Data

Data is the raw material that can be processed for any computing machine.

For example − Employee name, Product name, Name of the student, Marks of the student, Mobile number, Image etc.

Information

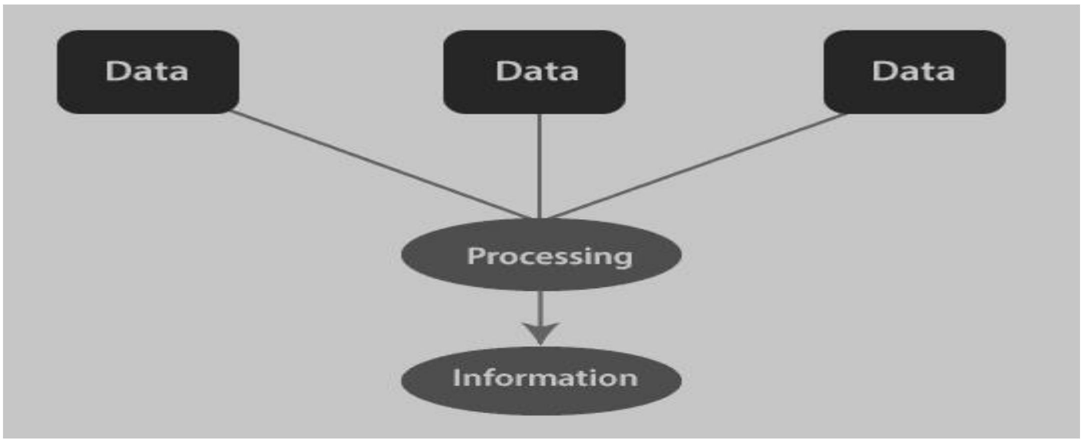
Information is the data that has been converted into more useful or intelligent form.

For example: Report card sheet.

The information is needed for the following reasons −

- To gain knowledge about the surroundings.
- To keep the system up to date.
- To know about the rules and regulations of the society.

Data	Information
Data is the raw fact.	It is a processed form of data.
It is not significant to a business.	It is significant to a business.
Example: Product name, Name of student.	Example: Report card of student.
It is a phenomenal fact.	It is organized data.
This is the primary level of intelligence.	It is a secondary level of intelligence.
May or may not be meaningful.	Always meaningful.
Understanding is difficult.	Understanding is easy.

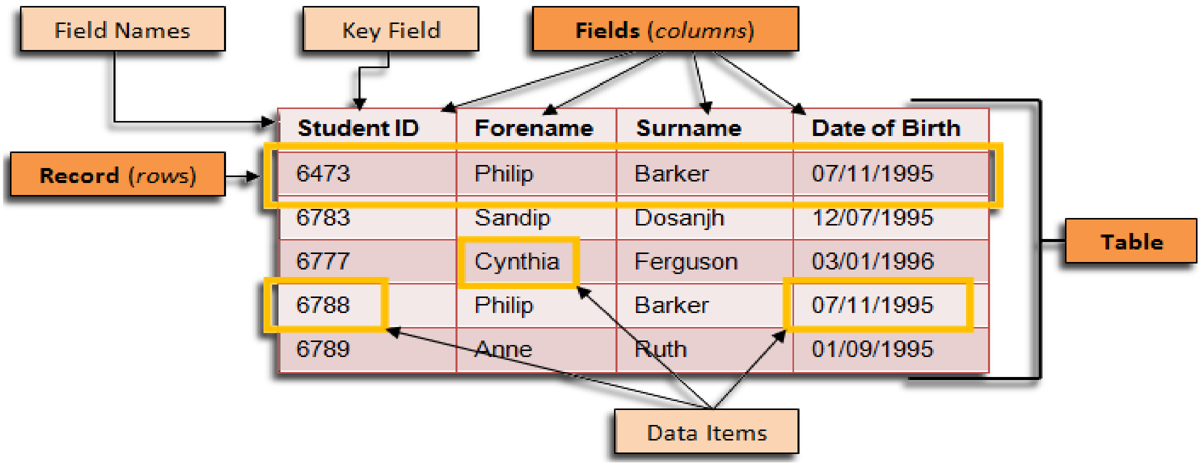


Field

A field is single unit of data that is unique within each entry/row, but the overall data category is common to all entries. For instance, "address" is a field that is common to all named entries in the phone book; however, the content of the address field will be unique for each named entry.

Record

A record is a row of a database, a horizontal grouping of fields; the content of those fields is unique to that row. In the phone book example, each last name begins a record/row which contains data in the name, address and phone number fields. A record is often used in a database search because each record has a unique, identifying quality (or value). A database record is, basically, a row that contains unique data in each of the fields. A database will usually contain a large number of records but only a small number of fields.



Student ID	Forename	Surname	Date of Birth
6473	Philip	Barker	07/11/1995
6783	Sandip	Dosanjh	12/07/1995
6777	Cynthia	Ferguson	03/01/1996
6788	Philip	Barker	07/11/1995
6789	Anne	Ruth	01/09/1995

Traditional File System

File system is collection of data. In this system, user has to write procedures for managing database. It provides details of data representation and storage of data. In this –

- Data is stored in files.
- Each file has specific format.
- Programs that use these files depend on knowledge about that format.

Advantages of Traditional File System:

- File processing cost less and can be more speed than database.
- Companies mainly use file processing to handle large volumes of structured data on a regular basis.
- It can be more efficient and cost less than DBMS in certain situations.
- Design is simple.
- Customization is easy and efficient.

Disadvantages of Traditional File System:

- Data redundancy and inconsistency.
- Difficulty in accessing data.
- Integrity problems
- Unauthorized access is not restricted.

What is a Database?

A database is an organized collection of data, so that it can be easily accessed and managed. We can organize data into tables, rows, columns, and index it to make it easier to find relevant information.

The main purpose of the database is to operate a large amount of information by storing, retrieving, and managing data.

What is DBMS?

- Database management system is a software which is used to manage the database. For example: MySQL, Oracle, etc are a very popular commercial database which is used in different applications.
- DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more.
- It provides protection and security to the database. In the case of multiple users, it also maintains data consistency.

DBMS allows users the following tasks:

- **Data Definition:** It is used for creation, modification, and removal of definition that defines the organization of data in the database.
- **Data Updation:** It is used for the insertion, modification, and deletion of the actual data in the database.
- **Data Retrieval:** It is used to retrieve the data from the database which can be used by applications for various purposes.
- **User Administration:** It is used for registering and monitoring users, maintain data integrity, enforcing data security, dealing with concurrency control, monitoring performance and recovering information corrupted by unexpected failure.

Characteristics of DBMS

Here are the characteristics and properties of Database Management System:

- Provides security and removes redundancy
- Support of multiple views of the data
- Sharing of data and multiuser transaction processing
- Database Management Software allows entities and relations among them to form tables.
- It follows the ACID concept (Atomicity, Consistency, Isolation, and Durability).
- DBMS supports multi-user environment that allows users to access and manipulate data in parallel.

Advantages of DBMS

- **Controls database redundancy:** It can control data redundancy because it stores all the data in one single database file and that recorded data is placed in the database.
- **Data sharing:** In DBMS, the authorized users of an organization can share the data among multiple users.
- **Easily Maintenance:** It can be easily maintainable due to the centralized nature of the database system.
- **Reduce time:** It reduces development time and maintenance need.
- **Backup:** It provides backup and recovery subsystems which create automatic backup of data from hardware and software failures and restores the data if required.
- **multiple user interface:** It provides different types of user interfaces like graphical user interfaces, application program interfaces

Disadvantages of DBMS

- **Cost of Hardware and Software:** It requires a high speed of data processor and large memory size to run DBMS software.
- **Size:** It occupies a large space of disks and large memory to run them efficiently.
- **Complexity:** Database system creates additional complexity and requirements.
- **Higher impact of failure:** Failure is highly impacted the database because in most of the organization, all the data stored in a single database and if the database is damaged due to electric failure or database corruption then the data may be lost forever.

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, maintaining 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.
HR Management	For information about employees, salaries, payroll, deduction, generation of paychecks, etc.

Need of DBMS

A Database Management System (DBMS) creates and manages databases. With the help of DBMS, programmers can easily retrieve, create, update, and manage data. There are various functions performed by DBMS, which gives it an upper hand over the traditional file system. Users can also create their personalized database as per requirement.

Need of DBMS

DBMS is useful in the following ways:

1. Ease of Accessing Data

In the file system, different files are created for each user containing which data they can access. Also, in the file system, for the user to extract data, there is a need for code or application. DBMS removes redundancy by granting access to users and decides which and how many parts of data are accessible to them from the database. Users can get easy access to data and can also specify the type of data they want to extract. In DBMS, users through queries can get easy access to data.

2. Storage and Management of Data

Data cannot be stored in the form of objects in the file system. The data in the practical world is generally stored in the form of objects and not files. So, an application is required to map the data into objects for further usage. In DBMS, the data can be directly stored in the form of objects. In DBMS, user can query the database whereas in file system level code is written for handling, saving, and scanning of data.

3. Easy and Efficient File Management

In the file system, the entire database runs for every query operation as files are indexed. It takes a lot of time compared to DBMS, where objects are indexed based on the attribute of data. The complex management of memory becomes easy to handle. With this, retrieval of data is faster than the traditional file system.

4. *Avoiding duplicates and Redundancy*

Redundancy means repetition of the same data. In the file system, the storage of data might take place multiple times. Like, if a student is pursuing two courses in the same institution say English and Science, then his general information might get stored in both the English dept. as well as Science dept. It results in prolonged hours of accessing and storing data. It further results in the inconsistency of data in both departments. Data normalization is used in DBMS to avoid duplicate data.

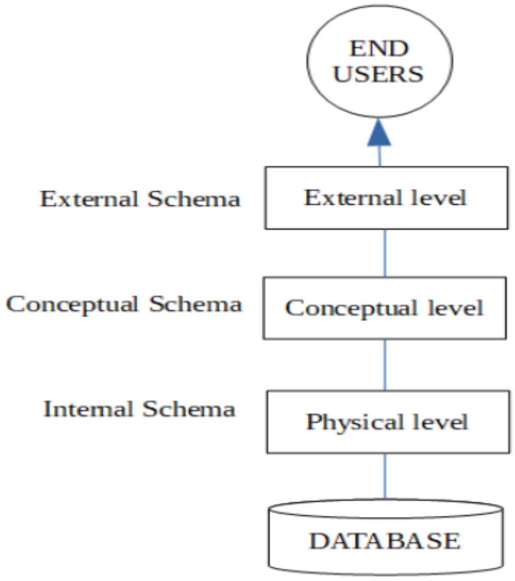
5. *Concurrent Data Accessing*

Users can access data simultaneously through different applications. In the file system, this simultaneous access leads to inconsistency. Let’s take a simple example of depositing money in a bank account. Suppose two depositions of A and B of amounts 100 and 200 are made in an account X which initially contains 1000. Now since these depositions are taking place simultaneously, different depositions update account differently. A reads 1000, credits 100, updates the account to 1100. B also reads 1000, credits 200, and updates the account to 1200. Both cases have wrong information regarding the amount in account X. The result is data inconsistency. In DBMS, users can access data concurrently, and a mechanism is provided to deal with this kind of inconsistency. DBMS uses the ACID approach. ACID stands for atomicity, durability, isolation, and consistency, which ensure efficient transactions without any corruption of data.

6. *Database Integrity*

Any data stored in the database needs to satisfy integrity constraints. For example, a database contains designations of various employees at a company say HR, account specialist, engineer, analyst, project manager, etc. Then we have a schema of employees working under these designations. A database management system ensures that the employee comes under only one out of the listed job profiles in the database. It helps in the preservation of database integrity.

Three Level Architecture of Database



The details of these levels are as follows −

Physical Level

This is the lowest level in the three level architecture. It is also known as the internal level. The physical level describes how data is actually stored in the database. In the lowest level, this data is stored in the external hard drives in the form of bits and at a little high level, it can be said that the data is stored in files and folders. The physical level also discusses compression and encryption techniques.

This level is also responsible for allocating space to the data. This is the lowest level of the architecture.

Conceptual Level

The conceptual level is at a higher level than the physical level. It is also known as the 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).

External Level

This is the highest level in the three level architecture and closest to the user. 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.

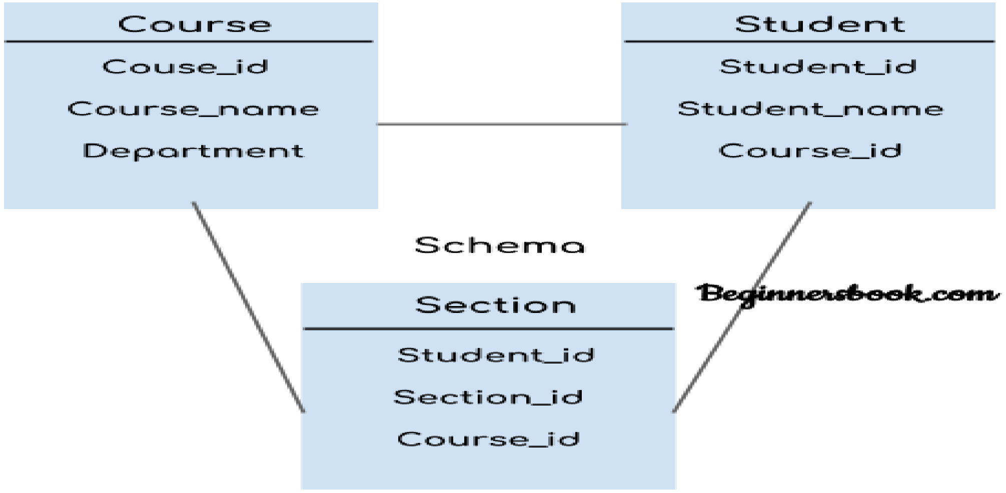
The user doesn’t need to know the database schema details such as data structure, table definition etc. user is only concerned about data which is what returned back to the view level after it has been fetched from database (present at the internal level).

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

DBMS Schema

Definition of schema: Design of a database is called the schema. Schema is of three types: Physical schema, logical schema and view schema.

For example: In the following diagram, we have a schema that shows the relationship between three tables: Course, Student and Section. The diagram only shows the design of the database, it doesn’t show the data present in those tables. Schema is only a structural view (design) of a database as shown in the diagram below.



The design of a database at physical level is called **physical schema**, how the data stored in blocks of storage is described at this level.

Design of database at logical level is called **logical schema**, programmers and database administrators work at this level, at this level data can be described as certain types of data records gets stored in data structures, however the internal details such as implementation of data structure is hidden at this level (available at physical level).

Design of database at view level is called **view schema**. This generally describes end user interaction with database systems. To learn more about these schemas, refer 3 level data abstraction architecture.

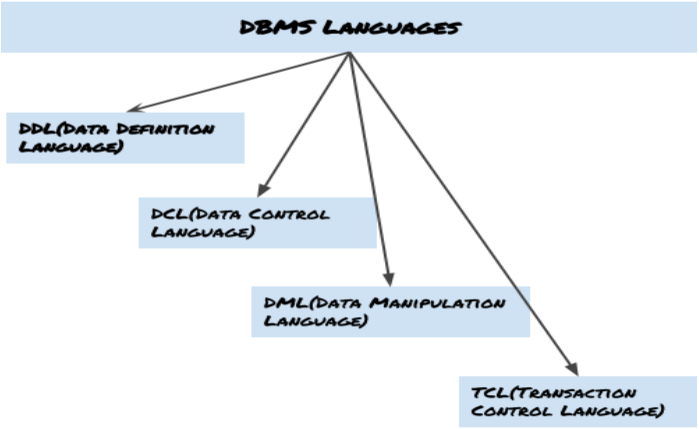
DBMS Instance

Definition of instance: The data stored in database at a particular moment of time is called instance of database.
For example, let's say we have a single table student in the database, today the table has 100 records, and so today the instance of the database has 100 records. Let's say we are going to add another 100 records in this table by tomorrow so the instance of database tomorrow will have 200 records in table. In short, at a particular moment the data stored in database is called the instance that changes over time when we add or delete data from the database.

DBMS languages

Database languages are used to read, update and store data in a database. There are several such languages that can be used for this purpose; one of them is SQL (Structured Query Language).

Types of DBMS languages:


DBMS Languages
DDL(Data Definition Language)
DCL(Data Control Language)
DML(Data Manipulation Language)
TCL(Transaction Control Language)

Data Definition Language (DDL)
DDL is used for specifying the database schema. It is used for creating tables, schema, indexes, constraints etc. in database. Let's see the operations that we can perform on database using DDL:

- To create the database instance – CREATE
- To alter the structure of database – **ALTER**
- To drop database instances – DROP
- To delete tables in a database instance – **TRUNCATE**
- To rename database instances – **RENAME**
- To drop objects from database such as tables – **DROP**
- To Comment – **Comment**

All of these commands either defines or update the database schema that's why they come under Data Definition language.

Data Manipulation Language (DML)
DML is used for accessing and manipulating data in a database. The following operations on database come under DML:

- To read records from table(s) – SELECT
- To insert record(s) into the table(s) – **INSERT**
- Update the data in table(s) – UPDATE
- Delete all the records from the table – DELETE

Data Control language (DCL)
DCL is used for granting and revoking user access on a database –

- To grant access to user – GRANT
- To revoke access from user – REVOKE

In practical data definition language, data manipulation language and data control languages are not separate language; rather they are the parts of a single database language such as SQL.

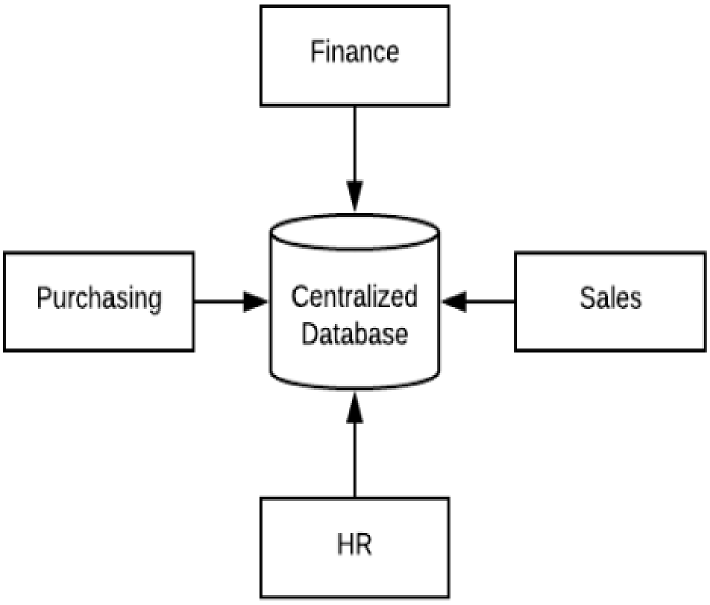
Transaction Control Language (TCL)

The changes in the database that we made using DML commands are either performed or rollbacked using TCL.

- To persist the changes made by DML commands in database – COMMIT
- To rollback the changes made to the database – ROLLBACK

Centralized Database Management System

A centralized database is stored at a single location such as a mainframe computer. It is maintained and modified from that location only and usually accessed using an internet connection such as a LAN or WAN. The centralized database is used by organisations such as colleges, companies, banks etc.



As can be seen from the above diagram, all the information for the organisation is stored in a single database. This database is known as the centralized database.

Advantages

Some advantages of Centralized Database Management System are −

- The data integrity is maximised as the whole database is stored at a single physical location. This means that it is easier to coordinate the data and it is as accurate and consistent as possible.

- The data redundancy is minimal in the centralised database. All the data is stored together and not scattered across different locations. So, it is easier to make sure there is no redundant data available.

- Since all the data is in one place, there can be stronger security measures around it. So, the centralised database is much more secure.

- Data is easily portable because it is stored at the same place.

- The centralized database is cheaper than other types of databases as it requires less power and maintenance.

- All the information in the centralized database can be easily accessed from the same location and at the same time.

Disadvantages

Some disadvantages of Centralized Database Management System are −

- Since all the data is at one location, it takes more time to search and access it. If the network is slow, this process takes even more time.

- There is a lot of data access traffic for the centralized database. This may create a bottleneck situation.

- Since all the data is at the same location, if multiple users try to access it simultaneously it creates a problem. This may reduce the efficiency of the system.

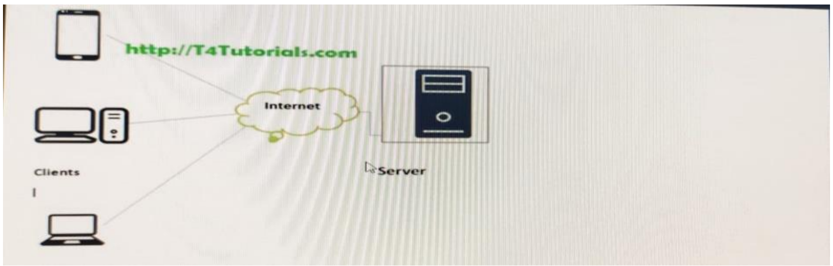
- If there are no database recovery measures in place and a system failure occurs, then all the data in the database will be destroyed.

Client-server Database Architecture in DBMS

In client-server architecture many clients connected with one server. The server is centerlines.it provides services to all clients. All clients request to the server for different Service. The server displays the results according to the client's request.

Client/server architecture is a computing model in which the server hosts (computer), send and manages most of the resources and works to be required by the client. In this type of architecture has one or more client computers attached to a central server over a network. This system shares different resources.

Client/server architecture is also called as a networking computing model and client-server network because all the requests and demands are sent over a network.



Working of Client-server Database Architecture in DBMS

Basically client-server model defines how the server provides a service to clients Server is a centralized computer that provides services to all attach clients.

For example file server, web server, etc. each the basic work of server to provide services to each client. The client can be a laptop computer, tablets, and smartphones, etc.

The server has many types of relationship with clients. Many servers have one too many relationships with clients.in one too many relationships many clients connected with one server.

When one client wants to communicate with the server. The server may be accepted or rejects the request of clients. When the server computer accepts the request of clients than server maintains a connection according to a defined protocol. The protocol rules over the network.

Advantages of Client-server Database Architecture in DBMS

1. All the data and resources are controlled by server .In this way all data and resources are very consistent.
2. You can easily increase the number of client in this architecture at any time. This all increases the scalability of the network.
3. This is very easy to maintain you can easily repair, replace or add clients in this network.
4. This network is very easy to use and it is not complicated.

Disadvantages of Client-server Database Architecture in DBMS

1. Traffic is a big problem in this network.
2. When you add large numbers of the client with server this network will be more complicated.
3. When the server goes down all the clients are not able to send their request. The whole work will be stopped
4. The hardware and software are very expensive.
5. The client does not have resources for each resource they need to request the server. Because of all resources exit on server

What is Database Architecture?

A **Database Architecture** is a representation of DBMS design. It helps to design, develop, implement, and maintain the database management system. DBMS architecture allows

dividing the database system into individual components that can be independently modified, changed, replaced, and altered. It also helps to understand the components of a database.
A Database stores critical information and helps access data quickly and securely. Therefore, selecting the correct Architecture of DBMS helps in easy and efficient data management.

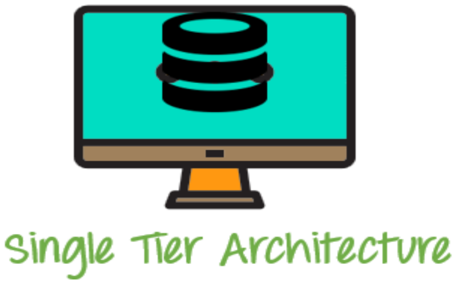
Types of DBMS Architecture

There are mainly three types of DBMS architecture:

- One Tier Architecture (Single Tier Architecture)
- Two Tier Architecture
- Three Tier Architecture

1-Tier Architecture

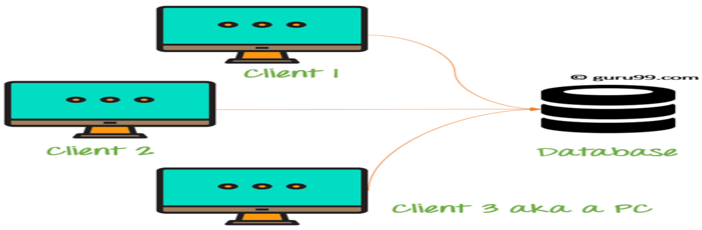
1 Tier Architecture in DBMS is the simplest architecture of Database in which the client, server, and Database all reside on the same machine. A simple one tier architecture example would be anytime you install a Database in your system and access it to practice SQL queries. But such architecture is rarely used in production.



1 Tier Architecture Diagram

2-Tier Architecture

A **2 Tier Architecture** in DBMS is a Database architecture where the presentation layer runs on a client (PC, Mobile, Tablet, etc.), and data is stored on a server called the second tier. Two tier architecture provides added security to the DBMS as it is not exposed to the end-user directly. It also provides direct and faster communication.



2 Tier Architecture Diagram

In the above 2 Tier client-server architecture of database management system, we can see that one server is connected with clients 1, 2, and 3.

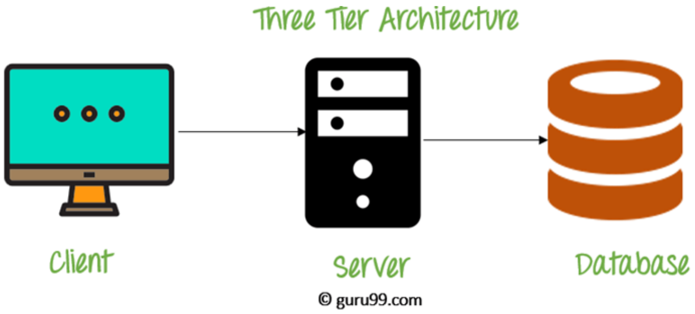
3-Tier Architecture

A **3 Tier Architecture** in DBMS is the most popular client server architecture in DBMS in which the development and maintenance of functional processes, logic, and data access, data storage, and user interface is done independently as separate modules. Three Tier architecture contains a presentation layer, an application layer, and a database server.

3-Tier database Architecture design is an extension of the 2-tier client-server architecture.

3-tier architecture has the following layers:

1. Presentation layer (your PC, Tablet, Mobile, etc.)
2. Application layer (server)
3. Database Server



3 Tier Architecture Diagram

The Application layer resides between the user and the DBMS, which is responsible for communicating the user's request to the DBMS system and send the response from the DBMS to the user. The application layer (business logic layer) also processes functional logic, constraint, and rules before passing data to the user or down to the DBMS.

The goal of Three Tier client-server architecture is:

- To separate the user applications and physical database
- To support DBMS characteristics
- Program-data independence
- Supporting multiple views of the data

Three Tier Architecture Examples:

- Any large website on the internet, including **guru99.com**.

Advantages and disadvantages of DBMS

1. **Redundancy problem can be solved.**

In the File System, duplicate data is created in many places because all the programs have their own files which create data redundancy resulting in wastage of memory. In DBMS, all the files are integrated in a single database. So there is no chance of duplicate data.

For example: A student record in a library or examination can contain duplicate values, but when they are converted into a single database, all the duplicate values are removed.

2. **Has a very high security level.**

Data security level is high by protecting your precious data from unauthorized access. Only authorized users should have the grant to access the database with the help of credentials.

3. **Presence of Data integrity.**

Data integrity makes unification of so many files into a single file. DBMS allows data integrity which makes it easy to decrease data duplicity Data integration and reduces redundancy as well as data inconsistency.

4. **Support multiple users.**

DBMS allows multiple users to access the same database at a time without any conflicts.

In DBMS, data is stored in a single database so data becomes more consistent in comparison to file processing systems.

5. **Shared data**

Data can be shared between authorized users of the database in DBMS. All the users have their own right to access the database. Admin has complete access to the database. He has a right to assign users to access the database.

6. **Enforcement of standards**

As DBMS have central control of the database. So, a DBA can ensure that all the applications follow some standards such as format of data, document standards etc. These standards help in data migrations or in interchanging the data.

7. **Any unauthorized access is restricted**

Unauthorized persons are not allowed to access the database because of security credentials.

8. **Provide backup of data**

Data loss is a big problem for all the organizations. In the file system users have to back up the files in regular intervals which lead to waste of time and resources.

DBMS solves this problem of taking backup automatically and recovery of the database.

Disadvantages of DBMS

The disadvantages of DBMS are as follows:

1. **Complexity**

The provision of the functionality that is expected of a good DBMS makes the DBMS an extremely complex piece of software. Database designers, developers, database administrators and end-users must understand this functionality to take full advantage of it.

Failure to understand the system can lead to bad design decisions, which leads to a serious consequence for an organization.

2. **Size**

The functionality of DBMS makes use of a large piece of software which occupies megabytes of disk space.

3. **Performance**

Performance may not run as fast as desired.

4. **Higher impact of a failure**

The centralization of resources increases the vulnerability of the system because all users and applications rely on the availability of DBMS, the failure of any component can bring operation to halt.

5. **Cost of software and hardware**

It requires a number of high powered processors and large size memory to run DBMS.

Difference between RDBMS and DBMS

S No.	Parameter	DBMS	RDBMS
1	**Definition**	DBMS stands for Database Management System, which includes n number of tables there is no relationship between another tables.	RDMBS stands for Relational Database Management System, which has a relationship with other tables.
2	**Storage**	Data Store as a file with the metadata.	The data will save in the tables.
3	**Client-Server**	It doesn’t support client-server architecture.	It supports client-server architecture.

4	**Support for distributed Database**	DBMS doesn’t support distributed database.	RDBMS supports distributed database.
5	**Data Redundancy**	Data redundancy is expected in this model.	Keys and indexes don’t permit Data redundancy.
6	**Storage Limitation**	There are limits to store records in a one database file.	There are infinite data stored in a single database file.
7	**Number of Users**	It supports a single user at the moment.	It supports more than one user at the moment.
8	**Suited for**	DBMS mostly deals with a small quantity of data.	RDBMS layout controls a large amount of data.
9	**Data Relationship**	The Relationship between tables in DBMS is Physical.	The relationship in RDBMS is Logical due to the foreign keys.
10	**Integrity constraints**	DBMS doesn’t support Data Integrity.	RDBMS support Data integrity.
11	**Index Creation**	Keys and Indexes not used.	In RDBMS, relationship establishment done via keys and indexes.
12	**Database structure**	Data storage is in hierarchical or navigational format (A navigation form is simply **a form that contains a Navigation Control**.).	RDBMS utilizes a tabular structure where the headers are the column names, and the rows include corresponding values.
13	**Hardware and software need**	Low software and hardware requirements.	Higher hardware and software requirements.
14	**Recovery Support**	It doesn’t support recovery.	RDBMS helps in recovery of the database in the event of loss of data.
15	**Examples**	Examples of DBMS are file systems, XML, Microsoft Access, Windows Registry etc.	Example of RDBMS is **MySQL**, SQL Server, oracle etc.