University of the West of Scotland

School of Computing, Engineering and Physical Sciences

MSc Masters Project Specification

Student name: **Prajwalaradhya Shivakumaraswamy Kesaramadu**

Banner ID: **B01759301**

Email: **B01759301@stuentmail.uws.ac.uk**

Project being undertaken on part-time or full-time basis: **Full-time**

MSc Programme (specify the specialist pathway, if any): **None (Plain MSc IT)**

MSc Programme Leader: **Volkan Tunali**

Project Title:

|  |
| --- |
| Adaptive Database Indexing: Switching Between B-Tree and LSM Tree based on workload |

Research Question to be answered:

|  |
| --- |
| What are the performance benefits for using different storage engine based on workload on the table. |

Overview, Justification and overall aim of project

|  |
| --- |
| Modern relational databases use indexing structures like B+ trees and LSM trees, each suited to different workload patterns. However, most systems statically apply a single indexing approach, which limits performance under mixed workloads. This project proposes a solution that allows users to specify different indexing engines (B+ tree or LSM tree) per column while creating a table. Multiple indexes with different structures can coexist, giving flexibility based on expected usage. The system will enable workload-aware choices at design time by the user. This addresses a key gap in relational database indexing by introducing the adaptability without runtime complexity. The aim is to enhance query performance and storage efficiency through a user-guided, hybrid indexing model. |

Objectives

*(List of tasks to be undertaken to achieve overall aim of the project and to answer the research question posed)*

|  |
| --- |
| 1. Review existing literature on indexing strategies in relational databases, focusing on B+ Trees and LSM Trees. 2. Understand the internal workings and performance characteristics of B+ Tree and LSM Tree indexes. 3. Study tools and systems that implement these index types in practice (e.g., MySQL, PostgreSQL, Rocks DB, Cockroach DB). 4. Design a sample database and load hybrid workloads simulating varied query patterns. 5. Benchmark performance of different index types applied per column under mixed workloads. 6. Evaluate results and conclude on the effectiveness of user-defined hybrid indexing in relational systems. |

Methodology

*(Explanation and justification of approach and methodology proposed. This section should explain the nature of the data and method of data collection. Identification and brief discussion of any analytical tools to be used; Identification of any ethical and practical issues.)*

|  |
| --- |
| This project adopts an experimental research approach, aimed at exploring how hybrid indexing—using both B+ Tree and LSM Tree structures—performs under mixed workloads typical of modern applications. The study is based on findings from existing academic literature and practical benchmarking of real-world database systems.  Synthetic data will be generated using tools like Faker.js/Faker for Python, and the same dataset will be loaded into multiple database systems to ensure consistency. The databases used include PostgreSQL (B+ Tree indexing), CockroachDB (LSM Tree-based), and MySQL with MyRocks storage engine (for hybrid indexing support). Indexing strategies will be manually defined based on the perceived importance of each column and how applications are expected to query them. This user-defined indexing approach is intended to give developers the flexibility to choose indexing methods best suited for their application workloads.  The experimental setup includes local testing using Docker and cloud-based benchmarking on AWS EC2 virtual machines. Tools such as HammerDB and custom Python/SQL scripts will be used to execute various types of queries, including range queries, indexed lookups, aggregations, and subqueries.  Benchmarking results will be evaluated based on key performance metrics such as CPU usage, memory consumption, disk I/O, latency, and network throughput. These results will help assess the performance trade-offs and practical benefits of adopting a hybrid indexing strategy in SQL-based relational databases.  There are no ethical concerns in this study, as all data is artificially generated and no personal or sensitive information is used. |

Work Plan

*(A timetable for completion of research. MSc Masters Project full-time is allowed 15 weeks and part-time is 30 weeks. You may use any appropriate method to present the well-thought-through plan of action reflecting your project activities)*

|  |
| --- |
| The project will be completed over the remaining 11 weeks of the MSc schedule. The timeline is structured to ensure a logical flow from research and exploration to implementation, benchmarking, and final evaluation:   * Weeks 1–4: Conduct in-depth research on storage engines and indexing strategies, focusing on the theory behind B+ Trees and LSM Trees. Study existing academic work and documentation to understand how these structures are implemented in modern relational databases. * Weeks 5–7: Deep dive into practical tools—PostgreSQL, CockroachDB, and MySQL with MyRocks—to understand their indexing implementations. Set up the environments using Docker and AWS, prepare datasets using Faker, and define indexing strategies for each system. * Weeks 8–10: Execute benchmarking tests on local and cloud platforms. Run various queries under mixed workloads using tools like HammerDB and custom scripts. Collect performance metrics including CPU, memory, disk I/O, and query latency. * Week 11: Analyse and evaluate the results. Identify performance trends, trade-offs, and insights. Draft conclusions on the effectiveness of user-defined hybrid indexing strategies and prepare final documentation for submission. |

Relationship of proposed project to MSc programme/stream

|  |
| --- |
| This project aligns closely with the MSc Information Technology programme, particularly the Database Design and Implementation module. It focuses on the internal architecture and behaviour of relational databases, with specific emphasis on how indexing strategies like B+ Trees and LSM Trees impact performance. By exploring storage engines, indexing mechanisms, and benchmarking under real-world workloads, the project deepens understanding of database internals—an essential aspect of advanced database design. |

Indicative reading list (references to be correctly presented) and resources (hardware, software, etc.)

|  |
| --- |
| Comer, D. (1979). Ubiquitous B-Tree. ACM Computing Surveys, 11(2). doi:https://doi.org/10.1145/356770.356776.  Dong, S., Kryczka, A., Jin, Y. and Stumm, M. (2021). RocksDB: Evolution of Development Priorities in a Key-value Store Serving Large-scale Applications. ACM Transactions on Storage, 17(4), pp.1–32. doi:https://doi.org/10.1145/3483840.  Evans, G. (2025). Understanding the mechanics of PostgreSQL B-Tree indexes. [online] Fastware.com. Available at: https://www.postgresql.fastware.com/pzone/2025-01-understanding-the-mechanics-of-postgresql-b-tree-indexes [Accessed 5 Jun. 2025].  Kleppmann, M. (2017). Designing Data-Intensive Applications. ‘O’Reilly Media, Inc.’  Lu, L., Pillai, T.S., Arpaci-Dusseau, A.C. and Arpaci-Dusseau, R.H. (2016). WiscKey: separating keys from values in SSD-conscious storage. File and Storage Technologies. [online] Available at: https://www.usenix.org/system/files/conference/fast16/fast16-papers-lu.pdf [Accessed 5 Jun. 2025].  Mattis, P. (2020). Introducing Pebble: A RocksDB-inspired key-value store written in Go. [online] Cockroachlabs.com. Available at: https://www.cockroachlabs.com/blog/pebble-rocksdb-kv-store/ [Accessed 5 Jun. 2025].  O’Neil, P., Cheng, E., Gawlick, D. and O’Neil, E. (1996). The log-structured merge-tree (LSM-tree). Acta Informatica, 33(4), pp.351–385. doi:https://doi.org/10.1007/s002360050048.  **Resources List:**  **Hardware:**   * **Computer** with sufficient cpu & ram to run docker * **AWS EC2** machines   **Software:**   * **Docker** for local testing * **MySQL, PostgreSQL, CockroachDB** are the databases used for testing * **Faker** for generating fake data * **HammerDB** for benchmarking database * **Python/NodeJs** for scripting and testing * **AWS Cloud** for testing in Cloud |

Marking scheme

|  |
| --- |
| 1. Introduction – 10% 2. Literature Review – 10% 3. Methodology – 10% 4. Implementation – 25% 5. Experimentation and Results – 25% 6. Analysis and Discussion – 10% 7. Conclusion and Future Work – 10% |

Supervisor

|  |
| --- |
|  |

Moderator

|  |
| --- |
|  |

Programme Leader

|  |
| --- |
| Volkan Tunali |

Date specification submitted

|  |
| --- |
| 6-6-2025 |

Please complete the ‘ethics’ & pathway confirmation form below for all projects.

**School of Computing, Engineering and Physical Sciences**

**MSc Masters PROJECT – REQUIREMENT FOR ETHICAL APPROVAL & PATHWAY CONFIRMATION**

**SECTION 1: TO BE COMPLETED BY THE STUDENT**

Does your proposed research involve: research with human subjects (including requirements gathering and product/software testing), access to company documents/records, questionnaires, surveys, focus groups and/or other interview techniques? Does your research entail any process which requires ethical approval? (please enter √ in the appropriate box)

|  |  |  |
| --- | --- | --- |
| YES |  | **You must submit an application for approval to the Ethics Review Manager** |
| NO | ✔️ | You do not need to submit an application to the Ethics Review Manager |

I confirm that the above project specification aligns with my MSc programme specialist pathway. (please enter √ in the box)

✔️

**Name of Student (Print name): Prajwalaradhya Shivakumaraswamy Kesaramadu**

**Signature: Prajwal Aradhya**

**Date: 6-6-2025**

**SECTION 2: TO BE COMPLETED BY THE PROJECT SUPERVISOR**

I understand that the above project requires/does not require\* ethical approval (\*please delete as appropriate).

I confirm that the above project aligns with the MSc programme specialist pathway the

student is enrolled in. (please enter √ in the box)

**Supervisor (print name):**

**Signature**:

**Date:**

**IMPORTANT: please note that by signing this form all signatories are confirming that any potential ethical issues have been considered and, where necessary, an application for ethical approval has been/will be made via the Ethical Review Manager software.**

**Any project requiring ethical approval but which has not been given approval will not be accepted for marking.**

**Ethical approval cannot be sought in retrospect.**