

Index

&= operator, 49, 65
 |= operator, 49, 65
 % wildcard, 128
 _ wildcard, 128
 3-tuple identifiers, 58
 4SuiteServer, 477

A

Abstract DTD, 473
 Abstract view, 422
 AbstractFeatureType element, 368
 accessPredicate element, 31
 ACS (attitude control system), 380
 Ad hoc queries, 170, 248–252, 345
 Ad hoc SQL, 225
 Adabas, 23
 Adapters, 377–378

- auto-configuration, 388
- automatically generated, 391
- building with generated mapping code, 392–393

drill-down methods, 388

- as executables, 392
- formalizing notation describing, 392
- interaction support for communication, 389
- interpreting specifications, 392
- quality of service support, 395
- to support component interactions, 389–392

 Adaptive spacecraft software, 380–382
 AddDocument() function, 263, 266, 284
 addNode method, 162
 address element, xxv
 //address expression, 280
 Address_Type object, 137
 AddSchema() function, 263, 266
 add_..._section procedures, 131–132
 Adenine (A), 293
 Adjacency List Model, 195
 ADLs (Architecture Description Languages), 389, 393–394
 ADO, 167, 169–170
 AGE attribute, 427, 432
 @AGE expression, 427
 Aggregation queries, 515

- Aliases, starting with at sign (@), 136
- AllChildren table, 468
- Alternative splicing of the mRNA, 304
- Amino acid sequence, 312
- Amino acids, 294–295, 295, 317
- Ancestor nesting in ancestor-descendant selection queries, 512
- Ancestor-descendant complex pattern selection queries, 514
- Ancestor-descendant selection queries, 512
- and logical operator, 158
- AND (&) operator, 128
- Annotating schemas, 27–28, 182–183
- Apache, 149
- Apache Web site, 192
- APB-1 benchmark, 478–479
- APIs
 - based on XQuery, 114–115
 - Tamino XML Server, 22–23
- appinfo element, 34
- APPLICANT element, 430, 433
- Application-level benchmarks, 500
- Applications
 - ADLs (Architecture Description Language), 393–394
 - building for embedded XML databases, 73
 - component integration, 375–376
 - components, 390
 - connectors, 390
 - constructing from reusable components, 388
 - DB2 support, 111–112
 - DOM-compliant, 194
 - easily moving across environments, 388
 - embedding XML database, 72
 - eXist development of, 50–56
 - local joins of table data, 112
 - rapid development of, 41
 - reclaiming log records space, 86
- Approval_Status.xml document, 332–333
- APPROVED flag, 332
- AreaFeatureType type, 368, 372
- arithmetic operators, 158
- AS3AP benchmark, 478
- ASSOCIATING clause, 407
- Association rules
 - aggregation, 406
 - binary model, 414
 - confidence, 405, 414
 - defining, 405, 413–414
 - EVALUATE RULE operator, 406–408
 - extracting and evaluating, 404–408
 - extraction of (XDM), 436–438
 - frequency, 414
 - MINE RULE operator, 405–406
 - PMML model, 418
 - support, 405
 - XDM, 435–447
- AssociationModel element, 418
- AssociationRules element, 418
- AssociationRulesData XDM database, 435–447
- ASSOCIATION-RULE-SET node, 438, 440
- atomic value, 271
- AT&T Labs, 541
- Attribute table, 468
- attribute table, 525
- attribute_name table, 201, 226–227
- Attributes, 5, 6, 307
 - applying to all data elements, 7
 - data portion, 10
 - long values, 202
 - mapping names to node identifiers, 62
 - metadata, 7, 10
 - names as tags, 145
 - names exceeding maximum length, 235
 - object types, 136
 - scope, 8
 - starting with at sign (@), 136
 - storing names, 201
 - units, 14
 - used as data elements, 10–11

attribute_value_leaf table, 202, 227
 author element, 466
 authorspublications view, 466
 AUTO_SECTION_GROUP section group, 131
 Axis SOAP toolkit, 52

B

Base pairs, 293
 BaseType element, 505–506
 Basket analysis applications, 402
 BDGP (Berkeley Drosophila Genome Project), 299
 BEA, 4
Benchmark Handbook, The (Gray), 478
 Benchmarks

- application-level, 500
- currently proposed, 499
- data set, 479–482, 500
- database disk space evaluation, 539
- database results, 539–540
- database storage evaluation, 536–538
- design of, 479
- directory servers, 535
- existing XML, 482–497
- generating synthetic XML data, 501
- guidelines for databases, 545
- native XML databases, 535
- object-oriented databases, 534–535
- portability, 478
- queries, 480–482
- relational databases, 534
- relevance, 478
- scalability, 478
- simplicity, 478
- specification, 478–479
- test results of database storage, 536–540
- Wisconsin benchmark, 500
- X007, 482–488, 501–502
- XMach-1, 488–490, 501
- XMark, 491–497, 501
- XML data, 499–501
- XML databases, 544–545, 549–550
- XML document storage, 533–535

 Berkeley DB library, 74
 Berkeley DB tables, 74
 Berkeley DB XML applications recovery

- processing after failures, 86

 Berkeley DB library, 74
 Berkeley DB tables, 74
 configuration settings, 75
 container within environment, 75
 creating index on collection, 76

- environment, 75
- indexing containers, 75–76
- indexing strategies, 76–77

 Java language bindings, 74
 opening and using container, 73
 overview, 73–74
 running query against collection, 79–80
 statistical index information, 74
 threading packages, 85
 transactional updates, 80–81
 XPath query processing, 78–80

Best-effort service networks, 384
 bib2html.xml file, 54
 Binary association rules, 414
 Bioengineering Institute at the University of Auckland, 299
 Bioinformatics, 291, 296, 311
 Biological information complexity, 307
 BIOML (BIOPolymer Markup Language), 298
 BIOML Web site, 298
 BIOXML Web site, 299
 BLAST (Basic Local Alignment Search Tool), 312

- complex query using tBLASTx option, 317
- integrating into NeoCore XMS, 311–319
- tBLASTx search, 317–318

BLASTp, 316
 blocking operation, 264–265
 BODY node, 440
 BODY-SCHEMA element, 437, 443
 boundedBy element, 368
 Broader Term operator, 128
 BSML (Bioinformatic Sequence Markup Language), 298
 B+-tree classes, 61
 BUCKY benchmark, 479
 Bulkload object, 186–187
 Bulkloading, 186
 Business information systems, 261–262
 Business process integration support, 112

C

Cache sizing, 84
 California Institute of Technology Web site, 299
 Callback functions, 33
 Capabilities XML document, 363
 CARD element, 437
 CAR-INSURANCE element, 424, 426, 430
 Carriage return/newline characters, 239
 CAR-TYPE attribute, 427, 432
 @CAR-TYPE expression, 427
 C/C++ XDK (XML Development Kit), 160
 CDATA, 220
 CDATA sections, 204, 239
 cdata_leaf table, 204, 227, 229
 CDATASection table, 524–526
 Celera Genomics, 292
 Celko, Joe, 195
 Cell type category, 303
 CellML, 299
 CellML Web site, 299
 Center for Computational Biology, 310
 Center for Computational Biology Web site, 310
 Character data, whitespaces as, 238–239

characters() method, 237–238
 Checkpoints, 85–86
 Chemical signaling, 295
 Children table, 468
 Cities table, 31
 City doctype, 29, 33
 City element, 31
 /City/Monument/Name query, 38–39
 Class Generator, 161–162
 Classification
 rule set, 409–410
 with XDM, 423–435
 Classification tree, 402, 409
 ClassificationData XDM database, 423–425
 CLASSIFICATION-MODEL element, 431
 Classification_Model table, 411
 CLASSIFICATION-TREE element, 429
 CLASSIFICATION-UNIT element,
 426, 427, 431
 Classified_Applicants table, 412
 CLASSIFIED-NEW-APPLICANTS element, 433
 Classifiers, 401–402
 Classifying data
 classification rule set, 409–410
 classification tree, 409
 for data mining, 408–413
 DEFAULT rule, 410
 test phase, 408, 409, 412–413
 training phase, 408, 409–411
 training set, 408–409
 CLASS-PARAM element, 427, 429
 CLASSROOM_RECORD table, 329
 Client/server databases, 71, 72, 82
 checkpoints, 85–86
 indexing, 76
 optimizing queries, 76
 recovery processing after failures, 86
 CLOB document, maintaining link to, 160
 CLOB system data type, 103
 CLOB tables, maintaining, 163

- CLOBs
 - comparisons, 126
 - storing XML as, 126–132
 - substring search, 126
- ClobTab table, 160
- Clustering applications, 402
- Clustering procedures, 402
- Cocoon, 43, 47, 54–56
- Cocoon Web site, 54
- Codons, 295, 314
- Collation facet, 34
- collection() function, 48
- Collections
 - doctypes, 25
 - documents included inquiry evaluation, 48
 - eXist, 44
 - mapping, 136
 - searching with indexes, 75–76
 - XML Schema descriptions, 25
 - XML:DB API, 50
- collections.dbx file, 61
- Collins, Francis, 292
- Colorado University Health Science Center, 310
- Colorimeters, 5
- command-line testing, 213–214
- comment_leaf table, 204, 227–228
- comment() method, 242–244
- Commit, 33
- Communications satellite
 - constellations, 379–380
 - spinning (at night) mode, 381
- COMPARE() function, 273–274
- Complete view, 422
- Complex pattern selection queries, 513
- Component integration, 375–376
- Components, 390
 - adaptation, 381–382
 - adapters, 377–378
 - configuration language, 388
 - difficulty of reuse, 378
 - integration with architecture, 383–384
 - multiple interactions between, 391
 - as query terms, 15
 - reducing cost of development, 376–377
 - reusability, 376, 377–378, 392
 - services, 390
 - solving interface problems, 383
- Composite elements, xxv
- Composite views, 458–459
- Compositors, xxv
- Computational Pharmacology Group, 310
- Concrete DTD, 473
- Condensed structure index, 36
- CONDITION element, 429
- condition element, 464
- Conditions, describing, 464
- Confidence, 405, 414
- CONFIDENCE element, 440, 446
- Configuration language, 388, 389–392
- Connect request, 38
- connection element, 462
- Connections, multiple interactions between
 - components, 391
- connect() method, 214–215
- Connectors, 390
- Constraints, xxvii
- Container-managed beans, 346
- Containers
 - common queries, 75
 - indexing, 75–77
 - opening and using, 73–74
- CONTAINS function, 127–131
- contains operator (~), 37
- Content, differentiating between types, 237–238
- Contention, 84–85
- Context list, 7–8
- Control actors, 304
- <cookies> tag, 154
- ©right entity reference, 238
- CORBA, 383, 397

CREATE FEDERATED VIEW statement, 112
 CREATE NICKNAME statement, 112
 CREATE SERVER statement, 112
 CREATE WRAPPER statement, 112
 createXML method, 132, 133, 135
 create_xmlrep_db.sql script, 192, 197, 225
 Crick, Frances, 291
 CTXSYS.context index type, 131
 Cursor to walk over results set, 256–257
 Custom complex types, xxv
 CUSTOMER attribute, 443
 Customer_Type object type, 139
 CustOrd.xdr file, 184
 Cytosine (C), 293

D

DAD (data access definition) files, 96, 107
 relational database node mapping, 97,
 99–100
 SQL node mapping, 97–98
 DADS file, 110–111
 DADX files, 107, 108, 110
 Data
 adding context to, 5
 application view, 113
 classifying for data mining, 408–413
 complete context, 5–6
 exchanging with XML, 332
 heterogeneity, 387
 heterogeneous federated access, 111–112
 mapping to functions, 31–33
 storage, 113
 storing with leafing mechanism, 203
 stronger typing constraints, xxv
 without context, 298
 Data elements, 307
 attributes used as, 10–11
 describing, 5
 inappropriate grouping, 12–14
 interpreting, 5–6
 parts of forms to file in, 14
 proper use of, 6–7
 something about, 5, 6
 truly representing data, 14
 used as metadata, 11–12
 Data engines, 116
 Data entry, 225–231
 Data management, xxix–xxx
 Web services technology, 107–111
 Data mining, 450
 background knowledge, 448
 classifying data, 408–413
 extracting and evaluating association
 rules, 404–408
 inductive databases, 413–416, 451
 integrating algorithms and databases,
 450–451
 PMML (Predictive Model Mark-up
 Language), 416–419
 XDM (XML for Data Mining), 419–449
 Data models
 DOM (Document Object Model), 193–194
 XML documents, 520–523
 Data space, 36
 Data store, 61, 116
 DATA table, 272
 Data types for bioinformatics data, 296
 Data warehouse applications, 401
 DATA-AND-RULE-SET element, 446
 Database attribute, 424
 Database server, as information repository for
 user interface, 189
 Database systems, 69
 switching between alternative, 47
 two-phase locking, 81
 write-ahead logging, 81–82
 DatabaseManager class, 50
 DatabaseManager.getCollection() method, 52

- DatabaseManager.registerDatabase method, 50
- Databases
 - accessing, 166
 - automatically building from patterns, 9
 - benchmark results, 539–540
 - browsing, 39
 - cache sizing, 84
 - collections, 44
 - converting data into XML, 344
 - disk space evaluation, 539
 - extracting data patterns from, 401–402
 - full functionality, 42
 - indexing, 35, 277
 - inductive, 402, 413–416
 - integration with data-mining tools, 403–416
 - native XML, xxix–xxx, 556–558
 - OAI access, 345–347
 - posting XML documents into, 151–152
 - root collection, 48
 - security, 42
 - storage evaluation, 536–538
 - for storing XML documents, 523–533
 - Tamino XML Server access, 28–31
 - test results for storage benchmarks, 536–540
 - traditional, 308
 - Transact-SQL statement, 170
 - typed DOM (Document Object Model)
 - implementation, 527
 - Web access to, 149–155
 - XML data management, xxix
 - XML-related database size, 556–558
- Data-centric documents, xxv, 21, 45
 - fine-grained storage of table attributes, 124–125
 - queries, 37
 - relational databases, 124–125
- Data-centric XML, xxiii–xxv
- DataDictionary element, 418
- Data-mining applications, 401–402
- Data-mining tools, integration with
 - databases, 403–416
- DATA-SECTION element, 442
- DataSet object, 170
- Datastore, 129
- Datawarehouse component, 468–469
- DAWAX, 461–462, 470–473
 - dawax package, 471
 - dawax.interrogation package, 471
 - dawax.management package, 471
 - dawax.specification package, 471
 - dawax.xml.documentmodel package, 471
 - dawax.xml.parser package, 471
- /db collection, 48
- DB2, 69, 92
 - access and support for unstructured data, 112–113
 - application support, 111–112
 - data management Web services
 - architecture, 107
 - data management Web services
 - technology, 107–111
 - database managers, 103
 - DB2 family and Informix IDS data
 - sources, 102
 - federated systems, 103
 - heterogeneous federated data access, 111–112
 - "Improve Cash Flow" case study, 327–334
 - "Our Most Valued Customers Come First" case study, 322–327
 - processing support, 94
 - relational data stored in, 325
 - searching XML documents, 327
 - shared architecture and technology, 92
 - solution to problem, 322–326
 - SQL XML support, 92, 102–103
 - Web service support, 92
 - WORF (Web services Object Runtime Framework) tools, 107
 - XML Extender, 92, 94–95, 323, 324, 477

- DB2 databases
 - SQL statements executed on relational data, 107
 - student and classroom record system, 328
- DB2 for Linux, 92
- DB2 for UNIX, 92
- DB2 for Windows, 92
- DB2 SQL Reference*, 104
- DBAccess bean, 163
- DBBroker interface, 46, 63
- DBDOM, 477
- DBMS hierarchical concepts, 272–273
- DBMS_XMLQuery, 139
- DBMS_XMLSave, 139
- DBMX_XMLGen PL/SQL package, 148–149
- DBMX_XMLGen.ctxHandle class, 148
- DBUri path, 157
- DBUriType objects, 158–160
- DBUriType subtype, 156–157
- DBViewer bean, 163
- dbXML, 477
- dbXML project, 61
- DCOM (Distributed Component Object Model), 383
- dc:title element, 56
- DDBJ Web site, 292
- Deadlocks, 81–84
- DEFAULT rule, 410
- DELETE method, 23, 134
- deleteXML method, 142
- Department of Education reimbursement request Web service, 328–333
- Derived XDM data item, 432–433
- description element, 368
- Detailed view, 422
- DEVICES, 397–398
- Digital-mapping database system, 353
- DIPCS (Distributed Interprocess Communication System), 397–399
- Directory servers
 - benchmarks, 535
 - nontyped DOM implementation, 530–532
 - parent-child relationships of DOM tree, 530–531
 - typed DOM implementation, 530
 - XML and, 543–544
 - XML documents, 528–533
- disconnect() method, 215
- Distance education technology, 398
- Distributed applications, 388, 399–400
- Distributed instructional software, 384
- DIVIDE BY clause, 407
- DLA (Defense Logistics Agency), 338
- DMAP (Digital Mapping, Charting and Geodesy Analysis Program), 353
- DMG (Data Mining Group), 416
- DMG Web site, 416, 452
- DML (Data Manipulation Language), 152
- DNA (Deoxyribonucleic Acid), 293–294, 314–315
- DNA double helix, 291
- DNA polymerase, 295
- DNA sequence model, 307–308
- DNA sequences
 - determining set of nucleotides making up, 314
 - different types of information, 308
 - genes, 303
 - tBLASTx search, 317–318
- <dna_sequence> element, 303, 308
- doc table, 199, 234
- DocTab table, 127
- Doctypes, 36
 - dynamically created, 25
 - open or closed content, 28
- <Document> element, 484
- Document queries, 480
- document table, 468
- Document tree, 57–59
- Document-centric documents, 21
 - addressing human users, 45
 - DTDs (Document Type Definitions), xxv
 - LOBs (Large Objects), 124

- queries, 37
 - relational databases, 124
 - Document-centric XML, xxiii–xxiv
 - document() function, 48
 - Documents
 - attributes, 26
 - complex, proprietary standards, xxii
 - data-centric, xxv, 21
 - doctype, 25
 - document-centric, 21
 - grouping, 25
 - hybrid, xxiii
 - maximum size of indexed, 59–60
 - ordered sequential access to nodes, 62
 - printing, 341
 - rapid serialization, 195
 - root element, 25
 - subtrees, 26
 - text nodes of elements, 251
 - against which expression evaluated, 48
 - DoD (Department of Defense), 338
 - DOM (Document Object Model), xxvi–xxvii, 193–194, 310
 - attributes, 201
 - interface, 160
 - nontyped implementation, 521–522
 - object-oriented databases, 528
 - translating XML document into in-memory tree structure, 343
 - typed implementation, 522–523
 - DOM API and repository, 258
 - DOM applications, memory limitations, 194
 - DOM Builder bean, 162
 - DOM (Document Object Model) parsers, xxvi, 343
 - DOM (Document Object Model) tree, xxvi, 359–360
 - DOM-compliant applications, 194
 - dom.dbx file, 61
 - DOMParser class, 162
 - DOM-to-GIDB mapping process, 360
 - Double helix DNA structure, 293
 - DPP (Digital Pattern Processing)
 - technology, 309
 - Drill-down methods, 388
 - Drivers, 50–51
 - DSx1 data set, 504–505
 - DSx10 data set, 504–505
 - DSx100 data set, 504–505
 - DTD events, 244
 - DTDHandler interface, 220
 - DTDHandler methods, 244
 - DTDs (Document Type Definitions), xxiii, xxiv–xxvi, 348, 520
 - document-centric XML, xxv
 - handling content, 220
 - Java class generation, 161–162
 - lacking information, 24
 - simple XML document, xxv
 - source file creation from, 161–162
 - dxxShredXML stored procedure, 324
 - Dynamic data types, 308
 - Dynamic Web pages, xxii
- ## E
- EBI Web site, 292
 - ebXML initiative, 569
 - edge table view, 177–180
 - EJB (Enterprise Java Beans), 23
 - EJB API, 23
 - EJB-based API (Enterprise Java Bean-based API), 337
 - Element content selection queries, 511
 - Element name selection queries, 510
 - ELEMENT node, 440
 - Element table, 468
 - element_name table, 200–201, 226
 - Elements
 - generating benchmark content, 507–508
 - long names, 201
 - mapping names to node identifiers, 62

- Elements (*continued*)
 - names exceeding maximum length, 235
 - not containing tags or child elements, xxv
- elements.dbx file, 61, 62
- Embedded XML databases, 69
 - ad hoc queries, 71
 - advantages, 72
 - application design, 71
 - basics, 70–71
 - building applications, 73–86
 - checkpoints, 85–86
 - code requirements, 71
 - component failure, 70
 - configuration, 74–75
 - connecting legacy applications, 71–72
 - deadlocks, 81–84
 - ease of use for end user, 70
 - end-user query tools, 70
 - function call or class method API, 70
 - graphical user interfaces, 70
 - indexing, 75–77
 - latency, 70
 - performance, 71
 - policy decisions by developer, 74–75
 - programming for transactions, 80–81
 - query interfaces, 70
 - recovery processing after failures, 86
 - reducing contention, 84–85
 - reliability, 71
 - report generators, 70
 - running directly in application or system, 70
 - running inside application, 73
 - scalability, 71
 - speed, 70
 - storing and fetching records, 70
 - threading model, 73
 - throughput, 70
 - two-phase locking, 81–84
- Emp element, 106
- endCDATA() method, 242–244
- endDocument method, 242
- endDTD method, 241
- endElement() method, 240–241
- endEntity() method, 242–244
- Enterprise Information Portal, 113
- EntireX Communicator, 31
- Entity beans, 346
- Entity references, 203
- entity_reference table, 203, 228
- Environment, 75
- Enzymes, 295
- equality keys, 78–79
- Equality operator, 64–65
- Error checking stored procedures, 205
- ESQL/C (Embedded SQL for C), 268
- Eukaryota/Prokaryota breakdown of cell, 303
- EVALUATE RULE operator, 406–408, 413, 444
- EVALUATE statement, 407
- EVALUATED-FOR element, 446–447
- EVALUATE-RULE operator, 441–443
- Evaluating with fast path join algorithms, 45
- EVALUATION-FEATURE element, 442
- Event-handling functions, 31
- Exact match attribute value selection
 - queries, 510
- Exact match query expressions, 65
- Exception-handling methods, 218–219, 223
- ExecXPath() function, 267
- ExecXPath() module, 264
- eXist, 43
 - document id, name id pairs for keys, 62
 - abbreviated XPath syntax, 49
 - accessing with SOAP, 52–54
 - application development, 46–47, 50–56
 - architecture overview, 45–47
 - Axis SOAP toolkit, 52
 - broker classes, 46
 - Cocoon, 47
 - collections, 44
 - command-line clients, 47
 - current versions, 47
 - database engine, 46

- default keyword tokenizer, 49
 - deployment, 46
 - direct access to local instance, 50
 - document identifiers, 58
 - driver implementations, 50
 - efficient, index-based query processing, 43
 - extensions for full-text searching, 45
 - fast path join algorithms, 45
 - full-text search extensions, 49
 - generated node identifiers, 61
 - hardware requirements, 47
 - index and storage implementation, 61–63
 - index files, 61
 - index-based access to full-text content of nodes, 48–49
 - index-based query processing, 44–45
 - indexing system, 58–61
 - installation process, 47
 - integration with Cocoon, 54–56
 - JDK (Java Development Kit), 47
 - native XML database, 45
 - node identifiers, 58
 - numbering scheme, 58
 - path join algorithms, 64
 - pluggable storage backends, 45–46
 - programming Java applications, 50–52
 - query engine, 66
 - query execution approaches, 57–58
 - query language extensions, 47–49
 - query language processing, 63–65
 - query performance, 65–66
 - query processor, 63–65
 - querying text, 48–49
 - relational backend, 45, 46
 - remote database engine using XML-RPC calls, 50
 - schema-less XML data storage, 44
 - search engine, 43
 - serializer, 62
 - specifying input document set, 48
 - technical background, 56–66
 - testing scalability, 66
 - Tomcat Web server, 47, 52
 - XML nodes, 58
 - XML resources, 50
 - XML:DB API, 55
 - XML-RPC interface, 52
 - XPath query engine, 49
 - eXist databases, 51–52
 - eXist Web site, 47
 - existsNode method, 133
 - existsNode SQL function, 144
 - Exons, 304
 - extended data type, 270
 - EXTEND-WITH-CLASS element, 432
 - Extensibility, 4
 - extract function, 133–135
 - extract SQL function, 144
 - EXTRACTING clause, 406
 - extract() method, 146
 - extractXML class, 245–247
 - extractXML.java file, 192
- ## F
- FALSE-BRANCH element, 429
 - FASTA, 312
 - featureAttributeDate subtype, 370
 - featureAttributeDecimal subtype, 369
 - featureAttributeInteger subtype, 369
 - featureAttributeString subtype, 369
 - FeatureCollection element, 368
 - featureMembers element, 368
 - Federated queries, 109
 - Fields, separating, 15
 - FILE_DATASTORE property, 131
 - Files, processing by URL, 142
 - Filter expressions, 279
 - Filtering, 129
 - Flat streams, 467
 - Flexibility, 4

Flight control software, 380
 FNMOC (Fleet Numerical Meteorology and Oceanography Center), 356
 FOR XML query, 170–171, 181, 184
 Formatting XML documents, xxviii
 Forms, 16
 fragment element, 463
 Fragments, 459, 463–464, 469
 Frequency, 414
 Frise, Erwin, 299
 Full structure index, 36
 Full-text searching, 45
 Fully qualified URI (Uniform Resource Identifier), 50
 fuzzy search (?) function, 128

G

GAME (Genome Annotation Markup Elements), 299
 GemStone, 353
 GenBank, 300
 <gene> . . . </gene> element, 303
 Gene Expression RFP (Request For Proposal), 300
 Gene sections, 303
 Generalized queries, 251
 Generic hash table, 343
 Genes, 293–294, 303–305
 Genetic code, redundancies in coding, 316
 Genetic Code Table, 295–296
 Genetic-based information models, 300–308
 Genome, 293
 Geographic information, transfer and storage, 368
 Geographical layer object, 363
 Georeferenced map images, 367
 George Mason University, 384
 Georgetown University Web site, 299
 GET method, 23
 GET request, 152
 getClob() method, 158
 getClobVal() method, 132, 133
 getCollection method, 50
 getExternalUrl method, 159
 getNumberVal() method, 132, 133, 134
 getNumRowsProcessed() method, 140, 148
 getStringVal() method, 132, 133, 134
 getStringValue method, 145
 getUri method, 159
 getURL method, 142
 getUrl() method, 158
 getXML method, 148
 getXMLDOC method, 140
 getXMLMetaData method, 141
 getXMLSAX method, 140
 getXMLSchema method, 141
 getXMLString method, 140
 GIDB (Geospatial Information Data Base), 353
 application schema document (.XSD), 368
 architecture, 355
 catalog, 360
 GML import and export, 368–372
 hierarchy, 359
 instance document (.XML), 368
 integration of METOC (Meteorological and Oceanographic) data, 356–362
 Metcast connectivity, 357
 Ozone, 354
 standard interface to data stores, 357
 success of, 355
 WMS architecture, 367
 WMS (Web Map Service) Interface Specification, 363
 GIDB catalog tree, 360–361
 GIDB Library node, 360
 GIDB mapping application, 354, 359, 366
 GIDB Metcast driver, 357, 362
 gidb namespace, 368, 370–372
 GIDB Ozone database, 354

GIDB portal
 drivers, 355
 geographical layers, 365
 retrieving data from many data
 repositories, 354
 URL request for Capabilities interface, 363
 WMS driver, 367
 GIDB portal driver, 363
 GIDB request, 361
 gidb:_featureAttribute subtype, 368, 369
 gidb:VectorFeatureType type, 368, 370
 gidb.xsd namespace application schema, 370
 GIS (Geographic Information System)
 applications, 355, 372
 Global sequence alignment algorithm, 312
 GML (Geographic Markup Language), 355,
 368–372
 gml namespace, 368
 gml:AbstractFeatureType feature type, 368
 Gordon, Paul, 300
 GROUP BY clause, 147, 406, 407
 groupby elements, 466
 Grouping documents, 25
 GROUPING element, 437, 442
 Groupings, 7
 Guanine (G), 293

H

handleText() method, 238, 239–240
 Harmon, Paul, 567
 HASPATH operator, 129–130, 131
 HASPATH query form, 126
 HDML (Handheld Device Markup
 Language), 151
 HEAD method, 23
 HEAD node, 440
 HEAD-SCHEMA element, 437, 443
 hello-world.xml file, 211–213
 hello-world-xml.sql script, 192, 211

Heterogeneity, 4
 Heterogeneous federated data access, 111–112
 Hierarchical operators, 271–272
 Hierarchy, 5–6
 High school reimbursement request
 application, 329–330
 HLA architecture, 400
 Hormones, 295
 HTML, 151
 Cocoon, 54
 resulting from XSLT transformation, xxvix
 HTML documents, xxi–xxii
 HTML files, search templates based on
 structure, xxi
 HTML pages, embedding JSP tags, 41
 HTTP, 351
 accessing remote eXist server, 46
 encoding information, 34
 Microsoft SQL Server 2000, 167–169
 security, 169
 Tamino XML Server, 22
 HttpUriType subtype, 156
 Human genome, starting and stopping
 points, 314
 Human Genome Project, 291, 292
 Hunter, Lawrence, 310
 Hybrid documents, xxiii

I

IBM Content Management and Enterprise
 Information Portal, 112–113
 IBM Informix IDS 9.x, 259
 IBM XML data management
 API goals, 113–116
 application interface, 113–116
 compiler updates to support XML, 119
 data manager goals, 116
 data type, 113–116
 engine, 116

- IBM XML data management (*continued*)
 - indexing, 119
 - logging and locking mechanisms, 120
 - new storage and runtime support, 119
 - storage, 116
 - transaction management, 120
 - utilities, 120
 - vision, 113
- IDREF attributes, 465
- IETF (Internet Engineering Task Force), 567
- ignorableWhitespace() method, 238–239
- IIS ISAPI extension, 167
- Image repository, 337
- Images
 - converting data into XML, 344–345
 - printing, 341
 - retrieving, 341
- Imielinski, T., 413
- "Improve Cash Flow" case study
 - addressing problem, 328–333
 - company scenario, 327–328
 - Department of Education, 328, 329
 - Department of Education reimbursement
 - request Web service, 330–333
 - future extensions, 333–334
 - high school, 328
 - high school reimbursement request
 - application, 329–330
 - input data to Web service, 330
 - reimbursing teachers, 327–328
 - response data, 330–331
- Index entries, 8
- Index space, 36
- Index-based query processing, 44–45
- Indexes
 - 3-tuple identifiers, 58
 - alternating level-order numbering scheme, 60–61
 - attribute node type, 76
 - Berkeley DB XML specifications, 76–77
 - edge node type, 79
 - edge path type, 76
 - edge-element-presence-none type, 79
 - element node type, 76
 - embedded XML databases, 75–77
 - equality key type, 76, 79
 - eXist implementation, 61–63
 - implementation, 61
 - level-order numbering, 60
 - node path type, 76
 - node-attribute-equality-string type, 78–79
 - node-attribute-presence-none type, 78
 - node-element-equality-string type, 78
 - node-element-presence-none type, 78
 - none syntax type, 76
 - number of children node may have, 60
 - number syntax type, 76, 79
 - numbering scheme, 58
 - organization, 61
 - organizing entries by collections, 62–63
 - presence keys, 76, 78
 - searching collections with, 75–76
 - string syntax type, 76
 - substring key type, 76, 79
 - traditional inverted, 63
- Indexing
 - embedded XML databases, 75–77
 - maximum document size, 59–60
 - queries, 280–281
- Inductive databases, 402, 451
 - for binary association rules, 414
 - instance of, 413
 - instances dealing with specific type of
 - pattern, 415–416
 - schema, 413
 - types of patterns represented, 415
 - XDM (XML for Data Mining), 403
- Information, 298
 - attributes, 5
 - context list, 7–8

- data elements, 5
- efficient use of storage, 309–310
- expressing, 5–6
- hierarchy, 5–6
- immediate availability, 308–309
- integration, 92
- ordering components, 5
- tags, 5
- transforming, xxviii
- Information integration-specific architecture
 - and technology, 111–113
- Information models
 - genetic-based, 300–308
 - pitfalls, 9–14
- Information systems, modeling, 4–5
- Informix, 477
- Informix Dynamic Server data, 111
- Informix IDS, 277
- Inhabitants attribute, 31
- Inner collection, 136
- INPATH operator, 129–130, 131
- INPATH query form, 126
- Input validation, 347
- /inquiry expression, 280
- /inquiry//address expression, 280
- /inquiry/inquiry-from expression, 280
- INRIA, 540, 541
- INSERT, 284
- insertValue() method, 217–218
- insertXML method, 142, 143
- InterestingCustomers table, 407
- intExecSQL() method, 215–216
- Introns, 304
- Invalid characters, 106
- INVENTORY table, 327
- ISAPI extension, 169
- isFragment() method, 132
- Isolation level, 38

J

- J2EE 1.3 platform, 351
- Japan International Protein Information Database, 299
- Java
 - class generation, 161–162
 - DAWAX packages, 471
 - GIDB database and system code, 354
 - processing XMLType, 135
 - SQL-J standard, 268
 - XDK (XML Development Kit), 160
 - XML, 342
 - XSU (XML SQL Utility), 125
- Java API, 23
- Java API for XML Processing (JAXP version 1.1), xxvi
- Java applications, 50–52
- Java Beans, 162–163
- Java classes, 192
- Java programs
 - constructing XML documents, 162
 - invoking, 154–155
- Java RMI (Remote Method Invocation), 383, 397
- Java Transaction API, 351
- Java-based Cloudscape, 277
- java.lang.String getStringVal() method, 135
- java.lang.String getXMLString method, 140
- java.net.URL object, 142
- JAXB (Java Architecture for XML Binding)
 - generating Java classes representing XML document structure, 343–344
 - support for XML Schema, 348
- JAXB compiler, 348
- Jaxen, 57, 65
- Jaxen Web site, 65
- JAXM (Java API for XML Messaging), 351
- JDBC, 268
- JDBC database connection, 135, 149–150

JDBC-ODBC Bridge, 219
 JDOM, 343
 JEDMICS (Joint Engineering Data Management Information Control System), 337–339
 inability to select and return different types of data, 341
 mass transfer capability, 350
 Oracle XSU, 347
 querying and producing reports on print jobs, 351
 querying repository, 347–348
 as XML documents, 350
 JMS (Java Messaging System), 351
 Join conditions, 459, 465
 join element, 465
 Journal space, 36
 JSP (Java Server Pages), 41
 <jsp:forward> JSP mechanism, 155
 <jsp:include> JSP mechanism, 155

K

KDD (Knowledge Discovery) process, 402
 keepObjectOpen method, 140
 <keyword> tag, 300
 Keyword-based expression, 65
 Keywords, 63, 300
 Knowledge management system, 292, 298
 Kweelt, 477

L

LabBook, Inc. Web site, 298
 Lancaster QoS-A (Quality of Service Architecture), 399
 Lanzi, P.L., 409
 Layered Queuing Networks, 400
 LDAP (Lightweight Directory Access Protocol), 529
 LDIF (Lightweight Directory Interchange Format), 529

LegoDB, 473
 Lewis, Suzanna, 299
 Lexers, 129
 Life sciences, modeling information with XML, 296–300
 LineFeatureType type, 368, 372
 Linnaean classification system, 297–298, 302–303
 Linnaeus, Carolus, 297
 LOBs (Large Objects), 124
 Location element, 28
 Locking mechanisms, 61
 Long-time logging, 36–37
 LRU (Least Recently Used) page-buffering, 61

M

MAGE-ML (Microarray Gene Expression Markup Language), 299
 MAGE-OM (Microarray Gene Expression Object Model), 299
 main() method, 213–214
 MAML (Microarray Markup Language), 300
 Mannila, H., 413
 Map formats, 365–366
 Map-in functions, 31, 33
 Map-out functions, 31, 33
 Mapping, 136
 asymmetric, 33
 data to functions, 31–33
 SQL names to XML names, 106
 symmetric, 33
 Mass transfer capability, 350
 match function, 49
 match-all function, 49
 match-any function, 49
 MATHML, 388–389
 MAX() function, 274
 MBL (Metcast Broker Language), 356
 MBL requests, 357, 361–362
 MEASURES element, 438

- MedNet, 398
- Messaging, 350–351
- Metabolic reactions, 295
- Metadata, 6, 263
 - attributes, 7, 10
 - data elements used as, 11
 - dynamic, 308
 - managing in XML warehouses, 462–466
 - mediated schema definition, 466
 - relational, 118
 - retrieving with JDBC statements, 345
 - unnecessary indirection, 12–14
 - XML, 118
- Metamodeling, 467
- Metacast
 - acquiring products from, 358
 - catalog, 356
 - implementation, 357–362
 - MBL requests, 357, 361–362
- Metacast catalog, 359–360, 362
- Metacast catalog DTD (Document Type Definition), 356
- Metacast data, 359
- Metacast data model, 356
- Metacast driver, 358–359
- Metacast server, 356
- Methionine, 314, 316
- Method query, 53
- Methods, exception-handling, 218–219
- METOC (Meteorological and Oceanographic)
 - data, 356–357
- MGED Web site, 300
- Michigan benchmark, 544–545
 - data set used by, 502–508
 - detail table, 516
 - generating string attributes and element content, 507–508
 - queries, 508–516
 - scaling, 504–505
 - schema of data, 505–507
 - summary table, 516
 - usage, 516–517
- Micro-benchmarks, 500
- Microsoft, 4
- Microsoft SQL Server 2000, 477
 - access via HTTP, 167–169
 - ad hoc URL query mechanism, 168
 - ADO, 167
 - database access, 166
 - direct query access, 168
 - IIS ISAPI extension, 167
 - ISAPI extension, 167
 - .NET interface, 167
 - providing relational views over XML, 176–180
 - querying using XPath, 183–185
 - rowset mechanism over XML, 177
 - serializing SQL query results into XML, 170–176
 - SOAP access, 169
 - SQL Server ISAPI, 167–168
 - SQLOLEDB provider, 167
 - template access, 168
 - XDR (XML Data Reduced) schema language, 182
 - XML access to, 166–170
 - XML features through SQLOLEDB, ADO, and .NET, 169–170
 - XML Schema format, 182
 - XPath XML view access, 168
- Microsoft.Data.SqlXml class, 170
- MINE CLASSIFICATION TEST operator, 413
- MINE-CLASSIFICATION operator, 409, 411–413, 426–427, 435, 441
- MINE-RULE operator, 405–406, 408, 413, 416, 422, 436–438, 441–443, 452
- MIN() function, 274
- Minimizing or eliminating deadlocks, 84
- MIX (Mediation of Information), 472
- Modeling information systems, 4–5

<Modulei> element, 484
 Molecular biology background, 293–295
 Molecular machines, 295
 Monument element, 28
 @mp:id meta property, 178
 @mp:parentid meta property, 178
 @mp:prev meta property, 178
 @mp:xmltext meta property, 178
 mRNA (messenger ribonucleic acid), 295, 304
 Multi-Media MedNet, 398
 Multiroot B+-tree, 61
 Munich Information Center for Protein
 Sequences, 299
 mycollection1 collection, 33
 MyCustomer view, 142, 146
 mydb database, 31
 myelement element, 26
 myGroup group section, 131
 myQuery.xsql file, 149
 MySQL, 470
 myStorage preferences, 131
 myXsl.xsl style sheet, 151

N

Name attribute, 33
 name element, 33, 368
 <Name>...</Name> tag, 136
 NAMESPACE_HANDLING Boolean, 221
 Namespaces, xxv
 long prefixes, 201
 prefixes exceeding maximum length, 235
 Tamino XML Server, 28
 XML Schema, 28
 Name_Type type, 137
 Nara Institute of Science and Technology, 542
 Narrower Term operator, 128
 National Biomedical Research Foundation, 299
 National Ocean Survey, 354

Native XML databases, xxix–xxx, 123, 477, 547
 basic and advanced features, 43
 benchmarks, 535
 best performance of, 563–564
 built and designed for handling XML, 21
 database size, 556–558
 direct mapping, 346
 eXist, 45
 performance, 347
 performance-enhancing technologies, 548
 reporting, 561–562
 round-tripping, 346
 SQL operations (mass records), 560–561
 SQL operations (one record), 558–560
 storing XML data directly, 548–549
 Tamino XML Server, 21
 XMach-1 benchmark, 550–551
 XML documents, 533
 XPath or XQL for generating queries, 346
 Native XML type, 114
 NATIX, 473
 Naval Oceanographic Office, 354
 Naval Research Laboratory, 353
 NCBI (National Center for Biotechnology
 Information), 292, 300
 NCBI Web site, 292
 <ND> tags, 313
 near function, 65
 NEAR operator, 128
 Needleman-Wunsch algorithm, 312
 NeoCore, 4
 NeoCore DPP technology, 318
 NeoCore sequence search plug-in, 312
 NeoCore SSE, 313–319
 NeoCore XMS (XML Management System)
 accessing, 311
 adding structural information to protein-
 record, 311
 dynamic management of metadata, 308
 efficient use of storage, 309–310

- hierarchical vector convergence
 - algorithm, 309
 - immediate availability of information, 308–309
 - indexing data for data-only queries, 310–311
 - information couplets, 308
 - integrating BLAST into, 311–319
 - loading information into, 310
 - powerful plug-ins, 316
 - query retrieval, 311
 - retrieving information, 309
 - scalability, 309
 - schema independence, 309
 - Swiss-Prot protein database, 310
 - well-informed XML, 317
- Nested Sets model, 195
- Nested Sets property, 252
- Nested structure queries, 137
- Nesting tags, 139
- .NET, 167
 - wsdl.exe tool, 52
 - XML features, 169–170
- Networks
 - best-effort service, 384
 - component integration, 375–376
 - heterogeneity, 388
 - QoS service levels, 383
- New Applications data item, 430
- NEW distance education, 384
- NEW-APPLICANTS element, 430, 431
- New_Applicants table, 412
- NHGRI (National Human Genome Research Institute), 298
- NIH (National Institutes of Health), 292
- NIMA (National Imagery and Mapping Agency), 354
- Nimble notation, 392
- NLM (National Library of Medicine), 292
- Node identifiers, 61
- node table, 200
 - inserting document node, 234
 - node ID, 226
- Node types, 59
- Node UDT, 277, 284
- NodeImpl class, 521
- Nodes
 - adding, 284–285
 - child nodes, 253, 271
 - descendants, 271
 - modifying data, 284
 - names, 279–280
 - names and kinds, 271
 - Nested Sets coordinates, 231–232
 - nulls value of y, 230
 - parent node, 252–253, 271
 - position in hierarchy, 264
 - relative locations, 271
 - unique value, 263–264
 - value of node_id, 230
 - value of y, 230
- node_type lookup table, 198–199
- Nontyped DOM (Document Object Model)
 - implementation, 521–522
 - directory servers, 530–532
 - relational databases, 524–526
- Non-Web based applications interactions, 394
- nonXML doctype, 25
- NOT (~) operator, 128
- not() logical operator, 158
- NS_PREFIX_HANDLING Boolean, 221
- Nucleotide sequences, 311–312, 318
- Nucleotide-Nucleotide Search (Blastn), 313
- Nucleotides, 293, 295
- NUM attribute, 138
- Numeric text, 133

O

- OAI
 - relational databases, 347
- OAI (Open Application Interface), 342
 - accessing, 339
 - architecture, 339
 - component architecture, 340
 - content validation, 347
 - control over presentation, 342
 - CORBA clients, 342
 - CORBA layer, 338
 - design choices, 341–345
 - DTDs for defining engineering documents, 342
 - early implementation of JAXB, 348
 - entity beans, 346
 - extensibility, 342
 - external components, 339
 - generating output DTD and XML Schema, 344
 - image metadata, 344
 - Image Query Service, 339, 341, 344–345
 - input and output to external client interfaces, 342
 - input validation, 347–348
 - JAXM (Java API for XML Messaging), 351
 - JDBC statements for retrieving metadata, 345
 - JEDMICS, 342
 - JMS (Java Messaging System), 351
 - limited subset of DOM capability, 343
 - mass transfer capability, 350
 - messaging, 350–351
 - middle-layer services components, 339
 - Open Standards-based interface, 338
 - overview, 339–341
 - persistence layer components, 339
 - portability, 342
 - Print Service, 339, 341
 - queries, 340
 - Query Service, 339, 340, 348, 349
 - querying JEDMICS repository, 347–348
 - retrieving image data, 344
 - Security Service, 339
 - stored procedures for manipulating collections of data, 345–346
 - structure validation, 347
 - unique session identifier, 339
 - updating engineering drawings and collection metadata, 346
 - user communities, 345
 - Web Services, 349–350
 - XML, 341–342
 - XML data, 339
 - XSLT (XML Transformations), 349
- OASIS (Organization for the Advancement of Structured Information Standards), 569
- Oasis Web site, xxxvi
- Object communication de facto standard, 383
- Object databases, 190
- Object types, 136–137
- Object view, 138–139
- Object-oriented databases
 - benchmarks, 534–535
 - DOM (Document Object Model), 528
 - XML and, 543
- Object-oriented programming languages, 190
- Object-relational databases and XML, 542–543
- Object-relational extensions, 116
- Objects, converting XML input into, 343–344
- ObjectStore, 353
- OccasionalType element, 505–506
- ODBC (Open Database Connectivity), 268
- ODBC data source, 219
- OGC (OpenGIS Consortium), 363
- OGC WMS (OpenGIS Consortium Web Map Service), 355
- OLAP (On-Line Analytical Processing) applications, 401
- OMEGA endpoint architecture, 399

- OMG (Object Management Group), 300, 569
- OO (object-oriented), 353
- Open Source PostgreSQL-DBMS, 260, 277
- OpenGIS Web Map Server Interface
 - Implementation Specification, 363
- OpenGIS Web site, 363
- OpenXML, 177–178, 180
- OpenXML rowset provider, 177
- Operators, hierarchical, 271–272
- Optimization technique, 312
- OR (Object-Relational) approach, 119
- or logical operator, 158
- OR (|) operator, 128
- Oracle, 69, 148, 344
 - database access, 345–347
 - full XPath syntax, 130
 - Java Beans, 162–163
 - object types to support URI concept, 155–156
 - special features, 155–163
 - "thin" JDBC driver, 150
- Oracle Text cartridge, 126–127
- Oracle XML TransViewer beans, 162–163
- Oracle XSU (XML SQL Utility), 344
- Oracle8i, 477
- Oracle9i, 477
 - CLOB (Character Large Object)
 - functionality, 125
 - parsers, 160–161
 - URIs (Uniform Resource Identifiers)
 - support, 155–160
 - as XML database, 124
 - XMLType object type, 132–135
- Oracle9iAS, 149
- oracle.sql.CLOBgetClob() method, 135
- OracleText, 127–130, 133
- oracle.xdb.XMLType Java class, 135
- OracleXMLQuery class, 139–141
- OracleXMLSave class, 139, 142
- oracle.xml.uri.OraDbUriServlet servlet, 158
- oracle.xml.xsql.XSQLRequest object, 155
- ORDBMS (Object-Relational Database Management System), 259, 268
 - Compare() UDF, 274
 - extended data type, 270
 - indexing queries, 280–281
 - user-defined functions, 276
 - user-defined types, 270
- ORDBMS Cloudscape, 259–260
- Order-based selection queries, 510
- ORDERS table, 325
- Order-sensitive parent-child selection queries, 511–512
- ORF (Open Reading Frames), 314
- org.w3c.dom.Document getXMLDOC
 - method, 140
- OSE (Oracle Servlet Engine), 158
- OT Land Web site, 567
- "Our Most Valued Customers Come First"
 - case study
 - acting on problems from most important customers first, 322
 - better business decisions for future product lines, 327
 - company scenario, 322
 - future extensions, 326–329
 - immediate response to customer, 326
 - more accurate marketing strategies, 327
 - order data, 325
 - problem reports as XML documents, 323–324
 - solution to problem, 322–326
 - triggered actions, 325–326
- Outbound Web services, 110
- OUTPUT element, 422, 432
- Ozone, 354
- Ozone Web site, 354

P

- Palomar Web site, 303
- Pan-enterprise information systems, 260
- <para> element, 484
- PARAM element, 426, 431
- Parent node, 271
- Parent-child complex pattern selection queries, 513
- Parent-child selection queries, 511
- Parsed character data, xxv
- parse() method, 221–223
- Parser Event Objects, 267
- parser interface, 264
- Parsers for Oracle9i, 160–161
- Parsing XML data, 102
- Partial view, 422
- Partstock table, 110
- path expressions, 271
- Path join algorithms, 64
- PATH_SECTION_GROUP section, 130–131
- pattern element, 464
- Pattern indices, 8–9
- pattern.node element, 464
- Patterns, 6–8, 459, 468
 - automatically building database, 9
 - describing, 464
 - evaluating, 469
- #PCDATA keyword, xxv
- #PCDATA nodes, 437
- PDF files and Cocoon, 54
- Pepper, Senator Claude, 292
- <phenotype> . . . </phenotype> element, 304
- PHONE elements, 146–147
- <PHONE> tag, 146
- PHONES tag, 138
- Phone_Tab table, 137
- Physiome Sciences, Inc., 299
- pi_data_leaf table, 203, 228
- PIP (payload interface processor), 380
- PIR (Protein Information Resource)
 - database, 299
- pi_target_leaf table, 203, 228–229
- PLAY node set, 64
- PLAY//SPEECH expression, 64
- PL/SQL
 - generating XML documents from query results, 148–149
 - XDK (XML Development Kit), 160
- PMML (Predictive Model Mark-up Language), 403, 416–419, 448–449, 452
- PMML documents, 416–418
- PNG image, 366
- Pointer-based join queries, 514
- PointFeatureType type, 368, 372
- POST method, 23
- posted-xml parameter, 154
- Preferred Term operator, 128
- Presence keys, 78
- "Primer on Molecular Genetics," 293
- Primitive data types, xxv
- Primitive elements, xxv
- print method, 161
- Printing
 - documents, 341
 - images, 341
- Printing bean, 350–351
- Problem reports as XML documents, 323–324
- PROBLEMS table, 325, 327
- Procedures, stored for error checking, 205
- Process hierarchies, 277
- processContents element, 26
- Processing instruction, 237
- processingInstruction method, 237
- PRODUCT element, 436, 437
- PROFILE element, 424, 427, 430
- Programming for transactions, 80–81
- <promoter> . . . </promoter> element, 304
- Promoter sections, 304
- Protein-Protein Search (Blastp), 313

Proteins, 293–295, 304
 Proteomics, LLC, 298
 Proteomics Canada, Ltd., 298
 Psaila, G., 409
 PSDML (Protein Sequence Database Markup Language), 299
 PUT method, 23

Q

QoS-aware configuration language, 395–397
 Quaternary system, 293
 Queries
 ad hoc, 345
 aggregation, 515
 ancestor nesting in ancestor-descendant selection, 512
 ancestor-descendant complex pattern selection, 514
 ancestor-descendant selection, 512
 benchmarks, 480–482
 blocking operation, 264–265
 complex pattern selection, 513
 components as terms, 15
 converting single SELECT expression to, 144–146
 cursor mechanism on results, 38
 DADX files, 110
 data-centric environments, 37
 describing structure, 139
 displaying results as XML documents, 163
 document, 480
 document-centric environments, 37
 element content selection, 511
 element name selection, 510
 exact match attribute value selection, 510
 execution, 39
 eXist execution approaches, 57–58
 eXist performance, 65–66

 federated, 109
 full-text searching, 45
 generalized, 251
 GET request, 152
 index-based processing, 43, 44–45
 indexing, 280–281
 information about, 248–249
 Michigan benchmark, 508–516
 nested structure, 137
 OAI, 340
 optimizing, 76
 order-based selection, 510
 order-sensitive parent-child selection, 511–512
 over hierarchy of type-specific tables, 270
 parameterization, 152–154
 parent-child complex pattern selection, 513
 parent-child selection, 511
 pointer-based join, 514
 relational, 480
 results as well-formed XML, 38–39
 returned structure, 509–510
 selection, 509–514
 session context, 38
 SQL, 124
 SQL XML output, 103
 SQL/XML, 105
 SSE (sequence search embedded), 312
 string distance selection, 511
 structural selection, 511–514
 Tamino XML Server, 37–39
 text content, 251
 text leaves, 250
 transactions, 38
 update, 515–516
 value-based join, 514
 X007 benchmark, 486–488
 XMach-1 benchmark, 489–490
 XMark benchmark, 495–497

Queries (*continued*)
 XML view, 168
 XPath, 78–80, 262
 queries
 standardization, 125
 Query functions, 31
 Query language extensions, 47–49
 Query language processing, 63–65
 Query languages for XML, 472
 Querying
 JEDMICS repository, 347–348
 repository, 248–258
 using XPath, 183–185
 XML documents, 254
 XML messages, 262
 QueryResponse type, 53
 QueryService class, 53
 Quilt, 472
 Quotes, escaping, 216

R

Rapid application development, 41
 Rapide, 393
 RDBMS (relational database management systems)
 attribute modeling approach, 555–556
 element modeling approach, 555–556
 as information store, 190
 long element names and namespace prefixes, 201
 modeling columns as XML DTD structure, 555–556
 RDF (Resource Description Framework), 54
 Rechenmann, Francois, 296
 <reference> element, 303
 registerHandler method, 159
 Regular path expressions
 decomposing into chain of basic steps, 63–65
 index structure, 57–58

re_i_avl procedure, 227
 Related Term operator, 128
 Relational data
 building XML documents from, 144–155
 database support, 124
 density, 118
 lack of order, 118
 providing XML views over, 182–187
 reusing, 124
 stored in DB2, 325
 XML, 165–166
 Relational database systems, 69, 123–124, 166
 Relational database-mining framework, 413
 Relational databases, 189–192
 Adjacency List Model, 195
 benchmarks, 534
 book sales and reports design, 551–554
 bulk loading XML data into, 166
 creation of, 196–213
 data-centric documents, 124–125
 data-mining operators built on top of, 413
 Department of Education, 331
 document-centric documents, 124
 DOM storage, 195
 emulating extensibility, 11–14
 mapping XML data to, 467–468
 name/value pairs, 12
 Nested Sets Model, 195
 node mapping, 97, 99–100
 nontyped DOM implementation, 524–526
 OAI system, 347
 physical data model, 197–198
 serializing document out of repository, 206–211
 source raw data, 402
 storing XML documents, 524–526
 table creation, 198–206
 user-defined data types, 198
 XML and, 541–542
 XML as transport format, 165
 xmlrep_user login, 197–198

- Relational metadata, 118
- Relational queries, 480
- Relational storage, supporting along with XML, 118–119
- Relational tables, mapping XML data to, 271
- Relational views, 176–180
- Relative path expressions, 280
- rep_gen_xpath procedure, 254–256
- rep_i_an procedure, 226–227
- rep_i_cdl procedure, 227
- rep_i_cl procedure, 228
- rep_i_d procedure, 225
- rep_i_en procedure, 226
- rep_i_er procedure, 228
- rep_i_n procedure, 226
- rep_i_pidl procedure, 228
- rep_i_pitl procedure, 228–229
- rep_i_tl procedure, 229
- replace() method, 216–217
- Reporting hierarchies, 277
- Repository
 - adding document data, 263
 - closing connection to, 215
 - connecting to, 213–220
 - DOM API, 258
 - initializing connection to, 214
 - locking granularity, 284
 - manually building XML documents, 211–213
 - new schema specifications, 263
 - predefined schema, 292
 - querying, 248–258
 - serializing XML documents out of, 206–211
 - SQL definition of schema, 269–270
 - storing XML data in shredded format, 264
 - supporting update operations, 284
 - tables, 270
 - testing connection, 220
- REPS table, 326
- rep_s_cdlid procedure, 229
- rep_s_children procedure, 253
- rep_serialise_nodes procedure, 245
- rep_serialise_nodes stored procedure, 212
- rep_s_n_last procedure, 230
- rep_s_parent procedure, 252–253
- rep_s_tlid procedure, 229
- rep_u_n_y procedure, 230
- rep_u_n_y_null procedure, 230
- Requests and URLs (uniform resource locators), 363
- Resources
 - WebDAV management, 40
 - XML:DB API, 50
- ResourceSet, 52
- Response XML document, 333
- restart() method, 140
- restartQuery function, 148
- result element, 465, 466
- Result pattern, 459
- result.getIterator() method, 52
- result.node element, 466
- Retrieve method, 54
- Reusable architecture, 377–378
- R.G. Edwards Web site, 257–258
- Ribosomes, 295
- RISK attribute, 432, 433
- RMI (Remote Method Invocation), xxvix
- Rollback, 33
- Rollback transactions, 219
- Ron Bourret's Web site, xxxvi
- Root tag, 8
- <ROW> tag, 145–146, 150–151
- <ROW>...</ROW> tag, 136, 138
- <ROWSET> element, 150
- ROWSET tag, 146
- <ROWSET> tag, 151
- <ROWSET>...</ROWSET> tag, 136
- RPCs (Remote Procedure Calls), 349
- RULE element, 446
- RULE-SCHEMA element, 437, 443–444
- RULE-SECTION element, 442
- Rule_Set1 result set, 406
- Runtime network control system, 397–398

S

- SALSA, 312
- Sanger, Frederick, 291
- SAX (Simple API for XML), xxvii, 220
 - data as text stream, 343
 - endElement event, 232
 - exception-handling methods, 223
 - LexicalEvent handler methods, 238
- SAX ContentHandler, 220
- SAX events, 62
- SAX interface, 160
- SAX LexicalHandler interface, 220
- SAX parser, 220
 - beginning and end of DTD (document type definition), 241–242
 - closing tag, 240–241
 - DTD events, 244
 - end of file, 242
 - instantiating and invoking, 221–223
 - lexical events, 242–244
 - parsing entity references, 238
 - startDocument() method, 234
 - well-formed XML files, 224–225
 - whitespace, 238
- SAX1parser, 220–221
- SAX2parsers, 220–225
- SBML (Systems Biology Markup Language), 299
- SBW (Systems Biology Workbench), 299
- Scalability and NeoCore XMS, 309
- Scalable tools, 375
- scanXML class, 224
- scanXML.java file, 192
- Schema documents based on W3C XML
 - Schema, 39
- Schema Editor, 40
- Schema representation language, 389
- Schemas
 - annotating, 27–28
 - arbitrary support for, 118
 - declaring, 178
 - default mapping, 182
 - dependence, 309
 - editing, 39–40
 - evolution of, 28
 - flexible approach to, 118
 - independence, 309
 - inductive databases, 413
 - mediated definition of, 466
 - of Michigan benchmark data, 505–507
 - organizing, 39
 - source file creation from definition, 161–162
 - SQL definition of repository schema, 269–270
 - subsets of, 279
- SCHOOLID value, 332
- SCHOOL_INFO table, 331
- SCORE function, 128
- Searching
 - for text, 252
 - XML data, 102
- Security, 42, 169
- SELECT statement, 107, 406
 - concatenation as function call, 145
 - left-to-right appearance of table alias, 173
- Select views, 458
- SELECT-FROM-WHERE query, 134, 137
- Selection queries, 509–514
- Self-constructing XML information system, 8
- Sequence alignment tools, 311
- sequence tag, xxv
- Sequoia benchmark, 479
- Serializing
 - document out of repository, 206–211
 - SQL query results into XML, 170–176
- Serine, 316
- service.query method, 52
- Services, 50
- <session> tag, 154
- Sessions, 38
- Set enumeration model, 402
- Set Query benchmark, 478

- setBatchSize method, 143
- setCommitBatch method, 143
- setMaxRows method, 140, 148
- setRowIdAttrName method, 141
- setRowIdAttrValue method, 141
- setRowSetTag method, 141, 148
- setRowTag method, 141, 144, 148
- setSkipRows method, 140, 148
- setStylesheetHeader method, 141
- Setting user permissions, 204–205
- setXSLT method, 144
- SEVERE_PROBLEMS table, 325, 326
- SEV_PROBLEMS trigger, 325
- SGML (Standard Generalized Markup Language), xxiv
- Shapefile format, 354
- shorten function, 160
- Shorttext view, 160
- Shredded rowset view, 177–180
- Sibling relationships, 7
- Simon, Hank, 300
- Simple .NET client in C#, 52–54
- Simple types, defining, xxv
- sitemap.xmap file, 54–55
- Sleepycat Software Web site, 73
- Smalltalk, 353
- SMIL, 389, 394, 399
- Smith-Waterman algorithm, 312
- SNPs (Single Nucleotide Polymorphisms), 304–305
- SOAP (Simple Object Access Protocol), xxvix, 349, 569
 - accessing eXist with, 52–54
 - accessing remote eXist server, 46
 - client libraries, 52
 - toolkits, 52
 - user-defined types, 52
 - WSDL (Web Services Description Language) service description, 52
- SOAP messages, 351
- Software
 - architectures and reusability, 376
 - components, 376
- Software AG, 21
- soundex (!) function, 128
- SOURCE element, 422, 426, 431, 437, 442, 443
- source element, 462–463
- Space applications, 376–378
- Spacecraft on-board memory, 382
- Spacecraft software, 380–382
- Spaces, 36–37
- SPEAKER="HAMLET" predicate
 - subexpression, 64
- SPEC table, 272, 279
- SPEECH elements, 64
- SPEECH[SPEAKER] expression, 64
- SQL
 - ad hoc, 225
 - combining XPath with, 134
 - data entry stored procedures, 225–231
 - data structured as relations, 118
 - definition of repository schema, 269–270
 - executing statements, 215–216
 - expressions, 177
 - node mapping, 97–98
 - Node User-Defined Type Expression, 275–276
 - queries, 124
 - stored procedures, 252–254
 - update operations, 107
- SQL Node type, 272–274, 276, 284–285
- SQL queries
 - ad hoc, 248–252
 - aliases, 136
 - auto mode, 170–171, 173–174
 - avoiding caching of large XML fragments on server, 172
 - checking XPath qualifiers, 282
 - explicit mode, 170–171, 174–176
 - generating arbitrary XML, 174–176
 - nested mode, 170–171, 173–174
 - parameterization, 152–154

- SQL queries (*continued*)
 - raw mode, 170–172
 - related to any tables or views, 136
 - relational NULL values serialized, 172
 - serialization process as post-processing step, 171
 - serializing results into XML, 170–176
 - as string value or as JDBC ResultSet, 139–140
 - universal table format, 174–176
 - well-formed XML documents, 172
- SQL XML
 - composed data, 113–114
 - data generation, 103
 - development, 114
 - enhanced query results, 112
 - functions, 103
 - information integration functionality, 105
 - output from queries, 103
 - publishing relational data in XML fragments, 102–103
 - support, 92
 - support architecture, 102–103
 - support technology, 104–106
 - viewing table, 104
- SQL XML View Mapper, 183
- SQL-3, 281–284
 - DBMS engines, 270
 - queries, 264
 - reconstructing XML document from repository schema, 276–277
 - sorting data, 280
 - XML_Doc_Trees view, 280
- SQL-3 structured types, 268
- SQLOLEDB, 169–170
- SQLOLEDB property, 172
- SQLOLEDB provider, 167, 168
- sql:relation annotation, 183
- SQLXML OLEDB provider, 168, 169
- SQL/XML query, 105
- SQLXML templates, 180–181
- SQLXML Web Microsoft Web site, 165
- SqlXmlAdapter class, 170
- SqlXmlCommand class, 170
- SqlXmlParameter class, 170
- sql:xpath-query element, 184
- SS7 (Signaling System 7)
 - adapters, 386
 - communication mechanism interface, 385–386
 - MG (manager) component, 385–386
 - MT (monitor) component, 385–386
 - Nimble notation, 386–387
 - potential integration problems, 384–387
- SSE (sequence search embedded), 312
- SSW (Space Wide Web)
 - adaptive spacecraft software, 380–382
 - adaptive Web-based configuration language, 378
 - scenario using future, 378–379
 - supporting communications satellites constellations, 378–382
- Standard indexes, 35–36
- startCDATA() method, 242–244
- startDocument method, 234
- startDTD method, 241–242
- startElement method, 235–237
- startEntity() method, 242–244
- starts-with function, 79
- State transitions, 421–422
- stem (\$) function, 128
- Stock table, 110
- Stop codons, 314
- StoPLists, 128
- Storage system
 - eXist implementation, 61–63
 - high volumes of information, 262
 - XML-enabled repository, 262
- Stored procedures, 252–254, 345–346
 - data entry, 225–231
 - parameter, 170
 - setting permissions, 231

- Storing XML documents, 31, 523–533
- STREAMs, 397–398
- String distance selection queries, 511
- String types, 34
- Strings
 - comparisons, 126
 - cutting into leaves, 217–218
 - generating benchmark attributes, 507–508
 - replacing, 216–217
 - searching for, 48
 - splitting into leaves, 237
 - substring search, 126
 - XMLType object, 145
- Structure index, 36
- Style sheets, 144–145
- Subelements, xxv
- SUBJECT_DATA table, 331
- Substitution groups with feature attributes, 369–370
- Substring keys, 79
- Subtrees, 26
- Support, 405
- SUPPORT element, 440, 446
- SVG (Scalable Vector Graphics), 54, 151
- Swiss-Prot, 300
- Swiss-Prot Web site, 292
- SWW (Space Wide Web), 375
 - distributed component-based software applications, 376
 - research and boundaries, 387
- SWWACXML (Space Wide Web Adapter Configuration eXtensible Markup Language), 387
 - adapters, 376
 - application-level network QoS control, 399–400
 - building adapters with generated mapping code, 392–393
 - components, 390
 - configuration language, 389–392
 - connectors, 390
 - formalizing notation adapters describing, 392
 - interaction support for adapter communication, 389–400
 - lower-level approach to describing structure and properties of components and connectors, 393
 - mapping notation, 391–392
 - <meta-head> section, 395, 396
 - MT and MG interactions, 390–391
 - network autoconfiguration, 395–397
 - Nimble notation, 392
 - parsing specifications, 392
 - performance validation and evaluation, 399
 - QoS, 395–397
 - research approach, 388–389
 - research tasks, 389–400
 - reusable components, 392
 - runtime network control system, 397–398
 - <SWWACXML-connector> section, 395, 396, 398
 - <SWWACXML-info> section, 395, 396, 398
 - <tagattribute> . . . <> pairs, 390
 - testing, 398
 - XML compliant, 390
 - <SWWACXML-build> . . .
 - </SWWACXML-build>, 391
 - <SWWACXML-elements> . . .
 - </SWWACXML-elements>, 390
- SWW-XML, 379
- <SYMPTOMS> element, 327
- Synonym operator, 128
- SYS_DBURIGen function, 159–160
- System Management Hub, 23
- Systema Naturae*, 297
- SYS_XMLAgg function, 146–147
- SYS_XMLGen functions, 144–147
- SYS-XMLGenFormatType type, 146
- SYS_XMLGenFormatType type, 145
- SYS_XMLGenFormatType.createFormat method, 145

T

- TableName-given rowset schema, 178
- Tables
 - alias as element name, 173
 - combination primary key, 200
 - creation of, 198–206
 - repository, 270
- Tagesspiegel Online, 543–544
- Tags, 5, 271
 - attribute names as, 145
 - describing data elements, 7
 - hierarchy, 8
 - inadequate use of, 12–14
 - naming, 148
 - nesting, 139
 - open, 235
 - preprinted parts of forms, 14
 - renaming, 144
- Tamino, 123, 452, 477
- Tamino databases, 25, 36–37
- Tamino EJB API, 23
- Tamino Manager, 23
- Tamino XML Schema, 31–33
- Tamino XML Server
 - accessing external relational databases, 23
 - accessing other databases, 28–31
 - APIs, 22–23
 - architecture, 22–23
 - collation information, 34–35
 - compression techniques, 36
 - configuration data, 25
 - different types of data, 22
 - doctypes, 25
 - editing schemas, 39–40
 - enhancing standard schematic descriptions, 24
 - full database functionality, 42
 - full-text search, 37
 - graphical browser-based administration interface, 23
 - handling query results, 38–39
 - HTTP, 22
 - identifier for documents or non-XML object, 25
 - indexing, 35–36
 - internationalization, 34–35
 - journal space, 36
 - long-time logging, 36–37
 - modified schemas, 28
 - modifying database information, 29
 - namespaces, 28
 - native XML databases, 21
 - non-Unicode encoding, 34
 - open content option, 26
 - propagating information to external systems, 23
 - queries, 37–39
 - Schema Editor, 40
 - schema example, 29–31
 - security, 42
 - sessions, 38
 - simple XML schema, 27–28
 - SSL communication, 22
 - standard indexes, 35–36
 - storing and retrieving information, 23
 - storing arbitrary objects, 25
 - structure index, 36
 - supported HTTP methods, 23
 - text indexes, 35–36
 - tools, 39
 - transactional logging, 36
 - transactions, 38
 - Unicode, 34
 - user-defined functions, 37
 - WebDAV (Web-based Distributed Authoring and Versioning) access, 40
 - word fragment index, 36
 - X-Application, 41
 - XML documents, 22
 - XML Schema, 26–28
 - XML storage, 23–25

- X-Node component, 28–31
- XPath, 37
- X-Plorer, 39
- X-Port component, 22
- XQuery, 37
- Tamino X-Node, 23
- Tamino X-Tension, 23
- Target, storing with leafing mechanism, 203
- <taxonomy> tag, 303
- Taylor, Ron, 310, 311
- Technologies, typical cycle of, 567
- Technology data management Web services, 107–111
- Telephone directory listing, 15, 16
- Templates, 180–181
- Test pages, automatically generating, 107
- TEST-CLASSIFICATION operator, 430, 432, 435
- Text, 220
 - numeric, 133
 - querying, 48–49
 - searching for, 252
 - splitting into multiple nodes, 239
- Text content queries, 251
- Text indexes, 35–36
- Text leaves queries, 250
- Text search operator (=), 49
- text() function, 35, 133, 134, 158
- text_leaf table, 202–203, 229
- Thymine (T), 293
- TIBCO, 4
- Tiny HTML, 151
- title element, 466
- Tokenizing text into keywords, 63
- Tomcat, 47, 52, 107
- Topography and Bathymetry layer, 365
- TPC Web site, 479
- TPC-D benchmark, 478
- TPC-H benchmark, 478
- TPC-R benchmark, 478
- TPC-W benchmark, 479
- Training set, 408–409
- Training_Set table, 409–410, 411
- Transaction log, 85
- Transaction-handling methods, 219
- Transactions, 38
 - beginning and ending, 80
 - commit or rollback, 33, 81
 - committing, 219
 - locking, 81–82
 - multiple operating on same data, 84
 - programming for, 80–81
 - rollback, 33, 82, 219
 - rolling back deadlocked, 83–84
 - terminating, 82
- TRANSACTIONS element, 436
- Transactions table, 404–405
- Transact-SQL statement, 170
- Transcription, 295
- Transcription factors, 304
- <transcription_factor> . . .
 - </transcription_factor> elements, 304
- Transforming
 - information, xxviii
 - XML data, 101–102
- transform.transformIn function, 33
- transform.transformOut function, 33
- Translation, 295
- Tree Java objects, 343
- Treeviewer bean, 163
- tRNA (transfer RNA), 295
- TRUE-BRANCH element, 429
- T-SQL FOR XML extensions, 168
- T-SQL fragment, 178–180
- T-SQL statements, 168, 180–181
- Two-phase locking, 81–84
- Typed DOM (Document Object Model)
 - implementation, 522–523
 - databases, 527
 - directory servers, 530

U

UDDI (Universal Discovery Description Integration), 349, 569
 UDFs (user-defined functions), 96, 117
 outbound Web services, 110
 XML Extender, 94
 UDTs (user-defined types), 94, 272
 UML (Unified Modeling Language), 521, 569
 Unicode (Unicode Consortium 2000), 34
 Universal table format, 174–176
 University of Wisconsin, 540
 University Personnel DTD, 520
 Unstructured data, 112–113
 UPDATE method, 267, 284
 Update queries, 515–516
 UPDATE statement, 134
 UPDATE WHERE CURRENT OF CURSOR, 266
 Updategrams, 180–181, 185–186
 updateXML method, 142, 143
 Updating XML documents, 142
 uploadFile method, 233–234
 Uploading XML documents, 220–247
 uploadXML class, 231–245
 uploadXML.java file, 192
 UriFactory type, 159
 UriFactoryType class, 156
 URIs (Uniform Resource Identifiers), 125
 URIs (Uniform Resource Identifiers) support, 155–160
 URIType object, 155
 UriType object, 156, 159
 URLs (uniform resource locators), 155, 363
 urn:oracle-xsql namespace, 149
 urn:schemas-microsoft-com:xml-sql namespace, 168, 182
 urn:schemas-microsoft-com:xml-updategram namespace, 186
 U.S. Army Corps of Engineers, 354
 U.S. Census Bureau, 354
 U.S. Geological Survey, 354

U.S. Navy, 338
 useLowerCaseTagNames method, 141
 User-Agent type, 154
 User-defined
 data types, 198
 facets, 34
 functions, 37, 272
 object types, 137
 useTypeForCollElemTag method, 141
 useUpperCaseTagNames method, 141

V

validateContent() method, 161, 162
 Validating XML data, 102
 Value-based join queries, 514
 Variables, 459
 VectorFeatureType type, 372
 Venter, Craig, 292
 view element, 462, 463
 Views, 469
 composite, 458–459
 defining, 463–466
 definition example, 460
 definition of result pattern, 465–466
 result pattern, 459
 select, 458
 XDM databases, 422
 Virtual names, 169
 voidExecSQL() method, 215
 VPF (Vector Product Format), 354
 vroot (virtual root), 167

W

W3C
 DOM methods node types, 59
 XML Query Language Working Group, 480
 XML Schema, 24

XPointer specification, 155
 WAP (Wireless Access Protocol), 54
 WAS, 107
 Watson, James, 291
 Web map servers, 363
 Web pages
 dynamic, xxii
 X-Application, 41
 Web servers, accessing databases from, 149–155
 Web Services, 92, 107–108, 349–350, 569
 Web-based interconnection, 394–395
 WebDAV (Web-based Distributed Authoring and Versioning)
 access, 40
 accessing remote eXist server, 46
 WebDAV (Web-based Distributed Authoring and Versioning) Server, 40
 WebDAV-based resource management, 40
 WebLogic, 149
 WebSphere Studio Application Developer, 107
 Well-informed XML, 317
 Whitespaces as character data, 238–239
 (*) wildcard, 48
 WIRE, xxi
 Wisconsin benchmark, 478, 500
 WITHIN operator, 129–130, 131
 WITHIN query form, 126
 WML (Wireless Markup Language), xxiii, 151
 wml-example.xml file, 248
 WMS (Web Map Service), 363, 367
 WMS Capabilities DTD, 363
 WMS driver, 365–367
 WMS-compliant map servers
 GIDB portal driver, 363
 image or raster maps, 365
 PNG image, 366
 Word fragment index, 36
 words.dbx file, 61, 63
 WORF (Web services Object Runtime Framework) tools, 107
 World Wide Web (Web), 375

Wright, 393
 Write-ahead logging, 81–82
 WSDL (Web Services Description Language),
 169, 349, 569
 WSDL documents, 107
 wsdl.exe tool, 52
 WWW (World Wide Web), 376

X

X007 benchmark, 479, 501–502, 544
 database parameters, 485
 queries, 486–488
 X007 database, 482–485
 XADT type, 543
 X-Application, 41
 xdb_g.jar JAR archive, 135
 XDK (XML Development Kit), 140, 160–162
 XDM (XML for Data Mining), 403
 association rules, 435–447
 basic concepts, 419–423
 benefits, 447–449
 classification model, 428–429
 classification with, 423–435
 comparing with PMML, 448–449
 derivation dependencies, 447–448
 extensibility, 449
 extraction of association rules, 436–438
 flexibility, 447, 449
 incremental computations, 449–450
 open description, 448
 as open framework, 427
 user interaction, 450
 XDM databases, 419–423
 XDM data items, 419
 containing purchase transaction data,
 435–436
 derivation with evaluated association rules,
 444–447

- XDM databases
 - building classification model, 425–427
 - classification process, 434–435
 - classification tree, 429
 - derivation of association rules, 438–441
 - derivation of classified test data, 432–434
 - derivation of evaluated association rules, 444–447
 - derived data items, 419
 - evaluating association rules, 441–444
 - initial state, 423–425
 - nonderived data items, 419–420
 - partial view, 428
 - state, 419–421
 - state transition, 421–422, 425–427, 436
 - test phase, 430–432
 - training set, 423–425
 - view showing test phase, 433–434
 - views, 422
 - XDM data items, 419
 - XDM element, 420
- XDM element, 420
- XDM-DATA-ITEM element, 419, 420–421, 424, 429
- XDM-DERIVATION element, 419, 421, 429, 433, 438, 444
- XDM-STATEMENT element, 422, 437, 442
- XDM-TRANSITION element, 421–422, 425–426, 436–437, 441
- Xerces XML parser, 192–193, 359, 363
- Xindice, 43
- Xindices project, 61
- XMach-1 benchmark, 501, 544, 549
 - book sales and reports database design, 551–554
 - native XML databases, 550–551
 - queries, 489–490
- XMach-1 database, 488–489
- XML-related databases, 550–551
- XMark benchmark, 501
 - queries, 495–497
- XMark database, 491–494
- XMAS, 472
- XMI, 569
- XML (eXtensible Markup Language),
 - xxi, 388–389
 - abuse of, 10
 - advantages, 567
 - business information systems, 261–262
 - complex distributed requests, 112
 - concepts, xxiv–xxvii
 - as container for data managed by legacy systems, 3
 - control over presentation, 342
 - conversion of Database Data into, 344
 - conversion of Image Data into, 344–345
 - conversion of XML input into objects, 343–344
 - converting documents from legacy format, 72
 - for data exchange, 332
 - data management, xxix–xxx
 - Data Model, 270
 - as data transport for external data, 346
 - database access, 345–347
 - data-centric, xxiii–xxiv
 - definition of, xxii
 - directory servers and, 543–544
 - document-centric, xxiii–xxiv
 - DOM (Document Object Model), xxvi–xxvii
 - DTDs (Document Type Definitions), xxiv–xxvi
 - existing benchmarks, 482–497
 - expressing information, 5–6
 - extensibility, 4, 16, 342
 - extensions to, 568
 - flexibility, xxiii, 4, 10
 - grammar, 3–4
 - heterogeneity, 4
 - as information domain, 4–5
 - information model, 292–293
 - informationally complete, 4
 - infrastructure to support business, 568

- interoperability, 342
- Java, 342
- linear/hierarchical structure, 118
- mapping legacy data formats to, 72
- mapping to another data model, 21
- marked-up content, 165
- Metcast catalog, 362
- Microsoft SQL Server 2000 access to, 166–170
- modeling behavior of entire application once, 3
- namespaces, 368
- navigating tree structure, 183
- Nested Sets Model, 195
- nesting manipulation operations, 114
 - with no data elements, 10
- OAI, 341–342
- object-oriented databases, 543
- object-relational databases and, 542–543
- patterns, 6–9
- platform-independent, 165, 190
- portability, 123, 342
- potential, 3
- query languages, 125, 472
- relational data, 165–166
- relational databases, 541–542
- relational views over, 176–180
- representation for DBUriType subtype, 156–157
- research on, 260, 451
- revolutionary nature, 297–298
- self-describing, 4
- semi-structured data, 165
- serializing SQL query results into, 170–176
 - simple data access, 112
- SQL-centric and XML-centric way to generate, 170
- SQLOLEDB, 169
- storing, 31
- storing as CLOB, 126–132
- structured data, 165
 - style, 3–4
 - supporting along with relational storage, 118–119
 - syntax, 3
 - transforming into another format, 141
 - Unicode (Unicode Consortium 2000), 34
 - Unicode-based syntax, 165
 - use-case scenario, 261–262
 - valid, xxiii
 - well-formed, xxiii
 - XML schemas, xxiv–xxvi
- XML applications
 - data store help with creating, 116
 - request support, 94
 - textual content ignorable, 202
- XML Benchmark Project, 550
- XML Collections, 92, 99–101, 107
- XML Columns, 92, 107
- XML comments, 204
- XML data
 - benchmark data set, 479–482
 - benchmarks, 499–501
 - bulk loading into relational databases, 166
 - Bulkloading, 186
 - changing data schema, 118
 - characteristics, 502–505
 - data set granularity, 504
 - database support, 124
 - depth, 502–503
 - depth and width of tree, 480
 - different approaches to storing, 467
 - document centric, 191
 - edge table view, 177–180
 - fanout, 502–503
 - flat streams, 467
 - graphical interface for management of, 470
 - indexing methods, 117
 - mapping to relational databases, 467–468
 - mapping to relational tables, 263–264, 271
 - meta properties, 177–178
 - metamodeling, 467

- XML data (*continued*)
 - mixed content, 191
 - mixed storage, 467
 - multirow updates, 180
 - OAI, 339
 - parsing, 102
 - posting into database, 151–152
 - searching, 102
 - shredded rowset view, 177–180
 - storage, 472
 - storing in native format, 72
 - storing in shredded format, 264
 - systems for integration, 472–473
 - as text stream, 343
 - transforming, 101–102
 - validating, 102
 - viewing mechanism for, 455–456
 - XML columns, 95–96, 98–101
- XML data component, 468
- XML data store, 61
- XML databases
 - benchmarks, 544–545, 549–550
 - client/server, 71, 72
 - embedded, 71–73
 - guidelines for benchmarking, 545
 - hierarchy of concepts, 157
 - observations, 427–428
 - position and structure of nodes, 61
- XML document tree, 177
- XML documents, xxviii
 - access privileges, 157
 - adding nodes, 284–285
 - as aggregations of information, 309
 - appending documents, 284
 - atomic types, 270
 - atomic value, 271
 - base64 encoding, 344–345
 - batch size, 151
 - BLOB (Binary Large Object) form, 266
 - building DOM tree from, 162
 - building from relational data, 144–155
 - closure property, 458
 - complex attribute values, 157
 - complex nested structure, 137–138
 - composing or decomposing, 96–98
 - composition functions, 326–327
 - data about person, xxii–xxiii
 - data models, 520–523
 - databases for storing, 523–533
 - data-centric, xxiii
 - decoding processes, 344
 - defining structure, xxiv–xxvi
 - deleting, 151
 - Department of Education, 329
 - describing structure, xxiii
 - directory servers, 528–533
 - displaying graphically as tree, 163
 - document-centric, xxiii
 - DOM representations, 194
 - editing, 163
 - extensible constructs, xxv
 - fine-grained storage, 135–144
 - flow semantics, xxviii
 - following XML rules, 150
 - formatting, xxviii
 - generating from PL/SQL queries, 148–149
 - generating Java classes representing structure, 343–344
 - Huffman coding, 345
 - implicit order-document order, 118
 - improved scalability, 114
 - indexing schemes, 58–61
 - input and output to external client
 - interfaces, 342
 - inserting, 151
 - as linked chunks, 191
 - locating and linking information, xxvii
 - maintaining document order, 284
 - management of large collections, xxii
 - managing within databases, 452

- manually building, 211–213
- mapping from side tables to, 96–98
- memory management in large, xxvi–xxvii
- MIME type, 158
- more complex XPath examples, 278
- native XML databases, 533
- navigating logical structure, 57–58
- navigation improvement, 114
- as node tree, 193
- node types, 270
- nodes, 271
- nontyped DOM implementation, 521–522
- opening, 234
- outputting, 31
- posting into databases, 151–152
- problem reports as, 323–324
- qualifying, 133–134
- query results as, 163
- querying, 254
- readable output, 245–247
- relational database storage, 524–526
- relational tables with CLOB columns, 126
- restructuring, 455–456
- restructuring possibilities, 458–459
- schema structure, 263
- schema-less storage, 44
- schemas, 26–28
- schematic information, 23–24
- self-describing, xxii
- serializing out of repository, 206–211
- side tables, 95
- simple method for designing, 14–16
- sparsity, 118
- SQLXML templates, 180–181
- storage, 125, 163, 190–191
- storage benchmarks, 533–535
- storage in one table, 141–142
- stored as CLOBs (Character Large Objects), 94
- stored as files linked from DBMS, 94
- stored as fragments in collection of table, 94–95
- stored in regular relational tables, 191
- stored intact in relational column, 94
- stored procedures for data entry, 225–231
- stored with XML Extender, 94
- storing as contiguous blocks of data, 284
- storing data with one node-per-second, 284
- studies in storing and retrieving, 541
- subsets of schema, 279
- table or view, 151
- tags, 271
- Tamino XML Server, 22
- templates, 168
- translating into in-memory tree structure, 343
- transparently composed from one or many data sources, 102
- two-character hexadecimal representation, 344
- updating, 142, 151–152
- uploading, 220–247
- valid, 151
- valid characters, 344
- validating, xxiii, 161–162
- view model, 458–460
- views, 141–142
- visualizing, 163
- well formed, xxiii
- well-formedness check, 233–234
- XSL transformations, 163
- XML elements, proper use of, 6–7
- XML Extender, 92, 107, 323, 324, 477
 - architecture, 94–95
 - DAD files, 107
 - DAD (data access definition) files, 96–98
 - data storage, 113
 - federated environments, 102
 - locating elements and attributes, 96
 - processing support for DB2, 94

- XML Extender, (*continued*)
 - shared architecture and technology, 92
 - subset of XPath standard, 102
 - technology, 95–100
 - UDFs (user-defined functions), 94, 102
 - UDTs (user-defined types), 94
 - XML application request support, 94
 - XML collections, 99–101
 - XML columns, 95–96, 98–101
 - XML object storage, 94
 - XML Schema support, 102
- XML Extender applications, 117
- XML files
 - transforming and creating output file, 162–163
 - well-formed, 224–225
- XML fragment
 - describing customer, 105
 - output, 105–106
- XML markup languages, 298–300
- XML metadata, 118
- XML objects storage, 94–101
- XML parsers, 125–126, 160
- XML Path Language. See XPath
- XML query language, 480, 547–548
- XML repository prototype
 - detailed design descriptions, 268–286
 - high-level system architecture, 262–268
 - overview, 261–262
- XML Schema, xxiii, xxiv–xxvi, 24, 270, 348
 - attributes, 26
 - collation facet, 34
 - complexity, xxv
 - data items within attributes, 307
 - data-centric, xxv
 - namespaces, xxv, 28
 - simple XML document, xxvi
 - string types, 34
 - stronger typing constraints on data, xxv
 - subtrees, 26
 - Tamino-specific notation, 25
 - user-defined facets, 34
 - wildcards, 26
 - xs:annotation element, 27–28
- XML Schema Language, 568
- XML SourceView Bean, 163
- XML storage, 116–117
 - handling objects of predefined type, 24
 - Tamino XML Server, 23–25
- XML tags, 342
- XML template, 170
- XML Transform Panel bean, 163
- XML views, 166
 - annotated schema, 182–183
 - defining Customer-Order hierarchy, 182–183
 - providing over relational data, 182–187
 - selecting data from, 183
 - updating using Updategrams, 185–186
 - as virtual views, 184
- XML warehouses, 455–456
 - architecture, 457–458
 - Datawarehouse component, 468–469
 - Datawarehouse implementation
 - component, 458
 - Datawarehouse specification component, 458
 - DAWAX, 470–473
 - defining views, 463–466
 - different approaches to storing XML
 - data, 467
 - extracting data, 469
 - graphic tool for specifications, 461–462
 - managing, 466–469
 - managing metadata, 462–466
 - mapping XML to relational databases, 467–468
 - mediated schema definition, 466
 - query manager, 458
 - root element, 462
 - specification, 458–462
 - storage, 466–469
 - View definition example, 460
 - view model, 458–460

- view storage, 468–469
- XML data component, 468
- XML2CLOB statement, 104, 106
- Xml2Sql component, 469
- XMLAGG statement, 103, 104
- XML-as-data model, 264–265
- XML-as-document model, 264–266, 284
- XMLATTRIBUTES statement, 103, 104
- xml_books.xml file, 54
- XMLCLOB type, 95
- XML_Data_Type SQL type, 270
- XML:DB API, 46–47, 50–52, 55
- xmldb namespace, 56
- XML:DB XSP logic sheet, 55–56
- xmldb:collection element, 56
- xmldb:get-xml function, 56
- xmldb:xpath tag, 56
- XML_Doc_Trees view, 276, 280
- XMLELEMENT statement, 104, 106
- XML-enabled databases, 477, 547–548
- XML-enabled repository, 262
- XMLFile type, 95
- XML_Node_Type type, 270
- XmlNote table, 468
- XML-QL (XML Query Language), 472, 548
- XML-related databases
 - database size, 556–558
 - reporting, 561–562
 - SQL operations (mass records), 560–561
 - SQL operations (single record), 558–560
 - XMach-1 benchmark, 550–551
- XML-related technologies
 - SOAP (Simple Object Access Protocol), xxvix
 - XPath, xxvii
 - XSL (eXtensible Stylesheet Language), xxviii–xxix
- xmlrep database, 197
- XMLREP ODBC data source, 214
- xmlrepDB class, 213
- xmlrepDB.java file, 192
- xmlrepSAX class, 220–225
- xmlrepSAX.java file, 192
- xmlrep_user login, 197–198
- XML-RPC (XML Remote Procedure Call)
 - accessing remote eXist server, 46
 - client libraries, 52
- XML-RPC interface, 52
- XML_SECTION_GROUP section group, 131
- XMLSERIALIZE statement, 106
- xml-stylesheet parameter, 154
- XMLType object type, 125
 - content in serialized format, 133
 - converting, 146
 - example, 133
 - methods, 132
 - processing in Java, 135
- XMLType objects
 - canonical mapping, 144–146
 - checking well formedness, 133–134
 - strings, 145
- XMLTypeTab table, 133
- XMLVarchar type, 95, 96
- XMS (XML Management Systems), 4
 - behavior of, 8
 - benchmark specification, 478–479
 - index entries, 8
 - native XML databases, 477
 - XML-enabled databases, 477
- X-Node component, 28–31
- XPath, xxvii, 37, 264
 - arbitrary information access, xxvii
 - axes operations, 275–276
 - coercion rules of strings, 184
 - combining with SQL, 134
 - comparison operators, 279
 - constraints, xxvii
 - constructs, 183–184
 - data type coercion, 184
 - eXist extensions, 47–49
 - hierarchical information access, xxvii
 - mathematical and string-manipulation functions, 279

- XPath (*continued*)
 - multiple qualifiers, 282
 - node-to-string conversions in predicates, 184
 - queries, 51, 183–185, 262
 - query processing, 78–80
 - regular path expressions, 57
 - searching for strings, 48
 - semantics, 184
 - slow query execution, 48
 - SQLLEDB, 169–170
 - substring searches, 79
 - supported features, 158
 - variety of queries over container, 76
- XPath expressions, 45, 280
 - converting to SQL-3 queries, 264
 - cursor to walk over results set, 256–257
 - describing path within target document, 278
 - ease of writing and understanding, 427
 - evaluating, 267–268
 - examples, 278–279
 - generating, 254–257
 - handling, 264
 - identifying subsets of XML data
 - document, 278
 - parsing, 264
 - paths, 254–256, 279
 - predicate expressions filtering document locations, 279
 - qualifiers, 282
 - with range predicates, 270
 - restrictions, 134
 - simplicity, 278
 - SPEC table, 279
 - terminating with qualifying filter
 - expression, 282
 - transparent information sources, 427
 - validating, 178, 279
- XPath format, 157
- XPath interpreter, 268
- XPath queries, 53, 78–79, 168, 180–181
 - calling SSE plug-in, 312–313
 - parameter, 170
- XPathQueryService service, 52
- Xperanto project, 92, 124
- X-Plorer, 39
- XPointer, 158
- X-Port component, 22
- XQL (XML Query Language), 472, 548
- Xqps (XML queries per second), 550
- XQuery, 37, 57, 124, 264, 451, 472, 551
 - interpreter, 267
 - statements, 115–116
 - support, 116–117
- XRel, 542
- XRM (eXtended integrated Reference Model), 399
- XRR (XML Repository), 118
- xs:annotation element, 27–28
- xs:appinfo element, 28
- XSD (XML Schema Definitions), 191
- xsd:anyType type, 271
- <xsnul="true"> special null attribute, 157
- XSL (eXtensible Stylesheet Language), xxiii, 101–102, 141, 349, 455–456
 - example, xxviii
- XSL FO (XSL Formatting Objects), xxviii
- XSL transformations, 163
- XSL Transformer bean, 162–163
- XSLT (XSL Transformations), xxviii, 101–102, 118, 141, 191
- XSLT style sheets, 54, 154
- XSP, 55
- XSQL
 - bind-params example, 152
 - default values for bind-params, 153
 - directly executing DML (Data Manipulation Language) statements, 152
 - include-request-params example, 153–154
 - interoperability with JSP, 149
 - invoking programs, 154–155
 - lexical substitution parameter, 153
 - parameterizing queries, 152–154
 - query examples, 150–151

.xsql extension, 149
.xsql files
 executing, 154
 registering XSLT style sheet, 151
XSQL servlet, 149–151
XSQLConfig.xml file, 149
<xsql:delete-request> tag, 151
<xsql:include-request-paramname="P"/> forms,
 153–154
<xsql:include-request-params/> forms, 153–154
<xsql:include-xml href="url"/> tag, 154
<xsql:insert-request> tag, 151
<xsql:query> tag, 151–152
<xsql:query...> tag, 149
<xsql:update-request> tag, 151
xs:schema element, 27

XSU (XML SQL Utility), 125, 135
 canonical mapping, 136–139
 executing query in Java environment and
 producing XML, 139–141
 performance, 148
 registering style sheet, 141
 splitting XML documents into pieces and
 storing, 136
 transforming data into XML, 136
X-Tension, 31–33, 37
Xyleme, 473

Z

Zipf distribution, 508













