comment

- Description
- Parameters
- Notes
- Examples
- Compatibility

COMMENT

COMMENT

7

SQL - Language Statements

COMMENT

define or change the comment of an object

COMMENT ON { ACCESS METHOD

object_name

| AGGREGATE

aggregate_name

(

aggregate_signature

) | CAST (

source_type

AS

target_type

) | COLLATION

object_name

| COLUMN

relation_name

.

column_name

| CONSTRAINT

constraint_name

ON

table_name

| CONSTRAINT

constraint_name

ON DOMAIN

domain_name

| CONVERSION

object_name

| DATABASE

object_name

| DOMAIN

object_name

| EXTENSION

object_name

| EVENT TRIGGER

object_name

| FOREIGN DATA WRAPPER

object_name

| FOREIGN TABLE

object_name

| FUNCTION

function_name

[([[

argmode

] [

argname

]

argtype

[, ...]])] | INDEX

object_name

| LARGE OBJECT

large_object_oid

| MATERIALIZED VIEW

object_name

| OPERATOR

operator_name

(

left_type

,

right_type

) | OPERATOR CLASS

object_name

USING

index_method

| OPERATOR FAMILY

object_name

USING

index_method

| POLICY

policy_name

ON

table_name

| [PROCEDURAL] LANGUAGE

object_name

| PROCEDURE

procedure_name

[([[

argmode

] [

argname

]

argtype

[, ...]])] | PUBLICATION

object_name

| ROLE

object_name

| ROUTINE

routine_name

[([[

argmode

] [

argname

]

argtype

[, ...]])] | RULE

rule_name

ON

table_name

| SCHEMA

object_name

| SEQUENCE

object_name

| SERVER

object_name

| STATISTICS

object_name

| SUBSCRIPTION

object_name

| TABLE

object_name

| TABLESPACE

object_name

| TEXT SEARCH CONFIGURATION

object_name

| TEXT SEARCH DICTIONARY

object_name

| TEXT SEARCH PARSER

object_name

| TEXT SEARCH TEMPLATE

object_name

| TRANSFORM FOR

type_name

LANGUAGE

lang_name

| TRIGGER

trigger_name

ON

table_name

| TYPE

object_name

| VIEW

object_name

} IS {

string_literal

| NULL }

where

aggregate_signature

is:

* | [

argmode

] [

argname

]

argtype

[, ...] | [[

argmode

] [

argname

]

argtype

[, ...]] ORDER BY [

argmode

] [

argname

]

argtype

[, ...]

Description

`COMMENT` stores, replaces, or removes the comment on a database object.

Only one comment string is stored for each object. Issuing a new `COMMENT` command for the same object replaces the existing comment. Specifying `NULL` or an empty string (`''`) removes the comment. Comments are automatically dropped when their object is dropped.

A `SHARE UPDATE EXCLUSIVE` lock is acquired on the object to be commented.

For most kinds of object, only the object's owner can set the comment. Roles don't have owners, so the rule for `COMMENT ON ROLE` is that you must be superuser to comment on a superuser role, or have the `CREATEROLE` privilege and have been granted `ADMIN OPTION` on the target role. Likewise, access methods don't have owners either; you must be superuser to comment on an access method. Of course, a superuser can comment on anything.

Comments can be viewed using psql's `\d` family of commands. Other user interfaces to retrieve comments can be built atop the same built-in functions that psql uses, namely `obj_description`, `col_description`, and `shobj_description` (see ???).

Parameters

<object_name>; <relation_name>.<column_name>; <aggregate_name>; <constraint_name>; <function_name>; <operator_name>; <policy_name>; <procedure_name>; <routine_name>; <rule_name>; <trigger_name>
The name of the object to be commented. Names of objects that reside in schemas (tables, functions, etc.) can be schema-qualified. When commenting on a column, <relation_name> must refer to a table, view, composite type, or foreign table.

<table_name>; <domain_name>
When creating a comment on a constraint, a trigger, a rule or a policy these parameters specify the name of the table or domain on which that object is defined.

<source_type>
The name of the source data type of the cast.

<target_type>
The name of the target data type of the cast.

<argmode>
The mode of a function, procedure, or aggregate argument: `IN`, `OUT`, `INOUT`, or `VARIADIC`. If omitted, the default is `IN`. Note that `COMMENT` does not actually pay any attention to `OUT` arguments, since only the input arguments are needed to determine the function's identity. So it is sufficient to list the `IN`, `INOUT`, and `VARIADIC` arguments.

<argname>
The name of a function, procedure, or aggregate argument. Note that `COMMENT` does not actually pay any attention to argument names, since only the argument data types are needed to determine the function's identity.

<argtype>
The data type of a function, procedure, or aggregate argument.

<large_object_oid>
The OID of the large object.

<left_type>; <right_type>
The data type(s) of the operator's arguments (optionally schema-qualified). Write `NONE` for the missing argument of a prefix operator.

`PROCEDURAL`
This is a noise word.

<type_name>
The name of the data type of the transform.

<lang_name>
The name of the language of the transform.

<string_literal>
The new comment contents, written as a string literal. An empty string (`''`) removes the comment.

`NULL`
Write `NULL` to remove the comment.

Notes

There is presently no security mechanism for viewing comments: any user connected to a database can see all the comments for objects in that database. For shared objects such as databases, roles, and tablespaces, comments are stored globally so any user connected to any database in the cluster can see all the comments for shared objects. Therefore, don't put security-critical information in comments.

Examples

Attach a comment to the table mytable:

```
COMMENT ON TABLE mytable IS 'This is my table.';
```

Remove it again:

```
COMMENT ON TABLE mytable IS NULL;
```

Some more examples:

```
COMMENT ON ACCESS METHOD gin IS 'GIN index access method';
COMMENT ON AGGREGATE my_aggregate (double precision) IS 'Computes sample variance';
COMMENT ON CAST (text AS int4) IS 'Allow casts from text to int4';
COMMENT ON COLLATION "fr_CA" IS 'Canadian French';
COMMENT ON COLUMN my_table.my_column IS 'Employee ID number';
COMMENT ON CONVERSION my_conv IS 'Conversion to UTF8';
COMMENT ON CONSTRAINT bar_col_cons ON bar IS 'Constrains column col';
COMMENT ON CONSTRAINT dom_col_constr ON DOMAIN dom IS 'Constrains col of domain';
COMMENT ON DATABASE my_database IS 'Development Database';
COMMENT ON DOMAIN my_domain IS 'Email Address Domain';
COMMENT ON EVENT TRIGGER abort_ddl IS 'Aborts all DDL commands';
COMMENT ON EXTENSION hstore IS 'implements the hstore data type';
COMMENT ON FOREIGN DATA WRAPPER mywrapper IS 'my foreign data wrapper';
COMMENT ON FOREIGN TABLE my_foreign_table IS 'Employee Information in other database';
COMMENT ON FUNCTION my_function (timestamp) IS 'Returns Roman Numeral';
COMMENT ON INDEX my_index IS 'Enforces uniqueness on employee ID';
COMMENT ON LANGUAGE plpython IS 'Python support for stored procedures';
COMMENT ON LARGE OBJECT 346344 IS 'Planning document';
COMMENT ON MATERIALIZED VIEW my_matview IS 'Summary of order history';
COMMENT ON OPERATOR ^ (text, text) IS 'Performs intersection of two texts';
COMMENT ON OPERATOR - (NONE, integer) IS 'Unary minus';
COMMENT ON OPERATOR CLASS int4ops USING btree IS '4 byte integer operators for btrees';
COMMENT ON OPERATOR FAMILY integer_ops USING btree IS 'all integer operators for btrees';
COMMENT ON POLICY my_policy ON mytable IS 'Filter rows by users';
COMMENT ON PROCEDURE my_proc (integer, integer) IS 'Runs a report';
COMMENT ON PUBLICATION alltables IS 'Publishes all operations on all tables';
```

```
COMMENT ON ROLE my_role IS 'Administration group for finance tables';
COMMENT ON ROUTINE my_routine (integer, integer) IS 'Runs a routine (which is a function or procedure)';
COMMENT ON RULE my_rule ON my_table IS 'Logs updates of employee records';
COMMENT ON SCHEMA my_schema IS 'Departmental data';
COMMENT ON SEQUENCE my_sequence IS 'Used to generate primary keys';
COMMENT ON SERVER myserver IS 'my foreign server';
COMMENT ON STATISTICS my_statistics IS 'Improves planner row estimations';
COMMENT ON SUBSCRIPTION alltables IS 'Subscription for all operations on all tables';
COMMENT ON TABLE my_schema.my_table IS 'Employee Information';
COMMENT ON TABLESPACE my_tablespace IS 'Tablespace for indexes';
COMMENT ON TEXT SEARCH CONFIGURATION my_config IS 'Special word filtering';
COMMENT ON TEXT SEARCH DICTIONARY swedish IS 'Snowball stemmer for Swedish language';
COMMENT ON TEXT SEARCH PARSER my_parser IS 'Splits text into words';
COMMENT ON TEXT SEARCH TEMPLATE snowball IS 'Snowball stemmer';
COMMENT ON TRANSFORM FOR hstore LANGUAGE plpython3u IS 'Transform between hstore and Python dict';
COMMENT ON TRIGGER my_trigger ON my_table IS 'Used for RI';
COMMENT ON TYPE complex IS 'Complex number data type';
COMMENT ON VIEW my_view IS 'View of departmental costs';
COMMENT ON VIEW my_view IS NULL;
```

Compatibility

There is no `COMMENT` command in the SQL standard.

□□□ #2
□ □□□ □□□ 2026-10-31 14:08:36 UTC
□ □□□ □□□ 2026-10-31 14:08:37 UTC