ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

**Maitighar, Kathmandu**

****

**Database Management Systems**

**Assignment #2**

**SUBMITTED BY:**

Victor Kumar Sapkota

013BSCCSIT047

**SUBMITTED TO**

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

**Simplified Database System Organization**

Simplified database system organization and environment is presented in Figure 1. Since it is a simplified representation, the picture speaks itself all about the organization and environment.

Figure 1: Simplified Database System Organization

**Approaches to Management of Data**

There are two types of approaches to data management, which we are going to discuss here in brief.

They are:

1. File system approach
2. Database approach

**File system approach**

File system approach is the traditional approach to data management. This was used when there was no database management software. With the development of DBMS, however, this approach is dwindling significantly, although it is still widely used in many organizations of our country. All data are kept in files in this approach, generally hardcopy, though softcopy is also possible (but for softcopy, anybody would prefer DBMS today). However, file system approach has many problems, e.g.:

* Slow and inefficient access of data
* Greater risks of errors/inaccuracy
* No or minimal sharing of data
* Low level of security
* No backup or difficulty in making backup
* Lack of networking
* Mostly manual

Etc.

**Database approach**

With the inventions of powerful digital technology and of powerful DBMSs, file system approach seems very trivial today. We have powerful database software which is very efficient and secure that can blow the minds of traditional thinkers of data management. Everything is digitized and maintained with strong integrity and security. Sharing and networking are enabled. So, lets summarize these points of database approach:

* Quick access
* Sharing
* Security
* Backup
* Networking
* Negligible risks of errors
* Automated systems/technologies

**Database vs. file system approaches**

**Data abstraction**

Data abstraction takes place in three levels in both approaches, though the mechanism is different in each. However, the level visible to the user is always only the view level. Figure 2 demonstrates the levels of abstraction and a brief idea of each level.

Figure 2: Levels of abstraction

**Reliability**

The reliability of database approach is obviously higher than that of file system approach from our intuition and common logic as well. From the point of view of security, data integrity, and efficiency, we have developed this intuition. However, more details shall be clearer as we study DBMS in further detail.

**Efficiency/ Performance**

The performance of database approach is greater than that of file system approach. More details shall be clearer as we study DBMS in further detail.

**3-Tier ANSI-SPARC Architecture**



Figure 3: 3-Tier Architecture

DataBase Management Systems (DBMS) are very relevant in today’s world where information matters. Most business operations of large companies are dependent on their databases in some way or the other. Many companies use their data analysis methods to leverage the data in their databases and provide better service to customers and compete with their business rivals.

Databases are collections of data that has been organized in a certain way. The term DBMS is a commonly used to refer to computer program that can help you store, change and retrieve the data in your database. Most DBMS software products use SQL as the main query language – the language that lets you interact with and extract results from your database quickly. SQL is the language used to query popular database systems like Oracle, SQL Server and MySQL. Learning SQL and DBMS can help you become a database administrator.

In this tutorial, we’re going to take an overview of the most popular database architecture – the three tier ANSI-SPARC DBMS Architecture.

**DBMS Architecture**

DBMS architecture is the way in which the data in a database is viewed (or represented to) by users. It helps you represent your data in an understandable way to the users, by hiding the complex bits that deal with the working of the system. Remember, DBMS architecture is not about how the DBMS software operates or how it processes data.

We’re going to take a look at the ANSI-SPARC DBMS standard model. ANSI is the acronym for American National Standards Institute. It sets standards for American goods so that they can be used anywhere in the world without compatibility problems. 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 Architectture is set up into three tiers. Let’s take a closer look at them.

**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. However, the internal level view is still abstract –even if it shows how the data is stored physically, it will not show how the database software operates on it.

So how exactly is data stored on this level? 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. Most DBMS software products make sure that data access is optimized and that data uses minimum storage space. The OS you’re running is actually in charge of managing the physical storage space.

**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 database administrator will have to be conversant with this layer, because most of his operations are carried out on it. Only a database administrator is allowed to modify or structure this level. It provides a global view of the database, as well as the hardware and software necessary for running it – all important info for a database.

**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. Multiple users can work on a database on the same time because of it.

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. Any data that is not needed will not be displayed.

Three “schemas” (internal, conceptual and external) show how the database is internally and externally structured, and so this type of database architecture is also known as the “three-schema” architecture.

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

The ANSI-SPARC standard architecture is three tiered- that is, there are three levels through which it is represented. This three tiered representation offers several advantages, which as follows:

* It makes the database abstract. It hides the details of how the data is stored physically in an electronic system, which makes it easier to understand and easier to use for an average user. It also allows the user to concentrate on the data rather than worrying about how it should be stored.
* It lets users view the same data, but it makes the data customizable to fit each user. Each user should be able to access a source of data, and any change made to their customized data source should not affect data sources being used by other users. This means that the data in your database is independent.
* The three tiered architecture model also allows migration to another system to be seamless. The database appears the same on different systems, even if the physical storage is changed. Its structure remains the same, which makes it portable.
* 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.

Most modern DBMS software products offer these advantages. The goal of this kind of database architecture is to allow a user with little to no knowledge of programming or database administration to operate a DBMS product successfully.

**Advantages and disadvantages of DBMS**

**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 to enforce 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.

**Disadvantage of DBMS**

• 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**

* Slow and inefficient access of data
* Greater risks of errors/inaccuracy
* No or minimal sharing of data
* Low level of security
* No backup or difficulty in making backup
* Lack of networking
* Mostly manual