

# Contents

ACKNOWLEDGMENTS .....	xxiii
INTRODUCTION .....	xxv

## PART I

### Background

<b>I Introduction to SQL .....</b>	<b>3</b>
The Basic Nature of SQL .....	4
The Beginnings of SQL .....	6
A Model for SQL .....	7
Select Operation .....	8
Project Operation .....	8
Join Operation .....	8
The SQL Optimizer .....	10
The Goals of SQL Tuning .....	10
SQL Tuning as a Phase of Oracle Tuning .....	11
Database Design and SQL Performance .....	13
The Barriers to SQL Tuning .....	14
The Process of SQL Tuning .....	16
Locate the High-Use SQL .....	16
Tune the SQL Statement .....	18
Make the Tuning Permanent .....	19

The Goals of SQL Tuning .....	20
The Guidelines for SQL Tuning .....	20
The SQL Tuning Toolkit .....	21
Reporting on SQL from the Library Cache .....	22
Conclusion .....	28
<b>2 Overview of Oracle SQL Extensions .....</b>	<b>29</b>
Oracle In-Line Views .....	30
Oracle Built-in Functions .....	35
BIFs and Function-Based Indexes .....	36
Oracle SQL and Object-Oriented Extensions .....	39
Abstract Datatypes and Oracle SQL .....	39
Violating the Relational Rules—Lists of Repeating Data Items .....	42
Varray Tables .....	44
Nesting of Tables .....	47
Performance of SQL Object Extensions .....	49
Conclusion .....	50
<b>3 Understanding SQL Execution .....</b>	<b>51</b>
Parsing the SQL Statement .....	53
Query Rewrite .....	54
Using cursor_sharing in Oracle8i and Oracle9i .....	55
Techniques to Reduce SQL Parsing .....	57
Generating the Execution Plan .....	57
Table Access Methods .....	59
Index Access Methods .....	61
Join Operations .....	65
Sorting of the SQL Result Set .....	70
Conclusion .....	72
<b>4 Overview of the SQL Optimizers .....</b>	<b>73</b>
Basic Optimizer Techniques .....	74
The Rule-Based Optimizer .....	76
The Cost-Based Optimizer .....	78
The Optimizer Modes .....	81
rule mode .....	81
choose mode .....	81
first_rows mode .....	82
all_rows mode .....	82
Which Is “Better”? Fast Execution Versus Minimal Resource Usage .....	82

Tuning with Rule-Based Optimization .....	84
Changing the Rule-Based Driving Table .....	84
When the Rule-Base Optimizer Fails to Use the Correct Index ..	85
Tuning with Cost-Based Optimization (CBO) .....	87
Invoking the Cost-Based Optimizer .....	89
Gathering Statistics for the CBO .....	89
Determining the Default Optimizer Mode .....	91
Migrating to the Cost-Based Optimizer .....	92
Retraining the Development Staff .....	93
Choose Your Cost-Based Optimizer Philosophy .....	93
Conclusion .....	96
<b>5 SQL Internal Processing .....</b>	<b>97</b>
Shared and Private SQL Areas .....	98
SGA Statistics for SQL .....	99
Inside the Library Cache .....	100
Reusable SQL Inside the Library Cache .....	101
Monitoring the Library Cache Miss Ratio .....	102
Monitoring Objects Within the Library Cache with STATSPACK ..	104
STATSPACK Reports for the Library Cache .....	104
Tuning Oracle SQL Sorting .....	106
Identifying High-Impact SQL in the Library Cache .....	112
A SQL Top-10 Report .....	113
Reporting on SQL from the Library Cache .....	116
Using the access.sql Script with STATSPACK .....	118
The access.sql Reports .....	118
The Full-table Scan Report .....	119
The Index Range Scan Report .....	121
The Index Unique Scan Report .....	121
The Full-index Scan Report .....	122
Limitations of the access.sql Reports .....	122
Conclusion .....	123
<b>6 Tuning SQL Table Access .....</b>	<b>125</b>
SQL Tuning and Full-Table Scans .....	127
Determining the Threshold for a Full-Table Scan .....	127
Finding Full-Table Scans .....	128
How the Optimizer Chooses a Full-Table Scan .....	129
When the CBO Chooses a False Full-Table Scan .....	130
Preventing Unwanted Full-Table Scans .....	131
Table Access via Indexes .....	138

Changing Table Access Methods .....	138
Changing Indexes from Unique to Nonunique .....	139
Rewriting SQL Statements to Change the Table Access Method .....	141
Resequencing Table Rows for Reducing I/O .....	144
Row Resequencing with Index Clusters .....	146
Row Resequencing with CTAS .....	149
Oracle Storage Parameters and Table Access Performance .....	152
The pctfree Storage Parameter .....	152
The pctused Storage Parameter .....	153
The freelists Storage Parameter .....	154
The freelist Groups Storage Parameter for OPS .....	154
Summary of Storage Parameter Rules .....	154
Freelist Management and Table Access Performance .....	156
Linking and Unlinking from the Freelists .....	157
Reducing Freelist Relinks .....	159
Conclusion .....	161

## PART II

### Basic SQL Tuning

<b>7 The Steps to Oracle SQL Tuning .....</b>	<b>165</b>
Goals of SQL Tuning .....	166
The SQL Tuning Process .....	168
Step 1: Identify High-Impact SQL .....	168
Ranking SQL Statements .....	169
Identifying High-Use SQL Statements .....	170
Using STATSPACK to Identify High-Impact SQL .....	171
Reporting on SQL from the Library Cache .....	174
Using Third-Party Tools to Locate Offensive SQL .....	180
Step 2: Extract and Explain the SQL Statement .....	181
Explaining a SQL Statement .....	181
Step 3: Tune the SQL Statement .....	185
An Actual Case-Study in SQL Tuning .....	185
Get the Full-Table Scan Report .....	186
Fast SQL Tuning with Third-Party Tools .....	189
Conclusion .....	190
<b>8 Understanding Oracle SQL Utilities .....</b>	<b>191</b>
Explaining a SQL Statement .....	192
Reading an Execution Plan .....	197
Running a Fast SQL Trace .....	198
The TKPROF Utility .....	200

Set the Environment for SQL Tracing .....	200
Generating the SQL Trace File .....	201
Formatting the Trace File .....	203
The TKPROF Report .....	205
Oracle's Center of Expertise (COE) SQL Analysis Report .....	207
Reporting on SQL from the Library Cache .....	210
Using the access.sql Script with STATSPACK .....	212
The access.sql Reports .....	212
Conclusion .....	216
<b>9 Locating Significant SQL Statements .....</b>	<b>217</b>
Establishing Your Instance-Wide SQL Baseline .....	218
Setting the Default Optimizer_Mode .....	218
Making the Default Optimizer Mode Permanent .....	220
Instance-Wide Parameters to Improve the Speed of Full-Table Scans .....	220
Other Initialization Parameters that Affect SQL Execution .....	221
Running the SQL Baseline Test .....	222
What Constitutes a Significant SQL Statement? .....	222
Techniques for Extracting Significant SQL for Tuning .....	223
Identifying High-Use SQL Statements .....	223
Extraction Scripts for SQL Statements .....	225
Ongoing Ranking of SQL Statements .....	225
Finding New SQL Without Hints .....	226
Finding New SQL When Using Optimizer Plan Stability .....	229
Conclusion .....	229
<b>10 Tuning Full-Table Scans and Parallel Query .....</b>	<b>231</b>
Evaluating the Legitimacy of a Full-Table Scan .....	232
Finding Candidate Tables for Oracle Parallel Query .....	234
Using the KEEP Pool .....	235
Introduction to Oracle Parallel Query .....	235
Invoking Oracle Parallel Query .....	236
Oracle Parallel Query Initialization Parameters .....	237
Setting the Optimal Degree of Parallelism .....	238
Using parallel query Hints .....	244
Parallel Queries and Table Joins .....	245
Monitoring the Use of Oracle Parallel Query .....	253
Monitoring Parallel Execution Activity .....	253
Monitoring Oracle Parallel Query with STATSPACK .....	254
Monitoring Oracle Parallel Query with V\$ Views .....	256
Parallel Queries and Distributed Tables .....	256
Conclusion .....	258

<b>11</b>	<b>Optimizing Sorting for Oracle SQL Statements</b>	<b>261</b>
	Initialization Parameters Relating to Oracle Sorting	263
	The sort_area_size Parameter	263
	The sort_area_retained_size Parameter	264
	The sort_multiblock_read_count Parameter	264
	Obsolete Oracle8 Sort Parameters	265
	Optimizer Mode and Sort Activity	265
	Removing Sorts by Adding Indexes	265
	Unnecessary Sorts	266
	Monitoring Sorting Activity	266
	Trend Reporting for Oracle Sorts	267
	Conclusion	270
<b>12</b>	<b>Tuning with Oracle Hints</b>	<b>273</b>
	Introduction and History of Hints	274
	Specifying Hints in SQL Queries	275
	Optimizer Hints	276
	The all_rows Hint	277
	The rule Hint	277
	The first_rows Hint	277
	Table Join Hints	277
	The use_hash Hint	278
	The use_merge Hint	280
	The use_nl Hint	281
	The star Hint	282
	Table Anti-Join Hints	283
	The merge_aj Hint	284
	The hash_aj Hint	286
	Index Hints	287
	The Index Hint	287
	The index_join Hint	288
	The and_equal Hint	288
	The index_asc Hint	290
	The no_index Hint	290
	The index_desc Hint	291
	The index_combine Hint	291
	The index_ffs Hint	293
	The use_concat Hint	295
	Parallel Hints	298
	The parallel Hint	298
	The pq_distribute Hint	298
	The noparallel Hint	299

Table Access Hints	300
The full Hint	300
Cluster Table Hints	300
The no_expand Hint	301
The nocache Hint	301
The ordered Hint	302
The ordered_predicates Hint	302
The push_subq Hint	303
Hints in Subqueries	303
Conclusion	307
<b>13 Tuning with Optimizer Plan Stability</b>	<b>309</b>
Introduction to Stored Outlines	310
The Philosophy Behind Optimizer Plan Stability	312
Using Optimizer Plan Stability When Migrating to the CBO	313
Preparing Oracle for Stored Outlines	313
Setting the use_stored_outlines Command	314
Verifying Important Initialization Parameters	314
Create the Outline Package	315
Create the Stored Outline Tablespace	316
How to Create and Modify a Stored Outline	317
Find the Fastest Execution Plan	317
Create the Stored Outline for the Original Query	319
Create a Stored Outline for the Hinted Query	321
Swap the Stored Outlines	322
Managing Stored Outlines	324
Using the Dictionary Views and Tables for Stored Outlines	324
Using the Outline Package	327
Managing Categories with Stored Outlines	328
Conclusion	332
<b>14 Tuning with the Cost-Based Optimizer</b>	<b>333</b>
Statistics and Cost-Based Optimization	334
Dynamic versus Static CBO Execution Plan Philosophy	335
Frequency of Statistics Gathering	336
Gathering Statistics for the CBO	338
Cost-Based Optimization and SQL Tuning	339
Cost-Based Table Joins	341
Tuning Cost-Based SQL Statements with Subqueries	342
Tuning Complex Boolean Queries	344
Initialization Parameters That Affect Cost-Based Optimizer Behavior	351
Initialization Parameters That Affect Table Joins	352
Initialization Parameters That Affect CBO Index Behavior	358
Conclusion	359

<b>15</b>	<b>Tuning with the Rule-Based Optimizer</b>	<b>361</b>
	Invoking Rule-Based Optimization	363
	The Problem of Using Choose as the Default Mode	363
	Using a Rule-Based Default Optimizer Mode	364
	The Driving Table Location Problem	364
	The Driving Table and Table Cardinality	366
	Tips for Tuning Rule-Based Queries	368
	When the Rule-Based Optimizer Fails	369
	When the Rule-Based Optimizer Is Best	369
	Conclusion	372
<b>16</b>	<b>Tuning Table Joins</b>	<b>373</b>
	The Table Join Types	374
	The Equi-Join	375
	The Outer Join	376
	The Self-Join	378
	The Anti-Join	381
	The Semi-Join	383
	Oracle Table Join Methods	387
	Nested Loop Joins	387
	The use_nl Hint	390
	The Hash Join	390
	The Sort Merge Join	395
	The Star Join	399
	Evaluating Table Join Order	404
	Tuning Distributed SQL Table Joins	405
	Always Use the CBO for Distributed Joins	407
	Viewing the Execution Plan for Distributed Joins	407
	Tuning Guidelines for Distributed Joins	409
	Conclusion	410

## PART III

### Advanced SQL Tuning

<b>17</b>	<b>Tuning SQL DML Statements</b>	<b>415</b>
	Oracle Storage Parameters and DML Performance	416
	The pctfree Storage Parameter	417
	The pctused Storage Parameter	417
	The freelists Parameter and DML performance	418
	The freelist groups Storage Parameter for OPS	418
	Summary of Storage Parameter Rules	419

Freelist Management and DML Performance .....	420
Linking and Unlinking from the Freelists .....	421
Reducing Freelist Relink Operations .....	424
Long Data Columns and DML Behavior .....	424
Setting pctfree and pctused by Average Row Length .....	426
Buffer Busy Waits and DML Contention .....	428
Using STATSPACK to Find DML Wait Contention .....	428
Finding Buffer Busy Waits with STATSPACK .....	431
Update SQL, Subqueries, and Parallel DML .....	439
Overhead of Constraints on DML Performance .....	442
Overhead of Maintaining Indexes with DML .....	444
Using PL/SQL Bulk Inserts to Improve SQL Insert Speed .....	445
Conclusion .....	447
<b>18 Tuning SQL with Temporary Tables .....</b>	<b>449</b>
Using CTAS with Dictionary Views .....	450
Tuning Aggregation Queries with Temporary Tables .....	451
Conclusion .....	460
<b>19 Tuning SQL Subqueries .....</b>	<b>461</b>
Basics of Oracle Subqueries .....	462
Correlated Versus Noncorrelated Subqueries .....	465
Issues of Scale and Subqueries .....	466
Timing Subquery Execution .....	466
Basic Characteristics of Subquery Execution .....	469
Automatic SQL Transformation of Subqueries .....	470
Tuning Subqueries with the IN and EXISTS Clauses .....	471
Noncorrelated Subqueries Using the IN Clause .....	471
Correlated Subqueries with the IN Clause .....	477
Noncorrelated Subqueries with the EXISTS Clause .....	477
Correlated Subqueries Using the EXISTS Clause .....	478
Tuning Subqueries with NOT IN and NOT EXISTS Clauses .....	480
Noncorrelated Subqueries Using the NOT IN Operator .....	480
Correlated Subqueries with the NOT IN Operator .....	482
Correlated Subqueries with the NOT EXISTS Operator .....	483
Noncorrelated Subqueries with the NOT EXISTS Operator .....	485
Tuning Subqueries with Nonequality Conditions .....	486
Using the ANY or ALL Condition in a Subquery .....	486
Using Nonequality Conditions in Subqueries .....	489
Hints to Improve Subquery Execution Speed .....	492
Conclusion .....	492

<b>20</b>	<b>Tuning SQL with Indexes</b>	<b>493</b>
	Finding Indexing Opportunities	494
	Extract the SQL Statement	496
	Add the Index	497
	Evaluate the Legitimacy of the Full-Table Scan	497
	Time the Query with the New Index	501
	Predict the Benefit for the Whole Database	501
	Removing Sorts by Adding Indexes	502
	Unnecessary Sorts	502
	Index Predicates	502
	Evaluating Index Usage for Skewed Column Values	506
	Indexing Alternatives to B-Tree Indexes	508
	Bitmap Indexes	508
	Function-Based Indexes	508
	Reverse-Key Indexes and SQL Performance	509
	Index Usage for Queries with IN Conditions	509
	Conclusion	511
<b>21</b>	<b>Tuning Data Warehouse SQL</b>	<b>513</b>
	Large Table Join Tuning	514
	The ordered Hint	515
	The ordered_predicates Hint	516
	Oracle Partitioning and Oracle SQL Tuning	516
	Partitioning and SQL Table Joining	517
	Partitioning and SQL Execution Speed	517
	Oracle Parallel Query and Oracle SQL Tuning	518
	parallel Query init.ora Parameters	519
	Setting the Optimal Degree of Parallelism	520
	Using parallel query Hints	522
	Oracle Optimization of Data Warehouse Queries	525
	The optimizer_search_limit Parameter	526
	The optimizer_max_permutations Parameter	527
	Conclusion	529
<b>22</b>	<b>SQL Tuning with STATSPACK</b>	<b>531</b>
	STATSPACK and SQL Data Collection	533
	Tuning Oracle SQL Sorting with STATSPACK	534
	Tracking SQL with STATSPACK	541
	SQL Snapshot Thresholds	541
	STATSPACK SQL Top-10 Report	542
	Conclusion	545

<b>23</b>	<b>Tuning SQL with Built-in Functions and Special Operators</b>	<b>547</b>
	Using the like and case Clauses in SQL	548
	Using the like Clause in Oracle Queries	548
	Combining Multiples Scans with CASE Statements	550
	Tuning SQL with BIFs	552
	Using BIFs with Character Datatypes	553
	Using BIFs with Date Datatypes	553
	Using the substr BIF	557
	Oracle9i BIF Enhancements	558
	Conclusion	559
	<b>Book Conclusion</b>	<b>560</b>
	 Index.	 <b>561</b>