

set_session_auth



- [Description](#)
 - [Notes](#)
 - [Examples](#)
 - [Compatibility](#)
 - [See Also](#)
-

SET SESSION AUTHORIZATION

SET SESSION AUTHORIZATION

7

SQL - Language Statements

SET SESSION AUTHORIZATION

set the session user identifier and the current user identifier of the current session

SET [SESSION | LOCAL] SESSION AUTHORIZATION

user_name

SET [SESSION | LOCAL] SESSION AUTHORIZATION DEFAULT RESET SESSION AUTHORIZATION

Description

This command sets the session user identifier and the current user identifier of the current SQL session to be <user_name>. The user name can be written as either an identifier or a string literal. Using this command, it is possible, for example, to temporarily become an unprivileged user and later switch back to being a superuser.

The session user identifier is initially set to be the (possibly authenticated) user name provided by the client. The current user identifier is normally equal to the session user identifier, but might change temporarily in the context of `SECURITY DEFINER` functions and similar mechanisms; it can

also be changed by `SET ROLE`. The current user identifier is relevant for permission checking.

The session user identifier can be changed only if the initial session user (the authenticated user) has the superuser privilege. Otherwise, the command is accepted only if it specifies the authenticated user name.

The `SESSION` and `LOCAL` modifiers act the same as for the regular `SET` command.

The `DEFAULT` and `RESET` forms reset the session and current user identifiers to be the originally authenticated user name. These forms can be executed by any user.

Notes

`SET SESSION AUTHORIZATION` cannot be used within a `SECURITY DEFINER` function.

Examples

```
SELECT SESSION_USER, CURRENT_USER;
```

```
session_user | current_user
-----+-----
peter        | peter
```

```
SET SESSION AUTHORIZATION 'paul';
```

```
SELECT SESSION_USER, CURRENT_USER;
```

```
session_user | current_user
-----+-----
paul         | paul
```

Compatibility

The SQL standard allows some other expressions to appear in place of the literal `<user_name>`, but these options are not important in practice. PostgreSQL allows identifier syntax (`"username"`), which SQL does not. SQL does not allow this command during a transaction; PostgreSQL does not

make this restriction because there is no reason to. The `SESSION` and `LOCAL` modifiers are a PostgreSQL extension, as is the `RESET` syntax.

The privileges necessary to execute this command are left implementation-defined by the standard.

See Also

<div><div></div><div></div><div></div></div>	#2	
<div><div></div><div></div><div></div></div>		2026-10-31 14:14:50 UTC
<div><div></div><div></div><div></div></div>		2026-10-31 14:14:51 UTC