This book pulls together what I learned over 10 years designing databases and building database systems for languages like XML and in using those systems to solve interesting, real-world problems.

XML databases are relatively new, but many of the concepts and techniques applicable for designing XML databases have been around for several years. I have gathered these concepts and techniques, tailored them to XML databases, and utilized the most recent stable technologies to provide a practical framework in which to apply those techniques. A new start-to-finish case study is developed throughout the book, including how to efficiently store the XML data, how to design a schema for XML, how to make the user interface work with other XML technologies, how to query XML, and how to put everything together in a well-designed system architecture.

This book is useful for programmers, database developers, students, system architects, and anyone else who wishes to effectively use, design, or build XML databases. A basic knowledge of XML and databases is assumed, and the focus of this book is on pulling them
together. Some advanced techniques are described in this book and the presentation is fairly dense in those areas.

The book covers:

- How to design a schema for an existing XML DBMS beginning with the concepts of the field being modeled and resulting in compatible schemas for XML documents, relational databases, and object-oriented applications.
- How to store XML data in a relational DBMS, object-oriented DBMS, or flat files, and how to make decisions on which approach to choose.
- How to design a system architecture that contains an XML database, Web server, and user applications.
- How to develop a user interface for XML data accessed via a Web browser or Java application.
- How to query an XML database and what algorithms support XML database querying.
- How to create a native store for an XML DBMS.

In addition, a theoretical foundation is presented for XML databases, querying, and interdocument links.

Some history on how this book came about may be helpful. When I was in graduate school, I heard someone say that it took about 10 years for database technology to go from academic research to industry. I decided that I might be able to get a head start by focusing on the application of database theory to real-world problems. At the same time, the Human Genome Project was starting, and I found it fascinating to contribute to the unique endeavor to understand biologically what makes us human. Along the way, I discovered what worked for databases and what did not.

My dissertation described a way to interconnect data that grew out of ideas in artificial intelligence, hypertext systems, and databases. The premise was that systems of interconnected links could be treated
as a database (or knowledge base), and that well-defined operations could be performed on the somewhat fuzzy entanglement of links. For lack of a better term, I called the connection of interconnected links a Web, the operations on them Spiders, and the whole system a Weave. However, in the early 1990s, there was no practical application for such a bizarre system, other than in artificial intelligence knowledge models, natural language processing, and interesting enough, the very early stages of computational biology.

I decided to begin work on capturing the interconnection of biological information in this system, and went to Baylor College of Medicine as a postdoctoral researcher in the very first computational molecular biology group. There I discovered that the graph-like structure of the links was very similar to the mechanisms biologists were developing to describe the relationship of interactions in the cell. My ideas were refined to support the graph-like interactions in biological data and incorporated into larger database systems.

About that time, another group developed a hypertext system called the World Wide Web that was geared toward exchanging text and images across the Internet, which was gaining in popularity. Although similar to what I was working on because of some of the shared hypertext ideas, its language, HyperText Markup Language (HTML), was geared more toward presentation and less toward data. It was applicable to user interfaces for a scientific database, but not applicable at all for capturing scientific data.

At the same time, the Human Genome Project was becoming more visible to biotech and pharmaceutical companies who started hiring almost every person in the very small, very new, and esoteric field of using computers to manage the rapidly growing biology data, called bioinformatics. I went to industry and began integrating what I had learned with even larger relational databases and delivering that data via Web browsers. Then, in 1998, the World Wide Web Consortium proposed a recommendation for a HTML-like language for data, called Extensible Markup Language (XML), which provided a flexible syntax for representing hierarchical data.
Since then, I have been adopting XML as the language for representing data and integrating that with commercial relational database management systems (DBMSs) in the framework I had been using for 10 years. This book pulls together what I learned during that time. In particular, I have strived to include techniques from databases that are particularly useful for XML and may not be accessible in other resources.

As it is rare for a technical book to be read from beginning to end, the following chapter groups may be useful. To:

• Effectively use an existing XML DBMS, see Chapters 1, 2, 5, 7, and 10.
• Purchase an XML DBMS or components of a solution, see Chapter 6. Chapters 1, 4, and 5 provide background material for understanding Chapter 6.
• Use an existing flat file, relational, or object-oriented DBMS with XML, see Chapters 2, 3, 4, 5, 7, 8, and 10.
• Obtain a manager’s overview of XML database technology, see Chapters 1, 6, and 10. Chapters 2 and 5 provide background material for understanding Chapter 10. Chapter 4 provides some project alternatives.
• Create or enhance an XML DBMS, see Chapters 3, 4, 8, and 9. Other chapters provide background material.
• Learn all about XML databases, see all the chapters.