Advantages of DBMS

1. Data Abstraction

This is one of the most required characteristics of any software system and DBMS works quite efficiently in this field too. It hides the unnecessary details from the end user and just shows what is useful to him for completion of his task. Thus, it also reduces the overhead of hidden complexities and makes it an easy system to work with.

2. Data Redundancy Control

Database Management System also removes the redundancy in the record system efficiently by using the ACID (Atomicity, Consistency, Isolation and Durability) and normalization properties.

3. Minimization of Data Inconsistency

Since the Database Management System reduces the redundancy of the system it also easily makes the system consistent by removing all the inconsistencies.

4. Ease in Data Manipulation

The DBMS uses database manipulative languages such as SQL with commands like SELECT (used for selection of records from one or more tables), INSERT (for insertion of a record into tables), UPDATE (updates the existing information of records) and DELETE (for removing one or more records from a table) to manipulate the information of the database so that it is always easily updated.

5. Data Security

The Database Management System, unlike the former file system comes with a secured login password (separate for admin and the user), through which the user and admin (also for changes) access can be secured up to a large extent.

6. Concurrent Access of Database

Multiple users can access the same database and thus data sharing is made possible easily. Users don’t have to worry about being only at a particular place for using the database; they may use it according to their ease and as per the basic software and hardware requirements of the database.

7. Improved Data Access

The data is stored in a sorted manner which makes the data access very efficient and easy. The user can directly retrieve what he wants to without bothering about the excess and redundant records other than what is required by him. This also saves a lot of time.

8. Data Independence

The data stored in a particular database is totally independent of any other data. We can look for the data of our choice without considering any other data.

Disadvantages of DBMS

1. Increased Cost

In order to have a Database Management System, we need to have a high speed processor along with a large memory size which requires an expensive hardware and hence correspondingly expensive software too.

Also, in order to convert our data into a Database Management System we need to spend a lot which adds on to the cost of the Database Management System.

In order to work with a Database Management System we also require a trained and educated staff and this also requires a good amount.

So, Database Management System results in a costlier system altogether.

2. Management Complexity

Once you have equipped Database Management System it is not a child's game to manage the same. You need to have good staff with management capabilities; at times it becomes quite complicated to decide from where to pick data and where to save that data.

3. Maintenance Cost

Database Management System also calls for a high maintenance cost which includes the cost to maintain the Database Management System once it is made.

4. Frequency Upgrade/Replacement Cycle

There need to be frequent upgrades and changes in the Database Management System in order to stay up-to-date and in accordance with the latest technological trends and developments in the markets. Sometimes these changes and updating are so fast that the users don't feel easy working with the systems and even the admin may find it a bit difficult to keep up with the system's changes (learning new commands and understanding them every time a new update is made).

DBMS is a wide area and before working in this field it is very important how it can be useful to us as per our needs. The above points give a great idea around that prospect.

Types of users in DBMS

Users are of 4 types:

1. Application programmers or Ordinary users
2. End users
3. Database Administrator (DBA)
4. System Analyst

1. Application programmers or Ordinary users: These users write application programs to interact with the database. Application programs can be written in some programming language such a COBOL, PL/I, C++, JAVA or some higher level fourth generation language. Such programs access the database by issuing the appropriate request, typically a SQL statement to DBMS.

2. End Users: End users are the users, who use the applications developed. End users need not know about the working, database design, the access mechanism etc. They just use the system to get their task done. End users are of two types:

a) Direct users b) Indirect users

a) Direct users: Direct users are the users who se the computer, database system directly, by following instructions provided in the user interface. They interact using the application programs already developed, for getting the desired result. E.g. People at railway reservation counters, who directly interact with database.

b) Indirect users: Indirect users are those users, who desire benefit form the work of DBMS indirectly. They use the outputs generated by the programs, for decision making or any other purpose. They are just concerned with the output and are not bothered about the programming part.

3. Database Administrator (DBA): Database Administrator (DBA) is the person which makes the strategic and policy decisions regarding the data of the enterprise, and who provide the necessary technical support for implementing these decisions. Therefore, DBA is responsible for overall control of the system at a technical level. In database environment, the primary resource is the database itself and the secondary resource is the DBMS and related software administering these resources is the responsibility of the Database Administrator (DBA).

4. System Analyst: System Analyst determines the requirement of end users, especially naïve and parametric end users and develops specifications for transactions that meet these requirements. System Analyst plays a major role in database design, its properties; the structure prepares the system requirement statement, which involves the feasibility aspect, economic aspect, technical aspect etc. of the system.

DBMS ARCHITECTURE:

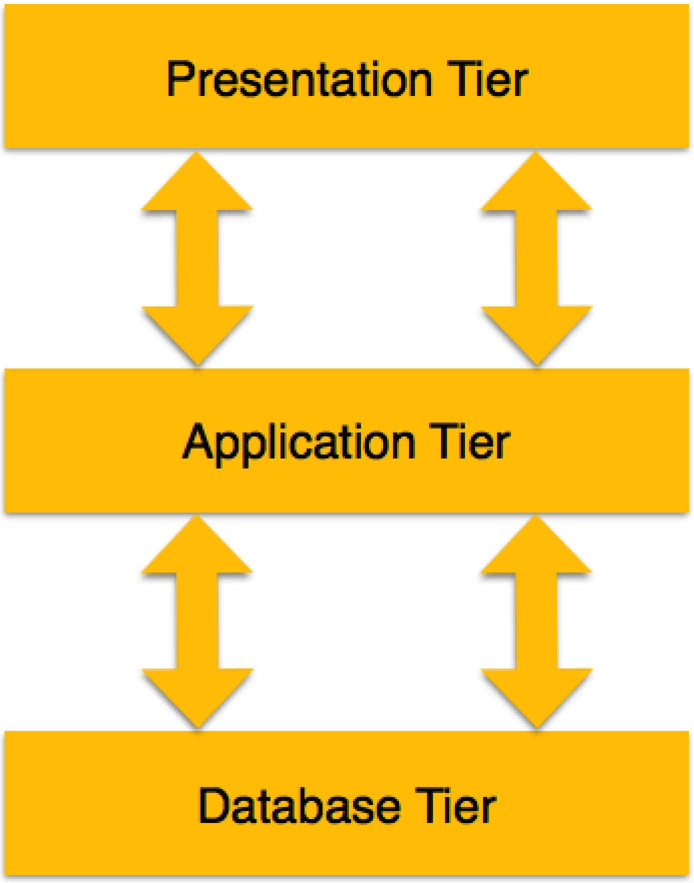
The design of a DBMS depends on its architecture. It can be centralized or decentralized or hierarchical. The architecture of a DBMS can be seen as either single tier or multi-tier. Ann-tier architecture divides the whole system into related but independent **n** modules, which can be independently modified, altered, changed, or replaced.

In 1-tier architecture, the DBMS is the only entity where the user directly sits on the DBMS and uses it. Any changes done here will directly be done on the DBMS itself. It does not provide handy tools for end-users. Database designers and programmers normally prefer to use single-tier architecture.

If the architecture of DBMS is 2-tier, then it must have an application through which the DBMS can be accessed. Programmers use 2-tier architecture where they access the DBMS by means of an application. Here the application tier is entirely independent of the database in terms of operation, design, and programming.

3-tier Architecture

A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is the most widely used architecture to design a DBMS.



- **Database (Data) Tier** – At this tier, the database resides along with its query processing languages. We also have the relations that define the data and their constraints at this level.

- **Application (Middle) Tier** – At this tier reside the application server and the programs that access the database. For a user, this application tier presents an abstracted view of the database. End-users are unaware of any existence of the database beyond the application. At the other end, the database tier is not aware of any other user beyond the application tier. Hence, the application layer sits in the middle and acts as a mediator between the end-user and the database.

- **User (Presentation) Tier** – End-users operate on this tier and they know nothing about any existence of the database beyond this layer. At this layer, multiple views of the database can be provided by the application. All views are generated by applications that reside in the application tier.

Multiple-tier database architecture is highly modifiable, as almost all its components are independent and can be changed independently.