

## Unit-2

### Database System - Concepts and Architecture

⊗ Data abstraction → It refers to hiding mechanism of details of data organization and storage and the highlighting of the essential features for an improved understanding of data. To provide data abstraction is one of the fundamental characteristic of database approach. There are three levels of abstraction.

- i) Physical level / lowest level → describes how data is actually stored in database.
- ii) Logical level / Second lowest level → describes type of data stored and relation between them.
- iii) View level / highest level → describes interaction between the users and system.

⊗ Data Model: Data model is a collection of concepts that can be used to describe structure of a database. It provides the necessary means to achieve the abstraction. It is a conceptual tool for describing data and relation among data. There are several data models which can be put into different categories:-

#### 1) Object-based logical models:

Object-based logical model describe data at the logical and view levels. It has flexible structuring capabilities. It is of two types:

(a) Entity-Relationship model: E-R model describes the design of database in terms of entities and relationship among them. An entity is a 'object' in real world with a set of attributes. Eg. attributes: customer\_id, customer\_name, customer\_address etc.

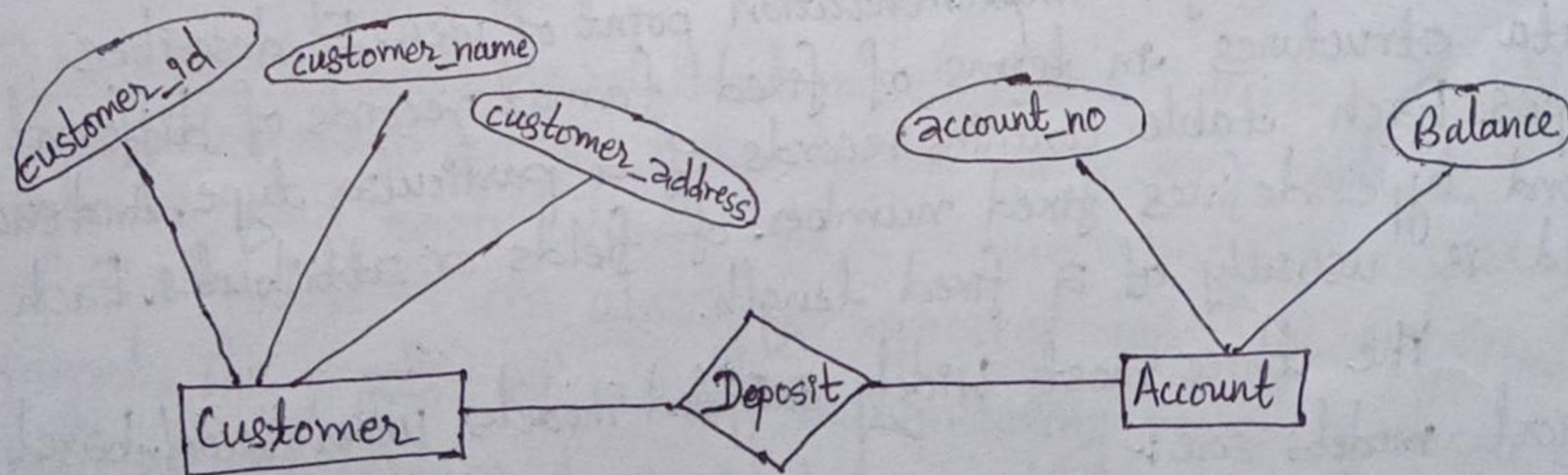


Fig. Sample of E-R model. (ER diagram)



An entity is a thing or object in real world which are described in database as a set of attributes. A relationship is an association among several entities. Logical structure of a database can be expressed graphically by ER diagram which include following components:

- Rectangles (represent entity sets).
- Ellipses (represent attributes)
- Diamonds (represent relationship sets among entity sets).
- Lines (link attributes to entity sets and entity sets to relationship sets).

### ⑥. Object-Oriented Model:

The object oriented model is based on a collection of objects like the E-R model. Object contains values and bodies of codes which are called methods. An object can contain another object to an arbitrarily deep level of nesting. Another feature of object-oriented model is inheritance i.e., new class can be derived from the existing class.

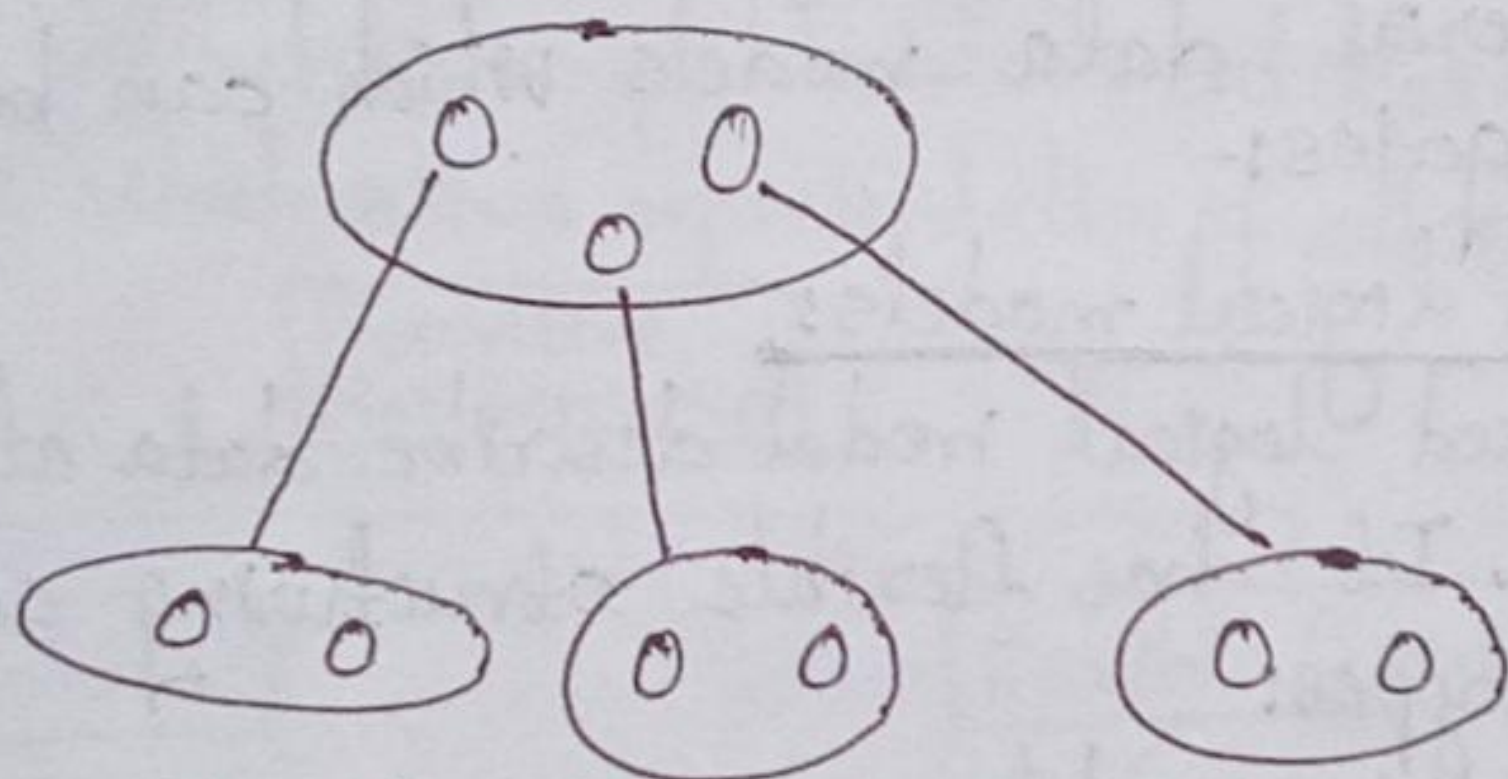


Fig. Object - Oriented Model

### 2) Record-based logical Model:

Record-based logical model also describes data at logical and view level but it describes logical structure of database in more detail for implementation point of view. It describes data structures in terms of fixed format records of different types. Each table contains records of a particular type. And each record type defines fixed number of fields or attributes. Each field is usually of a fixed length.

The three most widely accepted models under record-based logical models are:-



①. Hierarchical model: In hierarchical data model the data is organized into a tree-like structure. The structure allows repeating information using parent/child relationships; each parent can have many children, but each child only has one parent.

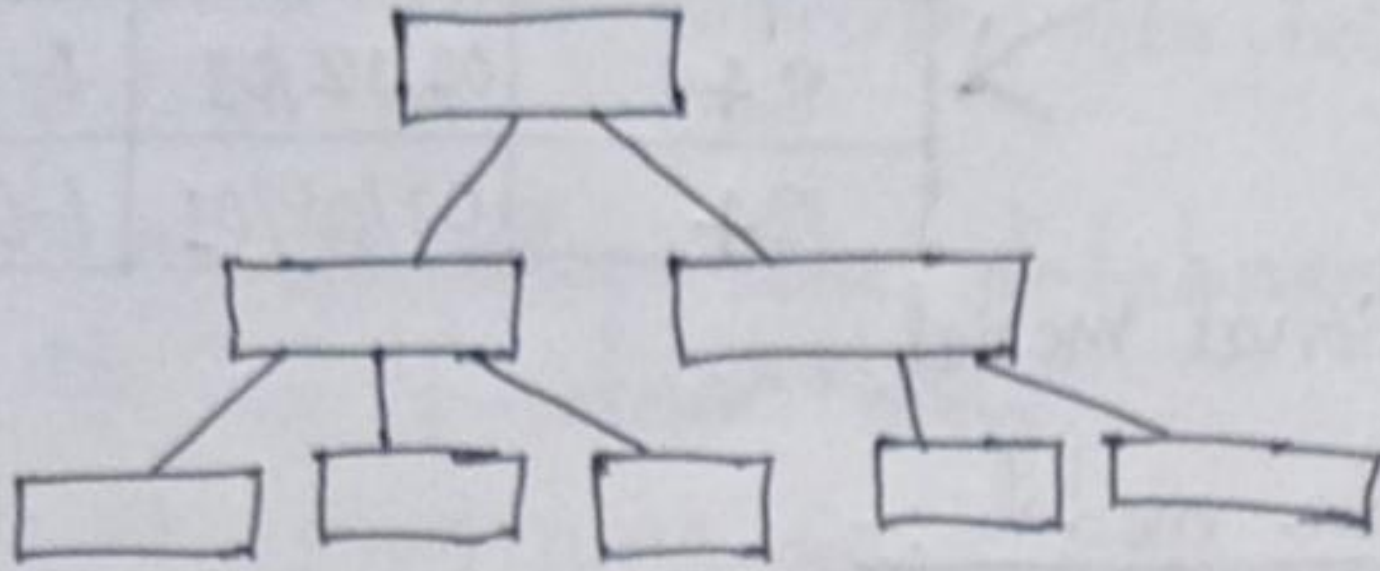


Fig. Structure of Hierarchical model

②. Network model: The data in network model are represented by collection of records and relationships among data are represented by links, which ~~are~~ can be viewed as pointers. The records in the database are organised as collection of arbitrary groups. The main advantage of this model is its representation of relationships between entities is implemented using pointers which allows representation of arbitrary relationship.

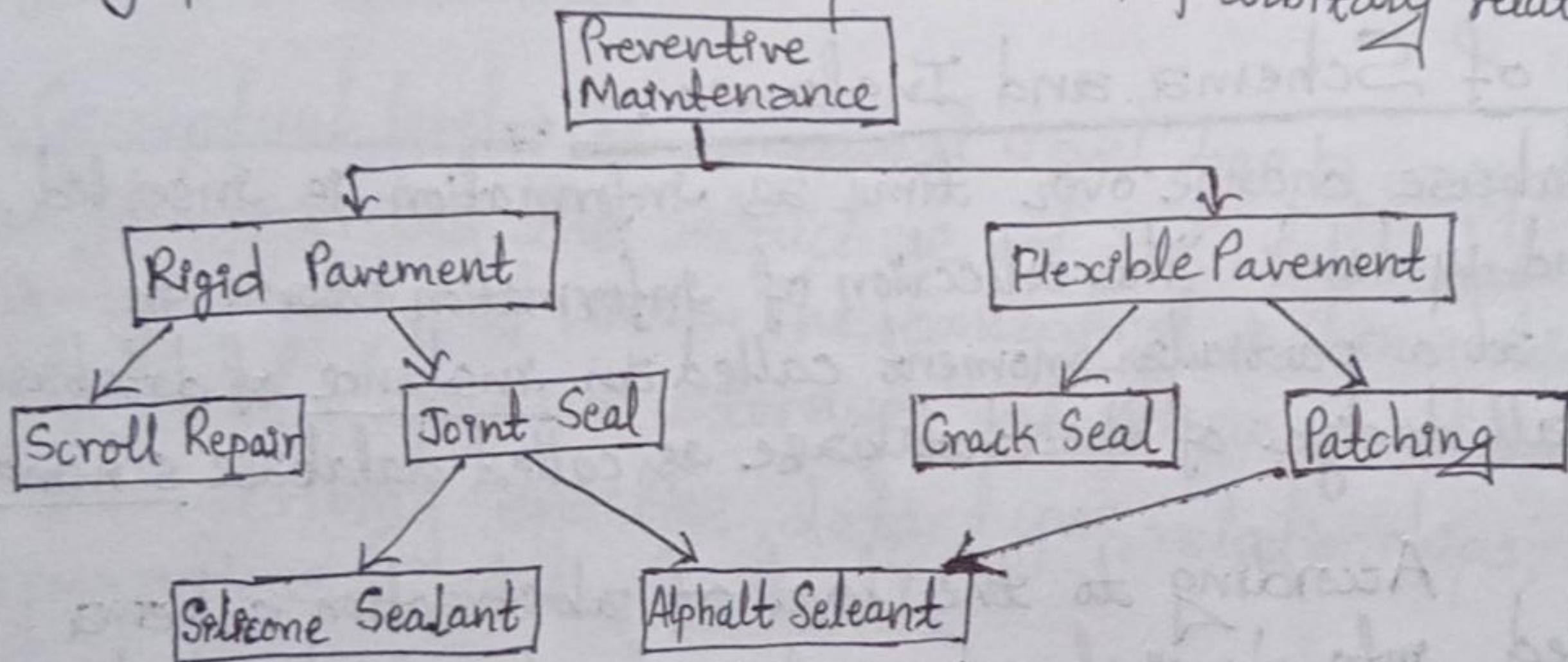


Fig. Network model

③. Relational model: The relational model uses a collection of tables and relationship among those tables. Tables have multiple columns and each column has a unique name. The main advantage of this model is its ability to represent data in a simplified format. The process of ~~manip~~ manipulating record is simplified with use of certain key attributes used to retrieve data.



Activity code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

key=24

Activity code	Date	Route No.
24	01/12/01	1-95
24	02/08/01	1-66

Fig. Relational model

### 3) Self-describing data models:-

Self-describing data model is a semi-structured data model. The data stored in this model is generally associated with a scheme that is contained within the data property known as self-describing property. This model is referred to as self-describing because it not only contains the database itself but also meta-data which defines and describes the data and relationships between the tables in the database.

### ⊗. Concept of Schema and Instance:

Database change over time as information is inserted, deleted and updated. The collection of information stored in database at a particular moment called an instance of database. The overall design of the database is called database schema.

According to the level of abstraction schema are divided into physical schema, logical schema and subschemas. The physical schema describes the database design at the physical level. Logical schema describes database design at logical level. Database system may have several schemas. At the view level, it is called subschemas (can be query). It describes different views of database.

Logical schema is more important for the development of applications programs. Programmer constructs applications by using logical schema. The physical schema is hidden under the logical schema and it can change without affecting application programs.



## ⑧ Three-Schema Architecture and Data Independence:

Among many of the characteristics of database approach three most important characteristics listed below specify an architecture for database systems called three-schema architecture that was proposed to help active and visualize three characteristics:

- i) Use of catalog to store the database description so as to make it self-describing.
- ii) Insulation of programs and data.
- iii) Support of multiple user views.

The goal of three-schema architecture is to separate the user applications from the physical database.

- a) Internal level → The internal level has an internal schema, which describes the physical structure of database. The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database.
- b) Conceptual level → The conceptual level has a conceptual schema, which describes the structure of the whole database for a community of users. The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations and constraints.
- c) External or view level → It includes a number of external schemas or user views. Each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database from that user groups.



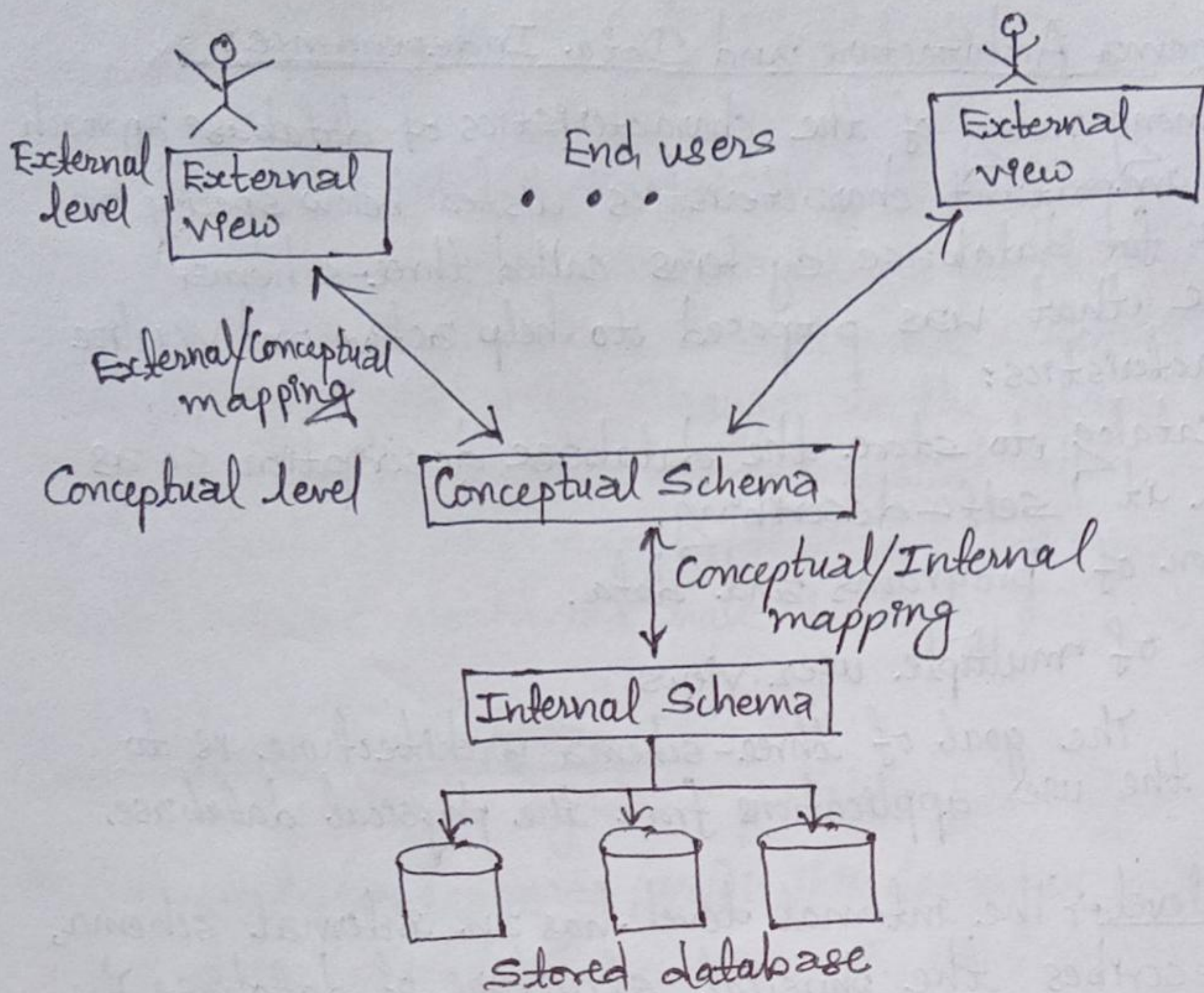


Fig. Three-Schema architecture.

The three-schema architecture is a convenient tool with which the user can visualize the schema levels in a database system. Most DBMSs do not separate the three levels completely and explicitly but they support the three-schema architecture to some extent. The three-schemas are only descriptions of data the actual data is stored at physical level only. The process of transforming requests and results between levels are called mappings.

### # Data Independence:

The capacity to change the schema at one level of database system without having to change the schema at the next higher level is called data independence. Following are the two types of data independence:

1) Logical data independence → It is the capacity to change the conceptual schema without having to change external schemas or application programs. We may change the constraints, or to reduce database. Only the view definition



and the mappings need to be changed in a DBMS that supports logical data independence.

ii) Physical data independence → It is the capacity to change the internal schema without having to change the conceptual schema. Hence, the external schemas need to be changed as well. Changes to the internal schema may be needed because some physical files were ~~recog~~ reorganized. If the same data as before remains in the database, we should not have to change the conceptual schema.

### \* Database Languages:-

Database system provides following languages:-

(A) Data Definition Language (DDL):- This language is used to define data structures and specially database schemas. These statements are used to create, alter or drop data structures. ALTER, CREATE, DROP are some examples of DDL. It also allows to define storage structure and access methods for database system.

(B) Storage Definition Language (SDL):- This language is used to define internal schema. It defines that what will be the physical structure of database like how many bites per field will be used, what will be the order of fields and how records will be accessed etc.

(C) View Definition Language (VDL):- This language is used to specify user views and their mapping to conceptual schema. It defines the subset of records available to classes of users. It creates virtual tables and the view appears to users like conceptual level. It specifies user interfaces.



④ Data Manipulation Language (DML):- It is used at conceptual level and external level and is used to perform operations like Query, Delete Update or Insert. Read Only Queries are also sometimes considered as component of DML. It modifies ~~that~~ the data but not schema or database objects. This language is further divided into following two types:-

i) Procedural DML → It is also called low-level DML. It simply contains series of computational steps to be carried out. Any procedure might be called at any point during program execution. In this user required to specify what data are needed and how they get those data.

ii) Non-Procedural DML → It is also called high-level or declarative DML. It describes the logic of computation without describing its control flow. In this user only required to what data needed without specifying how to get those data.

### ⊗ DBMS Interfaces:

User-friendly interfaces provided by a DBMS may include following:-

① Menu-based Interface for Web Clients or Browsing:- These interfaces present the user with list of options called menus that lead the user through formulation of a request. Pull-down menus are a very popular technique in Web-based user interfaces. They are also often used in browsing interfaces which allow user to look through contents of a database.

② Apps for Mobile Devices:- These interfaces present mobile users with access to their data. For example banking, reservations and insurance companies provide apps that allow users to access their data through a mobile phone or mobile device. The apps have built-in programmed interfaces that typically allow users to login using their name and password.



③ Forms-based Interfaces: - A forms-based interface displays a form to each user. Users can fill out all of the form entries to insert new data. Forms are usually designed and programmed for naive users as interfaces to canned transactions.

④ Graphical User Interfaces: - A GUI typically displays a schema to the user in diagrammatic form. The user then can specify a query by manipulating the diagram. In many cases, GUIs utilize both menus and forms.

⑤ Natural Language Interfaces: - These interfaces accept requests written in English or some other language and attempt to understand them. The natural language interface refers to the words in its schema, as well as to the set of standard words in its dictionary that are used to interpret the request.

⑥ Keyword-based Data Search: - These are somewhat similar to web search engines which accept strings of natural language words and match them with documents at specific sites or web pages. They use predefined indexes on words and use ranking functions to retrieve and present resulting documents in a decreasing degree of match.

⑦ Database System 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 hardware, people & techniques of handling database.



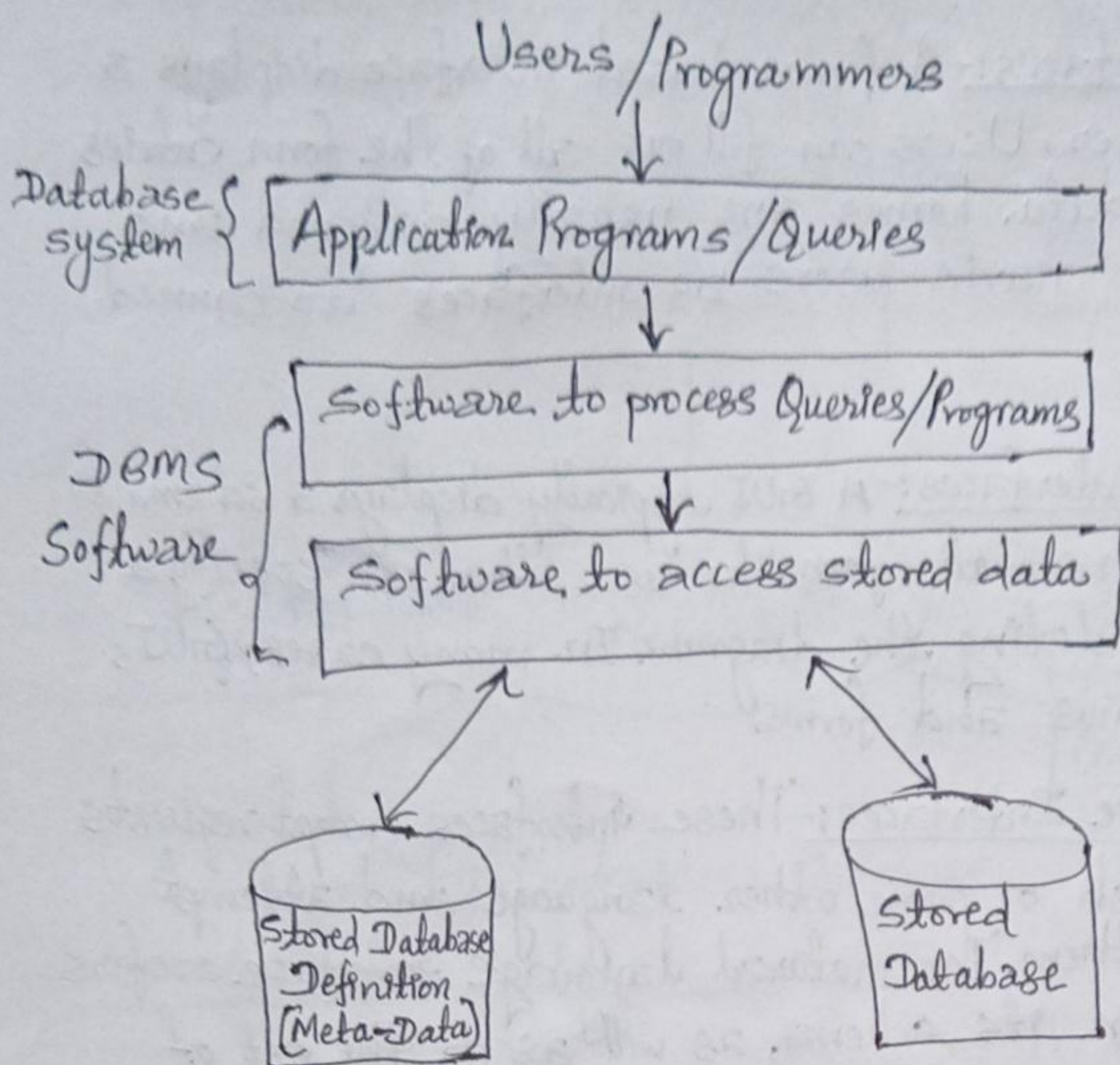


Fig. Database System Environment.

Here the hardware in a database environment means the computers and computer peripherals that are being used to manage a database, and the software means the whole thing right from the operating system (OS) to the application programs that include database management software like M.S. Access or SQL Servers. Again the people in the database environment include those people who administrate and use the system. The techniques are the rules concepts and instructions given to both the people and the software along with the data with the group of facts and information positioned within the database environment.

### ⊗. Centralized Database System:-

In centralized system, all programs run on the main host computer, including the DBMS, the application that accesses the database and the communication facilities that send and receive data from the users terminals. The users access the database through either locally connected or dial-up (remote) terminals. The terminals are generally dumb, having little or no processing



power of their own and consists of only a screen, keyboard and hardware to communicate with the host.

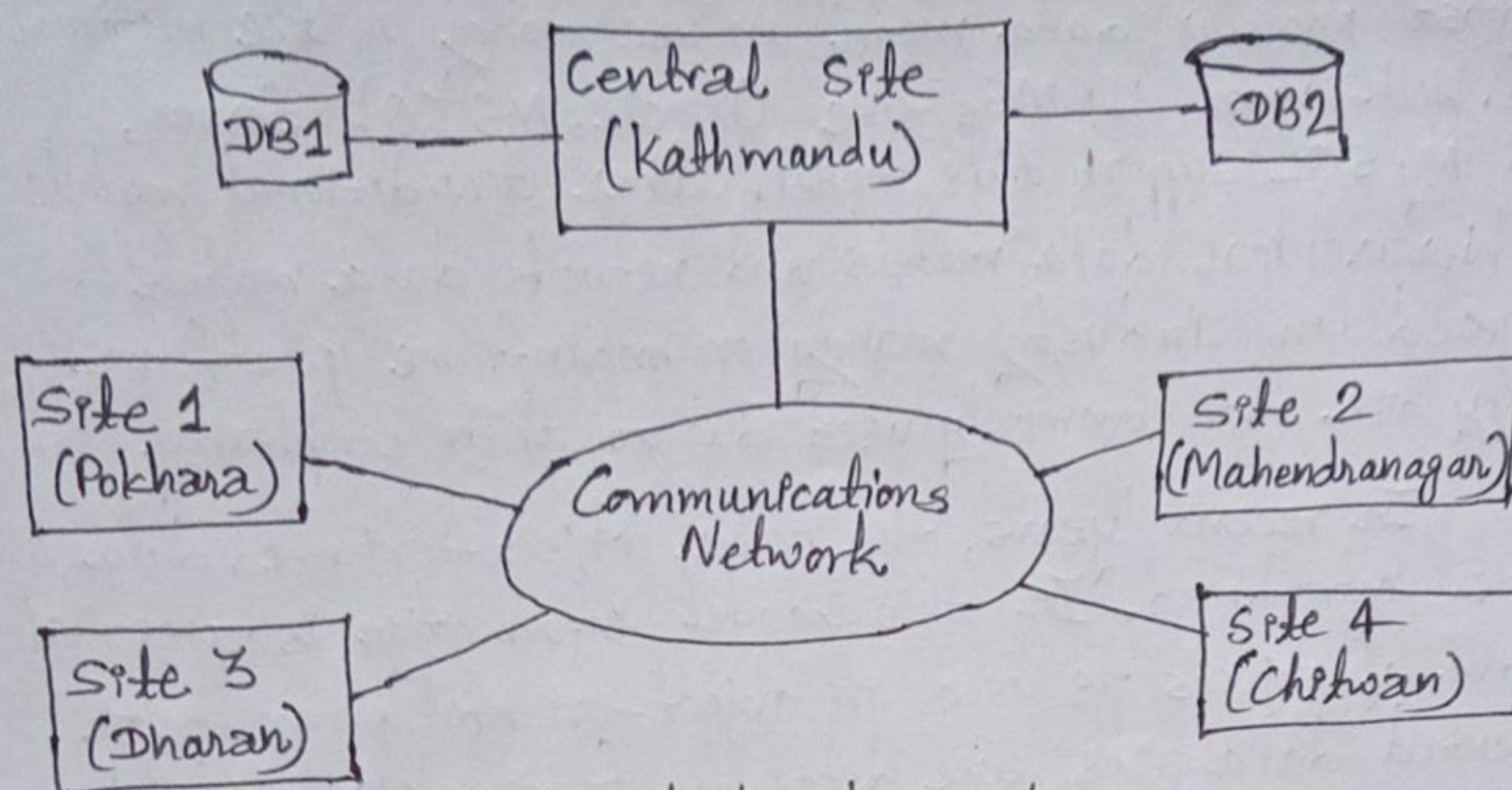


Fig. Centralized Database

### ⊗. Client/Server Database System:-

In a generalized concept, client PC is the computer from where the user requests for data and information and the server provides the requested information. The database application on the client PC referred to as the 'front end system' that handles all the screen and user input/output processing. The 'back end system' on the database server handles data processing and disk access.

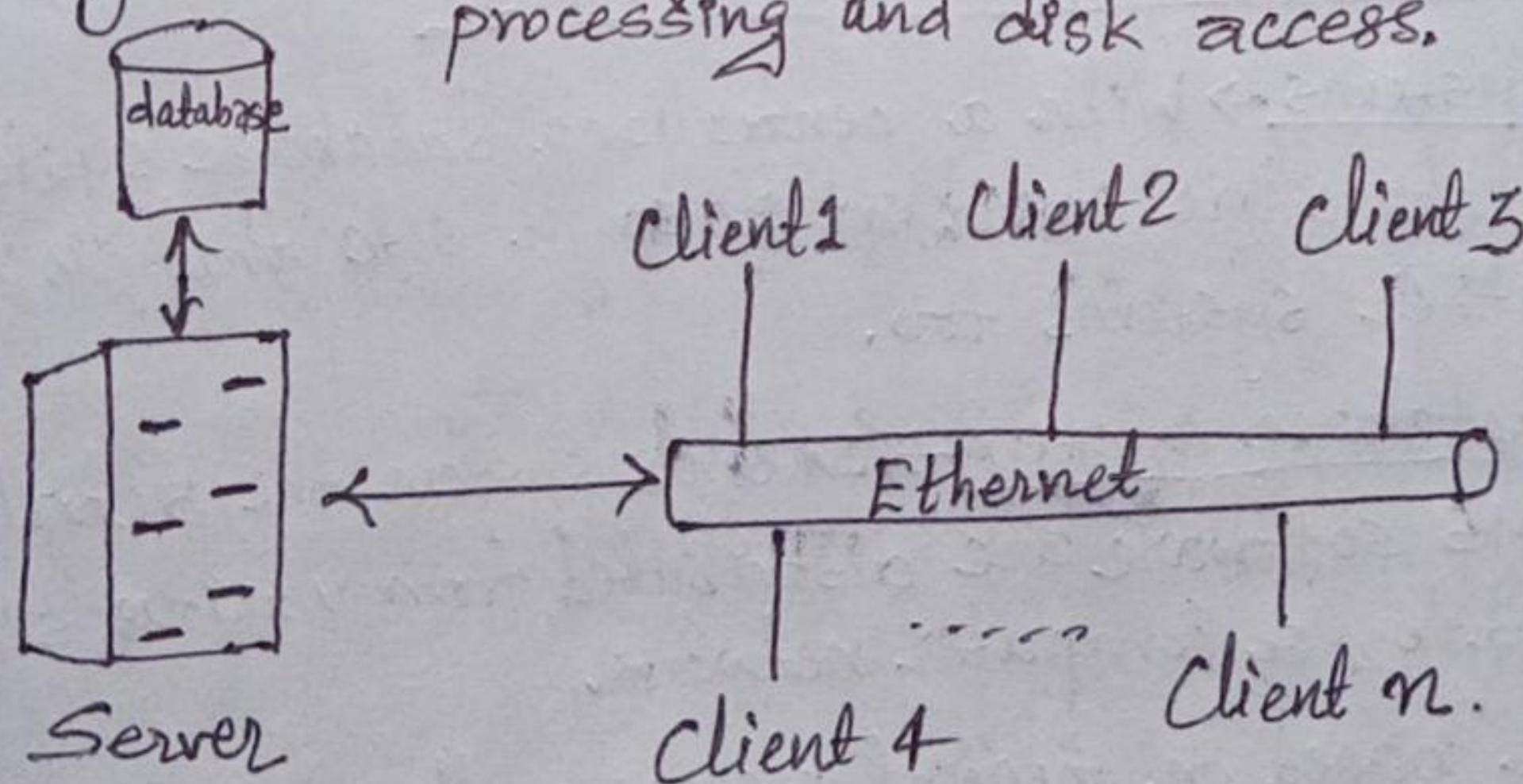


Fig. Client server database system.

Client server approach is implemented by two approaches: two-tier architecture and three-tier architecture.



## ⊗. Classification of Database Management Systems:-

### Ⓐ Classification based on data model:

The most popular data model in use today is the relational data model. Well-known DBMSs like Oracle, MS SQL Server, DB2 and MySQL support this model. Other traditional models such as hierarchical data models and network data models are still used in industry mainly on mainframe platforms. However, they are not commonly used due to their complexity.

In recent years, the newer object-oriented data models were introduced. In this model information is represented in the form of objects as used in object-oriented programming. Object-oriented databases are different from relational databases, which are table-oriented.

### Ⓑ Classification Based on user numbers:

A DBMS can be ~~ea~~ classified based on the number of users it supports. It can be a single-user database system, which supports one user at a time, or a multiuser database system, which supports multiple users concurrently.

### Ⓒ Classification based on number of sites:

➤ Centralized systems → With a centralized database system, the DBMS and database are stored at a single site that is used by several other systems too.

➤ Distributed database system → In this system the actual database and the DBMS software are distributed from various sites that are connected by a computer network.

### Ⓓ Classification based on type of access path:

➤ Homogenous distributed database systems → They use the same DBMS software for multiple sites. Data exchange between these various sites can be handled easily.

➤ Heterogenous distributed database systems → In this system different sites might use different DBMS software, but there is additional common software to support data exchange between these sites.