

NAME

ALTER PROCEDURE – change the definition of a procedure

SYNOPSIS

```
ALTER PROCEDURE name [ ( [ argmode ] [ argname ] argtype [, ...] ) ]
    action [ ... ] [ RESTRICT ]
ALTER PROCEDURE name [ ( [ argmode ] [ argname ] argtype [, ...] ) ]
    RENAME TO new_name
ALTER PROCEDURE name [ ( [ argmode ] [ argname ] argtype [, ...] ) ]
    OWNER TO { new_owner | CURRENT_ROLE | CURRENT_USER | SESSION_USER }
ALTER PROCEDURE name [ ( [ argmode ] [ argname ] argtype [, ...] ) ]
    SET SCHEMA new_schema
ALTER PROCEDURE name [ ( [ argmode ] [ argname ] argtype [, ...] ) ]
    [ NO ] DEPENDS ON EXTENSION extension_name
```

where *action* is one of:

```
[ EXTERNAL ] SECURITY INVOKER | [ EXTERNAL ] SECURITY DEFINER
SET configuration_parameter { TO | = } { value | DEFAULT }
SET configuration_parameter FROM CURRENT
RESET configuration_parameter
RESET ALL
```

DESCRIPTION

ALTER PROCEDURE changes the definition of a procedure.

You must own the procedure to use **ALTER PROCEDURE**. To change a procedure's schema, you must also have CREATE privilege on the new schema. To alter the owner, you must be able to SET ROLE to the new owning role, and that role must have CREATE privilege on the procedure's schema. (These restrictions enforce that altering the owner doesn't do anything you couldn't do by dropping and recreating the procedure. However, a superuser can alter ownership of any procedure anyway.)

PARAMETERS

name

The name (optionally schema-qualified) of an existing procedure. If no argument list is specified, the name must be unique in its schema.

argmode

The mode of an argument: IN, OUT, INOUT, or VARIADIC. If omitted, the default is IN.

argname

The name of an argument. Note that **ALTER PROCEDURE** does not actually pay any attention to argument names, since only the argument data types are used to determine the procedure's identity.

argtype

The data type(s) of the procedure's arguments (optionally schema-qualified), if any. See DROP PROCEDURE (**DROP PROCEDURE(7)**) for the details of how the procedure is looked up using the argument data type(s).

new_name

The new name of the procedure.

new_owner

The new owner of the procedure. Note that if the procedure is marked SECURITY DEFINER, it will subsequently execute as the new owner.

new_schema

The new schema for the procedure.

extension_name

This form marks the procedure as dependent on the extension, or no longer dependent on the extension

if NO is specified. A procedure that's marked as dependent on an extension is dropped when the extension is dropped, even if cascade is not specified. A procedure can depend upon multiple extensions, and will be dropped when any one of those extensions is dropped.

[EXTERNAL] SECURITY INVOKER

[EXTERNAL] SECURITY DEFINER

Change whether the procedure is a security definer or not. The key word EXTERNAL is ignored for SQL conformance. See CREATE PROCEDURE (**CREATE_PROCEDURE(7)**) for more information about this capability.

configuration_parameter
value

Add or change the assignment to be made to a configuration parameter when the procedure is called. If *value* is DEFAULT or, equivalently, RESET is used, the procedure-local setting is removed, so that the procedure executes with the value present in its environment. Use RESET ALL to clear all procedure-local settings. SET FROM CURRENT saves the value of the parameter that is current when **ALTER PROCEDURE** is executed as the value to be applied when the procedure is entered.

See **SET(7)** and Chapter 19 for more information about allowed parameter names and values.

RESTRICT

Ignored for conformance with the SQL standard.

EXAMPLES

To rename the procedure insert_data with two arguments of type integer to insert_record:

```
ALTER PROCEDURE insert_data(integer, integer) RENAME TO insert_record;
```

To change the owner of the procedure insert_data with two arguments of type integer to joe:

```
ALTER PROCEDURE insert_data(integer, integer) OWNER TO joe;
```

To change the schema of the procedure insert_data with two arguments of type integer to accounting:

```
ALTER PROCEDURE insert_data(integer, integer) SET SCHEMA accounting;
```

To mark the procedure insert_data(integer, integer) as being dependent on the extension myext:

```
ALTER PROCEDURE insert_data(integer, integer) DEPENDS ON EXTENSION myext;
```

To adjust the search path that is automatically set for a procedure:

```
ALTER PROCEDURE check_password(text) SET search_path = admin, pg_temp;
```

To disable automatic setting of *search_path* for a procedure:

```
ALTER PROCEDURE check_password(text) RESET search_path;
```

The procedure will now execute with whatever search path is used by its caller.

COMPATIBILITY

This statement is partially compatible with the **ALTER PROCEDURE** statement in the SQL standard. The standard allows more properties of a procedure to be modified, but does not provide the ability to rename a procedure, make a procedure a security definer, attach configuration parameter values to a procedure, or change the owner, schema, or volatility of a procedure. The standard also requires the RESTRICT key word, which is optional in PostgreSQL.

SEE ALSO

CREATE PROCEDURE (**CREATE_PROCEDURE(7)**), DROP PROCEDURE (**DROP_PROCEDURE(7)**), ALTER FUNCTION (**ALTER_FUNCTION(7)**), ALTER ROUTINE (**ALTER_ROUTINE(7)**)