

Contents

Preface — VII

Overview — IX

List of Figures — XIX

List of Tables — XXII

Part I: Introduction

- 1 Background — 3**
 - 1.1 Database Properties — 3
 - 1.2 Database Components — 5
 - 1.3 Database Design — 7
 - 1.3.1 Entity-Relationship Model — 8
 - 1.3.2 Unified Modeling Language — 11
 - 1.4 Bibliographic Notes — 14

- 2 Relational Database Management Systems — 17**
 - 2.1 Relational Data Model — 17
 - 2.1.1 Database and Relation Schemas — 17
 - 2.1.2 Mapping ER Models to Schemas — 18
 - 2.2 Normalization — 19
 - 2.3 Referential Integrity — 20
 - 2.4 Relational Query Languages — 22
 - 2.5 Concurrency Management — 24
 - 2.5.1 Transactions — 24
 - 2.5.2 Concurrency Control — 26
 - 2.6 Bibliographic Notes — 28

Part II: NOSQL And Non-Relational Databases

- 3 New Requirements, “Not only SQL” and the Cloud — 33**
 - 3.1 Weaknesses of the Relational Data Model — 33
 - 3.1.1 Inadequate Representation of Data — 33
 - 3.1.2 Semantic Overloading — 34
 - 3.1.3 Weak Support for Recursion — 34

3.1.4	Homogeneity	— 35
3.2	Weaknesses of RDBMSs	— 36
3.3	New Data Management Challenges	— 37
3.4	Bibliographic Notes	— 39
4	Graph Databases	— 41
4.1	Graphs and Graph Structures	— 41
4.1.1	A Glimpse on Graph Theory	— 42
4.1.2	Graph Traversal and Graph Problems	— 44
4.2	Graph Data Structures	— 45
4.2.1	Edge List	— 46
4.2.2	Adjacency Matrix	— 46
4.2.3	Incidence Matrix	— 48
4.2.4	Adjacency List	— 50
4.2.5	Incidence List	— 51
4.3	The Property Graph Model	— 53
4.4	Storing Property Graphs in Relational Tables	— 56
4.5	Advanced Graph Models	— 58
4.6	Implementations and Systems	— 62
4.6.1	Apache TinkerPop	— 62
4.6.2	Neo4j	— 65
4.6.3	HyperGraphDB	— 66
4.7	Bibliographic Notes	— 68
5	XML Databases	— 69
5.1	XML Background	— 69
5.1.1	XML Documents	— 69
5.1.2	Document Type Definition (DTD)	— 71
5.1.3	XML Schema Definition (XSD)	— 73
5.1.4	XML Parsers	— 75
5.1.5	Tree Model of XML Documents	— 76
5.1.6	Numbering Schemes	— 78
5.2	XML Query Languages	— 81
5.2.1	XPath	— 81
5.2.2	XQuery	— 82
5.2.3	XSLT	— 83
5.3	Storing XML in Relational Databases	— 84
5.3.1	SQL/XML	— 84
5.3.2	Schema-Based Mapping	— 86
5.3.3	Schemaless Mapping	— 89
5.4	Native XML Storage	— 90
5.4.1	XML Indexes	— 90

- 5.4.2 Storage Management — 92
- 5.4.3 XML Concurrency Control — 97
- 5.5 Implementations and Systems — 100
 - 5.5.1 eXistDB — 100
 - 5.5.2 BaseX — 102
- 5.6 Bibliographic Notes — 104

- 6 Key-value Stores and Document Databases — 105**
 - 6.1 Key-Value Storage — 105
 - 6.1.1 Map-Reduce — 106
 - 6.2 Document Databases — 109
 - 6.2.1 Java Script Object Notation — 110
 - 6.2.2 JSON Schema — 112
 - 6.2.3 Representational State Transfer — 116
 - 6.3 Implementations and Systems — 118
 - 6.3.1 Apache Hadoop MapReduce — 118
 - 6.3.2 Apache Pig — 121
 - 6.3.3 Apache Hive — 127
 - 6.3.4 Apache Sqoop — 128
 - 6.3.5 Riak — 129
 - 6.3.6 Redis — 132
 - 6.3.7 MongoDB — 133
 - 6.3.8 CouchDB — 136
 - 6.3.9 Couchbase — 139
 - 6.4 Bibliographic Notes — 140

- 7 Column Stores — 143**
 - 7.1 Column-Wise Storage — 143
 - 7.1.1 Column Compression — 144
 - 7.1.2 Null Suppression — 149
 - 7.2 Column striping — 151
 - 7.3 Implementations and Systems — 158
 - 7.3.1 MonetDB — 158
 - 7.3.2 Apache Parquet — 158
 - 7.4 Bibliographic Notes — 159

- 8 Extensible Record Stores — 161**
 - 8.1 Logical Data Model — 161
 - 8.2 Physical storage — 166
 - 8.2.1 Memtables and immutable sorted data files — 166
 - 8.2.2 File format — 169
 - 8.2.3 Redo logging — 171

- 8.2.4 **Compaction — 173**
- 8.2.5 **Bloom filters — 175**
- 8.3 **Implementations and Systems — 181**
- 8.3.1 **Apache Cassandra — 181**
- 8.3.2 **Apache HBase — 185**
- 8.3.3 **Hypertable — 187**
- 8.3.4 **Apache Accumulo — 189**
- 8.4 **Bibliographic Notes — 191**

- 9 Object Databases — 193**
- 9.1 **Object Orientation — 193**
- 9.1.1 **Object Identifiers — 194**
- 9.1.2 **Normalization for Objects — 196**
- 9.1.3 **Referential Integrity for Objects — 200**
- 9.1.4 **Object-Oriented Standards and Persistence Patterns — 200**
- 9.2 **Object-Relational Mapping — 202**
- 9.2.1 **Mapping Collection Attributes to Relations — 203**
- 9.2.2 **Mapping Reference Attributes to Relations — 204**
- 9.2.3 **Mapping Class Hierarchies to Relations — 204**
- 9.2.4 **Two-Level Storage — 208**
- 9.3 **Object Mapping APIs — 209**
- 9.3.1 **Java Persistence API (JPA) — 209**
- 9.3.2 **Apache Java Data Objects (JDO) — 215**
- 9.4 **Object-Relational Databases — 217**
- 9.5 **Object Databases — 222**
- 9.5.1 **Object Persistence — 223**
- 9.5.2 **Single-Level Storage — 224**
- 9.5.3 **Reference Management — 226**
- 9.5.4 **Pointer Swizzling — 226**
- 9.6 **Implementations and Systems — 229**
- 9.6.1 **DataNucleus — 229**
- 9.6.2 **ZooDB — 230**
- 9.7 **Bibliographic Notes — 232**

Part III: Distributed Data Management

- 10 Distributed Database Systems — 235**
- 10.1 **Scaling horizontally — 235**
- 10.2 **Distribution Transparency — 236**
- 10.3 **Failures in Distributed Systems — 237**
- 10.4 **Epidemic Protocols and Gossip Communication — 239**

- 10.4.1 Hash Trees — 241
- 10.4.2 Death Certificates — 243
- 10.5 Bibliographic Notes — 244

- 11 Data Fragmentation — 245**
 - 11.1 Properties and Types of Fragmentation — 245
 - 11.2 Fragmentation Approaches — 249
 - 11.2.1 Fragmentation for Relational Tables — 249
 - 11.2.2 XML Fragmentation — 250
 - 11.2.3 Graph Partitioning — 252
 - 11.2.4 Sharding for Key-Based Stores — 253
 - 11.2.5 Object Fragmentation — 254
 - 11.3 Data Allocation — 255
 - 11.3.1 Cost-based allocation — 256
 - 11.3.2 Consistent Hashing — 257
 - 11.4 Bibliographic Notes — 259

- 12 Replication And Synchronization — 261**
 - 12.1 Replication Models — 261
 - 12.1.1 Master-Slave Replication — 262
 - 12.1.2 Multi-Master Replication — 263
 - 12.1.3 Replication Factor and the Data Replication Problem — 263
 - 12.1.4 Hinted Handoff and Read Repair — 265
 - 12.2 Distributed Concurrency Control — 266
 - 12.2.1 Two-Phase Commit — 266
 - 12.2.2 Paxos Algorithm — 268
 - 12.2.3 Multiversion Concurrency Control — 276
 - 12.3 Ordering of Events and Vector Clocks — 276
 - 12.3.1 Scalar Clocks — 277
 - 12.3.2 Concurrency and Clock Properties — 280
 - 12.3.3 Vector Clocks — 281
 - 12.3.4 Version Vectors — 284
 - 12.3.5 Optimizations of Vector Clocks — 289
 - 12.4 Bibliographic Notes — 293

- 13 Consistency — 295**
 - 13.1 Strong Consistency — 295
 - 13.1.1 Write and Read Quorums — 298
 - 13.1.2 Snapshot Isolation — 300
 - 13.2 Weak Consistency — 302
 - 13.2.1 Data-Centric Consistency Models — 303
 - 13.2.2 Client-Centric Consistency Models — 305

- 13.3 Consistency Trade-offs — **306**
- 13.4 Bibliographic Notes — **307**

Part IV: Conclusion

- 14 Further Database Technologies — 311**
 - 14.1 Linked Data and RDF Data Management — **311**
 - 14.2 Data Stream Management — **312**
 - 14.3 Array Databases — **313**
 - 14.4 Geographic Information Systems — **314**
 - 14.5 In-Memory Databases — **315**
 - 14.6 NewSQL Databases — **315**
 - 14.7 Bibliographic Notes — **316**

- 15 Concluding Remarks — 317**
 - 15.1 Database Reengineering — **317**
 - 15.2 Database Requirements — **318**
 - 15.3 Polyglot Database Architectures — **320**
 - 15.3.1 Polyglot Persistence — **320**
 - 15.3.2 Lambda Architecture — **322**
 - 15.3.3 Multi-Model Databases — **322**
 - 15.4 Implementations and Systems — **324**
 - 15.4.1 Apache Drill — **324**
 - 15.4.2 Apache Druid — **326**
 - 15.4.3 OrientDB — **327**
 - 15.4.4 ArangoDB — **330**
 - 15.5 Bibliographic Notes — **331**

Bibliography — 333

Index — 347