

Extracted from:

Seven Databases in Seven Weeks, Second Edition

A Guide to Modern Databases and the NoSQL Movement

This PDF file contains pages extracted from *Seven Databases in Seven Weeks, Second Edition*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2018 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

Raleigh, North Carolina

Seven Databases in Seven Weeks

Second Edition

A Guide to Modern
Databases and the
NoSQL Movement

Luc Perkins
with Eric Redmond and Jim R. Wilson

Series editor: *Bruce A. Tate*
Development editor: *Jacquelyn Carter*



Seven Databases in Seven Weeks, Second Edition

A Guide to Modern Databases and the NoSQL Movement

Luc Perkins
with Eric Redmond
and Jim R. Wilson

The Pragmatic Bookshelf

Raleigh, North Carolina



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at <https://pragprog.com>.

The team that produced this book includes:

Publisher: Andy Hunt
VP of Operations: Janet Furlow
Managing Editor: Brian MacDonald
Supervising Editor: Jacquelyn Carter
Series Editor: Bruce A. Tate
Copy Editor: Nancy Rapoport
Indexing: Potomac Indexing, LLC
Layout: Gilson Graphics

For sales, volume licensing, and support, please contact support@pragprog.com.

For international rights, please contact rights@pragprog.com.

Copyright © 2018 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.
ISBN-13: 978-1-68050-253-4
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—April 2018

Preface

If we use oil extraction as a metaphor for understanding data in the contemporary world, then databases flat-out constitute—or are deeply intertwined with—all aspects of the extraction chain, from the fields to the refineries, drills, and pumps. If you want to harness the potential of data—which has perhaps become as vital to our way of life as oil—then you need to understand databases because they are quite simply the most important piece of modern data equipment.

Databases are tools, a means to an end. But like any complex tool, databases also harbor their own stories and embody their own ways of looking at the world. The better you understand databases, the more capable you'll be of tapping into the ever-growing corpus of data at our disposal. That enhanced understanding could lead to anything from undertaking fun side projects to embarking on a career change or starting your own data-driven company.

Why a NoSQL Book

What exactly does the term *NoSQL* even mean? Which types of systems does the term include? How will NoSQL impact the practice of making great software? These were questions we wanted to answer—as much for ourselves as for others.

Looking back more than a half-decade later, the rise of NoSQL isn't so much buzzworthy as it is an accepted fact. You can still read plenty of articles about NoSQL technologies on HackerNews, TechCrunch, or even WIRED, but the tenor of those articles has changed from starry-eyed prophecy (“NoSQL will change everything!”) to more standard reporting (“check out this new Redis feature!”). NoSQL is now a mainstay and a steadily maturing one at that.

But don't write a eulogy for relational databases—the “SQL” in “NoSQL”—just yet. Although NoSQL databases *have* gained significant traction in the technological landscape, it's still far too early to declare “traditional” relational database models as dead or even dying. From the release of Google's BigQuery

and Spanner to continued rapid development of MySQL, PostgreSQL, and others, relational databases are showing no signs of slowing down. NoSQL hasn't killed SQL; instead, the galaxy of uses for data has expanded, and both paradigms continue to grow and evolve to keep up with the demand.

So read this book as a guide to powerful, compelling databases with similar worldviews—not as a guide to the “new” way of doing things or as a nail in the coffin of SQL. We're writing a second edition so that a new generation of data engineers, application developers, and others can get a high-level understanding and deep dive into specific databases in one place.

Why Seven Databases

This book's format originally came to us when we read Bruce Tate's exemplary *Seven Languages in Seven Weeks [Tat10]* many years ago. That book's style of progressively introducing languages struck a chord with us. We felt teaching databases in the same manner would provide a smooth medium for tackling some of these tough NoSQL questions while also creating conceptual bridges across chapters.

What's in This Book

This book is aimed at experienced application developers, data engineers, tech enthusiasts, and others who are seeking a well-rounded understanding of the modern database landscape. Prior database experience is not strictly required, but it helps.

After a brief introduction, this book tackles a series of seven databases chapter by chapter. The databases were chosen to span five different database genres or styles, which are discussed in [Chapter 1, Introduction, on page ?](#). In order, the databases covered are PostgreSQL, Apache HBase, MongoDB, Apache CouchDB, Neo4J, DynamoDB, and Redis.

Each chapter is designed to be taken as a long weekend's worth of work, split up into three days. Each day ends with exercises that expand on the topics and concepts just introduced, and each chapter culminates in a wrap-up discussion that summarizes the good and bad points about the database. You may choose to move a little faster or slower, but it's important to grasp each day's concepts before continuing. We've tried to craft examples that explore each database's distinguishing features. To really understand what these databases have to offer, you have to spend some time using them, and that means rolling up your sleeves and doing some work.

Although you may be tempted to skip chapters, we designed this book to be read linearly. Some concepts, such as mapreduce, are introduced in depth in earlier chapters and then skimmed over in later ones. The goal of this book is to attain a solid understanding of the modern database field, so we recommend you read them all.

What This Book Is Not

Before reading this book, you should know what it *won't* cover.

This Is Not an Installation Guide

Installing the databases in this book is sometimes easy, sometimes a bit of a challenge, and sometimes downright frustrating. For some databases, you'll be able to use stock packages or tools such as apt-get (on many Linux systems) or Homebrew (if you're a Mac OS user) and for others you may need to compile from source. We'll point out some useful tips here and there, but by and large you're on your own. Cutting out installation steps allows us to pack in more useful examples and a discussion of concepts, which is what you really came for anyway, right?

Administration Manual? We Think Not

In addition to installation, this book will also not cover everything you'd find in an administration manual. Each of these databases offers myriad options, settings, switches, and configuration details, most of which are well covered online in each database's official documentation and on forums such as StackOverflow. We're much more interested in teaching you useful concepts and providing full immersion than we are in focusing on the day-to-day operations. Though the characteristics of the databases can change based on operational settings—and we discuss these characteristics in some chapters—we won't be able to go into all the nitty-gritty details of all possible configurations. There simply isn't space!

A Note to Windows Users

This book is inherently about choices, predominantly open source software on *nix platforms. Microsoft environments tend to strive for an integrated environment, which limits many choices to a smaller predefined set. As such, the databases we cover are open source and are developed by (and largely *for*) users of *nix systems. This is not our own bias so much as a reflection of the current state of affairs.

Consequently, our tutorial-esque examples are presumed to be run in a *nix shell. If you run Windows and want to give it a try anyway, we recommend setting up Bash on Windows¹ or Cygwin² to give you the best shot at success. You may also want to consider running a Linux virtual machine.

Code Examples and Conventions

This book contains code in a variety of languages. In part, this is a consequence of the databases that we cover. We've attempted to limit our choice of languages to Ruby/JRuby and JavaScript. We prefer command-line tools to scripts, but we will introduce other languages to get the job done—such as PL/pgSQL (Postgres) and Cypher (Neo4J). We'll also explore writing some server-side JavaScript applications with Node.js.

Except where noted, code listings are provided in full, usually ready to be executed at your leisure. Samples and snippets are syntax highlighted according to the rules of the language involved. Shell commands are prefixed by \$ for *nix shells or by a different token for database-specific shells (such as > in MongoDB).

Credits

Apache, Apache HBase, Apache CouchDB, HBase, CouchDB, and the HBase and CouchDB logos are trademarks of The Apache Software Foundation. Used with permission. No endorsement by The Apache Software Foundation is implied by the use of these marks.

Online Resources

The Pragmatic Bookshelf's page for this book³ is a great resource. There you'll find downloads for all the source code presented in this book. You'll also find feedback tools such as a community forum and an errata submission form where you can recommend changes to future releases of the book.

Thanks for coming along with us on this journey through the modern database landscape.

Luc Perkins, Eric Redmond, and Jim R. Wilson

April 2018

1. <https://msdn.microsoft.com/en-us/commandline/wsl/about>
2. <http://www.cygwin.com/>
3. <http://pragprog.com/book/pwrdata/seven-databases-in-seven-weeks>