Quantum-Ready Approximate Data Management using Tensor-Network Physical Pages

Muzhi Chen
Shanghai Jiao Tong University
inefable@sjtu.edu.cn

Hanwen Liu
University of Southern California
hanwen_liu@usc.edu

Xuanhe Zhou
Shanghai Jiao Tong University
zhouxuanhe@sjtu.edu.cn

Hongming Xu
Shanghai Jiao Tong University
muzhihai@sjtu.edu.cn

Zhenghao Li
Shanghai Jiao Tong University
lzhwasd123@sjtu.edu.cn

Wei Zhou
Shanghai Jiao Tong University
weizhoudb@sjtu.edu.cn

Fan Wu
Shanghai Jiao Tong University
fwu@cs.sjtu.edu.cn

ABSTRACT

Gate-based quantum routines cannot directly consume byte-oriented database pages; without practical QRAM, data loading can dominate execution cost. We present TN-Page, an auxiliary representation that bucketizes a relational block into an empirical distribution and factorizes it into tensor-network cores. The source remains authoritative for exact and transaction-critical semantics, whereas TN-Page exposes quality and resource metadata for routing requests among authoritative classical execution, approximate classical tensor contraction, and quantum export. On eight-attribute blocks with 100K tuples, TN-Page achieves 0.873–0.884 reconstruction fidelity and 1.8–1.9× compression. Backend-guided estimates reduce qubits by up to 32% and circuit depth by up to 95% versus native-binary loading; reduced loading fragments execute on Origin Wukong. The results establish representation-level and reduced-circuit feasibility, not end-to-end quantum advantage.

VLDB Workshop Reference Format:
Muzhi Chen, Hanwen Liu, Xuanhe Zhou, Hongming Xu, Zhenghao Li, Wei Zhou, and Fan Wu. Quantum-Ready Approximate Data Management using Tensor-Network Physical Pages. VLDB 2026 Workshop: QC&DKM 2026- 2nd Workshop on Quantum Computing and Data/Knowledge Management.

VLDB Workshop Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/OpenDataBox/Qute.

1 INTRODUCTION

Relational DBMSs rely on byte-addressable storage layouts and fixed-size pages to reconstruct tuples exactly for query processing, logging, recovery, and transaction management [1, 12]. While such storage layouts are successful for classical execution, gate-based quantum processors cannot directly consume buffer pages or byte streams. To enable quantum operations on relational information, the data must be transformed into a quantum-accessible representation, such as a prepared state, an oracle, or another circuit-level interface.

Proceedings of the VLDB Endowment. ISSN 2150-8097.

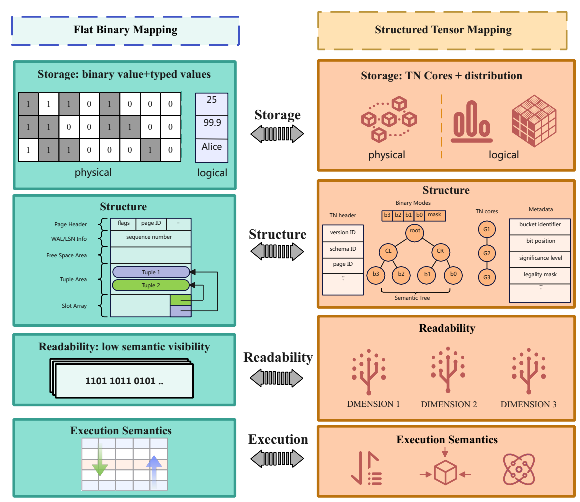

Figure 1: Comparison of byte-oriented native page storage and tensor-network page representation.

Recent studies have investigated quantum data management, quantum database operations, quantum storage, and hybrid quantum-classical execution for database workloads [9, 15, 16, 18–22, 28]. Other work explores quantum and quantum-inspired methods for query optimization and approximate query processing [3, 13, 17, 23, 24, 31]. While these efforts establish vital directions for integrating quantum computing into data systems, most assume the input is already encoded into a quantum-usable form. The preceding storage-layer question remains insufficiently addressed: *how should a conventional relational block be represented so that it can be reused as a valid, resource-aware quantum loading target?*

We argue that this loading question should be treated as a physical-design problem rather than an operator-local compilation detail. Constructing a separate encoding for each quantum operator repeatedly incurs loading costs at query time without leaving a persistent object that the DBMS can inspect, reuse, validate, or bypass. To solve this, we maintain the classical source block as authoritative for exact semantics, transactions, updates, and recovery, and design a derived quantum-oriented object summarizing selected distributional structures, deployed only when its approximation quality and resource costs match the request.

We introduce TN-Page, an auxiliary tensor-network physical representation for a page-sized relational block. Rather than replacing the DBMS page format or claiming persistent quantum storage, TN-Page provides a reusable, workload-dependent synopsis closer to a state-preparation target while linking back to the authoritative source. TN-Page differs from conventional byte-oriented pages in four aspects. **Storage:** it represents selected page contents as a bucketized empirical distribution instead of a flat byte image. **Structure:** it organizes this distribution as a semantic tensor, whose modes correspond to attributes, bucket groups, predicate-relevant dimensions or encoded bit groups, while tensor-network bonds preserve selected cross-attribute dependencies. **Readability:** it stores DBMS-visible metadata alongside tensor cores, including bucket boundaries, mode mappings, legality information, reconstruction quality, and resource estimates. **Execution:** these metadata allow the optimizer to distinguish among three paths: authoritative classical execution over source data, approximate classical tensor contraction, and quantum export to a constrained backend.

Tensor networks suit this role by representing structured high-dimensional tensors via compact cores with bounded internal dimensions. In particular, matrix-product-state and tensor-train representations expose an explicit trade-off between retained correlation, representation size, and contraction cost [26, 27, 29]. TN-Page applies this principle inversely to classical quantum simulation: instead of using tensor networks to simulate a quantum state on a classical machine, it preprocesses a relational distribution into a factorized object that can later serve as a classical approximation target or a quantum loading target.

In summary, we make the following contributions.

(1) We formalize database-to-quantum data loading as a storage-layer design problem rather than operator-level encoding, emphasizing page-level structure and resource considerations.
(2) We propose TN-Page, a tensor-network auxiliary page that summarizes a classical page with semantic layouts and DBMS-visible metadata, preserving relational structure while supporting approximate and distributional workloads.
(3) We make fidelity, approximation error, qubit usage, circuit depth, and backend feasibility visible to the optimizer, enabling safe selection among classical execution, classical tensor contraction, and quantum export.
(4) We evaluate TN-Page on synthetic 8-attribute, 100K-tuple blocks. It achieves 0.873–0.884 reconstruction fidelity and 1.8–1.9× compression. Under an Origin Wukong-guided resource model, it reduces estimated qubit demand by up to 32% and routed depth by up to 95% over native-binary loading. Hardware-feasible fragments are validated on Origin Wukong.

2 PRELIMINARY

2.1 Loading and Representations

A gate-based quantum backend cannot directly consume a byte-oriented database page. Given a page-sized relational block P, TN-Page first derives a bucketized empirical distribution $p_P(\mathbf{i})$ over attribute coordinates $\mathbf{i} = (i_1, \ldots, i_m)$. Its quantum loading target is

$$|\psi_P\rangle = \sum_{\mathbf{i} \in \mathcal{D}_P} \sqrt{p_P(\mathbf{i})}\, |\mathrm{enc}(\mathbf{i})\rangle ,$$

where $\mathrm{enc}(\mathbf{i})$ maps each valid bucket coordinate to a computational-basis label. Ideally, a QRAM interface would support coherent access of the form $\sum_j \alpha_j |j\rangle |0\rangle \longrightarrow \sum_j \alpha_j |j\rangle |d_j\rangle$, which associates every address component in a superposition with its corresponding data value [7]. However, this abstraction requires all addressed paths to remain coherent, values to be written reversibly into the output register, and routing ancillas to be uncomputed without observing intermediate states. Lacking practical hardware QRAM primitives as a DBMS storage interface, coherent access must be compiled into explicit state-preparation, controlled-routing, oracle, and measurement circuits [8, 25]. Their qubit demand, two-qubit connectivity, circuit depth, and accumulated noise can outweigh the benefit of the intended quantum routine on NISQ hardware [9, 13].

A dense representation of p_P requires $\prod_{j=1}^{m} d_j$ entries for bucket cardinalities $d_1, \ldots, d_m$. Tensor networks instead represent this distribution through compact connected cores. TN-Page uses a tensor-train (TT), or matrix-product-state (MPS) approximation:

$$\widehat{p}_P(i_1, \ldots, i_m) = G_1(i_1) G_2(i_2) \cdots G_m(i_m),$$

where the internal bond dimensions control the trade-off between cross-attribute correlations and representation cost [26, 27, 29].

2.2 Related Work

Data management for quantum computing. Recent work has studied database abstractions around quantum execution, including quantum database interfaces, hybrid execution stacks, and data-access models for NISQ hardware [4, 9]. These systems clarify how database operators or query plans may interact with quantum processors. TN-Page addresses a lower-level problem. It treats the database page as the unit of physical design and exposes quality and resource contracts that determine when an approximate page representation can be used safely.

Quantum techniques for database tasks. Quantum and quantum-inspired methods have been explored for database problems such as query optimization, multiple-query optimization, query containment, and approximate query processing [6, 30–32]. While these focus on logical operators, optimizer search, or algorithmic primitives after data encoding, TN-Page targets the storage-to-execution boundary. It specifies how page-level data are represented, validated, and exported before a quantum execution path is selected.

Approximate synopses in database systems. Database systems use samples, sketches, histograms, materialized summaries, and learned estimators to accelerate approximate analytics and cardinality estimation [2, 5, 10, 11, 14]. TN-Page is closest to this family as an auxiliary page-level synopsis with explicit error info, but adds a quantum-exportable structure. The same object can support classical tensor contraction, provide a state-preparation target, and expose backend resource estimates to the optimizer.

3 TN-PAGE PHYSICAL REPRESENTATION

TN-Page is an auxiliary physical object associated with an authoritative relational block P. It does not replace the original page: exact tuple access, transaction-critical execution, updates, logging, and

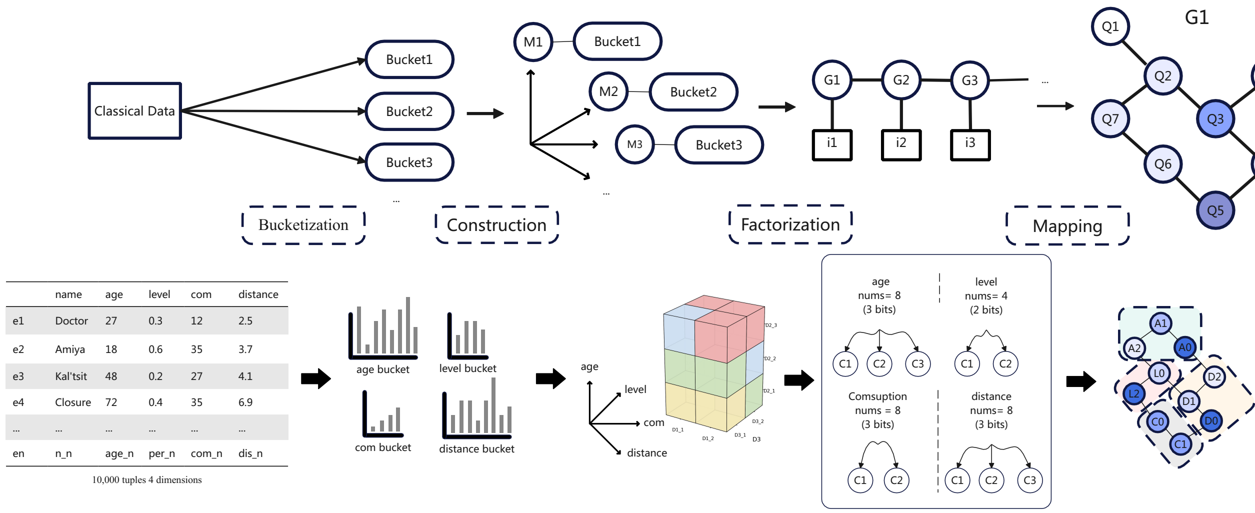

Figure 2: Pipeline of a TN-Page construction.

recovery remain on the classical source. Instead, TN-Page materializes a reusable distributional representation for approximate or quantum-oriented workloads.

For a source block P, we represent the object as

$$\Pi_P = (\text{Header}, \text{Encoding}, \text{Layout}, \text{Cores}, \text{Contract}).$$

The **header** records the source-page link, schema and version identifiers, tensor shape, and construction information. The **encoding** records bucket boundaries, binary mode mappings, and legality information for padded binary states. The **layout** maps tensor modes and mode groups back to database-visible attributes or predicate-relevant regions. The **cores** store the factorized empirical distribution, with bond dimensions controlling the retained cross-attribute dependence. The **contract** records representation quality and estimated resource requirements, including reconstruction fidelity, truncation statistics, core dimensions, and backend-oriented qubit and depth estimates.

This organization gives TN-Page two properties absent from operator-local quantum encodings. First, it is persistent and reusable: multiple requests can reference the same page-level distribution without independent reconstruction. Second, it is safely bypassable: the DBMS can ignore TN-Page whenever exact semantics are required or its quality and resource conditions are not satisfied.

The contract records database semantics, valid encoded states, representation quality, and backend resource estimates. It does not itself prescribe an execution policy; instead, it supplies the evidence consumed by the subsequent operations.

4 TN-PAGE CONSTRUCTION PIPELINE

Figure 2 presents how we construct a TN-Page from a database page by converting page contents into a workload-aware empirical distribution and then factorizing the distribution into tensor-network cores. The construction produces an auxiliary physical object. It does not replace the original page, which remains the source of exact tuple semantics, recovery, and transactional execution.

4.1 Bucketization and Construction

The input is a page-sized relational block P with N tuples and m attributes. A tuple is denoted by $x^{(r)}$, and its value on attribute D_j is denoted by $x_j^{(r)}$. For each attribute, TN-Page defines a bucketization function $B_j : D_j \to [d_j]$, where d_j is chosen according to the query tolerance and workload importance of the attribute. Attributes frequently used in range predicates receive finer buckets, while contextual attributes may use coarser buckets.

After bucketization, each tuple is mapped to a coordinate $(i_1, \ldots, i_m)$, where $i_j = B_j(x_j^{(r)})$. TN-Page constructs the page distribution

$$\mathcal{T}_P(i_1, \ldots, i_m) = \frac{1}{N} \sum_{r=1}^{N} \mathbf{1}\left[(B_1(x_1^{(r)}), \ldots, B_m(x_m^{(r)})) = (i_1, \ldots, i_m) \right].$$

For quantum export, TN-Page uses the amplitude lift of this distribution, $A_P(i_1, \ldots, i_m) = \sqrt{T_P(i_1, \ldots, i_m)}$. The amplitude tensor, rather than the probability tensor itself, is factorized in the following stage so that its reconstruction can directly define a normalized state-preparation target. The tensor $\mathcal{T}_P$ records probability mass over bucket combinations. It is a distributional representation of the page rather than a tuple-level copy.

4.2 Semantic Tensor Factorization

TN-Page factorizes the empirical distribution into tensor-network cores. In the default linear layout, the representation is

$$\widehat{\mathcal{A}}_P(i_1, \ldots, i_m) \approx G_1(i_1) G_2(i_2) \cdots G_m(i_m).$$

Each core G_j maps to a database attribute, and each bond captures the adjacent dependency. The bond dimension balances reconstruction fidelity, storage cost and contraction overheads.

The layout is selected as a physical design decision. Frequently co-used attributes can be placed nearby to capture their correlation via a small bond. High-cardinality attributes can be bucketized before factorization to limit physical dimension size. When the workload favors hierarchy, TN-Page uses a tree layout. For a tree cut separating dimensions S_v from $S_{\neg v}$, the builder applies a truncated decomposition $\mathcal{A}_{P,(S_v|S_{\neg v})} \approx U_r \Sigma_r V_r^\top$, where r is bounded by the maximum bond dimension and the local error budget. Each edge records bond dimensions, singular values, and truncation errors.

4.3 Mapping TN-Page to Qubit Registers

After factorization, TN-Page maps each tensor mode to a logical qubit register. For a mode with d_k valid buckets, it allocates $n_k = \lceil \log_2 d_k \rceil$ qubits and maps bucket i_k to its binary basis label $\mathrm{enc}_k(i_k)$. When d_k is not a power of two, unused basis labels are recorded in the legality mask and excluded during export and decoding.

TN-Page factorizes the amplitude lift $A_P(\mathbf{i}) = \sqrt{T_P(\mathbf{i})}$ rather than the probability tensor directly. For a valid bucket coordinate $\mathbf{i} = (i_1, \ldots, i_m)$, the tensor cores define the unnormalized amplitude and quantum loading target as

$$|\widehat{\psi}_P\rangle = \frac{1}{\sqrt{Z_P}} \sum_{\mathbf{i} \in \mathcal{D}_P} \widehat{A}_P(\mathbf{i}) \, |\mathrm{enc}(\mathbf{i})\rangle,$$

where $Z_P = \sum_{\mathbf{i} \in \mathcal{D}_P} |\widehat{A}_P(\mathbf{i})|^2$ normalizes the exported state.

Thus, measuring $|\widehat{\psi}_P\rangle$ yields the approximate bucket distribution $|\widehat{A}_P(\mathbf{i})|^2 / Z_P$. The mapping preserves the database meaning of each basis state as a valid bucket combination, rather than treating tensor indices as opaque array offsets. It also provides the logical register layout required by the later quality-resource contract and backend export procedure.

4.4 Maintaining TN-Page Validity

TN-Page maintains a set of encoding and quality metadata to keep the tensor representation consistent with database semantics and quantum export requirements. For quantum export, each bucket coordinate is encoded as a computational-basis label.

Meanwhile, TN-Page records validity masks to exclude these states during export, decoding, and error estimation. Together with encoding metadata, factorization-quality statistics, and backend resource estimates, these fields form the quality-resource contract used by the optimizer.

Beyond binary encoding, TN-Page stores DBMS-readable metadata that connects tensor cores back to page semantics. This metadata includes bucket boundaries, bucket identifiers, attribute-to-mode mappings, dimension order, workload tags, validity masks, bond dimensions, singular spectra, truncation errors, reconstruction fidelity, compression statistics, and backend export estimates. Together, these fields make TN-Page more than a compressed tensor: they define the quality-resource contract exposed to the optimizer. The optimizer can identify which tensor modes correspond to query predicates, which basis states are valid, how much approximation error are introduced, and whether TN-Page satisfies the quality and resource requirements of a candidate execution path.

5 PRELIMINARY RESULTS

Figure 3 reports backend-guided resource estimates parameterized by Origin Wukong 180. We evaluate TN-Page using synthetic relational blocks with eight attributes and 100K tuples. The evaluation considers both representation quality and loading feasibility. Across the evaluated configurations, TN-Page retains 0.873–0.884 reconstruction fidelity and achieves 1.8–1.9× compression relative to the dense bucketized distribution. These results indicate that the tensor factorization preserves the dominant bucket-level correlations while reducing the size of the representation.

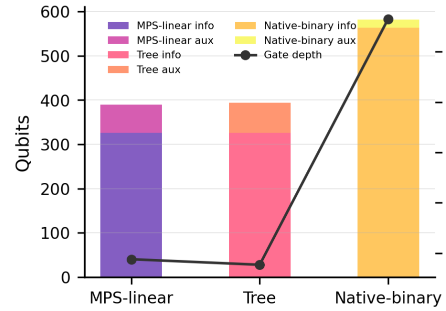

Figure 3: Estimated qubit consumption on uniform dataset.

We further compare TN-Page layouts with a compact native-binary loading baseline under an Origin Wukong-guided backend model. The factorized layouts reduce estimated qubit demand by up to 32% and routed circuit depth by up to 95%, with the tree layout minimizing the preparation dependency chain. To validate the real-device path, we compile and execute reduced hardware-feasible loading fragments on Origin Wukong and collect measurement outcomes. The page-scale resource figures remain backend-constrained compiler estimates because the full logical configurations exceed the available physical budget. Thus, the results establish representation-level feasibility and reduced-circuit hardware executability, rather than end-to-end query speedup or accuracy.

6 CONCLUSION AND FUTURE WORK

TN-Page is an auxiliary tensor-network representation that converts a relational block into a compact bucketized distribution while preserving the classical source as the authority for exact and transaction-critical semantics. Its metadata retains the semantic, quality, and resource information needed to reuse the same object for later classical or quantum-oriented operations. Across synthetic eight-attribute blocks with 100K tuples, TN-Page achieves 0.873–0.884 reconstruction fidelity and 1.8–1.9× compression, while backend-guided estimates indicate lower loading resources than native-binary encoding.

Future work will validate TN-Page at query granularity on real analytical pages, support version-consistent refresh under updates, and calibrate its resource estimates against backend-specific compilation and noise measurements.

ACKNOWLEDGMENTS

Xuanhe Zhou is the corresponding author. This work was supported in part by National Key R&D Program of China (No. 2023YFB4502400), China NSF grants (No. 62441236, 62372296, 62432007, U25A6024, U25A20437, 62525202, 62232009, 62502304), Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM103), ByteDance, Tencent, Ant Group through CCF-Ant Research Fund, Tencent Rhino Bird Key Research Project, Huawei Innovation Research Programs, Shenzhen Project (CJGJZD20230724093403007), Zhongguancun Lab, Beijing National Research Center for Information Science and Technology (BNRist), and Shanghai Jiao Tong University AI for Engineering Initiative.

REFERENCES

[1] Daniel Abadi, Peter Boncz, Stavros Harizopoulos, Stratos Idreos, and Samuel Madden. 2013. The Design and Implementation of Modern Column-Oriented Database Systems. *Foundations and Trends in Databases* 5, 3 (2013), 197–280. https://doi.org/10.1561/1900000024

[2] Sameer Agarwal, Barzan Mozafari, Aurojit Panda, Henry Milner, Samuel Madden, and Ion Stoica. 2013. BlinkDB: Queries with Bounded Errors and Bounded Response Times on Very Large Data. In *Proceedings of the 8th ACM European Conference on Computer Systems*. 29–42. https://doi.org/10.1145/2465351.2465355

[3] 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. *Proc. VLDB Endow.* 16, 9 (2023), 2344–2353.

[4] Muzhi Chen, Xuanhe Zhou, Wei Zhou, Bangrui Xu, Surui Tang, Guoliang Li, Bingsheng He, Yeye He, Yitong Song, and Fan Wu. 2026. Qute: Towards Quantum-Native Database. *CoRR* abs/2602.14699 (2026). https://doi.org/10.48550/arXiv.2602.14699

[5] Yannis Chronis, Yawen Wang, Yu Gan, Sami Abu-El-Haija, Chelsea Lin, Carsten Binnig, and Fatma Özcan. 2024. CardBench: A Benchmark for Learned Cardinality Estimation in Relational Databases. *CoRR* abs/2408.16170 (2024). https://doi.org/10.48550/arXiv.2408.16170

[6] Luisa Gerlach, Tobias Köppl, René Zander, Nicole Schweikardt, and Stefanie Scherzinger. 2026. Quantum Computing for Query Containment of Conjunctive Queries. *CoRR* abs/2602.21803 (2026). https://doi.org/10.48550/arXiv.2602.21803

[7] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. 2008. Quantum Random Access Memory. *Physical Review Letters* 100, 16 (2008), 160501. https://doi.org/10.1103/PhysRevLett.100.160501

[8] Martin Gogeißl, Hila Safi, and Wolfgang Mauerer. 2024. Quantum Data Encoding Patterns and their Consequences. In *Proceedings of the Workshop on Quantum Computing and Quantum-Inspired Technology for Data-Intensive Systems and Applications (Q-Data '24)*. Association for Computing Machinery, 27–37. https://doi.org/10.1145/3665225.3665446

[9] Rihan Hai, Shih-Han Hung, Tim Coopmans, Tim Littau, and Floris Geerts. 2025. Quantum Data Management in the NISQ Era. *Proceedings of the VLDB Endowment* 18, 6 (2025), 1720–1733. https://doi.org/10.14778/3725688.3725701

[10] Yuxing Han, Rong Zhu, Kai Zeng, Gao Cong, and Andreas Pfadler. 2021. Cardinality Estimation in DBMS: A Comprehensive Benchmark Evaluation. *CoRR* abs/2109.05877 (2021). https://doi.org/10.48550/arXiv.2109.05877

[11] Joseph M. Hellerstein, Peter J. Haas, and Helen J. Wang. 1997. Online Aggregation. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*. 171–182. https://doi.org/10.1145/253260.253291

[12] Joseph M. Hellerstein, Michael Stonebraker, and James Hamilton. 2007. Architecture of a Database System. *Foundations and Trends in Databases* 1, 2 (2007), 141–259. https://doi.org/10.1561/1900000002

[13] Manish Kesarwani and Jayant R. Haritsa. 2024. Is Quantum-Based SQL Query Execution Viable?. In *Proceedings of the Second International Workshop on Quantum Data Science and Management (QDSM '24)*. VLDB Endowment. https://vldb.org/workshops/2024/proceedings/QDSM/QDSM.8.pdf

[14] Kaiyu Li and Guoliang Li. 2018. Approximate Query Processing: What is New and Where to Go? A Survey on Approximate Query Processing. *Data Science and Engineering* 3, 4 (2018), 379–397. https://doi.org/10.1007/s41019-018-0074-4

[15] Tuodu Li, Gongsheng Yuan, Chang Yao, Meng Shi, Ziyue Wang, Ling Qian, and Jiaheng Lu. 2024. Quantum Storage Design for Tables in RDBMS. In *Proceedings of the Second International Workshop on Quantum Data Science and Management (QDSM '24)*. VLDB Endowment. https://vldb.org/workshops/2024/proceedings/QDSM/QDSM.4.pdf

[16] Tim Littau, Ziyu Li, and Rihan Hai. 2024. Towards Quantum Data Structures for Enhanced Database Performance. In *Proceedings of the Second International Workshop on Quantum Data Science and Management (QDSM '24)*. VLDB Endowment. https://vldb.org/workshops/2024/proceedings/QDSM/QDSM.6.pdf

[17] Hanwen Liu, Abhishek Kumar, Federico M. Spedalieri, and Ibrahim Sabek. 2025. Hybrid Quantum-Classical Optimization for Bushy Join Trees. In *Proceedings of the 2nd Workshop on Quantum Computing and Quantum-Inspired Technology for Data-Intensive Systems and Applications (Q-Data '25)*. Association for Computing Machinery, 20–24. https://doi.org/10.1145/3736393.3736695

[18] Hanwen Liu, Abhishek Kumar, Federico M. Spedalieri, and Ibrahim Sabek. 2026. QDBO: A Real-time Quantum-augmented Database System Optimizer. *Proceedings of the VLDB Endowment* (2026).

[19] Hanwen Liu, Yiran Li, Ibrahim Sabek, Xuanhe Zhou, and Hans-Arno Jacobsen. 2026. How Can Quantum Computing and Databases Meet 'in Practice'? From Algorithms to Systems. *Proceedings of the VLDB Endowment* 19, 12 (2026).

[20] Hanwen Liu and Ibrahim Sabek. 2026. Is Quantum Computing Ready for Real-Time Database Optimization?. In *2026 IEEE 42nd International Conference on Data Engineering (ICDE)*. IEEE.

[21] Hanwen Liu and Ibrahim Sabek. 2026. QFusion: A Demonstration of Boundary-Aware Fusion Planning and Execution for Large-Scale QUBO Optimization. *Proceedings of the VLDB Endowment* 19, 12 (2026).

[22] Hanwen Liu and Ibrahim Sabek. 2026. Towards a Hybrid Quantum-Classical Computing Framework for Database Optimization Problems in Real Time Setup. In *2026 IEEE 42nd International Conference on Data Engineering (ICDE), DEFT Track*. IEEE.

[23] Hanwen Liu, Pranshi Saxena, Federico M. Spedalieri, and Ibrahim Sabek. 2025. Optimizing Join Orders via Constrained Quadratic Models. In *2025 IEEE 41st International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 295. https://doi.org/10.1109/ICDEW67478.2025.00042

[24] Hanwen Liu, Federico M. Spedalieri, and Ibrahim Sabek. 2025. A Demonstration of Q^2O: Quantum-augmented Query Optimizer. *Proceedings of the VLDB Endowment* 18, 12 (2025), 5439–5443. https://doi.org/10.14778/3750601.3750691

[25] Michael A. Nielsen and Isaac L. Chuang. 2010. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press. https://doi.org/10.1017/CBO9780511976667

[26] Román Orús. 2014. A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States. *Annals of Physics* 349 (2014), 117–158. https://doi.org/10.1016/j.aop.2014.06.013

[27] Ivan V. Oseledets. 2011. Tensor-Train Decomposition. *SIAM Journal on Scientific Computing* 33, 5 (2011), 2295–2317. https://doi.org/10.1137/090752286

[28] Carla Rieger, Michele Grossi, Gian Giacomo Guerreschi, Sofia Vallecorsa, and Martin Werner. 2024. Operational Framework for a Quantum Database. *CoRR* abs/2405.14947 (2024). https://doi.org/10.48550/arXiv.2405.14947

[29] Ulrich Schollwöck. 2011. The Density-Matrix Renormalization Group in the Age of Matrix Product States. *Annals of Physics* 326, 1 (2011), 96–192. https://doi.org/10.1016/j.aop.2010.09.012

[30] Immanuel Trummer and Christoph Koch. 2016. Multiple Query Optimization on the D-Wave 2X Adiabatic Quantum Computer. *Proceedings of the VLDB Endowment* 9, 9 (2016), 648–659. https://doi.org/10.14778/2947618.2947621

[31] Sai Wu, Meng Shi, Dongxiang Zhang, Junbo Zhao, Gongsheng Yuan, and Gang Chen. 2024. When Quantum Computing Meets Database: A Hybrid Sampling Framework for Approximate Query Processing. *IEEE Transactions on Knowledge and Data Engineering* 36, 12 (2024), 9532–9546. https://doi.org/10.1109/TKDE.2024.3428498

[32] Gongsheng Yuan, Yuxing Chen, Jiaheng Lu, Sai Wu, Zhiwei Ye, Ling Qian, and Gang Chen. 2024. Quantum Computing for Databases: Overview and Challenges. *CoRR* abs/2405.12511 (2024). https://doi.org/10.48550/arXiv.2405.12511