Glossary

accepted plan

In the context of SQL plan management, a plan that is in a SQL plan baseline for a SQL statement and thus available for use by the optimizer. An accepted plan contains a set of hints, a plan hash value, and other plan-related information.

access path

The means by which the database retrieves data from a database. For example, a query using an index and a query using a full table scan use different access paths.

adaptive cursor sharing

A feature that enables a single statement that contains bind variables to use multiple execution plans. Cursor sharing is "adaptive" because the **cursor** adapts its behavior so that the database does not always use the same plan for each execution or bind variable value.

adaptive dynamic sampling

A feature of the adaptive optimizer that enables the automatic adjustment of the dynamic statistics level.

adaptive optimizer

A feature of the optimizer that enables it to adapt plans based on run-time statistics.

adaptive query plan

An execution plan that changes after optimization because run-time conditions indicate that optimizer estimates are inaccurate. An adaptive query plan has different built-in plan options. During the first execution, before a specific subplan becomes active, the optimizer makes a final decision about which option to use. The optimizer bases its choice on observations made during the execution up to this point. Thus, an adaptive query plan enables the final plan for a statement to differ from the default plan.

adaptive query optimization

A set of capabilities that enables the **adaptive optimizer** to make run-time adjustments to execution plans and discover additional information that can lead to better statistics. Adaptive optimization is helpful when existing statistics are not sufficient to generate an optimal plan.

ADDM

See **Automatic Database Diagnostic Monitor (ADDM)**.

antijoin

A join that returns rows that fail to match the subquery on the right side. For example, an antijoin can list departments with no employees. Antijoins use the `NOT EXISTS` or `NOT IN` constructs.

approximate query processing

A set of optimization techniques that speed analytic queries by calculating results within an acceptable range of error.

automatic capture filter

A SQL plan management feature that enables you to specify the eligibility criteria for automatic initial plan capture.

Automatic Database Diagnostic Monitor (ADDM)

ADDM is self-diagnostic software built into Oracle Database. ADDM examines and analyzes data captured in **Automatic Workload Repository (AWR)** to determine possible database performance problems.

automatic optimizer statistics collection

The automatic running of the `DBMS_STATS` package to collect optimizer statistics for all schema objects for which statistics are missing or stale.

automatic initial plan capture

The mechanism by which the database automatically creates a SQL plan baseline for any repeatable SQL statement executed on the database. Enable automatic initial plan capture by setting the `OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES` initialization parameter to `true` (the default is `false`).

See **repeatable SQL statement**.

automatic reoptimization

The ability of the optimizer to automatically change a plan on subsequent executions of a SQL statement. Automatic reoptimization can fix any suboptimal plan chosen due to incorrect optimizer estimates, from a suboptimal distribution method to an incorrect choice of degree of parallelism.

automatic SQL tuning

The work performed by Automatic SQL Tuning Advisor it runs as an automated task within system maintenance windows.

Automatic SQL Tuning Advisor

SQL Tuning Advisor when run as an automated maintenance task. Automatic SQL Tuning runs during system maintenance windows as an automated maintenance task, searching for ways to improve the execution plans of high-load SQL statements.

See **SQL Tuning Advisor**.

Automatic Tuning Optimizer

The optimizer when invoked by SQL Tuning Advisor. In SQL tuning mode, the optimizer performs additional analysis to check whether it can further improve the plan produced in normal mode. The optimizer output is not an execution plan, but a series of actions, along with their rationale and expected benefit for producing a significantly better plan.

Automatic Workload Repository (AWR)

The infrastructure that provides services to Oracle Database components to collect, maintain, and use statistics for problem detection and self-tuning.

AWR

See **Automatic Workload Repository (AWR)**.

AWR snapshot

A set of data for a specific time that is used for performance comparisons. The delta values captured by the snapshot represent the changes for each statistic over the time period. Statistics gathered by are queried from memory. You can display the gathered data in both reports and views.

band join

A special type of nonequijoin in which key values in one data set must fall within the specified range (“band”) of the second data set.

base cardinality

For a table, the total number of rows in the table.

baseline

In the context of **AWR**, the interval between two AWR snapshots that represent the database operating at an optimal level.

bind-aware cursor

A bind-sensitive cursor that is eligible to use different plans for different bind values. After a cursor has been made bind-aware, the optimizer chooses plans for future executions based on the bind value and its **cardinality** estimate.

bind-sensitive cursor

A cursor whose optimal plan may depend on the value of a bind variable. The database monitors the behavior of a bind-sensitive cursor that uses different bind values to determine whether a different plan is beneficial.

bind variable

A placeholder in a SQL statement that must be replaced with a valid value or value address for the statement to execute successfully. By using bind variables, you can write a SQL statement that accepts inputs or parameters at run time. The following query uses `v_empid` as a bind variable:

```
SELECT * FROM employees WHERE employee_id = :v_empid;
```

bind variable peeking

The ability of the optimizer to look at the value in a bind variable during a **hard parse**. By peeking at bind values, the optimizer can determine the selectivity of a `WHERE` clause condition as if literals *had* been used, thereby improving the plan.

bitmap join index

A bitmap index for the join of two or more tables.

bitmap piece

A subcomponent of a single bitmap index entry. Each indexed column value may have one or more bitmap pieces. The database uses bitmap pieces to break up an index entry that is large in relation to the size of a block.

B-tree index

An index organized like an upside-down tree. A B-tree index has two types of blocks: branch blocks for searching and leaf blocks that store values. The leaf blocks contain every indexed data value and a corresponding rowid used to locate the actual row. The "B" stands for "balanced" because all leaf blocks automatically stay at the same depth.

bulk load

A `CREATE TABLE AS SELECT` or `INSERT INTO ... SELECT` operation.

bushy join tree

A join tree in which the left or the right child of an internal node can be a join node. Nodes in a bushy join tree may have recursive structures in both its descendents.

cardinality

The number of rows that is expected to be or is returned by an operation in an execution plan.

Cartesian join

A join in which one or more of the tables does not have any join conditions to any other tables in the statement. The **optimizer** joins every row from one data source with every row from the other data source, creating the Cartesian product of the two sets.

child cursor

The cursor containing the plan, compilation environment, and other information for a statement whose text is stored in a **parent cursor**. The parent cursor is number `0`, the first child is number `1`, and so on. Child cursors reference the same SQL text as the parent cursor, but are different. For example, two queries with the text `SELECT * FROM t` use different cursors when they reference two different tables named `t`.

cluster scan

An access path for a table cluster. In an indexed table cluster, Oracle Database first obtains the rowid of one of the selected rows by scanning the cluster index. Oracle Database then locates the rows based on this rowid.

column group

A set of columns that is treated as a unit.

column group statistics

Extended statistics gathered on a group of columns treated as a unit.

column statistics

Statistics about columns that the **optimizer** uses to determine optimal execution plans. Column statistics include the number of distinct column values, low value, high value, and number of nulls.

complex view merging

The merging of views containing the `GROUP BY` or `DISTINCT` keywords.

composite database operation

In a **database operation**, the activity between two points in time in a database session, with each session defining its own beginning and end points. A session can participate in at most one composite database operation at a time.

concurrency

Simultaneous access of the same data by many users. A multiuser database management system must provide adequate concurrency controls so that data cannot be updated or changed improperly, compromising data integrity.

concurrent statistics gathering mode

A mode that enables the database to simultaneously gather optimizer statistics for multiple tables in a schema, or multiple partitions or subpartitions in a table. Concurrency can reduce the overall time required to gather statistics by enabling the database to fully use multiple CPUs.

condition

A combination of one or more expressions and logical operators that returns a value of `TRUE`, `FALSE`, or `UNKNOWN`.

cost

A numeric internal measure that represents the estimated resource usage for an **execution plan**. The lower the cost, the more efficient the plan.

cost-based optimizer (CBO)

The legacy name for the optimizer. In earlier releases, the cost-based optimizer was an alternative to the rule-based optimizer (RBO).

cost model

The internal optimizer model that accounts for the **cost** of the I/O, CPU, and network resources that a query is predicted to use.

cumulative statistics

A count such as the number of block reads. Oracle Database generates many types of cumulative statistics for the system, sessions, and individual SQL statements.

cursor

A handle or name for a **private SQL area** in the PGA. Because cursors are closely associated with private SQL areas, the terms are sometimes used interchangeably.

cursor cache

See **shared SQL area**.

cursor merging

Combining cursors to save space in the shared SQL area. If the optimizer creates a plan for a **bind-aware cursor**, and if this plan is the same as an existing cursor, then the optimizer can merge the cursors.

cursor-duration temporary table

A temporary, in-memory table that stores query results for the duration of a cursor. For complex operations such as `WITH` clause queries and star transformations, this optimization enhances the materialization of intermediate results from repetitively used subqueries. In this way, cursor-duration temporary tables improve performance and optimizes I/O.

data flow operator (DFO)

The unit of work between data redistribution stages in a parallel query.

data skew

Large variations in the number of duplicate values in a column.

database operation

A set of database tasks defined by end users or application code, for example, a batch job or ETL processing.

default plan

For an adaptive plan, the execution plan initially chosen by the optimizer using the statistics from the data dictionary. The default plan can differ from the **final plan**.

disabled plan

A plan that a database administrator has manually marked as ineligible for use by the optimizer.

degree of parallelism (DOP)

The number of parallel execution servers associated with a single operation. Parallel execution is designed to effectively use multiple CPUs. Oracle Database parallel execution framework enables you to either explicitly choose a specific degree of parallelism or to rely on Oracle Database to automatically control it.

dense key

A numeric key that is stored as a native integer and has a range of values.

dense grouping key

A key that represents all grouping keys whose grouping columns come from a specific fact table or dimension.

dense join key

A key that represents all join keys whose join columns come from a particular fact table or dimension.

density

A decimal number between `0` and `1` that measures the selectivity of a column. Values close to `1` indicate that the column is unselective, whereas values close to `0` indicate that this column is more selective.

direct path read

A single or multiblock read into the PGA, bypassing the SGA.

driving table

The table to which other tables are joined. An analogy from programming is a for loop that contains another for loop. The outer for loop is the analog of a driving table, which is also called an **outer table**.

dynamic performance view

A view created on dynamic performance tables, which are virtual tables that record current database activity. The dynamic performance views are called fixed views because they cannot be altered or removed by the database administrator. They are also called `V$` views because they begin with the string `V$` (`GV$` in Oracle RAC).

dynamic plan

A set of subplan groups. A subplan group is set of subplans. In an adaptive query plan, the optimizer chooses a subplan at run time depending on the statistics obtained by the statistics collector.

dynamic statistics

An optimization technique in which the database executes a **recursive SQL** statement to scan a small random sample of a table's blocks to estimate predicate selectivities.

dynamic statistics level

The level that controls both when the database gathers dynamic statistics, and the size of the sample that the optimizer uses to gather the statistics. Set the dynamic statistics level using either the `OPTIMIZER_DYNAMIC_SAMPLING` initialization parameter or a statement **hint**.

enabled plan

In **SQL plan management**, a plan that is eligible for use by the optimizer.

endpoint number

A number that uniquely identifies a bucket in a **histogram**. In frequency and hybrid histograms, the endpoint number is the cumulative frequency of endpoints. In height-balanced histograms, the endpoint number is the bucket number.

endpoint repeat count

In a hybrid histogram, the number of times the **endpoint value** is repeated, for each endpoint (bucket) in the histogram. By using the repeat count, the optimizer can obtain accurate estimates for almost popular values.

endpoint value

An endpoint value is the highest value in the range of values in a **histogram** bucket.

equijoin

A join whose join condition contains an equality operator.

estimator

The component of the optimizer that determines the overall cost of a given **execution plan**.

execution plan

The combination of steps used by the database to execute a SQL statement. Each step either retrieves rows of data physically from the database or prepares them for the session issuing the statement. You can override execution plans by using a hint.

execution tree

A tree diagram that shows the flow of row sources from one step to another in an **execution plan**.

expected cardinality

For a table, the estimated number of rows the table has after all filter predicates have been applied to the table.

expression

A combination of one or more values, operators, and SQL functions that evaluates to a value. For example, the expression `2*2` evaluates to `4`. In general, expressions assume the data type of their components.

expression statistics

A type of extended statistics that improves optimizer estimates when a `WHERE` clause has predicates that use expressions.

extended statistics

A type of optimizer statistics that improves estimates for cardinality when multiple predicates exist or when predicates contain an expression.

extensible optimizer

An optimizer capability that enables authors of user-defined functions and indexes to create statistics collection, selectivity, and cost functions that the optimizer uses when choosing an execution plan. The optimizer cost model is extended to integrate information supplied by the user to assess CPU and I/O cost.

extension

A column group or an expression. The statistics collected for column groups and expressions are called **extended statistics**.

external table

A read-only table whose metadata is stored in the database but whose data in stored in files outside the database. The database uses the metadata describing external tables to expose their data as if they were relational tables.

filter condition

A `WHERE` clause component that eliminates rows from a single object referenced in a SQL statement.

final plan

In an adaptive plan, the plan that executes to completion. The **default plan** can differ from the final plan.

fixed object

A dynamic performance table or its index. The fixed objects are owned by `SYS`. Fixed object tables have names beginning with `X$` and are the base tables for the `V$` views.

fixed plan

An **accepted plan** that is marked as preferred, so that the optimizer considers only the fixed plans in the **SQL plan baseline**. You can use fixed plans to influence the plan selection process of the optimizer.

frequency histogram

A type of histogram in which each distinct column value corresponds to a single bucket. An analogy is sorting coins: all pennies go in bucket 1, all nickels go in bucket 2, and so on.

full outer join

A combination of a left and right outer join. In addition to the inner join, the database uses nulls to preserve rows from both tables that have not been returned in the result of the inner join. In other words, full outer joins join tables together, yet show rows with no corresponding rows in the joined tables.

full table scan

A scan of table data in which the database sequentially reads all rows from a table and filters out those that do not meet the selection criteria. All data blocks under the high water mark are scanned.

global temporary table

A special temporary table that stores intermediate session-private data for a specific duration.

hard parse

The steps performed by the database to build a new executable version of application code. The database must perform a hard parse instead of a soft parse if the parsed representation of a submitted statement does not exist in the **shared SQL area**.

hash cluster

A type of table cluster that is similar to an indexed cluster, except the index key is replaced with a hash function. No separate cluster index exists. In a hash cluster, the data is the index.

hash collision

Hashing multiple input values to the same output value.

hash function

A function that operates on an arbitrary-length input value and returns a fixed-length hash value.

hash join

A method for joining large data sets. The database uses the smaller of two data sets to build a hash table on the join key in memory. It then scans the larger data set, probing the hash table to find the joined rows.

hash scan

An access path for a table cluster. The database uses a hash scan to locate rows in a hash cluster based on a hash value. In a hash cluster, all rows with the same hash value are stored in the same data block. To perform a hash scan, Oracle Database first obtains the hash value by applying a hash function to a cluster key value specified by the statement, and then scans the data blocks containing rows with that hash value.

hash table

An in-memory data structure that associates join keys with rows in a hash join. For example, in a join of the `employees` and `departments` tables, the join key might be the department ID. A hash function uses the join key to generate a hash value. This hash value is an index in an array, which is the hash table.

hash value

In a hash cluster, a unique numeric ID that identifies a bucket. Oracle Database uses a hash function that accepts an infinite number of hash key values as input and sorts them into a finite number of buckets. Each hash value maps to the database block address for the block that stores the rows corresponding to the hash key value (department 10, 20, 30, and so on).

hashing

A mathematical technique in which an infinite set of input values is mapped to a finite set of output values, called hash values. Hashing is useful for rapid lookups of data in a hash table.

heap-organized table

A table in which the data rows are stored in no particular order on disk. By default, `CREATE TABLE` creates a heap-organized table.

height-balanced histogram

A histogram in which column values are divided into buckets so that each bucket contains approximately the same number of rows.

hint

An instruction passed to the **optimizer** through comments in a SQL statement. The optimizer uses hints to choose an execution plan for the statement.

histogram

A special type of column statistic that provides more detailed information about the data distribution in a table column.

hybrid hash distribution technique

An adaptive parallel data distribution that does not decide the final data distribution method until execution time.

hybrid histogram

An enhanced height-based histogram that stores the exact frequency of each endpoint in the sample, and ensures that a value is never stored in multiple buckets.

hybrid partitioned table

A table in which some partitions are stored in data file segments and some are stored in external data source.

implicit query

A component of a DML statement that retrieves data without a **subquery**. An `UPDATE`, `DELETE`, or `MERGE` statement that does not explicitly include a `SELECT` statement uses an implicit query to retrieve the rows to be modified.

In-Memory scan

A table scan that retrieves rows from the In-Memory Column Store (IM column store).

incremental statistics maintenance

The ability of the database to generate global statistics for a partitioned table by aggregating partition-level statistics.

index

Optional schema object associated with a nonclustered table, table partition, or table cluster. In some cases indexes speed data access.

index cluster

An table cluster that uses an index to locate data. The cluster index is a B-tree index on the cluster key.

index clustering factor

A measure of row order in relation to an indexed value such as employee last name. The more scattered the rows among the data blocks, the lower the clustering factor.

index fast full scan

A scan of the index blocks in unsorted order, as they exist on disk. This scan reads the index instead of the table.

index full scan

The scan of an entire index in key order.

index-organized table

A table whose storage organization is a variant of a primary B-tree index. Unlike a heap-organized table, data is stored in primary key order.

index range scan

An index range scan is an ordered scan of an index that has the following characteristics:

- One or more leading columns of an index are specified in conditions.
- 0, 1, or more values are possible for an index key.

index range scan descending

An index range scan in which the database returns rows in descending order.

index skip scan

An index scan occurs in which the initial column of a composite index is "skipped" or not specified in the query. For example, if the composite index key is `(cust_gender,cust_email)`, then the query predicate does not reference the `cust_gender` column.

index statistics

Statistics about indexes that the **optimizer** uses to determine whether to perform a full table scan or an index scan. Index statistics include B-tree levels, leaf block counts, the index clustering factor, distinct keys, and number of rows in the index.

index unique scan

A scan of an index that returns either 0 or 1 rowid.

indextype

An object that specifies the routines that manage a domain (application-specific) index.

inner join

A join of two or more tables that returns only those rows that satisfy the join condition.

inner table

In a nested loops join, the table that is not the **outer table** (driving table). For every row in the outer table, the database accesses all rows in the inner table. The outer loop is for every row in the outer table and the inner loop is for every row in the inner table.

join

A statement that retrieves data from multiple tables specified in the `FROM` clause of a SQL statement. Join types include inner joins, outer joins, and Cartesian joins.

join condition

A condition that compares two row sources using an expression. The database combines pairs of rows, each containing one row from each row source, for which the join condition evaluates to `true`.

join elimination

The removal of redundant tables from a query. A table is redundant when its columns are only referenced in join predicates, and those joins are guaranteed to neither filter nor expand the resulting rows.

join factorization

A cost-based transformation that can factorize common computations from branches of a `UNION ALL` query. Without join factorization, the optimizer evaluates each branch of a `UNION ALL` query independently, which leads to repetitive processing, including data access and joins. Avoiding an extra scan of a large base table can lead to a huge performance improvement.

join group

A user-created database object that specifies a group of columns that participate in an join. Join groups are only supported in the In-Memory column store.

join method

A method of joining a pair of row sources. The possible join methods are nested loop, sort merge, and hash joins. A Cartesian join requires one of the preceding join methods

join order

The order in which multiple tables are joined together. For example, for each row in the `employees` table, the database can read each row in the `departments` table. In an alternative join order, for each row in the `departments` table, the database reads each row in the `employees` table.

To execute a statement that joins more than two tables, Oracle Database joins two of the tables and then joins the resulting row source to the next table. This process continues until all tables are joined into the result.

join predicate

A predicate in a `WHERE` or `JOIN` clause that combines the columns of two tables in a join.

key vector

A data structure that maps between dense join keys and dense grouping keys.

latch

A low-level serialization control mechanism used to protect shared data structures in the SGA from simultaneous access.

left deep join tree

A join tree in which the left input of every join is the result of a previous join.

left table

In an outer join, the table specified on the left side of the `OUTER JOIN` keywords (in ANSI SQL syntax).

library cache

An area of memory in the shared pool. This cache includes the shared SQL areas, private SQL areas (in a shared server configuration), PL/SQL procedures and packages, and control structures such as locks and library cache handles.

library cache hit

The reuse of SQL statement code found in the library cache.

library cache miss

During SQL processing, the act of searching for a usable plan in the library cache and not finding it.

maintenance window

A contiguous time interval during which automated maintenance tasks run. The maintenance windows are Oracle Scheduler windows that belong to the window group named `MAINTENANCE_WINDOW_GROUP`.

manual plan capture

The user-initiated bulk load of existing plans into a **SQL plan baseline**.

materialized view

A schema object that stores a query result. All materialized views are either read-only or updatable.

multiblock read

An I/O call that reads multiple database blocks. Multiblock reads can significantly speed up full table scans. For example, a data block might be 8 KB, but the operating system can read 1024 KB in a single I/O. For some queries, the optimizer may decide that it is more cost-efficient to read 128 data blocks in one I/O than in 128 sequential I/Os.

NDV

Number of distinct values. The NDV is important in generating cardinality estimates.

nested loops join

A type of join method. A nested loops join determines the outer table that drives the join, and for every row in the outer table, probes each row in the inner table. The outer loop is for each row in the outer table and the inner loop is for each row in the inner table. An analogy from programming is a `for` loop inside of another `for` loop.

nonequijoin

A join whose join condition does not contain an equality operator.

nonjoin column

A predicate in a `WHERE` clause that references only one table.

nonpopular value

In a histogram, any value that does *not* span two or more endpoints. Any value that is not nonpopular is a **popular value**.

noworkload statistics

Optimizer system statistics gathered when the database simulates a workload.

on-demand SQL tuning

The manual invocation of SQL Tuning Advisor. Any invocation of SQL Tuning Advisor that is not the result of an Automatic SQL Tuning task is on-demand tuning.

optimization

The overall process of choosing the most efficient means of executing a SQL statement.

optimizer

Built-in database software that determines the most efficient way to execute a SQL statement by considering factors related to the objects referenced and the conditions specified in the statement.

optimizer cost model

The model that the optimizer uses to select an execution plan. The optimizer selects the execution plan with the lowest cost, where cost represents the estimated resource usage for that plan. The optimizer cost model accounts for the I/O, CPU, and network resources that the query will use.

optimizer environment

The totality of session settings that can affect execution plan generation, such as the work area size or optimizer settings (for example, the optimizer mode).

optimizer goal

The prioritization of resource usage by the optimizer. Using the `OPTIMIZER_MODE` initialization parameter, you can set the optimizer goal best throughput or best response time.

optimizer statistics

Details about the database its object used by the optimizer to select the best **execution plan** for each SQL statement. Categories include **table statistics** such as numbers of rows, **index statistics** such as B-tree levels, **system statistics** such as CPU and I/O performance, and **column statistics** such as number of nulls.

Optimizer Statistics Advisor

A tool that inspects statistics gathering practices, automatically diagnoses problems with these practices, and generates a report of findings and recommendations.

Optimizer Statistics Advisor rules

System-supplied standards by which Optimizer Statistics Advisor performs its checks.

optimizer statistics collection

The gathering of optimizer statistics for database objects. The database can collect these statistics automatically, or you can collect them manually by using the system-supplied `DBMS_STATS` package.

optimizer statistics collector

A row source inserted into an execution plan at key points to collect run-time statistics for use in adaptive plans.

optimizer statistics preferences

The default values of the parameters used by automatic statistics collection and the `DBMS_STATS` statistics gathering procedures.

outer join

A join condition using the outer join operator (+) with one or more columns of one of the tables. The database returns all rows that meet the join condition. The database also returns all rows from the table without the outer join operator for which there are no matching rows in the table with the outer join operator.

outer table

See **driving table**

parallel execution

The application of multiple CPU and I/O resources to the execution of a single database operation.

parallel query

A query in which multiple processes work together simultaneously to run a single SQL query. By dividing the work among multiple processes, Oracle Database can run the statement more quickly. For example, four processes retrieve rows for four different quarters in a year instead of one process handling all four quarters by itself.

parent cursor

The cursor that stores the SQL text and other minimal information for a SQL statement. The **child cursor** contains the plan, compilation environment, and other information. When a statement first executes, the database creates both a parent and child cursor in the shared pool.

parse call

A call to Oracle to prepare a SQL statement for execution. The call includes syntactically checking the SQL statement, optimizing it, and then building or locating an executable form of that statement.

parsing

The stage of **SQL processing** that involves separating the pieces of a SQL statement into a data structure that can be processed by other routines.

A hard parse occurs when the SQL statement to be executed is either not in the shared pool, or it is in the shared pool but it cannot be shared. A soft parse occurs when a session attempts to execute a SQL statement, and the statement is already in the shared pool, and it can be used.

partition maintenance operation

A partition-related operation such as adding, exchanging, merging, or splitting table partitions.

partition-wise join

A join optimization that divides a large join of two tables, one of which must be partitioned on the join key, into several smaller joins.

pending statistics

Unpublished optimizer statistics. By default, the optimizer uses published statistics but does not use pending statistics.

performance feedback

This form of **automatic reoptimization** helps improve the degree of parallelism automatically chosen for repeated SQL statements when `PARALLEL_DEGREE_POLICY` is set to `ADAPTIVE`.

pipelined table function

A PL/SQL function that accepts a collection of rows as input. You invoke the table function as the operand of the table operator in the `FROM` list of a `SELECT` statement.

plan evolution

The manual change of an **unaccepted plan** in the **SQL plan history** into an **accepted plan** in the **SQL plan baseline**.

plan generator

The part of the optimizer that tries different access paths, join methods, and join orders for a given query block to find the plan with the lowest cost.

plan selection

The attempt to find a matching plan in the **SQL plan baseline** for a statement after performing a hard parse.

plan verification

Comparing the performance of an **unaccepted plan** to a plan in a **SQL plan baseline** and ensuring that it performs better.

popular value

In a histogram, any value that spans two or more endpoints. Any value that is not popular is an **nonpopular value**.

predicate pushing

A transformation technique in which the optimizer "pushes" the relevant predicates from the containing query block into the view query block. For views that are not merged, this technique improves the subplan of the unmerged view because the database can use the pushed-in predicates to access indexes or to use as filters.

private SQL area

An area in memory that holds a parsed statement and other information for processing. The private SQL area contains data such as **bind variable** values, query execution state information, and query execution work areas.

private temporary table

A memory-only temporary table whose data and metadata is session-private.

proactive SQL tuning

Using SQL tuning tools to identify SQL statements that are candidates for tuning *before* users have complained about a performance problem.

See **reactive SQL tuning**, **SQL tuning**.

projection view

An optimizer-generated view that appear in queries in which a `DISTINCT` view has been merged, or a `GROUP BY` view is merged into an outer query block that also contains `GROUP BY`, `HAVING`, or aggregates.

See **simple view merging**, **complex view merging**.

query

An operation that retrieves data from tables or views. For example, `SELECT * FROM employees` is a query.

query block

A top-level `SELECT` statement, **subquery**, or unmerged view

query optimizer

See **optimizer**.

reactive SQL tuning

Diagnosing and fixing SQL-related performance problems *after* users have complained about them.

See **proactive SQL tuning**, **SQL tuning**.

real-time statistics

Supplemental statistics collected automatically during conventional DML operations.

recursive SQL

Additional SQL statements that the database must issue to execute a SQL statement issued by a user. The generation of recursive SQL is known as a recursive call. For example, the database generates recursive calls when data dictionary information is not available in memory and so must be retrieved from disk.

reoptimization

See **automatic reoptimization**.

repeatable SQL statement

A statement that the database parses or executes after recognizing that it is tracked in the **SQL statement log**.

response time

The time required to complete a unit of work.

See **throughput**.

result set

In a query, the set of rows generated by the execution of a cursor.

right deep join tree

A join tree in which the right input of every join is the result of a previous join, and the left child of every internal node of a join tree is a table.

right table

In an outer join, the table specified on the right side of the `OUTER JOIN` keywords (in ANSI SQL syntax).

rowid

A globally unique address for a row in a table.

row set

A set of rows returned by a step in an **execution plan**.

row source

An iterative control structure that processes a set of rows in an iterated manner and produces a row set.

row source generator

Software that receives the optimal plan from the optimizer and outputs the execution plan for the SQL statement.

row source tree

A collection of row sources produced by the row source generator. The row source tree for a SQL statement shows information such as table order, access methods, join methods, and data operations such as filters and sorts.

rule filter

The use of `DBMS_STATS.CONFIGURE_ADVISOR_RULE_FILTER` to restrict an Optimizer Statistics Advisor task to a user-specified set of rules. For example, you might exclude the rule that checks for stale statistics.

sample table scan

A scan that retrieves a random sample of data from a simple table or a complex `SELECT` statement, such as a statement involving joins and views.

sampling

Gathering statistics from a random subset of rows in a table.

selectivity

A value indicating the proportion of a row set retrieved by a predicate or combination of predicates, for example, `WHERE last_name = 'Smith'`. A selectivity of `0` means that no rows pass the predicate test, whereas a value of `1` means that all rows pass the test.

The adjective *selective* means roughly "choosy." Thus, a highly selective query returns a low proportion of rows (selectivity close to `0`), whereas an **unselective** query returns a high proportion of rows (selectivity close to `1`).

semijoin

A join that returns rows from the first table when at least one match exists in the second table. For example, you list departments with at least one employee. The difference between a semijoin and a conventional join is that rows in the first table are returned at most once. Semijoins use the `EXISTS` or `IN` constructs.

shared cursor

A shared SQL area that is used by multiple SQL statements.

shared pool

Portion of the SGA that contains shared memory constructs such as shared SQL areas.

shared SQL area

An area in the shared pool that contains the parse tree and **execution plan** for a SQL statement. Only one shared SQL area exists for a unique statement. The shared SQL area is sometimes referred to as the **cursor cache**.

simple database operation

A database operation consisting of a single SQL statement or PL/SQL procedure or function.

simple view merging

The merging of select-project-join views. For example, a query joins the `employees` table to a subquery that joins the `departments` and `locations` tables.

SMB

See SQL management base (SMB).

snowflake schema

A star schema in which dimension tables reference other tables.

snowstorm schema

A combination of multiple snowflake schemas.

soft parse

Any parse that is not a hard parse. If a submitted SQL statement is the same as a reusable SQL statement in the shared pool, then Oracle Database reuses the existing code. This reuse of code is also called a **library cache hit**.

sort merge join

A type of join method. The join consists of a sort join, in which both inputs are sorted on the join key, followed by a merge join, in which the sorted lists are merged.

SQL Access Advisor

SQL Access Advisor is internal diagnostic software that recommends which materialized views, indexes, and materialized view logs to create, drop, or retain.

SQL compilation

In the context of Oracle SQL processing, this term refers collectively to the phases of parsing, optimization, and plan generation.

SQL handle

A string value derived from the numeric SQL signature. Like the signature, the handle uniquely identifies a SQL statement. It serves as a SQL search key in user APIs.

SQL ID

For a specific SQL statement, the unique identifier of the parent cursor in the library cache. A hash function applied to the text of the SQL statement generates the SQL ID. The `V$SQL.SQL_ID` column displays the SQL ID.

SQL incident

In the fault diagnosability infrastructure of Oracle Database, a single occurrence of a SQL-related problem. When a problem (critical error) occurs multiple times, the database creates an incident for each occurrence. Incidents are timestamped and tracked in the Automatic Diagnostic Repository (ADR).

SQL management base (SMB)

A logical repository that stores statement logs, plan histories, SQL plan baselines, and SQL profiles. The SMB is part of the data dictionary and resides in the `SYSAUX` tablespace.

SQL management object

A feature that stabilizes the execution plans of individual SQL statements. Examples include SQL profiles, SQL plan baselines, and SQL patches.

SQL plan baseline

A set of one or more accepted plans for a repeatable SQL statement. Each accepted plan contains a set of hints, a plan hash value, and other plan-related information. SQL plan management uses SQL plan baselines to record and evaluate the execution plans of SQL statements over time.

SQL plan capture

Techniques for capturing and storing relevant information about plans in the SQL management base (SMB) for a set of SQL statements. Capturing a plan means making SQL plan management aware of this plan.

SQL plan directive

Additional information and instructions that the optimizer can use to generate a more optimal plan. For example, a SQL plan directive might instruct the optimizer to collect missing statistics or gather dynamic statistics.

SQL plan history

The set of captured execution plans. The history contains both SQL plan baselines and unaccepted plans.

SQL plan management

SQL plan management is a preventative mechanism that records and evaluates the execution plans of SQL statements over time. SQL plan management can prevent SQL plan regressions caused by environmental changes such as a new optimizer version, changes to optimizer statistics, system settings, and so on.

SQL processing

The stages of parsing, optimization, row source generation, and execution of a SQL statement.

SQL profile

A set of auxiliary information built during automatic tuning of a SQL statement. A SQL profile is to a SQL statement what statistics are to a table. The optimizer can use SQL profiles to improve cardinality and selectivity estimates, which in turn leads the optimizer to select better plans.

SQL profiling

The verification and validation by the Automatic Tuning Advisor of its own estimates.

SQL signature

A numeric hash value computed using a SQL statement text that has been normalized for case insensitivity and white space. It uniquely identifies a SQL statement. The database uses this signature as a key to maintain SQL management objects such as SQL profiles, SQL plan baselines, and SQL patches.

SQL statement log

When automatic SQL plan capture is enabled, a log that contains the SQL ID of SQL statements that the optimizer has evaluated over time. A statement is tracked when it exists in the log.

SQL test case

A problematic SQL statement and related information needed to reproduce the execution plan in a different environment. A SQL test case is stored in an Oracle Data Pump file.

SQL test case builder

A database feature that gathers information related to a SQL statement and packages it so that a user can reproduce the problem on a different database. The `DBMS_SQLDIAG` package is the interface for SQL test case builder.

SQL trace file

A server-generated file that provides performance information on individual SQL statements. For example, the trace file contains parse, execute, and fetch counts, CPU and elapsed times, physical reads and logical reads, and misses in the library cache.

SQL tuning

The process of improving SQL statement efficiency to meet measurable goals.

SQL Tuning Advisor

Built-in database diagnostic software that optimizes high-load SQL statements.

See Automatic SQL Tuning Advisor.

SQL tuning set (STS)

A database object that includes one or more SQL statements along with their execution statistics and execution context.

star schema

A relational schema whose design represents a dimensional data model. The star schema consists of one or more fact tables and one or more dimension tables that are related through foreign keys.

statistics feedback

A form of automatic reoptimization that automatically improves plans for repeated queries that have cardinality misestimates. The optimizer may estimate cardinalities incorrectly for many reasons, such as missing statistics, inaccurate statistics, or complex predicates.

stored outline

A set of hints for a SQL statement. The hints in stored outlines direct the optimizer to choose a specific plan for the statement.

subplan

A portion of an adaptive plan that the optimizer can switch to as an alternative at run time. A subplan can consist of multiple operations in the plan. For example, the optimizer can treat a join method and the corresponding access path as one unit when determining whether to change the plan at run time.

subplan group

A set of subplans in an adaptive query plan.

subquery

A **query** nested within another SQL statement. Unlike implicit queries, subqueries use a `SELECT` statement to retrieve data.

subquery unnesting

A transformation technique in which the optimizer transforms a nested query into an equivalent join statement, and then optimizes the join.

synopsis

A set of auxiliary statistics gathered on a partitioned table when the `INCREMENTAL` value is set to `true`.

system statistics

Statistics that enable the **optimizer** to use CPU and I/O characteristics. Index statistics include B-tree levels, leaf block counts, clustering factor, distinct keys, and number of rows in the index.

table cluster

A schema object that contains data from one or more tables, all of which have one or more columns in common. In table clusters, the database stores together all the rows from all tables that share the same cluster key.

table expansion

A transformation technique that enables the optimizer to generate a plan that uses indexes on the read-mostly portion of a partitioned table, but not on the active portion of the table.

table statistics

Statistics about tables that the **optimizer** uses to determine table access cost, join cardinality, join order, and so on. Table statistics include row counts, block counts, empty blocks, average free space per block, number of chained rows, average row length, and staleness of the statistics on the table.

throughput

The amount of work completed in a unit of time.

See **response time**.

top frequency histogram

A variation of a **frequency histogram** that ignores nonpopular values that are statistically insignificant, thus producing a better histogram for highly popular values.

tuning mode

One of the two optimizer modes. When running in tuning mode, the optimizer is known as the **Automatic Tuning Optimizer**. In tuning mode, the optimizer determines whether it can further improve the plan produced in normal mode. The optimizer output is not an execution plan, but a series of actions, along with their rationale and expected benefit for producing a significantly better plan.

unaccepted plan

A plan for a statement that is in the **SQL plan history** but has not been added to the **SQL plan management**.

unselective

A relatively large fraction of rows from a row set. A query becomes more unselective as the **selectivity** approaches `1`. For example, a query that returns 999,999 rows from a table with one million rows is unselective. A query of the same table that returns one row is selective.

user response time

The time between when a user submits a command and receives a response.

See **throughput**.

V$ view

See **dynamic performance view**.

vector I/O

A type of I/O in which the database obtains a set of rowids, sends them batched in an array to the operating system, which performs the read.

view merging

The merging of a query block representing a view into the query block that contains it. View merging can improve plans by enabling the optimizer to consider additional join orders, access methods, and other transformations.

workload statistics

Optimizer statistics for system activity in a specified time period.