

Physical Database Design

**The Database Professional's Guide
to Exploiting Indexes, Views,
Storage, and More**

Sam Lightstone
Toby Teorey
Tom Nadeau



ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK* OXFORD* PARIS* SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Morgan Kaufmann Publishers is an imprint of Elsevier ,

MORGAN KAUFMANN PUBLISHERS

Contents

Preface	xv
Organization	xvi
Usage Examples	xvii
Literature Summaries and Bibliography	xviii
Feedback and Errata	xviii
Acknowledgments	xix
 Introduction to Physical Database Design	 I
1.1 Motivation—The Growth of Data and Increasing Relevance of Physical Database Design	2
1.2 Database Life Cycle	5
1.3 Elements of Physical Design: Indexing, Partitioning, and Clustering	7
1.3.1 Indexes	8
1.3.2 Materialized Views	9
1.3.3 Partitioning and Multidimensional Clustering	10
1.3.4 Other Methods for Physical Database Design	10
1.4 Why Physical Design Is Hard	11
1.5 Literature Summary	12
 Basic Indexing Methods	 15
2.1 B+tree Index	16
2.2 Composite Index Search	20

2.2.1	Composite Index Approach	24
2.2.2	Table Scan	24
2.3	Bitmap Indexing	25
2.4	Record Identifiers	27
2.5	Summary	28
2.6	Literature Summary	28
3	Query Optimization and Plan Selection	31
3.1	Query Processing and Optimization	32
3.2	Useful Optimization Features in Database Systems	32
3.2.1	Query Transformation or Rewrite	32
3.2.2	Query Execution Plan Viewing	33
3.2.3	Histograms	33
3.2.4	Query Execution Plan Hints	33
3.2.5	Optimization Depth	34
3.3	Query Cost Evaluation—An Example	34
3.3.1	Example Query 3.1	34
3.4	Query Execution Plan Development	41
3.4.1	Transformation Rules for Query Execution Plans	42
3.4.2	Query Execution Plan Restructuring Algorithm	42
3.5	Selectivity Factors, Table Size, and Query Cost Estimation	43
3.5.1	Estimating Selectivity Factor for a Selection Operation or Predicate	43
3.5.2	Histograms	45
3.5.3	Estimating the Selectivity Factor for a Join	46
3.5.4	Example Query 3.2	46
3.5.5	Example Estimations of Query Execution Plan Table Sizes	49
3.6	Summary	50
3.7	Literature Summary	51
4	Selecting Indexes	53
4.1	Indexing Concepts and Terminology	53
4.1.1	Basic Types of Indexes	54
4.1.2	Access Methods for Indexes	55
4.2	Indexing Rules of Thumb	55
4.3	Index Selection Decisions	58
4.4	Join Index Selection	62
4.4.1	Nested-loop Join	62
4.4.2	Block Nested-loop Join	65
4.4.3	Indexed Nested-loop Join	65

4.4.4	Sort-merge Join	66
4.4.5	Hash Join	67
4.5	Summary	69
4.6	Literature Summary	70
5	Selecting Materialized Views	71
5.1	Simple View Materialization	72
5.2	Exploiting Commonality	77
5.3	Exploiting Grouping and Generalization	84
5.4	Resource Considerations	86
5.5	Examples:The Good, the Bad, and the Ugly	89
5.6	Usage Syntax and Examples	92
5.7	Summary	95
5.8	Literature Review	96
6	Shared-nothing Partitioning -	97
6.1	Understanding Shared-nothing Partitioning	98
6.1.1	Shared-nothing Architecture	98
6.1.2	Why Shared Nothing Scales So Well	100
6.2	More Key Concepts and Terms	101
6.3	Hash Partitioning	101
6.4	Pros and Cons of Shared Nothing	103
6.5	Use in OLTP Systems	106
6.6	Design Challenges: Skew and Join Collocation	108
6.6.1	Data Skew	108
6.6.2	Collocation	109
6.7	Database Design Tips for Reducing Cross-node Data Shipping	I 10
6.7.1	Careful Partitioning	110
6.7.2	Materialized View Replication and Other Duplication Techniques	III
6.7.3	The Internode Interconnect	I 15
6.8	Topology Design	I 17
6.8.1	Using Subsets of Nodes	117
6.8.2	Logical Nodes versus Physical Nodes	I 19
6.9	Where the Money Goes	120
6.10	Grid Computing	120
6.11	Summary	121
6.12	Literature Summary	122

7	Range Partitioning	125
7.1	Range Partitioning Basics	126
7.2	List Partitioning	128
7.2.1	Essentials of List Partitioning	128
7.2.2	Composite Range and List Partitioning	128
7.3	Syntax Examples	129
7.4	Administration and Fast Roll-in and Roll-out	131
7.4.1	Utility Isolation	131
7.4.2	Roll-in and Roll-out	133
7.5	Increased Addressability	134
7.6	Partition Elimination	135
7.7	Indexing Range Partitioned Data	138
7.8	Range Partitioning and Clustering Indexes	139
7.9	The Full Gestalt: Composite Range and Hash Partitioning with Multidimensional Clustering	139
7.10	Summary	142
7.11	Literature Summary	142
8	Multidimensional Clustering	143
8.1	Understanding MDC	144
8.1.1	Why Clustering Helps So Much	144
8.1.2	MDC	145
8.1.3	Syntax for Creating MDC Tables	151
8.2	Performance Benefits of MDC	151
8.3	Not Just Query Performance: Designing for Roll-in and Roll-out	152
8.4	Examples of Queries Benefiting from MDC	153
8.5	Storage Considerations	157
8.6	Designing MDC Tables	159
8.6.1	Constraining the Storage Expansion Using Coarsification	159
8.6.2	Monotonicity for MDC Exploitation	162
8.6.3	Picking the Right Dimensions	163
8.7	Summary	165
8.8	Literature Summary	166
9	The Interdependence Problem	167
9.1	Strong and Weak Dependency Analysis	168
9.2	Pain-first Waterfall Strategy	170
9.3	Impact-first Waterfall Strategy	171
9.4	Greedy Algorithm for Change Management	172

9.5	The Popular Strategy (the Chicken Soup Algorithm)	173
9.6	Summary	175
9.7	Literature Summary	175
10	Counting and Data Sampling in Physical Design Exploration	177
10.1	Application to Physical Database Design	178
10.1.1	Counting for Index Design	180
10.1.2	Counting for Materialized View Design	180
10.1.3	Counting for Multidimensional Clustering Design	182
10.1.4	Counting for Shared-nothing Partitioning Design	183
10.2	The Power of Sampling	184
10.2.1	The Benefits of Sampling with SQL	184
10.2.2	Sampling for Database Design	185
10.2.3	Types of Sampling	189
10.2.4	Repeatability with Sampling	192
10.3	An Obvious Limitation	192
10.4	Summary	194
10.5	Literature Summary	195
11	Query Execution Plans and Physical Design	197
11.1	Getting from Query Text to Result Set	198
11.2	What Do Query Execution Plans Look Like?	201
11.3	Nongraphical Explain	201
11.4	Exploring Query Execution Plans to Improve Database Design	205
11.5	Query Execution Plan Indicators for Improved Physical Database Designs	211
11.6	Exploring without Changing the Database	214
11.7	Forcing the Issue When the Query Optimizer Chooses Wrong	215
11.7.1	Three Essential Strategies	215
11.7.2	Introduction to Query Hints	216
11.7.3	Query Hints When the SQL Is Not Available to Modify	219
11.8	Summary	220
11.9	Literature Summary	220

12 Automated Physical Database Design	223
12.1 What-if Analysis, Indexes, and Beyond	225
12.2 Automated Design Features from Oracle, DB2, and SQL Server	229
12.2.1 IBM DB2 Design Advisor	231
12.2.2 Microsoft SQL Server Database Tuning Advisor	234
12.2.3 Oracle SQL Access Advisor	238
12.3 Data Sampling for Improved Statistics during Analysis	240
12.4 Scalability and Workload Compression	242
12.5 Design Exploration between Test and Production Systems	247
12.6 Experimental Results from Published Literature	248
12.7 Index Selection	254
12.8 Materialized View Selection	254
12.9 Multidimensional Clustering Selection	256
12.10 Shared-nothing Partitioning	258
12.11 Range Partitioning Design	260
12.12 Summary	262
12.13 Literature Summary	262
13 Down to the Metal: Server Resources and Topology	265
13.1 What You Need to Know about CPU Architecture and Trends	266
13.1.1 CPU Performance	266
13.1.2 Amdahl's Law for System Speedup with Parallel Processing	269
13.1.3 Multicore CPUs	271
13.2 Client Server Architectures	271
13.3 Symmetric Multiprocessors and NUMA	273
13.3.1 Symmetric Multiprocessors and NUMA	273
13.3.2 Cache Coherence and False Sharing"	274
13.4 Server Clusters	275
13.5 A Little about Operating Systems	275
13.6 Storage Systems	276
13.6.1 Disks, Spindles, and Striping	277
13.6.2 Storage Area Networks and Network Attached Storage	278
13.7 Making Storage Both Reliable and Fast Using RAID	279
13.7.1 History of RAID	279
13.7.2 RAID0	281

13.7.3 RAID 1 ..	281
13.7.4 RAID 2 and RAID 3	282
13.7.5 RAID 4	284
13.7.6 RAID 5 and RAID 6	284
13.7.7 RAID 1+0	285
13.7.8 RAID 0+1	285
13.7.9 RAID 10+0 and RAID 5+0	286
13.7.10 Which RAID Is Right for Your Database Requirements?	288
13.8 Balancing Resources in a Database Server	288
13.9 Strategies for Availability and Recovery	290
13.10 Main Memory and Database Tuning	295
13.10.1 Memory Tuning by Mere Mortals	295
13.10.2 Automated Memory Tuning	298
13.10.3 Cutting Edge:The Latest Strategy in Self-tuningMemory Management	301
13.11 Summary	314
13.12 Literature Summary	314

14 Physical Design for Decision Support, Warehousing, and OLAP 317

14.1 What Is OLAP?	318
14.2 Dimension Hierarchies	320
14.3 Star and Snowflake Schemas	321
14.4 Warehouses and Marts	323
14.5 Scaling Up the System	327
14.6 DSS,Warehousing, and OLAP Design Considerations	328
14.7 Usage Syntax and Examples for Major Database Servers	329
14.7.1 Oracle	330
14.7.2 Microsoft's Analysis Services	331
14.8 Summary	333
14.9 Literature Summary	334

15 Denormalization 3 37

15.1 Basics of Normalization	338
15.2 Common Types of Denormalization	342
15.2.1 Two Entities in a One-to-One Relationship	342
15.2.2 Two Entities in a One-to-many Relationship	343
15.3 Table Denormalization Strategy	346
15.4 Example of Denormalization	347
15.4.1 Requirements Specification	347

15.4.2	Logical Design	349
15.4.3	Schema Refinement Using Denormalization	350
15.5	Summary	354
15.6	Literature Summary	354
16	Distributed Data Allocation	357
16.1	Introduction	358
16.2	Distributed Database Allocation	360
16.3	Replicated Data Allocation—"All-beneficial Sites"	
	Method	362
16.3.1	Example	362
16.4	Progressive Table Allocation Method	367
16.5	Summary	368
16.6	Literature Summary	369
Appendix A	A Simple Performance Model for Databases	371
A.1	I/O Time Cost—Individual Block Access	371
A.2	I/O Time Cost—Table Scans and Sorts	372
A.3	Network Time Delays	372
A.4	CPU Time Delays	374
Appendix B	Technical Comparison of DB2 HADR with Oracle Data Guard for Database Disaster Recovery	375
B.1	Standby Remains "Hot" during Failover	376
B.2	Subminute Failover	377
B.3	Geographically Separated	377
B.4	Support for Multiple Standby Servers	377
B.5	Support for Read on the Standby Server	377
B.6	Primary Can Be Easily Reintegrated after Failover	378
	Glossary	379
	Bibliography	391
	Index	411
	About the Authors	427