**VISION INSTITUTE OF TECHNOLOGY**

# DATABASE MANAGEMENT SYSTEM (KMBIT04)

**MBA 2nd YEAR/4th SEM**

**UNIT 1**

# Introduction to DBMS

A **DBMS** is a software that allows creation, definition and manipulation of database, allowing users to store, process and analyse data easily. DBMS provides us with an interface or a tool, to perform various operations like creating database, storing data in it, updating data, creating tables in the database and a lot more.

DBMS also provides protection and security to the databases. It also maintains data consistency in case of multiple users.

Here are some examples of popular DBMS used these days:

* MySql
* Oracle
* SQL Server
* IBM DB2
* PostgreSQL
* Amazon SimpleDB (cloud based) etc.

**Characteristics of Database Management System**

1. **Data stored into Tables:**Data is never directly stored into the database. Data is stored into tables, created inside the database. DBMS also allows to have relationships between tables which makes the data more meaningful and connected. You can easily understand what type of data is stored where by looking at all the tables created in a database.
2. **Reduced Redundancy:**In the modern world hard drives are very cheap, but earlier when hard drives were too expensive, unnecessary repetition of data in database was a big problem. But DBMS follows **Normalisation** which divides the data in such a way that repetition is minimum.
3. **Data Consistency:**On Live data, i.e. data that is being continuosly updated and added, maintaining the consistency of data can become a challenge. But DBMS handles it all by itself.
4. **Support Multiple user and Concurrent Access:**DBMS allows multiple users to work on it(update, insert, delete data) at the same time and still manages to maintain the data consistency.
5. **Query Language:**DBMS provides users with a simple Query language, using which data can be easily fetched, inserted, deleted and updated in a database.
6. **Security:**The DBMS also takes care of the security of data, protecting the data from un-authorised access. In a typical DBMS, we can create user accounts with different access permissions, using which we can easily secure our data by restricting user access.
7. DBMS supports **transactions**, which allows us to better handle and manage data integrity in real world applications where multi-threading is extensively used.

### **Advantages of DBMS**

* Segregation of application program.
* Minimal data duplicacy or data redundancy.
* Easy retrieval of data using the Query Language.
* Reduced development time and maintenance need.
* With Cloud Datacenters, we now have Database Management Systems capable of storing almost infinite data.
* Seamless integration into the application programming languages which makes it very easier to add a database to almost any application or website.

### **Disadvantages of DBMS**

* It’s Complexity
* Except MySQL, which is open source, licensed DBMSs are generally costly.
* They are large in size.

# Organization of Database

**Microsoft Access:**

Microsoft Office Access, previously known as Microsoft Access, is a relational database management system from Microsoft that combines the relational Microsoft jet database Engine with a graphical user interface and software-development tools. It is a member of the Microsoft Office suite of applications, included in the Professional and higher editions or sold separately. In mid-May 2010, the current version Microsoft Access 2010 was released by Microsoft in Office 2010; Microsoft Office Access 2007 was the prior version.

**My SQL**

MySQL is a relational database management system that runs as a server providing multi-user access to a number of databases. It is named after developer Michael Widenius’ daughter, my. The SQL phrase stands for Structured Query Language.

**Microsoft SQL Server**

Microsoft SQL Server is a relational model database server produced by Microsoft. Its primary query languages are T-SQL and ANSI S SQL.

**Oracle**

The Oracle Database (commonly referred to as Oracle RDBMS or simply as Oracle) is an object-relational database management system (ORDBMS), produced and marketed by Oracle Corporation.

**FileMaker Pro**

FileMaker Pro is a cross-platform relational database application from FileMaker Inc., formerly Claris, a subsidiary of Apple Inc. It integrates a database engine with a GUI-based interface, allowing users to modify the database by dragging new elements into layouts, screens, or forms.

### **Advantages**

Singh (2009) illustrates advantages and disadvantages of DBMS. A true DBMS offers several advantages over file processing. The principal advantages of a DBMS are the followings:

* Flexibility: Because programs and data are independent, programs do not have to be modified when types of unrelated data are added to or deleted from the database, or when physical storage changes.
* Fast response to information requests: Because data are integrated into a single database, complex requests can be handled much more rapidly then if the data were located in separate, non-integrated files. In many businesses, faster response means better customer service.
* Multiple access: Database software allows data to be accessed in a variety of ways (such as through various key fields) and often, by using several programming languages (both 3GL and nonprocedural 4GL programs).
* Lower user training costs: Users often find it easier to learn such systems and training costs may be reduced. Also, the total time taken to process requests may be shorter, which would increase user productivity.
* Less storage: Theoretically, all occurrences of data items need be stored only once, thereby eliminating the storage of redundant data. System developers and database designers often use data normalization to minimize data redundancy.

Warehouse of information, where large data can be stored.

Systematic storage meaning data can be stored in the form of tables.

Change of schema meaning it is not platform dependent tables can be edited to add new ones without hampering the applications.

No language dependence meaning use of various languages on various platforms.

Table joins meaning data can be in two or more tables and can be put into one table this enables easy retrieval of data.

* Data security meaning DBMS secures all your data.
* The data independence and efficient access of data
* Easy in data administration or data management.
* Provides concurrent access, recovers the data from the crashes.

### **Disadvantages**

A database system generally provides on-line access to the database for many users. In contrast, a conventional system is often designed to meet a specific need and therefore generally provides access to only a small number of users. Because of the larger number of users accessing the data when a database is used, the enterprise may involve additional risks as compared to a conventional data processing system in the following areas.

* Confidentiality, privacy and security.
* Data quality.
* Data integrity.
* Enterprise vulnerability may be higher.
* The cost of using DBMS.

**Confidentiality, Privacy and Security**

When information is centralised and is made available to users from remote locations, the possibilities of abuse are often more than in a conventional data processing system. To reduce the chances of unauthorised users accessing sensitive information, it is necessary to take technical, administrative and, possibly, legal measures.

Most databases store valuable information that must be protected against deliberate trespass and destruction.

**Data Quality**

Since the database is accessible to users remotely, adequate controls are needed to control users updating data and to control data quality. With increased number of users accessing data directly, there are enormous opportunities for users to damage the data. Unless there are suitable controls, the data quality may be compromised.

**Data Integrity**

Since a large number of users could be using a database concurrently, technical safeguards are necessary to ensure that the data remain correct during operation. The main threat to data integrity comes from several different users attempting to update the same data at the same time. The database therefore needs to be protected against inadvertent changes by the users.

**Enterprise Vulnerability**

Centralising all data of an enterprise in one database may mean that the database becomes an indispensible resource. The survival of the enterprise may depend on reliable information being available from its database. The enterprise therefore becomes vulnerable to the destruction of the database or to unauthorised modification of the database.

### **The Cost of using a DBMS**

Conventional data processing systems are typically designed to run a number of well-defined, pre-planned processes. Such systems are often “tuned” to run efficiently for the processes that they were designed for. Although the conventional systems are usually fairly inflexible in that new applications may be difficult to implement and/or expensive to run, they are usually very efficient for the applications they are designed for.

The database approach on the other hand provides a flexible alternative where new applications can be developed relatively inexpensively. The flexible approach is not without its costs and one of these costs is the additional cost of running applications that the conventional system was designed for. Using standardised software is almost always less machine efficient than specialised software.

Cost of hardware and software meaning having to upgrade the hardware used for file based system it is very costly.

Cost of data conversion meaning it’s costly to convert data of data files into database and have to hire database and system designers.

Cost of staff training meaning organization has to be aid a lot of amount for the training of staff to run dms.

Database damage meaning all data stored into a single file if database is damaged due to electric failure or database is corrupted on a storage media meaning all valuable data may be lost forever.

It also takes a lot of time and effort to get DBMS started.

Benefits of a Database Management System in an organization

As discuss by the Ezinearticles (2011), organizations often times find themselves in a dilemma on how they can effectively serve the needs of their members. At the same time, they also enter a situation wherein there is difficulty in handling data, given the sheer volume of information. More often than not, the results of these instances are not anything but desirable; confusion, chaos, mismanagement, lost confidence by members, etc.

To solve these problems, organizations turn to information technology (IT) experts for answers. With many years of experience in coming up with solutions for businesses and other entities, IT professionals have what it takes to come up with an effective data management and accreditation system that will be used by organizations. There must also be enhanced accessibility to the said system, which means that members from other locations can still gain entry into the system, regardless of elements such as time and geography.

Investing on the services of IT consultants is, however, not like buying something from a store. It entails huge sums of money to come up with a good data management and accreditation system for an organization, aside from engaging the professional services of IT experts. Fortunately, IT experts can help minimize the cost through the creation of a database system based on existing resources being held by the organization.

Investing on the services of IT experts and having a good data and accreditation management system may seem to be too costly, but the long-term benefits are worthy to be seen. By having a good database system, it will be much easier for organizations to deal with their members because of enhanced access to information. The placement of additional pieces of information can be done better if there is a good data management system.

Having a good data and accreditation management system empowers members of an organization. This is possible because they have the ability to gain the necessary information about things that matter to them as members. They conduction of business with the organization’s intervention is made faster and more efficient if a good data management system is in place.

There is no reason for organizations to hesitate when it comes to investing on a good data and accreditation management system. The long-term benefits of having one are far greater than the costs, and will be very helpful to the organization and their members in the long run.

* Improved strategic use of corporate data.
* Reduced complexity of the organization’s information systems environment.
* Reduced data redundancy and inconsistency.
* Enhanced data integrity.
* Application-data independence.
* Improved security.
* Reduced application development and maintenance costs.
* Improved flexibility of information systems.
* Increased access and availability of data and information.

# Components of Database

The database management system can be divided into five major components, they are:

1. Hardware
2. Software
3. Data
4. Procedures
5. Database Access Language

Let’s have a simple diagram to see how they all fit together to form a database management system

 

### **DBMS Components: Hardware**

When we say Hardware, we mean computer, hard disks, I/O channels for data, and any other physical component involved before any data is successfully stored into the memory.

When we run Oracle or MySQL on our personal computer, then our computer’s Hard Disk, our Keyboard using which we type in all the commands, our computer’s RAM, ROM all become a part of the DBMS hardware.

### **DBMS Components: Software**

This is the main component, as this is the program which controls everything. The DBMS software is more like a wrapper around the physical database, which provides us with an easy-to-use interface to store, access and update data.

The DBMS software is capable of understanding the Database Access Language and intrepret it into actual database commands to execute them on the DB.

### **DBMS Components: Data**

Data is that resource, for which DBMS was designed. The motive behind the creation of DBMS was to store and utilise data.

In a typical Database, the user saved Data is present and **Meta data** is stored.

**Metadata** is data about the data. This is information stored by the DBMS to better understand the data stored in it.

**For example:** When I store my **Name** in a database, the DBMS will store when the name was stored in the database, what is the size of the name, is it stored as related data to some other data, or is it independent, all this information is metadata.

### **DBMS Components: Procedures**

Procedures refer to general instructions to use a database management system. This includes procedures to setup and install a DBMS, To login and logout of DBMS software, to manage databases, to take backups, generating reports etc.

### **DBMS Components: Database Access Language**

Database Access Language is a simple language designed to write commands to access, insert, update and delete data stored in any database.

A user can write commands in the Database Access Language and submit it to the DBMS for execution, which is then translated and executed by the DBMS.

User can create new databases, tables, insert data, fetch stored data, update data and delete the data using the access language.

## **Users**

* **Database Administrators:**Database Administrator or DBA is the one who manages the complete database management system. DBA takes care of the security of the DBMS, it’s availability, managing the license keys, managing user accounts and access etc.
* **Application Programmer or Software Developer:**This user group is involved in developing and designing the parts of DBMS.
* **End User:**These days all the modern applications, web or mobile, store user data. How do you think they do it? Yes, applications are programmed in such a way that they collect user data and store the data on DBMS systems running on their server. End users are the one who store, retrieve, update and delete data.

# Advantages of DBMS

Compared to the File Based Data Management System, Database Management System has many advantages. Some of these advantages are given below:

#### **Reducing Data Redundancy**

The file based data management systems contained multiple files that were stored in many different locations in a system or even across multiple systems. Because of this, there were sometimes multiple copies of the same file which lead to data redundancy.

This is prevented in a database as there is a single database and any change in it is reflected immediately. Because of this, there is no chance of encountering duplicate data.

#### **Sharing of Data**

In a database, the users of the database can share the data among themselves. There are various levels of authorisation to access the data, and consequently the data can only be shared based on the correct authorisation protocols being followed.

Many remote users can also access the database simultaneously and share the data between themselves.

#### **Data Integrity**

Data integrity means that the data is accurate and consistent in the database. Data Integrity is very important as there are multiple databases in a DBMS. All of these databases contain data that is visible to multiple users. So it is necessary to ensure that the data is correct and consistent in all the databases and for all the users.

#### **Data Security**

Data Security is vital concept in a database. Only authorised users should be allowed to access the database and their identity should be authenticated using a username and password. Unauthorised users should not be allowed to access the database under any circumstances as it violates the integrity constraints.

#### **Privacy**

The privacy rule in a database means only the authorized users can access a database according to its privacy constraints. There are levels of database access and a user can only view the data he is allowed to. For example – In social networking sites, access constraints are different for different accounts a user may want to access.

#### **Backup and Recovery**

Database Management System automatically takes care of backup and recovery. The users don’t need to backup data periodically because this is taken care of by the DBMS. Moreover, it also restores the database after a crash or system failure to its previous condition.

#### **Data Consistency**

Data consistency is ensured in a database because there is no data redundancy. All data appears consistently across the database and the data is same for all the users viewing the database. Moreover, any changes made to the database are immediately reflected to all the users and there is no data inconsistency.

**DATA MODELS**

# Data Models: Entity-Relationship Model

Data models define how the logical structure of a database is modeled. Data Models are fundamental entities to introduce abstraction in a DBMS. Data models define how data is connected to each other and how they are processed and stored inside the system.

The very first data model could be flat data-models, where all the data used are to be kept in the same plane. Earlier data models were not so scientific, hence they were prone to introduce lots of duplication and update anomalies.

### **Entity-Relationship Model**

Entity-Relationship (ER) Model is based on the notion of real-world entities and relationships among them. While formulating real-world scenario into the database model, the ER Model creates entity set, relationship set, general attributes and constraints.

ER Model is best used for the conceptual design of a database.

ER Model is based on:

* **Entities**and their
* **Relationships**among entities.

These concepts are explained below.

* **Entity**− An entity in an ER Model is a real-world entity having properties called **attributes**. Every **attribute** is defined by its set of values called **domain**. For example, in a school database, a student is considered as an entity. Student has various attributes like name, age, class, etc.
* **Relationship**− The logical association among entities is called **relationship**. Relationships are mapped with entities in various ways. Mapping cardinalities define the number of association between two entities.

**Mapping cardinalities**

* one to one
* one to many
* many to one
* many to many

### **Relational Model**

The most popular data model in DBMS is the Relational Model. It is more scientific a model than others. This model is based on first-order predicate logic and defines a table as an **n-ary relation**.

 

The main highlights of this model are:

* Data is stored in tables called **relations**.
* Relations can be normalized.
* In normalized relations, values saved are atomic values.
* Each row in a relation contains a unique value.
* Each column in a relation contains values from a same domain.

# Hierarchy Data Model, Network Data Model

### **Hierarchy Data Model**

This database model organises 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.

In hierarchical model, data is organised into tree-like structure with one one-to-many relationship between two different types of data, for example, one department can have many courses, many professors and of-course many students.

 

### **Network Data Model**

This is an extension of the Hierarchical model. In this model data is organised 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.



# Relational Data Model, Semantic Data Model

#### **Relational Data Model**

In this model, data is organised 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, infact, 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.

 

### **Semantic Data Model**

The semantic data model is a method of structuring data in order to represent it in a specific logical way. It is a conceptual data model that includes semantic information that adds a basic meaning to the data and the relationships that lie between them. This approach to data modeling and data organization allows for the easy development of application programs and also for the easy maintenance of data consistency when data is updated.

The semantic data model is a relatively new approach that is based on semantic principles that result in a data set with inherently specified data structures. Usually, singular data or a word does not convey any meaning to humans, but paired with a context this word inherits more meaning.

In a database environment, the context of data is often defined mainly by its structure, such as its properties and relationships with other objects. So, in a relational approach, the vertical structure of the data is defined by explicit referential constraints, but in semantic modeling this structure is defined in an inherent way, which is to say that a property of the data itself may coincide with a reference to another object.

A semantic data model may be illustrated graphically through an abstraction hierarchy diagram, which shows data types as boxes and their relationships as lines. This is done hierarchically so that types that reference other types are always listed above the types that they are referencing, which makes it easier to read and understand.

Abstractions used in a semantic data model:

* Classification – “instance\_of” relations
* Aggregation – “has\_a” relations
* Generalization – “is\_a” relations