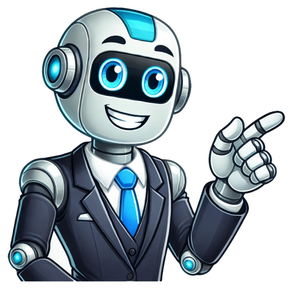
I'm not a robot



Share — copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt — remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation . No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. A database is a structured and systematic way of storing information to be accessed, analyzed, transformed, updated and moved (to other databases). A database is a means of organizing information in a way that users can quickly navigate data, spot trends and perform other actions. Although databases may come in different formats, most are stored on computers for greater convenience. To begin understanding databases, consider an Excel notebook or Google sheet. Spreadsheets like these are a basic form of a table. Databases are almost exclusively organized in tables and those tables have rows and columns. So, think of a simple database as a collection of spreadsheets (or tables) joined together in a systematic way. Related Reading From Built In ExpertsPython Databases 101: How to Choose a Database Library Database Tutorial for Beginners. | Video: Lucid Software What Is a Database? A database is a structured collection of information, or data, stored electronically in a computer system, often managed using a database management system (DBMS). Databases come in many formats — like relational, non-relational and more — in order to handle different types of data and perform various tasks. Databases can be hosted on servers located on-premises at an organization's office, off-premises at an organization's data center or within an organization's cloud infrastructure. Why Do We Use Databases? Computerized databases are the foundation for handling information related to products, data analysis, business processes and more. Many of the services you use online every day (banking, social media, shopping, email) are all built on top of databases. Today, databases are used for many reasons. Databases Hold Data Efficiently We use databases because they are an extremely efficient way of holding vast amounts of data and information. Databases around the world store everything from your credit card transactions to every click you make within one of your social media accounts. Given there are billions of people on the planet, that's a lot of data. Databases Allow Smooth Transactions Databases allow access to various services which, in turn, allow you to access your accounts and perform transactions all across the internet. For example, your bank's login page will ping a database to figure out if you've entered the right password and username. Your favorite online shop pings your credit card's database to pull down the funds needed for you to buy that item you've been eyeing. Databases Update Information Quickly Databases allow for easy information updates on a regular basis. Creating a post on a social media account, directly depositing your salary into your bank account or buying a plane ticket for your next vacation are all updates made to a database and displayed back to you almost instantaneously. Databases Simplify Data Analysis Databases make research and data analysis much easier because they are highly structured storage areas of data and information. This means businesses and organizations can easily analyze databases once they know how a database is structured. Common structures (e.g. table formats, cell structures like date or currency fields) and common database querying languages (e.g., SQL) make database analysis easy and efficient. Types of Databases There are many types of databases used today. Below are some of the more prominent ones. 1. Hierarchical Databases Hierarchical databases were the earliest form of databases. You can think of these databases like a simplified family tree. There's a singular parent object (like a table) that has child objects (or tables) under it. A parent can have one or many child objects but a child object only has one parent. The benefit of these databases is that they're incredibly fast and efficient, plus there's a clear, threaded relationship from one object to another. The downside to hierarchical databases is that they're very rigid and highly structured. 2. Relational Databases Relational databases are perhaps the most popular type of database. Relational databases are set up to connect their objects (like tables) to each other with keys. For example, there might be one table with user information (name, username, date of birth, customer number) and another table with purchase information (customer number, item purchased, price paid). In this example, the key that creates a relationship between the tables is the customer number. 3. Non-Relational or NoSQL Databases Non-relational databases were invented more recently than relational databases and hierarchical databases in response to the growing complexity of web applications. Non-relational databases are any database that doesn't use a relational model. You might also see them referred to as NoSQL databases. Non-relational databases store data in different ways, such as unstructured data, structured document format or as a graph. Relational databases are based on a rigid structure, whereas non-relational databases are more flexible. 4. Cloud Databases Cloud databases refer to information that's accessible in a hybrid or cloud environment. All users need is an internet connection to reach their files and manipulate them like any other database. A convenience of cloud databases is that they don't require extra hardware to create more storage space. Users can either build a cloud database themselves or pay for a service to get started. 5. Centralized Databases Centralized databases are contained within a single computer or another physical system. Although users may access data through devices connected within a network, the database itself operates from one location. This approach may work best for larger companies or organizations that want to prioritize data security and efficiency. 6. Distributed Databases Distributed databases run on more than one device. That can be as simple as operating several computers on the same site, or a network that connects to many devices. An advantage of this method is that if one computer goes down, the other computers and devices keep functioning. 7. Object-Oriented Databases Object-oriented databases perceive data as objects and classes. Objects are specific data — like names and videos — while classes are groups of objects. Storing data as objects means users don't have to distribute data across tables. This makes it easier to determine the relationships between variables and analyze the data. 8. Graph Databases Graph databases highlight the relationships between various data points. While users may have to do extra work to determine trends in other types of databases, graph databases store relationships right next to the data itself. Users can then immediately see how various data points are connected to each other. What Are the Components of a Database? The components of a database vary slightly depending on whether the database is hierarchical, relational or non-relational. However, here's a list of database components you might expect to be associated with any database. Schema The database schema is essentially the design of the database. A schema is developed at the early conceptual stages of building a database. It's also a valuable source of ongoing information for those wanting to understand the database's design. Constraints and Rules Databases use constraints to determine what types of tables can (and cannot) be stored and what types of data can live in the columns or rows of the database tables, for example. These constraints are important because they ensure data is structured, less corruptible by unsanctioned data structures and that the database is regulated so users know what to expect. These constraints are also the reason why databases are considered rigid. Metadata Metadata is essentially the data about the data. For example, the metadata of a photo may include the date it was taken, its location and its file format. Each database or object has metadata, which the database software reads in order to understand what's in the database. You can think of metadata as the database schema design and constraints combined together so a machine knows what kind of database it is and what actions can (or can't) be performed within the database. Query Language Each database can be queried. In this case, "queried" means people or services can access the database. That querying is done by way of a particular language or code snippet. The most common querying language is SQL (Structured Query Language) but there are also many other languages and even SQL variations like MySQL, Presto and Hive. Objects Each database is a collection of objects. There are a few different types of objects stored within databases such as tables, views, indexes, sequences and synonyms. The most well known of these are tables, which are like spreadsheets that organize data in rows and columns. You may also hear the term "object instance," which is simply an instance or element of an object. For example, a table called "Transactions" in a database is an instance of the object-type table. Database Advantages The structured nature of databases offers a range of benefits for professional and casual users alike. Below are some of the more prominent advantages: Improved data sharing and handling Improved data storage capacity Improved data integrity and data security Reduced data inconsistency Quick data access Increased productivity Improved data-driven decision making Database Disadvantages Although databases can be helpful for many, there are some limitations to consider before investing in a database: High cost High complexity Required dedicated database management staff Risk of database failure Applications of Databases When used correctly, databases can be a helpful tool for organizations in various industries looking to better arrange their information. Common use cases include: Healthcare: storing massive amounts of patient data. Logistics: monitoring and analyzing route information and delivery statuses. Insurance: storing customer data like addresses, policy details and driver history. Finance: handling account details, invoices, stock information and other assets. E-commerce: compiling and arranging data on products and customer behavior. Transportation: storing passengers' names, scheduled flights and check-in status. Manufacturing: keeping track of machinery status and production goals. Marketing: collecting data on demographics, purchasing habits and website visits. Education: tracking student grades, course schedules and more. Human resources: organizing personnel info, benefits and tax information. What Is a Database Management System? A database management system (DBMS) is a software package used to create and manage databases. In other words, a DBMS makes it possible for users to actually interact with the database. The DBMS is the user interface (UI) that allows us to access, add, modify and delete content from the database. There are several types of database management systems, including relational, non-relational and hierarchical. Evolution of Databases Storing information is nothing new, but the rise of computers in the 1960s marked a shift toward more digital forms of databases. While working for GE, Charles Bachman created the Integrated Data Store, ushering in a new age of computerized databases. IBM soon followed suit with its Information Management System, a hierarchical database. In the 1970s, IBM's Edgar F. Codd released a paper touting the benefits of relational databases, leading to IBM and the University of California, Berkeley releasing their own models. Relational databases became popular in the following years, with more businesses developing models and using SQL. Even though object-oriented databases became an alternative in the 1980s, relational databases remained the gold standard. The invention of the World Wide Web led to greater demand for databases in the 1990s. MySQL and NoSQL databases entered the scene, competing with the commercial databases developed by businesses. Object-oriented databases also began to replace relational databases in popularity. During the 2000s and 2010s, organizations began to collect larger volumes of data, and many turned to the scalability offered by NoSQL databases. Distributed databases provided another way to organize this proliferating data, storing it away in multiple locations. Future of Databases As organizations handle increasing amounts of data, future databases must be able to keep up. Users will expect databases to be accessible across the globe and able to deal with limitless volumes of data. As a result, it's likely that more companies will migrate their data to cloud environments. With the increase in data has also come a spike in cybersecurity threats, so organizations can be expected to complement their cloud environments with reinforced security measures. Databases will become more easily accessible for authorized personnel only, while companies adopt tools and best practices for keeping their data out of the wrong hands. Spreadsheets organize data into rows and columns, with each individual cell housing the actual data. Databases also employ rows and columns, but each cell contains a record of data gathered from an external table. As a result, databases provide more ways to arrange and structure information as opposed to spreadsheets. Relational databases are the most commonly used database type. A database is highly organized information that is designed to be easily accessible and navigable for users. Most databases are stored on computers, making it possible to quickly analyze, transform and manipulate data in other ways. Common types of databases include: Relational databases Non-relational (NoSQL) databases Hierarchal databases Cloud databases Centralized databases Distributed databases Object-oriented databases Graph databases No, Microsoft Excel is a spreadsheet software program used to organize data in a spreadsheet format. However, Excel can connect to data sources like Microsoft Access and Microsoft SQL Server databases. Here I am going to provide you DBMS Handwritten Notes so that you can increase your basic knowledge of Database Management system and you can prepare for your exam easily. Download Now Download Now Friends, if you face any problem in downloading DBMS Notes PDF, then tell in the comment below. Also Read -: Enjoy sharper detail, more accurate color, lifelike lighting, believable backgrounds, and more with our new model update. Your generated images will be more polished than ever.See What's NewExplore how consumers want to see climate stories told today, and what that means for your visuals.Download Our Latest VisualGPS ReportData-backed trends. Generative AI demos. Answers to your usage rights questions. Our original video podcast covers it all—now on demand.Watch NowEnjoy sharper detail, more accurate color, lifelike lighting, believable backgrounds, and more with our new model update. Your generated images will be more polished than ever.See What's NewExplore how consumers want to see climate stories told today, and what that means for your visuals.Download Our Latest VisualGPS ReportData-backed trends. Generative AI demos. Answers to your usage rights questions. Our original video podcast covers it all—now on demand.Watch Now A database is an organized collection of information that can be searched, sorted, and updated. This data is often stored electronically in a computer system called a database management system (DBMS). Databases typically organize data in rows and columns for easy processing and retrieval. Oftentimes, you'll need to use a programming language, such as structured query language (SQL), to interact with your database.Databases are similar to spreadsheets, but there are several key differences. In general, databases are much larger than spreadsheets and so can store more data, and they allow for multiple users to access data at the same time. For these reasons, people who work with data, such as data analysts and data scientists, often work with databases rather than spreadsheets.SQL is a programming language used on almost all relational databases to query and manipulate data. It is still widely used today for querying databases to access the right data needed, but new languages have begun to appear in this space.Common database typesThere are several types of databases, including:Relational database: A relational database stores and allows access to data. These types of databases are considered "relational" because the items within them have pre-determined relationships with one another. Data is stored in tables, which are connected by unique IDs or "keys." To access specific information, users enter the key to access the data that has been programmed to be related to that key.NoSQL database (or nonrelational database): A non-relational database (also known as a NoSQL database) stores data in whatever format is best for the type of data being stored. They tend to contain unstructured data, or data that is less defined, like emails, videos, images, and documents. They are called NoSQL because they don't use SQL.Distributed database: A distributed database stores data in several different physical locations. Processing data in this type of database is spread out. Distributed databases can be homogenous and have the same hardware in each physical location and run the same systems and applications, or they can be heterogeneous and have different operating systems in each location.Object-oriented database: An object-oriented database focuses on organizing objects rather than actions or logic. Instead of being assigned an alphanumeric value, it would remain its original object type.Graph database: A graph database is a type of NoSQL database. It stores, queries, and maps relationships according to the graph theory. Graph databases are used to analyze interactions and connections. They consist of nodes and edges and use a declarative programming language called SPARQL.Cloud database: A cloud database is built in a cloud to optimize for a virtual work setting and distribution. Organizations tend to be charged based on the amount of storage or bandwidth they use.Open-source database: These databases are open-source, meaning anyone can contribute or edit the source code. They can be SQL or NoSQL.Data warehouse: Data warehouses are central repositories for data. A data warehouse is designed to be swift, so users can query and analyze data quickly.Each database type is characterized by specific storage and retrieval practices, data types, job functions, and use cases.Learn more about databases from experts at Google:Components of a databaseThe different types of databases vary in terms of data structure, data types, and schema. Database schema is a blueprint that outlines a database's architecture, describing how the data inside is organized and how different elements (such as foreign and primary keys, fields, etc.) relate to each other.All types of databases have the following five components:Hardware: Database hardware is the physical device that the software runs on, so that users can query and pull data from it. Hardware examples include computers, hard drives, and servers.Software: From the hardware, database software allows users to manipulate the database for their needs. A database management system (DBMS) manages several databases.Data: The data itself is a core component of the database. It is the raw information that is stored on the database and managed by database administrators. Users can then access the data to derive meaning for specific teams and projects.Procedures: Procedures are the rules that determine how the database runs and handles data.Data access language: Data access languages are programming languages (such as SQL) that are used to control and manage the database. They must align with the DBMS and work in sync.Related termsLearn more about data base types and data analysis on CourseraGain hands-on experience with gathering, cleaning, and analyzing data using databases and other analytics tools by pursuing the Google Data Analytics Professional Certificate on Coursera. Develop skills for an in-demand career in data analytics while you learn at the pace that works for you. 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. Introduction of DBMS What is a database management system (DBMS)? A database management system (DBMS) is a software tool for creating, managing, and reading a database. With DBMS, users can 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 structure of DBMS. Furthermore, DBMS allows users to interact with a database securely and concurrently without interfering with each user and while maintaining data integrity. Unlock the potential of IT Service Management with BMC Helix ITSM. › What are the functions of DBMS? The typical DBMS tasks or functions include: User access and control. Administrators can easily configure user accounts, define access policies, modify restrictions and access scopes to limit access to underlying data, control user actions, and manage database users. Data backups and snapshots. DBMS can simplify the database backup process through a simpler and straightforward interface for managing backups and snapshots. For safekeeping, users can move these backups to third-party locations, such as cloud storage. Performance tuning. DBMS can monitor database performance using integrated tools. Users can tune databases by creating optimized indexes to reduce I/O usage and optimize SQL queries for the best database performance. Data recovery. DBMS provides a recovery platform and the necessary tools to fully or partially restore databases to their previous state—effortlessly. Database query language and APIs. Access and use data via a variety of query languages and API connections. Data dictionary management. Dictionaries include metadata about the structure of the data and relationships between data points so that functionality can rely on structural abstractions rather than complex coding. Data transformation and display. DBMS transforms data on command, such as assembling attributes for the month, day and year as December 14, 2024, or 12/14/24 or another specified display format. Management of data integrity. DBMS establishes and maintains data consistency and minimizes duplications. User access. This policy permits more than one user to access the database at a time and follows ACID to accommodate multiple users. User interface. Whether accessing data through a web form, a direct dashboard, or a third-party distributed network, a browser-based interface makes it easy. 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.) How does DBMS work? The various DBMS components work together to create an integrated system for structuring and storing data, supporting user queries and access, ensuring consistency and integrity, control, security, backups, and logging. The following DBMS schematic illustrates how a DBMS system works: What are the components of a DBMS? 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 of a DBMS that are universal across all database software: 1. Storage engine in a database The database 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. Which storage engine is the best for a database? The right storage engine depends on your data model. SQL engines supporting transactions work well with relational databases. Non-relational models, especially those that require scalability, work best with MongoDB or Cassandra. 2. Database query language What is a database access 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. What are the 4 types of DBMS languages? 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. 3. Query processor The query processor is the intermediary between user queries and the database. In DBMS, query processing is the process of interpreting user queries, such as SQL, and making them actionable commands that the database can understand to perform the appropriate functionality. What are the components of the query processor? The query processor components each work together to extract data. Parser. This component translates a user query into a database language such as SQL, parses it for correct syntax, and verifies its logical meaning. Optimizer. This component converts the query into logical relational operations, identifies how much time and energy it will take to execute the query, and then specifies the exact operations and sequence for the most efficient execution. Execution engine. This is the component that carries out the query, implements algorithms and operators according to the optimized plan, and finally retrieves and formats the results. Query cache. Some systems include a component that stores frequently executed queries and results to save time and improve performance. 4. Optimization engine in DBMS 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. 5. Metadata catalog A metadata catalog, also referred to as a data catalog, 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 6. Log manager The log manager is a component that 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.) 7. Reporting & monitoring tools Reporting and monitoring tools are another standard component that comes with a DBMS. DBMS reporting tools will enable users to generate reports while monitoring tools enable monitoring the databases for resource consumption, user activity, etc. 8. 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. Scale operational effectiveness with an artificial intelligence for IT operations. Learn more about AIOps with BMC! › What are the different types of DBMS? The evolution of data models, how data is structured, and the use cases of each has led to various types of DBMS. The most commonly used are: 1. Relational database management systems (RDBMS) Relational Database Management Systems are the most common type of DBMS. Relational databases 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. Some popular examples of RDBMS include: Microsoft SQL MySQL Oracle Database MariaDB PostgreSQL 2. NoSQL databases NoSQL (nonrelational) databases are designed for semi-structured and unstructured data. They offer greater data modeling flexibility and often don't use a schema. They also support scaling across distributed systems. Examples of nonrelational or NoSQL databases include: MongoDB Azure Cosmos DB Apache Cassandra CouchDB Amazon DynamoDB 3. Object-oriented DBMS (OODBMS) This type of database stores data and data relationships as objects that can be used by object-oriented programming languages like C++ and Java in applications such as CAD systems, databases containing scientific research, and multimedia. Examples of object-oriented databases include: ObjectDB Versant GemStone/S Objectivity/DB 4. Hierarchical DBMS This type of database uses tree-like structures to organize data in parent-child relationships. A parent node can have many children and grandchildren, but each child node has only one parent. These DBMSs work well when data has well-defined relationships that can be organized into files and directories. Examples of hierarchical databases include: IBM Information Management System (IMS) RDM Mobile Windows Registry XML data storage 5. Network DBMS This type of database supports complex interconnections in many-to-many data relationships, with records that have multiple and complex links. Examples of databases that use the network model include: IDMS (Integrated Database Management System) Oracle CODASYL 6. Columnar database management systems (CDBMS) As the name suggests, CDMBS 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. What are the advantages of 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 advantages of DBMS: 1. 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. 2. 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. 3. 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. 4. 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. 5. 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 6. 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. Why is DBMS important? Considering the many advantages, DBMS is essential for any organization when managing databases. 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. The scale, complexity, and feature set of a DBMS will depend on the specific DBMS and the organization's requirements. Related reading These postings are my own and do not necessarily represent BMC's position, strategies, or opinion. See an error or have a suggestion? Please let us know by emailing . Award Spotlight BMC AMI DevX was recognized for its built-in conversational AI, helping mainframe teams troubleshoot, explain code, and move faster with generative AI and plain-language guidance. Jobs Companies Articles Tracker