This 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. 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. The authors use the PageRank (PR) query as an example to demonstrate the need for iterative CTEs and to illustrate how they work. They then describe 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 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 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 description of the core algorithm in MPPDB is comprehensive, but it could benefit from some clarifications and improvements in terms of presentation. The authors could have given more details on the "new simple operator" mentioned in the introduction and 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 figures and tables used in the description could benefit from a clearer labeling or explanation. The description of the loop operator and how it handles the termination condition could benefit from more detail and explanation.

Overall, the authors provide a clear and concise overview of the research and its findings, but the abstract 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. 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.