# Metrics Reporting - DBMS Project

## Introduction

The Metrics Reporting project aims to provide an interactive interface for analyzing query processing metrics. The project utilizes the Streamlit library, along with other Python dependencies, to create a web-based application that allows users to enter SQL queries, execute them, and retrieve relevant metrics for performance analysis.

## Project Overview

The Metrics Reporting project focuses on the following key components:

1. **User Interface:** The project provides a user-friendly interface where users can enter SQL queries and initiate the analysis process. The interface is built using Streamlit, a Python library for creating interactive web applications.

2. **Query Execution:** The project connects to a PostgreSQL database using the `psycopg2` library and executes the provided SQL queries. It captures metrics related to CPU usage, memory usage, execution time, planning time, and throughput.

3. **Metric Computation:** The project calculates various performance metrics based on the query execution results. These metrics provide insights into the efficiency and resource utilization of the query processing.

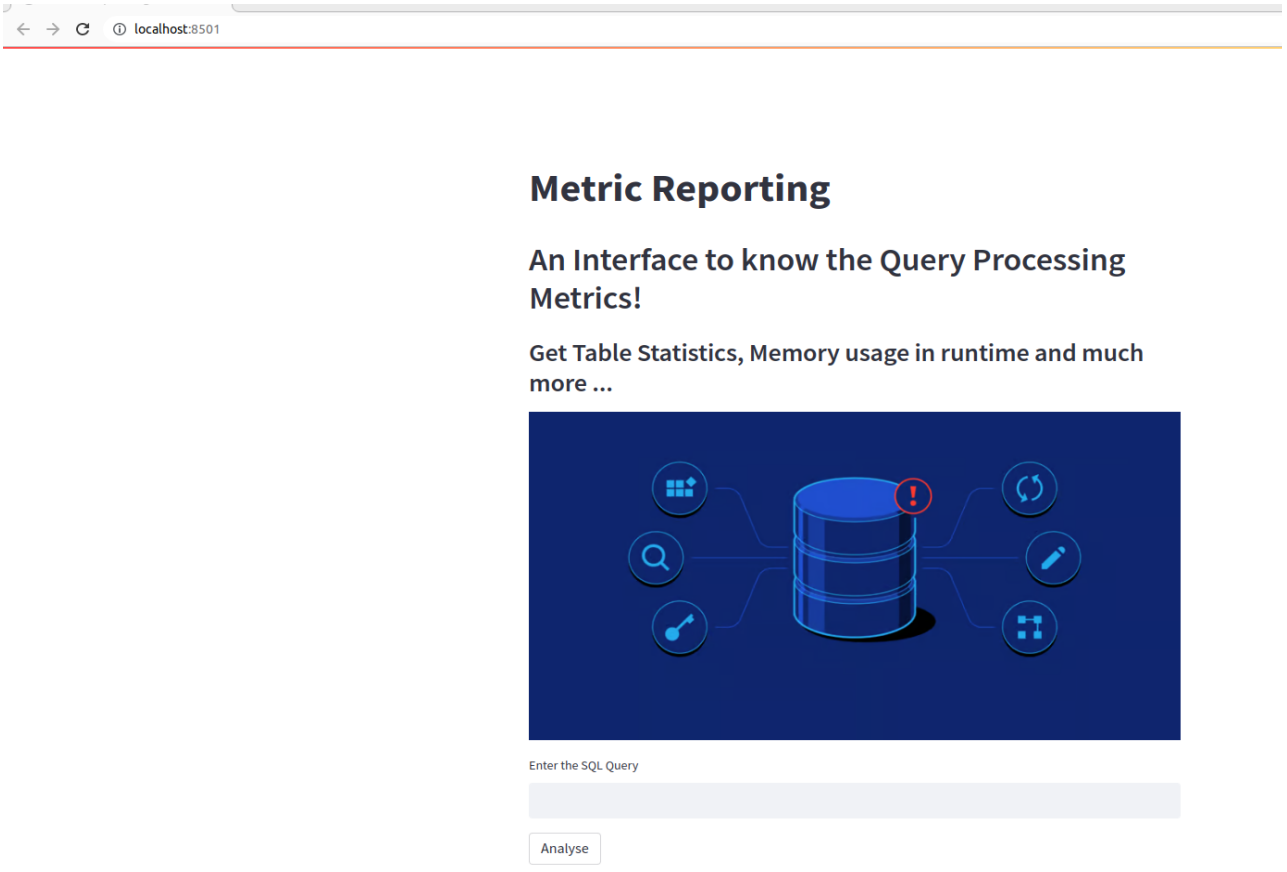
4. **Metric Visualization:** The project uses the `st_aggrid` library to display the computed metrics in a tabular format. This allows users to interactively explore and analyze the metric data.

# Project Workflow

The Metrics Reporting project follows the following workflow:

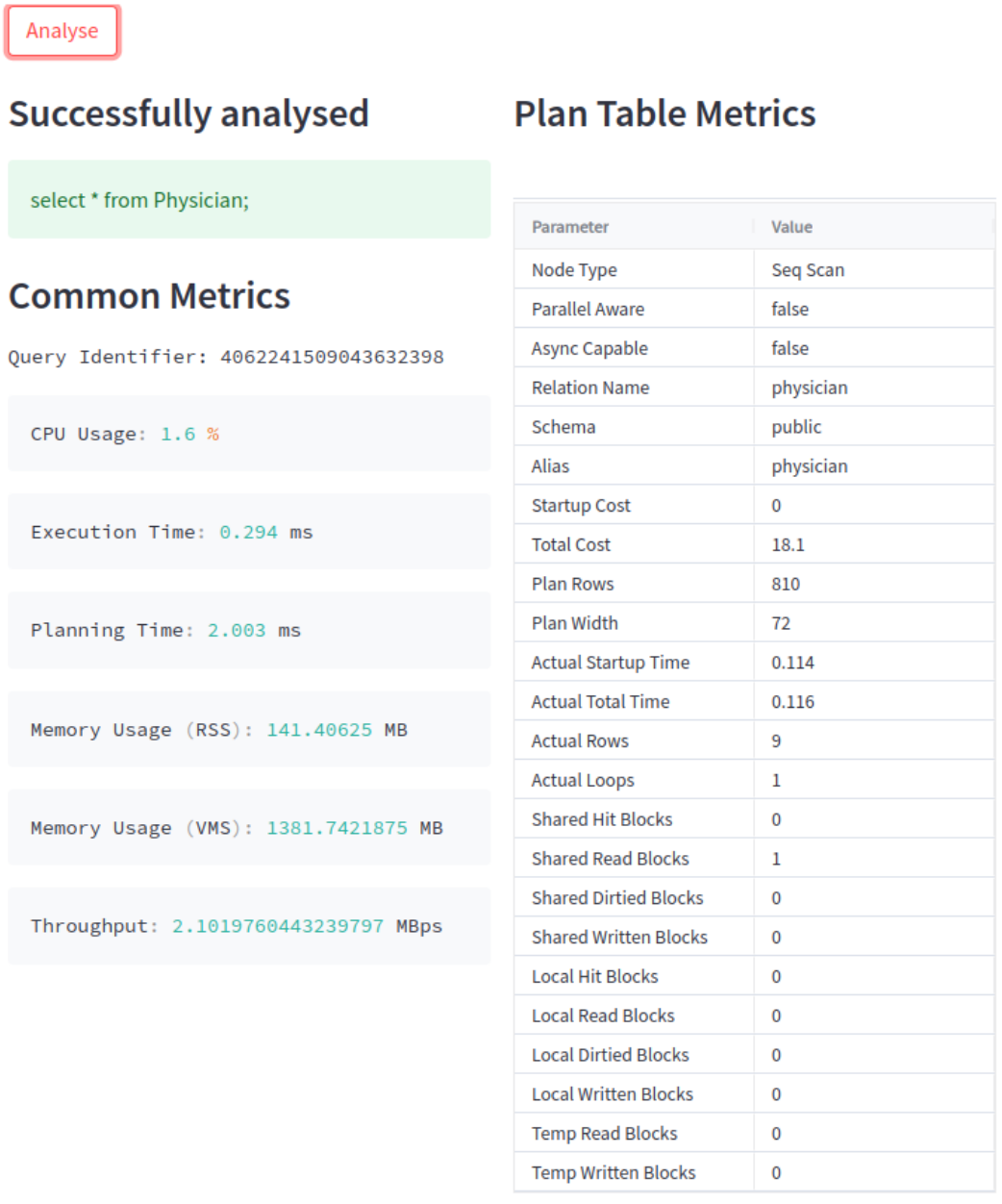
1. **User Input:** Users enter an SQL query into the provided input field in the user interface.

2. **Query Execution:** Upon clicking the "Analyse" button, the project connects to the PostgreSQL database and executes the SQL query using the `psycopg2` library. The query is executed with performance analysis parameters (`EXPLAIN ANALYZE`) to capture detailed metrics.



3. **Metric Calculation:** The project retrieves the query execution results and computes various performance metrics, including CPU usage, memory usage, execution time, planning time, and throughput. These metrics provide valuable insights into the efficiency of the query processing.

4. **Metric Display:** The computed metrics are displayed in the user interface using Streamlit's layout system. The interface shows both common metrics (such as query identifier, CPU usage, execution time, planning time, and memory usage) and plan table metrics (using the `st_aggrid` library).



# Dependencies

The Metrics Reporting project relies on the following Python libraries and dependencies:

- Streamlit: A library for building interactive web applications in Python.
- PIL (Python Imaging Library): A library for image processing tasks.
- Pandas: A data manipulation and analysis library.
- JSON: A library for handling JSON data.
- Psycopg2: A PostgreSQL adapter for Python, used for database connectivity.
- Psutil: A library for retrieving system information and process utilities.
- st_aggrid: A library for displaying interactive grids in Streamlit.

# Conclusion

The Metrics Reporting project provides a powerful and user-friendly interface for analyzing query processing metrics. By leveraging Streamlit and related libraries, users can input SQL queries, execute them, and gain insights into the performance characteristics of their queries. The project's interactive features and visualization capabilities make it a valuable tool for database administrators, developers, and analysts who need to optimize query performance and resource utilization.

Future enhancements to the project could include additional metrics, visualizations, and integration with other database management systems. Furthermore, incorporating user authentication and data persistence could enable collaborative analysis and long-term tracking of query performance.

By combining the power of Streamlit, PostgreSQL, and performance analysis techniques, the Metrics Reporting project offers a valuable solution for optimizing query performance and improving overall database efficiency.