Database Management System (DBMS)

Unit- 1

Introduction to DBMS:

What is Data?

- Data is a collection of raw, unorganised facts and details like text, observations, figures, symbols, and descriptions of things etc.
- In other words, **data does not carry any specific purpose and has no significance by itself**.

Types of Data

1. Quantitative

   i) Numerical form

   ii) Weight, volume, cost of an item

2. Qualitative

   i) Descriptive, but not numerical

   ii) Name, gender, hair color of a person

What is information?

- Information is processed, organised, and structured data.
- It provides context of the data and enables decision making.
- E.g.,you have data of all the people living in your locality, its Data, when you analyse and interpret the data and come to some conclusion that:
  - ➔ There are 10 senior citizens
  - ➔ There are 20 married people
  - ➔ There are 10 childrens

Difference b/w Data and Information

Data	Information
Data is the raw fact.	It is a processed form of data.
It is not significant to a business.	It is significant to a business.
Data is an atomic level piece of information.	It is a collection of data.
Understanding is difficult.	Understanding is easy.
Example: Product name, Name of student.	Example: Report card of student.

What is Database?

- A database is a collection of data that is organised, which is also called structured data. It can be accessed or stored in a computer system.
- To make real use of Data, we need Database management systems.

What is DBMS?

- A database-management system (DBMS) is a collection of interrelated data and a set of programs to access that 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.
- DBMS accepts the request from the application and provides specific data through the operating system

Applications of DBMS

- Banking: all transactions
- Airlines: reservations, schedules

- Universities: registration, grades
- Sales: customers, products, purchases
- Online retailers: order tracking, customised recommendations
- Manufacturing: production, inventory, orders, supply chain
- Human resources: employee records, salaries, tax deductions

❖ Advantages of DBMS:

➔ Data Independence.
➔ Efficient Data Access.
➔ Data Integrity and security.
➔ Data administration.
➔ Concurrent access and Crash recovery.
➔ Reduced Application Development Time.

❖ Disadvantages of DBMS:

➔ Increased cost
    -> cost of hardware and software
    -> cost of staff training
    -> cost of data conversion
➔ Complexity
➔ Performance

File Processing System and its Limitations

- In Computer Science, File Processing System (FPS) is a way of storing, retrieving and manipulating data which is present in various files.
- Files are used to store various documents. All files are grouped based on their categories.
- The file names are very related to each other and arranged properly to easily access the files.

- The file names are very related to each other and arranged properly to easily access the files.

- The file system doesn't have a crash recovery mechanism on the other hand, DBMS provides a crash recovery mechanism.

Approach:

- ➔ Each user maintains separate files—and programs to manipulate these files—because each requires some data not available from the other user's files.
- ➔ In traditional file processing, data definition is typically part of the application programs themselves; hence, these programs are constrained to work with only one specific database, whose structure is declared in the application programs.

Limitations:

- ➢ Separated and Isolated Data
- ➢ Duplication of data
- ➢ Data Dependence
- ➢ Difficulty in representing data from the user's view
- ➢ Data Inflexibility
- ➢ Incompatible file formats

ANSI/SPARC Model or Three Level Architecture

- ANSI - American National Standard Institute
- SPARC - Standards Planning And Requirement Committee
- The ANSI/SPARC model is a three-level database architecture with a hierarchy of levels, from the users and their applications at the top, down to the physical storage of data at the bottom.
- The three-level architecture aims to separate each user's view of the database from the way the database is physically represented[1]
- The three schema architecture is also used to separate the user applications and physical database.[2]

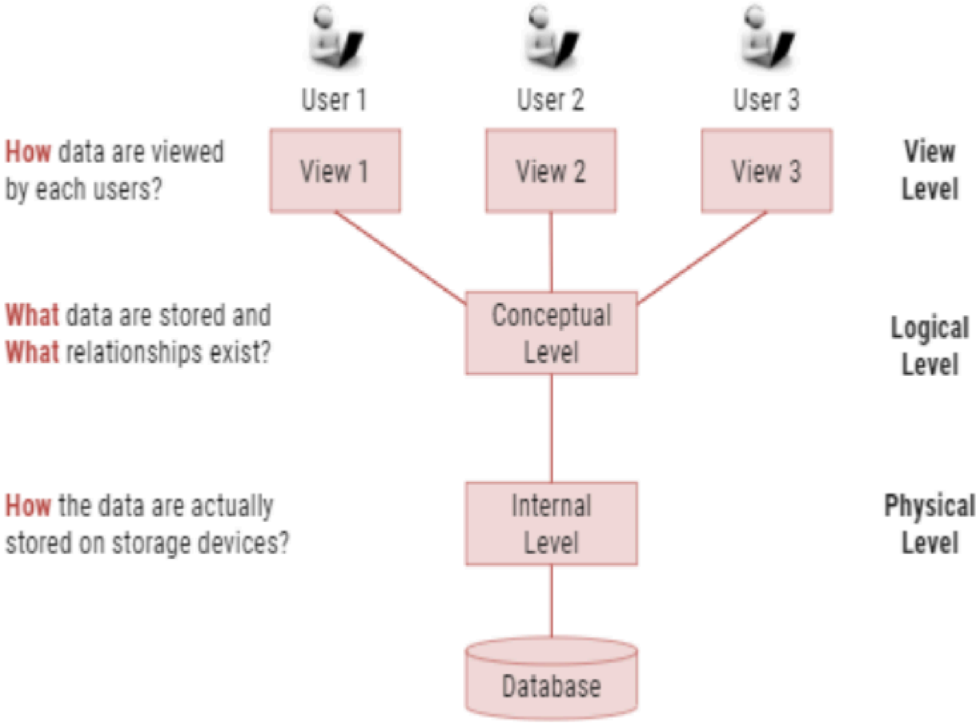
[1] **Faculty Of Engineering: Shubham Upadhyay**

[2]

Objectives of Three schema Architecture

➔ The main objective of three level architecture is to enable multiple users to access the same data with a personalised view while storing the underlying data only once. This separation is desirable for the following reasons:

- ➢ Different users need different views of the same data.
- ➢ All users should be able to access the same data according to their requirements.


User 1
User 2
User 3
How data are viewed by each users?
View 1
View 2
View 3
View Level
What data are stored and What relationships exist?
Conceptual Level
Logical Level
How the data are actually stored on storage devices?
Internal Level
Physical Level
Database

1. View/External level:
   - ➔ It describes only part of the entire database that an end user concerns or **how data is viewed** by each user.
   - ➔ Different users need different views of the database, so there can be many views in a view level abstraction of the database.
   - ➔ End users need to access only part of the database rather than the entire database.

2. Conceptual/Logical level:
    - ➔ The conceptual level describes what data is to be stored in the database and also describes what relationship exists among those data.
    - ➔ The conceptual schema describes the structure of the whole database.
    - ➔ Example, STUDENT database may contain STUDENT and COURSE tables which will be visible to users but users are unaware of their storage
3. Physical/Internal level:

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

Data Independence

- ➔ Data Independence is mainly defined as a property of DBMS that helps you to change the database schema at one level of a system [3]without requiring to change the schema at the next level.
- ➔ It helps to keep the data separated from all programs that make use of it.
- ➔ There are two levels of data independence:

[3] **Faculty Of Engineering: Shubham Upadhyay**

1.**Physical level data independence:** It 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 is not affected.

2. Logical level data independence: It refers to the characteristic of being able to modify the logical schema without affecting the external schema or application program.

->The user view of the data would not be affected by any changes to the conceptual view of the data.

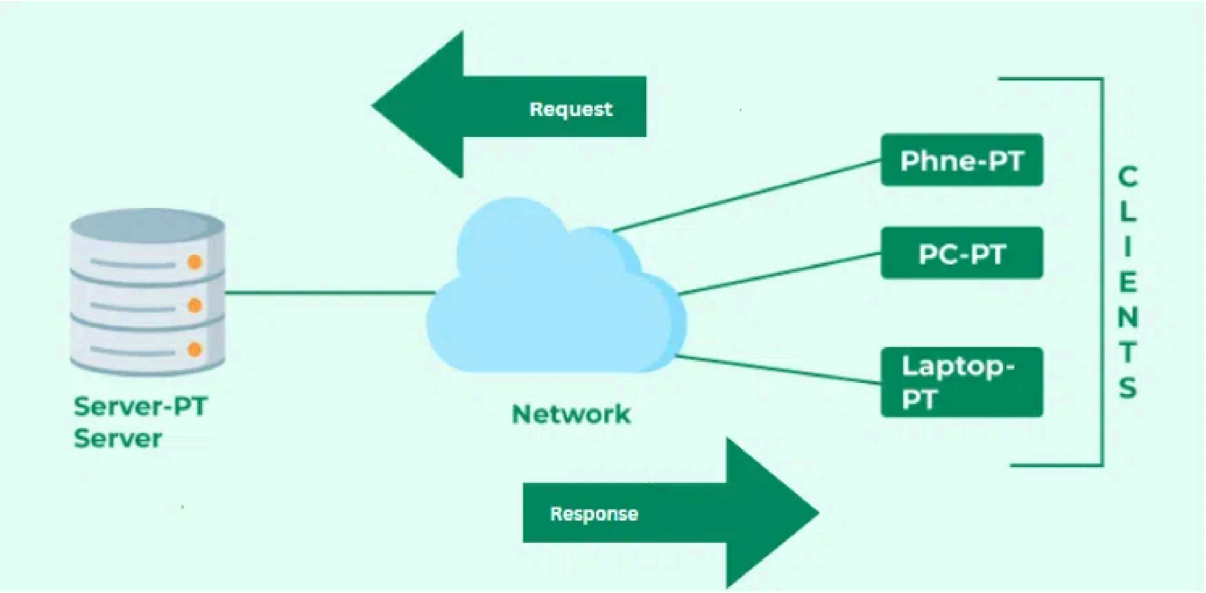
Data Abstraction

- ➔ Data abstraction is the process of hiding unwanted and irrelevant details from the end user.
- ➔ It helps to store information in such a way that the end user can access data which is necessary, the user will not be able to see what data is stored or how it is stored in a database.
- ➔ Data abstraction helps to keep data secure from unauthorised [4]access and it hides all the implementation details.

Client-Server Architecture

[4] **Faculty Of Engineering: Shubham Upadhyay**

- Client-server architecture is a computing model that divides task or workloads b/w service provider(server) and service requester(client), often over a network.
- **Client** is typically a user interface or application that requests services or resources from the server.
- **Server** is a powerful computer or software application that provides resources, services, or data to clients upon request.
- So, it is the **Client** requesting something and the **Server** serving it as long as it is in the database.

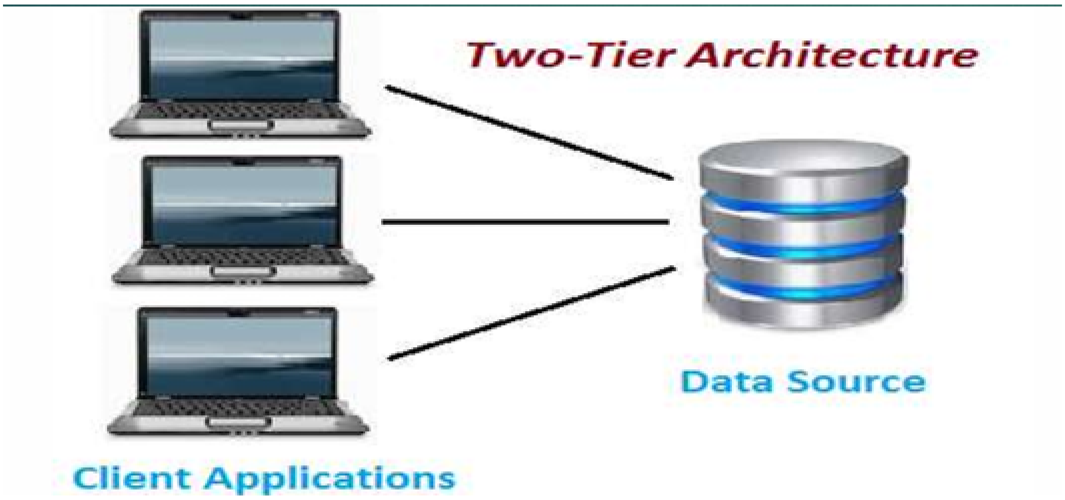


There are three types of Architecture:

1. 1-tier architecture: In 1-Tier Architecture the database is directly available to the user, the user can directly sit on the DBMS and use it, that is, the client, server, and Database are all present on the same machine.

2. 2-tier architecture:

- ➔ The 2-tier architecture is similar to a basic client-server model. The application at the client end directly communicates with the database on the server side.
- ➔ APIs like ODBC and JDBC are used for this interaction.
- ➔ The server side is responsible for providing query processing and transaction management functionalities.
- ➔ On the client side, the user interfaces and application programs are run.



Advantages of 2-Tier Architecture:

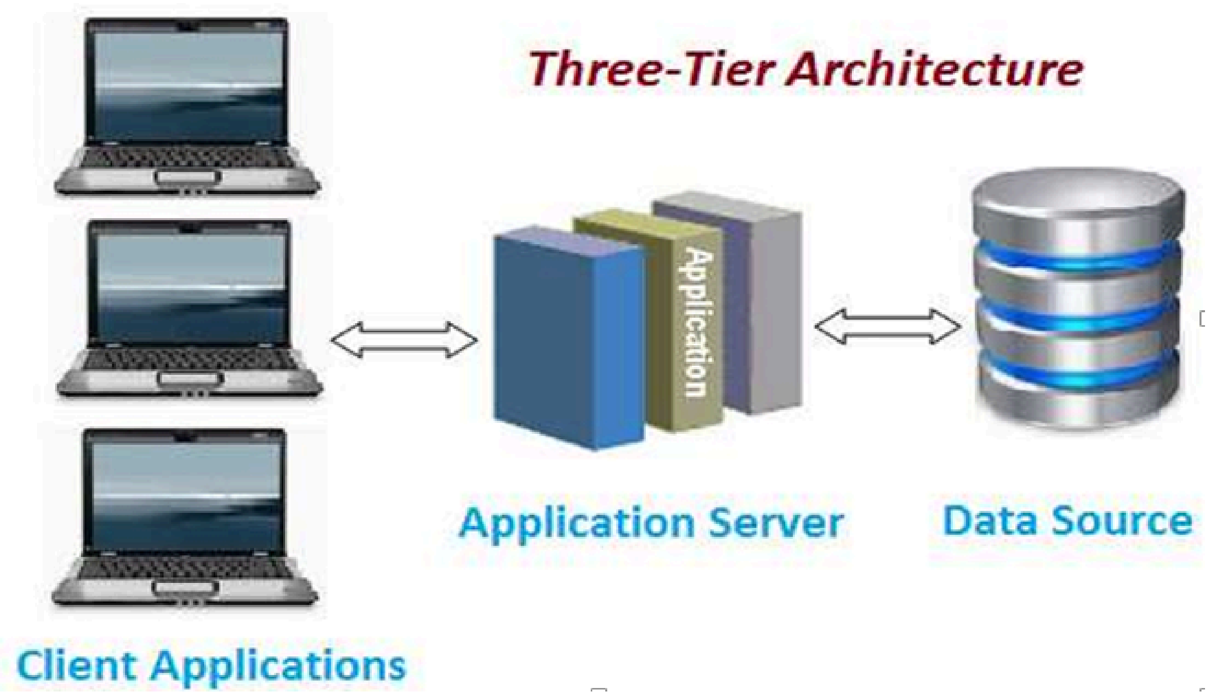
- **Easy to Access:** 2-Tier Architecture makes easy access to the databa[5]se, which makes fast retrieval.
- **Scalable:** We can scale the database easily, by adding clients or upgrading hardware.

[5] Faculty Of Engineering: Shubham Upadhyay

- **Low Cost:** 2-Tier Architecture is cheaper than 3-Tier Architecture.
- **Easy Deployment:** 2-Tier Architecture is easier to deploy than 3-Tier Architecture.
- **Simple:** 2-Tier Architecture is easily understandable as well as simple because of only two components.

Disadvantages: The application performance will be degraded upon increasing the users.

3. 3-Tier architecture:



➔ The 3-Tier architecture contains another layer between the client and server. In this architecture, the client can't directly [6]communicate with the server.

[6] Faculty Of Engineering: Shubham Upadhyay

- ➔ The application on the client end interacts with an application server which further communicates with the database system.
- ➔ This intermediate layer acts as a medium for the exchange of partially processed data between the server and the client.

Advantages of 3-Tier Architecture

- **Enhanced scalability:** Scalability is enhanced due to the distributed deployment of application servers.
- **Data Integrity:** 3-Tier Architecture maintains Data Integrity.
- **Security:** Client has no direct access to the database.

Disadvantages of 3-Tier Architecture

- More complex
- Difficult to interact

NOTES:

How the Browser Interacts With the Servers?

- ➔ There are a few steps to follow to interact with the servers of a client.
  - User enters the **URL**(Uniform Resource Locator) of the website or file. The Browser then requests the **DNS**(DOMAIN NAME SYSTEM) Server.
  - **DNS Server** lookup for the address of the **WEB Server**.
  - The **DNS Server** responds with the **IP address** of the **WEB Server**.
  - The Browser sends over an **HTTP/HTTPS** request to the **WEB Server's IP** (provided by the **DNS server**).
  - The Server sends over the necessary files for the website.
  - The Browser then renders the files and the website is displayed. This rendering is done with the help of **DOM** (Document Object

Model) interpreter, **CSS** interpreter, and **JS Engine** collectively known as the **JIT** or (Just in Time) Compilers.

Users & DBA

- ➔ A Database User is defined as a person who interacts with data daily, updating, reading, and modifying the given data.
- ➔ Four types of users:
    1. **Naive/End users:** Naive users are people who don't know much about computers and use the system by calling up one of the application programs that have already been written.
    2. **Application Programmer:** These are people who work on computers and write programs for application. It has a lot of tools to choose from when making a user interface.
    3. **Sophisticated User:** Sophisticated users can be engineers, scientists, business analysts, who are familiar with the database. They can develop their own database applications according to their requirements.
        - They don't write the program code but they interact with the database by writing SQL queries directly through the query processor.
    4. **Specialized Users:** Specialized users are advanced users who write specialized database applications that do not fit into the traditional data-processing framework.

Database Administrator

- ➔ A database administrator, or DBA, is responsible for maintaining, securing, and operating databases and also ensures that data is correctly stored and retrieved.
- ➔ DBAs often work with developers to design and implement new features and troubleshoot any issues.

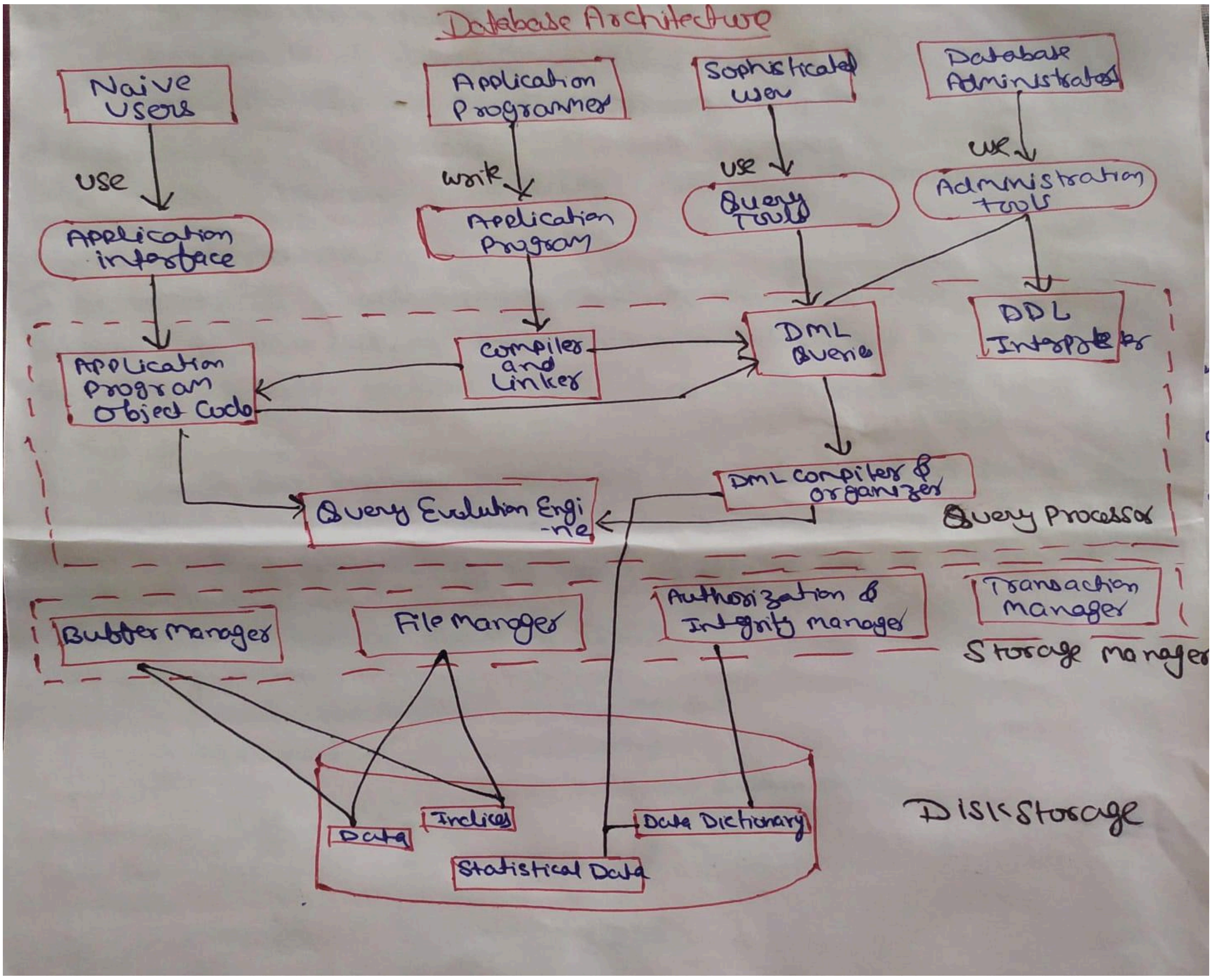
Role/Types of DBA:

- ➔ System administrators are responsible for the overall management and upkeep of a computer system, including installing and configuring software, applying security patches, and monitoring system performance.
- ➔ Database architects design databases to meet the specific needs of an organization.
- ➔ Database analysts collect and analyze data to help improve database performance. They may also be responsible for developing reports and providing recommendations to database administrators.

Database Architecture:

There are three components:

1. Query Processor
2. Storage Manager
3. Disk Storage



Query Processor:

- The query processing is handled by the query processor, as the name implies.
- It executes the user's query, to put it simply. In this way, the query processor aids the database system in making data access simple and easy.
- The Query Processor transforms (or interprets) the user's application program-provided requests into instructions that a computer can understand.

Storage Manager:

- Storage Manager is a program that provides an interface between the data stored in the database and the queries received.
- It maintains the consistency and integrity of the database by applying the constraints and executing the DCL statements.
- It contains the following components –
    1. **Authorization Manager:** Authorization manager verifies the user that he is valid and authenticated for the specific query or request.
    2. **Integrity Manager:** Whenever there is any change in the database, the Integrity manager will manage the integrity constraints
    3. **File Manager:** All the files and data structure of the database are managed by this component.
    4. **Transaction Manager:** It is responsible for making the database consistent before and after the transactions.
    5. **Buffer Manager:** The transfer of data between primary and main memory and managing the cache memory is done by the buffer manager.

Disk Storage:

- A DBMS can use various kinds of Data Structures as a part of physical system implementation in the form of disk storage.

1. **Data Dictionary:** It contains the metadata (data of data), which means each object of the database has some information about its structure. So, it creates a repository which contains the details about the structure of the database object.
2. **Data:** This component stores the data in the files.
3. **Indices:** These indices are used to access and retrieve the data in a very fast and efficient way.

[7]

[7] **Faculty Of Engineering - Shubham Upadhyay**