

Unifying the Data: An Analysis of Multi-Model Databases

Ragini Kalvade, University of Illinois at Chicago

1. Introduction

A significant challenge in big data management lies in addressing the variety of data. Data is generated from various sources, exists in multiple formats, and follows different data models. Initially, relational databases were the primary solution, excelling in handling highly structured, transactional data. However, the rise of semi-structured and unstructured data led to the emergence of NoSQL databases, offering flexibility in managing free-form data entries. Modern applications now encounter diverse datasets, including user transactions, activity logs, and error reports, which often involve intricate relationships and interconnectedness. Consequently, neither relational databases nor NoSQL systems fully satisfy these modern demands.

Multi-Model Databases (MMDBs) provide a unified platform capable of managing heterogeneous datasets effectively and efficiently. These databases integrate various storage engines, allowing data to be stored in formats best suited for their respective use cases, such as key-value pairs, relational tables, graphs, or XML/JSON documents.

This paper explores recent advancements in multi-model databases and their significance in addressing big data challenges. The structure of this paper is as follows: Section II discusses the historical development of MMDBs. Section III outlines the architectural design of MMDBs. Section IV examines their features and functionalities. Section V highlights applications and use cases. Finally, Section VI concludes with key insights and future directions.

2. Background and Historical Context

The evolution of database systems has been heavily influenced by the growing complexity and diversity of data. Initially, traditional relational databases dominated the landscape, excelling in managing highly structured data with clearly defined schemas. However, as the digital era advanced, a broader range of data types emerged, including semi-structured (e.g., XML and JSON) and unstructured formats (e.g., images, videos, and social media data). Relational databases struggled to adapt to these new data forms, leading to the rise of NoSQL databases in the late 2000s.

NoSQL systems introduced flexibility by accommodating key-value, document, columnar, and

graph models, each designed to handle specific use cases. For instance, graph databases like Neo4j were suited for relationship-heavy data, while document stores like MongoDB excelled in managing semi-structured content. However, using specialized databases for different tasks created data silos, increasing system complexity and maintenance challenges.

To address this issue, the concept of multi-model databases emerged in the early 2010s. Unlike their single-model counterparts, multi-model systems support multiple data representations within a single database engine. This innovation aimed to unify diverse data types, allowing organizations to handle various workloads more efficiently without relying on multiple database systems.

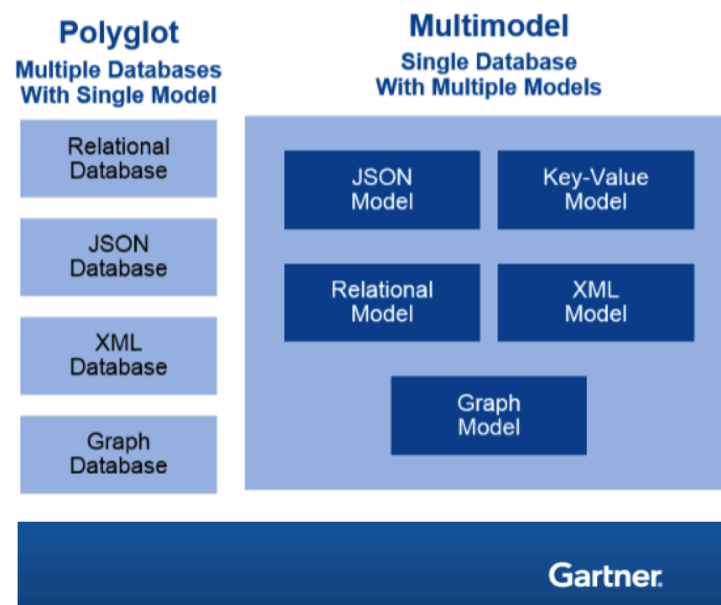


Image extracted from [2]

The term “multi-model database” was popularized by Luca Garulli, founder of OrientDB, in 2012. Since then, multi-model systems have gained traction, combining the strengths of relational databases and NoSQL systems while minimizing their limitations. These databases enable applications to store and query relational, graph, key-value, and document data seamlessly, offering a powerful and versatile solution for the ever-evolving data landscape.

Today, multi-model databases play a crucial role in addressing the challenges posed by big data, particularly in industries requiring diverse and interconnected datasets. Their adoption continues

to grow as they offer scalability, flexibility, and a unified platform for handling complex data ecosystems.

3. System Architecture of Multi-Model Databases

Core Components of MMDB Architecture

Data Storage Layer: MMDBs utilize a unified storage engine capable of managing multiple data types. The storage layer is designed to handle relational tables, hierarchical documents (e.g., JSON, XML), key-value pairs, and graph structures, ensuring consistency and optimized performance. Some systems leverage hybrid storage techniques to balance performance and scalability for mixed workloads.

Query Processing Engine: A versatile query processor forms the backbone of MMDBs. It often blends traditional SQL capabilities with NoSQL-like operations to support complex queries across models. For instance, a single query might involve joining relational data with document-based data or performing graph traversals.

Schema Management: MMDBs offer both schema-less flexibility for semi-structured data (e.g., JSON) and structured schemas for relational models. This hybrid approach allows developers to choose the schema design best suited for their application while maintaining the ability to adapt to evolving requirements.

Cross-Model Relationships: Cross-model relationships are a distinguishing feature of MMDBs, enabling seamless interactions between different data models. For example, relational data can be joined with graph data, or document metadata can be linked to key-value pairs. These relationships reduce redundancy and simplify complex queries.

Indexing and Optimization: MMDBs implement specialized indexing mechanisms tailored to each data model. For instance:

- B-trees for relational tables.
- Hash indexes for key-value stores.

- Adjacency lists or graph-specific indexes for graph data.
- Caching and parallel processing are also used to enhance query performance across heterogeneous datasets.

Data Integration and API Layer: MMDBs provide integration layers that support APIs for different programming languages, enabling developers to interact with various models through unified interfaces. This layer facilitates easy adoption and seamless integration into diverse application ecosystems.

Example: SurrealDB

SurrealDB is a modern multi-model database that integrates relational, document, graph, and key-value data models within a single platform. Here's how SurrealDB aligns with the MMDB architecture:

Unified Data Storage: SurrealDB uses a single engine to store and query relational tables, hierarchical document data (JSON), and graph data. This enables developers to build applications that handle interconnected and complex datasets without switching databases.

Query Language: SurrealDB introduces an SQL-like query language with extensions for document and graph data. For instance:

- Developers can write SQL queries for structured data.
- Use JSON-based syntax for working with semi-structured data.
- Perform graph traversals for exploring relationships between nodes.

Dynamic Schema Management: SurrealDB supports schema-less operations for flexibility but allows optional schema definitions for data validation. This ensures adaptability while maintaining data integrity.

Performance Optimization: SurrealDB implements hybrid indexing strategies, combining traditional relational indexing with graph-based optimizations. Its distributed architecture allows horizontal scaling, making it suitable for large-scale applications with diverse data workloads.

Cross-Model Relationships: SurrealDB natively supports linking relational and graph data. For

example, user profiles in a relational table can reference relationships (e.g., friendships) stored as graph edges, enabling powerful cross-domain queries.

Use Case Example: SurrealDB is particularly useful in building social network applications where relational data (user profiles) must interact with graph data (friendship relationships). Its unified architecture simplifies development and improves performance by avoiding the need for multiple database systems.

4. Features and Functionalities

Unified Query Language

MMDBs typically offer a unified query interface that integrates SQL and NoSQL paradigms. This enables users to perform complex operations, such as joins, graph traversals, and aggregations, across different data models with ease. Optimized query engines ensure consistent performance for mixed-model queries.

Flexibility and Scalability

By accommodating diverse data formats—ranging from structured relational data to semi-structured JSON or XML documents and unstructured graph data—MMDBs provide unmatched flexibility. Horizontal scalability further enables them to handle high-throughput operations while adapting to dynamic schema requirements.

Cross-Model Relationships

MMDBs simplify the management of relationships between different data models. For instance, graph traversals can incorporate relational data or key-value lookups, enabling comprehensive cross-domain queries without external integrations.

Performance Optimization

Sophisticated indexing mechanisms are tailored for multi-model operations. Hybrid indexing strategies—such as B-trees for relational data, adjacency lists for graph data, and hash indexes for key-value pairs—optimize query performance. Additional features like caching and parallel

processing enhance efficiency for complex workloads.

5. Use Cases and Applications

IBM Watson Health leverages multi-model databases (MMDBs) to efficiently manage and analyze diverse healthcare data, including structured, semi-structured, and unstructured formats. These databases are essential for integrating various data types in a unified platform, enabling advanced analytics and decision-making.

Integrating Diverse Data Sources

Healthcare data is highly diverse, requiring the integration of multiple data models:

1. **Relational Models:** These handle structured data, such as patient demographics, medical histories, and transactional information. Relational databases ensure data consistency, which is crucial for regulatory compliance and healthcare reporting.
2. **Document Models:** Semi-structured data, like lab results and genomic sequences, is stored using document models. These models support flexible data formats such as JSON and XML, accommodating the complex, evolving nature of genomic and clinical data.
3. **Key-Value Stores:** Unstructured data, including medical images (e.g., MRIs, X-rays) and physician notes, is stored in key-value databases. These systems efficiently store large volumes of unstructured data, making it easy to retrieve and analyze based on metadata or identifiers.

Enabling Advanced Analytics

MMDBs empower Watson Health to perform sophisticated analytics by integrating these diverse data sources:

- **Predictive Analytics:** By combining structured electronic health records (EHRs) with genomic data, Watson Health can create personalized treatment plans and predict health risks. For example, linking patient history with genetic information helps in early disease detection and tailored interventions.
- **Graph Models for Network Analysis:** Graph databases help analyze relationships

between patients, healthcare providers, and treatment protocols. This enables Watson Health to identify trends in patient care and improve healthcare delivery by visualizing the networks of interactions within the healthcare ecosystem.

Scalability and Compliance

Healthcare systems require both scalability and regulatory compliance:

- **Scalability:** MMDBs enable Watson Health to scale horizontally to manage the increasing volume of healthcare data from diverse sources, including wearable devices and medical imaging, ensuring robust performance and capacity growth.
- **Compliance with Regulations:** MMDBs support compliance with regulations like HIPAA by offering strong encryption, access controls, and audit features, ensuring that patient data remains secure and private.

In conclusion, MMDBs are crucial for Watson Health's ability to handle complex, large-scale healthcare data. Their integration of structured, semi-structured, and unstructured data models supports personalized care, predictive analytics, and compliance with regulatory standards, driving improvements in patient outcomes and operational efficiency.

6. Conclusion

In conclusion, multi-model databases (MMDBs) have emerged as a pivotal solution in the evolving landscape of data management, particularly in the era of big data. Their ability to handle diverse and complex datasets—ranging from structured to unstructured data—makes them an invaluable tool for modern applications across industries. The unified architecture of MMDBs combines scalability, flexibility, and performance optimization, addressing the need for data systems that can efficiently process heterogeneous data sources while maintaining high performance.

The integration of multiple data models within a single platform eliminates the need for disparate data systems, streamlining architecture and simplifying maintenance. This capability is especially important for businesses leveraging real-time analytics, machine learning, and AI, where the seamless handling of diverse data types is crucial for extracting actionable insights. As

the demand for data-driven decision-making grows, MMDBs will play an increasingly central role in enabling organizations to harness the full potential of their data.

However, several challenges remain that hinder the widespread adoption and optimization of MMDBs. Key among these are the lack of standardized query languages and benchmarking methodologies, which complicate the evaluation and comparison of different MMDB solutions. To fully unlock the capabilities of MMDBs, collaborative efforts between academia and industry are needed to develop unified standards and frameworks that will drive further innovation and make MMDBs more accessible to a broader range of applications.

As businesses continue to evolve and embrace advanced technologies like AI and machine learning, the demand for integrated, scalable, and efficient data management solutions will only increase. Multi-model databases are well-positioned to meet these demands, offering a versatile and future-proof solution that will support the next generation of data ecosystems.

References

1. Jiaheng Lu, Irena Holubová. *Multi-model Databases: A New Journey to Handle the Variety of Data*. ACM Computing Surveys, Volume 52, Issue 3, May 2020.
<https://doi.org/10.1145/3323214>
2. ChaosSearch. *Why Enterprises Need a True Multi-Model Platform*.
<https://www.chaossearch.io/blog/why-enterprises-need-a-true-multi-model-platform>
3. Oracle. *Introduction to Multi-Model Databases*.
<https://link.springer.com/article/10.1007/s10619-023-07433-1>
4. Medium. *The Rise of Multi-Model Databases*.
<https://medium.com/@igniobydigitate/the-rise-of-multi-model-databases-6e26c173c830>
5. SurrealDB, <https://surrealdb.com/docs/>