ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

**Maitighar, Kathmandu**

****

**Database Management System**

**Theory Lab Assignment #2**

**SUBMITTED BY:**

**Siddhant Rimal**

**013BSCCSIT039**

**SUBMITTED TO**

|  |  |
| --- | --- |
| **Mr. Sanjay Kr. Yadav**  **( Lecturer )** |  |
| **Department of Computer Science** | |

Submission Date: July 27th 2015

2.1. Simplified Database System Organization

Fig: Simplified Database System Organization [1]

A **database management system (DBMS)** is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications. [12]

* **Defining** a database involves specifying the data types, structures, and constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalog or dictionary; it is called **meta-data**. [12]
* **Constructing** the database is the process of storing the data on some storage medium that is controlled by the DBMS. [12]
* **Manipulating** a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the miniworld, and generating reports from the data [12]
* **Sharing** a database allows multiple users and programs to access the database simultaneously. [12]

2.2. Approaches to management of data

Data management can be defined as the development, execution and oversight of architectures, policies, practices, and procedures to manage the information lifecycle needs of an enterprise in an effective manner the information lifecycle needs of an enterprise in an effective manner as it pertains to data collection, storage, security, data inventory, analysis, quality control, reporting, and visualization. [8]

# 2.2.1. File System Approach

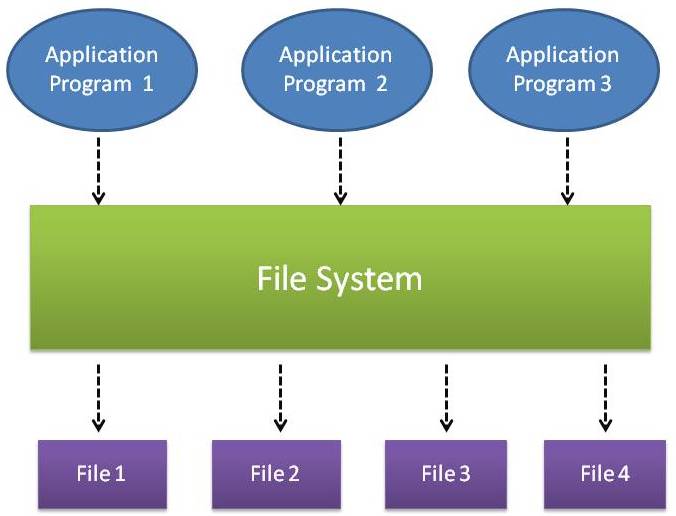


Fig: File System Approach [4]

* In file system approach, information is stored in flat files which are maintained by the file system under the operating system’s control.
* Application programs go through the file system in order to access these flat files[4]

# 2.2.2. 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.[2]

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.[2]

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.[2]

# 2.2.3. Database vs. File System Approach

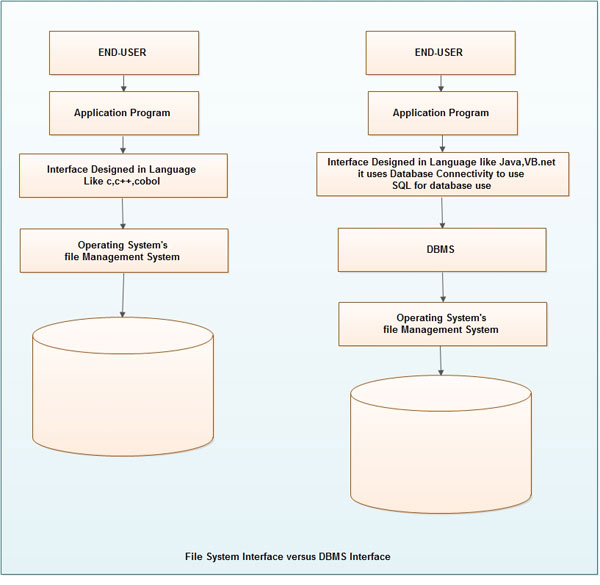


Fig: File System Interface vs. DBMS Interface[2]

Both systems contain a collection of data and a set of programs which access that data.[7]

|  |  |
| --- | --- |
| **DBMS [7]** | **File System [7]** |
| 1. Coordinates both the physical and the logical access to the data | 1. Coordinates only the physical access |
| 1. Reduces the amount of data duplication by ensuring that a physical piece of data is available to all programs authorized to have access to it | 1. Data written by one program in a file-processing system may not be readable by another program |
| 1. Designed to allow flexible access to data (i.e., queries) | 1. Designed to allow pre-determined access to data (i.e., compiled programs) |
| 1. Designed to coordinate multiple users accessing the same data at the same time | 1. Usually designed to allow one or more programs to access different data files at the same time i.e. a file can be accessed by two programs concurrently only if both programs have read-only access to the file |
| 1. Redundancy is controlled | 1. Redundancy is not controlled |
| 1. Unauthorized access is restricted | 1. No means to validate unauthorized access |
| 1. Backup and Recovery possible | 1. Data loss cannot be recovered |
| 1. Provides multiple user interfaces | 1. Data is isolated |

**2.2.3.1. 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. [5]

**2.2.3.2. Reliability**

A reliable DBMS is one that can continue to process user requests even when the underlying system is unreliable, i.e., failures occur [6]

Failures

* Transaction failures
* System (site) failures, e.g., system crash, power supply failure
* Media failures, e.g., hard disk failures
* Communication failures, e.g., lost/undeliverable messages [6]

Reliability is closely related to the problem of how to maintain the atomicity and durability properties of transactions [6]

**2.2.3.3. Efficiency/Performance**

DBMSs are 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 DBMSs 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. [9]

**2.2.3.4. Three Tier Architecture (ANSI/APARC) 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. [10]



Fig: 3-tier Architecture [10]

* **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. [10]
* **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. [10]
* **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. [10]

Multiple-tier database architecture is highly modifiable, as almost all its components are independent and can be changed independently. [10]

2.3. Advantages and Disadvantages of Database

**Advantages:**

* DBMSs 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. [11]

**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. [11]

In conclusion, we can say that situations can exist in which the adoption of a DBMS can be inconvenient: some applications with one or just a few users without the need for concurrent access can sometimes be achieved more profitably with ordinary files rather than with a DBMS. However, DBMS technology has evolved considerably in recent years, resulting in more and more efficient and reliable systems, on more and more widespread and less expensive architecture, so increasing the convenience of developing applications with a DBMS. [11]

2.4. Drawbacks of using file system of store data

There are following problems associated with the File Based Approach:

**1.** **Separated and Isolated Data:** To make a decision, a user might need data from two separate files. First, the files were evaluated by analysts and programmers to determine the specific data required from each file and the relationships between the data and then applications could be written in a programming language to process and extract the needed data. Imagine the work involved if data from several files was needed. [3]

**2.** **Duplication of data:** Often the same information is stored in more than one file. Uncontrolled duplication of data is not required for several reasons, such as:

* + Duplication is wasteful. It costs time and money to enter the data more than once
  + It takes up additional storage space, again with associated costs.
  + Duplication can lead to loss of data integrity; in other words the data is no longer consistent.

For example, consider the duplication of data between the Payroll and Personnel departments. If a member of staff moves to new house and the change of address is communicated only to Personnel and not to Payroll, the person's pay slip will be sent to the wrong address. A more serious problem occurs if an employee is promoted with an associated increase in salary. Again, the change is notified to Personnel but the change does not filter through to Payroll. Now, the employee is receiving the wrong salary. When this error is detected, it will take time and effort to resolve. Both these examples, illustrate inconsistencies that may result from the duplication of data. As there is no automatic way for Personnel to update the data in the Payroll files, it is difficult to foresee such inconsistencies arising. Even if Payroll is notified of the changes, it is possible that the data will be entered incorrectly. [3]

**3. Data Dependence:** In file processing systems, files and records were described by specific physical formats that were coded into the application program by programmers. If the format of a certain record was changed, the code in each file containing that format must be updated. Furthermore, instructions for data storage and access were written into the application's code. Therefore, .changes in storage structure or access methods could greatly affect the processing or results of an application. [3]

In other words, in file based approach application programs are data dependent. It means that, with the change in the physical representation (how the data is physically represented in disk) or access technique (how it is physically accessed) of data, application programs are also affected and needs modification. In other words application programs are dependent on the how the data is physically stored and accessed. [3]

If for example, if the physical format of the master/transaction file is changed, by making he modification in the delimiter of the field or record, it necessitates that the application programs which depend on it must be modified.

Let us consider a student file, where information of students is stored in text file and each field is separated by blank space as shown below:

1 Rahul 1995 Dhakal

Now, if the delimiter of the field changes from blank space to semicolon as shown below:

1; Rahul; 1995; Dhakal

Then, the application programs using this file must be modified, because now it will token the field on semicolon; but earlier it was blank space. [3]

**4. Difficulty in representing data from the user's view:** To create useful applications for the user, often data from various files must be combined. In file processing it was difficult to determine relationships between isolated data in order to meet user requirements. [3]

**5. Data Inflexibility:** Program-data interdependency and data isolation, limited the flexibility of file processing systems in providing users with ad-hoc information requests[3]

**6. Incompatible file formats:** As the structure of files is embedded in the application programs, the structures are dependent on the application programming language. For example, the structure of a file generated by a COBOL program may be different from the structure of a file generated by a 'C' program. The direct incompatibility of such files makes them difficult to process jointly. [3]

**7. Data Security.** The security of data is low in file based system because, the data is maintained in the flat file(s) is easily accessible. For Example: Consider the Banking System. The Customer Transaction file has details about the total available balance of all customers. A Customer wants information about his account balance. In a file system it is difficult to give the Customer access to only his data in the· file. Thus enforcing security constraints for the entire file or for certain data items are difficult. [3]

**8. Transactional Problems.** The File based system approach does not satisfy transaction properties like Atomicity, Consistency, Isolation and Durability properties commonly known as ACID properties.

For example: Suppose, in a banking system, a transaction that transfers Rs. 1000 from account A to account B with initial values' of A and B being Rs. 5000 and Rs. 10000 respectively. If a system crash occurred after the withdrawal of Rs. 1000 from account A, but before depositing of amount in account B, it will result an inconsistent state of the system. It means that the transactions should not execute partially but wholly. This concept is known as Atomicity of a transaction (either 0% or 100% of transaction). It is difficult to achieve this property in a file based system. [3]

**9. Concurrency problems.** When multiple users access the same piece of data at same interval of time then it is called as concurrency of the system. When two or more users read the data simultaneously there is ll( problem, but when they like to update a file simultaneously, it may result in a problem. [3]

For example:

Let us consider a scenario where in transaction T 1 a user transfers an amout1t 1000 from

Account A to B (initial value of A is 5000 and B is 8000). In mean while, another transaction T2, tries to display the sum of account A and B is also executed. If both the transaction runs in parallel it may results inconsistency as shown below.[3]:

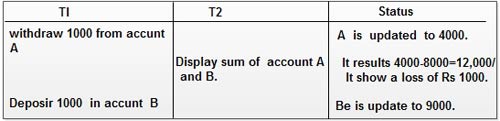


Fig: Inconsistency occurs when both transaction runs in parallel [3]

The above schedule results inconsistency of database and it shows Rs.12,000 as sum of accounts A and B instead of Rs .13,000. The problem occurs because second concurrently running transaction T2, reads A and B at intermediate point and computes its sum, which results inconsistent value. [3]

**10. Poor data modeling of real world**. The file based system is not able to represent the complex data and interfile relationships, which results poor data modeling properties.[3]

2.5. References

1. “Database”. Oklahoma State University. PPT Lecture. url: [www.okstate.edu/cocim/members/kamath/5743/LECTURE%204.ppt](http://www.okstate.edu/cocim/members/kamath/5743/LECTURE%204.ppt) 2015 [27/07/2015]
2. “Database Approach”. Ecomputernotes. Dinesh Thakur. Internet. url: <http://ecomputernotes.com/fundamental/what-is-a-database/database-approach> 2015 [27/07/2015]
3. “Traditional File Processing System”. Ecomputernotes. Dinesh Thakur. Internet. url: <http://ecomputernotes.com/fundamental/what-is-a-database/traditional-file-processing-system> 2015 [27/07/2015]
4. “Traditional Approach for Data Storage and the Need of DBMS”. Careerbless. Internet. url: <http://www.careerbless.com/db/rdbms/c1/traditionalApproach.php> 2015 [27/07/2015]
5. “Data Abstraction”. Techtarget. Ivy Wigmore and Margaret Rouse. Internet. url: <http://whatis.techtarget.com/definition/data-abstraction> 06/2014 [27/07/2015]
6. “Distributed DBMS Reliability”. PDF. url: <http://www.inf.unibz.it/dis/teaching/DDB07/ln/ddb10.pdf> 2015 [27/07/2015]
7. “Project 1— Overview of Database Management Systems”. Internet. url: <http://userpages.umbc.edu/~cgrasso/cmsc461/index_files/461_spring_2011_P1_database_concepts_solutions.htm> 2015 [27/07/2015]
8. “NCHRP Report 666”. Book. Page 57: Line 13. Google Books. Url: <https://books.google.com.np/books?id=8Qaai6gHO6gC&printsec=frontcover> 2010 [27/07/2015]
9. “Fundamentals of Database Systems, 4th Edition”. Ebook. Chapter 1: Page 1.3 : Line 4. R.Elmasri and S.Navathe. url: <http://refg.tu-sofia.bg/~bogi/France/BD/book/Fundamentals%20of%20Database%20Systems,%204th%20Edition.pdf> 2002 [27/07/2015]
10. “DBMS - Quick Guide”. Tutorialspoint. Internet. url: <http://www.tutorialspoint.com/dbms/dbms_quick_guide.htm> 2015 [27/07/2015]
11. “Database Systems – Concepts, Languages and Architectures”. Ebook. Paolo Atnezi, Stefano Ceri, Stefano Paraboschi and Riccardo Torlone. url: <http://dbbook.dia.uniroma3.it/dbbook.pdf> 1999 [27/07/2015]
12. “Fundamentals of Database Systems, 6th Edition”. Ebook. Chapter1: Page 5: Line 35. R.Elmasri and S.Navathe. url: [www.cse.hcmut.edu.vn/~ttqnguyet/CSDL/EbookDB.pdf](http://www.cse.hcmut.edu.vn/~ttqnguyet/CSDL/EbookDB.pdf) 2011 [27/07/2015]