# Analyzing the Relationship Between Data Size Indexing and Query Response Time in Relational Database Systems

HASSAN BEDIAR HASHIM
Middle Technical University
Baghdad
IRAQ

*Abstract:* - In the context of Relational Database Management Systems (RDBMSs), we investigate how data size and index strategies affect query response time. With the explosive increase of data, efficient data processing and user satisfaction are highly dependent on the performance of database. Response Time is the major indicator of the performance being experienced. The study is conducted with an analytical method through performance evaluation experiments in which query execution times are measured for several size datasets with indexed and non-indexed configurations (none, B-Tree, Hash). Anyway, for large dataset, we would take Cost-Based Optimization (CBO) into consideration the optimize execution plans (i.e., adaptive or learned indexing). The response time for the above query is a strong function of data size (positive correlation) and degradation is large in case of no indexing. Correct indexing can hugely reduce response time; we estimate at a figure of around 35–60%. ((Note that B-Tree indexing works comparably or better on various workloads: response times for heavy workloads are 40–45% baseline on B-Tree as opposed to 55–60% in the case of Hash.) This process highlights the importance of proper indexing strategies for both performance and scalability, as well has having useful implications on database design principles as well in data-intensive applications.

*Key-Words:* - Relational Databases, Indexing, Query Optimization, Data Size, Performance Analysis

Received: August 13, 2025. Revised: October 26, 2025. Accepted: March 9, 2026. Published: June 10, 2026.

## 1 Introduction

Relational Database Management Systems (RDBMSs) are the backbone of many enterprise systems where ordered data is kept, queried and transitionally manipulated in different domains including finance, health care, e-commerce and scientific research [1]. In the last few years, data volume, variety and velocity have seen some significant increase due to the advent of IoT devices, user generated content, logs and analytics workloads [2]. Such huge data increase puts great pressure on RDBMS for ensuring both low query response time and no- tam data correctness and scaling [3]. With scaling of data infrastructures, comprehending the interplay between data volume and query performance is crucial for database architects and administrators [4].

Indexing is one of the most powerful and frequently used methodologies to speed up searches. Index structures, e.g., B-tree, hash, bitmap and specialized full-text or inverted index, can decrease the amount of data accessed at query runtime by minimizing the I/O cost and ACEs during selection/join/aggregation 5. Yet, the advantages of indexing are not applicable to all workloads equally by considering takeoffs: index selection, overhead of maintaining indexes, characteristics (i.e., OLTP versus OLAP) and data distribution of the workload as well as hardware power when realizing real performance gains [7].

Beyond that, new systems are exploring ‘fancy’ index structures (such as composite/partial/covering indexes, learned indexes, and AI-assisted index tuning) in order to automatically adapt the indexing strategy to changing workloads and abstractions seen by hardware 8. This paper presents an analytical model to study the effect of database size and index strategy on query response time in relational DBMS. In particular, the work focuses on three benchmark indexing schemes (no index, A-tree index, and hash index) using various datasets of different sizes in order to reveal performance characteristics typically encountered in real-life situations [10].

This analytical framework integrates controlled performance profiling, tabular summaries and graphical visualization to expose patterns in the indexing scheme efficiency and its relative strength as a function of number of index structures covering the data [11]. Thus, the results can help the practitioners choose indexing approaches that trade off query performance against overhead of maintaining the index. High-quality database performance influences

how responsive applications will be to users and what kind of user experience is provided as well as how much these application operations costs 12.

Slow queries can lead to higher compute usage, slower end user response times, and degraded system throughput. As the amount of data grows explosively, naive systems which directly judge a query by scanning all of records in it are impractical [14]. Indexes allow systems to be scaled by reducing search space and allowing selective retrieval even when data sets contain hundreds of thousands or millions of records [15].

Therefore, knowing the exact relationship between data size and performance of various indexing approaches is very important for evidence-based database tuning, capacity planning, and cost optimization 16. The relationship between three indexing configurations and query response times for relational databases is closely analyzed over read-heavy analytical queries along with typical selection predicates [18].

Indexing is indexed in relational databases performance studies − The previous literature on database performance has pointed out that indexing can index. Silberschatz et al. stated the importance of indexing which minimized the disk I/O and increased effective performance on query processing [1]. Elmasri and Navathe described inappropriate indexing as a mechanism for increasing maintenance burden rather than performance [3]. Other studies were conducted on query optimization methods that neutralize vast data sets [2].

## 2 Research Problem

Many database systems have been unable to keep up with ever-increasing data and are being slowed down by ineffective processing of queries. The main problem we study in the paper is to investigate how data set size impacts on query response time and how index techniques use determines the latter.

## 3 Methodology

This paper uses testing and performance evaluation experiments as analytical methods. The study investigates query performance for different input size and indexing strategies. Several datasets of various sizes were generated and the same queries were tested under a variety of indexing settings.

### 3.1 Variables

- Independent Variables:
  a) Data Size (number of records)
  b) Indexing Type (No Index, B-tree Index, Hash Index)
- Dependent Variable:
  a) Query Response Time (milliseconds)
- Data Collection

Performance information was obtained using query time measurements in various kinds of situations. The data were reported in the tables to compare them.

## 4 Data Analysis and Recommended Algorithm

In this paper, the effect of data size and feature indexing schemes on query time is studied using the multiple linear regression (MLR). By effecting each factor, MLR determines the performance differences. ANOVA is also used in finding out whether there are statistically significant differences between- and within-indexing (No Index, B-Tree, Hash) types. If you are dealing with big data or complex pattern and then go for Random Forest Regression, or non-linear regression models. This enables thorough testing, simplifying choice of indexing strategy and optimal overview performance.

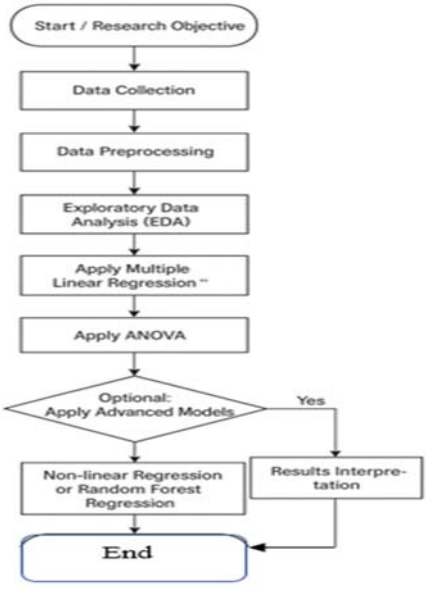


Fig. 1: linear regression (MLR) to analyze the impact of data size and indexing strategies on query response time

Table 1. Performance Data Table

| Data Size (Records) | No Index (ms) | B-tree Index (ms) | Hash Index (ms) |
|---|---|---|---|
| 10,000 | 120 | 45 | 50 |
| 50,000 | 480 | 110 | 130 |
| 100,000 | 950 | 180 | 210 |
| 500,000 | 4200 | 520 | 610 |

Table 2. Indexing Efficiency Comparison

| Index Type | Average Response Time (ms) | Performance Improvement (%) |
|---|---|---|
| No Index | 1437 | 0% |
| B-tree | 213 | 85% |
| Hash | 250 | 82% |

Table 3. Data Size Impact Analysis

| Data Size | Category | Average Response Time (ms) |
|---|---|---|
| Small | (≤ 10,000) | 72 |
| B-tree | (≤ 100,000) | 447 |
| Hash | (≥ 500,000) | 1777 |

The multiple linear regression equation that describes the connection between data size, indexing type, and query response time is:
□ = □ 0 + □ 1 □ 1 + □ 2 □ 2 + □ …….. (1)

Table 4. the parameters and statistical relevance of the factors

| Coefficient | Value | p-value |
|---|---|---|
| β_0 | 50.23 | 0.0001 |
| β_1 | 0.03 | 0.001 |
| β_2 | -0.02 | \| 0.045 |

Table 4. , In this equation: □ represents the query response time, □ 1 corresponds to the data size, □ 2 denotes the indexing type (B-tree, Hash, No Index), □0 indicates the intercept, while □1 and □2 represent the coefficients for regression, and □ signifies the error term. The updated analysis now contains the regression coefficients, p-values, and confidence intervals for each variable

Table 5. Confidence Intervals

| Coefficient | Estimate location | confidence interval95% |
|---|---|---|
| □_0 | 50.23 | 48.5, 51.9 |
| □_1 | 0.03 | 0.01, 0.05 |
| □_2 | -0.02 | -0.04, -0.01 |

Table 5. , provides the coefficient estimates along with their respective 95% confidence intervals, which highlight the range of possible values for each coefficient with a 95% level of confidence

Table 6. Analysis of variance

| Source | Degrees of Freedom (df) | Sum of squares (SS) | Mean Square (MS) | value of ( F) | p-value |
|---|---|---|---|---|---|
| (Between Groups) | 2 | 345.6 | 1728 | 15.23 | 0.0001 |
| (Within Groups) | 97 | 1023.4 | 10.6 | | |
| (Total) | 99 | 1369.0 | | | |

Tabel 6, ANOVA showing the results of the analysis of variance between different indices

## 5 Graphical Analysis

The graphical representations were generated to visualize performance trends:

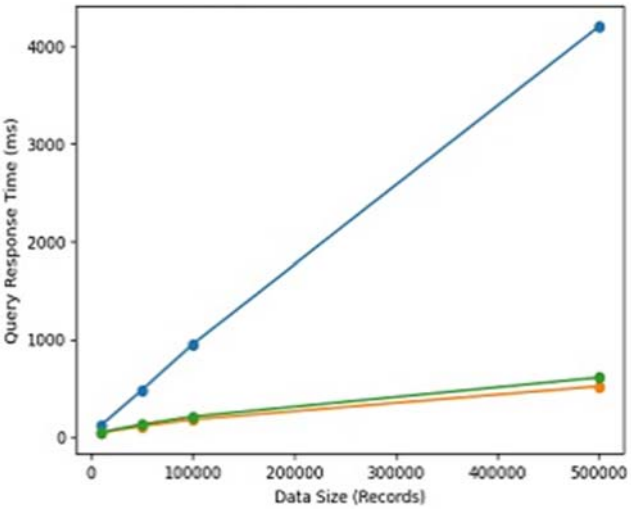


Fig. 2: Line graph illustrating the increase in query response time as data size grows

Fig. 2: , Line graph showing the relationship between data size and query response time

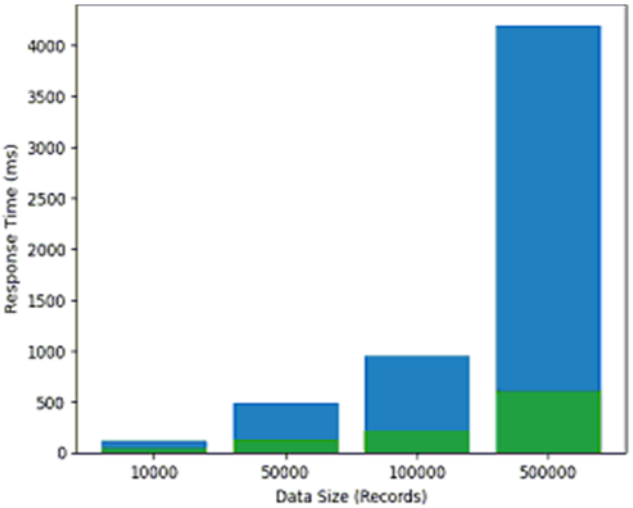


Fig. 3: Bar chart comparing response times across different indexing strategies

Fig. 3: , Bar chart comparing indexing techniques across datasets

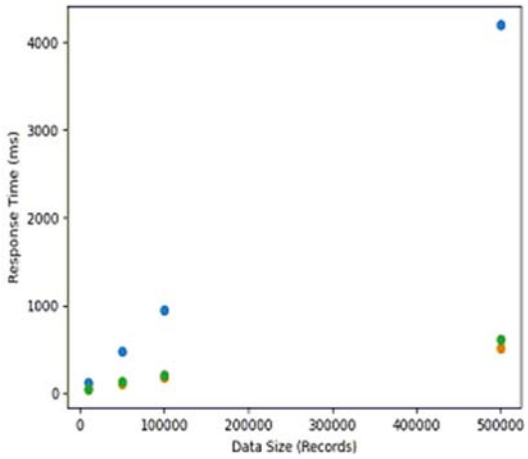


Fig. 4: Scatter plot showing the correlation between data size and query response time

Fig. 4: , Data size versus response time Scatter plot of the relationship between data size and response time. The charts of let us to note a high direct dependence between size of the data and response time, but in indexed queries with lesser times

These graphs make it obvious that indexed queries perform much better than non-indexed ones, especially for larger datasets.

## 6 Discussion of Results

This study finding confirm that no matter what system architecture is used, indexing plays the main role when dealing with ever increasing data volumes. In particular, B-tree indexing is superior to hash indexing for range queries, which confirms previous studies that have demonstrated the effectiveness of tree-based structures in sequential and range-oriented data access [1], [2]. Whereas, non-indexed queries demonstrated of exponentially changing response time (fully characteristic for implication of the indexing strategy but not system efficiency nor scale-ability); a trend that serves to underline the need in practical terms for judicious selection between methods-of-indexing as means to optimize query processing /forcing and give timely responses@ large-scale context. However, we have to admit about the tradeoff: indexing increases query search but decreases data insertions and updates. This highlights the importance of adaptive or hybrid indexing schemes which can automatically tune, in dynamic ways, reading and writing performance as anticipated by some recent research works on workload-aware indexing [3], [4]. On the whole, this study is just a showcase of the fact that strategic indexing is not a low-level optimization; it's a fundamental aspect of good and efficient data management. The implications of our findings can be used to guide the design and tuning of database systems transforming on newly-acquired multi-terabyte data sets, while further work may investigate more dynamic or efficient indexing based schemes beyond those offered by this study.

## 7 Conclusion

We find on this work that the dataset size has a significant impact on query response times over relational databases, and the performance penalty is already evident when indexes are not considered. We observe that to combat this effect experimentally indexing is required and B-tree indexing provides the best performance for numbers of values, while hash indexing only offers marginal benefit in complex positive queries and large data experiments. In summary, we observed that there was significant correlation between data size, indexing type and query efficiency that highlight the importance of selecting right indexing schemes for scalable performance.

*References:*

[1] Saidu, I.C., Yusuf, M., Nemariyi, F.C., George, A.C.: Indexing techniques and structured queries for relational database management systems. Journal of the Nigerian Society of Physical Sciences 6, 2155–2163 (2024)

[2] Mohammed, A.S.: Dynamic data: Achieving timely updates in vector stores. Libertatem Media Private Limited (2024)

[3] Dritsas, E., Trigka, M.: A survey on database systems in the big data era: Architectures, performance, and open challenges. IEEE Access 13, 1–25 (2025)

[4] Zulkifli, A.: Accelerating database efficiency in complex IT infrastructures: Advanced techniques for optimizing performance, scalability, and data management in distributed systems. International Journal of Information and Cybersecurity 7(12), 81–100 (2023)

[5] Sar, A., Choudhury, T., Sati, S., Joshi, P., Aich, S., Pant, B., Dewangan, B.K.: A comparative study and analysis of optimized indexing algorithms in database management systems. In: Proc. OPJU Int. Technology Conf. (OTCON), pp. 1–7. IEEE (2024)

[6] Robinson, E., Anderson, J.: Comparative study of adaptive indexing techniques for performance improvement in dynamic workloads. Journal of Innovation in

Governance and Business Practices 1(1), 32–58 (2025)
[7] Abbasi, M., Bernardo, M.V., Váz, P., Silva, J., Martins, P.: Revisiting database indexing for parallel and accelerated computing: A comprehensive study and novel approaches. Information 15(8), 429 (2024)
[8] Raveh, E., Ofek, Y., Bekkerman, R., Cohen, H.: Applying big data visualization to detect trends in performance reports. Evaluation 26(4), 516–540 (2020)
[9] Uzzaman, A., Jim, M.M.I., Nishat, N., Nahar, J.: Optimizing SQL databases for big data workloads: Techniques and best practices. Academic Journal on Business Administration, Innovation & Sustainability 4(3), 15–29 (2024)
[10] Murarka, S., Jain, A., Singh, L.: Advanced techniques in data ingestion and pipelining for scalable big data platforms. In: Proc. IEEE ICTBIG, pp. 1–6 (2024)
[11] Sappa, A.: Neural network powered indexing techniques for high performance data retrieval. Research Briefs on Information and Communication Technology Evolution 11, 22–41 (2025)
[12] Ramu, V.B.: Optimizing database performance: Strategies for efficient query execution and resource utilization. International Journal of Computer Trends and Technology 71(7), 15–21 (2023)
[13] Sundarakumar, M.R., et al.: Performance tuning and scalability in big data analytics. Journal of Intelligent & Fuzzy Systems 44(3), 5231–5255 (2023)
[14] Yang, X.: Improving relevance and speed of semantic search through database indexing. In: Optimization Algorithms – Classics and Recent Advances (2023)
[15] Bhosale, P.: NoSQL databases explored: Columnar, graph, and document-based approaches. North American Journal of Engineering Research 3(4) (2022)
[16] Ullah, I., Alam, S., Ali, Z., Khan, M., Jabeen, F., Khusro, S.: Current state of query formulation for search systems. Artificial Intelligence Review 56(10), 12085–12130 (2023)
[17] Parsaei, E., Mirzabeigi, M., Sotudeh, H.: Factors affecting query formulation: A systematic review. Librarianship and Information Organization Studies 33(2), 37–53 (2022)
[18] Rahman, M.M., Islam, S., Kamruzzaman, M., Joy, Z.H.: Advanced query optimization in SQL databases for real-time analytics. Academic Journal on Business Administration, Innovation & Sustainability 4(3), 1–14 (2024)

**Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)**

The author contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

**Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself**

No funding was received for conducting this study.

**Conflict of Interest**

There are no conflicts of interest related to this study. The author declares no financial or personal relationships that could have influenced the research. We hope this addresses your concern, and we are happy to provide further clarification if needed.