ADVANTAGES OF DBMS

A Database Management System (DBMS) offers significant advantages over file systems by reducing data redundancy, enhancing data security, and ensuring data integrity through centralized control. Key benefits include improved data sharing, efficient retrieval using indexes, robust backup/recovery options, and support for concurrent, multi-user access.
Here are the primary advantages of a DBMS:

- **Reduced Data Redundancy and Inconsistency:** A DBMS minimizes data duplication by storing data in one place, reducing inconsistencies, improving storage efficiency, and ensuring that updates are reflected across all views.
- **Enhanced Data Security:** DBMS allows for strict access control, ensuring that only authorized users can access or modify specific data, mitigating data leaks and unauthorized access.
- **Improved Data Integrity:** By enforcing data validation rules and constraints (integrity constraints), a DBMS ensures the accuracy and reliability of stored data.
- **Concurrent Access and Transactions:** Multiple users can access the database simultaneously without compromising data consistency, thanks to concurrency control mechanisms.
- **Easy Data Retrieval and Management:** A DBMS uses sophisticated query languages (like SQL) and indexing techniques to make data access faster, more efficient, and user-friendly.
- **Backup and Recovery:** DBMS systems provide robust, automatic backup and recovery tools, allowing the data to be restored in case of hardware or software failure, ensuring data safety.
- **Data Abstraction and Independence:** A DBMS hides the complex physical storage details from users and provides a conceptual view. It separates the data structure (schema) from the applications that use it, making it easier to modify the structure without changing applications.
- **Data Sharing:** Data can be shared across multiple applications and users, promoting better collaboration within an organization.
- **Scalability and Flexibility:** Modern systems, especially those hosted on cloud platforms (AWS, Azure, Google Cloud), offer on-demand scalability to handle increasing data volume.

FEATURES OF DBMS

A Database Management System (DBMS) is software that manages data, providing features like ACID compliance for reliability, reduced data redundancy, concurrent multi-user access, and enhanced security. Key features also include data independence, easy backup/recovery, and support for complex data relationships.
Here are the key features of a DBMS:

- **ACID Properties:** Ensures reliability in database transactions (Atomicity, Consistency, Isolation, Durability).
- **Minimized Redundancy:** Reduces data duplication, which ensures data consistency and saves storage space.
- **Concurrent Multi-User Access:** Allows multiple users to access and manipulate data simultaneously without conflicts.
- **Data Security:** Restricts access to sensitive data and provides security mechanisms, such as authorized user access.
- **Backup and Recovery:** Provides built-in mechanisms for automatic backup and recovery to protect data against failure.
- **Data Integrity:** Maintains the accuracy and consistency of data through constraints and validation rules.
- **Data Independence:** Separates data structure from application programs, allowing changes to the data structure without modifying applications.
- **Multiple Views of Data:** Allows different users to have customized views of the database based on their needs.
- **Query Language Support:** Provides languages (like SQL) for easy insertion, deletion, updating, and retrieval of data.
- **Data Relationships:** Manages complex data relationships and structures efficiently.

COMPONENTS OF DBMS

Organizations produce and gather data as they operate. Contained in a database, data is typically organized to model relevant aspects of reality in a way that supports processes requiring this information. Knowing how this can be managed effectively is vital to any organization.

What is a Database Management System (or DBMS)?

Organizations employ Database Management Systems (or DBMS) to help them effectively manage their data and derive relevant information out of it. A DBMS is a technology tool that directly supports efficient data management. It is a package designed to define, manipulate, and manage data in a database.

Some general functions of a DBMS:

- Designed to allow the definition, creation, querying, update, and administration of databases
- Define rules to validate the data and relieve users of framing programs for data maintenance
- Convert an existing database, or archive a large and growing one
- Run business applications, which perform the tasks of managing business processes, interacting with end-users and other applications, to capture and analyze data

Some well-known DBMSs are Microsoft SQL Server, Microsoft Access, Oracle, SAP, and others.

Components of DBMS

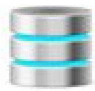
DBMS have several components, each performing very significant tasks in the database management system environment. Below is a list of components within the database and its environment.

Software

This is the set of programs used to control and manage the overall database. This includes the DBMS software itself, the Operating System, the network software being used to share the data among users, and the application programs used to access data in the DBMS.

Hardware

Consists of a set of physical electronic devices such as computers, I/O devices, storage devices, etc., this provides the interface between computers and the real world systems.



Data

DBMS exists to collect, store, process and access data, the most important component. The database contains both the actual or operational data and the metadata.

Procedures

These are the instructions and rules that assist on how to use the DBMS, and in designing and running the database, using documented procedures, to guide the users that operate and manage it.

Database Access Language

This is used to access the data to and from the database, to enter new data, update existing data, or retrieve required data from databases. The user writes a set of appropriate commands in a database access language, submits these to the DBMS, which then processes the data and generates and displays a set of results into a user readable form.

Query Processor

This transforms the user queries into a series of low level instructions. This reads the online user's query and translates it into an efficient series of operations in a form capable of being sent to the run time data manager for execution.

Run Time Database Manager

Sometimes referred to as the database control system, this is the central software component of the DBMS that interfaces with user-submitted application programs and queries, and handles database access at run time. Its function is to convert operations in user's queries. It provides control to maintain the consistency, integrity and security of the data.

Data Manager

Also called the cache manger, this is responsible for handling of data in the database, providing a recovery to the system that allows it to recover the data after a failure.

Database Engine

The core service for storing, processing, and securing data, this provides controlled access and rapid transaction processing to address the requirements of the most demanding data consuming applications. It is often used to create relational databases for online transaction processing or online analytical data processing.

Data Dictionary

This is a reserved space within a database used to store information about the database itself. A data dictionary is a set of read-only table and views, containing the different information about the data used in the enterprise to ensure that database representation of the data follow one standard as defined in the dictionary.

Report Writer

Also referred to as the report generator, it is a program that extracts information from one or more files and presents the information in a specified format. Most report writers allow the user to select records that meet certain conditions and to display selected fields in rows and columns, or also format the data into different charts.

Applications of Database Management Systems

An estimated 2.5 quintillion bytes of data are generated each day. This exponential growth in generated data has enhanced the need to integrate DBMS in almost every aspect of daily life. Important sectors where DBMS finds application include:

- **Reservation Systems**
  Planning travel and making reservations for trains and flights are simplified with DBMS. DBMS applications help manage the schedule and track delays, boarding, and departure of flights or trains.
- **Online Shopping**
  With the accelerated growth of the eCommerce sector, web-based shopping has increased on platforms such as Amazon, eBay, Walmart, etc. Data management systems store and track all the important information related to the orders, such as invoices, shipping details, refund status, etc.
- **Healthcare System**
  DBMSs are of critical importance in the healthcare system as they help organize, track and extract patient information such as appointments, treatment records, payments, doctor schedules, invoices, etc.
- **Finance Sector**
  In addition to its immense application in banking, DBMS is important in storing and managing accounting and finance-related information such as sales reports, financial statements, financial assets, stocks, bonds, etc.
- **HR Management System**

Database management systems help HR executives store and manage employee information such as name, designation, salary, tax, insurance details, etc.

- **Scientific Database**
  Researchers utilize DBMS to store scientific research data, track projects, conduct comparative studies, develop experiment protocols, and more.

3 LEVEL ARCHITECTURE OF DBMS

The 3-Level DBMS Architecture, or ANSI/SPARC model, divides database systems into three distinct layers—**External** (user views), **Conceptual** (logical structure), and **Internal** (physical storage). This structure ensures data independence, allowing physical storage changes without affecting user views, and enhances security by separating user access from data structure.

The Three Levels of Database Architecture

- **External Level (View Schema):** The highest level, describing how individual users or applications view the data. It offers personalized, secure, and specific views of the database, shielding users from the rest of the database structure.
- **Conceptual Level (Logical Schema):** The intermediate level that defines the overall logical structure of the entire database. It describes *what* data is stored (tables, relationships, constraints) and is independent of physical storage details.
- **Internal Level (Physical Schema):** The lowest level, defining *how* the data is actually stored on storage devices. It covers file organization, indexing, data compression, and encryption techniques.

Mappings and Data Independence

The DBMS connects these levels through mapping, which allows changes at one level to be handled without disrupting others.

1. **External/Conceptual Mapping:** Translates user-level requests into the conceptual structure.
2. **Conceptual/Internal Mapping:** Translates the logical structure into physical storage requests.

Benefits of 3-Level Architecture:

- **Physical Data Independence:** You can change physical storage (e.g., switching hard drives) without affecting the logical design.
- **Logical Data Independence:** You can change the logical structure (e.g., adding a table) without changing application views.
- **Security:** Users are limited to their specific view.

3-Level vs. 3-Tier

- **3-Level Architecture** refers to the *schema* organization (External/Conceptual/Internal).

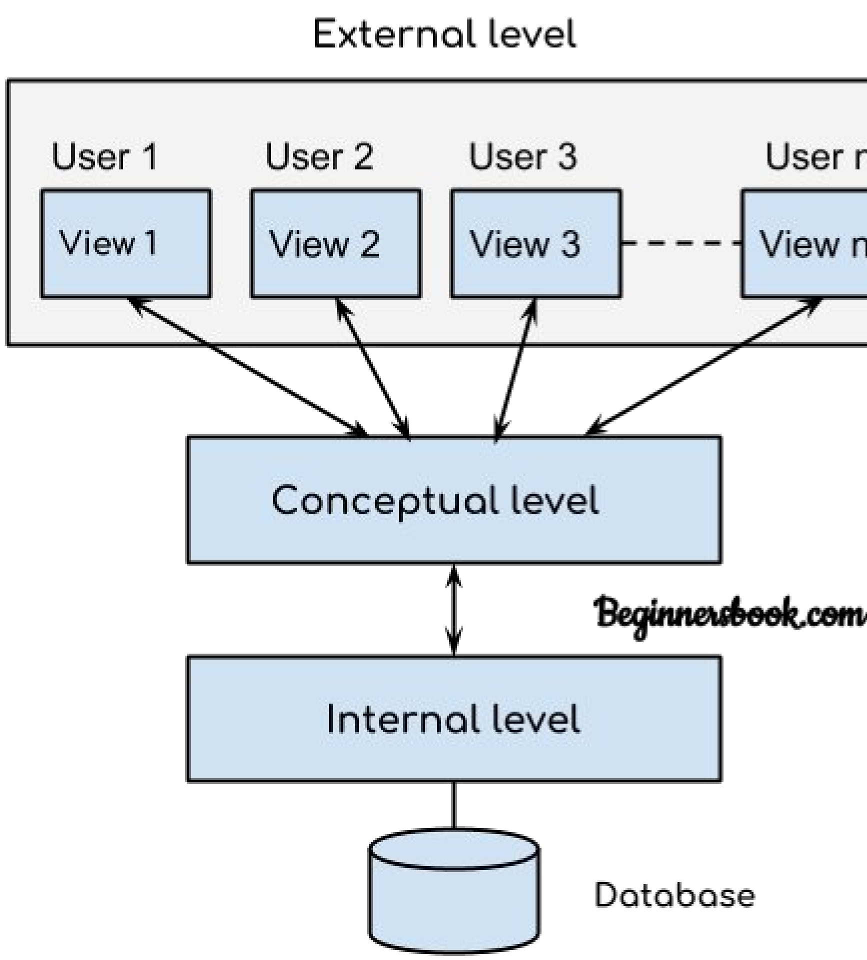
- **3-Tier Architecture** refers to the *physical infrastructure* (Presentation Layer, Application/Business Layer, Database Layer).

DBMS – Three Level Architecture

Data Management

In the previous tutorial we have seen the DBMS architecture – one-tier, two-tier and three-tier. In this guide, we will discuss the three level DBMS architecture in detail.

DBMS Three Level Architecture Diagram



This architecture has three levels:
1. External level
2. Conceptual level
3. Internal level

1. External level

It is also called **view level**. The reason this level is called “view” is because several users can view their desired data from this level which is internally fetched from database with the help of conceptual and internal level mapping.

Data Management

The user doesn’t need to know the database schema details such as data structure, table definition etc. user is only concerned about data which is what returned back to the view level after it has been fetched from database (present at the internal level).

External level is the “**top level**” of the Three Level DBMS Architecture.

2. Conceptual level

It is also called **logical level**. The whole design of the database such as relationship among data, schema of data etc. are described in this level.

Database constraints and security are also implemented in this level of architecture. This level is maintained by DBA (database administrator).

Data Management

3. Internal level

This level is also known as physical level. This level describes how the data is actually stored in the storage devices. This level is also responsible for allocating space to the data. This is the lowest level of the architecture.

CLASSIFICATION OF USERS

DBMS users are classified based on how they interact with the database, ranging from technical administrators managing the system to end-users accessing data. Key roles include Database Administrators (DBAs), Designers, Application Programmers, and End Users (Naive, Sophisticated, or Casual). These roles define, manage, build, or use data through applications or direct queries.

Key Classifications of DBMS Users:

- **Database Administrator (DBA):** Manages the entire DBMS, including security, performance, user access, and backups.
- **Database Designers:** Define the database structure (tables, relationships, constraints).
- **Application Programmers:** Develop tools and interfaces (APIs) for users to interact with the database.
- **End Users:** These are the primary users who interact with the data. They are classified as:
  - **Naive/Parametric Users:** Use standard, pre-written applications (e.g., bank tellers, ATM users).
  - **Sophisticated Users:** Create complex queries, interact directly with the database, and use tools for analysis.

- **Casual Users/Temporary Users:** Access the database occasionally and often need different information each time.

- **System Analysts:** Analyze user requirements and design the information systems.
- **Specialized Users:** Develop custom, specialized applications (e.g., AI or CAD systems).

Classifications by System Access:

- **Single-User Systems:** Support only one user at a time.
- **Multi-User Systems:** Support multiple users working simultaneously.