



Official Publication of the Northern California Oracle Users Group

# NoCOUG

J O U R N A L

Vol. 20, No. 3 · AUGUST 2006

\$15

## Refresh Your Career With NoCOUG

### Feuerthoughts

*An interview with Steven Feuerstein. See page 4.*

### Does the Optimizer Need a Hint?

*Six well-known Oracles answer our questions. See page 9.*

### Lies, Damn Lies, and SQL!

*Solve the puzzle and win a prize! See page 15.*

***Big discount on Steven Feuerstein seminar! See page 4. Much more inside . . .***

# Cost-Based Oracle—Fundamentals

A Book Review by Brian Hitchcock

## Summary

**Overall review:** Excellent, one of the best technical books I've seen.

**Target audience:** Experienced DBAs with specific interest in the details of how the Oracle Cost-Based Optimizer (CBO) works. This is not a book for beginners.

### Would you recommend this book to others?

Absolutely, but only if they are willing to spend considerable time and effort understanding the material.

**Who will get the most from this book?** Readers must have a midlevel understanding of Oracle database and DBA concepts.

**Is this book platform specific?** No.

**Why did I obtain this book?** I saw it on Amazon and wanted to read it. NoCOUG reimbursed me for the cost.

## Overall review

This book covers many aspects of the Oracle Cost-Based Optimizer, how it works, how it gets confused, and why upgrades can cause so much trouble. The process is not simple and requires covering a lot of material to explain. The reasons why upgrading can cause performance problems is a subject that keeps coming up. While an upgrade might only cause the optimizer to make a new (and very poor) choice for one query, that one query can interfere with all the others and cause a significant slowdown for the whole database.

This is not a book for a first-time Oracle DBA. It contains an unusually high amount of real information and it requires concentrated study to get the most out of it. If you aren't already familiar with indexes, execution plans, etc., you won't get very far. Having said that, the author provides excellent summaries at the end of each chapter, and these summaries are a very good place for any-

*This is not a book for beginners.*



one to begin. While I encourage anyone interested in the CBO to read the whole book from start to finish, I think you could get a lot of valuable insights from reading the chapter summaries and then perhaps reading the individual chapters that you find most interesting. Further, while I think junior DBAs could be lost in this book, they could greatly benefit if they started reading and then looked up anything (and there will be lots of things) that they didn't already know about in other sources. This book could be used as an excellent way to become much more expert on the subject as long as the reader is willing to go to other sources for additional information.

This book is also exceptional in that it is well crafted. I don't usually comment on this because most Oracle books are put together in a competent way, but this one is much better than the norm. The author's writing style is very fluid. You don't notice the writing, which means it is great. In most technical books, the author jumps from subject to subject without always completing a thought or point. I didn't see any of that here. This book was well written, edited, and produced. This book was fun to read even while it was at times challenging to absorb all the information. I actually looked forward to reading the next chapter because I wanted to find out what

*Relationships that are obvious to us are not known to the optimizer.*

happened next. The Introduction tells us about files on the CD but there isn't any CD with the book, but I did check that all the code in the book is available for download from websites maintained by the author and the publisher. With the exception of the CD issue, I didn't find any errors in the text, which is unusual and reflects great work from all involved.

### Chapter 1 What Do You Mean by Cost?

How the cost of a SQL statement is computed is reviewed starting with version 8i and going through 10g. This discussion was very interesting and helped me understand why our older databases could generate such poor execution plans. The author covers how the optimization code has evolved and this helped me understand why an 8i to 9i upgrade can cause significant performance problems. This chapter reinforces the importance of testing upgrades on real datasets. The way the Cost-Based Optimizer works changes from version to version, so you must test the new version against a dataset that is the same size as your existing database. The size of your dataset can cause different issues in different versions of Oracle. The

author also provides three good reasons why reality doesn't always meet expectations.

## Chapter 2 Tablescans

Four different strategies are presented for cost-based optimization. I didn't realize how complicated it was. And this is the point—it's easy to criticize the optimizer but when you really look at what has to be done, it makes you appreciate how hard the task is. CPU costing is described and the impact on upgrades explained. The optimizer can apply data for I/O and CPU performance, data that is measured or supplied by you, in the costing process. An example showing how the cost of the same query would be computed for both  $8i$  and  $9i$  is good. One of the best things I got from this chapter is that early versions of the CBO assume every data block visited required a physical read, as if none of the blocks would ever be cached. As the optimizer was improved, it moved away from this assumption and relied more on statistics. Again, a valuable insight into just how arbitrary the optimization process can be. The analysis of partitions and bind variables is great.

## Chapter 3 Single Table Selectivity

More reasons that bind variables can cause trouble are revealed. When bind variables are involved, things can get complicated. The CBO may simply assume 5% cardinality because it doesn't have any other information about the data. Or, if bind variable peeking is used, the CBO could make decisions based on the actual data present when the SQL is first parsed. This may or may not be better than an assumption of 5%, since the CBO only peeks once when first parsing. The author provides several good discussions of the impact of bind variables, specifically "Overusing Bind Variables" (page 38), "Bind Variables and Ranges" (page 52), and "Bind Variable Peeking" (page 54). I recommend studying these sections carefully to gain an appreciation of the complex tradeoffs you have to make when deciding when to use bind variables. There are many well-known (and very good) reasons to use bind variables, but it is also good to understand how they can limit or change the options available to the CBO.

The way that some application vendors decide to "interpret" NULL values is fascinating—another reason optimization can be so difficult. Another great insight is that if columns of data are somehow related (an example of birth dates and zodiac signs is used) the optimizer will probably make poor choices. Relationships that are obvious to us are not known to the optimizer. The author presents many examples with lots of code. It can be hard to keep up but it is worth it. The examples lead you to the conclusion.

## Chapter 4 Simple B-tree Access

Computing the cost of using an index is more complicated than I thought it was. Again, the process of optimization just isn't as easy as we think it should be. Examples show how a compound index may not be better than a simpler index. The reason that an upgrade from  $9i$  to  $10g$  can cause the optimizer to switch from table scan to index and why table scan was better is very good. The discussion of indexes on small tables is also good. There are many reasons that using an index may not be faster depending on the specifics of the data distribution.

## Chapter 5 The Clustering Factor

The clustering factor is a measure of how randomly the data is distributed. The optimizer relies on this factor and if it doesn't accurately reflect your data, the optimizer will make poor choices. This chapter explains how things like reverse key indexes, ASSM, column order, and extra columns all affect Oracle's computation of the clustering factor. I didn't appreciate how much the data distribution affects Oracle's ability to generate a good execution plan.

## Chapter 6 Selectivity Issues

The discussion of how the use of non-numeric data types can cause optimization problems is great. If an application is set up in which dates are stored not as Oracle date data type but as numbers or character strings, the optimizer assumes that there will be a continuous set of data when in reality we have gaps between sets of data. For example, the end of January (20060131) is followed by February first (20060201), but since the dates are stored as numbers or character strings, Oracle assumes there could be data in between these dates. This leads the optimizer to make huge mistakes. The use of NLS makes this worse, and deciding to use some extreme value to represent NULL (December 31 of the year 4000 for a NULL date) is also bad. If you use an application that is trying to be "database independent," you need to watch for these issues. There is a lot to learn from this chapter.

## Chapter 7 Histograms

A histogram is just a picture of your data. How histograms can help the optimization process is covered. They help the optimizer when the data set is non-uniform, but they are expensive so you don't want to use them in all cases. Bind variables can cause histograms to be ignored. Given the cost of maintaining histograms, it's good to study the situations in which they won't be used by the CBO. The author covers this in detail (page 166). Keep an open mind about using and not using bind variables; the tradeoffs are complicated. There are several diagrams in this chapter that really helped me understand how Oracle uses histograms and the many ways they can be misused or ignored.

## Chapter 8 Bitmap Indexes

The CBO doesn't care if the data is scattered or clustered and unless the data is static, bitmap indexes are more trouble than they are worth. Using CPU costing can affect the use of bitmap indexes. This chapter was difficult.

## Chapter 9 Query Transformation

How the CBO can change a query during the costing process changes from version to version, which is one reason that upgrades can cause performance issues. The author describes using various hidden `init.ora` parameters. I don't think this will be particularly useful for most DBAs in production environments. A very interesting example covers NULL and NOT IN and why constraints are valuable. The use of hints is covered as well as the increased maintenance they require and the upgrade risks they create. If you are using third-party software, do you know if the vendor used Hints? I think the author assumes the reader has control of the SQL.

We are advised to generate execution plans for all critical queries before any upgrade. While I agree that this would be a good thing, I don't think most DBAs have the resources to do this or to review all the same execution plans after an upgrade.

### **Chapter 10 Join Cardinality**

I found it very interesting how Oracle "computes" the cardinality of joins, which it uses to determine the best join order. NULL issues come up again and it turns out that the order of the tables in the SQL actually does matter.

### **Chapter 11 Nested Loops**

I am familiar with how nested loops work, but it was interesting to see how the execution plan that Oracle comes up with may not agree with what happens when Oracle actually executes the plan at runtime.

### **Chapter 12 Hash Joins**

This was interesting because I didn't have a clear picture of how a hash join is done and why it can be faster. I also didn't know that you could generate a trace file specifically for hash joins with the 10104 trace flag. A very good read: second only to the next chapter.

### **Chapter 13 Sorting and Merge Joins**

I think this was my favorite chapter. I found it interesting, I got the most information from it, and I think it will be useful to me at work. The author is excellent at communicating ideas, as illustrated by his card-dealing analogy. The description of the three ways that sorting gets done was great. I had no idea that

there are these various stages, and now I can more fully appreciate why sorting to disk is such a bad thing. But it's always a tradeoff. Increasing memory to reduce sorts to disk causes increased CPU usage. There can be situations where you get the best performance with reduced memory and a single pass sort to disk. There are specific boundaries where the sorting changes from multi-pass to disk, to single pass to disk, and finally to all sorting in memory. Adding more memory doesn't change things unless the increased amount of memory crosses one of these thresholds. This chapter makes all of this clear and provides a much more sophisticated explanation of how sorting works. Also good are the description of memory allocation and a concise summary of why most of what you can tinker with in Oracle isn't a good idea for the long term. Highly tuned systems are not robust. All the hours you might spend tuning your 9i environment could be completely lost when you upgrade to 10g. You need to balance the possible immediate benefits of any tuning exercise against the longer-term risk that your tuning will make things worse after the next upgrade. A specific example is covered that demonstrates how the runtime engine does things differently from what you get from the execution plan. Again, remember that after all the time you spend getting the execution plan you want, Oracle may go off and do things differently.

*All the hours you might spend tuning your 9i environment could be lost.*

## Chapter 14 10053 Trace File

---

This chapter is very focused on the specifics of interpreting the trace file for the optimizer. I have only needed to look at a 10053 trace file once, but when I needed to, it was useful. Even if you haven't needed to do this, it is interesting to follow what the CBO is doing step by step as it decides what the cost of each possible execution plan is. Sometimes the process seems very straightforward and other times it seems rather random.

## Appendix A Upgrade Headaches

---

This brings together the many observations made throughout the book about how upgrades can (and will?) cause performance issues.

## Appendix B Optimizer Parameters

---

This lists the `init.ora` parameters that affect performance. Many of these are hidden parameters that I would not change.

## Conclusion

---

If you want to spend the time needed to follow all the material and you want to learn this much about the Oracle Cost-Based Optimizer, this book is nearly ideal. It is very readable and very well written, edited, and produced. I believe the time I put into reading and looking up related subjects was well worth it. I look forward to the other two volumes that the author promises on this subject, and will certainly buy the

*I look forward to the other two volumes that the author promises on the subject.*

trilogy when it becomes available from a major Hollywood studio on DVD. Finally, I also learned a little Latin (*Tempus Fugit*) and that's not happened to me when reading an Oracle book before.

## About the Author

---

Jonathan Lewis has been specializing in Oracle for the last 17 years. He is a renowned international speaker, both for his presentations at public conferences and for his seminars and tutorials. Jonathan is currently a director of the UK Oracle User Group and writes regularly for their *Oracle Scene* magazine. He authored the acclaimed book *Practical Oracle 8i*, and runs the popular [www.jlcomp.demon.co.uk](http://www.jlcomp.demon.co.uk) website.

---

*Brian Hitchcock has worked at Sun Microsystems in Newark, California, for the past 11 years. He is a member of a DBA team that supports 2400+ databases for many different applications at Sun. He frequently handles issues involving tuning, character sets, and Oracle applications. Other interests include Formula One racing, finishing his second Tiffany Wisteria lamp, Springbok puzzles, Märklin model trains, Corel Painter 8, and watching TV (TiVo rules!). Previous book reviews by Brian and his contact information are available at [www.brianhitchcock.net](http://www.brianhitchcock.net).*

Copyright © 2006, Brian Hitchcock