**Database Approach**

1. Manual Filing System
2. Traditional File Based Approach
3. Database Approach

* **The Manual Filing System**

The manual filing system works well when the number of items to be stored is small. It even works quite adequately when there are large numbers of items and we have only to store and retrieve them. However, the manual filing system breaks down when we have to cross-reference or process the [information](https://ecomputernotes.com/fundamental/information-technology/what-do-you-mean-by-data-and-information) in the files. For example, a typical real estate agent’s office might have a separate file for each property for sale or rent, each potential buyer and renter, and each member of staff.

* **Traditional File Processing System**

File processing systems was an early attempt to computerize the manual filing system that we are all familiar with. A file system is a method for storing and organizing [computer](https://ecomputernotes.com/fundamental/introduction-to-computer/what-is-computer) files and the data they contain to make it easy to find and access them. File systems may use a storage device such as a hard disk or CD-ROM and involve maintaining the physical location of the files. In this approach, information is stored in flat files which are maintained by the file system under the [operating system](https://ecomputernotes.com/fundamental/disk-operating-system/what-is-operating-system)’s control. Flat files are files containing records having no structured relationship among them. The file handling which we learn under C/C ++ is the example of file processing system. The Application programs written in C/C ++ like programming languages go through the file system to access these flat.

**Characteristics of File Processing System**

* It is a group of files storing data of an organization.
* Each file is independent from one another.
* Each file is called a flat file.
* Each file contained and processed information for one specific function, such as accounting or inventory.
* Files are designed by using programs written in programming languages such as COBOL, C, C++.
* The physical implementation and access procedures are written into [database](https://ecomputernotes.com/fundamental/what-is-a-database/advantages-and-disadvantages-of-dbms) application; therefore, physical changes resulted in intensive rework on the part of the programmer.
* As systems became more complex, file processing systems offered little flexibility, presented many limitations, and were difficult to maintain.

## Limitations of the File Processing System I File-Based Approach

**1.** **Separated and Isolated Data:** To make a decision, a user might need data from two separate files. First, the files were evaluated by analysts and programmers to determine the specific data required from each file and the relationships between the data and then applications could be written in a programming language to process and extract the needed data. Imagine the work involved if data from several files was needed.

**2.** **Duplication of data:** Often the same information is stored in more than one file. Uncontrolled duplication of data is not required for several reasons, such as:

• Duplication is wasteful. It costs time and money to enter the data more than once

• It takes up additional storage space, again with associated costs.

**3. Data Dependence:** In file processing systems, files and records were described by specific physical formats that were coded into the application program by programmers. If the format of a certain record was changed, the code in each file containing that format must be updated. Furthermore, instructions for data storage and access were written into the application’s code. Therefore, .changes in storage structure or access methods could greatly affect the processing or results of an application.

**4. Difficulty in representing data from the user’s view:** To create useful applications for the user, often data from various files must be combined. In file processing it was difficult to determine relationships between isolated data in order to meet user requirements.

**5. Data Inflexibility:** Program-data inter dependency and data isolation, limited the flexibility of file processing systems in providing users with ad-hoc information requests

**6. Incompatible file formats:** As the structure of files is embedded in the application programs, the structures are dependent on the application programming language. For example, the structure of a file generated by a COBOL program may be different from the structure of a file generated by a ‘C’ program. The direct incompatibility of such files makes them difficult to process jointly.

**7. Data Security.** The security of data is low in file based system because, the data is maintained in the flat file(s) is easily accessible. For Example: Consider the Banking System. The Customer Transaction file has details about the total available balance of all customers. A Customer wants information about his account balance. In a file system it is difficult to give the Customer access to only his data in the· file. Thus enforcing security constraints for the entire file or for certain data items are difficult.

**8. Transactional Problems.** The File based system approach does not satisfy transaction properties like Atomicity, Consistency, Isolation and Durability properties commonly known as ACID properties.

For example: Suppose, in a banking system, a transaction that transfers Rs. 1000 from account A to account B with initial values’ of A and B being Rs. 5000 and Rs. 10000 respectively. If a system crash occurred after the withdrawal of Rs. 1000 from account A, but before depositing of amount in account B, it will result an inconsistent state of the system. It means that the transactions should not execute partially but wholly. This concept is known as Atomicity of a transaction (either 0% or 100% of transaction). It is difficult to achieve this property in a file based system.

**9. Concurrency problems.** When multiple users access the same piece of data at same interval of time then it is called as concurrency of the system. When two or more users read the data simultaneously there is no problem, but when they like to update a file simultaneously, it may result in a problem.

**10. Poor data modeling of real world**. The file based system is not able to represent the complex data and interfile relationships, which results poor data modeling properties.

* **Database Approach**

In order to remove all limitations of the File Based Approach, a new approach was required that must be more effective known as [Database](https://ecomputernotes.com/fundamental/what-is-a-database/advantages-and-disadvantages-of-dbms) approach.

The [Database](https://ecomputernotes.com/fundamental/what-is-a-database/advantages-and-disadvantages-of-dbms) is a shared collection of logically related data, designed to meet the [information](https://ecomputernotes.com/fundamental/information-technology/what-do-you-mean-by-data-and-information) needs of an organization.

A database is a [computer](https://ecomputernotes.com/fundamental/introduction-to-computer/what-is-computer) based record keeping system whose over all purpose is to record and maintains [information](https://ecomputernotes.com/fundamental/information-technology/what-do-you-mean-by-data-and-information).

The database is a single, large repository of data, which can be used simultaneously by many departments and users. Instead of disconnected files with redundant data, all data items are integrated with a minimum amount of duplication.

The database is no longer owned by one department but is a shared corporate resource. The database holds not only the organization’s operational data but also a description of this data. For this reason, a database is also defined as a **self-describing collection of integrated records.** The description of the data is known as the Data Dictionary or Meta Data (the ‘data about data’). It is the self-describing nature of a database that provides program-data independence.

## Building blocks of a Database

The following three components form the building blocks of a database. They store the data that we want to save in our database.

**Columns.** Columns are similar to fields, that is, individual items of data that we wish to store. A Student’ Roll Number, Name, Address etc. are all examples of columns. They are also similar to the columns found in spreadsheets (the A, B, C etc. along the top).

**Rows.** Rows are similar to records as they contain data of multiple columns (like the 1, 2, 3 etc. in a spreadsheet). A row can be made up of as many or as few columns as you want. This makes reading data much more efficient – you fetch what you want.

**Tables.** A table is a logical group of columns. For example, you may have a table that stores details of customers’ names and addresses. Another table would be used to store details of parts and yet another would be used .for supplier’s names and addresses.

It is the tables that make up the entire database and it is important that we do not duplicate data at all.

## Building blocks of a Database

Assignment

1. Write notes on the Evolution of Database Systems.