PLAN_TABLE Columns

The PLAN_TABLE used by the EXPLAIN PLAN statement contains the columns listed in Table 19–1.

Table 19–1 PLAN_TABLE Columns

Column	Type	Description
STATEMENT_ID	VARCHAR2(30)	Value of the optional STATEMENT_ID parameter specified in the EXPLAIN PLAN statement.
PLAN_ID	NUMBER	Unique identifier of a plan in the database.
TIMESTAMP	DATE	Date and time when the EXPLAIN PLAN statement was generated.
REMARKS	VARCHAR2(80)	Any comment (of up to 80 bytes) you want to associate with each step of the explained plan. This column is used to indicate whether an outline or SQL Profile was used for the query. If you need to add or change a remark on any row of the PLAN_TABLE, then use the UPDATE statement to modify the rows of the PLAN_TABLE.
OPERATION	VARCHAR2(30)	Name of the internal operation performed in this step. In the first row generated for a statement, the column contains one of the following values: ▪ DELETE STATEMENT ▪ INSERT STATEMENT ▪ SELECT STATEMENT ▪ UPDATE STATEMENT See Table 19–3 for more information on values for this column.
OPTIONS	VARCHAR2(225)	A variation on the operation described in the OPERATION column. See Table 19–3 for more information on values for this column.
OBJECT_NODE	VARCHAR2(128)	Name of the database link used to reference the object (a table name or view name). For local queries using parallel execution, this column describes the order in which output from operations is consumed.
OBJECT_OWNER	VARCHAR2(30)	Name of the user who owns the schema containing the table or index.
OBJECT_NAME	VARCHAR2(30)	Name of the table or index.
OBJECT_ALIAS	VARCHAR2(65)	Unique alias of a table or view in a SQL statement. For indexes, it is the object alias of the underlying table.

Table 19–1 (Cont.) PLAN_TABLE Columns

Column	Type	Description
OBJECT_INSTANCE	NUMERIC	Number corresponding to the ordinal position of the object as it appears in the original statement. The numbering proceeds from left to right, outer to inner with respect to the original statement text. View expansion results in unpredictable numbers.
OBJECT_TYPE	VARCHAR2(30)	Modifier that provides descriptive information about the object; for example, NON-UNIQUE for indexes.
OPTIMIZER	VARCHAR2(255)	Current mode of the optimizer.
SEARCH_COLUMNS	NUMBERIC	Not currently used.
ID	NUMERIC	A number assigned to each step in the execution plan.
PARENT_ID	NUMERIC	The ID of the next execution step that operates on the output of the ID step.
DEPTH	NUMERIC	Depth of the operation in the row source tree that the plan represents. The value can be used for indenting the rows in a plan table report.
POSITION	NUMERIC	For the first row of output, this indicates the optimizer's estimated cost of executing the statement. For the other rows, it indicates the position relative to the other children of the same parent.
COST	NUMERIC	Cost of the operation as estimated by the optimizer's query approach. Cost is not determined for table access operations. The value of this column does not have any particular unit of measurement; it is merely a weighted value used to compare costs of execution plans. The value of this column is a function of the CPU_COST and IO_COST columns.
CARDINALITY	NUMERIC	Estimate by the query optimization approach of the number of rows accessed by the operation.
BYTES	NUMERIC	Estimate by the query optimization approach of the number of bytes accessed by the operation.
OTHER_TAG	VARCHAR2(255)	Describes the contents of the OTHER column. Values are: ■ SERIAL (blank) - Serial execution. Currently, SQL is not loaded in the OTHER column for this case. ■ SERIAL_FROM_REMOTE (S -> R) - Serial execution at a remote site. ■ PARALLEL_FROM_SERIAL (S -> P) - Serial execution. Output of step is partitioned or broadcast to parallel execution servers. ■ PARALLEL_TO_SERIAL (P -> S) - Parallel execution. Output of step is returned to serial query coordinator (QC) process. ■ PARALLEL_TO_PARALLEL (P -> P) - Parallel execution. Output of step is repartitioned to second set of parallel execution servers. ■ PARALLEL_COMBINED_WITH_PARENT (PWP) - Parallel execution; Output of step goes to next step in same parallel process. No interprocess communication to parent. ■ PARALLEL_COMBINED_WITH_CHILD (PWC) - Parallel execution. Input of step comes from prior step in same parallel process. No interprocess communication from child.

Table 19–1 (Cont.) PLAN_TABLE Columns

Column	Type	Description
PARTITION_START	VARCHAR2(255)	Start partition of a range of accessed partitions. It can take one of the following values: *n* indicates that the start partition has been identified by the SQL compiler, and its partition number is given by *n*. KEY indicates that the start partition will be identified at run time from partitioning key values. ROW REMOVE_LOCATION indicates that the start partition (same as the stop partition) will be computed at run time from the location of each record being retrieved. The record location is obtained by a user or from a global index. INVALID indicates that the range of accessed partitions is empty.
PARTITION_STOP	VARCHAR2(255)	Stop partition of a range of accessed partitions. It can take one of the following values: *n* indicates that the stop partition has been identified by the SQL compiler, and its partition number is given by *n*. KEY indicates that the stop partition will be identified at run time from partitioning key values. ROW REMOVE_LOCATION indicates that the stop partition (same as the start partition) will be computed at run time from the location of each record being retrieved. The record location is obtained by a user or from a global index. INVALID indicates that the range of accessed partitions is empty.
PARTITION_ID	NUMERIC	Step that has computed the pair of values of the PARTITION_START and PARTITION_STOP columns.
OTHER	LONG	Other information that is specific to the execution step that a user might find useful. See the OTHER_TAG column.
DISTRIBUTION	VARCHAR2(30)	Method used to distribute rows from producer query servers to consumer query servers. See Table 19–2 for more information on the possible values for this column. For more information about consumer and producer query servers, see *Oracle Database Data Warehousing Guide*.
CPU_COST	NUMERIC	CPU cost of the operation as estimated by the query optimizer's approach. The value of this column is proportional to the number of machine cycles required for the operation. For statements that use the rule-based approach, this column is null.
IO_COST	NUMERIC	I/O cost of the operation as estimated by the query optimizer's approach. The value of this column is proportional to the number of data blocks read by the operation. For statements that use the rule-based approach, this column is null.
TEMP_SPACE	NUMERIC	Temporary space, in bytes, used by the operation as estimated by the query optimizer's approach. For statements that use the rule-based approach, or for operations that do not use any temporary space, this column is null.
ACCESS_PREDICATES	VARCHAR2(4000)	Predicates used to locate rows in an access structure. For example, start or stop predicates for an index range scan.
FILTER_PREDICATES	VARCHAR2(4000)	Predicates used to filter rows before producing them.

Table 19–1 (Cont.) PLAN_TABLE Columns

Column	Type	Description
PROJECTION	VARCHAR2(4000)	Expressions produced by the operation.
TIME	NUMBER(20,2)	Elapsed time in seconds of the operation as estimated by query optimization. For statements that use the rule-based approach, this column is null.
QBLOCK_NAME	VARCHAR2(30)	Name of the query block, either system-generated or defined by the user with the QB_NAME hint.