create_event_trigger

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

CREATE EVENT TRIGGER

CREATE EVENT TRIGGER

7

SQL - Language Statements

CREATE EVENT TRIGGER

define a new event trigger

CREATE EVENT TRIGGER

name

ON

event

[WHEN

filter_variable

IN (

filter_value

[, ...]) [AND ...]] EXECUTE { FUNCTION | PROCEDURE }

function_name

()

Description

`CREATE EVENT TRIGGER` creates a new event trigger. Whenever the designated event occurs and the `WHEN` condition associated with the trigger, if any, is satisfied, the trigger function will be executed.

For a general introduction to event triggers, see ???. The user who creates an event trigger becomes its owner.

Parameters

<name>
The name to give the new trigger. This name must be unique within the database.

<event>

The name of the event that triggers a call to the given function. See ??? for more information on event names.

<filter_variable>
The name of a variable used to filter events. This makes it possible to restrict the firing of the trigger to a subset of the cases in which it is supported. Currently the only supported <filter_variable> is `TAG`.

<filter_value>
A list of values for the associated <filter_variable> for which the trigger should fire. For `TAG`, this means a list of command tags (e.g., `'DROP FUNCTION'`).

<function_name>
A user-supplied function that is declared as taking no argument and returning type `event_trigger`.

In the syntax of `CREATE EVENT TRIGGER`, the keywords `FUNCTION` and `PROCEDURE` are equivalent, but the referenced function must in any case be a function, not a procedure. The use of the keyword `PROCEDURE` here is historical and deprecated.

Notes

Only superusers can create event triggers.

Event triggers are disabled in single-user mode (see ???) as well as when ??? is set to `false`. If an erroneous event trigger disables the database so much that you can't even drop the trigger, restart with ??? set to `false` to temporarily disable event triggers, or in single-user mode, and you'll be able to do that.

Examples

Forbid the execution of any DDL command:

```
CREATE OR REPLACE FUNCTION abort_any_command()
  RETURNS event_trigger
 LANGUAGE plpgsql
  AS $$
BEGIN
  RAISE EXCEPTION 'command % is disabled', tg_tag;
END;
$$;

CREATE EVENT TRIGGER abort_ddl ON ddl_command_start
   EXECUTE FUNCTION abort_any_command();
```

Compatibility

There is no `CREATE EVENT TRIGGER` statement in the SQL standard.

See Also

□□□ #2
□ □□□ □□□ 2026-10-31 14:08:55 UTC
□ □□□ □□□ 2026-10-31 14:08:56 UTC