**Question No-01: What is Data?**

In general, **data** is any set of [characters](https://www.computerhope.com/jargon/c/charact.htm) that is gathered and translated for some purpose, usually analysis. If data is not put into context, it doesn't do anything to a human or computer.

There are multiple types of data. Some of the more common types of data include the following:

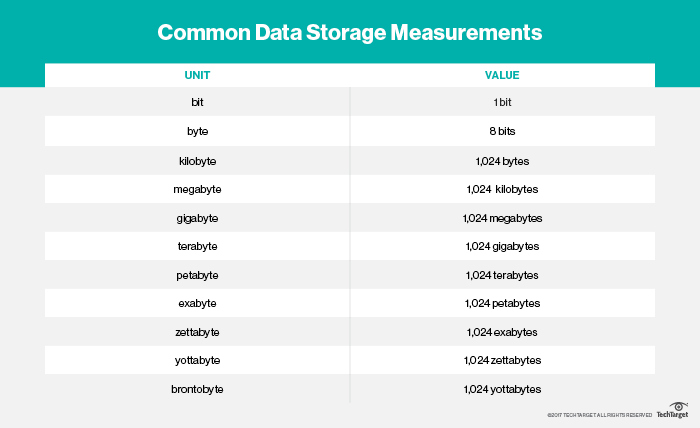
* Single character
* [Boolean](https://www.computerhope.com/jargon/b/boolean.htm) (true or false)
* Text ([string](https://www.computerhope.com/jargon/s/string.htm))
* Number ([integer](https://www.computerhope.com/jargon/i/integer.htm) or [floating-point](https://www.computerhope.com/jargon/f/floapoin.htm))
* [Picture](https://www.computerhope.com/jargon/p/picture.htm)
* [Sound](https://www.computerhope.com/jargon/s/sound.htm)
* [Video](https://www.computerhope.com/jargon/v/video.htm)

In a computer's storage, digital data is a sequence of [bits](https://www.computerhope.com/jargon/b/bit.htm) ([binary digits](https://www.computerhope.com/jargon/b/binary.htm)) that have the value one or zero. Data is processed by the [CPU](https://www.computerhope.com/jargon/c/cpu.htm), which uses [logical](https://www.computerhope.com/jargon/l/logic.htm) operations to produce new data ([output](https://www.computerhope.com/jargon/o/output.htm)) from source data ([input](https://www.computerhope.com/jargon/i/input.htm)).

**How data is stored:**

Computers represent data, including video, images, sounds and text, as binary values using patterns of just two numbers: 1 and 0. A [bit](https://www.techtarget.com/whatis/definition/bit-binary-digit) is the smallest unit of data, and represents just a single value. A byte is eight binary digits long. Storage and memory is measured in [megabytes](https://www.techtarget.com/searchstorage/definition/megabyte) and [gigabytes](https://www.techtarget.com/searchstorage/definition/gigabyte).

The units of data measurement continue to grow as the amount of data collected and stored grows. The relatively new term "[brontobyte](https://www.techtarget.com/searchstorage/definition/brontobyte)," for example, is [data storage](https://www.techtarget.com/searchstorage/definition/storage) that is equal to 10 to the 27th power of [bytes](https://www.techtarget.com/searchstorage/definition/byte).



**Question – 02:**

**What is Database?**

A database is a systematic collection of data. They support electronic storage and manipulation of data. Databases make data management easy.

Let us discuss a database example: An online telephone directory uses a database to store data of people, phone numbers, and other contact details. Your electricity service provider uses a database to manage billing, client-related issues, handle fault data, etc.

Let us also consider Facebook. It needs to store, manipulate, and present data related to members, their friends, member activities, messages, advertisements, and a lot more. We can provide a countless number of examples for the usage of databases.

## Types of Databases

Here are some popular types of databases.

* Distributed Database
* Relational Database
* Object-Oriented Database
* Centralized Database
* Open-Source Database
* Cloud Database
* Data Warehouse
* NoSql Database
* Graph Database
* OLTP Database
* Personal Database
* Multimodal Database
* Document/JSON Database
* Hierarchical Database

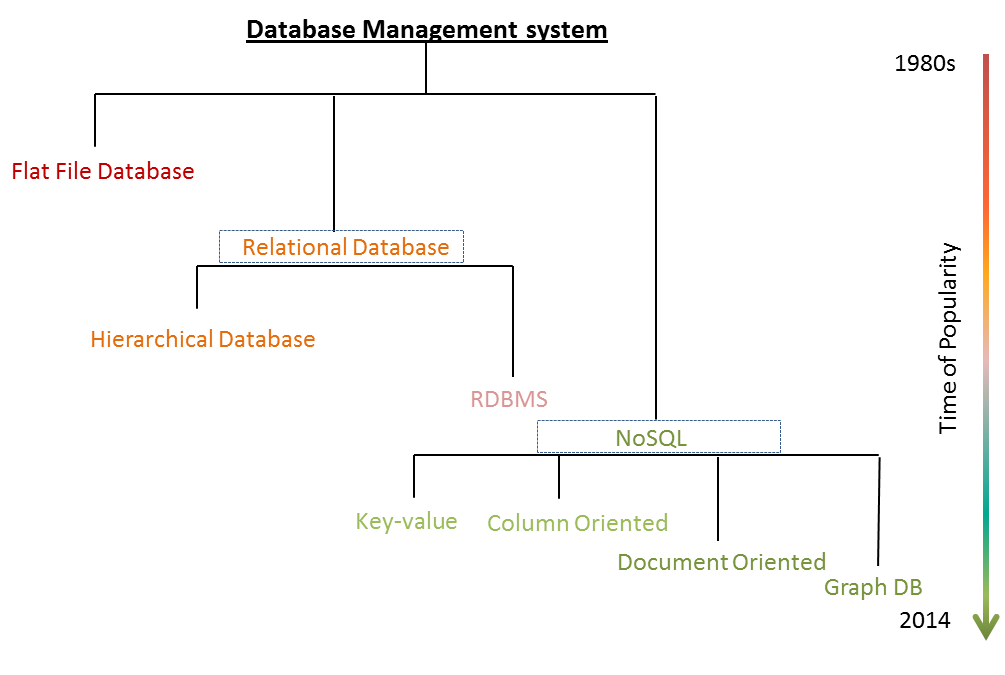
**Question-03:**

**Evolution of Database:**

A database management system is a set of programs that allows users to create and maintain a data base and interact with various users, including data base administrators, online users, application programmers, and naive users, as well as other users.

Hence we can say that basically database is a compilation of database files and each database file is further a collection of records.

**Database models:**

****

**1) Flat files (1960s – 1980s)**

**Implementation of a flat file database**

Flat file database is implemented in:

* Berkeley DB
* SQLite
* Mimesis
* TheIntegrationEngineer etc.

**2) Hierarchical database (1970s – 1990s)**

**3)** **Network database (1970s – 1990s)**

**Implementation of network database**

Network database is implemented in:

* Digital Equipment Corporation DBMS-10
* Digital Equipment Corporation DBMS-20
* RDM Embedded
* Turbo IMAGE
* Univac DMS-1100 etc.

**4)** **Relational database (1980s – present)**

Implementation of Relational Database:

* Oracle
* Microsoft
* IBM
* My SQL
* PostgreSQL
* SQLite

**5)** **Object-oriented database (1990s – present)**

**Question No-04:**

**NoSQL Database:**

**NoSQL database** technology stores information in JSON documents instead of columns and rows used by relational databases. To be clear, NoSQL stands for “not only SQL” rather than “no SQL” at all. This means a NoSQL JSON database can store and retrieve data using literally “no SQL.” Or you can combine the flexibility of JSON with the power of SQL for the best of both worlds. Consequently, NoSQL databases are built to be flexible, scalable, and capable of rapidly responding to the data management demands of modern businesses. The following defines the four most-popular types of NoSQL database:

* **Document databases** are primarily built for storing information as documents, including, but not limited to, JSON documents. These systems can also be used for storing XML documents, for example.
* **Key-value stores** group associated data in collections with records that are identified with unique keys for easy retrieval. Key-value stores have just enough structure to mirror the value of relational databases while still preserving the benefits of NoSQL.
* **Wide-column databases** use the tabular format of relational databases yet allow a wide variance in how data is named and formatted in each row, even in the same table. Like key-value stores, wide-column databases have some basic structure while also preserving a lot of flexibility
* **Graph databases** use graph structures to define the relationships between stored data points. Graph databases are useful for identifying patterns in unstructured and semi-structured information.

## Why use NoSQL?

 Customer experience has quickly become the most important competitive differentiator and ushered the business world into an era of monumental change. As part of this revolution, enterprises are interacting digitally – not only with their customers, but also with their employees, partners, vendors, and even their products – at an unprecedented scale. This interaction is powered by the internet and other 21st century technologies – and at the heart of the revolution are a company’s cloud, mobile, social media, big data, and IoT applications.

How are these applications different from legacy enterprise applications like ERP, HR, and financial accounting? Today’s web, mobile, and IoT applications share one or more (if not all) of the following characteristics. They need to:

 Support large numbers of concurrent users (tens of thousands, perhaps millions)

* Deliver highly responsive experiences to a globally distributed base of users
* Be always available – no downtime
* Handle semi- and unstructured data
* Rapidly adapt to changing requirements with frequent updates and new features

**Question No- 05:**

**Advantage of NoSQL:**

**What are the advantages of NoSQL?**

Like every other technology, NoSQL databases also offer some benefits and suffer from some limitations too.

In an era where relational databases are mainly used for data storage and retrieval, modern web technologies posed a major challenge in the form of unstructured data, high scale data, enormous concurrency etc.

Relational databases struggled especially to represent highly unstructured data and high scalability and thus came into being the NoSQL databases.

**Major advantages of NoSQL databases include:**

**(i) Flexible Data Model:**

NoSQL databases are highly flexible as they can store and combine any type of data, both structured and unstructured, unlike relational databases that can store data in a structured way only.

**(ii) Evolving Data Model :**

NoSQL databases allow you to dynamically update the schema to evolve with changing requirements while ensuring that it would cause no interruption or downtime to your application.

**(iii) Elastic Scalability:**

NoSQL databases can scale to accommodate any type of data growth while maintaining low cost.

**(iv) High Performance:**

NoSQL databases are built for great performance, measured in terms of both throughput (it is a measure of overall performance) and latency (it is the delay between request and actual response).

**(v) Open-source:**

NoSQL databases don’t require expensive licensing fees and can run on inexpensive hardware, rendering their deployment cost-effective.

**Question No- 06:**

**Disadvantage of NoSQL:**

**Major disadvantages of NoSQL databases are:**

**(i) Lack of Standardization:**

There is no standard that defines rules and roles of NoSQL databases. The design and query languages of NoSQL databases vary widely between different NoSQL products – much more widely than they do among traditional SQL databases.

**(ii) Backup of Database:**

Backups are a drawback in NoSQL databases. Though some NoSQL databases like MongoDB provide some tools for backup, these tools are not mature enough to ensure proper complete data backup solution.

**(iii) Consistency:**

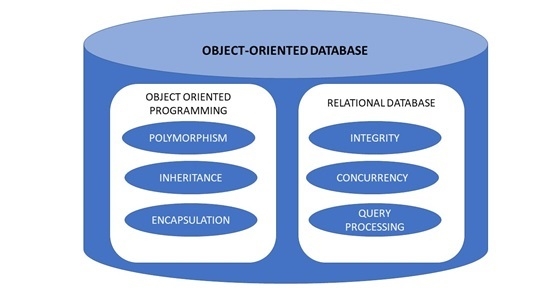
NoSQL puts a scalability and performance first but when it comes to a consistency of the data NoSQL doesn’t take much consideration so it makes it little insecure as compared to the relational database e.g., in NoSQL databases if you enter same set of data again, it will take it without issuing any error whereas relational databases ensure that no duplicate rows get entry in databases.

Now that you have an idea of what NoSQL databases are.

**Question No – 07:**

**The Object-Oriented Database:**

**Object-oriented databases** are a type of database management system. Different database management systems provide additional functionalities. Object-oriented databases add the [database](https://phoenixnap.com/kb/what-is-a-database) functionality to object programming languages, creating more manageable code bases.



## Object-Oriented Programming Concepts

Object-oriented databases closely relate to object-oriented programming concepts. The four main ideas of object-oriented programming are:

* **Polymorphism**
* **Inheritance**
* **Encapsulation**
* **Abstraction**

These four attributes describe the critical characteristics of object-oriented management systems.

### Polymorphism

**Polymorphism** is the capability of an object to take multiple forms. This ability allows the same program code to work with different data types. Both a car and a bike are able to break, but the mechanism is different. In this example, the action break is a polymorphism. The defined action is **polymorphic** — the result changes depending on which vehicle performs.

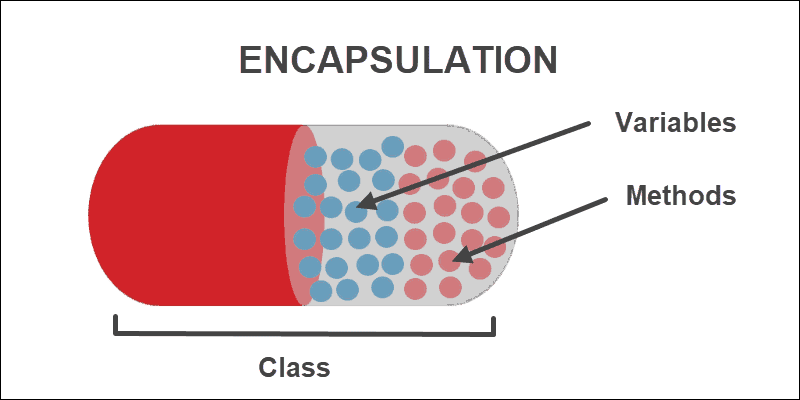
### Inheritance

**Inheritance** creates a hierarchical relationship between related classes while making parts of code reusable. Defining new types inherits all the existing class fields and methods plus further extends them. The existing class is the **parent** class, while the **child** class extends the parent.

For example, a parent class called Vehicle will have child classes Car and Bike. Both child classes **inherit** information from the parent class and **extend** the parent class with new information depending on the vehicle type.

### Encapsulation

**Encapsulation** is the ability to group data and mechanisms into a single object to provide access protection. Through this process, pieces of information and details of how an object works are **hidden**, resulting in data and function security. Classes interact with each other through methods without the need to know how particular methods work.



**Question No – 08:**

**Graph Database:**

Graph databases are purpose-built to store and navigate relationships. Relationships are first-class citizens in graph databases, and most of the value of graph databases is derived from these relationships. Graph databases use nodes to store data entities, and edges to store relationships between entities. An edge always has a start node, end node, type, and direction, and an edge can describe parent-child relationships, actions, ownership, and the like. There is no limit to the number and kind of relationships a node can have.

A graph in a graph database can be traversed along specific edge types or across the entire graph. In graph databases, traversing the joins or relationships is very fast because the relationships between nodes are not calculated at query times but are persisted in the database. Graph databases have advantages for use cases such as social networking, recommendation engines, and fraud detection, when you need to create relationships between data and quickly query these relationships.

The following graph shows an example of a social network graph. Given the people (nodes) and their relationships (edges), you can find out who the "friends of friends" of a particular person are—for example, the friends of Howard's friends.

### 5.png

### The structure of a graph database

Traditionally classified as a type of [NoSQL database](https://preview.techtarget.com:8080/searchDataManagement/definition/NoSQL-Not-Only-SQL), graph databases are sometimes referred to as triple stores. That's because this type of database uses a special index that stores information about nodes, edges and the relationship between them in groups of three.

A triple, which may also be referred to as an assertion, has three main fields: a subject, a predicate and an object. Each subject, predicate or object is represented by a unique resource identifier

**Question No – 09:**

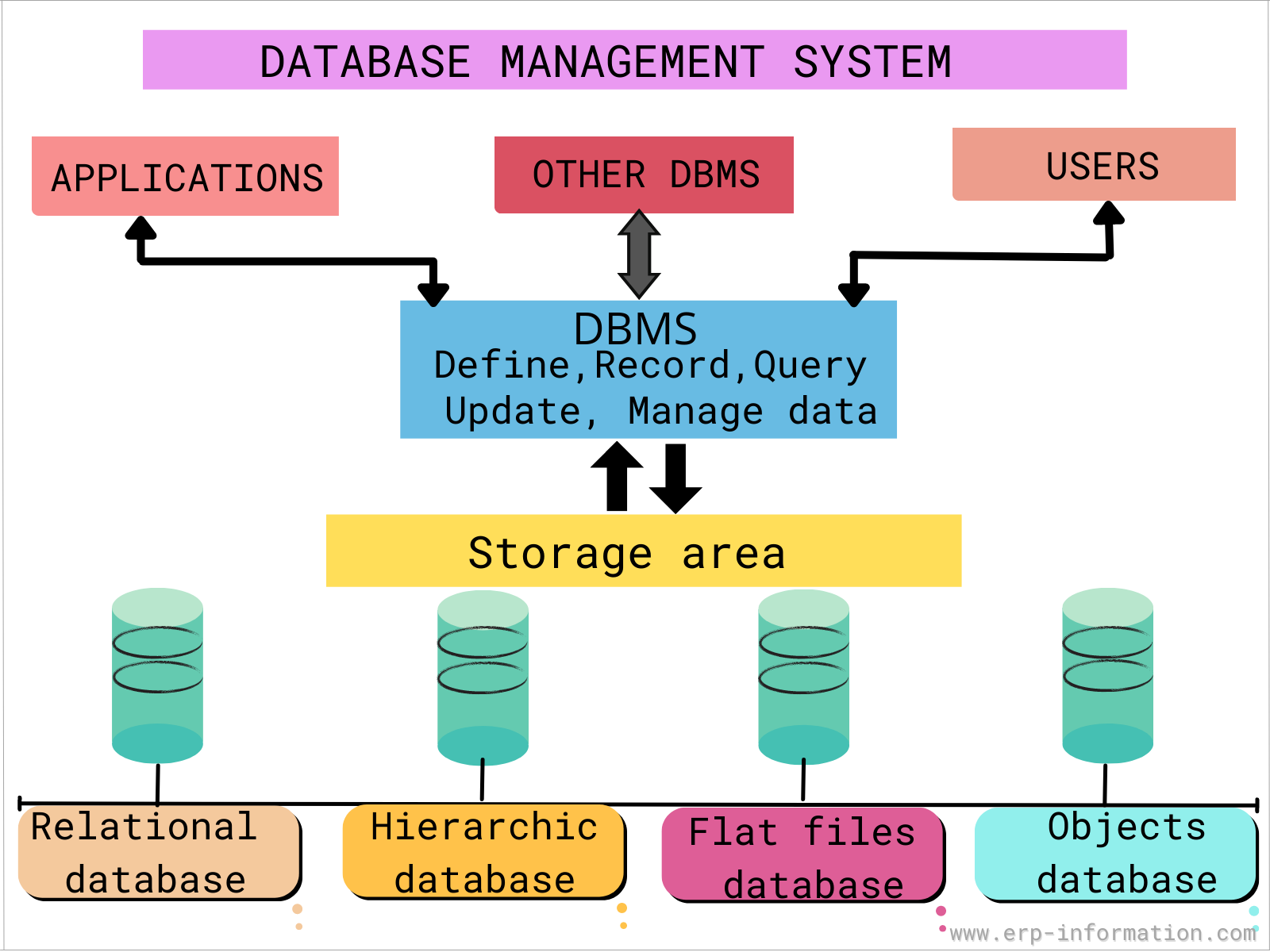
**DBMS (Data Base Management System):**

Database Management Systems (DBMS) are software systems used to store, retrieve, and run queries on data. A DBMS serves as an interface between an end-user and a database, allowing users to create, read, update, and delete data in the database.

DBMS manage the data, the database engine, and the database schema, allowing for data to be manipulated or extracted by users and other programs. This helps provide data security, data integrity, concurrency, and uniform data administration procedures.

DBMS optimizes the organization of data by following a database schema design technique called normalization, which splits a large table into smaller tables when any of its attributes have redundancy in values. DBMS offer many benefits over traditional file systems, including flexibility and a more complex backup system.

Database management systems can be classified based on a variety of criteria such as the data model, the database distribution, or user numbers. The most widely used types of DBMS software are relational, distributed, hierarchical, object-oriented, and network.



**Distributed database management system**

A distributed DBMS is a set of logically interrelated databases distributed over a network that is managed by a centralized database application. This type of DBMS synchronizes data periodically and ensures that any change to data is universally updated in the database.

**Hierarchical database management system**

Hierarchical databases organize model data in a tree-like structure. Data storage is either a top-down or bottom-up format and is represented using a parent-child relationship.

**Network database management system**

The network database model addresses the need for more complex relationships by allowing each child to have multiple parents. Entities are organized in a graph that can be accessed through several paths.

**Relational database management system**

Relational database management systems (RDBMS) are the most popular data model because of its user-friendly interface. It is based on normalizing data in the rows and columns of the tables. This is a viable option when you need a data storage system that is scalable, flexible, and able to manage lots of information.

**Object-oriented database management system**

Object-oriented models store data in objects instead of rows and columns. It is based on object-oriented programming (OOP) that allows objects to have members such as fields, properties, and methods.

## Examples of DBMS

There is a wide range of database software solutions, including both enterprise and open source solutions, available for database management.

Here are some of the most popular database management systems:

### Oracle

Oracle Database is a commercial relational database management system. It utilizes enterprise-scale database technology with a robust set of features right out of the box. It can be stored in the cloud or on-premises.

### MySQL

MySQL is a relational database management system that is commonly used with open-source content management systems and large platforms like Facebook, Twitter, and Youtube.

### SQL Server

Developed by Microsoft, SQL Server is a relational database management system built on top of structured query language (SQL), a standardized programming language that allows database administrators to manage databases and query data.

**Question No- 10:**

**Advantage of DBMS:**

## Reducing Data Redundancy

The file based data 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 a 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.

## Sharing of Data

In a database, the users of the database can share the data among themselves. There are various levels of authorization to access the data, and consequently the data can only be shared based on the correct authorization protocols being followed.

Many remote users can also access the database simultaneously and share the data between themselves.

## Data Integrity

Data integrity means that the data is accurate and consistent in the database. Data Integrity is very important as there are multiple databases in a DBMS. All of these databases contain data that is visible to multiple users. So it is necessary to ensure that the data is correct and consistent in all the databases and for all the users.

## Data Security

Data Security is vital concept in a database. Only authorized users should be allowed to access the database and their identity should be authenticated using a username and password. Unauthorized users should not be allowed to access the database under any circumstances as it violates the integrity constraints.

## Privacy

The privacy rule in a database means only the authorized users can access a database according to its privacy constraints. There are levels of database access and a user can only view the data he is allowed to. For example - In social networking sites, access constraints are different for different accounts a user may want to access.

## Backup and Recovery

Database Management System automatically takes care of backup and recovery. The users don't need to backup data periodically because this is taken care of by the DBMS. Moreover, it also restores the database after a crash or system failure to its previous condition.

## Data Consistency

Data consistency is ensured in a database because there is no data redundancy. All data appears consistently across the database and the data is same for all the users viewing the database. Moreover, any changes made to the database are immediately reflected to all the users and there is no data inconsistency.