



Craig S. Mullins

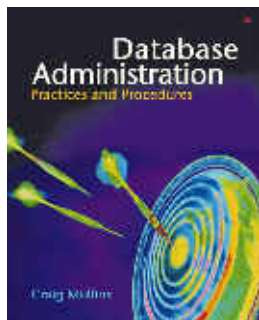
July 2007

[Return to Home Page](#)



The DBA Corner

by Craig S. Mullins



A Good Book on Physical Database Design

Recently I received an intriguing new book on physical database design called, appropriately enough, [Physical Database Design](#) (Morgan Kaufmann, ISBN 0-12-369389-6). This book fills a gap in the marketplace that has needed to be filled for some time now. Oh, there are a lot of good books on each individual



DBMS and there are some very good books on logical database design and data modeling. But until now, there has not been a good book that provides a comprehensive treatment of the physical side of database design from a DBMS agnostic perspective.

Actually, the last book that I can recall to tackle the subject was [Handbook of Relational Database Design](#) by Fleming and von Halle (Addison-Wesley, ISBN 0-201-11434-8) which was published way back in 1989. Now that book was quite good, but it is a bit long in the tooth, so the publication of a new book on physical database design has been long overdue.

[Physical Database Design](#) is co-authored by three database design experts: Sam Lightstone, Toby Teorey, and Tom Nadeau. Lightstone is a Senior Technical Staff Member and Development Manager with IBM's DB2 product development team. Teorey is a professor emeritus in the Electrical Engineering and Computer Science Department and Director of Academic Programs in the College of Engineering at the University of Michigan, Ann Arbor. And Nadeau is the founder of Aladdin Software and works in the area of data and text mining. So it is no surprise that the book has a solid technical foundation.

The book offers comprehensive coverage of how to design the physical structures and environment for the most popular database management systems. After reading [Physical Database Design](#) you will come away with a better understanding of how the choices you make during physical design affect the performance of your systems. The specific examples, guidelines, and best and worst practices included by the authors are instructive and enlightening, as well as being helpful across the spectrum of popular DBMSs. Examples and code illustrating the major concepts of physical database design are depicted for IBM DB2, Oracle, Microsoft SQL Server, and even sometimes for Informix. It might

have been useful to also cover MySQL, but that is not a major criticism as the three most popular DBMSs are nicely addressed by the authors.

So what types of issues are covered within the book? Well, of course, it offers guidance on how to create the objects needed to support a database; tables, indexes, etc. The book also deals with the nuances of setting up more complicated database structures such as partitioning and clustering. And performance issues are covered in some depth as well; in fact, two of the first four chapters are on indexing methods and issues.

One of the more interesting sections of the book is the chapter on automated physical database design. Over the past few years the major DBMS vendors have been adding more autonomic features to their products and the authors address the autonomic physical design aspects quite nicely. These features include the IBM DB2 Design Advisor, the Microsoft SQL Server Database Tuning Advisor, and Oracle's SQL Access Advisor.

If you are interested in how database systems interact with hardware the chapter titled "Down to the Metal: Server Resources and Topology" will be instructive. It discusses CPU architectures, client/server architectures, SMP and NUMA, server clusters, storage and RAID, resource balancing, and availability issues. This is an area where many DBAs lack in-depth knowledge, so this chapter should be welcome reading.

Another very nice feature of the book is the "Tips and Insights for Database Professionals" that end each chapter. These highlighted sections cover the primary ideas that were presented in that chapter in the form of tips you can follow to improve your database design.

As good as this book is though, it is not perfect. Unfortunately, the authors make the cardinal mistake of sometimes using the words “always” and “never.” It is almost never a good idea to use “always” or “never” when talking about database design and performance. For example, the authors offer the following advice on page 28: “Indexing should always be used for access to a small number of rows for queries.” Now this advice may sound reasonable, and it would be if you just remove the word “always.” For example, if you only have a small total number of rows in a table then just scanning the table is likely to be more efficient than going through an index because that would add I/O.

Putting small quibbles like that aside, [Physical Database Design](#) is a well-written and researched book that should prove useful to any DBA or developer looking to improve their skills in creating efficient and effective relational database implementation. Consider adding it to your library.

From [Database Trends and Applications](#), July 2007.

© 2007 Craig S. Mullins, All rights reserved.

[Home](#).