



Official Publication of the Northern California Oracle Users Group

NoCOUG

J O U R N A L

Vol. 24, No. 4 · NOVEMBER 2010

\$15

Let Knowledge Burst Forth at NoCOUG

Failure Is Not An Option

*An interview with Charles Hooper.
See page 4.*

Statistics—How and When?

*Advice from the Oracles!
See page 11.*

Become a Rock Star DBA

*Thomas LaRock tells you how.
See page 15.*

Much more inside . . .

Let Knowledge Burst Forth at NoCOUG

The picture on the front cover, taken by NASA's Wide-field Infrared Survey Explorer (WISE), shows a star-forming cloud filled with massive newborn stars. The cloud reportedly contains some of the most massive stars ever seen. Let knowledge burst forth at NoCOUG just like new stars burst forth in this star-forming cloud!

In the last issue of the *Journal*, NoCOUG's book reviewer Dave Abercrombie gave high marks to Thomas LaRock's new book, *Become a Rock Star DBA*. In this issue of the *Journal*, we bring you an excerpt from LaRock's book with the permission of the publisher. NoCOUG's own Chris Lawson makes an appearance at our fall conference on November 11 at the Oracle conference center in Redwood Shores. Lawson's presentation is a behind-the-scenes explanation of how experts really solve tough database performance problems, based on his experience of solving thousands of such problems. Lawson will show you how it's done and divulge his top ten tuning tactics.

I hope you enjoy this issue of the *Journal*, and I hope to see you at our fall conference on Thursday, November 11, at the Oracle conference center in Redwood Shores. ▲

—Iggy Fernandez, NoCOUG *Journal* Editor

Table of Contents

President's Message	3
Interview.....	4
Book Review	9
Ask the Oracles.....	11
Book Excerpt.....	15
SQL Corner	20
Sponsorship Appreciation.....	22
Session Descriptions.....	24
Conference Schedule.....	28

ADVERTISERS

Avior.....	3
Delphix.....	7
Rolta TUSC.....	13
Confio Software.....	23
3PAR.....	25
Database Specialists, Inc.	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 2011 issue is November 30, 2010. Article submissions should be made in Microsoft Word format via email.

Copyright © 2010 by the Northern California Oracle Users Group except where otherwise indicated.

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

2010 NoCOUG Board

President

Hanan Hit, HIT Consulting, Inc.
hithanan@gmail.com

Vice President

Jen Hong, Stanford University
hong_jen@yahoo.com

Secretary/Treasurer

Naren Nagtode, eBay · nagtode@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

Claudia Zeiler · girlgeek@wt.net

Director of Conference Programming

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

Director of Marketing

Jenny Lin, CBS Interactive
jenny.lin@cbs.com

Training Day Coordinator

Chen Shapira, Pythian
cshapi@gmail.com

Track Leader

Omar Anwar, GSC Logistics
[oanwar@gwmail.gwu.edu](mailto: oanwar@gwmail.gwu.edu)

Members-at-Large

Noelle Stimely, UCSF · noelle.stimely@ucsf.edu
Scott Alexander
alexander_scott@yahoo.com

NoCOUG Staff

Nora Rosingana

Book Reviewers

Brian Hitchcock, Dave Abercrombie

ADVERTISING RATES

The *NoCOUG Journal* is published quarterly.

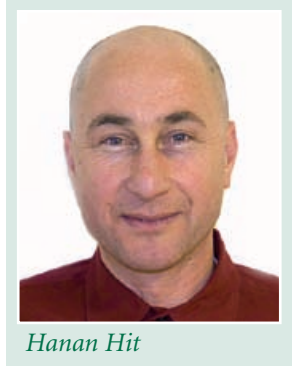
Size	Per Issue	Per Year
Quarter Page	\$125	\$400
Half Page	\$250	\$800
Full Page	\$500	\$1,600
Inside Cover	\$750	\$2,400

Personnel recruitment ads are not accepted.

journal@nocoug.org

Passing the NoCOUG Baton

by Hanan Hit



Greetings Oracle professionals! This is the fourth quarterly issue of the NoCOUG Journal for 2010 and my last presidential message of my two-year term. It is hard to summarize the past two years in such a short address. Let me just say that it was one of the most gratifying positions that I have held in my entire career—all the more so considering our success in the face of the tough economy and declining numbers from our treasurer's quarterly reports.

While my role as president is done, I consider it an honor to be a member of this group and will continue to contribute in the future.

There are so many people that I would like to thank, starting with former president Roger Schrag, who presented me

with the opportunity to run for the position in 2008. I would also like to give special thanks to two people who have been pillars of support for this organization and who will be stepping down from the NoCOUG board. Joel Rosingana served as director of membership as well as past president and vice president, to name just a few of his positions. Joel embodies the history of the NoCOUG organization and helped build the very foundations that have made this organization such a success. He leaves behind a strong organization; we wish him the best in his future endeavors. I would also like to thank Nora Rosingana, who served as a staff member and did a great job over the years assisting us with the financial reports, conference registration, and many other jobs. I wish them all

(continued on page 26)

Confused with all the
acronyms of Single Sign On?

AVIOR
Simple Solutions to Complex Problems



thinSSO

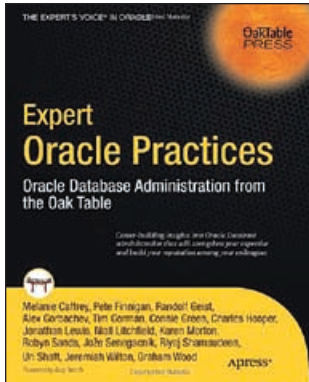
Federated **Single Sign On** for
Business Applications

- First to offer Zero Sign On out of the box for Oracle Applications
- Implementation included in product costing
- Free onsite prototype in 8 hours; test to production cycle in 3-4 weeks
- No duplication of user base; minimum administration effort
- Enable seamless navigation across applications and improve acceptability among your business users

Request a free prototype
(209) 757 8786
info@aviorinc.com

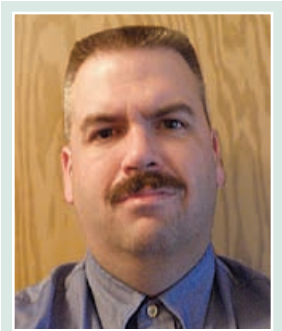


www.aviorinc.com



Failure Is Not An Option

by Charles Hooper



Charles Hooper

Charles Hooper is the IT Manager and an Oracle database administrator at K&M Machine-Fabricating. He is a member of the Oak Table Network and co-author of Expert Oracle Practices: Oracle Database Administration from the Oak Table. He blogs at <http://hoopercharles.wordpress.com>.

Tell us about your work, experience and what you like best about Oracle.

I am the IT manager and an Oracle DBA at K&M Machine-Fabricating, Inc., where I have worked since early 2000. I was state certified to teach computer science and mathematics courses, and was initially hired as a computer programmer, trainer, computer troubleshooter, and general “make it happen” person. The “make it happen” philosophy (also known as the “failure is not an option” philosophy) is one of the driving factors behind the desire for continuous improvement, and probably one of the reasons why the job responsibilities of IT manager and Oracle DBA were offered to me in mid-2002. Since that time my responsibilities have shifted a bit. I still try to maintain the “make it happen” philosophy, although my role has shifted a bit away from that of developer to more of primary Oracle DBA, network administrator (I was close to the point of taking the Netware 4.1 and/or 5.0 Certified Netware Engineer exams when I was hired), IT manager, ERP trainer, and—to the best of my abilities—senior “answer guy.”

What do I like best about Oracle Database? I remember thinking to myself when I first started working with Oracle Database, “Wow, Oracle sure is hard to use; I bet that I could just build a program that accomplishes the same task—and if I had to I could just use the Microsoft Jet engine to build an Access-compatible database on demand for special storage needs.” After reading a couple of very well-written books that were specific to Oracle Database, I quickly determined that Oracle Database was much more than just a bit bucket, a data dump for lack of better term. While the initial learning curve might be a bit steep, once that initial hurdle is overcome, Oracle Database is quite easy not only to use from a developer’s point of view, but also fairly easy to maintain from a DBA’s point of view. It also helps that Oracle Database is typically quite a stable platform for hosting data.

You are known for your very detailed and critical book reviews. What books can you recommend to NoCOUG members?

Great question, but before I answer it, I will list some of what I believe to be the most critical key targets for a technical book:

- A technical book must be nonfiction—a technical book should describe technically accurate details of an event, action, or object.
- A technical book is a permanent record of an event, action, or object. With that in mind, the contents of a technical book should require a significant amount of research, testing, and retesting to verify the accuracy of the book’s contents, not just in a Linux virtual machine but also in physical hardware on Unix, Linux, and Windows. If the book’s technical content only applies to a Linux virtual machine environment, readers will likely experience trouble implementing the suggestions in a production environment.
- A technical book that describes a feature should correctly identify which Oracle release version the feature works with.
- A technical book that describes additional cost options, such as partitioning, AWR, and Database Replay, should indicate that the feature must be purchased and/or is not available for the Standard Edition/Standard Edition One.
- A technical book that describes Oracle initialization parameters must avoid mentioning the hidden initialization parameters unless a warning is provided that such parameters should not be modified without first consulting Oracle support. Mentioning that warning 100 pages after the description of the first hidden parameter is of limited help. Mentioning hidden parameters that do not exist in readers’ Oracle databases probably is not a good idea.
- A technical book that states feature X is 25% (or any percent) more efficient than feature Y should provide a reproducible test case so that the reader is able to test the efficiency improvement in their environment as well as adapt the test case to more accurately model a problem in the reader’s database configuration.

- A technical book should provide forward and backward references to help readers locate detailed or advanced adaptations/implementations located elsewhere in the book. Also helpful are references to external resources found in other books, Metalink (MOS), or specific web-sites.
- A technical book that contains “recipes” or “lab projects” should provide all of the critical configuration elements within a couple of pages of the recipes or lab projects. Forcing the reader to page through hundreds of pages to piece together the lab project is unacceptable.
- A technical book that advertises “undocumented secrets” or “definitive” or “complete” on the front cover had better contain actual undocumented facts (not in the Oracle documentation, not in Metalink/MOS, and not found easily through an Internet web search) and/or contain the complete description of the topic.
- A technical book should look like a technical book: 10-point or smaller font size, small margins, no cartoon drawings, well-organized topics and sub-topics, few if any spelling or grammar errors, helpful table of contents and index, meaningful chapter names, few unnecessary distractions, etc.
- A technical book should draw the reader in—trick the reader into reading the book cover to cover even when the weather is better suited to outdoor activities.
- A technical book’s author(s) should have a reputation for producing accurate information about the subject—a Google search is helpful.
- A technical book should cover the current release of the product (Oracle Database), as well as older but still-supported releases.
- And last, but not least, a technical book should convey useful content that helps people just starting to learn Oracle Database, as well as people who have used the product for years and are just looking for new ideas or a way to help maintain their knowledge. A technical book that introduces nothing new is of limited value.

I only just recently realized that most of the above bullet points are inspired by what is found in Tom Kyte’s Oracle books, that his books subconsciously set the benchmark by which I have been measuring all other books. Correctly or incorrectly, the above bullet points effectively eliminate a number of books from consideration. I attempted to list several of the books that I would recommend in the following blog article: <http://hoopercharles.wordpress.com/2010/01/16/book-library-finding-motivation/>. Just because a book is not on the list does not mean I would not recommend the book, but I will only recommend books that I have actually read from cover to cover.

Some people read technical books like they read a novel; others read just the chapters they need. Do you have any advice on how to get the most learning from technical book reading?

A large percentage of my knowledge was derived from books and experiments that followed (testing prior to implementation) what was read. One of the useful pieces of information that I learned in an introduction to psychology course in college was that in order to effectively learn something new, the new material must be associated with something that is already known. When reading a book, keep asking yourself about the material, “Do I know this to be a fact based on my prior knowledge.” If the answer is either “maybe” or “no,” take a couple of minutes and try to work out from your prior knowledge how the author arrived at the conclusions printed in the book. It might be that the author is simply restating something you already know, just using different terminology. It might be that the author is introducing something entirely new and has provided you with knowledge that will be helpful in the future. It might also be the case that the author is simply making a wrong assertion, one that cannot possibly be true or one that is only true in very limited circumstances. With some books it might seem as though you are working through one very long true-or-false test; however, if you are to retain the information, it is important to relate the information to what you already know, and this is just a step in the process. I also try to take notes (which are later typed) on a sheet of paper while reading, indicating the page number of the original material. The notes usually are not direct quotes but instead are paraphrases of my understanding of what was just read.

Do you have any official certification? What do you think about the Oracle certification exams?

I do not have any official Oracle Database certifications, and I have never seriously considered taking an Oracle certification exam. There are a variety of reasons why I have not considered taking an Oracle certification exam (I do not mean to devalue the certifications that others have achieved through hard work):

- Passing a certification exam is largely dependent on the test taker’s ability to recall simple facts (the simplest level of Bloom’s Taxonomy of the Cognitive Domain: http://en.wikipedia.org/wiki/Bloom's_Taxonomy) that the test author believed to be important. Passing the certification exam is largely dependent on the test taker’s ability to, in some cases, guess what the test author believes to be the best answer—what if the test author’s answer is incorrect? More importantly, are the questions on the test actually applicable to what you will perform as a DBA?
- Reading through various Internet forum threads, it is not uncommon to see people requesting brain dumps for various Oracle exams as well as people who have mentioned obtaining OCP (or higher) certifications and then a couple of sentences later asking relatively simple questions that could be answered by a quick check of the Oracle Concepts Guide. This raises the question of what the successful completion of an Oracle certification exam really indicates. Does it indicate the ability of the person to pass a test or does it indicate the ability to perform analysis, synthesis, and evaluation

(the more advanced levels of Bloom's Taxonomy of the Cognitive Domain). I am a member of the Information Technology Advisory Board for a local college. That board provides advice to the college regarding what students need to achieve in order to graduate. Requiring students to obtain certain computer-related certificates was mentioned, and then the idea was softened a bit when I mentioned the person I interviewed for a job a couple years earlier, who had more than 20 computer certificates yet could not answer some of the simple questions that required higher-ordered thinking skills.

- An Oracle certificate is typically tied to a specific Oracle release. Does that mean that a person who obtained an OCP certificate for Oracle 8i but never attempted certification for a later release is unqualified to support Oracle Database 11.2.0.2? What about an OCM who passed the Oracle 9i exam? The person might be qualified or might not be qualified to support such a database—thus an out-of-date certificate really does not say much about the person, except that the person could pass an exam at some point in time.

You always test your ideas and present very clear test cases on your blog. I noticed that not all DBAs do that, partially because they find that building test cases is too time consuming or confusing. Do you have advice on how to test ideas?

It is far too easy to invest many hours into building a test case to demonstrate or explore a simple concept or model a particular problem that is being encountered. I recently had the opportunity to attend a session by Jonathan Lewis that was titled “The Beginner's Guide to Becoming an Expert,” a session that emphasized the use of test scripts to investigate various types of unexpected behavior when that unexpected behavior is found. Using examples from Jonathan's session, that unexpected behavior might include noticing a reduction in the redo generated when an update statement is driven through an index access path, for instance. Another example from Jonathan's session included the unexpected behavior where a significant amount of redo is generated when a bitmap index is updated, compared to the amount of redo generated when a b*tree index on the same column is updated, and then finding that the amount of redo generated for the bitmap index drops significantly when the Oracle Database is upgraded from 9.2 to 11.1. Creating a test script that models a particular situation allows that problem to be investigated in isolation and then repeatedly analyzed as one variable is changed at a time; the change might include modifying an initialization parameter, submission of different initial bind variables, execution of the script against a different Oracle Database release version, or any number of other small-scale changes.

One of the key concepts in the creation of test cases is to compare the intended end result of the test case with the starting point and then identify logical transformation stages between the two end points. At each transformation stage, confirm that the preceding step logically leads to the next step and then determine if there is any chance that a different result might be obtained given the same input. Work backward from

the last transformation stage to the first, and ask yourself if there is any other possible input that might have caused the outcome. It takes practice, a fair amount of research, and sometimes trial and error to discover the multiple potential cause-and-effect relationships.

I recently used this technique to build a test case for an OTN thread to discover why an index was not used to retrieve three rows from an 8-million-row table. The “given” information was that the index was used if an always-false comparison of two literal VARCHAR values was removed from an OR condition in the WHERE clause (<http://hoopercharles.wordpress.com/2010/09/22/test-case-showing-oracle-database-11-2-0-1-completely-ignoring-an-index-even-when-hinted/>). The first step in building a test case—assuming that the production environment where the problem was found is not available for use—involves building a model of the source data. If you are familiar with the real data that causes the unexpected behavior, build a model that simulates the real data; otherwise assume that the data to be returned is scattered among the other rows in the table data. In my test case, to simplify the calculations a bit, the number of rows to be inserted into the test table was increased to 10 million. Before building the data model, determine what might cause a full table scan to be performed rather than using an index access path: high clustering factor for the index, poor cardinality estimates (might be caused by out-of-date object statistics), incorrectly gathered system statistics, poorly specified optimizer parameters, sub-optimal index definition (might require an index skip scan or return a large number of rows that are later eliminated at the table level), optimizer bugs, maybe the full table scan really is faster than an index access path, etc.

Now that we have identified some of the causes for a full table scan, our test case may be used to experiment “what if?” and “how would I know if that was the cause?” To determine if the clustering factor was the problem, I might first need to know how the optimizer uses the clustering factor, and if I am able to artificially adjust it to see if a change in the value makes a difference in performance. To determine if the cardinality estimates are a problem, I might use DBMS_XPLAN.DISPLAY_CURSOR to retrieve the estimated cardinality and actual number of rows returned by every operation in the execution plan. I might remove the comparison of the two literal VARCHAR values to see how that change affects the cardinality estimates or see if changing one of the literal values causes a change—one of those changes just might also allow an index access path to be used. I might try to use a hint to force the use of the index (hints are, after all, directives and not suggestions), and if that does not work I might analyze why the hint was invalid. If the hint worked, after verifying that a performance improvement was achieved, I would likely consult a 10053 trace file to determine why the index access path was not automatically used—was the cost too high, or was the index access path cost not even calculated?

Once the cause of the full table scan is determined, additional what-if statements should be considered. What if the index hint worked, am I able to permanently add that hint to the SQL statement—are there any cases where adding that hint

(continued on page 8)



Delphix virtualizes database infrastructure to provide complete, fully functional databases that operate in 1/10th the space, with improved agility, manageability, and performance.



(continued from page 6)

might lead to other performance problems later? If the cardinality estimates are incorrect, is there anything that I am able to do to fix the cardinality estimates? If the optimizer parameters are poorly specified, am I able to adjust those parameters—and if I do, what might be the impact on the other SQL statements that depend on the current initialization parameter values for optimal performance and costing calculations? Are we able to modify the application that submitted the SQL statement to remove the unnecessary literal comparison?

Once the desired outcome is reached, work backward through the steps and determine where this approach might go wrong. Is it reasonable to conclude that the previous step always results in the next step? In short, test cases are not terribly difficult to construct once you learn what might cause a transformation step to simply lead to the wrong next transformation step.

Oracle Database behavior changes from one release version to another—and sometimes from one patch level to the next. Save the test case scripts once generated, because the behavior that is demonstrated by the test script might only be true for a small number of Oracle Database releases. My test case script for generating a deadlock on Oracle Database 11.1 and 11.2—while the same test case script simply generates an expected enqueue on Oracle Database 10.2—is just one example of the usefulness of test cases to demonstrate Oracle behavior.

You love posting quizzes on your blog, many of them based on suspicious “facts” you’ve read in books. Do you know the answer to the quiz before you publish it? Do you learn new facts from discussions with your readers?

The quizzes on my blog typically include references to books, commercial websites, forum posts, and blogs that are found through Internet searches. The quizzes serve to highlight some of the frustration that people experience when trying to find the correct answer to specific problems that they are experiencing. Suppose that you wonder how something works and perform a search using your favorite search engine. The first page found on the search engine seems to answer your question, yet how do you know if the answer is correct?

The questions that I include in the quizzes are typically reworded statements of fact found in one of the referenced articles. I typically have a general answer in mind when I write the question—something similar to “false, because a simple test case demonstrates different behavior.” I know that there are much more knowledgeable people than me in the Oracle community, and occasionally those people participate in the quizzes and provide an answer such as, “true in these circumstances and here is a test case that shows the answer as being true” or “false because of reasons X, Y, and Z.” The quizzes provide a form of peer review when peer review is not otherwise permitted for the original referenced article.

I try to learn from all of the answers provided by readers of the quizzes. Sometimes the answers provided by readers include keywords that trigger additional lines of thought, such as Niall Litchfield’s response that a SQL statement found in a book, which was intended to observe blocking locks “isn’t RAC aware.” Niall’s answer provoked additional ideas regarding the

validity of the SQL statement. A reader’s answer might not be “right” based on my understanding, but that does not mean that the answer is “wrong” either—there are opportunities to learn from every answer that is not simply a “yes” or “no” answer.

Do you have a specific tuning methodology that you can share?

Answering that question could take a chapter or two in a book . . . and they would certainly not be short chapters. I recently had the opportunity to attend one of Tanel Poder’s presentations, titled “Back to Basics—Choosing the Starting Point of Performance Tuning and Troubleshooting Wisely.” That presentation was excellent and it is one that I have no trouble recommending. The basic idea is to use tools with the necessary precision for the task at hand. If you have a problem with a single SQL statement, using an hour-long AWR or STATSPACK report probably will not offer enough precision. Even an AWR report with a snapshot duration that exactly matches the execution time of the SQL statement might not be sufficient; as such a report will include information that is not specific to the SQL statement of interest. Likewise, dumping the stack for a session’s process will likely be of limited value if we are trying to determine why the instance-wide disk performance is slower than expected.

Suppose I have a performance issue and want to ask for advice for OTN. What’s the best way to ask? How do I know which answers are actually correct?

One of the important tasks when posting requests for help with a performance issue is correctly stating the problem being experienced and accurately describing the Oracle environment (release version, altered initialization parameters, etc.); in some cases accurately describing the server platform and in some cases providing a test case with DDL and DML statements to create sample tables with demo data is absolutely necessary. It also helps to accurately state the steps that you have previously taken to resolve the performance issue. Randolph Geist created a template (<http://oracle-randolf.blogspot.com/2009/02/basic-sql-statement-performance.html>) for posting a SQL performance tuning request to a forum, such as OTN.

One of the benefits of posting a question to a site such as Oracle’s OTN forums is the opportunity for peer review of answers by people with diverse backgrounds. A provided answer might be “right,” but only when certain conditions are present in the database environment, so one of the benefits of peer reviews is to identify whether the controlling conditions apply to the original poster’s question. At times it really is hard to determine which answers are correct and which answers are not. Provided test cases, past forum activity of the answer provider, as well as corroboration of the forum answers with answers provided by other sources all help to determine which answers stand the best chance of being correct. There is a quote that I first heard a couple of years ago, “trust, but verify,” but I think the saying, “test, but verify,” describes what must be done to determine which answers are the most correct for a given problem. ▲

Interview conducted by Chen (Gwen) Shapira

Oracle Database 11g—Underground Advice for Database Administrators

A Book Review by Dave Abercrombie

Details

Author: April Sims

ISBN-13: 978-1-849680-00-4

ISBN-10: 1849680000

Pages: 348

Publication Date: April 2010

Price: \$49.99 (list)

Publisher: Packt Publishing, Birmingham, Mumbai

Summary

Overall Review: This book has a very broad scope, and the author's passion for Oracle's extensive management toolset carries this book forward. Its expansive scope inspires you with the notion that you will learn to fully diagnose and manage all aspects of an Oracle database. Unfortunately, the book's lack of depth and its informal style create a sense of sloppiness, sometimes becoming confusing, inconsistent, and even misleading. Even the title of the book itself is a little confusing, emphasizing the word "underground," but this word is not explained. The word "underground" suggests obscure, secret, dangerous, or even subversive information, yet the book's contents are none of those. However, the book does provide useful pointers to other information sources. Skim this book to get an idea of what is possible in Oracle 11g, but be sure to study the recommended outside sources before taking this book's advice.

Target Audience: Junior DBAs.

Would you recommend this to others? No.

Who will get the most from this book? A junior DBA just starting a job.

Is this book platform specific? Yes, Oracle 11g.

Why did I obtain this book? I was intrigued by the word "underground" in the book's title. Who isn't fascinated by the potential for learning something special, obscure, and maybe even secret? As a 10g DBA eagerly anticipating upgrading to 11g, I was especially interested in diving deeper into important, yet perhaps less obvious, features of 11g.

Overall Review

April Sims's enthusiasm for all things Oracle permeates every section of her book. She directs this inspiration toward the junior DBA, as "she considers the first year as a DBA to be the most critical." You are in her primary target audience if "you find yourself in charge of an Oracle database but are unsure of what tasks you need to perform."

When to Step Away from the Keyboard

Sims's advice goes beyond the usual technical topics. Her first chapter focuses on professional development and collaboration. For example, she wants us to "communicate with other IT professionals, and pass on the information you learn to others." Readers of this journal will certainly appreciate this advice!

Hopefully, her target audience will be able to find and appreciate the wisdom she tries to share. Unfortunately, this book has a rough, unpolished feel that can distract and annoy. Not merely informal, this book is practically sloppy at times. For example, Sims does not clearly define "when to step away from the keyboard"; several interpretations are possible. Does this mean taking "an extra minute to look at the database alert log before attempting to bring up the [crashed] database" as in the opening example? Does it mean to join users' groups and attend conferences as on the second page? A section in this chapter advises "staying away from dinosaurs," which means people who are too "stuck in their ways." Perhaps this avoidance is what is meant by "stepping away," but if so, it is counter to this chapter's usual emphasis on collaboration, sharing, and communication. The intent of this chapter's title is unclear.

This first chapter claims to be about "attitude and philosophy," but no coherent, consistent philosophy is presented. For example, the book states that "you must know everything about the Oracle software," which seems inconsistent with its reminder that "a DBA will not know how to fix every problem." It points out that "DBAs must work closely with other IT members," yet also claims that it might help to "change your phone number to be non-published within your organization." Sims wants to help the reader gain confidence and knowledge, but also suggests that DBA skills are innate, since the "best DBAs in the business seem to have an underlying sense or gut feeling when something is wrong." Which is it, learnable or innate? Can such intuition be developed, and if so, how?

This chapter also includes some general checklists with useful ideas, but they seem to have nothing to do with the supposed chapter focus on "attitude and philosophy" or with the chapter title of "when to step away from the keyboard." To be sure, many of these checklist ideas are useful, such as typing the word "production" into a production shell to prevent accidental issuing of actual commands from a cluttered desktop. However, the lack of a clear focus, bordering on sloppiness, in this first chapter sets the tone for the rest of the book.

Maintaining Oracle Standards

Sims makes a pitch for standardizing database maintenance operations. She begins with an overview of Oracle's Optimal Flexible Architecture (OFA). This initial emphasis, and the words "Oracle standards" in the chapter title, may mislead the reader into expecting a focus on Oracle's own standards.

Instead, the chapter scope broadens to mean standardization of DBA tasks in general. It touches lightly on a hodge-podge of topics such as the X Window System, SSH, DBMS_SCHEDULER, cron, Oracle Enterprise Manager (OEM), the Intelligent Agent, My Oracle Support (MOS), Remote Diagnostic Agent (RDA), UNIX shell utilities, shell scripting, environment variables, and the Oracle Configuration Manager (OCM), with a brief tangent into missing temp files. Whew!

Tucked away among the disparate heap of topics presented in the chapter, one of the book's greatest strengths makes its first appearance: Sims provides very useful references to MOS documents, identified by their Doc ID number. Yet even within this strength, the way to retrieve a Doc ID is not described until four pages after they are introduced. Sure, this is a relatively minor editorial error, yet her intended audience might be confused for a couple of pages.

Tracking the Bits and Bytes

Sims rightly points out that a DBA must "know how data moves from one database component to the next," particularly at the transaction level. However, this summary of Oracle internals is nowhere near as useful as those provided by other books recently reviewed in this journal.

Sims writes that "locks keep exclusive use of that block until a commit or rollback occurs." This statement is wrong in several ways. First and most obviously, Oracle uses row-level locking, not block-level. Second, the lock is not "exclusive," since other sessions can still read the block in order to read or update other rows within the block.

11g Tuning Tools

This chapter is intended as a quick introduction to tuning, and Sims points out that "entire books are dedicated to just this one topic." Indeed, I suggest that the reader skip this chapter, and instead consult one of those other books. Many fine examples have been reviewed recently in this journal.

For example, this chapter starts with "hardware load testing and forecasting." Confusingly, it refers to a particular suite of third-party tools but shows screenshots for OEM instead. These screenshots show time-series charts for "user calls per second" during "a huge web load test" run on two different servers. The values on these two charts vary tremendously: the chart for the first server shows about 50 times more "user calls per second" than the second chart, and both contain many spikes. This Oracle metric is not described here, other than vaguely as "the basic numbers in this graph came from measurements inside the database." Without any explanation whatsoever, the book concludes that the second server, the one with fewer "user calls per second," would "perform as well or better" than the first server. This example confuses the intended audience in many ways. How is this particular suite of third-party tools related to the charts from OEM? What is the

metric "user calls per second," and why should we care about it? Why do we apparently want to reduce it by a factor of 50? Do the spikes matter, and how do we deal with them? How do we draw specific conclusions from these data?

In fact, the metric "user calls per second" is often used to model the workload, but not performance or response time as this chapter implies. A "user call" is just as it sounds: a call from the client application for the database server to do something, like an insert or update. Since in this example the second server showed a "user call" workload 50 times smaller than the first server, it appears to me that the two tests were not comparable. The first server was asked to do something 50 times more often than the second server was asked, and so these data seem to say nothing about the performance of either server. This example confuses rather than informs.

This chapter then briefly introduces Automatic Diagnostic Database Manager (ADDM), Automatic Workload Repository (AWR), and Active Session History (ASH), SQL Advisors, and Statspack. These introductions are very brief. For example, the ASH introduction does not define "active" sessions, does not address the strengths and weakness of sampling, and gives the wrong value for the ASH sampling interval. The curious reader will need to consult other sources to learn how to use any of these tools.

Other chapters

Several chapters cover topics such as high availability, Data Guard, Flashback, RMAN backups, and migrating to 11g. As is consistent throughout the book, pointers to outside information sources are very helpful and welcome. Many of these topics are probably more advanced than the junior DBA target audience is likely to encounter. It is not easy to satisfy all experience levels, as seems to be attempted here. For example, an overview of disk storage concepts is provided at a level appropriate to someone who knows nothing about RAID. At the other extreme, the book also provides a recipe for using transportable tablespaces to merge two Automatic Storage Management (ASM) databases into one. It seems unlikely that a junior DBA would attempt such an operation. Sims has obvious experience with these techniques. I am guessing that her contributions to these topics would be better served in a book that is more focused and that is targeted to the more advanced DBA.

Conclusion

This book is targeted to the junior DBA. Its scope is very ambitious, ranging from collaboration, shell scripting, installation, backups, high availability, upgrades, patches, troubleshooting, and tuning. With such a comprehensive scope, its coverage is understandably thin in spots. The book is also written in a very informal tone. Unfortunately, the book's lack of depth and its informal style create a sense of sloppiness, sometimes becoming confusing, inconsistent, and even misleading. This book should not be seen as a stand-alone resource, especially for the junior DBA. The book does succeed, however, at providing an outline of Oracle 11g features and how they might be used to manage a database. Perhaps its biggest strengths are the pointers to other more authoritative and

(continued on page 26)

Statistics—How and When?

Ask the Oracles!

Previously printed in the August 2007 version of the NoCOUG Journal. Based on a panel discussion at Hotsos Symposium 2007.



Mogens Norgaard: Lies, damned lies, and statistics. Some things never change, including old jokes...

Anyway, I've always been mystified by statistics gathering in databases until Dave Ensor, the utterly bitter and twisted Scotsman with the razor-sharp mind, ended a presentation at

the UKOUG conference in Birmingham some years ago with these words . . . and of course I remember them 100% correctly (smiling).

"Oh, and by the way, could you please stop gathering statistics constantly? I don't know much about databases, but I do think I know the following: small tables tend to stay small, large tables tend to stay large, unique indexes have a tendency to stay unique, and non-unique indexes often stay non-unique."

With those words, he finished the presentation. On the way out of the room I asked Anjo Kolk what he thought about this, and he told me it was very true. People gather statistics way too often.

A couple of weeks later I got a call from a large customer, who had discovered that they hadn't gathered stats on their ERP system with more than 1,000 users for two years.

Normally, I would have laughed at them and told them to get it going. However, since I had heard the above from Dave Ensor, I asked instead: "So how has the system been performing in that period?" and got the surprising answer that it had worked very, very well and that the users had been really happy.

I bet. Because if you can keep the stats constant there's a very high chance that your SQL statements will execute in the same way every day, meaning that the vast majority of users will have the same experience w.r.t. response times day in and day out, which is what they want: predictability is king.

The goal of the CBO development team is to make it 100% dynamic, which interestingly enough is the way of the SQL Server optimizer too, whereas DB2 has this notion of having the execution plan stored physically in the data dictionary to ensure stability and predictability.

If you think about what Dave Ensor was really saying, it makes a lot of sense not to surrender to the view that everything should be dynamic and Oracle knows best.

Since the above experience, I have advised many customers to stop analyzing, thereby creating a more stable environment overnight.

"Oh, and by the way, could you please stop gathering statistics constantly? I don't know much about databases, but I do think I know the following: small tables tend to stay small, large tables tend to stay large, unique indexes have a tendency to stay unique, and non-unique indexes often stay non-unique."

—Dave Ensor as remembered by Mogens Norgaard

However, there are, of course, good cases for re-gathering stats. They include situations where the amount or distribution of the data changes dramatically.

But instead of suggesting various "good numbers" for the percentage of rows changed in a table before re-gathering stats or percentage of tables/indexes to be analyzed, I'd like to suggest a different approach:

Monitor the changes in execution plans and/or performance for the individual SQL statements. If the execution plan for a given statement changes, you should examine whether it's for the better or for the worse. On the other hand, if the execution plan stays the same but performance becomes worse, you should examine the real reason(s) for this, and perhaps as a consequence re-gather stats.

That way, you'd leave stuff alone that works very well, thank you, and you'd put your efforts into exactly the things that have become worse. In the process, you have also saved a lot of hardware resources and minimized the load on the system because of stats gathering.

Torben Holm of Miracle A/S has a free tool called MirPlan, which will do some of this for you. You can contact me (mno@MiracleAS.dk) or him (thh@MiracleAS.dk) if you're interested. ▲

Mogens Norgaard is the CEO of Miracle A/S (<http://www.miracleas.dk>), a database knowledge center and consulting/training company based in Denmark, and is the co-founder and father figure of the Oak Table network. He is a renowned speak-

er at Oracle conferences all over the world and organizes some highly respected events through Miracle A/S, including the annual Master Class and the Miracle Database Forum. He is also the co-founder of the Danish Oracle User Group (OUGKD), and was voted Educator of the Year in Oracle Magazine's Editor's Choice Awards, 2003. Mogens can be reached at mno@miracleas.dk.



Jonathan Lewis: 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.)

“Monitor the changes in execution plans and/or performance for the individual SQL statements . . . and perhaps as a consequence re-gather stats. That way, you'd leave stuff alone that works very well, thank you, and you'd put your efforts into exactly the things that have become worse.”

—Mogens Norgaard

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\$sql_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. ▲

Jonathan Lewis is well known to the Oracle community as a consultant, author, and speaker, with more than 22 years of experience in designing, optimizing, and troubleshooting on Oracle database systems. He has been working on the follow-up to Cost-Based Oracle Fundamentals for the last five years but in the meantime has contributed a chapter on the problems of statistics collection to the latest Oak Table book Expert Oracle Practices: Oracle Database Administration from the Oak Table.



Christian Antognini: The package dbms_stats provides many features to manage object statistics. The question is how and when should we use them to achieve a successful configuration? It's difficult to answer this question. Probably no definitive answer exists. In other words there is

no single method that can be implemented in all situations. Nevertheless, let me share my thoughts with you in this area.

The general rule, and probably the most important one, is that the query optimizer needs object statistics that describe the data stored in the database. Therefore, when data changes, object statistics should change as well. As you may understand,

I am an advocate of regularly gathering object statistics. Those who are opposed to this practice argue that if a database is running well, there is no need to re-gather object statistics. The problem with that approach is that more often than not, some of the object statistics are dependent on the actual data. For example one statistic that commonly changes is the low/high value of columns. True, there are not many of them that change in typical tables, but usually those that change are critical because they are used over and over again in the application. So in practice, I run into many more problems caused by object statistics that are not up-to-date than the opposite.

Obviously it makes no sense to gather object statistics on data that never changes. Only stale object statistics should be re-gathered. Therefore it is essential to take advantage of the feature that logs the number of modifications occurring to each table. In this way we re-gather object statistics only for those tables experiencing substantial modifications. By default a table is considered stale when more than 10% of the rows change. This is a good default value. As of Oracle 11g, this can be changed if necessary.

The frequency of the gathering is also a matter of opinion. I have seen everything from hourly to monthly or even less-frequent gatherings as being successful. It really depends on your data. Nevertheless, when the staleness of the tables is used as a basis for re-gathering object statistics, intervals between two gatherings that are too long lead to too many stale objects and, therefore, the time required for the gathering could be too long. For this reason I like to frequently schedule the gatherings to spread out the load and to keep a single gathering as short as possible. If your system has daily or weekly low-utilization periods, scheduling gatherings during those periods is usually a good thing. If your system is a true 24/7 system, it is usually better to use very frequent schedules (many times per day) to spread out the load as much as possible.

If for some good reasons object statistics should not be gathered on some tables, as of Oracle 10g you should lock them. In this way the job that regularly gathers object statistics will simply skip them. This is much better than completely deactivating the gathering job for the whole database. Unfortunately, up to Oracle 9i there is no feature that can be used simply to achieve the same goal.

If you have jobs that load or modify lots of data, you shouldn't wait for a scheduled gathering of object statistics. Simply make the gathering for the modified objects part of the job itself. In other words, if you know that something has substantially changed, trigger the gathering immediately.

As of Oracle 10g, you should take advantage as much as possible of the default gathering job. In order for this to meet your requirements, you should check—and if necessary change—the default configuration. Since a configuration at object level is only possible as of Oracle 11g, if you have particular requirements for some tables, you should schedule a job that processes them before the default job. In this way the default job will simply skip them. Locks might also be helpful to ensure that only a specific job is re-gathering object statistics on those critical tables.

If a gathering leads to inefficient execution plans, there are two things to do. The first one is to fix the problem by restoring

“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. ... The biggest problem is that you need to understand the data.”

—Jonathan Lewis

the object statistics that were successfully in use before the gathering. The second one is to understand why inefficient execution plans are generated by the query optimizer with the new object statistics. For that purpose, at first you should check whether the newly gathered statistics correctly describe the data or not. For example, it is possible that sampling along with a new data distribution will lead to different histograms. If object statistics are not good, you know that the gathering itself—or possibly a parameter used for the gathering—is the problem. If object statistics



The advertisement features two logos at the top: a stylized 'R' inside a square with the word 'ROLTA' below it, and a diamond-shaped logo with 'TISC' inside. Below the logos, the text 'Enterprise IT Solutions' is prominently displayed, followed by the tagline 'The Value of Experience...'. A list of services is provided with checkmarks: Oracle (Applications & Technology), IT Infrastructure, Profitability & Cost Mgmt, and Rolta Software Solutions. At the bottom, the Oracle logo is shown next to the text 'Platinum Partner'.

ROLTA

TISC

Enterprise IT Solutions
The Value of Experience...

- ✓ Oracle (Applications & Technology)
- ✓ IT Infrastructure
- ✓ Profitability & Cost Mgmt
- ✓ Rolta Software Solutions

ORACLE **Platinum Partner**

are in fact good, there are two more possible causes. Either the query optimizer is not correctly configured or the query optimizer is making a mistake. Over the latter we have little control. On the other hand, we should be able to find a solution for the former. In any case, you should avoid thinking too hastily that gathering object statistics is inherently problematic and, as a result, stop gathering them regularly. ▲

Since 1995, Christian Antognini has been focusing on understanding how the Oracle database engine works. He is currently working as a principal consultant and trainer at Trivadis AG (www.trivadis.com) in Zürich, Switzerland. If he is not helping one of his customers get the most out of Oracle, he is somewhere lecturing on optimization.



Wolfgang Breitling: When thinking about or deciding on a statistics-gathering strategy, we need to keep the eventual goal in mind. Gathering statistics is not an end in itself. The sole objective is to give the optimizer all the information it needs to generate good access paths for the queries it is

confronted with. We also should keep in mind that the statistics are not the only input into the access path generation process. There are at least three major parts that play a role:

- a. The optimizer itself, including its built-in assumptions and heuristics
- b. The statistics
- c. The SQL to be optimized

Frequently I hear developers bemoan the demise of the rule-based optimizer and the (plan) stability or predictability it afforded them. The claimed chaos created by the cost-based optimizer can be traced back to lack of understanding as well as a lack of stability of parts a and b above. The pace of development of the CBO is frantic. Patchsets and even patches often bring with them a change in “behavior” of the CBO. Then what about the statistics? Oracle keeps hammering out the message that the optimizer needs fresh statistics in order to create optimal plans. “Stale” statistics are evil. So DBAs have implemented scheduled statistics-gathering jobs to run when there are resources to spare, on weekends, or even nightly. But do stale statistics really deserve the bad reputation they get? One could argue that stable statistics will foster some semblance of plan stability. If the optimizer doesn’t change—between upgrades or patches—the SQL does not change; and if the statistics do not change, there is no reason for the optimizer to generate a different plan. I allege that this is unequivocally true, provided all inputs remain the same. After all, the optimizer is not a game governed by chance but by a deterministic algorithm.

The reason we may not be able to collect statistics once and leave them be, at least until the next upgrade to Oracle and the optimizer, is that even the third part (c) above, the SQL itself, is often not as static as it seems. Think of bind variables, or time-related queries (e.g., “give me the sales figures for the past month”). The translation of “past month” into dates changes as time progresses. Therefore, we will need to update some statistics occasionally. It is imperative to know which ones.

“It is my firm belief that most scheduled statistics-gathering jobs do not cause much harm only because (most) changes in the statistics were insignificant as far as the optimizer is concerned—meaning that it was an exercise in futility.”

—Wolfgang Breitling

Beyond that, we ought to keep the other goal in mind when devising a statistics gathering strategy: “predictable performance”—closely tied to “plan stability,” which in turn is aided by stable statistics—unless the underlying data changes so dramatically that what used to be a good plan no longer is. But that is rare; business does not change that dramatically, so why would the data that describes it? It is my firm belief that most scheduled statistics-gathering jobs do not cause much harm only because (most) changes in the statistics were insignificant as far as the optimizer is concerned—meaning that it was an exercise in futility.

Gather statistics as rarely as possible and as frequently as necessary. With its automatic statistics gathering, Oracle ties the frequency to the (perceived) rate of change. But some tables may be completely replaced without materially outdating the statistics that describe them, while for others the statistics can become outdated—as far as some crucial SQL is concerned—by the addition, deletion, or update of just a single row.

Gather statistics consistently. Do not gather statistics in the scheduled job with a full compute and for a one-off refresh with an estimate—or vice versa.

Gather only as many statistics as necessary. E.g., do not gather a histogram on every column in the database just in case.

Do not be afraid to modify the gathered statistics, or even create some yourself, if the gathered statistics lead the optimizer to generate a bad plan. Remember, the statistics are not an end in themselves but are there so that the optimizer has information to create good plans. If it does not, you need to change the SQL, the statistics, or the optimizer.

Last but not least: if what you are doing now for statistics gathering does not cause you any grief, do not change it because of this article! ▲

Wolfgang Breitling is an independent consultant specializing in administering and tuning Peoplesoft on Oracle. The particular challenges in tuning Peoplesoft, often with no access to the SQL, motivated him to explore Oracle’s cost-based optimizer in an effort to gain a better understanding of how it works and use that knowledge in tuning. He has shared the findings from this research in papers and presentations at IOUG, UKOUG, local Oracle user groups, and other conferences and newsgroups dedicated to Oracle performance topics.

Now What Do I Do?

An excerpt from *Become a Rock Star DBA* by Thomas LaRock

*We bring you an excerpt from the book *Become a Rock Star DBA* authored by Thomas LaRock and published by Apress, March 2010, ISBN 1430227877. Dave Abercrombie, our book reviewer, had this to say when he reviewed LaRock's book for the August 2010 issue of the NoCOUG Journal: "Why should an Oracle DBA be interested in this book when its few technical examples are all for Microsoft SQL Server, not Oracle? Because LaRock offers sage advice for the significant nontechnical parts of our jobs. His focus on communication, business protocols, career growth, and selling yourself is unique and refreshing. Although LaRock's target audience is someone just starting a DBA career, his advice provides general benefits, and it can be especially valuable for those who hire DBAs. LaRock's chapter on starting a new job was especially intriguing: I'll bet that many of us have yet to do all of the things it suggests for the first week on a job! I found that chapter so helpful that I shared it with friends who were starting non-DBA jobs!" (http://www.nocoug.org/Journal/NoCOUG_Journal_201008.pdf) We are glad to have the permission of Apress to bring you a portion of the chapter that so intrigued Dave.*

Now that your dream has come true and you are the resident database administrator, what do you do first? Like any other job in the world you need to get started on . . . well . . . something, right? But what? Where do you focus your energies in order to best demonstrate your value to your organization? Knowing where to begin is crucial for your success, and this chapter is going to help you best determine how to get started, plan for your first one hundred days, and even decide who you should eat lunch with on a regular basis.

Your Initial Checklist

By now you should be sitting at your desk on what we will call day one. Your initial meetings with HR are over, you have gotten a tour of the place, and you are making certain you have the access you need to at least get started. Things like e-mail, for example. Oh, and access to the database servers they expect you to administer.

That is the very first piece of information you need, right? What servers and systems am I responsible for? Without that little nugget of knowledge, it is going to be very difficult to make any headway as you start your long, slow journey upstream.

Your initial checklist is divided into sections. Why? Well, because I like to put things into lists and categorize the lists if possible. It just helps me remember the bigger picture and not worry about missing any particular detail. I would encourage you to try the same thing and see if it helps, but everyone or-

ganizes themselves in different ways, so don't worry if you want to start your list differently.

Now, the checklist has three main sections. One section pertains to gathering information on what I simply call your stuff. Another section deals with finding information on the customer's stuff. Still another section you need to consider is what I call your action plans. Those three areas are where you will start to focus your efforts on day one; find your stuff, find your customer's stuff, and start making an action plan. So, a sample of a checklist might look like this:

1. Create a list of servers.
2. Check that database backups are running.
3. Spot check and verify that you can do a restore from one of those backups.
4. Build a list of customers.
5. List the "most important" databases.
6. List upcoming deliverables/projects.
7. Establish environmental baselines.
 - a. Server configuration check
 - b. Database configuration check
8. Compose your recovery plan (not your backup plan, your recovery plan).

Notice that my checklist is missing a few things that a lot of people will tell you are a must for DBAs to be doing daily. It doesn't include things like routine index maintenance, performance tuning, reviewing event logs, and so on. Sure, all of those things are necessary, but we are still on your list of items for day one. Everything I have mentioned so far is going to take you more than a few days to gather. If you get tied up troubleshooting some stored procedure on day one, then you are setting yourself up for a massive failure should a disaster hit and you have not had time to document your recovery plan.

Would you rather be a hero for telling that developer to stop writing cursors or a hero for informing a customer that you can have their database back up and running in less than 30 minutes? I know which choice I would make so soon after taking a new position.

On your first day, explain to your manager that you'll be gathering this inventory data first. By taking the initiative to perform due diligence first, you're showing them that your first mission is to safeguard their data, your job, and their job too. They probably won't be able to produce the inventory for you, and they're going to want it even more than you do. You will have plenty of time later on for the other stuff, and it will fall naturally into your environment baseline and subsequent action plans as you bring standards to your enterprise.

Let's look at why each of the items in the checklist is important, and important to address from day one. If today is your first day, you want to begin these tasks right now.

Create a List of Servers

Not sure I really need to explain this one, but you best get an idea of exactly what you are expected to administer. Chances are your initial list will not be complete, but it will give you an immediate baseline of reference. Trust me, at some point some person will walk up to you and start talking about a server you never knew existed. And they will be very confused as to why you have never heard of it, since they work with it all the time, there is a database there, and you are the DBA.

Do your best to gather as much information right away as you can. That way you will know more about what you are up against, and it will help you when it comes time to formulate your action plans, which will be very different depending on whether you have five or five hundred instances to look after.

I know what you are asking yourself. You are asking, “Self, how do I find out what I am responsible for?” I suggest you start with your immediate supervisor and go from there. The trail may take you to application managers and server administrators. For example, your boss might say that you are responsible for the payroll databases. But what are “the payroll databases”? You may need to run with that initial bit of information and do some detective work to track down the specific databases involved. Along your journey, you will be given an overview of the complexity that is your new home. Any detective work that you’re forced to do will pay off handsomely by deepening your knowledge and understanding of where you work.

If you are looking for a technical solution to finding databases, there are a handful of ways to get the job done. One of the simplest ways is to use the SQLCMD utility with the `-L` parameter to return a list of database servers that are broadcasting on the network. Since it is possible that some servers may not be found with that tool, you would be wise to still ask around when putting together your list.

And where do you keep such a list? I like to keep a lot of notes written down in a notebook. When I say “notebook,” I’m not talking about a computer. I’m talking about a physical, paper book that I can hold in my hands. Others prefer to put everything into Word documents and store them on their computer. The paper notebook works best for me because I have found that I learn better by rote than by typing. What I write with a pen stays with me better than what I type with a keyboard.

One more thing to mention would be the importance of having a list of servers you are not responsible for. There is a chance that you have some systems in your environment that are maintained strictly by vendors. If something goes wrong with one of those servers, it is important to know who is responsible for what. And if someone tells you that you do not need to worry about a server, my advice would be to get that in writing. Believe me, when disaster strikes, you had better be able to provide proof about the systems that are your responsibility.

If you are storing your information in a spreadsheet or similar document, then you can go back over time to better track how your environment is changing. Are you administering more servers or fewer? More databases or fewer? Are all your jobs running? Backups working? This information will surely help with your approval ratings. If nothing else, you’ll have the backup detail to be able to clearly show to your boss

how your job is changing and how your responsibilities are increasing. You’ll be better able to document your value to the company.

Check Database Backups

Now that you know what servers you are responsible for, your next question had better be, are the backups running?” Do not assume that everything is working perfectly. Dig into the details yourself and verify that each and every instance is being backed up. Oh, and that includes the system databases (master, model, msdb) as well as all user databases. Check that the backup files exist, that the directory they are being stored in is on a disk that has adequate space, and if there have been any recent failures.

You will also want to note the backup schedule for the servers and databases. You can use that information later on to verify that the databases are being backed up to meet the business requirements. You would not want to find out that the business is expecting a point-in-time restore ability for a database that is only being backed up once a week.

I cannot stress this enough, but if there is one and only one thing for you to focus on as a DBA, it is to ensure that you can recover in the event of a disaster. And any good recovery plan starts with having a reliable database backup strategy.

Verify That You Can Restore

Every now and then I like to do a spot check of my backups by taking some and attempting to restore. For me, I like to attempt a same-server restore as well as a restore to a different instance. Note that this is not the same as just restoring the file header in order to verify that the file is readable. No sir, I want to restore the entire database—it’s just my preference. I would not typically have time to verify each and every backup file, and would not attempt to do so. I just want to select a few backup files from a few servers and ensure that there are no issues.

Now, I said a “same-server restore” in the last paragraph and I want to make something very clear:

Be mindful when performing a same-server restore to a production server.

There, now I feel better. Same-server restores should be done only when you know it is safe to do so. How do you know if it is safe? Well, you could always ask someone. Or you could look to see if people are currently connected to the instance. Or, if your database is fairly small and manageable, and the restore should only take a few minutes at most, it should be safe. And of course, if it is a non-production server, then you can feel safer.

Which databases should you verify can be restored? You could focus your efforts on just about any group or set of databases. The real goal here is for you to become familiar with the restore process in your new shop as well as to verify that the backups are usable. Make certain you know all aspects of the recovery process for your shop before you start poking around on any system of importance. It could save you some embarrassment later on should you sound the alarm that a backup is not valid and what turns out to not be valid is really just your understanding of how things work.

Build a List of Customers

If you know what servers you are responsible for, then start asking who the customers are for each of those servers. Note that this line of inquiry can result in a very large list. And with shared systems, you could find that everyone has a piece of every server.

The list of customers is vital information to have. For example, if there is a need to reboot a server, it is nice to know who you need to contact in order to explain that the server will be offline for five minutes while it is rebooted. And while you compile your list of customers, it does not hurt to know who the executives are and which servers they are most dependent upon.

When you start listing the customers, you should also start asking about the applications and systems those customers use, and the time of day they are being used the most. You may be surprised to find some systems that people consider to be relatively minor are used throughout the day, while other systems that are considered most important are used only once a month.

There is an extra benefit to building this list: you get your chance to begin building a relationship with your customers. Chances are they will be flattered that you care enough to seek them out. Putting together your list gives you an excuse to reach out to your customers. Think of it as an ice breaker—an easy excuse for the two of you to meet.

List the “Most Important” Databases

While you gather your list of your important customers, go one step further and find out what their most important databases are. This could be done by either (1) asking them or (2) asking others, and then (3) comparing those lists. You could be very surprised to find that they do not match. You will be even more surprised to find how many people can forget about some of their systems and need a gentle reminder about their importance. As DBAs, we treat all databases with equal importance, but we recognize that some databases are indeed more important than others, especially given a particular time of day, week, or month.

For example, you could have a mission-critical data warehouse. Everyone in the company could tell you that this system is vital. What they cannot tell you, however, is that it is only used for three days out of the month. So, the database could be offline for 21 days and no one would say a word.

Another example would be a trading platform. This system could be used heavily for nine or ten hours each business day. But for the remaining hours of the day it is not used at all. Does that mean that when these systems are not used they are not important? No, it does not. What it does mean is that you are gathering more details about the systems. If 17 different groups mention some tiny database, but they consider it to be of minor importance, you can consider it very important because it is touched by so many different people.

Another factor here is recoverability. If the business requires you to be able to recover a database to a point in time, then you should consider that to be an important database as well, even if that customer sits across from you and says “it’s not that important.” I have lost track of the number of times a

customer has told me offhand that “the system really is not a big deal—we need to recover to within five minutes, and we use it all day long, but you don’t have to worry about it.” Yeah, right. Until a disaster happens, of course, and you are on the hook to put everything back, and quickly!

List Upcoming Projects and Deliverables

If there is someone around that can help you see the current and upcoming projects, it would help for you to know what is about to be dumped in your lap. You want to minimize the number of surprises that await you; knowing what projects are currently planned helps you to understand how much time you will be asked to allocate for each one. And do keep in mind that you will be expected to maintain a level of production support in addition to your project support, in addition to the action tasks you are about to start compiling.

You’ll also want to know which servers will be decommissioned in the near future so that you don’t waste time performance tuning servers that are on death row. Keep an eye on this angle as well.

Establish Environmental Baselines

Baselining your environment is a function that gets overlooked frequently. The importance of having a documented starting point cannot be stressed enough. Without a starting point as a reference, it will be difficult for you to chart and report upon your progress over time.

The idea of gathering baseline information on the servers you administer is to simply find out how many deviations you have from an ideal state. You have already done one baseline item—you have evaluated your database backups. Since this checklist item can become a little cumbersome in a short amount of time, try to focus on the basics.

For example, on a stand-alone database server with direct attached storage, your data files should exist on a separate drive from your log files. How many servers have that same configuration? And what about those disk drives? Is there a standard RAID configuration? (You may be surprised to find that servers are not always physically built the same.) How about the memory settings or the number of processors for the instances? How big are the data and log files, and what is the average CPU utilization?

Focus on a handful of big-ticket items to gather as your baseline information and keep the information going forward. You can add additional items over time as needed.

I would also advise you to streamline the ways in which you collect this information. For example, if you have one hundred servers, you really do not want to remote to all of them just to examine the memory settings for each one. You will need to learn how to efficiently gather your information, much of which can be collected using tools such as System Center Configuration Manager (SCCM) and Operations Manager. If your environment does not have those tools, then you may need to script out your baseline queries using PowerShell, T-SQL, or some alternative.

Alternatively, you could use a defined central management server in SQL 2008 to run queries against multiple instances at the same time. Or you could look to use policy-based manage-

ment to report on and possibly enforce configuration options. No matter what method you choose, the goal remains the same: to track changes over time. Once you've captured a picture of how your environment looks now, you can start to document how to return servers to their original state if something goes wrong.

Compose Your Recovery Plan

Notice how I said "recovery plan" as opposed to "backup plan." In your checklist so far, you have already verified that your database backups are running, started to spot check that you can restore from your backups, and gotten an idea of your important databases. Now is the time to put all of this together in the form of a disaster recovery (DR) plan. For each one of those databases listed as important, you should write down exactly the steps involved to recover should a disaster happen.

Make no mistake about it: should a disaster happen, then your job is on the line. If you fail to recover because you are not prepared, then you could easily find yourself reassigned to "special projects" by the end of next week. The best way to avoid that happening to you is to practice, practice, practice. Your business should have some scheduled DR tests perhaps once a year, but that should not prevent you from doing your own smaller DR tests on a more frequent basis.

Document each system and all the steps required to recover. Is the database in full-recovery mode? How frequently are you doing transaction log backups? Write down the backup schedule so that it is clear where the restore points are. If your customers expect recovery to the minute and you are in simple mode, then you are heading for a true disaster.

And don't forget about recovering from past days or weeks. If your customer needs a database backup restored from two months ago, make certain you know every step in the process in order to get that job done. If your company uses an offsite tape storage company, and if it takes two days to recall a tape from offsite, then you need to communicate that fact to your users ahead of time as part of your DR plans.

You Have Your Info, Now What?

Everything discussed up to this point should keep you busy until at least lunchtime on day one. OK, maybe it will take you a few days to get all of the information on your checklist. You may or may not have a lot of that information already available, so the amount of time it takes to gather everything will depend upon the current structure of your particular department in your shop. (By the way, get used to hearing and saying "it depends"; more on that later.) No worries, because you can transition into the next phase while you continue to gather your data.

Meet with Your Manager

While gathering the data, you should meet with your manager to discuss your preliminary findings and work together to prioritize the work that needs to be done. If you have found one server that is not running any backups, or the backups are failing, then that server must take precedence over everything else. I don't care which one of your "most important" custom-

ers screams at you on day one to fix a query they wrote that filled up a 70GB log drive, you must ensure you can recover from a disaster.

Continue to review your checklist with your manager and determine which items need to be done, which items need the most resources and effort, and which ones can wait for now. It is very important that you and your new manager agree upon the tasks and their priority before you get started. This will serve both of you very well; it gives you a chance to be measured from this point forward.

Should you and your manager disagree, however, with regard to your priorities, make sure you have that documented. It will serve you well when someone comes around asking why something specific has not been done yet, and you politely explain why you have been working on something else. Then, if there is still disagreement, you, the other person, and your boss can get together to reevaluate your priorities.

After you have met with your manager, you are going to constantly be measured. Everyone is going to want or need you for one thing or another. Every interaction you have with others will serve as a form of measurement, to be verified by something tangible produced by actions on your part. If you have that checklist, and your manager is aware of the checklist, you now have something tangible to deliver on starting from day one.

Think about how important this will be for you. In three or four weeks someone, somewhere, will ask your manager how things are working out with the new DBA. Your manager will be able to quickly affirm all the great work you have been doing (let's just assume you have been doing great work and not sitting on your arse all day). You will be able to track your progress easily, and eventually you can prepare a brief summary or presentation that details your progress over your first few months.

Meet with the Developers

If your company has in-house development teams, take some time to meet and greet them. They are going to figure most prominently in your success in the coming months. Find out what projects they are working on and what roadblocks they are currently facing, and see if there is an opportunity for you to offer assistance.

There is no question that the developers are going to keep you very busy. The developers will be the ones that look to push the limits with the technologies at their disposal. The end result will be an abundance of little training opportunities for you to take advantage of. Fill up the transaction log? Let me understand more about what you were trying to do. Filled up the entire log drive? Your remote query has bad performance? Your stored procedure with 27 nested cursors is taking too long? Here, let me help you with that, and perhaps we can find a better way to get the job done.

Will every developer want to be your friend? Absolutely not—it is rare to be somewhere where everyone gets along at all times. If you come across a developer that believes they know more than anyone else in the room, you can sit back and chuckle because you already know that the DBAs are the smartest, otherwise you wouldn't always be called to fix their problems all day long.

Meet with the Server Administrators

Depending on your shop, you may or may not have duties. Many DBAs focus only on the administration of the server instances and leave the O/S administration to others. If that is the case for you, then you need to make an effort to get to know this team. The server administrators have their fingers on the pulse of the whole company. They know which systems are most important, and they know where to find some extra hardware when you are in a pinch.

System administrators are also going to be the ones to build out your infrastructure. It is vital that you are aware of the hardware you are using, as well as the hardware that might be coming in the door. If your company is looking to implement a storage area network (SAN), they will not be asking you how to configure it; they will be asking someone on the server team. Since a SAN can and will have dramatic impacts on database servers, you need to be involved in those discussions. Ideally you will be invited to the table to discuss such things early on, and the more time you spend with the server team, the better chance you have to know about what changes are coming your way.

You are also going to frequently be needed to help the developers and the server administrators talk to and understand each other. Why is that the case? Well, think of yourself as the universal translator. Developers know code, and they know various facets of the business; and because they focus on those items daily, they are not spending time racking servers, installing routers, replacing hard disks, allocating space on the SAN, and so forth. Server administrators do those things, but they do not know how to code or build applications that will drive the business forward at the same speed as a developer.

Unfortunately these two groups rarely interface unless there is a problem (or an opportunity to point fingers at each other). That is where you come in. The developers can point their fingers at you and explain what the roadblock is at the moment, and you can turn and speak to the server administrators in a language they can understand. Without you, that does not happen, frustrations levels rise, and your shop ends up spending far too much money on hardware to solve issues that could be remedied by altering a few lines of code.

So by meeting with both groups, you continue to brush up on your language skills and help to keep the peace, and ultimately you can also help to reduce costs.

Meet with Your Customers

By now you have met with almost everyone possible, except for your actual customers. Actually, when you are a DBA, everyone is your customer, but let's focus on the actual end users of the applications and servers. These are the people that are going to give you a much different view than any other group.

Why is it so important to meet with them? Well, consider this story.

One day a man answered an ad for renting a bedroom above a garage at a family's home. The man was well kept and had a job (no one knew what it was), but was also always around. But he was a nice enough fellow and never late with his rent, and the family liked having him around. The family never noticed that the man always liked to go with them for rides in their car. It did not matter the destination; the man

would ask, "Mind if I take a ride?" and the family would let him.

Every time he rode in the car he would ask one or two questions about the car, but not so many as to make the family think anything was strange. He would simply work the questions into a conversation somehow. "So, think they have enough cup holders in here?" could lead to all sorts of comments about the design and functionality of the car.

Eventually the man told the family it was time for him to move out; he had found a new apartment to live at. Before he left, he finally told them what he did for a living: he worked for the car company. The reason he asked those questions was to get honest feedback about the car's design. The feedback they provided was much more reliable and honest than any survey could hope to be; the man observed the family actually using the product!

And so you need to do the same with your end users. Get to understand more about how they are using their tools, what their frustrations are, and what little things would help them (and therefore your business) be more efficient and productive. Go and ask your customers if you can take a ride in their car. You will be amazed at the things you will find.

Meeting by Eating

How are people who are shy or naturally introverted able to build and manage relationships with others? It is not an easy thing to do. In fact, it can seem overwhelming at times. But I have found one very common trait that everyone shares: everyone eats.

It is true. At some point everyone will eat. You may find them sitting down to have lunch. You may find them going to get a cup of coffee in the morning. Chances are you are going to see everyone you work with, at one time or another, eating. And there is a good chance that you tend to eat from time to time as well. What's to stop you from eating together every now and then? To break the ice, go to someone else's office or cubicle around 11:45, introduce yourself as the new guy, and ask if they know of any good places to grab lunch in the area.

I take a walk to get a cup of coffee every morning. I do not always buy a cup for myself, but at least one person in the group does. So why do I go? Because the ten minutes I spend with them gives me ten more minutes of interaction with them outside of work that I would not have if I sit in my cubicle.

Every now and then I find myself having lunch, or walking to get lunch, with others. And on days when I feel like I have been eating too much, I head to the gym with a different group of coworkers. In other words, I have been able to find ways to interact with different groups of people in different ways in an effort for us to get to know each other a little better. Does that mean we always get along? Absolutely not. What it does mean is that we make an effort to get along. ▲

From *Become a Rock Star DBA* by Thomas LaRock

SQL 101: Which Query Is Better?—Part II

by Iggy Fernandez



Iggy Fernandez

In the August issue of the *Journal*, I reviewed alternative formulations of the query “which departments have employees with salaries greater than a certain cutoff?”

```
VARIABLE salary_cutoff NUMBER

-- Correlated subquery

SELECT d.department_name
FROM departments d
WHERE EXISTS (
  SELECT *
  FROM employees e
  WHERE salary > :salary_cutoff
  AND department_id = d.department_id
);

-- the JOIN keyword

SELECT d.department_name
FROM (
  SELECT DISTINCT department_id
  FROM employees
  WHERE salary > :salary_cutoff
) e
JOIN departments d ON e.department_id = d.department_id;
```

I concluded that the “relational calculus” formulation using a “correlated subquery” was better because it gave more options to the query optimizer. In fact, the founder of relational database theory, Edgar (Ted) Codd, predicted that queries based on relational calculus would be easier to optimize. In his words: “*The relational calculus permits a user to request the data he desires by its properties. This is an ideal starting point for search optimization. The algebra, on the other hand, requires the user to formulate a sequence of algebraic operations that will generate the desired data from the data base relations. For queries other than very simple ones, the properties of the desired data tend to get hidden in the particular operation sequence (one of many possible ones) which the user selects. Therefore, starting from an algebraic source language, one has the choice of locally optimizing the execution of each operation (a very limited form of optimization) or tackling the difficult problem of analyzing sequences of such operations to discover the intended defining properties of the desired data.*” (“Relational Completeness of Data Base Sublanguages,” 1972).

This leads us to ask what exactly constitutes a relational calculus formulation. We can find the following definition on page 24 of the referenced paper by Codd:

Let the given “alpha expression” be $z = \{t : w\}$, where

1. $t = (t_1, t_2, \dots, t_k)$ is the target list;
2. $w = U_1 \text{ AND } U_2 \text{ AND } \dots \text{ AND } U_p \text{ AND } V$ is the qualification (a range-separable well-formed formula);
3. there are $q \geq 0$ bound variables in t ;
4. all p of the variables free in w occur in t .

Codd wrote his paper before the SQL language was created and therefore he used the language of mathematics. An “alpha expression” is simply what we now call a SELECT statement, (t_1, t_2, \dots, t_k) is the select-list in the statement, $U_1 \text{ AND } U_2 \text{ AND } \dots \text{ AND } U_p$ is the from-list (each U refers to one table and there are p of them), V is the where-clause, “free variables” are tables listed in the from-list, and “bound variables” are those that are listed in the from-list of a correlated subquery contained in the statement.

Since all p free variables must occur in (t_1, t_2, \dots, t_k) , the key point in Codd’s definition is that *the only tables listed in the from-list should be those that are referenced in the select-list.*

Let’s look at some examples. The first example is of managers who have an employee with a higher salary. The formulation below does not meet the requirements of the above definition because the variable “emp” is not referenced in the select-list. Note that the employees table is used twice in the from-list but each reference has been given a separate alias; for the purposes of the query, the two references are distinct tables.

```
SELECT DISTINCT
mgr.employee_id AS mgr_id,
mgr.salary AS mgr_salary
FROM
employees emp,
employees mgr
WHERE
emp.manager_id = mgr.employee_id
AND
emp.salary > mgr.salary;
```

The following is a more modern formulation of the above query and uses the JOIN keyword. However, it is only cosmetically different from the above query and it too does not meet the requirements of Codd’s definition.

```
SELECT DISTINCT
mgr.employee_id AS mgr_id,
mgr.salary AS mgr_salary
FROM
```



```
employees emp
JOIN employees mgr ON emp.manager_id = mgr.employee_id
WHERE
emp.salary > mgr.salary;
```

Next, we have a formulation that does meet the requirements of Codd's definition. The only variable in the from-list is "mgr" which is the only variable referenced in the select-list.

```
SELECT
mgr.employee_id AS mgr_id,
mgr.salary AS mgr_salary
FROM
employees mgr
WHERE EXISTS (
SELECT
*
FROM
employees emp
WHERE
emp.manager_id = mgr.employee_id
AND
emp.salary > mgr.salary
);
```

The next example is of employees who have a higher salary than their respective managers. The formulation below does not meet the requirements of Codd's definition because the variable "emp" is not referenced in the select-list.

```
SELECT
emp.employee_id AS emp_id,
emp.salary AS emp_salary
FROM
employees emp,
employees mgr
WHERE
emp.manager_id = mgr.employee_id
AND
emp.salary > mgr.salary;
```

The next formulation of the above query uses the JOIN keyword but it too does not meet the requirements of Codd's definition.

```
SELECT
emp.employee_id AS emp_id,
emp.salary AS emp_salary
FROM
employees emp
JOIN employees mgr ON emp.manager_id = mgr.employee_id
WHERE
emp.salary > mgr.salary;
```

Finally, we have a formulation that does meet the requirements of Codd's definition. The only variable in the from-list is "mgr" which is the only variable referenced in the select-list.

```
SELECT
emp.employee_id AS emp_id,
emp.salary AS emp_salary
FROM
employees emp
WHERE EXISTS (
SELECT
*
FROM
employees mgr
WHERE
emp.manager_id = mgr.employee_id
AND
emp.salary > mgr.salary
);
```

“The relational calculus permits a user to request the data he desires by its properties. This is an ideal starting point for search optimization. The algebra, on the other hand, requires the user to formulate a sequence of algebraic operations that will generate the desired data from the data base relations.”

Our third example requires information about employees as well as managers. It lists the identification numbers and salaries of employees who have a higher salary than their respective managers together with the identification numbers and salaries of the managers. The formulation below meets the requirements of Codd's definition.

```
SELECT
emp.employee_id AS emp_id,
emp.salary AS emp_salary,
mgr.employee_id AS mgr_id,
mgr.salary AS mgr_salary
FROM
employees emp,
employees mgr
WHERE
emp.manager_id = mgr.employee_id
AND
emp.salary > mgr.salary;
```

The formulation below uses the modern JOIN keyword. Even though references to algebraic operations such as JOIN, INTERSECT, MINUS, and UNION are not part of the language of relational calculus, I have to admit that it is only cosmetically different than the formulation above and does meet Codd's requirements in spirit if not in letter.

```
SELECT
emp.employee_id AS emp_id,
emp.salary AS emp_salary,
mgr.employee_id AS mgr_id,
mgr.salary AS mgr_salary
FROM
employees emp
JOIN employees mgr ON emp.manager_id = mgr.employee_id
AND
emp.salary > mgr.salary;
```

Summary

Relational calculus formulations of SQL queries give more options to the query optimizer than the corresponding relational algebra formulations. A key requirement of such a formulation is that the only tables listed in the from-list be those that are referenced in the select-list. ▲

Iggy Fernandez is the author of *Beginning Oracle Database 11g Administration* (Apress, 2009) and the editor of the NoCOUG Journal. He blogs at <http://iggyfernandez.wordpress.com>.

Copyright © 2010, Iggy Fernandez

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, Hanan Hit, at hithanan@gmail.com. ▲

Long-term event sponsorship:

CHEVRON

ORACLE CORP.

Thank you! Year 2010 Gold Vendors:

- 3PAR
- Confio Software
- Database Specialists, Inc.
- Delphix
- Rolta TUSC

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



TREASURER'S REPORT

Naren Nagtode, *Treasurer*

Beginning Balance

July 1, 2010

\$ 42,278.68

Revenue

Membership Dues	3,515.00	
Meeting Fees	400.00	
Vendor Receipts	6,000.00	
Advertising Fee	—	
Training Day	9,700.00	
Sponsorship	—	
Interest	4.87	
Paypal balance	—	
Total Revenue		\$ 19,619.87

Expenses

Regional Meeting	4,324.66	
Journal	2,153.09	
Membership	111.11	
Administration	1,160.00	
Website	—	
Board Meeting	617.28	
Marketing	—	
Insurance	—	
Vendors	136.40	
Tax	—	
Training Day	1,953.32	
IOUG-rep	—	
Miscellaneous	—	
Total Expenses		\$ 10,455.86

Ending Balance

September 30, 2010

\$ 51,442.69



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.

CONFIO 
SOFTWARE

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

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

NoCOUG Fall Conference

Session Descriptions

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

—Keynote—

Oracle OpenWorld 2010 Highlights

Sushil Kumar, Vice President, Oracle Corporation. . . 9:30–10:30

This keynote will showcase the latest Oracle updates—straight from Oracle OpenWorld 2010! In this session, you will learn about the newest technology innovations, including the Oracle Exalogic Elastic Cloud, Exadata Database Machine, Oracle Linux with the Unbreakable Enterprise Kernel, Oracle Solaris, Java roadmap, and more. Attendees will also hear how Oracle's next generation of applications, Oracle Fusion Applications, features the best from Oracle E-Business Suite, PeopleSoft, Siebel, and JD Edwards. Oracle Fusion Applications are built using industry standards-based Java middleware and integrated with business intelligence not just process automation.

—Auditorium—

Oracle Exadata: The World's Fastest Database Machine

Ron Weiss, Oracle Corporation. 11:00–12:00

Oracle Exadata is the only database machine that provides extreme performance for both OLTP and data warehousing applications at the lowest cost, making it the ideal private cloud consolidation platform. And the recent introduction of Oracle Exadata Database Machine X2-8 extends Oracle's family of database machines, offering customers a choice of secure, scalable, fully redundant consolidation platforms. In this session, we'll examine how Oracle Exadata delivers extreme performance for all database applications at the lowest cost.

Building Oracle Enterprise Linux Guests on a Virtual Box Windows Host

Ahbaid Gaffoor, Amazon. 1:00–2:00

In this session attendees will be shown how to build Virtualized Oracle Enterprise Linux hosts on Oracle's Virtual Box running on Windows 7. We will cover the use of the GUI to build a Linux host and then cover cloning of that host toward building a virtualized RAC environment. We will also cover creation of shared storage for use in a virtualized RAC environment, along with creation of bridged and host-only network interfaces. Attendees will be shown how to use the command-line utilities provided with Virtual Box. When you leave this presentation you will have all the tools needed to build virtualized environments on Windows, OS X, or Linux.

Building a Virtualized 11gR2 Linux RAC on a Virtual Box Windows Host

Ahbaid Gaffoor, Amazon. 2:30–3:30

In this session we continue with our virtualized environments and install an 11gR2 RAC in Virtual Box hosts. We will

review the steps needed to set up a RAC environment on a commodity desktop or laptop. Attendees will leave this presentation with the tools needed to get their own RAC environments running on their laptop or desktop machines. Tips and tricks to make your Virtual Host desktop or laptop perform better will also be shared.

Avoiding SQL Performance Regressions: New Techniques for Solving an Old Problem

Prabhaker Gongloor, Oracle Corporation. 4:00–5:00

This session covers how Oracle Database's SQL Performance Analyzer feature can be used to accomplish end-to-end testing of SQL response time for system changes such as optimizer statistics refresh, database upgrades, and patch sets. It covers various aspects of SQL performance testing and how SQL Performance Analyzer simplifies them—from capturing the workload to conducting trials for a given system change, comparing the performance data and execution plans, and identifying and tuning the SQL regressions. The session also covers advanced topics such as how to use SQL Performance Analyzer when the test system is not similar to the production system, and testing in a scoped and resource-controlled manner in the production environment. Finally, we cover how SPA was used successfully by DIRECTV, Bank of America, and NHN for above-mentioned system changes.

—Room 101—

Estimating Cardinality and Query Execution Cost: Use of Jonathan Lewis' CBO Methodology

Dave Abercrombie, Convio 11:00–12:00

To successfully tune a query, you need to be able to think like Oracle's query optimizer. Sure, it helps to have intuitive, general knowledge, based on experience and informal rules of thumb. But such general knowledge can take you only so far. To really understand the optimizer, you need to be able to reproduce its cost estimates. Jonathan Lewis has published reverse-engineered optimizer algorithms in his book *Cost-Based Oracle Fundamentals* (Apress 2006). In this presentation, Dave Abercrombie will lead you through those algorithms using real-world examples. You will gain a degree of understanding and confidence with the Oracle optimizer that will improve your tuning skills. This presentation builds upon previous presentations on physical data storage and reading execution plans.

Extreme Performance Tuning

Chris Lawson 1:00–2:00

Editor's Pick

CAUTION: Do not attend this session if you wish to hear a fluffy marketing spiel designed to sell some fancy new system,

or if you want to see abstract 3D charts that cause your eyes to glaze over within 5 seconds. This presentation is a practical, down-to-earth explanation of how experts really solve tough performance problems. Based on his experience solving many performance problems, Chris Lawson will show you how it's done and divulge his top 10 tuning tactics.

Reliable Datagram Sockets and InfiniBand

Hanan Hit, HIT Consulting 2:30–3:30

Reliable Datagram Sockets (RDS) over InfiniBand, which initially was supported in 10g Release 2 in order to support the cluster interconnect, is becoming a major player in the new grid infrastructure. RDS is expected to have low overhead, low latency, and high bandwidth while being ultra-reliable.

This presentation will focus on the advantages of using RDS over InfiniBand in order to build scalable, high-performance RAC clusters. The topics will include a general overview, configuration settings, and a benchmark evaluation of using RDS when the network is 40G versus 10G, as well as different RAC configurations.

Oracle Data Warehouse Management with Erwin

Don Soulsby, Sandhill Consultants 4:00–5:00

The purpose of this seminar is to provide the attendee with an understanding of Data Warehouse Modeling in Erwin and its extensions into Oracle's Data Warehouse Platform.

- Part 1: Define and differentiate OLTP, DW, and multi-dimensional models

- Part 2: Define dimensional model characteristics in Erwin
- Part 3: Application of concepts to Oracle Warehouse Builder
- Part 4: Erwin-assisted Oracle Cube Design

–Room 102–

And UNIX Begat ADF Faces: How Rich Is ADF Faces Rich Client

Peter Koletzke, Quovera 11:00–12:00

Back in the early days of light-client web applications, Oracle created User Interface XML (UIX), a user-interface component set used to render HTML items. Oracle used UIX in JDeveloper 9i to build E-Business Suite (EBS) applications up through EBS release 12i. However, with the advent of the Java-standard JavaServer Faces (JSF), Oracle transformed UIX components into standard JSF components and called the new component set “ADF Faces.” ADF Faces further evolved into ADF Faces Rich Client (ADF Faces RC) in JDeveloper 11g. Oracle is now using ADF Faces RC to develop Oracle Fusion Applications (the next release of EBS). Thus, ADF Faces RC is highly evolved and is now an important part of the next version of Oracle Applications. In addition, you can use it to create custom web applications.

This presentation provides a close look at the components available in ADF Faces RC. It explains how to take advantage of native ADF Faces RC features such as skins, internationalization support, and built-in Asynchronous JavaScript and XML (AJAX, also called “Partial Page Rendering”). Since an important aspect of web page design is managing screen real estate, this presentation explores how to achieve the perfect



Does Your Storage Need Liposuction?

Save **50%** with a 3PAR Storage Technology Refresh—**Guaranteed!***

There is no better time to make the move to 3PAR®. Simply purchase any 3PAR InServ® Storage Server with Thin Built In™ and use 3PAR Thin Conversion to non-disruptively “liposuction the fat” from fully allocated volumes on traditional storage—transforming them into new “thin” volumes on the InServ.

We guarantee that you'll reduce your current capacity requirements by 50% or more, or we'll make up the difference with free disk capacity and associated software and support.*

What could be simpler?

Get the 3PAR Get Thin Guarantee today by visiting www.3PAR.com/GetThin

*The above is intended to highlight certain aspects of our Get Thin Guarantee and does not contain the full terms, conditions, limitations, definitions, and other provisions (“Terms”) of the Get Thin Guarantee. The Terms shall be contained in a written Get Thin Offer which shall take precedence over the above. Qualification for our Get Thin Guarantee is subject to your acceptance of a Get Thin Offer containing the Terms and satisfaction of those Terms.



Get Agile. Get Efficient. Get 3PAR.

(continued from page 25)

layout using the ADF Faces RC layout components. The presentation also explains and demonstrates container component techniques such as modifying the standard tab order. In addition, this presentation introduces the Data Visualization Tools (DVT) components, which provide super-rich Web 2.0 functions such as interactive graphs, Gantt charts, pivot tables, and hierarchy viewers.

Developing High-Performance Database Applications with In-Memory Database Cache

Simon Law, Oracle Corporation 1:00–2:00

Oracle In-Memory Database Cache is a database option that provides real-time, updatable caching for the Oracle database. Oracle In-Memory Database Cache improves application transaction response time by caching a performance-critical subset of tables and table fragments from an Oracle database to the application tier. Cache tables are managed like regular SQL relational database tables within the TimesTen In-Memory Database. Applications continue to read/update data with JDBC, Oracle Call Interface, PL/SQL, Pro*C, and Oracle Data Provider for .NET while benefiting from accelerated performance, automatic synchronization, and high availability. This session covers Oracle TimesTen In-Memory Database functionality and performance.

Visualize Your Data on the Web—the Oracle Fusion Way

Shay Shmeltzer, Oracle Corporation. 2:30–3:30

In this session we'll show you how to build a visually appealing, dynamic web dashboard that presents data from your database. You'll see how you can build interactive graphs, geographic maps, pivot tables, gauges, and more in a visual and declarative way. Using simple SQL queries and the power of the Oracle ADF Faces data-visualization components, we'll show you how to create great-looking web interfaces and extract meaning from your data. The session is focused around an interactive development demo.

End-to-End Complete Oracle 10g Grid Control Implementation

Sakthivel Selvarajan, Ross Stores. 4:00–5:00

Grid Control offers a single-window solution that allows you to monitor and manage the complete Oracle IT infrastructure from a single console. Grid Control also offers support for monitoring certain non-Oracle products, including IBM WebSphere Application Server, Microsoft SQL Server, Juniper Networks NetScreen Firewall, and more.

With a broad set of end-to-end monitoring, administration, configuration management, provisioning, and security capabilities, Grid Control reduces the cost and complexity of managing such grid computing environments.

This presentation will cover components, architecture, and the complete steps to implement Grid Control quickly and easily in your environment. Different agent installation methods will be covered with pros and cons. The procedure to install the diagnostic kit, configure target monitoring and secure the Grid Control infrastructure will be explained. You will also learn best practices and performance tuning of the Grid Control infrastructure. ▲

President's Message

(continued from page 3)

the best on behalf of the NoCOUG board and the entire member community.

As for myself, a few words will suffice, as I have done enough speaking for a lifetime! But I do want to tell you how much I have appreciated the things that make this organization so special—not just the technical aspects but the fun and laughter we have had, including the hours that we invested together preparing for conferences and getting the *Journal* out in time. It has been an honor to serve as NoCOUG president in front of great and supportive members. Together we have shared many memories that will stay with me forever. I want to thank everybody in the organization. It has been great serving as your president

I look forward to seeing you at NoCOUG's Fall Conference, November 11 at the Oracle conference center in Redwood Shores. We will kick off the event with a keynote titled "Oracle OpenWorld 2010 Highlights" by Oracle VP, Sushil Kumar. You shouldn't miss this presentation if you didn't go to Oracle OpenWorld. After the keynote there will be twelve technical presentations in three tracks, including presentations by NoCOUG members Ahbaid Gaffoor and Dave Abercrombie. You'll find the complete agenda at <http://www.nocoug.org/next.html>.

I'll see you there. I wish us all a great conference and a great next decade. ▲

Book Review

(continued from page 10)

more complete information sources. The junior DBA would be well advised to skim this book to get an idea of what is possible, but to thoroughly study the other sources before putting these techniques into production. ▲

Dave Abercrombie has worked at Convio (with a "v," not an "f") for more than ten years, having helped to found GetActive Software before its merger with Convio. This company's business model is rather like a distributed denial of service attack against itself. Its customers are nonprofit membership organizations who want to use the web to engage and activate their members. So each day, Convio sends tens of millions of emails to these members, and then tracks the ensuing member transactions and activities, such as donations, advocacy, and click-throughs. Dave has honed his troubleshooting and scalability skills keeping these very busy databases happy. He has presented at Hotsos, and is a regular presenter at NoCOUG. He can be reached at dabercrombie@convio.com.

Copyright © 2010, Dave Abercrombie

Database Specialists: DBA Pro Service



DBA PRO BENEFITS

- *Cost-effective and flexible extension of your IT team*
- *Proactive database maintenance and quick resolution of problems by Oracle experts*
- *Increased database uptime*
- *Improved database performance*
- *Constant database monitoring with Database Rx*
- *Onsite and offsite flexibility*
- *Reliable support from a stable team of DBAs familiar with your databases*

CUSTOMIZABLE SERVICE PLANS FOR ORACLE SYSTEMS

Keeping your Oracle database systems highly available takes knowledge, skill, and experience. It also takes knowing that each environment is different. From large companies that need additional DBA support and specialized expertise to small companies that don't require a full-time onsite DBA, flexibility is the key. That's why Database Specialists offers a flexible service called DBA Pro. With DBA Pro, we work with you to configure a program that best suits your needs and helps you deal with any Oracle issues that arise. You receive cost-effective basic services for development systems and more comprehensive plans for production and mission-critical Oracle systems.

DBA Pro's mix and match service components

Access to experienced senior Oracle expertise when you need it

We work as an extension of your team to set up and manage your Oracle databases to maintain reliability, scalability, and peak performance. When you become a DBA Pro client, you are assigned a primary and secondary Database Specialists DBA. They'll become intimately familiar with your systems. When you need us, just call our toll-free number or send email for assistance from an experienced DBA during regular business hours. If you need a fuller range of coverage with guaranteed response times, you may choose our 24 x 7 option.

24 x 7 availability with guaranteed response time

For managing mission-critical systems, no service is more valuable than being able to call on a team of experts to solve a database problem quickly and efficiently. You may call in an emergency request for help at any time, knowing your call will be answered by a Database Specialists DBA within a guaranteed response time.

Daily review and recommendations for database care

A Database Specialists DBA will perform a daily review of activity and alerts on your Oracle database. This aids in a proactive approach to managing your database systems. After each review, you receive personalized recommendations, comments, and action items via email. This information is stored in the Database Rx Performance Portal for future reference.

Monthly review and report

Looking at trends and focusing on performance, availability, and stability are critical over time. Each month, a Database Specialists DBA will review activity and alerts on your Oracle database and prepare a comprehensive report for you.

Proactive maintenance

When you want Database Specialists to handle ongoing proactive maintenance, we can automatically access your database remotely and address issues directly — if the maintenance procedure is one you have pre-authorized us to perform. You can rest assured knowing your Oracle systems are in good hands.

Onsite and offsite flexibility

You may choose to have Database Specialists consultants work onsite so they can work closely with your own DBA staff, or you may bring us onsite only for specific projects. Or you may choose to save money on travel time and infrastructure setup by having work done remotely. With DBA Pro we provide the most appropriate service program for you.



CALL 1 - 8 8 8 - 6 4 8 - 0 5 0 0 TO DISCUSS A SERVICE PLAN

NoCOUG Fall Conference Schedule

November 11, 2010—Oracle Corporation, Redwood Shores, CA

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

Cost: \$50 admission fee for non-members. Members free. Includes lunch voucher.

8:00 a.m.–9:00	Registration and Continental Breakfast—Refreshments served
9:00–9:30	Welcome: Hanan Hit, NoCOUG president
9:30–10:30	Keynote: <i>Oracle OpenWorld 2010 Highlights</i> —Sushil Kumar, Vice President, Oracle Corporation
10:30–11:00	Break
11:00–12:00	Auditorium: <i>Oracle Exadata: The World's Fastest Database Machine</i> —Ron Weiss, Oracle Corporation Room 101: <i>Estimating Cardinality and Query Execution Cost: Use of Jonathan Lewis' CBO Methodology</i> —Dave Abercrombie, Convio Room 102: <i>And UNIX Begat ADF Faces: How Rich Is ADF Faces Rich Client</i> —Peter Koletzke, Quovera
12:00–1:00 p.m.	Lunch
1:00–2:00	Parallel Sessions #2 Auditorium: <i>Building Oracle Enterprise Linux Guests on a Virtual Box Windows Host</i> —Ahbaid Gaffoor, Amazon Room 101: <i>Extreme Performance Tuning</i> —Chris Lawson Editor's Pick Room 102: <i>Developing High-Performance Database Applications with In-Memory Database Cache</i> —Simon Law, Oracle Corporation
2:00–2:30	Break and Refreshments
2:30–3:30	Parallel Sessions #3 Auditorium: <i>Building a Virtualized 11gR2 Linux RAC on a Virtual Box Windows Host</i> —Ahbaid Gaffoor, Amazon Room 101: <i>Reliable Datagram Sockets and InfiniBand</i> —Hanan Hit, HIT Consulting Room 102: <i>Visualize Your Data on the Web—the Oracle Fusion Way</i> —Shay Shmeltzer, Oracle Corporation
3:30–4:00	Raffle
4:00–5:00	Parallel Sessions #4 Auditorium: <i>Avoiding SQL Performance Regressions: New Techniques for Solving an Old Problem</i> —Prabhaker Gongloor, Oracle Corporation Room 101: <i>Oracle Data Warehouse Management with Erwin</i> —Don Soulsby, Sandhill Consultants Room 102: <i>End-to-End Oracle 10g Grid Control Implementation</i> —Sakthivel Selvarajan, Ross Stores
5:00–	Networking and No-Host Happy Hour