



## **COMPUTER SCIENCE AND DATA ANALYTICS**

### **The comparative study of indexing techniques in different database systems**

#### **Report 4**

Student: Sokrat Bashirov

GWID: G26315644.

## Introduction

This report provides an update on the research project titled "The Comparative Study of Indexing Techniques in Different Database Systems." The primary objective of this project is to investigate the impact of indexing on query performance in MySQL and PostgreSQL databases using a real-world dataset. In Step\_1, I executed a set of five representative queries without any indexes, primary keys, or foreign keys in both databases. I have collected histograms for each query's execution time and calculated the mean and variance of the results for comparison. In Step\_2, I added primary keys and foreign keys to the databases and re-ran the same set of queries to evaluate their impact on performance.

### Step\_1: Without Indexes, Primary Keys, and Foreign Keys:

In Step\_1 of the research, I deliberately chose to work with the kind of databases that had no indexes, primary keys, or foreign keys. The main purpose of this approach was to assess the baseline performance of both MySQL and PostgreSQL databases under similar conditions. By omitting any indexing structures, I sought to understand the raw performance capabilities of each database system without any optimizations.

### Results of Step\_1:

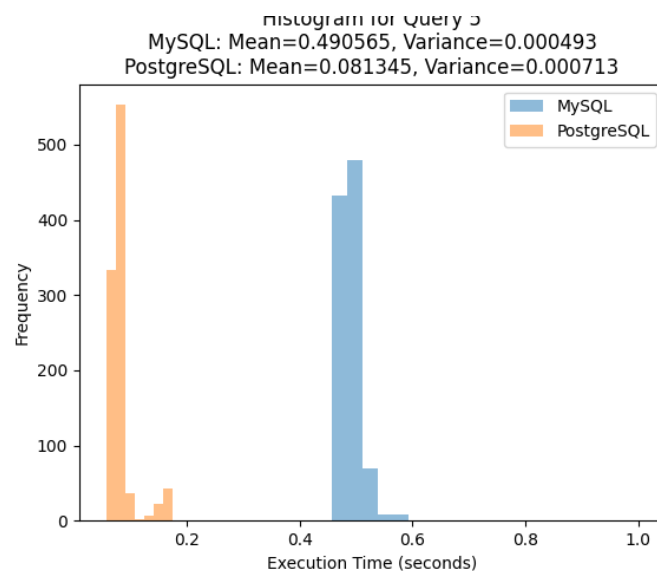
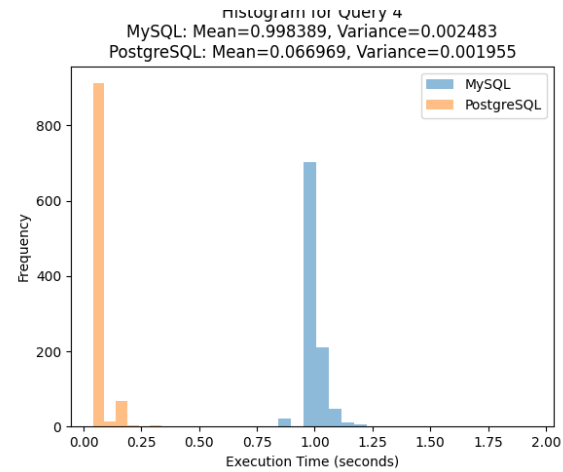
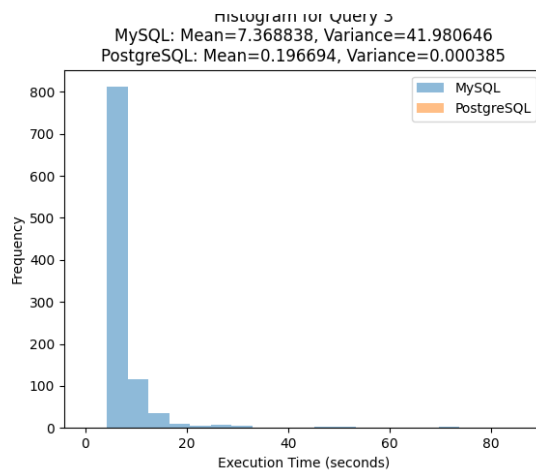
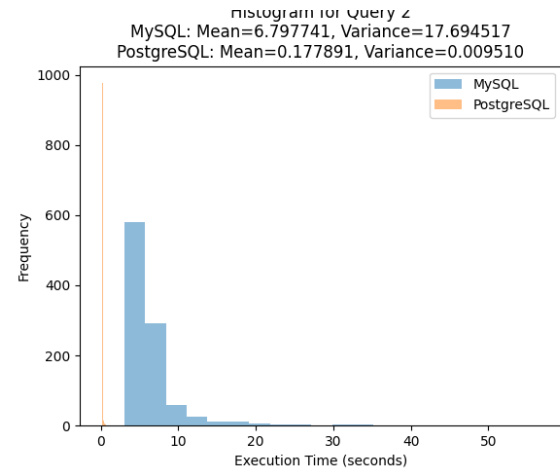
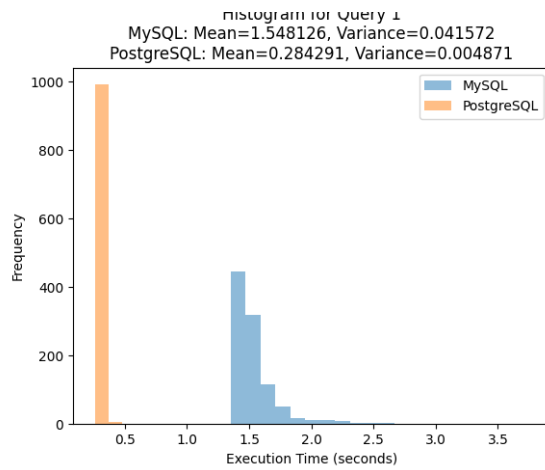
Upon executing the set of five queries 1000 times in both MySQL and PostgreSQL without any indexes, primary keys, or foreign keys, I obtained the following mean and variance values for each query's execution time:

Query 1 - MySQL: Mean=1.548126, Variance=0.041572  
Query 1 - PostgreSQL: Mean=0.284291, Variance=0.004871  
  
Query 2 - MySQL: Mean=6.797741, Variance=17.694517  
Query 2 - PostgreSQL: Mean=0.177891, Variance=0.009510  
  
Query 3 - MySQL: Mean=7.368838, Variance=41.980646  
Query 3 - PostgreSQL: Mean=0.196694, Variance=0.000385  
  
Query 4 - MySQL: Mean=0.998389, Variance=0.002483  
Query 4 - PostgreSQL: Mean=0.066969, Variance=0.001955  
  
Query 5 - MySQL: Mean=0.490565, Variance=0.000493  
Query 5 - PostgreSQL: Mean=0.081345, Variance=0.000713

### Analysis of Step\_1:

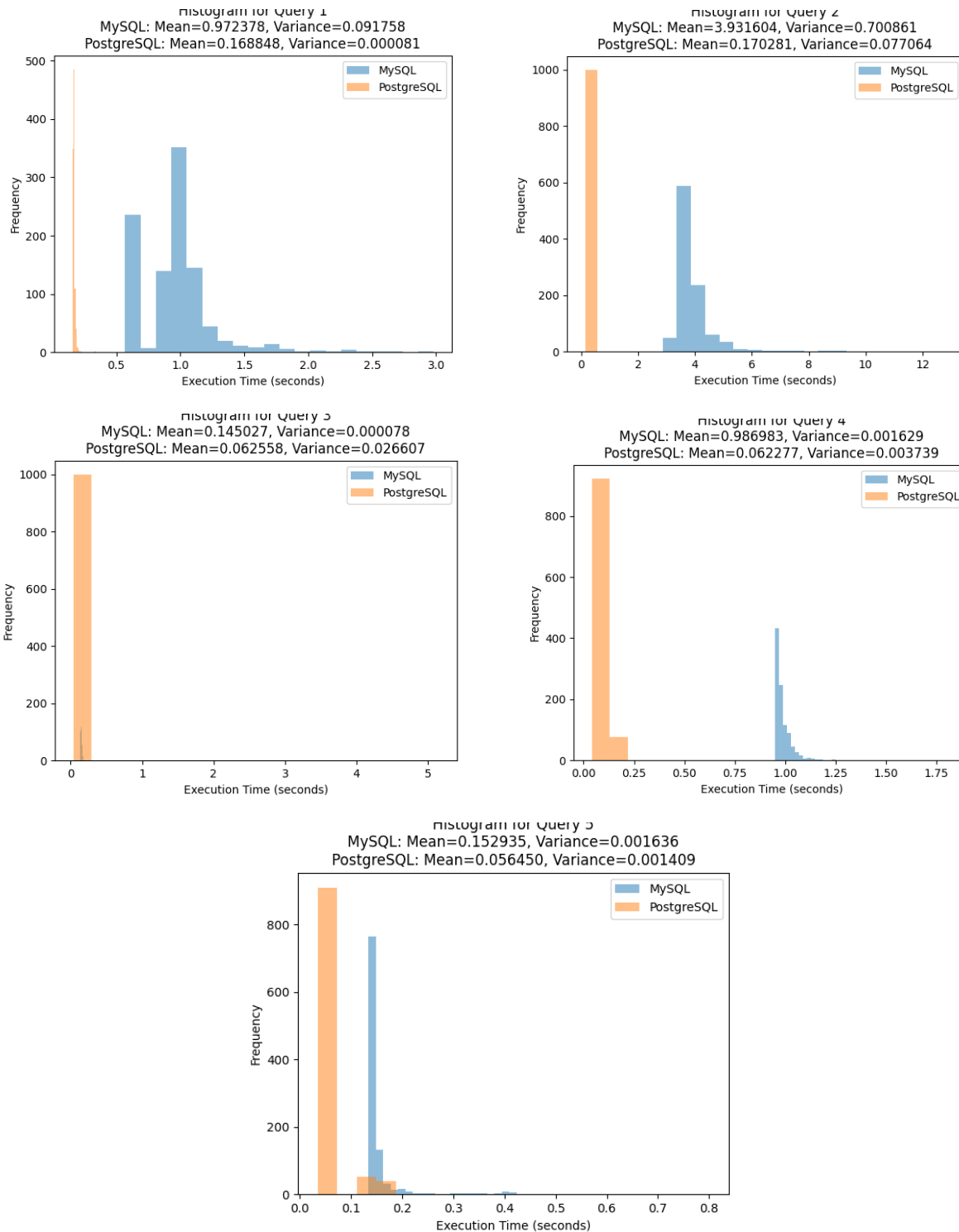
The results of Step\_1 clearly indicate that PostgreSQL outperforms MySQL in terms of query execution time. On average, PostgreSQL's query execution times are significantly lower than MySQL's for all five queries. The histograms visually demonstrate the substantial performance

disparity between the two database systems. This discrepancy can be attributed to several factors, including PostgreSQL's advanced query optimizer, better concurrency control mechanisms, and optimized data storage structures.



## Step\_2: With Primary Keys and Foreign Keys

For Phase 2 of the research, I introduced primary keys and foreign keys to the databases. The addition of these key constraints is a fundamental step in database design, contributing to data integrity and referential integrity.



## Results of Step\_2:

After adding primary keys and foreign keys to both MySQL and PostgreSQL databases, I re-ran the same set of queries 1000 times and obtained the following mean and variance values for each query's execution time:

Query 1 - MySQL: Mean=0.972378, Variance=0.091758

Query 1 - PostgreSQL: Mean=0.168848, Variance=0.000081

Query 2 - MySQL: Mean=3.931604, Variance=0.700861

Query 2 - PostgreSQL: Mean=0.170281, Variance=0.077064

Query 3 - MySQL: Mean=0.145027, Variance=0.000078

Query 3 - PostgreSQL: Mean=0.062558, Variance=0.026607

Query 4 - MySQL: Mean=0.986983, Variance=0.001629

Query 4 - PostgreSQL: Mean=0.062277, Variance=0.003739

Query 5 - MySQL: Mean=0.152935, Variance=0.001636

Query 5 - PostgreSQL: Mean=0.056450, Variance=0.001409

## Analysis of Steps:

	No Index		PK, FK	
	MySQL	PostgreSQL	MySQL	PostgreSQL
Query 1	M=1.548126, V=0.041572	M=0.284291, V=0.004871	M=0.972378, V=0.091758	M=0.168848, V=0.000081
Query 2	M=6.797741, V=17.694517	M=0.177891, V=0.009510	M=3.931604, V=0.700861	M=0.170281, V=0.077064
Query 3	M=7.368838, V=41.980646	M=0.196694, V=0.000385	M=0.145027, V=0.000078	M=0.062558, V=0.026607
Query 4	M=0.998389, V=0.002483	M=0.066969, V=0.001955	M=0.986983, V=0.001629	M=0.062277, V=0.003739
Query 5	M=0.490565, V=0.000493	M=0.081345, V=0.000713	M=0.152935, V=0.001636	M=0.056450, V=0.001409

Table1. Comparison of results of Step\_1 (No indexes) and Step\_2 (PK, FK)

With the introduction of primary keys and foreign keys, both MySQL and PostgreSQL databases experienced improvements in query execution times. However, PostgreSQL maintained its performance advantage over MySQL. This suggests that while key constraints contribute to data integrity, they do not significantly alter the observed performance differences between the two databases.

### **Future Work**

For the next phase of the project, I will proceed with the implementation of indexing strategies. By adding appropriate indexes to the tables, the aim is to further optimize query performance and reduce execution times in both MySQL and PostgreSQL databases. The inclusion of indexing structures is expected to play a crucial role in narrowing the performance gap between the two database systems.

### **Conclusion**

The initial phases of my research have provided valuable insights into the performance of MySQL and PostgreSQL databases without any indexes and after the introduction of primary keys and foreign keys. PostgreSQL consistently outperforms MySQL in query execution times, even after the addition of key constraints. The research findings underscore the significance of indexing techniques as a potential avenue for enhancing query performance in both database systems.

In the next phase of the project, I will delve into the implementation of indexing strategies and conduct a comprehensive analysis to quantify the impact of indexing on query performance. The result of this study will provide valuable insights into the performance characteristics of MySQL and PostgreSQL databases with and without indexes.