**ST. XAVIER’S COLLEGE**

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**Database Management System**

**Lab Assignment #2**

**Submitted by:**

Arun Sanjel  
013BSCCSIT010

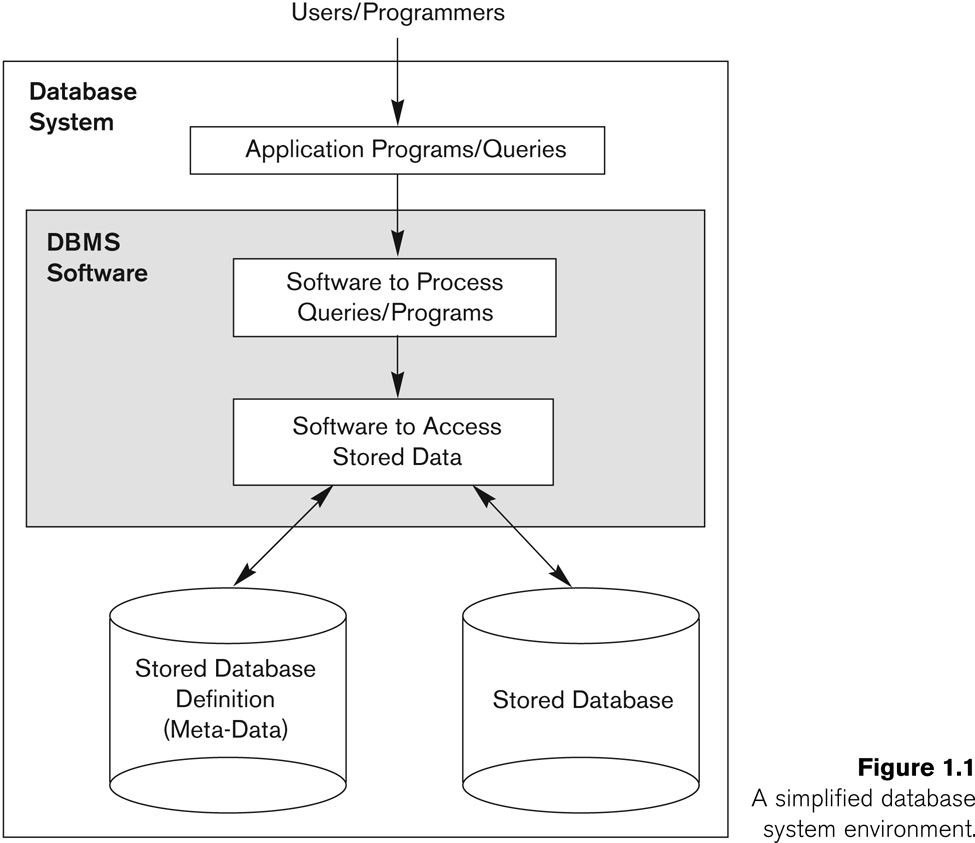
**Submitted to:**

|  |  |
| --- | --- |
| **Er. Sanjay Kumar Yadav**  Lecturer  St. Xavier’s College |  |

**Date of Submission:** July 27, 2015

**Simplified Database System Organisation**

The term database system refers to an organization of components that define and regulate the collection, storage, management, and use of data within a database environment. A simplified database system organization consists of basic components like user, simple DBMS software, and a Database. Below figure graphical shows a simplified database system organization.



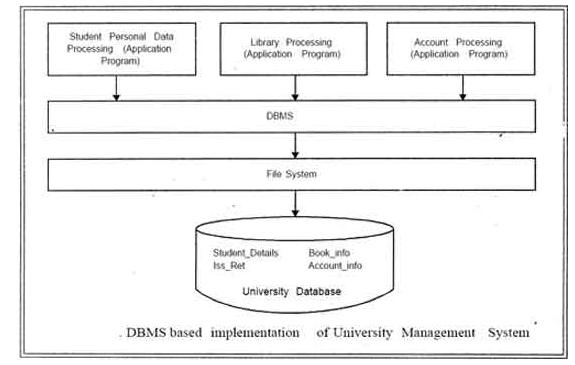
Simplified Database System Organization [1]

**Approaches to management of data**

Data management comprises all the disciplines related to managing data as a valuable resource. For managing huge data, different approaches are used like file system, database system and so on. File system management is primitive method of managing data and is seldom used whereas database approach is widely used.

**Database Approach**

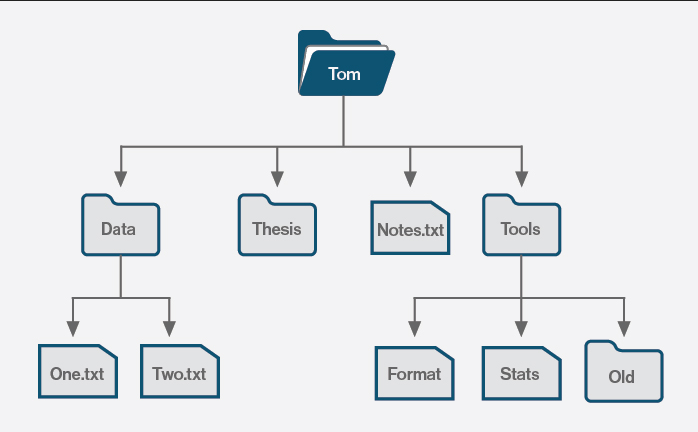
In order to remove all limitations of the File Based Approach, a new approach was required that must be more effective known as Database approach. The Database is a shared collection of logically related data, designed to meet the information needs of an organization. A database is a computer based record 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.



Database Approach [2]

**File system Approach**

A file system is the method an operating system uses to name files and assign them locations for efficient storage and retrieval. For example, DOS, Windows, OS/2, Macintosh and Unix-based operating systems (OS) all have file systems in which files are placed somewhere in a hierarchical (tree) structure. A file is placed in a directory (folder in Windows) or subdirectory at the desired place in the tree structure. File systems specify conventions for naming files, including the maximum number of characters in a name, which characters can be used and, in some systems, how long the file name suffix can be. A file system also includes a format for specifying the path to a file through the structure of directories. [3]



**File vs. Database approach**

File system approach are traditional method of data management which uses files to manage data. They are now being replaced by Database approach because of features like high reliability, quick access, feasibility, high security, and backup. Three major difference in both approaches are:

**Data Abstraction**

Data abstraction is the reduction of a particular body of data to a simplified representation of the whole. Abstraction, in general, is the process of taking away or removing characteristics from something in order to reduce it to a set of essential characteristics. As in abstract art, the representation is likely to be one potential abstraction of a number of possibilities. A database abstraction layer, for example, is one of a number of such possibilities. Data abstraction is usually the first step in database design. A complete database is much too complex a system to be developed without first creating a simplified framework. Data abstraction makes it possible for the developer to start from essential elements -- data abstractions -- and incrementally add data detail to create the final system.[4]

**Data Reliability**

Database approach as high reliability of data as the backup of database is taken numerous of times in different location so that any disaster in database doesn’t corrupt the data and can be accessed again without any loss of data. File system doesn’t have same backup policy which cause to loss of data after any disaster in file storage.

**Efficiency/Performance**

Using database approach, the efficiency and productive of an organization increases as they don’t have to waste time on searching different files to check records and data for processing the client request. Thus, making database approach very productive and efficient tool to manage data.

**Different Layer 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. N-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**

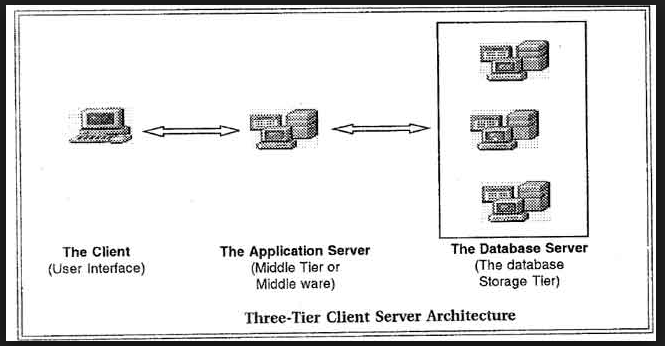
A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is the most widely used architecture to design a DBMS.

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. [5]



[6]

**Advantages of DBMS**

The different advantages of DBMS are as follows.

1. Improved data sharing.

The DBMS helps create an environment in which end users have better access to more and better-managed data. Such access makes it possible for end users to respond quickly to changes in their environment.

2. Improved data security.

The more users access the data, the greater the risks of data security breaches. Corporations invest considerable amounts of time, effort, and money to ensure that corporate data are used properly. A DBMS provides a framework for better enforcement of data privacy and security policies.

3. Better data integration.

Wider access to well-managed data promotes an integrated view of the organization’s operations and a clearer view of the big picture. It becomes much easier to see how actions in one segment of the company affect other segments.

4. Minimized data inconsistency.

Data inconsistency exists when different versions of the same data appear in different places. For example, data inconsistency exists when a company’s sales department stores a sales representative’s name as “Bill Brown” and the company’s personnel department stores that same person’s name as “William G. Brown,” or when the company’s regional sales office shows the price of a product as $45.95 and its national sales office shows the same product’s price as $43.95. The probability of data inconsistency is greatly reduced in a properly designed database.

5. Improved data access.

The DBMS makes it possible to produce quick answers to ad hoc queries. From a database perspective, a query is a specific request issued to the DBMS for data manipulation—for example, to read or update the data. Simply put, a query is a question, and an ad hoc query is a spur-of-the-moment question. The DBMS sends back an answer (called the query result set) to the application. For example, end users, when dealing with large amounts of sales data, might want quick answers to questions (ad hoc queries) such as:

- What was the dollar volume of sales by product during the past six months?

- What is the sales bonus figure for each of our salespeople during the past three months?

- How many of our customers have credit balances of $3,000 or more?

6.Improved decision making.

Better-managed data and improved data access make it possible to generate better-quality information, on which better decisions are based. The quality of the information generated depends on the quality of the underlying data. Data quality is a comprehensive approach to promoting the accuracy, validity, and timeliness of the data. While the DBMS does not guarantee data quality, it provides a framework to facilitate data quality initiatives.

7.Increased end-user productivity.

The availability of data, combined with the tools that transform data into usable information, empowers end users to make quick, informed decisions that can make the difference between success and failure in the global economy.

**Disadvantages of Database**

Although the database system yields considerable advantages over previous data management approaches, database systems do carry significant disadvantages. For example:

1. Increased costs.

Database systems require sophisticated hardware and software and highly skilled personnel. The cost of maintaining the hardware, software, and personnel required to operate and manage a database system can be substantial. Training, licensing, and regulation compliance costs are often overlooked when database systems are implemented.

2. Management complexity.

Database systems interface with many different technologies and have a significant impact on a company’s resources and culture. The changes introduced by the adoption of a database system must be properly managed to ensure that they help advance the company’s objectives. Given the fact that database systems hold crucial company data that are accessed from multiple sources, security issues must be assessed constantly.

3. Maintaining currency.

To maximize the efficiency of the database system, you must keep your system current. Therefore, you must perform frequent updates and apply the latest patches and security measures to all components. Because database technology advances rapidly, personnel training costs tend to be significant. Vendor dependence. Given the heavy investment in technology and personnel training, companies might be reluctant to change database vendors. As a consequence, vendors are less likely to offer pricing point advantages to existing customers, and those customers might be limited in their choice of database system components.

4. Frequent upgrade/replacement cycles.

DBMS vendors frequently upgrade their products by adding new functionality. Such new features often come bundled in new upgrade versions of the software. Some of these versions require hardware upgrades. Not only do the upgrades themselves cost money, but it also costs money to train database users and administrators to properly use and manage the new features.[7]

**Drawbacks of File management System**

**Data Redundancy**

Data Redundancy means same information is duplicated in several files. This makes data redundancy.

**Data Inconsistency**

Data Inconsistency means different copies of the same data are not matching. That means different versions of same basic data are existing. This occurs as the result of update operations that are not updating the same data stored at different places.

Example: Address Information of a customer is recorded differently in different files.

**Difficulty in Accessing Data**

It is not easy to retrieve information using a conventional file processing system. Convenient and efficient information retrieval is almost impossible using conventional file processing system.

**Data Isolation**

Data are scattered in various files, and the files may be in different format, writing new application program to retrieve data is difficult.

**Integrity Problems**

The data values may need to satisfy some integrity constraints. For example the balance field Value must be grater than 5000. We have to handle this through program code in file processing systems. But in database we can declare the integrity constraints along with definition itself.

**Atomicity Problem**

It is difficult to ensure atomicity in file processing system.For example transferring $100 from Account A to account B.If a failure occurs during execution there could be situation like $100 is deducted from Account A and not credited in Account B.

**Concurrent Access anomalies**

If multiple users are updating the same data simultaneously it will result in inconsistent data state. In file processing system it is very difficult to handle this using program code. This results in concurrent access anomalies.

**Security Problems**

Enforcing Security Constraints in file processing system is very difficult as the application programs are added to the system in an ad-hoc manner.[8]

**References**

[1] <http://www.cs.kau.se/cs/education/courses/dvgb04/lectures/kap1_figurer.pdf>

[2] <http://ecomputernotes.com/fundamental/what-is-a-database/database-approach>

[3] <http://searchstorage.techtarget.com/definition/file-system>

[4] <http://whatis.techtarget.com/definition/data-abstraction>

[5] <http://www.tutorialspoint.com/dbms/dbms_architecture.htm>

[6]<http://ecomputernotes.com/images/Three%20Tier%20Client%20Server%20Architecture.jpg>

[7] <http://www.myreadingroom.co.in/about/226-advantages-and-disadvantages-of-dbms.html>

[8] http://www.jkinfoline.com/disadvantages-of-file-system.html