



SQL Antipatterns

Bill Karwin

[Download now](#)

[Read Online ➔](#)

SQL Antipatterns

Bill Karwin

SQL Antipatterns Bill Karwin

Bill Karwin has helped thousands of people write better SQL and build stronger relational databases. Now he's sharing his collection of antipatterns--the most common errors he's identified in those thousands of requests for help.

Most developers aren't SQL experts, and most of the SQL that gets used is inefficient, hard to maintain, and sometimes just plain wrong. This book shows you all the common mistakes, and then leads you through the best fixes. What's more, it shows you what's *behind* these fixes, so you'll learn a lot about relational databases along the way.

Each chapter in this book helps you identify, explain, and correct a unique and dangerous antipattern. The four parts of the book group the anti?patterns in terms of logical database design, physical database design, queries, and application development.

The chances are good that your application's database layer already contains problems such as *Index Shotgun*, *Keyless Entry*, *Fear of the Unknown*, and *Spaghetti Query*. This book will help you and your team find them. Even better, it will also show you how to fix them, and how to avoid these and other problems in the future.

SQL Antipatterns gives you a rare glimpse into an SQL expert's playbook. Now you can stamp out these common database errors once and for all.

Whatever platform or programming language you use, whether you're a junior programmer or a Ph.D., *SQL Antipatterns* will show you how to design and build databases, how to write better database queries, and how to integrate SQL programming with your application like an expert. You'll also learn the best and most current technology for full-text search, how to design code that is resistant to SQL injection attacks, and other techniques for success.

SQL Antipatterns Details

Date : Published July 2nd 2010 by Pragmatic Bookshelf (first published June 25th 2010)

ISBN : 9781934356555

Author : Bill Karwin

Format : Paperback 328 pages

Genre : Computer Science, Programming, Science, Technology, Technical, Software

 [Download SQL Antipatterns ...pdf](#)

 [Read Online SQL Antipatterns ...pdf](#)

[Download and Read Free Online SQL Antipatterns Bill Karwin](#)

From Reader Review SQL Antipatterns for online ebook

Ji says

This book was recommended by some programmer friends of mine, and proved to be a good read. It is not about TSQL thought. The book is mainly about some basic ideas that people should have in mind when designing database models. It's a fast read.

Todor says

While I enjoyed reading this book, I found most of the example trivial from the RoR best practices and conventions, I would recommend it if you want to level up your sql game.

Johnny Graber says

In the world of SQL you can do many mistakes. This will not only result in slow queries but in hours wasted to figure out what you try to achieve. Bill Karwin collected the most uses anti-patterns and explains how you can do a better job. His solutions are not only shorter to write and faster to execute, but they are simpler to maintain as well.

The catalogue of anti-patterns follows a bit the one from Refactoring: Improving the Design of Existing CodeRefactoring in which every (anti-) pattern is described following the same structure. An especially helpful point is the legitimate use of the anti-pattern. Then even if it mostly is harmful, there are situations when that pattern is the better solution. By explicitly talking about the good parts you can easily figure out if you are in that situation or if you better follow the advised approach.

Michael Gaudet says

I thought this book was ok. All the “anti patterns” are good to be aware of, but I’d say they vary in likelihood. Some are mistakes I would expect only a real newbie to make. Others, like SQL injection, are useful sections that try to dispel myths believed by experienced users even now.

My edition was published in 2010, so I shouldn’t be surprised that the non-SQL code examples (such as for dynamic SQL) are in PHP. I also can’t fault the book’s age for having such a small section on relational mappers.

All in all, a decent read for a new SQL user, but if you have any experience with SQL you might want to skip this one for a newer reference.

Phil Eaton says

This enjoyable read covers many anti-patterns, clearly and with compelling reason, that I've both been thrust into and put myself into. It is the first book I've read on database design and optimization after 5 years of professional software development. It is a great jumping point for the topic with the numerous other books it cites.

Anton Antonov says

The book that every modern developer should read. As we all know, databases are an essential part of software development. With the rise of NoSQL databases, the relational ones start to get less attention than they deserve, but that all comes right whenever you read about the next big project/company deciding to store their relational data in a non-relational NoSQL solution *cough* *cough* *cough* MongoDB followers *cough* *cough*.

And here we are. You stopped to read this review and possibly read the sample or go straight to read the book. What you need to know is that this is something worth reading and taking your time. I, contrary to my 'give it time' advice, did not spend more than a week with this book. I justify 1 week as enough time to understand the issues since I faced most of them in 3-4 projects that had databases falling in at least 2-3 anti-patterns.

A big plus for the reader is the format used throughout the book - chapter intro, objective, anti pattern 1 ... anti pattern N, how to identify the anti-pattern based on questions/discussions in your team, valid use of the anti-pattern (if any) and then solution 1 ... solution N. Really, really easy to read format that easily transforms the book into a reference book when you need to confirm an anti-pattern in your project and search for a solution asap. Good job!

What's most important to read and understand thoroughly is the "Logical Database Design Anti-patterns" chapters. I think this is where people make most of their bad decisions in RDB usage.

The "Polymorphic associations" anti-patterns really resonated with me. That's something that I've been trying to address a lot in projects and teams I participate in. Mostly successful, I convince them to keep the cleverness and trickiness out of the DB design and instead keep it simple without relying on polymorphic associations if possible.

Other reviewers said that the "Application development Anti-patterns" section is a hit or miss in regards of content quality and relevance. I think otherwise. While it chews on the all-known topics nowadays of - 'readable passwords' and 'sql injections', the 'pseudokey neat-freak' isn't exactly the most popular topic, is it?

Chapter 24 "Diplomatic Immunity" is a worth to be part of Robert C. Martin's "Clean Coder". Perfect scenarios that people face in a corporate environment of washed up responsibilities and diplomatic immunity.

Chapter 25 "Magic Beans" is a bit weird. It starts off as an attack to ActiveRecord without focus and then transforms into something more reasonable. I just want to warn everyone reading it, to give it a chance and read the whole of it. I did not like it until I reached the solutions part and it all made sense. The author

definitely could use a different tone and language to get his point across in 25. Either way, the solutions is mature and introducing an abstraction above the ActiveRecord objects usage in controllers is reasonable. If you're a Rails/Django developer you might scratch your head a bit where would these lie. Rails especially has the mysterious controller helpers that no one really understands, but the always-safe-bet is put your services-like logic in... the services folder!

Tech choice preferences: I wouldn't really justify the author's decision for picking out MySQL and primally PHP for the book, but these are just my two cents. The book would shine even brighter in the solutions part if a way better relational database as PostgreSQL and a programming language with "class" (as in style, wit, manners, good intentions) were used. Either way, this does not affect my scoring.

Code(SQL) examples: I'm giving great importance to the code examples quality in every tech book I read. If you can't follow the examples - the book is only 1/3 as useful as they're selling it to you. "SQL Antipatterns" does not have any code examples problems. I was pleasantly surprised to see all the code sections in the book having a code example file path that you can instantly find if you download the source code zip from the site. That way I could easily translate all the code into PostgreSQL-compliant SQL.

The source code is not hosted on GitHub but nothing is ever perfect.

The final verdict is: read this book sooner than you could mess up your database design. :)

Ronald says

I'm a software developer who gets paid mostly for working on the back end portion of solutions and I've worked in both small and large companies. I am not an SQL guru and I am more than happy to let somebody else review and fine tune any SQL that the solution uses. If I'm lucky, I'll be working in an organization large enough to have full-time SQL experts who can help me out. More times than not, however, the development team has to craft the SQL ourselves. As of late, I've been letting Hibernate do most of the heavy lifting for me, trusting that it will generate reasonable SQL and keep me out of hot water. I fully understand that, like most tools, Hibernate is only as useful as the developer's understanding of it which is why I try and read up on the ins and outs of Hibernate as much as I can. To that end, I figured I'd pick up a copy of *SQL Antipatterns: Avoiding the Pitfalls of Database Programming* to help me understand if I was asking Hibernate to do something silly and I'm really glad that I did.

Some of the anti-patterns I've seen before, like not using constraints or using a column to mean multiple things, but there are many I haven't seen before but should now be able to spot. I enjoyed the discussions around security and can appreciate that storing passwords in the database is a bad idea (I've seen that more than once) and have a better understanding on how to protect against SQL injection attacks. The section on application development was a pleasant surprise to me because it talks about testing and possible migration strategies. I also enjoyed the comparison of the Active Record pattern versus the Repository pattern as it convinced me that the decision to use Repository on my current project was a reasonable one.

All in all, I loved the book and recommend that anybody who has to deal with a relational store pick up a copy and give it a read. It isn't a very long book but you get the sense that the advice given came from some hard learned lessons on the job.

Javier says

If in one way or another you are involved with databases you should read this book. Developers and DBAs alike would benefit from this read. It is very well written, great content and easy to follow. Also, no superfluous content in this one.

Majed Daas says

???? ?????? ??? ?? ?????? ?????? ?????????? ?????? ?????? ? ?????? ?????? ?????? ??? ?????? ?????? ?????? ?????? ?????? ?????? ?????? ?????? ?????? ?????? ?????? ?????? .

Matias Fernandez says

As someone struggling to design my website's database, this book hit the spot. I couldn't have imagined all the things I had wrong. I'm so glad I found this gem.

Ispiriants Volodymyr says

It's a very good book for the people, who just learned SQL. I regret that I did not read it before, because it will resolve some of issues during my work.

Jason says

First, this is based on B6.0 printing, version 2010-4-1.

I was hoping to get a little more out of this book. I bought it because I saw the excerpt from the chapter "Naive Trees" and the statement "most developers aren't SQL experts, and most of the SQL that gets used is inefficient, hard to maintain, and sometimes just plain wrong," on PragProg Bookshelf. Not wanting to create poor SQL, I knew I needed this book.

I struggled for a long time with a database that had a large tree structure, so I'm always looking for better ways of dealing with trees in databases. After a lot of work, I finally ended up with what basically equates to the Enumerated Path pattern. I wish I'd had this book a few years ago when I started that project; it would have saved me a *ton* of time.

I think the chapters are well written, and seem to cover a lot of issues a developer might encounter in his first few years of development. The author uses good, real world, examples written in clear language.

If you're a developer struggling with database problems, I'd say, pick this book up. If you've been developing for upwards of 10 years, you can probably skip it, hopefully, you'll already know most everything he covers.

Horia says

It's a pretty good book on SQL. It is aimed at people with beginner-medium SQL experience. Beginners will not really understand what's happening there. And if you've written the database interface layer for 3-5 apps, it might be partially interesting for you. If you're advanced with SQL, you already know these patterns.

David says

The book was easy to read and well explained. However, I haven't seen very many of his anti-patterns in practice with actual projects that I've worked on in the past. Maybe I'm just lucky to work on things that were well built or I'm more experienced than the target audience for the book. However, I didn't learn too much that I didn't already know.

Lisa says

With many relational database systems available in the market, I find there is not much need to design a RDBMS product to the level this book gives. Working with data analytics, I found part 3 to be the most useful with Query Antipatterns. Majority of the issues are addressed with proper training so I can see this book would be good for those who are self learners who may not have many opportunities for formal training. The book is written in a format that gives the antipattern as an example, then provides the best solution. Decent book overall, but I didn't get much out of the content or saw examples that shouldn't happen unless proper practices weren't followed.
