IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES
ISSN PRINT 2319 1775 Online 2320 7876
 Journal UGC CARE Listed (Group-I) Volume 10,Special Iss 03 2021

An Analytical Study of Database Management Systems: Design, Optimization and Performance Metrics

Dr. Rasika Khandal
Assistant Professor, B.C.A. Department,
Sudha Sureshbhai Maniar College of Computer and Management, RTM Nagpur University
Nagpur, India, Email:raut.rashu@gmail.com

Abstract: The proposed research paper is meant to examine the main features of Database Management Systems (DBMS), their design, optimization, and performance metrics. DBMS supports the overall effectiveness of management of large quantities of data. The paper also gives an elaborate description of how databases are organized, how queries can be optimized and how their performance may be assessed and enhanced. This research makes it possible to comprehend how to [enhance] databases in a real-life situation based on the comparison of numerous systems and methods.

Keywords: Database Management System, Database Design, Query Optimization, Performance Metrics, Data Storage, Indexing, SQL

Introduction:

We handle masses of data on every day basis in the modern world. This information should be stored, administered and retrieved effectively. A Database Management System (DBMS) enters the picture there. A DBMS is a program system, which assists its users to store and categorize data in a manner which they can find, modify, and utilize every time when the need arises.

Almost all spheres of life employ databases: bank, hospital, school, enterprise, even on mobile applications. Take a simple example of a situation where we log in to a website, search some product in the internet or we want to reserve a ticket, we are actually interrelating to a database in the background. However, to ensure that these systems run without some hitches, the database has to be modeled well and that it performs optimally.

The DBMS must be well designed in such a way that the data stored is secure as well as can be retrieved fast. When a database is not optimized, then it may be slow, may consume excessive space and will result into unhappy user experiences. Thus, an important subject in the study of DBMS includes the database design (such as table usage, relationships and keys) and query optimization (such as indexes and efficient SQL queries).

The performance metrics is another element that is of great importance this is what helps one measure the workability of a DBMS. Typical measures are query run time, amount of storage space it consumes and response time of the data. This study of these metrics can help us to gain an increase in the performance of the system.

The paper is directed at the issue of the studying structure of DBMS as well as optimization of their work, their effectiveness measurement. It tries to make the readers comprehend the most effective rules of the development and management of competent databases.

Review of Literature:

Kumar and Sharma (2020) analyzed various techniques of indexing in relational databases and identified that indexing speeds up the execution of the queries since it saves time to search data. The results of their work indicated that a proper selection of indexing strategy is highly relevant to the enhancement of the DBMS performance. In similar ideas, a case study was carried out by Singh and Arora (2018) on the optimization of MySQL queries and they revealed that through the optimization of the queries, faster data retrieval and the time the system responds is improved.

The article by Deshmukh and Joshi (2019) offered a comparison between normalized and denormalized database structure. They have also concluded that though the benefits of normalized databases are averting duplication of data and enhancing data integrity, it can be a bit slower since it has multiple joins. In the same regard, Patel and Patel (2019) discussed the effects of normalization on performance and proposed the relevance of a balanced scheme as too much normalization contributes to the growth of complexity and, at the same time, denormalization may cause efficiencies to be squandered on wasting storage space.

The study by Chatterjee and Banerjee (2018) discussed the effects of SQL indexing and execution plans on the performance of queries. In their study, they said that an index should be applied by a developer in a properly planned way, in this way the database works even when large data sets are to be used. Iyer and Reddy (2019) were concentrating on design approaches that have been embraced by Indian enterprise systems. They concluded that the design of schema structure, normalization as well as intelligent indexing is needed to ensure the stability and high-performance databases in businesses.

Verma and Srivastava (2019) examined different approaches to SQL query optimization and found that such practice as indexing, sub-queries and views could have a great effect on the speed of execution. Jain and Gupta (2018) conducted benchmarking performance on various relational databases. Their research discovered that performance depends on such factors as data amount, the type of query, and level of optimization.

Bansal and Singh (2018) have concentrated on how the normalization can be used to enhance the efficiency in the system and how the normalization avoids the redundancy of data and enables the updates to be done easily. Tiwari and Sharma (2019) demonstrated the impact of database design decisions on the query execution time. They demonstrated that the tables that contain appropriate relationships and indexing are much faster to respond to the queries of the users.

Kulkarni and Joshi (2018) used tests to analyze the various indexing methods and their impact on execution time, and it was confirmed that Clustered indexes and B-tree indexes are more advantageous as they perform better compared to others in most scenarios. Mehta and Tripathi (2020) examined the workload-behavior of databases. They have found that systems that are built rather than optimized work a lot better with heavy loads.

Finally, Patil and Deshpande (2018) evaluated the efficiency of queries and design characteristics of different DBMS software tools (such as MySQL, Oracle and PostgreSQL).

They discovered that indexing as well as structure and software selection determine speed and reliability of the database.

Objectives of the Study:

1. To study the basic structure and design principles of Database Management Systems (DBMS).
2. To understand different techniques used to optimize database performance.
3. To analyze key performance metrics that help evaluate the efficiency of a DBMS.

Hypothesis:

H_0 (Null Hypothesis):
Database designing and optimization strategies (indexing, normalization etc) do not have a major influence on the performance parameters (execution-time, CPU-time, memory consumption) of a Database-Management-System.

H_1 (Alternative Hypothesis):
Use of optimization techniques (indexing, normalization etc.) in the design of the database and the application of appropriate database schema improve the performance metrics (execution time, CPU seconds and memory occupied) of a Database Management System considerably.

Research Methodology:

This research work is comparative and analytical. It concentrates on analyzing various things about Database Management Systems (DBMS), which include structure of design, optimization methods and indicators of performance. The chosen methodology involves the theoretical study and hands-on analysis to aid the study purpose.

1. Research Design:

The research is qualitative and quantitative in nature. It consists of the review of available publications, case study of the most popular DBMS (such as MySQL, PostgreSQL, Oracle) and the performance test with using sample databases.

2. Methods of Data Collection:

Primary Data: Obtained by testing and observation of performance of various DBMS with the help of practical queries, use of indexing, and optimization techniques.

Secondary Data: The data were gathered by use of books, research articles, academic journal articles, the internet as well as websites about the topic DBMS design and performance.

3. The tools and techniques:

Software Used: MYSQL, PostgreSQL, SQLite

- Letters of Testamentary
- Generation of sample databases of various size

- Unoptimized and optimized SQL Query performance
- Timing the time of execution, storage space and CPU utilization
- Analysis Tools: SQL Profiler, Query Execution Plans and Benchmarking tools

4. Variables Studied:

- Independent Variables: Database size, indexing, type of a query
- Dependent Variables: The time taken by query, memory, and CPU load

5. Technique of Data Analysis:

The data obtained is compared based on charts, the performance indicator (such as the time in which queries are run), and graphs to demonstrate how the design and optimization affect DBMS performance.

Analysis of the study:

Table 1: Performance Comparison of Queries with and Without Indexing

Query No.	Query Description	Index Used	Execution Time (ms)	CPU Usage (%)	Memory Usage (MB)
Q1	SELECT * FROM students	No	120	25	15
Q2	SELECT * FROM students	Yes	45	12	12
Q3	SELECT name FROM employees WHERE salary > 50000	No	95	20	14
Q4	SELECT name FROM employees WHERE salary > 50000	Yes	30	10	11
Q5	SELECT COUNT (*) FROM orders	No	105	22	13
Q6	SELECT COUNT (*) FROM orders	Yes	40	13	10

Interpretation:

The table gives the comparison of effect of indexing and non-indexing on database query performance.

Indexes play a huge role in reducing the time of execution thus faster data retrieval.

The utilization of CPU together with memory is also reduced when indexes are utilized, an attribute which demonstrates good utilization of resources.

e.g., Query Q1 was executed in 120ms without indexing and 45ms with indexing, a 62.5 percent improvement.

This is an indication that indexing is an effective technique in optimization that boosts the performance of DBMS.

Conclusion:

In databases supported by indexing, faster query speed, less CPU requirement and less memory usage are experienced.

Table 2: Effect of Database Size on Query Performance

DB Size (No. of Records)	Query Type	Execution Time (ms)	CPU Usage (%)	Memory Usage (MB)
1,000	SELECT * FROM products	25	10	8
10,000	SELECT * FROM products	75	20	12
50,000	SELECT * FROM products	220	30	18
100,000	SELECT * FROM products	480	45	25

Interpretation:

This table demonstrates the impact of the number of the records included in the database on the performance of the queries.

- With the increase in the size of database, the execution time jumps up drastically to 480 ms (100,000 records) as opposed to the 25 ms (1,000 records).
- Memory usage and CPU usage are proportional to the flow of data too.
- This is an indication of the need to have databases that can be scaled.
- Bigger databases require powerful optimization strategy like indexing, data segmentation, and beneficial by design schema to support good performance.

Table 3: Performance Comparison Based on Normalization Level

Query No.	Database Design Level	Query Description	Execution Time (ms)	No. of Joins Used
Q1	Unnormalized	SELECT * FROM customers	40	0
Q2	1NF	SELECT * FROM customers	50	1
Q3	2NF	SELECT orders.id, customers.name FROM orders JOIN customers ON orders. Cust id = customers.id	65	1
Q4	3NF	SELECT o.id, c.name, p.name FROM orders o JOIN customers c ON o. cust id = c.id JOIN products p ON o. prod id = p.id	90	2

Interpretation:

This table gives the comparison of the impact of the level of normalization (ranging between being unnormalized and normalized to the 3NF) on the query performance.

- Unnormalized databases have the virtue of performing faster since there are no joins involved but at the same time are plagued with the issue of data redundancy.
- Execution time gets longer as normalization progresses (1NF -> 2NF -> 3NF) as there is a growing amount of JOIN.
- Whereas normalized databases are slow, they enhance data consistency and integrity.
- The cost of performance is against data quality. Although normalization raises the level of complexity of queries, it makes the data more structured and stable.

Conclusions Overall Results:

The paper has described the main features of Database Management Systems (DBMS), and this has paid special attention to look at the design structure, optimization methods and performance measures. It has been noted explicitly about the efficiency of a DBMS, that the way this system is designed and optimized is very vital to the performance of a DBMS.

Overall Results:

- Indexing contributes significantly towards optimization of execution of queries, optimization of CPU, and memory usage. It was observed that when indexes were used queries were 60-70 percent faster as compared to those which were not indexed.
- With the growth of the database, the performance of the system is likely to be downgraded. Nevertheless, optimization of queries and an effective design assist in keeping the performance levels within reasonable range.
- Database normalization enhances data uniformity and prevents duplication and redundancy but this can make a query very slow because of the table joins that it would have to perform. Thus, normalization and performance have to be balanced.
- Effective metrics of performance of a DBMS in varying circumstances are execution time, processor consumption, and RAM consumption.

Conclusion:

The paper validates that an optimized and well-organised database results in improved performance, scale-ability and reliability. The practice of such best steps as indexing, proper normalization and the optimization of a query can greatly enhance the general performance of any given database system.

Future Scope of the Study:

The work gives the basis to conduct additional studies in the direction of DBMS performance. The future research can be discussed in the following directions:

- Comparative Analysis of the Modern NoSQL Databases like Cassandra, Firebase, MongoDB, are compared with regard to scalability and performance in real-time.
- Influence of cloud-based DBMS on performance indicators of different network states and under loads.
- Application of Artificial intelligence and Machine Learning in DBMS to smart query planning, auto-indexing, and proactive performance tuning.
- Security and Privacy Metrics in DBMS specifically in large scale and distributed systems.
- The practices of Energy Efficiency and Green Computing in large database environments tries to change the consumption of energy and resources.

References:

[1] A. Kumar and R. Sharma, Performance Evaluation of Indexing Techniques in Relational Database Systems, *Proceedings of the International Conference on Computing, Communication & Automation (ICCCA)*, Greater Noida, India, pp. 125-129, 2020. doi: 10.1109/ICCCA.2020.9121064

[2] P. Singh and M. Arora, Query Optimization in Relational Databases: A Case Study on MySQL, *International Journal of Advanced Research in Computer Science (IJARCS)*, vol. 9, no. 5, pp. 62-66, Sept.-Oct. 2018.

[3] S. Deshmukh and A. Joshi, Comparative Analysis of Normalized and Denormalized Database Structures, *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 6, no. 4, pp. 350-355, Apr. 2019.

[4] N. Patel and D. Patel, Impact of Normalization on Database Performance: A Comparative Study, *International Journal of Engineering Research & Technology (IJERT)*, vol. 8, no. 6, pp. 240-243, June 2019.

[5] R. Chatterjee and S. Banerjee, A Study on the Performance of SQL Queries Using Indexing and Execution Plans, *International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, pp. 198-202, 2018 doi: 10.1109/ICICCS51141.2018.9432283

[6] T. R. Iyer and M. K. Reddy, Design Strategies for Efficient Databases: Lessons from Indian Enterprise Systems, *Indian Journal of Computer Science and Engineering (IJCSE)*, vol. 10, no. 2, pp. 123-128, Mar.-Apr. 2019.

[7] R. Verma and A. Srivastava, Optimization Techniques for SQL Queries in Relational Databases: A Review, *International Journal of Scientific & Engineering Research (IJSER)*, vol. 10, no. 3, pp. 451-455, Mar. 2019.

[8] M. Jain and S. Gupta, Evaluation of Relational Database Systems Using Performance Benchmarks, *International Journal of Computer Applications (IJCA)*, vol. 182, no. 12, pp. 21-25, Aug. 2018. doi: 10.5120/ijca2018917649

[9] V. Bansal and R. K. Singh, Database Normalization and Its Impact on System Efficiency, *Journal of Computer Engineering (IOSR-JCE)*, vol. 20, no. 2, pp. 1-6, Mar.-Apr. 2018.

[10] D. Tiwari and R. Sharma, A Study on the Effect of Database Design on Query Processing Time, *International Journal of Advanced Computer Research (IJACR)*, vol. 9, no. 44, pp. 182-188, Dec. 2019.

[11] S. Kulkarni and R. Joshi, Indexing Techniques and Performance Evaluation in DBMS, *Proceedings of the International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, pp. 2050-2054, 2018. doi: 10.1109/ICCES51350.2018.9489113

[12] A. K. Mehta and M. Tripathi, An Empirical Study on the Performance of Database Systems under Varying Loads, *International Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 7, no. 5, pp. 765-770, May 2020.

[13] B. R. Patil and K. Deshpande, Comparative Study of DBMS Software Based on Design and Query Efficiency, *Indian Journal of Computer Science*, vol. 8, no. 4, pp. 225-230, 2018.