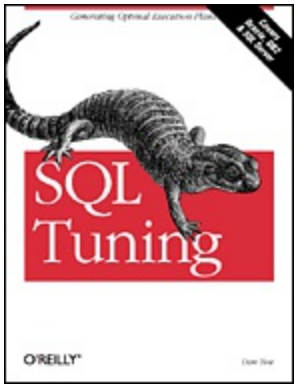


- Table of Contents
- Index
- Reviews
- Examples
- Reader Reviews
- Errata
- Academic

SQL Tuning

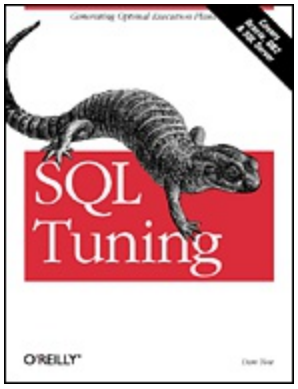
By Dan Tow

Start Reading

Publisher : O'Reilly
Pub Date : November 2003
ISBN : 0-596-00573-3
Pages : 336

SQL Tuning author Dan Tow outlines a timesaving method he's developed for finding the optimum execution plan--rapidly and systematically--regardless of the complexity of the SQL or the database platform being used. You'll learn how to understand and control SQL execution plans and how to diagram SQL queries to deduce the best execution plan for a query. Key chapters in the book includes exercises to reinforce the concepts you've learned. *SQL Tuning* concludes by addressing special concerns and unique solutions to

"unsolvable" problems.



- Table of Contents
- Index
- Reviews
- Examples
- Reader Reviews
- Errata
- Academic

SQL Tuning

By Dan Tow

Start Reading

Publisher : O'Reilly
Pub Date : November 2003
ISBN : 0-596-00573-3
Pages : 336

Copyright

Copyright © 2004 O'Reilly & Associates, Inc.

Printed in the United States of America.

Published by O'Reilly & Associates, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly & Associates 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 & Associates, 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 & Associates, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps. The association between the image of salamander and the topic of SQL tuning is a trademark of O'Reilly & Associates, Inc.

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.

Dedication

To Prarva, and Abe,
for a dream of more,
and life surpassing the dream

Foreword

During my years as a developer, and later as a database administrator, the only performance tuning area in which I developed any expertise and had any level of success was that of tuning individual SQL statements. Because of that background, I was very interested when Dan Tow came to me with his idea for a book on SQL tuning.

The problem with SQL tuning, at least from my own perspective, is that it was often reasonably easy to pinpoint badly performing SQL statements that needed to be tuned, and it was reasonably easy to determine the execution plan currently being used for those badly performing statements, but then came the hard problem: finding a better execution plan, to make an offending SQL statement execute faster. Many is the time I've stared at a poorly performing SQL statement, reviewing its poorly performing execution plan, and wondered just what I should do next. For that matter, was there even any room for improvement? Perhaps the execution plan that performed so badly from a user's perspective was, in fact, the best possible execution plan. Perhaps I was wasting my time trying to guess at a better one.

There, I used that word *guess*, and that's at the heart of what sometimes made tuning SQL statements a frustrating activity. It all came down to my looking at a SQL statement and trying to guess at a better plan. Of course, I'd attempt to factor in my experience, my intuition, and my knowledge of the data being queried, and I'd pull out tips and tricks that I'd read about in books and magazine articles, but in the end I'd make a guess, try a new plan, make a guess, try a new plan, and so forth. I'd stop when one of two things happened:

- I got lucky and guessed a plan that was a good-enough improvement over the old plan to satisfy my client.
- I ran out of ideas.

It always bothered me when I ran out of ideas, because I never knew for sure whether the current plan really was optimal or whether I was just too thick-headed to intuit a better-performing execution plan than the current one. To be honest, I was always pretty hard on myself, chalking up any failure to improve a SQL statement as a personal inadequacy.

Dan doesn't guess. Let me boldface that and underline it. He doesn't execute an iterative guess-loop like I did, trying one idea after another in the hope of stumbling across an improvement. Instead, Dan uses an innovative and

mathematically based diagramming method to derive the optimal, or near-optimal, execution plan for a SQL statement. Then he puts that plan into effect. And that's it. There's no guesswork, and there's no uncertainty as to whether further improvement is possible.

At first, I was skeptical of Dan's approach. But the more I read his chapters, the more I began to see the logic behind his method. Math doesn't lie, and experience is a good indicator. Dan has over 10 years of tuning experience, and he's been very successful using the method described in this book.

There are three legs that any SQL tuning effort needs to stand on. You need to know how to identify a badly performing SQL statement. You then need to be able to see the execution plan used for that statement. Finally, you need to somehow come up with an improved plan. I'm convinced that Dan's method represents a viable third leg on which your SQL tuning efforts can rest. Read his book, apply his method, and save yourself hours of wondering what to do next.

Jonathan Gennick, author, editor, Oracle DBA