

**NAME**

ANALYZE – collect statistics about a database

**SYNOPSIS**

```
ANALYZE [ ( option [, ...] ) ] [ table_and_columns [, ...] ]
```

where *option* can be one of:

```
VERBOSE [ boolean ]
SKIP_LOCKED [ boolean ]
BUFFER_USAGE_LIMIT size
```

and *table\_and\_columns* is:

```
[ ONLY ] table_name [ * ] [ ( column_name [, ...] ) ]
```

**DESCRIPTION**

**ANALYZE** collects statistics about the contents of tables in the database, and stores the results in the pg\_statistic system catalog. Subsequently, the query planner uses these statistics to help determine the most efficient execution plans for queries.

Without a *table\_and\_columns* list, **ANALYZE** processes every table and materialized view in the current database that the current user has permission to analyze. With a list, **ANALYZE** processes only those table(s). It is further possible to give a list of column names for a table, in which case only the statistics for those columns are collected.

**PARAMETERS****VERBOSE**

Enables display of progress messages at INFO level.

**SKIP\_LOCKED**

Specifies that **ANALYZE** should not wait for any conflicting locks to be released when beginning work on a relation: if a relation cannot be locked immediately without waiting, the relation is skipped. Note that even with this option, **ANALYZE** may still block when opening the relation's indexes or when acquiring sample rows from partitions, table inheritance children, and some types of foreign tables. Also, while **ANALYZE** ordinarily processes all partitions of specified partitioned tables, this option will cause **ANALYZE** to skip all partitions if there is a conflicting lock on the partitioned table.

**BUFFER\_USAGE\_LIMIT**

Specifies the Buffer Access Strategy ring buffer size for **ANALYZE**. This size is used to calculate the number of shared buffers which will be reused as part of this strategy. 0 disables use of a Buffer Access Strategy. When this option is not specified, **ANALYZE** uses the value from vacuum\_buffer\_usage\_limit. Higher settings can allow **ANALYZE** to run more quickly, but having too large a setting may cause too many other useful pages to be evicted from shared buffers. The minimum value is 128 kB and the maximum value is 16 GB.

*boolean*

Specifies whether the selected option should be turned on or off. You can write TRUE, ON, or 1 to enable the option, and FALSE, OFF, or 0 to disable it. The *boolean* value can also be omitted, in which case TRUE is assumed.

*size*

Specifies an amount of memory in kilobytes. Sizes may also be specified as a string containing the numerical size followed by any one of the following memory units: B (bytes), kB (kilobytes), MB (megabytes), GB (gigabytes), or TB (terabytes).

*table\_name*

The name (possibly schema-qualified) of a specific table to analyze. If omitted, all regular tables, partitioned tables, and materialized views in the current database are analyzed (but not foreign tables). If ONLY is specified before the table name, only that table is analyzed. If ONLY is not specified, the

table and all its inheritance child tables or partitions (if any) are analyzed. Optionally, `*` can be specified after the table name to explicitly indicate that inheritance child tables (or partitions) are to be analyzed.

*column\_name*

The name of a specific column to analyze. Defaults to all columns.

## OUTPUTS

When **VERBOSE** is specified, **ANALYZE** emits progress messages to indicate which table is currently being processed. Various statistics about the tables are printed as well.

## NOTES

To analyze a table, one must ordinarily have the **MAINTAIN** privilege on the table. However, database owners are allowed to analyze all tables in their databases, except shared catalogs. **ANALYZE** will skip over any tables that the calling user does not have permission to analyze.

Foreign tables are analyzed only when explicitly selected. Not all foreign data wrappers support **ANALYZE**. If the table's wrapper does not support **ANALYZE**, the command prints a warning and does nothing.

In the default PostgreSQL configuration, the autovacuum daemon (see Section 24.1.6) takes care of automatic analyzing of tables when they are first loaded with data, and as they change throughout regular operation. When autovacuum is disabled, it is a good idea to run **ANALYZE** periodically, or just after making major changes in the contents of a table. Accurate statistics will help the planner to choose the most appropriate query plan, and thereby improve the speed of query processing. A common strategy for read-mostly databases is to run **VACUUM** and **ANALYZE** once a day during a low-usage time of day. (This will not be sufficient if there is heavy update activity.)

While **ANALYZE** is running, the `search_path` is temporarily changed to `pg_catalog, pg_temp`.

**ANALYZE** requires only a read lock on the target table, so it can run in parallel with other non-DDL activity on the table.

The statistics collected by **ANALYZE** usually include a list of some of the most common values in each column and a histogram showing the approximate data distribution in each column. One or both of these can be omitted if **ANALYZE** deems them uninteresting (for example, in a unique-key column, there are no common values) or if the column data type does not support the appropriate operators. There is more information about the statistics in Chapter 24.

For large tables, **ANALYZE** takes a random sample of the table contents, rather than examining every row. This allows even very large tables to be analyzed in a small amount of time. Note, however, that the statistics are only approximate, and will change slightly each time **ANALYZE** is run, even if the actual table contents did not change. This might result in small changes in the planner's estimated costs shown by **EXPLAIN**. In rare situations, this non-determinism will cause the planner's choices of query plans to change after **ANALYZE** is run. To avoid this, raise the amount of statistics collected by **ANALYZE**, as described below.

The extent of analysis can be controlled by adjusting the `default_statistics_target` configuration variable, or on a column-by-column basis by setting the per-column statistics target with **ALTER TABLE ... ALTER COLUMN ... SET STATISTICS**. The target value sets the maximum number of entries in the most-common-value list and the maximum number of bins in the histogram. The default target value is 100, but this can be adjusted up or down to trade off accuracy of planner estimates against the time taken for **ANALYZE** and the amount of space occupied in `pg_statistic`. In particular, setting the statistics target to zero disables collection of statistics for that column. It might be useful to do that for columns that are never used as part of the **WHERE**, **GROUP BY**, or **ORDER BY** clauses of queries, since the planner will have no use for statistics on such columns.

The largest statistics target among the columns being analyzed determines the number of table rows sampled to prepare the statistics. Increasing the target causes a proportional increase in the time and space needed to do **ANALYZE**.

One of the values estimated by **ANALYZE** is the number of distinct values that appear in each column.

Because only a subset of the rows are examined, this estimate can sometimes be quite inaccurate, even with the largest possible statistics target. If this inaccuracy leads to bad query plans, a more accurate value can be determined manually and then installed with **ALTER TABLE ... ALTER COLUMN ... SET (n\_distinct = ...)**.

If the table being analyzed has inheritance children, **ANALYZE** gathers two sets of statistics: one on the rows of the parent table only, and a second including rows of both the parent table and all of its children. This second set of statistics is needed when planning queries that process the inheritance tree as a whole. The autovacuum daemon, however, will only consider inserts or updates on the parent table itself when deciding whether to trigger an automatic analyze for that table. If that table is rarely inserted into or updated, the inheritance statistics will not be up to date unless you run **ANALYZE** manually. By default, **ANALYZE** will also recursively collect and update the statistics for each inheritance child table. The **ONLY** keyword may be used to disable this.

For partitioned tables, **ANALYZE** gathers statistics by sampling rows from all partitions. By default, **ANALYZE** will also recursively collect and update the statistics for each partition. The **ONLY** keyword may be used to disable this.

The autovacuum daemon does not process partitioned tables, nor does it process inheritance parents if only the children are ever modified. It is usually necessary to periodically run a manual **ANALYZE** to keep the statistics of the table hierarchy up to date.

If any child tables or partitions are foreign tables whose foreign data wrappers do not support **ANALYZE**, those tables are ignored while gathering inheritance statistics.

If the table being analyzed is completely empty, **ANALYZE** will not record new statistics for that table. Any existing statistics will be retained.

Each backend running **ANALYZE** will report its progress in the `pg_stat_progress_analyze` view. See Section 27.4.1 for details.

## COMPATIBILITY

There is no **ANALYZE** statement in the SQL standard.

The following syntax was used before PostgreSQL version 11 and is still supported:

```
ANALYZE [ VERBOSE ] [ table_and_columns [, ...] ]
```

## SEE ALSO

**VACUUM**(7), **vacuumdb**(1), Section 19.10.2, Section 24.1.6, Section 27.4.1