UNIT I (**Overview of Database Systems**)

Notes 1.1

Overview of Database Management: (Examples)

Enterprise Information

- Sales: For customer, product, and purchase information.
- Accounting: For payments, receipts, account balances, assets and other accounting information.
- Human resources: For information about employees, salaries, payroll taxes, and benefits, and for generation of paychecks.
- Manufacturing: For management of the supply chain and for tracking production of items in factories, inventories of items in warehouses and stores, and orders for items.
- Online retailers: For sales data noted above plus online order tracking, generation of recommendation lists, and maintenance of online product evaluations.

Universities

- student information
- course registrations, and
- grades (in addition to human resources and accounting)

Airlines

- For reservations and schedule information.

Telecommunication

- For keeping records of calls made
- generating monthly bills
- maintaining balances on prepaid calling cards, and
- storing information about the communication networks

What is Database Management System?

Definition

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.

Purpose

a. **The primary goal of a DBMS** is to provide a way to:
   - Store and
   - Retrieve database information that is both convenient and efficient.

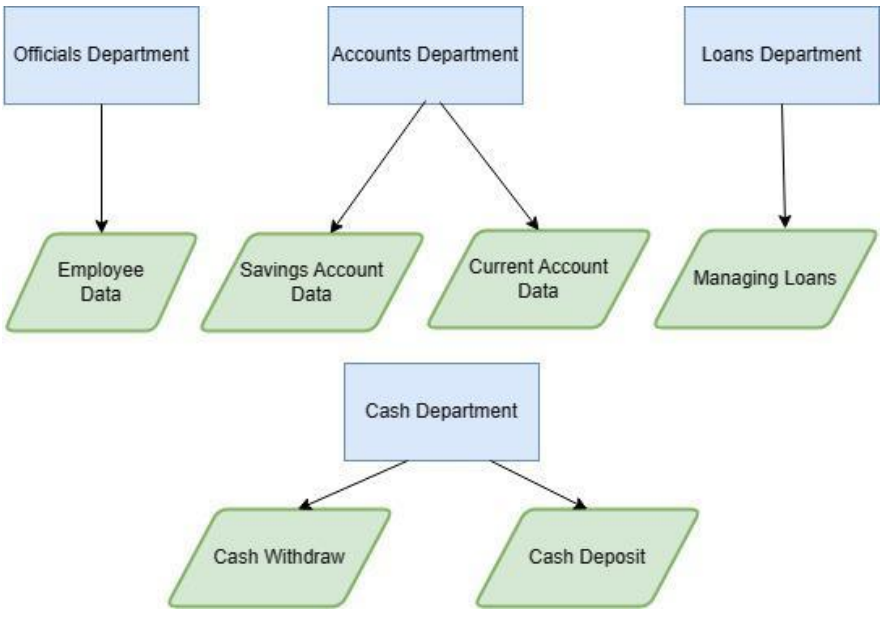
b. **Management of data involves**:
   - **Defining structures for storage** of information
   - **Providing mechanisms** for the manipulation of information.
   - The database system must ensure the **safety of the information stored**, despite system crashes or attempts at unauthorized access.
   - If data are to be shared among several users, the system must avoid possible anomalous results.

What is the Traditional "File System"?

- It is a manual way of storing data as "Files".

Example: Banking Data

Considering a scenario of a bank before the introduction of DBMS, for example, say someone went to the bank to deposit a certain amount in their account. So as the DBMS is not available so the bank employee has to manually register their account number, name, and amount in either a written manner or type and store them locally in the computer as a file. Manual errors may occur while writing any digit of their account number.



Characteristics of Traditional File Systems

Following are some of the notable characteristics of Traditional File Systems:

- **The data** of certain companies or organizations **were kept as "Files" also called flat files.**
- **The files stored in different departments** were **independent** of each other, which caused severe data redundancy.
- Those files were developed using programming languages like COBOL, C, and C++.
- The maintenance of those files was also of high cost.
- NTFS(New Technology File System), EXT(Extended File System).
- The **file system** is basically a way **of arranging the files in a storage medium** like a hard disk.
- The file system organizes the files and helps in the retrieval of files when they are required. File systems consist of different files which are grouped into directories. The directories further contain other folders and files. The file system performs basic operations like management, file naming, giving access rules, etc.

Disadvantages of the Traditional File Systems

The following issues are related to the file-based approach:

- **Distinguished and Isolated Data:** Imagine a user needs information that is not possible to be provided using a single file, multiple different files were required, which are situated in different departments. So all the employees first need to manually and carefully check each of the files in each department and find the relationship between them to interpret the information that the user wants.

- **Data Duplication / Data Redundancy –** Due to the manual storing of data, the same data used to be present in multiple locations using the space in each of the Hard Disk, below are some following reasons for which Data Redundancy can be a major issue
    - Storing the same data multiple times not only wastes resources in every machine but also is costly to maintain and wastes time.
    - **Loss of data integrity** is another major issue of Data Redundancy, imagine someone's address is present in multiple systems and he has applied to update the address, in one system the address gets updated but in the rest of them it remains the same, so the if someone from any different department where the data is not updated tries to send them any letter or something then it would go to a wrong address.

- **Dependence on Data –** Files and information were stored in a certain specific format in files which is hard coded by programmers in languages like C/C++, COBOL, etc. So if any of the file's format changes then the programmers need to update the code every time and the

format of every piece of data stored in that file will be changed, which is a rigorous task for programmers.

- **Data Protection –** Data protection was very less due to different reasons like Data Redundancy, manual storing of data, easy access of confidential data by unauthorized parties, etc.

- **Issues with Transactions –** It didn't follow the ACID (Atomicity, Consistency, Isolation, and Durability) properties, for that if in the middle of any transaction the system crashed then it would leave the system in an inconsistent state.

- **Concurrent issues –** Two or more users can view the same file simultaneously, but the problem arises when they try to update the same file simultaneously.

Advantages of using Database System in place of Traditional File Systems

- **Redundancy and inconsistency in the data –** The data redundancy problem of the Traditional File System has been minimized by the *DBMS, here one piece of data can be present in a single place* only without scattering throughout multiple places

- **Concurrent Data –** DBMS provides a locking system using which one or more users can only read the same data but can't make any changes simultaneously.

- **Searching Data –** DBMS allows easy retrieval of information using a query language.

- **System Failure –** As DBMS follows ACID properties, even if a system failure happens in between a transaction, nothing will be lost, and we can restart that transaction from its previous stable state.

- **Data Protection, Backups and, Maintenance –** DBMS comes up with lots of methods to protect the data stored inside it rather than just Passwords. It is efficient in taking backups and maintaining a database.

* * *