

Official Publication of the Northern California Oracle Users Group

NoCOUG

J O U R N A L

Vol. 22, No. 4 · NOVEMBER 2008

\$15

Volunteer at NoCOUG



Everybody Loves Jonathan

Candid questions and candid answers.

See page 4.

Secret ORACLE

One of the best books our book reviewer has ever read.

See page 10.

Data Quality

The last article in a series by Michael Scofield.

See page 18.

Much more inside . . .

Volunteer at NoCOUG

Every year NoCOUG brings you four great educational conferences and four great issues of the *NoCOUG Journal*. Where in the world can you find such great educational value for only \$80 per year? All this would not be possible if not for all the time and effort contributed by volunteers, from the members of the Board to the speakers at our conferences and the contributors to the *NoCOUG Journal*.

Won't you consider giving back to the organization and to the community by becoming a NoCOUG volunteer? You will find it a very rewarding experience. In the words of Brian Hitchcock, who writes for the *NoCOUG Journal* and makes presentations at our conferences: "Without NoCOUG, I'd just be someone that had some experience with Oracle. With NoCOUG, I'm someone that can present what I have experienced and help the Oracle community."

I hope to see you at our fall conference on November 13 at the CarrAmerica Conference Center in Pleasanton. Please do take the opportunity to speak to one of the Board members about volunteer opportunities. ▲

—Iggy Fernandez, *NoCOUG Journal* Editor

Table of Contents

| | | | |
|--|----|----------------------------------|----|
| President's Message | 3 | ADVERTISERS | |
| Everybody Loves Jonathan..... | 4 | Precise Software Solutions | 9 |
| Jonathan Lewis Training Days | 7 | Roundstone Systems..... | 15 |
| Book Review: Secret Oracle..... | 10 | BEZ Systems | 15 |
| Compliance Corner: CDS..... | 16 | Database Specialists, Inc. | 15 |
| Fundamentals of Data Quality—Part IV | 18 | Princeton Softech | 15 |
| Teamwork at NoCOUG..... | 22 | Electronic Commerce..... | 17 |
| Sponsorship Appreciation..... | 23 | IT Convergence..... | 25 |
| NoCOUG Fall Conference | 24 | Quest Software | 25 |
| Fall Conference Schedule | 28 | Embarcadero Technologies..... | 25 |
| | | Network Appliance..... | 25 |
| | | Confio Software..... | 27 |

Publication Notices and Submission Format

The *NoCOUG Journal* is published four times a year by the Northern California Oracle Users Group (NoCOUG) approximately two weeks prior to the quarterly educational conferences.

Please send your questions, feedback, and submissions to the *NoCOUG Journal* editor at journal@nocoug.org.

The submission deadline for the upcoming February 2009 issue is November 30, 2008. Article submissions should be made in Microsoft Word format via email.

Copyright © 2008 by the Northern California Oracle Users Group except where otherwise indicated.

NoCOUG does not warrant the NoCOUG Journal to be error-free.

NoCOUG BOARD

President

Roger Schrag
Bay Area Go Players Association
www.nocoug.org/contact_us.html?recipient=roger

Vice President

Hanan Hit, DBA InfoPower

Secretary/Treasurer

Jen Hong, Stanford University
hong_jen@yahoo.com

Director of Membership

Joel Rosingana, Independent Consultant
joelros@pacbell.net

Journal Editor

Iggy Fernandez, Database Specialists
iggy_fernandez@hotmail.com

Webmaster

Eric Hutchinson, Independent Consultant
erichutchinson@comcast.net

Vendor Coordinator

Lisa Loper, Independent Consultant

Director of Conference Programming

Randy Samberg
Access Systems Americas, Inc.
rsamberg@sbcglobal.net

Director of Marketing

Naren Nagtode, Franklin Templeton
nagtode@yahoo.com

IOUG Representative

Claudia Zeiler, ECWise Inc.
czeiler@ecwise.com

NoCOUG Staff

Nora Rosingana

Book Reviewer

Brian Hitchcock

Technical Reviewers

Ted Syrett, Danny Chow, Sumit Aneja

ADVERTISING RATES

The *NoCOUG Journal* is published quarterly.

| Size | Per Issue | Per Year |
|--------------|-----------|----------|
| Quarter Page | \$125 | \$400 |
| Half Page | \$250 | \$800 |
| Full Page | \$500 | \$1,600 |
| Inside Cover | \$750 | \$2,400 |

Personnel recruitment ads are not accepted.

journal@nocoug.org

Changing of the Guard!

by Roger Schrag



Roger Schrag

It was more than eight years ago that Vilin Roufchiae, NoCOUG's president at the time, approached me about getting more involved with the users group. At that time I was an occasional speaker at the quarterly conferences, and Vilin invited me to attend a board meeting to see if I might want to take my participation to a new level.

At the time I didn't know what a board of directors was and I had no idea what a board meeting would be like. My nervousness at the first board meeting quickly melted away when everyone was very friendly and welcoming to me. I already knew most of the people at the meeting from the quarterly conferences, and I found the topics of conversation fascinating because I got to see NoCOUG from a totally different perspective.

With that first glimpse behind the scenes of NoCOUG, I quickly realized that by getting more involved with the users group leadership, I could leverage some of my existing strengths—Oracle database expertise and a DBA's eye for details—to gain experience in whole new areas. Volunteering with NoCOUG, it seemed, would give me an opportunity to exercise areas of my brain that had, until then, been dormant.

And so when I was asked if I would come back to the next board meeting, I said, "Sure." And the one after that. And then I started volunteering to help out in various ways. Before I knew it, Vilin asked me if I would like to become a member of NoCOUG's board. I accepted, and that was the start of my service on the NoCOUG board of directors. During these last eight years, I have held a variety of positions ranging from webmaster to president and I've learned a lot that I would never have learned just being an Oracle DBA or application developer.

I've learned a lot about working with people, about how the whole is so much more than the sum of the parts, about how to get things done that are too big to do all by myself. I may not be a social butterfly or the life of every party, but I've learned a lot about how to work effectively with people and get things done together.

Volunteering with NoCOUG helped me round out my professional skill set by augmenting my technical skills with the "soft" skills that I think are so very important. It has also given me so much experience that helps me every day in both my professional and personal life.

Back in early 2000 I never thought that I would go on to mentor future NoCOUG presidents Darrin Swan and Lisa Loper. That I would become a "go-to" person for all sorts of NoCOUG challenges. And that I would instigate so much change within the organization by suggesting and following through on new ideas of my own and helping to implement those of others.

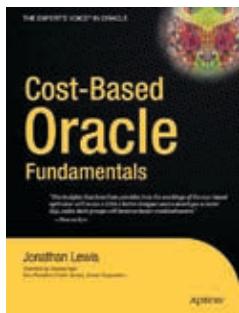
But now I believe it's time for NoCOUG to get some fresh ideas and a shot of new energy. In some ways my ideas are so very "yesterday" and not on the cutting edge. Heck, I don't even have a LinkedIn account or a page on MySpace. I believe that I have done a lot of good things for NoCOUG, but in many ways I've come to the end of my usefulness. I don't want NoCOUG to stagnate, and a changing of the guard in NoCOUG's leadership is a great way to keep things dynamic, current, and fresh.

In December I will be stepping down from the NoCOUG board to make way for new leaders and new ideas. I told the board of my plans back in July, and I'm working closely with NoCOUG's core volunteers to ensure a smooth transition. Great opportunities lie ahead for NoCOUG, and I look forward to watching the organization continue to grow and develop.

And what about me? Today I run a Go players' association and sit on the board of directors of Berkeley's oldest community theatre group. Who knows? In 2009 I may produce my first play!

I hope to see you at the Fall Conference on November 13 at CarrAmerica in Pleasanton. This will be my last NoCOUG conference, and so I will be bidding you farewell. But this is shaping up to be a great conference, and I'm happy to be going out on a high note. We've got Jonathan Lewis from across the pond to give the keynote address and a technical presentation as well. Jonathan is one of NoCOUG's most popular speakers, and he will be joined by a solid lineup of Oracle users sharing real-life experiences, and folks from Oracle Corporation sharing visions for the future. As always, we'll round out the day with vendor exhibits, book raffles, networking opportunities, and more.

See you on November 13th! ▲



Everybody Loves Jonathan



Jonathan Lewis

Jonathan Lewis is well-known to the Oracle community as a consultant, author, and speaker, 20 years of experience in designing, optimizing, and troubleshooting on Oracle database systems. His latest book is Cost-Based Oracle Fundamentals.

What projects are you working on at the present time?



It's always a good idea to start an interview with an easy question. Unfortunately this is a question I can't answer. Most of the work I do doesn't last long enough to be called a "project"—it's rarely more than five consecutive days and usually only two or three days of troubleshooting—although I have had a few "calendar" contracts specifying things like 20 days of call-up over a six-month period.

Sometimes the work is just solving an unusual performance problem; sometimes it has a more strategic flavor like working out a proper set of indexes or designing a cost-effective statistics collection strategy.

On top of that, of course, I have a fairly long-running agreement with Oracle University to fly off somewhere about once each month to do a couple of days under their "Celebrity Seminar" series, and I do a few in-house training sessions each year.



Mogens Norgaard said: "Once upon a time Jonathan Lewis was asked if his book Practical Oracle8i would ever be published in an updated 9i version? His response was to suggest that you put a sticker with a '9' over the '8' on the cover of his book, because the advise, [sic] methods and techniques described in the book were still very valid with Oracle9i."¹ Could we replace the "8" with "10" or "11" instead?

I still find that the answers to a lot of the "strategic" questions that appear on Metalink, or the OTN forum can be found in *Practical Oracle 8i*. And there was a posting on the OTN Oracle forum quite recently asking: "What are the BEST

¹ www.scaleabilities.co.uk/book/NewForward.html.

books for Oracle database architect/designer?" to which someone suggested my book with the comment: "Consider the effective title to be Practical Oracle. The lessons contained therein are still valuable ones to learn. I have not a shred of doubt that this book made me a better DBA."

So, as far as the broad sweep of the information in the book is concerned, the answer to your question is probably yes; as far as particular examples and details are concerned, the answer is no. That makes it very difficult for me to consider writing an "upgrade"—in many ways it would be just the same book with a few relatively minor changes.

Let's say that I'm a junior or intermediate Oracle DBA and my manager offered to buy ten books on Oracle for me. Which books would you recommend?

Ten is a pretty big number, and I think you'd have to consider getting some books which are tailored to the parts of Oracle used by the business (such as RAC, streams, and so on) but I'll restrict myself to a fairly generic response—to the extent that some of the books aren't even specific to Oracle.

For my first choice I'd try to claim a Kindle or Sony e-Reader as the first "book." I haven't used one in anger, and the Kindle isn't even available in the UK, but the brief play I had with the eReader left me keen to buy one. (Unfortunately the demo model in the shop was sealed in a tamper-proof frame and had no real books on it because someone had accidentally deleted all (100) books from its memory—so maybe I should be a little cautious about recommending it.)

The second choice(s) ought to be some Oracle manuals. I know you can get them in HTML or PDF form online (I prefer the PDFs with Acrobat indexing); but for simply reading (rather than reference), there's nothing quite like a real book—and I don't mean a stack of paper from a laser printer. My books of choice would be the *Concepts Guide*, the *Database Admin Guide (Fundamentals)* and the *Performance Tuning Guide*. I might pick the 9i versions, rather than the 10g versions because some of the 10g manuals are getting to the stage where they lean too heavily on "how to use the OEM GUI."

But if you've got a Kindle or eReader, you can probably

load the entire set of PDF files for the last four versions of Oracle, and carry them all around at once! So I'm not going to count any of the Oracle manuals towards my list of ten.

In no particular order, then, here are the other nine:

- Tapio Lahdenmaki and Mike Leach—*Relational Database Index Design and the Optimizers*: The authors are better known in the DB2 world than the Oracle world, but the concepts and understanding needed to design good indexes are the same even when there are variations in vendor implementation. (For more details, check my review on www.amazon.co.uk)
- Dan Tow—*SQL Tuning*: This is the book that does for SQL tuning what Tapio Lahdenmaki and Mike Leach do for understanding indexes. There is a way to think through the problems of data access—that method is going to help you find the best execution plan and work out what indexes your system needs.
- Toon Koppelaars and Lex de Haan: *Applied Mathematics for Database Professionals*. This is a book that many might think too theoretical for a practising DBA—but if you don't have some idea of why the relational model is a good thing, then you can't do your job properly. This book may not be a direct help to you in your work, but it may help you to appreciate what you can achieve with the Oracle software. (For more details, check my review on www.amazon.co.uk.)
- Tom Kyte—*Expert Oracle Database Architecture*: This book contains a wealth of information about a wide range of the Oracle technology and how best to use it. It tells you a lot about what Oracle can do, but also shows you, by the way it's written, how to extend your knowledge into new areas. If I were allowed a longer list, I would also include Tom's *Expert One-on-One Oracle*.
- Jonathan Lewis—*Cost Based Oracle—Fundamentals*: I know that it's my own, but it is the best book you're going to find about how the optimizer works, and if you understand the core features of the optimizer, you'll find it much easier to solve problems with slow SQL. (And if I were allowed a longer list, I probably would include *Practical Oracle 8i*.)
- Christian Antognini—*Troubleshooting Oracle Performance*. This is a recent release that contains a huge amount of information about how to identify and address performance problems.
- Cary Millsap and Jeff Holt—*Optimizing Oracle Performance*. There are a couple of key points about Oracle performance that are rarely stated as clearly as they are in this book. One is the focus on addressing the users' complaints; the other is the impact of concurrency. It won't be casual reading for everyone—but the chapter on queueing theory is a must-read for all DBAs.
- “Recovery”—If you're a DBA, then the most important job you'll ever do is to make sure that you can recover the database quickly after a disaster—and if you're lucky you'll never have to do it. There's probably a good book out there somewhere, but I've never had any need to go looking for it. But if there is one, then every DBA should have a copy and be made to practice. The difficulty with recovery is that the bits you need to cover

well are site dependent, and no one book will tell you all you really need to know about the way you end up doing things for your site.

➤ “X”: One of my pet theories is that people learn more by seeing things go wrong, rather than being told what things look like when they are going well. So all DBAs should be given at least one really bad book about Oracle and told to compare the clarity of writing, quality of explanation, and method of teaching with a good book so that they can learn to recognize suspect material.



*In his latest book, Oracle on VMware, Dr. Bert Scalzo makes a case for “solving” problems with hardware upgrades. What is your opinion about this approach?*²

I think the key sentence in Dr. Scalzo's case is “So, with such cheap hardware, it might be a wiser business bet to throw hardware at some solutions sooner than was done in the past.” And in this sentence, the most important word is “sooner”.

Throwing hardware at a problem has always been an option—and sometimes the correct option—but 20 years ago hardware was much more expensive than it is now, took up more space, used more power, and generated more heat so, in the standard analysis of cost/benefit/risk, correcting the bad code was usually seen to be more cost effective than using more hardware to live with bad code.

Nothing has really changed in the analysis—except the pricing. If you know that more hardware will give you the scalability you need for as long as you need; the installation process is fast enough and safe enough; and the alternative is going to take too long, need too much testing, or introduce too much risk, then the hardware is the right choice.

It has to be said, though that the analysis is not always carried out correctly. Issues of scalability can be deceptive—especially when you are talking about highly concurrent systems, and installing new hardware isn't always risk-free or fast. The main drawback to the hardware solution (if the analysis isn't done properly) is that you can still end up having to fix the code, except the fix is now an emergency that appears unexpectedly at some future time.

² Editor's Note: Here's the full quote from Dr. Scalzo's book: “Person hours cost so much more now than computer hardware even with inexpensive offshore outsourcing. It is now considered a sound business decision these days to throw cheap hardware at problems. It is at least, if not more, cost effective than having the staff [sic] tuned and optimized for the same net effect. Besides, a failed tuning and optimization effort leaves you exactly where you started. At least the hardware upgrade approach results in a faster/better server experiencing the same problem that may still have future value to the business once the fundamental problem is eventually corrected. And, if nothing else, the hardware can be depreciated, whereas the time spent tuning is always just a cost taken off the bottom line. So, with such cheap hardware, it might be a wiser business bet to throw hardware at some solutions sooner than was done in the past. One might go so far as to make an economic principle claim that the opportunity cost of tuning is foregoing cheap upgrades that might fix the issue and also possess intrinsic value. Stated this way, it is a safe bet that is where the business people would vote to spend.”

I've been called out on several occasions to sites who greet me with the complaint: "Everything has been fine for the last X months, nothing has changed, but the system just started running slowly." It usually turns out that what they had been doing was non-scalable—even though it had been protected by hardware for months—and the data, or number of users, had been growing slowly and had finally grown to the point where the system hit a brick wall. I've been lucky so far—solving these problems has usually been a simple structural or code fix.



We love stories. Tell us a performance-tuning story. Tell us two!

This reminds me of a book review by Samuel Johnson that said: "Your manuscript is both good and original. Unfortunately the part that is good is not original, and the part that is original is not good."

I have plenty of performance-tuning stories—but the ones that I can tell you are not interesting, and the ones that are interesting are the ones I can't tell you (I sign a lot of NDAs!).

Strange things do happen occasionally, though, and I try to get permission to talk about them. A couple of years ago I had a client call me because their system was shuddering to a halt for a few seconds fairly randomly about once per week, then slowly recovering over the next couple of minutes. As the system jammed, their application servers would fire in a load of new Oracle connections—jumping from around 600 sessions to about 2,500.

They had a couple of diagnostic traps in place which showed that every time the system halted there was a huge amount of free memory in the shared pool, even though the shared pool normally ran at about 95% full.

Sometimes it's easy to confuse cause and effect. In this case, the problem was simple. The application servers were configured to allow a large number of connections, and one detail of the configuration was that (a) if there were no free connections to service an incoming call, five new connections should be made; (b) if a connection had been idle for five minutes it should be shut down; and (c) there was no lower limit to the number of connections that had to be held open.

Occasionally the number of incoming client calls would spike—and some new connections would be made. But a connection requires something like 22KB of contiguous memory in the shared pool. There were always a few megabytes of free memory in the shared pool, but the memory wasn't always contiguous—so there were random occasions when the heap manager would start to kick some objects out of the shared pool to try and free up 22KB of contiguous memory, and sometimes this caused lots of objects to get kicked out of the shared pool.

This made the shared pool and library cache latches busy—which slowed down the rate at which existing sessions could run their code, which meant the front-end needed to create more connections, which would cause more of the

³ Originally printed in the "Ask the Oracles" column in the February 2008 issue of the *NoCOUG Journal*.

shared pool to be flushed, which would hit the latches, and so on . . .

Thirty seconds later you have 2,500 connections queuing and nothing left in the shared pool—then the system slowly starts to recover, reloading all the objects and execution plans that have been flushed and killing off lots of the now-redundant connections.

The solution was very simple. Multiply the maximum number of processes allowed by 22KB, add a few MB—based on current usage of the reserved pool—and use the result for the `shared_pool_reserved_size` (the 22KB is platform and version specific), so that you don't have to flush the shared pool to try to find a contiguous chunk of 22KB.

As a secondary fix, we changed the application server configuration to create or drop just one connection at a time (interestingly, 10g has three new parameters with names like "`_logout_storm_rate`" that look as if they are relevant to such circumstances), and set a minimum for the number of live connections that was little larger than the typical peak number so that the application server wouldn't have to start and stop lots of connections all at the same time.



Ask the Oracle

Best and Worst Practices³

It's quite hard to say anything sensible or inspiring about a topic like this. Many of the specifics we call "best practices" are extremely obvious common sense, but sometimes circumstances conspire to make it impossible to follow them. Many of the worst practices are things that we should obviously avoid, but again circumstances (often pressures of time) may make them unavoidable.

The worst generic practice I know of, then, is applying a fix to a problem without understanding why that fix might work.

I have a simple approach to problem solving that involves three steps: 1) "What is the problem?"; 2) "Why will my solution fix the problem?"; and 3) "Where am I going to pay for implementing this solution?"

The last question will sometimes tell us that we can't afford to implement the "perfect" solution—perhaps we want to change a heap table to an index-organized table (IOT), but can't because we have too much code that has done something a little exotic with traditional rowids and needs to be rewritten to deal with the "urowid" used for IOTs.

But before we worry about such side effects, overheads, and implementation costs, we need to be confident that our solution really is addressing the root cause; if it isn't, we may spend time and effort implementing a change that seems to fix our problem temporarily—until things go wrong again.

Take a simple example: A query takes an unreasonable amount of time. You check the execution plan and decide the problem might go away if Oracle used a particular index. So you rebuild the index and the query runs much more quickly. Is your job finished?

No doubt the first thing you do is check the execution plan to see that it changed to use the index the way you expected.

NoCOUG presents

Jonathan Lewis Training Days

Register at www.speak-tech.com

TROUBLESHOOTING AND TUNING

November 11, 2008

Troubleshooting or Tuning: What's the difference? What are the strategies? Why tuning is hard but troubleshooting is easy. Key targets, indicators, and mechanisms for producing a well-tuned system on day one. Strategies for dealing with badly performing systems after go-live.

Frequently Occurring Problems: Some of the most commonly occurring issues that affect performance after a system has gone into production. Methods for spotting them, measuring the impact, and dealing with the cost/risk/benefit triangle involved in fixing them. Getting into the habit of pre-emptive analysis and proactive fixing.

Quick Fixes: Methods, workarounds, dirty tricks, and parameters for dealing with classic performance problems when the system is in production. There really aren't many quick fixes that can be applied across the board—each one needs careful examination of costs, risk, and benefits. In this session we consider some of the options that are most likely to be worthwhile.

V\$ and X\$: It's a good idea to be familiar with just a few of the dynamic performance views—and there are a couple of items still hidden away in the X\$ objects that can add a little value. This session will describe the views that are most commonly of use, and explain the meaning of some of the more useful items.

WRITING OPTIMAL SQL

November 12, 2008

Background: We will examine the need for a global viewpoint and a general strategy for minimizing work while recognizing that there will always be a conflict between local and global optimization. A quick review of the use of indexes, including some of the less-common uses and common errors in index design. Closing with the need to be able to investigate and understand the data and business requirements.

Basic Practices: How to approach the task of translating business requirements into SQL. The need for clarity in code, with a suggested set of standards for presentation to improve ease of understanding. An introduction to the essential strategy for writing SQL that gets its result by doing the minimum work, and a model for investigating and re-engineering poorly performing statements.

Sundry Methods: A few examples demonstrating classic problems and their solutions that use some of the less-well-known features of SQL. A brief discussion of the mechanics (and costing) of nested loop and hash joins. Warnings about the abuse of views and the problems of predicates. And a few closing comments about tuning distributed queries.

More Methods: Fixing the code without touching the code—the latest option. Problems and options with knowing internal mechanisms and taking advantage of them. A little time with OEM, and a look at the threats of hints and how to use them properly. ▲

EARLY-BIRD PRICING (UNTIL SEPTEMBER 19)

One day: \$350 for non-members,
\$300 for NoCOUG members

Both days: \$650 for non-members,
\$550 for NoCOUG members

REGULAR PRICING

One day: \$375 for non-members,
\$325 for NoCOUG members

Both days: \$700 for non-members,
\$600 for NoCOUG members

But does that prove that the performance improvement came from the change in plan? Is it possible that your index rebuild used a table-scan that resulted in the target table being cached somewhere (in the SAN cache, for example) so that accesses to the table were much quicker than they would normally be?

Even if the improvement was due to the change in execution plan, do you know why the plan changed? Was it because the rebuild packed the index, giving you a smaller leaf_blocks (or blevel, even), or was it because index re-builds automatically compute statistics in your version of Oracle, and a small change in the statistics (distinct_keys or leaf_blocks being the most likely) made the difference? Will the new plan survive the next statistics collection, even if the index doesn't start growing immediately after the rebuild?

If you don't check, you may end up performing a regular, yet redundant and potentially dangerous, rebuild of this index—and when, exactly, does it need to be rebuilt anyway?

What if the change was due to a change in the distinct_keys? There may be random occasions when the re-build strategy just doesn't work, or a day may come when the rebuild will no longer work because even the freshly rebuilt index has grown past a critical number of leaf_blocks.

Whenever you make a change that's supposed to fix a problem, try to capture the before-and-after information (e.g., statistics, execution plans, work done, number and type of waits). If there's any doubt in your mind about the root cause, never be afraid to document what you've done and the reasons why you did it—it may save you a lot of time in the future when your "fix" turns out to have been just a lucky coincidence.



Statistics—How and When!

Imagine you own a supermarket: as time passes, the number of distinct suppliers you have doesn't change much, the number of distinct products changes slowly and steadily, but the volume of sales changes constantly. What does this tell you about the statistics that Oracle needs to describe your data?

There are some statistics about your data that can be left unchanged for a long time, possibly forever; there are some statistics that need to be changed periodically; and there are some statistics that need to be changed constantly.

You may even need to massage some statistics so that they describe part of the data rather than all of the data—and there is no way that any supplied Oracle package can create such statistics. For example, many businesses hold data for seven years, but the end-users are often only interested in what's been happening in the last few weeks. The "average statistics" for seven years may give the optimizer a misleading image of the data the user wants to query. (Think of the odd sales patterns that a supermarket has around Christmas and Thanksgiving.)

⁴ Originally printed in the "Ask the Oracles" column in the August 2007 issue of the *NoCOUG Journal*.

For most people, the critical problem is to work out how to do the minimum amount of work to generate the best possible statistics, and it can take a lot of up-front effort to work out which statistics fall into which of the classes identified above. Fortunately it is possible to start simply and enhance your collection strategy over time.

My guidelines are simple: you may as well compute statistics for small tables, but large tables usually need just a small sample size (can you check this on a backup); some indexes will need their clustering_factor adjusted; a few columns will need histograms; a few columns will need specially constructed programs to manufacture "business aware" statistics; and partitioned tables will, in general, need programs to construct their statistics (possibly only at the table level after a "normal" call to the dbms_stats package has created some new partition-level statistics).

The biggest problem is that you need to understand the data. You may be able to take advantage of table monitoring to determine which tables are subject to significant change and v\$col_usage to see how columns are used in predicates, but neither of these helps you understand which columns have data distributions that require special treatment and which indexes need correction.

Ultimately I believe you need a table-driven mechanism—and initially you could probably set up a simple system to emulate your current stats collection. For each non-partitioned table, you need to record the frequency of collection, sample size, collection method, and whether or not to include indexes—in other words, most of the parameters to the gather_table_stats() procedure. For tables where you don't automatically gather index statistics, you need to include records for the relevant indexes with a similar list of parameter values. Finally, for any special columns, you need entries showing how to handle them—which may simply mean calls to create histograms. In all three cases, you may choose to reference a (homegrown) procedure that specifies a method for generating a completely artificial (though appropriate) set of figures. For partitioned tables, my approach is to design a custom stats collection package for each table as soon as I define the table.

Your stats collection routine can start as a loop to scan the table and obey the instructions it finds there—and you can start simply with a driving table that emulates your current dbms_stats calls, enhancing the system as your knowledge grows.

There's a lot more to say—but with only 600 words, no room to say it—but this is the core of the optimum strategy.



Does the optimizer need a hint?⁵

If you design your application perfectly, make all the smart choices with data structures, create all the relevant constraints, generate suitable statistical information about your data, and write carefully crafted code, then you will still find that there are a few statements that need Hints before the Opti-

⁵ Originally printed in the "Ask the Oracles" column in the August 2006 issue of the *NoCOUG Journal*.

mizer follows the “best” execution path.

There are many reasons why the Optimizer may need help—including bugs, simple deficiencies in the Optimizer model, and the inherent problems of collecting, storing, and processing the complex statistics needed to describe real-world data—thoroughly described by other authors in this compendium.

Counterintuitively, another reason why the Optimizer may need help is that there are some extremely cunning strategies built into the runtime engine. Consider, for example, the following simple query.

```
select outer.*  
from emp outer  
where outer.sal > (  
    select avg(inner.sal)  
    from emp inner  
    where inner.dept_no = outer.dept_no  
)
```

There are two major strategies that the Optimizer could adopt for this query—create a result set with the structure (deptno, avg_sal) and do a join to the driving table (an unnest operation), or scan the driving table and execute the sub query whenever necessary (a filter operation). If you want the un-nesting to happen, you could include the UNNEST Hint in the sub query; for the filter operation you could include the NO_UNNEST Hint.

Of course, you might try to avoid using Hints by manually un-nesting—rewriting the query as follows.

```
select outer.*  
from emp outer  
(  
    select dept_no, avg(inner.sal) avg_sal  
    from emp  
    group by dept_no  
) inner  
Where outer.sal > inner.avg_sal;
```

Alas, if you do this in recent versions of Oracle you might then need to stop the Optimizer from doing a cunning—but possibly catastrophically inefficient—piece of complex view merging by including the NO_MERGE Hint in what is now the inline view (or a NO_MERGE (INNER) in the main query).

But why might you want to control the strategy that the Optimizer chooses for the original query anyway? Because there is a clever trick, known as scalar sub query caching, that can occur at runtime to minimize the number of times the filter sub query is executed—but it is impossible for the Optimizer to know how many times the filter sub query will actually run. (The Optimizer may be able to work out the minimum number of times the filter sub query has to run, but that’s not necessarily a good estimate of the actual runtime activity.)

In this specific example it is likely that un-nesting will be the better option; in other cases it will be less obvious. And if the Optimizer chooses the wrong option, you have to give it a Hint or rewrite the query to make it do the right thing.

⁶ Author’s note: The text printed in the August 2006 issue contained an error. You really need at least *two* Hints per table.

But look at the comment I made about rewriting this query. In Oracle 8i, my rewrite with the inline view could be a good idea—in Oracle 9i the inline view might get merged, with disastrous effects on performance.

The same type of issue appears with Hinting—you find a Hint that seems to solve a problem in one version of Oracle and causes a problem when you upgrade to the next version. (The ORDERED Hint is a good example of this in 8i, and the PUSH_SUBQ Hint is a good example in 9i). The biggest problem with Hints is that they are badly documented; it is almost impossible to find out exactly what each specific Hint is supposed to do, and if you don’t know what a particular Hint does, how can you work out why it seems to solve a particular problem?

Hints can be very useful to solve urgent problems—but my general advice is (a) don’t use them as a first resort, (b) check whether the real problem is in the statistics, (c) if you really need to hint your SQL, you probably need an average of at least one Hint per table⁶ to lock in the execution path you expect, and (d) assume that you’re going to have to revisit and retest any hinted SQL on the next upgrade. ▲

Precise Transaction Performance Management

Precise TPM is the only complete Application Performance Management solution for all Oracle Business Applications and Databases

- E-Business Suite
- Siebel
- PeopleSoft
- Application Server (BEA)
- Databases
- Coherence

For more information visit: Precise.com

3 Twin Dolphin Drive, Suite 350
Redwood Shores, CA 94065
1877 845 1886

PRECISE

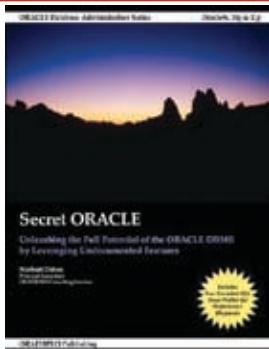
Secret ORACLE

Unleashing the Full Potential of the ORACLE DBMS by Leveraging Undocumented Features

A Book Review by Brian Hitchcock

Details

Author: Norbert Debes
ISBN: 978-1-4357-0551-7
Pages: 404
Year of Publication: 2008
Edition: 1
Price: \$41.95
Publisher: Lulu.com
 (self-published)



Summary

Overall review: Very good, as long as readers understand what they are getting into.

Target audience: Oracle DBAs and anyone who supports Oracle databases.

Would you recommend to others: Yes.

Who will get the most from this book: Those with DBA experience.

Is this book platform specific: No.

Why did I obtain this book: It was recommended to me.

Overall Review

The content is focused on Oracle database features that are undocumented. This raises several issues. How can the reader know if this material has been verified? If you actually use this material to make changes to your Oracle environment, will you be supported? And how do you know if the next patch you apply will change the way one or more of these undocumented features works? The answer to all of these questions is that the reader won't know. Having said that, this doesn't mean the material isn't valuable. Learning more details of how something works, even if you don't (or can't) use them, forces you to gain a better understanding of how it all works normally. The Oracle documents tell what to do but don't often tell you what you can't do. Other sources, such as this book, will tell you what you can't do, what works well, and why.

Learning about undocumented features can help you use the normal features better. A specific example is found on page 33. Knowing more about how OS_AUTHENT_PREFIX works allows me to improve security in my databases, even though I won't be using the undocumented feature described.

While the material is useful, remember that undocumented generally means unsupported and probably will change as patches and new versions are released. I think you will benefit from the material presented in this book, but I am not advising anyone to actually try to use these undocumented features.

This book is not a quick, easy read. It covers a lot of ground in very short bursts. It doesn't spend (waste?) your time describing a database and explaining what a tablespace is. It is assumed you want something more. This implies that the reader should be an experienced Oracle DBA, and while that is certainly an appropriate audience, it is also true that a junior DBA could benefit as well.

With a basic understanding of Oracle database structure and features, reading this book would be an excellent way to be drawn into many new areas of the Oracle RDBMS. Reading the various sections and referring to other sources for further explanation would be of value for any new DBA.

How much did I like this book? In the first 30 pages I found so many things that I didn't know but that I think are worth knowing that I can't describe them all to you in this review. That's how good this book is. This book does, however, redefine the concept of a *chapter*. Chapter 6 is two pages long, the longest chapter is 26 pages, and over half of the chapters are under 10 pages. This doesn't affect the quality of the content, but it is different and takes some getting used to.

I also found very few errors. This is interesting because I just finished reading a book from Oracle Press that had many errors, both typos and factual errors. I was surprised that a self-published book would have many fewer errors than a book from a major publisher.

Given the wide range of topics covered, I would recommend approaching this book as follows. Each chapter has Status section and a Benefits section. I would read the Status and if it interests you, read the Benefits Learned. Then decide if the entire section is relevant to your interests. Don't misunderstand, I personally recommend reading the entire book, because you don't know what you'll run across.

Since there are 37 chapters in this book, I don't break out each one; rather, I identify each of the eleven parts of the book and provide comments on each.

I also want to point out that while a book's introduction doesn't usually warrant comment, this time it does. It con-

tains the best explanation I've seen of database versus instance. Page xiii of the introduction is really worth reading.

I also liked the author's use of the word *disambiguate*—I think we need more disambiguation, and we need it as soon as possible!

Part 1—Initialization Parameters (Chapters 1–2)

I didn't know the details of the unconditional auditing for SYSDBA and SYSOPER. Similarly, the combinations of possible parameters for AUDIT_SYSLOG_LEVEL is useful. For PGA_AGGREGATE_TARGET it is interesting to know what the undocumented features are, but I would follow the author's advice that this should only be considered "under rare circumstances." The description of exactly how memory is allocated and released for manual PGA memory management is good to know.

Another section that I found to be excellent was the Misconceptions about PGA_AGGREGATE_TARGET. Again, while this book is mainly focused on undocumented features of Oracle RDBMS, the explanations of how things really work are great and don't require you to use any of these features.

Details of the EVENT parameter relating to 10046 tracing of multiple processes is very good. The section covering OS_AUTHENT_PREFIX is great. I would recommend this section and especially the Lessons Learned for all DBAs.

I had no idea that there are 540, 1124, and 1627 hidden (undocumented) parameters in 9iR2, 10gR2, and 11g respectively. While interesting, remember that these are hidden for a reason, and I would not change them without a good reason such as direction from Oracle Support.

Part 2—Data Dictionary Base Tables (Chapters 3–4)

Very interesting, good information about the DBA views we use all the time and the base tables they really access.

The explanation of the V\$OBJECT_USAGE view and index usage is very good. Even if this isn't directly relevant to your database, you should read this. It illustrates an important point. Official Oracle documents don't always tell us exactly how a feature works. How much time has been spent debating how to use a feature when we really don't know how it actually works because the manuals are wrong? Specifically, the V\$ implies that this is a dynamic performance view, but it's not. Cool. Good interview question. Similarly, two of the columns of this view store dates but are of data type VARCHAR2. That's not the way we are told to do things.

Part 3—Events (Chapters 5–8)

A detailed description of deadlocking is given along with using the Event 10027 to generate lots more diagnostic information. This could be very useful when developing applications and less so for operational support.

Event 10053, which outputs the decisions of the cost-based optimizer (CBO), is described in detail and output is shown. This is useful, although it would be better to see how all this could be applied to a real-world example. One example of the detail given is that if the CBO uses a cached execution plan there will be little output to see, and a workaround for this situation is given.

I had never heard of Event 10079, let alone how to use it to

document passwords being sent without encryption. Very interesting.

Part 4—X\$ Fixed Tables (Chapters 9–12)

The author discusses the issues around SGA sampling tools and doesn't advocate their use. I completely agree.

The discussion of how V\$ views are used to expose information in C data structures using X\$ tables as an intermediate step is great. I have never fully understood how this works. It is good to understand what exactly the X\$ tables are: they are not database tables.

Another useful discussion focuses on the security risks of the SELECT ANY DICTIONARY system privilege. Perhaps the most useful information in the discussion of X\$ Fixed Tables is that the author believes equally useful information is available from the V\$ views versus the X\$ tables, and that the author has never needed the contents of any X\$ tables to resolve a performance issue. This is something to think about as you decide where to put your training time.

The detailed example of how to go from the V\$ View to the related GV\$ view, and eventually to the lowest level X\$ table, is very instructive and concludes with a review of code available from the book's source code depot that will generate listings of the X\$ fixed tables for any V\$ view and vice versa.

A more detailed review of X\$BH and Latch Contention demonstrates that looking at the undocumented X\$ tables isn't the best way to go about improving system performance. The author asks the question, is it worth the DBA's time to pursue this level of investigation, or does this just lead to Compulsive Tuning Disorder (CTD)? A good discussion to keep in mind when you are faced with a tuning engagement.

While poking around in the X\$ tables may not always be the best use of your time, the section covering X\$KSLED and Enhanced Session Wait Data could be very useful. Here we are shown how to use the X\$ fixed tables related to the V\$SESSION_WAIT view to get better wait event timing data. It is shown why we only get centisecond resolution now and how to change this to get microsecond resolution, which can be used to examine very short wait events—and that this approach works for 9i as well as 10g.

The final section discusses X\$KFFXP and ASM Metadata. I don't use ASM so I can't comment on how useful this is, but we are shown how to use operating system-level commands to access ASM database and server parameter files, which may be useful for recovering data.

Part 5—SQL Statements (Chapters 13–16)

The use of ALTER SESSION (SYSTEM) SET EVENTS is covered. I didn't realize that other than turning on the 10046 trace event, use of these commands is not documented. For example, we aren't officially told how to turn off 10046. This section covers how to use these statements to trace database sessions while including wait event and bind variable information.

Similarly, the ALTER SESSION SET CURRENT_SCHEMA section is very useful. While I have oftentimes used the workaround of documenting a user's existing schema password, resetting the password, completing my work, and resetting the user password using the "by values" clause of the alter user

statement, it is cleaner to be able to actually change to the schema user you need to be. While there are some restrictions on what you can do in the user schema after using this command, and these are covered in detail, it is still a viable alternative.

The next section formally documents what we have used for a long time, namely the ALTER USER IDENTIFIED BY VALUES statement. A thorough example is given for using this. There is also an example of how to lock database accounts that offers improved security. This prevents hackers from being able to confirm even the existence of the locked account, which is often the first step to breaking in.

I didn't know about SELECT FOR UPDATE SKIP LOCKED and again, learning about an undocumented feature could help your general understanding of Oracle. This feature is used by Advanced Queuing to improve scalability of applications that try to update the same set of rows at the same time. It is also explained, and I didn't realize, that Oracle Streams is Advanced Queuing renamed.

Part 6—Supplied PL/SQL Packages (Chapters 17–21)

The first package discussed is DBMS_BACKUP_RESTORE. I really liked this because I've always wondered how RMAN works when it connects to and works on a database. In addition to the detailed discussion of how RMAN uses this package, there is a very worthwhile explanation of the issues with the control file holding all the backup information. A vital detail is that while the RDBMS from 9i forward has a feature to automatically backup the control file, this feature is disabled by default. It is also interesting to know that unlike other PL/SQL packages, this package can be executed when the db is not mounted.

DBMS_IJOB is used internally by DBMS_JOB, and can be used directly to work around limitations of DBMS_JOB such as creating jobs in other schemas. An amusing (to me, anyway) usage note is offered. It turns out that if a job is broken, the DBMS_JOB will return the NEXT_DATE parameter as a fixed date of Jan 1, 4000. Clearly the Y4K crisis is on the horizon. Remember, you read about it first in the *NoCOUG Journal*!

Next up is DBMS_SCHEDULER, which is certainly well documented, but there are many more details about this package that are not, such as executing external jobs and default privileges of external jobs. I didn't know the details of how exit code handling worked and this is explained very well. Also interesting is how to use DBMS_SCHEDULER to schedule RMAN jobs so that the database can back itself up. This requires using some of the undocumented features to control environment variables and argument passing among other issues.

I was fascinated to read the explanation of why the UNIX user "nobody" is a requirement for installing Oracle. It turns out that this is related to how external jobs not owned by SYS are executed. I recommend this section as yet another great example of learning useful security information by learning about undocumented features.

The DBMS_SYSTEM package includes an undocumented procedure GET_ENV. This is useful because you can use this to find out which ORACLE_HOME an instance is running from. This is something I wish I had known about in the past;

it would have saved me a lot of time. This package has many other undocumented features, such as enabling SQL trace in any database session and changing parameters in running sessions.

While you may not ever need to use these packages as described in these sections—and perhaps you shouldn't use or rely on unsupported features—I learned a number of useful things that help me understand what is going on in the database.

Part 7—Application Development (Chapters 22–23)

The first section is all about Perl DBI and DBD::Oracle. I'm not a Perl user so I can't comment on any of the technical issues covered here. Basically this tells us that ORACLE_HOME comes with a Perl installation and this includes the modules DBI and DBD, which can be used to develop applications that access the Oracle database. Specific issues addressed include Perl programming in an Oracle environment to deal with named bind variables, methods to connect to the database, and many more. This is very detailed and looks like it could be very useful if you need to develop your own database tools to track performance, etc.

Next we see material discussing Application Instrumentation. This includes the TRCSESS utility. There is a very detailed explanation of the problems caused by a shared server process. In short, tracing a shared server process may generate incorrect results because multiple sessions are being serviced by the single shared server process. It is details like this that are not covered in the official Oracle documents. Also discussed are issues relating to the meaning of the SQL*Net message from client wait event, a topic that has been discussed many times by many experts. I found this review to be worth reading, and it reinforces my perception that performance tuning is complicated and that you need to be very experienced with real-world experience to be sure what you are doing. The author is clear: ignoring this wait event is a mistake.

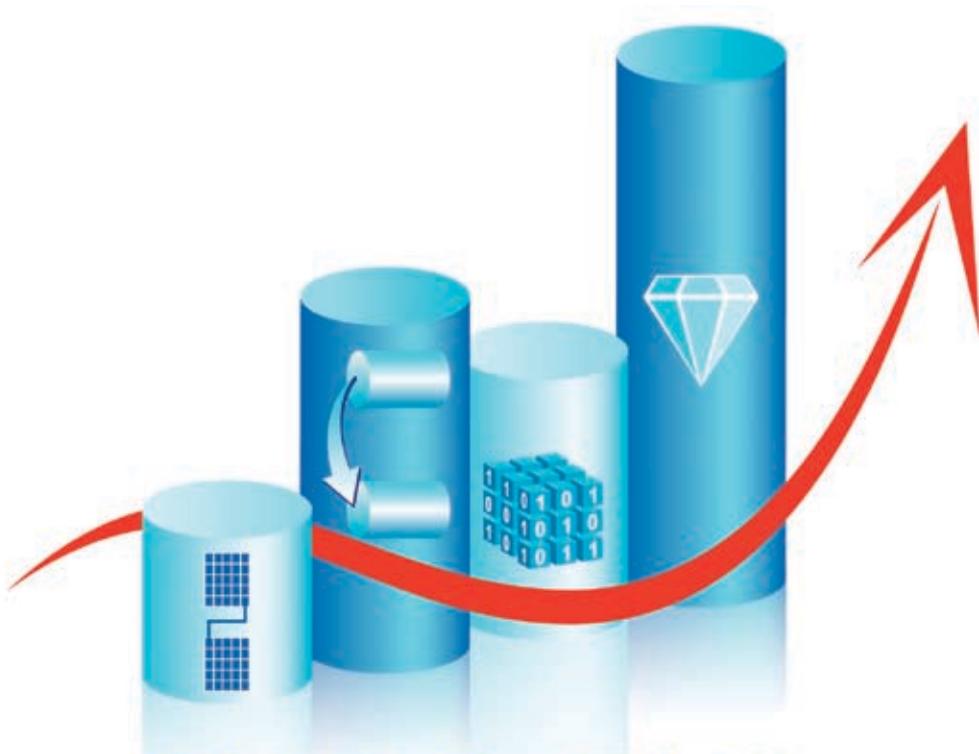
A case study using JDBC end-to-end metrics and TRCSESS is described in great detail. I'm not experienced with these features, and while I found the case study interesting, I can't comment on how useful it may be to other readers.

Part 8—Performance (Chapters 24–28)

Here we see discussion of trace files and how to read them. This is very detailed and will be useful if you spend (or plan to spend) lots of time reading the raw trace files. Even the casual observer will learn some things that will help reading trace files once in a while. It is the format of the SQL trace file that is undocumented. Users generally format these trace files using TKPROF and TRCSESS. We are told that reading these files in their raw state is useful because processing (with TKPROF for example) can obscure important information.

I'm not sure how useful this level of detail will be unless you are working with trace files all the time. I found it worthwhile to read because it is interesting to see what is in the trace files.

I've heard for a long time about how good performance tuning can be when you have a fully instrumented application. The problem is, I've never seen such an application; but then, perhaps I'm just unfortunate. The author offers this



BIWA SIG

Better Information Better Results

www.oraclebiwa.org

Oracle BIWA Summit 2008, December 2–3 at the Oracle HQ Conference Center in Redwood Shores, will have keynotes, hands-on sessions, and over 60 sessions focusing on Oracle Business Intelligence, Data Warehousing, and Analytics. Oracle products featured will include Oracle database, Analytic SQL, Data Mining, OLAP, Partitioning, RAC, Spatial, ETL, Oracle Warehouse Builder, Oracle Business Intelligence, BI Publisher, Essbase, Hyperion, and other related technologies and concepts. The Oracle BIWA Summit 2007 was a sellout with over 300 participants, and this year's conference will be even bigger and better. For details and registration, see www.biwasummit08.org.

comment on properly instrumented applications, “... something that has eluded me in my career as a DBA.” So perhaps it isn’t just me after all.

The STATSPACK chapter appealed to me more than some of the others because I have used Statspack before. While it would be difficult to say that Statspack is an undocumented feature, the author discusses undocumented report parameters, the structure of the Statspack repository, and how to use the analytic function LAG to examine large numbers of snapshots. I found it interesting to read about a specific bug that affects versions before 11g, a bug that omits wait events which happen only in the end snapshot.

Also interesting is the licensing issue for Active Workload Repository (AWR). AWR is bundled with the Database Diagnostics Pack, which costs extra. Yes, the AWR is installed and enabled by default, but that doesn’t mean you’re licensed to use it.

Next is how to integrate Extended SQL Trace and AWR, which is useful for finding all the execution plans used for a specific SQL statement over time. Further, some plans may not be captured normally because they don’t get sent to the trace file until the cursor closes. A detailed example shows how to retrieve such plans from the AWR.

A free profiler for extended SQL trace files, ESQLTRCPROF, is discussed. Note that this departs somewhat from the theme of the book, i.e., undocumented Oracle features. This profiler is not an Oracle product; it was written by the author and it appears to be documented. On the other hand, the author shows how to make good use of this utility. I haven’t tried to use this tool so I can’t comment on its usefulness. The examples in this section are very detailed and make clear that this utility is designed to be a replacement for TKPROF.

This is followed by a discussion of the MERITS Performance Optimization Method, an optimization method that relies on ESQLTRCPROF and uses undocumented features. I have not used this method so I can’t comment on it from personal experience. It does rely on AWR and ASH, which raises the licensing issues for AWR mentioned previously.¹ The case study is interesting and is described in great detail, which should offer readers more than enough information to determine if this method is something they want to try. Two things stood out for me. First, you must have a test system that is identical to the original environment. I agree, it makes sense, but I’ve never seen the resources spent to make this happen. Second, an excellent example of just how hard it is to really know what is happening inside your database: specifically, cursors without associated SQL statements. Turns out this is due to poorly instrumented LOB routines. This brings up again the issue of needing, but not having, fully instrumented application code to do the best tuning.

Part 9—Oracle Net (Chapters 29–32)

This was my favorite part, wherein I think I learned the most things that I can actually use in my environments. I had never heard of IP=FIRST in the listener.ora file. This feature can be used to control which network adapter is used on sys-

¹ Book author’s clarification: The MERITS method also works with Statspack. If someone does not have a license for AWR, they should use the script sp_capture.sql instead of awr_capture.sql.

tems that have more than one. I learned a lot reading how IP address binding works; about multihome systems; and how tnsping can be used with an IP address, not just a Net service name.

Valid Node Checking is a feature I have worked with in my Oracle Applications 11*i* environments. It allows configuring a list of hosts that may or may not connect to the database. While this is documented, the fact that you can change this dynamically for 10g and 11g is not. This means you can change or enable/disable this feature without restarting the listener.

While I probably won’t need to configure any of this, knowing that these things exist and how they work may help me troubleshoot issues.

Part 10—Real Application Clusters (Chapters 33–34)

I don’t support RAC environments. However, I learned some good things here, for example, the details of how Transparent Application Failover (TAF) works. One of the undocumented parts here is that you can use ALTER SYSTEM DISCONNECT SESSION to explicitly disconnect a session and have TAF take affect. The details of what TAF does and doesn’t do were new to me, specifically, that TAF for select statements “may fail.”

It turns out that you can remove the RAC option and this is undocumented. This could be useful in certain failure scenarios where you have to start the database but can’t repair RAC. The case study walks through all the steps needed to do this conversion.

Part 11—Utilities (Chapters 35–37)

These sections describe undocumented aspects of the OERR script, which provides error messages for Oracle errors, RMAN Pipe Interface, and how to use ORADEBUG for performance and hanging issues. I would stay away from using ORADEBUG to modify SGA structures unless Oracle Support told me to.

Conclusion

Change is good, change is scary, change is inevitable. The way Oracle books are published is changing as well. There was a time when authors had to submit a proposal to a major publisher and hope that their ideas would be found worthy of publication. Now, however, an author can choose to become published without the approval or oversight of anyone else. There are services that will take your content and offer it for sale on Amazon.com and print the actual book when someone orders it. No editor, no publisher, no technical review. Is this a good thing or just an inevitable change?

This is the first time I’ve read an Oracle book that was not associated in some way with a publisher. And this book is better in many ways than most I’ve read that came from major publishers.

Here is a specific example from my own experience: I read a book on Oracle Applications, a book that was not from a formal publisher. The information was very detailed and very relevant to someone that supports an Oracle 11*i* environment. From this book I learned some things that you must

(continued on page 21)



ROUNDSTONE
ROCK SOLID RELATIONSHIPS

Roundstone Systems is Northern California's
HP Gold Business Partner authorized in
HP's Oracle Solution Network Program.

HP, Oracle and Roundstone Systems have joined forces to help you reap the benefits of IT consolidation.

HP BladeSystem Servers:

- Increase agility - get products to market faster
- Reduce costs - attain higher system utilization, share resources, reduce management and maintenance
- Increase the overall effectiveness of computing systems and networks

510-764-2080

Gold Business Partner



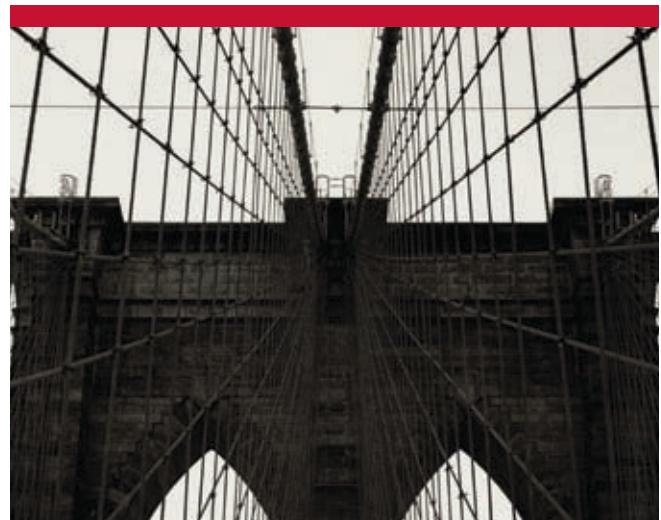
www.roundstonesystems.com



Enterprise data. Aligned.

You dedicate huge resources to enterprise applications like Oracle®, PeopleSoft®, JD Edwards®, Siebel®. And every year they grow in size and complexity — increasing maintenance challenges. Now you can take back control with Princeton Softech Optim™. Optim gives you the ability to align application data management to performance goals. Optimizing results, mitigating risk and controlling the cost of every IT investment — servers, software, storage, networks and people. Is this heaven? No, it's the power of Enterprise Data Management.

Learn more at princetonsoftech.com.



- Analyze performance by workload/application
- Predict performance shortfalls in the next 12 months
- Understand the effects of change: new apps, 10g, more users, new hardware...in seconds

BEZ systems The Prediction People™



Real-World Experience

For Oracle database consulting and support, it makes sense to work with a company that has a proven track record. Since 1995 our clients have relied on us for:

- Performance tuning
- Migrations and upgrades
- Backup and recovery strategies
- Database security audits

Plus, we offer ongoing **remote DBA** support plans that are tailored to your business needs and budget.

Call Us Today!

(415) 344-0500 • (888) 648-0500

www.dbspecialists.com

Database Specialists

ORACLE | CERTIFIED SOLUTION PARTNER

Compliance Data Systems

The Business Case

by John Weathington



John Weathington

Your company's best chance at staying in compliance and surviving any audit is to invest in what I call a *Compliance Data System (CDS)*. In general terms, a Compliance Data System is a subsystem in your company's information systems architecture whose raison d'être is the empowerment of your company to support its governance, risk, and compliance (GRC) efforts. Imagine the CIO approaching a company's IT infrastructure from a strategic point of view. There needs to be a way for your company to process transactions, perhaps an ERP implementation. There also needs to be a way for management to use this transactional data for strategic intelligence, perhaps an enterprise data warehouse. What often gets overlooked is the company's need for a robust data system to manage its compliance responsibilities.

So what I've seen done at other companies is an attempt to leverage their data warehouse to support their compliance needs. This strategy has a lot of problems, and they're all rooted in the fact that your strategic system was never built to solve compliance problems. It's like using a pair of needle-nose pliers to unscrew a nut. It might work sometimes on a nut that's already somewhat loose, but to handle the heavy work you need a good wrench.

So the value in constructing a data system specifically devoted to compliance is in the bulletproofing that it does for your company. With the proper data systems in place, your company can avoid unnecessary penalties and fines for non-compliance. A recent study has shown that if your company has revenues over \$1 billion, the average cost of one compliance failure is \$81 million! That's a heavy price to pay for not being organized—but it's the reality today. In realistic terms, it might cost the same company \$5 to \$10 million to implement a robust compliance data system. That's a return on investment approaching a factor of 10 in the first year, and the returns are greater when you start analyzing the total cost of ownership over its usable life. I recently managed the implementation of a compliance data system for a large high-tech company. They spent \$5 million to fortify a \$100 million government contract, which makes a lot of sense to me.

The Requirements

Now that we understand the need, let's talk about what's required. First of all, the people driving the requirements should be your internal audit team, or whoever is responsible for making sure you survive an audit. What kind of audit?

Well, if your company is public, SOX compliance is a good place to start, but it doesn't have to start or stop there. What's popular in the compliance industry now that a few years have gone by since SOX was first enacted is what's called *compliance convergence*—or leveraging existing infrastructure to solve multiple compliance problems simultaneously. To make things simple though, let's focus for a moment on SOX compliance.

When your company gets audited, the auditors are going to need evidence that you are in control of your financial-reporting process. Your company will need to clearly demonstrate how it came up with its numbers, with a clear audit trail from the transactional systems to the financial statement.

They will also look at your control policy. Remember, control policies are put in place to mitigate some sort of risk. For instance, to control for the risk that inexperienced finance people enter in the wrong journal entries, your company might have a control policy stating that a finance manager shall approve all journal summaries. The auditors will look for evidence of this approval.

When your company does have a control failure, the auditor wants to know how your company handles it. First of all, is your company even aware that controls are failing? If so, what remedial actions are being taken?

Here are some basic requirements around building a data system to support compliance:

- A system that collects data from transactional systems and reports on how well control policies are being followed.
- A system that provides an audit trail from the transactional systems to the publicly reported financial numbers.
- A system that stores and quickly retrieves any evidence that supports its assertions of control and compliance.
- A system that tracks control failures and the actions that are being taken to remediate.

The Design

As we start thinking about the design of our CDS, we can leverage some of the thought work that has already been done. Since it seems like our CDS is collecting data from (i.e., downstreaming) the transactional systems, it seems logical

that we can reference the designs used to build a strategic data warehouse.

Bill Inmon, one of the fathers of present-day data warehousing, wrote *The Corporate Information Factory* (John Wiley, 1998) in which he discusses architecture that is seminal to the way data warehouses are constructed today. The more salient components of Inmon's corporate information factory are as follows:

- **The Enterprise Data Warehouse.** The enterprise data warehouse is the central place where data from the transactional systems (i.e., ERP) are warehoused. The data are organized into subject areas, which intelligently organizes data from the disparate transactional systems. Data in the enterprise data warehouse are non-volatile, meaning that the data are written once and never updated. It's also time-variant, which means that change data capture is taken into account. This allows you to go back in time and see what the data looked like last week, last month, or even last year.
- **The Operational Data Store.** En route from the source systems to the enterprise data warehouse, the data will typically pass through an operational data store. Like the enterprise data warehouse, the data here are subject oriented; however, unlike its counterpart, the data here are volatile and current. Every time the operational data store is loaded, the data are refreshed with a new snapshot of the current state of the business. In this way, the operational data store consolidates data from different sources into one subject area. For instance, you may have several different sources that contain customer data, and the operational data store would combine all this data to contain the proverbial "single source of the truth" for any customer.
- **The Data Mart.** The data mart is an architecture that's constructed to solve a very specific business problem. Typically created from the enterprise data warehouse, it's commonly seen as a "star schema" with a fact table that contains metrics and dimensions that contain the qualitative measures that you can slice and dice the metrics by.

So given this background, let's start designing our CDS. Our CDS could be integrated into our strategic data warehousing architecture, but it's better to keep it separate. To begin, your source tables will feed a compliance operational data store. The goal of your compliance operational data store is to report on your state of compliance at any given moment. A compliance operational data store that refreshes every day (i.e., Class 3) should be sufficient.

The difference between your traditional operational data store and your compliance data store is that your compliance operational data store will contain assertions and the result of those assertions. Assertions are a statement of compliance or control, and when tested will produce a Boolean value—true or false.

In our example above, our manager needed to approve journal summaries. The assertion is that the manager is indeed doing this. To prove this, you might need to source in all

journal entries and all summaries. The assertion would be that all journal entries have a summary and all summaries have a manager's approval code. You might even need to bring in data from your HR system to validate that the person approving the summary is indeed a finance manager.

In your transformations that move the data from your source systems to your compliance operational data store, build logic to test for these conditions and physically populate a Boolean field that reports on the result—true or false. If the result is false, you have a control failure. Replicate this strategy for all of your compliance points, and then create a single table that consolidates all control failures into one place. In this way, you have an instant snapshot of all of your company's control failures.

In the next article of this series, we'll explore how to leverage the enterprise data warehouse and the data mart, and also introduce some new ideas that complete our compliance data system. In the meantime, feel free to email any questions you may have, so I can incorporate the answers in the next publication. ▲

John Weathington is a management consultant who helps finance executives save money, reduce stress, and avoid penalties and fines. His San Francisco Bay Area-based company, Excellent Management Systems, Inc., has helped companies all over the world such as Sun Microsystems, Silicon Graphics, Hitachi Data Systems, and Hogan and Hartson, LLP. He is a Project Management Professional and a Six Sigma Black Belt, as well as an Oracle DBA and Business Intelligence Architect. He runs an expert blog for Quest Software called John Weathington's Quest for Compliance, a monthly newsletter called Flawless Compliance™ and a public blog called Hard-Boiled Compliance. For more information, please access his website at www.excellentmanagementsystems.com.

Copyright © 2008, John Weathington

Save Time, Energy and \$\$'s

by the seamless integration of your orders, invoices, ASN's or your supply chain with Oracle's Gateway.

We have partnered with Softshare to bring you Delta, their low cost but powerful—translator with EDI, XML, VAN services, and Web. 2.0. Or, we can integrate with prominent solutions such as Gentran, GIS, Oracle Fusion with BPL, or Inovis.

We also offer e-SupplyLink's warehouse integration products for UCC/EAN-128/ASN labeling, packing, shipping and ASN's.



Fundamentals of Data Quality—Part IV

by Michael Scofield



Michael Scofield

In the first article in this series, I defined the fundamentals of data quality and explained various more detailed characteristics of data quality.

In the second installment, we focused upon simple techniques for understanding data behavior, using reporting tools you may have around the shop. Data behavior in a mature, legacy database may be quite different from the “normative” documentation (reflecting the designers original intent).

In the third installment, we surveyed the different potential roles of a data quality analyst, and what tools or functionality of tools would be needed to support those roles.

A major challenge in ensuring and preserving high-quality data is when we attempt to merge data from various sources. This article introduces those issues.

Data integration

Data integration involves bringing data together from two more different sources, to form a new database (real or virtual) where the whole is more valuable than the sum of the parts, and where the resultant data is richer and makes sense. “Making sense” is often the challenge.

Getting two sources of data onto the same physical platform is generally easy. Putting them into the same database (albeit with totally separate table sets) does not accomplish true integration. They may be side-by-side, but still unrelated. To achieve true integration (what I often call “semantic integration”) requires placing data from various sources into common tables in the target database.

Data integration can be in a tabular paradigm (associating facts about common subject entities of interest), or it can be spatial (associating objects which are close to each other) as done in a geographic information system. In this article we will focus primarily upon tabular data integration.

A common situation is a company attempts to integrate facts about its customers into a customer relationship management (CRM) data warehouse—said facts coming from different sources. Again, a DW is defined here as a copy of data originated elsewhere. But the principles of data integration extend beyond the data warehouse to operational data stores and even the exchange of transactional data between business applications.

If data integration is not designed and implemented properly, the result can be bad data quality. Proper design also in-

volves thoroughly understanding the architectures of the data sources, and how the data actually behaves in those sources (see my previous installments on data profiling)

There can be many points of disconnect in the logical data architecture of two or more sources.

- A subject entity (and its table) may exist in one source, but not the other.
- Two tables may describe the same entity, but have different business life cycles.
- While tables may correspond (in the business entity they describe) they may have a different set of attributes.
- Two hopefully corresponding entities may have different subordinate or characteristic entities. For example, the employee record from one company may have a characteristic entity of “language spoken” while no such table exists in the other source.
- When two tables represent the same entity, their keys may be incompatible in format.
- When two tables represent the same entity, their keys may be format-compatible, but contains values which clash with each other. (e.g. Acct # 10005 means Mary Smith in one customer file, while #10005 means Philip Jones in another; this can be described as a value homonym.)
- Two nominally corresponding (or “paired”) fields (columns) from corresponding tables in the two sources may have different formats, different domains (even with the same format), or be populated at different rates.

Target database: new or legacy?

The target database may pre-exist the integration project (such as a surviving business application) or it can be custom designed (by us) which often makes things somewhat easier. With a DW, you often have the option of designing the target architecture, yet you may encounter new challenges of incorporating some of the architectural nuances of each of the sources into the target.

It is rare that a single subject entity from each source must be considered in a simple match. More often, there are more complex architectures which include reference files, and logical children of critical entities.

Then, for each table pair (one from one source, one from the other) we must be specific in how we intend to “merge” the files. Generally, most integration at the entity or table level involves either a logical join or a logical union.

There is a profound difference between these two concepts. In the logical join, we are integrating data to make each record richer (and wider) with more information about each instance. In the logical union, we are expanding the set of instances—enlarging its scope or range. An example of the latter is when two companies merge (often with lawyers involved) and they wish to combine their two customer files. In reality, such a “union” is not entirely simple because we must often deal with potential duplicate records (e.g. a person being a customer of both companies).

Lateral join

As we said, to make integration work, there first must be some logical similarity between basic subject entities. We cannot rely upon mere table names to evaluate this. We must look at the primary keys to the tables, and the non-key attributes.

It is often possible that primary keys on two tables (one from each source) which are to be joined together are incompatible. A classic situation is customer relationship management (CRM) where companies try to construct a more complete picture (wider and deeper) of each customer. The problem is that many legacy applications, the object identified by the primary key is the account, but there is no reliable customer identifier (other than, perhaps, the social security number, which may not always be captured properly).

So we may have a savings account master file (supporting one application) with one kind of key (shown below). For simplicity, only a few records are shown.

| Acct No. | Name | Open Dt | Balance |
|----------|------------|----------|----------|
| 4001 | Mary Jones | 5/8/1999 | 48.28 |
| 4002 | John Smith | 5/9/1999 | 1,897.22 |

We also have a customer loan master file (in an application entirely different from the savings system) with a different primary key (shown below).

| Cust # | Name | Open Dt | Balance |
|--------|-------------|----------|---------|
| 81005 | Mary Jones | 9/2/2001 | 338.24 |
| 81009 | Jane Harlow | 9/3/2001 | 427.91 |

How do we “join” this data together to get a more complete picture of the customer? We cannot use the primary keys.

Matching by name is tempting, but simplistic. Two people can have identical names. (There are easily over 14,000 people in the United States with the name “John Smith”. Surely the middle name or initial could provide some differentiation, but we still cannot rely on that.)

In solving this matching-of-persons problem, it is often necessary to employ software or an external data processing service which merges data about persons using more clues (such as Social Security Number, date-of-birth, and address). Some outsourced services bring additional market-available data of their own to the matching process.

Another challenge is that the same person may have multiple accounts in any given table or entity set. So we need to know how much de-duping we should do prior to the matching.

But the challenges are not over. Integrating these two simple tables may result in orphan records on one side or the other. What are we going to do about those? These are not technical challenges per se; they are business-oriented data management and data design questions. In other words, what results do you want, and how should those results appear?

Instance ambiguity

Even when the primary keys of two source tables are format compatible, any individual value may not represent the same instance in reality. Consider that we are trying to “link” data about bank customers from two legacy applications. Again, we will show only the record of interest for this example.

| Mortgage Loan Master File | | | |
|---------------------------|------------|----------|----------|
| Acct No. | Name | Open Dt | Balance |
| 5013 | John Smith | 5/9/1999 | 1,897.22 |

| Safe Deposit Box Master File | | | |
|------------------------------|-----------------|----------|----------|
| Box No. | Name | Box size | Open Dt |
| 5013 | Priscilla Davis | B | 2/4/1997 |

While we may assume these are both unique primary keys (not always a safe assumption), and we see they are the same format and value, they do not refer to the same person or instance in reality. In one sense, we may call this a “key value homonym”. The values look the same, but their meaning is different.

Unioning two data sources

The other (“non-join”) data integration situation (at the table level) is where two tables are unioned together to create an expanded set of instances. Again, a common occurrence of this is when two companies merge their customer master files. (The “union” concept would also be useful in integrating vendor master files, perhaps product master files, and some kinds of common assets.)

In all of these union situations, a major merge-design question involves the mutual-exclusivity of records (or the instances described by the records). This would be true within each source file (is each instance represented only once in the file?) and we can ask the same question about the target file (after unioning, is there only one record for each instance in the real world?).

This again points to the classic question: How do you know if two records are describing the same instance? Is “John Garagiola” the same person as “Jack Garagiola”? As in the CRM integration situation discussed above, the de-duping of records (before and/or after the union) may require more sophisticated software which considers more clues.

Precedence?

In either case, when two records are deemed to describe the same person with a high degree of confidence, then if any non-key facts are different in the two records, we have question of precedence of data. Which fact is more reliable?

Rules of precedence of duplicate or quasi-duplicate data can be complex. And they are, of course, business-oriented (as are many of the decisions discussed in this article). If update dates are available for two competing records, perhaps the

most recently-updated record would be more accurate (or current). But can we be sure?

Joining and Unioning at the same time

There can also be situations where data from two tables (each from a different source) can be, in a sense, both joined and unioned at the same time. For example, two companies are merging and combining their customer master files. Both have name and address of all customers. Company-A has date-of-birth on customers, but Company-B does not. However, Company-B has education level, and Company-A does not.

When these two tables are combined, there may be duplicates (a person is a customer of both companies) in which case a rich record can be created containing both date-of-birth and education level for that person. However, for all the customers coming only from Company-A, the education level field in the target file will be empty. Similarly, all the persons coming solely from Company-B will have no date-of-birth.

This situation creates an interesting dilemma for designing the target file if it is for a data warehouse. The DW designer must make clear to the business users that some customer attributes (particularly, DOB and education level) are going to be missing (hopefully designated by some kind of “null” expression). Obviously, this should go in the metadata, but is that metadata obvious and visible to the occasional business user?

If data warehouse users absolutely need both fields (DOB and education) fully-populated for some particular kind of analysis, they may have to deal with a smaller subset of the full, integrated customer file—namely only those persons who were customers of both companies before the merger.

Inconsistent dimensions

A common purpose of building a data warehouse is to combine “cause” and “effect” data from divergent applications or sources. For example, advertising spending by media market (“cause”) may come from one source (perhaps an accounts payable system) while actual sales performance (“effect”) may come from another application.

But the analysis will not be done at a detail level, but at some aggregate levels, generally on dimensions of time, geography (or media markets), and perhaps brand or product.

It is common for sales activity to be rolled up into marketing zones which may or may not match the media data. For example, sales activity may be divided into “North Bay”, “Peninsula” and “East Bay” territories (for ease of sales calls and delivery), with the “East Bay” territory extending all the way to Stockton and Davis.

Yet, the advertising influence upon those sales may come from different broadcast outlets. So sales in the East Bay territory may be influenced by advertising in the San Francisco media market, and the Sacramento media market.

While we want to correctly associate cause and effect, there are no simple identifiers for what are essentially, two radically different objects (media markets, and sales measurement zones). The solution here may require some subdivision of totals into pro-rated allocations. Or, the aggregate shipment

data (by sales zones) is not useable, and individual customers (or retailers) must be assigned to media markets at a more granular level (more work, but my preferred solution).

A similar situation may exist on the time dimension. Sales are the results of individual events (orders from, or shipments to a specific customer), and media exposure is often also the sum of discrete events (ads run on a particular date, in a particular market, by a particular media outlet). These discrete events add up to ad spending. But how they are added up (over what time period) may bias the analysis.

The most common aggregate on the time dimension may be calendar month, but some businesses report their sales in fiscal months, or other promotional time periods (for example, a 6-week promotional period during which a certain price or discount is offered). Immediately, aggregate data may not “fit” between two sources. My bias is always to get the detail (individual ad buys, individual shipments), and roll them up in the context of the data warehouse into a “conformed dimension” of time.

Data warehouses are built for analysis, therefore the data integration and preparation must be done in the cultural context of the corporate world view. So if executives view the world in a particular dimension (spatially or time-wise), sometimes the reporting must be done in those dimensions.

Data flow and timing

Additionally, data integration usually involves two or more data flows, usually at some predictable point in time. Inside those flows, we may have to insert some kind of translation of codes. But all data flows exist in a time context. Part of the ETL (extract, transformation, and load) design involves the timing of those flows, to reduce the chances of orphans, or having rules or automated procedures which dictate how to handle the orphans.

So the extracts from source-A and source-B should probably happen at the same time.

Updates, amendments, and corrections

One final issue, which is not strictly a data integration issue, but is important in creating data (in the target database) which accurately reflects the source. This involves the rules of updating the data in each source. One question is this: after you have periodically skimmed off the transactional records (a.k.a. “facts”) from the source, can any of those records be changed? An example might be where you do a month-end skimming of the order master file, and a day later, a customer amends one of those orders. Now, your data warehouse and the source are not in sync!

So at the end of the next month, are you going to scan more than the most recent month? Are you going back to see if previous month’s transactions have been amended? To plan this, one must understand the business rules about amending transactions after they have been posted.

One must also understand the difference between updates and corrections. The difference may seem trivial, but it is not. An update is changing one true fact to a different value. He used to live at 123 Elm Street, but now he lives at 479 River Road.

A correction, on the other hand, applies to a fact which was



Winners at the free raffle at the summer conference. The giveaways included a full pass to Oracle OpenWorld 2008.

wrong . . . never true. "She never lived at 501 Canal Street. It was always the wrong address. The correct address should have been 601 Canal Street."

Data warehouses often keep more history than is retained in source applications. So while a customer-related application may only be interested in retaining the current address, data warehouse analysis often wants the address at which previous transactions or events took place. Which means corrections are more important in bringing more precision to historical data. Hence, for on-going data warehouse refreshment, one may need to carefully design the flows, distinguishing between updates and corrections.

Conclusion

This article has only scratched the surface of the complexities and potential points of failure in data integration. Perhaps the most important points to remember are first, understand the logical data architectures (not just the physical architectures) of all sources. To do that, you must examine the actual data (not just data documentation or file descriptions). You must ensure that primary keys behave the way you expect them to; you must look for data duplication, data omissions, and inconsistencies. You must be sure you understand all the anomalies and make provision for them in your ETL design. ▲

Michael Scofield is manager of data asset development at ESRI, Inc. in Redlands, CA. He is also an adjunct faculty member at Loma Linda University in the Department of Health Information Management. He is the 2008 recipient of the DAMA International (Data Management Assn.) Community Award for his contributions to the education efforts of that organization.

He lectures widely on data quality, data management, and geospatial data.

Copyright © 2008, Michael Scofield

(continued from page 14)

not do when installing a new 11*i* environment—things that are not specifically identified in the official Oracle documents as unsupported. I don't think it was a coincidence that some of the most useful information I have found about supporting 11*i* came from a book that was not from a major publisher.

Now that it is apparent that self-publishing of Oracle books is really practical and being done, we need to change the way we look for them. This self-published book was only available on Amazon.com. This tells me that some of the best, most timely, and unique books won't ever make an appearance in your local bookstore. Also note that when you search Amazon.com, Oracle books from the major publishers will appear first. If you search for "Oracle," you will find *Secret Oracle* as the 222nd result, or on the 19th page of search results. You have to realize that these books are out there and you have to go find them. ▲

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.

Copyright © 2008, Brian Hitchcock

Teamwork at NoCOUG



Back row, left to right: Naren Nagtode—Director of Marketing, Claudia Zeiler—IUG Representative, Roger Schrag—Director of Conference Programming, Joel Rosingana—Director of Membership, Eric Hutchinson—Webmaster, Lisa Loper—Vendor Coordinator, Hanan Hit—Vice President; *Front row, left to right:* Jen Hong—Secretary and Treasurer, Randy Samberg—Track Leader, Iggy Fernandez—Journal Editor, Nora Rosingana—NoCOUG Staff Member.

NoCOUG is a successful organization with more than 500 members, and there's no way it could run without teamwork. We have a full and active Board of Directors, plus other volunteers who contribute regularly. All the people on the NoCOUG team contribute in both big and small ways, depending on what they have time for. And it's all of us working together as a team that makes for the great conferences, training days, and other benefits.

But volunteering your time is far from without rewards. In fact, volunteering with NoCOUG offers opportunities to meet and talk with speakers, authors, and other professionals in the Oracle field, as well as other activities. In fact, if your day-to-day job has become routine or doesn't offer you the chance to use some of your other skills—interacting with people, writing, organizing events, etc.—volunteering is a great way to utilize those skills. It's surprisingly fun once you get started. You'll find we are a welcoming bunch of people, and most volunteers say their favorite aspect of volunteering is the people they meet. So, if you would like to get involved but don't know where to start, here are some quick things you can do that don't take much time:

- Contribute an article to the *NoCOUG Journal*
- Volunteer to speak and share your knowledge at a conference
- Recruit a knowledgeable Oracle colleague to speak at a conference or contribute an article
- Help with registration on the day of our quarterly conference
- Assist with marketing our conferences and training days

And, there are plenty of other opportunities to help out. Remember, it takes a lot of teamwork to keep our successful organization growing and providing value to its members. So, if you want to be part of a great team, just send an email to board@nocoug.org and let us know how you want to get involved.

What are you waiting for. Join the NoCOUG Team! ▲

"I have met many wonderful people during my years of volunteering. Some long-term friendships have developed. Every quarter brings new friends. This is what I enjoy about NoCOUG volunteering."

"A big reason for volunteering is the give-back factor. I have received a great deal of valuable experience as a NoCOUG member. I feel it's important to give something back to the organization."

—Joel Rosingana

NoCOUG Membership Director
and Past President
Independent Consultant

"Volunteering with NoCOUG helped me round out my professional skill set by augmenting my technical skills with the 'soft' skills that I think are so very important. During these last eight years, I have held a variety of positions ranging from webmaster to president and I've learned a lot that I would never have learned just being an Oracle DBA or application developer. I've learned a lot about working with people, about how the whole is so much more than the sum of the parts, about how to get things done that are too big to do all by myself."

—Roger Schrag

NoCOUG President

Many Thanks to Our Sponsors

NoCOUG would like to acknowledge and thank our generous sponsors for their contributions. Without this sponsorship, it would not be possible to present regular events while offering low-cost memberships. If your company is able to offer sponsorship at any level, please contact NoCOUG's president, Roger Schrag, at www.nocoug.org/contact_us.html?recipient=roger. ▲

Long-term event sponsorship:

LOCKHEED MARTIN

CHEVRON

ORACLE CORP.

PG&E

Thank you! Year 2008 Gold Vendors:

- BEZ Systems
- Confio Software
- Database Specialists, Inc.
- Embarcadero Technologies
- IT Convergence
- Network Appliance
- Precise Software Solutions
- Princeton Softech
- Quest Software
- Roundstone Systems

*For information about our Gold Vendor Program, contact the NoCOUG vendor coordinator via email at:
vendor_coordinator@nocoug.org.*



TREASURER'S REPORT

Jen Hong, Treasurer

Beginning Balance
July 1, 2008

\$ 42,564.25

Revenue

| | |
|----------------------|---------------------|
| Membership Dues | 3,510.00 |
| Meeting Fees | 1,200.00 |
| Vendor Receipts | 5,300.00 |
| Advertising Fee | — |
| Training Day | — |
| Sponsorship | 60.00 |
| Interest | 51.69 |
| Paypal balance | 500.00 |
| Total Revenue | \$ 10,621.69 |

Expenses

| | |
|-----------------------|---------------------|
| Regional Meeting | 6,947.34 |
| Journal | 6,588.39 |
| Membership | 34.06 |
| Administration | 1,158.98 |
| Website | 69.63 |
| Board Meeting | 587.35 |
| Marketing | 108.42 |
| Insurance | 500.00 |
| Vendors | 19.80 |
| P.O. Box | — |
| Training Day | — |
| Accounting | 800.00 |
| Miscellaneous | — |
| Total Expenses | \$ 16,813.97 |

Ending Balance
September 30, 2008

\$ 36,371.97

NoCOUG Fall Conference

Session Descriptions

For the most up-to-date information, please visit www.nocoug.org.

—Keynote—

Testing, Testing, Testing

Jonathan Lewis, *JL Computer Consultancy* 9:30–10:30

In the Oracle world, there are two main reasons for constructing tests. One is to track down the cause of an existing problem; the other is to predict probable outcomes of a design strategy.

In theory, creating tests sounds as if it ought to be easy—and if you take the wrong approach you can waste a lot of time.

This presentation looks at the problems of testing, making its point and highlighting the accidents that can happen by discussing a few production cases.

From a technical perspective, we list a few key mechanisms that should be the focal point of your testing if you want to spend the smallest possible amount of time winning the largest possible rewards.

Jonathan Lewis is well known to the Oracle community as a consultant, author, and speaker, with 20 years of experience in designing, optimizing, and troubleshooting on Oracle database systems. His latest book is Cost-Based Oracle Fundamentals.

—Auditorium—

Beginners Guide to Partitioning

Jonathan Lewis, *JL Computer Consultancy* 11:00–12:00

Partitioning has been around for a long time—and Oracle 11g has added yet more options to the feature. But are there any good reasons to use partitioning, can you bolt it on to an existing application, and what are you going to get out of it?

In this presentation we discuss the costs and benefits of partitioning, with particular attention to the fact that a bad implementation can damage performance rather than enhancing it.

After seeing this presentation, you will be better equipped to evaluate situations that might benefit from the introduction of the partitioning option.

Xtreme SQL Tuning: The Tuning Limbo

Iggy Fernandez, *Database Specialists* 1:00–2:00

We will illustrate a plethora of SQL tuning concepts using a simple join of two tables. The tables are modeled after the well-known Oracle views, DBA_TABLES and DBA_INDEXES, and the exercise is to list the names of just those tables that have indexes of a specified type, e.g., BITMAP. The efficiency of an SQL statement can be measured by the proxy of “logical reads”—that’s your first SQL tuning concept right there—and our first solution requires hundreds. Using a variety of methods, we bring the number of logical reads lower and lower—all the way down to two. But we don’t stop there; the limbo continues and we lower the number of reads to just one . . . and reveal how you can get data from two tables with just one logical read. Finally, we eliminate logical reads altogether—and we’ll have you rubbing your eyes in disbelief. Learning can be fun too—you’ll enjoy the limbo music and limbo videos!

First-Failure Fault Detection and Diagnostics:

11g New Features and Novel Methods

Jeremiah Wilton, *ORA-600 Consulting* 2:30–3:30

Most unplanned Oracle downtime and service reductions are caused not by the initial occurrence of a fault, but by the many subsequent occurrences of the same problem that take place before DBAs or support find a solution or workaround. Diagnosing a problem the first time it happens in a mission-critical environment is difficult even for the most experienced Oracle professionals. To address such circumstances, several new 11g features have been introduced under the umbrella of the Diagnosability Infrastructure. In addition, there are a variety of novel methods that make it possible to automatically detect and diagnose faults upon the first occurrence. Instead of suffering through dozens of failures from the same problem, mission-critical sites will benefit from implementing first-failure fault detection and diagnostics strategies as described in this session.

Breaking Oracle: Creating Realistic Failures for Testing and Diagnostic Practice

Jeremiah Wilton, *ORA-600 Consulting* 4:00–5:00

To become highly skilled in the area of troubleshooting and failure resolution, most Oracle professionals endure a wide variety of trials by fire in production environments over many years. By learning to create realistic load levels and failure scenarios in test systems, Oracle professionals with little experience in troubleshooting and fault resolution can hone their skills without the customary stressful ordeals and business impact. This session will provide live simulations of problems, including hangs, spins, and crashes, followed by effective techniques for diagnosing and resolving faults.

—Tassajara—

Using Oracle in a Cloud

Editor's Pick

Bill Hodak, *Oracle Corporation* 11:00–12:00

Cloud computing? Utility computing? Elastic computing? Are you having a hard time understanding what these new industry buzzwords mean? If so, you don’t want to miss this session. Have you ever wanted a new server to start developing your next database application but had to wait for months? How does having a new server in 30 minutes sound? Imagine

Simplify Identity Management



Enhance security, provision users quickly, and effectively protect your resources with IT Convergence's Oracle Identity Management solutions.



Contact us today: www.itconvergence.com

Database Management

| | |
|----------|---|
| Develop | Ensure the highest quality code is developed regardless of user skill set or location |
| Optimize | Maximize code performance and eliminate time-intensive manual tuning processes |
| Validate | Performance test your code for scalability before deploying to production |

Can Toad® Turn Your Team Into Experts?

Sit back and relax. You have Toad's development best practices on your side.

You could spend your day tracking down bad code. Or you can take control with Toad® for Oracle.

Promote development best practices in your organization. Read "Implementing Database Development Best Practices" at www.quest.com/relax



©2007 Quest Software, Inc. All rights reserved. Quest and Quest Software are trademarks or registered trademarks of Quest Software. All other brand or product names are trademarks or registered trademarks of their respective holders. www.quest.com/NoCOUG.

ENTERPRISE STORAGE SOLUTIONS FOR ORACLE.
SCALABLE STORAGE INFRASTRUCTURE.
SCALED DOWN STRESS.

PICK ANY THREE.

SETTLE FOR EVERYTHING WITH NETAPP.

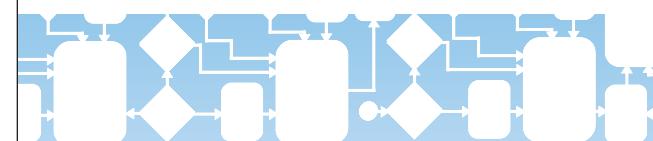
Find out why we're trusted by the world's greatest companies that run Oracle.

Visit netapp.com/bizapps



© 2007 Network Appliance, Inc. All rights reserved. Specifications subject to change without notice. NetApp and the Network Appliance logo are registered trademarks and Network Appliance is a trademark of Network Appliance, Inc. in the U.S. and other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.

Announcing Embarcadero® EA/Studio™



Join your processes and data with EA/Studio.

- Conceptual modeling
- Real-world process modeling
- Speak the same language as business users

Visit www.embarcadero.com/eastudio to download a free trial.



EMBARCADERO
TECHNOLOGIES

if it already had Oracle Database installed and configured. The presentation explores how Oracle can help you derive the most out of the exciting developments in the cloud-computing arena.

Oracle's Real Application Testing: How Real Is It?

Hanan Hit, *DBA InfoPower* 1:00–2:00

Oracle's RAT (Real Application Testing), which is part of the 11g release, allows businesses to quickly adopt new technologies while eliminating the risks associated with change. Real Application Testing combines a workload capture and replay feature with an SQL performance analyzer to help identify performance changes by replaying real-life workloads in a new database environment, and then helps to fine-tune them before production deployment.

In this presentation we describe the ways to record, replay, and analyze the database workload transactions. We discuss ways to assist in smooth migration from prior versions (10g), and the ability to test changes. We discuss different database connections methods like JDBC, PHP/OCI, BEQ, and Loopback. We conclude with a brief discussion of the benefits and drawbacks, and comparison to other database load-testing methods from the user and system perspective.

SQL Tuning: Reaching Recent Data Fast

Dan Tow, *Singing SQL* 2:30–3:30

In the simplest view of SQL tuning, we ask what number of rows and fractions of rows meet a given filter, or WHERE-clause condition. In this presentation I explain the usefulness of another consideration—is the filter correlated with the age of the rows? “Timely” (age-related, that is) filters have subtle advantages, as explained in this presentation, as conditions to use to reach the driving table.

Database Development Tasks with SQL Developer

Kris Rice, *Oracle Corporation* 4:00–5:00

SQL Developer provides database developers with a powerful tool for database tasks. Users can browse, create, edit, and delete Oracle database objects; create, edit, and debug SQL and PL/SQL code; manipulate and export data; run reports; and place files under version control. This session demonstrates activities including creating connections, creating basic objects, and interrogating the database. Participants see features including schema copy and compare, code templates, advanced formatting, source code control, and the latest features.

—Diablo—

Service-Oriented Data Management Using Oracle Data Integration

Nabin Bilas, *Oracle Corporation* 11:00–12:00

Data Integration is more than simply Extract, Transform, and Load (ETL); it is about turning data into agile information, re-usable across the enterprise. Your business needs to expand to keep up with growing demands, external pressures, and compliance mandates. At the same time, your IT struggles to manage its data-centric assets more efficiently. This presentation will highlight to enterprise architects, data architects, and data stewards what steps they can take to successfully

manage terabytes of complexity of their data-centric architectures, data warehouses, and information applications. It will define the benefits of embracing technologies in data services, real-time data warehousing, data governance, and master data management, along with real examples of enterprise-wide data integration solutions. Also included are details on Oracle's strategy for data integration, including Oracle Data Integration Suite and the role of BEA technologies going forward.

Oracle Database 11g: Security and Regulatory Compliance

Roxana Bradescu, *Oracle Corporation* 1:00–2:00

Data breaches, insider theft, outsourcing, data consolidation, and attacks targeting databases have made protecting mission-critical business data harder than ever. Additionally, government regulations and industry standards have raised the stakes for enterprises failing to implement necessary database controls. During this session, we will discuss how you can use Oracle database security solutions to protect business data from internal and external threats, and achieve regulatory compliance without changing existing applications.

What DBAs Need to Know About Data Quality

Michael Scofield, *ESRI* 2:30–3:30

A survey of the basic elements of data quality (currency, validity, reasonableness, accuracy, precision, etc.) and some discussion on the politics of DQ (who in the organization should be doing what to improve data quality). This presentation will give particular emphasis to a simple technique for detecting bad data, without the need to purchase a data quality tool.

Data Quality and the Functional Requirements for DQ Tools

Michael Scofield, *ESRI* 4:00–5:00

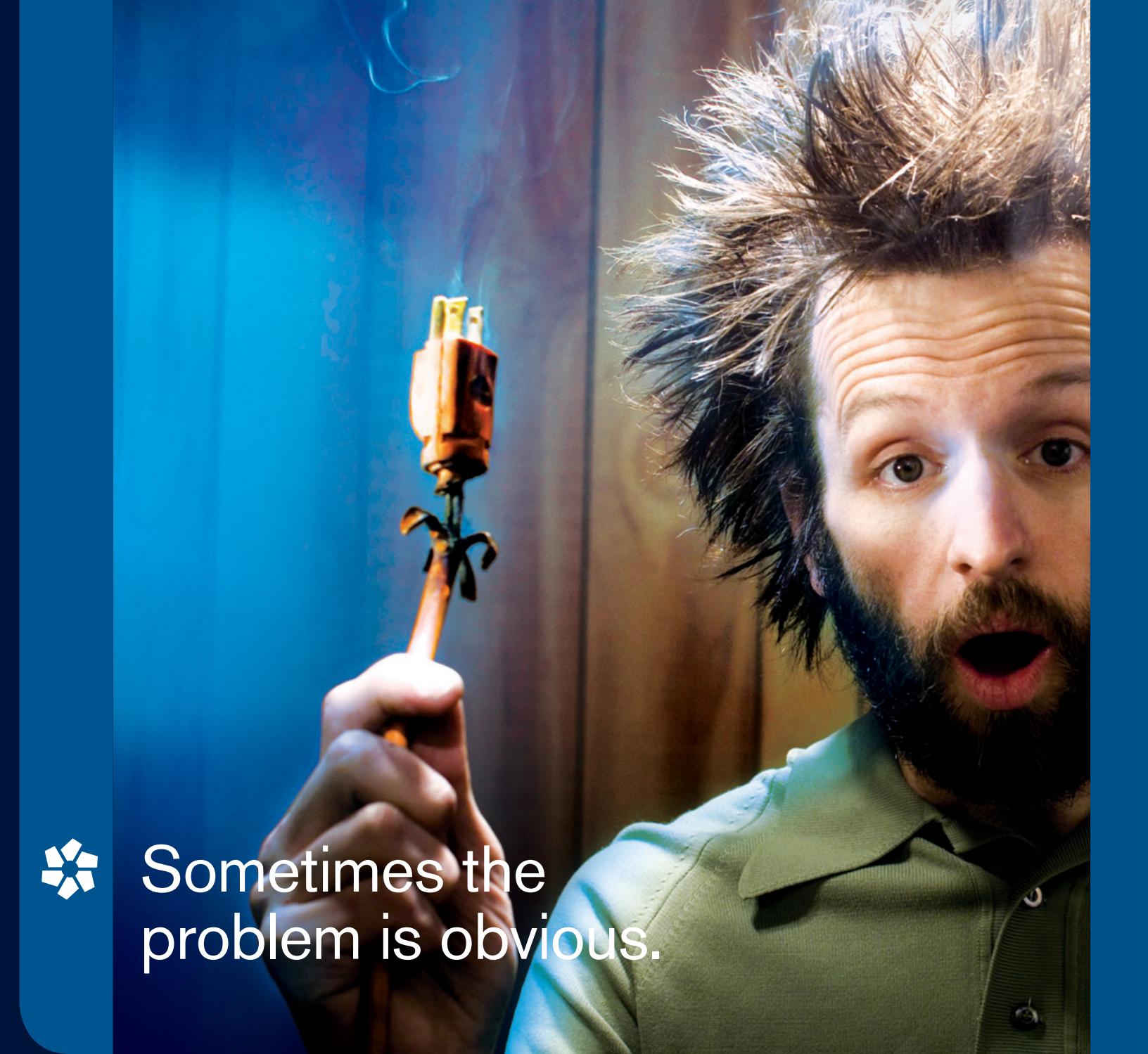
Many vendors offer tools for data quality survey, testing, and even “data scrubbing.” This presentation addresses the questions you need to ask before selecting a tool. Questions include: What are the expectations of what the tool will do? What kind of data (by subject, by timing, by structure) must the tool address? Is data examination a one-time event or an ongoing process? Where is the data? At rest or in a data flow? On premises or off-site? And, of course, in what technical environment?

It is important to understand the kinds of data problems that need to be detected and the general kinds of tests that can or should be performed (e.g., absolute tests vs. reasonability tests).

Topics discussed in this session include:

- The trade-off between a tool and human judgment
- When and where to do testing
- A survey of kinds of granular data tests
- Kinds of “reasonability testing”

Jonathan Lewis Training Days—details on page 7.



Sometimes the problem is obvious.

Usually, it's harder to pinpoint.

Amazing what you can accomplish once you have the information you need.

When the source of a database-driven application slowdown isn't immediately obvious, try a tool that can get you up to speed. One that pinpoints database bottlenecks and calculates application wait time *at each step*. Confio lets you unravel slowdowns at the database level with no installed agents. And solving problems where they exist costs a *tenth* of working around it by adding new server CPU's. Now that's a vision that can take you places.

A smarter solution makes everyone look brilliant.

Download your **FREE** trial of Confio Ignite™ at www.confio.com/obvious

Download our **FREE** whitepaper by visiting www.oraclewhitepapers.com/listc/confio

CONFIO
SOFTWARE

NoCOUG Fall Conference Schedule

November 13, 2008, at CarrAmerica Conference Center, Pleasanton, CA

Please visit www.nocoug.org for updates and directions, and to submit your RSVP.

Cost: \$40 admission fee for non-members. Members free. Includes lunch voucher.

| | |
|-----------------|--|
| 8:00–9:00 a.m. | Registration and Continental Breakfast—Refreshments served |
| 9:00–9:30 | Welcome: Roger Schrag, NoCOUG president |
| 9:30–10:30 | Keynote: <i>Testing, Testing, Testing</i> —Jonathan Lewis, JL Computer Consultancy |
| 10:30–11:00 | Break |
| 11:00–12:00 | Parallel Sessions #1 |
| | Auditorium: <i>Beginners Guide to Partitioning</i> —Jonathan Lewis, JL Computer Consultancy |
| | Tassajara: <i>Using Oracle in a Cloud</i> —Bill Hodak, Oracle Corporation Editor's Pick |
| | Diablo: <i>Service-Oriented Data Management Using Oracle Data Integration</i> —Nabin Bilas, Oracle Corporation |
| 12:00–1:00 p.m. | Lunch |
| 1:00–2:00 | Parallel Sessions #2 |
| | Auditorium: <i>Xtreme SQL Tuning: The Tuning Limbo</i> —Iggy Fernandez, Database Specialists |
| | Tassajara: <i>Oracle's Real Application Testing: How Real Is It?</i> —Hanan Hit, DBA InfoPower |
| | Diablo: <i>Oracle Database 11g: Security and Regulatory Compliance</i> —Roxana Bradescu, Oracle Corporation |
| 2:00–2:30 | Break and Refreshments |
| 2:30–3:30 | Parallel Sessions #3 |
| | Auditorium: <i>First-Failure Fault Detection and Diagnostics: 11g New Features and Novel Methods</i> —Jeremiah Wilton, ORA-600 Consulting |
| | Tassajara: <i>SQL Tuning: Reaching Recent Data Fast</i> —Dan Tow, Singing SQL |
| | Diablo: <i>What DBAs Need to Know About Data Quality</i> —Michael Scofield, ESRI |
| 3:30–4:00 | Raffle |
| 4:00–5:00 | Parallel Sessions #4 |
| | Auditorium: <i>Breaking Oracle: Creating Realistic Failures for Testing and Diagnostic Practice</i> —Jeremiah Wilton, ORA-600 Consulting |
| | Tassajara: <i>Database Development Tasks with SQL Developer</i> —Kris Rice, Oracle Corporation |
| | Diablo: <i>Data Quality and the Functional Requirements for DQ Tools</i> —Michael Scofield, ESRI |
| 5:00– | NoCOUG Networking and No-Host Happy Hour at Faz, 5121 Hopyard Rd, Pleasanton |

**Session descriptions
appear on page 24.**

RSVP online at www.nocoug.org/rsvp.html

NoCOUG

P.O. Box 3282
Danville, CA 94526

RETURN SERVICE REQUESTED

Jonathan
Lewis Training
Days—details
on page 7.