

nocoug

JOURNAL

Official Publication of the Northern California Oracle Users Group

VOL. 17, No. 1 · FEBRUARY, 2003

\$15



Author and NoCOUG member, Chris Lawson, getting a little frustrated during his book project.

Writing a Book About Oracle: Art or Science?

by Chris Lawson

A casual scan of the bookshelves at any large bookstore reveals an interesting trend: There are more and more Oracle books available today, from an increasing number of publishers. The number of new titles has exploded just in the last few years—almost as fast as Oracle releases new database versions!

This huge expansion of new titles also means that there are lots of opportunities for new writers. Presumably, these books are all being published because there really is a

(continued on page 14)

Renew Your Membership!

Annual membership for NoCOUG runs from January 1 through December 31. This means that your membership has expired if you haven't renewed for 2003. Now is the time to renew your membership!

On page 17, you will find the Membership Application for 2003. Or, if you prefer, you can visit our website at <http://www.nocoug.org/joinnc.htm>. There, you can download the application, fill it out, and send it in with a check.

Don't miss any of the benefits of NoCOUG in 2003, including our upcoming Winter Conference on February 20 at Oracle in Redwood Shores. There's no charge if you're a member, and \$40 for non-members. That alone makes the \$70 Individual Membership price a bargain! ▲

TABLE OF CONTENTS

Writing A Book About Oracle	1
Renew Your Membership.....	1
Don't Miss Our Winter Conference	2
NoCOUG Board	2
User Groups.....	2
Publication and Submission Format.....	2
Advertising Rates.....	2
President's Message	3
Board Member Spotlight.....	5
How Not to Be a DBA.....	7
Treasurer's Report.....	13
Sponsorship Appreciation.....	13
NoCOUG Membership Application	17
Being Prepared When Disaster Strikes	18
Tech Tips.....	19
The Art of Creating Subset Databases	22
Tech Tips	23
Resource Corner	25
NoCOUG Winter Conference Session Descriptions.....	27
NoCOUG Winter Conference Schedule.....	28

—ADVERTISERS—

Quest Software.....	4
BMC Software.....	6
Embarcadero Technologies.....	10
Informatica.....	10
LECCOTECH.....	16
Mission Critical	16
IOUG Live! 2003.....	26

Don't Miss Our Winter Conference!

The NoCOUG Board has planned a great conference that you won't want to miss. It's taking place on Thursday, February 20 at the Oracle Conference Center in Redwood Shores. Sponsored by Oracle Corporation, the one-day conference will have three parallel tracks:

- Database Administration
- Application Development
- Special Presentations by Oracle Corporation

Registration begins at 8:00 a.m. For more details, visit our website at www.nocoug.org. ▲

Other User Groups

Local

NorCalOAUG – Northern California Oracle Applications Users Group

- **Contact:** Michael Capelle
(650) 562-1167
- **Email:** capelle@tru-course.com
- **Website:** www.norcaloaug.org

Sacramento

SacOUG – The Sacramento Oracle User Group

- **Contact:** Ravi Verma
(916) 705-3261
- **Email:** ravi.verma@telcommand.com
- **Website:** www.sacoug.org

International

IOUG – International Oracle Users Group

- **Website:** www.ioug.org

U.S. Domestic

OAUG – Oracle Applications Users Group

- **Website:** www.oaug.org

ODTUG – Oracle Development Tools User Group

- **Website:** www.odtug.com

Canvassing calls by employment recruiters to local chapter contacts is strongly discouraged.

NOCOUG BOARD

President

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

Vice President

Darrin Swan, LECCOTECH
darrin@leccotech.com

Treasurer/Secretary

Judy Lyman, Contra Cost County Public Works
gooma@california.com

Membership and Past President

Joel Rosingana, Independent Consultant
joelros@pacbell.net

Webmaster

Vadim Barilko, Independent Consultant
vabarus@onebox.com

Journal Editor

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

Vendor Relations

Mike DeVito, BankServ
mike@devito.com

IOUG-A Representative

Hamid Minoui, UPS
hamid.minoui@ups-scs.com

Members, At Large

Ganesh Sankar, Providian Financial
bgs2k2@yahoo.com

Vilin Roufchaie, Cingular Wireless
vilin.roufchaie@cingular.com

Conference Agenda Coordinator

Colette Lamm, Independent Consultant
colette_lamm@yahoo.com

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 May issue is April 1, 2003. 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 © 2003 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.

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.



Happy New Year!

by Roger Schrag, President, NoCOUG

Hello! My name is Roger Schrag, and I am the new President of your Northern California Oracle Users Group. I always look forward to the new year because of the new beginnings and fresh hopes that it brings. 2002 was not the best year for most of us, and so we can all hope for a brighter 2003.

I can't promise you that the job market will improve or that the economy will bounce back. But one thing I am sure of is that NoCOUG will be a great resource for Oracle DBAs, developers, and Oracle users in 2003. I'm excited about what we've got planned for the year. We will put on four quarterly conferences, publish the *NoCOUG Journal*, and keep the NoCOUG website up-to-date with valuable information. We've also got some surprises in store for you as well.

The NoCOUG Winter Conference is coming up real soon, so let's talk about that first. We'll be meeting at the Oracle Conference Center in Redwood Shores on February 20. We've got some great sessions planned: Craig Shallahamer, who runs OraPub and teaches expert performance tuning seminars across the U.S., will talk about how Oracle complexity breeds simplicity and also about Oracle's touch-count data block buffer algorithm. Bert Scalzo, one of the Quest Software folks who brings us TOAD, will talk about advanced SQL tuning and successful dimensional modeling of very large data warehouses.

There will be other great sessions as well, including a full track of presentations put together by Oracle Corporation. Be sure to check www.nocoug.org periodically for the most up-to-date conference information.

We also have the Spring, Summer, and Fall conferences to look forward to as well. We'll be bringing you a good mix of sessions and activities to help

satisfy your hunger for more knowledge. We are working with local Oracle users who want to share their real-life experience, famous Oracle experts who can impart their wisdom, and Oracle Corporation who can enlighten us as to where they are headed.

I have been an Oracle DBA and application architect for 14 years now. I've been active in Oracle user groups since 1993, primarily because I'm one of those Oracle users who always wants to learn new things. What better way to learn than from your fellow Oracle users? And where will you find a cheaper or more convenient Oracle training opportunity than your local user group?

I got involved in the leadership of NoCOUG back in 2000 because I wanted to help make a difference. I joined the NoCOUG board that year, and since then I've served as webmaster, vice president, and now president. I think that being a NoCOUG board member is fun because I get to work with people—fellow board members, NoCOUG members, and internationally recognized Oracle experts who come from all over to participate in the quarterly conferences.

I am part of a great group—there are 11 of us on the board, and our objective is to have fun while bringing you world-class events, publications, and networking opportunities to help you advance your Oracle knowledge and career.

As a DBA sometimes I get too buried in SQL and V\$ views, so work-



Roger Schrag

ing with NoCOUG gives me a great opportunity to exercise a different part of my brain. And that's not to mention the fun I have with the nice guys and gals on the board, nor the sense of accomplishment we feel as a group after a successful conference!

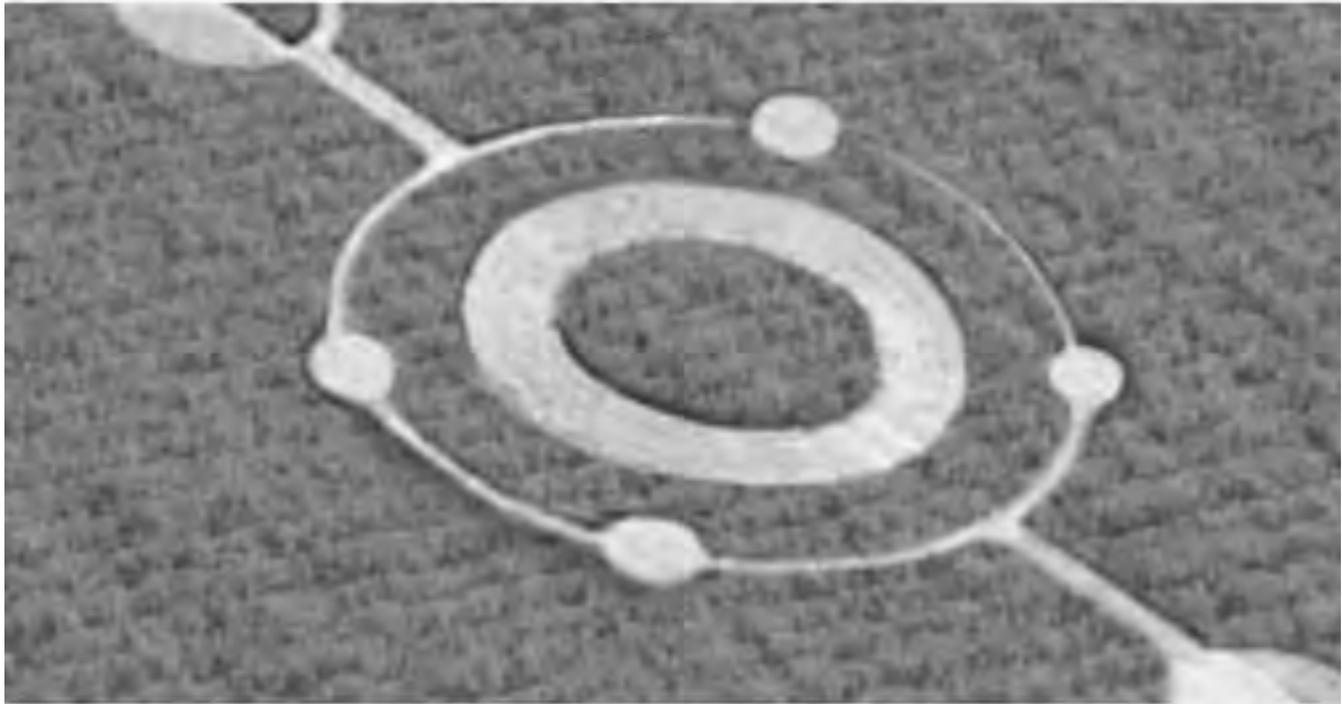
But enough about me. What about you? Why do you come to NoCOUG conferences? What ideas do you have for how to make NoCOUG events even better? I'd love to hear your thoughts! Shoot me an email at rschrag@dbspecialists.com. Or speak up when you see me at the February conference. I really want to hear what you have to say.

At a conference a year or two ago, a DBA told me that the lunch break was too long. So we introduced the Roundtable Discussions at the tail end of the lunch break, and the response has been tremendous. Many people have asked if they could pay their membership fees by credit card, so we hooked up with PayPal. These are just two examples of how your input shapes the organization. Let me know what you think, and I'll see what we can do!

Do you have any interest in speaking at a future NoCOUG conference? Or do you know somebody who does? Or would you like to help us on conference day to keep the event running smoothly? Let me know, because we are a volunteer-run organization, and we can always use your help.

Here's to a successful 2003 for all of us. Let's hope the economy bounces back and the job market improves for everyone. Meanwhile get the most out of NoCOUG that you can, and let me know what we can do to make NoCOUG even better. See you on February 20th! ▲

With the Right Tools, Anything is Possible



Quest Software Brings Higher Intelligence to Database Development

Smart people with good tools can mystify the masses and achieve the impossible. With Quest Software's market-leading tools for Oracle development, so can you. Quest Software offers the complete solution to design, develop, test and deploy code quickly and accurately, leaving you plenty of time to consider the possibilities.

The evidence is clear: more than 200,000 users worldwide believe Quest Software's development tools are the intelligent choice. Download your free trial versions from www.quest.com and see for yourself.



www.quest.com • 1.800.306.9329 (U.S.) • 1.949.754.8000 (outside U.S.)



Meet the Vendor Relations Coordinator

If you've been regularly attending our NoCOUG quarterly conferences, then you'll recognize Mike DeVito. Mike is entering his second year on the NoCOUG board. This year he'll take on the role of Vendor Relations Coordinator working to bring in vendors to exhibit and sponsor our quarterly conferences. Mike enjoys helping to organize the conferences. "It's always nice to mingle," he says adding that it's nice to share experiences and learn what challenges other Oracle users are facing.

Mike has been a great addition to the board. He became more involved through attending IOUG conferences and working with colleagues who were involved with NoCOUG. "I've always been a member of the IOUG, and getting involved with the local chapter was a natural extension of that," says Mike.

"I first used Oracle technology when I was subcontracting as a developer for the Air Force in the 80s," says Mike. They decided to move their technology to HP-9000 and Oracle and Mike learned a lot about Oracle and PL/SQL.

Mike has been a fan of technology since his teens and majored in management of information systems at Cal State Fullerton. "My father encouraged me to get into computers," says Mike who grew up in the shadow of Disneyland. "In high school, Rockwell International had a training program that I was involved in." The company bussed students in to help work on the IBM mainframe.

"Knowing Oracle enabled me to move around," says Mike, who enjoys traveling and learning about other cultures. "I was able to get contracts in the south of France and

in Amsterdam," says Mike who enjoyed the change from California. Mike currently works as a DBA in downtown San Francisco where he has lived for the past two years. When he's at home, his two 10-year-old cats, Bart and Lisa, keep him company.

For two years Mike has been taking improvisation classes which focus on helping people become more spontaneous, creative, and flexible. He says, "By taking the classes, I've learned how to get more fun out of life."

Sailing is another passion that



Mike DeVito

Mike has. He was just elected the president of the San Francisco Bay Oceanic Crew Group, which has about 100 members. Mike explains, "We are a group of boat owners and crew members. We match people up to go on sails once or twice a month. We also have speakers on boating safety and marine environments." You can check out the group's website at www.crewgroup.org.

After speaking with him and hearing about all his activities, you'll wonder how Mike has time to volunteer to help NoCOUG! We don't know the answer, but we're glad he does! ▲

Thank you to NoCOUG board member Vilin Roufchaie and Cingular Wireless for hosting our Saturday morning NoCOUG board meetings in Pleasanton each quarter.



SPACE EXPERT™ GIVES YOU A SMARTER WAY TO
visualize YOUR DATABASE'S SPACE PROBLEMS.

It's not always easy to see where pesky space problems are lurking, ready to degrade the performance of the database that your business depends on. And it's not always easy to keep your database operating at peak levels without lots of highly trained DBAs.

Until now, that is.

Introducing Space Expert™ for Oracle from BMC Software. It intelligently and automatically visualizes, isolates, analyzes and corrects space-related problems. Enabling even a novice DBA to perform the work of many. Which means the IT department's SLAs are good as gold. IT managers can concentrate on managing. Your database's performance is enhanced. And your company is more competitive than ever.

See for yourself. **Register for a 30-day trial copy of Space Expert for Oracle today at www.bmc.com/distdata/spaceexpert.** Then you'll discover just how easy we make it for you to help your database, and your business, really perform. Because it takes intelligence, not hocus pocus.



How NOT to Be a DBA —The Top 20 Mistakes

by Rachel Carmichael, Dragonfly Consulting

Abstract

This paper will take an in-depth but lighthearted look at some of the “worst practices” a DBA can follow. You will learn just why each of these practices is a problem and how to turn a “worst practice” into a “best practice.”

Overview—Why Do We Care About Mistakes?

Think about how you learned when you were a child. Your parents would try to tell you what to do or not do, based on their own experiences. Sometimes, you listened and didn't have to repeat their mistake to learn what not to do. Sometimes, you didn't listen and you made your own mistakes and learned from them. Mistakes are a valuable learning aid. Some of the world's best inventions came about as a result of a mistake—Velcro, Silly Putty, the microwave oven. Unfortunately, most mistakes don't have such good fallout and we have to spend hours and days repairing the damage they cause. This paper is intended to give you an overview of some of the more dangerous mistakes you can make as a DBA, why they are a mistake, and what to do instead.

Backup and Recovery

There are three types of backups you can do—online, offline, and logical. An online backup is done while the database is open and accessible by users and applications. You must be in archivelog mode for this to be a valid backup. You may back up some or all of the datafiles and you must be sure to back up the archived redo logs that are created between online backups. An offline backup is done while the database is shutdown and should include all the datafiles, online redo logs, and at least one controlfile. A logical backup is an export and may be done as a full database export, or you may specify users or particular tables to be exported.

Recovery also has several different types. You can recover tables or users by importing from an export, do a full database restore from a point in time from either an offline or online backup or recover only datafiles or tablespaces from an online backup. With an offline backup or export you can only recover to the point in time at which the backup was done, while with an online backup you can manage the recovery process much more closely. You can recover to an SCN (transaction), a specific point in time, or to a point in time **after** the backup was made (using the archived redo logs).

Details on how to perform these types of backup and recovery are beyond the scope of this paper and can be found in the Oracle documentation and in *The Backup and*

Recovery Handbook by Rama Velpuri. Okay, now that we know what backup and recovery are, let's take a look at some of the more common mistakes.

Don't Backup If You Are on RAID 5

This one appears to make sense, as a RAID 5 device is able to rebuild missing blocks by using the other blocks and the parity block if a disk goes bad. The problem is, one bad disk is not the only reason you might need to perform a recovery. And without a backup, recovery is almost impossible.

Some of the reasons you might want to recover from a backup:

The data in the database gets corrupted

It doesn't matter if all the disks in the RAID 5 device are there and accessible if you have logical corruption of your data. What happens when a programmer forgets the where clause and updates 1000 more records than he or she was supposed to? Or what do you do when Oracle returns a block corruption error?

The entire disk farm crashes

RAID 5 only obviates the need for recovery when you have enough disks from the raid set to recreate the data. With two or more disks unavailable, your database is gone.

You lose your data center

As we found out only too well on 9/11/2001, disasters happen. Without some form of backup, **kept at an offsite location**, when you lose your data center, you have lost your database. And with so many people depending on 24/7/365 access, loss of the database could mean severe financial problems for the company.

Don't Do Exports If You Are Backing Up

Just because you have a backup, doesn't mean you may not need a daily or weekly export of your database. What do you do if someone does a delete of all the data or the wrong data and then commits the change? If you have online backups, know the exact time of the error and have another server and a lot of disk space. You **can** recover from this situation without taking your production database down. But that takes a lot of time and luck. If you need to recover everything to the moment of the problem, then yes, you will have to do this. But for a table that changes rarely or one that is updated with flat file feeds, wouldn't it just be simpler to import the data back into the table from a recent export? Even if you have to recreate some missing data, it still may be worthwhile to do the import.

And an export file gives you something that an online backup doesn't. Using a full database export and importing with `show=y` and `rows=n` parameters set, you can generate an accurate set of documentation of your database. The output of this import will list every SQL statement, in the proper order of execution, to recreate an empty copy of your database. The Utilities manual has the details on all the parameters for the export and import programs.

Never Test Recover Procedures

This is one of the worst mistakes people make. They ensure that they are doing the backups, they might even write recovery procedures, but they don't test them under real conditions. Three a.m., sleepy, and in crisis mode, is not the time to find that you've left out an essential piece in your recovery checklist. Recovery procedures should be tested on a regular basis. Sit down with your data entry management and review the process. Do a step-by-step walk-through on a test server to see exactly what needs to be done when and how long each step will take.

Document what you've done, as if you were writing these instructions for someone who has never seen a computer before. List the files that will be needed and where they are. Include contact information for everyone who is affected by a recovery—users, IT management, Data Center personnel.

Testing recovery also tests that your backup procedures are complete and include every file you will need should you have to do a recovery. Do you have the controlfiles? What about all the varied Oracle parameter files? Does someone back up the more static files, such as the Oracle binaries themselves? Are there shell scripts you need to run your daily database monitoring? Even if you have database recovery procedures in place and test them frequently, you need to make sure that all the essential, associated files are also backed up on a regular basis. And while we're thinking about it, every time you back up a datafile, it's a good idea to run a verification of the backed up datafiles to make sure there are no corrupted blocks. Oracle provides a utility, `dbv` (on *nix platforms) or `dbv.exe` (on Windows platforms), located in the `ORACLE_HOME/bin` directory. The utility is documented in the Administrator's Guide.

Monitoring/Tuning

Once you have the backups under control, and you have a way to recover from a failure, you should next move on to monitoring and tuning the database. In this case, I am not talking about tuning the various SQL statements that the applications run (although that is also something you should be doing) but about tuning the overall parameters and monitoring the health of the database. To be able to tune, you first need to monitor the database, and get a sense of what is normal for each of your databases.

There are many good monitoring products on the market. But even if you choose an "off the shelf" product, it is not necessarily a bad idea to familiarize yourself with the various `V$` and `DBA_` views that Oracle provides. There are

often times when you must investigate problems while you are in a remote location such as a user's workstation or your home. At these locations you might not have access to your monitoring or tuning products. Also consider the fact that we do not all stay at one job forever. What would you do if your new employer does not use those products and does not allow you to purchase them?

In the latest releases, Oracle has replaced the old `BSTAT/ESTAT` SQL scripts with the `STATSPACK` stored package. This package now allows you to collect and store snapshots of the database statistics so that you can do some trending and keep a baseline of when performance was acceptable. If you don't know what it was like when it was good, you can't know when to stop fixing things.

In addition to monitoring the performance health of your database, you should be monitoring other things about your database. If you run in `archivelog` mode, you should be checking on how quickly your `archivelog` destination is filling. No matter what mode you run in, you should be checking on the frequency of log switches.

No Need to Monitor, Users Will Inform You About Problems

How many times have you picked up your phone to hear "Oracle is down"? Users are quick to call when there is a problem accessing the database. You'll get complaints that the database is slow or that it's "down." Fortunately, the problem is not usually that severe.

If you wait until the users call, and don't do any proactive monitoring, you are going to have a really tough time figuring out what's wrong and where the problem really lies. Oracle statistics are getting better and better: With the release of `STATSPACK`, you have the ability to save snapshots of your database statistics so that you can look for changes over time. It's always better to catch a potential problem and deal with it before you have a line of screaming users outside your door.

Tune Using Cache Hit Ratios Only

In older versions of Oracle, we were told to look at the buffer cache hit ratios to see if our databases were healthy and performing well. If you had a hit ratio over 80%, you could figure that you were doing well, and that there wasn't much else you could do to increase performance. With the release of Oracle8i, that recommendation changed. Oracle now recommends that you look not just at cache hit ratios, but also at the wait events in the `v$sysstat`, `v$session_wait` and `v$system_event` views. Why the change of direction? Well, hit ratios can be misleading. You can have a nearly perfect hit ratio of 99% and still have users waiting endlessly (at least in their view) for queries to return data to them. Or you could have a ratio of less than 70% and no one complaining of waiting. So the ratios may not really tell you everything. In the first case, Oracle could find every block you needed in the cache each time you asked for it. But if you had too few latches controlling the buffer cache chains, you would still spend a long time waiting for your session to get access to the buffer cache latch so that you could get to that block. In the second case, you might have a data warehouse application that does a number of large block

reads, which will flush the buffer. But since you've made sure there are enough latches available, there is no waiting. On the other hand, if there are waits in your database, that is a clear indication of a problem. If you can identify not only who is waiting, but also what they are waiting on and why, you can go a long way towards resolving the problem. You should look at **both** cache hit ratios and wait events to accurately monitor your database. More detailed information on tuning can be found in Oracle documentation and *Performance Tuning 101* by Vaidyanatha and Deshpande.

Always Increase the Shared Pool to Solve Performance Problems

This one sounds right, doesn't it? After all, if you are getting "unable to allocate x bytes of shared memory," it must mean that you need more memory. Except it doesn't. You might actually want to decrease the size of your shared pool so that it flushes itself more frequently. Remember, the larger the shared pool, the more memory Oracle has to read through before it decides that it has to re-parse the statement. Sometimes it's actually faster to force the re-parse. And if the shared pool is too large, then older, unused statements don't get flushed out and space that could be reused isn't.

Database Standards

"Standards? We don't need no stinkin' standards!" Does it ever feel like that where you work? Everything is a rush, no time to organize how things are done within the database. Too often, we give in to the pressure to "get it done" and don't maintain the standards we have set up to help keep the database running smoothly. The problem with that comes later on, when you have to deal with the fallout of mislabeled datafiles or users who make changes they shouldn't because you haven't locked down their privileges.

Don't Change Oracle Defaults

If Oracle has something set as a default, it must be right, don't mess with it. Well, for years, until the release of Oracle8i, if you used Oracle's defaults to create your database, you'd end up with a 2K blocksize. While that might have been fine in the days when a large database was under 100MB, and you weren't storing lots of data, in today's world of huge data warehouses, a 2K blocksize is ridiculously small. Oracle doesn't know what your application is going to do. You may need a very large blocksize for your database because each of the rows in your tables are going to be large. Or you might need to set the `db_block_buffers` parameter higher (it defaults to 2500 in 8.1.7 on NT). If you aren't going to use Java in your database, there is no need to set the `java_pool_size` and you can bring it down to something negligible.

If you don't need to change the default parameters, don't. But don't assume that if Oracle sets them, they are correct for your needs.

Reorganize Tables When They Have More Than a Few Extents

This is another of those myths from the beginning of time (or at least Version 6). In the earlier versions of Oracle,

it was indeed recommended that you reorganize tables and indexes when they had more than about 10-20 extents in them. These days, if you have properly sized your tablespaces (see "How to Stop Defragmenting and Start Living" in the Metalink Top Tech doc archives, under Database Administration/Storage) so that the extents are all either the same size or multiples of the same size, you don't need to worry about a large number of extents fragmenting your tablespaces. With the newest releases of Oracle, this is no longer a problem. You can set the maximum extents in the storage clause to "unlimited" (this works out to 2147483645 or 2GB, somewhat excessive). In general, Oracle can handle about 4096 extents before it starts to affect performance.

Index Every Column in Every Table

You would think that one way to ensure that you can get to any row quickly, no matter what the query is, is to index every column in every table. The problem is, the more indexes you have, the more upkeep Oracle has to do every time you make a change to the data in the table. And if an index doesn't have a large distribution of values (a gender column as an example), but the table has a large number of rows, then Oracle will tend to ignore the index anyway, and all you've done is add unnecessary overhead. Make sure you know exactly **why** an index is needed. Talk to both the developers and the users to make sure that they understand what they are asking for. Do an explain plan on the query to see what Oracle thinks is the best way to access the data. Try tuning the existing query before throwing an index on the table.

Turn On AutoExtend to Fix Space Problems

AutoExtend will add space to your datafile, up to the maximum you set, every time you run out of space while adding data. The problem is, Oracle doesn't let you know when it's done that. Nothing is recorded in the alert log, there is no flag set or count updated in any of the data dictionary tables. So you don't know that you are eating up disk space until you suddenly run out of space.

For systems doing large data loads that take a long time to complete, AutoExtend can be a godsend. You don't want to have a job fail four hours into a five-hour run because you ran out of room in the tablespace. However, you should also be monitoring the amount of free space available in the datafiles and monitoring whether or not there is enough space for your objects to allocate another extent.

Don't Analyze Tables or Generate Statistics

If you are running a legacy system that was tuned for the rule-based optimizer, this is not a bad thing to do—but make sure to set the `init.ora` parameter `optimizer_mode = rule`, or you will run the risk of Oracle attempting to execute your query using the cost-based optimizer without any statistics. In that case Oracle "guesses" at what the statistics are and you can get a wildly inappropriate execution plan. If you **do** analyze your tables, make sure you analyze all of them. If you are using `optimizer_mode = choose` and even one table in the query is not analyzed, Oracle will estimate

the statistics for you, and they will almost certainly be wrong. Please note that you should **never** analyze tables owned by SYS or SYSTEM, but only application tables.

Grant Everyone Connect, Resource, and DBA

Oracle supplies three default roles with every database, although they keep telling us that this can change with any new release. If only for this reason, it's not a good idea to depend on these roles. In addition, take a look at the privileges that are granted to these roles. The most dangerous role to grant indiscriminately is the DBA role. Granting this role to a user is like handing the keys to the kingdom to them. With the DBA role, a user can create new users, change passwords, create and drop tablespaces, make changes to any user's tables or data. In 9.0.1, the DBA role had 124 different system privileges, all of them with the admin option allowing the user to pass these privileges on to other users. With RESOURCE, you only get eight privileges, all of them **create** object privileges. That doesn't sound too bad, now does it? Well, there is a "hidden" privilege that is granted along with the others when you grant RESOURCE and that is **unlimited tablespace**. This is exactly what it sounds like—the ability to use any tablespace and any amount of room within any tablespace. Which means that if the user is careless and you haven't modified their default tablespace, they can accidentally create objects in the SYSTEM tablespace. Not a good idea.

The final default role Oracle supplies is CONNECT. This

one also sounds innocuous; after all, it should mean that the user could just connect to the database. Unfortunately, that's not all it means. A user with the CONNECT role can still create objects, but at least this user is restricted to the tablespaces that you have given them access to. A much better idea is to spend some time with the developers and business users and find out the real role of each user account. Then create as many roles as necessary (PAY-ROLL_CLERK, CUSTOMER_SERVICE, etc.) and grant them appropriately.

Drop Datafiles When You Need More Space

Many times we allocate more space than we need, either by adding extra datafiles when we build the tablespace because the information on sizing we have is wrong or because we mistakenly create a datafile on the wrong disk. While it would be nice to get that OS disk space back by just dropping the unneeded datafile, you will not be able to open the database the next time you do a shutdown and startup. There are only two ways to reclaim that space. If you are sure that Oracle has not written anything to the datafile, you can resize it to a very small number of blocks. This will keep the datafile in the database, but will reduce the amount of actual disk space it uses. Or you can export the contents of the tablespace, drop the entire tablespace, recreate it with the correct number, size, and location of files and then import everything back in. Time consuming and in today's world of constant accessibility, not usually possible.

Dive into any database administration challenge. Without the risk.

DBArtisan. The fastest, easiest—and safest—way to manage multiple databases.

www.embarcadero.com/download/artisan.htm

 **Embarcadero Technologies**
do more now

©2000 Embarcadero Technologies, Inc. All rights reserved. Embarcadero Technologies and DBArtisan are registered trademarks of Embarcadero Technologies, Inc. All other products are the trademarks of their respective companies.

INFORMATICA
NOW YOU KNOW™

Visit informatica.com to see how you can use your enterprise data for smarter decisions.

User/Developer Standards

It's usually easier to set and keep to database standards yourself than it is to get developers and users to accept and adhere to standards. Often you'll hear "but it's just this once" or "I promise to fill in the paperwork later." Unfortunately, it's never just once and the paperwork never gets done.

Have Only One Account for All Users

This sounds reasonable, doesn't it? It definitely makes your life and that of the developers easier. Only one account to worry about, one set of privileges to maintain, one username/password to code for. And no way to track who did what when there is a problem. It doesn't help you to audit changes if they are all made by the same username. If the application has only one use, then you don't need different roles and privileges for it. But if the database is going to be used by several departments, each doing a different sort of work, then using a single account is a sure recipe for disaster. You can get around this by using roles, but then each application has to set the default role on login and no user can be allowed to connect to the database in any way but through the application. And while roles will prevent privilege problems, they still don't allow you to audit what and when something was done and by whom.

Implement Critical Fixes in Production Without Testing

"It's an emergency, we have to put this in **right now**." Often a developer will come to you and insist that a change has to go in immediately. Most of the time they haven't tested the change or worked out a back out plan. If you can't stop the change from going in, and the change is to data, make sure you have a good copy of the data before the change is implemented. If it's to a table structure, export the table so you can reload it if necessary. Often times you can simply say "no" and insist that changes be tested or go through the normal change control process before it goes into production.

Don't Keep a Source Code Repository

Can you guarantee that the code in development is an exact copy of production? Unless no changes are being made to the application, you can take it as a given that the code will be different. If you don't have a source code repository and you lose your production code, you will have no option but to try to recreate from code that is in flux. A source code repository maintains a record of what is being changed, who is changing it and what state the code is in. Even if you don't have a formal, third party repository package, it's essential that you maintain a clean copy of production code somewhere.

Give Programmers Unrestricted Access to Production

No programmer would ever go in and change code or data without testing it, right? To be fair, we all have had times when it would be easier to just go in and make that quick fix. Unfortunately, that quick fix often leads to more time spent fixing the fix. At one site, one of the program-

mers went to update an old order in production. The problem was the code he used:

```
update order_table
set code = '...';
where order_number = '...';
```

If you look closely at the second line, you'll see a ";" Yes, he updated every single order in the table. Then, to make matters worse, he exited the SQL*Plus session, mistakenly thinking that would force the change to rollback. It doesn't. We spent three hours creating a copy of the production database on a staging machine, rolling forward to just before his change. Then, through database links, we put the data back the way it was before. No programmer was ever allowed direct access to production after that.

General Practices

Assume Every Application/Database/Environment is the Same

I once had someone tell me "there is only one DBA job in the world and you have it." Once I stopped laughing, I realized that this applies not just to the job but also to most work environments in general. No two applications are identical, no two databases are the same and no two environments are identical. Even the backup site for a database/application is going to be slightly different.

If you tune an OLTP database the same way you tune a data warehouse, you have a recipe for disaster. Backups, data loads, tuning, even indexing will be different because the needs of the users will be different. You need to look at each database and application as unique and deal with it on its own.

Don't Script or Document Anything

If you are absolutely positive that you will never, ever have to run that "ad hoc" query for the user again, then don't save it to a script. If you are certain that you wrote it perfectly the first time and the user will never ask for an adjustment to what you gave him or her, then don't save the query to a script.

More often than not, that simple little quick and dirty SQL script you wrote becomes a nightly/weekly/monthly run that has to be done. Or else the user you wrote it for has finally looked at the output (three days later) and realizes that while you gave them what they asked for, it wasn't what they really wanted. So could you please run it again, but change just this one thing? If the script is at all complicated, you are going to spend an hour or so regenerating it if you haven't saved a copy. The same thing goes for documentation. You may be so close to the application/database that you know exactly what has to be done without needing to refer to a procedural manual. Many people believe that they don't have time to document the code they've written or the processes and procedures they follow on a regular basis. In one sense, that's job security, since no one else knows how to do what you do. On the other hand, it does make it difficult to take time off, since no one else knows how to do what you do.

New Releases

Oracle puts out new releases of the software on a regular basis. Most often they are patch set releases (where the major version number does not change) but about every 18 months to 2 years a major new version is released.

Upgrade Immediately

Always install a new release of Oracle the instant it comes out. After all, the new release is designed to fix all those problems in earlier versions of the software. New releases of software, no matter who the manufacturer is, tend to have “undocumented features” in them. (No one ever sends out buggy software!) Make sure you have spent some time looking at the software and the features, so that you know if the new release changes something your developers have been using. Oracle will occasionally change the syntax of a statement or the default for a feature and if you haven’t spent time checking things out, you end up with production problems.

Use All New Features

If Oracle put the new feature in the software, you must need it. Wrong! Not all the new features will make sense for your applications and users. Spend some time looking at what the new functionality is and how it will affect your database (both performance and space) as well as how much recoding your developers will have to do to implement the features. It’s not always necessary to be on the “bleeding edge” when it comes to the database.

Upgrade in Place

Upgrading in place saves disk space, as you simply install the software over the existing software. Of course, you never have a problem with an installation and the users are always happy to give up access to the database for several hours while you do the upgrade. And most especially, upgrades never have problems and have to be backed out. Whenever possible, install the new version of the software in its own ORACLE_HOME. This can be another sub-directory under the existing tree, or a new tree in and of itself. By separating the versions you can deal with installation problems without taking the database away from the users. Once the new software is installed, you can then shutdown the database and try to bring it up under the new ORACLE_HOME. If there is a problem (assuming you didn’t have to run conversion scripts) you can move back to the prior version by shutting down, resetting the ORACLE_HOME and reopening the database.

Don’t Read the Release Notes

We are all, always, pressed for time. This means we tend to skip things like reading documentation until we have to dive in and find something specific—this especially means we don’t read the release notes. In this day and age, when we have short or no maintenance windows, we **have** to read the release notes to see if the release has anything in it that will cause our database or applications to blow up. At one site, the programmers had taken advantage of an “undocumented feature” and coded for it.

We upgraded, no one read the release notes and the

application blew up. Instead of planned changes to the system, we had to make emergency fixes to application code. Not something you want to have to do on a regular basis.

It really doesn’t take much time to at least skim through the release notes and decide if this release is critical or if you can wait for the next one before you upgrade the database software.

Information Sources

In addition to the documentation, there are some valuable resources online. These are:

Websites

- technet.oracle.com - Online source for documentation, product downloads, discussion forums
- metalink.oracle.com - patches, technical forums, bug database, iTARs, platform-specific documentation, white papers
- <http://www.oracle.com/education/oln/index.html> - Online Oracle education—recorded presentations

Listservs

Email listservs allow you to post and read messages to and from a group of other Oracle professionals. While the lists can generate a large volume of mail, and many of the posts won’t be relevant, there will always be those one or two posts that solve a problem that’s been troubling you for days. Some of the better known listservs are:

- comp.databases.oracle.server
- ORACLE-L email list (to subscribe, send a message to ListGuru@fatcity.com with SUBSCRIBE ORACLE-L <your name> in the body)
- www.lazydba.com.

Conclusions

After all this, there are only two critical mistakes you really need to watch for. The first is to not have a backup so that you can go back to a point in time and start over. As long as you have that, almost anything else can be dealt with. The second is to not test and plan for all modifications you make. It’s much faster in the long run to take some time up front and **think** about what it is you are going to do and then do it, rather than just rush in blindly and spend hours recovering from the problems you end up causing. ▲

Rachel Carmichael has been an Oracle DBA for over ten years. She chairs the DBA Special Interest Group for the New York Oracle Users Group, and is the meeting agenda coordinator for NYOUG. She co-authored the Oracle Press book, Oracle SQL and PL/SQL Annotated Archives, 1999 with Kevin Loney, the Oracle Press book, Oracle DBA 101, 2000 with Marlene Theriault and James Viscusi. She can be reached at wisernet100@yahoo.com.

This article is excerpted and reprinted with permission from the IOUG Live! Conference, April 2002.



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 membership dues. If your company is able to offer sponsorship at any level, please contact NoCOUG President Roger Schrag at rschrag@dbspecialists.com.

Long-term event sponsorship:

LOCKHEED MARTIN

CHEVRON TEXACO

ORACLE CORP.

Thank you! Year 2003 Gold Level Support Vendors:

- BMC Software
- Database Specialists, Inc.
- Embarcadero Technologies
- Informatica
- LECCOTECH
- Quest Software, Inc.

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



TREASURER'S REPORT

Judy Lyman, Treasurer

Beginning Balance		
September 30, 2002		\$ 52,787.79
Revenue		
<hr/>		
Advertising	\$ 100.00	
Sponsorships	-	
Membership Dues	1,808.69	
Meeting Fees	540.00	
Vendor Receipts	3,485.20	
Interest	65.08	
Vendor Mailing	-	
Total Revenue		\$ 5,998.97
Expenses		
<hr/>		
Regional Meeting	\$ 4,430.43	
Journal	6,019.85	
Membership	-	
Administration	1,220.00	
Website	75.00	
Board Meeting	587.42	
Miscellaneous	425.79	
IRS	-	
FTB Tax	500.00	
Total Expenses		\$ 13,258.50
Ending Balance		
December 31, 2002		\$ 45,528.26

continued from page 1

big demand for good database books. To exploit these opportunities, however, some preparation is required. The question is, how should the aspiring writer get started? What is involved?



In this article we explore one author's experience in writing a technical book about Oracle. We will answer questions such as:

- How much time is required?
- How difficult is it *really*?
- How do I gain credibility?
- Where do I start?
- How does the edit process work?
- Is it worth the effort?

For Oracle DBAs, designers or developers interested in writing a book on Oracle (or any other technical subject), I hope you find my experience useful. Perhaps this information will save you some time, or alleviate some concerns. I do not claim to be an expert on publishing books, but I think my experience could be useful to you if you are considering writing a technical book.

How Much Time is Required?

As the dot-com bubble was starting to burst a few years ago, it became likely that in the (near) future I would have some spare time on my hands. It dawned on me one afternoon that I could use this extra time to do something that I had considered for some time—write a book on Oracle performance tuning. I wanted to write a book more for the beginner, as opposed to the very sophisticated books really intended just for the “guru.”

I have always enjoyed performance tuning and other database “mysteries,” and I wanted to show the novice performance tuner how to get started. I wondered, however, how much time would be required. Could I keep up with tough editing deadlines?

I was fortunate enough to have someone who could help me get started. Don Bursleson, editor of *Oracle Internals* and author of numerous Oracle books, was kind enough to offer some suggestions. He estimated that a first book would require an investment of 360 hours.

Don was very accurate. Here is the breakdown of my actual time spent writing and editing the book:

- 240 hours for the first draft
- 130 hours for the first set of edits
- 15 hours for answering questions during the second edits

You can see that my total time investment was 385 hours. This time was spread out over six months or so. This

means that I spent about 15 hours each week just working on the book.

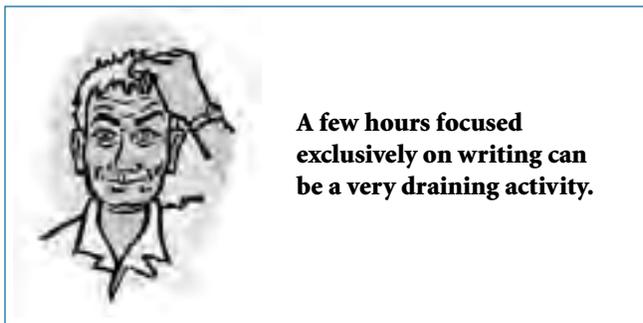
How Difficult is it, Really?

If you were to just look at the number of hours required, you might get the impression that the whole job is really pretty simple. Just 15 hours a week—how tough can that be?

Nevertheless, if you were to begin a book-writing project with the idea that it is going to be easy, you would be sadly mistaken. That 15 hours per week is a very tough 15 hours. To my surprise, I discovered that writing a book is far more difficult than writing technical papers.

Why is it so difficult? For one thing, the scope of the job is much broader than a single technical article. For a single article, you can become the subject matter expert without spending an enormous amount of effort. Now, compare that to writing about 20 technical papers! That roughly gives you an idea of the work involved.

Another big hurdle is the need for technical accuracy and timeliness. The book I wrote was about 400 pages, which is not really very large for a technical book. Nevertheless, for



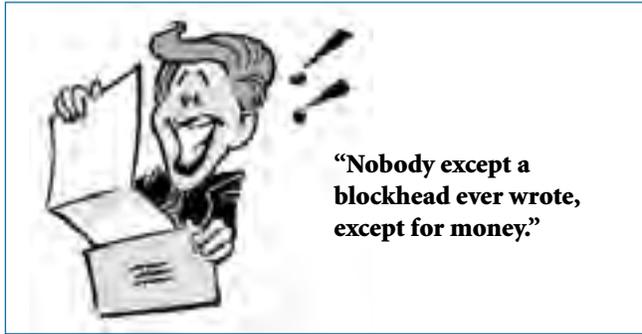
each of the 14 chapters, I needed to be technically accurate, clear, and up to date.

In the Oracle world of fast database releases, techniques and ideas that are technically accurate in one database release may be outdated in the next. I discovered that some processes and techniques that I used had become obsolete with Oracle 9i. For instance, my method of gathering statistics using the Analyze command is really on the way out. Thus, I found it necessary to research and document the newer method suggested by Oracle. There were other techniques that I had to research as well.

How Can I Gain Credibility?

Publishers are interested in selling lots of copies of the books they publish. They can more easily do this if they sign up an author who is already well known, or at least has some impressive credentials. These well-known authors are oftentimes the speakers you see at the large national Oracle conferences; they draw a huge crowd every time they speak.

Remember the saying, “Nobody except a blockhead ever wrote, except for money.”¹ This is true for publishers as well as writers. Publishing houses are not in the business of printing books for charity; they want to maximize the chances of success. That means either using an established name (e.g., Kevin Loney, Rich Niemiec, Thomas Kyte), or using an author who has otherwise proven his or her writing ability. In other words, the publisher wants someone with *credibility*.



Probably the easiest way to enhance your credibility is to write articles for technical journals. Besides the local user groups, they are numerous technical journals that are happy to review submissions—even from unknown writers.

Many popular Oracle authors have used this strategy. For instance, Don Burleson advertises that he has penned over 100 articles! In contrast, why would a publisher listen seriously to someone who has not bothered to publish even a single technical article?

Where Do I Start?

Before your project can start, the publisher will want to see a detailed proposal for your book idea. This proposal should include an analysis of the market for your book, the anticipated competition, plus detailed chapter outlines.

I wrote the original proposal for *The Art and Science of Oracle Performance Tuning* in December of 2001. Although very supportive, the first publisher I contacted already had contracted for a book on a similar topic. Naturally, they did not want to compete with their own book.

The second publisher I contacted, Curlingstone (subsidiary of WROX), was interested in reviewing the proposal. I emailed the book proposal to Curlingstone, and they hired a small group of highly respected DBAs² to review my proposal. (At the time, I didn't know that Curlingstone had signed up such a renowned team.)

The Contract

If your proposal fits the needs of the publisher, they will offer you a contract. They will possibly want you to consider certain changes, based on the comments from the reviewers. In my case, the senior editor wanted me to consider including material on ways to prevent performance problems. I agreed to include this whenever possible.

If you think that you will get rich from your book royalties, think again. As a first-time author, you will be given a small advance (typically a few thousand dollars), and a minor percentage of the net sales. Your advance will probably be spread-out over several months, and paid upon completing certain chapters. As part of your compensation, you will also be given a handful of free copies.

In return for your advance, you will be obligated to deliver chapters according to an agreed-upon schedule. You can simply email the chapters to the editor.

How Does the Edit Process Work?

Once you have completed your draft of the entire book, you will enter the editing phase, in which you work with

one or more editors assigned by the publisher. I found the editors a very reasonable and professional group of people. They did not claim to be super DBAs, but they were certainly very competent writers.

The publisher will hire a review team of experts in the field. They will review each chapter, and make suggestions on where the manuscript could be improved. You will then have to review their comments and make appropriate changes.

I found this stage to be very trying, and more difficult than I had anticipated. Consider this: Every paragraph of every page in your book is subjected to intense scrutiny by a group of people you probably don't even know. Although my reviewers were well qualified, it was no fun to have my writing put through such a rigorous critique.

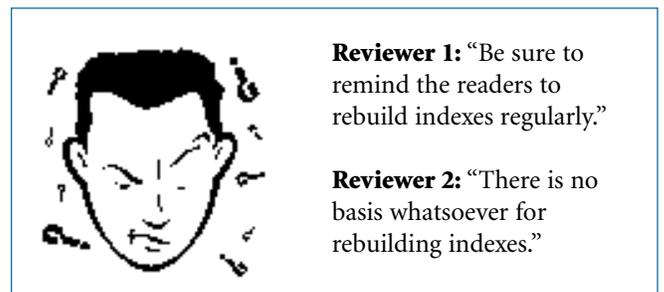
Of course, the final product was undoubtedly much better because of this review, but that doesn't make it any more fun! (A football player may become a better athlete by "running the gauntlet," but he probably doesn't look forward to it.)

The reviewers will be of varying skill levels and backgrounds. At times, they may become confused, or you may believe that their comments are completely wrong. For instance, the main technical editor suggested that 50% of the OS-specific material should be Windows. He wanted to substantially trim down the Unix section and add more new material on Windows. My experience tells me that Unix is a much more important platform, so I argued against his suggestion. (I ended up adding several pages on Windows, but did not remove any of the Unix material.)

At other times, the reviewers may contradict each other, and you will have to use your own judgment. I remember one section toward the end of a chapter. Here is what two reviewers said about the same section:

- **Reviewer 1:** "Eek!" [I think she didn't like this section.]
- **Reviewer 2:** "This is great. I will make my entire staff read this."

Since I was aiming for controversy in the book, I left this section unchanged. If nothing else, I felt it would get a good discussion going.



In a chapter about indexing, here is what two other reviewers said:

Once again, my theory was confirmed: If you get 10 DBAs in a room together, you will get 10 different "right" ways to do something.

I found the most helpful suggestions to be comments such as "I don't understand what you're saying." This type of comment told me that I had not really communicated well. I almost always took action to modify a section whenever a reviewer made a comment like this.

Keep in mind that the reviewers' comments are mostly *suggestions* and observations based on their personal experience. Rarely are technical issues so black and white that there is really only one correct answer. Of course, you would be wise to seriously consider most remarks; by doing so, I learned quite a bit from the reviewers. Naturally, errors in syntax or usage should always be corrected.

After the first stage of edits, things move rapidly. I was surprised at how quickly the editors incorporated and reviewed all the changes. There were actually only a few remaining technical questions, plus some formatting difficulties. (Did you ever notice, for example, exactly how the output is formatted in Sql*Plus for *numbers* versus text?)

Off to the Printers!

Once all the edits and reviewer questions have been addressed, there are just a few more minor issues to handle. You will have the opportunity to write/review a few preliminary sections, such as the *Introduction*, *Acknowledgments*, and *Dedication* sections. I found these to be very simple.

Finally, when you have answered all the editors' questions, and the proof-readers have blessed the work, it will be sent to the printers. There's really little for you to do at this time, except relax!

At this point, you will probably also receive another portion of your advance.



Afterword: Was It Worth the Effort?

Writing a technical book about Oracle databases is a formidable challenge. I found it to be one of the most difficult things I have ever done. Ironically, I could not have done it without the big "dot-com meltdown." I simply would not have had the time.

For me, the toughest part in writing *The Art and Science of Oracle Performance Tuning* was wading through the myriad of comments from the technical reviewers. Facing a huge amount of critical comments can really test your patience.

There really isn't any other way, however. No one has all the answers; everyone makes lots of mistakes. This means that an extensive review process is necessary in order to build a good product.

Now that it's all over, I can look back and smile. I have learned a lot more about Oracle, and some of my faulty ideas have been corrected. (Not all, though, I still have a few more.)

Now, getting back to those reviewer comments . . . ▲

Chris Lawson is a DBA consultant and author of The Art and Science of Oracle Performance Tuning (Curlingstone). He specializes in performance tuning of data warehousing and financial applications. Chris is also the editor of the online magazine, The Oracle Magician, available at www.OracleMagician.com.

Notes

- ¹ Dr. Johnson.
- ² Steve Adams, Thomas Kyte, and Jonathan Lewis [C1].

LECCO SQLExpert

SQL with a Higher IQ

LECCO SQL Expert automates the task of performance tuning and optimization of database applications by providing developers and DBAs with expert knowledge of SQL. Through its proprietary Feedback Searching engine, it provides optimum SQL statements—guaranteed to provide the correct results. No guesswork or hopeful suggestions, statements are actually rewritten.

Join the ranks of the experts and download a FREE evaluation copy today!

- Oracle
- Sybase
- MS SQL Server

Where the experts go for performance.

LECCOTECH
www.leccotech.com

That's right...
no more sleepless nights!

- Monitoring and Support
- Customized Solutions
- Free monitoring up to 24/7*

* The Fine Print: With minimum professional services commitment. See website for details.

MISSION CRITICAL 24/7

ORACLE PartnerNetwork CERTIFIED PARTNER

510.352.7300 • 866.352.7300
info@mc247.com • www.mc247.com

NoCOUG

Northern California Oracle Users Group

Membership Application for 2003

Your membership extends from January 1 to December 31, 2003

Code: NC112001-_____

Name:		
Title:	Work Phone:	
Company:	Mail Stop:	
Address:		
City:	State:	ZIP:
Email:		
URL:		

Do you want your mailing information made available on a third-party mailing list? YES NO

Do you want your email address added to the NoCOUG only Distribution List? YES NO

Oracle Product Profile

Please tell us a little about the systems and products you use most often:

Computer platform: _____ RDBMS version: _____ OS: _____

Please check each product used:

- Oracle Developer® Oracle Designer® Oracle Discoverer® J Developer BC4J®
 Oracle Applications: *Rel# _____ Web-DB/Oracle Portal® Oracle Application Server

Membership Dues

Membership Type	Includes	Qty	Rate		Amount
NoCOUG Individual Membership	Quarterly newsletter, roster and one meeting attendee		70.00	=	
NoCOUG Corporate Membership	Six individual memberships, plus four additional newsletters		450.00	=	
	Total Membership Fee Enclosed				

NoCOUG and the International Oracle Users Group (IOUG) are alliance partners. We encourage you to learn more about IOUG by visiting the IOUG website at <http://www.ioug.org>. NoCOUG members get a \$15 discount on IOUG membership for 2003. To get the discount, choose Northern California Oracle Users Group in the list of user groups when filling out your IOUG membership application.

Payment

On site NoCOUG Memberships **must** be paid by check. Online registration is also available at www.nocoug.org. Enclosed check # _____. **Make checks payable in U.S. Dollars to: NoCOUG, P.O. Box 3282, Danville, CA 94526**

Signature: _____ Date: _____ NoCOUG Tax ID: 94-3237851

Being Prepared When Disaster Strikes

Steve Lemme, Computer Associates

Preface

The global nature of eBusiness now contributes a new list of concerns for today's company's that now must consider new impacts that could affect their enterprises. The demands of global eBusiness electronic commerce along with each standard work week carries a challenge of new data, objects, and business applications to support. Changes in businesses are being further driven by the impact of business-to-consumer (B2C) processes and business-to-business (B2B) transacting 24 hours a day, seven days a week.

As these new eBusiness exchanges become more embedded in the business process, planning, administration, and support become critical components in any new business initiatives. Consequently, service level agreements and other binding performance contracts have now become a defining metric in most recovery/continuity strategies.

Here are a few facts to analyze your company's Disaster Recovery Plan (DRP) maturity level:

The top 5 causes of data loss¹ are contributed by:

- Hardware or system malfunction - 44%
- Human error - 33%
- Software program error - 14%
- Viruses - 6%
- Natural Disasters - 3%

A University of Texas study found:

- Of companies that suffered a catastrophic data loss, 43% never reopened and 51% closed within two years. Only 6% survived.
- Of those firms hit by the World Trade Center bombing, 50% of the businesses without a DRP were out of business within two years.

And this time the impact of the terrorists acts of 9/11/01 tragedy are still being assessed. Sadly many employee's lives, as well as businesses, were lost. Some companies were prepared while others were not. The unfortunate events of that day have once again exposed to need for companies to be prepared for when disaster strikes. Business critical databases, systems and networks have become so intolerant of disruption or outages, that many companies need to go back to the drawing board and re-architect so that any disruptions do not occur or

are made transparent to the user. Diminishing maintenance and recovery windows challenge most long-standing guidelines and require outside-the-box thinking to rationalize and deploy new solutions.

A Disaster Recovery Plan (DRP) provides documentation and procedures to follow in the event of a major outage or disaster situation. A disaster could be a fire, tornado, or any unanticipated event that makes the computers or networks unusable or compromises the integrity of the hardware, data, or software on the systems for any extended period of time. It is the responsibility and corporate obligation of each employee to strive towards constant prevention of disastrous effects by periodically reviewing recovery and continuity situations so as to match existing and new business needs.

Disaster Recovery Planning

A Disaster Recovery Plan, sometimes called a Business Resumption or Contingency Plan, is a series of pre-planned steps which serves as a guide through the process of recovering a system, data center or business in the event of a major hardware or site outage. The plan, and all the material required to execute it, is stored off site at a safe location that would not be impacted in a disaster. A Disaster Recovery Plan performs two functions: 1) to aid in the recovery of a real disaster; 2) to help prevent disasters from happening. Even in the event of a disaster, proper prevention efforts can help minimize the effect of the disaster on financial assets, customer satisfaction, legal obligation, and corporate image.

Cost of Downtime

The financial impact of downtime can be staggering. As reported by Contingency Planning Research² the average hourly impact per industry is:

- \$6.45 Million -Retail Brokerage
- \$2.6 Million - Credit Card Sales Authorization
- \$113,750 - Home Shopping Channels
- \$90,000 - Catalog Sales Centers
- \$89,500 - Airline Reservation Centers
- \$41,000 - Cellular Service Activation
- \$28,250 -Package Shipping Service
- \$25,250 - On-line Network Connect Fees
- \$14,500 - ATM Service Fees.

“... Enterprises that today tolerate two-day recovery-time objectives will see that horizon diminish to one day or less by year-end 1999 ...”

—D. Scott, Gartner Group, 2/24/98

Disaster Recovery: Weighing Data Replication Alternatives

Coupled with the primary financial impact of downtime are other attributed losses a percentage of company's experience³:

- Lost Productivity - 88.9%
- End-user/management dissatisfaction - 87.1 %
- Customer dissatisfaction - 66.9%
- Overtime - 59.3%
- Lost revenues - 41.8%
- Lost transactions - 34.4%
- Lost customers - 23.1%
- Penalties or fines - 7.6%

Assumptions

Part of the success of a Disaster Recovery Plan is in its simplicity. Therefore, there are certain tasks you will assume will be performed by other parties either within or outside of the company. To increase the effectiveness of the DRP, these issues should be considered prior to a disaster to

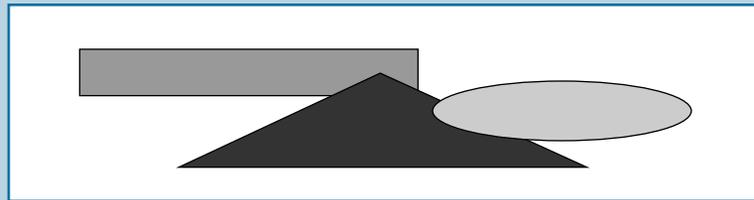
ensure rapid and effective responses.

Facilities - Facilities to be used in the event of a disaster should be considered before any computing function disaster. Initially there is the need to have a pre-arranged recovery location. Departments should consider what resources would be needed in the event of a disaster and where to find them. Issues like square footage, number of desks, hardware, should be investigated and documented in advance.

Project Priority Directives - Although situation specific priorities will arise when managing a crisis situation, management should be involved in pre-determining priorities for a disaster situation. The most critical business functions must be considered and documented ahead of time so that the highest priority items will be addressed appropriately. If not available, do what you can to resume business functions of projects in whatever order can be determined, depending on the state of emergency.



TECH TIPS



CRM: Reining In Unrealistic Expectations

“While Customer Relationship Management (CRM) is a popular topic in marketing and data warehousing, within the CRM value set there lies the seeds of a consumer backlash. Consumers will eventually figure out what is happening and rebel against the institutions which employ CRM in the wrong way.”

The paragraph above is just the beginning of an interesting article on Customer Relationship Management by popular NoCOUG speaker Michael Scofield. Published in *DM Review*, December 2002 (<http://www.dmreview.com/master.cfm?NavID=193&EdID=6082>), the ezine, *DM Review*, calls itself the “Premier Publication for Business Intelligence and Analytics.” You’ll find it at www.dmreview.com where current topics are:

The Financial Impact of Business Analytics: Build vs. Buy

By Henry Morris

When considering whether to build or buy an analytic application, the approach adopted depends on the problem to be addressed, the level of skill within the organization, and the availability of packaged solutions.

Nontechnical Infrastructure for BI Applications

By Larissa Moss

Business intelligence is about an enterprise architecture solution to the decision support chaos that exists today. Because it is a cross-organizational initiative, a nontechnical infrastructure is of critical importance. ▲

Security - Functions involving the securing of disaster and/or recovery areas are assumed to be handled by Corporate Security, unless directed otherwise. DRP teams and all involved in the recovery of a disaster are required to work with any directives issued by Corporate Security throughout the disaster.

Plant Services - Plant Services provides services such as environmental control, communications, wiring, phone services.

Human Resources - It is assumed that Human Resources (HR) will act as necessary to aid in locating temporary and/or permanent replacement personnel and providing assistance to current employees and ensuring their personal safety. It is also assumed that the majority of personnel will survive the disaster.

Legal Consultation - It is assumed that the company will provide adequate legal consultation and counseling should it become required. The staff will work as closely to the guidelines as possible to avoid possible legal conflict.

Vendor Disaster Recovery Commitments - Suppliers will be responsible for providing disaster recovery plans for their equipment.

A Disaster Recovery Plan Is Comprised of Four Parts:

1. Business Continuity Planning

The idea of a Disaster Recovery Plan is to perform two functions: one is to aid in the recovery of a real disaster, the second is to help prevent disasters from happening. Even in the event of a disaster, proper prevention efforts can help minimize the effect of the disaster on financial assets, customer satisfaction, legal obligation, and corporate image.

During the design of computing systems, protection against various failure scenarios should be taken into account. Fire prevention techniques, local codes, and company insurance requirements for data centers should not be overlooked. It is important to have a solid backup and recovery strategy. Data recovery is key to effective business resumption.

2. Risk Management

For a Disaster Recovery Plan to be feasible, all the risks need to be identified and categorized. In the event of a disaster, the DRP will help guide toward minimizing or eliminating those risks hindering the goal of complete recovery. Disaster avoidance through Risk Management minimizes the likelihood of preventable disaster potentials by deploying appropriate safeguards such as: education, security, backups, recovery strategies, fire suppression, water detection, power fluctuations or failures. Risk Management takes into account:

- Minimization of effect due to risk assessment
- Testing of DRP
- Cross-training of personnel
- Outside assistance

3. Project/Department Impact Assessment

No insurance policy can be bought to compensate the cost of lost data. Recovery strategies have changed with

the evolution of distributed computing. With distributed computing, the network has become an integral part of disaster recovery. In the past, the recovery of the data center and reconnection of users from terminals was the primary strategy. With the new decentralized client server organization, even more planning has to be done to identify critical networks and applications. This may involve an extensive multi-department site plan.

Power Influences - When evaluating power, all disturbances should be identified. Consider not only outages and weather disturbances, but also internal sources such as: microwave ovens, radio equipment, wiring, welders and large electrical motors.

Replacement of Equipment - The plan should address the replacement of inoperable equipment. One approach is to replace damaged machines with units acquired off the shelf. Another approach is to replace damaged machines from a pre-designated vendor within a contracted period of time.

Backup and Recovery Considerations / Databases - Backup windows have shrunk. Databases are now being used around the clock, and downtime is rare and expensive. Adding to this problem is storage size. Data warehouses are creating multi-gigabyte behemoths. Backups and recoveries have to be fast and reliable. For a database backup it's important that the names and locations of all the files that comprise the database tables, as well as the master control files, be tracked so that the appropriate files will be backed up for each database. As the database grows and changes, additions must be tracked, and scripts maintained and managed. Tape media contents and locations must be documented. There are several methods of backups to consider:

- Full system backup using file system tools
- Online backup using scripts
- Online backups using Third Party tools

Full offline backups are the traditional method for database backup. Full offline backups work well for small databases and those databases that are not governed by high availability. Full Physical backups can be combined with incremental backups to reduce backup time. An alternative is to combine full physical backups with regular archive logs to accomplish recovery until a given point in time. Using online backups opens the window to backing up larger databases and insuring availability. Online backups based on custom scripts involve interaction with the database to extract data and transaction log information while the database is online. Unfortunately, online backups involving custom scripts require a higher level of administrator expertise as well as management of the scripts. The last method involves using third party tools. Increased reliability and faster recovery is made possible by reducing manual intervention and automating tape volume management. "A zero data loss solution can be assembled today using Oracle technology coupled with third party offerings."

4. The Disaster Recovery Procedure.

Once a Disaster Recovery Plan has been designed, procedures for implementing these strategies need to be documented, personnel need to be trained, and the process needs to be tested. Rehearsing the actual plan is key to proper disaster planning. Rehearsing the plan helps identify potential pitfalls. While rehearsing, the results should be carefully documented and lessons learned should be integrated into the planning guide.

Incident Notification - This section walks through in short, concise, one-line action statements, the various steps which would need to be followed when an incident occurs. A plan should be created for a minor loss and a major loss situation.

Recovery Facilities - This section documents the locations, phone numbers, hardware, software and activation procedures of a pre-arranged recovery location. The type of recovery approach is usually determined by the criticality of the business supported. If required, a "Hot Site" can be set up. This "Hot Site" can range from a basic system where backup tapes can be restored to new systems to a complete duplicate of a company's data centers to resume business activities. Several vendors offer this service: COMDISCO (been acquired), Wang, HP, and SunGard.

System Recovery Plans - This section contains all of the step-by-step action points and procedures that would need to be performed at the recovery site in order to restore the Systems and the applications.

LAN Environment Recovery Plans - This section contains all of the step-by-step action points and procedures that would need to be performed at the recovery site in order to restore the LAN and its applications.

System Back-up/Restore Procedures - The backed-up data should be stored at an offsite facility and will be required for the recovery. This section also contains the procedures on how to restore the backed-up data as well as how to recall the offsite tapes.

Special Forms and Supplies Lists - The lists contained in this section will be used to gather any special forms, manuals and supplies that are kept at offsite locations for transport to the recovery site.

Software Recovery and Installation - This section lists all software needed for recovery.

Vendor Lists - This section lists all vendors, i.e., hardware, software, network, support services, which might need to be contacted in the event of a disaster.

Inventory and Required Information - This section contains miscellaneous documentation: detailed hardware, network infrastructure and backbone configurations, inventory lists and warrantee information.

Master Phone Lists - This section lists the business, home and pager numbers of all personnel who might need to be contacted in the event of a disaster.

DRP Document Location - This lists the location of the actual Disaster Recovery Plan.

Conclusion

By no means is this the definitive list for Disaster Recovery and Planning. Each site needs to access their

business needs and requirements and plan. At a minimum, this information can be used as a template to help start that process. These procedures ensure the continuity of business and engineering operations by providing access to pertinent data about projects, minimum system configurations with respect to hardware, software, and data, and personnel to aid in defining priorities necessary to allow for restoration of critical computing needs in a timely and orderly fashion. ▲

Steve Lemme is a VP of eDatabase Management Solutions <http://www.ca.com/manageit> for Computer Associates International and author of the new book "Implementing and Managing Oracle Databases." Mr. Lemme has over 10+ years of experience with mission-critical Oracle system architecture and eBusiness multi-tier enterprise computing. Prior to joining Computer Associates, Mr. Lemme has designed and managed mission critical database systems for several Fortune 500 companies where downtime was \$150,000/hr. Mr. Lemme also holds the position of President in the Arizona Oracle Users Group (AZORA) and is a Director for the International Oracle Users Group. Mr. Lemme is a regular speaker at IOUG, OracleWorld, CAWorld, HPWorld, EOUG, and UKOUG.

Disaster Recovery Resources

- ▶ Disaster Recovery Institute (314) 434-2272 St. Louis, MO
- ▶ COMDISCO Disaster Recovery Services, Inc. (800) 321-1111 Rosemont, IL www.comdisco.com
- ▶ Ontrack Data Recovery Services (800) 872-2599 Eden Prairie, MN Specialty: Data recovery from damaged disk and tape media
- ▶ Disaster Recovery Yellow Pages (617) 332-3496 Newton, MA Full guide to available disaster recovery products and services.
- ▶ SunGard Recovery Service Inc. (800) 247-7832 <http://recovery.sungard.com/>
- ▶ IBM Recovery Service (800) 599-9950 <http://www.brs.ibm.com/>

References

- ¹ *SunExpert Magazine*, July 1996, pg 46
- ² Contingency Planning Research
- ³ *HP Professional*, November 1995, pg 34
- ⁴ *Oracle Corporation Database Disaster Recovery Planning, An Oracle Technical White Paper*, Jan. 2001

This article is excerpted and reprinted with permission from the IOUG Live! Conference, April 2002.

The Art of Creating Subset Databases

Dale Edgar, Net 2000 Ltd.

It all begins, innocently enough . . .

Mgr: *The developers are on the warpath again.*

DBA: *Hmmm... they do seem moody, what's wound them up?*

Mgr: *Development databases, specifically the shortage of development databases.*

DBA: *(edging away) That's an interesting problem you have there, please do let me know how it works out.*

Mgr: *Nice try, wise guy, it's now your problem—they need three more instances.*

DBA: *We only have the disk space for half a full size copy.*

Mgr: *We'll then make some small ones for them.*

DBA: *Do you have any idea how hard . . .*

Mgr: *I have absolute confidence in your abilities. Be sure to let me know how it works out.*

Subset Databases

So there is no escaping it, no more disk space can be purchased and copies of a rather large production database are required for development. Small databases which contain a subset of the production data are the obvious solution. Such databases must be far smaller than the original and yet still form a self-contained system composed of data that is relevant, interrelated and complete. Building subset databases can be a daunting experience. Get it right and the extra database resources will dramatically speed up the development and testing cycle. Get it wrong and it is possible that the development process will be compromised. What follows is a distillation of experiences involved with the creation of subsets of Oracle schemas; the issues which will be met and some practical tips on how to approach the problem.

Why Subset?

Fundamentally the problem lies in the ever-increasing size of databases. Once a database exceeds a certain size it becomes very expensive to provide full size copies of the original database. Yet development, testing and training must continue. Many organizations resolve the problem by creating far fewer copies of the production database than are really needed (often only one) and then ask the development and testing teams to share its usage. Needless to say, this approach quickly becomes an exercise in diplomacy. The trouble soon starts: the developers collide with one another, the testers overwrite each other's data and everybody fights for time slots.

Databases required by test or development teams rarely need to be full size—in fact full size databases can be a drawback. Often, all that is required is a smaller version of the database that faithfully replicates the structure and content of the larger database. Truth be known—developers, testers, and trainers don't like to work on full size copies of production databases as it slows them down. They also hate sharing. Development and testing teams need small copies of the production database but it is not good enough to simply populate the smaller database with a sample of data taken randomly from each table. To do so might provide a database smaller than the original by the desired amount, but the data would be meaningless. For example, such a database might have invoices without customers, customers with no addresses, and so on.

Manually extracting a set of data that is meaningfully interrelated without taking too much, leaving any out or having duplicate records is surprisingly hard to do. Unless the schema is extraordinarily simple, PL/SQL procedures for creating referentially correct subset databases quickly become very complex and an appropriate development time must be budgeted for them. There will be a development effort and usually it will not be trivial.

Be aware, that due to the complexity of the task, manually created subsets are almost always a compromise with some foreign key constraints left disabled or some tables containing rows that do not have supporting rows in other tables.

First, Plan Your Subset

It cannot be over emphasized that, before you begin work developing the extraction routines for a subset database, you should have a clear idea of what it is you want from the subset.

The first questions you will want to ask are: "who will be using the database" and "what information do they need to see in it." These are important issues, and the answers will largely define the content and focus of the created subset. Usually a primary table is specified and the subset is focused around this table. The primary table is also known as the "pivot" table. A base set of data rows is retrieved for the pivot table and the remaining tables in the schema are populated with rows which are related to the rows contained in the pivot table. Because the rows populating the remaining tables in the subset are directly or indirectly derived from the pivot table, the subset database will be populated with data that is referentially relevant and interrelated.

An example

Assuming the development team was implementing improvements to the invoicing system, the INVOICE table

might be a logical choice for a pivot table. Taking 10% of the INVOICE table could provide the base data and the CUSTOMER, INVENTORY and INVOICE_LINE tables would be populated with data relevant only to the rows currently selected for the INVOICE table. Carrying the concept onto the next stage, tables related to the CUSTOMER table (an ADDRESS table for example) would then be populated with rows relevant only to the customers in the subset. In this manner, the content of the subset ADDRESS table is populated with data which is only indirectly related to the original INVOICE table but is still referentially meaningful within the subset.

Another example

The previous example assumed that the INVOICE table was to be the pivot for the subset. What if, after some discussion, it was determined that the development team really needed a subset database consisting of certain representative customer types. The previous approach would probably not guarantee that the CUSTOMER table would be populated with a desirable diversity of data. Rather than go through the exercise of adjusting the base data for the INVOICE table to somehow pull a desired range of customers, it would be far better simply to change the pivot table to CUSTOMER.

The CUSTOMER table could then be populated, (by whatever criteria is desired) and the INVOICE table would be populated with rows containing invoices relevant to the customers. All dependent tables for the INVOICE table would then be populated as before. The choice of the pivot table for the subset database can dramatically change the data content of the subset database.

Choose Your Base Data

Once the choice of pivot table has been made, the next step is to define the base data which forms the content of the pivot table. This issue requires careful consideration, as the end users will not be able to develop or test for situations which are unsupported by the data content of the subset schema.

An example

The development team wishes to develop a new invoicing system—but the subset database, although referentially correct, only contains examples of three industrial customers. Thus, their screens and reports are hard to test because they have only a minimal amount of data to work with. Ideally a subset database creation system should be able to use multiple criteria to populate a pivot table. For example, it should be possible to populate the pivot table with a sample of data (20% of all rows). It should also be possible to populate via a date range and then again via specific WHERE clauses in a SELECT statement. Of course, such multiple queries have an extremely high probability of returning some of the same rows. Such duplicates must be filtered out. This can be done by de-duplicating via an additional SQL run, or by the formulation of the base table creation statement in a carefully designed multi-join or SQL UNION statement.

NorCalOAUG Meeting February 27

The next Northern California Oracle Applications User Group Training Day will be held on February 27, 2003 in San Ramon. NorCalOAUG's goal is to provide users with a forum for sharing information and experiences on the selection, implementation and effective use of Oracle Applications and to learn about future directions in the Oracle product line. For up-to-date information, you can view NorCalOAUG's website at <http://www.norcaloaug.org>.

Relationship Mechanics

Once the pivot table is loaded, all new rows inserted into the subset database must be related somehow to the data in this "driving" table. If this is not the case, then the incoming data will not be referentially relevant. This requirement necessarily forms a parent-child relationship between the tables. The parent table with rows already in the subset needs supporting rows to be inserted in the child. Conversely the child table must obtain rows from the full size database but select only those related to the rows already in the parent. In other words, in order to populate the subset database table you will eventually have to construct a select statement which selects rows from the child table on the full size database and inserts them into the subset schema. The rows selected from the full size child table must be related to the rows already in the parent table in the subset database and the relationship is a standard join between one or more columns. The exact methodology of the select can be performed in any one of a number of different ways: via database links, temporary tables, etc.



TECH TIPS

A J2EE Community

Are you looking for more information about J2EE? You'll find a lot of resources on TheServerSide.com. On the home page, you'll find articles on topics such as an "Overview of XML and Web services support in J2EE 1.4." You'll also find links to discussion groups, reviews on application servers, book information, and events. Look for it all at <http://www2.theserverside.com/home/index.jsp>. ▲

Oracle FAQs

If you've got a question about Oracle, check out <http://www.orafaq.com/navdba.htm>. It's full of Frequently Asked Questions (and answers) on Oracle database administration issues such as:

- Oracle database internals
- Database administration
- Backup and recovery
- Performance tuning
- Common DBA procedures

An example

Consider a simple two-table system. The parent table on the subset database is the INVOICE table which has 100 rows. These invoices were pulled from the much larger table on the full size system and are all identified by a unique invoice number. The INVOICE table is the parent and it requires supporting rows in the child INVOICE_LINE table. These rows must be relevant to the 100 invoices already present. Pseudo code for the SQL insert statement would look as follows:

```
INSERT into SUBSET.INVOICE_LINE
(select * from FULLSIZE.INVOICE_LINE FS
where FS.INVOICE_NUMBER IN
(select INVOICE_NUMBER from SUBSET.INVOICE));.
```

Finding the Table-to-Table

Relationships

The previous section indicated how to populate the tables in the subset schema—but of course the relationship and join condition between the two tables must be known. Finding the relationships between tables forms a major part of the development effort in the creation of subset databases. The sections that follow provide some suggestions which will assist you in finding these relationships.

Follow the Foreign Keys

The foreign keys (if present) intrinsically define relationships between tables. Each one of the foreign keys defines a relationship which absolutely must be supported. If the rows in the foreign key parent table are not present in the subset database the foreign keys will not enable. The logic

of foreign keys can be reversed (for the purposes of creating a subset database) so if there is already data in the INVOICE table the foreign key relationship can be used to construct a population statement for the INVOICE_LINE table—even though the constraint really works in the opposite direction. It is usually possible to automate the creation of extraction SQL statements based on the foreign keys.

Column Similarity

Many schemas have the referential integrity implemented at the application level rather than in the foreign keys. One useful way to find table relationships is to find tables with the same column names. For example, it is probably a reasonable assumption that every table with a EMP_ID column can reasonably be expected to be related to each other via that column. Of course, this is not always the case—but it is an area worth investigating. Manually such an exploration requires a large number of selects on the USER_TAB_COLUMNS view.

ER Diagrams

Many table-to-table relationships should be determinable from an ER diagram—if it exists. Experience suggests that at many sites this is not a very useful option because the ER diagrams, if they exist at all, are usually not kept up-to-date.

Knowledgeable Person

One of the most useful ways of discovering table-to-table relationships is to ask a person knowledgeable in the schema. Unless this person has been specifically tasked with the job of helping you build the subset schema, you will find yourself using up goodwill at a rapid rate. A useful way of maintaining harmonious relations is to create a list of tables in the subset that are already populated (using rules you have figured out yourself) and a list of tables that have no rows. Then ask the schema guru: “Are any of these empty tables joinable to any of these non-empty ones?”

Subset Population Issues

There are a number of issues that need to be considered during the population of a subset database. These issues are some of the primary sources of trouble when creating subsets manually. Be aware of them—you will run into them and will need to plan your strategy accordingly.

The Duplicate Row Problem

It is quite common for a child table to have two or more parents. This means that the child will need to contain rows that support all of the parent rows in each parent table. The problem arises because the child table is generally loaded separately for each parent table. Of course, many of the rows required to support one parent table will be duplicates of the rows required to support another. The duplicate rows cannot remain in the child table—at the very least they will prevent you from enabling any primary keys. There are really only two available options to resolve this matter. The duplicate rows can be removed manually or the population SQL query has to be written to join against all parent tables.

Mark Your Calendars!

Our Spring Conference
will take place on
Thursday, May 15, 2003 at
Lockheed Martin in Sunnyvale.
You won't want to miss it!

The Circular Path Problem

When populating subset schema tables it is usual to focus in on only two tables at any one time—a parent table and a child table. This simplistic viewpoint can have some unusual side effects. The table-to-table relationships in most schemas are very complex. It is quite common to have circular paths of parent-child relationships. For example: table A needs supporting rows in table B, table B needs supporting rows in table C and table C needs rows from table A. Circular path relationships can have quite large numbers in the chain and because tables can require supporting rows from more than one child table one sees a complicated array of multiple interlinked table-to-table relationships. The effect of this complexity is that new rows may well appear in tables you previously thought you had completely processed. For example, you may well have populated table A and retrieved supporting rows for all of table A's children. Later on, some other table-to-table relationship will treat table A as a child and add a few more rows to it. The newly added rows in table A will not have supporting rows in that table's children. The arrival of new rows must be noted and the child tables must again be populated to contain supporting rows for the incoming table A rows.

The Self-Referential Relationship Problem

Tables can have a table-to-table relationship with themselves. This is most often seen in self-referential foreign key constraints. Theoretically there is no real problem with self-referential relationships—they are just another table join. However, in practice self-referential relationships cause a massive number of iterations to be performed. Since the parent and child tables are the same on each run, populating the child also populates the parent. Then the Circular Path Problem begins and the new parent record must then be processed for its own children. Depending on the database design these iterative chains can be tens of thousands of links long. Self-referential relationships should be identified and the normal SQL population mechanism should be replaced with a CONNECT BY PRIOR or similar functionality. This will prevent massive amounts of iteration as the self-referential relationship chains are followed.

A Production Process

Once a subset database has been created for the development and test teams you will probably find that they like it so much they want another. Of course, if things haven't changed all that much in the full size version you can probably just clone the existing subset or restore it from a backup somewhere. But what do you do if there have been major changes in data content on the original full size database? The point of having a subset database is that it should be a faithful, referentially correct small copy of the original. The only option is to create a new copy of the subset database. This is where time spent automating the process really pays off. A well-documented and designed subset creation procedure should reduce the number of DBA hours required to create a subset database. Investment of development time and effort in automating

the subset creation procedure pays for itself many times over in the long run. Schema changes are also an issue—if the schema structure changes in the full size database, then the subset databases derived from it will also need to change.

Ideally the creation process should be robust enough to cope with schema changes without major rewrites of the software.

Creating the Subset Schema DDL

The discussion up to this point has focused on the population of the tables in the subset database. There is another relevant issue—and that is the creation and sizing of the empty subset schema prior to population. The issue is more complex than it seems. It is not possible to do a simple export and import because the tablespace and storage clauses of the DDL in the subset would be the same size as in the source database. Typically subset databases need a dramatic reduction in the number of tablespaces and the removal or modification of the storage clause DDL. This can be done manually or via a number of freely available utilities.

Summary

Oracle database subsets provide many advantages—not the least of which are a much more versatile development environment and reduced resource costs. However, the creation of subset databases does require a certain amount of development effort and there are areas of complexity which serve to complicate the problem. Once the extraction routines have been developed, the creation of database subsets will become a production process. Time spent streamlining and automating this process will soon pay for itself. ▲

Dale Edgar, in many years of experience as a DBA, has built numerous test and development environments. He is one of the creative influences behind DataBee—software which provides an automated solution to the creation of subset databases. Dale can be reached at Dale@DataBee.com



RESOURCE CORNER

News for Nerds

If you are a self-described nerd, you may be entertained by Slashdot.org. Their tagline is “News for Nerds. Stuff that Matters.” The site is full of interesting news bits on science and technology topics. You can also do searches on popular topics such as security, handhelds, hardware, and space. You can check it out at <http://slashdot.org/>. ▲

The Education Conference for **ORACLE** Users by Oracle **Users.**

IOUG LIVE! 2003



Conference: April 27 - May 1, 2003 • Exhibits: April 28 - 30, 2003
Walt Disney World Dolphin Hotel • Orlando, Florida

Learn from Experience. Focus.

Save countless dollars and hours on the job with a single new tip or technique. **Network** with seasoned Oracle users.

Resolve performance issues with tuning techniques from the experts. Find **all the training you need** under one roof. Meet with today's leading Oracle vendors and get the answers you need. Learn backup and recovery strategies from the masters!

Bulletproof your environment by learning about the newest database security strategies. Get **Oracle9i** tips from qualified users.

Participate in advanced technical training through University Seminars, Technical Sessions, Partner Educational Forums, Networking Events, and Special Interest and Local/Regional User Group Meetings.

Customize your schedule with more than 250 technical sessions in 7 focus areas:

- Database Administration (DBAs)
- Architecture & Infrastructure
- Data Warehousing & Business Intelligence
- Application Server & Web Services
- eBusiness & Application Development
- ERP, CRM, SFA and SCM
- Career Development

REGISTER NOW! Visit www.ioug.org/live2003 to review complete conference information, registration details and session abstracts—or call +1.312.245.1579 for more information.

E-mail live@ioug.org to receive IOUG Live! 2003 updates!



Register now at
www.ioug.org/live2003
and enter Priority Code AF06



NoCOUG Winter Conference

Thursday, February 20, 2003
Session Descriptions

For up-to-date information, see www.nocoug.org

DBA TRACK

11:45–12:45

All About Oracle's Touch-Count Data Block Buffer Algorithm

Craig Shallahamer, *President, OraPub, Inc.*

Oracle introduced a touch-count based data buffer management algorithm to address the growing size, performance requirements, and complexities of current day systems. Data block buffers are no longer managed using a modified least recently used (LRU) algorithm. The touch-based algorithm introduced in Oracle 8i significantly reduces latch contention, forces buffers to earn their way higher up the main list, and allows for increased granular buffer performance optimization opportunities. This paper details Oracle's touch-count algorithm, how to monitor its performance, and how to manage for optimal performance (*Advanced Level*)

2:30–3:30

Managing Production Databases—Best Practices

Noel Yuhanna, *Senior Industry Analyst, Giga Information Group*

With the increased number of databases and their ever-growing size, manageability has become a challenge for most customers. In this presentation we look at some of the “best practices in the industry” to manage production databases. We also look at where DBAs put more effort and how they can become more efficient. We then proceed on ways to monitor databases including writing your own custom scripts. We also review some of the third party vendors offering monitoring products, and validate considerations for remote DBA solutions. Finally, we wrap up with some guidelines for improving database availability. (*All Levels*)

4:00–5:00

Eradicating Corruptions with Ease

Rama Velpuri, *CEO, ORAMASTERS, Inc.*

Recovering from block corruptions has always been a daunting task for any DBA. When a corruption is detected, Oracle marks the block as media corrupt. To a DBA/user it means that none of the read operations on the data block would be successful until the block is fully recovered! Introducing Oracle9i's incredible new feature, Block Media Recovery (BMR) for eradicating corrupted blocks in the database, with ease. This session discusses different types of

corruptions, and how to diagnose them proactively. It then describes the new feature, BMR, and shows how to fix block corruptions using RMAN thereby increasing availability. (*All Levels*)

DEVELOPER TRACK

11:45–12:45

Advanced SQL Application Tuning

Bert Scalzo, *Product Architect, Quest Software*

Attendees will learn successful techniques for identifying and correcting resource-consumptive SQL code. Quite often, system and database administrators are tasked with tuning the database to improve performance. However, 90% of the time the real culprit is poor application SQL code or ad-hoc queries. There are ever-improving tools and techniques for improving SQL performance, but tuning tools alone cannot solve all the problems—human intuition and insight most often make the most difference. Issues covered will include proactive versus reactive tuning, database tuning scripts, and SQL tuning tools. Attendees should be DBAs or senior developers familiar with Oracle database design and have some SQL tuning experience. (*Intermediate Level*)

2:30–3:30

Successful Dimensional Modeling of Very Large Data Warehouses

Bert Scalzo, *Product Architect, Quest Software*

Attendees will learn successful techniques for dimensional modeling of very large data warehouses using traditional Entity Relationship diagramming tools. While there are numerous new modeling conventions and tools, Entity Relationship modeling tools have proven best at real-world database design and implementation. This presentation provides the ground level, detailed recipe for the optimal dimensional modeling techniques of tables whose sizes exceed 500 million rows. Issues covered will include “Star Schema's,” fact and dimension tables, plus aggregate table design and population. Attendees should be DBAs or senior developers familiar with Oracle database design and any form of data modeling. (*Intermediate Level*)

4:00–5:00

Much Ado About Nothing (NULL)

Walter Schenk, *Principal Consultant, Solutech*

Unlike a great deal of programming languages, relational databases work with the concept of null values. This presentation first provides an overview of the concept of NULL and discusses the functions and operators available in Oracle to work with NULL values. It then shows how various programming languages interface with NULL values; good code practices as they relate to NULL are also presented. (*Beginning/Intermediate Level*)

NoCOUG Winter Conference

Thursday, February 20, 2003

Location: The NoCOUG Winter Conference will be held on Thursday, February 20, 2003 at the Oracle Conference Center, Redwood Shores (located at 350 Oracle Parkway). Sponsored by Oracle Corporation. See complete directions to the meeting location at www.nocoug.org.

- 8:00 **Registration and welcome. Refreshments served.**
- 9:00–9:30 **General Session.**
- 09:30–10:15 **Keynote: “How Oracle’s Complexity Breeds Simplicity”**
Craig Shallahamer, President, OraPub, Inc.
- 10:15–10:30 **Break**
- 10:30–11:15 **Welcome from our host and keynote from Oracle Corporation**
- 11:15–11:45 **Break**
- 11:15–11:45 **Morning Break**
- 11:45–12:45 **Parallel Sessions #1**
DBA Track: All About Oracle’s Touch-Count Data Block Buffer Algorithm
Application Developer: Advanced SQL Application Tuning
Oracle by Oracle: TBD
- 12:45–1:45 **Lunch Break**
- 1:45–2:25 **Roundtable Discussions for DBA and Developer Tracks**
- 1:15–1:30 **Break**
- 2:30–3:30 **Parallel Sessions #2**
DBA Track: Managing Production Databases—Best Practices
Developer Track: Successful Dimensional Modeling of Very
Large Data Warehouses
Oracle by Oracle: TBD
- 3:30–4:00 **Afternoon Break and Raffle**
- 4:00–5:00 **Parallel Sessions #3**
DBA Track: Eradicating Corruptions with Ease
Developer Track: Much Ado About Nothing (NULL)
Oracle by Oracle: TBD
- 5:15–?? **NoCOUG Networking and Happy Hour/No Host Bar**
- Cost: **\$40 admission fee for non-members. Members free. Includes lunch voucher.**



**Thank you to Oracle
our meeting sponsor.**

**View session
descriptions at
www.nocoug.org.**

RSVP online at <http://www.nocoug.org/rsvp.htm>

NoCOUG

P.O. Box 3282
Danville, CA 94526

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