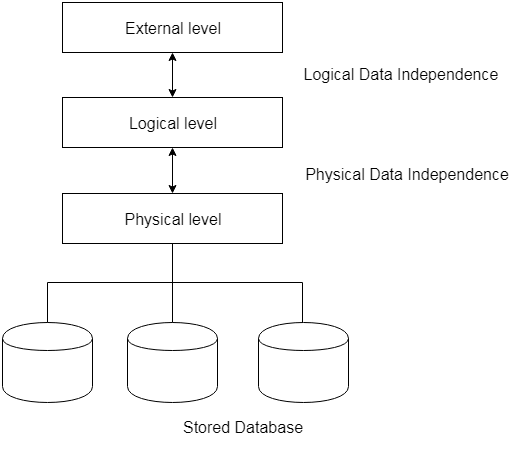
There are two types of data independence:

## 1. Logical Data Independence

* Logical data independence refers characteristic of being able to change the conceptual schema without having to change the external schema.
* Logical data independence is used to separate the external level from the conceptual view.
* If we do any changes in the conceptual view of the data, then the user view of the data would not be affected.
* Logical data independence occurs at the user interface level.

## 2. Physical Data Independence

* Physical data independence can be defined as the capacity to change the internal schema without having to change the conceptual schema.
* If we do any changes in the storage size of the database system server, then the Conceptual structure of the database will not be affected.
* Physical data independence is used to separate conceptual levels from the internal levels.
* Physical data independence occurs at the logical interface level.



**Fig: Data Independence**