Topics to be covered:

- Introduction, Characteristics of Databases
- File system V/s Database system

Learning Outcomes:

- Differentiate between data and information.
- Define the following key terms: database, Database Management System (DBMS), data inconsistency, data integrity, Atomicity, Concurrent –access anomalies
- Familiar with basic concepts and appreciate the applications of database systems.
- Explain the importance of database design.
- Define the term data redundancy and explain its effects on the quality of information produced.
- Explain how a database system differs from a file system.

What is a data and information?

Data

- Data means known facts that can be recorded for the future use and that have implicit meaning.
- For example, consider the names, telephone numbers and addresses of people you know.
- Data is raw, unorganized facts that need to be processed. Data can be something simple and seemingly random and useless until it is organized.
- Example: Each student's test score is one piece of data

Information

- Information is the processed data. That means, once the data that has been put into a meaningful and useful context which can be communicated to a recipient who uses it to make decisions.
- When data is processed, organized, structured or presented in a given context so as to make it useful, it is called Information.
- The class' average score or the school's average score is the information that can be concluded from the given data.

Question *Can you think for few more examples. Think about some data and after processing how it becomes information.*

Levels of Information

Within an organization planning, control and decision-making is carried out at various levels within the structure of the organization.

The three levels at which information can be used are strategic, tactical and operational and there is a direct correlation between the levels of importance of individuals or groups within an organization and the level of information that is being communicated.

TYPES OF INFORMATION

Strategic information

Strategic information is used at the very top level of management within an organization. These are chief executives or directors who have to make decisions for the long term.

Strategic information is broad based and will use a mixture of information gathered from both internal and external sources.

In general a timescale may be from one to five years or even longer depending on the project.

Examples: Some oil related projects are planned from the outset to last for 25 or more years. A supermarket building a new superstore will look at a timescale of 20 years or so, whilst even a small business may have a five-year strategy.

Strategic plans will have little or no detail in them and more detailed strategic plans will be made slightly lower down the managerial ladder. A good strategic plan will be easier to flesh out lower down than a poor or vague strategic plan. Similarly, well constructed and more detailed plans will be easier to implement than poorly constructed plans.

Tactical information

The next level down is the tactical level, and tactical planning and decision-making takes place within the guidelines set by the strategic plan.

Tactical information will be mostly internal with a few external sources being used. Internal information is likely to be function related: for example, how much ‘down time’ a production line must allocate for planned maintenance. Tactical information is used by middle management (employees) when managing or planning projects. The timescale is usually at least between 6 months and 5 years (depending on the scale of the strategic project). Circumstances vary but a small project may have a tactical timescale of between one and six months. Tactical plans have a medium level of detail and will be very specific; they deal with such matters as who is doing what and within what specific budgets and timescales.

These plans have medium scope and will address details at the operational level. They will generally have specific objectives and be geared towards implementation by operational level employees.

Operational information

The lowest level is operational and operational planning takes place based on the tactical plans. The lowest level of management or workers in an organization implements operational plans. These may be section leaders or foremen in a large organization or workers such as shop assistants, waiting staff, and kitchen staff, etc., in smaller businesses where there is no supervisory layer.

The timescale is usually very short, anything from immediately, daily or at most a week or month. Results of operational work will usually be passed upwards to let the tactical planners evaluate their plans.

What is Database?

A database is a collection of related data organized for convenient access. The diagram representation for a database is a cylinder.

Now, answer the following questions in YES/NO form:

- How many of you have ever kept details of your friends or families birthdays in a birthday book?
- How many of you have written your friend's addresses into an address book?
- How many of you have a copy of the Yellow Pages at home?
- How many of you are using dictionary?

If you have answered YES to any of these questions, then you have used a 'paper-based' database in your daily life.
- Birthdays were organised in logical order of 'Month'.
- Addresses were stored in logical order of 'Name'.
- Telephone numbers were stored in logical order of 'profession' e.g. gardener, builder, plumber.
- All English words with thier meaning are arranged in alphabatical order.

These examples were of paper-based databases.However, when we use the term 'database' we generally think of a computerised database.

Information	*Yellow Pages* in Mumbai Find Information Directories Phone Numbers, Addresses, Best Deals, Latest Reviews & Ratings.

Database examples

Maybe you don't realize it but you probably use a database almost everyday of your life. Lets have a look of few examples:

1. *Search Engine*

You will no doubt be familiar with using a search engine. Examples of search engines include Google, Altavista, Dogpile and Bing.

Search engines work by cataloging all of the content on every webpage on the internet. When you enter a keyword search into a search engine it will find every result that matches your terms and present them in the form of a list to you.

2. *Colmation system in DBIT*

Don Bosco Institute of Technology has electronic registration system named as Colmation. The respective subject teacher might require to record attendance on the computer.

The attendance records are stored in a database. Every student will have their own individual record in the database and it will be updated every time data is entered by the subject teacher.

The database can be searched or sorted for data about the attendance of individual students, subject-wise attendance or attendance for a particular week/term or overall attendance. This system also generates defaulter students list.

3. Library database

A library stores details of all their books in a database. When you want to know if a book is in stock you can enter either the title, author or ISBN number and search for information about the book. You can find out how many copies are stored in your library. You can check when the book is due to be returned and also reserve it. The database also records details of all the borrowers, what books they currently have out on loan and when they are due back. When they return their books the librarian will be informed if they are overdue and whether there are any fines outstanding.

Question *Can you think for few more database examples which you probably use in your daily life ?*

Computerized vs non-computerized databases

What is it that makes computerized databases much more popular than paper-based ones?

Computerized Database	Paper-Based Database
Can hold a vast amount of data	Limited by physical storage space available
Very fast to find a specific record	Can take a while to manually search through all of the records
Can easily search for a specific criteria e.g. "all of the people who live in Mumbai"	Difficult to search for a specific criteria; every record would have to be manually looked at.
Can be used to analyse the data e.g. 'most popular selling item'	Very difficult to analyse the data
Data can be sorted into ascending or descending order on multiple criteria	Difficult to sort data on more than one criteria.
Can easily update or amend a record e.g. customer's address after moving house	Changes have to be done manually. Records can look messy if scribbled out.
Records are stored safely, they are available when needed	Records can be lost or misfiled making it hard to find them
The database can be kept secure by use of passwords	The only security would be locking up the records.

Easy to make a back-up in case of data loss	Difficult to make a back up because every page/card would have to be re-written or photocopied. This means extra storage space is needed.

Database:

A database is a collection of related data.
A database has the following implicit properties:

- A database represents some aspect of the real world, sometimes called the **mini world** or the **universe of discourse (UoD)**. Changes to the mini world are reflected in the database.
- A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

A database-management system

(DBMS) is a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred to as the **database**, contains information relevant to an enterprise. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both *convenient* and *efficient*.

Database System Applications

Databases are widely used. Here are some representative applications:

- *Banking*: For customer information, accounts, and loans, and banking transactions.
- *Airlines*: For reservations and schedule information. Airlines were among the first to use databases in a geographically distributed manner—terminals situated around the world accessed the central database system through phone lines and other data networks.
- *Universities*: For student information, course registrations, and grades.
- *Credit card transactions*: For purchases on credit cards and generation of monthly statements.
- *Telecommunication*: For keeping records of calls made, generating monthly bills, maintaining balances on prepaid calling cards, and storing information about the communication networks.
- *Finance*: For storing information about holdings, sales, and purchases of financial instruments such as stocks and bonds.
- *Sales*: For customer, product, and purchase information.
- *Manufacturing*: For management of supply chain and for tracking production of items in factories, inventories of items in warehouses/stores, and orders for items.
- *Human resources*: For information about employees, salaries, payroll taxes and benefits, and for generation of paychecks.

FILE SYSTEMS Versus a DBMS (Bank Example)

Consider part of a savings-bank enterprise that keeps information about all customers and savings accounts. One way to keep the information on a computer is to store it in operating system files. To allow users to manipulate the information, the system has a number of application programs that manipulate the files, including:

- A program to debit or credit an account
- A program to add a new account
- A program to find the balance of an account
- A program to generate monthly statements
- A program to generate list of all customers

System programmers wrote these application programs to meet the needs of the bank. New application programs are added to the system as the need arises. For example, suppose that the savings bank decides to offer checking accounts. As a result, the bank creates new permanent files that contain information about all the checking accounts maintained in the bank, and it may have to write new application programs to deal with situations that do not arise in savings accounts, such as overdrafts. Thus, as time goes by, the system acquires more files and more application programs. This typical **file-processing system** is supported by a conventional operating system. The system stores permanent records in various files, and it needs different application programs to extract records from, and add records to, the appropriate files. Before database management systems (DBMSs) came along, organizations usually stored information in such systems. Keeping organizational information in a file-processing system has a number of major disadvantages:

- **Data redundancy and inconsistency**. Since different programmers create the files and application programs over a long period, the various files are likely to have different formats and the programs may be written in several programming languages. Moreover, the same information may be duplicated in several places (files). For example, the address and telephone number of a particular customer may appear in a file that consists of savings-account records and in a file that consists of checking-account records. This redundancy leads to higher storage and access cost. In addition, it may lead to **data inconsistency**; that is, the various copies of the same data may no longer agree. For example, a changed customer address may be reflected in savings-account records but not elsewhere in the system.

- **Difficulty in accessing data**. Suppose that one of the bank officers needs to find out the names of all customers who live within a particular postal-code area. The officer asks the data-processing department to generate such a list. Because the designers of the original system did not anticipate this request, there is no application program on hand to meet it. There is, however, an application program to generate the list of *all* customers. The bank officer has now two choices: either obtain the list of all customers and extract the needed information manually or ask a system programmer to write the necessary application program. Both alternatives are obviously unsatisfactory. Suppose that such a program is written, and that, several days later, the same officer needs to trim that list to include only those customers who have an account balance of $10,000 or more. As expected, a program to generate such a list does not exist. Again, the officer has the preceding two options, neither of which is satisfactory.
  The point here is that conventional file-processing environments do not allow needed data to be retrieved in a convenient and efficient manner. More responsive data-retrieval systems are required for general use.

- **Data Isolation**: Since the data files are created at different times and supposedly by different people the structures of different files generally will not match. The data will be scattered in different files for a particular entity. So it will be difficult to obtain appropriate data.
  For example: Suppose the Address in Saving Account file have fields: ***Add line1, Add line2, City, State, Pin*** while the fields in address of Current account are: ***House No., Street No., Locality*, City, State, Pin**. Administrator is asked to provide the list of customers living in a particular locality. Providing consolidated list of all the customers will require looking in both files. But they both have different way of storing the address. Writing a program to generate such a list will be difficult.

- **Integrity problems**. The data values stored in the database must satisfy certain types of **consistency constraints**. For example, the balance of a bank account may never fall below a prescribed amount (say, $25). Developers enforce these constraints in the system by adding appropriate code in the various application programs. However, when new constraints are added, it is difficult to change the programs to enforce them. The problem is compounded when constraints involve several data items from different files.

- **Atomicity problems**. A computer system, like any other mechanical or electrical device, is subject to failure. In many applications, it is crucial that, if a failure occurs, the data be restored to the consistent state that existed prior to the failure. Consider a program to transfer $50 from account *A* to account *B*. If a system failure occurs during the execution of the program, it is possible that the $50 was removed from account *A* but was not credited to account *B*, resulting in an inconsistent database state. Clearly, it is essential to database consistency that either both the credit and debit occur, or that neither occur. That is, the funds transfer must be *atomic*—it must happen in its entirety or not at all. It is difficult to ensure atomicity in a conventional file-processing system.

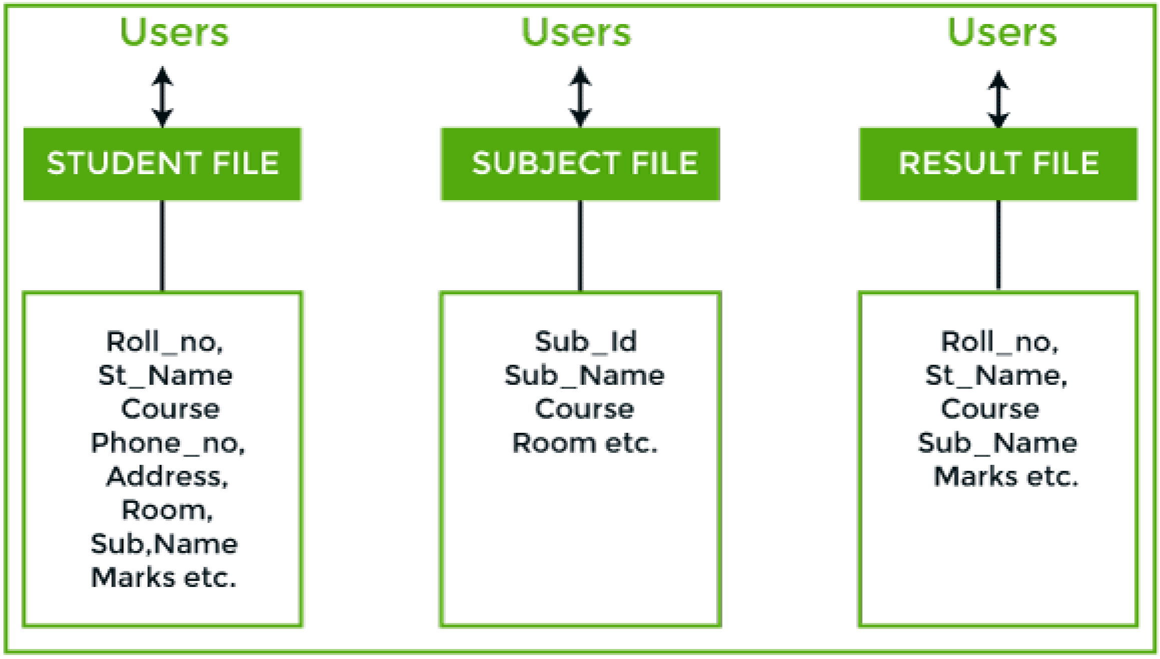
- **Concurrent-access anomalies**. For the sake of overall performance of the system and faster response, many systems allow multiple users to update the data simultaneously. In such an environment, interaction of concurrent updates may result in inconsistent data. Consider bank account *A*, containing $500. If two customers withdraw funds (say $50 and $100 respectively) from account *A* at about the same time, the result of the concurrent executions may leave the account in an incorrect (or inconsistent) state. Suppose that the programs executing on behalf of each withdrawal read the old balance, reduce that value by the amount being withdrawn, and write the result back. If the two programs run concurrently, they may both read the value $500, and write back $450 and $400, respectively. Depending on which one writes the value last, the account may contain either $450 or $400, rather than the correct value of $350. To guard against this possibility, the system must maintain some form of supervision. But supervision is difficult to provide because data may be accessed by many different application programs that have not been coordinated previously.

- **Security problems**. Not every user of the database system should be able to access all the data. For example, in a banking system, payroll personnel need to see only that part of the database that has information about the various bank employees. They do not need access to information about customer accounts. But, since application programs are added to the system in an ad hoc manner, enforcing such security constraints is difficult.

FILE SYSTEMS Versus a DBMS (College Example)

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.



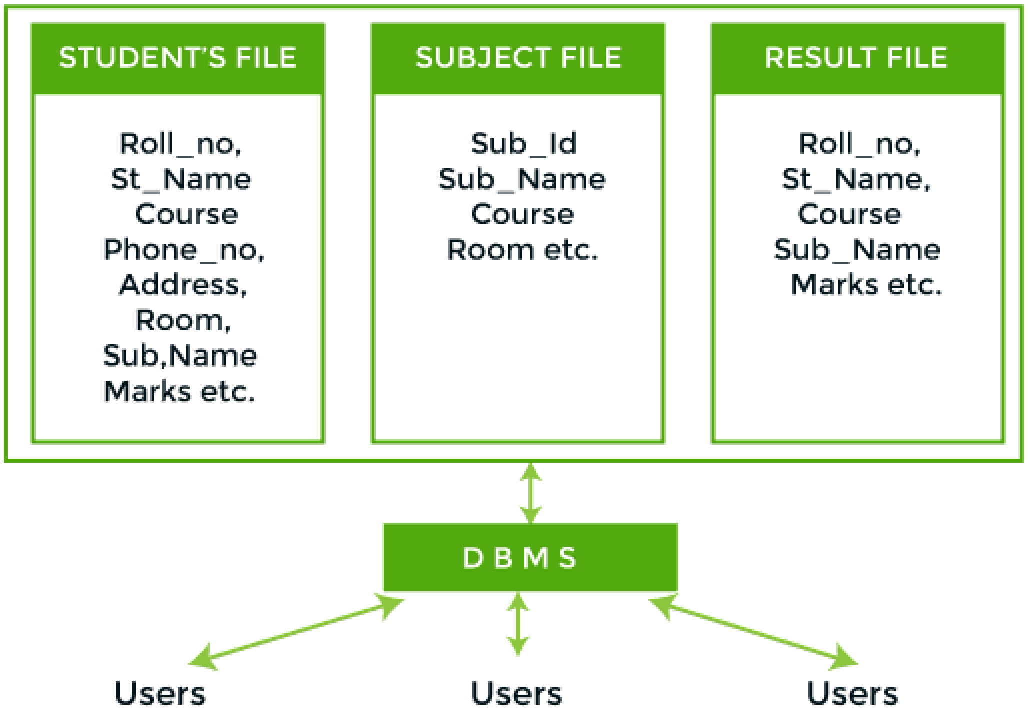
In the above figure:

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.



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	DBMS contains a wide variety of	The file system can't efficiently store

Techniques	sophisticated techniques to store and retrieve the data.	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.

Source:

1. Silberschatz−Korth−Sudarshan:
   Database SystemConcepts,
   Fourth Edition
2. Raghu Ramakrishnan / Johannes Gehrke
   Database Management System
   Second Edition
3. Ramez Elmasri, Shamkant B. Navathe
   Fundamentals of Database System
   Fourth Edition
4. http://www.diffen.com/difference/Data_vs_Information accessed on - 02/01/14
5. Rini Chakrabarti, Shilbhadra Dasgupta
   Advance Database Management System.
   DreamTech Press
6. http://www.cl500.net/pros_cons.html accessed on - 02/01/14
7. http://navdeep19.blogspot.in/2012/04/advantages-and-disadvantages-of.html accessed on – 02/01/14
8. http://www.jhigh.co.uk/Intermediate2/Using%20Information/6_levels_of_info.html accessed on – 05/01/15
9. https://www.javatpoint.com/dbms-vs-files-system accessed on – 13/01/22

Long Answer Type Questions:

1. List disadvantages of the file processing system. Explain how database can overcome this.
2. List four significant differences between file processing system and Database Management System.
3. Give the advantages of DBMS over File system.
4. This lecture has described several major advantages of a database system. What are two disadvantages?
5. Give examples of systems in which it may make sense to use traditional file processing instead of a database approach.
6. Define the following terms:
   data, database, DBMS, RDBMS.