DBMS: DATABASE MANAGEMENT SYSTEMS EXPLAINED

Data is the cornerstone of any modern software application, and databases are the most common way to store and manage data used by applications.

With the explosion of web and cloud technologies, databases have evolved from traditional relational databases to more advanced types of databases such as NoSQL, columnar, key-value, hierarchical, and distributed databases. Each type has the ability to handle structured, semi-structured, and even unstructured data.

On top of that, databases are continuously handling mission-critical and sensitive data. When this is coupled with compliance requirements and the distributed nature of most data sets, managing databases has become highly complex. As a result, organizations require robust, secure, and user-friendly tools to maintain these databases.

This is where database management systems come into play—by offering a platform to manage databases. Let's take a look.

What is a database management system?

A database management system (DBMS) is a software tool that enables users to manage a database easily. It allows users to access and interact with the underlying data in the database. These actions can range from simply querying data to defining database schemas that fundamentally affect the database structure.

Furthermore, DBMS allow users to interact with a database securely and concurrently *without* interfering with each user *and* while maintaining data integrity.

Database tasks in a DBMS

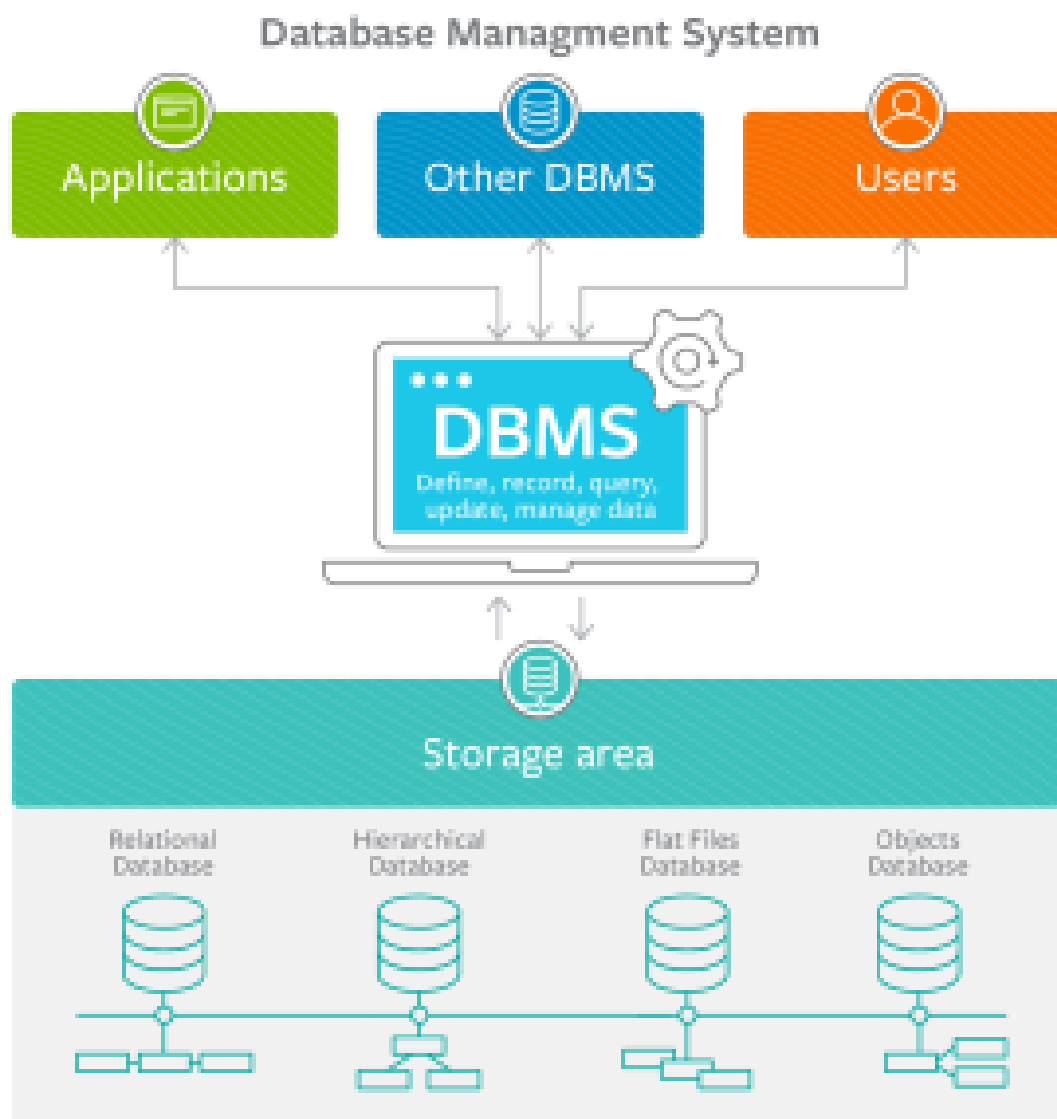The typical database administrative tasks that can be performed using a DBMS include:

- **Configuring authentication and authorization.** Easily configure user accounts, define access policies, modify restrictions, and access scopes. These operations allow administrators to limit access to underlying data, control user actions, and manage users in databases.
- **Providing data backups and snapshots.** DBMS can simplify the backup process of databases by providing a simpler and straightforward interface to manage backups and snapshots. They can even move these backups to third-party locations such as cloud storage for safekeeping.
- **Performance tuning.** DBMS can monitor the performance of databases using integrated tools and enable users to tune databases by creating optimized indexes. It reduces [I/O usage](#) to optimize SQL queries, enabling the best performance from the database.
- **Data recovery.** In a recovery operation, DBMS provides a recovery platform with the necessary tools to fully or partially restore databases to their previous state—effortlessly.

All these administrative tasks are facilitated using a single management interface. Most modern DBMS support handling multiple database workloads from a centralized DBMS software, even in a distributed database scenario. Furthermore, they allow organizations to have a governable top-down view of all the data, users, groups, locations, etc., in an organized manner.

(Explore the [role of DBAs](#), or database administrators.)

DBMS system schematic

The following diagram illustrates the schematic of a DBMS system:

Database Managment System

Applications | Other DBMS | Users

DBMS
Define, record, query, update, manage data

Storage area

Relational Database | Hierarchical Database | Flat Files Database | Objects Database

bmc

Components of a database management system

All DBMS comes with various integrated components and tools necessary to carry out almost all database management tasks. Some DBMS software even provides the ability to extend beyond the core functionality by integrating with third-party tools and services, directly or via plugins.

In this section, we will look at the common components that are universal across all DBMS software, including:

- Storage engine
- Query language
- Query processor
- Optimization engine
- Metadata catalog
- Log manager
- Reporting and monitoring tools
- Data utilities
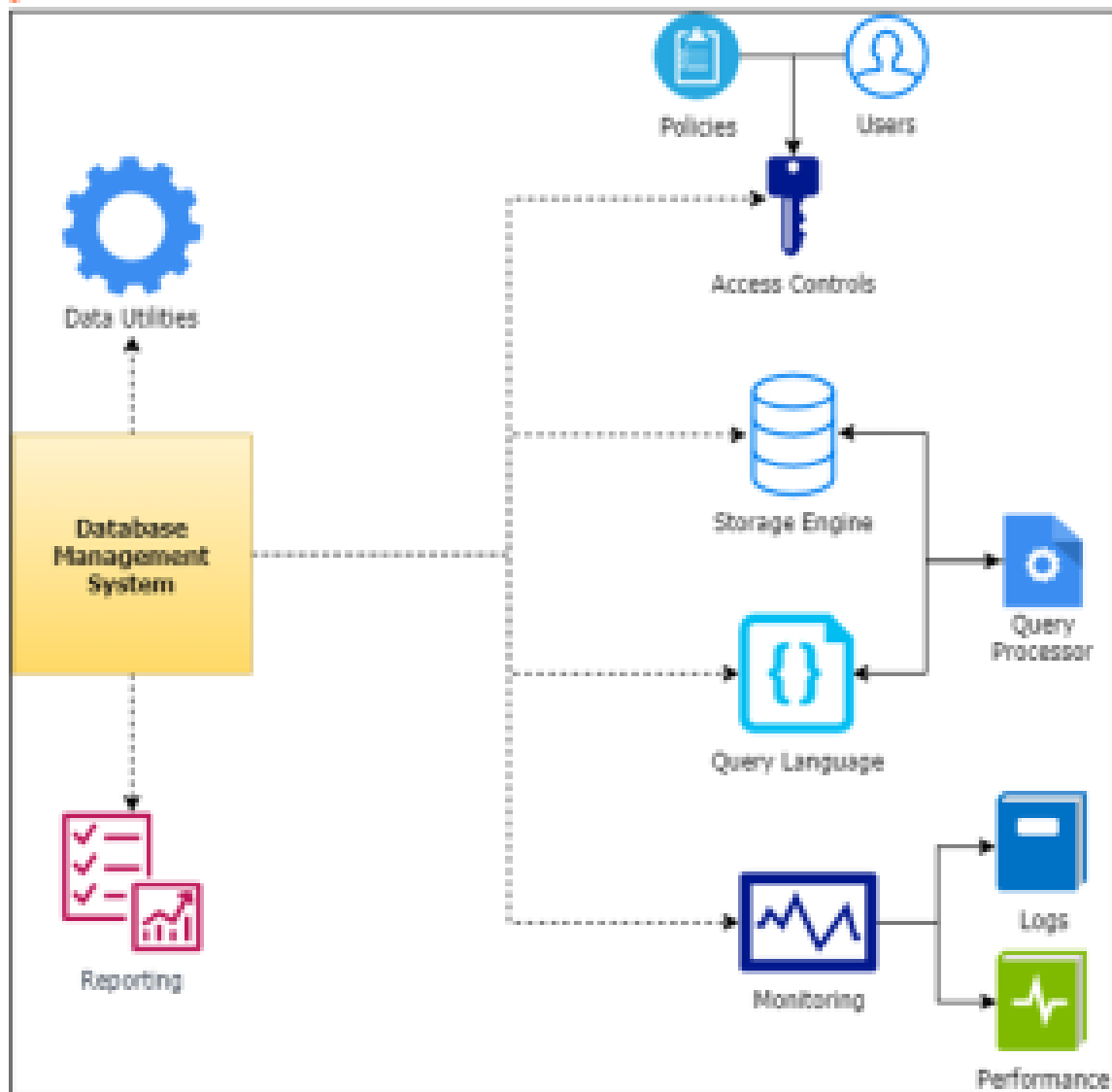
Storage engine

The storage engine is the core component of the DBMS that interacts with the file system at an OS level to store data. All SQL queries which interact with the underlying data go through the storage engine.

Query language

A database access language is required for interacting with a database, from creating databases to simply inserting or retrieving data. A proper DBMS must support one or multiple query languages and language dialects. Structured query language (SQL) and MongoDB Query Language (MQL) are two query languages that are used to interact with the databases.

In many query languages, the query language functionality can be further categorized according to specific tasks:

- **Data Definition Language (DDL).** This consists of commands that can be used to define database schemas or modify the structure of database objects.
- **Data Manipulation Language (DML).** Commands that directly deal with the data in the

database. All [CRUD operations](#) come under DML.

- **Data Control Language (DCL).** This deals with the permissions and other access controls of the database.
- **Transaction Control Language (TCL).** Command which deals with internal database transactions.

Query processor

This is the intermediary between the user queries and the database. The query processor interprets the queries of users and makes them actionable commands that can be understood by the database to perform the appropriate functionality.

Optimization engine

The optimization Engine allows the DBMS to provide insights into the performance of the database in terms of optimizing the database itself and queries. When coupled with [database monitoring tools](#), it can provide a powerful toolset to gain the best performance out of the database.

Metadata catalog

This is the centralized catalog of all the objects within the database. When an object is created, the DBMS keeps a record of that object with some metadata about it using the metadata catalog. Then, this record can be used to:

- Verify user requests to the appropriate database objects
- Provide an overview of the complete database structure

Log manager

This component will keep all the logs of the DBMS. These logs will consist of user logins and activity, database functions, backups and restore functions, etc. The log manager ensures all these logs are properly recorded and easily accessible.

(Compare [logs to monitoring](#).)

Reporting & monitoring tools

Reporting and monitoring tools are another standard component that comes with a DBMS. Reporting tools will enable users to generate reports while monitoring tools enable monitoring the databases for resource consumption, user activity, etc.

Data utilities

In addition to all the above, most DBMS software comes with additional inbuilt utilities to provide functionality such as:

- Data integrity checks
- Backup and restore
- Simple database repair

- Data validations
- Etc.

Types of database management systems

There are many different types of DBMS, yet we can categorize the most commonly used DBMS into three types.

Relational database management systems (RDBMS)

This is the most common type of DBMS. They are used to interact with databases that contain structured data in a table format with predefined relationships. Moreover, they use structured query language (SQL) to interact with databases. Microsoft SQL, MySQL, and Oracle Database are some popular DBMS that come under this category.

Document database management systems (DoDBMS)

These DoDBMS are used to manage databases that contain data stored in JSON-like structures with limited or no relationship structure. They are powered by query languages such as MongoDB query language (MQL) for database operations. MongoDB, Azure Cosmos DB are some prominent examples of DoDBMS.

Columnar database management systems (CDBMS)

As the name suggests, this type of DBMS is used to manage columnar databases that store data in columns instead of rows, emphasizing high performance. Some databases that use columnar format are Apache Cassandra, Apache HBase, etc.

Advantages of a DBMS

DBMS was introduced to solve the fundamental issues associated with storing, managing, accessing, securing, and auditing data in traditional file systems. Software users and organizations can gain the following benefits by using DBMS:

Increased data security

DBMS provides the ability to control users and enforce policies for security and compliance management. This controlled user access increases the database security and makes the data less vulnerable to security breaches.

Simple data sharing

DBMS enables users to access the database securely regardless of their location. Thus, they can handle any database-related task promptly without the need for complex access methods or worrying about database security. On top of that, DBMS allows multiple users to collaborate effectively when interacting with the database.

Data integration

DBMS allows users to gain a centralized view of databases spread across multiple locations and manage them using a single interface rather than operating them as separate entities.

Abstraction & independence

DBMS enables users to change the physical schema of a database without changing the logical schema that governs database relationships. As a result, organizations can scale the underlying database infrastructure without affecting the database operations.

Furthermore, any change to the logical schema can also be carried out without affecting applications that access the databases.

Streamlined backup & recovery mechanism

Most databases have built-in backup and recovery tools. Yet, DBMS offers centralized tools to facilitate backup and recovery functionality more conveniently and thereby provide a better user experience. Securing data has become easier than ever with functionality like:

- Automated snapshots
- Backup scheduling
- Backup verifications
- Multiple recovery methods

Uniform management & monitoring

DBMS provides a single interface to carry out all the management and monitoring tasks, thus simplifying the workload of database administrators. These tasks can range from database creation and schema modifications to reporting and auditing.

DBMSs are essential

DBMS is an essential component for any organization when it comes to managing databases. The scale, complexity, and feature set of a DBMS will depend on the specific DBMS and requirements of the organizations.

With different DBMS providing different feature sets, it is paramount that organizations rigorously evaluate the DBMS software before committing to a single system. However, a properly configured DBMS will greatly simplify the management and maintenance of databases at any scale.

Related reading

- BMC Big Data & Machine Learning Blog
- BMC IT Operations Blog
- Introduction To Database DevOps
- What Is DBaaS? Database-as-a-Service Explained
- CAP Theorem for Databases: Consistency, Availability & Partition Tolerance
- Data Ethics for Companies