PostgreSQL Tutorial

An Overview

PostgreSQL is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

PostgreSQL is used as the **primary data store or data warehouse for many web, mobile, geospatial, and analytics applications**. The latest major version is PostgreSQL 12.

PostgreSQL has earned a strong reputation for its proven architecture, reliability, data integrity, robust feature set, extensibility, and the dedication of the open source community behind the software to consistently deliver performant and innovative solutions. PostgreSQL runs on all major operating systems, has been ACID-compliant since 2001, and has powerful add-ons such as the popular PostGIS geospatial database extender. It is no surprise that PostgreSQL has become the open source relational database of choice for many people and organisations.

Getting started with using PostgreSQL has never been easier - pick a project you want to build, and let PostgreSQL safely and robustly store your data.

Advantage of using PostgreSQL

PostgreSQL comes with many features aimed to help developers build applications, administrators to protect data integrity and build fault-tolerant environments, and help you manage your data no matter how big or small the dataset. In addition to being free and open source, PostgreSQL is highly extensible. For example, you can define your own data types, build out custom functions, even write code from different programming languages without recompiling your database!

PostgreSQL tries to conform with the SQL standard where such conformance does not contradict traditional features or could lead to poor architectural decisions. Many of the features required by the SQL standard are supported, though sometimes with slightly differing syntax or function. Further moves towards conformance can be expected over time. As of the version 14 release in September 2021, PostgreSQL conforms to at least 170 of the 179 mandatory features for SQL:2016 Core conformance. As of this writing, no relational database meets full conformance with this standard.

Below is an inexhaustive list of various features found in PostgreSQL, with more being added in every major release:

- **Data Types**
  - Primitives: Integer, Numeric, String, Boolean
  - Structured: Date/Time, Array, Range / Multirange, UUID
  - Document: JSON/JSONB, XML, Key-value (Hstore)
  - Geometry: Point, Line, Circle, Polygon
  - Customizations: Composite, Custom Types
- **Data Integrity**
  - UNIQUE, NOT NULL
  - Primary Keys
  - Foreign Keys
  - Exclusion Constraints
  - Explicit Locks, Advisory Locks
- **Concurrency, Performance**

    - Indexing: B-tree, Multicolumn, Expressions, Partial
    - Advanced Indexing: GiST, SP-Gist, KNN Gist, GIN, BRIN, Covering indexes, Bloom filters
    - Sophisticated query planner / optimizer, index-only scans, multicolumn statistics
    - Transactions, Nested Transactions (via savepoints)
    - Multi-Version concurrency Control (MVCC)
    - Parallelization of read queries and building B-tree indexes
    - Table partitioning
    - All transaction isolation levels defined in the SQL standard, including Serializable
    - Just-in-time (JIT) compilation of expressions
- **Reliability, Disaster Recovery**
    - Write-ahead Logging (WAL)
    - Replication: Asynchronous, Synchronous, Logical
    - Point-in-time-recovery (PITR), active standbys
    - Tablespaces
- **Security**
    - Authentication: GSSAPI, SSPI, LDAP, SCRAM-SHA-256, Certificate, and more
    - Robust access-control system
    - Column and row-level security
    - Multi-factor authentication with certificates and an additional method
- **Extensibility**
    - Stored functions and procedures
    - Procedural Languages: PL/PGSQL, Perl, Python (and many more)
    - SQL/JSON path expressions
    - Foreign data wrappers: connect to other databases or streams with a standard SQL interface
    - Customizable storage interface for tables
    - Many extensions that provide additional functionality, including PostGIS
- **Internationalisation, Text Search**
    - Support for international character sets, e.g. through ICU collations
    - Case-insensitive and accent-insensitive collations
    - Full-text search

There are many more features that you can discover in the PostgreSQL documentation. Additionally, PostgreSQL is highly extensible: many features, such as indexes, have defined APIs so that you can build out with PostgreSQL to solve your challenges.

PostgreSQL has been proven to be highly scalable both in the sheer quantity of data it can manage and in the number of concurrent users it can accommodate. There are active PostgreSQL clusters in production environments that manage many terabytes of data, and specialized systems that manage petabytes.

Getting Started on Windows

There are three steps to complete the PostgreSQL installation:

1. Download PostgreSQL installer for Windows
2. Install PostgreSQL

3. Verify the installation

1. Downloading PostgreSQL Installer for Windows

First, you need to go to the download page of PostgreSQL installers on the EnterpriseDB. > go for **Windows x86-64**, version **12.3**. It will take a few minutes to complete the download.

2. Install PostgreSQL on Window step by step

To install PostgreSQL on Windows, you need to have administrator privileges.

Step 1. Double click on the installer file, an installation wizard will appear and guide you through multiple steps where you can choose different options that you would like to have in PostgreSQL.

Step 2. Click the Next button.

Step 3. Specify installation folder, choose your own or keep the default folder suggested by PostgreSQL installer and click the Next button

Step 4. Select software components to install:

- The PostgreSQL Server to install the PostgreSQL database server
- pgAdmin 4 to install the PostgreSQL database GUI management tool.
- Command Line Tools to install command-line tools such as psql, pg_restore, etc. These tools allow you to interact with the PostgreSQL database server using the command-line interface.
- Stack Builder provides a GUI that allows you to download and install drivers that work with PostgreSQL.

Step 5. Select the database directory to store the data or accept the default folder. And click the Next button to go to the next step

Step 6. Enter the password for the database superuser (postgres)

Note: PostgreSQL runs as a service in the background under a service account named `postgres`. If you already created a service account with the name `postgres`, you need to provide the password of that account in the following window.

After entering the password, you need to retype it to confirm and click the Next button:

Step 7. Enter a port number on which the PostgreSQL database server will listen. The default port of PostgreSQL is 5432. You need to make sure that no other applications are using this port.

Step 8. Choose the default locale used by the PostgreSQL database. If you leave it as default locale, PostgreSQL will use the operating system locale. After that click the Next button.

Step 9. The setup wizard will show the summary information of PostgreSQL. You need to review it and click the Next button if everything is correct. Otherwise, you need to click the Back button to change the configuration accordingly.

Now, you're ready to install PostgreSQL on your computer. Click the **Next** button to begin installing PostgreSQL.

Step 10. Click the **Finish** button to complete the PostgreSQL installation.

3. Verify the Installation

There are several ways to verify the PostgreSQL installation. You can try to connect to the PostgreSQL database server from any client application e.g., psql and pgAdmin. The quick way to verify the installation is through the psql program.

1. Click the `psql` application to launch it. The psql command-line program will display.
2. Enter all the necessary information such as the server, database, port, username, and password. To accept the default, you can press **Enter**. Note that you should provide the password that you entered during installing the PostgreSQL.
3. Issue the command `SELECT version();`

Connect To a PostgreSQL Database Server

When you installed the PostgreSQL database server, the PostgreSQL installer also installed some useful tools for working with the PostgreSQL database server. In this tutorial, you will learn how to connect to the PostgreSQL database server via the following tools:

- psql – a terminal-based front-end to PostgreSQL database server.
- pgAdmin – a web-based front-end to PostgreSQL database server.

1. Connect to PostgreSQL database server using psql

psql is an interactive terminal program provided by PostgreSQL. It allows you to interact with the PostgreSQL database server such as executing SQL statements and managing database objects.

The following steps show you how to connect to the PostgreSQL database server via the *psql* program:

1. Launch the **psql** program and connect to the PostgreSQL Database Server using the **postgres** user.
2. Enter all the information such as Server, Database, Port, Username, and Password. If you press Enter, the program will use the default value specified in the square bracket `[]` and move the cursor to the new line.
3. Interact with the PostgreSQL Database Server by issuing an SQL statement. Please do not forget to end the statement with a semicolon (;). After pressing **Enter**, psql will return the current PostgreSQL version on your system.

2. Connect to PostgreSQL database server using pgAdmin

The second way to connect to a database is by using a pgAdmin application. The pgAdmin application allows you to interact with the PostgreSQL database server via an intuitive user interface.

The following illustrates how to connect to a database using pgAdmin GUI application:

1. Launch the pgAdmin application.
2. Rght-click the Servers node and select **Create > Server…** menu to create a server.
3. Enter the server name e.g., **PostgreSQL** and click the **Connection** tab.
4. Enter the host and password for the **postgres** user and click the **Save** button.
5. Click on the Servers node to expand the server. By default, PostgreSQL has a database named postgres
6. Open the query tool by choosing the menu item **Tool > Query Tool** or click the lightning icon.
7. Enter the query in the **Query Editor**, click the **Execute** button, you will see the result of the query displaying in the **Data Output** tab.

Downloading PostgreSQL installer for macOS

To download the PostgreSQL installer, you follow these steps:

- First, visit the PostgreSQL installer download page.
- Then, download the PostgreSQL for macOS.

Install PostgreSQL on macOS

To install PostgreSQL on macOS, you follow these steps:

1. Launch the setup wizard by double-click the installer file
2. Select the directory where the PostgreSQL will be installed and click the Next button
3. Select the components that you want to install, uncheck the Stack Builder, and click the **Next** button
4. Specify a directory where PostgreSQL stores the data and click the Next button
5. Enter the password for the **postgres** user account. You should note down this password for logging in to the PostgreSQL database server later. After that, click the Next button.
6. Specify the port number on which the PostgreSQL server will listen. By default, PostgreSQL uses port number 5432.
7. Select the locale used by PostgreSQL. By default, PostgreSQL uses the locale of the current operating system.
8. Review the installation information. If everything looks correct, click the Next button to begin the installation.
9. Click the Next button to start installing the PostgreSQL database server on your computer. It will take few minutes to complete the installation.
10. Finally, click the **Finish** button once the installation is completed.

Load the sample database

1. Launch pgAdmin from Launchpad.
2. Enter the password for the **postgres** user.
3. Right-click the PostgreSQL 12 and select **Create > Database..** to open a dialog for creating a new database.
4. Enter dvdrental as the database, postgres as the owner, and click the Save button to create the dvdrental database.
5. Download the sample database and unzip it. You'll get a directory with many files.
6. Right-click the **dvdrental** database and select the **Restore** menu item.
7. Select the directory as the Format (1), the directory that contains sample database as the Filename (2), and postgres as the Role name (3), and click the Restore button.