



# Designing Document SQL (DSQL): An Accessible yet Comprehensive Ad-Hoc Querying Frontend for XQuery

*Arijit Sengupta, Wright State University, USA*

*V. Ramesh, Indiana University, USA*

---

## ABSTRACT

*This article presents DSQL, a conservative extension of SQL, as an ad-hoc query language for XML. The development of DSQL follows the theoretical foundations of first order logic, and uses common query semantics already accepted for SQL. DSQL represents a core subset of XQuery that lends well to optimization techniques, while at the same time allows easy integration into current databases and applications that use SQL. The intent of DSQL is not to replace XQuery, the current W3C recommended XML query language, but to serve as an ad-hoc querying frontend to XQuery. Further, the authors present proofs for important query language properties such as complexity and closure. An empirical study comparing DSQL and XQuery for the purpose of ad-hoc querying demonstrates that users perform better with DSQL for both flat and tree structures, in terms of both accuracy and efficiency.*

*Keywords:* Ad-Hoc Querying, Empirical Study, Query Language, Query Language Comparison, Query Optimization, XML, Xquery

---

## INTRODUCTION

XQuery, the query language for XML, originally proposed as early as 2001, currently a 2007 W3C recommendation (Don Chamberlin, Clark, et al., 2001), was ratified as a candidate W3C recommendation late 2005, and became an official W3C recommendation in Feb 2007 (Boag, et al., 2007). With the increase in the popularity of XML as the next generation of documentation representation language for the

hyped Web 2.0 (O'Reilly, 2005), the need for a standard way of retrieving information from XML documents was considered a critical issue, which resulted in the design and eventual recommendation of XQuery. XQuery came from a marriage of two directions of querying: (i) pattern-based languages based on the tree structure of XML documents such as XPath (Clark & DeRose, 1999) and XQL (Robie, Lapp, & Schach, 1998), and (ii) a more logic-oriented approach with conditions and output specifications such as XML-QL (Deutsch, Fernandez, Florescu, Levy, & Suciu, 1998). Although there

DOI: 10.4018/jdm.2009062502

are some attempts towards including XML querying support in SQL, including an effort by the International Standards Organization - SQL-03 (ANSI/ISO, 2003) from the International Standards Organization (ISO), a common decision among XML query language designers was to create a completely new language for the purpose of querying XML data. XQuery is defined as a “full declarative functional programming language, with support for arbitrary levels of recursion and arbitrarily large memory usage”. This is a direction away from previous query language research which tended to ensure the complexity of SQL stayed within reasonable complexity bounds. The problem is that having a full programming language may not be suitable for ad-hoc querying, which will explain why after 7 years of the development of XQuery, it is still not close to the level of popularity as an ad-hoc query language for XML.

Typically, a declarative (as opposed to a procedural) language is one that can specify an expression by declaring the structure and conditions of the intended result, instead of explicitly providing the steps necessary to obtain those results. For example, an SQL query need only specify the output attributes, the input relations, and properties of the output. The advantage of a declarative language is that the query engine can decide what steps to take to generate the output, by considering all optimization possibilities. Some characteristics of XML schema make it possible to write queries using a declarative language. Although XML documents have a complex hierarchical structure, the strong presence of meta-data in XML documents makes it fairly intuitive to write declarative queries based purely on logical combinations of the properties of the intended results. Declarative query languages where the primary focus is on the properties of the result, rather than the process of extracting the result itself, are very suitable for structured data, because they allow the possibility of letting the system optimize the queries instead of relying on the users’ capabilities for writing an efficient query. We present a declarative query language, Document SQL (DSQL) that has the same look and feel as SQL

and was designed by updating the semantics of SQL operations in the structured document domain. At the same time, DSQL was designed such that all queries written in it have equivalent counterparts in XQuery. Thus, by using such a language, users can take advantage of their existing SQL knowledge when writing ad-hoc queries without losing the expressive power of XQuery. In addition to describing the syntax and semantics of the language we also present the results of an experiment that investigates whether a language like DSQL make it possible for users to write more accurate and efficient queries than XQuery.

The rest of the article is organized as follows. Section 2 reviews current research in this area. Section 3 illustrates a data model for representing XML documents. Section 4 introduces the DSQL query language, and Section 5 provides a comparison between DSQL and XQuery. We describe a study comparing DSQL with XQuery in Section 6, discuss the findings of this study in Section 7 and provide some concluding remarks in Section 8.

## LITERATURE REVIEW

Query languages for documents and other hierarchical structures have been a topic of research ever since XML became a force in document representation. W3C (<http://www.w3.org>) initiated a working group for XML Query, for the purpose of designing query languages for document databases, and proposed a set of requirements (D. Chamberlin, Fankhauser, Marchiori, & Robie, 2003) for a query language for XML. A number of query languages came out of this stream of research, including XQL (Robie, et al., 1998), XML-QL (Deutsch, et al., 1998), Quilt (Don Chamberlin, Robie, & Florescu, 2001). All of these efforts were incorporated by the XQuery Working group into the query language XQuery (Don Chamberlin, Clark, et al., 2001). Several other query languages have been designed for XQuery or with the intent of performing structured queries on complex data.

26 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage:

[www.igi-global.com/article/designing-document-sql-dsql/37211?camid=4v1](http://www.igi-global.com/article/designing-document-sql-dsql/37211?camid=4v1)

This title is available in InfoSci-Journals, InfoSci-Journal Disciplines Library Science, Information Studies, and Education. Recommend this product to your librarian:

[www.igi-global.com/e-resources/library-recommendation/?id=2](http://www.igi-global.com/e-resources/library-recommendation/?id=2)

## Related Content

---

### Concurrency Control for Replicated Data in Distributed Real-Time Systems

Sang H. Son, Fengjie Zhang and Buhyun Hwang (1996). *Journal of Database Management* (pp. 12-23).

[www.igi-global.com/article/concurrency-control-replicated-data-distributed/51162?camid=4v1a](http://www.igi-global.com/article/concurrency-control-replicated-data-distributed/51162?camid=4v1a)

### The Expert's Opinion

Mohammad Dadashzadeh (1990). *Journal of Database Administration* (pp. 48-0).

[www.igi-global.com/article/expert-opinion/51081?camid=4v1a](http://www.igi-global.com/article/expert-opinion/51081?camid=4v1a)

### A Constraint Based Fuzzy Object Oriented Database Model

Guy de Tre and Rita de Caluwe (2005). *Advances in Fuzzy Object-Oriented Databases: Modeling and Applications* (pp. 1-45).

[www.igi-global.com/chapter/constraint-based-fuzzy-object-oriented/4806?camid=4v1a](http://www.igi-global.com/chapter/constraint-based-fuzzy-object-oriented/4806?camid=4v1a)

### On the Adaptation of an Agile Information Systems Development Method

Mehmet N. Aydin, Frank Harmsen, Kees van Slooten and Robert A. Stagwee (2005). *Journal of Database Management* (pp. 25-40).

[www.igi-global.com/article/adaptation-agile-information-systems-development/3340?camid=4v1a](http://www.igi-global.com/article/adaptation-agile-information-systems-development/3340?camid=4v1a)