Introduction to Database Management Systems **2nd sem (DBMS)**

Characteristics of Database Approach: The database approach has proven far better than the traditional file management system. Database Approach has many characteristics that make it more robust in nature.

Characteristics of Database Approach

- Represent Some Aspects of real world applications
  A database represents some features of real world applications. Any change in the real world is reflected in the database. If we have some changes in our real applications like railway reservation system then it will be reflected in database too.
  For example, let us take railway reservation system; we have in our mind some certain applications of maintaining records of attendance, waiting list, train arrival and departure time, certain day etc. related to each train.
- Manages Information
  A database always takes care of its information because information is always helpful for whatever work we do. It manages all the information that is required to us. By managing information using a database, we become more deliberated user of our data.
- Easy Operation implementation
  All the operations like insert, delete, update, search etc. are carried out in a flexible and easy way. Database makes it very simple to implement these operations. A user with little knowledge can perform these operations. This characteristic of database makes it more powerful.
- Multiple views of database
  Basically, a view is a subset of the database. A view is defined and devoted for a particular user of the system. Different users of the system may have different views of the same system. Every view contains only the data of interest to a user or a group of users. It is the responsibility of users to be aware of how and where the data of their interest is stored.
- Data for specific purpose
  A database is designed for data of specific purpose. For example, a database of student management system is designed to maintain the record of student's marks, fees and attendance etc. This data has a specific purpose of maintaining student record.
- It has Users of Specific interest
  A database always has some indented group of users and applications in which these user groups are interested.
  For example, in a library system, there are three users, official administration of the college, the librarian, and the students.
- Self-Describing nature
  A database is of self describing nature; it always describes and narrates itself. It contains the description of the whole data structure, the constraints and the variables.
  It makes it different from traditional file management system in which definition was not the part of application program. These definitions are used by the users and DBMS software when needed.

9 Disadvantages of Database Management System (DBMS)

- Cost
  DBMS requires high initial investment for hardware, software and trained staff. A significant investment based upon size and functionality of organization if required. Also organization has to pay concurrent annual maintenance cost.
- Complexity
  A DBMS fulfill lots of requirement and it solves many problems related to database. But all these functionality has made DBMS an extremely complex software. Developer, designer, DBA and End user of database must have complete skills if they want to user it properly. If they don't understand this complex system then it may cause loss of data or database failure.
- Technical staff requirement
  Any organization have many employees working for it and they can perform many others tasks too that are not in their domain but it is not easy for them to work on DBMS. A team of technical staff is required who understand DBMS and company have to pay handsome salary to them too.
- Database Failure
  As we know that in DBMS, all the files are stored in single database so chances of database failure become more. Any accidental failure of component may cause loss of valuable data. This is really a big question mark for big firms.
- Extra Cost of Hardware
  A DBMS requires disk storage for the data and sometimes you need to purchase extra space to store your data. Also sometimes you need to a dedicated machine for better performance of database. These machines and storage space increase extra costs of hardware.
- Size
  As DBMS becomes big software due to its functionalities so it requires lots of space and memory to run its application efficiently. It gains bigger size as data is fed in it.
- Cost of Data Conversion
  Data conversion may require at any time and organization has to take this step. It is unbelievable that data conversion cost is more than the costs of DBMS hardware and machine combined. Trained staff is needed to convert data to new system. It is a key reason that most of the organizations are still working on their old DBMS due to high cost of data conversion.
- Currency Maintenance
  As new threats comes daily, so DBMS requires to updates itself daily. DBMS should be updates according to the current scenario.
- Performance
  Traditional files system was very good for small organizations as they give splendid performance. But DBMS gives poor performance for small scale firms as its speed is slow.

Introduction to Database Management Systems **2nd sem (DBMS)**

10 Advantages of Database Management System (DBMS)

- Minimize Data Redundancy
  In File Processing System, duplicate data is created in many places because all the programs have their own files. This creates data redundancy which in turns wastes labor and space. In Database Management System, all the files are integrated in a single database. The whole data is stored only once at a single place so there is no chance of duplicate data.
  For example: A student record in library or examination can contain duplicate values, but when they are converted into a single database, all the duplicate values are removed.

- Sharing Of Data
  In DBMS, Data can be shared in between authorized user of database. All the users have their own right to access the database up to a level. Database Administration has complete access of database. He can assign users to access the database. Others users are also authorized to access database and also they can share data between them. Many users have same authority to access the database.

- Data Consistency
  DBMS controls data redundancy which in turn controls data consistency. Data consistency means if you want to update data in any files then all the files should not be updated again. As in DBMS, data is stored in a single database so data becomes more consistent in comparison to file processing system. Also updated values are available to all the users immediately.

- Data Integrity
  Data integrity means unification of so many files into a single file. In DBMS data is stored in different tables. A database contains different tables that are linked to each other. Many users feed entries in these tables so it is important to maintain data items and association between data items. DBMS allows data integrity that makes it easy to decrease data duplicity Data integration reduces redundancy as well as data inconsistency.

- Search Capability
  Users of database may require to fetch data from the database. There are numerous queries users may ask about the data. Search speed of the database must be fast to produce quick results. If users execute any query then it is required that he get fastest results from the database. It is an objective of database to maintain flexible search capability.

- Security
  Data security means protecting your precious data from unauthorized access. Data in database should be kept secure and safe to unauthorized modifications. Only authorized users should

have the grant to access the database. There is a username set for all the users who access the database with password so that no other guy can access these information. DBMS always keep database tamperproof, secure and theft free.

- Privacy
  Privacy means up to what extent a user can access the data. It is predetermined by the DBA that who will access the data and up to what level he will be able to access it. Let say when you make a Facebook page then you have the power to give rights to other users that who will be the promoter, editor and admin.

- Simplicity
  Simplicity means to represent the overall logical view of data in a simple and clear manner. DBMS is very simple for its users who use it. All the operations like insert, delete, create and update are very easy to implement.

- Backup and Recovery
  Data loss is a very big problem for all the organizations. In traditional file processing system, a user needs to backup the database after a regular interval of time that wastes lots of time and resources. If the volume of data is large then this process may take a very long time.
  DBMS solves this problem of taking backup again and again because it allows automatic backup and recovery of database. For examples, if a system fails in the middle of any process then DBMS stores the values of that state in which database were before query execution.

- Integrity Constraints
  Constraints are used to store accurate data because there are many users who feed data in database. Data stored in database should always be correct and accurate. DBMS provides the capability to enforce these constraints on database.
  For example, the maximum marks obtained by the students can never be more than 100. Also account balance of Banks like Axis should not be less than 2500 otherwise you will be penalized.

What Is A Database?

The database is a collection of inter-related data which is used to retrieve, insert and delete the data efficiently. It is also used to organize the data in the form of a table, schema, views, and reports, etc.

For example: The college Database organizes the data about the admin, staff, students and faculty etc.

Introduction to Database Management Systems **2nd sem (DBMS)**

DBMS Database Models

A Database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system. While the Relational Model is the most widely used database model, there are other models too:

- Hierarchical Model
- Network Model
- Entity-relationship Model
- Relational Model

Hierarchical Model

This database model organizes data into a tree-like-structure, with a single root, to which all the other data is linked. The hierarchy starts from the Root data, and expands like a tree, adding child nodes to the parent nodes.

In this model, a child node will only have a single parent node.

This model efficiently describes many real-world relationships like index of a book, recipes etc.

Network Model

This is an extension of the Hierarchical model. In this model data is organized more like a graph, and are allowed to have more than one parent node.

In this database model data is more related as more relationships are established in this database model. Also, as the data is more related, hence accessing the data is also easier and fast. This database model was used to map many-to-many data relationships.

This was the most widely used database model, before Relational Model was introduced.

Entity-relationship Model

In this database model, relationships are created by dividing object of interest into entity and its characteristics into attributes.

Different entities are related using relationships.

E-R Models are defined to represent the relationships into pictorial form to make it easier for different stakeholders to understand.

This model is good to design a database, which can then be turned into tables in relational model (explained below).

Let's take an example, if we have to design a School Database, then Student will be an entity with attributes name, age, address etc. As Address is generally complex, it can be another entity with attributes street name, pin code, city etc, and there will be a relationship between them.

Introduction to Database Management Systems **2nd sem (DBMS)**

Relational Model

In this model, data is organized in two-dimensional tables and the relationship is maintained by storing a common field.

This model was introduced by E.F Codd in 1970, and since then it has been the most widely used database model, in fact, we can say the only database model used around the world.

The basic structure of data in the relational model is tables. All the information related to a particular type is stored in rows of that table.

Hence, tables are also known as relations in relational model.

In the coming tutorials we will learn how to design tables, normalize them to reduce data redundancy and how to use Structured Query language to access data from tables.

DBMS Architecture

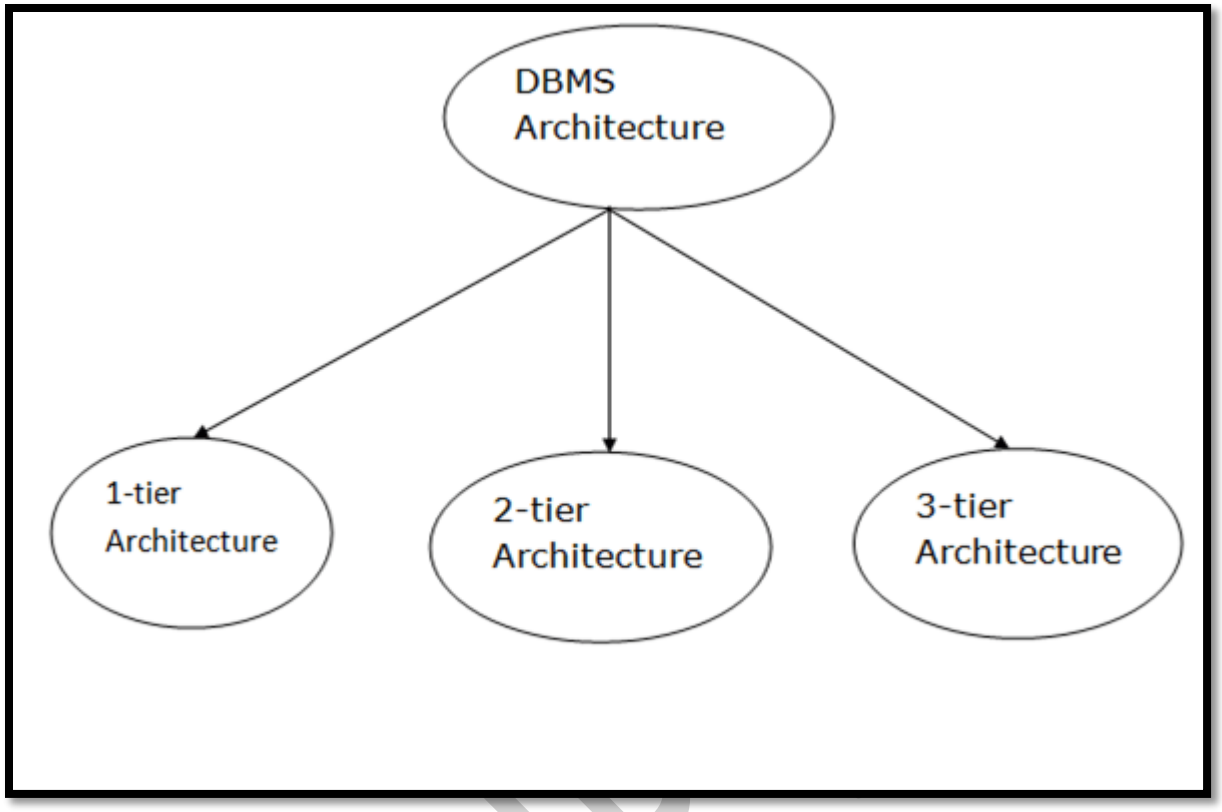
The DBMS design depends upon its architecture. The basic client/server architecture is used to deal with a large number of PCs, web servers, database servers and other components that are connected with networks.

The client/server architecture consists of many PCs and a workstation which are connected via the network.

DBMS architecture depends upon how users are connected to the database to get their request done.

Types of DBMS Architecture

Database architecture can be seen as a single tier or multi-tier. But logically, database architecture is of two types like: 2-tier architecture and 3-tier architecture.



1-Tier Architecture

In this architecture, the database is directly available to the user. It means the user can directly sit on the DBMS and uses it.

Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users.

The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

2-Tier Architecture

The 2-Tier architecture is same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side. For this interaction, API's like: ODBC, JDBC are used.

The user interfaces and application programs are run on the client-side.

The server side is responsible to provide the functionalities like: query processing and transaction management.

To communicate with the DBMS, client-side application establishes a connection with the server side.

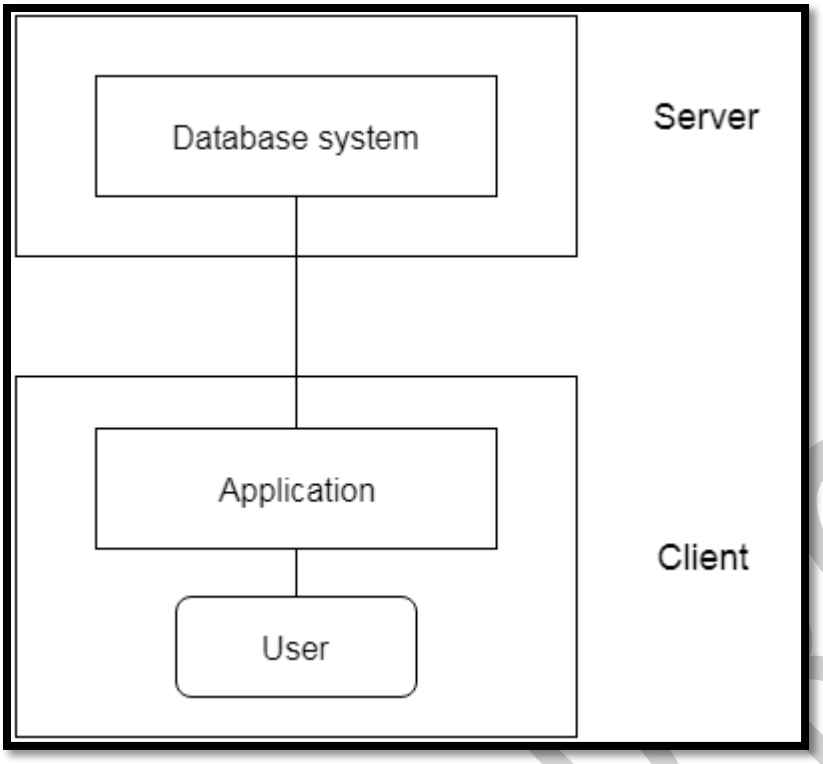


Fig: 2-tier Architecture

3-Tier Architecture

The 3-Tier architecture contains another layer between the client and server. In this architecture, client can't directly communicate with the server.

The application on the client-end interacts with an application server which further communicates with the database system.

End user has no idea about the existence of the database beyond the application server. The database also has no idea about any other user beyond the application.

The 3-Tier architecture is used in case of large web application.

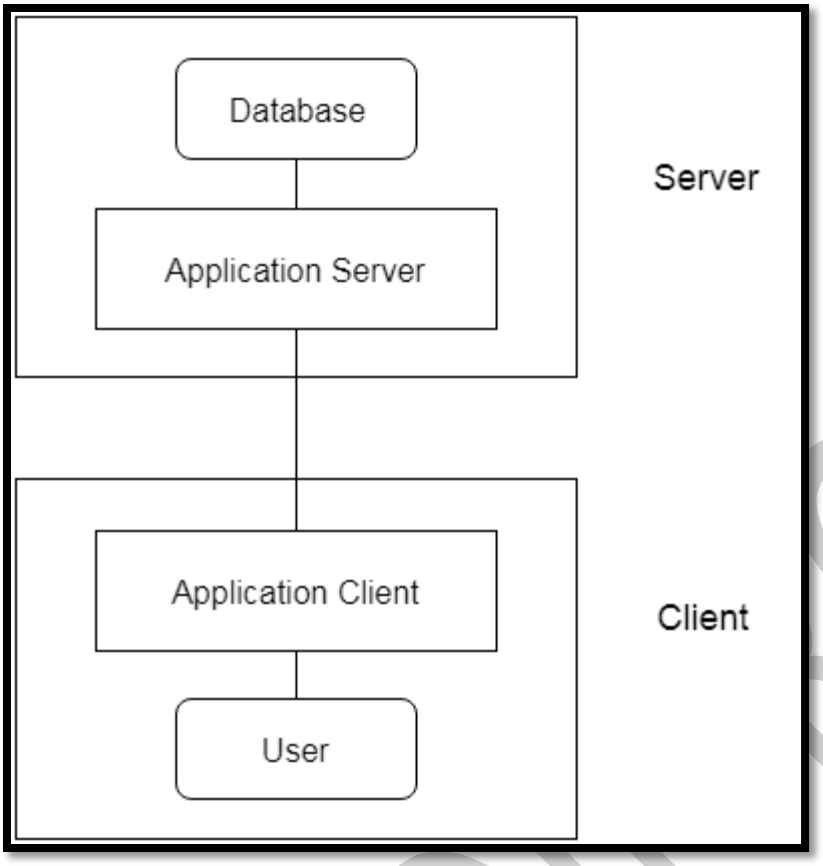


Fig: 3-tier Architecture

What is Data Independence of DBMS?

Data Independence is defined as a property of DBMS that helps you to change the Database schema at one level of a database system without requiring to change the schema at the next higher level. Data independence helps you to keep data separated from all programs that make use of it.

You can use this stored data for computing and presentation. In many systems, data independence is an essential function for components of the system.

Types of Data Independence

In DBMS there are two types of data independence

- Physical data independence
- Logical data independence.

Physical Data Independence

Physical data independence helps you to separate conceptual levels from the internal/physical levels. It allows you to provide a logical description of the database without the need to specify physical structures. Compared to Logical Independence, it is easy to achieve physical data independence.

With Physical independence, you can easily change the physical storage structures or devices with an effect on the conceptual schema. Any change done would be absorbed by the mapping between the conceptual and internal levels. Physical data independence is achieved by the presence of the internal level of the database and then the transformation from the conceptual level of the database to the internal level.

Examples of changes under Physical Data Independence

- Due to Physical independence, any of the below change will not affect the conceptual layer.
- Using a new storage device like Hard Drive or Magnetic Tapes
- Modifying the file organization technique in the Database
- Switching to different data structures.
- Changing the access method.
- Modifying indexes.
- Changes to compression techniques or hashing algorithms.
- Change of Location of Database from say C drive to D Drive

Logical Data Independence

Logical Data Independence is the ability to change the conceptual scheme without changing

- External views
- External API or programs

Any change made will be absorbed by the mapping between external and conceptual levels.

When compared to Physical Data independence, it is challenging to achieve logical data independence.

Examples of changes under Logical Data Independence

Due to Logical independence, any of the below change will not affect the external layer.

- Add/Modify/Delete a new attribute, entity or relationship is possible without a rewrite of existing application programs
- Merging two records into one
- Breaking an existing record into two or more records

Introduction to Database Management Systems **2nd sem (DBMS)**

Difference between Physical and Logical Data Independence

Logica Data Independence	**Physical Data Independence**
Logical Data Independence is mainly concerned with the structure or changing the data definition.	Mainly concerned with the storage of the data.
It is difficult as the retrieving of data is mainly dependent on the logical structure of data.	It is easy to retrieve.
Compared to Logic Physical independence it is difficult to achieve logical data independence.	Compared to Logical Independence it is easy to achieve physical data independence.
You need to make changes in the Application program if new fields are added or deleted from the database.	A change in the physical level usually does not need change at the Application program level.
Modification at the logical levels is significant whenever the logical structures of the database are changed.	Modifications made at the internal levels may or may not be needed to improve the performance of the structure.
Concerned with conceptual schema	Concerned with internal schema
Example: Add/Modify/Delete a new attribute	Example: change in compression techniques, hashing algorithms, storage devices, etc

Importance of Data Independence

- Helps you to improve the quality of the data
- Database system maintenance becomes affordable
- Enforcement of standards and improvement in database security
- You don't need to alter data structure in application programs
- Permit developers to focus on the general structure of the Database rather than worrying about the internal implementation
- It allows you to improve state which is undamaged or undivided
- Database incongruity is vastly reduced.
- Easily make modifications in the physical level is needed to improve the performance of the system.