**ST. XAVIER’S COLLEGE**

**Maitighar, Kathmandu**



Data Base Management System Assignment #2

**Submitted by**

Yub Raj Basnet

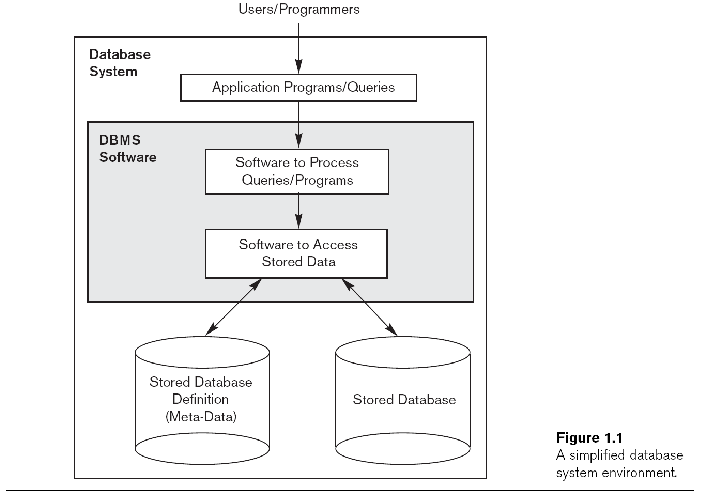
013BScCSIT048 (4th Semester)

**Submitted to**

|  |  |
| --- | --- |
| Er. Sanjaya Kumar Yadav  (Lecturer, St. Xavier’s College ) |  |

# SIMPLIFIED DATA BASE SYSTEM ORGANIZATION:

Database management system (DBMS) as a software system able to manage collections of data that are large, shared and persistent, and to ensure their reliability and privacy. Like any software product, a DBMS must be efficient and effective. A database is a collection of data managed by a DBMS.

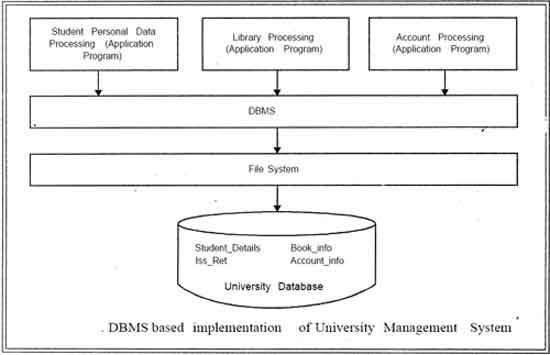


# APPROACHES TO MANAGEMENT OF DATA:

## Data Base Approach:

The Database is a shared collection of logically related data, designed to meet the information needs of an organization. A database is a computer based record keeping system whose over all purpose is to record and maintains information. The database is a single, large repository of data, which can be used simultaneously by many departments and users. Instead of disconnected files with redundant data, all data items are integrated with a minimum amount of duplication.

The database is no longer owned by one department but is a shared corporate resource. The database holds not only the organization's operational data but also a description of this data. For this reason, a database is also defined as a self-describing collection of integrated records. The description of the data is known as the Data Dictionary or Meta Data (the 'data about data'). It is the self-describing nature of a database that provides program-data independence.



A database implies separation of physical storage from use of the data by an application program to achieve program/data independence. Using a database system, the user or programmer or application specialist need not know the details of how the data are stored and such details are "transparent to the user". Changes (or updating) can be made to data without affecting other components of the system. These changes include, for example, change of data format or file structure or relocation from one device to another.

## File System Approach

The conventional approach to data management exploits the presence of files to store data permanently. A file allows for the storage and searching of data, but provides only simple mechanisms for access and sharing. Data of possible interest to more than one program is replicated as many times as there are programs that use it, with obvious redundancy and the possibility of inconsistency. Databases were created, for the most part, to overcome this type of inconvenience.

# DATABASE VS FILE SYSTEM APPROACH

There are many differences between file system approach and database approach. Some of them are:

## Data Abstraction:

For the system to be usable, it must retrieve data efficiently. The need for efficiency has led designers to use complex data structures to represent data in the database. Since many database-systems users are not computer trained, developers hide the complexity from users through several levels of abstraction, to simplify users’ interactions with the system. The three levels of data abstraction:

* **Physical Level** : The lowest level of abstraction describes how the data are actually stored. The physical level describes complex low-level data structures in detail.
* **Logical Level** : The next-higher level of abstraction describes what data are stored in the database, and what relationships exist among those data. The logical level thus describes the entire database in terms of a small number of relatively simple structures. Although implementation of the simple structures at the logical level may involve complex physical-level structures, the user of the logical level does not need to be aware of this complexity. Database administrators, who must decide what information to keep in the database, use the logical level of abstraction.
* **View Level** : The highest level of abstraction describes only part of the entire database. Even though the logical level uses simpler structures, complexity remains because of the variety of information stored in a large database. Many users of the database system do not need all this information; instead, they need to access only a part of the database. The view level of abstraction exists to simplify their interaction with the system. The system may provide many views for the same database.

## Reliability:

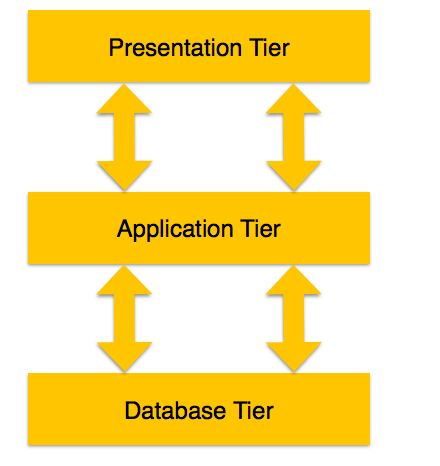
DB approach ensure reliability; that is, the capacity of the system to preserve the content of the database (or at least to allow its reconstruction) in case of hardware or software failure. To fulfill this requirement, DB System provides specific functions for backup and recovery. But there is no any provision available in terms of file system approach.

## Efficiency / performance:

DB approach is also concerned with efficiency, that is, the capacity to carry out operations using an appropriate amount of resources (time and space) for each user. This characteristic relies on the techniques used in the implementation of the dbms, and on how well the product has been designed. It should be stressed that DB System provide a wide-ranging combination of features that require many resources, and therefore they often put heavy requirements on the resources provided by the operating environment. But in file system approach the tools are limited and to perform the certain task lots of time and resource are needed which decreases the efficiency.

# 3RD LAYER 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

# ADVANTAGE AND DISADVANTAGE OF DBMS

## Advantages:

• DBMS allow data to be considered as a common resource of an organization, available to all its authorized members.

• The database provides a standardized and precise model of that part of the real world of interest to the organization, usable in existing applications and, with the necessary extensions, in future applications.

• With the use of a DBMS, centralized control of the data is possible, which can be improved by forms of standardization and can benefit from an ‘economy of scale’.

• Sharing allows the reduction of redundancy and inconsistency.

• Data independence, the fundamental characteristic of DBMSs, favors the development of applications that are more flexible and more easily modifiable.

## Disadvantages:

• DBMSs are expensive products, complex and quite different from many other software tools. Their introduction therefore represents a considerable investment, both direct (the cost of the product) and indirect (the acquisition of the necessary hardware and software resources, application migration, personnel training).

• DBMSs provide, in standardized form, a whole set of services, which necessarily carry a cost. In the cases where some of these services are not needed, it is difficult to extract the services actually required from the others, and this can generate inefficiencies.