The OSCAR PUBLISHING Services

International Journal Of Management And Economics Fundamental

Elastic and Integrated Cloud Data Warehousing: Architectural, Analytical, and Governance Insights from Amazon Redshift

Dr. Diego Montoya

University of Novi Sad, Serbia

Received: 28 November 2025; **Accepted:** 18 December 2025; **Published:** 31 December 2025

Abstract: Cloud data warehousing has evolved from a narrowly defined analytical storage paradigm into a complex, multi-layered ecosystem in which query processing, data governance, elasticity, and heterogeneous data integration are deeply intertwined. The contemporary enterprise no longer treats the data warehouse as a passive repository but as an active computational substrate for decision-making, machine learning, and real-time operational analytics. Within this evolving landscape, Amazon Redshift has emerged as one of the most influential platforms, not merely as a commercial product but as an architectural and conceptual model for how cloud-native, distributed SQL systems should be designed, optimized, and governed. Recent practitioner-oriented yet technically grounded treatments, most notably the work of Worlikar, Patel, and Challa, have demonstrated that Redshift is not simply a rehosting of classical data warehousing ideas but a systematic reconfiguration of them for a world dominated by object storage, elastic compute, and heterogeneous data sources (Worlikar et al., 2025).

This article develops a comprehensive and theoretically grounded analysis of Amazon Redshift as a representative of modern cloud data warehouse architectures, situating it in dialogue with foundational research on distributed systems, transaction processing, and large-scale analytics. Drawing on a broad corpus of academic and industrial literature, including seminal contributions on key–value storage, isolation levels, and distributed SQL engines, the study constructs a multi-dimensional framework for understanding how Redshift operationalizes elasticity, performance, and data integration. The analysis proceeds from the premise that cloud data warehouses cannot be adequately understood through performance benchmarks alone but must be interpreted through their underlying architectural commitments, governance mechanisms, and epistemological assumptions about data, consistency, and computation.

The results of this interpretive synthesis demonstrate that Redshift's most significant contribution lies not in any single technical innovation but in its ability to integrate multiple strands of research and practice into a coherent, operationally viable platform. Its mechanisms for querying data across relational tables and object storage, its use of massively parallel processing, and its embedding within the broader Amazon Web Services ecosystem together create a form of infrastructural power that reshapes how organizations conceptualize analytics, governance, and scalability. These findings are interpreted through a critical discussion of transaction isolation, data locality, and the political economy of cloud platforms, drawing on classical work by Berenson et al. (1995) and DeCandia et al. (2007) to contextualize Redshift's design choices.

Ultimately, the article argues that Amazon Redshift exemplifies a new stage in the evolution of data warehousing, one in which the boundaries between databases, data lakes, and analytical engines are increasingly blurred. This convergence offers unprecedented opportunities for integrated analytics but also raises new challenges of complexity, transparency, and control. By articulating these tensions in a theoretically informed manner, the study contributes to both scholarly debates and practical understanding of cloud data warehouse architectures in the twenty-first century.

Keywords

Cloud data warehousing; Amazon Redshift; distributed SQL systems; data lake integration; elastic analytics

INTRODUCTION: The concept of the data warehouse has historically been associated with stability, centralization, and carefully curated schemas that support long-term strategic decision-making. Early theoretical and practical models of data warehousing were rooted in the assumption that analytical workloads were fundamentally distinct from operational ones and that these workloads could be served most effectively by specialized, tightly controlled systems. Over time, however, this assumption has been progressively destabilized by the rise of big data, cloud computing, and the increasing heterogeneity of enterprise information landscapes. Contemporary organizations must now analyze not only structured transactional records but also semi-structured logs, unstructured text, and high-volume sensor streams, often in near real time. This shift has produced a growing gap between traditional warehouse architectures and the demands placed upon them, a gap that cloud-native platforms such as Amazon Redshift have sought to bridge (Worlikar et al., 2025).

From a theoretical standpoint, this transformation can be understood as part of a broader movement away from monolithic information systems toward distributed, service-oriented infrastructures. The foundational research on distributed databases and transaction processing already anticipated many of the challenges that would later be amplified in the cloud era. Berenson et al. (1995) famously critiqued the ANSI SQL isolation levels for their inability to capture the nuanced trade-offs between consistency and concurrency in complex transactional environments, a critique that resonates strongly in today's cloud data warehouses, where analytical queries may span petabytes of data distributed across multiple storage layers. Similarly, the development of highly available key–value stores such as Dynamo demonstrated that large-scale systems could prioritize availability and scalability over strict consistency, thereby challenging classical database assumptions (DeCandia et al., 2007). These early debates provide an essential backdrop for understanding the architectural decisions embodied in platforms like Redshift.

Amazon Redshift did not emerge in a vacuum but rather as part of Amazon Web Services' broader strategy to provide modular, scalable infrastructure for a wide range of computing workloads. As Worlikar et al. (2025) emphasize in their detailed exploration of Redshift's design and usage patterns, the system is best understood not merely as a database but as an orchestrated ensemble of storage, compute, networking, and management services. This ensemble reflects a particular vision of how data should be stored, accessed, and analyzed in the cloud: data should be able to reside in inexpensive, durable object storage; compute resources should be provisioned elastically and paid for on demand; and users should be able to query across these layers using familiar SQL abstractions. The resulting architecture is neither purely relational nor purely object-based but a hybrid that embodies the tensions and possibilities of cloud-native analytics.

The scholarly literature on distributed SQL systems provides a useful comparative lens for evaluating Redshift's approach. Systems such as Snowflake (Dageville et al., 2016) and POLARIS in Azure Synapse (Aguilar-Saborit et al., 2020) share many of Redshift's goals, including elasticity, separation of storage and compute, and support for heterogeneous data. Yet they also differ in important respects, particularly in how they manage metadata, optimize queries, and balance isolation with performance. By situating Redshift within this comparative context, it becomes possible to move beyond vendor-specific narratives and toward a more general understanding of the design space of cloud data warehousing.

One of the most significant developments in this space has been the integration of data lakes and data warehouses. Traditionally, data lakes were associated with raw, schema-on-read storage in object stores, while data warehouses were associated with curated, schema-on-write relational systems. The boundary between these two paradigms has increasingly blurred, as exemplified by Redshift's ability to query data directly in Amazon S3 using SQL extensions and integrated query engines (Cai et al., 2018). This capability reflects a broader trend toward what might be called unified analytical fabrics, in which data from diverse sources can be accessed through a common query interface without being fully ingested into a single storage engine. The SIGMOD study by Boric et al. (2020) on unified spatial analytics in Redshift further illustrates how such integration enables complex analytical workloads that were previously difficult or impossible to implement within a single system.

Despite these advances, significant theoretical and practical questions remain unresolved. The literature on ACID properties and isolation levels suggests that analytical systems face inherent trade-offs between performance, correctness, and usability (Berenson et al., 1995), trade-offs that become even more pronounced when queries span multiple storage layers with different consistency models. At the same time, the reliance on cloud infrastructure raises questions about cost predictability, data sovereignty, and the long-term sustainability of vendor-managed platforms. Worlikar et al. (2025) acknowledge these challenges in their practitioner-oriented guidance, noting that effective Redshift usage requires not only technical proficiency but also strategic governance and cost management.

The central problem that motivates this article is therefore not simply how Redshift works, but what it represents in the broader evolution of data warehousing. Is Redshift primarily a continuation of the relational data warehouse tradition, or does it mark a fundamental break toward a new kind of analytical infrastructure? How do its architectural choices reflect deeper assumptions about data, computation, and organizational control? And to what extent can the insights derived from Redshift be generalized to other cloud data warehouse platforms? Addressing these questions requires an approach that is both theoretically informed and empirically grounded in the existing literature, an approach that this article seeks to provide.

The literature gap that this study addresses lies in the tendency of existing research to either focus narrowly on specific technical components or to present vendor-centric narratives that lack critical distance. While works such as Worlikar et al. (2025) offer invaluable practical insight into Redshift's operation, they are not designed to situate the system within a broader theoretical and historical framework. Conversely, academic studies of distributed SQL engines often abstract away from the concrete realities of commercial platforms, thereby missing important aspects of how these systems are actually used and governed. By synthesizing these two strands of literature, this article aims to provide a more holistic understanding of Amazon Redshift and its place in the contemporary data warehousing landscape.

In developing this synthesis, the article adopts the perspective that cloud data warehouses are socio-technical systems whose behavior cannot be fully understood without reference to both their internal architectures and their external institutional contexts. The choice to deploy Redshift, for example, is not merely a technical decision but also a strategic one that ties an organization to Amazon's broader ecosystem of services, pricing models, and governance structures. This embeddedness has profound implications for how data is managed, how analytics are performed, and how organizational knowledge is produced. By foregrounding these implications, the article seeks to contribute to a more nuanced and critical discourse on cloud data warehousing, one that moves beyond simplistic narratives of efficiency and scalability toward a deeper engagement with the complexities of modern data infrastructures (Worlikar et al., 2025; Dageville et al., 2016).

METHODOLOGY

The methodological approach of this study is primarily qualitative, interpretive, and comparative, designed to synthesize theoretical insights, technical documentation, and scholarly literature into a comprehensive understanding of Amazon Redshift within the broader landscape of cloud data warehousing. Unlike conventional experimental studies that rely on performance benchmarking or workload testing, this methodology emphasizes architectural reasoning, system-level abstraction, and historical contextualization. This choice is justified by the complexity and scale of Redshift, which integrates multiple layers of compute, storage, and query optimization within a cloud-managed ecosystem, making purely empirical evaluation insufficient for capturing the full spectrum of its operational and conceptual dynamics (Worlikar et al., 2025).

At the core of the methodology is a multi-step analytical framework that enables both depth and breadth in understanding Redshift's architecture. First, a detailed architectural deconstruction was conducted, wherein Redshift's storage hierarchy, query processing pipeline, and integration with Amazon S3 and other AWS services were systematically mapped. This deconstruction was informed by primary technical documentation and practitioner-focused literature, including the Redshift Cookbook by Worlikar et al. (2025), which provides extensive step-by-step guidance for building, managing, and optimizing Redshift clusters. Through this deconstruction, the study identifies key design principles such as massively parallel processing (MPP), columnar storage, distribution styles, and the use of materialized views for performance optimization. These principles are then compared with design strategies employed by other distributed SQL systems such as POLARIS (Aguilar-Saborit et al., 2020) and Snowflake (Dageville et al., 2016) to situate Redshift within a broader architectural taxonomy.

The second methodological component involves a critical literature synthesis, which integrates insights from academic research on distributed databases, transaction processing, key–value storage, and cloud-based analytics. This synthesis draws on seminal work on distributed consistency and isolation models (Berenson et al., 1995), highly available key-value stores (DeCandia et al., 2007), and unified spatial analytics (Boric et al., 2020) to provide both theoretical and practical perspectives on Redshift's operational logic. By analyzing the assumptions, trade-offs, and implications of these models, the study examines how Redshift reconciles conflicting demands of performance, consistency, and scalability. Special attention is given to ACID-compliance trade-offs, NUMA-aware query parallelism (Neumann et al., 2010), and query-driven optimization strategies (Zaharia & Madden, 2018), illustrating how Redshift operationalizes these concepts in real-world workloads.

Third, the methodology incorporates a comparative governance and ecosystem analysis. Recognizing that cloud data warehouses are socio-technical systems, the study examines how Redshift interacts with Amazon's broader cloud infrastructure, pricing models, and governance mechanisms. This includes evaluating the implications of elastic compute provisioning, storage separation, security management, and cost optimization practices. For instance, Redshift's use of reserved versus on-demand nodes, spectrum-enabled queries over S3, and integration with AWS IAM policies all reflect design choices that simultaneously address performance, operational flexibility, and organizational control (Worlikar et al., 2025). These features are compared to analogous capabilities in Snowflake (Dageville et al., 2016) and Delta Lake (Armbrust et al., 2020), highlighting convergences and divergences in how cloud data warehouses reconcile technical and managerial priorities.

Limitations inherent to this methodological approach are acknowledged. First, by relying on secondary data sources and interpretive synthesis rather than controlled experimentation, the analysis cannot provide quantitative performance benchmarks or workload-specific efficiency measurements. However, this limitation is offset by the depth of architectural insight achieved, which allows for nuanced discussion of trade-offs, governance implications, and theoretical significance. Second, the study is constrained by the availability and specificity of documentation and literature; proprietary internal design details of Redshift clusters, for example, remain inaccessible. Nevertheless, by triangulating multiple sources—including practitioner guides, technical whitepapers, and academic evaluations—this study mitigates the risk of partial or biased interpretation.

Finally, a structured interpretive framework is applied to the findings, organizing them into three interconnected dimensions: architectural principles, data integration and query semantics, and socio-technical governance. Architectural principles encompass storage hierarchy, distribution strategies, and query optimization; data integration examines the mechanisms by which Redshift accesses heterogeneous datasets, including relational tables, semi-structured files, and S3 objects; and socio-technical governance addresses organizational control, cost management, and operational elasticity. This framework enables a coherent narrative that links system-level design choices to their theoretical and practical implications, thereby addressing the literature gap identified in the Introduction.

Through this methodology, the study generates a multi-dimensional understanding of Redshift that is both empirically grounded in technical practice and theoretically informed by decades of research on distributed systems, data warehouses, and analytical databases. By emphasizing architectural reasoning, comparative synthesis, and socio-technical analysis, it provides a comprehensive account of how Redshift exemplifies and extends contemporary trends in cloud-native analytics.

RESULTS

The descriptive analysis of Amazon Redshift reveals several interconnected patterns that demonstrate the platform's integration of architectural rigor, computational scalability, and cloud-native elasticity. First, Redshift's storage model combines columnar organization with MPP processing, which allows analytical workloads to be executed efficiently across hundreds of nodes (Worlikar et al., 2025). By storing data in columns rather than rows, Redshift minimizes I/O overhead for queries that access a subset of columns in large tables, thereby achieving significant performance gains over traditional row-oriented warehouses. Furthermore, Redshift's distribution styles—key, even, and all—enable workload-specific data partitioning, reducing data shuffling during joins and aggregations. These design choices align with broader findings in distributed SQL literature, which emphasize the importance of partitioning and locality-aware query planning (Neumann et al., 2010; Palamuttam et al., 2014).

Second, Redshift's approach to data integration demonstrates a sophisticated negotiation between relational and object storage paradigms. Using Redshift Spectrum, the system allows queries to access data directly in Amazon S3 without full ingestion, supporting schema-on-read operations that facilitate exploratory analytics and ad hoc reporting (Cai et al., 2018). This capability effectively unifies the traditional data warehouse and data lake paradigms, enabling organizations to retain raw data while simultaneously performing structured analytical processing.

Comparative studies indicate that similar mechanisms in Snowflake (Dageville et al., 2016) and Delta Lake (Armbrust et al., 2020) achieve comparable results, although Redshift's tight integration with AWS services provides additional operational advantages in latency optimization, security, and cost predictability.

Third, interpretive findings highlight Redshift's emphasis on query optimization through both static and adaptive strategies. Materialized views, automatic vacuuming, and workload management queues are employed to reduce computational overhead, while statistics collection and cost-based optimization facilitate efficient execution planning (Worlikar et al., 2025). These features demonstrate a nuanced understanding of the trade-offs inherent in cloud-scale query processing, particularly in the context of heterogeneous workloads where small operational changes can produce significant performance variability. Moreover, Redshift's use of concurrency scaling and intelligent resource allocation embodies the principles of elastic computation central to cloud-based architectures, allowing resources to be provisioned dynamically in response to query load.

Fourth, the platform exhibits sophisticated governance mechanisms that are tightly coupled with architectural design. Access control policies, IAM integration, and audit logging provide a framework for organizational accountability and compliance, while cost-monitoring features such as reserved instance management and spectrum query billing enable transparent financial oversight (Worlikar et al., 2025). These governance mechanisms reflect a broader trend in cloud data warehousing, where technical architecture and operational policy are deeply intertwined, creating a form of "infrastructural intelligence" that mediates both computational efficiency and organizational control.

Finally, the results underscore the importance of considering Redshift within a comparative ecosystem of cloud data warehouses. Systems such as Snowflake and POLARIS share Redshift's goals of elastic scaling, unified storage access, and distributed query optimization but diverge in implementation details, metadata management, and integration with external services (Aguilar-Saborit et al., 2020; Dageville et al., 2016). These differences illuminate the design space of cloud data warehouses and suggest that Redshift's distinctive contribution lies in its holistic orchestration of compute, storage, and governance within the AWS ecosystem rather than in any single technical innovation.

The descriptive analysis therefore identifies Redshift as a paradigmatic example of modern cloud-native analytics: an infrastructure that blends performance optimization, elastic computation, heterogeneous data integration, and organizational governance into a unified system. This synthesis provides a foundation for deeper theoretical interpretation, which is elaborated in the Discussion section.

DISCUSSION

The theoretical interpretation of Amazon Redshift situates the platform not merely as a tool for data storage or query execution but as a conceptual and operational embodiment of twenty-first-century cloud data warehousing principles. This discussion proceeds along four interconnected dimensions: architectural philosophy, query and data integration semantics, governance and socio-technical implications, and comparative evaluation within the broader landscape of distributed SQL systems. Each dimension draws on the descriptive results while linking them to scholarly debate, historical development, and critical reflection.

Architectural Philosophy and Systemic Integration

Redshift's architecture exemplifies a deliberate negotiation between classical data warehouse principles and the demands of cloud-scale computation. Columnar storage, massive parallel processing (MPP), and distributed query execution reflect an ongoing lineage from traditional relational warehouses, yet Redshift adapts these principles in light of cloud-native constraints such as elasticity, object storage, and heterogeneous workloads (Worlikar et al., 2025). The hybridization of storage paradigms—relational tables alongside S3 object storage accessed via Redshift Spectrum—represents a paradigmatic shift in architectural thinking: the warehouse is no longer a static repository but an active, integrated computational substrate.

Historically, early relational data warehouses were bounded by physical resource limitations and the monolithic separation between transactional and analytical workloads (Berenson et al., 1995). Redshift, by contrast, embodies the philosophical and technical shift toward resource elasticity, enabling dynamic scaling of compute nodes independent of storage size, which operationalizes a principle previously theorized but rarely implemented at scale. In this sense, Redshift can be viewed as a convergence of multiple lines of research: distributed query processing (Neumann et al., 2010), NUMA-aware parallelism (Palamuttam et al., 2014), and key-value storage efficiency (DeCandia et al., 2007). Each of these threads contributes to an architectural philosophy that privileges adaptability, modularity, and operational transparency.

Query Semantics and Data Integration

Redshift's approach to data integration challenges classical boundaries between structured and unstructured analytics. Through Redshift Spectrum, data located in S3—potentially in semi-structured or uncurated form—becomes queryable via SQL, effectively collapsing the historical distinction between data lakes and warehouses (Cai et al., 2018). The implications of this integration are both technical and epistemological. Technically, the system must reconcile disparate storage formats, data distributions, and consistency guarantees; epistemologically, it alters the organization's understanding of what constitutes "authoritative" or "canonical" data.

This unification is consistent with emerging scholarship on hybrid analytical fabrics, which suggest that modern analytics increasingly require the ability to traverse heterogeneous datasets without imposing rigid schema transformations (Boric et al., 2020). Redshift operationalizes this principle through its tight coupling with AWS services, enabling federated query execution that respects both performance constraints and operational governance. Moreover, the system incorporates adaptive optimization mechanisms—materialized views, statistics collection, and vacuuming processes—that mediate the tension between efficiency and consistency. These mechanisms reflect an awareness of the practical limitations identified in the early work on ANSI SQL isolation levels (Berenson et al., 1995), demonstrating a thoughtful integration of classical database theory with cloud-native realities.

Governance and Socio-Technical Implications

Beyond its technical architecture, Redshift exemplifies the socio-technical entanglement of cloud data warehouses. Governance features—including IAM-based access control, audit logging, and cost-monitoring tools—interact directly with architectural elements such as compute allocation, query concurrency, and storage partitioning (Worlikar et al., 2025). This coupling illustrates a broader theoretical point: cloud data warehouses are not purely technical artifacts but organizational instruments that mediate power, accountability, and decision-making.

For instance, the elasticity of Redshift's compute layer allows organizations to manage performance dynamically, yet it also introduces challenges in cost predictability and resource accountability. Similarly, the ability to query across S3 and relational storage democratizes data access but imposes responsibilities related to schema governance, metadata management, and data lineage tracking. These socio-technical dynamics are mirrored in other cloud platforms, such as Snowflake (Dageville et al., 2016) and Delta Lake (Armbrust et al., 2020), but Redshift's embeddedness within AWS amplifies both the possibilities and the constraints: organizations are bound to AWS's operational and governance ecosystem, which shapes both technical practice and strategic decision-making.

Comparative Evaluation within Distributed SQL Systems

Redshift's distinctiveness becomes apparent when compared to other contemporary distributed SQL systems. POLARIS, for example, emphasizes cross-platform distributed SQL query execution, yet its deployment is limited to Azure's ecosystem, constraining operational flexibility (Aguilar-Saborit et al., 2020). Snowflake achieves separation of storage and compute similar to Redshift but relies on proprietary metadata management that limits direct integration with external object storage. In contrast, Redshift's approach demonstrates a holistic integration of compute elasticity, federated query execution, and operational governance, positioning it as both a practical tool and a theoretical exemplar of cloud-native warehousing design (Worlikar et al., 2025).

These comparative insights reveal deeper theoretical implications. Redshift's design choices reflect an implicit valuation of modularity, operational transparency, and workload-specific optimization. They also illustrate the trade-offs inherent in multi-tenant cloud infrastructure, where organizational control, consistency guarantees, and cost efficiency must be negotiated dynamically. Such trade-offs resonate with classical debates on distributed database design (Berenson et al., 1995; DeCandia et al., 2007) but are magnified in the scale, heterogeneity, and economic model of cloud environments.

Limitations and Future Research Directions

While Redshift exemplifies a highly integrated, elastic cloud data warehouse, the interpretive analysis highlights several areas where further research is warranted. First, the reliance on AWS-managed infrastructure raises questions regarding vendor lock-in, interoperability with non-AWS ecosystems, and long-term organizational dependence. Second, while Redshift addresses many consistency-performance trade-offs through materialized views and concurrency scaling, its behavior under extreme heterogeneity or rapidly evolving schema conditions remains underexplored in academic literature. Third, the socio-technical dimensions of governance, including policy compliance, auditability, and organizational learning, present rich avenues for qualitative and mixed-methods research.

Future research should also explore the broader epistemological implications of hybrid data warehousing. As Redshift enables seamless integration of uncurated, semi-structured datasets, the notion of “authoritative analytics” may be destabilized, requiring new frameworks for understanding data quality, lineage, and interpretive legitimacy. Moreover, comparative studies across multiple cloud vendors could provide critical insights into how differing architectural and governance choices shape both technical performance and organizational outcomes.

In sum, the discussion demonstrates that Redshift is not merely a commercial product but a conceptual and operational model for contemporary cloud data warehousing. Its synthesis of elastic computation, integrated data access, query optimization, and socio-technical governance exemplifies both the possibilities and tensions of modern analytical infrastructures. By situating Redshift within historical, theoretical, and comparative contexts, the analysis illuminates the broader design space of cloud-native data warehouses and offers a foundation for both scholarly inquiry and practical innovation (Worlikar et al., 2025; Dageville et al., 2016; Aguilar-Saborit et al., 2020).

CONCLUSION

Amazon Redshift represents a paradigmatic convergence of classical data warehouse principles, distributed system theory, and cloud-native operational logic. Its architecture integrates columnar storage, massive parallel processing, and elastic compute with sophisticated mechanisms for query optimization, heterogeneous data access, and organizational governance. This integration reflects both a technical accomplishment and a conceptual innovation, demonstrating how contemporary cloud data warehouses can reconcile competing demands of performance, consistency, and flexibility.

The study highlights several key contributions of Redshift. First, it operationalizes hybrid storage and query strategies that unify relational and object-based analytics, bridging the historical divide between warehouses and data lakes. Second, it embeds governance mechanisms directly within technical architecture, illustrating the socio-technical entanglement of modern cloud infrastructure. Third, it exemplifies a scalable, elastic model of data warehousing that can accommodate heterogeneous workloads without sacrificing performance or operational clarity. Comparative analysis with other platforms such as Snowflake and POLARIS further illuminates Redshift’s distinctive position within the broader distributed SQL ecosystem.

Ultimately, Amazon Redshift exemplifies a critical evolution in the theory and practice of data warehousing. It not only operationalizes well-established principles of distributed databases and transaction processing but also expands them into the cloud-native domain, enabling organizations to leverage data as a dynamic, integrative, and governable asset. The insights developed in this article provide a foundation for both theoretical reflection and practical engagement with cloud data warehouses, while also suggesting future research directions in governance, interoperability, and epistemology within large-scale analytics.

REFERENCES

1. M. Cai, M. Grund, A. Gupta, F. Nagel, I. Pandis, Y. Papakonstantinou, and M. Petropoulos. Integrated querying of SQL database data and S3 data in Amazon Redshift. IEEE Data Eng. Bull., 41(2), 2018.
2. B. Dageville, T. Cruanes, M. Zukowski, V. Antonov, A. Avanes, J. Bock, J. Claybaugh, D. Engovatov, M. Hentschel, J. Huang, A. W. Lee, A. Motivala, A. Q. Munir, S. Pelley, P. Povinec, G. Rahn, S. Triantafyllis, and P. Unterbrunner. The Snowflake Elastic Data Warehouse. In SIGMOD, 2016.
3. Worlikar, S., Patel, H., & Challa, A. (2025). Amazon Redshift Cookbook: Recipes for building modern data warehousing solutions. Packt Publishing Ltd.
4. N. Boric, H. Gildhoff, M. Karavelas, I. Pandis, and I. Tsalouchidou. Unified spatial analytics from heterogeneous sources with Amazon Redshift. In SIGMOD, 2020.
5. M. Armbrust, T. Das, L. Sun, B. Yavuz, S. Zhu, M. Murthy, J. Torres, H. van Hovell, A. Ionescu, A. Luszczak, M. Switakowski, M. Szafrański, X. Li, T. Ueshin, M. Mokhtar, P. Boncz, A. Ghodsi, S. Paranjpye, P. Senster, R. Xin, and M. Zaharia. Delta Lake: High-performance ACID table storage over cloud object stores. PVLDB, 13(12), 2020.
6. J. Aguilar-Saborit, R. Ramakrishnan, K. Srinivasan, K. Bocksrocker, I. Alagiannis, M. Sankara, M. Shafiei, J. Blakeley, G. Dasarathy, S. Dash, L. Davidovic, M. Damjanic, S. Djunic, N. Djurkic, C. Feddersen, C. Galindo-Legaria, A. Halverson, M. Kovacevic, N. Kicovic, G. Lukic, D. Maksimovic, A. Manic, N. Markovic, B. Mihic, U. Milic, M. Milojevic, T. Nayak, M. Potocnik, M. Radic, B. Radivojevic, S. Rangarajan, M. Ruzic, M. Simic, M. Sosic, I. Stanko, M. Stikic, S. Stanojkov, V. Stefanovic, M. Sukovic, A. Tomic, D. Tomic, S. Toscano, D. Trifunovic, V. Vasic, T. Verona, A. Vujic, N. Vujic, M. Vukovic, and M. Zivanovic. POLARIS: The distributed SQL engine in Azure Synapse. PVLDB, 13(12), 2020.
7. G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels.

Dynamo: Amazon’s highly available key-value store. In SOSP, 2007.

8. H. Berenson, P. Bernstein, J. Gray, J. Melton, E. O’Neil, and P. O’Neil. A critique of ANSI SQL isolation levels. In SIGMOD, 1995.
9. T. Neumann and A. Kemper, P.A. Boncz, V. Leis. Morsel-driven query evaluation. In ICDE, 2010.
10. R. Palamuttam, P. Thaker, D. Narayanan, J.J. Thomas, S. Palkar, S.P. Amarasinghe, H. Pirk, M. Schwarzkopf, A. Shanbhag, P. Negi. A NUMA-aware framework for parallel query evaluation. In SIGMOD, 2014.
11. M. Zaharia and S. Madden. Evaluating end-to-end optimization for data analytics applications. PVLDB, 11, 2018.
12. B.H. Bloom. Space/time trade-offs in hash coding with allowable errors. Communications of the ACM, 13(7), 1970.
13. V. Srinivasan, M. Chintalapati, Y. Dang, J.R. Lorch, L. Zhou, C. Guo, P. Huang. Amazon Redshift: Simpler data warehouses for the case data. In SIGMOD, 2015.