**What is a database management system?**

A database management system (DBMS) is system software for creating and managing databases. A DBMS makes it possible for end users to create, protect, read, update and delete data in a [database](https://www.techtarget.com/searchdatamanagement/definition/database). The most prevalent type of [data management](https://www.techtarget.com/searchdatamanagement/definition/data-management) platform, the DBMS essentially serves as an interface between databases and users or application programs, ensuring that data is consistently organized and remains easily accessible.

**Application of DBMS**

Below are the popular database system applications:

| **Sector** | **Use of DBMS** |
| --- | --- |
| Banking | For customer information, account activities, payments, deposits, loans, etc. |
| Airlines | For reservations and schedule information. |
| Universities | For student information, course registrations, colleges, and grades. |
| Telecommunication | It helps to keep call records, monthly bills, maintain balances, etc. |
| Finance | For storing information about stock, sales, and purchases of financial instruments like stocks and bonds. |
| Sales | Use for storing customer, product & sales information. |
| Manufacturing | It is used to manage the supply chain and track the production of items. Inventories status in warehouses. |
| HR Management | For information about employees, salaries, payroll, deduction, generation of paychecks, etc. |

## ****Features of Database Management System****

### 1. Minimum Redundancy and Duplication

Because databases are used by so many people, the risks of data duplication are relatively high. But in a database management system, data files are shared which brings down data duplication and redundancy. Due to the fact that all information in a database management system occurs only once, the odds of duplication are quite low. In other words, the same data file is accessible to all the people using the database, and the changes made by any one of the users get reflected for the data file of all the users and therefore, redundancy and duplication are avoided.

### 2. Reduced amount of space and money spent on storage

All database management systems must save a large amount of data. However, proper data integration saves a lot of space in the database management system. Companies spend a lot of money to keep their data safe. They will save money on data storage and data entry if they have managed data to store.

### 3. Data Organization

In a Database Management system, a digital repository’s information is structured in a clear hierarchical structure using records, tables, and objects. Every piece of information which we enter into our database will be structured in a catalog, making it easy to search and edit our records later.

### 4. Customization of the Database

Along with the default and required components (records, tables, or objects) that make up a database’s structure, custom elements can be constructed to fit the demands of unique users. For example, Binary Large Objects or BLOBS can be used to store images in databases and mappings can be maintained between various tables to implement complex entities.

### 5. Data Retrieval

The database management system, or DBMS, accepts and stores data from users. Users can subsequently get their records from the database and save them as a file, print them, or display them on the screen. Data Retrieval becomes a big [advantage of the database management systems](https://www.scaler.com/topics/advantages-of-dbms/) as only authenticated users can fetch data from the database and unauthenticated users are denied access, thus improving the security of the data.

### 6. Usage Of Query Languages

A typical Database Management System allows users to utilize query languages for collecting, searching, sorting, altering, and other tasks that enable them to manipulate their database entries. An example of a famous query language is SQL (Structured Query Language). Anyone, even without the knowledge of any programming language, can access a Database Management System easily without hassle.

# **Types of Databases**

There are various types of databases used for storing different varieties of data:



## 1) Centralized Database

It is the type of database that stores data at a centralized database system. It comforts the users to access the stored data from different locations through several applications. These applications contain the authentication process to let users access data securely. An example of a Centralized database can be Central Library that carries a central database of each library in a college/university.

## 2) Distributed Database

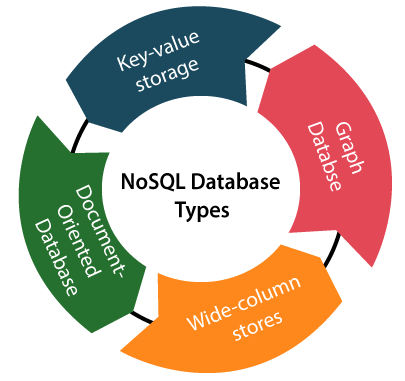
Unlike a centralized database system, in distributed systems, data is distributed among different database systems of an organization. These database systems are connected via communication links. Such links help the end-users to access the data easily. **Examples** of the Distributed database are Apache Cassandra, HBase, Ignite, etc.

## 3) Relational Database

This database is based on the relational data model, which stores data in the form of rows(tuple) and columns(attributes), and together forms a table(relation). A relational database uses SQL for storing, manipulating, as well as maintaining the data. E.F. Codd invented the database in 1970. Each table in the database carries a key that makes the data unique from others. **Examples** of Relational databases are MySQL, Microsoft SQL Server, Oracle, etc

4) NoSQL Database

Non-SQL/Not Only SQL is a type of database that is used for storing a wide range of data sets. It is not a relational database as it stores data not only in tabular form but in several different ways. It came into existence when the demand for building modern applications increased. Thus, NoSQL presented a wide variety of database technologies in response to the demands. We can further divide a NoSQL database into the following four types:



5) Cloud Database

A type of database where data is stored in a virtual environment and executes over the cloud computing platform. It provides users with various cloud computing services (SaaS, PaaS, IaaS, etc.) for accessing the database. There are numerous cloud platforms, but the best options are:

* Amazon Web Services(AWS)
* Microsoft Azure
* Kamatera
* PhonixNAP
* ScienceSoft
* Google Cloud SQL, etc.

6) Object-oriented Databases

The type of database that uses the object-based data model approach for storing data in the database system. The data is represented and stored as objects which are similar to the objects used in the object-oriented programming language.

7) Hierarchical Databases

It is the type of database that stores data in the form of parent-children relationship nodes. Here, it organizes data in a tree-like structure.



Data get stored in the form of records that are connected via links. Each child record in the tree will contain only one parent. On the other hand, each parent record can have multiple child records.

8) Network Databases

It is the database that typically follows the network data model. Here, the representation of data is in the form of nodes connected via links between them. Unlike the hierarchical database, it allows each record to have multiple children and parent nodes to form a generalized graph structure.

**Example of DBMS**

1.Oracle Database

2.MySQl Database

3.Postgre Database

4.MongoDB

5.MS Access

6.Microsoft SQl Server etc…

# **What is MySQL?**

MySQL is the world’s most popular open source database. According to [DB-Engines](https://db-engines.com/en/ranking), MySQL ranks as the second-most-popular database, behind [Oracle Database](https://www.oracle.com/database/). MySQL powers many of the most accessed applications, including Facebook, Twitter, Netflix, Uber, Airbnb, Shopify, and Booking.com.

Since MySQL is open source, it includes numerous features developed in close cooperation with users over more than 25 years. So it’s very likely that your favorite application or programming language is supported by MySQL Database.

## MySQL is a relational database management system

[Databases](https://www.oracle.com/database/what-is-database/) are the essential data repository for all software applications. For example, whenever someone conducts a web search, logs in to an account, or completes a transaction, a database system is storing the information so it can be accessed in the future.

A [relational database](https://www.oracle.com/database/what-is-a-relational-database/) stores data in separate tables rather than putting all the data in one big storeroom. The database structure is organized into physical files optimized for speed. The logical data model, with objects such as data tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one to one, one to many, unique, required, or optional, and “pointers” between different tables. The database enforces these rules so that with a well-designed database your application never sees data that’s inconsistent, duplicated, orphaned, out of date, or missing.

The “SQL” part of “MySQL” stands for “Structured Query Language.” SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

## MySQL is open source

[Open source](https://developer.oracle.com/open-source/what-is-open-source/) means it’s possible for anyone to use and modify the software. Anybody can download MySQL software from the internet and use it without paying for it. You can also change its source code to suit your needs. MySQL software uses the [GNU General Public License](http://www.fsf.org/licenses/) (GPL) to define what you may and may not do with the software in different situations.

If you feel uncomfortable with the GNU GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from Oracle. See the [MySQL Licensing Information section](https://www.mysql.com/about/legal/) for more information.

## MySQL benefits

**Reliability:** MySQL is one of the most mature and widely used databases. It has been tested in a wide variety of scenarios for more than 25 years, including by many of the world’s largest companies. Organizations depend on MySQL to run business-critical applications because of its reliability.

**Scalability:**MySQL scales to meet the demands of the most accessed applications. MySQL’s native replication architecture enables organizations such as Facebook to scale applications to support billions of users.

**Performance:** MySQL HeatWave is [faster and less expensive](https://www.oracle.com/mysql/heatwave/performance/) than other database services, as demonstrated by multiple standard industry benchmarks, including TPC-H, TPC-DS, and CH-benCHmark.

**High availability:** MySQL delivers a complete set of native, fully integrated replication technologies for high availability and disaster recovery. For business-critical applications, and to meet service-level agreement commitments, customers can achieve

* Recovery point objective = 0 (zero data loss)
* Recovery time objective = seconds (automatic failover)

**Security:** [Data security](https://www.oracle.com/security/database-security/what-is-data-security/) entails protection and compliance with industry and government regulations, including the European Union General Data Protection Regulation, the Payment Card Industry Data Security Standard, the Health Insurance Portability and Accountability Act, and the Defense Information Systems Agency’s Security Technical Implementation Guides. MySQL Enterprise Edition provides advanced security features, including authentication/authorization, transparent data encryption, auditing, data masking, and a database firewall.

**Flexibility:**The MySQL Document Store gives users maximum flexibility in developing traditional SQL and NoSQL schema-free database applications. Developers can mix and match relational data and JSON documents in the same database and application.

Difference Between DBMS and RDBMS

|  |  |  |
| --- | --- | --- |
| **No.** | **DBMS** | **RDBMS** |
| 1) | DBMS applications store **data as file**. | RDBMS applications store **data in a tabular form**. |
| 2) | In DBMS, data is generally stored in either a hierarchical form or a navigational form. | In RDBMS, the tables have an identifier called primary key and the data values are stored in the form of tables. |
| 3) | **Normalization is not** present in DBMS. | **Normalization is** present in RDBMS. |
| 4) | DBMS does **not apply any security** with regards to data manipulation. | RDBMS **defines the integrity constraint** for the purpose of ACID (Atomocity, Consistency, Isolation and Durability) property. |
| 5) | DBMS uses file system to store data, so there will be **no relation between the tables**. | in RDBMS, data values are stored in the form of tables, so a **relationship** between these data values will be stored in the form of a table as well. |
| 6) | DBMS has to provide some uniform methods to access the stored information. | RDBMS system supports a tabular structure of the data and a relationship between them to access the stored information. |
| 7) | DBMS **does not support distributed database**. | RDBMS **supports distributed database**. |
| 8) | DBMS is meant to be for small organization and **deal with small data**. it supports **single user**. | RDBMS is designed to **handle large amount of data**. it supports **multiple users**. |
| 9) | Examples of DBMS are file systems, **xml** etc. | Example of RDBMS are **mysql**, **postgre**, **sql server**, **oracle** etc. |