

NAME

DO – execute an anonymous code block

SYNOPSIS

DO [LANGUAGE *lang_name*] *code*

DESCRIPTION

DO executes an anonymous code block, or in other words a transient anonymous function in a procedural language.

The code block is treated as though it were the body of a function with no parameters, returning void. It is parsed and executed a single time.

The optional LANGUAGE clause can be written either before or after the code block.

PARAMETERS

code

The procedural language code to be executed. This must be specified as a string literal, just as in **CREATE FUNCTION**. Use of a dollar-quoted literal is recommended.

lang_name

The name of the procedural language the code is written in. If omitted, the default is plpgsql.

NOTES

The procedural language to be used must already have been installed into the current database by means of **CREATE EXTENSION**. plpgsql is installed by default, but other languages are not.

The user must have USAGE privilege for the procedural language, or must be a superuser if the language is untrusted. This is the same privilege requirement as for creating a function in the language.

If **DO** is executed in a transaction block, then the procedure code cannot execute transaction control statements. Transaction control statements are only allowed if **DO** is executed in its own transaction.

EXAMPLES

Grant all privileges on all views in schema public to role webuser:

```
DO $$DECLARE r record;
BEGIN
  FOR r IN SELECT table_schema, table_name FROM information_schema.tables
    WHERE table_type = 'VIEW' AND table_schema = 'public'
  LOOP
    EXECUTE 'GRANT ALL ON ' || quote_ident(r.table_schema) || '.' || quote_ident(r.table_name) || ' TO webuser';
  END LOOP;
END$$;
```

COMPATIBILITY

There is no **DO** statement in the SQL standard.

SEE ALSO

CREATE LANGUAGE (**CREATE_LANGUAGE(7)**)