Data
It is a collection of facts – unorganized, but able to be organized into useful information. Data are raw facts and figures in isolation. These isolated facts and figures convey meanings but generally are not useful by them. For example: Aryan, lives, boy, chabahil, etc.

Information
It is the product or result of processing data into a meaningful form. In other words, we can say that information is a data which is placed in a meaningful form to the users. For example: "Aryan is a boy and he lives in chabahil".

Database
A collection of data designed to be used by different people is called a database. A database is a collection of interrelated data that is organized so that its contents can easily be accessed, managed and updated. In other words, a database is an organized collection of data. The data are stored in such a fashion that they are independent of the programs of people using the data.

Database Management System (DBMS)
A database management system (DBMS) is a software tool that allows multiple users to store, access, and process data or facts into useful information. In other words, a database management system (DBMS) is a system or software designed to manage a database, and run operations on the data requested by numerous clients.Some popular DBMS includes dBase, Visual FoxPro, Oracle, DB2, Informix, MS SQL Server, MySQL, and Microsoft Access

Objective of DBMS
Some of the objectives of DBMS are as follows:

- Provides relevant data to users
- Easy access to data and information
- Provides quick response to the user request for data
- Eliminates the duplicate data
- Allows multiple users to access and share data
- Allows the scalability of database
- Protects data from unauthorized access
- Provides an abstract view of data that hides details of data from users
- Creates relationships between items of data

ADVANTAGES OF DBMS

1. Makes easy to add new data.
2. Makes easy to modify the database.
3. Makes easy to delete existing data
4. Organized the data in proper sequence.
5. It reduces the data redundancy to a large extent.
6. It can control data inconsistency to a large extent.
7. Maintains data integrity i.e. accurate, consistent and up-to-date data
8. Make easy to access the data for the authorized user.
9. Allow multiple users to be active at one time (i.e. data in the database may be shared among several users)
10. Protecting data against unauthorized access.
11. Allow for growth in the database system.

DISADVANTAGES OF DBMS

1. Complex to understand and implement
2. Costly
3. Too many rules
4. Fast changing technology
5. Change of losing the data
6. Chance of data leakage and hacking
7. Unavailability of trained manpower

DBA (Database Administrator)

DBA is a special person (super users) who controls both the data and the programs that access those data, i.e. controls overall system. The DBA is responsible for ensuring that the data in the database meets the information needs of the organization. The DBA must have a sound knowledge of the structure of the database & of the DBMS. The DBA must also be thoroughly conversant with the organization, its system & the information needs of the managers.

A DBA needs the following:

a. Knowledge of the operating system in which database server is running.
b. Knowledge of SQL
c. Sound knowledge in database design
d. General understanding of network architectures
e. Knowledge about the database server.

Roles/Functions/Responsibilities of database administrator (DBA)

The DBA is responsible for ensuring that:

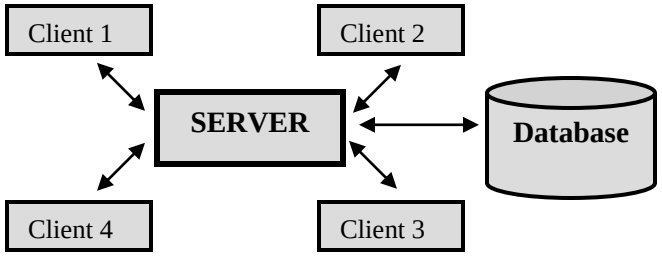
a) The data in the database meets the information needs of the organization
b) The facilities for retrieving data and for structuring reports are appropriate to the needs of the organization.
c) The DBA is responsible for the data dictionary (data about data or meta- data, i.e. define the structure of data) and manuals for users describing the facilities the database offers and how to make use of these facilities.
d) Another function of DBA is to supervise the modification (insert, delete and update) of data.
e) The DBA is also responsible for security of database and requirements of privacy.
f) The DBA is also responsible for database integrity maintenance (changes made to the database, do not result in a loss of data).
g) The DBA is also responsible for periodic appraisal of the data held in the database to ensure that is complete, accurate and not duplicated.

Database System Architecture

The architecture of a database system is greatly influenced by the underlying computer system on which the database system runs.

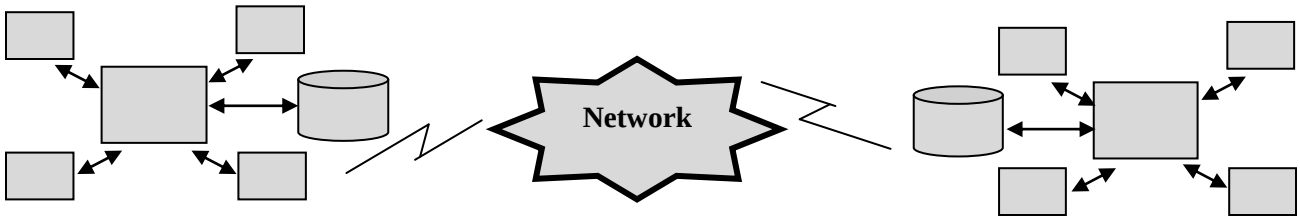
1. Centralized Database Systems

Centralized database systems are those that run on a single computer system and do not interact with other computer systems i.e. these database systems are used only in single user system (personal computers). A typical single user system is a desktop unit used by a single person, has only one CPU, one or two hard disks and has an operating system that may support only one user. Centralized database works on a client-server basis. The structure of a centralized databases system is shown in figure.



2. Distributed Database System

A distributed database system is a collection of databases that shares a common schema and coordinates to access no local data, i.e., in a distributed database system, the database is stored on several computers. The computers in a distributed system communicate with one another through various communication media, such as networks or telephone lines. They do not share main memory or disks. The computer in a distributed system is called site or nodes. The general structure of distributed system is shown in the figure below.



DIFFERENCE BETWEEN CENTRALIZED AND DISTRIBUTED DATABASE SYSTEM

Centralized Database System	Distributed Database System
a. Data reside in a single location	a. The data reside in several locations
b. Files are kept on the basis of location of disk drives and names.	b. Files are kept on the basis of names rather than location.
c. It does not contain several sites or nodes (i.e., does not contain several computers)	c. It contains several computers (sites) and communicates with one another through communication media
d. Once the database system fails, it doesn't operate	d. If one site fails, the remaining sites may able to continue operating
e. No risk of data lost	e. High risk of data theft, hacking, etc.
f. Suitable for a single organization like school, college, industries, etc. of a location.	f. Suitable for a large organization spread in different geographical locations.
g. Low cost	g. High cost

Database Models or Data Models

A data or database model is the organizing principle that specifies particular mechanisms for data storage and retrieval. The model explains, in terms of services available to an interfacing application, how to access a data element when other related data elements are known. The primary difference between the different database models lie in the methods of expressing the relationships and constraints among the data elements. Database Models may be classified as follows:

1. The Hierarchical database model
2. The Network database model
3. The Relational database model
4. Entity-Relationship model

1. Hierarchical database model: this is one of the oldest type of database models. In this model data is represented in the form of records. Each record has multiple fields. All records are arranged in database as tree like structure. The relationship between the records is called parent child relationship in which any child record relates to only a single parent type record.

Advantages of Hierarchical database model

- **a.** Easiest model of database
- **b.** Searching is fast if root node is known
- **c.** Handles 'one-to-one' relation very efficiently

Disadvantages of Hierarchical database model

- **a.** Old and outdated database model
- **b.** Non-Flexible database model
- **c.** Can't handle 'many-to-many' relationship
- **d.** Increases redundancy

2. Network database model: it replaced hierarchical network database model due to some limitations on the model. Suppose, if an employee relates to two departments, then the hierarchical database model cannot able to arrange records in proper place. So network, database model was emerged to arranged non-hierarchical database. The structure of database is more like graph rather than tree structure.

Advantages of Network database model

- **a.** Accepts 'many-to-many' relationship
- **b.** More flexible model
- **c.** Reduces redundancy
- **d.** Searching is faster due to multidirectional pointers

Disadvantages of Network database model

- **a.** Complex type of database model
- **b.** Less secured than the hierarchical database model
- **c.** Needs large storage

3. Relational database model: in this model, the data is organized into tables which contain multiple rows and columns. These tables are called relations. A row in a table represents a relationship among a set of values. Since a table is a collection of such relationships, it is generally referred to the mathematical term relation, from which the relational database model derives its name.

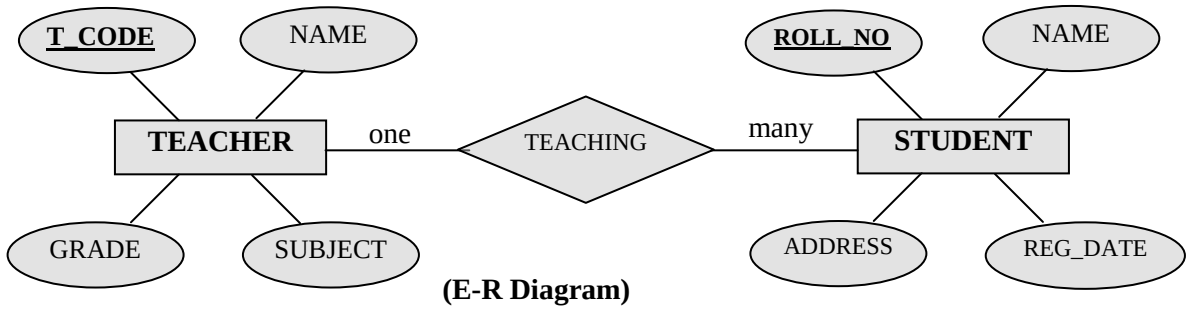
Advantages of Relational database model

- **a.** Accepts 'many-to-many' relationship
- **b.** More flexible model
- **c.** Reduces redundancy
- **d.** Normalization of database is possible

Disadvantages of Relational database model

- **a.** Complex type of database model
- **b.** Makes database non-user friendly due to many tables
- **c.** Hard to manage all tables
- **d.** Too many rules

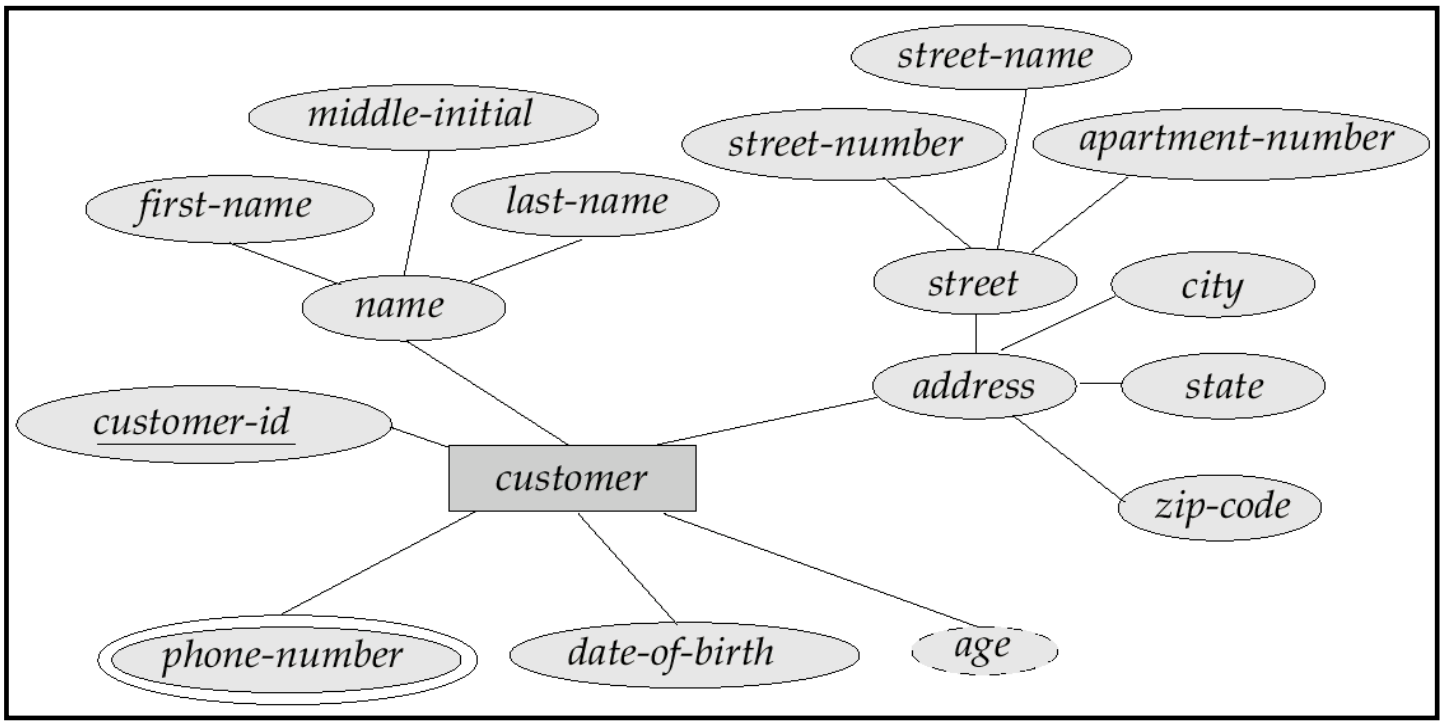
4. Entity-Relationship database model: this model is based on perception of a real world that contains a collection of basic objects, called entities and of relationship among these objects and characteristics of an entity. It shows relationship between different entities.
   The E-R (entity-relationship) data model views the real world as a set of basic objects (entities) and relationships among these objects. This represents the overall logical structure of the DB.



(E-R Diagram)

Some of the notations used in E-R Diagram are:

- Rectangles represent entity
- Diamonds represent relationship
- Lines link attributes to entity sets and entity sets to relationship sets
- Ellipses represent attributes
- Double ellipses represent multi-valued attributes
- Dashed ellipses denote derived attributes
- Underline indicates primary key attributes



Relationship

A relationship is an association among entities. The relationship may represent an event that links the entities or may represent a logical affinity that exists between the entities. For example, there is a relationship between student and instructors. This relationship represents the fact that an instructor teaches several students, and a student is taught by several instructors. This relationship can be named '**teaches**'.

Types of Relationships

There are three types of relationships:

1. One-to-one
2. One-to-many or many-to-one
3. Many-to-many

1. One-to-one
   The relationship in which one entity is associated with another entity is known as one to one relationship.

For Example: Driver drives a car.

2. Many-to-one
   The relationship in which many entities are associated with one entity is known as many to one relationship.

For Example: Many Students can admit in one stream.

3. Many-to-many
   The relationship in which many entities are associated with many entities is known as many to many relationship.

For Example: A STUDENT can take many COURSE and many students can register for a given course.

Normalization

It is a step-by-step decomposition of a complex relation into simple relations. It is the process of organizing data in a database to reduce the redundancies. It helps to simplify the structure of tables. A fully normalized record consists of:

- A primary key that identifies an entity
- A set of attributes that describe the entity

Objective of normalization

- To reduce redundancy.
- To avoid the inability to represent certain information, i.e., to avoid insertion anomaly.
- To avoid the loss of information i.e. to avoid deletion anomaly.
- To avoid update anomaly.

Types of Normalization

There are three types of Normalization. They are

a. 1 NF (1st Normal Form)
b. 2 NF (2nd Normal Form)
c. 3 NF (3rd Normal Form)

a. 1 NF (1st Normal Form)

In 1st NF

- The table cells must be of a single value.
- Eliminate repeating groups in individual tables.
- Create a separate table for each set of related data.
- Identify each set of related data with a primary key.

Definition: A relation is said to be in 1NF if it contains no repeating groups.

EXAMPLE OF 1 NF:

ECODE	DEPT	PROJCODE	HOURS
E101	Systems	P27	90
		P51	101
		P20	60
E305	Sales	P27	109
E508	Admin	P51	NULL
		P27	72

Table 1: Un-normalized Data

ECODE	DEPT	PROJCODE	HOURS
E101	Systems	P27	90
E101	Systems	P51	101
E101	Systems	P20	60
E305	Sales	P27	109
E508	Admin	P51	NULL
E508	Admin	P27	72

Table 2: A table in 1NF

b. 2 NF (2nd Normal Form)

In 2nd NF

- Remove Partial Dependencies.
- Keep Full Dependency
- Create a separate table with the functionally dependent data and the part of the key on which it depends.

Definition: A relation is said to be in 2NF if & only if it is in 1NF & all attributes dependent upon the whole key, and not just part of the key.

EXAMPLE OF 2 NF:

ECODE	DEPT	PROJCODE	HOURS
E101	Systems	P27	90
E101	Systems	P51	101
E101	Systems	P20	60
E305	Sales	P27	109
E508	Admin	P51	NULL
E508	Admin	P27	72

Table 2: A table in 1NF

ECODE	PROJCODE	HOURS
E101	P27	90
E101	P51	101
E101	P20	60
E305	P27	109
E508	P51	NULL
E508	P27	72

ECODE	DEPT
E101	Systems
E305	Sales
E508	Admin

Table 3: A table in 2 NF

c. 3 NF (3rd Normal Form)

In 3rd NF

- Remove transitive dependencies.
- Create a separate table containing the attribute and the fields that are functionally dependent on it.

Definition: A relation is said to be in 3NF if & only if it is in 2NF & every non-key attributes is functionally dependent on just the primary key.

EXAMPLE OF 3 NF:

ECODE	DEPT	DEPT_HEAD
E101	Systems	E901
E305	Sales	E906
E402	Sales	E906
E508	Admin	E908
E607	Finance	E909
E608	Finance	E909

ECODE	DEPT
E101	Systems
E305	Sales
E402	Sales
E508	Admin
E607	Finance
E608	Finance

Table 4: A table in 3NF

DEPT	DEPT_HEAD
Systems	E901
Sales	E906
Admin	E908
Finance	E909

Database Languages

Database Languages are the set of statements that are used to define and manipulate a database. It provides the tools to implement and manipulate a database. It can be used to read, store and update the data in the database. Some languages which are commonly used are as follows:

1. **Data Description Language (DDL)**
   It is used to define database structure or pattern. It is used to create schema, tables, indexes, constraints, etc. in the database. Using the DDL statements, user can create the skeleton of the database. It is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc. Using DDL statements user can **create, alter, drop or rename** the information in the database.

2. **Data Manipulation Language (DML)**
   Data Manipulation Language (DML) is a language which enables users to access and manipulate data. It handles user requests. Using DML statements user can **retrieve**, **insert**, **delete** or **modify** the information in the database. There are two types of DML. They are Procedural DML and Non Procedural DML.

3. **Structure Query Language (SQL)**
   Structured Query Language is used to build and manipulate relational databases. SQL is a non-procedural language. Most RDBMS follows the client-server architecture. The data is stored and managed by the server and user request to the server. Information is derived when data is processed. In order to retrieve information in RDBMS, queries are used.

4. **Unified Modeling Language (UML)**
   Unified Modeling Language (UML) is designed for software engineering of large systems using object-oriented programming (OOP) languages. It is the tool for communicating with the client in terms that are used in the enterprise. This language is used for object-oriented database design.

Data Redundancy

Data redundancy is a condition created within a database or data storage technology in which the same piece of data is held in two separate places. This can mean two different fields within a single database, or two different spots in multiple software environments or platforms. Whenever data is repeated, it basically constitutes data redundancy.

Data Integrity

Data integrity is the accuracy and consistency or validity of data over its lifecycle. It is a critical aspect to the design, implementation, and usage of any system that stores, processes, or retrieves data. It is maintained by a collection of processes, rules, and standards implemented during the design phase. When the integrity of data is secure, the information stored in a database will remain complete, accurate, and reliable no matter how long it's stored or how often it's accessed.

Data Security

Data security refers to the process of protecting data from unauthorized access and data corruption throughout its lifecycle. Data security includes data encryption, hashing, tokenization, and key management practices that protect data across all applications and platforms. T he core elements of data security are confidentiality, integrity, and availability.
Types of data security controls include Authentication, Access control, Backups & recovery, Encryption, Data masking, Tokenization and Deletions & erasure.