

*Optimization, Backups,
Replication, and more*

2nd Edition
Covers Version 5.1



High Performance MySQL

O'REILLY®

*Baron Schwartz, Peter Zaitsev,
Vadim Tkachenko, Jeremy D. Zawodny,
Arjen Lentz & Derek J. Balling*

High Performance MySQL, 2nd Edition

Baron Schwartz

Peter Zaitsev

Vadim Tkachenko

Jeremy D. Zawodny

Arjen Lentz

Derek J. Balling

Editor

Andy Oram

Copyright © 2008 Peter Zaitsev, Vadim Tkachenko, Arjen Lentz, Baron Schwartz, Peter Zaitsev, Vadim Tkachenko, Arjen Lentz, Baron Schwartz

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

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *High Performance MySQL*, the image of a sparrow hawk, and related trade dress are trademarks of O'Reilly Media, Inc.

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 O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

O'REILLY[®]

O'Reilly Media

Foreword

I have known Peter, Vadim, and Arjen a long time and have witnessed their long history of both using MySQL for their own projects and tuning it for a lot of different high-profile customers. On his side, Baron has written client software that enhances the usability of MySQL.

The authors' backgrounds are clearly reflected in their complete reworking in this second edition of *High Performance MySQL: Optimizations, Replication, Backups, and More*. It's not just a book that tells you how to optimize your work to use MySQL better than ever before. The authors have done considerable extra work, carrying out and publishing benchmark results to prove their points. This will give you, the reader, a lot of valuable insight into MySQL's inner workings that you can't easily find in any other book. In turn, that will allow you to avoid a lot of mistakes in the future that can lead to suboptimal performance.

I recommend this book both to new users of MySQL who have played with the server a little and now are ready to write their first real applications, and to experienced users who already have well-tuned MySQL-based applications but need to get "a little more" out of them.

—Michael Widenius

March 2008

Preface

We had several goals in mind for this book. Many of them were derived from thinking about that mythical perfect MySQL book that none of us had read but that we kept looking for on bookstore shelves. Others came from a lot of experience helping other users put MySQL to work in their environments.

We wanted a book that wasn't just a SQL primer. We wanted a book with a title that didn't start or end in some arbitrary time frame ("... in Thirty Days," "Seven Days To a Better ... ") and didn't talk down to the reader. Most of all, we wanted a book that would help you take your skills to the next level and build fast, reliable systems with MySQL—one that would answer questions like "How can I set up a cluster of MySQL servers capable of handling millions upon millions of queries and ensure that things keep running even if a couple of the servers die?"

We decided to write a book that focused not just on the needs of the MySQL application developer but also on the rigorous demands of the MySQL administrator, who needs to keep the system up and running no matter what the programmers or users may throw at the server. Having said that, we assume that you are already relatively experienced with MySQL and, ideally, have read an introductory book on it. We also assume some experience with general system administration, networking, and Unix-like operating systems.

This revised and expanded second edition includes deeper coverage of all the topics in the first edition and many new topics as well. This is partly a response to the changes that have taken place since the book was first published: MySQL is a much larger and more complex piece of software now. Just as importantly, its popularity has exploded. The MySQL community has grown much larger, and big corporations are now adopting MySQL for their mission-critical applications. Since the first edition, MySQL has become recognized as ready for the enterprise. ^[1] People are also using it more and more in applications that are exposed to the Internet, where downtime and other problems cannot be concealed or tolerated.

As a result, this second edition has a slightly different focus than the first edition. We emphasize reliability and correctness just as much as performance, in part because we have used MySQL ourselves for applications where significant amounts of money are riding on the database server. We also have deep experience in web applications, where MySQL has become very popular. The second

edition speaks to the expanded world of MySQL, which didn't exist in the same way when the first edition was written.

How This Book Is Organized

We fit a lot of complicated topics into this book. Here, we explain how we put them together in an order that makes them easier to learn.

A Broad Overview

[Chapter 1](#), *MySQL Architecture*, is dedicated to the basics—things you'll need to be familiar with before you dig in deeply. You need to understand how MySQL is organized before you'll be able to use it effectively. This chapter explains MySQL's architecture and key facts about its storage engines. It helps you get up to speed if you aren't familiar with some of the fundamentals of a relational database, including transactions. This chapter will also be useful if this book is your introduction to MySQL but you're already familiar with another database, such as Oracle.

Building a Solid Foundation

The next four chapters cover material you'll find yourself referencing over and over as you use MySQL.

[Chapter 2](#), *Finding Bottlenecks: Benchmarking and Profiling*, discusses the basics of benchmarking and profiling—that is, determining what sort of workload your server can handle, how fast it can perform certain tasks, and so on. You'll want to benchmark your application both before and after any major change, so you can judge how effective your changes are. What seems to be a positive change may turn out to be a negative one under real-world stress, and you'll never know what's really causing poor performance unless you measure it accurately.

In [Chapter 3](#), *Schema Optimization and Indexing*, we cover the various nuances of data types, table design, and indexes. A well-designed schema helps MySQL perform much better, and many of the things we discuss in later chapters hinge on how well your application puts MySQL's indexes to work. A firm understanding of indexes and how to use them well is essential for using MySQL effectively, so you'll probably find yourself returning to this chapter repeatedly.

[Chapter 4](#), *Query Performance Optimization*, explains how MySQL executes queries and how you can take advantage of its query optimizer's strengths. Having a firm grasp of how the query

optimizer works will do wonders for your queries and will help you understand indexes better. (Indexing and query optimization are sort of a chicken-and-egg problem; reading [Chapter 3](#) again after you read [Chapter 4](#) might be useful.) This chapter also presents specific examples of virtually all common classes of queries, illustrating where MySQL does a good job and how to transform queries into forms that take advantage of its strengths.

Up to this point, we've covered the basic topics that apply to any database: tables, indexes, data, and queries. [Chapter 5](#), *Advanced MySQL Features*, goes beyond the basics and shows you how MySQL's advanced features work. We examine the query cache, stored procedures, triggers, character sets, and more. MySQL's implementation of these features is different from other databases, and a good understanding of them can open up new opportunities for performance gains that you might not have thought about otherwise.

Tuning Your Application

The next two chapters discuss how to make changes to improve your MySQL-based application's performance.

In [Chapter 6](#), *Optimizing Server Settings*, we discuss how you can tune MySQL to make the most of your hardware and to work as well as possible for your specific application. [Chapter 7](#), *Operating System and Hardware Optimization*, explains how to get the most out of your operating system and hardware. We also suggest hardware configurations that may provide better performance for larger-scale applications.

Scaling Upward After Making Changes

One server isn't always enough. In [Chapter 8](#), *Replication*, we discuss replication—that is, getting your data copied automatically to multiple servers. When combined with the scaling, load-balancing, and high availability lessons in [Chapter 9](#), *Scaling and High Availability*, this will provide you with the groundwork for scaling your applications as large as you need them to be.

An application that runs on a large-scale MySQL backend often provides significant opportunities for optimization in the application itself. There are better and worse ways to design large applications. While this isn't the primary focus of the book, we don't want you to spend all your time concentrating on MySQL. [Chapter 10](#), *Application-*

Level Optimization, will help you discover the low-hanging fruit in your overall architecture, especially if it's a web application.

Making Your Application Reliable

The best-designed, most scalable architecture in the world is no good if it can't survive power outages, malicious attacks, application bugs or programmer mistakes, and other disasters.

In [Chapter 11](#), *Backup and Recovery*, we discuss various backup and recovery strategies for your MySQL databases. These strategies will help minimize your downtime in the event of inevitable hardware failure and ensure that your data survives such catastrophes.

[Chapter 12](#), *Security*, provides you with a firm grasp of some of the security issues involved in running a MySQL server. More importantly, we offer many suggestions to allow you to prevent outside parties from harming the servers you've spent all this time trying to configure and optimize. We explain some of the rarely explored areas of database security, showing both the benefits and performance impacts of various practices. Usually, in terms of performance, it pays to keep security policies simple.

Miscellaneous Useful Topics

In the last few chapters and the book's appendixes, we delve into several topics that either don't "fit" in any of the earlier chapters or are referenced often enough in multiple chapters that they deserve a bit of special attention.

[Chapter 13](#), *MySQL Server Status* shows you how to inspect your MySQL server. Knowing how to get status information from the server is important; knowing what that information means is even more important. We cover `SHOW INNODB STATUS` in particular detail, because it provides deep insight into the operations of the InnoDB transactional storage engine.

[Chapter 14](#), *Tools for High Performance* covers tools you can use to manage MySQL more efficiently. These include monitoring and analysis tools, tools that help you write queries, and so on. This chapter covers the Maatkit tools Baron created, which can enhance MySQL's functionality and make your life as a database administrator easier. It also demonstrates a program called *innotop*, which Baron wrote as an easy-to-use interface to what your MySQL server is presently doing. It functions much like the Unix *top*

command and can be invaluable at all phases of the tuning process to monitor what's happening inside MySQL and its storage engines.

[Appendix A](#), *Transferring Large Files*, shows you how to copy very large files from place to place efficiently—a must if you are going to manage large volumes of data. [Appendix B](#), *Using EXPLAIN*, shows you how to really use and understand the all-important EXPLAIN command. [Appendix C](#), *Using Sphinx with MySQL*, is an introduction to Sphinx, a high-performance full-text indexing system that can complement MySQL's own abilities. And finally, [Appendix D](#), *Debugging Locks*, shows you how to decipher what's going on when queries are requesting locks that interfere with each other.

[1] We think this phrase is mostly marketing fluff, but it seems to convey a sense of importance to a lot of people.

Software Versions and Availability

MySQL is a moving target. In the years since Jeremy wrote the outline for the first edition of this book, numerous releases of MySQL have appeared. MySQL 4.1 and 5.0 were available only as alpha versions when the first edition went to press, but these versions have now been in production for years, and they are the backbone of many of today's large online applications. As we completed this second edition, MySQL 5.1 and 6.0 were the bleeding edge instead. (MySQL 5.1 is a release candidate, and 6.0 is alpha.)

We didn't rely on one single version of MySQL for this book. Instead, we drew on our extensive collective knowledge of MySQL in the real world. The core of the book is focused on MySQL 5.0, because that's what we consider the "current" version. Most of our examples assume you're running some reasonably mature version of MySQL 5.0, such as MySQL 5.0.40 or newer. We have made an effort to note features or functionalities that may not exist in older releases or that may exist only in the upcoming 5.1 series. However, the definitive reference for mapping features to specific versions is the MySQL documentation itself. We expect that you'll find yourself visiting the annotated online documentation (<http://dev.mysql.com/doc/>) from time to time as you read this book.

Another great aspect of MySQL is that it runs on all of today's popular platforms: Mac OS X, Windows, GNU/Linux, Solaris, FreeBSD, you name it! However, we are biased toward GNU/Linux [2] and other Unix-like operating systems. Windows users are likely to encounter some differences. For example, file paths are completely different. We also refer to standard Unix command-line utilities; we assume you know the corresponding commands in Windows. [3]

Perl is the other rough spot when dealing with MySQL on Windows. MySQL comes with several useful utilities that are written in Perl, and certain chapters in this book present example Perl scripts that form the basis of more complex tools you'll build. Maatkit is also written in Perl. However, Perl isn't included with Windows. In order to use these scripts, you'll need to download a Windows version of Perl from ActiveState and install the necessary add-on modules (DBI and DBD::mysql) for MySQL access.

[2] To avoid confusion, we refer to Linux when we are writing about the kernel, and GNU/Linux when we are writing about the whole operating system infrastructure that supports applications.

[3] You can get Windows-compatible versions of Unix utilities at <http://unxutils.sourceforge.net> or <http://gnuwin32.sourceforge.net>.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Used for new terms, URLs, email addresses, usernames, hostnames, filenames, file extensions, pathnames, directories, and Unix commands and utilities.

Constant width

Indicates elements of code, configuration options, database and table names, variables and their values, functions, modules, the contents of files, or the output from commands.

Constant width bold

Shows commands or other text that should be typed literally by the user. Also used for emphasis in command output.

Constant width italic

Shows text that should be replaced with user-supplied values.

Tip

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

Warning

This icon indicates a warning or caution.

Using Code Examples

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 don't 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 doesn't 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 doesn't require permission. Incorporating a significant amount of example code from this book into your product's documentation *does* require permission.

Examples are maintained on the site <http://www.highperfmysql.com> and will be updated there from time to time. We cannot commit, however, to updating and testing the code for every minor release of MySQL.

We appreciate, but don't require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*High Performance MySQL: Optimization, Backups, Replication, and More, Second Edition*, by Baron Schwartz et al. Copyright 2008 O'Reilly Media, Inc., 9780596101718."

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.

Safari® Books Online

When you see a Safari® Books Online icon on the cover of your favorite technology book, that means the book is available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books. It's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at <http://safari.oreilly.com>.

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)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

<http://www.oreilly.com/catalog/9780596101718/>

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our web site at:

<http://www.oreilly.com>

You can also get in touch with the authors directly. Baron's weblog is at <http://www.xaprb.com>.

Peter and Vadim maintain two weblogs, the well-established and popular <http://www.mysqlperformanceblog.com> and the more recent <http://www.webscalingblog.com>. You can find the web site for their company, Percona, at <http://www.percona.com>.

Arjen's company, OpenQuery, has a web site at <http://openquery.com.au>. Arjen also maintains a weblog at <http://arjen-lentz.livejournal.com> and a personal site at <http://lentz.com.au>.

Acknowledgments for the Second Edition

Sphinx developer Andrew Aksyonoff wrote [Appendix C](#), *Using Sphinx with MySQL*. We'd like to thank him first for his in-depth discussion.

We have received invaluable help from many people while writing this book. It's impossible to list everyone who gave us help—we really owe thanks to the entire MySQL community and everyone at MySQL AB. However, here's a list of people who contributed directly, with apologies if we've missed anyone: Tobias Asplund, Igor Babaev, Pascal Borghino, Roland Bouman, Ronald Bradford, Mark Callaghan, Jeremy Cole, Britt Crawford and the HiveDB Project, Vasil Dimov, Harrison Fisk, Florian Haas, Dmitri Joukovski and Zmanda (thanks for the diagram explaining LVM snapshots), Alan Kasindorf, Sheeri Kritzer Cabral, Marko Makela, Giuseppe Maxia, Paul McCullagh, B. Keith Murphy, Dhiren Patel, Sergey Petrunia, Alexander Rubin, Paul Tuckfield, Heikki Tuuri, and Michael "Monty" Widenius.

A special thanks to Andy Oram and Isabel Kunkle, our editor and assistant editor at O'Reilly, and to Rachel Wheeler, the copyeditor. Thanks also to the rest of the O'Reilly staff.

From Baron

I would like to thank my wife Lynn Rainville and our dog Carbon. If you've written a book, I'm sure you know how grateful I am to them. I also owe a huge debt of gratitude to Alan Rimm-Kaufman and my colleagues at the Rimm-Kaufman Group for their support and encouragement during this project. Thanks to Peter, Vadim, and Arjen for giving me the opportunity to make this dream come true. And thanks to Jeremy and Derek for breaking the trail for us.

From Peter

I've been doing MySQL performance and scaling presentations, training, and consulting for years, and I've always wanted to reach a wider audience, so I was very excited when Andy Oram approached me to work on this book. I have not written a book before, so I wasn't prepared for how much time and effort it required. We first started talking about updating the first edition to cover recent versions of MySQL, but we wanted to add so much material that we ended up rewriting most of the book.

This book is truly a team effort. Because I was very busy bootstrapping Percona, Vadim's and my consulting company, and because English is not my first language, we all had different roles. I provided the outline and technical content, then I reviewed the material, revising and extending it as we wrote. When Arjen (the former head of the MySQL documentation team) joined the project, we began to fill out the outline. Things really started to roll once we brought in Baron, who can write high-quality book content at insane speeds. Vadim was a great help with in-depth MySQL source code checks and when we needed to back our claims with benchmarks and other research.

As we worked on the book, we found more and more areas we wanted to explore in more detail. Many of the book's topics, such as replication, query optimization, InnoDB, architecture, and design could easily fill their own books, so we had to stop somewhere and leave some material for a possible future edition or for our blogs, presentations, and articles.

We got great help from our reviewers, who are the top MySQL experts in the world, from both inside and outside of MySQL AB. These include MySQL's founder, Michael Widenius; InnoDB's founder, Heikki Tuuri; Igor Babaev, the head of the MySQL optimizer team; and many others.

I would also like to thank my wife, Katya Zaytseva, and my children, Ivan and Nadezhda, for allowing me to spend time on the book that should have been Family Time. I'm also grateful to Percona's employees for handling things when I disappeared to work on the book, and of course to Andy Oram and O'Reilly for making things happen.

From Vadim

I would like to thank Peter, who I am excited to have worked with on this book and look forward to working with on other projects; Baron, who was instrumental in getting this book done; and Arjen, who was a lot of fun to work with. Thanks also to our editor Andy Oram, who had enough patience to work with us; the MySQL team that created great software; and our clients who provide me the opportunities to fine tune my MySQL understanding. And finally a special thank you to my wife, Valerie, and our sons, Myroslav and Timur, who always support me and help me to move forward.

From Arjen

I would like to thank Andy for his wisdom, guidance, and patience. Thanks to Baron for hopping on the second edition train while it was already in motion, and to Peter and Vadim for solid background information and benchmarks. Thanks also to Jeremy and Derek for the foundation with the first edition; as you wrote in my copy, Derek: "Keep 'em honest, that's all I ask."

Also thanks to all my former colleagues (and present friends) at MySQL AB, where I acquired most of what I know about the topic; and in this context a special mention for Monty, whom I continue to regard as the proud parent of MySQL, even though his company now lives on as part of Sun Microsystems. I would also like to thank everyone else in the global MySQL community.

And last but not least, thanks to my daughter Phoebe, who at this stage in her young life does not care about this thing called "MySQL," nor indeed has she any idea which of The Wiggles it might refer to! For some, ignorance is truly bliss, and they provide us with a refreshing perspective on what is really important in life; for the rest of you, may you find this book a useful addition on your reference bookshelf. And don't forget your life.

Acknowledgments for the First Edition

A book like this doesn't come into being without help from literally dozens of people. Without their assistance, the book you hold in your hands would probably still be a bunch of sticky notes on the sides of our monitors. This is the part of the book where we get to say whatever we like about the folks who helped us out, and we don't have to worry about music playing in the background telling us to shut up and go away, as you might see on TV during an awards show.

We couldn't have completed this project without the constant prodding, begging, pleading, and support from our editor, Andy Oram. If there is one person most responsible for the book in your hands, it's Andy. We really do appreciate the weekly nag sessions.

Andy isn't alone, though. At O'Reilly there are a bunch of other folks who had some part in getting those sticky notes converted to a cohesive book that you'd be willing to read, so we also have to thank the production, illustration, and marketing folks for helping to pull this book together. And, of course, thanks to Tim O'Reilly for his continued commitment to producing some of the industry's finest documentation for popular open source software.

Finally, we'd both like to give a big thanks to the folks who agreed to look over the various drafts of the book and tell us all the things we were doing wrong: our reviewers. They spent part of their 2003 holiday break looking over roughly formatted versions of this text, full of typos, misleading statements, and outright mathematical errors. In no particular order, thanks to Brian "Krow" Aker, Mark "JDBC" Matthews, Jeremy "the other Jeremy" Cole, Mike "VBMySQL.com" Hillyer, Raymond "Rainman" De Roo, Jeffrey "Regex Master" Friedl, Jason DeHaan, Dan Nelson, Steve "Unix Wiz" Friedl, and, last but not least, Kasia "Unix Girl" Trapszo.

From Jeremy

I would again like to thank Andy for agreeing to take on this project and for continually beating on us for more chapter material. Derek's help was essential for getting the last 20-30% of the book completed so that we wouldn't miss yet another target date. Thanks for agreeing to come on board late in the process and deal with my sporadic bursts of productivity, and for handling the XML grunt work, [Chapter 10](#), [Appendix C](#), and all the other stuff I threw your