February 16, 2026: Out-of-cycle release scheduled for February 26, 2026

Documentation → PostgreSQL 18
Supported Versions: Current (18) / 17 / 16 / 15 / 14
Development Versions: devel
Unsupported versions: 13 / 12 / 11 / 10 / 9.6 / 9.5 / 9.4 / 9.3 / 9.2 / 9.1 / 9.0 / 8.4 / 8.3 / 8.2 / 8.1 / 8.0 / 7.4 / 7.3 / 7.2 / 7.1

Search the documentation for...

CREATE DATABASE
Prev Up SQL Commands Home Next

CREATE DATABASE

CREATE DATABASE — create a new database

Synopsis

```
CREATE DATABASE name
    [ WITH ] [ OWNER [=] user_name ]
           [ TEMPLATE [=] template ]
           [ ENCODING [=] encoding ]
           [ STRATEGY [=] strategy ]
           [ LOCALE [=] locale ]
           [ LC_COLLATE [=] lc_collate ]
           [ LC_CTYPE [=] lc_ctype ]
           [ BUILTIN_LOCALE [=] builtin_locale ]
           [ ICU_LOCALE [=] icu_locale ]
           [ ICU_RULES [=] icu_rules ]
           [ LOCALE_PROVIDER [=] locale_provider ]
           [ COLLATION_VERSION = collation_version ]
           [ TABLESPACE [=] tablespace_name ]
           [ ALLOW_CONNECTIONS [=] allowconn ]
           [ CONNECTION LIMIT [=] connlimit ]
           [ IS_TEMPLATE [=] istemplate ]
           [ OID [=] oid ]
```

Description

`CREATE DATABASE` creates a new PostgreSQL database.

To create a database, you must be a superuser or have the special `CREATEDB` privilege. See CREATE ROLE.

By default, the new database will be created by cloning the standard system database `template1`. A different template can be specified by writing `TEMPLATE` ***`name`***. In particular, by writing `TEMPLATE template0`, you can create a pristine database (one where no user-defined objects exist and where the system objects have not been altered) containing only the standard objects predefined by your version of PostgreSQL. This is useful if you wish to avoid copying any installation-local objects that might have been added to `template1`.

Parameters

`name`

The name of a database to create.

`user_name`

The role name of the user who will own the new database, or `DEFAULT` to use the default (namely, the user executing the command). To create a database owned by another role, you must be able to `SET ROLE` to that role.

`template`

The name of the template from which to create the new database, or `DEFAULT` to use the default template (`template1`).

`encoding`

Character set encoding to use in the new database. Specify a string constant (e.g., `'SQL_ASCII'`), or an integer encoding number, or `DEFAULT` to use the default encoding (namely, the encoding of the template database). The character sets supported by the PostgreSQL server are described in Section 23.3.1. See below for additional restrictions.

strategy

Strategy to be used in creating the new database. If the `WAL_LOG` strategy is used, the database will be copied block by block and each block will be separately written to the write-ahead log. This is the most efficient strategy in cases where the template database is small, and therefore it is the default. The older `FILE_COPY` strategy is also available. This strategy writes a small record to the write-ahead log for each tablespace used by the target database. Each such record represents copying an entire directory to a new location at the filesystem level. While this does reduce the write-ahead log volume substantially, especially if the template database is large, it also forces the system to perform a checkpoint both before and after the creation of the new database. In some situations, this may have a noticeable negative impact on overall system performance. The `FILE_COPY` strategy is affected by the file_copy_method setting.

locale

Sets the default collation order and character classification in the new database. Collation affects the sort order applied to strings, e.g., in queries with `ORDER BY`, as well as the order used in indexes on text columns. Character classification affects the categorization of characters, e.g., lower, upper, and digit. Also sets the associated aspects of the operating system environment, `LC_COLLATE` and `LC_CTYPE`. The default is the same setting as the template database. See Section 23.2.2.3.1 and Section 23.2.2.3.2 for details.

Can be overridden by setting ***lc_collate***, ***lc_ctype***, ***builtin_locale***, or ***icu_locale*** individually.

If ***locale_provider*** is `builtin`, then ***locale*** or ***builtin_locale*** must be specified and set to either `C`, `C.UTF-8`, or `PG_UNICODE_FAST`.

> ### Tip
>
> The other locale settings lc_messages, lc_monetary, lc_numeric, and lc_time are not fixed per database and are not set by this command. If you want to make them the default for a specific database, you can use `ALTER DATABASE ... SET`.

lc_collate

Sets `LC_COLLATE` in the database server's operating system environment. The default is the setting of ***locale*** if specified, otherwise the same setting as the template database. See below for additional restrictions.

If ***locale_provider*** is `libc`, also sets the default collation order to use in the new database, overriding the setting ***locale***.

lc_ctype

Sets `LC_CTYPE` in the database server's operating system environment. The default is the setting of ***locale*** if specified, otherwise the same setting as the template database. See below for additional restrictions.

If ***locale_provider*** is `libc`, also sets the default character classification to use in the new database, overriding the setting ***locale***.

builtin_locale

Specifies the builtin provider locale for the database default collation order and character classification, overriding the setting ***locale***. The locale provider must be `builtin`. The default is the setting of ***locale*** if specified; otherwise the same setting as the template database.

The locales available for the `builtin` provider are `C`, `C.UTF-8` and `PG_UNICODE_FAST`.

icu_locale

Specifies the ICU locale (see Section 23.2.2.3.2) for the database default collation order and character classification, overriding the setting ***locale***. The locale provider must be ICU. The default is the setting of ***locale*** if specified; otherwise the same setting as the template database.

icu_rules

Specifies additional collation rules to customize the behavior of the default collation of this database. This is supported for ICU only. See Section 23.2.3.4 for details.

locale_provider

Specifies the provider to use for the default collation in this database. Possible values are `builtin`, `icu` (if the server was built with ICU support) or `libc`. By default, the provider is the same as that of the ***template***. See Section 23.1.4 for details.

collation_version

Specifies the collation version string to store with the database. Normally, this should be omitted, which will cause the version to be computed from the actual version of the database collation as provided by the operating system. This option is intended to be used by `pg_upgrade` for copying the version from an existing installation.

See also ALTER DATABASE for how to handle database collation version mismatches.

tablespace_name #

The name of the tablespace that will be associated with the new database, or DEFAULT to use the template database's tablespace. This tablespace will be the default tablespace used for objects created in this database. See CREATE TABLESPACE for more information.

allowconn

If false then no one can connect to this database. The default is true, allowing connections (except as restricted by other mechanisms, such as GRANT/REVOKE CONNECT).

connlimit

How many concurrent connections can be made to this database. -1 (the default) means no limit.

istemplate

If true, then this database can be cloned by any user with CREATEDB privileges; if false (the default), then only superusers or the owner of the database can clone it.

oid

The object identifier to be used for the new database. If this parameter is not specified, PostgreSQL will choose a suitable OID automatically. This parameter is primarily intended for internal use by pg_upgrade, and only pg_upgrade can specify a value less than 16384.

Optional parameters can be written in any order, not only the order illustrated above.

Notes

CREATE DATABASE cannot be executed inside a transaction block.

Errors along the line of “could not initialize database directory” are most likely related to insufficient permissions on the data directory, a full disk, or other file system problems.

Use DROP DATABASE to remove a database.

The program **createdb** is a wrapper program around this command, provided for convenience.

Database-level configuration parameters (set via ALTER DATABASE) and database-level permissions (set via GRANT) are not copied from the template database.

Although it is possible to copy a database other than template1 by specifying its name as the template, this is not (yet) intended as a general-purpose “COPY DATABASE” facility. The principal limitation is that no other sessions can be connected to the template database while it is being copied. CREATE DATABASE will fail if any other connection exists when it starts; otherwise, new connections to the template database are locked out until CREATE DATABASE completes. See Section 22.3 for more information.

The character set encoding specified for the new database must be compatible with the chosen locale settings (LC_COLLATE and LC_CTYPE). If the locale is C (or equivalently POSIX), then all encodings are allowed, but for other locale settings there is only one encoding that will work properly. (On Windows, however, UTF-8 encoding can be used with any locale.) CREATE DATABASE will allow superusers to specify SQL_ASCII encoding regardless of the locale settings, but this choice is deprecated and may result in misbehavior of character-string functions if data that is not encoding-compatible with the locale is stored in the database.

The encoding and locale settings must match those of the template database, except when template0 is used as template. This is because other databases might contain data that does not match the specified encoding, or might contain indexes whose sort ordering is affected by LC_COLLATE and LC_CTYPE. Copying such data would result in a database that is corrupt according to the new settings. template0, however, is known to not contain any data or indexes that would be affected.

There is currently no option to use a database locale with nondeterministic comparisons (see CREATE COLLATION for an explanation). If this is needed, then per-column collations would need to be used.

The CONNECTION LIMIT option is only enforced approximately; if two new sessions start at about the same time when just one connection “slot” remains for the database, it is possible that both will fail. Also, the limit is not enforced against superusers or background worker processes.

Examples

To create a new database:

```
CREATE DATABASE lusiadas;
```

To create a database sales owned by user salesapp with a default tablespace of salesspace:

```
CREATE DATABASE sales OWNER salesapp TABLESPACE salesspace;
```

To create a database `music` with a different locale:

```
CREATE DATABASE music
    LOCALE 'sv_SE.utf8'
    TEMPLATE template0;
```

In this example, the `TEMPLATE template0` clause is required if the specified locale is different from the one in `template1`. (If it is not, then specifying the locale explicitly is redundant.)

To create a database `music2` with a different locale and a different character set encoding:

```
CREATE DATABASE music2
    LOCALE 'sv_SE.iso885915'
    ENCODING LATIN9
    TEMPLATE template0;
```

The specified locale and encoding settings must match, or an error will be reported.

Note that locale names are specific to the operating system, so that the above commands might not work in the same way everywhere.

Compatibility

There is no `CREATE DATABASE` statement in the SQL standard. Databases are equivalent to catalogs, whose creation is implementation-defined.

See Also

ALTER DATABASE, DROP DATABASE

Prev Up Next
CREATE CONVERSION Home CREATE DOMAIN

Submit correction

If you see anything in the documentation that is not correct, does not match your experience with the particular feature or requires further clarification, please use this form to report a documentation issue.

Policies | Code of Conduct | About PostgreSQL | Contact