ISSN: 2393-8048
International Advance Journal of Engineering, Science and Management (IAJESM)
Multidisciplinary, Multilingual, Indexed, Double Blind, Open Access, Peer-Reviewed, Refereed-International Journal.
SJIF Impact Factor =8.152, July-December 2025, Submitted in July 2025

An Examination of The Function of Database Systems in The Educational Sector

Punam Dilip Patil, Ph.D Research Scholar, Department of Computer Science, Shri Jagdishprasad Jhabarmal Tibrewala University, Jhunjhunu, Rajasthan
Dr. S. K. Yadav, Department of Computer Science, Shri Jagdishprasad Jhabarmal Tibrewala University, Jhunjhunu, Rajasthan

Abstract

Database management systems (DBMS) are crucial for the progression of information technology, facilitating the storage, retrieval, and manipulation of vast data across several fields. This review article provides a thorough analysis of the history, evolution, current trends, and future challenges in database management. The DBMS curriculum includes various disciplines such as data modeling, implementation, and analysis. This course offers a robust basis for IT students to analyze, design, and implement a database system. Besides fundamental skills, IT students must be proficient in diagnosing database performance issues when the system does not meet end-user expectations. In this regard, database tuning involves several tactics designed to improve database performance and is crucial for providing an effective database curriculum. Nonetheless, these competencies are frequently lacking in database management systems curricula.

Furthermore, the study investigates issues pertaining to scalability, security, privacy, and ethical considerations in the era of ubiquitous data. This comprehensive research seeks to furnish scholars, practitioners, and enthusiasts with essential insights into the history, present condition, and future of database administration.

Keywords: Database management systems, information technology, database tuning, ethical concerns

Introduction

A database is a sector within Information Technology that analyzes electronic data processing, encompassing creation, modification, organization, and systematic data management to achieve designated objectives. A database consists of a collection of interconnected files that collectively form information referred to as data. The database is administered by a Database Management System (DBMS), allowing multiple users to access and modify the contents. The database functions to improve efficiency and accessibility. The database design aims to optimize information processing efficiency. Owing to the concepts of primary keys, indexing, searching, sorting, and additional functions. Moreover, an efficient database necessitates user-friendliness. The database is stored on a dedicated PC, thereby optimizing space by centralizing data storage instead of dispersing it across all client PCs. All transactions entering and exiting the database records must be processed through this Dedicated-PC Server. All information sources and processed data are unified under a singular database, assuring a reliable level of accuracy. All information and data processing results are aggregated in one spot. Enhance data administration. Big data is one of the most coveted domains in corporate software development and improvement today. The ubiquity of big data is a socio-technological phenomenon propelled by the rapid and ongoing expansion of information volume. The advancement of the education sector and its facilities necessarily conforms to the utilization of database management system technology. This may be relevant and advantageous to the roles of supervision, implementation, and evaluation in the execution of continuing education or sustainability education. Continuing education seeks to augment basic education and prepare students to participate as community members who may nurture reciprocal ties with their social, cultural, and natural settings, while also developing further skills for the workforce or advanced studies.

Historical development of database management Primitive databases and file systems

In the early days of computing, data organizing and storage were rudimentary procedures.

Computers primarily relied on basic file systems, analogous to digital versions of physical filing cabinets. These file systems allowed users to create folders and store data within them. However, managing data in this manner was limited in terms of structure and retrieval capabilities. Users had to navigate directory structures to locate specific files, leading to inefficiencies and challenges in data management.

Despite these limitations, early databases and file systems were essential in the developmental stages of computing. They provided a mechanism for individuals and enterprises to store and access information, albeit in a fairly primitive fashion akin to contemporary database management systems.

Emergence of Relational Database Management Systems (RDBMS)

The evolution of database management experienced significant development with the introduction of Relational Database Management Systems (RDBMS). The introduction of RDBMS in the 1970s revolutionized data management and manipulation. RDBMS is a structured approach to data storage that use tables, rows, and columns instead of flat files.

The relational model, conceived by Edgar Codd, provided a solid framework for the logical and efficient representation and querying of data. Data was organized into tables, each comprising rows (records) and columns (attributes). Relationships between tables were established using keys, enabling complex searches and data processing activities.

The emergence of RDBMS marked a transformative shift in database management, offering improved data integrity, adaptability, and scalability. It became the dominant method for structuring data and laid the foundation for modern database systems.

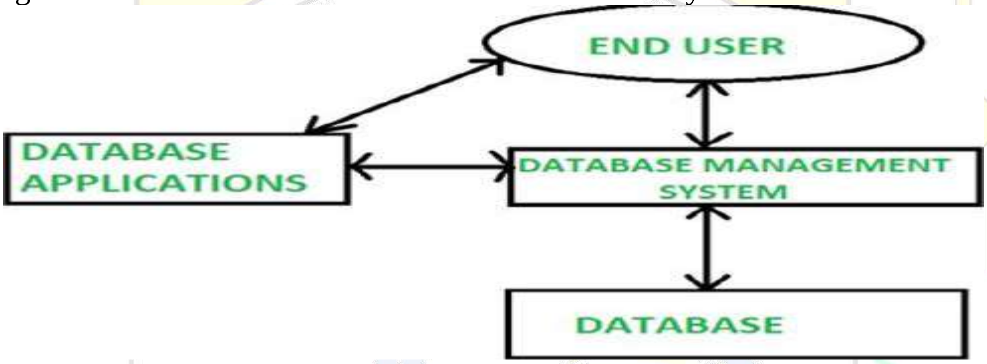


Fig 1: Diagram of relational database management system

Progression of database models - transition from relational to NoSQL

While RDBMS gained prominence for managing structured data, the emergence of new data kinds and applications necessitated alternative database management approaches. This led to the birth of NoSQL (Not Only SQL) databases, which diverged from the traditional relational model.

NoSQL databases were created to support various data types, including semi-structured and unstructured data, which presented challenges for management within the constraints of RDBMS. NoSQL databases eschew the rigid schema constraints of relational databases and embrace a more adaptable data architecture. Examples of NoSQL databases encompass document-oriented databases such as MongoDB, key-value stores like Redis, and graph databases such as Neo4j.

The advent of NoSQL databases represented a significant shift in database technology, offering scalability, performance, and flexibility for modern applications. These databases were employed in web development, extensive data analytics, and real-time data processing, domains where traditional relational databases proved inadequate.

Transformative changes in database technologies

Numerous paradigm shifts driven by advances in computation, storage, and data processing have impacted the development of database technology. Database deployment, administration, and access have all undergone radical changes with the advent of cloud computing.

Scalable and cost-effective solutions for data management and storage are offered by cloud-based database services from businesses like Google Cloud Platform (GCP), Microsoft Azure, and Amazon Web Services (AWS). Businesses can focus on application development and innovation by using cloud databases to relieve infrastructure administration duties.

The use of machine learning (ML) and artificial intelligence (AI) techniques to database administration is a noteworthy advancement. AI-driven databases have the potential to improve performance and efficiency by automating tasks like anomaly detection, data classification, and query optimization. Additionally, businesses may gain important insights from large amounts of data thanks to AI-powered analytics, which improves competitive advantage and makes decision-making easier.

Database Methodology

Regardless of the DBMS provider, the factors listed could have a negative impact on database system performance. The basic implication is that in order to increase performance, the factors must be thoroughly analyzed and evaluated collectively rather than separately. Oracle Corporation has supported this claim with the concept of Total Performance Management (TPM). Unlike a top-down method, the TPM strategy provides a thorough experience that is distinct. It is possible to define the holistic approach as a methodology that tackles problems from multiple angles, allowing for a systematic synthesis of tools and procedures in a dynamic computing environment. Its programs' longevity is attributed to the fact that they were created to address particular needs and are carried out with intention and purpose rather than out of habit. The method elevates the importance of performance management by viewing the database as an entity whose size and complexity grow over time. The process is carried out in three stages.

Principles of database management systems Data models and schemas

Imagine a huge container that holds everything you own. Separate sections or categories, such as one for clothing, another for books, and so on, can be used to keep things organized. Data models and schemas are used to organize data in a database management system (DBMS).

Similar to blueprints, data models specify the structure and organization of data in a database. One popular data architecture that arranges data into tables with rows and columns is the relational model. Every row represents a distinct record or entry, and every column represents a specific attribute or fact.

For example, the table "Books" in a bookshop database might have columns labeled "Title," "Author," "Genre," and "Price." A unique book is indicated by each row in the table, and specific details about that book are contained in each column.

The database's logical structure, including tables, columns, inter-table relationships, and data limitations, is defined by schemas. Think of schemas as the rules that specify how data should be arranged and stored within a database.

For instance, our bookstore database's design might require that each book's "Title" column in the "Books" table be unique in order to avoid duplicate titles. Effective data management, retrieval, and storage are made possible by data models and schemas, which offer a framework for data organizing and structuring.

Query languages and optimization

Asking questions to obtain specific information is known as querying in a database. Query languages are used to communicate with a database and retrieve the required information, much like when you ask a librarian to find a specific book in the library.

A well-liked query language designed specifically for interacting with relational databases is called SQL (Structured Query Language). SQL makes it possible to carry out commands to update existing records, remove unneeded data, insert new records, and retrieve data from tables.

For instance, you could create a SQL query like this to get a list of books from our bookshop database:
SQL code
SELECT BOOKS BY TITLE, AUTHORITY, AND PRICE
WHERE Genre = 'Fiction'; This query would pull from the "Books" table the title, author, and cost of every book in the "Fiction" genre.

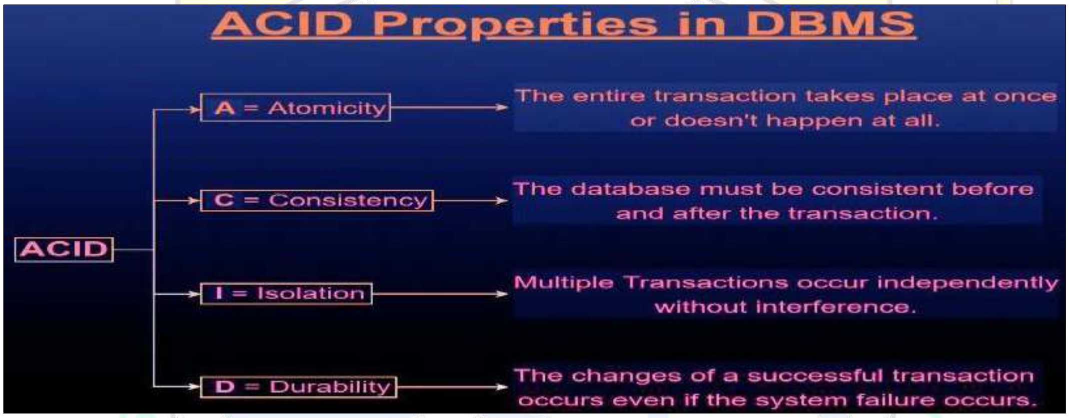
The process of improving database query performance to guarantee as quick and effective execution as possible is known as query optimization. Like a librarian organizing books by author or genre for convenience, a database optimizer analyzes queries to determine the best way to retrieve data. Database queries are frequently optimized using strategies like query rewriting, which reformulates requests to increase efficiency, and indexing, which creates a data structure to improve data retrieval time.

Effective and efficient user interaction with databases is made possible by query languages and optimization techniques.

Transaction oversight

Transactions in everyday life include things like buying groceries or taking money out of an ATM. Transactions in a database represent a series of actions that need to be carried out as a single unit.

Imagine an online store where a customer plans to purchase a book. The process may involve a number of steps, such as updating the inventory to reflect the sale and deducting the book's price from the customer's account balance.



ACID Properties in DBMS

The entire transaction takes place at once or doesn't happen at all.

The database must be consistent before and after the transaction.

Multiple Transactions occur independently without interference.

The changes of a successful transaction occurs even if the system failure occurs.

Fig 2: ACID properties

Even in the event of system errors or malfunctions, transaction management ensures that operations are carried out consistently and dependably. The ACID properties are used to achieve this:

a) Atomicity: Assures that either every operation within a transaction is carried out accurately or none at all. The entire purchase transaction in our bookstore scenario is either completed, including the inventory update and the deduction of the book's price, or it is not completed at all.

b) Consistency: Ensures that both before and after a transaction, the database remains in a consistent state. After the inventory data is updated, the total inventory count needs to stay accurate.

c) Isolation: Ensures that transactions stay separate in order to prevent interference. Many customers buying books at the same time shouldn't interfere with each other's transactions.

d) Durability: Ensures that a committed transaction's effects are permanent and withstand system outages. Once a transaction is committed, the changes made by the transaction must remain in effect even if the system fails.

Data dependability and integrity in transaction management inside database operations are ensured by adherence to these ACID standards.

Concurrency control and recovery

The regulation of many users' or transactions' concurrent access to common data is known as concurrency control. It is common for multiple transactions to access and update the same data at the same time in a multi-user database system. Imagine a scenario where two customers are fighting for the last copy of a book from an internet seller. An inventory update is required for both clients' purchases in order to reflect the transaction. Conflicts brought on by poor concurrency management could lead to mistakes or inconsistencies within the database.

Locking and timestamping are two examples of concurrency management methods that are used to control concurrent access and preserve data consistency. By momentarily limiting access to the data being altered, locking prevents concurrent operations. By giving each transaction a unique timestamp, timestamping helps the database management system determine the order in which transactions must be finished.

Restoring the database to a logical state following a system failure or error is known as recovery management. It includes techniques like checkpoints, which create snapshots of the database's state at predetermined intervals, and logging, which records changes made by transactions.

Database in higher education

IT graduates receive training and internships that prepare them to create and improve database applications for potential employers. However, graduate students are not equipped to handle issues pertaining to suboptimal database performance—a topic that is still ambiguous and controversial—by the current IT curricula and internships. Students are forced to rely on their own resources because they are unable to use their academic and internship experiences to problem-solving. It appears that most institutions' IT curriculum have mostly ignored the topic. The researcher may draw the conclusion that database tuning is done haphazardly in this situation, lacking a formalized process or theoretical underpinnings that could be incorporated into the course syllabus. One may also argue that in order to shed light on database optimization and enable a more thorough examination of the topic, a wider range of computer courses must be included to the curriculum. Furthermore, it is crucial to remember that vocational technology education and traditional higher education curricula can differ in a subtle way. As a result, the subject—including database optimization—is frequently linked to particular software developers rather than academic fields that could carve out a place in university curricula. Instead of offering vocational training in this context, traditional university education aims to develop students' intellectual capacities in order to maximize their learning capacity. The American Council on Education (ACE), an association of American colleges and universities, stated that its core principles were diversity and inclusivity, acknowledging the social responsibility of higher education and emulating the idea of broad participation.

In this regard, some universities aim to offer a broad curriculum for undergraduate and graduate students, while others decide to focus on their offerings and refine their distinctiveness, claiming that mission specificity will allow them to serve students from diverse backgrounds.

Advantages of DBMS in the education sector

a) It ensures data security

Learning institutions record data on student attendance, teacher attendance, and grades every day. With a data management system, this information is safe online. Through data management, all information in the school is secured. Data management follows guidelines and outlines the necessary improvements to ensure data security. Data management systems only give access to authorized network users. Without a password to the systems, it is not possible to access school information. This way, confidential student information is kept safe and secure.

b) It helps with generating school calendar and events Learning institutions need a flexible calendar to run smoothly. With a reliable data management system, the school administrator can modify the school calendar, keeping the school staff, parents, and students updated on the activities and events the school is planning to have. When parents and students are informed of tournaments and other school calendar activities, they can prepare in advance. Any slight change in the school calendar is reflected in the data management system, and parents and teachers can get a notification on their phones.

It helps you to monitor student performance

Learning institutions can use an Education Management Information System (EMIS) to monitor student performance. EMIS is software that collects and stores data to generate reports after analysing it. Using this tool, teachers can track the day-to-day student progress. It records all information on student library details, exam records, and other personal data like class status and demographic data. This kind of data management system provides essential insights that teachers and school management can use to monitor student performance.

c) It allows efficient management of assignments Distributing student homework and assignments is such a tedious task for teachers. Collecting, reviewing, and getting feedback from students is challenging. If you don't have data management systems in place, it's even worse. With a school database management system, teachers can assign homework and assignments to many students at a go. They can also check assignment deadlines and get real-time feedback from students. Handling assignments and projects is cumbersome for both teachers and students. DBMS makes it easier for students and teachers to handle projects anywhere. This way, they can improve their productivity

Strategies for effective integration of DBMS in education sector

a) Policymakers need to prioritize investing in technology infrastructure in schools and educational institutions. Investing in technology infrastructure will ensure that all students have equal access to technology and bridge the digital divide.
b) Educators need to be trained in the use of technology in education. Training programs should focus on providing educators with the necessary technical expertise to integrate technology into the classroom effectively.
c) The development of technology-based learning materials should be based on pedagogical principles. Technology-based learning materials should be designed to support learning outcomes and be tailored to meet the specific needs of individual learners
d) Educators need to collaborate with technology experts to create technology-based learning materials. Collaboration between educators and technology experts will ensure that technology-based learning materials are effective and support learning outcomes.

Conclusion

The purpose of this study is to evaluate the value of including database tuning into the university's computer information systems program's curriculum. Only specific aspects of database tuning are covered in undergraduate curriculum, according to an analysis of higher education institutions; the problem is not fully covered.

Teaching the problem necessitates a methodical approach that takes into account both macro- and micro-level factors. This is particularly pertinent because database tuning is not limited to just one DBMS vendor product; it also includes a theoretical framework that may greatly engage a topic. It does not, however, negate the necessity of handshaking and actual DBMS instruction.

Teachers must, however, overcome the difficulties associated with incorporating technology into the classroom, including restricted access, a lack of technical know-how, and exorbitant expenses. IT integration in education can produce a more engaging, dynamic, and individualized learning environment that can enhance learning outcomes and close the digital divide with the right tactics and teamwork.

References

1. Abdullah A, Zhuge Q. From relational databases to NoSQL databases: performance evaluation. Res J Appl Sci Eng Technol. 2015;11(4):434-439.

2. Abiteboul S, Hull R, Vianu V. Foundations of databases. Addison-Wesley, 2015.
3. Afolabi AO, Ajayi AO. Performance evaluation of a database management system (a case study of INTERBASE and MySQL). J Eng Appl Sci. 2008;3(2):155-160.
4. Date CJ. Database design and relational theory: normal forms and all that jazz. Apress, 2019.
5. Garcia-Molina H, Ullman JD, Widom J. Database systems: the complete book. 2nd ed. Pearson; 2018.
6. Grolinger K, Higashino WA, Tiwari A, Capretz MA. Data management in cloud environments: NoSQL and NewSQL data stores. J Cloud Comput Adv Syst Appl. 2013;2(1):22. DOI: 10.1186/2192-113x-2-22.
7. Rob P, Coronel C. Database systems: design, implementation, & management. Cengage Learning, 2018.