

Extracted from:

SQL Antipatterns, Volume 1

Avoiding the Pitfalls of Database Programming

This PDF file contains pages extracted from *SQL Antipatterns, Volume 1*, 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 © 2022 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

The
Pragmatic
Programmers

SQL Antipatterns, Volume 1

Avoiding the Pitfalls of Database
Programming



Bill Karwin
edited by Jacquelyn Carter

SQL Antipatterns, Volume 1

Avoiding the Pitfalls of Database Programming

Bill Karwin

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.

For our complete catalog of hands-on, practical, and Pragmatic content for software developers, please visit <https://pragprog.com>.

The team that produced this book includes:

CEO: Dave Rankin

COO: Janet Furlow

Managing Editor: Tammy Coron

Development Editor: Jacquelyn Carter

Copy Editor: Karen Galle

Indexing: Potomac Indexing, LLC

Layout: Gilson Graphics

Founders: Andy Hunt and Dave Thomas

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

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

Copyright © 2022 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.

ISBN-13: 978-1-68050-898-7

Encoded using the finest acid-free high-entropy binary digits.

Book version: P1.0—November 2022

To my wife Jan, my best supporter.

Introduction

This book is about SQL, the popular language programmers use for data. Specifically, it's about the worst ways to use SQL.

Everyone makes mistakes, but experts try to learn from their mistakes, turning them into opportunities to improve their skills. You'll become a better software developer by studying the most common errors made by other developers, and how to fix them.

My first encounter with SQL involved turning down a job. I had just finished college, and I had been approached by a manager who worked at the university and knew me through campus activities. He had an idea for his own software startup company, and he wanted to develop a database management system portable between various UNIX platforms using shell scripts. He needed a programmer like me to write the code to recognize and execute a limited version of the SQL language.

He said, "I don't need to support the full language—that would be too much work. I need only one SQL statement: SELECT."

I hadn't been taught SQL in any of my college classes. But I had developed complete applications in shell, and I knew a little about parsers and domain-specific languages. So, I thought about taking the job. How hard could it be to parse a single statement of a specialized language like SQL?

As I started to read about SQL, I noticed immediately that this was a different sort of language from those I had used before. To call SELECT only one statement in that language is like calling an engine only one part of an automobile. Both statements are true enough, but they certainly belie the complexity and depth of their subjects. To support execution of that single SQL statement, I realized I would have to develop a fully functional relational database management system and query engine. I could tell it would take years for an experienced developer to do that, and I was still too junior to take on a big project like that by myself.

I declined this opportunity to code an SQL parser and RDBMS engine in shell script. The manager had underestimated the scope of his project, perhaps because he didn't understand what an RDBMS does.

My early experience with SQL seems to be a common one. Most developers are self-taught in SQL, learning it out of self-defense when they find themselves working on a project that requires it. Whether the person is a hobbyist or a professional programmer or an accomplished researcher with a PhD, SQL is a language that tends to be used by programmers without training. This leads to many common mistakes being made over and over.

Notes on the Second Edition

Since writing the first edition of this book, I've worked as an SQL consultant, trainer, developer, and database administrator. I've visited dozens of companies in all sorts of fields of business, all of which use SQL. I've talked with other expert developers and database administrators at conferences and meetups, hearing about their successes and failures.

All software developers still work with data, no matter what language or system they use, and SQL is still the dominant language used for data. Since the software development field in general keeps growing, the number of software developers using SQL is always increasing, even as alternative database technologies are also gaining popularity.

The second edition of *SQL Antipatterns* is updated with the latest observations about common mistakes of SQL and data-driven application development. Feedback about the first edition has been addressed. Internet resources are updated to reference current sites and the latest information.

A number of all-new “mini-antipatterns” appear in between the existing chapters. These briefly cover all-new types of blunders and describe “quick wins” you can use to avoid them.

The code examples are updated to be compatible with the latest versions of MySQL and Python, the most popular open source database and dynamic programming language in today's technology market.

Who This Book Is For

SQL Antipatterns is for any software developer who needs to use SQL—which is virtually all software developers. It doesn't matter whether you're a beginner or a seasoned professional. People at all levels of experience will benefit from the subjects in this book.

You may have read a reference on SQL syntax. Now you know all the clauses of a SELECT statement, and you can get some work done. Gradually, you increase your SQL skills by reading code, books, and blogs. In spite of this, it's hard to tell if you're learning best practices, or another way to paint yourself into a corner.

You may find some topics in *SQL Antipatterns* that are well known to you. You'll see new ways of looking at the problems, even if you're already aware of the solutions. It's good to reinforce your good practices by reviewing widespread programmer misconceptions, and the reasons we avoid them.

It's not uncommon for the relationship between developers and database administrators to be contentious. If you're a DBA, this book can help you explain good practices to the software developers you work with and the consequences of straying from that path.

About This Book

There are plenty of books and internet resources for the basics of the SQL language, so this book assumes the reader has learned enough SQL syntax already to use the language and get some work done.

Performance, scalability, and optimization are important topics for database-driven applications, especially on the web, but it's not the main focus of this book. Recommended books specifically about performance and scalability include [*SQL Performance Tuning \[GP03\]*](#), [*High Performance MySQL, 4th Edition \[BT21\]*](#), [*Efficient MySQL Performance \[Nic21\]*](#), and [*Effective MySQL Optimizing SQL Statements \[Bra11\]*](#).

Every brand of SQL database product has its own tools and commands, but this book is not a command reference or a collection of recipes.

Data access frameworks and object-relational mapping libraries are helpful tools, but these aren't the focus of this book either.

Database administration and operation tasks such as server sizing, capacity planning, installation and configuration, monitoring, backups, log analysis, and security are important and deserve a book of their own.

Conventions

The following sections describe some conventions in this book.

Typography

SQL keywords are formatted in all-capitals and in a monospaced font to make them stand out from the text, as in `SELECT`.

SQL tables, also in a monospaced font, are spelled with a capital for the initial letter of each word in the table name, as in `Accounts` or `BugsProducts`. SQL columns, also in a monospaced font, are spelled in lowercase, and words are separated by underscores, as in `account_name`.

Literal strings are formatted in italics, as in *bill@example.com*.

Terminology

SQL is pronounced “ess-cue-ell,” not “see-quell.” Both usages are common enough that everyone will know what you mean. In this book, the former is used. You will read phrases like “*an* SQL query,” not “*a* SQL query.”

In this book, the plural of *index* is *indexes*. In other contexts, it may be pluralized as *indices*. Both are correct according to most dictionaries.

In SQL, the terms *query* and *statement* are somewhat interchangeable, being any complete SQL command that you can execute. In this book, for the sake of clarity, *query* refers to `SELECT` statements and *statement* for all others.

Online Resources

The examples and source code shown in this book are under the source code link on the Pragmatic Bookshelf website.¹ You can also report any errors or suggestions using the errata link on the same site.

If you like this book and it serves you well, I hope that you will let others know about it—your reviews really do help. Tweets and posts are a great way to help spread the word. You can find me on Twitter at @billkarwin, or you can tweet @pragprog directly.

Bill Karwin

October 2022

1. <https://pragprog.com/book/bksap1>