

nocoug

JOURNAL

Official Publication of the Northern California Oracle Users Group

VOL. 16, No. 1 · FEBRUARY, 2002

\$15

NoCOUG is Thriving!

Did you know that the Northern California Oracle Users Group ended 2001 with over 450 members? We are consistently one of the premier Oracle user groups in the country, and well over 200 people attended each of our last two conferences. We've got a terrific new board of directors in place (see page 2), and an outstanding conference planned for February 21. There are a lot of great things in store for our members in 2002.



Renew Your Membership

Annual memberships for NoCOUG run from January 1 through December 31. This means that your membership has expired if you haven't renewed for 2002. Now is the time to renew your membership!

On page 7, you will find the Membership Application for 2002. 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 2002, including our upcoming Winter Conference on February 21 at Oracle in Redwood Shores. There is no charge for members and \$40 for non-members. That alone makes the \$70 Individual Membership price a bargain! ▲

TABLE OF CONTENTS

NoCOUG is Thriving	1
Renew Your Membership.....	1
User Groups.....	2
Don't Miss Our Winter Conference!.....	2
NoCOUG Board	2
Advertising Rates.....	2
Publication and Submission Format.....	2
President's Message	3
Tech Tips.....	4, 23
Meet the New VP!	5
NoCOUG Membership Application	7
Configuring SQL*NET and NET8.....	8
Sponsorship Appreciation.....	15
Treasurer's Report.....	15
Mary Lou and Bobbie Sue's Beginners' Guide to SQL* Plus	16
Resource Corner	23
Advanced SQL Tips	24
NoCOUG Winter Conference Session Descriptions.....	27
NoCOUG Winter Conference Schedule.....	28

—ADVERTISERS—

XIOtech	5
BMC Software	6
Quest Software	10
Promatis	12
Precise	12
Network Appliance	14
Database Specialists	18
Embarcadero Technologies.....	18
dbDoctor	20
Informatica.....	20
Seek Systems	22
LECCOTECH	22
IOUG.....	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 21 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, see our website at

www.nocoug.org. ▲

NOCOUG BOARD

President

Joel Rosingana, Independent Consultant
joelros@pacbell.net

Vice President

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

Treasurer/Secretary

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

Membership

Vacant Position

Webmaster

Vadim Barilko, Independent Consultant
vabarus@onebox.com

Journal Editor

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

Vendor Relations

Ganesh Sankar, Providian Financial
bgs2k2@yahoo.com

IOUG-A Representative and Past President

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

Members, At Large

Hamid Minoui, Fritz Companies
hamid.minoui@fritz.com

Darrin Swan, LECCOTECH
darrin@leccotech.com

Ganesh Sankar, Providian
bgs2k2@yahoo.com

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-A – International Oracle Users
Group of the Americas

- **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.

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, 2002. 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 © 2002 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.

The rates are:

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

Personnel recruitment ads are not accepted.



A NoCOUG New Year

by Joel Rosingana, President, NoCOUG

I want to wish all the NoCOUG members a very Happy New Year! I know with all the problems of last year, things seemed very dark. Even with the darkness, NoCOUG was able to have its best year ever. This could not have been achieved without the dedication of the 2001 board of directors and volunteers. Our success is measured by the ever-increasing attendance at the Quarterly Meetings.

A company does not do well without the firm and steady guidance of the CEO. That CEO would be our Past President Vilin Roufchaie. Vilin has guided us for the past two years. He has done an exceptional job in meeting and exceeding NoCOUG's objective of providing a quality program. I hope I can do half as well. Thanks Vilin! I should add that Vilin is stepping down because of term limits. A while back, the board placed a limit of two consecutive terms for the president and the vice president. Vilin will be staying on the board as the IOUG Rep and continue to provide his wisdom as past president.

Please check the NoCOUG Board listing at the front of the Journal. This will list our new board and officers for the coming year. The board contact information is listed here as well as on the website. Please contact us with any questions or comments. *Questions on how to volunteer are always welcome.* You will notice that most all the board has volunteered for another year. This means that we will have the experience level to achieve another banner year. Thank you.

We have had some changes to the board due to job and personal commitments. I would like to thank Ken Leonard for his hard work over the past years. He has given invaluable service and insight to the board. Due to increased job commitments and a new addition to the family, Ken has

asked to be retired for a while.

The next board member to step back has been a prime mover of the User Group for about 15 years. The net service date says 1987. I'm sure her organizational activity goes beyond that date. Year after year, Merrilee Nohr has been the backbone of NoCOUG. She has helped guide NoCOUG through the many changes needed to keep the organization current with the changing times. Merrilee did this while being on the board of IOUG and being IOUG's two term president. She has guided us all through some very complex times. Merrilee would like to take time to do those things she has put on hold all these past years. I know I speak for myself and the board – Merrilee will



Joel Rosingana

be sorely missed. And, Merrilee, because of you, NoCOUG is the strong and healthy organization it is today. Merrilee has agreed to stay on through initial registration. This will help us through the online conversion. This has been a special project of Merrilee's.

I would like to welcome our most recent members to the board. Hamid Minoui, Darrin Swan, Vadim Barilko, and Ganesh Sankar. The new board members not only provide help with ongoing tasks, they provide new and innovative ideas. This helps keep the organization responsive to the user community's needs. *Step up and share your thoughts.*

Let me close with a short synopsis of 2002's expectations. We have host and site commitments for the next four meetings. This allows us to spend more energy on the program, thus bringing in the speakers you want to hear. We plan to have an online registration system working in 2002. I'm hoping to have the credit card portion of the system working in the next month or two. Please contact me with any questions or comments. ▲

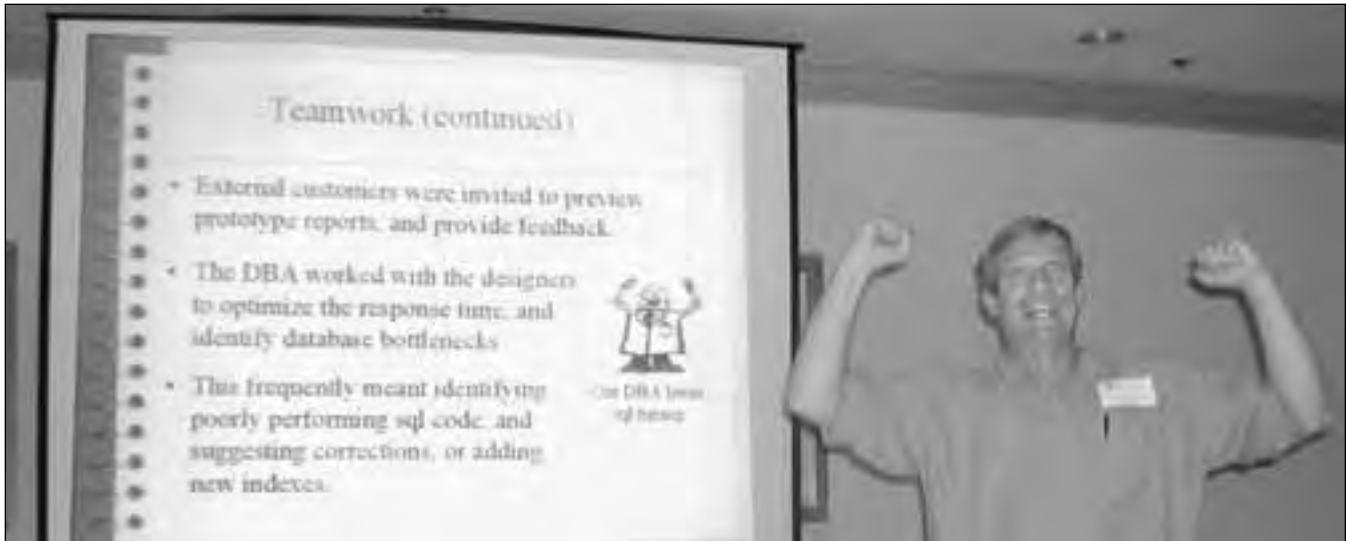
Launching NoCOUG Discussion Board!

A Note from the Webmaster

I am excited to have launched a Discussion Board on our website. It is a great tool for improving communication in our user group. We have special topics regarding conferences and for the main interests of NoCOUG users. Now you've got a way to continue your discussions after a conference. The Discussion Board is located at www.nocoug.org/discus.

Please let me know if you have any questions or ideas about improving the Discussion Board. You can write to me at vabarus@onebox.com.

— Vadim Barilko, NoCOUG Webmaster



Oracle DBA Chris Lawson, a popular presenter at NoCOUG conferences, demonstrated how much he loves SQL tuning! Chris spoke about data warehousing at our NoCOUG Fall Conference. Don't miss our upcoming conference on February 21!



TECH TIPS

Tuning Buffer Busy Waits by File

by Jeff Hunter, Sr. Database Administrator

<http://www.idevelopment.info/>

Most DBAs are aware of the view V\$WAITSTAT. This view (based on X\$KCBWAIT) reports the number of times an instance has had buffer busy waits on different classes of blocks since the instance was started.

Oracle also provides a companion view called X\$KCBFWAIT which duplicates the function of X\$KCBWAIT, but summarizes the waits by file id.

The following script will list the contents of X\$KCBFWAIT along with the datafiles. This script has been tested in Oracle7, Oracle8 and Oracle8i.

```

SET linesize 180
SET pagesize 9000

COLUMN filename      FORMAT a80           HEAD "File Name"
COLUMN file#         FORMAT 999           HEAD "File #"
COLUMN ct            FORMAT 999,999,999   HEAD "Waits"
COLUMN time          FORMAT 999,999,999   HEAD "Time"
COLUMN avg           FORMAT 999.999      HEAD "Avg Time"

SPOOL file_wait

SELECT
  indx+1          file#
  , b.name        filename
  , count         ct
  , time
  , time/(DECODE(count,0,1,count)) avg
FROM
  x$kcbfwait      a
  , v$datafile    b
WHERE
  indx < (select count(*) from v$datafile)
  AND a.indx+1 = b.file#
/
spool off

```

Meet the New VP!

Roger Schrag has been on the NoCOUG Board of Directors for two years, first as Webmaster (a position now taken by Vadim Barilko) and now as our newly-elected Vice President.

Roger has been a member of NoCOUG and a speaker at the quarterly conferences for many years. Since he thought that NoCOUG was a great resource for DBAs, developers, and managers in the Bay Area, he wanted to find out more about the organization and attended his first board meeting as a guest. "I was impressed by the group of people running NoCOUG and decided to join in," he says. "It's been a great way to meet people, and I've made some lasting friendships."

One of Roger's jobs as VP is to help put together high-quality meetings for NoCOUG members. You might wonder what someone gets out of giving so much of their time and energy to the Oracle community. Roger says he gets a lot of personal satisfaction knowing that he's helped put together a successful quarterly conference. He also enjoys sharing his knowledge with others by giving occasional DBA technical sessions at the conferences. Roger also contributes to the user community outside of Northern California. He joined the University Masters Class faculty for the IOUG-A last year. He enjoys the friendships he has made with fellow UMC instructors as well as helping students to be more successful in their Oracle careers.

Speaking of careers, Roger's technology career started early. "When I was 14, I started developing games for the TRS-80 computer. Those with really good memories may remember *Toxic Dumpsite* and *Spook House*." While creating these games didn't



Roger Schrag

earn Roger "huge sums of money" in royalties, it was sure better than flipping burgers like most of his friends!

He went on to study computer science at UC Berkeley, and his first job out of college was—believe it or not—with Microsoft. Though he and other employees enjoyed the occasional pizza and beer with Bill Gates, living

in the Pacific Northwest was a little too wet and Roger moved back to the Bay Area. That's where he found himself in 1989 as an employee of Oracle Corporation.

Back in those days, things were a lot different. Oracle had fewer than one thousand employees, and the finance department was the beta test site for Oracle Applications. "Back then," says Roger, "the other developers on the Oracle Financials team and

I keyed in UPDATE statements by hand in SQL*Plus to fix the company's financials when a journal entry got posted incorrectly."

Roger spent his initial three weeks of employment at Oracle in boot-camp—a three week intensive training where he and his fellow employees were immersed in Oracle life. They endured long days learning everything they could about Oracle technology, but were rewarded with the opportunity to meet senior staff members, including Larry Ellison. "Back in those days," says Roger, "I think Larry was more interested in fast cars than fighter planes or racing yachts." After a few years, Roger left Oracle Corporation and worked as a DBA at various San Francisco companies. In 1995, he became an entrepreneur and founded a consulting team called Database Specialists, Inc.

When he's not working with Oracle

Back in those days, things were a lot different. Oracle had fewer than one thousand employees, and the finance department was the beta test site for Oracle Applications.

databases, Roger enjoys bicycling, backpacking, camping, and hiking. Be sure to ask him about backpacking for a month in New Zealand on his honeymoon and having to trek through mud in a river at low tide! He and his wife have also traveled around the world to such places as India, Israel, Turkey, Denmark, and Costa Rica, just to name a few.

So, next time you see Roger, be sure to say hello! ▲

tech[®]
A Seagate Company



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.



NoCOUG

Northern California Oracle Users Group

Membership Application for 2002

Your membership extends from January 1 to December 31, 2002

Code: NC112001-_____

Name:		
Title:	Work Phone:	
Company:	Mail Stop:	
Address:		
City:	State:	Zip:
E-Mail:		
URL:		

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

Do you want your e-mail address added to the NoCOUG only Distribution List? YES NO

Oracle Product Profile

Please tell us a little about the systems & 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 2002. To get the discount, choose Northern California Oracle Users Group in the list of user groups when filling out your IOUG membership application.

Membership Dues

NoCOUG Memberships *must* be paid by check. 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

Configuring SQL*NET and NET8

By R. Matthew Adams, General Electric

This paper will describe the minimum configuration necessary to get NET8 up and running as fast as possible for new DBAs. Since this is very easy to do and would only require about two pages of documentation, it will also describe many of the optional features available in the NET8 product family.

What is SQLNET/NET8?

SQLNET/NET8 is the layer of software that allows application programs to communicate with Oracle databases and allows databases to communicate with each other. It utilizes a native networking protocol, such as TCP/IP or SPX or Decnet to accomplish this. The name SQLNET refers to the product as it was named when bundled with Oracle7 databases and clients. When Oracle8 was released, it was renamed NET8. Since Oracle7 is no longer supported, this paper will concentrate on NET8; however, the simple versions of the configurations described here should work for most older SQLNET implementations. The file names and parameter names are unchanged. The more advanced features described here may or may not work with SQLNET.

Unfortunately, configuring NET8 is one of Oracle's most poorly documented topics. The Net8 Administrators Guide adequately documents how to configure NET8, but it does leave much to be desired in terms of easy understandability and getting up to speed quickly.

The NET8 Assistant is a Java-based tool to help DBAs configure their NET8 environment. While it is adequate for most tasks, it is incapable of configuring some of the more advanced features of NET8. To configure these features, it is necessary to edit the configuration files manually using a text editor of your choice, usually "notepad" on Windows variants and "vi" on Unix. The problem is that once you hand edit the configuration files, you cannot later run the NET8 Assistant without risking losing some of the hand-entered configuration. Therefore, it is best to decide from the beginning which method of configuring NET8 you will use and stick with it. This paper assumes you will be using a text editor to manually apply changes to the NET8 configuration files.

Throughout this paper, I will be using the terms "client" and "server" to refer to various computers using NET8. In this paper, client will refer to any machine requesting a connection. This client could be an application program or another database. A database may act as a client by requesting a NET8 connection, such as when accessing another database via a database link.

The server is the machine hosting the database receiving the connection request. Note that any machine can be both a client and a server. A process called the listener runs on

this server, listening for database connection requests, which it then passes along to the database.

The configuration parameters listed in this paper are by no means a complete or definitive guide. For further explanation and additional configuration information, please refer to the NET8 Administrators Guide.

All configurations in this document were tested using client and server installations of NET8 version 8.1.6 under HP-UX version 11.

Style

In general, using a consistent style for your NET8 configuration files will pay off in the long run. There are no rules about how parentheses must line up or how the file must look overall. But, using a consistent style makes it easier to spot mistakes. Many a DBA has spent long hours staring at complicated NET8 configuration files only to discover that a parenthesis was in the wrong place. Also, only the first line of an entry should begin in column 1. Many versions of NET8 still have bugs that occur if open or close parentheses occur in column 1.

In general, NET8 configuration files are not case sensitive, but it is a good idea to pick one style of capitalization and stick with it throughout. The exception to this is host names, which should have been capitalized to match the actual machine names.

What Files are Needed to Make It Work

For the simplest possible configuration, only two files are needed to configure NET8 and get it working. The listener.ora file on the server configures the listener process. The tnsnames.ora file on the client lists all information that NET8 needs to know to find and connect to the correct database.

These are the only two required files, but it is usually a good idea to have at least a simple sqlnet.ora file on both the client and server. All NET8 configuration files are usually found in \$ORACLE_HOME/network/admin, but if not found there, NET8 will look for them in other places, such as /etc (under Unix). The locations where NET8 will look for these files vary among operating systems. If the \$TNS_ADMIN environmental variable or registry entry is

set to a directory, then NET8 will look for configuration files in that directory only.

Listener.ora

The listener.ora file tells the listener process how to listen for database requests and where to connect them once they arrive. It is possible to configure multiple listeners, with each listener having a different name, but in our examples here we have just the standard listener using the default name of LISTENER. Starting with Oracle8i, the listener.ora file is optional. If you want to use all of the defaults for the listener process, you can just start the listener with no listener.ora file. Oracle will automatically detect what databases are on the server and register them with the listener. I personally don't like this because I usually need to change some of these default values to fit my environment.

The listener.ora file usually resides in \$ORACLE_HOME/network/admin, but can be in other locations, such as /etc. You may also use the \$TNS_ADMIN environment variable to indicate the location of this file.

The three basic sections of this file are the listener address section, the SID list, and optional control parameters.

A simple example is below:

```
LISTENER=
  (ADDRESS_LIST=
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=mach1)
      (PORT=1521)
    )
  )
SID_LIST_LISTENER=
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=prod1)
      (ORACLE_HOME=/oracle/app/oracle/product/8.1.6)
    )
  )
LOG_DIRECTORY_LISTENER=$ORACLE_HOME/network/log
```

In this example, the first section, which begins LISTENER, configures the listener to run on the machine mach1 and listen for connection requests on TCP/IP port 1521, which is the default port for most NET8 implementations. The second section, which begins SID_LIST_LISTENER, lists the databases that exist on this machine and their ORACLE_HOME locations. If there were another instance on this machine, it would not be accessible via NET8 because it is not listed in the SID_LIST section of the listener.ora file. It is possible to have a listener listen on multiple TCP/IP ports if desired. Repeat the ADDRESS=(...) section, using an additional port.

```
LISTENER=
  (ADDRESS_LIST=
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=mach1)
      (PORT=1521)
    )
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=mach1)
      (PORT=1521)
    )
  )
SID_LIST_LISTENER=
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=prod1)
      (ORACLE_HOME=/oracle/app/oracle/product/8.1.6)
    )
  )
```

There are a number of optional parameters that can be added to this file to tell the listener how to do its job. I'll list a few of the more useful ones here; this is not a complete list. For that, see the NET8 Administrators Guide for a more complete list of parameters:

- log_directory_listener_name – put the log file for the named listener in this directory
- log_file_listener_name – specifies the name of the log file for the named listener

Tnsnames.ora

The tnsnames.ora file holds information that tells NET8 where and how to connect to a database. The entries in this file can be very simple or very complicated, depending on the features you want to use. A very simple form of a tnsnames.ora entry is:

```
prod1 =
  (DESCRIPTION =
    (ADDRESS =
      (PROTOCOL = TCP)
      (HOST = mach1)
      (PORT = 1521)
    )
    (CONNECT_DATA =
      (SID = prod1)
    )
  )
```

This entry tells NET8 to connect to a database using the TCP/IP protocol on machine mach1 using port 1521 and that the name of the database is prod1. The "prod1" on the very first line is called the *connection alias*. It is common for the connection alias to be the same as the SID you want to connect to.

For the vast majority of NET8 implementations, this is enough to get your client working. However, the tnsnames.ora can be configured to use some of the more advanced features of NET8.

Connect-Time Failover

It is possible to configure a tnsnames.ora connection alias to have multiple addresses. If multiple addresses are present, NET8 will try them in order until one succeeds or the list of addresses is exhausted.

```
prod1=
  (DESCRIPTION =
    (ADDRESS_LIST=
      (ADDRESS =
        (PROTOCOL = tcp)
        (Host = mach1)
        (Port = 1523)
      )
      (ADDRESS =
        (PROTOCOL = tcp)
        (Host = mach1)
        (Port = 1521)
      )
    )
    (CONNECT_DATA =
      (SID = prod1)
    )
  )
```

If desired, the *LOAD_BALANCE=yes* parameter can be added. This causes NET8 to attempt to connect to the addresses in random order while attempting to load balance the connections across addresses.

Prepare and Prosper.

SharePlex®: Do More Than Just Recover.

While you can't prevent disaster, you can stop it from affecting your bottom line. Hundreds of CIOs worldwide are prepared for the unforeseen using SharePlex® for Oracle as part of a successful business continuity plan.

SharePlex is more than disaster recovery. By giving you a real-time, accessible replica of your database, it solves other business-critical needs such as report offloading and load balancing.

So, you're not just investing in redundant systems that sit idle. You're investing in a solution that will recover your vital data when the unforeseen occurs and ensure peak performance every day — when disaster doesn't strike.

So don't just recover; prosper. With your vital intelligence safe, you can proceed confidently into your company's future. Learn how customers are thriving with SharePlex today at www.quest.com.

 **QUEST
SOFTWARE**
Business runs better on us

www.quest.com • 1.800.306.9328 (U.S.) • 1.949.754.6000 (outside U.S.)

```

prod1=
  (DESCRIPTION =
    (ADDRESS_LIST=
      (LOAD_BALANCE=yes)
      (ADDRESS =
        (PROTOCOL = tcp)
        (Host = mach1)
        (Port = 1523)
      )
      (ADDRESS =
        (PROTOCOL = tcp)
        (Host = mach1)
        (Port = 1521)
      )
    )
    (CONNECT_DATA =
      (SID = prod1)
    )
  )

```

Transparent Application Failover

Transparent Application Failover (TAF) allows an established Oracle connection to re-connect to an Oracle database upon failure of the connection. This failover is transparent to the user. This feature is probably most commonly used in conjunction with Oracle Parallel Server; however, it can be useful in other circumstances. One example would be if you had a replicated database environment where any of several databases could be used for queries. TAF is implemented by adding a new `FAILOVER_MODE` parameter to the `CONNECT_DATA` portion of a `tnsnames.ora` entry. TAF can take many different forms. The three documented in the *NET8 Administrators Guide* are connect-time failover, retrying a failed connection, and pre-establishing a backup connection. There are many other ways to set up TAF. The `FAILOVER_MODE` parameter varies based on the type of TAF in use. The general form of `FAILOVER_MODE` is:

```

(FAILOVER_MODE=
  (BACKUP= xxx)
  (TYPE= xxx )
  (METHOD = xxx )
  (RETRIES = xxx )
  (DELAY = xxx )
)

```

- **BACKUP** – specifies connection alias to use for backup connection. Used for pre-established backup connections.
- **TYPE** – can be `SESSION`, `SELECT`, or `NONE`. `SESSION` causes a failed session to reconnect. `SELECT` causes a failed session to reconnect and attempt to rerun the statement in process at time of failure. This has significant additional overhead to track statements being issued. `NONE` explicitly disables TAF.
- **METHOD** – can be `BASIC` or `PRECONNECT` – specifies how failover occurs. If set to `preconnect`, an unused connection to the backup server is established at the same time the primary connection is established.
- **RETRIES** – specifies the number of attempts to reconnect. Defaults to five if `DELAY` is set.
- **DELAY** – specifies the number of seconds to wait between connection attempts. Defaults to 1 second if `RETRIES` is set.

Connect-Time Failover in TAF

In this form, TAF connects to one server in the list. If that connection fails, then it will then attempt the other connection. In the example here, the `load_balance=on` parameter

tells `NET8` to pick the primary connection at random. The `failover=on` parameter tells `NET8` to try the second address if the first address fails upon initial connection attempt. Neither of these parameters is needed to implement TAF; they are included here to complete the example. Note that the alternate address is on a separate machine from the primary.

```

prod1=
  (DESCRIPTION=
    (LOAD_BALANCE=on)
    (FAILOVER=on)
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=mach1)
      (PORT=1521)
    )
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=mach2)
      (PORT=1521)
    )
    (CONNECT_DATA =
      (SID=prod1)
      (FAILOVER_MODE=
        (TYPE=select)
        (METHOD=basic)
      )
    )
  )

```

Retrying a Failed Connection

In this form, TAF will retry a failed connection over and over until it either succeeds in reconnecting or the retry limit is reached.

```

prod1=
  (DESCRIPTION =
    (ADDRESS =
      (PROTOCOL = TCP)
      (Host = mach1)
      (Port = 1521)
    )
    (CONNECT_DATA =
      (SID = prod1)
      (FAILOVER_MODE=
        (TYPE=select)
        (METHOD=basic)
        (RETRIES=20)
        (DELAY=10)
      )
    )
  )

```

Pre-Establishing Backup Connections

Pre-establishing a backup connection causes two connections to be created for the client at the time of initial connection. If the primary session fails, the session is automatically switched over to the backup connection. This configuration requires two entries into the `tnsnames.ora` file, as you can see from the example below.

```

prod1=
  (DESCRIPTION =
    (ADDRESS =
      (PROTOCOL = TCP)
      (Host = mach1)
      (Port = 1521)
    )
    (CONNECT_DATA =
      (SID = prod1)
      (FAILOVER_MODE=
        (BACKUP=backup_alias)
        (METHOD=preconnect)
        (TYPE=select)
        (METHOD=basic)
      )
    )
  )
)
backup_alias=
  (DESCRIPTION =
    (ADDRESS =
      (PROTOCOL = TCP)
      (Host = mach2)
      (Port = 1521)
    )
  )

```

```

)
(CONNECT_DATA =
(SID = prod1)
)
)

```

Sqlnet.ora

The sqlnet.ora file is an optional NET8 configuration file that holds general configuration parameters. These parameters define how NET8 should act. A few of the more common ones are listed below.

- log_directory_client
- log_file_client
- log_directory_server
- log_file_server

These four parameters define where sqlnet logs should go. As you would expect, the parameters with *directory* in the name are to indicate in what directory the log file should be placed. The parameters with *file* in the name designate the name of the log file.

- trace_level_client
- trace_directory_client
- trace_file_client
- trace_level_server
- trace_directory_server
- trace_file_server

Tracing is a facility built into NET8 products that generates information to be used in troubleshooting NET8.

Tracing levels can be set to four values: OFF (default), USER, ADMIN, and SUPPORT. The directory and file parameters indicate the directories where the resulting trace file should be placed and what they should be named, respectively. Turning on tracing will have an impact on the performance of your NET8 application and is not recommended unless necessary to diagnose problems. ADMIN and SUPPORT level tracing can consume disk space rapidly.

- sqlnet.expire_time = 10

This parameter is used by the server to determine how often to poll the client to see if the client is still alive. If the client has aborted, then the session in the server is killed and in-flight transactions are rolled back. If this parameter is not set, then such polling never occurs. Note that it is not checking for an idle connection; it is checking for an aborted application. If the client program has been idle for a long time, but is still running, the session will not be killed. Be careful setting this parameter. On some versions of NET8, it does not work well.

- names.default_domain=world

This parameter indicates what domain to attach to connection requests when looking them up in the tnsnames.ora file. For example, if some attempts to use a connect string of "xyz," and name.default_domain is set to "world" (a common value for this parameter), then NET8 attempts to find xyz.world in the tnsnames.ora file.

Be the HERO

ROI. Revenue generation. Extending usage. Issues every IT department worries about; especially now.

Be the hero with total and cost-effective enterprise solutions from PROMATIS INCOME Suite. Build business intelligence portals quickly with pre-configured Knowledge Bases. Control and organize knowledge in multiple languages. Integrate all components through the web. Know everything's working with wireless monitoring. Others have pieces; only PROMATIS has it all.

INCOME
Make Your Business Fly
get-INCOME.com

Business today is performance

Performance doesn't just happen.
Make it happen.

And make it *Precise*.

Brian Gilgeous 301-214-4272
bgilgeous@precise.com

Mike Horne 408-977-0692
mhorne@precise.com

Jeff Rudden 650-583-7999
jrudden@precise.com

Terry Prowant 925-980-2681
tprowant@precise.com

- names.preferred_servers

This parameter lists the names, addresses, and order of Oracle Names server processes to use to resolve connection requests, if Oracle Names is in use. See the section on the names.ora file below.

What Other Files are Used

There are many optional files that can be used in conjunction with the listener.ora, tnsnames.ora and sqlnet.ora files. Most of these are seldom used but they are included here for completeness. For more information regarding any of these, see the NET8 Administrators Guide.

Protocol.ora

The protocol.ora file is used for protocol specific parameters. All of the parameters are of the form protocol.parameter_name where protocol is replaced with the name of the protocol in question, usually TCP for systems using the TCP/IP protocol. These parameters affect all incoming database connections request and cannot be set on a per-instance basis.

- tcp.excluded_nodes - lists addresses of machines not allowed to connect
- tcp.invited_nodes - lists addresses of machines allowed to connect. This parameter will take precedence over the tcp.excluded_node parameter if both are present
- tcp.validnode_checking – YES/NO - (defaults to NO) indicates whether address checking should be enabled
- tcp.nodelay – YES/NO – (defaults to NO) Specifies no delay in buffer flushing within the TCP/IP stack

An example protocol.ora file is below.

```
TCP.VALIDNODE_CHECKING=YES
TCP.INVITED_NODES=(10.10.10.11, 10.10.10.12)
TCP.NODELAY=YES
```

Names.ora

The names.ora file is used to configure the Oracle Names product. Oracle Names makes NET8 information available to any node on the network, eliminating the need for local tnsnames.ora files on each client. When using Oracle Names, a connection request is sent to an Oracle Names server process that can reside locally or remotely. The Oracle Names server then looks up the appropriate information. This information is maintained by continuous replication between all the Oracle Names server processes or is stored in an Oracle database. Once the appropriate information for the request is located, it is sent back to the application requesting the connection to a database. The application can then connect to the database. The names.preferred_servers parameter should be set in the sqlnet.ora file if Oracle Names is in use. This tells the client where to find the Oracle Names server processes. Multiple Oracle Names servers can be listed in this parameter. A couple Oracle Names parameters are listed here. These parameters are set in the names.ora file. See the NET8 Administrators Guide for more information regarding configuring Oracle Names.

- names.addresses – Lists the protocols and addresses where the names server will listen for requests

- names.auto_refresh_retry – how often an Oracle Names server process should refresh its information from other Oracle Names server processes

Cman.ora

Oracle Connection Manager allows you to have multiple logical client NET8 sessions through a single connection to a database using Oracle's multi-threaded server option. This is most useful in environments where you must support a large number of simultaneous connections.

The Connection Manager processes may coexist on the same machine with the database or may exist on another machine. If you are using Connection Manager, you must make some parameter changes to the init.ora parameters that control your MTS environment. Specifically, the MTS_DISPACTERS init.ora parameter must have multiplexing turned on.

Also, the entry in the tnsnames.ora file will have to be modified to indicate that this request is going to be routed through a connection manager. For example, two tnsnames entries are listed below. The first one is the same as in an earlier example.

```
prod1 =
  (DESCRIPTION =
    (ADDRESS =
      (PROTOCOL = tcp)
      (HOST = mach1)
      (PORT = 1521)
    )
    (CONNECT_DATA =
      (SID = prod1)
    )
  )
```

In order to use Connection Manager, the tnsnames.ora entry must be modified to indicate that the connection uses a particular connection manager. This is accomplished by adding a *source_route=yes* parameter in the description and having multiple addresses. The first address is the address of the Connection Manager. The second address is the same as before.

```
prod1=
  (DESCRIPTION=
    (SOURCE_ROUTE=yes)
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=cman-host)
      (PORT=1630)
    )
    (ADDRESS =
      (PROTOCOL = TCP)
      (HOST = mach1)
      (PORT = 1521)
    )
    (CONNECT_DATA =
      (SID=prod1)
    )
  )
```

In addition, the Connection Manager itself has to be configured on the server it is running on. Some of the configuration parameters, which would be set in the cman.ora file, for Connection Manager, are below. Note that this is not a complete list. See the NET8 Administrators Guide for more information.

- cman – This parameter specifies the protocol and address for the Connection Manager process to listen for incoming requests.
- cman_profile – This is used to specify a large number of parameters that tell Connection manager how to work.

- Cman_rules – This parameter is used to specify rules about what clients are allowed to connect and what servers these clients are allowed to connect to.

Performance Considerations

Performance of applications using NET8 will depend on how much data NET8 has to send and retrieve from the database. Anything you can do to reduce that network traffic will help the application. Here are some things to consider.

- Using arrays fetches rather than individual fetches when returning multiple rows can dramatically reduce NET8 traffic.
- Moving database manipulations from the application to the database server by using stored procedures; packages and triggers can also have a significant impact.
- When using TCP/IP, NET8 packets may not be flushed to the network immediately. This is usually more pronounced when large amounts of data are being sent in a stream from server to client. If this occurs try setting the tcp.nodelay flag in the protocol.ora file.

There is one tnsnames.ora configuration parameter that can also have an impact on performance. The Session Data Unit (SDU) is a buffer where NET8 places data before it is transmitted. The default size of the SDU is 2048 bytes. According to the Oracle NET8 Administrators Guide, this size may be adjusted to be a multiple of the underlying transport size. For example, the normal size for an ethernet frame is 1500 bytes, so the SDU could be set to be a multiple of 1500. The SDU is specified in the description portion of a tnsnames.ora entry.

```
prod1 =
  (DESCRIPTION =
    (SDU=3000)
    (ADDRESS =
      (PROTOCOL = TCP)
      (HOST = mach1)
      (PORT = 1521)
```

NET8 is the glue that holds a network of Oracle databases together and allows them to communicate.

```
)
(CONNECT_DATA =
  (SID = prod1)
)
```

Unfortunately, at the time this paper had to be submitted, I had not had time to run performance tests to determine the effectiveness of making this change. I would advise performing significant testing to ensure that adjusting this parameter does not cause problems or performance degradation before implementing it on a production system.

Conclusion

NET8 is the glue that holds a network of Oracle databases together and allows them to communicate. All too often, configuring NET8 is a matter of trial and error. While this paper has not attempted to show you all details of configuring a NET8 implementation, you now have the knowledge to approach the configuration of NET8 with confidence.

References

Oracle8 NET8 Administrators Guide release 2 (8.1.6), primary author Deborah Steiner, copyright 1999, Oracle Corporation, Part No. A76933-01.

About the Author

Matthew Adams has been working with Oracle since the late '80s, when he started as a Forms 2.3 and Pro*C developer using Oracle 6.0. He has been working as a DBA for the past eight years and has spent time as a consultant or

employee at a wide variety of companies including AT&T, Mutual Of Omaha, the Naval Weapons Support Center, the Air Force Weather Agency, CSG Systems, Pamida, Sitel, and Midwest Laboratories. He currently works as a DBA for General Electric in the Appliances division.

Although every effort has been made to ensure that the contents of the paper are accurate, it is always possible for errors to slip through. Any and all comments/suggestions are welcome at rma_ocp@yahoo.com. ▲

This article is reprinted with permission from the IOUG-A Live! Conference, April 2001.

Is Your Data Available?

Worried about your data? Network Appliance®, an industry leader in enterprise data storage, has the solution. NetApp® storage appliances (NAS) provide dedicated database storage, delivering unparalleled reliability, availability and scalability. Our patented Snap technology replaces any data availability doubt with quiet confidence.

- SnapShot™ Safely backup your database in seconds
- SnapRestore™ Full data recovery in minutes, not hours
- Scale your database seamlessly with zero downtime

Network Appliance. The reliable, simple, and cost-effective data management solution.

ORACLE

Call 1-800-443-4537, or visit our website at www.netapp.com.



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 Joel Rosingana at joelros@pacbell.net.

Long-term full event sponsorship:

LOCKHEED MARTIN

CHEVRON

Long-term supplemental event sponsorship:

ORACLE

GENENTECH

Thank you! Year 2002 Gold Level Support Vendors:

- BMC Software
- Cast Software, Inc.
- Database Specialists, Inc.
- dbDoctor
- Embarcadero Technologies
- Informatica
- LECCOTECH
- Network Appliance
- Precise Software Solutions
- Quest Software, Inc.
- Seek Systems
- XIOtech Corporation

For information about our Gold Level Vendor Program, contact Ganesh Sankar, Vendor Relations, at: bgs2k2@yahoo.com.



TREASURER'S REPORT

Judy Lyman, Treasurer

Beginning Balance		
October 1, 2001		\$ 60,103.16
Revenue		
<hr/>		
Advertising	100.00	
Sponsorships	-	
Membership Dues	1,990.00	
Meeting Fees	1,600.00	
Vendor Receipts	3,000.00	
Interest	73.60	
Vendor Mailing	150.00	
Total Revenue		\$ 6,913.60
Expenses		
<hr/>		
Regional Meeting	9,408.25	
Journal	5,267.88	
Membership	2,561.92	
Administration	3,331.05	
Board Meeting	455.73	
Website	511.86	
Miscellaneous	20.00	
Total Expenses		\$ 21,556.69
Ending Balance		
December 31, 2001		\$ 45,460.07

Mary Lou and Bobbie Sue's Beginners' Guide to SQL*Plus

By Mary K. Platt, MND Service, Inc.

Bobbie Lambert, MND Service, Inc.

Abstract

This is a beginner's guide to SQL*Plus. The tables are built and they have been populated with data. How can you now use SQL*Plus to retrieve this data? This session will provide the information necessary for the retrieving and manipulation of existing table data using only SQL*Plus. The session will cover commonly used system tables, query syntax, ordering and grouping select results, operators, and conditions as well as SQL*Plus editing commands and functions. No SQL*Plus experience is necessary, but an understanding of Oracle table structure would be helpful.

Topics to be Discussed

- Why is it Important to Learn SQL*Plus?
- Editing Commands
- Common and Useful SQL Tables
- Queries
 - Simple Query
 - Where Clause
 - Order By
 - Having & Group By
 - Query Joins
 - Outer Joins
- Arithmetic Expressions
- Converting Values
 - To_Date
 - To_Char
 - Last_Day
 - Add_Months
 - Months_Between
 - Next_Day
- Decode
- Functions
 - Numbers
 - Characters
 - Groups
- Error Messages

Why is it Important to Learn SQL*Plus?

Every Oracle installation you will ever be involved with should have availability to SQL*Plus. While third party products are helpful and easy to learn and use, what happens when you find yourself at a site where they don't use third party products to access Oracle? What happens if you have to write your own SQL? If you learn SQL*Plus you'll find that the threat of ever being without a third party product to access Oracle is no longer valid. SQL*Plus is easy

to use and reliable. It is worth learning and may benefit you in the future.

Editing Commands

When in SQL*Plus, you would type your editing commands at the SQL> prompt. You would enter return to start a new line. When finished with your last line of typing, press an extra enter key to be able to view the buffer contents before running. Following are the most common commands you would use.

- l lists contents of current buffer
- ln lists contents of line = n in current buffer, i.e., l3 would list the current contents of line 3
- c change first occurrence in a single line, i.e., c/oldval/newval/
- i inputs a line below the current line position
- a appends text to end of current line
- del deletes line at the current position
- / runs the currently buffered statement

Common and Useful SQL Tables

While you could access the following SQL tables from a variety of tools, we believe it is important to make note of these common tables. The following objects, with the exception of dual, which is a table, are views into the data dictionary tables that hold all data pertinent to the database. Knowing what data these tables hold and being able to access that data can make your life much easier. The following are some, but by no means all, of the views that any user can access from SQL*Plus. Any views listed that begin with User_ show only information valid as the currently logged in user.

For example, you have a user account MPLATT and while logged into MPLATT you created a table named DUMMY. If you then log into Oracle as SCOTT and select all tables you own using User_Tables, you would not see DUMMY as you are now logged in under a different account name. It does not matter that YOU are the same person; view information is determined by the name of the Oracle account currently logged in, i.e., MPLATT versus SCOTT.

Hint: To see what Oracle account you are currently logged in as (assuming you may someday use several), you would use the SHOW USER command at the SQL> prompt.

- User_Tables - List of tables, views and synonyms that are created for the current schema
- User_Tab_Columns - List of all column definitions for tables in the current schema
- Table_Privileges - Listing of all tables that the current schema has the right to see and/or modify

- *User_Synonyms* - Listing of all synonyms created by the current schema
- *User_Views* - Contains most of the text used to create views in the current schema
- *User_Sys_Privs* - Listing of all system privileges granted the current schema
- *User_Role_Privs* - Listing of all roles that the current schema has been granted
- *User_Constraints* - Constraint information for each table in the current schema
- *All_Constraints* - Listing of all constraints for all of the tables the user can access
- *User_Cons_Columns* - Listing of columns associated with constraints listed in *User_Constraints*
- *User_Tab_Privs_Made* - Displays information about table privileges to other users for which the current schema is the owner
- *User_Sequences* - Displays information for sequences owned by the current schema
- *User_Source* - Contains source code for existing procedures, functions, packages, and package bodies
- *Dual* - Sys-owner table useful for testing functions or performing calculations

Queries

Of course, all the stored data information in the world won't help you if you don't know how to get it back out and view it. In order to extract data from SQL tables, you can write queries. Once again, you ask why you should learn to write queries when there are third party tools to write them for you. For one, what if you have a very complex query that was written with a third party tool that you want to modify, but no longer have that tool? Wouldn't it be great to be able to modify that query yourself? And of course, there's the chance that you might not always have a third party tool to do it for you. There is a specific syntax to query structure. This syntax follows:

```
SELECT ...
FROM ...
[WHERE ...]
GROUP BY ...
[HAVING ...]
[ORDER BY ...]
```

For our examples, we will use tables *WELL_INFO*, *STATE*, and *COUNTY*. The CREATE TABLE statements used for the examples are listed in Figures 1-A, 1-B, and 1-C. The structure (Figures 2-A, 2-B, and 2-C) and data within these tables (Figures 3-A, 3-B, and 3-C) follow. Data included in these tables are for example purposes only.

Figure 1-A

```
CREATE TABLE state
  (stateno NUMBER(2) NOT NULL CONSTRAINT pk_state PRIMARY KEY,
  stateab VARCHAR2(2),
  statenm VARCHAR2(15));
```

Figure 1-B

```
CREATE TABLE county
  (countyno NUMBER(3) NOT NULL,
  countynm VARCHAR2(20),
  stateno NUMBER(2) NOT NULL,
  CONSTRAINT pk_county PRIMARY KEY (countyno, stateno),
  CONSTRAINT fk_county FOREIGN KEY (stateno) REFERENCES state (stateno));
```

Figure 1-C

```
CREATE TABLE well_info
  (wellno NUMBER(3) NOT NULL PRIMARY KEY,
  wellnm VARCHAR2(15),
  stateno NUMBER(2) NOT NULL,
  countyno NUMBER(3) NOT NULL,
  cost NUMBER(6),
  welldate DATE,
  welltype VARCHAR2(1),
  CONSTRAINT fk_ctyst FOREIGN KEY (countyno, stateno)
  REFERENCES county(countyno, stateno));
```

Figure 2-A

```
WELL_INFO-----
Wellno  number(4)
Wellnm  varchar2(15)
Stateno  number(2)
Countyno number(3)
Cost     number(6)
Welldate Date
Welltype varchar2(1)
```

Figure 2-B

```
STATE-----
Stateno  number(2)
Stateab  varchar2(15)
Stateab  varchar2(2)
```

Figure 2-C

```
COUNTY-----
Countyno number(3)
Countynm varchar2(20)
Stateno  number(2)
```

Figure 3-A

Table = WELL_INFO

```
WELLNO  WELLNM          STATENO  COUNTYNO  COST    WELLDATE  WELLTYPE
-----
1        BUBBAS WELL        5         2  50000  18-OCT-1988  P
2        BIGWELL           5         2  56000  12-JUL-1986  P
3        DRY WELL          1         2  30000  17-JAN-1954  P
4        GOODWELL HUNTIN    2         1  35000  26-MAR-1967  P
5        HONEY WELL        3         1         1    04-JUN-1962  P
6        WELL BREAD        3         1  39525  05-JUL-1973  I
7        WELL & GRACE      4         2  37200  05-SEP-1971  I
```

Figure 3-B

Table = STATE

```
STATENO  STATEAB  STATENM
-----
1         AR       ARKANSAS
2         LA       LOUISIANA
3         NM       NEW MEXICO
4         OK       OKLAHOMA
5         TX       TEXAS
```

Figure 3-C

Table = COUNTY

```
STATENO  COUNTYNM  STATENO
-----
2         ALFALFA   4
1         HARDING   3
2         HIDALGO   3
1         ACADIA    2
2         ALLEN     2
1         ARKANSAS  1
1         WILSON    5
2         WINKLER   5
1         ADAIR     4
2         ASHLEY    1
```

Now, let's evaluate each section of the query statement.

➤ SELECT ...

- First you see the SELECT section. This is where you will put which columns you want to select from the table or view. If you choose to select all of the columns in the table, in the order in which the table was created, you would type `SELECT *` in the SELECT part of the statement. If you prefer to select only specific columns, i.e., `Wellno`, `Countyno` and `Wellnm` from the `WELL_INFO` table, you would type `SELECT WELLNO, COUNTYNO, WELLNM` in the SELECT part of your query. You can type the columns in any order you prefer. The order in which they appear in your SELECT statement will be the order in which they will be displayed when you execute your query.

➤ FROM ...

- Next you see the FROM section. You will put the name(s) of the table(s) and/or view(s) here that contain the fields you want to retrieve. For example, if you want to retrieve the fields `Wellno`,

Countyno and Wellnm from the WELL_INFO table, you would have FROM WELL_INFO in your from section, where WELL_INFO is the table where the fields requested can be found.

► WHERE . . .

- Following the FROM section is the optional WHERE clause section. This section enables you to restrict the rows that are returned from your query based on some type of criteria. For example, you only want to see the selected fields for records whose stateno equals 3. You would type WHERE STATENO = 3 in the where section.
- In our examples we are using the operator “equal to” (=) in our WHERE clause. Valid operators are:

=	Equal to
!= or <>	Not Equal to
>	Greater Than
<	Less Than
>=	Greater than or Equal to
<=	Less than or Equal to
Like	Matches a pattern
%	matches zero or more characters
_	matches one character
Not Like	Does not Match a Pattern
%	matches zero or more characters
_	matches one character
IS NULL	is missing a value
IS NOT NULL	contains a value, is not missing a value

BETWEEN >= first value AND <= second value

IN (list) value is part of a specific list, separated by commas, quote character strings or dates

NOT IN (list) value is not part of a specific list

- For multiple WHERE clause conditions you would use AND or OR to join your conditions

► ORDER BY . . .

- The ORDER BY is also an optional section. You can use it to control how the results of your query are displayed back to you. If we wanted to see the results that are returned to you from your query ordered alphabetically by Wellnm, you would put ORDER BY WELLNM in your optional order by clause.

If you now put all the query parts together to make a SQL statement, you have . . .

```
SELECT WELLNO, COUNTYNO, WELLNM
FROM WELL_INFO
WHERE STATENO = 3
ORDER BY WELLNM
```

You run the query by typing a semicolon at the end of the last line of your SQL statement, or by typing a forward slash (/) at the SQL> prompt. The “/” runs the statement that is currently in the SQL buffer. Remember, to view the contents of the current SQL buffer you can type “L” at the SQL> prompt. This will list the current buffer contents.



Helping You Get Things Done

When you need no-nonsense, straightforward Oracle expertise, call on Database Specialists. We'll roll up our sleeves and help you get things done.

- Oracle Applications
- System Architecture
- Database Design
- Production Support
- Performance Tuning
- Database Development
- Backup and Recovery Strategies
- Migrations and Upgrades
- Database Monitoring
- Part-time DBA Support

Call Us Today!

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



ORACLE | CERTIFIED SOLUTION PARTNER

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.

And our results are:

WELLNO	COUNTYNO	WELLNM
5	1	HONEY WELL
6	1	WELL BREAD

➤ HAVING & GROUP BY. . .

- What if you wanted to see the countyno for the state of Texas (stateno = 5) where there was more than one well per county? We would check for records *HAVING* multiple records for field countyno. We are *GROUPING* the records by countyno before returning our results, so we only have one record returned for each *GROUP BY* field. Your statement would look as follows:

```
SELECT countyno
FROM well_info
WHERE stateno = 5
HAVING count(countyno) > 1
GROUP BY countyno;
```

Result: COUNTYNO = 2

So far we have written only simple queries. Let's now look at a Join Query. In the following example, we are going to join all three example tables together so that we can retrieve fields from each table and display only one set of results. We will retrieve the Wellnm, Welldate and Cost from the WELL_INFO table. We want to also see the Countynm from the COUNTY table and the Statenum from the STATE table. Our joined query follows.

```
SELECT wellnm, welldate, cost, countynm, statenum
FROM well_info, county, state
WHERE well_info.countyno = county.countyno
AND well_info.stateno = county.stateno
AND well_info.stateno = state.stateno
ORDER BY statenum, countynm, wellnm;
```

WELLNM	WELLDATE	COST	COUNTYNM	STATENUM
DRY WELL	17-JAN-1954	30000	ASHLEY	ARKANSAS
GOODWELL HUNTIN	26-MAR-1967	35000	ACADIA	LOUISIANA
HONEY WELL	04-JUN-1962		HARDING	NEW MEXICO
WELL BREAD	05-JUL-1973	39525	HARDING	NEW MEXICO
WELL & GRACE	05-SEP-1971	37200	ALFALFA	OKLAHOMA
BIGWELL	12-JUL-1986	56000	WINKLER	TEXAS
BUBBAS WELL	18-OCT-1988	50000	WINKLER	TEXAS

➤ QUERY JOINS

- Because we did not select fields that were in both tables with the same fieldname, we did not need to prefix the column names with either an alias or the table name. If we wanted to see the countynm and statenum for each countyno in the county table, we would need to use either a table's alias or the table name. Let's first look at the error if we don't.

```
SELECT countynm, statenum
FROM county, state
WHERE stateno = stateno
SQL> /
WHERE stateno = stateno
*
ERROR at line 3:
ORA-00918: column ambiguously defined
```

We can retype this SELECT in two ways. We can assign an alias to each table as in Fig. 4-A or simply prefix each field with the name of table (Fig. 4-B). Aliases are nice, as sometimes table names can be long and would require extra typing. Both Fig. 4-A and Fig. 4-B would give the same results:

Fig. 4-A

```
SELECT countynm, statenum
FROM county A, state B
WHERE A.stateno = B.stateno;
```

Fig. 4-B

```
SELECT countynm, statenum
FROM county, state
WHERE county.stateno =
```

➤ OUTER JOINS

- Suppose you want to see all possible countyno(s) where statenum = 'TEXAS' and at the same time display the wellnm from WELL_INFO for current wells in this state.

You could write:

```
SELECT wellnm, countynm, statenum
FROM well_info, county, state
WHERE well_info.countyno = county.countyno
AND well_info.stateno = county.stateno
AND well_info.stateno = state.stateno
AND statenum = 'TEXAS';
```

Your results would be:

WELLNM	COUNTYNM	STATENUM
BUBBAS WELL	WINKLER	TEXAS
BIGWELL	WINKLER	TEXAS

The previous results show only one county in Texas (Winkler) because that is the only county that shows up with a record in well_info when you join the tables. To see all counties in Texas, regardless of whether they have a record in well_info or not, you would use the Outer Join symbol (+). That means if you don't find a record in the table with the (+) following it in the WHERE clause join, create a blank record for the join condition. Therefore, if we wanted to see all counties in Texas (stateno = 5), even if there weren't a record in the well_info table, we would use the (+) in the following manner with the results as shown:

```
SELECT wellnm, countynm
FROM well_info, county
WHERE well_info.countyno(+) = county.countyno
AND well_info.stateno(+) = county.stateno
AND county.stateno = 5;
```

Results:

WELLNM	COUNTYNM
WILSON	
BUBBAS	WELLWINKLER
BIGWELL	WINKLER

As you can see, you see both valid counties for Texas (Wilson and Winkler) even though there are no records in WELL_INFO for the county = WILSON.

Arithmetic Expressions

Valid arithmetic operators are +, -, * and /. You could use them in the following way to alter the cost or welldate fields in the WELL_INFO table.

- Add 200 dollars to the dollar cost stored in the COST field of the WELL_INFO table:
SELECT cost + 200 FROM well_info;
- Add 200 dollars to the dollar cost stored in the COST field of the WELL_INFO table then multiple that result by 1.5
SELECT (cost + 200) * 1.5 FROM well_info;
- Add 365 days to the welldate field in the WELL_INFO table
SELECT welldate + 365 FROM well_info;

- Minus 15 days from the current system date
SELECT sysdate - 15 FROM dual;

Converting Values

- To_Date – Converting a value to a date format

Let's assume we have a field named FIELDDATE in table FIELDTABLE which is defined as a VARCHAR2(8) and is in the format of YYYYMMDD with a value of 19990228. We want to convert this field's value to a true date format 5, and then write a statement to select the value and add one to it without any conversion, and then finally write a statement to select the date-converted value of this field and add one to it.

```
SELECT (TO_DATE(fielddate, 'YYYYMMDD')) FROM fieldtable;
Result Returned = 28-FEB-1999
```

```
SELECT fielddate + 1 FROM fieldtable;
Result Returned = 19990229. This result might imply that
if you converted the result to a date that Feb. 29, 1999
was a valid date, which it is not.
```

```
SELECT TO_DATE(fielddate, 'YYYYMMDD') + 1 FROM fieldtable;
Result Returned = 19990301. This result would be correct
because Feb. 28, 1999 + 1 day = Mar. 1, 1999.
```

- TO_CHAR – Converting a value to a character format

Let's assume we have field WELLDATE in table WELLINFO that is of type date. The default date format for returned rows in our database is DD-MON-YYYY. We have an application where we really would like to see the data printed as MM/DD/YYYY. We can convert the date to a character string formatted as MM/DD/YYYY by doing the following:

```
SELECT TO_CHAR(welldate, 'MM/DD/YYYY') FROM well_info;
If Date = 18-OCT-1988, converted character string would be
10/18/1988
```

- LAST_DAY – Display the last day for the month and year specified. If the current system date (sysdate) = 05-FEB-2001, find the last day of this month for this year.

```
SELECT LAST_DAY(sysdate) FROM dual;
Result would be 28-FEB-2001
```

- ADD_MONTHS – Add a specified number of months to the current date to find a new date. For example, assuming that WELLDATE = 26-MAR-1967, find the date 6 months in the past.

```
SELECT ADD_MONTHS(welldate, -6) FROM
well_info;
Result returned would be 26-SEP-1966
```

- MONTHS BETWEEN – Find the number of months between two dates, i.e., 18-OCT-1988 and 18-OCT-2000

```
SELECT MONTHS_BETWEEN('18-oct-2000','18-
oct-1988') FROM DUAL;
Result = 144 months
```

- NEXT_DAY – Find the date for the next specific day of the week, i.e., what is the date of the first Friday following April 21, 2001?

db Doctor
Empowering Database Administrators

Health Watch for Oracle

**AUTOMATED, PROACTIVE, SYSTEMIC
DATABASE REPORTING**

*Save time and
avoid pain with*

- 80+ Automatic Reports
- 90 Day Performance History Archive

visit us on the web, or
call for a free trial

<http://www.dbDoctor.net> (866) 323-6286

INFORMATICA
NOW YOU KNOW™

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

```
SELECT NEXT_DAY('21-APR-2001','FRIDAY')
FROM dual;
Result is 27-APR-2001
```

What is the date of the first Friday of the following month if the current date is 21-APR-2001?

```
SELECT NEXT_DAY(LAST_DAY('21-APR-
2001'),'FRIDAY') FROM dual;
Result is 04-MAY-2001
```

Decode

Syntax is: `DECODE(value,if_1,then_1,if_2,then_2,...,else_this)`

Change a value to another if certain conditions are met. In the table WELL_INFO we have a field named WELL-TYPE. If WELLTYPE is equal to P, we want to return a value in our select of PRODUCER. If WELLTYPE is equal to I, we want to return a value of INJECTOR. If neither P nor I apply, we'll just return the actual value found in the well-type field.

The SELECT statement follows (You'll notice a Heading of TYPE for the decode statement):

```
SELECT wellnm, welltype,
DECODE(welltype,'P','PRODUCER','I','INJECTOR',welltype) "TYPE"
FROM well_info
ORDER BY wellnm DESC;
```

WELLNM	W	TYPE
WELL BREAD	I	INJECTOR
WELL & GRACE	I	INJECTOR
HONEY WELL	P	PRODUCER...

What if you are trying to divide a number by another value that ends up being equal to zero?

```
SELECT 100 / 0 FROM dual;
ERROR:
ORA-01476: divisor is equal to zero
```

Let's try it again using DECODE to decode the zero value to be null, then again to 1. Assume FIELD1 in the example is equal to zero.

```
SELECT 100 / DECODE(field1,0,NULL,field1) val0,
100 / DECODE(field1,0,1,field1) val1
FROM table1;
```

Results are:

VAL0	VAL1
	100

VAL0 returns a NULL value, because 100 / NULL would have to return NULL as NULL is an unknown value, not a zero.

*It is important to learn SQL*Plus, and not to rely solely on third party end-user tools as your only method of data retrieval.*

Functions

► For Numbers

- **ABS (value)** – Returns the Absolute Value of value
- **CEIL (value)** – Returns the smallest integer larger than or equal to value
- **FLOOR (value)** – Returns the largest integer <= value
- **GREATEST (value1, value2)** – Returns the largest value of the values listed
- **LEAST (value1, value2)** – Returns the smallest value of the values listed
- **MOD (value, divisor)** – Returns the modulus after calculation for value
- **NVL (value, alternate_value)** – Substitutes alternate_value for value if value is null
- **POWER (value, exponent)** – Returns value raised to power of exponent
- **ROUND (value, precision)** – Rounds the specified value to a specific precision
- **SIGN (value)** – Returns 1 if positive value, 1 if negative value, 0 if value = 0
- **SQRT (value)** – Returns the square root of value
- **TRUNC (value, precision)** – Returns value truncated to specified precision
- **VSIZE (value)** – Returns the Oracle storage size of value

► For Characters

- **INITCAP (string)** – Converts the initial letter of each word in the string to a capital
- **UPPER (string)** – Returns the capitalized value of the entire character string
- **SUBSTR (string,start_column,length)** – Returns a specific section of string text
- **LOWER (string)** – Returns the lowercase value of the entire string
- **LENGTH (string)** – Returns the number of characters in a string including spaces

► For Groups

- **AVG** – determines average value of a group of values
- **COUNT** – determines the number of not null values for a column or total number of rows in a table
- **MAX** – determines maximum non-null value
- **MIN** – determines minimum non-null value
- **SUM** – determines sum of not-null values

For the values of the field COST in the table WELL_INFO in Fig.3-A, evaluate the group functions using the following SQL statement. You'll notice that COUNT(*) returns a value > COUNT(cost) since one of the records has a null value in the cost field. Therefore, it is not counted, whereas COUNT(*) counts all records regardless of whether fields have null values. Only non-null cost values are included in the AVG, MIN or MAX results.

```
SELECT COUNT(*), COUNT(cost),AVG(cost),MAX(cost),MIN(cost) FROM well_info;
```

Results are:

COUNT(*)	COUNT(COST)	AVG(COST)	MAX(COST)	MIN(COST)
7	6	41287.5	56000	30000

Error Messages

If you try to access the Oracle database and it is shut down to ALL users, you will see the following when trying to connect:

```
ERROR: ORA-01034: ORACLE not available
ORA-07429: smmsgsg: shmget( ) failed to get segment
SVR4 Error: 2: No such file or directory
```

If you try to access the Oracle database and it is shut down to regular users, but open to those with DBA privileges, you will see the following during your connection try:

```
ERROR: ORA-01035: ORACLE is available to users with
RESTRICTED SESSION privilege
```

If you were logged into the Oracle database from SQL*Plus, and Oracle was brought down during your session, you would see the following when you issued your next SQL*Plus statement:

```
ERROR:
ORA-03114: not connected to ORACLE
```

Conclusions

In this paper, we have discussed some SQL*Plus information and commands that can be used to help you retrieve and format your data from the Oracle database. As with most programming techniques, there is usually more than one way to accomplish the same end result. Once again, it is important to learn SQL*Plus, and not to rely solely on third party end-user tools as your only method of data retrieval. Someday, you may find yourself without your

third party tool, unlike SQL*Plus, which seems to remain a constant part of Oracle's tools.

References and Credits

Documentation, Oracle Corporation
Oracle MetaLink, Oracle Corporation
MND Service, Inc.
Mitchell Energy Company L.P.

About the Authors

Mary Platt is a DBA/Developer and Bobbie Lambert is a Developer/Analyst with MND Service, Inc., a division of Mitchell Energy Company L.P., based in The Woodlands and Ft. Worth, Texas respectively. Mary has been working with Oracle products since 1988 when Oracle Version 5 was installed.

Bobbie has developed with Oracle products since joining the company in 1993. Together Mary and Bobbie present the Mary Lou and Bobbie Sue's series of papers. Their papers have been presented at IOUG-A '99, SCOUG Training Days '99 and IOUG-A 2000. Neither MND Service, Inc. nor the authors guarantee this document to be error free. ▲

This article is excerpted and reprinted with permission from the IOUG-A Live! Conference, April 2001.

WHEN YOUR STORAGE
SYSTEM PERFORMS WELL,
SO DO YOU

FasFile RAID from Seek Systems accelerates the performance of database applications without modifications to your existing infrastructure.

Call us...

You can download our technical paper on how to improve your Oracle database application performance at:

<http://seeksystems.com/products/adaptive.html#downloads>

OR ... Call Seek Systems Region Manager, Chris



www.seeksystems.com
1-800-790-7335

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.

- Oracle
- Sybase
- MS SQL Server

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

Where the experts go
for performance.

LECCOTECH
www.leccotech.com



TECH TIPS

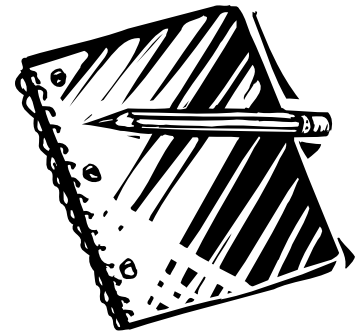
Tora- Freeware for Development

Database administrators and developers alike have come to rely on specialized tools to increase productivity when working with Oracle. One of these helpful tools is a GUI interface that allows for faster PL/SQL development. Many of you already know about and use TOAD from Quest Software. However, unfortunately, TOAD only runs in a Windows environment.

For those of you using Linux, we'd like to point to an alternative. TORA (Toolkit for Oracle) features tools including a SQL worksheet with PL/SQL block parsing, PL/SQL debugger and editor with syntax highlighting, and Schema browser, just to name a few.

You can find both the Linux and the Windows versions available for free download at:

<http://cnet.search.com/search?curl=1,0,0-0-0&timeout=3&tag=srch&qt=tora&cn=&ca=0> ▲



Mark Your Calendars!

The NoCOUG Spring Conference is scheduled to take place on Thursday, May 16 at Lockheed Martin in Sunnyvale. Check the NoCOUG website for updates. ▲



RESOURCE CORNER

Get in the Zone

We'd like to share with you an interesting site that we think you developer folks will like. The "database dev" zone by DevX puts many database developer resources at your fingertips. There is an archive of thousands of technical articles on software development, reviews of development products, discussion groups, and, of course, lots of helpful tips.

The DevX mission is to provide "the international IT community with the most comprehensive information available on software development for corporate applications." Check it out for yourself at <http://www.devx.com/dbzone>. To get Oracle-specific information, click on the tab halfway down the page in the navigation bar on the left side of the page.

A Good Way to Explore Oracle

Element K Journals is a provider of computer technology journals and information products. They've got a great magazine called Exploring Oracle which is a comprehensive publication for Oracle professionals. For those who are not regular subscribers, there are many free as well as pay-per-view white papers. Folks with real-life experience, just like you, write these white papers. Recent topics include:

- To 9i or not to 9i
- SQL*Load and Views for Quick File Review
- Oracle Dimensions and Hierarchies

Visit <http://www.elementkjournals.com/dbm/> to check it out for yourself.

Send your favorite resources to journal@nocoug.org for publication. ▲



Advanced SQL Tips for Developers and DBAs

The following are excerpts from a white paper by Tim Quinlan, TLQ Consulting Inc.

Temporary Tables

Query too complicated? Consider Temporary Tables.

You may run across situations where you find yourself scratching your head as you try to figure out a way to write a very complicated query. In some cases, resolving part of the query and writing data to a temporary table (or tables) and then joining that table to other tables will solve your problem.

When using this approach, keep the following points in mind:

- There is overhead involved with creating and storing temporary data.
- This is not a good approach for high transaction volume OLTP (online transaction processing) systems.
- This is useful for one-time SQL or for background tasks that are not frequently executed.
- When creating temporary tables use the “temporary” option to reduce the overhead incurred when creating normal (non-temporary) tables.
- This is a very simple solution that is perhaps not the most efficient, but can simplify logic for infrequently run queries.
- A useful approach to use when multiple, different

groupings of data need to be applied.

Oracle supports Global Temporary Tables. The table is temporary and its definition is visible to all oracle sessions. BUT, the data in the table is only visible to the session that inserts the data. The data in this table exists only for the session or the transaction. The “on commit” keyword states whether the data is transaction- or session-specific.

“On Commit Delete Rows” states that this is transaction-specific and that the table will be truncated at commit time. “On Commit Preserve Rows” is session-specific and the table is truncated when the session ends. An example from Oracle’s documentation is:

```
CREATE GLOBAL TEMPORARY TABLE work_area
  (startdate DATE,
  enddate DATE,
  class CHAR(20))
  ON COMMIT DELETE ROWS;
```

Another example of this is:

```
Create Global Temporary Table new_temp_table (
column1 char(10))
On Commit Preserve Rows;
```

Mixing DDL and DML in Transactions

In cases where you must mix Data Definition Language (DDL) statements such as “Create Table” with Data Manipulation Language (DML) statements such as Insert, Update, and Delete you must be careful where you place the DDL statements. You must always keep in mind that the DDL statement will perform an implicit commit and that this may occur in a place that you didn’t expect. Take a look at the following series of steps that are performed in order in a single transaction:

```
Begin Transaction
Insert into ...
Update ...
Create Table table_name ...
```

```
Update ...
Rollback Transaction
End Transaction
```

Given the above steps, it appears as though the “Rollback Transaction” will undo all work back to the “Begin Transaction” including the Insert and two Update statements. This, however, is not what will happen. The DDL statement “Create Table table_name ...” will perform an implicit commit which will commit the insert and update statement before it. The rollback statement will only undo the last update statement. Your best bet is to avoid mixing DDL and DML in a single transaction.

Quick Null Tips

Searching for and Comparing Null values

- 1) One Null value can never be considered to be equal to, not equal to, greater than or less than another. We cannot compare one indeterminate value to another. Therefore the following statement will never return a value - even if the customer_name is NULL:

```
Select *
From Customer
Where customer_name = '';
```

When comparing Null values use the 'IS NULL' value rather than '='. The following corrects this statement:

```
Select *
From Customer
Where customer_name IS NULL;
```

- 2) If you want to show all total_amounts < 100 including those that are null, code the query as follows:

```
Select *
From Sales
Where total_amount < 100
or total_amount IS NULL;
```

- 3) When are Null Values Equal?

Consider the following three statements:

- ▶ If either value in a comparison (=) is Null, the result is unknown.
- ▶ A null is never equal to another null value.
- ▶ A null is neither equal to nor is it not equal to another value.

The above statements are all true. There is, however, one exception to the above. That is in the case of duplicates.

When using the distinct function, two null values will be grouped as one. Also, a unique index on a null column will allow only one null value (if it is a single column index, or if other uniquely-indexed columns from this row match the values of the same columns on another row) and the second will be considered a duplicate. Keep this in mind when grouping data and creating unique indexes.

- 4) Null Indexes

Be careful of index entries that are completely null - an index entry may not be kept. So, queries using IS NULL logic will be forced to perform a table scan.

Using a Function in SQL

Using a function inside a SQL statement can be a handy feature in some cases where the function exists and where it is complicated to duplicate the logic in the SQL statement. An example of a SQL statement that uses a function or package is shown below:

```
SELECT *
FROM Customer
WHERE customer.state =
state_package.active_state;
```

In the above SQL statement, the state column on the customer table is compared to the result of the value of active_state in the state_package. In this example, the use of the function is not obvious. The expression "state_package.active_state" looks like a table and column name rather than a function call. In fact, this is a call to a package rather than a function, but the result is the same. There are function calls that will look more obvious such as the one shown below where the value in the program variable "active_state_variable" is passed to the state_function. The result of this call is then compared to the value in column customer.state.

```
SELECT *
FROM Customer
WHERE customer.state =
state_function(active_state_variable);
```

An alternate approach to this would be to move the state_package.active_state value to a variable and use that variable in your program.

One thing to remember when using this approach is that the function must be called for every row retrieved in the table. So, if this is a situation where the result of the function call will always be the same, move the result to a variable and compare the variable to the column in the SQL statement where clause.

NoCOUG Winter Conference Session Descriptions

(continued from page 27)

least understood by information technology practitioners. This is both a cause and a consequence of the failure by SQL and its commercial dialects to implement relational domains, which are nothing but data types of arbitrary complexity. Consequently, blame is being misplaced on the relational approach for the SQL products' lack of support for so-called "complex" types, which permits proponents of the object approach to claim with impunity that object DBMSs are superior in this respect to relational DBMSs.

This presentation will help define the data type concept as well as explain the distinctions between "simple" and "complex" types and "system-supplied" and "user-defined" types. We'll clarify what proper DBMS support of data types means, and we'll help you derive some practical conclusions and recommendations. *(Intermediate Level)*

Conference: **April 14-18, 2002**
Exhibits: **April 15-17, 2002**

San Diego Convention Center
San Diego, California

IOUG LIVE! 2002

THE ORACLE USERS' CHART FOR SUCCESS



Chart your course to San Diego for International Oracle Users Group (IOUG) Live! 2002, April 14-18, and steer your organization towards a more efficient and successful Oracle implementation.

Diving into the following seven focus areas, the Live! 2002 technical program is designed to help Oracle professionals like you chart your course to success:

- Database Administration (DBA)
- Architecture & Infrastructure
- Data Warehousing & Business Intelligence
- Application Server Technologies
- eBusiness & Application Development
- ERP, CRM, SFA and SCM
- Career Development & Project Management

You can't afford to miss IOUG Live! 2002 - THE Education Conference for Oracle Users by Oracle Users.

To receive important IOUG Live! 2002 updates email live@ioug.org!

REGISTER NOW! at
www.ioug.org/live2002 -

where you'll find session abstracts, registration details, and complete conference information — or call +1.312.245.1579 for more information.

LIVE!

The Education Conference for Oracle Users by Oracle Users

IOUG
INTERNATIONAL ORACLE USERS GROUP

NoCOUG Winter Conference

Thursday, February 21, 2002

Session Descriptions

KEYNOTE

9:45-10:45

TBA

Oracle Corporation

DBA TRACK

11:15-12:15

Interpreting Wait Events to Boost System Performance

Roger Schrag,
Senior Consultant,
Database Specialists, Inc.

At any given moment, every Oracle server process is either busy servicing a request or waiting for something specific to happen—a condition we call a “wait event.” The Oracle instance maintains detailed statistics of all wait events—how many times each session has waited for a specific event, total time waited, and exactly what events are being waited on right now. This information can be used to gain insight into the performance of both your application and database. In this presentation we will define what wait events are, detail how to extract wait event information from the instance, and learn how to interpret the data we collect. Many of us have heard of wait events and know how to query the v\$ views, but in this presentation we’ll go further than that. We’ll walk through real life examples of how to use wait event data to diagnose problems and boost system performance. (*Advanced Level*)

1:30-2:30

Top 7 Oracle Database Tuning Techniques and Case Studies

Lev Moltyaner
Managing Partner,
Procise Consulting, Inc.

Tuning experience is difficult to share because it is often application specific. In this session we will define seven practical tuning techniques, including: eliminating inefficiencies in process flow, utilizing intended index-

es, adopting best indexing options, decreasing PL/SQL context switches, reducing I/O contention, increasing memory utilization, and increasing CPU usage by multi-threading. We will present a case study of how each technique was applied and summarize its performance gains. We will also illustrate how these techniques represent a tuning methodology that can be applied to any database. We hope that each attendee will be able to identify immediate benefits for their databases. (*Intermediate Level*)

3:15-4:15

Oracle9i Real Application Clusters on 8 Node Intel-based Clusters – An OLTP Case Study

Kevin Closson, Sr.
Staff Software Engineer, Database
Engineering Group, PolyServe, Inc.

Clustered database technology is a reality with Oracle9i Real Application Clusters (RAC). Compelling hardware economies and the robust horizontal scaling characteristics of RAC are motivating datacenter managers to investigate deploying applications on clustered commodity-based hardware. This presentation is based upon a case study of an Oracle9i RAC OLTP application deployed on an eight-node Intel-based cluster running Linux. Configuration and tuning tips, along with lessons learned, will be covered. Special attention will be given to alerts that can be gleaned from several new internal (gv\$) performance views. Additionally, platform issues such as Clustered File Systems will be covered. (*Intermediate Level*)

DEVELOPER TRACK

11:15-12:15

Oracle9i JDeveloper Part I: Introduction and New Features

Peter Koletzke,
Technical Director, Quovera

JDeveloper is the tool of choice for Java development because of its rich

support for the Oracle database. In addition, it offers the major ease-of-use and code-generation features of a sophisticated Integrated Development Environment. The 9i release added a number of powerful features that you can use for assistance with Java application development. In addition to new features, some operations have changed from previous versions. This presentation discusses and demonstrates the basics of the tool as well as the major new features and operations. This will help both the developer who has not yet examined the tool as well as the developer who is familiar with previous releases. (*Beginner Level*)

1:30-2:30

Oracle9i JDeveloper Part II: Developing Java Client and JSP Applications

Peter Koletzke,
Technical Director, Quovera

JDeveloper’s interface allows you to quickly develop bug-free, default applications in many different styles. The development challenges lie in understanding these different application styles and in knowing where to go in the tool to get started. This presentation explains the characteristics of two application options that you can create in JDeveloper: Java Client or JClient (Java application and applet) and JavaServer Pages (JSPs). We’ll also explore the architectures of each of these options and compare their benefits and drawbacks as well as how to use JDeveloper to develop each alternative. This presentation demonstrates application development methods and explains how to go beyond the wizards to create a production-ready application. (*Beginner Level*)

This presentation is intended as a follow up to the “Introduction and New Features” presentation, but attendance at that session is not required.

3:15-4:15

Be Careful What You Wish For: Complex Data Types

Fabian Pascal
Independent Consultant

The data type concept is one of the
(continued on page 25)

NoCOUG Winter Conference

Thursday, February 21, 2002

Location: Oracle Conference Center, Redwood Shores (located at 350 Oracle Parkway).

Sponsored by Oracle Corporation

- 8:00-9:00 Registration and welcome. Refreshments served.
- 9:00-9:45 Opening Remarks and Announcements/Vendor Introduction
- 9:45-10:45 *Keynote:* Oracle Corporation—TBA
- 10:45-11:15 Morning Break
- 11:15-12:15 **Parallel Sessions**
DBA Track: Interpreting Wait Events to Boost System Performance
Developer Track: Oracle9i JDeveloper Part I: Introduction and New Features
Oracle by Oracle: 9iAS Overview
- 12:15-1:30 Lunch Break
- 1:30-2:30 **Parallel Sessions**
DBA Track: Top 7 Oracle Database Tuning Techniques and Case Studies
Developer Track: Oracle9i JDeveloper Part II: Developing Java Client and JSP Applications
Oracle by Oracle: Oracle9i Integrated Business Intelligence
- 2:30-3:15 Afternoon Break
- 3:15-4:15 **Parallel Sessions**
DBA Track: Oracle9i Real Application Clusters on 8 Node Intel-based Clusters – An OLTP Case Study
Developer Track: Be Careful What You Wish For: Complex Data Types
Oracle by Oracle: RAC Overview
- 4:30-?? Networking and Happy Hour: *location TBA*
- Cost:** \$40 admission fee for non-members. Members free. Includes lunch voucher.



For session descriptions, check out page 27.

Thank you to Oracle Corporation, our meeting sponsor.

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