



Report 1: Problem Diagnosis

1. Introduction

We have observed significant latency in database response times across various system modules. This delay negatively impacts the end-user experience and decreases overall system efficiency.

2. Data Analysis Scope

To diagnose the root cause, we extracted a sample of **100 diverse SQL queries** from the production-like environment. We analyzed key performance metrics including execution time, rows examined, and index utilization.

3. Core Issues Identified



Performance Instability

Execution times are highly inconsistent. While the average execution time is **5ms**, certain critical queries spike to **26ms**, creating unpredictable bottlenecks during peak usage.



Resource Waste (Efficiency Gap)

We identified a major "Efficiency Gap." Some queries are forced to examine thousands of rows just to return a single result. This results in an **Efficiency Ratio of < 1%**, causing massive and unnecessary CPU and I/O load.



Missing Indexes

A significant portion of the slowest queries do not utilize database indexes. This forces the database engine to perform a **"Full Table Scan,"** which is the most resource-intensive way to retrieve data.



Complexity Correlation

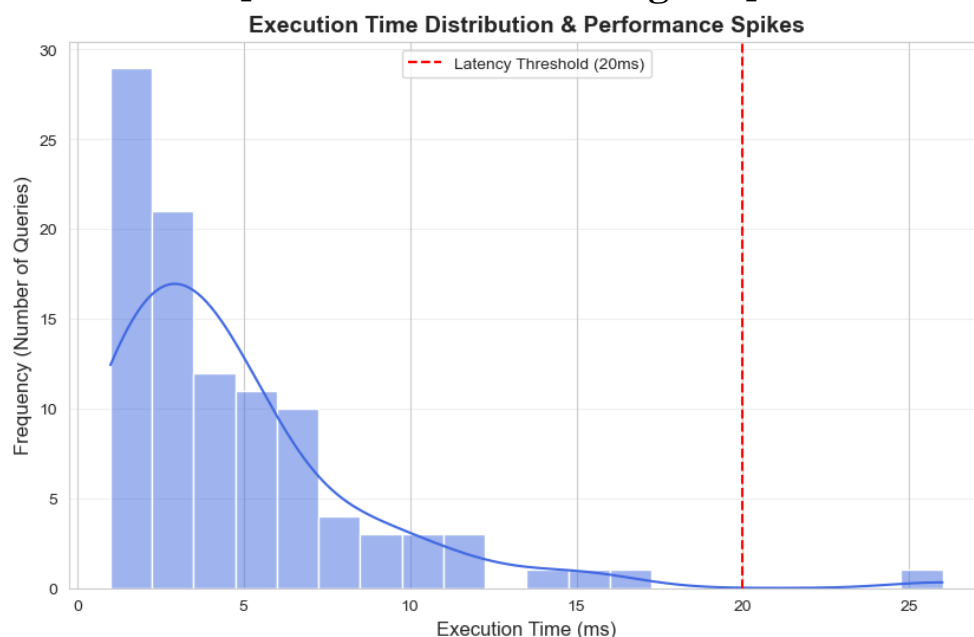
The data confirms a direct mathematical correlation between the **"Complexity Score"** and **"Execution Time."** High complexity—driven by unoptimized joins and logic—is the primary driver of system latency.



Visual Evidence & Data Proof

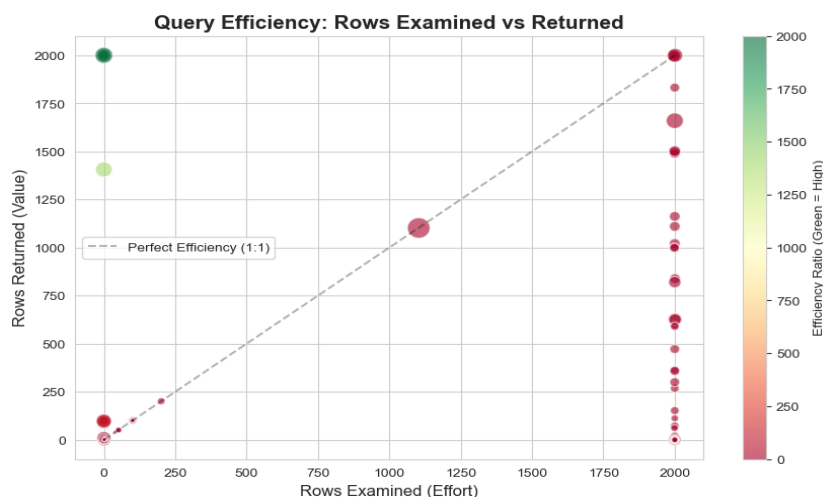


[Execution Time Histogram]



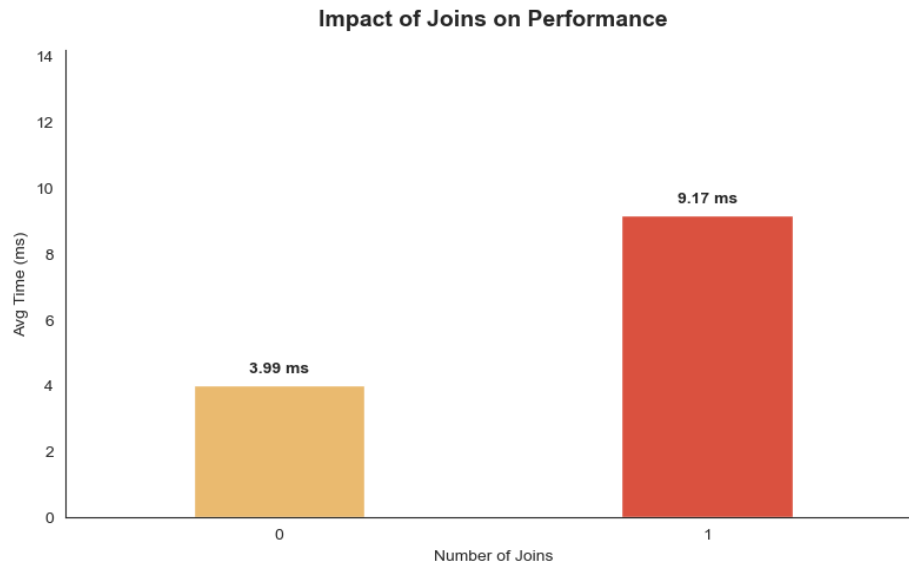
Description: This visual shows the frequency of slow vs. fast queries and highlights the dangerous spikes.

[Efficiency Scatter Plot]



Description: This chart reveals the "Red Zone" queries that exert high effort (Rows Examined) for low value (Rows Returned).

[Joins vs. Time Bar Chart]



Description: Clearly demonstrates how query time climbs as the number of tables joins increases.