**ST. XAVIER’S COLLEGE**

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**Database Management System**

**Lab Assignment #2**

**Submitted by:**

Anlisha Maharjan  
013BSCCSIT006

**Submitted to:**

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

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

**SIMPLIFIED DATABASE SYSTEM ORGANIZATION:**

A database management system (DBMS) is a computer program (or more typically, a suite of them) designed to manage a database; a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMS use include accounting, human resources and customer support systems.

**APPROACHES TO MANAGEMENT OF DATA:**

The two approaches to management of data are described below:

**DATABASE APPROACH:**

The Database is a shared collection of logically related data, designed to meet the information needs of an organization. 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. 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.

In the DBMS approach, application program written in some programming language like Java, Visual Basic.Net, and Developer 2000 etc. uses database connectivity to access the database stored in the disk with the help of operating system's file management system.

**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 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. File systems use metadata to store and retrieve files. Some examples of metadata tags include: Date created, Date modified and File size.

**DATABASE VS FILE SYSTEM APPROACHES:**

* Mini world as data source
* Universe of Discourse (UOD)
* Logically integrated files
* Intended users and applications
* Shared and Self-describing

Compared with file-based approach:

* program-data independence
* multiple view of data
* multi-user transaction processing

**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. Data abstraction is usually the first step in database design. 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.

There are several levels of abstraction:

**Physical Level:** It deals with how the data are stored. Example: index, B-tree, hashing. This is the lowest level of abstraction.

**Conceptual Level:** Next highest level of abstraction. Describes what data are stored. It describes the relationships among data.

**View Level:** Highest level. It describes part of the database for a particular group of users. Example: tellers in a bank get a view of customer accounts, but not of payroll data.

**RELIABILITY:**

Data reliability is the ability to establish logical relationships between different types of records usually in different files. For example, show the relationships between the master files and account files in CNB's database. Because customers have checking accounts, and checking accounts have balances, a relationship exists between the files containing these two data items.

**EFFICIENCY/PERFORMANCE:**

Data Efficiency refers to efficiency of the many processes that can be applied to data such as storage, access, filtering, sharing, etc., and whether or not the processes lead to the desired outcome within resource constraints. A management definition of Data Efficiency would be the measure of how data storage and usage across an enterprise or within a department or within a project - impacts the organization’s costs and revenues.

**3-LAYER ARCHITECTURE:**

ANSI is the acronym for American National Standards Institute. In the case of DBMS software, ANSI has standardized SQL, so that most DBMS products use SQL as the main query language. The ANSI has also standardized a three level DBMS architecture model followed by most database systems, and it’s known as the abstract ANSI-SPARC design standard. The ANSI-SPARC Database Architecture is set up into three tiers.

**The Internal Level (Physical Representation of Data)**

The internal level is the lowest level in a three tiered database. This level deals with how the stored data on the database is represented to the user. This level shows exactly how the data is stored and organized for access on your system. This is the most technical of the three levels. There are several considerations to be made when storing data. Some of them include figuring out the right space allocation techniques, data compression techniques (if necessary), security and encryption and the access paths the software can take to retrieve the data.

**The Conceptual Level (Holistic Representation of Data)**

The conceptual level tells you how the database was structured logically. This level tells you about the relationship between the data members of your database, exactly what data is stored in it and what a user will need to use the database. This level does not concern itself with how this logical structure will actually be implemented. It’s actually an overview of your database.The conceptual level acts as a sort of a buffer between the internal level and the external level. It helps hide the complexity of the database and hides how the data is physically stored in it.

**The External Level (User Representation of Data)**

This is the uppermost level in the database. It implements the concept of abstraction as much as possible. This level is also known as the view level because it deals with how a user views your database. The external level is what allows a user to access a customized version of the data in your database. The external level also hides the working of the database from your users. It maintains the security of the database by giving users access only to the data which they need at a particular time.

**Advantages of the Three Tiered ANSI-SPARC Architecture:**

It makes the database abstract.

It lets users view the same data, but it makes the data customizable to fit each user.

The three tiered architecture model also allows migration to another system to be seamless.

The model also allows a database admin to make changes to the database structure or make upgrades to it without disturbing a user currently on the system.

The model allows a database admin to change the storage medium of the database without disturbing a user who is currently on the system.

**ADVANTAGES OF DBMS:**

**Data independence:** DBMS provides abstract view of data. Application programs are independent from details of data representation and storage.

**Efficient data access:** DBMS provides verity of sophisticated techniques to store and retrieve data efficiently.

**Data integrity and security:** DBMS allow enforcing integrity constraints on data. For example before inserting salary information for an employee, DBMS can enforce integrity constraint to check salary is not exceeded department budget. DBMS can also enforce access controls, what data is visible to what class of users.

**Data administration:** DBMS provides centralized administration of data. It is appropriate when several no. of database user shares data. It improves the overall performance of database system.

**Concurrent access and crash recovery:** DBMS has a capability manage concurrent access. It schedules concurrent access to the data in such a manner that user fell data is being accessed by only one user at a time. Moreover, DBMS protects users from the effects of system failures.

**Reduced application development time:** Since DBMS supports many important functions that are common to many applications accessing data stored in database. It provides high level interface to data and facilitates quick development of applications.

**DISADVANTAGES OF DBMS:**

The disadvantages of DBMS are:

* Complex architecture of DBMS software
* DBMS software cost
* Since DBMS is optimized certain kind of workloads (e.g. answering complex queries or handling many concurrent requests) its performance may not appropriate for certain specialized applications.
* Abstract view of data presented by DBMS may not match for certain applications. For example, relational databases does not supports flexible analysis of text data
* If specialized performance or data manipulation requirements are central to an application, DBMS is not appropriate for such application. The added benefits of a DBMS (e.g. flexible querying, security, concurrent access and crash recovery) may not require for applications.

**DRAWBACKS OF USING FILE SYSTEMS TO STORE DATA:**

* Data Redundancy
* Data Inconsistency
* Difficulty in Accessing Data
* Data Isolation
* Integrity problem
* Security and Access Control

**REFERENCES:**

[1] <http://rashedul.info/materials/DisadvantagesofFilesystemoverDBMS.pdf>

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

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

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

[5] <http://www.techopedia.com/definition/24649/three-tier-architecture>

[6] <https://en.wikipedia.org/wiki/Category:Database_management_systems>