

Contents

Preface	xix	
Acknowledgments	xxxv	
Part I	What Is XML?	1
Chapter 1	Information Modeling with XML	3
1.1	Introduction	3
1.2	XML as an Information Domain	4
1.3	How XML Expresses Information	5
1.4	Patterns in XML	6
1.5	Common XML Information-Modeling Pitfalls	9
1.5.1	Attributes Used as Data Elements	10
1.5.2	Data Elements Used as Metadata	11
1.5.3	Inadequate Use of Tags	12
1.6	A Very Simple Way to Design XML	14
1.7	Conclusion	16

Part II Native XML Databases 19

Chapter 2	Tamino—Software AG's Native XML Server	21
2.1	Introduction	21
2.2	Tamino Architecture and APIs	22
2.3	XML Storage	23
2.3.1	Collections and Doctypes	25
2.3.2	Schemas	26
2.3.3	Access to Other Databases—Tamino X-Node	28
2.3.4	Mapping Data to Functions—Tamino X-Tension	31
2.3.5	Internationalization Issues	34
2.3.6	Indexing	35
2.3.7	Organization on Disk	36
2.4	Querying XML	37
2.4.1	Query Language—Tamino X-Query	37
2.4.2	Sessions and Transactions	38
2.4.3	Handling of Results	38
2.4.4	Query Execution	39
2.5	Tools	39
2.5.1	Database Browsing	39
2.5.2	Schema Editing	39
2.5.3	WebDAV Access	40
2.5.4	X-Application	41
2.6	Full Database Functionality	42
2.7	Conclusion	42
Chapter 3	eXist Native XML Database	43
3.1	Introduction	43
3.2	Features	44
3.2.1	Schema-less XML Data Store	44
3.2.2	Collections	44
3.2.3	Index-Based Query Processing	44
3.2.4	Extensions for Full-Text Searching	45
3.3	System Architecture Overview	45
3.3.1	Pluggable Storage Backends	45
3.3.2	Deployment	46
3.3.3	Application Development	46
3.4	Getting Started	47
3.5	Query Language Extensions	47
3.5.1	Specifying the Input Document Set	48
3.5.2	Querying Text	48
3.5.3	Outstanding Features	49

3.6 Application Development	50
3.6.1 Programming Java Applications with the XML:DB API	50
3.6.2 Accessing eXist with SOAP	52
3.6.3 Integration with Cocoon	54
3.7 Technical Background	56
3.7.1 Approaches to Query Execution	57
3.7.2 Indexing Scheme	58
3.7.3 Index and Storage Implementation	61
3.7.4 Query Language Processing	63
3.7.5 Query Performance	65
3.8 Conclusion	67
 Chapter 4 Embedded XML Databases	 69
4.1 Introduction	69
4.2 A Primer on Embedded Databases	70
4.3 Embedded XML Databases	71
4.4 Building Applications for Embedded XML Databases	73
4.4.1 Overview of Berkeley DB XML	73
4.4.2 Configuration	74
4.4.3 Indexing and Index Types	75
4.4.4 XPath Query Processing	78
4.4.5 Programming for Transactions	80
4.4.6 Two-Phase Locking and Deadlocks	81
4.4.7 Reducing Contention	84
4.4.8 Checkpoints	85
4.4.9 Recovery Processing after Failures	86
4.5 Conclusion	87
 Part III XML and Relational Databases	 89
 Chapter 5 IBM XML-Enabled Data Management Product Architecture and Technology	 91
5.1 Introduction	91
5.2 Product and Technology Offering Summaries	92
5.2.1 DB2 Universal Database	92
5.2.2 Information Integration Technology	92
5.3 Current Architecture and Technology	93
5.3.1 Shared Architecture and Technology	93
5.3.2 XML Extender Architecture	94
5.3.3 XML Extender Technology	95
5.3.4 Using Both XML Collections and XML Columns	100
5.3.5 Transforming XML Data	101

5.3.6 Searching, Parsing, and Validating XML Data	102
5.3.7 XML Extender Federated Support	102
5.3.8 SQL XML Support Architecture	102
5.3.9 SQL XML Support Technology	104
5.3.10 Data Management Web Services Architecture	107
5.3.11 Data Management Web Services Technology	107
5.3.12 Information Integration-Specific Architecture and Technology	111
5.4 Future Architecture and Technology	113
5.4.1 The Vision	113
5.4.2 Application Interface, Data Type, and API Goals	113
5.4.3 Storage, Engine, and Data Manager Goals	116
5.4.4 Why Support Both XML and Relational Storage in One System?	118
5.4.5 Why Not Object-Relational Long Term?	119
5.4.6 Impacted Technology Areas	119
5.5 Conclusion	120
Notices	120
 Chapter 6 Supporting XML in Oracle9i	 123
6.1 Introduction	123
6.2 Storing XML as CLOB	126
6.2.1 Using CLOB and the OracleText Cartridge	126
6.2.2 Search Predicates in OracleText	127
6.2.3 XML-Specific Functionality	129
6.2.4 Prerequisites	130
6.3 XMLType	132
6.3.1 Object Type XMLType	132
6.3.2 Processing of XMLType in Java	135
6.4 Using XSU for Fine-Grained Storage	135
6.4.1 Canonical Mapping	136
6.4.2 Retrieval	139
6.4.3 Modifications	141
6.5 Building XML Documents from Relational Data	144
6.5.1 SQL Functions existsNode and extract	144
6.5.2 The SQL Function SYS_XMLGen	144
6.5.3 The SQL Function SYS_XMLAgg	146
6.5.4 PL/SQL Package DBMS_XMLGen	148
6.5.5 Web Access to the Database	149
6.5.6 The Principle of XSQL	149
6.5.7 Posting XML Data into the Database	151
6.5.8 Parameterization	152
6.5.9 Servlet Invocations	154
6.6 Special Oracle Features	155
6.6.1 URI Support	155
6.6.2 Parsers	160

6.6.3 Class Generator	161
6.6.4 Special Java Beans	162
6.7 Conclusion	163
Chapter 7 XML Support in Microsoft SQL Server 2000	165
7.1 Introduction	165
7.2 XML and Relational Data	165
7.3 XML Access to SQL Server	166
7.3.1 Access via HTTP	167
7.3.2 Using the XML Features through SQLOLEDB, ADO, and .NET	169
7.4 Serializing SQL Query Results into XML	170
7.4.1 The Raw Mode	172
7.4.2 The Auto and Nested Modes	173
7.4.3 The Explicit Mode	174
7.5 Providing Relational Views over XML	176
7.6 SQLXML Templates	180
7.7 Providing XML Views over Relational Data	182
7.7.1 Annotated Schemata	182
7.7.2 Querying Using XPath	183
7.7.3 Updating Using Updategrams	185
7.7.4 Bulk Loading	186
7.8 Conclusion	187
Chapter 8 A Generic Architecture for Storing XML Documents in a Relational Database	189
8.1 Introduction	189
8.2 System Architecture	192
8.2.1 Installing Xerces	192
8.3 The Data Model	193
8.3.1 DOM Storage in Relational Databases	195
8.3.2 The Nested Sets Model	195
8.4 Creating the Database	196
8.4.1 The Physical Data Model	197
8.4.2 Creating User-Defined Data Types	198
8.4.3 Creating the Tables	198
8.4.4 Serializing a Document out of the Repository	206
8.4.5 Building an XML Document Manually	211
8.5 Connecting to the Repository	213
8.5.1 The xmlrepDB Class	213
8.6 Uploading XML Documents	220
8.6.1 The xmlrepSAX Class	220
8.6.2 Stored Procedures for Data Entry	225

8.6.3 The uploadXML Class	231
8.6.4 The extractXML Class	245
8.7 Querying the Repository	248
8.7.1 Ad Hoc SQL Queries	248
8.7.2 Searching for Text	252
8.7.3 Some More Stored Procedures	252
8.7.4 Generating XPath Expressions	254
8.8 Further Enhancements	257
8.9 Conclusion	258
Chapter 9 An Object-Relational Approach to Building a High-Performance XML Repository	259
9.1 Introduction	259
9.2 Overview of XML Use-Case Scenario	261
9.3 High-Level System Architecture	262
9.4 Detailed Design Descriptions	268
9.5 Conclusion	286
Part IV Applications of XML	289
Chapter 10 Knowledge Management in Bioinformatics	291
10.1 Introduction	291
10.2 A Brief Molecular Biology Background	293
10.3 Life Sciences Are Turning to XML to Model Their Information	296
10.4 A Genetic Information Model	300
10.5 NeoCore XMS	308
10.6 Integration of BLAST into NeoCore XMS	311
10.6.1 Sequence Search Types	313
Conclusion	319
Chapter 11 Case Studies of XML Used with IBM DB2 Universal Database	321
11.1 Introduction	321
11.2 Case Study 1: "Our Most Valued Customers Come First"	322
11.2.1 Company Scenario	322
11.2.2 How This Business Problem Is Addressed	322
11.2.3 Future Extensions	326
11.3 Case Study 2: "Improve Cash Flow"	327
11.3.1 Company Scenario	327
11.3.2 How This Business Problem Is Addressed	328
11.3.3 Future Extensions	333
11.4 Conclusion	334
Notices	334

Chapter 12	The Design and Implementation of an Engineering Data Management System Using XML and J2EE	337
12.1	Introduction	337
12.2	Background and Requirements	338
12.3	Overview	339
12.3.1	Security Service	339
12.3.2	Query Service	340
12.3.3	Image Query Service	341
12.3.4	Print Service	341
12.4	Design Choices	341
12.4.1	Using XML in OAI	341
12.4.2	Conversion of XML Input into Objects	343
12.4.3	Conversion of Database Data into XML	344
12.4.4	Conversion of Image Data into XML	344
12.4.5	Database Access	345
12.4.6	Validation	347
12.5	Future Directions	349
12.5.1	XSLT	349
12.5.2	Web Services	349
12.5.3	Mass Transfer Capability	350
12.5.4	Messaging	350
12.6	Conclusion	351
Chapter 13	Geographical Data Interchange Using XML-Enabled Technology within the GIDB System	353
13.1	Introduction	353
13.2	GIDB METOC Data Integration	356
13.2.1	Background	356
13.2.2	Implementation	357
13.3	GIDB Web Map Service Implementation	363
13.4	GIDB GML Import and Export	368
13.5	Conclusion	373
Chapter 14	Space Wide Web by Adapters in Distributed Systems Configuration from Reusable Components	375
14.1	Introduction	375
14.2	Advanced Concept Description: The Research Problem	376
14.2.1	Future Supporting Communications Satellites Constellations	378
14.3	Integration of Components with Architecture	383
14.4	Example	384
14.5	Future Generation NASA Institute for Advanced Concepts, Space Wide Web Research, and Boundaries	387

14.6 Advanced Concept Development	388
14.6.1 The Research Approach	388
14.6.2 The Research Tasks	389
14.7 Conclusion	400
 Chapter 15 XML as a Unifying Framework for Inductive Databases	 401
15.1 Introduction	401
15.2 Past Work	403
15.2.1 Extracting and Evaluating Association Rules	404
15.2.2 Classifying Data	408
15.2.3 Inductive Databases	413
15.2.4 PMML	416
15.3 The Proposed Data Model: XDM	419
15.3.1 Basic Concepts	419
15.3.2 Classification with XDM	423
15.3.3 Association Rules with XDM	435
15.4 Benefits of XDM	447
15.5 Toward Flexible and Open Systems	449
15.6 Related Work	450
15.7 Conclusion	452
 Chapter 16 Designing and Managing an XML Warehouse	 455
16.1 Introduction	455
16.1.1 Why a View Mechanism for XML?	455
16.1.2 Contributions	456
16.1.3 Outline	456
16.2 Architecture	457
16.3 Data Warehouse Specification	458
16.3.1 View Model for XML Documents	458
16.3.2 Graphic Tool for Data Warehouse Specification	461
16.4 Managing the Metadata	462
16.4.1 Data Warehouse	462
16.4.2 View Definition	463
16.4.3 Mediated Schema Definition	466
16.5 Storage and Management of the Data Warehouse	466
16.5.1 The Different Approaches to Storing XML Data	467
16.5.2 Mapping XML to Relational	467
16.5.3 View Storage	468
16.5.4 Extraction of Data	469
16.6 DAWAX: A Graphic Tool for the Specification and Management of a Data Warehouse	470
16.6.1 Data Warehouse Manager	470
16.6.2 The Different DAWAX Packages	470

16.7 Related Work	472
16.7.1 Query Languages for XML	472
16.7.2 Storing XML Data	472
16.7.3 Systems for XML Data Integration	472
16.8 Conclusion	473
Part V Performance and Benchmarks	475
Chapter 17 XML Management System Benchmarks	477
17.1 Introduction	477
17.2 Benchmark Specification	478
17.3 Benchmark Data Set	479
17.3.1 Benchmark Queries	480
17.4 Existing Benchmarks for XML	482
17.4.1 The XOO7 Benchmark	482
17.4.2 The XMach-1 Benchmark	488
17.4.3 The XMark Benchmark	491
17.5 Conclusion	497
Chapter 18 The Michigan Benchmark: A Micro-Benchmark for XML Query Performance Diagnostics	499
18.1 Introduction	499
18.2 Related Work	501
18.3 Benchmark Data Set	502
18.3.1 A Discussion of the Data Characteristics	502
18.3.2 Schema of Benchmark Data	505
18.3.3 Generating the String Attributes and Element Content	507
18.4 Benchmark Queries	508
18.4.1 Selection	509
18.4.2 Value-Based Join	514
18.4.3 Pointer-Based Join	514
18.4.4 Aggregation	515
18.4.5 Updates	515
18.5 Using the Benchmark	516
18.6 Conclusion	517
Chapter 19 A Comparison of Database Approaches for Storing XML Documents	519
19.1 Introduction	519
19.2 Data Models for XML Documents	520
19.2.1 The Nontyped DOM Implementation	521
19.2.2 The Typed DOM Implementation	522

19.3 Databases for Storing XML Documents	523
19.3.1 Relational Databases	524
19.3.2 Object-Oriented Databases	528
19.3.3 Directory Servers	528
19.3.4 Native XML Databases	533
19.4 Benchmarking Specification	533
19.4.1 Benchmarking a Relational Database	534
19.4.2 Benchmarking an Object-Oriented Database	534
19.4.3 Benchmarking a Directory Server	535
19.4.4 Benchmarking a Native XML Database	535
19.5 Test Results	536
19.5.1 Evaluation of Performance	536
19.5.2 Evaluation of Space	539
19.5.3 Conclusion	539
19.6 Related Work	540
19.6.1 Studies in Storing and Retrieving XML Documents	541
19.6.2 XML and Relational Databases	541
19.6.3 XML and Object-Relational Databases	542
19.6.4 XML and Object-Oriented Databases	543
19.6.5 XML and Directory Servers	543
19.6.6 Benchmarks for XML Databases	544
19.6.7 Guidelines for Benchmarking XML Databases	545
19.7 Summary	546
 Chapter 20 Performance Analysis between an XML-Enabled Database and a Native XML Database	 547
20.1 Introduction	547
20.2 Related Work	549
20.3 Methodology	550
20.4 Database Design	551
20.5 Discussion	555
20.6 Experiment Result	556
20.6.1 Database Size	556
20.6.2 SQL Operations (Single Record)	558
20.6.3 SQL Operations (Mass Records)	560
20.6.4 Reporting	561
20.7 Conclusion	563
 Chapter 21 Conclusion	 567
 References	 571

Contributors	587
Editors	587
Chapter 1: Information Modeling with XML	588
Chapter 2: Tamino—Software AG's Native XML Server	588
Chapter 3: eXist Native XML Database	589
Chapter 4: Embedded XML Databases	589
Chapter 5: IBM XML-Enabled Data Management Product Architecture and Technology	589
Chapter 6: Supporting XML in Oracle9i	590
Chapter 7: XML Support in Microsoft SQL Server 2000	590
Chapter 8: A Generic Architecture for Storing XML Documents in a Relational Database	590
Chapter 9: An Object-Relational Approach to Building a High-Performance XML Repository	591
Chapter 10: Knowledge Management in Bioinformatics	591
Chapter 11: Case Studies of XML Used with IBM DB2 Universal Database	592
Chapter 12: The Design and Implementation of an Engineering Data Management System Using XML and J2EE	592
Chapter 13: Geographical Data Interchange Using XML-Enabled Technology within the GIDB System	593
Chapter 14: Space Wide Web by Adapters in Distributed Systems Configuration from Reusable Components	595
Chapter 15: XML as a Unifying Framework for Inductive Databases	595
Chapter 16: Designing and Managing an XML Warehouse	596
Chapter 17: XML Management System Benchmarks	596
Chapter 18: The Michigan Benchmark: A Micro-Benchmark for XML Query Performance Diagnostics	597
Chapter 19: A Comparison of Database Approaches for Storing XML Documents	598
Chapter 20: Performance Analysis between an XML-Enabled Database and a Native XML Database	598
Index	601