Is Quantum-Based SQL Query Execution Viable?

Manish Kesarwani
IBM Research & Indian Institute of Science
Bangalore, India
manishkesarwani@in.ibm.com

Jayant R. Haritsa
Indian Institute of Science
Bangalore, India
haritsa@iisc.ac.in

ABSTRACT

Recently, there have been significant scaleups in quantum computing capacity, with 1000+ qubit machines already in deployment, and development roadmaps promising 100, 000+ qubits by 2033. To leverage this prospective power, we need to investigate concurrently the hosting of relational database engines on these platforms. On the positive side, there has been a promising exploration of quantum computing for various *optimization-based* components within the engine, including join-order and index-configuration choices. However, hosting of SQL query *execution* on quantum platforms is still in a nascent research stage. In this paper, we outline various challenges that are likely to arise in this endeavour, which cover the gamut from data loading to probabilistic results. We also discuss potential mechanisms for addressing some of these hurdles.

VLDB Workshop Reference Format:
Manish Kesarwani and Jayant R. Haritsa. Is Quantum-Based SQL Query Execution Viable?. VLDB 2024 Workshop: The Second International Workshop on Quantum Data Science and Management (QDSM'24).

1 INTRODUCTION

In recent years, the database research community has shown considerable interest in hosting relational database management systems (RDBMS) on quantum platforms. Several vision papers have highlighted the need to accelerate database tasks using quantum computing [4, 12, 33, 36, 39]. There have also been successful demonstrations that quantum computing can significantly improve the efficiency and performance of specific components of RDBMS engines. These efforts have predominantly focused on *optimization* problems, such as multi-query planning [8, 28, 32], join order selection [24, 29, 35], transaction schedules [1, 2, 10], index configuration choices [20], and estimation of cardinality, cost, and execution times for SQL queries [34]. However, there has been comparatively much less effort wrt hosting SQL query *execution* on quantum platforms. We are aware of only a few limited efforts [6, 13, 18, 19, 38] in this domain. Therefore, it remains unclear whether the end-to-end processing of industrial-strength database queries – for instance, the SQL query depicted in Figure 1 from the popular TPCH benchmark [31] – is viable on quantum platforms. More importantly, we ask whether such an approach, even if viable, would offer any substantive computational advantages.

Proceedings of the VLDB Endowment. ISSN 2150-8097.

At first glance, SQL queries appear a prime candidate to take advantage of the computational power of quantum computing for improved data management and analytics. For instance, Grover Search (GS) algorithm [11] could be used to speed up complex selection operations on non-indexed columns by reducing the time complexity from $O(n)$ to $O(\sqrt{n})$. Quantum computing's ability to process multiple possibilities simultaneously can reduce the time complexity of finding matching tuples across tables joined by arbitrary predicates. Union, Intersection, and Set Difference operators could also be benefited using quantum computing – an initial algorithm for this task was demonstrated in [18]. The ideas proposed in [22] could be used to improve the efficiency of the Sort operation in space-bounded settings. Quantum Fourier Transform (QFT) [25] could offer more effective handling of temporal and spatial data queries for applications in climate modeling and geographic information systems (GIS). Moreover, quantum networking via entanglement could enable instantaneous data synchronization across distributed databases, improving global data consistency and availability. As quantum technology progresses, the development of a quantum-enhanced SQL query execution engine could utilize these capabilities and realize the vision of a multi-modal quantum database system enunciated in [39].

In this paper, we begin by examining the impact of quantum computing on the semantics and contractual obligations of current SQL query execution engines. Subsequently, we investigate the research challenges and potential benefits associated with the design and implementation of a quantum-based system for SQL query processing. This involves an investigation of the intrinsic complexities and potential hurdles, such as the fundamentally probabilistic nature of quantum computation and the conversion of classical data structures and existing data into quantum-compatible formats. Next, we propose an updated SQL query processing contract to bridge the gap between RDBMS expectations and quantum computing capabilities. Finally, we present an overview of a Quantum-Classical hybrid SQL query execution engine, which attempts to implement this updated contract.

```sql
SELECT L_ORDERKEY, O_ORDERDATE, O_SHIPPRIORITY,
SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)) AS REVENUE,
FROM CUSTOMER, ORDERS, LINEITEM
WHERE C_MKTSEGMENT = 'BUILDING' AND C_CUSTKEY = O_CUSTKEY
AND L_ORDERKEY = O_ORDERKEY AND O_ORDERDATE < DATE '1996-04-20'
AND L_SHIPDATE > DATE '1996-04-20'
GROUP BY L_ORDERKEY, O_ORDERDATE, O_SHIPPRIORITY
ORDER BY REVENUE DESC, O_ORDERDATE
FETCH FIRST 10 ROWS only;
```

Figure 1: TPCH Q3

The rest of the paper is organized as follows: A brief introduction to quantum computing is given in Section 2. The impact of quantum computing on ACID semantics is discussed in Section 3. This is followed by a discussion on the research challenges and potential benefits of hosting SQL engines on quantum platforms in Section 4. Then, in Section 5, we propose a Quantum-Classical Hybrid SQL Query Contract and present our vision for the realization of such a system in Section 6. Related work is reviewed in Section 7, and finally our conclusions are highlighted in Section 8.

2 QUANTUM BACKGROUND

Quantum computation brings to bear the fundamental phenomena of quantum mechanics, such as *superposition*, *interference*, *entanglement*, *tunnelling*, and *irreversible measurement* for information processing. A comprehensive review of quantum computation is provided in [25]. In this section, we provide an overview of the basic building blocks of quantum computation, with a particular focus on hosting SQL query processing.

2.1 Quantum Computing Paradigms

Quantum computing platforms can be broadly categorized into two main paradigms: *quantum annealing* and *quantum circuits*. They represent, as explained below, distinct methods for harnessing the principles of quantum mechanics to perform computations.

Quantum annealing, exemplified by commercially deployed systems such as D-Wave [7], is particularly suited for optimization problems. It leverages quantum tunneling to escape local minima, potentially finding better solutions more efficiently than classical algorithms. However, quantum annealing is limited in its versatility as it is primarily designed for specific types of optimization tasks and does not naturally extend to other kinds of computational problems.

In contrast, quantum circuit processors, implemented by platforms such as IBM's Qiskit [16], provide a more flexible framework for quantum computation. In this approach, quantum information is processed through a series of quantum gates arranged in circuits, analogous to classical circuits but with the ability to handle superposition and entanglement. Quantum circuits can implement a wide variety of algorithms beyond just optimization. Also, in a practical RDBMS setting, it appears reasonable to expect a single quantum platform that is usable for both generic computation as well as optimization – a circuit processor provides this computational flexibility. Therefore, hosting SQL query execution on quantum circuit processors is of independent interest.

2.2 Grover Search (GS)

Grover Search (GS) [11] is a quantum algorithm for solving unstructured search problems with high probability. Specifically, given an unordered list of N items, GS identifies a desired item with high probability, using just $O(\sqrt{N})$ operations, compared to the $O(N)$ operations of classical algorithms. Consequently, GS is a highly promising candidate for supporting SQL query execution on quantum platforms.

The key component of the GS algorithm is a problem-specific *quantum oracle*, designed to recognize the desired item from the input list and signal its qualification using the quantum phase kickback phenomenon, where the phase of the target qubit is transferred to the control qubit [25]. This quantum oracle is used in conjunction with the *diffusion operator* [11], together forming the Grover Iterate. Each application of the Grover Iterate incrementally increases the probability of identifying the target item by effectively rotating the state vector towards the desired solution. To achieve a success probability of $\geq 50\%$, the optimal number of repetitions of the Grover Iterate needs to be precisely calculated, as too many repetitions can result in over-rotation of the state vector, thereby reducing the success probability instead of increasing it. However, this calculation depends on the number of qualifying items, which is usually unknown and hence requires specialized strategies to solve this estimation issue [3].

Therefore, the key challenges in implementing the GS algorithm are as follows: 1) Develop an efficient, problem-specific quantum oracle; 2) Determine the optimal number of Grover iterations; 3) Enhance the success probability; 4) Preserve the quadratic speedup provided by the GS algorithm; and 5) Utilize only standard quantum gates to ensure feasibility on actual quantum hardware.

3 QUANTUM COMPUTATION AND ACID SEMANTICS

The current RDBMS contract is underpinned by the ACID properties of Atomicity, Consistency, Isolation, and Durability. By adhering to these properties, RDBMS ensures that SQL queries are processed reliably and consistently, producing deterministic results. Such determinism is crucial for RDBMS applications, as even minor inconsistencies can lead to significant correctness issues. In this section, we evaluate each property and examine the advantages and limitations of quantum computing within this context.

Atomicity: Quantum computing has the capability to execute complex operations rapidly, thus reducing the likelihood of partial transactions and enhancing atomicity. However, quantum operations are inherently error-prone and necessitate specialized error correction mechanisms. These mechanisms can introduce additional computational overhead, which in turn increases the number of operations required for a transaction, thereby increasing the duration and challenging the atomicity. Nonetheless, the field is continually evolving. More robust and efficient error correction methods are being developed [21], along with improvements in the stability of qubits and the fidelity of measurements [15] to address these challenges.

Consistency: Quantum computing, leveraging superposition and entanglement, can efficiently perform intricate consistency checks to ensure data integrity. However, quantum states' fragility due to decoherence can introduce inconsistencies as information degrades over time. Additionally, the probabilistic nature of quantum computing can amplify these issues, potentially leading to uncertainties in transaction outcomes. To mitigate these challenges, mechanisms like the powering lemma [17] could be employed to improve the success probability of quantum algorithms, thereby enhancing their reliability and consistency.

Isolation: Quantum parallelism allows multiple transactions to be processed simultaneously, potentially enhancing isolation. However, quantum entanglement can inadvertently link transactions, complicating true isolation since operations on one qubit can affect its entangled partners. Particularly, in the implementation of quantum algorithms for SQL execution, ancilla qubits may be employed to load data in superposition and entangle data qubits corresponding to different columns of the same table. Proper management of these ancilla qubits is crucial to prevent side effects from residual entanglement that could interfere with isolation. Fortunately, since all quantum operations (except measurement) are reversible, algorithms can reset the states of ancilla qubits, mitigating unwanted effects of residual entanglement, and preserving isolation.

Durability: Potential future advances in quantum memory systems [14] could provide robust data storage, maintaining durability even under significant computational load and system failures. However, the inability to perfectly clone quantum states [37] makes it difficult to create the redundant copies necessary to ensure data consistency and recovery.

4 QUANTUM CHALLENGES AND BENEFITS FOR SQL QUERY EXECUTION

In this section, we first explore a broad spectrum of technical challenges that must be addressed to successfully execute SQL queries on quantum platforms. Following this, we delve into the expected benefits of this endeavor. Figure 2 visually summarizes both the challenges and the advantages of quantum processing.

4.1 Quantum Challenges

4.1.1 Probabilistic Results. Enterprise database applications usually expect deterministic production of correct results, but quantum computers inherently work in a probabilistic space. This means that the quantum results can include:

(1) **False Positives (Incorrect Results):** The quantum algorithm might identify a solution as correct when it is actually incorrect. This occurs due to the inherent uncertainty and probabilistic nature of quantum states, where the system can collapse to an incorrect state.
(2) **False Duplicates (Repeat Productions of a Correct Result):** The algorithm might produce the correct result multiple times. While this may seem beneficial, it can lead to inconsistencies downstream, particularly when unique solutions or aggregates need to be computed. This redundancy, in addition to probabilistic computation, stems from quantum systems being highly susceptible to noise and decoherence. These factors can cause qubits to lose their quantum states and introduce errors into the computation, resulting in false duplicates in the results.
(3) **False Negatives (Missing Results):** The quantum algorithm might fail to identify some of the correct solutions. This could happen, for instance, in the GS algorithm, where, the final measurement operation randomly picks one of the qualifying solutions with equal probability. With the possibility of picking the same result multiple times and missing some of the valid solutions.

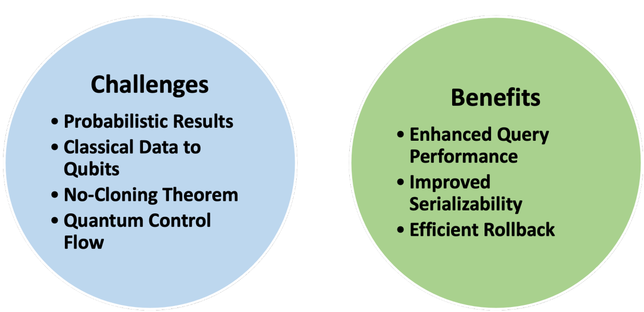

Figure 2: Research Challenges and Potential Benefits

4.1.2 Classical Data to Qubits. The issue of quantum data loading may appear to be an already solved problem given the extensive research in the quantum optimization domain. However, each proposal uniquely transforms the input data. For example, in [32], query workload instances are translated into weights on and between qubits so that the configuration minimizing the input formula can be found via adiabatic quantum annealing. In contrast, in [29], the join ordering problem undergoes multiple transformations: First, it is expressed as a mixed integer linear programming (MILP) problem, then adjusted to a binary integer linear programming (BILP) model, and finally transformed into a Quadratic Unconstrained Binary Optimization (QUBO) format for quantum processing. As a third example, in [20], the index selection problem instance is loaded into the relative phase of a quantum superposition state. These instances demonstrate that each application employs a customized solution.

Similarly, for SQL query processing, the classical data need to be explicitly loaded in the qubits. Current data loading techniques [30], including basis encoding, amplitude encoding, and angle encoding, must be re-evaluated to assess their effectiveness for the new computations. Should these techniques fall short, it will be necessary to explore new quantum data structures and loading mechanisms.

4.1.3 No-Cloning Theorem. The no-cloning theorem [37] is a fundamental principle in quantum computing. This theorem states that it is impossible to create an identical copy of an arbitrary unknown quantum state. This constraint presents a considerable challenge for SQL query processing because it restricts our ability to duplicate an intermediate quantum state for reuse or to dynamically generate multiple independent copies of a database for parallel processing.

4.1.4 Quantum Control Flow. Another challenge is the inability to directly observe a qubit's state without collapsing it. This collapse due to measurement is a fundamental aspect of quantum mechanics and could present obstacles for implementing quantum conditional statements, which may be needed for certain SQL query constructs.

In addition, quantum computing does not directly support iterative control structures. For example, in the GS algorithm, repeating the Grover iterate I times necessitates appending the corresponding quantum circuit for the iterate operation I times in the final quantum circuit. This approach results in large and complex quantum circuits with increased depth. Implementing such circuits requires qubits to maintain their quantum states for extended durations. However, qubits are extremely sensitive to environmental factors, and maintaining coherence long enough to perform meaningful calculations remains a significant technical challenge.

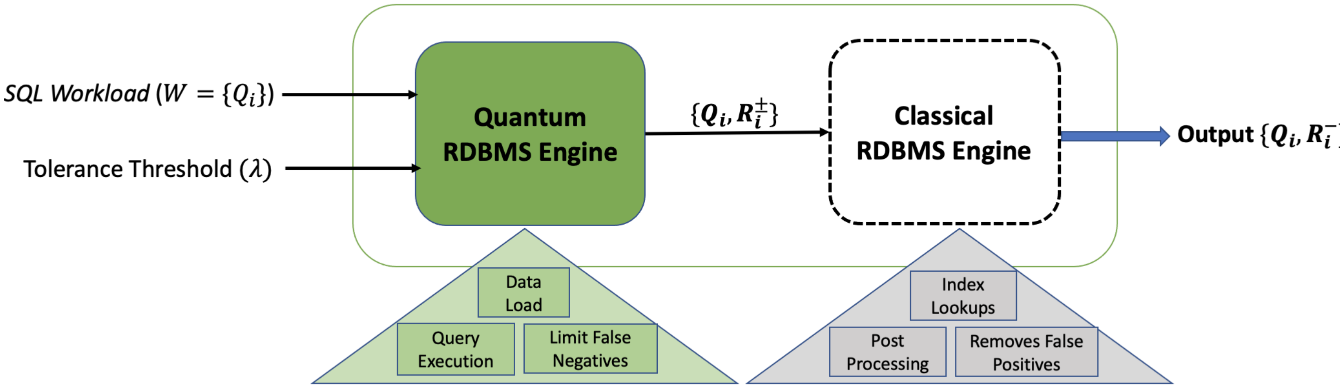

Figure 3: Proposed Quantum-Classical Hybrid SQL Query Contract

4.2 Quantum Benefits

4.2.1 Enhanced Query Performance. Quantum computers using the GS algorithm could potentially offer quadratic speedups for complex SQL queries involving joins and multiple complex filtering conditions, which are typically computationally intensive on classical hardware in the absence of appropriate index structures. Furthermore, quantum parallelism could support the simultaneous evaluation of multiple query paths and thereby enable parallel execution of subqueries within an SQL statement, potentially leading to significant performance gains.

4.2.2 Improved Transaction Concurrency. The fast execution of SQL queries could lead to almost-instantaneous commits of transactions, reducing the number of parallel conflicting transactions and thereby facilitating enhanced concurrency. This can significantly reduce the overheads associated with specialized algorithms traditionally used to ensure serializability. Consequently, transaction management systems could become simpler and more efficient.

4.2.3 Efficient Rollback. The inherently reversible nature of quantum operators could facilitate efficient rollback of transactions. This reversibility allows the system to efficiently retrace its steps if a transaction encounters an error or needs to be undone, without the significant overheads associated with traditional data structures. This not only simplifies the rollback process, but also enhances the overall reliability and robustness of the transaction management component.

Overall, it is our expectation that the benefits of executing SQL queries on quantum platforms are likely to justify the efforts needed to address the associated research challenges.

5 PROPOSED QUANTUM-CLASSICAL HYBRID SQL QUERY CONTRACT

The challenges presented in Section 4.1 are complex and do not lend themselves to straightforward solutions, especially in relation to maintaining the current RDBMS contract discussed in Section 3. Therefore, to design a quantum-enhanced RDBMS for SQL query processing and ensure reliable performance and consistent results, we propose updating the existing contract. Firstly, the system should be developed as a hybrid quantum-classical system, rather than exclusively as a quantum RDBMS. This hybrid approach should extend beyond what is currently achieved by Variational Quantum Algorithms [5]. The quantum and classical components should jointly mitigate research challenges – for instance, eliminating false negatives in the quantum domain whereas false positives are removed in the classical landscape. A similar hybrid approach was adopted in a recent study on index tuning [20], where the initial modules of the index tuning pipeline were executed in the classical domain and only the computationally hard problem was outsourced to the quantum engine. Major industry players, such as IBM, are also implementing and advocating for this approach, which they term *quantum-centric supercomputing* [9].

Secondly, given the inherently probabilistic nature of quantum computers, the users should specify a nonzero tolerance threshold ($\lambda \in (0, 1]$) when submitting an SQL workload (or query). This tolerance threshold represents the *expected percentage of false negatives* that the user is willing to accept in the final output. It determines the balance between result quality and computational effort, and should be adjusted carefully according to the application's needs.

The revised quantum-classical hybrid RDBMS contract is illustrated in Figure 3. In this model, the user submits an SQL query workload (W) along with a non-zero tolerance threshold (λ). Each SQL query within the workload W is processed by the Quantum RDBMS Engine, which utilizes a quantum computer for execution. Subsequently, the output from the Quantum RDBMS Engine is passed over to a classical DBMS engine that produces the final output tuples, fulfilling the user's request. *Note that we are not proposing approximate query processing here; rather, we are focusing on identifying qualifying tuples with some tolerance for false negatives.* The remainder of this section provides a more detailed review of these components.

5.1 Quantum RDBMS Engine

Given an SQL query, the Quantum RDBMS Engine first loads the classical data into qubits and constructs an appropriate quantum circuit to represent the query in a format that the quantum computer can process. This involves encoding the data and query logic into quantum states, ensuring that the quantum operations align with the SQL semantics. It then employs and adapts a quantum

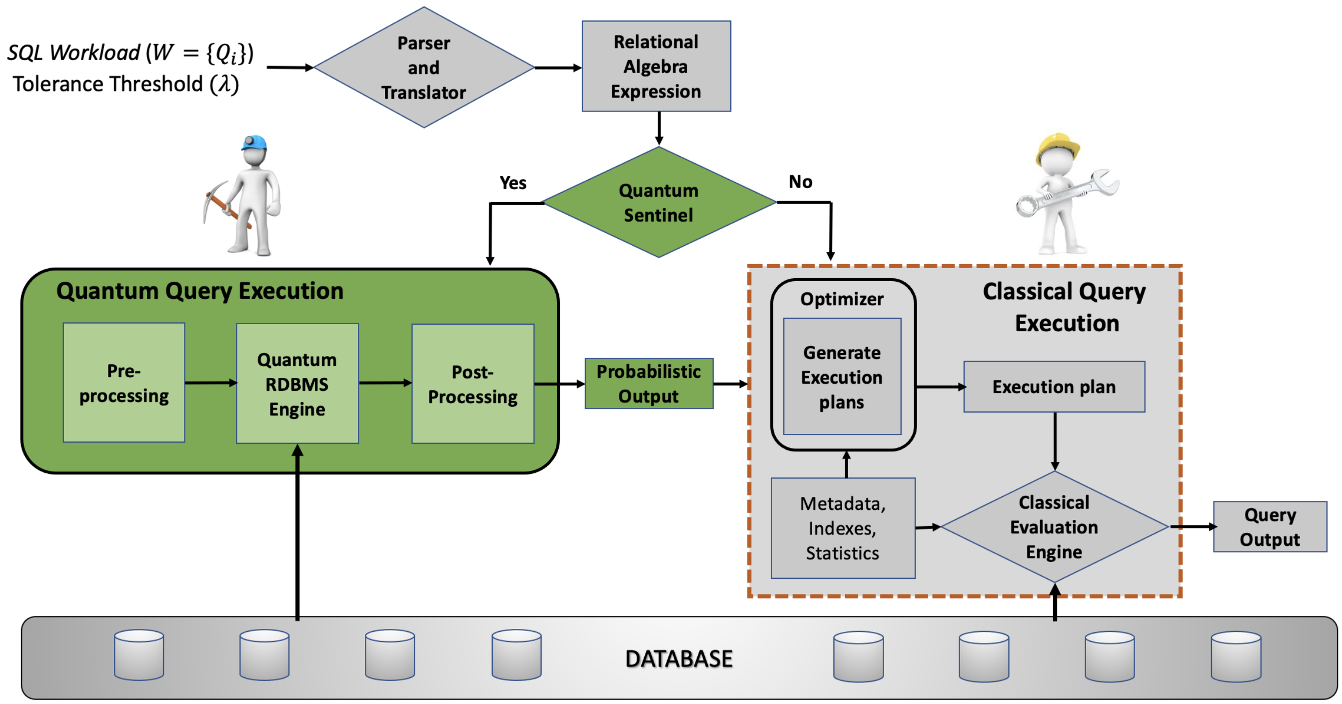

Figure 4: Quantum-Classical Hybrid SQL Query Execution

algorithm, such as the Grover Search (GS), to identify and produce qualifying tuples $R_i^{\pm}$, which may include both false positives and false negatives.

To enhance query execution, the engine should leverage quantum superposition and entanglement properties to process multiple potential solutions concurrently. This parallelism can considerably reduce the time complexity for certain classes of SQL queries, such as those involving complex filter predicates on non-indexed columns. The engine must also intelligently apply the quantum algorithm, dynamically adjusting its parameters to meet the user-defined tolerance threshold λ, thereby minimizing false negatives and ensuring that the results are both precise and reliable.

5.2 Classical RDBMS engine

Subsequently, the output of Quantum RDBMS Engine is passed to a classical RDBMS engine, which produces the final output tuples to fulfill the user request. The classical RDBMS engine plays a crucial role in this stage by deterministically filtering out any spurious false-positive tuples. Given the practical expectation that the output of the quantum processing is small in comparison to the input database, the overheads for removing false positives may be acceptable. Ideally, it should be achieved by leveraging the well-established lightweight index lookups and efficient validation mechanisms with simple post-processing. Thus, the classical engine ensures that only valid results are included in the final output. This dual-stage processing, where quantum computing accelerates the initial identification and classical computing ensures final accuracy, optimizes the overall query performance.

The updated contract is an initial effort towards addressing the research challenge identified in Section 4.1.1.

6 OUR VISION: QUANTUM-CLASSICAL HYBRID SQL QUERY EXECUTION

Figure 4 illustrates our vision of a quantum-classical hybrid RDBMS SQL query execution engine. In this vision, our objective is not to throw away over five decades of research and replace the existing classical RDBMS query execution engine. Instead, we aim to augment it and develop a hybrid execution engine. This approach will add the capabilities of quantum computation to the ambit of the existing RDBMS and, whenever it deems productive, will execute SQL queries more efficiently using quantum computers. The hybrid design will help create an RDBMS engine that leverages the strengths of both quantum and classical technologies.

Given an SQL query workload (W) along with a non-zero tolerance threshold (λ), each SQL query in the workload is first analyzed by a quantum sentinel component which decides whether the SQL query should follow the existing classical execution path or the new quantum-classical hybrid path. For instance, simple SQL queries that scan a whole relational table like `SELECT * FROM Nation` on the TPCH Nation table [31] may not be suitable candidates for the quantum-classical path. In contrast, more complex queries, such as the one illustrated in Figure 1, may benefit significantly from this hybrid processing. The quantum sentinel's decision will depend on various database characteristics, including query complexity, data distribution, user indexes, the estimated effort for classical post-processing and the user's tolerance threshold for errors. By effectively routing queries, the sentinel will optimize performance and accuracy within the hybrid system.

Moreover, the processing of each SQL query can be distributed between quantum and classical RDBMS engines. The quantum engine could focus on the segments of the SQL query that would benefit from quantum computation, thereby enhancing efficiency. Meanwhile, the classical DBMS engine will handle the rest of the query, producing the final output tuples. Furthermore, this design will help reduce the complexity of the generated quantum circuits, potentially allowing them to be supported on the currently limited quantum platforms. Therefore, we anticipate that this design could help reduce the requirements for control flows in quantum processing, effectively addressing the research challenge identified in Section 4.1.4.

From a business perspective, this hybrid design offers a seamless transition for users, allowing them to quickly adopt the quantum-classical RDBMS without necessitating any modifications to the application layer. This means that existing applications can continue to operate as usual, while benefiting from the performance improvements brought by quantum computing. This compatibility reduces barriers to entry, enabling organizations to integrate advanced quantum technologies into their database operations with minimal disruption and investment. Consequently, businesses can achieve enhanced data processing capabilities and maintain a competitive edge in a rapidly evolving technological landscape.

7 RELATED WORK

The first proposal for executing SQL queries on quantum platforms was presented in [6]. This work provides an abstract framework suggesting that basic operations like selection, projection, and join could be implemented using quantum primitives. However, a thorough analysis of the correctness of the system and implementation details is missing. For instance, the implementation of the join operation using a similarity operator, a combining operator, and GS algorithm needs to address the challenges associated with the use of GS as discussed in Section 2.2.

Elementary operations such as INSERT, UPDATE, DELETE, SELECT, and database backup and recovery are proposed in [38] as a set of Quantum Query Language (QQL) operations. However, these operations are limited to an implicit database consisting of a single column with unique values. Moreover, the work does not discuss the circuit implementation or the methods to identify the qualifying tuples. Next, in [19], a mechanism is provided to load a multi-column relational table and apply GS algorithm to identify the qualifying tuples, but it does not consider the INSERT and DELETE operations. Further, [13] extends both previous work and additionally exhibits an approach to perform an inner join between relational tables and then applies the GS algorithm to identify qualifying tuples. However, none of these works demonstrate the processing of a complex SQL query, such as the one shown in Figure 1, which includes GroupBy and Aggregate operations. Furthermore, in their restricted scope, they do not address key questions raised in Section 2.2 and Section 4, such as handling false positives, false negatives, and false duplicates, and ensuring the integrity and consistency of database operations in a quantum context.

An approach for performing set intersection using GS was proposed in [18]. This concept was further extended to handle set difference and union operations by leveraging the results of set intersections. The authors demonstrated the feasibility by implementing their approach using a toy example on IBM Qiskit [16]. However, they did not provide algorithmic details for the quantum circuit design in the general case. In particular, it was unclear how to construct the appropriate quantum oracle for set operations and how to address the issues related to the proper application of the GS algorithm.

Quantum-based resource allocation techniques to support virtual commits during transaction processing were proposed in [26]. They leverage the ideas of quantum superposition and measurement, allowing transactions to be committed without assigning concrete resource instances. And in [23, 27], the authors support retrieval-style SQL queries, which return result tuples associated with a degree of relevance to the query. They borrow concepts from the quantum mathematical model and quantum superposition to extend the relational domain calculus by integrating a similarity operator. However, in these works, the ideas are sourced from quantum computing principles, but the solutions are designed using classical paradigms and algorithms, and therefore it is not clear if these solutions can be implemented on a quantum platform.

8 CONCLUSIONS

In this paper, we explored the necessity and feasibility of Quantum-Based SQL query execution. Subsequently, we examined the impact of quantum computing on the traditional SQL query processing contract and identified the research challenges and benefits associated with leveraging quantum platforms for this task. Following this analysis, we proposed a Quantum-Classical Hybrid SQL Query contract aimed at enhancing the practicality and effectiveness of quantum-based SQL query execution. Finally, we articulated a vision for an SQL query execution engine that integrates both quantum and classical technologies. We hope that this discussion will spur new research efforts to address the challenges of making quantum-enhanced RDBMS systems a practical reality.

REFERENCES

[1] Tim Bittner and Sven Groppe. 2020. Avoiding blocking by scheduling transactions using quantum annealing. In *Proceedings of the 24th Symposium on International Database Engineering & Applications* (Seoul, Republic of Korea) *(IDEAS '20)*. Association for Computing Machinery, New York, NY, USA, Article 21, 10 pages.

[2] Tim Bittner and Sven Groppe. 2020. Hardware accelerating the optimization of transaction schedules via quantum annealing by avoiding blocking. *Open Journal of Cloud Computing (OJCC)* 7, 1 (2020), 1–21.

[3] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. 1998. Tight bounds on quantum searching. *Fortschritte der Physik: Progress of Physics* 46, 4-5 (1998), 493–505.

[4] Umut Çalikyilmaz, Sven Groppe, Jinghua Groppe, Tobias Winker, Stefan Prestel, Farida Shagieva, Daanish Arya, Florian Preis, and Le Gruenwald. 2023. Opportunities for quantum acceleration of databases: Optimization of queries and transaction schedules. *Proceedings of the VLDB Endowment* 16, 9 (2023), 2344–2353.

[5] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. 2021. Variational quantum algorithms. *Nature Reviews Physics* 3, 9 (2021), 625–644.

[6] Paul Cockshott. 1997. Quantum Relational Databases. arXiv:quant-ph/9712025 https://arxiv.org/abs/quant-ph/9712025

[7] D-Wave. 2024. *D-Wave Quantum Annealer*. D-Wave. Retrieved July 15, 2024 from https://www.dwavesys.com/learn/quantum-computing

[8] Tobias Fankhauser, Marc E. Solèr, Rudolf M. Füchslin, and Kurt Stockinger. 2021. Multiple Query Optimization using a Hybrid Approach of Classical and Quantum Computing. arXiv:2107.10508 https://arxiv.org/abs/2107.10508

[9] Jay Gambetta. 2022. *Quantum-centric supercomputing*. IBM. Retrieved July 15, 2024 from https://www.ibm.com/quantum/blog/next-wave-quantum-centric-supercomputing

[10] Sven Groppe and Jinghua Groppe. 2021. Optimizing Transaction Schedules on Universal Quantum Computers via Code Generation for Grover's Search Algorithm. In *Proceedings of the 25th International Database Engineering & Applications Symposium* (Montreal, QC, Canada) *(IDEAS '21)*. Association for Computing Machinery, New York, NY, USA, 149–156.

[11] Lov K. Grover. 1996. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (Philadelphia, Pennsylvania, USA) *(STOC '96)*. Association for Computing Machinery, New York, NY, USA, 212–219.

[12] Le Gruenwald, Tobias Winker, Umut Çalikyilmaz, Jinghua Groppe, and Sven Groppe. 2023. Index Tuning with Machine Learning on Quantum Computers for Large-Scale Database Applications. In *Joint Proceedings of Workshops at the 49th International Conference on Very Large Data Bases (VLDB 2023), Vancouver, Canada, August 28 - September 1, 2023 (CEUR Workshop Proceedings)*, Vol. 3462. CEUR-WS.org.

[13] Amor Gueddana, Rihab Chatta, and Noureddine Boudriga. 2010. Optimized methods for inserting and deleting records and data retrieving in quantum database. In *2010 12th International Conference on Transparent Optical Networks*. IEEE, 1–5.

[14] Khabat Heshami, Duncan G England, Peter C Humphreys, Philip J Bustard, Victor M Acosta, Joshua Nunn, and Benjamin J Sussman. 2016. Quantum memories: emerging applications and recent advances. *Journal of modern optics* 63, 20 (2016), 2005–2028.

[15] Petr Ivashkov, Gideon Uchehara, Liang Jiang, Derek S. Wang, and Alireza Seif. 2023. High-fidelity, multi-qubit generalized measurements with dynamic circuits. arXiv:2312.14087 [quant-ph] https://arxiv.org/abs/2312.14087

[16] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. 2024. Quantum computing with Qiskit. https://doi.org/10.48550/arXiv.2405.08810 arXiv:2405.08810 [quant-ph]

[17] Mark R Jerrum, Leslie G Valiant, and Vijay V Vazirani. 1986. Random generation of combinatorial structures from a uniform distribution. *Theoretical computer science* 43 (1986), 169–188.

[18] Szabolcs Jóczik and Attila Kiss. 2020. Quantum Computation and Its Effects in Database Systems. In *New Trends in Databases and Information Systems - ADBIS 2020 Short Papers, Lyon, France, August 25-27, 2020, Proceedings (Communications in Computer and Information Science)*, Vol. 1259. Springer, 13–23.

[19] Yi-Lin Ju, I-Ming Tsai, and Sy-Yen Kuo. 2007. Quantum circuit design and analysis for database search applications. *IEEE Transactions on Circuits and Systems I: Regular Papers* 54, 11 (2007), 2552–2563.

[20] Manish Kesarwani and Jayant Haritsa. 2024. Index Advisors on Quantum Platforms. *Proceedings of the VLDB Endowment* 17, 11 (2024).

[21] Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout Van Den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme, et al. 2023. Evidence for the utility of quantum computing before fault tolerance. *Nature* 618, 7965 (2023), 500–505.

[22] Hartmut Klauck. 2003. Quantum Time-Space Tradeoffs for Sorting. arXiv:quant-ph/0211174 https://arxiv.org/abs/quant-ph/0211174

[23] Sebastian Lehrack and Ingo Schmitt. 2010. QSQL: Incorporating Logic-Based Retrieval Conditions into SQL. In *Database Systems for Advanced Applications, 15th International Conference, DASFAA 2010, Tsukuba, Japan, April 1-4, 2010, Proceedings, Part I (Lecture Notes in Computer Science)*, Hiroyuki Kitagawa, Yoshiharu Ishikawa, Qing Li, and Chiemi Watanabe (Eds.), Vol. 5981. Springer, 429–443.

[24] Nitin Nayak, Jan Rehfeld, Tobias Winker, Benjamin Warnke, Umut Çalikyilmaz, and Sven Groppe. 2023. Constructing Optimal Bushy Join Trees by Solving QUBO Problems on Quantum Hardware and Simulators. In *Proceedings of the International Workshop on Big Data in Emergent Distributed Environments* (Seattle, WA, USA) *(BiDEDE '23)*. Association for Computing Machinery, New York, NY, USA, Article 7, 7 pages.

[25] Michael A Nielsen and Isaac Chuang. 2002. Quantum computation and quantum information.

[26] Sudip Roy, Lucja Kot, and Christoph Koch. 2013. Quantum Databases. In *Sixth Biennial Conference on Innovative Data Systems Research, CIDR 2013, Asilomar, CA, USA, January 6-9, 2013, Online Proceedings*. www.cidrdb.org.

[27] Ingo Schmitt. 2008. Qql: A db&ir query language. *The VLDB journal* 17, 1 (2008), 39–56.

[28] Manuel Schönberger. 2022. Applicability of Quantum Computing on Database Query Optimization *(SIGMOD '22)*. Association for Computing Machinery, New York, NY, USA, 2512–2514.

[29] Manuel Schönberger, Stefanie Scherzinger, and Wolfgang Mauerer. 2023. Ready to Leap (by Co-Design)? Join Order Optimisation on Quantum Hardware. *Proc. ACM Manag. Data* 1, 1, Article 92 (may 2023), 27 pages.

[30] Maria Schuld and Francesco Petruccione. 2018. *Supervised learning with quantum computers*. Vol. 17. Springer.

[31] Transaction Processing Council. [n.d.]. *TPC-H*. www.tpc.org/tpch

[32] Immanuel Trummer and Christoph Koch. 2015. Multiple Query Optimization on the D-Wave 2X Adiabatic Quantum Computer. *Proceedings of the VLDB Endowment* 9 (10 2015). https://doi.org/10.14778/2947618.2947621

[33] Valter Uotila. 2022. Synergy between Quantum Computers and Databases. In *Proceedings of the VLDB 2022 PhD Workshop co-located with the 48th International Conference on Very Large Databases (VLDB 2022), Sydney, Australia, September 5, 2022 (CEUR Workshop Proceedings)*, Vol. 3186. CEUR-WS.org.

[34] Valter Uotila. 2023. SQL2Circuits: Estimating Metrics for SQL Queries with A Quantum Natural Language Processing Method. *CoRR* abs/2306.08529 (2023). arXiv:2306.08529

[35] Tobias Winker, Umut Çalikyilmaz, Le Gruenwald, and Sven Groppe. 2023. Quantum Machine Learning for Join Order Optimization using Variational Quantum Circuits *(BiDEDE '23)*. Association for Computing Machinery, New York, NY, USA, Article 5, 7 pages.

[36] Tobias Winker, Sven Groppe, Valter Uotila, Zhengtong Yan, Jiaheng Lu, Maja Franz, and Wolfgang Mauerer. 2023. Quantum Machine Learning: Foundation, New Techniques, and Opportunities for Database Research. In *Companion of the 2023 International Conference on Management of Data, SIGMOD/PODS 2023, Seattle, WA, USA, June 18-23, 2023*. ACM, 45–52.

[37] William K Wootters and Wojciech H Zurek. 2009. The no-cloning theorem. *Physics Today* 62, 2 (2009), 76–77.

[38] Ahmed Younes. 2007. Database Manipulation on Quantum Computers. arXiv:0705.4303 https://arxiv.org/abs/0705.4303

[39] Gongsheng Yuan, Jiaheng Lu, Yuxing Chen, Sai Wu, Chang Yao, Zhengtong Yan, Tuodu Li, and Gang Chen. 2023. Quantum Computing for Databases: A Short Survey and Vision. In *Joint Proceedings of Workshops at the 49th International Conference on Very Large Data Bases (VLDB 2023), Vancouver, Canada, August 28 - September 1, 2023 (CEUR Workshop Proceedings)*, Vol. 3462. CEUR-WS.org.