

Contents

Preface

Audience	xxv
Documentation Accessibility	xxv
Related Documents	xxv
Conventions	xxvi

Part I SQL Performance Fundamentals

1 Introduction to SQL Tuning

About SQL Tuning	1-1
Purpose of SQL Tuning	1-1
Prerequisites for SQL Tuning	1-1
Tasks and Tools for SQL Tuning	1-2
SQL Tuning Tasks	1-2
SQL Tuning Tools	1-4
Automated SQL Tuning Tools	1-4
Manual SQL Tuning Tools	1-13
User Interfaces to SQL Tuning Tools	1-15
About Automatic Error Mitigation	1-16

2 SQL Performance Methodology

Guidelines for Designing Your Application	2-1
Guideline for Data Modeling	2-1
Guideline for Writing Efficient Applications	2-1
Guidelines for Deploying Your Application	2-3
Guideline for Deploying in a Test Environment	2-3
Guidelines for Application Rollout	2-4

Part II Query Optimizer Fundamentals

3 SQL Processing

About SQL Processing	3-1
SQL Parsing	3-2
Syntax Check	3-2
Semantic Check	3-2
Shared Pool Check	3-3
SQL Optimization	3-5
SQL Row Source Generation	3-5
SQL Execution	3-6
How Oracle Database Processes DML	3-8
How Row Sets Are Fetched	3-8
Read Consistency	3-8
Data Changes	3-9
How Oracle Database Processes DDL	3-9

4 Query Optimizer Concepts

Introduction to the Query Optimizer	4-1
Purpose of the Query Optimizer	4-1
Cost-Based Optimization	4-1
Execution Plans	4-2
Query Blocks	4-2
Query Subplans	4-3
Analogy for the Optimizer	4-3
About Optimizer Components	4-4
Query Transformer	4-4
Estimator	4-5
Selectivity	4-6
Cardinality	4-7
Cost	4-8
Plan Generator	4-8
About Automatic Tuning Optimizer	4-10
About Adaptive Query Optimization	4-11
Adaptive Query Plans	4-11
About Adaptive Query Plans	4-12
Purpose of Adaptive Query Plans	4-12
How Adaptive Query Plans Work	4-13
When Adaptive Query Plans Are Enabled	4-20
Adaptive Statistics	4-20
Dynamic Statistics	4-20
Automatic Reoptimization	4-21

SQL Plan Directives	4-23
When Adaptive Statistics Are Enabled	4-24
About Approximate Query Processing	4-24
Approximate Query Initialization Parameters	4-25
Approximate Query SQL Functions	4-26
About SQL Plan Management	4-28
About Quarantined SQL Plans	4-28
About the Expression Statistics Store (ESS)	4-29

5 Query Transformations

OR Expansion	5-1
View Merging	5-3
Query Blocks in View Merging	5-3
Simple View Merging	5-4
Complex View Merging	5-6
Predicate Pushing	5-9
Subquery Unnesting	5-10
Query Rewrite with Materialized Views	5-11
About Query Rewrite and the Optimizer	5-11
About Initialization Parameters for Query Rewrite	5-12
About the Accuracy of Query Rewrite	5-12
Example of Query Rewrite	5-13
Star Transformation	5-14
About Star Schemas	5-14
Purpose of Star Transformations	5-15
How Star Transformation Works	5-15
Controls for Star Transformation	5-15
Star Transformation: Scenario	5-16
Temporary Table Transformation: Scenario	5-19
In-Memory Aggregation (VECTOR GROUP BY)	5-21
Cursor-Duration Temporary Tables	5-21
Purpose of Cursor-Duration Temporary Tables	5-21
How Cursor-Duration Temporary Tables Work	5-21
Cursor-Duration Temporary Tables: Example	5-22
Table Expansion	5-23
Purpose of Table Expansion	5-23
How Table Expansion Works	5-23
Table Expansion: Scenario	5-24
Table Expansion and Star Transformation: Scenario	5-27
Join Factorization	5-29
Purpose of Join Factorization	5-29

How Join Factorization Works	5-29
Factorization and Join Orders: Scenario	5-30
Factorization of Outer Joins: Scenario	5-31

Part III Query Execution Plans

6 Explaining and Displaying Execution Plans

Introduction to Execution Plans	6-1
Contents of an Execution Plan	6-1
Why Execution Plans Change	6-2
Different Schemas	6-2
Different Costs	6-2
Generating Plan Output Using the EXPLAIN PLAN Statement	6-3
About the EXPLAIN PLAN Statement	6-3
About PLAN_TABLE	6-3
EXPLAIN PLAN Restrictions	6-4
Explaining a SQL Statement: Basic Steps	6-4
Specifying a Statement ID in EXPLAIN PLAN: Example	6-7
Specifying a Different Location for EXPLAIN PLAN Output: Example	6-7
EXPLAIN PLAN Output for a CONTAINERS Query: Example	6-8
Displaying Execution Plans	6-9
About the Display of PLAN_TABLE Output	6-9
DBMS_XPLAN Display Functions	6-10
Plan-Related Views	6-12
Displaying Execution Plans: Basic Steps	6-13
Displaying Adaptive Query Plans: Tutorial	6-15
Display Execution Plans: Examples	6-19
Customizing PLAN_TABLE Output	6-19
Displaying Parallel Execution Plans: Example	6-20
Displaying Bitmap Index Plans: Example	6-22
Displaying Result Cache Plans: Example	6-23
Displaying Plans for Partitioned Objects: Example	6-24
Comparing Execution Plans	6-32
Purpose of Plan Comparison	6-32
User Interface for Plan Comparison	6-32
Comparing Execution Plans: Tutorial	6-36
Comparing Execution Plans: Examples	6-39

7 PLAN_TABLE Reference

PLAN_TABLE Columns	7-1
--------------------	-----

OPERATION and OPTION Columns of PLAN_TABLE	7-4
DISTRIBUTION Column of PLAN_TABLE	7-9

Part IV SQL Operators: Access Paths and Joins

8 Optimizer Access Paths

Introduction to Access Paths	8-1
Table Access Paths	8-2
About Heap-Organized Table Access	8-2
Row Storage in Data Blocks and Segments: A Primer	8-2
Importance of Rowids for Row Access	8-3
Direct Path Reads	8-3
Full Table Scans	8-4
When the Optimizer Considers a Full Table Scan	8-4
How a Full Table Scan Works	8-6
Full Table Scan: Example	8-7
Table Access by Rowid	8-7
When the Optimizer Chooses Table Access by Rowid	8-8
How Table Access by Rowid Works	8-8
Table Access by Rowid: Example	8-8
Sample Table Scans	8-9
When the Optimizer Chooses a Sample Table Scan	8-9
Sample Table Scans: Example	8-9
In-Memory Table Scans	8-10
When the Optimizer Chooses an In-Memory Table Scan	8-10
In-Memory Query Controls	8-10
In-Memory Table Scans: Example	8-11
B-Tree Index Access Paths	8-12
About B-Tree Index Access	8-12
B-Tree Index Structure	8-12
How Index Storage Affects Index Scans	8-13
Unique and Nonunique Indexes	8-14
B-Tree Indexes and Nulls	8-14
Index Unique Scans	8-16
When the Optimizer Considers Index Unique Scans	8-16
How Index Unique Scans Work	8-17
Index Unique Scans: Example	8-18
Index Range Scans	8-19
When the Optimizer Considers Index Range Scans	8-19
How Index Range Scans Work	8-20

Index Range Scan: Example	8-21
Index Range Scan Descending: Example	8-22
Index Full Scans	8-23
When the Optimizer Considers Index Full Scans	8-23
How Index Full Scans Work	8-23
Index Full Scans: Example	8-24
Index Fast Full Scans	8-25
When the Optimizer Considers Index Fast Full Scans	8-25
How Index Fast Full Scans Work	8-25
Index Fast Full Scans: Example	8-25
Index Skip Scans	8-26
When the Optimizer Considers Index Skip Scans	8-26
How Index Skip Scans Work	8-26
Index Skip Scans: Example	8-26
Index Join Scans	8-28
When the Optimizer Considers Index Join Scans	8-28
How Index Join Scans Work	8-29
Index Join Scans: Example	8-29
Bitmap Index Access Paths	8-30
About Bitmap Index Access	8-30
Differences Between Bitmap and B-Tree Indexes	8-30
Purpose of Bitmap Indexes	8-31
Bitmaps and Rowids	8-32
Bitmap Join Indexes	8-33
Bitmap Storage	8-35
Bitmap Conversion to Rowid	8-35
When the Optimizer Chooses Bitmap Conversion to Rowid	8-35
How Bitmap Conversion to Rowid Works	8-35
Bitmap Conversion to Rowid: Example	8-35
Bitmap Index Single Value	8-36
When the Optimizer Considers Bitmap Index Single Value	8-36
How Bitmap Index Single Value Works	8-36
Bitmap Index Single Value: Example	8-36
Bitmap Index Range Scans	8-37
When the Optimizer Considers Bitmap Index Range Scans	8-37
How Bitmap Index Range Scans Work	8-37
Bitmap Index Range Scans: Example	8-38
Bitmap Merge	8-39
When the Optimizer Considers Bitmap Merge	8-39
How Bitmap Merge Works	8-39
Bitmap Merge: Example	8-39
Table Cluster Access Paths	8-40

Cluster Scans	8-40
When the Optimizer Considers Cluster Scans	8-40
How a Cluster Scan Works	8-40
Cluster Scans: Example	8-41
Hash Scans	8-42
When the Optimizer Considers a Hash Scan	8-42
How a Hash Scan Works	8-42
Hash Scans: Example	8-42

9 Joins

About Joins	9-1
Join Trees	9-1
How the Optimizer Executes Join Statements	9-3
How the Optimizer Chooses Execution Plans for Joins	9-4
Join Methods	9-5
Nested Loops Joins	9-5
When the Optimizer Considers Nested Loops Joins	9-6
How Nested Loops Joins Work	9-6
Nested Nested Loops	9-7
Current Implementation for Nested Loops Joins	9-10
Original Implementation for Nested Loops Joins	9-12
Nested Loops Controls	9-14
Hash Joins	9-15
When the Optimizer Considers Hash Joins	9-16
How Hash Joins Work	9-16
How Hash Joins Work When the Hash Table Does Not Fit in the PGA	9-18
Hash Join Controls	9-19
Sort Merge Joins	9-19
When the Optimizer Considers Sort Merge Joins	9-20
How Sort Merge Joins Work	9-20
Sort Merge Join Controls	9-24
Join Types	9-24
Inner Joins	9-24
Equijoins	9-24
Nonequijoins	9-25
Band Joins	9-26
Outer Joins	9-29
Nested Loops Outer Joins	9-30
Hash Join Outer Joins	9-30
Sort Merge Outer Joins	9-32
Full Outer Joins	9-32

Multiple Tables on the Left of an Outer Join	9-33
Semijoins	9-34
When the Optimizer Considers Semijoins	9-34
How Semijoins Work	9-34
Antijoins	9-36
When the Optimizer Considers Antijoins	9-36
How Antijoins Work	9-37
How Antijoins Handle Nulls	9-38
Cartesian Joins	9-41
When the Optimizer Considers Cartesian Joins	9-41
How Cartesian Joins Work	9-42
Cartesian Join Controls	9-42
Join Optimizations	9-43
Bloom Filters	9-43
Purpose of Bloom Filters	9-43
How Bloom Filters Work	9-44
Bloom Filter Controls	9-45
Bloom Filter Metadata	9-45
Bloom Filters: Scenario	9-45
Partition-Wise Joins	9-47
Purpose of Partition-Wise Joins	9-47
How Partition-Wise Joins Work	9-47
In-Memory Join Groups	9-50

Part V Optimizer Statistics

10 Optimizer Statistics Concepts

Introduction to Optimizer Statistics	10-1
About Optimizer Statistics Types	10-2
Table Statistics	10-2
Permanent Table Statistics	10-2
Temporary Table Statistics	10-3
Column Statistics	10-6
Index Statistics	10-7
Types of Index Statistics	10-7
Index Clustering Factor	10-8
Effect of Index Clustering Factor on Cost: Example	10-12
System Statistics	10-12
User-Defined Optimizer Statistics	10-13
How the Database Gathers Optimizer Statistics	10-13

DBMS_STATS Package	10-13
Supplemental Dynamic Statistics	10-14
Online Statistics Gathering	10-15
Online Statistics Gathering for Bulk Loads	10-15
Online Statistics Gathering for Partition Maintenance Operations	10-17
Real-Time Statistics	10-18
When the Database Gathers Optimizer Statistics	10-25
Sources for Optimizer Statistics	10-26
SQL Plan Directives	10-26
When the Database Creates SQL Plan Directives	10-27
How the Database Uses SQL Plan Directives	10-27
SQL Plan Directive Maintenance	10-28
How the Optimizer Uses SQL Plan Directives: Example	10-29
How the Optimizer Uses Extensions and SQL Plan Directives: Example	10-33
When the Database Samples Data	10-37
How the Database Samples Data	10-39

11 Histograms

Purpose of Histograms	11-1
When Oracle Database Creates Histograms	11-1
How Oracle Database Chooses the Histogram Type	11-3
Cardinality Algorithms When Using Histograms	11-4
Endpoint Numbers and Values	11-4
Popular and Nonpopular Values	11-4
Bucket Compression	11-5
Frequency Histograms	11-5
Criteria For Frequency Histograms	11-6
Generating a Frequency Histogram	11-6
Top Frequency Histograms	11-9
Criteria For Top Frequency Histograms	11-9
Generating a Top Frequency Histogram	11-10
Height-Balanced Histograms (Legacy)	11-13
Criteria for Height-Balanced Histograms	11-13
Generating a Height-Balanced Histogram	11-14
Hybrid Histograms	11-17
How Endpoint Repeat Counts Work	11-17
Criteria for Hybrid Histograms	11-19
Generating a Hybrid Histogram	11-20

12 Configuring Options for Optimizer Statistics Gathering

Purpose of Optimizer Statistics Collection	12-1
User Interfaces for Optimizer Statistics Management	12-1
Graphical Interface for Optimizer Statistics Management	12-1
Accessing the Database Home Page in Cloud Control	12-1
Accessing the Optimizer Statistics Console	12-2
Command-Line Interface for Optimizer Statistics Management	12-2
Setting Optimizer Statistics Preferences	12-2
About Optimizer Statistics Preferences	12-2
Purpose of Optimizer Statistics Preferences	12-2
Examples of Statistics Preferences	12-3
DBMS_STATS Procedures for Setting Statistics Preferences	12-4
Statistics Preference Overrides	12-5
Setting Statistics Preferences: Example	12-7
Setting Global Optimizer Statistics Preferences Using Cloud Control	12-9
Setting Object-Level Optimizer Statistics Preferences Using Cloud Control	12-9
Setting Optimizer Statistics Preferences from the Command Line	12-10
Configuring Options for Dynamic Statistics	12-11
About Dynamic Statistics Levels	12-11
Setting Dynamic Statistics Levels Manually	12-13
Disabling Dynamic Statistics	12-15
Dynamic Statistics for PL/SQL Functions	12-16
The Use Case for PL/SQL Dynamic Statistics	12-16
Controls for Dynamic Sampling With PL/SQL Functions	12-17
Setting Global Level PL/SQL Preferences	12-18
Setting Function-Level Preferences	12-19
Setting the PLSQL_FUNCTION_DYNAMIC_STATS Session Level Parameter	12-20
Managing SQL Plan Directives	12-20

13 Gathering Optimizer Statistics

Configuring Automatic Optimizer Statistics Collection	13-1
About Automatic Optimizer Statistics Collection	13-1
Configuring Automatic Optimizer Statistics Collection Using Cloud Control	13-2
Configuring Automatic Optimizer Statistics Collection from the Command Line	13-3
Configuring High-Frequency Automatic Optimizer Statistics Collection	13-5
About High-Frequency Automatic Optimizer Statistics Collection	13-5
Setting Preferences for High-Frequency Automatic Optimizer Statistics Collection	13-6
High-Frequency Automatic Optimizer Statistics Collection: Example	13-7
Gathering Optimizer Statistics Manually	13-10
About Manual Statistics Collection with DBMS_STATS	13-10

Guidelines for Gathering Optimizer Statistics Manually	13-11
Guideline for Setting the Sample Size	13-11
Guideline for Gathering Statistics in Parallel	13-12
Guideline for Partitioned Objects	13-13
Guideline for Frequently Changing Objects	13-13
Guideline for External Tables	13-13
Determining When Optimizer Statistics Are Stale	13-14
Gathering Schema and Table Statistics	13-15
Gathering Statistics for Fixed Objects	13-16
Gathering Statistics for Volatile Tables Using Dynamic Statistics	13-17
Gathering Optimizer Statistics Concurrently	13-18
About Concurrent Statistics Gathering	13-18
Enabling Concurrent Statistics Gathering	13-20
Monitoring Statistics Gathering Operations	13-23
Gathering Incremental Statistics on Partitioned Objects	13-25
Purpose of Incremental Statistics	13-25
How DBMS_STATS Derives Global Statistics for Partitioned tables	13-25
Gathering Statistics for a Partitioned Table: Basic Steps	13-28
Maintaining Incremental Statistics for Partition Maintenance Operations	13-31
Maintaining Incremental Statistics for Tables with Stale or Locked Partition Statistics	13-34
Gathering System Statistics Manually	13-36
About System Statistics	13-36
Guidelines for Gathering System Statistics	13-37
Gathering System Statistics with DBMS_STATS	13-38
About the GATHER_SYSTEM_STATS Procedure	13-38
Gathering Workload Statistics	13-40
Gathering Noworkload Statistics	13-43
Deleting System Statistics	13-44
Running Statistics Gathering Functions in Reporting Mode	13-45

14 Managing Extended Statistics

Managing Column Group Statistics	14-1
About Statistics on Column Groups	14-1
Why Column Group Statistics Are Needed: Example	14-2
Automatic and Manual Column Group Statistics	14-4
User Interface for Column Group Statistics	14-4
Detecting Useful Column Groups for a Specific Workload	14-5
Creating Column Groups Detected During Workload Monitoring	14-8
Creating and Gathering Statistics on Column Groups Manually	14-10
Displaying Column Group Information	14-11
Dropping a Column Group	14-12

Managing Expression Statistics	14-13
About Expression Statistics	14-13
When Expression Statistics Are Useful: Example	14-14
Creating Expression Statistics	14-15
Displaying Expression Statistics	14-16
Dropping Expression Statistics	14-17

15 Controlling the Use of Optimizer Statistics

Locking and Unlocking Optimizer Statistics	15-1
Locking Statistics	15-1
Unlocking Statistics	15-2
Publishing Pending Optimizer Statistics	15-3
About Pending Optimizer Statistics	15-3
User Interfaces for Publishing Optimizer Statistics	15-5
Managing Published and Pending Statistics	15-6
Creating Artificial Optimizer Statistics for Testing	15-9
About Artificial Optimizer Statistics	15-9
Setting Artificial Optimizer Statistics for a Table	15-10
Setting Optimizer Statistics: Example	15-11

16 Managing Historical Optimizer Statistics

Restoring Optimizer Statistics	16-1
About Restore Operations for Optimizer Statistics	16-1
Guidelines for Restoring Optimizer Statistics	16-1
Restrictions for Restoring Optimizer Statistics	16-2
Restoring Optimizer Statistics Using DBMS_STATS	16-2
Managing Optimizer Statistics Retention	16-4
Obtaining Optimizer Statistics History	16-4
Changing the Optimizer Statistics Retention Period	16-5
Purging Optimizer Statistics	16-6
Reporting on Past Statistics Gathering Operations	16-7

17 Importing and Exporting Optimizer Statistics

About Transporting Optimizer Statistics	17-1
Purpose of Transporting Optimizer Statistics	17-1
How Transporting Optimizer Statistics Works	17-1
User Interface for Importing and Exporting Optimizer Statistics	17-2
Transporting Optimizer Statistics to a Test Database: Tutorial	17-3

18 Analyzing Statistics Using Optimizer Statistics Advisor

About Optimizer Statistics Advisor	18-1
Purpose of Optimizer Statistics Advisor	18-2
Problems with a Traditional Script-Based Approach	18-2
Advantages of Optimizer Statistics Advisor	18-3
Optimizer Statistics Advisor Concepts	18-3
Components of Optimizer Statistics Advisor	18-3
Operational Modes for Optimizer Statistics Advisor	18-7
Command-Line Interface to Optimizer Statistics Advisor	18-7
Basic Tasks for Optimizer Statistics Advisor	18-9
Creating an Optimizer Statistics Advisor Task	18-12
Listing Optimizer Statistics Advisor Tasks	18-13
Creating Filters for an Optimizer Advisor Task	18-13
About Filters for Optimizer Statistics Advisor	18-13
Creating an Object Filter for an Optimizer Advisor Task	18-14
Creating a Rule Filter for an Optimizer Advisor Task	18-17
Creating an Operation Filter for an Optimizer Advisor Task	18-19
Executing an Optimizer Statistics Advisor Task	18-23
Generating a Report for an Optimizer Statistics Advisor Task	18-24
Implementing Optimizer Statistics Advisor Recommendations	18-28
Implementing Actions Recommended by Optimizer Statistics Advisor	18-28
Generating a Script Using Optimizer Statistics Advisor	18-30

Part VI Optimizer Controls

19 Influencing the Optimizer

Techniques for Influencing the Optimizer	19-1
Influencing the Optimizer with Initialization Parameters	19-2
About Optimizer Initialization Parameters	19-2
Enabling Optimizer Features	19-8
Choosing an Optimizer Goal	19-9
Controlling Adaptive Optimization	19-10
Influencing the Optimizer with Hints	19-11
About Optimizer Hints	19-12
Purpose of Hints	19-12
Types of Hints	19-13
Scope of Hints	19-14
Guidelines for Join Order Hints	19-14
Reporting on Hints	19-15
Purpose of Hint Usage Reports	19-16

User Interface for Hint Usage Reports	19-16
Reporting on Hint Usage: Tutorial	19-18
Hint Usage Reports: Examples	19-20
SQL Analysis Report	19-29

20 Improving Real-World Performance Through Cursor Sharing

Overview of Cursor Sharing	20-1
About Cursors	20-1
Private and Shared SQL Areas	20-1
Parent and Child Cursors	20-3
About Cursors and Parsing	20-7
About Literals and Bind Variables	20-10
Literals and Cursors	20-10
Bind Variables and Cursors	20-12
Bind Variable Peeking	20-13
About the Life Cycle of Shared Cursors	20-16
Cursor Marked Invalid	20-16
Cursors Marked Rolling Invalid	20-18
CURSOR_SHARING and Bind Variable Substitution	20-20
CURSOR_SHARING Initialization Parameter	20-20
Parsing Behavior When CURSOR_SHARING = FORCE	20-21
Adaptive Cursor Sharing	20-23
Purpose of Adaptive Cursor Sharing	20-23
How Adaptive Cursor Sharing Works: Example	20-23
Bind-Sensitive Cursors	20-25
Bind-Aware Cursors	20-29
Cursor Merging	20-32
Adaptive Cursor Sharing Views	20-33
Real-World Performance Guidelines for Cursor Sharing	20-33
Develop Applications with Bind Variables for Security and Performance	20-33
Do Not Use CURSOR_SHARING = FORCE as a Permanent Fix	20-35
Establish Coding Conventions to Increase Cursor Reuse	20-36
Minimize Session-Level Changes to the Optimizer Environment	20-37

Part VII Monitoring and Tracing SQL

21 Gathering Diagnostic Data with SQL Test Case Builder

Purpose of SQL Test Case Builder	21-1
Concepts for SQL Test Case Builder	21-1
SQL Incidents	21-1

What SQL Test Case Builder Captures	21-2
Output of SQL Test Case Builder	21-3
User Interfaces for SQL Test Case Builder	21-5
Graphical Interface for SQL Test Case Builder	21-5
Accessing the Incident Manager	21-5
Accessing the Support Workbench	21-6
Command-Line Interface to SQL Tuning Advisor	21-6
Running SQL Test Case Builder	21-7

22 Monitoring Database Operations

About Monitoring Database Operations	22-1
About Database Operations	22-1
Purpose of Monitoring Database Operations	22-2
How Database Monitoring Works	22-3
User Interfaces for Database Operations Monitoring	22-4
Monitored SQL Executions Page in Cloud Control	22-4
DBMS_SQL_MONITOR Package	22-5
Attributes of composite Database Operations	22-6
MONITOR and NO_MONITOR Hints	22-7
Views for Monitoring and Reporting on Database Operations	22-8
Basic Tasks in Database Operations Monitoring	22-10
Enabling and Disabling Real-Time Monitoring of Database Operations	22-10
Enabling Monitoring of Database Operations at the System Level	22-10
Enabling and Disabling Monitoring of Database Operations at the Statement Level	22-11
Defining a Composite Database Operation	22-12
Generating and Accessing SQL Monitor Reports	22-15
Monitoring Database Operations: Scenarios	22-19
Reporting on a Simple Database Operation: Scenario	22-19
Reporting on Composite Database Operation: Scenario	22-22
Real-Time SQL Monitoring Across Multiple PDBs	22-24
Enabling or Disabling Real-Time SQL Monitoring on PDBs	22-25
Querying Real-Time SQL Monitor Dictionary Tables and Imported Snapshots	22-25
Views for Examining Real-Time SQL Monitoring	22-26
Reporting Tools for Real-Time SQL Monitoring	22-26
SQL History Monitoring and Reporting	22-27
Enabling and Viewing SQL History Monitoring and Reporting	22-28
DBMS_SQLTUNE APIs for Query History and Reporting	22-28

23 Performing Application Tracing

Overview of End-to-End Application Tracing	23-1
Purpose of End-to-End Application Tracing	23-1
End-to-End Application Tracing for PDBs	23-2
Tools for End-to-End Application Tracing	23-2
Overview of the SQL Trace Facility	23-3
Overview of TKPROF	23-4
Enabling Statistics Gathering for End-to-End Tracing	23-4
Enabling Statistics Gathering for a Client ID	23-4
Enabling Statistics Gathering for Services, Modules, and Actions	23-5
Enabling End-to-End Application Tracing	23-6
Enabling Tracing for a Client Identifier	23-6
Enabling Tracing for a Service, Module, and Action	23-7
Enabling Tracing for a Session	23-8
Enabling Tracing for an Instance or Database	23-9
Generating Output Files Using SQL Trace and TKPROF	23-10
Step 1: Setting Initialization Parameters for Trace File Management	23-10
Step 2: Enabling the SQL Trace Facility	23-11
Step 3: Generating Output Files with TKPROF	23-13
Step 4: Storing SQL Trace Facility Statistics	23-14
Generating the TKPROF Output SQL Script	23-14
Editing the TKPROF Output SQL Script	23-14
Querying the Output Table	23-14
Guidelines for Interpreting TKPROF Output	23-16
Guideline for Interpreting the Resolution of Statistics	23-16
Guideline for Recursive SQL Statements	23-16
Guideline for Deciding Which Statements to Tune	23-17
Guidelines for Avoiding Traps in TKPROF Interpretation	23-18
Guideline for Avoiding the Argument Trap	23-18
Guideline for Avoiding the Read Consistency Trap	23-18
Guideline for Avoiding the Schema Trap	23-18
Guideline for Avoiding the Time Trap	23-19
Application Tracing Utilities	23-20
TRCSESS	23-20
Purpose	23-20
Guidelines	23-20
Syntax	23-21
Options	23-21
Examples	23-21
TKPROF	23-22
Purpose	23-22

Guidelines	23-22
Syntax	23-23
Options	23-23
Output	23-25
Examples	23-28
Views for Application Tracing	23-32
Views Relevant for Trace Statistics	23-33
Views Related to Enabling Tracing	23-33
Controls Over Application Tracing and Access to Trace Events	23-33

Part VIII Automatic SQL Tuning

24 Capturing Workloads in SQL Tuning Sets

About SQL Tuning Sets	24-1
Purpose of SQL Tuning Sets	24-1
Concepts for SQL Tuning Sets	24-1
User Interfaces for SQL Tuning Sets	24-3
Accessing the SQL Tuning Sets Page in Cloud Control	24-3
Command-Line Interface to SQL Tuning Sets	24-3
Basic Tasks for Managing SQL Tuning Sets	24-4
Creating a SQL Tuning Set Using CREATE_SQLSET	24-6
Loading a SQL Tuning Set Using LOAD_SQLSET	24-7
Querying a SQL Tuning Set	24-8
Modifying a SQL Tuning Set Using UPDATE_SQLSET	24-11
Transporting a SQL Tuning Set	24-12
About Transporting SQL Tuning Sets	24-12
Basic Steps for Transporting SQL Tuning Sets	24-12
Basic Steps for Transporting SQL Tuning Sets When the CON_DBID Values Differ	24-13
Transporting SQL Tuning Sets with DBMS_SQLTUNE	24-14
Dropping a SQL Tuning Set Using DROP_SQLSET	24-16

25 Analyzing SQL with SQL Tuning Advisor

About SQL Tuning Advisor	25-1
Purpose of SQL Tuning Advisor	25-1
SQL Tuning Advisor Architecture	25-2
Input to SQL Tuning Advisor	25-3
Output of SQL Tuning Advisor	25-4
Automatic Tuning Optimizer Analyses	25-5
SQL Tuning Advisor Operation	25-14
Automatic and On-Demand SQL Tuning	25-14

Local and Remote SQL Tuning	25-15
Managing the Automatic SQL Tuning Task	25-20
About the Automatic SQL Tuning Task	25-20
Purpose of Automatic SQL Tuning	25-21
Automatic SQL Tuning Concepts	25-21
Command-Line Interface to SQL Tuning Advisor	25-21
Basic Tasks for Automatic SQL Tuning	25-22
Enabling and Disabling the Automatic SQL Tuning Task	25-23
Enabling and Disabling the Automatic SQL Tuning Task Using Cloud Control	25-23
Enabling and Disabling the Automatic SQL Tuning Task from the Command Line	25-24
Configuring the Automatic SQL Tuning Task	25-26
Configuring the Automatic SQL Tuning Task Using Cloud Control	25-26
Configuring the Automatic SQL Tuning Task Using the Command Line	25-27
Viewing Automatic SQL Tuning Reports	25-28
Viewing Automatic SQL Tuning Reports Using the Command Line	25-29
Running SQL Tuning Advisor On Demand	25-31
About On-Demand SQL Tuning	25-32
Purpose of On-Demand SQL Tuning	25-32
User Interfaces for On-Demand SQL Tuning	25-32
Basic Tasks in On-Demand SQL Tuning	25-34
Creating a SQL Tuning Task	25-35
Configuring a SQL Tuning Task	25-37
Executing a SQL Tuning Task	25-39
Monitoring a SQL Tuning Task	25-40
Displaying the Results of an SQL Tuning Task	25-41
The Automatic SQL Tuning Set	25-43

26 Optimizing Access Paths with SQL Access Advisor

About SQL Access Advisor	26-1
Purpose of SQL Access Advisor	26-1
SQL Access Advisor Architecture	26-2
Input to SQL Access Advisor	26-3
Filter Options for SQL Access Advisor	26-3
SQL Access Advisor Recommendations	26-4
SQL Access Advisor Actions	26-5
SQL Access Advisor Repository	26-6
User Interfaces for SQL Access Advisor	26-7
Accessing the SQL Access Advisor: Initial Options Page Using Cloud Control	26-7
Command-Line Interface to SQL Tuning Sets	26-8
Using SQL Access Advisor: Basic Tasks	26-8
Creating a SQL Tuning Set as Input for SQL Access Advisor	26-10

Populating a SQL Tuning Set with a User-Defined Workload	26-11
Creating and Configuring a SQL Access Advisor Task	26-13
Executing a SQL Access Advisor Task	26-15
Viewing SQL Access Advisor Task Results	26-16
Generating and Executing a Task Script	26-20
Performing a SQL Access Advisor Quick Tune	26-21
Using SQL Access Advisor: Advanced Tasks	26-22
Evaluating Existing Access Structures	26-22
Updating SQL Access Advisor Task Attributes	26-23
Creating and Using SQL Access Advisor Task Templates	26-24
Terminating SQL Access Advisor Task Execution	26-26
Interrupting SQL Access Advisor Tasks	26-26
Canceling SQL Access Advisor Tasks	26-27
Deleting SQL Access Advisor Tasks	26-28
Marking SQL Access Advisor Recommendations	26-29
Modifying SQL Access Advisor Recommendations	26-30
SQL Access Advisor Examples	26-31
SQL Access Advisor Reference	26-31
Action Attributes in the DBA_ADVISOR_ACTIONS View	26-31
Categories for SQL Access Advisor Task Parameters	26-33
SQL Access Advisor Constants	26-33

Part IX SQL Management Objects

27 Managing SQL Profiles

About SQL Profiles	27-1
Purpose of SQL Profiles	27-1
Concepts for SQL Profiles	27-2
Statistics in SQL Profiles	27-2
SQL Profiles and Execution Plans	27-2
SQL Profile Recommendations	27-3
SQL Profiles and SQL Plan Baselines	27-5
User Interfaces for SQL Profiles	27-6
Basic Tasks for SQL Profiles	27-6
Implementing a SQL Profile	27-7
About SQL Profile Implementation	27-7
Implementing a SQL Profile	27-8
Listing SQL Profiles	27-9
Altering a SQL Profile	27-10
Dropping a SQL Profile	27-11

28 Overview of SQL Plan Management

About SQL Plan Baselines	28-1
Purpose of SQL Plan Management	28-1
Benefits of SQL Plan Management	28-1
Differences Between SQL Plan Baselines and SQL Profiles	28-2
Plan Capture	28-3
Automatic Initial Plan Capture	28-3
Eligibility for Automatic Initial Plan Capture	28-4
Plan Matching for Automatic Initial Plan Capture	28-5
Manual Plan Capture	28-5
Plan Selection	28-7
Plan Evolution	28-8
Purpose of Plan Evolution	28-8
How Plan Evolution Works	28-8
PL/SQL Subprograms for Plan Evolution	28-9
About Automatic SQL Plan Management	28-10
Use Cases for Automatic SQL Plan Management	28-10
SQL Plan Management (With Background Verification)	28-10
About Real-Time SQL Plan Management	28-10
Storage Architecture for SQL Plan Management	28-11
SQL Management Base	28-11
SQL Statement Log	28-12
SQL Plan History	28-13
Enabled Plans	28-14
Accepted Plans	28-14
Fixed Plans	28-14

29 Managing SQL Plan Baselines

About Managing SQL Plan Baselines	29-1
User Interfaces for SQL Plan Management	29-1
Accessing the SQL Plan Baseline Page in Cloud Control	29-1
DBMS_SPM Package	29-2
Basic Tasks in SQL Plan Management	29-3
Configuring the Capture and Use of SQL Plan Baselines	29-4
Enabling Automatic Initial Plan Capture for SQL Plan Management	29-5
Configuring Filters for Automatic Plan Capture	29-6
Disabling All SQL Plan Baselines	29-8
Configuring SQL Plan Management	29-8

Managing the SPM Evolve Advisor Task	29-9
About the SPM Evolve Advisor Task	29-9
Enabling and Disabling the Automatic SPM Evolve Advisor Task	29-9
Configuring the Automatic SPM Evolve Advisor Task	29-10
Automatic SQL Plan Management	29-13
Configuring Automatic SQL Plan Management	29-15
Enabling Automatic SQL Plan Management: Tutorial	29-15
Monitoring and Reporting in Automatic SQL Plan Management	29-17
Displaying Plans in a SQL Plan Baseline	29-17
Loading SQL Plan Baselines	29-18
About Loading SQL Plan Baselines	29-18
Loading Plans from AWR	29-19
Loading Plans from the Shared SQL Area	29-22
Loading Plans from a SQL Tuning Set	29-24
Loading Plans from a Staging Table	29-26
Evolving SQL Plan Baselines Manually	29-29
About the DBMS_SPM Evolve Functions	29-29
Managing an Evolve Task	29-31
Adding SQL Plans to a Baseline	29-39
Dropping SQL Plan Baselines	29-40
Managing the SQL Management Base	29-42
About Managing the SMB	29-42
Changing the Disk Space Limit for the SMB	29-43
Changing the Plan Retention Policy in the SMB	29-44

30 Migrating Stored Outlines to SQL Plan Baselines

About Stored Outline Migration	30-1
Purpose of Stored Outline Migration	30-1
How Stored Outline Migration Works	30-2
Stages of Stored Outline Migration	30-2
Outline Categories and Baseline Modules	30-3
User Interface for Stored Outline Migration	30-4
Basic Steps in Stored Outline Migration	30-6
Preparing for Stored Outline Migration	30-6
Migrating Outlines to Utilize SQL Plan Management Features	30-7
Migrating Outlines to Preserve Stored Outline Behavior	30-8
Performing Follow-Up Tasks After Stored Outline Migration	30-9

Glossary
