Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information.

Mostly data represents recordable facts. Data aids in producing information, which is based on facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks.

A **database management system** stores data in such a way that it becomes easier to retrieve, manipulate, and produce information.

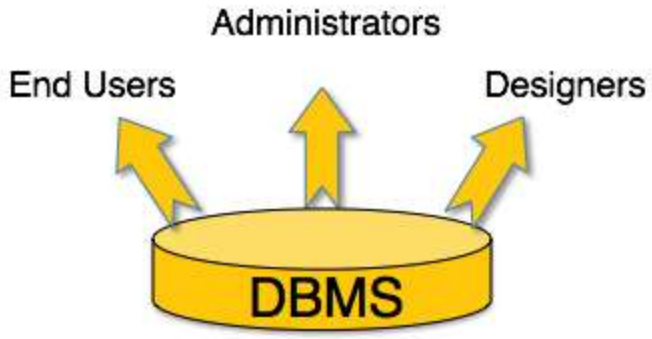
Characteristics

Traditionally, data was organized in file formats. DBMS was a new concept then, and all the research was done to make it overcome the deficiencies in traditional style of data management. A modern DBMS has the following characteristics −

- **Real-world entity** − A modern DBMS is more realistic and uses real-world entities to design its architecture. It uses the behavior and attributes too. For example, a school database may use students as an entity and their age as an attribute.
- **Relation-based tables** − DBMS allows entities and relations among them to form tables. A user can understand the architecture of a database just by looking at the table names.
- **Isolation of data and application** − A database system is entirely different than its data. A database is an active entity, whereas data is said to be passive, on which the database works and organizes. DBMS also stores metadata, which is data about data, to ease its own process.
- **Less redundancy** − DBMS follows the rules of normalization, which splits a relation when any of its attributes is having redundancy in values. Normalization is a mathematically rich and scientific process that reduces data redundancy.
- **Consistency** − Consistency is a state where every relation in a database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state. A DBMS can provide greater consistency as compared to earlier forms of data storing applications like file-processing systems.
- **Query Language** − DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many and as different filtering options as required to retrieve a set of data. Traditionally it was not possible where file-processing system was used.

Users

A typical DBMS has users with different rights and permissions who use it for different purposes. Some users retrieve data and some back it up. The users of a DBMS can be broadly categorized as follows −



- **Administrators** − Administrators maintain the DBMS and are responsible for administrating the database. They are responsible to look after its usage and by whom it should be used. They create access profiles for users and apply limitations to maintain isolation and force security. Administrators also look after DBMS resources like system license, required tools, and other software and hardware related maintenance.
- **Designers** − Designers are the group of people who actually work on the designing part of the database. They keep a close watch on what data should be kept and in what format. They identify and design the whole set of entities, relations, constraints, and views.
- **End Users** − End users are those who actually reap the benefits of having a DBMS. End users can range from simple viewers who pay attention to the logs or market rates to sophisticated users such as business analysts.

Two types of database structure

Databases typically have one of two basic forms:

- single-file or flat file database
- multi-file relational or structured database

A **flat file database** stores data in a plain text file, with each line of text typically holding one record. Delimiters such as commas or tabs separate fields. A flat file database uses a simple structure and, unlike a relational database, cannot contain multiple tables and relations. Read more about **flat file databases**.

A **relational database** contains multiple tables of data with rows and columns that relate to each other through special key fields. These databases are more flexible than flat file structures, and provide functionality for reading, creating, updating, and deleting data. Relational databases use Structured Query Language (SQL) - a standard user application that provides an easy programming interface for database interaction. Read more about **relational databases**.

Types of relationships in a database

Four types of relationships exist in relational database design:

- one to one - where one table record relates to another record in another table

- one to many - where one table record relates to multiple records in another table
- many to one - where more than one table record relates to another table record
- many to many - where multiple records relate to more than one record in another table

Tables Design:

User Registration

Column Name	Data Type	Allow Nulls
Name	varchar(50)	☐
Password	varchar(50)	☑
Type	varchar(50)	☑
Status	varchar(50)	☑

Customer

Column Name	Data Type	Allow Nulls
CustomerId	int	☐
Name	varchar(50)	☑
Address	varchar(MAX)	☑
Phone	varchar(50)	☑
Nic	varchar(50)	☑

Product

Column Name	Data Type	Allow Nulls
ProductId	int	☐
ProductName	varchar(50)	☑
ProductDescription	varchar(MAX)	☑
ProductPrice	int	☑
ProductManufacture	varchar(50)	☑

Booker

Column Name	Data Type	Allow Nulls
BookerId	int	☐
BookerDesignation	varchar(50)	☑
BookerName	varchar(50)	☑
BookerAddress	varchar(50)	☑
BookerPhone	varchar(50)	☑
Nic	varchar(50)	☑

Order

Column Name	Data Type	Allow Nulls
OrderId	int	☐
OrderDateTime	nchar(10)	☑
BookerId	int	☐
CustomerId	int	☐
OrderStatus	varchar(50)	☑

Order Detail

Column Name	Data Type	Allow Nulls
OrderDetailId	int	☐
OrderId	int	☐
ProductId	int	☐
Quantity	int	☑
Discount	int	☑
Price	int	☑

Delivery

Column Name	Data Type	Allow Nulls
DeliveryId	int	☐
OrderId	int	☐
DeliverDateTime	varchar(50)	☑

Delivery Detail

Column Name	Data Type	Allow Nulls
DetailId	int	☐
DeliveryId	int	☐
ProductId	int	☐
Quantity	int	☑
Price	int	☑