This abstract describes research on the integration of iterative Common Table Expressions (CTEs) into a production relational database management system. The authors claim that their approach, based on a functional rewrite of iterative CTEs, can be efficiently incorporated into the system without major intrusion and can improve performance by an order of magnitude in some cases. The implementation is based on the MPPDB, a shared-nothing relational parallel database engine, and has been verified through experimental evaluation using real-world datasets and queries.

The abstract provides a clear overview of the research and its findings, but it could benefit from a clearer explanation of the motivation behind the research and the problem being addressed. Additionally, the abstract could provide more context on the limitations of the current RDBMSs and how iterative CTEs address these limitations. Additionally, the abstract could include more details on the specific datasets and queries used in the experimental evaluation, as well as a more comprehensive discussion of the results.

The introduction to a research paper proposes a solution to extend the capabilities of Relational Database Management Systems (RDBMSs) to handle iterative computations. The authors argue that the Structured Query Language (SQL) used by RDBMSs is limited in expressing iterative computations, which are essential for certain types of queries. As a result, many users switch from RDBMSs to specialized graph processing engines or Datalog systems that can handle these types of queries more effectively. However, the authors believe that there is a need for a solution that enables RDBMSs to run iterative queries without transferring data to a different engine, as this can be time-consuming or infeasible.

The authors point out that recent efforts have explored the possibility of extending recursive Common Table Expressions (CTEs) in SQL to accommodate iterative queries, but these solutions are implemented outside the system and have limitations such as difficulty in maintaining ACID properties and lack of system-specific optimizations. The authors propose a solution that addresses these limitations by extending the planner and optimizer of the RDBMS to support iterative CTEs natively, while retaining existing structures and resources of the system.

This section describes the background of a SQL extension that proposes the use of iterative CTEs to address the limitations of traditional recursive views and CTEs. The authors use the PageRank (PR) query as an example to demonstrate the need for iterative CTEs and to illustrate how they work. They explain how PR can be implemented using a custom-made SQL-based application, which requires multiple SQL statements and is prone to errors. The authors then describe how iterative CTEs are defined and executed in the SQL extension. The query is executed by a middleware system, which eliminates the need for custom SQL scripts but comes with limitations, such as the complexity of query processing and difficulties in handling failures and aborts.

The authors provide a clear and concise overview of the background and limitations of the existing approach to SQL processing, which motivates the need for their proposed SQL extension. The explanation of the iterative CTEs is straightforward and easy to understand. The authors could have further improved their critique by explaining the benefits of their proposed solution over the existing approach and discussing the limitations of the proposed solution.

The writing in this section describes the implementation of iterative CTEs in Futurewei's MPPDB database management system. The authors explain that their implementation constructs a single query plan for SQL queries involving one or more iterative CTEs, similar to how regular or recursive CTE queries are processed. The authors highlight the advantages of their approach, such as ensuring ACID properties, allowing the workload manager to schedule iterative CTEs, avoiding unnecessary overhead, and applying existing query optimizations and cost estimations.

The authors describe the modifications made to various components of the MPPDB system, such as the parser, rewrite subsystem, planner, and execution engine, to enable the processing of iterative CTEs. The changes are summarized and the details are described in other sections of the paper.

Overall, the description of the implementation is clear and concise, and the authors provide sufficient detail on the modifications made to the MPPDB system to support iterative CTEs. However, the authors could have given more details on the "new simple operator" mentioned in the introduction, as this is an important part of their implementation.

The description of the core algorithm in MPPDB is comprehensive, but it could benefit from some clarifications and improvements in terms of presentation. Here are a few suggestions:

* Consider providing more context or background information on MPPDB and what it is designed to do before diving into the technical details of the algorithm.
* It would be helpful to have a clear definition of terms used in the description, such as "CTE," "parse tree," "SQL operator," and "functional rewrite," for the reader to fully understand the algorithm.
* The description mentions "the generic algorithm used inside the planner" and then refers to Algorithm 1 as an example, but it is not clear how Algorithm 1 is related to the generic algorithm. Consider providing a clearer explanation or connecting the two concepts better.
* The figures and tables used in the description could benefit from a clearer labeling or explanation. For example, it is not clear what "Table I" refers to, and the relationship between Figures 3 and 4 and the algorithm could be clearer.
* The description of the loop operator and how it handles the termination condition could benefit from more detail and explanation. Consider explaining how the loop operator is used in the functional rewrite and what information is passed to it from the parse tree.
* The description mentions the simple case of the PR query, but it is not clear what makes it simple compared to other cases. Consider clarifying this point or providing an example of a more complex case to illustrate the full capabilities of MPPDB.
* Finally, consider adding a conclusion or summarizing the algorithm in a clear and concise manner to make it easier for the reader to understand the main takeaways from the description.

This is a technical text that describes the optimization techniques applied to iterative Common Table Expressions (CTEs) in a relational database management system (RDBMS). The text explains that the optimizer treats iterative CTEs like any other regular SQL statement and applies various optimization rewrites, including join elimination, outer to inner join conversions, cost-based rewrites, join re-ordering, and aggregate planning. The text also mentions two special rule-based optimizations for iterative CTEs, the Common Result Optimization and the Predicate Push Down.

The Common Result Optimization is a heuristic rewrite that identifies and materializes constant parts of a query to avoid recomputing the same results multiple times. The text explains that this optimization is crucial for iterative CTEs and has been included in the RDBMS as a heuristic rewrite.

The Predicate Push Down is a rewrite rule that pushes predicates within or across query blocks. The text explains that this rewrite is widely covered by the RDBMS, but some restrictions need to be considered when applying it to iterative CTEs.

In general, the text provides a clear explanation of the optimization techniques applied to iterative CTEs in the RDBMS, but it assumes some prior knowledge of RDBMS concepts and optimization techniques. It would be helpful to have more context and explanation for readers who are not familiar with these topics.

This paper describes a proposal for optimizing iterative Common Table Expressions (CTEs) in a database management system. The paper outlines the implementation of two new operators, rename and loop, to be added to the execution engine of the system. The rename operator is used to rename references to temporary results, and the loop operator is used to handle conditional execution flow in iterative CTEs. The authors describe the implementation of these operators and provide results from experiments that evaluate the effectiveness of the proposed optimizations. The experiments show that minimizing data movement in each iteration can improve performance by 48%, materializing parts that remain constant in all iterations can improve performance by 21%, and pushing down predicates to avoid unnecessary data processing can improve performance by at least an order of magnitude. The authors also compare the performance of optimized iterative CTEs with stored procedures and observe that CTEs can be executed at least 25% faster. The paper provides detailed information on the system setup and methodology used in the experiments.

The introduction of this section is well written and provides a clear overview of the topic being discussed, which is the related work in the field of implementing iterative SQL queries in RDBMSs. The author effectively summarizes the different approaches taken by previous studies, including specialized graph systems, SQL systems with graph extensions, and recursive SQL queries.

The author then goes on to discuss the limitations of each of these approaches and provides a brief overview of the work done in each of these areas. The discussion of the limitations of each approach is thorough and provides a good basis for the author's proposed solution.

The conclusion of this section is also well written, providing a summary of the author's proposed solution and the optimizations that have been implemented. The author effectively highlights the importance of a native solution and the benefits of implementing the proposed solution in a commercial RDBMS.

Overall, the related work section is well written and provides a comprehensive overview of the topic. The author has effectively summarized the previous studies in this field and provided a clear rationale for their proposed solution.