NAME

ALTER_ROLE − change a database role

SYNOPSIS

ALTER ROLE *role_specification* [WITH] *option* [...]

where *option* can be:

```
    SUPERUSER | NOSUPERUSER
  | CREATEDB | NOCREATEDB
  | CREATEROLE | NOCREATEROLE
  | INHERIT | NOINHERIT
  | LOGIN | NOLOGIN
  | REPLICATION | NOREPLICATION
  | BYPASSRLS | NOBYPASSRLS
  | CONNECTION LIMIT connlimit
  | [ ENCRYPTED ] PASSWORD 'password' | PASSWORD NULL
  | VALID UNTIL 'timestamp'
```

ALTER ROLE *name* RENAME TO *new_name*

ALTER ROLE { *role_specification* | ALL } [IN DATABASE *database_name*] SET *configuration_parameter* { TO | = } { *val*
ALTER ROLE { *role_specification* | ALL } [IN DATABASE *database_name*] SET *configuration_parameter* FROM CURRE
ALTER ROLE { *role_specification* | ALL } [IN DATABASE *database_name*] RESET *configuration_parameter*
ALTER ROLE { *role_specification* | ALL } [IN DATABASE *database_name*] RESET ALL

where *role_specification* can be:

```
    role_name
  | CURRENT_ROLE
  | CURRENT_USER
  | SESSION_USER
```

DESCRIPTION

ALTER ROLE changes the attributes of a PostgreSQL role.

The first variant of this command listed in the synopsis can change many of the role attributes that can be specified in **CREATE ROLE**. (All the possible attributes are covered, except that there are no options for adding or removing memberships; use **GRANT** and **REVOKE** for that.) Attributes not mentioned in the command retain their previous settings. Database superusers can change any of these settings for any role, except for changing the SUPERUSER property for the bootstrap superuser. Non−superuser roles having CREATEROLE privilege can change most of these properties, but only for non−superuser and non−replication roles for which they have been granted ADMIN OPTION. Non−superusers cannot change the SUPERUSER property and can change the CREATEDB, REPLICATION, and BYPASSRLS properties only if they possess the corresponding property themselves. Ordinary roles can only change their own password.

The second variant changes the name of the role. Database superusers can rename any role. Roles having CREATEROLE privilege can rename non−superuser roles for which they have been granted ADMIN OPTION. The current session user cannot be renamed. (Connect as a different user if you need to do that.) Because MD5−encrypted passwords use the role name as cryptographic salt, renaming a role clears its password if the password is MD5−encrypted.

The remaining variants change a role's session default for a configuration variable, either for all databases or, when the IN DATABASE clause is specified, only for sessions in the named database. If ALL is specified instead of a role name, this changes the setting for all roles. Using ALL with IN DATABASE is effectively the same as using the command ALTER DATABASE ... SET ....

Whenever the role subsequently starts a new session, the specified value becomes the session default, overriding whatever setting is present in postgresql.conf or has been received from the **postgres** command line. This only happens at login time; executing **SET ROLE** or **SET SESSION AUTHORIZATION** does not cause new configuration values to be set. Settings set for all databases are overridden by database−specific settings attached to a role. Settings for specific databases or specific roles override settings for all roles.

Superusers can change anyone's session defaults. Roles having CREATEROLE privilege can change defaults for non−superuser roles for which they have been granted ADMIN OPTION. Ordinary roles can only set defaults for themselves. Certain configuration variables cannot be set this way, or can only be set if a superuser issues the command. Only superusers can change a setting for all roles in all databases.

PARAMETERS

name
: The name of the role whose attributes are to be altered.

CURRENT_ROLE
CURRENT_USER
: Alter the current user instead of an explicitly identified role.

SESSION_USER
: Alter the current session user instead of an explicitly identified role.

SUPERUSER
NOSUPERUSER
CREATEDB
NOCREATEDB
CREATEROLE
NOCREATEROLE
INHERIT
NOINHERIT
LOGIN
NOLOGIN
REPLICATION
NOREPLICATION
BYPASSRLS
NOBYPASSRLS
CONNECTION LIMIT *connlimit*
[ENCRYPTED] PASSWORD '*password*'
PASSWORD NULL
VALID UNTIL '*timestamp*'
: These clauses alter attributes originally set by **CREATE ROLE**. For more information, see the **CREATE ROLE** reference page.

new_name
: The new name of the role.

database_name
: The name of the database the configuration variable should be set in.

configuration_parameter
value
: Set this role's session default for the specified configuration parameter to the given value. If *value* is DEFAULT or, equivalently, RESET is used, the role−specific variable setting is removed, so the role will inherit the system−wide default setting in new sessions. Use RESET ALL to clear all role−specific settings. SET FROM CURRENT saves the session's current value of the parameter as the role−specific value. If IN DATABASE is specified, the configuration parameter is set or removed for the given role and database only.

Role−specific variable settings take effect only at login; **SET ROLE** and **SET SESSION AUTHORIZATION** do not process role−specific variable settings.

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

NOTES

Use **CREATE ROLE** to add new roles, and **DROP ROLE** to remove a role.

ALTER ROLE cannot change a role's memberships. Use **GRANT** and **REVOKE** to do that.

Caution must be exercised when specifying an unencrypted password with this command. The password will be transmitted to the server in cleartext, and it might also be logged in the client's command history or the server log. **psql**(1) contains a command **\password** that can be used to change a role's password without exposing the cleartext password.

It is also possible to tie a session default to a specific database rather than to a role; see ALTER DATABASE (**ALTER_DATABASE**(7)). If there is a conflict, database−role−specific settings override role−specific ones, which in turn override database−specific ones.

EXAMPLES

Change a role's password:

```
ALTER ROLE davide WITH PASSWORD 'hu8jmn3';
```

Remove a role's password:

```
ALTER ROLE davide WITH PASSWORD NULL;
```

Change a password expiration date, specifying that the password should expire at midday on 4th May 2015 using the time zone which is one hour ahead of UTC:

```
ALTER ROLE chris VALID UNTIL 'May 4 12:00:00 2015 +1';
```

Make a password valid forever:

```
ALTER ROLE fred VALID UNTIL 'infinity';
```

Give a role the ability to manage other roles and create new databases:

```
ALTER ROLE miriam CREATEROLE CREATEDB;
```

Give a role a non−default setting of the maintenance_work_mem parameter:

```
ALTER ROLE worker_bee SET maintenance_work_mem = 100000;
```

Give a role a non−default, database−specific setting of the client_min_messages parameter:

```
ALTER ROLE fred IN DATABASE devel SET client_min_messages = DEBUG;
```

COMPATIBILITY

The **ALTER ROLE** statement is a PostgreSQL extension.

SEE ALSO

CREATE ROLE (**CREATE_ROLE**(7)), DROP ROLE (**DROP_ROLE**(7)), ALTER DATABASE (**ALTER_DATABASE**(7)), **SET**(7)