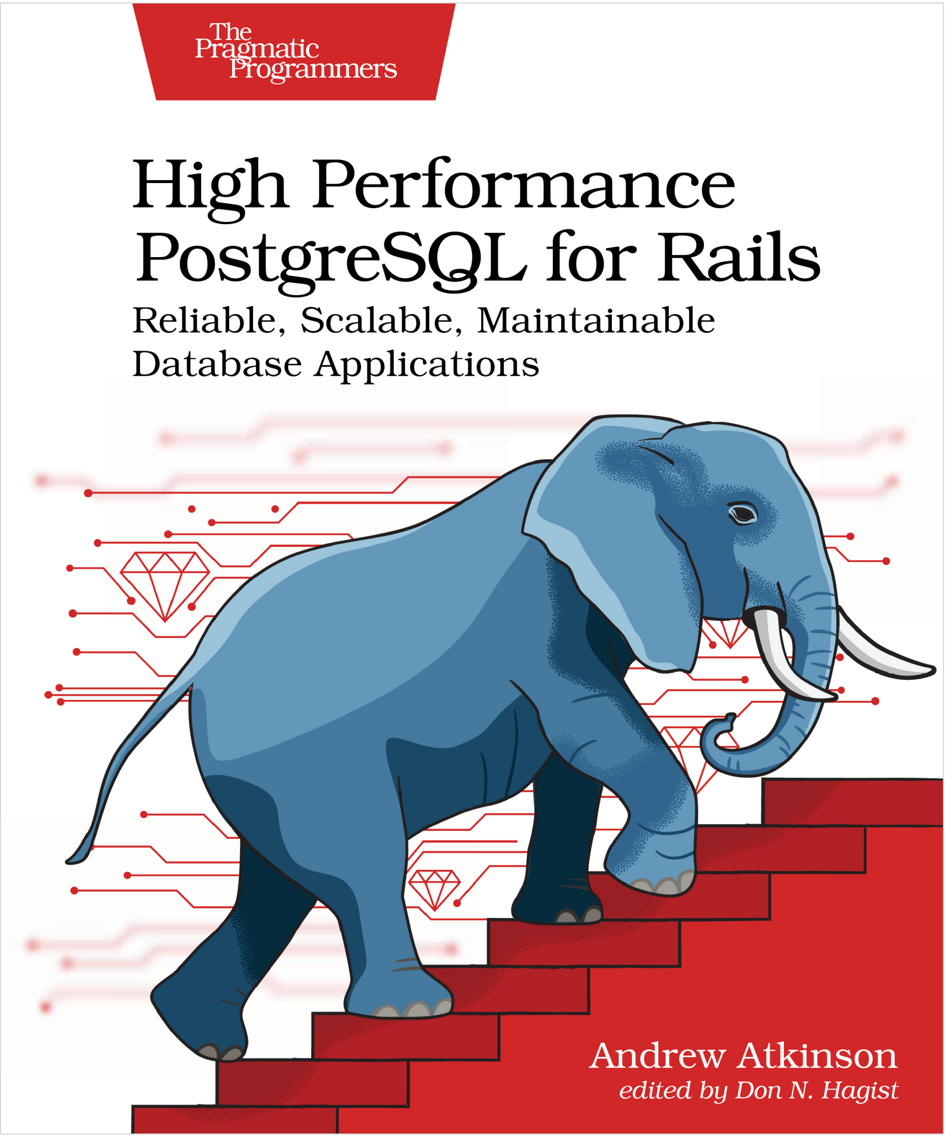

The Pragmatic Programmers

High Performance PostgreSQL for Rails

Reliable, Scalable, Maintainable Database Applications

Andrew Atkinson

edited by Don N. Hagist

This extract shows the online version of this title, and may contain features (such as hyperlinks and colors) that are not available in the print version.

For more information, or to purchase a paperback or ebook copy, please visit https://www.pragprog.com.

Preface

If you're looking to expand your knowledge and build new skills with PostgreSQL and Ruby on Rails, you've come to the right place. This book is loaded with practical examples and exercises and is inspired by real-world challenges. Completing them will help you build the skills needed to squeeze out[1] all the performance possible.

Maybe the use of your application has grown, and your team is facing performance problems. Maybe your data has ballooned in size, and you're wondering how to preserve good performance despite high growth. You may be wondering how to make changes to indexes, parameters, or your schema design to improve overall performance while making your operations more predictable. You'll get the context you need while building confidence in your skills.

While the traditional database administrator (DBA) role has declined in popularity,[2] there are more database-backed web applications than ever. Relational databases are the system of record,[3] simultaneously safeguarding irreplaceable company data while serving it to thousands of clients.

Businesses expect that database systems store data safely and reliably and are scalable and cost-efficient. Companies, from small startups to huge corporations, choose PostgreSQL. This means it can grow with your organization, but there isn't a one-size-fits-all configuration. A lot of pressure is placed on databases as data and query volumes grow. That's where you come in! Maybe you'll be the hero who can fix problems and design solutions that sidestep them entirely.

The skills you'll develop as you complete exercises are portable and practical. You'll use modern versions of PostgreSQL and Ruby on Rails, which have the freshest features. You'll start from an existing Rails application and begin

[1] https://blog.danslimmon.com/2023/08/11/squeeze-the-hell-out-of-the-system-you-have

[2] https://builtin.com/software-engineering-perspectives/database-administrator

[3] https://en.wikipedia.org/wiki/System_of_record

iterating on it, learning how to make database operations faster, more reliable, and more resilient.

The Rails application and databases are the test lab that you'll use to develop your skills. You'll work with millions of rows of data to help simulate a production environment with slow queries and maintenance operations. Besides core PostgreSQL and Ruby on Rails functionality, you'll add more than 40 Ruby gems and PostgreSQL extensions from the open source ecosystem.

Who Is This Book For?

Topics are chosen for back-end engineers looking to deepen their knowledge and skills with PostgreSQL and Ruby on Rails. Whether you're working on consumer scale Internet applications or enterprise B2B Software as a Service (SaaS), scaling PostgreSQL and Rails application codebases is mission critical for the success of your business. Your team expects to build on and operate these technologies while managing huge data growth amidst shifting business priorities.

If your job responsibilities or career aspirations include any of the following descriptions, this book will help you grow:

- Ruby on Rails application developers sharpening their database skills
- PostgreSQL database administrators (DBAs) learning Ruby on Rails
- Infrastructure and DevOps engineers learning Ruby on Rails
- Database reliability engineers (DBRE) learning how to perform sharding, replication, and table partitioning
- Developers coming from other relational databases
- Web developers with experience in frameworks like Laravel[4] or Django,[5] learning Ruby on Rails
- Data engineers learning PostgreSQL, replication, and change data capture (CDC)

Two major categories of database work are transactional work, also called *online transaction processing* (OLTP), and analytical, also called *online analytical processing* (OLAP).

[4] https://laravel.com
[5] https://www.djangoproject.com

Here, we're focused on OLTP. OLTP has short-duration queries running in high volume, with high concurrency. The result set sizes are small. Wikipedia defines online transaction processing as follows:[6]

> Such systems are expected to respond to user requests and process them in real time.

Now that you've seen a bit about what is covered, what's not covered?

What's Not Covered in This Book?

This book is designed for intermediate professional programmers and isn't an introduction to relational database systems, the Ruby on Rails framework, or the Ruby programming language. Readers are expected to have familiarity with PostgreSQL, Ruby on Rails, or comparable technologies. Readers are also assumed to be familiar with SQL basics, shell scripting, and Docker containers.

If you're new to PostgreSQL, consider *PostgreSQL: Up and Running*[7] as an introduction to PostgreSQL.

If you're new to Ruby on Rails, consider books like *Agile Web Development with Rails 7*[8] for a broader overview of Ruby on Rails beyond the more narrow focus here on Active Record.

If you're new to the Ruby language, consider *Programming Ruby 3.3 (5th Edition)*.[9]

The internals of PostgreSQL, like storage and data layout, are outside the scope of this book, although high-level information is provided. Concepts like multiversion concurrency control (MVCC) and transaction isolation levels are covered at a high level. Readers are directed to PostgreSQL documentation for further information. The online document, "The Internals of PostgreSQL for Database Administrators and System Developers,"[10] is highly regarded.

While PostgreSQL administration and authorization concepts are outside the scope, there are scripted configurations that readers use and can learn from. Concepts like roles and users, grants, and privileges are not covered deeply. The post, "PostgreSQL Basics: Roles and Privileges,"[11] is a great starting point. Concepts like row-level security (RLS) and policies[12] are not covered.

[6] https://en.wikipedia.org/wiki/Online_transaction_processing
[7] https://www.oreilly.com/library/view/postgresql-up-and/9781491963401/
[8] https://pragprog.com/titles/rails7/agile-web-development-with-rails-7/
[9] https://pragprog.com/titles/ruby5/programming-ruby-3-3-5th-edition/
[10] https://www.interdb.jp/pg/
[11] https://www.red-gate.com/simple-talk/databases/postgresql/postgresql-basics-roles-and-privileges/
[12] https://www.postgresql.org/docs/current/ddl-rowsecurity.html

In later chapters, readers set up multiple types of replication using multiple PostgreSQL instances with Docker. Concepts like high availability (HA) and automatic failover are mentioned but not covered deeply. Consider the book *Database Reliability Engineering*[13] for more on those topics.

Collecting and testing database backups is *critical*. Make sure to collect them automatically and periodically verify their integrity. With that said, backups are outside the scope. They fall more into the realm of infrastructure DBA topics, as compared with application-focused topics. Backups also depend on how PostgreSQL has been deployed. For general information, check out the PostgreSQL documentation chapter, "Backup and Restore."[14]

Finally, we're focused here on the traditional "single primary" (or "single writer") form of PostgreSQL. This is the default open source community distribution. Other types of distributed PostgreSQL variations are outside the scope.

Extra software that readers will work with is open source, permissively licensed, and noncommercial. This allows readers to install all the software locally for hands-on exercises.

Ruby on Rails Skills Are in Demand

In the post, "Big Transitions in the Tech Industry" from Hired's "2023 State of Software Engineers Survey,"[15] Ruby on Rails was the most sought-after skill.

> Ruby on Rails surfaced as the most in-demand skill for software engineering roles, creating 1.64x more interview requests for the developers proficient in it.

Combining Ruby on Rails and PostgreSQL is a great way to build database-backed web applications. Rails Guides even have a page dedicated to "Active Record with PostgreSQL."[16]

Active Record continues to add support for PostgreSQL capabilities in new versions. For example, generated columns from PostgreSQL 12 was added to Active Record as virtual columns.[17] Common table expressions (CTE) were added to Active Record in version 7.1.

Native support for multiple databases[18] in Active Record was added in version 6, providing developers with powerful ways to leverage multiple PostgreSQL

[13] https://www.oreilly.com/library/view/database-reliability-engineering/9781491925935/
[14] https://www.postgresql.org/docs/current/backup.html
[15] https://hired.com/state-of-software-engineers/2023/
[16] https://guides.rubyonrails.org/active_record_postgresql.html
[17] https://blog.saeloun.com/2022/01/25/rails-7-postgres-support-for-generated-columns.html
[18] https://guides.rubyonrails.org/active_record_multiple_databases.html

instances to scale out their workload. With multiple databases, PostgreSQL instances can support read and write splitting, application-level sharding, and horizontal sharding. Readers will implement these capabilities with PostgreSQL and Active Record in upcoming chapters.

PostgreSQL Is a Popular Award Winner

You've made a great choice to invest your time learning PostgreSQL. PostgreSQL is praised for reliability, SQL standards conformance,[19] and having a high bar for features and documentation quality.

PostgreSQL was the #1 most popular database in use with Rails according to the "2022 Ruby on Rails Community Survey,"[20] with data spanning 13 years, from 2009 through 2022.

PostgreSQL was a three-time #1 Winner in 2017, 2018, and 2020 in "DB-Engines Ranking,"[21] an annual ranking of the most popular databases in the world.

In 2022, PostgreSQL was the third most popular database behind Snowflake[22] and Google BigQuery,[23] which are both primarily OLAP databases.

The 2022 "Stack Overflow Developer Survey"[24] gathered input from nearly 50,000 professional developers. When the developers were asked what database they used most, PostgreSQL was the #1 answer.

Both technologies have vibrant, international communities, helping improve their core features and documentation year after year. Ruby on Rails 7.1 had more than 800 contributors,[25] while the PostgreSQL 16 Release Notes[26] acknowledged more than 350 people.

PostgreSQL and Ruby on Rails are vibrant today and will be for years to come!

[19] https://www.postgresql.org/docs/current/features.html
[20] https://rails-hosting.com/2022/#databases
[21] https://db-engines.com/en/ranking
[22] https://www.snowflake.com
[23] https://cloud.google.com/bigquery
[24] https://survey.stackoverflow.co/2022/#most-popular-technologies-database-prof
[25] https://rubyonrails.org/2023/10/5/Rails-7-1-0-has-been-released
[26] https://www.postgresql.org/docs/current/release-16.html#RELEASE-16-ACKNOWLEDGEMENTS