

## Complete Learning Resources for PostgreSQL with Python in 2021

Start your PostgreSQL Journey from Here

*SQL, Lisp, and Haskell are the only programming languages that I've seen where one spends more time thinking than typing.*

*Philip Greenspun*

In this blog, I have discussed PostgreSQL RDBMS under a few headlines like Why should you learn SQL, why is PostgreSQL the best available option, A brief history of PostgreSQL, Best Resource for learning, and Other Resources. In the end, I also shared my timeline for learning PostgreSQL. There I have written about the best learning resource about using PostgreSQL databases with the Psycopg2 library of Python.

### Why should you learn Structured Query Language (SQL)?

SQL was initially created to enable non-IT specialists to query information from datasets without needing to ask a programmer for help. SQL aimed to be much more powerful than graphical user interfaces and more flexible than pre-built reports and queries. The objective of creating SQL was to enable new ways of examining and querying existing data from the database.

SQL is a declarative language. In declarative languages like SQL, the user tells the database which data is required and the database decides how to perform the command. The importance is on the result of the process rather than the process itself. The success and efficiency of SQL rely upon users asking the correct questions.

Besides being able to use SQL to work with databases, learning it also encourages us to build powerful mental models to think about data. By using SQL, we can gather and store more and more data, and then we will be capable of reasoning that data using a powerful mental ability.

### Why is PostgreSQL the best available option?

PostgreSQL is available for free and it is open-source. It will never be bought. This makes it the best tool for learning about relational databases. PostgreSQL has very impressive and detailed official documentation.

Though the NoSQL database systems' hype is all over the world, relational databases are still the workhorses of the technology industry. PostgreSQL has an advanced implementation of SQL and is very close to the SQL standard.

PostgreSQL provides all services that we expect a relational database system to do. Data is stored in user-created tables, with each entry being a row of a specific table. Columns

on a table correspond to various attributes. As it is a relational database system, tables can be related to each other using foreign keys to create complex data structures. What makes PostgreSQL interesting is the number of additional features it offers.

PostgreSQL is not just a relational database, it is an object-relational database. This means that it supports object-oriented programming principles such as inheritance, polymorphism, abstraction, and encapsulation. These features really make PostgreSQL exceptional than other relational databases.

PostgreSQL provides an expanded set of data types. All RDBMSs support several varieties of string, numeric, floating-point, Boolean, date, and time types. PostgreSQL also provides:

- UUID, Universally Unique IDs, which are essential for distributed database systems as they are unique universally that means in distributed database systems, we don't have to worry about multiple records having the same id.
- Monetary, a fixed-decimal type that eliminates the rounding and calculation errors found in floating-point numbers.
- Enumerated, a static set of options.
- Geometric types: point, box, line segment, line, path, polygon, and circle. They are useful for mathematics and geometric related applications.
- Binary, similar to SQL's blob object.
- Network addresses, PostgreSQL provides useful network addresses according to several protocols.
- Text vectors which are useful for full-text search functions.
- Data representation types like XML, JSON, and arrays.
- composite types, PostgreSQL supports composite data types which are created by grouping several other types together.
- Range types, including numerical ranges, date ranges, and time range.
- User-defined types, User can define custom data types in PostgreSQL following some rules.

Besides, PostgreSQL can handle enormous amounts of data. The row size limit is 1.6 TB, and a single field can hold 1 GB of data.

When we run an SQL query in PostgreSQL against a set of database tables, the result is another table. It is called the Virtual Table. It is a database feature that allows us to create result tables and then run additional queries against them. This allows for more complex querying and opportunities for read-performance enhancements.

In addition to creating new neural connections in your brain, learning PostgreSQL could do wonders for your career prospects as it will make you a desirable candidate. Having

“PostgreSQL knowledge” in your CV and online portfolio will certainly attract the attention of recruiters, as the role of Data Scientist has been called, by the Harvard Business Review, “The Sexiest Job of the 21st Century”. And as Data Scientists work with data, knowing about databases management systems is a must skill for those who want to move in that direction.

### A brief history of PostgreSQL

In 1973, two computer scientists at UC Berkeley, Eugene Wong, and Michael Stonebraker began working on a database system they called Ingres (INteractive GReaphics REtreival System). The work was originally intended for use by the University's economists, but Stonebraker and Wong expanded the project to work on a relational database system, inspired by the concepts in the new Structured Query Language.

In 1982, Stonebraker left Berkeley to work on a proprietary relational database system, which he called POSTGRES (Post-Ingres). POSTGRES used many ideas from Ingres but was not a fork of its codebase. Stonebraker was intent on adding explicit support for data types and relationships into the database.

The prototype for the new system was demonstrated in 1988, and the Version 1 release came in 1989. Version 2 came the next year and Version 3 the next year, each time with re-written query rules. By 1993, there were so many users that the POSTGRES team could not keep up with support requests. In 1994, the team released Version 4 and ended the project. They released all of the work under an open (MIT-style) license, allowing other developers to take over the project.

That same year, two UC Berkeley students replaced POSTGRES's query language with SQL and released their new version (Postgres95) on the web. In 1996, Marc Fournier of Hub.org provided a development server and Postgres became an open-source development project. It was renamed PostgreSQL (post-gres-cue-ell). The development has continued since then by a large community of volunteers and developers known as The PostgreSQL Global Development Group.

### Best resource

The first stop is the online docs of PostgreSQL. The official documentation of any product is the best place to find the largest range of information. For most people nowadays, official docs and manuals are typically the last places to look for help. It should, however, always be the first stop on the list for various reasons such as:

- Official docs explain the internals of various components of a product and how they relate to each other.

- They link to various other sections of manuals discussing a concept when a new concept is introduced.
- There is sample code to be executed and its expected output with an explanation
- There is a logical flow from one idea to another.
- There is a “Tip” and “Quick Setup” section wherever required that gives bonus information for newbies.
- The manuals are divided into appropriate sections as per the need such as developer-oriented, administrator-related, programming focused, utilities, command reference, internals, and appendices, etc.

### Other resources

Besides the official documentation of the PostgreSQL website, I highly recommend you to go through the resources below. They are super helpful and beginner-friendly.

- PostgreSQL for Absolute Beginners, a GitHub repository to learn the basics of Postgres and the Psycpg2 library of Python which works with Postgres.
- PostgreSQL Tutorial, an entire website dedicated to Postgres tutorials.
- PostgreSQL Tutorial a very simple and straightforward introduction from Tutorials Point.
- Learn PostgreSQL Tutorial - Full Course for Beginners, 4-hour long freeCodeCamp’s full course on PostgreSQL for beginners.
- Beginner's Guide to PostgreSQL, a paid video-based tutorial at Udemy.
- Fashion Is Hard; PostgreSQL Is Easy.
- AmigosCode, A YouTube channel that provides useful video content related to PostgreSQL.
- PostgreSQL Wiki, the community-driven source for all the documentation that isn't in the official documentation — an invaluable source of practical information.
- PostgreSQL command line cheat sheet, in case you need a quick reminder of PostgreSQL's PSQL commands.
- Psycpg2 official documentation, learn about working with PostgreSQL in Python.
- Learning PostgreSQL 11: A beginner's guide to building high-performance PostgreSQL database (book by Salahaldin Juba & Andrey Volkov)

Now that you have started learning and exploring the possibilities of PostgreSQL, it’s time to meet some real people in person. One way of achieving that would be to attend events and technical symposia organized. These events are generally run anywhere from a few hours a day to one full week of activities revolving around PostgreSQL development, PostgreSQL hacks, boot camps, and workshops, etc.

There are plenty of conferences held all year round across the globe such as listed below:

- PGDay
- Postgres Vision
- PGConf
- PostgreSQL Developer Day
- FOSDEM PGDay
- PG Down Under
- PostgresConf
- PostgresOpen
- FOSSASIA PostgreSQL Summit
- PGCon

The sponsored conferences listed above are held at various geographical locations and they are named after the region being conducted at, such as PGDay UK, PGConf Asia, PGConf EU, and so on (Please note that some of them are only held in the region's local language).

The most important conference is PGCon. This is an annual conference for users and developers of PostgreSQL held during the last week of May every year at the University of Ottawa in Ottawa City, Canada. This is where the top developers and committers of PostgreSQL meet each year to discuss enhancements, new features, and the development activities of PostgreSQL (in addition to presenting and conducting training boot camps). During this event, the community recognized developers and committers that have contributed immensely to PostgreSQL. Some are also formally inducted into the panel of contributors. The boot camps and training conducted during PGCon are handled by industry experts who have developed the core features of PostgreSQL, which means you get to know the internals of PostgreSQL from the people who designed it. While a good reason to attend the community events is so you can expand your network in the technology industry.

## My Timeline

Now let me share my timeline for learning PostgreSQL very quickly. But before that, I want to clarify that I am not an expert at PostgreSQL. Though I know the basics and intermediate topics of PostgreSQL, I want to study more about it and grasp the advanced topics as well. So, you can say that I am just a learner.

I started learning PostgreSQL 3 months earlier. First, I searched for the full course on PostgreSQL and found that freeCodeCamp has already created a 4-hour long video tutorial. So, I watched the full tutorial in 7 days. That tutorial kickstarted my PostgreSQL journey. Then I read the official docs of PostgreSQL and also go through the "PostgreSQL Tutorial" website. The text-based tutorials were so helpful for me. At the same time, I

created a GitHub repository named “PostgreSQL for Absolute Beginner” where I pushed my every knowledge which I have gathered from the previous resources. So, you should check that repository and show some ♥ by giving it a star.

After learning a new topic, I always try to connect my previous knowledge with that topic. It helps me to understand the topic deeply and to apply my knowledge across my skill-set. So, after learning the basics and intermediate topics of PostgreSQL, I thought it would be nice if I could connect the database with Python (my favorite language) and use PSQL commands inside my Python code so that I could instantly implement the PostgreSQL knowledge into my projects. For that reason, I learned about the Psycopg2 library of Python by reading their official documentation. And the documentations were amazing.

Finally, I started using PostgreSQL and implementing it in a few of my personal projects. And if I forget any topics of Postgres I don't hesitate to look for that specific topic in my GitHub repo or the PostgreSQL command line cheat sheet. Because every coder can forget some of their previous learnings and there is no shame in looking for help in Google or asking other developers to figure out the problem and solve it. And it is the best way to learn a topic and connect with people.