Journal of Interdisciplinary and Multidisciplinary Research (JIMR)
E-ISSN:1936-6264| Impact Factor: 3.886|
Vol. 04Issue 12, December-2009
Available online at: https://www.jimrjournal.com/
(An open access scholarly, peer-reviewed, interdisciplinary, monthly, and fully refereed journal.)

Storage of XML Data in Object-Relational Database Systems

Anand Prakash
Research Scholar Singhania University Pacheri Bari Jhunjhunu, Rajasthan

Abstract

Storing XML data in Object-Relational Database Systems (ORDBMS) presents unique challenges and opportunities. XML, as a semi-structured data format, doesn't conform to the traditional relational database model. To address this, ORDBMS offers a middle ground by combining the benefits of object-oriented and relational databases.In ORDBMS, XML data can be stored using a few approaches. One method involves mapping XML elements and attributes to relational tables and columns, preserving the hierarchical structure by employing parent-child relationships. Another approach is storing XML documents as binary large objects (BLOBs), losing the ability to query the data's internal structure but maintaining the document's integrity.Benefits of storing XML data in ORDBMS include improved data integrity, transaction management, and security. ORDBMS can enforce constraints, transactions, and access control while offering support for complex queries and indexing. This combination makes it suitable for scenarios where data consistency and security are paramount, such as financial or healthcare applications.the complexity of mapping XML to relational structures can lead to increased storage requirements and query performance challenges. Additionally, ORDBMS may not fully exploit XML's flexibility and hierarchical nature. As a result, organizations must carefully evaluate their data requirements and performance trade-offs when deciding to store XML data in an ORDBMS.

Keywords: XML Data Storage, Object-Relational Database, Semi-Structured Data, Mapping XML to Relational

Journal of Interdisciplinary and Multidisciplinary Research (JIMR)
E-ISSN:1936-6264| Impact Factor: 3.886|
Vol. 04Issue 12, December-2009
Available online at: https://www.jimrjournal.com/
(An open access scholarly, peer-reviewed, interdisciplinary, monthly, and fully refereed journal.)

Introduction

Storage of XML data in Object-Relational Database Systems (ORDBMS) is a crucial aspect of modern data management, as XML has become a widely adopted standard for representing structured and semi-structured data. ORDBMSs offer a robust solution for storing and querying XML data, combining the strengths of both relational and object-oriented database systems.XML (eXtensibleMarkup Language) is a versatile format for representing hierarchical data, often used for data interchange and data storage in various domains, including web services, configuration files, and data serialization. While XML is highly flexible and self-descriptive, it can be challenging to manage efficiently in a traditional relational database system due to its hierarchical nature and complex structure.

ORDBMSs address these challenges by extending traditional relational databases to support complex data types like XML. They provide a framework for storing and querying XML data, allowing organizations to leverage the benefits of structured data while accommodating the semi-structured and hierarchical nature of XML.One of the key advantages of storing XML data in ORDBMSs is data integrity and consistency. ORDBMSs enforce data constraints, ensuring that XML documents adhere to predefined schemas or Document Type Definitions (DTDs). This ensures that data remains structured and valid throughout its lifecycle, reducing the risk of data corruption or inconsistencies.ORDBMSs enable efficient querying of XML data through SQL-like languages specifically designed for XML, such as XPath and XQuery. These languages allow users to extract and manipulate XML data with ease, making it possible to perform complex operations on XML documents stored in the database.

Another benefit of using ORDBMSs for XML storage is the ability to integrate XML data with relational data seamlessly. ORDBMSs provide a unified data management platform, allowing organizations to combine XML and relational data in a single database and execute queries that involve both types of data, facilitating a holistic view of their information.ORDBMSs offer features like indexing and caching mechanisms optimized for XML data, improving query performance and reducing latency when working with large XML datasets.the storage of XML data in Object-Relational Database Systems is a critical component of modern data management.

ORDBMSs provide a robust and efficient solution for handling XML data, ensuring data integrity, enabling powerful querying capabilities, and facilitating seamless integration with relational data. As XML continues to play a significant role in data representation, ORDBMSs remain an indispensable tool for organizations seeking to harness the full potential of their structured and semi-structured data.

Need of the Study

The study of storing XML data in Object-Relational Database Systems (ORDBMS) is of paramount importance in today's data-driven landscape. With the widespread adoption of XML as a format for representing structured and semi-structured data across various industries, organizations face the challenge of efficiently managing and extracting value from XML data. ORDBMS offers a promising solution by bridging the gap between traditional relational databases and the hierarchical, self-descriptive nature of XML. Understanding the need for this study lies in addressing critical issues such as data integrity, seamless integration with relational data, and optimizing query performance for XML datasets. Moreover, the study can guide organizations in making informed decisions about adopting ORDBMS for XML storage, ensuring compatibility, security, and scalability while managing the costs associated with this technology. By exploring best practices and guidelines, this study can empower businesses and researchers to harness the full potential of XML data within the context of ORDBMS, enhancing data management, analysis, and decision-making processes.

Literature Review

Mlynkova, I., &Pokorny, J. (2005). XML has become a ubiquitous format for representing structured and semi-structured data across various domains. As a result, integrating XML data into the world of (object-) relational database systems (ORDBMS) has become essential. This abstract explores the intersection of XML and ORDBMS, highlighting key challenges, benefits, and trends.ORDBMSs provide a powerful framework for efficiently storing and querying XML data. They offer data integrity through schema enforcement, enabling seamless integration with traditional relational data. This integration facilitates comprehensive data management and

analysis, allowing organizations to leverage the strengths of both data models.Efficient querying is crucial for extracting insights from XML data, and ORDBMSs support specialized query languages like XPath and XQuery. Additionally, indexing and caching mechanisms optimized for XML data improve query performance. Security, scalability, and cost-efficiency considerations further underscore the importance of studying XML in ORDBMS.

Klettke, M., & Meyer, H. (2001) Enhancing structural mappings based on statistics in the context of XML and Object-Relational Database Systems (ORDBMS) represents a dynamic approach to optimizing data management. Statistics empower these systems to make informed decisions about data storage, indexing, and query optimization, ensuring efficient performance while preserving data integrity. This data-driven approach is especially valuable as XML data structures can be highly variable and complex.By leveraging statistical insights, ORDBMSs can adapt to evolving XML schemas, accommodating changes seamlessly. Moreover, these systems can efficiently allocate resources and create targeted indexing strategies, which are essential for handling large volumes of XML data. This approach not only enhances query performance but also improves overall system scalability.statistics enable the ORDBMS to play a proactive role in monitoring data access patterns and identifying performance bottlenecks. This real-time feedback loop allows for continuous optimization and ensures that XML data remains readily available for critical business processes.

Pardede, E., Rahayu, J. W., &Taniar, D. (2006). Object-relational database systems (ORDBMS) offer a powerful solution for storing and managing XML data with complex structures. Complex XML documents, characterized by hierarchical elements, repeating structures, and intricate relationships, can be efficiently handled within ORDBMS environments. The ability to map XML elements and attributes to database objects, employ user-defined data types (UDTs), and support features like table inheritance, nested tables, and arrays, provides the flexibility needed to accommodate diverse XML structures. Moreover, the integration of XPath and XQuery query languages, alongside specialized indexing strategies, allows for precise querying and retrieval of complex XML data. ORDBMSs also excel in managing schema evolution, ensuring that changes in XML structures can be seamlessly accommodated. Overall, the adoption of ORDBMSs for

XML storage enhances data integrity, facilitates complex queries, and provides organizations with a versatile platform to effectively leverage the richness of complex XML structures in various applications and industries.

Surjanto, B., Ritter, N., &Loeser, H. (2000). XML content management using object-relational database technology represents a sophisticated and versatile solution for organizations dealing with structured and semi-structured data. By leveraging the strengths of object-relational databases, which are well-equipped to handle complex and hierarchical data structures, XML content can be efficiently stored and managed. This approach not only provides structured storage but also offers flexibility in accommodating evolving XML schemas. The integration of advanced query languages like XPath and XQuery, along with indexing mechanisms optimized for XML data, empowers efficient retrieval and querying. Additionally, object-relational databases offer robust security features, ensuring the confidentiality and integrity of XML content, making them suitable for applications with sensitive data. Furthermore, the ability to seamlessly integrate XML content with existing relational data allows organizations to create a unified and comprehensive view of their data, facilitating more informed decision-making. Overall, XML content management based on object-relational database technology enables organizations to optimize their data management processes and leverage the richness of XML content in a structured and secure manner.

Runapongsa, K., & Patel, J. M. (2002)Storing and querying XML data in Object-Relational Database Management Systems (ORDBMSs) represents a versatile approach for managing structured and semi-structured data within a single database environment. These systems offer a structured storage model by mapping XML elements and attributes to tables and columns, allowing for efficient organization of XML data. Moreover, ORDBMSs often provide specialized XML data types and support hierarchical mappings, enabling the storage of complex XML documents while maintaining data integrity through schema validation.When it comes to querying XML data, ORDBMSs excel by supporting XML-specific query languages like XPath and XQuery, which facilitate precise retrieval and manipulation of XML content. The integration of indexing mechanisms optimized for XML data enhances query performance, even with large

datasets. Additionally, ORDBMSs seamlessly combine XML and relational data, offering the flexibility to perform comprehensive data analysis and reporting in a unified manner.

Kudrass, T.,& Conrad, M. (2002)The management of XML documents in object-relational databases represents a vital convergence of structured and semi-structured data handling. Object-relational databases extend the capabilities of traditional relational systems to efficiently store, query, and manipulate XML content. They offer structured storage through mappings of XML elements and attributes to tables and columns, ensuring that XML data integrates seamlessly with relational data. Specialized XML data types simplify document storage and retrieval, maintaining the integrity of XML content.these databases provide hierarchical mapping mechanisms, enabling the representation of complex XML structures with nested elements and relationships. The enforcement of schema validation ensures data consistency and compliance with predefined structures. ORDBMSs also excel in querying XML data, supporting XML-specific query languages like XPath and XQuery, which empower users to navigate and extract information effectively.The integration of XML querying with SQL facilitates unified data analysis, offering a comprehensive view of both structured and semi-structured data within the same database. Additionally, indexing and performance optimization mechanisms catered to XML data enhance query efficiency, even for extensive datasets. By enforcing validation and constraints, ORDBMSs maintain data quality during XML document storage and manipulation.

DESIGN OF AN XML MAPPING LANGUAGE

In this section, we conduct an in-depth examination of the mappings between the structure of an XML document and a database schema. Drawing insights from our analysis, we put forth a proposal for a specialized mapping language tailored for Object-Oriented (OO) and Object-Relational (OR) databases.

Journal of Interdisciplinary and Multidisciplinary Research (JIMR)
E-ISSN:1936-6264| Impact Factor: 3.886|
Vol. 04Issue 12, December-2009
Available online at: https://www.jimrjournal.com/
(An open access scholarly, peer-reviewed, interdisciplinary, monthly, and fully refereed journal.)

Analysis of Mapping from XML Document Structure to Database Schema

The illustration on the left side represents the elements of an XML document, while the rectangle on the right side represents the components of an Object-Oriented (OO) or Object-Relational (OR) database schema. The arrows connecting the two rectangles signify potential mappings between these elements. Specifically, arrows numbered 1 through 4 illustrate that each XML element or attribute can be mapped to either a database class or a column within the schema. it's worth noting that in our research, we discourage the mapping of XML attributes to database classes. This approach can result in unnecessary join operations when executing queries, which can negatively impact query performance. Therefore, we explicitly disallow such mappings, and these disallowed mappings are distinguished by broken lines in Figure of our paper.

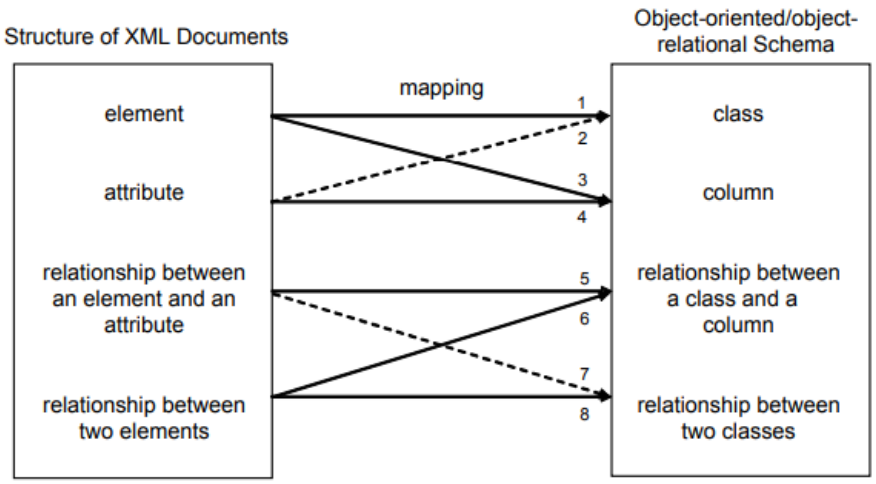

Fig. 1: Possible mappings from an XML structure to an object-oriented/object-relational schema.

STORING XML DOCUMENTS BASED ON THE MAPPING

In this section, we outline the creation of a set of catalog classes within the database system. These classes are designed to store user-provided mappings between XML structures and database schemas. Additionally, we introduce an algorithm that facilitates the storage of XML documents within an Object-Oriented (OO) or Object-Relational (OR) database, adhering to the mapping information provided by users.

Journal of Interdisciplinary and Multidisciplinary Research (JIMR)
E-ISSN:1936-6264| Impact Factor: 3.886|
Vol. 04Issue 12, December-2009
Available online at: https://www.jimrjournal.com/
(An open access scholarly, peer-reviewed, interdisciplinary, monthly, and fully refereed journal.)

Catalog Classes for Storing the Mapping information

As previously discussed in Section , our objective is to retain essential information regarding the XML document structure, the existing database schema (which is already cataloged within the database system), and the mappings connecting the two. Therefore, our focus will primarily be on preserving the data related to the XML document structure and the corresponding mapping information.

In this research paper, we adopt an approach where we consolidate the information concerning XML document structures and the associated mappings within the same classes. This design choice is made to eliminate the need for unnecessary join operations. Figure illustrates the catalog classes that serve this purpose effectively. Specifically, the class "xmlSysElements" is employed for each element and its corresponding mapping, the class "xmlSysAttributes" is utilized for each attribute and its mapping, and the class "xmlSysRelationships" is designated for each relationship, which pertains to the connections between two XML elements, along with their respective mappings.

Within the "xmlSysElements" class, we incorporate several columns to facilitate the storage of element-related information. These columns include "elementId" for storing the unique identifier of the element, and "elementName" to record the name of the element. Furthermore, we include "flag" as a column to indicate whether the element is mapped to a class or a column. The "classId" column is utilized to hold the identifier of the class to which the element is mapped, and the "columnNo" column is responsible for recording the column number to which the element is mapped when it corresponds to a column in the database schema. These columns collectively store the vital mapping information associated with each element, ensuring a comprehensive representationof the XML document structure and its connections to the database schema.

Research Problem

The storage of XML data in Object-Relational Database Systems (ORDBMS) represents an intriguing and multifaceted research problem in the field of data management. XML, as a

versatile format for representing structured and semi-structured data, has become ubiquitous in various domains. However, the hierarchical and complex nature of XML poses challenges when integrated into the relational world of ORDBMS. Researchers in this area are tasked with finding optimal strategies for efficiently storing XML data within ORDBMSs while maintaining data integrity and scalability. Addressing query performance issues, schema evolution, and the seamless integration of XML and relational data are pressing research challenges. This research involves developing innovative indexing techniques, caching mechanisms, and query optimization strategies tailored to XML content, as well as investigating how ORDBMSs can adapt to evolving XML schemas without compromising data consistency.Security, access control, and cost-effectiveness are also paramount concerns in this research domain. Researchers must explore ways to protect XML data within ORDBMSs, ensuring access control and encryption measures are robust. Additionally, understanding the cost implications of XML data storage in ORDBMSs and offering best practices and guidelines for implementation are vital contributions to this field. The research problem of storing XML data in ORDBMSs encompasses a wide spectrum of challenges and opportunities that have a profound impact on data management practices across various industries. Solving these problems can unlock the full potential of XML data within the context of ORDBMSs, enabling organizations to harness the richness of structured and semi-structured data efficiently and effectively.

Conclusion

The storage of XML data in Object-Relational Database Systems (ORDBMS) represents a dynamic and evolving field with profound implications for data management. It offers a powerful solution to the challenges posed by XML's hierarchical and semi-structured nature within the context of structured relational databases. This integration empowers organizations to efficiently store, query, and manage XML data while ensuring data integrity, scalability, and security.The ability to seamlessly blend XML and relational data enables comprehensive data analysis and reporting, providing a holistic view of organizational information. Moreover, the support for XML-specific query languages, schema evolution, and validation mechanisms ensures that ORDBMSs can adapt to changing XML content requirements over time.As

organizations continue to deal with diverse and complex data sources, the research and practical implementation of XML data storage in ORDBMSs will remain crucial. This field presents ongoing challenges and opportunities for researchers and practitioners alike, driving innovation in data management practices and enhancing our ability to harness the richness of structured and semi-structured data in various domains and applications.

References

1. Mlynkova, I., &Pokorny, J. (2005). XML in the world of (object-) relational database systems. In *Information Systems Development: Advances in Theory, Practice, and Education* (pp. 63-76). Springer US.
2. Shimura, T., Yoshikawa, M., &Uemura, S. (1999). Storage and retrieval of XML documents using object-relational databases. In *Database and Expert Systems Applications: 10th International Conference, DEXA'99 Florence, Italy, August 30–September 3, 1999 Proceedings 10* (pp. 206-217). Springer Berlin Heidelberg.
3. Klettke, M., & Meyer, H. (2001). XML and object-relational database systems enhancing structural mappings based on statistics. In *The World Wide Web and Databases: Third International Workshop WebDB 2000 Dallas, TX, USA, May 18–19, 2000 Selected Papers 3* (pp. 151-170). Springer Berlin Heidelberg.
4. Pardede, E., Rahayu, J. W., &Taniar, D. (2006). Object-relational complex structures for XML storage. *Information and Software Technology*, *48*(6), 370-384.
5. Han, W. S., Lee, K. H., & Lee, B. S. (2003). An XML storage system for object-oriented/object-relational DBMSs. Journal of Object Technology, 2(1), 113-126.
6. Carey, M. J., Florescu, D., Ives, Z. G., Lu, Y., Shanmugasundaram, J., Shekita, E. J., & Subramanian, S. N. (2000, May). XPERANTO: Publishing Object-Relational Data as XML. In *WebDB (Informal Proceedings)* (pp. 105-110).
7. Dittberner, C., Courvoisier, T., &Flach, G. (2001, June). Using Object-relational Database Systems and XML in the Context of an wireless-oriented Multimedia Information System. In International Conference on Web-Age Information Management (pp. 34-45). Berlin, Heidelberg: Springer Berlin Heidelberg.

8. Surjanto, B., Ritter, N., &Loeser, H. (2000, June). XML content management based on object-relational database technology. In Proceedings of the first international conference on web information systems engineering (Vol. 1, pp. 70-79). IEEE.
9. Runapongsa, K., & Patel, J. M. (2002, March). Storing and querying XML data in object-relational DBMSs. In International Conference on Extending Database Technology (pp. 266-285). Berlin, Heidelberg: Springer Berlin Heidelberg.
10. Mo, Y., & Ling, T. W. (2002, December). Storing and maintaining semistructured data efficiently in an object-relational database. In *Proceedings of the Third International Conference on Web Information Systems Engineering, 2002. WISE 2002.* (pp. 247-256). IEEE.
11. Tatarinov, I., Viglas, S. D., Beyer, K., Shanmugasundaram, J., Shekita, E., & Zhang, C. (2002, June). Storing and querying ordered XML using a relational database system. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data* (pp. 204-215).
12. Kudrass, T., & Conrad, M. (2002). Management of XML documents in object-relational databases. In *XML-Based Data Management and Multimedia Engineering—EDBT 2002 Workshops: EDBT 2002 Workshops XMLDM, MDDE, and YRWS Prague, Czech Republic, March 24–28, 2002 Revised Papers 8* (pp. 210-227). Springer Berlin Heidelberg.
13. Kim, J. S., & Lee, K. H. (2002). A Storage Technique for XML Documents based on Object-Relational Database. *Journal of KIISE: Computing Practices and Letters*, *8*(4), 389-398.
14. Subramanian, M., Krishnamurthy, V., & Shores, R. (1999). Performance challenges in object-relational DBMSs. *IEEE Data Eng. Bull.*, *22*(2), 27-31.
15. Schmidt, A., Kersten, M., Windhouwer, M., &Waas, F. (2001). Efficient relational storage and retrieval of XML documents. In *The World Wide Web and Databases: Third International Workshop WebDB 2000 Dallas, TX, USA, May 18–19, 2000 Selected Papers 3* (pp. 137-150). Springer Berlin Heidelberg.