Unit-1

Database and Database Users:-

Data → Data are the raw facts that are found after some experiment, observation or after some experiments. Data itself do not provide any meaning but after processing it becomes information.

Database → The collection of data organized in some specific manner is known as database. For example, university database for maintaining information about students, courses and grades in university.

Database Management System → The database, its processing methods and the set of rules and conditions to be followed is collectively known as database management system (DBMS). It is the way to store data and also provides mechanism for mainupulation of data. It is basically just a computerized record-keeping system. The primary goal of DBMS is to store and retrive data in both convinent and efficient manner. Oracle, MySQL, SQL-Server, MS Access etc. are some examples of DBMS.

⊛ Applications of DBMS:

- Banking → To store information about customers, their account number, balance etc.
- Airlines → For reservations and schedule information.
- Telecommunication → To keep records of customers, call made, balance left, generating monthly bills etc.
- Sales → To keep information of customers, products list, purchase information etc.
- Universities → To keep record of students, courses, marks of students etc.
- Human Resources → To keep record of employee, their ~~salert~~ salery etc.
- Manufacturing → To store orders, tracking production of items etc.

⊛. Characteristics of the Database Approach:

In past traditional file processing approach was used in which each user defines and implements the files needed for a specific software application as part of programming the application. But in database approach, a single repository maintains data that is defined once and then accessed by various users repeatedly through queries, transactions and application programs. The main characteristics of the database approach versus the file-processing approach are as follows:-

1) Self-describing nature of a database system:

The database system contains not only the database itself but also a complete description. The definition is stored in the DBMS catalog which contains information such as structure of each file, the type and storage format of each data item. The information stored in the catalog is called meta-data and it describes the structure of the primary database.

The newer types of database systems do not require meta-data. Rather the data is stored as self-describing data that includes the data item names and data values together in one structure. These database systems are known as NOSQL systems.

2) Insulation between programs and data, and data abstraction:

In traditional file processing, the structure of data files is embedded in the application programs, so any changes to the structure of a file may require changing all programs that access the file. But DBMS access programs do not require such changes in most cases. The structure of data files is stored in the DBMS catalog seperately from the access programs. We call this property program-data-independence.

iii) Support of Multiple Views of the Data:

A database typically has many types of users, each of whom may require a different perespective or view of the database. A view may be a subset of the database or it may contain virtual data. Some users may not need to be aware of whether the data they refer to is stored or derived. A multiuser DBMS whose users have a variety of distinct applications must provide facilities for defining multiple views. For example one user of database may be interested only in accessing and printing the transcript and a second user who is interested only in checking that students have taken all the prerequisites of each course.

iv) Sharing of data and multiuser transaction processing:

A DBMS must allow multiple users to access the database at the same time. This is essential if data for multiple applications is to be integrated and maintained in a single database. The database must include concurrency control software to ensure that several users trying to update the same data.

DBMS should ensure that concurrent transactions operate correctly and efficiently. These types of applications are generally called online transaction processing (OLTP) applications. The concept of ~~a~~ a transaction has become central to many database applications. A transaction is an executing program or process that includes one or more database accesses such as reading or updating of database records.

⊛. Actors on the Scene:-

For a small personal database such as the list of addresses, typically one person defines, constructs and manipulates the database and there is no sharing. But in large organizations many people are involved in the design, use and maintenance of a large database with hundreds or thousands of users. The people whose jobs involve the day-to-day use of a large database are called actors on the scene.

i) Database Administrators → In database environment the primary resource is database itself and secondry resource is the DBMS and related software. Administrating these resources is the responsibility of the database administrator (DBA). The DBA is responsible for authorizing access to the database, coordinating and monitoring its use, and acquiring software and hardware resources as needed. The DBA is also accountable for problems such as security breaches and poor system response time.

ii) Database Designers → Database designers are responsible for identifying the data to be stored in the database and for choosing appropriate structures to represent and store this data. It is the responsibility of database designers to communicate with all prospective database users in order to understand their requirement and to create a design that meets the requirements. The database design must be capable of supporting the requirements of all user groups.

iii) End Users → End users are the people whose jobs require access to the database for querying, updating and generating reports. There are several categories of end users:

- Causal end users → They occasionally access the database, but they may need different information each time. They use a sophisticated database query interface to specify their requests.
- Navie end users → They constantly query and update the database using standard types of queries and updates called canned transactions that have been carefully programmed and tested.
- Sophisticated end users → It includes engineers, scientists, business analysists and others in order to meet their complex requirements.

• Standalone users → They maintain personal databases by using ready-made, program packages that provide easy-to-use menu-based or graphics-based interfaces.

iv) System Analysists and Application Programmers (Software Engineers):

System analysists determine the requirement of end users specially navie and parametric end users and develop specifications for standard canned transactions that meet the requirements. Application programmers implement these specifications as programs then they test, debug, document and maintain these canned transactions.

⊛ Workers Behind the Scene:

Those who work to maintain the database system environment but who are not actively interested in the database contents as part of their daily job are called workers behind the scene. They include following categories:

i) DBMS system designers and implementors → They design and implement the DBMS modules and interfaces as a software package. A DBMS is very complex software consisting many components including modules for implementing catalog, query language processing, interface processing, handling data recovery and security etc. The DBMS must interface with other system software such as operating system and compilers for various programming languages.

ii) Tool developers → They design and implement tools which are software packages that facilitate database modeling and design and improve performance. Tools are optional packages that are often purchased seperately. In many cases, independent software vendors develop and market these tools.

iii) Operators and maintenance personal → They are also called system administration personnel and are responsible for the actual running and maintenance of the hardware and software environment for the database system.

⊛. Advantages of using the DBMS Approach:-

The DBA must utilize the different capabilities to accomplish a variety of objectives related to the design, administration and use of a large multiuser database. Following are some advantages of using DBMS and the capabilities that a good DBMS should possess.

i) Controlling Redundancy:- The file based management systems contained multiple files that were stored in many different locations in a system or even across multiple systems. Because of this, there were sometimes multiple copies of the same file which lead to data redundancy.

This is prevented in database as there is a single database and any change in it is reflected immediately. Because of this, there is no chance of encountering duplicate data.

ii) Restricting unauthorized access:- Data Security is vital concept in a database. Only authorised users should be allowed to access the database and their identity should be authenticated using a username and password. Unauthorised users should not be allowed to access the database under any circumstances as it violates the integrity constraints.

iii) Providing Persistant Storage:- Programming languages typically have complex data structures such as struct or class definitions in C++ or Java. The values of program variables or objects are discarded once a program terminates unless the programmer explicitly stores them in permanent files, which often involves converting these complex structures into a format suitable for file storage. When need arises to read this data, once more the programmer must convert the file.

The DBMS software is compatible with these programming languages such as C++ and Java and automatically performs necessary conversions. Hence a complex object can be stored in an object-oriented DBMS. Such an object is said to be persistent, since it survives the termination of program execution and can be later directly retrived by another program.

iv) Providing Backup and Recovery:- The backup and recovery subsystem of DBMS is responsible for recovery. For example, if the computer fails in the middle of a complex update transaction, the recovery system is responsible for making sure that the database is restored to the state it was in before the transaction started executing. Disk backup is also necessary in case of a catastrophic disk failure.

v) Providing Multiple User Interfaces:- Many types of users with varying levels of technical knowledge use a database, so a DBMS should provide a variety of user interfaces. These includes apps for mobile users, query language for causal users, programming language interfaces for application programmers, forms and commands for parametric users, and menu-driven interfaces and natural language interface for standalone users. Both forms-style interfaces and menu-driven interfaces are commonly known as graphical user interfaces (GUIs).

vi) Flexibility:- It maybe necessary to change the structure of a database as requirements change. For example, a new user group may emerge that needs information necessary to add a file to the database or to extend the data elements in an existing file. Modern DBMSs allow certain ~~database without~~ structure of the database types of evolutionary changes to the and the existing application programs. without affecting the stored data

vii) Availability of Up-to-Date Information → As soon as users update is applied to the database, all other users can immediately see that update. This availability of up-to-date information is essential for many transaction-processing applications such as reservation systems or banking databases.

viii) Economics of Scale:- The DBMS approach reduces the amount of wasteful overlap between activities of data-processing personnel in different projects or departments. This enables the whole organization to invest in more powerful processors, storage devices, rather than having each department purchase its own low performance equipment. This reduces overall costs of operation and management.

⊛. Purpose of DBMS / Why DBMS is required?

Traditionally, file processing system was used to manage information. It used to store data in various files of different application programs to extract or insert data to appropriate file.

File processing system has several drawbacks due to which DBMS is required. DBMS removes the problems found in file processing system. Some major problems of file processing system are:

i) Data redundancy and inconsistency.
ii) Difficult in accessing data.
iii) Data isolation.
iv) Integrity problem.
v) Security problem
vi) Atomicity problem etc.

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:

@. **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.

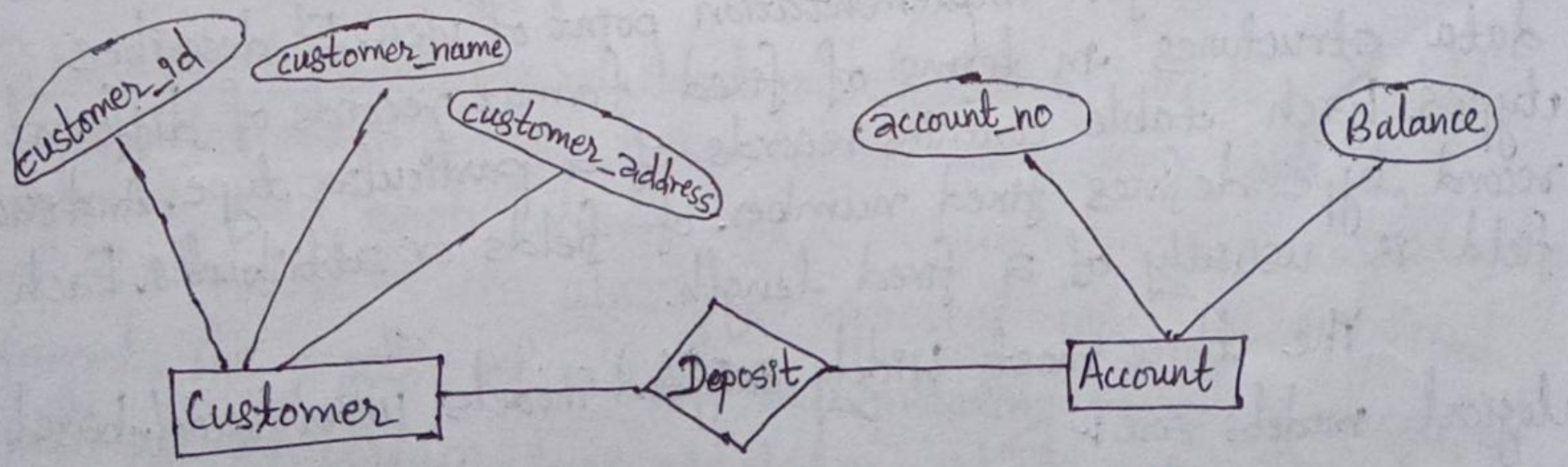

Fig. Sample of E-R model. (ER diagram)

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).

(b). 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 arbitarily deep level of nesting. Another feature of object-oriented model is inheritance i.e, new class can be derived from the existing class.

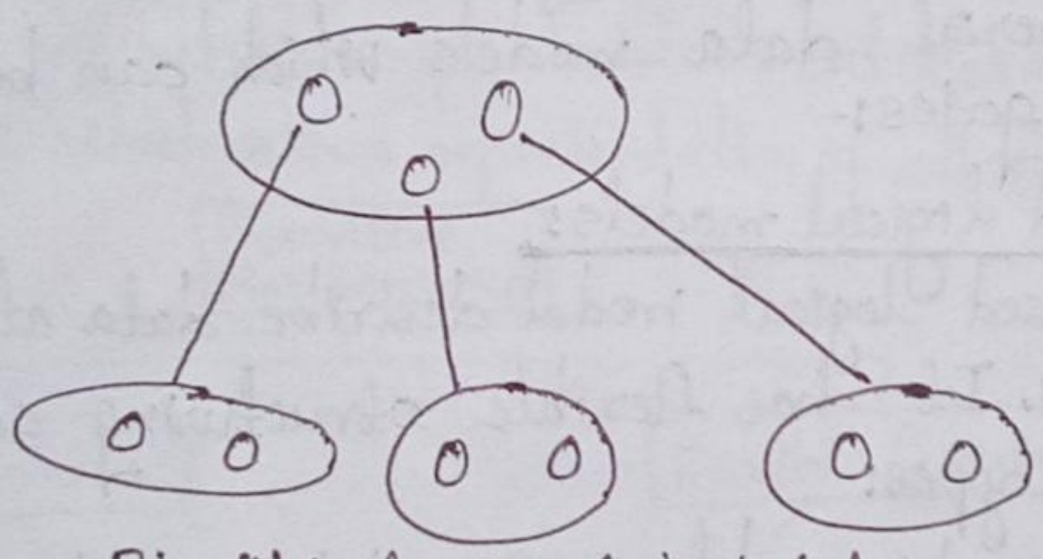

Fig. Object-Oriented Model

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. 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:-

@. 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.

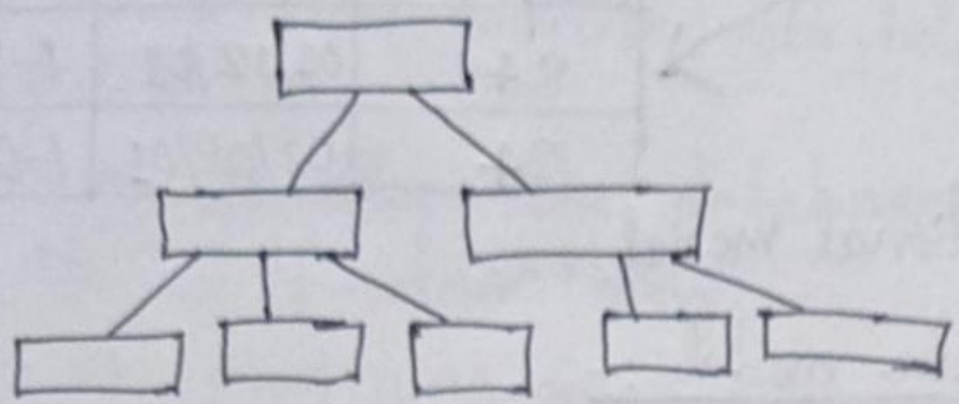

Fig. Structure of Hierarchical model

ⓑ. 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 arbitary groups. The main advantage of this model is its representation of relationships between entities is implemented using pointers which allows representation of arbitary relationship.

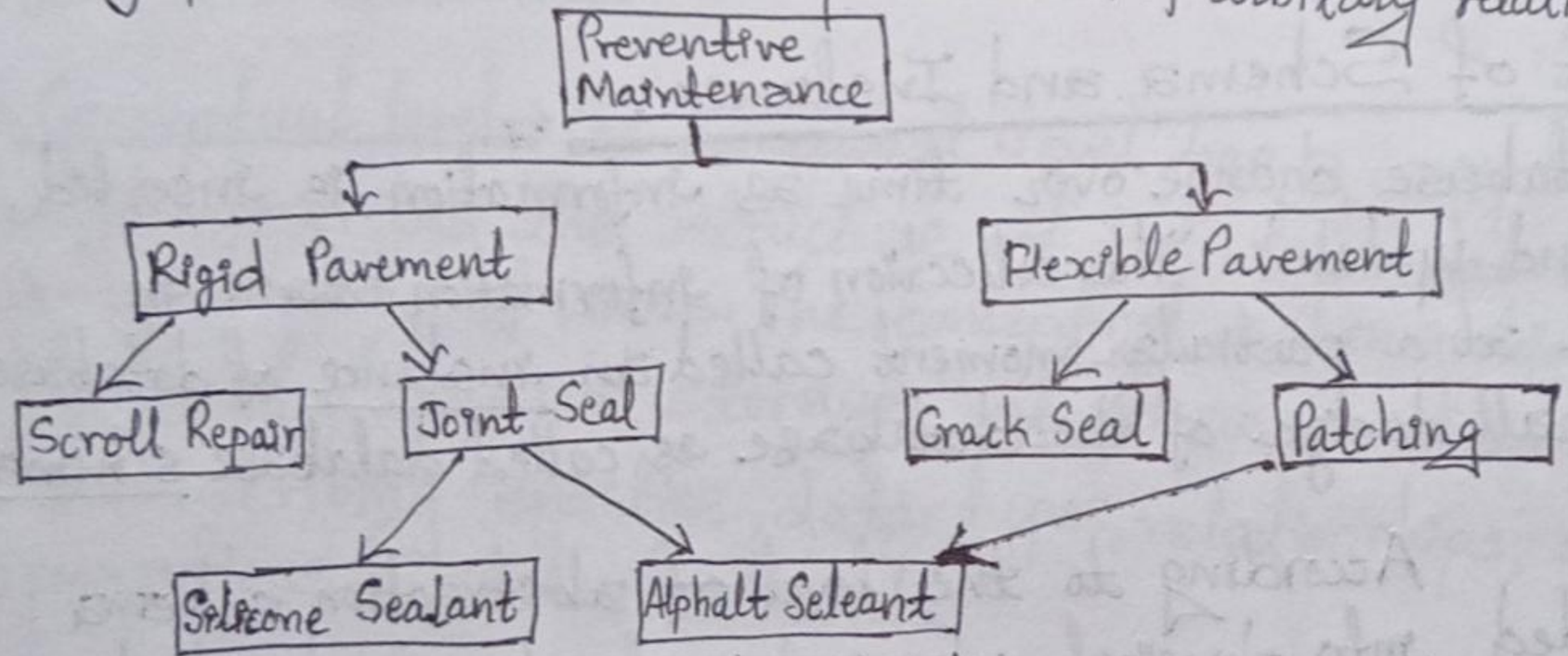

Fig. Network model

©. 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 ~~mainup~~ manipulating record is simplefied with use of certain key attributes used to retrive data.

Activity code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

key=24

Activity code	Date	Route No.
24	01/12/01	I-95
24	02/08/01	I-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 application 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.

⊗. 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 mak it self-describing.
ii) Insulation of programs and data.
iii) Support of multiple user views.

The goal of three-schema architecture is to seperate 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.

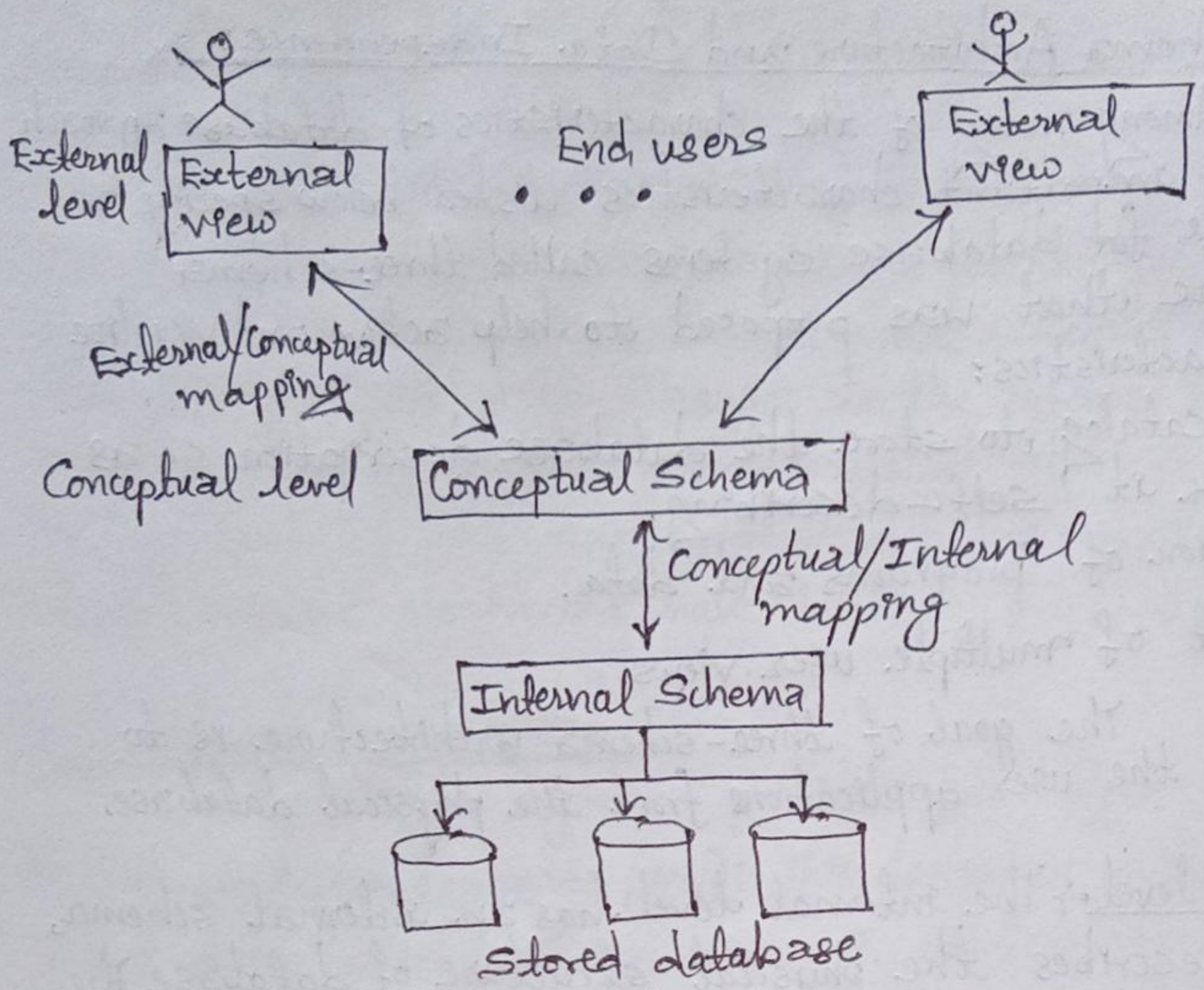

Fig. Three-Schema architecture.

The three-schema architecture is a convinent tool with which the user can visualize the schema levels in a database system. Most DBMSs do not seperate 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:-

Ⓐ 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.

Ⓑ. 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.

Ⓒ 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.

(d). 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 d~~ 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.

(e). DBMS Interfaces:

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

(a) 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.

(b). 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.

© 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 navie users as interfaces to canned transactions.

(d) Giraphical User Interfaces:- A GUI tipically 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.

(e) 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.

(f). 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 retrive and present resulting documents in a decreasing degree of match.

(g). 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.

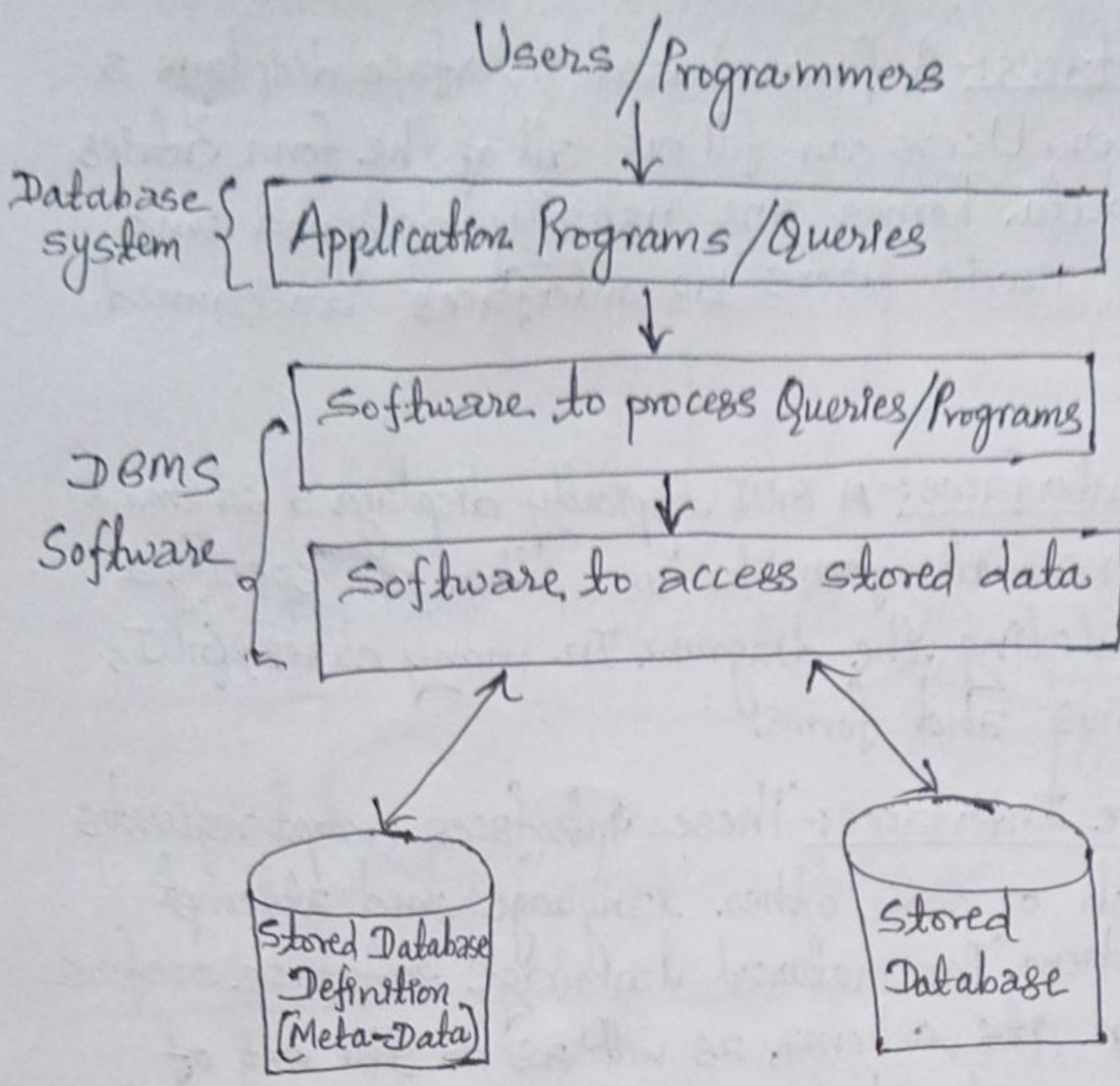

Fig. Database System Environment.

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.

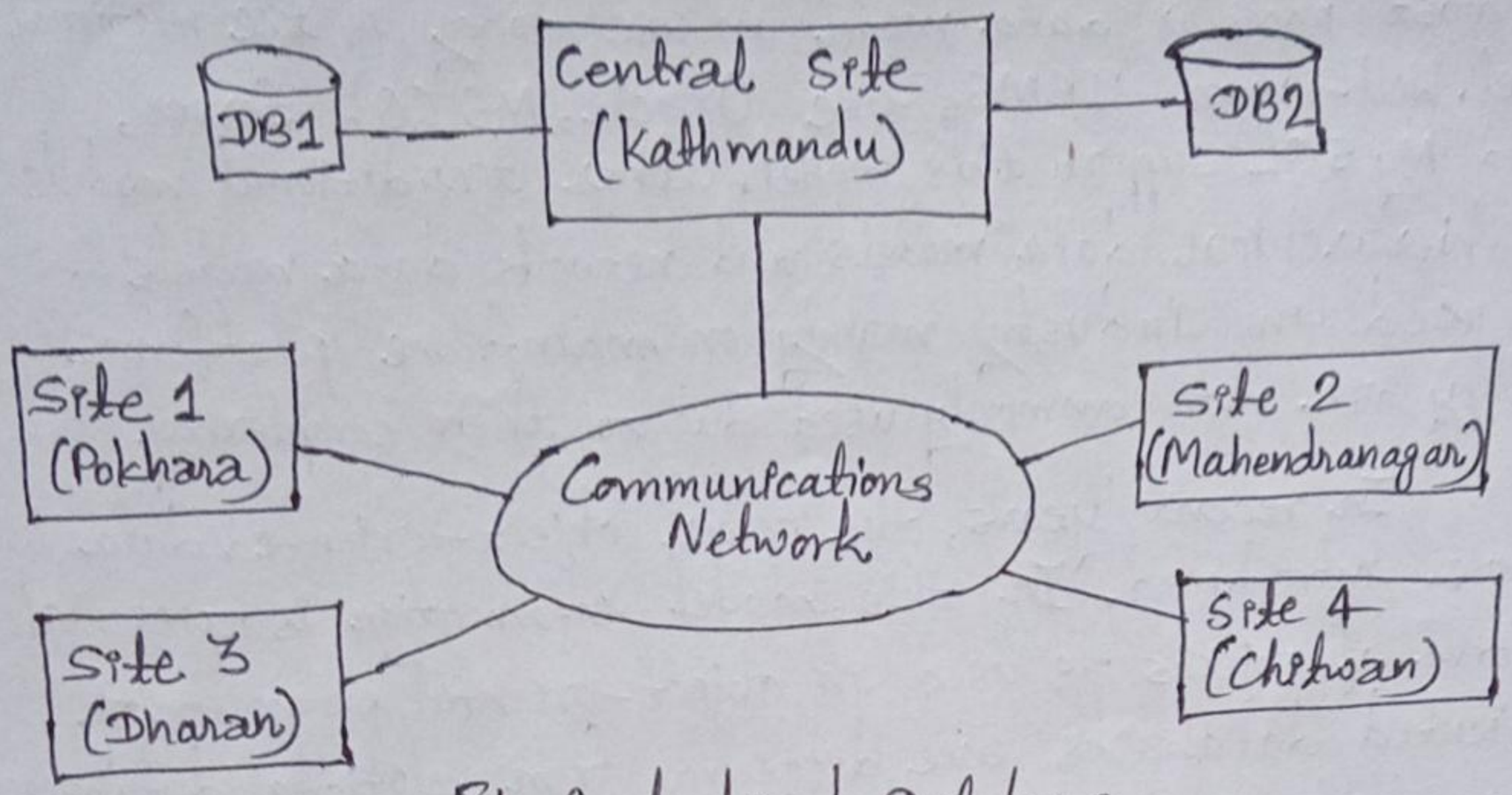

Fig. Centralized Database

⊛. 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.

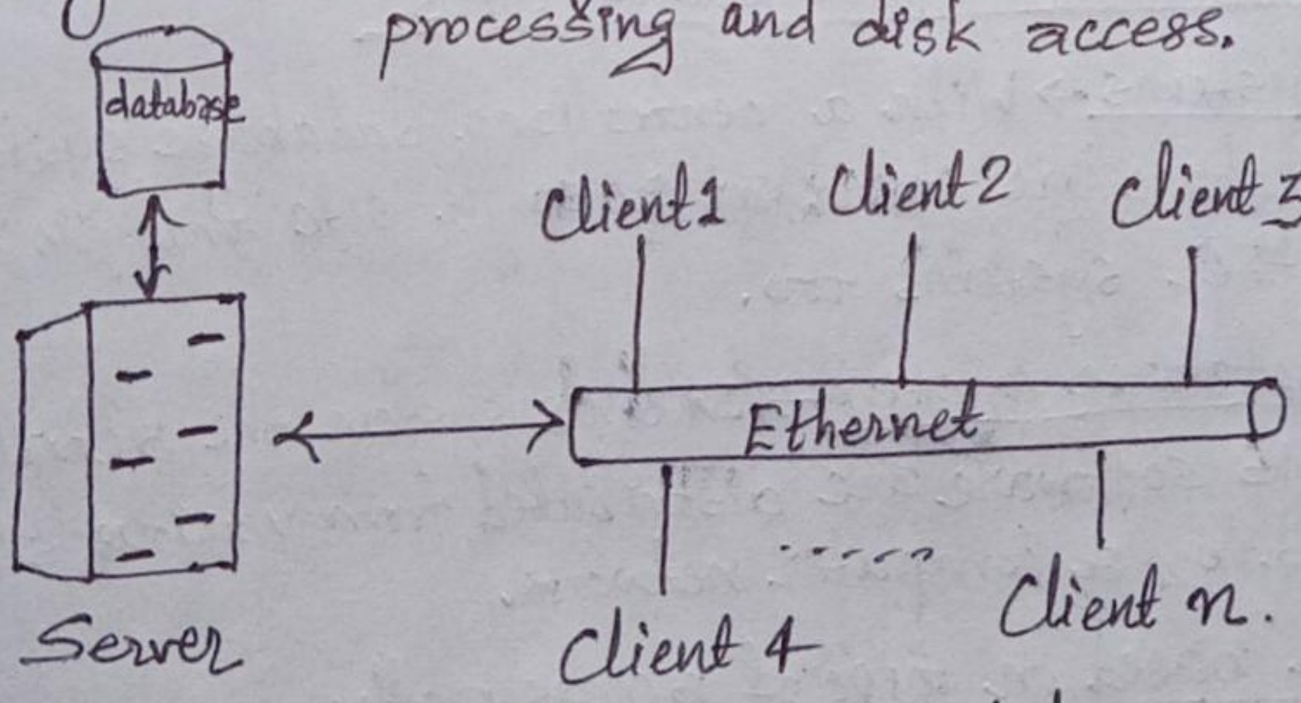

Fig. Client server database system.

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.