

Index

A

- accepted plans, [4-28](#)
- access paths, [3-6](#), [8-1](#)
 - B-tree indexes, [8-12](#)
 - bitmap index, [8-30](#)
 - bitmap index range scans, [8-37](#)
 - bitmap indexes, [8-30](#), [8-31](#), [8-39](#)
 - execution plans, [6-1](#)
 - full table scans, [8-4](#), [10-8](#), [11-1](#)
 - heap-organized tables, [8-2](#)
 - In-Memory table scans, [8-10](#), [8-11](#)
 - index fast full, [8-25](#)
 - index full scans, [8-23](#)
 - index join scans, [8-28](#)
 - index range scans, [8-19](#)
 - index skip scans, [8-26](#)
 - index unique scans, [8-16](#)
 - sample table scans, [8-9](#)
 - table, [8-2](#)
 - table access by rowid, [8-7](#)
 - table cluster, [8-40](#)
- adaptive query optimization, [4-11](#)
 - adaptive plans, [21-2](#)
 - adaptive query plans, [4-11](#), [6-12](#), [6-15](#)
 - controlling, [19-10](#)
 - dynamic statistics, [10-14](#), [13-26](#)
- adaptive query plans, [4-11](#), [6-12](#), [6-15](#), [21-2](#)
 - about, [4-12](#)
 - bitmap index pruning, [4-18](#)
 - cardinality misestimates, [4-12](#)
 - enabling, [4-20](#)
 - join methods, [4-13](#)
 - optimizer statistics collector, [4-13](#)
 - parallel distribution methods, [4-16](#)
 - subplans, [4-13](#)
- adaptive sampling, [13-26](#)
- adaptive statistics, [4-20](#)
 - automatic reoptimization, [4-21](#)
 - dynamic statistics, [4-20](#)
 - SQL plan directives, [4-23](#), [14-1](#)
 - when enabled, [4-24](#)
- ADDM, [1-5](#)
- ADMINISTER ANY SQL TUNING SET system privilege, [24-3](#)
- advisors
 - Optimizer Statistics Advisor, [18-1](#), [18-2](#)
 - SPM Evolve Advisor, [29-8](#), [29-9](#)
 - SQL Access Advisor, [26-1](#), [26-7](#), [26-8](#)
 - SQL Tuning Advisor, [25-1](#)
- aggregate functions, [4-24](#)
- ALTER SESSION statement, [23-11](#)
- ANALYZE command, [16-2](#)
- antijoins, [9-4](#), [9-36](#)
 - handling nulls, [9-38](#)
 - how they work, [9-37](#)
 - when the optimizer considers, [9-36](#)
- APPEND hint, [10-15](#)
- applications
 - implementing, [2-3](#)
- APPROX_COUNT function, [4-26](#)
- APPROX_COUNT_DISTINCT function, [4-26](#)
- APPROX_COUNT_DISTINCT_AGG function, [4-26](#)
- APPROX_COUNT_DISTINCT_DETAIL function, [4-26](#)
- APPROX_FOR_AGGREGATION initialization parameter, [4-25](#)
- APPROX_FOR_COUNT_DISTINCT initialization parameter, [4-25](#)
- APPROX_FOR_PERCENTILE initialization parameter, [4-25](#)
- APPROX_MEDIAN function, [4-26](#)
- APPROX_PERCENTILE function, [4-26](#)
- APPROX_RANK function, [4-26](#)
- APPROX_SUM function, [4-26](#)
- approximate query processing, [4-24](#)
 - initialization parameters, [4-25](#)
 - SQL functions, [4-26](#)
- automatic error mitigation, [1-16](#)
- automatic indexing, [1-6](#)
- automatic plan capture
 - creating automatic capture filters, [29-6](#)
 - enabling, [29-5](#)
- automatic reoptimization, [4-11](#), [4-21](#), [6-15](#), [10-29](#)
 - cardinality misestimates, [4-21](#)
 - performance feedback, [4-23](#)
 - statistics feedback, [4-21](#)
- automatic statistics collection, [13-1](#)
- Automatic Tuning Optimizer, [1-5](#), [4-10](#)
 - SQL profiles, [25-6](#)

Automatic Workload Repository (AWR), [1-5](#)

B

B-tree indexes, [8-12](#)
 band joins, [9-26](#)
 big bang rollout strategy, [2-4](#)
 bind variables, [20-1](#)
 bind peeking, [20-13](#)
 cursors, [20-12](#)
 substitution, [20-20](#)
 bind-aware cursors, [20-29](#)
 bind-sensitive cursors, [20-25](#)
 bitmap indexes
 access paths, [8-30](#)
 bitmap merge, [8-39](#)
 conversion to rowid, [8-35](#)
 inlist iterator, [6-31](#)
 pruning, [4-18](#)
 purpose, [8-31](#)
 range scans, [8-37](#)
 single value, [8-36](#)
 Bloom filters, [9-43](#)
 controls, [9-45](#)
 how they work, [9-44](#)
 metadata, [9-45](#)
 purpose, [9-43](#)
 BOOLEAN data type, [8-19](#)
 bulk loads
 creating histograms, [10-16](#)
 gathering index statistics, [10-16](#)
 restrictions for online statistics gathering, [10-16](#)
 bushy join trees, [9-1](#)
 BYTES column
 PLAN_TABLE table, [7-1](#)

C

cardinality, [1-5](#), [4-5](#), [4-12](#), [4-13](#), [4-21](#), [4-23](#), [8-31](#), [10-2](#), [10-27](#), [11-1](#), [11-4](#), [27-2](#)
 CARDINALITY column
 PLAN_TABLE table, [7-1](#)
 Cartesian joins, [9-41](#), [9-42](#)
 child cursors, [20-3](#)
 V\$SQL, [20-5](#)
 column group statistics, [10-27](#)
 automatic and manual, [14-4](#)
 column groups, [14-1](#), [14-5](#)
 about, [14-1](#)
 creating, [14-8](#), [14-10](#)
 displaying information, [14-11](#)
 dropping, [14-12](#)
 optimizer statistics, [14-1](#)
 why needed, [14-2](#)

columns
 cardinality, [4-5](#)
 COMPARE_PLANS function, [6-32](#)
 compilation, SQL, [10-26](#), [10-27](#), [10-39](#)
 composite partitioning
 examples of, [6-26](#)
 concurrent statistics gathering, [13-18](#), [13-20](#), [6](#)
 CONFIGURE procedure, [28-4](#)
 consistent mode
 TKPROF, [23-26](#)
 COST column
 PLAN_TABLE table, [7-1](#)
 CPU
 system statistics, [13-36](#)
 current mode
 TKPROF, [23-26](#)
 cursor sharing
 adaptive, [20-23](#), [20-33](#)
 Real-World Performance guidelines, [20-33](#)
 CURSOR_NUM column
 TKPROF_TABLE table, [23-14](#)
 CURSOR_SHARING initialization parameter, [20-1](#), [20-20](#)
 cursor-duration temporary tables, [5-21](#), [5-22](#)
 cursors, [3-2](#)
 about, [20-1](#)
 adaptive sharing, [20-23](#), [20-33](#)
 bind variable peeking, [20-13](#)
 bind variables, [20-12](#)
 bind-aware, [20-29](#)
 bind-sensitive, [20-25](#)
 child, [20-3](#), [20-5](#)
 invalidating, [20-16](#)
 literals, [20-10](#)
 marked DEFERRED INVALIDATION, [20-18](#)
 marked rolling invalid, [20-18](#)
 merging, [20-32](#)
 parent, [20-3](#), [20-5](#)
 parsing and, [20-7](#)
 Real-World Performance guidelines, [20-33](#)
 sharing, [20-1](#)
 temporary tables and, [5-21](#), [5-22](#)

D

data blocks, [3-8](#)
 data dictionary cache, [3-3](#)
 data modeling, [2-1](#)
 Data Pump
 Export utility
 statistics on system-generated columns names, [17-3](#)
 Import utility
 copying statistics, [17-3](#)
 data skew, [11-1](#)

- data types
 - BOOLEAN, [8-19](#)
 - database operations
 - attributes, [22-6](#)
 - composite, [1-14](#)
 - definition, [1-14](#), [22-1](#)
 - monitoring, [22-1](#)
 - simple, [1-14](#), [22-1](#)
 - database operations, monitoring, [1-14](#), [22-1](#)
 - creating database operations, [22-12](#)
 - enabling with hints, [22-11](#)
 - enabling with initialization parameters, [22-10](#)
 - Enterprise Manager interface, [22-4](#)
 - PL/SQL interface, [22-5](#)
 - purpose, [22-2](#)
 - real-time SQL, [22-1](#)
 - user interfaces, [22-4](#)
 - using Cloud Control, [22-15](#)
 - Database Resource Manager, [13-20](#)
 - DATE_OF_INSERT column
 - TKPROF_TABLE table, [23-14](#)
 - DB_FILE_MULTIBLOCK_READ_COUNT
 - initialization parameter, [8-6](#)
 - DBA_ADVISOR_ACTIONS view, [26-31](#)
 - DBA_ADVISOR_EXECUTIONS view, [18-13](#)
 - DBA_IND_STATISTICS view, [13-14](#)
 - DBA_OPTSTAT_OPERATION_TASKS view, [13-23](#)
 - DBA_OPTSTAT_OPERATIONS view, [13-23](#)
 - DBA_OUTLINES view, [30-6](#)
 - DBA_SQL_MANAGEMENT_CONFIG view, [28-4](#), [29-42](#)
 - DBA_SQL_PROFILES view, [27-2](#)
 - DBA_STAT_EXTENSIONS view, [14-16](#)
 - DBA_TAB_COL_STATISTICS view, [10-20](#), [11-4](#), [14-2](#)
 - DBA_TAB_MODIFICATIONS view, [13-14](#)
 - DBA_TAB_PENDING_STATS view, [15-5](#)
 - DBA_TAB_STATISTICS view, [10-20](#), [13-14](#)
 - DBMS_ADVISOR package, [26-1](#), [26-23](#), [26-26](#), [26-27](#)
 - DBMS_AUTO_SQLTUNE package, [25-27](#)
 - DBMS_AUTO_TASK_ADMIN package, [12-2](#)
 - DBMS_MONITOR package, [1](#), [23-7](#), [23-11](#)
 - end-to-end application tracing, [23-2](#)
 - DBMS_SESSION package, [23-11](#)
 - DBMS_SPM package, [29-1](#), [29-18](#), [29-31](#), [29-43](#), [30-6](#)
 - DBMS_SQL_MONITOR package, [22-4](#)
 - DBMS_SQLSET package, [24-3](#)
 - DBMS_SQLTUNE package, [24-3](#), [25-29](#), [25-33](#), [25-40](#), [25-41](#), [27-6](#)
 - SQL Tuning Advisor, [25-32](#)
 - DBMS_STATS package, [12-2](#), [13-13](#), [13-20](#), [13-25](#), [13-38](#), [13-43](#)–[13-45](#), [14-1](#), [14-8](#), [14-11](#), [14-16](#), [15-1](#), [15-5](#), [15-6](#), [16-2](#), [16-5](#), [16-6](#), [17-3](#)
 - DBMS_XPLAN, [6-10](#)
 - DBMS_XPLAN package, [6-9](#), [6-32](#)
 - DDL (data definition language)
 - processing of, [3-9](#)
 - deadlocks, [3-2](#)
 - debugging designs, [2-3](#)
 - dedicated server, [3-3](#)
 - deferred invalidation, [20-18](#)
 - density, histogram, [11-4](#)
 - DEPTH column
 - TKPROF_TABLE table, [23-14](#)
 - designs
 - debugging, [2-3](#)
 - testing, [2-3](#)
 - validating, [2-3](#)
 - development environments, [2-3](#)
 - DIAGNOSTIC_DEST initialization parameter, [23-10](#)
 - DISTRIBUTION column
 - PLAN_TABLE table, [7-1](#), [7-9](#)
 - domain indexes
 - and EXPLAIN PLAN, [6-31](#)
 - DROP_SQL_PLAN_DIRECTIVE procedure, [12-20](#)
 - dynamic statistics, [4-20](#), [10-14](#), [10-26](#), [10-37](#), [13-17](#), [21-2](#)
 - controlling, [12-11](#)
 - process, [12-11](#)
 - sampling levels, [12-11](#)
 - setting statistics level, [12-13](#)
 - when to use, [12-15](#)
- ## E
-
- end-to-end application tracing, [1-14](#), [1](#)
 - action and module names, [23-2](#)
 - creating a service, [23-2](#)
 - DBMS_APPLICATION_INFO package, [23-2](#)
 - DBMS_MONITOR package, [23-2](#)
 - enabling, [23-6](#), [23-7](#)
 - gathering statistics, [23-4](#)
 - multitenant environment, [23-2](#)
 - overview, [23-1](#)
 - performing, [23-1](#)
 - purpose, [23-1](#)
 - endpoint repeat counts, in histograms, [11-17](#)
 - ESS
 - See Expression Statistics Store
 - examples
 - EXPLAIN PLAN output, [23-17](#)
 - EXECUTE_TASK procedure, [26-15](#)
 - execution plans, [1-13](#), [3-3](#)
 - about, [6-3](#)
 - adaptive, [4-11](#), [6-12](#), [6-15](#)

execution plans (*continued*)
 comparing, [6-32](#), [6-39](#)
 contents, [6-1](#)
 displaying, [6-1](#)
 generating, [6-1](#)
 generation, [4-8](#)
 overview of, [6-1](#)
 parallel execution, [6-20](#)
 reading, [6-9](#)
 reference, [7-1](#)
 TKPROF, [23-13](#), [23-23](#)
 V\$ views, [6-12](#)
 viewing with the utlxpls.sql script, [6-3](#)
 execution trees, [3-6](#)
 EXPLAIN PLAN statement
 access paths, [8-9](#), [8-11](#)
 and full partition-wise joins, [6-29](#)
 and partial partition-wise joins, [6-28](#)
 and partitioned objects, [6-24](#)
 basic steps, [6-4](#)
 examples of output, [23-17](#)
 invoking with the TKPROF program, [23-23](#)
 PLAN_TABLE table, [6-3](#)
 restrictions, [6-4](#)
 viewing the output, [6-3](#)
 expression statistics
 creating, [14-15](#)
 dropping, [14-17](#)
 managing, [14-13](#)
 Expression Statistics Store (ESS), [4-29](#)
 expressions, optimizer
 statistics, [14-13](#), [14-17](#)
 extended statistics, [10-6](#)
 extensions, optimizer, [10-26](#), [14-1](#)
 displaying, [14-16](#)
 external tables
 guideline for gathering optimizer statistics, [13-13](#)

F

fixed objects
 gathering statistics for, [13-16](#)
 FLUSH_DATABASE_MONITORING_INFO
 procedure, [13-14](#)
 FLUSH_SQL_PLAN_DIRECTIVE procedure, [12-20](#)
 frequency histograms, [11-5](#), [11-6](#), [11-9](#)
 full outer joins, [9-32](#)
 full partition-wise joins, [6-29](#)
 full table scans, [8-4](#), [10-8](#), [11-1](#)

G

global temporary tables, [10-3](#)

H

hard parsing, [2-1](#), [3-3](#), [3-5](#), [20-7](#)
 hash clusters, [8-42](#)
 hash joins, [9-15](#)
 cost-based optimization, [9-4](#)
 how they work, [9-16](#)
 outer joins, [9-30](#)
 hash partitions, [6-24](#)
 examples of, [6-24](#)
 height-balanced histograms, [11-13](#), [11-14](#)
 high-frequency automatic optimizer statistics
 collection
 about, [13-5](#)
 configuring, [13-5](#)
 enabling, [13-6](#)
 high-load SQL
 tuning, [12-2](#), [25-32](#)
 hints, optimizer, [1-15](#), [19-11](#)
 about, [19-12](#)
 APPEND, [10-15](#)
 conflicting, [19-20](#)
 examples, [19-20](#)
 GATHER_OPTIMIZER_STATISTICS, [10-17](#)
 guidelines, [19-16](#)
 INDEX_JOIN, [8-29](#)
 MONITOR, [22-7](#)
 multiple, [19-16](#)
 NO_GATHER_OPTIMIZER_STATISTICS, [10-17](#), [10-20](#)
 NO_MONITOR, [22-7](#), [22-11](#)
 NO_PX_JOIN_FILTER, [9-45](#)
 OPTIMIZER_FEATURES_ENABLE, [19-8](#)
 ORDERED, [9-42](#)
 PQ_DISTRIBUTE, [6-28](#)
 purpose, [19-12](#)
 PX_JOIN_FILTER, [9-45](#)
 reporting, [19-15](#), [19-18](#)
 statement-level, [19-20](#)
 types, [19-13](#)
 USE_NL, [9-14](#), [9-30](#)
 user interface for hint reports, [19-16](#)
 histograms, [11-1](#)
 bucket compression, [11-5](#)
 cardinality algorithms, [11-4](#)
 creation during bulk loads, [10-16](#)
 criteria for creation, [11-3](#)
 data skew, [11-1](#)
 definition, [11-1](#)
 density, [11-4](#)
 endpoint numbers, [11-4](#)
 endpoint repeat counts, [11-17](#)
 endpoint values, [11-4](#)
 frequency, [11-5](#), [11-6](#), [11-9](#)
 height-balanced, [11-13](#), [11-14](#)
 hybrid, [11-17](#), [11-19](#)

histograms (*continued*)

- NDV, [11-1](#)
 - nonpopular values, [11-4](#)
 - popular values, [11-4](#)
 - purpose, [11-1](#)
 - top frequency, [11-9](#)
- hybrid histograms, [11-17](#), [11-19](#)
- HyperLogLog algorithm, [13-26](#)

I

I/O

- system statistics, [13-36](#)

ID column

- PLAN_TABLE table, [7-1](#)

In-Memory Aggregation, [5-21](#)

In-Memory Expressions (IMEs)

- Expression Statistics Store (ESS), [4-29](#)

In-Memory table scans, [8-10](#)

- controls, [8-10](#)
- example, [8-11](#)
- when chosen, [8-10](#)

Incident Manager, accessing, [21-5](#)incidents, SQL, [21-1](#)incremental statistics, [13-31](#), [13-34](#)

- considerations, [13-29](#)
- maintenance, [13-25](#)
- synopses, [13-26](#)

index clustering factor, [10-8](#)index statistics, [10-7](#)

- index clustering factor, [10-8](#)

INDEX_JOIN hint, [8-29](#)

indexes

- automatic, [1-6](#)
- B-tree, [8-12](#)
- bitmap, [4-18](#), [8-30](#), [8-31](#), [8-36](#)
- domain, [6-31](#)
- fast full scans, [8-25](#)
- full scans, [8-23](#)
- join scans, [8-28](#)
- range scans, [8-19](#)
- skip scans, [8-26](#)
- unique scans, [8-16](#)

inner loop, in nested loops joins, [9-6](#)invalidation, deferred, [20-18](#)

Jjoin conditions, [9-1](#)

join methods

- about, [9-5](#)
- hash joins, [9-16](#)
- nested loops, [9-5](#), [9-7](#), [9-10](#)
- sort merge joins, [9-19](#)

join optimizations, [9-43](#)

- Bloom filters, [9-43](#)

join optimizations (*continued*)

- partition-wise, [9-47](#)

join types, [9-24](#)

- antijoins, [9-36](#)
- inner joins, [9-24](#)
- outer joins, [5-31](#), [9-29](#), [9-30](#)
- semijoins, [9-34](#)

joins, [9-1](#)

- about, [9-1](#)
- antijoins, [9-4](#), [9-37](#)
- band, [9-26](#)
- Bloom filters, [9-43](#)
- Cartesian, [9-1](#), [9-41](#), [9-42](#)
- conditions, [9-1](#)
- full outer, [9-32](#)
- hash, [9-15](#)
- how the optimizer executes, [9-3](#)
- index join scans, [8-28](#)
- inner, [9-24](#)
- join types, [9-24](#)
- nested loops, [3-6](#), [9-5](#), [9-6](#)
- nested loops and cost-based optimization, [9-4](#)
- order, [19-14](#)
- outer, [5-31](#), [9-29](#)
- partition-wise, [9-47](#)
 - examples of full, [6-29](#)
 - examples of partial, [6-28](#)
 - full, [6-29](#)
- semijoins, [9-4](#), [9-34](#)
- sort merge, [9-20](#)
- sort-merge and cost-based optimization, [9-4](#)
- trees, [9-1](#)

L

latches

- parsing and, [3-3](#)

left deep join trees, [9-1](#)

library cache

- misses, [3-3](#)

literals, in cursors, [20-10](#)

Mmanagement objects, SQL, [1](#)manual plan capture, [28-5](#)

materialized views

- query rewrite, [5-11](#)
- parameters, [5-12](#)

modeling, data, [2-1](#)multiversion read consistency, [3-8](#)

NNDV, [11-1](#)

nested loops joins, [9-5](#), [9-6](#)
 controls, [9-14](#)
 cost-based optimization, [9-4](#)
 current implementation, [9-10](#)
 how they work, [9-6](#)
 nested nested loops, [9-7](#)
 original implementation, [9-12](#)
 outer joins, [9-30](#)
 USE_NL hint, [9-14](#)
 NO_GATHER_OPTIMIZER_STATISTICS hint, [10-20](#)
 NO_PX_JOIN_FILTER hint, [9-45](#)
 nonpopular values, in histograms, [11-4](#)
 NOT IN subquery, [9-4](#)
 nulls
 in antijoins, [9-38](#)

O

OBJECT_INSTANCE column
 PLAN_TABLE table, [7-1](#)
 OBJECT_NAME column
 PLAN_TABLE table, [7-1](#)
 OBJECT_NODE column
 PLAN_TABLE table, [7-1](#)
 OBJECT_OWNER column
 PLAN_TABLE table, [7-1](#)
 OBJECT_TYPE column
 PLAN_TABLE table, [7-1](#)
 OPERATION column
 PLAN_TABLE table, [7-1](#)
 optimizer
 about, [3-5](#)
 access paths, [8-1](#)
 adaptive, [6-15](#)
 concepts, [4-1](#)
 cost-based, [4-1](#)
 definition, [4-1](#)
 environment, [3-3](#)
 estimator, [4-5](#)
 execution, [3-6](#)
 Expression Statistics Store (ESS), [4-29](#)
 extensions, [10-26](#), [14-16](#)
 goals, [19-9](#)
 plan generation, [4-8](#)
 purpose of, [4-1](#)
 query transformer, [4-4](#)
 row sources, [3-5](#), [9-5](#)
 SQL plan directives, [10-26](#)
 SQL processing, [3-1](#)
 throughput, [19-9](#)
 with rewrite, [5-11](#)
 OPTIMIZER column
 PLAN_TABLE, [7-1](#)
 optimizer hints, [1-15](#), [19-11](#)
 about, [19-12](#)
 optimizer hints (*continued*)
 APPEND, [10-15](#)
 examples, [19-20](#)
 guidelines, [19-16](#)
 INDEX_JOIN, [8-29](#)
 MONITOR, [22-11](#)
 multiple, [19-16](#)
 OPTIMIZER_FEATURES_ENABLE, [19-8](#)
 ORDERED, [9-42](#)
 PQ_DISTRIBUTE, [6-28](#)
 purpose, [19-12](#)
 reporting, [19-15](#), [19-18](#)
 types, [19-13](#)
 USE_NL, [9-30](#)
 user interface for reports, [19-16](#)
 optimizer statistics, [1](#), [10-1](#)
 adaptive statistics, [4-20](#)
 automatic collection, [13-1](#)
 bulk loads, [10-15](#)
 cardinality, [11-1](#)
 column groups, [10-27](#), [14-1](#), [14-2](#), [14-4](#), [14-8](#), [14-10](#)
 controlling, [15-1](#)
 determining staleness, [13-14](#)
 dynamic, [10-14](#), [10-26](#), [12-11](#), [13-17](#), [21-2](#)
 exporting and importing, [17-1](#)
 expressions, [14-13](#), [14-17](#)
 extended, [10-6](#), [14-1](#)
 gathering, [13-1](#)
 gathering concurrently, [13-18](#), [13-20](#), [6](#)
 gathering during bulk loads, [10-16](#)
 gathering manually, [13-10](#), [13-11](#)
 global statistics, [13-25](#)
 guideline for gathering external table, [13-13](#)
 high-frequency collection, [13-5](#), [13-6](#)
 histograms, [11-1](#), [11-14](#)
 history, [16-4](#)
 how gathered, [10-13](#)
 incremental, [13-25](#), [13-29](#), [13-31](#), [13-34](#)
 index, [10-7](#)
 locking, [15-1](#)
 managing, [16-1](#)
 manual collection, [13-10](#)
 monitoring collection, [13-23](#)
 noworkload, [13-43](#)
 Optimizer Statistics Advisor, [18-1](#), [18-2](#)
 options for gathering, [12-1](#)
 partitioned tables, [13-13](#), [13-28](#)
 pending, [15-3](#), [15-6](#)
 pluggable databases and, [13-1](#)
 preferences, [12-2](#), [12-4](#), [12-5](#)
 publishing, [15-5](#)
 purging, [16-6](#)
 reporting mode, [13-45](#)
 restoring, [16-1](#), [16-2](#)
 retention, [16-4](#)

optimizer statistics (*continued*)

- retention period, [16-5](#)
- setting, [15-9](#), [15-10](#)
- SQL plan directives, [10-26](#), [14-1](#)
- system, [13-37](#), [13-38](#), [13-43](#), [13-44](#), [25-5](#)
- table, [10-2](#)
- temporary, [10-3](#)
- transporting, [17-3](#)
- types, [10-2](#)
- unlocking, [15-2](#)
- when gathered, [10-25](#)
- workload, [13-40](#)

Optimizer Statistics Advisor, [18-1](#)

- about, [18-3](#)
- actions, [18-6](#)
- basic tasks, [18-9](#)
- benefits, [18-3](#)
- components, [18-3](#)
- creating a rule filter, [18-17](#), [18-19](#)
- creating a task, [18-12](#), [18-24](#)
- creating an object filter, [18-14](#)
- DBA_ADVISOR_EXECUTIONS view, [18-13](#)
- DBMS_STATS PL/SQL package, [18-7](#)
- executing a task, [18-23](#)
- filters, [18-13](#)
- findings, [18-5](#)
- generating a script, [18-30](#)
- implementing actions automatically, [18-28](#)
- listing advisor tasks, [18-13](#)
- modes of operation, [18-7](#)
- problems with script-based approach, [18-2](#)
- purpose, [18-2](#)
- recommendations, [18-6](#)
- rules, [18-4](#)

optimizer statistics collection, [13-5](#), [13-6](#)

optimizer statistics collectors, [4-13](#)

OPTIMIZER_ADAPTIVE_STATISTICS

- initialization parameter, [4-24](#)

OPTIMIZER_CAPTURE_SQL_PLAN_BASELINE

- S initialization parameter, [28-3](#), [29-4](#)

OPTIMIZER_FEATURES_ENABLE initialization parameter, [19-8](#)

OPTIMIZER_USE_PENDING_STATISTICS

- initialization parameter, [15-5](#)

OPTIMIZER_USE_SQL_PLAN_BASELINES

- initialization parameter, [28-3](#), [29-4](#)

OPTIONS column

- PLAN_TABLE table, [7-1](#)

OR expansion, [5-1](#)

ORA\$AUTOTASK consumer group, [13-20](#)

Oracle Exadata Database Machine, [25-5](#), [27-2](#)

ORDERED hint, [9-42](#)

OTHER column

- PLAN_TABLE table, [7-1](#)

OTHER_TAG column

- PLAN_TABLE table, [7-1](#)

outer joins, [9-29](#), [9-30](#)

- hash joins, [9-30](#)

outer loops, in nested loops join, [9-6](#)

P

parent cursors, [20-3](#), [20-5](#)

PARENT_ID column

- PLAN_TABLE table, [7-1](#)

parsing, SQL, [3-2](#)

- hard, [2-1](#), [20-7](#)
- hard parse, [3-3](#)
- parse trees, [3-6](#)
- soft, [2-1](#)
- soft parse, [3-3](#)

PARTITION_ID column

- PLAN_TABLE table, [7-1](#)

PARTITION_START column

- PLAN_TABLE table, [7-1](#)

PARTITION_STOP column

- PLAN_TABLE table, [7-1](#)

partition-wise joins, [9-47](#)

- full, [6-29](#)
- full, and EXPLAIN PLAN output, [6-29](#)
- how they work, [9-47](#)
- partial, and EXPLAIN PLAN output, [6-28](#)
- purpose, [9-47](#)

partitioned objects

- and EXPLAIN PLAN statement, [6-24](#)
- gathering incremental statistics, [13-25](#), [13-29](#)
- gathering statistics, [13-28](#)
- guideline for gathering statistics, [13-13](#)

partitioning

- examples of, [6-24](#)
- examples of composite, [6-26](#)
- hash, [6-24](#)
- maintenance operations, [13-31](#)
- range, [6-24](#)
- start and stop columns, [6-24](#)

plan evolution

- purpose, [28-8](#)

PLAN_TABLE table

- BYTES column, [7-1](#)
- CARDINALITY column, [7-1](#)
- COST column, [7-1](#)
- creating, [6-3](#)
- displaying, [6-9](#)
- DISTRIBUTION column, [7-1](#)
- ID column, [7-1](#)
- OBJECT_INSTANCE column, [7-1](#)
- OBJECT_NAME column, [7-1](#)
- OBJECT_NODE column, [7-1](#)
- OBJECT_OWNER column, [7-1](#)
- OBJECT_TYPE column, [7-1](#)
- OPERATION column, [7-1](#)
- OPTIMIZER column, [7-1](#)

PLAN_TABLE table (*continued*)

- OPTIONS column, [7-1](#)
- OTHER column, [7-1](#)
- OTHER_TAG column, [7-1](#)
- PARENT_ID column, [7-1](#)
- PARTITION_ID column, [7-1](#)
- PARTITION_START column, [7-1](#)
- PARTITION_STOP column, [7-1](#)
- POSITION column, [7-1](#)
- REMARKS column, [7-1](#)
- SEARCH_COLUMNS column, [7-1](#)
- STATEMENT_ID column, [7-1](#)
- TIMESTAMP column, [7-1](#)

pluggable databases

- automatic optimizer statistics collection, [13-1](#)
- manageability features, [13-1](#)
- SQL management base, [28-11](#)
- SQL Tuning Advisor, [25-1](#)
- SQL tuning sets, [24-1](#)

popular values, in histograms, [11-4](#)

POSITION column

- PLAN_TABLE table, [7-1](#)

PQ_DISTRIBUTE hint, [6-28](#)

preferences, optimizer statistics, [12-2](#), [12-5](#)

private SQL areas, [20-1](#)

- parsing and, [3-2](#)

processes

- dedicated server, [3-3](#)

profiles, SQL

- about, [27-1](#)
- concepts, [27-2](#)
- implementing, [27-7](#)
- managing, [27-1](#)

programming languages, [2-3](#)

PX_JOIN_FILTER hint, [9-45](#)

Q

quarantined SQL, [4-28](#)

query rewrite

- correctness, [5-12](#)
- parameters, [5-12](#)

query transformations, [5-1](#)

- In-Memory Aggregation, [5-21](#)
- join factorization, [5-29](#)
- OR expansion, [5-1](#)
- predicate pushing, [5-9](#)
- purpose of table expansions, [5-23](#)
- query rewrite with materialized views, [5-11](#)
- star transformation, [5-16](#)
- star transformations, [5-14](#), [8-31](#)
- subquery unnesting, [5-10](#)
- table expansion, [5-23](#), [5-24](#)
- view merging, [5-3](#)

query transformer, [4-4](#)

R

range

- examples of partitions, [6-24](#)
- partitions, [6-24](#)

Real-Time Database Operations, [1-14](#)

Real-Time SQL Monitoring, [1-14](#), [22-1](#)

- architecture, [22-3](#)

Real-World Performance

- cursor sharing, [20-1](#)

recursive SQL, [3-9](#), [10-14](#), [10-39](#), [23-16](#)

REMARKS column

- PLAN_TABLE table, [7-1](#)

reoptimization, automatic, [4-21](#), [6-15](#), [10-29](#)

- cardinality misestimates, [4-21](#)
- performance feedback, [4-23](#)
- statistics feedback, [4-21](#)

restoring optimizer statistics, [16-1](#), [16-2](#)

result sets, SQL, [3-5](#), [3-8](#)

rewrites

- parameters, [5-12](#)

right deep join trees, [9-1](#)

rolling invalid cursors, [20-18](#)

rollout strategies

- big bang approach, [2-4](#)
- trickle approach, [2-4](#)

rowids

- table access by, [8-7](#)

rows

- row sets, [3-5](#)
- row sources, [3-5](#), [1](#), [9-5](#)
- rowids used to locate, [8-7](#)

S

SAMPLE BLOCK clause, [8-9](#)

SAMPLE clause, [8-9](#)

sample table scans, [8-9](#)

scans

- In-Memory, [8-10](#)
- sample table, [8-9](#)

SEARCH_COLUMNS column

- PLAN_TABLE table, [7-1](#)

SELECT statement

- SAMPLE clause, [8-9](#)

semijoins, [9-4](#), [9-34](#)

- how they work, [9-34](#)
- when the optimizer consider, [9-34](#)

SET_EVOLVE_TASK_PARAMETER procedure, [29-13](#)

shared cursors

- life cycle, [20-16](#)

shared pool, [10-27](#), [20-1](#)

- parsing check, [3-3](#)

shared SQL areas, [3-3](#), [20-1](#)

SMB

- See SQL management base
- soft parsing, [2-1](#), [3-3](#)
- sort merge joins, [9-19](#)
 - cost-based optimization, [9-4](#)
 - how they work, [9-20](#)
- SPM Evolve Advisor, [29-8](#), [29-9](#)
- SPM Evolve Advisor task
 - about, [29-13](#)
 - managing, [29-9](#)
 - scheduling, [29-9](#)
- SQL
 - execution, [3-6](#)
 - management objects, [1](#)
 - operators, [1](#)
 - optimization, [4-1](#)
 - parsing, [3-2](#)
 - processing, [3-1](#)
 - quarantined, [4-28](#)
 - recursive, [3-9](#), [10-39](#), [12-11](#)
 - result sets, [3-5](#), [3-8](#)
 - stages of processing, [8-2](#), [8-12](#)
 - test cases, [21-1](#)
- SQL Access Advisor, [1-6](#), [26-1](#)
 - about, [26-1](#)
 - actions, [26-5](#)
 - advanced tasks, [26-22](#)
 - basic tasks, [26-8](#)
 - constants, [26-33](#)
 - creating task templates, [26-24](#)
 - deleting tasks, [26-28](#)
 - demo script, [26-31](#)
 - evaluation mode, [26-22](#)
 - EXECUTE_TASK procedure, [26-15](#)
 - filter options, [26-3](#)
 - input sources, [26-3](#)
 - marking recommendations, [26-29](#)
 - modifying recommendations, [26-30](#)
 - populating SQL tuning sets, [26-11](#)
 - quick tune, [26-21](#)
 - recommendations, [26-6](#)
 - reference, [26-31](#)
 - repository, [26-6](#)
 - task categories, [26-33](#)
 - terminating tasks, [26-26](#)
 - updating task attributes, [26-23](#)
 - user interfaces, [26-7](#)
- SQL compilation, [10-26](#), [10-27](#), [10-39](#)
- SQL incidents, [21-1](#)
- SQL management base, [29-43](#)
 - changing disk space limits, [29-43](#)
 - managing, [29-42](#)
 - plan retention policy, [29-44](#)
 - pluggable databases and, [28-11](#)
- SQL Performance Analyzer, [1-10](#)
- SQL plan baselines, [1-9](#), [4-28](#), [28-1](#)
 - automatic plan capture, [28-4](#)
- SQL plan baselines (*continued*)
 - configuring initial capture, [29-4](#)
 - displaying, [29-17](#)
 - dropping, [29-39](#), [29-40](#)
 - loading, [29-18](#)
 - loading from AWR, [29-19](#)
 - loading from shared SQL area, [29-22](#)
 - loading using DBMS_SPM, [29-18](#)
 - managing, [29-1](#)
 - plan history, [28-13](#)
 - plan matching, [28-5](#)
 - plan retention, [29-44](#)
- SQL plan capture, [28-3](#)
 - configuring filters of automatic plan capture, [29-6](#)
 - enabling automatic, [29-5](#)
- SQL plan directives, [4-23](#), [10-26](#), [14-1](#)
 - cardinality misestimates, [10-27](#)
 - creation, [10-27](#)
 - managing, [12-20](#)
- SQL plan evolution
 - managing an evolve task, [29-31](#)
- SQL plan history, [28-13](#)
- SQL plan management, [1-9](#)
 - accepted plans, [4-28](#)
 - automatic plan capture, [28-3](#), [29-5](#)
 - basic tasks, [29-3](#)
 - benefits, [28-1](#)
 - configuring, [29-8](#)
 - DBMS_SPM, [29-2](#)
 - DBMS_SPM package, [29-1](#)
 - filters for automatic plan capture, [29-6](#)
 - introduction, [28-1](#)
 - loading plans from a staging table, [29-26](#)
 - loading plans from SQL tuning sets, [29-24](#)
 - manual plan capture, [28-5](#)
 - plan capture, [28-1](#)
 - plan evolution, [28-1](#), [28-8](#), [29-31](#)
 - plan retention, [29-44](#)
 - plan selection, [28-1](#), [28-7](#)
 - plan verification, [28-8](#)
 - purpose, [28-1](#)
 - SPM Evolve Advisor, [29-9](#)
 - SQL management base, [29-42](#), [29-43](#)
 - SQL plan baselines, [4-28](#), [28-1](#), [29-1](#), [29-39](#), [29-40](#)
 - SQL plan capture, [28-3](#)
 - storage architecture, [28-11](#)
 - user interfaces, [29-1](#)
- SQL processing, [3-1](#)
 - semantic check, [3-2](#)
 - shared pool check, [3-3](#)
 - stages, [3-1](#)
 - syntax check, [3-2](#)
- SQL profiles, [1-5](#), [25-6](#)
 - about, [27-1](#)

- SQL profiles (*continued*)
 - and SQL plan baselines, [28-2](#)
 - concepts, [27-2](#)
 - Exadata-aware, [27-2](#)
 - implementing, [27-7](#)
 - managing, [27-1](#)
 - statistics, [27-2](#)
 - user interface, [27-6](#)
- SQL statement log, [28-12](#)
- SQL statements
 - execution plans of, [6-1](#)
- SQL Test Case Builder, [21-1](#)
 - accessing the Incident Manager, [21-5](#)
 - accessing the Support Workbench, [21-6](#)
 - gathering diagnostic data, [21-1](#)
 - graphical interface, [21-5](#)
 - key concepts, [21-1](#)
 - output, [21-3](#)
 - running, [21-7](#)
 - SQL incidents, [21-1](#)
 - user interfaces, [21-5](#)
 - what it captures, [21-2](#)
- SQL trace facility, [1](#), [23-3](#), [23-10](#)
 - enabling, [23-11](#)
 - generating output, [23-13](#)
 - output, [23-26](#)
 - statement truncation, [23-13](#)
 - trace files, [1-14](#), [23-10](#)
- SQL tuning
 - automation, [1-4](#)
 - definition, [1-1](#)
 - introduction, [1-1](#)
 - manual tools, [1-13](#)
 - proactive, [1-2](#)
 - purpose, [1-1](#)
 - reactive, [1-2](#)
 - tools overview, [1-2](#), [1-4](#)
- SQL Tuning Advisor, [1-5](#), [25-1](#)
 - about, [25-1](#)
 - administering with APIs, [25-32](#)
 - analyses, [25-5](#)
 - architecture, [25-2](#)
 - automatic, [25-23](#), [25-29](#)
 - automatic tuning task, [25-20](#), [25-27](#)
 - command-line interface, [25-33](#)
 - configuring a SQL tuning task, [25-37](#)
 - controls, [25-14](#)
 - creating a task, [25-35](#)
 - DBMS_SQLTUNE, [25-33](#)
 - displaying tuning results, [25-41](#)
 - executing a task, [25-39](#)
 - input sources, [25-3](#)
 - monitoring tasks, [25-40](#)
 - pluggable databases and, [25-1](#)
 - reports, [25-28](#)
 - running on-demand, [25-31](#), [25-32](#)
- SQL Tuning Advisor (*continued*)
 - statistical analysis, [25-5](#)
 - using, [12-2](#), [25-32](#)
 - using remotely, [25-15](#)
- SQL tuning sets
 - command-line interface, [24-3](#)
 - creating, [24-6](#)
 - loading, [24-7](#)
 - loading in SQL plan baselines, [29-24](#)
 - managing with APIs, [24-1](#)
 - pluggable databases and, [24-1](#)
 - populating with user-defined workload, [26-11](#)
- SQL_STATEMENT column
 - TKPROF_TABLE, [23-14](#)
- star transformations, [5-14](#), [5-16](#), [8-31](#)
- start columns
 - in partitioning and EXPLAIN PLAN statement, [6-24](#)
- STATEMENT_ID column
 - PLAN_TABLE table, [7-1](#)
- STATISTICS_LEVEL initialization parameter, [16-6](#)
- statistics, optimizer, [4-1](#), [1](#)
 - adaptive statistics, [4-20](#), [13-26](#)
 - automatic collection, [13-1](#)
 - bulk loads, [10-15](#)
 - cardinality, [11-1](#)
 - column group, [10-27](#)
 - column groups, [14-1](#), [14-2](#), [14-5](#), [14-8](#), [14-10](#)
 - controlling, [15-1](#)
 - deleting, [13-44](#)
 - determining staleness, [13-14](#)
 - dynamic, [10-14](#), [10-26](#), [12-11](#), [13-17](#), [21-2](#)
 - dynamic statistics, [10-37](#)
 - exporting and importing, [17-1](#)
 - expressions, [14-13](#)
 - extended, [10-6](#), [14-1](#)
 - gathering concurrently, [13-18](#)
 - gathering manually, [13-10](#), [13-11](#)
 - global statistics, [13-25](#)
 - guideline for external tables, [13-13](#)
 - history, [16-4](#)
 - how gathered, [10-13](#)
 - incremental, [13-25](#), [13-29](#), [13-31](#), [13-34](#)
 - index, [10-7](#)
 - locking, [15-1](#)
 - managing, [16-1](#)
 - manual collection, [13-10](#)
 - monitoring collection, [13-23](#)
 - noworkload, [13-43](#)
 - Optimizer Statistics Advisor, [18-1](#), [18-2](#)
 - options for gathering, [12-1](#)
 - partitioned tables, [13-28](#)
 - pending, [15-3](#), [15-6](#)
 - preferences, [12-2](#)
 - publishing, [15-5](#)
 - purging, [16-6](#)

statistics, optimizer (*continued*)
 reporting mode, [13-45](#)
 restoring, [16-1](#), [16-2](#)
 retention, [16-4](#)
 retention period, [16-5](#)
 setting, [15-9](#)
 system, [10-12](#), [13-36–13-38](#)
 table, [10-2](#)
 transporting, [17-3](#)
 types, [10-2](#)
 unlocking, [15-2](#)
 user-defined, [10-13](#)
 when gathered, [10-25](#)
 workload, [13-40](#)

stop columns
 in partitioning and EXPLAIN PLAN statement, [6-24](#)

stored outlines
 about, [30-1](#)
 categories, [30-3](#)
 how they work, [30-2](#)
 purpose of migrating, [30-1](#)

subqueries
 NOT IN, [9-4](#)

Support Workbench, accessing, [21-6](#)

synopses, [13-26](#)

SYS_AUTO_SQL_TUNING_TASK, [25-39](#)

system statistics, [13-37](#), [13-38](#), [25-5](#)
 about, [13-36](#)
 deleting, [13-44](#)
 gathering, [13-41](#)
 noworkload, [13-43](#)

T

table clusters
 access paths, [8-40](#)

table expansion
 about, [5-23](#)
 how it works, [5-23](#)
 purpose, [5-23](#)
 scenario, [5-24](#)

table statistics, [10-2](#)

task templates, SQL Access Advisor, [26-24](#)

temporary tables
 cursor-duration, [5-21](#), [5-22](#)
 global, [10-3](#)
 types, [10-3](#)

testing designs, [2-3](#)

throughput
 optimizer goal, [19-9](#)

TIMED_STATISTICS initialization parameter, [23-26](#)

TIMESTAMP column
 PLAN_TABLE table, [7-1](#)

TIMESTAMP column (*continued*)

TKPROF program, [23-4](#), [23-10](#), [23-20](#), [23-22](#)
 editing the output SQL script, [23-14](#)
 example of output, [23-17](#)
 generating the output SQL script, [23-14](#)
 row source operations, [23-27](#)
 timing statistics, [23-16](#)
 using the EXPLAIN PLAN statement, [23-23](#)
 wait event information, [23-27](#)

TKPROF_TABLE, [23-14](#)

top frequency histograms, [11-9](#)

tracing, [1](#), [23-1](#)
 consolidating with TRCSESS, [23-20](#)
 enabling, [23-6](#), [23-7](#)
 enabling statistics gathering, [23-4](#)
 generating output files, [23-10](#)
 identifying files, [23-10](#)

transformations, query
 In-Memory Aggregation, [5-21](#)
 join factorization, [5-29](#)
 predicate pushing, [5-9](#)
 purpose of table expansions, [5-23](#)
 star, [5-14](#)
 subquery unnesting, [5-10](#)
 table expansion, [5-23](#), [5-24](#)
 view merging, [5-3](#)

transformer, query, [4-4](#)

TRCSESS program, [23-20](#)

trees, join, [9-1](#)

trickle rollout strategy, [2-4](#)

TRUNCATE command, [16-2](#)

U

unique index scans, [8-16](#)

USE_NL hint, [9-14](#), [9-30](#)

USER_ID column, TKPROF_TABLE, [23-14](#)

utlxlpl.sql script, [6-9](#)

utlxlpls.sql script, [6-9](#)

V

V\$SESSION view, [23-4](#)

V\$SQL view, [20-5](#), [20-33](#)

V\$SQL_CS_HISTOGRAM view, [20-33](#)

V\$SQL_CS_SELECTIVITY view, [20-33](#)

V\$SQL_CS_STATISTICS view, [20-33](#)

V\$SQLAREA view, [20-5](#)

validating designs, [2-3](#)

view merging, [5-3](#)

W

workloads, [2-3](#)