

Official Publication of the Northern California Oracle Users Group

NoCOUG

J O U R N A L

VOL. 19, No. 3 · AUGUST 2005

\$15

Gliding Through Changing Times

Are Times Changing for Oracle DBAs and Developers?

*We asked some Oracle authors and trainers to look into the future.
See page 4.*

Optimizing Oracle 10g on Linux

*Get the scoop on Automated Storage Management from Bert Scalzo.
See page 19.*

Managing and Architecting a Project

*In part two of this series, Ralph Hughes outlines the toolkits you'll need.
See page 12.*

Much More Inside . . .

From the Editor

It's been fun to put this issue of the *NoCOUG Journal* together. Laurie Robbins and I had the opportunity to talk with several leaders in the Oracle community to find out their thoughts on the future of Oracle and the roles of database administrators and developers. One of the perks of doing this volunteer work is having the opportunity to call up Craig Shallahamer or Guy Harrison and ask them their thoughts on various issues of the day.

Over the past twelve months, we've had the opportunity to speak with many industry luminaries, including Chris Date, Jonathan Lewis, Gaja Krishna Vaidyanatha, and more. Now, we're asking for your input. Who would you like to know more about? Have you been intrigued by an Oracle-related book you've read and want to know more about the author? Have you seen a presentation and want to know the speaker's background and how they achieved their success? Let us know who we should track down next! Just send a note to journal@nocoug.org. We look forward to hearing from you.

—Lisa Loper,
NoCOUG Journal Editor

Table of Contents

Editor's Note	2	Advertising Rates.....	26
NoCOUG Board	2	NoCOUG Summer Conference Schedule.....	28
Publication and Submission Format.....	2		
President's Message.....	3	—ADVERTISERS—	
Gliding Through Changing Times	4	Database Specialists.....	9
Having a Laugh with Mogens Nørgaard.....	6	Embarcadero Technologies.....	9
SQL Corner.....	7	Quovera.....	15
How to Architect and Manage a Project.....	12	RightOrder.....	15
Optimizing Oracle 10g on Linux Using Automated Storage Management.....	19	Data Domain.....	16
Sponsorship Appreciation.....	24	Confio.....	16
Treasurer's Report.....	24	Quest Software	18
NoCOUG Summer Conference Session Descriptions.....	25	EMC ²	27

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 November 2005 issue is October 1, 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.

Copyright © 2005 by the Northern California Oracle Users Group. Permission to reproduce articles from this publication, in whole or in part, is given to other computer user groups for nonprofit use, with appropriate credit to the original author and the *Northern California Oracle Users Group Journal*. All other reproduction is strictly prohibited without written permission of the editor. Two copies of each reprint should be sent to the editor.

NoCOUG BOARD

President

Darrin Swan, Quest Software
darrin.swan@quest.com

Vice President

Colette Lamm, Independent Consultant
colette_lamm@yahoo.com

Membership Director

Joel Rosingana, Independent Consultant
joelros@pacbell.net

Director of Conference Programming/Past President

Roger Schrag, Database Specialists, Inc.
rschrag@dbspecialists.com

Webmaster

Eric Hutchinson, Independent Consultant
erichutchinson@comcast.net

Journal Editor

Lisa Loper, Database Specialists, Inc.
lloper@dbspecialists.com

Vendor Coordinator

Diane Lee, Lockheed Martin
diane.c.lee@lmc.com

Secretary/IOUG Representative

Iggy Fernandez, Intacct
iggy_fernandez@hotmail.com

Director of Marketing

Jen Hong, Cisco Systems
Hong_jen@yahoo.com

Director of Public Relations

Les Kopari, Corner Pine Consulting
(650) 802-0563

Training Day Coordinator

Hamid Minoui, Schwab
hamid.minoui@schwab.com

Treasurer/Track Leader

Randy Samberg, PalmSource
rsamberg@sbcglobal.net

Member at Large

Naren Nagtode,
Franklin Templeton Investments
nnagtod@frk.com

Other Contributors

Assistant Journal Editor

Laurie Robbins
lrobbinsmis@yahoo.com

NoCOUG Staff

Nora Rosingana, Independent Consultant

Staying Up to Date as an Oracle Professional



Darrin Swan

There are terabytes worth of published data available daily covering all the various topics that are important for you to know as an Oracle professional. Mining through all the available content and finding what's valuable and pertinent for you is a daunting task. I personally work to stay informed by regularly reading through my own datamart of Oracle content, with topic areas ranging from database application development and software quality and testing to database administration. I also work to stay up to date on the latest industry trends through special interest groups and analyst reports. Here are some of the publications I read:

- *NoCOUG Journal*
- *Oracle Magazine*
- *Database Trends and Applications*
- *Application Development Trends*
- *eWEEK*
- *Dr. Dobb's Journal* (yes, I admit it)
- *DM Review*

Combine all these publications, email newsletters, and alerts received, and the volume of newly published content on database trends is staggering. Of course, a favorite source of quality Oracle content that delivers the goods every time is the *NoCOUG Journal*. With the quality of content published, it amazes me that the journal is created by two very talented and dedicated volunteers, Laurie Robbins and Lisa Loper.

Each quarter the *NoCOUG Journal* team is hard at work interviewing industry luminaries, writing and editing articles, and seeking out new, valuable content to deliver usable best practices for developing and managing Oracle technology. I am personally impressed with Laurie and Lisa's

appreciation and understanding of what information will be meaningful and valuable to the NoCOUG membership. Remember, each new quarterly journal is accomplished by only two NoCOUG volunteers, with very tight and pressing deadlines. It is no wonder that these two have won the IOUG's Best and Most Creative Oracle Users Group Journal Award two years in a row.

In this summer edition, industry luminary, author, and columnist Guy Harrison discusses his *thoughts on the future of Oracle technology*. The journal also covers topics of interest to Oracle DBAs and developers. As you read through this edition, think about your own Oracle technology strategies and experiences and consider writing an article for the journal. If you have articles you're interested in publishing or just want to thank our *NoCOUG Journal* team for a job well done, send Laurie and Lisa a note at journal@nocoug.org.

On a related topic, Laurie's recent interview with Chris Date, one of the founding fathers of relational database theory, translated into a very interesting article in our last journal and a keynote address by Date at the upcoming summer conference. On August 18 at Chevron in San Ramon, Chris Date will kick off our well-attended conference with his address, "Foundation Matters." This will be followed by twelve technical sessions, including a live RAC demo, database development and administration topics, data risk management, and real-world Oracle case studies from NoCOUG members. See the back page of the journal for conference details and presentation abstracts.

Thank you, Laurie and Lisa, for another great *NoCOUG Journal*!

See you August 18 in San Ramon. ▲

Gliding Through Changing Times

“Continual self-education will be the new mantra of the successful DBA,” wrote Howard Fosdick in 1996. We searched the archives of Intelligent Enterprise’s Database magazine, and found the following future prediction from Fosdick: “The current trend toward GUIs will continue and be accelerated by multimedia interfaces, making typical DBA tasks much easier. The continued explosion of the server population mandates ever-higher levels of server automation for distributed systems management, ultimately culminating in self-administered systems. Intelligent agents and self-monitoring systems will be required to control increasingly dispersed and diverse distributed databases. These developments will whipsaw DBAs. On one hand, the demand for DBA services will be undercut as jobs are increasingly de-skilled and automated, with centralized control ever more possible for widely distributed systems. On the other hand, the geometric explosion in the number and functionality of machines ensures that DBA expertise will be more valued than ever.”

Believe it or not, the potential threats of automation and outsourcing to your job as a developer or database administrator are not unique to our current times. These professions flourished with the introduction of software programs and database software in the ’70s and ’80s. And with technology changing so rapidly, it’s no surprise that these professions are going through changes, like many other professions. In the early ’90s there was “buzz” in the industry of centralizing computer support, similar to the talk of outsourcing today. Likewise, software automation to keep staffing costs down has been on the minds of managers and software companies since the introduction of computers and programs. Go to any NoCOUG meeting, and you’re bound to hear colleagues discuss the future of Oracle and jobs as Oracle DBAs and developers.

As with most professions, changing times create changes to the demands and skills needed for your career. But how do you stay on top and drive your career in the direction that will keep you afloat? Predicting the future is not a science, but there are indicators that can be evaluated to help understand what the future might hold. We glanced at the past and then tried to see the future by talking to some industry leaders to get their opinions on what the future holds for the future of DBAs and developers. We asked them what skills are likely to be in demand, how to keep involved with the changes in the industry, and more. We’d like to share these discussions with you.

Stay on top of the learning curve.

Oracle trainer Craig Shallahamer of Orapub says that Oracle technology will always keep moving forward. “There’s always a lag between what’s being automated and the new features,” says Shallahamer. “If DBAs stay stagnant, the automation will pass them. If they keep moving, they’ll

be okay.” One example he gives is dictionary cache sizing. “If that is your skill, you’ve been replaced by Oracle,” he says. But there are always new things to learn to stay ahead of the curve. “If you look at Oracle Applications today, is it becoming more or less complex? While the individual components may be simpler or automated, the overall application is much more complex.”

Guy Harrison, a Chief Architect for Database Solutions at Quest Software discusses the job of a DBA, “Fundamentally the DBA is responsible for the integrity, functionality, availability, security and performance of the database. “ These fundamentals haven’t changed much over the years, but the way they are executed has changed. Harrison adds, “the impact of self-managing features on the lives of a DBA is likely to be more evolutionary than revolutionary. Oracle has made more of a marketing pitch about self-management in release 10g, but it’s really just a further progression of making the database easier to manage that has been going on since the very early releases of Oracle.”

He concurs with Shallahamer and continues, “Older technologies are becoming easier to manage, while the new technologies, RAC for example, are giving the DBA more to manage.” He also believes tasks such as database design and tuning will always be difficult to self-automate.

Dan Hotka, a training specialist with over 22 years of experience with Oracle products, says, “There’s always something new to learn for DBAs and developers, and that’s the exciting thing about working with a product like Oracle. The learning curve never ends.” Hotka makes the case that change creates opportunities.

Challenges Ahead

Harrison says some of the greatest challenges to the job of a DBA today are to be focused on trying to do more with less.

He says, “It’s more difficult to solve problems through hardware or software purchases, since managers are asking DBAs to reduce the amount of money spent on the database.” He thinks this increases the challenge for DBAs to get the most of what they’ve got, and resourceful DBAs are learning to leverage their tuning skills to manage this challenge.

It must be said that not all Oracle visionaries feel as strongly about a stable future for Oracle DBAs. When asked about what computer science students can do to prepare for a DBA career, Mogens Nørgaard of Miracle Consulting says, “I can’t answer your question without being controversial.” He goes on to state, “I honestly think that there is not a lot of future in the database world. If you see the trends at the moment, it’s all about moving all of the work out of the databases. All the work has to be done these days by mid-tiers, several mid-tiers with Java, XML and .Net applications. If all the work is done outside the database, all the money is allocated outside the database. Then nobody cares about the database. We can see the trend in our company.” He says they are doing more and more troubleshooting and work with the application layer with Java, XML, and such, and that Java programmers think of the database as “persistence storage.”

And what about some of the other potential threats to technology jobs, such as offshoring and outsourcing? Harrison says, “Outsourcing looks good on paper but the reality of having the support organization in a distant country . . . well it only takes a few incidents or failures before the value is seen. Ultimately it’s a marginal amount of savings when you factor the risks, especially for a mission critical production system.” He believes that although larger companies will find ways to make it work for part of the organization, mission critical support will stay local.

Trends for the future

Shallahamer sees that specific DBA duties have usually evolved from being handled by a DBA, then offshoring, then automation. To stay ahead of the curve, he recommends looking at industry trends. “Be careful of Oracle marketing,” he warns, “but look at it.”

“If I were a young DBA starting out, I would get to know RAC very well.” Shallahamer also recommends that Oracle professionals get as much experience as possible managing many Oracle databases. He believes DBAs of the future will be managing “a crazy number” of Oracle databases. He recommends having “a solid method of diagnosing Oracle problems. If you have a solid method, the technology can change, but the principles are the same. You have to have a solid foundation and then learn the new features.” Shallahamer also stresses the importance of learning how to effectively work with people.

Hotka also believes you have to be the navigator of your career and you can start where you are today. He says, “I don’t know of many companies that will come to you and ask you to work on something new. You have to be proactive; go to your manager and say you would like to be a part of the new project coming up.” Of course, it helps to be prepared for these opportunities, so continual education is key.

Harrison believes the top jobs for DBAs in the future will go to those who know more than one database platform. He says, “Pay attention to MySQL; the projected growth for database servers is Linux so the demand for MySQL is likely to grow. And although Oracle also runs on Linux, the number of companies that support more than one database platform is growing.” He also points out that SAP is working with MySQL to get their software suite integrated, which is another indicator of an increased demand.

Hotka sees skills with PHP and Java continuing to increase in demand. He says, “These products started off hot and caught on with many companies. New development will continue, but maintaining existing code will also keep up demand.”

Summary

The consensus does seem to indicate that the outlook is bright. That’s not to suggest there won’t be challenges, but to stay on top, the experts say you should stay informed of new features and learn them. Your tuning skills will be a valuable tool in keeping your career ahead. And your ability to multi-task and develop ways to manage a large number of databases will be key. Consider learning other database software products like MySQL. Remember that your manager or client probably has a shrinking budget, so if you can find ways to help reduce costs, you will become more valuable. ▲

Guy Harrison is a recognized Oracle expert with over 15 years experience in Oracle administration, development, performance tuning and project management. Harrison is the author of Oracle SQL High Performance Tuning (Prentice-Hall, 1997, 2000) and Oracle Desk Reference (Prentice-Hall, 2000) and is a regular contributor to Oracle-related technical journals. He was chief architect of Quest’s Spotlight™ family of diagnostic products, and collaborated on the design and development of other Quest tools for monitoring, tuning and administration. Harrison obtained a B.S. degree from the University of Melbourne in 1981.

Dan Hotka is a training specialist who has over 27 years in the computer industry and over 22 years of experience with Oracle products. He is an internationally recognized Oracle expert with Oracle experience dating back to the Oracle V4.0 days. Dan’s latest book is Oracle10g on Linux by Oracle Press. He is also the author of Oracle9i Development by Example and Oracle8i from Scratch by Que and has co-authored 6 other popular books. He is frequently published in Oracle trade journals, and regularly speaks at Oracle conferences and user groups around the world. Visit his website at www.DanHotka.com. Dan can be reached at dhotka@earthlink.net.

Mogens Nørgaard has over 17 years of experience working with Oracle database technology. He is the cofounder of Miracle A/S, which provides consulting, support, and training on Oracle and SQL Server technologies in Maaloev, Denmark.

Craig Shallahamer is the founder of OraPub, Inc. The company’s mission is “Doing and helping others do Oracle Performance Management.” OraPub offers Oracle consulting and training, and sells a rightsizing product called HoriZone at www.orapub.com.

Having a Laugh with Mogens Nørgaard

Sit with Mogens Nørgaard for a few minutes or attend one of his technical sessions, and you'll soon be laughing out loud along with everyone else in the room. While sharing his wisdom about things Oracle, he sprinkles in comments about his numerous ex-wives, plans for a brewery, and drinking at his oak table. We caught up with him last year when he came to the Bay Area during Oracle OpenWorld.

As a visitor from Denmark, Mogens wasn't speaking at Oracle OpenWorld, he was passing through on his honeymoon. "I just came to show Mrs. 'No Good' the city," he deadpans. So, one wonders if his comments about ex-wives are real. "Yes, I have two real ex-wives and several unreal ex-wives," he answers. Hoping this is the last marriage, he says, "It's too expensive to keep buying all this furniture. I always joke that the reason that I like my oak table so much is that I've bought it a couple of times already."

In 1987 Mogens started working with Oracle technology as an Oracle DBA at a bank using version 5. In 1990, he joined Oracle Corporation and was in the support organization for 10 years. He became the Manager of Premium Services in Europe and managed 40 people. "I traveled around a lot, worked with Cary Millsap, and taught many classes." When that came to an end in 2000, Mogens founded a company called Miracle A/S along with an old school friend. They currently have about 14 employees.

Mogens is known for bucking the trends and not following the party line with Oracle. "I've always been different in that way," he says. "If somebody was a bully in the sixth grade bullying someone else and everyone else was just looking at the bully without saying anything, I would attack him. Not to say that Oracle is a bully. I've always spoken my mind. I can't believe that I did that when I worked at Oracle. I like Oracle a lot. I enjoyed working there for 10 years."

"If you're talking about my article entitled *You Probably Don't Need RAC*, at least it got me an immediate invitation to meet with the VP responsible for High Availability at Oracle." Mogens says that RAC is probably good for some people, but not good for most people. Then he says that "whenever I recommend that people don't use something, they start using it immediately." When he wrote something with James Morle about RAID, "Everybody starting using RAID 5 systems," he says, continuing, "Since I wrote my RAC paper, sales have exploded." We suggest they should honor Mogens. He agrees. "If I say don't use 10g, sales will go absolutely through the roof. That's the sort of weight I carry on the market," he laughs.



Mogens Nørgaard (right) with James Morle announcing the formation of Ban Against Any Raid Five (BAARF).

Mogens has learned to adapt in changing times, especially with all the new features of Oracle. "One of my guiding principles in life is that when something reaches the state of perfection, it will be replaced by something more chaotic."

He says that the philosophy of technology makers has been that if we just increase complexity a little bit more, we will have this "plateau of perfect happiness." However, if many products and aspects of technology are adding at least one or two components per year, you will see linear growth in new technology but exponential growth in problems and complexity. Most systems have increasing complexity.

"I know of no one to date who is an expert in all the Oracle features. There is RAC, indexing, CBO, etc. Back in version 6, I could pretty much get a handle on things," says Mogens.

"Mainframes are a good example. They are incredibly stable, incredibly predictable, and incredibly solid. They were, of course, replaced by UNIX and Windows machines, which are much more chaotic and unpredictable. Problems equal money, which is good," he laughs. "Consultants are paid by the hour, not by the solution, so I welcome chaos."

Miracle A/S is moving into the SQL Server consulting area as well, Mogens says, "Because if you know how the best database on the market works, you can make all the other databases work, too. You know what needs to be done. That takes us to different parts of the world."

"All the things we've done, presentations we've given, myths we've killed—we can kill them again in the SQL

(continued on page 25)

RATTY SQL

- Part II -

By Iggy Fernandez

If you need a copy of Part I of this article, please send a request to iggy_fernandez@hotmail.com.



Iggy Fernandez

Great rats, small rats, lean rats, brawny rats,
Brown rats, black rats, grey rats, tawny rats,
Grave old plodders, gay young friskers,
Fathers, mothers, uncles, cousins,
Cocking tails and pricking whiskers,
Families by tens and dozens,
Brothers, sisters, husbands, wives—
Followed the Piper for their lives.

—“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.

Recap of Part I

In Part I of this article, we discussed the difference between efficiency (the cost per unit) and scalability. The advantage of a scalable procedure is that the cost per unit goes down as we process larger and larger volumes. In Fig. 1, Procedure A is more efficient than Procedure B at small volumes. However, the unit cost of Procedure A remains constant even at higher volumes, while Procedure B quickly becomes more efficient. In other words, Procedure B is scalable but Procedure A is not.

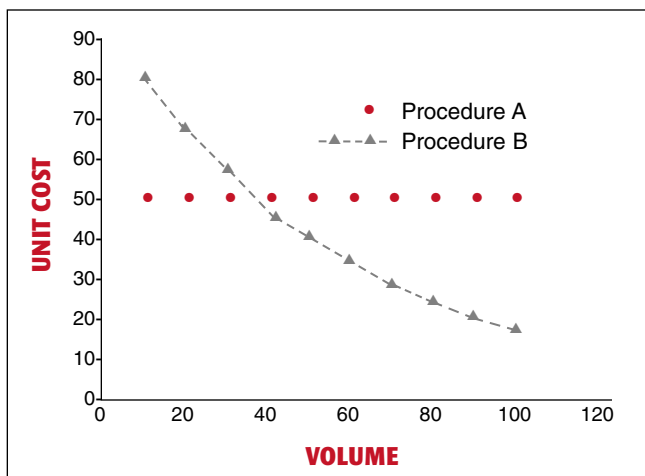


Fig. 1: Scalable Procedures.

We also argued that RATTY (record-at-a-time) processing is less scalable than the equivalent single SQL statement. Table 1 lists the results we obtained from our tests.

ELAPSED TIME PER THOUSAND RECORDS			
Total Number of Records Processed	Single SQL Statement	RATTY Server-Side Stored Procedure	RATTY Client-Side PRO*C Program
1000	0.124	0.414	1.457
2000	0.061	0.174	1.234
3000	0.038	0.148	1.198
4000	0.034	0.130	1.171
5000	0.026	0.121	1.170
10000	0.015	0.106	1.169
20000	0.010	0.102	1.150
30000	0.007	0.108	1.151
40000	0.006	0.094	1.150

Table 1: Test Results.

We left it as an exercise for the reader to explain why a single SQL statement was so much more scalable than the RATTY methods, but the reasons include both the reduced SQL*Net messaging costs as well as the reduced parsing costs. Note that even RATTY methods stand to gain from block caching effects, which explains why even RATTY methods experience some cost reduction as volumes increase.

We closed Part I with the following SQL challenge. The solution is provided at the end of this part.

First create the following 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);
```

(code continued on page 8)

(code continued from page 7)

```
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 challenge is to 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, with the newly added rows highlighted in red:

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

Mail Bag

Kailash Awati from the “land down under” submitted two solutions to the SQL challenge in Part I, one of which we feature at the end of this part:

I'm an Oracle developer/DBA from Sydney, Australia. I came across your enjoyable article (Ratty SQL - Part I, in the NoCOUG Journal) while surfing the Net (I'll definitely be using this term (Ratty) when describing the perils of record at a time processing to some of my colleagues)

Ravi Kulkarni, from Phoenix, Arizona, cautions that single SQL statements may not always outperform RATTY programs because SQL expressions can quickly get convoluted and Oracle may not always do a good job of constructing a query execution plan:

I read your article in the NoCOUG journal with interest. It is well written and probably applies to a lot of cases. However, I think it is a stretch to say that the Oracle optimizer can optimize every possible SQL statement written using convoluted but severely limited syntax, the same way as an expert DBA with a procedural language. I would use SQL or PL/SQL depending upon the complexity of the application.

Recent SQL enhancements

We now turn our attention to the claims that SQL is “not powerful” enough to handle “the most complex problems.” We list some fairly recent SQL enhancements, which can be used in the battle against RATTY SQL. There are many such enhancements, but we only have space for a few.

The Merge statement

The “Merge” statement is a powerful antidote for a great deal of RATTY processing. The problem that is addressed is

that of processing a transaction file. If a corresponding record exists in a master file, then the master record needs to be updated, and if a corresponding record does not exist in the master file, then a new master record must be created. Previously, it was necessary to write a RATTY program to process the records in the transaction file. Nowadays, this can be achieved with a single SQL statement.

The following example (from the Oracle 10g SQL Reference Manual) performs some final adjustments to employee bonuses. Only employees with salaries less than or equal to \$8000 should receive bonuses. Those who have *not* made sales get a bonus of 1% of their salary. Those who *have* made sales are granted an *additional* bonus amount of 1% of their salary.

The statement below performs all three adjustments in a single step:

```
MERGE INTO bonuses D
USING (
  SELECT employee_id, salary, department_id
  FROM employees
  WHERE department_id = 80
) S
ON (D.employee_id = S.employee_id)
WHEN MATCHED THEN
  UPDATE
  SET D.bonus = D.bonus + S.salary * 0.01
  DELETE WHERE (S.salary > 8000)
WHEN NOT MATCHED THEN
  INSERT (D.employee_id, D.bonus)
  VALUES (S.employee_id, S.salary * 0.01)
  WHERE (S.salary <= 8000);
```

Case expressions

“Case” expressions give us the powerful ability to use IF-THEN-ELSE logic in SQL statements. A case expression may be used wherever a scalar value or scalar expression may appear in an SQL statement. They are much more powerful than “decode” expressions because they support Boolean operators such as “greater than,” “less than,” and “not Equal.” A case expression may even be nested within another case expression.

The following example (from the Oracle 10g SQL Reference Manual) finds the average salary of the employees, using \$2000 as the lowest salary possible:

```
SELECT AVG(
  CASE
    WHEN e.salary > 2000
    THEN e.salary
    ELSE 2000
  END
) "Average Salary"
from employees e;
```

SQL for Aggregation in Data Warehouses

Oracle provides the CUBE, ROLLUP and GROUPING SETS extensions to the GROUP BY clause. They are more efficient than the equivalent UNION ALL of differently grouped rows. ROLLUP calculates aggregations at increasing levels of aggregation, while CUBE calculates all possible combinations of aggregations. The GROUPING SETS extension allows flexible control over aggregations. For

more information, refer to Chapter 20: “SQL for Aggregation in Data Warehouses” of the Oracle 10g Data Warehousing Guide.

SQL for Analysis and Reporting

“Analytic” functions are a wonderful extension of “aggregate functions.” The key difference is that they can return a *different* value for each row in a group of rows, instead of just one value for the entire group of rows. They can be used for “top-N” queries, “bottom-N” queries, “middle-N” queries, running totals, moving averages, inter-row calculations, and other equally amazing things. For more information, refer to Chapter 21: “SQL for Analysis and Reporting” of the Oracle 10g Data Warehousing Guide.

The following example (from the Oracle 10g SQL Reference Manual) ranks employees based on their salary and commission:

```
SELECT
  department_id,
  last_name,
  salary,
  commission_pct,
  RANK() OVER (
    PARTITION BY department_id
    ORDER BY salary DESC, commission_pct
  ) "Rank"
FROM employees
WHERE department_id = 80;
```

Another example is provided at the end of this article, where we present solutions to the SQL challenge.



There's No Substitute for Experience

Our team represents some of the most knowledgeable and experienced in the industry. We are authors and speakers with long careers as Oracle experts, averaging 12 years. Our specialty is providing remote DBA services and onsite Oracle database consulting.

We offer a free consultation to discuss:

- Increasing uptime and reliability
- Minimizing downtime and data loss
- Optimizing performance
- Reducing cost of database operations

Call Us Today!

(415) 344-0500 • (888) 648-0500
www.dbspecialists.com

ORACLE | CERTIFIED SOLUTION PARTNER


Database Specialists

Serializability: The Hobgoblin of Little Minds?

In his presentation to NoCOUG last year, Tom Kyte characterized differences in approaches to data consistency and concurrency as the fundamental difference between Oracle and the other database vendors such as Microsoft and IBM. He said that it can be Oracle's best feature or its worst feature (if you don't understand it). He also said that, if you don't understand it, you are probably doing some transactions wrong in your system!

Interested already? Then run, don't walk to Iggy's presentation at our next conference. “Serializability,” or the notion that concurrently executing transactions must not interfere with each other to produce unexpected results, is such a tricky beast that original research is still being published on the topic. Come learn about the issues and stop doing transactions wrong! ▲

SQL for Modeling

The MODEL clause mimics the capabilities of spreadsheet programs by treating tabular data as if it were a multi-dimensional array. Rules can be supplied to create additional rows and columns of data. For more information, refer to Chapter 22: “SQL for Modeling” of the Oracle 10g Data Warehousing Guide.

The following example (from the Oracle 10g Data Warehousing Guide) answers the question:

● In a perfect world, all databases would be created equal. But in reality, it's a jungle out there. Different database products. Different operating systems. And increasing pressure to keep them all under control. Fortunately, there's one way to manage it all: one idea. With Embarcadero's Evolution, Rapid SQL, and the Analytic Index, you'll handle more data with fewer resources. Better access. And instantly across your existing or new multiple database platforms. To learn more, please visit www.embc.com

EMBARCADERO TECHNOLOGIES
WE MAKE DATA WORK™

EMBARCADERO IS
ORACLE PLATFORM DATA MANAGEMENT.

“If sales for each product in each country grew (or declined) at the same monthly rate from November 2000 to December 2000 as they did from October 2000 to November 2000, what would the fourth quarter’s sales be for the whole company and for each country?”:

```
SELECT
  country,
  SUM(sales)
FROM
  (
    SELECT
      product,
      country,
      month,
      sales
    FROM
      sales_view2
    WHERE year = 2000
    AND month IN ('October', 'November')
    MODEL
      PARTITION BY (product, country)
      DIMENSION BY (month)
      MEASURES (sales)
      RULES
        (
          sales['December'] = (sales['November']
          / sales['October']) * sales['November']
        )
  )
GROUP BY GROUPING SETS ((), (country));
```

Note the use of the GROUPING SETS clause to produce multiple levels of aggregation.

Orthogonality

Some recent enhancements have improved the “orthogonality” of the SQL language. The term “orthogonal” comes from two Greek words, “orthos,” meaning “perpendicular,” and “gonia,” meaning “angle.” In Fig. 2, the unit vectors i

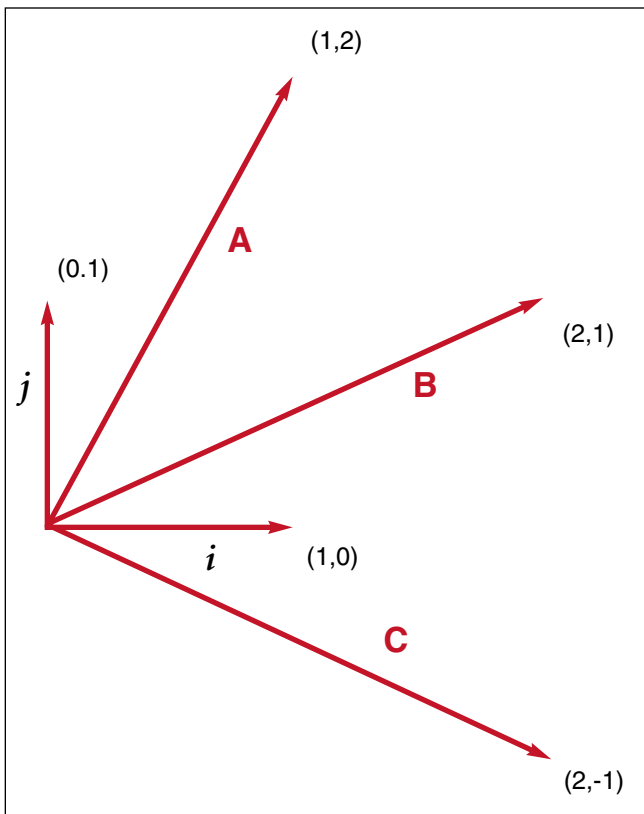


Fig. 2: Orthogonal Vectors.

and j are perpendicular (orthogonal) to each other. Every other vector in the plane of the picture can be constructed by using the unit vectors as little building blocks—e.g., $A = i + j + j$, $B = i + i + j$, and $C = i + i - j$.

In a somewhat analogous fashion, the SQL language has certain “primitive operations,” e.g., “join.” The desirable ideal is that any derived data set can be constructed from the base tables using the primitive operations as the building blocks of a single SQL expression. As a counter example, the “union” operation was not permitted in the definition of a “view” in the early implementations of the SQL language, and this went against orthogonal principles.

We mention two advances made by Oracle in the area of orthogonality.

The first innovation is “in-line views,” which are sub-queries in the FROM clause of an SQL statement. Previously it was necessary to store these sub-query definitions in the database prior to using them in an SQL statement. To complicate matters, ordinary users are not usually permitted to store new definitions in the database.

The second innovation is “scalar sub-query expressions,” which are sub-queries that return exactly one value. They may be placed anywhere a scalar expression is permitted.

The following trivial example combines the use of in-line views and scalar sub-query expressions:

```
SELECT (SELECT 1 FROM dual)
FROM (SELECT 1 FROM dual);
```

Better examples are provided at the end of this article when we present solutions to the SQL challenge.

Extensibility

As if its vast array of scalar functions and aggregate functions were not enough, Oracle offers the ability to create user-defined scalar functions as well as user-defined aggregate functions.

User-defined scalar functions can even incorporate SQL queries and this is another example of orthogonality. The following example searches the *sale* table for the highest *sale_id* for any given *book_id*:

```
CREATE FUNCTION get_max_sale_id (
  new_book_id IN INTEGER
)
RETURN INTEGER
IS
  max_sale_id INTEGER;
BEGIN
  SELECT NVL(MAX(sale_id),0)
  INTO max_sale_id
  FROM sale
  WHERE book_id = new_book_id;

  RETURN max_sale_id;
END;
```

The Oracle-supplied “packages” are a good example of the use of user-defined functions. For example, *dbms_random* is a library of functions that can be used to generate random numbers.

Solution to the SQL challenge

We feature a solution submitted by Kailash Awati from

Sydney, Australia. He cleverly combines a scalar sub-query expression with an analytic function to generate the necessary sale ID:

```
INSERT INTO sale (
  book_id,
  sale_id,
  sale_price
)
SELECT
  book_id,
  (
    SELECT NVL(MAX(sale_id),0)
    FROM sale
    WHERE book_id=new_sale.book_id
  )
  +
  ROW_NUMBER() OVER (
    PARTITION BY book_id
    ORDER BY sale_price desc
  ),
  sale_price
FROM new_sale;
```

As an exercise, try incorporating the `get_max_sale_id` function into the above solution and explain why Oracle generates an error message when the statement is submitted (even though the statement is syntactically correct).

Here is a solution that uses in-line views:

```
INSERT INTO sale (
  book_id,
  sale_id,
  sale_price
)
SELECT
  book_id,
  max_sale_id+new_sale_id,
  sale_price
FROM
  (
    /* compute max_sale_id */
    SELECT
      new_sale.book_id,
      NVL(MAX(sale.sale_id),0) AS max_sale_id
    FROM new_sale LEFT OUTER JOIN sale
    ON new_sale.book_id=sale.book_id
    GROUP BY new_sale.book_id
  )
  NATURAL JOIN
  (
    /* compute new_sale_id */
    SELECT
      book_id,
      ROW_NUMBER() OVER (
        PARTITION BY book_id
        ORDER BY sale_price desc
      ) AS new_sale_id,
      sale_price
    FROM new_sale
  );
```

Baby, I'm amazed!

We end by quoting everybody's favorite Oracle guru, Tom Kyte. Here are some quotes from Tom's website <http://asktom.oracle.com>.

One of Tom's readers suggested that SQL was not powerful enough to handle the most complex problems:

The planning system I supported was very complex due to the nature of the business (microelectronics manufacturing). [...] This is a system that I cannot even imagine being written in SQL. There was too much data manipulation and too many calculations [...]

Here is how Tom replied:

give me a lot of procedural code, let me turn it into set operations - and well, there is a song out there "baby I'm amazed..."

You see, we have to agree to disagree on your point that doing it in sql is crazy hard to do. If you don't feel as comfortable in SQL as you do in C, sure — but.... if you do, then it is just a language after all.

Another reader observed that long SQL expressions were not very readable:

You keep suggesting that if you can do it in sql, do it instead of using procedural (pl/sql) code; and I agree. However, I am wondering whether the readability of code decreases once the select stmt becomes too complex and we start hitting a point of diminishing returns. For example, do you think the following query in the procedure (performance wise it is fine) should be rewritten in procedural code (which would do it step by step and presumably would be more readable)?

Tom made the following astute observation:

i would never rewrite working, fast sql as procedural code. Tell me — how would a procedural implementation of that be "more readable" or understandable. (maybe comments on what it is doing would help — but that would be as true for a procedural implementation wouldn't it)

A third reader did not want to overload the database server:

I have some fairly heavy batch jobs in my programs and I could, of course, do it the Tom way and put a lot of it in the db. But then, on the other hand, Oracle is doing the processing and calculations and will quicker reach the state of high load.

And Tom suggested that the opposite was true:

you will actually find in most cases that the more processing you do in SQL, the less CPU/resources on the server you actually use. Believe it. Or not. The guys who take the data OUT, massage it, put it back IN use more database resources than the guys that just process it in place.

Conclusion

We hope that our readers have a newfound appreciation for the dangers of RATTY processing and for the ability of the SQL language to fight it. Comments on this article or suggestions for future installments of the SQL Corner may be sent to iggy_fernandez@hotmail.com. ▲

Acknowledgements

Thanks to Kailash Awati, Database Architect at Boehringer-Ingelheim in Sydney, Australia, and to Sumit Sengupta, Senior DBA at EDS in Columbus, Ohio, for carefully reading this article and providing helpful comments.

Iggy Fernandez is the lead DBA for a Silicon Valley startup company and is Oracle 10g certified. Previously, he was the manager of database administration for Corio, an application services 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.

Copyright © 2005, Iggy Fernandez

How to Architect and Manage a Project (Part II)

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

Editor's Note: In the last issue of the NoCOUG Journal, Ralph Hughes discussed project management and the Project Architect and Project Manager's roles and resources. In this section, he continues with the starter toolkits needed, along with a wrap-up and reference list.

Project Architect's Starter Toolkit

The toolkit for the Project Architect will require the most thought of the two, to some extent because architecting lags behind project management as a profession, but more so because the subject matter is divergent and still undergoing rapid innovation. We can say "divergent" because software architecture spans an awesome range of applications, from GUI-based front ends to data warehouses to integrated enterprise web services platforms. We cite "rapid innovation" because of its frequent turnover in major paradigms—witness the last decade's speedy progression from client server to web-based Java, then J2EE, later Struts, and now JavaServer Faces technology.

Despite this moving target, there are many fundamentals of the PA profession that remain as the paradigms fly by. Accordingly, the project lead must assemble a PA toolkit that contains both timeless implements and those that may have a shorter shelf life. Beginning with the timeless notions, the project lead should acquire the following:

- A good analytical framework
- A strong software development life cycle (SDLC)
- Proficiency at UML
- Mastery of an RM tool

Focusing on tools that may someday be replaced, a project lead should also acquire the following:

- A good lightweight architectural methodology
- A good full-strength architectural methodology
- A good GUI-app metamodel
- A good data warehouse metamodel
- A good enterprise IS metamodel
- A good collection of design patterns and universal data models

Armed with the PA toolkit outlined above, the project lead will be ready to "hit the ground running" when it comes to designing and communicating an information technology solution.

A Good Analytical Framework

Frameworks are conceptual strategies for relating the

notions constituting the real world to those that describe an information system and its implementation. The classic architectural structure is the Zachman Framework (Zachman 1987), which arose out of IBM in the mid-1980s. In this framework one analyzes both the requirements and the candidate high-level design for a system using a matrix of six viewpoints against five levels of design abstraction. The greatest criticism of the approach is that all thirty cells of the matrix are overkill, leading to analysis paralysis. Accordingly, project leads should scan the full range of these viewpoints and pick only those most appropriate for the project at hand.

The object-oriented world has provided some alternative frameworks that may work better for the Internet-based information environment we have today. The Object Management Group publishes an architectural framework, TOGAF, that includes the Model Driven Architecture architectural methodology (Miller 2003). OMG has mapped its architectural framework to many older frameworks, and therefore it is quite comprehensive without being too complex. For full-strength analysis, the PA should look into the International Organization for Standardization's *Reference Model for Open Distributed Processing* (RM-ODP), which addresses highly concurrent, highly distributed applications. RM-ODP has more detail than PA/PMs will need on 90% of their projects and thus would be best adopted selectively.

One aspect that PAs should try to locate in whatever framework they adopt is RM-ODP's notion of viewing five architectural aspects of any application: business, data, process, platform, and technology. RM-ODP also defines an important hierarchy of architectural "transparencies" that will permit each service layer in a design to operate smoothly with minimal understanding of the other layer's functionality.

A Good SDLC

Given that a PA's focus is on guiding the actual work of building an application, project leads will do well to start with a strong SDLC that they can use as a starter template for the project's WBS. The classic waterfall or spiral methodology sets for an SDLC are divided into success phases of business requirements, requirements analysis, logical design, physical design, coding, testing, deployment, and maintenance. Invariably, IS professionals find this too rigid an approach because one phase often bleeds into the next and teams frequently work on more than one release

for an application at a time. In the mid-1980s, the spiral methodology was developed to address these shortcomings, and it has evolved over the years into Rapid Application Development (RAD), the Rational Unified Process (RUP), and even the Java world's Extreme Programming (XP). PAs seeking a thorough foundation with a minimum of effort might start with Boehm's seminal 1988 paper and then invest in learning RUP (Kruchten 2004) in order to feel at home with the object-oriented nature required for many applications today.

Proficiency with UML

Ninety percent of a PA's work involves models in some way. There are many modeling languages, but the Unified Modeling Language has received the highest degree of international acceptance and third-party tool support. First published in the late 1990s, UML is now in its fifth iteration and has proven means of depicting for a system the user requirements, the static components, and the dynamic behaviors the architect has in mind. It is also easily extensible so that PAs can adapt the language to express the innovative solutions they may need to communicate.

With its seven base diagram types, UML suggests an architectural process that will lead the PA to providing a thorough expression of the application desired (see Schmuller 1999, Chapter 16). For example, UML would guide the first few stages of the waterfall methodology as follows:

- Capture requirements as contracts between developers and users with **Use Cases**.
- Capture the steps and decision points in the supported business processes with **Activity Diagrams**.
- Document the major entities involved in the business domain with a **High-Level Class Diagram**.
- Identify cooperating systems that will form the new system's operating environment with a **Deployment Diagram**.
- Organize the detailed system requirements, such as those uncovered by JAD sessions, with a **Package Diagram**.
- Document the analysis effort with refinements upon the **Use Case and Class Diagrams**.
- During the design phase, depict the details of processing with **Sequence Diagrams**.
- Depict the interfaces between candidate components using **Collaboration Diagrams**.

Mastery of an RM Tool

Research as to why 75% of development projects either fail or suffer serious challenges has pointed repeatedly at poor requirements gathering as a root cause. Much has been written about working requirements until they are "SMART" (Mannion 1995), but this does not address the fact that the volume of requirements that good analysis generates is overwhelming. Luckily there is **requirements management** software available. Two packages that the PA/PM can investigate to become more familiar with RM are ReqPro by Rational and CaliberRM by Borland.

Key to an RM package's value is not only the fact that it serves as a repository of user requirements once expressed, but also due to its breakdown of high-level business requirements into detailed user requirements, and those into system requirements. RM tools will also track the design decisions that are made and keep them linked to the user or system requirement that drives them. With this data loaded, the RM tools can then offer the project lead an invaluable set of benefits:

- Automated impact analysis when a requirement changes
- Automated test script generation
- Automated test results tracking
- A single point of communication for the entire development team and other stakeholders

Shared RM packages give common visibility to the extent that design, development, and testing have been completed, greatly reducing the communication burden placed upon project leads in both their architect and manager roles.

Two Good Architectural Methodologies

Project leads will need two architectural methodologies in their careers: a lightweight one and one that can go as deep as needed. The Software Engineering Institute's Architecture Based Design (ABD) methodology serves very well for the first approach—it is lightweight in that it can be memorized and deployed easily, yet its capabilities are sufficiently extensive. ABD involves a repeated five-step decomposition process where the "system" idea is continually subdivided into progressively finer design components and the system requirements carefully assigned among them. The decomposition and component refinement continues until all components can be assigned to as few "component archetypes" as possible, and the requirements of these archetypes are detailed enough that a software engineer can take over and begin assigning each to a named class (in the object-oriented sense), which can be specified in detail.

For more complex projects, the Object Management Group offers its Model Driven Architecture (MDA), a standards-based approach to model-based system development. MDA expands upon ABD's objectives with emphases upon reusability and maintainability of the architectural components through an application's life cycle (Miller 2003).

Three Sets of Metamodels

Depending upon the type of projects in which they specialize, project leads will require one to three "metamodels," which will guide the breadth and depth of the actual models of envisioned systems they must deliver as architects. A metamodel lists the typical components that will be found in most models and also suggest how those components will interrelate. By using metamodels, PA/PMs still ensure that they have thought of everything and understand how the long list of suggested pieces will fit together.

For data capture applications, the project lead will require a graphical user interface (GUI) metamodel. GUIs

have seen tremendous innovation in the past decade and will probably continue to do so for the indeterminate future. For now, Java is the GUI technology making the greatest strides, so PA/PMs will do best to adopt a GUI metamodel from the Java world because it will keep them current with the “standard” means of building a user interface, even if their projects actually use a different technology.

In the Java world, J2EE is a heavyweight approach for applications spanning the enterprise and would probably give the PA/PM too much to learn for simple metamodeling purposes. A less-demanding Java metamodel can be found in JavaServer Faces technology, the recent successor of Struts. Documentation on this metamodel can be easily found by starting with the Sun Java website (<http://java.sun.com/j2ee/javaserverfaces>). Data warehouses are the other bookend to data capture systems, and luckily a definition of their standard architecture is well under way as part of OMG’s “common warehouse metamodel” (CWM) (Poole 2002 and OMG 2001). CWM is based upon the overarching need for data warehouse integration, a goal that necessarily forces the metamodel to be comprehensive in the pieces it depicts for any possible warehouse; otherwise it would leave entire areas of functionality with no means to be exchanged between applications. This foundation plus a base notion of an “information supply chain” (ISC) lets CWM also serve as a starter metamodel for enterprise system integration, the third domain that PA/PMs might want to have preloaded in their minds.

When venturing into multi-application or multi-business unit projects where ISC begins to wear thin, PA/PMs can turn to the Common Object Request Broker Architecture (CORBA), also from OMG, a twenty-year-old framework devised to address in detail how enterprise application integration should be pursued (see Baker 2003 and Wales 2003).

All of the metamodels discussed above nicely circle back to underscore an earlier suggestion: they are all depicted using UML; thus even PA/PMs who believe their work will remain entirely in the relational database world and never venture into object-oriented concepts would be wise to invest in learning the OO paradigm, if for no other reason than to be comfortable with UML and for assistance in acquiring the head start offered by the metamodels recommended here.

Design Patterns and Universal Data Models

According to Nicolas Wirth, the creator of Pascal, “data plus algorithms equals programs,” a notion that suggests that the PA/PM will need at least a moderate capability in expressing both process and data designs. Luckily this can be acquired by a straightforward method: The process side of this duo is best addressed by **design patterns** (Gamma 1995), which represent a standard notion of how software engineers typically realize common functionality required by data capture systems. The data side is explicitly covered by the **universal data model** (UDM) effort (Hay 1996 and Silverston 2001), where the relational diagrams and data dictionaries relational data structures that data architects

repeatedly reinvent have been gathered, so as to save us all a tremendous amount of work. The UDM authors have gone so far as to document the specific, large-scale data structures required for more than a dozen industries, from telecommunications to pharmaceuticals (e.g., see Hughes 2003). It probably goes without saying that every PA/PM should be familiar with the UDMs for their industry at least, else they condemn themselves to a slow start on their project designs and undoubtedly a few major conceptual omissions.

Project Manager’s Starter Toolkit

Because of the greater maturity of the PM profession, the project management toolkit will be considerably easier for the project lead to amass than that of the PA, and its components will remain relatively stable for decades to come. The most important PM notions for the project lead to acquire are as follows:

- Proper project initiation results in a project charter.
- Requirements, requirements, requirements—that’s SMART.
- PMs drill down three layers of estimates before committing to a date/cost/scope.
- All estimates must have a defensible basis.
- An MS Project document is not a plan—you must have BARE.
- Unbeknownst to all, PMs focus on risk.

Project Charters

A project charter is a document that formally authorizes a project. Project charters are derived from the entire project initialization process, which contains several crucial deliverables or findings, such as (1) project objectives; (2) high-level deliverables, estimates, constraints, assumptions, and resource requirements; (3) statement of business need; (4) high-level description of the desired product or result; and (5) definition of the responsibilities and authority of the project manager.

Hopefully it is obvious that project leads would be reckless to undertake the management of a project without any one of the essential items listed above. Without any one of the above, there are simply too many unknowns to be assured of project success, and it is a measure of a project lead’s professionalism and savvy that she insists on a project charter from management before she will allow anyone to call her the project’s manager.

SMART Requirements

Of the projects surveyed for the Standish *Chaos Report* (Standish 1995), 25% failed because of lack of requirements and user involvement. Though PAs need good requirements to develop a design, requirements are so crucial that PMs should consider themselves ethically bound not to proceed with a project until the requirements upon which work will be based are comprehensive and thorough. In the mid-1990s, the Association for Computing Machinery promoted “SMART Requirements” (Mannion 1995), which provide easy-to-employ guidelines to crafting good require-

ments, suggesting that PMs rework requirements until they are Specific, Measurable, Attainable, Realizable, and Time bounded. Though “attainable” has been often re-expressed during the last decade as “action oriented,” and “realizable” is often recast as “realistic,” PMs should be prepared to deploy one of the SMART flavors so that their teams will achieve the highest quality in this crucial area of work.

Three-Tiered Estimating

Nothing stresses a project team more than too tight a delivery schedule, yet PMs regularly commit themselves to unrealistic deadlines simply because management wants them to quote a date, weeks before the team has the information necessary to properly gauge how much time, money, and resources will be needed. Once quoted, this poorly informed choice of deadlines becomes a noose with which PMs hang themselves and their project teams. Because it is not realistic to refuse to give management a target date when asked, PMs need to instead learn to qualify the estimates they provide. PMI suggests using a graduated series of three estimates, as follows:

- Order-of-magnitude estimate: made during project initiation and ranges from -25% to +75%
- Budgeting estimate: made during detailed project planning and ranges from -10% to +25%
- Definitive estimate: made at the conclusion of the planning phase and ranges between only -5% and +10%

The tight range of the definitive estimate demands a

very high level of certainty from PMs and their teams, requiring them to really know the system they propose to implement and forcing them to complete a thorough analysis before verbalizing a target date. If acquiring the information needed extends the project, this must be a choice for which the PM fights. To commit the team to anything less than a definitive estimate is unfair to all stakeholders and might even be considered a form of deception.

To avoid “schedule suicide,” PMs must build the delivery of budgeting and definitive estimates into the project plan and regularly review that plan management, constantly reminding stakeholders that until the definitive estimate milestone has been reached, *everyone* is working within a very fuzzy guess as to delivery dates and overall expense.

Basis of Estimate

Tightly interwoven with the three-tiered estimating are estimating techniques. When PMs deliver any one of the three types of estimate, they should be able to explain unambiguously from whence the numbers were derived. If the PM cannot quote this “basis of estimate” for projected time and costs, then the projection is not an estimate but rather a SWAG—a silly, wild-^{ss} guess—and would be better off left unsaid.

The PMI has several estimating methods to recommend, including parametric and analogous estimating, but absolutely the worst to employ is the PM’s expert judgment, because the PM is the coordinator, not the person doing the work. For the sake of accuracy as well as fairness, estimates must come from the team that will be doing the

Quovera

Oracle Consulting Solutions Specialists


With over 10 years of Oracle consulting excellence, Quovera has performed more than 300 engagements for over 200 clients. Quovera’s specialized Oracle consulting services include:

- ◆ 11i E-Business Suite implementations and upgrades
- ◆ Full lifecycle custom and web development and data warehousing
- ◆ Business intelligence, reporting and business performance management

Quovera also provides expert mentoring and training on JDeveloper, Developer and Designer with its team of renowned Oracle Press Authors and Certified Masters.

Quovera brings the experience of its Oracle consulting staff to work with you to improve your Information Technology processes and achieve your business goals.

Quovera, Inc.
800 West El Camino Real, Suite 100
Mountain View, CA 94040
Tel. (650) 962-6319 · www.quovera.com



Query performance slowing you down?

RightOrder QueryEdge: Accelerating the Performance of Resource-Intensive Queries by 10-50x

For more information, go to www.rightorder.com, call 408.983.4000, or stop by our table at the next NoCOUG event.

For the latest analyst whitepaper on improving query performance visit: <http://www.rightorder.com/doc/products/library>

www.rightorder.com | **RightOrder**

work. A multi-pass Delphi technique will yield a range of numbers that the PM can use to build a series of accurately ranging estimates for the project, and the PMBOK provides the proper formula for combining the mean values, given their variances.

Embracing BARF

As important as solid project charters, thorough requirements, and accurate estimates are for project success, they reflect only the potential of the project. To be assembled into a true project plan, their information and implications must be properly processed within the sponsoring organization. PMs should proceed with nothing less than a BARF plan, where the acronym stands for **B**ought into, **A**ccurate, **R**ealistic, and **F**ormal. “Accurate” and “realistic” are products of the team’s research and planning, as described above. It is the PM who makes the plan “formal” and “bought into” by circulating it among all stakeholders and getting their endorsement in writing that the plan meets their needs. Without this endorsement, the PM places the project and his entire team at the mercy of a moving target. Too many contingencies arise during the life of a project for all stakeholders to accurately remember the limited scopes to which PMs committed their teams. A written and widely endorsed project plan will give PMs an unmoving reference against which their efforts will be measured. Veteran PMs (usually with an excess of gray hair) commonly lament the many times the goal their team accepted versus the goal the stakeholder later expected were not the same, so that ironically the team met its obligation but was still considered a failure.

Managing Risks

Risks are events that can impact any aspect of the project’s “triple constraint” and for which there is uncertainty about whether they will occur. A thorough risk inventory, which will typically run into hundreds of items, is a necessary part of an accurate and realistic project plan, as is a list of responses the team will take if the risks occur. During project planning, all risks should receive a qualitative screening so that responses to high-impact and high-probability threats to the project can be thoroughly prepared. Veteran PMs do not spend status meetings listening to team members drone on about what they’ve done, but instead question each about whether anticipated risks have transpired and whether any new risks have materialized.

Gathering Further Expertise in Both PA and PM

Given the many essential strategies for each role presented above, it should be understandable now that an MS Project document is not a project plan, and an “architecture” for a new system is more than an MS PowerPoint presentation. The lists above were necessarily cursory, given the small space available for this article. There is far more depth and interconnectiveness between the elements of both disciplines, and accordingly the third major exhortation of this paper is for project leads to gather true expertise in both the world of software architecture and the world of project management.



Pioneering the Field of Capacity Optimized Storage

The Data Domain DD200 Restorer:

A disk-based recovery storage appliance designed for Oracle database backups:

- ▶ Works with leading backup software vendors and RMAN
- ▶ Patented Capacity Optimized Storage technology that enables long term data protection at low cost
- ▶ Data Invulnerability Architecture ensures recoverability of backup data
- ▶ Rapid disaster recovery via cost effective site-to-site data replication



Data Domain®
Next Generation Data Protection

www.datadomain.com
1.877.622.2587



You've got the skill now get the tool

DBFlash™ for Oracle helps you:

Gain visibility of Oracle Wait-Events

Accelerate Oracle performance by more than 100%

Cut costs and find problems before they find you

www.confio.com
info@confio.com
303.938.8282

CONFIO 
SOFTWARE
Impact IT Performance™

Wrap-Up

Hopefully the strategies and tactics surveyed in this paper will be the project lead's first step toward a personally tailored PA/PM practice that inherently rules out the all-too-common death march. We cannot change the larger forces that cause management to ask one person to simultaneously perform the two full-time roles of Project Architect and Project Manager, but through careful consideration of how these roles are distinct and what each requires, we should be able to greatly increase our performance and our success rates.

As a closing observation for present or future managers who will someday be forced to assign PA/PM duties to a single staff member, let us ask which would be more effective: elevating a technologist into the project lead role or giving a practicing PM responsibility for system design and specification. The thirty-year head start that project management has over software architecture argues for the former. As long as senior technologists have decent people skills, sending them to a PM boot camp to give them a thorough survey of project management tools and techniques should yield fairly good results. Because technical knowledge is far more arcane and detailed, we can bet that it takes far longer to acquire even entry-level insights into proper system architecture than into proper project planning.

One person *can* both manage and architect an application at the same time, but as we have seen above, it requires compartmentalizing PA work and PM concerns into separate areas of consciousness, that is, cultivating a well-organized, schizophrenic mind. A starter for such organization effort is outlined in this paper. All the project lead needs to succeed now is a lot of hard work. ▲

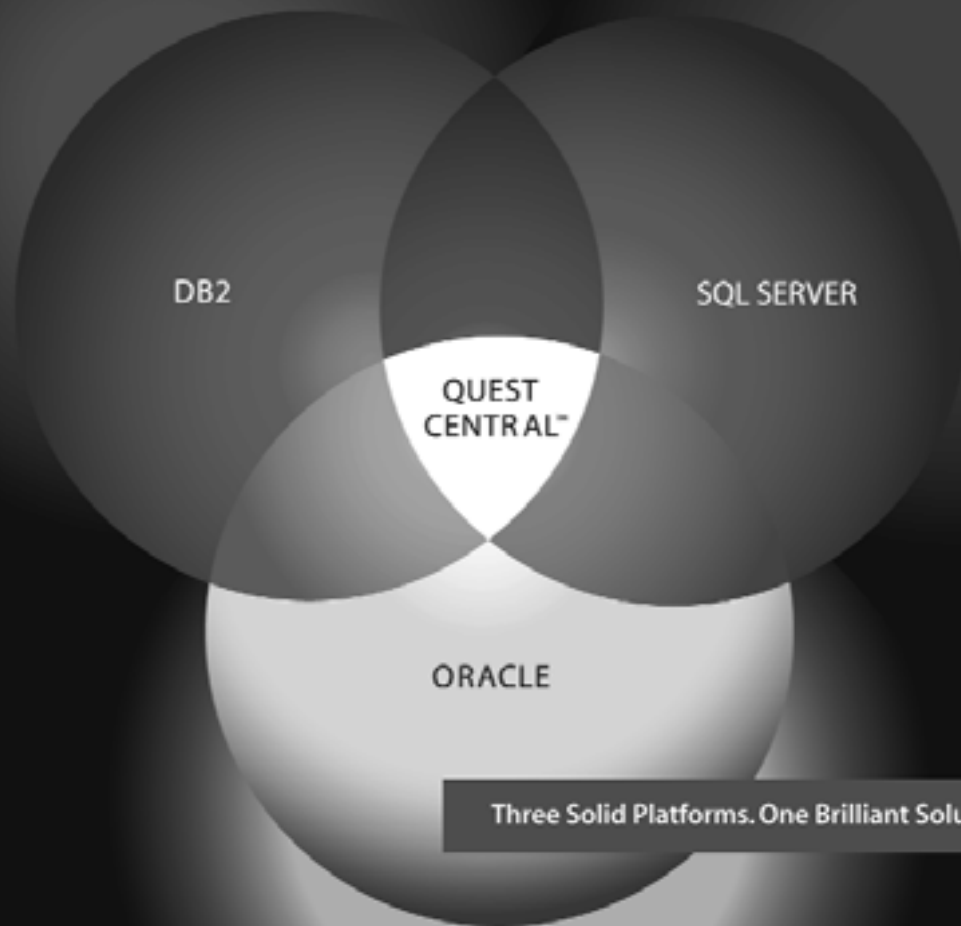
Ralph Hughes, M.A., PMP, serves as Managing Architect for Ceregenics, Inc., a data warehouse consulting association in Denver, Colorado. He can be reached at ralph.hughes@ceregenics.com or at 720-951-2100.

References

Note: All electronic sources cited were retrieved during December 2004.

- [Assmann 2004] Uwe Assmann. No date. *Design Patterns*. [Online] <http://www.ida.liu.se/~TDDb61/Teori/Slides/tddb61-8-design-patterns.pdf>.
- [Bachmann 2000] Felix Bachmann, Len Bass, Gary Chastek, et al. January 2000. *The Architecture Based Design Method*. (CMU/SEI-2000-TR-001.) [Online] Source site home page: Carnegie Mellon Software Engineering Institute (<http://www.cmu.sei.edu>).
- [Baker 2003] Sean Baker. 2003. *CORBA in 2003*. Presentation before CORBA in Action, Dallas, 2003. [Online] http://www.omg.org/corba-corner/03-Iona_Presentation.pdf.
- [Blevins 2004] Terence Blevins, John Spencer, Fred Waskiewicz. July 2004. *TOGAF ADM and MDA: The Power of Synergy*. [Online] Source site section: <http://www.opengroup.org/architecture/wp>.
- [Boehm 2000] Barry Boehm, Wilfred Hansen, editors. July 2000. *Spiral Development: Experience, Principles, and Refinements*. (CMU/SEI-2000-SR-008.) [Online] Source site home page: Carnegie Mellon Software Engineering Institute (<http://www.cmu.sei.edu>).
- [Brooks 1995] Frederick Brooks. 1995. *The Mythical Man-Month: Essays on Software Engineering*, 20th anniversary edition. (First edition published 1975.) Boston, MA: Addison-Wesley. ISBN 0-201-83595-9.
- [Gamma 1995] Erich Gamma, Richard Helm, Ralph Johnson, Jon Vlissides. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley. ISBN 0-201-63361-2.
- [Geary 2004] David Geary, Cay Horstmann. 2004. *Core JavaServer Faces*. Santa Clara, CA: Sun Microsystems Press. ISBN 0-13-146305-5.
- [Goyal 2004] Deepak Goyal, Vikas Varma. No date. *Introduction to JavaServer Faces (JSF)*. [Online] <http://web.princeton.edu/sites/isapps/jasig/2004summerWestminster/Presentations/java%20server%20faces.pdf>.
- [Hay 1996] David Hay. 1996. *Data Model Patterns: Conventions of Thought*. New York, NY: Dorset House Publishing. ISBN 0-932633-29-3.
- [Heldman 2004] Kim Heldman. 2004. *Project Management Professional Study Guide*, 2nd edition. San Francisco: Sybex. ISBN 0-7821-4323-7.
- [Hughes 2003] Ralph Hughes. February 2003. *Toward a Universal Data Model for Pharmaceutical Clinical Trial Data*. [Online] <http://www.datawarehouse.com/article/?articleId=3129>.
- [Kruchten 2004] Philippe Kruchten. 2004. *The Rational Unified Process: An Introduction*, 3rd edition. New York, NY: Addison-Wesley. ISBN 0-321-19770-4.
- [Malveau 2001] Raphael Malveau, Thomas Mowbray, Ph.D. 2001. *Software Architect Bootcamp*. Upper Saddle River, NJ: Prentice Hall PTR. ISBN 0-13-027407-0.
- [Mannion 1995] Mike Mannion, Barry Keepence. April 1995. *SMART Requirements*. Association for Computing Machinery SIG-SOFT, *SE Notes* Vol. 20, No. 2, April 1995.
- [Miller 2003] Joaquin Miller, Jishnu Mukerji, editors. June 2003. *MDA Guide Version 1.0.1*. Object Management Group. (Document no. omg/2003-06-01.) [Online] <http://www.omg.org/docs/omg/03-06-01.pdf>.
- [Mulcahy 2002] Rita Mulcahy. 2002. *PMP Exam Prep*. Minneapolis, MN: RMC Project Management, Inc. ISBN 0-9711647-3-8. [Online] <http://www.rmcpj.com>.
- [OMG 2001] Object Management Group. February 2001. *Common Warehouse Metamodel (CWM) Specification*. [Online] <http://www.omg.org/cgi-bin/doc?ad/2001-02-01>.
- [PMI 2000] *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*. 2000. Newtown Square, PA: Project Management Institute. ISBN 1-880410-23-0.
- [Poole 2002] John Poole, Dan Chang, Douglas Tolbert, David Mellor. April 2002. *Common Warehouse Metamodel*. New York, NY: John Wiley & Sons. ISBN: 0-471-20052-2.
- [Schmuller 1999] Joseph Schmuller. 1999. *Teach Yourself UML in 24 Hours*. Indianapolis, IN: SAMS Publishing. ISBN 0-672-31636-6.
- [Silverston 2001] Len Silverston. 2001. *The Data Model Resource Book*, volumes 1 and 2, revised edition. New York, NY: John Wiley & Sons. ISBNs 0-471-38023-7 and 0-471-35348-5.
- [Standish 1995] Standish Group International. 1995. *The Chaos Report*. [Online] http://www3.uta.edu/faculty/reyes/teaching/general_presentations/chaos1994.pdf.
- [Standish 1999] Standish Group International. 1999. *Chaos: A Recipe for Success*. [Online] http://www.standishgroup.com/sample_research/PDFpages/chaos1999.pdf.
- [Wales 2003] Charlotte Wales, Fred Waskiewicz. September 2003. *The CORBA State of the Union in Application Development Trends*. [Online] <http://www.adtmag.com/article.asp?id=8137>.
- [Yourdon 1999] Edward Yourdon. 1999. *Death March*. Upper Saddle River, NJ: Prentice Hall PTR. ISBN 0-13-014659-5.
- [Zachman 1987] J. A. Zachman. 1987. *A Framework for Information Systems Architecture*. *IBM Systems Journal*, Vol. 26, No. 3. [Online] <http://www.research.ibm.com/journal/sj/263/ibmsj2603E.pdf>.

(APPLICATION CONFIDENCE)



Three Solid Platforms. One Brilliant Solution.

You work in a heterogeneous environment. Your database management software should, too. With Quest Central you can manage more critical databases with astonishing ease — regardless of platform. It delivers comprehensive production management from administration to diagnostics, analysis, space management, SQL tuning and more — all with built-in expertise and a single intuitive, console-based interface. Try Quest Central. It's not just for Oracle anymore. Download "Surviving In a Multi-database Environment," at www.quest.com/nocoug and find out how Quest Central changes database management forever.



© 2004 Quest Software Inc., Irvine, CA. 916.181.9491/54.8000 Fax: 949./54.8999

QUEST CENTRAL™: MANAGE MORE DATABASES TODAY

Optimizing Oracle 10g on Linux Using Automated Storage Management

By Bert Scalzo

10g—Exciting and Useful

I often react to new Oracle releases like Steve Martin acted in the movie *The Jerk* when the new phone books arrived. I cannot help but get excited about all the new technology that Oracle routinely delivers. But with the release of Oracle 10g, that enthusiasm cannot be overstated. And while this Oracle version delivers numerous nifty new features, we'll examine just one, Automatic Storage Management (ASM), which makes Linux database disk space management a snap.

With today's explosive database size requirements and the proliferation of SAN (Storage Area Network) and NAS (Network Attached Storage) disk technologies, both the system administrator and the DBA must manage hundreds to thousands of disks—sometimes for just one database! Thus the task of planning, initializing, allocating, managing, and tuning so many disks becomes unwieldy. Capitulation is often inevitable. Thus many shops simply treat the disk storage farm as a black box, thus abstracting that complexity away from the database. The phrases “You don't need to know” and “Just trust the hardware to handle it” are often given as justification. Frequently this black box approach can lead to database IO bottlenecks that are time consuming to diagnose and remedy. For those few shops where the DBA does still in fact manage all those disks, the task consumes far too much precious time. Neither scenario should be generally acceptable, as both represent the extremes.

Oracle 10g's new Automatic Storage Management directly addresses these real-world scenarios by providing an effective and simple middle ground solution. Now instead of spending inordinate amounts of time managing disk complexity or abstracting it away from the database in the hope that all is well, we can simply permit Oracle 10g's ASM to manage it all. We can now merely allocate disks to Oracle with preferences for striping and mirroring stated via templates, and let ASM manage the space—thus eliminating the need for traditional tools used to manage lots of disk space: a Logical Volume Manager (LVM), filesystems, and the numerous

commands necessary to manage both. Thus Linux database servers can now be deployed more easily and quickly for ever-growing database size requirements—and with far greater IO efficiency than before.

In this article, we'll be examining the detailed differences on Linux of using an LVM with cooked files versus Oracle 10g's ASM. For a more general ASM overview, I recommend reading the Oracle Technology Network (OTN) article titled *Storage on Automatic* by Lannes Morris-Murphy at http://www.oracle.com/technology/oramag/webcolumns/2003/techarticles/murphy_asm.html.

Traditional Methods

To begin, let's review how DBAs have historically allocated disk space to Oracle. We do this for two reasons—first and foremost, to ensure that we clearly see just how much simpler ASM makes the diagram. That simplicity will expose itself as “cleaner pictures,” which translates to fewer steps to set up and fewer commands to manage. And second, we'll examine some scenarios where each option might make sense. This will help us to see how the newer technologies and all their options fit into the overall database storage picture. Figure 1 shows the traditional options.

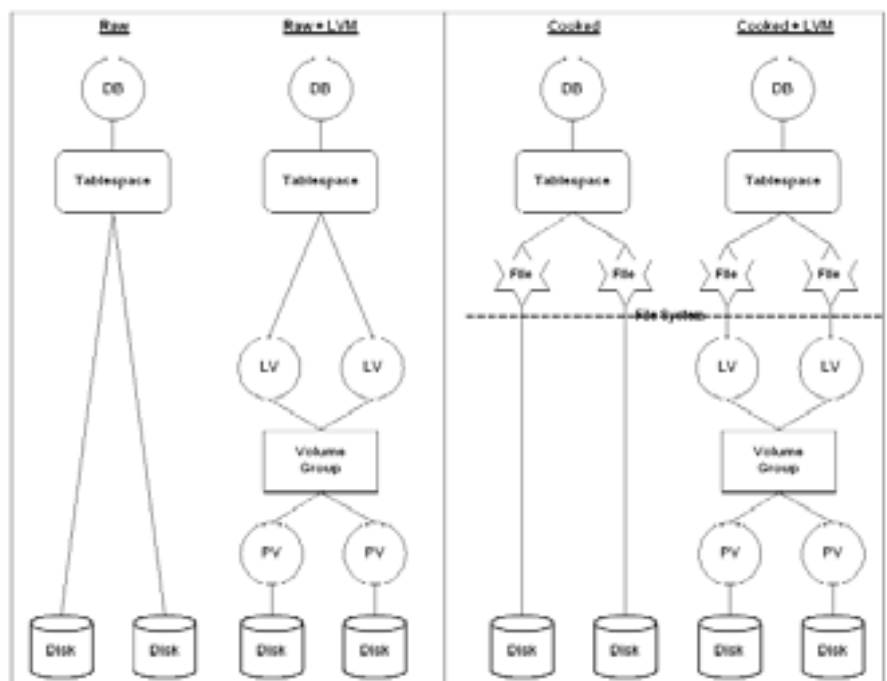


Figure 1.

Figure 1 shows that there were essentially two choices to be made: raw versus cooked and with or without an LVM. These pictures could further be complicated, since what's labeled a disk might in fact be a pseudo-disk. For example, in an EMC disk array, disks are referred to as spindles and can be subdivided into hyper-disks. And in Linux, what we label a disk is simply the device reference—which might be just a partition of an actual disk. To keep things manageable and thus discussable, let's assume that a disk is not to be subdivided. So each disk in the diagram means a separate physical disk drive.

Returning to our original challenge (ever bigger databases with lots and lots of disks), we often have had to choose an option from Figure 1 that more easily and best scales to hundreds or even thousands of disks. With such large numbers of disks, the general consensus has been to use an LVM. And while on some operating systems raw devices are an attractive choice, I've found that raw devices on Linux are not the best way to go (for complete details, see my OTN article *Tuning an Oracle8i Database Running Linux*). Plus, of course, there are the extra management headaches associated with using raw devices. So we can conclude that a cooked filesystem with an LVM (at right in Figure 1) is really our best choice—and therefore the scenario we'll compare with Oracle 10g's ASM.

The New Methods

Now let's look at Automatic Storage Management in more detail. Figure 2 shows both 9i's Oracle Managed Files (OMF) and 10g's ASM. As with many powerful Oracle features, there were incremental steps in delivering ASM—namely 9i's OMF, although we did not know it at the time. Hence I show these two technologies side by side.

Figure 2.

In many respects, OMF was a first step toward internalizing within the Oracle database engine the concept of



Figure 2.

automatic file management. However, even as just an incremental offering, it had some real usefulness. Those DBAs who were abstracting disk complexity away from the database (i.e., the black box extreme scenario from my introduction) merely had to provide Oracle with a single file destination, via the `DB_CREATE_FILE_DEST` parameter. That parameter value can represent only a single filesystem-based storage location—although it could be either a single pseudo-disk representing multiple physical disks hidden behind hardware RAID or a single logical volume combining numerous physical disks. Either way, Oracle 9i automatically manages all the filesystem files. The main disadvantages (besides the obvious single storage location) are that it does not work with raw devices and it still requires using native operating system based filesystems. Even the Oracle manuals suggest that OMF is limited in the following contexts:

- Databases that are supported by the following:
 - An LVM that supports striping/RAID and dynamically extensible logical volumes
 - A filesystem that provides large, extensible files
- **Low-end or test databases** (*which I think makes a statement in itself*)

Now look again at Figure 2 and concentrate on ASM (example at right). There is no need for either an LVM or filesystems. There are simply disk groups under ASM management and available for tablespace consumption. And while the file management is again automatic as with OMF, it is now 100% internal. There are no filesystems to mount via the `/etc/fstab` file and no files to access using operating system file commands like `ls`, `cp`, `tar`, etc. You now query the database to see all such information.

Furthermore, ASM can stripe or mirror your data at the file level—versus by disk with other methods. And Oracle automatically keeps everything fully mirrored or striped for you whenever disks within the disk groups are added, are dropped, or fail—and all with the database completely online. Oracle accomplishes this via new and highly efficient methods that are radically different from the simple arithmetic algorithms used in many LVMs. And although you might make the point that there's an additional ASM Oracle instance required, the LVM-versus-ASM example developed below will show that it's tiny, it's very easy to set up, and it permits 100% effective, efficient, and automatic storage management—and it can do so for all your database instances. So ASM will be shown to be easier to create and manage, as well as providing all the performance expected. You can't lose with ASM.

Easier to Set Up

ASM is just plain easier to set up than its filesystem and LVM counter-

**Don't miss NoCOUG's
Summer Conference:
Thursday, August 18!
ChevronTexaco
San Ramon, CA**

part. Let's assume we have a simple database to create—just the required tablespaces and one for user data. Let's also assume the following hardware and software constraints:

- RAID 0—stripe everything across all drives
 - Stripe width = 4
 - Stripe length = 64 K
- Four 20 GB IDE disks, each with a single partition
- Linux ext3 filesystem (2 GB file size limit)
- Five tablespaces:
 - SYSTEM 2 GB 1 data files
 - SYSAUX 2 GB 1 data files
 - UNDO 8 GB 4 data files
 - TEMP 8 GB 4 data files
 - USER 60 GB 30 data files
- One Physical Volume (PV) per disk drive
- One Volume Group (VG)—VG01
- Four Logical Volumes (LV):
 - LV01 4 GB SYSTEM and SYSAUX
 - LV02 8 GB UNDO
 - LV03 8 GB TEMP
 - LV04 60 GB USER

Figure 3 (above) shows the relative complexities of each environment. And remember, this was a simple case for just four disk drives. Now imagine doing this with thousands of drives!

Now let's compare the actual steps to create the database in each environment. Note that in the LVM example I am purposefully avoiding the handling of various overhead issues with all size settings in order to keep the example simple. But be warned that for optimal disk space usage, you'll have to tackle this issue as well. It's yet another example of complex storage planning and management issues that you can avoid by using ASM.

Here are the steps using cooked files with an LVM (note that the IDE disk drives are assumed to be the second through fifth drives in the example, hence the b–e device name designations of /dev/hdb through /dev/hde):

1. fdisk /dev/hdb set its type to 0x8e (LVM partition)
2. fdisk /dev/hdc set its type to 0x8e (LVM partition)
3. fdisk /dev/hdd set its type to 0x8e (LVM partition)
4. fdisk /dev/hde set its type to 0x8e (LVM partition)
5. pvcreate /dev/hdb /dev/hdc /dev/hdd /dev/hde
6. vgcreate VG01 /dev/hdb /dev/hdc /dev/hdd /dev/hde
7. lvcreate -L 4 G -i 4 -I 64 -n LV01 VG01

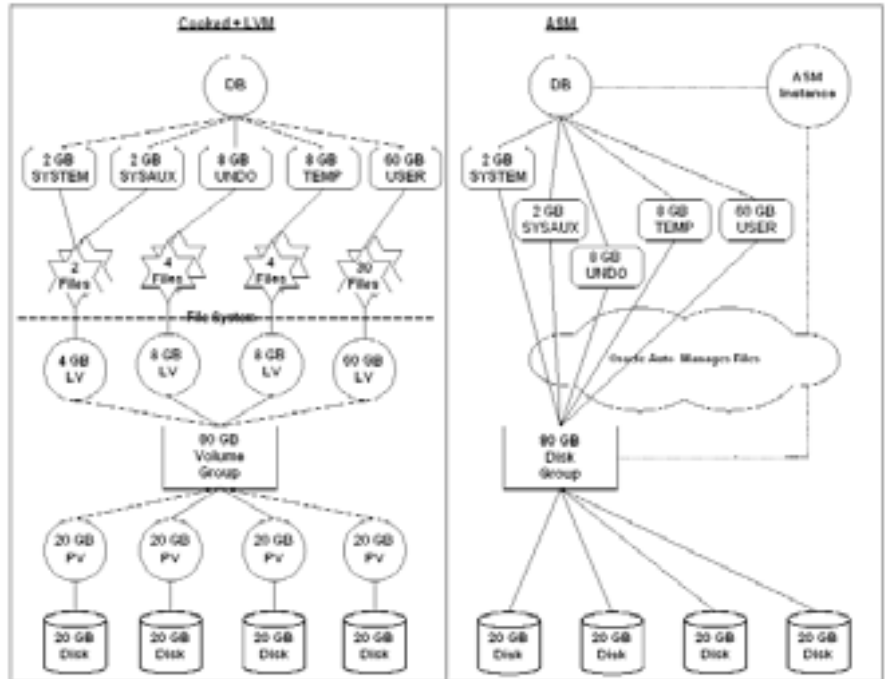


Figure 3.

8. lvcreate -L 8 G -i 4 -I 64 -n LV02 VG01
9. lvcreate -L 8 G -i 4 -I 64 -n LV03 VG01
10. lvcreate -L 60 G -i 4 -I 64 -n LV04 VG01
11. mkfs -t ext3 /dev/VG01/LV01
12. mkfs -t ext3 /dev/VG01/LV02
13. mkfs -t ext3 /dev/VG01/LV03
14. mkfs -t ext3 /dev/VG01/LV04
15. mount /dev/VG01/LV01
/home/oracle/oradata/LVMDB/system
16. mount /dev/VG01/LV02
/home/oracle/oradata/LVMDB/undo
17. mount /dev/VG01/LV03
/home/oracle/oradata/LVMDB/temp
18. mount /dev/VG01/LV04
/home/oracle/oradata/LVMDB/user1
19. edit /etc/fstab and add the new mount point entries
20. Create initLVMDB.ora file:


```

      INSTANCE_TYPE = RDBMS
      
```
21. SQL Plus connect as SYSDBA for SID=LVMDB
22. STARTUP NOMOUNT PFILE=initLVMDB.ora
23. CREATE SPFILE FROM PFILE=initLVM.ora
24. Create Oracle database and user tablespace using SQL code below:

```

create database LVMDB
controlfile reuse
logfile '/home/oracle/oradata/LVMDB/redo_log01.dbf' size 16M,
'/home/oracle/oradata/LVMDB/redo_log02.dbf' size 16M
datafile '/home/oracle/oradata/LVMDB/system/system01.dbf' size 2 G
sysaux datafile '/home/oracle/oradata/LVMDB/system/sysaux01.dbf' size 2 G
default temporary tablespace temp
tempfile '/home/oracle/oradata/LVMDB/temp/temp01.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/temp/temp02.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/temp/temp03.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/temp/temp04.dbf' size 2 G
extent management local uniform size 64k
undo tablespace undo
datafile '/home/oracle/oradata/LVMDB/undo/undo01.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/undo/undo02.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/undo/undo03.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/undo/undo04.dbf' size 2 G;
  
```

(code continued on page 22)

(code continued from page 21)

```
create tablespace USER
datafile '/home/oracle/oradata/LVMDB/user1/user01.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user02.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user03.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user04.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user05.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user06.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user07.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user08.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user09.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user10.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user11.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user12.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user13.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user14.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user15.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user16.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user17.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user18.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user19.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user20.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user21.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user22.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user23.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user24.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user25.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user26.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user27.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user28.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user29.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user1/user30.dbf' size 2 G
extent management local uniform size 64k;
```

Now simply imagine that you'll need dozens of such user tablespaces, each with lots and lots of data files. It's not hard to see why this example does not scale well. Keeping track of all the tablespaces, data files, logical volumes, physical volumes, and actual disk drives (for finding hot spots) is just too darn hard for anyone but Superman.

Now here's the much simpler ASM example:

1. Create initASM.ora file:
 - INSTANCE_TYPE = OSM
2. SQL Plus connect as SYSDBA for SID=ASM
3. STARTUP NOMOUNT PFILE=initASM.ora
4. CREATE SPFILE FROM PFILE=initASM.ora
5. CREATE DISKGROUP dgroup1 EXTERNAL REDUNDANCY DISK '/dev/hdb';/dev/hdc;/dev/hdd;/dev/hde'
6. Create initASMDB.ora file:
 - INSTANCE_TYPE = RDBMS
 - DB_CREATE_FILE_DEST = '+dgroup1'
7. SQL Plus connect as SYSDBA for SID=ASMDB
8. STARTUP NOMOUNT PFILE=initASMDB.ora
9. Create Oracle database and user tablespace using SQL code below:

```
create database ASMDB
controlfile reuse
logfile '+dgroup1' size 16 M
datafile '+dgroup1' size 2 G
sysaux datafile '+dgroup1' size 2 G
default temporary tablespace temp
tempfile '+dgroup1' size 8 G
undo tablespace undo
datafile '+dgroup1' size 8 G;

create tablespace USER_LOCAL
datafile '+dgroup1' size 60 G;
```

And that's the syntax if we still want to explicitly control the data allocation sizes (which really is no longer a con-

cern, as we're working now at the disk level). So we really could just let Oracle handle all the internal space needs and issue the much simpler syntax of:

```
create database ASMDB;
```

Not only is this much shorter and thus easier to read, but also note that we even got our redo logs created in our striped disk group (in the prior example they were just placed on the filesystem). The main point is that the process of laying out the storage and creating the database is just so much simpler that if you're managing lots and lots of disk drives (i.e., SAN or NAS), you can't go wrong by upgrading to Oracle 10g for ASM alone.

Easier to Change

Of course, you create the database only once, so you might argue that the savings shown above are not reason enough to undertake any changes. So now let's examine what would happen in each scenario if we added four disks. This is where ASM really shines.

Let's assume that our single USER tablespace is nearly full, containing just 10 tables and 10 indexes, where each table consumes 4 GB and each index consumes 2 GB. If we need to create another table and index, we won't have enough room. So we are given four more disks identical to the first four to add to our storage design in order to accommodate additional space requests. In other words, we're going to add 80 GB to our single USER tablespace. Seems easy enough, right?

In the LVM example, we have three primary options:

- Create a new Volume Group VG02 with a new Logical Volume LV05
- Extend existing Volume Group VG01 with a new Logical Volume LV05
- Extend existing Volume Group VG01 by extending Logical Volume LV04

Note: Since we're assuming that all 80 GB is to be dedicated to the USER tablespace, there is no need to create more than a single new Logical Volume for the first two options.

Most people will choose the third option, since we're merely trying to add space to our existing storage design. So here are the steps to implement that choice:

1. fdisk /dev/hdf set its type to 0x8e (LVM partition)
2. fdisk /dev/hdg set its type to 0x8e (LVM partition)
3. fdisk /dev/hdh set its type to 0x8e (LVM partition)
4. fdisk /dev/hdi set its type to 0x8e (LVM partition)
5. pvcreate /dev/hdf /dev/hdg /dev/hdh /dev/hdi
6. vgextend VG01 /dev/hdf /dev/hdg /dev/hdh /dev/hdi
7. lvextend -L +80 G /dev/VG01/LV04
8. ext2online /dev/VG01/LV04
9. SQL Plus connect as SYSDBA for SID=LVMDB
10. Add new space to the tablespace using SQL code below:

```
alter tablespace USER
add datafile '/home/oracle/oradata/LVMDB/user2/user01.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user02.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user03.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user04.dbf' size 2 G,
```

```
'/home/oracle/oradata/LVMDB/user2/user05.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user06.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user07.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user08.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user09.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user10.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user11.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user12.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user13.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user14.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user15.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user16.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user17.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user18.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user19.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user20.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user21.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user22.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user23.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user24.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user25.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user26.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user27.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user28.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user29.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user30.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user31.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user32.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user33.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user34.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user35.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user36.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user37.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user38.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user39.dbf' size 2 G,
'/home/oracle/oradata/LVMDB/user2/user40.dbf' size 2 G;
```

Not only is this example very long, but it also has several problems. First, online filesystem resizing is very tricky business. The author of the `ext2online` utility states that “resizing a mounted file system is inherently dangerous and may corrupt file systems.” And the `ext2online` utility can enlarge only a mounted filesystem; to shrink or enlarge filesystems, one would instead use `ext2resize`—which works only on unmounted filesystems. Of course, that would require taking the tablespace offline.

The LVM approach has some not-so-obvious drawbacks as well. We very likely expected that this solution would result in our data being striped across all eight of our drives—not true. While we can add space to a Logical Volume, we cannot change its striping nature on Linux (although some UNIX platforms’ LVMs do provide such capabilities). So our 10 old tables and 10 old indexes are striped across drives b–e, while our new table and index are striped across drives f–i (since the USER tablespace was already full, new objects will be created in the new space). Even if we exported the tablespace objects, dropped them, coalesced the tablespace, and then imported them back into the tablespace, the Logical Volume would still be set for four-way striping. We’d have to manually do the following if we really wanted eight-way striping:

1. Export the objects in that tablespace (database in restricted session to be safe)
2. Drop the tablespace
3. Drop the Logical Volume
4. Create a new Logical Volume (with striping parameter set as `-i 8`)
5. Create the tablespace (this would have lots of data file lines for all 140 GB)

6. Import the objects into the tablespace

That’s where ASM steps in and makes life easy. The steps are shorter:

1. SQL Plus connect as SYSDBA for SID=ASM
2. ALTER DISKGROUP dgroup1 ADD DISK ‘/dev/hdf’;/dev/hdg’;/dev/hdh’;/dev/hdi’

That’s it! But there’s more. ASM automatically rebalances both its striping and its mirroring of a disk group whenever disks are added, are dropped, or fail—and all with the database completely online. Therefore, Oracle automatically takes care of keeping all your objects fully striped. That’s why ASM can make the claim that it provides near optimal IO balancing without any manual tuning. It simply internalizes and automates that which DBAs have been doing manually for years—trying to eliminate hot spots by spreading things across as many drives as possible. Note that you can control when and how Oracle performs that rebalancing via the `OSM_POWER_LIMIT` and other parameters (but that deserves an article of its own).

Some Performance Too . . .

Oracle 10g’s mantra seems to be “simpler and more automated database management.” All the above exemplifies for some people why ASM itself might be worth an upgrade to Oracle 10g. But this article would be incomplete without some examples comparing the relative performance of the LVM versus ASM scenarios. While we are eliminating the Logical Volume Manager and filesystem, we are nonetheless utilizing more Oracle technology to do much the same thing. So initial expectations are to see fairly similar or slightly better performance—with the advantages once again being far greater simplicity to create and manage.

Here are some relative results for consideration (note—tested using 10g beta #1):

ASM vs. LVM	
Populate an 80 GB database	11% faster
Build indexes for the database	9% faster
200-user concurrent access (OLTP)	5% faster

While these results are not earth shattering, roughly 10% improvements from something that makes the DBA’s life easier is not a bad return on investment for the cost of doing an upgrade. And I’m optimistic that when Oracle 10g is released as production ready, those numbers might improve. Of course, that’s probably just the “Jerk” in me getting overly excited once again. ▲

Bert Scalzo (Bert.Scalzo@Quest.com) is a product architect for Quest Software, which has been very active in the Oracle 10g beta program. Bert has B.S., M.S., and Ph.D. degrees in computer science, has worked as an Oracle DBA for over 18 years, and designed much of the DBA module for TOAD. Bert has also written two books: Oracle DBA: Guide to Data Warehousing and Star Schemas (Prentice Hall, 2003) and, with Dan Hotka, TOAD Handbook (Sams Publishing, 2003).

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

CHEVRON

ORACLE CORP.

Thank you! Year 2005 Gold Vendors:

- Confio Software
- Database Specialists, Inc.
- Data Domain
- Embarcadero Technologies
- Quest Software
- Quovera, Inc.
- RightOrder

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



TREASURER'S REPORT

Randy Samberg, *Treasurer*

Beginning Balance

April 1, 2005 \$ 60,607.92

Revenue

Membership Dues	2,300.00
Meeting Fees	440.00
Vendor Receipts	500.00
Training Day	6,750.00
Advertising	---
Interest	45.84
Miscellaneous	---
Total Revenue	\$ 10,035.84

Expenses

Regional Meeting	520.00
Journal	6,828.14
Membership	11.65
Administration	57.00
Website	---
Board Meeting	541.16
Training Day	6,857.91
Marketing	25.00
PayPal	14.80
Miscellaneous	66.50
IRS	331.00
FTB	800.00
Insurance	---
IOUG-Rep	600.00
Total Expenses	\$ 16,653.16

Ending Balance

June 30, 2005 \$ 53,990.60

NoCOUG Summer Conference

Thursday, August 18, 2005

Session Descriptions

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

KEYNOTE

Foundation Matters

Chris Date

This session is meant as a wake-up call . . . The foundation of the database field is, of course, the relational model. Sad to say, however, there are some in the database community—certainly in industry, and to some extent in academia also—who don't seem to be as familiar with that model as they ought to be; there are others who seem to think it isn't very interesting or relevant to the day-to-day business of earning a living; and there are still others who seem to think all of the foundation-level problems have been solved. Indeed, there seems to be a widespread feeling that "the world has moved on," so to speak, and the relational model as such is somehow passé. In my opinion, nothing could be further from the truth! In this session, I want to sketch the results of some of my own

Having a Laugh with Mogens Nørgaard

(continued from page 6)

Server world. We can take the presentations that said we shouldn't focus on index reorgs and just say SQL Server instead of Oracle."

Outside of technology, we asked Mogens for his hobbies. Writing is one of the activities he enjoys the most. In addition, he says, "My future third ex-wife, as I call her, and I have four kids combined, so that's a hobby in itself." They also stay busy with two goats, 13 chickens, rabbits, about 30 birds, and an aquarium. The company headquarters is in the garage. And, Mogens proudly announces that they are starting a microbrewery at his co-director's house. ▲

Mogens Nørgaard is co-founder and technical director of Miracle A/S (www.miracleas.com), which provides consulting, support, training, and licensing advisory on Oracle and SQL Server, in Maaloev, Denmark. He is one of the original founders of the OakTable Network and the BAARF (Ban Against Any Raid Five—www.baarf.com) movement.

investigations into database foundations over the past twenty years or so; my aim is to convey some of the excitement and abiding interest that is still to be found in those investigations, with a view—I hope—to inspiring others in the field to become involved in such activities.

TRACK 1

High Availability and Oracle RAC

Daniel Morgan, *University of Washington*, and Kent Stroker, *Advanced Technology Services*

Daniel Morgan of the University of Washington and Kent Stroker of Advanced Technology Services will present an overview of the various technologies that contribute to the creation of a High-Availability Oracle environment. They will specifically focus on Real Application Clusters (RAC) and discuss, in detail, 10g RAC and the new Cluster Ready Services (CRS), and then demo Transparent Application Failover (TAF) on their two-node traveling cluster using both the Oracle TAF-OCI demo and SwingBench.

Wait Event Enhancements in Oracle 10g

Roger Schrag and Terry Sutton, *Database Specialists Inc.*

This presentation is designed for DBAs who have experience using the wait event facility in Oracle 9i or earlier, and want to learn what changes have been made in Oracle 10g. Although introduced way back in Oracle 7.0, not a lot changed in the wait event interface through and including Oracle 9i. Sure, Oracle 9i boasts over 400 wait events while Oracle 7.3 had a paltry 106. And yes, Oracle 9i TKPROF reports include wait event information and Oracle 9i v\$sql views show wait times in microseconds. But these are minor enhancements, really. Oracle 10g, on the other hand, brings more significant change to the wait event interface than we have seen in years. Now there are over 900 wait events, a categorization of wait events into classes, wait event accounting at the SQL statement level, easy access to recent wait events for each session, built-in statistics collection by the Automatic Workload Repository, a new time model concept for looking at how sessions spend their time, and a lot more.

Exploring Change Data Capture in Oracle 9i/10g (a Case Study)

Venki Krishnababu, *Nordstrom*

In many organizations a requirement exists for transactional data to feed into decision support systems. Aggregation and summarization of data in DSS and data warehouse systems allow organizations to make informed decisions based on trends and statistical analysis. Traditionally, changes from transactional systems are captured using DML triggers or complex batch programs that extract the change on a periodic basis. With versions 9i and 10g, Oracle introduced a new feature for capturing change at the source very efficiently.

Change Data Capture (CDC), first introduced in Oracle 9i, allows the capture of change at the source

(continued on page 26)

(continued from page 25)

without having to write traditional triggers. This presentation discusses a case study performed at a major Oracle customer site, exploring how CDC compared to traditional triggers and batch processes in terms of development, testing, maintenance and support costs. This presentation focuses on synchronous CDC, including the process flow and steps involved in implementation in a real production database. In addition to providing an end-to-end overview of how an efficient ETL mechanism can be used to load in near real time to a data warehouse, the presentation also shows how to extend CDC to feed cross-functional systems for information integration. This presentation also touches on asynchronous CDC, new in Oracle 10g, which reads from redo logs in order to obtain data changes. This presentation includes a live demo of CDC.

TRACK 2

Assertions, Exceptions, and Module Stability: Applying Design by Contract to PL/SQL Programming

John Beresniewicz, *Oracle Corporation*

This presentation introduces the software engineering techniques of Design by Contract as applied to PL/SQL programming for purposes of creating bug-free code. We will discuss preconditions, post-conditions and invariants and learn how a standardized package-level ASSERT procedure can be leveraged to enforce contract preconditions between calling and called modules. Modularization and exception handling issues will also be covered to some degree, as well as best practices for hardening code. Specific code examples will be presented to illustrate the techniques.

Case Study: Creating Price Lists in an Oracle ERP Dynamic Pricing Environment

Tim Lindemulder, *Agilent Technologies*

At Agilent Technologies, once a month, we need to generate static price lists containing over 14 million items. However, all of our prices are dynamically generated by formulas. Over the course of the past year, Agilent has implemented a set of batch programs to generate a static snapshot of our dynamic prices. The process of designing,

building, testing, and deploying this project uncovered many interesting issues ranging from basic software testing principles to obscure Oracle database setup issues. This talk will describe the project, describe the issues we found, describe their resolutions, and give you a set of lessons that we learned

Table and Query Design for Hierarchical Data without CONNECT BY

Charles Yu, *Elance*

Hierarchical data is popularly represented in an RDBMS by a table where the parent-child relationship is implemented in such a way that if record x is a child of record y, then the value of x's parent_id column is the same as the id column of y's. Since records in such tables may serve both as children and as parents, these tables are referred to as recursive tables. Recursive tables seem to be a very good design for hierarchical data. However, the standard SQL syntax does not support a general query on such hierarchical data with this kind of table design. Oracle comes with native support for querying hierarchical data, known as CONNECT BY. For all its niceness, the CONNECT BY mechanism falls short in two accounts: It is slow in many cases due to the fact that the parent-child relationship cannot be directly indexed; and SQL code built upon CONNECT BY cannot be easily adapted to other database vendors as there is no such support from other vendors so far. This presentation introduces an alternative new mechanism for storing hierarchical data that overcomes these shortcomings.

When Good Optimizers Make Bad Choices (Sometimes)

Dan Tow, *Singing SQL*

Smart optimizers do sometimes make bad choices. This presentation will describe the main root causes for cost-based optimization failures to find the best execution plans, and what the developer can and should do about these failures. I make a case that inaction, motivated by a wish to avoid over-directing the optimizer, or motivated by any other reason, is not generally the best choice.

TRACK 3

Managing Data Risk Management, or, Honey, I Shrank the Beast

Ian Colvin, *Quest Software*

Data risk management poses a complex and costly challenge to many organizations. Regulatory requirements and best practices dictate the need to provide comprehensive, timely and reliable controls that have been proven to minimize risks to the confidentiality, integrity, and availability of the data. This is a familiar story of woe. Ian will present three key principles to put things into perspective. **Look up.** Step back from the myriad details and get a clear vision. Understand the key components and processes that drive compliance and build a model. Do all the requirements have a home? Does everyone have the same vision

ADVERTISING RATES

Contact: **Nora Rosingana**

325 Camaritas Way · Danville, CA 94526

Ph: (925) 820-1589

The NoCOUG Journal is published quarterly.

Size	Per Issue	Per Year
Quarter Page	\$100	\$320
Half Page	\$200	\$640
Full Page	\$400	\$1,280

Personnel recruitment ads are not accepted.

in mind? Who are the drivers of the whole and the pieces? Do we have all the tools we need? Once a satisfactory model is in place, it provides the framework on which to prioritize, plan, manage, and change compliance. One such risk management model will be presented, together with example walkthroughs. **Step up.** Communicate compliance requirements to all affected staff. Ensure that roles, responsibilities, and reporting structures are not at odds with the risk management goals. How do you work with internal and external auditors? Instill a sense of ownership, assess and improve skills, and check the segregation of duties. **Tool up.** Compliance products provide repeatability, scalability, and automation opportunities. There are different categories of compliance products. The challenge is to achieve control over the compliance products themselves, through integration of their risk control models, compliance incident registries, security models, and compliance management standards. How do you pressure the vendors with specific requirements, individually and at the group level?

Serializability: The Hobgoblin of Little Minds?

Iggy Fernandez, *Intacct*

In his presentation to NoCOUG last year, Tom Kyte characterized differences in approaches to data consistency and concurrency as the fundamental difference between Oracle and the other database vendors such as Microsoft and IBM. He said that it can be Oracle's best feature or its worst feature, if you don't understand it. He also said that if you don't understand it, you are probably doing some transactions wrong in your system! Interested already? Then run, don't walk to this presentation! "Serializability," or the notion that concurrently executing transactions must not interfere with each other to produce unexpected results, is such a tricky beast that original research is still being published on the topic. Come learn about the issues and stop doing transactions wrong!

A Brief History of Data (with Apologies to Stephen Hawking)

Michael Scofield, *Loma Linda University*

This session presents a survey of the history of data (particularly structured or tabular data) and the technology used to manage it, with the focus on what characteristics of raw data and information have not changed through the years. What the IT advances do change is our ability to find information, and our ability to associate it, convert it, and display it in newer and meaningful ways. Rows and columns existed in the 17th century! So did null conditions in cells of data. The first non-story commercial data occurred in both the Egyptian and Babylonian empires. The first tabular scientific data is evidenced in a 6th-century astronomer being able to predict a lunar eclipse. During World War II, both American and English scientists invented electronic data processing (computers), which were used to process data, but not manage data. Big difference. Their greatest limitation was their inability to store data. Going back to the ultimate of unstructured

**Don't miss NoCOUG's
Summer Conference:**

Thursday, August 18!

**ChevronTexaco
San Ramon, CA**

data (the human brain) and how it stores and associates business data, we compare this to the feeble efforts of CRM (customer relationship management) to reconstruct that brilliant, intuitive sole proprietor's brain. We shall look at how proprietorships grow into complex (and dysfunctional) bureaucracies, and what happens to data in this evolution.

Managing Data about Terrorists

Michael Scofield, *Loma Linda University*

This session looks at the difficulty of linking data from diverse government agencies to get a better picture ("connect the dots") of suspect activities. The problem with that is the danger of "false positives," especially with insufficient data or misinterpreted data. And, of course, all such data are observations, and have issues of accuracy and reliability. How do you design a database to capture the reliability of suspect observations?



NoCOUG Summer Conference Schedule

August 18, 2005 · 6101 Bollinger Canyon Road, San Ramon, CA 94583

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

- 8:00–9AM **Registration and Continental Breakfast**—Refreshments served
- 9:00–9:15 **General Session and Welcome**—Darrin Swan, NoCOUG President
- 9:15–10:30 **Keynote: “Foundation Matters”**—Chris Date
- 10:30–11:00 **Break and Book Signing**
- 11:00–12:00PM **Parallel Sessions #1**
Track 1: “High Availability and Oracle RAC—I” by Daniel Morgan, University of Washington, and Kent Stroker, Advanced Technology Services
Track 2: “Assertions, Exceptions, and Module Stability: Applying Design by Contract to PL/SQL Programming” by John Beresiewicz, Oracle Corporation
Track 3: “Managing Data Risk Management, or, Honey, I Shrank the Beast” by Ian Colvin, Quest Software
- 12:00–1:00 Lunch
- 1:00–2:00 **Parallel Sessions #2**
Track 1: “High Availability and Oracle RAC—II” by Daniel Morgan, University of Washington, and Kent Stroker, Advanced Technology Services
Track 2: “Case Study: Creating Price Lists in an Oracle ERP Dynamic Pricing Environment” by Tim Lindemulder, Agilent Technologies
Track 3: “Serializability: The Hobgoblin of Little Minds?” by Iggy Fernandez, Intacct
- 2:00–2:15 Break
- 2:15–3:15 **Parallel Sessions #3**
Track 1: “Wait Event Enhancements in Oracle 10g” by Roger Schrag and Terry Sutton, Database Specialists Inc.
Track 2: “Table and Query Design for Hierarchical Data Without CONNECT BY” by Charles Yu, Elance
Track 3: “A Brief History of Data (with Apologies to Stephen Hawking)” by Michael Scofield, Loma Linda University
- 3:15–3:45 Raffle and Refreshments
- 3:45–4:45 **Parallel Sessions #4**
Track 1: “Exploring Change Data Capture in Oracle 9i/10g (a Case Study)” by Venki Krishnababu, Nordstrom
Track 2: “When Good Optimizers Make Bad Choices (Sometimes)” by Dan Tow, Singing SQL
Track 3: “Managing Data About Terrorists” by Michael Scofield, Loma Linda University
- 5:00–6:00 **NoCOUG Networking and Happy Hour. Location TBD.**
Cost: \$40 admission fee for nonmembers. Members free. Includes lunch voucher.

Session descriptions
appear on page 25.

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

NoCOUG
P.O. Box 3282
Danville, CA 94526

FIRST-CLASS MAIL
U.S. POSTAGE
PAID
SAN FRANCISCO, CA
PERMIT NO. 11882