Back To Course

LIVE BATCHES

Learn

Classroom

Theory

Quiz

Overview Learn Quiz

Filter

We have combined Classroom and Theory tab and created a new Learn tab for easy access. You can access Classroom and Theory from the left panel.

- DBMS | Introduction, Evolution and Advantages

Important Terminologies

Database: Database is a collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc. For Example, university database organizes the data about students, faculty, and admin staff etc. which helps in efficient retrieval, insertion and deletion of data from it.

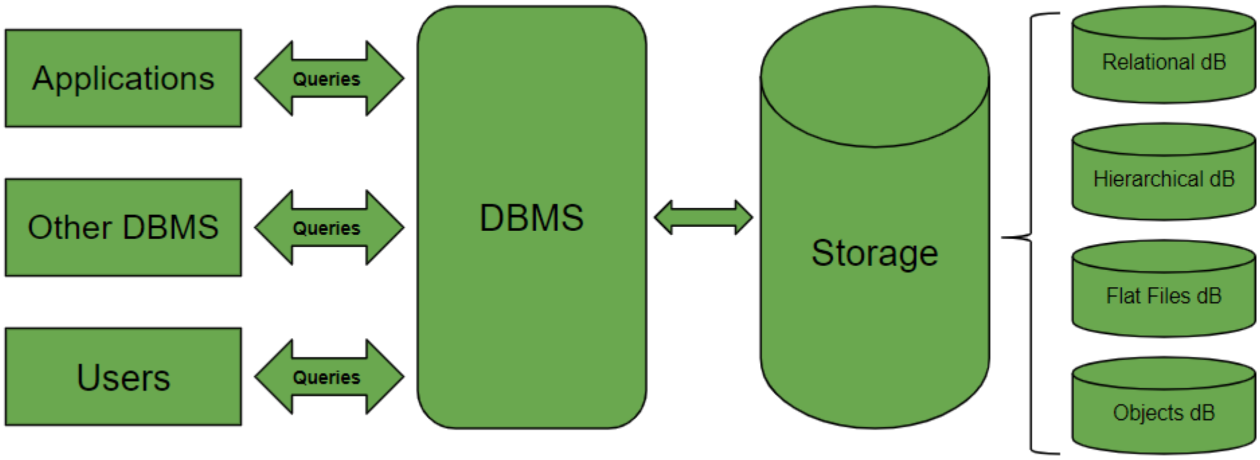
Database Management System: The software which is used to manage the database is called Database Management System (DBMS). For Example, MySQL, Oracle, etc. are popular commercial DBMS used in different applications. DBMS allows users the following tasks:

Data Definition: It helps in creation, modification, and removal of definitions that define the organization of data in the database.

Data Updation: It helps in insertion, mod ▲ and deletion of the actual data in the database.

Data Retrieval: It helps in retrieval of data from the database which can be used by applications for various purposes.

User Administration: It helps in registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control and recovering information corrupted by unexpected failure.



The history of DBMS evolved in three primary phases:

1. File-based System

2. Relational DBMS

3. NoSQL

Paradigm Shift from File System to DBMS

A File Management system is a DBMS that allows access to single files or tables at a time. In a File System, data is directly stored in a set of files. It contains flat files that have no relation to other files (when only one table is stored in a single file, then this file is known as flat file). File System manages data using files in the hard disk. Users are allowed to create, delete, and update the files according to their requirements. Let us consider the example of a file-based University Management System. Data of students is available to their respective Departments, Academics Section, Result Section, Accounts Section, Hostel Office, etc. Some of the data is common for all sections like Roll No, Name, Father Name, Address and Phone number of students but some data is available to a particular section only like Hostel allotment number which is a part of hostel office. Let us discuss the issues with this system:

- **Redundancy of data:** Data is said to be redundant if same data is copied at many places. If a student wants to change a Phone number, he has to get it updated at various sections. Similarly, old reco ▲ st be deleted from all sections representing that student.

- **Inconsistency of Data:** Data is said to be inconsistent if multiple copies of the same data do not match with each other. If a Phone number is different in the Accounts Section and Academics Section, it will be inconsistent. Inconsistency may be because of typing errors or not updating all copies of same data.
- **Difficult Data Access:** A user should know the exact location of the file to access data, so the process is very cumbersome and tedious. If a user wants to search student hostel allotment number of a student from 10000 unsorted students' records, how difficult it can be.
- **Unauthorized Access:** File System may lead to unauthorized access to data. If a student gets access to file having his marks, he can change it in unauthorized way.
- **No Concurrent Access:** The access of same data by multiple users at same time is known as concurrency. File system does not allow concurrency as data can be accessed by only one user at a time.
- **No Backup and Recovery:** File system does not incorporate any backup and recovery of data if a file is lost or corrupted.

These are the main reasons which made a shift from file system to Relational DBMS.

Relational DBMS

Relational database means the data is stored as well as retrieved in the form of relations (tables). The table below shows the relational database with only one relation called **STUDENT** which stores **ROLL_NO**, **NAME**, **ADDRESS**, **PHONE** and **AGE** of students.

STUDENT

ROLL_NO	**NAME**	**ADDRESS**	**PHONE**	**AGE**
1	RAM	DELHI	9455123451	18
2	RAMESH	GURGAON	9652431543	18
3	SUJIT	ROHTAK	9156253131	20
4	SURESH	DELHI	9156768971	18

These are some important terminologies that are used in terms of relationships that we will learn later. The relational DBMS provides us with Structured Query Language or SQL which is a standard Database language that is used to create, maintain and retrieve the relational database. Following are some interesting facts about SQL.

- SQL is case insensitive. But it is a recommended practice to use keywords (like SELECT, UPDATE, CREATE, etc.) in capital letters and use user defined thi

- We can write comments in SQL using "–" (double hyphen) at the beginning of any line.

- SQL is the programming language for relational databases (explained below) like MySQL, Oracle, Sybase, SQL Serve ▲ re, etc. Other non-relational databases (also called NoSQL) databases like MongoDB, DynamoDB, etc do not use SQL

- Although there is an ISO standard for SQL, most of the implementations slightly vary in syntax. So we may encounter queries that work in SQL Server but do not work in MySQL.

NoSQL

A NoSQL originally referring to non SQL or non-relational is a database that provides a mechanism for storage and retrieval of data. This data is modelled in means other than the tabular relations used in relational databases. NoSQL databases are used in real-time web applications and big data and their use are increasing over time. NoSQL systems are also sometimes called Not only SQL to emphasize the fact that they may support SQL-like query languages. A NoSQL database includes simplicity of design, simpler horizontal scaling to clusters of machines and finer control over availability. The data structures used by NoSQL databases are different from those used by default in relational databases which makes some operations faster in NoSQL. The suitability of a given NoSQL database depends on the problem it should solve. Data structures used by NoSQL databases are sometimes also viewed as more flexible than relational database tables. Types of NoSQL databases and the name of the databases system that falls in that category are:

1. MongoDB falls in the category of NoSQL document-based database.

2. Key value store: Memcached, Redis, Coherence

3. Tabular: Hbase, Big Table, Accumulo

4. Document based: MongoDB, CouchDB, Cloudant

Advantages of DBMS

DBMS helps in efficient organization of data in database which has following advantages over typical file system.

- **Minimized redundancy and data consistency:** Data is normalized in DBMS to minimize the redundancy which helps in keeping data consistent. For Example, student information can be kept at one place in DBMS and accessed by different users.
- **Simplified Data Access:** A user need only name of the relation not exact location to access data, so the process is very simple.
- **Multiple data views:** Different views of same data can be created to cater the needs of different users. For Example, faculty salary information can be hidden from student view of data but shov ▲ min view.

- **Data Security:** Only authorized users are allowed to access the data in DBMS. Also, data can be encrypted by DBMS which makes it secure.
- **Concurrent access to data:** Data can be accessed concurrently by different users at same time in DBMS.
- **Backup and Recovery mechanism:** DBMS backup and recovery mechanism helps to avoid data loss and data inconsistency in case of catastrophic failures.

Report An Issue

If you are facing any issue on this page. Please let us know.

GeeksforGeeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Company

About Us

Careers

Privacy Policy

Contact Us

Terms of Service

Learn

Algorithms

Data Structures

Languages

CS Subjects

Video Tutorials

▲

Practice

Courses

Company-wise

Topic-wise

How to begin?

Contribute

Write an Article

Write Interview Experience

Internships

Videos

LIVE BATCHES

▲