The DML is of the two types :

(*i*) *Procedural DML* : It allows the user to tell the system what data is needed and how to retrieve it.

(*ii*) *Non-procedural DML* : It allows the user to state what data are needed, rather than how it is to be retrieved. More about DML in chapter 7.

1.5.5 Fourth-Generation Language (4-GL)

It is a compact, efficient and non-procedural programming language used to improve the efficiency and productivity of the DBMS. In this, the user defines what is to be done and not how it is to be done. The 4-GL has the following components in it. These are :

(*a*) Query languages
(*b*) Report
(*c*) Spread sheets
(*d*) Database languages
(*e*) Application generators
(*f*) High level languages to generate application program.

System Query Language (SQL) is an example of 4-GL. More about SQL in Chapter 7.

1.6 SCHEMAS, SUBSCHEMA AND INSTANCES

The plans of the database and data stored in the database are most important for an organization, since database is designed to provide information to the organization. The data stored in the database changes regularly but the plans remain static for longer periods of time.

1.6.1 Schema

A schema is plan of the database that give the names of the entities and attributes and the relationship among them. A schema includes the definition of the database name, the record type and the components that make up the records. Alternatively, it is defined as a frame-work into which the values of the data items are fitted. The values fitted into the frame-work changes regularly but the format of schema remains the same *e.g.*, consider the database consisting of three files ITEM, CUSTOMER and SALES. The data structure diagram for this schema is shown in Figure 1.5. The schema is shown in database language.

Generally, a schema can be partitioned into two categories, *i.e.*, (*i*) *Logical schema* and (*ii*) *Physical schema.*

(*i*) The *logical schema* is concerned with exploiting the data structures offered by the DBMS so that the schema becomes understandable to the computer. It is important as programs use it to construct applications.

(*ii*) The *physical schema* is concerned with the manner in which the conceptual database get represented in the computer as a stored database. It is hidden behind the logical schema and can usually be modified without affecting the application programs.

The DBMS's provide DDL and DSDL to specify both the logical and physical schema.

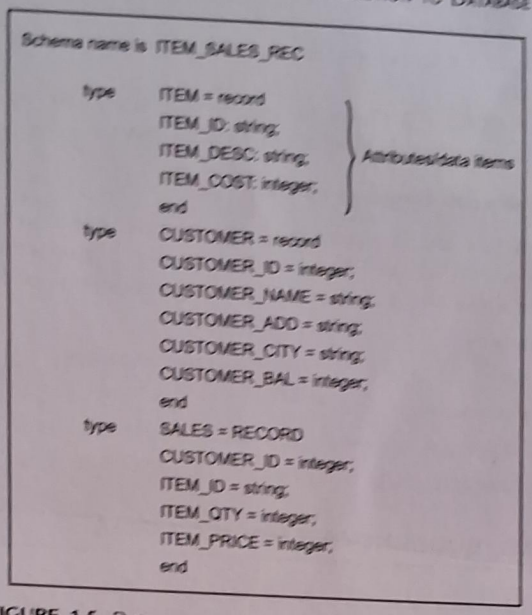

FIGURE 1.5. Data structure diagram for the item sales record.

1.6.2 Subschema

A subschema is a subset of the schema having the same properties that a schema has. It identifies a subset of areas, sets, records, and data names defined in the database schema available to user sessions. The subschema allows the user to view only that part of the database that is of interest to him. The subschema defines the portion of the database as seen by the application programs and the application programs can have different view of data stored in the database.

The different application programs can change their respective subschema without affecting other's subschema or view.

The Subschema Definition Language (SDL) is used to specify a subschema in the DBMS.

1.6.3 Instances

The data in the database at a particular moment of time is called an *instance* or a *database state*. In a given instance, each schema construct has its own current set of instances. Many instances or database states can be constructed to correspond to a particular database schema. Everytime we update (*i.e.*, insert, delete or modify) the value of a data item in a record, one state of the database changes into another state. The Figure 1.6 shows an instance of the ITEM relation in a database schema.

ITEM

ITEM-ID	ITEM_DESC	ITEM_COST
1111A	Nutt	3
1112A	Bolt	5
1113A	Belt	100
1144B	Screw	2

FIGURE 1.6. An instance/database state of the ITEM relation.

1.7 THREE LEVEL ARCHITECTURE OF DATABASE SYSTEMS (DBMS)

The architecture is a framework for describing database concepts and specifying the structure of database system. The three level architecture was suggested by ANSI/SPARC. Here database is divided into three levels **external level, conceptual level** and **internal level** as shown in Figure 1.7.

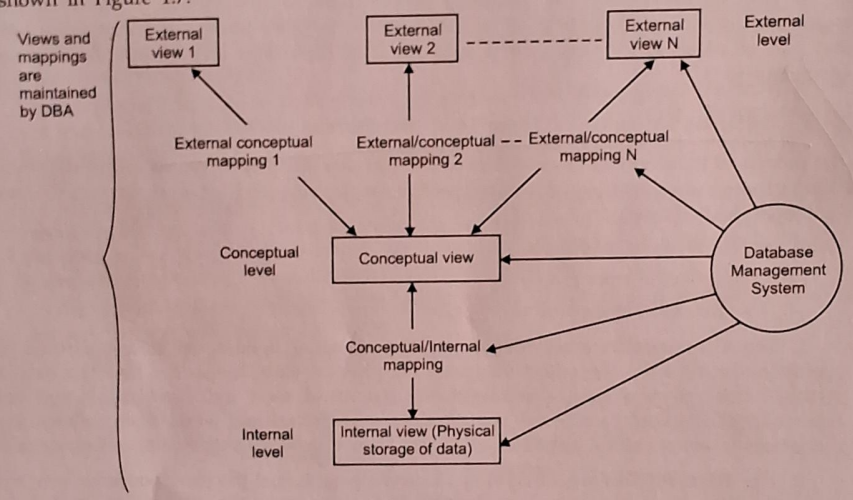

FIGURE 1.7. Three level architecture of DBMS.

1.7.1 Levels or Views

The three levels or views are discussed below:

(*i*) Internal Level : Internal level describes the actual physical storage of data or the way in which the data is actually stored in memory. This level is not relational because data is stored according to various coding schemes instead of tabular form (in tables). This is the low level representation of entire database. The internal view is described by means of an internal schema.

The internal level is concerned with the following aspects:

- Storage space allocation
- Access paths
- Data compression and encryption techniques
- Record placement etc.

The internal level provides coverage to the data structures and file organizations used to store data on storage devices.

(ii) Conceptual Level : The conceptual level is also known as logical level which describes the overall logical structure of whole database for a community of users. This level is relational because data visible at this level will be relational tables and operators will be relational operators. This level represents entire contents of the database in an abstract form in comparison with physical level. Here conceptual schema is defined which hides the actual physical storage and concentrate on relational model of database.

(iii) External Level : The external level is concerned with individual users. This level describes the actual view of data seen by individual users. The external schema is defined by the DBA for every user. The remaining part of database is hidden from that user. This means user can only access data of its own interest. In other words, user can access only that part of database for which he is authorized by DBA. This level is also relational or very close to it.

1.7.2 Different Mappings in Three Level Architecture of DBMS

The process of transforming requests and results between the three levels are called mappings. The database management system is responsible for this mapping between internal, external and conceptual schemas.

There are two types of mappings:

1. Conceptual/Internal mapping.
2. The External/Conceptual mapping.

1. The Conceptual/Internal Mapping : This mapping defines the correspondence or operations between the conceptual view and the physical view. It specifies how the data is retrieved from physical storage and shown at conceptual level and vice-versa. It specifies how conceptual records and fields are represented at the internal level. It also allows any differences in entity names, attribute names and their orders, data types etc., to be resolved.

2. The External/Conceptual Mapping : This mapping defines the correspondence between the conceptual view and the physical view. It specifies how the data is retrieved from conceptual level and shown at external level because at external level some part of database is hidden from a particular user and even names of data fields are changed etc.

There could be one mapping between conceptual and internal level and several mappings between external and conceptual level. The **physical data independence** is achieved through conceptual/internal mapping while the **logical data independence** is achieved through external/conceptual mapping. The information about the mapping requests among various schema levels are included in the system catalog of DBMS. When schema is changed at some level, the schema at the next higher level remains unchanged, only the mapping between the two levels is changed.

(*a*) Unifying Model.

(*b*) Frame Memory Model.

1.8.1.1 Record based Data Models

Record based data models represent data by using the record structures. These are used to describe data at the conceptual view level. These are named because the database is structured in a fixed format records of several types. The use of fixed length records simplify the physical level implementation of the database. These models lie between the object based data models and the physical data models. These models provide the concepts that may be understood by the end users. These data models do not implement the full detail of the data storage on a computer system. Thus, these models are used to specify overall logical structure of the database and to provide high level description of implementation. These are generally used in traditional DBMS's and are also known as 'Representational Data Models'. The various categories of record based data models are as follows:

(*i*) Hierarchical Data Model

(*ii*) Network Data Model

(*iii*) Relational Data Model.

(*i*) Hierarchical Data Model : Hierarchical Data Model is one of the oldest database models. The hierarchical model became popular with the introduction of IBM's Information Management System (IMS).

The hierarchical data model organizes records in a tree structure *i.e.*, hierarchy of parent and child records relationships. This model employs two main concepts : Record and Parent Child Relationship. A record is a collection of field values that provide information of an entity.

A Parent Child Relationship type is a 1 : N relationship between two record types. The record type of one side is called the parent record type and the one on the N side is called the child record type. In terms of tree data structure, a record type corresponds to node of a tree and relationship type corresponds to edge of the tree.

The model requires that each child record can be linked to only one parent and child can only be reached through its parent.

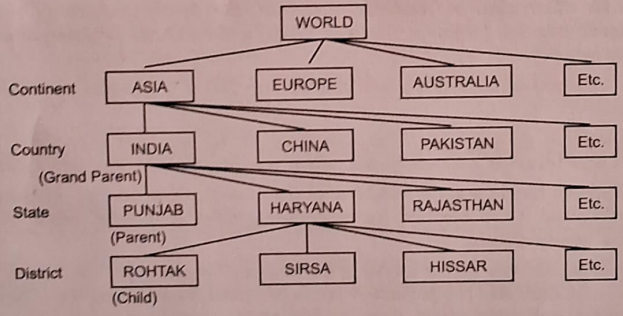

FIGURE 1.8. Hierarchical Model.

In the Figure 1.8, the 'WORLD' acts as a root of the tree structure which has many children's like Asia, Europe, Australia etc. These children can act as a parent for different countries such as ASIA continents acts as a parent for countries like India, China, Pakistan etc. Similarly these children can act as a parent for different states such as INDIA country acts as a parent for states Punjab, Haryana, Rajasthan etc. Further the same follows.

Consider child 'ROHTAK' which has a parent 'HARYANA' which further has a parent 'INDIA' and so on. Now 'India' will acts a grandparent for the child 'ROHTAK'.

The major advantages of Hierarchical Model are that it is simple, efficient, maintains data integrity and is the first model that provides the concept of data security. The major disadvantages of Hierarchical model are that it is complex to implement, Lacking of structural independence, operational anomalies and data management problem.

***(ii)* Network Data Model :** As a result of limitations in the hierarchical model, designers developed the Network Model. The ability of this model to handle many to many (N : N) relations between its records is the main distinguishing feature from the hierarchical model. Thus, this model permits a child record to have more than one parent. In this model, directed graphs are used instead of tree structure in which a node can have more than one parent. This model was basically designed to handle non-hierarchical relationships.

The relationships between specific records of 1 : 1 (one to one), 1 : N (one to many) or N : N (many to many) are explicitly defined in database definition of this model.

The Network Model was standardized as the CODASYL DBTG (Conference of Data System Languages, Database Task Group) model.

There are two basic data structures in this model—Records and Sets. The record contains the detailed information regarding the data which are classified into record types. A set type represents relationship between record types and this model use linked lists to represent these relationships. Each set type definition consists of three basic elements : a name for set type an owner record type (like parent) and a member record type (like child).

To represent many to many relationship in this model, the relationship is decomposed into two one to many (1 : N) relationships by introducing an additional record type called an Intersection Record or *Connection Record.*

The major advantages of Network Model are that it is conceptually simple, Handles more relationship types, promotes database integrity, data access flexibility and conformance to the standards.

The major disadvantages of Network Model are that it is complex and lack of structural independence.

***(iii)* Relational data Model :** The Relational Model was first introduced by Dr. Edgar Frank, an Oxford-trained Mathematician, while working in IBM Research Centre in 1970's.

The Relational Model is considered one of the most popular developments in the database technology because it can be used for representing most of the real world objects and the relationships between them.

The main significance of the model is the absolute separation of the logical view and the physical view of the data. The physical view in relational model is implementation dependent and not further defined.

The logical view of data in relational model is set oriented. A relational set is an unordered group of items. The field in the items are the columns. The column in a table have names.

1.7.3 Advantages of Three-level Architecture

The motive behind the three-level architecture is to isolate each user's view of the database from the way the database is physically stored or represented. The advantages of the three-level architecture are as follows :

1. Each user is able to access the same data but have a different customized view of the data as per the requirement.
2. The changes to physical storage organization does not affect the internal structure of the database. *e.g.*, moving the database to a new storage device.
3. To use the database, the user is no need to concern about the physical data storage details.
4. The conceptual structure of the database can be changed by the DBA without affecting any user.
5. The database storage structure can be changed by the DBA without affecting the user's view.

1.7.4 Data Independence

It is defined as the characteristics of a database system to change the schema at one level without having to change the schema at the next higher level. It can also be defined as the immunity of the application programs to change in the physical representation and access techniques of the database. The above definition says that the application programs do not depend on any particular physical representation or access technique of the database. The DBMS achieved the data independence by the use of three-level architecture. The data independence is of TWO types:

1. Physical Data Independence : It indicates that the physical storage structures or devices used for storing the data could be changed without changing the conceptual view or any of the external views. Only the mapping between the conceptual and internal level is changed. Thus, in physical data independence, the conceptual schema insulates the users from changes in the physical storage of the data.

2. Logical Data Independence : It indicates that the conceptual schema can be changed without changing the existing external schemas. Only the mapping between the external and conceptual level is changed and absorbed all the changes of the conceptual schema. DBMS that supports logical data independence, changes to the conceptual schema is possible without making any change in the existing external schemas or rewriting the application programs. Logical data independence also insulates application programs from operations like combining of two records into one or splitting an existing record into more than one records.

1.8 DATA MODELS

A data model is a collection of concepts that can be used to describe the structure of the database including data types, relationships and the constraints that apply on the data.

A data model helps in understanding the meaning of the data and ensures that, we understand.

- The data requirements of each user.
- The use of data across various applications.
- The nature of data independent of its physical representations.

A data model supports communication between the users and database designers. The major use of data model is to understand the meaning of the data and to facilitate communication about the user requirements.

Characteristics of Data Models

A data model must posses the following characteristics so that the best possible data representation can be obtained.

(*i*) Diagrammatic representation of the data model.

(*ii*) Simplicity in designing *i.e.*, Data and their relationships can be expressed and distinguished easily.

(*iii*) Application independent, so that different applications can share it.

(*iv*) Data representation must be without duplication.

(*v*) Bottom-up approach must be followed.

(*vi*) Consistency and structure validation must be maintained.

1.8.1 Types of Data Models

The various data models can be divided into three categories, such as

(*i*) Record Based Data Models.

(*ii*) Object Based Data Models.

(*iii*) Physical Data Models.

(*i*) Record Based Data Models : These models represent data by using the record structures. These models lie between the object based data models and the physical data models. These data models can be further categorised into three types:

(*a*) Hierarchical Data Model

(*b*) Network Data Model

(*c*) Relational Data Model.

(*ii*) Object Based Data Models : These models are used in describing the data at the logical and user view levels. These models allow the users to implicity specify the constraints in the data. These data models can be further categorised into four types:

(*a*) Entity Relationship Model (ER-Model)

(*b*) Object Oriented Model

(*c*) Semantic Data Model

(*d*) Functional Data Model.

The models are discussed in the coming sections.

(*iii*) Physical Data Models : These models provide the concepts that describes the details of how the data is stored in the computer along with their record structures, access paths and ordering. Only specialized or professional users can use these models. These data models can be divided into two types:

The rows are unordered and unnamed. A database consists of one or more tables plus a catalogue describing the database.

The relational model consists of three components:

1. A structural component—A set of tables (also called relations) and set of domains that defines the way data can be represented.
2. A set of rules for maintaining the integrity of the database.
3. A manipulative component consisting of a set of high-level operations which act upon and produce whole tables.

In the relational model the data is represented in the form of tables which is used interchangeably with the word **Relation.** Each table consists of rows also knowns as **tuples** (A tuple represents a collection of information about an item, *e.g.*, student record) and column also known as **attributes.** (An attribute represents the characteristics of an item, *e.g.*, Student's Name and Phone No.). There are relationships existing between different tables. This model doesn't require any information that specifies how the data should be stored physically.

The major advantages of Relational Model are that it is structurally independent, improved conceptual simplicity adhoc query capability and powerful DBMS. The major disadvantages of relational model are substantial hardware and software overhead and facilitates poor design and implementation.

1.8.1.2 Object Based Data Models

Object Based Data Models are also known as conceptual models used for defining concepts including entries, attributes and relationships between them. These models are used in describing data at the logical and user view levels. These models allow the constraints to be specified on the data explicitly by the users.

An entity is a distinct object which has existence in real world. It will be implemented as a table in a database.

An attribute is the property of an entity, in other words, attribute is a single atomic unit of information that describes something about its entity. It will be implemented as a column or field in the database.

The associations or links between the various entities is known as relationships.

There are 4 types of object based data models. These are:

(*a*) Entity-relationship (E-R) Model

(*b*) Object-Oriented Model

(*c*) Semantic Data Model

(*d*) Functional Data Model

These are discussed as follows:

(*a*) Entity-Relationship (E-R) Model : The E-R model is a high level conceptual data model developed by Chen in 1976 to facilitate database design. The E-R model is the generalization of earlier available commercial model like the hierarchical and network model. It also allows the representation of the various constraints as well as their relationships.

The relationship between entity sets is represented by a name. E-R relationship is of 1 : 1, 1 : N or N : N type which tells the mapping from one entity set to another.

E-R model is shown diagrammatically using entity-relationship (E-R) diagrams which represents the elements of the conceptual model that show the meanings and relationships

between those elements independent of any particular DBMS. The various features of E-R model are:

(*i*) E-R Model can be easily converted into relations (tables).

(*ii*) E-R Model is used for purpose of good database design by database developer.

(*iii*) It is helpful as a problem decomposition tool as it shows entities and the relationship between those entities.

(*iv*) It is an iterative process.

(*v*) It is very simple and easy to understand by various types of users.

The major advantages of E-R model are that it is conceptually simple, have vishal representation, an effective communication tool and can be integrated with the relational data model.

The major disadvantages of E-R model are that there are limited constraint representation, limited relationship representation, no data manipulation language and loss of information content.

(*b*) Object-Oriented Data Model : Object-oriented data model is a logical data model that captures the semantics of objects supported in an object-oriented programming. It is based on collection of objects, attributes and relationships which together form the static properties. It also consists of the integrity rules over objects and dynamic properties such as operations or rules defining new database states.

An **object** is a collection of data and methods. When different objects of same type are grouped together they form a **class.** This model is used basically for multimedia applications as well as data with complex relationships. The object model is represented graphically with object diagrams containing object classes. Classes are arranged into hierarchies sharing common structure and behaviour and are associated with other classes.

Advantages of Object-Oriented Data Models

The various advantages of object-oriented data model are as follows:

(*i*) Capability to handle various data types : The object-oriented databases has the capability to store various types of data such as text, video pictures, voices etc.

(*ii*) Improved data access : Object oriented data models represent relationships explicitly. This improves the data access.

(*iii*) Improved productivity : Object-oriented data models provide various features such as inheritance, polymorphism and dynamic binding that allow the users to compose objects. These features increase the productivity of the database developer.

(*iv*) Integrated application development system : Object-oriented data model is capable of combining object-oriented programming with database technology which provides an integrated application development system.

Disadvantages of Object-Oriented Data Models

The various disadvantages of object-oriented data models are as follows:

(*i*) Not suitable for all applications : Object-oriented data models are used where there is a need to manage complex relationships among data objects. They are generally

suited for applications such as e-commerce, engineering and science etc. and not for all applications.

(*ii*) **No precise definition :** It is difficult to define what constitutes an object-oriented DBMS since the name has been applicable to wide variety of products.

(*iii*) **Difficult to maintain :** The definition of object is required to be changed periodically and migration of existing databases to confirm to the new object definition. It creates problems when changing object definitions and migrating databases.

(*c*) **Semantic Data Models :** These models are used to express greater interdependencies among entities of interest. These interdependencies enable the models to represent the semantic of the data in the database. This class of data models are influenced by the work done by artificial intelligence researchers. Semantic data models are developed to organize and represent knowledge but not data. This type of data models are able to express greater interdependencies among entities of interest. Mainframe database are increasingly adopting semantic data models. Also, its growth usage is seen in PC's. In coming times database management systems will be partially or fully intelligent.

(*d*) **Functional Data Model :** The functional data model describes those aspects of a system concerned with transformation of values-functions, mappings, constraints and functional dependencies. The functional data model describes the computations within a system. It shows how output value in computation are derived from input values without regard for the order in which the values are computed. It also includes constraints among values. It consists of multiple data flow diagrams. Data flow diagrams show the dependencies between values and computation of output values from input values and functions, without regard for when the functions are executed. Traditional computing concepts such as expression trees are examples of functional models.

1.8.2 Comparison of Various Data Models

The most commonly used data models are compared on the basis of various properties. The comparison table is given below.

Property	Hierarchical	Network	Relational	E-R Diagram	Object-oriented
1. Data element organization	Files, records	Files, records	Tables/tuples	Objects, entity sets	Objects
2. Identity	Record based	Record based	Value based	Value based	Record based
3. Data Independence	Yes	Yes	Yes	Yes	Yes
4. Relationship Organization	Logical proximity in a linearised tree.	Intersecting Networks	Identifiers of rows in one table are embedded as attribute values in another table.	Relational extenders that support specialized applications.	Logical containment
5. Access Language	Procedural	Procedural	Non-procedural	Non-procedural	Procedural
6. Structural Independence	No	No	Yes	Yes	Yes

1.8.3 Which Data Models to Use?

So far we have discussed a large number of data models. Data models are essential as they provide access techniques and data structure for defining a DBMS. In other words, a data model describe the logical Organization of data along with operations that manipulate the data.

We have large number of data models, the one which is best for the Organization depends upon the following factors:

- Is the database too small or too big.
- What are the costs involved.
- The volume of daily transactions that will be done.
- The estimated number of queries that will be made from the database by the organization to enquire about the data.
- The data requirements of the organization using it.

From the available record based data models, the relational data model is most commonly used model by most of the organizations because of the following reasons:

1. It increases the productivity of application programmers in designing the database. Whenever changes are made to the database there is no need of changing the application programs because of separation of logical level from conceptual level.
2. It is useful for representing most of the real world objects and relationships between them.
3. It provides very powerful search, selection and maintenance of data.
4. It hides the physical level details from the end users so end users are not bothered by physical storage.
5. It provides data integrity and security so that data is not accessed by unauthorized users and data is always accurate.
6. It provides adhoc query capability.

Some of the common DBMS using Relational model are MS-Access, Informix, Ingres, Oracle etc.

The hierarchical data model is used in those organizations which use databases consisting of large number of one to many relationships. Because of the restriction to one to many relationships, complexity of tree structure diagrams, lack of declarative querying facilities the hierarchical model lost its importance.

The network data model is used in those organizations which use databases consisting of large number of many to many relationships, but due to its complex nature it is also not preferred.

Most of the DBMS use object oriented data modelling techniques which are used by large number of organizations. For example—Latest versions of oracle are object relational hybrids because they support both relational and Object Oriented features.

1.9 TYPES OF DATABASE SYSTEMS

The database systems can be classified into three categories *i.e.,*

(*i*) According to the number of users

(*ii*) According to the type of use

(*iii*) According to database site locations

The various types of database systems are as follows:

1.9.1 According to the Number of Users

According to the number of users, the database systems can be further subdivided into two categories, namely:

(*a*) Single-user database systems

(*b*) Multiuser database systems.

(*a*) Single-user database systems : In a single user database system, the database reside on a PC–on the hard disk. All the applications run on the same PC and directly access the database. In single user database systems, the application is the DBMS. A single user accesses the applications and the business rules are enforced in the applications running on PC. A single user database system is shown in Figure 1.9. The example is DBASE files on a PC.

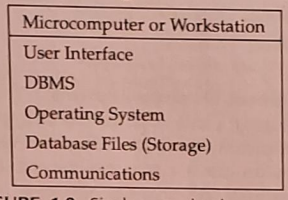

FIGURE 1.9. Single user database system.

(*b*) Multiuser database systems : In a multiuser database system, many PC's are connected through a Local Area Network (LAN) and a file server stores a copy of the database files. Each PC on the LAN is given a volume name on the file server. Applications run on each PC that is connected to the LAN and access the same set of files on the file server. The application is the DBMS and each user runs a copy of the same application and accesses the same files. The applications must handle the concurrency control and the business rules are enforced in the application. The example is MS-Access or Oracle files on a file server. A multiuser database system is shown in Figure 1.10.

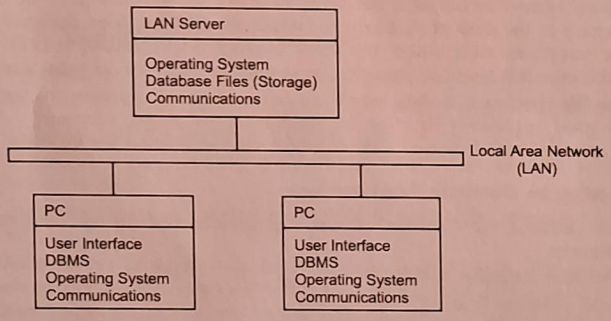

FIGURE 1.10. Multiuser database system.

Advantages of Multiuser Database System

There are many advantages of multiuser database system. Some of them are as follows:

(*i*) Ability to share data among various users.

(*ii*) Cost of storage is now divided among various users.

(*iii*) Low cost since most components are now commodity items.

Disadvantages of Multiuser Database System

The major disadvantage of the multiuser database system is that it has a limited data sharing ability *i.e.*, only a few users can share the data at most.

1.9.2 According to the Type of Use

According to the type of use, the database systems can be further subdivided into three categories, namely:

(*a*) Production or Transactional Database Systems

(*b*) Decision Support Database Systems

(*c*) Data Warehouses.

(*a*) Production or Transactional Database Systems : The production database systems are used for management of supply chain and for tracking production of items in factories, inventories of items in warehouses/stores and orders for items. The transactional database systems are used for purchases on credit cards and generation of monthly statements. They are also used in Banks for customer information, accounts, loans and banking transactions.

(*b*) Decision Support Database Systems : Decision support database systems are interactive, computer-based systems that aid users in judgement and choice activities. They provide data storage and retrieval but enhance the traditional information access and retrieval functions with support for model building and model based reasoning. They support framing, modelling and problem solving. Typical application areas of DSS's are management and planning in business, health care, military and any area in which management will encounter complex decision situations. DSS's are generally used for strategic and tactical decisions faced by upper level management *i.e.*, decisions with a reasonably low frequency and high potential consequences.

A database system serves as a databank for the DSS. It stores large quantities of data that are relevant to the class of problems for which the DSS has been designed and provides logical data structures with which the users interact. The database system is capable of informing the user the types of data that are available and how to gain access to them.

(*c*) Data Warehouses : A data warehouse is a relational database management system (RDMS) designed specifically to meet the transaction processing systems. It can be loosely defined as any centralised data repository which can be queried for business benefit.

1.9.3 According to Database Site Locations

According to database site locations, database systems can be further subdivided into four categories namely:

(*a*) Centralized database systems

(*b*) Parallel database systems

(*c*) Distributed database systems

(*d*) Client/Server database systems.

(a) Centralized database systems : The centralised database system consists of a single processor together with its associated data storage devices and other peripherals. Database files resides on a personal computer (small enterprise) or on a mainframe computer (large enterprise). The applications are run on the same PC or mainframe computer. Multiple users access the applications through simple terminals that have no processing power of their own. The user interface is text-mode screens and the business rules are enforced in the applications running on the mainframe or PC. The example of centralized database system is DB2 database and Cobol application programs running on IBM 390.

The centralized database system is shown in Figure 1.11.

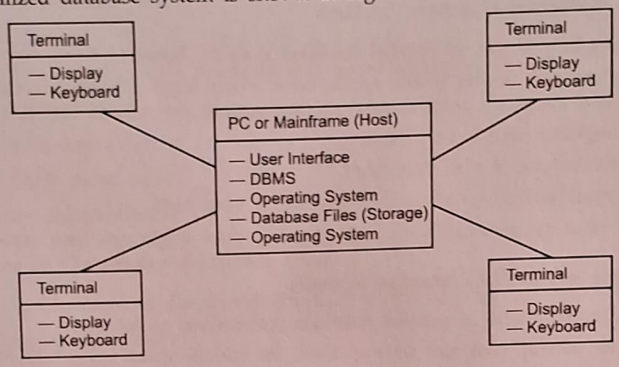

FIGURE 1.11. Centralized database system.

Advantages of Centralized Database System

There are many advantages of centralized database system some of them are as follows:

(*i*) The control over applications and security is excellent.

(*ii*) The incremental cost per user is very low.

(*iii*) The centralized systems are highly reliable due to proven mainframe technology.

(*iv*) Many functions such as query, backup, update etc., are easier to accomplish.

Disadvantages of Centralized Database System

The various disadvantages of centralized database system are as follows:

(*i*) The users are not able to effectively manipulate data outside of standard applications.

(*ii*) The system is not able to effectively serve advance user interfaces.

(*iii*) The failure of central computer blocks every user from using the system until the system comes back.

(*iv*) The communication costs from the terminal to the central computer is a matter of concern.

(b) Parallel database systems : A parallel database system can be defined as a database system implemented on a tightly coupled multiprocessor or on a loosely coupled multiprocessor. Parallel database systems link multiple smaller machines to achieve the same throughput as a single larger machine, often with greater scalability and reliability than single processor

database system. Parallel database systems are used in the applications that have to query extremely large databases or have to process an extremely large number of transactions per second. There are three main architectures for parallel database system. These are

(*i*) Shared memory architecture

(*ii*) Shared disk architecture

(*iii*) Shared nothing architecture.

More about these types is discussed in Chapter 12.

Advantages of Parallel Database Systems

There are many advantages of parallel database systems. Some of these are as follows:

(*i*) These are very useful in the applications where large databases have to be queried or where extremely large number of transactions per second has to be processed.

(*ii*) The response time is very high.

(*iii*) The throughput is also very high.

(*iv*) The input/output speeds and processing is very high.

(*v*) They have greater scalability and reliability than single processor system.

Disadvantages of Parallel Database Systems

The various disadvantages of parallel database systems are as follows:

(*i*) Due to start-up cost and start-up time, the overall speed up is adversely affected.

(*ii*) Due to processes executed in parallel, sharing the resources, a slow down may result offer each new process as it competes with existing processes for the resources.

(c) Distributed database systems : A distributed database system is a database system, in which, the data is spread across a variety of different databases. These are managed by a variety of DBMS's that are running on various types of machines having different operating systems. These machines are widely spread and are connected through the communication networks. Each machine can have is own data and applications, and can access data stored on other machines. Thus, each machine acts as a server as well as client.

Thus, distributed database system is a combination of logically interrelated databases distributed over a computer network and the distributed database management system (DDBMS). A distributed database system can be homogeneous or heterogeneous. A distributed database system is shown in Figure 1.12.

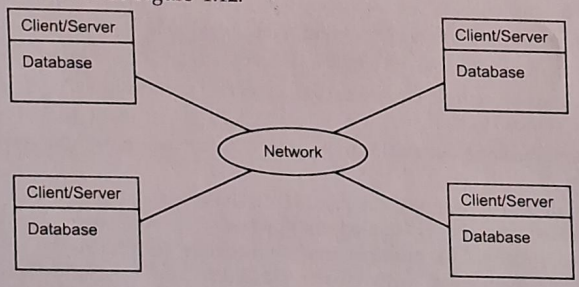

FIGURE 1.12. Distributed database system.

Advantages of Distributed Database Systems

The various advantages of distributed database systems are as follows:

1. Improved sharing ability
2. Local autonomy
3. Availability
4. Reliability
5. Improved performance
6. Easier expansion
7. Reduced communications overhead and better response time
8. More economical
9. Direct user interaction
10. No a single point failure
11. Processor independence.

Disadvantages of Distributed Database Systems

The various disadvantages of distributed database systems are as follows:

1. Architectural complexity
2. Lack of standards
3. Lack of professional support
4. Data integrity problems
5. Problem of security
6. High cost
7. Complex database design.

(*d*) Client/Server Database System : With the development of technology, hardware cost become cheaper and cheaper and more personal computers are used. There was a change and enterprises started use of client-server technology instead of centralized system. In client-server technology, there is a server which acts as a whole data base management system and some clients or personal computers which are connected with server through a network interface. The complete architecture is shown in Figure 1.13.

Components of Client-Server Architecture

There are three major components of client server architecture:

1. Server
2. Client
3. Network interface

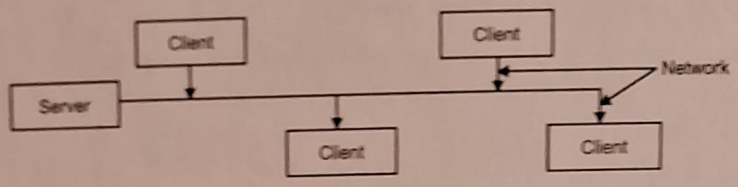

FIGURE 1.13. Client server database system.

1. Server : Server is DBMS itself. It consists of DBMS and supports all basic DBMS functions. Server components of DBMS are installed at server. It acts as monitor of all of its clients. It distributes work-load to other computers. Clients must obey their servers.

Functions of Server : The server performs various functions, which are as follows.

1. It supports all basic DBMS functions.
2. Monitor all his clients.
3. Distribute work-load over clients.
4. Solve problems which are not solved by clients.
5. Maintain security and privacy.
6. Avoiding unauthorized access of data.

2. Clients : Client machine is a personal computer or workstation which provide services to both server and users. It must obey his server. Client components of DBMS are installed at client site. Clients are taking instructions from server and help them by taking their load. When any user want to execute a query on client, the client first take data from server then execute the query on his own hardware and returns the result to the server. As a result, server is free to do more complex applications.

3. Network Interface : Clients are connected to server by network interface. It is useful in connecting the server interface with user interface so that server can run his applications over his clients.

In the client server architecture, there are more than one server. Sometimes, a server is used as Database Server, other as Application Server, other as Backup Server etc.

Advantages of Client-Server Database System

1. It increase the overall performance of DBMS.
2. Load can be distributed among clients.
3. It provides better user interface.
4. It is used to develop highly complex applications.
5. Clients with different operating systems can be connected with each other.
6. Single copy of DBMS is shared.
7. It reduces cost.

Disadvantages of Client-Server Database System

1. Network is error prone.
2. It is a kind of centralized system. If server is crashed or failed, there is loss of data.

3. Recovery is typical and additional burden on DBMS server to handle concurrency control.
4. Programming cost is high.
5. The implementation is more complex since one needs to deal with the middle ware and the network.

1.10 COMPARISON BETWEEN CLIENT/SERVER AND DISTRIBUTED DATABASE SYSTEM

Client/Server Database System	Distributed Database System
1. In this, different platforms are often difficult to manage.	1. In this, different platforms can be managed easily.
2. Here, application is usually distributed across clients.	2. Here, application is distributed across sites.
3. In this database system, whole system comes to a halt if server crashes.	3. Here, failure of one site doesn't bring the entire system down as system may be able to reroute the one site's request to another site.
4. Maintenance cost is low.	4. Maintenance cost is much higher.
5. In this system, access to data can be easily controlled.	5. In DDS not only does the access to replicate the data has to be controlled at multiple locations but also the network has to be made secure.
6. In this, new sites can not be added easily.	6. In this, new sites can be added with little or no problem.
7. Speed of database access is good.	7. Speed of database access is much better.