**ST. XAVIER’S COLLEGE**

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**Database Management System**

**Lab Assignment #2**

**Submitted by:**

Divya Jyoti Pokharel

013BSCCSIT019

**Submitted to:**

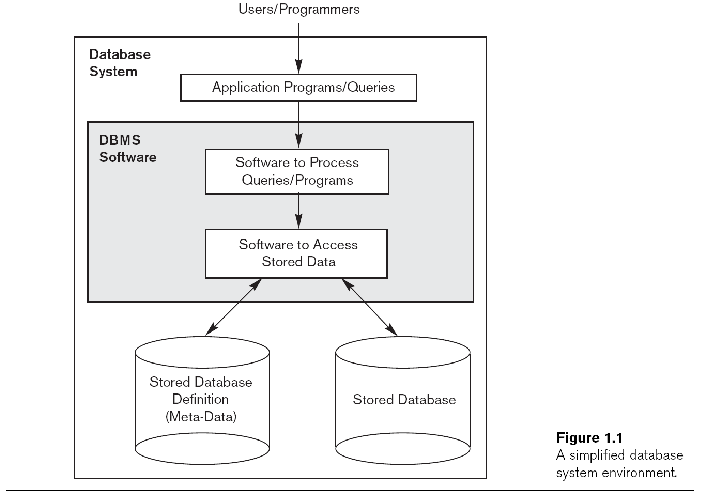
|  |  |
| --- | --- |
| **Er. Sanjay Kumar Yadav**  Lecturer  St. Xavier’s College |  |

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

**Simplified database system organization**

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. In a high level view the database system is composed of the following five major parts.

* Hardware
* Software
* People
* Procedures
* Data



**Approach to management ode data**

Data management is the development and execution of architectures, policies, practices and procedures in order to manage the information lifecycle needs of an enterprise in an effective manner.

**Database approach**

The database approach is a way in which data is stored within a computer. It is organized into various charts that are accessed by a variety of computer applications from different locations. Databases are composed of a variety of information that is pertinent and relevant to the organization that is using the database.

**File system approach**

In computing, a file system (or filesystem) is used to control how data is stored and retrieved.

Files linked to a specified application

**Database vs filesystem approach**

A database management system coordinates both the physical and the logical access to the data, whereas a file-processing system coordinates only the physical access.

A database management system is designed to allow flexible access to data (i.e. queries), whereas a file-processing system is designed to allow predetermined access to data (i.e. compiled programs).

A database management system is designed to coordinate multiple users accessing the same data at the same time. A file-processing system is usually designed to allow one or more programs to access different data files at the same time. In a file-processing system, a file can be accessed by two programs concurrently only if both programs have read-only access to the file.

Redundancy is control in DBMS, but not in file system.

Unauthorized access is restricted in DBMS but not in the file system.

DBMS provide backup and recovery whereas data lost in file system can't be recovered.

DBMS provide multiple user interfaces. Data is isolated in file system.

**3 layer Architecture**

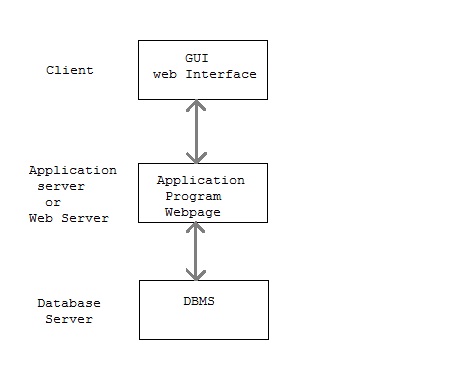
The design of a DBMS depends on its architecture. It can be centralized or decentralized or hierarchical. The architecture of a DBMS can be seen as either single tier or multi-tier. An n-tier architecture divides the whole system into related but independent n modules, which can be independently modified,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

Client/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.

**Advantages of DBMS**

* Controlling Redundancy
* Integrity can be enforced
* Inconsistency can be avoided
* Data can be shared
* Standards can be enforced
* Restricting unauthorized access
* Solving Enterprise Requirement than Individual Requirement
* Providing Backup and Recovery
* Cost of developing and maintaining system is lower
* Data Model can be developed
* Concurrency Control - DBMS systems provide mechanisms to provide concurrent access of data to multiple users.

**Disadvantages of DBMS**

* Management Complexity
* Size - The complexity and breadth of functionality makes the DBMS an extremely large piece of software, occupying many megabytes of disk space and requiring substantial amounts of memory to run efficiently.
* Performance
* Higher impact of a failure: The centralization of resources increases the vulnerability of the system. Since all users and applications rely on the ~vailabi1ity of the DBMS, the failure of any component can bring operations to a halt.
* Increased Cost of DBMS and Additional Hardware costs
* Cost of Conversion

**Drawbacks of filesystem to store data**

* Data Redundancy
* Data Inconsistency
* Difficulty in Accessing Data
* Data Isolation
* Integrity Problems
* Security and access control
* Concurrency Problems
* Problem in atomicity of data