DATABASE MANAGEMENT SYSTEMS

UNIT- I

Introduction: Database system, Characteristics (Database Vs File System), Database Users (Actors on Scene, Workers behind the scene), Advantages of Database systems, Database applications. Brief introduction of different Data Models; Concepts of Schema, Instance, and data independence; Three tier schema architecture for data independence; Database system structure, environment, Centralized and Client Server architecture for the database.

Introduction

Datum

Data is usually organized into structures such as tables. So, an individual value in a collection of data (or) a single piece of information.

Data: Data can be facts related to any object in consideration.

Example: Name, age, height, weight etc.

These are related data to Person

Definition 2: data are plain facts or A collection of individual facts are statistics. The data can be come in the form of text, figures, numbers, graphs, or symbols

Definition 3: data is raw form of knowledge & its own deos not carry any significance or purpose.

Definition 4: Data is distinct piece of information that is gathered and translated for some purpose.

Types of data

There are two types of data

1. Quantitative data
2. Qualitative data

1. *Quantitative Data:* The data can be collected in numerical forms.

   Ex: Height, weight, age, cost of an object.

   Quantitative data refers to numerical data that can be measured or counted. This type of data is often used in scientific research and is typically collected through methods such as surveys, experiments, and statistical analysis.

There are two main types of quantitative data:

- Discrete
- Continuous.

Discrete data: Discrete data refers to numerical values that can only take on specific, distinct values. This type of data is typically represented as whole numbers and cannot be broken down into smaller units.
Examples of discrete data include the number of students in a class, the number of cars in a parking lot, and the number of children in a family.

Continuous data: Continuous data refers to numerical values that can take on any value within a certain range or interval. This type of data is typically represented as decimal or fractional values and can be broken down into smaller units.

Examples of continuous data include measurements of height, weight, temperature, and time.

2. *Qualitative data*

   The data can be represented or collected in the form of descriptive but not numerical.

   Example: name, gender, color of a person

Methods for collecting qualitative data

- Interviews
- Focous Groups
- Case Studies
- Cultural Research
- Field work
- Surveys (Open-ended Questions)

Information: The result of analyzing and interpreting the piece of data.

Definition 2: information is defined as knowledge gained through study, communication, research, or Instruction.

- Information is classified and organized data that has some meaningful value for user
- Information is verified or processed data about organizations & events, procedures.

Differences between Data and information

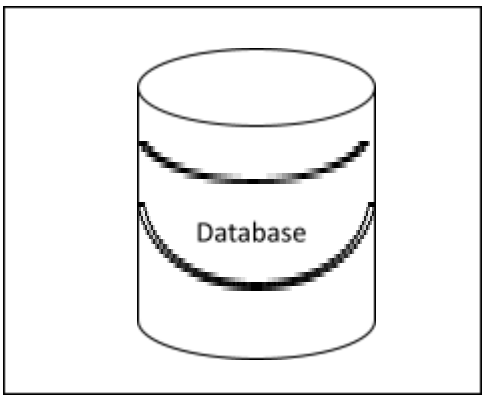
S. No	Data	Information
1.	Data is a collection of facts	Information put those facts into context
2.	Data is raw & unorganized	Information is organized
3.	Data is unrelated & no accurate	Information is related & accurate
4.	Data is meaningless	Information is meaningful
5.	Data does not depend on information	Information depends on data
6.	Data is not sufficient to make a decision	To make a decision based on information
7.	Data typically comes in the form of graphs, numbers, figures, or statics	Information typically presented through words, thoughts, and ideas

Database

- A database is a collection of data typically describing the activities of one or more organizations

  Example: University database contains the information, entities such as students, faculty, courses and classrooms etc.

- Database can be represented as cylinder in diagrammatically.



- *Definition 2:* A database is a organized collection of structural information or data typically stored economically in a computer system.
- An organized collection of logically related data without having any redundancy(duplication) stored in one place and access by multiple users
- A collection of related data is known as database
- A database is usually controlled by a database management system (DBMS)
- The short form of database is DB, de means decibel
- The database contains the model to store the data into the database i.e. tables

Process: The manipulation of data is called as process

Manipulation: The process of changing the data from one form to another form.

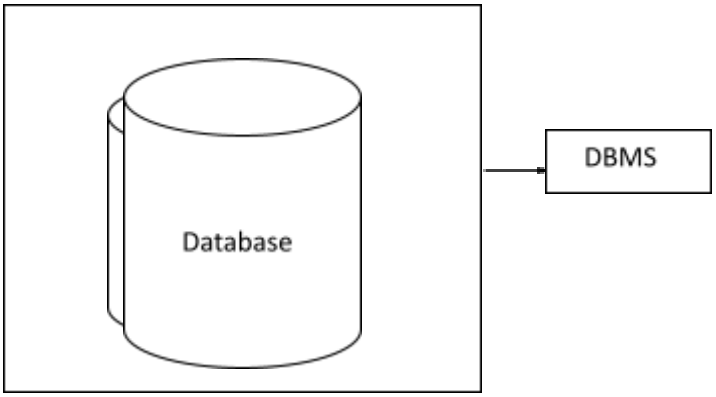
An organization must necessarily maintain A lot of data about its operations this data must should be accurate & reliable for effective decision making such related data is collectively known as database

Basic operations of database:

The basic operations that can be performed on a database are as follows

1. Creating new files
2. Inserting new records
3. Retrieving data
4. Updating data
5. Deleting data

Database Management Systems (DBMS)



- **Definition:** A collection of data to managing by a set of programs with effective & efficient manner.

- DBMS is a software to manage the database

  Example: MySQL, Oracle, MongoDB

Characteristics (database vs 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.
Example: *NTFS (New Technology File System), EXT (Extended File System).*

In earlier days, the databases were created directly on top of file systems

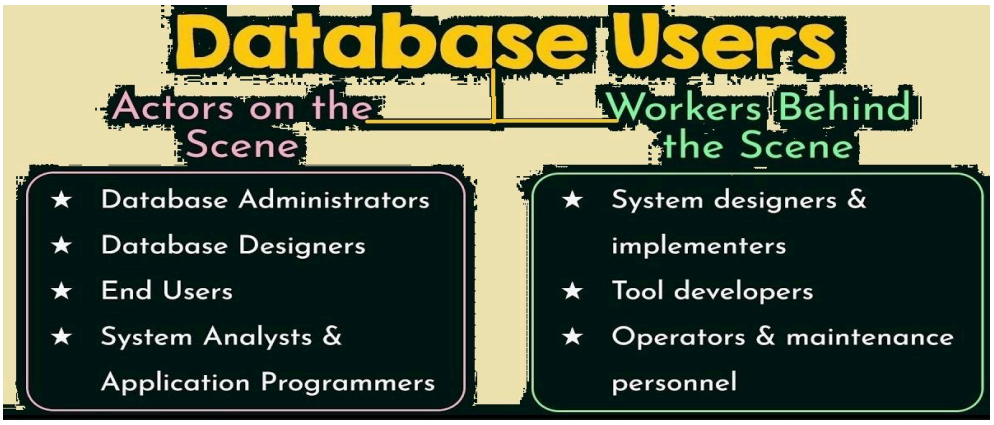
Basics	**File System**	**DBMS**
Structure	The file system is a way of arranging the files in a storage medium within a computer.	DBMS is software for managing the database.
Data Redundancy	Redundant data can be present in a file system.	In DBMS there is no redundant data.
Backup and Recovery	It does not provide Inbuilt mechanism for backup and recovery of data if it is lost.	It provides in house tools for backup and recovery of data even if it is lost.
Query processing	There is no efficient query processing in the file system.	Efficient query processing is there in DBMS.
Consistency	There is less data consistency in the file system.	There is more data consistency because of the process of normalization.

Basics	File System	DBMS
Complexity	It is less complex as compared to DBMS.	It has more complexity in handling as compared to the file system.
Security Constraints	File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file systems.
Cost	It is less expensive than DBMS.	It has a comparatively higher cost than a file system.
Data Independence	There is no data independence.	In DBMS data independence exists, mainly of two types: 1) Logical Data Independence. 2)Physical Data Independence.
User Access	Only one user can access data at a time.	Multiple users can access data at a time.
Meaning	The users are not required to write procedures.	The user must write procedures for managing databases
Sharing	Data is distributed in many files. So, it is not easy to share data.	Due to centralized nature data sharing is easy
Data Abstraction	It gives details of storage and representation of data	It hides the internal details of Database
Integrity Constraints	Integrity Constraints are difficult to implement	Integrity constraints are easy to implement

Basics	File System	DBMS
Attributes	To access data in a file, user requires attributes such as file name, file location.	No such attributes are required.
Example	Cobol, C++	Oracle, SQL Server

Database Users

A variety of people are associated with creation and use the databases Obviously there are "database implementers" builds DBMS software



The database implementors are divided into two groups

1. *Actors on the scene*
2. *Workers behind the scene*

- *Actors on the scene*
  - End-users to store the data & use the data in DBMS
  - Database implementors works for vendors such as IBM, Oracle.
  - End-users comes from a diverse and increasing number of fields
  - In single user DBMS, a person himself defines and manipulates queries. But multiple user DBMS a single person cannot manage all tasks.

T*here are four types of database users*

1. **Database administrator**

2. **Database designers**
3. **End-users**
4. **System analyst/application programmer/software developer**

Database administrator

- The DBA is the group of people that includes everything required to manage and solve every complex. The DBA can easily use the database to find the information they need and to plan the goal of the database. To meet future needs, they are ready for future scope and provide solutions for end users. Therefore, they are known for high-level management.

  For example −

  - *To handle the data loss.*
  - *To secure the privacy of data.*
  - *Monitor the recovery and backup of the database.*

End-users

- These users access the database from the front end with the help of a pre-developed application. They have little knowledge about the design and working of databases.

  There are two types of end users −

  - **Naive Users** − These naive users are those users who don't have any database knowledge. They depend on pre-developed applications like Bank Management Systems, Library Management Systems, Hospital Management Systems, and Railway Ticket Booking Systems (IRCTC) and get the desired result.
  - **Sophisticated Users** − These users interact with the system without writing a program and have separate databases for personal use. In the database, the user passes each query to the query processor.

    There are two ways to interact with a system −

    - They use the structure query language to run the query on the database.
    - They use the tool of data analysis software. For example, data engineers and data scientists are familiar with databases.
  - **Specialized users:** To write the application programs for queries which cannot answered directly.

- **Application Programmers**: These programmers write the code for an application program that uses the database. The application programmer can make the application according to user requirements and control software that runs an entire computer system. The application program is written in any programming language like C#, .net, JAVA, etc., and focuses on business, engineering, and science program.

  Application Programmers are divided into four different types −

  - Web Developers
  - Computer Hardware Programmers
  - Database Developers
  - Software Developers

  Examples of Application programmers develop software like −

  - Content access software
  - Educational Software
  - Information Worker Software
  - Media Development Software
  - Product Engineering Software
  - Enterprise Software
- **System Analyst** − A System Analyst has also known as a business technology analyst. These professionals are responsible for the design, structure, and properties of databases. The application programmer uses the specifications provided by the system analyst to construct the software that is used by end users. The analyst will gather information from the shareholders as well as end users to understand their requirements and translate it into functional specifications for the new system.

  Examples of System Analysts −

  - They serve as team leaders.
  - They are responsible for managing projects.
  - They are the supervisor who manages the lower-level information Staff.

Advantages of Database system

Database Management System (DBMS) is a collection of interrelated data and a set of software tools/programs that access, process, and manipulate data. It allows access, retrieval, and use of that data by considering appropriate security measures. The Database Management system (DBMS) is useful for better data integration and security.

- **Better Data Transferring:** Database management creates a place where users have an advantage of more and better-managed data. Thus, making it possible for end-users to have a quick look and to respond fast to any changes made in their environment.
- **Better Data Security:** The more accessible and usable the database, the more it is prone to security issues. As the number of users increases, the data transferring or data sharing rate also increases thus increasing the risk of data security.

  A Database Management System (DBMS) provides a better platform for data privacy and security policies thus, helping companies to improve Data Security.
- **Minimized Data Inconsistency:** Data inconsistency occurs between files when different versions of the same data appear in different places. For Example, data inconsistency occurs when a student's name is saved as "John Wayne" on the main computer of the school but on the teacher registered system same student name is "William J. Wayne", or when the price of a product is $86.95 in the local system of the company and its national sales office system shows the same product price as $84.95. So, if a database is properly designed then Data inconsistency can be greatly reduced hence minimizing data inconsistency.
- **Faster data Access:** The Database management system (DBMS) helps to produce quick answers to database queries thus making data access faster and more accurate.
- **Better decision making:** Better Data quality improves accuracy, validity, and the **a** time it takes to read data. DBMS does not guarantee data quality, it provides a framework to make it easy to improve data quality. DBMS provides powerful data analysis and reporting tools that allow users to make informed decisions based on data insights. This helps organizations to improve their decision-making processes and achieve better business outcomes.
- **Increased end-user productivity:** The data which is available with the help of a combination of tools that transform data into useful information, helps end-users to make quick, informative, and better decisions that can make a difference between success and failure in the global economy.
- **Simple:** Database management system (DBMS) gives a simple and clear logical view of data. Many operations like insertion, deletion, or creation of files or data are easy to implement.
- **Data abstraction:** The major purpose of a database system is to provide users with an abstract view of the data. Since many complex algorithms are used by the developers to

increase the efficiency of databases that are being hidden by the users through various data abstraction levels to allow users to easily interact with the system.

- **Reduction in data Redundancy:** When working with a structured database, DBMS provides the feature to prevent the input of duplicate items in the database. for e.g. – If there are two same students in different rows, then one of the duplicate data will be deleted.
- **Application development:** A DBMS provides a foundation for developing applications that require access to large amounts of data, reducing development time and costs.
- **Data sharing:** A DBMS provides a platform for sharing data across multiple applications and users, which can increase productivity and collaboration.
- **Data organization:** A DBMS provides a systematic approach to organizing data in a structured way, which makes it easier to retrieve and manage data efficiently.
- **The atomicity of data can be maintained:** That means, if some operation is performed on one table of the database, then the change must be reflected for the entire database.

  The DBMS allows concurrent access to multiple users by using the synchronization technique.
- **Data consistency and accuracy:** DBMS ensures that data is consistent and accurate by enforcing data integrity constraints and preventing data duplication. This helps to eliminate data discrepancies and errors that can occur when data is stored and managed manually.
- **Improved data security:** DBMS provides a high level of data security by offering user authentication and authorization, data encryption, and access control mechanisms. This helps to protect sensitive data from unauthorized access, modification, or theft.
- **Efficient data access and retrieval:** DBMS allows for efficient data access and retrieval by providing indexing and query optimization techniques that speed up data retrieval. This reduces the time required to process large volumes of data and increases the overall performance of the system.
- **Scalability and flexibility:** DBMS is highly scalable and can easily accommodate changes in data volumes and user requirements. DBMS can easily handle large volumes of data, and can scale up or down depending on the needs of the organization. It provides flexibility in data storage, retrieval, and manipulation, allowing users to easily modify the structure and content of the database as needed.

- **Improved productivity:** DBMS reduces the time and effort required to manage data, which increases productivity and efficiency. It also provides a user-friendly interface for data entry and retrieval, which reduces the learning curve for new users.

Advantages of Database Management System Over Traditional File System

- **Data Integrity and Security**: DBMS provides a centralized approach to data management that ensures data integrity and security. DBMS allows defining constraints and rules to ensure that data is consistent and accurate.
- **Reduced Data Redundancy**: DBMS eliminates data redundancy by storing data in a structured way. It allows sharing data across different applications and users, reducing the need for duplicating data.
- **Improved Data Consistency**: DBMS ensures data consistency by enforcing data validation rules and constraints. This ensures that data is accurate and consistent across different applications and users.
- **Improved Data Access and Availability**: DBMS provides efficient data access and retrieval mechanisms that enable quick and easy data access. It allows multiple users to access the data simultaneously, ensuring data availability.
- **Improved Data Sharing**: DBMS provides a platform for sharing data across different applications and users. It allows sharing data between different departments and systems within an organization, improving collaboration and decision-making.
- **Improved Data Integration**: DBMS allows integrating data from different sources, providing a comprehensive view of the data. It enables data integration from different systems and platforms, improving the quality of data analysis.
- **Improved Data Backup and Recovery**: DBMS provides backup and recovery mechanisms that ensure data is not lost in case of a system failure. It allows restoring data to a specific point in time, ensuring data consistency.
- **Data sharing:** DBMSs prevent conflicts and data loss by enabling numerous people to view and edit the same data at once. This promotes teamwork and enhances data uniformity throughout the company.
- **Data independence:** By separating the logical and physical views of data, database management systems (DBMS) enable users to work with data without being aware of its exact location or structure. This offers adaptability and lowers the possibility of data damage as a result of modifications to the underlying hardware or software.

- **Data integrity:** To avoid data mistakes and inconsistencies, database management systems (DBMSs) apply data integrity requirements including referential integrity, entity integrity, and domain integrity. This guarantees the consistency, accuracy, and completeness of the data.
- **Data security:** To prevent illegal access, alteration, or theft, database management systems (DBMS) include a number of security features, including encryption, authentication, and authorization. Sensitive data is safeguarded against both internal and external attacks thanks to this.
- **Data backup and recovery:** Database management systems (DBMS) offer backup and recovery features that let businesses swiftly and effectively restore lost or damaged data. This guarantees business continuity and lowers the chance of data loss.
- **Decreased data redundancy:** By keeping data centrally and offering methods for sharing and reusing it, database management systems (DBMS) remove data redundancy. As a result, less data storage is needed, and data consistency is increased.

Database Applications

Databases are widely used. Here are some representative applications:

1. *Banking*: For customer information, accounts, and loans, and banking transactions.

2. *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.

3. *Universities:* For student information, course registrations, and grades.

4. *Credit card transactions:* For purchases on credit cards and generation of monthly statements.

5. *Telecommunication:* For keeping records of calls made, generating monthly bills, maintaining balances on prepaid calling cards, and storing information about the communication networks.

6. *Finance:* For storing information about holdings, sales, and purchases of financial instruments such as stocks and bonds.

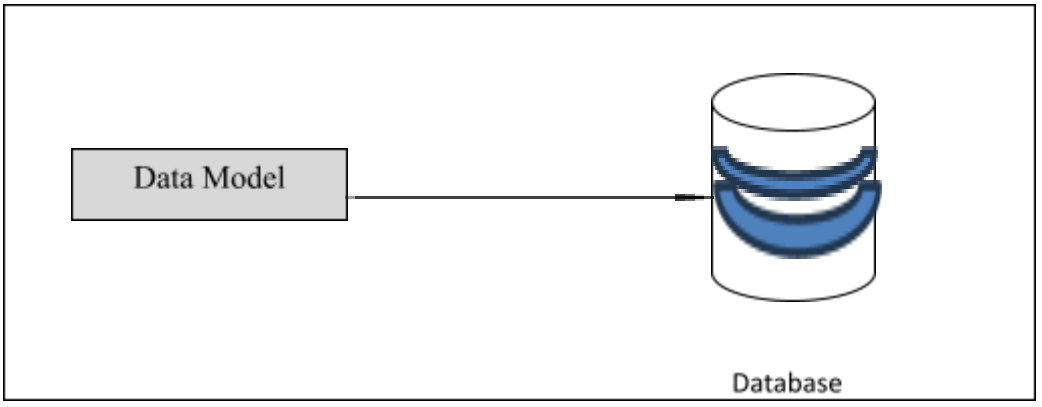
7. *Sales:* For customer, product, and purchase information.

8. *Manufacturing:* For management of supply chain and for tracking production of items in factories, inventories of items in warehouses / stores, and orders for items.

9. *Human resources*: For information about employees, salaries, payroll taxes and benefits, and for generation of paychecks

Brief Introduction to Data Models

- A database model defines the logical design and structure of a database. It defines how data will be stored, accessed, and updated in database management system.
- The database model sets the rules, relationships constraints etc. to define how the data is stored in the database
- It is like a blue print of database
- The main purpose of data model is to provide a level of abstraction



A data model is a collection of high-level data description constructs that hide many low-level storage details. A DBMS allows a user to define the data to be stored in terms of a data model. Most database management systems today are based on the relational data model.

There are basically three types of data models

1. Record-based logical model
2. Object-based logical model
3. Physical model

Record –based logical model: these models are used in describing data at the logical & view level of abstraction.

Record –based logical model is based on notation of data stored as a set of records. Each record consist of set of fields and each field is of fixed length.

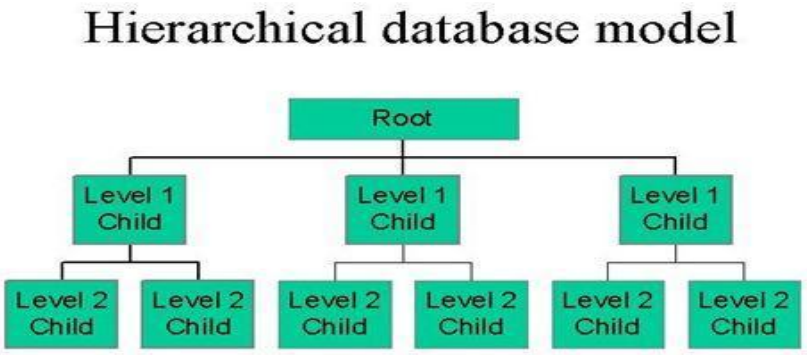
Fixed length records reduce the implementation part of physical level.

There are mainly three types of models

1. Hierarchical model
2. Network model
3. Relational model

Hierarchical Model:

- This model is oldest among database architecture
- In this model hierarchical relationships between data i.e parent – child relation ship
- In this model the data ca be arranged in the form of top-down structure.in this data has one-many relationship



- Data at the top node is root node data in the node at bottom most layers are called leaves
- Data in the intermediate level have one parent & one or several child nodes
- Example of hierarchical model is IBM's Information management system.

Advantages

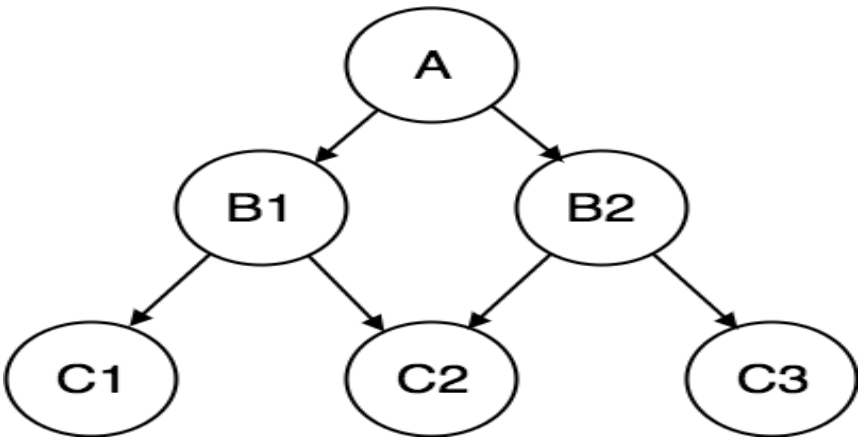
- Hierarchical model is simple to construct and operate on.¬

- Corresponds to a number of natural hierarchical organized domains – e.g., assemblies in¬ manufacturing, personal organization in companies.
- Language is simple; uses constructs like GET, GET UNIQUE, GET NEXT, GET NEXT WITHIN PARENT etc.,

Disadvantages

- Navigational and procedural nature of processing.
- Database is visualized as a linear arrangement of records.
- Little scope for "query optimization."
- One-to-many relationships

Network Model



- The network model is formerly defined in the 1971in the conference on the data system languages
- It is the extension of hierarchical model.
- In this model data is organized more like graph and allowed more than one parent node.
- This model uses many to many relationships
- The implementation of the network model is complex and it is very difficult to maintain it.
- The network model is difficult to modify also.

Advantages

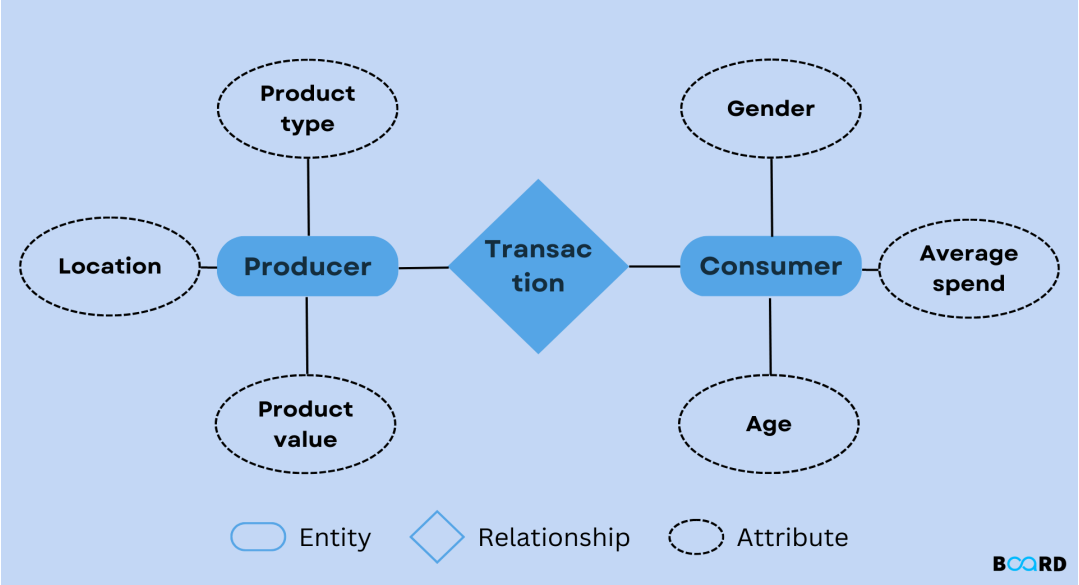
1. It supports complex relation ships
2. It allows more flexibility

Disadvantages

- Navigational and procedural nature of processing.
- Database contains a complex array of pointers that are expensive and difficult to update when inserting and deleting.

- Little scope for automated "query optimization."

Entity –Relationship Model



- In this model relationships are created by dividing object of interest into entities and their characteristics into attributes
- Different entities are related using relationships
- ER model is defined to represent the relationships
- This model is good to design a database which can then be turned into tables in a relational model

  Example: school database

Relational Model

- In this model data is organized in two dimensional tables and relationships is maintained by strong common field.
- This model was introduced by E.F Codd in 1970 & hence it has been most widely used database model.

- The basic structure of relational model is tables, hence tables also known as relations
- To design tables, normalize them to reduce the data redundancy & use SQL to access data from the tables.

Advantages

- **Simple model:** Relational Model is simple and easy to use in comparison to other languages.
- **Flexible:** Relational Model is more flexible than any other relational model present.
- **Secure:** Relational Model is more secure than any other relational model.
- **Data Accuracy:** Data is more accurate in the relational data model.
- **Data Integrity:** The integrity of the data is maintained in the relational model.
- **Operations can be Applied Easily:** It is better to perform operations in the relational model.

Disadvantages

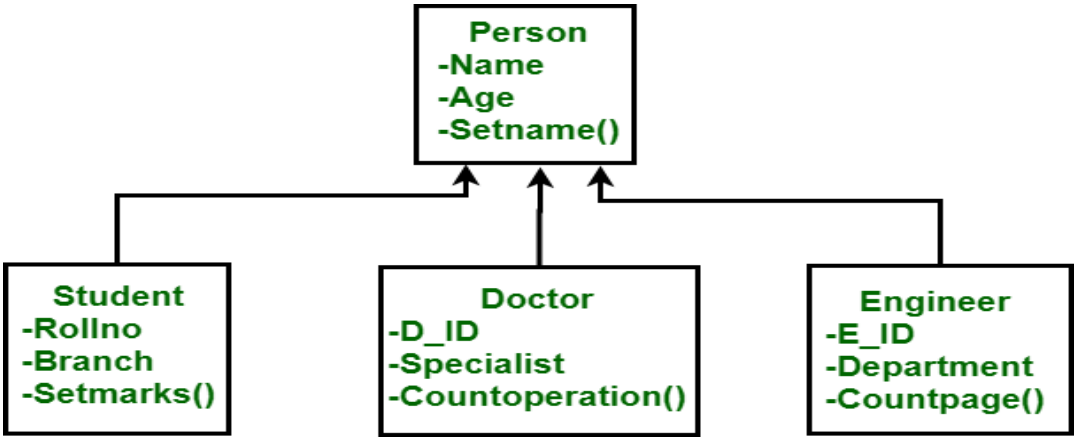
- Relational Database Model is not very good for large databases.
- Sometimes, it becomes difficult to find the relation between tables.
- Because of the complex structure, the response time for queries is high

Object-Oriented database Model

The **ODBMS** which is an abbreviation for **object-oriented database management system** is the data model in which data is stored in form of objects, which are instances of classes. These classes and objects together make an object-oriented data model.

Components of** Object-Oriented **Data Model

The OODBMS is based on three major components, namely: Object structure, Object classes, and Object identity. These are explained below.



- In this model data is stored in the form of objects
- The behavior of object-oriented database is like a object-oriented programming
- A very popular example of an object database management system or DBMS is MongoDB
- This database is not mature as compare to the relational database model

Advantages of Object-Oriented Data Model

- Codes can be reused due to inheritance.
- Easily understandable.
- Cost of maintenance can reduce due to reusability of attributes and functions because of inheritance.

Disadvantages of Object-Oriented Data Model

- It is not properly developed so not accepted by users easily

NoSQL Model

- No SQL database model supports unstructured style of database
- Data is stored as document
- The document looks like JSON string or key value-based object relations
- It provides flexible schemas
- It does provide features like indexing relationships between data.

Types of NoSQL Database: Types of NoSQL databases and the name of the database system that falls in that category are:

- **Graph Databases**: Examples – Amazon Neptune, Neo4j
- **Key value store:** Examples – Memcached, Redis, Coherence
- **Column:** Examples – Hbase, Big Table, Accumulate
- **Document-based:** Examples – MongoDB, CouchDB, Cloudant

Advantages of NoSQL

There are many advantages of working with NoSQL databases such as MongoDB and Cassandra. The main advantages are high scalability and high availability.

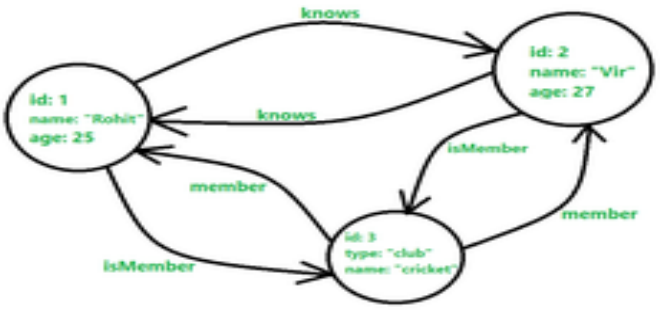
- High scalability
- Flexibility

- High availability
- Scalability
- Performance
- Cost-effectiveness.
- Agility

Disadvantages of NoSQL: *NoSQL has the following disadvantages.*

- Lack of standardization
- Lack of ACID compliance
- Narrow focus
- Open-source:
- Lack of support for complex queries
- Lack of maturity
- Management challenge
- *GUI is not available*: GUI mode tools to access the database are not flexibly available in the market.
- *Backup*: Backup is a great weak point for some NoSQL databases like MongoDB. MongoDB has no approach for the backup of data in a consistent manner.
- *Large document size*: Some database systems like MongoDB and CouchDB store data in JSON format. This means that documents are quite large (Bigdata, network bandwidth, speed), and having descriptive key names hurts since they increase the document size.

Graph Model



Advantages: Frequent schema changes, managing volume of data, real-time query response time, and more intelligent data activation requirements are done by graph model.

Disadvantages: Note that graph databases are not always the best solution for an application. We will need to assess the needs of application before deciding the architecture.

Limitations of Graph Databases

- Graph Databases may not be offering better choice over the NoSQL variations.
- If application needs to scale horizontally this may introduces poor performance.
- Not very efficient when it needs to update all nodes with a given parameter.

Schema

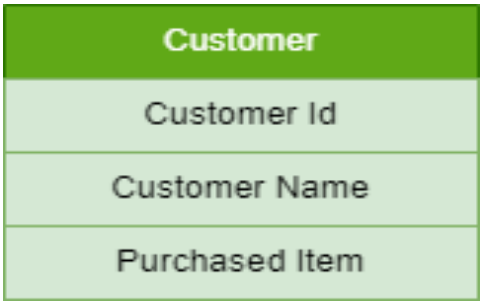
- Schema means framework or a diagrammatic representation or outline.

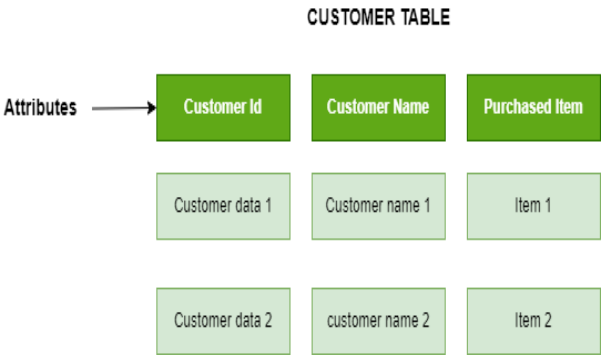
- A database schema is a **logical representation of data** that shows how the data in a database should be stored logically. It shows how the data is organized and the relationship between the tables.
  - A schema is essentially a blueprint of the database that defines its logical structures constraints and integrity constraints like table, primary key, etc.
  - A database schema is the skeleton structure that represents the logical view of the entire database. Logical view means the relationships between data and constraints.
  - Schema is the overall description of the database.
  - The schema does not represent the data type of the attributes.
  - Database schema contains table, field, views and relation between different keys like primary key, foreign key.
  - Database schema provides the organization of data and the relationship between the stored data.
  - Database schema defines a set of guidelines that control the database along with that it provides information about the way of accessing and modifying the data.

    Example for Schema

    Student

Sno	Sname	Scourse	Saddress	sphoneno





Types of Database Schemas

There are 3 types of database schema

Physical Database Schema

- A Physical schema defines, how the data or information is stored physically in the storage systems in the form of files & indices. This is the actual code or syntax needed to create the structure of a database, we can say that when we design a database at a physical level, it is called physical schema.

  .

Logical Database Schema

- The Logical schema describes how the data is stored in the form of tables & how the attributes of a table are connected.
- Using **ER modelling** the relationship between the components of the data is maintained.
- In logical schema different integrity constraints are defined in order to maintain the quality of insertion and update the data.

View Database Schema

- It is a view level design which is able to define the interaction between end-user and database.
- User is able to interact with the database with the help of the interface without knowing much about the stored mechanism of data in database.

Instance

The situation where a data or information is stored in the database at a particular moment of time is called an instance. An instance is also called a current state or database state. The

database schema that defines variables in tables which belong to a specific database, the records of these variables at a particular moment are called the instance of the database.

Many instances are constructed to correspond to a specific database schema. Every time we can insert, modify, or delete the value of a data item in a record. One state of data can change into another state.

Consider a table given below which has the student (Schema) −

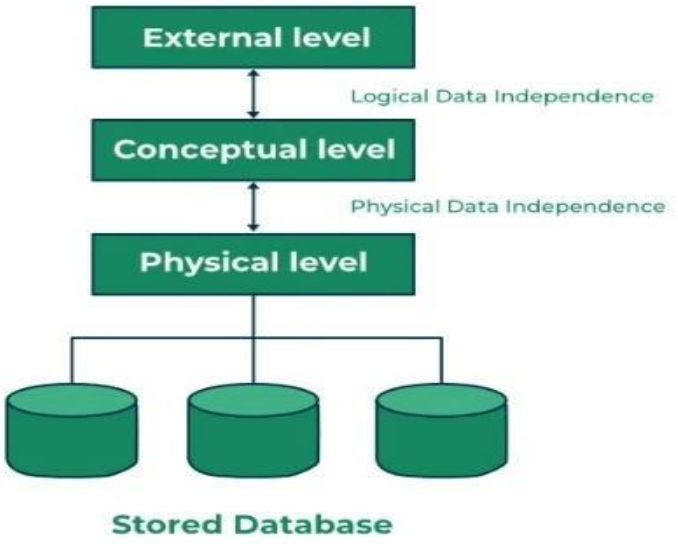
Std ID	Name	City
100	Lucky	Hyderabad
101	Pinky	Delhi
102	Bob	Hyderabad

- In the above table, rows are called instances.
- Finally, we can say that the content of database at a point of time is called instance or database state.
- There are three types of states present in database −
    - Empty state: Whenever a new database is defined.
    - Initial state: first time data is loaded in database.
    - Current state: the present operation is applied to database.

Differences between schema and instance

Schema	Instance
It is the overall description of the database.	It is the collection of information stored in a database at a particular moment.
The schema is same for the whole database.	Data in instances can be changed using addition, deletion, and updation.
Does not change Frequently.	Changes Frequently.
Defines the basic structure of the database i.e. how the data will be stored in the database.	It is the set of Information stored at a particular time.

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 without requiring to change the schema at the next level. it helps to keep the data separated from all program that makes use of it.

We have namely two levels of data independence arising from these levels of abstraction:

- Physical level data independence
- Logical level data independence

Physical Level Data Independence

It refers to the characteristic of being able to modify the physical schema without any alterations to the conceptual or logical schema, done for optimization purposes, e.g., the Conceptual structure of the database would not be affected by any change in storage size of the database system server. Changing from sequential to random access files is one such example. These alterations or modifications to the physical structure may include:

- Utilizing new storage devices.
- Modifying data structures used for storage.
- Altering indexes or using alternative file organization techniques etc.

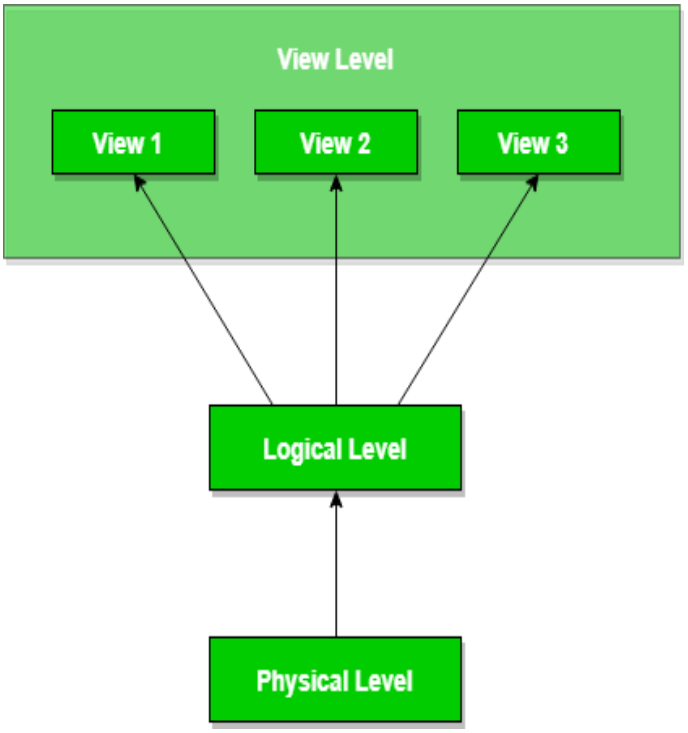
Logical Level Data Independence

It refers 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. These changes may include insertion or

deletion of attributes, altering table structures entities or relationships to the logical schema, etc.

Levels of Abstraction or III –tier schema architecture

Database systems comprise complex data structures. In order to make the system efficient in terms of retrieval of data, and reduce complexity in terms of usability of users, developers use abstraction i.e. hide irrelevant details from the users. This approach simplifies database design.



Level of Abstraction in a DBMS

There are mainly 3 levels of data abstraction:

- Physical or Internal Level
- Logical or Conceptual Level
- View or External Level

Physical or Internal Level

This is the lowest level of data abstraction. It tells us how the data is actually stored in memory. Access methods like sequential or random access and file organization methods like B+ trees and hashing are used for the same. Usability, size of memory, and the number of times the records are factors that we need to know while designing the database.

Suppose we need to store the details of an employee. Blocks of storage and the amount of memory used for these purposes are kept hidden from the user.

Logical or Conceptual Level

This level comprises the information that is actually stored in the database in the form of tables. It also stores the relationship among the data entities in relatively simple structures. At this level, the information available to the user at the view level is unknown.

We can store the various attributes of an employee and relationships, e.g. with the manager can also be stored.

View or External Level

This is the highest level of abstraction. Only a part of the actual database is viewed by the users. This level exists to ease the accessibility of the database by an individual user. Users view data in the form of rows and columns. Tables and relations are used to store data. Multiple views of the same database may exist. Users can just view the data and interact with the database, storage and implementation details are hidden from them.

Example: *In case of storing customer data,*

- **Physical level** – it will contain block of storages (bytes, GB, TB, etc)
- **Logical level –** it will contain the fields and the attributes of data.
- **View level** – it works with CLI or GUI access of database

The main purpose of data abstraction is to achieve data independence in order to save the time and cost required when the database is modified or altered.

Mapping between three levels

- Two mappings are required in a database system with three different views.
- A mapping between the external and conceptual views gives the correspondence among the records and the relationships of the external and conceptual views.
- The external views is the abstraction of the conceptual view which is turn is an abstraction of the internal view.
- There is a mapping from conceptual view to an internal one, the mapping specifies the method of deriving the conceptual record from the physical database.

Database system Structure or Architecture of DBMS

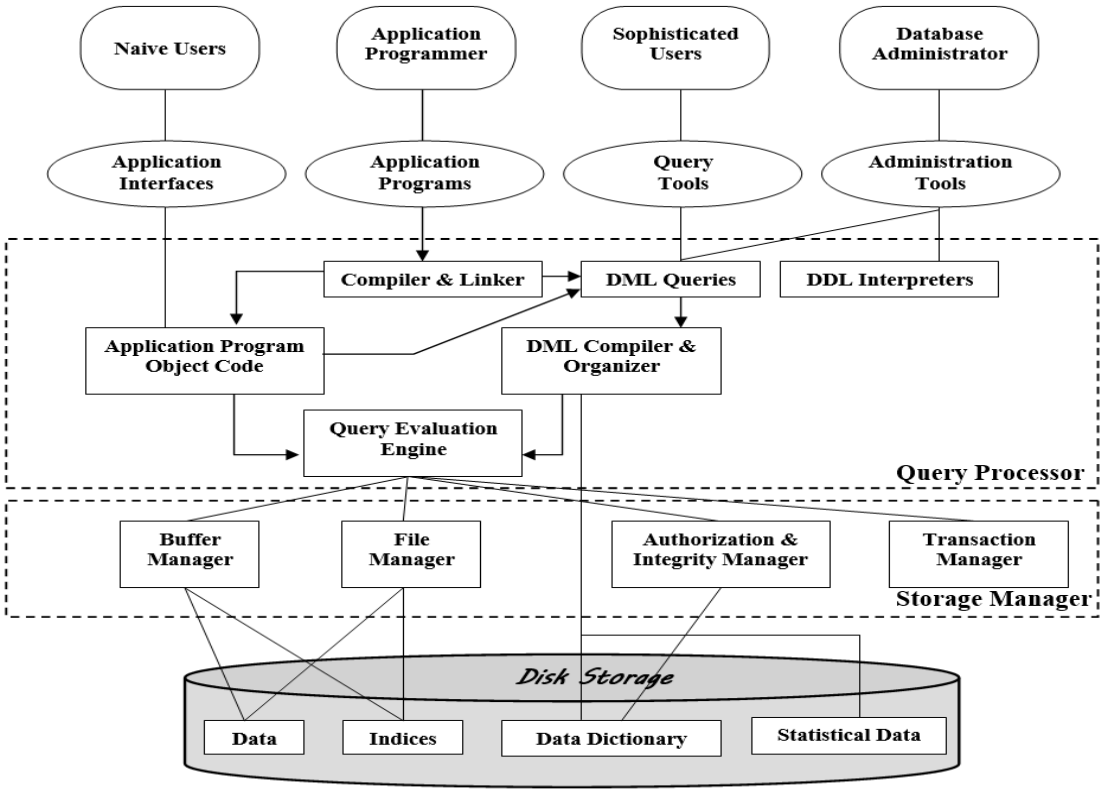


Figure: System Architecture

The functional components of a database system can be broadly divided into three modules

1. Query processor components
2. storage manager components
3. Disk Storage components

The query processor includes:

1. *DML Compiler*: It translates DML statements in a query language into low-level instructions that the query evaluation engine understands.

2. *Embedded DML Pre-compiler*: It converts DML statements embedded in an application program to normal procedure calls in the host language. The pre-compiler must interact with the DML compiler to generate the appropriate code.

3. *DDL Interpreter*: It interprets DDL Stateline its and records them in a set of tables containing

4. *Transaction Manager*: Ensures that the database remains in a consistent (correct) state despite system failures, and that concurrent transaction executions proceed without conflicting.

5. *File Manager*: Manages the allocation of space on disk storage and the data structures used to represent information stored on disk.

6. *Buffer Manager*: Is responsible for fetching data from disk storage into main memory and deciding what data to cache in memory. Also, some data structures are required as part of the physical system implementation

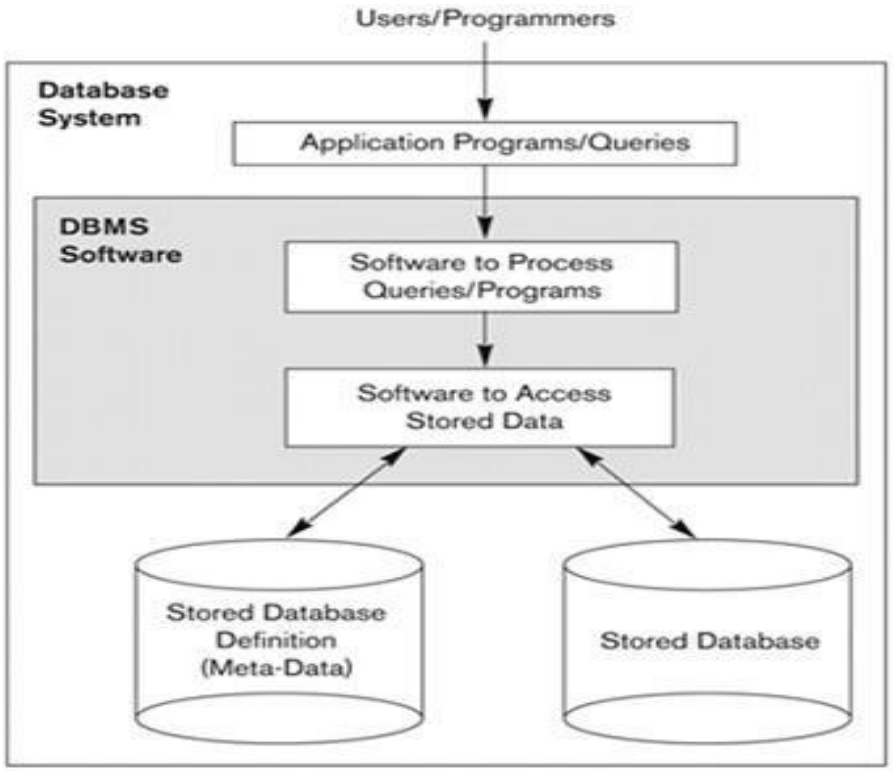
7. *Authorization & Integrity Manager*: Responsible for to provide & check the authorization of users and verify the constraints of relation before apply the data

1. *Data Files:* The data files store the database by itself.

2. *Data Dictionary*: It stores metadata about the structure of the database, as it is used heavily. 3. *Indices*: It provides fast access to data items that hold values.

4. *Statistical Data:* It stores statistical information about the data in the database. This information used by the query processor to select efficient ways to execute a query

Database Environment



A database environment is a collective system of components that comprise and regulates the group of data, management, and use of data, which consist of software, hardware, people, techniques of handling database, and the data also.

A database management system (DBMS) is a collection programs that enables users to create and maintain database

The DBMS is hence a general-purpose software system that facilitative the process of

1. Defining
2. Constructing
3. Manipulating
4. Sharing

Defining a database involves to specifying the data types, structures, and constraints for the data to be stored in the database.

*Constructing t*he database is the process of storing the data itself on some storage medium that is controlled by DBMS.

Manipulating a database includes such functions as querying the database to retrive specific data, updating the database to reflect changes in the mini world and generating reports from the data.

Sharing a database allows multiple users and programs to access the database concurrently.

DBMS can provide the two other important two functions

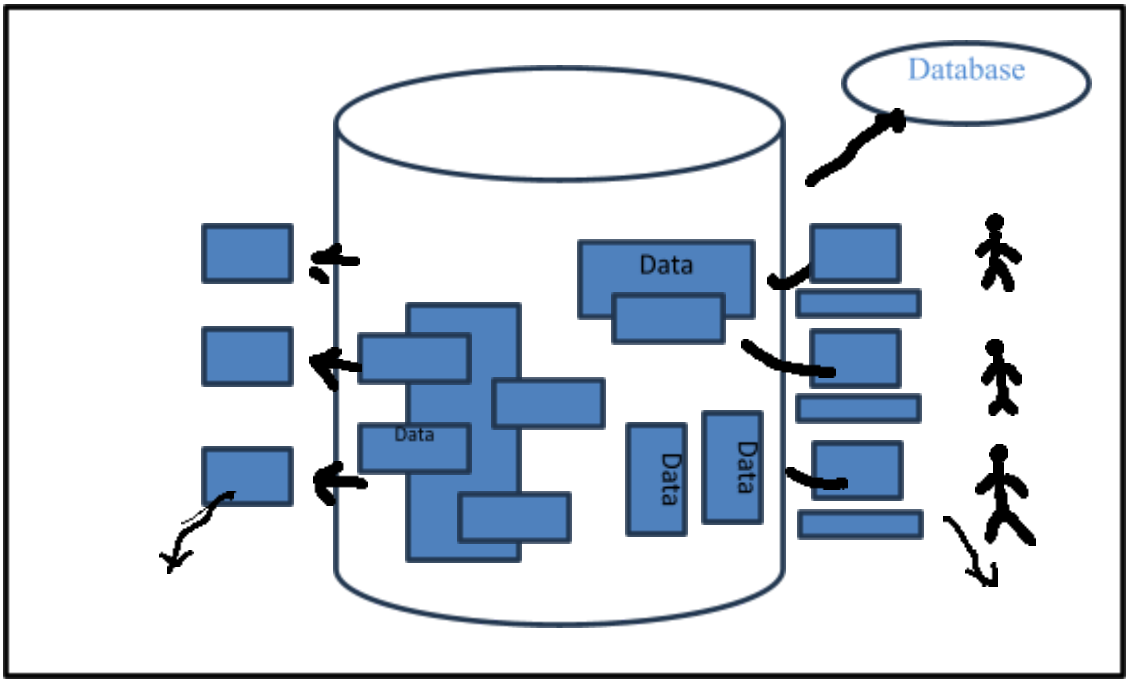
1. Protecting the database
2. Maintain the database

Protecting the database includes both system protection against hardware and software malfunctions or crashes and security protection against unauthorized or malicious access.

Maintenance of database: A typical large database may have a life cycle of many years so the DBMS must be able to maintain the database system by allowing the system to evolve as

requirements change over time. We can call the database and DBMS software together a database system.

Database Components



A database is an organized integrated and shared collection of related data designed to meet the needs of multiple users.

The database environment consists of four components

1. Data
2. Users
3. Hardware
4. Software

- ***Data:*** An application can create a data and enter the database this data can then be accessed and processed by another application.
- These are two main things that distinguish the database approach from other methods of data storage i.e a database is integrated & shared.
- Integrated data means that instead of applications having their own database management systems (DBMS)
- Data is shared means in database all data is stored together and shared by each application

- The database should support concurrent data sharing

Users:

- The second component in the database environment is users. The users of database can be broadly classified or categorized into three broad classes
- End-users interact with the database by operating application programs
- Database administrator Co-ordinate the activities of all users of the database and have ultimate control of the database.
- Application developers develop programs for applications that process the data stored in database
- The term end-users is used to describe the person who access the database in the course of their day-to-day work.
- The Database administrator (DBA) manages the database.

Hardware:

- The third component of database is hardware. A database can be run on conventional computers such as mainframes, PC's, and MINIS.
- A Database can also be run a dedicated server.

Software:

1. The final component of database is software. The software in data environment can be divided into two made categories.
    - DBMS
    - Application Softwires
        i. DBMS: this software controls the database.

    ii .Application softwares: these are written by application developers for end-users.

Application software utilizes the DBMS software for data handling such as retrieval ,creation, updating or deleting.

Centralized Database Architecture

A database management system (DBMS) is a software system that is designed to manage and organize data in a structured manner. In order to accomplish this, DBMS uses a specific architecture that dictates how data is stored, retrieved, and updated.

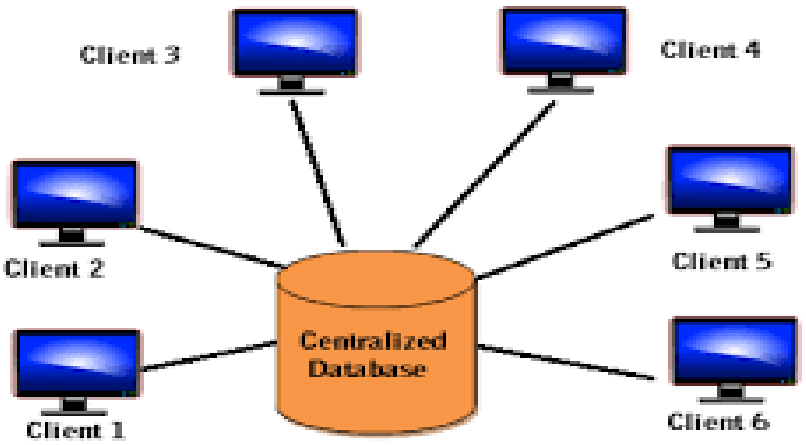
Two of the most commonly used architectures in DBMS

1. Centralized Database Architecture
2. Client-Server Architectures.

Centralized Database Architecture

A centralized architecture for DBMS is one in which all data is stored on a single server, and all clients connect to that server in order to access and manipulate the data.

This type of architecture is also known as a monolithic architecture.



A centralized database is basically a type of database that is stored, located as well as maintained at a single location only. This type of database is modified and managed from that location itself. This location is thus mainly any database system or a centralized computer system.

The centralized location is accessed via an internet connection (LAN, WAN, etc). This centralized database is mainly used by institutions or organizations.

Advantages: This database can be easily expanded as data is already spread across different physical locations.

The distributed database can easily be accessed from different networks.

- This database is more secure in comparison to a centralized database.

  Disadvantages:

- This database is very costly and is difficult to maintain because of its complexity.

Client-server Architecture of DBMS:

We first talk about client/server architecture in general, and then we look at how DBMSs use it. In order to handle computing settings with a high number of PCs, workstations, file servers, printers, database servers, etc., the client/server architecture was designed.

A network connects various pieces of software and hardware, including email and web server software. To define specialized servers with a particular functionality is the aim. For instance, it is feasible to link a number of PCs or compact workstations to a file server that manages the client machines' files as clients. By having connections to numerous printers, different devices can be designated as a printer server; all print requests from clients are then directed to this machine. The category of specialized servers also includes web servers and email servers. Many client machines can utilize the resources offered by specialized servers. The user is given the proper user interfaces for these servers as well as local processing power to run local applications on the client devices. This idea can be applied to various types of software, where specialist applications, like a CAD (computer-aided design) package, are kept on particular server computers and made available to a variety of clients. Some devices (such as workstations or PCs with discs that only have client software installed) would only be client sites.

The idea of client/server architecture presupposes an underpinning structure made up of several PCs and workstations as well as fewer mainframe computers connected via LANs as well as other types of computer networks. In this system, a client is often a user machine that

offers local processing and user interface capabilities. When a client needs access to extra features-like database access-that are not available on that system, it connects to a server that offers those features. A server is a computer system that includes both hardware and software that can offer client computer services like file access, printing, archiving, or database access. Generally speaking, some workstations install both client and server software, while others just install client software. Client and server software, however, typically run on separate workstations, which is more typical. On this underlying client/server framework, **Two-tier** and **Three-tier** fundamental DBMS architectures were developed.

Two-Tier Client Server Architecture:

Here, the term "two-tier" refers to our architecture's two layers-the Client layer and the Data layer. There are several client computers in the client layer that can contact the database server. The API on the client computer will use **JDBC** or some other method to link the computer to the database server. This is due to the possibility of various physical locations for clients and database servers.

Three-Tier Client-Server Architecture:

The Business Logic Layer is an additional layer that serves as a link between the Client layer and the Data layer in this instance. The layer where the application programs are processed is the business logic layer, unlike a Two-tier architecture, where queries are performed in the database server. Here, the application programs are processed in the application server itself.