Project Proposal

The chosen database system that will be used is PostgreSQL. PostgreSQL is a widely-used open-source RDBMS known for its extreme extensibility. It offers features like Multi-Version Concurrency Control (MVCC) for high performance under load, meaning that it can both read and write simultaneously without locking tables. ANALYZE", enabling direct observation into how queries are planned and executed. PostgreSQL also supports both traditional relational workloads and semi-structured data, which allows more meaningful application design aligned with internals. This platform is well-suited for this project because of its sophisticated cost-based query optimizer and comprehensive indexing support. It supports multiple index types (B-tree, GIN, GiST, BRIN) and provides transparent query planning decisions, which allows internal mechanisms to be observed and analyzed. Additionally, PostgreSQL provides transparency in its execution plans through tools such as "EXPLAIN" and "EXPLAIN ANALYZE", enabling direct observation into how queries are planned and executed. PostgreSQL also supports both traditional relational workloads and semi-structured data, which allows more meaningful application design aligned with internals. The combination of flexible indexing mechanisms and observable query planning decisions makes PostgreSQL an ideal database system for analyzing indexing behavior and execution strategies in modern relational databases.

The project will focus on the following two internal components of PostgreSQL:

1) Indexing - The project will analyze how PostgreSQL implements and utilizes different index types, primarily B-tree indexes. B-tree indexes are the default index structure in PostgreSQL and are widely used for equality lookups, range queries, and sorting operations. Indexing directly affects query performance and is observable through execution plans and performance measurements.
2) Query Planning and Execution - PostgreSQL uses a cost-based optimizer to analyze SQL statements and provide the most efficient execution to retrieve the data. The project will investigate how PostgreSQL estimates query cost, how it decides between sequential and index scans, and how factors such as selectivity, data distribution, and indexing strategies influence the plan selection.

Using the “EXPLAIN” and “EXPLAIN ANALYZE” commands in PostgreSQL, we will examine different execution plans to understand how optimizer decisions affect performance.

Preliminary Application Idea:

The application will be a simple Book Search and Filtering System. The application will be able to store book records containing fields such as title, author, publication year, and rating. Users will be able to filter books by author, perform range queries on publication year, and sort results by rating. Indexes will be created on selected columns to evaluate their effect on query performance. The project will then compare execution plans and timing results before and after index creation to show how indexing influences PostgreSQL’s planner decision.

Team Information and Responsibilities:

Group Members:
William Khuu 3398086799
Khang Thai 5721113147

Khang Thai - Indexing Focus

- Analyze B-tree index structures and behavior
- Design and implement indexing strategies for the application
- Conduct experiments comparing sequential scans vs index scans
- Measure the performance of indexes using the built-in PostgreSQL functions

William Khuu - Query Planning and Execution Focus

- Analyze PostgreSQL’s cost-based optimization
- Research how PostgreSQL estimates query costs
- Investigate planner decisions between sequential vs index scans
- Analyze execution plans under different data distributions

Shared Responsibilities

- Application schema design
- Dataset prep
- Final report writing
- Demo prep

References

- https://www.postgresql.org/docs/current/indexes.html
- https://www.cybertec-postgresql.com/en/how-the-postgresql-query-optimizer-works/