
A Survey of Temporal Databases Real-Time Databases and Data Management Systems

www.surveyx.cn

Abstract

This survey paper provides a comprehensive analysis of temporal databases, real-time databases, and general database systems, emphasizing their crucial role in contemporary data management practices. Temporal databases are designed to handle time-sensitive data changes, enabling operations across various temporal states, which are essential for applications like scheduling and planning. Real-time databases optimize immediate data processing, crucial for applications demanding instantaneous responses, such as financial trading. General databases serve as structured collections of data supporting a wide range of operations, from transaction processing to analytical querying. The paper explores methodologies, technologies, applications, and challenges associated with these databases, highlighting innovations like semantic models for enhancing time-series databases and frameworks like TCAM-SSD for efficient data processing. It examines the integration of large language models into database systems, improving natural language query capabilities. The survey also addresses performance and optimization challenges, data integrity, accessibility, governance, and security, underscoring the importance of effective data management practices. By analyzing case studies and applications, the survey demonstrates the versatility and efficacy of these databases in addressing contemporary challenges. Future research opportunities include optimizing mutation strategies in Query Plan Guidance, expanding frameworks to include additional temporal mechanisms, and integrating database functionalities into blockchain systems. These directions emphasize the need for continuous innovation to ensure the adaptability and effectiveness of database systems in diverse and dynamic applications.

1 Introduction

The rapid proliferation of data in today's digital landscape necessitates robust mechanisms for managing, processing, and transforming information. As organizations increasingly rely on data-driven decision-making, the significance of effective data transformation in database systems has become paramount. This review paper focuses on the critical roles played by temporal and real-time databases, examining their architectures, methodologies, and applications. By synthesizing current research and advancements in the field, this survey aims to provide a comprehensive understanding of the challenges and solutions associated with data transformation in these dynamic environments. The findings will not only highlight the importance of data transformation but also set the stage for future research directions in database technologies.

1.1 Significance of Data Transformation

Data transformation plays a pivotal role in the architecture of temporal and real-time databases, serving as the essential mechanism for adapting and integrating dynamic data to meet the demands of modern applications. Within temporal databases, which are specifically designed to manage time-sensitive data changes, transformation techniques are crucial for accurately representing temporal

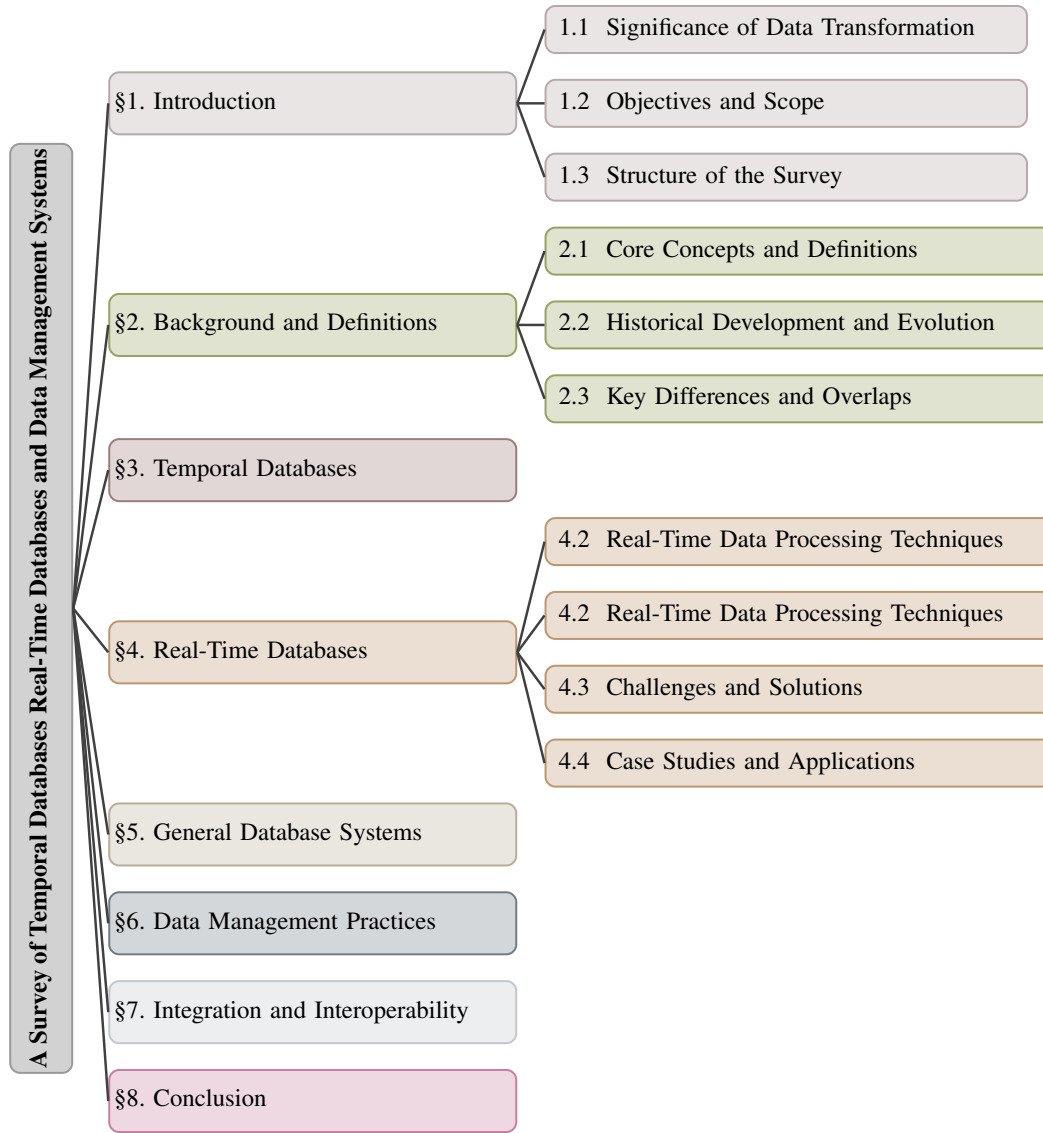


Figure 1: chapter structure

dimensions. This accuracy is vital for ensuring data relevance in swiftly evolving environments, thereby enhancing the overall integrity of the information being processed [1]. Such processes are indispensable for applications requiring structured data under diverse schemas, including planning, scheduling, medical, and fraud detection systems, where timely and accurate data representation can significantly impact decision-making.

In the context of real-time databases, data transformation is indispensable for maintaining dynamic views over rapidly evolving datasets, which is a necessity for applications such as algorithmic trading and scientific research [2]. The capability to transform and process high-velocity multidimensional sensor data, particularly in IoT applications, demands robust transformation mechanisms. These mechanisms must be capable of managing millions of records per second, ensuring that real-time insights are both accurate and actionable [3]. Furthermore, integrating advanced query optimization techniques enhances these capabilities, traditionally reliant on complex heuristics requiring extensive tuning [4].

The integration of large language models into database frameworks showcases the transformative potential of data transformation, improving natural language query capabilities and user interactions through context-aware responses. This evolution reflects a growing trend towards user-centric database design, where the interface and interaction patterns are as critical as the underlying data

structures. Additionally, the configuration knobs of database systems are crucial in achieving high throughput and low latency, underscoring the significance of data transformation in optimizing database performance [5].

Moreover, data transformation is critical for ensuring the scalability of dynamic spatial databases, which face challenges in processing soft real-time spatial queries on extensive sets of moving objects [6]. The SP theory of computing and cognition offers a versatile model for intelligent databases, integrating various concepts in computing and cognitive science to enhance data transformation processes [7]. This interdisciplinary approach not only broadens the scope of data transformation techniques but also emphasizes the need for innovative solutions that cater to the complexities of modern data environments.

The revolution in database architectures emphasizes the integration of algorithms and data with programming languages, highlighting the transformative role of data transformation in modern database systems [8]. The challenge of designing query processing systems that efficiently handle geo-distributed data and user requests further illustrates the importance of flexible data transformation strategies [9]. Furthermore, strategic information systems are highlighted for their role in gaining competitive advantage and improving operational efficiency, reinforcing the importance of effective data transformation [10].

In conclusion, data transformation plays a crucial role in enhancing the efficiency, security, and integrity of both temporal and real-time databases. It supports a diverse array of applications and addresses pressing contemporary challenges in data management, such as the integration of multimedia data, the automation of data access and summarization, and the seamless propagation of changes in transaction data. As database systems evolve to accommodate complex data types and dynamic query optimization, effective data transformation becomes essential for maintaining reliable and secure access to information across the ever-expanding digital landscape [11, 12, 13, 1, 14].

1.2 Objectives and Scope

The primary objective of this survey is to offer a comprehensive analysis of temporal databases, real-time databases, and data management systems, emphasizing their crucial role in contemporary data management practices. This survey aims to explore methodologies, technologies, and applications pertinent to these databases while addressing inherent challenges and solutions in real-time data processing and management. A focal point will be the development of semantic models for enhancing time-series databases (TSDBs), which improve data management and retrieval by adding semantic capabilities [15]. These advancements are essential for ensuring that time-series data can be effectively utilized across various applications, enhancing overall data usability and accessibility.

Additionally, the survey will investigate the potential of novel frameworks like TCAM-SSD, which facilitate search-based computation directly within NAND flash memory arrays, thereby enhancing data processing efficiency [16]. This exploration is particularly relevant in the context of increasing data volumes and the need for faster processing capabilities. The scope extends to examining advanced techniques for managing temporal data exchange, highlighting complexities introduced by the temporal aspect [12]. The integration of temporal data exchange methods with existing systems presents unique challenges that require innovative solutions to ensure data consistency and reliability.

Furthermore, it will delve into challenges of estimating subgraph occurrences and join query selectivity in databases, addressing these through sublinear time algorithms [17]. The exploration of distributed shared-memory databases (DSM-DB) will also be covered, focusing on memory disaggregation to improve performance and scalability [18]. These considerations are vital for enhancing the efficiency of data management systems, particularly in environments characterized by high data throughput and complex query requirements.

In addressing inefficiencies in current systems, the survey will consider methods for detecting performance bugs in database management systems (DBMSs), including new approaches for performance testing [19]. The introduction of Query Plan Guidance (QPG) as a method to enhance test case generation for uncovering logic bugs in DBMSs will be examined [20]. This focus on performance optimization reflects a growing recognition of the importance of maintaining system reliability and efficiency in the face of increasing data complexities. The survey will also address challenges of managing heterogeneous datasets, which include structured, unstructured, and time-series data, by exploring architectures like BigDAWG that aim to overcome limitations of existing systems [21].

Moreover, the paper will explore advancements in database systems such as object-relational integration, web services, and data mining while excluding traditional static models [13]. The survey also aims to propose a new framework that effectively interprets temporal linguistic mechanisms and enhances user interaction with temporal databases [22]. This proposal is indicative of the need for more intuitive and user-friendly database interfaces that can accommodate the complexities of temporal data. Additionally, it will argue for a new approach using deep reinforcement learning (DRL) to create a hands-free optimizer that can automatically tune itself and learn from past optimizations [23].

The survey will propose a novel distributed architecture to overcome limitations in handling soft real-time spatial queries [24] and explore the mutual influence between database technology and blockchain technology, addressing how each has impacted the other's development and identifying features for enhanced functionalities [25]. The scope also focuses on private blockchains, categorizing existing systems based on distributed ledger, cryptography, consensus protocol, and smart contracts [6]. Furthermore, the benchmark for providing a common infrastructure for astronomical data resources will be discussed [7].

The survey aims to demonstrate performance improvements of the SQL MaxBCG algorithm over traditional file-based methods, highlighting the potential of database systems in enhancing data-intensive scientific applications [8]. It also provides an overview of information systems in organizations, addressing their critical role in business operations and decision-making [10]. This comprehensive approach ensures that the survey not only highlights current advancements but also lays the groundwork for future research trajectories in the domain of temporal and real-time databases.

1.3 Structure of the Survey

The survey is meticulously structured to offer an in-depth examination of temporal databases, real-time databases, and data management systems, incorporating a wide range of perspectives and methodologies. This includes the evolution of these systems, their integration with emerging technologies such as artificial intelligence and cloud computing, and the challenges of processing complex data types and user behavior trends [26, 27, 28, 11]. The organization of the paper is designed to facilitate a clear understanding of the complex interactions and innovations within these fields, making it accessible to both researchers and practitioners.

The survey begins with an **Introduction** that establishes the significance of data transformation in temporal and real-time databases, setting the stage for a detailed examination of their roles in modern data management. This is followed by the **Background and Definitions** section, which elucidates core concepts and historical developments, providing a foundational understanding of temporal and real-time databases, as well as general database systems [29]. This foundational knowledge is crucial for grasping the complexities discussed in subsequent sections.

Subsequent sections delve into specific database types: **Temporal Databases** and **Real-Time Databases**, each dissecting methodologies, technologies, applications, and case studies. These sections highlight the unique challenges and solutions pertinent to each database type, supported by literature and case studies [30]. This thematic organization allows for a focused exploration of each database type, ensuring that the reader can appreciate the distinct characteristics and requirements of temporal versus real-time systems.

The survey then transitions to **General Database Systems**, discussing foundational principles, technological advancements, and performance challenges, thereby contextualizing the role of these systems in broader data management practices [31]. This contextualization is essential for understanding how general database principles apply to the more specialized realms of temporal and real-time databases. Following this, an exploration of **Data Management Practices** emphasizes data integrity, accessibility, governance, and security [10], which are critical components of effective data management in any context.

The penultimate section, **Integration and Interoperability**, examines the integration of temporal and real-time databases with general systems, addressing interoperability challenges and solutions, and exploring integration with emerging technologies [14]. This section highlights the importance of ensuring that various database systems can work together seamlessly, which is increasingly important in a landscape where data is often distributed across multiple platforms and technologies.

Finally, the **Conclusion** synthesizes the key findings of the survey, reflecting on the current state and future directions of the field, and identifying potential areas for further research and development [32]. This structured approach ensures a holistic and nuanced understanding of the evolving landscape of database technologies and data management systems, paving the way for future innovations and research endeavors. The following sections are organized as shown in Figure 1.

2 Background and Definitions

2.1 Core Concepts and Definitions

Temporal and real-time databases are critical components of modern data management, each fulfilling distinct roles through specialized functionalities. Temporal databases are designed to handle data with temporal attributes, facilitating queries across past, present, and future states, essential for applications like scheduling and planning [33]. They incorporate interval-annotated nulls to manage incomplete or uncertain data, enhancing data integrity [12]. Incremental maintenance of cyclic association rules optimizes performance by avoiding complete database rescans [34].

Real-time databases prioritize instantaneous data processing, crucial for applications requiring immediate responses such as financial trading and network analytics [2]. They achieve this by maintaining consistent memory snapshots, ensuring clients access the latest information [35]. Efficient streaming of updates to large associative arrays is vital for rapid processing scenarios [36]. They also address event inference under uncertainty, crucial for reasoning about complex event compositions [37].

The SP model offers a versatile framework for intelligent databases, supporting diverse AI functions through a simple representation [38]. The integration of database features into blockchain technology presents challenges in decentralized data management [25]. Retroactive operations highlight the need for selective rollback and replay of queries, emphasizing security in database systems [5].

Operational databases focus on transaction processing, ensuring efficient management of daily operations, while data warehouses facilitate analytical querying for insights from historical data [1]. Hybrid transactional and analytical processing (HTAP) systems enhance memory management, transaction throughput, and data compression. Federated learning systems face challenges in customization and scalability due to tightly-coupled functionalities [21]. Differential privacy systems underscore the importance of managing privacy budgets to prevent breaches when executing multiple queries on sensitive datasets [3].

Ensuring consistent query answering in databases with integrity constraints complicates management [19]. Accurately estimating SQL query cardinality is addressed by effectively communicating uncertainty [4]. Ambiguity in handling tuple deletions when multiple constraints are violated necessitates innovative solutions for interoperability [22].

Temporal and real-time databases are pivotal in enhancing operational efficiency and decision-making, addressing contemporary challenges in database systems [10]. Their evolution incorporates innovations from fields like signal processing [39] and grid computing [8]. They play a significant role in detecting performance changes in software systems amid frequent code modifications [40]. Addressing data discovery, access, and retrieval challenges in large heterogeneous astronomical datasets further underscores their importance [7]. The risk of insider attacks in DBMS emphasizes the need for robust security measures [9].

2.2 Historical Development and Evolution

The historical development of temporal and real-time databases has been crucial in meeting modern data management demands. Initially designed for static data, databases evolved to manage temporal dimensions, enhancing query precision and efficiency [14]. Temporal databases introduced models for time-varying data, enabling retrieval based on specific intervals and conditions. Advancements like interval-annotated nulls bolster robustness in applications requiring temporal accuracy [12]. Standardized query languages have facilitated broader adoption and integration into existing systems.

Real-time databases emerged to meet instantaneous processing needs in critical applications like financial trading [41]. Their evolution includes maintaining consistent memory snapshots for rapid

data access [42]. Innovations in real-time architectures have improved performance metrics, allowing scalability without compromising response times.

Recent advancements include deep reinforcement learning, revolutionizing data processing in complex environments [23]. Realistic datasets from performance tests inspire benchmarks capturing dynamic characteristics [40]. These developments highlight the importance of adapting database technologies to meet contemporary challenges [13]. Methodologies like IMESD emphasize (meta)data integration and iterative development [43]. Research in hardware, software, database management, networking, and cloud computing illustrates comprehensive domain developments [10].

2.3 Key Differences and Overlaps

Understanding the distinctions and overlaps among temporal, real-time, and general databases is vital for appreciating their contributions to data management. Temporal databases manage time-sensitive data, enabling operations across various states, crucial for historical data analysis [1]. They employ sophisticated indexing techniques to handle continuous temporal data efficiently [7].

Real-time databases are optimized for immediate processing, essential for applications demanding instantaneous responses [2]. Their architecture, like DSM-DB systems, improves scalability and performance [10]. General databases, while not focused on time sensitivity, support a wide range of operations, from transaction processing to analytical querying [25]. Integrating machine learning into these systems presents challenges in handling complex queries [7].

Differences in data compaction and error analysis methodologies distinguish these databases. The transition from OLTP to OLAP systems highlights differences in transaction processing and memory usage [1]. Complexity in consistent query answering, especially with cardinality-based repairs, further differentiates these systems [10]. Despite differences, overlaps exist in shared objectives like efficient data management and robust transaction handling. The management of uncertainties and dynamic document storage underscores the collaborative potential of these systems [7]. Exploring these overlaps highlights database technologies' versatility and paves the way for future innovations in data management practices. Tailoring solutions to specific applications ensures effectiveness in diverse environments [10].

3 Temporal Databases

Temporal databases play a pivotal role in managing time-sensitive data, essential for applications requiring historical data analysis and real-time decision-making. They are engineered to store, retrieve, and manipulate data with temporal attributes, supporting efficient handling of dynamic information. This section examines various methodologies and technologies that enhance temporal data management, illustrating their contributions to the robustness and applicability of temporal databases across diverse domains. As depicted in Figure 2, the hierarchical structure of temporal databases categorizes methodologies and technologies alongside applications and case studies. The methodologies section details distributed architectures, AI integrations, framework advancements, and query system enhancements. Meanwhile, the applications section highlights use cases in astronomy, IoT, interoperability, forecasting, and natural language processing, demonstrating the versatility and adaptability of temporal databases across various domains. Table 1 provides a comparative overview of different methodologies and technologies utilized in temporal data management, illustrating their architectural designs, optimization strategies, and primary applications. This visual representation not only complements the textual analysis but also provides a clearer understanding of how these elements interconnect within the broader context of temporal data management.

3.1 Methodologies and Technologies

Temporal data management employs a range of methodologies and technologies designed to efficiently handle time-sensitive information. The Scalable Dynamic Spatial Database (SDSD) exemplifies a distributed architecture for managing moving objects, enabling soft real-time spatial queries crucial for rapid data processing [24]. Advanced indexing techniques within SDSD optimize query performance in dynamic environments. The SP model integrates AI capabilities with traditional database functions, enhancing adaptability to various applications [38]. Incremental maintenance techniques, such as the

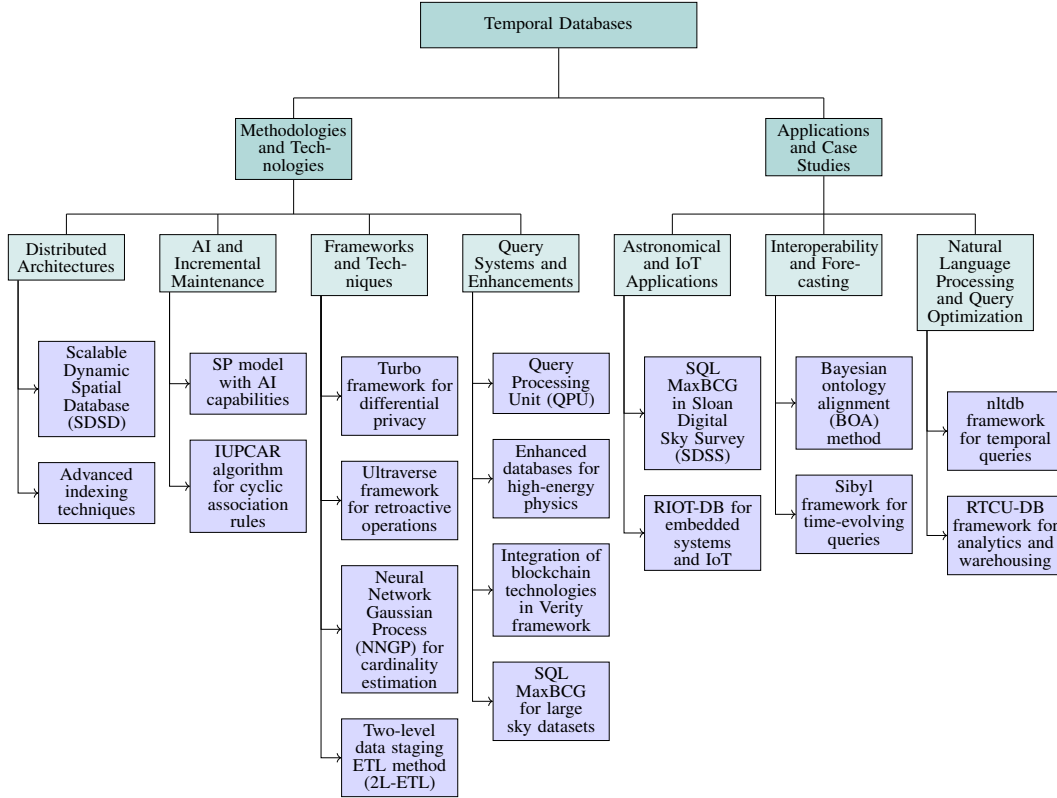


Figure 2: This figure illustrates the hierarchical structure of temporal databases, categorizing methodologies and technologies alongside applications and case studies. The methodologies section details distributed architectures, AI integrations, framework advancements, and query system enhancements, while the applications section highlights use cases in astronomy, IoT, interoperability, forecasting, and natural language processing, demonstrating the versatility and adaptability of temporal databases across various domains.

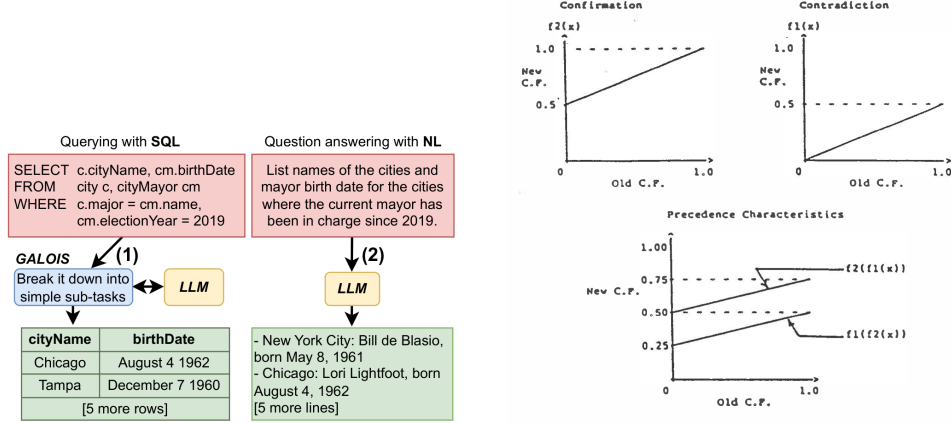
IUPCAR algorithm, facilitate efficient updates of cyclic association rules, minimizing computational overhead [34].

The Turbo framework introduces a caching layer for differential privacy SQL databases, optimizing query responses while maintaining privacy [3]. The Ultraverse framework enhances retroactive operations through dependency analysis, supporting efficient rollback and replay [5]. Neural Network Gaussian Process (NNGP) applies Bayesian deep learning to provide uncertainty-aware cardinality estimation, crucial for managing temporal queries [4]. The two-level data staging ETL method (2L-ETL) streamlines transaction data processing, reducing ETL time [1].

The Query Processing Unit (QPU) offers modular customization for application-specific query systems [44]. Enhanced databases for high-energy physics utilize star-schema structures for complex queries [2]. Integrating blockchain technologies, as in the Verity framework, provides secure and scalable temporal data management [9]. SQL MaxBCG efficiently processes large sky datasets, showcasing SQL’s capability in large-scale temporal data processing [8].

Performance evaluation through application benchmarks and microbenchmark suites for InfluxDB and VictoriaMetrics ensures robustness in handling dynamic workloads [40]. These methodologies collectively enhance temporal databases’ adaptability and effectiveness in managing time-sensitive data across applications.

The methodologies in Figure 4 illustrate the categorization of methodologies and technologies in temporal data management, focusing on distributed architectures, AI integration, and data processing techniques. Each category highlights key approaches and frameworks that enhance the adaptability and efficiency of temporal databases. The comparison between SQL queries and natural language



(a) Querying with SQL and Question Answering with NL: A Comparison of Approaches[45] (b) Update functions for meta-knowledge certainty factor and temporal precedence characteristics[46]

Figure 3: Examples of Methodologies and Technologies

systems highlights the intuitive accessibility of NL techniques, while the graphical representation of update functions emphasizes the linear progression of certainty factors over time. These examples demonstrate the multifaceted nature of temporal databases, combining traditional and innovative approaches to data management [45, 46].

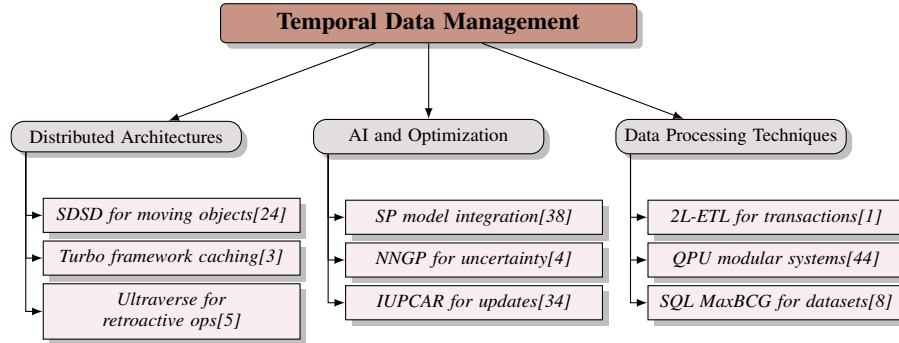


Figure 4: This figure illustrates the categorization of methodologies and technologies in temporal data management, focusing on distributed architectures, AI integration, and data processing techniques. Each category highlights key approaches and frameworks that enhance the adaptability and efficiency of temporal databases.

3.2 Applications and Case Studies

Temporal databases prove effective in a variety of applications, optimizing complex data interactions and managing time-sensitive information. The SQL MaxBCG implementation in the Sloan Digital Sky Survey (SDSS) exemplifies temporal databases' capability in astronomical data analysis, improving performance over file-based approaches [8]. The RIOT-DB system demonstrates efficient I/O management under constrained memory conditions, crucial for embedded systems and IoT applications [47, 27, 11].

The Bayesian ontology alignment (BOA) method facilitates dataset alignment with temporal attributes, enhancing interoperability and data consistency in social media and healthcare records [12, 11]. The Sibyl framework forecasts time-evolving queries, aiding decision-making in dynamic environments like transportation and sales [27, 48, 49, 28, 50].

In natural language processing, the nltdb framework enables intuitive handling of temporal queries, enhancing user interaction and data accessibility [27, 44, 51, 47, 11]. The RTCU-DB framework

integrates Aggregation, GroupBy, and Scan operations, optimizing query performance for analytics and data warehousing [12, 52, 27, 28].

These applications and case studies highlight temporal databases’ adaptability in addressing modern data management challenges, including automating access to complex data types, enhancing ranking mechanisms, facilitating data exchange, and integrating insights from multidisciplinary fields [12, 26, 27, 11]. Their capability to handle intricate data interactions underscores their critical role in facilitating efficient and effective data management across diverse domains.

Feature	Scalable Dynamic Spatial Database	SP Model	Turbo Framework
Architecture Type	Distributed Architecture	AI Integration	Caching Layer
Optimization Technique	Advanced Indexing	Adaptability Enhancement	Privacy Optimization
Primary Application	Moving Objects	Various Applications	Sql Databases

Table 1: The table presents a comparative analysis of three distinct methodologies employed in temporal data management: Scalable Dynamic Spatial Database, SP Model, and Turbo Framework. Each method is evaluated based on its architecture type, optimization technique, and primary application, highlighting their unique contributions to enhancing the efficiency and adaptability of temporal databases. This comparison underscores the diverse approaches in managing time-sensitive data across various domains.

4 Real-Time Databases

Category	Feature	Method
Real-Time Data Processing Techniques	Algorithmic Optimization	STACSS[17]
	Adaptive Processing Techniques	TEE4EHR[37], OT[53], DRL-QO[23], QPU[44]
	Data Integrity Assurance	U[5]
Challenges and Solutions	Adaptive Data Processing	SDSD[24], IUPCAR[34], T[3], 2L-ETL[1]
	Robust Prediction Techniques	NNGP[4]
	Data Integrity Assurance	N/A[9]
Case Studies and Applications	Real-Time Efficiency	Aserv[54], VT[32], SJ-Tree[55], TCAM-SSD[16]
	Privacy and Communication	DPC[56]
	Scheduling Optimization	ViCC[57]

Table 2: This table provides a comprehensive overview of various real-time data processing techniques, categorized into real-time data processing methods, challenges and solutions, and case studies and applications. It highlights key features and methods, referencing specific frameworks and systems that address the complexities of modern data environments. The table serves as a valuable resource for understanding the advancements and applications of real-time databases in diverse fields.

The increasing demand for immediate data processing capabilities has led to significant advancements in the field of real-time databases. These systems are designed to handle dynamic data environments where timely decision-making is crucial. To understand the mechanisms that enable such efficiency, it is essential to explore various real-time data processing techniques that have emerged. Table 2 presents a detailed classification of real-time data processing techniques, challenges, and solutions, as well as case studies and applications, underscoring the innovative methodologies employed in modern data management systems. This exploration will delve into the cutting-edge frameworks and methodologies that enhance the efficiency of rapid data handling, specifically focusing on how real-time databases are being optimized to tackle the complexities and demands of modern applications. By examining the evolution of data management systems, which have transitioned from automating basic transaction recording to managing diverse multimedia content and employing advanced techniques like automated data exploration, we will uncover how these innovations are shaping the future of data accessibility and usability in various fields, including healthcare, cloud computing, and artificial intelligence. [26, 47, 11]

4.1 Real-Time Data Processing Techniques

In the realm of real-time databases and data processing, understanding the evolution and structure of data management techniques is crucial. As shown in Figure 6, this figure illustrates the hierarchical classification of real-time data processing techniques, adaptive systems, and associated challenges and solutions, highlighting key methodologies and innovations in each category. The provided examples

Method Name	Data Processing Efficiency	Adaptability	Application Domains
TCAM-SSD[16]	Rapid Searching	Changing Conditions	Database Analytics
TEE4EHR[37]	Rapidly And Accurately	Dynamically Adjust Conditions	Healthcare Finance Network
STACSS[17]	Efficient Estimation	Dynamic Adjustment	Databases, Graphs
OT[53]	High Throughput	Dynamic Environments	Database Systems
DRL-QO[23]	Rapidly And Accurately	Adaptively Learn	Database Management Systems
QPU[44]	Query Response Time	Flexible Configuration	In-game Advertisement
U[5]	Significant Speedups	Dependency Analysis	Web Applications

Table 3: This table compares various real-time data processing methods, highlighting their efficiency in data processing, adaptability to changing conditions, and application domains. It provides insights into the strengths and applicability of each method in sectors such as healthcare, database systems, and web applications, underscoring their role in enhancing real-time data processing capabilities.

offer a comprehensive overview of both the intricate data structures used in real-time processing and the historical development of data management methods.

The first image illustrates a sophisticated data structure comprising a section index, attribute bitmap, data archive, and position index, which collectively enable efficient data retrieval and processing. Each component serves a specific function: the section index uses hexadecimal numbers to identify sections within the data structure, while the attribute bitmap employs binary representation to indicate the activity status of each section. This setup is instrumental in optimizing real-time data processing tasks.

The second image presents a timeline of the six generations of data management, charting the journey from rudimentary manual methods, such as paper and pencil, to advanced automated systems. This evolution highlights the significant technological advancements that have transformed data handling, underscoring the shift towards real-time processing capabilities that modern databases demand. Together, these examples provide valuable insights into the mechanisms and historical context of real-time data processing techniques [58, 11].

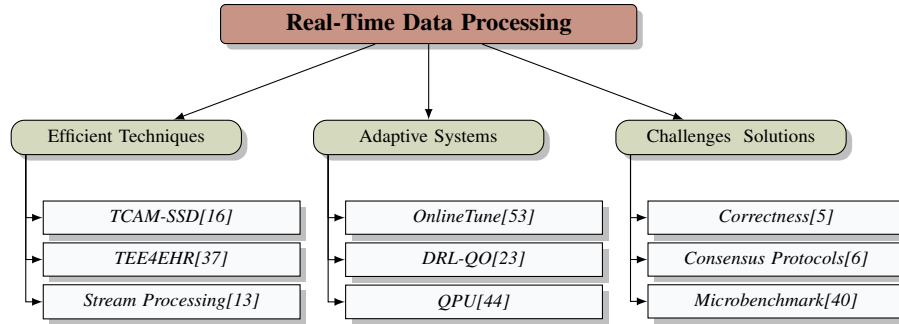


Figure 5: This figure illustrates the hierarchical classification of real-time data processing techniques, adaptive systems, and associated challenges and solutions, highlighting key methodologies and innovations in each category.

4.2 Real-Time Data Processing Techniques

Method Name	Data Processing Efficiency	Adaptability	Application Domains
TCAM-SSD[16]	Rapid Searching	Changing Conditions	Database Analytics
TEE4EHR[37]	Rapidly And Accurately	Dynamically Adjust Conditions	Healthcare Finance Network
STACSS[17]	Efficient Estimation	Dynamic Adjustment	Databases, Graphs
OT[53]	High Throughput	Dynamic Environments	Database Systems
DRL-QO[23]	Rapidly And Accurately	Adaptively Learn	Database Management Systems
QPU[44]	Query Response Time	Flexible Configuration	In-game Advertisement
U[5]	Significant Speedups	Dependency Analysis	Web Applications

Table 4: This table compares various real-time data processing methods, highlighting their efficiency in data processing, adaptability to changing conditions, and application domains. It provides insights into the strengths and applicability of each method in sectors such as healthcare, database systems, and web applications, underscoring their role in enhancing real-time data processing capabilities.

Real-time databases are engineered to facilitate immediate data processing, ensuring rapid responses in dynamic environments. Techniques such as the TCAM-SSD framework have been pivotal in achieving efficient in-storage computation by modifying SSD firmware and utilizing a specialized command interface for search operations, which are crucial for real-time data processing [16]. This framework enhances data access efficiency, which is vital for applications demanding swift data retrieval and processing.

The integration of transformer-based event encoders, exemplified by TEE4EHR, improves representation learning from irregularly sampled time series by employing a deep attention module, thereby enhancing the accuracy and efficiency of real-time data processing [37]. This methodology is critical in sectors such as healthcare, where timely and precise data processing can significantly influence decision-making outcomes.

Stream processing serves as a fundamental technique for managing real-time data streams, enabling immediate data processing and response [13]. This approach is particularly beneficial for applications requiring continuous data input and output, such as financial trading and network monitoring systems. Moreover, the use of sublinear time algorithms for counting subgraph occurrences, as proposed by Aliakbarpour et al., allows for efficient approximation of complex structures in real-time databases by sampling edges and vertices based on their degrees [17]. This technique is vital for maintaining high performance in applications with large and complex datasets.

OnlineTune introduces an online tuning system that interacts directly with live databases to adapt to changing workloads without necessitating copied instances, thereby optimizing real-time data processing [53]. This system ensures that databases dynamically adjust to workload variations, maintaining optimal performance. The development of a DRL-based query optimizer, as proposed by Marcus et al., enables autonomous learning of optimal execution strategies based on past query performance, further enhancing real-time data processing capabilities [23]. This approach allows real-time databases to continuously improve their query execution efficiency, adapting to evolving data environments.

The Query Processing Unit (QPU) provides a modular framework for creating application-specific query systems, allowing customization of real-time data processing to meet specific application needs [44]. The evaluation conducted using a case study of an in-game advertisement service demonstrated the performance of different QPU configurations in terms of query response time and result freshness, highlighting the adaptability of real-time databases.

Real-time data processing also involves ensuring correctness in application semantics during retroactive operations, which is crucial for maintaining data integrity [5]. Challenges such as inefficiencies in consensus protocols and high overheads in transaction processing, as discussed in blockchain applications, further illustrate the complexities involved in real-time data management [6].

Finally, the optimized microbenchmark suite developed by Grambow et al. offers faster execution times and reliable performance feedback for frequent code changes, providing essential benchmarks for real-time database systems [40]. These techniques collectively highlight the advancements in real-time data processing, emphasizing the need for innovative solutions to ensure high performance and adaptability in real-time database environments.

As illustrated in Figure 6, the hierarchical classification of real-time data processing techniques, adaptive systems, and associated challenges and solutions underscores the key methodologies and innovations within each category, providing a visual representation that complements the textual analysis presented. Table 4 presents a comparative analysis of different real-time data processing techniques, emphasizing their efficiency, adaptability, and application domains in dynamic environments.

4.3 Challenges and Solutions

Real-time data management presents a myriad of challenges that necessitate innovative solutions to ensure efficient and accurate data processing. A primary challenge is the unpredictability of transaction latencies, significantly impacting systems that require consistent and timely data processing, such as financial trading platforms and network monitoring systems [42]. The reactor model addresses this issue by enhancing control over transaction latencies, improving scalability, and allowing for the expression of complex application logic in a clear manner.

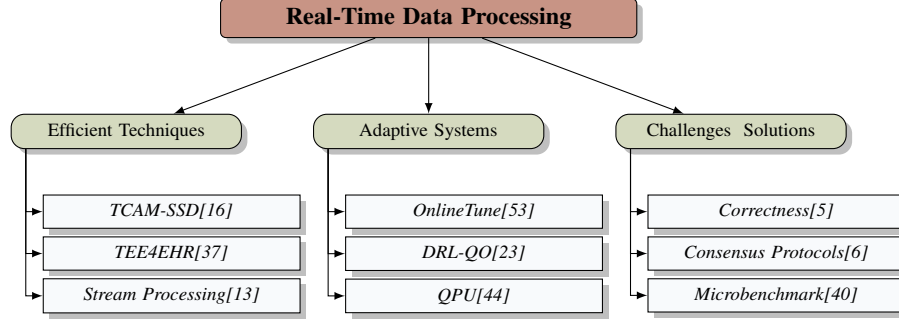


Figure 6: This figure illustrates the hierarchical classification of real-time data processing techniques, adaptive systems, and associated challenges and solutions, highlighting key methodologies and innovations in each category.

Concurrency control and memory management within Distributed Shared Memory Database (DSMDB) architectures pose significant challenges, particularly in the absence of hardware cache coherence. This limitation complicates efficient data retrieval and processing, necessitating advanced techniques to manage memory and ensure data consistency across distributed nodes [18]. The complexity of database interactions and the need for real-time data retrieval further exacerbate these challenges, demanding user-friendly interfaces for operators to maintain system usability [42].

The integration of complex data types and the necessity for dynamic query optimization are critical challenges in real-time data management. These challenges require systems to adapt to varying data structures and optimize query execution dynamically, ensuring efficient data processing across diverse applications [13]. The centralized nature of existing algorithms struggles to manage the resource demands of high-frequency updates and queries for massive datasets, highlighting the need for distributed approaches that can scale effectively [24].

Incremental maintenance of cyclic association rules presents a significant computational cost, as rescanning large databases whenever new transactions are added is inefficient for real-time applications [34]. Solutions like Turbo offer efficient caching of query results, reducing privacy budget consumption and providing accurate responses to a larger number of queries in differential privacy systems [3]. This capability is crucial for maintaining data privacy and efficiency in real-time environments.

Traditional ETL processes face challenges due to the sequential nature of jobs, particularly when processing early-arriving data. These inefficiencies necessitate innovative ETL strategies that can handle data dynamically, improving data processing speed and accuracy [1]. The Neural Network Gaussian Process (NNGP) offers solutions with uncertainty-aware predictions, faster training times, and robustness to varying query workloads, addressing the need for accurate and adaptable data processing methods [4].

Blockchain systems, while promising, face scalability, efficiency, and usability challenges, indicating they are not yet ready for widespread adoption in data-intensive applications [6]. Furthermore, existing methods like access controls and logging are insufficient against insider threats, as attackers can manipulate logs to cover their tracks, necessitating more robust security measures [9].

To effectively tackle the challenges of high-performance data processing across a wide range of applications, it is essential to foster continuous innovation and develop adaptable solutions for real-time data management. This includes automating access to diverse data formats such as images and videos, implementing hybrid transactional and analytical processing systems that optimize performance without compromising data integrity, and designing application-specific query processing architectures that can be tailored to meet the unique requirements of various use cases. These advancements underscore the critical need for flexible and efficient data management strategies in an era characterized by rapidly increasing data volumes and the demand for real-time analysis. [26, 44, 59, 60, 11]

4.4 Case Studies and Applications

Real-time databases have been instrumental across various industries, providing critical solutions for immediate data processing and timely decision-making. The Aserv system exemplifies the

application of real-time databases in astronomical sky surveys, where its low latency performance and efficient data handling capabilities enable the provision of timely alerts for scientific events [54]. This application is particularly valuable in fast sky survey projects, where rapid data processing is essential for capturing transient phenomena.

The DBToaster system showcases real-time database applications in algorithmic trading and TPC-H benchmark queries, demonstrating its ability to handle high update rates through viewlet transforms [32]. This capability is crucial for financial applications requiring instantaneous data updates and analysis to inform trading strategies.

In decentralized communication systems, the DPCrowd framework has been evaluated against existing methods such as FAST and RescueDP, highlighting its accuracy and communication efficiency in handling real-time data from synthetic and real-world datasets [56]. This framework is vital for applications that necessitate privacy-preserving and efficient data communication across distributed networks.

ViCC's schedulers have demonstrated superior performance over traditional SI methods, making it a practical solution for modern distributed database systems [57]. This advancement is significant for industries relying on distributed databases to manage complex data interactions and maintain consistency across nodes.

The application of real-time databases in social media analytics is exemplified by experiments conducted on datasets from New York Times articles and Tencent Weibo, which showed significant improvements in processing time for continuous queries [55]. This capability is essential for applications requiring real-time analysis of large volumes of social media data to derive insights and trends.

Polynesia has demonstrated substantial improvements in transactional and analytical throughput, achieving increases of 1.70X and 3.74X respectively, showcasing its application in real-time analytics [61]. This system is particularly beneficial for industries that require efficient handling of both transactional and analytical workloads in real-time environments.

In-memory database engines like DBToaster have been tested in multi-tenant environments to assess performance, measuring query latencies for various queries [62]. This experimentation underscores the importance of real-time databases in managing concurrent queries efficiently, especially in environments with diverse tenant requirements.

The TCAM-SSD framework has been evaluated across transactional databases, database analytics, and graph analytics, demonstrating superior performance compared to conventional SSDs [16]. This framework is crucial for applications that demand high-speed data retrieval and processing, such as financial services and scientific research.

Significant discrepancies in tail latencies, as observed in experiments involving different JVMs and garbage collectors, illustrate the impact of benchmarking choices on performance measurements in real-time databases [63]. Understanding these discrepancies is vital for optimizing real-time database performance in various industrial applications.

The Vsystem's real-time database has been compared with relational databases, highlighting their complementary roles in enhancing operational efficiency [42]. This comparison emphasizes the importance of integrating real-time databases with traditional systems to optimize data management and processing across industries.

5 General Database Systems

Understanding the foundational principles and structures of general database systems is crucial for robust data management, influencing design, implementation, and operational efficiency. This section explores these core principles and structures, setting the stage for discussing technological advancements that enhance modern database systems' capabilities and performance.

5.1 Foundational Principles and Structures

Foundational principles in general database systems ensure robust data management, efficient processing, and secure storage across diverse applications. Central to these systems is the optimization

of query execution plans, essential for minimizing performance issues and ensuring responsive systems. Inefficient plans can significantly slow down execution, necessitating advanced algorithms that analyze query patterns and data distributions to maximize resource utilization [64]. Optimizing operating system configurations to minimize interference and ensure deterministic performance is vital, involving resource allocation, process scheduling, and memory management [62].

Security integration within these systems is paramount for data integrity and protection. Systems like Verity enhance security by integrating with existing SQL databases without altering their structure or query language, maintaining usability while strengthening security frameworks [9]. Encryption, access controls, and auditing mechanisms further secure data against threats. Modern database architectures often adopt a layered approach, integrating components like data storage, query processing, and transaction management to enhance functionality. This structure supports features such as object-relational capabilities and real-time data processing, facilitating efficient management of complex operations [1, 44, 13].

These foundational principles focus on optimizing performance through dynamic query strategies, ensuring robust security, and maintaining flexibility for diverse data management needs, including multimedia data integration and support for machine learning applications. As these systems evolve, they increasingly enable rapid, reliable, and secure access to distributed information, supporting a wide range of applications from transactional processing to analytical querying [13, 11].

5.2 Advancements in Database Technologies

Recent advancements in database technologies have transformed data storage and retrieval, enhancing efficiency, scalability, and reliability. As illustrated in Figure 7, these advancements focus on critical areas such as query optimization, persistent memory integration, and modern data management strategies. The figure highlights key techniques and frameworks that drive innovation in these domains, emphasizing the pivotal role of machine learning, in-storage computation, and automated data processes. Despite extensive research, query optimization remains challenging due to the lack of ground truth for execution times, necessitating continuous innovation in optimization techniques [64]. Researchers are exploring machine learning and statistical modeling to improve query performance predictions.

The integration of persistent memory into database systems offers improved bandwidth and latency, optimizing database operations [65]. This advancement is crucial for real-time processing applications demanding high throughput and low latency. Frameworks like the TCAM-SSD enable efficient in-storage computation, reducing data retrieval times and paving the way for new applications [16]. Deep reinforcement learning-based query optimizers further enhance adaptability, learning from past executions to optimize future performance [23].

These technological advancements are essential for addressing modern data management challenges, integrating diverse data types, and automating data access and summarization processes. They leverage multidisciplinary research areas, including ontology management and AI, to improve user interaction with dynamic data environments [26, 14, 11].

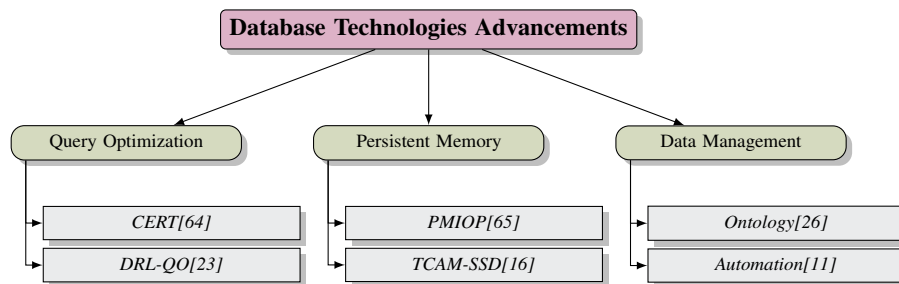


Figure 7: This figure illustrates the recent advancements in database technologies, focusing on query optimization, persistent memory integration, and modern data management strategies. It highlights key techniques and frameworks that drive innovation in these areas, emphasizing the role of machine learning, in-storage computation, and automated data processes.

5.3 Performance and Optimization Challenges

Performance and optimization challenges significantly impact the efficiency and reliability of general database systems. Technologies like instant restore address the need for immediate data access during recovery, minimizing downtime and ensuring operational continuity [66]. Optimizing query execution plans remains complex, with inefficient plans causing bottlenecks. Advanced optimization techniques, such as adaptive and cost-based optimization, are crucial for enhancing system responsiveness [64].

Integrating persistent memory presents opportunities and challenges, requiring careful implementation of I/O primitives to maximize benefits [65]. Dynamic tuning of system configurations is necessary to minimize interference and ensure deterministic performance, highlighting the need for robust optimization frameworks [62]. These frameworks must incorporate predictive analytics and machine learning to adapt to workload changes proactively.

Addressing these challenges is vital for the continued effectiveness of database systems, supporting a wide range of applications. This evolution involves integrating multidisciplinary research, enhancing query processing capabilities, and adapting to architectural changes, including machine learning and intelligent query optimization strategies [26, 44, 13]. Ongoing innovation in these areas is crucial for overcoming performance and optimization challenges, ensuring the reliability and efficiency of general database systems in the future.

6 Data Management Practices

Effective data management practices are essential for ensuring the integrity and accessibility of information in modern systems. The rapid increase in data across sectors necessitates robust frameworks and methodologies for efficient management. This section explores foundational aspects of data management, focusing on data integrity and accessibility, governance, security, and innovations, providing a comprehensive analysis of strategies critical to contemporary data environments.

6.1 Data Integrity and Accessibility

Ensuring data integrity and accessibility is fundamental to effective data management, maintaining accuracy, consistency, and availability across applications. SQL MaxBCG demonstrates efficient data management by leveraging existing database technologies for data analysis [8]. Integrity constraints enforce rules to prevent anomalies, ensuring reliable data. The VLKB integrates diverse data sources, supporting scientific tasks with reliable data and standardization protocols [7]. Automated query generation frameworks like AMOEBA identify performance issues affecting data integrity [40]. Proactive database monitoring ensures optimal data integrity by addressing deviations promptly [20]. Web-based enterprise systems enhance data management efficiency, reducing implementation time and costs [43]. Dynamic spatial databases distribute loads across servers for real-time updates while maintaining consistency [24]. Two-level data staging ensures referential integrity, crucial in transaction processing [1]. Practices like Query Plan Guidance (QPG) and hyper-parameter optimization enhance performance and user experience, facilitating complex data pattern discovery with minimal effort [67, 47, 20, 68]. These strategies are vital for supporting diverse and dynamic applications, safeguarding critical information's accuracy and accessibility.

6.2 Data Governance and Security

Data governance and security are essential for ensuring integrity, compliance, and protection of sensitive information in modern database systems. Robust governance frameworks optimize resource utilization and empower user control [8]. Verity exemplifies data governance advancements by verifying integrity without storing actual data, maintaining privacy while detecting tampering [9]. Privacy-preserving techniques like DPCrowd enable real-time collaborative statistical estimation without sharing raw data, balancing privacy and utility [56, 69]. Despite advancements, challenges in privacy, security, and data organization persist, necessitating continuous development to accommodate richer data types and multidisciplinary approaches [26, 14, 11]. Technologies like blockchain enhance database functionalities, addressing vulnerabilities through innovative frameworks that maintain data integrity without full migrations [9, 25, 35, 11]. These practices are integral to ethical and efficient data management, ensuring systems meet contemporary data-driven demands.

6.3 Innovations in Data Management

Innovative strategies and tools in data management enhance efficiency, scalability, and adaptability. Flexible operator embeddings reduce manual feature engineering, allowing autonomous adaptation to diverse data landscapes [70]. Large language models like BERT and GPT-3 optimize complex interactions and improve decision-making [71, 45]. Incremental high average utility itemset mining enhances algorithm efficiency, crucial for managing large transactional data [72]. Future research will refine data distribution strategies and optimize query performance [73, 74]. Multi-objective optimization approaches maintain personalization while controlling output size, integrating flexible skylines with emerging techniques [75, 3]. Nemesyst improves predictive performance, emphasizing effective real-time data handling [76]. CFICA processes incremental data efficiently, supporting real-time updates [77]. Future research will focus on optimizing algorithms for larger datasets and enhancing online control of false discovery rates [78, 79]. Innovations in buffer management, index design, and approximate reasoning are critical for advancing data management strategies [18, 13]. Incremental learning techniques offer paths for further research, enhancing SP model learning capabilities and user interfaces [23, 38]. Incremental maintenance of cyclic association rules benefits real-time analysis and decision-making [34]. Future research will develop cost model-based approaches for query processing and expand datasets to include diverse types and sources [44, 2].

7 Integration and Interoperability

Integration and interoperability are essential in database technologies, enabling diverse systems to function cohesively. With growing data complexity, understanding integration frameworks and models is crucial for facilitating seamless data interactions. These frameworks support data-driven decision-making by efficiently managing and integrating data from multiple sources. This section examines integration frameworks and models, highlighting their roles in enhancing interoperability and managing complex data exchanges.

7.1 Integration Frameworks and Models

Sophisticated frameworks and models are key to integrating diverse database technologies, ensuring seamless interoperability. Geerts et al. provide a taxonomy of spatio-temporal queries and transformations essential for effective integration across database technologies [80]. This taxonomy aids in designing systems that accommodate the dynamic nature of spatio-temporal data. Integration models often use first-order logic for computational completeness, enabling complex queries and transformations necessary for synthesizing data from diverse sources. Multiple Viewpoint (MVP) models enhance understanding of complex processes by relating events and activities through objects and classes, facilitating process mining techniques across various viewpoints [26, 81, 14, 1].

Integration frameworks must adapt to emerging technologies, such as machine learning and blockchain, within existing database systems. These frameworks evolve with multimedia database access, artificial intelligence integration, and self-managing architectures, supporting diverse applications and workloads [26, 11, 44, 13, 14]. Innovations like association rule mining integration facilitate ad hoc data mining queries, enhancing user control and flexibility. Multidisciplinary research converges database management with fields like artificial intelligence and natural language processing, enriching database capabilities. DB-GPT exemplifies integrating large language models with databases, transforming user interactions through context-aware responses and complex query generation [26, 81, 82, 68, 14]. These frameworks are crucial for supporting complex data interactions in contemporary applications.

7.2 Interoperability Challenges and Solutions

Interoperability among database systems faces challenges requiring innovative solutions for seamless data exchange across platforms. Data format heterogeneity complicates integration, necessitating sophisticated translation and mapping mechanisms [7]. Spatio-temporal data management adds complexity, demanding robust frameworks for dynamic environments [80]. Inconsistent query semantics and execution strategies across systems lead to discrepancies, requiring standardized languages and protocols for uniform query execution [53]. Machine learning integration introduces additional challenges, demanding adaptable mechanisms for complex queries and structures [71].

Middleware technologies serve as intermediaries, providing translation and mapping services for consistent data exchange [18]. Flexible data models and schemas enhance interoperability by accommodating varying data formats [73]. Decentralized architectures like blockchain offer secure frameworks for data exchange across distributed networks, ensuring data security and integrity [9]. Continuous innovation is essential for developing adaptable solutions to manage complex data exchanges, facilitating advanced exploration and enhancing contemporary applications through automated access and multidisciplinary integration [26, 11, 47, 68, 14].

7.3 Integration with Emerging Technologies

Integrating emerging technologies into database systems transforms data management capabilities, particularly through machine learning techniques. Flexible Operator Embeddings (FOE) automatically transform query operators into feature vectors, optimizing data interactions and decision-making [70]. Large language models like BERT, GPT-3, and Codex automate and enhance data management tasks, continuously adapting to evolving environments for efficient data management [71]. Blockchain technologies offer secure frameworks for scalable data management across distributed networks, ensuring data protection and compliance [9].

These integrations enhance database systems' effectiveness and adaptability, supporting efficient management across dynamic applications. They combine traditional queries with advanced mining techniques, facilitating meaningful pattern extraction from complex datasets. Integration supports diverse data types, including multimedia content, and addresses challenges in semantics and ontology management, optimizing interactions and providing users with improved access to varied data sources in an interconnected digital landscape [26, 68, 11].

8 Conclusion

8.1 Future Directions and Research Opportunities

The dynamic evolution of database technologies opens a myriad of avenues for advancing data management practices and system functionalities. As data volume and complexity continue to escalate, there is a growing imperative for sophisticated approaches that enhance database performance. A promising avenue involves refining mutation strategies within Query Plan Guidance (QPG) to boost query execution efficiency. Additionally, extending QPG's application beyond conventional Database Management System (DBMS) testing could elevate test suite quality, ensuring robust system performance and reliability. This advancement not only fortifies database systems but also spurs innovation in testing methodologies applicable across diverse domains.

Improving the BigDAWG polystore architecture should focus on expanding query planning capabilities, crucial for effectively managing heterogeneous data sources. Broadening the range of supported database engines and applying the system to varied datasets can enhance interoperability and scalability across platforms. This development is particularly pertinent in the context of big data, where seamless integration of multiple data sources can yield more comprehensive analyses. Addressing data heterogeneity challenges and ensuring efficient query processing are vital for polystore systems' future success.

Empirical assessments of Distributed Shared Memory Database (DSM-DB) performance and advancements in concurrency control mechanisms present another promising research area. Enhancing data retrieval and processing is essential for boosting the scalability and efficiency of distributed systems. As these systems become more widespread, understanding concurrency control intricacies and their performance impact is crucial. Additionally, improving user interfaces, developing context-sensitive help systems, and expanding external network access while maintaining security are key to enhancing database usability and integration. These improvements will foster user engagement and ensure broader accessibility, promoting system adoption.

Future research should aim to broaden frameworks to incorporate additional temporal mechanisms, significantly enhancing temporal databases' functionality. Strengthening integration with commercial temporal database systems can improve adaptability and functionality in dynamic settings. This is particularly relevant as organizations increasingly depend on time-sensitive data for decision-making. Furthermore, integrating OnlineTune with offline tuning methods to enhance configuration discovery and adaptation speed offers another fruitful research avenue. This optimization is crucial for

ensuring high throughput and low latency in database operations, critical for maintaining performance standards in modern applications.

Implementing and testing Scalable Dynamic Spatial Database (SDSD) algorithms, exploring GPU-based computations for performance gains, and refining load balancing strategies are vital research areas. These efforts will improve spatial data processing, ensuring efficient management of dynamic workloads. The SP model provides a versatile alternative to existing systems, integrating various functionalities while supporting intelligent processing. This adaptability can significantly enhance data management and optimize complex interactions, paving the way for innovative applications across fields.

Improving the quality of extracted cyclic association rules, exploring adaptive support thresholds during incremental updates, and leveraging vertical database representations to boost IUPCAR's performance are critical research areas. These advancements will refine data processing techniques, leading to more accurate and efficient analyses in transactional environments. Future research should also focus on optimizing staging databases' storage and retrieval mechanisms, essential for efficient data processing workflows. Integrating proposed ETL methods with real-time frameworks can further enhance system responsiveness, meeting contemporary data environments' demands.

Incorporating database functionalities into blockchain systems could improve performance and security, identifying suitable databases for specific blockchain applications. This integration is particularly significant as blockchain-database intersections gain traction across sectors. Optimizing consensus mechanisms, exploring sharding techniques, and developing high-level declarative languages for smart contracts are crucial for enhancing performance and usability in blockchain applications. These advancements can lead to more efficient transaction processing and improved scalability, addressing key challenges faced by blockchain systems.

The benchmark highlights potential improvements in data organization and retrieval in High Energy Physics, paving the way for better data management and analysis. Future research should explore further optimizations of microbenchmark suites and investigate strategies to reduce false positives in performance change detection. Enhancing interoperability with other systems and expanding VLKB capabilities to accommodate evolving scientific requirements will be crucial for advancing database technologies. Additionally, integrating database technologies in Grid applications and developing robust systems for managing large astronomical datasets represent significant future directions. These initiatives will not only improve data accessibility but also facilitate collaborative research across scientific domains.

Exploring these research opportunities underscores the importance of continuous innovation in database technologies. By addressing modern data-centric environments' evolving demands, researchers can ensure database systems remain adaptable and effective. This ongoing commitment to research and development will ultimately lead to more robust and efficient database solutions, capable of meeting the challenges posed by the increasing complexity of data management in today's digital landscape.

References

- [1] Xiufeng Liu. Two-level data staging etl for transaction data, 2014.
- [2] Andrea Ceccarelli, Andrea Cioni, Maria Vittoria Garzelli, Piergiulio Lenzi, and Laura Redapi. Towards enhanced databases for high energy physics, 2019.
- [3] Kelly Kostopoulou, Pierre Tholoniati, Asaf Cidon, Roxana Geambasu, and Mathias Lécuyer. Turbo: Effective caching in differentially-private databases, 2023.
- [4] Kangfei Zhao, Jeffrey Xu Yu, Zongyan He, and Hao Zhang. Uncertainty-aware cardinality estimation by neural network gaussian process, 2021.
- [5] Ronny Ko, Chuan Xiao, Makoto Onizuka, Yihe Huang, and Zhiqiang Lin. Ultraverse: A system-centric framework for efficient what-if analysis for database-intensive web applications, 2025.
- [6] Tien Tuan Anh Dinh, Rui Liu, Meihui Zhang, Gang Chen, Beng Chin Ooi, and Ji Wang. Untangling blockchain: A data processing view of blockchain systems, 2017.
- [7] Marco Molinaro, Robert Butora, Marilena Bandieramonte, Ugo Becciani, Massimo Brescia, Stefano Cavuoti, Alessandro Costa, Anna M. Di Giorgio, Davide Elia, Akos Hajnal, Hermann Gabor, Peter Kacsuk, Scige J. Liu, Sergio Molinari, Giuseppe Riccio, Eugenio Schisano, Eva Sciacca, Riccardo Smareglia, and Fabio Vitello. Vialactea knowledge base homogenizing access to milky way data, 2016.
- [8] Maria A. Nieto-Santisteban, Alexander S. Szalay, Aniruddha R. Thakar, William J. O’Mullane, Jim Gray, and James Annis. When database systems meet the grid, 2005.
- [9] Shubham S. Srivastava, Medha Atre, Shubham Sharma, Rahul Gupta, and Sandeep K. Shukla. Verity: Blockchains to detect insider attacks in dbms, 2019.
- [10] Ralph M Stair and George W Reynolds. *Fundamentals of information systems*. Cengage Learning, 2018.
- [11] Jim Gray. Data management: Past, present, and future, 2007.
- [12] Ladan Golshanara and Jan Chomicki. Temporal data exchange, 2019.
- [13] Jim Gray. The revolution in database system architecture, 2004.
- [14] Serge Abiteboul, Marcelo Arenas, Pablo Barceló, Meghyn Bienvenu, Diego Calvanese, Claire David, Richard Hull, Eyke Hüllermeier, Benny Kimelfeld, Leonid Libkin, Wim Martens, Tova Milo, Filip Murlak, Frank Neven, Magdalena Ortiz, Thomas Schwentick, Julia Stoyanovich, Jianwen Su, Dan Suciu, Victor Vianu, and Ke Yi. Research directions for principles of data management (dagstuhl perspectives workshop 16151), 2017.
- [15] Shuai Zhang, Wenxi Zeng, I-Ling Yen, and Farokh B. Bastani. Semantically enhanced time series databases in iot-edge-cloud infrastructure, 2019.
- [16] Ryan Wong, Nikita Kim, Kevin Higgs, Sapan Agarwal, Engin Ipek, Saugata Ghose, and Ben Feinberg. Tcam-ssd: A framework for search-based computing in solid-state drives, 2024.
- [17] Maryam Aliakbarpour, Amartya Shankha Biswas, Themistoklis Gouleakis, John Peebles, Ronitt Rubinfeld, and Anak Yodpinyanee. Sublinear-time algorithms for counting star subgraphs with applications to join selectivity estimation, 2016.
- [18] Ruihong Wang, Jianguo Wang, Stratos Idreos, M. Tamer Özsu, and Walid G. Aref. The case for distributed shared-memory databases with rdma-enabled memory disaggregation, 2022.
- [19] Xinyu Liu, Qi Zhou, Joy Arulraj, and Alessandro Orso. Testing dbms performance with mutations, 2021.
- [20] Jinsheng Ba and Manuel Rigger. Testing database engines via query plan guidance, 2023.

-
- [21] Vijay Gadepally, Peinan Chen, Jennie Duggan, Aaron Elmore, Brandon Haynes, Jeremy Kepner, Samuel Madden, Tim Mattson, and Michael Stonebraker. The bigdawg polystore system and architecture, 2016.
 - [22] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch. Time, tense and aspect in natural language database interfaces, 1998.
 - [23] Ryan Marcus and Olga Papaemmanouil. Towards a hands-free query optimizer through deep learning, 2018.
 - [24] Joaquín Keller, Raluca Diaconu, and Mathieu Valero. Towards a scalable dynamic spatial database system, 2012.
 - [25] Mayank Raikwar, Danilo Gligoroski, and Goran Velinov. Trends in development of databases and blockchain, 2020.
 - [26] Anisoara Nica, Fabian Suchanek, and Aparna Varde. Emerging multidisciplinary research across database management systems, 2011.
 - [27] Jeffrey Jestes, Jeff M. Phillips, Feifei Li, and Mingwang Tang. Ranking large temporal data, 2012.
 - [28] Dawei Jiang, Qingchao Cai, Gang Chen, H. V. Jagadish, Beng Chin Ooi, Kian-Lee Tan, and Anthony K. H. Tung. Cohort query processing, 2016.
 - [29] Ioana Banicescu, Trisha Chakraborty, Seth Gilbert, and Maxwell Young. A survey on adversarial contention resolution, 2024.
 - [30] Qingpeng Cai, Can Cui, Yiyuan Xiong, Wei Wang, Zhongle Xie, and Meihui Zhang. A survey on deep reinforcement learning for data processing and analytics, 2022.
 - [31] Maciej Besta, Robert Gerstenberger, Emanuel Peter, Marc Fischer, Michał Podstawski, Claude Barthels, Gustavo Alonso, and Torsten Hoefer. Demystifying graph databases: Analysis and taxonomy of data organization, system designs, and graph queries, 2023.
 - [32] Yanif Ahmad, Oliver Kennedy, Christoph Koch, and Milos Nikolic. Dbtoaster: Higher-order delta processing for dynamic, frequently fresh views, 2012.
 - [33] Hanxian Huang, Tarique Siddiqui, Rana Alotaibi, Carlo Curino, Jyoti Leeka, Alekh Jindal, Jishen Zhao, Jesus Camacho-Rodriguez, and Yuanyuan Tian. Sibyl: Forecasting time-evolving query workloads, 2024.
 - [34] Eya ben Ahmed and Mohamed Salah Gouider. Towards an incremental maintenance of cyclic association rules, 2010.
 - [35] Meihui Zhang, Zhongle Xie, Cong Yue, and Ziyue Zhong. Spitz: A verifiable database system, 2020.
 - [36] Sho Nakazono, Hiroyuki Uchiyama, Yasuhiro Fujiwara, and Hideyuki Kawashima. Scheduling space expander: An extension of concurrency control for data ingestion queries, 2023.
 - [37] Hojjat Karami, David Atienza, and Anisoara Ionescu. Tee4ehr: Transformer event encoder for better representation learning in electronic health records, 2024.
 - [38] J. Gerard Wolff. Towards an intelligent database system founded on the sp theory of computing and cognition, 2003.
 - [39] Tarek A. Lahlou and Thomas A. Baran. Web services for asynchronous, distributed optimization using conservative signal processing, 2015.
 - [40] Martin Grambow, Denis Kovalev, Christoph Laaber, Philipp Leitner, and David Bermbach. Using microbenchmark suites to detect application performance changes, 2022.
 - [41] Vivek Shah and Marcos Antonio Vaz Salles. Reactors: A case for predictable, virtualized actor database systems, 2018.

-
- [42] D. E. R. Quock, F. H. Munson, K. J. Eder, and S. L. Dean. The relational database aspects of argonne’s atlas control system, 2001.
 - [43] Sergey V. Zykov. Web-based enterprise information systems development: The integrated methodology, 2006.
 - [44] Dimitrios Vasilas, Marc Shapiro, Bradley King, and Sara Hamouda. Towards application-specific query processing systems, 2020.
 - [45] Mohammed Saeed, Nicola De Cao, and Paolo Papotti. Querying large language models with sql, 2023.
 - [46] Uttam Mukhopadhyay. An uncertainty management calculus for ordering searches in distributed dynamic databases, 2013.
 - [47] Kyriaki Dimitriadou, Olga Papaemmanouil, and Yanlei Diao. Aide: An automated sample-based approach for interactive data exploration, 2015.
 - [48] Sabah Al-Fedaghi. Conceptual temporal modeling applied to databases, 2021.
 - [49] Bonil Shah, P. M. Jat, and Kalyan Sashidhar. Performance study of time series databases, 2022.
 - [50] Manel Zarrouk and Med Salah Gouider. Frequent patterns mining in time-sensitive data stream, 2012.
 - [51] Ion Androutsopoulos. A principled framework for constructing natural language interfaces to temporal databases, 1996.
 - [52] Fotis Psallidas, Ashvin Agrawal, Chandru Sugunan, Khaled Ibrahim, Konstantinos Karanasos, Jesús Camacho-Rodríguez, Avrielia Floratou, Carlo Curino, and Raghu Ramakrishnan. One-provenance: Efficient extraction of dynamic coarse-grained provenance from database logs [technical report], 2023.
 - [53] Xinyi Zhang, Hong Wu, Yang Li, Jian Tan, Feifei Li, and Bin Cui. Towards dynamic and safe configuration tuning for cloud databases, 2022.
 - [54] Chen Yang, Xiaofeng Meng, and Zhihui Du. Cloud based real-time and low latency scientific event analysis, 2018.
 - [55] Sutanay Choudhury, Lawrence Holder, George Chin, and John Feo. Fast search for dynamic multi-relational graphs, 2013.
 - [56] Xuebin Ren, Chia-Mu Yu, Wei Yu, Xinyu Yang, Jun Zhao, and Shusen Yang. Dpcrowd: Privacy-preserving and communication-efficient decentralized statistical estimation for real-time crowd-sourced data, 2020.
 - [57] Xuan Zhou, Xin Zhou, Zhengtai Yu, Hua Guo, and Kian-Lee Tan. Decentralizing mvcc by leveraging visibility, 2018.
 - [58] Mahnoosh Kholghi and MohammadReza Keyvanpour. Comparative evaluation of data stream indexing models, 2012.
 - [59] Amirali Boroumand, Saugata Ghose, Geraldo F. Oliveira, and Onur Mutlu. Enabling high-performance and energy-efficient hybrid transactional/analytical databases with hardware/software cooperation, 2022.
 - [60] Mohit Kumar Gupta, Vishal Verma, and Megha Singh Verma. In-memory database systems - a paradigm shift, 2014.
 - [61] Amirali Boroumand, Saugata Ghose, Geraldo F. Oliveira, and Onur Mutlu. Polynesia: Enabling effective hybrid transactional/analytical databases with specialized hardware/software co-design, 2021.
 - [62] Wolfgang Mauerer, Ralf Ramsauer, Edson R. F. Lucas, and Stefanie Scherzinger. Silentium! run-analyse-eradicate the noise out of the db/os stack, 2021.

-
- [63] Michael Fruth, Stefanie Scherzinger, Wolfgang Maurer, and Ralf Ramsauer. Tell-tale tail latencies: Pitfalls and perils in database benchmarking, 2021.
 - [64] Jinsheng Ba and Manuel Rigger. Cert: Finding performance issues in database systems through the lens of cardinality estimation, 2024.
 - [65] Alexander van Renen, Lukas Vogel, Viktor Leis, Thomas Neumann, and Alfons Kemper. Persistent memory i/o primitives, 2019.
 - [66] Caetano Sauer, Goetz Graefe, and Theo Härder. Instant restore after a media failure, 2017.
 - [67] Xinyi Zhang, Zhuo Chang, Yang Li, Hong Wu, Jian Tan, Feifei Li, and Bin Cui. Facilitating database tuning with hyper-parameter optimization: A comprehensive experimental evaluation, 2022.
 - [68] Raj P. Gopalan, Tariq Nuruddin, and Yudho Giri Sucahyo. A seamless integration of association rule mining with database systems, 2001.
 - [69] Hagit Grushka-Cohen, Oded Sofer, Ofer Biller, Michael Dymshits, Lior Rokach, and Bracha Shapira. Sampling high throughput data for anomaly detection of data-base activity, 2017.
 - [70] Ryan Marcus and Olga Papaemmanouil. Flexible operator embeddings via deep learning, 2019.
 - [71] Immanuel Trummer. From bert to gpt-3 codex: Harnessing the potential of very large language models for data management, 2023.
 - [72] Jing Chen, Shengyi Yang, Weiping Ding, Peng Li, Aijun Liu, Hongjun Zhang, and Tian Li. Incremental high average-utility itemset mining: survey and challenges, 2024.
 - [73] Sylvain Kubler, William Derigent, André Thomas, and Eric Rondeau. Key factors for information dissemination on communicating products and fixed databases, 2011.
 - [74] Maxime Schoemans, Walid G. Aref, Esteban Zimányi, and Mahmoud Sakr. Multi-entry generalized search trees for indexing trajectories, 2024.
 - [75] Giovanni Lupi. Multi-objective optimization: basic approaches and moving beyond them through flexible skyline queries, 2022.
 - [76] George Onoufriou, Ronald Bickerton, Simon Pearson, and Georgios Leontidis. Nemesyst: A hybrid parallelism deep learning-based framework applied for internet of things enabled food retailing refrigeration systems, 2019.
 - [77] A. M. Sowjanya and M. Shashi. New proximity estimate for incremental update of non-uniformly distributed clusters, 2013.
 - [78] Amir Gilad, Daniel Deutch, and Sudeepa Roy. On multiple semantics for declarative database repairs, 2020.
 - [79] Aaditya Ramdas, Fanny Yang, Martin J. Wainwright, and Michael I. Jordan. Online control of the false discovery rate with decaying memory, 2017.
 - [80] Floris Geerts, Sofie Haesevoets, and Bart Kuijpers. First-order complete and computationally complete query languages for spatio-temporal databases, 2005.
 - [81] Alessandro Berti and Wil van der Aalst. Extracting multiple viewpoint models from relational databases, 2020.
 - [82] Siqiao Xue, Caigao Jiang, Wenhui Shi, Fangyin Cheng, Keting Chen, Hongjun Yang, Zhiping Zhang, Jianshan He, Hongyang Zhang, Ganglin Wei, Wang Zhao, Fan Zhou, Danrui Qi, Hong Yi, Shaodong Liu, and Faqiang Chen. Db-gpt: Empowering database interactions with private large language models, 2024.

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.