www.postgresql.org /docs/current/sql-alterdefaultprivileges.html

Unknown Title

⋮ 13/11/2025

ALTER DEFAULT PRIVILEGES

ALTER DEFAULT PRIVILEGES — define default access privileges

Synopsis

```
ALTER DEFAULT PRIVILEGES
    [ FOR { ROLE | USER }
 target_role
 [, ...] ]
    [ IN SCHEMA
 schema_name
 [, ...] ]

 abbreviated_grant_or_revoke


where
 abbreviated_grant_or_revoke
 is one of:

GRANT { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE | REFERENCES | TRIGGER
| MAINTAIN }
    [, ...] | ALL [ PRIVILEGES ] }
    ON TABLES
    TO { [ GROUP ]
 role_name
 | PUBLIC } [, ...] [ WITH GRANT OPTION ]

GRANT { { USAGE | SELECT | UPDATE }
    [, ...] | ALL [ PRIVILEGES ] }
    ON SEQUENCES
    TO { [ GROUP ]
 role_name
 | PUBLIC } [, ...] [ WITH GRANT OPTION ]

GRANT { EXECUTE | ALL [ PRIVILEGES ] }
    ON { FUNCTIONS | ROUTINES }
    TO { [ GROUP ]
 role_name
 | PUBLIC } [, ...] [ WITH GRANT OPTION ]

GRANT { USAGE | ALL [ PRIVILEGES ] }
    ON TYPES
    TO { [ GROUP ]
 role_name
 | PUBLIC } [, ...] [ WITH GRANT OPTION ]

GRANT { { USAGE | CREATE }
    [, ...] | ALL [ PRIVILEGES ] }
    ON SCHEMAS
    TO { [ GROUP ]
 role_name
 | PUBLIC } [, ...] [ WITH GRANT OPTION ]

GRANT { { SELECT | UPDATE }
    [, ...] | ALL [ PRIVILEGES ] }
    ON LARGE OBJECTS
    TO { [ GROUP ]
 role_name
 | PUBLIC } [, ...] [ WITH GRANT OPTION ]

REVOKE [ GRANT OPTION FOR ]
    { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE | REFERENCES | TRIGGER |
MAINTAIN }
    [, ...] | ALL [ PRIVILEGES ] }
    ON TABLES
    FROM { [ GROUP ]
 role_name
 | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ]

REVOKE [ GRANT OPTION FOR ]
    { { USAGE | SELECT | UPDATE }
    [, ...] | ALL [ PRIVILEGES ] }
    ON SEQUENCES
    FROM { [ GROUP ]
```

```
role_name
```

```
 | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ]

REVOKE [ GRANT OPTION FOR ]
    { EXECUTE | ALL [ PRIVILEGES ] }
    ON { FUNCTIONS | ROUTINES }
    FROM { [ GROUP ]
```

```
role_name
```

```
 | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ]

REVOKE [ GRANT OPTION FOR ]
    { USAGE | ALL [ PRIVILEGES ] }
    ON TYPES
    FROM { [ GROUP ]
```

```
role_name
```

```
 | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ]

REVOKE [ GRANT OPTION FOR ]
    { { USAGE | CREATE }
    [, ...] | ALL [ PRIVILEGES ] }
    ON SCHEMAS
    FROM { [ GROUP ]
```

```
role_name
```

```
 | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ]

REVOKE [ GRANT OPTION FOR ]
    { { SELECT | UPDATE }
    [, ...] | ALL [ PRIVILEGES ] }
    ON LARGE OBJECTS
    FROM { [ GROUP ]
```

```
role_name
```

```
 | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ]
```

Description

`ALTER DEFAULT PRIVILEGES` allows you to set the privileges that will be applied to objects created in the future. (It does not affect privileges assigned to already-existing objects.) Privileges can be set globally (i.e., for all objects created in the current database), or just for objects created in specified schemas.

While you can change your own default privileges and the defaults of roles that you are a member of, at object creation time, new object permissions are only affected by the default privileges of the current role, and are not inherited from any roles in which the current role is a member.

As explained in Section 5.8, the default privileges for any object type normally grant all grantable permissions to the object owner, and may grant some privileges to `PUBLIC` as well. However, this behavior can be changed by altering the global default privileges with `ALTER DEFAULT PRIVILEGES`.

Currently, only the privileges for schemas, tables (including views and foreign tables), sequences, functions, types (including domains), and large objects can be altered. For this command, functions include aggregates and procedures. The words `FUNCTIONS` and `ROUTINES` are equivalent in this command. (`ROUTINES` is preferred going forward as the standard term for functions and procedures taken together. In earlier PostgreSQL releases, only the word `FUNCTIONS` was allowed. It is not possible to set default privileges for functions and procedures separately.)

Default privileges that are specified per-schema are added to whatever the global default privileges are for the particular object type. This means you cannot revoke privileges per-schema if they are granted globally (either by default, or according to a previous `ALTER DEFAULT PRIVILEGES` command that did not specify a schema). Per-schema `REVOKE` is only useful to reverse the effects of a previous per-schema `GRANT`.

Parameters

`target_role`

Change default privileges for objects created by the *`target_role`*, or the current role if unspecified.

`schema_name`

The name of an existing schema. If specified, the default privileges are altered for objects later created in that schema. If `IN SCHEMA` is omitted, the global default privileges are altered. `IN SCHEMA` is not allowed when setting privileges for schemas and large objects, since schemas can't be nested and large objects don't belong to a schema.

`role_name`

The name of an existing role to grant or revoke privileges for. This parameter, and all the other parameters in *`abbreviated_grant_or_revoke`*, act as described under GRANT or REVOKE, except that one is setting permissions for a whole class of objects rather than specific named objects.

Notes

Use psql's \ddp command to obtain information about existing assignments of default privileges. The meaning of the privilege display is the same as explained for \dp in Section 5.8.

If you wish to drop a role for which the default privileges have been altered, it is necessary to reverse the changes in its default privileges or use DROP OWNED BY to get rid of the default privileges entry for the role.

Examples

Grant SELECT privilege to everyone for all tables (and views) you subsequently create in schema myschema, and allow role webuser to INSERT into them too:

```
ALTER DEFAULT PRIVILEGES IN SCHEMA myschema GRANT SELECT ON TABLES TO PUBLIC;
ALTER DEFAULT PRIVILEGES IN SCHEMA myschema GRANT INSERT ON TABLES TO
webuser;
```

Undo the above, so that subsequently-created tables won't have any more permissions than normal:

```
ALTER DEFAULT PRIVILEGES IN SCHEMA myschema REVOKE SELECT ON TABLES FROM
PUBLIC;
ALTER DEFAULT PRIVILEGES IN SCHEMA myschema REVOKE INSERT ON TABLES FROM
webuser;
```

Remove the public EXECUTE permission that is normally granted on functions, for all functions subsequently created by role admin:

```
ALTER DEFAULT PRIVILEGES FOR ROLE admin REVOKE EXECUTE ON FUNCTIONS FROM
PUBLIC;
```

Note however that you *cannot* accomplish that effect with a command limited to a single schema. This command has no effect, unless it is undoing a matching GRANT:

```
ALTER DEFAULT PRIVILEGES IN SCHEMA public REVOKE EXECUTE ON FUNCTIONS FROM
PUBLIC;
```

That's because per-schema default privileges can only add privileges to the global setting, not remove privileges granted by it.

Compatibility

There is no ALTER DEFAULT PRIVILEGES statement in the SQL standard.

See Also

GRANT, REVOKE