NAME

ALTER_MATERIALIZED_VIEW − change the definition of a materialized view

SYNOPSIS

```
ALTER MATERIALIZED VIEW [ IF EXISTS ] name
  action [, ... ]
ALTER MATERIALIZED VIEW name
  [ NO ] DEPENDS ON EXTENSION extension_name
ALTER MATERIALIZED VIEW [ IF EXISTS ] name
  RENAME [ COLUMN ] column_name TO new_column_name
ALTER MATERIALIZED VIEW [ IF EXISTS ] name
  RENAME TO new_name
ALTER MATERIALIZED VIEW [ IF EXISTS ] name
  SET SCHEMA new_schema
ALTER MATERIALIZED VIEW ALL IN TABLESPACE name [ OWNED BY role_name [, ... ] ]
  SET TABLESPACE new_tablespace [ NOWAIT ]
```

where *action* is one of:

```
  ALTER [ COLUMN ] column_name SET STATISTICS integer
  ALTER [ COLUMN ] column_name SET ( attribute_option = value [, ... ] )
  ALTER [ COLUMN ] column_name RESET ( attribute_option [, ... ] )
  ALTER [ COLUMN ] column_name SET STORAGE { PLAIN | EXTERNAL | EXTENDED | MAIN | DEFAULT }
  ALTER [ COLUMN ] column_name SET COMPRESSION compression_method
  CLUSTER ON index_name
  SET WITHOUT CLUSTER
  SET ACCESS METHOD new_access_method
  SET TABLESPACE new_tablespace
  SET ( storage_parameter [= value] [, ... ] )
  RESET ( storage_parameter [, ... ] )
  OWNER TO { new_owner | CURRENT_ROLE | CURRENT_USER | SESSION_USER }
```

DESCRIPTION

ALTER MATERIALIZED VIEW changes various auxiliary properties of an existing materialized view.

You must own the materialized view to use **ALTER MATERIALIZED VIEW**. To change a materialized view'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 materialized view's schema. (These restrictions enforce that altering the owner doesn't do anything you couldn't do by dropping and recreating the materialized view. However, a superuser can alter ownership of any view anyway.)

The statement subforms and actions available for **ALTER MATERIALIZED VIEW** are a subset of those available for **ALTER TABLE**, and have the same meaning when used for materialized views. See the descriptions for **ALTER TABLE** for details.

PARAMETERS

name

The name (optionally schema−qualified) of an existing materialized view.

column_name

Name of an existing column.

extension_name

The name of the extension that the materialized view is to depend on (or no longer dependent on, if NO is specified). A materialized view that's marked as dependent on an extension is automatically dropped when the extension is dropped.

new_column_name

New name for an existing column.

new_owner
The user name of the new owner of the materialized view.

new_name
The new name for the materialized view.

new_schema
The new schema for the materialized view.

EXAMPLES

To rename the materialized view foo to bar:

```
ALTER MATERIALIZED VIEW foo RENAME TO bar;
```

COMPATIBILITY

ALTER MATERIALIZED VIEW is a PostgreSQL extension.

SEE ALSO

CREATE MATERIALIZED VIEW (**CREATE_MATERIALIZED_VIEW**(7)), DROP MATERIALIZED VIEW (**DROP_MATERIALIZED_VIEW**(7)), REFRESH MATERIALIZED VIEW (**REFRESH_MATERIALIZED_VIEW**(7))