

NAME

ALTER_PUBLICATION – change the definition of a publication

SYNOPSIS

```
ALTER PUBLICATION name ADD publication_object [, ...]
ALTER PUBLICATION name SET publication_object [, ...]
ALTER PUBLICATION name DROP publication_drop_object [, ...]
ALTER PUBLICATION name SET ( publication_parameter [= value] [, ...] )
ALTER PUBLICATION name OWNER TO { new_owner | CURRENT_ROLE | CURRENT_USER | SESSION_USER }
ALTER PUBLICATION name RENAME TO new_name
```

where *publication_object* is one of:

```
TABLE table_and_columns [, ... ]
TABLES IN SCHEMA { schema_name | CURRENT_SCHEMA } [, ... ]
```

and *publication_drop_object* is one of:

```
TABLE [ ONLY ] table_name [ * ] [, ... ]
TABLES IN SCHEMA { schema_name | CURRENT_SCHEMA } [, ... ]
```

and *table_and_columns* is:

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

DESCRIPTION

The command **ALTER PUBLICATION** can change the attributes of a publication.

The first three variants change which tables/schemas are part of the publication. The SET clause will replace the list of tables/schemas in the publication with the specified list; the existing tables/schemas that were present in the publication will be removed. The ADD and DROP clauses will add and remove one or more tables/schemas from the publication. Note that adding tables/schemas to a publication that is already subscribed to will require an ALTER SUBSCRIPTION ... REFRESH PUBLICATION action on the subscribing side in order to become effective. Note also that DROP TABLES IN SCHEMA will not drop any schema tables that were specified using FOR TABLE/ ADD TABLE.

The fourth variant of this command listed in the synopsis can change all of the publication properties specified in CREATE PUBLICATION (**CREATE_PUBLICATION(7)**). Properties not mentioned in the command retain their previous settings.

The remaining variants change the owner and the name of the publication.

You must own the publication to use **ALTER PUBLICATION**. Adding a table to a publication additionally requires owning that table. The ADD TABLES IN SCHEMA and SET TABLES IN SCHEMA to a publication requires the invoking user to be a superuser. 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 database. Also, the new owner of a FOR ALL TABLES or FOR TABLES IN SCHEMA publication must be a superuser. However, a superuser can change the ownership of a publication regardless of these restrictions.

Adding/Setting any schema when the publication also publishes a table with a column list, and vice versa is not supported.

PARAMETERS

name

The name of an existing publication whose definition is to be altered.

table_name

Name of an existing table. If ONLY is specified before the table name, only that table is affected. If ONLY is not specified, the table and all its descendant tables (if any) are affected. Optionally, * can be specified after the table name to explicitly indicate that descendant tables are included.

Optionally, a column list can be specified. See CREATE PUBLICATION (CREATE_PUBLICATION(7)) for details. Note that a subscription having several publications in which the same table has been published with different column lists is not supported. See Warning: Combining Column Lists from Multiple Publications for details of potential problems when altering column lists.

If the optional WHERE clause is specified, rows for which the *expression* evaluates to false or null will not be published. Note that parentheses are required around the expression. The *expression* is evaluated with the role used for the replication connection.

schema_name

Name of an existing schema.

SET (*publication_parameter* [= *value*] [, ...])

This clause alters publication parameters originally set by CREATE PUBLICATION (CREATE_PUBLICATION(7)). See there for more information.

Caution

Altering the `publish_via_partition_root` parameter can lead to data loss or duplication at the subscriber because it changes the identity and schema of the published tables. Note this happens only when a partition root table is specified as the replication target.

This problem can be avoided by refraining from modifying partition leaf tables after the **ALTER PUBLICATION ... SET** until the **ALTER SUBSCRIPTION ... REFRESH PUBLICATION** is executed and by only refreshing using the `copy_data = off` option.

new_owner

The user name of the new owner of the publication.

new_name

The new name for the publication.

EXAMPLES

Change the publication to publish only deletes and updates:

```
ALTER PUBLICATION noinsert SET (publish = 'update, delete');
```

Add some tables to the publication:

```
ALTER PUBLICATION mypublication ADD TABLE users (user_id, firstname), departments;
```

Change the set of columns published for a table:

```
ALTER PUBLICATION mypublication SET TABLE users (user_id, firstname, lastname), TABLE departments;
```

Add schemas marketing and sales to the publication sales_publication:

```
ALTER PUBLICATION sales_publication ADD TABLES IN SCHEMA marketing, sales;
```

Add tables users, departments and schema production to the publication production_publication:

```
ALTER PUBLICATION production_publication ADD TABLE users, departments, TABLES IN SCHEMA production;
```

COMPATIBILITY

ALTER PUBLICATION is a PostgreSQL extension.

SEE ALSO

CREATE PUBLICATION (CREATE_PUBLICATION(7)), DROP PUBLICATION (DROP_PUBLICATION(7)), CREATE SUBSCRIPTION (CREATE_SUBSCRIPTION(7)), ALTER SUBSCRIPTION (ALTER_SUBSCRIPTION(7))