# The comparative study of indexing techniques in different database systems

Student: Sokrat Bashirov

Instructors: Steve Kaisler and Jamaladdin Hasanov

Class: CSCI_6917_10

Date: 8/10/2023

# Project Objective

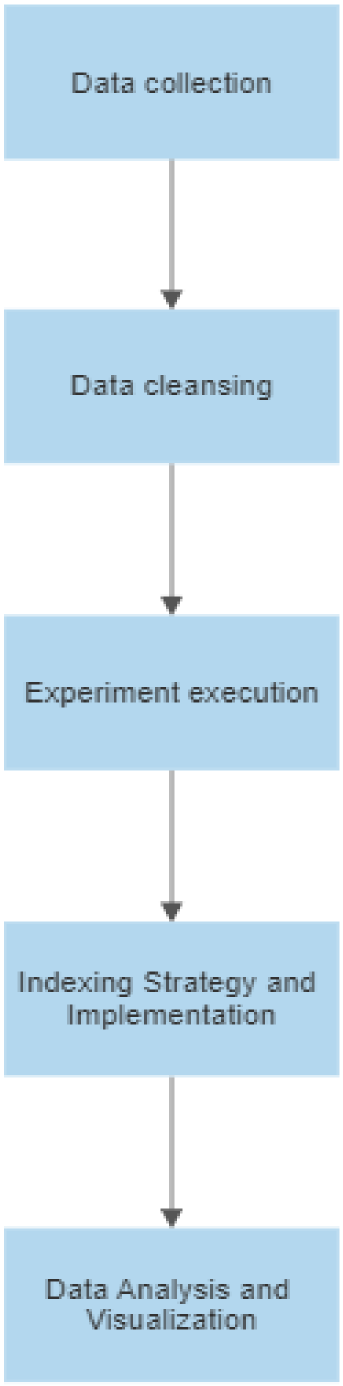
- a comparative study of indexing techniques
- MySQL and PostgreSQL database systems

- compare the query performance of MySQL and PostgreSQL databases with and without indexes to understand the influence of indexing on query execution.
- approach involves systematically testing both databases without any indexes and then introducing indexes incrementally to study their impact.

- valuable to database administrators, developers, and researchers who seek to optimize database performance for their applications.

- a better understanding of the impact of indexing on query execution time, enabling users to make informed decisions about index usage in their databases.

# Research plan



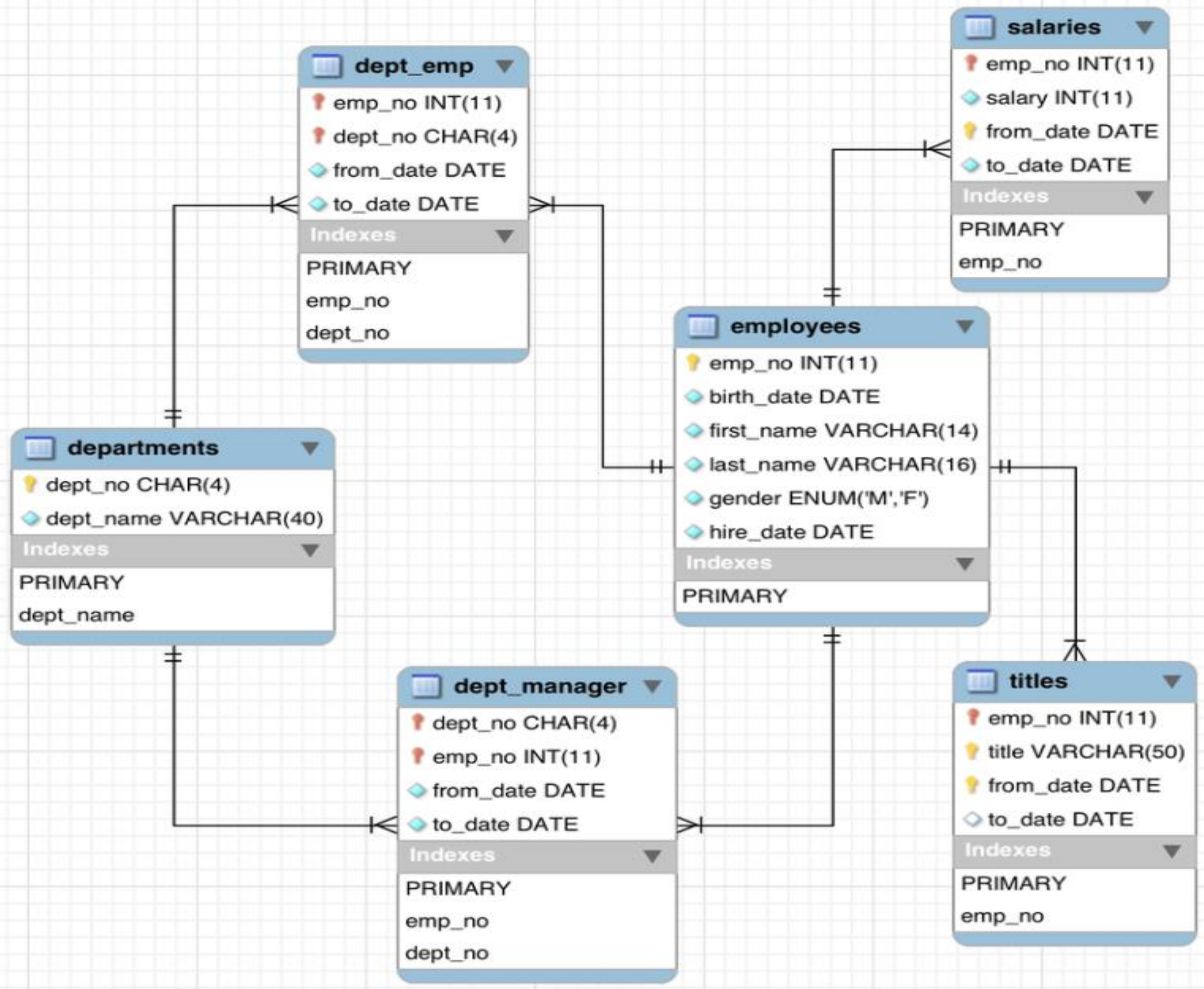
Data collection

Data cleansing

Experiment execution

Indexing Strategy and Implementation

Data Analysis and Visualization



salaries
emp_no INT(11)
salary INT(11)
from_date DATE
to_date DATE
Indexes
PRIMARY
emp_no

dept_emp
emp_no INT(11)
dept_no CHAR(4)
from_date DATE
to_date DATE
Indexes
PRIMARY
emp_no
dept_no

employees
emp_no INT(11)
birth_date DATE
first_name VARCHAR(14)
last_name VARCHAR(16)
gender ENUM('M','F')
hire_date DATE
Indexes
PRIMARY

departments
dept_no CHAR(4)
dept_name VARCHAR(40)
Indexes
PRIMARY
dept_name

dept_manager
dept_no CHAR(4)
emp_no INT(11)
from_date DATE
to_date DATE
Indexes
PRIMARY
emp_no
dept_no

titles
emp_no INT(11)
title VARCHAR(50)
from_date DATE
to_date DATE
Indexes
PRIMARY
emp_no

- SELECT emp_no, COUNT(*) AS count FROM employees GROUP BY emp_no;

- SELECT * FROM salaries WHERE salary = 94443 OR salary = 59571;

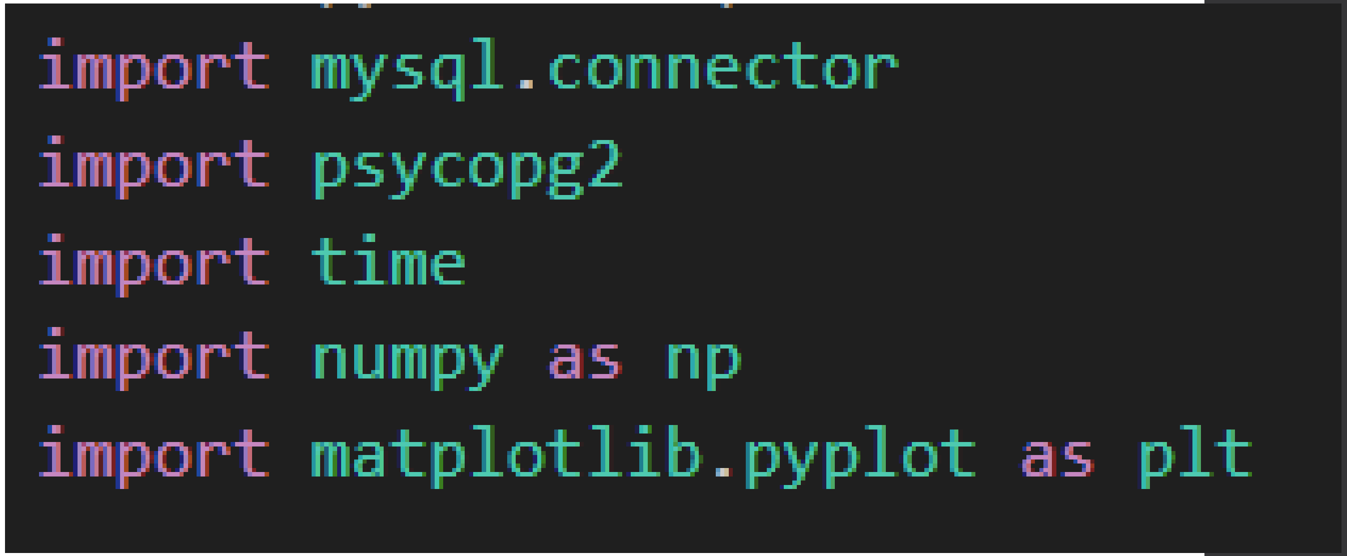
- SELECT E.*, S.* FROM employees E JOIN salaries S ON E.emp_no = S.emp_no WHERE E.first_name = 'Duangkaew';

- SELECT * FROM titles WHERE title LIKE 'senior%';

- SELECT E.*, T.* FROM employees E JOIN titles T ON E.emp_no = T.emp_no WHERE E.first_name = 'Duangkaew';

**Python because:**

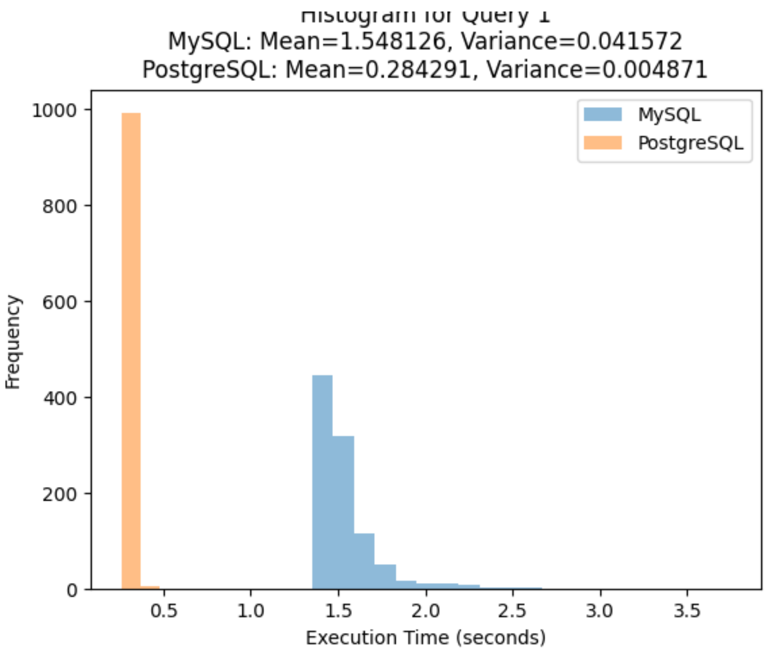
- **Ease of Use**
- **Rich Ecosystem**
- **Data Analysis Capabilities**
- **Database Connectivity**
- **Platform Independence**

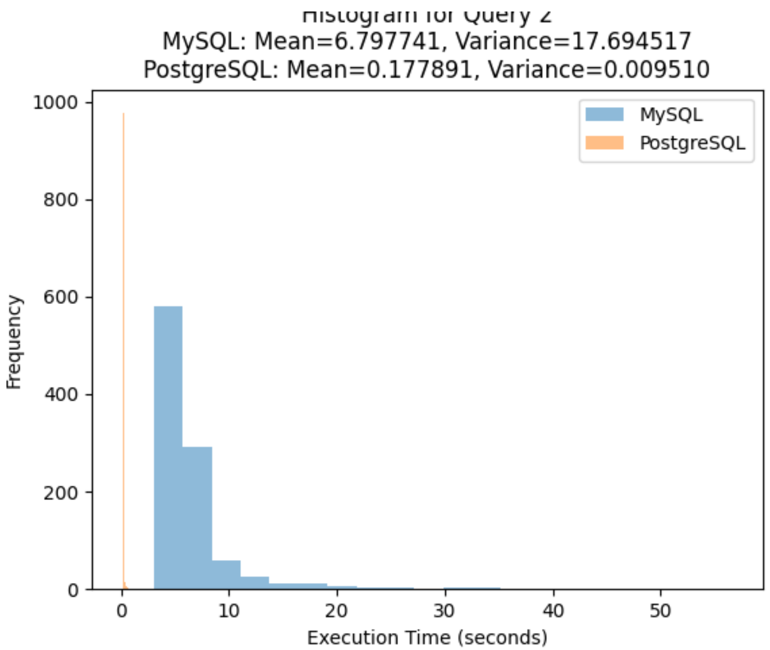


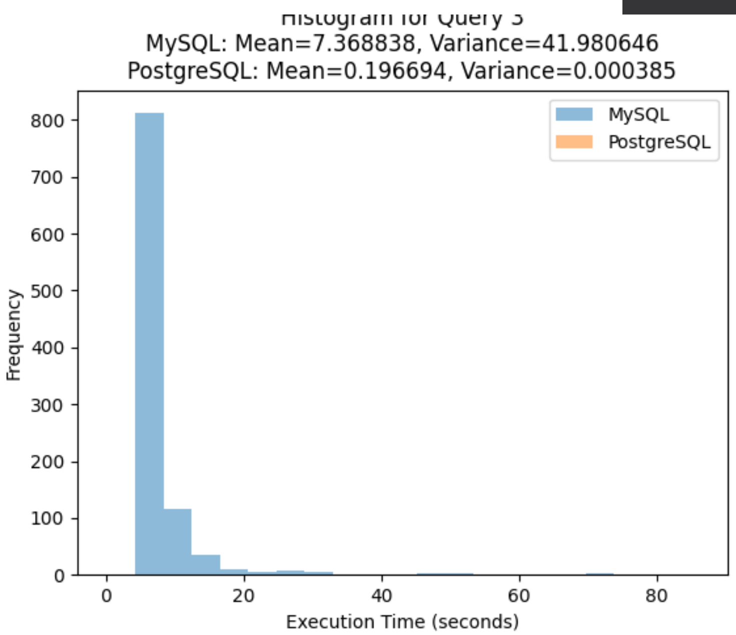
```
import mysql.connector
import psycopg2
import time
import numpy as np
import matplotlib.pyplot as plt
```

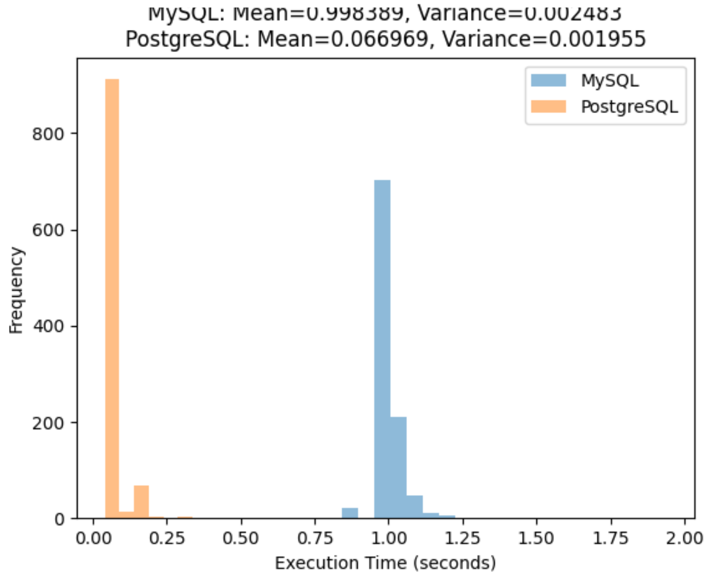
Metrics: Time taken by each query to execute in both MySQL and PostgreSQL databases with and without any indexes.

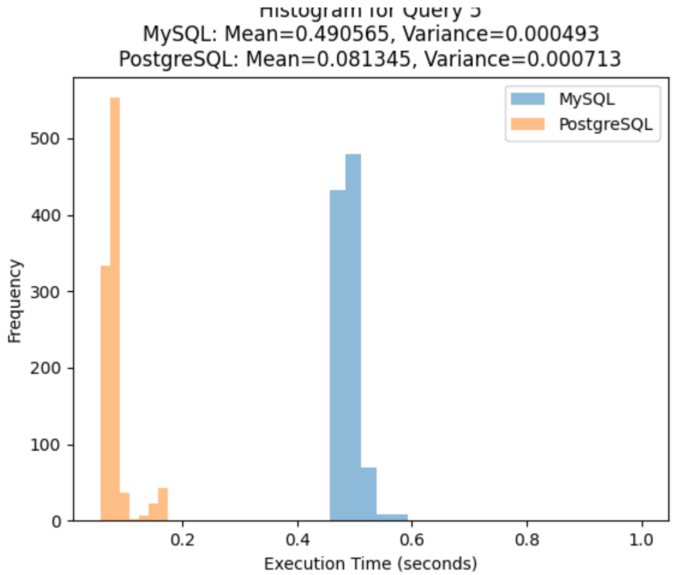
# Query run without indexes



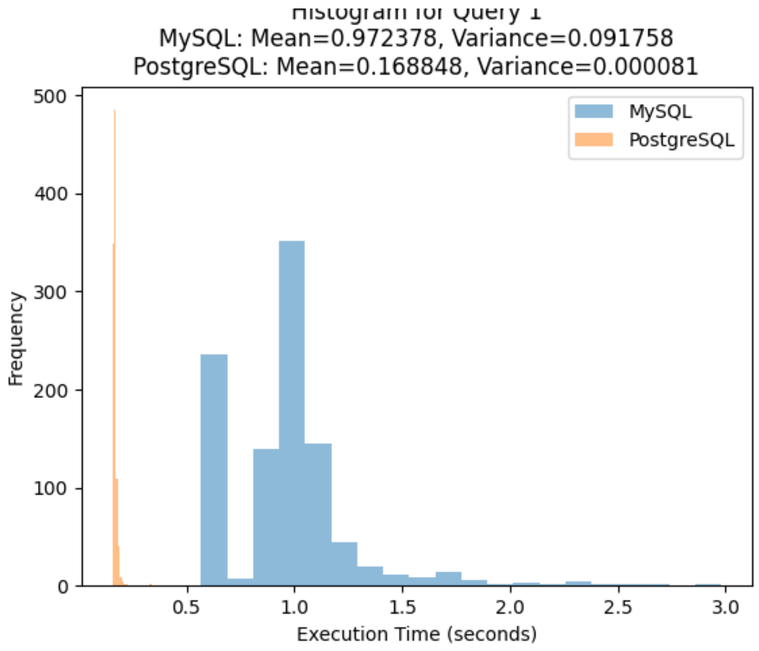




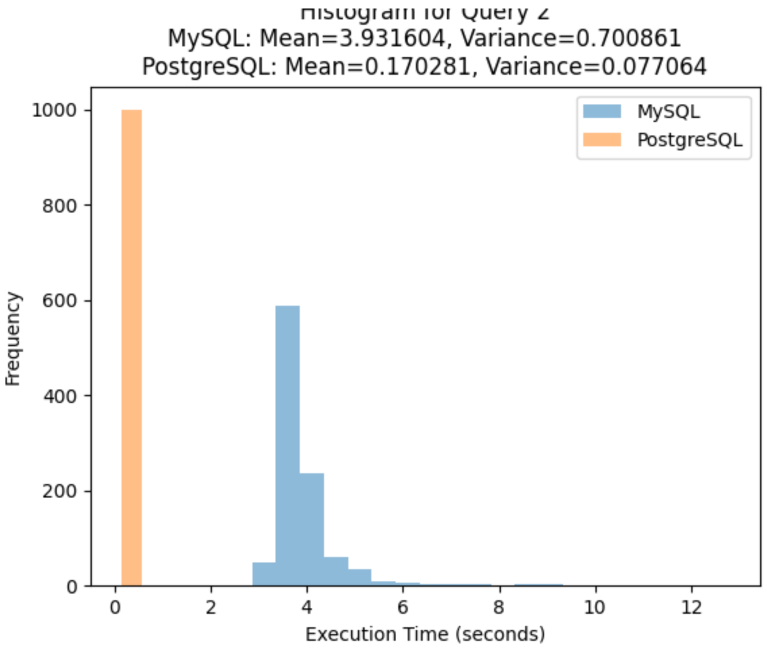




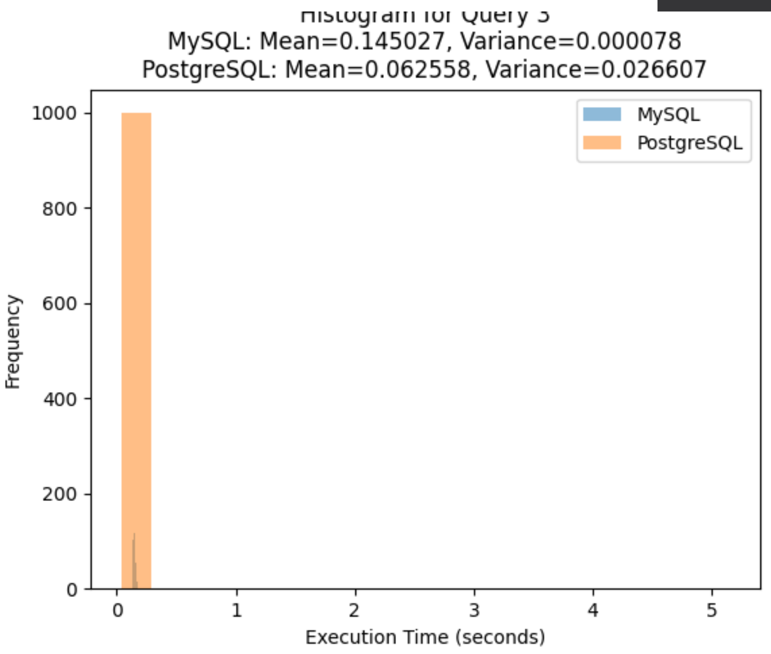
# Query run with PK, FK



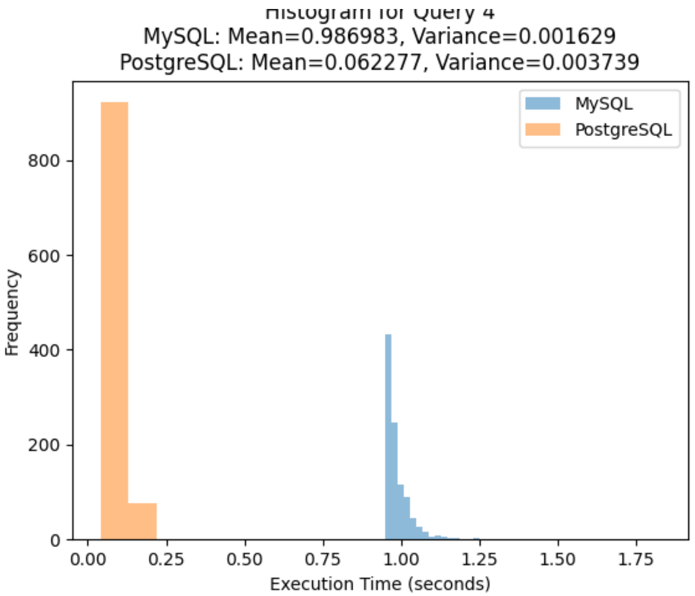
Histogram for Query 1
MySQL: Mean=0.972378, Variance=0.091758
PostgreSQL: Mean=0.168848, Variance=0.000081



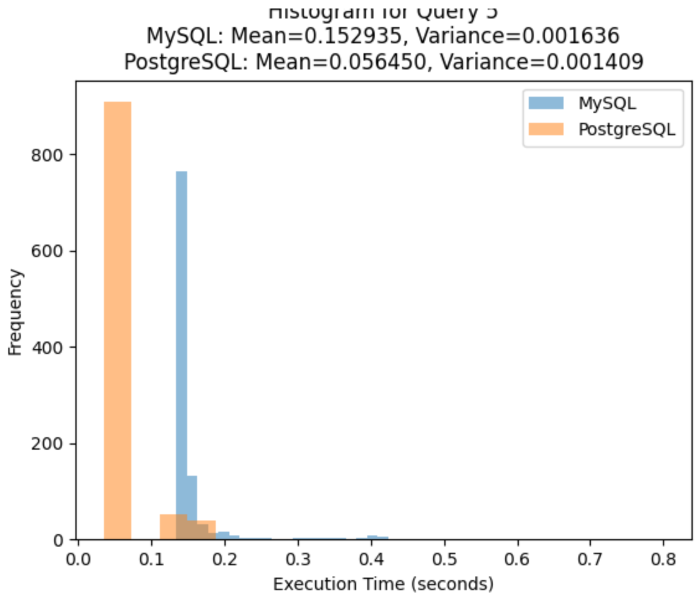
Histogram for Query 2
MySQL: Mean=3.931604, Variance=0.700861
PostgreSQL: Mean=0.170281, Variance=0.077064



Histogram for Query 3
MySQL: Mean=0.145027, Variance=0.000078
PostgreSQL: Mean=0.062558, Variance=0.026607



Histogram for Query 4
MySQL: Mean=0.986983, Variance=0.001629
PostgreSQL: Mean=0.062277, Variance=0.003739



Histogram for Query 5
MySQL: Mean=0.152935, Variance=0.001636
PostgreSQL: Mean=0.056450, Variance=0.001409

# Results

| | 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 |

# Key Findings:

- Without Indexes: PostgreSQL consistently outperformed MySQL in query execution time, indicating its inherent optimization and advanced query processing capabilities.
- With Indexes: The introduction of indexes in both databases led to significant improvements in query execution time, reducing the overall response time for queries.

# Conclusion

- The comparative study of indexing techniques in MySQL and PostgreSQL databases has provided valuable insights into the impact of indexing on query performance. We executed a set of representative queries 1000 times in both databases without indexes and with indexes added, enabling a thorough evaluation of their respective performances.

# Future Work

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.

Future Work Possibilities:

- Fine-tuning Indexing Strategies
- Benchmarking with Other Databases
- Scale Testing

Thank you for your attention!