INTERNATIONAL JOURNAL OF WORLD ACADEMIC RESEARCH

IJOWAR

Volume 1, Issue 2, April, 2015.

www.worldresearchacademy.com

ASSESSMENT OF DATABASE MANAGEMENT IN TERTIARY EDUCATION SYSTEM IN NIGERIA.

Dr. Ikoro, Stanley Ibekwe.
Department of Educational Foundations
Ebonyi State College of Education, Ikwo.
Ebonyi State, Nigeria

Abstract
A database management (DBM) manages the data. The database engine enables data to be accessed, locked and modified and the database schema defines the database's logical structure. These three foundational data elements help provide concurrency, security, data integrity and uniform data administration procedures. Others include according to Gusenbauer, M. (2015), include : **Administration tasks.** A DBMS supports many typical database administration tasks, including change management, performance monitoring and tuning, security, and backup and recovery. Most database management systems are also responsible for automated rollbacks and restarts as well as logging and auditing of activity in databases and the applications that access them, **Storage.** A DBMS provides efficient data storage and retrieval by ensuring that data is stored in tables, rows and columns. **Concurrency control.** In environments where multiple users access and modify the database simultaneously, a DBMS guarantees controlled transaction execution to prevent data corruption or inconsistency.

Keywords: Data, Database, Database Management and Tertiary Education

Introduction

Database management has been variously defined and viewed as, the collection of data which is usually referred to as the database containing information relevant to an enterprise. Also as collection of related data with an implicit meaning and hence is a database. A Database Management System (DBMS) can be viewed as a set of software programmes that controls the organization, storage, management, and retrieval of data in a database. It is equally a set of pre-written programmes that are used to store, update and retrieve a data and accepts requests for data from the application program and instructs the operating system to transfer the appropriate data (Blessinger, and Olle,2004). When a DBMS is used, information systems can be changed much more easily as the organization's information requirements change. New categories of data can be added to the database without disruption to the existing system. Organizations may use one kind of DBMS for daily transaction processing and then move the detail onto another computer that uses another DBMS better suited for random inquiries and analysis. On the other hand, database-management system (DBMS) could be defined as a collection of interrelated data and a set of programmes to access those data. It is a collection of related data with an implicit meaning and hence is a database. The collection of data, usually referred to as the database, contains information relevant to an enterprise. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient. The data must be facts that can be recorded and that have implicit meaning.

The management system is important because without the existence of some kind of rules and regulations it is not possible to maintain the database. They are attributes which should be included in a particular table; the common attributes to create relationship between two tables; if a new record has to be inserted or deleted then which tables should have to be handled (Abraham, Henry, and Sudarshan 2006). These issues must be resolved by having some kind of rules to follow in order to maintain the integrity of the database.

Database systems are designed to manage large bodies of information. Management of data involves both defining structures for storage of information and providing mechanisms for the manipulation of information. In addition, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access. If data are to be shared among several users, the system must avoid possible anomalous results. Because information is so important in most organizations, computer scientists have developed a large body of concepts and techniques for managing data.

Database Management System (DBMS) Applications n Tertiary Institution In Nigeria

A Database management system is a computerized record-keeping system. It is a repository or a container for collection of computerized data files (Abraham, Henry , and Sudarshan, 2006) They asserted that the overall purpose of DBMS is to allow the users to define, store, retrieve and update the information contained in the database on demand. Information can be anything that is of significance to an individual or organization. Database touch all aspects of our lives; however the focus of this term paper is on tertiary institution, where students' information, course registrations, and grades are stored.

Purpose of Database Systems in Tertiary Education in Nigeria

Database systems arose in response to early methods of computerized management of commercial data. It is a method considered part of digitalization of tertiary institution to, among other things , collect data, keep information about all lecturers(teaching staff) and non teaching staff, students, departments, and course offerings, (Gusenbauer,2015). One way to keep the information on a computer is to store it in operating system files. To allow users to manipulate the information, the system has a number of application programmes that manipulate the files, including programs to:

- Add new students, staff(Academic and non academic), and courses
- Register students for courses and generate class rosters
- Assign grades to students, compute grade point averages (GPA), and generate transcripts System programmers wrote these application programmes to meet the needs of the university.

New application programmes are added to the system as the need arises. For example, suppose that a **t**ertiary institution decides to create a new major (say, computer science).As a result, the university creates a new department and creates new permanent files (or adds information to existing files) to record information about all the staff in the department, students in that major, course offerings, degree requirements, etc. The **t**ertiary institutions in Nigeria may have to write new application programs to deal with rules specific to the new major. New application programs may also have to be written to handle new rules in the institution. Thus, as time goes by, the system acquires more files and more application programs.

This typical file-processing system is supported by a conventional operating system. The system stores permanent records in various files, and it needs different application programmes to extract records from, and add records to, the appropriate files. Before database management systems (DBMSs) were introduced, organizations usually stored information in such systems.

The functions of Database Management (DBM) in Tertiary Education System:

A database management (DBM) manages the data. The database engine enables data to be accessed, locked and modified and the database schema defines the database's logical structure. These three foundational data elements help provide concurrency, security, data integrity and uniform data administration procedures. Others include according to Gusenbauer, M. (2015), include :

- **Administration tasks.** A DBMS supports many typical database administration tasks, including change management, performance monitoring and tuning, security, and backup and recovery. Most database management systems are also responsible for automated rollbacks and restarts as well as logging and auditing of activity in databases and the applications that access them.
- **Storage.** A DBMS provides efficient data storage and retrieval by ensuring that data is stored in tables, rows and columns.
- **Concurrency control.** In environments where multiple users access and modify the database simultaneously, a DBMS guarantees controlled transaction execution to prevent data corruption or inconsistency.
- **Centralized view.** A DBMS provides a centralized view of data that multiple users can access from multiple locations in a

controlled manner. A DBMS can limit what data end users see and how they view the data, providing many views of a single database schema. End users and software programs are free from having to understand where the data is physically located or on what type of storage medium it resides because the DBMS handles all requests.

- **Data manipulation.** A DBMS ensures data integrity and consistency by letting users insert, update, delete and modify data inside a database.
- **Data independence.** A DBMS offers both logical and physical data independence to protect users and applications from having to know where data is stored or from being concerned about changes to the physical structure of data. As long as programs use the application programming interface (API) for the database that the DBMS provides, developers won't have to modify programs just because changes have been made to the database.
- **Backup and recovery.** A DBMS facilitates backup and recovery options by creating backup copies so that data can be restored to a consistent state. This protects against data loss due to hardware failures, software errors or other unforeseen events. In a relational database management system (RDBMS) -- the most widely used type of DBMS -- the API is structured query language (SQL), a standard programming language for defining, protecting and accessing data.

The components of a DBMS

A DBMS is a sophisticated piece of system software consisting of multiple integrated components that deliver a consistent, managed environment for creating, accessing and modifying data in databases. According to Darko-Ampem (2003), the components include the following:

- **Storage engine.** This basic element of a DBMS is used to store data. The DBMS must interface with a file system at the operating system (OS) level to store data. It can use additional components to store data or interface with the actual data at the file system level.
- **Metadata catalog.** Sometimes called a *system catalog* or *database dictionary*, metadata catalog functions as a repository for all the database objects that have been created. When databases and other objects are created, the DBMS automatically registers information about them in the metadata catalog. The DBMS uses this catalog to verify user requests for data, and users can query the catalog for information about the database structures that exist in the DBMS. The metadata catalog can include information about database objects, schemas, programs, security, performance, communication and other environmental details about the databases it manages.
- **Database access language.** The DBMS must also provide an API to access the data, typically in the form of a database access language that can be used to modify data but also create database objects and secure and authorize access to the data. SQL is an example of a database access language and encompasses several sets of commands, including data control language for authorizing data access, data definition language for defining database structures and data manipulation language for reading and modifying data.
- **Optimization engine.** A DBMS can also provide an optimization engine that's used to parse database access language requests and turn them into actionable commands for accessing and modifying data.
- **Query processor.** After a query is optimized, the DBMS must provide a way to run the query and return results.
- **Lock manager.** This crucial component of the DBMS manages concurrent access to the same data. Locks are required to ensure multiple users aren't trying to modify the same data simultaneously.
- **Log manager.** The DBMS records all changes made to data managed by the DBMS. The record of changes is known as the log, and the log manager component of the DBMS is used to ensure that log records

are made efficiently and accurately. The DBMS uses the log manager during shutdown and startup to ensure data integrity, and it interfaces with database utilities to create backups and run recoveries.

- **Data utilities.** A DBMS also provides a set of utilities for managing and controlling database activities. Examples of database utilities include reorganization, RUNSTATS, backup and copy, recover, integrity check, load data, unload data and repair database.
- **Reporting and monitoring tools.** Most DBMS are integrated with reporting and monitoring tools to offer enhanced functionality for managing and analyzing data. Reporting tools generate reports, whereas monitoring tools track various database metrics, such as resource consumption and user activity.

However, keeping tertiary institutions' information in a file- processing system has some major challenges in Nigeria. They include:
Data redundancy and inconsistency. Since different programmers create the files and application programs over a long period, the various files are likely to have different structures and the programs may be written in several programming languages. Moreover, the same information may be duplicated in several places /files, (Irenoa, Tijani, & Bakare, 2012).

For example, if a student has a double major (say, music and mathematics) the address and telephone number of that student may appear in a file that consists of student records of students in the Music department and in a file that consists of student records of students in the Mathematics department. This redundancy leads to higher storage and access cost. In addition, it may lead to data inconsistency; that is, the various copies of the same data may no longer agree. For example, a changed student address may be reflected in the Music department records but not elsewhere in the system.

Difficulty in accessing data. Suppose that one of the **t**ertiary institutions clerks needs to find out the names of all students who live within a particular postal-code area. The clerk asks the data-processing department to generate such a list. Because the designers of the original system did not anticipate this request, there is no application programme on hand to meet it. There is, however, an application programme to generate the list of all students. The university clerk has now two choices: either obtain the list of all students and extract the needed information manually or ask a programmer to write the necessary application program. Both alternatives are obviously unsatisfactory. Suppose that such a program is written, and hat, several days later, the same clerk needs to trim that list to include only those students who have taken at least 60 credit hours. As expected, a program to generate such a list does not exist. Again, the clerk has the preceding two options, neither of which is satisfactory. The point here is that conventional file-processing environments do not allow needed data to be retrieved in a convenient and efficient manner. More responsive data-retrieval systems are required for general use.

Data isolation. Because data are scattered in various files, and files may be in different formats, writing new application programs to retrieve the appropriate data is difficult. Integrity problems. The data values stored in the database must satisfy certain types of consistency constraints. Suppose the university maintains an account for each department, and records the balance amount in each account. Suppose also that the university requires that the account balance of a department may never fall below zero. Developers enforce these constraints in the system by adding appropriate code in the various application programs. However, when new constraints are added, it is difficult to change the programs to enforce them. The problem is compounded when constraints involve several data items from different files.

Atomicity problems. A computer system, like any other device, is subject to failure. In many applications, it is crucial that, if a failure occurs, the data be restored to the consistent state that existed prior to the failure. Consider a program to transfer $500 from the account balance of department A to the account balance of department B. If a system failure occurs during the execution of the program, it is possible that the $500 was removed from the balance of department A but was not credited to the balance of department B, resulting in an inconsistent database state. Clearly, it is essential to database consistency that either both the credit and debit

occur, or that neither occur.
That is, the funds transfer must be atomic—it must happen in its entirety or not at all. It is difficult to ensure atomicity in a conventional file-processing system. Concurrent-access anomalies. For the sake of overall performance of the system and faster response, many systems allow multiple users to update the data simultaneously. Indeed, today, the largest Internet retailers may have millions of accesses per day to their data by shoppers. In such an environment, interaction of concurrent updates is possible and may result in inconsistent data. Consider department A, with an account balance of $10,000. If two department clerks debit the account balance (by say $500 and $100, respectively) of department A at almost exactly the same time, the result of the concurrent executions may leave the budget in an incorrect (or inconsistent) state. Suppose that the programs executing on behalf of each withdrawal read the old balance, reduce that value by the amount being withdrawn, and write the result back. If the two programs run concurrently, they may both read the value $10,000, and write back $9500 and $9900, respectively. Depending on which one writes the value last, the account balance of department A may contain either $9500 or $9900, rather than the correct value of $9400. To guard against this possibility, the system must maintain some form of supervision. But supervision is difficult to provide because data may be accessed by many different application programs that have not been coordinated previously.

As another example, suppose a registration program maintains a count of students registered for a course, in order to enforce limits on the number of students registered. When a student registers, the program reads the current count for the courses, verifies that the count is not already at the limit, adds one to the count, and stores the count back in the database. Suppose two students register concurrently, with the count at (say) 39. The two program executions may both read the value 39, and both would then write back 40, leading to an incorrect increase of only 1, even though two students successfully registered for the course and the count should be 41.Furthermore, suppose the course registration limit was 40; in the above case both students would be able to register, leading to a violation of the limit of 40 students. Security problems. Not every user of the database system should be able to access all the data. For example, in a university, payroll personnel information about academic records. But, since application programs are added to the file-processing system in an ad hoc manner, enforcing such security constraints is difficult.

Conclusion

- The data base management (DBM) offers several key advantages over traditional file-based systems, Patrick ,O'N and Elizabeth O'N, (2001) listed them to include:

- **Data integrity and concurrency.** One of the biggest advantages of using a DBMS is that it lets users and application programmers access and use the same data concurrently while managing data integrity. Data is better protected and maintained when shared using a DBMS instead of creating new iterations of the same data stored in new files for every new application.
- **Central storage.** A DBMS provides a central store of data that multiple users can access in a controlled manner. Central storage and management of data within the DBMS provide the following:
    - Data abstraction and independence.
    - Data security
    - Locking mechanism for concurrent access.
    - An efficient handler to balance the needs of multiple applications using the same data.
    - The ability to swiftly recover from crashes and errors.
    - Strong data integrity capabilities.
    - Logging and auditing of activity.
    - Simple access using a standard API.
    - Uniform administration procedures for data.
- **Data sharing and redundancy.** A DBMS enables efficient sharing between multiple users and applications. Its capability of centralized storage also reduces data redundancy which typically occurs if the same data is stored unnecessarily in multiple locations.
- **Logical and structural organization of data.** Database administrators (DBA) can use a DBMS to impose a logical, structured

organization on the data. It delivers economy of scale for processing large amounts of data because it's optimized for such operations.

- **Data backup and recovery.** With a DBMS, users are spared the hassle of routinely backing up data as it handles backup and recovery automatically. A DBMS restores a database to its initial state in case of a server crash or other system malfunction.
- **Multiple views.** A DBMS can also provide many views of a single database schema. A view defines what data the user sees and how they see the data. The DBMS provides a level of abstraction between the conceptual schema that defines the logical structure of the database and the physical schema that describes the files, indexes and other physical mechanisms the database uses.
- **System modification.** A DBMS lets users modify systems much more easily when business requirements change. A DBA can add new categories of data to the database without disrupting the existing system, thereby insulating applications from how data is structured and stored.

However, DBMS perform many functions. It uses more memory and CPU than a simple file storage system, and different types of DBMS require different types and levels of system resources, which is better for Tertiary education system in Nigeria.

Recommendations

The staff of tertiary education system should be trained on proper database management (DBM) procedures, and be able to function on the three foundational data elements which helps to provide concurrency, security, data integrity and uniform data administration procedures.

References

Abraham, S., Henry F. K., and Sudarshan S, (2006) *Database System Concepts.* Fifth edition, Asia: McGraw-Hill Education

Blessinger, K., & Olle, M. (2004). Content analysis of the leading general academic databases. *Library Collections, Acquisition and Technical Services*, 28(3), 335–346.

Darko-Ampem, K. O. (2003*). Study of the Policies and use of data base and Practices in African University* . University of Stirling: Scholarly Publishing .

Gusenbauer, M. (2015). Comparing the sizes of 12 academic search engines and bibliographic databases. Scientometrics, 1–38.https://doi.org/10.1007/s11192-018-2958-5

Irenoa, K. O., Tijani, R. I., & Bakare, O. (2012). Enhancing Library Services Delivery in the 21st Century in Africa: the Role of Cloud Technologies. *International Journal of Library and Information Science Studies,* 4(4), 1–9.

Patrick ,O'N and Elizabeth O'N, (2001) *Database Principles, Programming and Performance*, Harcourt Asia: Pte. Ltd.