

O'REILLY®

3rd Edition
Covers Version 10



PostgreSQL Up & Running

A PRACTICAL GUIDE TO THE ADVANCED OPEN SOURCE DATABASE

Regina Obe & Leo Hsu

PostgreSQL: Up and Running

Thinking of migrating to PostgreSQL? This clear, fast-paced introduction helps you understand and use this open source database system. Not only will you learn about the enterprise class features in versions 9.5 to 10, you'll also discover that PostgreSQL is more than a database system—it's an impressive application platform as well.

With examples throughout, this book shows you how to achieve tasks that are difficult or impossible in other databases. This third edition covers new features, such as ANSI-SQL constructs found only in proprietary databases until now: foreign data wrapper (FDW) enhancements; new full text functions and operator syntax introduced in version 9.6; XML constructs new in version 10; query parallelization features introduced in 9.6 and enhanced in 10; built-in logical replication introduced in version 10.

If you're a current PostgreSQL user, you'll pick up gems you may have missed before.

- Learn basic administration tasks such as role management, database creation, backup, and restore
- Apply the `psql` command-line utility and the `pgAdmin` graphical administration tool
- Explore PostgreSQL tables, constraints, and indexes
- Learn powerful SQL constructs not generally found in other databases
- Use several different languages to write database functions
- Tune your queries to run as fast as your hardware will allow
- Query external and variegated data sources with foreign data wrappers
- Learn how to use built-in replication to replicate data

Regina Obe is a coprincipal of Paragon Corporation, a database consulting company based in Boston. She has over 20 years of professional experience in various programming languages and database systems, with special focus on spatial databases. She is a member of the PostGIS and GEOS steering and core development teams and a developer on the `pgRouting` team. Regina holds a BS degree in mechanical engineering from the Massachusetts Institute of Technology. She coauthored *PostGIS in Action* (Manning) and *pgRouting: A Practical Guide* (Locate Press).

Leo Hsu is a coprincipal of Paragon Corporation, a database consulting company based in Boston. He has over 20 years of professional experience developing and thinking about databases for organizations large and small. Leo holds an MS degree in engineering of economic systems from Stanford University and BS degrees in mechanical engineering and economics from the Massachusetts Institute of Technology. He coauthored *PostGIS in Action* (Manning) and *pgRouting: A Practical Guide* (Locate Press).

US \$39.99

CAN \$52.99

ISBN: 978-1-491-96341-8



Twitter: @oreillymedia
facebook.com/oreilly

THIRD EDITION

PostgreSQL: Up and Running

*A Practical Guide to the Advanced
Open Source Database*

Regina O. Obe and Leo S. Hsu

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

PostgreSQL: Up and Running

by Regina O. Obe and Leo S. Hsu

Copyright © 2018 Regina Obe, Leo Hsu. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://oreilly.com/safari>). For more information, contact our corporate/institutional sales department: 800-998-9938 or corporate@oreilly.com.

Editor: Andy Oram

Production Editor: Melanie Yarbrough

Copyeditor: Kim Cofer

Proofreader: Christina Edwards

Indexer: Lucie Haskins

Interior Designer: David Futato

Cover Designer: Karen Montgomery

Illustrator: Rebecca Demarest

October 2017: Third Edition

Revision History for the Third Edition

2017-10-10: First Release

See <http://oreilly.com/catalog/errata.csp?isbn=9781491963418> for release details.

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. *PostgreSQL: Up and Running*, the cover image, and related trade dress are trademarks of O'Reilly Media, Inc.

While the publisher and the authors have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the authors disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

978-1-491-96341-8

[LSI]

Table of Contents

Preface.....	xi
1. The Basics.....	1
Why PostgreSQL?	1
Why Not PostgreSQL?	3
Where to Get PostgreSQL	4
Administration Tools	4
psql	4
pgAdmin	5
phpPgAdmin	6
Adminer	7
PostgreSQL Database Objects	8
What's New in Latest Versions of PostgreSQL?	13
Why Upgrade?	13
Features Introduced in PostgreSQL 10	13
Features Introduced in PostgreSQL 9.6	15
Features Introduced in PostgreSQL 9.5	16
Features Introduced in PostgreSQL 9.4	17
Database Drivers	19
Where to Get Help	19
Notable PostgreSQL Forks	20
2. Database Administration.....	21
Configuration Files	21
Making Configurations Take Effect	22
The postgresql.conf File	23
The pg_hba.conf File	27
Managing Connections	30

Check for Queries Being Blocked	32
Roles	32
Creating Login Roles	32
Creating Group Roles	33
Database Creation	36
Template Databases	36
Using Schemas	37
Privileges	39
Types of Privileges	39
Getting Started	40
GRANT	40
Default Privileges	41
Privilege Idiosyncrasies	42
Extensions	43
Installing Extensions	44
Common Extensions	46
Backup and Restore	48
Selective Backup Using pg_dump	49
Systemwide Backup Using pg_dumpall	51
Restoring Data	51
Managing Disk Storage with Tablespaces	53
Creating Tablespaces	53
Moving Objects Among Tablespaces	53
Verboten Practices	54
Don't Delete PostgreSQL Core System Files and Binaries	54
Don't Grant Full OS Administrative Privileges to the Postgres System	
Account (postgres)	55
Don't Set shared_buffers Too High	55
Don't Try to Start PostgreSQL on a Port Already in Use	56
3. psql.....	57
Environment Variables	57
Interactive versus Noninteractive psql	58
psql Customizations	59
Custom Prompts	61
Timing Executions	61
Autocommit Commands	61
Shortcuts	62
Retrieving Prior Commands	62
psql Gems	63
Executing Shell Commands	63
Watching Statements	63

Retrieving Details of Database Objects	64
Crosstabs	64
Dynamic SQL Execution	65
Importing and Exporting Data	66
psql Import	66
psql Export	67
Copying from or to Program	68
Basic Reporting	68
4. Using pgAdmin.	73
Getting Started	74
Overview of Features	74
Connecting to a PostgreSQL Server	75
Navigating pgAdmin	76
pgAdmin Features	77
Autogenerating Queries from Table Definitions	77
Accessing psql from pgAdmin3	78
Editing postgresql.conf and pg_hba.conf from pgAdmin3	79
Creating Database Assets and Setting Privileges	79
Import and Export	82
Backup and Restore	85
pgScript	88
Graphical Explain	90
Job Scheduling with pgAgent	92
Installing pgAgent	92
Scheduling Jobs	93
Helpful pgAgent Queries	95
5. Data Types.	97
Numerics	97
Serials	98
Generate Series Function	98
Textuals	99
String Functions	100
Splitting Strings into Arrays, Tables, or Substrings	101
Regular Expressions and Pattern Matching	101
Temporals	103
Time Zones: What They Are and Are Not	105
Datetime Operators and Functions	107
Arrays	109
Array Constructors	109
Unnesting Arrays to Rows	111

Array Slicing and Splicing	112
Referencing Elements in an Array	112
Array Containment Checks	113
Range Types	114
Discrete Versus Continuous Ranges	114
Built-in Range Types	115
Defining Ranges	115
Defining Tables with Ranges	116
Range Operators	117
JSON	118
Inserting JSON Data	118
Querying JSON	119
Outputting JSON	120
Binary JSON: jsonb	121
Editing JSONB data	123
XML	125
Inserting XML Data	125
Querying XML Data	126
Full Text Search	128
FTS Configurations	129
TSVectors	132
TSQueries	134
Using Full Text Search	136
Ranking Results	137
Full Text Stripping	138
Full Text Support for JSON and JSONB	139
Custom and Composite Data Types	140
All Tables Are Custom Data Types	140
Building Custom Data Types	141
Composites and NULLs	141
Building Operators and Functions for Custom Types	142
6. Tables, Constraints, and Indexes.....	145
Tables	145
Basic Table Creation	145
Inherited Tables	147
Partitioned Tables	148
Unlogged Tables	150
TYPE OF	151
Constraints	152
Foreign Key Constraints	152
Unique Constraints	153

Check Constraints	153
Exclusion Constraints	154
Indexes	155
PostgreSQL Stock Indexes	155
Operator Classes	158
Functional Indexes	159
Partial Indexes	160
Multicolumn Indexes	161
7. SQL: The PostgreSQL Way.....	163
Views	163
Single Table Views	164
Using Triggers to Update Views	165
Materialized Views	167
Handy Constructions	169
DISTINCT ON	169
LIMIT and OFFSET	170
Shorthand Casting	170
Multirow Insert	171
ILIKE for Case-Insensitive Search	171
ANY Array Search	172
Set-Returning Functions in SELECT	172
Restricting DELETE, UPDATE, and SELECT from Inherited Tables	173
DELETE USING	173
Returning Affected Records to the User	174
UPSERTs: INSERT ON CONFLICT UPDATE	174
Composite Types in Queries	175
Dollar Quoting	177
DO	177
FILTER Clause for Aggregates	179
Percentiles and Mode	180
Window Functions	182
PARTITION BY	183
ORDER BY	184
Common Table Expressions	186
Basic CTEs	186
Writable CTEs	187
Recursive CTE	188
Lateral Joins	189
WITH ORDINALITY	191
GROUPING SETS, CUBE, ROLLUP	193

8. Writing Functions.....	197
Anatomy of PostgreSQL Functions	197
Function Basics	197
Triggers and Trigger Functions	200
Aggregates	202
Trusted and Untrusted Languages	203
Writing Functions with SQL	203
Basic SQL Function	204
Writing SQL Aggregate Functions	205
Writing PL/pgSQL Functions	208
Basic PL/pgSQL Function	208
Writing Trigger Functions in PL/pgSQL	208
Writing PL/Python Functions	209
Basic Python Function	210
Writing PL/V8, PL/CoffeeScript, and PL/LiveScript Functions	211
Basic Functions	213
Writing Aggregate Functions with PL/V8	214
Writing Window Functions in PL/V8	215
9. Query Performance Tuning.....	219
EXPLAIN	219
EXPLAIN Options	219
Sample Runs and Output	220
Graphical Outputs	223
Gathering Statistics on Statements	225
Writing Better Queries	226
Overusing Subqueries in SELECT	226
Avoid SELECT *	229
Make Good Use of CASE	230
Using FILTER Instead of CASE	231
Parallelized Queries	231
What Does a Parallel Query Plan Look Like?	232
Parallel Scans	235
Parallel Joins	236
Guiding the Query Planner	236
Strategy Settings	236
How Useful Is Your Index?	237
Table Statistics	238
Random Page Cost and Quality of Drives	240
Caching	241

10. Replication and External Data	243
Replication Overview	243
Replication Jargon	244
Evolution of PostgreSQL Replication	246
Third-Party Replication Options	246
Setting Up Full Server Replication	247
Configuring the Master	247
Configuring the Slaves for Full Server Cluster Replication	249
Initiating the Streaming Replication Process	250
Replicating Only Some Tables or Databases with Logical Replication	250
Foreign Data Wrappers	252
Querying Flat Files	252
Querying Flat Files as Jagged Arrays	253
Querying Other PostgreSQL Servers	254
Querying Other Tabular Formats with ogr_fdw	257
Querying Nonconventional Data Sources	259
 A. Installing PostgreSQL.....	 263
 B. PostgreSQL Packaged Command-Line Tools.....	 267
 Index.....	 277

Preface

PostgreSQL bills itself as the world's most advanced open source database. We couldn't agree more.

What we hope to accomplish in this book is to give you a firm grounding in the concepts and features that make PostgreSQL so impressive. Along the way, we should convince you that PostgreSQL does indeed stand up to its claim to fame. Because the database is advanced, no book short of the 3500 pages of documentation can bring out all its glory. But then again, most users don't need to delve into the most abstruse features that PostgreSQL has to offer. So in our shorter 300-pager, we hope to get you, as the subtitle proclaims, *Up and Running*.

Each topic is presented with some context so you understand when to use it and what it offers. We assume you have prior experience with some other database so that we can jump right to the key points of PostgreSQL. We generously litter the pages of this book with links to references so you can dig deeper into topics of interest. These links lead to sections in the manual, to helpful articles, to blog posts of PostgreSQL vanguarders. We also link to our own site at [Postgres OnLine Journal](#), where we have collected many pieces that we have written on PostgreSQL and its interoperability with other applications.

This book focuses on PostgreSQL versions 9.5, 9.6, and 10, but we will cover some unique and advanced features that are also present in prior versions.

Audience

For migrants from other database engines, we'll point out parallels that PostgreSQL shares with other leading products. Perhaps more importantly, we highlight feats you can achieve with PostgreSQL that are difficult or impossible to do in other databases.

We stop short of teaching you SQL, as you'll find many excellent sources for that. SQL is much like chess—a few hours to learn, a lifetime to master. You have wisely chosen PostgreSQL. You'll be greatly rewarded.

If you're currently a savvy PostgreSQL user or a weather-beaten DBA, much of the material in this book should be familiar terrain, but you'll be sure to pick up some pointers and shortcuts introduced in newer versions of PostgreSQL. Perhaps you'll even find the hidden gem that eluded you. If nothing else, this book is at least ten times lighter than the PostgreSQL manual.

Not using PostgreSQL yet? This book is propaganda—the good kind. Each day you continue to use a database with limited SQL capabilities, you handicap yourself. Each day that you're wedded to a proprietary system, you're bleeding dollars.

Finally, if your work has nothing to do with databases or IT, or if you've just graduated from kindergarten, the cute picture of the elephant shrew on the cover should be worthy of the price alone.

For More Information on PostgreSQL

PostgreSQL has a well-maintained set of online documentation: [PostgreSQL manuals](#). We encourage you to bookmark it. The manual is available both as HTML and as a PDF. Hardcopy collector editions are available for purchase.

Other PostgreSQL resources include:

- *Planet PostgreSQL* is an aggregator of PostgreSQL blogs. You'll find PostgreSQL core developers and general users showcasing new features, novel ways to use existing ones, and reporting of bugs that have yet to be patched.
- *PostgreSQL Wiki* provides tips and tricks for managing various facets of the database and migrating from other databases.
- *PostgreSQL Books* is a list of books about PostgreSQL.
- *PostGIS in Action Books* is the website for the books we've written on PostGIS, the spatial extender for PostgreSQL, and more recently pgRouting, another PostgreSQL extension that provides network routing capabilities useful for building driving apps.

Code and Output Formatting

For elements in parentheses, we gravitate toward placing the open parenthesis on the same line as the preceding element and the closing parenthesis on a line by itself. This is a classic C formatting style that we like because it cuts down on the number of blank lines:

```
function(  
    Welcome to PostgreSQL  
);
```

We also remove gratuitous spaces in screen output, so if the formatting of your results doesn't match ours exactly, don't fret.

We omit the space after a serial comma for short elements. For example, ('a','b','c').

The SQL interpreter treats tabs, newlines, and carriage returns as whitespace. In our code, we generally use whitespaces for indentation, not tabs. Make sure that your editor doesn't automatically remove tabs, newlines, and carriage returns or convert them to something other than spaces.

After copying and pasting, if you find your code not working, check the copied code to make sure it looks like what we have in the listing.

We use examples based on both Linux and Windows. Path notations differ between the two, namely the use of solidus (/) versus reverse solidus (\). While on Windows, use the Linux solidus, always! /, not \. You may see a path such as `/postgresql_book/somefile.csv`. These are always relative to the root of your server. If you are on Windows, you must include the drive letter: `C:/postgresql_book/somefile.csv`.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, and file extensions.

Constant width

Used for program listings. Used within paragraphs, where needed for clarity, to refer to programming elements such as variables, functions, databases, data types, environment variables, statements, and keywords.

Constant width bold

Shows commands or other text that should be typed literally by the user.

Constant width italic

Shows text that should be replaced with user-supplied values or by values determined by context.



This icon signifies a tip, suggestion, or general note.



This icon indicates a warning or caution.

Using Code Examples

Code and data examples are available for download at http://www.postgresonline.com/downloads/postgresql_book_3e.zip.

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: “*PostgreSQL: Up and Running, Third Edition* by Regina Obe and Leo Hsu (O'Reilly). Copyright 2018 Regina Obe and Leo Hsu, 978-1-491-96341-8.”

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

O'Reilly Safari

 **Safari**® Safari (formerly Safari Books Online) is a membership-based training and reference platform for enterprise, government, educators, and individuals.

Members have access to thousands of books, training videos, Learning Paths, interactive tutorials, and curated playlists from over 250 publishers, including O'Reilly Media, Harvard Business Review, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Adobe, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, and Course Technology, among others.

For more information, please visit <http://oreilly.com/safari>.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

Please submit errata using the [book's errata page](#).

The companion site for this book is at <http://bit.ly/postgresql-up-and-running-3e>.

To contact the authors, send email to lr@pcorp.us.

To comment or ask technical questions to the publisher, send email to bookquestions@oreilly.com.

For more information about our books, courses, conferences, and news, see our website at <http://www.oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>

Follow us on Twitter: <http://twitter.com/oreillymedia>

Watch us on YouTube: <http://www.youtube.com/oreillymedia>

The Basics

PostgreSQL is an extremely powerful piece of software that introduces features you may not have seen before. Some of the features are also present in other well-known database engines, but under different names. This chapter lays out the main concepts you should know when starting to attack PostgreSQL documentation, and mentions some related terms in other databases.

We begin by pointing you to resources for downloading and installing PostgreSQL. Next, we provide an overview of indispensable administration tools followed by a review of PostgreSQL nomenclature. PostgreSQL 10 was recently released. We'll highlight some of the new features therein. We close with resources to turn to when you need additional guidance and to submit bug reports.

Why PostgreSQL?

PostgreSQL is an enterprise-class relational database management system, on par with the very best proprietary database systems: Oracle, Microsoft SQL Server, and IBM DB2, just to name a few. PostgreSQL is special because it's not just a database: it's also an application platform, and an impressive one at that.

PostgreSQL is fast. In benchmarks, PostgreSQL either exceeds or matches the performance of many other databases, both open source and proprietary.

PostgreSQL invites you to write stored procedures and functions in numerous programming languages. In addition to the prepackaged languages of C, SQL, and PL/pgSQL, you can easily enable support for additional languages such as PL/Perl, PL/Python, PL/V8 (aka PL/JavaScript), PL/Ruby, and PL/R. This support for a wide variety of languages allows you to choose the language with constructs that can best solve the problem at hand. For instance, use R for statistics and graphing, Python for calling web services, the Python SciPy library for scientific computing, and PL/V8 for

validating data, processing strings, and wrangling with JSON data. Easier yet, find a freely available function that you need, find out the language that it's written in, enable that specific language in PostgreSQL, and copy the code. No one will think less of you.

Most database products limit you to a predefined set of data types: integers, texts, Booleans, etc. Not only does PostgreSQL come with a larger built-in set than most, but you can define additional data types to suit your needs. Need complex numbers? Create a composite type made up of two floats. Have a triangle fetish? Create a coordinate type, then create a triangle type made up of three coordinate pairs. A dozenal activist? Create your own duodecimal type. Innovative types are useful insofar as the operators and functions that support them. So once you've created your special number types, don't forget to define basic arithmetic operations for them. Yes, PostgreSQL will let you customize the meaning of the symbols (+, -, /, *). Whenever you create a type, PostgreSQL automatically creates a companion array type for you. If you created a complex number type, arrays of complex numbers are available to you without additional work.

PostgreSQL also automatically creates types from any tables you define. For instance, create a table of dogs with columns such as breed, cuteness, and barkiness. Behind the scenes, PostgreSQL maintains a dogs data type for you. This amazingly useful bridge between the relational world and the object world means that you can treat data elements in a way that's convenient for the task at hand. You can create functions that work on one object at a time or functions that work on sets of objects at a time. Many third-party extensions for PostgreSQL leverage custom types to achieve performance gains, provide domain-specific constructs for shorter and more maintainable code, and accomplish feats you can only fantasize about with other database products.

Our principal advice is this: don't treat databases as dumb storage. A database such as PostgreSQL can be a full-fledged application platform. With a robust database, everything else is eye candy. Once you're versant in SQL, you'll be able to accomplish in seconds what would take a casual programmer hours, both in coding and running time.

In recent years, we've witnessed an upsurge of NoSQL movements (though much of it could be hype). Although PostgreSQL is fundamentally relational, you'll find plenty of facilities to handle nonrelational data. The ltree extension to PostgreSQL has been around since time immemorial and provides support for graphs. The hstore extensions let you store key-value pairs. JSON and JSONB types allow storage of documents similar to MongoDB. In many ways, PostgreSQL accommodated NoSQL before the term was even coined!

PostgreSQL just celebrated its 20th birthday, dating from its christening to PostgreSQL from Postgres95. The beginnings of the PostgreSQL code-base began well