UNIT - I
Introduction to DBMS

As the name suggests, the database management system consists of two parts. They are:

1. Database and
2. Management System

What is a Database?

To find out what database is, we have to start from data, which is the basic building block of any DBMS.

Data: Facts, figures, statistics etc. having no particular meaning (e.g. 1, ABC, 19 etc).

Record: Collection of related data items, e.g. in the above example the three data items had no meaning. But if we organize them in the following way, then they collectively represent meaningful information.

Roll	Name	Age
1	ABC	19

Table or **Relation**: Collection of related records.

Roll	Name	Age
1	ABC	19
2	DEF	22
3	XYZ	28

The columns of this relation are called **Fields**, **Attributes** or **Domains**. The rows are called **Tuples** or **Records**.

Database: Collection of related relations. Consider the following collection of tables:

T1

Roll	Name	Age
1	ABC	19
2	DEF	22
3	XYZ	28

T2

Roll	Address
1	KOL
2	DEL
3	MUM

T3

Roll	Year
1	I
2	II
3	I

T4

Year	Hostel
I	H1
II	H2

We now have a collection of 4 tables. They can be called a "related collection" because we can clearly find out that there are some common attributes existing in a selected pair of tables. Becauseof these common attributes we may combine the data of two or more tables together to find out the complete details of a student. Questions like "Which hostel does the youngest student live in?" canbe answered now, although ***Age*** and ***Hostel*** attributes are in different tables.

In a database, data is organized strictly in row and column format. The rows are called **Tuple** or **Record**. The data items within one row may belong to different data types. On the other hand, the columns are often called **Domain** or **Attribute**. All the data items within a single attribute are of the same data type.

What is Management System?

A management system is a set of rules and procedures which help us to create organize andmanipulate the database. It also helps us to add, modify delete data items in the database. The management system can be either manual or computerized.

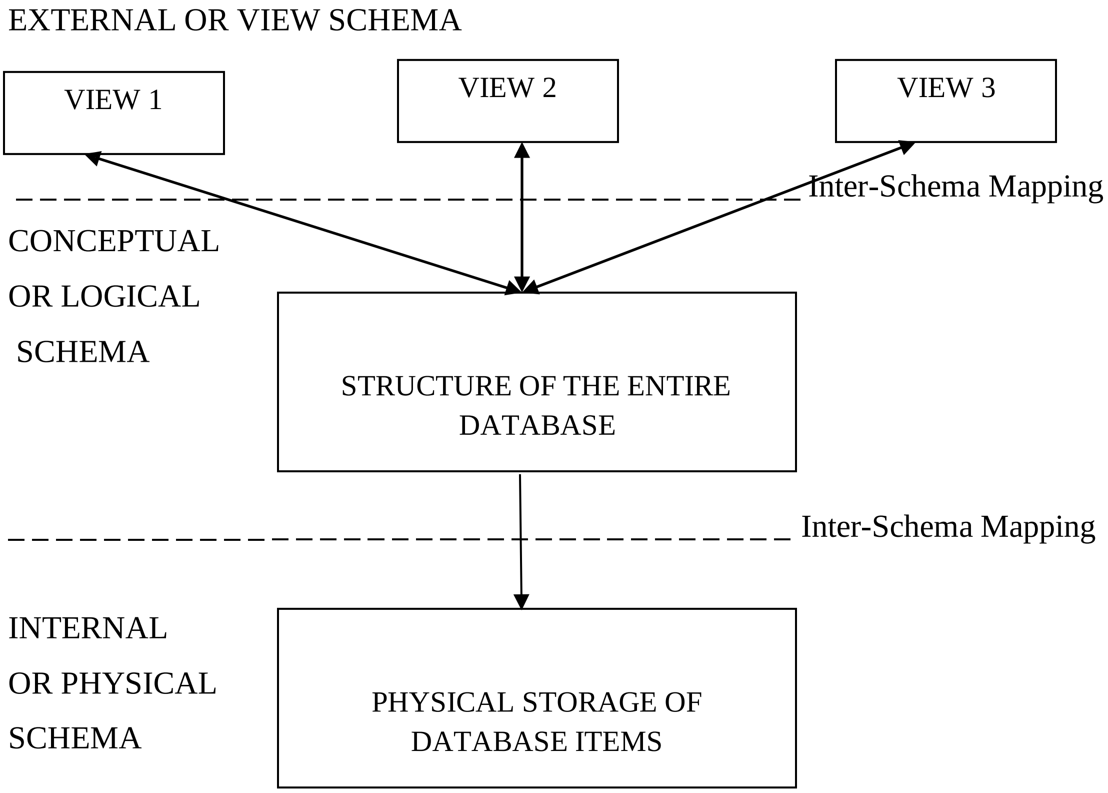
The management system is important because without the existence of some kind of rules and regulations it is not possible to maintain the database. We have to select the particular attributes which should be included in a particular table; the common attributes to create relationship between two tables; if a new record has to be inserted or deleted then which tables should have to be handled etc. These issues must be resolved by having some kind of rules to follow in order to maintain the integrity of the database.

Three Views of Data

We know that the same thing, if viewed from different angles produces difference sights. Likewise, the database that we have created already can have different aspects to reveal if seen from different levels of abstraction. The term **Abstraction** is very important here. Generally it means the amount of detail you want to hide. Any entity can be seen from different perspectives and levels of complexity to make it a reveal its current amount of abstraction. Let us illustrate by a simple example.

A computer reveals the minimum of its internal details, when seen from outside. We do not know what parts it is built with. This is the highest level of abstraction, meaning very few details are visible. If we open the computer case and look inside at the hard disc, motherboard, CD drive, CPU and RAM, we are in middle level of abstraction. If we move on to open the hard disc and examine itstracks, sectors and read-write heads, we are at the lowest level of abstraction, where no details are invisible.

In the same manner, the database can also be viewed from different levels of abstraction to reveal different levels of details. From a bottom-up manner, we may find that there are three levels of abstraction or views in the database. We discuss them here.



The word schema means arrangement – how we want to arrange things that we have to store. The diagram above shows the three different schemas used in DBMS, seen from different levels of abstraction.

The lowest level, called the **Internal or Physical schema**, deals with the description of how raw data items (like 1, ABC, KOL, H2 etc.) are stored in the physical storage (Hard Disc, CD, Tape Drive etc.). It also describes the data type of these data items, the size of the items in the storage media, the location (physical address) of the items in the storage device and so on. This schema is useful for database application developers and database administrator.

The middle level is known as the **Conceptual or Logical Schema**, and deals with the structure of the entire database. Please note that at this level we are not interested with the raw data items anymore, we are interested with the structure of the database. This means we want to know the information about the attributes of each table, the common attributes in different tables that help them to be combined, what kind of data can be input into these attributes, and so on. Conceptual or Logical schema is very useful for database administrators whose responsibility is to maintain the entire database.

The highest level of abstraction is the **External or View Schema**. This is targeted for the end users. Now, an end user does not need to know everything about the structure of the entire database, rather than the amount of details he/she needs to work with. We may not want the end user to become confused with astounding amount of details by allowing him/her to have a look at the entiredatabase, or we may also not allow this for the purpose of security, where sensitive information must remain hidden from unwanted persons. The database administrator may want to create custom made tables, keeping in mind the specific kind of need for each user. These tables are also known as**virtual tables**, because they have no separate physical existence. They are crated dynamically for theusers at runtime. Say for example, in our sample database we have created earlier, we have a special officer whose responsibility is to keep in touch with the parents of any under aged student living in the hostels. That officer does not need to know every detail except the ***Roll, Name, Addresss*** and ***Age***. The database administrator may create a virtual table with only these four attributes, only for the use of this officer.

Data Independence

This brings us to our next topic: data independence. It is the property of the database which tries to ensure that if we make any change in any level of schema of the database, the schema immediately above it would require minimal or no need of change.

What does this mean? We know that in a building, each floor stands on the floor below it. If we change the design of any one floor, e.g. extending the width of a room by demolishing the western wall of that room, it is likely that the design in the above floors will have to be changed also. As a result, one change needed in one particular floor would mean continuing to change the design ofeach floor until we reach the top floor, with an increase in the time, cost and labour. Would not lifebe easy if the change could be contained in one floor only? Data independence is the answer for this. It removes the need for additional amount of work needed in adopting the single change into all the levels above.

Data independence can be classified into the following two types:

1. **Physical Data Independence**: This means that for any change made in the physical schema, the need to change the logical schema is minimal. This is practically easier to achieve. Let us explain with an example.

Say, you have bought an Audio CD of a recently released film and one of your friends has bought an Audio Cassette of the same film. If we consider the physical schema, they are entirely different. The first is digital recording on an optical media, where random access is possible. The second one is magnetic recording on a magnetic media, strictly sequential access. However, how this change is reflected in the logical schema is very interesting. For music tracks, the logical schema for both the CD and the Cassette is the title card imprinted on their back. We have information like Track no, Name of the Song, Name of the Artist and Duration of the Track, things which are identical for both the CD and the Cassette. We can clearly say that we have achieved the physical data independence here.

2. **Logical Data Independence**: This means that for any change made in the logical schema, the need to change the external schema is minimal. As we shall see, this is a little difficult to achieve. Let us explain with an example.

   Suppose the CD you have bought contains 6 songs, and some of your friends are interested in copying some of those songs (which they like in the film) into their favorite collection. One friend wants the songs 1, 2, 4, 5, 6, another wants 1, 3, 4, 5 and another wants 1, 2, 3, 6. Each of these collections can be compared to a view schema for that friend. Now by some mistake, a scratch has appeared in the CD and you cannot extract the song 3. Obviously, you will have to ask the friends who have song 3 in their proposed collection to alter their view by deleting song 3 from their proposed collection as well.

Database Administrator

The Database Administrator, better known as DBA, is the person (or a group of persons) responsible for the well being of the database management system. S/he has the flowing functions and responsibilities regarding database management:

1. Definition of the schema, the architecture of the three levels of the data abstraction, data independence.
2. Modification of the defined schema as and when required.
3. Definition of the storage structure i.e. and access method of the data stored i.e. sequential, indexed or direct.
4. Creating new used-id, password etc, and also creating the access permissions that each user can or cannot enjoy. DBA is responsible to create user roles, which are collection of the permissions (like read, write etc.) granted and restricted for a class of users. S/he can

also grant additional permissions to and/or revoke existing permissions from a user if need be.

5. Defining the integrity constraints for the database to ensure that the data entered conform to some rules, thereby increasing the reliability of data.
6. Creating a security mechanism to prevent unauthorized access, accidental or intentional handling of data that can cause security threat.
7. Creating backup and recovery policy. This is essential because in case of a failure the database must be able to revive itself to its complete functionality with no loss of data, as if the failure has never occurred. It is essential to keep regular backup of the data so that if the system fails then all data up to the point of failure will be available from a stable storage. Only those amount of data gathered during the failure would have to be fed to thedatabase to recover it to a healthy status.

Advantages and Disadvantages of Database Management System

We must evaluate whether there is any gain in using a DBMS over a situation where we do not use it. Let us summarize the advantages.

1. **Reduction of Redundancy:** This is perhaps the most significant advantage of using DBMS. Redundancy is the problem of storing the same data item in more one place. Redundancy creates several problems like requiring extra storage space, entering same data more than once during data insertion, and deleting data from more than one place during deletion. Anomalies may occur in the database if insertion, deletion etc are not done properly.
2. **Sharing of Data:** In a paper-based record keeping, data cannot be shared among many users. But in computerized DBMS, many users can share the same database if they are connected via a network.
3. **Data Integrity:** We can maintain data integrity by specifying integrity constrains, which are rules and restrictions about what kind of data may be entered or manipulated within the database. This increases the reliability of the database as it can be guaranteed that no wrong data can exist within the database at any point of time.
4. **Data security:** We can restrict certain people from accessing the database or allow them to see certain portion of the database while blocking sensitive information. This is not possible very easily in a paper-based record keeping.

However, there could be a few disadvantages of using DBMS. They can be as following:

1. As DBMS needs computers, we have to invest a good amount in acquiring the hardware, software, installation facilities and training of users.
2. We have to keep regular backups because a failure can occur any time. Taking backup is a lengthy process and the computer system cannot perform any other job at this time.
3. While data security system is a boon for using DBMS, it must be very robust. If someone can bypass the security system then the database would become open to any kind of mishandling.