SUBERVISED BY:
PROF. MAHA A. HASSANEIN

PRO ABILITIES

OPTIMIZING POSTGRESQL QUERY PERFORMANCE: INDEXING STRATEGIES AND CONFIGURATION PARAMETERS

AUTHORS

- OMAR GAMAL
- AHMED AMR
- HABIBA AYMAN
- AHMED SOBHY
- HELANA NADY
- ANAS MAGDY

OBJECTIVE

To determine the best methods for optimizing database query performance and execution time.

WHY

With data growing exponentially, optimizing SQL query performance is essential for fast and reliable data access.

METHODOLOGY

QUERIES

1. SIMPLE SELECT WITH WHERE CLAUSE
2. SELECT WITH WHERE CLAUSE ON CATEGORICAL DATA
3. GROUP BY WITH AGGREGATE FUNCTION
4. SELECT WITH INNER JOIN AND WHERE CLAUSE
5. SELECT WITH MULTIPLE JOINS AND DATE RANGE
6. SELECT WITH MULTIPLE JOINS AND DATE RANGE.
7. LEFT JOIN WITH IS NULL CONDITION.

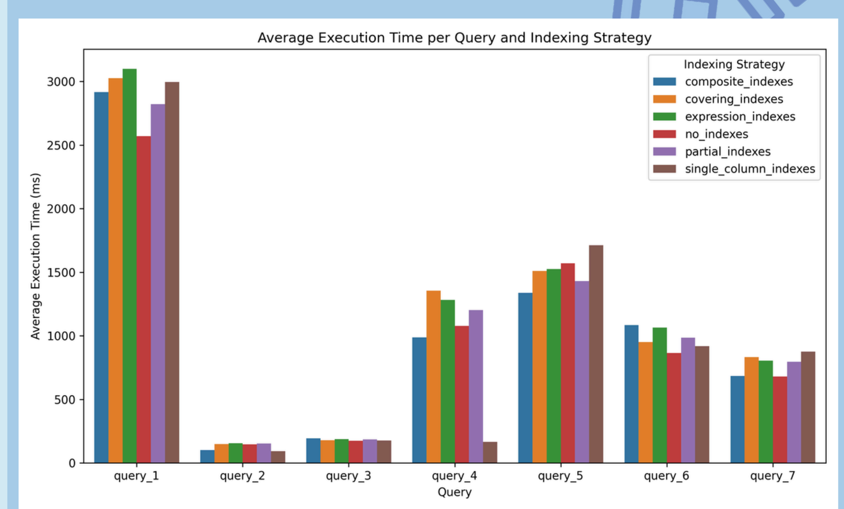
CONFIGURATION PARAMETERS

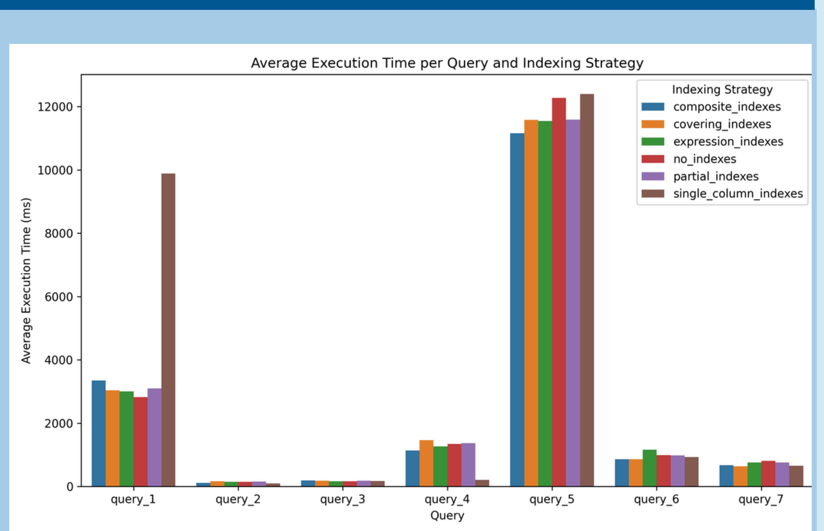
- SHARED BUFFER
- WORK MEMORY
- EFFECTIVE CACHE SIZE
- MAINTENANCE WORK MEMORY
- MAX PARALLEL WORKERS PER GATHER

INDEXING STRATEGIES

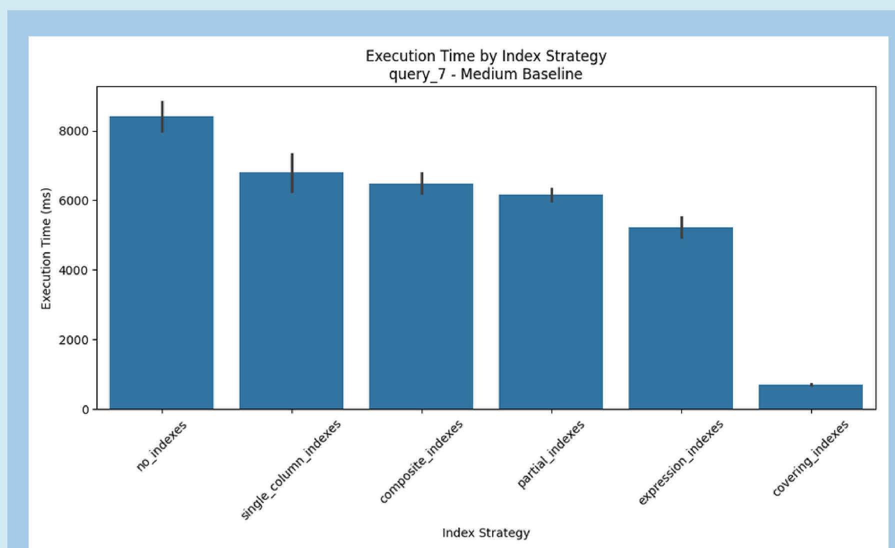
1. NO INDEXES (BASELINE)
2. SINGLE-COLUMN → Indexes on columns frequently used in WHERE clauses.
3. COMPOSITE → Indexes on multiple columns for multi-column searches.
4. PARTIAL → Indexes on subsets of data.
5. EXPRESSION → Indexes on expressions or functions.
6. COVERING → Include additional columns to satisfy query needs without accessing the table.

STATISTICAL ANALYSIS

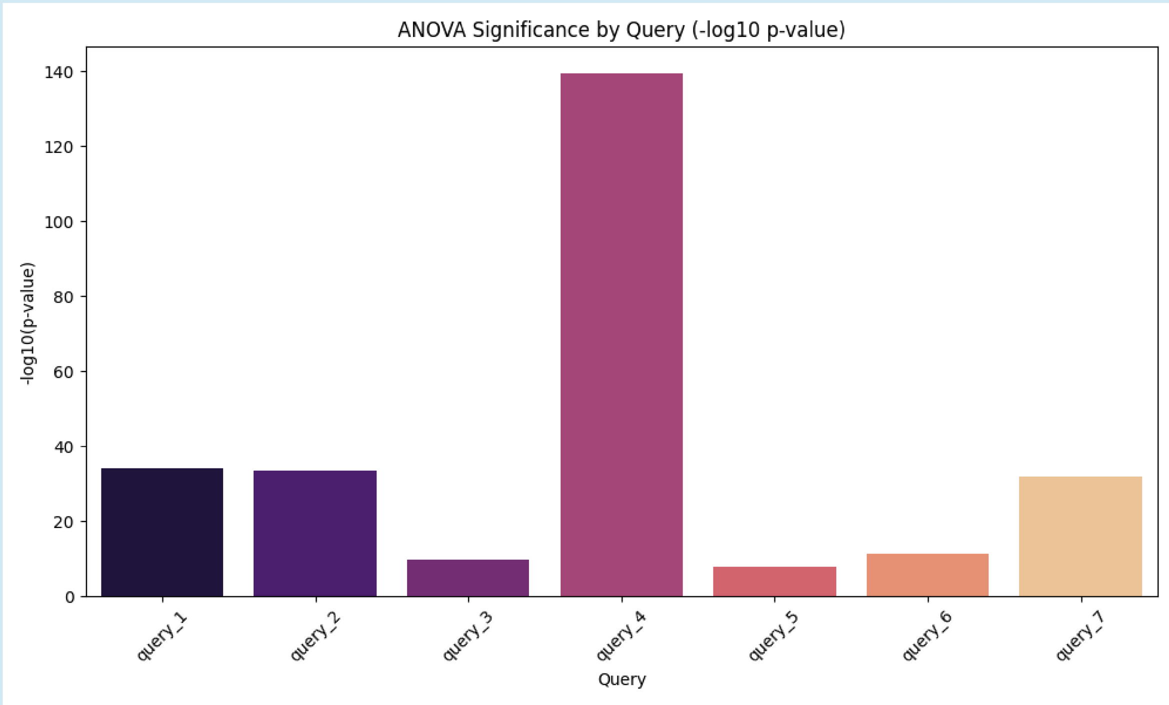




The results show that memory configurations significantly impact query performance. Limited memory leads to inefficiencies, especially for Query 5 and single-column indexes in Query 1. High memory improves performance, particularly for Queries 2, 3, and 4, with covering and composite indexes performing best. Effective indexing remains essential regardless of memory availability.



Here, the performance differences are even more pronounced, with the "no_indexes" strategy taking over 8000ms, while the other strategies range from around 5000ms down to just over 4000ms. This further emphasizes the substantial impact that index strategy selection can have on query performance.



ANOVA Chart Results

The ANOVA chart shows that indexing strategies have a significant effect on the performance of different queries.

Query-Dependent Impact

- Some queries, like Query 4, are highly sensitive to the choice of indexing strategy, showing noticeable performance differences
- Other queries, such as Query 3 and Query 6, show smaller variations in performance across different indexing methods

Key Insight

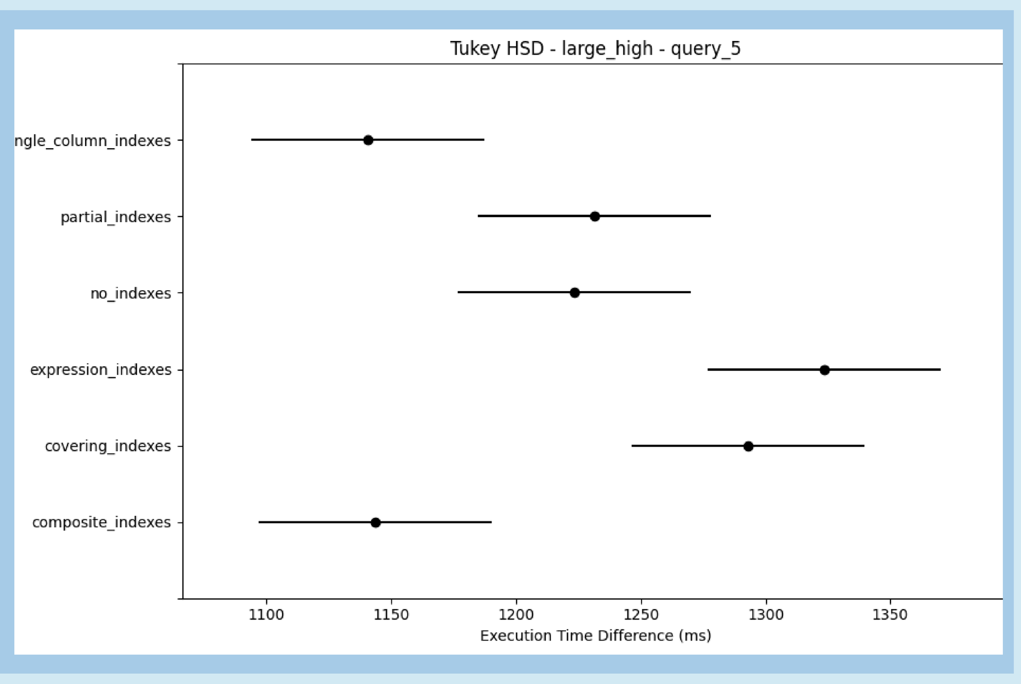
The results suggest that indexing strategies should be tailored for each query, as a one-size-fits-all approach may not optimize performance for all queries.

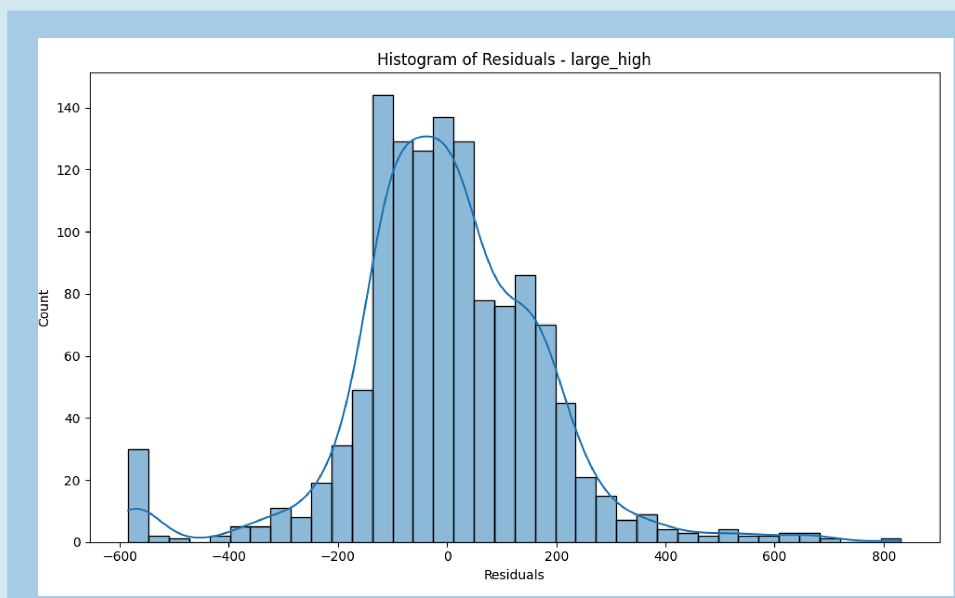
Next Steps

To gain more insights, a post-hoc Tukey test can be used for pairwise comparisons, helping to identify which specific indexing strategies work best for each query

Tukey HSD charts

This analysis confirms the insights from the ANOVA tests, which suggested that the impact of indexing strategies is query-dependent. The Tukey HSD charts further identify the specific indexing techniques that offer the best performance for these two distinct queries, providing data-driven guidance on optimal index selection, answering the statistical question: which indexing strategies work best for which query type ?





The regression analysis revealed a positive relationship between dataset size and query execution time, indicating that larger datasets significantly increase processing time. The model performed reasonably well, with residuals concentrated near zero, though a slight right skew suggests unexplained variance and potential model limitations. Future work should explore non-linear models or additional variables to enhance prediction accuracy and address observed deviations.

REGRESSION Results

QUESTIONS

1. DO INDEXING STRATEGIES SIGNIFICANTLY IMPACT QUERY EXECUTION TIMES?
2. WHAT INDEXING STRATEGIES BEST REDUCE QUERY EXECUTION TIME IN DIFFERENT CONDITIONS?
3. DO QUERY TYPES (E.G., SELECT, JOIN, COMPLEX) SHOW SIGNIFICANT PERFORMANCE DIFFERENCES?
4. HOW DO EXECUTION TIMES CORRELATE WITH DATASET SIZE AND QUERY TYPE?
5. HOW DO DATABASE CONFIGURATIONS (E.G., CACHE SIZE, BUFFER POOL) AFFECT EXECUTION TIMES UNDER DIFFERENT CONDITIONS?

CONCLUSION

This study examined how indexing strategies and database configurations affect query performance in PostgreSQL, using medium and large datasets. Single-column indexes excel for simple queries in medium datasets, while composite and covering indexes are ideal for complex, multi-column queries in large datasets. Execution times vary significantly with indexing strategies, highlighting the need to align indexing choices with query patterns and data characteristics. Additionally, higher memory allocation consistently improves performance by enabling efficient caching and query execution.