alter_database

- Description
- Parameters
- Notes
- Examples
- Compatibility
- See Also

ALTER DATABASE

ALTER DATABASE

7

SQL - Language Statements

ALTER DATABASE

change a database

ALTER DATABASE

name

[[WITH]

option

[...]]

where

option

can be:

ALLOW_CONNECTIONS

allowconn

CONNECTION LIMIT

connlimit

IS_TEMPLATE

istemplate

ALTER DATABASE

name

RENAME TO

new_name

ALTER DATABASE

name

OWNER TO {

new_owner

| CURRENT_ROLE | CURRENT_USER | SESSION_USER } ALTER DATABASE

name

SET TABLESPACE

new_tablespace

ALTER DATABASE

name

REFRESH COLLATION VERSION ALTER DATABASE

name

SET

configuration_parameter

{ TO | = } {

value

| DEFAULT } ALTER DATABASE

name

SET

configuration_parameter

FROM CURRENT ALTER DATABASE

name

RESET

configuration_parameter

ALTER DATABASE

name

RESET ALL

Description

`ALTER DATABASE` changes the attributes of a database.

The first form changes certain per-database settings. (See below for details.) Only the database owner or a superuser can change these settings.

The second form changes the name of the database. Only the database owner or a superuser can rename a database; non-superuser owners must also have the `CREATEDB` privilege. The current database cannot be renamed. (Connect to a different database if you need to do that.)

The third form changes the owner of the database. To alter the owner, you must be able to `SET ROLE` to the new owning role, and you must have the `CREATEDB` privilege. (Note that superusers have all these privileges automatically.)

The fourth form changes the default tablespace of the database. Only the database owner or a superuser can do this; you must also have create privilege for the new tablespace. This command physically moves any tables or indexes in the database's old default tablespace to the new tablespace. The new default tablespace must be empty for this database, and no one can be connected to the database. Tables and indexes in non-default tablespaces are unaffected. The

method used to copy files to the new tablespace is affected by the ??? setting.

The remaining forms change the session default for a run-time configuration variable for a PostgreSQL database. Whenever a new session is subsequently started in that database, the specified value becomes the session default value. The database-specific default overrides whatever setting is present in `postgresql.conf` or has been received from the `postgres` command line. Only the database owner or a superuser can change the session defaults for a database. Certain variables cannot be set this way, or can only be set by a superuser.

Parameters

<name>
The name of the database whose attributes are to be altered.

<allowconn>
If false then no one can connect to this database.

<connlimit>
How many concurrent connections can be made to this database. -1 means no limit.

<istemplate>
If true, then this database can be cloned by any user with `CREATEDB` privileges; if false, then only superusers or the owner of the database can clone it.

<new_name>
The new name of the database.

<new_owner>
The new owner of the database.

<new_tablespace>
The new default tablespace of the database.

This form of the command cannot be executed inside a transaction block.

`REFRESH COLLATION VERSION`

Update the database collation version. See ??? for background.

<configuration_parameter>; <value>
Set this database's session default for the specified configuration parameter to the given value. If <value> is `DEFAULT` or, equivalently, `RESET` is used, the database-specific setting is removed, so the system-wide default setting will be inherited in new sessions. Use `RESET ALL` to clear all database-specific settings. `SET FROM CURRENT` saves the session's current value of the parameter as

the database-specific value.

See ??? and ??? for more information about allowed parameter names and values.

Notes

It is also possible to tie a session default to a specific role rather than to a database; see ???. Role-specific settings override database-specific ones if there is a conflict.

Examples

To disable index scans by default in the database `test`:

```
ALTER DATABASE test SET enable_indexscan TO off;
```

Compatibility

The `ALTER DATABASE` statement is a PostgreSQL extension.

See Also

□□□ #2
□ □□□ □□□ 2026-10-31 14:07:06 UTC
□ □□□ □□□ 2026-10-31 14:07:06 UTC