

PostgreSQL for Jobseekers

*Introduction to PostgreSQL
administration for modern DBAs*

Sonia Valeja
David Gonzalez



www.bpbonline.com

Copyright © 2023 BPB Online

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor BPB Online or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

BPB Online has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, BPB Online cannot guarantee the accuracy of this information.

First published: 2023

Published by BPB Online

WeWork

119 Marylebone Road

London NW1 5PU

UK | UAE | INDIA | SINGAPORE

ISBN 978-93-55514-004

www.bpbonline.com

Dedicated to

*My supportive parents – Rupa & Hasanand and
my loving husband – Sunil*

– Sonia Valeja

My loved Sofi, the reason I do all this.

– David Gonzalez

About the Author

- **Sonia Valeja:** Sonia is an Indian DBA who started working in PostgreSQL in the year 2012. Since she has performed multiple migrations from Oracle to PostgreSQL, she learnt Oracle while working with PostgreSQL. She is a SME for beginner-level course on Open Source Database-PostgreSQL in one of her company. She has conducted multiple trainings on PostgreSQL for freshers as well as experienced professionals. Also, she has been an active participant at pgConf and has given lightning talk. She has worked extensively in Taxation as well as manufacturing based projects for Indian as well as African sub-continent.
- **David Gonzalez:** David is a Mexican DBA with 10+ years of hands-on experience with different RDBMS such as PostgreSQL, Oracle, and MySQL. He is good at PostgreSQL Administration and Development and has written multiple blogs on PostgreSQL. He has participated in various projects for banking, IT, and Software companies.

About the Reviewer

Amul Sul is a passionate developer with extensive professional experience in the domain of core database development.

He is an active contributor in PostgreSQL feature development as well as in EDB Postgres Advanced Server propriety feature development.

He is currently working in EnterpriseDB and is part of the database developer team. Prior to EDB, he worked at NTT Data. Amul holds a Master's degree from Mumbai University.

Acknowledgements

I want to express my deepest gratitude to my family and friends for their unwavering support and encouragement throughout this book's writing, especially my parents Rupa & Hasanand and my better half Sunil.

— *Sonia Valeja*

I want to say thank you from the bottom of my heart to my family, my wife Pam, and my beautiful daughter Sofi, for all the support and patience during this book process. They were always there cheering me up and pushing me to reach the goal.

— *David Gonzalez*

We are also grateful to BPB Publications for their guidance and expertise in bringing this book to fruition. It was a long journey of revising this book, with valuable participation and collaboration of reviewers, technical experts, and editors.

We would also like to acknowledge the valuable contributions of my colleagues and co-worker during many years working in the tech industry, who have taught me so much and provided valuable feedback on my work.

Finally, we would like to thank all the readers who have taken an interest in our book and for their support in making it a reality. Your encouragement has been invaluable.

Preface

Welcome to the world of PostgreSQL! Whether you are a novice exploring the realm of databases or a seasoned professional looking to expand your knowledge, this book is designed to be your comprehensive guide to everything PostgreSQL offers. We have carefully crafted this book to cover a wide range of topics, from the history and installation of PostgreSQL to its intricate internals, advanced tuning techniques, and securing your environment. We also delve into data replication, backup and restore operations, and the process of contributing to the vibrant PostgreSQL community.

In the first section of this book, we embark on a journey through time to explore the rich history of the PostgreSQL project. From its humble beginnings as a research project at the University of California, Berkeley, to becoming one of the most powerful and popular open-source databases, you will gain insight into the evolution and key milestones that have shaped PostgreSQL into what it is today. Understanding its roots will help you appreciate the philosophy and ethos behind this remarkable database system.

The subsequent chapters focus on the practical aspects of PostgreSQL, starting with the installation process on modern operating systems. We will guide you through each step, providing clear instructions and best practices to ensure a smooth setup. Additionally, we delve into alternative deployment options, such as Docker and Cloud, offering flexibility in how you harness the power of PostgreSQL in your own environment.

Once you have PostgreSQL up and running, we dive deep into its internals. Understanding how PostgreSQL processes, stores, and retrieves data is crucial for effective performance tuning and troubleshooting. We unravel the inner workings of the query optimizer, storage engine, and transaction management system, equipping you with the knowledge and tools to optimize your database for optimal performance. Whether dealing with a sluggish query or facing scalability challenges, this section will be your indispensable resource.

We also explore the essential backup and restore operations practices, enabling you to protect your data against unforeseen disasters. Later, we shed light on implementing data replication, which not only ensures high availability but also opens doors to advanced use cases like distributed architectures and load

balancing. Moreover, we delve into securing the access and integrity of your data, covering topics such as authentication and authorization.

Finally, this book wouldn't be complete without acknowledging the vibrant and collaborative PostgreSQL community. We will learn about some of the most used open-source projects, in the form of extensions and compatible tools that can enhance the native PostgreSQL capabilities. We share valuable insights on how you can contribute to this remarkable community through code contributions, documentation, or engaging in user forums. By becoming an active member, you not only enhance your knowledge and skills but also play a role in shaping the future of PostgreSQL.

Now, fasten your seatbelt and get ready for an exhilarating journey into the world of PostgreSQL. This book is your trusty companion, empowering you to master every facet of this powerful database system. Let's dive in and unlock the true potential of PostgreSQL together!

Chapter 1: Introduction to Opensource Database - PostgreSQL - introduces the open-source paradigm and covers the main general aspects of PostgreSQL like PostgreSQL history, what kind of applications work with PostgreSQL database and much more. Furthermore, the chapter also gives the reader an overview of the PostgreSQL, a brief history of PostgreSQL, PostgreSQL release cycle and the current impact of PostgreSQL in the market.

Chapter 2: Getting PostgreSQL to work - presents a detailed overview two ways of installing PostgreSQL - Source Code Installation and Binary Installation.

Chapter 3: Modern Options to get PostgreSQL - covers some other options to get PostgreSQL such as DBaaS/Cloud vendors and containers/Kubernetes.

Chapter 4: Global Objects in PostgreSQL - allows the reader to learn fundamental concepts of the global objects in a PostgreSQL cluster. These are handled per instance rather than per database. It describes Roles/Users/Groups, Tablespaces and Databases with examples.

Chapter 5: Architecture of PostgreSQL - gives special attention to the main components with respect to Memory Architecture, Background Processes and Physical Architecture.

Chapter 6: PostgreSQL Internals - explains Internals of the PostgreSQL from the perspective of ACID, MVCC, Transaction Isolation levels, Query Processing, and Vacuum.

Chapter 7: Backup and Restore in PostgreSQL - explains the importance of a backup strategy for a PostgreSQL solution, and teaches the different ways to get them. Then it shows how to restore these backups into a working service. Finally, it presents some of the most widely used tools for backup and restore such pgBackRest, WAL-G and so on

Chapter 8: Replicating Data - is dedicated to high availability architecture. It covers the main current replication techniques. It explains their differences, how to configure them, and their advantages and limitations.

Chapter 9: Security and Access Control - explains the importance of access control and security, and how PostgreSQL handles the authentication and authorization. The pg_hba and the GRANT/REVOKE are explored here.

Chapter 10: Most used Extensions/Tools - focus on some of the important extensions used which helps in extending the default behaviour of Postgresql in most frequent utilities. It cover the extensions like pg_stat_statements & pg_repack.

Chapter 11: Basic Database Objects - touches the concepts of database schemas, DDL command, DML Queries and DCL commands that are important from PostgreSQL Developer perspective. The main topics covered are Schema management, Data Definition Language command, Data Manipulation Language commands, control statements and constraints.

Chapter 12: Advance Database Objects - covers the advanced PL/PgSQL concepts like procedures, triggers and rules. It also covers details on custom data types in PostgreSQL.

Chapter 13: Performance Tuning - focuses on index creations along with best practices to be followed in configuration files. Once data size increases, there comes need to tune the database queries using Index and Explain Plan.

Chapter 14: Troubleshooting - gives insights on analyzing log files using database commands as well as OS commands. A DBA should also be aware on basics command of Operating System which are also covered from the DBA perspective.

Chapter 15: Contributing to PostgreSQL Community - discusses how to be part of PostgreSQL Community and build online presence.

Coloured Images

Please follow the link to download the *Coloured Images* of the book:

<https://rebrand.ly/gcsuy63>

The code bundle for the book is also hosted on GitHub at **<https://github.com/bpbpublications/PostgreSQL-for-Jobseekers>**. In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **<https://github.com/bpbpublications>**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Did you know that BPB offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.bpbonline.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at :

business@bpbonline.com for more details.

At www.bpbonline.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on BPB books and eBooks.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at business@bpbonline.com with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit www.bpbonline.com. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit www.bpbonline.com.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

<https://discord.bpbonline.com>



Table of Contents

1. Introduction to Opensource Database - PostgreSQL	1
Introduction.....	1
Structure.....	1
Objectives.....	2
Open-source introduction	2
<i>The origin of free software.....</i>	<i>2</i>
<i>The open source concept.....</i>	<i>3</i>
<i>The open source nowadays.....</i>	<i>5</i>
A brief history of PostgreSQL.....	6
<i>The POSTGRES project.....</i>	<i>6</i>
Postgres95	7
PostgreSQL	7
<i>PostgreSQL versions key features.....</i>	<i>8</i>
<i>PostgreSQL in stats on a single image</i>	<i>10</i>
PostgreSQL release cycle	10
<i>The current impact of PostgreSQL on the market.....</i>	<i>12</i>
Companies that use PostgreSQL.....	14
<i>Companies that help in enhancing PostgreSQL</i>	<i>14</i>
<i>Advantages of PostgreSQL</i>	<i>16</i>
<i>Distributed architecture with PostgreSQL.....</i>	<i>17</i>
<i>StatefulSet with PostgreSQL</i>	<i>18</i>
Conclusion	18
Bibliography.....	19
2. Getting PostgreSQL to work	21
Introduction.....	21
Structure.....	21
Objectives.....	22
Source code installation	22
<i>Short version of source code installation</i>	<i>22</i>
<i>Pre-requisites</i>	<i>23</i>
<i>Downloading the source.....</i>	<i>23</i>
Installation procedure.....	25

<i>Verifying directory structure</i>	27
<i>Adding postgres user</i>	27
<i>Creating data directory</i>	28
Initializing PostgreSQL	28
<i>Validating the data directory</i>	30
<i>Start PostgreSQL database</i>	31
<i>Verify postgres process is running</i>	31
Binary installation.....	33
<i>Create repository configuration</i>	34
<i>Import the repository signing key</i>	35
<i>Update the package list</i>	35
Installing PostgreSQL	36
Conclusion	38
Bibliography.....	39
3. Modern Options to get PostgreSQL	41
Introduction.....	41
Structure.....	41
Objectives.....	42
Other ways to get PostgreSQL.....	42
<i>On-premise, virtualization, containers, and cloud</i>	42
<i>On-premise</i>	42
<i>Virtualization</i>	43
<i>Containers</i>	44
<i>The Cloud</i>	45
Getting PostgreSQL on modern systems	46
<i>PostgreSQL on Docker</i>	47
<i>PostgreSQL on Kubernetes</i>	48
<i>PostgreSQL on The Cloud</i>	53
Conclusion	58
Bibliography.....	58
4. Global Objects in PostgreSQL	59
Introduction.....	59
Structure.....	59
Objectives.....	59

Users/Groups/Roles	60
Tablespaces.....	64
Databases.....	65
<i>CREATE DATABASE</i> command.....	67
<i>createdb</i> program	68
Using <i>pgAdmin Wizard</i>	69
Conclusion	72
Bibliography.....	72
5. Architecture of PostgreSQL	73
Introduction.....	73
Structure.....	73
Objectives.....	74
Memory architecture	74
<i>Shared memory</i>	74
<i>Shared buffers</i>	75
<i>WAL buffers</i>	75
<i>CLOG buffers</i>	75
<i>Process memory</i>	76
<i>Temporary buffers</i>	76
<i>Work memory</i>	76
<i>Maintenance work memory</i>	77
Background processes	77
<i>Postmaster</i>	77
<i>Checkpoint</i>	78
<i>Writer or background writer</i>	78
<i>Autovacuum</i>	79
<i>Stats collector</i>	79
<i>Logger</i>	79
<i>Archiver</i>	79
<i>WAL writer</i>	80
<i>WAL sender</i>	80
<i>WAL receiver</i>	81
Physical files.....	81
<i>Data files</i>	84
<i>WAL files</i>	84

<i>Temporary files</i>	85
<i>CLOG files</i>	86
<i>Stat files</i>	86
<i>Log files</i>	86
<i>WAL archive files</i>	86
<i>Conclusion</i>	87
Bibliography.....	89
6. PostgreSQL Internals	91
Introduction.....	91
Structure.....	91
Objectives.....	92
ACID.....	92
<i>Atomicity</i>	92
<i>Consistency</i>	92
<i>Isolation</i>	92
<i>Durability</i>	92
MVCC.....	93
Vacuum.....	94
<i>Autovacuum</i>	94
<i>VACUUM FULL</i>	95
<i>Manual VACUUM</i>	95
<i>pg_repack</i>	96
<i>Preventing transaction ID wraparound failures</i>	97
Transaction isolation levels.....	97
<i>Phenomena</i>	97
<i>Dirty read</i>	97
<i>Non-repetable read</i>	98
<i>Phantom read</i>	99
<i>Serialization anomaly</i>	99
<i>Isolation levels</i>	100
<i>Read uncommitted/committed</i>	100
<i>Repeatable read</i>	101
<i>Serializable</i>	102
Query processing.....	102
<i>Parser</i>	103

<i>Rewriter</i>	104
<i>Planner</i>	104
<i>Executor</i>	105
Conclusion	105
Bibliography	106
7. Backup and Restore in PostgreSQL	107
Introduction	107
Structure	107
Objectives	108
Backup	108
<i>Physical backup</i>	109
<i>pg_basebackup</i>	109
<i>Point in time Recovery/Archival</i>	111
<i>Pros and cons of physical backup</i>	111
<i>Logical backup</i>	112
<i>pg_dump</i>	112
<i>pg_dumpall</i>	112
<i>Pros and cons of logical backup</i>	115
Restore	115
<i>psql</i>	115
<i>pg_restore</i>	117
Useful backup and restore tools	118
<i>pgBackRest</i>	119
<i>Barman</i>	125
<i>pg_probackup</i>	130
Conclusion	133
Bibliography	134
8. Replicating Data	135
Introduction	135
Structure	135
Objectives	135
Physical replication	136
<i>Hot standby</i>	138
<i>Archive recovery</i>	139

Streaming replication	141
Cascading	142
Delayed replica	144
Configuration	146
Logical replication.....	146
Architecture	147
Publication.....	147
Subscription.....	148
Publisher node as well as subscription node	148
Conclusion	150
9. Security and Access Control	151
Introduction.....	151
Structure.....	151
Objectives.....	152
Authentication	152
The <i>pg_hba.conf</i>	152
<i>local</i>	153
<i>host</i>	153
<i>hostssl</i>	154
<i>hostnossl</i>	154
<i>hostgssenc</i>	154
<i>hostnogssenc</i>	154
Database.....	154
User.....	155
Address.....	155
Method.....	156
[Options]	156
Authentication methods.....	157
The <i>pg_ident.conf</i>	158
Examples	159
Authorization.....	161
Role attributes	161
Object ownership.....	163
Objects privileges.....	164
Conclusion	165

Bibliography.....	165
10. Most used Extensions/Tools.....	167
Introduction.....	167
Structure.....	167
Objectives.....	168
Extensions.....	168
<i>pg_cron</i>	169
<i>pg_stat_statements</i>	171
<i>pg_repack</i>	174
Tools.....	177
<i>pgbadger</i>	177
<i>pgbench</i>	179
<i>pgbouncer</i>	181
Conclusion.....	183
Bibliography.....	183
11. Basic Database Objects.....	185
Introduction.....	185
Structure.....	185
Objectives.....	185
Managing schemas.....	186
<i>DB cluster</i>	186
<i>Users/roles</i>	188
<i>Databases</i>	188
<i>Tablespaces</i>	188
<i>Schemas</i>	188
<i>Default - Public Schema</i>	188
<i>SEARCH_PATH in Schema</i>	189
Managing DB Objects using DDL commands.....	192
<i>Data types</i>	192
<i>Table</i>	193
<i>Create table</i>	193
<i>Alter table</i>	194
<i>Drop table</i>	194
<i>Truncate</i>	194

View	195
Sequences.....	195
Enforcing data integrity using constraints.....	197
Manipulating data using DML Queries	201
Inserting data	201
Updating data.....	202
Deleting data	203
Select (Retrieve) data	203
Joins used in data retrieval	204
Inner join	204
Left outer join.....	206
Right outer join.....	207
Full outer join.....	208
Aggregate functions.....	209
Conclusion	210
Bibliography.....	210
12. Advance Database Objects.....	211
Introduction.....	211
Structure.....	211
Objectives.....	212
Managing procedures/functions.....	212
Function.....	212
Function execution syntax.....	214
Function execution example.....	215
Procedure.....	216
Procedure execution syntax.....	217
Procedure execution	219
Managing triggers	222
Trigger function.....	222
Event trigger	227
Managing rules	227
Trigger versus rules	229
Custom data type	230
Conclusion	231
Bibliography.....	231

13. Performance Tuning	233
Introduction.....	233
Structure.....	233
Objectives.....	234
Indexes.....	234
<i>Reindex</i>	235
<i>Index types</i>	236
<i>Btree index</i>	236
<i>Hash index</i>	237
<i>GiST and SP-GiST index</i>	237
<i>Gin index</i>	237
<i>Brin index</i>	237
<i>Indexes and expressions</i>	237
Statistics.....	238
<i>Statistics in pg_statistics</i>	240
<i>Statistics in pg_statistics_ext_data</i>	242
<i>Functional dependencies</i>	243
<i>Number of distinct values counts</i>	244
<i>Most common values list</i>	246
Explain plan	247
Best practices for the postgresql.conf parameters	250
<i>shared_buffers</i>	251
<i>work_mem</i>	251
<i>autovacuum</i>	251
<i>effective_cache_size</i>	252
<i>maintenance_work_mem</i>	252
<i>max_connections</i>	252
Summary	253
Conclusion	253
Bibliography.....	253
14. Troubleshooting.....	255
Introduction.....	255
Structure.....	255
Objectives.....	256
Debugging using log files	256

<i>Where to log</i>	256
<i>logging_collector</i>	256
<i>log_destination</i>	257
<i>log_directory</i>	257
<i>log_filename</i>	258
<i>log_rotation_age</i>	258
<i>log_rotation_size</i>	258
<i>log_truncate_on_rotation</i>	258
<i>What to log</i>	259
<i>log_line_prefix</i>	259
<i>log_connections/log_disconnections</i>	260
<i>log_min_duration_statement</i>	260
<i>log_lock_waits</i>	260
<i>log_autovacuum_min_duration</i>	260
<i>Parameters summary</i>	261
Debugging using PostgreSQL tools and commands	261
<i>Gather information</i>	262
<i>Check the PostgreSQL version</i>	262
<i>Check database and objects size</i>	262
<i>Check database connections</i>	263
<i>Check slow queries</i>	264
<i>Instruct PostgreSQL</i>	266
<i>Vacuuming and analyzing</i>	266
<i>Terminate queries or user sessions</i>	266
<i>Manage replication</i>	267
Debugging using Operating System tools and commands.....	268
<i>Service and system-wide tools</i>	268
<i>systemctl</i>	268
<i>free</i>	269
<i>df</i>	270
<i>Processes-oriented tools</i>	270
<i>ps/pgrep</i>	271
<i>Log and events</i>	272
<i>journalctl</i>	272
Conclusion	273

15. Contributing to PostgreSQL Community	275
Introduction.....	275
Structure.....	275
Objectives.....	276
PostgreSQL community and its members	276
<i>Core members of the PostgreSQL community.....</i>	<i>276</i>
<i>Working pattern of PostgreSQL community</i>	<i>277</i>
<i>Earning of PostgreSQL community</i>	<i>278</i>
Different ways to contribute	279
<i>Code contributor</i>	<i>279</i>
<i>Bug reporter.....</i>	<i>280</i>
<i>Participate in the mailing lists.....</i>	<i>280</i>
<i>Improving or creating new documentation.....</i>	<i>281</i>
<i>Participating in events</i>	<i>282</i>
<i>Supporting the community</i>	<i>283</i>
Conclusion	283
Index.....	285

CHAPTER 1

Introduction to OpenSource Database - PostgreSQL

Introduction

PostgreSQL, also known as “Postgres,” is a popular open-source database management system. It is known for its strong support for reliability, data integrity, and concurrency. Postgres has a large and active development community and is widely used in businesses, government agencies, and other organizations around the world.

Its path in the industry as open-source software is the result of the initiative and efforts of many people through the years.

Structure

This chapter introduces the open-source concept and covers the main general aspects of PostgreSQL like its history, how it was developed and released and how much it has impacted the current industry scene.

Topics to be covered:

- OpenSource Introduction
- A Brief History of PostgreSQL

- PostgreSQL Release Cycle
- Current impact of PostgreSQL in the market

Objectives

You will learn about the establishment of the open-source software concept and how it became what it is nowadays. We will review the history of PostgreSQL from its initial conception and the way it evolved to “The World’s Most Advanced Open Source Relational Database.”

Then you will learn how the release cycle of PostgreSQL works to deliver new features, fixes, and improvements with every minor and major release. Finally, we will look at the current panorama of PostgreSQL and the main advantages you could get when choosing PostgreSQL.

Open-source introduction

Open source refers to a type of software whose source code is available to the public, meaning that anyone can view, modify, and distribute the code. Open-source software is typically developed by a community of volunteers, who work together to improve the software and share their modifications with others.

The origin of free software

Free software, also known as “libre software,” is software that is distributed with the freedom to use, study, modify, and distribute the software and its source code. The concept of free software has its roots in the early days of computer programming, when programmers would freely share their code and ideas with each other.

By the mid-1970s, companies created the software with rigorous licenses to *protect* it and ensure all the profit for its usage goes to the creator; this is what we know as proprietary software or closed software.

After years, other people found this way of creating and distributing software tedious and frustrating. Some thought they could improve the existing software, but their ideas were not viable since the vendors spread the software without its source code, and multiple laws did not allow modifications. The following are some of the main events in the free software history.

- In the early 1980s Richard M. Stallman announced GNU (from GNU is Not Unix), the first genuine free software initiative. He aimed to release an

operating system everyone could get, use, and distribute. He also started the ideology of free software. In his own words, all the software should be free and “accessible to everyone as freely as possible.” (*Reference: Free Software Foundation*)

- A couple of years later, in the mid-1980s, the **Free Software Foundation (FSF)** was established, with Richard as its president.
- In the late 1980s, the Free Software Foundation published the GNU **General Public License (GPL)**. This license’s essence is to clarify that all the software created under it will be free, so everyone can run the software, study it (get the source code), distribute it, make changes, and share it. This license introduced significant changes to how the software was created and opened the door to various new free software projects.

The ideology of free software had some caveats. It got a philosophical-political solid sense. The concept was beyond that everyone should be able to access the software without paying; it represented the idea that the software is a human right, which should be taken and accepted by other elemental organisms such as the governments.

Also, the “label” free software usually requires to be distinguished between the concept of free (no charge) and free (as in freedom), which caused some confusion in general.

The open source concept

The open-source concept is based on the idea that the source code for a piece of software should be freely available to anyone, and that anyone should be able to use, modify, and distribute the software without restriction. This allows for collaboration and transparency in the development process, as anyone can contribute to the software and see how it works.

Open-source software is typically developed using a decentralized, distributed model, with contributions coming from a wide range of individuals and organizations. This can make it more resilient to changes in leadership or funding, as there is no single entity in control of the software.

By the late 1990s, the people supporting the *free* software initiatives considered introducing some adjustments to the free software concept. Two important events heavily impulsed their interest in collaborative and shared software development: 1) the release of the Netscape browser source code and 2) the impact of the Linux kernel.

These happenings led to think the label “free” was not precisely adequate to describe the actual situation of these and any other project created with this development ideology. Then a new *code word* came into the scene: the **open-source** software.

We can look at some details from the two main projects that impulse this change.

Netscape browser:

- The Netscape browser was widely used, they used to release their software as any other proprietary software. But motivated by the idea of having the help of many developers around the world, the company decided to make their code open source.
- Making the browser code open source their vast existing community actively participate in the development process, making the bug identification and fix creation and release a more quick and efficient process.
- Undoubtedly, many other companies and organizations consider open source a viable option.

The Linux Kernel:

- The creator of the Linux kernel is the Finnish software engineer Linus Torvalds.
- He started his work just trying to adapt the “home” version of **UNiplexed Information Computing System** (UNIX) called **MINIX** (from Mini UNIX) to use on his **Personal Computer** (PC).
- As the project started with the idea of free software, some GNU tools were used, so multiple people related to the existing GNU projects got interested.
- The number of users and collaborators of the Linux project multiplied very quickly.
- People from many different places integrated into the community with diverse backgrounds and interests.
- This apparently “non-organized” way of work proved solid and highly effective.
- All the stages required in the development process, such as tests, bug identification, fixes, releases, and so on., got executed highly faster than the usual time.

The speed of the community growth, the quality of the products, and all the popularity these projects gained set them in a notorious place in the industry and a big part of this success was the usage of the open-source methodology.

These two relevant events made adopting the open-source concept more accessible and transparent. Undoubtedly, the initial efforts from the **Free Software Foundation (FSF)** put things rolling in the direction of getting software differently from the privative model. However, its orientation to have a political-looking side and the way their members built the projects, mainly by small groups or even single dedicated persons, was overpassed by the collaborative communities' notorious engagement. The open-source label got a meaning that many people felt identified with.

The open-source software will keep its path regarding making the software available for everyone. It also promotes a collaborative and distributed way of development. It is not making the idea that software is a human right its primary purpose, but that anyone can participate and share efforts to build it. Please find the high-level overview of Open-source timeline in the *Figure 1.1*:

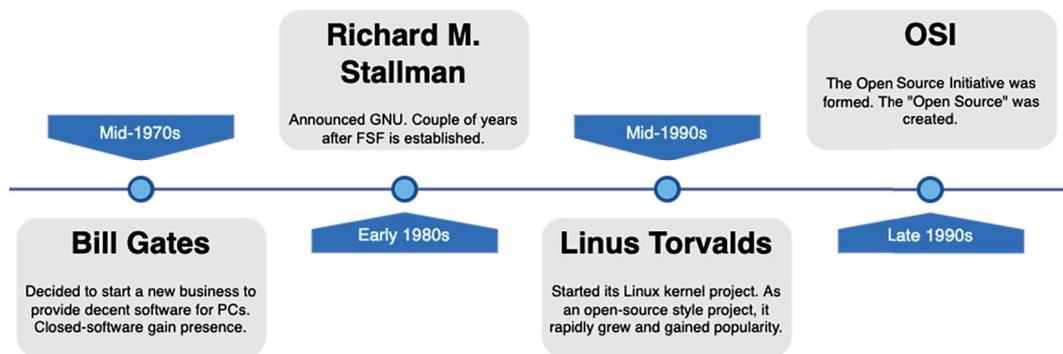


Figure 1.1: Open Source timeline

The open source nowadays

The Open Source has grown in adoption and usage. Multiple big projects are created under its ideology and support the concept that everyone can use and improve the software. The communities of the open-source projects usually are very active, and the results of their collaborative efforts are remarkable.

The way these communities work lets them provide early access releases and make the beta, or even delta, versions available, unlike the closed software (A.K.A privative software). Which usually only releases "complete" final versions of their products and fixes the bugs through their internal process. The open-source model leverages the capacity of the community of users, often collaborators, to find and fix the bugs and issues from the early access releases. So, following the release cycle, these will be addressed and solved by the time the alpha or stable release is made public.

All these characteristics have made open source reliable and sufficient for large enterprise solutions. Modern software takes the best from the collaborative model and the dynamism of the user's communities and software engineers to build compelling, secure, and resilient solutions. We can get coverage for all our system layers, backend, middleware, frontend, security, high availability, and so on., from the current open-source options.

And you know what? Our loved PostgreSQL is one of these open-source options. Now that you know what Open Source is, let's dive into PostgreSQL.

A brief history of PostgreSQL

PostgreSQL, also known as "Postgres," is a powerful and feature-rich open-source database management system. It was first developed at the University of California, Berkeley in the mid-1980s as a research project by a team led by computer science professor Michael Stonebraker.

The POSTGRES project

The POSTGRES project was a research project at the University of California, Berkeley that developed the first version of the PostgreSQL database management system. The project was led by computer science professor Michael Stonebraker and a team of researchers, and it was funded by the **Defense Advanced Research Projects Agency (DARPA)** and the **National Science Foundation (NSF)**.

The goal of the POSTGRES project was to develop a database management system that could handle complex data types and support user-defined types and functions.

By 1993, the number of users from the external community was twice. The participation of these users and their findings to improve the product made project maintenance more difficult. Some members of the original project thought the time they were using to address the code issues and support could be used for development and research. So finally, they decided to terminate the Berkeley POSTGRES project that year; the final release was version 4.2.

The original POSTGRES project found application in several different projects. The financial market, the aerospace industry, medical, and several geographic projects used the software for their purposes. It was commercialized by Illustra, merged into Informix, and finally bought by IBM.

Postgres95

The next part is the beginning of the history of our current PostgreSQL as an open-source database.

By the mid-1990s, after the termination of the initial POSTGRES projects. Andrew Yu and Jolly Chen, two Ph.D. students from Stonebraker's laboratory, took the original code and made many changes and improvements, the more relevant one was the addition of an SQL interpreter, and then they released it to the web. This new project, inspired by the POSTGRES project and continuing its legacy, started its history as an open-source database.

PostgreSQL

After a couple of years, the new name would not stand the pass of time. So it was re-launched in 1996 as PostgreSQL version 6.0. The name was chosen to keep the relationship with the original POSTGRES project and added the SQL suffix to make clear the support of the SQL language; the original project used a query language called PostQUEL. Also, the version number was set to keep continuity from the releases of the previous project. Even now, the nickname *Postgres* is still used for easy pronunciation and keeps remembering its origins.

This just released system rapidly got the attention of multiple hackers and experts from the databases world around the globe. All they got compromised to the improvement and development of new features. The contributions of this community added to PostgreSQL a great code consistency, fixed an innumerable number of bugs, created a mail list to report bugs and usability issues, tested the quality of the software extensively, and filled the documentation gaps for both developers and users.

The PostgreSQL community behind all this work is known as The PostgreSQL Global Development Group. The code comes from contributions from proprietary vendors, support companies, and open-source programmers. Its presence on the web at the www.postgresql.org site started on October 22, 1996, and since then, this portal has gained an essential place in the open-source world.

PostgreSQL versions are released under the **Free and Open-source Software (FOSS)** license. This means anyone can get, execute, study, change and distribute the software (the free part). Also, the source code is openly shared, and people are encouraged to participate voluntarily in improving the software (the open-source part).

PostgreSQL versions key features

PostgreSQL has undergone many versions and updates since its inception, and each version has introduced new features and improvements. Here are some key features that have been introduced in various versions of PostgreSQL. (Reference: *About PostgreSQL*)

Versions 6.0 - 8.0

- **Multiversion Concurrency Control (MVCC).** Table-level locking was replaced with a sophisticated multiversion concurrency control system, which allows readers to continue reading consistent data during writer activity and enables online (hot) backups while the database is running.
- **Important SQL features.** Many SQL enhancements were made, including subselects, defaults, constraints, primary keys, foreign keys, quoted identifiers, literal string type coercion, type casting, and binary and hexadecimal integer input.
- **Improved built-in types.** New native types were added.
- **Speed.** Major speed and performance increases of 20-40% were made, and backend startup time was decreased by 80%. (Reference: *PostgreSQL Community History*)
- **Write-Ahead Log (WAL)** is an efficient mechanism to log all the data changes into files that can be used to replay on top of data files after a checkpoint. This brings the possibility of performing recovery after a system crash and Point-In-Time Recovery (PITR).
- **SQL schemas.** Logical containers to organize the tables and other relational objects in a database.
- **TOAST.** Additional data files to handle those records can surpass the size of the data pages, usually 8KB.
AUTOVACUUM. An automated background process to trigger the VACUUM operations over the database tables.

Versions 8.1 - 9.6:

- **Window functions:** Powerful SQL functions to perform calculations over a set of rows related to the current one. Used mainly for analytics and specialized reports.
- **Background Checkpointer:** Background process in charge of triggering the checkpoint in a database. This will throttle the write of data changes from

memory (cache) to the physical data files (disk). So the IO impact on the system is lower as possible.

- **Parallel query:** A feature that enabled the capacity to create a coordinator process and some workers to split the reads operation over large tables, speeding up the query performance.
- **Unlogged tables:** A special table type that does not log its data changes to the WAL. This is especially useful to boost the performance in certain write workloads that can be temporary or recreated and don't need crash recovery protection.
- **Foreign table inheritance:** A table relation technique that enables the possibility of creating a "master" or template table so that others can inherit from it. This also made possible the first table partitioning solution.
- **Streaming Replication:** A versatile and robust data replication mechanism. Built on top of the WAL recovery feature, this replication mode lets a standby get the data changes from the primary WAL directly without needing to copy the physical file.
- **Hot Standby.** When a replica or standby is configured with this option, the instance becomes available for reading operations. This permits design clusters to scale out the reads without impact in the primary instance.
- **Extension Installation.** This enabled the possibility of extending the core features of PostgreSQL by adding extensions, which are pieces of code that the PostgreSQL database can execute once installed.

Versions 10 – 14:

- **B-tree deduplication.** A new enhancement for the general-purpose index structure B-tree avoids adding duplicated values, for example, when creating an index in a column that allows NULL values.
- **REINDEX CONCURRENTLY.** An addition that allows rebuilding indexes in stages, avoiding a prolonged lock in the table so other operations can continue working.
- **Declarative table partitioning.** When this was added, PostgreSQL could create partitioned tables with native syntax, an improvement from the previous inherit approach.
- **Logical replication.** Even when the support for logical decoding was added in the 9.4 version, the native syntax was added until version 10. Now is possible the establish the replication technique without needing external extensions.