Building a Solid Foundation

Foundation Matters

Relational theory legend Chris Date talks about where a solid foundation for your database starts.

See page 4.

Keep Your Career on Solid Ground

Read Brian Hitchcock's latest book review and learn his strategies for passing the Oracle Certified Professional 10g upgrade exam.
See page 8.

SQL Corner

A humorous tale by Iggy Fernandez recounting the adventures of his group of "fellow prisoners in Academia" and some valuable insights on record-at-a-time ("RATTY") processing.
See page 13.

Much More Inside . . .

NoCOUG's Foundation

The theme of this issue is *Building a Solid Foundation*. That means a lot of things, from getting the right education for your career, and continually learning, to designing your applications and your databases with the right foundation.

The foundation of NoCOUG rests on our history and the dedication of a group of volunteers who make things happen. We gather monthly, work together through challenges and successes, and have a good time. Without these volunteers, NoCOUG would not have the solid foundation that makes our group one of the strongest Oracle Users Groups in the country.

And our foundation has only grown stronger over the past few years. Our membership base is over 400, and we have kept improving and adding to the many benefits we offer. Think about what you might be able to suggest or do to help strengthen NoCOUG's foundation. Then, contact a board member and share your thoughts. You'll find board members' contact information on this page.

We look forward to hearing from you.

—Lisa Loper,
NoCOUG Journal Editor

Table of Contents

Publication and Submission Format

The *NoCOUG Journal* is published four times a year by the Northern California Oracle Users Group approximately two weeks prior to the quarterly regional meetings. Please send your questions, feedback, and submissions to Lisa Loper, *NoCOUG Journal* Editor, at journal@nocoug.org.

The submission deadline for the upcoming August 2005 issue is June 15, 2005. Article submissions should be made in electronic format via email if possible. Word documents are preferred.

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

My NoCOUG Moment

Darrin Swan

I live on Telegraph Hill in San Francisco and enjoy the daily benefit of a fifteen-minute walk to the office. As I made my way down Montgomery Street the other day, I had one of those thought-provoking moments that kind of put things into perspective as a fellow Oracle professional. With a red NetFlix DVD return envelope in hand, I was protected from the rain by a tent-sized Oracle umbrella (a parting gift from the Oracle/PeopleSoft launch event of which I personally noticed Ellison taking two before leaving in his silver Bentley). During my city-block by city-block walk, I made mental notes of other Oracle shops I passed by, such as Wells Fargo, Bank of America, and Charles Schwab, to name a few. As I finished my last block and entered my building, I began to wonder about the various Oracle initiatives that you, our NoCOUG membership, had planned or were currently implementing:

➤ How are you developing or implementing service-oriented architecture and web services?

➤ How are you embracing open standards?

➤ What are your business intelligence and business analytics strategies?

➤ How are you implementing and/or adhering to regulatory compliance, auditing, security, etc.?

➤ How are you mapping your business processes to your technology strategy?

➤ To RAC or not to RAC?

The list of old, new, and complex trends that you must be aware of gets bigger by the day.

With continued business consolidation, the proliferation of open standards, and many rapidly evolving and complex industry trends, the Oracle professional has a lot to keep up with. How do you best stay informed, and adapt and retool your career for these inevitable changes? When my elevator arrived at my floor, the bell dinged, and the doors opened, I had my NoCOUG moment. As Oracle professionals and members of NoCOUG, it is our duty to keep our community of Bay Area Oracle professionals informed, since we help make these industry trends happen.

This brings me to a very important topic that I hope drives and helps us all to stay up-to-date and well informed—our next NoCOUG conference. Our spring conference will take place on May 19, 2005, and will be hosted by Lockheed Martin in Sunnyvale, California. We have a full day of Oracle education planned, with presentations ranging from Cary Millsap of Hotsos and his keynote address, "Why 'System' Is a Four-Letter Word," to Noel Yuhanna, a leading industry analyst from Forrester, with "Future of DBMS and DBA." We also have presentations from other Oracle experts, such as Peter Koletzke of Quovera, Inc., and Prabhaker Gongloor of Oracle Corporation, and presentations with real-world examples from Andy Rivenes of Lawrence Livermore National Laboratory and Virag Saksena of Auptyma. For a detailed summary of our May conference, please see page 26.

During our spring conference, or even in between conferences, please let us know what current trends you have questions about or share your successes with the rest of the users group in the form of your own presentation. Your success is our success.
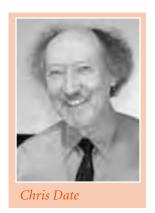
We look forward to seeing you in May. ▲

Relational Database Theory for a Solid Foundation

An Interview with Chris Date

If you studied relational database theory in college, the names E. F. "Ted" Codd and C. J. "Chris" Date should be familiar to you. They are the founding fathers of relational database theory. Dr. E. F. Codd developed the theory of relational database design while working at IBM in the 1970s. Chris Date became acquainted with Dr. Codd after reading one of his papers on this theory, and together they formed a team that introduced these ideas and principles to a larger audience, which led to its acceptance in the business world.

Chris Date

In Date's college text *An Introduction to Database Systems* (6th ed., Addison-Wesley, 1995), he states, "The foundation of modern database technology is without question the relational model; it is the foundation that makes the field a science." To take this thought a little further, I'd like to suggest that building your database application on a solid foundation would require an understanding of relational database theory. And there are others that share this opinion. Greg Jorgensen states in his paper "Introduction to Relational Database Systems," "You must understand relational theory to correctly design a database—just learning a particular RDBMS won't get you all the way there."

During my interview with Date, he talked about the changing attitude toward database theory. He recognized that the audience is changing, as many colleges are promoting classes in specific vendor implementations of the relational model instead of the basic theory. Since this can leave a gap in the essential knowledge of the basics, Date has written a new book, *Database in Depth: Relational Model for Practitioners*, which is scheduled to be released by O'Reilly in May 2005. Unlike many of the academic works he's published, this book is aimed at an audience of professionals already working in the field.

Interviewing Date for this article was a real privilege. I think you will enjoy reading more about relational theory from his perspective. Date grew up in the UK and studied mathematics at Cambridge University. He joined IBM (UK) in the late 1960s and eventually made his way to California to work more closely with Codd. Today, Date lives in Healdsburg, California, spending his time writing and lecturing.

How did you get started in the relational database field?

Date: Well, like so many things, I was in the right place at the right time. I was working at IBM in England, where I had joined as an instructor. I had been an instructor for a few years, but they had a very enlightened policy which dictated that you couldn't spend all your time teaching. You had to rotate out and get into the trenches. So, I rotated out and joined a little research group, where I was given the job of looking for a solution for a database project. During my research, I stumbled upon Ted Codd's paper. I had been reading a lot of material and experimenting with different products. And since I had a mathematical background when I read Ted Codd's paper, it seemed obvious to me his principles were based on solid thinking and logic. Looking back, if I'd realized how long it would have taken to get the world to accept that point of view, I don't know if I would have continued, but it seemed obvious to me at the time. So, I began corresponding with Ted, met him a little bit later, and one thing led to another.

Why was it so difficult? What was the biggest challenge you found in trying to introduce the relational model?

Date: Well, IBM was making a lot of money off other database technologies at the time, and they saw these new ideas as potentially threatening those other products and revenue. In fact, Ted and I developed a very deliberate policy. Instead of trying to encourage or befriend people within IBM to build a product, we went and talked to the customers. We said, "Look, this is how it could be: you could have a product that does x, y, and z." Then, these customers turned things around by putting pressure on IBM. That's exactly what we wanted them to do, and that's exactly what happened. But it took ten years, and I had no idea it was going to take ten years.

What was your relationship with Ted Codd, and what influence did he have over you?

Date: Well, his influence was tremendous. I generally like to say Ted was a genius. There's no question about that; and he came up with these wonderful ideas. But like many geniuses, he wasn't too good at communicating with ordinary mortals. So, my job was to be the ordinary mortal. I would take his ideas and explain them to the rest of the universe. And, for many years, Ted was the inventor and I was the explainer. Later on it became less like that; we went into business together in the 1980s and had a joint company. About that time I started doing some of my own research, so the balance shifted then. But for many, many years Ted was the creator, and I didn't contribute much in those days—I just went around putting it together in technical terms for other people to understand it.

What about your background provided you with the ability to explain these ideas to others?

Date: I'm a slow learner—really, it's true. Therefore I can appreciate the problems people have when they are trying to learn something. To be a slow learner helps you be a good teacher. If you are too quick at learning, you can't understand the difficulties most people experience trying to learn something new. Well, that's my view, anyway.

Since relational database theory is based on mathematical principles, is mathematical training or education important for understanding the theory?

Date: Well, before I answer the question let me just say: this is one of the reasons I've stuck with the database field after discovering it. I'd been in the computer field for about eight years at the time, and I'd done programming and so on but hadn't really used any of the mathematics I'd learned at college. To discover there was an application for some of my college training was wonderful. But back to the question: I would say yes and no. What I really think you need is the ability to think clearly and apply logic. Mathematical training will help you acquire those skills, but I don't think it's the only way. I have colleagues with liberal arts backgrounds that are terrific programmers and database practitioners.

What business value is there in your concern that the relational model isn't being followed properly in currently available commercial products?

Date: This is an example of the kind of question I always want to respond to by turning it on its head. Database management is something where—unlike some parts of computer science—there is some solid theory. We know the value of the theory; we know the benefits that accrue if we follow the theory. We know there are costs associated with not following the theory. We don't know what all the costs are; at least it's hard to quantify them.

For example, if you are on an airplane, you want to know the airplane is constructed according to the principles of physics. If you are in a high-rise building, you want to know it's been built according to architectural principles. If I'm using a database, I want to know it's been built according to database principles. If it hasn't, I know there is the potential for things to go wrong. It's hard to say exactly what—and it's hard to say if these problems are going to be minor or major —but we know there are going to be things that are wrong.

So, I think it's incumbent for people to not say, "Tell me the business value of implementing the relational model." I want them to tell me the business value of not doing it. A quote I like by Leonardo da Vinci sums it up rather well: "Those who are enamored of practice without theory are like a pilot who goes into a ship without rudder or compass and never has any certainty where he's going. Practice should always be based upon a sound knowledge of theory." Those who ask questions like "What's the business value of the relational model?" are basically saying, "What's the value of theory"? And I want them to tell me the value of not abiding to theory.

There are parts of the theory that are in debate, namely the use of nulls in the database. Why do you consider nulls to be a disaster, and what is the best way to handle missing information without the use of nulls?

Date: Well, for me it's a question of science. There are solid scientific reasons for objecting to the idea of nulls. And they boil down to the following: you can find queries where, according to the logic of nulls, the result is correct. But this is not the correct answer in the real world. What that means is, if you build a system that implements nulls 100% correctly—and by the way, no one does—but even if you build a system that implemented nulls correctly you couldn't use the system and you couldn't trust it because you never know when the answer you get is correct or not. And further, you have no way of knowing which queries are giving you correct answers and which are not. Everything becomes suspect, all bets are off, and you can't trust the system at all. I think that's a showstopper, and I'd like to turn the question around, since I can show you easily that if you use nulls you can get catastrophic failure. So if you want to use nulls, it's incumbent on you to tell me why you are not getting catastrophic failures and never will. *(Editor's note: For more specific examples of the problems with nulls, see "Further Reading" at the end of the article.)*

Date *(continued)***:** Now the other part of the question is the hard one. Nulls are supposed to address the problem of missing information. We all know that we have missing information; that's a fact. Nulls in fact are a disastrous and bad attempt to solve the problem. So if you decide not to use nulls, what do you do? Unfortunately, you have a problem here because almost all of the products have been built and designed on the assumption that you will use nulls. As a trivial example, perhaps you have a column in a table that is supposed to be Joe's salary, but you don't know Joe's salary. You decide to put a value such as -1 in that field with the understanding that -1 indicates you don't know the salary. Well, that might work for salary, but if the column is something like a date (datatype), it turns out that every possible date you can represent represents a real date, so there's no real value you can use to indicate you don't know the proper value. There is a whole new implementation technology that's coming down the road that effectively means you can build systems and design your databases right, which means you can design away the need for nulls, and you don't pay a performance penalty. So I realize I'm living in a bit of an

ideal world here, because the solution I'm talking about is not something that's currently commercially available.

What is this new implementation technology?

Date: There's a technology called the TransRelational Model. I can't talk about it very much because I'm under a nondisclosure. But I believe this technology has incredible promise. Although I can't talk much about this technology, there is some information that is publicly available by searching the Internet. Going back to the example of Joe's salary, which for employee Joe you don't know the salary: well, if you don't know the salary, don't put anything in the database. It's absurd to put something in the database that says you don't know something. The database is supposed to contain what you do know. The correct way to represent missing information is to have information missing.

How do you represent the missing salary information in the database?

Date: It's related to normalizing. You have one table with employee numbers and salaries for employees with known salaries, and you have another table with just employee numbers for employees with unknown salaries. So just like with normalizing, you break your tables into lots of little tables. The problem with that approach in today's products is it leads to joins, and joins are not free. But in the technology I'm talking about, joins are effectively free.

What do you enjoy the most and the least about the work that you do?

Date: I really enjoy developing my own ideas and formulating them in a way so I can communicate them understandably to others. That's a real intellectual challenge. You may think you understand something, but until you try to explain it—especially try to explain it on paper— you don't understand it. And I'm afraid it's embarrassing the number of times I've thought I understood something until I tried to write about it. It's hard work, but I really enjoy it.

As far as what I enjoy least, well, there's always SQL. Having to work with it and struggle through its problems. The biggest problem with SQL is that it doesn't implement the relational model; it violates the principles in so many ways. I think you should learn the relational model and then you can learn SQL. (*Editor's note: For specifics on how SQL violates the principles of the relational model, see "Further Reading" at the end of the article.*)

What are you most proud of in your career?

Date: Perhaps that's been keeping my sanity! But I think I was instrumental in getting the relational model accepted. And I'm proud of my first book, and that I do get good reviews—although I also get bad ones.

What activities do you enjoy in your spare time?

Date: I'm a voracious reader. My favorite nontechnical genre is science fiction. My wife and I also enjoy hiking, especially in the desert. Lots of interesting wildlife in the desert. We try to get out there at least once a year. *(Editor's note: A special thanks to Brian Hitchcock and Les Kopari for their contribution of technical questions.)* ▲

Interview conducted by Laurie Robbins, assistant journal editor.

Further Reading

To learn more about some of Date's recent writings and educational seminar schedules, visit **http://www.thethirdmanifesto.com**

E. F. Codd's initial paper on relational database theory: **http://www.acm.org/classics/nov95/toc.html**

Greg Jorgensen's article "Introduction to Relational Database Systems": **http://www.pdxperts.com/article-rdbms-intro.php**

Problems with the use of nulls, missing information, and other SQL flaws—*An Introduction to Database Systems* (6th ed., Addison-Wesley, 1995), Chapter 20: **http://c2.com/cgi/wiki?SqlFlaws** and **http://www.dbdebunk.com/page/page/1858002.htm**

Tutorial D, a relational language that, unlike SQL, is truly relational: **http://c2.com/cgi/wiki?TutorialDee**

TECH TIPS

NoCOUG Conference Recap and Resources

I f you missed February's Northern California Oracle Users Group (NoCOUG) meeting, you can still benefit from all the knowledge shared at the full-day conference. Many of the speakers have provided their presentations for viewing and download. Here's a short list of what's available on NoCOUG's website, at **www.nocoug.org**:

➤ "Speeding Up Queries with Semi-Joins and Anti-Joins: How Oracle Evaluates EXISTS, NOT EXISTS, IN, and NOT IN" by Roger Schrag

➤ "Oracle 10g Backup and Recovery New Features" by Daniel Liu

➤ "A Structured Approach to Database Administration Using the Principles of ITSM and ITIL" by Iggy Fernandez

➤ "Essentials of Real Application Clusters" by David Austin

➤ "Minimizing Risks Through Deployment Standardization" by Sudip Datta

➤ "Oracle 10g for Data Warehousing' by Hermann Baer ▲

Strategies for Passing the OCP 10g Upgrade Exam

Review by Brian Hitchcock

OCP Oracle Database 10g: New Features for Administrators Exam Guide

By Sam R. Alapati
Published by Oracle Press, 2004

Summary

Overall review: *Excellent.*

Target audience: *Anyone interested in passing the Oracle 10g upgrade exam.*

Would you recommend to others? *Yes, absolutely.*

Who will get the most from this book? *Anyone who is preparing to take the 10g upgrade exam in the near future.*

Is this book platform specific? *No. The author notes that he prepared the examples in the book using both Unix and Windows systems.*

Why did I obtain this book? *I was studying to take the 10g upgrade exam at Oracle OpenWorld 2004 in San Francisco.*

Overall Review

Let's get to the good part. I passed my exam on the first try, and, therefore, this study guide did its intended job. Based on this alone, I recommend this book. I had virtually no previous formal training with the Oracle 10g database, yet I was able to pass the upgrade exam by reading the chapters of this book and using the review sections and the sample questions at the end of each chapter. This exam guide provided complete and thorough coverage of the material needed to prepare for the exam. The book covers the material without being any longer than necessary. I could not have prepared for this exam so quickly without this guide, and I don't believe I could have passed the exam without using a book like this.

What Does OCP Mean?

OCP stands for Oracle Certified Professional. Oracle provides training classes and sets up the certification exams for many different topics. This exam guide covers DBA certification, specifically the certification of existing OCP DBAs who are moving up to the 10g DBA certification. In order to take this exam, you need to have your Oracle 9i OCP designation. The exam guide also assumes that you've familiarized yourself with all of the 9i DBA material before you begin studying for the 10g upgrade exam. You need to realize that—even though you are studying for the 10g certification upgrade exam—the exam can ask about material that is unique to Oracle 9i or even 8i. I originally got my OCP certification back on Oracle 8. Since then, I've taken the upgrade exams for 8i, 9i, and now 10g.

In my experience, only one or two of the 60 exam questions specifically related to older material. Therefore, this exam guide does not completely cover all of the material that could possibly be in one of the upgrade exam questions. However, this guide does cover all of the material that will be in the exam that relates to the 10g database.

Overall Comments on Certification in General

I find most DBAs I know have very strong feelings about whether or not certification is worthwhile. Therefore, I think it is important that I explain my views on certification before I get into my review of the certification exam guide to give it proper context.

Understand that the entire certification process is a marketing tool for Oracle. When I was studying for my 10g upgrade exam, I was learning the many new features of the latest Oracle RDBMS product. Even though I was not supporting any 10g databases at the time, by studying for my exam, I was becoming aware of the new features of 10g. This got me thinking about why and how I would implement 10g in my work environment. This way, I was already preparing to use 10g at work, which is to the benefit of Oracle Corporation.

Critics of the certification process will be quick to point out that someone can pass the certification exams without necessarily having any practical experience with any Oracle products. My experience with the 10g upgrade exam will only confirm their criticisms. Other than my preparation for the exam, I do not have any work-related experience with Oracle 10g. And I passed the exam.

While both of the previous criticisms of the certification process are valid, I think there are valid benefits of certification. My organization needs to continually look forward for new products and processes. Certification helps me and the company I work for to be fully aware of

the new features of our vendors' products. The more we know in detail what things are coming next, the better we can prepare for them. If I were to wait until I had experience with the new features of an Oracle product before I tried to learn about those new features, I probably would never implement any of them. I think the certification process provides an organized, systematic way for me to continually be up-to-date on the new features that are coming in the latest Oracle products. I realize that this does make me part of the Oracle marketing machine, but I believe the benefits to me and my organization are worthwhile.

Finally, I personally encourage you to become certified. If you are already certified at the 9i level, and you feel you must be using Oracle 10g before you can try to pass this upgrade certification exam, I want to encourage you to go ahead and get certified now. In the beginning of Chapter 1, the author states that it is essential to practice using the new features of Oracle 10g in order to pass the test. I disagree. In studying for my exam, I did not have any practice using any of the new features and I passed. While I realize this will encourage critics of the certification process, you should still go ahead and get yourself certified just as soon as you can. It's good for you to have it on your résumé, and it's good for you to get it out of the way as soon as possible. Do not use your lack of experience with the latest Oracle product as a reason to delay getting certified.

The Book Itself

The quality of the book is very high. The list of chapters is shown below, and from what I remember of the actual exam, this covers the exam topics very well. Understand that anything I say comparing the material in the book and in the practice questions to the questions in the actual exam is based solely on memory. Since you are not allowed to make any notes that you can take with you from the exam, I have no way to clinically assess how well the book covers the exam topics.

Any publication will, of course, have some errors. For a normal book, this is less of an issue than for an exam guide. Since you are using the exam study guide to prepare for a test that you are paying for, you'd really like the study guide to be absolutely perfect. I believe this study guide is as close to that ideal as possible. I found very few errors in the text of the book, in the sample problems at the end of each chapter, and in the sample questions on the CD that came with the book.

The flow of the book is very good. The topics in each chapter seem to move easily from subject to subject, and the writing style is approachable. The way the book is written makes it easy to assimilate the relevant information. Those who are critics of the certification process will feel that this is a criticism of this book: this study guide is not the ideal resource for learning all of the possible implications of Oracle 10g (e.g., architecture). What I am saying, though, is that it is a very good way to learn the information you need to pass the OCP upgrade exam.

The chapters are:
1. Installation, Server Configuration, and Database Upgrades
2. Loading and Unloading Data
3. Automatic Database Management
4. Manageability Infrastructure
5. Application Tuning
6. Space and Storage Management Enhancements

TECH TIPS

Graceful Failover and Failback Procedures in Non–Data Guard Environments

By Brian Keating

Switching a standby database into the primary role without losing data is called a "graceful switchover" in the Data Guard documentation. When a graceful switchover is performed properly, the standby and primary databases can swap roles without losing data and with no need to rebuild the standby database from scratch. Using graceful switchovers, you can switch your standby database into the primary role so that your regular primary server can undergo maintenance, and then you can switch back when the maintenance is complete.

The good news is that you do not need to use Data Guard in order to perform graceful switchovers. I have written a detailed paper that describes the procedures to perform graceful switchovers and switchbacks of hot standby databases in Oracle environments that are not using Data Guard. I use the term "graceful" to mean failovers that do not require databases to be opened with the RESETLOGS option—and as a result, graceful switchbacks do not require the primary database to be rebuilt. (That is, they do not require the standby database's datafiles to be copied to the primary server.)

You can read my paper at http://www.dbspecialists.com/presentations.html#graceful_failover. ▲

7. The Oracle Scheduler and the Database Resource Manager
8. Backup and Recovery Enhancements
9. Flashback Technology Enhancements
10. Automatic Storage Management
11. Enhancements in Analytical SQL and Materialized Views
12. Miscellaneous New Features

Each chapter consists of the material, a two-minute drill, and a self-test of sample OCP test questions. There are also lab questions at the end of each chapter, but I didn't use them. These questions cover tasks that require actually using the Oracle 10g RDBMS, and as we all know, the OCP exam doesn't test your ability to actually use the database. The answers to the self-test questions are provided at the very end of each chapter.

The CD That Comes with the Book

The CD that comes with the book contains two complete sample tests. These sample tests contain 60 questions each, just like the real OCP exam. These sample tests are very similar to what the actual OCP exam will be like. On the other hand, based on my memory, the sample tests on the CD didn't cover nearly the breadth of topics that are on the actual exam.

Understand that the CD with its practice exams is really a teaser. The company that put the practice exams on the CD hopes to get you to purchase more practice exams. Therefore, I'm not really surprised that the sample exams that you get for free on the CD don't cover all of the top-

I want to be very clear that based on my experience, I firmly believe I would've done just as well on the OCP upgrade exam whether I had the day-long class or not.

ics on the exam. I didn't purchase any additional practice exams, and I did well on the real exam. So, I don't think you need to purchase any additional practice exams for yourself.

What Could Have Been Better

Since I passed the exam on the first try, it isn't obvious what needs to be improved in this study guide. I think my one suggestion for improvement, which probably isn't practical, would be to include more practice questions. Having studied for, taken, and passed multiple OCP DBA certification exams over the years, I believe that the practice questions are the absolute best way to prepare for these exams. So, while it isn't a criticism of this study guide per se, I think any way that you can get exposure to more practice questions is the best way you can become even better prepared for any of the OCP exams. If you are willing to spend the money for the additional practice

exams from the company that put together the CD that comes with the book, I think that would be my only suggestion for further improving your odds of passing the exam after you have used this study guide.

Notice that I'm not suggesting that you install and work with the Oracle 10g database as a way to prepare for the exam. This may seem odd to you. If you have a very long lead time, say six months, before you plan to take the exam, then I would agree that actually installing and using the various new features of Oracle 10g would probably help you with the exam. However, I firmly believe that while this would be worthwhile, it would only slightly increase your ability to do well on the certification exam compared to working with more practice exam questions.

Exactly What I Did to Prepare for the Exam

I was not planning to take the certification exam until I received an email from Oracle telling me that they were offering a free OCP 10g upgrade exam during Oracle OpenWorld 2004 in San Francisco. That forced me to make a decision. I decided that I was going to try to pass the exam while it was free. This all happened about six weeks before the conference, which meant I had very little time to get up to speed to pass this exam.

I had also been waiting to see if and when the official upgrade OCP study guide would be published. As it turns out, the study guide was published just in time. I ordered the study guide from Amazon and it arrived within a few weeks of Oracle OpenWorld. Oracle also sent me an email telling me about an exam cram session that was going to be held the Sunday of Oracle OpenWorld. This session was free for Oracle OpenWorld attendees. So my plan was to go through the study guide, attend this cram session at Oracle OpenWorld, and take the OCP upgrade exam the following day. The material presented during this all-day cram session was the same material they normally present in their week-long class to prepare people to take the Oracle 10g OCP upgrade exam. I'm confident that the book was much more helpful than the class.

Don't get me wrong: the class was fine, and it provided good reinforcement for what I'd gotten from the study guide. However, I want to be very clear that based on my experience, I firmly believe I would've done just as well on the OCP upgrade exam whether I had the day-long class or not. The class itself, again, while being OK, went over material at a pretty high level. It didn't go into the level of detail that I think is needed to be well prepared for the kinds of questions that come up on an OCP exam.

From my experience, the upgrade exams are actually more challenging than the regular OCP DBA exams. The reason is that on an upgrade exam, you can be asked anything about the latest version of Oracle and all previous versions of Oracle. For example, during the 10g exam, there were one or two questions that were specific to Oracle 9i. I find it to be more challenging than what I remember of studying for the Oracle 8 OCP exams, where a single exam would cover, for example, backup and

recovery. On the upgrade exam, the single exam can cover backup and recovery, networking, utilities, installation, and just about anything else you can think of with respect to Oracle DBA topics.

How I Used the Book to Prepare for the Exam

I went through the book chapter by chapter, from start to finish. I read every page, read the review section at the end of each chapter, and then answered the sample test questions to see how I was doing. It quickly became obvious that I needed to study a whole lot more. Based on how I did on the actual exam, I think this was a very effective way to go.

Going chapter by chapter, answering the questions as you go, gives you a very precise measurement of which topics you have mastered, and which topics you haven't. I also found that by using the study guide, I was quickly able to greatly increase my ability to answer the test questions. For example, the first time I would read through a chapter and answer the sample questions, I might only get 40% or so correct. Then I would review the chapter again the next day and go over the sample questions again, scoring significantly better.

Now obviously, since you see the same questions over and over, your score naturally goes up. If you don't write the answers to the questions in the book, and use a separate piece of paper for your answers, you'd be surprised how effective it is to look at the same questions again the next day. The best preparation for these tests is lots and

lots of practice answering OCP test questions.

This also highlights why I think you're much better off preparing for the exam using this study guide than, for example, a book that covers simply the new features of Oracle 10g. A book covering the new features simply doesn't have the focus that you need to pass an OCP exam.

The OCP exam questions are very particular—some would even say insane—in the way they probe for detail and at times ask questions that I don't believe have a single correct answer. Passing an OCP exam is a game. You need to learn the rules of the game to be successful. You don't have to agree with the way the certification exams are written, but you do have to be good at figuring out the correct answer if you want to pass the exam.

After going through all the sample exam questions at

the end of each chapter as well as the two full-length practice exams on the CD, you will have seen over 300 questions. Again, lots of practice answering OCP exam questions is the best way to prepare for an OCP exam.

After I had completely gone through the book and answered all the practice exam questions, I took the first of the two practice exams on the CD to see how I was doing. My score was acceptable but not great, so then I reviewed the material on which I knew I was weak. Again, from the practice exams at the end of each chapter and the first practice exam on the CD, I was very aware of which areas I needed to go over again. Finally, I took the second practice exam on the CD and my score was a lot better. At this point, I was feeling fully prepared. Actually, I think what I was feeling was burned out on studying, and I decided that I was about as prepared as I was going to get, and further effort probably wasn't going to improve my results.

Conclusion

Using the study guide, I was able to go from virtually no knowledge of Oracle 10g DBA features to passing the exam by a comfortable margin in about two weeks. Note that I did this at a total cost of just a study guide versus the cost of the full-week Oracle class. This means the book is an excellent value. Even if you paid the full retail price for the OCP exam, which I believe is around $150, and around $50 for the book, you could become Oracle 10g DBA OCP certified for $200 and roughly two weeks of evenings. This is not a bad deal compared to a week-long Oracle class, which can easily cost $2,000 and requires you to be away from work for a week. For those who want to become Oracle 10g DBA OCP certified, this study guide is excellent. It works, and if you commit yourself to the book, and answer the sample questions in the book and on the CD, I believe you will pass the exam by a comfortable margin. ▲

About the Author

Sam R. Alapati is an Oracle Database Administrator Certified Professional (OCP) who manages Oracle databases for the Boy Scouts of America at their National Office in Irving, Texas. Prior to this, Sam worked at Sabre, Lehman Brothers, AT&T, and the ABC Corporation in New York City. He also worked in the New York Practice of Oracle Corporation as a Senior Principal Consultant. In 2003, he published the book Expert Oracle 9i Database Administration.

About the Reviewer

Brian Hitchcock has worked at Sun Microsystems in Newark, California, for the past nine years. Before that he worked at Sybase in Emeryville, California. Even further in the past, he spent twelve years at Lockheed in Sunnyvale, California, designing microwave antennas for spacecraft. He supports development databases for many different applications at Sun, handling installation, tuning, NLS, and character set issues as well as Unicode conversions. Other interests include Formula One racing, finishing a Tiffany Wisteria lamp, Springbok puzzles, Marklin model trains, Corel Painter 8, and watching TV (TiVo® rules!). Brian can be reached at brian.hitchcock@aol.com *or* brhora@aol.com.

TECH TIPS

Oracle Backup and Recovery Using RMAN

I f you are looking for an introductory article on using RMAN for Oracle backup and recovery, check out the resources on **www.orafusion.com**. Kailash Awati and Arati Apte have started a three-part series on the subject: Part one explains how to script RMAN backups, and part two discusses some recovery scenarios in detail. The focus is on Windows servers, but much of the information applies to other platforms as well. In addition, you'll find the following and more in the "Articles" section of the website, at **http://www.orafusion.com/resources.htm**:

➤ "Oracle Fine Grained Access Control": Implementing row-level security in Oracle.

➤ "Oracle SQL Aggregation Extensions": An introduction to "cube" and "rollup" extensions of the "group by" clause.

➤ "Oracle Generic Heterogeneous Services": A quickstart guide on connecting to non-Oracle databases from within Oracle.

➤ "Java Stored Functions in Oracle": Write, compile, and invoke a Java stored module in Oracle. ▲

RATTY SQL
– Part I –

By Iggy Fernandez

And ere three shrill notes the pipe uttered,
You heard as if an army muttered;
And the muttering grew to a grumbling;
And the grumbling grew to a mighty rumbling;
And out of the houses the rats came tumbling.

—"The Pied Piper of Hamelin"
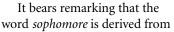by Robert Browning

Summary

RATTY processing is "record-at-a-time" processing using "cursors" and procedural loops. It is very intuitive but is several orders of magnitude slower than if the data were processed using SQL's powerful set processing capabilities. In this two-part article, we test the speed of the various approaches and discuss new features of SQL that make light of the most challenging problems.

Youthful Folly

The year was 1987. *Star Trek—The Next Generation* premiered on television with the episode "Encounter at Farpoint." In the dark bowels of the National Center for Software Technology in a remote suburb of Bombay, I and my fellow prisoners in Academia, Dinesh Shenoy, Nakul Mahimker, and Ambuj Jain, toiled tirelessly on our relational database project. We were developing a student information system and had created hundreds of tables and provided an admirable quantity of reports and data capture methods. Our "important innovation," as we proudly described it, was that we had created an "Application Programming Interface" using the C language, to make it easy for future generations of software developers, such as might wish to extend our system, to query and update the data. For each table in our system, we had provided a retrieval procedure that accepted the "primary key" as query argument and retrieved the data from the corresponding data record. We also provided procedures for inserting, updating, and deleting data, all based on the primary key.

Needless to say, we were suitably proud of our "important innovation" and expected to receive an excellent grade. But our enormous egos were crushed like Humpty Dumpty, and could never be put together again, when the professor handed out a shameful "B" grade, tastefully garnished with choice criticisms. He tried to explain to us, though it would not fully sink in until many years later, that, in our sophomoric zeal, we had managed to turn a mighty relational database management system into a puny little weakling by locking away its powerful weaponry (SQL) and substituting RATTY functions in their place.

It bears remarking that the word *sophomore* is derived from

Iggy Fernandez

two Greek words: *sophos*, which means wise, and *moros*, which means foolish. The professor was calling us "sophisticated morons."

RATTY SQL Defined

Of course, the kindly old professor was right, as professors often are. Structured Query Language, or SQL, as it is called by the cognoscenti, is the mighty weapon of a relational database management system such as our beloved Oracle. Whole legions of data records can be dispatched using a single powerful SQL statement, such as the one below, which gives all employees a 10% pay raise. I would consider using it myself, if I did not fear the strong arm of the law.

```
update employee set salary=1.1*salary;
```

RATTY processing, on the other hand, is the archenemy of database performance. Each record is individually retrieved, the appropriate modifications are computed, and then the modified values are stored in the database.

The Archenemy of Performance

Of course, we need to explain why RATTY processing is antithetical to performance, but the reasoning is simple. The performance penalty is a direct result of the huge cost attached to sending frequent messages between the client program and the RDBMS.

It is, of course, possible to mitigate the performance penalty by encapsulating the RATTY statements into "stored procedures"; however, even so, set processing will always be orders of magnitude faster, and the discerning reader is invited to mull the reasons for himself.

Scalable Pizzas

We can go one step further and claim that RATTY processing is inherently "unscalable," but first we need to define the term. We use an analogy from the "real world," the favorite haunting place of Oracle writers. Consider, then, the example of a pizza oven. It takes a lot of coal to heat a pizza oven to the correct temperature for baking pizzas, but it takes very little additional coal to maintain the temperature. It costs a small fortune in coal, therefore, to bake the first pizza, but it only costs pennies to bake every pizza after that. We say that there are "economies of scale."

A procedure is "scalable" when there are "economies of scale," and "unscalable" when there are none. It is unscalable to have to start with a cold oven every time we want to make a pizza, which is the case when we attempt to do so at home. It would be more scalable, and hence cheaper, to bake a pizza in a communal oven, such as those that probably existed before the great American dream of home ownership took root and tract homes spread for miles in every direction.

Note that is possible for a procedure to be extremely efficient, but not scalable; e.g., no unit-cost savings are derived when large volumes are processed. Also, it is possible for one procedure to be more costly than another for small volumes, yet more "scalable" for large volumes. For example, a computerized accounting system would be more expensive than manual bookkeeping methods if the business only had a few transactions to process, but would be more "scalable" if the business had millions of transactions to process.

The Proof of the Pudding

A claim is only as good as it can be proven in practice, and we prove our point with a series of tests. We create a large table and find out how long it takes to update a larger and larger number of records using three different methods:

1. A single powerful SQL statement
2. RATTY commands encapsulated in a stored procedure
3. RATTY commands embedded in a Pro*C program

Test Data

We created a large set of test data using the following statement.

```
create table testdata as
select rownum as id,object_name as name,0 as value
from dba_objects
order by object_name;
```

Here is the first program. It submits a single SQL statement to the database. Note that, for consistency with the other programs, we use Pro*C as the delivery medium.

```
#include <stdio.h>
#include <sqlca.h>

main(argc,argv)
int argc;
char *argv[];
{
  EXEC SQL begin declare section;
     int max_id = atoi(argv[1]);
     char *connect_string = "scott/tiger";
  EXEC SQL end declare section;

  EXEC SQL connect :connect_string;
  EXEC SQL update testdata
          set value = value + 1
          where id <= :max_id;
  EXEC SQL commit;
}
```

Here is the second program. It invokes a PL/SQL procedure called update_testdata stored in the database. Once again Pro*C is used as the delivery mechanism.

```
#include <stdio.h>
#include <sqlca.h>

main(argc,argv)
int argc;
char *argv[];
{
  EXEC SQL begin declare section;
      int max_id = atoi(argv[1]);
      char*connect_string = "scott/tiger";
  EXEC SQL end declare section;

  EXEC SQL connect :connect_string;
  EXEC SQL call update_testdata(:max_id);
  EXEC SQL commit;
}
```

Here is the PL/SQL procedure used by the above program.

```
create or replace
procedure update_testdata ( p_id in number )
as
    l_value integer;
    cursor c is
       select value from testdata
       where id <= p_id
       for update of value;
begin
    open c;
    fetch c into l_value;
    while ( c%found )
```

```
    loop
       l_value := l_value + 1;
       update testdata
       set value = l_value
       where current of c;
       fetch c into l_value;
    end loop;
    close c;
end;
```

Finally, here is the third program.

```
#include <stdio.h>
#include <sqlca.h>

main(argc,argv)
int argc;
char *argv[];
{
 EXEC SQL begin declare section;
       int p_id = atoi(argv[1]);
       char *connect_string = "scott/tiger";
       int l_value;
 EXEC SQL end declare section;

 EXEC SQL connect :connect_string;

 EXEC SQL declare c cursor for
        select value from testdata
        where id <= :p_id
        for update of value;
 EXEC SQL open c;
 EXEC SQL fetch c into :l_value;
```
(code continues in next column)

```
 while(sqlca.sqlcode == 0)
 {
       l_value = l_value + 1;
       EXEC SQL update testdata
             set value = :l_value
             where current of c;
       EXEC SQL fetch c into :l_value;
 }

 EXEC SQL close c;
 EXEC SQL commit;
}
```

To prevent "data buffering" from influencing the results, we flushed the Shared SQL Area and the Buffer Cache after every run using the following commands.

```
alter system flush shared_pool;
alter system flush buffer_cache;
```

The Results

Here, then, are the results of updating the data using each of the three methods. For the largest data set, a set operation is fifteen times faster than the stored procedure and eighty times faster than Pro*C.

ELAPSED TIME PER THOUSAND RECORDS			
Records	Set Operations	PL/SQL Cursors	PRO*C Cursors
1,000	0.124	0.414	1.457
2,000	0.061	0.174	1.234
3,000	0.038	0.148	1.198
4,000	0.034	0.130	1.171
5,000	0.026	0.121	1.170
10,000	0.015	0.106	1.169
20,000	0.010	0.102	1.150
30,000	0.007	0.108	1.151
40,000	0.006	0.094	1.150

Pretty Pictures

That a picture is worth a thousand words is proved by the following graph showing the "returns of scale," i.e., the improvement in unit cost as we process larger and larger sets of data. We leave it as an exercise to determine which particular attributes of this example make set operations so much more scalable than with Pro*C cursors. In other words, not only do they clearly have the speed advantage, but also the improvements in unit cost when we process large data sets are simply astonishing.
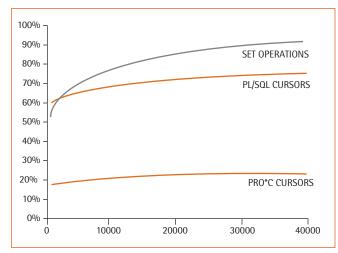

Figure 1: Economies of Scale

In Defense of SQL

We now address the complaint that the SQL language is not powerful enough to handle very complex tasks. This may have been true in the early days of Oracle but not today, for the following reasons.

1. Enhancements: SQL has been extended both by ANSI committees and by Oracle over the years. It's not my father's SQL any more. We will list a number of these enhancements in Part II of this article, including the "outer join" and "windowing" functions.

2. Orthogonality: An independent SQL statement can be embedded within another SQL statement anywhere a table name can appear or a variable name can appear. These SQL statements are often referred to as "inline views."

3. Extensibility: Oracle has provided a number of ways to extend the SQL language, including user-defined functions, user-defined aggregate functions, and even user-defined operators.

We will illustrate some of the above features in Part II of this article by solving the following problem. First create the following two tables and populate them with sample data.

```
create table sale (
  book_id integer not null,
  sale_id integer not null,
  sale_price number not null,
  constraint pk_sale primary key (book_id, sale_id)
);

create table new_sale (
  book_id integer not null,
  sale_price number not null
);

insert into sale values (1,1,20);
insert into sale values (1,2,19);
insert into sale values (2,1,10);
insert into sale values (2,2,9);
insert into sale values (2,3,8);

insert into new_sale values (1,18);
insert into new_sale values (1,17);
insert into new_sale values (1,16);
insert into new_sale values (2,7);
insert into new_sale values (2,6);
insert into new_sale values (3,5);
```

The Problem

Insert the data from the *new_sale* table into the *sale* table using a single SQL statement. Note that the primary key for the *sale* table is (*book_id, sale_id*), where *sale_id* is a sequence number starting with 1, and incremented by 1 for each new record with the same *book_id*. Here is a listing of the contents of the *sale* table after the new data has been inserted.

BOOK_ID	SALE_ID	SALE_PRICE
1	1	20
1	2	19
1	3	18
1	4	17
1	5	16
2	1	10
2	2	9
2	3	8
2	4	7
2	5	6
3	1	5

Try solving this problem yourself without creating any additional objects such as stored procedures, triggers, sequences, or views, and send your solution to iggy_fernandez@hotmail.com. The best three solutions will be acknowledged in Part II of this article. ▲

Iggy Fernandez is the lead DBA for a Silicon Valley startup and is Oracle 10g certified. Previously, he was the manager of database administration for Corio, an application service provider (ASP), and was responsible for a mixed portfolio of nearly one thousand Oracle and SQL Server databases. You can contact him at iggy_fernandez@hotmail.com.

TECH TIPS

Super Detailed Execution Plan Statistics

By Roger Schrag

T KPROF is a tool provided by Oracle that reads a trace file generated by the database and formats the information into a useful report. As far as execution plan statistics are concerned, TKPROF simply displays whatever information it finds in the trace file.

Starting in Oracle 9i, if you enable SQL trace and set the statistics level for your session to ALL, Oracle will write additional statistics to the trace file that enable TKPROF to show how many consistent reads, physical reads, and physical writes were performed and how much time elapsed for each step of a statement's execution plan. Here is an example:

```
Rows        Row Source Operation
————-       ————————————————————————————-
  10        SORT GROUP BY NOSORT (cr=2004 r=656 w=0 time=9656825 us)
 990          TABLE ACCESS BY INDEX ROWID ORDERS (cr=2004 r=656 w=0 time=9647523 us)
1001            NESTED LOOPS (cr=23 r=6 w=0 time=13367 us)
  10              TABLE ACCESS BY INDEX ROWID CUSTOMERS (cr=7 r=3 w=0 time=822 us)
  10                INDEX RANGE SCAN CUSTOMERS_N1 (cr=2 r=1 w=0 time=424 us)
 990              INDEX RANGE SCAN ORDERS_N1 (cr=16 r=3 w=0 time=9810 us)
```

From this report you can see that the nested loops join between the CUSTOMERS table and the ORDERS_N1 index performed 23 consistent reads, 6 physical reads, and 0 physical writes, and took 13,367 microseconds. Two of the consistent reads came from a range scan of CUSTOMERS_N1, 5 from accessing the CUSTOMERS table, and 16 from scanning ORDERS_N1. The query got slow when the ORDERS table was accessed. This step added 1981 consistent reads (2004 minus 23), 656 physical reads, and 9,634,156 microseconds of elapsed time.

The query took over 9.6 seconds to complete, and almost all of the time was spent accessing the ORDERS table. If you did not have the extended statistics and only looked at the row counts shown at the left, you could not have known this.

The extra execution plan statistics were added in Oracle 9i. The intention by Oracle had been that these statistics would only be collected when the statistics_level parameter was set to ALL. Since this parameter defaults to TYPICAL, the extra statistics were not supposed to be collected by default.

Many releases of Oracle 9i up through 9.2.0.4 have a bug that causes Oracle to collect these extra statistics when SQL trace is enabled, even if the statistics_level parameter is not set to ALL. This bug has been fixed in a subsequent patch set. Many people have discovered that when they applied the 9.2.0.6 patch set to their system, the execution plan statistics disappeared from TKPROF reports. This is due to the bug fix.

So, the upshot is this: If you are running Oracle 9i or 10g and you would like to trace a session and see the most detailed execution plan statistics possible, set the statistics level in the session being traced with the following command:

```
ALTER SESSION SET statistics_level = ALL;
```

Setting the statistics_level parameter to ALL does impose a fair amount of overhead. For this reason, you probably will not want to set this parameter to ALL at the instance level. ▲

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, Darrin Swan, at darrin.swan@quest.com. ▲

Long-term event sponsorship:

LOCKHEED MARTIN

CHEVRONTEXACO

ORACLE CORP.

Thank you! Year 2005 Gold Vendors:

➤ Confio Software

➤ Database Specialists, Inc.

➤ Data Domain

➤ Embarcadero Technologies

➤ Quest Software

➤ Quovera, Inc.

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

$ TREASURER'S REPORT

Judy Lyman, *Treasurer*

Beginning Balance		
January 1, 2005		**$ 56,299.91**
Revenue		
Membership Dues	18,510.00	
Meeting Fees	640.00	
Vendor Receipts	5,000.00	
Training Day	2,355.00	
Advertising	---	
Interest	57.25	
Miscellaneous	110.00	
Total Revenue		**$ 26,672.25**
Expenses		
Regional Meeting	9,084.51	
Journal	6,420.17	
Membership	---	
Membership Administration	891.04	
Website	275.00	
Board Meeting	1,136.51	
Training Day	1,184.06	
Marketing	---	
PayPal	250.95	
Miscellaneous	---	
IRS	982.00	
FTB	830.00	
Insurance	---	
Total Expenses		**$ 21,054.24**
Ending Balance		
March 31, 2005		**$ 61,917.92**

How to Architect and Manage a Project (Part I)

by Ralph Hughes, MA, PMP, Ceregenics, Inc.

Building Things Right the First Time

A Q & A with Ralph Hughes of Ceregenics, Inc.

Ralph Hughes is quite an authority on the topic of enterprise architecture, so we thought he'd be a good person to speak with on our *NoCOUG Journal* theme of *building the proper foundation.* Because Ralph has a Project Management Professional (PMP) certification, he brings both the project management technical perspectives together.

When you are starting to analyze and architect a system for a client, Ralph advises that step one is finding or developing a draft of the project charter. This includes high-level requirements and restraints, along with a description of the project. "Without that," he says, "you will be shooting at the wrong target or shooting at no target at all."

From there you'll need to develop a scope statement to help you keep focused. Next is developing an architectural framework (see The Open Group's Architectural Framework at **www.opengroup.org/architecture**). Then Ralph says in his own practice he works to "sell" using UML and use cases.

"Cheaper, faster, better" is Ralph's quick response to the question of the benefits to be gained from laying the proper foundation of a system. "There is less rework. Things will integrate better and scale better."

What can you do as your company embarks on a new project? Ralph says, "As a senior member of the team, a DBA can be a champion who advocates that a successful endeavor needs project management and project architecture." He continues, "Someone needs to make sure all the design elements trace back to a requirement. They need to make sure a delivery date is not assigned until the technical people review the requirements. If there's no plan, then someone takes a SWAG (a silly, wild-*ss guess) at it, and that's how we end up with a 'death march' project."

To make sure you don't end up in this situation, be sure to read Ralph's article "How to Architect and Manage a Project." ▲

The software application industry is in a perpetual state of crisis. From Fredrick Brook's "Mythical Man" (1975) to Edward Yourdon's recent "Death March," (1999) much has been written over the past few decades as to why, yet little has been said to date about how often failure rises from the confusion between the role of the Project Architect (PA) and the Project Manager (PM).

Painting in broad brushstrokes the argument we will explore in detail below, many of the largest, most sophisticated organizations today still have not fully realized that it takes two types of leadership to deliver new or modified applications, what we term a project architect and a project manager. In confusing the roles, these organizations ask one person to fulfill both roles, and if the employer does not see the distinction between PA and PM, then it is more than likely that this unfortunate employee or contractor will be equally confused.

It is possible to simultaneously architect and manage a development project, but it requires a highly organized, schizophrenic mind. Unable to provide the schizophrenic half of that equation, this paper strives to communicate at least the organizational mind set required for succeeding in the PA/PM role. It brings into focus the fundamental differences between the two, enumerates for each the most important strategies to master, and finishes with a list of resources for developing greater depth in both. Hopefully such insights will help readers in PA/PM assignments more successful, but even make the role enjoyable as an interdisciplinary endeavor that provides an inestimable value to one's organization.

The High Cost of Muddling Through

On innumerable software development projects every year, a senior technical person is elevated to "project lead," and asked to deliver an undefined level of functionality by a given date using a very finite amount of money. These new project leads could use a big dose of training and guidance, but typically receive neither one. "They know the technology better than anyone," reasons management, "so of course their projects will be successful."

There are two dark secrets buried in this line of thinking: First, management is asking the project lead to fulfill two fulltime roles at once: project architect and project manager. Second, a third of all software projects fail, and two thirds of the rest are considered "seriously challenged." (Standish 1995.)

After one sits through enough post mortems for projects that have gone seriously over budget and over schedule, often contributing to their cancellation, certain causative factors appear time and time again:

> ➤ The project never received good or stable requirements from management and users.
> ➤ Management set unrealistic goals, demanding that too much be delivered at once.
> ➤ The estimates for the level of effort required were far too low.
> ➤ Not enough time was spent on design due to an excessive desire to begin coding.

The ubiquity of these observations suggests they are but symptoms of a deeper flaw in the way we approach

This dangerous practice of asking a senior technologist to simultaneously fill two roles without specific training in both allows the common neglect of thorough design and planning to occur.

software development. Time and time again the necessary design and planning are skipped. We need to look at the overarching process that allows these errors to repeatedly occur.

Even in large organizations, management frequently elevates senior technologists into "project lead" based upon their mastery of an important technology such as the DBMS. Occasionally, the organization has a Project Management Office that provides a few templates and a little advice, but more commonly all the lead receives is a copy of MS Project, leaving them with only their technical experience to draw upon as they guide their team through the complete development life cycle. This dangerous practice of asking a senior technologist to simultaneously fill two roles without specific training in both allows the common neglect of thorough design and planning to occur. The underprepared PA/PM role is a common but largely unacknowledged root cause of the application development crisis.

With the advent of such industry associations as the Project Management Institute and more recently the Worldwide Institute of Software Architects, both the PM and the PA roles have received a good dose of definition and training material. Data processing professionals in the know acknowledged that any non-trivial development project requires both a systems architect to define the application's design and a project manager to mind the budget and schedule. Yet far too rarely do projects have one of each assigned. Why?

Why PA/PM Roles are Inevitable

There are multiple reasons why the double duty of PM/PA is so frequently assigned to senior technical staff. One could say the practice is "over-determined." First, business managers are still largely ignorant of the "professional-

ization" of software architecture and project management. Project management has slowly pulled into focus only in the last decade or so after thirty years of investment by practicing PMs and academicians. What is worse, the software architect professional organizations lag behind their PM counterparts by twenty years at least. Consequently, many large institutions have only a cursory definition of a PM, and very few even recognize the notion of project architect. Without widespread clarity on these roles, managers sponsoring projects can honestly claim to be without better resources to assign.

Secondly, IT projects always look far smaller during conceptualization than after they are underway, letting managers honestly claim that their projects did not seem large enough to warrant two individuals with in-depth training in project leadership. Furthermore, managers perhaps willingly err on the side of overloading a project lead because it is far cheaper to elevate a senior staff member into doing two management roles at once rather than hiring separate and certified individuals for these positions.

Finally, the newly appointed senior developers have typically performed extraordinarily well on past projects and are victim of the "halo effect," in which management assumes that superb skills as a developer naturally translate into above-average ability to architect and manage a project.

Succeeding in the PA/PM Role

The forces and perhaps the ignorance that leads management into instructing senior technologists to walk unprotected into such a minefield may never be corrected, so project leads will need to venture forth forewarned and forearmed. Above all, project leads must understand that "project architect" and "project manager" are very different roles, with different life cycles, constituency, and skill sets. One cannot address both roles with the same mind set.

Taking this distinction firmly in mind, the project lead can only then pursue three initiatives that will greatly increase their chances of success in this overloaded job: a) pursue project architecture and project management as separately endeavors, b) acquire a starter "toolkit" for each role, and c) gather expertise in both from their respective professional associations.

After first exploring the definitions of PA and PM, we will examine each of these efforts in detail in the sections to follow.

Understanding that Architecture and Management are Different

The Project Management Institute defines an project manager as a person who initiates, plans, executes, controls, and concludes the activities required to fulfill the requirements of a project. Importantly, they do not actually "do" any of these activities. Consider the following list of responsibilities for the PM found in a typical project services company:

Whatever is needed to meet clients' objectives / expectations; (re)creating / (re)communicating the solution

vision and its potential; envisioning impacts upon operational systems and processes; planning, tracking status, coordinating team efforts and communication; managing client expectations; phasing the project plan; managing project risk factors; managing staff assignments; ensuring adequate team training; tracking and resolving project issues; ensuring quality processes are defined and followed; coordinating user acceptance of deliverables; tracking and managing all requests for change in scope or requirements.

In contrast, the definition of project architect is subject to far greater variety than that of a PM, given its shorter history as a specific project role. PAs must also define their place in a more crowded arena: given that IT shops already have such roles as software engineers, data modelers, and senior developers, exactly what is the focus of the "architect"? Drawing from such sources as the WWISA "field manual" (Malveau 2001) and the Software Engineering Institute's ABD architectural methodology (Bachman 2000), one can state that the PA focuses upon the highest level of system design, simultaneously ensuring that the application will meet the needs of the organization, properly guide the developers, and comply with internal and external standards. Software engineers and the like take up the work once the development team can name the object classes that will fulfill each bundle of functionality defined by the architect. Contrast the PM responsibilities above with this list of PA responsibilities typically found in the same hypothetical software project consultancy: overall design of the system(s); derivation of technical requirements; supporting formation of technical strategy; leading development team; educating members on standards and target design; communicating between technical sub-teams; setting and enforcing standards and team processes; selecting project software, tools, and development techniques; monitoring and incorporating likely changes in technologies ("futureproofing"); ensuring compatibility of all components; ensuring completeness of unit, system, volume testing.

Managing PA and PM Duties Separately

A review of the definitions of the PA and the PM reveals many intersections between the roles. First, they are both leadership roles, responsible for guiding the actions of the team and judging the quality of its deliverables. Secondly, they are both responsible for communicating to the sponsoring organization the obstacles and possibilities confronting the project, with the PA focusing on technical considerations and the PM attending to issues of time and money.

Yet they are not the same roles, and the project cannot succeed if only one is assigned, or if the project lead charged with fulfilling both roles emphasizes one at the expense of the other. The project lead must attend equally to both, and this goal is made easier by taking to heart three key differences between the roles, forcing each of them to be managed separately: The two roles manage different life cycles, face different struggles, and report to very different constellations of project stakeholders.

Different Life Cycles

The Project Architect is charged with the application and execution of the software development life cycle (SDLC). Whether the organization desires the project to employ a waterfall or spiral methodology, the sequence of "business requirements, analysis and design, development, testing, deployment, and maintenance" is the foremost construct in the minds of PAs as they plan the work of the project.

The Project Manager on the other hand is focused upon what PMI labels the "project management life cycle" (PMLC), which floats above the actual work steps of the project to focus upon how the team will plan, execute, control, and conclude *each phase* of work. In practical terms, when the project lead acts as PA, s/he will provide the detailed work tasks that must be accomplished to deliver the project. Each of these work tasks get assigned to the various phases of the project by the PM, so that the PMLC provides the compartments into which the output of the SDLC will be organized.

While acting as PM, the project lead will focus upon knowing when a new phase should be initiated (entailing much communication with management), when the team should provide detailed plans of the work within the phase, when each team member should be doing a specific task, how quality will be monitored and controlled, and finally when and how the phase should be wrapped up because all of its tasks have been completed.

Different Struggles

The struggle PMs face is easy to delineate, thanks to the work of the PMI. PMs strive for balance between the multiple poles of the "triple constraint," so called because above all else a project must find a solution to the competing demands to deliver results that are simultaneously 1) within budget, 2) on schedule, and 3) provide all the required functionality.

More often than not, meeting all three is impossible, and the PM makes trade-offs between time, money, and project scope to best meet the priorities of the organization. The term "triple constraint" is actually a slightly outdated term, because PMs can make trade-offs also on the quality of the deliverables, their reliance on key resources, the risks the project must accept, and the satisfaction of various subsets among the project's stakeholders. When confronted by a conflict between the seven poles of this "triple" constraint, the PM arrives at a balance by a) choosing between options identified by the team, b) reshuffling the sequence and staffing of tasks, and c) requesting more time or resources from management.

Project architects face a very different struggle than the PM and have very different means of achieving a balance. Their primary tool is to re-visit the design or their choice of building blocks with which the application will be built, making the major output of their balancing a combination of revised models or revised component libraries. Of the seven components involved in the PM's triple constraint, the PA can adapt the design only by altering the relative emphasis between risk, stakeholder satisfaction, quality,

and cost because these are the factors most affected directly by changes in design. Scope and schedule cannot be addressed without negotiating with management, and therefore lie in the domain of the PM.

While revising the design, there are several architect-only concepts that can be adjusted:

1) Their designs must ensure that the components of the application are **decoupled**, that is, involve as few interdependencies as possible that could make development, maintenance, and redesign difficult.

2) PAs must strive to **minimize complexity** of the design while achieving the greatest functionality, usually accomplished by simplifying the design to employ a dominant architectural strategy (such as "layering" or "repositories"), and by casting the majority of design components into as few conceptual objects as possible that will be used repeatedly throughout the design.

3) PAs must also steer their designs to comply with an ever increasing number of **standards** from both within and without the organization. Such compliance is part and parcel with their responsibility to "future-proof" their designs so that the resulting application is, at a minimum, maintainable and adaptable. Yet, the benefits of every compliance must be measured against the cost of the greater complexity that standards often introduce into the design.

Reviewing these points for the project lead who must serve as both PA and PM, we can offer two vital recommendations: First, be sure to attend to both life cycles at work in a project.

The sidebar on application "abilities" lists further concepts that the PA can employ to achieve the particular design balance they seek.

Different Bosses

Our final point of contrast between the roles of PA and PM is to examine the difference in their communication patterns. The PA's predominant concern is to keep the developers building the application in the prescribed manner, so that the results meet the project's quality requirements. Accordingly, the preponderance of their communication is inward, toward the team members. On the other hand, the PA communication pattern is outward – toward management, target users, and all other stakeholders out-

side of the team. So demanding are the information needs of this outer community, that PMI reports that 90 percent of a PM's work is communication, rather than planning and re-planning as one might expect. If one determines a person's "boss" as the party they must satisfy, then we can say the PM's boss is the party that signed the project charter, and the PA's boss is the project team who can only be mollified if the design makes sense and is doable in the time allotted.

Reviewing these points for the project lead who must serve as both PA and PM, we can offer two vital recommendations: First, be sure to attend to both life cycles at work in a project. First, put on the PA hat and paint a proper picture of the work that needs to be done. Derive an accurate and comprehensive work breakdown structure (more on this topic below) with truly reasonable estimates on the level of effort involved. Then switch to the PM hat and bring into focus the phases of the project between which the WBS tasks will be divided. Continuing as PM, be sure to initiate, execute, control, and close them properly. Only when each phase is properly proceeding can you switch back to PA to focus upon how the developers are realizing in code the components the architect designed.

Second, pay equal attention to the primary struggles of the PM and the PA. Whenever the management or other stakeholders to the team become concerned, realize you must address them as project manager and resolve the issue with a new balance of time, money, scope, quality, and key resources, negotiating with management as necessary. As project manager you can alter the sequence in which tasks are undertaken, the number of people tackling given assignments, or even pursue sequential assignments simultaneously. When it is the developers that express concerns, respond as an architect, addressing the challenge with a new choice of design or component libraries while leaving the deadlines, quality, and targeted requirements unchanged.

The project lead who solves PA problems with PM solutions or vise versa is risking disaster. Deadlines, targeted requirements, quality, or expense cannot be changed without communication outward to management as a PA is trained to do. And conversely, developer confusion and ineffectiveness cannot be solved by simply changing the schedule or adding more resources, as a PA would resort to, but instead requires a team review of the design and tools, often with a few modifications to be made.

Third, be prepared to draw different "lines in the sand" depending upon which role one is filling at the time. As architect, the project lead is ultimately responsible for making the system's high-level design everything it should be, that is, ensuring that the design covers the organization's needs in as economical a fashion as possible, and that it complies with all necessary internal and external standards so that it will not waste major portions of development and funds when it must be updated or maintained in the future. As architect, the PA in fact is ethically bound to resist the start of application development until the design has demonstrably achieved this high level of quality.

For PMs, the commitment is to ensure that the entire project management life cycle has been anticipated and doc-

umented. As part of the PMP certification process, the project manager must sign a pledge to abide by the PMI's Code of Conduct which includes accurately reporting anticipated and actual project timelines, budgets, and quality achievements, even when management desperately wants to paint a rosy picture to the larger community of stakeholders. As a product of that code of conduct, the PM is obligated to "push back" against management when asked to represent the project as ready to proceed when in fact it is not.

Get a Starter Toolkit for Both

While acknowledging the separate PA and PM threads involved is a necessary step, it is not sufficient all on its own. To guarantee project success, the project lead must also have a separate, ready-to-deploy toolkit for each side of the PA/PM equation. This section lists a technique that will greatly assist both roles, the WBS LOE from Delphi, and then focuses upon a starter bag of tricks for each role separately. Together these lists contain crucial strategies that are commonly overlooked by project leads if they do not get formal training in both roles. These lists will be presented only in summary fashion, the goal being to alert readers to their existence and set them in the most expedient direction as they begin to gather their implements. Source material for many of the tools mentioned can be found in the references for this paper. Furthermore, for true mastery of the two skill sets in question, project leads should mine the resources that we will review in the next section.

The WBS LOE from Delphi

The work breakdown structure (WBS) is a list of all the tasks that must be done to deliver the required product. A WBS for even a moderately scoped application can exceed one thousand tasks when fully elaborated. In theory, the project leaders must first derive the WBS and then estimate the level-of-effort (LOE) for each task in order to know the labor resources the project will require.

PAs and PMs live or die by the WBS LOE because it sets management's expectation of when the project will be delivered and how much it will cost. Any inaccuracy or omission in the WBS LOE will condemn the project lead to delivering the promised functionality without the time or the resources required. From whence should the project lead derive the hundreds of details and the accurate LOE estimates needed? Out of thin air?

The Delphi practice avoids "group think" in which team members subconsciously subscribe to the prevailing wind and eventually strive for quick closure rather than raising new ideas and challenging existing notions.

Unfortunately, this is what many project leads do, and the industry is all the poorer for the failed projects that this practice engenders. A better answer is to get an accurate and complete WBS LOE from Delphi, that is, from the Delphi technique. PMI states that the PM should not think up the WBS nor estimate the LOE of the tasks, rather, they should ask the team to do so. The team, after all, are the experts in how to build the required output. Does this notion suggest gathering the team into a room and having them brainstorm the WBS and the LOEs in a single sitting? This practice may work as a first step, but typically yields very poor results on it own.

A much better approach is to use the Delphi technique where the experts send in their suggestions regarding tasks and effort required, and the project lead collates them and remands the assembled whole back to the group for individual review and correction. The first iteration may involve tasks only, but the second and third and beyond should include LOEs for tasks.

TECH TIPS

Oracle DBA Scripts, Articles, Tools, and More

I f you're looking to add to your library of Oracle-related information, check out Oracle-Base. It's packed with tons of information, including dozens of DBA scripts; articles on Oracle10g, 9i, and 8i; industry news; tools and utilities; and more. It's also a great place to find out more about Oracle on Linux. You'll find Oracle-Base at http://www.oracle-base.com. Here is a quick sampling of a few main topics and subtopics:

DBA Scripts
- ➤ Monitoring
- ➤ Security
- ➤ Resource Management
- ➤ Constraints
- ➤ Script Creation

Oracle 9i Articles
- ➤ Oracle 9i Installation
- ➤ Oracle 9i XML

- ➤ Oracle 9i Web
- ➤ Oracle 9i Application Server

Oracle 10g
- ➤ Oracle Database 10g: New Features for Administrators OCP Exam
- ➤ Oracle Database 10g Installations
- ➤ Oracle Application Server 10g ▲

The Delphi practice avoids "group think" in which team members subconsciously subscribe to the prevailing wind and eventually strive for quick closure rather than raising new ideas and challenging existing notions. After several iterations of the Delphi technique, the PM will have a much wider universe of tasks and a range of estimates. Though out the scope for this paper, let us acknowledge that the range of estimates represents the risk entailed with each task, and therefore can be combined to yield the overall risk of the project as well, through what PMI calls the "PERT" estimating technique.

Project Architect's Resources

The relative youth of the software architect profession is in this case a good thing, largely because the project lead can get involved with the definition of the profession. Helping create a body of knowledge is always far more interesting than memorizing lists that some committee has compiled over the years. There are several professional computer associations, such as DAMA, IEEE, and the others listed in Table 1, that cover at least some piece of the PA's work domain. These organizations are well established and should be among the first places a project lead should mine for a particular architectural question, however, they all either focus on a particular application of software or take a extremely broad perspective on data processing.

More expediently, the project lead seeking a bedrock foundation in software architecture should explore a younger association, the Worldwide Institute of Software Architects, exclusively focused upon the definition, education, certification, and practice of software architects. WWISA has an excellent "field manual," ***Software Architect Bootcamp***, in which project leads will find several crucial items such as:

➤ the big picture of integrating enterprise, application, and platform architecture.
➤ comparisons of several architectural methodologies.
➤ an overview of the major architectural strategies for managing complexity.

WWISA is currently defining a body of knowledge (BOK) for an software architect (SWA), which will then serve as a basis for at least a level one SWA certification. There could be no better place for the project lead to acquire both the required breadth and depth of a PA than the WWISA's discussions of the books, standards, and methodologies should comprise the SWA BOK.

Table 1. Some Important Resource Organizations for Software Architects
(Ratings are the author's, and specific to the topic of this paper only.)

AITP*   **Association of Information Technology Professionals –** Founded 1951 for "machine accountants," retains "big iron" orientation. Closely affiliated with the Institute for the Certification of Computer Professionals (ICCP), which provides many, detailed certifications upon which some organizations have based their own certification programs. **www.aitp.org**

(chart continues next column)

ACM*   **Association for Computing Machinery –** Founded in 1947, an important provider of research and publications on a wide range of focused topics in computers including hardware architecture, programming languages, and data management. **www.acm.org**

DAMA*   **Data Management Association –** Focused upon the area of data resource management, currently establishing special interest groups in Data Quality, Data Architecture, Data Modeling, Data Repositories, Data Stewardship, Data Warehousing, and Data Management Lexicon. **www.dama.org**

IEEE*   **Institute of Electrical and Electronics Engineers –** Large and established professional association in a broad computer engineering topics. Producer of 30 percent of the world's published literature in electrical engineering, computers and control technology. Creator of many standard recommendations later accepted by international and governmental organizations. **www.ieee.org**

OMG**   **Object Management Group –** Association funded by corporations, universities, and standards bodies. Strongly involved with furthering object-oriented paradigms. Best known for its object request broker (ORB), and Common ORB Architecture (CORBA). Provides/supports many evolving methods for architecture, including Architecture-Based Design (ABD), and TOGAF (The Object Group's Architectural Framework), Unified Modeling Language (UML), and Common Warehouse Meta-model (CWM). **www.omg.org**

SEI**   **Software Engineering Institute –** Federally funded research and development center with sponsorship from the Department of Defense. Operated by Carnegie Mellon University. Provides technical leadership to advance the practice of software engineering with the hope to empower organizations to acquire/sustain s/w- intensive systems with improved cost, schedule, quality. Creators of CMM, the Capability Maturity Model. **www.sei.cmu.edu/about/about.html**

WWISA**   A nonprofit corporation founded to accelerate the establishment of the profession of software architecture and to provide information and services to software architects and their clients. Current efforts include promoting formal academic degrees and certification programs for software architects, standards for tool builders and researchers, and practice definition to give customers more accurate expectations and increased confidence. **www.wwisa.org**

Project Manager's Resources

Fortunately for the project lead, the Project Management Institute provides a readily accessible source of PM expertise. PMI was founded in 1969 and is now the world's leading not-for-profit professional association in the area of project management. PMI has active chapters around the globe focusing upon standards, certification, research, and education of PMs and their profession.

The best first step toward PM knowledge is to is to purchase PMI's Project Management Body of Knowledge (PMI 2004), which presents a framework and details for

nine "knowledge areas" defining the full scope of project management that project leads should consider as they guide their teams toward success. This "PMBOK" is a reference, and therefore somewhat opaque upon first reading. Many project leads will find it far more time efficient to also acquire a good introduction to project management (e.g. Mulcahy 2002 or Heldman 2002) available through the www.pmi.org.

The world of project management is so well documented and formalized that a project lead should in fact seriously consider earning the PMI's Project Management Professional (PMP) certificate. The PMP is best attained by attending a PMP boot camp and then investing another 40 hours of study before taking the PMI's exam. In exchange for this moderate amount of effort, project leads acquire of course a good familiarity with several useful tools (such as the WBS and Delphi technique), but more importantly the confidence that they have a solid understanding of the breadth and depth of the PM role they are playing.

The PMP certification process can be quite consuming, so as a final word on acquiring PM expertise for IT projects, we must consider "bringing it all back home" with Edward Yourdon's *DEATH MARCH, (1999)* a guide to surviving impossible IT projects. Yourdon defines a death march as a project with only half the time it needs, half the money required, or both. The majority of IT projects develop the pall of a death march, hence our industry's miserable record of project failures. Project leads are duty bound to guide their teams and their employers as far from one as humanly possible. ▲
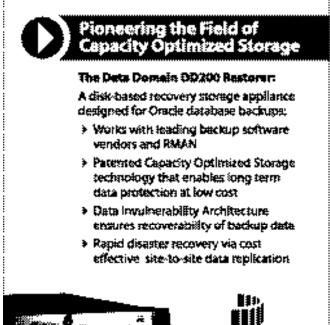
The world of project management is so well documented and formalized that a project lead should in fact seriously consider earning the PMI's Project Management Professional (PMP) certificate.

About the Author

Ralph Hughes, M.A., PMP, serves as Managing Architect for Ceregenics, Inc., a data warehouse consulting association in Denver, Colorado. With 24 years in data processing, he has spent the last 10 years as data warehousing architect for the telecommunications, aerospace, government, and pharma/biomed industries. He is a founding member of the World Wide Institute of Software Architects (www.wwisa.com <http://www.wwisa.com/>) whose mission is to streamline and popularize international standards for enterprise architecture. He can be reached at ralph.hughes@ceregenics.com or 720.951.2100.

NoCOUG Spring Conference

Thursday, May 19, 2005

Session Descriptions

For more detailed descriptions and up-to-date information, see www.nocoug.org.

KEYNOTE

Why "System" Is a Four-Letter Word
Cary Millsap, *Hotsos Enterprises Ltd.*

Perhaps the most common origin of failed performance improvement projects is the failure to communicate. At the root of the matter in the Oracle world is the word system. This presentation explores the reasons for the problem, the consequences of the problem, and how to solve the problem once and for all.

TRACK 1

Profiling Oracle: How It Works
Cary Millsap, *Hotsos Enterprises Ltd.*

Conventional Oracle tuning methods are outrageously complex, and they don't work. For more than three decades, good software developers have known how to optimize computer-executed tasks using a simple method that works every time. This presentation describes how to use this method in an Oracle environment, and it demonstrates the method's use with an example.

An Oracle 10g Upgrade Case Study: Looking at System Performance Before and After the Upgrade
Roger Schrag, *Database Specialists, Inc.*

Our company uses an Oracle database and an application developed in-house to manage information regarding customer databases. In this presentation we will discuss our company's experience moving this system from Oracle 8i to 10g. We won't focus on the migration steps themselves, because you can read about that in the documentation. Instead we'll focus on our real-life experience of how Oracle 10g handled an application that ran very smoothly on

Oracle 8i. In particular, we will look at changes in execution plans (for better or for worse), management of resources such as memory and CPU, and what the various advisors in Oracle 10g had to say.

SQL Tuning: An Overview of Why and *How* to Tune SQL
Dan Tow, *Singing SQL*

SQL tuning frequently offers the best opportunities to enhance overall application performance. It is, fortunately, a highly decoupled problem, enabling even a tuner completely unfamiliar with the application to rapidly find safe, easy-to-implement tuning recommendations that can deliver huge performance improvements. This makes the discipline of SQL tuning highly leveraged, with high rewards for the specialist. This presentation will show an overview of the SQL tuning approach described by Dan Tow in his book *SQL Tuning*, published by O'Reilly.

Future of DBMS and DBA
Noel Yuhanna, *Forrester Research*

Although DBMS technology is very mature, vendors continue to find new ways to innovative, expanding its scope and supporting new types of applications. Besides relational databases, more products are offering integrated support for XML, web services, and unstructured data types. So what can you expect in the future for DBMSes? Will database management challenges still exist in the future? This presentation looks at key areas where DBMSes will continue to innovate, such as automation, on-demand computing, unstructured data, data grids, and in-memory technologies. Learn how this will impact DBAs and how they should prepare themselves to adapt to these new changes.

TRACK 2

Index Organized Tables—Are They Right for You?
Scott Martin, *Terlingua Software*

By storing non-key values along with the key values in an index, Oracle can dramatically decrease the work needed to perform certain queries when compared to a standard B-Tree implementation. This talk not only covers how Oracle implements IOTs, but also walks through detailed examples using IOTs, backed by statistics and execution plans. We will also cover the effects on performance of the overflow table and secondary indices created on IOTs. You will leave knowing when and when not to consider using IOTs in your application.

JDeveloper 10g View Layer Alternatives—JSP and UIX
Peter Koletzke, *Quovera, Inc.*

The user interface (View) layer for Java-based web applications is often coded as JavaServer Pages (JSP) files, but Oracle offers an alternative, called ADF UIX, that defines the page using XML instead of HTML. UIX provides some benefits compared to JSP pages, but it has many similarities.

Development of both of these alternatives is well supported in Oracle JDeveloper 10g. This presentation explains these two view layer technologies and describes how to develop them using Oracle JDeveloper 10g. It reviews the principles of the MVC design pattern and how work in JDeveloper follows that design pattern. The presentation also discusses both JSP and UIX architectures, code libraries, coding styles, benefits and drawbacks, key features, and intended uses. In addition, it provides conclusions about which style is best for specific situations. Additionally, the presentation demonstrates how to develop an application in both styles using JDeveloper.

Tuning Batch Performance—A Real-World Case Study
Virag Saksena, *Auptyma*

In this presentation we will take an example from the real world of slow programs, and walk through the analysis and implementation processes. We will cover querying Oracle fixed views to get a quick indicator of bottlenecks and tracing program execution to get a more comprehensive perspective. We will look at different types of fixes, assessing their risk, and effectiveness and effort. We will also be measuring the impact of changes. We will use a case study from a real-world example to put some meat behind each of these, so attendees will come away with a good understanding of the performance tuning process.

Oracle Workload Measurement
Andy Rivenes, *Lawrence Livermore National Laboratory*

This presentation will explore Oracle workload measurement and how it relates to Oracle system management. Our profession confuses the processes of performance tuning and workload management, and this presentation will try to draw out the distinctions between the two. It will also address what system performance really means and then will explore how workload measurement can be used and what economic benefits can be achieved. Real-world examples will be used to illustrate how workload measurement can be used as inputs to capacity planning and workload optimization. Although this is not a tool presentation, a couple of options will be addressed to allow participants to implement workload measurement in their environments.

TRACK 3

XML—What's in It for Me?
Greg Matus, *Maverick Solutions, Inc.*

XML has been touted as the lingua franca (a medium of communication between peoples of different languages) and has become an important ingredient in enterprise-level web applications. With the latest releases of Oracle and JDeveloper, Oracle has significantly increased its support of XML. This presentation will explain what XML is and what all the fuss is about. Examples will be presented demonstrating a broad range of uses and benefits of XML. Examples of XML will include supplying data to a web page, storing project metadata, and allowing client-side parsing and formatting of a server-side result set. Examples will also illustrate

support for XML in JDeveloper, the Oracle database, Java, JSPs/Servlets, and HTML.

What's New with Oracle's Identity Management Solutions
David Saslav, *Oracle Corporation*

Oracle Identity Management includes a collection of components for managing user identities and controlling user access to applications. Oracle Application Server and Oracle Database have a unified framework for identity and access management that is used by applications throughout the Oracle enterprise. This session provides an overview of new features and capabilities in Oracle Identity Management. These include support for heterogeneous application environments, advanced deployment topologies, expanded authentication modes, and multiple security enhancements. Attendees learn how and why to leverage these features in their Oracle deployments.

Oracle Database 10g: Automatic Performance Diagnosis and Troubleshooting
Prabhaker Gongloor (GP), *Oracle Corporation*

This presentation focuses on automatic performance diagnosis and troubleshooting capabilities in Oracle 10g. The presentation will cover the following topics:
1. Overview of the intelligent infrastructure supporting the performance self-diagnosis capabilities
2. Performance overhead associated with the intelligent infrastructure
3. How to use the EM interface to effectively diagnose performance problems in a proactive and reactive manner
4. Best practices and lessons learned from real-world customer deployments using this technology

Introduction to the SEI Capability Maturity Model (CMM)
Jeffrey Jacobs, *Jeffrey Jacobs and Associates*

This presentation will provide an overview of the Software Engineering Institute's Capability Maturity Model (SEI/CMM). Topics will include the purpose of the SEI/CMM, the five levels, the importance of metrics, and how the SEI/CMM framework can be used to understand and improve the quality of software development. ▲

Important Note Regarding Conference Attendance

*A*ll conference attendees, vendors, speakers, and NoCOUG staff must RSVP prior to May 12 in order to receive a drive-in pass at the Visitors Center. The same forms of identification are required as last year.
U.S. citizens—U.S. government issued photo ID (i.e., drivers license, U.S. passport). Permanent residents—green card along with U.S. government issued photo ID (i.e., driver's license, U.S. passport). Foreign nationals—due to more stringent security requirements imposed by Lockheed Martin and the Department of Defense, foreign nationals are not allowed entry into the Lockheed Martin facility and therefore cannot attend the meeting this year. ▲

NoCOUG Spring Conference Schedule

Thursday, May 19, 2005 · Lockheed Martin, Bldg. 157, Sunnyvale, CA

Hosted by Lockheed Martin Corporation. Please visit **www.nocoug.org** for session abstracts, for directions to the conference, and to submit your RSVP.

8:00–9:00 a.m.	**Registration and Continental Breakfast**—Refreshments served
9:00–9:30	**General Session and Welcome**—Darrin Swan, NoCOUG President
9:30–10:30	**Keynote:** *Why "System" Is a Four-Letter Word*, Cary Millsap, Hotsos Enterprises Ltd.
10:30–11:00	**Break**
11:00–12 p.m.	**Parallel Sessions #1**
	Track 1: *Profiling Oracle: How It Works* by Cary Millsap, Hotsos Enterprises Ltd.
	Track 2: *Index Organized Tables—Are They Right for You?* by Scott Martin, Terlingua Software
	Track 3: *XML—What's in It for Me?* by Greg Matus, Maverick Solutions
12:00–1:00	**Lunch**
1:00–2:00	**Parallel Sessions #2**
	Track 1: *An Oracle 10g Upgrade Case Study: Looking at System Performance Before and After the Upgrade* by Roger Schrag, Database Specialists, Inc.
	Track 2: *JDeveloper 10g View Layer Alternatives—JSP and UIX* by Peter Koletzke, Quovera, Inc.
	Track 3: *What's New with Oracle's Identity Management Solutions* by David Saslav, Oracle Corporation
2:00–2:15	**Break**
2:15–3:15	**Parallel Sessions #3**
	Track 1: *SQL Tuning: An Overview of Why and *How* to Tune SQL* by Dan Tow, Singing SQL
	Track 2: *Tuning Batch Performance—A Real-World Case Study* by Virag Saksena, Auptyma
	Track 3: *Oracle Database 10g: Automatic Performance Diagnosis and Troubleshooting* by Prabhaker Gongloor, Oracle Corporation
3:15–3:45	**Raffle and Refreshments**
3:45–4:45	**Parallel Sessions #4**
	Track 1: *Future of DBMS and DBA* by Noel Yuhanna, Forrester Research
	Track 2: *Oracle Workload Measurement* by Andy Rivenes, Lawrence Livermore National Laboratory
	Track 3: *Introduction to the SEI Capability Maturity Model (CMM)* by Jeffrey Jacobs, Jeffrey Jacobs and Associates
5:00–6:00	**NoCOUG Networking and Happy Hour**
Cost:	$40 admission fee for nonmembers. Members free. Includes lunch voucher.

Session descriptions appear on pages 26–27.

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

Cover photos copyright © Photodisc, Inc.

NoCOUG

P.O. Box 3282
Danville, CA 94526

NoCOUG is excited to announce that Chris Date, one of the founding fathers of relational database theory, will deliver the keynote address at our Summer Conference. Don't miss our interview with Chris Date on page 4.