

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

This document in other formats: [PDF](#) | [ePub](#) | [Tardball](#)

[pgAdmin 4.4.18 documentation](#) >

[next](#) | [index](#)

Contents

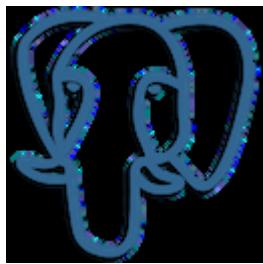
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

pgAdmin 4

Welcome to pgAdmin 4. pgAdmin is the leading Open Source management tool for Postgres, the world's most advanced Open Source database. pgAdmin 4 is designed to meet the needs of both novice and experienced Postgres users alike, providing a powerful graphical interface that simplifies the creation, maintenance and use of database objects.



- [Getting Started
 - \[Deployment\]\(#\)
 - \[Login Dialog\]\(#\)
 - \[User Management Dialog\]\(#\)
 - \[Change User Password Dialog\]\(#\)
 - \[User Interface\]\(#\)
 - \[Menu Bar\]\(#\)
 - \[Toolbar\]\(#\)
 - \[Tabbed Browser\]\(#\)
 - \[Tree Control\]\(#\)
 - \[Preferences Dialog\]\(#\)
 - \[Keyboard Shortcuts\]\(#\)](#)
- [Connecting To A Server
 - \[Server Group Dialog\]\(#\)
 - \[Server Dialog\]\(#\)
 - \[Master Password\]\(#\)
 - \[Connect to Server\]\(#\)
 - \[Connection Error\]\(#\)
 - \[Import/Export Servers\]\(#\)](#)
- [Managing Cluster Objects
 - \[Database Dialog\]\(#\)](#)



Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

This document in other formats: [PDF](#) | [ePub](#) | [Tarball](#)



[pgAdmin 4 4.19 documentation »](#)

[next](#) | [index](#)

[Online Demo](#)

Contents



[Getting Started](#)

[Connecting To A Server](#)

[Download](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[Management Basics](#)

[FAQ](#)

[Backup and Restore](#)

[Developer Tools](#)



[pgAgent](#)

[pgAdmin Project Contributions](#)

[Latest Docs](#)

[Release Notes](#)

[Licence](#)



pgAdmin 4 

[Ask a](#)

[Question](#)

Welcome to pgAdmin 4. pgAdmin is the leading Open
Source management tool for Postgres, the world's most



advanced Open Source database. pgAdmin 4 is
designed to meet the needs of both novice and

[Report a Bug](#)

experienced Postgres users alike, providing a powerful graphical interface that simplifies the creation, maintenance and use of



database objects.

[Screenshots](#)

[Getting Started](#)

[Deployment](#)

[Login Dialog](#)

[User Management Dialog](#)

[Change User Password Dialog](#)

[User Interface](#)

[Menu Bar](#)

[Toolbar](#)

[Tabbed Browser](#)

[Tree Control](#)

[Preferences Dialog](#)

[Keyboard Shortcuts](#)

[Connecting To A Server](#)

[Server Group Dialog](#)

[Server Dialog](#)

[Master Password](#)

[Connect to Server](#)

[Connection Error](#)

[Import/Export Servers](#)

[Managing Cluster Objects](#)

[Database Dialog](#)

[Move Objects Dialog](#)

[Resource Group Dialog](#)

[Login/Group Role Dialog](#)

[Tablespace Dialog](#)

[Managing Database Objects](#)

[Cast Dialog](#)

[Col ation Dialog](#)

[Domain Dialog](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)

[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Foreign Server Dialog](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)

[FTS Dictionary Dialog](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[Language Dialog](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Check Dialog](#)

[Column Dialog](#)

[Compound Trigger Dialog](#)

[Exclusion Constraint Dialog](#)

[Foreign key Dialog](#)

[Index Dialog](#)

[Primary key Dialog](#)

[Rule Dialog](#)

[Table Dialog](#)

[Trigger Dialog](#)

[Unique Constraint Dialog](#)

[Management Basics](#)

[Add Named Restore Point Dialog](#)

[Change Password Dialog](#)

[Grant Wizard](#)

[Import/Export Data Dialog](#)

[Maintenance Dialog](#)

[Backup and Restore](#)

[Backup Dialog](#)

[Backup Globals Dialog](#)

[Backup Server Dialog](#)

[Restore Dialog](#)

[Developer Tools](#)

[Debugger](#)

[Query Tool](#)

[View/Edit Data](#)

[Schema Diff](#)

[pgAgent](#)

[Using pgAgent](#)

[Installing pgAgent](#)

[Creating a pgAgent Job](#)

[pgAdmin Project Contributions](#)

[Submitting Patches](#)

[Code Overview](#)

[Coding Standards](#)

[Code Snippets](#)

[Code Review Notes](#)

[Translations](#)

[Release Notes](#)

[Version 4.19](#)

[Version 4.18](#)

[Version 4.17](#)

[Version 4.16](#)

[Version 4.15](#)

[Version 4.14](#)

[Version 4.13](#)

[Version 4.12](#)

[Version 4.11](#)

[Version 4.10](#)

[Version 4.9](#)

[Version 4.8](#)

[Version 4.7](#)

[Version 4.6](#)

[Version 4.5](#)

[Version 4.4](#)

[Version 4.3](#)

[Version 4.2](#)

[Version 4.1](#)

[Version 4.0](#)

[Version 3.6](#)

[Version 3.5](#)

[Version 3.4](#)

[Version 3.3](#)

[Version 3.2](#)

[Version 3.1](#)

[Version 3.0](#)

[Version 2.1](#)

[Version 2.0](#)

[Version 1.6](#)

[Version 1.5](#)

[Version 1.4](#)

[Version 1.3](#)

[Version 1.2](#)

[Version 1.1](#)

[Version 1.0](#)

[Licence](#)

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Management Basics](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
 - [Add Named Restore Point Dialog](#)
 - [Change Password Dialog](#)
 - [Grant Wizard](#)
 - [Import/Export Data Dialog](#)
 - [Maintenance Dialog](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Add Named Restore Point Dialog

Use the *Add named restore point* dialog to take a named snapshot of the state of the server for use in a recovery file. To create a named restore point, the server's `postgresql.conf` file must specify a `wal_level` value of `archive`, `hot_standby`, or `logical`. You must be a database superuser to create a restore point.

Restore point name

Enter the name of the restore point to add

Cancel OK

When the *Restore point name* window launches, use the field *Enter the name of the restore point to add* to provide a descriptive name for the restore point.

For more information about using a restore point as a recovery target, please see the [PostgreSQL documentation](#).

- Click the **OK** button to save the restore point.
- Click the **Cancel** button to exit without saving work.

Restore point name

Enter the name of the restore point to add

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Management Basics](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[FAQ](#)

[Management Basics](#)

Add Named Restore Point Dialog



[Change Password Dialog](#)

[Grant Wizard](#)

[Latest Docs](#)

[Import/Export Data Dialog](#)

[Maintenance Dialog](#)



[Backup and Restore](#)

[Developer Tools](#)

[Ask a](#)

[pgAgent](#)

[Question](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)



[Licence](#)

[Report a Bug](#)

Add Named Restore Point Dialog 



Use the *Add named restore point* dialog to take a named snapshot of the state of the server for use in a recovery file. To create a named restore

[Screenshots](#)

point, the server's postgresql.conf file must specify a *wal_level* value of *archive*, *hot_standby*, or *logical*. You must be a database superuser to create a restore point.

When the *Restore point name* window launches, use the field *Enter the name of the restore point to add* to provide a descriptive name for the restore point.

For more information about using a restore point as a recovery target, please see the [PostgreSQL documentation](#).

Click the *OK* button to save the restore point.

Click the *Cancel* button to exit without saving work.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Backup and Restore](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
 - [Backup Dialog](#)
 - [Backup Globals Dialog](#)
 - [Backup Server Dialog](#)
 - [Restore Dialog](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Backup Dialog

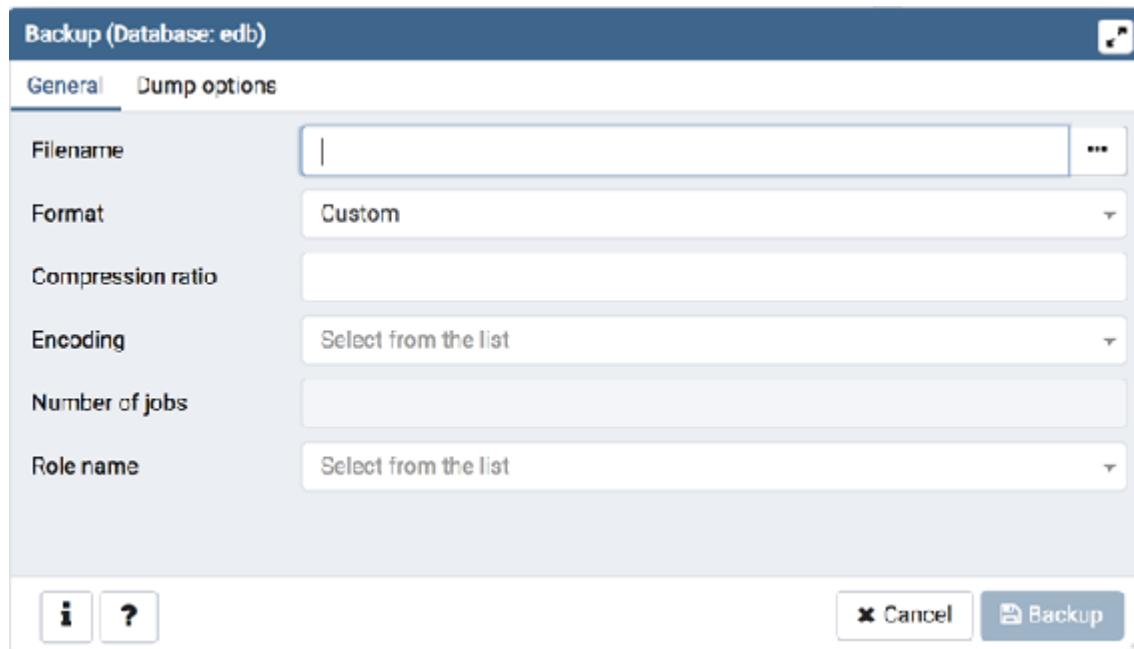
Using the `pg_dump` utility, pgAdmin provides an easy way to create a backup in a plain-text or archived format. You can then use a client application (like `psql` or the `Query Tool`) to restore a plain-text backup file, or use the Postgres `pg_restore` utility to restore an archived backup. The `pg_dump` utility must have read access to all database objects that you want to back up.

You can backup a single table, a schema, or a complete database. Select the name of the backup source in the pgAdmin tree control, right click to open the context menu, and select `Backup...` to open the `Backup` dialog. The name of the object selected will appear in the dialog title bar.

The screenshot shows the pgAdmin Backup dialog box. The title bar says "Backup (Database: edb)". The tabs at the top are "General" (selected) and "Dump options".
The "General" tab contains the following fields:

- Filename: An input field containing a placeholder for a file path.
- Format: A dropdown menu set to "Custom".
- Compression ratio: An input field for specifying compression levels.
- Encoding: A dropdown menu set to "Select from the list".
- Number of jobs: An input field for specifying the number of parallel jobs.
- Role name: A dropdown menu set to "Select from the list".

The "Dump options" tab is partially visible behind the "General" tab.
At the bottom of the dialog are three buttons: "Cancel" (with an X icon), "Backup" (with a disk icon), and "OK" (with a checkmark icon).



Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Backup and Restore](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[FAQ](#)

[Management Basics](#)

[Backup and Restore](#)



Backup Dialog

[Backup Globals Dialog](#)

[Latest Docs](#)

[Backup Server Dialog](#)

[Restore Dialog](#)



[Developer Tools](#)

[pgAgent](#)

[Ask a](#)

[pgAdmin Project Contributions](#)

[Question](#)

[Release Notes](#)

[Licence](#)



Backup Dialog

[Report a Bug](#)

Using the *pg_dump* utility, *pgAdmin* provides an easy way to create a



backup in a plain-text or archived format. You can then use a client application (like *psql* or the *Query Tool*) to restore a plain-text backup file,

[Screenshots](#)

or use the Postgres *pg_restore* utility to restore an archived backup. The *pg_dump* utility must have read access to all database objects that you want to back up.

You can backup a single table, a schema, or a complete database. Select the name of the backup source in the *pgAdmin* tree control, right click to open the context menu, and select *Backup...* to open the *Backup* dialog.

The name of the object selected will appear in the dialog title bar.

Use the fields in the *General* tab to specify parameters for the backup: Enter the name of the backup file in the *Filename* field. Optionally, select the *Browser* icon (...) to the right to navigate into a directory and select a file that will contain the archive.

Use the drop-down listbox in the *Format* field to select the format that is best suited for your application. Each format has advantages

and disadvantages:

Select *Custom* to create a custom archive file that you can use with *pg_restore* to create a copy of a database. Custom archive file formats must be restored with *pg_restore*. This format offers the opportunity to select which database objects to restore from the backup file. *Custom* archive format is recommended for medium to large databases as it is compressed by default.

Select *Tar* to generate a tar archive file that you can restore with *pg_restore*. The tar format does not support compression.

Select *Plain* to create a plain-text script file. A plain-text script file contains SQL statements and commands that you can execute at the *psql* command line to recreate the database objects and load the table data. A plain-text backup file can be edited in a text editor, if desired, before using the *psql* program to restore database objects. *Plain* format is normally recommended for smaller databases; script dumps are not recommended for blobs. The SQL commands within the script will reconstruct the database to the last saved state of the database. A plain-text script can be used to reconstruct the database on another machine, or (with modifications) on other architectures.

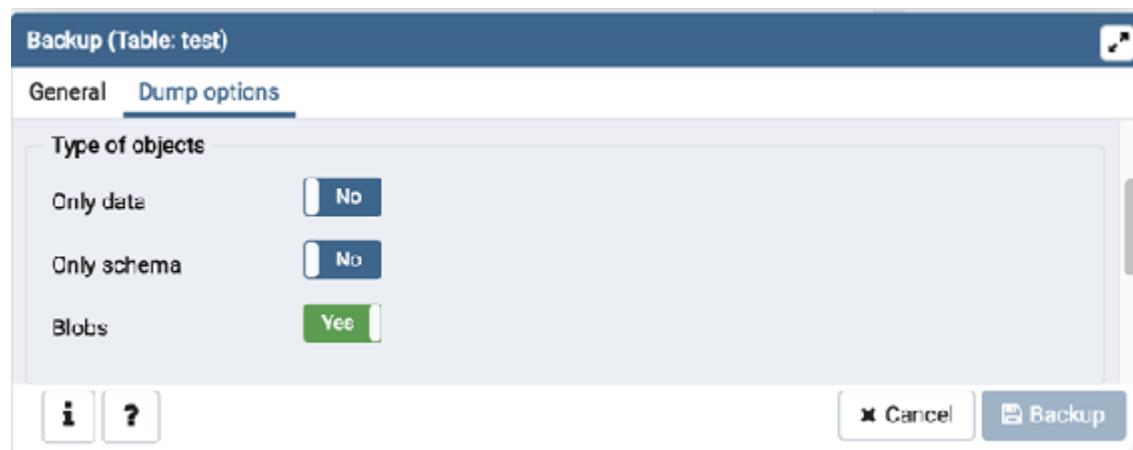
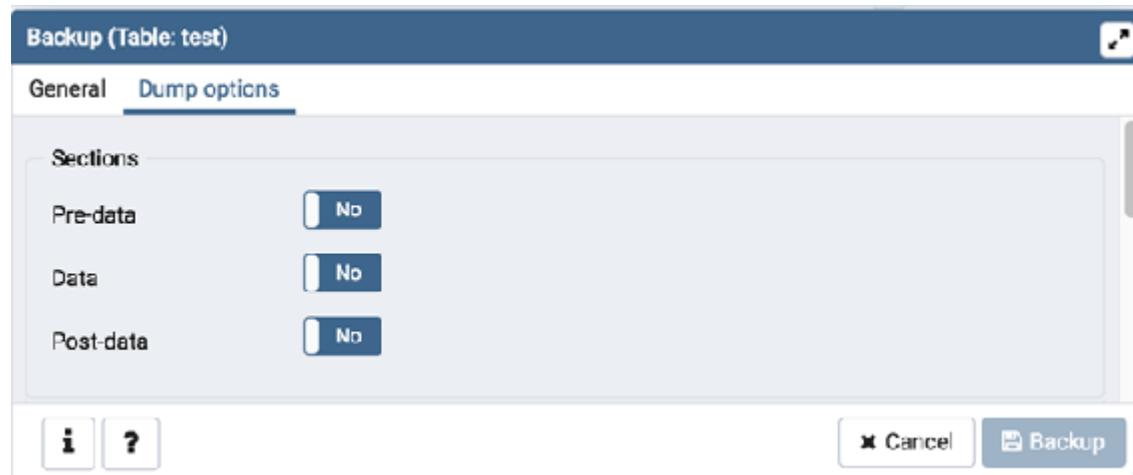
Select *Directory* to generate a directory-format archive suitable for use with *pg_restore*. This file format creates a directory with one file for each table and blob being dumped, plus a *Table of Contents* file describing the dumped objects in a machine-readable format that *pg_restore* can read. This format is compressed by default.

Use the *Compression Ratio* field to select a compression level for the backup. Specify a value of zero to mean use no compression; specify a maximum compression value of 9. Please note that tar archives do not support compression.

Use the *Encoding* drop-down listbox to select the character encoding method that should be used for the archive.

Use the *Number of Jobs* field (when applicable) to specify the number of tables that will be dumped simultaneously in a parallel backup.

Use the dropdown listbox next to *Rolename* to specify the role that owns the backup.



Click the *Dump options* tab to continue. Use the box fields in the *Dump options* tab to provide options for *pg_dump*.

Move switches in the Sections field box to select a portion of the object that will be backed up.

Move the switch next to *Pre-data* to the Yes position to include all data definition items not included in the data or post-data item lists.

Move the switch next to *Data* to the Yes position to backup actual table data, large-object contents, and sequence values.

Move the switch next to *Post-data* to the Yes position to include definitions of indexes, triggers, rules, and constraints other than validated check constraints.

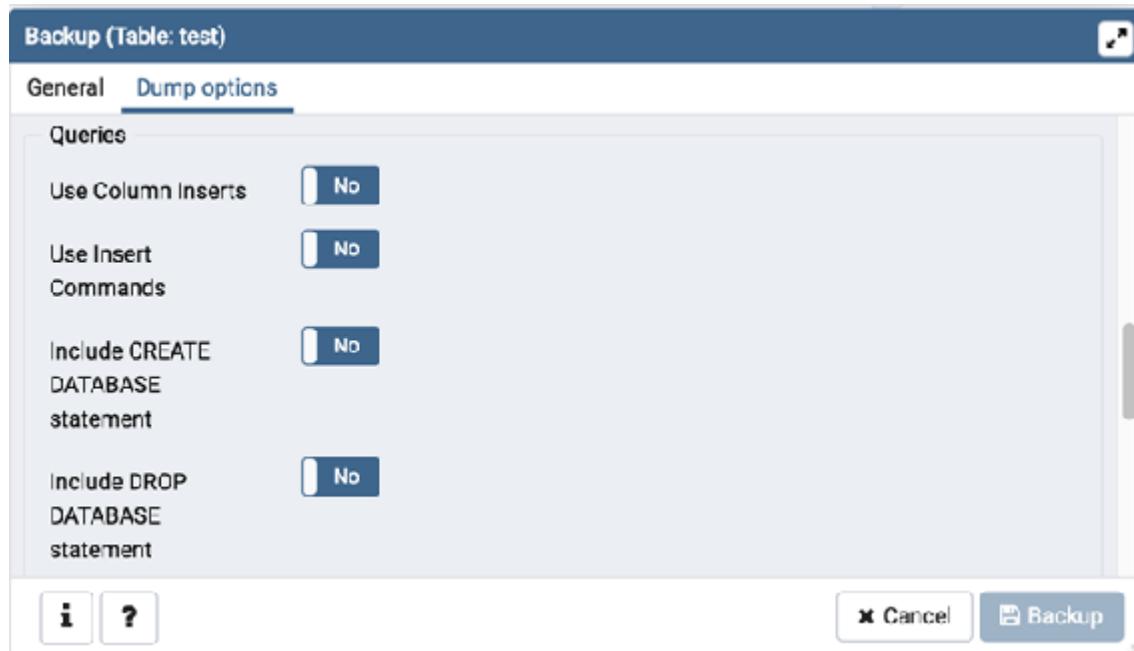
Move switches in the Type of objects field box to specify details about the type of objects that wil be backed up.

Move the switch next to *Only data* to the Yes position to limit the back up to data.

Move the switch next to *Only schema* to limit the back up to schema-level database objects.

Move the switch next to *Blobs* to the No position to exclude large objects in the backup.





Move switches in the Do not save field box to select the objects that will not be included in the backup.

Move the switch next to *Owner* to the Yes position to exclude commands that set object ownership.

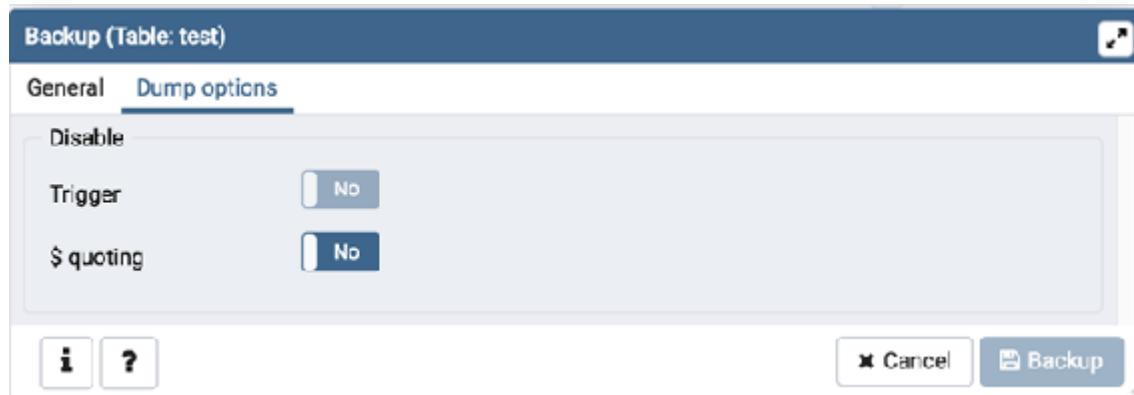
Move the switch next to *Privilege* to the Yes position to exclude commands that create access privileges.

Move the switch next to *Tablespace* to the Yes position to exclude tablespaces.

Move the switch next to *Unlogged table data* to the Yes position to exclude the contents of unlogged tables.

Move the switch next to *Comments* to the Yes position to exclude commands that set the comments. Note: This option is visible only for database server greater than or equal to 11.

Move switches in the Queries field box to specify the type of statements that should be included in the backup.



Move the switch next to *Use Column Inserts* to the Yes position to dump the data in the form of INSERT statements and include explicit column names. Please note: this may make restoration from backup slow.

Move the switch next to *Use Insert commands* to the Yes position to dump the data in the form of INSERT statements rather than using a COPY command. Please note: this may make restoration from backup slow.

Move the switch next to *Include CREATE DATABASE statement* to the Yes position to include a command in the backup that creates a new database when restoring the backup.

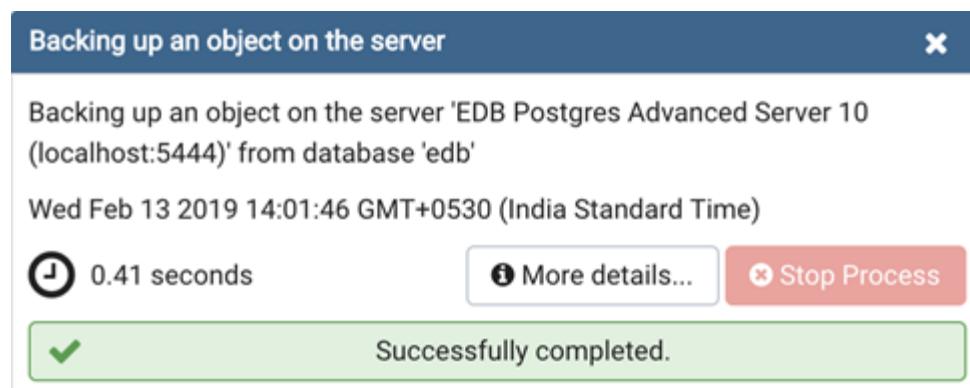
Move the switch next to *Include DROP DATABASE statement* to the Yes position to include a command in the backup that will drop any existing database object with the same name before recreating the object during a backup.

Move the switch next to *Load Via Partition Root* to the Yes position, so when dumping a COPY or INSERT statement for a partitioned table, target the root of the partitioning hierarchy which contains it rather than the partition itself. Note: This option is visible only for database server greater than or equal to 11.

Move switches in the Disable field box to specify the type of statements that should be excluded from the backup.

Move the switch next to *Trigger* (active when creating a data-only backup) to the Yes position to include commands that will disable triggers on the target table while the data is being loaded.

Move the switch next to *\$ quoting* to the Yes position to enable dollar quoting within function bodies; if disabled, the function body will be quoted using SQL standard string syntax.



Move switches in the Miscellaneous field box to specify miscellaneous backup options.

Move the switch next to *With OIDs* to the Yes position to include object identifiers as part of the table data for each table.

Move the switch next to *Verbose messages* to the *No* position to instruct *pg_dump* to exclude verbose messages.

Move the switch next to *Force double quotes on identifiers* to the *Yes* position to force the quoting of all identifiers.

Move the switch next to *Use SET SESSION AUTHORIZATION* to the *Yes* position to include a statement that will use a *SET SESSION AUTHORIZATION* command to determine object ownership (instead of an *ALTER OWNER* command).

When you've specified the details that will be incorporated into the *pg_dump* command:

Click the *Backup* button to build and execute a command that builds a backup based on your selections on the *Backup* dialog.

Click the *Cancel* button to exit without saving work.

Use the Stop Process button to stop the Backup process.

If the backup is successful, a popup window will confirm success. Click *Click here for details* on the popup window to launch the *Process Watcher*. The *Process Watcher* logs all the activity associated with the backup and provides additional information for troubleshooting.

Process Watcher - Backing up an object on the server

Backing up an object on the server 'EDB Postgres Advanced Server 10 (localhost:5444)' from database 'edb'...

Running command:

```
/opt/edb/as10/bin/pg_dump --file '/root/file' --host 'localhost' --port '5444' --username 'enterprisedb' --no-password --verbose --format=c --blobs "edb"
```

⌚ Start time: Wed Feb 13 2019 14:05:28 GMT+0530 (India Standard Time) ⌚ Stop Process

```
pg_dump: last built-in OID is 16383
pg_dump: reading extensions
pg_dump: identifying extension members
pg_dump: reading schemas
pg_dump: reading user-defined tables
pg_dump: reading packages variables
pg_dump: reading user-defined packages
pg_dump: reading user-defined synonyms
pg_dump: reading user-defined functions
pg_dump: reading user-defined types
pg_dump: finding typelem dependencies for user-defined types
pg_dump: reading procedural languages
pg_dump: reading user-defined aggregate functions
pg_dump: reading user-defined operators
```

✓ Successfully completed. Execution time: 0.17 seconds

If the backup is unsuccessful, you can review the error messages returned by the backup command on the *Process Watcher*.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Backup and Restore](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
 - [Backup Dialog](#)
 - [Backup Globals Dialog](#)
 - [Backup Server Dialog](#)
 - [Restore Dialog](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Backup Globals Dialog

Use the *Backup Globals* dialog to create a plain-text script that recreates all of the database objects within a cluster, and the global objects that are shared by those databases. Global objects include tablespaces, roles, and object properties. You can use the pgAdmin Query Tool to play back a plain-text script, and recreate the objects in the backup.



Use the fields in the General tab to specify the following:

- Enter the name of the backup file in the *Filename* field. Optionally, select the Browser icon (ellipsis) to the right to navigate into a directory and select a file that will contain the archive.



Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Backup and Restore](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[FAQ](#)

[Management Basics](#)

[Backup and Restore](#)



[Backup Dialog](#)

Backup Globals Dialog

[Latest Docs](#)

[Backup Server Dialog](#)

[Restore Dialog](#)



[Developer Tools](#)

[pgAgent](#)

[Ask a](#)

[pgAdmin Project Contributions](#)

[Question](#)

[Release Notes](#)

[Licence](#)



Backup Globals Dialog 

[Report a Bug](#)

Use the *Backup Globals* dialog to create a plain-text script that recreates

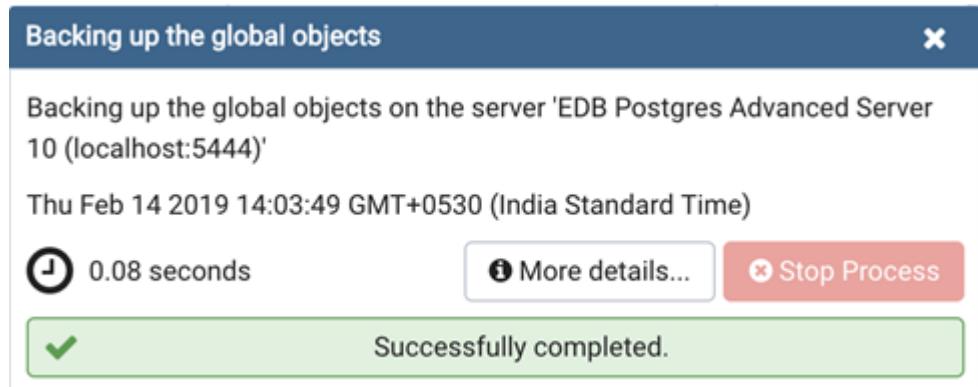


all of the database objects within a cluster, and the global objects that are shared by those databases. Global objects include tablespaces, roles,

[Screenshots](#)

and object properties. You can use the pgAdmin *Query Tool* to play back a plain-text script, and recreate the objects in the backup.

Use the fields in the *General* tab to specify the following: Enter the name of the backup file in the *Filename* field. Optionally, select the *Browser* icon (ellipsis) to the right to navigate into a directory and select a file that will contain the archive.



Process Watcher - Backing up the global objects

```
Backing up the global objects on the server 'EDB Postgres Advanced Server 10 (localhost:5444)'...
Running command:
/opt/edb/as10/bin/pg_dumpall --file "/home/File2" --host "localhost" --port "5444" --username "enterprisedb" --no-password --database "postgres" --globals-only --verbose --role "enterprisedb"

⌚ Start time: Thu Feb 14 2019 14:03:49 GMT+0530 (India Standard Time) ⌚ Stop Process
```

```
pg_authid.on ug.old = a.grantor WHERE NOT (ur rolname ~ 'pg_') AND um.rolname ~ 'pg_') ORDER BY 1,2,3
pg_dumpall: executing SELECT passhistroeid, r.rolname as rolename, passhlist,password, passhlistpasswordsetat FROM
edb_password_history eph, pg_authid r WHERE eph.passhistroeid = r.old AND (r.rolpasswordsetat IS NULL OR
eph.passhlistpasswordsetat != r.rolpasswordsetat) ORDER BY passhistroeid, passhlistpasswordsetat
pg_dumpall: executing SELECT oid, spcname, pg_catalog.pg_get_userbyid(spcowner) AS spcowner,
pg_catalog.pg_tablespace_location(oid), (SELECT pg_catalog.array_agg(acl) FROM (SELECT
pg_catalog.unnest(coalesce(spcacl,pg_catalog.acidefault('t',spcowner))) AS acl EXCEPT SELECT
pg_catalog.unnest(pg_catalog.acidefault('t',spcowner))) AS fco)AS spcacl,(SELECT pg_catalog.array_agg(ecl) FROM (SELECT
pg_catalog.unnest(pg_catalog.acidefault('t',spcowner)) AS acl EXCEPT SELECT
pg_catalog.unnest(coalesce(spcacl,pg_catalog.acidefault('t',spcowner)))) AS fco)AS rspcacl,array_to_string(epcoptions, ',')
'),pg_catalog.shobj_description(oid, 'pg_tablespace') FROM pg_catalog.pg_tablespace WHERE spcname !~ 'pg_' ORDER BY 1
pg_dumpall: executing SELECT rgrpname, rgrppcpuarealimit, rgrpdirtyratelimit FROM edb_resource_group ORDER BY 1
pg_dumpall: executing SELECT rolname, datname, unnest(setconfig) FRCM pg_db_role_setting, pg_authid u, pg_database
WHERE setrole = u.old AND setdatabase = pg_database.old
```

⌚ Successfully completed. Execution time: 0.08 seconds

Use the drop-down listbox next to *Role name* to specify a role with connection privileges on the selected server. The role will be used for authentication during the backup.

Move switches in the Miscellaneous field box to specify the type of statements that should be included in the backup.

Move the *Verbose messages* switch to the *No* position to exclude status messages from the backup. The default is *Yes*.

Move the *Force double quote on identifiers* switch to the *Yes* position to name identifiers without changing case. The default is *No*.

Click the *Backup* button to build and execute a command based on your selections; click the *Cancel* button to exit without saving work.

Use the Stop Process button to stop the Backup process.

If the backup is successful, a popup window will confirm success.

Click *Click here for details* on the popup window to launch the *Process Watcher*. The *Process Watcher* logs all the activity associated with the backup and provides additional information for troubleshooting.

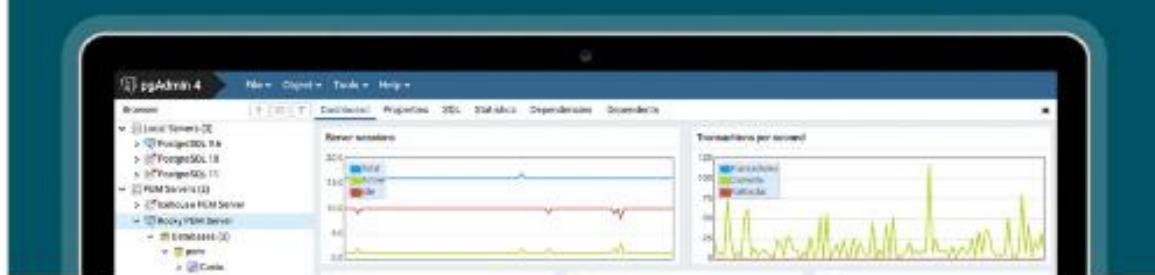
If the backup is unsuccessful, review the error message returned by the *Process Watcher* to resolve any issue.

pgAdmin

PostgreSQL Tools

pgAdmin is the most popular and feature rich Open Source administration and development platform for PostgreSQL, the most advanced Open Source database in the world.

pgAdmin may be used on Linux, Unix, Mac OS X and Windows to manage PostgreSQL 9.2 and above.



Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

News

2020-03-05 - pgAdmin 4 v4.19 Released

The pgAdmin Development Team are pleased to announce pgAdmin 4 version 4.19. This release of pgAdmin 4 includes over 18 bug fixes and new features. For more details please see the release notes [here](#).

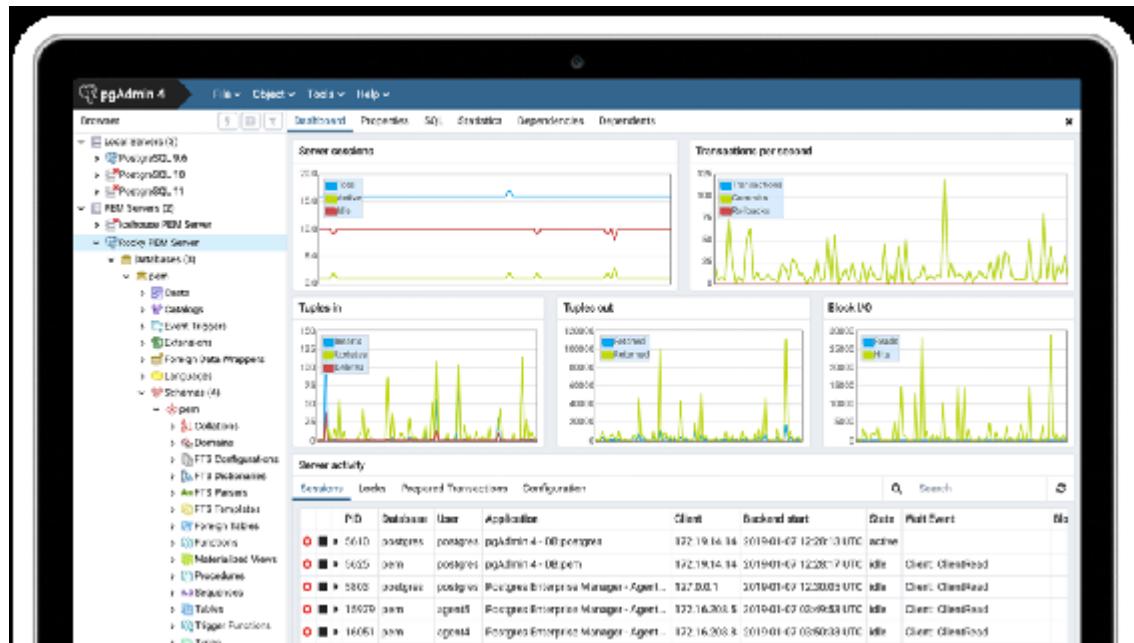
Notable changes in this release include:

- Added Python 3.8 support.
- Added accessibility support in AlertifyJS.
- Added Czech language support.
- Improvements in the UI for both default and dark themes.
- Fix an issue where debugger not showing all arguments anymore after hitting SQL error while debugging.
- Ensure that all the transactions should be canceled before closing the connections when a server is disconnected using pgAdmin.
- Fixed Firefox monospaced issue by updating the font to the latest version.
- Improve the style of the highlighted code after query execution for Dark mode.
- Changed background pattern for geometry viewer to use #fff for all themes.

Download your copy [here](#).

2020-02-06 - pgAdmin 4 v4.18 Released

The pgAdmin Development Team are pleased to announce pgAdmin 4 version 4.18. This release of pgAdmin 4 includes over 23 bug fixes and new features. For more details please see the release notes [here](#).



pgAdmin

PostgreSQL Tools

pgAdmin is the most popular and feature rich Open Source administration and development platform for PostgreSQL, the most advanced Open Source database in the world.

pgAdmin may be used on Linux, Unix, Mac OS X and Windows to manage PostgreSQL 9.2

and above.

Quick Links

News



2020-03-05 - pgAdmin 4 v4.19 Released

[Online Demo](#)

The pgAdmin Development Team are pleased to announce pgAdmin 4

version 4.19. This release of pgAdmin 4 includes over 18 bug fixes and new features. For more details please see the release notes [here](#).



Notable changes in this release include:

[Download](#)

Added Python 3.8 support.



Added accessibility support in AlertifyJS.

Added Czech language support.

[FAQ](#)

Improvements in the UI for both default and dark themes.

Fix an issue where debugger not showing al arguments anymore after



hitting SQL error while debugging.

Ensure that al the transactions should be canceled before closing

[Latest Docs](#)

the connections when a server is disconnected using pgAdmin.

Fixed Firefox monospaced issue by updating the font to the latest



version.

[Ask a](#)

Improve the style of the highlighted code after query execution for Dark mode.

[Question](#)

Changed background pattern for geometry viewer to use #fff for all themes.



Download your copy [here](#).

[Report a Bug](#)

2020-02-06 - pgAdmin 4 v4.18 Released



The pgAdmin Development Team are pleased to announce pgAdmin 4

[Screenshots](#)

version 4.18. This release of pgAdmin 4 includes over 23 bug fixes and new features. For more details please see the release notes [here](#).

Notable changes in this release include: Added a Schema Diff tool to compare two schemas and generate a diff script.

Added support for a multi-level partitioned table.

Added labels and titles after parsing and validating all the pgAdmin4

web pages for accessibility.

Set input controls as read-only instead of disabled will allow tab navigation in the properties tab and also allow screen readers to read it.

Fix an issue where select, insert and update scripts on tables throwing an error.

Logout the pgAdmin session when no user activity of mouse move, click or keypress.

Fixed an issue where Grant wizard unable to handle multiple objects when the query string parameter exceeds its limit.

Ensure that path file name should not disappear when changing ext from the dropdown in file explorer dialog.

Fix an issue where setting STORAGE_DIR to empty should show all the volumes on Windows in server mode.

Download your copy [here](#).

2020-01-09 - pgAdmin 4 v4.17 Released

The pgAdmin Development Team are pleased to announce pgAdmin 4

version 4.17. This release of pgAdmin 4 includes over 20 bug fixes and new features. For more details please see the release notes [here](#).

Notable changes in this release include:

Allow screen-reader to read relationship attributes in nested elements.

Ensure all binaries are securely signed and linked with the hardened runtime in the macOS bundle.

Added support to view multilevel partitioned tables.

Rename some internal environment variables that could conflict with Kubernetes.

Fix an issue where operator, access method and operator class is not visible for exclusion constraints.

Ensure that constraints, indexes, rules, triggers, and compound triggers should be created on partitions.

Fix syntax highlighting in code mirror for backslash and escape constant.

Ensure that the user should be able to select/modify tablespace for the partitioned table on v12 and above.

Added support for on-demand loading of items in Select2

Download your copy [here](#).

2019-12-12 - pgAdmin 4 v4.16 Released

The pgAdmin Development Team are pleased to announce pgAdmin 4

version 4.16. This release of pgAdmin 4 includes over 43 bug fixes and new features. For more details please see the release notes [here](#).

Warning: This release includes a change to the container distribution to run pgAdmin as a non-root user. Those users of the container who are running with mapped storage directories may need to change the ownership on the host machine, for example: sudo chown -R 5050:5050 <host_directory> Notable changes in this release include:

Run pgAdmin in the container as a non-root user.

Added aria-label to provide an invisible label where a visible label cannot be used.

Added role="status" attribute to all the status messages for accessibility.

Use a 'play' icon for the Execute Query button in the Query Tool for greater consistency with other applications.

Allow Gunicorn logs in the container to be directed to a file specified through GUNICORN_ACCESS_LOGFILE.

Fix network disconnect issue while establishing the connection via SSH

Tunnel and it impossible to expand the Servers node.

Ensure that the superuser should be able to create role, as the superuser overrides all the access restrictions.

Enhance the logic to change the label from 'Delete/Drop' to 'Remove' for the server and server group node.

Fix VPN network disconnect issue where pgAdmin4 hangs on expanding the Servers node.

Ensure that the Servers collection node should expand independently of server connections.

Fix pgAdmin4 failed to start issue after upgrading to version 4.15.

Download your copy [here](#).

2019-11-21 - Try pgAdmin online!

Our friends at [EnterpriseDB](#) have kindly made an online demo environment for pgAdmin available to allow new users to try it out in their browser without needing to download or install any software. The environment (based on the

[Katacoda](#) learning platform) includes pgAdmin and PostgreSQL 12, preloaded with the pagila sample database, and is unique to every user al owing a far better experience than typical shared demo environments.

Click [here](#) to try it out now!

2019-11-14 - pgAdmin 4 v4.15 Released

The pgAdmin Development Team are pleased to announce pgAdmin 4

version 4.15. This release of pgAdmin 4 includes over 33 bug fixes and new features. For more details please see the release notes [here](#).

Notable changes in this release include:

Added Dark(Beta) UI Theme option.

Added support for custom theme creation and selection.

Added encrypted password in reverse engineered SQL for roles.

Support Enable Always and Enable Replica on triggers.

Add an option to request confirmation before cancel ing/resetting changes on a Properties dialog.

Rename the context menu from 'Drop Server' to 'Remove Server'.

Fixed issue where Drop and Disconnect connection menu points are too close to each other.

Ensure al messages are retained in the Query Tool from long running queries.

Ensure compression level is passed to pg_dump when backing up in directory format.

Ensure the number of jobs can be specified when backing up in directory format.

Download your copy [here](#).

2018-07-12 - pgAgent v4.0.0 Released

The pgAdmin Development Team are pleased to announce the release of pgAgent v4.0.0.

pgAgent is a job scheduler for PostgreSQL; for more information please see the documentation included as part of the pgAdmin documentation at

<https://www.pgadmin.org/docs/pgadmin4/3.x/pgagent.html>.

[Download \(source\): https://www.pgadmin.org/download/pgagent-source-](#)

[code/](#)

We expect DEB and RPM packages to be available from the PostgreSQL

APT/YUM repositories in due course, as well as an updated installer from EnterpriseDB available through StackBuilder.

Notable changes in this release include:

Remove the dependency on wxWidgets. pgAgent now uses Boost for thread management and synchronisation [Neel Patel]

Refactor connection parsing logic to fix issues with and improve handling of connection strings [Thomas Krennwalner]

Handle 2 digit server version numbers [Ashesh Vashi]

Fix handling of the "succeeded" flag on job steps which could prevent failure of steps causing the following steps to fail [Sanket

Mehta]

The current version of pgAdmin 4 is [4.19](#)

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 4.19 documentation](#) »

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
 - [Using pgAgent](#)
 - [Installing pgAgent](#)
 - [Creating a pgAgent Job](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

pgAgent

pgAgent is a job scheduling agent for Postgres databases, capable of running multi-step batch or shell scripts and SQL tasks on complex schedules.

pgAgent is distributed independently of pgAdmin. You can download pgAgent from the [download area](#) of the pgAdmin website.

- [Using pgAgent](#)
 - [Security Concerns](#)
- [Installing pgAgent](#)
 - [Database setup](#)
 - [Daemon installation on Unix](#)
 - [Service installation on Windows](#)
- [Creating a pgAgent Job](#)

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#)
[Development](#)

[pgAdmin 4 4.19 documentation](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[FAQ](#)

[Management Basics](#)

[Backup and Restore](#)

Developer Tools



pgAgent

Latest Docs

[Using pgAgent](#)

[Installing pgAgent](#)



[Creating a pgAgent Job](#)

[pgAdmin Project Contributions](#)

[Ask a](#)

[Release Notes](#)

[Question](#)

[Licence](#)



pgAgent

[Report a Bug](#)

pgAgent is a job scheduling agent for Postgres databases, capable of running multi-step batch or shell scripts and SQL tasks on complex



schedules.

[Screenshots](#)

pgAgent is distributed independently of pgAdmin. You can download pgAgent from the [download area](#) of the pgAdmin website.

[Using pgAgent](#)

[Security Concerns](#)

[Installing pgAgent](#)

[Database setup](#)

[Daemon installation on Unix](#)

[Service installation on Windows](#)

[Creating a pgAgent Job](#)

Browser Details Properties SQL Statistics Dependencies Dependents x

> Servers

Welcome

 pgAdmin
Management Tools for PostgreSQL

Feature rich | Maximises PostgreSQL | Open Source

pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.

Quick Links

 Add New Server  Configure pgAdmin

Getting Started

 PostgreSQL Documentation  pgAdmin Website  Planet PostgreSQL  Community Support

Browser

[Dashboard](#) [Properties](#) [SQL](#) [Statistics](#) [Dependencies](#) [Dependents](#)



Servers

Welcome

Feature rich | Maximises PostgreSQL | Open Source pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.

Quick Links



Add New Server

Configure pgAdmin

Getting Started



[PostgreSQL](#)

[pgAdmin Website](#)

[Planet PostgreSQL](#)

[Community Support](#)

[Documentation](#)

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Managing Cluster Objects](#) > [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Tablespace Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
 - [Database Dialog](#)
 - [Move Objects Dialog](#)
 - [Resource Group Dialog](#)
 - [Login/Group/Role Dialog](#)
 - [Tablespace Dialog](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Tablespace Dialog

Use The Tablespace dialog to define a tablespace. A tablespace allows superusers to define an alternative location on the file system where the data files containing database objects (such as tables and indexes) reside. Tablespaces are only supported on systems that support symbolic links. Note that a tablespace cannot be used independently of the cluster in which it is defined.

The Tablespace dialog organizes the definition of a tablespace through the following tabs: *General*, *Definition*, *Parameters*, and *Security*. The SQL tab displays the SQL code generated by dialog selections.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#)
[Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Cluster Objects](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Tablespace Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Database Dialog](#)

[Move Objects Dialog](#)



[Resource Group Dialog](#)

[Login/Group Role Dialog](#)

[Latest Docs](#)

Tablespace Dialog

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[Ask a](#)

[Management Basics](#)

[Question](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)



[pgAdmin Project Contributions](#)

[Report a Bug](#)

[Release Notes](#)

[Licence](#)

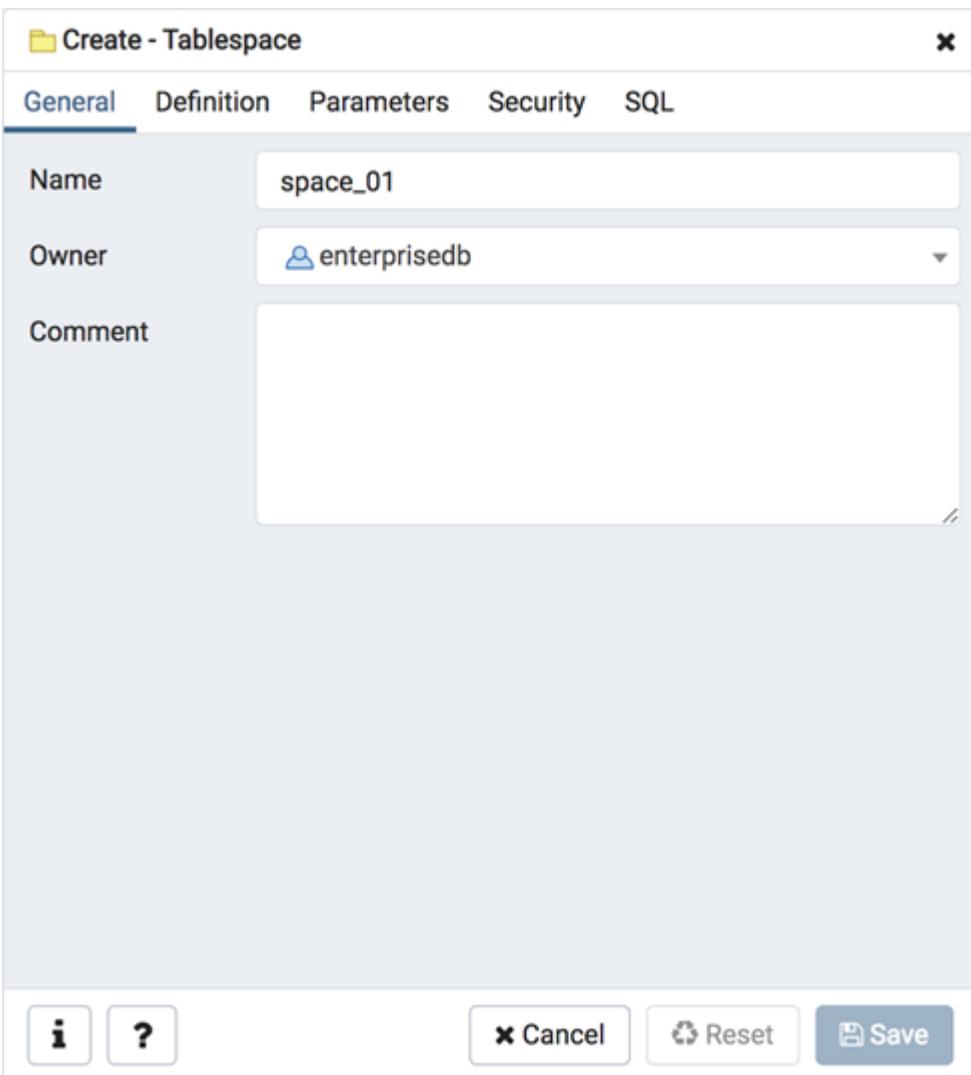


Tablespace Dialog

Screenshots

Use The *Tablespace* dialog to define a tablespace. A tablespace allows superusers to define an alternative location on the file system where the data files containing database objects (such as tables and indexes) reside. Tablespaces are only supported on systems that support symbolic links. Note that a tablespace cannot be used independently of the cluster in which it is defined.

The *Tablespace* dialog organizes the definition of a tablespace through the following tabs: *General*, *Definition*, *Parameters*, and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

A screenshot of a "Create - Tablespace" dialog box. The title bar says "Create - Tablespace". Below it is a navigation bar with tabs: General (which is selected), Definition, Parameters, Security, and SQL. The main area contains three input fields: "Name" with the value "space_01", "Owner" with the value "enterprisedb", and a "Comment" field which is empty. At the bottom are buttons for "Cancel", "Reset", and "Save".

Create - Tablespace

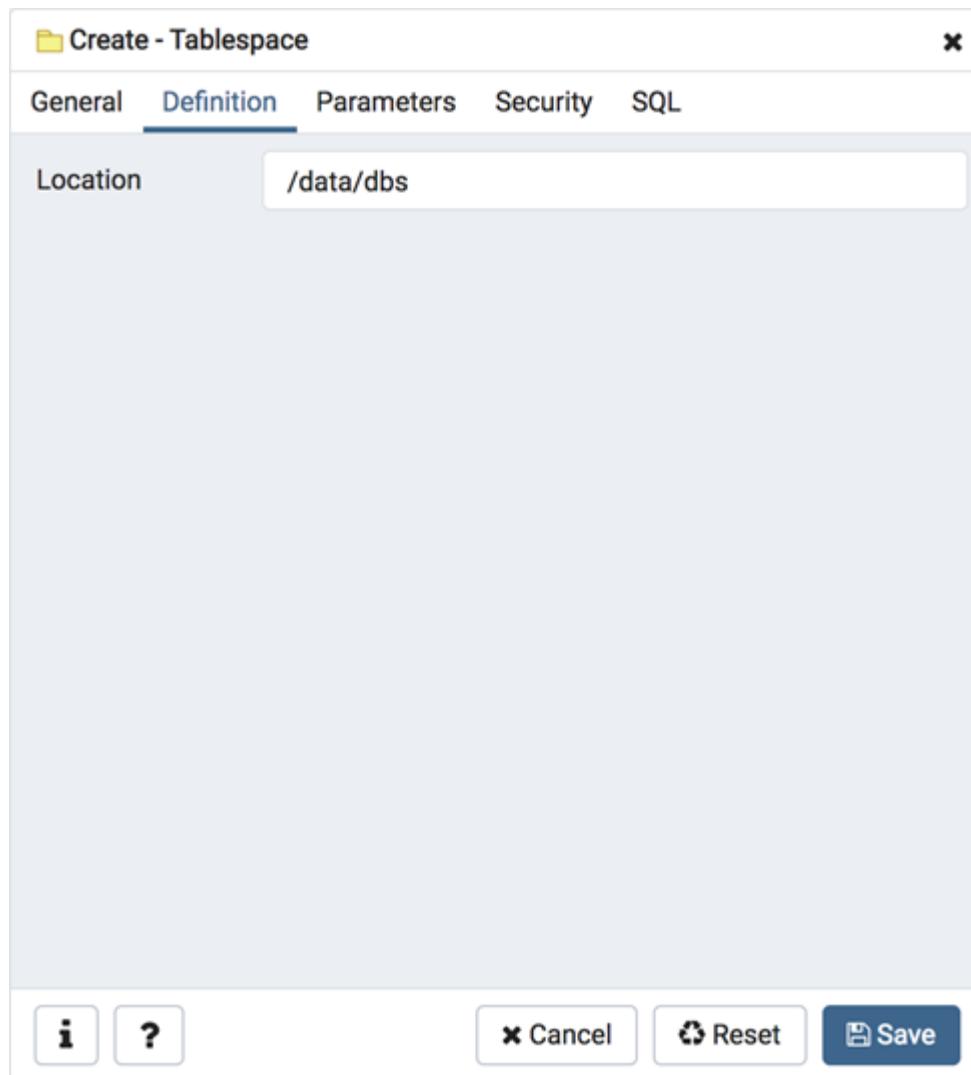
General Definition Parameters Security SQL

Name: space_01

Owner: enterprisedb

Comment:

i **?** **Cancel** **Reset** **Save**



Use the *Name* field to identify the tablespace with a descriptive name. The name cannot begin with pg_ ; these names are reserved for system tablespaces.

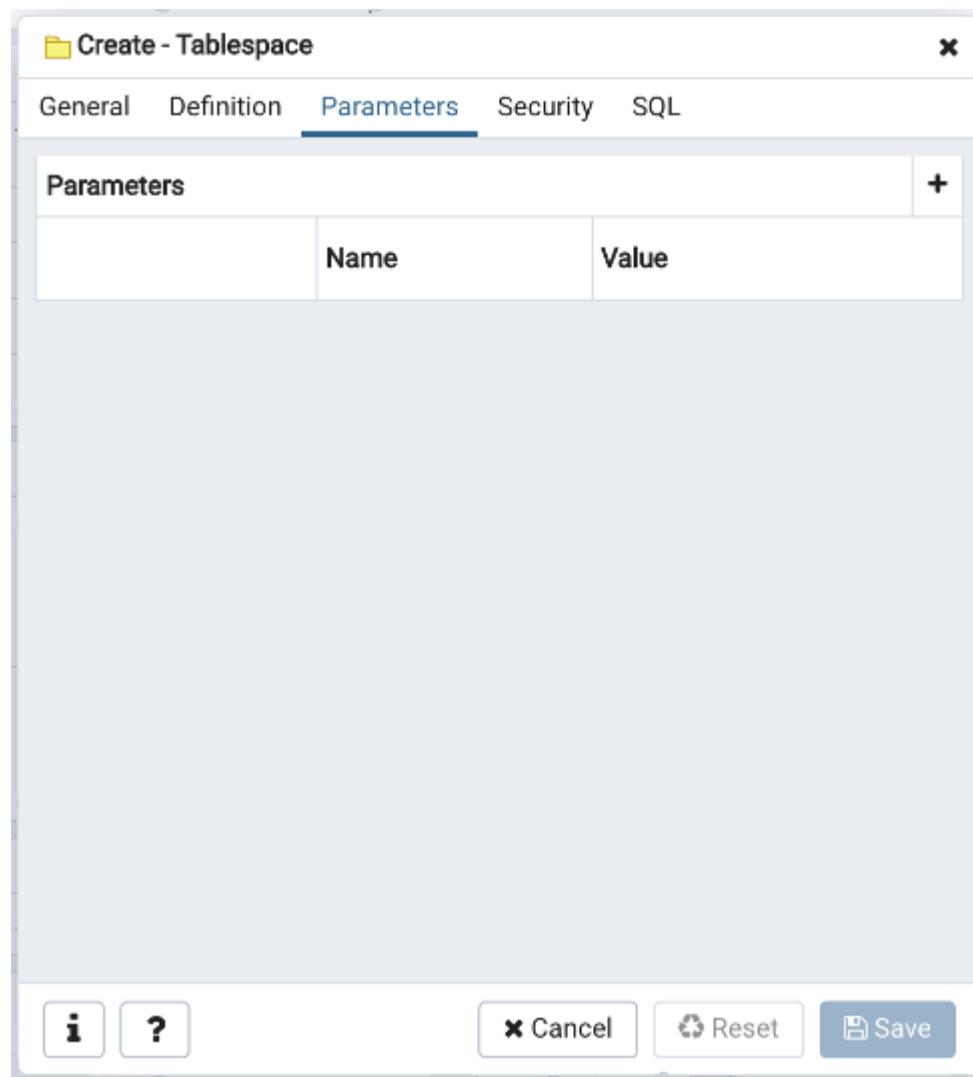
Select the owner of the tablespace from the drop-down listbox in the *Owner* field.

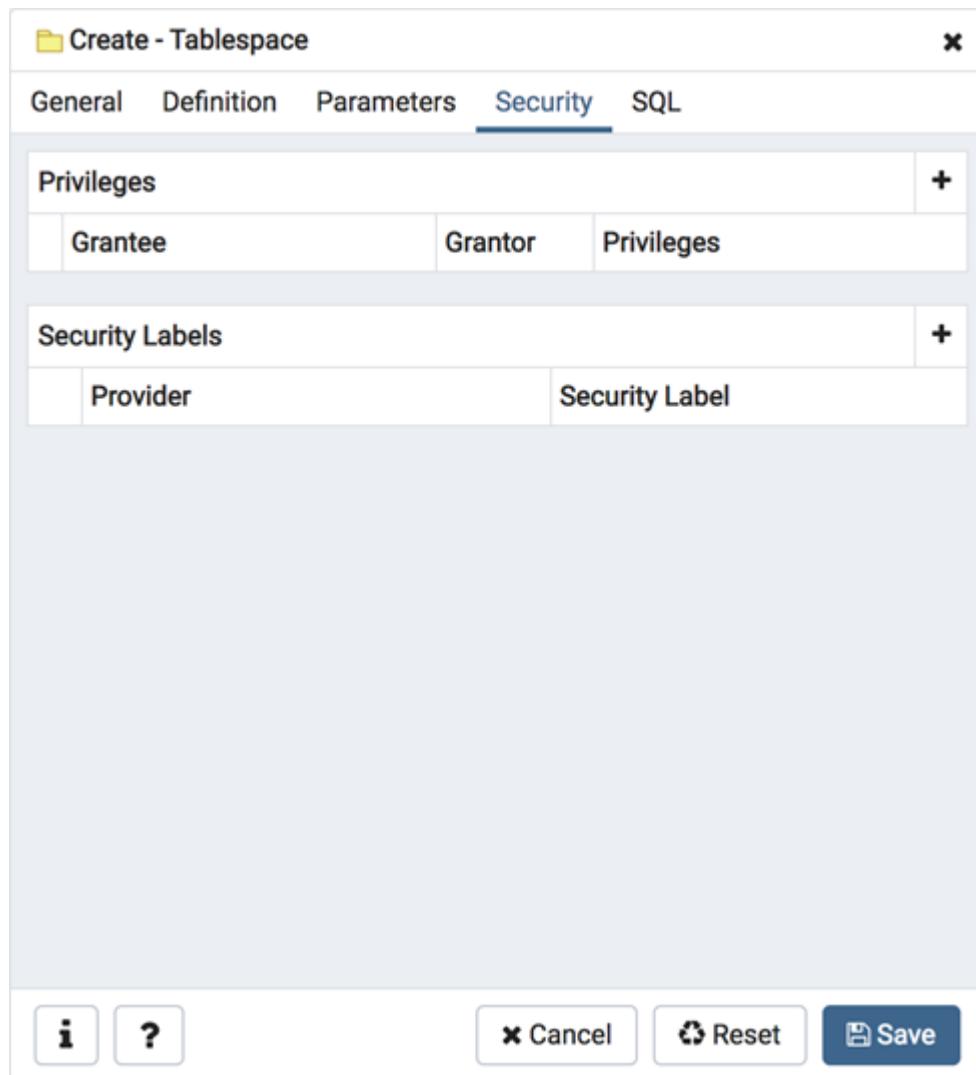
Store notes about the tablespace in the *Comment* field.

Click the *Definition* tab to continue.

Use the *Location* field to specify an absolute path to a directory that will contain the tablespace.

Click the *Parameters* tab to continue.





Use the *Parameters* tab to set parameters for the tablespace. Click the *Add* icon (+) to add a row to the table below.

Use the drop-down listbox next to *Name* to select a parameter.

Use the *Value* field to set a value for the parameter.

Click the *Add* icon (+) to specify each additional parameter; to discard a parameter, click the trash icon to the left of the row and confirm deletion in the *Delete Row* dialog.

Click the *Security* tab to continue.

Use the *Security* tab to assign privileges and define security labels for the tablespace.

Use the *Privileges* panel to assign security privileges. Click the *Add* icon (+) to assign a set of privileges:

Select the name of the role from the drop-down listbox in the *Grantee* field.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.

Click inside the *Privileges* field. Check the boxes to the left of one or more privileges to grant the selected privileges to the specified user.

Click the *Add* icon to assign additional sets of privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Use the *Security Labels* panel to define security labels applied to the tablespace. Click the *Add* icon (+) to add each security label selection: Specify a security label provider in the *Provider* field. The named provider must be loaded and must consent to the proposed labeling operation.

Specify a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL

places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

To discard a security label, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

Your entries in the *Tablespace* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or

switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Tablespace* dialog:



The screenshot shows the 'Create - Tablespace' dialog box. The 'SQL' tab is selected, displaying the following SQL commands:

```
1 CREATE TABLESPACE space_01
2   OWNER enterprisedb
3   LOCATION '/data/dbs';
4
5 ALTER TABLESPACE space_01
6   OWNER TO enterprisedb;
7
8 ALTER TABLESPACE space_01
9
10  SET (random_page_cost=1);
```

At the bottom of the dialog, there are buttons for 'Info' (i), 'Cancel' (x), 'Reset' (refresh), and 'Save' (blue button).

The example shown demonstrates creating a tablespace named *space_01*. It has a *random_page_cost* value equal to 1.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | Development

[pgAdmin 4 4.19 documentation](#) > [Getting Started](#) > [Documentation](#) > previous | next | index

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
 - [Deployment](#)
 - [Login Dialog](#)
 - [User Management Dialog](#)
 - [Change User Password Dialog](#)
 - [User Interface](#)
 - [Menu Bar](#)
 - [Toolbar](#)
 - [Tabbed Browser](#)
 - [Tree Control](#)
 - [Preferences Dialog](#)
 - [Keyboard Shortcuts](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

The config.py File

There are multiple configuration files that are read at startup by pgAdmin. These are as follows:

- `config.py`: This is the main configuration file, and should not be modified. It can be used as a reference for configuration settings, that may be overridden in one of the following files.
- `config_distro.py`: This file is read after `config.py` and is intended for packagers to change any settings that are required for their pgAdmin distribution. This may typically include certain paths and file locations. This file is optional, and may be created by packagers in the same directory as `config.py` if needed.
- `config_local.py`: This file is read after `config_distro.py` and is intended for end users to change any default or packaging specific settings that they may wish to adjust to meet local preferences or standards. This file is optional, and may be created by users in the same directory as `config.py` if needed.

Note

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) » [Deployment](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Deployment](#)

[Login Dialog](#)

[User Management Dialog](#)



[Change User Password Dialog](#)

[FAQ](#)

[User Interface](#)

[Menu Bar](#)



[Toolbar](#)

[Tabbed Browser](#)

[Latest Docs](#)

[Tree Control](#)

[Preferences Dialog](#)



[Keyboard Shortcuts](#)

[Connecting To A Server](#)

[Ask a](#)

[Managing Cluster Objects](#)

[Question](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Management Basics](#)

[Backup and Restore](#)

[Report a Bug](#)

[Developer Tools](#)

[pgAgent](#)



[pgAdmin Project Contributions](#)

[Release Notes](#)

[Screenshots](#)

[Licence](#)

The config.py File

There are multiple configuration files that are read at startup by pgAdmin.

These are as follows:

config.py: This is the main configuration file, and should not be modified. It can be used as a reference for configuration settings, that may be overridden in one of the following files.

config_distro.py: This file is read after config.py and is intended for packagers to change any settings that are required for their pgAdmin distribution. This may typically include certain paths and file locations. This file is optional, and may be created by packagers in the same directory as config.py if needed.

config_local.py: This file is read after config_distro.py and is intended for end users to change any default or packaging specific settings that they may wish to adjust to meet local preferences or standards. This file is optional, and may be created by users in the same directory as config.py if needed.

Note

If the SERVER_MODE setting is changed in config_distro.py or config_local.py, you will most likely need to re-set the LOG_FILE, SQLITE_PATH, SESSION_DB_PATH and STORAGE_DIR values as well as they will have been set based on the default configuration or overridden by the runtime.

The default config.py file is shown below for reference:

```
# -*- coding: utf-8 -*-

#####
#####

#  
# pgAdmin 4 - PostgreSQL Tools  
#  
# Copyright (C) 2013 - 2020, The pgAdmin Development Team  
# This software is released under the PostgreSQL Licence  
#  
# config.py - Core application configuration settings  
#  
#####
#####

import logging  
import os  
import sys  
import json  
if sys.version_info[0] >= 3:  
    import builtins  
else:
```

```
import __builtin__ as builtins

# We need to include the root directory in sys.path to ensure that we can

# find everything we need when running in the standalone runtime.

root = os.path.dirname(os.path.realpath(__file__)) if sys.path[0] != root:
    sys.path.insert(0, root)

from pgadmin.utils import env, IS_PY2, IS_WIN, fs_short_path

#####
#####

# Application settings

#####
#####

# Name of the application to display in the UI APP_NAME =
'pgAdmin 4'

APP_ICON = 'pg-icon'

#####
#####

# Application settings

#####
#####

# NOTE!!!

# If you change any of APP_RELEASE, APP_REVISION or
APP_SUFFIX, then you
```

```
# must also change APP_VERSION_INT to match.  
#  
# Any changes made here must also be made in  
# runtime/pgAdmin4.pro and  
# runtime/Info.plist  
#  
# Application version number components  
APP_RELEASE = 4  
APP_REVISION = 19  
# Application version suffix, e.g. 'beta1', 'dev'.  
# Usually an empty string  
# for GA releases.  
APP_SUFFIX = "  
# Numeric application version for upgrade checks. Should be in the  
# format:  
# [X]XYZZ, where X is the release version, Y is the revision, with a  
# leading  
# zero if needed, and Z represents the suffix, with a leading zero if  
# needed  
APP_VERSION_INT = 41900  
# DO NOT CHANGE!  
# The application version string, constructed from the components
```

```
if not APP_SUFFIX:
```

```
APP_VERSION = ' %s. %s' % (APP_RELEASE, APP_REVISION)  
else:
```

```
APP_VERSION = ' %s. %s- %s' % (APP_RELEASE,  
APP_REVISION, APP_SUFFIX)
```

```
# Copyright string for display in the app
```

```
# Any changes made here must also be made in  
runtime/pgAdmin4.pro
```

```
APP_COPYRIGHT = 'Copyright (C) 2013 - 2020, The pgAdmin  
Development Team'
```

```
#####
####
```

```
# Misc stuff
```

```
#####
####
```

```
# Path to the online help.
```

```
HELP_PATH = '../..../docs/en_US/_build/html/'
```

```
# Languages we support in the UI
```

```
LANGUAGES = {
```

```
'en': 'English',
```

```
'zh': 'Chinese (Simplified)',
```

```
'cs': 'Czech',
```

```
'fr': 'French',
```

```
'de': 'German',
'it': 'Italian',
'ja': 'Japanese',
'ko': 'Korean',
'pl': 'Polish',
'ru': 'Russian',
'es': 'Spanish',
}

# DO NOT CHANGE UNLESS YOU KNOW WHAT YOU ARE DOING!

# List of modules to skip when dynamically loading
MODULE_BLACKLIST = ['test']

# DO NOT CHANGE UNLESS YOU KNOW WHAT YOU ARE DOING!

# List of treeview browser nodes to skip when dynamically loading
NODE_BLACKLIST = []

#####
#####

# Server settings

#####
#####

# The server mode determines whether or not we're running on a web server
```

```
# requiring user authentication, or desktop mode which uses an
automatic

# default login.

#

# DO NOT DISABLE SERVER MODE IF RUNNING ON A
WEBSERVER!!

#

# We only set SERVER_MODE if it's not already set. That's to allow
the

# runtime to force it to False.

#

# NOTE: If you change the value of SERVER_MODE in an included
config file,
# you may also need to redefine any values below that are derived
# from it, notably various paths such as LOG_FILE
and anything

# using DATA_DIR.

if (not hasattr(builtins, 'SERVER_MODE')) or
builtins.SERVER_MODE is None:
    SERVER_MODE = True
else:
    SERVER_MODE = builtins.SERVER_MODE
```

HTTP headers to search for CSRF token when it is not provided in the form.

Default is ['X-CSRFToken', 'X-CSRF-Token']

WTF_CSRF_HEADERS = ['X-pgA-CSRFToken']

User ID (email address) to use for the default user in desktop mode.

The default should be fine here, as it's not exposed in the app.

DESKTOP_USER = 'pgadmin4@pgadmin.org'

This option allows the user to host the application on a LAN

Default hosting is on localhost

(DEFAULT_SERVER='localhost').

To host pgAdmin4 over LAN set DEFAULT_SERVER='0.0.0.0'

(or a specific

adaptor address.

#

NOTE: This is NOT recommended for production use, only for debugging

or testing. Production installations should be run as a WSGI application

behind Apache HTTPD.

DEFAULT_SERVER = '127.0.0.1'

The default port on which the app server will listen if not set in the

```
# environment by the runtime

DEFAULT_SERVER_PORT = 5050

# Enable X-Frame-Option protection.

# Set to one of "SAMEORIGIN", "ALLOW-FROM origin" or ""
# to disable.

# Note that "DENY" is NOT supported (and will be silently ignored).

# See https://tools.ietf.org/html/rfc7034 for more info.

X_FRAME_OPTIONS = "SAMEORIGIN"

# Hashing algorithm used for password storage
SECURITY_PASSWORD_HASH = 'pbkdf2_sha512'

# Reverse Proxy parameters

# You must tell the middleware how many proxies set each header
# so it knows what values to trust.

# See https://tinyurl.com/yyg7r9av
# for more information.

# Number of values to trust for X-Forwarded-For
PROXY_X_FOR_COUNT = 1

# Number of values to trust for X-Forwarded-Proto.

PROXY_X_PROTO_COUNT = 1

# Number of values to trust for X-Forwarded-Host.

PROXY_X_HOST_COUNT = 0
```

```
# Number of values to trust for X-Forwarded-Port.  
  
PROXY_X_PORT_COUNT = 1  
  
# Number of values to trust for X-Forwarded-Prefix.  
  
PROXY_X_PREFIX_COUNT = 0  
  
# NOTE: CSRF_SESSION_KEY, SECRET_KEY and  
# SECURITY_PASSWORD_SALT are no  
# longer part of the main configuration, but are stored in the  
# configuration databases 'keys' table and are auto-generated.  
  
# COMPRESSION  
  
COMPRESS_MIMETYPES = [  
    'text/html', 'text/css', 'text/xml',  
    'application/json',  
    'application/javascript'  
]  
  
COMPRESS_LEVEL = 9  
  
COMPRESS_MIN_SIZE = 500  
  
# Set the cache control max age for static files in flask to 1 year  
  
SEND_FILE_MAX_AGE_DEFAULT = 31556952  
  
# This will be added to static urls as url parameter with value as  
# APP_VERSION_INT for cache busting on version upgrade.  
  
If the value is set as
```

```
# None or empty string then it will not be added.

# eg - http://localhost:5050/pgadmin.css?intver=3.13

APP_VERSION_PARAM = 'ver'

# Add the internal version param to below extensions only
APP_VERSION_EXTN = ('.css', '.js', '.html', '.svg',
'.png', '.gif', '.ico')

# Data directory for storage of config settings etc. This shouldn't
# normally

# need to be changed - it's here as various other settings depend on
it.

# On Windows, we always store data in %APPDATA%\pgAdmin.

On other platforms,

# if we're in server mode we use /var/lib/pgadmin, otherwise
~/pgadmin

if IS_WIN:

    # Use the short path on windows

    DATA_DIR = os.path.realpath(
        os.path.join(fs_short_path(env('APPDATA')),
        u"pgAdmin"))

else:

    if SERVER_MODE:
```

```
DATA_DIR = '/var/lib/pgadmin'

else:

DATA_DIR =

os.path.realpath(os.path.expanduser(u'~/pgadmin/'))

# An optional login banner to show security warnings/disclaimers etc.
at

# login and password recovery etc. HTML may be included for basic
formatting,

# For example:

# LOGIN_BANNER = "<h4>Authorised Users Only!</h4>" \
# "Unauthorised use is strictly forbidden."

LOGIN_BANNER = ""

#####
#####

# Log settings

#####
#####

# Debug mode?

DEBUG = False

# Application log level - one of:

# CRITICAL 50

# ERROR 40
```

```
# WARNING 30

# SQL 25

# INFO 20

# DEBUG 10

# NOTSET 0

CONSOLE_LOG_LEVEL = logging.WARNING

FILE_LOG_LEVEL = logging.WARNING

# Log format.

CONSOLE_LOG_FORMAT = ' %(asctime)s: %(levelname)s\t %
%(name)s:\t %(message)s'

FILE_LOG_FORMAT = ' %(asctime)s: %(levelname)s\t %
%(name)s:\t %(message)s'

# Log file name. This goes in the data directory, except on non-Windows

# platforms in server mode.

if SERVER_MODE and not IS_WIN: LOG_FILE =
'/var/log/pgadmin/pgadmin4.log'

else:

LOG_FILE = os.path.join(DATA_DIR, 'pgadmin4.log')

#####
####
```

```
# Server Connection Driver Settings

#####
#####

# The default driver used for making connection with PostgreSQL

PG_DEFAULT_DRIVER = 'psycopg2'

# Maximum allowed idle time in minutes before which releasing the
connection

# for the particular session. (in minutes)
MAX_SESSION_IDLE_TIME = 60

#####
#####

# User account and settings storage

#####
#####

# The default path to the SQLite database used to store user
accounts and

# settings. This default places the file in the same directory as this
# config file, but generates an absolute path for use throughout the
app.

SQLITE_PATH = env('SQLITE_PATH') or os.path.join(DATA_DIR,
'pgadmin4.db')

# SQLITE_TIMEOUT will define how long to wait before throwing the
error -

# OperationError due to database lock. On slower system, you may
need to change
```

```
# this to some higher value.

# (Default: 500 milliseconds)

SQLITE_TIMEOUT = 500

# Allow database connection passwords to be saved if the user
# chooses.

# Set to False to disable password saving.

ALLOW_SAVE_PASSWORD = True

# Maximum number of history queries stored per
# user/server/database

MAX_QUERY_HIST_STORED = 20

#####
#####

# Server-side session storage path

# SESSION_DB_PATH (Default: $HOME/.pgadmin4/sessions)

#####
#####

# We use SQLite for server-side session storage. There will be one
# SQLite database object per session created.

# Specify the path used to store your session objects.
```

```
#  
  
# If the specified directory does not exist, the setup script will create  
# it with permission mode 700 to keep the session database secure.  
  
#  
  
# On certain systems, you can use shared memory (tmpfs) for  
maximum  
  
# scalability, for example, on Ubuntu:  
  
#  
  
# SESSION_DB_PATH = '/run/shm/pgAdmin4_session'  
  
#  
  
#####  
####  
  
SESSION_DB_PATH = os.path.join(DATA_DIR, 'sessions')  
SESSION_COOKIE_NAME = 'pga4_session'  
  
#####  
####  
  
# Mail server settings  
  
#####  
####  
  
# These settings are used when running in web server mode  
for confirming  
  
# and resetting passwords etc.
```

```
# See: http://pythonhosted.org/Flask-Mail/ for more info
MAIL_SERVER = 'localhost'

MAIL_PORT = 25

MAIL_USE_SSL = False

MAIL_USE_TLS = False

MAIL_USERNAME = ""

MAIL_PASSWORD = ""

MAIL_DEBUG = False

# Flask-Security overrides Flask-Mail's

MAIL_DEFAULT_SENDER setting, so

# that should be set as such:

SECURITY_EMAIL_SENDER = 'no-reply@localhost'

#####
#####

# Mail content settings

#####
#####

# These settings define the content of password reset emails

SECURITY_EMAIL SUBJECT_PASSWORD_RESET = "Password
reset instructions for %s" \

% APP_NAME
```

```
SECURITY_EMAIL SUBJECT_PASSWORD_NOTICE = "Your %s  
password has been reset" \  
% APP_NAME  
  
SECURITY_EMAIL SUBJECT_PASSWORD_CHANGE_NOTICE =  
\  
"Your password for %s has been changed" % APP_NAME  
  
#####  
####  
  
# Upgrade checks  
  
#####  
####  
  
# Check for new versions of the application?  
  
UPGRADE_CHECK_ENABLED = True  
  
# Where should we get the data from?  
  
UPGRADE_CHECK_URL =  
  
'https://www.pgadmin.org/versions.json'  
  
# What key should we look at in the upgrade data file?  
  
UPGRADE_CHECK_KEY = 'pgadmin4'  
  
# Which CA file should we use?  
  
# Default to cacert.pem in the same directory as config.py et al.  
  
CA_FILE =  
  
os.path.join(os.path.dirname(os.path.realpath(__file__)),
```

```
"cacert.pem")  
#####  
####  
  
# Storage Manager storage url config settings  
  
# If user sets STORAGE_DIR to empty it will show all volumes if  
platform  
  
# is Windows, '/' if it is Linux, Mac or any other unix type system.  
  
# For example:  
  
# 1. STORAGE_DIR = get_drive("C") or get_drive() # return C:/ by  
default  
  
# where C can be any drive character such as "D", "E",  
"G" etc  
  
# 2. Set path manually like  
  
# STORAGE_DIR = "/path/to/directory/"  
#####  
####  
  
STORAGE_DIR = os.path.join(DATA_DIR, 'storage')  
#####  
####  
  
# Default locations for binary utilities (pg_dump, pg_restore etc)  
  
#  
  
# These are intentionally left empty in the main config file, but are
```

```
# expected to be overridden by packagers in config_distro.py.

#
# A default location can be specified for each database driver ID, in
# a dictionary. Either an absolute or relative path can be specified.

# In cases where it may be difficult to know what the working
# directory

# is, "$DIR" can be specified. This will be replaced with the path to
# the

# top-level pgAdmin4.py file. For example, on macOS we might use:

#
# $DIR/../../SharedSupport

#
#####
#####

DEFAULT_BINARY_PATHS = {

    "pg": "",

    "ppas": "",

    "gpdb": ""

}

#####
#####

# Test settings - used primarily by the regression suite, not for users
```

```
#####
####
```

```
# The default path for SQLite database for testing
TEST_SQLITE_PATH = os.path.join(DATA_DIR,
'test_pgadmin4.db')
```

```
#####
####
```

```
# Allows flask application to response to the each request
asynchronously
```

```
#####
####
```

```
THREADED_MODE = True
```

```
#####
####
```

```
# Do not allow SQLALCHEMY to track modification as it is going to
be
```

```
# deprecated in future
```

```
#####
####
```

```
SQLALCHEMY_TRACK_MODIFICATIONS = False
```

```
#####
####
```

```
# Number of records to fetch in one batch in query tool when query
result
```

```
# set is large.
```

```
#####
#####

ON_DEMAND_RECORD_COUNT = 1000

#####
#####

# Allow users to display Gravatar image for their username in Server mode

#####
#####

SHOW_GRAVATAR_IMAGE = True

#####
#####

# Set cookie path

#####
#####

COOKIE_DEFAULT_PATH = '/'

COOKIE_DEFAULT_DOMAIN = None

SESSION_COOKIE_DOMAIN = None

SESSION_COOKIE_SAMESITE = 'Lax'

#####
#####

# Skip storing session in files and cache for specific paths

#####
#####
```

```
SESSION_SKIP_PATHS = [
    '/misc/ping'
]

#####
#####

# Session expiration support

#####
#####

# SESSION_EXPIRATION_TIME is the interval in Days.

Session will be

# expire after the specified number of *days*.

SESSION_EXPIRATION_TIME = 1

# CHECK_SESSION_FILES_INTERVAL is interval in Hours.

Application will check

# the session files for cleanup after specified number of
*hours*.

CHECK_SESSION_FILES_INTERVAL = 24

# USER_INACTIVITY_TIMEOUT is interval in Seconds. If the
pgAdmin screen is left

# unattended for <USER_INACTIVITY_TIMEOUT> seconds then the
user will

# be logged out. When set to 0, the timeout will be disabled.
```

```
# If pgAdmin doesn't detect any activity in the time specified (in
seconds),

# the user will be forcibly logged out from pgAdmin. Set to zero to
disable

# the timeout.

# Note: This is applicable only for SERVER_MODE=True.

USER_INACTIVITY_TIMEOUT = 0

# OVERRIDE_USER_INACTIVITY_TIMEOUT when set to True will
override

# USER_INACTIVITY_TIMEOUT when long running queries in the
Query Tool

# or Debugger are running. When the queries complete, the inactivity
timer

# will restart in this case. If set to False, user inactivity may cause
# transactions or in-process debugging sessions to be aborted.

OVERRIDE_USER_INACTIVITY_TIMEOUT = True

#####
#####

# SSH Tunneling supports only for Python 2.7 and 3.4+

#####
#####

SUPPORT_SSH_TUNNEL = True

# Allow SSH Tunnel passwords to be saved if the user chooses.
```

```
# Set to False to disable password saving.  
ALLOW_SAVE_TUNNEL_PASSWORD = False  
#####  
####  
  
# Master password is used to encrypt/decrypt saved server  
passwords  
  
# Applicable for desktop mode only  
#####  
####  
  
MASTER_PASSWORD_REQUIRED = True  
#####  
####  
  
# Allows pgAdmin4 to create session cookies based on IP  
address, so even  
  
# if a cookie is stolen, the attacker will not be able to connect to the  
# server using that stolen cookie.  
  
# Note: This can cause problems when the server is deployed in  
dynamic IP  
  
# address hosting environments, such as Kubernetes or behind load  
# balancers. In such cases, this option should be set to False.  
#####  
####  
  
ENHANCED_COOKIE_PROTECTION = True
```

```
#####
#####

# Local config settings

#####
#####

# Load distribution-specific config overrides try:
from config_distro import *
except ImportError:
pass

# Load local config overrides

try:
from config_local import *
except ImportError:
pass

# SUPPORT_SSH_TUNNEL can be override in local config file and
if that

# setting is False in local config then we should not check the Python
version.

if (SUPPORT_SSH_TUNNEL is True and ((sys.version_info[0] == 2
and sys.version_info[1] < 7) or

(bsys.version_info[0] == 3 and sys.version_info[1] < 4))):

SUPPORT_SSH_TUNNEL = False
```

```
ALLOW_SAVE_TUNNEL_PASSWORD = False
```

```
# Disable USER_INACTIVITY_TIMEOUT when  
SERVER_MODE=False if not SERVER_MODE:
```

```
USER_INACTIVITY_TIMEOUT = 0
```

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | Development)

[pgAdmin 4 documentation](#) > [Getting Started](#) >

[previous](#) [next](#) [index](#)

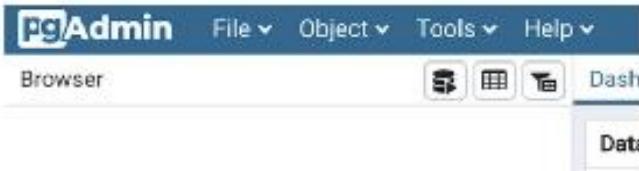
Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

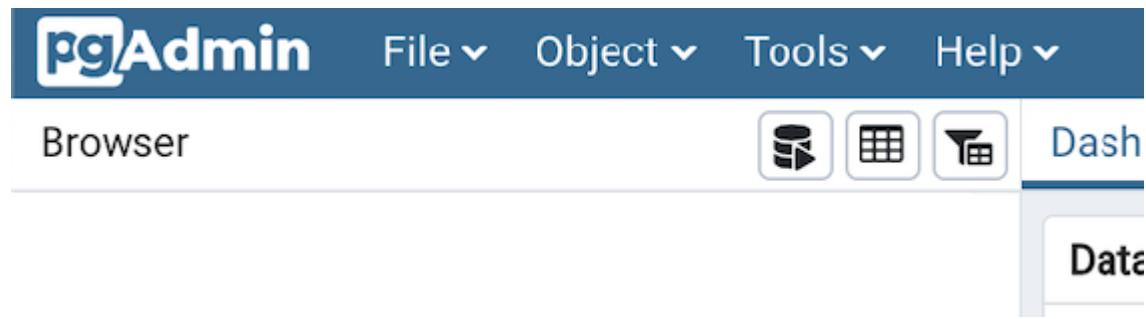
- [Getting Started](#)
 - [Deployment](#)
 - [Login Dialog](#)
 - [User Management Dialog](#)
 - [Change User Password Dialog](#)
 - [User Interface](#)
 - [Menu Bar](#)
 - [Toolbar](#)
 - [Tabbed Browser](#)
 - [Tree Control](#)
 - [Preferences Dialog](#)
 - [Keyboard Shortcuts](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Toolbar

The pgAdmin toolbar provides shortcut buttons for frequently used features like View Data and the Query Tool which are most frequently used in pgAdmin. This toolbar is visible on the Browser panel. Buttons get enabled/disabled based on the selected browser node.



- Use the [Query Tool](#) button to open the Query Tool in the current database context.
- Use the [View Data](#) button to view/edit the data stored in a selected table.
- Use the [Filtered Rows](#) button to access the Data Filter popup to apply a filter to a set of data for viewing/editing.



Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Deployment](#)

[Login Dialog](#)

[User Management Dialog](#)



[Change User Password Dialog](#)

[FAQ](#)

[User Interface](#)

[Menu Bar](#)



Toolbar

[Tabbed Browser](#)

[Latest Docs](#)

[Tree Control](#)

[Preferences Dialog](#)



[Keyboard Shortcuts](#)

[Connecting To A Server](#)

[Ask a](#)

[Managing Cluster Objects](#)

[Question](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Management Basics](#)

[Backup and Restore](#)

[Report a Bug](#)

[Developer Tools](#)

[pgAgent](#)



[pgAdmin Project Contributions](#)

[Release Notes](#)

[Screenshots](#)

[Licence](#)

Toolbar A small blue square icon with a white outline, representing a toolbar.

The pgAdmin toolbar provides shortcut buttons for frequently used features like View Data and the Query Tool which are most frequently used in pgAdmin. This toolbar is visible on the Browser panel. Buttons get enabled/disabled based on the selected browser node.

Use the [Query Tool](#) button to open the Query Tool in the current database context.

Use the [View Data](#) button to view/edit the data stored in a selected table.

Use the [Filtered Rows](#) button to access the Data Filter popup to apply a filter to a set of data for viewing/editing.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | Development)

[pgAdmin 4 4.19 documentation](#) > [Getting Started](#) >

[previous](#) [next](#) [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
 - [Deployment](#)
 - [Login Dialog](#)
 - [User Management Dialog](#)
 - [Change User Password Dialog](#)
 - [User Interface](#)
 - [Menu Bar](#)
 - [Toolbar](#)
 - [Tabbed Browser](#)
 - [Tree Control](#)
 - [Preferences Dialog](#)
 - [Keyboard Shortcuts](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Tree Control

The left pane of the main window displays a tree control (the pgAdmin tree control) that provides access to the objects that reside on a server.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Deployment](#)

[Login Dialog](#)

[User Management Dialog](#)



[Change User Password Dialog](#)

[FAQ](#)

[User Interface](#)

[Menu Bar](#)



[Toolbar](#)

[Tabbed Browser](#)

[Latest Docs](#)

Tree Control

[Preferences Dialog](#)



[Keyboard Shortcuts](#)

[Connecting To A Server](#)

[Ask a](#)

[Managing Cluster Objects](#)

[Question](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Management Basics](#)

[Backup and Restore](#)

[Report a Bug](#)

[Developer Tools](#)

[pgAgent](#)



[pgAdmin Project Contributions](#)

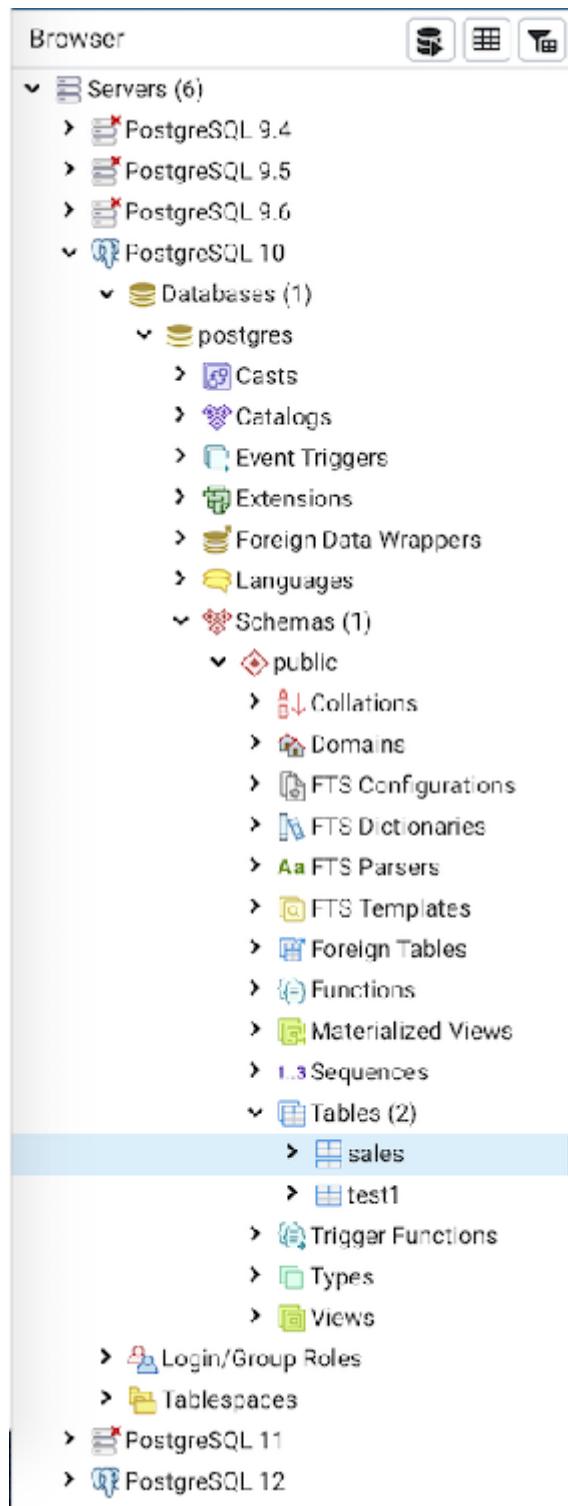
[Release Notes](#)

[Screenshots](#)

[Licence](#)

Tree Control

The left pane of the main window displays a tree control (the *pgAdmin* tree control) that provides access to the objects that reside on a server.



You can expand nodes in the tree control to view the database objects that reside on a selected server. The tree control expands to display a hierarchical view:

Use the plus sign (+) to the left of a node to expand a segment of the tree control.

Click the minus sign (-) to the left of a node to close that node.

You can also drag and drop certain objects to the Query Tool which can save time in typing long object names. Text containing the object name will be fully qualified with schema. Double quotes will be added if required. For functions and procedures, the function name along with parameter names will be pasted in the Query Tool.

Access context-sensitive menus by right-clicking on a node of the tree control to perform common tasks. Menus display options that include one or more of the following selections (options appear in alphabetical order):

- Option
- Action
- Option
- Action

*Add named
restore point*

Click to create and enter the name of a restore point.

Backup...

Click to open the [Backup...](#) dialog to backup database objects.

Backup

Click to open the [Backup Globals...](#) dialog to backup *Globals...* cluster objects.

Backup

Click to open the [Backup Server...](#) dialog to backup a Server... server.

Connect

Click to open the [Connect to Server](#) dialog to establish a Server... connection with a server.

Create

Click to access a context menu that provides context-sensitive selections. Your selection opens a *Create* dialog for creating a new object.

CREATE Script Click to open the [Query tool](#) to edit or view the CREATE script.

Debugging

Click through to open the [Debug](#) tool or to select *Set breakpoint* to stop or pause a script execution.

Delete/Drop

Click to delete the currently selected object from the server.

Disconnect

Click to terminate a database connection.

Database...

Disconnect

Click to refresh the currently selected object.

Server...

Drop Cascade Click to delete the currently selected object and all dependent objects from the server.

Debugging

Click to access the [Debugger](#) tool.

Grant Wizard Click to access the [Grant Wizard](#) tool.

Maintenance... Click to open the [Maintenance...](#) dialog to VACUUM, ANALYZE, REINDEX, or CLUSTER.

Properties...

Click to review or modify the currently selected object's properties.

Refresh...

Click to refresh the currently selected object.

Reload

Click to update configuration files without restarting the *Configuration...* server.

Restore...

Click to access the [Restore](#) dialog to restore database files from a backup.

View Data

Use the *View Data* option to access the data stored in a selected table with the *Data Output* tab of the *Query Tool*.

The context-sensitive menus associated with *Tables* and nested *Table* nodes provides additional display options (options appear in alphabetical order):

Option

Action

Option

Action

Import/Export... Click open the [Import/Export...](#) dialog to import data to or export data from the selected table.

Reset Statistics Click to reset statistics for the selected table.

Scripts

Click to open the [Query tool](#) to edit or view the selected script from the flyout menu.

Truncate

Click to remove all rows from a table.

Truncate

Click to remove all rows from a table and its child tables.

Cascade

View First 100 Click to access a data grid that displays the first 100

Rows

rows of the selected table.

View Last 100 Click to access a data grid that displays the last 100
Rows

rows of the selected table.

View All Rows Click to access a data grid that displays all rows of the selected table.

View Filtered

Click to access the *Data Filter* popup to apply a filter to a *Rows...* set of data.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Trigger Function Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Trigger Function Dialog ¶

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Trigger Function Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)



[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)



[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

Trigger Function Dialog

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

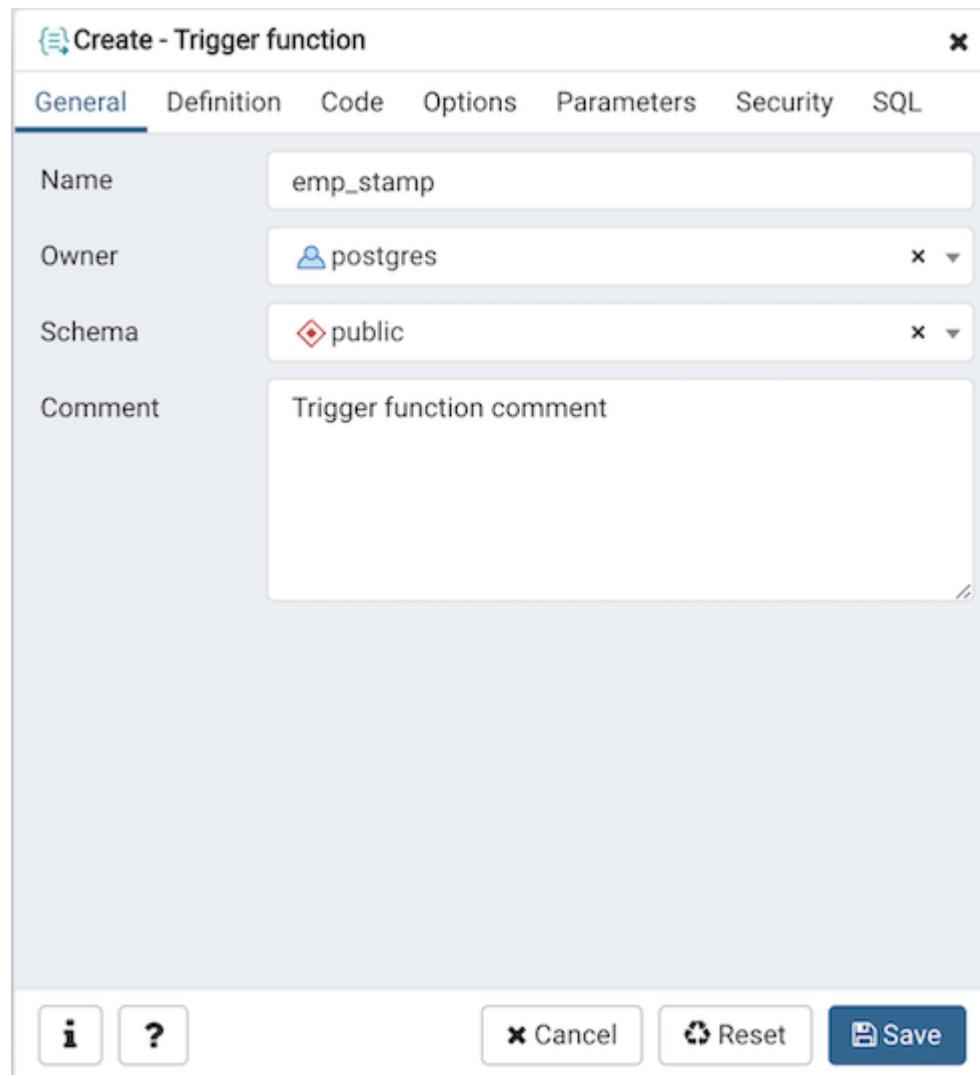
[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

Trigger Function Dialog



Use the *Trigger function* dialog to create or manage a trigger_function. A trigger function defines the action that will be invoked when a trigger fires.

The *Trigger function* dialog organizes the development of a trigger function through the following dialog tabs: *General*, *Definition*, *Code*, *Options*, *Parameters* and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

Use the fields in the *General* tab to identify the trigger function: Use the *Name* field to add a descriptive name for the trigger function.

The name will be displayed in the *pgAdmin* tree control. Please note that trigger functions will be invoked in alphabetical order.

Use the drop-down listbox next to *Owner* to select the role that will own the trigger function.

Select the name of the schema in which the trigger function will reside from the drop-down listbox in the *Schema* field.

Store notes about the trigger function in the *Comment* field.

Click the *Definition* tab to continue.

Create - Trigger function

General **Definition** Code Options Parameters Security SQL

Return type

trigger

Language

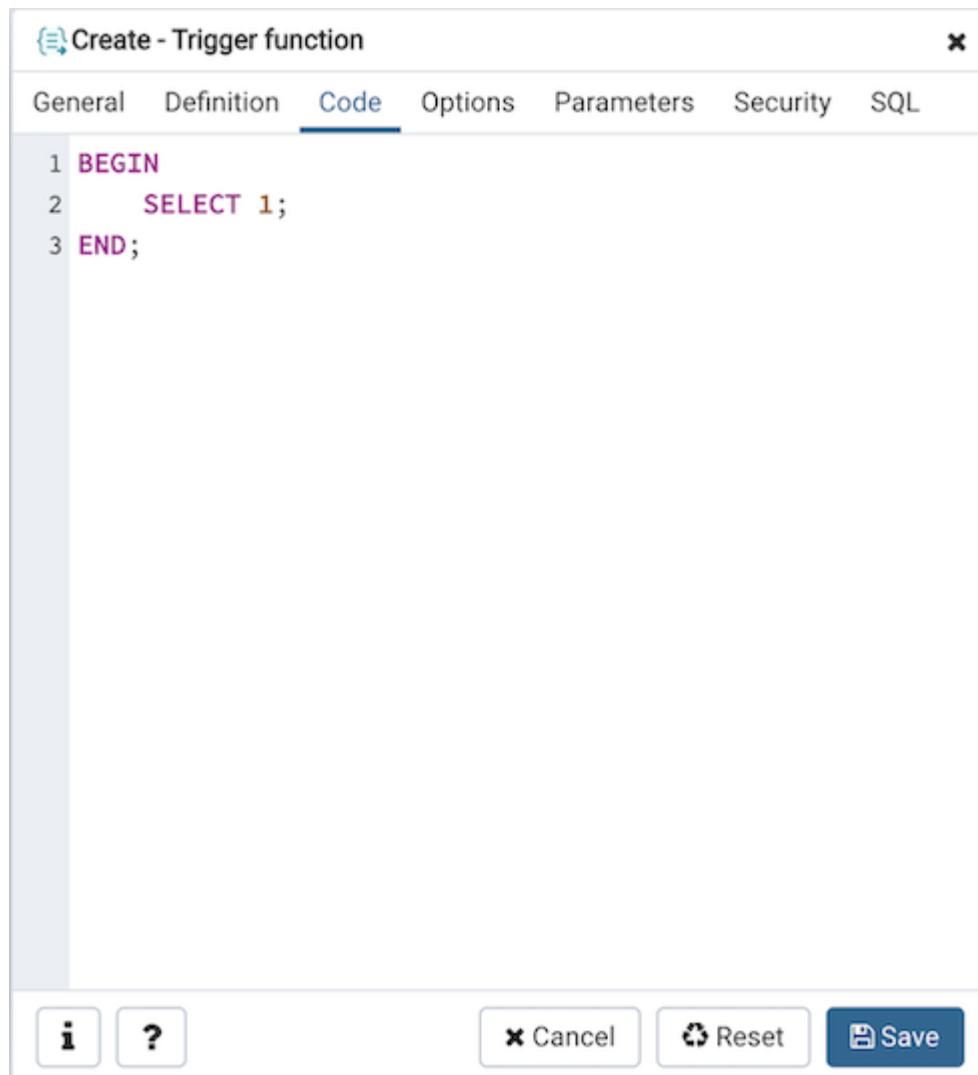
plpgsql



Cancel

Reset

Save



Use the fields in the *Definition* tab to define the trigger function: Use the drop-down listbox next to *Return type* to specify the pseudotype that is associated with the trigger function:

Select *trigger* if you are creating a DML trigger.

Select *event_trigger* if you are creating a DDL trigger.

Use the drop-down listbox next to *Language* to select the implementation language. The default is *plpgsql*.

Click the *Code* tab to continue.

Use the *Code* field to write the code that will execute when the trigger function is called.

Create - Trigger function

General Definition Code **Options** Parameters Security SQL

Volatility Select an item...

Returns a set? No

Strict? No

Security of definer? No

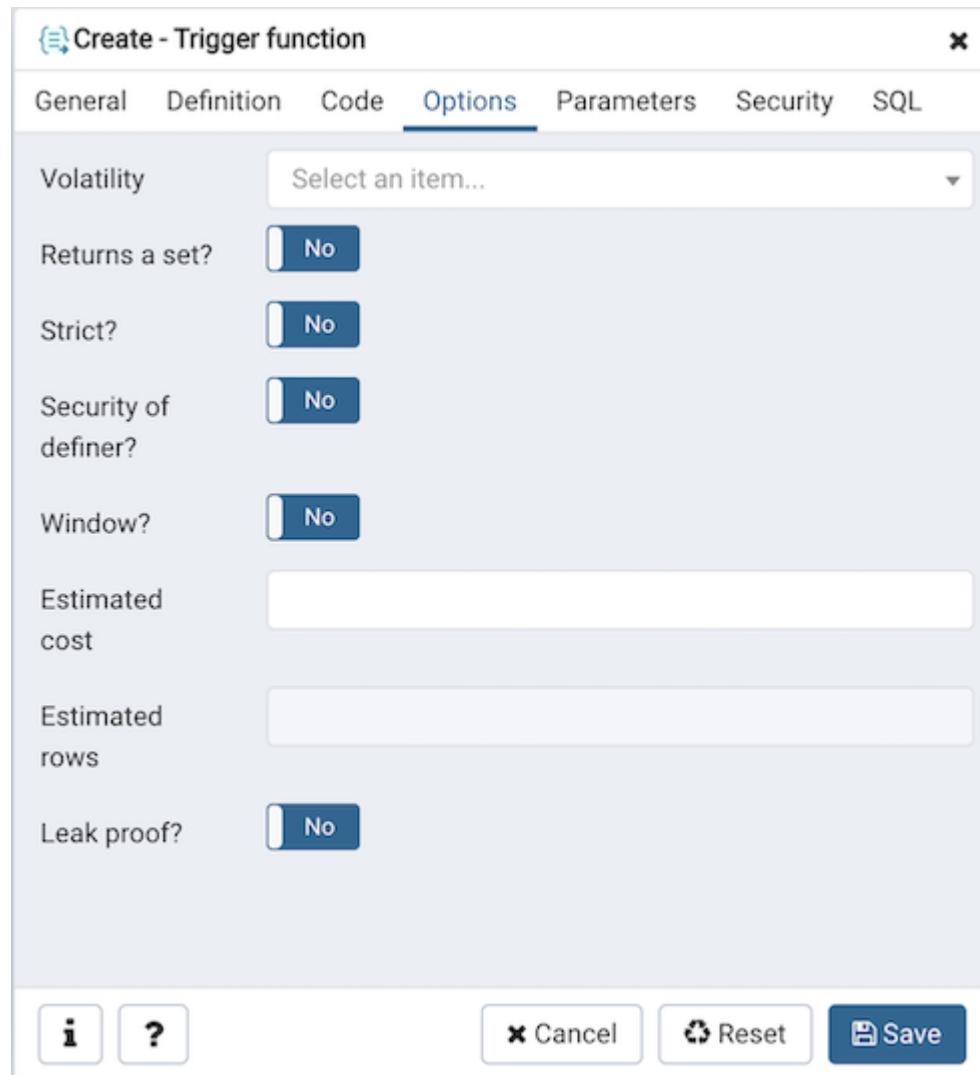
Window? No

Estimated cost

Estimated rows

Leak proof? No

i ? Cancel Reset Save



Click the *Options* tab to continue.

Use the fields in the *Options* tab to describe or modify the action of the trigger function:

Use the drop-down listbox next to *Volatility* to select one of the following:

VOLATILE indicates that the trigger function value can change even within a single table scan.

STABLE indicates that the trigger function cannot modify the database, and that within a single table scan it will consistently return the same result for the same argument values.

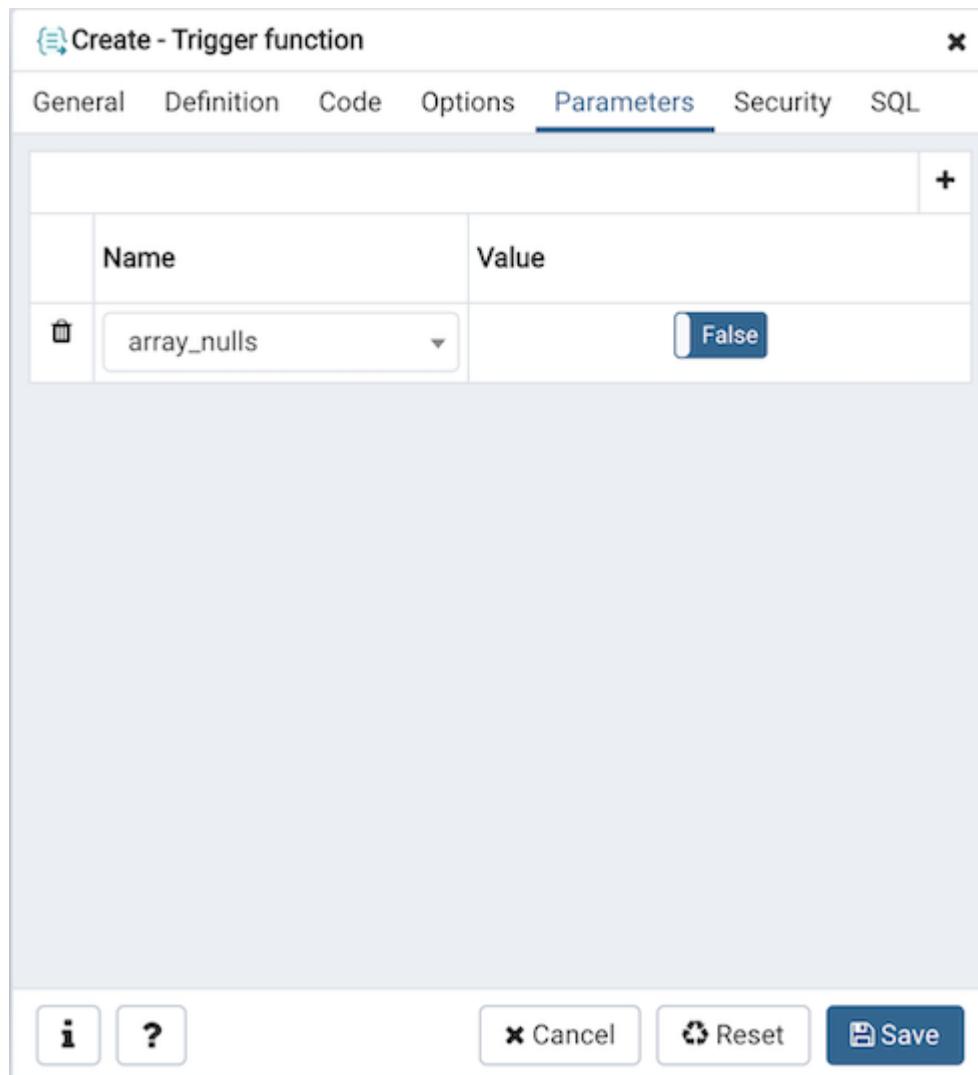
IMMUTABLE indicates that the trigger function cannot modify the database and always returns the same result when given the same argument values.

Move the *Returns a Set?* switch to indicate if the trigger function returns a set that includes multiple rows. The default is *No*.

Move the *Strict?* switch to indicate if the trigger function always returns NULL whenever any of its arguments are NULL. If *Yes*, the function is not executed when there are NULL arguments; instead a NULL result is assumed automatically. The default is *No*.

Move the *Security of definer?* switch to specify that the trigger function is to be executed with the privileges of the user that created it. The default is *No*.

Move the *Window?* switch to indicate that the trigger function is a window function rather than a plain function. The default is *No*. This is currently only useful for trigger functions written in C.



Use the *Estimated cost* field to specify a positive number representing the estimated execution cost for the trigger function, in units of `cpu_operator_cost`. If the function returns a set, this is the cost per returned row.

Use the *Estimated rows* field to specify a positive number giving the estimated number of rows that the query planner should expect the trigger function to return. This is only allowed when the function is declared to return a set. The default assumption is 1000 rows.

Move the *Leak proof?* switch to indicate whether the trigger function has side effects. The default is *No*. This option can only be set by the superuser.

Click the *Parameters* tab to continue.

Use the fields in the *Parameters* tab to specify settings that will be applied when the trigger function is invoked. Click the *Add* icon (+) to add a *Name/ Value* pair to the table below.

Use the drop-down listbox in the *Name* field to select a parameter.

Use the *Value* field to specify the value that will be associated with the selected parameter. This field is context-sensitive.

Click the *Add* icon (+) to set additional parameters; to discard a parameter, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *Security* tab to continue.

Create - Trigger function

General Definition Code Options Parameters Security SQL

Privileges

Grantee	Privileges	Grantor
<input type="button" value="postgres"/>	<input checked="" type="checkbox"/> EXECUTE <input checked="" type="checkbox"/> WITH GRANT OPTION	<input type="button" value="postgres"/>

Security labels

Provider	Security label
----------	----------------

Buttons: Cancel Reset Save

Use the *Security* tab to assign privileges and define security labels.

Use the *Privileges* panel to assign usage privileges for the trigger function to a role. Click the *Add* icon (+) to add a role to the table.

Select the name of the role from the drop-down listbox in the *Grantee* field.

Click inside the *Privileges* field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.

Click the *Add* icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Use the *Security Labels* panel to define security labels applied to the trigger function. Click the *Add* icon (+) to add each security label selection:

Specify a security label provider in the *Provider* field. The named provider must be loaded and must consent to the proposed labeling operation.

Specify a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the *Add* icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

Create - Trigger function

General Definition Code Options Parameters Security SQL

```
1 CREATE FUNCTION public.emp_stamp()
2     RETURNS trigger
3     LANGUAGE 'plpgsql'
4     NOT LEAKPROOF
5     SET array_nulls=false
6 AS $BODY$BEGIN
7     SELECT 1;
8 END;$BODY$;
9
10 ALTER FUNCTION public.emp_stamp()
11     OWNER TO postgres;
12
13 GRANT EXECUTE ON FUNCTION public.emp_stamp() TO postgres;
14
15 REVOKE ALL ON FUNCTION public.emp_stamp() FROM PUBLIC;
16
17 COMMENT ON FUNCTION public.emp_stamp()
18     IS 'Trigger function comment';
```

i ? Cancel Reset Save

Your entries in the *Trigger function* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit other tabs to modify the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Trigger function* dialog:

The example shown demonstrates creating a trigger function named *emp_stamp*.

Click the *Info* button (i) to access online help.

Click the **Save** button to save work.

Click the **Cancel** button to exit without saving work.

Click the **Reset** button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) > [Creation or Modifying a Table](#) > [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Trigger Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
 - [Check Dialog](#)
 - [Column Dialog](#)
 - [Compound Trigger Dialog](#)
 - [Exclusion Constraint Dialog](#)
 - [Foreign key Dialog](#)
 - [Index Dialog](#)
 - [Primary key Dialog](#)
 - [Rule Dialog](#)
 - [Table Dialog](#)
 - [Trigger Dialog](#)
 - [Unique Constraint Dialog](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Trigger Dialog

Use the *Trigger* dialog to create a trigger or modify an existing trigger. A trigger executes a specified function when certain events occur.

The *Trigger* dialog organizes the development of a trigger through the following dialog tabs: *General*, *Definition*, *Events*, and *Code*. The *SQL* tab displays the SQL code generated by dialog selections.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Creating or Modifying a Table](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



Trigger Dialog

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[□](#)

[Check Dialog](#)

[Column Dialog](#)

[Latest Docs](#)

[Compound Trigger Dialog](#)

[Exclusion Constraint Dialog](#)

[□](#)

[Foreign key Dialog](#)

[Index Dialog](#)

[Ask a](#)

[Primary key Dialog](#)

[Question](#)

[Rule Dialog](#)

[Table Dialog](#)

[□](#)

[Trigger Dialog](#)

[Report a Bug](#)

[Unique Constraint Dialog](#)

[Management Basics](#)

[Backup and Restore](#)



[Developer Tools](#)

[Screenshots](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Trigger Dialog

Use the *Trigger* dialog to create a trigger or modify an existing trigger. A trigger executes a specified function when certain events occur.

The *Trigger* dialog organizes the development of a trigger through the following dialog tabs: *General*, *Definition*, *Events*, and *Code*. The *SQL* tab displays the SQL code generated by dialog selections.

>Create - Trigger

X

General Definition Events Transition Code SQL

Name log_update

Comment Log any changes to date for distributors table



Cancel

Reset

Save

→ Create - Trigger

General Definition Events Transition Code SQL

Row trigger? Yes

Constraint trigger? Yes

Deferrable? Yes

Deferred? Yes

Trigger function

Arguments

i ? Cancel Reset Save

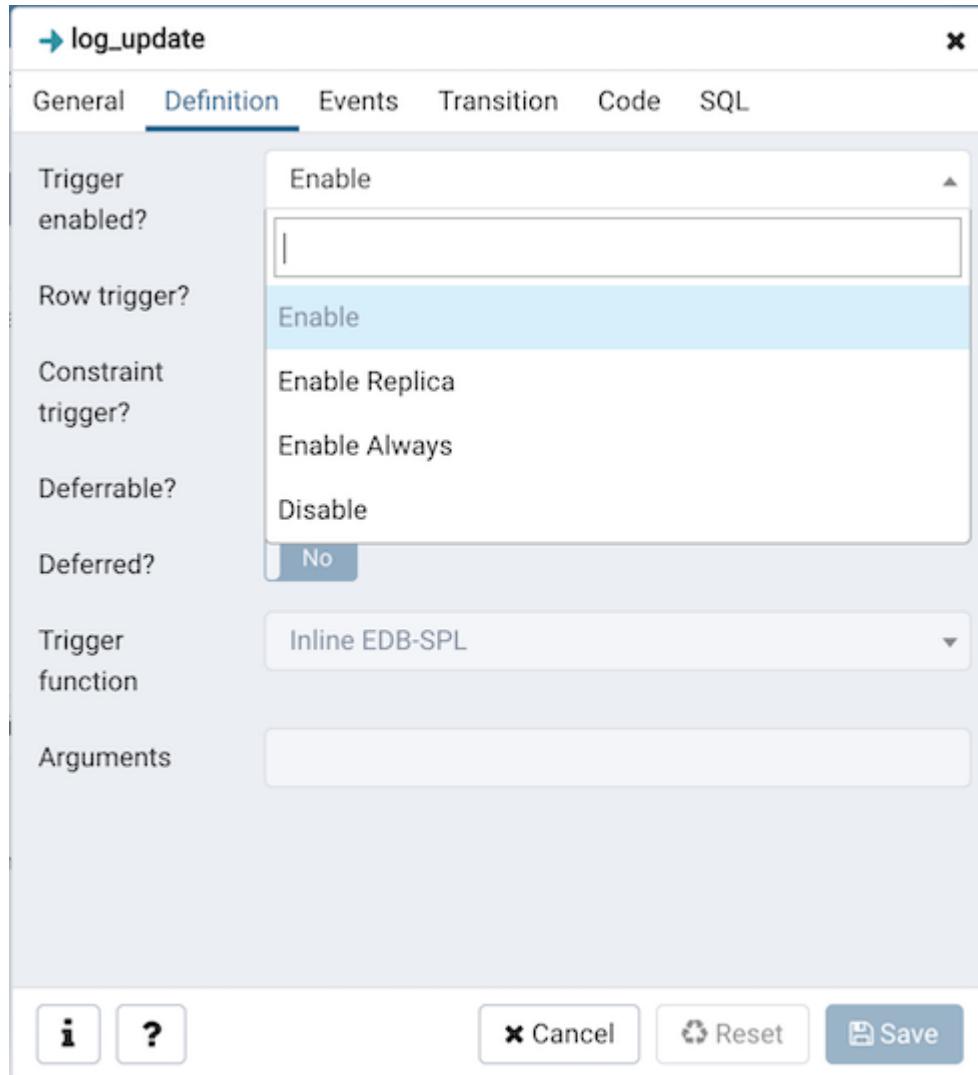
Use the fields in the *General* tab to identify the trigger: Use the *Name* field to add a descriptive name for the trigger. This must be distinct from the name of any other trigger for the same table. The name will be displayed in the *pgAdmin* tree control. Note that if multiple triggers of the same kind are defined for the same event, they will be fired in alphabetical order by name.

Store notes about the trigger in the *Comment* field.

Click the *Definition* tab to continue.

Use the fields in the *Definition* tab to define the trigger: Move the *Row trigger?* switch to the *No* position to disassociate the trigger

from firing on each row in a table. The default is *Yes*.



Move the *Constraint trigger?* switch to the *Yes* position to specify the trigger is a constraint trigger.

If enabled, move the *Deferrable?* switch to the *Yes* position to specify the timing of the constraint trigger is deferrable and can be postponed until the end of the statement. The default is *No*.

If enabled, move the *Deferred?* switch to the *Yes* position to specify the timing of the constraint trigger is deferred to the end of the statement causing the triggering event. The default is *No*.

Use the drop-down listbox next to *Trigger Function* to select a trigger function or procedure.

Use the *Arguments* field to provide an optional (comma-separated) list of arguments to the function when the trigger is executed. The arguments are literal string constants.

Trigger enabled field is available in trigger dialog once the trigger is created. You can select one of the four options available.

Click the *Events* tab to continue.

→ Create - Trigger ×

General Definition **Events** Transition Code SQL

Fires AFTER ▼

Events

INSERT	<input checked="" type="checkbox"/> Yes	UPDATE	<input checked="" type="checkbox"/> Yes
DELETE	<input checked="" type="checkbox"/> Yes	TRUNCATE	<input type="checkbox"/> No

When 1

Columns x distributors ×

i ? ✖ Cancel ↻ Reset 💾 Save

Use the fields in the *Events* tab to specify how and when the trigger fires: Use the drop-down listbox next to the *Fires* fields to determine if the trigger fires *BEFORE* or *AFTER* a specified event. The default is *BEFORE*.

Select the type of event(s) that wil invoke the trigger; to select an event type, move the switch next to the event to the *YES* position.

The supported event types are *INSERT*, *UPDATE*, *DELETE*, and *TRUNCATE*.

Use the *When* field to provide a boolean condition that wil invoke the trigger.

If defining a column-specific trigger, use the *Columns* field to specify the columns or columns that are the target of the trigger.

Click the *Code* tab to continue.

>Create - Trigger

X

General Definition Events Transition **Code** SQL

```
1 BEGIN
2     SELECT 1;
3 END
```



X Cancel

Reset

Save

>Create - Trigger

General Definition Events Transition Code SQL

```
1 CREATE OR REPLACE TRIGGER log_update
2   AFTER INSERT OR DELETE OR UPDATE OF distributors
3   ON public.test
4   DEFERRABLE INITIALLY DEFERRED      FOR EACH ROW
5   BEGIN
6     SELECT 1;
7   END;
8
9 COMMENT ON TRIGGER log_update ON public.test
10    IS 'Log any changes to date for distributors tabl
```

i ? Cancel Reset Save

Use the *Code* field to specify any additional code that will be invoked when the trigger fires.

Click the *SQL* tab to continue.

Your entries in the *Trigger* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example 

The following is an example of the sql command generated by user selections in the *Trigger* dialog:

The example demonstrates creating a trigger named *log_update*.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Type Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Type Dialog

Use the Type dialog to register a custom data type.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



Type Dialog

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)



[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)



[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)



[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

Type Dialog

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)

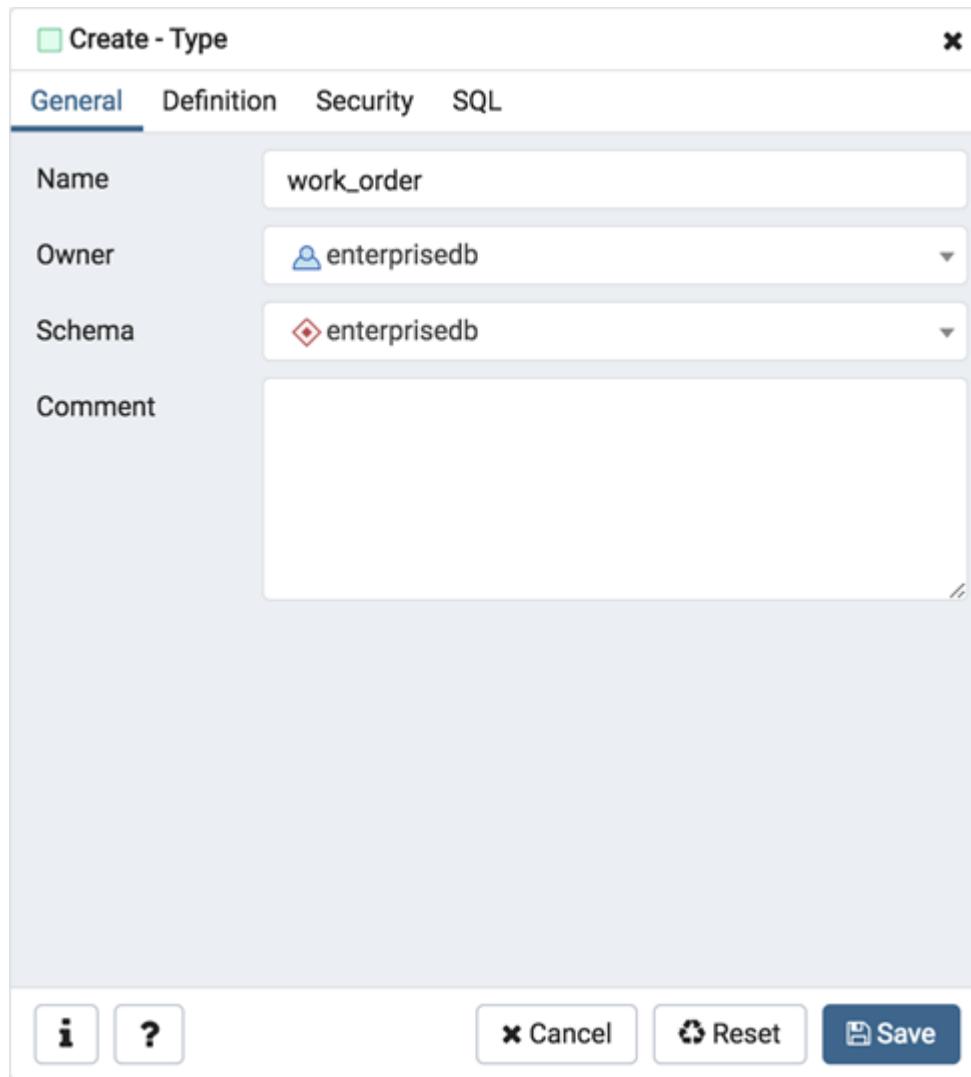
[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

Type Dialog

Use the *Type* dialog to register a custom data type.



The *Type* dialog organizes the development of a data type through the following dialog tabs: *General*, *Definition*, and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

Use the fields in the *General* tab to identify the custom data type:
Use the *Name* field to add a descriptive name for the type. The name will be displayed in the *pgAdmin* tree control. The type name

must be distinct from the name of any existing type, domain, or table in the same schema.

Use the drop-down listbox next to *Owner* to select the role that will own the type.

Select the name of the schema in which the type will reside from the drop-down listbox in the *Schema* field.

Store notes about the type in the *Comments* field.

Click the *Definition* tab to continue.

Select a data type from the drop-down listbox next to *Type* on the *Definition* tab; the panel below changes to display the options appropriate for the selected data type. Use the fields in the panel to define the data type.

There are five data types:

Composite Type

Enumeration Type

Range Type

External Type (or *Base Type*)

Shell Type

If you select *Composite* in the *Type* field, the *Definition* tab displays the *Composite Type* panel:

Create - Type

General **Definition** Security SQL

Type: Composite

Composite Type

	Member Name	Type	Length/precision	Scale	Collation
	data_member	"char"			

i **?** **Cancel** **Reset** **Save**

Click the *Add* icon (+) to provide attributes of the type. Fields on the *General* panel are context sensitive and may be disabled.

Use the *Member Name* field to add an attribute name.

Use the drop-down listbox in the *Type* field to select a datatype.

Use the *Length/Precision* field to specify the maximum length of a non-numeric type, or the total count of significant digits in a numeric type.

Use the *Scale* field to specify the number of digits to the right of the decimal point.

Use the drop-down listbox in the *Collation* field to select a collation (if applicable).

Click the *Add* icon (+) to define an additional member; click the trash icon to the left of the row to discard a row.

If you select the *Enumeration* in the *Type* field, the *Definition* tab displays the *Enumeration Type* panel:

Create - Type

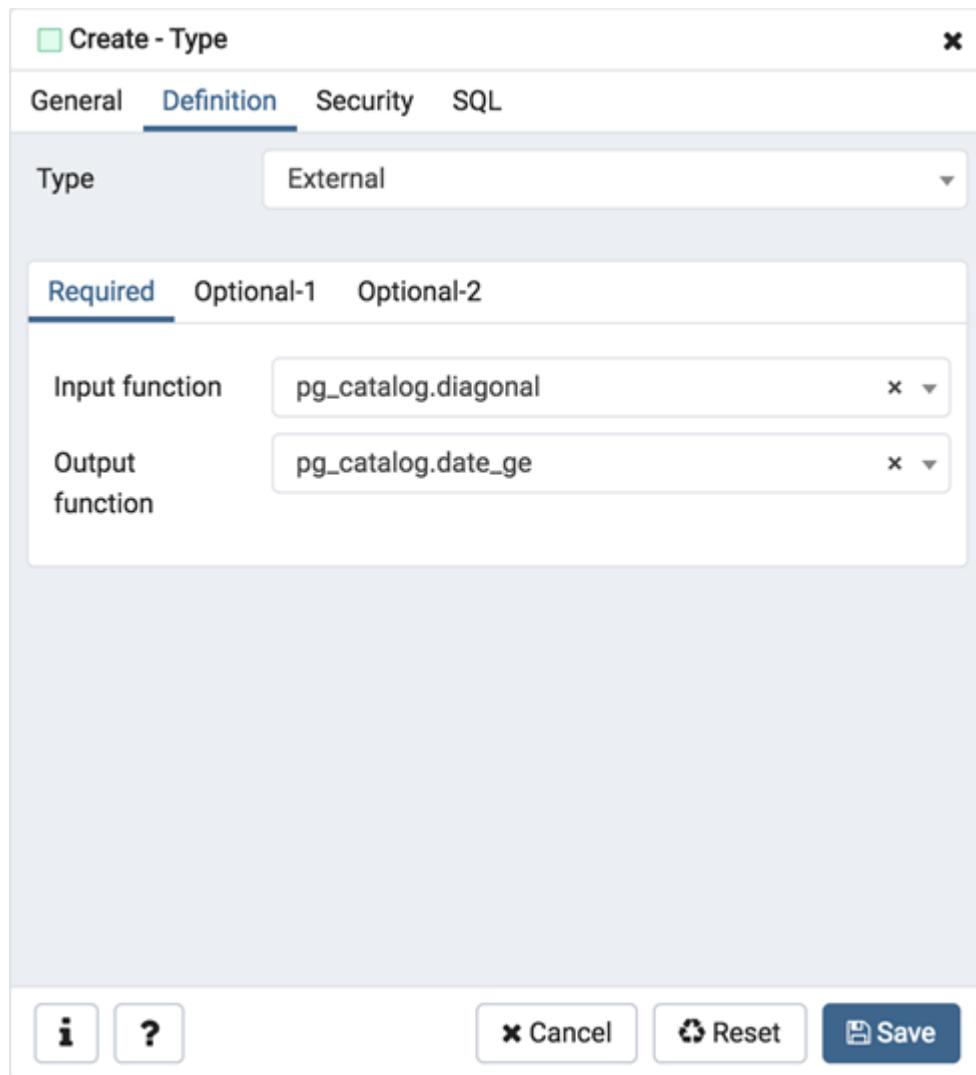
General **Definition** Security SQL

Type Enumeration

Enumeration Type

Label
trash data_label

i **?** **x Cancel** **Reset** **Save**



Click the *Add* icon (+) to provide a label for the type.

Use the *Label* field to add a label, which must be less than 64 bytes long.

Click the *Add* icon (+) after each selection to create additional labels; to discard a label, click the trash icon to the left of the row.

If you select *External*, the *Definition* tab displays the *External Type* panel: On the *Required* tab:

Use the drop-down listbox next to the *Input function* field to add an *input_function*. The *input_function* converts the type's external

textual representation to the internal representation used by the operators and functions defined for the type.

Use the drop-down listbox next to the *Output function* field to add an output_function. The output_function converts the type's internal representation used by the operators and functions defined for the type to the type's external textual representation.

On the *Optional-1* tab:

Use the drop-down listbox next to the optional *Receive Function* field to select a receive_function. The optional receive_function converts the type's external binary representation to the internal representation. If this function is not supplied, the type cannot participate in binary input.

Use the drop-down listbox next to the optional *Send function* field to select a send_function. The optional send_function converts from the internal representation to the external binary representation. If this function is not supplied, the type cannot participate in binary output.

Use the drop-down listbox next to the optional *Typmod in function* field tab to select a type_modifier_input_function.

Use the drop-down listbox next to the optional *Typmod out function* field tab to select a type_modifier_output_function. It is allowed to omit the type_modifier_output_function, in which case the default display format is the stored typmod integer value enclosed in parentheses.

Use the optional *Internal length* to specify a value for internal representation.

Move the *Variable?* switch to specify the internal representation is of variable length (VARIABLE). The default is a fixed length positive integer.

Specify a default value in the optional *Default* field in cases where a column of the data type defaults to something other than the null value. Specify the default with the DEFAULT key word. (A default can be overridden by an explicit DEFAULT clause attached to a particular column.)

Use the drop-down listbox next to the optional *Analyze function* field to select a function for performing type-specific statistics collection for columns of the data type.

Use the drop-down listbox next to the optional *Category type* field to help control which implicit cast will be applied in ambiguous situations.

Move the *Preferred?* switch to Yes to specify the selected category type is preferred. The default is No.

On the *Optional-2* tab:

Use the drop-down listbox next to the optional *Element type* field to specify a data type.

Use the optional *Delimiter* field to indicate the delimiter to be used between values in the external representation of arrays for this data

Create - Type

General **Definition** Security SQL

Type	Range
Subtype	abstime
Subtype operator class	
Collation	
Canonical function	
Subtype diff function	

i **?** **x Cancel** **Reset** **Save**

type. The default delimiter is the comma (,). Note that the delimiter is associated with the array element type, not the array type itself.

Use the drop-down listbox next to *Alignment type* to specify the storage alignment required for the data type. The allowed values (char, int2, int4, and double) correspond with alignment on 1, 2, 4, or 8 byte boundaries.

Use the drop-down listbox next to optional *Storage type* to select a strategy for storing data.

Move the *Passed by value?* switch to *Yes* to override the existing data type value. The default is *No*.

Move the *Col atable?* switch to *Yes* to specify column definitions and expressions of the type may carry col ation information through use of the COLLATE clause. The default is *No*.

If you select *Range* in the *Type* field, the *Definition* tab displays the *Range* panel. Fields on the *Range* panel are context-sensitive and may be disabled.

Use the drop-down listbox next to *Sub-type* to select an associated b-tree operator class (to determine the ordering of values for the range type).

Use the drop-down listbox next to *Sub-type operator class* to use a non-default operator class.

Use the drop-down listbox next to *Col ation* to use a non-default col ation in the range's ordering if the sub-type is col atable.

Use the drop-down listbox next to *Canonical function* to convert range values to a canonical form.

Use the drop-down listbox next to *Sub-type diff function* to select a user-defined subtype_diff function.

If you select *Shel* in the *Type* field, the *Definition* tab displays the *Shel* panel:

A screenshot of a "Create - Type" dialog box. The title bar says "Create - Type" and has a close button. Below the title bar are tabs: "General", "Definition" (which is underlined, indicating it's selected), "Security", and "SQL".
The main area contains a "Type" field with a dropdown menu showing "Shell".
At the bottom are buttons: an "i" button, a "?" button, "Cancel" (with a cancel icon), "Reset" (with a reset/circular arrow icon), and a "Save" button (with a save disk icon).

Create - Type

General Definition **Security** SQL

Privileges

Grantee	Privileges	Grantor
<input type="button" value="Delete"/> enterpriseedb	U*	<input type="button" value="Delete"/> enterpriseedb

Security Labels

Provider	Security Label
<input type="button" value="Delete"/> my_provider	my_security

Buttons: Cancel Reset Save

A shell type is a placeholder for a type and has no parameters.

Click the **Security** tab to continue.

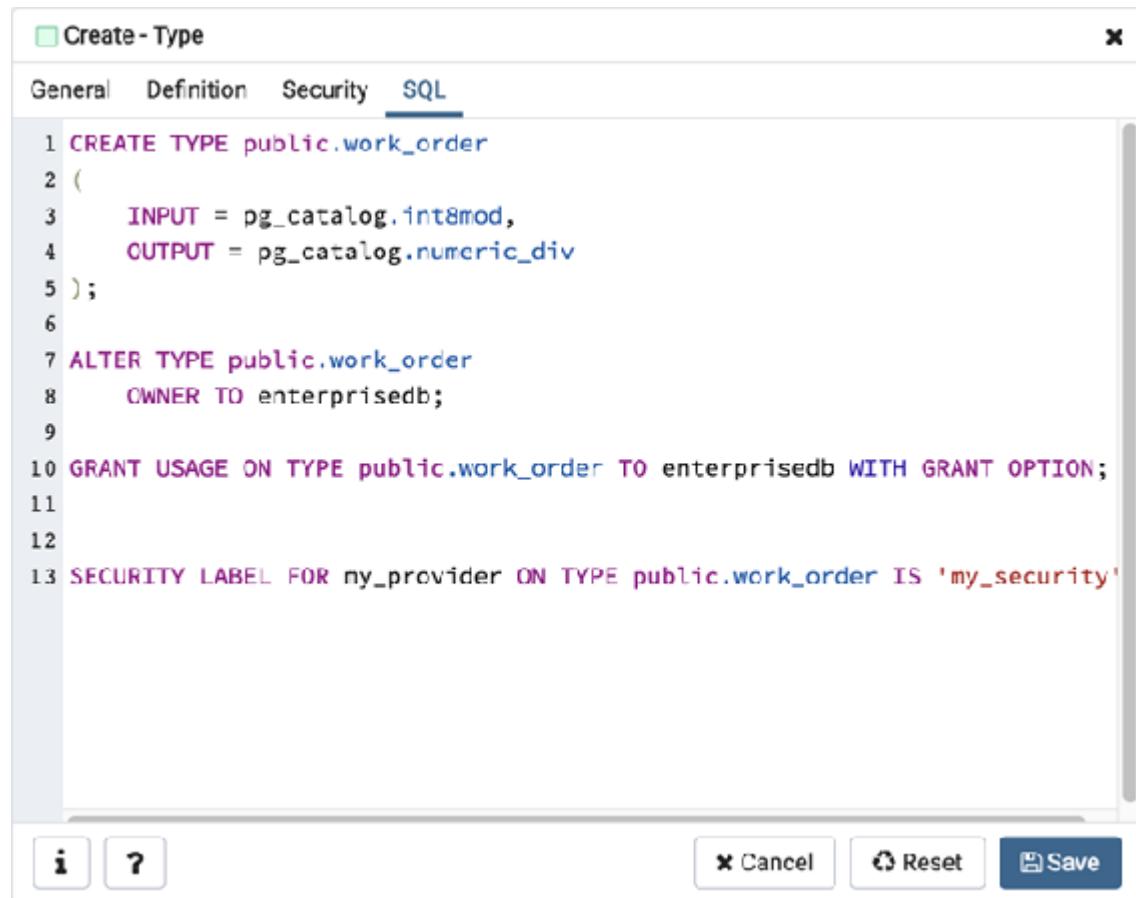
Use the **Security** tab to assign privileges and define security labels.

Use the **Privileges** panel to assign privileges for the type; click the **Add** icon (+) to grant privileges:

Select the name of the role that will be granted privileges on the type from the drop-down listbox in the **Grantee** field.

Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.



The screenshot shows a 'Create - Type' dialog box with the 'SQL' tab selected. The SQL code is as follows:

```
1 CREATE TYPE public.work_order
2 (
3     INPUT = pg_catalog.int8mod,
4     OUTPUT = pg_catalog.numeric_div
5 );
6
7 ALTER TYPE public.work_order
8     OWNER TO enterprisedb;
9
10 GRANT USAGE ON TYPE public.work_order TO enterprisedb WITH GRANT OPTION;
11
12
13 SECURITY LABEL FOR my_provider ON TYPE public.work_order IS 'my_security'
```

At the bottom of the dialog, there are buttons for 'Cancel', 'Reset', and 'Save'.

Click the *Add* icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Use the *Security Labels* panel to define security labels applied to the type.

Click the *Add* icon (+) to add each security label selection: Specify a security label provider in the *Provider* field. The named provider must be loaded and must consent to the proposed labeling operation.

Specify a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL

places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the *Add* icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

Your entries in the *Type* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of a sql command generated by user selections made in the *Type* dialog:

The example shown demonstrates creating a data type named *work_order*. The data type is an enumerated type with three labels: new, open and closed.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Welcome!

pgAdmin Sandbox

Difficulty: Beginner

Estimated Time: However long you like!

This is a sandbox in which you can experiment with pgAdmin 4 and PostgreSQL.

When you start, pgAdmin will be launched in a frame on the right hand side of the browser window. Follow the instructions to the left to login to pgAdmin, connect to PostgreSQL, and start exploring.

The **pagila** database (which is a port of the Sakila sample database which has been updated to be more Postgres idiomatic) is pre-loaded into the database server.

[Start Scenario](#)

Congratulations!

You've completed the scenario!

Scenario Rating



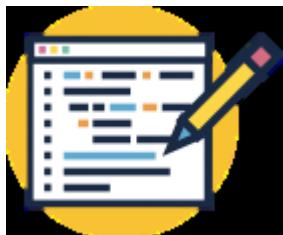
Your environment is currently being packaged as a Docker container and the download will begin shortly. To run the image locally, once Docker has been installed, use the commands

```
cat scrapbook enterprisepg pgadmin-sandbox container.tar | docker load  
docker run -it /enterprisepg_pgadmin-sandbox:
```

Oops!! Sorry, it looks like this scenario doesn't currently support downloads. We'll fix that shortly.

[Next Scenario](#)





Welcome!

pgAdmin Sandbox

Difficulty: Beginner

Estimated Time: However long you like!

This is a sandbox in which you can experiment with pgAdmin 4 and PostgreSQL.

When you start, pgAdmin will be launched in a frame on the right hand side of the browser window. Follow the instructions to the left to login to pgAdmin, connect to PostgreSQL, and start exploring.

The **pagila** database (which is a port of the Sakila sample database which has been updated to be more Postgres idiomatic) is pre-loaded into the database server.

Start Scenario

Congratulations!

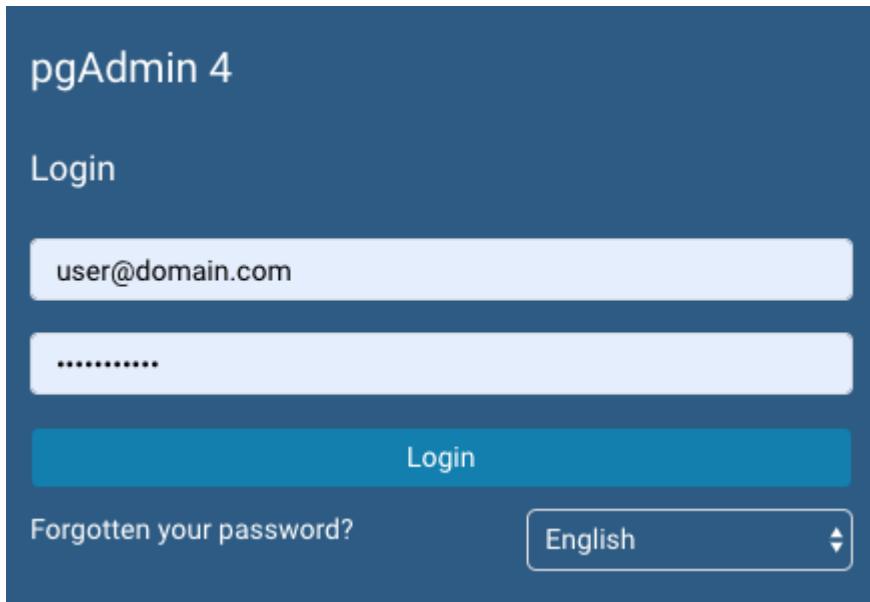
You've completed the scenario!

Scenario Rating

Your environment is currently being packaged as a Docker container and the download will begin shortly. To run the image locally, once Docker has been installed, use the commands
`cat scrapbook_enterprisedb_pgadmin-sandbox_container.tar | docker load`
`docker run -it /enterprisedb_pgadmin-sandbox:` Oops!! Sorry, it

looks like this scenario doesn't currently support downloads. We'll fix that shortly.

[Next Scenario](#)



Steps

pgAdmin Sandbox

Step 1 of 2

pgAdmin 4 Login

Use the **pgAdmin 4** frame to the right to experiment with pgAdmin. First, you need to login. The username and password are as follows:

Username: **user@domain.com**

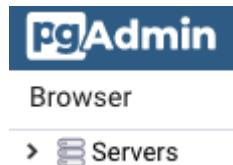
Password: **SuperSecret**

Once you have logged into pgAdmin, press the **Continue** button.

Continue

Connect to PostgreSQL and play!

Expand the **Servers** icon on the treeview to the left of the pgAdmin user interface. As there is only one server registered in this instance of pgAdmin, it will automatically try to connect by default.



You will be prompted for a password to connect to the server; enter **SuperSecret** and then click on the **OK** button.

Take some time to familiarise yourself with the pgAdmin user interface. You can find additional help on the **Help** menu of the application. If you have any questions, they can be asked on

[the pgAdmin Support mailing list.](#)

We hope you enjoy playing with PostgreSQL

and pgAdmin! If you wish to install your own copy for further experimentation or to start working with PostgreSQL or EDB Postgres

[Advanced Server, please visit: the pgAdmin download page.](#)

Web Page

pgAdmin 4

Welcome to

pgAdmin 4 v4.15

Debug Information

Content Version: ff30c51aa6ac9dc5638f49a0cbe3e6c9dea9bf4e
Last Updated: 2020-03-10T10:47:00.354Z

Connected - Katacoda Host: kitek08

host0 IP: 172.17.0.29

client1 IP: 172.17.0.35

OK

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) » [Creation or Modifying a Table](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Unique Constraint Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
 - [Check Dialog](#)
 - [Column Dialog](#)
 - [Compound Trigger Dialog](#)
 - [Exclusion Constraint Dialog](#)
 - [Foreign key Dialog](#)
 - [Index Dialog](#)
 - [Primary key Dialog](#)
 - [Rule Dialog](#)
 - [Table Dialog](#)
 - [Trigger Dialog](#)
 - [Unique Constraint Dialog](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Unique Constraint Dialog

Use the Unique constraint dialog to define a unique constraint for a specified table. Unique constraints ensure that the data contained in a column, or a group of columns, is unique among all the rows in the table.

The Unique constraint dialog organizes the development of a unique constraint through the following dialog tabs: General and Definition. The SQL tab displays the SQL code generated by dialog selections.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Creating or Modifying a Table](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Unique Constraint Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[□](#)

[Check Dialog](#)

[Column Dialog](#)

[Latest Docs](#)

[Compound Trigger Dialog](#)

[Exclusion Constraint Dialog](#)

[□](#)

[Foreign key Dialog](#)

[Index Dialog](#)

[Ask a](#)

[Primary key Dialog](#)

[Question](#)

[Rule Dialog](#)

[Table Dialog](#)

[□](#)

[Trigger Dialog](#)

[Report a Bug](#)

Unique Constraint Dialog

[Management Basics](#)

[Backup and Restore](#)



[Developer Tools](#)

[Screenshots](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Unique Constraint Dialog

Use the *Unique constraint* dialog to define a unique constraint for a specified table. Unique constraints ensure that the data contained in a column, or a group of columns, is unique among all the rows in the table.

The *Unique constraint* dialog organizes the development of a unique constraint through the following dialog tabs: *General* and *Definition*. The *SQL* tab displays the SQL code generated by dialog selections.

① Create - Unique constraint

X

General Definition SQL

Name

uc_book

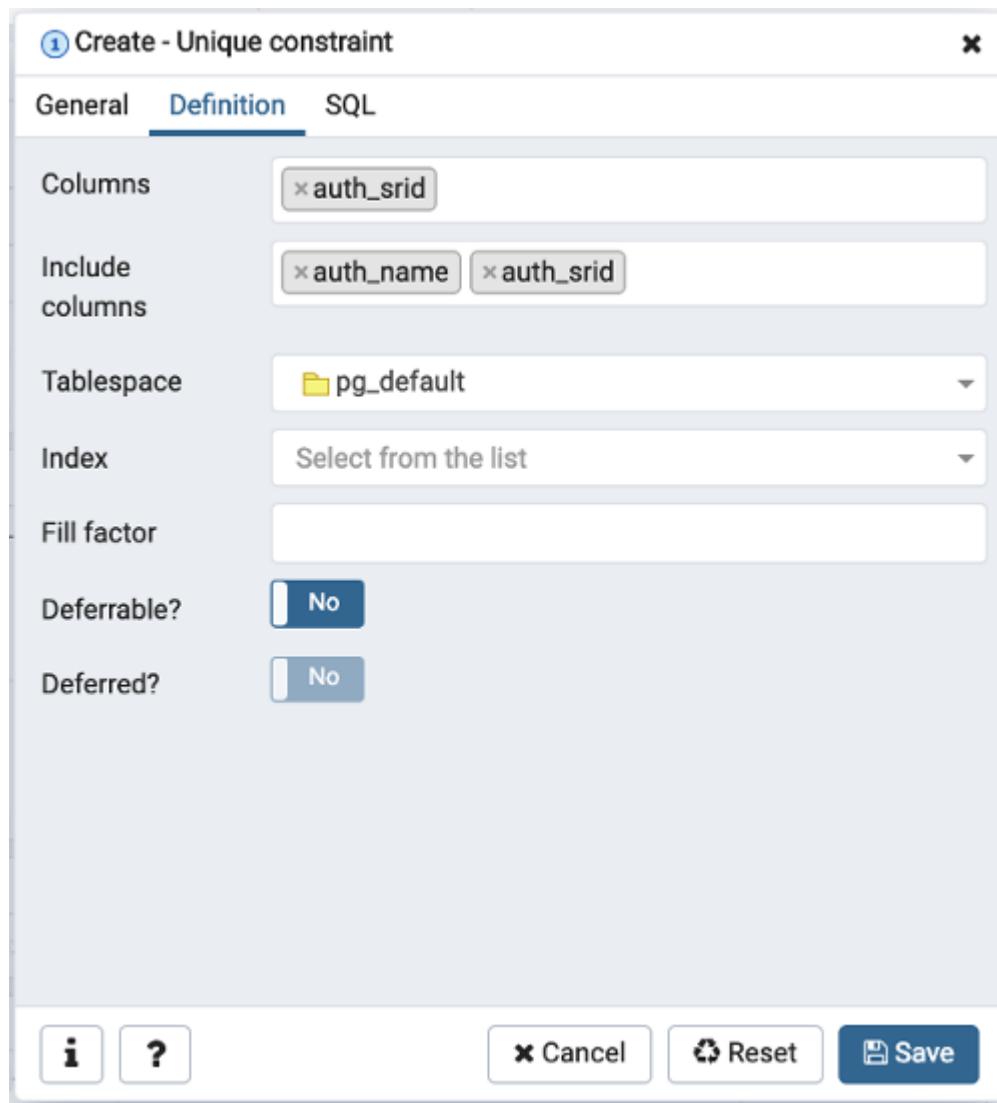
Comment



Cancel

Reset

Save



Use the fields in the *General* tab to identify the unique constraint:
Use the *Name* field to add a descriptive name for the unique constraint. The name will be displayed in the *pgAdmin* tree control.

Click the *Definition* tab to continue.

Use the fields in the *Definition* tab to define the unique constraint:
Click inside the *Columns* field and select one or more column names from the drop-down listbox. To delete a selection, click the x to the left of the column name. The unique constraint should be different from the primary key constraint defined for the same table; the selected column(s) for the constraints must be distinct.

① Create - Unique constraint

General Definition SQL

```
1 ALTER TABLE public.spatial_ref_sys
2   ADD CONSTRAINT uc_book UNIQUE (auth_srid)
3   INCLUDE(auth_name, auth_srid);
```

i **?** **Cancel** **Reset** **Save**

Use *Include columns* field to specify columns for *INCLUDE* clause of the constraint. This option is available in Postgres 11 and later.

Select the name of the tablespace in which the unique constraint will reside from the drop-down listbox in the *Tablespace* field.

Select the name of an index from the drop-down listbox in the *Index* field. This field is optional. Adding a unique constraint will

automatically create a unique B-tree index on the column or group of columns listed in the constraint, and will force the column(s) to be marked NOT NULL.

Use the *Fil Factor* field to specify a fil factor for the table and index.

The fil factor for a table is a percentage between 10 and 100. 100 (complete packing) is the default.

Move the *Deferrable?* switch to the *Yes* position to specify the timing of the constraint is deferrable and can be postponed until the end of the statement. The default is *No*.

If enabled, move the *Deferred?* switch to the *Yes* position to specify the timing of the constraint is deferred to the end of the statement.

The default is *No*.

Click the *SQL* tab to continue.

Your entries in the *Unique constraint* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Unique constraint* dialog:

The example shown demonstrates creating a unique constraint named *name_con* on the *name* column of the *distributors* table.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | Development

[pgAdmin 4 documentation](#) > [Getting Started](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
 - [Deployment](#)
 - [Login Dialog](#)
 - [User Management Dialog](#)
 - [Change User Password Dialog](#)
 - [User Interface](#)
 - [Menu Bar](#)
 - [Toolbar](#)
 - [Tabbed Browser](#)
 - [Tree Control](#)
 - [Preferences Dialog](#)
 - [Keyboard Shortcuts](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

User Interface

pgAdmin 4 supports all PostgreSQL features, from writing simple SQL queries to developing complex databases. It is designed to query an active database (in real-time), allowing you to stay current with modifications and implementations.

Features of pgAdmin 4 include:

- auto-detection and support for objects discovered at run-time
- a live SQL Query Tool with direct data editing
- support for administrative queries
- a syntax-highlighting SQL editor
- redesigned graphical interfaces
- powerful management dialogs and tools for common tasks
- responsive, context-sensitive behavior
- supportive error messages
- helpful hints
- online help and information about using pgAdmin dialogs and tools.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Deployment](#)

[Login Dialog](#)

[User Management Dialog](#)



[Change User Password Dialog](#)

[FAQ](#)

User Interface

[Menu Bar](#)



[Toolbar](#)

[Tabbed Browser](#)

[Latest Docs](#)

[Tree Control](#)

[Preferences Dialog](#)



[Keyboard Shortcuts](#)

[Connecting To A Server](#)

[Ask a](#)

[Managing Cluster Objects](#)

[Question](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Management Basics](#)

[Backup and Restore](#)

[Report a Bug](#)

[Developer Tools](#)

[pgAgent](#)



[pgAdmin Project Contributions](#)

[Release Notes](#)

[Screenshots](#)

[Licence](#)

User Interface

pgAdmin 4 supports all PostgreSQL features, from writing simple SQL

queries to developing complex databases. It is designed to query an active database (in real-time), allowing you to stay current with modifications and implementations.

Features of pgAdmin 4 include:

auto-detection and support for objects discovered at run-time

a live SQL Query Tool with direct data editing

support for administrative queries

a syntax-highlighting SQL editor

redesigned graphical interfaces

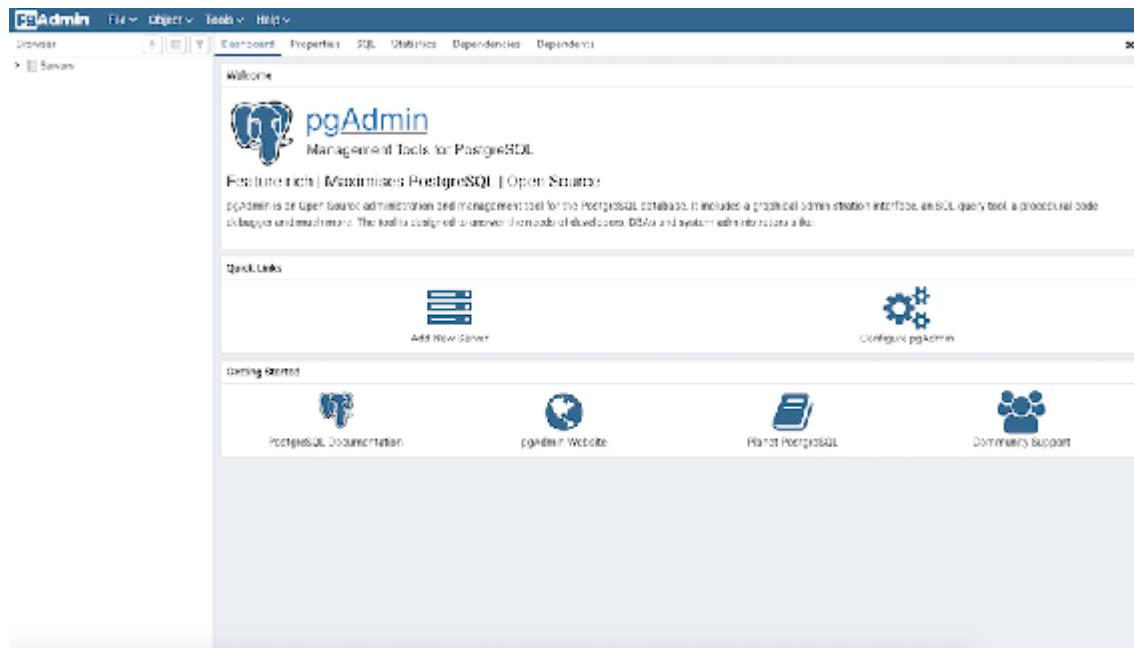
powerful management dialogs and tools for common tasks

responsive, context-sensitive behavior

supportive error messages

helpful hints

online help and information about using pgAdmin dialogs and tools.



When pgAdmin opens, the interface features a menu bar and a window divided into two panes: the *Browser* tree control in the left pane, and a tabbed browser in the right pane.

Select an icon from the *Quick Links* panel on the *Dashboard* tab to: Click the *Add New Server* button to open the [Create - Server dialog](#) to add a new server definition.

Click the *Configure pgAdmin* button to open the [Preferences dialog](#) to customize your pgAdmin client.

Links in the *Getting Started* panel open a new browser tab that provide useful information for Postgres users:

Click the *PostgreSQL Documentation* link to navigate to the *Documentation* page for the PostgreSQL open-source project; once at the project site, you can review the manuals for the currently supported versions of the PostgreSQL server.

Click the *pgAdmin Website* link to navigate to the pgAdmin project website. The pgAdmin site features news about recent pgAdmin releases and other project information.

Click the *Planet PostgreSQL* link to navigate to the blog aggregator for Postgres related blogs.

Click the *Community Support* link to navigate to the *Community* page at the PostgreSQL open-source project site; this page provides information about obtaining support for PostgreSQL features.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | Development

[pgAdmin 4 documentation](#) > [Getting Started](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
 - [Deployment](#)
 - [Login Dialog](#)
 - [User Management Dialog](#)
 - [Change User Password Dialog](#)
 - [User Interface](#)
 - [Menu Bar](#)
 - [Toolbar](#)
 - [Tabbed Browser](#)
 - [Tree Control](#)
 - [Preferences Dialog](#)
 - [Keyboard Shortcuts](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

User Management Dialog

When invoking pgAdmin in desktop mode, a password is randomly generated, and then ignored. If you install pgAdmin in server mode, you will be prompted for an administrator email and password for the pgAdmin client.

When you authenticate with pgAdmin, the server definitions associated with that login role are made available in the tree control. An administrative user can use the *User Management* dialog to

- add or delete pgAdmin roles
- assign privileges
- manage the password associated with a role

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Deployment](#)

[Login Dialog](#)

User Management Dialog



[Change User Password Dialog](#)

[FAQ](#)

[User Interface](#)

[Menu Bar](#)



[Toolbar](#)

[Tabbed Browser](#)

[Latest Docs](#)

[Tree Control](#)

[Preferences Dialog](#)



[Keyboard Shortcuts](#)

[Connecting To A Server](#)

[Ask a](#)

[Managing Cluster Objects](#)

[Question](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Management Basics](#)

[Backup and Restore](#)

[Report a Bug](#)

[Developer Tools](#)

[pgAgent](#)



[pgAdmin Project Contributions](#)

[Release Notes](#)

[Screenshots](#)

[Licence](#)

User Management Dialog

When invoking pgAdmin in desktop mode, a password is randomly generated, and then ignored. If you instal pgAdmin in server mode, you wil be prompted for an administrator email and password for the pgAdmin client.

When you authenticate with pgAdmin, the server definitions associated with that login role are made available in the tree control. An administrative user can use the *User Management* dialog to add or delete pgAdmin roles

assign privileges

manage the password associated with a role

User Management					
<input type="text"/> Search +					
Email	Role	Active	New password	Confirm password	
alihasha.narendra@enterprisedb.com	Administrator	Yes			
robert.vadra@enterprisedb.com	Administrator	Yes			

?
Close

User Management					
<input type="text"/> Search +					
Email	Role	Active	New password	Confirm password	
alihasha.narendra@enterprisedb.com	Administrator	Yes			
robert.vadra@enterprisedb.com	Administrator	Yes			
	User	Yes			

Administrator
User

?
Close

Use the *Filter by email* search field to find a user; enter a user's email address to find a user. If the user exists, the *User Management* table will display the user's current information.

To add a user, click *Add* to add new role.

Provide information about the new pgAdmin role in the row: Click in the *Email* field, and provide an email address for the user; this address will be used to recover the password associated with the role should the password be lost.

Use the drop-down listbox next to *Role* to select whether a user is an *Administrator* or a *User*.

Select *Administrator* if the user will have administrative privileges within the pgAdmin client.

Select *User* to create a non-administrative user account.

Move the *Active* switch to the *No* position if the account is not currently active; the default is *Yes*. Use this switch to disable account activity without deleting an account.

Use the *New password* field to provide the password associated with the user specified in the *Email* field.

Re-enter the password in the *Confirm password* field.

To discard a user, and revoke access to pgAdmin, click the trash icon to the left of the row and confirm deletion in the *Delete user?* dialog.

Users with the *Administrator* role are able to add, edit and remove pgAdmin users, but otherwise have the same capabilities as those with the *User* role.

Click the *Help* button (?) to access online help.

Click the *Close* button to save work. You will be prompted to return to the dialog if your selections cannot be saved.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [User Mapping Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

User Mapping Dialog 

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[User Mapping Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)



[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)



[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

User Mapping Dialog

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

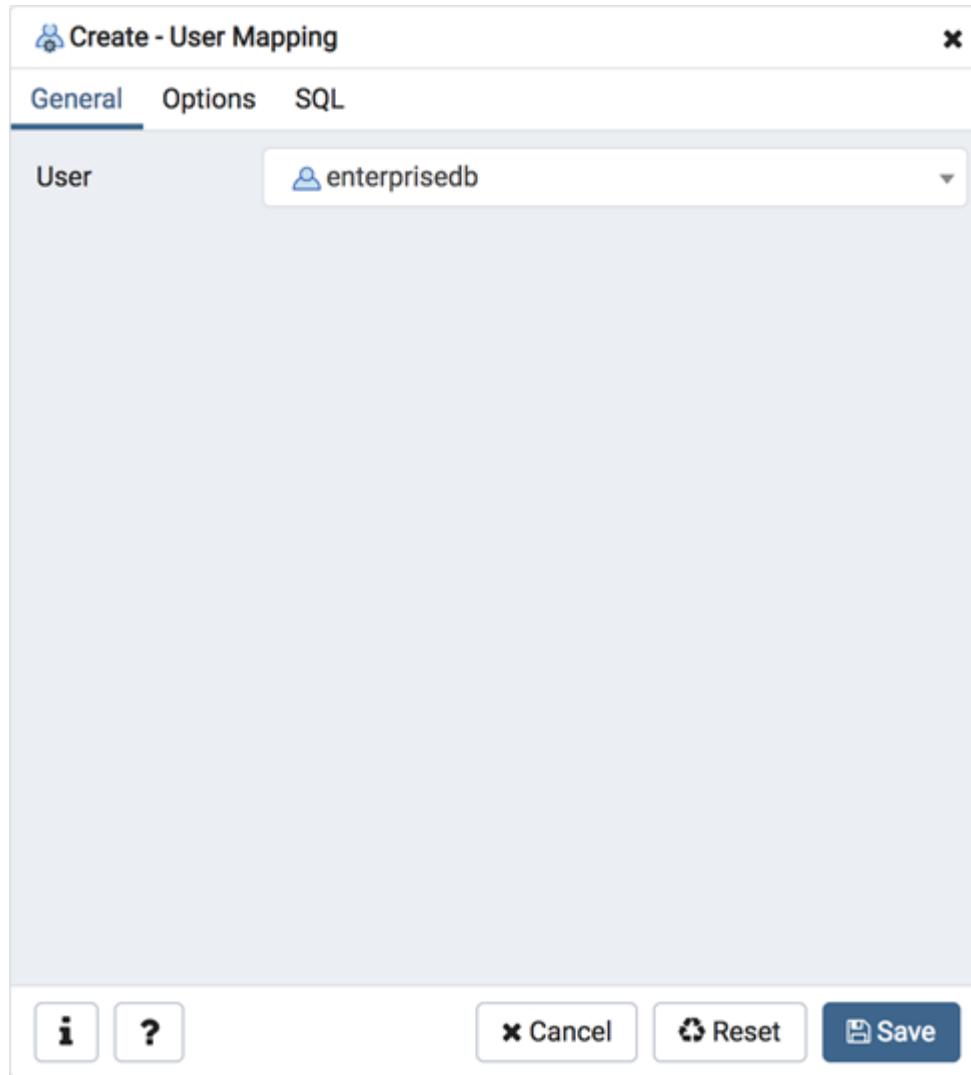
[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

User Mapping Dialog



Use the *User Mapping* dialog to define a new mapping of a user to a foreign server.

The *User Mapping* dialog organizes the development of a user mapping through the following dialog tabs: *General* and *Options*. The *SQL* tab displays the SQL code generated by dialog selections.

Use the drop-down listbox in the *User* field in the *General* tab to identify the connecting role:

Select `CURRENT_USER` to use the name of the current role.

Select `PUBLIC` if no other user-specific mapping is applicable.

Select a pre-defined role name to specify the name of an existing user.

Click the *Options* tab to continue.

The screenshot shows a software interface for creating a user mapping. At the top, there's a title bar with the text "Create - User Mapping" and a close button (X). Below the title bar, there are three tabs: "General", "Options" (which is underlined, indicating it's the active tab), and "SQL". The main area is titled "Options" and contains a table with two columns: "Options" and "Value". There is one entry in the table: "data_options" in the Options column and "data_value" in the Value column. At the bottom of the dialog box, there are several buttons: an information icon (i), a question mark icon (?), a "Cancel" button with a cancel symbol, a "Reset" button with a circular arrow symbol, and a "Save" button with a disk symbol.

Options	Value
data_options	data_value

Use the fields in the *Options* tab to specify connection options; the accepted option names and values are specific to the foreign data wrapper associated with the server specified in the user mapping. Click the *Add* button to add an option/value pair.

Specify the option name in the *Option* field.

Provide a corresponding value in the *Value* field.

Click *Add* to specify each additional option/value pair; to discard an option, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

Your entries in the *User Mapping* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example 

The following is an example of the sql command generated by user selections in the *User Mapping* dialog:

Create - User Mapping x

General Options **SQL**

```
1 CREATE USER MAPPING FOR enterprisedb SERVER hdfs
2   OPTIONS (data_options 'data_value');
3
```

i ? x Cancel ↻ Reset Save

The example shown demonstrates a user mapping for the *hdfs_server*.

The user is *CURRENT_USER* with a password *secret*.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [pgAgent](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Using pgAgent](#)
 - [Security Concerns](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
 - [Using pgAgent](#)
 - [Installing pgAgent](#)
 - [Creating a pgAgent Job](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Using pgAgent

pgAgent is a scheduling agent that runs and manages jobs; each job consists of one or more steps and schedules. If two or more jobs are scheduled to execute concurrently, pgAgent will execute the jobs in parallel (each with its own thread).

A step may be a series of SQL statements or an operating system batch/shell script. Each step in a given job is executed when the previous step completes, in alphanumeric order by name. Switches on the pgAgent Job dialog (accessed through the Properties context menu) allow you to modify a job, enabling or disabling individual steps as needed.

Each job is executed according to one or more schedules. Each time the job or any of its schedules are altered, the next runtime of the job is re-calculated. Each instance of pgAgent periodically polls the database for jobs with the next runtime value in the past. By polling at least once every minute, all jobs will normally start within one minute of the specified start time. If no pgAgent instance is running at the next runtime of a job, it will run as soon as pgAgent is next started, following which it will return to the normal schedule.

When you highlight the name of a defined job in the pgAdmin tree control, the Properties tab of the main pgAdmin window will display details about the job, and the Statistics tab will display details about the job's execution.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#)
[Development](#)

[pgAdmin 4 4.19 documentation](#) » [pgAgent](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Using pgAgent](#)

[Download](#)

[Security Concerns](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Management Basics](#)

[Backup and Restore](#)

[Latest Docs](#)

[Developer Tools](#)

[pgAgent](#)



Using pgAgent

[Installing pgAgent](#)

[Ask a](#)

[Creating a pgAgent Job](#)

[Question](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)



[Licence](#)

[Report a Bug](#)

Using pgAgent



pgAgent is a scheduling agent that runs and manages jobs; each job consists of one or more steps and schedules. If two or more jobs are

Screenshots

scheduled to execute concurrently, pgAgent wil execute the jobs in parallel (each with it's own thread).

A step may be a series of SQL statements or an operating system batch/shel script. Each step in a given job is executed when the previous step completes, in alphanumeric order by name. Switches on the *pgAgent Job* dialog (accessed through the *Properties* context menu) al low you to modify a job, enabling or disabling individual steps as needed.

Each job is executed according to one or more schedules. Each time the job or any of its schedules are altered, the next runtime of the job is re-calculated. Each instance of pgAgent periodical y pol s the database for jobs with the next runtime value in the past. By pol ing at least once every minute, al jobs wil normal y start within one minute of the specified start time. If no pgAgent instance is running at the next runtime of a job, it wil run as soon as pgAgent is next started, fol owing which it wil return to the normal schedule.

When you highlight the name of a defined job in the pgAdmin tree control, the *Properties* tab of the main pgAdmin window wil display details about the job, and the *Statistics* tab wil display details about the job's execution.

Security Concerns

pgAgent is a very powerful tool, but does have some security considerations that you should be aware of:

Database password - *DO NOT* be tempted to include a password in the pgAgent connection string - on Unix systems it may be visible to al users in *ps* output, and on Windows systems it wil be stored in the registry in plain text. Instead, use a libpq *~/.pgpass* file to store the passwords for every database that pgAgent must access. Details of

this technique may be found in the [PostgreSQL documentation on .pgpass file](#).

System/database access - all jobs run by pgAgent will run with the security privileges of the pgAgent user. SQL steps will run as the user that pgAgent connects to the database as, and batch/shell scripts will run as the operating system user that the pgAgent service or daemon is running under. Because of this, it is essential to maintain control over the users that are able to create and modify jobs. By default, only the user that created the pgAgent database objects will be able to do this - this will normally be the PostgreSQL superuser.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [View Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

[View Dialog](#)

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[View Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)



[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)



[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

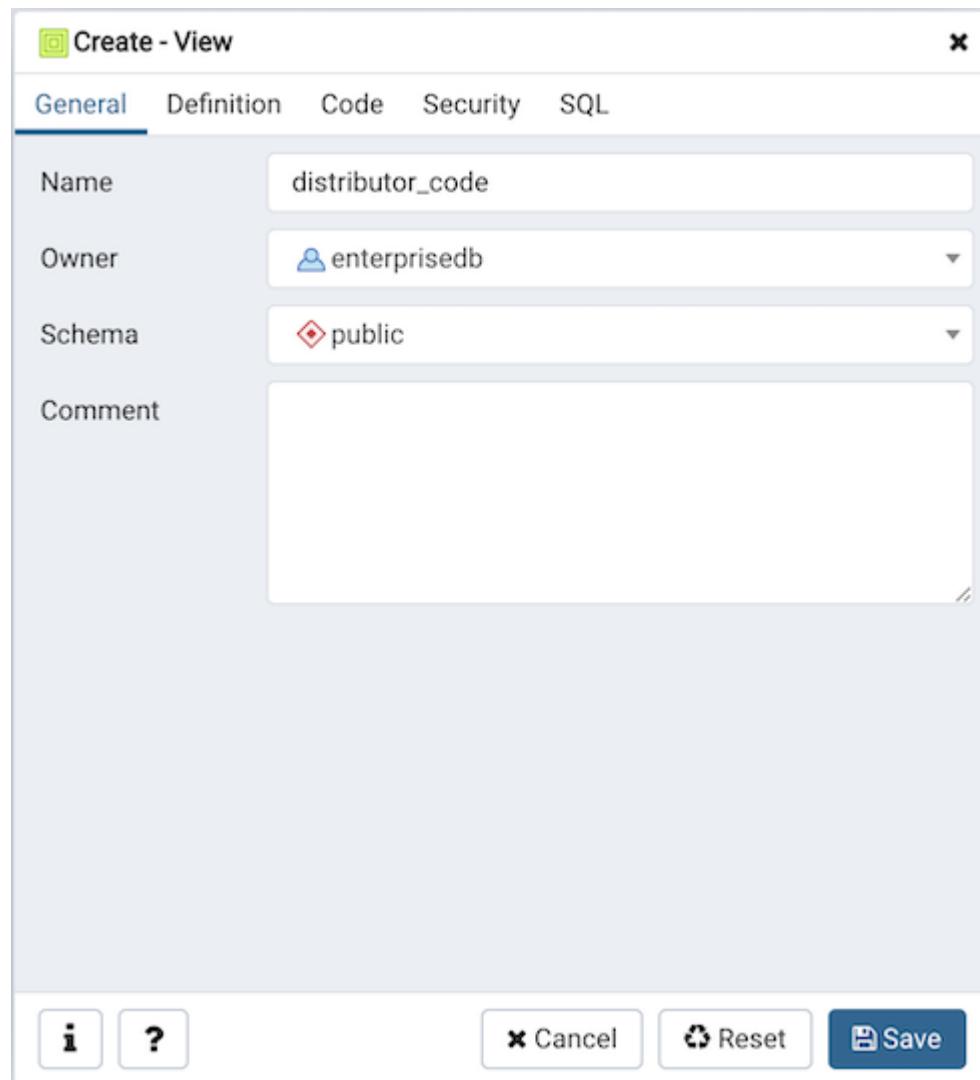
[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

View Dialog 



The screenshot shows the 'Create - View' dialog box. The 'General' tab is selected. The 'Name' field contains 'distributor_code'. The 'Owner' dropdown shows 'enterprisedb'. The 'Schema' dropdown shows 'public'. The 'Comment' area is empty. At the bottom are buttons for 'Cancel', 'Reset', and 'Save'.

Use the *View* dialog to define a view. The view is not physical y materialized; the query is executed each time the view is referenced in a query.

The *View* dialog organizes the development of a View through the following dialog tabs: *General*, *Definition*, *Code* and *Security*". The *SQL* tab displays the SQL code generated by dialog selections.

Click the *General* tab to begin.

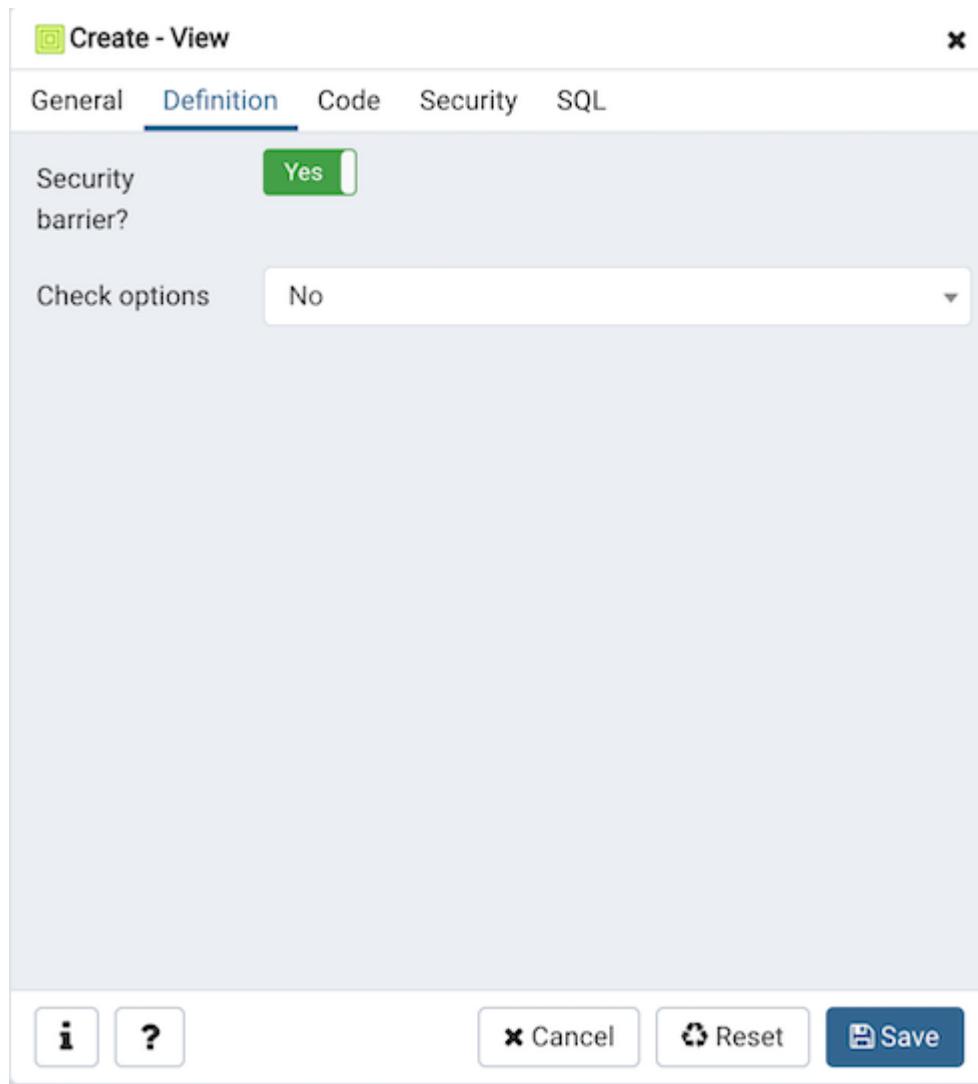
Use the fields in the *General* tab to identify a view: Use the *Name* field to add a descriptive name for the view. The name of the view must be distinct from the name of any other view, table, sequence, index or foreign table in the same schema. The name will be displayed in the *pgAdmin* tree control.

Use the drop-down listbox next to *Owner* to select the role that will own the view.

If applicable, select the name of the schema in which the view will reside from the drop-down listbox in the *Schema* field.

Store notes about the view in the *Comments* field.

Click the *Definition* tab to continue.



Use the fields in the *Definition* tab to define properties of the view: Set the *Security Barrier* switch to *Yes* to indicate that the view is to act as a security barrier. For more information about defining and using a security barrier rule, see Section 38.5 of the PostgreSQL documentation.

Use the drop-down listbox next to *Check options* to select from *No*, *Local* or *Cascaded*:

The *Local* option specifies that new rows are only checked against the conditions defined in the view. Any conditions defined on

underlying base views are not checked (unless you specify the CHECK OPTION).

The *Cascaded* option specifies new rows are checked against the conditions of the view and all underlying base views.

Click the *Code* tab to continue.

distributor_code

General Definition **Code** Security SQL

```
1 SELECT * FROM pg_class;
```

Cancel Reset Save

Create - View

General Definition Code **Security** SQL

Privileges

	Grantee	Privileges	Grantor
	enterpriseedb	a*r*w*d*D*x*t*	enterpriseedb

Security labels

Provider	Security label

i **?** **x Cancel** **Reset** **Save**

The screenshot shows the 'Create - View' dialog in DBeaver. The 'Security' tab is selected. The 'Privileges' section contains one row: a grantee named 'enterpriseedb' with privileges 'a*r*w*d*D*x*t*' granted by 'enterpriseedb'. The 'Security labels' section is empty. At the bottom are buttons for information, help, cancel, reset, and save.

Use the workspace in the *Code* tab to write a query to create a view.

Click the *Security* tab to continue.

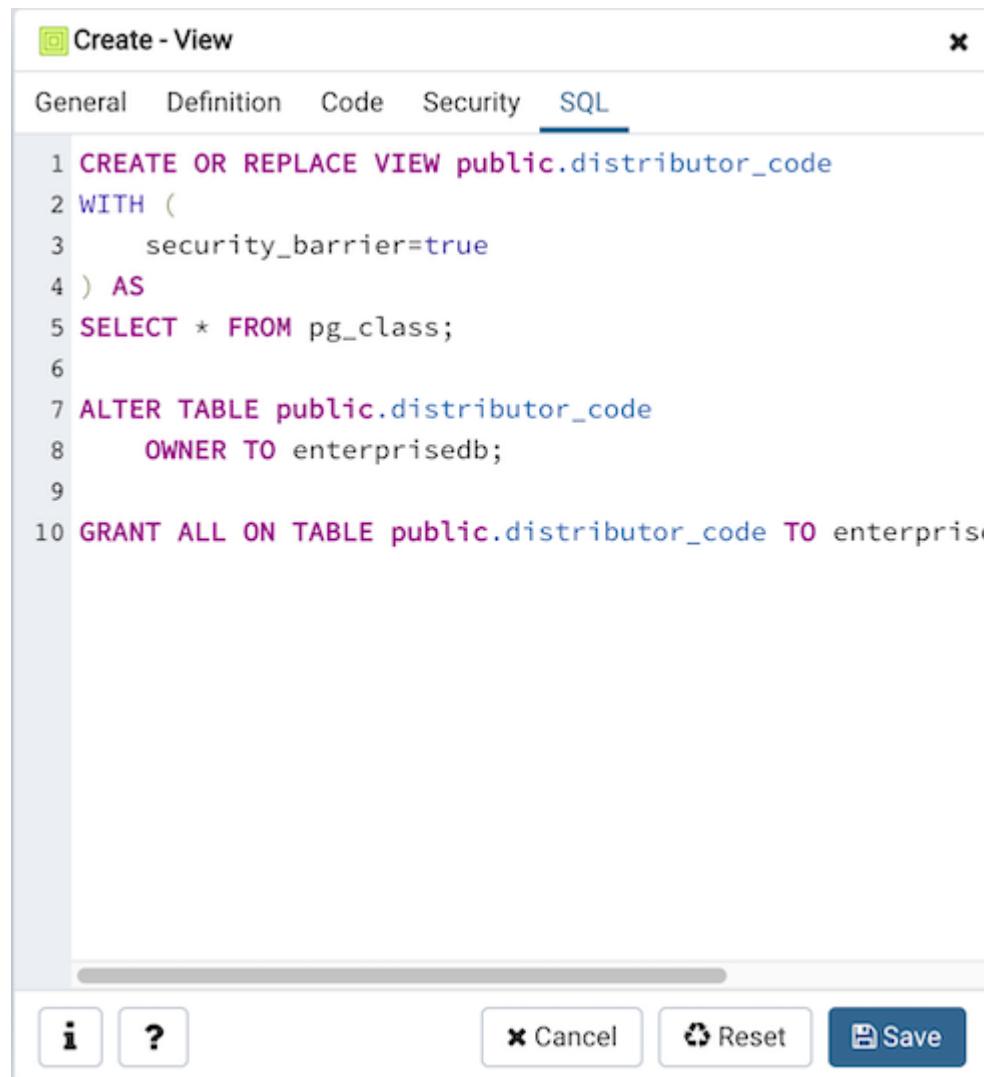
Use the *Security* tab to assign privileges and define security labels.

Use the *Privileges* panel to assign privileges to a role. Click the *Add* icon (+) to set privileges for the view:

Select the name of the role that will be granted privileges from the drop-down listbox in the *Grantee* field.

Click inside the *Privileges* field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.



The screenshot shows a 'Create - View' dialog box with the 'SQL' tab selected. The SQL code is as follows:

```
1 CREATE OR REPLACE VIEW public.distributor_code
2 WITH (
3     security_barrier=true
4 ) AS
5 SELECT * FROM pg_class;
6
7 ALTER TABLE public.distributor_code
8     OWNER TO enterprise;
9
10 GRANT ALL ON TABLE public.distributor_code TO enterprise;
```

At the bottom of the dialog are buttons for 'Cancel', 'Reset', and 'Save'.

Click the *Add* icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Use the *Security Labels* panel to define security labels applied to the view. Click the *Add* icon (+) to add each security label selection: Specify a security label provider in the *Provider* field. The named

provider must be loaded and must consent to the proposed labeling operation.

Specify a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL

places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the *Add* icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

Your entries in the *View* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *View* dialog:

The example shown demonstrates creating a view named *distributor_code*.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | Development

[pgAdmin 4 documentation](#) > [Developer Tools](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [View/Edit Data](#)
 - [The View/Edit Data Toolbar](#)
 - [The Data Grid](#)
 - [Sort/Filter options dialog](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
 - [Debugger](#)
 - [Query Tool](#)
 - [View/Edit Data](#)
 - [Schema Diff](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

View/Edit Data

To view or modify data, right click on a table or view name in the Browser tree control. When the context menu opens, use the *View/Edit Data* menu to specify the number of rows you would like to display in the editor panel.

To modify the content of a table, each row in the table must be uniquely identifiable. If the table definition does not include an OID or a primary key, the displayed data is read only. Note that views cannot be edited.

The screenshot shows the pgAdmin 4 interface. The top bar has tabs for 'Query Editor' and 'Scratch Pad'. The main area contains a query editor with the following SQL code:

```
1 SELECT * FROM public.coffee_shops
```

Below the query editor is a results grid titled 'Data Output' with columns: id, name, address, city, state, zip, lat, lon, and geometry. The data consists of 14 rows of coffee shop information.

	id	name	address	city	state	zip	lat	lon	geometry
1	04	Aja Grill-Cambridge	1354 Massachusetts Ave	Cambridge	MA	02138	-71.11094	42.366100	POINT(-71.11094 42.366100)
2	1	Coffee House	1354 Cambridge St	Cambridge	MA	02141	-71.10444	42.366100	POINT(-71.10444 42.366100)
3	2	100 Coffee House	759 Massachusetts Ave	Cambridge	MA	02139	-71.10543	42.366100	POINT(-71.10543 42.366100)
4	3	Americafats	625 Brattle St	Cambridge	MA	02142	-71.08192	42.366100	POINT(-71.08192 42.366100)
5	4	Aja Grill-Cambridge	1354 Massachusetts Ave	Cambridge	MA	02138	-71.11094	42.366100	POINT(-71.11094 42.366100)
6	5	Ajzen's Coffeeshop	625 Brattle St	Cambridge	MA	02142	-71.12138	42.366100	POINT(-71.12138 42.366100)
7	6	Aero Four	500 Technology Sq	Cambridge	MA	02139	-71.06128	42.366100	POINT(-71.06128 42.366100)
8	7	B & B Beach Bar	50 Brattle	Cambridge	MA	02142	-71.06002	42.366100	POINT(-71.06002 42.366100)
9	8	Basha Cafe	26 New St	Cambridge	MA	02138	-71.14082	42.366100	POINT(-71.14082 42.366100)
10	9	Bear Creek Coffee Roaster	1 Kendall St # 501-35	Cambridge	MA	02139	-71.09088	42.366100	POINT(-71.09088 42.366100)
11	10	Benton's Tea Shop	54 Brattle St	Cambridge	MA	02138	-71.12962	42.366100	POINT(-71.12962 42.366100)
12	11	Broadway Marketplace LLC	450 Brattle	Cambridge	MA	02139	-71.11363	42.366100	POINT(-71.11363 42.366100)
13	12	Caribou Java	100 Brattle St	Cambridge	MA	02138	-71.12738	42.366100	POINT(-71.12738 42.366100)
14	13	Cafe Panorama	12 Dow St	Cambridge	MA	02138	-71.11527	42.366100	POINT(-71.11527 42.366100)

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Developer Tools](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[View/Edit Data](#)

[Download](#)

[The View/Edit Data Toolbar](#)

[The Data Grid](#)

[Sort/Filter options dialog](#)



[Getting Started](#)

[FAQ](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)



[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[Latest Docs](#)

[Management Basics](#)

[Backup and Restore](#)



[Developer Tools](#)

[Debugger](#)

[Ask a](#)

[Query Tool](#)

Question

[View/Edit Data](#)

Schema Diff



[pgAgent](#)

Report a Bug

[pgAdmin Project Contributions](#)

Release Notes

Licence



[View/Edit Data](#) 

Screenshots

To view or modify data, right click on a table or view name in the *Browser* tree control. When the context menu opens, use the *View/Edit Data* menu to specify the number of rows you would like to display in the editor panel.

To modify the content of a table, each row in the table must be uniquely identifiable. If the table definition does not include an OID or a primary key, the displayed data is read only. Note that views cannot be edited;

updatable views (using rules) are not supported.

The editor features a toolbar that allows quick access to frequently used options, and a work environment divided into two panels: The

upper panel displays the SQL command that was used to select the content displayed in the lower panel.

The lower panel (the Data Grid) displays the data selected from the table or view.

The View/Edit Data Toolbar

The [Query Tool](#) and [View/Edit Data](#) tools are actually different operating modes of the same tool. Some controls will be disabled in either mode.

Please see [The Query Tool Toolbar](#) for a description of the available controls.

The Data Grid

The top row of the data grid displays the name of each column, the data type, and if applicable, the number of characters allowed. A column that is part of the primary key will additionally be marked with [PK].

To modify the displayed data:

To change a numeric value within the grid, double-click the value to select the field. Modify the content in the square in which it is displayed.

To change a non-numeric value within the grid, double-click the content to access the edit bubble. After modifying the content of the edit bubble, click the *Ok* button to display your changes in the data grid, or *Cancel* to exit the edit bubble without saving.

To enter a newline character, click Ctrl-Enter or Shift-Enter. Newline formatting is only displayed when the field content is accessed via an edit bubble.

To add a new row to the table, enter data into the last (unnumbered) row of the table. As soon as you store the data, the row is assigned

a row number, and a fresh empty line is added to the data grid.

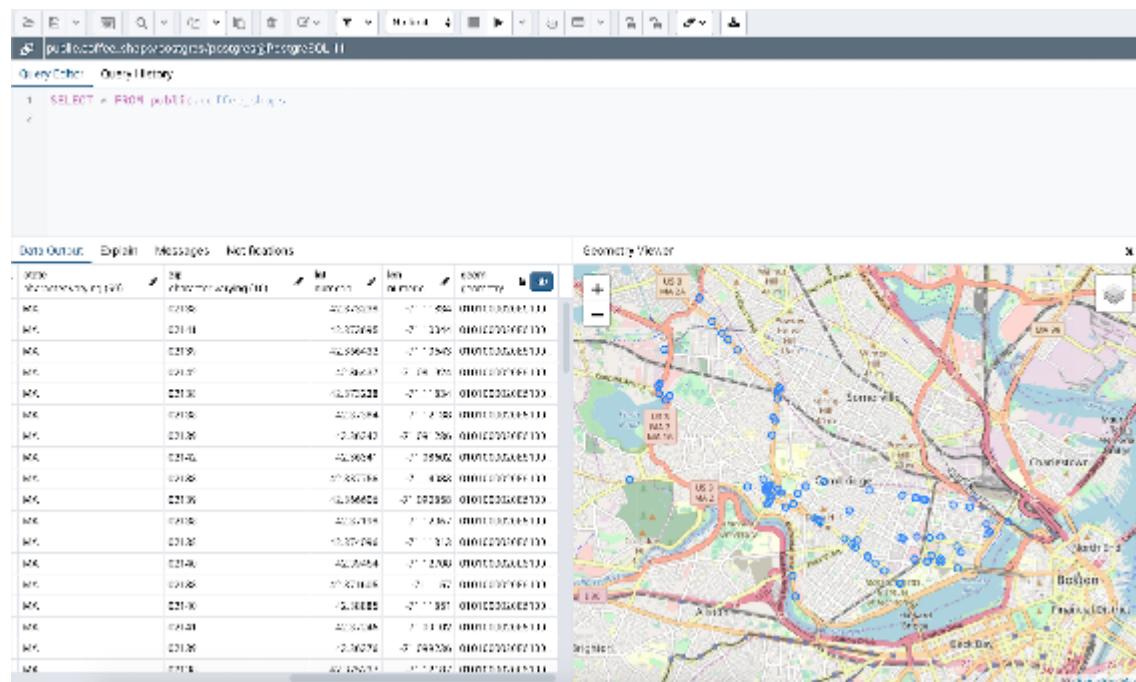
To write a SQL NULL to the table, simply leave the field empty. When you store the new row, the wil server fil in the default value for that column. If you store a change to an existing row, the value NULL wil explicitly be written.

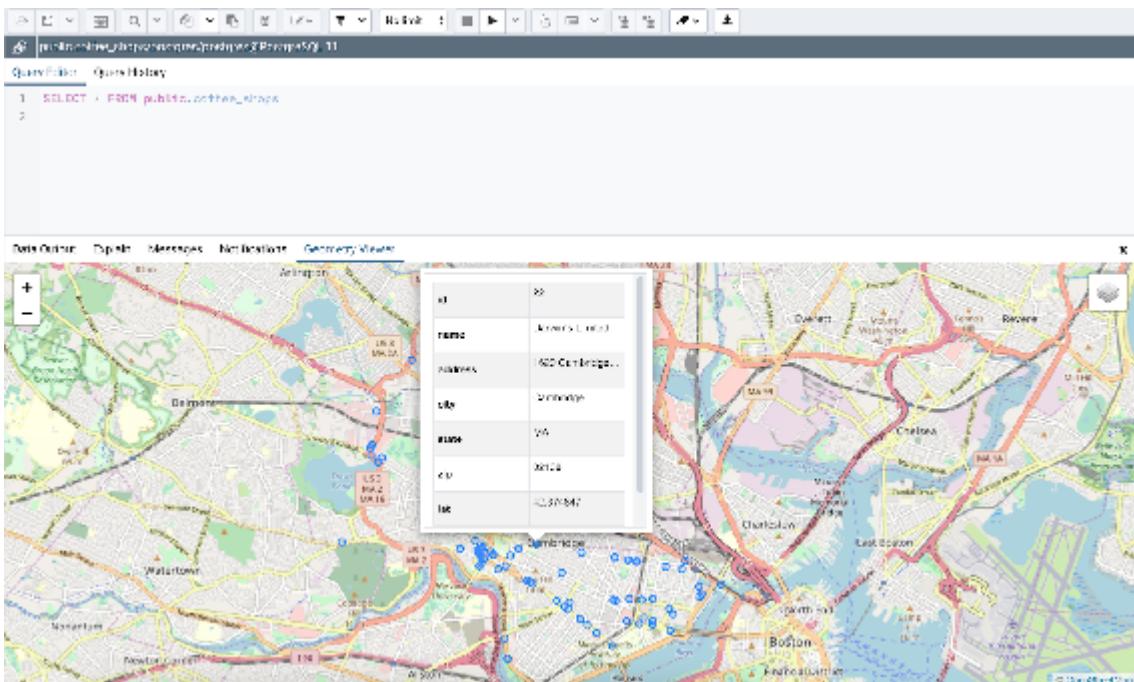
To write an empty string to the table, enter the special string “ (two single quotes) in the field. If you want to write a string containing solely two single quotes to the table, you need to escape these quotes, by typing “

To delete a row, press the *Delete* toolbar button. A popup will open, asking you to confirm the deletion.

To commit the changes to the server, select the *Save Data* toolbar button.

Geometry Data Viewer





If PostGIS is installed, you can view GIS objects in a map by selecting row(s) and clicking the 'View Geometry' button in the column. If no rows are selected, the entire data set will be rendered:

You can adjust the layout by dragging the title of the panel. To view the properties of the geometries directly in map, just click the specific geometry:

Note

Supported data types: The Geometry Viewer supports 2D and 3DM geometries in EWKB format including *Point*, *LineString*, *Polygon*, *MultiPoint*, *MultiLineString*, *MultiPolygon* and *GeometryCollection*.

SRIDs: If there are geometries with different SRIDs in the same column, the viewer will render geometries with the same SRID in the map. If SRID=4326 the OSM tile layer will be added into the map.

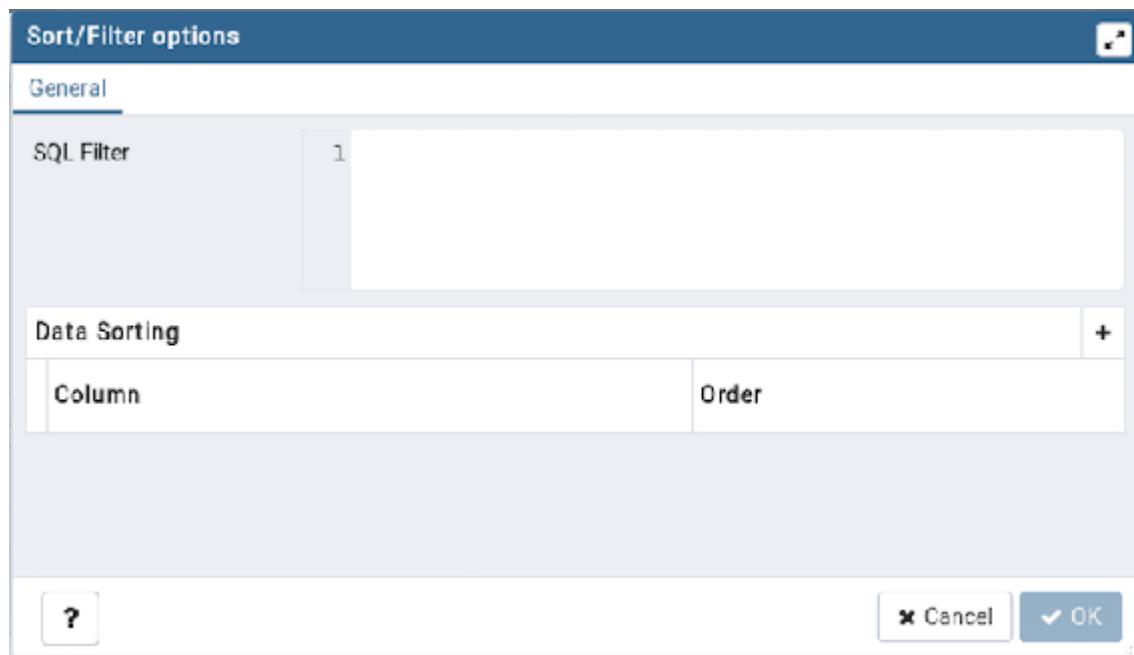
Data size: For performance reasons, the viewer will render no more than 100000 geometries, totaling up to 20MB.

Internet access: An internet connection is required for the Geometry Viewer to function correctly.

Sort/Filter options dialog

You can access *Sort/Filter options dialog* by clicking on Sort/Filter button.

This allows you to specify an SQL Filter to limit the data displayed and data sorting options in the edit grid window:



Use *SQL Filter* to provide SQL filtering criteria. These will be added to the “WHERE” clause of the query used to retrieve the data. For example, you might enter:

`id > 25 AND created > '2018-01-01'`

Use *Data Sorting* to sort the data in the output grid To add new column(s) in data sorting grid, click on the [+] icon.

Use the drop-down *Column* to select the column you want to sort.

Use the drop-down *Order* to select the sort order for the column.

To delete a row from the grid, click the trash icon.

Click the *Help* button (?) to access online help.

Click the *Ok* button to save work.

Click the *Close* button to discard current changes and close the dialog.

[View/Edit Data Filter](#)

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 4.19 documentation](#) »

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
 - [Backup Dialog](#)
 - [Backup Globals Dialog](#)
 - [Backup Server Dialog](#)
 - [Restore Dialog](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Backup and Restore

A powerful, but user-friendly Backup and Restore tool provides an easy way to use `pg_dump`, `pg_dumpall`, and `pg_restore` to take backups and create copies of databases or database objects for use in a development environment.

- [Backup Dialog](#)
- [Backup Globals Dialog](#)
- [Backup Server Dialog](#)
- [Restore Dialog](#)

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#)
[Development](#)

[pgAdmin 4 4.19 documentation](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[FAQ](#)

[Management Basics](#)

Backup and Restore



[Backup Dialog](#)

[Backup Globals Dialog](#)

[Latest Docs](#)

[Backup Server Dialog](#)

[Restore Dialog](#)



[Developer Tools](#)

[pgAgent](#)

[Ask a](#)

[pgAdmin Project Contributions](#)

[Question](#)

[Release Notes](#)

[Licence](#)



Backup and Restore

[Report a Bug](#)

A powerful, but user-friendly Backup and Restore tool provides an easy



way to use pg_dump, pg_dumpal , and pg_restore to take backups and create copies of databases or database objects for use in a development

[Screenshots](#)

environment.

[Backup Dialog](#)

[Backup Globals Dialog](#)

[Backup Server Dialog](#)

[Restore Dialog](#)

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Cast Dialog

Use the Cast dialog to define a cast. A cast specifies how to convert a value from one data type to another.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



Cast Dialog

[FAQ](#)

[Collation Dialog](#)

[Domain Dialog](#)

[□](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)

[Latest Docs](#)

[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[□](#)

[Foreign Server Dialog](#)

[Foreign Table Dialog](#)

[Ask a](#)

[FTS Configuration Dialog](#)

[Question](#)

[FTS Dictionary Dialog](#)

[FTS Parser Dialog](#)

[□](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[Report a Bug](#)

[Language Dialog](#)

[Materialized View Dialog](#)

[□](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Screenshots](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

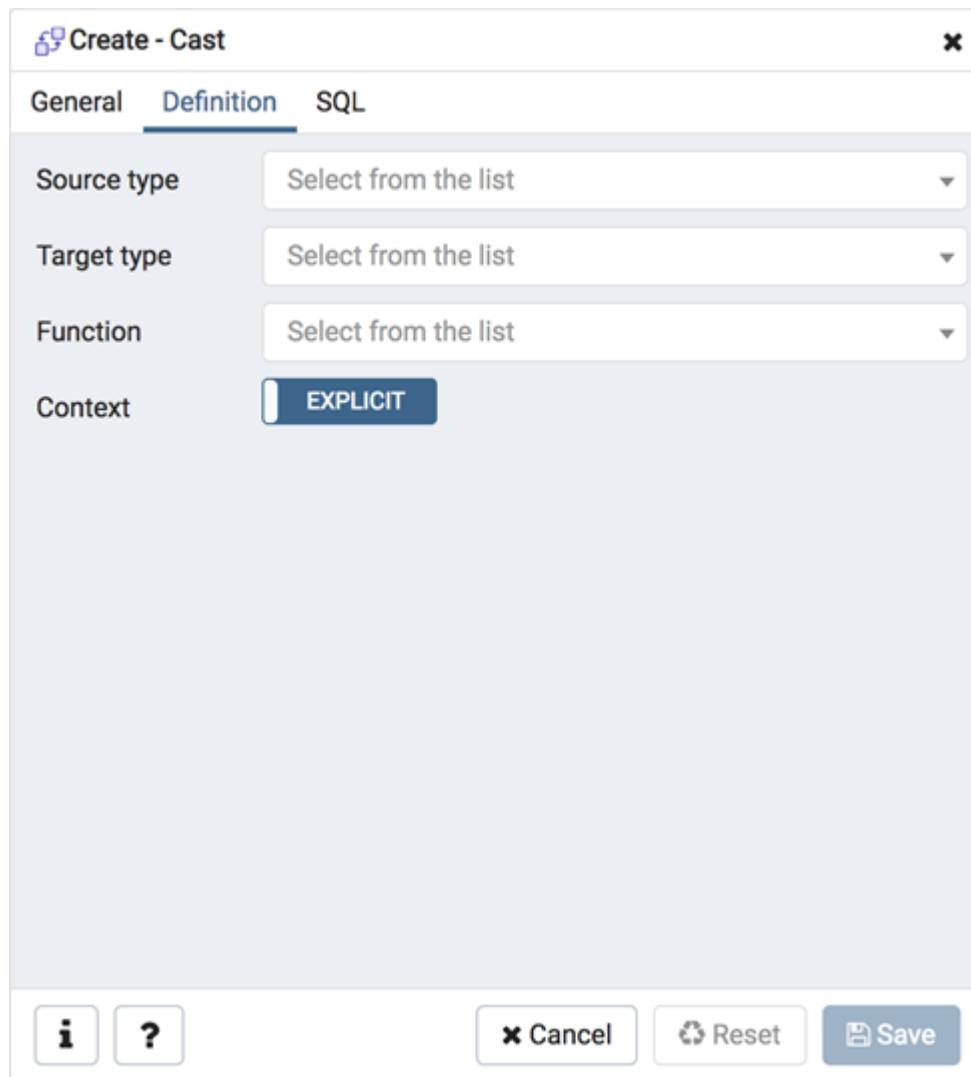
[Release Notes](#)

[Licence](#)

[Cast Dialog](#) 

Use the *Cast* dialog to define a cast. A cast specifies how to convert a value from one data type to another.





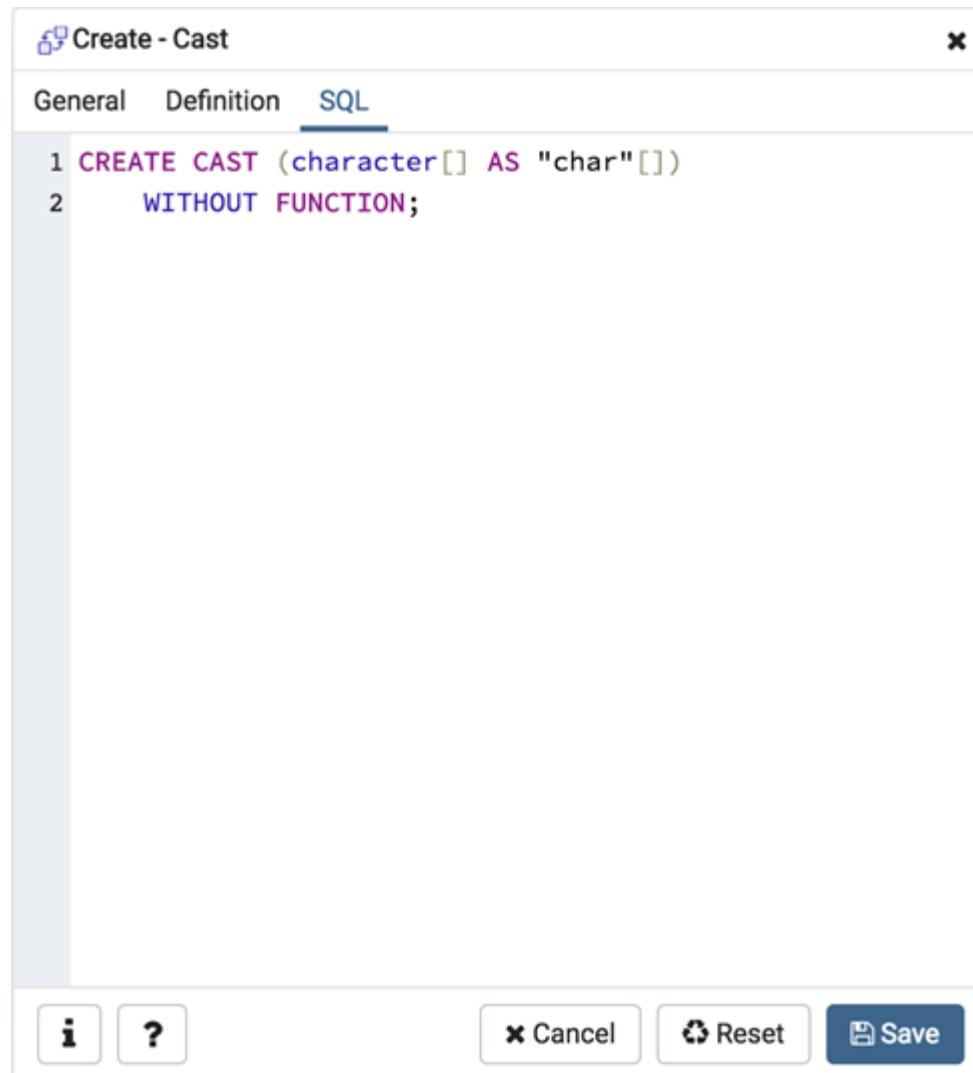
The *Cast* dialog organizes the development of a cast through the following dialog tabs: *General* and *Definition*. The *SQL* tab displays the SQL code generated by dialog selections.

Use the fields in the *General* tab to identify the cast: The *Name* field is disabled. The name that will be displayed in the *pgAdmin* tree control is the *Source* type concatenated with the *Target* type, and is generated automatically when you make selections on the *Cast* dialog *Definition* tab.

Store notes about the cast in the *Comment* field.

Click the *Definition* tab to continue.

Use the fields in the *Definition* tab to define parameters:



Use the drop-down listbox next to *Source type* to select the name of the source data type of the cast.

Use the drop-down listbox next to *Target type* to select the name of the target data type of the cast.

Use the drop-down listbox next to *Function* to select the function used to perform the cast. The function's result data type must match the target type of the cast.

Move the *Context* switch to the *Implicit* position if the cast is implicit.

By default, a cast can be invoked only by an explicit cast request. If the cast is marked *Implicit* then it can be invoked implicitly in any context, whether by assignment or internally in an expression.

Click the **SQL** tab to continue.

Your entries in the *Cast* dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Cast* dialog:

The cast uses a function named *int4(bigint)* to convert a bigint data type to an integer.

Click the *Info* button (i) to access online help.

Click the **Save** button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Backup and Restore](#) >

[previous](#) | [next](#) | [index](#)

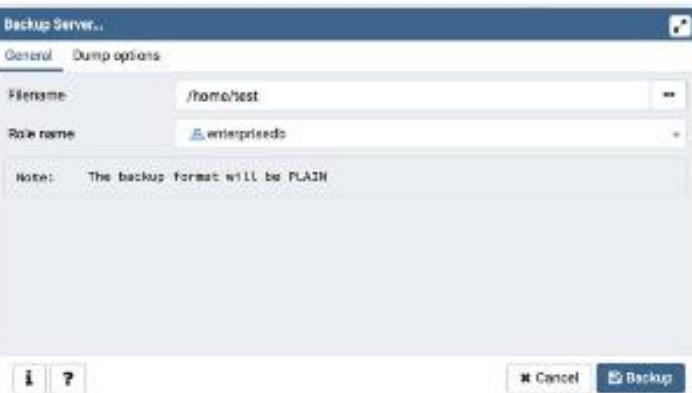
Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
 - [Backup Dialog](#)
 - [Backup Globals Dialog](#)
 - [Backup Server Dialog](#)
 - [Restore Dialog](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Backup Server Dialog

Use the *Backup Server* dialog to create a plain-text script that will recreate the selected server. You can use the pgAdmin Query Tool to play back a plain-text script, and recreate the server.



Use the fields in the General tab to specify the following:

- Enter the name of the backup file in the *Filename* field. Optionally, select the *Browser* icon (ellipsis) to the right to navigate into a directory and select a file that will contain the archive.
- Use the *Encoding* drop-down listbox to select the character encoding method that should be used for the archive. **Note:** This option is visible only for database server greater than or equal to 11.



Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Backup and Restore](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[FAQ](#)

[Management Basics](#)

[Backup and Restore](#)



[Backup Dialog](#)

[Backup Globals Dialog](#)

[Latest Docs](#)

Backup Server Dialog

[Restore Dialog](#)



[Developer Tools](#)

[pgAgent](#)

[Ask a](#)

[pgAdmin Project Contributions](#)

[Question](#)

[Release Notes](#)

[Licence](#)



Backup Server Dialog 

[Report a Bug](#)

Use the *Backup Server* dialog to create a plain-text script that wil

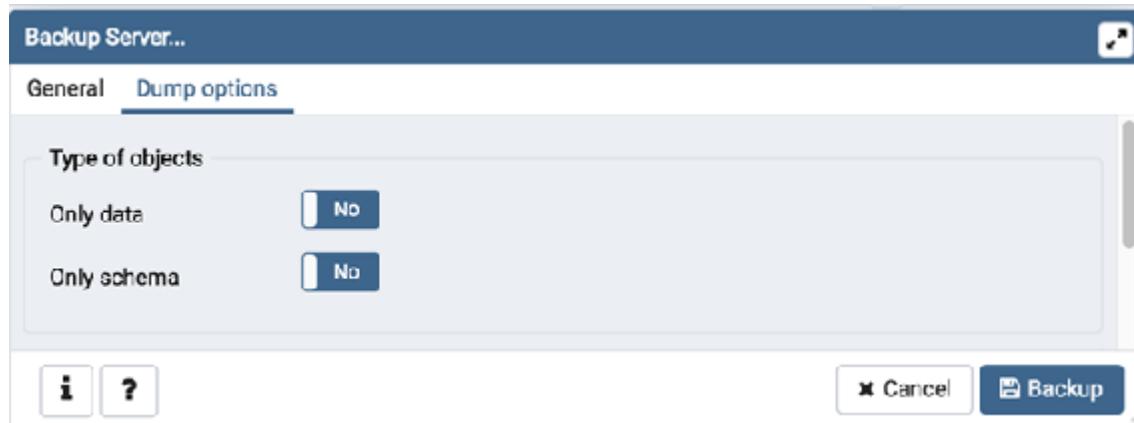


recreate the selected server. You can use the pgAdmin *Query Tool* to play back a plain-text script, and recreate the server.

[Screenshots](#)

Use the fields in the *General* tab to specify the fol owing: Enter the name of the backup file in the *Filename* field. Optional y, select the *Browser* icon (el ipsis) to the right to navigate into a directory and select a file that wil contain the archive.

Use the *Encoding* drop-down listbox to select the character encoding method that should be used for the archive. Note: This option is visible only for database server greater than or equal to 11.



Use the drop-down listbox next to *Role name* to specify a role with connection privileges on the selected server. The role will be used for authentication during the backup.

Move switches in the Type of objects field box to specify details about the type of objects that will be backed up.

Move the switch next to *Only data* to the Yes position to limit the back up to data.

Move the switch next to *Only schema* to limit the back up to schema-level database objects.

Move switches in the Do not save field box to select the objects that wil not be included in the backup.

Move the switch next to *Owner* to the Yes position to exclude commands that set object ownership.

Move the switch next to *Privilege* to the Yes position to exclude commands that create access privileges.

Move the switch next to *Tablespace* to the Yes position to exclude tablespaces.

Move the switch next to *Unlogged table data* to the Yes position to exclude the contents of unlogged tables.

Move the switch next to *Comments* to the Yes position to exclude commands that set the comments. Note: This option is visible only for database server greater than or equal to 11.



Move switches in the Queries field box to specify the type of statements that should be included in the backup.

Move the switch next to *Use Column Inserts* to the Yes position to dump the data in the form of INSERT statements and include explicit column names. Please note: this may make restoration from backup slow.

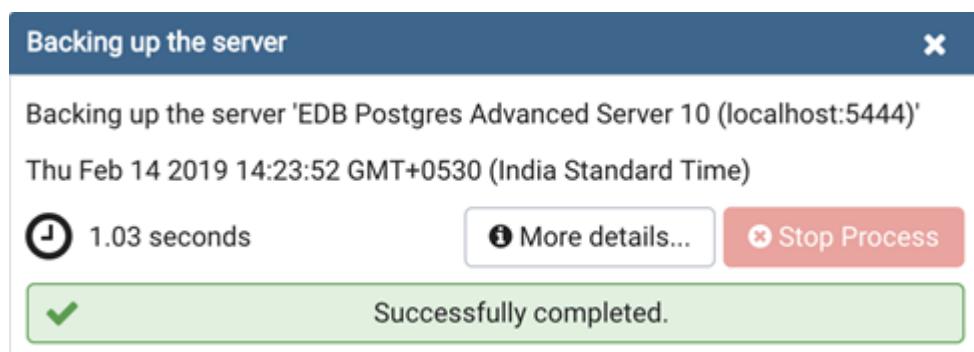
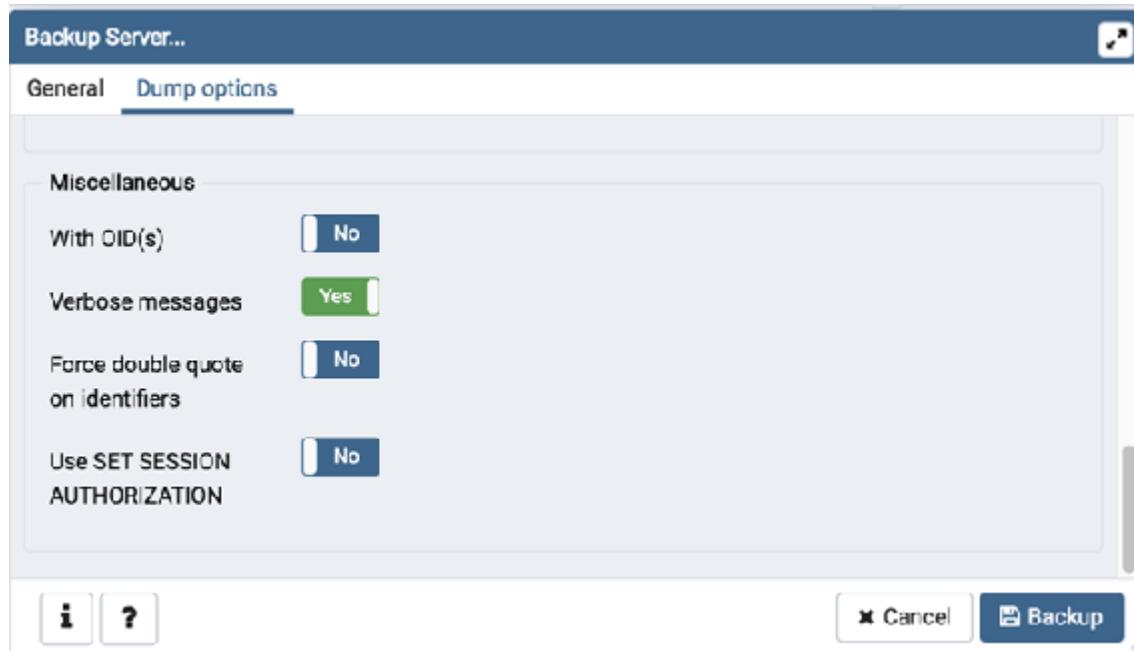
Move the switch next to *Use Insert commands* to the Yes position to dump the data in the form of INSERT statements rather than using a COPY command. Please note: this may make restoration from backup slow.

Move the switch next to *Include DROP DATABASE statement* to the Yes position to include a command in the backup that wil drop any existing database object with the same name before recreating the object during a backup.

Move switches in the Disable field box to specify the type of statements that should be excluded from the backup.

Move the switch next to *Trigger* (active when creating a data-only backup) to the Yes position to include commands that wil disable triggers on the target table while the data is being loaded.

Move the switch next to *\$ quoting* to the Yes position to enable dol ar quoting within function bodies; if disabled, the function body wil be quoted using SQL standard string syntax.



Move switches in the Miscellaneous field box to specify miscellaneous backup options.

Move the switch next to *With OIDs* to the Yes position to include object identifiers as part of the table data for each table.

Move the switch next to *Verbose messages* to the No position to instruct *pg_dump* to exclude verbose messages.

Move the switch next to *Force double quotes on identifiers* to the Yes position to force the quoting of all identifiers.

Move the switch next to *Use SET SESSION AUTHORIZATION* to the Yes position to include a statement that wil use a SET

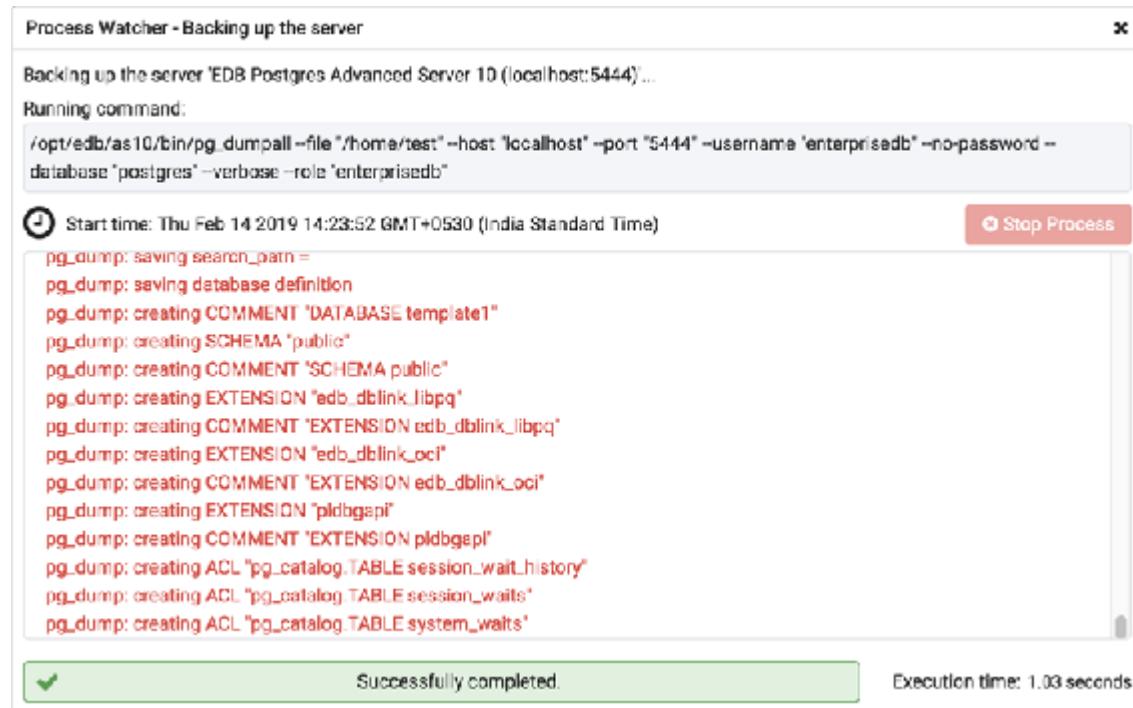
SESSION AUTHORIZATION command to determine object

ownership (instead of an ALTER OWNER command).

Click the *Backup* button to build and execute a command based on your selections; click the *Cancel* button to exit without saving work.

Use the Stop Process button to stop the Backup process.

If the backup is successful, a popup window wil confirm success. Click *Click here for details* on the popup window to launch the *Process Watcher*. The *Process Watcher* logs al the activity associated with the backup and provides additional information for troubleshooting.



The screenshot shows a window titled "Process Watcher - Backing up the server". The main area displays the command being run: "/opt/edb/as10/bin/pg_dumpall --file "/home/test" --host "localhost" --port "5444" --username 'enterprisedb' --no-password --database "postgres" --verbose --role 'enterprisedb'".

Below the command, it shows the start time: "Thu Feb 14 2019 14:23:52 GMT+0530 (India Standard Time)". On the right, there is a "Stop Process" button.

The log output shows the progress of the pg_dumpall command:

```
pg_dump: saving search_path =  
pg_dump: saving database definition  
pg_dump: creating COMMENT "DATABASE template1"  
pg_dump: creating SCHEMA "public"  
pg_dump: creating COMMENT "SCHEMA public"  
pg_dump: creating EXTENSION "edb_dblink_libpq"  
pg_dump: creating COMMENT "EXTENSION edb_dblink_libpq"  
pg_dump: creating EXTENSION "edb_dblink_ocl"  
pg_dump: creating COMMENT "EXTENSION edb_dblink_ocl"  
pg_dump: creating EXTENSION "pldbgepl"  
pg_dump: creating COMMENT "EXTENSION pldbgepl"  
pg_dump: creating ACL "pg_catalog" TABLE session_wait_history"  
pg_dump: creating ACL "pg_catalog" TABLE session_waits"  
pg_dump: creating ACL "pg_catalog" TABLE system_waits"
```

At the bottom, a green bar indicates "Successfully completed." and the execution time is listed as "Execution time: 1.03 seconds".

If the backup is unsuccessful, review the error message returned by the *Process Watcher* to resolve any issue.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) » [Management Basics](#) »

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
 - [Add Named Restore Point Dialog](#)
 - [Change Password Dialog](#)
 - [Grant Wizard](#)
 - [Import/Export Data Dialog](#)
 - [Maintenance Dialog](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Change Password Dialog

It is a good policy to routinely change your password to protect data, even in what you may consider a 'safe' environment. In the workplace, failure to apply an appropriate password policy could leave you in breach of Data Protection laws.

Please consider the following guidelines when selecting a password:

- Ensure that your password is an adequate length; 6 characters should be the absolute minimum number of characters in the password.
- Ensure that your password is not open to dictionary attacks. Use a mixture of upper and lower case letters and numerics, and avoid words or names. Consider using the first letter from each word in a phrase that you will remember easily but is an unfamiliar acronym.
- Ensure that your password is changed regularly; at minimum, change it every ninety days.

The guidelines above should be considered a starting point: They are not a comprehensive list and they will not guarantee security.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#)
[Development](#)

[pgAdmin 4 4.19 documentation](#) » [Management Basics](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[FAQ](#)

[Management Basics](#)

[Add Named Restore Point Dialog](#)



[Change Password Dialog](#)

[Grant Wizard](#)

[Latest Docs](#)

[Import/Export Data Dialog](#)

[Maintenance Dialog](#)



[Backup and Restore](#)

[Developer Tools](#)

[Ask a](#)

[pgAgent](#)

[Question](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)



[Licence](#)

[Report a Bug](#)

Change Password Dialog



It is a good policy to routinely change your password to protect data, even in what you may consider a 'safe' environment. In the

workplace, failure to

Screenshots

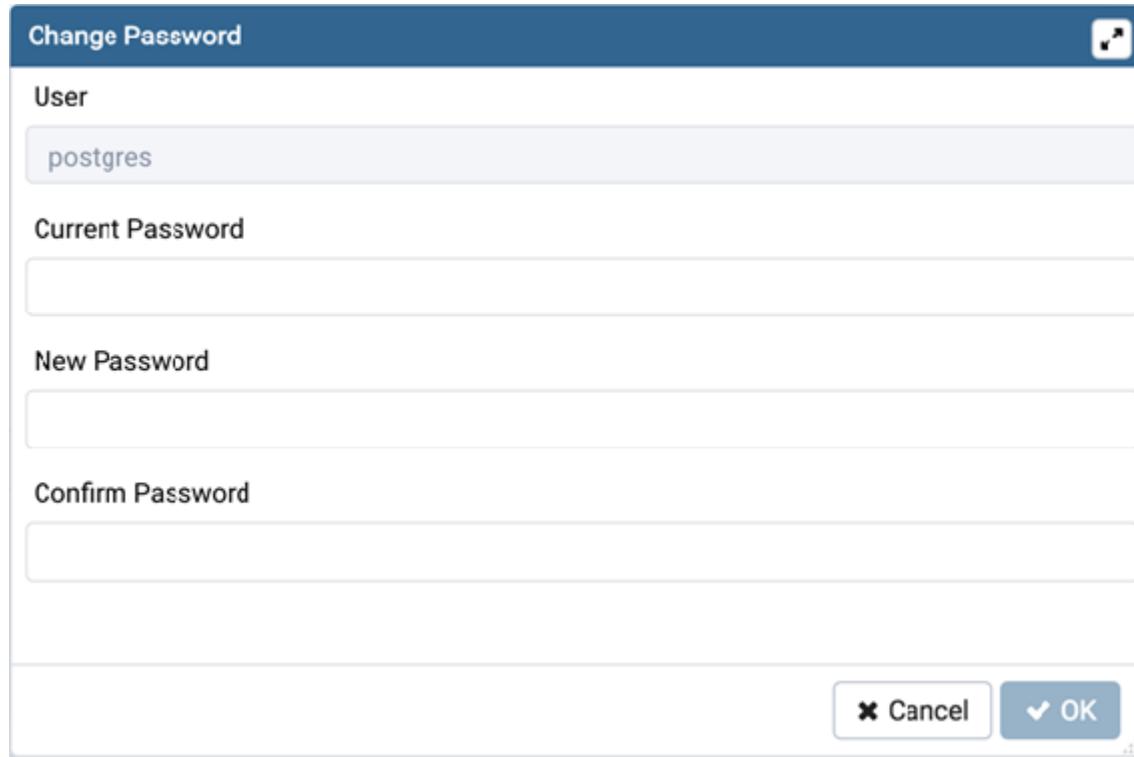
apply an appropriate password policy could leave you in breach of Data Protection laws.

Please consider the following guidelines when selecting a password:
Ensure that your password is an adequate length; 6 characters should be the absolute minimum number of characters in the password.

Ensure that your password is not open to dictionary attacks. Use a mixture of upper and lower case letters and numerics, and avoid words or names. Consider using the first letter from each word in a phrase that you will remember easily but is an unfamiliar acronym.

Ensure that your password is changed regularly; at minimum, change it every ninety days.

The guidelines above should be considered a starting point: They are not a comprehensive list and they will not guarantee security.



Use the *Change Password* dialog to change your password: The name displayed in the *User* field is the role for which you are modifying the password; it is the role that is associated with the server connection that is highlighted in the tree control.

Enter the password associated with the role in the *Current Password* field.

Enter the desired password for in the *New Password* field.

Re-enter the new password in the *Confirm Password* field.

Click the *OK* button to change your password; click *Cancel* to exit the dialog without changing your password.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | Development

[pgAdmin 4 4.19 documentation](#) > [Getting Started](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
 - [Deployment](#)
 - [Login Dialog](#)
 - [User Management Dialog](#)
 - [Change User Password Dialog](#)
 - [User Interface](#)
 - [Menu Bar](#)
 - [Toolbar](#)
 - [Tabbed Browser](#)
 - [Tree Control](#)
 - [Preferences Dialog](#)
 - [Keyboard Shortcuts](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Change User Password Dialog

It is a good policy to routinely change your password to protect data, even in what you may consider a 'safe' environment. In the workplace, failure to apply an appropriate password policy could leave you in breach of Data Protection laws.

Please consider the following guidelines when selecting a password:

- Ensure that your password is an adequate length; 6 characters should be the absolute minimum number of characters in the password.
- Ensure that your password is not open to dictionary attacks. Use a mixture of upper and lower case letters and numerics, and avoid words or names. Consider using the first letter from each word in a phrase that you will remember easily but is an unfamiliar acronym.
- Ensure that your password is changed regularly; at minimum, change it every ninety days.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Deployment](#)

[Login Dialog](#)

[User Management Dialog](#)



Change User Password Dialog

[FAQ](#)

[User Interface](#)

[Menu Bar](#)



[Toolbar](#)

[Tabbed Browser](#)

[Latest Docs](#)

[Tree Control](#)

[Preferences Dialog](#)



[Keyboard Shortcuts](#)

[Connecting To A Server](#)

[Ask a](#)

[Managing Cluster Objects](#)

[Question](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Management Basics](#)

[Backup and Restore](#)

[Report a Bug](#)

[Developer Tools](#)

[pgAgent](#)



[pgAdmin Project Contributions](#)

[Release Notes](#)

[Screenshots](#)

[Licence](#)

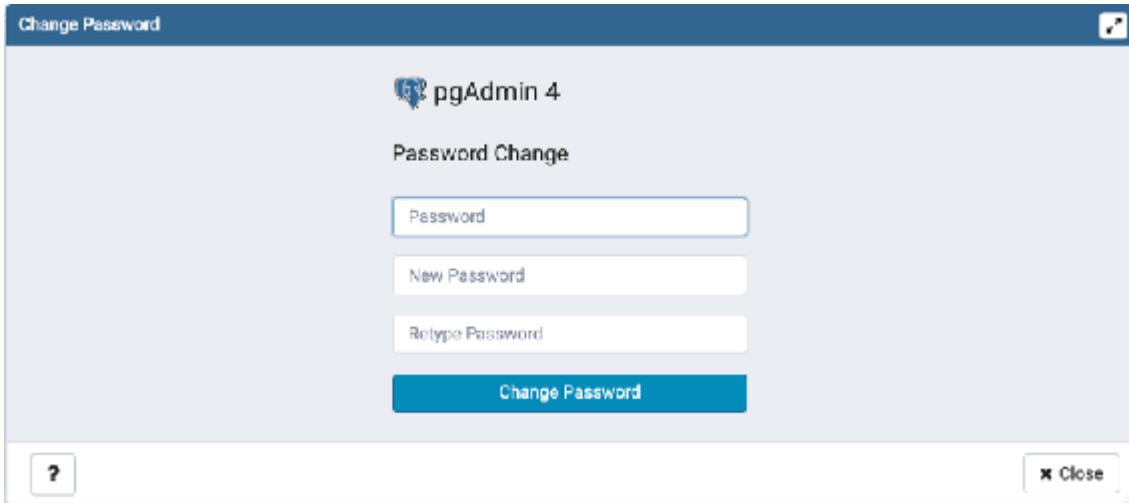
Change User Password Dialog

It is a good policy to routinely change your password to protect data, even in what you may consider a 'safe' environment. In the workplace, failure to apply an appropriate password policy could leave you in breach of Data Protection laws.

Please consider the following guidelines when selecting a password:
Ensure that your password is an adequate length; 6 characters should be the absolute minimum number of characters in the password.

Ensure that your password is not open to dictionary attacks. Use a mixture of upper and lower case letters and numerics, and avoid words or names. Consider using the first letter from each word in a phrase that you will remember easily but is an unfamiliar acronym.

Ensure that your password is changed regularly; at minimum, change it every ninety days.



The guidelines above should be considered a starting point: They are not a comprehensive list and they will not guarantee security.

Use the *Change Password* dialog to change your password: Enter your existing password in the *Current Password* field.

Enter the desired password for in the *New Password* field.

Re-enter the new password in the *Confirm Password* field.

Click the *Change Password* button to change your password; click *Close* to exit the dialog.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) » [Creation or Modifying a Table](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
 - [Check Dialog](#)
 - [Column Dialog](#)
 - [Compound Trigger Dialog](#)
 - [Exclusion Constraint Dialog](#)
 - [Foreign key Dialog](#)
 - [Index Dialog](#)
 - [Primary key Dialog](#)
 - [Rule Dialog](#)
 - [Table Dialog](#)
 - [Trigger Dialog](#)
 - [Unique Constraint Dialog](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Check Dialog

Use the *Check* dialog to define or modify a check constraint. A check constraint specifies an expression that produces a Boolean result that new or updated rows must satisfy for an insert or update operation to succeed.

The *Check* dialog organizes the development of a check constraint through the *General* and *Definition* tabs. The *SQL* tab displays the SQL code generated by dialog selections.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Creating or Modifying a Table](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[FAQ](#)

Check Dialog

[Column Dialog](#)

[□](#)

[Compound Trigger Dialog](#)

[Exclusion Constraint Dialog](#)

[Latest Docs](#)

[Foreign key Dialog](#)

[Index Dialog](#)

[□](#)

[Primary key Dialog](#)

[Rule Dialog](#)

[Ask a](#)

[Table Dialog](#)

[Question](#)

[Trigger Dialog](#)

[Unique Constraint Dialog](#)

[□](#)

[Management Basics](#)

[Backup and Restore](#)

[Report a Bug](#)

[Developer Tools](#)

[pgAgent](#)



[pgAdmin Project Contributions](#)

[Release Notes](#)

[Screenshots](#)

[Licence](#)

Check Dialog

Use the *Check* dialog to define or modify a check constraint. A check constraint specifies an expression that produces a Boolean result that new or updated rows must satisfy for an insert or update operation to succeed.

The *Check* dialog organizes the development of a check constraint through the *General* and *Definition* tabs. The *SQL* tab displays the SQL

code generated by dialog selections.

✓ Create - Check

X

General Definition SQL

Name chk_orders

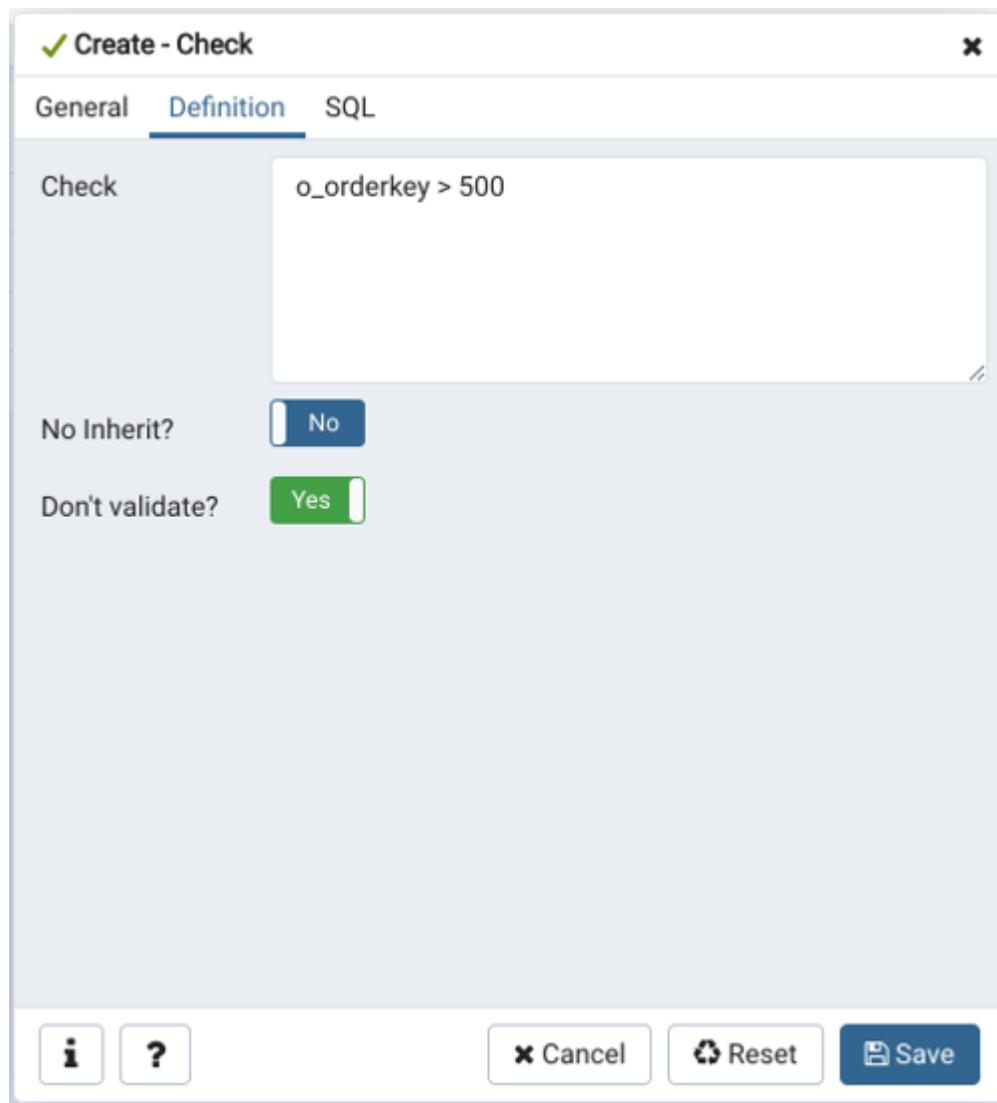
Comment



X Cancel

↻ Reset

Save



Use the fields in the *General* tab to identify the check constraint: Use the *Name* field to provide a descriptive name for the check constraint that will be displayed in the *pgAdmin* tree control. With PostgreSQL 9.5 forward, when a table has multiple check

constraints, they will be tested for each row in alphabetical order by name and after NOT NULL constraints.

Store notes about the check constraint in the *Comment* field.

Click the *Definition* tab to continue.

Use the fields in the *Definition* tab to define the check constraint:
Provide the expression that a row must satisfy in the *Check* field.

```
✓ Create - Check
General Definition SQL
1 ALTER TABLE public.orders
2   ADD CONSTRAINT chk_orders CHECK (o_orderkey > 500)
3   NOT VALID;
```

Cancel Reset Save

Move the *No Inherit?* switch to the *Yes* position to specify that this constraint is not automatically inherited by a table's children. The default is *No*, meaning that the constraint will be inherited by any children.

Move the *Don't validate?* switch to the *No* position to skip validation of existing data; the constraint may not hold for all rows in the table.

The default is *Yes*.

Click the *SQL* tab to continue.

Your entries in the *Check* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Check* dialog:

The example shown demonstrates creating a check constraint named *check_price* on the *price* column of the *products* table. The constraint confirms that any values added to the column are greater than 0.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Collation Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Collation Dialog

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Collation Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)



[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)



[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

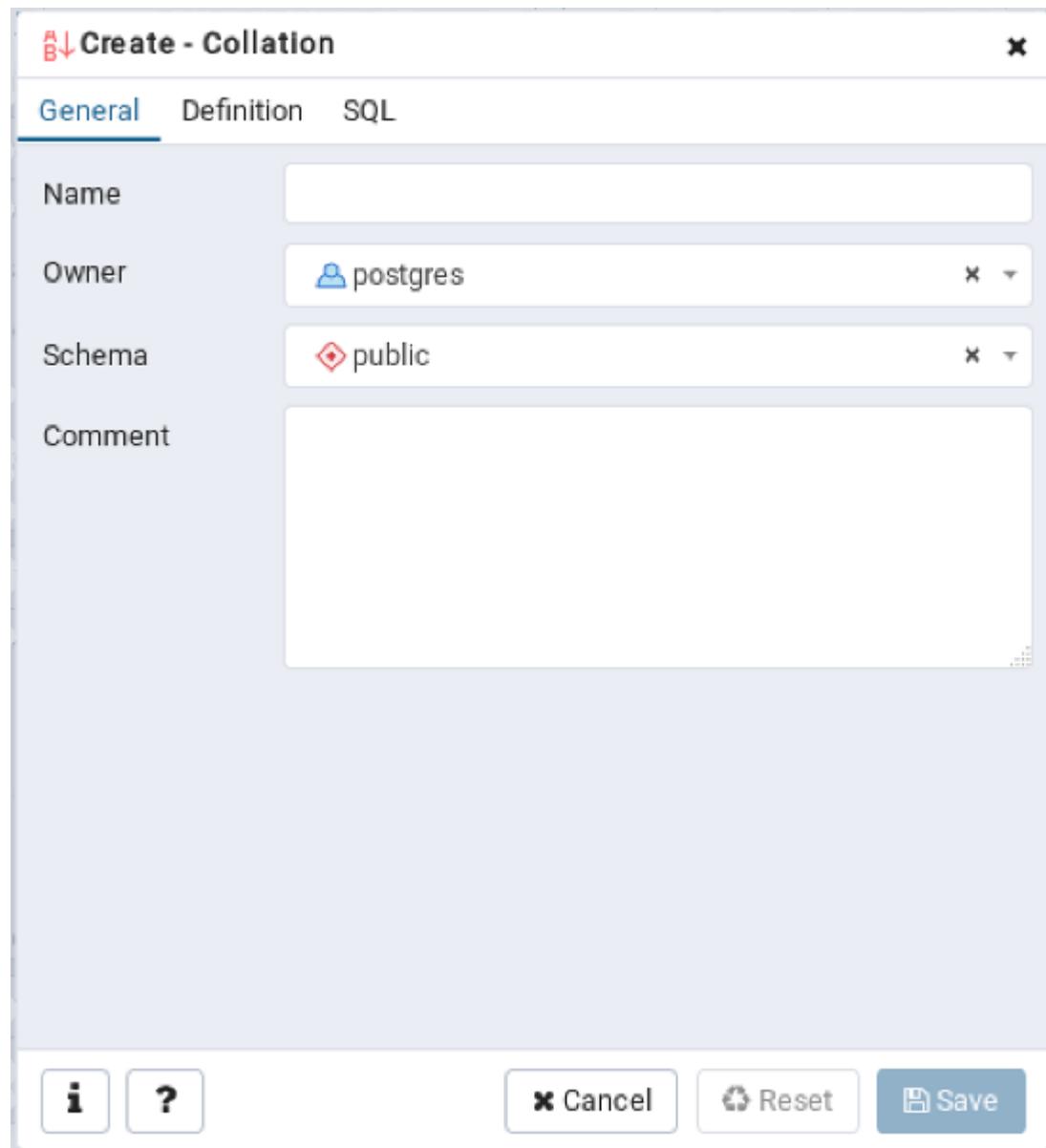
[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

Col ation Dialog



Use the *Col ation* dialog to define a col ation. A col ation is an SQL schema object that maps a SQL name to operating system locales. To create a col ation, you must have a CREATE privilege on the destination schema.

The *Col ation* dialog organizes the development of a col ation through the fol owing dialog tabs: *General* and *Definition*. The *SQL* tab displays the SQL code generated by dialog selections.

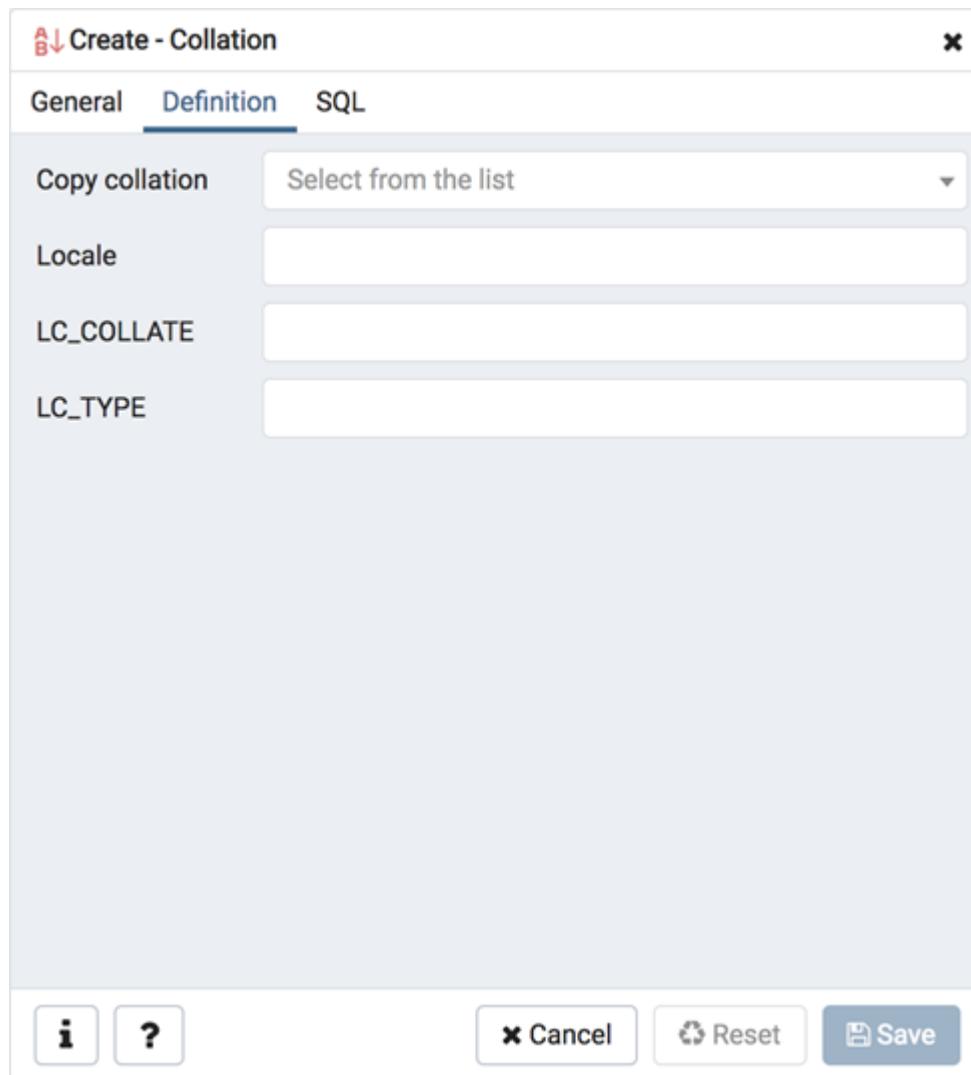
Use the fields in the *General* tab to identify the col ation: Use the *Name* field to provide a name for the col ation. The col ation name must be unique within a schema. The name wil be displayed in the *pgAdmin* tree control.

Select the name of the owner from the drop-down listbox in the *Owner* field.

Select the name of the schema in which the col ation wil reside from the drop-down listbox in the *Schema* field.

Store notes about the col ation in the *Comment* field.

Click the *Definition* tab to continue.



Use the fields in the *Definition* tab to specify the operating system locale settings:

Use the drop-down listbox next to *Copy collation* to select the name of an existing collation to copy. The new collation will have the same properties as the existing one, but will be an independent object. If you choose to copy an existing collation, you cannot modify the collation properties displayed on this tab.

Use the *Locale* field to specify a locale; a locale specifies language and language formatting characteristics. If you specify this, you cannot specify either of the following parameters. To view a list of locales supported by your Linux system use the command `locale -a`.

Use the *LC_COLLATE* field to specify a locale with specified string sort order. The locale must be applicable to the current database encoding. (See CREATE DATABASE for details.)

Use the *LC_CTYPE* field to specify a locale with specified character classification. The locale must be applicable to the current database encoding. (See CREATE DATABASE for details.)

Click the *SQL* tab to continue.

Your entries in the *Collation* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Collation* dialog:

Create - Collation

General Definition **SQL**

```
1 CREATE COLLATION public.french
2   FROM pg_catalog."fr-BI-x-icu";
3
4 ALTER COLLATION public.french
5   OWNER TO enterprisedb;
```

i **?** **Cancel** **Reset** **Save**

The example shown demonstrates creating a collation named *french* that uses the rules specified for the locale, *fr-BI-x-icu*. *The collation is owned by *postgres.*

Click the *Info* button (i) to access online help. For more information about setting a locale, see Chapter 22.1 Locale Support of the

PostgreSQL core documentation:

<https://www.postgresql.org/docs/current/locale.html>

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6) Development

[pgAdmin 4 4.19 documentation](#) > [Creation or Modifying a Table](#) > [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Column Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
 - [Check Dialog](#)
 - [Column Dialog](#)
 - [Compound Trigger Dialog](#)
 - [Exclusion Constraint Dialog](#)
 - [Foreign key Dialog](#)
 - [Index Dialog](#)
 - [Primary key Dialog](#)
 - [Rule Dialog](#)
 - [Table Dialog](#)
 - [Trigger Dialog](#)
 - [Unique Constraint Dialog](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Column Dialog

Use the Column dialog to add a column to an existing table or modify a column definition.

The Column dialog organizes the development of a column through the following dialog tabs: General, Definition, and Security. The SQL tab displays the SQL code generated by dialog selections.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Creating or Modifying a Table](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Column Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[□](#)

[Check Dialog](#)

[Column Dialog](#)

[Latest Docs](#)

[Compound Trigger Dialog](#)

[Exclusion Constraint Dialog](#)

[□](#)

[Foreign key Dialog](#)

[Index Dialog](#)

[Ask a](#)

[Primary key Dialog](#)

[Question](#)

[Rule Dialog](#)

[Table Dialog](#)

[□](#)

[Trigger Dialog](#)

[Report a Bug](#)

[Unique Constraint Dialog](#)

[Management Basics](#)

[Backup and Restore](#)



[Developer Tools](#)

[Screenshots](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Column Dialog

Use the *Column* dialog to add a column to an existing table or modify a column definition.

The *Column* dialog organizes the development of a column through the following dialog tabs: *General*, *Definition*, and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

Create - Column

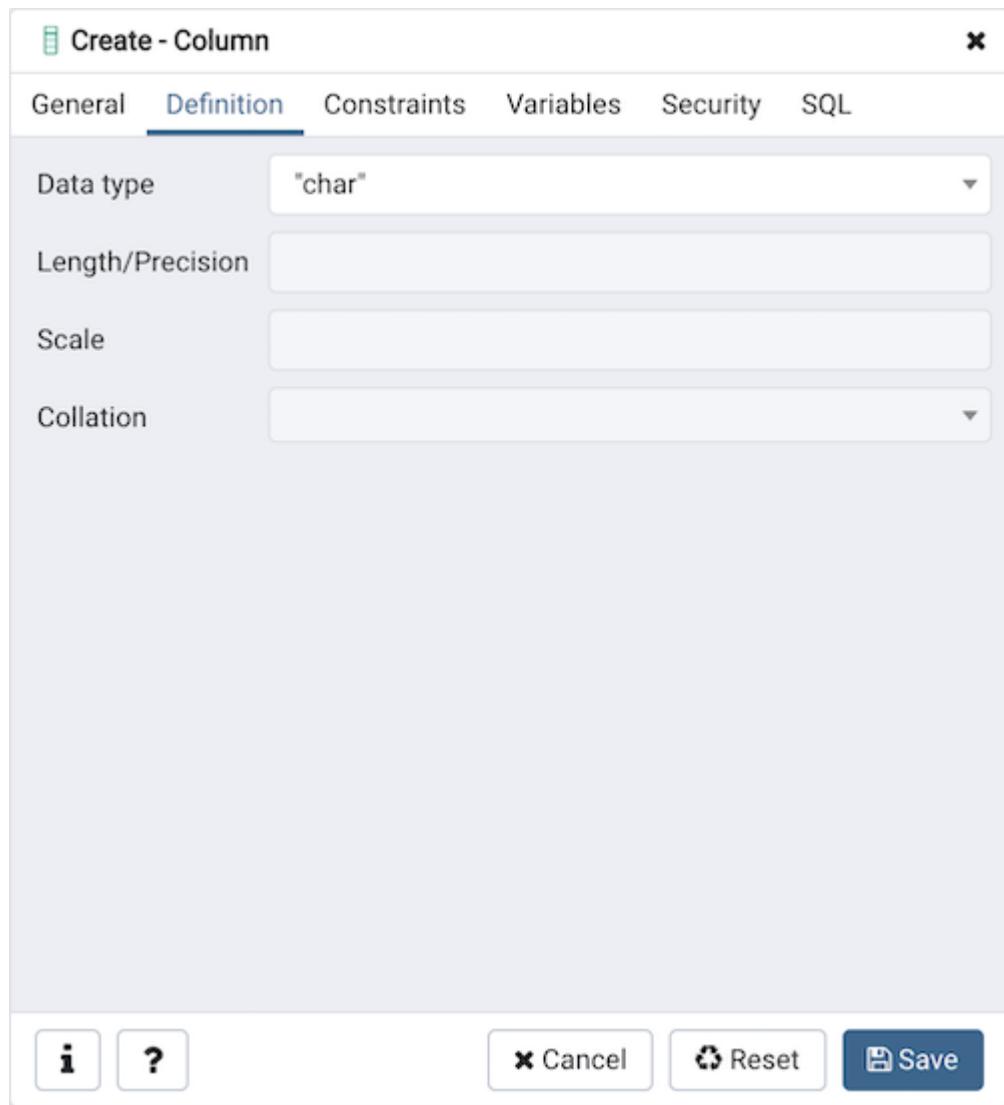
X

General Definition Constraints Variables Security SQL

Name: distributors

Comment:

i **?** **X** Cancel **↻** Reset **Save**



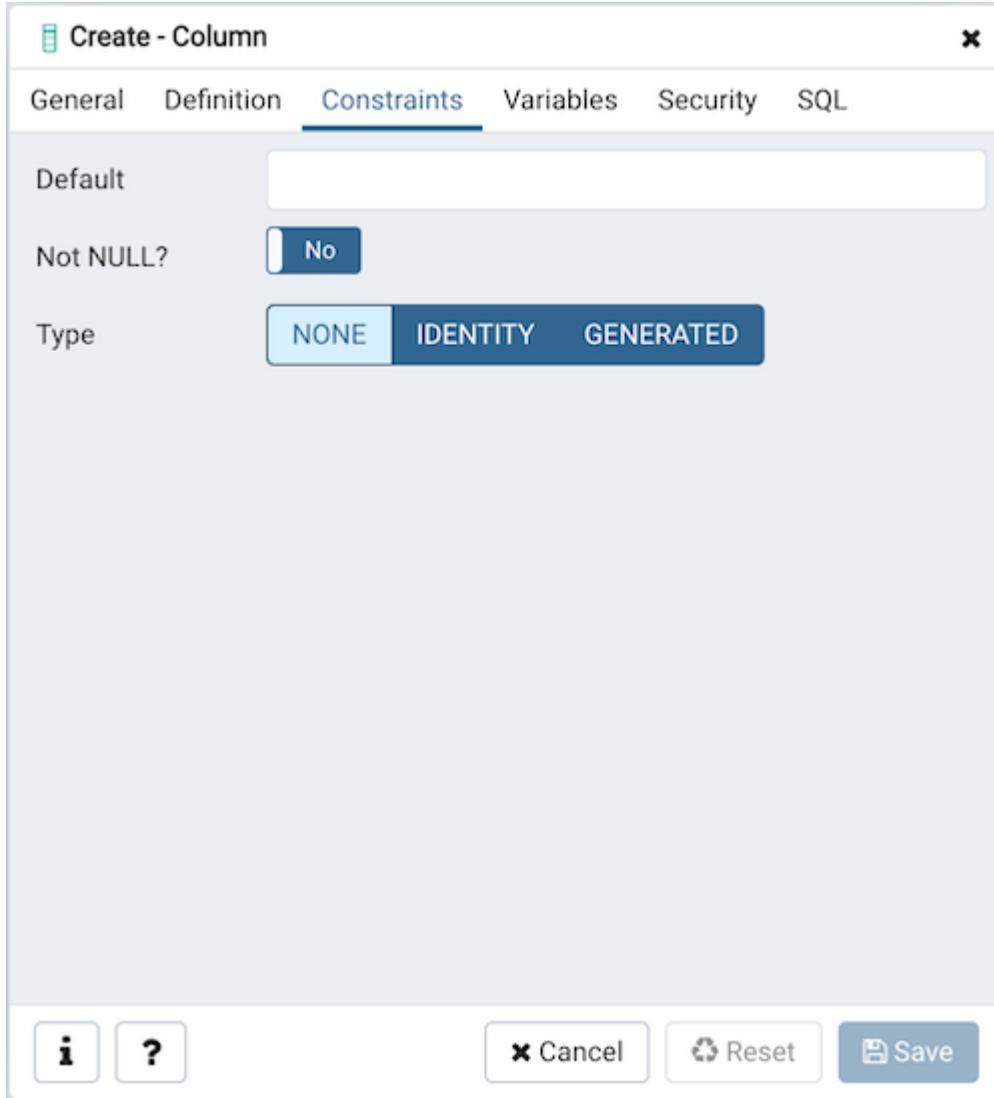
Use the fields in the *General* tab to identify the column: Use the *Name* field to add a descriptive name for the column. The name will be displayed in the *pgAdmin* tree control. This field is required.

Store notes about the column in the *Comment* field.

Click the *Definition* tab to continue.

Use the fields in the *Definition* tab to add parameters for the column.
(Fields are disabled if inapplicable.)

Use the drop-down listbox next to *Data Type* to select a data type for the column. For more information on the data types that are



supported by PostgreSQL, refer to Chapter 8 of the Postgres core documentation. This field is required.

Use the *Length/Precision* and *Scale* fields to specify the maximum number of significant digits in a numeric value, or the maximum number of characters in a text value.

Use the drop-down listbox next to *Collation* to apply a collation setting to the column.

Click the *Constraints* tab to continue.

Use the fields in the *Constraints* tab to specify constraints for the column.

(Fields are disabled if inapplicable.)

Use the *Default Value* field to specify a default data value.

Move the *Not Null* switch to the *Yes* position to specify the column may not contain null values. The default is *No*.

Use the *Type* field to specify the column type

(NONE/IDENTITY/GENERATED). The default is *NONE*.

Click the *IDENTITY* type to create Identity column.

Create - Column

General Definition **Constraints** Variables Security SQL

Default

Not NULL? Yes

Type NONE IDENTITY GENERATED

Identity ALWAYS

Increment 1

Start 1

Minimum 1

Maximum 1000

Cache 1

Cycled Yes

? **i** **Cancel** **Reset** **Save**

The screenshot shows the 'Create - Column' dialog box with the 'Constraints' tab selected. The 'Type' section is set to 'IDENTITY'. Other fields include 'Identity' (set to 'ALWAYS'), 'Increment' (1), 'Start' (1), 'Minimum' (1), 'Maximum' (1000), 'Cache' (1), and 'Cycled' (Yes). Buttons at the bottom include 'Cancel', 'Reset', and 'Save'.

Use the following fields to create *IDENTITY* column. Identity columns are applicable for PG/EPAS version 10 and above.

Use the *Identity* field to specify ALWAYS or BY DEFAULT. This clause is used to determine how the sequence value is given precedence over a user-specified value in an INSERT statement.

Use the *Increment* field to specify which value is added to the current sequence value to create a new value.

Provide a value in the *Start* field to specify the beginning value of the sequence. The default starting value is MINVALUE for ascending sequences and MAXVALUE for descending ones.

Provide a value in the *Minimum* field to specify the minimum value a sequence can generate. If this clause is not supplied or NO

MINVALUE is specified, then defaults wil be used. The defaults are 1 and -263-1 for ascending and descending sequences, respectively.

Provide a value in the *Maximum* field to specify the maximum value for the sequence. If this clause is not supplied or NO MAXVALUE is specified, then default values wil be used. The defaults are 263-1

and -1 for ascending and descending sequences, respectively.

Provide a value in the *Cache* field to specify how many sequence numbers are to be preal ocated and stored in memory for faster access. The minimum value is 1 (only one value can be generated at a time, i.e., no cache), and this is also the default.

Move the *Cycled* switch to the *Yes* position to al ow the sequence to wrap around when the MAXVALUE or the MINVALUE has been

reached by an ascending or descending sequence respectively. If the limit is reached, the next number generated wil be the MINVALUE or MAXVALUE, respectively. The default is *No*.

Click the *GENERATED* type to create Generated column.

Create - Column

x

General Definition **Constraints** Variables Security SQL

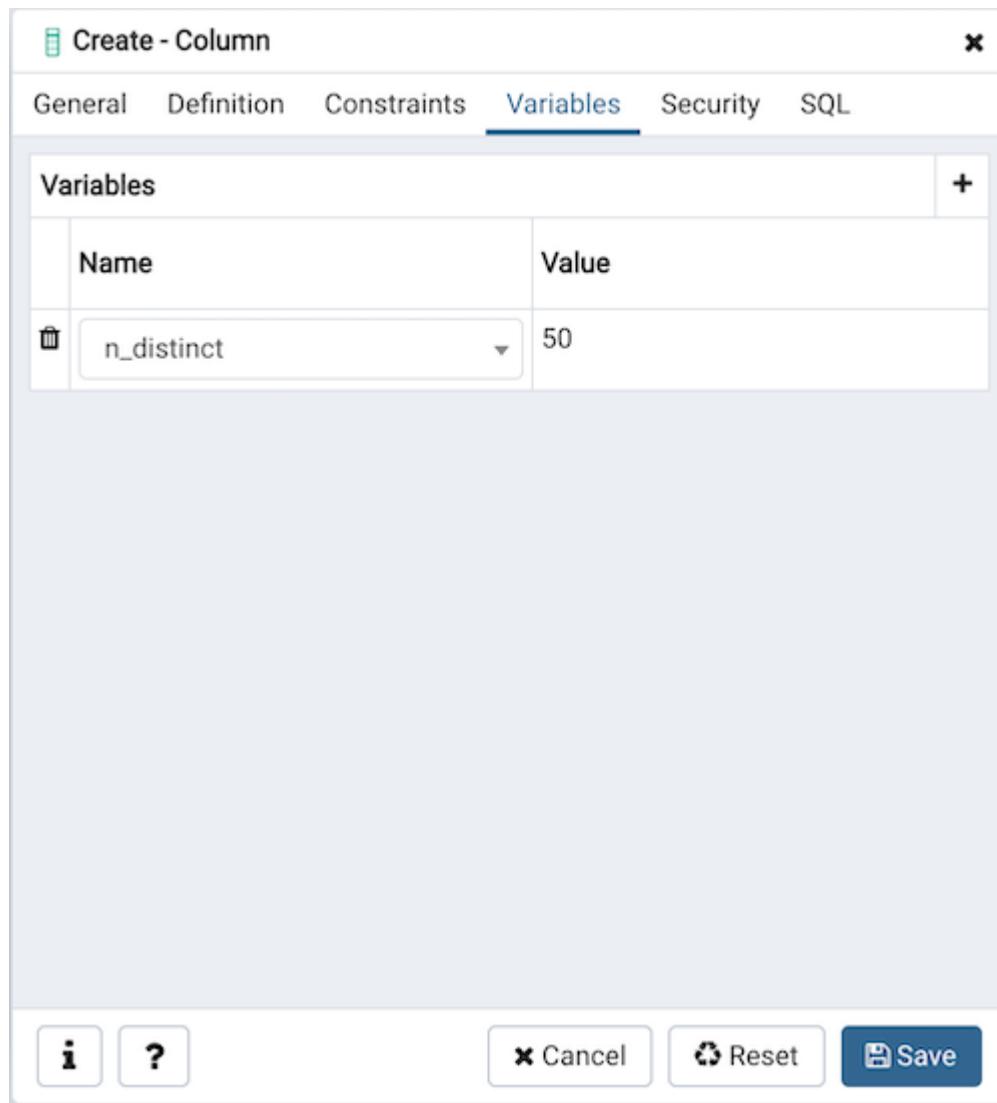
Default

Not NULL? Yes

Type NONE IDENTITY GENERATED

Expression

i ? Cancel Reset Save



Use the following fields to create *GENERATED* column. Generated columns are applicable for PG/EPAS version 12 and above.

Use the *Expression* field to specify the generation expression. It can refer to other columns in the table, but not other generated columns.

Any functions and operators used must be immutable. References to other tables are not allowed.

Click the *Variables* tab to continue.

Use the *Variables* tab to specify the number of distinct values that may be present in the column; this value overrides estimates made

by the ANALYZE command. Click the *Add* icon (+) to add a *Name/Value* pair:

Security labels		+
	Provider	Security label
!	provider1	label_group_1

i **?** ✖ Cancel ♻ Reset 💾 Save

Select the name of the variable from the drop-down listbox in the *Name* field.

Select *n_distinct* to specify the number of distinct values for the column.

Select *n_distinct_inherited* to specify the number of distinct values for the table and its children.

Specify the number of distinct values in the *Value* field. For more information, see the documentation for [ALTER TABLE](#).

Click the *Add* icon (+) to specify each additional *Name/ Value* pair; to discard a variable, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *Security* tab to continue.

Use the *Security* tab to assign attributes and define security labels. Click the *Add* icon (+) to add each security label selection: Specify a security label provider in the *Provider* field. The named provider must be loaded and must consent to the proposed labeling operation.

Specify a a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL

places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the *Add* icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.



Your entries in the *Column* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Column* dialog:

The example shown demonstrates creating a column named *territory* in the table named *distributors*.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | Development

[pgAdmin 4 documentation](#) > [Creation or Modifying a Table](#) > [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Compound Trigger Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
 - [Check Dialog](#)
 - [Column Dialog](#)
 - [Compound Trigger Dialog](#)
 - [Exclusion Constraint Dialog](#)
 - [Foreign key Dialog](#)
 - [Index Dialog](#)
 - [Primary key Dialog](#)
 - [Rule Dialog](#)
 - [Table Dialog](#)
 - [Trigger Dialog](#)
 - [Unique Constraint Dialog](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Compound Trigger Dialog

Use the *Compound Trigger* dialog to create a compound trigger or modify an existing compound trigger. *Compound Trigger* is supported only for EPAS server 12 and above. A compound trigger executes a specified code when certain events occur.

The *Compound Trigger* dialog organizes the development of a compound trigger through the following dialog tabs: General, Events, and Code. The SQL tab displays the SQL code generated by dialog selections.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Creating or Modifying a Table](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



Compound Trigger Dialog

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Check Dialog](#)

[Column Dialog](#)

[Latest Docs](#)

Compound Trigger Dialog

[Exclusion Constraint Dialog](#)



[Foreign key Dialog](#)

[Index Dialog](#)

[Ask a](#)

[Primary key Dialog](#)

[Question](#)

[Rule Dialog](#)

[Table Dialog](#)



[Trigger Dialog](#)

[Report a Bug](#)

[Unique Constraint Dialog](#)

[Management Basics](#)

[Backup and Restore](#)



[Developer Tools](#)

[Screenshots](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Compound Trigger Dialog 

Use the *Compound Trigger* dialog to create a compound trigger or modify an existing compound trigger. *Compound Trigger* is supported only for EPAS server 12 and above. A compound trigger executes a specified code when certain events occur.

The *Compound Trigger* dialog organizes the development of a compound trigger through the following dialog tabs: *General*, *Events*, and *Code*. The *SQL* tab displays the SQL code generated by dialog selections.

_CREATE - Compound Trigger

X

General Events Code SQL

Name

test_ct

Comment

This is a compound trigger

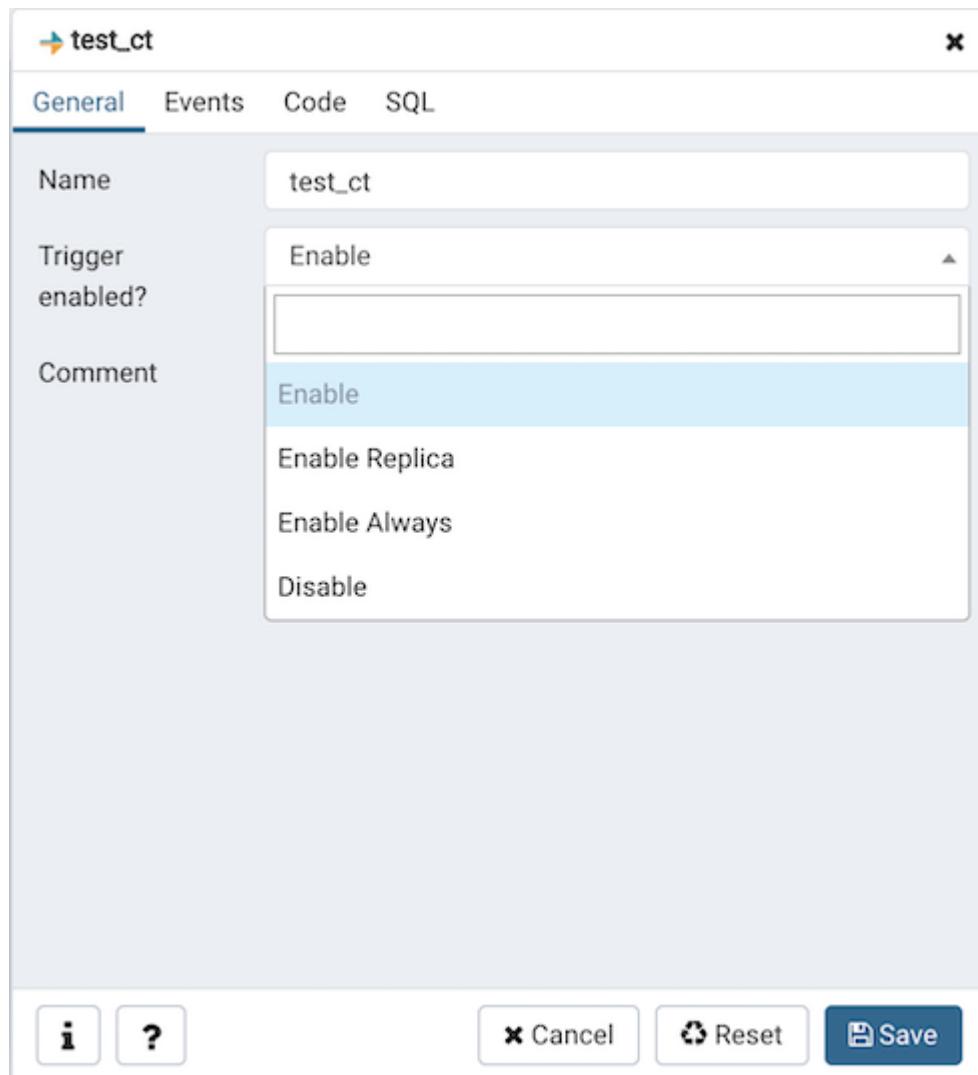
i

?

Cancel

Reset

Save

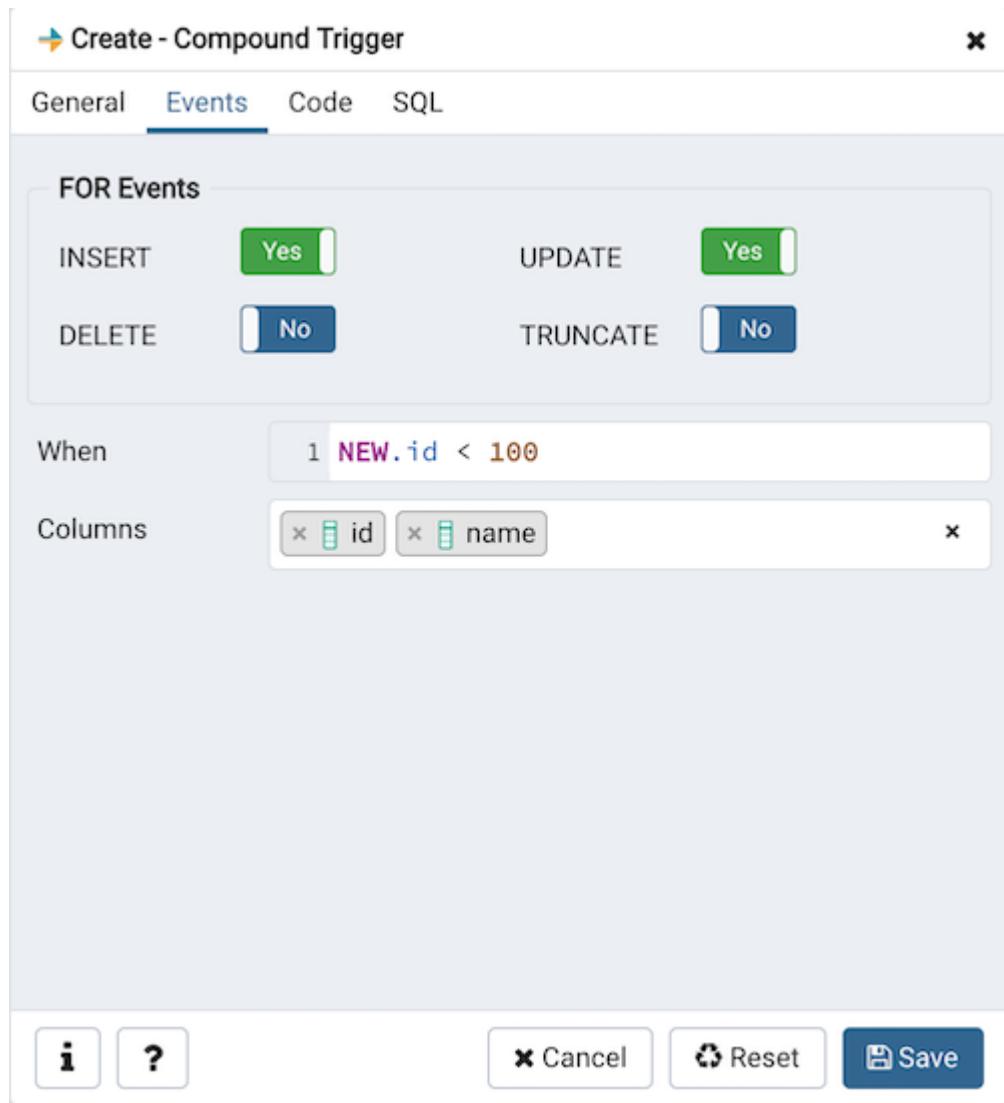


Use the fields in the *General* tab to identify the compound trigger:
Use the *Name* field to add a descriptive name for the compound trigger. This must be distinct from the name of any other compound trigger for the same table. The name will be displayed in the *pgAdmin* tree control.

Store notes about the compound trigger in the *Comment* field.

Trigger enabled field is available in compound trigger dialog once the trigger is created. You can select one of the four options available.

Click the *Events* tab to continue.



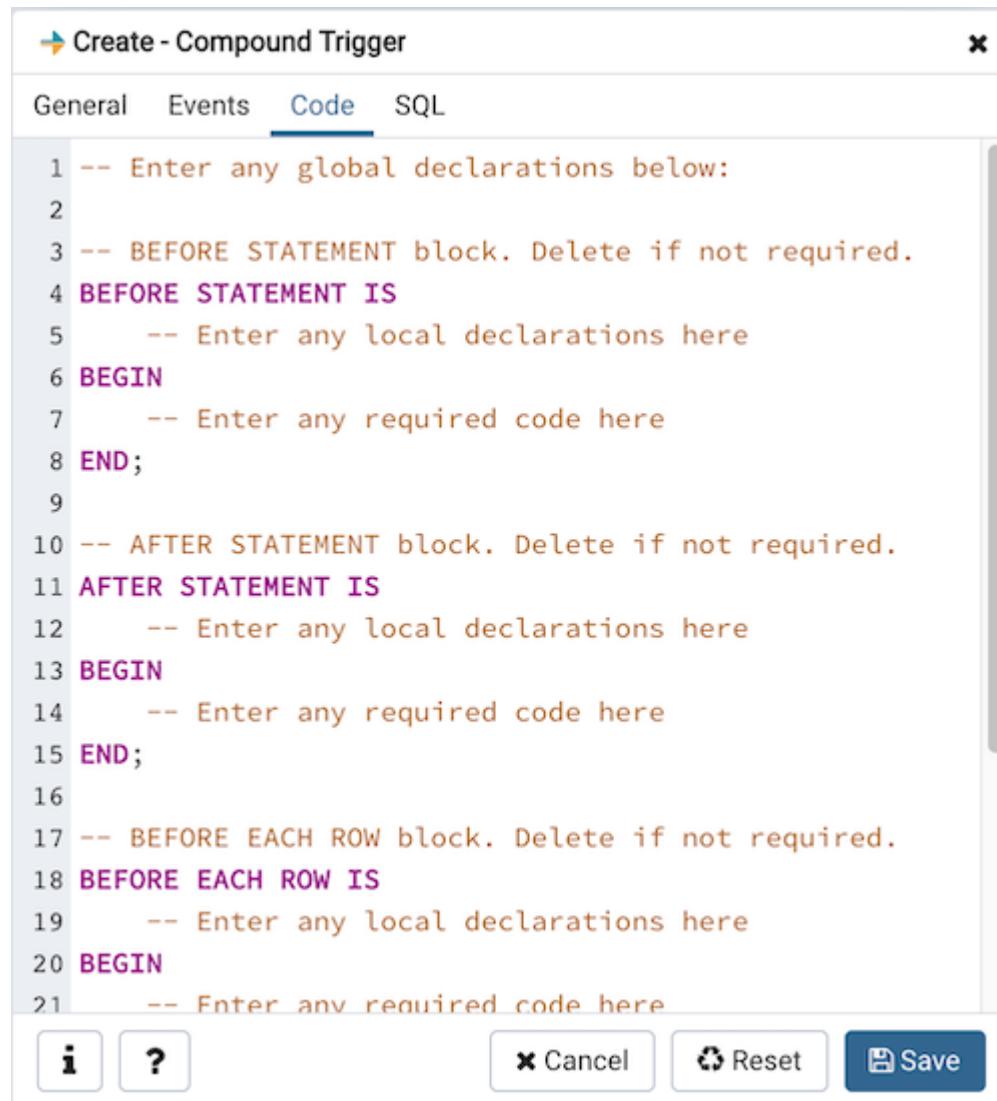
Use the fields in the *Events* tab to specify how and when the compound trigger fires:

Select the type of event(s) that will invoke the compound trigger; to select an event type, move the switch next to the event to the **YES** position. The supported event types are *INSERT*, *UPDATE*, *DELETE* and *TRUNCATE*. Views cannot have TRUNCATE triggers.

Use the *When* field to provide a boolean condition that will invoke the compound trigger.

If defining a column-specific compound trigger, use the *Columns* field to specify the columns or columns that are the target of the compound trigger.

Click the *Code* tab to continue.



The screenshot shows a dialog box titled "Create - Compound Trigger". The "Code" tab is selected, indicated by a blue underline. The code area contains template SQL for a compound trigger:

```
1 -- Enter any global declarations below:  
2  
3 -- BEFORE STATEMENT block. Delete if not required.  
4 BEFORE STATEMENT IS  
5     -- Enter any local declarations here  
6 BEGIN  
7     -- Enter any required code here  
8 END;  
9  
10 -- AFTER STATEMENT block. Delete if not required.  
11 AFTER STATEMENT IS  
12     -- Enter any local declarations here  
13 BEGIN  
14     -- Enter any required code here  
15 END;  
16  
17 -- BEFORE EACH ROW block. Delete if not required.  
18 BEFORE EACH ROW IS  
19     -- Enter any local declarations here  
20 BEGIN  
21     -- Enter any required code here
```

At the bottom of the dialog are three buttons: "i" (Information), "?" (Help), "Cancel" (with a cancel icon), "Reset" (with a recycle icon), and "Save" (a blue button).

Use the *Code* field to specify the code for the five timing events *BEFORE*

STATEMENT, AFTER STATEMENT, BEFORE EACH ROW, AFTER EACH

ROW, INSTEAD OF EACH ROW that will be invoked when the compound trigger fires. Basic template is provided with place holders.

Click the **SQL** tab to continue.

Your entries in the *Compound Trigger* dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Compound Trigger* dialog:

>Create - Compound Trigger

General Events Code SQL

```
1 CREATE OR REPLACE TRIGGER test_ct
2   FOR INSERT
3   ON enterprisedb.test
4   WHEN (NEW.id < 100)
5   COMPOUND TRIGGER
6   -- Global declaration.
7   var varchar2(20) := 'Global_var';
8
9   BEFORE STATEMENT IS
10  BEGIN
11    DBMS_OUTPUT.PUT_LINE('Before Statement: ' || var);
12    var := 'BEFORE STATEMENT';
13  END;
14 END;
```

i ? Cancel Reset Save

The example demonstrates creating a compound trigger named `test_ct`.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Connecting To A Server](#) > [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
 - [Server Group Dialog](#)
 - [Server Dialog](#)
 - [Master Password](#)
 - [Connect to Server](#)
 - [Connection Error](#)
 - [Import/Export Servers](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

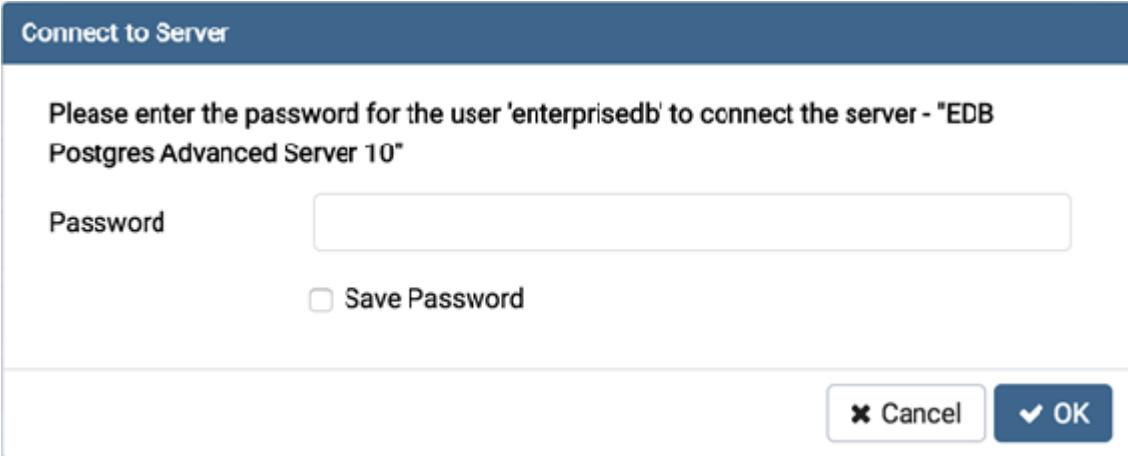
Connect to Server

Use the Connect to Server dialog to authenticate with a defined server and access the objects stored on the server through the pgAdmin tree control. To access the dialog, right click on the server name in the pgAdmin tree control, and select Connect Server... from the context menu.



Provide authentication information for the selected server:

- Use the Password field to provide the password of the user that is associated with the defined server.
- Check the box next to Save Password to instruct the server to save the password for future connections; if you save the password, you will not be prompted when reconnecting to the database server with this server definition.



Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Connecting To A Server](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Server Group Dialog](#)

[Server Dialog](#)



[Master Password](#)

[FAQ](#)

Connect to Server

[Connection Error](#)



[Import/Export Servers](#)

[Managing Cluster Objects](#)

[Latest Docs](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Management Basics](#)

[Backup and Restore](#)

[Ask a](#)

[Developer Tools](#)

[Question](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)



[Release Notes](#)

[Licence](#)

[Report a Bug](#)

Connect to Server



Use the *Connect to Server* dialog to authenticate with a defined server

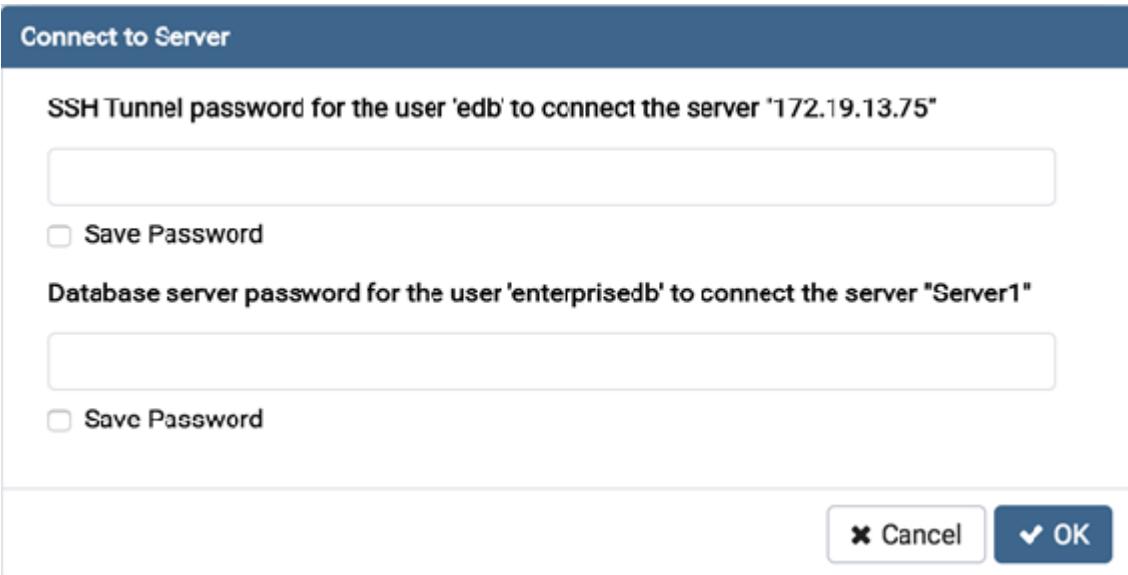
[Screenshots](#)

and access the objects stored on the server through the pgAdmin tree control. To access the dialog, right click on the server name in the *pgAdmin* tree control, and select *Connect Server...* from the context menu.

Provide authentication information for the selected server:

Use the *Password* field to provide the password of the user that is associated with the defined server.

Check the box next to *Save Password* to instruct the server to save the password for future connections; if you save the password, you will not be prompted when reconnecting to the database server with this server definition.



When using SSH Tunneling, the *Connect to Server* dialog will prompt for the SSH Tunnel and Database server passwords if not already saved.

Provide authentication information for the selected server:

Use the *Password* field to provide the password of the user that is associated with the defined server.

Check the box next to respective *Save Password* to instruct the server to save the password for future connections; if you save the password, you will not be prompted when reconnecting to the database server with this server definition.

The pgAdmin client displays a message in a green status bar in the lower right corner when the server connects successfully.

If you receive an error message while attempting a connection, verify that your network is allowing the pgAdmin host and the host of the database server to communicate. For detailed information about a specific error message, please see the [Connection Error](#) help page.

To review or modify connection details, right-click on the name of the server, and select *Properties...* from the context menu.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 4.19 documentation](#) »

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
 - [Server Group Dialog](#)
 - [Server Dialog](#)
 - [Master Password](#)
 - [Connect to Server](#)
 - [Connection Error](#)
 - [Import/Export Servers](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Connecting To A Server

Before you can use the pgAdmin client to manage the objects that reside on your Postgres server, you must define a connection to the server. You can (optionally) use the Server Group dialog to create server groups to organize the server connections within the tree control for easier management. To open the Server Group dialog, right-click on the Servers node of the tree control, and select Server Group from the Create menu.

- [Server Group Dialog](#)

Use the fields on the Server dialog to define the connection properties for each new server that you wish to manage with pgAdmin. To open the Server dialog, right-click on the Servers node of the tree control, and select Server from the Create menu.

- [Server Dialog](#)
 - [Clear Saved Passwords](#)

A master password is required to secure and later unlock saved server passwords. It is set by the user and can be disabled using config.

- [Master Password](#)

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#)
[Development](#)

[pgAdmin 4 4.19 documentation](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

Connecting To A Server

[Server Group Dialog](#)

[Server Dialog](#)



[Master Password](#)

[FAQ](#)

[Connect to Server](#)

[Connection Error](#)



[Import/Export Servers](#)

[Managing Cluster Objects](#)

[Latest Docs](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Management Basics](#)

[Backup and Restore](#)

[Ask a](#)

[Developer Tools](#)

[Question](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)



[Release Notes](#)

[Licence](#)

[Report a Bug](#)

Connecting To A Server



Before you can use the pgAdmin client to manage the objects that reside

[Screenshots](#)

on your Postgres server, you must define a connection to the server. You can (optional y) use the *Server Group* dialog to create server groups to organize the server connections within the tree control for easier

management. To open the *Server Group* dialog, right-click on the *Servers* node of the tree control, and select *Server Group* from the *Create* menu.

[Server Group Dialog](#)

Use the fields on the *Server* dialog to define the connection properties for each new server that you wish to manage with pgAdmin. To open the

Server dialog, right-click on the *Servers* node of the tree control, and select *Server* from the *Create* menu.

[Server Dialog](#)

[Clear Saved Passwords](#)

A master password is required to secure and later unlock saved server

passwords. It is set by the user and can be disabled using config.

[Master Password](#)

After defining a server connection, right-click on the server name, and select *Connect to server* to authenticate with the server, and start using pgAdmin to manage objects that reside on the server.

[Connect to Server](#)

[Connection Error](#)

Server definitions (and their groups) can be exported to a JSON file and re-imported to the same or a different system to enable easy pre-configuration of pgAdmin.

[Import/Export Servers](#)

[Exporting Servers](#)

[Importing Servers](#)

[JSON format](#)

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | Development

[pgAdmin 4 documentation](#) » [Getting Started](#) » [Deployment](#) » previous | next | index

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Container Deployment](#)
 - [PostgreSQL Utilities](#)
 - [Environment Variables](#)
 - [Mapped Files and Directories](#)
 - [Examples](#)
 - [Reverse Proxying](#)
 - [pgAdmin X-Forwarded-* Configuration](#)
 - [HTTP via Nginx](#)
 - [HTTPS via Nginx](#)
 - [Traefik](#)
- [Getting Started](#)
 - [Deployment](#)
 - [Login Dialog](#)
 - [User Management Dialog](#)
 - [Change User Password Dialog](#)
 - [User Interface](#)
 - [Menu Bar](#)
 - [Toolbar](#)
 - [Tabbed Browser](#)
 - [Tree Control](#)
 - [Preferences Dialog](#)
 - [Keyboard Shortcuts](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [onAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Container Deployment

pgAdmin can be deployed in a container using the image at:

<https://hub.docker.com/r/dpage/pgadmin4/>

[PostgreSQL Utilities](#)

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) » [Deployment](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



Container Deployment

[Download](#)

[PostgreSQL Utilities](#)

[Environment Variables](#)

[Mapped Files and Directories](#)



[Examples](#)

[FAQ](#)

[Reverse Proxying](#)

[pgAdmin X-Forwarded-* Configuration](#)



[HTTP via Nginx](#)

[HTTPS via Nginx](#)

[Latest Docs](#)

[Traefik](#)

[Getting Started](#)



[Deployment](#)

[Ask a](#)

[Login Dialog](#)

[Question](#)

[User Management Dialog](#)

[Change User Password Dialog](#)

[User Interface](#)



[Menu Bar](#)

[Report a Bug](#)

[Toolbar](#)

[Tabbed Browser](#)

[Tree Control](#)



[Preferences Dialog](#)

[Screenshots](#)

[Keyboard Shortcuts](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Container Deployment 

pgAdmin can be deployed in a container using the image at:

<https://hub.docker.com/r/dpage/pgadmin4/>

PostgreSQL Utilities 

The PostgreSQL utilities *pg_dump*, *pg_dumpal* , *pg_restore* and *psql* are included in the container to allow backups to be created and restored and other maintenance functions to be executed. Multiple

versions are included in the following directories to allow use with different versions of the database server:

PostgreSQL 9.4: `/usr/local/pgsql-9.4`

PostgreSQL 9.5: `/usr/local/pgsql-9.5`

PostgreSQL 9.6: `/usr/local/pgsql-9.6`

PostgreSQL 10: `/usr/local/pgsql-10`

PostgreSQL 11: `/usr/local/pgsql-11`

PostgreSQL 12: `/usr/local/pgsql-12`

The most recent version of the utilities is used by default; this may be changed in the [Preferences Dialog](#).

Environment Variables

The container will accept the following variables at startup:

PGADMIN_DEFAULT_EMAIL

This is the email address used when setting up the initial administrator account to login to pgAdmin. This variable is required and must be set at launch time.

PGADMIN_DEFAULT_PASSWORD

This is the password used when setting up the initial administrator account to login to pgAdmin. This variable is required and must be set at launch time.

PGADMIN_ENABLE_TLS

Default: <nul>

If left un-set, the container will listen on port 80 for connections in plain text. If set to any value, the container will listen on port 443 for

TLS

connections.

When TLS is enabled, a certificate and key must be provided. Typically these should be stored on the host file system and mounted from the container. The expected paths are /certs/server.crt and /certs/server.key

PGADMIN_LISTEN_ADDRESS

Default: [::]

Specify the local address that the servers listens on. The default should work for most users - in IPv4-only environments, this may need to be set to 127.0.0.1.

PGADMIN_LISTEN_PORT

Default: 80 or 443 (if TLS is enabled)

Allows the port that the server listens on to be set to a specific value rather than using the default.

PGADMIN_SERVER_JSON_FILE

Default: /pgadmin4/servers.json

Override the default file path for the server definition list. See the /pgadmin4/servers.json mapped file below for more information.

GUNICORN_ACCESS_LOGFILE

Default: - (stdout)

Specify an output file in which to store the Gunicorn access logs, instead of sending them to stdout.

GUNICORN_THREADS

Default: 25

Adjust the number of threads the Gunicorn server uses to handle incoming requests. This should typically be left as-is, except in highly loaded systems where it may be increased.

PGADMIN_CONFIG_

This is a variable prefix that can be used to override any of the configuration options in pgAdmin's *config.py* file. Add the *PGADMIN_CONFIG_* prefix to any variable name from *config.py* and give the value in the format 'string value' for strings, True/False for booleans or 123 for numbers. See below for an example.

Settings are written to */pgadmin4/config_distro.py* within the container, which is read after */pgadmin4/config.py* and before

/pgadmin4/config_local.py. Any settings given will therefore override anything in *config.py*, but can be overridden by settings in *config_local.py*.

See [The config.py File](#) for more information on the available configuration settings.

Mapped Files and Directories

The following files or directories can be mapped from the container onto the host machine to allow configuration to be customised and shared between instances.

Warning

Warning: pgAdmin runs as the *pgadmin* user (UID: 5050) in the *pgadmin* group (GID: 5050) in the container. You must ensure that all files are readable, and where necessary (e.g. the working/session directory) writeable for this user on the host machine. For example:
`sudo chown -R 5050:5050 <host_directory>`

On some filesystems that do not support extended attributes, it may not be possible to run pgAdmin without specifying a value for *PGADMIN_LISTEN_PORT* that is greater than 1024. In such cases, specify an alternate port when launching the container by adding the environment variable, for example:

```
-e 'PGADMIN_LISTEN_PORT=5050'
```

Don't forget to adjust any host-container port mapping accordingly.

/var/lib/pgadmin

This is the working directory in which pgAdmin stores session data, user files, configuration files, and its configuration database. Mapping this directory onto the host machine gives you an easy way to maintain configuration between invocations of the container.

/pgadmin4/config_local.py

This file can be used to override configuration settings in pgAdmin.

Settings found in config.py can be overridden with deployment specific values if required. Settings in config_local.py will also override anything specified in the container environment through *PGADMIN_CONFIG_*

prefixed variables.

/pgadmin4/servers.json

If this file is mapped, server definitions found in it will be loaded at launch time. This allows connection information to be pre-loaded into the instance of pgAdmin in the container. Note that server definitions are only loaded on first launch, i.e. when the configuration database is created, and not on subsequent launches using the same configuration database.

/certs/server.cert

If TLS is enabled, this file will be used as the servers TLS certificate.

/certs/server.key

If TLS is enabled, this file will be used as the key file for the servers TLS

certificate.

Examples ↴

Run a simple container over port 80:

```
docker pull dpage/pgadmin4
```

```
docker run -p 80:80 \
```

```
-e 'PGADMIN_DEFAULT_EMAIL=user@domain.com' \
```

```
-e 'PGADMIN_DEFAULT_PASSWORD=SuperSecret' \
```

```
-d dpage/pgadmin4
```

Run a simple container over port 80, setting some configuration options:

```
docker pull dpage/pgadmin4
```

```
docker run -p 80:80 \
```

```
-e 'PGADMIN_DEFAULT_EMAIL=user@domain.com' \
```

```
-e 'PGADMIN_DEFAULT_PASSWORD=SuperSecret' \
```

```
-e
```

```
'PGADMIN_CONFIG_ENHANCED_COOKIE_PROTECTION=True' \
```

```
-e 'PGADMIN_CONFIG_LOGIN_BANNER="Authorised users only!"'
```

```
\
```

```
-e 'PGADMIN_CONFIG_CONSOLE_LOG_LEVEL=10' \
-d dpage/pgadmin4
```

Run a TLS secured container using a shared config/storage directory in

/private/var/lib/pgadmin on the host, and servers pre-loaded from /tmp/servers.json on the host:

```
docker pull dpage/pgadmin4
```

```
docker run -p 443:443 \
```

```
-v /private/var/lib/pgadmin:/var/lib/pgadmin \
```

```
-v /path/to/certificate.cert:/certs/server.cert \
```

```
-v /path/to/certificate.key:/certs/server.key \
```

```
-v /tmp/servers.json:/pgadmin4/servers.json \
```

```
-e 'PGADMIN_DEFAULT_EMAIL=user@domain.com' \
```

```
-e 'PGADMIN_DEFAULT_PASSWORD=SuperSecret' \
```

```
-e 'PGADMIN_ENABLE_TLS=True' \
```

```
-d dpage/pgadmin4
```

Reverse Proxying

Sometimes it's desirable to have users connect to pgAdmin through a reverse proxy rather than directly to the container it's running in. The following examples show how this can be achieved. With traditional reverse proxy servers such as [Nginx](#), pgAdmin is running in a container on the same host, with port 5050 on the host mapped to port 80 on the container, for example:

```
docker pull dpage/pgadmin4  
docker run -p 5050:80 \  
-e "PGADMIN_DEFAULT_EMAIL=user@domain.com" \  
-e "PGADMIN_DEFAULT_PASSWORD=SuperSecret" \  
-d dpage/pgadmin4
```

pgAdmin X-Forwarded-* Configuration [¶](#)

pgAdmin needs to understand how many proxies set each header so it knows what values to trust. The configuration parameters for the X-Forwarded-* options which are used for this purpose are shown below, along with their default values.

pgAdmin is configured by default to be able to run behind a reverse proxy even on a non-standard port and these config options don't normally need to be changed. If you're running an unusual configuration (such as multiple reverse proxies) you can adjust the configuration to suit.

Number of values to trust for X-Forwarded-For
PROXY_X_FOR_COUNT = 1

Number of values to trust for X-Forwarded-Proto.

PROXY_X_PROTO_COUNT = 0

Number of values to trust for X-Forwarded-Host.

PROXY_X_HOST_COUNT = 0

Number of values to trust for X-Forwarded-Port.

PROXY_X_PORT_COUNT = 1

Number of values to trust for X-Forwarded-Prefix.

```
PROXY_X_PREFIX_COUNT = 0
```

HTTP via Nginx 

A configuration similar to the following can be used to create a simple HTTP reverse proxy listening for all hostnames with [Nginx](#):

```
server {  
    listen 80;  
  
    server_name _;  
  
    location / {  
  
        proxy_set_header Host $host;  
  
        proxy_pass http://localhost:5050/;  
  
        proxy_redirect off;  
  
    }  
  
}
```

If you wish to host pgAdmin under a subdirectory rather than on the root of the server, you must specify the location and set the *X-Script-Name* header which tells the pgAdmin container how to rewrite paths: **server {**

```
listen 80;  
  
server_name _;  
  
location /pgadmin4/ {  
  
    proxy_set_header X-Script-Name /pgadmin4; proxy_set_header  
    Host $host;
```

```
proxy_pass http://localhost:5050/;

proxy_redirect off;

}

}
```

If Nginx is also running in a container, there is no need to map the pgAdmin port to the host, provided the two containers are running in the same Docker network. In such a configuration, the *proxy_pass* option would be changed to point to the pgAdmin container within the Docker network.

HTTPS via Nginx

The following configuration can be used to serve pgAdmin over HTTPS to the user whilst the backend container is serving plain HTTP to the proxy server. In this configuration we not only set *X-Script-Name*, but also *X-Scheme* to tell the pgAdmin server to generate any URLs using the correct scheme. A redirect from HTTP to HTTPS is also included. The certificate and key paths may need to be adjusted as appropriate to the specific deployment:

```
server {

listen 80;

return 301 https://$host$request_uri;

}

server {

listen 443;

server_name _;

ssl_certificate /etc/nginx/server.crt;
```

```
ssl_certificate_key /etc/nginx/server.key; ssl on;

ssl_session_cache builtin:1000 shared:SSL:10m; ssl_protocols
TLSv1 TLSv1.1 TLSv1.2;

ssl_ciphers

HIGH:!aNULL:!eNULL:!EXPORT:!CAMELLIA:!DES:!MD5:!PSK:!RC4
; ssl_prefer_server_ciphers on;

location /pgadmin4/ {

proxy_set_header X-Script-Name /pgadmin4; proxy_set_header
X-Scheme $scheme;

proxy_set_header Host $host;

proxy_pass http://localhost:5050/;

proxy_redirect off;

}

}
```

Traefik

Configuring [Traefik](#) is straightforward for either HTTP or HTTPS when running pgAdmin in a container as it will automatically configure itself to serve content from containers that are running on the local machine, virtual hosting them at `<container_name>`. `<domain_name>`, where the domain name is that specified in the Traefik configuration. The container is typically launched per the example below:

```
docker pull dpage/pgadmin4

docker run --name "pgadmin4" \
```

```
-e "PGADMIN_DEFAULT_EMAIL=user@domain.com" \
-e "PGADMIN_DEFAULT_PASSWORD=SuperSecret" \
-d dpage/pgadmin4
```

Note that the TCP/IP port has not been mapped to the host as it was in the Nginx example, and the container name has been set to a known value as it will be used as the hostname and may need to be added to the DNS

zone file.

The following configuration will listen on ports 80 and 443, redirecting 80

to 443, using the default certificate shipped with Traefik. See the Traefik documentation for options to use certificates from LetsEncrypt or other issuers.

```
defaultEntryPoints = ["http", "https"]
```

[entryPoints]

[entryPoints.http]

```
address = ":80"
```

[entryPoints.http.redirect]

```
entryPoint = "https"
```

[entryPoints.https]

```
address = ":443"
```

[entryPoints.https.tls]

[docker]

```
domain = "domain_name"
```

```
watch = true
```

If you wish to host pgAdmin under a subdirectory using Traefik, the configuration changes are typical y made to the way the container is launched and not to Traefik itself. For example, to host pgAdmin under

/pgadmin4/ instead of at the root directory, the Traefik configuration above may be used if the container is launched like this: docker pull dpage/pgadmin4

```
docker run --name "pgadmin4" \
-e "PGADMIN_DEFAULT_EMAIL=user@domain.com" \
-e "PGADMIN_DEFAULT_PASSWORD=SuperSecret" \
-e "SCRIPT_NAME=/pgadmin4" \
-l "traefik.frontend.rule=PathPrefix:/pgadmin4" \
-d dpage/pgadmin4
```

The `SCRIPT_NAME` environment variable has been set to tel the container it is being hosted under a subdirectory (in the same way as the `X-Script-Name` header is used with Nginx), and a label has been added to tel Traefik to route requests under the subdirectory to this container.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Connecting To A Server](#) > [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
 - [Server Group Dialog](#)
 - [Server Dialog](#)
 - [Master Password](#)
 - [Connect to Server](#)
 - [Connection Error](#)
 - [Import/Export Servers](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Connection Error

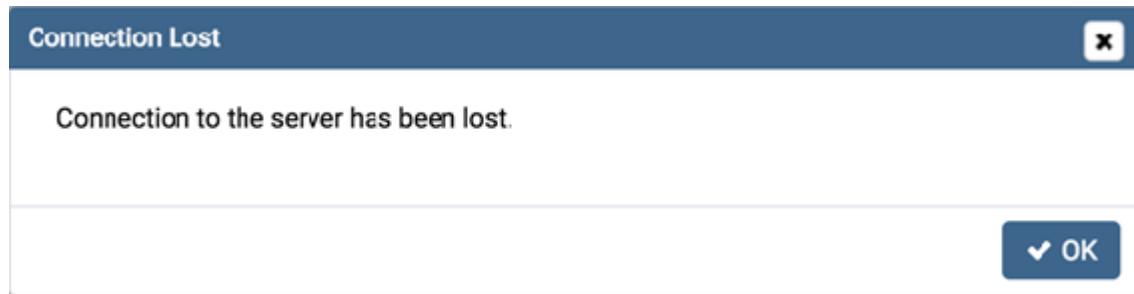
When connecting to a PostgreSQL server, you may get an error message. If you encounter an error message, please review the message carefully; each error message attempts to incorporate the information you'll need to resolve the problem. For more details about specific errors, please locate the error message in the list below:

Connection to the server has been lost



This error message indicates that the connection attempt has taken longer than the specified threshold; there may be a problem with the connection properties provided on the Server dialog, network connectivity issues, or the server may not be running.

could not connect to Server: Connection refused



Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Connecting To A Server](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Server Group Dialog](#)

[Server Dialog](#)



[Master Password](#)

[FAQ](#)

[Connect to Server](#)

Connection Error



[Import/Export Servers](#)

[Managing Cluster Objects](#)

[Latest Docs](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Management Basics](#)

[Backup and Restore](#)

[Ask a](#)

[Developer Tools](#)

[Question](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)



[Release Notes](#)

[Licence](#)

[Report a Bug](#)

Connection Error 



When connecting to a PostgreSQL server, you may get an error message.

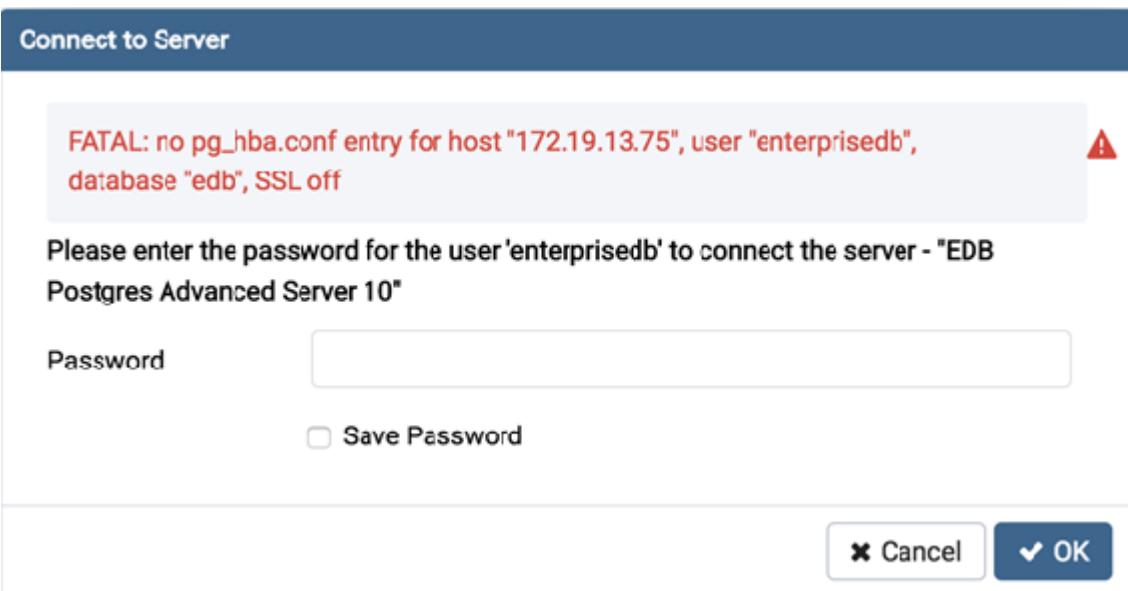
[Screenshots](#)

If you encounter an error message, please review the message carefully; each error message attempts to incorporate the information you'll need to resolve the problem. For more details about specific errors, please locate the error message in the list below:

Connection to the server has been lost

This error message indicates that the connection attempt has taken longer than the specified threshold; there may be a problem with the connection properties provided on the *Server* dialog, network connectivity issues, or the server may not be running.

could not connect to Server: Connection refused



If pgAdmin displays this message, there are two possible reasons for this: the database server isn't running - simply start it.

the server isn't configured to accept TCP/IP requests on the address shown.

For security reasons, a PostgreSQL server “out of the box” doesn’t listen on TCP/IP ports. Instead, it must be enabled to listen for TCP/IP requests.

This can be done by adding `listen_addresses='*'`; this will make the server accept connections on any IP interface.

For further information, please refer to the PostgreSQL documentation about [runtime configuration](#).

FATAL: no pg_hba.conf entry

If pgAdmin displays this message when connecting, your server can be contacted correctly over the network, but is not configured to accept your connection. Your client has not been detected as a legal user for the database.

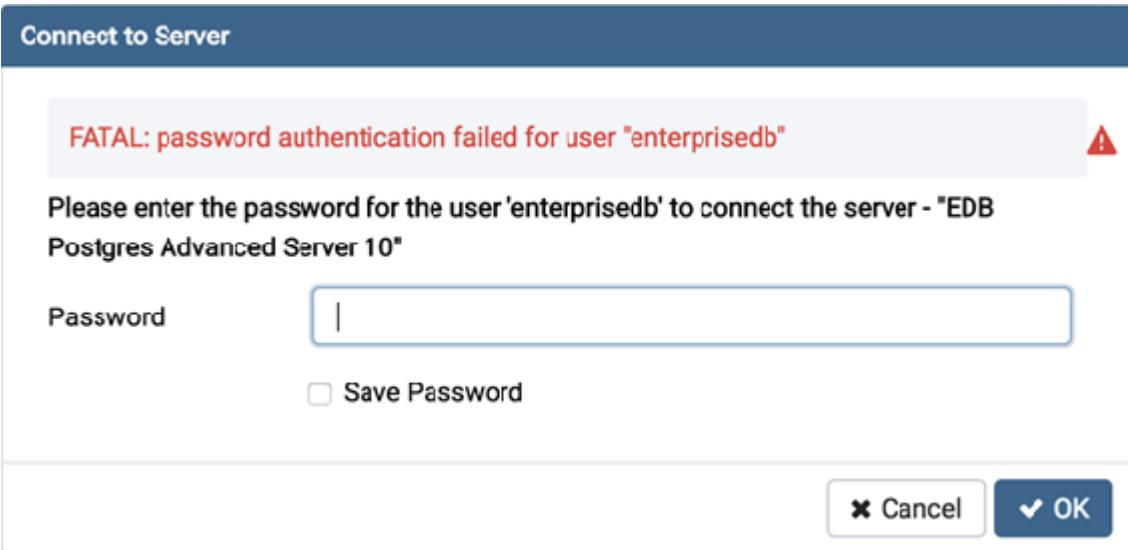
To connect to a server, the pg_hba.conf file on the database server must be configured to accept connections from the host of the pgAdmin client.

Modify the pg_hba.conf file on the database server host, and add an entry in the form:

```
host template1 postgres 192.168.0.0/24 md5 for an IPV4 network  
host template1 postgres ::ffff:192.168.0.0/120 md5 for an IPV6
```

network

For more information, please refer to the PostgreSQL documentation about [client authentication](#).



FATAL: password authentication failed

The *password authentication failed for user* error message indicates there may be a problem with the password you entered. Retry the password to confirm you entered it correctly. If the error message returns, make sure that you have the correct password, that you are authorized to access the server, and that the access has been correctly configured in the server's postgresql.conf configuration file.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4.4.19 documentation](#) »

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
 - [Check Dialog](#)
 - [Column Dialog](#)
 - [Compound Trigger Dialog](#)
 - [Exclusion Constraint Dialog](#)
 - [Foreign key Dialog](#)
 - [Index Dialog](#)
 - [Primary key Dialog](#)
 - [Rule Dialog](#)
 - [Table Dialog](#)
 - [Trigger Dialog](#)
 - [Unique Constraint Dialog](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Creating or Modifying a Table

pgAdmin 4 provides dialogs that allow you to modify all table properties and attributes.

To access a dialog that allows you to create a database object, right-click on the object type in the pgAdmin tree control, and select the Create option for that object. For example, to create a new database, right-click on the Casts node, and select Create Cast...

- [Check Dialog](#)
- [Column Dialog](#)
- [Compound Trigger Dialog](#)
- [Exclusion Constraint Dialog](#)
- [Foreign key Dialog](#)
- [Index Dialog](#)
- [Primary key Dialog](#)
- [Rule Dialog](#)
- [Table Dialog](#)
- [Trigger Dialog](#)

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#)
[Development](#)

[pgAdmin 4 4.19 documentation](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



Creating or Modifying a Table

[FAQ](#)

[Check Dialog](#)

[Column Dialog](#)

[□](#)

[Compound Trigger Dialog](#)

[Exclusion Constraint Dialog](#)

[Latest Docs](#)

[Foreign key Dialog](#)

[Index Dialog](#)

[□](#)

[Primary key Dialog](#)

[Rule Dialog](#)

[Ask a](#)

[Table Dialog](#)

[Question](#)

[Trigger Dialog](#)

[Unique Constraint Dialog](#)

[□](#)

[Management Basics](#)

[Backup and Restore](#)

[Report a Bug](#)

[Developer Tools](#)

[pgAgent](#)



[pgAdmin Project Contributions](#)

[Release Notes](#)

[Screenshots](#)

[Licence](#)

[Creating or Modifying a Table](#)

pgAdmin 4 provides dialogs that allow you to modify all table properties and attributes.

To access a dialog that allows you to create a database object, right-click on the object type in the pgAdmin tree control, and select the *Create* option for that object. For example, to create a new database, right-click on the *Casts* node, and select *Create Cast...*

[Check Dialog](#)

[Column Dialog](#)

[Compound Trigger Dialog](#)

[Exclusion Constraint Dialog](#)

[Foreign key Dialog](#)

[Index Dialog](#)

[Primary key Dialog](#)

[Rule Dialog](#)

[Table Dialog](#)

[Trigger Dialog](#)

Unique Constraint Dialog

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [pgAgent](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
 - [Using pgAgent](#)
 - [Installing pgAgent](#)
 - [Creating a pgAgent Job](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Creating a pgAgent Job

pgAgent is a scheduling agent that runs and manages jobs; each job consists of steps and schedules.

To create or manage a job, use the pgAdmin tree control to browse to the server on which the pgAgent database objects were created. The tree control will display a *pgAgent Jobs* node, under which currently defined jobs are displayed. To add a new job, right click on the *pgAgent Jobs* node, and select *Create pgAgent Job...* from the context menu.

When the pgAgent dialog opens, use the tabs on the pgAgent Job dialog to define the steps and schedule that make up a pgAgent job.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [pgAgent](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[FAQ](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)



[pgAgent](#)

[Latest Docs](#)

[Using pgAgent](#)

[Installing pgAgent](#)



Creating a pgAgent Job

[pgAdmin Project Contributions](#)

[Ask a](#)

[Release Notes](#)

[Question](#)

[Licence](#)



Creating a pgAgent Job

[Report a Bug](#)

pgAgent is a scheduling agent that runs and manages jobs; each job consists of steps and schedules.

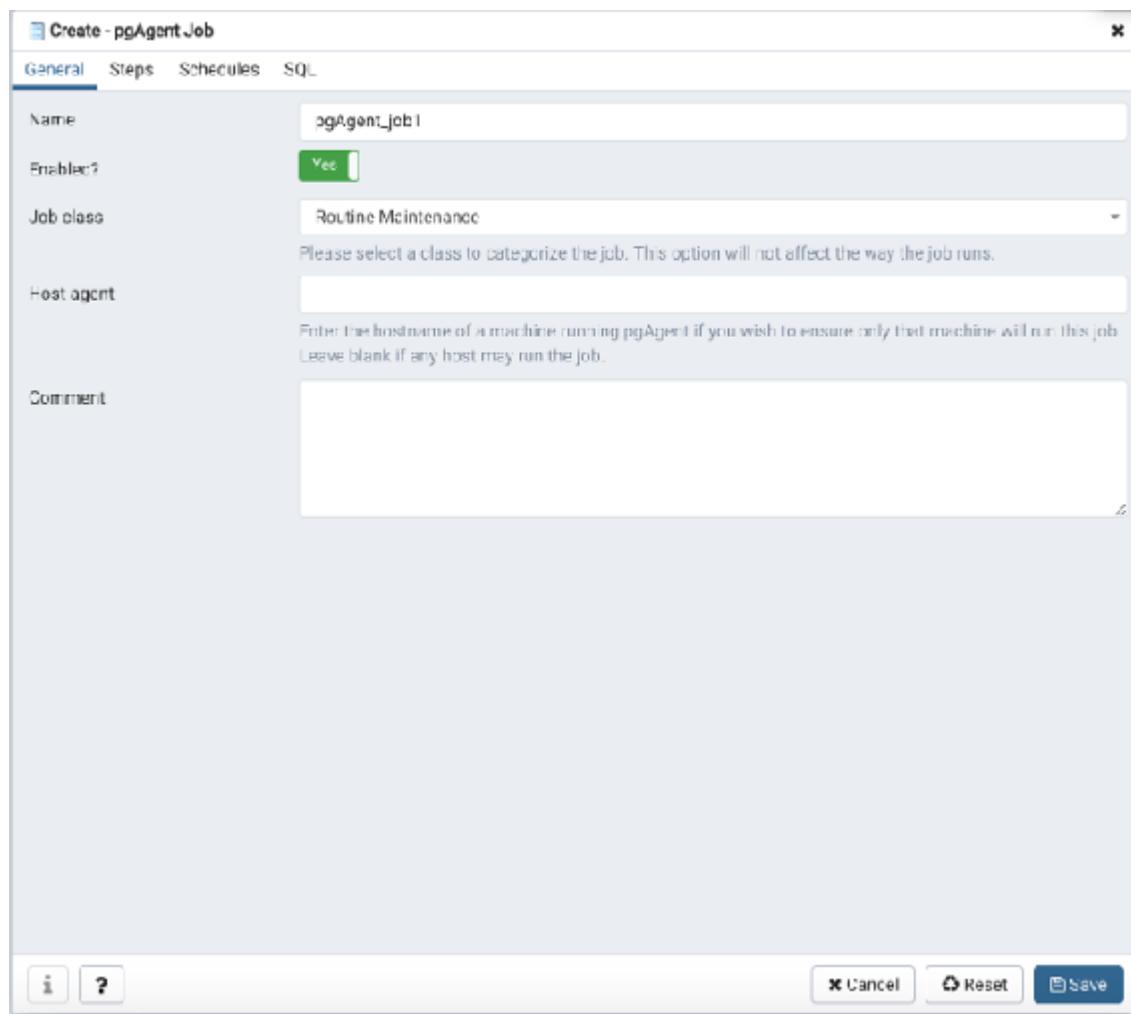


To create or manage a job, use the pgAdmin tree control to browse to the

Screenshots

server on which the pgAgent database objects were created. The tree control will display a *pgAgent Jobs* node, under which currently defined jobs are displayed. To add a new job, right click on the *pgAgent Jobs* node, and select *Create pgAgent Job...* from the context menu.

When the pgAgent dialog opens, use the tabs on the *pgAgent Job* dialog to define the steps and schedule that make up a pgAgent job.



Use the fields on the *General* tab to provide general information about a job:

Provide a name for the job in the *Name* field.

Move the *Enabled* switch to the *Yes* position to enable a job, or *No* to disable a job.

Use the *Job Class* drop-down to select a class (for job categorization).

Use the *Host Agent* field to specify the name of a machine that is running pgAgent to indicate that only that machine may execute the job. Leave the field blank to specify that any machine may perform the job.

Note

It is not always obvious what value to specify for the Host Agent in order to target a job step to a specific machine. With pgAgent running on the required machines and connected to the scheduler database, you can use the following query to view the hostnames as reported by each agent:

SELECT jagstation **FROM** pgagent.pga_jobagent Use the hostname exactly as reported by the query in the Host Agent field.

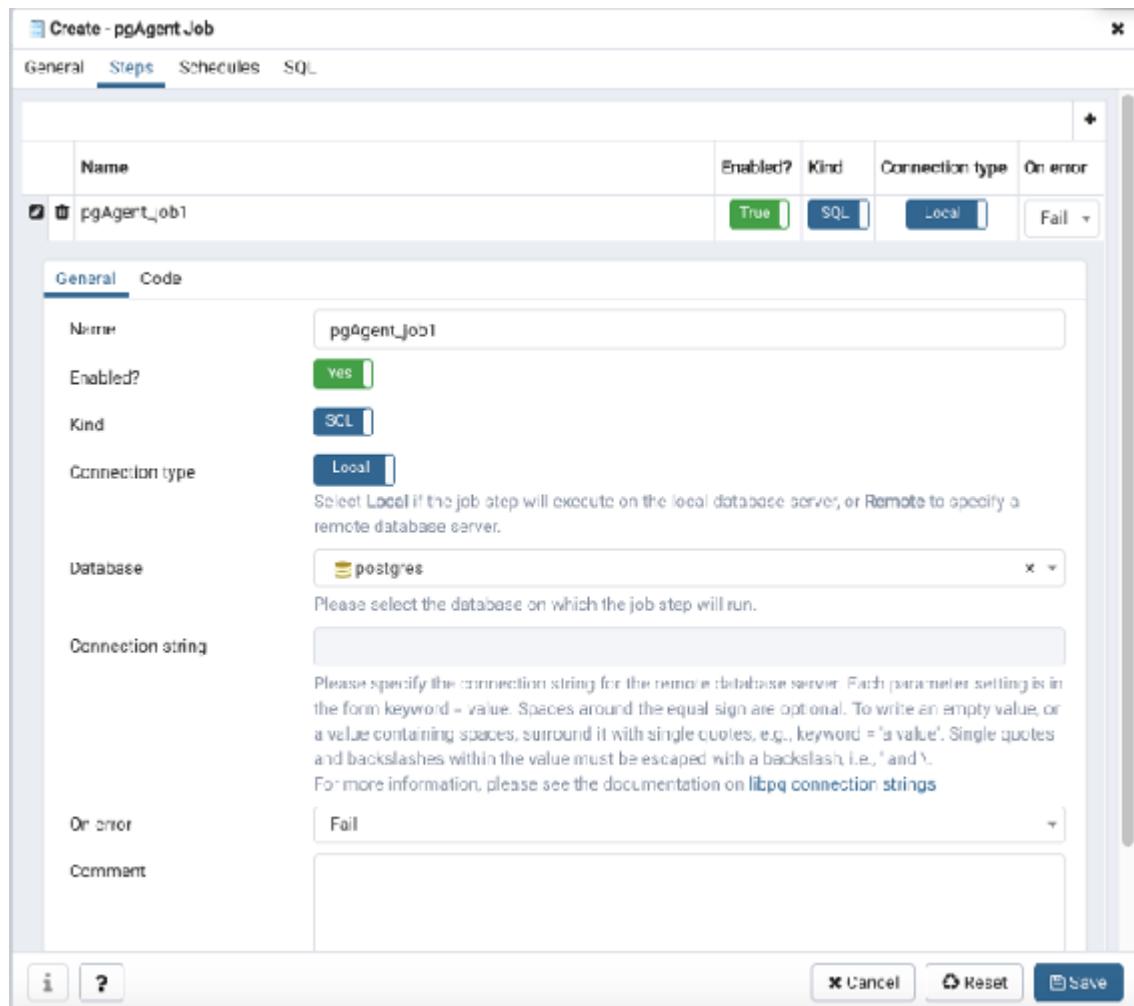
Use the *Comment* field to store notes about the job.

Create - pgAgent Job

General Steps Schedules SQL

	Name	Enabled?	Kind	Connection type	On error
<input checked="" type="checkbox"/>	pg_Agent_job1_steps	<input checked="" type="button"/> True	<input checked="" type="button"/> SCL	<input checked="" type="button"/> Local	<input checked="" type="button"/> Fail

Cancel Reset Save



Use the *Steps* tab to define and manage the steps that the job will perform. Click the Add icon (+) to add a new step; then click the compose icon (located at the left side of the header) to open the step definition dialog:

Use fields on the step definition dialog to define the step: Provide a name for the step in the *Name* field; please note that steps will be performed in alphanumeric order by name.

Use the *Enabled* switch to include the step when executing the job (*True*) or to disable the step (*False*).

Use the *Kind* switch to indicate if the job step invokes SQL code (*SQL*) or a batch script (*Batch*).

If you select *SQL*, use the *Code* tab to provide SQL code for the step.

If you select *Batch*, use the *Code* tab to provide the batch script that will be executed during the step.

Note

The fields *Connection type*, *Database* and *Connection string* are only applicable when *SQL* is selected because *Batch* cannot be run on remote servers.

Use the *Connection type* switch to indicate if the step is performed on a local server (*Local*) or on a remote host (*Remote*). If you specify a remote connection should be used for the step, the *Connection string* field will be enabled, and you must provide a libpq-style connection string.

Use the *Database* drop-down to select the database on which the job step will be performed.

Use the *Connection string* field to specify a libpq-style connection string to the remote server on which the step will be performed. For more information about writing a connection string, please see the

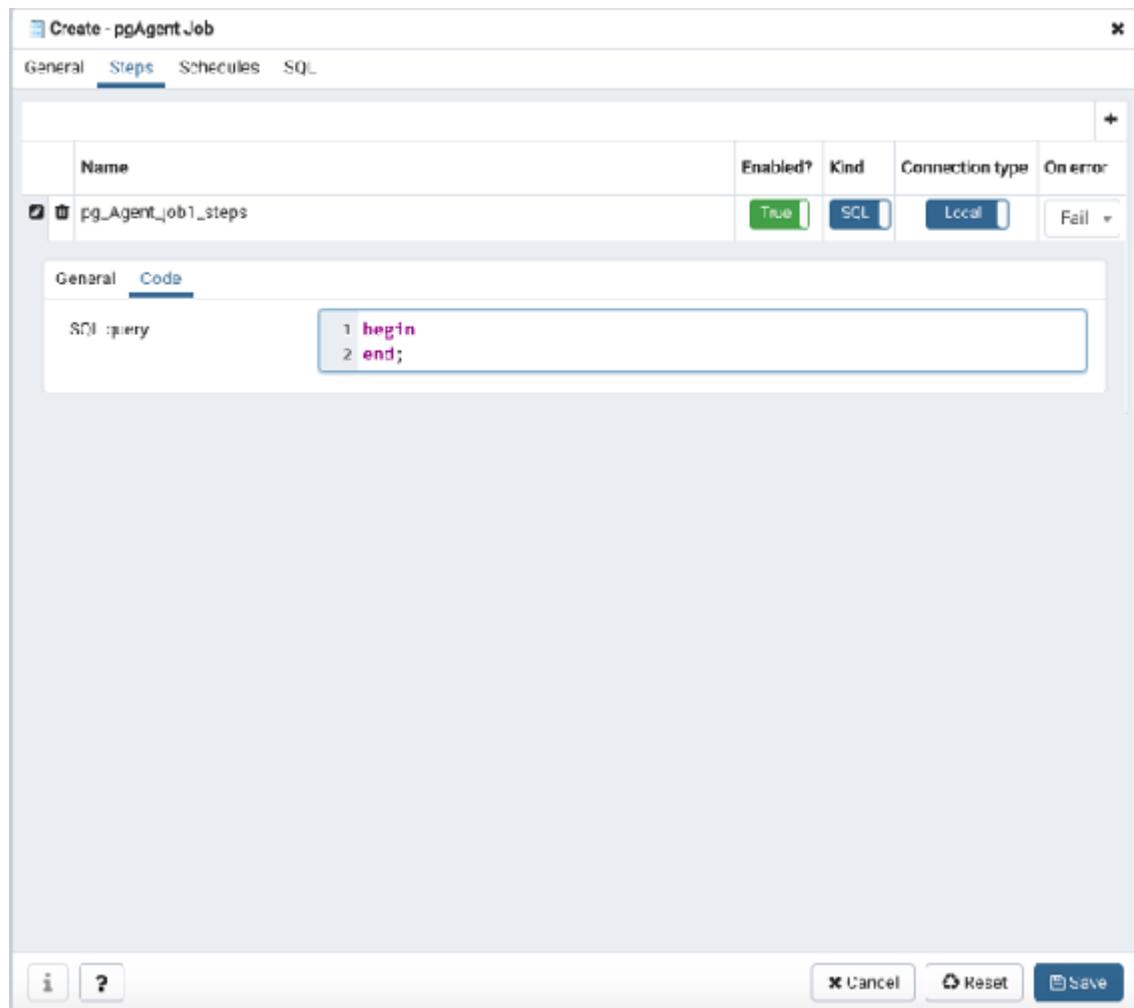
[PostgreSQL documentation](#).

Use the *On error* drop-down to specify the behavior of pgAgent if it encounters an error while executing the step. Select from: *Fail* - Stop the job if you encounter an error while processing this step.

Success - Mark the step as completing successfully, and continue.

Ignore - Ignore the error, and continue.

Use the *Comment* field to provide a comment about the step.



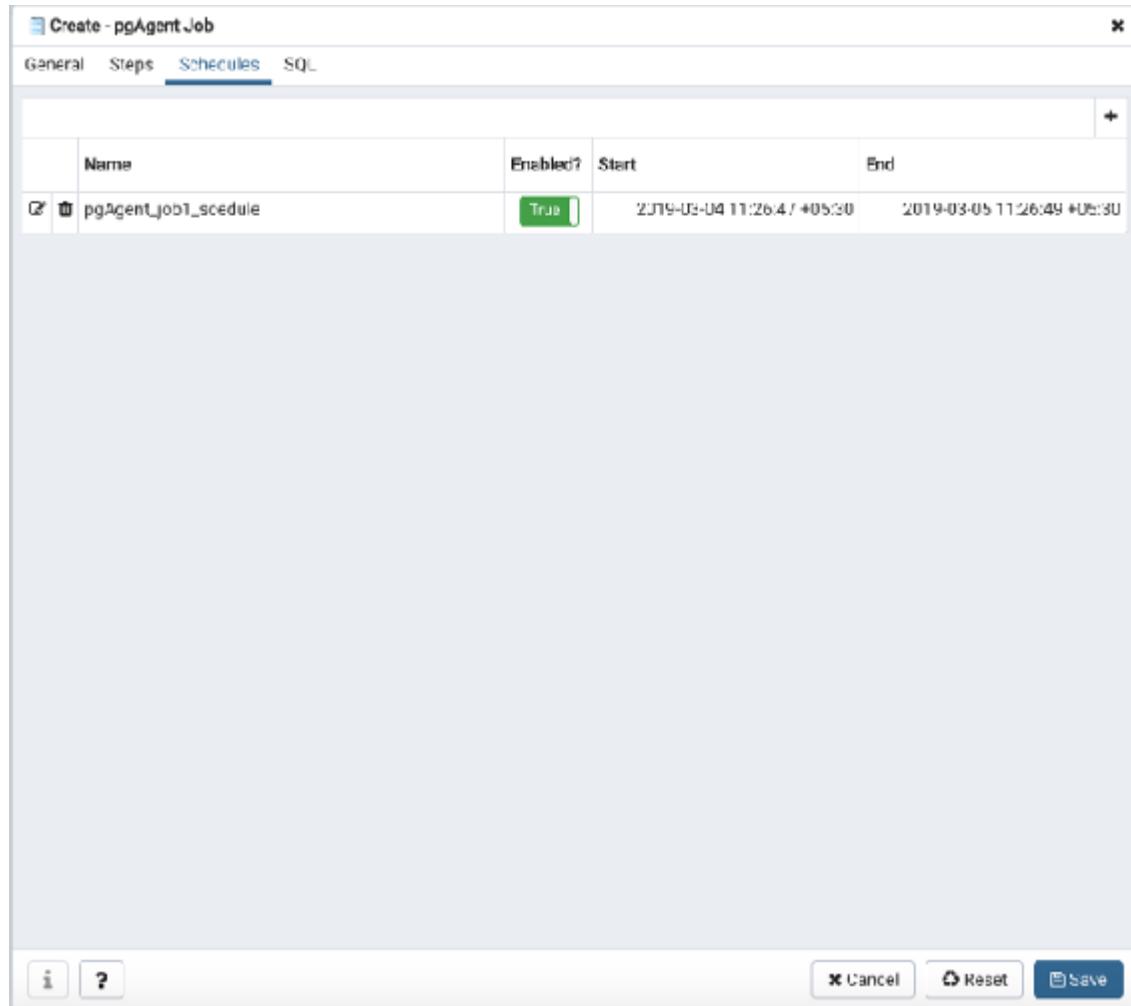
Use the context-sensitive field on the step definition dialog's *Code* tab to provide the SQL code or batch script that will be executed during the step:

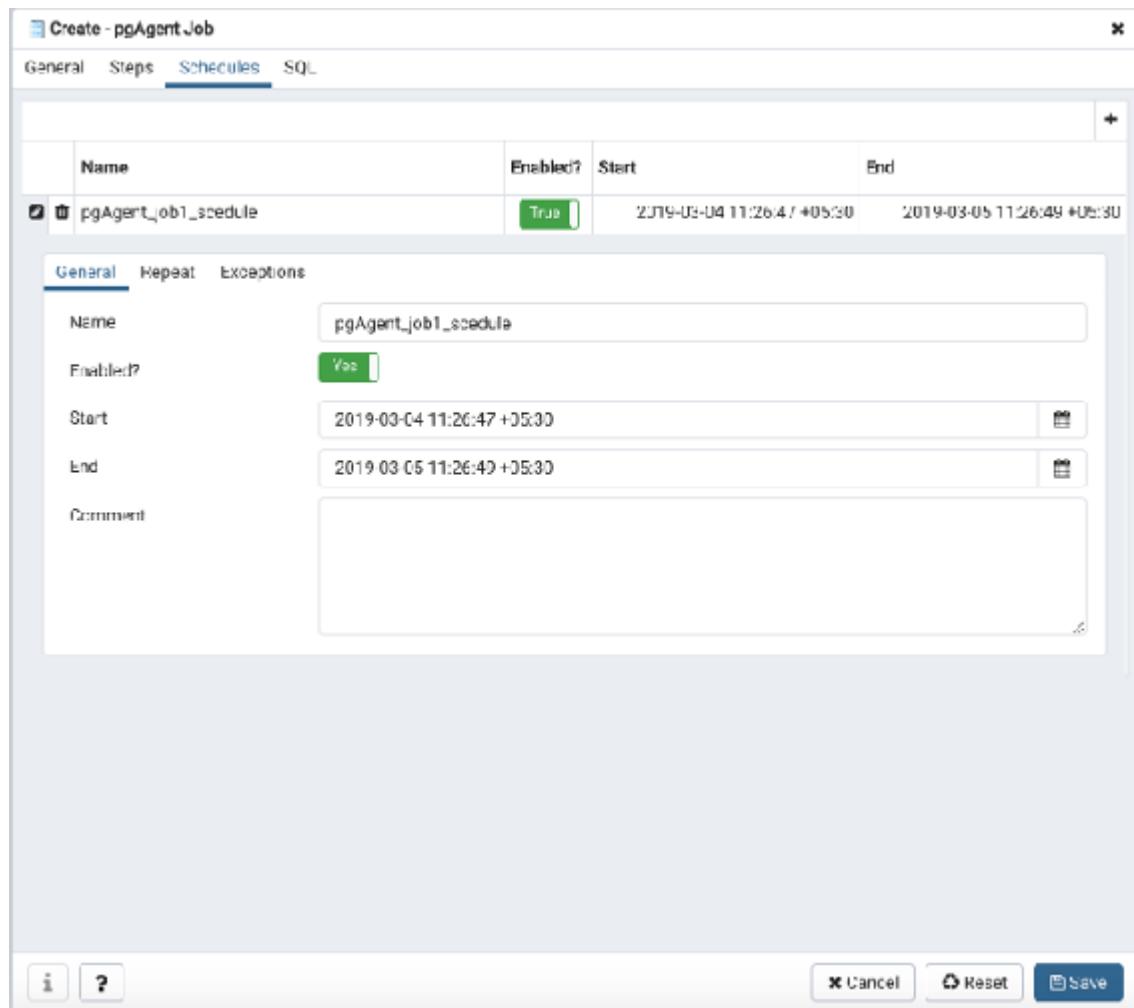
If the step invokes SQL code, provide one or more SQL statements in the *SQL query* field.

If the step performs a batch script, provide the script in the *Script* field. If you are running on a Windows server, standard batch file syntax must be used. When running on a Linux server, any shell script may be used, provided that a suitable interpreter is specified on the first line (e.g. `#!/bin/sh`).

When you've provided all of the information required by the step, click the compose icon to close the step definition dialog. Click the

add icon (+) to add each additional step, or select the *Schedules* tab to define the job schedule.





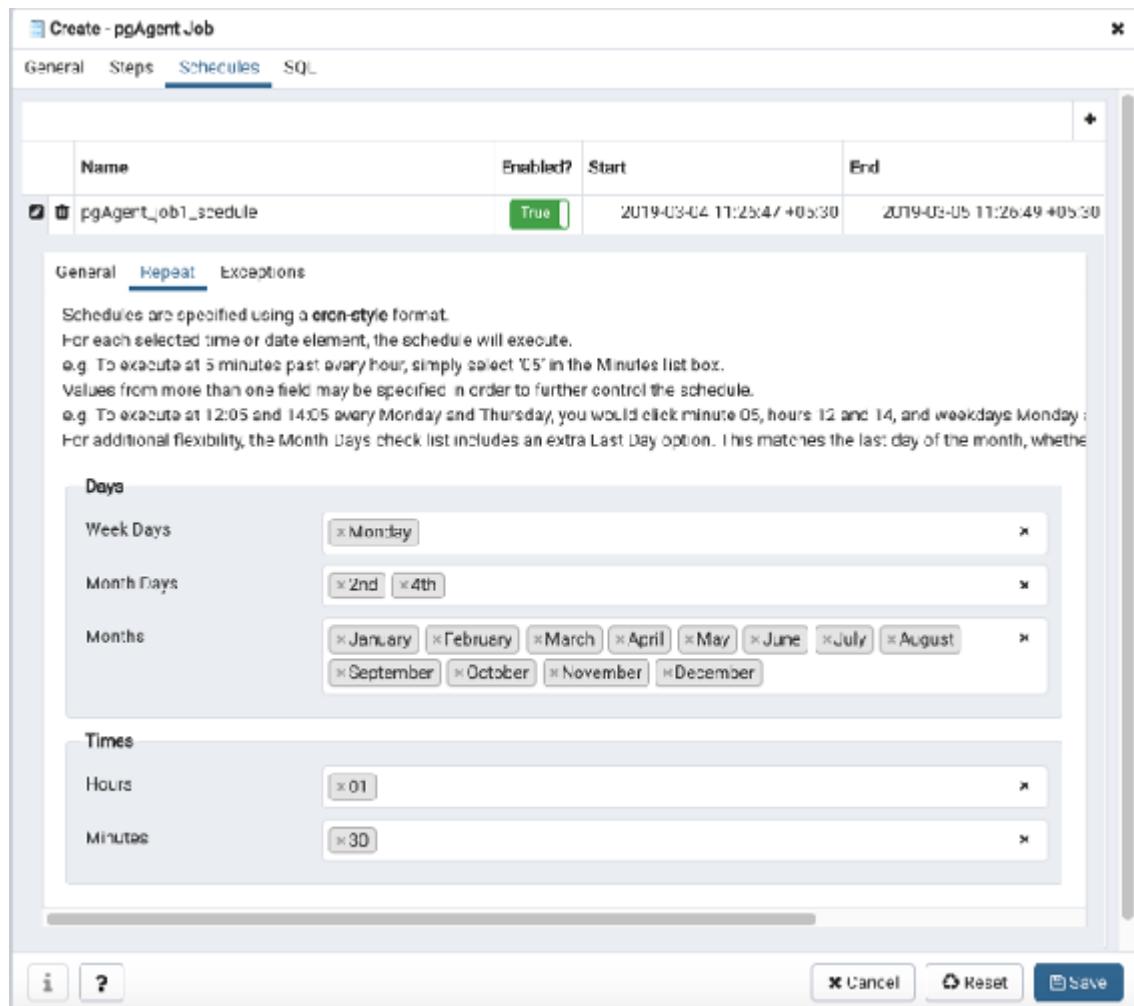
Click the Add icon (+) to add a schedule for the job; then click the compose icon (located at the left side of the header) to open the schedule definition dialog:

Use the fields on the schedule definition tab to specify the days and times at which the job will execute.

Provide a name for the schedule in the *Name* field.

Use the *Enabled* switch to indicate that pgAgent should use the schedule (Yes) or to disable the schedule (No).

Use the calendar selector in the *Start* field to specify the starting date and time for the schedule.



Use the calendar selector in the *End* field to specify the ending date and time for the schedule.

Use the *Comment* field to provide a comment about the schedule.

Select the *Repeat* tab to define the days on which the schedule will execute.

Use the fields on the *Repeat* tab to specify the details about the schedule in a cron-style format. The job will execute on each date or time element selected on the *Repeat* tab.

Click within a field to open a list of valid values for that field; click on a specific value to add that value to the list of selected values for the field.

To clear the values from a field, click the X located at the right-side of the field.

Use the fields within the *Days* box to specify the days on which the job wil execute:

Use the *Week Days* field to select the days on which the job wil execute.

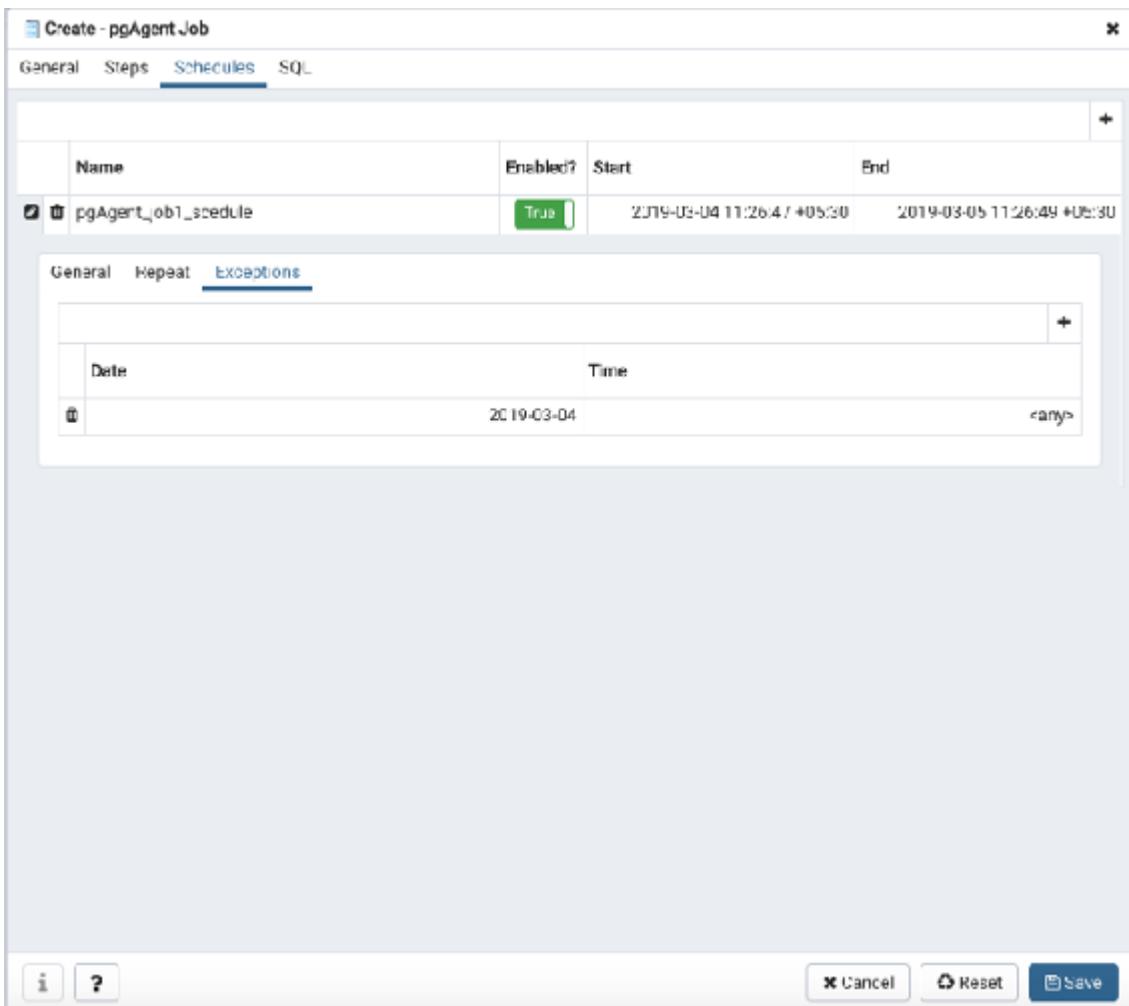
Use the *Month Days* field to select the numeric days on which the job wil execute. Specify the *Last Day* to indicate that the job should be performed on the last day of the month, irregardless of the date.

Use the *Months* field to select the months in which the job wil execute.

Use the fields within the *Times* box to specify the times at which the job wil execute:

Use the *Hours* field to select the hour at which the job wil execute.

Use the *Minutes* field to select the minute at which the job wil execute.



Select the *Exceptions* tab to specify any days on which the schedule will not execute.

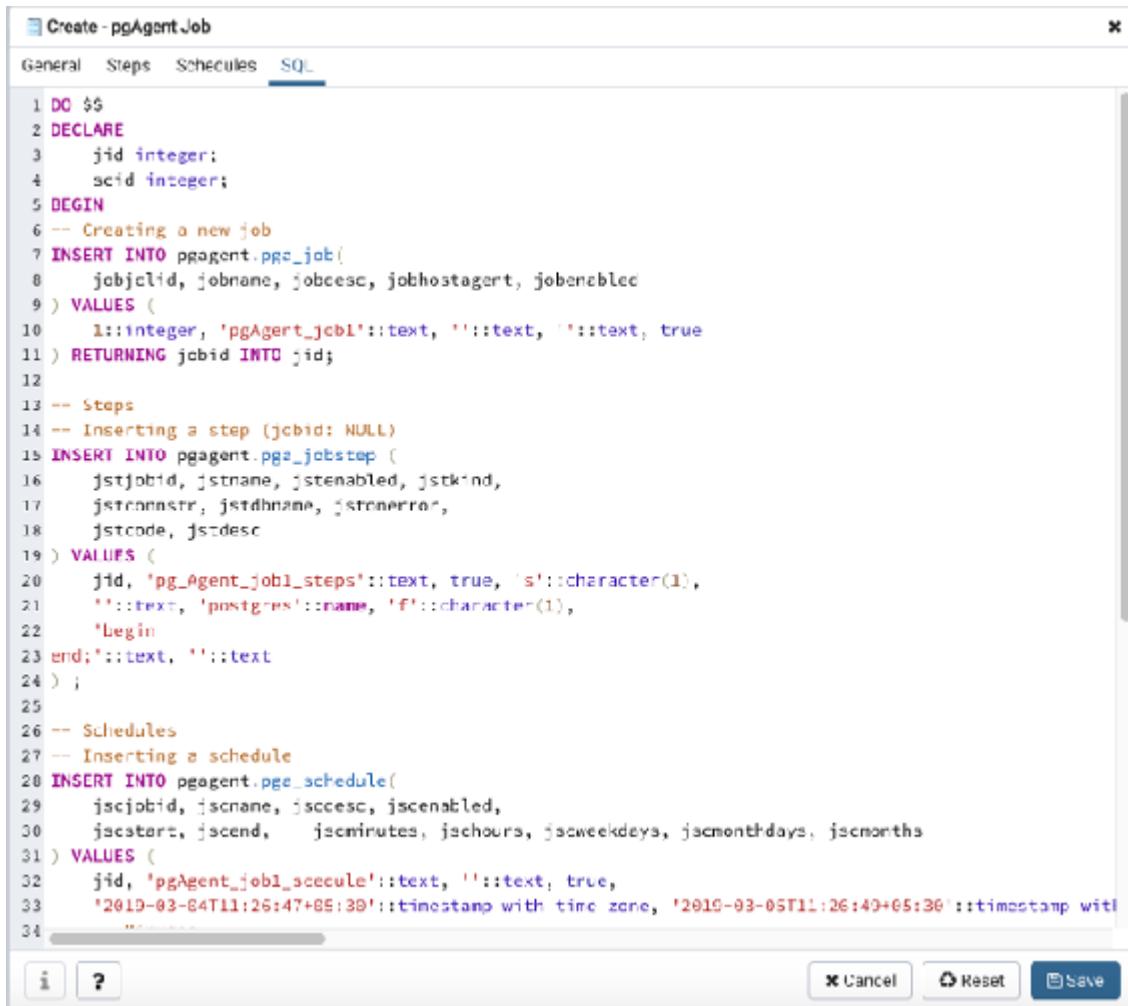
Use the fields on the *Exceptions* tab to specify days on which you wish the job to not execute; for example, you may wish for jobs to not execute on national holidays.

Click the Add icon (+) to add a row to the exception table, then: Click within the *Date* column to open a calendar selector, and select a date on which the job will not execute. Specify <Any> in the *Date* column to indicate that the job should not execute on any day at the time selected.

Click within the *Time* column to open a time selector, and specify a time on which the job will not execute. Specify <Any> in the *Time*

column to indicate that the job should not execute at any time on the day selected.

When you've finished defining the schedule, you can use the **SQL** tab to review the code that will create or modify your job.



The screenshot shows a software interface titled "Create - pgAgent.Job". The window has tabs for "General", "Steps", "Schedules", and "SQL". The "SQL" tab is currently selected and displays the following SQL code:

```
1 DO $$  
2 DECLARE  
3     jid integer;  
4     scid integer;  
5 BEGIN  
6 -- Creating a new job  
7 INSERT INTO pgagent.pge_job(  
8     jobclid, jobname, jobccesc, jobhostagent, jobenabled  
9 ) VALUES (  
10    1::integer, 'pgAgent_job1 '::text, ''::text, ''::text, true  
11 ) RETURNING jcbid INTO jid;  
12  
13 -- Steps  
14 -- Inserting a step (jcbid: NULL)  
15 INSERT INTO pgagent.pge_jobstep (  
16     jstjobid, jstname, jstenabled, jstkind,  
17     jstmonstr, jstdhname, jstmonerrr,  
18     jstcode, jstdesc  
19 ) VALUES (  
20    jid, 'pgAgent_job1_steps'::text, true, 's'::character(1),  
21    ''::text, 'postgres'::name, 'f'::character(1),  
22    'begin'  
23 end;''::text, ''::text  
24 );  
25  
26 -- Schedules  
27 -- Inserting a schedule  
28 INSERT INTO pgagent.pge_schedule(  
29     jscjobid, jscname, jscccesc, jscenabled,  
30     jscstart, jscend,    jscminutes, jshours, jscweekdays, jscmonthdays, jscmonths  
31 ) VALUES (  
32    jid, 'pgAgent_job1_schedule'::text, ''::text, true,  
33    '2010-03-04T11:26:47+05:30'::timestamp with time zone, '2010-03-05T11:26:40+05:30'::timestamp with  
34    ''
```

At the bottom of the dialog box, there are buttons for "Cancel", "Reset", and "Save".

Screenshot of the pgAdmin Properties tab for a job named "pgAgent_job1". The General section is expanded, showing the following details:

Setting	Value
Name	pgAgent_job1
ID	4
Enabled?	Yes
Job class	Routine Maintenance
Host agent	
Created	2019-02-11 11:29:23.557053+05:30
Changed	2019-02-11 11:29:23.557053+05:30
Next run	
Last run	
Last result	Unknown
Running at	Not running currently.
Comment	

Click the **Save** button to save the job definition, or **Cancel** to exit the job without saving. Use the **Reset** button to remove your unsaved entries from the dialog.

After saving a job, the job will be listed under the *pgAgent Jobs* node of the pgAdmin tree control of the server on which it was defined. The *Properties* tab in the main pgAdmin window will display a high-level overview of the selected job, and the *Statistics* tab will show the details of each run of the job.

To modify an existing job or to review detailed information about a job, right-click on a job name, and select *Properties* from the context menu.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Managing Cluster Objects](#) > [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Database Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
 - [Database Dialog](#)
 - [Move Objects Dialog](#)
 - [Resource Group Dialog](#)
 - [Login/Group/Role Dialog](#)
 - [Tablespace Dialog](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Database Dialog

Use the *Database* dialog to define or modify a database. To create a database, you must be a database superuser or have the `CREATE` privilege.

The *Database* dialog organizes the development of a database through the following dialog tabs: *General*, *Definition*, *Security*, and *Parameters*. The *SQL* tab displays the SQL code generated by dialog selections.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#)
[Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Cluster Objects](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Database Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Database Dialog](#)

[Move Objects Dialog](#)



[Resource Group Dialog](#)

[Login/Group Role Dialog](#)

[Latest Docs](#)

[Tablespace Dialog](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[Ask a](#)

[Management Basics](#)

[Question](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)



[pgAdmin Project Contributions](#)

[Report a Bug](#)

[Release Notes](#)

[Licence](#)



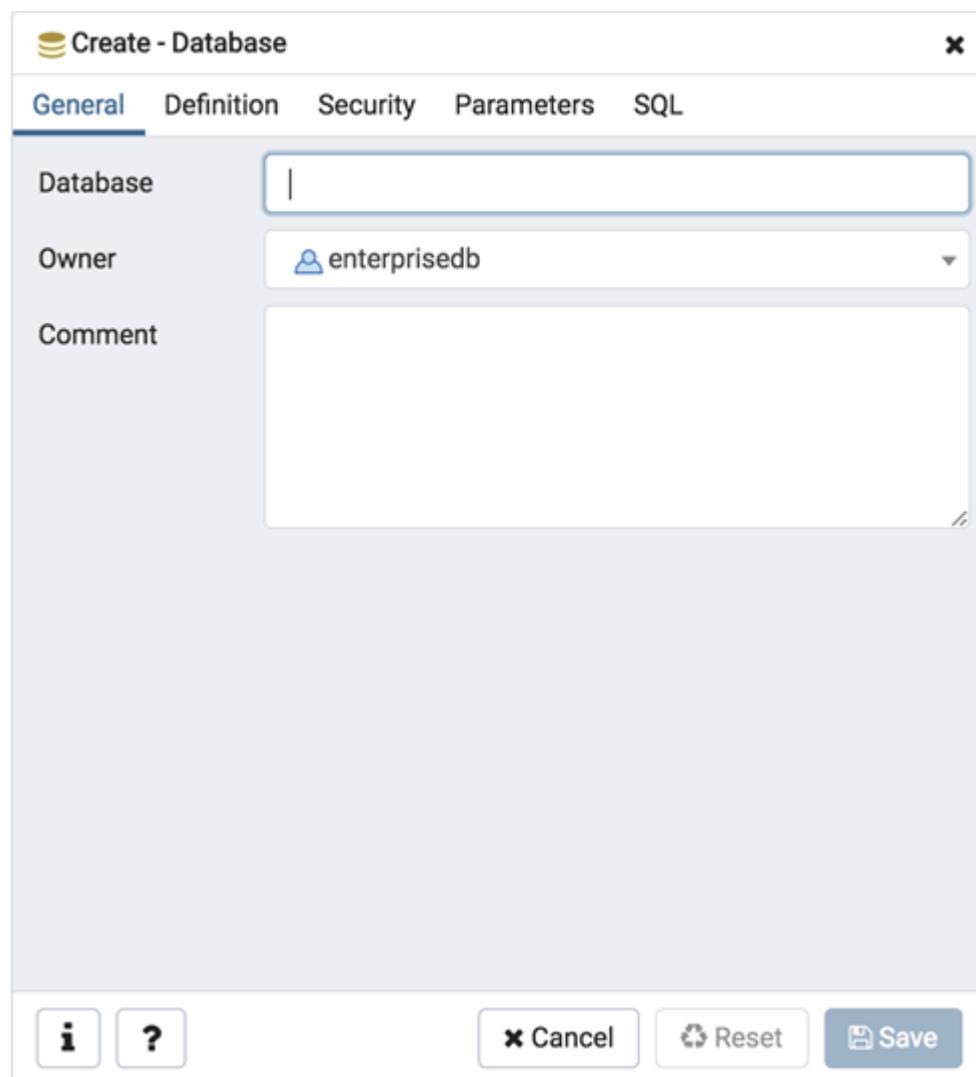
Database Dialog

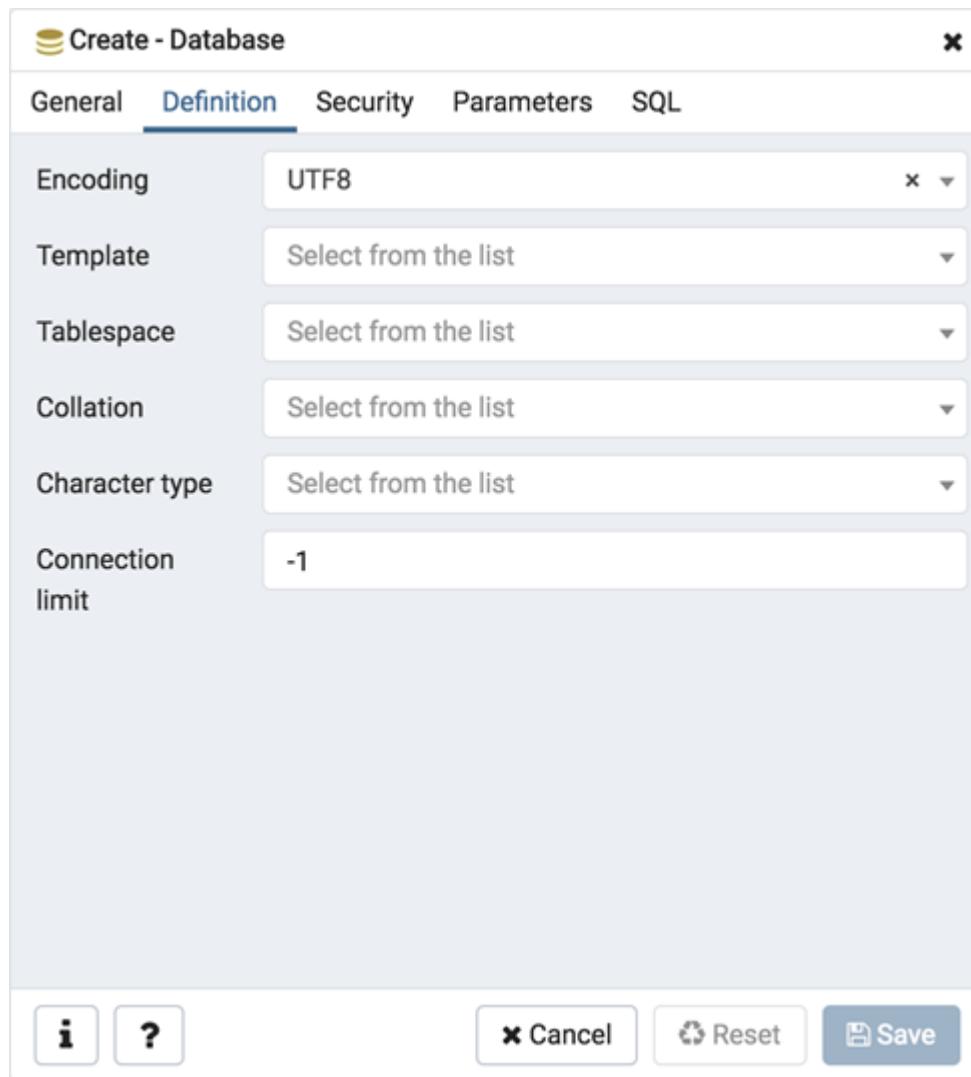
Screenshots

Use the *Database* dialog to define or modify a database. To create a database, you must be a database superuser or have the CREATE privilege.

The *Database* dialog organizes the development of a database through the following dialog tabs: *General*, *Definition*, *Security*, and *Parameters*.

The SQL tab displays the SQL code generated by dialog selections.



A screenshot of the pgAdmin 'Create - Database' dialog box. The 'Definition' tab is selected. The form contains the following fields:

- Encoding: UTF8
- Template: Select from the list
- Tablespace: Select from the list
- Collation: Select from the list
- Character type: Select from the list
- Connection limit: -1

At the bottom are buttons for Help (i), Help (?) (question mark), Cancel (x), Reset (refresh), and Save (disk).

Use the fields in the *General* tab to identify the database: Use the *Database* field to add a descriptive name for the database.

The name will be displayed in the *pgAdmin* tree control.

Select the owner of the database from the drop-down listbox in the *Owner* field.

Store notes about the database in the *Comment* field.

Click the *Definition* tab to continue.

Use the *Definition* tab to set properties for the database: Select a character set from the drop-down listbox in the *Encoding* field. The

default is *UTF8*.

Create - Database

General Definition **Security** Parameters SQL

Privileges

Grantee	Privileges	Grantor
<input type="button" value=""/> enterpriseedb	<input checked="" type="checkbox"/> ALL <input type="checkbox"/> WITH GRANT OPTION <input checked="" type="checkbox"/> CREATE <input type="checkbox"/> WITH GRANT OPTION <input checked="" type="checkbox"/> TEMPORARY <input type="checkbox"/> WITH GRANT OPTION <input checked="" type="checkbox"/> CONNECT <input type="checkbox"/> WITH GRANT OPTION	<input type="button" value=""/> enterpriseedb

Security Labels

Provider	Security Label
<input type="button" value=""/> myProvider	mySecurity

Buttons:

Select a template from the drop-down listbox in the *Template* field. If you do not specify a template, the database will use template1.

Select a tablespace from the drop-down listbox in the *Tablespace* field. The selected tablespace will be the default tablespace used to contain database objects.

Select the collation order from the drop-down listbox in the *Collation* field.

Select the character classification from the drop-down listbox in the *Character Type* field. This affects the categorization of characters, e.g. lower, upper and digit. The default, or a blank field, uses the character classification of the template database.

Specify a connection limit in the *Connection Limit* field to configure the maximum number of connection requests. The default value (-1) allows unlimited connections to the database.

Click the *Security* tab to continue.

Use the *Security* tab to assign privileges and define security labels.

Use the *Privileges* panel to assign privileges to a role. Click the *Add* icon (+) to set privileges for database objects:

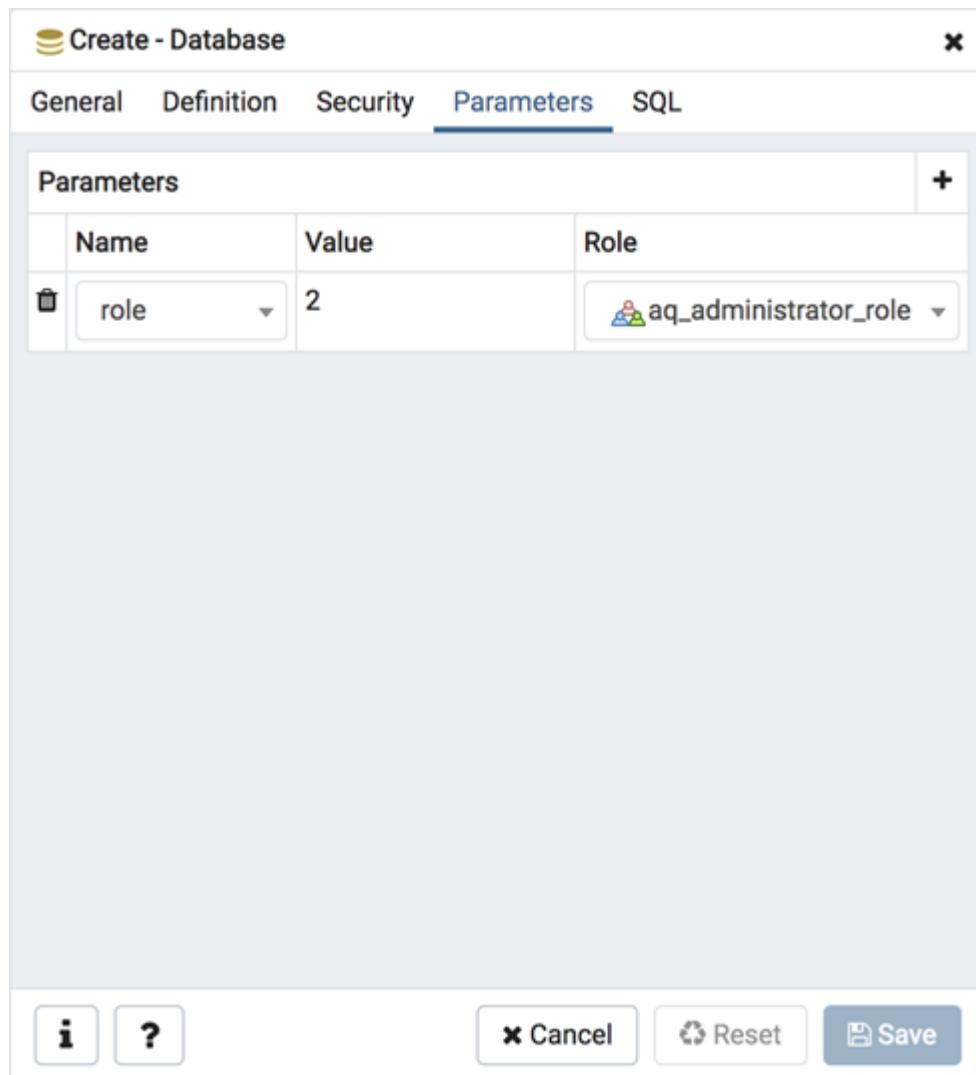
Select the name of the role from the drop-down listbox in the *Grantee* field.

Click inside the *Privileges* field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.

Click add to set additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Use the *Security Labels* panel to define security labels applied to the database. Click the *Add* icon (+) to add each security label selection:



Specify a security label provider in the *Provider* field. The named provider must be loaded and must consent to the proposed labeling operation.

Specify a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL

places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

To discard a security label, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *Parameters* tab to continue.

Use the *Parameters* tab to set parameters for the database. Click the *Add* icon (+) to add each parameter:

Use the drop-down listbox in the *Name* field to select a parameter.

Use the *Value* field to set a value for the parameter.

Use the drop-down listbox next to *Role* to select a role to which the parameter setting specified will apply.

Follow these steps to add additional parameter value definitions; to discard a parameter, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

Your entries in the *Database* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example 

```
1 CREATE DATABASE hr
2   WITH
3     OWNER = enterprisedb
4     TEMPLATE = template1
5     ENCODING = 'UTF8'
6     TABLESPACE = pg_default
7     CONNECTION LIMIT = -1;
8
9 COMMENT ON DATABASE hr
10   IS 'Database for human resources';
11
12 ALTER ROLE enterprisedb IN DATABASE hr
13   SET role TO 2;
14
15 GRANT ALL ON DATABASE hr TO enterprisedb WITH GRANT OPTION;
```

The following is an example of the sql command generated by user selections in the *Database* dialog:

The example creates a database named *hr* that is owned by *postgres*. It allows unlimited connections, and is available to all authenticated users.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Developer Tools](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Debugger](#)
 - [In-context Debugging](#)
 - [Direct Debugging](#)
 - [Using the Debugger](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
 - [Debugger](#)
 - [Query Tool](#)
 - [View/Edit Data](#)
 - [Schema Diff](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Debugger

The screenshot shows the pgAdmin 4 debugger interface. At the top, there's a toolbar with various icons. Below it is a title bar with the word "Debugger". The main area contains a code editor with the following PL/pgSQL code:

```
1
2+ FUNCTION
3+   RETURNS INT IS
4+   DECLARE IF aa IS NOT NULL
5+   THEN
6+     IF aa > 100 THEN
7+       RETURN TRUE;
8+     END IF;
9+     RETURN aa - 100 + 1;
10+ END IF;
11+
```

Below the code editor is a table titled "Parameters" with one row. The table has columns: Name, Type, and Value. The row contains:

Name	Type	Value
\$1	int4range	20007

The debugger may be used to debug PL/pgSQL functions in PostgreSQL, as well as EDB-SPL Functions, stored procedures and packages in EDB Postgres Advanced Server. The Debugger is available as an extension for your PostgreSQL installation, and is distributed as part of Advanced Server. You must have superuser privileges to use the debugger.

Before using the debugger, you must modify the `postgresql.conf` file, adding the server-side debugger components to the the value of the `shared_preload_libraries` parameter:

The screenshot shows the pgAdmin 4 interface with a debugger window. The code in the editor is:

```
1
2 BEGIN
3   IF $1 % 4 != 0 THEN
4     RETURN FALSE;
5   END IF;
6   IF $1 % 100 != 0 THEN
7     RETURN TRUE;
8   END IF;
9   RETURN $1 % 400 = 0;
10 END;
```

The line `IF $1 % 4 != 0 THEN` is highlighted with a blue background. Below the code is a table titled "Parameters" with one row:

Parameters	Localvariables	Messages	Results	Stack
Name	Type			Value
\$1	smallint			2007

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Developer Tools](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



Debugger

[Download](#)

[In-context Debugging](#)

[Direct Debugging](#)

[Using the Debugger](#)



[Getting Started](#)

[FAQ](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)



[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[Latest Docs](#)

[Management Basics](#)

[Backup and Restore](#)



[Developer Tools](#)

Debugger

[Ask a](#)

[Query Tool](#)

[Question](#)

[View/Edit Data](#)

[Schema Diff](#)



[pgAgent](#)

[Report a Bug](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)



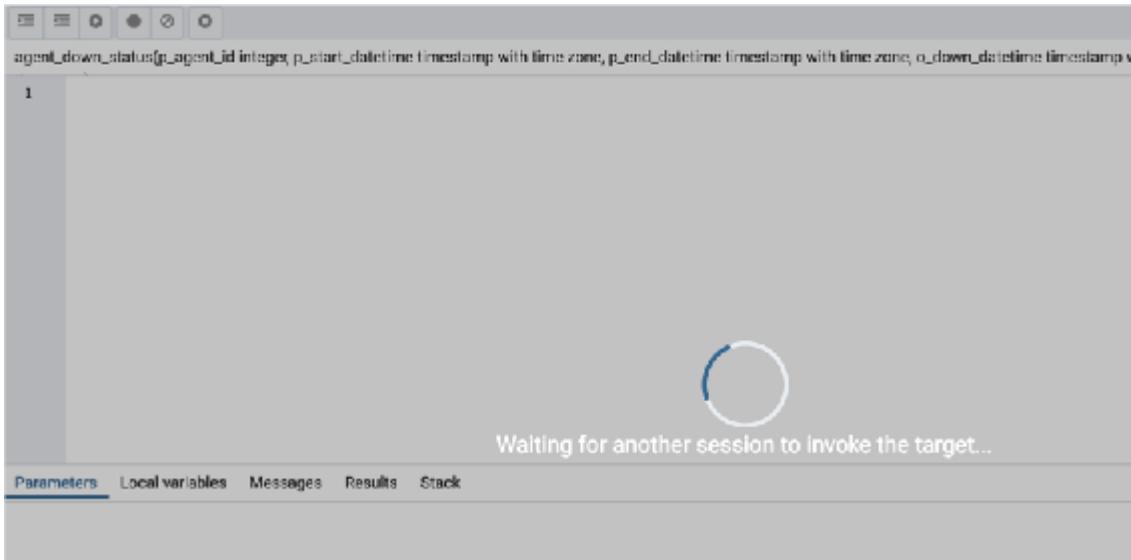
Debugger

[Screenshots](#)

The debugger may be used to debug PL/pgSQL functions in PostgreSQL, as well as EDB-SPL functions, stored procedures and packages in EDB

Postgres Advanced Server. The Debugger is available as an extension for your PostgreSQL installation, and is distributed as part of Advanced Server. You must have superuser privileges to use the debugger.

Before using the debugger, you must modify the *postgresql.conf* file, adding the server-side debugger components to the value of the *shared_preload_libraries* parameter:



The screenshot shows the Oracle SQL Developer interface with a PL/SQL code editor. The code is a function named s_leap_year(). It contains several IF statements and RETURN statements. Line 3, which contains an IF statement, has a blue rectangular highlight around it, indicating it is the current line of execution or a breakpoint. The code is as follows:

```
s_leap_year()
1
2 BEGIN
3   IF $1 % 4 != 0 THEN
4     RETURN FALSE;
5   END IF;
6   IF $1 % 100 != 0 THEN
7     RETURN TRUE;
8   END IF;
9   RETURN $1 % 400 = 0;
10 END;
```

Below the code editor, there are tabs for Parameters, Local variables, Messages, Results, and Stack. The Parameters tab is selected. A table shows the parameter values:

Name	Type	Value
\$1	smallint	2007

shared_preload_libraries = '\$libdir/ *other_libraries*/plugin_debugger'

After modifying the *shared_preload_libraries* parameter, restart the server to apply the changes.

The debugger may be used for either in-context debugging or direct debugging of a target function or procedure. When you use the debugger for in-context debugging, you set a breakpoint at the first line of a program; when a session invokes the target, control is transferred to the debugger. When using direct debugging, the debugger prompts you for any parameters required by the target, and then allows you to step through the code.

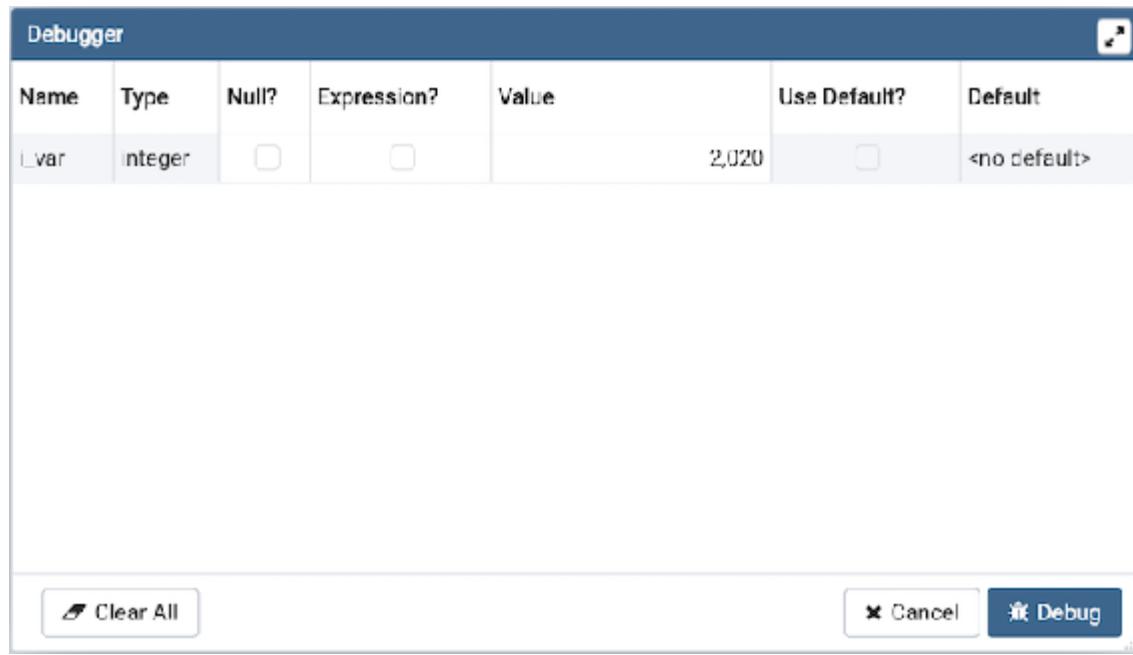
In-context Debugging

To set a breakpoint at the first line of a program, right-click the name of the object you would like to debug, and select *Set breakpoint* from the *Debugging* sub-menu. The debugger window will open, waiting for another session to invoke the program.

When another session invokes the target, the debugger will display the code, allowing you to add break points, or step through line-by-line. The other session is suspended until the debugging completes; then control is returned to the session.

Direct Debugging

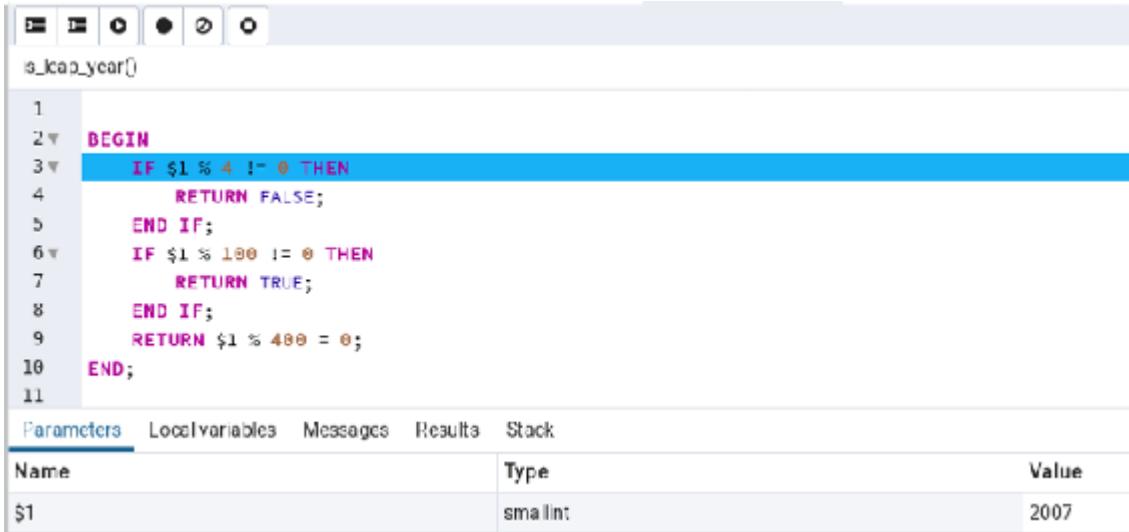
To use the debugger for direct debugging, right click on the name of the object that you wish to debug in the pgAdmin tree control and select *Debug* from the *Debugging* sub-menu. The debugger window will open, prompting you for any values required by the program:



The screenshot shows the pgAdmin Debugger window. The title bar says "Debugger". The main area is a table with the following data:

Name	Type	Null?	Expression?	Value	Use Default?	Default
_var	integer	<input type="checkbox"/>	<input type="checkbox"/>	2,020	<input type="checkbox"/>	<no default>

At the bottom are three buttons: "Clear All", "Cancel", and "Debug".



```

s_isap_year()

1
2 BEGIN
3   IF $1 % 4 != 0 THEN
4     RETURN FALSE;
5   END IF;
6   IF $1 % 100 != 0 THEN
7     RETURN TRUE;
8   END IF;
9   RETURN $1 % 400 = 0;
10 END;
11

```

Parameters	Localvariables	Messages	Results	Stack
Name	Type			Value
\$1	smallint			2007

Use the fields on the *Debugger* dialog to provide a value for each parameter:

The *Name* field contains the formal parameter name.

The *Type* field contains the parameter data type.

Check the *Nul?* checkbox to indicate that the parameter is a NULL value.

Check the *Expression?* checkbox if the Value field contains an expression.

Use the *Value* field to provide the parameter value that will be passed to the program. When entering parameter values, type the value into the appropriate cell on the grid, or, leave the cell empty to represent NULL, enter “” (two single quotes) to represent an empty string, or to enter a literal string consisting of just two single quotes, enter “”.

PostgreSQL 8.4 and above supports variadic function parameters.

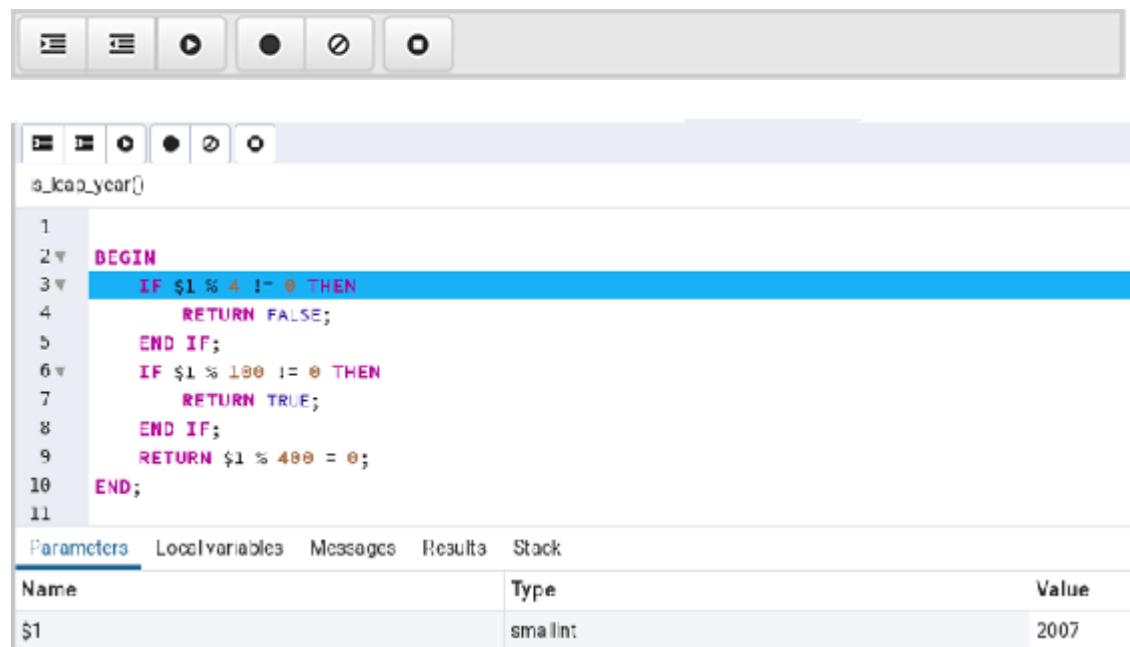
These may be entered as a comma-delimited list of values, quoted and/or cast as required.

Check the *Use default?* checkbox to indicate that the program should use the value in the Default Value field.

The *Default Value* field contains the default value of the parameter.

Provide values required by the program, and click the *Debug* button to start stepping through the program. The values of the arguments provided here are saved. The values will be pre-filled next time the dialog opens. To clear the values, use the *Clear All* button.

Using the Debugger



The screenshot shows the debugger window with the following components:

- Toolbar:** At the top, there is a toolbar with several icons: a magnifying glass, a list icon, a play button, a stop button, a break point icon, and another play button.
- Code Editor:** Below the toolbar, the code for `s_leap_year()` is displayed. The line `IF $1 % 4 != 0 THEN` is highlighted with a blue background.
- Parameter Table:** A table at the bottom shows the current values of variables:

Parameters	LocalVariables	Messages	Results	Stack
Name	Type			Value
\$1	smallint			2007

The main debugger window consists of two panels and a context-sensitive toolbar. Use toolbar icons to manage breakpoints and step into or through code; hover over an icon for a tooltip that identifies the option associated with the icon. The toolbar options are:

Option

Action

Step into

Click the *Step into* icon to execute the currently highlighted line of code.

Step over

Click the *Step over* icon to execute a line of code, stepping over any sub-functions invoked by the code.

The sub-function executes, but is not debugged unless it contains a breakpoint.

Continue/Start Click the *Continue/Start* icon to execute the highlighted code, and continue until the program encounters a breakpoint or completes.

Toggle

Use the *Toggle breakpoint* icon to enable or disable a *breakpoint* breakpoint (without removing the breakpoint).

Clear all

Click the *Clear all breakpoints* icon to remove all *breakpoints* breakpoints from the program.

Stop

Click the *Stop* icon to halt the execution of a program.

The top panel of the debugger window displays the program body; click in the grey margin next to a line number to add a breakpoint. The highlighted line in the top panel is the line that is about to execute.

The lower panel of the debugger window provides a set of tabs that allow you to review information about the program:

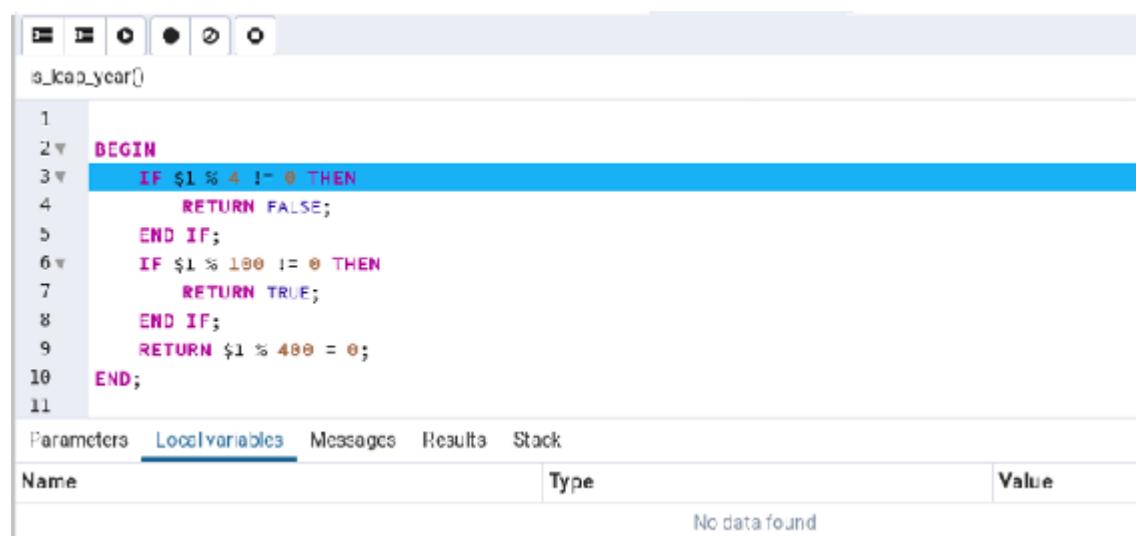
The *Parameters* tab displays the value of each parameter.

The *Local variables* tab displays the current value of the program variables.

The *Messages* tab displays any messages returned by the server (errors, warnings and informational messages).

The *Results* tab displays the server message when the program completes.

The *Stack* tab displays the list of functions that have been invoked, but which have not yet completed.



The screenshot shows the Oracle SQL Developer debugger interface. The code editor at the top contains the following PL/SQL code:1
2^y BEGIN
3^y IF \$1 % 4 != 0 THEN
4^y RETURN FALSE;
5^y END IF;
6^y IF \$1 % 100 != 0 THEN
7^y RETURN TRUE;
8^y END IF;
9^y RETURN \$1 % 400 = 0;
10^y END;
11Line 3 is highlighted with a blue background. Below the code editor is a navigation bar with tabs: Parameters, Localvariables, Messages, Results, and Stack. The Localvariables tab is selected. A table below the tabs shows the current value of the variable \$1:

Name	Type	Value
No data found		



The screenshot shows the Oracle SQL Developer debugger interface. The code editor at the top contains the same PL/SQL code as the previous screenshot. Below the code editor is a navigation bar with tabs: Parameters, Localvariables, Messages, Results, and Stack. The Stack tab is selected. A table below the tabs shows the current value of the variable \$1:

Name	Value	LineNo.
permis_leap_year(smallint)	\$1=2007	3

```
is_leap_year()
1v BEGIN
2 select p;
3v     IF $1 % 4 != 0 THEN
4         RETURN FALSE;
5     END IF;
6v     IF $1 % 100 != 0 THEN
7         RETURN TRUE;
8     END IF;
9     RETURN $1 % 400 = 0;
10 END;
11
```

Parameters Localvariables **Messages** Results Stack

ERROR: column "p" does not exist
LINE 1: select p
 ^
QUERY: select p
CONTEXT: PL/pgSQL function pm.is_leap_year(smallint) line 2 at SQL statement

As you step through a program, the *Local variables* tab displays the current value of each variable:

When you step into a subroutine, the *Stack* tab displays the call stack, including the name of each caller, the parameter values for each caller (if any), and the line number within each caller:

Select a caller to change focus to that stack frame and display the state of the caller in the upper panel.

When the program completes, the *Results* tab displays the message returned by the server. If the program encounters an error, the *Messages* tab displays details:

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) > [Getting Started](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
 - [Deployment](#)
 - [Login Dialog](#)
 - [User Management Dialog](#)
 - [Change User Password Dialog](#)
 - [User Interface](#)
 - [Menu Bar](#)
 - [Toolbar](#)
 - [Tabbed Browser](#)
 - [Tree Control](#)
 - [Preferences Dialog](#)
 - [Keyboard Shortcuts](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Deployment

Pre-compiled and configured installation packages for pgAdmin 4 are available for a number of desktop environments; we recommend using an installer whenever possible. If you are interested in learning more about the project, or if a pgAdmin installer is not available for your environment, the pages listed below will provide detailed information about creating a custom deployment.

- [The config.py File](#)
- [Desktop Deployment](#)
 - [Configuration](#)
 - [Runtime](#)
- [Server Deployment](#)
 - [Requirements](#)
 - [Configuration](#)
 - [Hosting](#)
- [Container Deployment](#)
 - [PostgreSQL Utilities](#)

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#)
[Development](#)

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

Deployment

[Login Dialog](#)

[User Management Dialog](#)



[Change User Password Dialog](#)

[FAQ](#)

[User Interface](#)

[Menu Bar](#)



[Toolbar](#)

[Tabbed Browser](#)

[Latest Docs](#)

[Tree Control](#)

[Preferences Dialog](#)



[Keyboard Shortcuts](#)

[Connecting To A Server](#)

[Ask a](#)

[Managing Cluster Objects](#)

[Question](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Management Basics](#)

[Backup and Restore](#)

[Report a Bug](#)

[Developer Tools](#)

[pgAgent](#)



[pgAdmin Project Contributions](#)

[Release Notes](#)

[Screenshots](#)

[Licence](#)

Deployment

Pre-compiled and configured installation packages for pgAdmin 4 are

available for a number of desktop environments; we recommend using an

installer whenever possible. If you are interested in learning more about

the project, or if a pgAdmin installer is not available for your environment, the pages listed below will provide detailed information about creating a

custom deployment.

[The config.py File](#)

[Desktop Deployment](#)

[Configuration](#)

[Runtime](#)

[Server Deployment](#)

[Requirements](#)

[Configuration](#)

[Hosting](#)

[Container Deployment](#)

[PostgreSQL Utilities](#)

[Environment Variables](#)

[Mapped Files and Directories](#)

[Examples](#)

[Reverse Proxying](#)

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 4.19 documentation](#) > [Getting Started](#) > [Deployment](#) > previous | next | index

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Desktop Deployment](#)
 - [Configuration](#)
 - [Runtime](#)
- [Getting Started](#)
 - [Deployment](#)
 - [Login Dialog](#)
 - [User Management Dialog](#)
 - [Change User Password Dialog](#)
 - [User Interface](#)
 - [Menu Bar](#)
 - [Toolbar](#)
 - [Tabbed Browser](#)
 - [Tree Control](#)
 - [Preferences Dialog](#)
 - [Keyboard Shortcuts](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Desktop Deployment

pgAdmin may be deployed as a desktop application by configuring the application to run in desktop mode and then utilising the desktop runtime to host the program on a supported Windows, Mac OS X or Linux installation.

The desktop runtime is a system-tray application that when launched, runs the pgAdmin server and launches a web browser to render the user interface. If additional instances of pgAdmin are launched, a new browser tab will be opened and be served by the existing instance of the server in order to minimise system resource utilisation. Clicking the icon in the system tray will present a menu offering options to open a new pgAdmin window, configure the runtime, view the server log and shut down the server.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) » [Deployment](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



Desktop Deployment

[Download](#)

[Configuration](#)

[Runtime](#)



[Getting Started](#)

[Deployment](#)

[FAQ](#)

[Login Dialog](#)

[User Management Dialog](#)



[Change User Password Dialog](#)

[User Interface](#)

[Latest Docs](#)

[Menu Bar](#)

[Toolbar](#)



[Tabbed Browser](#)

[Tree Control](#)

[Ask a](#)

[Preferences Dialog](#)

[Question](#)

[Keyboard Shortcuts](#)

[Connecting To A Server](#)



[Managing Cluster Objects](#)

[Report a Bug](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[Management Basics](#)



[Backup and Restore](#)

[Screenshots](#)

[Developer Tools](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Desktop Deployment

pgAdmin may be deployed as a desktop application by configuring the application to run in desktop mode and then utilising the desktop runtime to host the program on a supported Windows, Mac OS X or Linux instal ation.

The desktop runtime is a system-tray application that when launched, runs the pgAdmin server and launches a web browser to render the user interface. If additional instances of pgAdmin are launched, a new browser tab wil be opened and be served by the existing instance of the server in order to minimise system resource utilisation. Clicking the icon in the system tray wil present a menu offering options to open a new pgAdmin window, configure the runtime, view the server log and shut down the server.

Note

Pre-compiled and configured instal ation packages are available for a number of platforms. These packages should be used by end-users wherever possible - the fol owing information is useful for the maintainers of those packages and users interested in understanding how pgAdmin works.

See also

For detailed instructions on building and configuring pgAdmin from scratch, please see the README file in the top level directory of the source code. For convenience, you can find the latest version of the file

[here](#), but be aware that this may differ from the version included with the source code for a specific version of pgAdmin.

Configuration

From pgAdmin 4 v2 onwards, the default configuration mode is server, however, this is overridden by the desktop runtime at startup. In most environments, no Python configuration is required unless you wish to override other default settings.

See [The config.py File](#) for more information on configuration settings.

Runtime

When executed, the runtime will automatically try to execute the pgAdmin Python application. If execution fails, it will prompt you to adjust the Python Path to include the directories containing the pgAdmin code as well as any additional Python dependencies. You can enter a list of paths by separating them with a semi-colon character, for example:

```
/Users/dpage/.virtualenvs/pgadmin4/lib/python2.7/site-packages;/Users/dpage/python-libs/
```

The configuration settings are stored using the QSettings class in Qt, which will use an INI file on Unix systems

(`~/.config/pgadmin/pgadmin4.conf`), a plist file on Mac OS X

(~/Library/Preferences/org.pgadmin.pgadmin4.plist), and the registry on Windows (HKEY_CURRENT_USER\Software\pgadmin\pgadmin4).

The configuration settings:

Key

Type

Purpose

ApplicationPath

String

The directory containing pgAdmin4.py

BrowserCommand String

An alternate command to run instead of
the default browser.

ConnectionTimeout Integer

The number of seconds to wait for
application server startup.

PythonPath

String

The Python module search path

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Domain Constraints Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Domain Constraints Dialog 

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Domain Constraints Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

Domain Constraints Dialog

[Event Trigger Dialog](#)



[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)



[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

Domain Constraints Dialog



Use the *Domain Constraints* dialog to create or modify a domain constraint. A domain constraint confirms that the values provided for a domain meet a defined criteria. The *Domain Constraints* dialog implements options of the ALTER DOMAIN command.

The *Domain Constraints* dialog organizes the development of a domain constraint through the following dialog tabs: *General* and *Definition*. The *SQL* tab displays the SQL code generated by dialog selections.

Use the fields in the *General* tab to identify the domain constraint:
Use the *Name* field to add a descriptive name for the constraint. The name will be displayed in the pgAdmin tree control.

Store notes about the constraint in the *Comment* field.

Click the *Definition* tab to continue.

The screenshot shows the 'Create - Domain Constraints' dialog box. The 'Definition' tab is selected. In the 'Check' field, the value 'Value='Friday'' is entered. Below this, a 'Validate?' dropdown menu is open, with 'Yes' selected. At the bottom of the dialog, there are buttons for 'Cancel', 'Reset', and 'Save'.

Use the fields in the *Definition* tab to define the domain constraint:
Use the *Check* field to provide a CHECK expression. A CHECK

expression specifies a constraint that the domain must satisfy. A constraint must produce a Boolean result; include the key word

VALUE to refer to the value being tested. Only those expressions that evaluate to TRUE or UNKNOWN will succeed. A CHECK expression

cannot contain subqueries or refer to variables other than VALUE. If a domain has multiple CHECK constraints, they will be tested in alphabetical order.

Move the *Validate?* switch to the *No* position to mark the constraint NOT VALID. If the constraint is marked NOT VALID, the constraint will not be applied to existing column data. The default value is Yes.

Click the *SQL* tab to continue.

Your entries in the *Domain Constraints* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Domain Constraints* dialog:

A screenshot of a software interface titled "Create - Domain Constraints". The "SQL" tab is selected. The code area contains the following SQL command:
1 ALTER DOMAIN public.testdom
2 ADD CONSTRAINT timesheets CHECK (Value='Friday');

The example shown demonstrates creating a domain constraint on the domain *timesheets* named *weekday*. It constrains a value to equal *Friday*.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Domain Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Domain Dialog 

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Domain Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

Domain Dialog

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)



[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)



[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

Domain Dialog



The screenshot shows the 'Create - Domain' dialog box. The 'General' tab is selected. The 'Name' field is empty. The 'Owner' field is set to 'enterprisedb'. The 'Schema' field is set to 'public'. The 'Comment' field is empty. At the bottom, there are buttons for 'i' (info), '?', 'Cancel', 'Reset', and 'Save'.

Use the *Domain* dialog to define a domain. A domain is a data type definition that may constrain permissible values. Domains are useful when you are creating multiple tables that contain comparable columns; you can create a domain that defines constraints that are common to the columns and re-use the domain definition when creating the columns, rather than individually defining each set of constraints.

The *Domain* dialog organizes the development of a domain through the following tabs: *General*, *Definition*, *Constraints*, and *Security*.
The *SQL*

tab displays the SQL code generated by dialog selections.

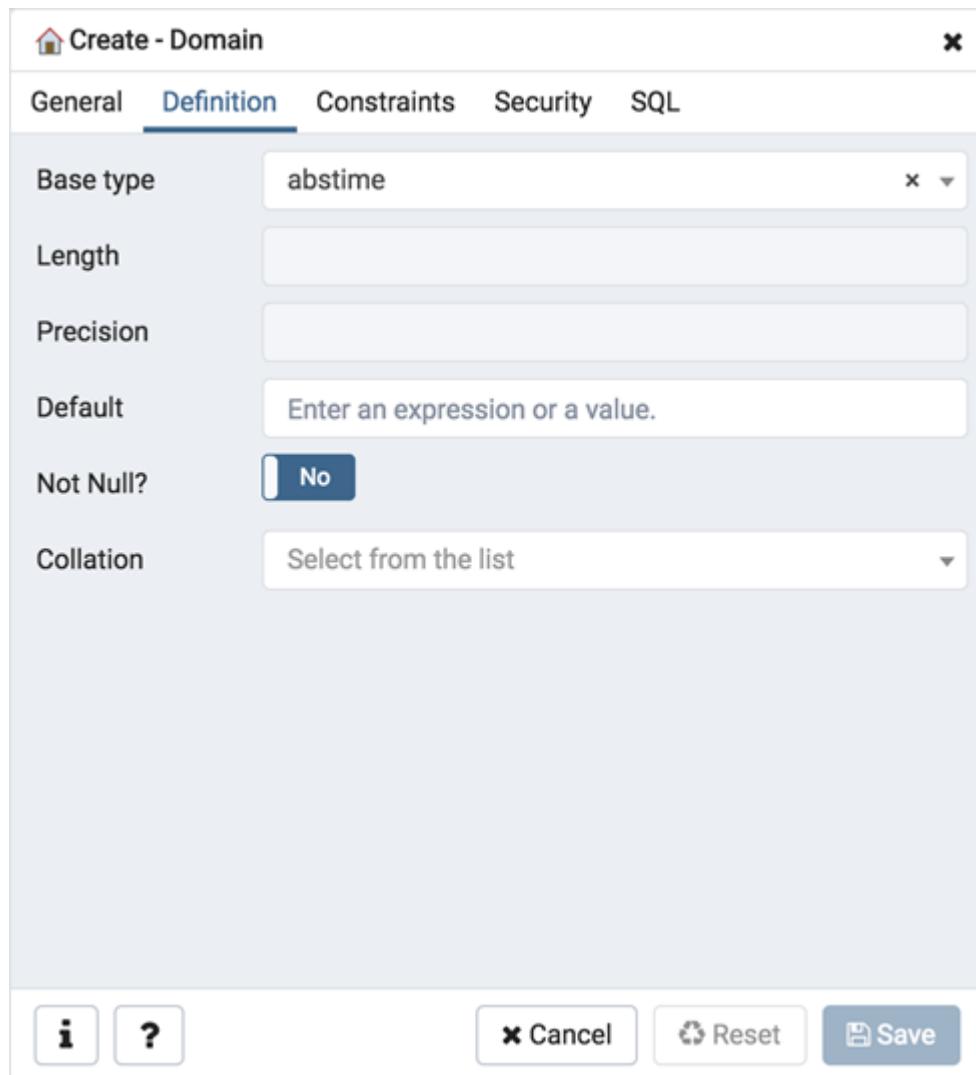
Use the fields on the *General* tab to identify a domain: Use the *Name* field to add a descriptive name for the domain. The name will be displayed in the *pgAdmin* tree control.

Use the drop-down listbox next to *Owner* to select a role that will own the domain.

Select the name of the schema in which the domain will reside from the drop-down listbox in the *Schema* field.

Store notes about the domain in the *Comment* field.

Click the *Definition* tab to continue.

A screenshot of a software dialog box titled "Create - Domain". The "Definition" tab is selected. The "Base type" field contains "abstime". The "Length" and "Precision" fields are empty. The "Default" field contains the placeholder "Enter an expression or a value.". The "Not Null?" field has a dropdown menu set to "No". The "Collation" field has a dropdown menu set to "Select from the list". At the bottom are buttons for "Cancel", "Reset", and "Save".

Use the fields in the *Definition* tab to describe the domain: Use the drop-down listbox next to *Base type* to specify a data type.

Use the context-sensitive *Length* field to specify a numeric length for a numeric type.

Use the context-sensitive *Precision* field to specify the total count of significant digits for a numeric type.

Specify a default value for the domain data type in the *Default* field.

The data type of the default expression must match the data type of the domain. If no default value is specified, then the default value is

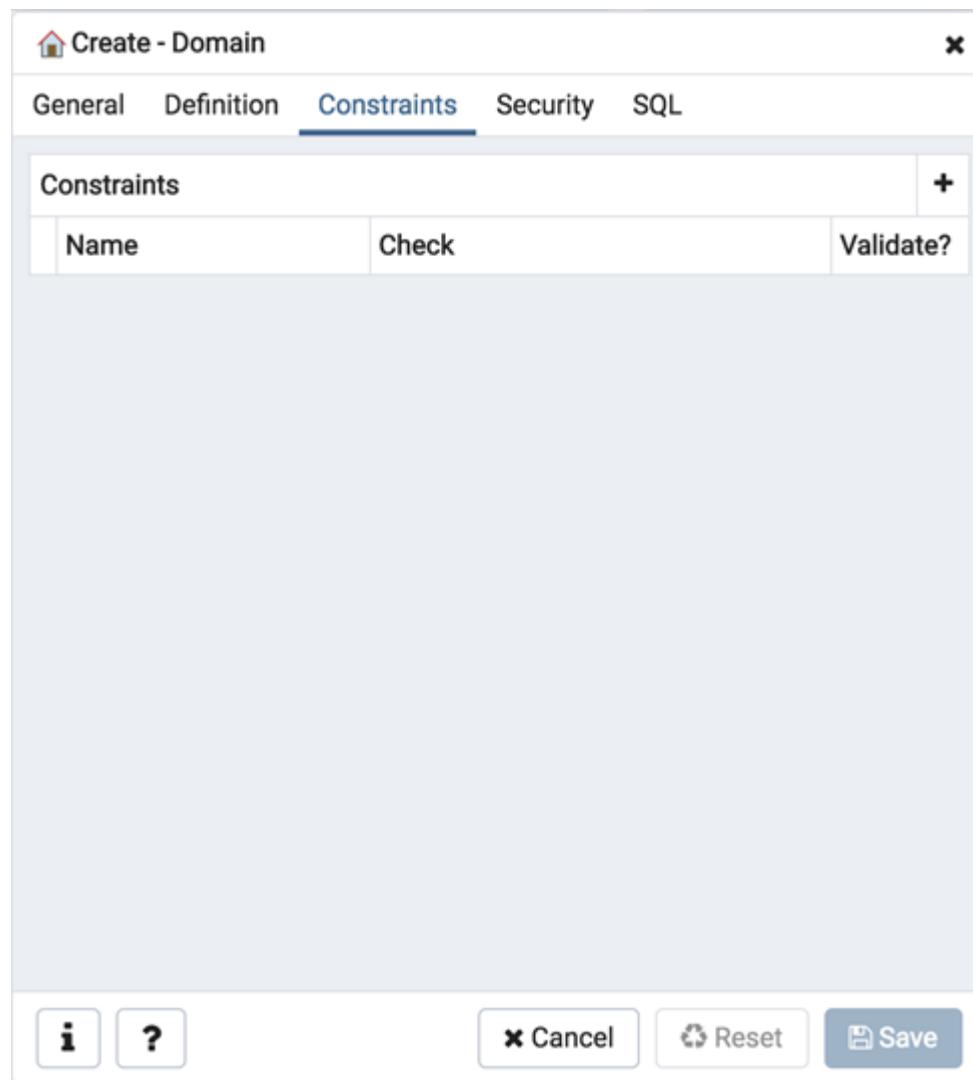
the nul value.

Move the *Not Nul* switch to specify the values of this domain are prevented from being nul .

Use the drop-down listbox next to *Col ation* to apply a col ation cast.

If no col ation is specified, the underlying data type's default col ation is used. The underlying type must be col atable if COLLATE is specified.

Click the *Constraints* tab to continue.



Use the fields in the *Constraints* tab to specify rules for the domain. Click the *Add* icon (+) to set constraints:

Use the *Name* field to specify a name for the constraint.

Use the *Check* field to provide an expression for the constraint.

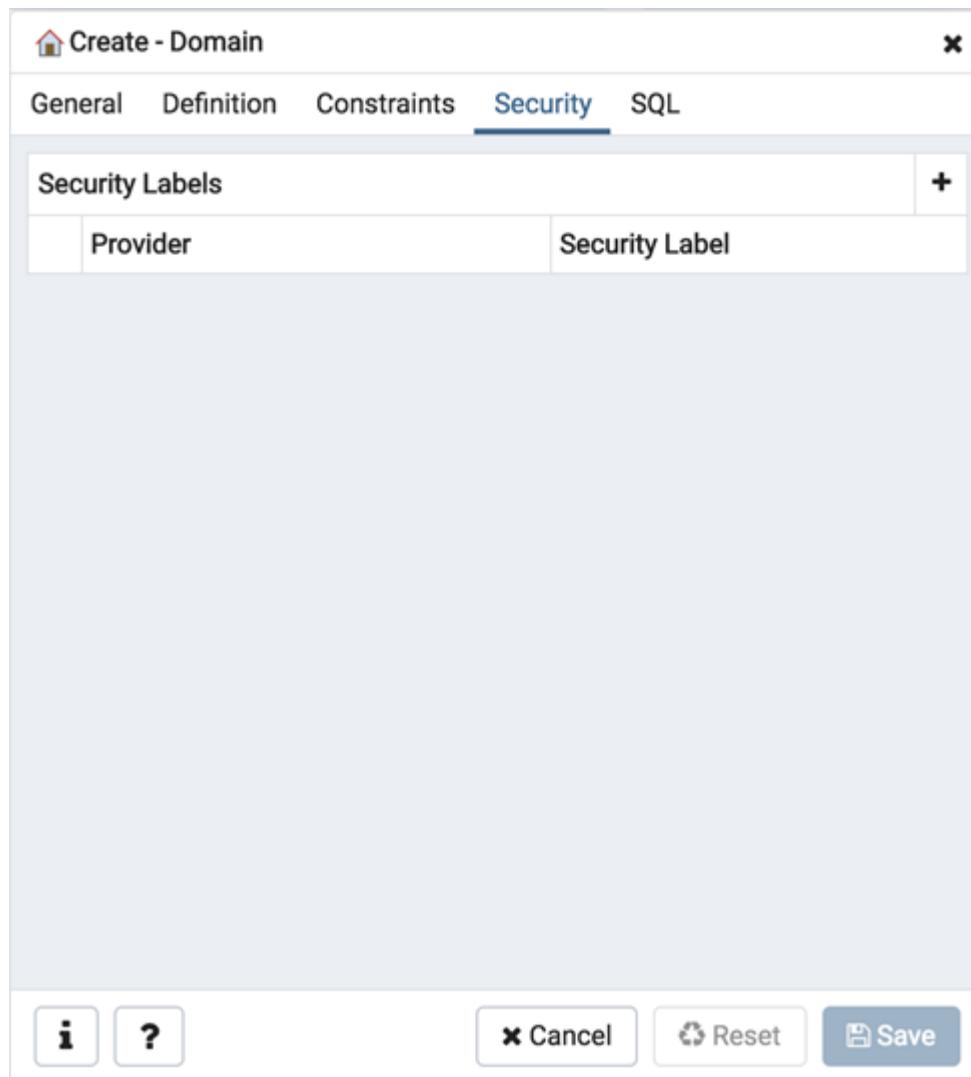
Use the *Validate* checkbox to determine whether the constraint will be validated. The default checkbox is checked and sets a validation requirement.

A CHECK clause specifies an integrity test which values of the domain must satisfy. Each constraint must be an expression that produces a Boolean result. Use the key word VALUE to refer to the value being tested.

Expressions evaluating to TRUE or UNKNOWN succeed. If the expression produces a FALSE result, an error is reported and the value is not allowed to be converted to the domain type. A CHECK expression cannot contain subqueries nor refer to variables other than VALUE. If a domain has multiple CHECK constraints, they will be tested in alphabetical order by name.

Click the *Add* icon (+) to set additional constraints; to discard a constraint, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *Security* tab to continue.



Use the *Security Labels* panel to assign security labels. Click the *Add* icon (+) to add a label:

Specify a security label provider in the *Provider* field. The named provider must be loaded and must consent to the proposed labeling operation.

Specify a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

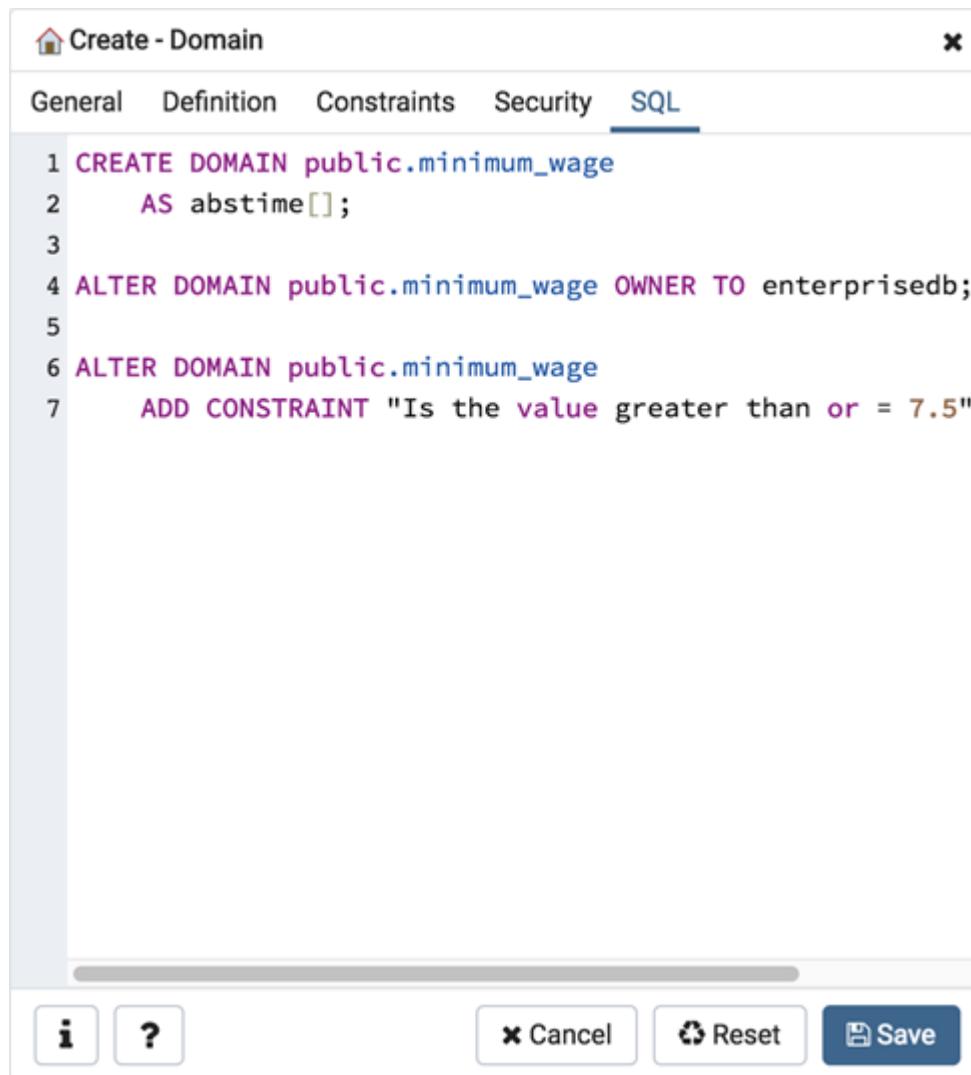
Click the *Add* icon (+) to specify each additional label; to discard a label, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

Your entries in the *Domain* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example 

The following is an example of the sql command generated by selections made in the *Domain* dialog:

A screenshot of a PostgreSQL 'Create - Domain' dialog. The window title is 'Create - Domain'. The tabs at the top are 'General', 'Definition', 'Constraints', 'Security', and 'SQL', with 'SQL' being the active tab. The SQL code area contains the following:

```
1 CREATE DOMAIN public.minimum_wage
2     AS abstime[];
3
4 ALTER DOMAIN public.minimum_wage OWNER TO enterprisedb;
5
6 ALTER DOMAIN public.minimum_wage
7     ADD CONSTRAINT "Is the value greater than or = 7.5"
```

The bottom of the dialog has several buttons: 'Info' (i), 'Help' (?), 'Cancel' (with an 'x'), 'Reset' (with a circular arrow), and 'Save' (highlighted in blue).

The example shown demonstrates creating a domain named *minimum-wage* that confirms that the value entered is greater than or equal to 7.25.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Event Trigger Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Event Trigger Dialog 

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Event Trigger Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

Event Trigger Dialog



[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)



[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

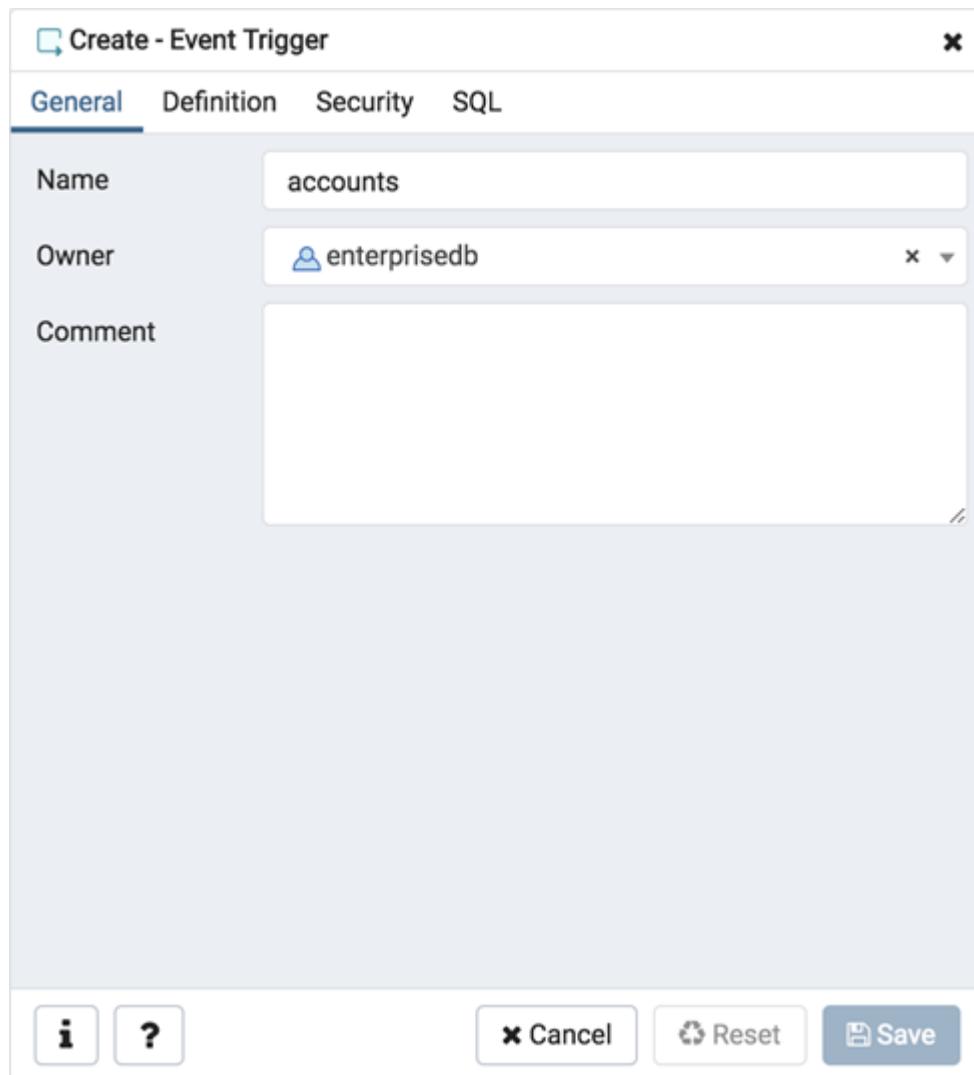
[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

Event Trigger Dialog



Use the *Domain Trigger* dialog to define an event trigger. Unlike regular triggers, which are attached to a single table and capture only DML

events, event triggers are global to a particular database and are capable of capturing DDL events. Like regular triggers, event triggers can be written in any procedural language that includes event trigger support, or in C, but not in SQL.

The *Domain Trigger* dialog organizes the development of a event trigger through the fol owing dialog tabs: *General*, *Definition*, and *Security Labels*. The *SQL* tab displays the SQL code generated by dialog selections.

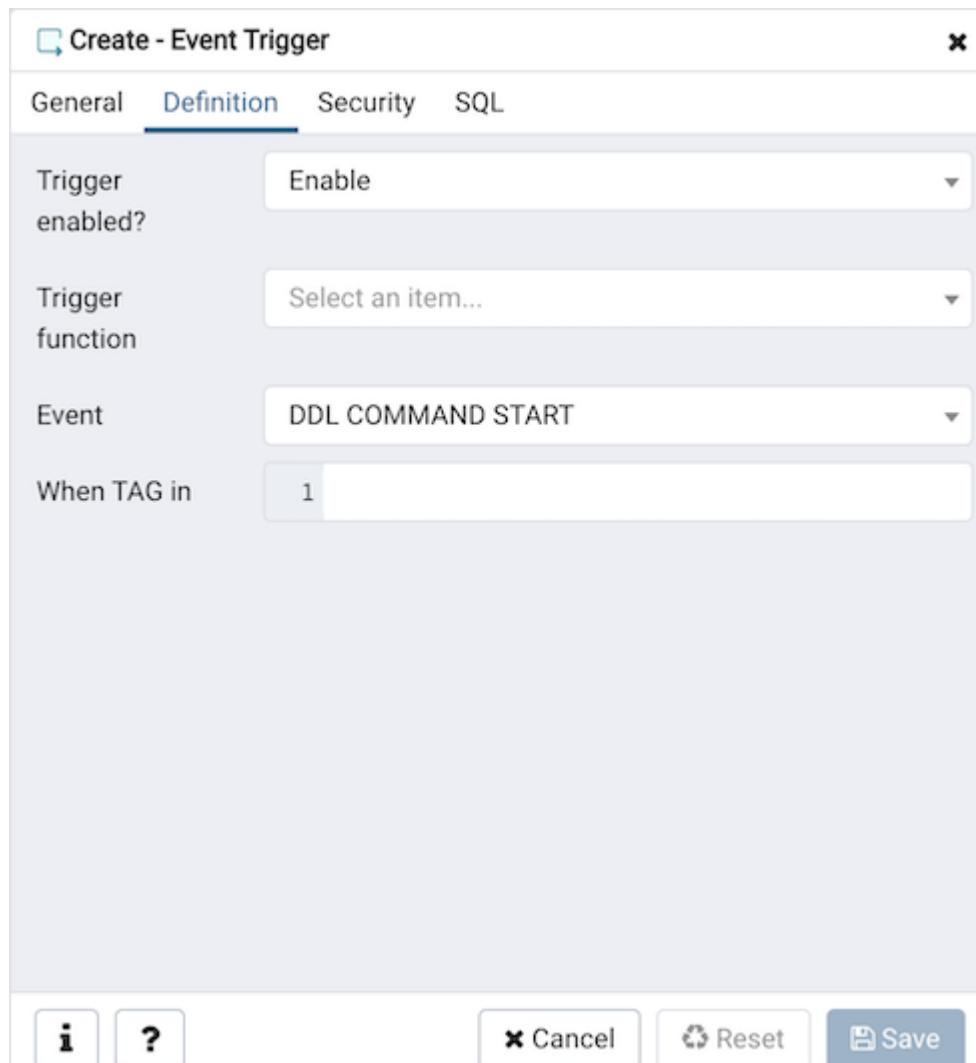
Use the fields in the *General* tab to identify the event trigger: Use the *Name* field to add a descriptive name for the event trigger.

The name wil be displayed in the *pgAdmin* tree control.

Use the drop-down listbox next to *Owner* to specify the owner of the event trigger.

Store notes about the event trigger in the *Comment* field.

Click the *Definition* tab to continue.

A screenshot of a software interface titled "Create - Event Trigger". The window has tabs at the top: "General", "Definition" (which is selected and highlighted in blue), "Security", and "SQL".
The "Definition" tab contains the following fields:

- "Trigger enabled?" dropdown: "Enable"
- "Trigger function" dropdown: "Select an item..."
- "Event" dropdown: "DDL COMMAND START"
- "When TAG in" dropdown: "1"

At the bottom of the window are several buttons: an information icon (i), a question mark icon (?), a "Cancel" button with a cancel symbol, a "Reset" button with a recycle symbol, and a "Save" button with a disk symbol.

Use the fields in the *Definition* tab to define the event trigger: Select a value from the drop down of *Trigger Enabled* field to specify a status for the trigger: *Enable* *Disable*, *Replica Always*.

Use the drop-down listbox next to *Trigger function* to specify an existing function. A trigger function takes an empty argument list, and returns a value of type `event_trigger`.

Select a value from the drop down of *Events* field to specify when the event trigger wil fire: *DDL COMMAND START*, *DDL COMMAND END*, or *SQL DROP*.

Use the *When TAG in* field to enter filter values for TAG for which the trigger will be executed. The values must be in single quotes separated by comma.

Click the *Security Labels* tab to continue.

Create - Event Trigger

General Definition Security SQL

Security Labels		
	Provider	Security Label
	myProvider	mySecurity

Cancel Reset Save

Use the *Security* tab to define security labels applied to the trigger. Click the *Add* icon (+) to add each security label.

Specify a security label provider in the *Provider* field. The named provider must be loaded and must consent to the proposed labeling operation.

Specify a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL

places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the *Add* icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

Your entries in the *Domain Trigger* dialog generate a generate a SQL command. Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example 

The following is an example of the sql command generated by user selections in the *Domain Trigger* dialog:

Create - Event Trigger

General Definition Security SQL

```
1 CREATE EVENT TRIGGER accounts ON DDL_COMMAND_START
2     EXECUTE PROCEDURE public."Test_Event_Trigger"();
3
4 SECURITY LABEL FOR my_provider ON EVENT TRIGGER accounts IS 'my_security';
5
6 ALTER EVENT TRIGGER accounts
7     OWNER TO enterprisedb;
```

i **?** **Cancel** **Reset** **Save**

The command creates an event trigger named *accounts* that invokes the procedure named *acct_due*.

Click the *Info* button (*i*) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

Download

pgAdmin is a free software project released under the [PostgreSQL / Artistic Licence](#). The software is available in source and binary format from the [PostgreSQL mirror network](#). Because compiling from source requires technical knowledge, we recommend installing binary packages whenever possible.

The pages in this section give additional details about each binary package available as well as more direct download links. In addition, you can download source tarballs and pgAgent for your servers to enable additional functionality.

pgAdmin 4

pgAdmin 4 is a complete rewrite of pgAdmin, built using Python and Javascript/jQuery. A desktop runtime written in C++ with Qt allows it to run standalone for individual users, or the web application code may be deployed directly on a webserver for use by one or more users through their web browser. The software has the look and feel of a desktop application whatever the runtime environment is, and vastly improves on pgAdmin III with updated user interface elements, multi-user/web deployment options, dashboards and a more modern design.

[Container](#)[macOS](#)[Python
Wheel](#)[APT](#)[RPM](#)[Source
Code](#)[Windows](#)

pgAdmin 3

pgAdmin 3 is the most popular and feature rich Open Source administration and development platform for PostgreSQL, the most advanced Open Source database in the world. The application may be used on Linux, FreeBSD, Solaris, macOS and Windows platforms to manage PostgreSQL 8.4 through 9.5 running on any platform, as well as commercial and derived versions of PostgreSQL such as EDB Postgres Advanced Server.

WARNING: pgAdmin 3 is no longer supported. It is recommended that you download pgAdmin 4 instead.

[macOS](#)[Source Code](#)[Windows](#)

pgAgent

pgAgent is a job scheduler for PostgreSQL which may be managed using pgAdmin. Prior to pgAdmin v1.9, pgAgent shipped as part of pgAdmin. From pgAdmin v1.9 onwards, pgAgent is shipped as a separate application.

Quick Links

Download



[pgAdmin is a free software project released under the PostgreSQL Artistic licence. The software is available in source and binary format from the](#)

[Online Demo](#)

[PostgreSQL mirror network](#). Because compiling from source requires technical knowledge, we recommend installing binary packages whenever



possible.

[Download](#)

The pages in this section give additional details about each binary package available as well as more direct download links. In addition, you can download



source tarballs and pgAgent for your servers to enable additional functionality.

[FAQ](#)

pgAdmin 4



pgAdmin 4 is a complete rewrite of pgAdmin, built using Python and Javascript/jQuery. A desktop runtime written in C++ with Qt allows it to run

[Latest Docs](#)

standalone for individual users, or the web application code may be deployed directly on a webserver for use by one or more users through their web



browser. The software has the look and feel of a desktop application whatever the runtime environment is, and vastly improves on pgAdmin II with

[Ask a](#)

updated user interface elements, multi-user/web deployment options,

[Question](#)

dashboards and a more modern design.



[Report a Bug](#)

[Container](#)

[macOS](#)

[Python](#)

[APT](#)

[RPM](#)

[Source](#)

[Windows](#)



[Wheel](#)

[Code](#)

[Screenshots](#)

pgAdmin 3

pgAdmin 3 is the most popular and feature rich Open Source administration and development platform for PostgreSQL, the most advanced Open Source database in the world. The application may be used on Linux, FreeBSD, Solaris, macOS and Windows platforms to manage PostgreSQL 8.4 through 9.5 running on any platform, as well as commercial and derived versions of PostgreSQL such as EDB Postgres Advanced Server.

WARNING: pgAdmin 3 is no longer supported. It is recommended that you download pgAdmin 4 instead.



[macOS](#)

[Source Code](#)

[Windows](#)

pgAgent

pgAgent is a job scheduler for PostgreSQL which may be managed using pgAdmin. Prior to pgAdmin v1.9, pgAgent shipped as part of pgAdmin. From pgAdmin v1.9 onwards, pgAgent is shipped as a separate application.



[APT](#)

[macOS](#)

[RPM](#)

[Windows](#)

[Source](#)

[Code](#)

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Creation or Modifying a Table](#) > [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Exclusion Constraint Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
 - [Check Dialog](#)
 - [Column Dialog](#)
 - [Compound Trigger Dialog](#)
 - [Exclusion Constraint Dialog](#)
 - [Foreign key Dialog](#)
 - [Index Dialog](#)
 - [Primary key Dialog](#)
 - [Rule Dialog](#)
 - [Table Dialog](#)
 - [Trigger Dialog](#)
 - [Unique Constraint Dialog](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Exclusion Constraint Dialog

Use the *Exclusion constraint dialog* to define or modify the behavior of an exclusion constraint. An exclusion constraint guarantees that if any two rows are compared on the specified column or expression (using the specified operator), at least one of the operator comparisons will return false or null.

The *Exclusion constraint dialog* organizes the development of an exclusion constraint through the following dialog tabs: *General*, *Definition*, and *Columns*. The *SQL* tab displays the SQL code generated by dialog selections.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Creating or Modifying a Table](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



Exclusion Constraint Dialog

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[□](#)

[Check Dialog](#)

[Column Dialog](#)

[Latest Docs](#)

[Compound Trigger Dialog](#)

Exclusion Constraint Dialog

[□](#)

[Foreign key Dialog](#)

[Index Dialog](#)

[Ask a](#)

[Primary key Dialog](#)

[Question](#)

[Rule Dialog](#)

[Table Dialog](#)

[□](#)

[Trigger Dialog](#)

[Report a Bug](#)

[Unique Constraint Dialog](#)

[Management Basics](#)

[Backup and Restore](#)



[Developer Tools](#)

[Screenshots](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Exclusion Constraint Dialog

Use the *Exclusion constraint* dialog to define or modify the behavior of an exclusion constraint. An exclusion constraint guarantees that if any two rows are compared on the specified column or expression (using the specified operator), at least one of the operator comparisons will return false or null.

The *Exclusion constraint* dialog organizes the development of an exclusion constraint through the following dialog tabs: *General*, *Definition*, and *Columns*. The *SQL* tab displays the SQL code generated by dialog selections.

Create - Exclusion constraint

X

General Definition Columns SQL

Name

ex_c_orders

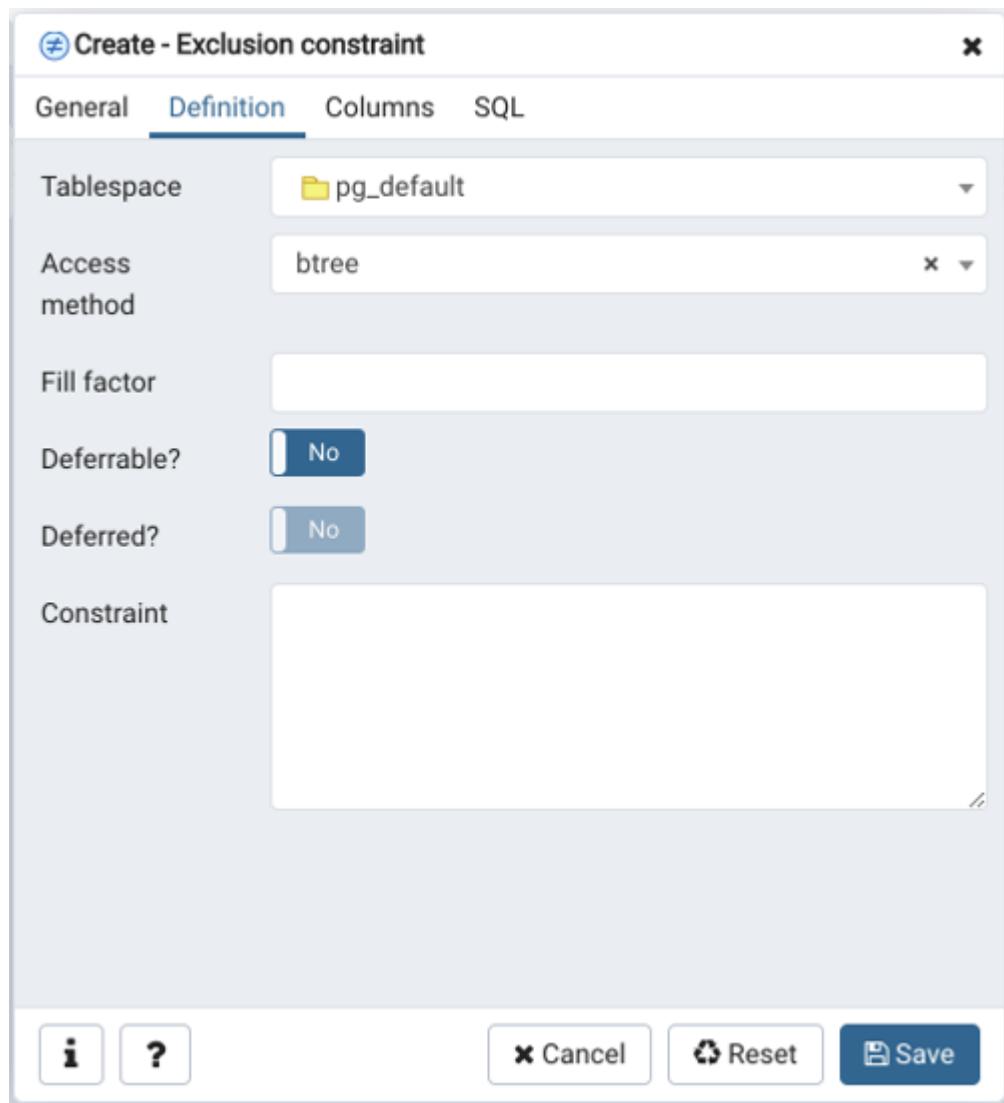
Comment



Cancel

Reset

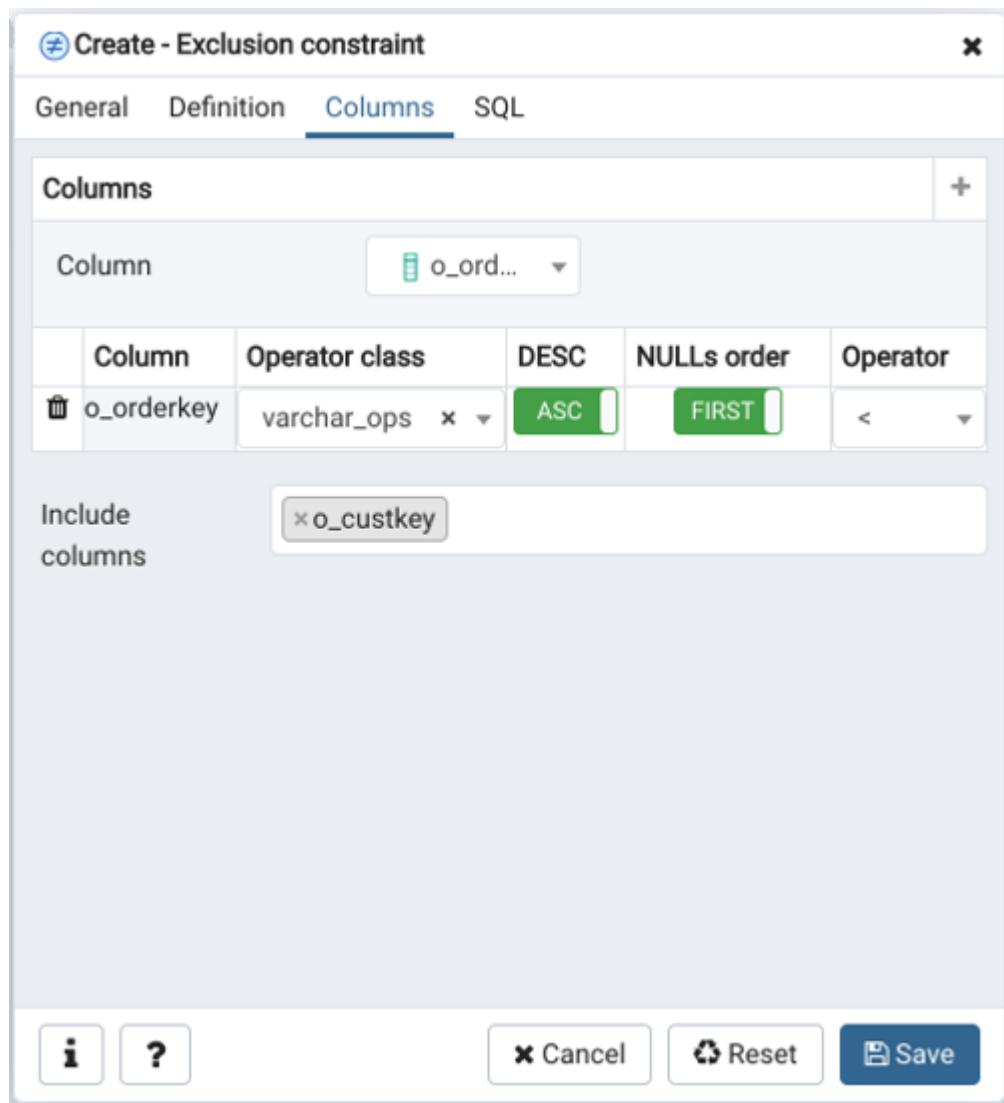
Save



Use the fields in the *General* tab to identify the exclusion constraint:
Use the *Name* field to provide a descriptive name for the exclusion constraint. The name will be displayed in the *pgAdmin* tree control.

Click the *Definition* tab to continue.

Use the fields in the *Definition* tab to define the exclusion constraint:
Use the drop-down listbox next to *Tablespace* to select the tablespace in which the index associated with the exclude constraint will reside.



Use the drop-down listbox next to *Access method* to specify the type of index that will be used when implementing the exclusion constraint:

Select *gist* to specify a GiST index.

Select *spgist* to specify a space-partitioned GiST index.

Select *btree* to specify a B-tree index.

Select *hash* to specify a hash index.

Use the *Fil Factor* field to specify a fil factor for the table and associated index. The fil factor is a percentage between 10 and 100.

100 (complete packing) is the default.

Move the *Deferrable?* switch to the *Yes* position to specify that the timing of the constraint is deferrable, and can be postponed until the end of the statement. The default is *No*.

If enabled, move the *Deferred?* switch to the *Yes* position to specify the timing of the constraint is deferred to the end of the statement.

The default is *No*.

Use the *Constraint* field to provide a condition that a row must satisfy to be included in the table.

Click the *Columns* tab to continue.

Use the fields in the *Columns* tab to to specify the column(s) to which the constraint applies. Use the drop-down listbox next to *Column* to select a column and click the *Add* icon (+) to provide details of the action on the column:

The *Column* field is populated with the selection made in the *Column* drop-down listbox.

Create - Exclusion constraint

General Definition Columns SQL

```
1 ALTER TABLE public.orders
2   ADD CONSTRAINT ex_c_orders EXCLUDE USING btree (
3     o_orderkey varchar_ops ASC NULLS FIRST WITH <)
4   INCLUDE(o_custkey);
```

Cancel Reset Save

If applicable, use the drop-down listbox in the *Operator class* to specify the operator class that will be used by the index for the column.

Move the *DESC* switch to *DESC* to specify a descending sort order.

The default is *ASC* which specifies an ascending sort order.

Use the *NULLs order* column to specify the placement of NULL values (when sorted). Specify *FIRST* or *LAST*.

Use the drop-down list next to *Operator* to specify a comparison or conditional operator.

Use *Include columns* field to specify columns for *INCLUDE* clause of the constraint. This option is available in Postgres 11 and later.

Click the *SQL* tab to continue.

Your entries in the *Exclusion Constraint* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example 

The following is an example of the sql command generated by user selections in the *Exclusion Constraint* dialog:

The example shown demonstrates creating an exclusion constraint named *exclude_department* that restricts additions to the dept table to those additions that are not equal to the value of the *deptno* column. The constraint uses a btree index.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Extension Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Extension Dialog 

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Extension Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)

[□](#)

[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)

[□](#)

[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)

[□](#)

[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

Extension Dialog



Use the *Extension* dialog to instal a new extension into the current database. An extension is a col ection of SQL objects that add targeted functionality to your Postgres instal ation. The *Extension* dialog adds the functionality of an extension to the current database only; you must register the extension in each database that use the extension. Before you load an extension into a database, you should confirm that any pre-requisite files are instal ed.

The *Extension* dialog allows you to implement options of the CREATE

EXTENSION command through the following dialog tabs: *General* and *Definition*. The SQL tab displays the SQL code generated by dialog selections.

Use the fields in the *General* tab to identify an extension: Use the drop-down listbox in the *Name* field to select the extension.

Each extension must have a unique name.

Store notes about the extension in the *Comment* field.

Click the *Definition* tab to continue.



Use the *Definition* tab to select the *Schema* and *Version*: Use the drop-down listbox next to *Schema* to select the name of the schema in which to instal the extension's objects.

Use the drop-down listbox next to *Version* to select the version of the extension to instal .

Click the *SQL* tab to continue.

Your entries in the *Extension* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Extension* dialog:



The command creates the *chkpass* extension in the *public* schema. It is version 1.0 of *chkpass*.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Foreign Data Wrapper Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Foreign Data Wrapper Dialog 

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Foreign Data Wrapper Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)



[Extension Dialog](#)

Foreign Data Wrapper Dialog

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)



[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

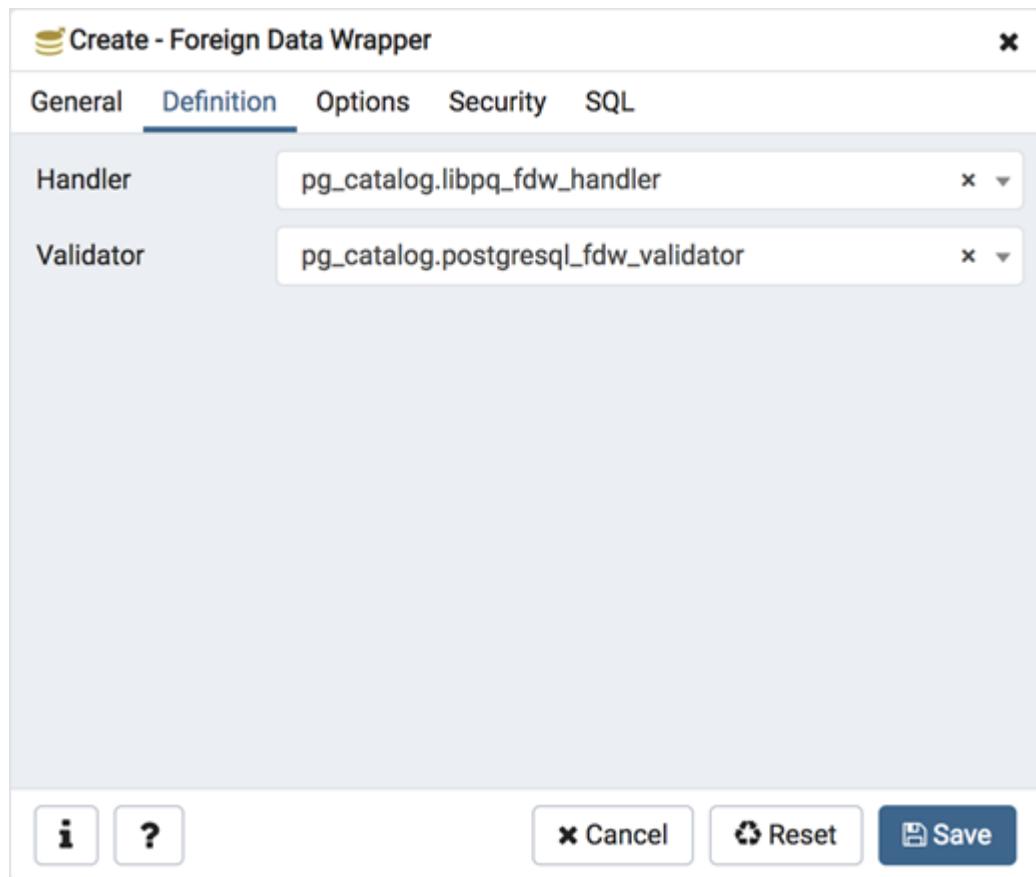
Foreign Data Wrapper Dialog

 Create - Foreign Data Wrapper x

General Definition Options Security SQL

Name	lib_qp_debug
Owner	enterprisedb
Comment	This FDW enables debugging

i ? Cancel Reset Save



Use the *Foreign Data Wrapper* dialog to create or modify a foreign data wrapper. A foreign data wrapper is an adapter between a Postgres database and data stored on another data source.

You must be a superuser to create a foreign data wrapper.

The *Foreign Data Wrapper* dialog organizes the development of a foreign data wrapper through the following dialog tabs: *General*, *Definition*, *Options*, and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

Use the fields in the *General* tab to identify the foreign data wrapper: Use the *Name* field to add a descriptive name for the foreign data wrapper. A foreign data wrapper name must be unique within the database. The name will be displayed in the *pgAdmin* tree control.

Use the drop-down listbox next to *Owner* to select the name of the role that wil own the foreign data wrapper.

Store notes about the foreign data wrapper in the *Comment* field.

Click the *Definition* tab to continue.

The screenshot shows a dialog box titled "Create - Foreign Data Wrapper". At the top, there are tabs: General, Definition, Options (which is underlined, indicating it is active), Security, and SQL. Below the tabs is a section titled "Options" with a table header row containing "Option" and "Value". At the bottom of the dialog are buttons for "Cancel", "Reset", and "Save".

Use the fields in the *Definition* tab to set parameters: Select the name of the handler function from the drop-down listbox in the *Handler* field. This is the name of an existing function that wil be cal ed to retrieve the execution functions for foreign tables.

Select the name of the validator function from the drop-down listbox in the *Validator* field. This is the name of an existing function that wil be cal ed to check the generic options given to the foreign data wrapper, as wel as options for foreign servers, user mappings and foreign tables using the foreign data wrapper.

Click the *Options* tab to continue.

Use the fields in the *Options* tab to specify options:

Click the the *Add* icon (+) button to add an option/value pair for the foreign data wrapper. Supported option/value pairs wil be specific to the selected foreign data wrapper.

Specify the option name in the *Option* field and provide a corresponding value in the *Value* field.

Click the *Add* icon (+) to specify each additional pair; to discard an option, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *Security* tab to continue.

The screenshot shows a dialog box titled "Create - Foreign Data Wrapper". At the top, there are tabs for "General", "Definition", "Options", "Security" (which is underlined, indicating it is active), and "SQL". Below the tabs is a section titled "Privileges" with a table header row containing "Grantee", "Privileges", and "Grantor". A "+" button is located at the top right of this table area. At the bottom of the dialog are buttons for "Cancel", "Reset", and "Save".

Use the **Security** tab to assign security privileges. Click the *Add* icon (+) to assign a set of privileges.

Select the name of the role from the drop-down listbox in the *Grantee* field.

Click inside the *Privileges* field. Check the boxes to the left of one or more privileges to grant the selected privileges to the specified user.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.

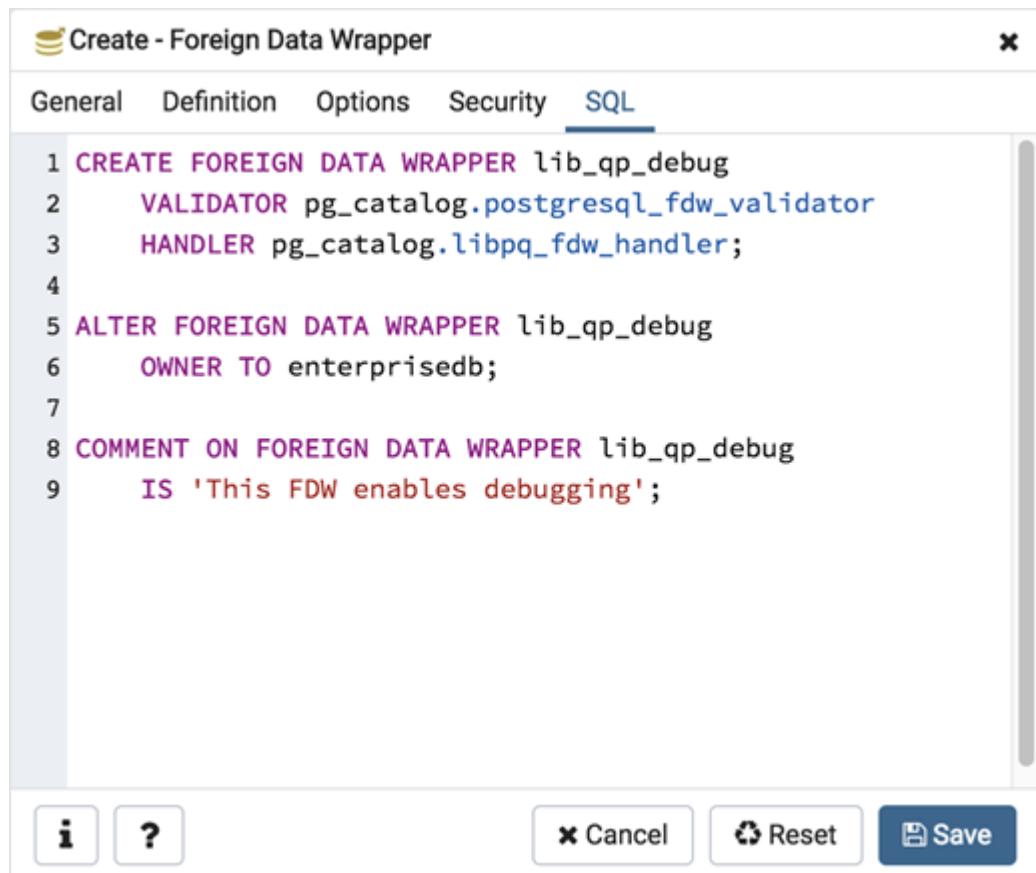
Click *add* to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the **SQL** tab to continue.

Your entries in the *Foreign Data Wrapper* dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Foreign Data Wrapper* dialog:

A screenshot of a PostgreSQL Foreign Data Wrapper (FDW) creation dialog. The title bar says "Create - Foreign Data Wrapper". The tabs at the top are General, Definition, Options, Security, and SQL, with SQL being the active tab. The SQL code area contains the following script:

```
1 CREATE FOREIGN DATA WRAPPER lib_qp_debug
2   VALIDATOR pg_catalog.postgresql_fdw_validator
3   HANDLER pg_catalog.libpq_fdw_handler;
4
5 ALTER FOREIGN DATA WRAPPER lib_qp_debug
6   OWNER TO enterprisedb;
7
8 COMMENT ON FOREIGN DATA WRAPPER lib_qp_debug
9   IS 'This FDW enables debugging';
```

At the bottom are several buttons: an info icon (i), a help icon (?), a cancel button (x Cancel), a reset button (Reset), and a save button (Save).

The example creates a foreign data wrapper named *libpq_debug* that uses pre-existing validator and handler functions, *dblink_fdw_validator* and *libpg_fdw_handler*. Selections on the *Options* tab set *debug* equal to *true*. The foreign data wrapper is owned by *postgres*.

Click the *Help* button (?) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) » [Creation or Modifying a Table](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Foreign key Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
 - [Check Dialog](#)
 - [Column Dialog](#)
 - [Compound Trigger Dialog](#)
 - [Exclusion Constraint Dialog](#)
 - [Foreign key Dialog](#)
 - [Index Dialog](#)
 - [Primary key Dialog](#)
 - [Rule Dialog](#)
 - [Table Dialog](#)
 - [Trigger Dialog](#)
 - [Unique Constraint Dialog](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Foreign key Dialog

Use the Foreign key dialog to specify the behavior of a foreign key constraint. A foreign key constraint maintains referential integrity between two tables. A foreign key constraint cannot be defined between a temporary table and a permanent table.

The Foreign key dialog organizes the development of a foreign key constraint through the following dialog tabs: General, Definition, Columns, and Action. The SQL tab displays the SQL code generated by dialog selections.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Creating or Modifying a Table](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Foreign key Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[□](#)

[Check Dialog](#)

[Column Dialog](#)

[Latest Docs](#)

[Compound Trigger Dialog](#)

[Exclusion Constraint Dialog](#)

[□](#)

[Foreign key Dialog](#)

[Index Dialog](#)

[Ask a](#)

[Primary key Dialog](#)

[Question](#)

[Rule Dialog](#)

[Table Dialog](#)

[□](#)

[Trigger Dialog](#)

[Report a Bug](#)

[Unique Constraint Dialog](#)

[Management Basics](#)

[Backup and Restore](#)



[Developer Tools](#)

[Screenshots](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Foreign key Dialog

Use the *Foreign key* dialog to specify the behavior of a foreign key constraint. A foreign key constraint maintains referential integrity between two tables. A foreign key constraint cannot be defined between a temporary table and a permanent table.

The *Foreign key* dialog organizes the development of a foreign key constraint through the following dialog tabs: *General*, *Definition*, *Columns*, and *Action*. The *SQL* tab displays the SQL code generated by dialog selections.

Create - Foreign key



General Definition Columns Action SQL

Name

fk_orders

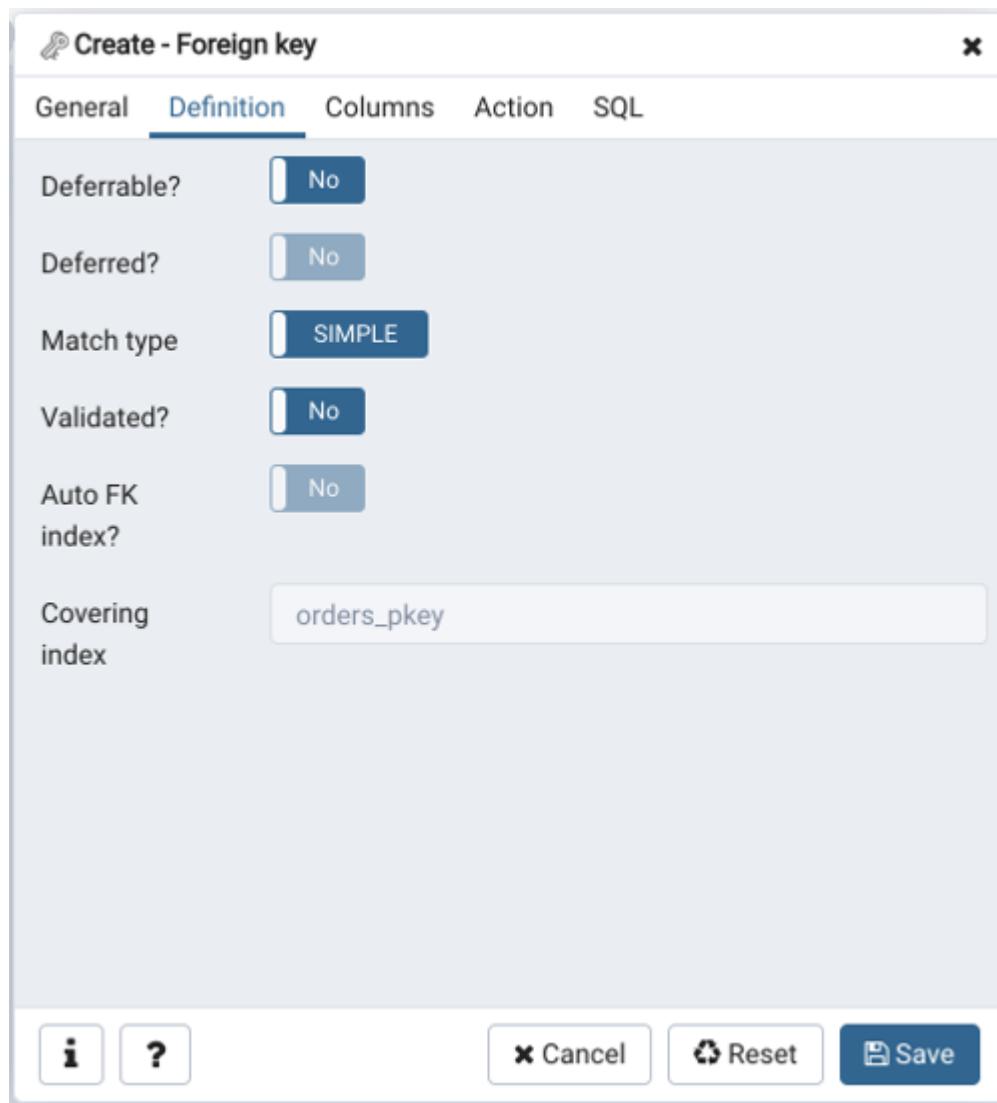
Comment



Cancel

Reset

Save

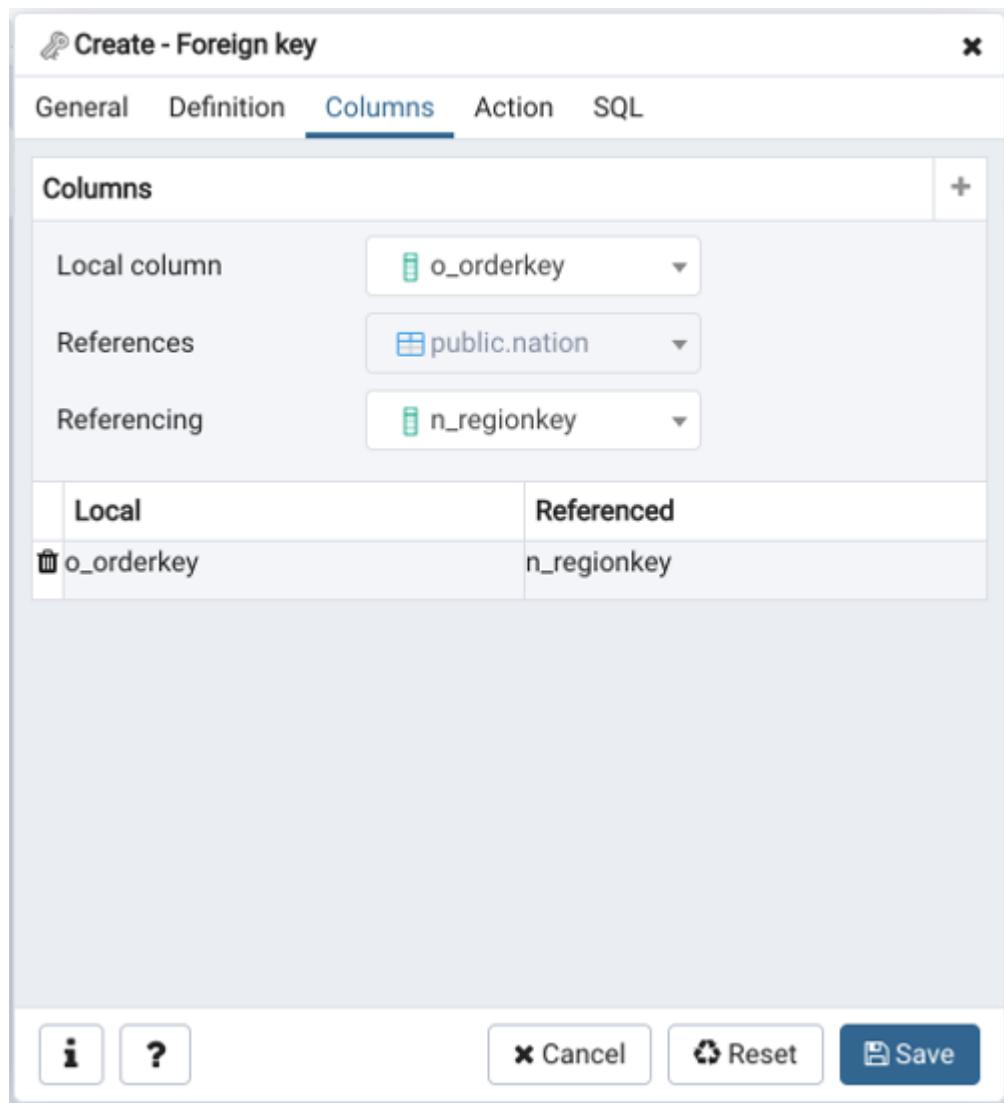


Use the fields in the *General* tab to identify the foreign key constraint: Use the *Name* field to add a descriptive name for the foreign key. The name will be displayed in the pgAdmin tree control.

Store notes about the foreign key constraint in the *Comment* field.

Click the *Definition* tab to continue.

Use the fields in the *Definition* tab to define the foreign key constraint: Move the *Deferrable?* switch to the *Yes* position to specify the timing of the constraint is deferrable and can be postponed until the end of the statement. The default is *No*.



If enabled, move the *Deferred?* switch to the *Yes* position to specify the timing of the constraint is deferred to the end of the statement.

The default is *No*.

Move the *Match type* switch specify the type of matching that is enforced by the constraint:

Select *Full* to indicate that all columns of a multicolumn foreign key must be null if any column is null ; if all columns are null , the row is not required to have a match in the referenced table.

Select *Simple* to specify that a single foreign key column may be null; if any column is null, the row is not required to have a match in the referenced table.

Move the *Validated* switch to the *Yes* position to instruct the server to validate the existing table content (against a foreign key or check constraint) when you save modifications to this dialog.

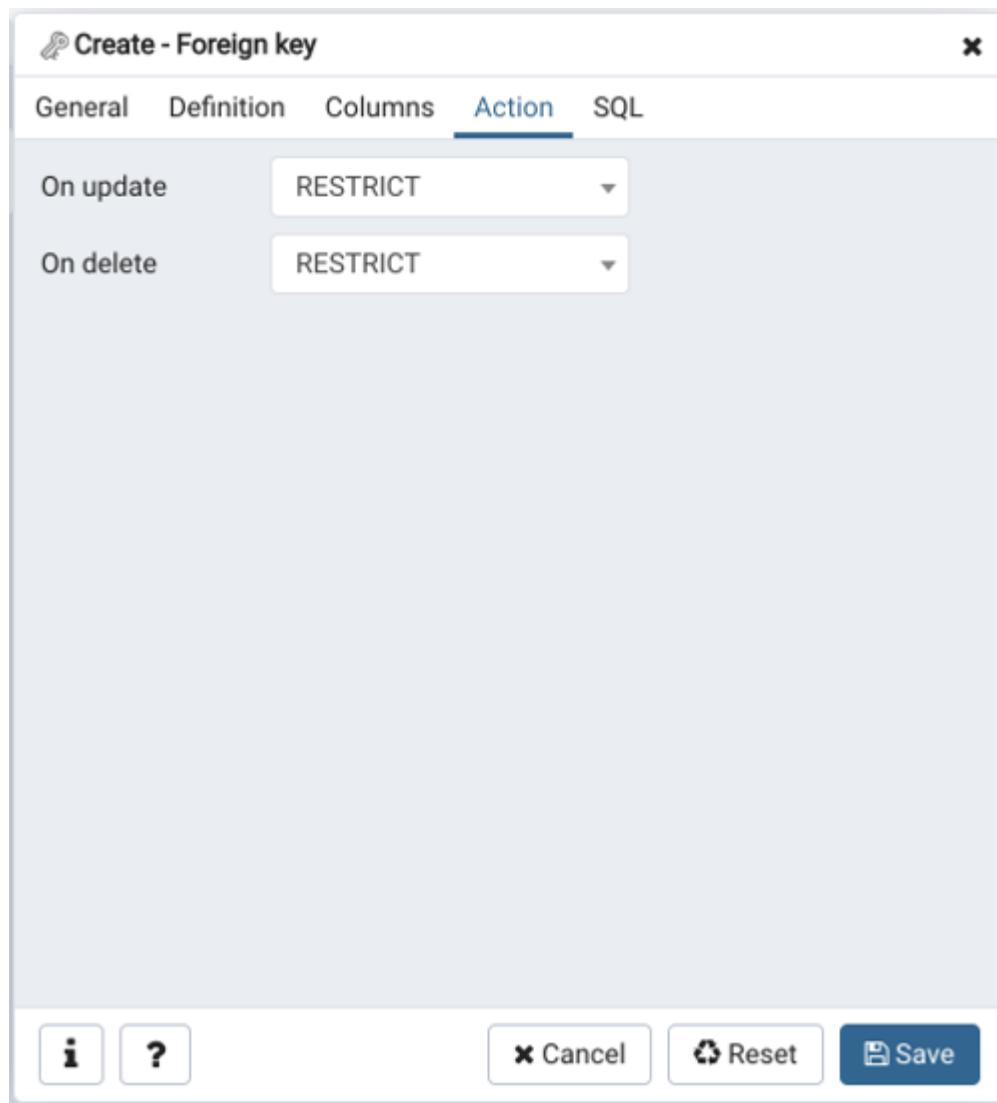
Move the *Auto FK Index* switch to the *No* position to disable the automatic index feature.

The field next to *Covering Index* generates the name of an index if the *Auto FK Index* switch is in the *Yes* position; or, this field is disabled.

Click the *Columns* tab to continue.

Use the fields in the *Columns* tab to specify one or more reference column(s). A Foreign Key constraint requires that one or more columns of a table must only contain values that match values in the referenced column(s) of a row of a referenced table:

Use the drop-down listbox next to *Local column* to specify the column in the current table that will be compared to the foreign table.



Use the drop-down listbox next to *References* to specify the name of the table in which the comparison column(s) resides.

Use the drop-down listbox next to *Referencing* to specify a column in the foreign table.

Click the *Add* icon (+) to add a column to the list; repeat the steps above and click the *Add* icon (+) to add additional columns. To discard an entry, click the trash icon to the left of the entry and confirm deletion in the *Delete Row* popup.

Click the *Action* tab to continue.

Use the drop-down listboxes on the *Action* tab to specify behavior related to the foreign key constraint that wil be performed when data within the table is updated or deleted:

Use the drop-down listbox next to *On update* to select an action that wil be performed when data in the table is updated.

Use the drop-down listbox next to *On delete* to select an action that wil be performed when data in the table is deleted.

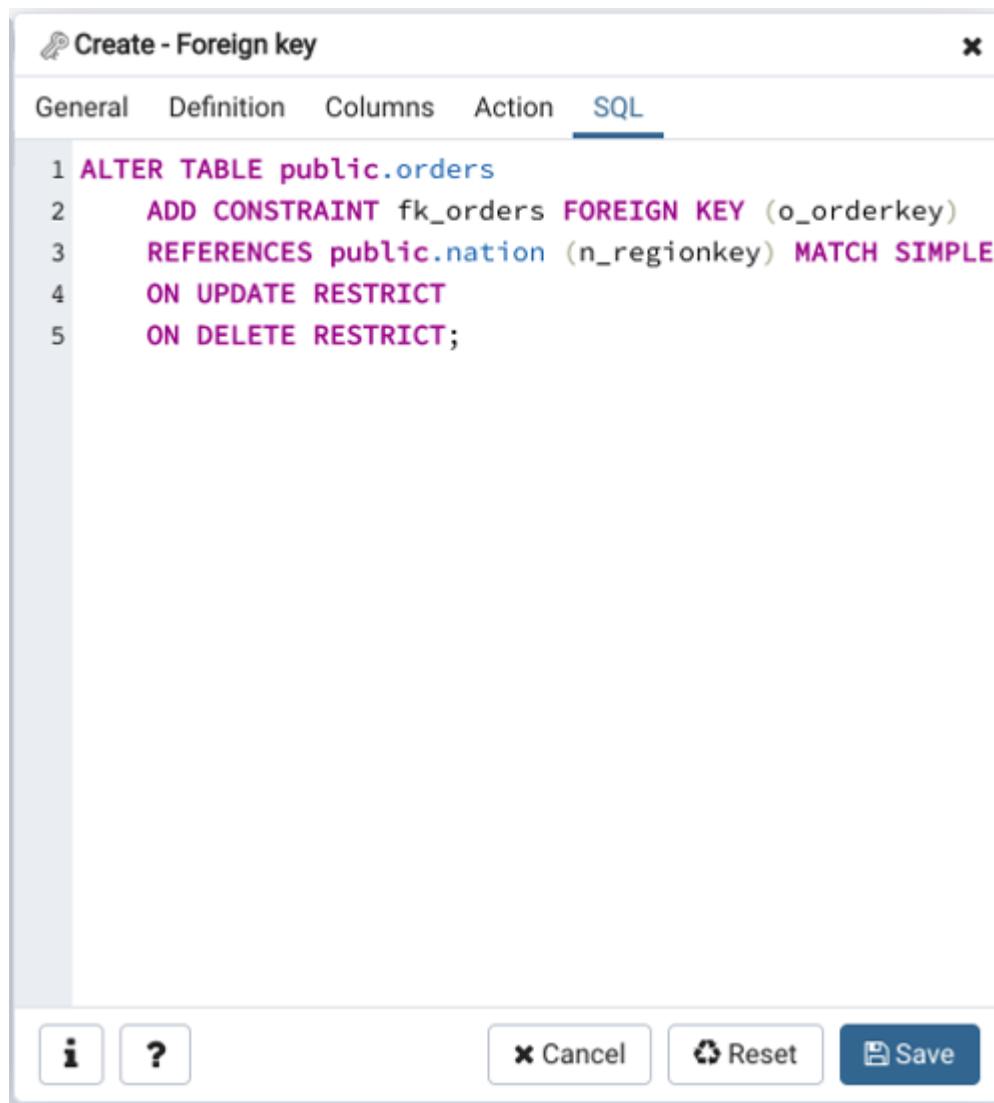
The supported actions are:

NO

Produce an error indicating that the deletion or update wil ACTION create a foreign key constraint violation. If the constraint is deferred, this error wil be produced at constraint check time if any referencing rows stil exist. This is the default.

RESTRICT Throw an error indicating that the deletion or update would create a foreign key constraint violation. This is the same as NO ACTION except that the check is not deferrable.

CASCADE Delete any rows referencing the deleted row, or update the values of the referencing column(s) to the new values of the referenced columns, respectively.

A screenshot of a database management software interface titled "Create - Foreign key". The window has tabs at the top: General, Definition, Columns, Action, and SQL. The SQL tab is selected, showing the following SQL code:

```
1 ALTER TABLE public.orders
2   ADD CONSTRAINT fk_orders FOREIGN KEY (o_orderkey)
3     REFERENCES public.nation (n_regionkey) MATCH SIMPLE
4     ON UPDATE RESTRICT
5     ON DELETE RESTRICT;
```

At the bottom of the window are several buttons: an information icon (i), a question mark icon (?), a "Cancel" button, a "Reset" button, and a "Save" button.

SET NULL Set the referencing column(s) to null .

SET

Set the referencing column(s) to their default values. There DEFAULT must be a row in the referenced table that matches the default values (if they are not null), or the operation will fail.

Click the **SQL** tab to continue.

Your entries in the *Foreign key* dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Foreign key* dialog:

The example shown demonstrates creating a foreign key constraint named *territory_fkey* that matches values in the *distributors* table *territory* column with those of the *sales_territories* table *region* column.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Foreign Server Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Foreign Server Dialog 

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Foreign Server Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)



[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

Foreign Server Dialog

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)



[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

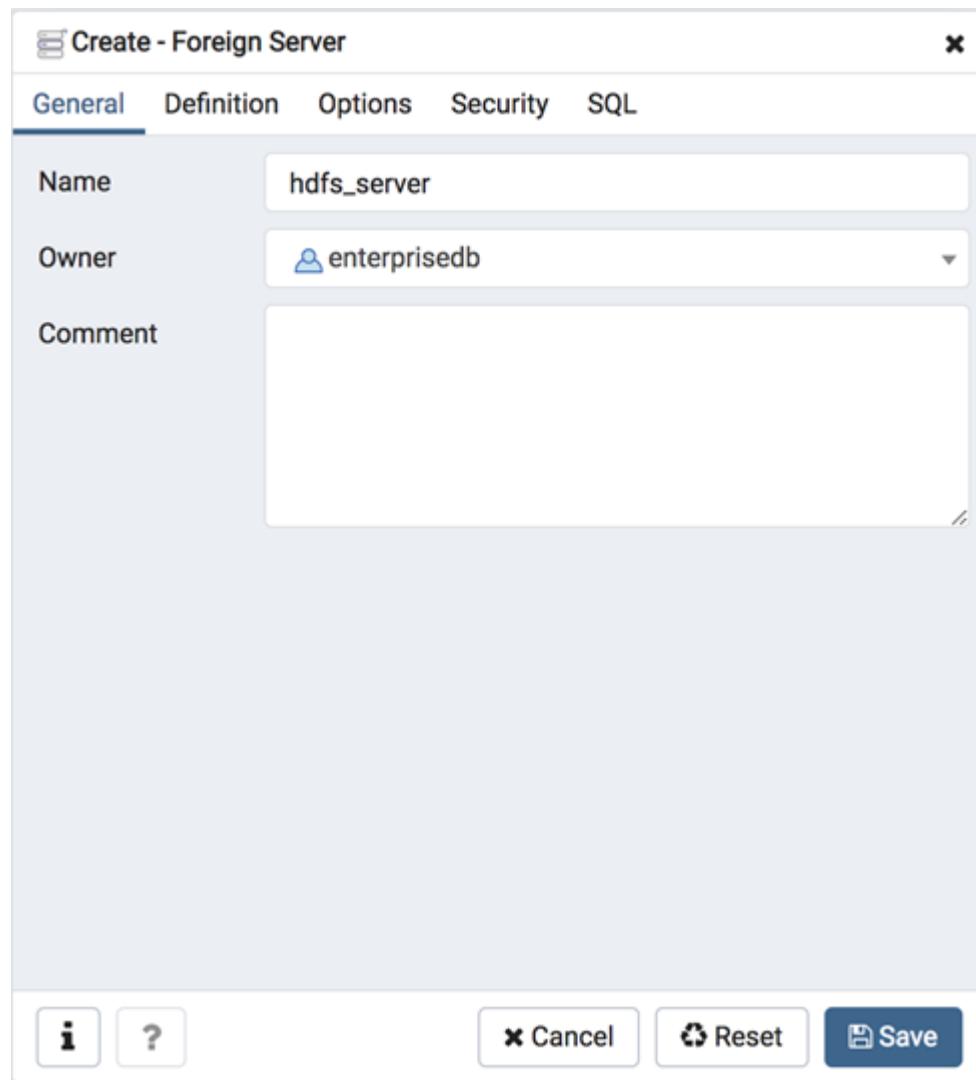
[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

Foreign Server Dialog



Use the *Foreign Server* dialog to create a foreign server. A foreign server typically encapsulates connection information that a foreign-data wrapper uses to access an external data resource. Each foreign data wrapper may connect to a different foreign server; in the *pgAdmin* tree control, expand the node of the applicable foreign data wrapper to launch the *Foreign Server* dialog.

The *Foreign Server* dialog organizes the development of a foreign server through the following dialog tabs: *General*, *Definition*,

Options, and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

Use the fields in the *General* tab to identify the foreign server: Use the *Name* field to add a descriptive name for the foreign server.

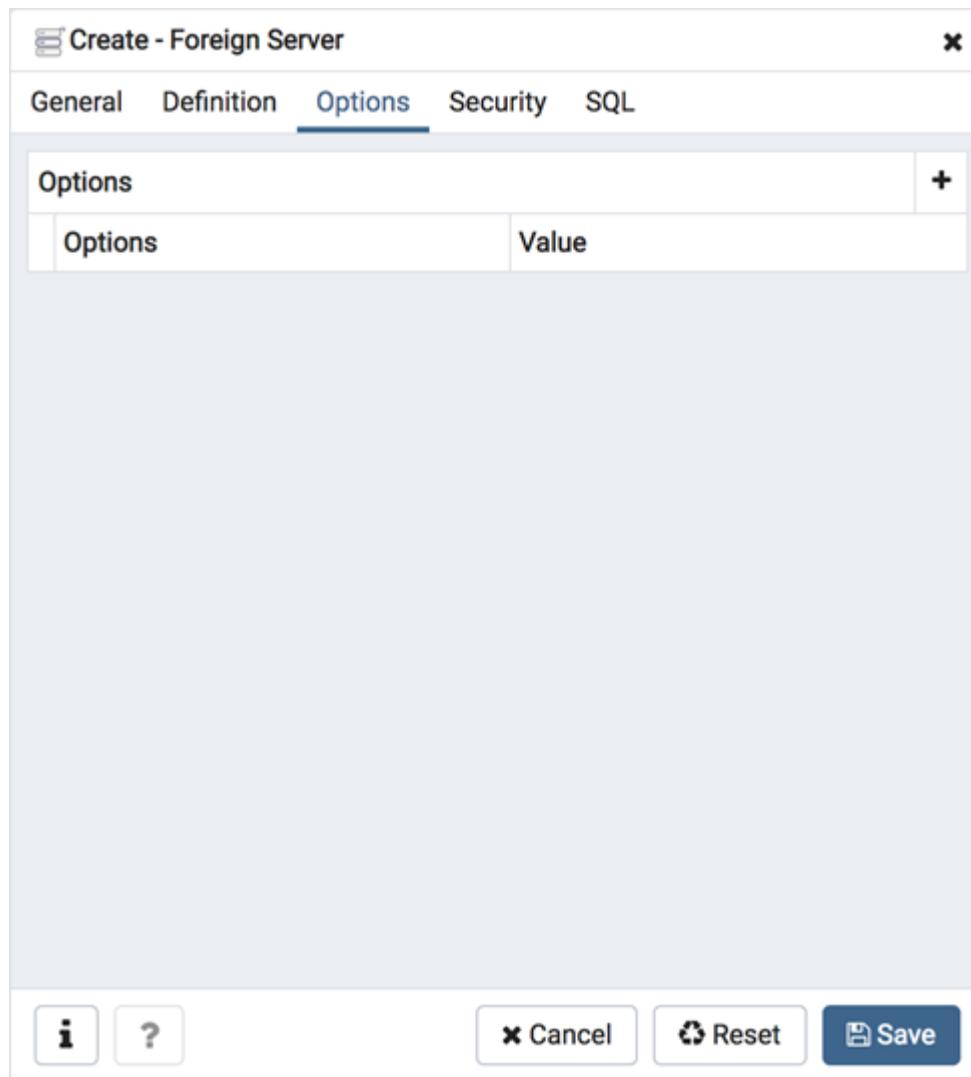
The name will be displayed in the *pgAdmin* tree control. It must be unique within the database.

Use the drop-down listbox next to *Owner* to select a role.

Store notes about the foreign server in the *Comment* field.

Click the *Definition* tab to continue.

The screenshot shows a dialog box titled "Create - Foreign Server". At the top, there are five tabs: "General", "Definition" (which is underlined, indicating it is active), "Options", "Security", and "SQL". Below the tabs, there are two input fields: "Type" and "Version", each with a corresponding text input box. At the bottom of the dialog, there is a row of buttons: an information icon ("i"), a question mark icon ("?"), a "Cancel" button with a "x" icon, a "Reset" button with a circular arrow icon, and a "Save" button with a disk icon.



Use the fields in the *Definition* tab to set parameters: Use the *Type* field to specify a server type.

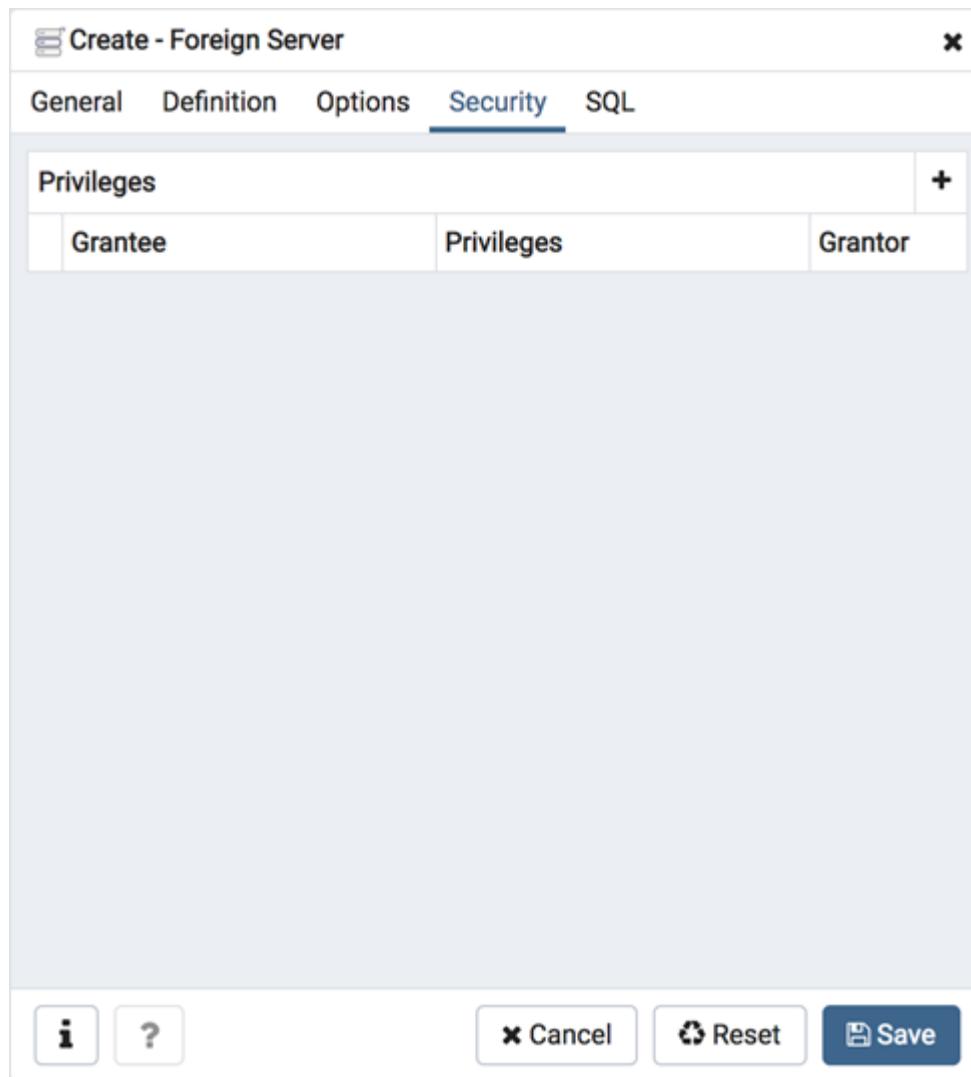
Use the *Version* field to specify a server version.

Click the *Options* tab to continue.

Use the fields in the *Options* tab to specify options. Click the *Add* button to create an option clause for the foreign server.

Specify the option name in the *Option* field.

Provide a corresponding value in the *Value* field.



Click *Add* to create each additional clause; to discard an option, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *Security* tab to continue.

Use the *Security* tab to assign security privileges to the foreign server.

Click *Add* before you assign a set of privileges.

Select the name of the role from the drop-down listbox in the *Grantee* field.

Click inside the *Privileges* field. Check the boxes to the left of one or more privileges to grant the selected privileges to the specified user.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.

Click *Add* to assign a new set of privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the *Delete Row* dialog.

Click the *SQL* tab to continue.

Your entries in the *Foreign Server* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Foreign Server* dialog:

Create - Foreign Server X

General Definition Options Security **SQL**

```
1 CREATE SERVER hdfs_server
2   FOREIGN DATA WRAPPER libpq_dblink;
3
4
5 ALTER SERVER hdfs_server
6   OWNER TO enterprisedb;
```

i **?** **Cancel** **Reset** **Save**

The screenshot shows a 'Create - Foreign Server' dialog box. At the top, there are tabs for General, Definition, Options, Security, and SQL. The SQL tab is selected, displaying the following PostgreSQL SQL code:

```
1 CREATE SERVER hdfs_server
2   FOREIGN DATA WRAPPER libpq_dblink;
3
4
5 ALTER SERVER hdfs_server
6   OWNER TO enterprisedb;
```

At the bottom of the dialog are several buttons: Info (i), Help (?), Cancel, Reset, and Save. The Save button is highlighted in blue.

The example shown demonstrates creating a foreign server for the foreign data wrapper *hdfs_fdw*. It has the name *hdfs_server*; its type is *hiveserver2*. Options for the foreign server include a host and a port.

Click the *Info* button (i) to access online help.

Click the **Save** button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Foreign Table Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Foreign Table Dialog

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Foreign Table Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)



[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

Foreign Table Dialog

[FTS Configuration Dialog](#)



[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

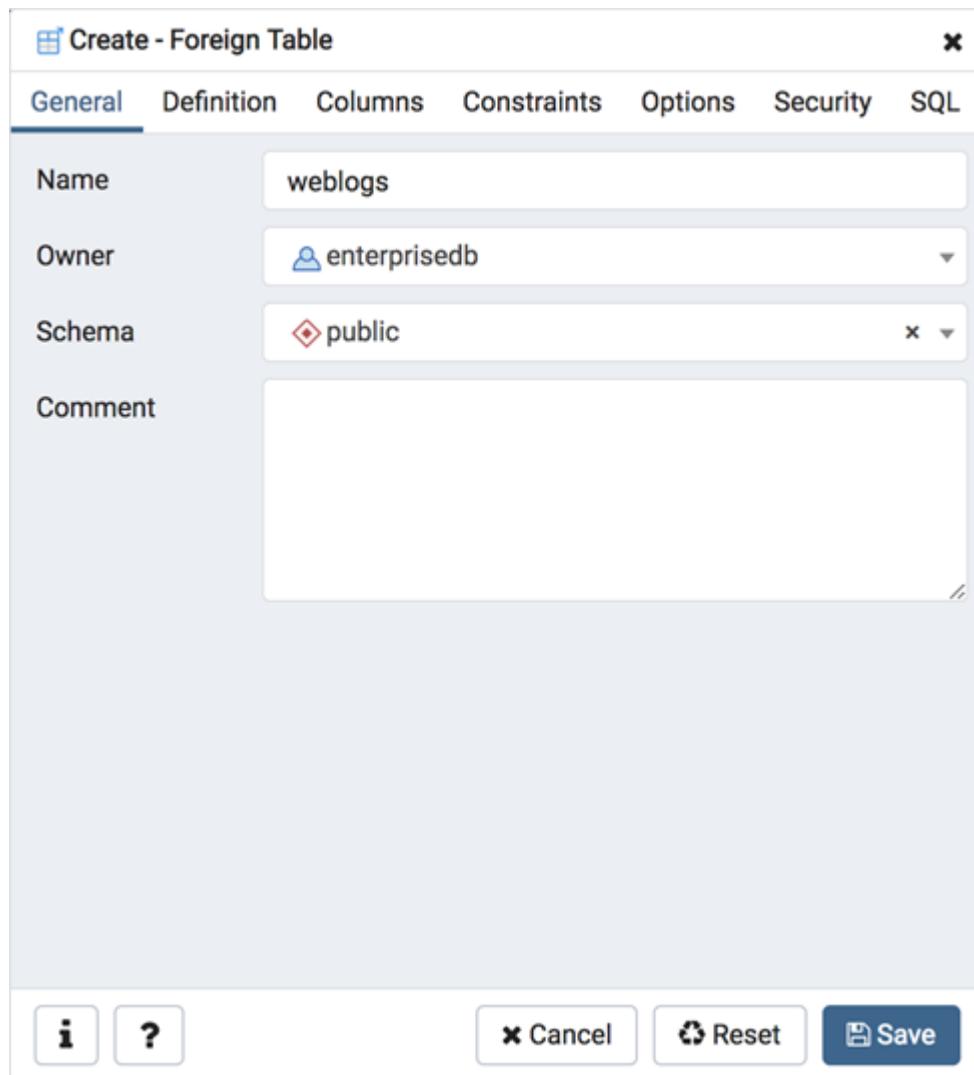
[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

Foreign Table Dialog



Use the *Foreign Table* dialog to define a foreign table in the current database. Foreign tables define the structure of an external data source that resides on a foreign server.

The *Foreign Table* dialog organizes the development of a foreign table through the following dialog tabs: *General*, *Definition*, *Columns*, *Constraints*, *Options*, and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

Use the fields in the *General* tab to identify the foreign table: Use the *Name* field to add a descriptive name for the foreign table.

The name of the foreign table must be distinct from the name of any other foreign table, table, sequence, index, view, existing data type, or materialized view in the same schema. The name will be displayed in the *pgAdmin* tree control.

Use the drop-down listbox next to *Owner* to select the name of the role that will own the foreign table.

Select the name of the schema in which the foreign table will reside from the drop-down listbox in the *Schema* field.

Store notes about the foreign table in the *Comment* field.

Click the *Definition* tab to continue.

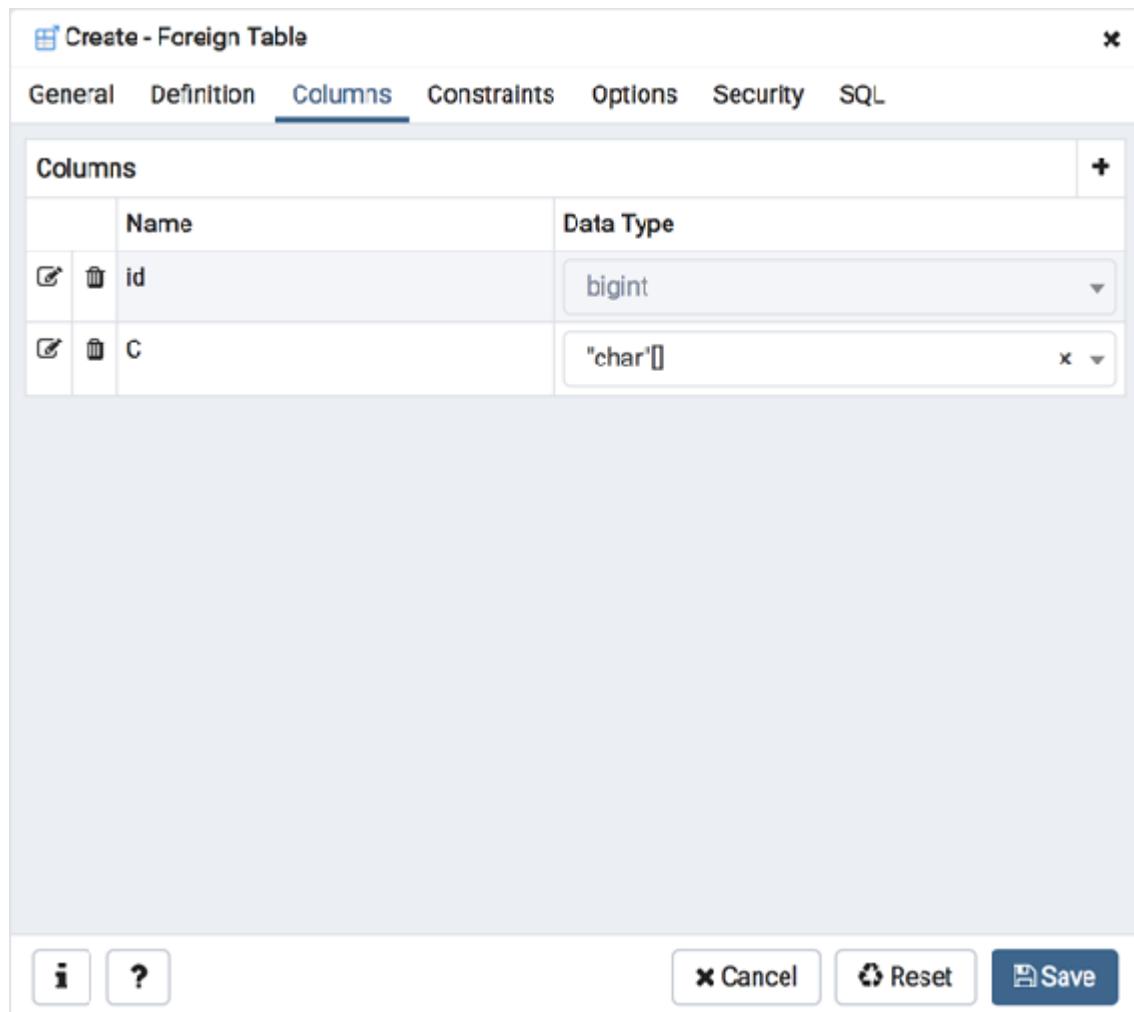
 Create - Foreign Table x

General **Definition** Columns Constraints Options Security SQL

Foreign server x ▾

Inherits

i ? ✖ Cancel ↻ Reset 💾 Save



Use the fields in the *Definition* tab to define the external data source:
Use the drop-down listbox next to *Foreign server* to select a foreign server. This list is populated with servers defined through the *Foreign Server* dialog.

Use the drop-down listbox next to *Inherits* to specify a parent table.

The foreign table will inherit all of its columns. This field is optional.

Click the *Columns* tab to continue.

Use the fields in the *Columns* tab to add columns and their attributes to the table. Click the Add icon (+) to define a column: Use the *Name* field to add a descriptive name for the column.

The screenshot shows the 'Create - Foreign Table' dialog box with the 'Constraints' tab selected. The 'Constraints' section contains a table with the following data:

	Name	Check	No Inherit	Validate?
<input checked="" type="checkbox"/> X	myconstraint	mycheck	<input type="checkbox"/>	<input checked="" type="checkbox"/>

At the bottom of the dialog are buttons for 'Cancel', 'Reset', and 'Save'.

Use the drop-down listbox in the *Data Type* field to select a data type for the column. This can include array specifiers. For more information on which data types are supported by PostgreSQL, refer to Chapter 8 of the core documentation.

Click the *Add* icon (+) to specify each additional column; to discard a column, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *Constraints* tab to continue.

Use the fields in the *Constraints* tab to apply a table constraint to the foreign table. Click the *Add* icon (+) to define a constraint: Use the *Name* field to add a descriptive name for the constraint. If the constraint is violated, the constraint name is present in error

messages, so constraint names like *col must be positive* can be used to communicate helpful information.

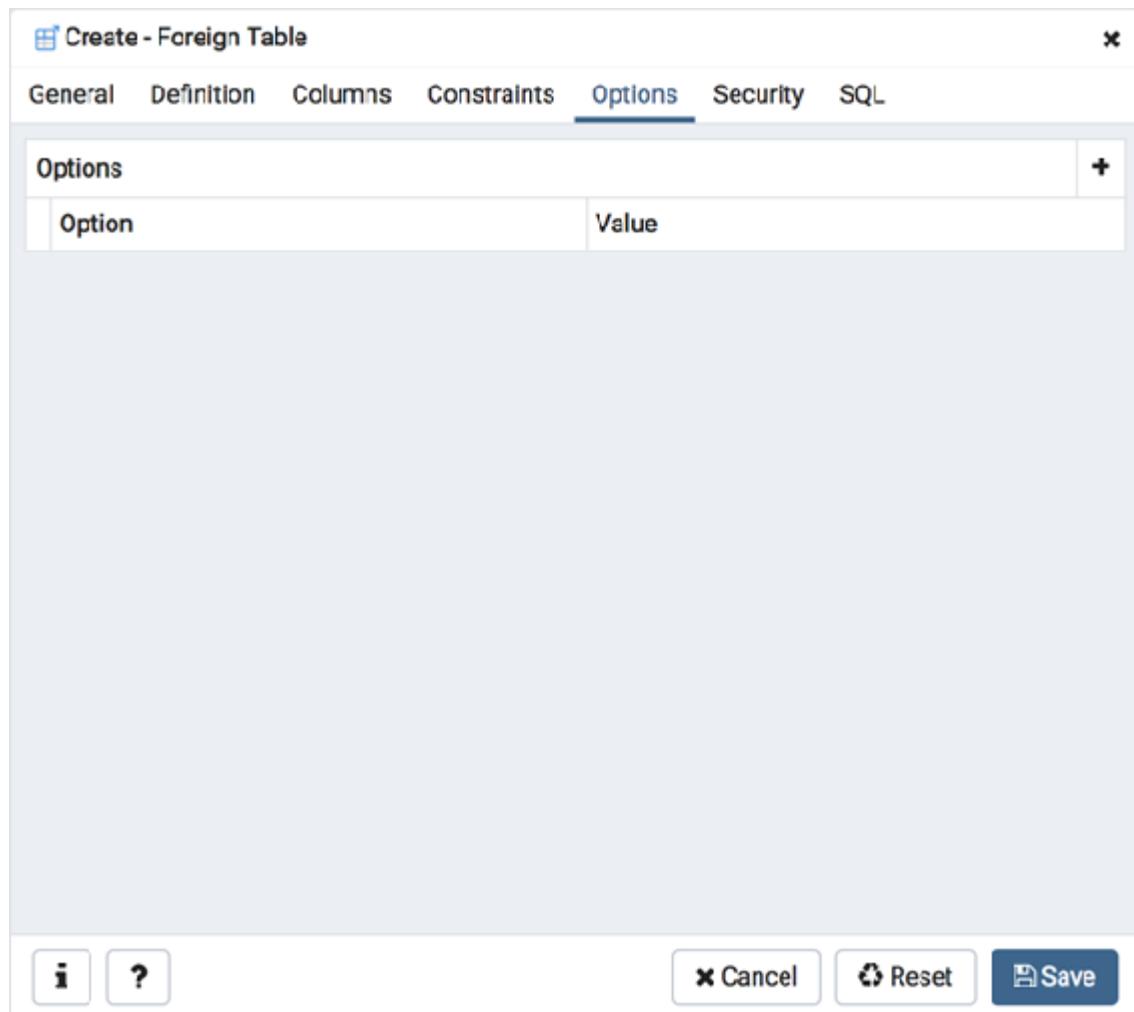
Use the *Check* field to write a check expression producing a Boolean result. Each row in the foreign table is expected to satisfy the check expression.

Check the *No Inherit* checkbox to specify that the constraint will not propagate to child tables.

Uncheck the *Validate* checkbox to disable validation. The database will not assume that the constraint holds for all rows in the table.

Click the *Add* icon (+) to specify each additional constraint; to discard a constraint, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *Options* tab to continue.



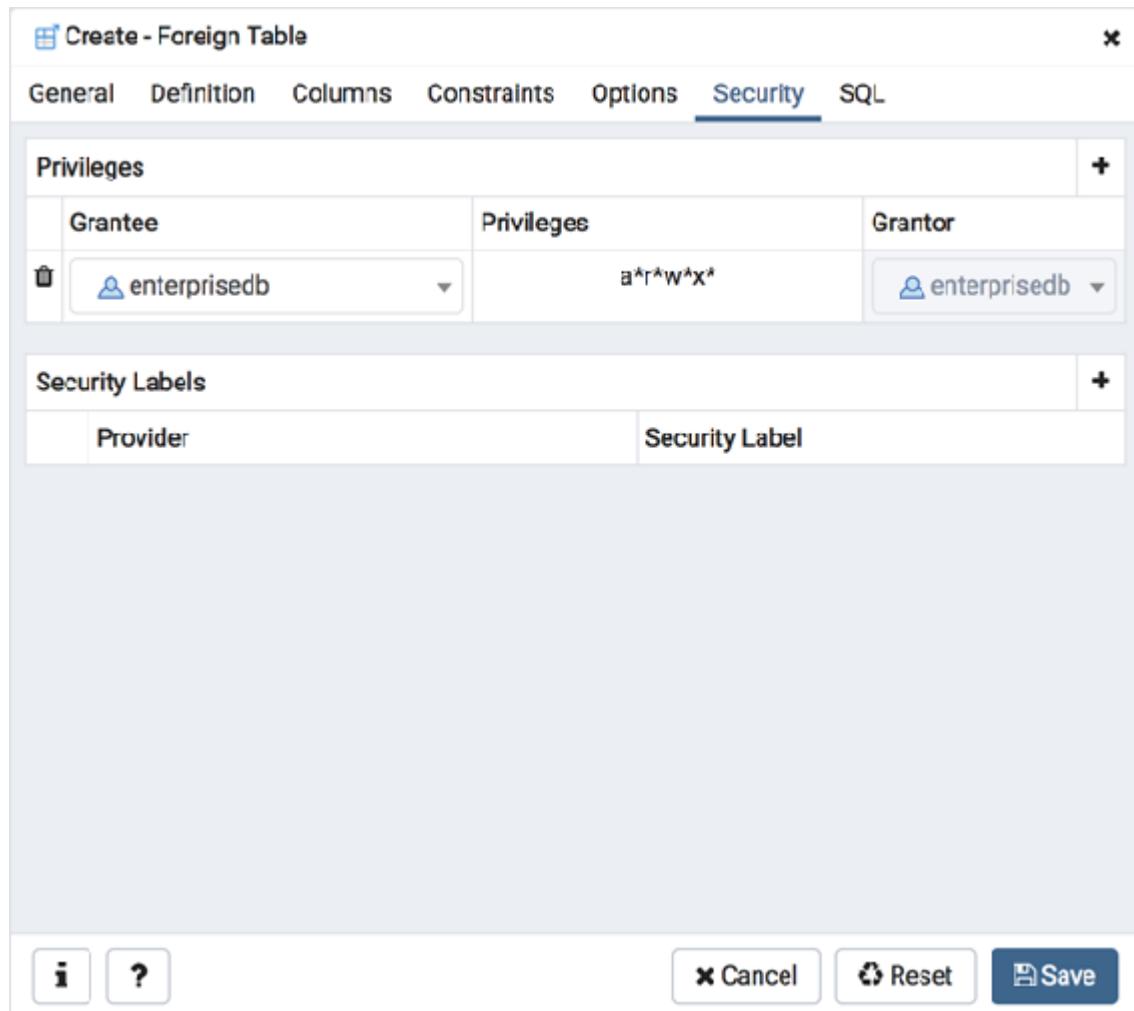
Use the fields in the *Options* tab to specify options to be associated with the new foreign table or one of its columns; the accepted option names and values are specific to the foreign data wrapper associated with the foreign server. Click the *Add* icon (+) to add an option/value pair.

Specify the option name in the *Option* field. Duplicate option names are not allowed.

Provide a corresponding value in the *Value* field.

Click the *Add* icon (+) to specify each additional option/value pair; to discard an option, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *Security* tab to continue.



Use the *Security* tab to assign privileges and define security labels.

Use the *Privileges* panel to assign privileges to a role. Click the Add icon (+) to set privileges for database objects:

Select the name of the role to which privileges will be assigned from the drop-down listbox in the *Grantee* field.

Click inside the *Privileges* field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.

Click the *Add* icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Use the *Security Labels* panel to define security labels applied to the function. Click the *Add* icon (+) to add each security label selection: Specify a security label provider in the *Provider* field. The named provider must be loaded and must consent to the proposed labeling operation.

Specify a a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL

places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the *Add* icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

Create - Foreign Table

General Definition Columns Constraints Options Security SQL

```
1 CREATE FOREIGN TABLE public.weblogs(
2
3 )
4     INHERITS (public.test)
5     SERVER hdfs_server;
6
7 ALTER FOREIGN TABLE public.weblogs
8     OWNER TO enterprisedb;
9
10 ALTER FOREIGN TABLE public.weblogs
11     ADD CONSTRAINT myconstraint CHECK (mycheck);
12
13 GRANT ALL ON TABLE public.weblogs TO enterprisedb WITH GRANT OPTION;
14
15
```

i ? Cancel Reset Save



Your entries in the *Foreign Table* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example ↴

The following is an example of the sql command generated by user selections in the *Foreign Table* dialog:

The example shown demonstrates creating a foreign table *weblogs* with multiple columns and two options.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [FTS Configuration Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

FTS Configuration Dialog ¶

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[FTS Configuration Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)

[□](#)

[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)

[□](#)

[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

FTS Configuration Dialog

[□](#)

[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

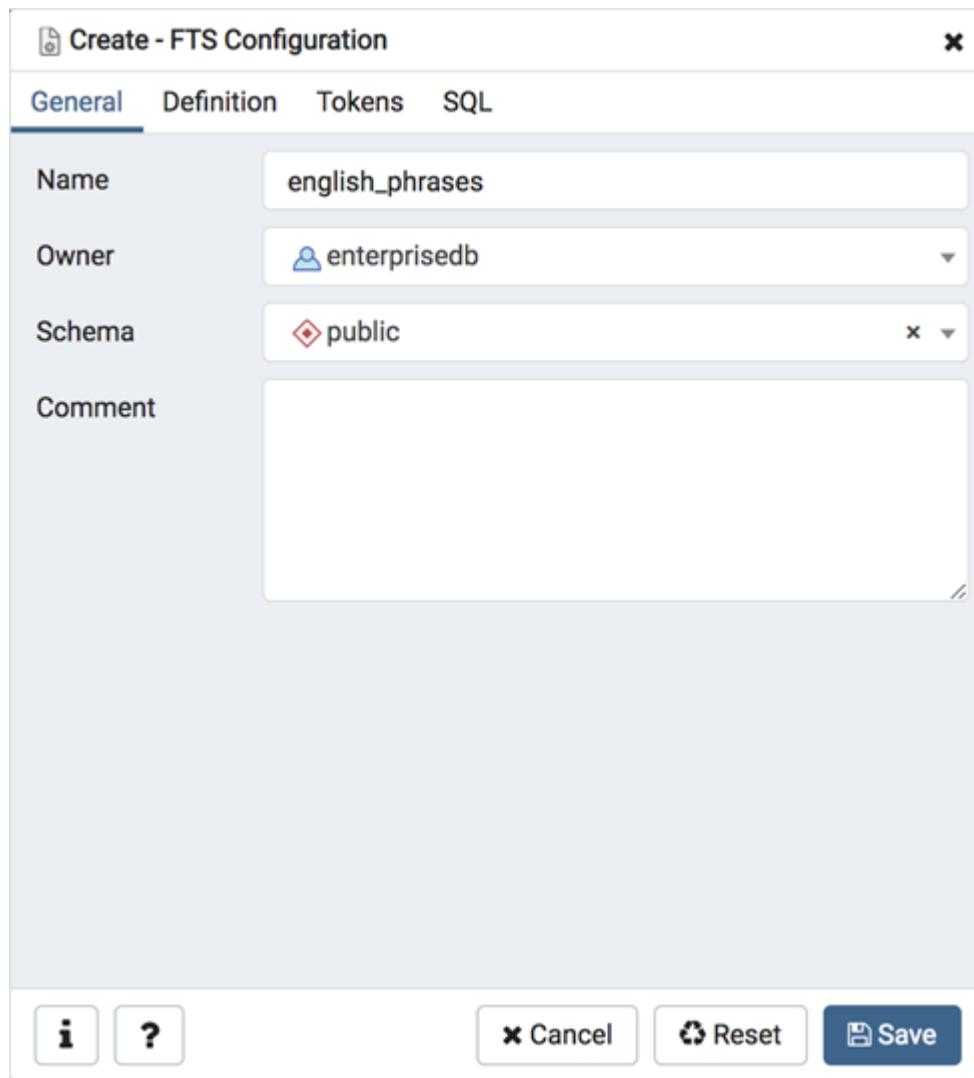
[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

FTS Configuration Dialog



Use the *FTS Configuration* dialog to configure a full text search. A text search configuration specifies a text search parser that can divide a string into tokens, along with dictionaries that can identify searchable tokens.

The *FTS Configuration* dialog organizes the development of a FTS configuration through the following dialog tabs: “*General*, *Definition*, and *Tokens*. The *SQL* tab displays the SQL code generated by

dialog selections.

Click the *General* tab to begin.

Use the fields in the *General* tab to identify a FTS configuration: Use the *Name* field to add a descriptive name for the FTS

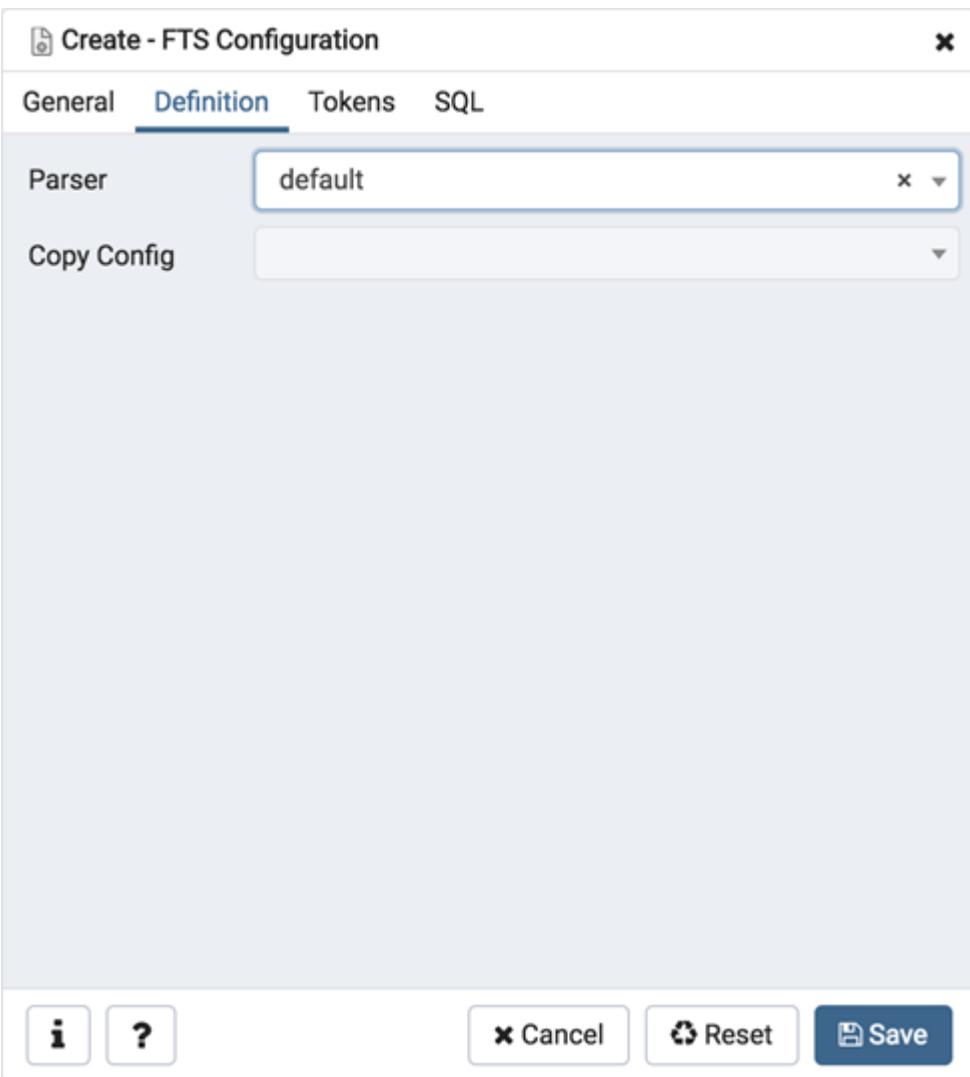
configuration. The name will be displayed in the *pgAdmin* tree control.

Use the drop-down listbox next to *Owner* to specify the role that will own the configuration.

Select the name of the schema in which the FTS configuration will reside from the drop-down listbox in the *Schema* field.

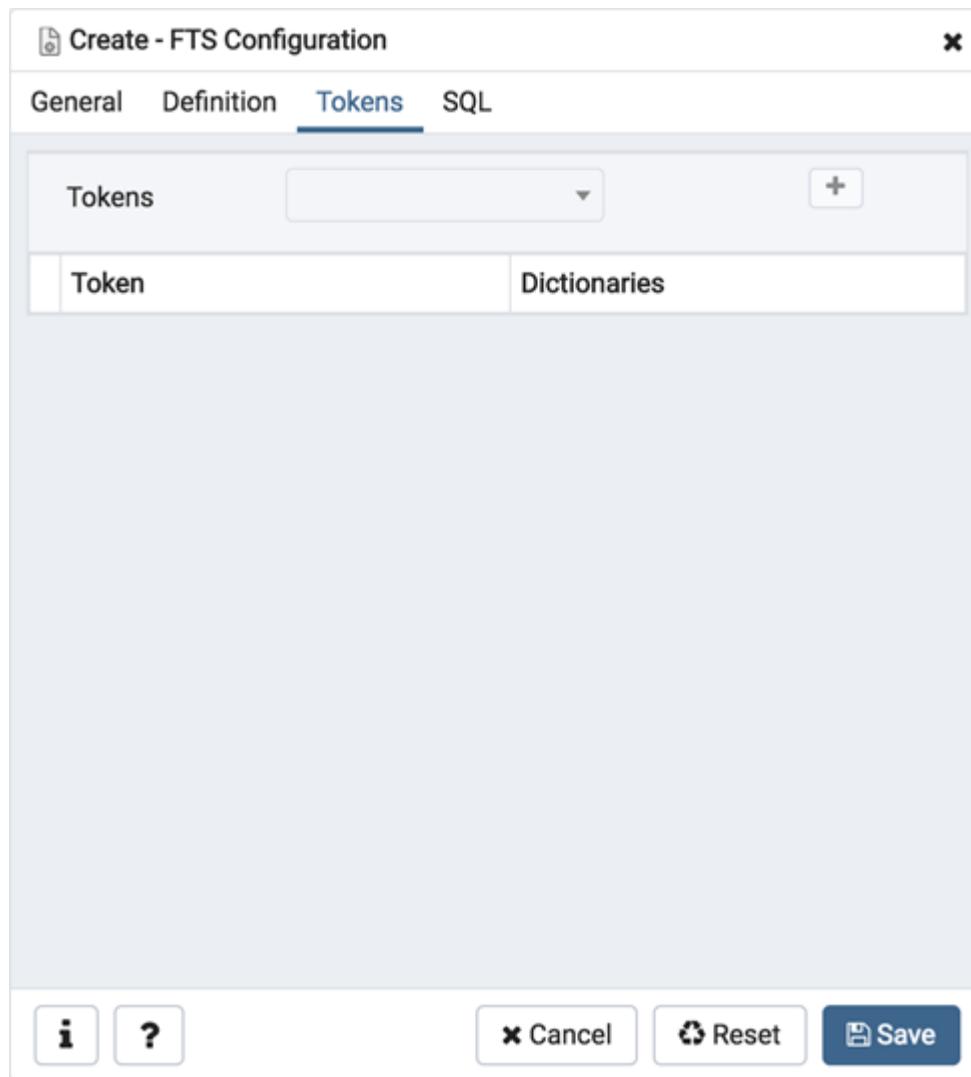
Store notes about the FTS configuration in the *Comment* field.

Click the *Definition* tab to continue.

A screenshot of a "Create - FTS Configuration" dialog box. The title bar shows the window title. Below it is a navigation bar with tabs: "General", "Definition" (which is selected and underlined), "Tokens", and "SQL".
The main content area contains two sections:

- "Parser": A dropdown menu set to "default".
- "Copy Config": A dropdown menu.

At the bottom of the dialog are several buttons: an "i" button (info), a "?" button (help), a "Cancel" button, a "Reset" button, and a "Save" button (highlighted in blue).



Use the fields in the *Definition* tab to define parameters: Select the name of the text search parser from the drop-down listbox in the *Parser* field.

Select a language from the drop-down listbox in the *Copy Config* field.

Click the *Tokens* tab to continue.

Use the fields in the *Tokens* tab to add a token:

Use the *Tokens* field to specify the name of a token.

Click the *Add* icon (+) to create a token.

Use the *Dictionaries* field to specify a dictionary.



Repeat these steps to add additional tokens; to discard a token, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

Your entries in the *FTS Configuration* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example [¶](#)

The following is an example of the sql command generated by user selections in the *FTS Configuration* dialog:

The example shown demonstrates creating a FTS configuration named *meme_phrases*. It uses the *default* parser.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [FTS Dictionary Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

FTS Dictionary Dialog ¶

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[FTS Dictionary Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)



[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)



FTS Dictionary Dialog

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

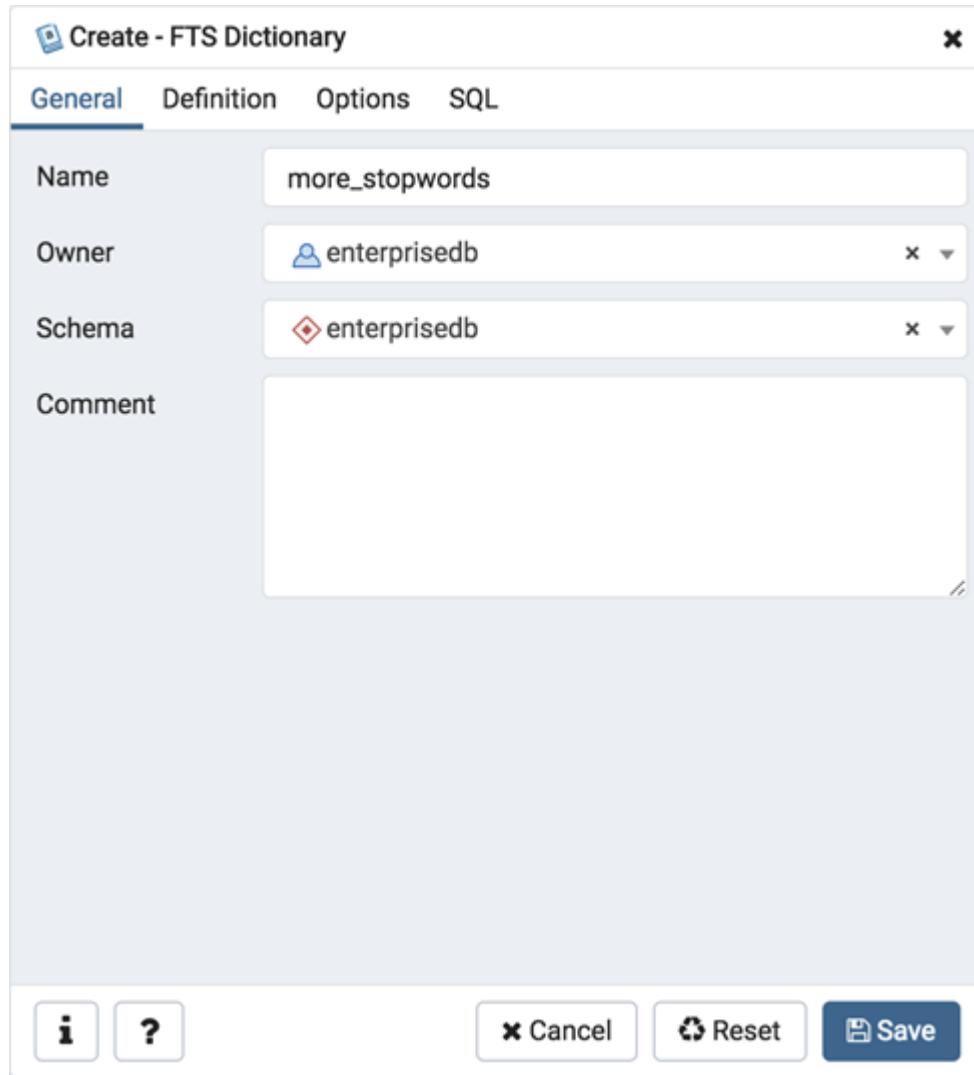
[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

FTS Dictionary Dialog



Use the *FTS Dictionary* dialog to create a full text search dictionary. You can use a predefined template or create a new dictionary with custom parameters.

The *FTS Dictionary* dialog organizes the development of a FTS dictionary through the following dialog tabs: *General*, *Definition*, and *Options*. The *SQL* tab displays the SQL code generated by dialog selections.

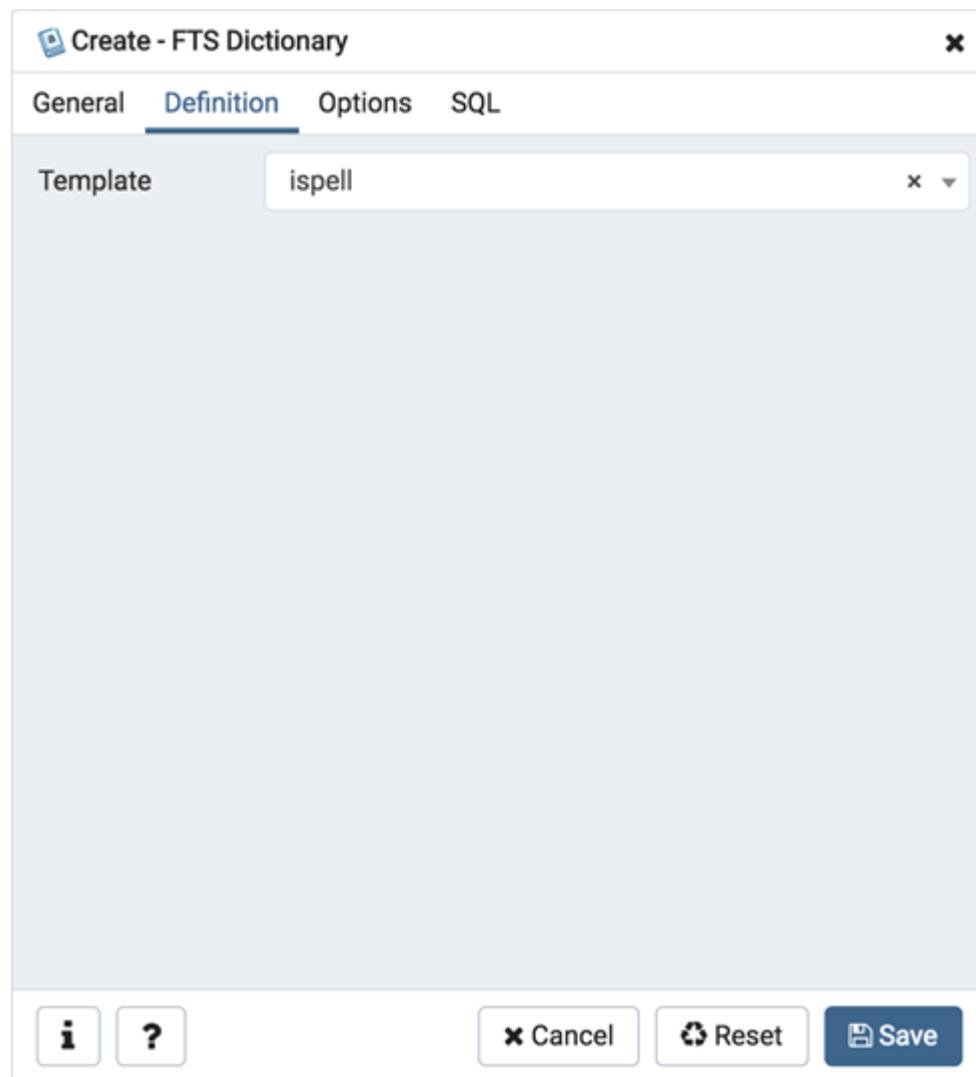
Use the fields in the *General* tab to identify the dictionary: Use the *Name* field to add a descriptive name for the dictionary. The name will be displayed in the *pgAdmin* tree control.

Use the drop-down listbox next to *Owner* to select the role that will own the FTS Dictionary.

Select the name of the schema in which the dictionary will reside from the drop-down listbox in the *Schema* field.

Store notes about the dictionary in the *Comment* field.

Click the *Definition* tab to continue.



Use the field in the *Definition* tab to choose a template from the drop-down listbox:

Select *ispel* to select the Ispell template. The Ispell dictionary template supports morphological dictionaries, which can normalize many different linguistic forms of a word into the same lexeme. For example, an English Ispell dictionary can match all declensions and conjugations of the search term bank, e.g., banking, banked, banks, banks', and bank's. Ispell dictionaries usually recognize a limited set of words, so they should be followed by another broader dictionary; for example, a Snowball dictionary, which recognizes everything.

Select *simple* to select the simple template. The simple dictionary template operates by converting the input token to lower case and checking it against a file of stop words. If it is found in the file then an empty array is returned, causing the token to be discarded. If not, the lower-cased form of the word is returned as the normalized lexeme.

Alternatively, the dictionary can be configured to report non-stopwords as unrecognized, allowing them to be passed on to the next dictionary in the list.

Select *snowbal* to select the Snowball template. The Snowball dictionary template is based on a project by Martin Porter, inventor of the popular Porter's stemming algorithm for the English language.

Snowball now provides stemming algorithms for many languages (see the Snowball site for more information). Each algorithm understands how to reduce common variant forms of words to a base, or stem, spelling within its language. A Snowball dictionary recognizes everything, whether or not it is able to simplify the word, so it should be placed at the end of the dictionary list. It is useless to have it before any other dictionary because a token will never pass through it to the next dictionary.

Select *synonym* to select the synonym template. This dictionary template is used to create dictionaries that replace a word with a

Create - FTS Dictionary

General Definition Options SQL

Option		
	Option	Value
	data_option	data_value

Cancel Reset Save

synonym. Phrases are not supported (use the thesaurus template (Section 12.6.4) for that). A synonym dictionary can be used to overcome linguistic problems, for example, to prevent an English stemmer dictionary from reducing the word Paris to pari.

Select *thesaurus* to select the thesaurus template. A thesaurus dictionary replaces all non-preferred terms by one preferred term and, optionally, preserves the original terms for indexing as well.

PostgreSQL's current implementation of the thesaurus dictionary is an extension of the synonym dictionary with added phrase support.

Click the *Options* tab to continue.

Use the fields in the *Options* tab to provide template-specific options.

Click the *Add* icon (+) to add an option clause:

Specify the name of an option in the *Option* field

Provide a value for the option in the *Value* field.

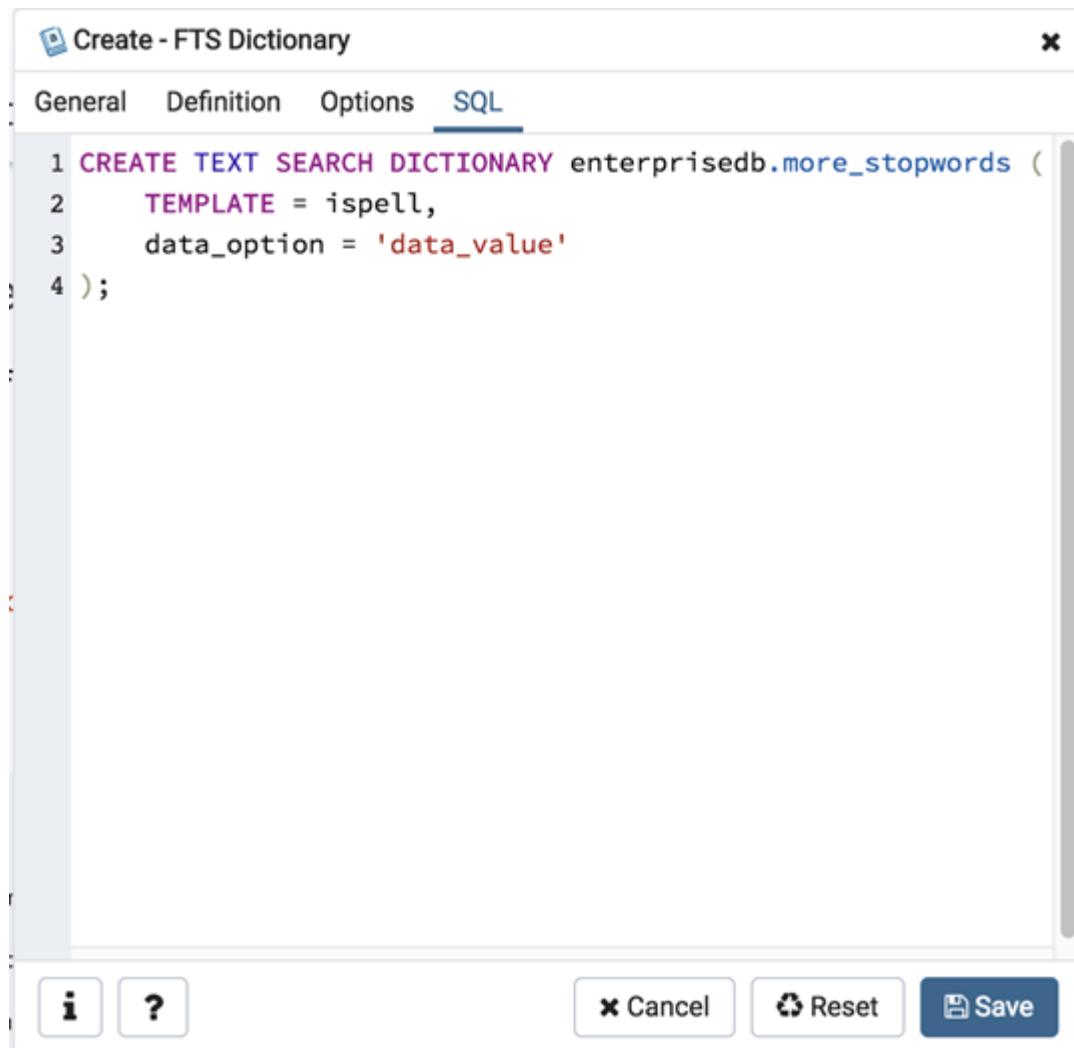
Click the *Add* icon (+) to specify each additional option/value pair; to discard an option, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

Your entries in the *FTS Dictionary* dialog generate a generate a SQL command. Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example 

The following is an example of the sql command generated by user selections in the *FTS Dictionary* dialog:



The example shown demonstrates creating a custom dictionary named *more_stopwords* which is based on the simple template and is configured to use standard English.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

FTS Parser Dialog

Use the FTS Parser dialog to create a new text search parser. A text search parser defines a method for splitting a text string into tokens and assigning types (categories) to the tokens.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Cast Dialog](#)

[FAQ](#)

[Collation Dialog](#)

[Domain Dialog](#)

[□](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)

[Latest Docs](#)

[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[□](#)

[Foreign Server Dialog](#)

[Foreign Table Dialog](#)

[Ask a](#)

[FTS Configuration Dialog](#)

[Question](#)

[FTS Dictionary Dialog](#)

[FTS Parser Dialog](#)

[□](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[Report a Bug](#)

[Language Dialog](#)

[Materialized View Dialog](#)

[□](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Screenshots](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

[FTS Parser Dialog](#) 

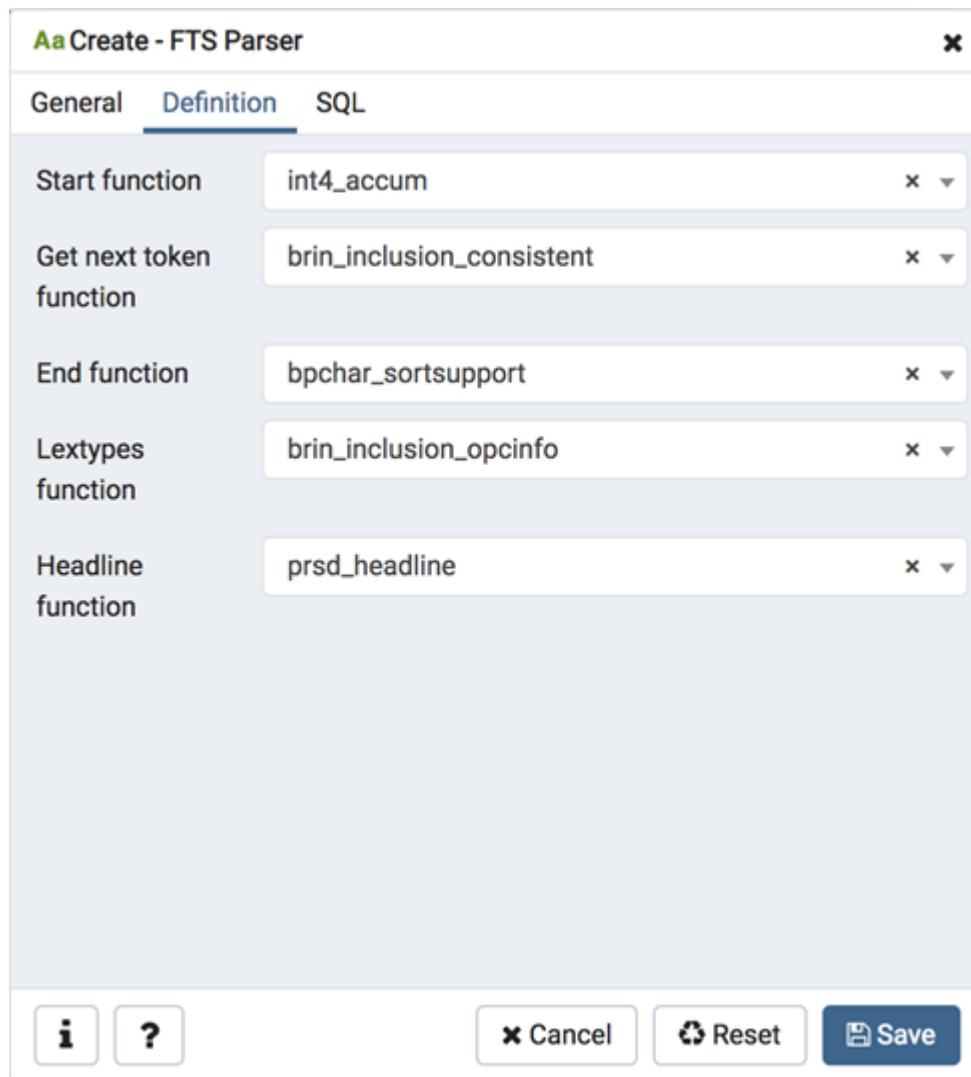
Use the *FTS Parser* dialog to create a new text search parser. A text search parser defines a method for splitting a text string into tokens and assigning types (categories) to the tokens.

Aa Create - FTS Parser x

General Definition SQL

Name	data_parser
Schema	public x ▾
Comment	

i ? ✖ Cancel ↻ Reset 💾 Save



The *FTS Parser* dialog organizes the development of a text search parser through the following dialog tabs: *General*, and *Definition*. The *SQL* tab displays the SQL code generated by dialog selections.

Use the fields in the *General* tab to identify a text search parser: Use the *Name* field to add a descriptive name for the parser. The name will be displayed in the *pgAdmin* tree control.

Select the name of the schema in which the parser will reside from the drop-down listbox in the *Schema* field.

Store notes about the domain in the *Comment* field.

Click the *Definition* tab to continue.

Use the fields in the *Definition* tab to define parameters:

Aa Create - FTS Parser x

General Definition **SQL**

```
1 CREATE TEXT SEARCH PARSER public.data_parser (
2     START = int4_accum,
3     GETTOKEN = brin_inclusion_consistent,
4     END = bpchar_sortsupport,
5     LEXTYPES = brin_inclusion_opcinfo,
6     HEADLINE = prsd_headline);
```

i ? x Cancel Reset Save

Use the drop-down listbox next to *Start function* to select the name of the function that will initialize the parser.

Use the drop-down listbox next to *Get next token function* to select the name of the function that will return the next token.

Use the drop-down listbox next to *End function* to select the name of the function that is called when the parser is finished.

Use the drop-down listbox next to *Lextypes function* to select the name of the lextypes function for the parser. The lextypes function returns an array that contains the id, alias, and a description of the tokens used by the parser.

Use the drop-down listbox next to *Headline function* to select the name of the headline function for the parser. The headline function returns an excerpt from the document in which the terms of the query are highlighted.

Click the *SQL* tab to continue.

Your entries in the *FTS Parser* dialog generate a generate a SQL command. Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [FTS Template Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

FTS Template Dialog

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[FTS Template Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)

[□](#)

[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)

[□](#)

[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)

[□](#)

[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

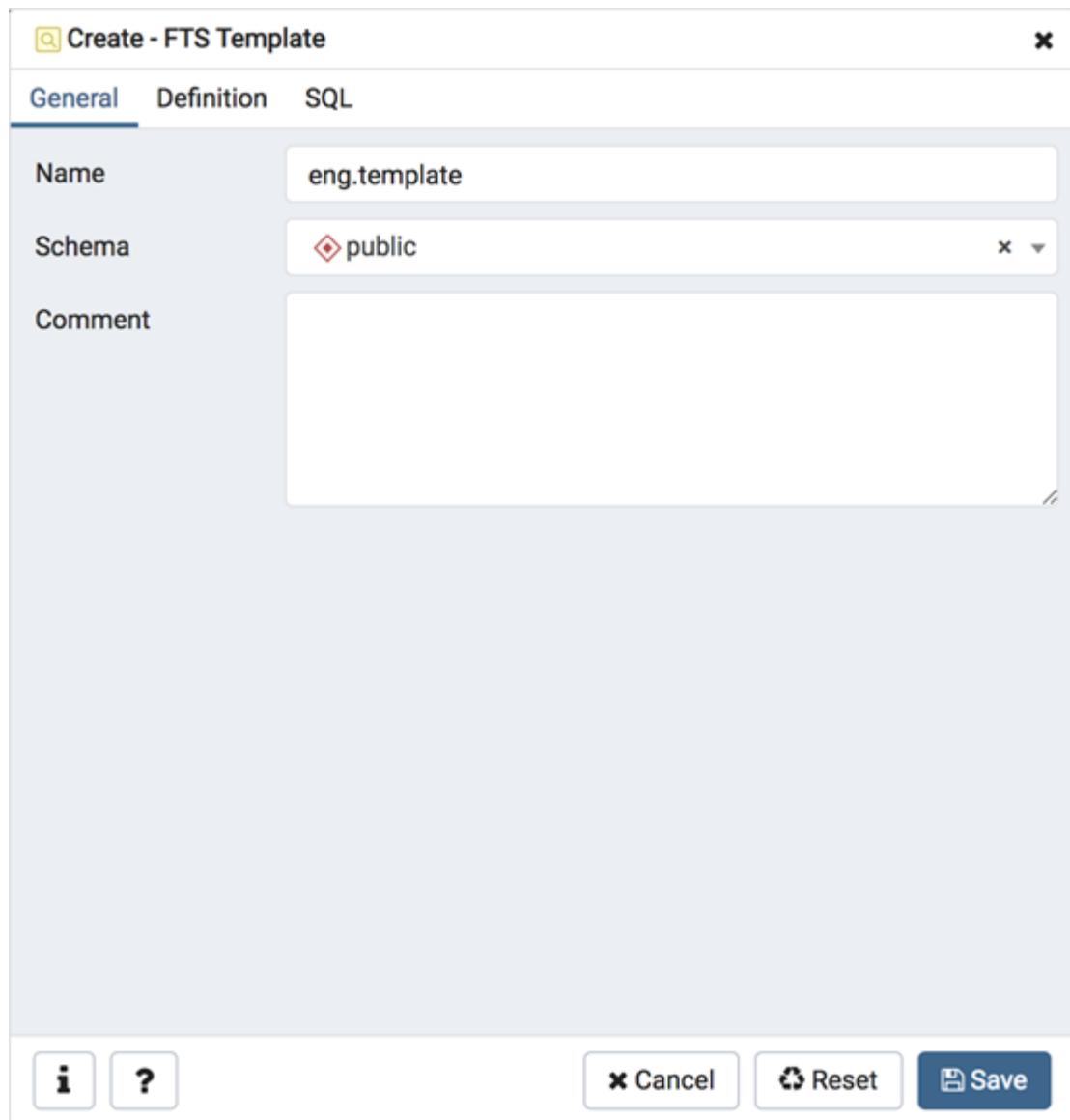
[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

FTS Template Dialog



Use the *FTS Template* dialog to create a new text search template. A text search template defines the functions that implement text search dictionaries.

The *FTS Template* dialog organizes the development of a text search Template through the following dialog tabs: *General*, and

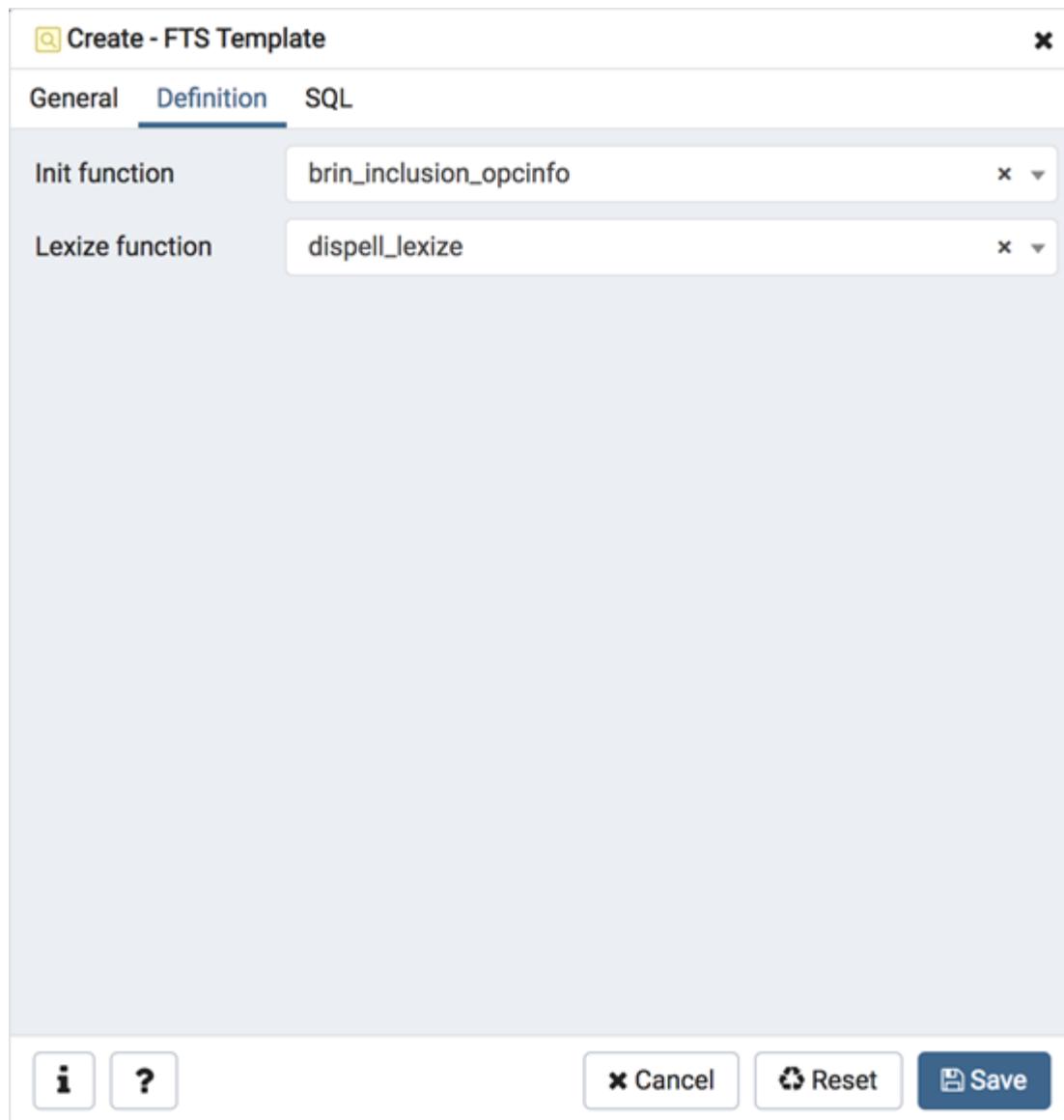
Definition. The **SQL** tab displays the SQL code generated by dialog selections.

Use the fields in the **General** tab to identify a template: Use the **Name** field to add a descriptive name for the template. The name will be displayed in the pgAdmin tree control.

Select the name of the schema in which the template will reside from the drop-down listbox in the **Schema** field.

Store notes about the template in the **Comment** field.

Click the **Definition** tab to continue.



Use the fields in the *Definition* tab to define function parameters: Use the drop-down listbox next to *Init function* to select the name of the init function for the template. The init function is optional.

Use the drop-down listbox next to *Lexize function* to select the name of the lexize function for the template. The lexize function is required.

Click the *SQL* tab to continue.

Your entries in the *FTS Template* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or

switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *FTS Template* dialog:



The screenshot shows a dialog window titled "Create - FTS Template". The "SQL" tab is selected. The main area contains the following SQL code:

```
1 CREATE TEXT SEARCH TEMPLATE public."eng.template" (
2     INIT = brin_inclusion_opcinfo,      LEXIZE = disspell_lexize
3 );
```

At the bottom of the dialog, there are buttons for "Info" (i), "Help" (?), "Cancel" (with an X icon), "Reset" (with a circular arrow icon), and "Save" (with a disk icon).

The example shown demonstrates creating a fts template named *ru_template* that uses the ispel dictionary.

Click the *Info* button (i) to access online help.

Click the **Save** button to save work.

Click the **Cancel** button to exit without saving work.

Click the **Reset** button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Function Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Function Dialog 

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Function Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)



[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)



[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

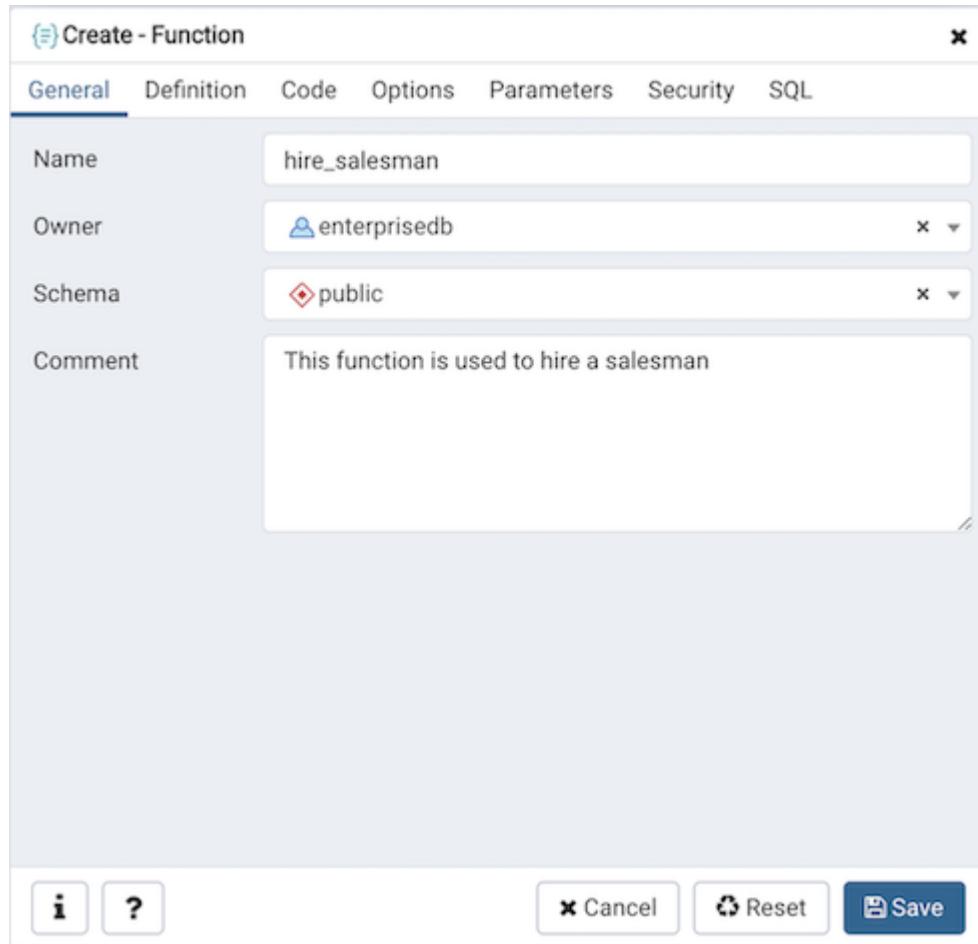
[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

Function Dialog



The screenshot shows the 'Create - Function' dialog box. The 'General' tab is selected. The 'Name' field contains 'hire_salesman'. The 'Owner' field shows 'enterprisedb'. The 'Schema' field shows 'public'. The 'Comment' field contains the text 'This function is used to hire a salesman'. At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save'.

Use the *Function* dialog to define a function. If you drop and then recreate a function, the new function is not the same entity as the old; you must drop existing rules, views, triggers, etc. that refer to the old function.

The *Function* dialog organizes the development of a function through the following dialog tabs: *General*, *Definition*, *Code*, *Options*, *Parameters*, and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

Use the fields in the *General* tab to identify a function: Use the *Name* field to add a descriptive name for the function. The name will be

displayed in the *pgAdmin* tree control.

Use the drop-down listbox next to *Owner* to select the name of the role that will own the function.

Use the drop-down listbox next to *Schema* to select the schema in which the function will be created.

Store notes about the function in the *Comment* field.

Click the *Definition* tab to continue.

The screenshot shows the 'Create - Function' dialog box in pgAdmin. The 'Definition' tab is active. The 'Return type' is set to 'numeric' and 'Language' to 'plpgsql'. The 'Arguments' section contains three entries:

Data type	Mode	Argument name	Default
character varying		p_ename	
numeric		p_sal	
numeric		p_comm	

At the bottom are buttons for 'Cancel', 'Reset', and 'Save'.

Use the fields in the *Definition* tab to define the function: Use the drop-down listbox next to *Return type* to select the data type returned by the function, if any.

Use the drop-down listbox next to *Language* to select the implementation language. The default is *sql*.

Use the fields in the *Arguments* to define an argument. Click the *Add* icon (+) to set parameters and values for the argument:

Use the drop-down listbox in the *Data type* field to select a data type.

Use the drop-down listbox in the *Mode* field to select a mode.

Select *I/N* for an input parameter; select *OUT* for an output parameter; select *INOUT* for both an input and an output parameter; or, select *VARIADIC* to specify a VARIADIC

parameter.

Provide a name for the argument in the *Argument Name* field.

Specify a default value for the argument in the *Default Value* field.

Click the *Add* icon (+) to define another argument; to discard an argument, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *Code* tab to continue.

{(Create - Function X

General Definition **Code** Options Parameters Security SQL

```
1 DECLARE
2     v_empno      NUMERIC(4);
3     v_ename      VARCHAR(10);
4     v_job        VARCHAR(9);
5     v_mgr        NUMERIC(4);
6     v_hiredate   DATE;
7     v_sal        NUMERIC(7,2);
8     v_comm       NUMERIC(7,2);
9     v_deptno    NUMERIC(2);
10 BEGIN
11     v_empno := new_empno();
12     INSERT INTO emp VALUES (v_empno, p_ename, 'SALESMAN', 7698,
13                             CURRENT_DATE, p_sal, p_comm, 30);
14     SELECT INTO
15         v_empno, v_ename, v_job, v_mgr, v_hiredate, v_sal, v_c
16         empno, ename, job, mgr, hiredate, sal, comm, deptno
17     FROM emp WHERE empno = v_empno;
18     RAISE INFO 'Department : %', v_deptno;
19     RAISE INFO 'Employee No: %', v_empno;
20     RAISE INFO 'Name      : %', v_ename;
21     RAISE INFO 'Salary    : %', v_sal;
```

i ? Cancel Reset Save

Create - Function

General Definition Code **Options** Parameters Security SQL

Volatility	Select an item...
Returns a set?	<input checked="" type="checkbox"/> No
Strict?	<input checked="" type="checkbox"/> No
Security of definer?	<input checked="" type="checkbox"/> No
Window?	<input checked="" type="checkbox"/> No
Parallel	Select an item...
Estimated cost	
Estimated rows	
Leak proof?	<input checked="" type="checkbox"/> No
Support function	

i **?** **Cancel** **Reset** **Save**

Use the *Code* field to write the code that will execute when the function is called.

Click the *Options* tab to continue.

Use the fields in the *Options* tab to describe or modify the action of the function:

Use the drop-down listbox next to *Volatility* to select one of the following. *VOLATILE* is the default value.

VOLATILE indicates that the function value can change even within a single table scan, so no optimizations can be made.

STABLE indicates that the function cannot modify the database, and that within a single table scan it will consistently return the same

result for the same argument values.

IMMUTABLE indicates that the function cannot modify the database and always returns the same result when given the same argument values.

Move the *Returns a Set?* switch to indicate if the function returns a set that includes multiple rows. The default is *No*.

Move the *Strict?* switch to indicate if the function always returns NULL whenever any of its arguments are NULL. If *Yes*, the function is not executed when there are NULL arguments; instead a NULL result is assumed automatically. The default is *No*.

Move the *Security of definer?* switch to specify that the function is to be executed with the privileges of the user that created it. The default is *No*.

Move the *Window?* switch to indicate that the function is a window function rather than a plain function. The default is *No*. This is currently only useful for functions written in C. The WINDOW

attribute cannot be changed when replacing an existing function definition. For more information about the CREATE FUNCTION command, see the PostgreSQL core documentation available at:

<https://www.postgresql.org/docs/current/functions-window.html>

Use the *Estimated cost* field to specify a positive number representing the estimated execution cost for the function, in units of `cpu_operator_cost`. If the function returns a set, this is the cost per returned row.

Use the *Estimated rows* field to specify a positive number giving the estimated number of rows that the query planner should expect the function to return. This is only allowed when the function is declared to return a set. The default assumption is 1000 rows.

Move the *Leak proof?* switch to indicate whether the function has side effects. The default is *No*. This option can only be set by the superuser.

Use the *Support function* field to specify a planner support function to use for the function.

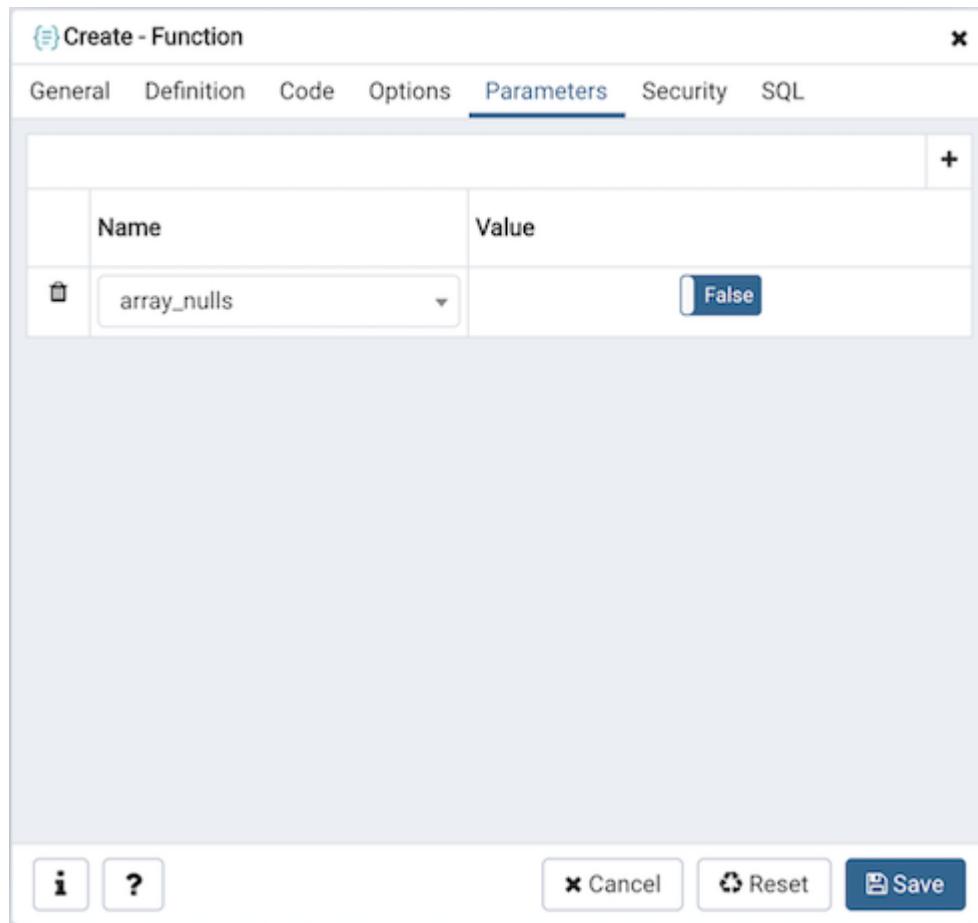
Click the *Parameters* tab to continue.

Create - Function

General Definition Code Options **Parameters** Security SQL

Name	Value
array_nulls	False

i **?** **Cancel** **Reset** **Save**



Create - Function

General Definition Code Options Parameters **Security** SQL

Privileges

Grantee	Privileges	Grantor
<input type="button" value="Delete"/>  enterpriseedb	<input checked="" type="checkbox"/> EXECUTE <input checked="" type="checkbox"/> WITH GRANT OPTION	 enterpriseedb

Security labels

Provider	Security label

Buttons:    Cancel  Reset  Save

Use the fields in the *Parameters* tab to specify settings that will be applied when the function is invoked. Click the *Add* icon (+) to add a *Name/ Value* field in the table.

Use the drop-down listbox in the *Name* column in the *Parameters* panel to select a parameter.

Use the *Value* field to specify the value that will be associated with the selected variable. This field is context-sensitive.

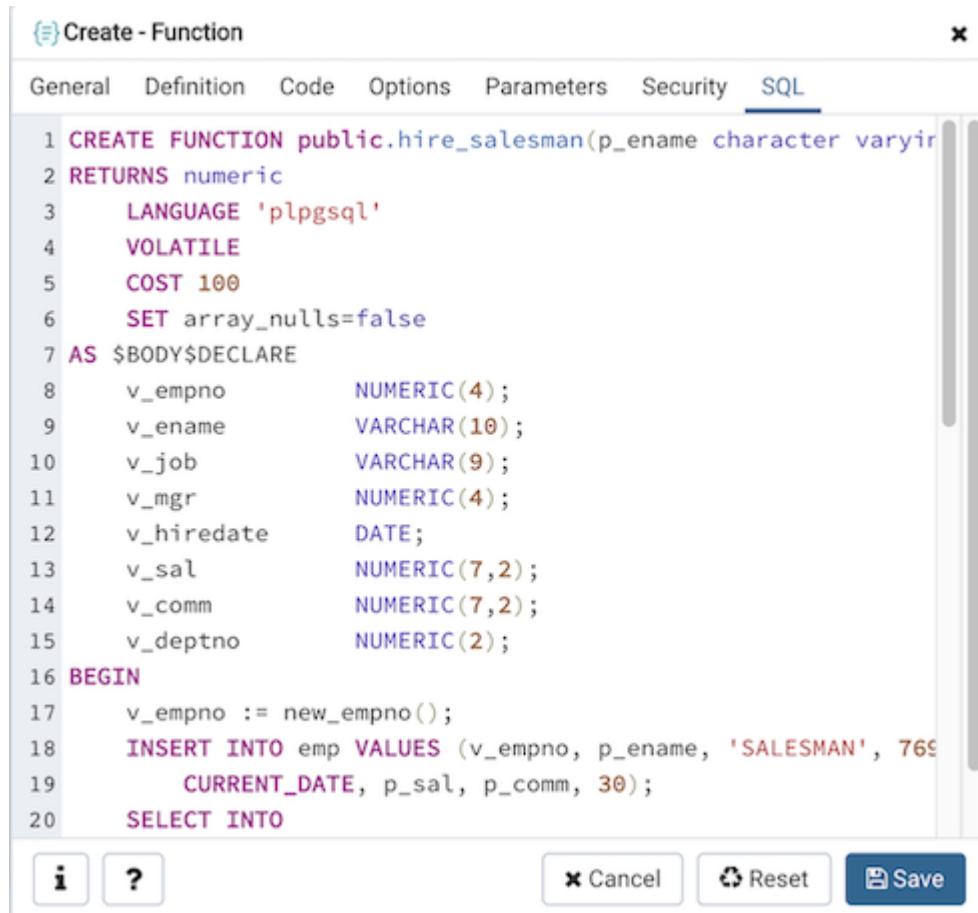
Click the *Security* tab to continue.

Use the *Security* tab to assign privileges and define security labels.

Use the *Privileges* panel to assign usage privileges for the function to a role.

Select the name of the role from the drop-down listbox in the *Grantee* field.

Click inside the *Privileges* field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.



The screenshot shows a software interface for creating a database function. The title bar says 'Create - Function'. Below it is a tab bar with 'General', 'Definition', 'Code', 'Options', 'Parameters', 'Security', and 'SQL'. The 'SQL' tab is selected. The main area contains the following SQL code:

```
1 CREATE FUNCTION public.hire_salesman(p_ename character varying
2 RETURNS numeric
3   LANGUAGE 'plpgsql'
4   VOLATILE
5   COST 100
6   SET array_nulls=false
7 AS $BODY$DECLARE
8   v_empno      NUMERIC(4);
9   v_ename      VARCHAR(10);
10  v_job        VARCHAR(9);
11  v_mgr        NUMERIC(4);
12  v_hiredate   DATE;
13  v_sal        NUMERIC(7,2);
14  v_comm       NUMERIC(7,2);
15  v_deptno     NUMERIC(2);
16 BEGIN
17   v_empno := new_empno();
18   INSERT INTO emp VALUES (v_empno, p_ename, 'SALESMAN', 7654,
19                         CURRENT_DATE, p_sal, p_comm, 30);
20   SELECT INTO
```

At the bottom are buttons for 'Cancel', 'Reset', and 'Save'.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.

Click the *Add* icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Use the *Security Labels* panel to define security labels applied to the function. Click the *Add* icon (+) to add each security label selection: Specify a security label provider in the *Provider* field. The named

provider must be loaded and must consent to the proposed labeling operation.

Specify a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL

places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the *Add* icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

Your entries in the *Function* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by selections made in the *Function* dialog:

The example demonstrates creating an *plpgsql* function named *hire_salesmen*. The function have three columns (*p_ename*, *p_sal* and *p_comm*).

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 documentation](#) » [Management Basics](#) »

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Grant Wizard](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
 - [Add Named Restore Point Dialog](#)
 - [Change Password Dialog](#)
 - [Grant Wizard](#)
 - [Import/Export Data Dialog](#)
 - [Maintenance Dialog](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Grant Wizard

The *Grant Wizard* tool is a graphical interface that allows you to manage the privileges of one or more database objects in a point-and-click environment. A search box, dropdown lists, and checkboxes facilitate quick selections of database objects, roles and privileges.

The wizard organizes privilege management through a sequence of windows: *Object Selection (step 1 of 3)*, *Privileges Selection (step 2 of 3)* and *Final (Review Selection) (step 3 of 3)*. The *Final (Review Selection)* window displays the SQL code generated by wizard selections.

To launch the *Grant Wizard* tool, select a database object in the pgAdmin tree control, then navigate through *Tools* on the menu bar to click on the *Grant Wizard* option.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#)
[Development](#)

[pgAdmin 4 4.19 documentation](#) » [Management Basics](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Grant Wizard](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Management Basics](#)

[Add Named Restore Point Dialog](#)

[Latest Docs](#)

[Change Password Dialog](#)

Grant Wizard



[Import/Export Data Dialog](#)

[Maintenance Dialog](#)

[Ask a](#)

[Question](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)



[pgAdmin Project Contributions](#)

[Report a Bug](#)

[Release Notes](#)

[Licence](#)



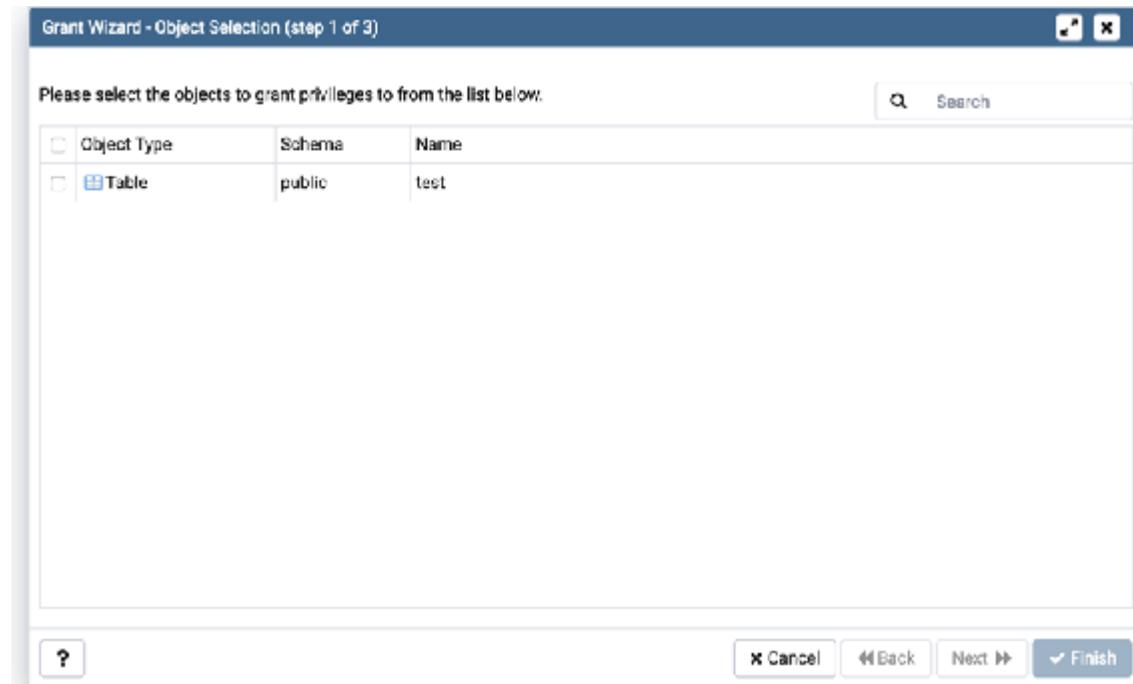
Grant Wizard

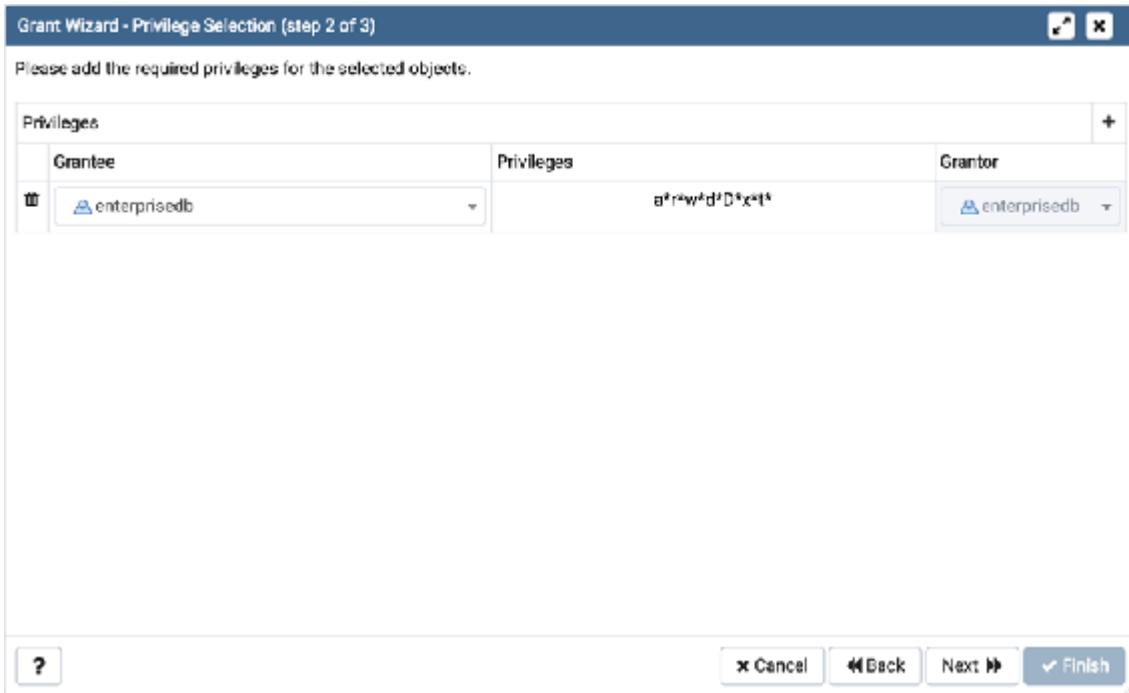
Screenshots

The *Grant Wizard* tool is a graphical interface that allows you to manage the privileges of one or more database objects in a point-and-click environment. A search box, dropdown lists, and checkboxes facilitate quick selections of database objects, roles and privileges.

The wizard organizes privilege management through a sequence of windows: *Object Selection (step 1 of 3)*, *Privileges Selection (step 2 of 3)* and *Final (Review Selection) (step 3 of 3)*. The *Final (Review Selection)* window displays the SQL code generated by wizard selections.

To launch the *Grant Wizard* tool, select a database object in the *pgAdmin* tree control, then navigate through *Tools* on the menu bar to click on the *Grant Wizard* option.





Use the fields in the *Object Selection (step 1 of 3)* window to select the object or objects on which you are modifying privileges. Use the *Search by object type or name* field to locate a database object, or use the scroll bar to scroll through the list of all accessible objects.

Each row in the table lists object identifiers; check the checkbox in the left column to include an object as a target of the Grant Wizard.

The table displays:

The object type in the *Object Type* field

The schema in which the object resides in the *Schema* field The object name in the *Name* field.

Click the *Next* button to continue, or the *Cancel* button to close the wizard without modifying privileges.

Use the fields in the *Privileges Selection (step 2 of 3)* window to grant privileges. If you grant a privilege WITH GRANT OPTION, the Grantee will have the right to grant privileges on the object to others. If WITH GRANT

OPTION is subsequently revoked, any role who received access to that object from that Grantee (directly or through a chain of grants) will lose their privileges on the object.

Click the *Add* icon (+) to assign a set of privileges.



Select the name of the role from the drop-down listbox in the *Grantee* field.

Click inside the *Privileges* field. Check the boxes to the left of one or more privileges to grant the selected privileges to the specified user.

If privileges have previously been granted on a database object, unchecking a privilege for a group or user will result in revoking that privilege.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.

Click the *Add* icon (+) to assign a set of privileges to another role; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the *Delete Row* dialog.

For more information about granting privileges on database objects, see the [PostgreSQL core documentation](#).

Click the *Next* button to continue, the *Back* button to select or deselect additional database objects, or the *Cancel* button to close the wizard without modifying privileges.

Your entries in the *Grant Wizard* tool generate a SQL command; you can review the command in the *Final (Review Selection) (step 3 of 3)* window (see an example below).

Example

The following is an example of the sql command generated by user selections in the *Grant Wizard* tool:

The commands displayed assign a role named *Bob* *INSERT* and *UPDATE*

privileges *WITH GRANT OPTION* on the *sales_meetings* and the *sales_territories* tables.

Click the *Back* button to select or deselect additional database objects, roles and privileges.

Click the *Cancel* button to exit without saving work.

Click the *Finish* button to save selections and exit the wizard.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Management Basics](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
 - [Add Named Restore Point Dialog](#)
 - [Change Password Dialog](#)
 - [Grant Wizard](#)
 - [Import/Export Data Dialog](#)
 - [Maintenance Dialog](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Import/Export Data Dialog

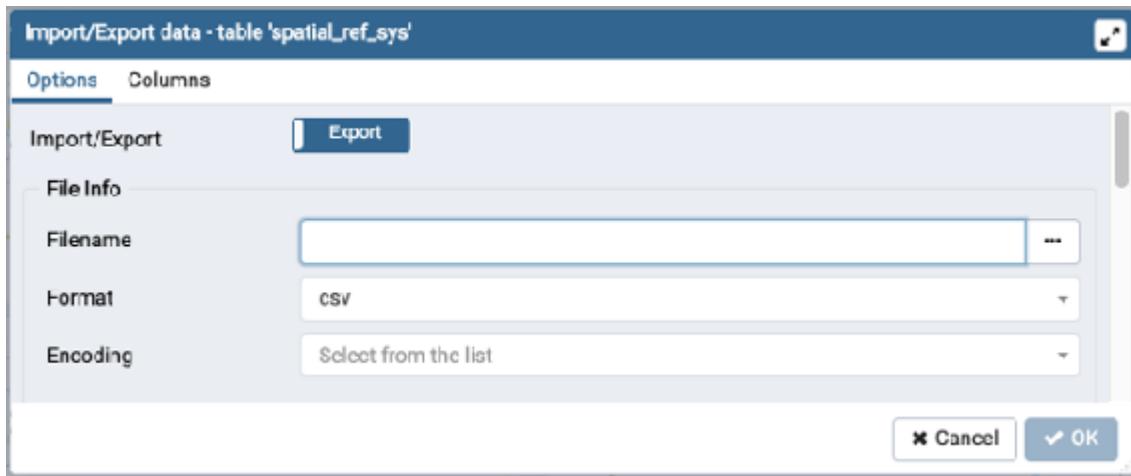
Use the *Import/Export* data dialog to copy data from a table to a file, or copy data from a file into a table.

The *Import/Export* data dialog organizes the import/export of data through the *Options* and *Columns* tabs.



Use the fields in the *Options* tab to specify import and export preferences:

- Move the *Import/Export* switch to the *Import* position to specify that the server should import data to a table from a file. The default is *Export*.
- Use the fields in the *File Info* field box to specify information about the source or target file:



Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Management Basics](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

Managing Database Objects



Creating or Modifying a Table

FAQ

Management Basics

Add Named Restore Point Dialog



Change Password Dialog

Grant Wizard

Latest Docs

Import/Export Data Dialog

Maintenance Dialog



Backup and Restore

Developer Tools

Ask a

pgAgent

Question

pgAdmin Project Contributions

Release Notes



[Licence](#)

[Report a Bug](#)

Import/Export Data Dialog



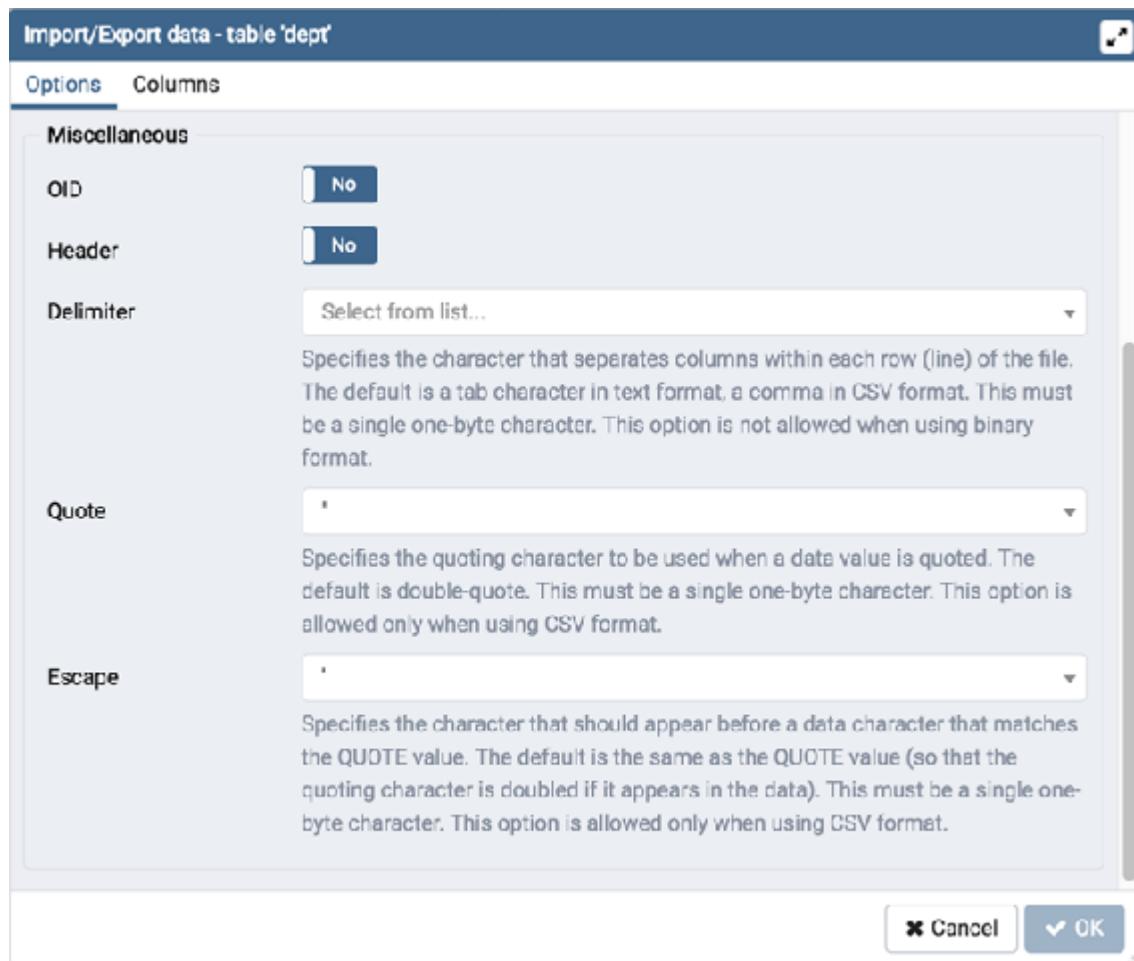
Use the *Import/Export data* dialog to copy data from a table to a file, or copy data from a file into a table.

[Screenshots](#)

The *Import/Export data* dialog organizes the import/export of data through the *Options* and *Columns* tabs.

Use the fields in the *Options* tab to specify import and export preferences: Move the *Import/Export* switch to the *Import* position to specify that the server should import data to a table from a file. The default is *Export*.

Use the fields in the *File Info* field box to specify information about the source or target file:



Enter the name of the source or target file in the *Filename* field.

Optional y, select the *Browser* icon (el ipsis) to the right to navigate into a directory and select a file.

Use the drop-down listbox in the *Format* field to specify the file type.
Select:

binary for a .bin file.

csv for a .csv file.

text for a .txt file.

Use the drop-down listbox in the *Encoding* field to specify the type of character encoding.

Use the fields in the *Miscellaneous* field box to specify additional information:

Move the *O/D* switch to the *Yes* position to include the *O/D*

column. The *O/D* is a system-assigned value that may not be modified. The default is *No*.

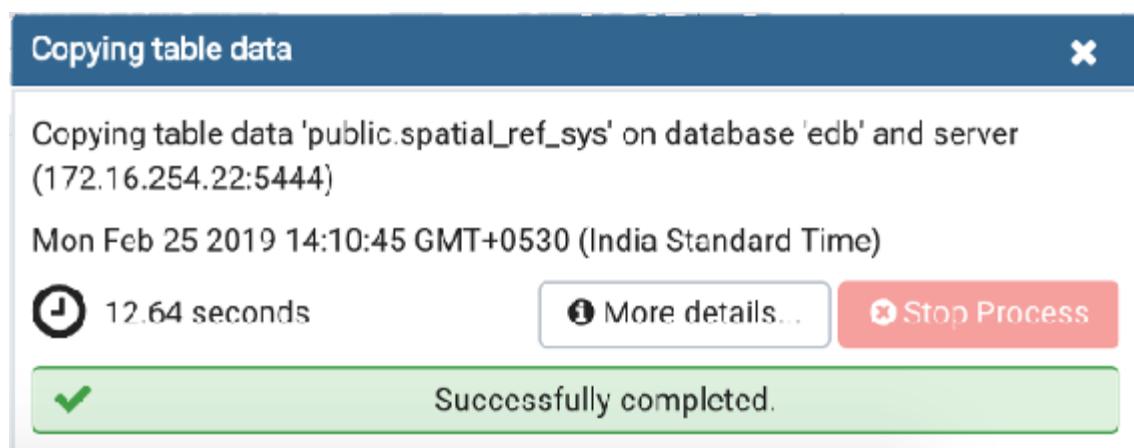
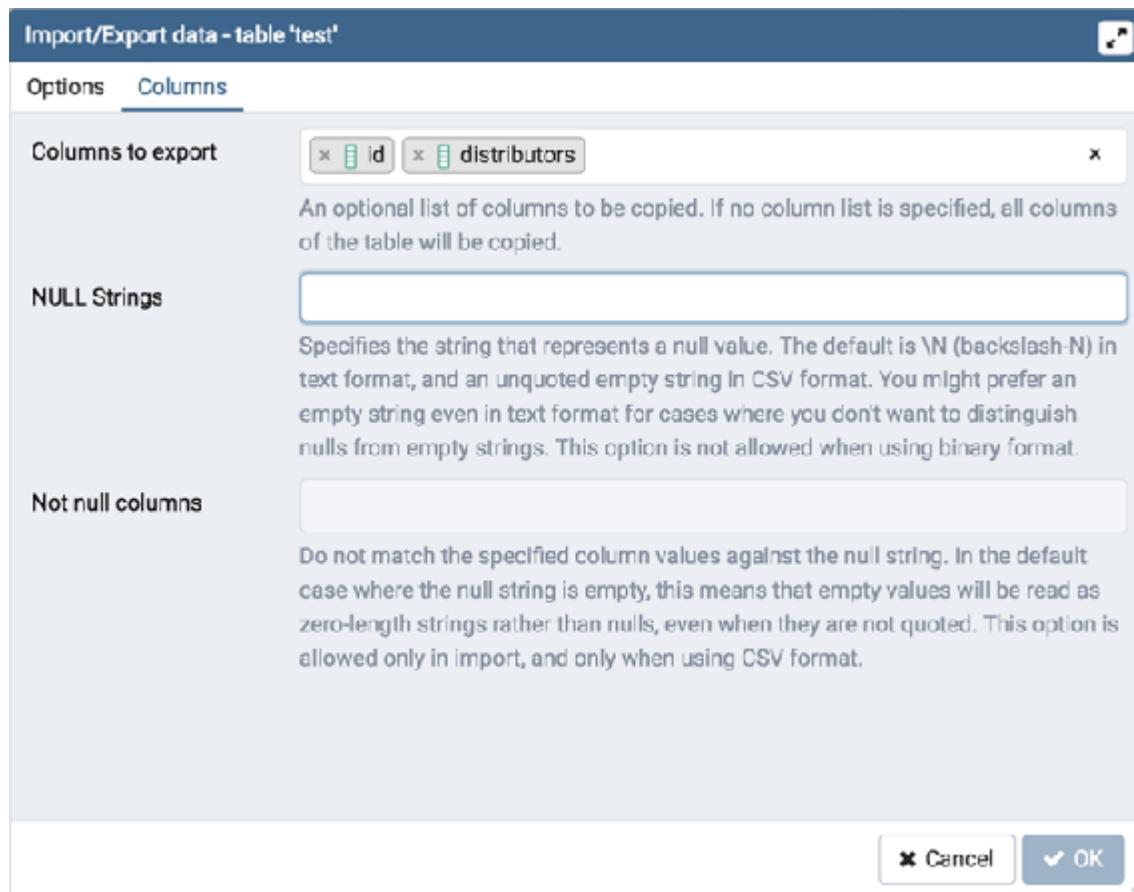
Move the *Header* switch to the *Yes* position to include the table header with the data rows. If you include the table header, the first row of the file will contain the column names.

If you are exporting data, specify the delimiter that will separate the columns within the target file in the *Delimiter* field. The separating character can be a colon, semicolon, a vertical bar, or a tab.

Specify a quoting character used in the *Quote* field. Quoting can be applied to string columns only (i.e. numeric columns will not be quoted) or all columns regardless of data type. The character used for quoting can be a single quote or a double quote.

Specify a character that should appear before a data character that matches the *QUOTE* value in the *Escape* field.

Click the *Columns* tab to continue.



Use the fields in the *Columns* tab to select the columns that will be imported or exported:

Click inside the *Columns to export/import* field to deselect one or more columns from the drop-down listbox. To delete a selection, click

the x to the left of the column name. Click an empty spot inside the field to access the drop-down list.

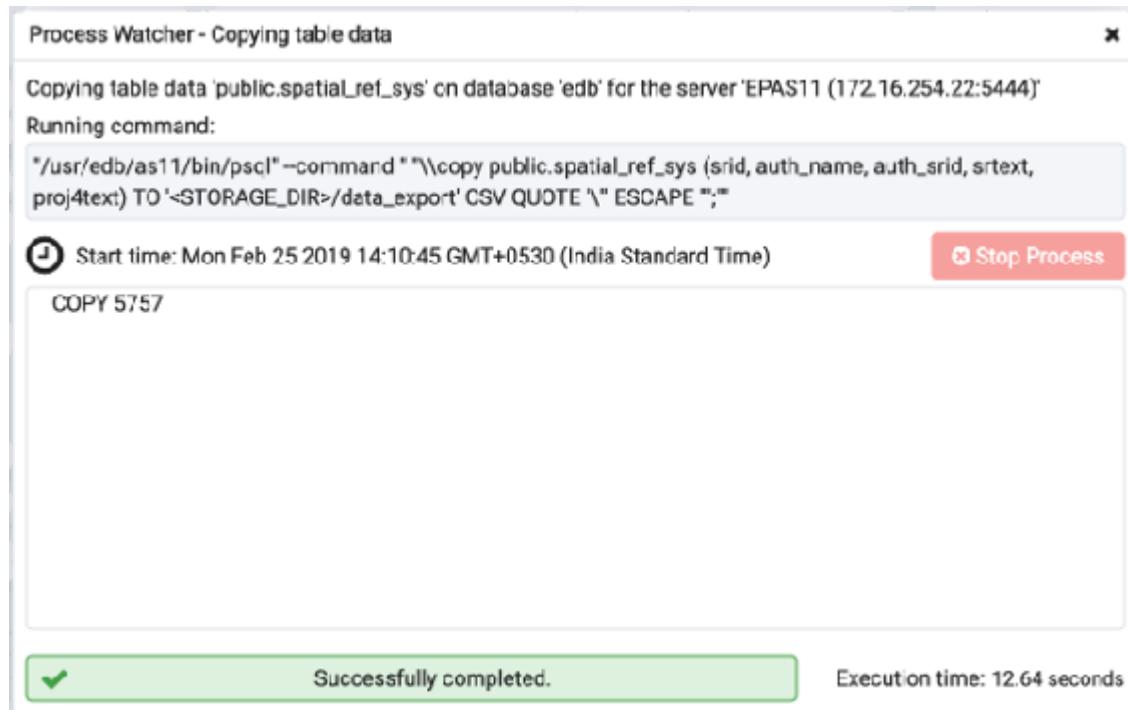
Use the *NULL Strings* field to specify a string that will represent a null value within the source or target file.

If enabled, click inside the *Not null columns* field to select one or more columns that will not be checked for a NULL value. To delete a column, click the x to the left of the column name.

After completing the *Import/Export data* dialog, click the *OK* button to perform the import or export. pgAdmin will inform you when the background process completes:

Use the Stop Process button to stop the Import/Export process.

Use the *Click here for details* link on the notification to open the *Process Watcher* and review detailed information about the execution of the command that performed the import or export:



Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | Development

[pgAdmin 4 documentation](#) > [Connecting To A Server](#) > [previous](#) [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Import/Export Servers](#)
 - [Exporting Servers](#)
 - [Importing Servers](#)
 - [JSON format](#)
- [Getting Started](#)
- [Connecting To A Server](#)
 - [Server Group Dialog](#)
 - [Server Dialog](#)
 - [Master Password](#)
 - [Connect to Server](#)
 - [Connection Error](#)
 - [Import/Export Servers](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Import/Export Servers

Server definitions (and their groups) can be exported to a JSON file and re-imported to the same or a different system to enable easy pre-configuration of pgAdmin. The `setup.py` script is used for this purpose.

Note

To export or import servers, you must use the Python interpreter that is normally used to run pgAdmin to ensure that the required Python packages are available. In most packages, this can be found in the Python Virtual Environment that can be found in the installation directory. When using platform-native packages, the system installation of Python may be the one used by pgAdmin.

Exporting Servers

To export the servers defined in an installation, simply invoke `setup.py` with the `--dump-servers` command line option, followed by the name (and if required, path) to the desired output file. By default, servers owned

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Connecting To A Server](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



Import/Export Servers

[Download](#)

[Exporting Servers](#)

[Importing Servers](#)

[JSON format](#)



[Getting Started](#)

[FAQ](#)

[Connecting To A Server](#)

[Server Group Dialog](#)



[Server Dialog](#)

[Master Password](#)

[Latest Docs](#)

[Connect to Server](#)

[Connection Error](#)



Import/Export Servers

[Ask a](#)

[Managing Cluster Objects](#)

[Question](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[Management Basics](#)



[Backup and Restore](#)

[Report a Bug](#)

[Developer Tools](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)



[Release Notes](#)

[Screenshots](#)

[Licence](#)

Import/Export Servers

Server definitions (and their groups) can be exported to a JSON file and re-imported to the same or a different system to enable easy pre-configuration of pgAdmin. The setup.py script is used for this purpose.

Note

To export or import servers, you must use the Python interpreter that is normally used to run pgAdmin to ensure that the required Python packages are available. In most packages, this can be found in the Python Virtual Environment that can be found in the installation directory. When using platform-native packages, the system installation of Python may be the one used by pgAdmin.

Exporting Servers

To export the servers defined in an installation, simply invoke setup.py with the --dump-servers command line option, followed by the name (and if required, path) to the desired output file. By default, servers owned

by the desktop mode user will be dumped (pgadmin4@pgadmin.org by default - see the DESKTOP_USER setting in config.py). This can be overridden with the --user command line option. For example:

```
/path/to/python /path/to/setup.py --dump-servers
```

```
output_file.json
```

or, to specify a non-default user name:

```
/path/to/python /path/to/setup.py --dump-servers  
output_file.json --user user@example.com
```

To export only certain servers, use the --servers option and list one or more server IDs. For example:

```
/path/to/python /path/to/setup.py --dump-servers  
output_file.json --server 1 2 5
```

Importing Servers

To import the servers defined in a JSON file, simply invoke setup.py with the --load-servers command line option, followed by the name (and if required, path) of the JSON file containing the server definitions. Servers will be owned by the desktop mode user (pgadmin4@pgadmin.org) by default - see the DESKTOP_USER setting in config.py). This can be overridden with the --user command line option. For example:

```
/path/to/python /path/to/setup.py --load-servers  
input_file.json
```

or, to specify a non-default user name to own the new servers:

```
/path/to/python /path/to/setup.py --load-servers  
input_file.json --user user@example.com
```

If any Servers are defined with a Server Group that is not already present in the configuration database, the required Group will be created.

JSON format

The JSON file format used when importing or exporting servers is quite straightforward and simply contains a list of servers, with a number of attributes. The following attributes are required to be present in every server definition: Name, Group, Port, Username, SSLMode, MaintenanceDB and one of Host, HostAddr or Service.

Password fields cannot be imported or exported.

The following example shows both a minimally defined and a fully defined server:

```
{  
  "Servers": {  
    "1": {  
      "Name": "Minimally Defined Server",  
      "Group": "Server Group 1",  
      "Port": 5432,  
      "Username": "postgres",  
      "Host": "localhost",  
      "SSLMode": "prefer",  
      "MaintenanceDB": "postgres"  
    },  
    "2": {  
      "Name": "Fully Defined Server",  
      "Group": "Server Group 2",  
      "Host": "127.0.0.1",  
      "Port": 5433,  
      "Username": "postgres",  
      "Password": "mysecretpassword",  
      "SSLMode": "require",  
      "MaintenanceDB": "postgres"  
    }  
  }  
}
```

```
"Host": "host.domain.com",
"HostAddr": "192.168.1.2",
"Port": 5432,
"MaintenanceDB": "postgres",
"Username": "postgres",
"Role": "my_role_name",
"SSLMode": "require",
"Comment": "This server has every option configured in the JSON",
"DBRestriction": "live_db test_db",
"PassFile": "/path/to/pgpassfile",
"SSLCert": "/path/to/sslcert.crt",
"SSLKey": "/path/to/sslcert.key",
"SSLRootCert": "/path/to/sslroot.crt",
"SSLCrl": "/path/to/sslcrl.crl",
"SSLCompression": 1,
"BGColor": "#ff9900",
"FGColor": "#000000",
"Service": "postgresql-10",
"Timeout": 60,
"UseSSHTunnel": 1,
```

"TunnelHost": "192.168.1.253",

"TunnelPort": 22,

"TunnelUsername": "username",

"TunnelAuthentication": 0

}

}

}

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) > [Creation or Modifying a Table](#) > [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Index Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
 - [Check Dialog](#)
 - [Column Dialog](#)
 - [Compound Trigger Dialog](#)
 - [Exclusion Constraint Dialog](#)
 - [Foreign key Dialog](#)
 - [Index Dialog](#)
 - [Primary key Dialog](#)
 - [Rule Dialog](#)
 - [Table Dialog](#)
 - [Trigger Dialog](#)
 - [Unique Constraint Dialog](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Index Dialog

Use the Index dialog to create an index on a specified table or materialized view.

The Index dialog organizes the development of an index through the following dialog tabs: General and Definition. The SQL tab displays the SQL code generated by dialog selections.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Creating or Modifying a Table](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Index Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[□](#)

[Check Dialog](#)

[Column Dialog](#)

[Latest Docs](#)

[Compound Trigger Dialog](#)

[Exclusion Constraint Dialog](#)

[□](#)

[Foreign key Dialog](#)

[Index Dialog](#)

[Ask a](#)

[Primary key Dialog](#)

[Question](#)

[Rule Dialog](#)

[Table Dialog](#)

[□](#)

[Trigger Dialog](#)

[Report a Bug](#)

[Unique Constraint Dialog](#)

[Management Basics](#)

[Backup and Restore](#)



[Developer Tools](#)

[Screenshots](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Index Dialog

Use the *Index* dialog to create an index on a specified table or materialized view.

The *Index* dialog organizes the development of a index through the following dialog tabs: *General* and *Definition*. The *SQL* tab displays the SQL code generated by dialog selections.

Create - Index

General Definition SQL

Name index_prod

Tablespace pg_default

Comment

i ? Cancel Reset Save

Create - Index

General **Definition** SQL

Access Method	btree				
Fill factor					
Unique?	No				
Clustered?	No				
Concurrent build?	No				
Constraint	1				
Columns					
	Column	Operator class	Sort order	NULLS	Collation
	c_name	varchar_ops	ASC	LAST	pg_catalog."aa_DJ"
Include columns		Select the column(s)			

i **?** **Cancel** **Reset** **Save**

Use the fields in the *General* tab to identify the index: Use the *Name* field to add a descriptive name for the index. The name will be displayed in the pgAdmin tree control.

Use the drop-down listbox next to *Tablespace* to select the tablespace in which the index will reside.

Store notes about the index in the *Comment* field.

Click the *Definition* tab to continue.

Use the fields in the *Definition* tab to define the index: Use the drop-down listbox next to *Access Method* to select an index type:

Select *btree* to create a B-tree index. A B-tree index may improve performance when managing equality and range

queries on data that can be sorted into some ordering (the default).

Select *hash* to create a hash index. A hash index may improve performance when managing simple equality comparisons.

Select *gist* to create a GiST index. A GiST index may improve performance when managing values with more than one key.

Select *gin* to create a GIN index. A GIN index may improve performance when managing two-dimensional geometric data

types and nearest-neighbor searches.

Select *spgist* to create a space-partitioned GiST index. A SPGiST index may improve performance when managing non-balanced data structures.

Select *brin* to create a BRIN index. A BRIN index may improve performance when managing minimum and maximum values and ranges.

Use the *Fil Factor* field to specify a fil factor for the index. The fil factor specifies how ful the selected method wil try to fil each index page.

Move the *Unique?* switch to the Yes position to check for duplicate values in the table when the index is created and when data is added.

The default is No.

Move the *Clustered?* switch to the Yes position to instruct the server to cluster the table.

Move the *Concurrent build?* switch to the Yes position to build the index without taking any locks that prevent concurrent inserts, updates, or deletes on the table.

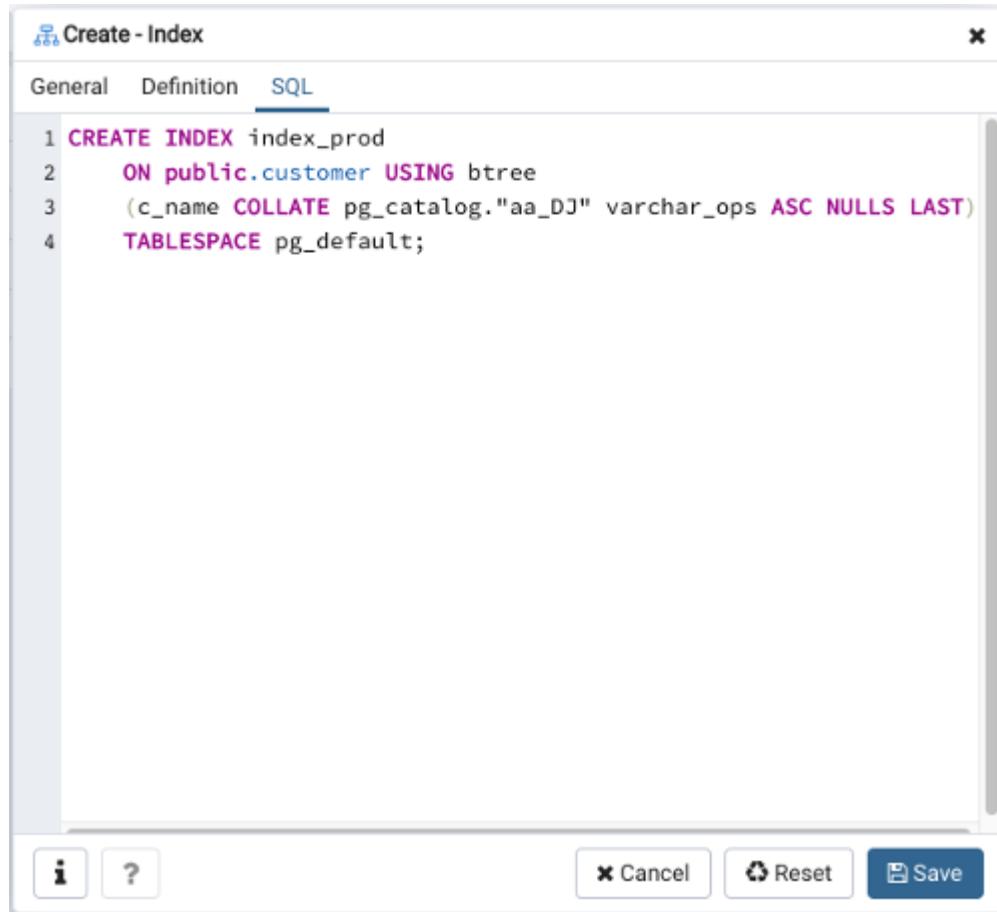
Use the *Constraint* field to provide a constraint expression; a constraint expression limits the entries in the index to those rows that satisfy the constraint.

Use the context-sensitive fields in the *Columns* panel to specify which column(s) the index queries. Click the *Add* icon (+) to add a column: Use the drop-down listbox in *Column* field to select the name of the column from the table.

If enabled, use the drop-down listbox to select an available *Operator class* to specify the type of action performed on the column.

If enabled, move the *Sort order* switch to specify the sort order: Select *ASC* to specify an ascending sort order (the default); Select *DESC* to specify a descending sort order.

If enabled, move the *Nuls* switch to specify the sort order of nulls:



The screenshot shows a software interface for creating an index. The title bar says "Create - Index". Below it, there are three tabs: "General", "Definition", and "SQL", with "SQL" being the active tab. The SQL tab displays the following PostgreSQL CREATE INDEX statement:

```
1 CREATE INDEX index_prod
2   ON public.customer USING btree
3     (c_name COLLATE pg_catalog."aa_DJ" varchar_ops ASC NULLS LAST)
4 TABLESPACE pg_default;
```

At the bottom of the dialog, there are several buttons: an information icon (i), a question mark icon (?), "Cancel" (with a red X), "Reset" (with a circular arrow), and "Save" (highlighted in blue).

Select *First* to specify nulls sort before non-nulls; Select *Last* to specify nulls sort after non-nulls (the default).

Use the drop-down listbox in the *Collation* field to select a collation to use for the index.

Use *Include columns* field to specify columns for *INCLUDE* clause of the index. This option is available in Postgres 11 and later.

Click the *SQL* tab to continue.

Your entries in the *Index* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Index* dialog:

The example shown demonstrates creating an index named *dist_codes* that indexes the values in the *code* column of the *distributors* table.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4.4 documentation](#) > [pgAgent](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Installing pgAgent](#)
 - [Database setup](#)
 - [Daemon installation on Unix](#)
 - [Service installation on Windows](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
 - [Using pgAgent](#)
 - [Installing pgAgent](#)
 - [Creating a pgAgent Job](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Installing pgAgent

pgAgent runs as a daemon on Unix systems, and a service on Windows systems. In most cases it will run on the database server itself – for this reason, pgAgent is not automatically configured when pgAdmin is installed. In some cases however, it may be preferable to run pgAgent on multiple systems, against the same database; individual jobs may be targeted at a particular host, or left for execution by any host. Locking prevents execution of the same instance of a job by multiple hosts.

Database setup

Before using pgAdmin to manage pgAgent, you must create the pgAgent extension in the maintenance database registered with pgAdmin. To install pgAgent on a PostgreSQL host, connect to the `postgres` database, and navigate through the `Tools` menu to open the `Query` tool. For server versions 9.1 or later, and pgAgent 3.4.0 or later, enter the following command in the query window, and click the `Execute` icon:

```
CREATE EXTENSION pgagent;
```

This command will create a number of tables and other objects in a schema called 'pgagent'.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [pgAgent](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Instal ing pgAgent](#)

[Download](#)

[Database setup](#)

[Daemon instal ation on Unix](#)

[Service instal ation on Windows](#)



[Getting Started](#)

[FAQ](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)



[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[Latest Docs](#)

[Management Basics](#)

[Backup and Restore](#)



[Developer Tools](#)

[pgAgent](#)

[Ask a](#)

[Using pgAgent](#)

[Question](#)

Installing pgAgent

[Creating a pgAgent Job](#)



[pgAdmin Project Contributions](#)

[Report a Bug](#)

[Release Notes](#)

[Licence](#)



Installing pgAgent

Screenshots

pgAgent runs as a daemon on Unix systems, and a service on Windows systems. In most cases it will run on the database server itself - for this reason, pgAgent is not automatically configured when pgAdmin is installed. In some cases however, it may be preferable to run pgAgent on multiple systems, against the same database; individual jobs may be targeted at a particular host, or left for execution by any host. Locking prevents execution of the same instance of a job by multiple hosts.

Database setup

Before using pgAdmin to manage pgAgent, you must create the pgAgent extension in the maintenance database registered with pgAdmin. To install pgAgent on a PostgreSQL host, connect to the *postgres* database, and navigate through the *Tools* menu to open the Query tool. For server versions 9.1 or later, and pgAgent 3.4.0 or later, enter the following command in the query window, and click the *Execute* icon: **CREATE EXTENSION pgagent;**

This command will create a number of tables and other objects in a schema called 'pgagent'.

The database must also have the plpgsql procedural language installed -

use the PostgreSQL CREATE LANGUAGE command to install plpgsql if necessary. To install plpgsql, enter the following command in the query window, and click the *Execute* icon:

CREATE LANGUAGE plpgsql;

If you are using an earlier version of PostgreSQL or pgAgent, use the *Open file* icon on the Query Tool toolbar to open a browser window and locate the *pgagent.sql* script. The installation script is

installed by pgAdmin, and the installation location varies from operating system to operating system:

On Windows, it is usually located under *C:\Program files\pgAdmin 11* (or *C:\Program files\PostgreSQL\8.x\pgAdmin 11* if installed with the PostgreSQL server installer).

On Linux, it is usually located under

/usr/local/pgadmin3/share/pgadmin3 or */usr/share/pgadmin3*.

After loading the file into the Query Tool, click the *Execute* icon to execute the script. The script will create a number of tables and other objects in a schema named *pgagent*.

Daemon installation on Unix 

Note

pgAgent is available in Debian/Ubuntu (DEB) and Redhat/Fedora (RPM)

[packages for Linux users, as well as source code. See the pgAdmin Website.](#) for more information.

To install the pgAgent daemon on a Unix system, you will normally need to have root privileges to modify the system startup scripts. Modifying system startup scripts is quite system-specific so you should consult your system documentation for further information.

The program itself takes few command line options, most of which are only needed for debugging or specialised configurations: Usage:

`/path/to/pgagent [options] <connect-string>`

options:

-f run in the foreground (do **not** detach from the terminal)

- t <poll time interval in seconds (default 10)>
- r <retry period after connection abort in seconds (>=10, default 30)>
- s <log file (messages are logged to STDOUT if not specified)>
- l <logging verbosity (ERROR=0, WARNING=1, DEBUG=2, default 0)>

The connection string is a standard PostgreSQL libpq connection string (see the [PostgreSQL documentation on the connection string](#) for further details). For example, the following command line will run pgAgent against a server listening on the localhost, using a database called

'pgadmin', connecting as the user 'postgres':

```
/path/to/pgagent hostaddr=127.0.0.1 dbname=postgres  
user=postgres
```

Service installation on Windows [¶](#)

Note

[pgAgent is available in a pre-built installer if you use EnterpriseDB's PostgreSQL Installers. Use the StackBuilder application to download](#) and install it. If installed in this way, the service will automatically be created and the instructions below can be ignored.

pgAgent can install itself as a service on Windows systems. The command line options available are similar to those on Unix systems, but include an additional parameter to tell the service what to do: Usage:

```
pgAgent REMOVE <serviceName>
```

```
pgAgent INSTALL <serviceName> [options] <connect-string>
```

`pgAgent DEBUG [options] <connect-string>`

options:

`-u <user or DOMAIN\user>`

`-p <password>`

`-d <displayname>`

`-t <poll time interval in seconds (default 10)>`

`-r <retry period after connection abort in seconds (>=10, default 30)>`

`-l <logging verbosity (ERROR=0, WARNING=1, DEBUG=2, default 0)>`

The service may be quite simply installed from the command line as follows (adjust the path as required):

```
"C:\Program Files\pgAgent\bin\pgAgent" INSTALL pgAgent -u
postgres -p secret hostaddr=127.0.0.1 dbname=postgres
user=postgres
```

You can then start the service at the command line using `net start pgAgent`, or from the `Services` control panel applet. Any logging output or errors will be reported in the Application event log. The `DEBUG` mode may be used to run `pgAgent` from a command prompt. When run this way, log messages will output to the command window.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) > [Getting Started](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Keyboard Shortcuts](#)
 - [Main Browser Window](#)
 - [Dialog Tabs](#)
 - [Property Grid Controls](#)
 - [SQL Editors](#)
 - [Query Tool](#)
 - [Debugger](#)
 - [Inner Tab and Panel Navigation](#)
 - [Access Key](#)
- [Getting Started](#)
 - [Deployment](#)
 - [Login Dialog](#)
 - [User Management Dialog](#)
 - [Change User Password Dialog](#)
 - [User Interface](#)
 - [Menu Bar](#)
 - [Toolbar](#)
 - [Tabbed Browser](#)
 - [Tree Control](#)
 - [Preferences Dialog](#)
 - [Keyboard Shortcuts](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Keyboard Shortcuts

Keyboard shortcuts are provided in pgAdmin to allow easy access to specific functions. Alternate shortcuts can be configured through File > Preferences if desired."

Main Browser Window

When using main browser window, the following keyboard shortcuts are available:

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#)
[Development](#)

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Keyboard Shortcuts](#)

[Download](#)

[Main Browser Window](#)

[Dialog Tabs](#)

[Property Grid Controls](#)



[SQL Editors](#)

[FAQ](#)

[Query Tool](#)

[Debugger](#)



[Inner Tab and Panel Navigation](#)

[Access Key](#)

[Latest Docs](#)

[Getting Started](#)

[Deployment](#)



[Login Dialog](#)

[User Management Dialog](#)

[Ask a](#)

[Change User Password Dialog](#)

[Question](#)

[User Interface](#)

[Menu Bar](#)



[Toolbar](#)

[Report a Bug](#)

[Tabbed Browser](#)

[Tree Control](#)

[Preferences Dialog](#)



Keyboard Shortcuts

[Screenshots](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Keyboard Shortcuts

Keyboard shortcuts are provided in pgAdmin to allow easy access to specific functions. Alternate shortcuts can be configured through File > Preferences if desired.”

Main Browser Window

When using main browser window, the following keyboard shortcuts are available:

Sh

S o

h r

o t

r c

t u

c t

u tfo

f r

o ral

a lpl

p at

a f

t o

f r

o m

r s

m

Fu

F n

u c

n t

c iton

o

Alt+Shift+F

Open the File menu

Alt+Shift+O

Open the Object menu

Alt+Shift+L

Open the Tools menu

Alt+Shift+H

Open the Help menu

Alt+Shift+B

Focus the browser tree

Alt+Shift+[

Move tabbed panel backward

Alt+Shift+]

Move tabbed panel forward

Alt+Shift+Q

Open the Query Tool in the current

database

Alt+Shift+V

View Data in the selected table/view

Alt+Shift+C

Open the context menu

Alt+Shift+N

Create an object

Alt+Shift+E

Edit object properties

Alt+Shift+D

Delete the object

Alt+Shift+G

Direct debugging

Dialog Tabs 

Use the shortcuts below to navigate the tabs on dialogs: Shortcut for all platforms

Function

Control+Shift+[

Dialog tab backward

Control+Shift+]

Dialog tab forward

Property Grid Controls

Use the shortcuts below when working with property grid controls:
Shortcut for all platforms

Function

Control+Shift+A

Add row in Grid

Tab

Move focus to the next control

Shift+Tab

Move focus to the previous control

Return

Pick the selected item in a combo box Control+Shift+A

Add row in Grid

SQL Editors

When using the syntax-highlighting SQL editors, the following shortcuts are available:

Shortcut

(Windows/Linux)

Shortcut (Mac)

Function

Alt + Left

Option + Left

Move to the beginning of the
line

Alt + Right

Option + Right

Move to the end of the line

Ctrl + Alt + Left

Cmd + Option + Left Move left one word Ctrl + Alt + Right

Cmd + Option +

Move right one word

Right

Shortcut

(Windows/Linux)

Shortcut (Mac)

Function

Ctrl + /

Cmd + /

Comment selected code

(Inline)

Ctrl + .

Cmd + .

Uncomment selected code

(Inline)

Ctrl + Shift + /

Cmd + Shift + /

Comment/Uncomment code

(Block)

Ctrl + a

Cmd + a

Select al

Ctrl + c

Cmd + c

Copy selected text to the
clipboard

Ctrl + r

Cmd + r

Redo last edit un-done

Ctrl + v

Cmd + v

Paste text from the clipboard

Ctrl + z

Cmd + z

Undo last edit

Tab

Tab

Indent selected text

Shift + Tab

Shift + Tab

Un-indent selected text

Alt + g

Option + g

Jump (to line:column)

Ctrl + Space

Ctrl + Space

Auto-complete

Ctrl + f

Cmd + f

Find

Ctrl + g

Cmd + g

Find next

Ctrl + Shift + g

Cmd + Shift + g

Find previous

Ctrl + Shift + f

Cmd + Shift + f

Replace

Query Tool 

When using the Query Tool, the following shortcuts are available:
Shortcut

(Windows/Linux)

Shortcut (Mac)

Function

F5

F5

Execute query

F6

F6

Save data changes

F7

F7

EXPLAIN query

Shift + F7

Shift + F7

EXPLAIN ANALYZE query

F8

F8

Execute query to CSV file

<accesskey> + o

<accesskey> + o

Open file

<accesskey> + s

<accesskey> + s

Save file

<accesskey> + n

<accesskey> + n

Find option drop down

<accesskey> + c

<accesskey> + c

Copy row(s)

<accesskey> + p

<accesskey> + p

Paste row(s)

<accesskey> + d

<accesskey> + d

Delete row(s)

<accesskey> + f

<accesskey> + f

Filter dialog

<accesskey> + i

<accesskey> + i

Filter options drop down

<accesskey> + r

<accesskey> + r

Row limit

<accesskey> + q

<accesskey> + q

Cancel query

Shortcut

(Windows/Linux)

Shortcut (Mac)

Function

<accesskey> + l

<accesskey> + l

Clear option drop down

<accesskey> + x

<accesskey> + x

Execute option drop down

<accesskey> + t

<accesskey> + t

Display connection status

<accesskey> + y

<accesskey> + y

Copy SQL on history panel

Debugger 

When using the Debugger, the following shortcuts are available:

Shortcut

(Windows/Linux)

Shortcut (Mac)

Function

<accesskey> + i

<accesskey> + i

Step in

<accesskey> + o

<accesskey> + o

Step over

<accesskey> + c

<accesskey> + c

Continue/Restart

<accesskey> + t

<accesskey> + t

Toggle breakpoint

<accesskey> + x

<accesskey> + x

Clear all breakpoints

<accesskey> + s

<accesskey> + s

Stop

Alt + Shift + q

Option + Shift + q

Enter or Edit values in Grid

Inner Tab and Panel Navigation 

When using the Query Tool and Debugger, the following shortcuts are available for inner panel navigation:

Shortcut

(Windows/Linux)

Shortcut (Mac)

Function

Alt + Shift +]

Alt + Shift + [

Move to next tab within a
panel

Alt + Shift + [

Alt + Shift +]

Move to previous tab within a
panel

Alt + Shift + Tab

Alt + Shift + Tab

Move between inner panels

Access Key 

<accesskey> is browser and platform dependant. The following table lists the default access keys for supported browsers.

Windows

Linux

Mac

Internet Explorer

Alt

Alt

Chrome

Alt

Alt

Ctrl + Option

Firefox

Alt + Shift

Alt + Shift

Ctrl + Option

Safari

Alt

Ctrl + Option

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Language Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Language Dialog ||

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Language Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)



[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)



[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)



[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

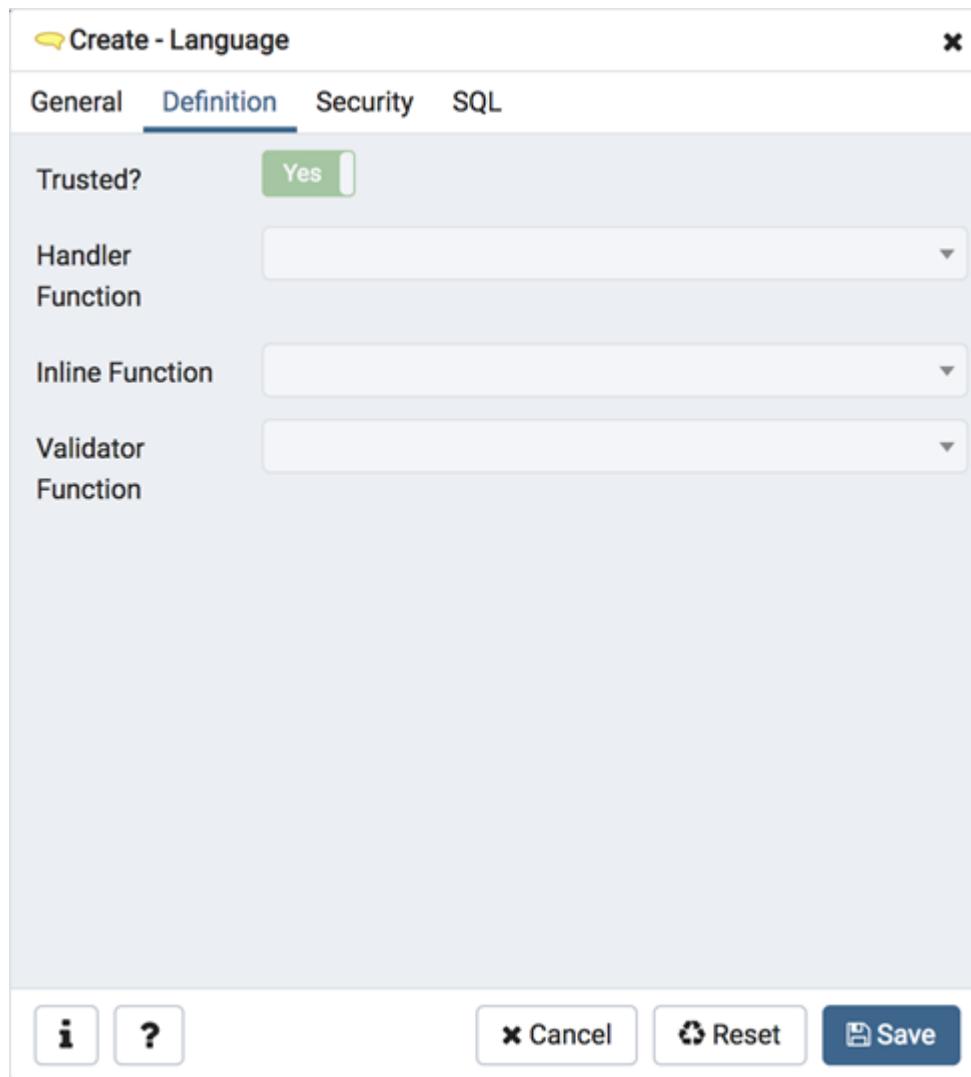
Language Dialog

Create - Language

General Definition Security SQL

Name	plperlu
Owner	enterprisedb
Comment	

i **?** **x Cancel** **♻ Reset** **💾 Save**



Use the CREATE LANGUAGE dialog to register a new procedural language.

The *Language* dialog organizes the registration of a procedural language through the following dialog tabs: *General*, *Definition*, and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

Use the fields in the *General* tab to identify a language: Use the drop-down listbox next to *Name* to select a language script.

Use the drop-down listbox next to *Owner* to select a role.

Store notes about the language in the *Comment* field.

Click the *Definition* tab to continue.

The screenshot shows a 'Create - Language' dialog box with the 'Security' tab selected. The 'Privileges' section contains a table with columns 'Grantee', 'Privileges', and 'Grantor'. Under 'Grantee', there is a dropdown set to 'PUBLIC'. Under 'Privileges', there are two checkboxes: 'USAGE' (checked) and 'WITH GRANT OPTION' (unchecked). Under 'Grantor', there is a dropdown set to 'enterprisedb'. The 'Security Labels' section is currently empty. At the bottom, there are buttons for 'Save' (highlighted in blue), 'Cancel', and 'Reset'.

Use the fields in the *Definition* tab to define parameters: Move the *Trusted?* switch to the *No* position to specify only users with PostgreSQL superuser privilege can use this language. The default is *Yes*.

When enabled, use the drop-down listbox next to *Handler Function* to select the function that will be called to execute the language's functions.

When enabled, use the drop-down listbox next to *Inline Function* to select the function that will be called to execute an anonymous code block (DO command) in this language.

When enabled, use the drop-down listbox next to *Validator Function* to select the function that will be called when a new function in the language is created, to validate the new function.

Click the *Security* tab to continue.

Use the *Security* tab to assign privileges and define security labels.

Use the *Privileges* panel to assign privileges to a role. Click the *Add* icon (+) to set privileges for database objects:

Select the name of the role from the drop-down listbox in the *Grantee* field.

Click inside the *Privileges* field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.

Click the *Add* icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

 Create - Language x

General Definition Security **SQL**

```
1 CREATE LANGUAGE plperl;
2 ALTER LANGUAGE plperl
3     OWNER TO enterprisedb;
4 GRANT USAGE ON LANGUAGE plperl TO PUBLIC;
```

i ? ✖ Cancel ↻ Reset 💾 Save

Use the *Security Labels* panel to define security labels applied to the function. Click the *Add* icon (+) to add each security label selection: Specify a security label provider in the *Provider* field. The named provider must be loaded and must consent to the proposed labeling operation.

Specify a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL

places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the *Add* icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the **SQL** tab to continue.

Your entries in the *Language* dialog generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Language* dialog:

The example shown demonstrates creating the procedural language named *plperl*.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) » [Managing Cluster Objects](#) » [previous](#) [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Login/Group Role Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
 - [Database Dialog](#)
 - [Move Objects Dialog](#)
 - [Resource Group Dialog](#)
 - [Login/Group Role Dialog](#)
 - [Tablespace Dialog](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Login/Group Role Dialog

Use the Login/Group Role dialog to define a role. A role may be an individual user (with or without login privileges) or a group of users. Note that roles defined at the cluster level are shared by all databases in the cluster.

The Login/Group Role dialog organizes the creation and management of roles through the following dialog tabs: General, Definition, Privileges, Parameters, and Security. The SQL tab displays the SQL code generated by dialog selections.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Cluster Objects](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Login/Group Role Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Database Dialog](#)

[Move Objects Dialog](#)



[Resource Group Dialog](#)

Login/Group Role Dialog

[Latest Docs](#)

[Tablespace Dialog](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[Ask a](#)

[Management Basics](#)

[Question](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)



[pgAdmin Project Contributions](#)

[Report a Bug](#)

[Release Notes](#)

[Licence](#)

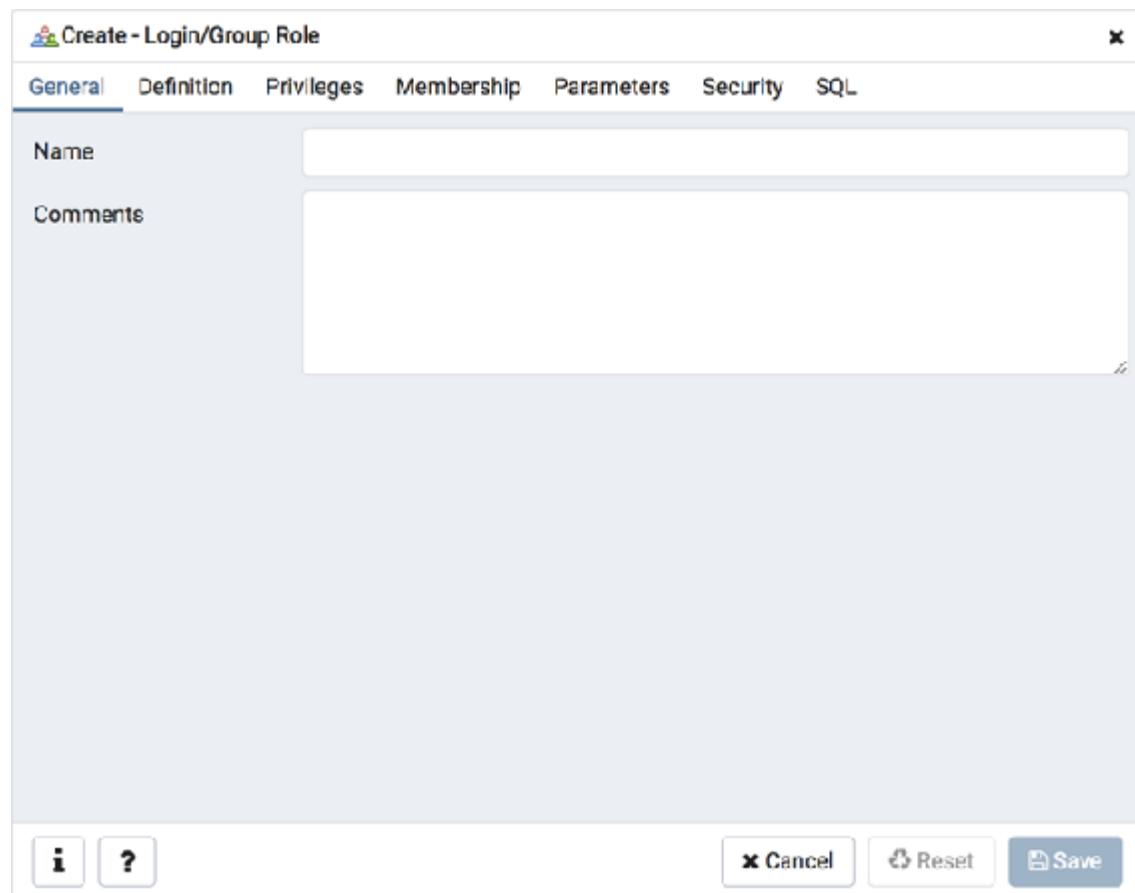


Login/Group Role Dialog

Screenshots

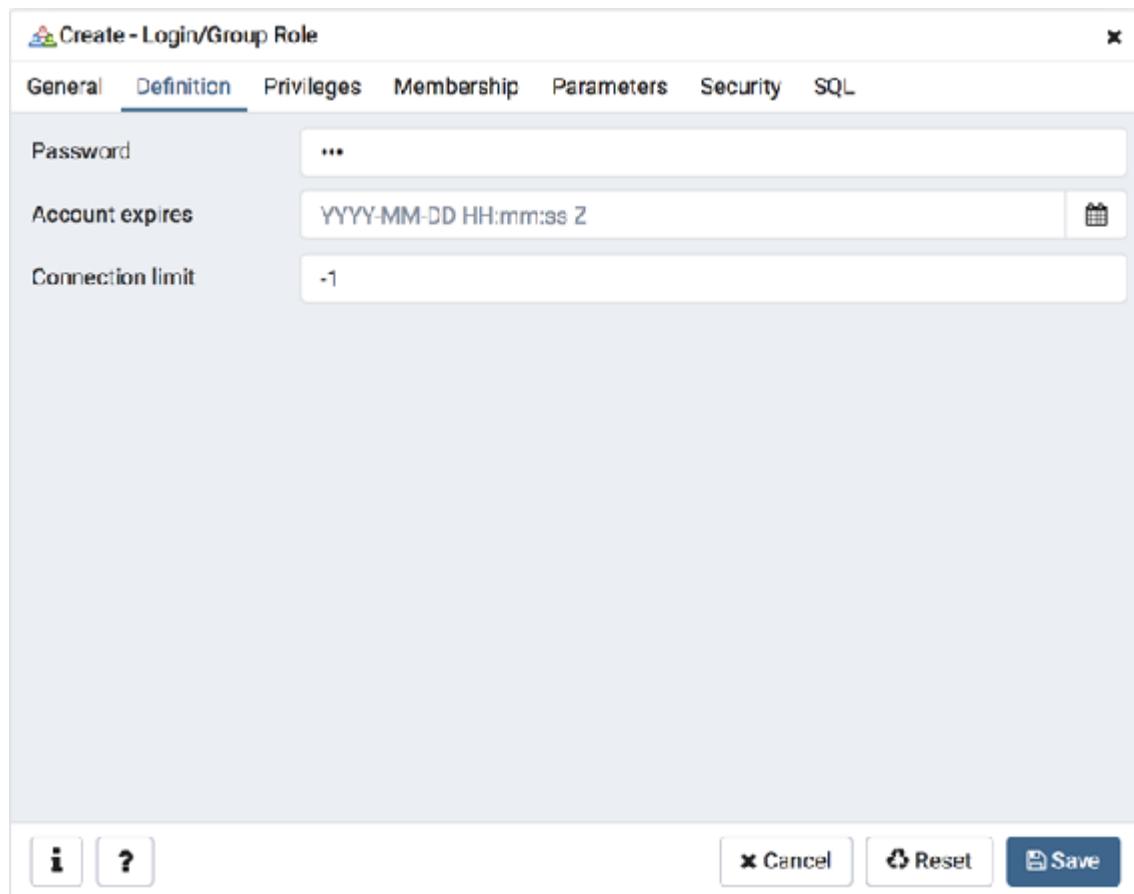
Use the *Login/Group Role* dialog to define a role. A role may be an individual user (with or without login privileges) or a group of users. Note that roles defined at the cluster level are shared by all databases in the cluster.

The *Login/Group Role* dialog organizes the creation and management of roles through the following dialog tabs: *General*, *Definition*, *Privileges*, *Membership*, *Parameters*, *Security*, and *SQL*. The *SQL* tab displays the SQL code generated by dialog selections.



The screenshot shows the 'Create - Login/Group Role' dialog box. The 'General' tab is selected. The interface includes:

- Name:** A text input field.
- Comments:** A large text area for notes.
- Buttons:** At the bottom left are 'i' and '?' buttons. At the bottom right are 'Cancel', 'Reset', and 'Save' buttons.

A screenshot of a software dialog box titled "Create - Login/Group Role". The "General" tab is selected. The interface includes tabs for "Definition", "Privileges", "Membership", "Parameters", "Security", and "SQL".
Fields on the General tab:

- Password:** A password field containing three asterisks (...).
- Account expires:** A date and time input field set to "YYYY-MM-DD HH:mm:ss Z", with a calendar icon to its right.
- Connection limit:** An input field containing the value "-1".

At the bottom of the dialog are buttons for "Cancel", "Reset", and "Save".

Use the fields on the *General* tab to identify the role.

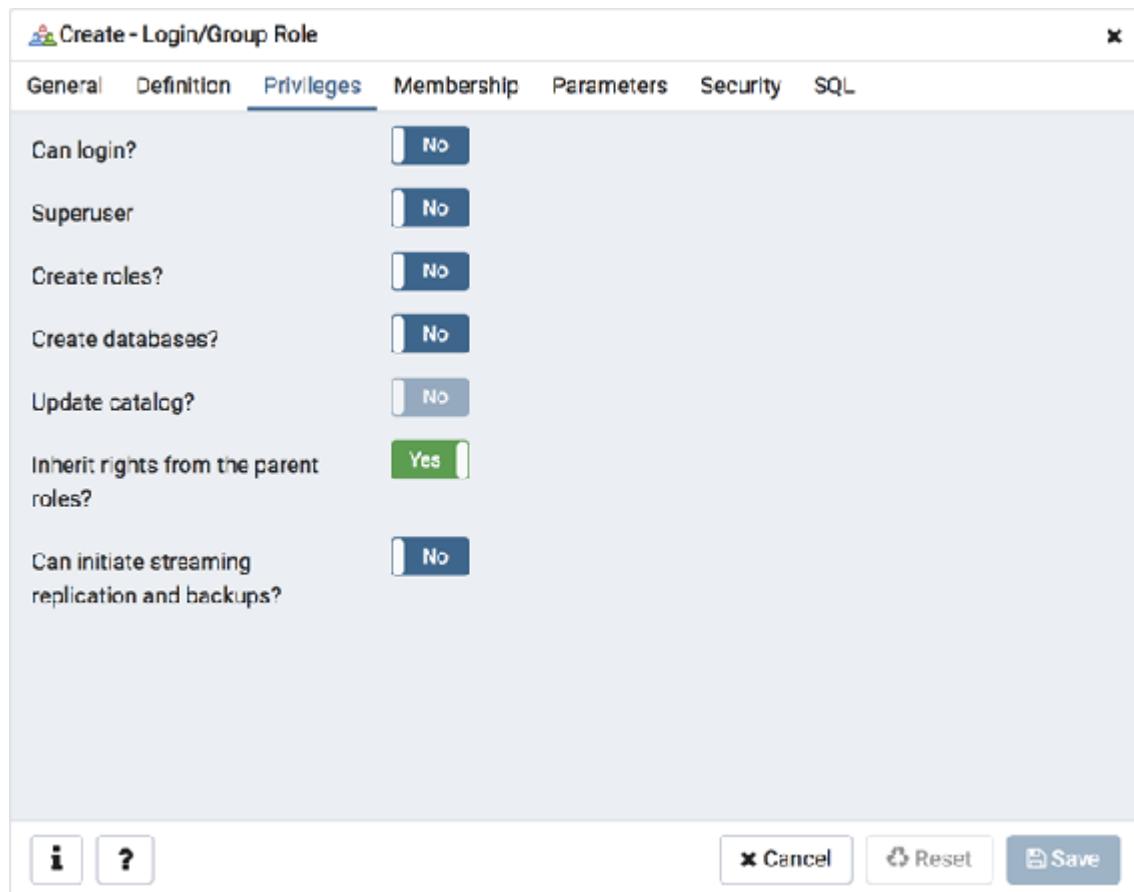
Use the *Name* field to provide the name of the role. The name will be displayed in the tree control.

Provide a note about the role in the *Comments* field.

Click the *Definition* tab to continue.

Use the *Definition* tab to set a password and configure connection rules: Provide a password that will be associated with the role in the *Password* field.

Provide an expiration date for the password in the *Account Expires* field (the role does not expire). The expiration date is not enforced when a user logs in with a non-password-based authentication method.

A screenshot of the 'Create - Login/Group Role' dialog box. The 'Privileges' tab is selected. The form contains several configuration options with their current values:

- Can login? (No)
- Superuser (No)
- Create roles? (No)
- Create databases? (No)
- Update catalog? (No)
- Inherit rights from the parent roles? (Yes)
- Can initiate streaming replication and backups? (No)

At the bottom, there are informational buttons (Info, Help), and action buttons: 'Cancel', 'Reset', and 'Save'.

If the role is a login role, specify how many concurrent connections the role can make in the *Connection Limit* field. The default value (-1) allows unlimited connections.

Click the *Privileges* tab to continue.

Use the *Privileges* tab to grant privileges to the role.

Move the *Can login?* switch to the *Yes* position if the role has login privileges. The default value is *No*.

Move the *Superuser* switch to the *Yes* position if the role is a superuser within the database. The default value is *No*.

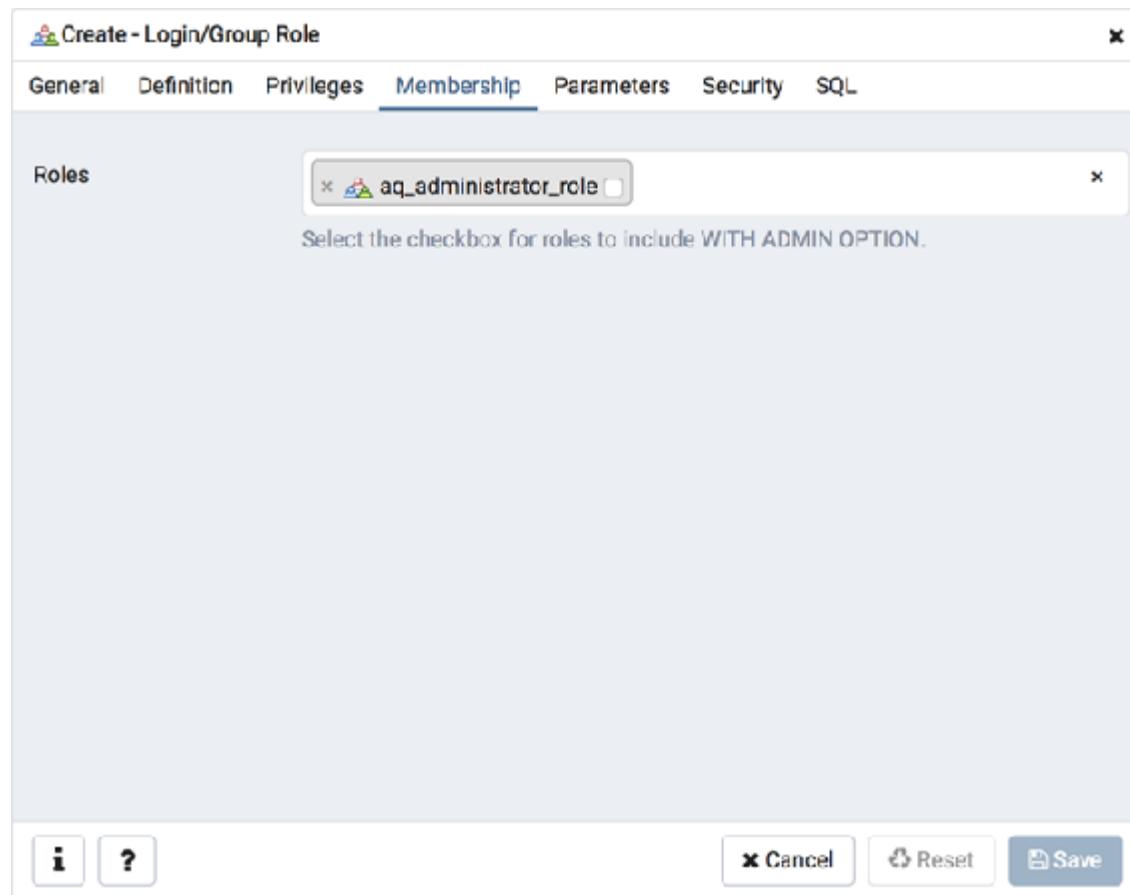
Move the *Create roles?* switch to the *Yes* position to specify whether a role is permitted to create roles. A role with this privilege can alter and drop roles. The default value is *No*.

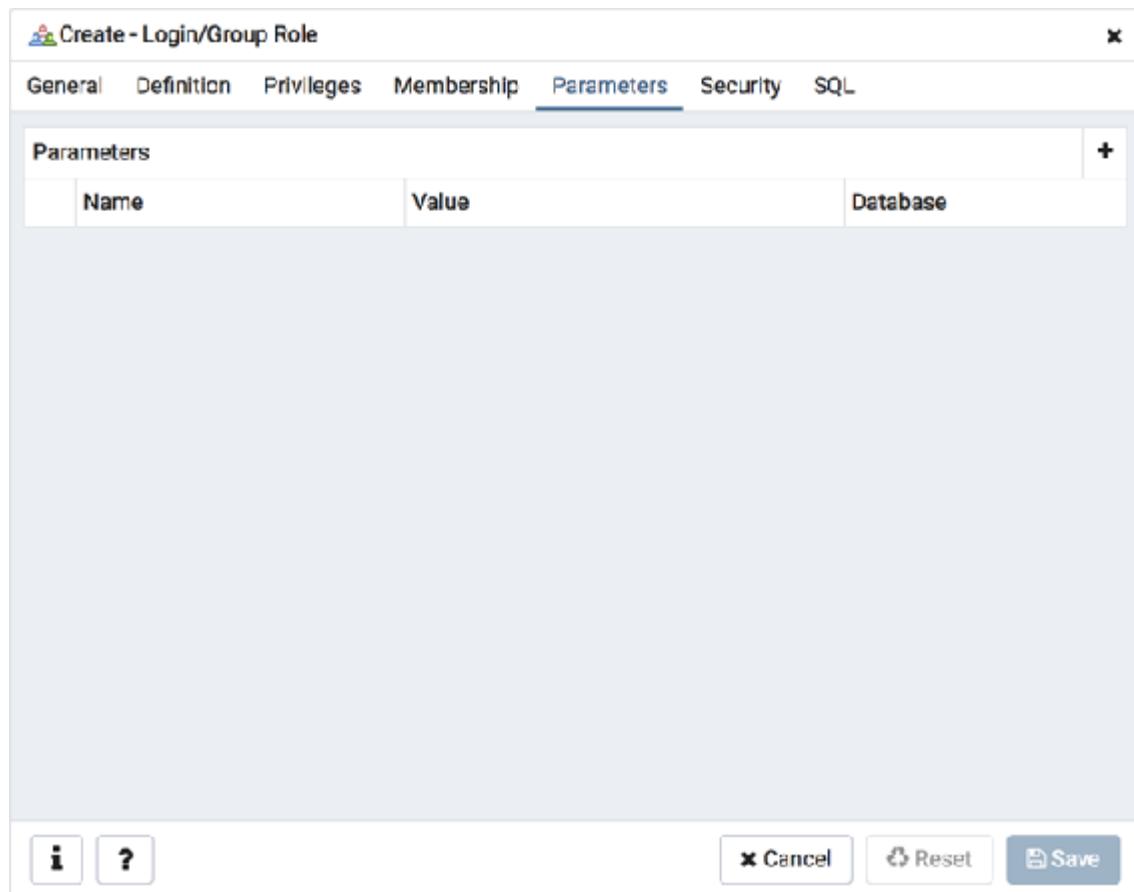
Move the *Create databases* switch to the Yes position to control whether a role can create databases. The default value is No.

The *Update catalog?* switch is disabled until the role is given superuser privileges. Move the *Update catalogs?* switch to the No position to control whether a role can update catalogs. The default value is Yes when the *Superuser* switch is in the Yes position.

Move the *Inherit rights from the parent roles?* switch to the No position if a role does not inherit privileges. The default value is Yes.

Move the *Can initiate streaming replication and backups?* switch to the Yes position to control whether a role can initiate streaming replication or put the system in and out of backup mode. The default value is No.





Specify members of the role in the *Role Membership* field. Click inside the *Roles* field to select role names from a drop down list.

Confirm each selection by checking the checkbox to the right of the role name; delete a selection by clicking the x to the left of the role name. Membership conveys the privileges granted to the specified role to each of its members.

Click the *Parameters* tab to continue.

Use the fields on the *Parameters* tab to set session defaults for a selected configuration parameter when the role is connected to a specified database. This tab invokes the ALTER ROLE... SET configuration_parameter syntax. Click the *Add* icon (+) to assign a value for a parameter.

Use the drop-down listbox in the *Name* field to select a parameter.

Use the *Value* field to specify a value for the parameter.

The screenshot shows a software interface titled "Create - Login/Group Role". The "Security" tab is currently selected. Below it, there is a table titled "Security Labels" with two columns: "Provider" and "Security Label". A plus sign (+) icon is located to the right of the table, indicating the ability to add new entries. At the bottom of the dialog, there are several buttons: "Cancel", "Reset", and "Save".

Use the drop-down listbox in the *Database* field to select a database.

Click the *Add* icon (+) to specify each additional parameter; to discard a parameter, click the trash icon to the left of the row and confirm the deletion in the *Delete Row* popup.

Click the *Security* tab to continue.

Use the *Security* tab to define security labels applied to the role. Click the *Add* icon (+) to add each security label selection.

Specify a security label provider in the *Provider* field. The named provider must be loaded and must consent to the proposed labeling operation.

Specify a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

To discard a security label, click the trash icon to the left of the row and confirm the deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

Your entries in the *Login/Group Role* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example 

The following is an example of the sql command generated by user selections in the *Login/Group Role* dialog:

```
1 CREATE ROLE "EDB" WITH
2     NOLOGIN
3     NOSUPERUSER
4     NOCREATEDB
5     NOCREATEROLE
6     INHERIT
7     NOREPLICATION
8     CONNECTION LIMIT -1
9     PASSWORD 'xxxxxx';
10 GRANT aq_administrator_role TO "EDB" WITH ADMIN OPTION;
```

The screenshot shows a software interface for creating a login/group role. The title bar says 'Create - Login/Group Role'. Below it is a navigation bar with tabs: General, Definition, Privileges, Membership, Parameters, Security, and SQL. The SQL tab is currently selected. The main area contains a block of SQL code. At the bottom are several buttons: Info (i), Help (?), Cancel, Reset, and Save.

The example creates a login role named *alice* with *pem_user* privileges; the role can make unlimited connections to the server at any given time.

Click the Info button (*i*) to access online SQL help.

Click the Help button (?) to access the documentation for the dialog.

Click the Save button to save work.

Click the Cancel button to exit without saving work.

Click the Reset button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest (4.19) | 4.18 | 4.17 | Development

[pgAdmin 4 documentation](#) > [Getting Started](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

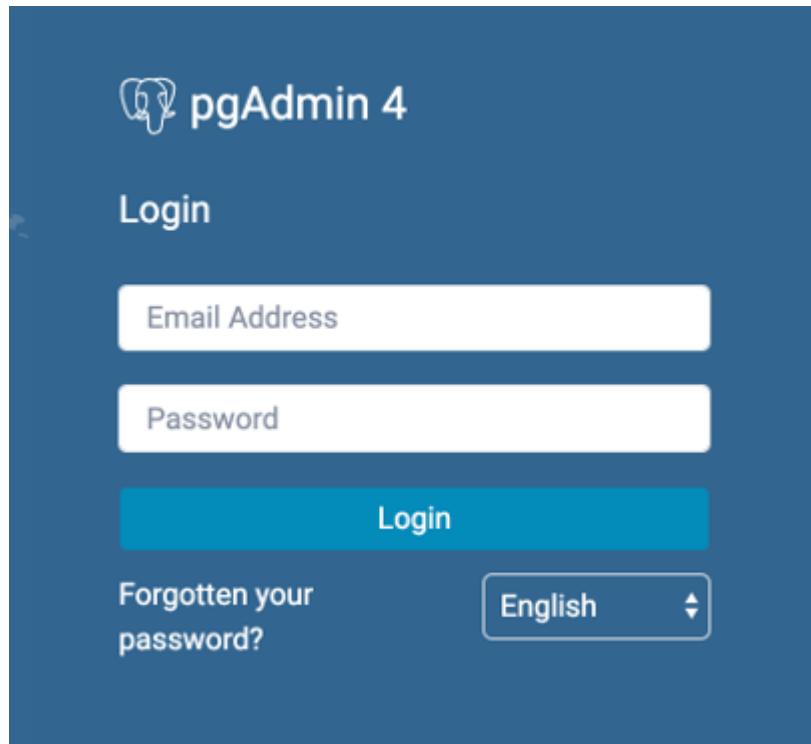
Contents

- [Login Dialog](#)
 - [Recovering a Lost Password](#)
- [Getting Started](#)
 - [Deployment](#)
 - [Login Dialog](#)
 - [User Management Dialog](#)
 - [Change User Password Dialog](#)
 - [User Interface](#)
 - [Menu Bar](#)
 - [Toolbar](#)
 - [Tabbed Browser](#)
 - [Tree Control](#)
 - [Preferences Dialog](#)
 - [Keyboard Shortcuts](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Login Dialog

Use the Login dialog to log in to pgAdmin:





Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



Login Dialog

[Download](#)

[Recovering a Lost Password](#)

[Getting Started](#)



[Deployment](#)

Login Dialog

[FAQ](#)

[User Management Dialog](#)

[Change User Password Dialog](#)



[User Interface](#)

[Menu Bar](#)

[Latest Docs](#)

[Toolbar](#)

[Tabbed Browser](#)



[Tree Control](#)

[Preferences Dialog](#)

[Ask a](#)

[Keyboard Shortcuts](#)

[Question](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)



[Managing Database Objects](#)

[Report a Bug](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)



[Developer Tools](#)

[Screenshots](#)

[pgAgent](#)

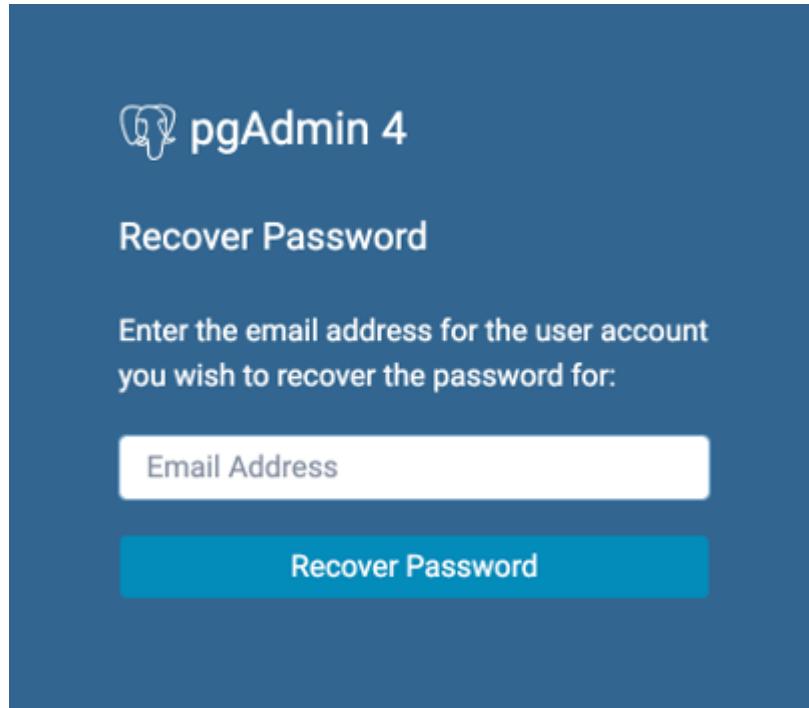
[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Login Dialog

Use the *Login* dialog to log in to pgAdmin:



Use the fields in the *Login* dialog to authenticate your connection:
Provide the email address associated with your account in the *Email Address* field.

Provide your password in the *Password* field.

Click the *Login* button to securely log into pgAdmin.

Please note that, if the pgAdmin server is restarted then you will be logged out. You need to re-login to continue.

Recovering a Lost Password

If you cannot supply your password, click the *Forgotten your password?*

button to launch a password recovery utility.

Provide the email address associated with your account in the *Email Address* field.

Click the *Recover Password* button to initiate recovery. An email, with directions on how to reset a password, will be sent to the address entered in the *Email Address* field.

If you have forgotten the email associated with your account, please contact your administrator.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Management Basics](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
 - [Add Named Restore Point Dialog](#)
 - [Change Password Dialog](#)
 - [Grant Wizard](#)
 - [Import/Export Data Dialog](#)
 - [Maintenance Dialog](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Maintenance Dialog

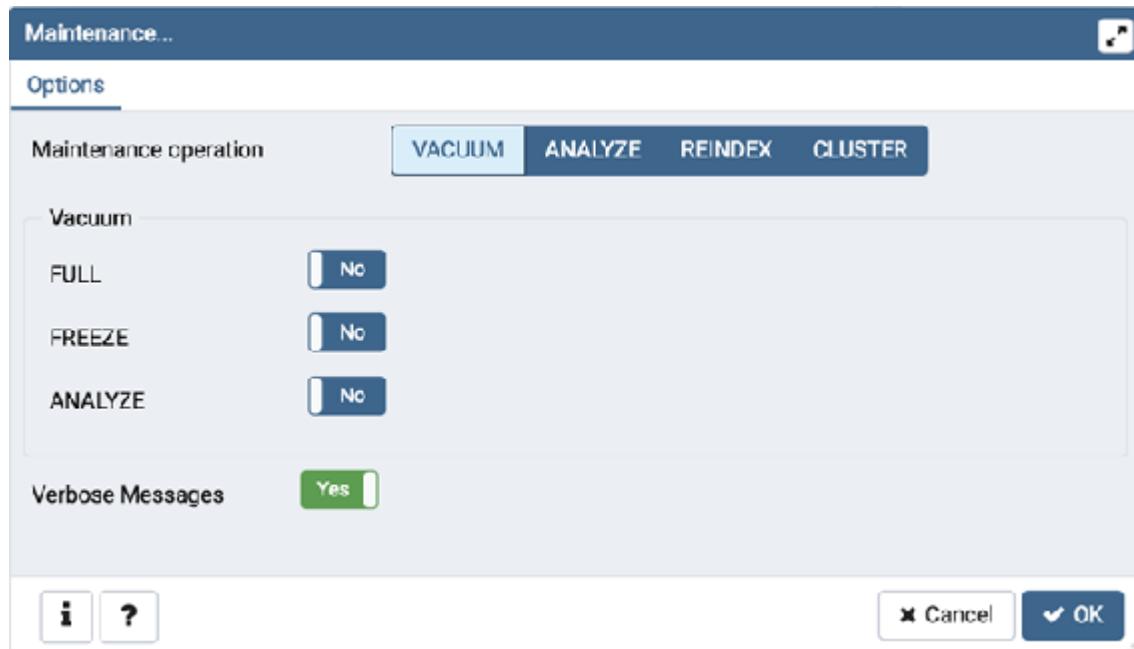
Use the Maintenance dialog to VACUUM, ANALYZE, REINDEX or CLUSTER a database or selected database objects.



While this utility is useful for ad-hoc maintenance purposes, you are encouraged to perform automatic VACUUM jobs on a regular schedule.

Select a button next to Maintenance operation to specify the type of maintenance:

- Click VACUUM to scan the selected database or table to reclaim storage used by dead tuples.



Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Management Basics](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[FAQ](#)

[Management Basics](#)

[Add Named Restore Point Dialog](#)



[Change Password Dialog](#)

[Grant Wizard](#)

[Latest Docs](#)

[Import/Export Data Dialog](#)

Maintenance Dialog



[Backup and Restore](#)

[Developer Tools](#)

[Ask a](#)

[pgAgent](#)

[Question](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)



[Licence](#)

[Report a Bug](#)

Maintenance Dialog



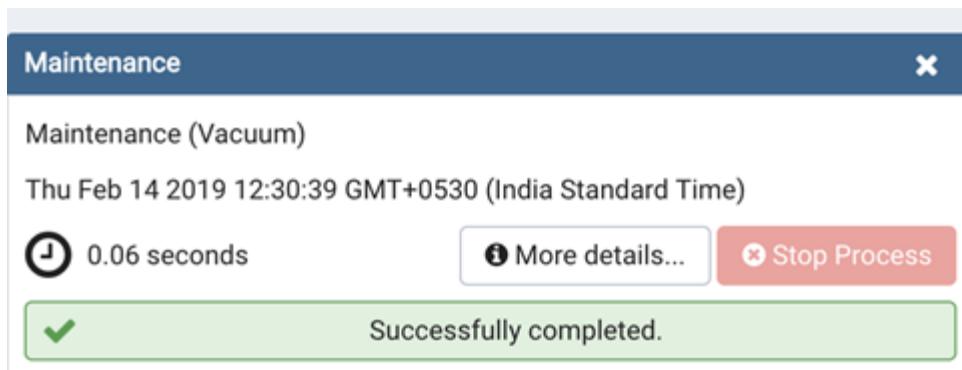
Use the *Maintenance* dialog to VACUUM, ANALYZE, REINDEX or CLUSTER a database or selected database objects.

[Screenshots](#)

While this utility is useful for ad-hoc maintenance purposes, you are encouraged to perform automatic VACUUM jobs on a regular schedule.

Select a button next to *Maintenance operation* to specify the type of maintenance:

Click *VACUUM* to scan the selected database or table to reclaim storage used by dead tuples.



Move the *FULL* switch to the *Yes* position to compact tables by writing a completely new version of the table file without dead space. The default is *No*.

Move the *FREEZE* switch to the *Yes* position to freeze data in a table when it wil have no further updates. The default is *No*.

Move the *ANALYZE* switch to the *Yes* position to issue *ANALYZE* commands whenever the content of a table has changed sufficiently. The default is *No*.

Click *ANALYZE* to update the stored statistics used by the query planner. This enables the query optimizer to select the fastest query plan for optimal performance.

Click *REINDEX* to rebuild any index in case it has degenerated due to the insertion of unusual data patterns. This happens, for example, if you insert rows with increasing index values, and delete low index values.

Click *CLUSTER* to instruct PostgreSQL to cluster the selected table.

To exclude status messages from the process output, move the *Verbose Messages* switch to the *No* position; by default, status messages are included.

When you've completed the dialog, click *OK* to start the background process; to exit the dialog without performing maintenance operations, click *Cancel*.

pgAdmin wil inform you when the background process completes: Use the Stop Process button to stop the Maintenance process.

Use the *Click here for details* link on the notification to open the *Process Watcher* and review detailed information about the execution of the command that performed the import or export:

Process Watcher - Maintenance

VACUUM (VERBOSE)

Running Query:

```
VACUUM VERBOSE;
```

⌚ Start time: Thu Feb 14 2019 12:30:05 GMT+0530 (India Standard Time) ⌚ Stop Process

There were 0 unused item pointers.
Skipped 0 pages due to buffer pins, 0 frozen pages.
0 pages are entirely empty.
CPU: user: 0.00 s, system: 0.00 s, elapsed: 0.00 s.
INFO: vacuuming "sys.edb\$statio_all_tables"
INFO: Index "edb\$statio_tab_pk" now contains 0 row versions in 1 pages
DETAIL: 0 index row versions were removed.
0 index pages have been deleted, 0 are currently reusable.
CPU: user: 0.00 s, system: 0.00 s, elapsed: 0.00 s.
INFO: "edb\$statio_all_tables": found 0 removable, 0 nonremovable row versions in 0 out of 0 pages
DETAIL: 0 dead row versions cannot be removed yet, oldest xmin: 1275
There were 0 unused item pointers.
Skipped 0 pages due to buffer pins, 0 frozen pages.
0 pages are entirely empty.
CPU: user: 0.00 s, system: 0.00 s, elapsed: 0.00 s.

✓ Successfully completed. Execution time: 0.45 seconds

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 4.19 documentation](#) »

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
 - [Database Dialog](#)
 - [Move Objects Dialog](#)
 - [Resource Group Dialog](#)
 - [Login/Group Role Dialog](#)
 - [Tablespace Dialog](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Managing Cluster Objects

Some object definitions reside at the cluster level; pgAdmin 4 provides dialogs that allow you to create these objects, manage them, and control their relationships to each other. To access a dialog that allows you to create a database object, right-click on the object type in the pgAdmin tree control, and select the Create option for that object. For example, to create a new database, right-click on the *Databases* node, and select *Create Database...*

- [Database Dialog](#)
- [Move Objects Dialog](#)
- [Resource Group Dialog](#)
- [Login/Group Role Dialog](#)
- [Tablespace Dialog](#)

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#)
[Development](#)

[pgAdmin 4 4.19 documentation](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

Managing Cluster Objects

[Database Dialog](#)



[Move Objects Dialog](#)

[FAQ](#)

[Resource Group Dialog](#)

[Login/Group Role Dialog](#)



[Tablespace Dialog](#)

[Managing Database Objects](#)

[Latest Docs](#)

[Creating or Modifying a Table](#)

[Management Basics](#)



[Backup and Restore](#)

[Developer Tools](#)

[Ask a](#)

[pgAgent](#)

[Question](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)



[Licence](#)

[Report a Bug](#)

Managing Cluster Objects



Some object definitions reside at the cluster level; pgAdmin 4 provides dialogs that allow you to create these objects, manage

them, and control

[Screenshots](#)

their relationships to each other. To access a dialog that allows you to create a database object, right-click on the object type in the pgAdmin tree control, and select the *Create* option for that object. For example, to create a new database, right-click on the *Databases* node, and select *Create Database...*

[Database Dialog](#)

[Move Objects Dialog](#)

[Resource Group Dialog](#)

[Login/Group Role Dialog](#)

[Tablespace Dialog](#)

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 4.19 documentation](#) »

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
 - [Add Named Restore Point Dialog](#)
 - [Change Password Dialog](#)
 - [Grant Wizard](#)
 - [Import/Export Data Dialog](#)
 - [Maintenance Dialog](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Management Basics

pgAdmin provides point and click dialogs that help you perform server management functions. Dialogs simplify tasks such as managing named restore points, granting user privileges, and performing VACUUM, ANALYZE and REINDEX functions.

- [Add Named Restore Point Dialog](#)
- [Change Password Dialog](#)
- [Grant Wizard](#)
- [Import/Export Data Dialog](#)
- [Maintenance Dialog](#)

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#)
[Development](#)

[pgAdmin 4 4.19 documentation](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[FAQ](#)

Management Basics

[Add Named Restore Point Dialog](#)



[Change Password Dialog](#)

[Grant Wizard](#)

[Latest Docs](#)

[Import/Export Data Dialog](#)

[Maintenance Dialog](#)



[Backup and Restore](#)

[Developer Tools](#)

[Ask a](#)

[pgAgent](#)

[Question](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)



[Licence](#)

[Report a Bug](#)

Management Basics



pgAdmin provides point and click dialogs that help you perform server management functions. Dialogs simplify tasks such as

managing named

[Screenshots](#)

restore points, granting user privileges, and performing VACUUM, ANALYZE and REINDEX functions.

[Add Named Restore Point Dialog](#)

[Change Password Dialog](#)

[Grant Wizard](#)

[Import/Export Data Dialog](#)

[Maintenance Dialog](#)

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 4.19 documentation](#) »

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Managing Database Objects

pgAdmin 4 provides simple but powerful dialogs that you can use to design and create database objects. Each dialog contains a series of tabs that you use to describe the object that will be created by the dialog; the

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#)
[Development](#)

[pgAdmin 4 4.19 documentation](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

Managing Database Objects



[Cast Dialog](#)

[FAQ](#)

[Collation Dialog](#)

[Domain Dialog](#)



[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)

[Latest Docs](#)

[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)



[Foreign Server Dialog](#)

[Foreign Table Dialog](#)

[Ask a](#)

[FTS Configuration Dialog](#)

[Question](#)

[FTS Dictionary Dialog](#)

[FTS Parser Dialog](#)



[FTS Template Dialog](#)

[Function Dialog](#)

[Report a Bug](#)

[Language Dialog](#)

[Materialized View Dialog](#)



[Package Dialog](#)

[Procedure Dialog](#)

[Screenshots](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Managing Database Objects

pgAdmin 4 provides simple but powerful dialogs that you can use to design and create database objects. Each dialog contains a series of tabs that you use to describe the object that will be created by the dialog; the

SQL tab displays the SQL command that the server will execute when creating the object.

To access a dialog that allows you to create a database object, right-click on the object type in the pgAdmin tree control, and select the *Create* option for that object. For example, to create a new cast, right-click on the *Casts* node, and select *Create Cast...*

[Cast Dialog](#)

[Collation Dialog](#)

[Domain Dialog](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)

[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Foreign Server Dialog](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)

[FTS Dictionary Dialog](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[Language Dialog](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Materialized View Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Materialized View Dialog 

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Materialized View Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)



[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)



[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

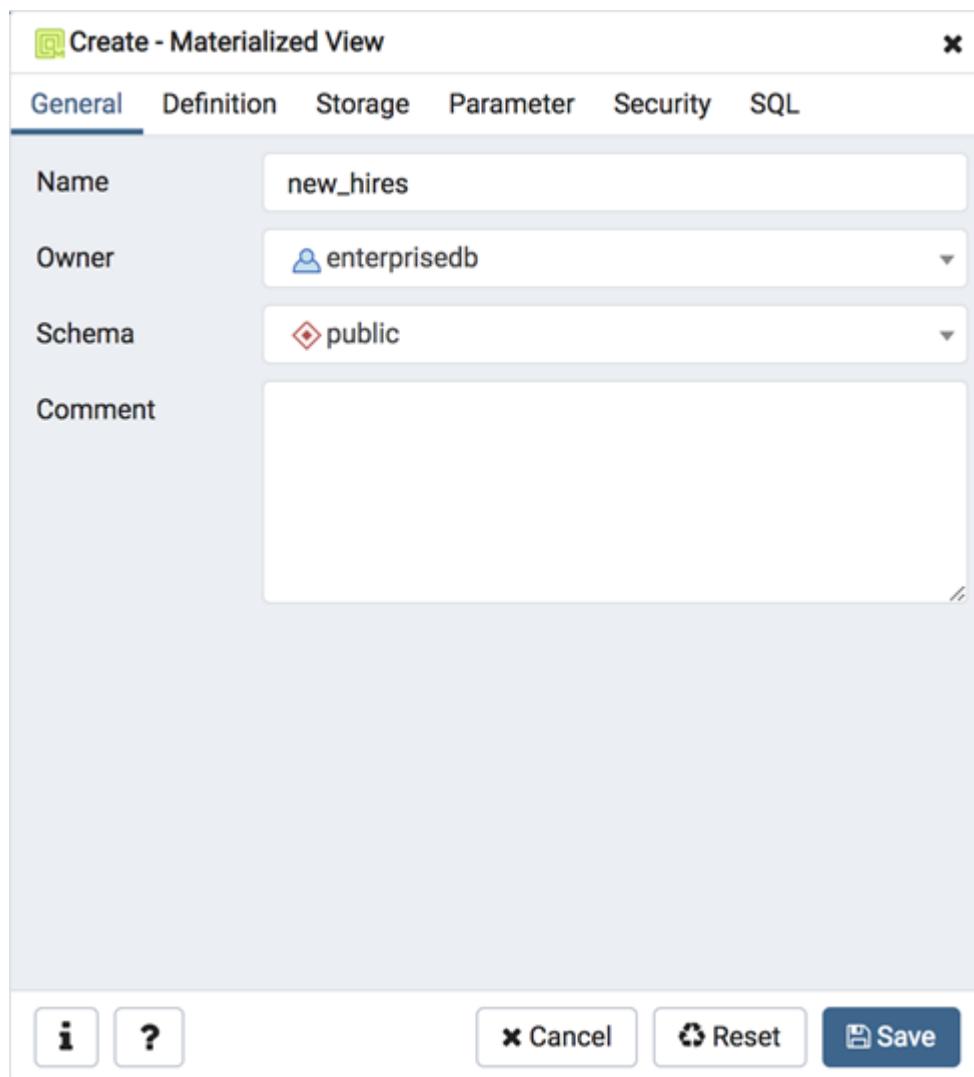
[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

Materialized View Dialog



Use the *Materialized View* dialog to define a materialized view. A materialized view is a stored or cached view that contains the result set of a query. Use the REFRESH MATERIALIZED VIEW command to update the content of a materialized view.

The *Materialized View* dialog organizes the development of a materialized_view through the following dialog tabs: *General*, *Definition*, *Storage*, *Parameter*, and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

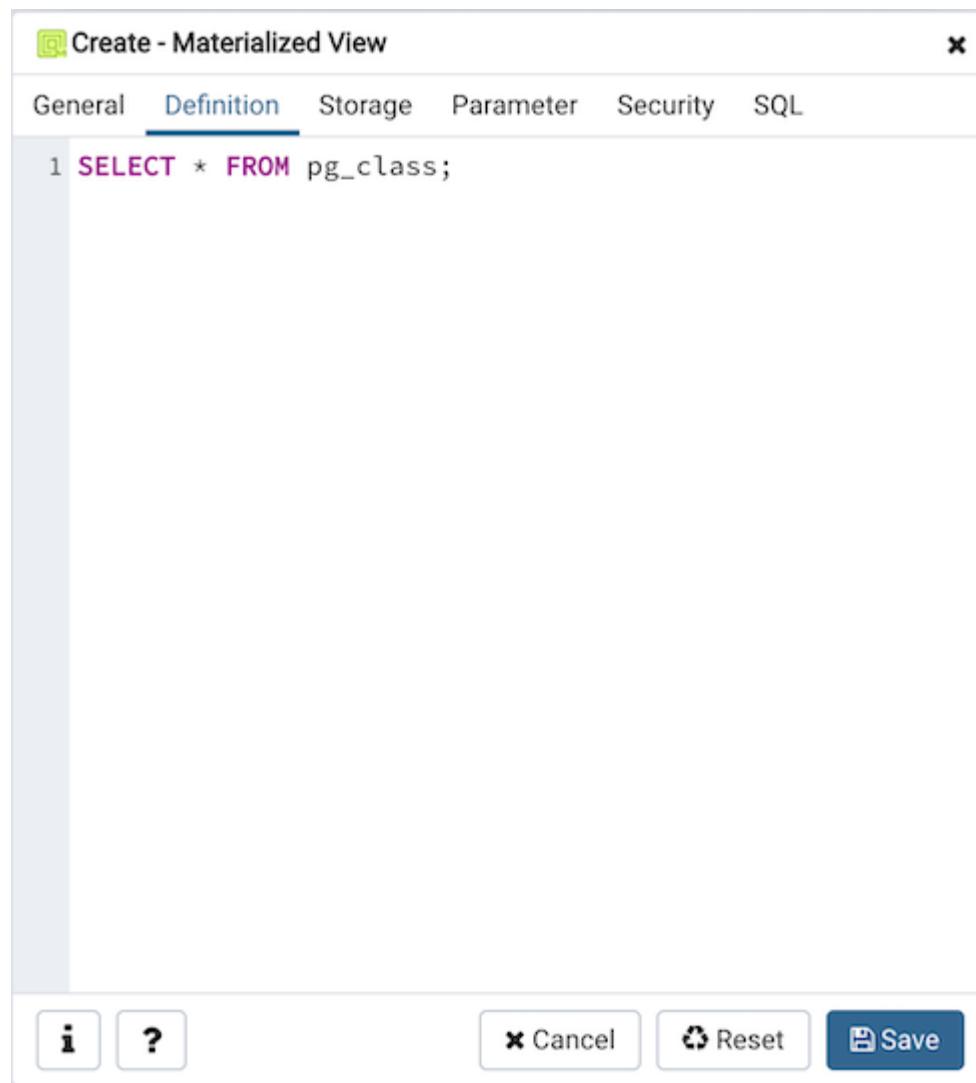
Use the fields in the *General* tab to identify the materialized view:
Use the *Name* field to add a descriptive name for the materialized view. The name will be displayed in the *pgAdmin* tree control.

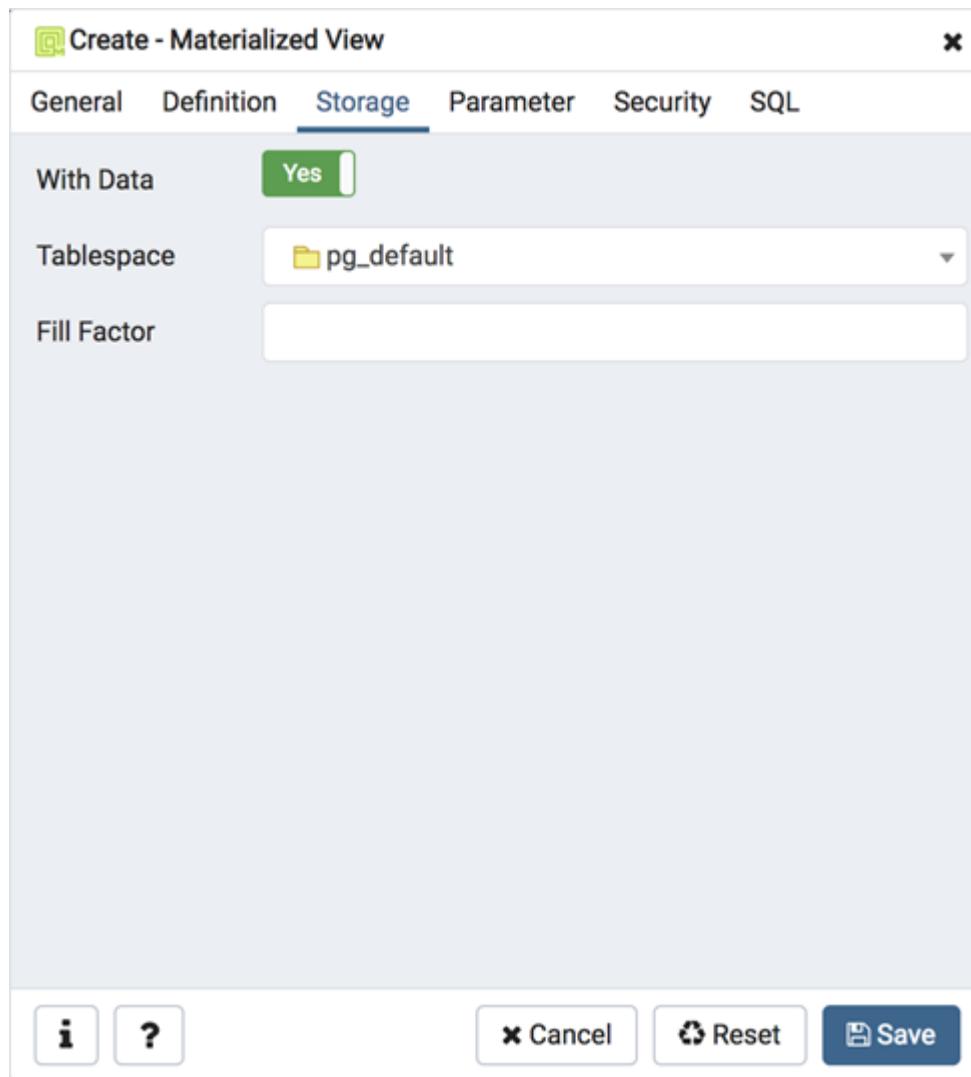
Use the drop-down listbox next to *Owner* to select the role that will own the materialized view.

Select the name of the schema in which the materialized view will reside from the drop-down listbox in the *Schema* field.

Store notes about the materialized view in the *Comment* field.

Click the *Definition* tab to continue.



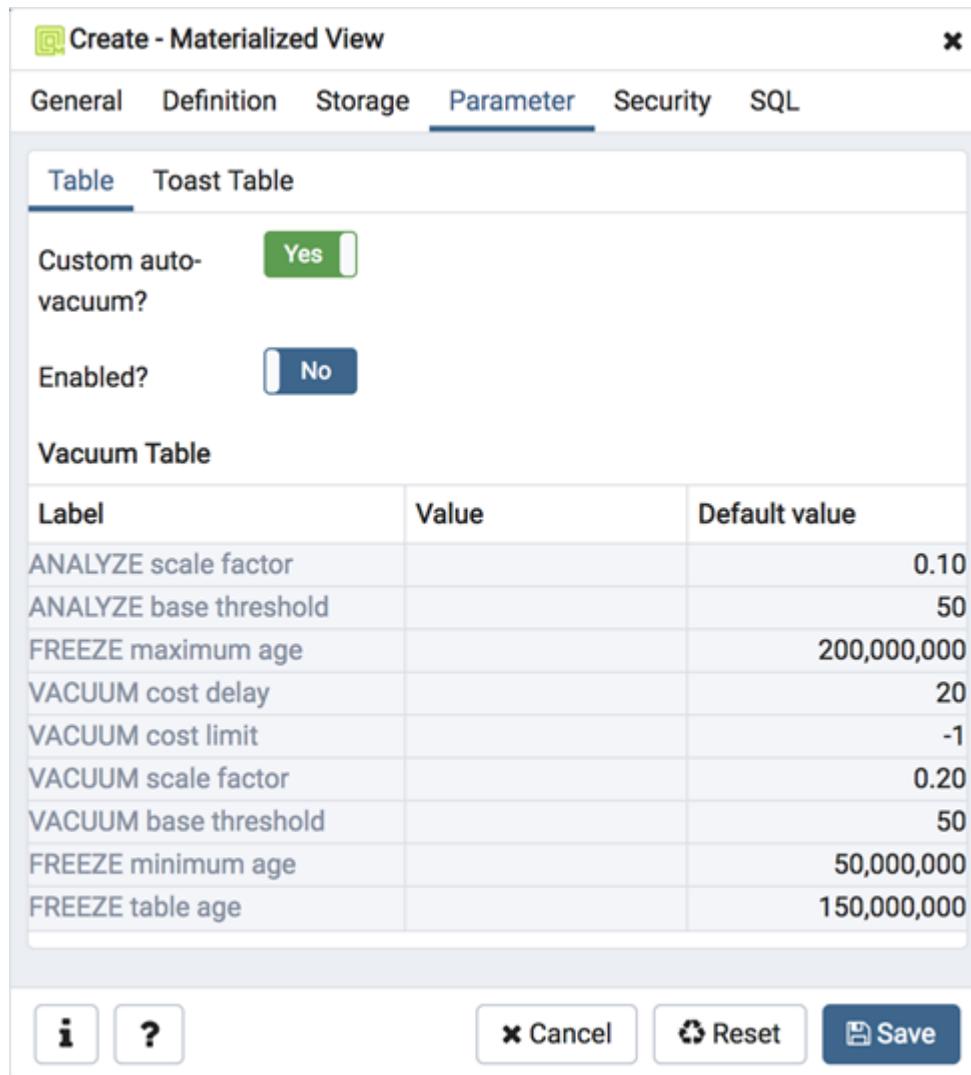


Use the text editor field in the *Definition* tab to provide the query that will populate the materialized view. Please note that updating the definition of existing materialized view would result in loss of Parameter(Table, Toast), Security(Privileges & Security labels), Indexes and other dependent objects.

Click the *Storage* tab to continue.

Use the fields in the *Storage* tab to maintain the materialized view: Move the *With Data* switch to the *Yes* position to specify the materialized view should be populated at creation time. If not, the materialized view cannot be queried until you invoke REFRESH

MATERIALIZED VIEW.

A screenshot of a PostgreSQL 'Create - Materialized View' dialog box. The 'Parameter' tab is selected. The 'Table' tab is active, showing 'Toast Table' settings. Under 'Custom auto-vacuum?', 'Yes' is selected. Under 'Enabled?', 'No' is selected. The 'Vacuum Table' section contains a table of parameters:

Label	Value	Default value
ANALYZE scale factor		0.10
ANALYZE base threshold		50
FREEZE maximum age		200,000,000
VACUUM cost delay		20
VACUUM cost limit		-1
VACUUM scale factor		0.20
VACUUM base threshold		50
FREEZE minimum age		50,000,000
FREEZE table age		150,000,000

At the bottom are buttons for 'Cancel', 'Reset', and 'Save'.

Use the drop-down listbox next to *Tablespace* to select a location for the materialized view.

Use the *Fil Factor* field to specify a fil factor for the materialized view. The fil factor for a table is a percentage between 10 and 100.

100 (complete packing) is the default.

Click the *Parameter* tab to continue.

Use the tabs nested inside the *Parameter* tab to specify VACUUM and ANALYZE thresholds; use the *Table* tab and the *Toast Table* tab

to customize values for the table and the associated toast table. To change the default values:

Move the *Custom auto-vacuum?* switch to the *Yes* position to perform custom maintenance on the materialized view.

Move the *Enabled?* switch to the *Yes* position to select values in the *Vacuum table*. Provide values for each row in the *Value* column.

Click the *Security* tab to continue.

Grantee	Privileges		Grantor
enterprisedb	<input checked="" type="checkbox"/> ALL	<input checked="" type="checkbox"/> WITH GRANT OPTION	enterprise
	<input checked="" type="checkbox"/> INSERT	<input checked="" type="checkbox"/> WITH GRANT OPTION	
	<input checked="" type="checkbox"/> SELECT	<input checked="" type="checkbox"/> WITH GRANT OPTION	
	<input checked="" type="checkbox"/> UPDATE	<input checked="" type="checkbox"/> WITH GRANT OPTION	
	<input checked="" type="checkbox"/> DELETE	<input checked="" type="checkbox"/> WITH GRANT OPTION	
	<input checked="" type="checkbox"/> TRUNCATE	<input checked="" type="checkbox"/> WITH GRANT OPTION	
	<input checked="" type="checkbox"/> REFERENCES	<input checked="" type="checkbox"/> WITH GRANT OPTION	
	<input checked="" type="checkbox"/> TRIGGER	<input checked="" type="checkbox"/> WITH GRANT OPTION	

Security Labels

Provider	Security Label

Buttons: i ? Cancel Reset Save

Use the *Security* tab to assign privileges and define security labels.

Use the *Privileges* panel to assign privileges to a role. Click the *Add* icon (+) to set privileges for the materialized view:

Select the name of the role from the drop-down listbox in the *Grantee* field.

Click inside the *Privileges* field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.

Click the *Add* icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Use the *Security Labels* panel to define security labels applied to the materialized view. Click the *Add* icon (+) to add each security label selection:

Specify a security label provider in the *Provider* field. The named provider must be loaded and must consent to the proposed labeling operation.

Specify a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL

places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the *Add* icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

The screenshot shows the 'Create - Materialized View' dialog box. At the top, there are tabs for General, Definition, Storage, Parameter, Security, and SQL. The SQL tab is selected and highlighted with a blue underline. Below the tabs, there is a large text area containing the generated SQL code:

```
1 CREATE MATERIALIZED VIEW public.new_hires
2 WITH (
3     autovacuum_enabled = false
4 )
5 TABLESPACE pg_default
6 AS
7 SELECT * FROM pg_class
8 WITH DATA;
9
10 ALTER TABLE public.new_hires
11     OWNER TO enterprisedb;
12
13 GRANT ALL ON TABLE public.new_hires TO enterprisedb WITH
```

At the bottom of the dialog, there are several buttons: an 'Info' button (i), a question mark button (?), a 'Cancel' button (x), a 'Reset' button (refresh), and a 'Save' button (disk icon).

Your entries in the *Materialized View* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Materialized View* dialog:

The example shown creates a query named *new_hires* that stores the result of the displayed query in the *pg_default* tablespace.

Click the *Info* button (i) to access online help.

Click the **Save** button to save work.

Click the **Cancel** button to exit without saving work.

Click the **Reset** button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | Development

[pgAdmin 4 documentation](#) > [Connecting To A Server](#) > [previous](#) [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
 - [Server Group Dialog](#)
 - [Server Dialog](#)
 - [Master Password](#)
 - [Connect to Server](#)
 - [Connection Error](#)
 - [Import/Export Servers](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Master Password

A master password is required to secure and later unlock the saved server passwords. This is applicable only for desktop mode users.

- You are prompted to enter the master password when you open the window for the first time after starting the application.
- Once you set the master password, all the existing saved passwords will be re-encrypted using the master password.
- The server passwords which are saved in the SQLite DB file are encrypted and decrypted using the master password.

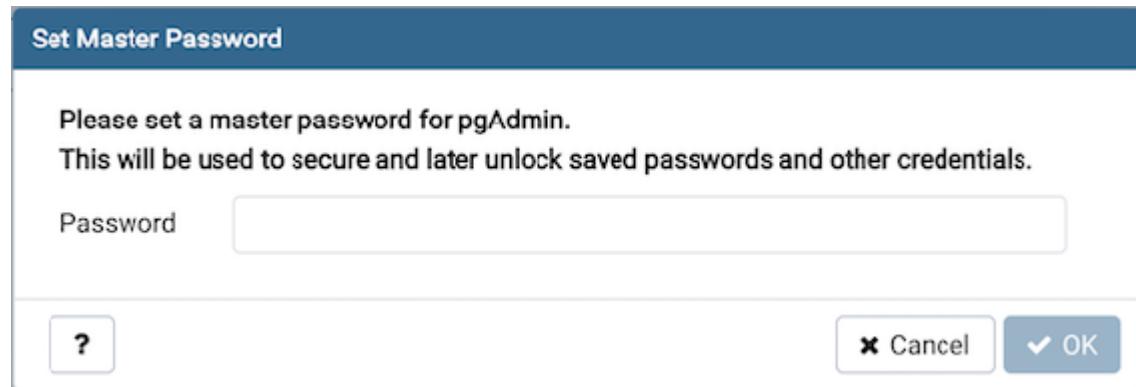
Set Master Password

Please set a master password for pgAdmin.
This will be used to secure and later unlock saved passwords and other credentials.

Password:

- You can disable the master password by setting the configuration parameter `MASTER_PASSWORD_REQUIRED=False`. See [Desktop Deployment](#) for more information.

Note



Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Connecting To A Server](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Server Group Dialog](#)

[Server Dialog](#)



[**Master Password**](#)

[FAQ](#)

[Connect to Server](#)

[Connection Error](#)



[Import/Export Servers](#)

[Managing Cluster Objects](#)

[Latest Docs](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Management Basics](#)

[Backup and Restore](#)

[Ask a](#)

[Developer Tools](#)

[Question](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)



[Release Notes](#)

[Licence](#)

[Report a Bug](#)

Master Password 



A master password is required to secure and later unlock the saved server

[Screenshots](#)

passwords. This is applicable only for desktop mode users.

You are prompted to enter the master password when you open the window for the first time after starting the application.

Once you set the master password, all the existing saved passwords will be re-encrypted using the master password.

The server passwords which are saved in the SQLite DB file are encrypted and decrypted using the master password.

You can disable the master password by setting the configuration

[parameter *MASTER_PASSWORD_REQUIRED=False*. See Desktop Deployment for more information.](#)

Note

Unlock Saved Passwords

Please enter your master password.

This is required to unlock saved passwords and reconnect to the database server(s).

Password



Reset Master Password

Cancel

OK

Reset Master Password

This will remove all the saved passwords. This will also remove established connections to the server and you may need to reconnect again. Do you wish to continue ?

No

Yes

If the master password is disabled, then all the saved passwords will be removed.

Warning

If the master password is disabled, then the saved passwords will be encrypted using a key which is derived from information within the configuration database. Use of a master password ensures that the encryption key does not need to be stored anywhere, and thus prevents possible access to server credentials if the configuration database becomes available to an attacker.

It is strongly recommended that you use the master password if you use the *Save Password* option.

The master password is not stored anywhere on the physical storage.

It is temporarily stored in the application memory and it does not get saved when the application is restarted.

You are prompted to enter the master password when pgAdmin server is restarted.

If you forget the master password, you can use the *Reset Master Password* button to reset the password.

Warning

Resetting the master password will also remove all saved passwords and close all existing established connections.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest (4.19) | 4.18 | 4.17 | Development

[pgAdmin 4 documentation](#) » [Getting Started](#) »

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Menu Bar](#)
 - [The File Menu](#)
 - [The Object Menu](#)
 - [The Tools Menu](#)
 - [The Help Menu](#)
- [Getting Started](#)
 - [Deployment](#)
 - [Login Dialog](#)
 - [User Management Dialog](#)
 - [Change User Password Dialog](#)
 - [User Interface](#)
 - [Menu Bar](#)
 - [Toolbar](#)
 - [Tabbed Browser](#)
 - [Tree Control](#)
 - [Preferences Dialog](#)
 - [Keyboard Shortcuts](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Menu Bar

The pgAdmin menu bar provides drop-down menus for access to options, commands, and utilities. The menu bar displays the following selections: *File*, *Object*, *Tools**, and *Help*. Selections may be grayed out which indicates they are disabled for the object currently selected in the pgAdmin tree control.

[The File Menu](#)

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Menu Bar](#)

[Download](#)

[The File Menu](#)

[The Object Menu](#)

[The Tools Menu](#)



[The Help Menu](#)

[FAQ](#)

[Getting Started](#)

[Deployment](#)



[Login Dialog](#)

[User Management Dialog](#)

[Latest Docs](#)

[Change User Password Dialog](#)

[User Interface](#)



Menu Bar

[Toolbar](#)

[Ask a](#)

[Tabbed Browser](#)

[Question](#)

[Tree Control](#)

[Preferences Dialog](#)



[Keyboard Shortcuts](#)

[Report a Bug](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[Screenshots](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

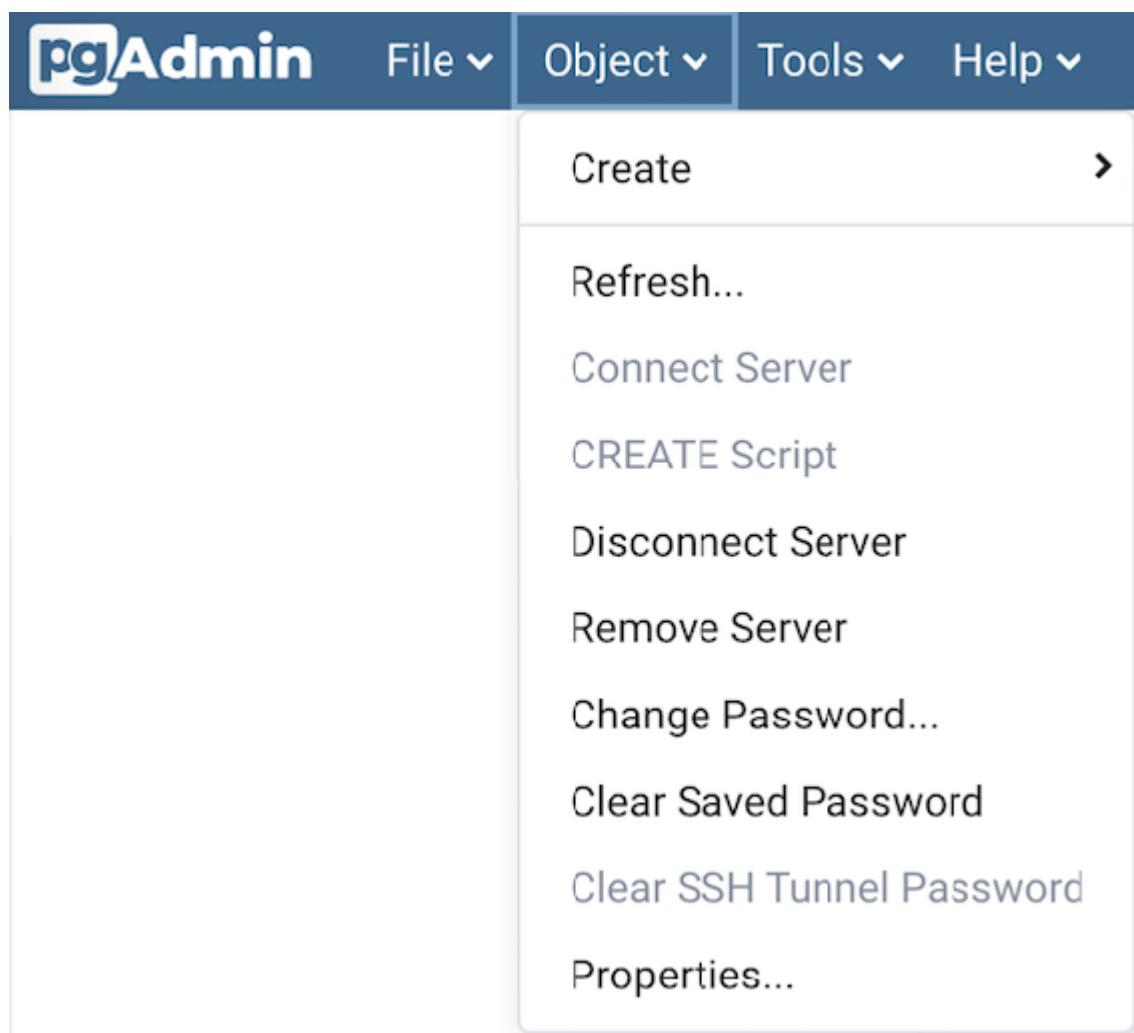
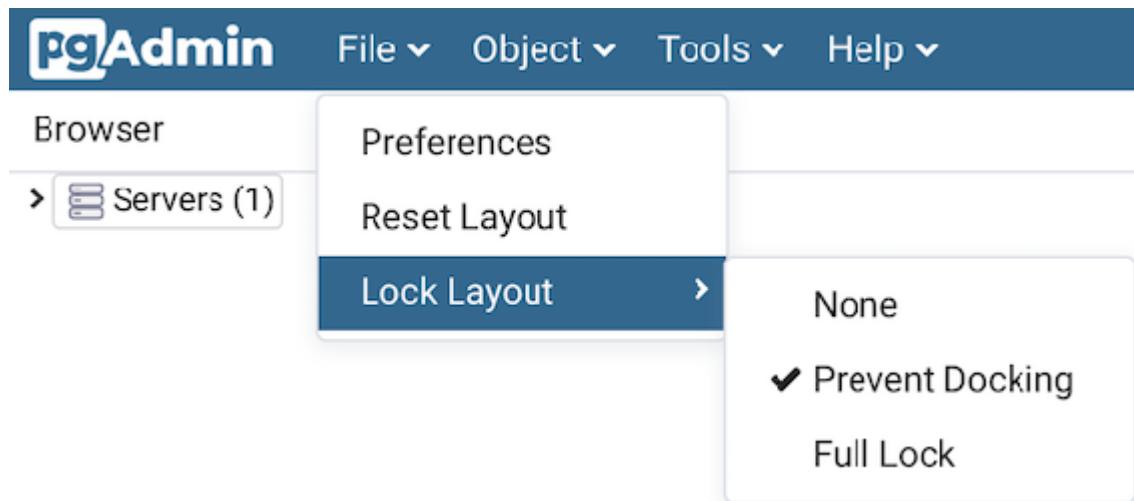
[Release Notes](#)

[Licence](#)

[Menu Bar](#) ¶

The pgAdmin menu bar provides drop-down menus for access to options, commands, and utilities. The menu bar displays the following selections: *File*, *Object*, *Tools**, and *Help*. Selections may be grayed out which indicates they are disabled for the object currently selected in the *pgAdmin* tree control.

[The File Menu](#) ¶



Use the *File* menu to access the following options: Option

Action

Preferences Click to open the [Preferences](#) dialog to customize your pgAdmin settings.

Reset Layout If you have modified the workspace, click to restore the default layout.

Lock Layout Click to open a submenu to select the level for locking the UI layout. This can also be changed from browser display

[preferences](#)

The Object Menu

The *Object* menu is context-sensitive. Use the *Object* menu to access the following options (in alphabetical order):

Option

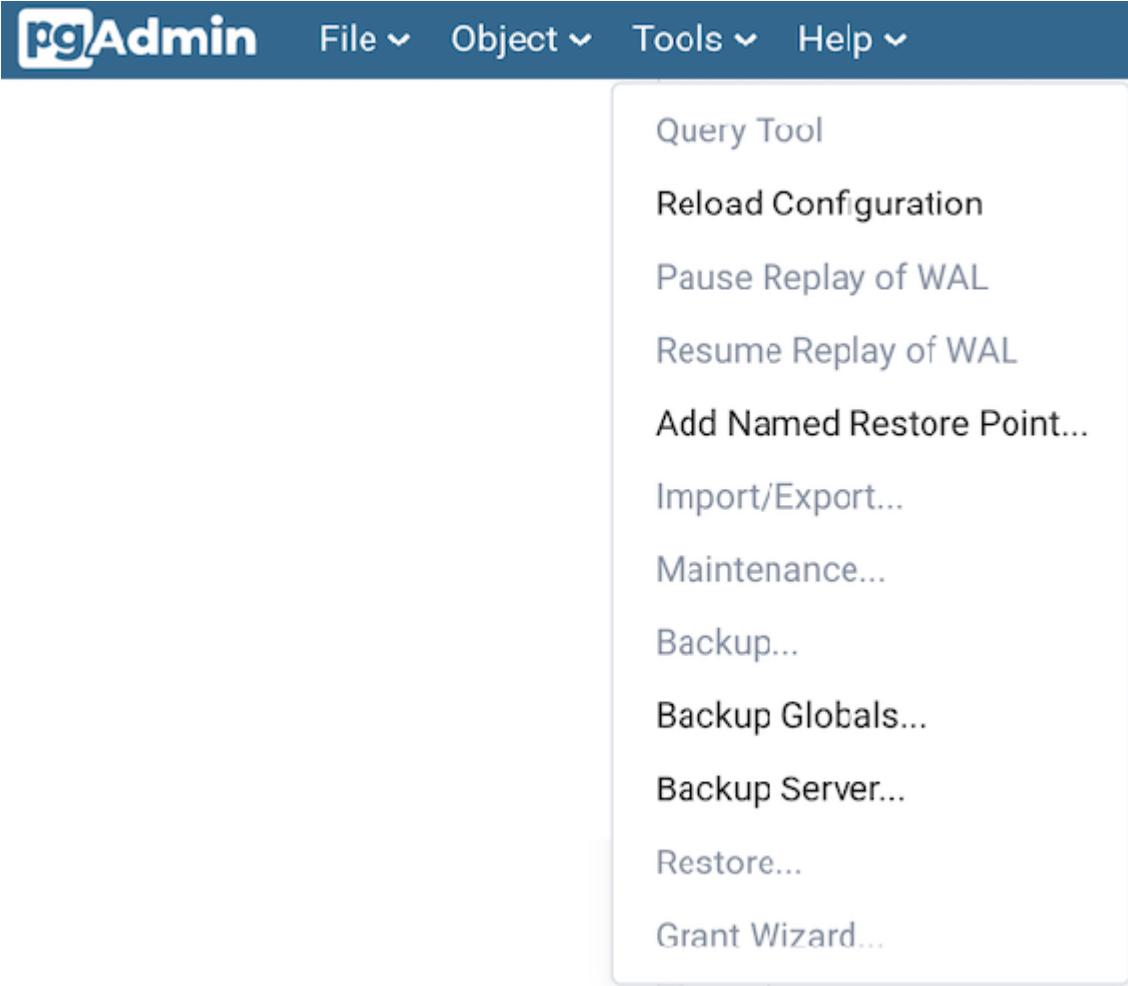
Action

Change

Click to open the [Change Password...](#) dialog to change *Password...* your password.

Clear Saved If you have saved the database server password, click to *Password*

clear the saved password. Enable only when password is already saved.



Option

Action

Clear SSH

If you have saved the ssh tunnel password, click to clear *Tunnel Password* the saved password. Enable only when password is already saved.

Connect

Click to open the [Connect to Server](#) dialog to establish a *Server...*

connection with a server.

Create

Click *Create* to access a context menu that provides context-sensitive selections. Your selection opens a

Create dialog for creating a new object.

Delete/Drop Click to delete the currently selected object from the server.

Disconnect Click to disconnect the currently selected server.

Server...

Drop

Click to delete the currently selected object and al

Cascade

dependent objects from the server.

Properties... Click to review or modify the currently selected object's properties.

Refresh...

Click to refresh the currently selected object.

Remove

Click to remove the currently selected server.

Server

Scripts

Click to open the [Query tool](#) to edit or view the selected script from the flyout menu.

Trigger(s)

Click to *Disable* or *Enable* trigger(s) for the currently selected table. Options are displayed on the flyout menu.

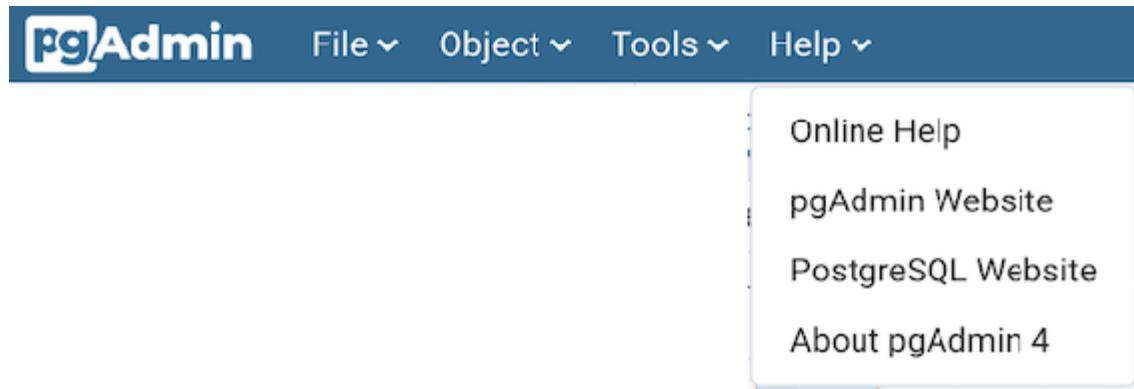
Truncate

Click to remove all rows from a table (*Truncate*) or to remove all rows from a table and its child tables (*Truncate Cascade*). Options are displayed on the flyout menu.

View Data

Click to access a context menu that provides several options for viewing data (see below).

The Tools Menu



Use the *Tools* menu to access the following options (in alphabetical order):

Option

Action

Add named

Click to open the [Add named restore point...](#) dialog to *restore point* take a point-in-time snapshot of the current server state.

Backup...

Click to open the [Backup...](#) dialog to backup database objects.

Backup

Click to open the [Backup Globals...](#) dialog to backup *Globals...* cluster objects.

Backup

Click to open the [Backup Server...](#) dialog to backup a *Server...* server.

Grant Wizard... Click to access the [Grant Wizard](#) tool.

Import/Export... Click to open the [Import/Export data...](#) dialog to import or export data from a table.

Maintenance... Click to open the [Maintenance...](#) dialog to VACUUM, ANALYZE, REINDEX, or CLUSTER.

Pause replay of Click to pause the replay of the WAL log.

WAL

Query tool

Click to open the [Query tool](#) for the currently selected object.

Reload

Click to update configuration files without restarting the *Configuration...* server.

Restore...

Click to access the [Restore](#) dialog to restore database files from a backup.

Resume replay Click to resume the replay of the WAL log.

of WAL

The Help Menu 

Use the options on the *Help* menu to access online help documents, or to review information about the pgAdmin installation (in alphabetical order): Option

Action

About

Click to open a window where you will find information

pgAdmin 4 about pgAdmin; this includes the current version and the current user.

Online Help Click to open documentation support for using pgAdmin utilities, tools and dialogs. Navigate (in the newly opened tab?) help documents in the left browser pane or use the search bar to specify a topic.

Option

Action

pgAdmin

Click to open the *pgAdmin.org* website in a browser *Website*

window.

PostgreSQL Click to access the PostgreSQL core documentation hosted *Website*

at the PostgreSQL site. The site also offers guides, tutorials, and resources.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Package Dialog

Use the Package dialog to create a (user-defined) package specification.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Cast Dialog](#)

[FAQ](#)

[Collation Dialog](#)

[Domain Dialog](#)

[□](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)

[Latest Docs](#)

[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[□](#)

[Foreign Server Dialog](#)

[Foreign Table Dialog](#)

[Ask a](#)

[FTS Configuration Dialog](#)

[Question](#)

[FTS Dictionary Dialog](#)

[FTS Parser Dialog](#)

[□](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[Report a Bug](#)

[Language Dialog](#)

[Materialized View Dialog](#)



[Package Dialog](#)

[Procedure Dialog](#)

[Screenshots](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)

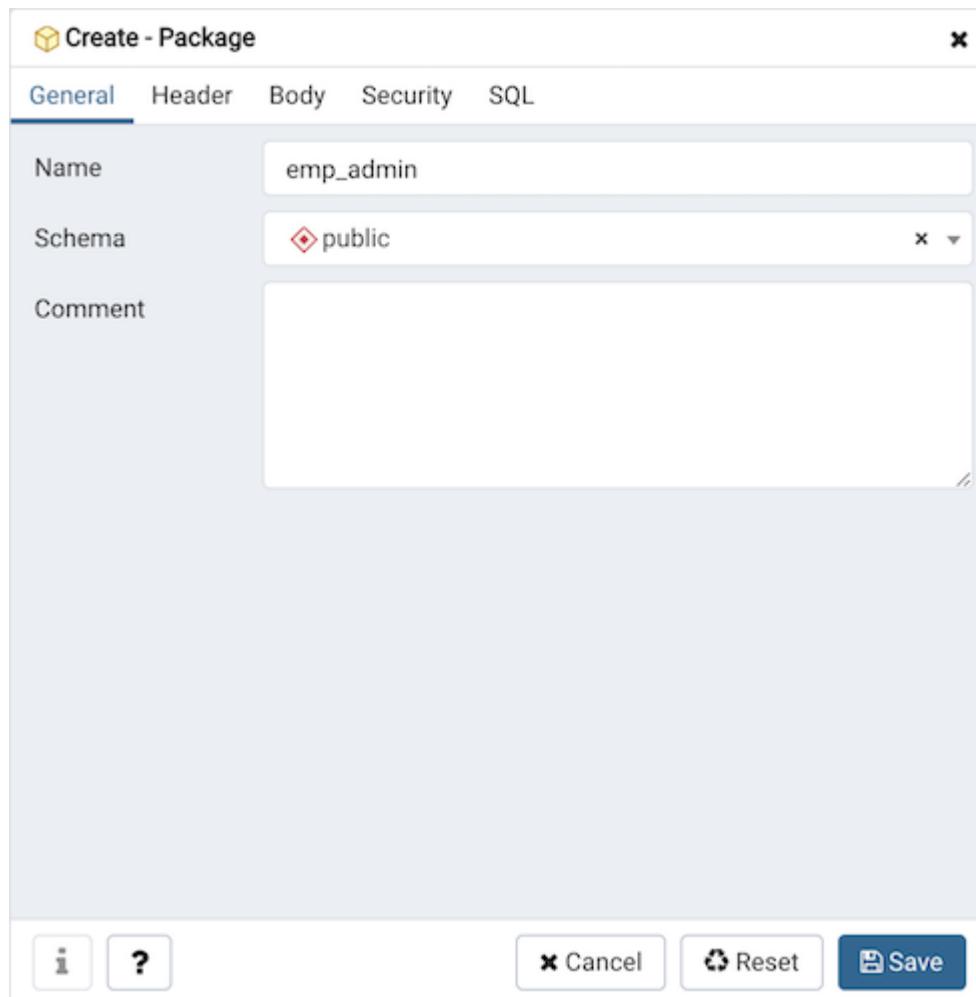
[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Package Dialog

Use the *Package* dialog to create a (user-defined) package specification.



```
1 FUNCTION get_dept_name(p_deptno numeric)
2 RETURN character varying;
3
4 FUNCTION update_emp_sal(p_empno numeric, p_raise numeric)
5 RETURN numeric;
6
7 PROCEDURE hire_emp(p_empno numeric,
8                     p_ename character varying,
9                     p_job character varying,
10                    p_sal numeric,
11                    p_hiredate timestamp without time zone,
12                    p_comm numeric,
13                    p_mgr numeric,
14                    p_deptno numeric);
15
16 PROCEDURE fire_emp(p_empno numeric);
```

The *Package* dialog organizes the management of a package through the following dialog tabs: *General*, *Header*, *Body*, and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

Use the fields in the *General* tab to identify the package: Use the *Name* field to add a descriptive name for the package. The name of a new package must not match any existing package in the same schema.

Select the schema in which the package will reside from the drop-down listbox in the *Schema* field.

Store notes about the package in the *Comment* field.

Click the *Header* tab to continue.

Use the *Header* field to define the public interface for the package.

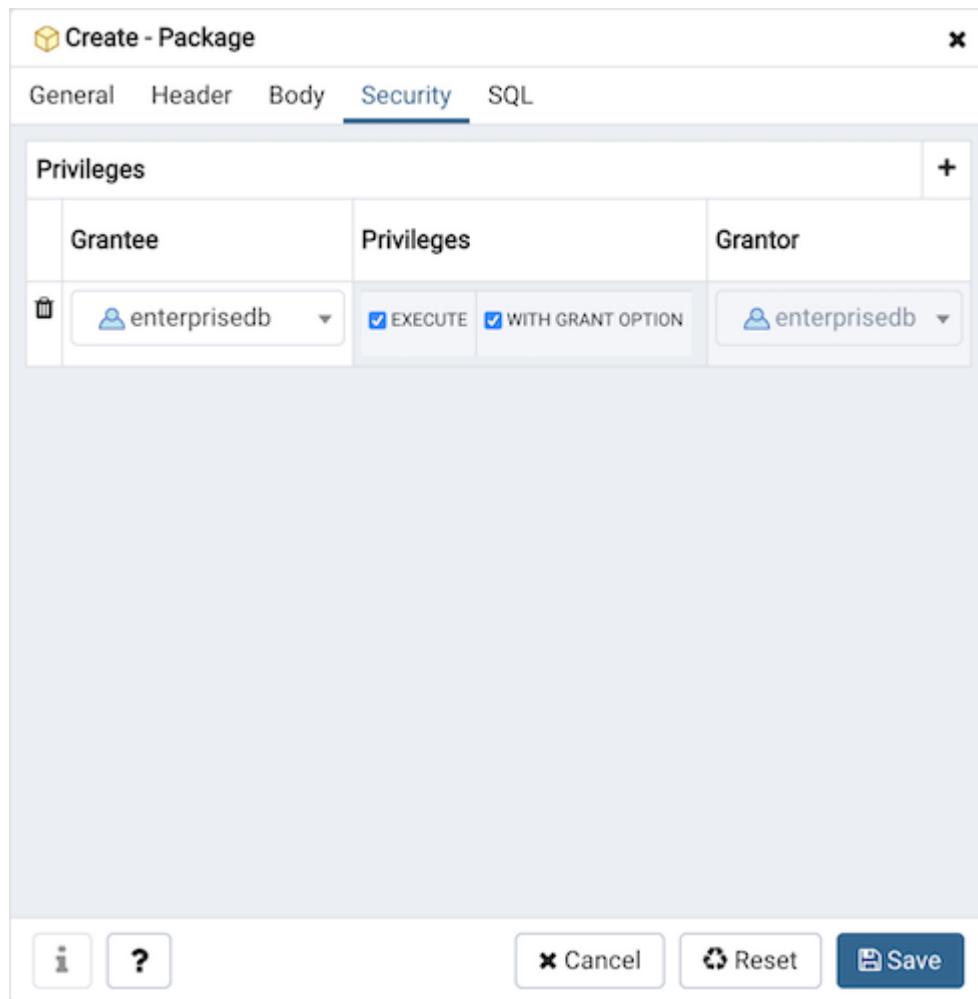
Click the *Body* tab to continue.

Create - Package

General Header Body Security SQL

```
1 --
2 -- Function that queries the 'dept' table based on the dep-
3 -- number and returns the corresponding department name.
4 --
5 FUNCTION get_dept_name(p_deptno numeric) RETURN character \/
6     v_dname          VARCHAR2(14);
7 BEGIN
8     SELECT dname INTO v_dname FROM dept WHERE deptno = p_de-
9         RTURE v_dname;
10 EXCEPTION
11     WHEN NO_DATA_FOUND THEN
12         DBMS_OUTPUT.PUT_LINE('Invalid department number ' |
13             RETURN '';
14 END;
15 --
16 -- Function that updates an employee's salary based on the
17 -- employee number and salary increment/decrement passed
18 -- as IN parameters. Upon successful completion the funct-
19 -- returns the new updated salary.
20 --
21 FUNCTION update_emp_sal(p_emppno numeric, p_raise numeric) R
```

Cancel Reset Save



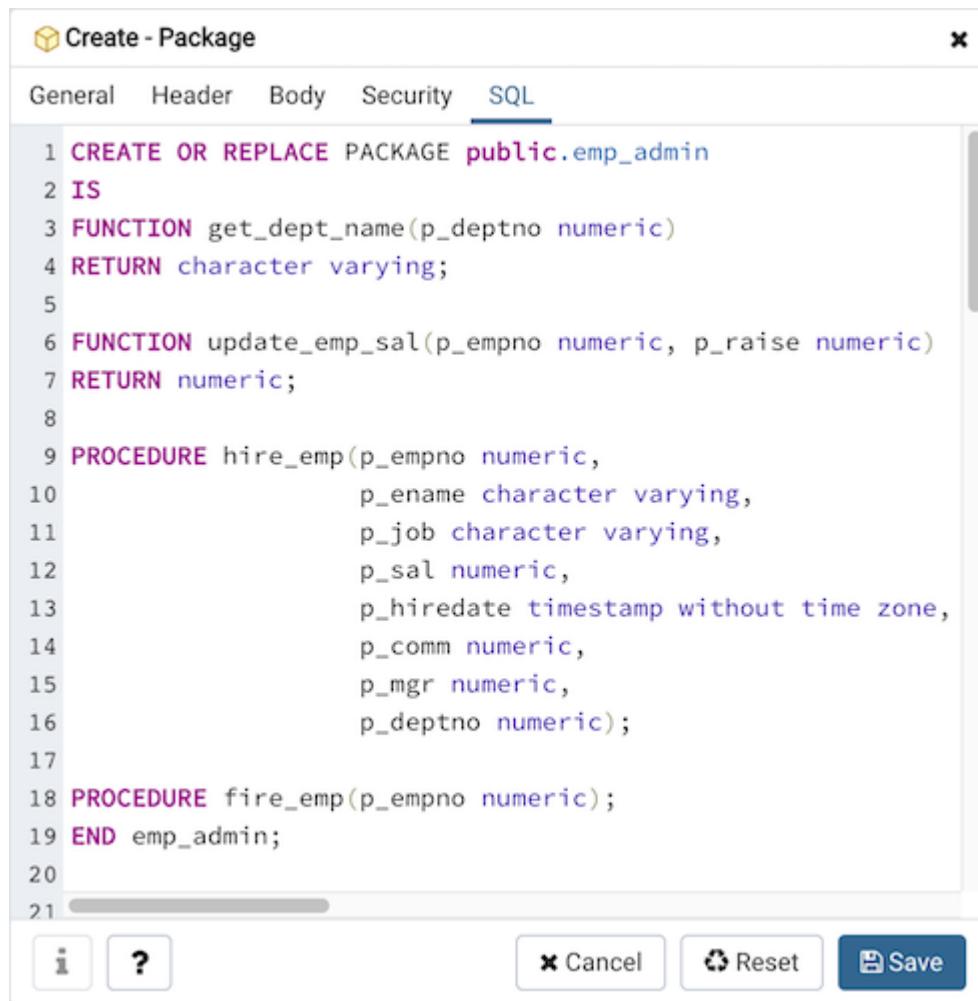
Use the *Body* field to provide the code that implements each package object.

Click the *Security* tab to continue.

Use the fields in the *Security* tab to assign EXECUTE privileges for the package to a role. Click the *Add* icon (+) to set privileges for the package: Select the name of the role from the drop-down listbox in the *Grantee* field.

Click inside the *Privileges* field. Check the boxes to the left of a privilege to grant the selected privilege to the specified user.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.

A screenshot of the Oracle Database 'Create - Package' dialog box. The title bar says 'Create - Package'. Below it is a navigation bar with tabs: General, Header, Body, Security, and SQL. The SQL tab is selected and highlighted in blue. The main area contains the following PL/SQL code:

```
1 CREATE OR REPLACE PACKAGE public.emp_admin
2 IS
3     FUNCTION get_dept_name(p_deptno numeric)
4     RETURN character varying;
5
6     FUNCTION update_emp_sal(p_empno numeric, p_raise numeric)
7     RETURN numeric;
8
9     PROCEDURE hire_emp(p_empno numeric,
10                         p_ename character varying,
11                         p_job character varying,
12                         p_sal numeric,
13                         p_hiredate timestamp without time zone,
14                         p_comm numeric,
15                         p_mgr numeric,
16                         p_deptno numeric);
17
18     PROCEDURE fire_emp(p_empno numeric);
19 END emp_admin;
20
21
```

At the bottom of the dialog are three buttons: an info icon, a question mark icon, a 'Cancel' button with a trash icon, a 'Reset' button with a recycle bin icon, and a 'Save' button.

Click the *Add* icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row, and confirm the deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

Your entries in the *Package* dialog generate a SQL command that creates or modifies a package definition:

The example shown demonstrates creating a package named *empinfo* that includes two function and two procedure.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to delete any changes to the dialog.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Managing Cluster Objects](#) > [previous](#) [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Move Objects Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
 - [Database Dialog](#)
 - [Move Objects Dialog](#)
 - [Resource Group Dialog](#)
 - [Login/Group/Role Dialog](#)
 - [Tablespace Dialog](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Move Objects Dialog

Use the Move Objects dialog to move database objects from one tablespace to another tablespace.

The Move Objects dialog organizes the movement of database objects with the General tab; the SQL tab displays the SQL code generated by dialog selections.



Use the fields in the General tab to identify the items that will be moved and the tablespace to which they will be moved:

- Use the New tablespace drop-down listbox to select a pre-existing tablespace to which the object will be moved. (To create a tablespace, use the Tablespace dialog; access the dialog by right



Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Cluster Objects](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



Move Objects Dialog

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Database Dialog](#)

Move Objects Dialog



[Resource Group Dialog](#)

[Login/Group Role Dialog](#)

[Latest Docs](#)

[Tablespace Dialog](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[Ask a](#)

[Management Basics](#)

[Question](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)



[pgAdmin Project Contributions](#)

[Report a Bug](#)

[Release Notes](#)

[Licence](#)



Move Objects Dialog

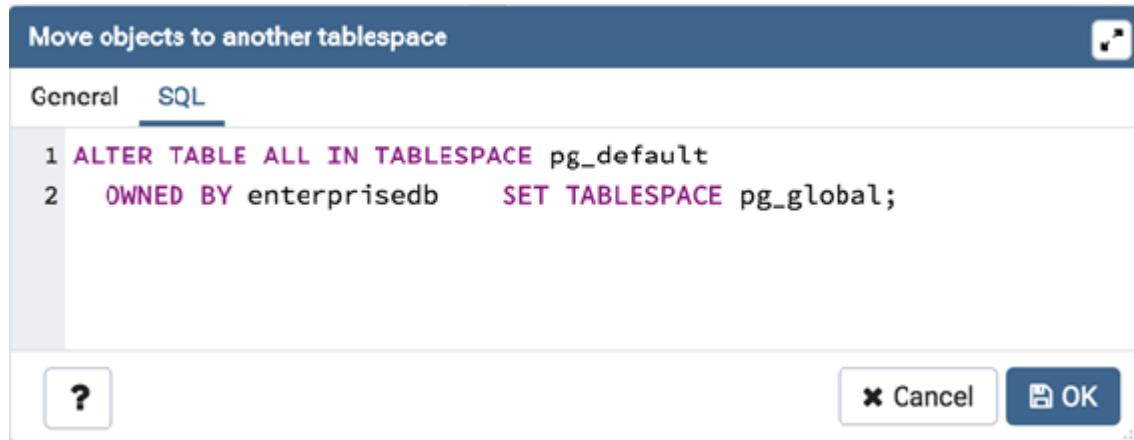
[Screenshots](#)

Use the *Move Objects* dialog to move database objects from one tablespace to another tablespace.

The *Move Objects* dialog organizes the movement of database objects with the *General* tab; the *SQL* tab displays the SQL code generated by dialog selections.

Use the fields in the *General* tab to identify the items that will be moved and the tablespace to which they will be moved:

Use the *New tablespace* drop-down listbox to select a pre-existing tablespace to which the object will be moved. (To create a tablespace, use the *Tablespace* dialog; access the dialog by right



clicking *Tablespaces* in the *pgAdmin* tree control and selecting *Create Tablespace...* from the context-menu.)

Use the *Object type* drop-down listbox to select from the following:
Select *All* to move all tables, indexes, and materialized views from the current tablespace (currently selected in the *pgAdmin* tree control) to the new tablespace.

Select *Tables* to move tables from the current tablespace to the new tablespace.

Select *Indexes* to move indexes from the current tablespace to the new tablespace.

Select *Materialized views* to move materialized views from the current tablespace to the new tablespace.

Use the *Object owner* drop-down listbox to select the role that owns the objects selected in the *Object type* field. This field is optional.

Click the *SQL* tab to continue.

Your entries in the *Move Objects* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit the *General* tab to modify the SQL command.

Example 

The following is an example of the sql command generated by user selections in the *Move Objects* dialog:

The example shown demonstrates moving materialized views owned by Alice from tablespace *tbspace_01* to *tbspace_02*.

Click the *Help* button (?) to access online help.

Click the *OK* button to save work.

Click the *Cancel* button to exit without saving work.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 4.19 documentation](#) > [Getting Started](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Preferences Dialog](#)
 - [The Browser Node](#)
 - [The Dashboards Node](#)
 - [The Debugger Node](#)
 - [The Miscellaneous Node](#)
 - [The Paths Node](#)
 - [The Query Tool Node](#)
 - [The Schema Diff Node](#)
 - [The Storage Node](#)
- [Getting Started](#)
 - [Deployment](#)
 - [Login Dialog](#)
 - [User Management Dialog](#)
 - [Change User Password Dialog](#)
 - [User Interface](#)
 - [Menu Bar](#)
 - [Toolbar](#)
 - [Tabbed Browser](#)
 - [Tree Control](#)
 - [Preferences Dialog](#)
 - [Keyboard Shortcuts](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Preferences Dialog

Use options on the Preferences dialog to customize the behavior of the client. To open the Preferences dialog, select *Preferences* from the *File* menu. The left pane of the Preferences dialog displays a tree control; each node of the tree control provides access to options that are related to the node under which they are displayed.

- Use the plus sign (+) to the left of a node name to expand a segment of the tree control.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#)
[Development](#)

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Preferences Dialog](#)

[Download](#)

[The Browser Node](#)

[The Dashboards Node](#)

[The Debugger Node](#)



[The Miscellaneous Node](#)

[FAQ](#)

[The Paths Node](#)

[The Query Tool Node](#)



[The Schema Diff Node](#)

[The Storage Node](#)

[Latest Docs](#)

[Getting Started](#)

[Deployment](#)



[Login Dialog](#)

[User Management Dialog](#)

[Ask a](#)

[Change User Password Dialog](#)

[Question](#)

[User Interface](#)

[Menu Bar](#)



[Toolbar](#)

[Report a Bug](#)

[Tabbed Browser](#)

[Tree Control](#)

Preferences Dialog



[Keyboard Shortcuts](#)

[Screenshots](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

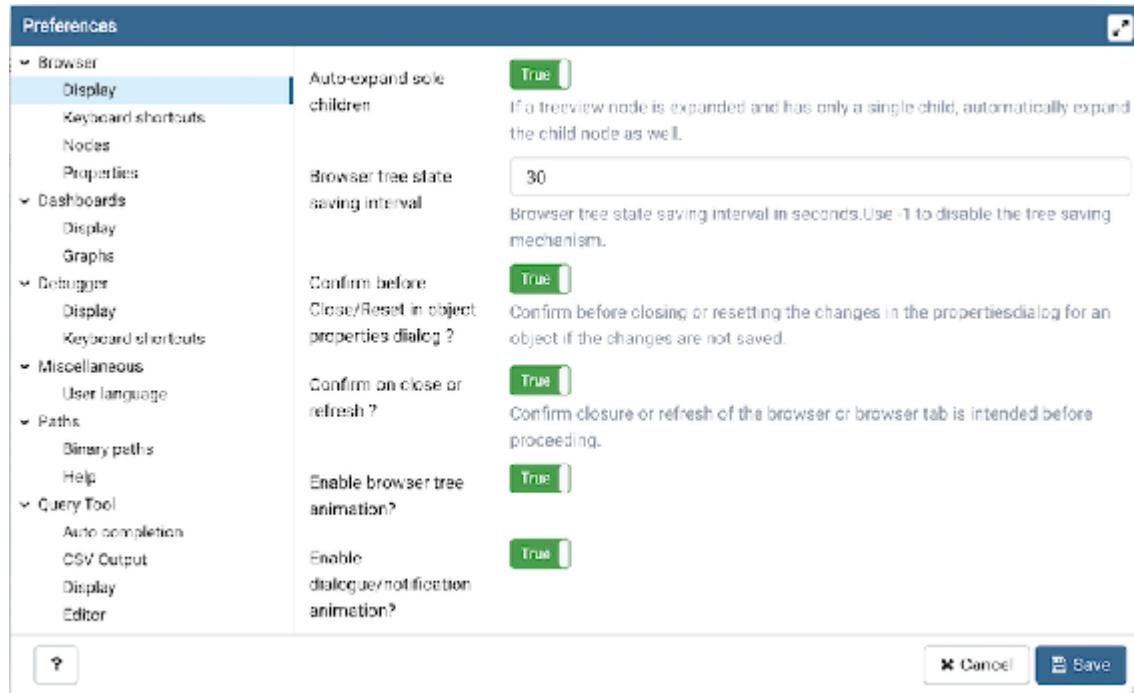
[Release Notes](#)

[Licence](#)

Preferences Dialog 

Use options on the *Preferences* dialog to customize the behavior of the client. To open the *Preferences* dialog, select *Preferences* from the *File* menu. The left pane of the *Preferences* dialog displays a tree control; each node of the tree control provides access to options that are related to the node under which they are displayed.

Use the plus sign (+) to the left of a node name to expand a segment of the tree control.



Use the minus sign (-) to the left of a node name to close that node.

The Browser Node

Use preferences found in the *Browser* node of the tree control to personalize your workspace.

Use the fields on the *Display* panel to specify general display preferences: When the *Auto-expand sole children* switch is set to *True*, child nodes will be automatically expanded if a treeview node is expanded and has only a single child.

Use the *Browser tree state saving interval* field to set the treeview state saving interval. A value of -1 will disable the treeview state saving functionality.

When the *Confirm before closing properties with unsaved changes* switch is set to *True*, pgAdmin will warn you before closing the properties dialog of an object if there are any unsaved changes. On user confirmation, the properties dialog will close.

When the *Confirm on close or refresh* switch is set to *True*, pgAdmin will attempt to catch browser close or refresh events and prompt before allowing them to continue.

When the *Enable browser tree animation?* switch is set to *True*, the client will display the animated tree control otherwise it will be unanimated.

When the *Enable dialogue/notification animation?* switch is set to *True*, the client will display the animated dialogues/notifications otherwise it will be unanimated.

Use the *Lock layout* field to lock the UI layout at different levels. This can also be changed from File menu on the [menu bar](#)

Option

Action

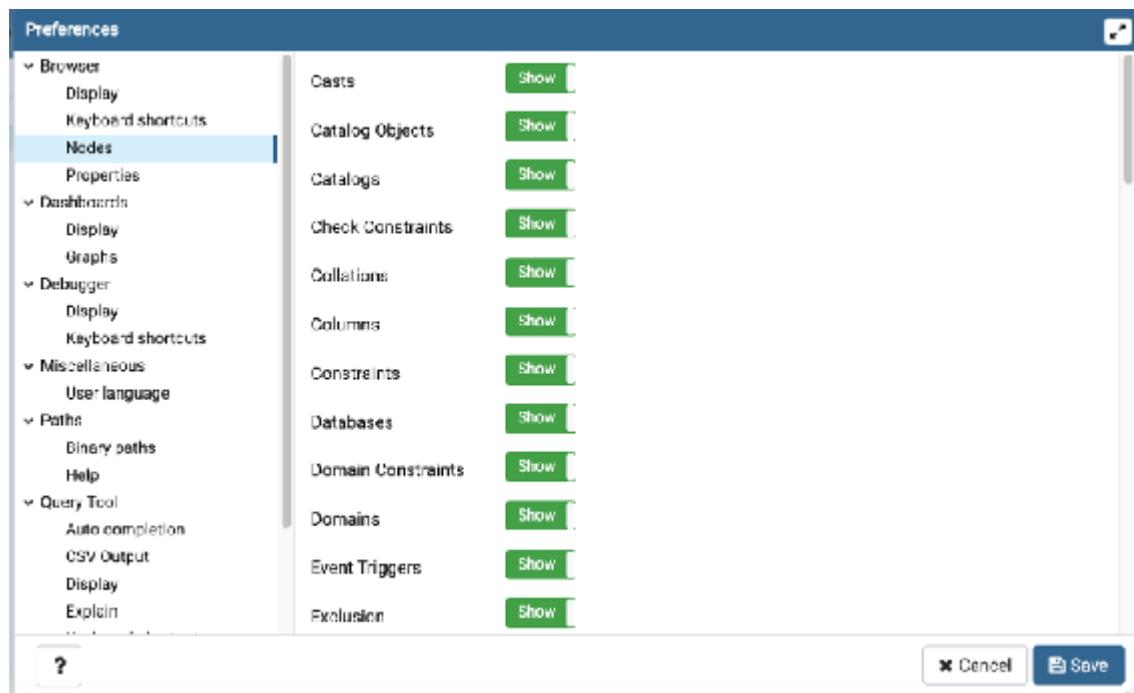
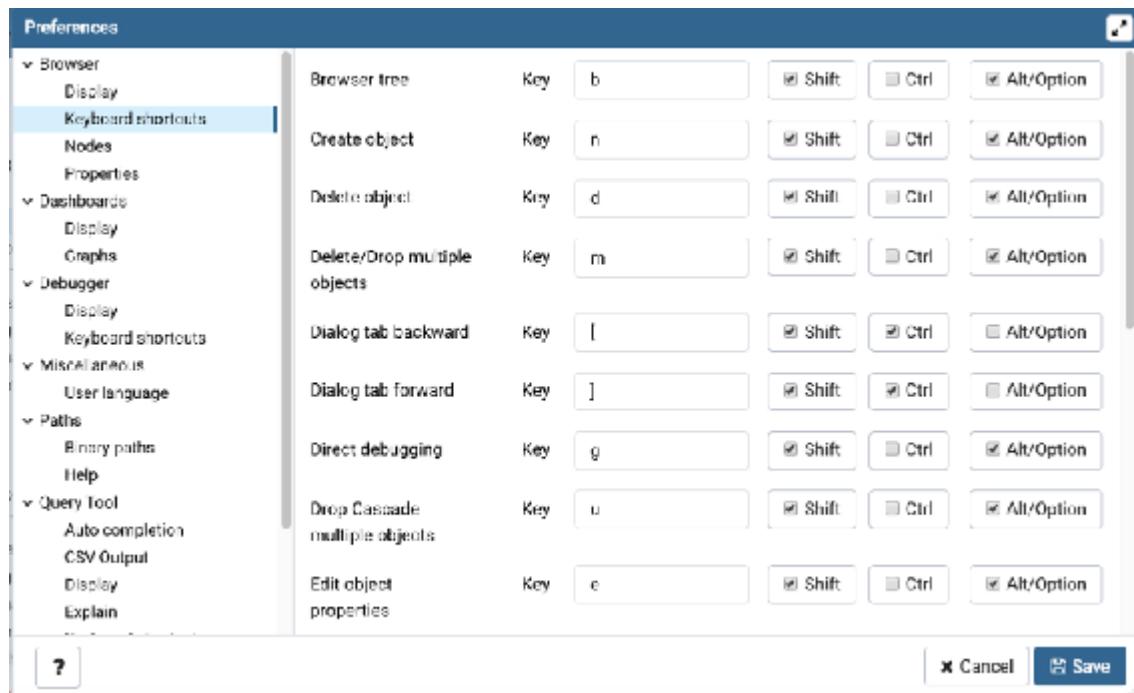
None

No locking. Every panel is resizable and dockable.

Prevent docking This will disable the docking/undocking of the panels
Ful

This will disable resizing, docking/undocking of the panels

When the *Show system objects?* switch is set to *True*, the client will display system objects such as system schemas (for example,



pg_temp) or system columns (for example, *xmin* or *ctid*) in the tree control.

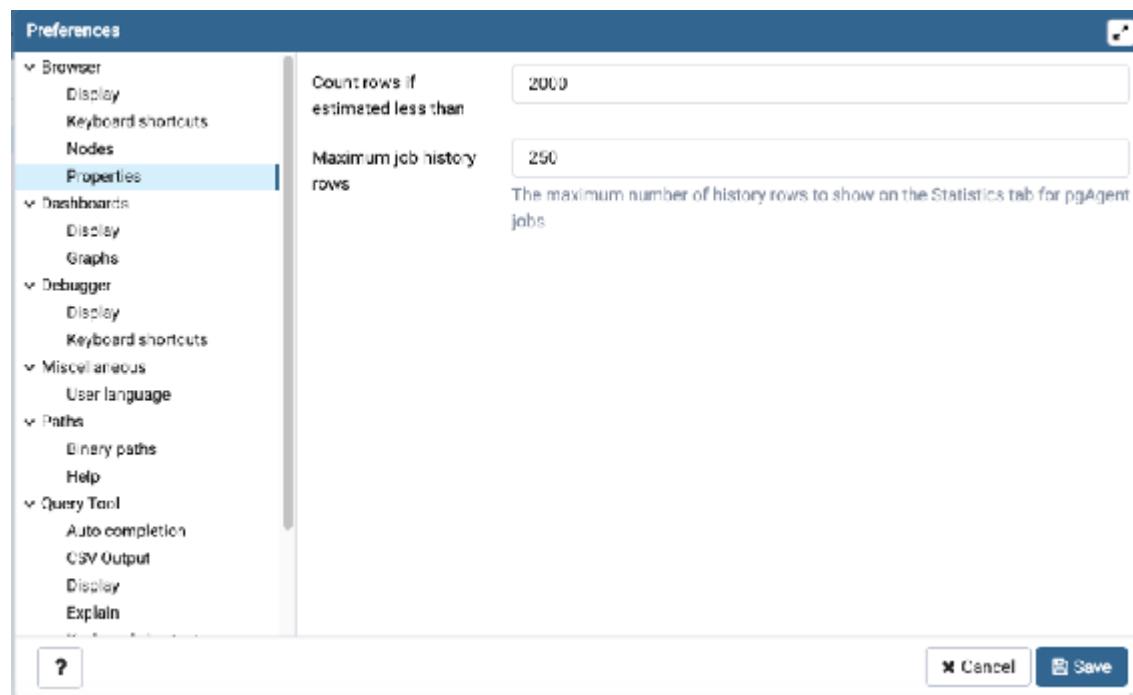
Use the fields on the *Keyboard shortcuts* panel to configure shortcuts for the main window navigation:

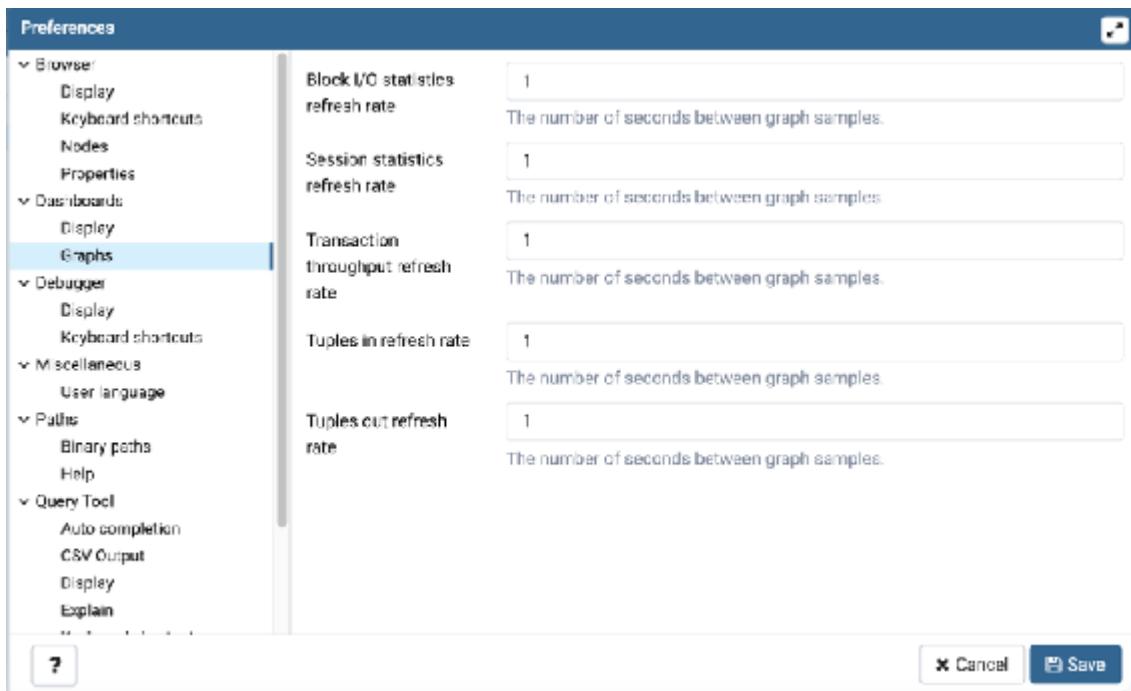
The panel displays a list of keyboard shortcuts available for the main window; select the combination of the modifier keys along with the key to configure each shortcut.

Use the fields on the *Nodes* panel to select the object types that will be displayed in the *Browser* tree control:

The panel displays a list of database objects; slide the switch located next to each object to *Show* or *Hide* the database object. When querying system catalogs, you can reduce the number of object types displayed to increase speed.

Use fields on the *Properties* panel to specify browser properties:





Include a value in the *Count rows if estimated less than* field to perform a SELECT count(*) if the estimated number of rows in a table (as read from the table statistics) is below the specified limit. After performing the SELECT count(*), pgAdmin will display the row count.

The default is 2000.

Provide a value in the *Maximum job history rows* field to limit the number of rows to show on the statistics tab for pgAgent jobs. The default is 250.

The Dashboards Node 

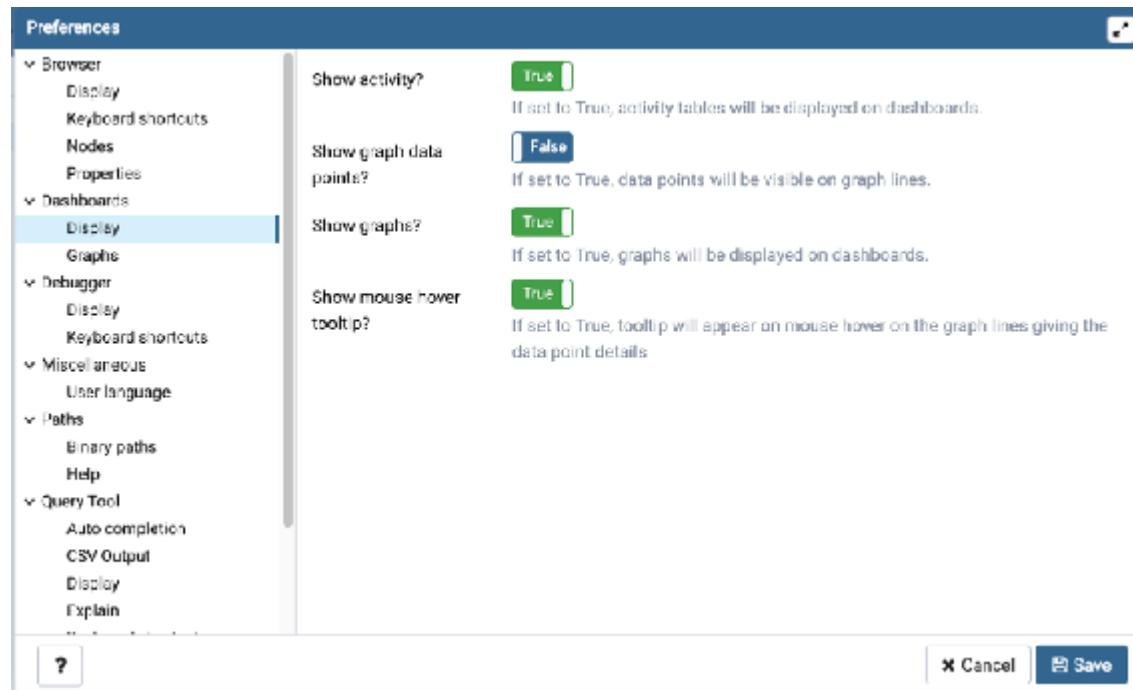
Expand the *Dashboards* node to specify your dashboard display preferences.

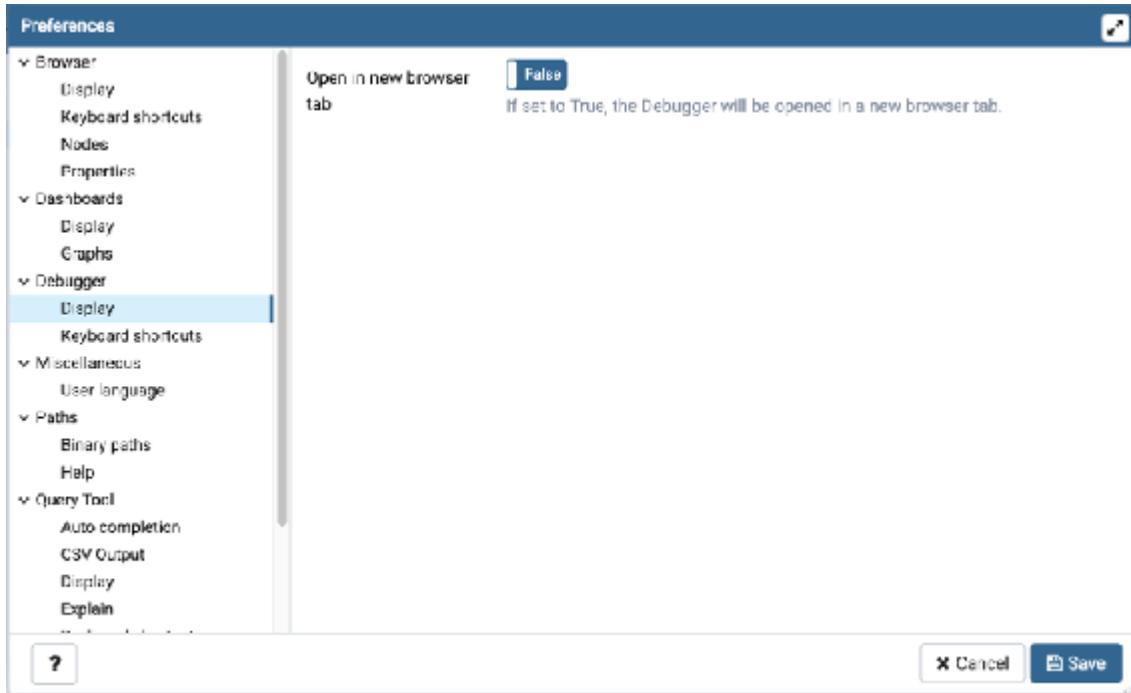
Use the fields on the *Graphs* panel to specify your display preferences for the graphs on the *Dashboard* tab:

Use the *Block I/O statistics refresh rate* field to specify the number of seconds between block I/O statistic samples displayed in graphs.

Use the *Session statistics refresh rate* field to specify the number of seconds between session statistic samples displayed in graphs.

Use the *Transaction throughput refresh rate* field to specify the number of seconds between transaction throughput samples displayed in graphs.





Use the *Tuples in refresh rate* field to specify the number of seconds between tuples-in samples displayed in graphs.

Use the *Tuples out refresh rate* field to specify the number of seconds between tuples-out samples displayed in graphs.

When the *Show activity?* switch is set to *True*, activity tables wil be displayed on dashboards.

When the *Show graph data points?* switch is set to *True*, data points wil be visible on graph lines.

When the *Show graphs?* switch is set to *True*, graphs wil be displayed on dashboards.

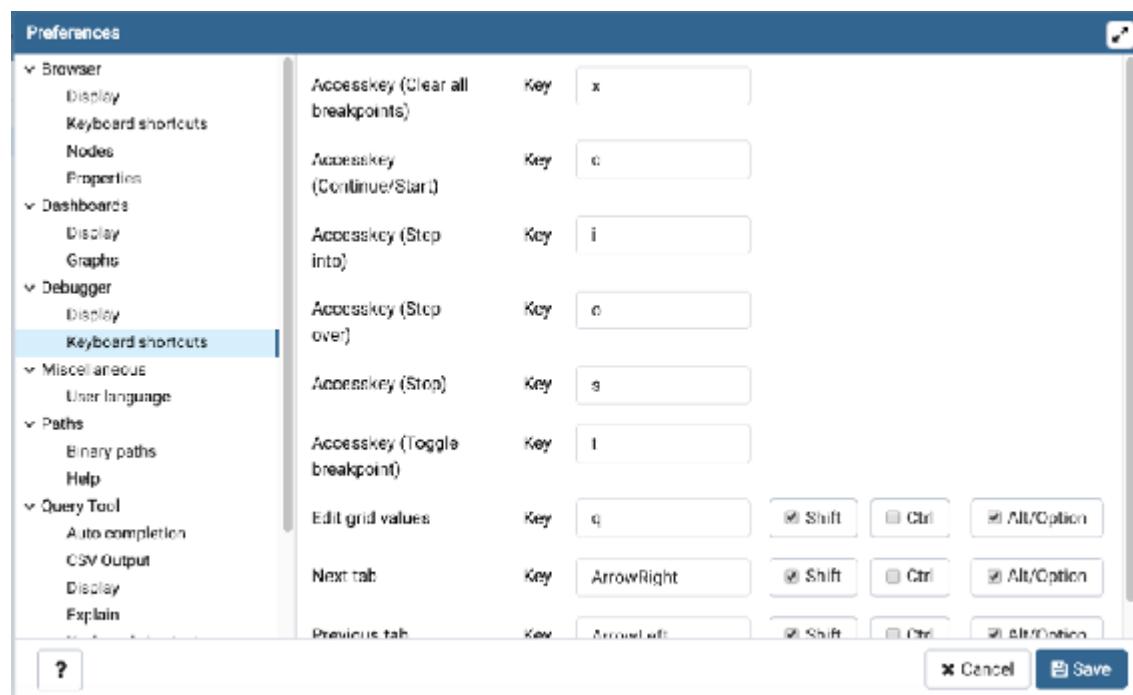
When the *Show mouse hover tooltip?* switch is set to *True*, a tooltip wil appear on mouse hover on the graph lines giving the data point details.

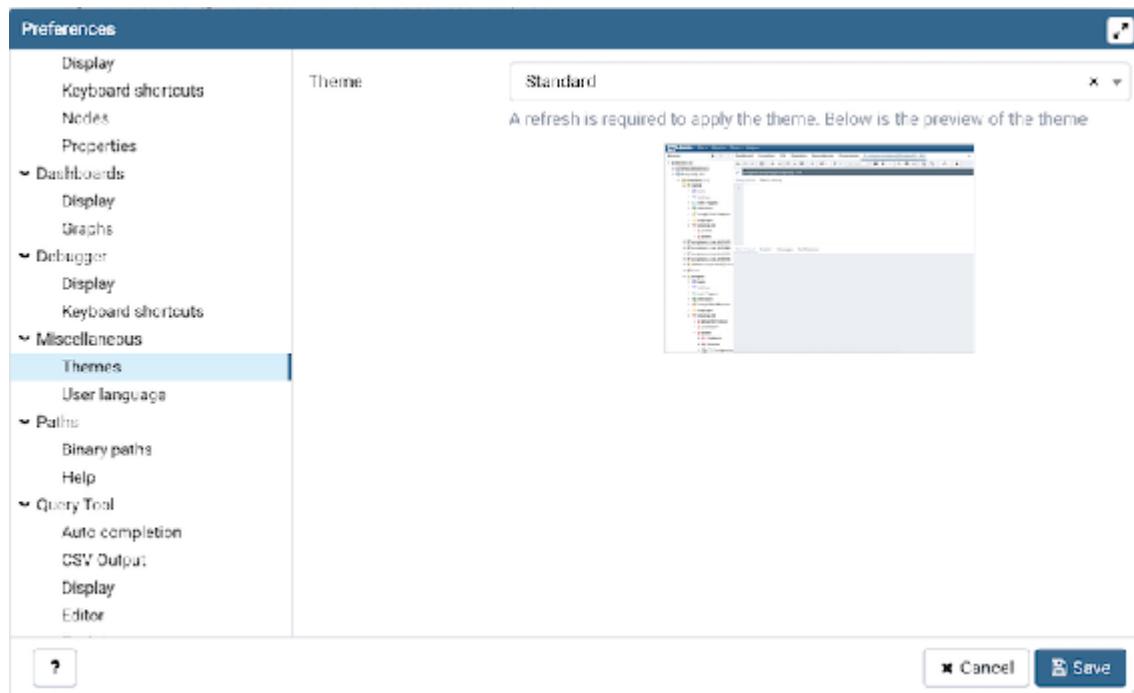
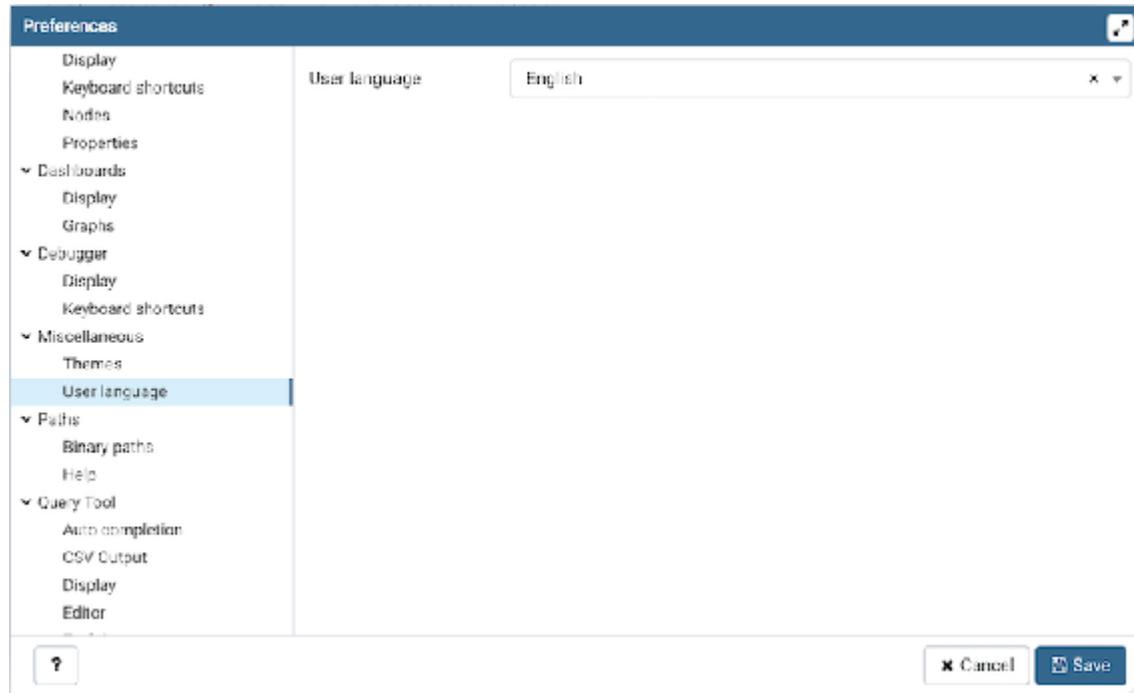
The Debugger Node ¶

Expand the *Debugger* node to specify your debugger display preferences.

When the *Open in new browser tab* switch is set to *True*, the Debugger will open in a new browser tab when invoked.

Use the fields on the *Keyboard shortcuts* panel to configure shortcuts for the debugger window navigation:





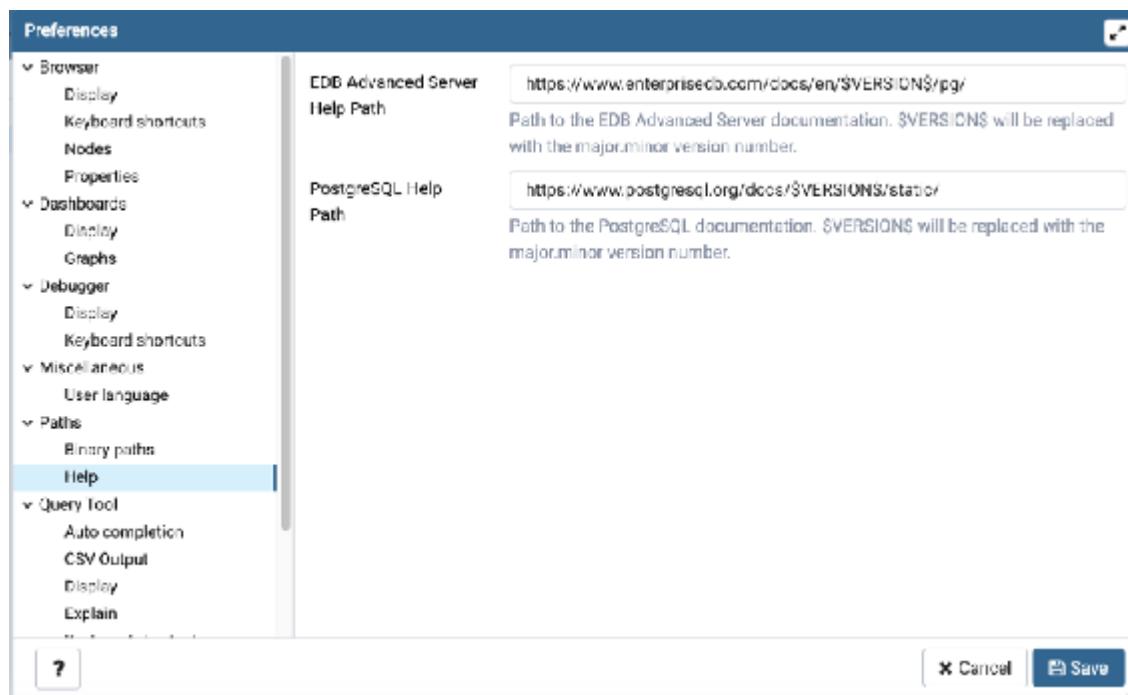
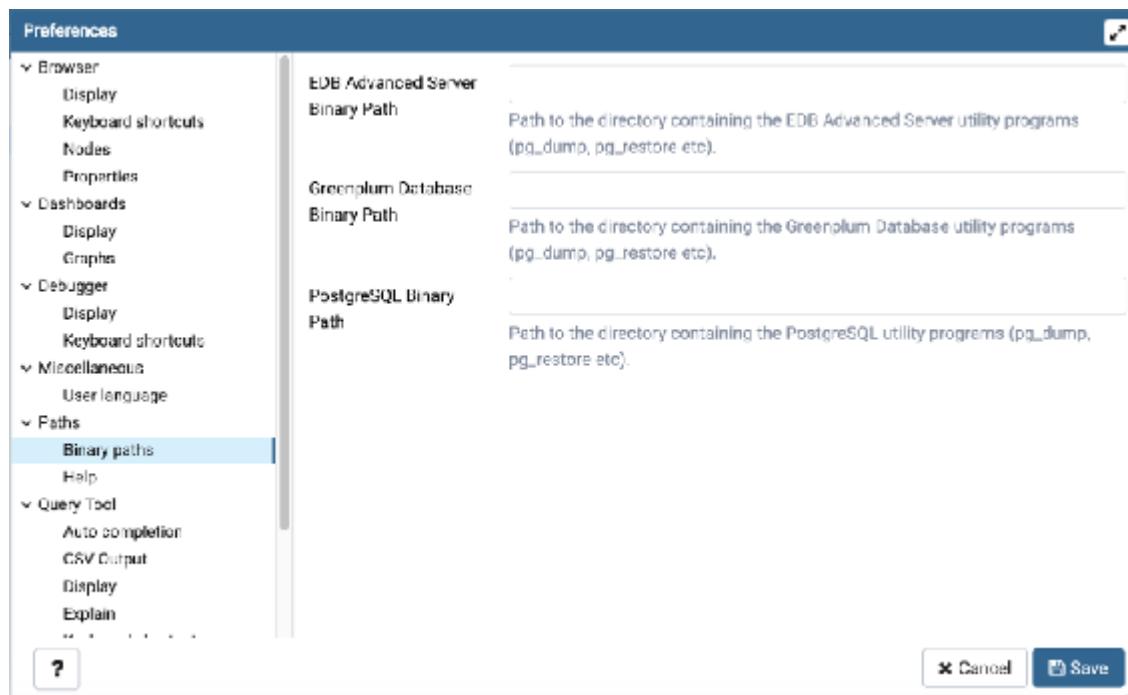
The Miscellaneous Node

Expand the *Miscellaneous* node to specify miscellaneous display preferences.

Use the *User language* drop-down listbox to select the display language for the client.

Use the *Themes* drop-down listbox to select the theme for pgAdmin.

You'll also get a preview just below the drop down. Note that, to apply



the theme you need to refresh the pgAdmin page. You can also submit your own themes, check [here](#) how.

The Paths Node

Expand the *Paths* node to specify the locations of supporting utility and help files.

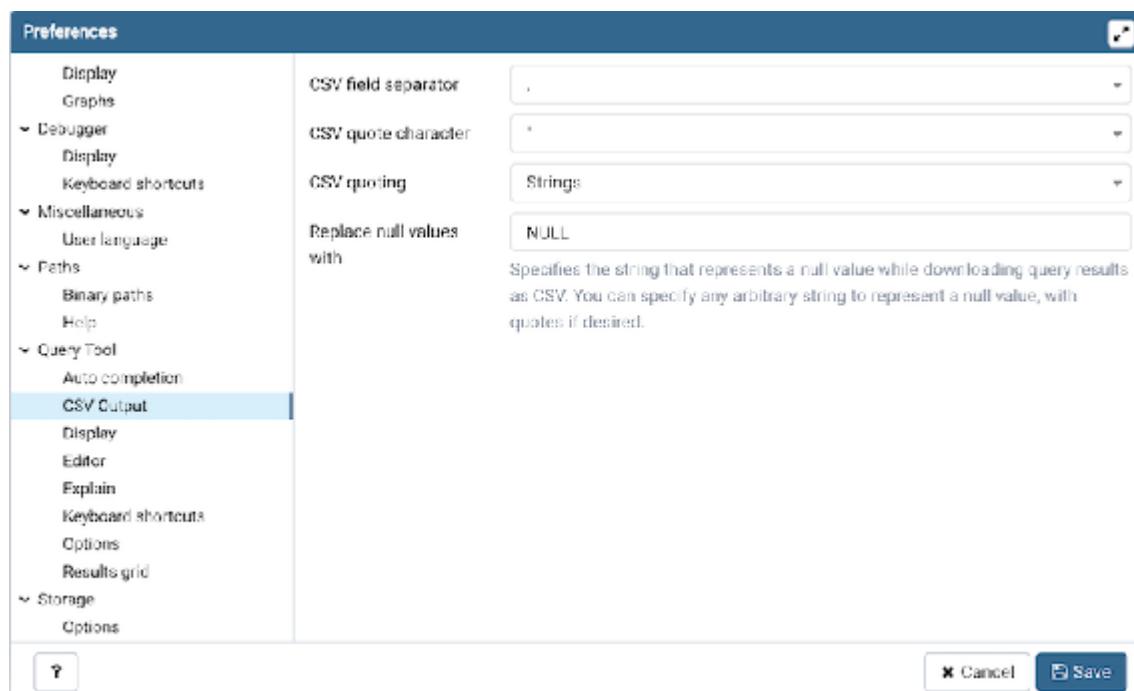
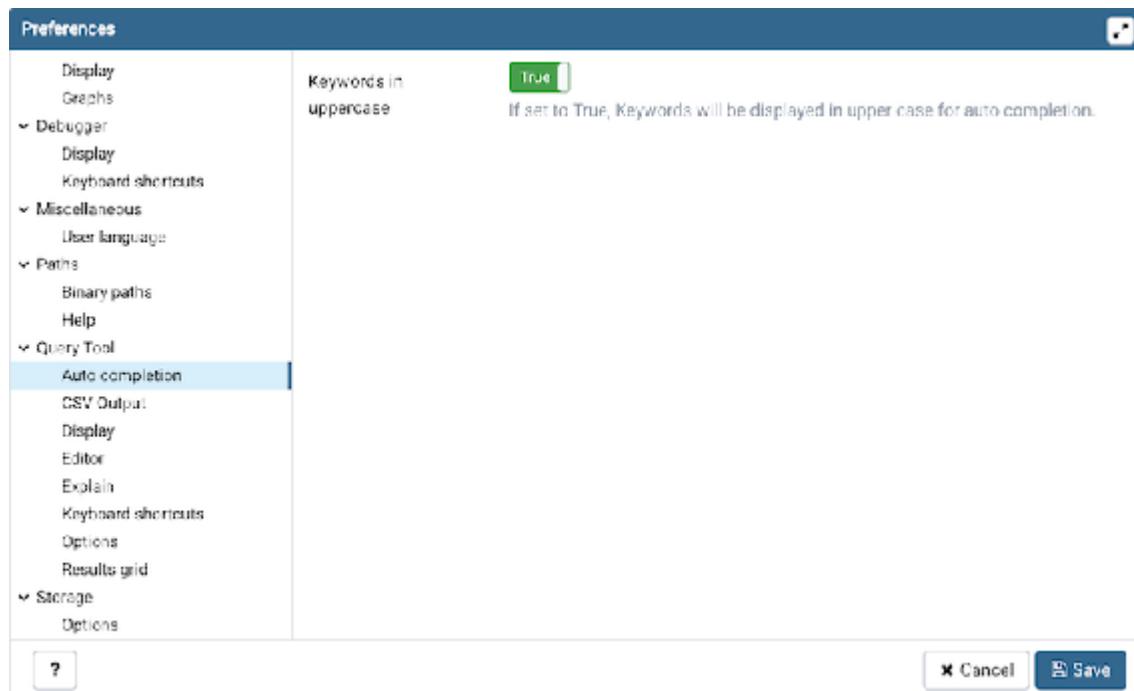
Use the fields on the *Binary paths* panel to specify the path to the directory that contains the utility programs (pg_dump, pg_restore, and pg_dumpal) for monitored databases:

Use the *EDB Advanced Server Binary Path* field to specify the location of the EDB Postgres Advanced Server utility programs. If this path is not set, pgAdmin wil attempt to find the utilities in standard locations used by EnterpriseDB.

Use the *Greenplum Database Binary Path* field to specify the location of the Greenplum database utility programs. If this path is not set, pgAdmin wil attempt to find the utilities in standard locations used by Greenplum.

Use the *PostgreSQL Binary Path* field to specify the location of the PostgreSQL utility programs. If this path is not set, pgAdmin wil attempt to find the utilities in standard locations used by PostgreSQL.

Use the fields on the *Help* panel to specify the location of help files.



Use the *EDB Advanced Server Help Path* to specify the path to EDB Postgres Advanced Server documentation.

Use the *PostgreSQL Help Path* to specify the path to PostgreSQL

documentation.

Please note: the default help paths include the *VERSION* placeholder; the \$VERSION\$ placeholder will be replaced by the current database version.

The Query Tool Node

Expand the *Query Tool* node to access panels that allow you to specify your preferences for the Query Editor tool.

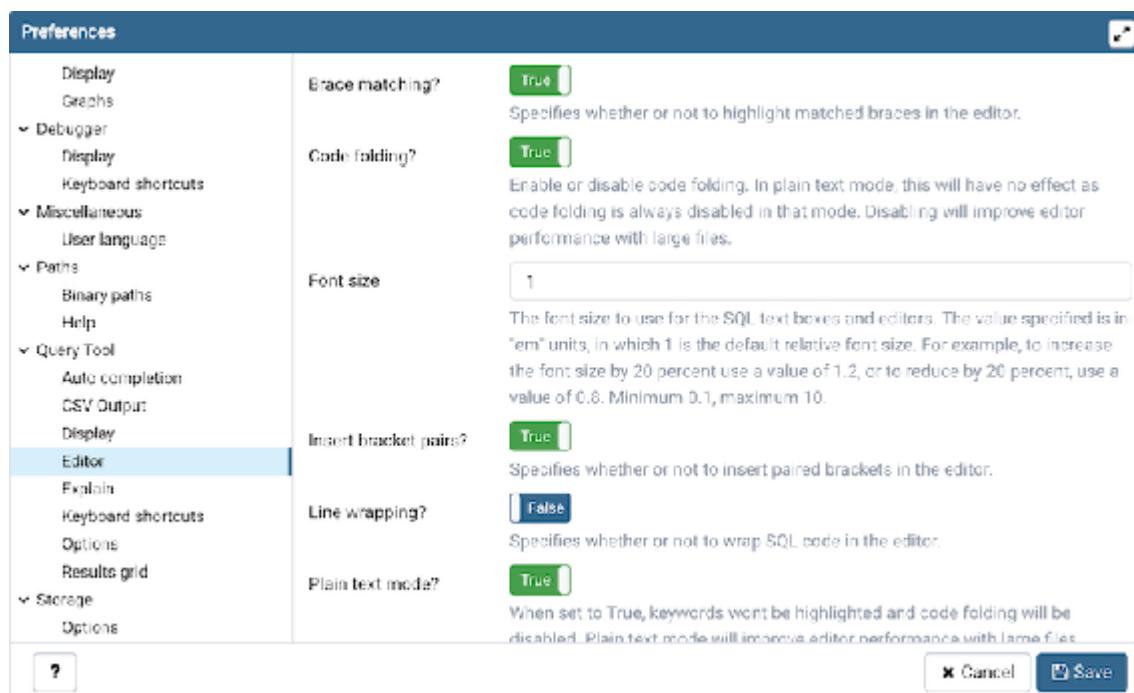
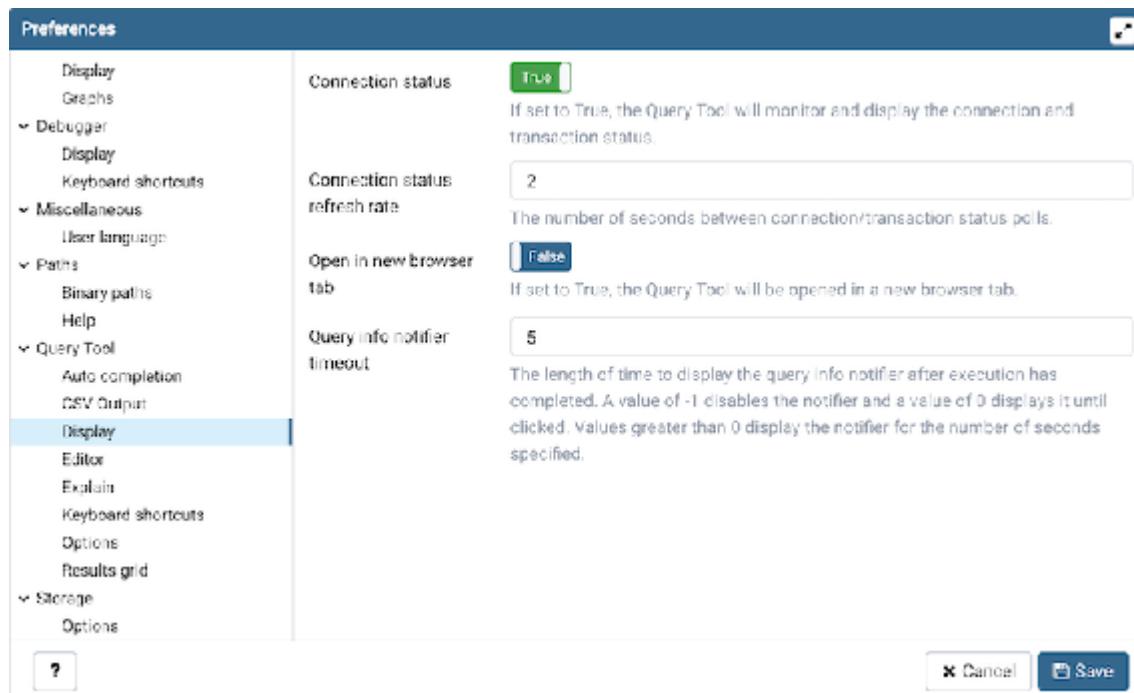
Use the fields on the *Auto Completion* panel to set the auto completion options.

When the *Keywords in uppercase* switch is set to *True* then keywords are shown in upper case.

Use the fields on the *CSV Output* panel to control the CSV output.

Use the *CSV field separator* drop-down listbox to specify the separator character that will be used in CSV output.

Use the *CSV quote character* drop-down listbox to specify the quote character that will be used in CSV output.



Use the CSV quoting drop-down listbox to select the fields that wil be quoted in the CSV output; select *Strings*, *All* , or *None*.

Use the Replace nul values with option to replace nul values with specified string in the output file. Default is set to 'NULL'.

Use the fields on the *Display* panel to specify your preferences for the Query Tool display.

When the *Connection status* switch is set to *True*, each new instance of the Query Tool wil display connection and transaction status.

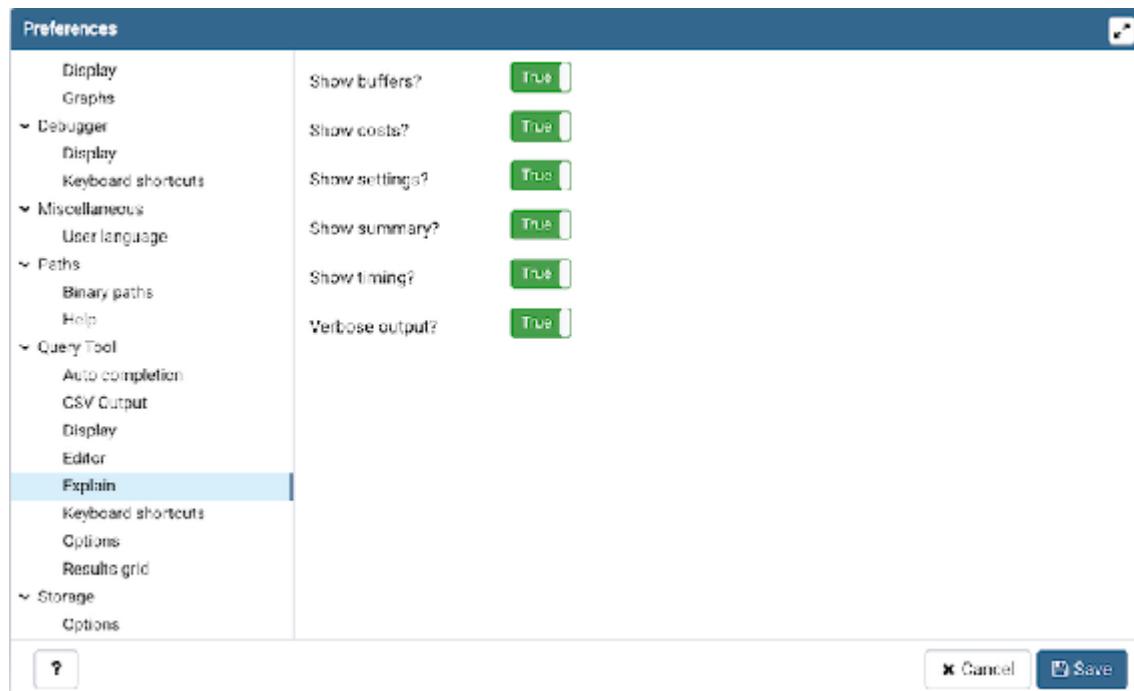
Use the *Connection status refresh rate* field to specify the number of seconds between connection/transaction status updates.

When the *Open in new browser tab* switch is set to *True*, each new instance of the Query Tool wil open in a new browser tab.

Use the *Query info notifier timeout* field to control the behaviour of the notifier that is displayed when query execution completes. A value of -1 wil disable the notifier, and a value of 0 wil display it until clicked. If a positive value above zero is specified, the notifier wil be displayed for the specified number of seconds. The default is 5.

Use the fields on the *Editor* panel to change settings of the query editor.

When the *Brace matching?* switch is set to *True*, the editor wil highlight pairs of matched braces.



When the *Code folding?* switch is set to *False*, the editor wil disable code folding. Disabling wil improve editor performance with large files.

Use the *Font size* field to specify the font size that wil be used in text boxes and editors.

When the *Insert bracket pairs?* switch is set to *True*, the editor wil automatical y insert paired brackets.

When the *Line wrapping* switch is set to *True*, the editor wil implement line-wrapping behavior.

When the *Plain text mode?* switch is set to *True*, the editor mode wil be changed to text/plain. Keyword highlighting and code folding wil be disabled. This wil improve editor performance with large files.

Use the *Tab size* field to specify the number of spaces per tab character in the editor.

When the *Use spaces* switch is set to *True*, the editor wil insert spaces (instead of tab characters) when the tab key or auto-indent

are used.

Use the fields on the *Explain* panel to specify the level of detail included in a graphical EXPLAIN.

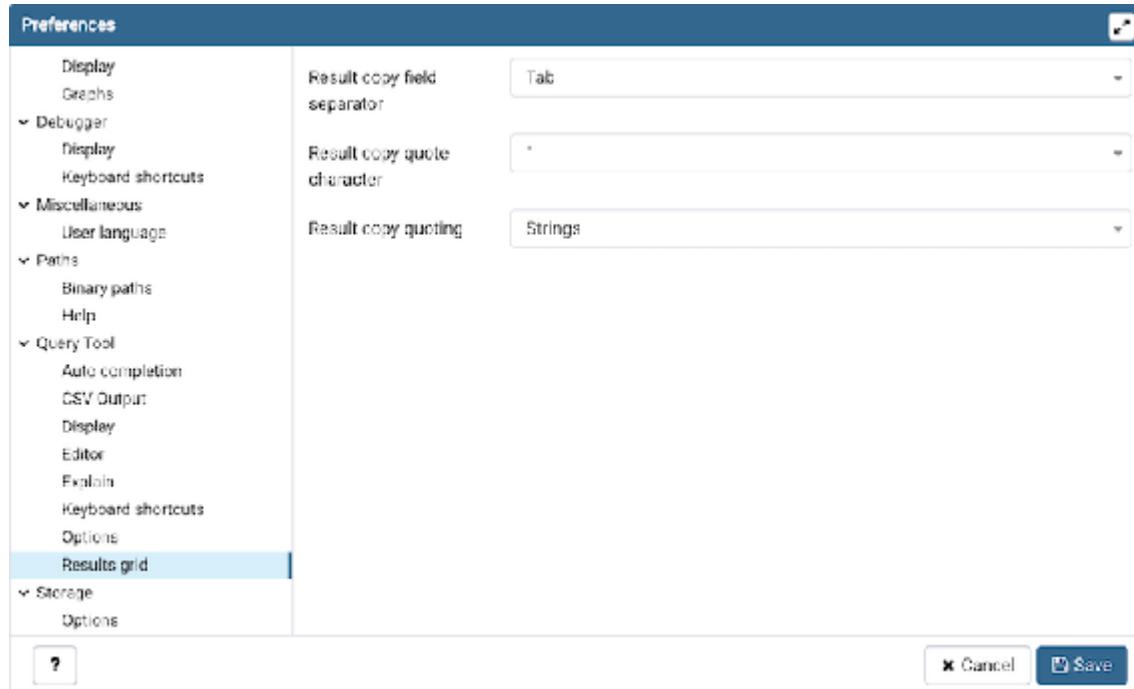
When the *Show Buffers?* switch is set to *True*, graphical explain details will include information about buffer usage.

When the *Show Costs?* switch is set to *True*, graphical explain details will include information about the estimated startup and total cost of each plan, as well as the estimated number of rows and the estimated width of each row.

When the *Show Timing?* switch is set to *True*, graphical explain details will include the startup time and time spent in each node in the output.

When the *Verbose output?* switch is set to *True*, graphical explain details will include extended information about the query execution plan.





Use the fields on the *Options* panel to manage editor preferences.

When the *Auto-Commit?* switch is set to *True*, each successful query is committed after execution.

When the *Auto-Roll back?* switch is set to *True*, failed queries are rolled back.

When the *Prompt to save unsaved data changes?* switch is set to *True*, the editor will prompt the user to save unsaved data when exiting the data editor.

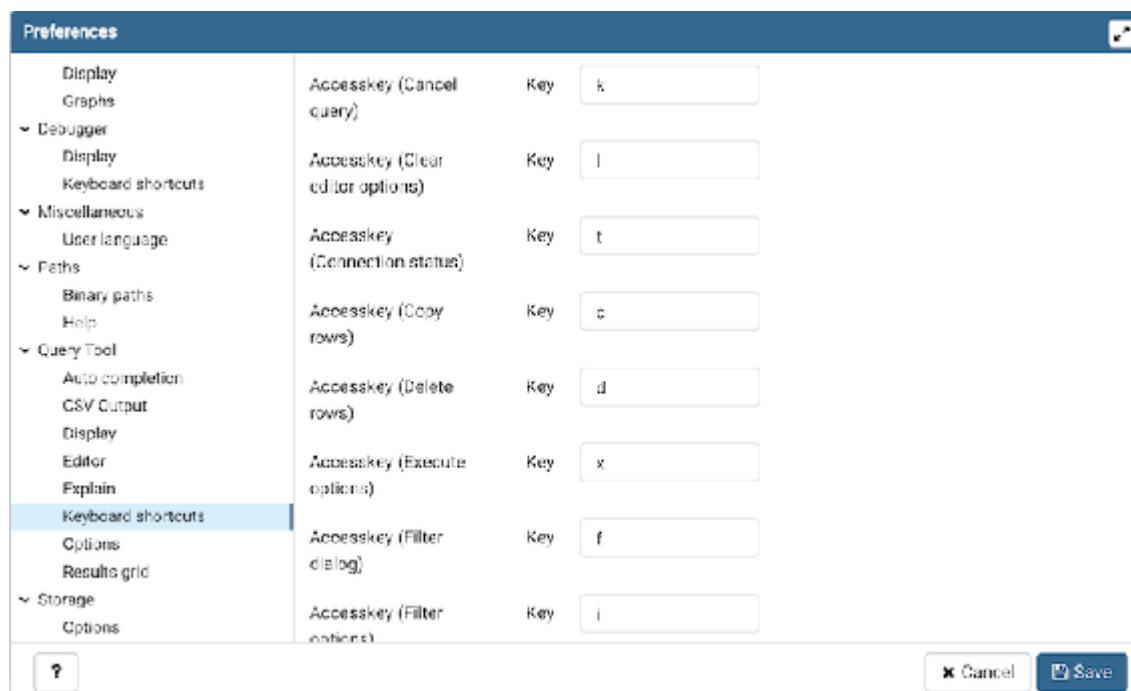
When the *Prompt to save unsaved query changes?* switch is set to *True*, the editor will prompt the user to save unsaved query modifications when exiting the Query Tool.

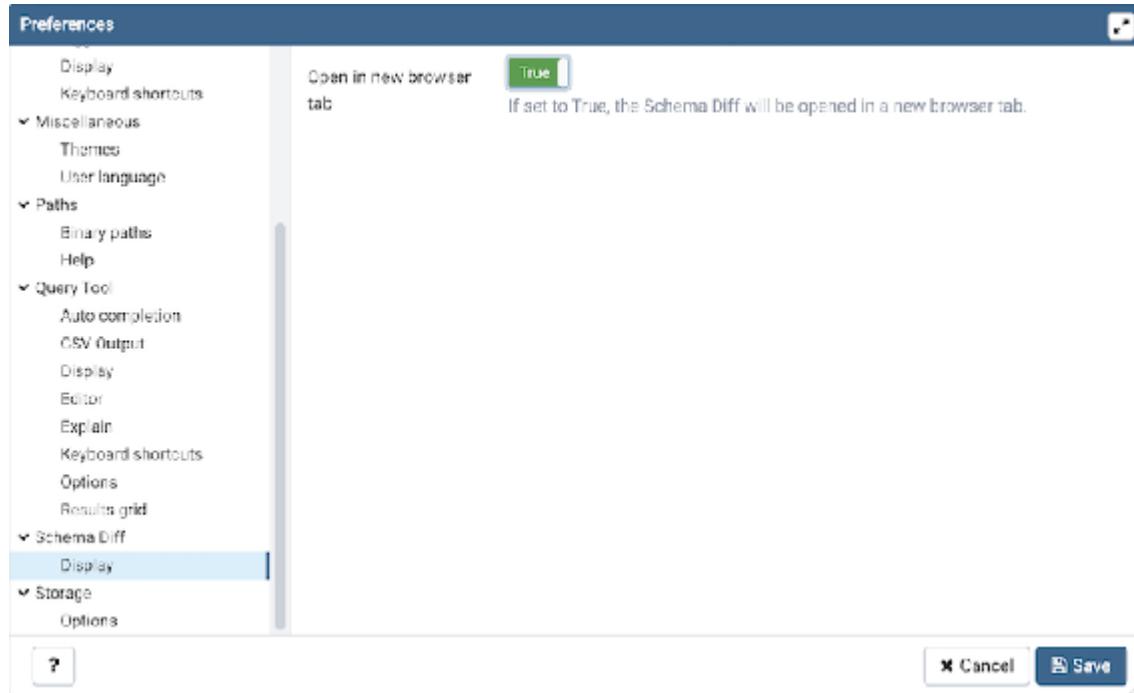
When the *Prompt to commit/roll back active transactions?* switch is set to *True*, the editor will prompt the user to commit or roll back changes when exiting the Query Tool while the current transaction is not committed.

Use the fields on the *Results grid* panel to specify your formatting preferences for copied data.

Use the *Result copy field separator* drop-down listbox to select the field separator for copied data.

Use the *Result copy quote character* drop-down listbox to select the quote character for copied data.





Use the *Result copy quoting* drop-down listbox to select which type of fields require quoting; select *All*, *None*, or *Strings*.

Use the fields on the *Keyboard shortcuts* panel to configure shortcuts for the Query Tool window navigation:

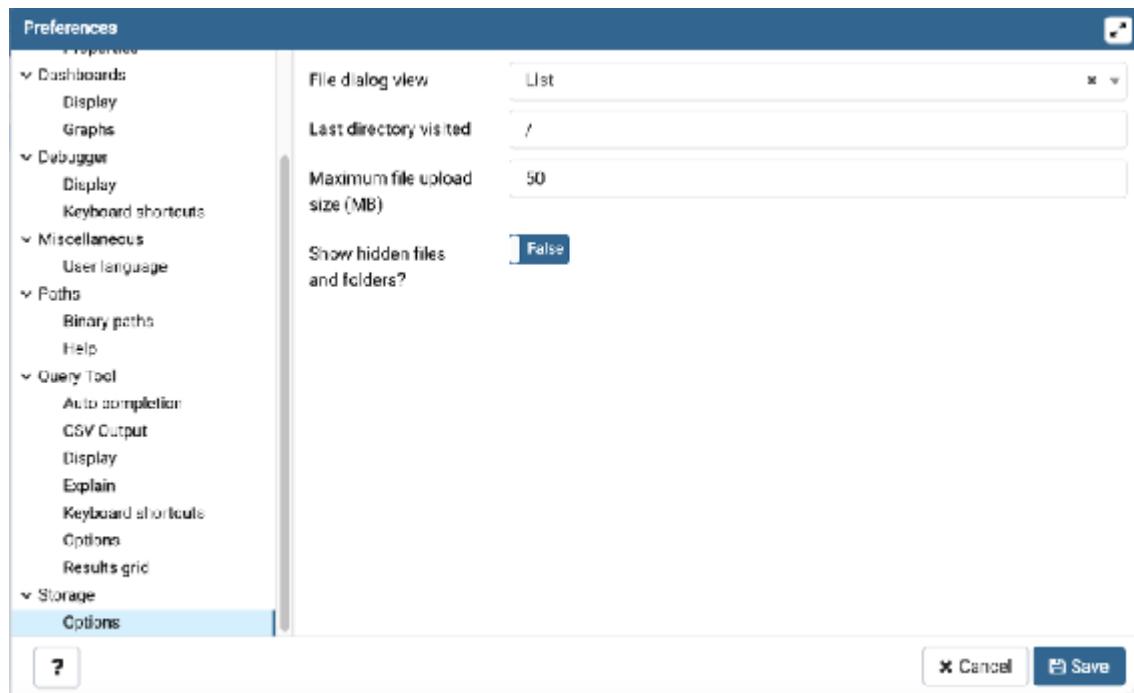
The Schema Diff Node

Expand the *Schema Diff* node to specify your display preferences.

Use the *Open in new browser tab* switch to indicate if you would like Schema Diff to open in a new tab.

The Storage Node

Expand the *Storage* node to specify your storage preferences.



Use the fields on the *Options* panel to specify storage preferences.

Use the *File dialog view* drop-down listbox to select the style of icons and display format that wil be displayed when you open the file manager; select *List* to display a list view, or *Grid* to display folder icons.

Use the *Last directory visited* field to specify the name of the folder in which the file manager wil open.

Use the *Maximum file upload size(MB)* field on the *Options* panel of the Storage node to specify the maximum file size for an upload.

When the *Show hidden files and folders?* switch is set to *True*, the file manager wil display hidden files and folders.

Quick Links

- » Documentation
- » Manuals
 - » Archive
 - » Chinese
 - » French
 - » Japanese
 - » Russian
- » Release Notes
- » Books
- » Tutorials & Other Resources
- » FAQ
- » Wiki

Documentation

This section contains [current](#) and [archived](#) manuals for PostgreSQL users. You can read the [release notes](#), and view a listing of [books](#) written about PostgreSQL.

Online Manuals

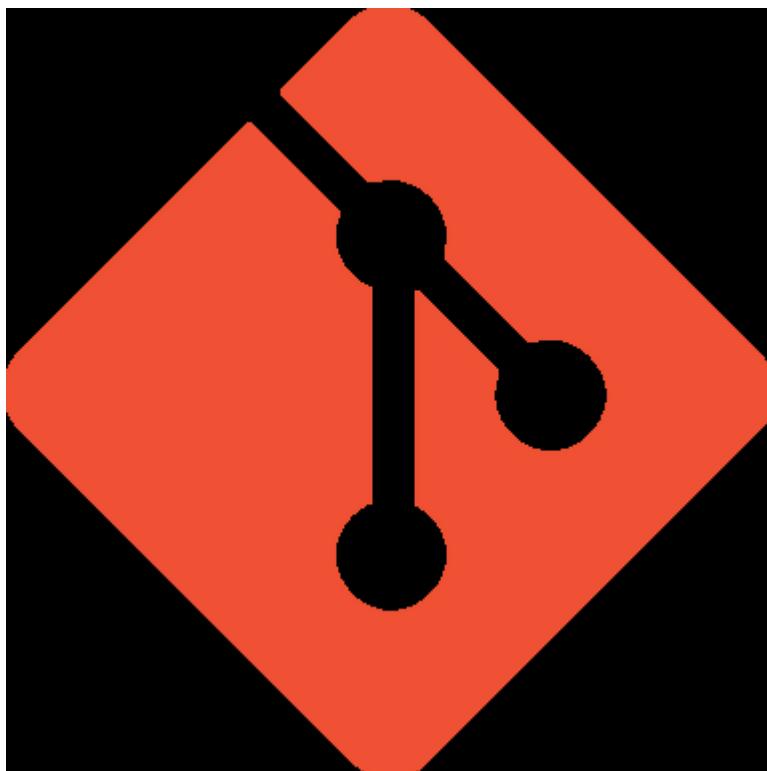
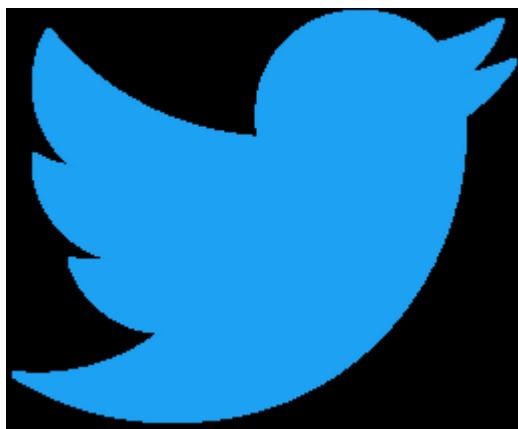
- 12
- 11
- 10
- 9.6
- 9.5

Translated Manuals

- [Chinese](#)
- [French](#)
- [Japanese](#)
- [Russian](#)

[More manuals](#)





13th February 2020: [PostgreSQL 12.2, 11.7, 10.12, 9.6.17, 9.5.21, and 9.4.26 Released!](#)

Quick Links

[Documentation](#)

[Manuals](#)

[Archive](#)

[Chinese](#)

[French](#)

[Japanese](#)

[Russian](#)

[Release Notes](#)

[Books](#)

[Tutorials & Other Resources](#)

[FAQ](#)

[Wiki](#)

Documentation □

Online Manuals

[This section contains current and archived manuals for PostgreSQL users. You can read the release](#)

[12](#)

[notes, and view a listing of books written about PostgreSQL.](#)

[11](#)

[10](#)

[9.6](#)

[9.5](#)

Translated Manuals

[Chinese](#)

[French](#)

[Japanese](#)

[Russian](#)

[More manuals](#)

[Policies](#) | [Code of Conduct](#) | [About PostgreSQL](#) | [Contact](#)

Copyright © 1996-2020 The PostgreSQL Global Development Group

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) » [Creation or Modifying a Table](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Primary key Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
 - [Check Dialog](#)
 - [Column Dialog](#)
 - [Compound Trigger Dialog](#)
 - [Exclusion Constraint Dialog](#)
 - [Foreign key Dialog](#)
 - [Index Dialog](#)
 - [Primary key Dialog](#)
 - [Rule Dialog](#)
 - [Table Dialog](#)
 - [Trigger Dialog](#)
 - [Unique Constraint Dialog](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Primary key Dialog

Use the Primary key dialog to create or modify a primary key constraint. A primary key constraint indicates that a column, or group of columns, uniquely identifies rows in a table. This requires that the values in the selected column(s) be both unique and not null.

The Primary key dialog organizes the development of a primary key constraint through the General and Definition tabs. The SQL tab displays the SQL code generated by dialog selections.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Creating or Modifying a Table](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Primary key Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[□](#)

[Check Dialog](#)

[Column Dialog](#)

[Latest Docs](#)

[Compound Trigger Dialog](#)

[Exclusion Constraint Dialog](#)

[□](#)

[Foreign key Dialog](#)

[Index Dialog](#)

[Ask a](#)

[Primary key Dialog](#)

[Question](#)

[Rule Dialog](#)

[Table Dialog](#)

[□](#)

[Trigger Dialog](#)

[Report a Bug](#)

[Unique Constraint Dialog](#)

[Management Basics](#)

[Backup and Restore](#)



[Developer Tools](#)

[Screenshots](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Primary key Dialog

Use the *Primary key* dialog to create or modify a primary key constraint. A primary key constraint indicates that a column, or group of columns, uniquely identifies rows in a table. This requires that the values in the selected column(s) be both unique and not null.

The *Primary key* dialog organizes the development of a primary key constraint through the *General* and *Definition* tabs. The *SQL* tab displays the SQL code generated by dialog selections.

Create - Primary key

X

General Definition SQL

Name

pr_customer

Comment



Cancel

Reset

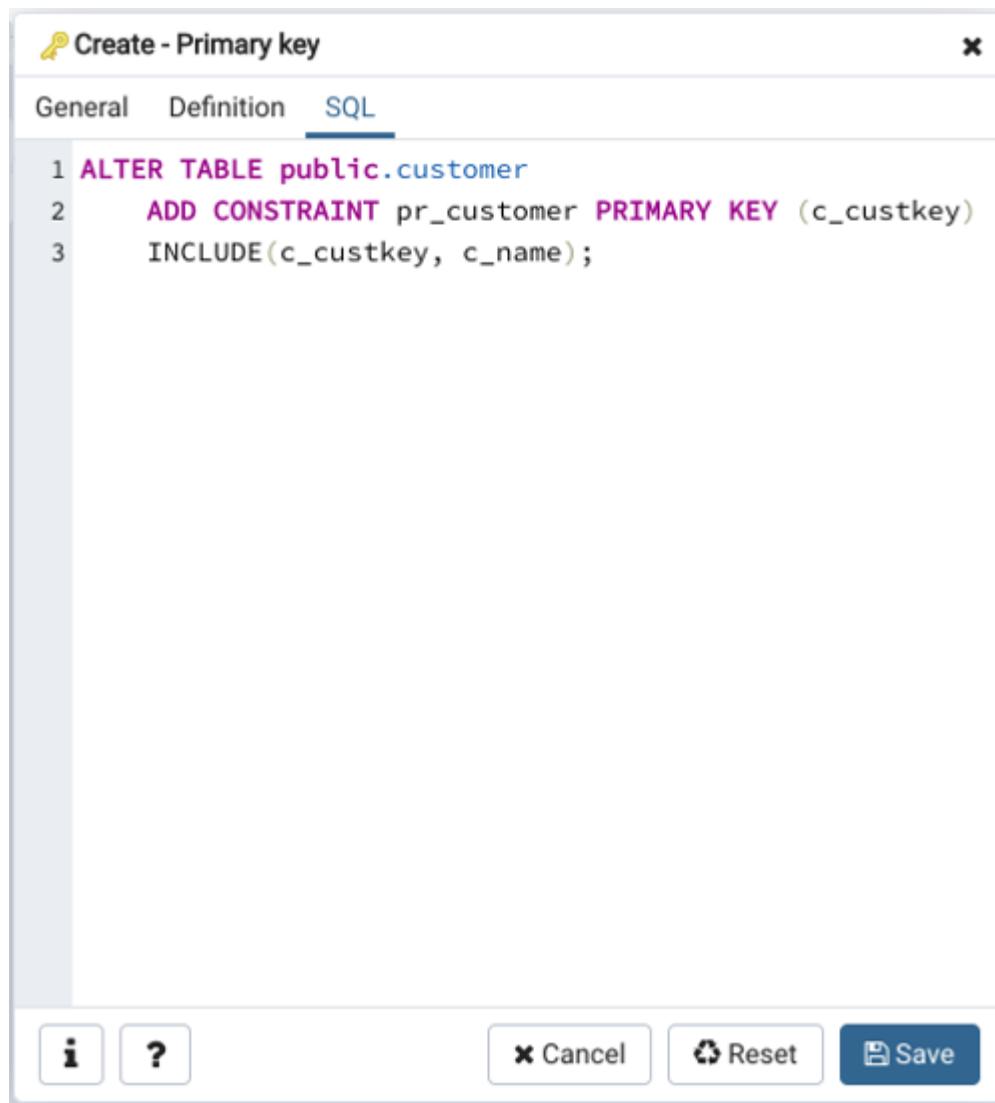
Save



Use the fields in the *General* tab to identify the primary key: Use the *Name* field to add a descriptive name for the primary key constraint. The name will be displayed in the *pgAdmin* tree control.

Click the *Definition* tab to continue.

Use the fields in the *Definition* tab to define the primary key constraint: Click inside the *Columns* field and select one or more column names from the drop-down listbox. To delete a selection, click the *x* to the left of the column name. The primary key constraint should be different from any unique constraint defined for the same table; the selected column(s) for the constraints must be distinct.

A screenshot of a software interface titled "Create - Primary key". The "SQL" tab is selected. The SQL code shown is:

```
1 ALTER TABLE public.customer
2   ADD CONSTRAINT pr_customer PRIMARY KEY (c_custkey)
3   INCLUDE(c_custkey, c_name);
```

The interface includes standard buttons at the bottom: "Cancel", "Reset", and "Save".

Use *Include columns* field to specify columns for *INCLUDE* clause of the index. This option is available in Postgres 11 and later.

Select the name of the tablespace in which the primary key constraint will reside from the drop-down listbox in the *Tablespace* field.

Select the name of an index from the drop-down listbox in the *Index* field. This field is optional. Adding a primary key will automatically create a unique B-tree index on the column or group of columns listed in the primary key, and will force the column(s) to be marked NOT NULL.

Use the *Fil Factor* field to specify a fil factor for the table and index.

The fil factor for a table is a percentage between 10 and 100. 100 (complete packing) is the default.

Move the *Deferrable?* switch to the *Yes* position to specify the timing of the constraint is deferrable and can be postponed until the end of the statement. The default is *No*.

If enabled, move the *Deferred?* switch to the *Yes* position to specify the timing of the constraint is deferred to the end of the statement.

The default is *No*.

Click the *SQL* tab to continue.

Your entries in the *Primary key* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Primary key* dialog:

The example shown demonstrates creating a primary key constraint named *dept_pkey* on the *dept_id* column of the *dept* table.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Procedure Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Procedure Dialog ¶

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Procedure Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)

[□](#)

[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)

[□](#)

[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)

[□](#)

[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

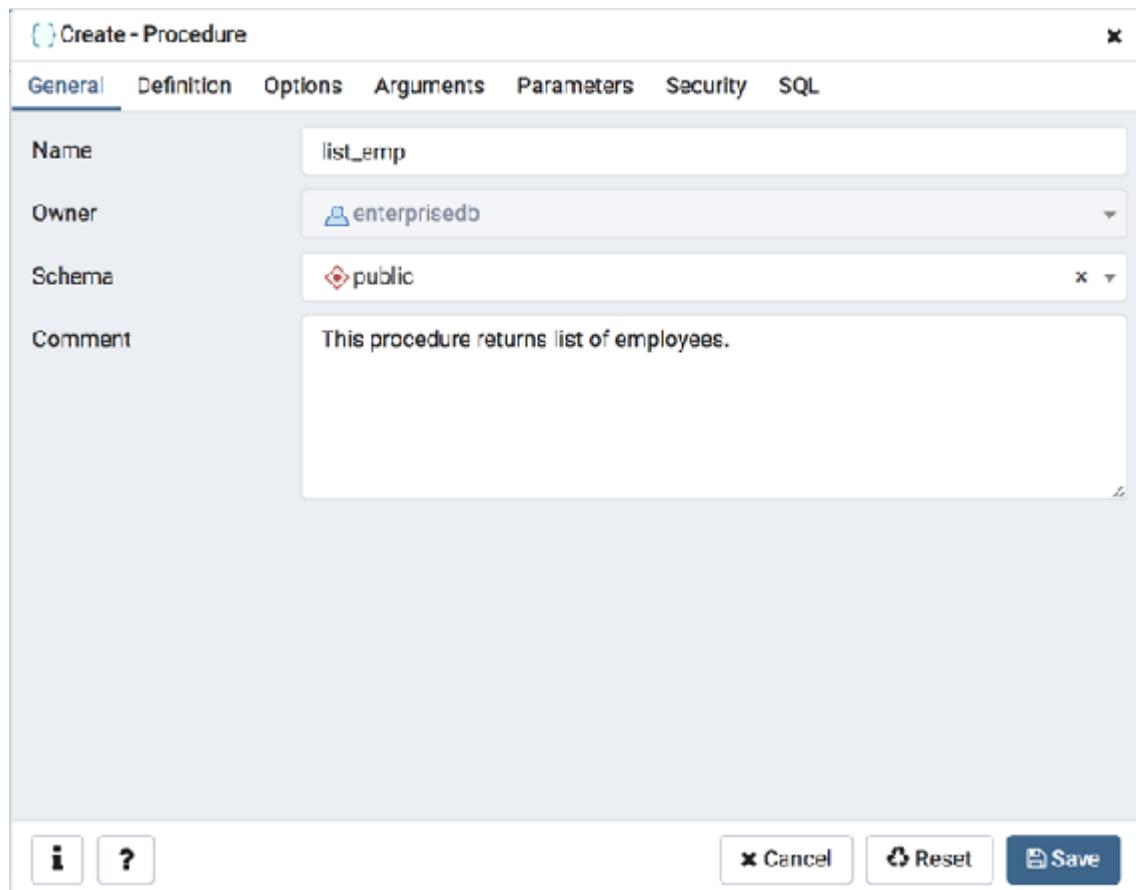
[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

Procedure Dialog



Use the *Procedure* dialog to create a procedure; procedures are supported by PostgreSQL v11+ and EDB Postgres Advanced Server. The *Procedure* dialog allows you to implement options of the CREATE PROCEDURE command.

The *Procedure* dialog organizes the development of a procedure through the following dialog tabs: *General*, *Definition*, *Options*, *Arguments*, *Parameters*, and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

Use the fields in the *General* tab to identify a procedure: Use the *Name* field to add a descriptive name for the procedure. The name

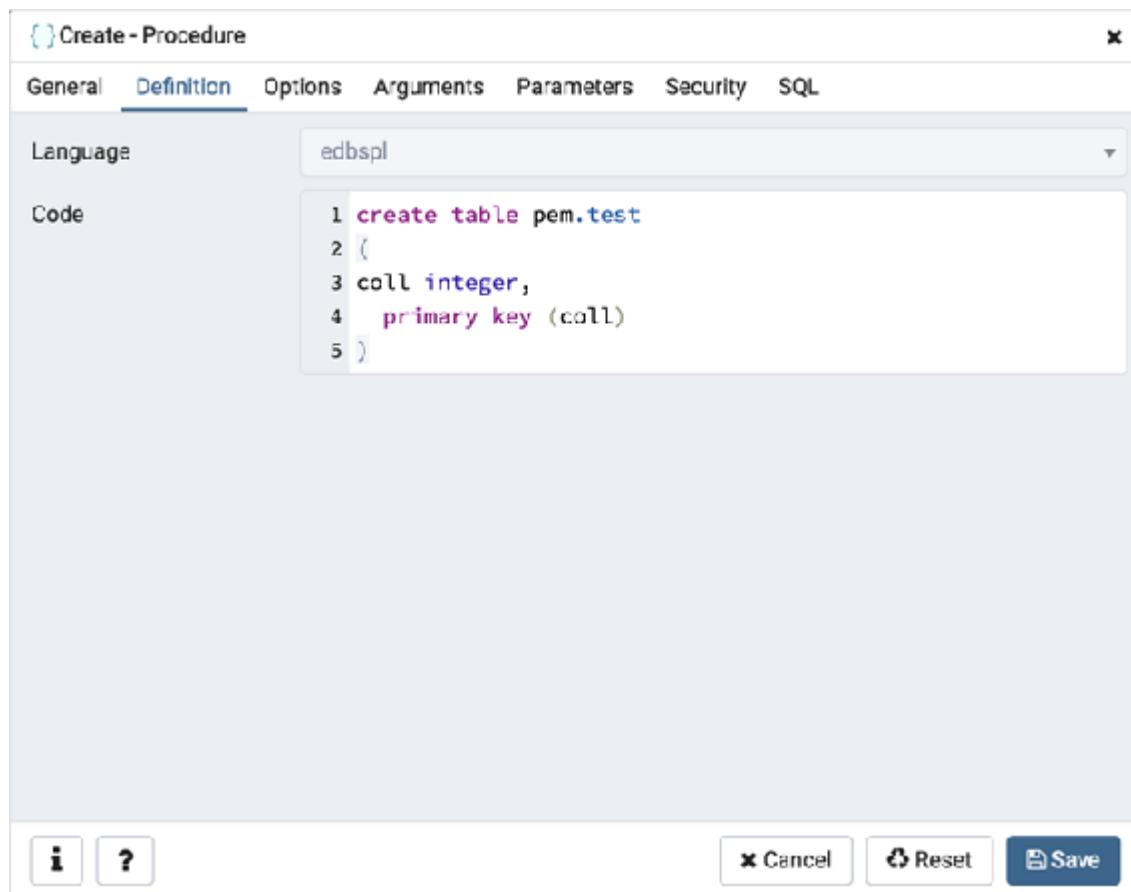
wil be displayed in the *pgAdmin* tree control.

Use the drop-down listbox next to *Owner* to select a role.

Select the name of the schema in which the procedure wil reside from the drop-down listbox in the *Schema* field.

Store notes about the procedure in the *Comment* field.

Click the *Definition* tab to continue.



Create - Procedure

General Definition Options Arguments Parameters Security SQL

Volatility: VOLATILE

Strict?: Yes

Security of definer?: Yes

Parallel: UNSAFE

Estimated cost:

Leak proof?: Yes

i ? **Cancel** **Reset** **Save**

The screenshot shows the 'Create - Procedure' dialog box with the 'Options' tab selected. It contains several configuration fields: 'Volatility' (set to 'VOLATILE'), 'Strict?' (set to 'Yes'), 'Security of definer?' (set to 'Yes'), 'Parallel' (set to 'UNSAFE'), and an empty 'Estimated cost' field. At the bottom are standard buttons: 'i ?' (info/help), 'Cancel', 'Reset', and 'Save'.

Use the fields in the *Definition* tab to define the procedure: Use the drop-down listbox next to *Language* to select a language.

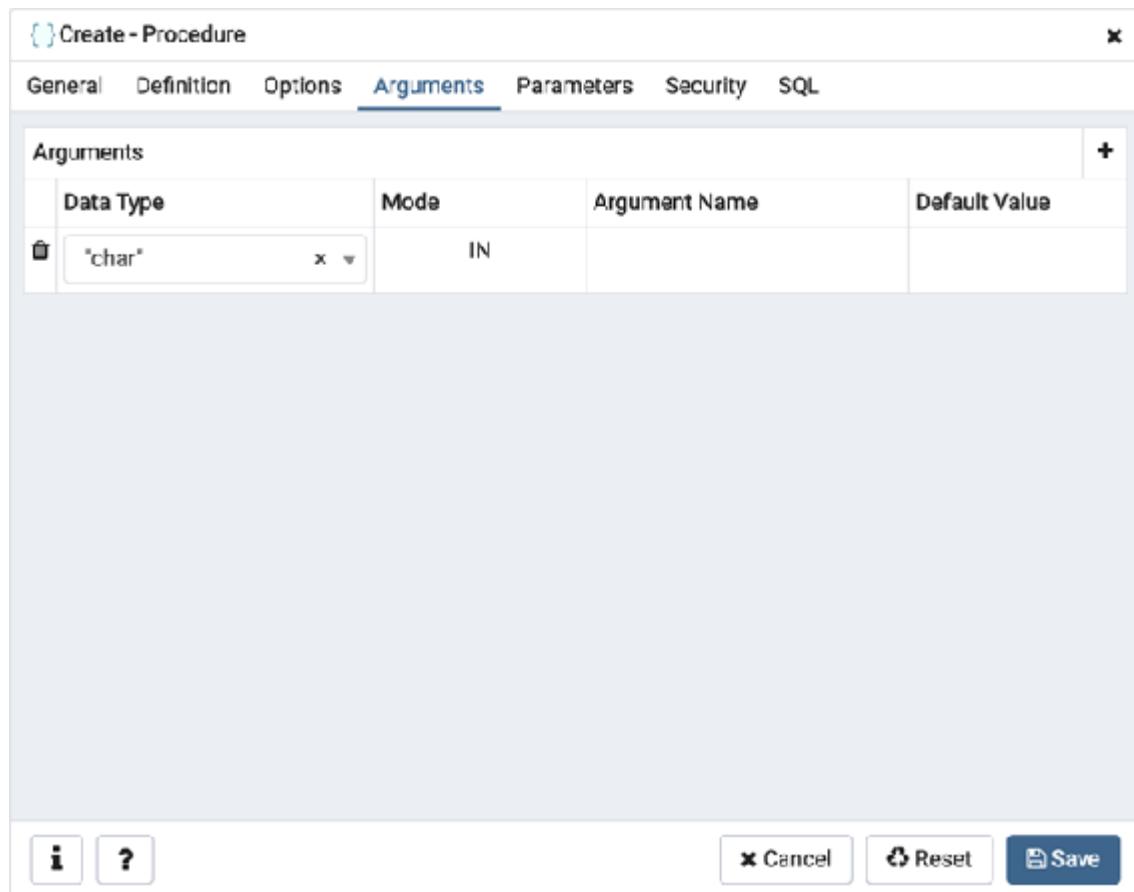
The default is `edbspl`.

Use the *Code* field to specify the code that will execute when the procedure is called.

Click the *Options* tab to continue.

Use the fields in the *Options* tab to describe or modify the behavior of the procedure:

Use the drop-down listbox under *Volatility* to select one of the following. *VOLATILE* is the default value.



VOLATILE indicates that the value can change even within a single table scan, so no optimizations can be made.

STABLE indicates that the procedure cannot modify the database, and that within a single table scan it will consistently return the same result for the same argument values, but that its result could change across SQL statements.

IMMUTABLE indicates that the procedure cannot modify the database and always returns the same result when given the same argument values.

Move the **Strict?** switch to indicate if the procedure always returns NULL whenever any of its arguments are NULL. If Yes, the procedure is not executed when there are NULL arguments; instead a NULL

result is assumed automatically. The default is *No*.

Move the *Security of definer?* switch to specify that the procedure is to be executed with the privileges of the user that created it. The default is *No*.

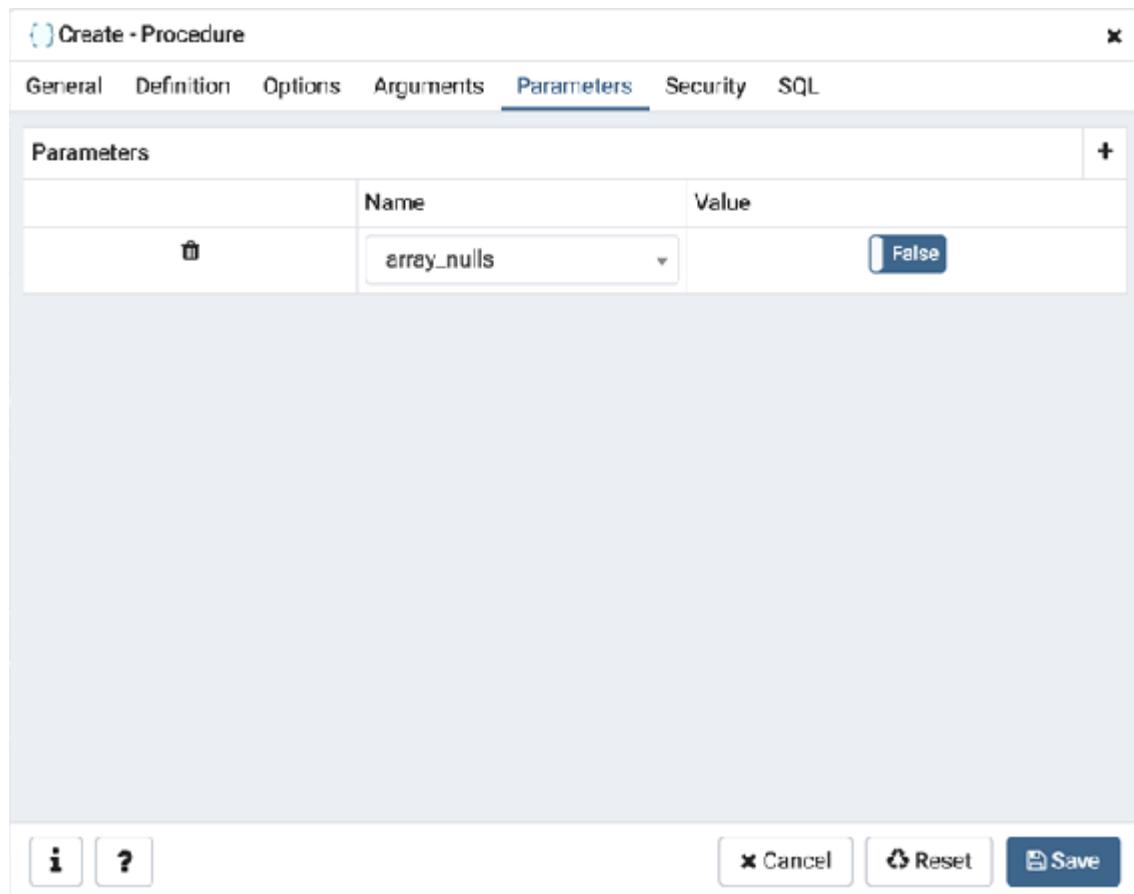
Use the *Estimated cost* field to specify a positive number representing the estimated execution cost for the procedure, in units of `cpu_operator_cost`. If the procedure returns a set, this is the cost per returned row.

Move the *Leak proof?* switch to indicate whether the procedure has side effects — it reveals no information about its arguments other than by its return value. The default is *No*.

Click the *Arguments* tab to continue.

Use the fields in the *Arguments* tab to define an argument. Click *Add* to set parameters and values for the argument:

Use the drop-down listbox next to *Data type* to select a data type.



Use the drop-down listbox next to *Mode* to select a mode. Select *IN* for an input parameter; select *OUT* for an output parameter; select *INOUT* for both an input and an output parameter; or, select *VARIADIC* to specify a VARIADIC parameter.

Write a name for the argument in the *Argument Name* field.

Specify a default value for the argument in the *Default Value* field.

Click *Add* to define another argument; to discard an argument, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *Parameters* tab to continue.

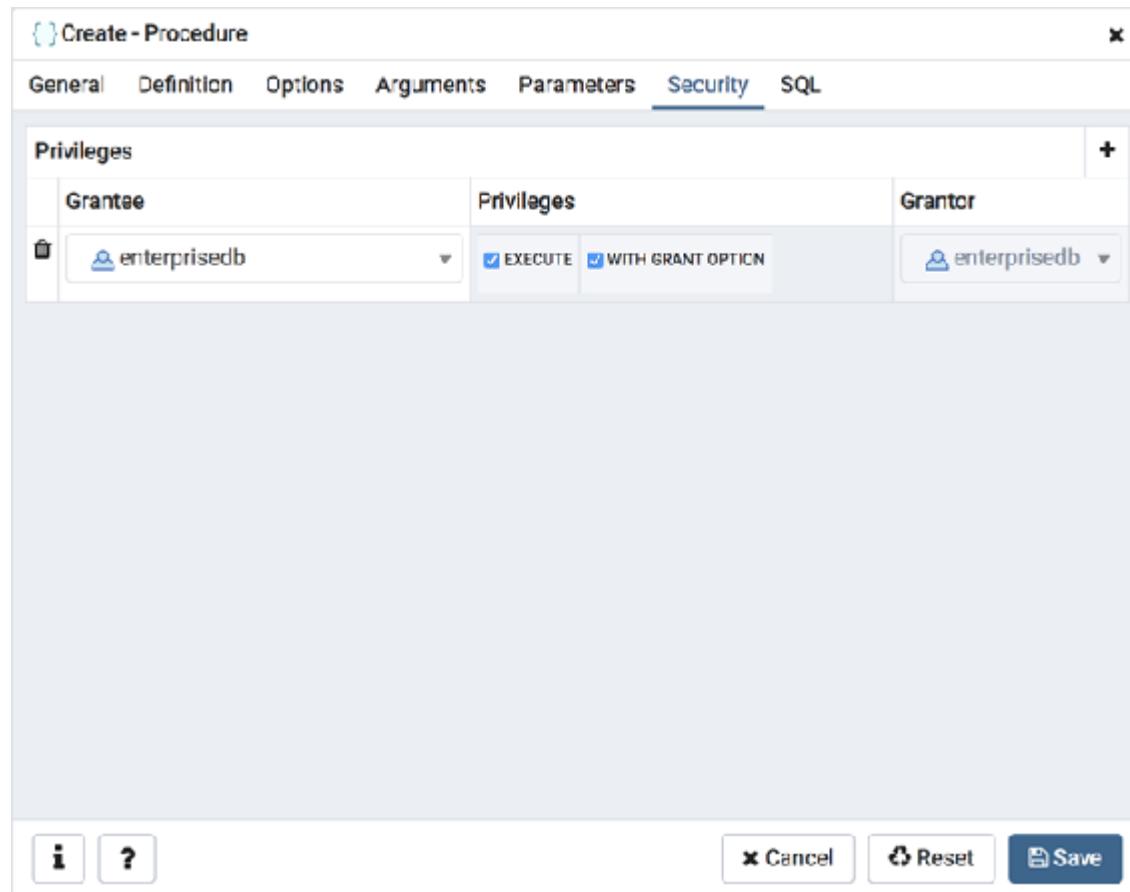
Use the fields in the *Parameters* tab to specify settings that will be applied when the procedure is invoked:

Use the drop-down listbox next to *Parameter Name* in the *Parameters* panel to select a parameter.

Click the *Add* button to add the variable to *Name* field in the table.

Use the *Value* field to specify the value that will be associated with the selected variable. This field is context-sensitive.

Click the *Security* tab to continue.



Use the *Security* tab to assign privileges and define security labels.

Use the *Privileges* panel to assign execute privileges for the procedure to a role:

Select the name of the role from the drop-down listbox in the *Grantee* field.

Click inside the *Privileges* field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.

Click *Add* to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

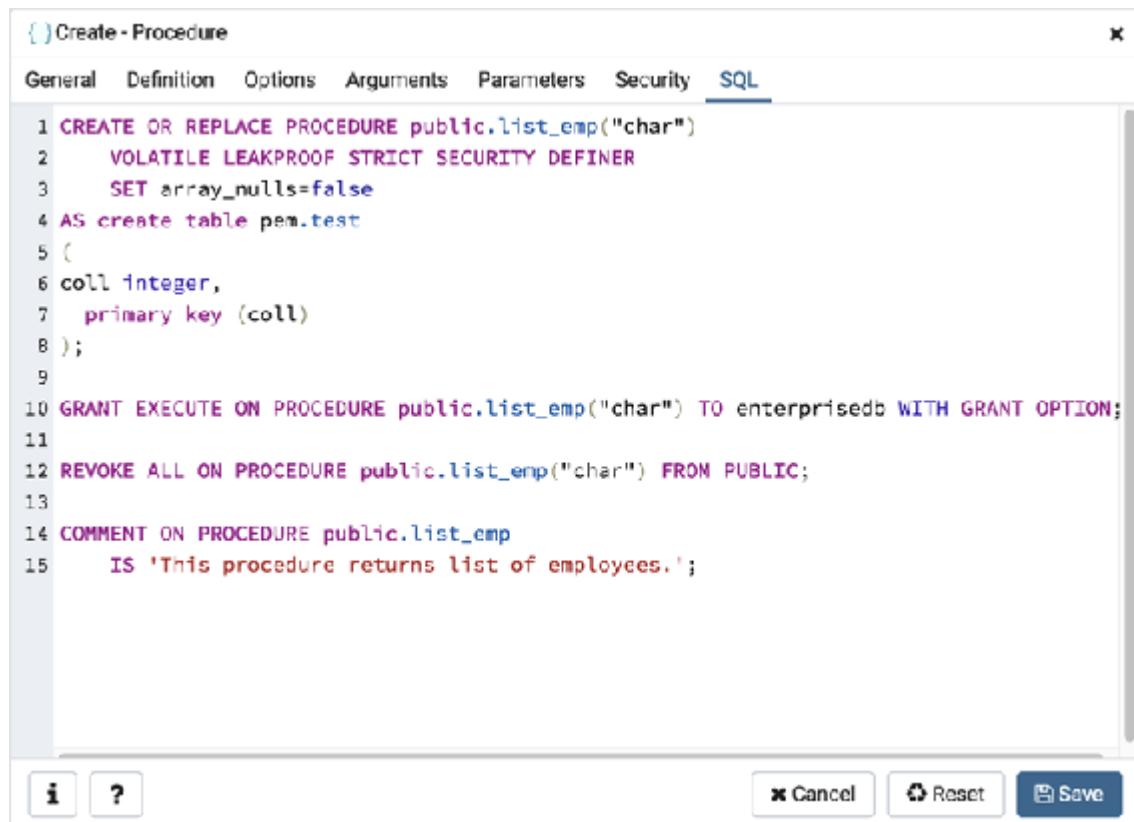
Use the *Security Labels* panel to define security labels applied to the procedure. Click *Add* to add each security label selection: Specify a security label provider in the *Provider* field. The named provider must be loaded and must consent to the proposed labeling operation.

Specify a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL

places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click *Add* to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.



The screenshot shows the 'Create - Procedure' dialog box in Oracle SQL Developer. The 'SQL' tab is selected. The code area contains the following SQL command:

```
1 CREATE OR REPLACE PROCEDURE public.list_emp("char")
2   VOLATILE LEAKPROOF STRICT SECURITY DEFINER
3   SET array_nulls=false
4 AS create table pem.test
5 (
6 coll integer,
7 primary key (coll)
8 );
9
10 GRANT EXECUTE ON PROCEDURE public.list_emp("char") TO enterpriseDB WITH GRANT OPTION;
11
12 REVOKE ALL ON PROCEDURE public.list_emp("char") FROM PUBLIC;
13
14 COMMENT ON PROCEDURE public.list_emp
15 IS 'This procedure returns list of employees.';
```

At the bottom of the dialog, there are three buttons: 'Cancel', 'Reset', and 'Save'. The 'Save' button is highlighted.

Your entries in the *Procedure* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by selections made in the *Procedure* dialog:

The example demonstrates creating a procedure that returns a list of employees from a table named `emp`. The procedure is a SECURITY

DEFINER, and will execute with the privileges of the role that defined the procedure.

Click the *Info* button () to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Developer Tools](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Query Tool](#)
 - [Toolbar](#)
 - [The SQL Editor Panel](#)
 - [The Data Output Panel](#)
 - [Explain Panel](#)
 - [Messages Panel](#)
 - [Query History Panel](#)
 - [Connection Status](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
 - [Debugger](#)
 - [Query Tool](#)
 - [View/Edit Data](#)
 - [Schema Diff](#)
- [onAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Query Tool

The Query Tool is a powerful, feature-rich environment that allows you to execute arbitrary SQL commands and review the result set. You can access the Query Tool via the *Query Tool* menu option on the *Tools* menu, or through the context menu of select nodes of the Browser tree control.

The Query Tool allows you to:

- Issue ad-hoc SQL queries.
- Execute arbitrary SQL commands.
- Edit the result set of a SELECT query if it is [updatable](#).
- Displays current connection and transaction status as configured by the user.
- Save the data displayed in the output panel to a CSV file.
- Review the execution plan of a SQL statement in either a text, a graphical format or a table format (similar to <https://explain.depesz.com>).
- View analytical information about a SQL statement.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#)
[Development](#)

[pgAdmin 4 4.19 documentation](#) » [Developer Tools](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Query Tool](#)

[Download](#)

[Toolbar](#)

[The SQL Editor Panel](#)

[The Data Output Panel](#)



[Explain Panel](#)

[FAQ](#)

[Messages Panel](#)

[Query History Panel](#)



[Connection Status](#)

[Getting Started](#)

[Latest Docs](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)



[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[Ask a](#)

[Management Basics](#)

[Question](#)

[Backup and Restore](#)

[Developer Tools](#)



[Debugger](#)

[Report a Bug](#)

Query Tool

[View/Edit Data](#)

[Schema Diff](#)



[pgAgent](#)

[Screenshots](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Query Tool

The Query Tool is a powerful, feature-rich environment that allows you to execute arbitrary SQL commands and review the result set. You can access the Query Tool via the *Query Tool* menu option on the *Tools* menu, or through the context menu of select nodes of the Browser tree control.

The Query Tool allows you to:

Issue ad-hoc SQL queries.

Execute arbitrary SQL commands.

Edit the result set of a SELECT query if it is [updatable](#).

Displays current connection and transaction status as configured by the user.

Save the data displayed in the output panel to a CSV file.

Review the execution plan of a SQL statement in either a text, a graphical format or a table format (similar to

<https://explain.depesz.com>).

View analytical information about a SQL statement.

The screenshot shows the PostgreSQL Query Tool interface. The top bar displays the connection information: `postgres/yquery@PostgreSQL 11`. Below the toolbar, the title bar says "Query Editor" and "Query History". The main area contains a SQL query:

```
1 select * from pg_class;
```

Below the query, there are four tabs: "Data Output", "Explain", "Messages", and "Notifications". The "Data Output" tab is selected and displays a table with 9 rows of data. The columns are:

	relname	relnamespace	reltype	relortype	relowner	relam	relfilenode	reltablespace	relpages	reltuples	relvisible
#	name	oid	oid	oid	oid	oid	oid	oid	integer	real	integer
1	pg_statistic	11	11259	0	10	0	2619	0	16	303	16
2	pg_type	11	71	0	10	0	0	0	0	375	0
3	pg_toast_	99	11515	0	10	0	2830	0	0	0	0
4	pg_toast_	99	0	0	10	403	2831	0	1	0	0
5	pg_toast_	99	11516	0	10	0	2832	0	0	0	0
6	pg_toast_	99	0	0	10	403	2833	0	1	0	0
7	pg_toast_	99	11517	0	10	0	2834	0	0	0	0
8	pg_toast_	99	0	0	10	403	2835	0	1	0	0
9	pg_toast_	99	11518	0	10	0	0	0	0	0	0

You can open multiple copies of the Query tool in individual tabs simultaneously. To close a copy of the Query tool, click the *X* in the upper-right hand corner of the tab bar.

The Query Tool features two panels:

The upper panel displays the *SQL Editor*. You can use the panel to enter, edit, or execute a query. It also shows the *History* tab which can be used to view the queries that have been executed in the session, and a *Scratch Pad* which can be used to hold text snippets during editing. If the Scratch Pad is closed, it can be re-opened (or additional ones opened) by right-clicking in the SQL Editor and other panels and adding a new panel.

The lower panel displays the *Data Output* panel. The tabbed panel displays the result set returned by a query, information about a query's execution plan, server messages related to the query's execution and any asynchronous notifications received from the server.

Toolbar

The toolbar is described in the following subsections.

[Query Tool Toolbar](#)

[File Options](#)

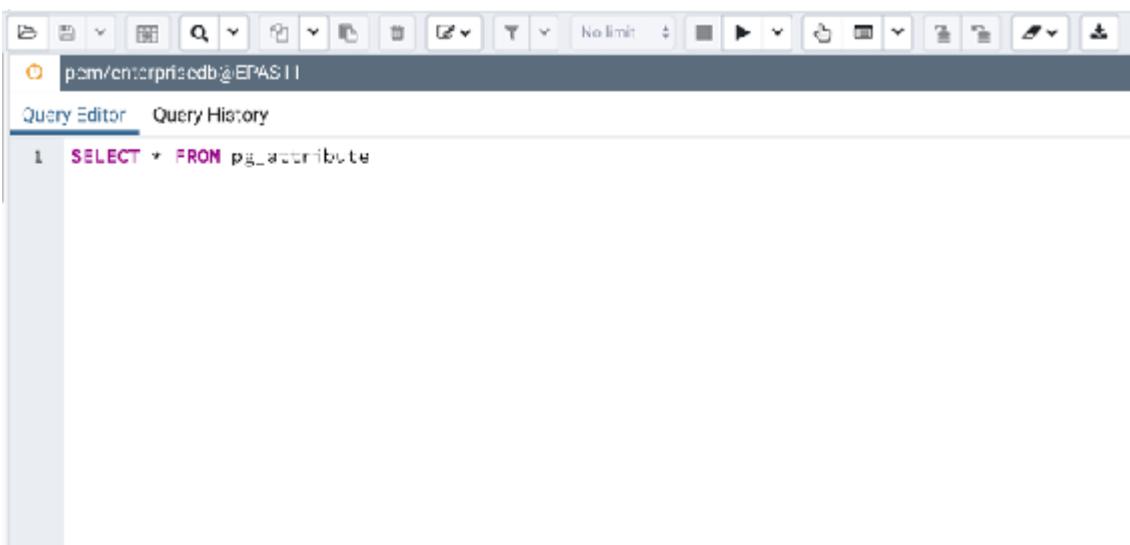
[Editing Options](#)

[View/Edit Data Resultset Control](#)

[Query Execution](#)

The SQL Editor Panel 

The *SQL editor* panel is a workspace where you can manually provide a query, copy a query from another source, or read a query from a file. The SQL editor features syntax coloring and autocompletion.



The screenshot shows a database query editor interface with a toolbar at the top. The connection is set to 'pem/enterprisedb@EPAS11'. The 'Query Editor' tab is active. In the query pane, the user has typed '1 select * from |' and is using the Control+Space key combination to trigger autocomplete. A dropdown menu lists several PL/pgSQL functions: 'bit()', 'char()', 'current_schema()', 'current_timestamp()', 'current_user()', 'decode()', 'delete()', 'each() (hs)', 'family()', and 'interval()'. The 'Data Output' tab is also visible.

The screenshot shows a database query editor interface with a toolbar at the top. The connection is set to 'pem/enterprisedb@EPAS11'. The 'Query Editor' tab is active. The user has run two queries: '1 select GENERATE_SERIES (1,100) as ID, 'ABC' as NAME' and '2 select GENERATE_SERIES (1,100) as ID, 'XYZ' as NAME'. The results are displayed in the 'Data Output' tab, which shows a table with columns 'id' (integer) and 'name' (text). The data consists of two rows of 100 entries each, with IDs ranging from 1 to 100 and names alternating between 'XYZ' and 'ABC'.

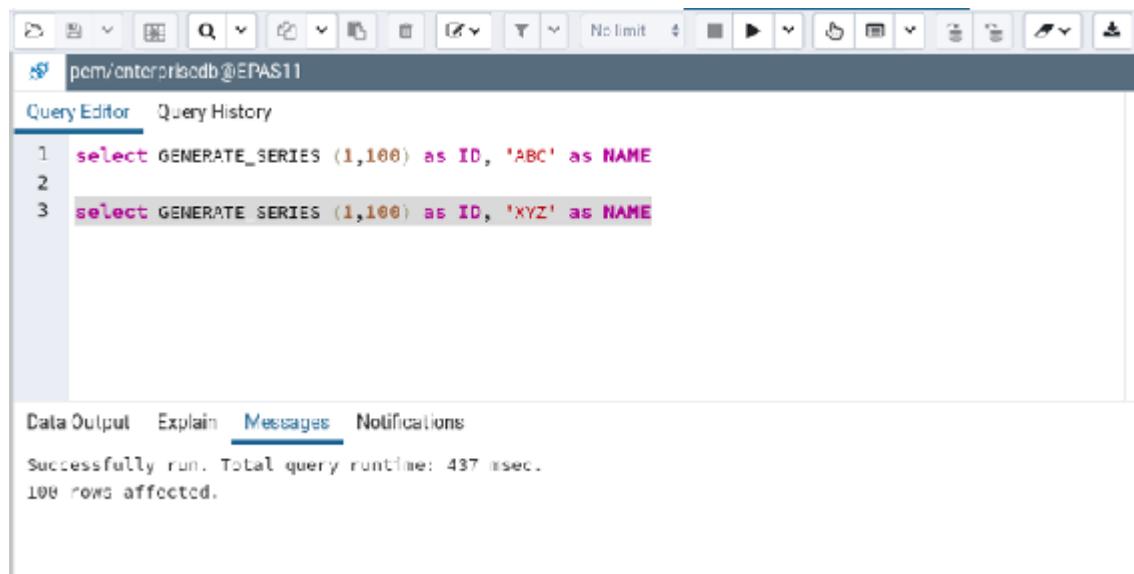
	id	name
	integer	text
1	1	XYZ
2	2	XYZ
3	3	XYZ
4	4	XYZ
5	5	XYZ
6	6	XYZ

To use autocomplete, begin typing your query; when you would like the Query editor to suggest object names or commands that might be next in your query, press the Control+Space key combination. For example, type

“**SELECT * FROM** ” (without quotes, but with a trailing space), and then press the Control+Space key combination to select from a popup menu of autocomplete options.

After entering a query, select the *Execute/Refresh* icon from the toolbar.

The complete contents of the SQL editor panel will be sent to the database server for execution. To execute only a section of the code that is displayed in the SQL editor, highlight the text that you want the server to execute, and click the *Execute/Refresh* icon.



```
1 select GENERATE_SERIES (1,100) as ID, 'ABC' as NAME
2
3 select GENERATE SERIES (1,100) as ID, 'XYZ' as NAME
```

Data Output Explain Messages Notifications

Successfully run. Total query runtime: 437 msec.
100 rows affected.

The screenshot shows the pgAdmin Query Editor interface. At the top, there's a toolbar with various icons. Below it is a header bar with the connection information: "postgres/yoary@PostgreSQL 11". Underneath the header, the tabs "Query Editor" and "Query History" are visible, with "Query Editor" being the active tab. In the main area, there's a code editor containing the following SQL query:

```
1 select * from pg_class
```

Below the code editor is a horizontal navigation bar with four tabs: "Data Output", "Explain", "Messages", and "Notifications". The "Data Output" tab is selected. A table is displayed with the following data:

	relname	relnamespace	reltype	relortype	relowner	relam	relfilenode	reltablespace	relpages	reltuples	relvisible
	name	oid	oid	oid	oid	oid	oid	oid	integer	real	integer
1	pg_statistic	11	11259	0	10	0	2619	0	16	303	16
2	pg_type	11	71	0	10	0	0	0	0	375	0
3	pg_toast_	99	11515	0	10	0	2830	0	0	0	0
4	pg_toast_	99	0	0	10	403	2831	0	1	0	0
5	pg_toast_	99	11516	0	10	0	2832	0	0	0	0
6	pg_toast_	99	0	0	10	403	2833	0	1	0	0
7	pg_toast_	99	11517	0	10	0	2834	0	0	0	0
8	pg_toast_	99	0	0	10	403	2835	0	1	0	0
9	pg_toast_	99	11518	0	10	0	0	0	0	0	0

The message returned by the server when a command executes is displayed on the *Messages* tab. If the command is successful, the *Messages* tab displays execution details.

Options on the *Edit* menu offer functionality that helps with code formatting and commenting:

The auto-indent feature will automatically indent text to the same depth as the previous line when you press the Return key.

Block indent text by selecting two or more lines and pressing the Tab key.

Implement or remove SQL style or toggle C style comment notation within your code.

You can also drag and drop certain objects from the treeview which can save time in typing long object names. Text containing the object name will be fully qualified with schema. Double quotes will be added if required. For functions and procedures, the function name along with parameter names will be pasted in the Query Tool.

The Data Output Panel

The *Data Output* panel displays data and statistics generated by the most recently executed query.

The *Data Output* tab displays the result set of the query in a table format.

You can:

Select and copy from the displayed result set.

A screenshot of the DataGrip IDE interface. At the top, there's a toolbar with various icons. Below it is a menu bar with 'File', 'Edit', 'View', 'Tools', 'Help'. The main area shows a database connection bar with 'db/yosry@PostgreSQL 11'. Underneath is a 'Query Editor' tab and a 'Query History' tab. The 'Query Editor' tab is active, displaying the following SQL code:

```
1 select id, first_name, first_name, last_name, last_name as lname, first_name || ' ' || last_name as name
2 from test_table
```

Below the code is the 'Data Output' tab, which is selected. It shows a table with the following data:

	Id	first_name	first_name	last_name	lname	name
1	1	Yosry	Yosry	Ahmed	Ahmed	Yosry Ahmed

Use the *Execute/Refresh* options to retrieve query execution information and set query execution options.

Use the *Download as CSV* icon to download the content of the *Data Output* tab as a comma-delimited file.

Edit the data in the result set of a SELECT query if it is updatable.

A result set is updatable if:

All columns are either selected directly from a single table, or are not table columns at all (e.g. concatenation of 2 columns). Only columns that are selected directly from the table are editable, other columns are read-only.

All the primary key columns or OIDs of the table are selected in the result set.

Any columns that are renamed or selected more than once are also read-only.

Editable and read-only columns are identified using pencil and lock icons (respectively) in the column headers.

The psycopg2 driver version should be equal to or above 2.8 for updatable query result sets to work.

An updatable result set is identical to the [Data Grid](#) in View/Edit Data mode, and can be modified in the same way.

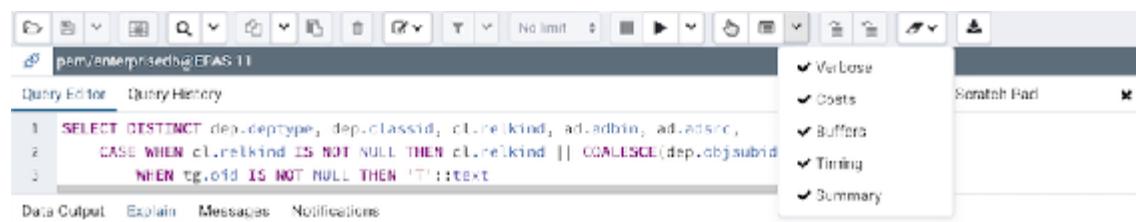
If Auto-commit is off, the data changes are made as part of the ongoing transaction, if no transaction is ongoing a new one is initiated. The data changes are not committed to the database unless the transaction is committed.

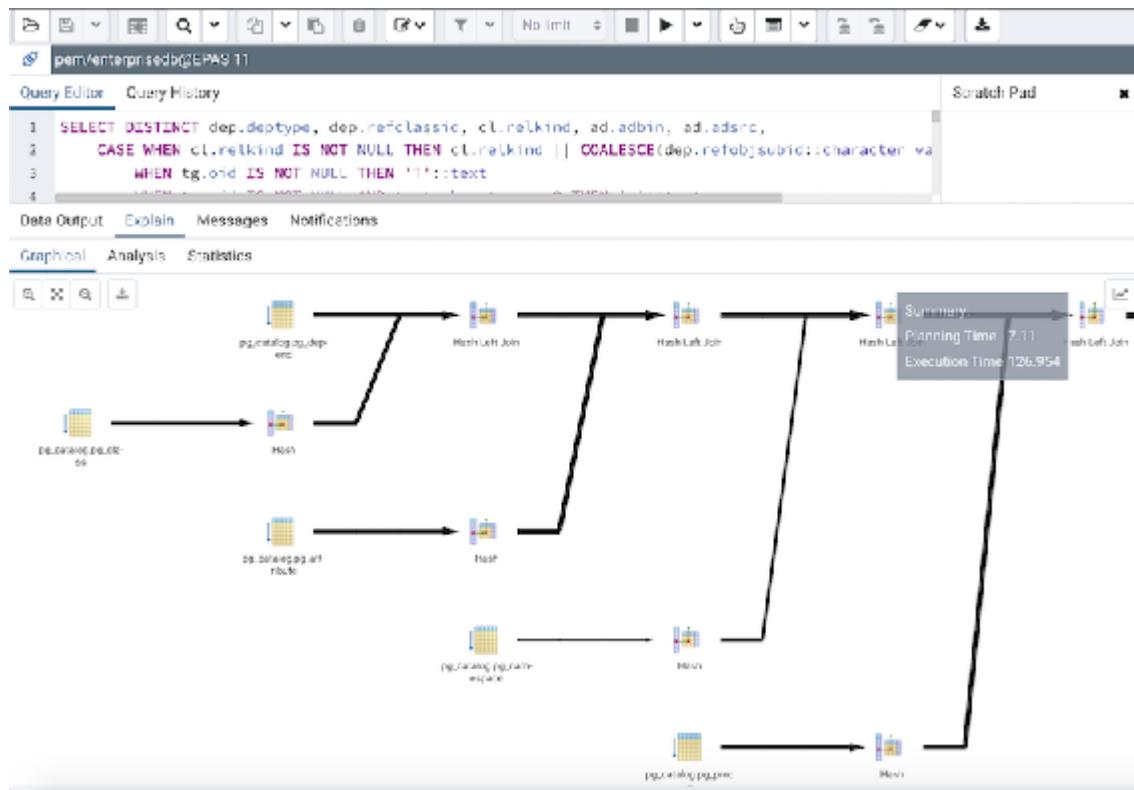
If any errors occur during saving (for example, trying to save NULL into a column with NOT NULL constraint) the data changes are rolled back to an automatically created SAVEPOINT to ensure any previously executed queries in the ongoing transaction are not rolled back.

All rowsets from previous queries or commands that are displayed in the *Data Output* panel will be discarded when you invoke another query; open another Query Tool tab to keep your previous results available.

Explain Panel

To generate the *Explain* or *Explain Analyze* plan of a query, click on *Explain* or *Explain Analyze* button in the toolbar.





More options related to *Explain* and *Explain Analyze* can be selected from the drop down on the right side of *Explain Analyze* button in the toolbar.

Please note that pgAdmin generates the *Explain [Analyze]* plan in JSON format.

On successful generation of *Explain* plan, it will create three tabs/panels under the Explain panel.

Graphical

Please note that *EXPLAIN VERBOSE* cannot be displayed graphical y.

Hover over an icon on the *Graphical* tab to review information about that item; a popup window will display information about the selected object.

For information on JIT statistics, triggers and a summary, hover over the icon on top-right corner; a similar popup window will be displayed when appropriate.

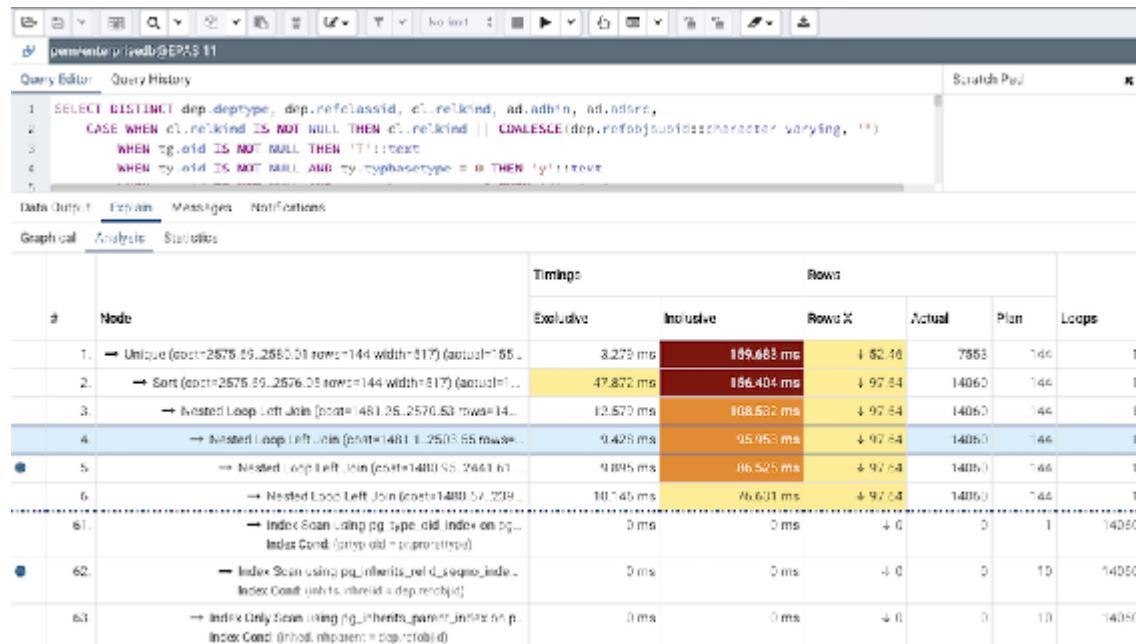
Use the download button on top left corner of the *Explain* canvas to download the plan as an SVG file.

Note: Download as SVG is not supported on Internet Explorer.

Note that the query plan that accompanies the *Explain analyze* is available on the *Data Output* tab.

Table

Table tab shows the plan details in table format, it generates table format similar to explain.depsez.com. Each row of the table represent the data for a *Explain Plan Node*. It may contain the node information, exclusive timing, inclusive timing, actual vs planned rows differences, actual rows, planned rows, loops.



The screenshot shows the pgAdmin interface with the Explain tab selected. At the top, there's a toolbar and a menu bar. Below that is a Query Editor window containing a SQL query:

```

1 SELECT DISTINCT dep.deptype, dep.refclassid, cl.refclkind, ad.adbh, ad.adssn,
2 CASE WHEN cl.refclkind IS NOT NULL THEN cl.refclkind || COALESCE(dep.refobjsubid||character varying, '')
3 WHEN tg.ref IS NOT NULL THEN 'T'||tg.ref
4 WHEN my.ref IS NOT NULL AND my.myphaserotype = 0 THEN 'Y'||my.ref
5

```

Below the Query Editor is a Data Output tab where the explain plan is displayed as a table. The table has columns for # (Node ID), Node (the plan node description), Timings (Exclusive, Inclusive), Rows (Row X, Actual, Plan), and Loops. The table data is as follows:

#	Node	Timings		Rows			Loops
		Exclusive	Inclusive	Row X	Actual	Plan	
1.	→ Unique (cost=2676.19..2680.01 rows=144 width=817) (actual=166 ..)	3.279 ms	189.683 ms	1 52.16	7863	146	1
2.	→ Sort (cost=2676.39..2676.05 rows=144 width=817) (actual=1 ..)	47.872 ms	186.404 ms	1 97.64	14060	146	1
3.	→ Nested Loop Left Join (cost=1481.26..2570.53 rows=14 ..)	12.570 ms	108.532 ms	1 97.64	14060	146	1
4.	→ Nested Loop Left Join (cost=1481.1..2503.55 rows=14 ..)	9.428 ms	95.963 ms	1 97.64	14060	146	1
5.	→ Nested Loop Left Join (cost=1480.95..2661.61 ..)	9.095 ms	86.525 ms	1 97.64	14060	146	1
6.	→ Nested Loop Left Join (cost=1480.57..239 ..)	10.745 ms	86.631 ms	1 97.64	14060	146	1
61.	→ Index Scan using pg_type.old_index on pg_type (Index Cond: (typoid = pg_typeof(p)))	0 ms	0 ms	+ 0	0	1	14060
62.	→ Index Scan using pg_inherits_rel_d_sequ_index (Index Cond: (inhrelid >= dep.relid))	0 ms	0 ms	+ 0	0	10	14060
63.	→ Index Only Scan using pg_inherits_garbage_index on pg_inherits (Index Cond: (inhrelid >= dep.relid))	0 ms	0 ms	+ 0	0	10	14060

Statistics per Node Type				Statistics per Table			
Node type	Count	Time spent	% of query	Table name	Scan count	Total time	% of query
Bitmap Heap Scan	1	0.008 ms	0.01%	pg_catalog.pg_attrdef	1	0 ms	0%
Bitmap Index Scan	1	0.021 ms	0.02%	Index Scan	1	0 ms	0%
Hash	17	3.096 ms	2.45%	pg_catalog.pg_attribute	1	1.522 ms	1.21%
Hash Inner Join	1	3.593 ms	2.84%	Seq Scan	1	1.522 ms	100%
Hash Left Join	15	42.634 ms	33.7%	pg_catalog.pg_class	4	0.744 ms	0.59%
Hash Right Join	1	0.15 ms	0.12%	Bitmap Heap Scan	1	0.008 ms	1.08%
Index Only Scan	1	0 ms	0%	Seq Scan	3	0.736 ms	98.93%
Index Scan	3	0 ms	0%	pg_catalog.pg_constraint	1	0.081 ms	0.07%
Nested Loop Left Join	4	34.451 ms	27.23%	Seq Scan	1	0.081 ms	100%
Seq Scan	17	5.448 ms	4.31%	pg_catalog.pg_depend	1	1.518 ms	1.28%
Sort	1	34.451 ms	2.7%	Seq Scan	1	1.518 ms	100%
Unique	1	2.985 ms	2.36%				

background color of the exclusive, inclusive, and Rows X columns may vary based on the difference between actual vs planned.

If percentage of the exclusive/inclusive timings of the total query time is: > 90 - Red color > 50 - Orange (between red and yellow) color > 10 -

Yellow color

If planner mis-estimated number of rows (actual vs planned) by 10 times -

Yellow color 100 times - Orange (between Red and Yellow) color 1000

times - Red color

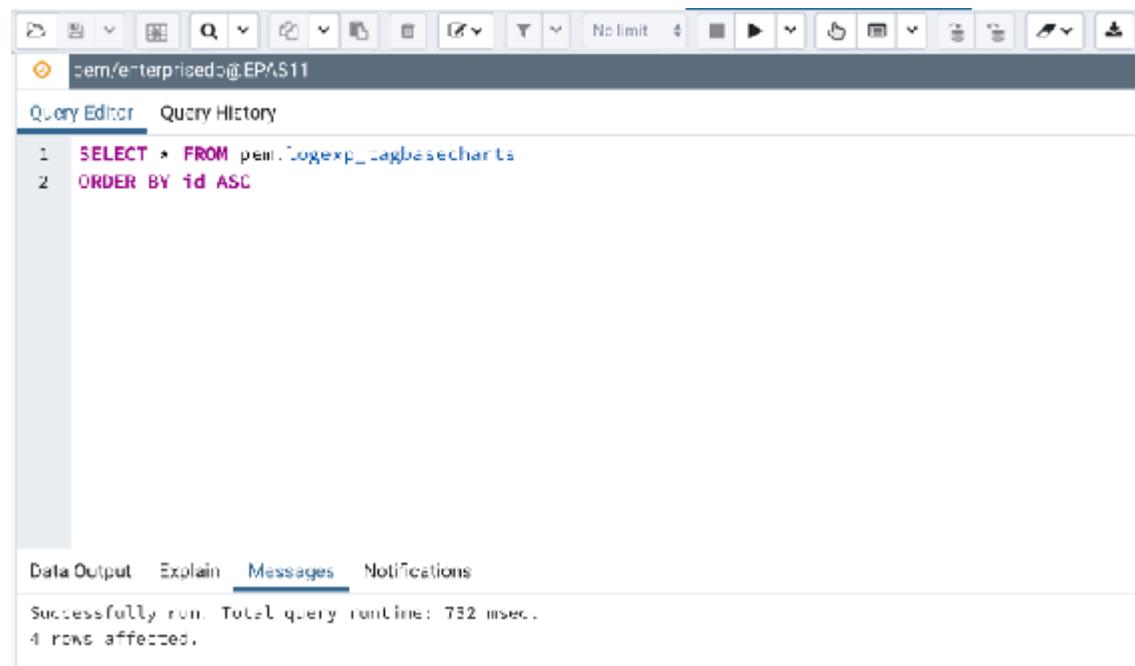
Statistics

Statistics tab shows two tables: 1. Statistics per Plan Node Type 2.

Statistics per Table

Messages Panel

Use the *Messages* tab to view information about the most recently executed query:

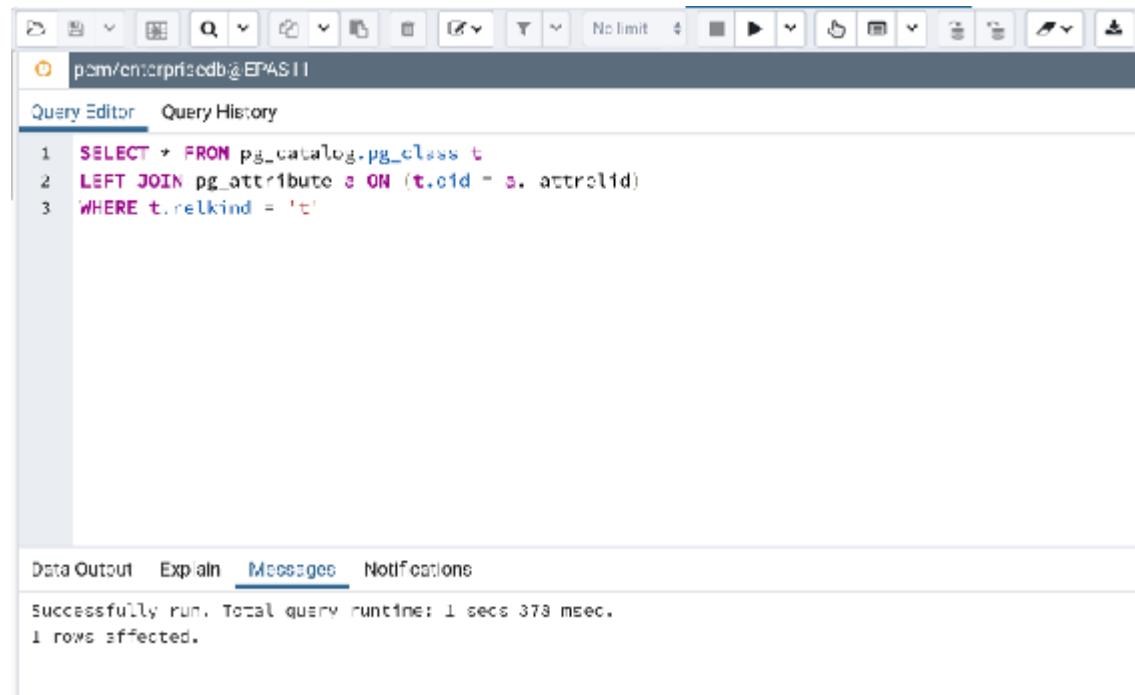


The screenshot shows the pgAdmin III interface with the following details:

- Toolbar:** Standard pgAdmin toolbar with various icons.
- Connection Bar:** Shows the connection as "pm/enterprisedb@EPAS11".
- Query Editor:** Contains the following SQL code:

```
1 SELECT * FROM pgexp_logbasecharts
2 ORDER BY id ASC
```
- Messages Tab:** The active tab, showing the output of the query execution:

```
Successfully run. Total query runtime: 732 msec.
4 rows affected.
```
- Bottom Navigation:** Buttons for Data Output, Explain, Messages (underlined), Notifications.



The screenshot shows the pgAdmin III interface with the following details:

- Toolbar:** Standard pgAdmin toolbar with various icons.
- Connection Bar:** Shows the connection as "pm/enterprisedb@EPAS11".
- Query Editor:** Contains the following SQL code:

```
1 SELECT * FROM pg_catalog.pg_class t
2 LEFT JOIN pg_attribute a ON (t.oid = a.attrelid)
3 WHERE t.relkind = 't'
```
- Messages Tab:** The active tab, showing the output of the query execution:

```
Successfully run. Total query runtime: 1 secs 379 msec.
1 rows affected.
```
- Bottom Navigation:** Buttons for Data Output, Explain, Messages (underlined), Notifications.

postgres/postgres@PostgreSQL 12

Show queries generated internally by pgAdmin? Yes

Today - 16/08/2019

```

SELECT * FROM test;
17:02:15
SELECT * FROM test;
17:02:17
COMMIT;
17:02:47
INSERT INTO public.test ( id, name) VALUES ( '2'::bigint,..
17:02:47
INSERT INTO public.test ( id, name) VALUES ( '1'::bigint,..
17:02:47
BEGIN;
17:02:47

```

16/08/2019 17:02:19 1 38 msec

Date Rows Affected Duration

[Copy](#) [Copy to Query Editor](#)

SELECT * FROM test;

Messages

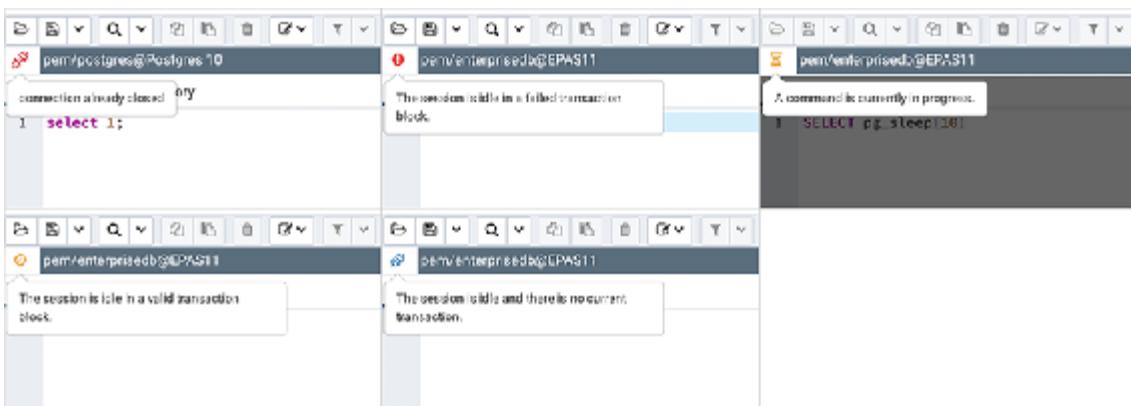
Successfully run. Total query runtime: 38 msec.
1 rows affected.

If the server returns an error, the error message will be displayed on the *Messages* tab, and the syntax that caused the error will be underlined in the SQL editor. If a query succeeds, the *Messages* tab displays how long the query took to complete and how many rows were retrieved: **Query History Panel**

Use the *Query History* tab to review activity for the current session: The Query History tab displays information about recent commands: The date and time that a query was invoked.

The text of the query.

The number of rows returned by the query.



The amount of time it took the server to process the query and return a result set.

Messages returned by the server (not noted on the *Messages* tab).

The source of the query (indicated by icons corresponding to the toolbar).

You can show or hide the queries generated internally by pgAdmin (during

'View/Edit Data' or 'Save Data' operations).

To erase the content of the *Query History* tab, select *Clear history* from the *Clear* drop-down menu.

Query History is maintained across sessions for each database on a per-user basis when running in Query Tool mode. In View/Edit Data mode, history is not retained. By default, the last 20 queries are stored for each database. This can be adjusted in *config_local.py* by overriding the *MAX_QUERY_HIST_STORED* value. See the [Deployment](#) section for more information.

Connection Status 

Use the *Connection status* feature to view the current connection and transaction status by clicking on the status icon in the Query Tool:

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) » [Managing Cluster Objects](#) » [previous](#) [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Resource Group Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
 - [Database Dialog](#)
 - [Move Objects Dialog](#)
 - [Resource Group Dialog](#)
 - [Login/Group/Role Dialog](#)
 - [Tablespace Dialog](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Resource Group Dialog

Use the Resource Group dialog to create a resource group and set values for its resources. A resource group is a named, global group on which various resource usage limits can be defined. The resource group is accessible from all databases in the cluster. To use the Resource Group dialog, you must have superuser privileges. Please note that resource groups are supported when connected to EDB Postgres Advanced Server; for more information about using resource groups, please see the EDB Postgres Advanced Server Guide, available at:

<http://www.enterprisedb.com/>

Fields used to create a resource group are located on the General tab. The SQL tab displays the SQL code generated by your selections on the Resource Group dialog.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Cluster Objects](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Resource Group Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Database Dialog](#)

[Move Objects Dialog](#)



Resource Group Dialog

[Login/Group Role Dialog](#)

[Latest Docs](#)

[Tablespace Dialog](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[Ask a](#)

[Management Basics](#)

[Question](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)



[pgAdmin Project Contributions](#)

[Report a Bug](#)

[Release Notes](#)

[Licence](#)



Resource Group Dialog

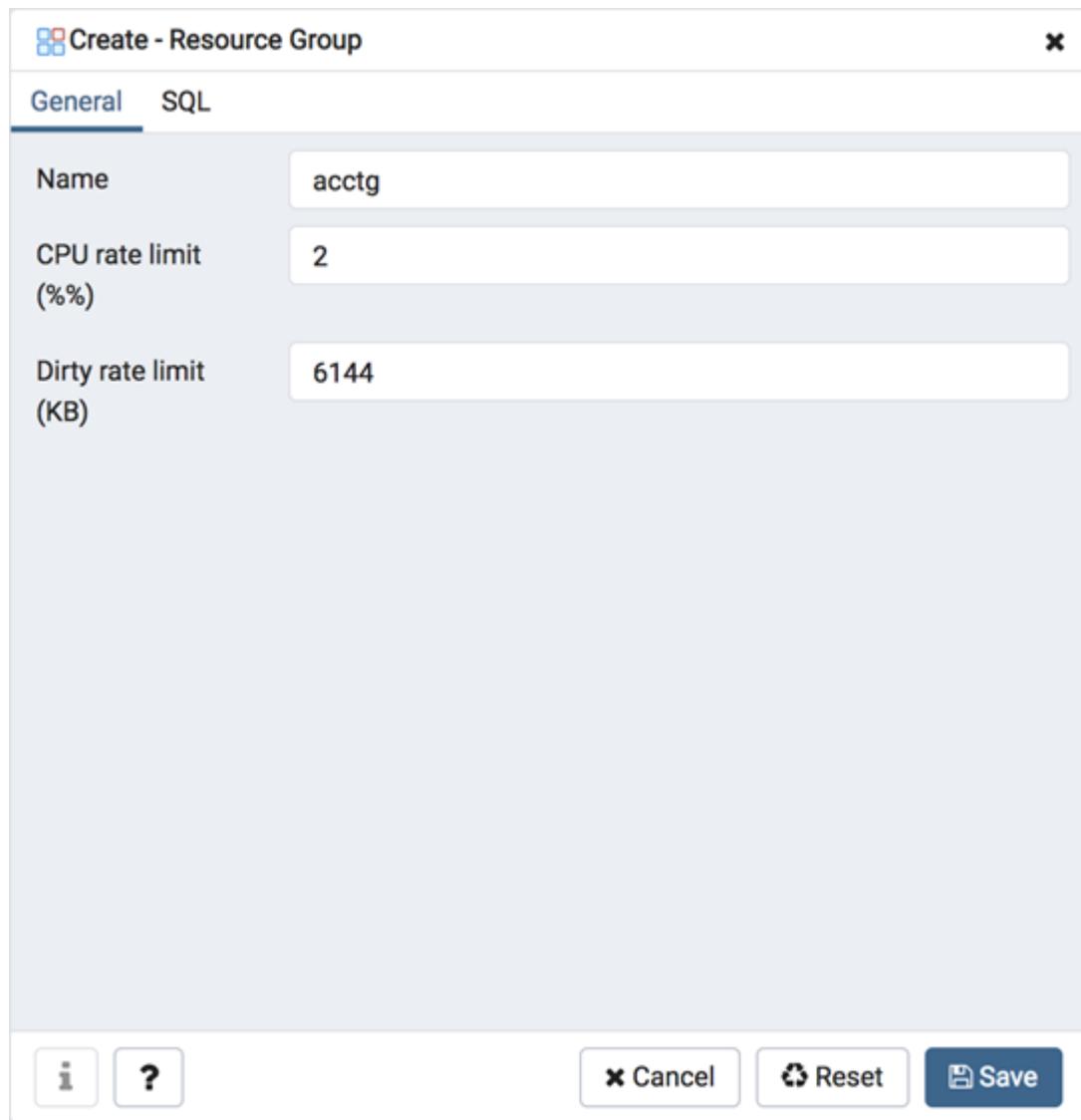
Screenshots

Use the *Resource Group* dialog to create a resource group and set values for its resources. A resource group is a named, global group on which various resource usage limits can be defined. The resource group is accessible from all databases in the cluster. To use the *Resource Group* dialog, you must have superuser privileges. Please note that resource groups are supported when connected to EDB Postgres Advanced Server; for more information about using resource groups, please see the EDB

Postgres Advanced Server Guide, available at:

<http://www.enterprisedb.com/>

Fields used to create a resource group are located on the *General* tab. The *SQL* tab displays the SQL code generated by your selections on the *Resource Group* dialog.

A screenshot of a 'Create - Resource Group' dialog box. The 'General' tab is selected. The 'Name' field contains 'acctg'. The 'CPU rate limit (%)' field contains '2'. The 'Dirty rate limit (KB)' field contains '6144'. At the bottom, there are informational icons (i, ?), a 'Cancel' button, a 'Reset' button, and a 'Save' button.

Use the fields on the *General* tab to specify resource group attributes: Use the *Name* field to add a descriptive name for the resource group.

This name will be displayed in the tree control.

Use the *CPU rate limit (%)* field to set the value of the CPU rate limit resource type assigned to the resource group. The valid range for a CPU rate limit is from 0 to 1.67772e+07. The default value is 0.

Use the *Dirty rate limit (KB)* field to set the value of the dirty rate limit resource type assigned to the resource group. The valid range for a

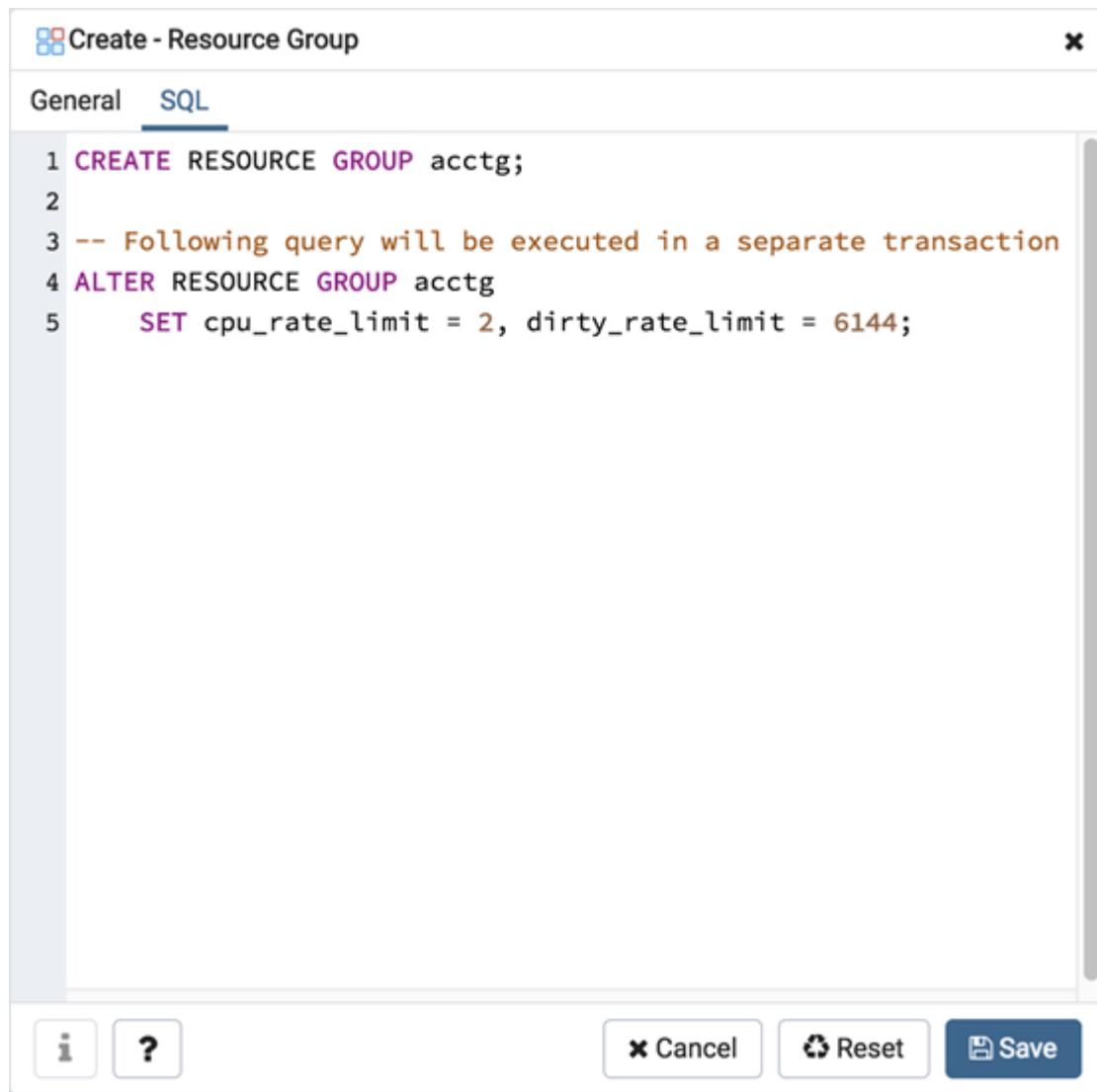
dirty rate limit is from 0 to 1.67772e+07. The default value is 0.

Click the *SQL* tab to continue.

Your entries in the *Resource Group* dialog generate a SQL command. Use the *SQL* tab for review; revisit the *General* tab to make any changes to the SQL command.

Example 

The following is an example of the sql command generated by selections made in the *Resource Group* dialog:



The screenshot shows the 'Create - Resource Group' dialog box. The 'SQL' tab is selected. The SQL code generated is:

```
1 CREATE RESOURCE GROUP acctg;
2
3 -- Following query will be executed in a separate transaction
4 ALTER RESOURCE GROUP acctg
5     SET cpu_rate_limit = 2, dirty_rate_limit = 6144;
```

At the bottom of the dialog, there are three buttons: 'Cancel', 'Reset', and a blue 'Save' button.

The example creates a resource group named *acctg* that sets *cpu_rate_limit* to 2, and *dirty_rate_limit* to 6144.

Click the Info button (*i*) to access online SQL syntax reference material.

Click the Help button (*?*) to access online documentation about Resource Groups.

Click the Save button to save work.

Click the Cancel button to exit without saving work.

Click the Reset button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Creation or Modifying a Table](#) > [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Rule Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
 - [Check Dialog](#)
 - [Column Dialog](#)
 - [Compound Trigger Dialog](#)
 - [Exclusion Constraint Dialog](#)
 - [Foreign key Dialog](#)
 - [Index Dialog](#)
 - [Primary key Dialog](#)
 - [Rule Dialog](#)
 - [Table Dialog](#)
 - [Trigger Dialog](#)
 - [Unique Constraint Dialog](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Rule Dialog

Use the Rule dialog to define or modify a rule for a specified table or view. A PostgreSQL rule allows you to define an additional action that will be performed when a SELECT, INSERT, UPDATE, or DELETE is performed against a table.

The Rule dialog organizes the development of a rule through the General, Definition, Condition, Commands tabs. The SQL tab displays the SQL code generated by dialog selections.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Creating or Modifying a Table](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Rule Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[□](#)

[Check Dialog](#)

[Column Dialog](#)

[Latest Docs](#)

[Compound Trigger Dialog](#)

[Exclusion Constraint Dialog](#)

[□](#)

[Foreign key Dialog](#)

[Index Dialog](#)

[Ask a](#)

[Primary key Dialog](#)

[Question](#)

Rule Dialog

[Table Dialog](#)

[□](#)

[Trigger Dialog](#)

[Report a Bug](#)

[Unique Constraint Dialog](#)

[Management Basics](#)

[Backup and Restore](#)



[Developer Tools](#)

[Screenshots](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Rule Dialog

Use the *Rule* dialog to define or modify a rule for a specified table or view.

A PostgreSQL rule allows you to define an additional action that will be performed when a SELECT, INSERT, UPDATE, or DELETE is performed against a table.

The *Rule* dialog organizes the development of a rule through the *General*, *Definition*, *Condition*, *Commands* tabs. The *SQL* tab displays the SQL

code generated by dialog selections.

Create - rule

X

General Definition Condition Commands SQL

Name replace_expression_answer

Comment Rule to replace expression answer

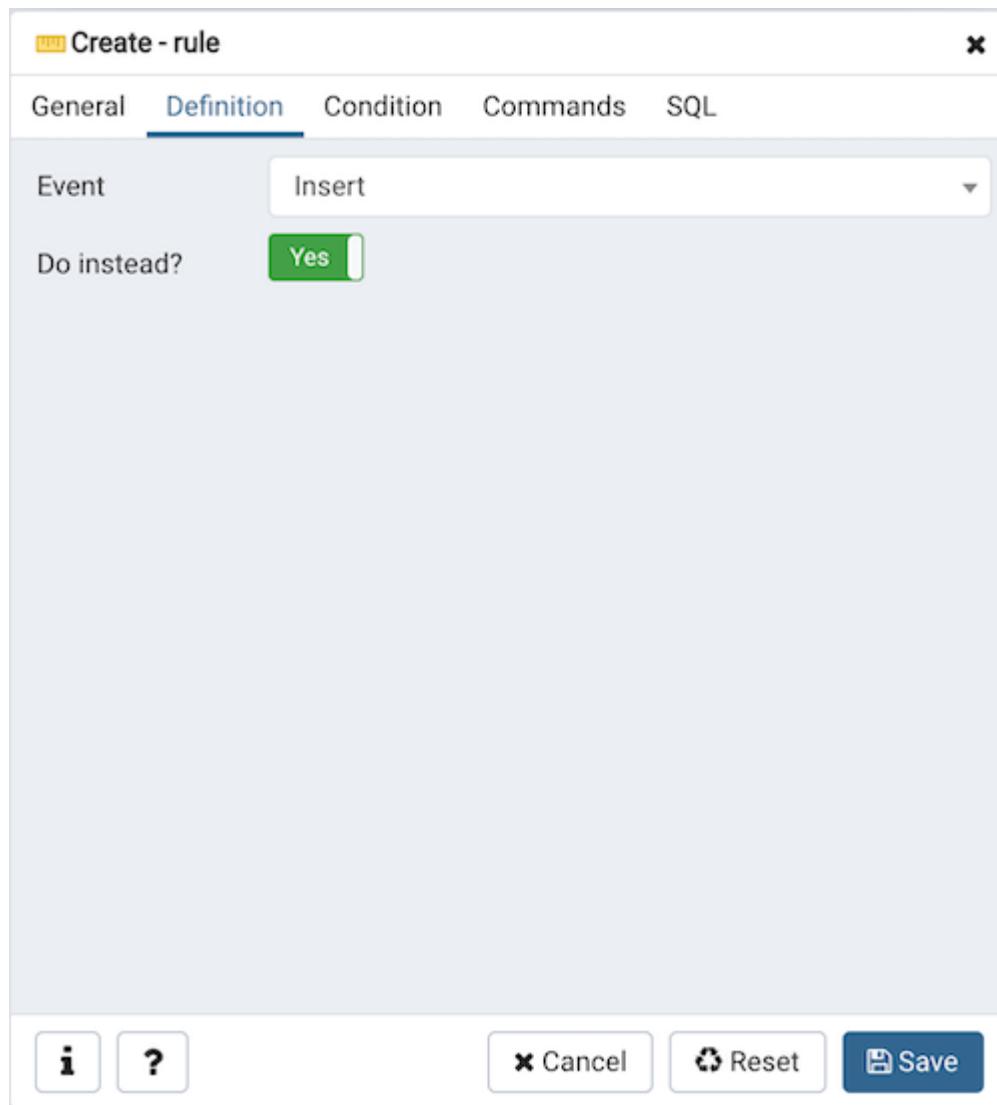
i

?

X Cancel

Reset

Save



Use the fields in the *General* tab to identify the rule: Use the *Name* field to add a descriptive name for the rule. The name will be displayed in the *pgAdmin* tree control. Multiple rules on the same table are applied in alphabetical name order.

Store notes about the rule in the *Comment* field.

Click the *Definition* tab to continue.

Use the fields in the *Definition* tab to write parameters: Click inside the *Event* field to select the type of event that will invoke the rule; event may be *Select*, *Insert*, *Update*, or *Delete*.

Create - rule



General Definition Condition Commands SQL

```
1 (EXISTS (SELECT 1
2   FROM expression_answer expression_answer_1
3   WHERE expression_answer_1.id = new.id
4   AND expression_answer_1.expression_id = new.expression_id
5 )
6 )|
```



Cancel

Reset

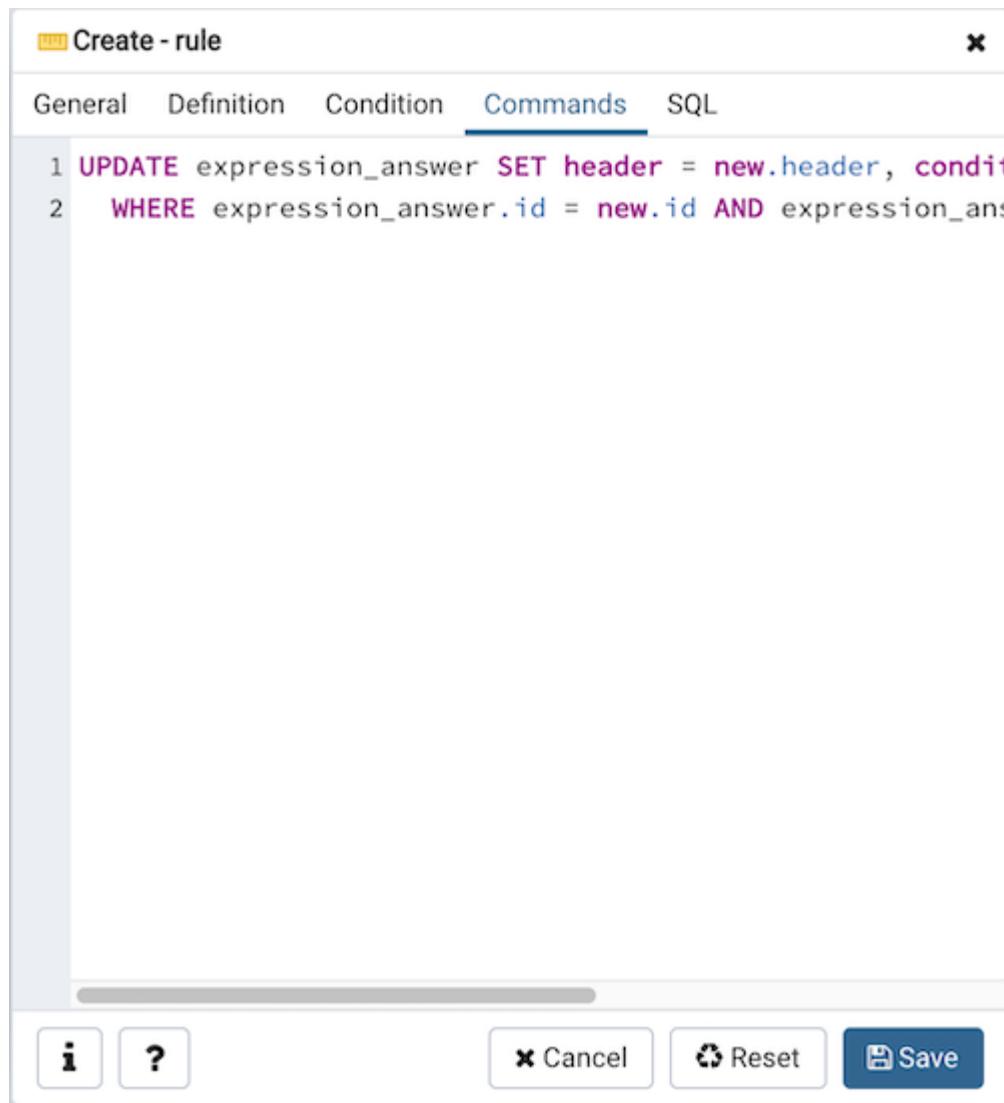
Save

Create - rule

General Definition Condition **Commands** SQL

```
1 UPDATE expression_answer SET header = new.header, condit
2 WHERE expression_answer.id = new.id AND expression_ans
```

i **?** **Cancel** **Reset** **Save**



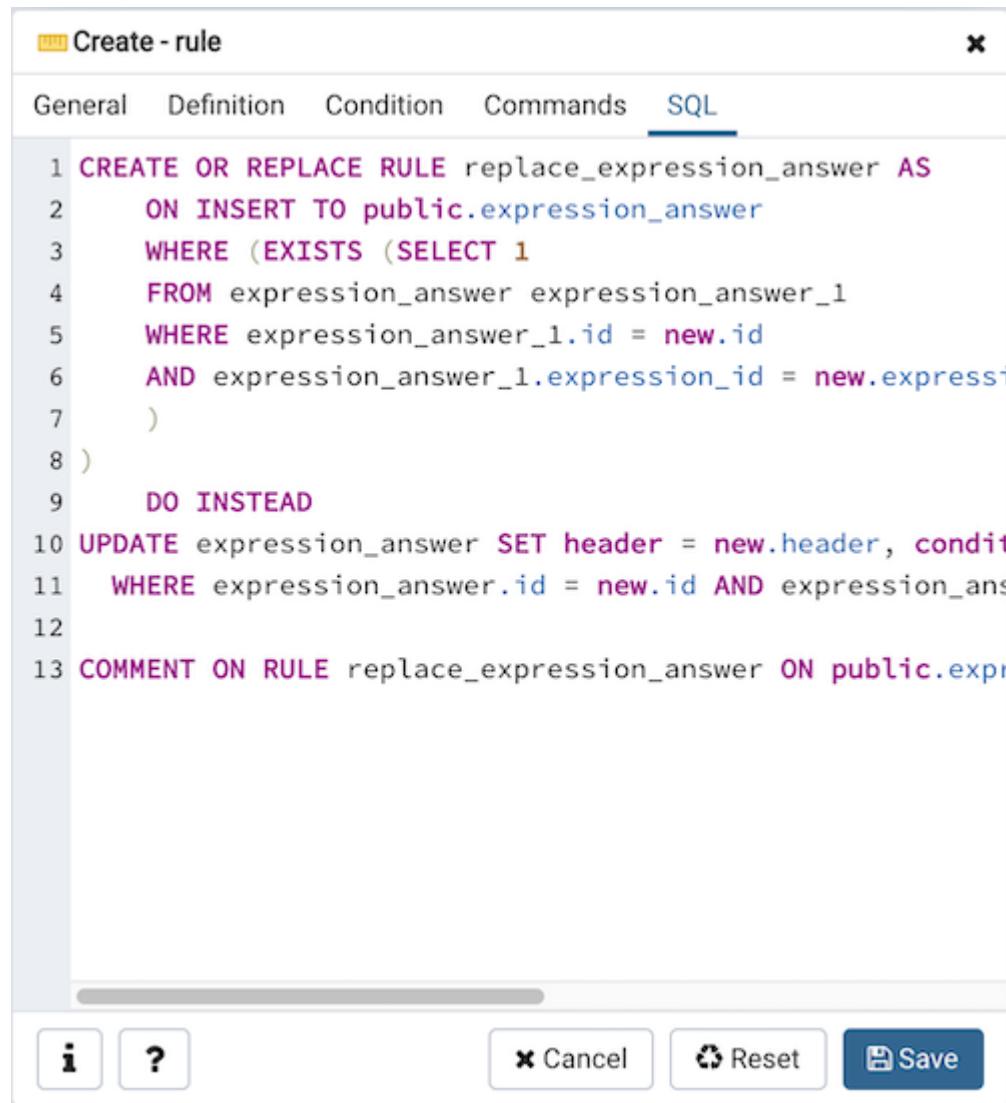
Move the *Do Instead* switch to *Yes* indicate that the commands should be executed instead of the original command; if Do Instead specifies *No*, the rule wil be invoked in addition to the original command.

Click the *Condition* tab to continue.

Specify a SQL conditional expression that returns a boolean value in the editor.

Click the *Commands* tab to continue.

Provide a command in the editor that defines the action performed by the rule.



The screenshot shows the 'Create - rule' dialog box. The 'SQL' tab is selected. The code area contains the following SQL command:

```
1 CREATE OR REPLACE RULE replace_expression_answer AS
2   ON INSERT TO public.expression_answer
3     WHERE (EXISTS (SELECT 1
4       FROM expression_answer expression_answer_1
5         WHERE expression_answer_1.id = new.id
6           AND expression_answer_1.expression_id = new.expression_id
7         )
8       )
9     DO INSTEAD
10    UPDATE expression_answer SET header = new.header, condition = new.condition
11      WHERE expression_answer.id = new.id AND expression_answer.expression_id = new.expression_id
12
13 COMMENT ON RULE replace_expression_answer ON public.expression_answer IS 'Replaces the expression answer with the new one';
```

At the bottom of the dialog, there are buttons for 'Cancel', 'Reset', and 'Save'.

Click the *SQL* tab to continue.

Your entries in the *Rule* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example [¶](#)

The following is an example of the sql command generated by user selections in the *Rule* dialog:

The example sends a notification when an UPDATE executes against a table.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Backup and Restore](#) >

[previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
 - [Backup Dialog](#)
 - [Backup Globals Dialog](#)
 - [Backup Server Dialog](#)
 - [Restore Dialog](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Restore Dialog

The *Restore* dialog provides an easy way to use a Custom, tar, or Directory format backup taken with the pgAdmin *Backup* dialog to recreate a database or database object. The *Backup* dialog invokes options of the `pg_dump` client utility; the *Restore* dialog invokes options of the `pg_restore` client utility.

You can use the *Query Tool* to play back the script created during a plain-text backup made with the *Backup* dialog. For more information about backing up or restoring, please refer to the documentation for [pg_dump](#) or [pg_restore](#).

Restore (Database: edb)

General Restore options

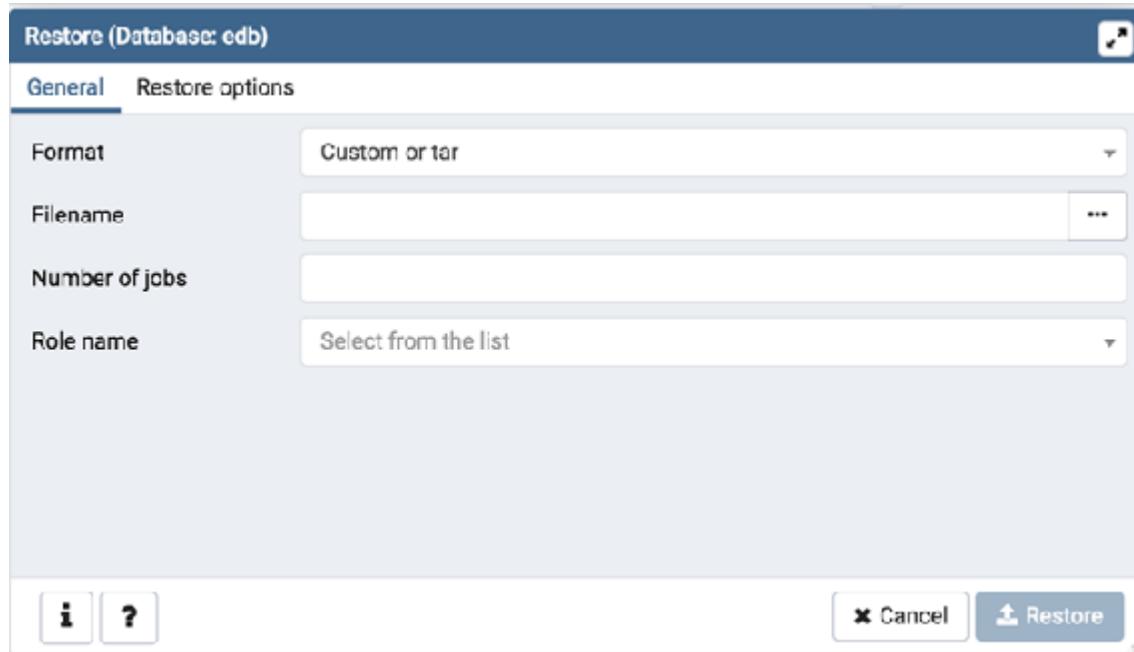
Format: Custom or tar

Filename:

Number of jobs:

Role name: Select from the list

Cancel **Restore**



Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Backup and Restore](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[FAQ](#)

[Management Basics](#)

[Backup and Restore](#)



[Backup Dialog](#)

[Backup Globals Dialog](#)

[Latest Docs](#)

[Backup Server Dialog](#)

Restore Dialog



[Developer Tools](#)

[pgAgent](#)

[Ask a](#)

[pgAdmin Project Contributions](#)

[Question](#)

[Release Notes](#)

[Licence](#)



Restore Dialog

[Report a Bug](#)

The *Restore* dialog provides an easy way to use a Custom, tar, or



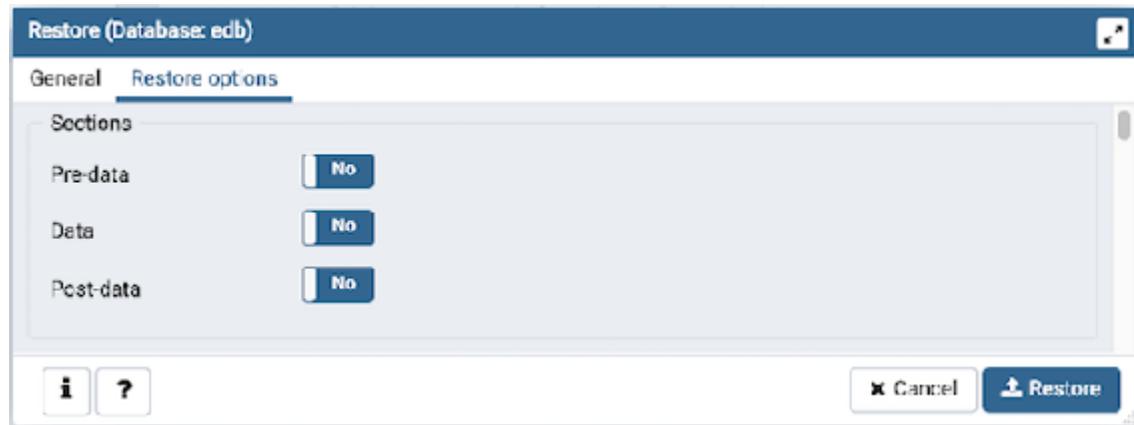
Directory format backup taken with the pgAdmin *Backup* dialog to recreate a database or database object. The *Backup* dialog invokes

[Screenshots](#)

options of the `pg_dump` client utility; the *Restore* dialog invokes options of the `pg_restore` client utility.

You can use the *Query Tool* to play back the script created during a plain-text backup made with the *Backup* dialog. For more information about backing up or restoring, please refer to the documentation for [pg_dump](#) or

[pg_restore](#).



Use the fields on the *General* tab to specify general information about the restore process:

Use the drop-down listbox in the *Format* field to select the format of your backup file.

Select *Custom or tar* to restore from a custom archive file to create a copy of the backed-up object.

Select *Directory* to restore from a compressed directory-format archive.

Enter the complete path to the backup file in the *Filename* field.

Optional, select the *Browser* icon (el ipsis) to the right to navigate into a directory and select the file that contains the archive.

Use the *Number of Jobs* field to specify if pg_restore should use multiple (concurrent) jobs to process the restore. Each job uses a separate connection to the server.

Use the drop-down listbox next to *Rolename* to specify the role that will be used to authenticate with the server during the restore process.

Click the *Restore options* tab to continue. Use the fields on the *Restore options* tab to specify options that correspond to pg_restore options.

Use the switches in the Sections box to specify the content that will be restored:

Move the switch next to *Pre-data* to the Yes position to restore all data definition items not included in the data or post-data item lists.

Move the switch next to *Data* to the Yes position to restore actual table data, large-object contents, and sequence values.

Move the switch next to *Post-data* to the Yes position to restore definitions of indexes, triggers, rules, and constraints (other than validated check constraints).





Use the switches in the Type of objects box to specify the objects that will be restored:

Move the switch next to *Only data* to the Yes position to limit the restoration to data.

Move the switch next to *Only schema* to limit the restoration to schema-level database objects.

Use the switches in the Do not save box to specify which objects will not be restored:

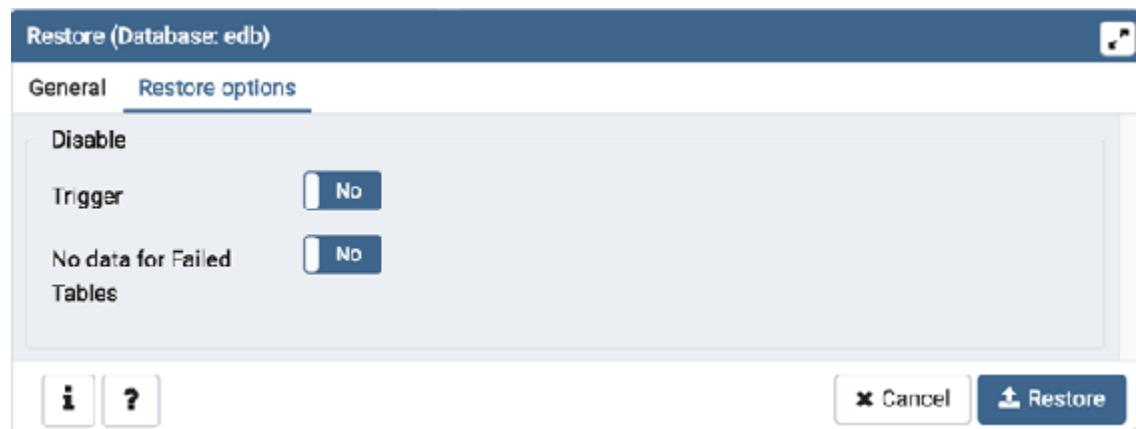
Move the switch next to *Owner* to the Yes position to exclude commands that set object ownership.

Move the switch next to *Privilege* to the Yes position to exclude commands that create access privileges.

Move the switch next to *Tablespace* to the Yes position to exclude tablespaces.

Move the switch next to *Comments* to the Yes position to exclude commands that set the comments. Note: This option is visible only for database server greater than or equal to 11.

Use the switches in the Queries box to specify the type of statements that should be included in the restore:



Move the switch next to *Include CREATE DATABASE statement* to the Yes position to include a command that creates a new database before performing the restore.

Move the switch next to *Clean before restore* to the Yes position to drop each existing database object (and data) before restoring.

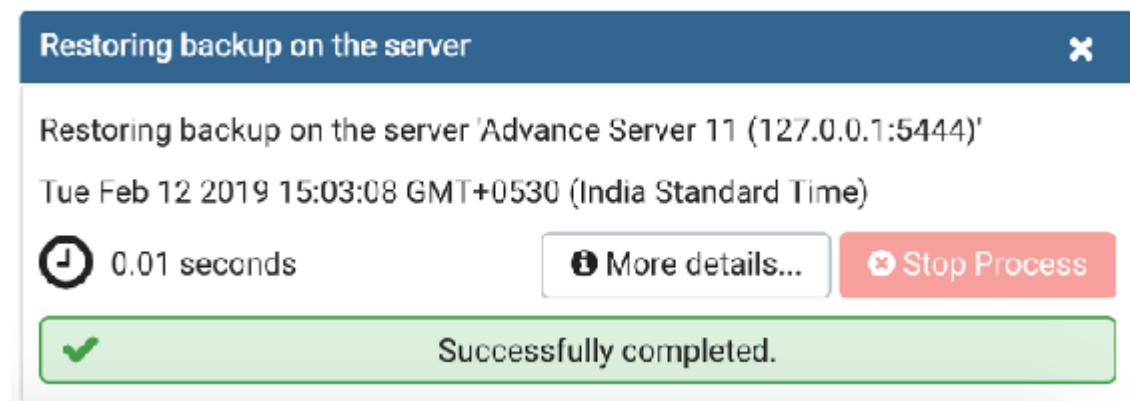
Move the switch next to *Single transaction* to the Yes position to execute the restore as a single transaction (that is, wrap the emitted commands in *BEGIN/COMMIT*). This ensures that either all the commands complete successfully, or no changes are applied. This option implies *-exit-on-error*.

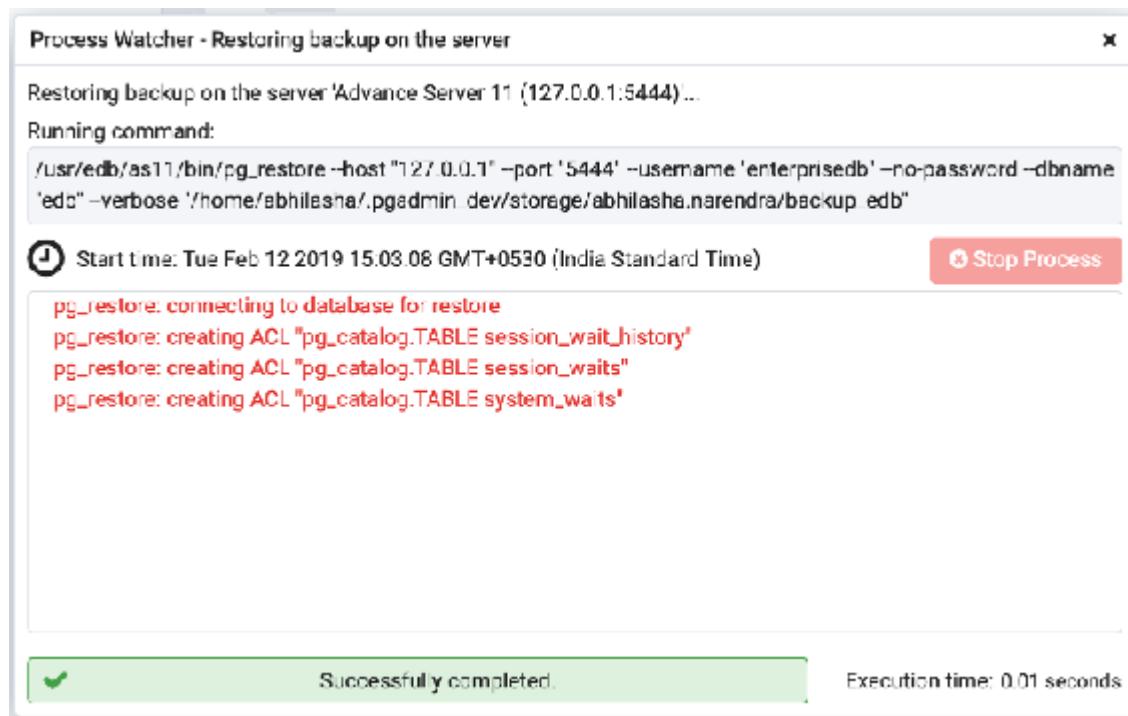
Use the switches in the Disable box to specify the type of statements that should be excluded from the restore:

Move the switch next to *Trigger* (active when creating a data-only restore) to the Yes position to include commands that will disable triggers on the target table while the data is being loaded.

Move the switch next to *No data for Failed Tables* to the Yes position to ignore data that fails a trigger.

Use the switches in the Miscellaneous/Behavior box to specify miscellaneous restore options:





Move the switch next to *Verbose messages* to the *No* position to instruct *pg_restore* to exclude verbose messages.

Move the switch next to *Use SET SESSION AUTHORIZATION* to the *Yes* position to include a statement that wil use a *SET*

SESSION AUTHORIZATION command to determine object ownership (instead of an *ALTER OWNER* command).

Move the switch next to *Exit on error* to the *Yes* position to instruct *pg_restore* to exit restore if there is an error in sending SQL commands. The default is to continue and to display a count of errors at the end of the restore.

When you've specified the details that wil be incorporated into the *pg_restore* command, click the *Restore* button to start the process, or click the *Cancel* button to exit without saving your work. A popup wil confirm if the restore is successful.

Use the *Stop Process* button to stop the *Restore* process.

Click [Click here for details](#) on the popup to launch the *Process Watcher*.

The *Process Watcher* logs all the activity associated with the restore, and provides additional information for troubleshooting should the restore command encounter problems.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Schema Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Schema Dialog ¶

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Schema Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)



[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)



[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

Schema Dialog

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

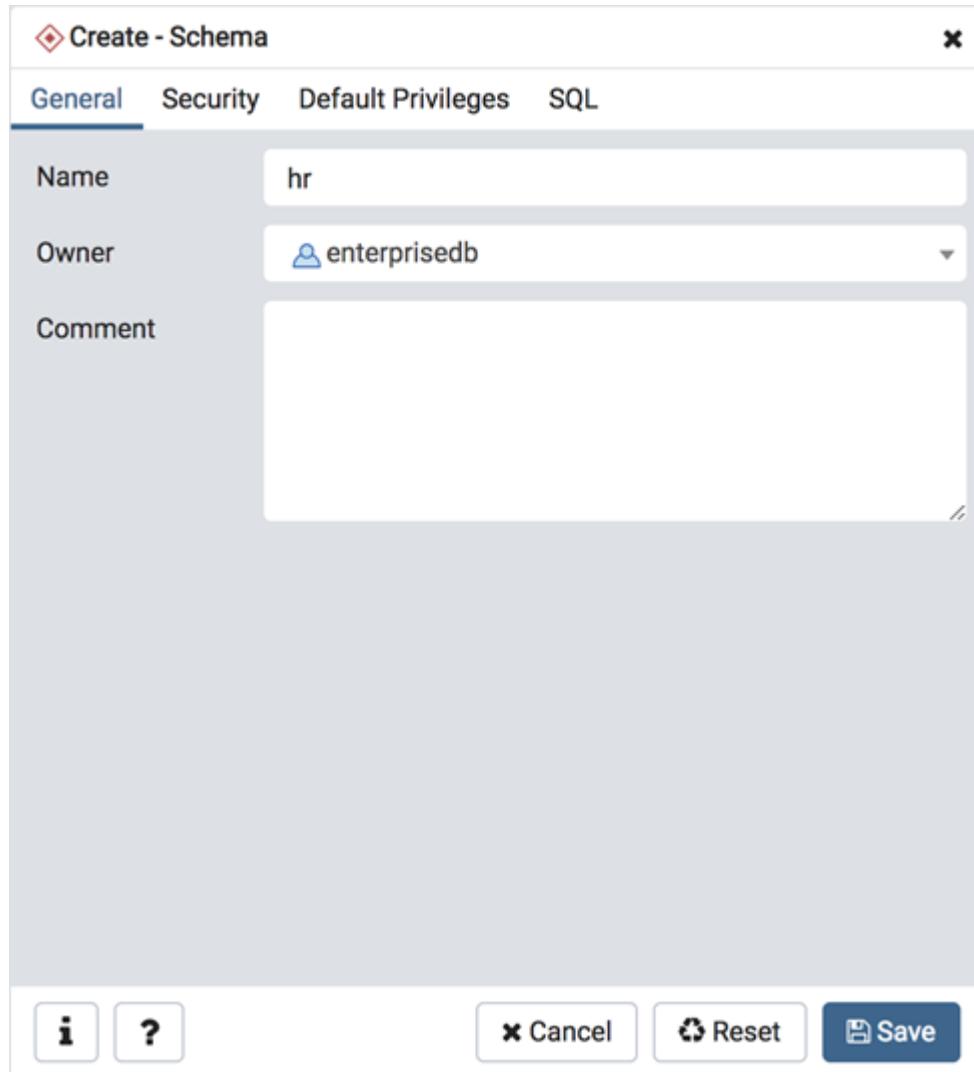
[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

Schema Dialog



The screenshot shows the 'Create - Schema' dialog box. The 'General' tab is selected. The 'Name' field contains 'hr'. The 'Owner' field is set to 'enterprisedb'. The 'Comment' field is empty. At the bottom, there are buttons for 'i', '?', 'Cancel', 'Reset', and 'Save'.

Use the *Schema* dialog to define a schema. A schema is the organizational workhorse of a database, similar to directories or namespaces. To create a schema, you must be a database superuser or have the CREATE privilege.

The *Schema* dialog organizes the development of schema through the following dialog tabs: *General* and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

Use the fields on the *General* tab to identify the schema.

Use the *Name* field to add a descriptive name for the schema. The name will be displayed in the *pgAdmin* tree control.

Select the owner of the schema from the drop-down listbox in the *Owner* field.

Store notes about the schema in the *Comment* field.

Click the *Security* tab to continue.

The screenshot shows the 'Create - Schema' dialog box with the 'Security' tab selected. The dialog has four tabs at the top: General, Security (which is active), Default Privileges, and SQL. The main area is divided into two sections: 'Privileges' and 'Security Labels'. The 'Privileges' section contains a table with columns 'Grantee', 'Privileges', and 'Grantor'. It shows one entry: 'enterprisedb' under 'Grantee', 'C*U*' under 'Privileges', and 'enterprisedb' under 'Grantor'. The 'Security Labels' section contains a table with columns 'Provider' and 'Security Label'. It shows one entry: 'myprovider' under 'Provider' and 'mysecurity' under 'Security Label'. At the bottom of the dialog are buttons for 'Cancel', 'Reset', and 'Save'.

Use the *Security* tab to assign privileges and security labels for the schema.

Click the *Add* icon (+) to assign a set of privileges in the *Privileges* panel: Select the name of the role from the drop-down listbox in the *Grantee* field.

Click inside the *Privileges* field. Check the boxes to the left of one or more privileges to grant the selected privileges to the specified user.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.

Click the *Add* icon (+) to assign additional sets of privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *Add* icon (+) to assign a security label in the *Security Labels* panel:

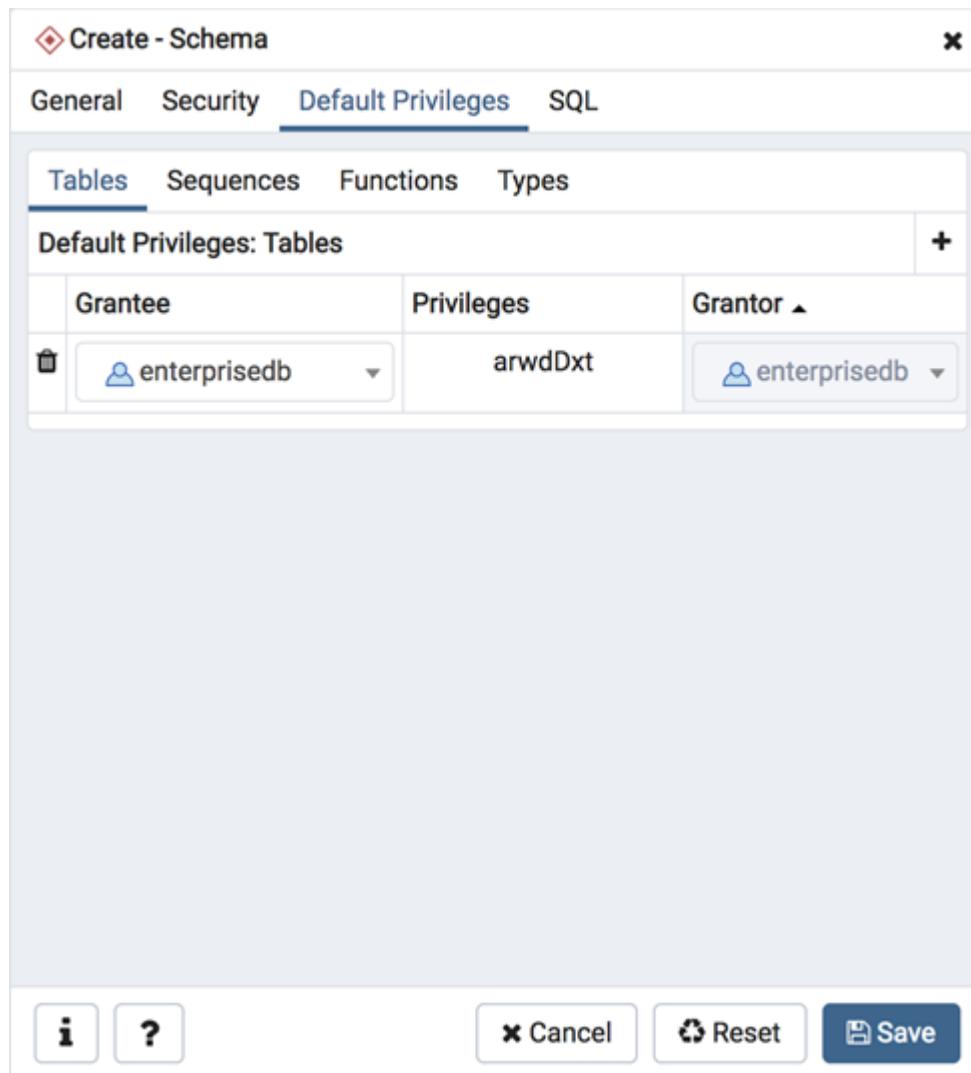
Specify a security label provider in the *Provider* field. The named provider must be loaded and must consent to the proposed labeling operation.

Specify a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL

places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the *Add* icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *Default Privileges* tab to continue.



Use the *Default Privileges* tab to grant privileges for tables, sequences, functions and types. Use the tabs nested inside the *Default Privileges* tab to specify the database object and click the Add icon (+) to assign a set of privileges:

Select the name of a role that will be granted privileges in the schema from the drop-down listbox in the *Grantee* field.

Click inside the *Privileges* field. Check the boxes to the left of one or more privileges to grant the selected privileges to the specified user.

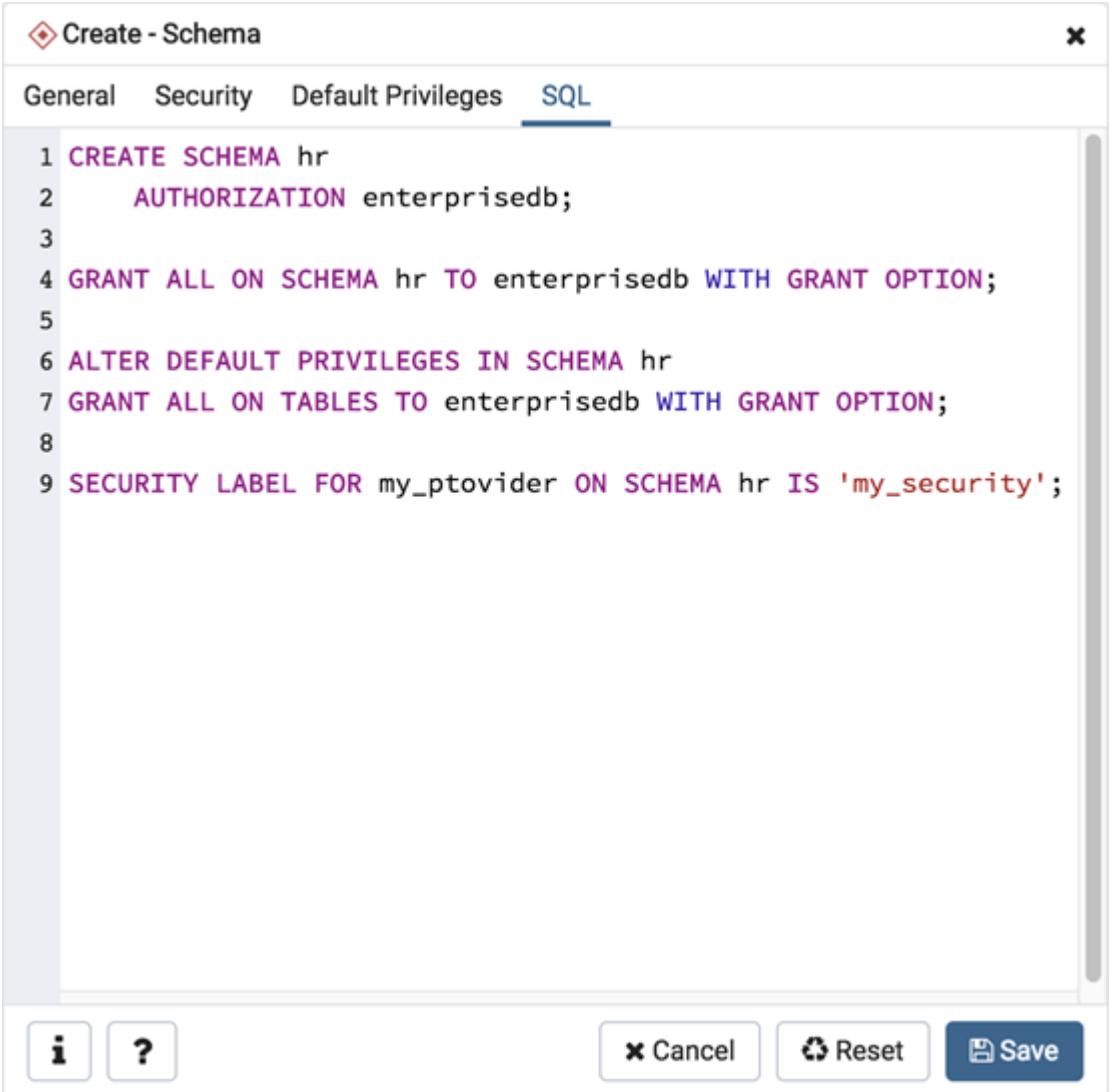
The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.

Click the *SQL* tab to continue.

Your entries in the *Schema* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example 

The following is an example of the sql command generated by selections made in the *Schema* dialog:



The screenshot shows the 'Create - Schema' dialog box. The title bar says 'Create - Schema'. Below it is a tab bar with four tabs: 'General', 'Security', 'Default Privileges', and 'SQL'. The 'SQL' tab is selected and highlighted in blue. The main area contains a multi-line text editor with the following SQL code:

```
1 CREATE SCHEMA hr
2     AUTHORIZATION enterprisedb;
3
4 GRANT ALL ON SCHEMA hr TO enterprisedb WITH GRANT OPTION;
5
6 ALTER DEFAULT PRIVILEGES IN SCHEMA hr
7 GRANT ALL ON TABLES TO enterprisedb WITH GRANT OPTION;
8
9 SECURITY LABEL FOR my_ptovider ON SCHEMA hr IS 'my_security';
```

At the bottom of the dialog are several buttons: an info icon (i), a question mark icon (?), a 'Cancel' button with a cancel symbol, a 'Reset' button with a recycle symbol, and a 'Save' button with a disk symbol.

The example creates a schema named hr; the command grants *USAGE*

privileges to *public* and assigns the ability to grant privileges to *alice*.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | Development

[pgAdmin 4 documentation](#) > [Developer Tools](#) >

[previous](#) [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Schema Diff](#)
 - [The Schema Diff Object Comparison Panel](#)
 - [Schema Diff DDL Comparison Panel](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
 - [Debugger](#)
 - [Query Tool](#)
 - [View/Edit Data](#)
 - [Schema Diff](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Schema Diff

Schema Diff is a feature that allows you to compare schema objects between two database schemas. Use the **Tools** menu to access Schema Diff.

The Schema Diff feature allows you to:

- Compare and synchronize the database schemas (from source to target).
- Visualize the differences between database schemas.
- List the differences in SQL statement for target schema objects.
- Generate synchronization scripts.

Note – The source and target databases must be of the same major version.

Click on **Schema Diff** under the **Tools** menu to open a selection panel. Choose the source and target servers, databases, and schemas that will be compared. After selecting the objects, click on the **Compare** button.

You can open multiple copies of **Schema Diff** in individual tabs simultaneously. To close a copy of **Schema Diff**, click the **X** in the upper-right hand corner of the tab bar.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Developer Tools](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



Schema Diff

[Download](#)

[The Schema Diff Object Comparison Panel](#)

[Schema Diff DDL Comparison Panel](#)



[Getting Started](#)

[Connecting To A Server](#)

[FAQ](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Creating or Modifying a Table](#)

[Management Basics](#)

[Latest Docs](#)

[Backup and Restore](#)

[Developer Tools](#)



[Debugger](#)

[Query Tool](#)

[Ask a](#)

[View/Edit Data](#)

[Question](#)

Schema Diff

[pgAgent](#)



[pgAdmin Project Contributions](#)

[Report a Bug](#)

[Release Notes](#)

[Licence](#)



Schema Diff

Screenshots

Schema Diff is a feature that allows you to compare schema objects between two database schemas. Use the *Tools* menu to access Schema Diff.

The Schema Diff feature allows you to:

Compare and synchronize the database schemas (from source to target).

Visualize the differences between database schemas.

List the differences in SQL statement for target schema objects.

Generate synchronization scripts.

Note - The source and target databases must be of the same major version.

Click on *Schema Diff* under the *Tools* menu to open a selection panel.

Choose the source and target servers, databases, and schemas that will be compared. After selecting the objects, click on the *Compare* button.

You can open multiple copies of *Schema Diff* in individual tabs simultaneously. To close a copy of Schema Diff, click the *X* in the upper-right hand corner of the tab bar.

The screenshot shows two panels of the Schema Diff tool. The top panel, titled 'Schema Diff (Basic)', displays a tree view of database objects for comparison. It includes sections for 'Pfunctions', 'Materialized Views', and 'Tables'. The bottom panel, titled 'DDL Comparison', shows the SQL DDL scripts for the 'source' and 'target' databases, highlighting differences between them.

Schema Objects

- Pfunctions - Identical: 1000 Differences: 0 Source Only: 0 Target Only: 0
- Materialized Views - Identical: 1001 Differences: 0 Source Only: 0 Target Only: 0
- Tables - Identical: 0 Differences: 1000 Source Only: 8 Target Only: 1

Object	Status
tbl1	Source Only
tbl2	Target Only
tbl3	Different
tbl4	Different
tbl5	Different
tbl6	Different
tbl7	Different
tbl8	Different
tbl9	Different
tbl10	Different

DDL Comparison

Source	Target	Difference
1 CREATE TABLE source_sc.t215	1 CREATE TABLE target_sc.t215	
2 (2 (
3 id bigint NOT NULL,	3 id bigint NOT NULL,	
4 name text COLLATE pg_catalog."default",	4 name text COLLATE pg_catalog."default",	
5 CONSTRAINT t215_pk PRIMARY KEY (id),	5 CONSTRAINT t215_pk PRIMARY KEY (id),	
6 CONSTRAINT t215_ns_check CHECK (id > 100) NO	6 CONSTRAINT t215_ns_check CHECK (id > 200) NO	
7)	7)	
8 NNTL (8 NNTL (
9 OIDS = FALSE,	9 OIDS = FALSE,	
10 autovacuum_analyze = TRUE,	10 autovacuum_analyze = TRUE,	
11 autovacuum_analyze_threshold = 20,	11 autovacuum_analyze_threshold = 40,	
12 autovacuum_vacuum_cost_delay = 10	12 autovacuum_vacuum_cost_delay = 30	
13)	13)	
14 TABLESPACE pg_default;	14 TABLESPACE pg_default;	
15	15	
16 ALTER TABLE source_sc.t215	16 ALTER TABLE target_sc.t215	
17	17	
18	18	

Select the server, database and schema for the source and target and click Compare to compare them.

Use the [Preferences](#) dialog to specify if *Schema Diff* should open in a new browser tab. Set *Open in new browser tab* option to true.

The *Schema Diff* panel is divided into two panels; an Object Comparison panel and a DDL Comparison panel.

The Schema Diff Object Comparison Panel

In the object comparison panel, you can select the source and target servers of the same major version, databases, and schemas to be compared. You can select any server listed under the browser tree whether it is connected or disconnected. If you select a server that is not connected then it will prompt you for the password before using the server.

Next, select the databases that will be compared. The databases can be the same or different (and within the same server or from different servers).

Lastly, select the source and target schemas which will be compared.

After you select servers, databases, and schemas, click on the *Compare* button to obtain the *Comparison Result*.

The screenshot shows the Schema Diff interface comparing two PostgreSQL databases, both named "PostgreSQL 10". The "Select Source" dropdown is set to "PostgreSQL 10" and "Select Target" is also set to "PostgreSQL 10". The "Source" schema is "postgres" and the "Target" schema is "target_01". The "Compare" button is highlighted in blue. The "Comparison Result" pane displays the differences found:

Category	Difference Type	Count
Functions	Identical	1000
Materialized Views	Identical	1000
Tables	Identical	1000
Trigger Functions	Identical	1000
Views	Different	1000

Under the "Views" category, there are eight specific views listed, all marked as "Identical". Below this, the "DDL Comparison" section shows a table with three columns: "Source", "Target", and "Difference". All three columns show the value "1".

At the bottom right of the comparison result pane, there is a filter dropdown menu with the following options checked:

- ✓ Materialized
- ✓ Differed
- ✓ Source Only
- ✓ Target Only

Use the drop-down lists of Functions, Materialized Views, Tables, Trigger Functions, Procedures, and Views to view the DDL

statements of all the schema objects.

In the upper-right hand corner of the object comparison panel is a *Filter* option that you can use to filter the schema objects based on the following comparison criteria:

Identical – If the object is found in both schemas with the same SQL statement, then the comparison result is identical.

Different – If the object is found in both schemas but have different SQL statements, then the comparison result is different.

Source Only – If the object is found in source schema only and not in target schema, then the comparison result is source only.

Target Only – If the object is found in target schema only and not in source schema, then the comparison result is target only.

Click on any of the schema objects in the object comparison panel to display the DDL Statements of that object in the DDL Comparison panel.

Schema Diff DDL Comparison Panel

The *DDL Comparison* panel displays three columns: The first column displays the DDL statement of the object from the source schema.

The second column displays the DDL statement of the object from the target schema.

The third column displays the difference in the SQL statement of the target schema object.

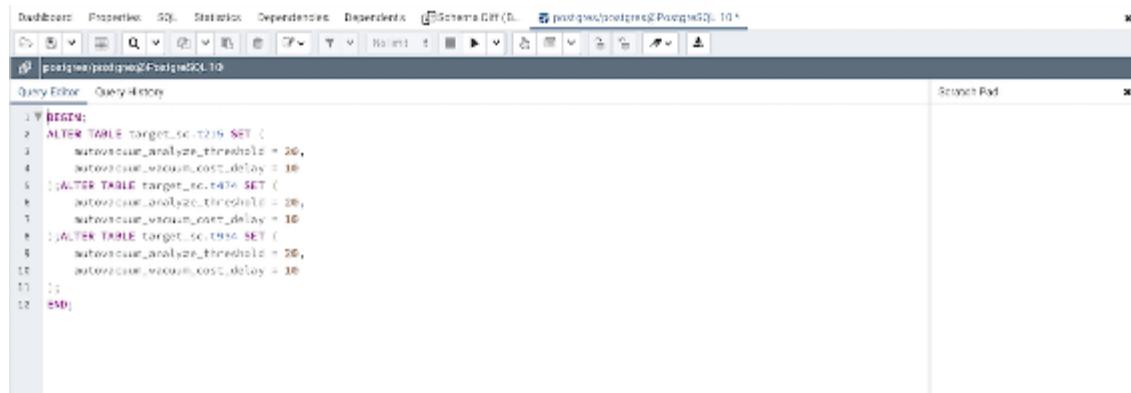
You can review the DDL statements of all the schema objects to check for the differences in the SQL statements.

Also, you can generate the SQL script of the differences found in the target schema object based on the SQL statement of the source schema object. To generate the script, select the checkboxes of the schema objects in the object comparison panel and then click on the **Generate Script** button in the upper-right hand corner of the object comparison panel.

Select the schema objects and click on the **Generate Script** button to open the *Query Tool* in a new tab, with the difference in the SQL statement displayed in the *Query Editor*.

If you have clicked on the schema object to check the difference generated in the *DDL Comparison Panel*, and you have not selected the checkbox of the schema object, PEM wil open the *Query Tool* in a new tab, with the differences in the SQL statements displayed in the *Query Editor*.

You can also use the **Copy** button to copy the difference generated in the *DDL Comparison* panel.



The screenshot shows the Oracle Database Query Editor window. The title bar reads "Schema Diff (D... prod|prod|analyze|PurgeSQL-10)". The main pane displays the following SQL code:

```
1  BEGIN;
2  2    ALTER TABLE target_sc_t2n SET (
3  3      autovacuum_analyze_threshold = 20,
4  4      autovacuum_vacuum_cost_delay = 10
5  5    );
6  6    ALTER TABLE target_sc_t4t SET (
7  7      autovacuum_analyze_threshold = 20,
8  8      autovacuum_vacuum_cost_delay = 10
9  9    );
10 10   ALTER TABLE target_sc_t3n SET (
11 11     autovacuum_analyze_threshold = 20,
12 12     autovacuum_vacuum_cost_delay = 10
13 13   );
14 14 END;
```

Apply the SQL Statement in the target schema to synchronize the schemas.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)

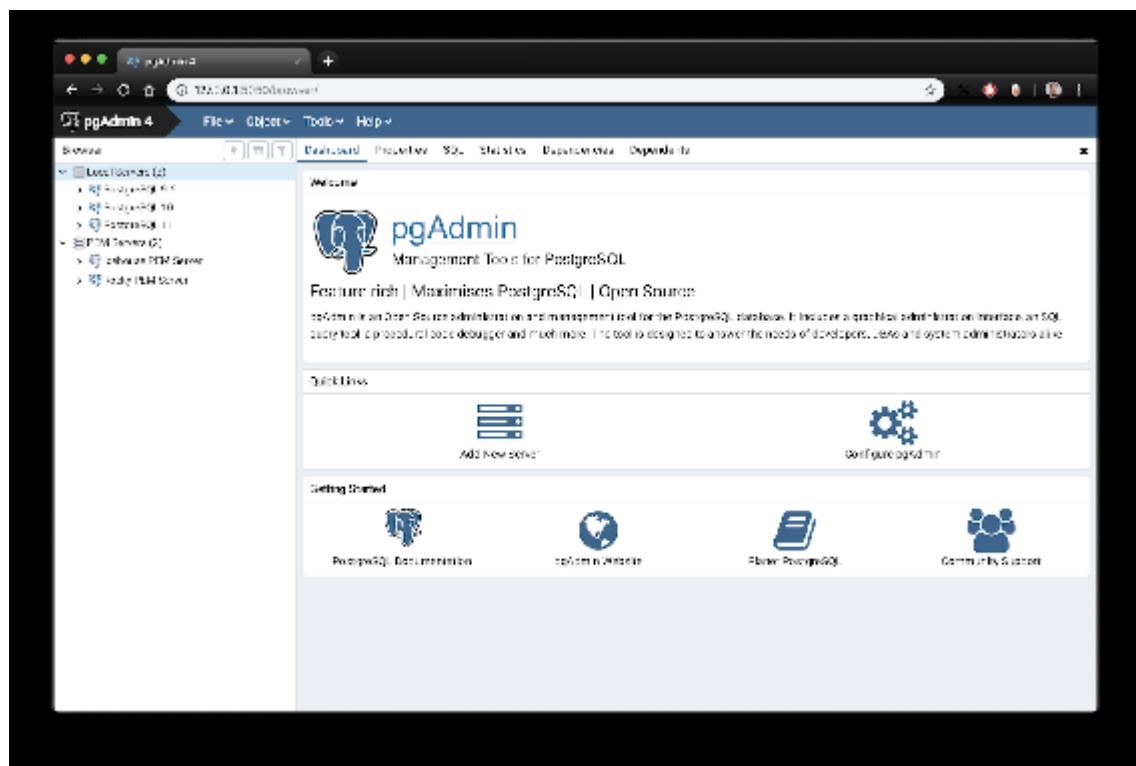
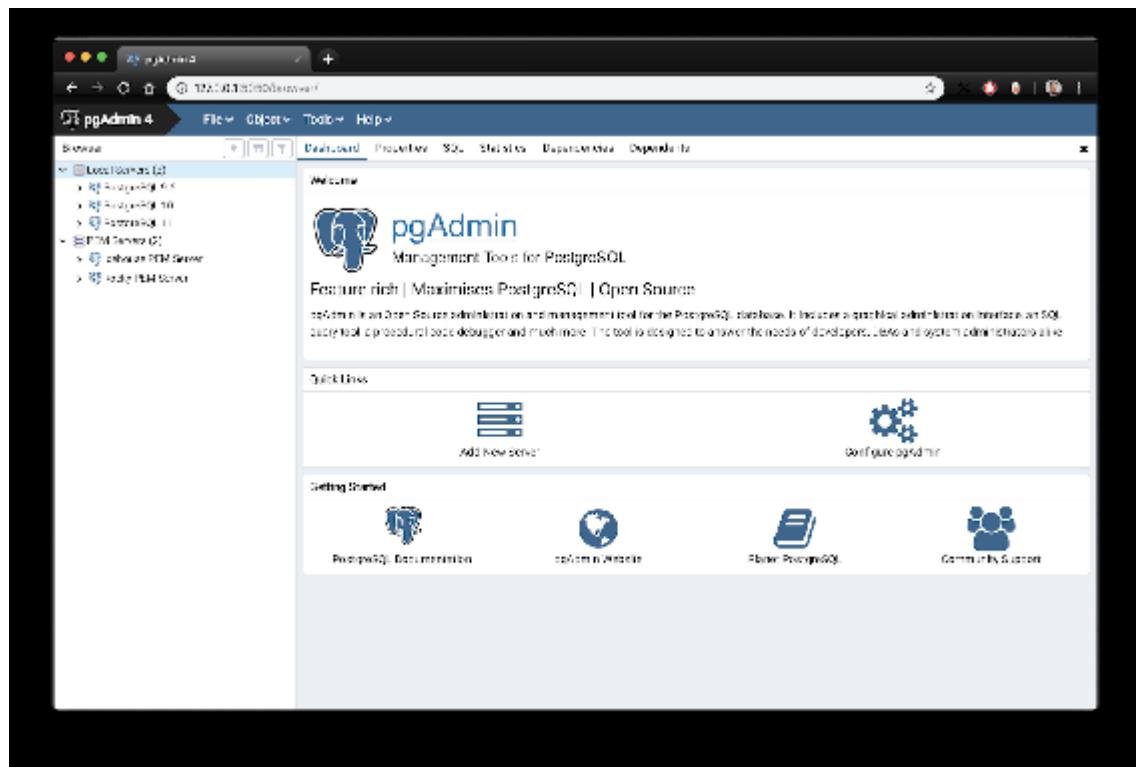


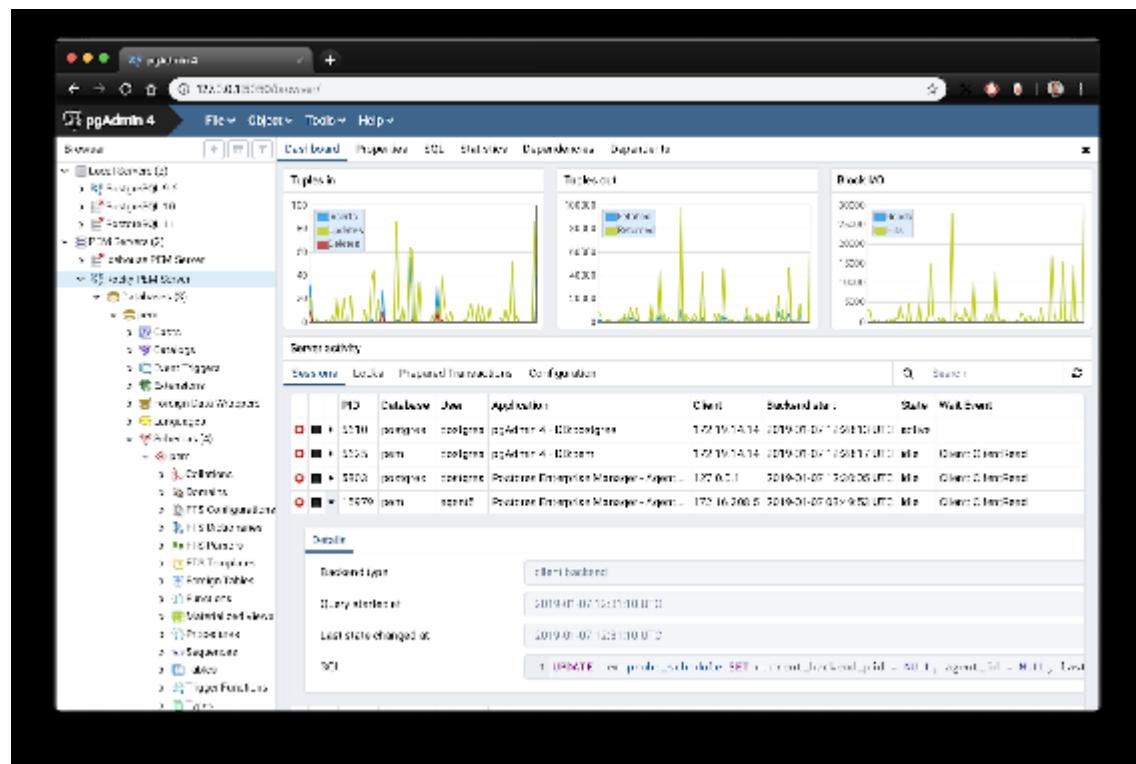
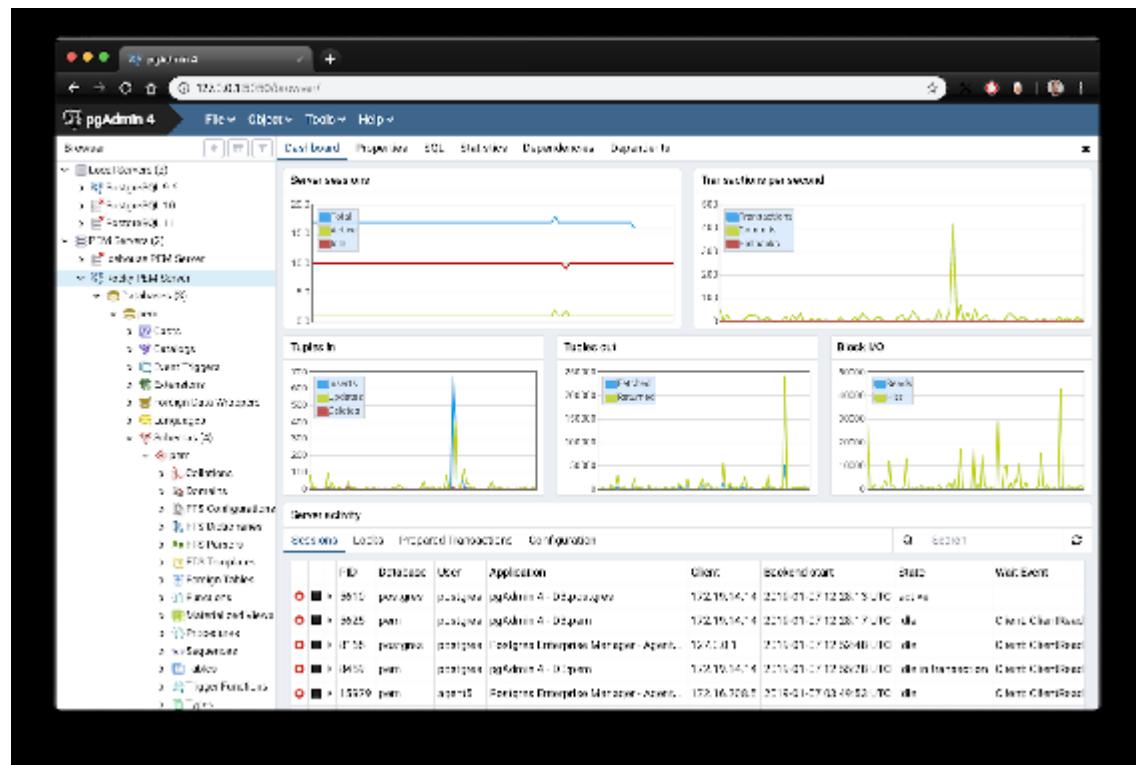
[Screenshots](#)

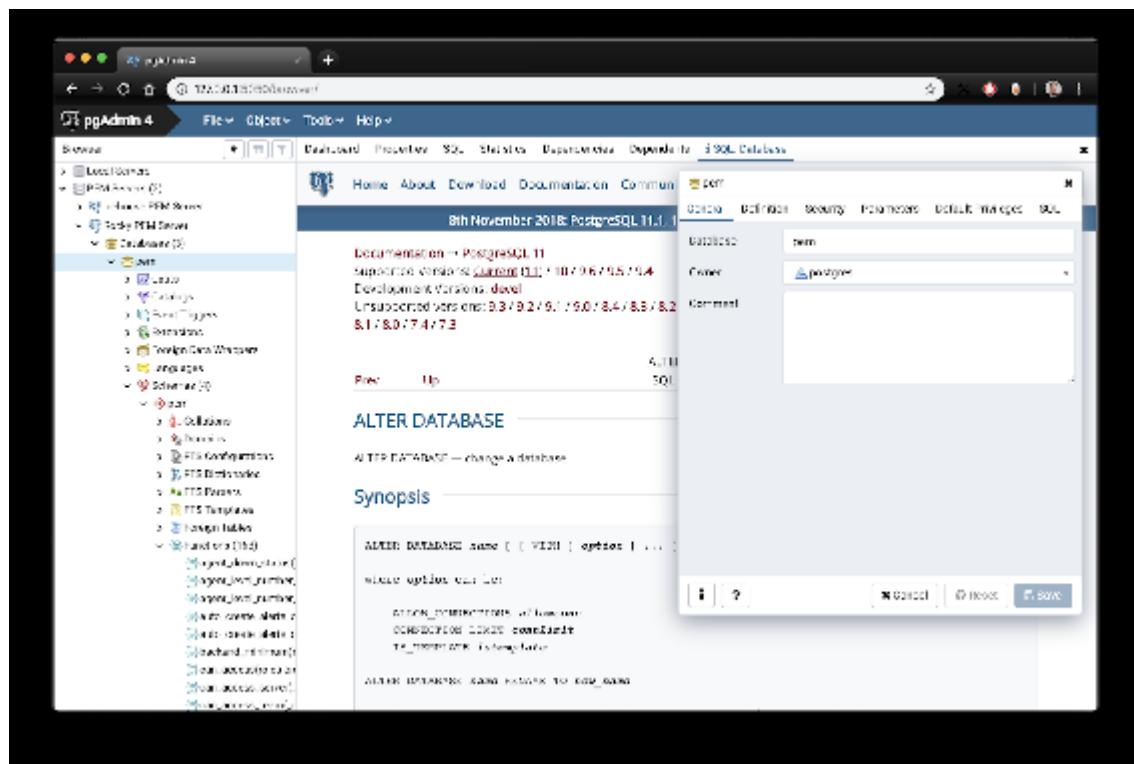
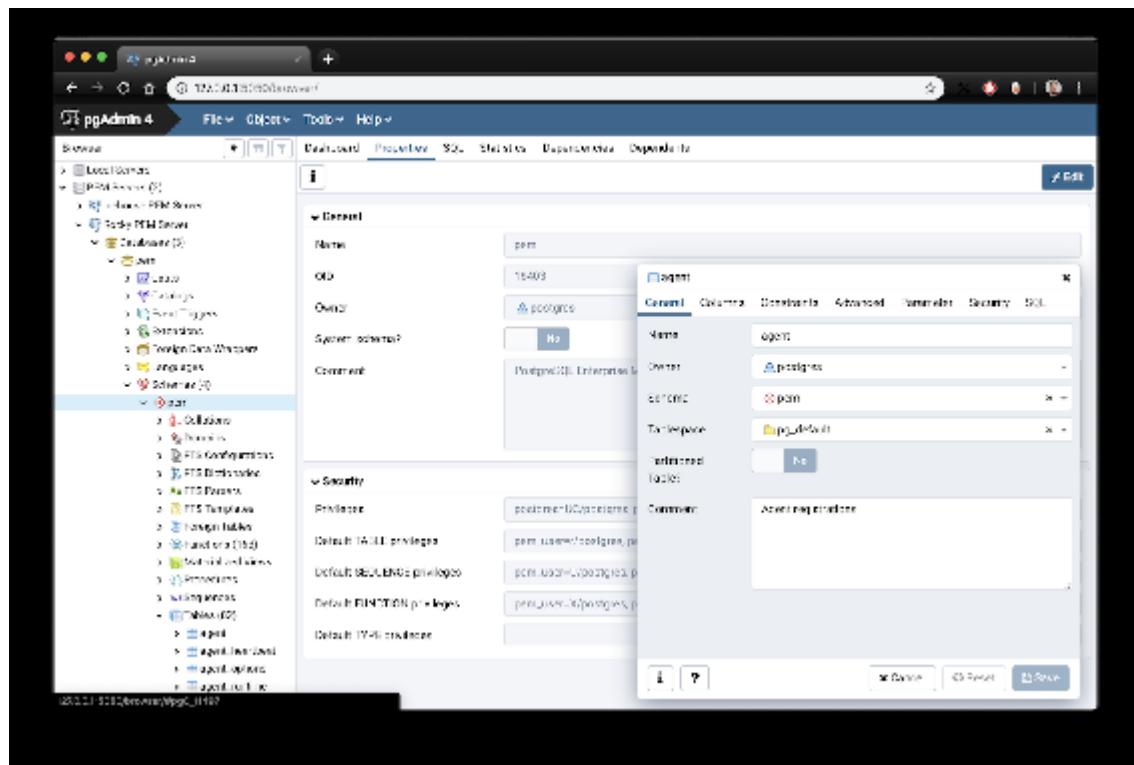
Screenshots

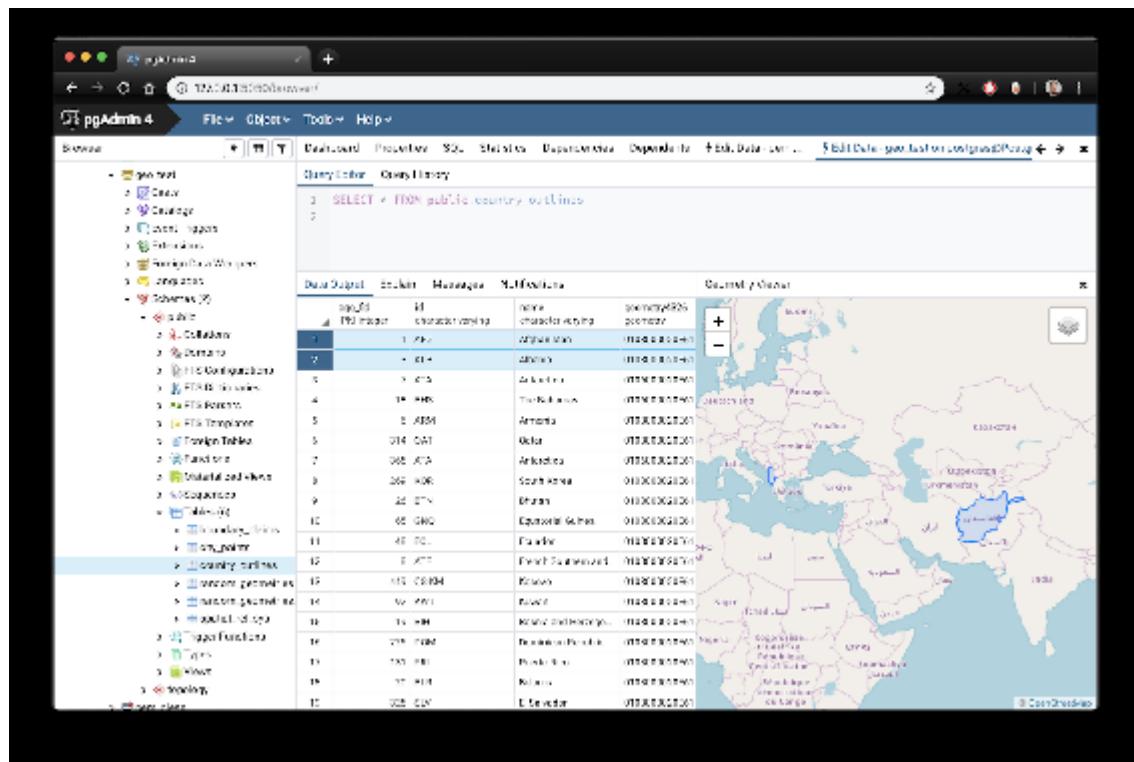
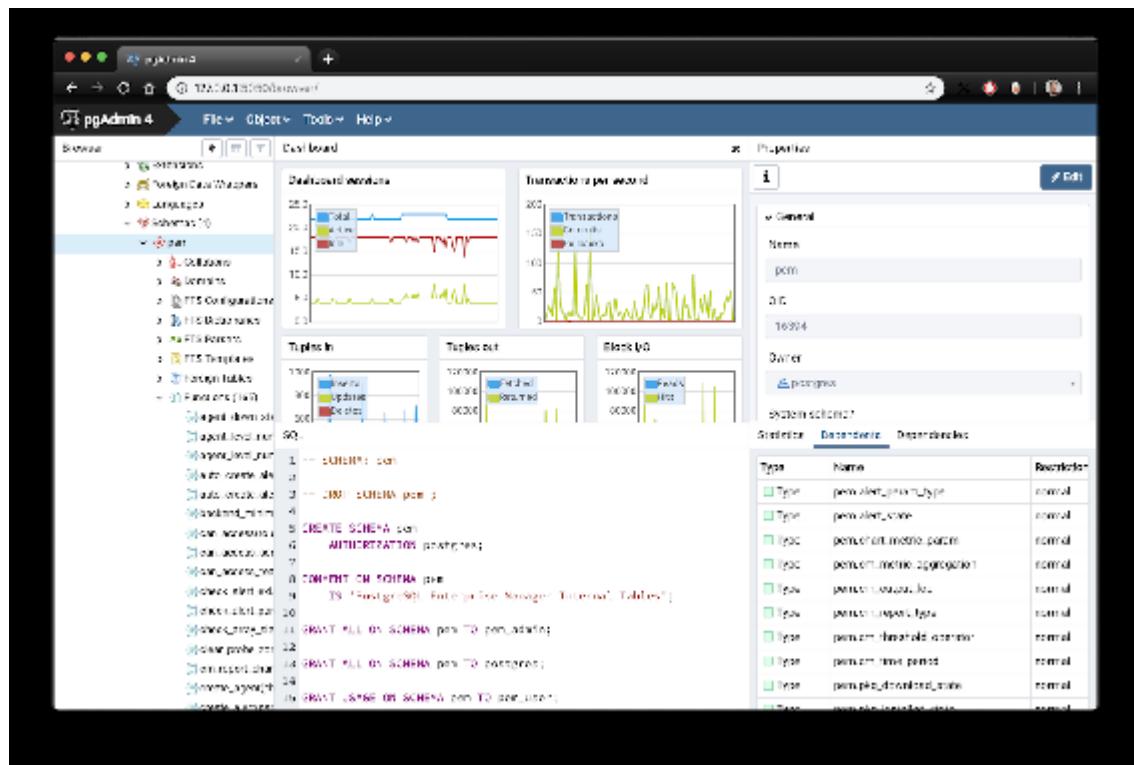


Welcome screen









pgAdmin 4

File → Object → Tools → Help

Browser

- Local Databases
- PGDialects (1)
- pgAdmin PG Dialect
- PGDialect PG Dialects
- pgAdmin PG Dialect
- Database (1)
- dev
- Local
- Functions
- Triggers
- Rule Triggers
- Constraints
- Foreign Data Wrappers
- Languages
- Schemas (2)
 - pgAdmin
 - pgAdmin (2)
 - Agent
 - agent_commands
 - agent_options
 - agent_out
 - agent_outgoing

Dashboard → Properties → SQL → Statistics → Dependencies → Edges → View → pgAdmin (2) PG Dialect

Query Editor → Open Heavy

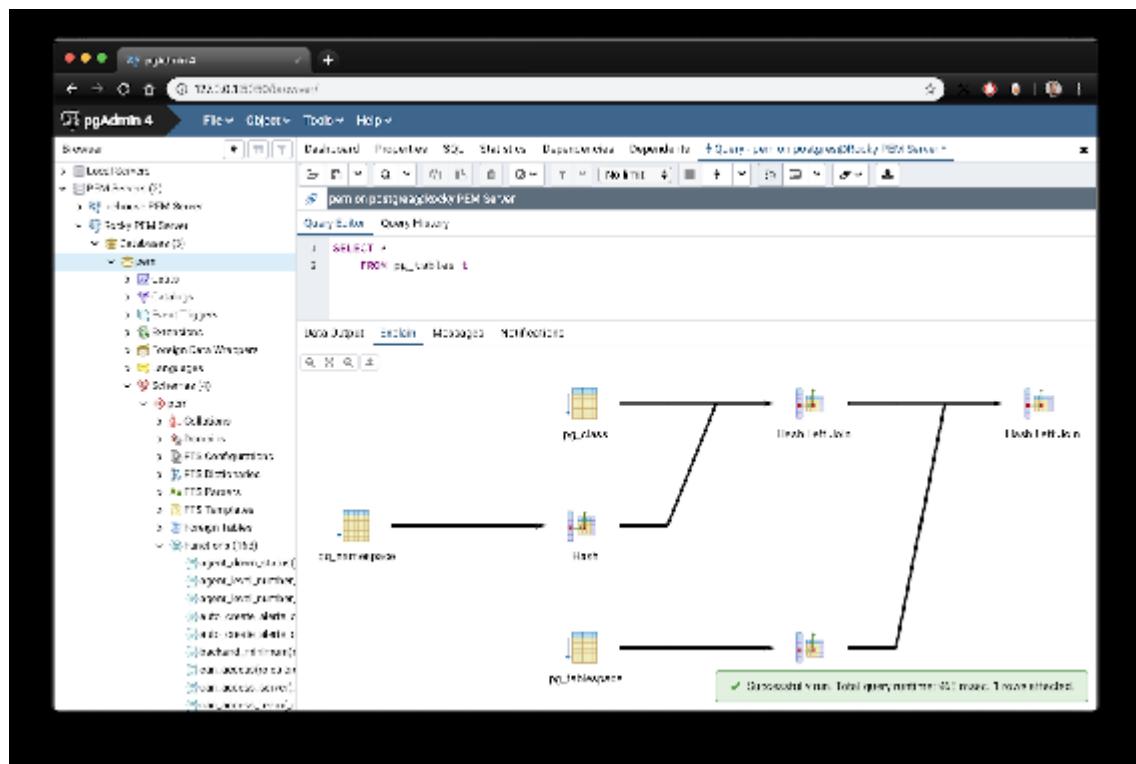
Editor → Dev → 2019-07-23 18:50:24

SELECT * FROM pgAdmin_heavy

Agent ID	Run Number	Run Description
1	2019-07-23 18:50:24.74507463	
2	2019-07-23 18:50:24.84102963	
3	2019-07-23 18:50:24.84102963	
4	2019-07-23 18:50:24.84102963	
5	2019-07-23 18:50:24.84102963	
6	2019-07-23 18:50:24.84102963	

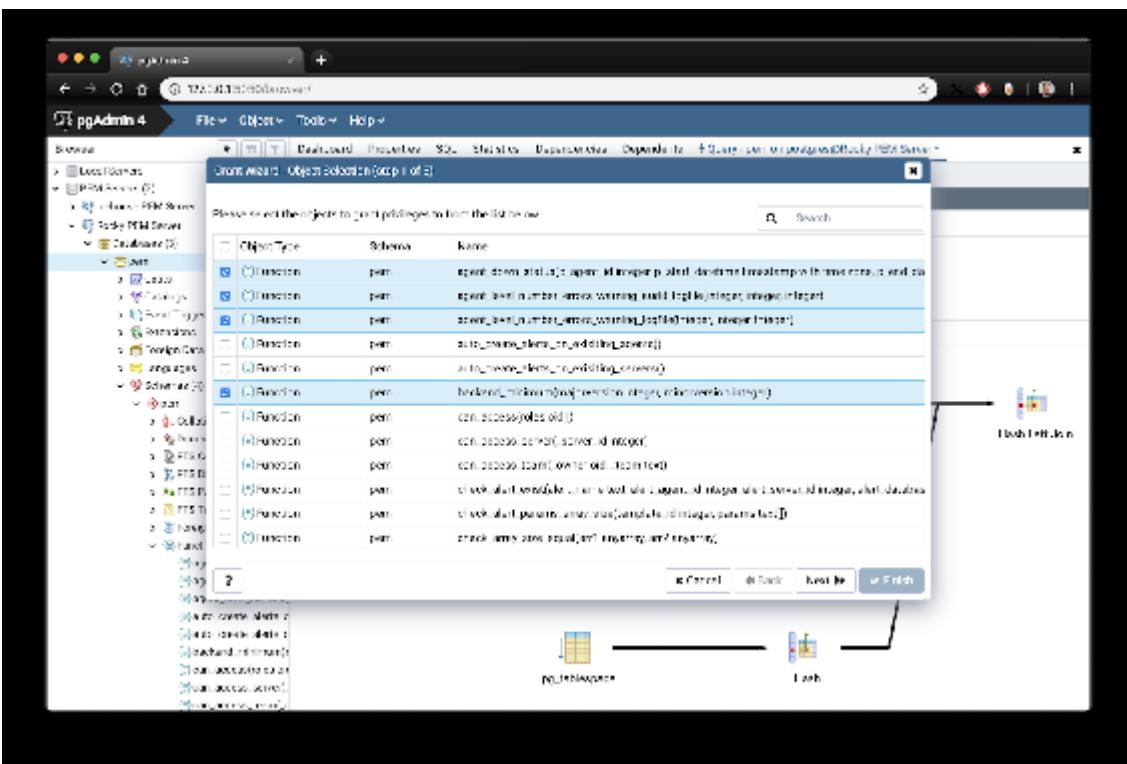
Info

Successfully ran total query number 6 rows affected.



```
backward_maintenance(integer) integer backward_maintenance_id integer
BEGIN
    version_warning := version_alm := major := minor := bugfix := 0;
    CASE
        WHEN backward_maintenance <= 0 THEN
            version_warning := version_alm := major := minor := bugfix := 1;
        WHEN backward_maintenance > 0 AND backward_maintenance < 100 THEN
            version_alm := major := minor := bugfix := 1;
        ELSEIF (backward_maintenance >= 100) THEN
            -- No action
    END CASE;
    RETURN;
END;
```

Name	Type	Value
backward_maintenance	integer	10
backward_maintenance_id	integer	5



Quick Links

Screenshots



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a](#)

[Question](#)

Welcome screen



[Report a Bug](#)



[Screenshots](#)

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Sequence Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Sequence Dialog ¶

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Sequence Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Cast Dialog](#)



[Col ation Dialog](#)

[Domain Dialog](#)

[Latest Docs](#)

[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)



[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)

[Ask a](#)

[Foreign Server Dialog](#)

[Question](#)

[Foreign Table Dialog](#)

[FTS Configuration Dialog](#)



[FTS Dictionary Dialog](#)

[Report a Bug](#)

[FTS Parser Dialog](#)

[FTS Template Dialog](#)

[Function Dialog](#)

[□](#)

[Language Dialog](#)

[Screenshots](#)

[Materialized View Dialog](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Schema Dialog](#)

[Sequence Dialog](#)

[Synonym Dialog](#)

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

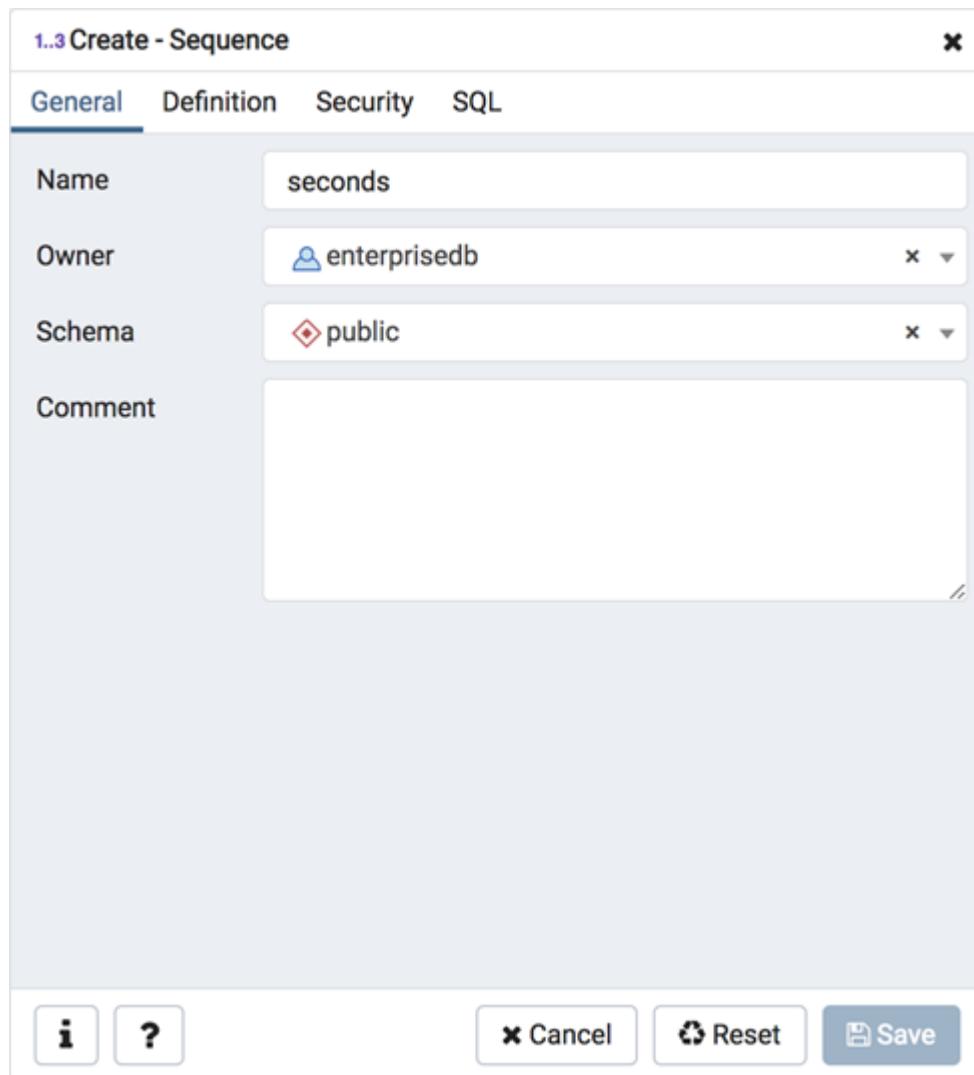
[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

Licence

Sequence Dialog



Use the *Sequence* dialog to create a sequence. A sequence generates unique values in a sequential order (not necessarily contiguous).

The *Sequence* dialog organizes the development of a sequence through the following dialog tabs: *General*, *Definition*, and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

Use the fields in the *General* tab to identify a sequence: Use the *Name* field to add a descriptive name for the sequence. The name will be displayed in the *pgAdmin* tree control. The sequence name must be distinct from the name of any other sequence, table, index, view, or foreign table in the same schema.

Use the drop-down listbox next to *Owner* to select the name of the role that will own the sequence.

Use the drop-down listbox next to *Schema* to select the schema in which the sequence will reside.

Store notes about the sequence in the *Comment* field.

Click the *Definition* tab to continue.

1.3 Create - Sequence

General Definition Security SQL

Increment

Start

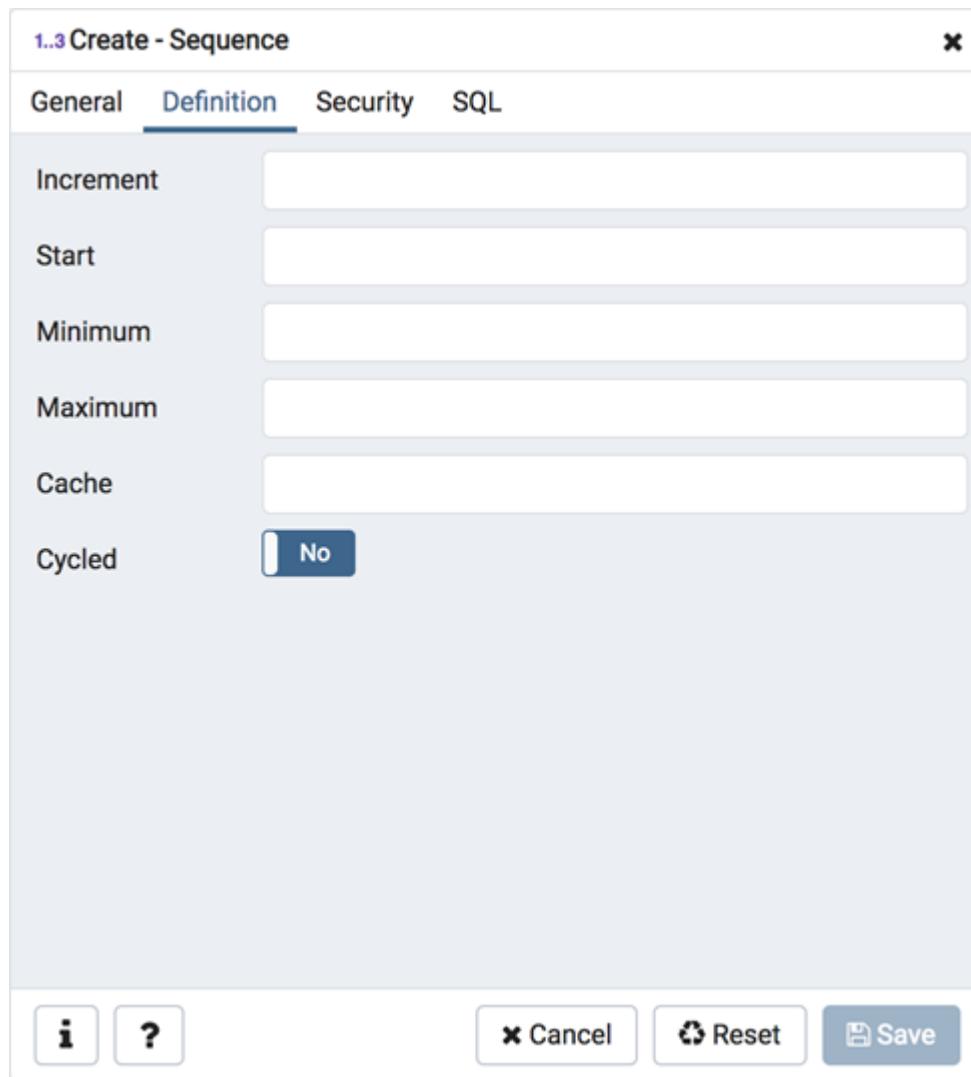
Minimum

Maximum

Cache

Cycled No

i **?** **x Cancel** **Reset** **Save**



Use the fields in the *Definition* tab to define the sequence: Use the *Increment* field to specify which value is added to the current sequence value to create a new value.

Provide a value in the *Start* field to specify the beginning value of the sequence. The default starting value is MINVALUE for ascending sequences and MAXVALUE for descending ones.

Provide a value in the *Minimum* field to specify the minimum value a sequence can generate. If this clause is not supplied or NO

MINVALUE is specified, then defaults will be used. The defaults are 1

and -263-1 for ascending and descending sequences, respectively.

Provide a value in the *Maximum* field to specify the maximum value for the sequence. If this clause is not supplied or NO MAXVALUE is specified, then default values will be used. The defaults are 263-1

and -1 for ascending and descending sequences, respectively.

Provide a value in the *Cache* field to specify how many sequence numbers are to be preallocated and stored in memory for faster access. The minimum value is 1 (only one value can be generated at a time, i.e., no cache), and this is also the default.

Move the *Cycled* switch to the *Yes* position to allow the sequence to wrap around when the MAXVALUE or the MINVALUE has been

reached by an ascending or descending sequence respectively. If the limit is reached, the next number generated will be the MINVALUE or MAXVALUE, respectively. The default is *No*.

Click the *Security* tab to continue.

1.3 Create - Sequence

General Definition Security SQL

Privileges

Grantee	Privileges		Grantor
<input type="button" value="enterprisedb"/>	<input type="checkbox"/> ALL <input checked="" type="checkbox"/> SELECT <input checked="" type="checkbox"/> UPDATE <input checked="" type="checkbox"/> USAGE	<input type="checkbox"/> WITH GRANT OPTION <input checked="" type="checkbox"/> WITH GRANT OPTION <input checked="" type="checkbox"/> WITH GRANT OPTION <input checked="" type="checkbox"/> WITH GRANT OPTION	<input type="button" value="enterprisedb"/>

Security Labels

Provider	Security Label
<input type="button" value="data_provider"/>	data_security

Use the **Security** tab to assign privileges and define security labels for the sequence.

Use the **Privileges** panel to assign privileges. Click the **Add** icon (+) to set privileges:

Select the name of a role that will be granted privileges from the drop-down listbox in the **Grantee** field.

Click inside the **Privileges** field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.

Click the *Add* icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

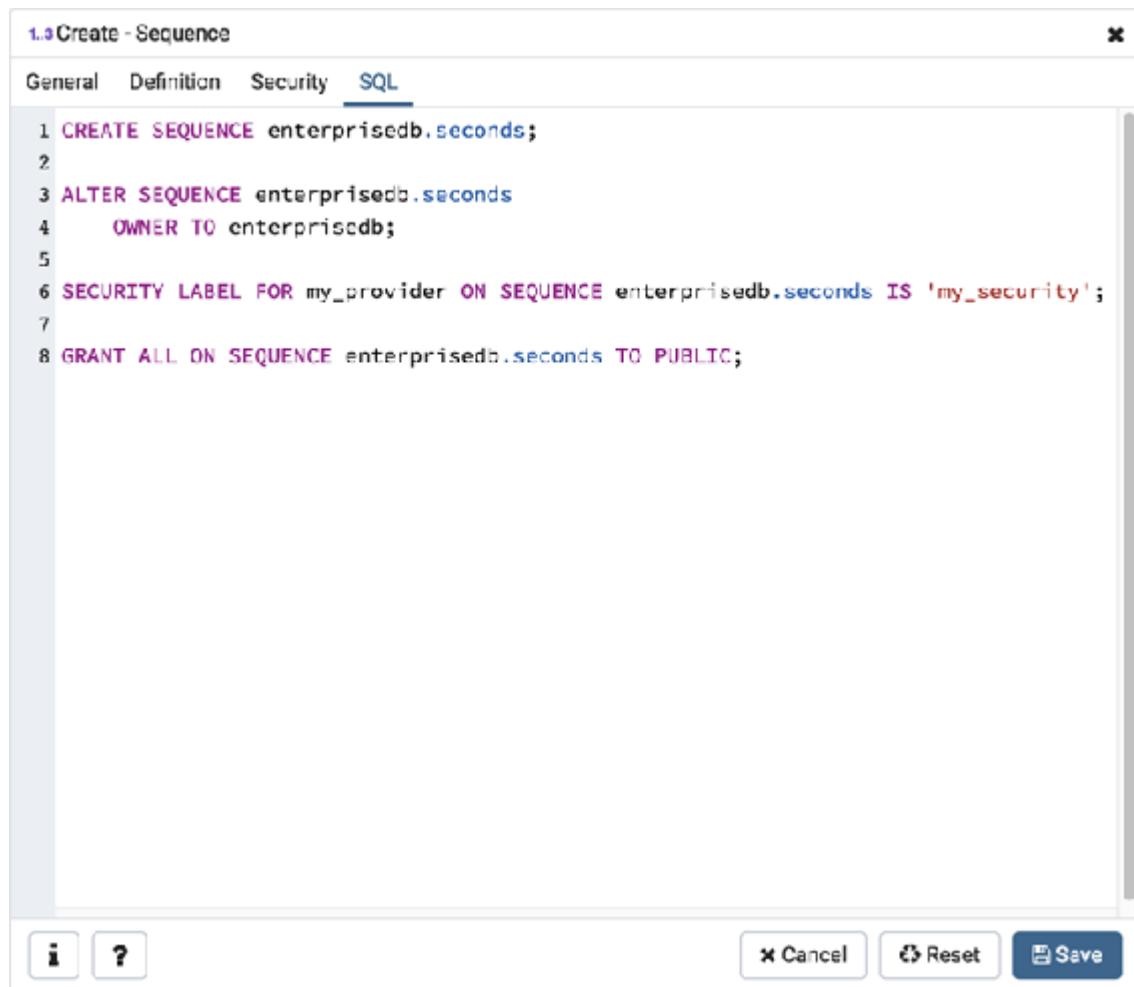
Use the *Security Labels* panel to define security labels applied to the sequence. Click the *Add* icon (+) to add each security label selection: Specify a security label provider in the *Provider* field. The named provider must be loaded and must consent to the proposed labeling operation.

Specify a a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL

places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the *Add* icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.



Your entries in the **Sequence** dialog generate a generate a SQL command (see an example below). Use the **SQL** tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the **Sequence** dialog:

The example shown demonstrates a sequence named *seconds*. The sequence will increase in 5 second increments, and stop when it reaches a maximum value equal of 60.

Click the *Info* button (i) to access online help.

Click the **Save** button to save work.

Click the **Cancel** button to exit without saving work.

Click the **Reset** button to restore configuration parameters.

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 | Development

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) » [Deployment](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Server Deployment](#)
 - [Requirements](#)
 - [Configuration](#)
 - [Python](#)
 - [Hosting](#)
 - [Apache HTTPD Configuration \(Windows\)](#)
 - [Apache HTTPD Configuration \(Linux/Unix\)](#)
 - [Standalone Gunicorn Configuration](#)
 - [Standalone uWSGI Configuration](#)
 - [NGINX Configuration with Gunicorn](#)
 - [NGINX Configuration with uWSGI](#)
- [Getting Started](#)
 - [Deployment](#)
 - [Login Dialog](#)
 - [User Management Dialog](#)
 - [Change User Password Dialog](#)
 - [User Interface](#)
 - [Menu Bar](#)
 - [Toolbar](#)
 - [Tabbed Browser](#)
 - [Tree Control](#)
 - [Preferences Dialog](#)
 - [Keyboard Shortcuts](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Server Deployment

pgAdmin may be deployed as a web application by configuring the app to run in server mode and then deploying it either behind a webserver running as a reverse proxy, or using the WSGI interface.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) » [Deployment](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



Server Deployment

[Download](#)

[Requirements](#)

[Configuration](#)



[Python](#)

[Hosting](#)

[FAQ](#)

[Apache HTTPD Configuration \(Windows\)](#)

[Apache HTTPD Configuration \(Linux/Unix\)](#)



[Standalone Gunicorn Configuration](#)

[Latest Docs](#)

[Standalone uWSGI Configuration](#)

[NGINX Configuration with Gunicorn](#)

[NGINX Configuration with uWSGI](#)



[Getting Started](#)

[Ask a](#)

[Deployment](#)

[Question](#)

[Login Dialog](#)

[User Management Dialog](#)



[Change User Password Dialog](#)

[User Interface](#)

[Report a Bug](#)

[Menu Bar](#)

[Toolbar](#)



[Tabbed Browser](#)

[Tree Control](#)

[Screenshots](#)

[Preferences Dialog](#)

[Keyboard Shortcuts](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

[Server Deployment](#) 

pgAdmin may be deployed as a web application by configuring the app to run in server mode and then deploying it either behind a webserver running as a reverse proxy, or using the WSGI interface.

The following instructions demonstrate how pgAdmin may be run as a WSGI application under Apache HTTPD, using mod_wsgi, standalone using uWSGI or Gunicorn, or under NGINX using uWSGI or Gunicorn.

See also

For detailed instructions on building and configuring pgAdmin from scratch, please see the README file in the top level directory of the source code. For convenience, you can find the latest version of the file

[here](#), but be aware that this may differ from the version included with the source code for a specific version of pgAdmin.

Requirements ¶

Important: Some components of pgAdmin require the ability to maintain affinity between client sessions and a specific database connection (for example, the Query Tool in which the user might run a BEGIN command followed by a number of DML SQL statements, and then a COMMIT).

pgAdmin has been designed with built-in connection management to handle this, however it requires that only a single Python process is used because it is not easily possible to maintain affinity between a client session and one of multiple WSGI worker processes.

On Windows systems, the Apache HTTP server uses a single process, multi-threaded architecture. WSGI applications run in embedded mode, which means that only a single process will be present on this platform in all cases.

On Unix systems, the Apache HTTP server typically uses a multi-process, single threaded architecture (this is dependent on the MPM that is chosen at compile time). If embedded mode is chosen for the WSGI application, then there will be one Python environment for each Apache process, each with its own connection manager which

wil lead to loss of connection affinity. Therefore one should use mod_wsgi's daemon mode, configured to use a single process. This wil launch a single instance of the WSGI application which is utilised by al the Apache worker processes.

Whilst it is true that this is a potential performance bottleneck, in reality pgAdmin is not a web application that's ever likely to see heavy traffic unlike a busy website, so in practice should not be an issue.

Future versions of pgAdmin may introduce a shared connection manager process to overcome this limitation, however that is a significant amount of work for little practical gain.

Configuration

In order to configure pgAdmin to run in server mode, it may be necessary to configure the Python code to run in multi-user mode, and then to configure the web server to find and execute the code.

See [The config.py File](#) for more information on configuration settings.

Python

From pgAdmin 4 v2 onwards, server mode is the default configuration. If running under the desktop runtime, this is overridden automatical y. There should typical y be no need to modify the configuration simply to enable server mode to work, however it may be desirable to adjust some of the paths used.

In order to configure the Python code, fol ow these steps:

- . Create a config_local.py file alongside the existing config.py file.
- . Edit config_local.py and add the fol owing settings. In most cases, the default file locations should be appropriate:

NOTE: You must ensure the directories specified are writeable by the user that the web server processes will be running as, e.g.

apache or www-data.

`LOG_FILE = '/var/log/pgadmin4/pgadmin4.log'`

`SQLITE_PATH = '/var/lib/pgadmin4/pgadmin4.db'`

`SESSION_DB_PATH = '/var/lib/pgadmin4/sessions'`

`STORAGE_DIR = '/var/lib/pgadmin4/storage'`

. Run the following command to create the configuration database:

`# python setup.py`

. Change the ownership of the configuration database to the user that the web server processes will run as, for example, assuming that the web server runs as user www-data in group www-data, and that the SQLite path is /var/lib/pgadmin4/pgadmin4.db:

`# chown www-data:www-data`

`/var/lib/pgadmin4/pgadmin4.db`

Hosting 

There are many possible ways to host pgAdmin in server mode. Some examples are given below:

Apache HTTPD Configuration (Windows) 

Once Apache HTTP has been configured to support mod_wsgi, the pgAdmin application may be configured similarly to the example below:

`<VirtualHost *>`

```
ServerName pgadmin.example.com  
WSGIScriptAlias / "C:\Program  
Files\pgAdmin4\web\pgAdmin4.wsgi"  
<Directory "C:\Program Files\pgAdmin4\web" > Order deny,allow  
Allow from all  
</Directory>  
</VirtualHost>
```

Now open the file C:\Program Files\pgAdmin4\web\pgAdmin4.wsgi with your favorite editor and add the code below which wil activate Python virtual environment when Apache server runs.

```
activate_this = 'C:\Program  
Files\pgAdmin4\venv\Scripts\activate_this.py'  
exec(open(activate_this).read())
```

Note: The changes made in pgAdmin4.wsgi file wil revert when pgAdmin4 is either upgraded or downgraded.

Apache HTTPD Configuration (Linux/Unix)

Once Apache HTTP has been configured to support mod_wsgi, the pgAdmin application may be configured similarly to the example below:

```
<VirtualHost *>  
ServerName pgadmin.example.com  
WSGIProcessGroup pgadmin processes=1 threads=25
```

```
python-home=/path/to/python/virtualenv  
WSGIScriptAlias / /opt/pgAdmin4/web/pgAdmin4.wsgi  
<Directory /opt/pgAdmin4/web>  
WSGIProcessGroup pgadmin  
WSGIApplicationGroup %{GLOBAL}  
Order deny,allow  
Allow from all  
</Directory>  
</VirtualHost>
```

Note: If you're using Apache HTTPD 2.4 or later, replace the lines:
Order deny,allow

Allow from **all**

with:

Require **all** granted

Adjust as needed to suit your access control requirements.

Standalone Gunicorn Configuration [¶](#)

pgAdmin may be hosted by Gunicorn directly simply by running a command such as the one shown below. Note that this example assumes pgAdmin was installed using the Python Wheel (you may need to adjust the path to suit your installation):

```
gunicorn --bind 0.0.0.0:80 \  
--workers=1 \
```

```
--threads=25 \
--chdir /usr/lib/python3.7/dist-
packages/pgadmin4 \
pgAdmin4:app
```

Standalone uWSGI Configuration [¶](#)

pgAdmin may be hosted by uWSGI directly simply by running a command such as the one shown below. Note that this example assumes pgAdmin was installed using the Python Wheel (you may need to adjust the path to suit your installation):

```
uwsgi --http-socket 0.0.0.0:80 \
--processes 1 \
--threads 25 \
--chdir /usr/lib/python3.7/dist-packages/pgadmin4/
\
--mount /=pgAdmin4:app
```

NGINX Configuration with Gunicorn [¶](#)

pgAdmin can be hosted by Gunicorn, with NGINX in front of it. Note that these examples assume pgAdmin was installed using the Python Wheel (you may need to adjust the path to suit your installation).

To run with pgAdmin in the root directory of the server, start Gunicorn using a command similar to:

```
gunicorn --bind unix:/tmp/pgadmin4.sock \
```

```
--workers=1 \
--threads=25 \
--chdir /usr/lib/python3.7/dist-
packages/pgadmin4 \
pgAdmin4:app
```

And configure NGINX:

```
location / {
    include proxy_params;
    proxy_pass http://unix:/tmp/pgadmin4.sock;
}
```

Alternatively, pgAdmin can be hosted in a sub-directory (/pgadmin4 in this case) on the server. Start Gunicorn as when using the root directory, but configure NGINX as follows:

```
location /pgadmin4/ {
    include proxy_params;
    proxy_pass http://unix:/tmp/pgadmin4.sock;
    proxy_set_header X-Script-Name /pgadmin4;
}
```

NGINX Configuration with uWSGI [¶](#)

pgAdmin can be hosted by uWSGI, with NGINX in front of it. Note that these examples assume pgAdmin was installed using the

Python Wheel (you may need to adjust the path to suit your installation).

To run with pgAdmin in the root directory of the server, start Gunicorn using a command similar to:

```
uwsgi --socket /tmp/pgadmin4.sock \
--processes 1 \
--threads 25 \
--chdir /usr/lib/python3.7/dist-packages/pgadmin4/
\
--manage-script-name \
--mount /=pgAdmin4:app
```

And configure NGINX:

```
location / { try_files $uri @pgadmin4; }

location @pgadmin4 {
    include uwsgi_params;
    uwsgi_pass unix:/tmp/pgadmin4.sock;
}
```

Alternatively, pgAdmin can be hosted in a sub-directory (/pgadmin4 in this case) on the server. Start uWSGI, noting that the directory name is specified in the mount parameter:

```
uwsgi --socket /tmp/pgadmin4.sock \
--processes 1 \
```

```
--threads 25 \
--chdir /usr/lib/python3.7/dist-packages/pgadmin4/
\
--manage-script-name \
--mount /pgadmin4=pgAdmin4:app
```

Then, configure NGINX:

```
location = /pgadmin4 { rewrite ^ /pgadmin4/; }

location /pgadmin4 { try_files $uri @pgadmin4; }

location @pgadmin4 {
    include uwsgi_params;
    uwsgi_pass unix:/tmp/pgadmin4.sock;
}
```

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | 3.5 | 2.1) | 1.6 Development

[pgAdmin 4 4.19 documentation](#) > [Connecting To A Server](#) > [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
 - [Server Group Dialog](#)
 - [Server Dialog](#)
 - [Master Password](#)
 - [Connect to Server](#)
 - [Connection Error](#)
 - [Import/Export Servers](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Server Dialog

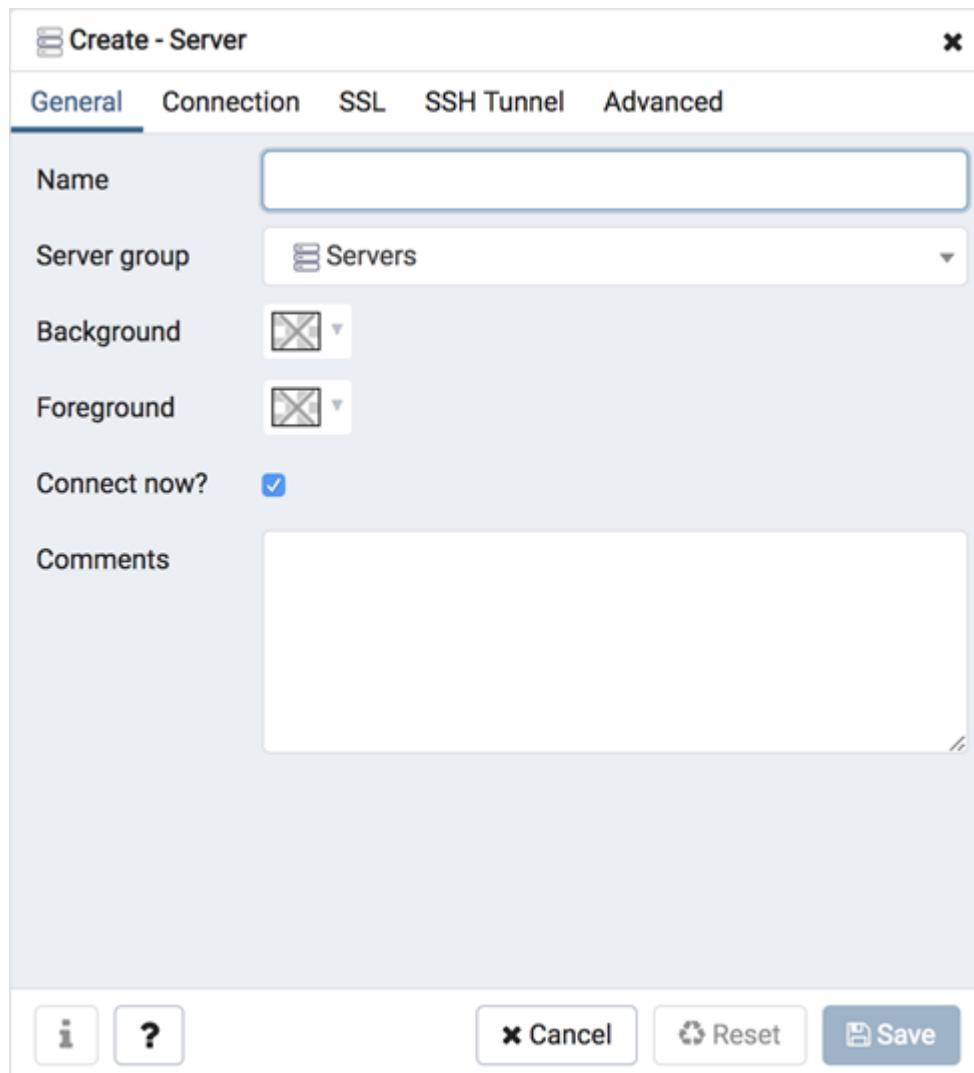
Use the Server dialog to describe a connection to a server. Note: you must ensure that the pg_hba.conf file of the server from which you are connecting allows connections from the host of the client.

Create - Server

General Connection SSL SSH Tunnel Advanced

Name	<input type="text"/>
Server group	<input type="button" value="Servers"/>
Background	<input checked="" type="checkbox"/>
Foreground	<input checked="" type="checkbox"/>
Connect now?	<input checked="" type="checkbox"/>
Comments	<input type="text"/>

Cancel Reset Save

A screenshot of the pgAdmin 4 'Create - Server' dialog box. The title bar says 'Create - Server'. Below it is a tab bar with 'General' selected, followed by 'Connection', 'SSL', 'SSH Tunnel', and 'Advanced'.
The 'General' tab contains the following fields:

- Name: A text input field.
- Server group: A dropdown menu set to 'Servers'.
- Background: A small icon with a red 'X' and a dropdown arrow.
- Foreground: A small icon with a red 'X' and a dropdown arrow.
- Connect now?: A checked checkbox.
- Comments: A large text area with a scroll bar.

At the bottom are three buttons: 'i' (Info), '?', 'Cancel' (with a red 'X'), 'Reset' (with a circular arrow), and 'Save' (highlighted in blue).

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Connecting To A Server](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Server Group Dialog](#)

Server Dialog



[Master Password](#)

[FAQ](#)

[Connect to Server](#)

[Connection Error](#)



[Import/Export Servers](#)

[Managing Cluster Objects](#)

[Latest Docs](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Management Basics](#)

[Backup and Restore](#)

[Ask a](#)

[Developer Tools](#)

[Question](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)



[Release Notes](#)

[Licence](#)

[Report a Bug](#)

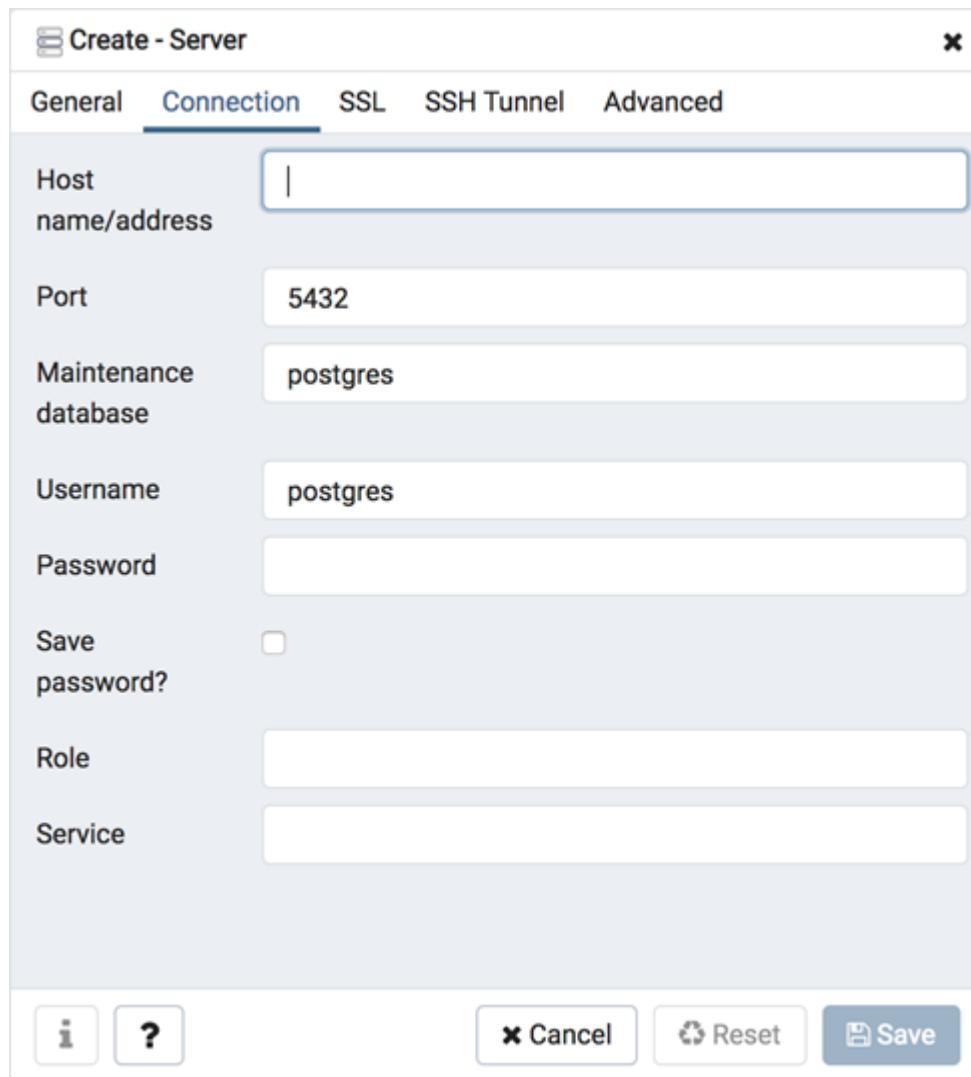
Server Dialog A small blue square icon with a white border, representing a link to an external resource.



Use the *Server* dialog to describe a connection to a server. Note: you

[Screenshots](#)

must ensure that the pg_hba.conf file of the server from which you are connecting allows connections from the host of the client.

A screenshot of a 'Create - Server' dialog box. The 'Connection' tab is selected. The form contains the following fields:

- Host name/address: A text input field containing a placeholder ' | '.
- Port: A text input field containing '5432'.
- Maintenance database: A text input field containing 'postgres'.
- Username: A text input field containing 'postgres'.
- Password: A text input field with a visible password mask.
- Save password?: A checkbox that is unchecked.
- Role: A text input field.
- Service: A text input field.

At the bottom are buttons for 'Cancel', 'Reset', and 'Save'.

Use the fields in the *General* tab to identify the server: Use the *Name* field to add a descriptive name for the server; the name specified will be displayed in the *Browser* tree control.

Use the drop-down list box in the *Server group* field to select the parent node for the server; the server will be displayed in the *Browser* tree control within the specified group.

Use the color-picker in the *Background* field to specify the background color for the server.

Use the color-picker in the *Foreground* field to specify the foreground color for the server.

If the *Connect now?* checkbox is checked, the client wil attempt a connection to the server upon completion of the dialog; this is the default

Provide a comment about the server in the *Comments* field.

Click the *Connection* tab to continue.

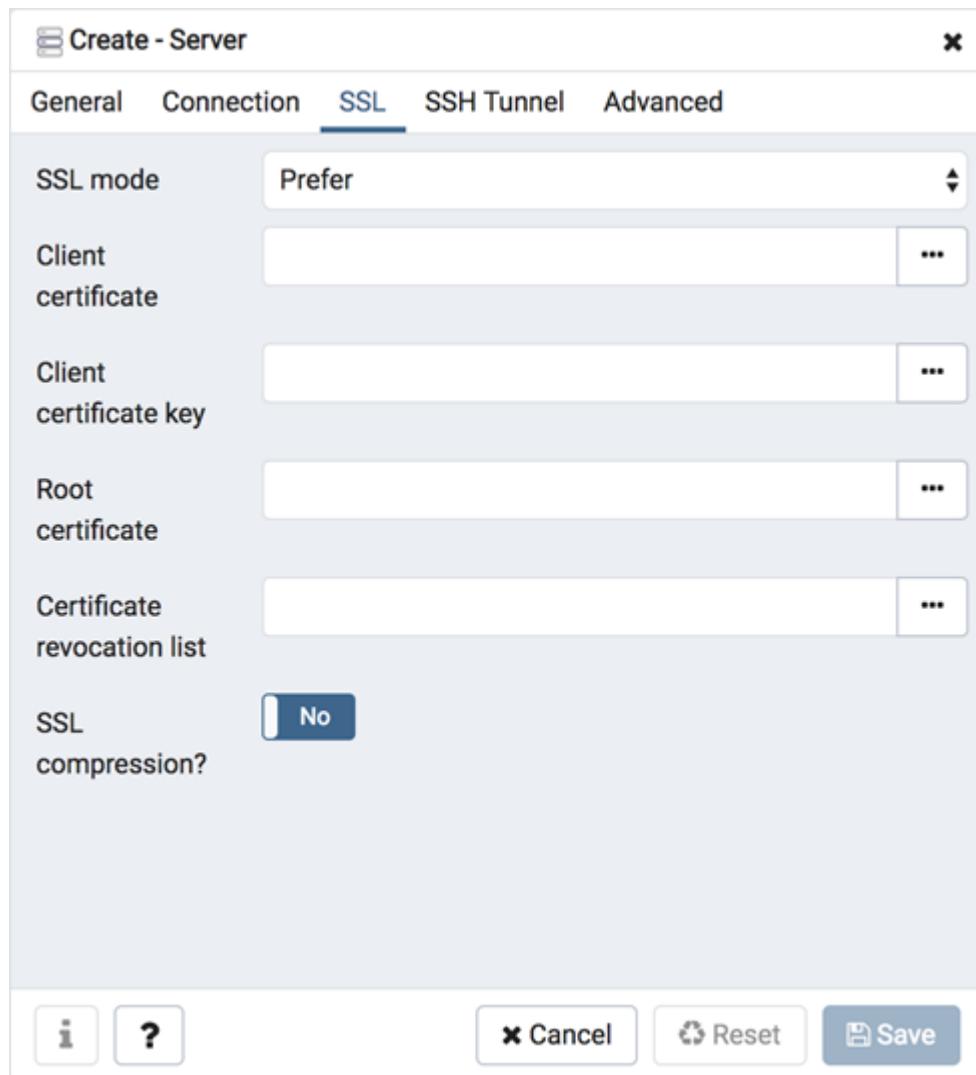
Use the fields in the *Connection* tab to configure a connection: Specify the IP address of the server host, or the ful y qualified domain name in the *Host name/address* field. If you provide a unix domain socket, the directory name must begin with a “/”.

Enter the listener port number of the server host in the *Port* field. The default is 5432.

Use the *Maintenance database* field to specify the name of the initial database to which the client wil connect. If you wil be using pgAgent or adminpack objects, the pgAgent schema and adminpack objects should be instal ed on that database.

Use the *Username* field to specify the name of a role that wil be used when authenticating with the server.

Use the *Password* field to provide a password that wil be supplied when authenticating with the server.



Check the box next to *Save password?* to instruct pgAdmin to save the password for future use. Use [Clear Saved Password](#) to remove the saved password.

Use the *Role* field to specify the name of a role that has privileges that will be conveyed to the client after authentication with the server.

This selection allows you to connect as one role, and then assume the permissions of this specified role after the connection is established. Note that the connecting role must be a member of the role specified.

Use the *Service* field to specify the service name. For more information, see [Section 33.16 of the Postgres documentation](#).

Click the *SSL* tab to continue.

Use the fields in the *SSL* tab to configure SSL: Use the drop-down list box in the *SSL* field to select the type of SSL

connection the server should use. For more information about using SSL encryption, see [Section 33.18 of the Postgres documentation](#).

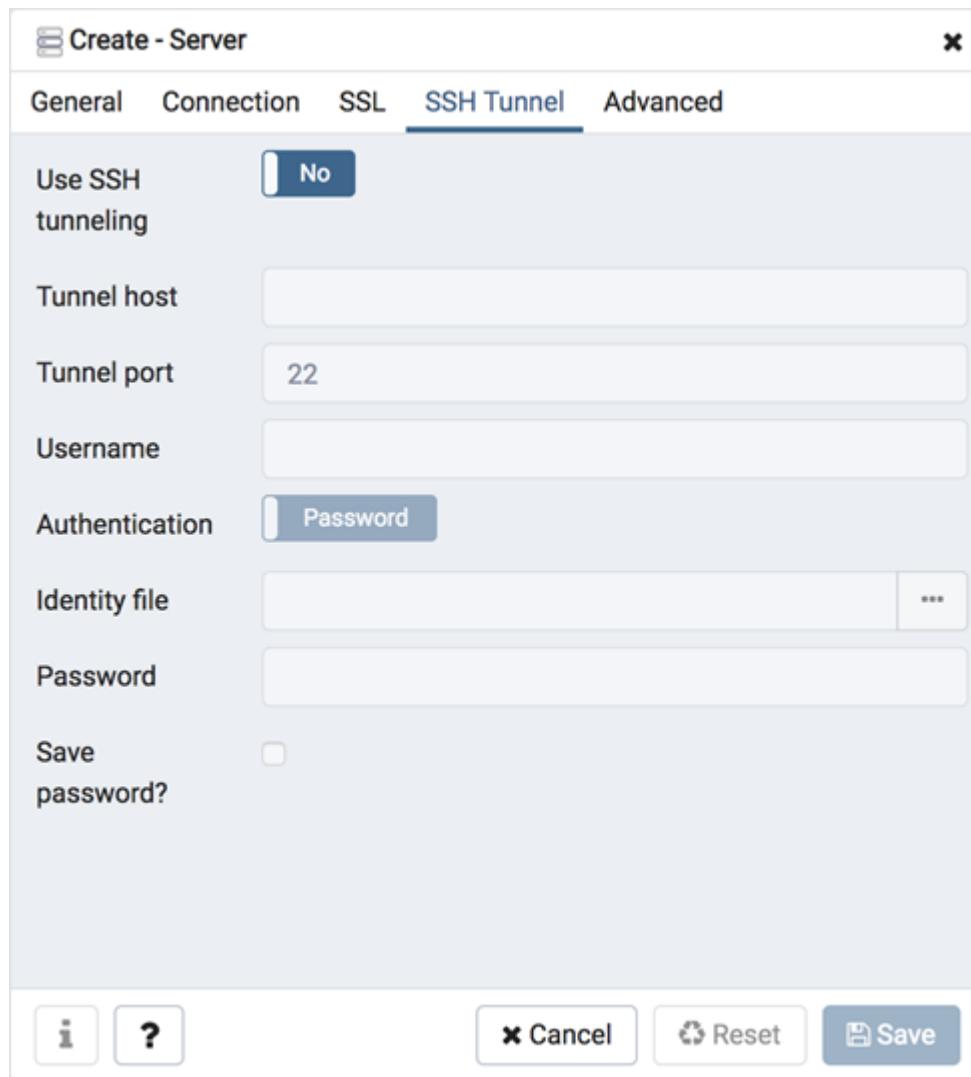
If pgAdmin is installed in Server mode (the default mode), you can use the platform-specific File manager dialog to upload files that support SSL

encryption to the server. To access the File manager dialog, click the icon that is located to the right of each of the following fields.

Use the *Client certificate* field to specify the file containing the client SSL certificate. This file will replace the default

`~/.postgresql/postgresql.crt` if pgAdmin is installed in Desktop mode, and `<STORAGE_DIR>/<USERNAME>/.postgresql/postgresql.crt` if pgAdmin is installed in Web mode. This parameter is ignored if an SSL connection is not made.

Use the *Client certificate key* field to specify the file containing the secret key used for the client certificate. This file will replace the



default `~/.postgresql/postgresql.key` if pgAdmin is installed in Desktop mode, and

`<STORAGE_DIR>/<USERNAME>/.postgresql/postgresql.key` if pgAdmin is installed in Web mode. This parameter is ignored if an SSL connection is not made.

Use the *Root certificate* field to specify the file containing the SSL certificate authority. This file will replace the default `~/.postgresql/root.crt`. This parameter is ignored if an SSL connection is not made.

Use the *Certificate revocation list* field to specify the file containing the SSL certificate revocation list. This list will replace the default list, found in `~/.postgresql/root.crl`. This parameter is ignored if an SSL connection is not made.

When *SSL compression?* is set to *True*, data sent over SSL connections will be compressed. The default value is *False* (compression is disabled). This parameter is ignored if an SSL connection is not made.

Warning

In Server mode, certificates, private keys, and the revocation list are stored in the per-user file storage area on the server, which is owned by the user account under which the pgAdmin server process is run. This means that administrators of the server may be able to access those files; appropriate caution should be taken before choosing to use this feature.

Click the *SSH Tunnel* tab to continue.

Use the fields in the *SSH Tunnel* tab to configure SSH Tunneling:

You can use the “SSH Tunnel” tab to connect pgAdmin (through an intermediary proxy host) to a server that resides on a network to which the client may not be able to connect directly.

Set “Use SSH tunneling” to *Yes* to specify that pgAdmin should use an SSH tunnel when connecting to the specified server.

Specify the name or IP address of the SSH host (through which client connections will be forwarded) in the *Tunnel host* field.

Specify the port of the SSH host (through which client connections will be forwarded) in the *Tunnel port* field.

Specify the name of a user with login privileges for the SSH host in the *Username* field.

Specify the type of authentication that will be used when connecting to the SSH host in the *Authentication* field:

Select the *Password* option to specify that pgAdmin will use a password for authentication to the SSH host. This is the default.

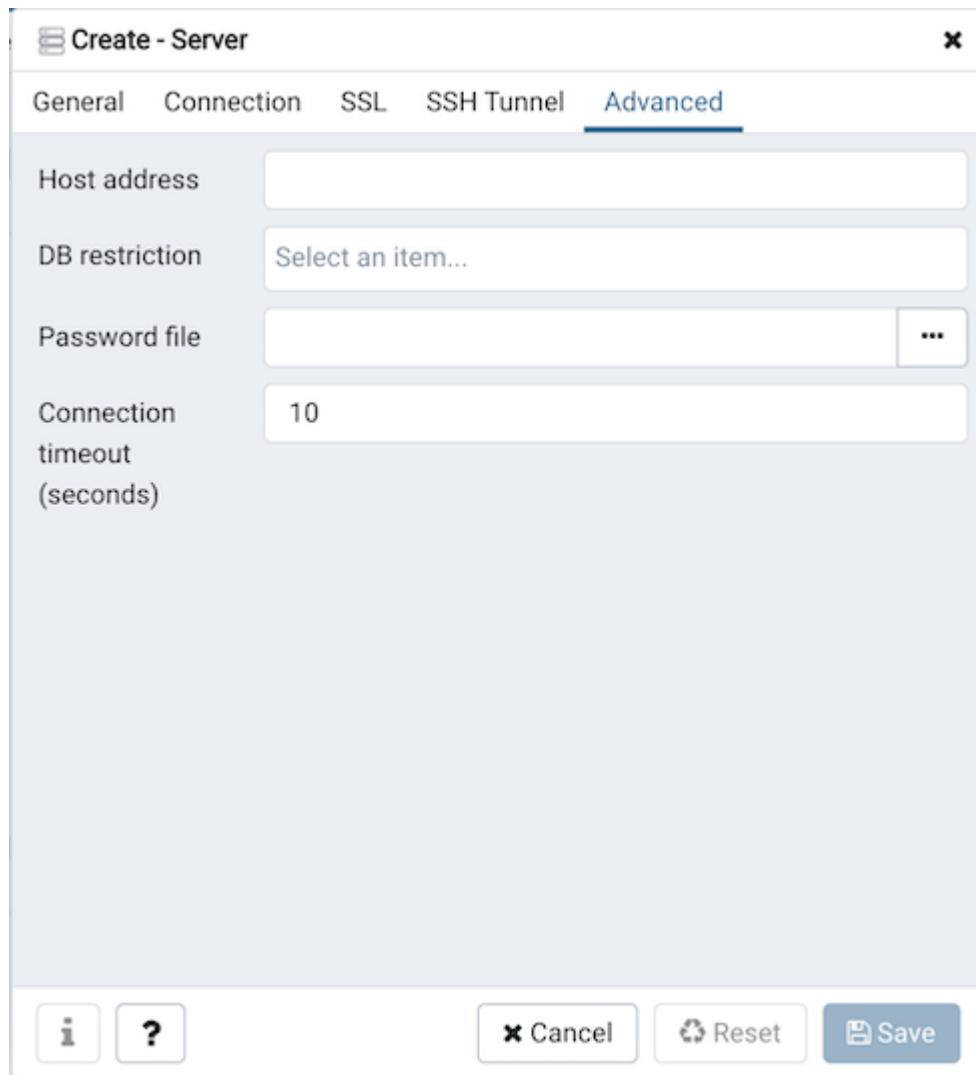
Select the *Identity file* to specify that pgAdmin will use a private key file when connecting.

If the SSH host is expecting a private key file for authentication, use the *Identity file* field to specify the location of the key file.

If the SSH host is expecting a password of the user name or an identity file if being used, use the *Password* field to specify the password.

Check the box next to *Save password?* to instruct pgAdmin to save the password for future use. Use [Clear SSH Tunnel Password](#) to remove the saved password.

Click the *Advanced* tab to continue.



Use the fields in the *Advanced* tab to configure a connection: Specify the IP address of the server host in the *Host address* field.

Using this field to specify the host IP address may save time by avoiding a DNS lookup on connection, but it may be useful to specify both a host name and address when using Kerberos, GSSAPI, or SSPI authentication methods, as well as for verify-ful SSL certificate verification.

Use the *DB restriction* field to provide a SQL restriction that will be used against the pg_database table to limit the databases that you see. For example, you might enter: *live_db test_db* so that only

`live_db` and `test_db` are shown in the pgAdmin browser. Separate entries with a comma or tab as you type.

Use the *Password File* field to specify the location of a password file (`.pgpass`). A `.pgpass` file allows a user to login without providing a [password when they connect. For more information, see Section 33.15 of the Postgres documentation.](#)

Use the *Connection timeout* field to specify the maximum wait for connection, in seconds. Zero or not specified means wait indefinitely.

It is not recommended to use a timeout of less than 2 seconds. By default it is set to 10 seconds.

Note

The password file option is only supported when pgAdmin is using libpq v10.0 or later to connect to the server.

Click the **Save** button to save your work.

Click the **Cancel** button to exit without saving your work.

Click the **Reset** button to return the values specified on the Server dialog to their original condition.

[Clear Saved Passwords](#)

Quick Links



[Online Demo](#)



[Download](#)



[FAQ](#)



[Latest Docs](#)



[Ask a
Question](#)



[Report a Bug](#)



[Screenshots](#)

This page in other versions: Latest (4.19 | 4.18 | 4.17 | Development)

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) »

[previous](#) [next](#) [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
 - [Deployment](#)
 - [Login Dialog](#)
 - [User Management Dialog](#)
 - [Change User Password Dialog](#)
 - [User Interface](#)
 - [Menu Bar](#)
 - [Toolbar](#)
 - [Tabbed Browser](#)
 - [Tree Control](#)
 - [Preferences Dialog](#)
 - [Keyboard Shortcuts](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Tabbed Browser

The right pane of the pgAdmin window features a collection of tabs that display information about the object currently selected in the pgAdmin tree control in the left window. Select a tab to access information about the highlighted object in the tree control.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Getting Started](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Deployment](#)

[Login Dialog](#)

[User Management Dialog](#)



[Change User Password Dialog](#)

[FAQ](#)

[User Interface](#)

[Menu Bar](#)



[Toolbar](#)

Tabbed Browser

[Latest Docs](#)

[Tree Control](#)

[Preferences Dialog](#)



[Keyboard Shortcuts](#)

[Connecting To A Server](#)

[Ask a](#)

[Managing Cluster Objects](#)

[Question](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Management Basics](#)

[Backup and Restore](#)

[Report a Bug](#)

[Developer Tools](#)

[pgAgent](#)



[pgAdmin Project Contributions](#)

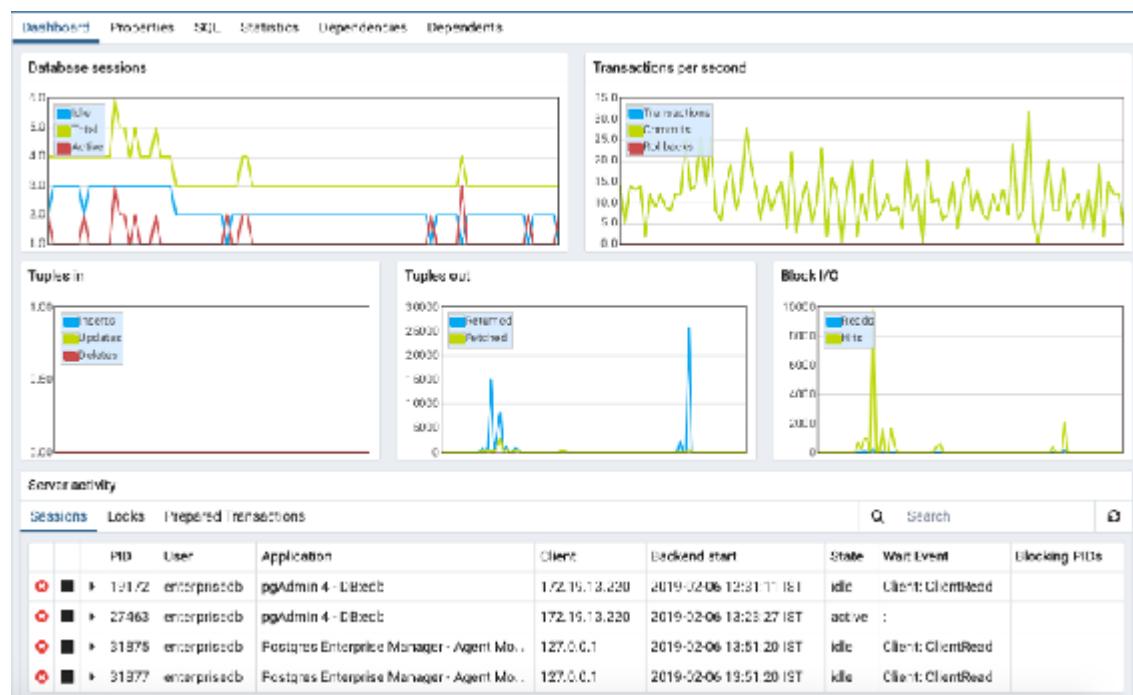
[Release Notes](#)

[Screenshots](#)

[Licence](#)

Tabbed Browser

The right pane of the *pgAdmin* window features a collection of tabs that display information about the object currently selected in the *pgAdmin* tree control in the left window. Select a tab to access information about the highlighted object in the tree control.



The graphs on the *Dashboard* tab provides an active analysis of the usage statistics for the selected server or database:

The *Server sessions* or *Database sessions* graph displays the interactions with the server or database.

The *Transactions per second* graph displays the commits, rol backs, and total transactions per second that are taking place on the server or database.

The *Tuples in* graph displays the number of tuples inserted, updated, and deleted on the server or database.

The *Tuples out* graph displays the number of tuples fetched and returned from the server or database.

The *Block I/O* graph displays the number of blocks read from the filesystem or fetched from the buffer cache (but not the operating system's file system cache) for the server or database.

The *Server activity* panel displays information about sessions, locks, prepared transactions, and server configuration (if applicable). The information is presented in context-sensitive tables. Use controls located above the table to:

Click the *Refresh* button to update the information displayed in each table.

Enter a value in the *Search* box to restrict the table content to one or more sessions that satisfy the search criteria. For example, you can enter a process ID to locate a specific session, or a session state (such as *idle*) to locate all of the sessions that are in an idle state.

You can use icons in the *Sessions* table to review or control the state of a session:

Use the *Terminate* icon (located in the first column) to stop a session and remove the session from the table. Before the server terminates the session, you will be prompted to confirm your selection.

Use the *Cancel* icon (located in the second column) to terminate an active query without closing the session. Before canceling the query, the server will prompt you to confirm your selection. When you cancel a query, the value displayed in the *State* column of the table will be

Dashboard Properties SQL Statistics Dependencies Dependents

	Database	Owner	Comment
<input type="checkbox"/>	edb	enterprisedb	
<input type="checkbox"/>	postgres	enterprisedb	default administrative connection database

Dashboard Properties SQL Statistics Dependencies Dependents

	Database	Owner	Comment
<input type="checkbox"/>	edb	enterprisedb	
<input type="checkbox"/>	postgres	enterprisedb	default administrative connection database

Dashboard Properties SQL Statistics Dependencies Dependents + Query editor edit

General

Name	edb_enterprisedb
OID	1E705
Owner	enterprisedb
Comment	EnterpriseDB Foreign Data Wrapper for PostgreSQL

Definition

Schemas	
Renounceable?	No
Version	1.0

Dashboard Properties SQL Statistics Dependencies Dependents

```

1 -- Database: edb
2
3 -- DROP DATABASE edb;
4
5 CREATE DATABASE edb
6   WITH
7     OWNER = enterprisedb
8     ENCODING = 'UTF8'
9     LC_COLLATE = 'en_US.UTF-8'
10    LC_CTYPE = 'en_US.UTF-8'
11    TABLESPACE = pg_default
12    CONNECTION LIMIT = -1
  
```

updated from *Active* to *Idle*. The session wil remain in the table until the session is terminated.

Use the *Details* icon (located in the third column) to open the *Details* tab; the tab displays information about the selected session.

The *Properties* tab displays information about the object selected.

Click the *Delete* icon in the toolbar under the browser tab to delete the selected objects in the Properties panel.

Click the *Drop Cascade* icon in the toolbar under the browser tab to delete the selected objects and all dependent objects in the Properties panel.

Click the *Edit* icon in the toolbar under the browser tabs to launch the *Properties* dialog for the selected object.

To preserve any changes to the *Properties* dialog, click the *Save* icon; your modifications will be displayed in the updated *Properties* tab.

Details about the object highlighted in the tree control are displayed in one or more collapsible panels. You can use the arrow to the left of each panel label to open or close a panel.

The *SQL* tab displays the SQL script that created the highlighted object, and when applicable, a (commented out) SQL statement that will *DROP*

the selected object. You can copy the SQL statements to the editor of

The screenshot shows a statistics table with two columns: 'Statistic' and 'Value'. The statistics listed are:

Statistic	Value
Backends	5
Autocommitted	0.000500
Batched back	36
Blks read	5259146
Blks read	1M 31.25 MB
Blks returned	2269197300
Blks fetched	5487615212
Blks inserted	13520
Blks updated	152
Blks deleted	48
Last statistics reset	2018-12-21 14:30:08.829022+05:30
Tablespace conflicts	0
Lock conflicts	0
Snapshot conflicts	0
Bufpage conflicts	0
Deadlock conflicts	0
Temporary files	0
Size of temporary file	0 bytes
Deadlocks	0
Block read time	0
Block write time	0
Blks	18 MB

your choice using cut and paste shortcuts.

The *Statistics* tab displays the statistics gathered for each object on the tree control; the statistics displayed in the table vary by the type of object that is selected. Click a column heading to sort the table by the data displayed in the column; click again to reverse the sort order. The following table lists some of the statistics that are available: Panel

Description

PID

The process ID associated with the row.

User

The name of the user that owns the object.

Database

displays the database name.

Backends

displays the number of current connections to the database.

Backend start The start time of the backend process.

Xact

displays the number of transactions committed to the *Committed* database within the last week.

Xact Roll ed

displays the number of transactions rolled back within the *Back* last week.

Blocks Read displays the number of blocks read from memory (in megabytes) within the last week.

Blocks Hit

displays the number of blocks hit in the cache (in megabytes) within the last week.

Tuples

displays the number of tuples returned within the last *Returned* week.

Tuples

displays the number of tuples fetched within the last *Fetched* week.

Tuples

displays the number of tuples inserted into the database *Inserted* within the last week.

Tuples

displays the number of tuples updated in the database *Updated* within the last week.

Tuples

displays the number of tuples deleted from the database *Deleted* within the last week.

Last statistics displays the time of the last statistics reset for the reset

database.

Type	Name	Restriction
Schema	public	normal

Panel

Description

Tablespace

displays the number of queries canceled because of *conflicts*

recovery conflict with dropped tablespaces in database.

Lock conflicts displays the number of queries canceled because of recovery conflict with locks in database.

Snapshot

displays the number of queries canceled because of
conflicts
recovery conflict with old snapshots in database.

Bufferpin

displays the number of queries canceled because of
conflicts
recovery conflict with pinned buffers in database.

Temporary

displays the total number of temporary files, including *files*
those used by the statistics collector.

Size of

displays the size of the temporary files.

temporary

files

Deadlocks

displays the number of queries canceled because of a recovery
conflict with deadlocks in database.

Block read

displays the number of milliseconds required to read the *time*
blocks read.

Block write

displays the number of milliseconds required to write the *time* blocks read.

Size

displays the size (in megabytes) of the selected database.

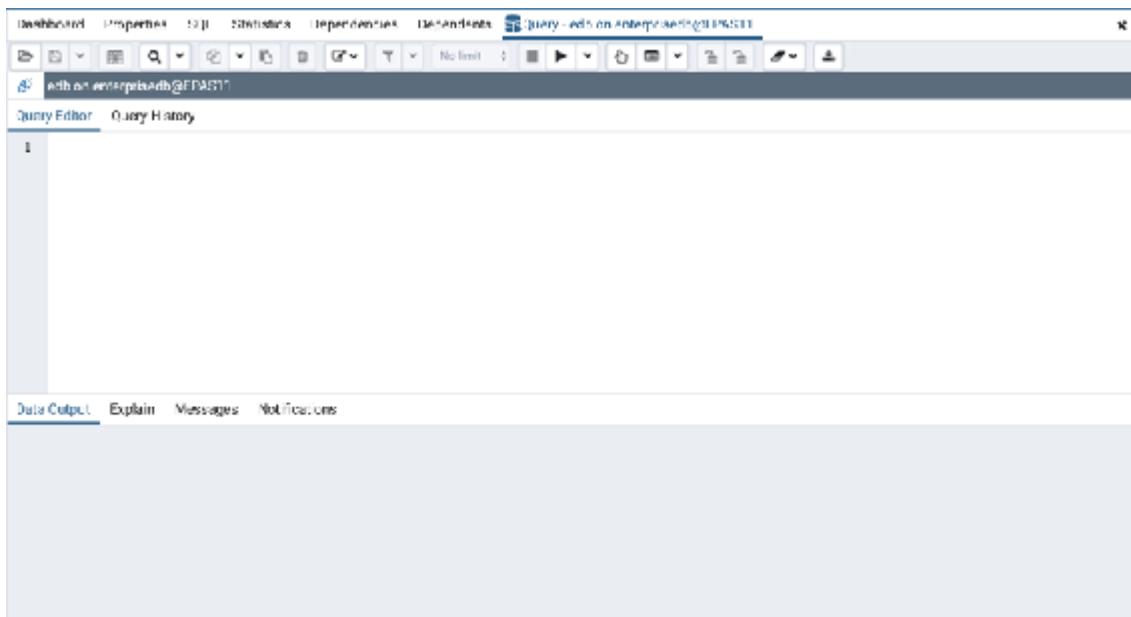
The *Dependencies* tab displays the objects on which the currently selected object depends. If a dependency is dropped, the object currently selected in the pgAdmin tree control will be affected. To ensure the integrity of the entire database structure, the database server makes sure that you do not accidentally drop objects that other objects depend on; you must use the DROP CASCADE command to remove an object with a dependency.

The *Dependencies* table displays the following information: The *Type* field specifies the parent object type.

The *Name* field specifies the identifying name of the parent object.

The *Restriction* field describes the dependency relationship between the currently selected object and the parent:

Type	Name	Restriction
✓ Check	public.annular_ref_sys_id_check	none
⚠ Primary Key	public.annular_ref_sys_pk	none
✓ Check	public.spatial_ref_sys_id_check	normal



If the field is *auto*, the selected object can be dropped separately from the parent object, and will be dropped if the parent object is dropped.

If the field is *internal*, the selected object was created during the creation of the parent object, and will be dropped if the parent object is dropped.

If the field is *normal*, the selected object can be dropped without dropping the parent object.

If the field is *blank*, the selected object is required by the system, and cannot be dropped.

The *Dependents* tab displays a table of objects that depend on the object currently selected in the *pgAdmin* browser. A dependent object can be dropped without affecting the object currently selected in the *pgAdmin* tree control.

The *Type* field specifies the dependent object type.

The *Name* field specifies the identifying name for the dependent object.

The *Database* field specifies the database in which the object resides.

Additional tabs open when you access the extended functionality offered by pgAdmin tools (such as the Query tool, Debugger, or SQL editor). Use the close icon (X) located in the upper-right corner of each tab to close the tab when you are finished using the tool. Like permanent tabs, these tabs may be repositioned in the pgAdmin client window.

By default, each time you open a tool, pgAdmin will open a new browser tab. You can control this behavior by modifying the *Display* node of the *Preferences* dialog for each tool. To open the *Preferences* dialog, select *Preferences* from the *File* menu.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
 - [Cast Dialog](#)
 - [Collation Dialog](#)
 - [Domain Dialog](#)
 - [Domain Constraints Dialog](#)
 - [Event Trigger Dialog](#)
 - [Extension Dialog](#)
 - [Foreign Data Wrapper Dialog](#)
 - [Foreign Server Dialog](#)
 - [Foreign Table Dialog](#)
 - [FTS Configuration Dialog](#)
 - [FTS Dictionary Dialog](#)
 - [FTS Parser Dialog](#)
 - [FTS Template Dialog](#)
 - [Function Dialog](#)
 - [Language Dialog](#)
 - [Materialized View Dialog](#)
 - [Package Dialog](#)
 - [Procedure Dialog](#)
 - [Schema Dialog](#)
 - [Sequence Dialog](#)
 - [Synonym Dialog](#)
 - [Trigger Function Dialog](#)
 - [Type Dialog](#)
 - [User Mapping Dialog](#)
 - [View Dialog](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Synonym Dialog

Use the Synonym dialog to substitute the name of a target object with a user-defined synonym.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Managing Database Objects](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

[Managing Cluster Objects](#)

[Managing Database Objects](#)



[Cast Dialog](#)

[FAQ](#)

[Collation Dialog](#)

[Domain Dialog](#)



[Domain Constraints Dialog](#)

[Event Trigger Dialog](#)

[Latest Docs](#)

[Extension Dialog](#)

[Foreign Data Wrapper Dialog](#)



[Foreign Server Dialog](#)

[Foreign Table Dialog](#)

[Ask a](#)

[FTS Configuration Dialog](#)

[Question](#)

[FTS Dictionary Dialog](#)

[FTS Parser Dialog](#)



[FTS Template Dialog](#)

[Function Dialog](#)

[Report a Bug](#)

[Language Dialog](#)

[Materialized View Dialog](#)

[□](#)

[Package Dialog](#)

[Procedure Dialog](#)

[Screenshots](#)

[Schema Dialog](#)

[Sequence Dialog](#)

Synonym Dialog

[Trigger Function Dialog](#)

[Type Dialog](#)

[User Mapping Dialog](#)

[View Dialog](#)

[Creating or Modifying a Table](#)

[Management Basics](#)

[Backup and Restore](#)

[Developer Tools](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Synonym Dialog 

Use the *Synonym* dialog to substitute the name of a target object with a user-defined synonym.



The *Synonym* dialog organizes the development of a synonym through the *General* tab. The *SQL* tab displays the SQL code generated by dialog selections.

Use the fields in the *General* tab to identify the synonym: Use the *Name* field to specify the name of synonym. The name will be displayed in the *pgAdmin* tree control.

Select the name of the schema in which the synonym will reside from the drop-down listbox in the *Schema* field.

In the definition panel, identify the target:

Use the drop-down listbox next to *Target Type* to select the type of object referenced by the synonym.

Use the drop-down listbox next to *Target Schema* to select the name of the schema in which the object resides.

Use the drop-down listbox next to *Target Object* to select the name of the object referenced by the synonym.

Click the *SQL* tab to continue.

Your selections and entries in the *Synonym* dialog generate a SQL command.

 Create - Synonym x

General Definition **SQL**

```
1 CREATE OR REPLACE PUBLIC SYNONYM employee
2   FOR public.emp;
```

i ? ✖ Cancel ↻ Reset Save

The example creates a synonym for the `emp` table named `emp_hist`.

Click the **Save** button to save work.

Click the **Cancel** button to exit without saving work.

Click the **Reset** button to restore configuration parameters.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) > [Connecting To A Server](#) > [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Getting Started](#)
- [Connecting To A Server](#)
 - [Server Group Dialog](#)
 - [Server Dialog](#)
 - [Master Password](#)
 - [Connect to Server](#)
 - [Connection Error](#)
 - [Import/Export Servers](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Server Group Dialog

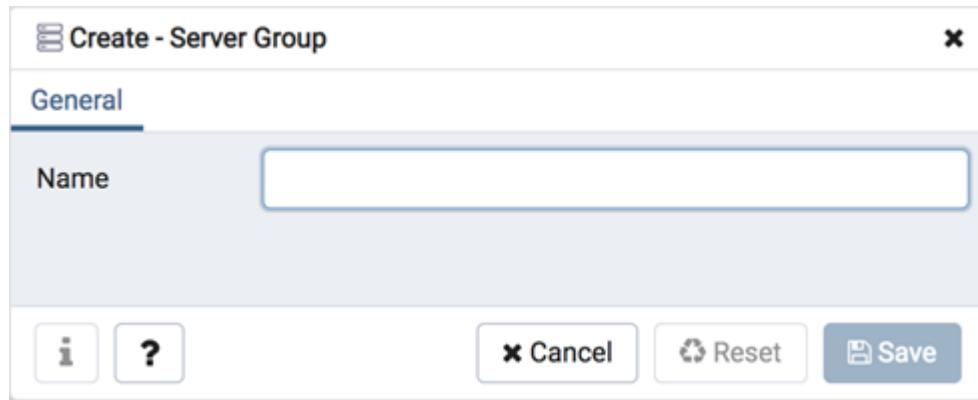
Use the *Server Group* dialog to add a new server group. Assign servers to server groups to simplify management of multiple servers. Server groups are displayed as part of the *pgAdmin* tree control.



Use the *Name* field on the *Server Group* dialog to specify a name that will identify the server group in the *pgAdmin* tree control.

- Click the *Save* button to save work.
- Click the *Cancel* button to exit without saving work.
- Click the *Reset* button to restore configuration parameters.

To create server connections in a server group, right click on the named server group and select the *Create* option to open the *Create - Server* dialog.



Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Connecting To A Server](#) »

[previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Getting Started](#)

[Download](#)

[Connecting To A Server](#)

Server Group Dialog

[Server Dialog](#)



[Master Password](#)

[FAQ](#)

[Connect to Server](#)

[Connection Error](#)



[Import/Export Servers](#)

[Managing Cluster Objects](#)

[Latest Docs](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)



[Management Basics](#)

[Backup and Restore](#)

[Ask a](#)

[Developer Tools](#)

[Question](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)



[Release Notes](#)

[Licence](#)

[Report a Bug](#)

Server Group Dialog



Use the *Server Group* dialog to add a new server group. Assign servers to

[Screenshots](#)

server groups to simplify management of multiple servers. Server groups are displayed as part of the *pgAdmin* tree control.

Use the *Name* field on the *Server Group* dialog to specify a name that will identify the server group in the *pgAdmin* tree control.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.

To create server connections in a server group, right click on the named server group and select the *Create* option to open the *Create - Server* dialog.

Quick Links

[Online Demo](#)[Download](#)[FAQ](#)[Latest Docs](#)[Ask a Question](#)[Report a Bug](#)[Screenshots](#)

This page in other versions: Latest | 4.19 | 4.18 | 4.17 | 3.5 | 2.1 | 1.6 | Development

[pgAdmin 4 documentation](#) » [Creation or Modifying a Table](#) » [previous](#) | [next](#) | [index](#)

Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

- [Table Dialog](#)
 - [Example](#)
- [Getting Started](#)
- [Connecting To A Server](#)
- [Managing Cluster Objects](#)
- [Managing Database Objects](#)
- [Creating or Modifying a Table](#)
 - [Check Dialog](#)
 - [Column Dialog](#)
 - [Compound Trigger Dialog](#)
 - [Exclusion Constraint Dialog](#)
 - [Foreign key Dialog](#)
 - [Index Dialog](#)
 - [Primary key Dialog](#)
 - [Rule Dialog](#)
 - [Table Dialog](#)
 - [Trigger Dialog](#)
 - [Unique Constraint Dialog](#)
- [Management Basics](#)
- [Backup and Restore](#)
- [Developer Tools](#)
- [pgAgent](#)
- [pgAdmin Project Contributions](#)
- [Release Notes](#)
- [Licence](#)

Table Dialog

Use the *Table* dialog to create or modify a table.

The *Table* dialog organizes the development of a table through the following dialog tabs: *General*, *Columns*, *Constraints*, *Advanced*, *Parameter*, and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

Quick Links

This page in other versions: [Latest \(4.19\)](#) | [4.18](#) | [4.17](#) | [3.6](#) | [2.1](#) | [1.6](#) | [Development](#)

[pgAdmin 4 4.19 documentation](#) » [Creating or Modifying a Table](#) » [previous](#) | [next](#) | [index](#)



Warning: This documentation is for a pre-release version of pgAdmin 4

[Online Demo](#)

Contents



[Table Dialog](#)

[Download](#)

[Example](#)

[Getting Started](#)



[Connecting To A Server](#)

[Managing Cluster Objects](#)

[FAQ](#)

[Managing Database Objects](#)

[Creating or Modifying a Table](#)

[□](#)

[Check Dialog](#)

[Column Dialog](#)

[Latest Docs](#)

[Compound Trigger Dialog](#)

[Exclusion Constraint Dialog](#)

[□](#)

[Foreign key Dialog](#)

[Index Dialog](#)

[Ask a](#)

[Primary key Dialog](#)

[Question](#)

[Rule Dialog](#)

[Table Dialog](#)

[□](#)

[Trigger Dialog](#)

[Report a Bug](#)

[Unique Constraint Dialog](#)

[Management Basics](#)

[Backup and Restore](#)



[Developer Tools](#)

[Screenshots](#)

[pgAgent](#)

[pgAdmin Project Contributions](#)

[Release Notes](#)

[Licence](#)

Table Dialog 

Use the *Table* dialog to create or modify a table.

The *Table* dialog organizes the development of a table through the following dialog tabs: *General*, *Columns*, *Constraints*, *Advanced*, *Parameter*, and *Security*. The *SQL* tab displays the SQL code generated by dialog selections.

Create - Table

General Columns Constraints Advanced Partition Parameter Security SQL

Name: perm.test

Owner: enterprisesdb

Schema: public

Tablespace: Select from the list

Partitioned Table?: No

Comment:

i ? **Cancel** **Reset** **Save**

The screenshot shows the 'Create - Table' dialog box in pgAdmin. The 'General' tab is selected. In the 'Name' field, 'perm.test' is entered. The 'Owner' dropdown contains 'enterprisesdb'. The 'Schema' dropdown shows 'public'. The 'Tablespace' dropdown says 'Select from the list'. The 'Partitioned Table?' checkbox is unchecked. A large empty text area for 'Comment' is present. At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save', with 'Save' being the primary button.

Use the fields in the *General* tab to identify the table: Use the *Name* field to add a descriptive name for the table. A table cannot have the same name as any existing table, sequence, index, view, foreign table, or data type in the same schema. The name specified will be displayed in the *pgAdmin* tree control. This field is required.

Select the owner of the table from the drop-down listbox in the *Owner* field. By default, the owner of the table is the role that creates the table.

Select the name of the schema in which the table will reside from the drop-down listbox in the *Schema* field.

Use the drop-down listbox in the *Tablespace* field to specify the tablespace in which the table will be stored.

Move the *Partitioned Table?* switch to the Yes in case you want to create a partitioned table. Option is available for PostgreSQL 10 and above.

Store notes about the table in the *Comment* field.

Click the *Columns* tab to continue.

The screenshot shows the 'Create - Table' dialog box with the 'Columns' tab selected. At the top, there are tabs for General, Columns, Constraints, Advanced, Partition, Parameters, Security, and SQL. Below the tabs, there is a section for 'Inherited from table(s)' with a dropdown labeled 'Select to inherit from...'. A large table below is titled 'Columns' and has columns for Name, Data type, Length/Precision, Scale, Not NULL?, and Primary key?. At the bottom of the dialog are buttons for i, ?, Cancel, Reset, and Save.

Use the drop-down listbox next to *Inherited from table(s)* to specify any parent table(s); the table will inherit columns from the selected parent table(s). Click inside the *Inherited from table(s)* field to select a table name from a drop-down list. Repeat to add any other parent tables. Delete a selected table by clicking the x to the left of the parent name. Note that inherited column names and datatypes are not editable in the current dialog; they must be modified at the parent level.

Click the *Add* icon (+) to specify the names of columns and their datatypes in the *Columns* table:

Use the *Name* field to add a descriptive name for the column.

Use the drop-down listbox in the *Data type* field to select a data type for the column. This can include array specifiers. For more information on the data types supported by PostgreSQL, refer to Chapter 8 of the core documentation.

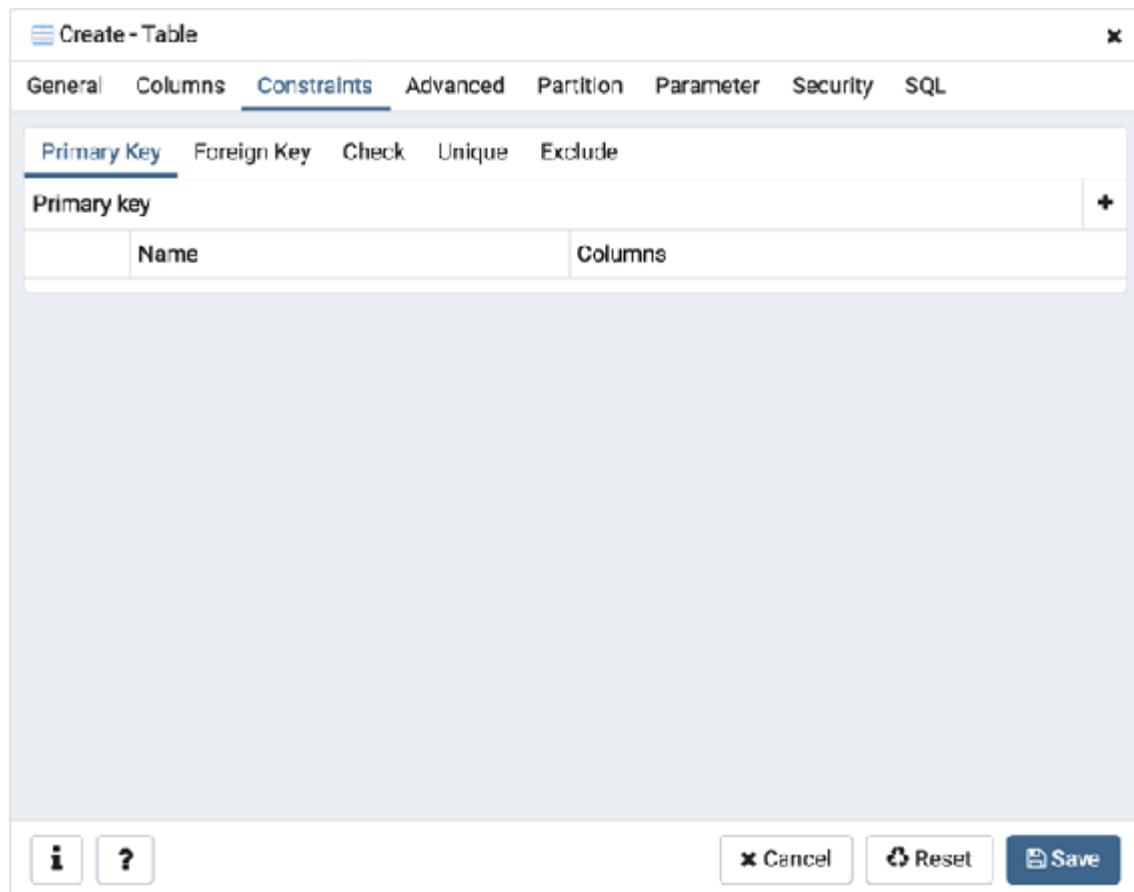
If enabled, use the *Length/Precision* and *Scale* fields to specify the maximum number of significant digits in a numeric value, or the maximum number of characters in a text value.

Move the *Not NULL?* switch to the *Yes* position to require a value in the column field.

Move the *Primary key?* switch to the *Yes* position to specify the column is the primary key constraint.

Click the *Add* icon (+) to add additional columns; to discard a column, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *Constraints* tab to continue.



Use the fields in the *Constraints* tab to provide a table or column constraint. Optional constraint clauses specify constraints (tests) that new or updated rows must satisfy for an *INSERT* or *UPDATE* operation to succeed. Select the appropriate constraint type by selecting one of the following tabs on the *Constraints* panel:

Tab Name

Constraint

Primary Key Provides a unique identifier for each row in the table.

Foreign Key Maintains referential integrity between two tables.

Check

Requires data satisfies an expression or condition before insertion or modification.

Unique

Ensures that the data contained in a column, or a group of columns, is unique among all the rows in the table.

Exclude

Guarantees that if any two rows are compared on the specified column or expression (using the specified operator), at least one of the operator comparisons will return false or null .

To add a primary key for the table, select the *Primary Key* tab, and click the *Add* icon (+). To define the primary key, click the *Edit* icon to the left of the *Trash* icon. A dialog similar to the *Primary key* dialog (accessed by right clicking on *Constraints* in the pgAdmin tree control) opens.

Use the fields in the *General* tab to identify the primary key: Use the *Name* field to add a descriptive name for the primary key constraint. The name will be displayed in the pgAdmin tree control.

Provide notes about the primary key in the *Comment* field.

Click the *Definition* tab to continue.



Use the fields in the *Definition* tab to define the primary key constraint: Click inside the *Columns* field and select one or more column names from the drop-down listbox. To delete a selection, click the *x* to the left of the column name. The primary key constraint should be different from any unique constraint defined for the same table; the selected column(s) for the constraints must be distinct.

Select the name of the tablespace in which the primary key constraint will reside from the drop-down listbox in the *Tablespace* field.

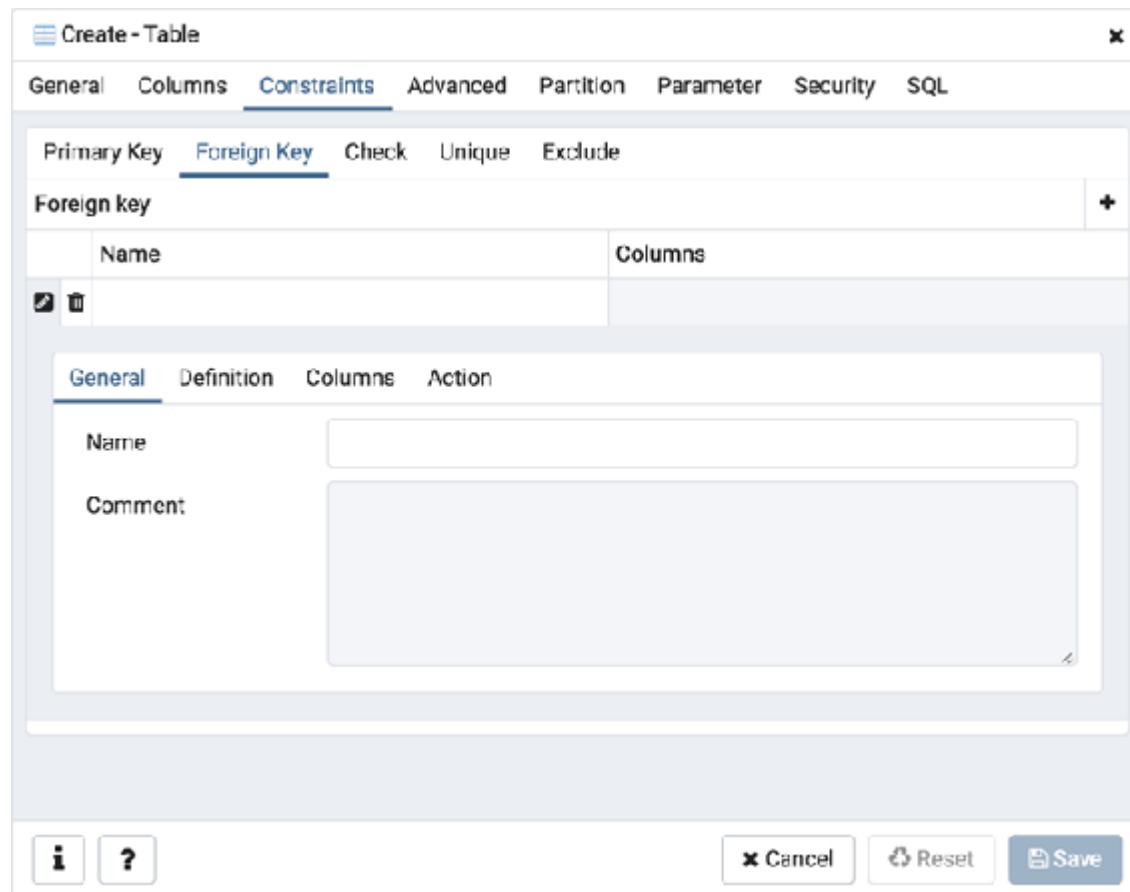
Use the *Fill Factor* field to specify a fill factor for the table and index.

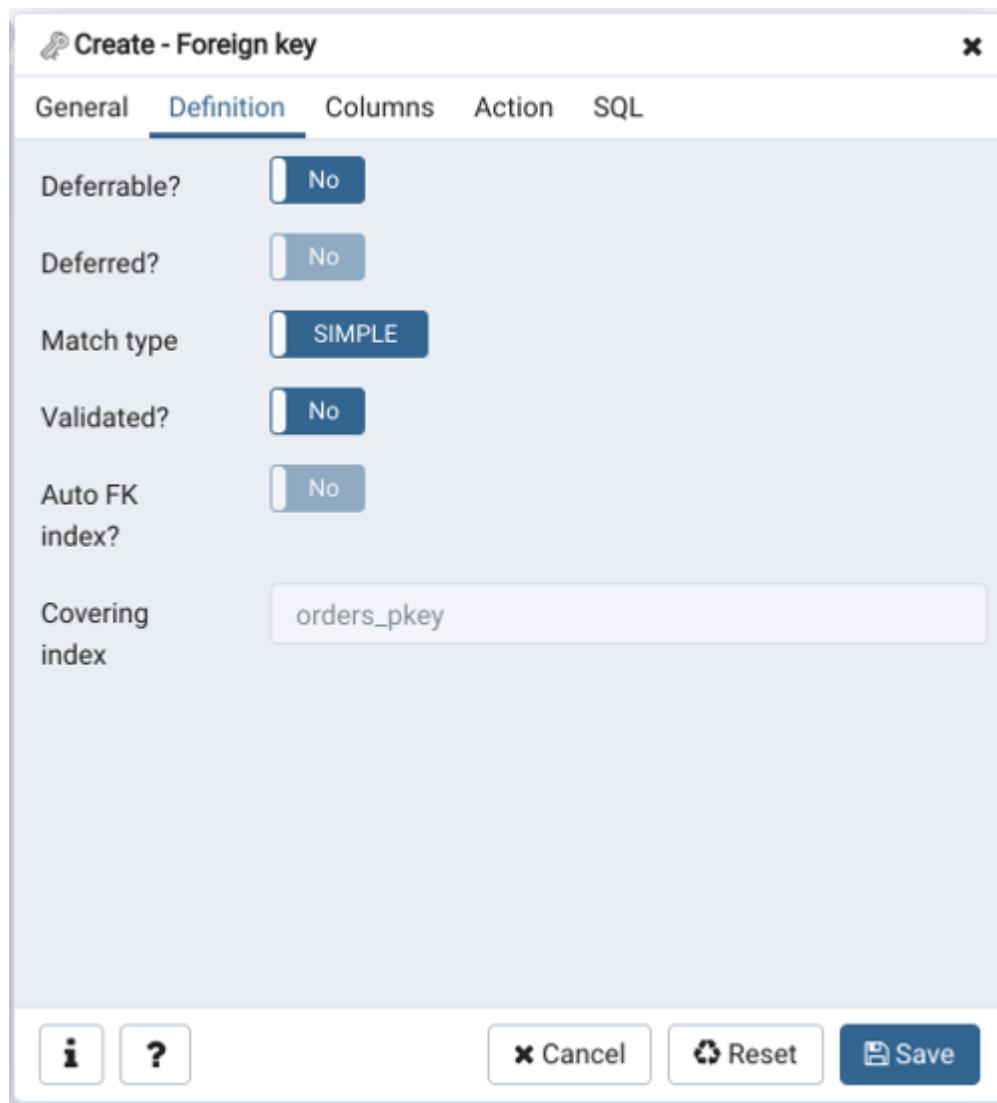
The fil factor for a table is a percentage between 10 and 100. 100 (complete packing) is the default.

Move the *Deferrable?* switch to the Yes position to specify the timing of the constraint is deferrable and can be postponed until the end of the statement. The default is No.

If enabled, move the *Deferred?* switch to the Yes position to specify the timing of the constraint is deferred to the end of the statement.

The default is No.





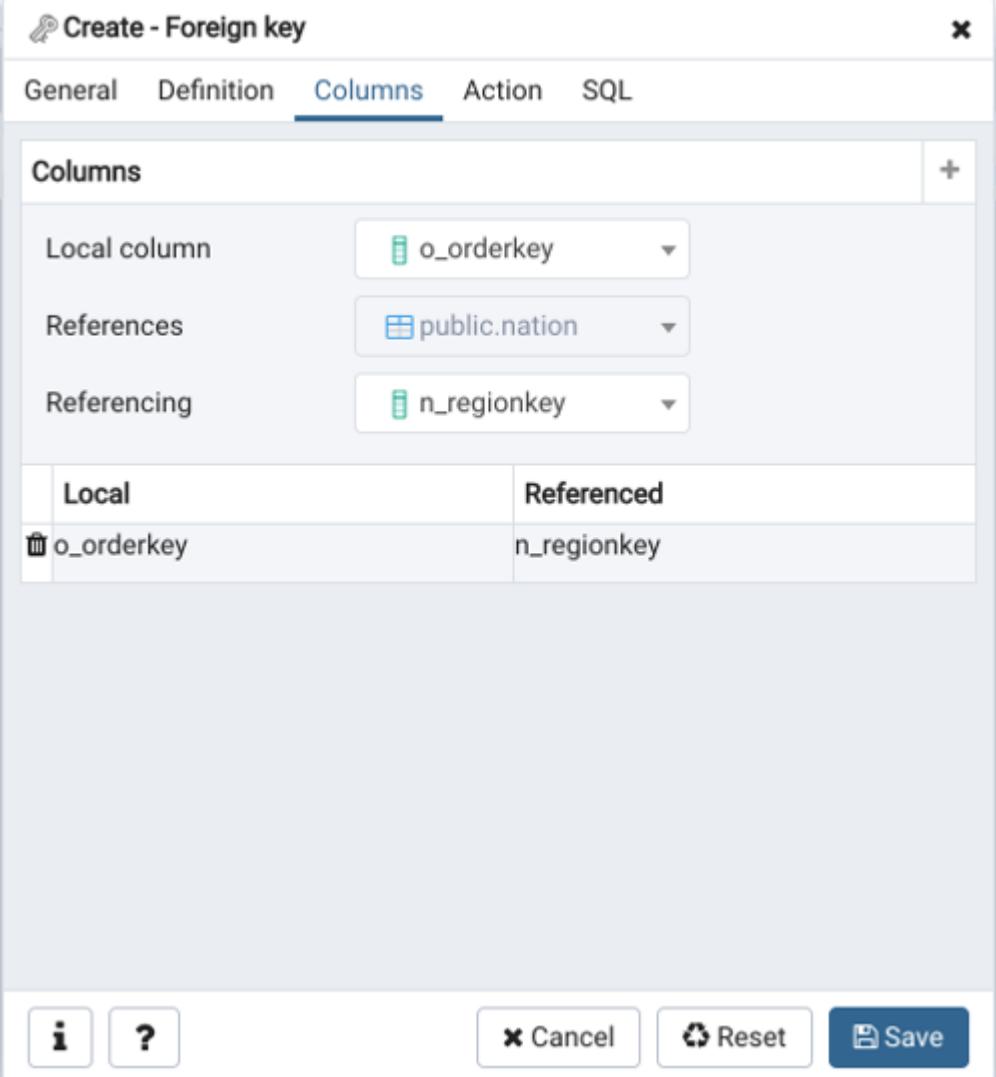
To add a foreign key constraint, select the *Foreign Key* tab, and click the *Add* icon (+). To define the constraint, click the *Edit* icon to the left of the *Trash* icon. A dialog similar to the *Foreign key* dialog (accessed by right clicking on *Constraints* in the *pgAdmin* tree control) opens.

Use the fields in the *General* tab to identify the foreign key constraint: Use the *Name* field to add a descriptive name for the foreign key constraint. The name will be displayed in the *pgAdmin* tree control.

Provide notes about the foreign key in the *Comment* field.

Click the *Definition* tab to continue.

Use the fields in the *Definition* tab to define the foreign key constraint:

 Create - Foreign key

General Definition **Columns** Action SQL

Columns

Local column	<input type="text" value="o_orderkey"/>
References	<input type="text" value="public.nation"/>
Referencing	<input type="text" value="n_regionkey"/>

Local	Referenced
<input type="checkbox"/> o_orderkey	n_regionkey

Cancel Reset Save

Move the *Deferrable?* switch to the *Yes* position to specify the timing of the constraint is deferrable and can be postponed until the end of the statement. The default is *No*.

If enabled, move the *Deferred?* switch to the *Yes* position to specify the timing of the constraint is deferred to the end of the statement.

The default is *No*.

Move the *Match type* switch specify the type of matching that is enforced by the constraint:

Select *Ful* to indicate that all columns of a multicolumn foreign key must be null if any column is null ; if all columns are null , the row is not required to have a match in the referenced table.

Select *Simple* to specify that a single foreign key column may be null ; if any column is null , the row is not required to have a match in the referenced table.

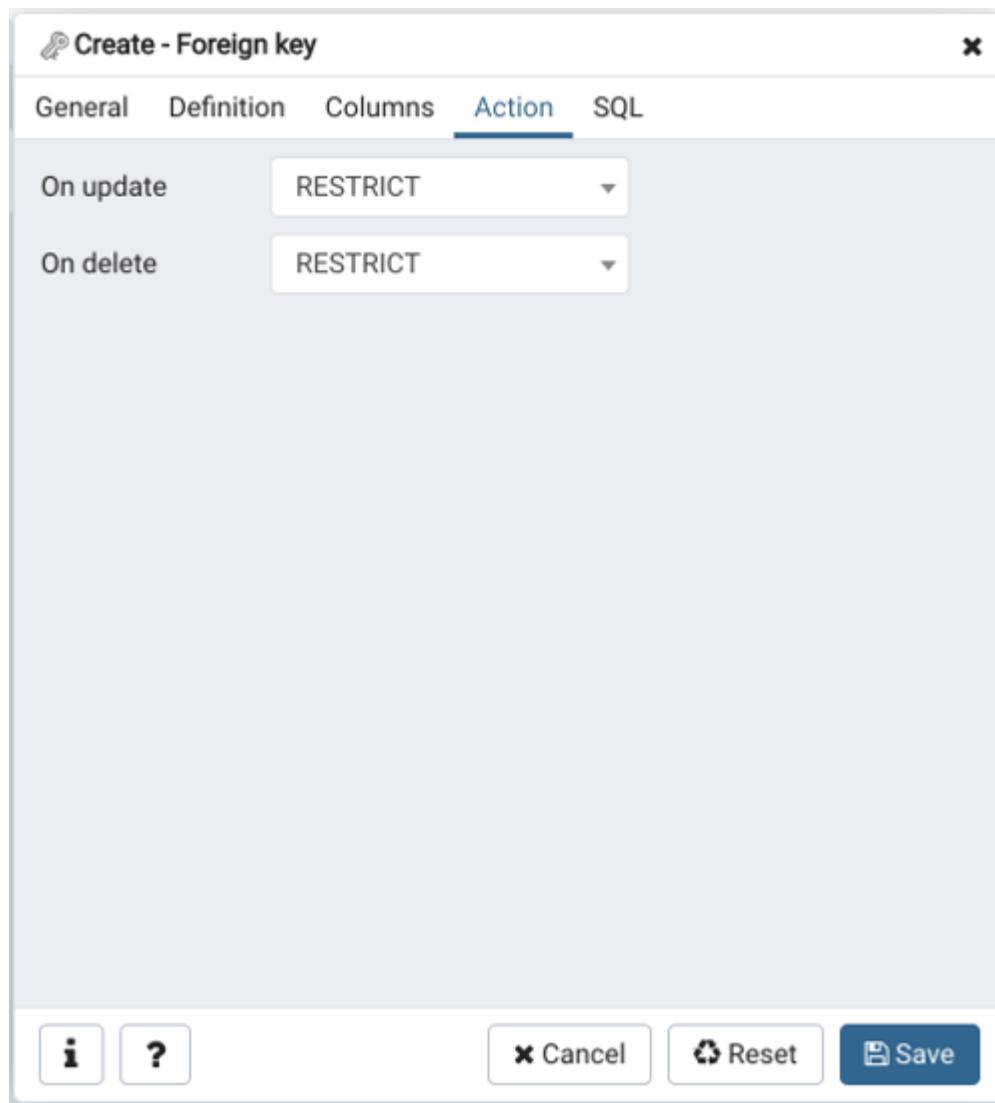
Move the *Validated* switch to the *Yes* position to instruct the server to validate the existing table content (against a foreign key or check constraint) when you save modifications to this dialog.

Move the *Auto FK Index* switch to the *No* position to disable the automatic index feature.

The field next to *Covering Index* generates the name of an index if the *Auto FK Index* switch is in the *Yes* position; or, this field is disabled.

Click the *Columns* tab to continue.

Use the fields in the *Columns* tab to specify one or more reference column(s).



A Foreign Key constraint requires that one or more columns of a table must only contain values that match values in the referenced column(s) of a row of a referenced table:

Use the drop-down listbox next to *Local column* to specify the column in the current table that will be compared to the foreign table.

Use the drop-down listbox next to *References* to specify the name of the table in which the comparison column(s) resides.

Use the drop-down listbox next to *Referencing* to specify a column in the foreign table.

Click the *Add* icon (+) to add a column to the list; repeat the steps above and click the *Add* icon (+) to add additional columns. To discard an entry, click the trash icon to the left of the entry and confirm deletion in the *Delete Row* popup.

Click the *Action* tab to continue.

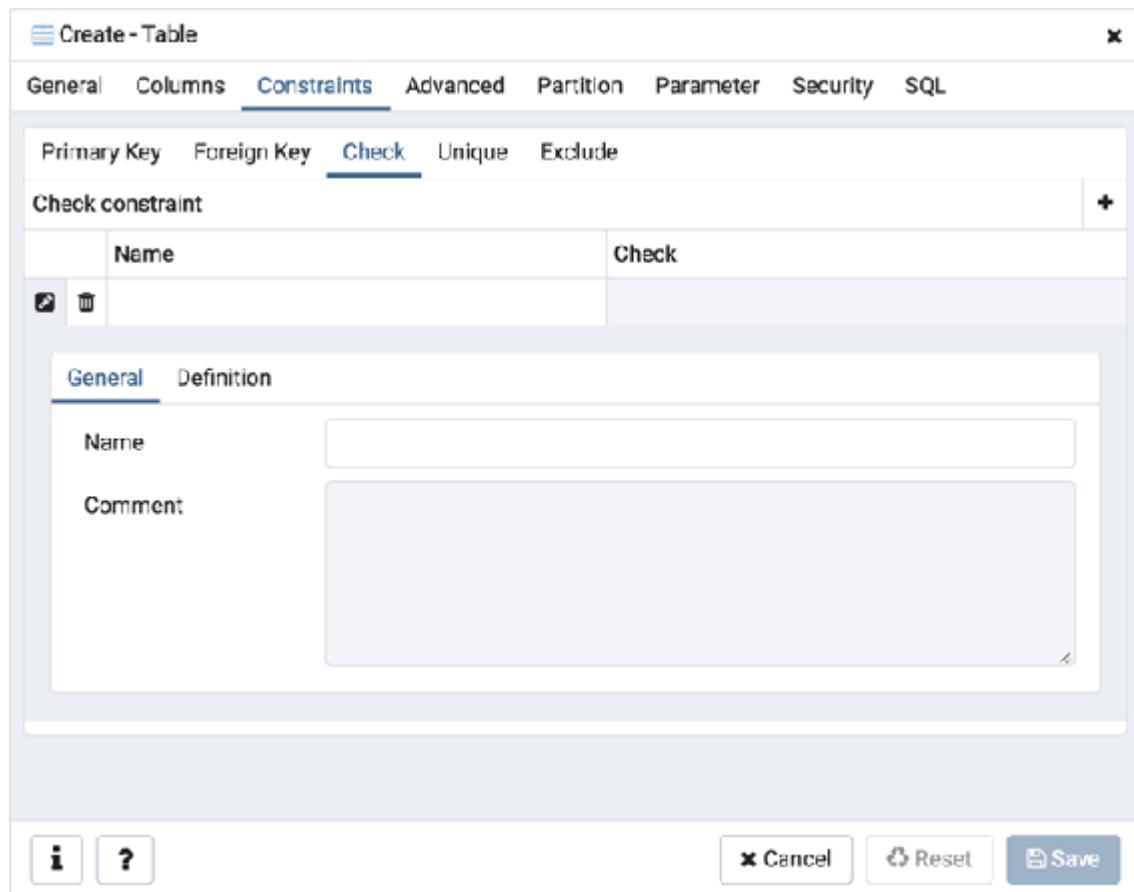
Use the drop-down listboxes on the *Action* tab to specify behavior related to the foreign key constraint that wil be performed when data within the table is updated or deleted:

Use the drop-down listbox next to *On update* to select an action that wil be performed when data in the table is updated.

Use the drop-down listbox next to *On delete* to select an action that wil be performed when data in the table is deleted.

The supported actions are:

NO ACTION Produce an error indicating that the deletion or update wil create a foreign key constraint violation. If the constraint is deferred, this error wil be produced at constraint check time if any referencing rows stil exist. This is the default.



RESTRICT

Throw an error indicating that the deletion or update would create a foreign key constraint violation. This is the same as NO ACTION except that the check is not

deferrable.

CASCADE

Delete any rows referencing the deleted row, or update the values of the referencing column(s) to the new values of the referenced columns, respectively.

SET NULL

Set the referencing column(s) to null.

SET

Set the referencing column(s) to their default values.

DEFAULT

There must be a row in the referenced table that matches the default values (if they are not null), or the operation will fail.

To add a check constraint, select the *Check* tab on the panel, and click the *Add* icon (+). To define the check constraint, click the *Edit* icon to the left of the *Trash* icon. A dialog similar to the *Check* dialog (accessed by right clicking on *Constraints* in the pgAdmin tree control) opens.

Use the fields in the *General* tab to identify the check constraint: Use the *Name* field to add a descriptive name for the check constraint. The name will be displayed in the pgAdmin tree control.

With PostgreSQL 9.5 forward, when a table has multiple check constraints, they will be tested for each row in alphabetical order by name and after NOT NULL constraints.

Provide notes about the check constraint in the *Comment* field.

Click the *Definition* tab to continue.

✓ Create - Check

X

General **Definition** SQL

Check

`o_orderkey > 500`

No Inherit?

No

Don't validate?

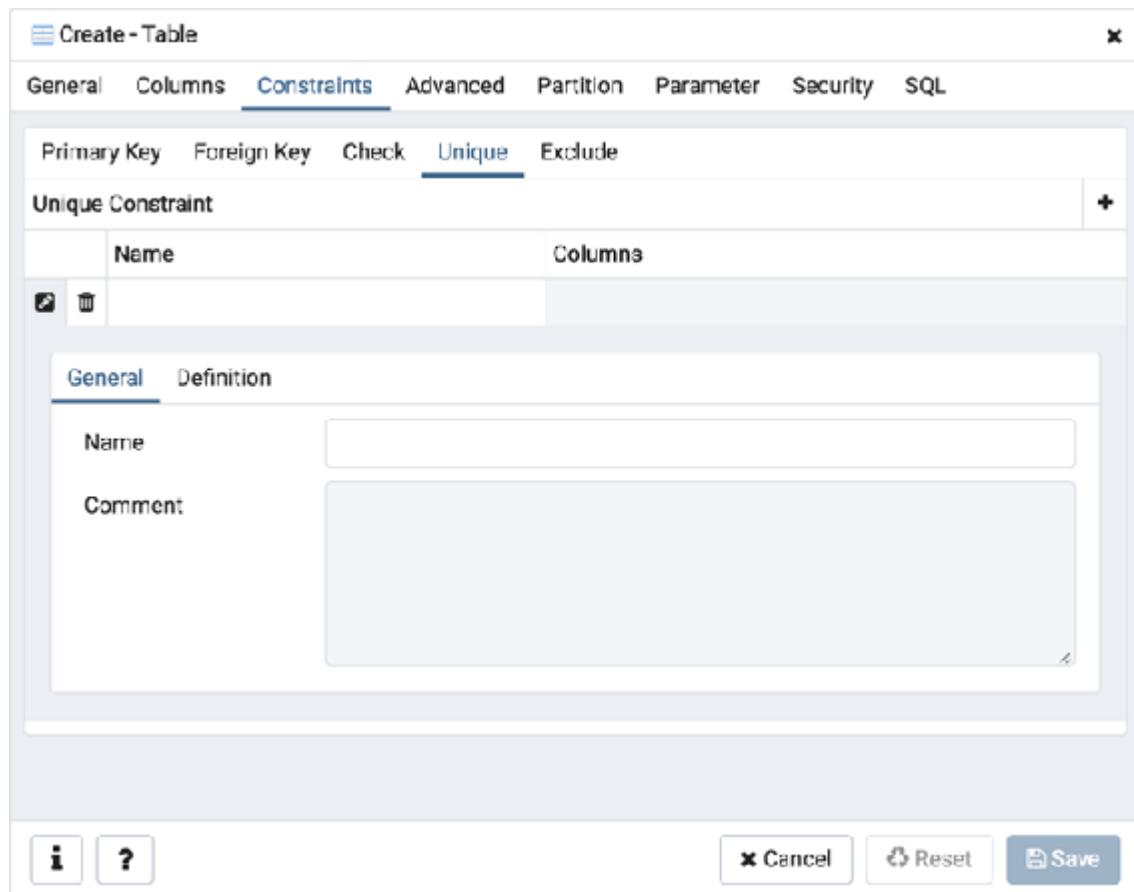
Yes



Cancel

Reset

Save



Use the fields in the *Definition* tab to define the check constraint: Provide the expression that a row must satisfy in the *Check* field. This field is required.

Move the *No Inherit?* switch to the *Yes* position to specify that this constraint is not automatically inherited by a table's children. The default is *No*, meaning that the constraint will be inherited by any children.

Move the *Don't validate?* switch to the *No* position to skip validation of existing data; the constraint may not hold for all rows in the table.

The default is *Yes*.

To add a unique constraint, select the *Unique* tab on the panel, and click the *Add* icon (+). To define the constraint, click the *Edit* icon to

the left of the *Trash* icon. A dialog similar to the *Unique constraint* dialog (accessed



by right clicking on *Constraints* in the *pgAdmin* tree control) opens.

Use the fields in the *General* tab to identify the unique constraint:
Use the *Name* field to add a descriptive name for the unique constraint. The name will be displayed in the *pgAdmin* tree control.

Provide notes about the unique constraint in the *Comment* field.

Click the *Definition* tab to continue.

Use the fields in the *Definition* tab to define the unique constraint: Click inside the *Columns* field and select one or more column names from the drop-down listbox. To delete a selection, click the x to the left of the column name. The unique constraint should be different from the primary key constraint defined for the same table; the selected column(s) for the constraints must be distinct.

Select the name of the tablespace in which the unique constraint will reside from the drop-down listbox in the *Tablespace* field.

Use the *Fil Factor* field to specify a fil factor for the table and index.

The fil factor for a table is a percentage between 10 and 100. 100 (complete packing) is the default.

Move the *Deferrable?* switch to the *Yes* position to specify the timing of the constraint is deferrable and can be postponed until the end of the statement. The default is *No*.

If enabled, move the *Deferred?* switch to the *Yes* position to specify the timing of the constraint is deferred to the end of the statement.

The default is *No*.

Create - Table

General Columns Constraints Advanced Partition Parameter Security SQL

Primary Key Foreign Key Check Unique **Exclude**

Exclude constraint

Name	Columns
<input checked="" type="checkbox"/> Test	

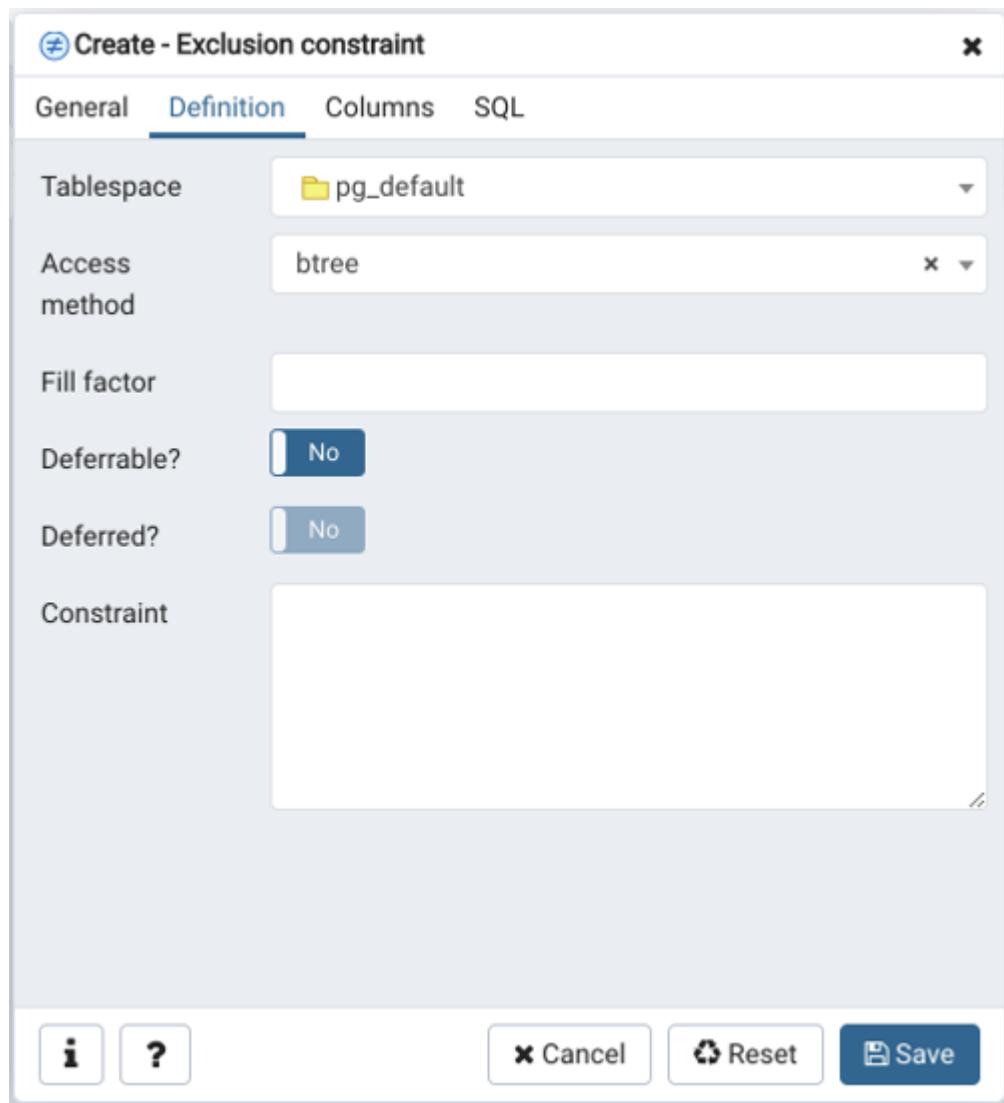
General Definition Columns

Name: Test

Comment:

Please specify columns for exclusion constraint.

i **?** **Cancel** **Reset** **Save**



To add an exclusion constraint, select the *Exclude* tab on the panel, and click the *Add* icon (+). To define the constraint, click the *Edit* icon to the left of the *Trash* icon. A dialog similar to the *Exclusion constraint* dialog (accessed by right clicking on *Constraints* in the *pgAdmin* tree control) opens.

Use the fields in the *General* tab to identify the exclusion constraint: Use the *Name* field to provide a descriptive name for the exclusion constraint. The name will be displayed in the *pgAdmin* tree control.

Provide notes about the exclusion constraint in the *Comment* field.

Click the *Definition* tab to continue.

Use the fields in the *Definition* tab to define the exclusion constraint:



Use the drop-down listbox next to *Tablespace* to select the tablespace in which the index associated with the exclude constraint will reside.

Use the drop-down listbox next to *Access method* to specify the type of index that will be used when implementing the exclusion constraint:

Select *gist* to specify a GiST index (the default).

Select *spgist* to specify a space-partitioned GiST index.

Select *btree* to specify a B-tree index.

Select *hash* to specify a hash index.

Use the *Fil Factor* field to specify a fil factor for the table and associated index. The fil factor is a percentage between 10 and 100.

100 (complete packing) is the default.

Move the *Deferrable?* switch to the *Yes* position to specify that the timing of the constraint is deferrable, and can be postponed until the end of the statement. The default is *No*.

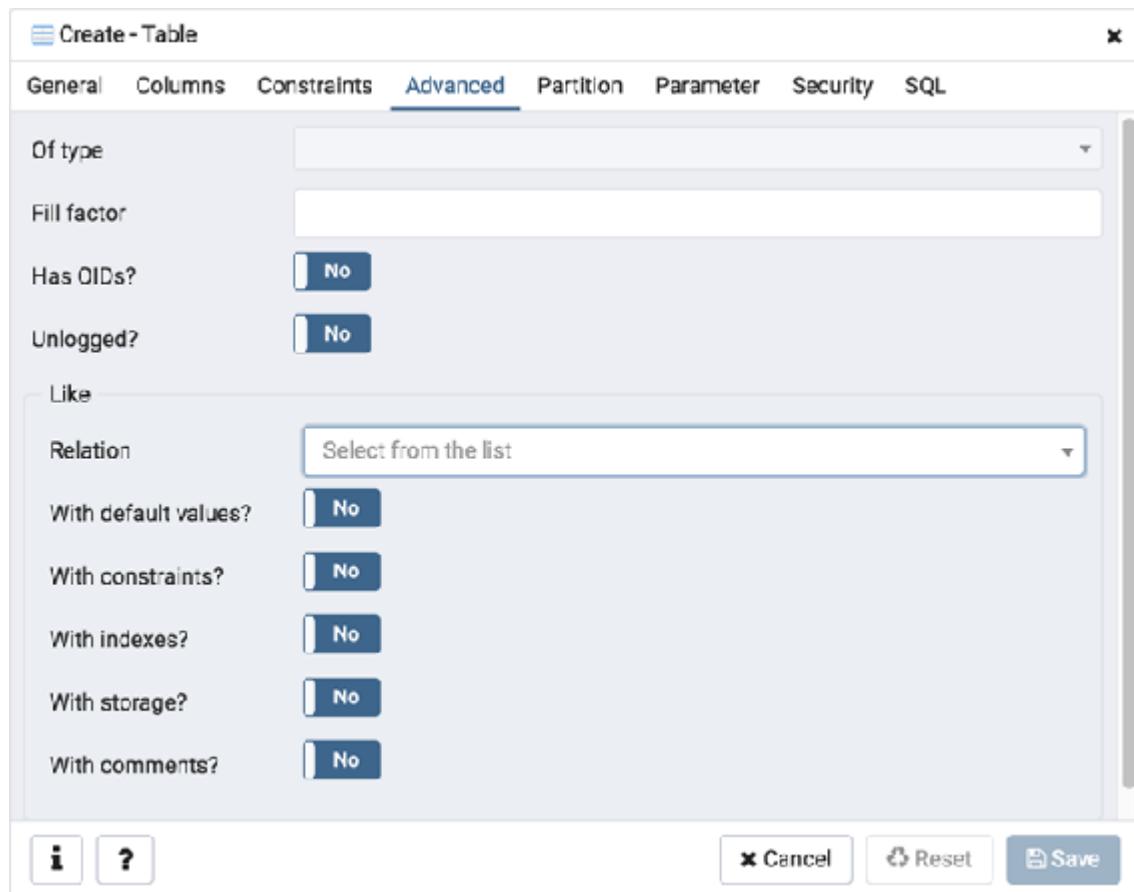
If enabled, move the *Deferred?* switch to the *Yes* position to specify the timing of the constraint is deferred to the end of the statement.

The default is *No*.

Use the *Constraint* field to provide a condition that a row must satisfy to be included in the table.

Click the *Columns* tab to continue.

Use the fields in the *Columns* tab to to specify the column(s) to which the constraint applies. Use the drop-down listbox next to *Column* to select a column and click the *Add* icon (+) to provide details of the action on the column:



The *Column* field is populated with the selection made in the *Column* drop-down listbox.

If applicable, use the drop-down listbox in the *Operator class* to specify the operator class that wil be used by the index for the column.

Move the *DESC* switch to *DESC* to specify a descending sort order.

The default is *ASC* which specifies an ascending sort order.

Move the *NULLs order* switch to *LAST* to define an ascending sort order for NULLs. The default is *FIRST* which specifies a descending order.

Use the drop-down list next to *Operator* to specify a comparison or conditional operator.

Click the *Advanced* tab to continue.

Use the fields in the *Advanced* tab to define advanced features for the table:

Use the drop-down listbox next to *Of type* to copy the table structure from the specified composite type. Please note that a typed table will be dropped if the type is dropped (with DROP TYPE ... CASCADE).

Use the *Fil Factor* field to specify a fil factor for the table. The fil factor for a table is a percentage between 10 and 100. 100 (complete packing) is the default.

Move the *Has OIDs?* switch to the *Yes* position to specify that each row within a table has a system-assigned object identifier. The default is *No*.

Move the *Unlogged?* switch to the *Yes* position to disable logging for the table. Data written to an unlogged table is not written to the write-ahead log. Any indexes created on an unlogged table are automatically unlogged as well. The default is *No*.

Create - Table

General Columns Constraints Advanced Partition Parameters Security SQL

Partition Type Range

Partition Keys

Key type	Column	Expression
Column	empid	

Partition table supports two types of keys:
Column: User can select any column from the list of available columns.
Expression: User can specify expression to create partition key.
Example: Let's say, we want to create a partition table based per year for the column 'saledate', having datatype 'date/timestamp', then we need to specify the expression as 'extract(YEAR from saledate)' as partition key.

Partitions

	Operation	Name	Default	From	To	In	Modulus	Remainder
<input checked="" type="checkbox"/>	<input type="button" value="Create"/>	part1	No	1	200			

General Partition

Partitioned table? Yes

Partition Type Range

Partition Keys

Key type	Column	Expression
Column	salary	

Buttons: i ? Cancel Reset Save

Use the fields in the Like box to specify which attributes of an existing table from which a table will automatically copy column names, data types, and not-null constraints; after saving the new or

modified table, any changes to the original table will not be applied to the new table.

Use the drop-down listbox next to *Relation* to select a reference table.

Move the *With default values?* switch to the Yes position to copy default values.

Move the *With constraints?* switch to the Yes position to copy table and column constraints.

Move the *With indexes?* switch to the Yes position to copy indexes.

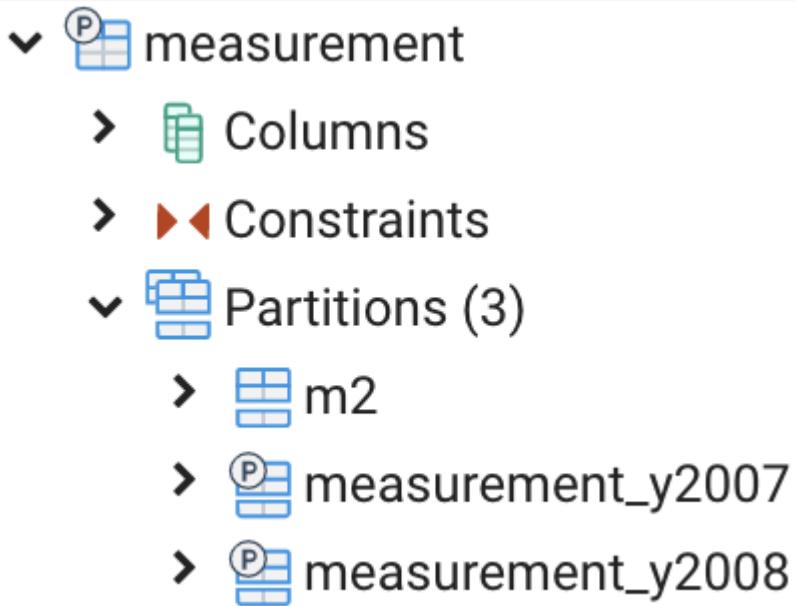
Move the *With storage?* switch to the Yes position to copy storage settings.

Move the *With comments?* switch to the Yes position to copy comments.

With PostgreSQL 10 forward, the *Partition* tab will be visible.

Click the *Partition* tab to continue.

Use the fields in the *partition* tab to create the partitions for the table:



Select a partition type from the *Partition Type* selection box. There are 3 options available; Range, List and Hash. Hash option will only enable for PostgreSQL version ≥ 11 .

Use the *Partition Keys* panel to define the partition keys. Click the Add icon (+) to add each partition keys selection:

Select a partition key type in the *Keytype* field.

Select a partition column in the *Column* field if Column option selected for *Keytype* field .

Specify the expression in the *Expression* field if Expression option selected for the *Keytype* field.

Use the *Partitions* panel to define the partitions of a table. Click the Add icon (+) to add each partition:

Move the *Operation* switch to *attach* to attach the partition, by default it is *create*.

Use the *Name* field to add the name of the partition.

If partition type is Range or List then *Default* field wil be enabled.

If partition type is Range then *From* and *To* fields wil be enabled.

If partition type is List then *In* field wil be enabled.

If partition type is Hash then *Modulus* and *Remainder* fields wil be enabled.

Users can create a partition and define them as a partitioned table. Click the *Edit* icon to expand the properties of a partition. Use the *Partition* tab to create that partition as a partitioned table.

Move the *Partitioned Table?* switch to the Yes in case you want to create a partitioned table.

Select a partition type from the *Partition Type* selection box.

Use the *Partition Keys* panel to define the partition keys.

View of multi level Partitioned Table in browser tree: Click the *Parameter* tab to continue.

Create - Table

General Columns Constraints Advanced Partition Parameter Security SQL

Table Toast Table

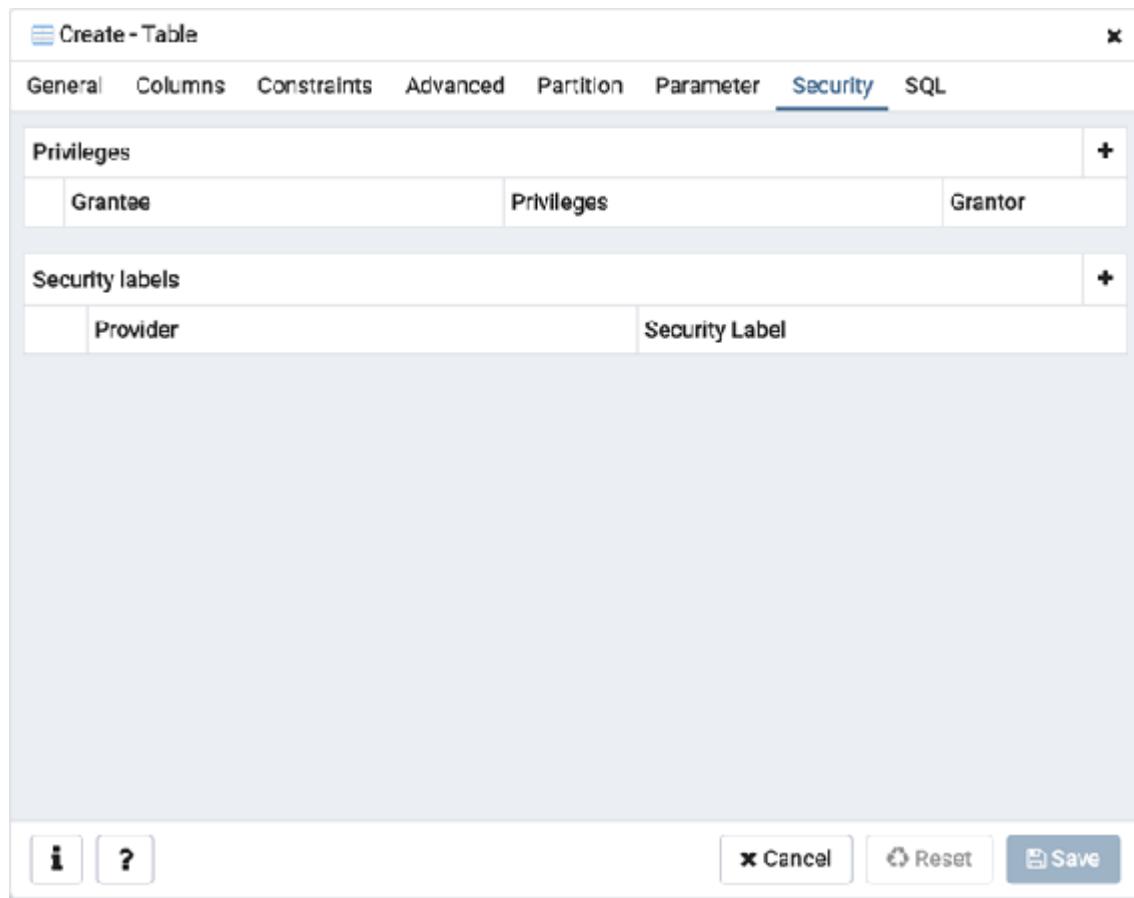
Custom auto-vacuum? No

Enabled? No

Vacuum Table

Label	Value	Default value
ANALYZE scale factor		0.10
ANALYZE base threshold		50
FREEZE maximum age		200,000,000
VACUUM cost delay		20
VACUUM cost limit		-1
VACUUM scale factor		0.20
VACUUM base threshold		50
FREEZE minimum age		50,000,000
FREEZE table age		150,000,000

i **?** **Cancel** **Reset** **Save**



Use the tabs nested inside the *Parameter* tab to specify VACUUM and ANALYZE thresholds; use the *Table* tab and the *Toast Table* tab to customize values for the table and the associated toast table: Move the *Custom auto-vacuum?* switch to the *Yes* position to perform custom maintenance on the table.

Move the *Enabled?* switch to the *Yes* position to select values in the *Vacuum table*. The *Vacuum Table* provides default values for maintenance operations.

Provide a custom value in the *Value* column for each metric listed in the *Label* column.

Click the *Security* tab to continue.

Use the *Security* tab to assign privileges and define security labels.

Use the *Privileges* panel to assign privileges to a role. Click the *Add* icon (+) to set privileges for database objects:

Select the name of the role from the drop-down listbox in the *Grantee* field.

Click inside the *Privileges* field. Check the boxes to the left of one or more privileges to grant the selected privilege to the specified user.

The current user, who is the default grantor for granting the privilege, is displayed in the *Grantor* field.

Click the *Add* icon (+) to assign additional privileges; to discard a privilege, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Use the *Security Labels* panel to define security labels applied to the function. Click the *Add* icon (+) to add each security label selection: Specify a security label provider in the *Provider* field. The named provider must be loaded and must consent to the proposed labeling operation.

Specify a security label in the *Security Label* field. The meaning of a given label is at the discretion of the label provider. PostgreSQL places no restrictions on whether or how a label provider must interpret security labels; it merely provides a mechanism for storing them.

Click the *Add* icon (+) to assign additional security labels; to discard a security label, click the trash icon to the left of the row and confirm deletion in the *Delete Row* popup.

Click the *SQL* tab to continue.

Your entries in the *Table* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example

The following is an example of the sql command generated by user selections in the *Table* dialog:



The screenshot shows the 'Create - Table' dialog box. The title bar says 'Create - Table'. Below it is a tab bar with 'General', 'Columns', 'Constraints', 'Advanced', 'Partition', 'Parameter', 'Security', and 'SQL'. The 'SQL' tab is selected. The main area contains the following SQL code:

```
1 CREATE TABLE public."pem.test"
2 (
3 )
4
5 WITH (
6     OTDS = FALSE
7 );
8
9 ALTER TABLE public."pem.test"
10    OWNER to enterprisedb;
```

At the bottom of the dialog are three buttons: 'Info' (with an 'i' icon), 'Reset' (with a circular arrow icon), and 'Save' (with a disk icon).

The example shown demonstrates creating a table named *product_category*. It has three columns and a primary key constraint on the *category_id* column.

Click the *Info* button (i) to access online help.

Click the *Save* button to save work.

Click the *Cancel* button to exit without saving work.

Click the *Reset* button to restore configuration parameters.