Vlado Barun & Graham Wood
A Guide To Improving Performance

Peter Carter
Understanding Threat Analysis

Jonathan Gennick
Lessons From The Bicycle

Darl Kuhn
Explaining Temporary Tables

Glenn Schwartzberg
Using SQL With Essbase

APEX 2015 Educator of the Year - Denise Gardner
Make an unexpected impact by delivering 30x faster performance and more storage with a Meta7 Exadata refresh.

Database Training with Dan Hotka

www.informit.com/hotka

Video Courses 50% OFF
with code HOTKAVISD

- Oracle SQL Performance Tuning for Developers
- Oracle PL/SQL Programming (3 options)
  1. Fundamentals
  2. Advanced
  3. Complete Course
- Oracle SQL
- Toad for Oracle
- OBIEE 11g Reports and Dashboards

Book and eBook 37% OFF
with code HOTKABOOK

- Toad for Oracle
  by Bert Scala and Dan Hotka
  ISBN 9780134131856

RMOUG Congratulates Denise Gardner

Recipient of the Colorado Technology Association 2015 APEX Educator of the Year Award

See her interview with Tim Gorman on Page 16
Featured Articles

6 Explaining Temporary Tables
by Darl Kuhn
The Generation of Redo and Undo

10 Lessons From The Bicycle
by Jonathan Gennick
Everything Means Something

14 A Guide To Improving Performance
by Vlado Barun & Graham Wood
Without Upgrading Your Hardware

18 Using SQL With Essbase
by Glenn Schwartzberg
Introduction & Loading Data

22 Understanding Threat Analysis
by Peter Carter
Who's At Risk?

On the Cover:

Cover photo by Sal Jr. Sal was born and raised on Long Island in New York. He moved to Bailey, Colorado on 30 November 2012 and works as a computer programmer for a world wide freight forwarding company. He has also been a stock broker and owner of a bowling center. Sal has four children, 3 girls and 1 boy. His hobbies are chess, bowling, card playing, and taking lots of pictures.

This photo was taken on a bus ride on State Route 550 from Durango to Silverton.
For the 27th year in a row, it's time for our Training Days conference, coming up in February. For those who don't know, your little old local Rocky Mountain Oracle Users Group annually produces the fourth largest Oracle event in North America...

1. Oracle Open World 2015 attracted about 60,000 attendees to San Francisco last October
2. Collaborate 2016 will probably attract about 5,000 attendees to Las Vegas next April
3. KScope16 will probably attract about 2,000 attendees to Chicago next June
4. RMOUG Training Days 2016 will probably attract about 700 attendees to Denver next February - hopefully you're planning to come!

There is a possibility that RMOUG Training Days is as big as, or bigger, than any events in Latin America too, so that would give us the fourth largest Oracle event in the Western Hemisphere. If you keep expanding the territory, we certainly have one of the top ten Oracle events events in the known universe. Not bad for this little old cow town perched between the Rocky Mountains and the prairie.

This year, the conference will once again be a two-and-a-half day event, with great presentations and labs packed into every minute...

- Tuesday 09-February 2016 (afternoon only)
  - twelve 2-hour “deep dive” sessions
  - twelve 30-min “tips” sessions

- Wednesday 10-February 2016 (all day)
  - fifty 1-hour sessions across 10 tracks
  - keynote by Carlos Sierra and Mauro Pagano
  - welcome reception featuring “Geeks Who Drink”
  - trivia contests awarding “Beer Bucks” good at any time at dozens of breweries on the Front Range

- Thursday 11-February 2016 (all day)
  - fifty 1-hour sessions across 10 tracks

This year’s conference has 88 speakers from Colorado, all around the USA, Mexico, Canada, Europe, and Australia

- 45 speakers are members of the Oracle ACE program
- 18 speakers are product managers or senior technical staff from Oracle Corporation
- 5 speakers are members of the Oak Table network

As you can see, we’ve also managed to avoid side-swiping Valentine’s Day this year, so we won’t even get you in trouble at home.

Besides the annual Training Days conference, watch for other RMOUG events happening quarterly or monthly, such as the monthly Big Data meetups (more info at “http://www.meetup.com/RMOUG-Big-Data-SIG-Meetup/”), the Raspberry Pi and STEAM meetups (more info at “http://www.meetup.com/RaspberryPiSig/”), and the Regis DBLab meetups (more info at “http://www.meetup.com/RMOUGLabs/”). Meetups are informal events, usually after work, usually with a dozen people or less.

Also, starting in January are our social “RMOUG@Breakfast” and “RMOUG@Lunch” events. These have no agenda, just an opportunity to chat and share a meal with your friends and colleagues in the RMOUG community. We’re going to try to organize these light events all over the Front Range, from Colorado Springs all the way up to Fort Collins, so if you’re interested, please contact “president@rmoug.org” if you’d like to organize a breakfast or lunch, or watch the RMOUG website Calendar page at “http://www.rmoug.org/calendar” if you’d like to join us!

And of course, don’t miss our quarterly educational workshops (QEWs), with the next one focusing on APEX and scheduled for the first week of May 2016. Please watch the RMOUG website Calendar page at “http://www.rmoug.org/calendar”, as well as the front page of the RMOUG website, as we get closer to that time. These events are usually for half a day, presenting 3-4 speakers.

If you have the desire and opportunity to learn something new and keep up with our fast-changing industry without a big commitment in time or money, then please check out the RMOUG website Calendar page at “http://www.rmoug.org/calendar” has accumulated one of the most complete listings of free educational Oracle and technical webinars available anywhere. All of the webinars posted there are free, produced either by Oracle Corporation, by other Oracle users groups such as IOUG or ODTUG, by other user groups such as TDWI or Tableau, or by companies like SolarWinds, AuraPlayer, and Delphix. Bookmark this page and remind yourself to check back from time to time, as new free events are posted every few days.

Also, in case you didn’t know, RMOUG has been awarding scholarships to high-school and undergraduate students for the past 10 years under our own Stan Yellott Scholarship Foundation (SYSF). There’s information on how the scholarship
The Scholarship Fund started in 2001 to encourage future IT professionals in their efforts to broaden their knowledge. In 2007, RMOUG voted to rename the scholarship fund to honor the memory of Stan Yellott. Stan was a long time member of RMOUG where he supported the user community by serving on the RMOUG board. Stan focused on expanding Oracle educational opportunities. Stan’s vision was to include high school and college students as the next generation of IT professionals.

Application Deadlines
- March 15th for an April scholarship distribution
- June 15th for a July scholarship distribution
- September 15th for an October scholarship distribution
- December 15th for a January scholarship distribution for the following year in January

Scholarship Recipients receive a complimentary pass to RMOUG’s Annual Training Days.

Please see a detailed outline of RMOUG’s scholarship policies on page 13 or visit www.rmoug.org.

The Scholarship Fund started in 2001 to encourage future IT professionals in their efforts to broaden their knowledge. In 2007, RMOUG voted to rename the scholarship fund to honor the memory of Stan Yellott. Stan was a long time member of RMOUG where he supported the user community by serving on the RMOUG board. Stan focused on expanding Oracle educational opportunities. Stan’s vision was to include high school and college students as the next generation of IT professionals.

Application Deadlines
- March 15th for an April scholarship distribution
- June 15th for a July scholarship distribution
- September 15th for an October scholarship distribution
- December 15th for a January scholarship distribution for the following year in January

Scholarship Recipients receive a complimentary pass to RMOUG’s Annual Training Days.

Please see a detailed outline of RMOUG’s scholarship policies on page 13 or visit www.rmoug.org.
Explaining Temporary Tables and
The Generation of Redo and Undo

by Darl Kuhn

Temporary tables are used to hold intermediate resultsets for the duration of either a transaction or a session. This feature has been within Oracle for many releases now (introduced in Oracle8i version 8.1.5). Even though temporary tables have been around for a while, there is still some confusion surrounding them, in particular in the area of logging. In this article, we'll explore only the question “How do temporary tables work with respect to logging of changes?” In Oracle Database 12c, the processing of undo for temporary tables is significantly enhanced. Therefore we'll break this topic into two sections: Prior to Oracle Database 12c and Starting with Oracle Database 12c.

Prior to Oracle Database 12c
Temporary tables generate no redo for their blocks. Therefore, an operation on a temporary table is not recoverable. When you modify a block in a temporary table, no record of this change will be made in the redo log files. However, temporary tables do generate undo, and the undo is logged. Hence, temporary tables will generate some redo. At first glance, this doesn't seem to make total sense: Why would they need to generate undo? This is because you can roll back to a SAVEPOINT within a transaction. You might erase the last 50 INSERTs into a temporary table, leaving the first 50. Temporary tables can have constraints and everything else a normal table can have. They might fail a statement on the five-hundredth row of a 500-row INSERT, necessitating a rollback of that statement. Since temporary tables behave in general just like normal tables, temporary tables must generate undo. Since undo data must be logged, temporary tables will generate some redo log for the undo they generate.

This is not nearly as ominous as it seems. The primary SQL statements used against temporary tables are INSERTs and SELECTs. Fortunately, INSERTs generate very little undo (you need to restore the block to “nothing,” and it doesn't take very much room to store “nothing”), and SELECTs generate no undo. Hence, if you use temporary tables for INSERTs and SELECTs exclusively, this section means nothing to you. It is only if you UPDATE or DELETE that you might be concerned about this.

We'll set up a small test to demonstrate the amount of redo generated while working with temporary tables, an indication therefore of the amount of undo generated for temporary tables, since only the undo is logged for them. To demonstrate, we'll take identically configured permanent and temporary tables, and then perform the same operations on each, measuring the amount of redo generated each time. The tables we'll use are as follows:

SQL> create table perm
2  ( x char(2000) ,
3    y char(2000) ,
4    z char(2000) )
5  /
Table created.

SQL> create global temporary table temp
2  ( x char(2000) ,
3    y char(2000) ,
4    z char(2000) )
5  on commit preserve rows
6  /
Table created.

As we do these INSERTs, UPDATEs, and DELETEs, we'll measure the amount of redo our session generates using this small utility function:

SQL> create or replace function get_stat_val( p_name in varchar2 ) return number
2  as
3    l_val number;
4  begin
5    select b.value
6    into l_val
7    from v$statname a, v$mystat b
8    where a.statistic# = b.statistic#
9    and a.name = p_name;
10  return l_val;
11  end;
12   /
Function created.

Note The owner of the previous function will need to have been directly granted the SELECT privilege on
the V$ views V_$STATNAME and V_$MYSTAT. If the direct grants do not exist you'll receive a "PL/SQL: ORA-00942: table or view does not exist" error when creating the function.

We'll also set up a small stored procedure to allow me to perform arbitrary SQL and report the amount of redo generated by that SQL. We'll use this routine to perform INSERTs, UPDATEs, and DELETEs against both the temporary and permanent tables:

```sql
SQL> create or replace procedure do_sql( p_sql in varchar2 )
2  as
3      l_start_redo number;
4      l_redo number;
5  begin
6          l_start_redo := get_stat_val( 'redo size' );
7          execute immediate p_sql;
8          commit;
9          l_redo := get_stat_val( 'redo size' ) - l_start_redo;
10         dbms_output.put_line
11          ( to_char(l_redo,'99,999,999') ||' bytes of redo generated for "'||
12            substr( replace( p_sql, chr(10), ' '), 1, 25 ) ||""
13            ||'"..."');
14  end;
15  /
```

Procedure created.

Then we run equivalent INSERTs, UPDATEs, and DELETEs against both the PERM and TEMP tables:

```sql
SQL> set serveroutput on format wrapped
SQL> begin
2      do_sql( 'insert into perm
3               select 1,1,1
4                 from all_objects
5                where rownum <= 500' );
6      do_sql( 'insert into temp
7               select 1,1,1
8                 from all_objects
9                where rownum <= 500' );
10     dbms_output.new_line;
11     do_sql( 'update perm set x = 2' );
12     do_sql( 'update temp set x = 2' );
13     dbms_output.new_line;
14     do_sql( 'delete from perm' );
15     do_sql( 'delete from temp' );
16  end;
17  /
```

3,268,384 bytes of redo generated for "update perm set x = 2"
1,946,432 bytes of redo generated for "update temp set x = 2"
3,245,112 bytes of redo generated for "delete from perm"
3,224,460 bytes of redo generated for "delete from temp"

PL/SQL procedure successfully completed.

As you can see,

- The INSERT into the “real” table generated a lot of redo, while almost no redo was generated for the temporary table. This makes sense—there is very little undo data generated for INSERTs and only undo data is logged for temporary tables.
- The UPDATE of the real table generated about twice the amount of redo as the temporary table. Again, this makes sense. About half of that UPDATE, the “before image,” had to be saved. The “after image” (redo) for the temporary table did not have to be saved.
- The DELETEs each took about the same amount of redo space. This makes sense, because the undo for a DELETE is big, but the redo for the modified blocks is very small. Hence, a DELETE against a temporary table takes place very much in the same fashion as a DELETE against a permanent table.

Therefore, the following generalizations can be made regarding DML activity on temporary tables:

- An INSERT will generate little to no undo/redo activity.
- An UPDATE will generate about half the redo as with a permanent table.
- A DELETE will generate the same amount of redo as with a permanent table.

There are notable exceptions to the next to last statement. For example, if we UPDATE a column that is entirely NULL with 2,000 bytes of data, there will be very little undo data generated. This UPDATE will behave like the INSERT. On the other hand, if we UPDATE a column with 2,000 bytes of data to be NULL, it will behave like the DELETE as far as redo generation is concerned. On average, you can expect an UPDATE against a temporary table to produce about 50 percent of the undo/redo you’d experience with a permanent table.

In addition, you must consider any indexes in place on your temporary tables. Index modifications will also generate undo—which in turn generates redo. If you rerun the above example with these two indexes in place:

```sql
SQL> create index perm_idx on perm(x);
Index created.
SQL> create index temp_idx on temp(x);
Index created.
```

With the indexes in place, we run the code again to observe the redo generated:
set serveroutput on format wrapped
SQL> begin
  2    do_sql( 'insert into perm
  3       select 1,1,1
  4          from all_objects
  5          where rownum <= 500' );
  6
  7    do_sql( 'insert into temp
  8       select 1,1,1
  9          from all_objects
 10          where rownum <= 500' );
 11    dbms_output.new_line;
 12
 13    do_sql( 'update perm set x = 2' );
 14    do_sql( 'update temp set x = 2' );
 15    dbms_output.new_line;
 16
 17    do_sql( 'delete from perm' );
 18    do_sql( 'delete from temp' );
 19  end;
 20 /

11,735,576 bytes of redo generated for "insert into perm"
  3,351,864 bytes of redo generated for "insert into temp"

  9,257,748 bytes of redo generated for "update perm set x = 2"
  5,465,868 bytes of redo generated for "update temp set x = 2"

  4,434,992 bytes of redo generated for "delete from perm"
  4,371,620 bytes of redo generated for "delete from temp"

PL/SQL procedure successfully completed.

The numbers hold true from what we saw before—but you can see that the index definitely added to the redo generated. The INSERT into the global temporary table went from generating almost no redo to generating 3.3MB of redo. All of this additional redo was related to the undo produced for the index maintenance.

**Note** This is an exaggerated example. The index in question was on a CHAR(2000) column; the index key is much larger than you'll normally see in real life. Don't expect this much additional redo typically.

In general, common sense prevails in estimating the amount of redo created. If the operation you perform causes undo data to be created, then determine how easy or hard it will be to reverse (undo) the effect of your operation. If you INSERT 2,000 bytes, the reverse of this is easy. You simply go back to no bytes. If you DELETE 2,000 bytes, the reverse is INSERTing 2,000 bytes. In this case, the redo is substantial.

Armed with this knowledge, you will avoid deleting from temporary tables. You can use TRUNCATE, bearing in mind, of course, that TRUNCATE is DDL that will commit your transaction, and in Oracle9i and before invalidate your cursors. Or just let the temporary tables empty themselves automatically after a COMMIT or when your session terminates. All of these methods generate no undo and, therefore, no redo. You should try to avoid updating a temporary table unless you really have to for some reason. You should use temporary tables mostly as something to be INSERTed into and SELECTed from. In this fashion, you'll make optimum use of their unique ability to not generate redo.

**Starting with Oracle Database 12c**

As you saw in the previous section, when issuing INSERT, UPDATE, and DELETE statements in a temporary table, the undo for those changes is recorded in the undo tablespace, which in turn will generate redo. With the advent of Oracle Database 12c, you can instruct Oracle to store the undo for a temporary table in a temporary tablespace via the TEMP_UNDO_ENABLED parameter. When blocks are modified in a temporary tablespace, no redo is generated. Therefore, when TEMP_UNDO_ENABLED is set to TRUE, any DML issued against a temporary table will generate little or no redo.

**Note** By default, TEMP_UNDO_ENABLED is set to FALSE. So unless otherwise configured, temporary tables will generate the same amount of redo in Oracle Database 12c as in prior releases.

The TEMP_UNDO_ENABLED parameter can be set at the session or system level. Here's an example of setting it to TRUE at the session level:

SQL> alter session set temp_undo_enabled=true;

Once enabled for a session, any modifications to data in a temporary table in that session will have a subsequent undo logged to the temporary tablespace. Any modifications to permanent tables will still have undo logged to the undo tablespace. To see the impact of this, we'll rerun the exact same code (from the “Prior to 12c” section) that displays the amount of redo generated when issuing transactions against a permanent table and a temporary table—with the only addition being that TEMP_UNDO_ENABLED is set to TRUE.

set serveroutput on format wrapped
SQL> begin
  2    do_sql( 'insert into perm
  3       select 1,1,1
  4          from all_objects
  5          where rownum <= 500' );
  6
  7    do_sql( 'insert into temp
  8       select 1,1,1
  9          from all_objects
 10          where rownum <= 500' );
 11    dbms_output.new_line;
 12
 13    do_sql( 'update perm set x = 2' );
 14    do_sql( 'update temp set x = 2' );
 15    dbms_output.new_line;
 16
 17    do_sql( 'delete from perm' );
 18    do_sql( 'delete from temp' );
 19  end;
 20 /

8 SQL> UPDATE • Winter 2015
no way to disable that. The redo is generated for the rollback data, and in most typical uses it will be negligible. If you only *INSERT* and *SELECT* from temporary tables, the amount of redo generated will not be noticeable. Only if you *DELETE* or *UPDATE* a temporary table heavily will you see large amounts of redo generated.

Starting with Oracle Database 12c, you can instruct Oracle to write undo to the temporary tablespace and thereby eliminate almost all of the redo generation. This is done by setting the `TEMP_UNDO_ENABLED` parameter to `TRUE`. Therefore if you need to temporarily persist and process a set of rows, using a temporary table can have significant performance benefits over using a permanent table.

**Acknowledgements**

This article was taken from material contained in *Expert Oracle Database Architecture*, 3rd edition, Apress, authored by Tom Kyte and Darl Kuhn.

**About the Author**

Darl is currently a DBA working for Oracle. He has written books on a variety of IT topics including SQL, Performance Tuning, Oracle Internals, Linux, Backup and Recovery, RMAN, and Database Administration. Darl also teaches Oracle classes for Regis University.
Lessons From The Bicycle

by Jonathan Gennick

I’m obsessed with bicycles. At various times in my life I’ve also been obsessed with books, photography, caving, skiing, kayaking, but bicycles -- riding and repairing them -- are my current and lasting obsession. Bicycles bring health and joy, and some fun object-lessons.

Pulling Together is Power

I’m heavier than I’d like and some bicycle wheels seem awfully lightweight and fragile. The more so when I’m landing a jump at the local skatepark or blasting down some rocky trail in Marquette. What keeps my wheels from flying apart? Individual spokes are weak, but in pulling together they support my weight. Likewise with teams at work. There’s power in a group working toward an agreed-upon goal.

Small Gains Add Up

It’s easy to look at a big task or a change we want to make and become intimidated at the wide chasm between where we are and where we want to be. In summer of 2006 I could not bike up the hills surrounding Munising without pausing to gasp for air every hundred yards, and sometimes every hundred feet. Then I bought studded tires and rode the bike on errands around town all winter. Unbeknownst to me, all those short trips around town through snowy streets and slush were slowly building up my endurance. I was stunned in spring of 2007 to find that I could ascend any hill around town in one go.

I feel the same intimidation when faced with learning new technology, whether it’s a website builder like Squarespace or a language like CSS or SQL. I used to be able to do magic in COBOL, but can’t code my way out of a paper bag in Java. Frustrating! But I need to remember the lesson from that winter long ago: Don’t try to leap the chasm one big jump. Take small steps. Be persistent! Be consistent. Let time be your friend. Focus on small wins, and suddenly one day you’ll find yourself on the other side looking back.

Invest in Yourself

Cycling became tremendously more fun after I gave myself permission to buy tools and try new parts just because I wanted to. I was cheap at first, refusing to spend and thereby missing opportunities to learn and grow. Looking back on my career, this exact issue is one on which I wish I could go back in time and tell my younger self to do better on. Spend a little money on that conference. Take some vacation time for it. Pay a little to join some user groups. Spend some money on hardware and software and training in order to grow your skillset and broaden your grasp of the field. These are all things I would tell my younger self.
Look Outside your Discipline

I was a mountain-biker and mainly still am, but one day I began to look closely at the BMXers riding the skate park down the street from my home. They did not sit on their seats. They floated their bodies over their bikes in what I later learned is termed the “attack position”. From these young riders I learned better positioning on the bike, how to let the bike move around under you, got comfortable at getting air, and got the hang of “the pump”. Looking outside my discipline made me a better rider, and I have more fun on the trails. (I have more friends too!)

Database administrators across brands like SQL Server and Oracle have much to learn from each other. Developers in PL/SQL can learn from developers in other languages like Java, and vice-versa. Look across disciplines and see what issues are being confronted and whether they apply to you. You’ll be amazed at the people you’ll meet and what you can learn from them and take back to your own work.

Be generous. Be inclusive. Share knowledge

I’ve gained friends and riding partners and a great deal of joy from helping others repair their bikes, and even more so from teaching others so they are able to do their own work. There’s a ripple effect from teaching others that’s fun to see, and that’s just as true in technology as it is in bicycles. You can’t go wrong in helping others grow their skills.

More than just the ripple effect there is the benefit to yourself and your career. You can’t help but learn new skills when sharing the ones you have. You’ll gain mentoring and presentation skills, and other soft skills that will surprise you. Plus you’ll get whatever skills the other party shares back at you. Consider that mentoring someone to take over your role is a necessary prerequisite of moving up the career ladder. Don’t want to be sat at the same desk for a decade? Step up and help others grow and advance, and you will too.

Challenge Yourself

Sometimes working a tough problem alone can be a good thing. I stubbornly tackled a compression damper problem this summer on one of my suspension forks. Such a simple device, and I couldn’t get it to work. The fork would not lock out. The damper settings didn’t seem to matter. I read the manual, spent several evenings repeatedly taking the damper assembly apart, wasted lots of suspension oil, made a big oily mess on my office floor, and finally, finally I groked what was going on and came away with valuable and hard-won knowledge that will continue to pay off in the future.

Likewise in technology. Every time I’ve stubbornly pursued a problem past the point of feeling incompetent and through to finding a solution has been a moment of growth that I look back on fondly as taking me to new levels of understanding.

Invest in Tools

Bicycles are simple objects and one can sometimes work around not having the right tools. But the work goes faster and is more fun when one has the right tools for the job. Back in the day we would manage Oracle Database with nothing but Server Manager and SQL*Plus. Today I’d be looking for Enterprise Manager and SQL Developer. I do enough work on bikes that I hardly think twice at ordering a new tool when it’s needed. Whatever your job is, find the right tools and learn to use them well.

Bling! Everyone Needs Some.

One of my favorite bikes is a blue frame that came with a black seatpost clamp. I replaced that black clamp with one that’s anodized red. The red really pops against the blue frame. It’s a small thing, but that bit-o-the-bling adds a lot to my enjoyment of the bike. Managers! (This lesson is for you). Your people need a bit of bling too. Fight for some nicer monitors, upgraded laptops, or extra nice chairs. Find ways to add some bling to the dreary cubicle life.

A Balanced Build is a Good Build

Database systems perform well when there is balance between available CPU cycles and input/output rates, and with available memory for caching and other uses. It’s the same with bicycles. Bolting a high-end derailleur onto a bike having cheap shifters and cables won’t give much of a gain. Better to buy a more modest derailleur and allocate the leftover money to upgraded shifters and cables and housing so that the entire system is in balance. Similarly in databases it’s best to consider the big picture and keep your systems in balance.

Experiments Are Ok

Will my Avid BB7 disk brake calipers for road bikes play well with mountain-bike levers? Sometimes one just needs to bolt together whatever parts one has and see what happens. Theorizing about what might happen is useful to a point, but the better answer
is always going to be the one from experience. Set up a test case when confronted a database question. See what really happens. Because experiential knowledge is powerful.

Failures Are Ok Too
If they are the right sort of failure. On my office wall is a cracked and ovalized chainring that I keep as a trophy of a glorious failure. Failing to stay tight in a turn one day, I swung a bit wide and crashed the crankset into a tree’s root crown. The bashguard shattered, I lost some chaining bolts, and ovalized my ring. It was a glorious failure and a point of pride because the “failure” was due to pushing myself to ride harder and faster. I wrecked the bike but stayed on and in control and came away undamaged. The tree is fine too, and is still there and growing.

Respect Client Priorities
Sometimes clients have priorities that are baffling to a DBA or a developer. My across-the-street neighbor this summer insisted upon going to a single-chainring setup with no bash guard and no retention device. That’s because he wanted one of my nice, anodized blue, wide/narrow rings. Sometimes his chain does drop and I get annoyed, but I grit my teeth and have learned to respect his choice to prioritize color over retention.

Get Away From the Desk
We all have desk jobs. I spend long days motionless in a chair and staring at text on a monitor. Then I spend evenings on my blog, researching and writing books, helping friends with their websites, and sometimes preparing sermons to deliver at my local church. All that sitting is not conducive to health. Bicycles remind me that sometimes what’s needed is to get away from the desk and engage in some physical activity. Give the brain a rest and the body a workout.

Some of my best ideas hit me during walks and rides. We all need to take care of ourselves, despite that our career choices tend to motivate too much time staring at monitors. Of all the lessons from the bicycle, this one is perhaps the one that matters most.
Scholarships will be awarded on a quarterly basis for amounts up to at least $1000 and with amounts not to exceed $4000 annually.

- Ideally, two $500 scholarships will be awarded quarterly

  - This is based on all applicants having submitted their paperwork by the quarterly due date and are approved by the scholarship committee and the RMOUG board of directors.

  ◊ If only one applicant submits all of the required paperwork by the quarterly due date, then he/she is eligible for both $500 scholarships for a total of $1000 for that quarter.

  ◊ Applicant(s) may also be awarded additional scholarship funds if there were scholarship roll-overs from other quarters where funds may have not been disbursed and upon discussion and approval of the RMOUG Board of Directors.

If no completed paperwork is submitted by the due date for a particular quarter, then the scholarship funds will be available for distribution for the next quarter or at a later time in the fiscal year.

All scholarship recipients will also receive one complementary RMOUG Training Days Pass for the next regularly scheduled RMOUG Training Days Event.

RMOUG Scholarship Recipients are not excluded from applying for future scholarship awards, but will be required to submit updated applications as defined in this document.

Awarding an additional scholarship award to a previous scholarship recipient will be at the discretion of the RMOUG BOD after appropriate discussions and deliberations on the merits.

Complete applications are defined as follows:

- A copy of official transcripts that must be received by RMOUG by the quarterly due date.
- A letter of reference from an IT instructor that can be emailed or mailed to RMOUG and must be received by the quarterly due date.
- An email or letter from the applicant that must be received by the quarterly due date and the letter must address the following topics:
  - Applicant’s intent to pursue a career in the Information Technology field
  - Academic achievements within the Computer Science/Information Technologies curriculum
  - Personal experiences as it relates to Information Technologies

Complete applications with all of the required paperwork must be received by:

- March 15th for an April scholarship distribution,
- June 15th for a July scholarship distribution,
- Sept 15th for an August scholarship Distribution,
- Dec 15th for a January scholarship distribution for the following year in January.

Each member of the committee will score the qualified applicants based on the following point system:

- The base score is equal to the GPA score on the current official transcript.
- The Applicant’s letter can score one point for adequately addressing each of the three topics for a maximum of three points:
  - Applicant’s intent to pursue a career in the Information Technology field
  - Academic achievements within the Computer Science/Information Technologies curriculum
  - Personal experiences as it relates to Information Technologies

The qualified applicant’s application, transcript, Letter of Recommendation, and score along with a written recommendation from the scholarship committee, are presented at the Board Meeting for discussion and final selection.

Once final selection is agreed upon, the Vice President will make a motion for Board approval.

The scholarship check that is given to the recipient will be addressed (written) to a college or university and then the check will be mailed to the scholarship recipient.

---

**Be Sure To Attend The**

**Spring Quarterly Education Workshop**

**May 6, 2016**

Building an application using APEX with David Peake, Oracle APEX Product Manager

*Announcements Coming Soon*

Visit [www.rmooug.org](http://www.rmooug.org) For Further Details
Performance is among the top topics of interest for users of the Oracle RDBMS. For decades, the Real-World Performance Team (RWP) at Oracle Corporation has worked with users to improve performance of their database applications by orders of magnitude (i.e. 10x, 100x, 1000x) without upgrading hardware. This article guides readers on their quest to getting the most performance out of the available hardware.

Identifying the root cause

To maximize the performance of a database application is it crucial to understand the reason for suboptimal performance. The following is one way of categorizing these root causes:

1. The database is not being used as it was designed to be used
2. The application architecture/code design is suboptimal
3. There is a suboptimal algorithm in the database (i.e. a database bug)

In RWPs experience, a major reason why organizations continue to suffer from suboptimal performance is because they are chasing symptoms rather than identifying the true root cause (see Figure 1).

Most organizations debug issues from the bottom up. The database is well instrumented to show performance issues, and most applications are not; this is frequently the reason that they assume the database is the issue.

However, to identify the true root cause of a performance issue, a top-down approach should be taken to get a holistic view of the system. As part of this top-down approach, we perform these steps:

- Verify that the application code is using appropriate algorithms and code
- Ensure application/ETL servers have a sound connection policy
- Investigate the database in conjunction with the full system infrastructure

Reading this you might think this is all common sense, but in the real world you probably don’t have access to the up-to-date application code/documentation nor is the application sufficiently instrumented. Fortunately, the Oracle database exposes a wide range of metrics that can be utilized to start learning about an application. For an example on how to do this, watch video #19: Architecture with an AWR Report in the RWP Learning Library, mentioned later in this article.

Top OLTP issues

Throughout the years, we have noticed that organizations have the most trouble identifying the root cause of OLTP suboptimal performance when presented with the following symptoms:

- Periods of acceptable performance and then unexplainable spikes of poor performance/availability
- The inability to determine what is happening in real-time because the system appears to hang

When following the top-down approach, the vast majority of these cases are due to:

- Inappropriate connection pool strategies causing, for example, connection storms and CPU oversubscription race conditions
- Undetected programmer error causing resource leakages, for example, open cursors exceeded, TX-row lock contention, etc.

To find out whether your system is susceptible to these symptoms, check the number of connections to the database (usually in the 1,000s) and the existences of a dynamic connection pool with a large number of logon/off (> 1/Sec ). For more details see videos #6 and #12-14.

Top DW issues

For Data Warehouse type workloads, the most common issues we observed are due to:
Sub optimal execution plans. These are usually caused by incorrect or outdated statistics gathering techniques, excessive or inappropriate hinting and inappropriate physical design (i.e. partitioning, indexing, compression, etc.)

Applications designed for small datasets working on large datasets. This mismatch is commonly due to the prevalence of put/get and row/object type processing in today’s applications and the lack of appreciation for SQL and database performance capabilities

Inappropriate (or non-existent) parallel execution strategy

Examples include application level parallelism causing inter-thread contention and inefficient degree of parallelism (DOP) causing resource under/over-utilization

How do we fix it?

Of course, the above discussion of the symptoms and root causes for suboptimal performance is very brief and lacks the needed details to determine the best course of action.

To help organizations address these challenges, RWP has created a set of free resources:
1. The RWP Learning Library at www.oracle.com/goto/oll/rwp
2. An intense hands-on 4 day training course, focused exclusively on how to achieve maximum performance and scalability for your database applications

RWP Learning Library

The Learning Library consists of 22 short videos (more are being added) that address specific aspects of performance (i.e. Index Contention, Log Writer, Bind Variables, etc.).

What makes these videos unique is that they

- contain live demonstrations of common mistakes
- explain the root cause of the performance bottleneck
- demonstrate a solution that achieves orders of magnitude performance improvements

<table>
<thead>
<tr>
<th>Topics/Symptom(s) covered</th>
<th>Video #</th>
</tr>
</thead>
<tbody>
<tr>
<td>Don't know where to start?</td>
<td>RWP Introduction Videos:</td>
</tr>
<tr>
<td>How to think about Performance</td>
<td>• What does Real-World Performance Mean?</td>
</tr>
<tr>
<td>Information about the RWP Training course</td>
<td>• Real-World Performance Education</td>
</tr>
<tr>
<td>• Introduction to Real-World Performance</td>
<td></td>
</tr>
<tr>
<td>How to efficiently access the Database</td>
<td>RWP #1: Cursors and Connections</td>
</tr>
<tr>
<td>RWP #2: Bad Performance with Frequent Logons</td>
<td></td>
</tr>
<tr>
<td>How to efficiently submit SQL to the Database</td>
<td>RWP #3: Connection Pools and Hard Parsing</td>
</tr>
<tr>
<td>How to improve security by avoiding SQL Injection</td>
<td>RWP #4: Bind Variables and Soft Parsing</td>
</tr>
<tr>
<td>What is the performance and scalability impact of using bind variables</td>
<td>RWP #5: Shared Cursors and One Parse</td>
</tr>
<tr>
<td>Encountering “ORA-01000: maximum open cursor exceeded”?</td>
<td>RWP #6: Leaking Cursors</td>
</tr>
<tr>
<td>Do you have large number of cursors/SQL?</td>
<td>RWP #7: Set Based Processing</td>
</tr>
<tr>
<td>Do you have sessions waiting for “SQL*Net break/reset to client”?</td>
<td>RWP #8: Set Based and Parallel Processing</td>
</tr>
<tr>
<td>Do you have scripts killing idle/blocking sessions?</td>
<td>RWP #9: Set Based Processing—Data Deduplication</td>
</tr>
<tr>
<td>Do you lose transactions and data?</td>
<td>RWP #10: Set Based Processing—Data Transformations</td>
</tr>
<tr>
<td>Serial set-based processing vs multi-threaded parallel row-by-row processing</td>
<td>RWP #11: Set Based Processing—Data Aggregation</td>
</tr>
<tr>
<td>Techniques for efficiently processing large data sets</td>
<td></td>
</tr>
<tr>
<td>Got an Unstable Database?</td>
<td>RWP #12: Resource Management</td>
</tr>
<tr>
<td>How to keep control of your Database</td>
<td></td>
</tr>
<tr>
<td>The dangers of large, dynamic connection pools</td>
<td>RWP #13: Large Dynamic Connection Pools - Part 1</td>
</tr>
<tr>
<td>Do you have sessions waiting for a combination of strange wait events, i.e. “latch: enqueue hash chains”, “latch: row cache objects”, “buffer busy waits”, etc.?</td>
<td>RWP #14: Large Dynamic Connection Pools - Part 2</td>
</tr>
<tr>
<td>Does your system suddenly hang?</td>
<td></td>
</tr>
<tr>
<td>Do you think you need more database connections?</td>
<td></td>
</tr>
<tr>
<td>Do you have sessions waiting for “eng: TX – index contention” and “buffer busy waits”?</td>
<td>RWP #15: Index Contention</td>
</tr>
<tr>
<td>Using reverse key indexes or hash-partitioned indexes to resolve index contention?</td>
<td></td>
</tr>
<tr>
<td>Suffering from cluster related wait events?</td>
<td>RWP #16: Application Induced Database Contention</td>
</tr>
<tr>
<td>Adding more RAC nodes does not improve scalability?</td>
<td></td>
</tr>
<tr>
<td>Suffering from slow “log file sync” waits?</td>
<td>RWP #17: Database Log Writer</td>
</tr>
<tr>
<td>Why use Huge Pages</td>
<td>RWP #18: Large Pages on Linux</td>
</tr>
<tr>
<td>Missing application architecture documentation?</td>
<td>RWP #19: Architecture with an AWR Report</td>
</tr>
<tr>
<td>Top-down interpretation AWR Reports</td>
<td></td>
</tr>
</tbody>
</table>

Continued on page 21
Congratulations to Denise Gardner, recipient of the APEX 2015 Educator of the Year in Technology Award, presented by the Colorado Technology Association. Tim Gorman had the pleasure of interviewing Denise regarding her award:

**When (and how) did you first become interested in technology?**

I attended Vanderbilt University in Nashville, TN, and in my sophomore year my father encouraged me to take programming classes. At the time I was focused on pre-med and business courses. Programming and technology were completely off my radar. But this was the early 1980’s and, with personal computers just emerging, my father said programming was important. I distinctly remember him saying it was the “wave of the future.” I immediately loved coding. I was drawn to the logic and problem solving. I continued taking Computer Science classes, learning Fortran, Pascal, COBOL, and Assembly. After graduating I worked for several years as a programmer and systems analyst.

**What did you think you would be doing as a career when you were young?**

I wanted to be a doctor for many years. But the reality of working in medicine dissuaded me from pursuing that field. I volunteered in Vanderbilt Hospital’s emergency room during my freshman and sophomore years. That was the best thing I could have done because I saw things that were hard to stomach. I also realized that the only times I have fainted have been in doctors’ offices and hospitals. That’s a sign! I knew that my childhood fantasy of being a doctor wasn’t a good match for me.

**How did you begin teaching technology?**

After working as a programmer and systems analyst, I joined the Peace Corps. It may seem odd to take a step “back” from a good early tech career to be a Peace Corps volunteer, but joining the Peace Corps had been a goal of mine since I was a teenager. I wanted to do the Peace Corps while I was still young and unencumbered. I served in the Peace Corps in the Federated States of Micronesia for the usual term of two years. My primary assignment was to teach English as a Second Language to middle school students. Being a teacher was another thing that was never on my radar. It never occurred to me to be an educator even though a big focus of my early volunteer work had been children. But teaching kids was a very good fit for me. I grew up in a family that very much valued education. Plus, it was always important to me to try to make a difference in the world. When I returned to the States I decided to combine the two things I enjoyed: education and computer programming.

To me, teaching Computer Science is the best of both worlds. I shape student lives and still write code. My students and I enjoy that computer programming is hands-on problem solving. We value that it is both a creative and analytical endeavor. We all become better thinkers and become more systematic and logical in problem solving. Every day my students and I are challenged and grow intellectually because of the work we do. I am certainly thankful that I can make a difference while using and developing my mind.

**What do you think is the absolutely coolest thing you’ve seen or done lately?**

Well, winning the APEX Technology Educator of the Year award was certainly cool! So many people came up to me the night of the awards dinner with congratulations. Many people asked about Pine Creek High School and the technology courses we offer. It made me feel like a rock star … and teachers never feel like rock stars.

**How did you become involved with Oracle Academy?**

In 2000, Pine Creek High School was contacted by the Oracle Academy. Oracle Academy was a new high school program and Oracle needed schools to pilot the program. Pine Creek was recommended by an Oracle employee who worked at the nearby Oracle office in Colorado Springs. I piloted the two-year program which included database design, SQL programming, and Java programming. I was also hired by the Oracle Academy as an adjunct instructor, training new OA teachers in the United States. In the second year of teaching Oracle Academy I attended RMOUG’s Training Days conference. At the time Stan Yellott was the president of RMOUG. At the conference I introduced myself to Stan and he suggested I bring my students to Training Days the next year. Ever since then my students have participated in Training Days thanks to the generosity of Stan Yellott and the RMOUG board of directors.
What do you enjoy doing in your free time?

One of the biggest contradictions in my life is that I’m not much of a techie in my private life. I got my first cell phone only four years ago! I skipped the whole flip phone era. I’m somewhat embarrassed to say that I still have a Windows 98 machine at home. I spend my time outdoors and in the gym. I enjoy swimming, cycling, hiking, camping, and snowshoeing. This past year I finally learned how to fly fish, which is something I’ve wanted to do since college. I’m passionate about travel and explore as often as possible. I also coach a year-round competitive swim team.

If you were awarded a MacArthur “genius grant” of $1m, what would you consider doing with it?

I wouldn’t say this is a “genius” idea, but it’s based on a critical need. High school students need to get real-world experience as well as an accurate idea of what working in technology is like. In high school I had a pie-in-the-sky perception of what being a doctor entailed. Doing volunteer work in medicine helped me make a better decision. If I won the MacArthur “genius grant” I would use it to give my students practical experience: I would have students create and run a school-based or district-based start-up tech company in which students provide the community with a variety of tech services such as app development, website development and hosting, coding competitions, etc. It would be a student-run organization … and I would hire a professional to manage it. That manager would not be me – I want to stay in the classroom! I would also use the money to offer my students tech-based travel opportunities across the country such as participating in national coding competitions, attending technology conferences, visiting tech companies in Colorado and on both coasts, visiting places such as The Tech Museum of Innovation in San Jose, and the like.

If you had a year to work with any technology organization, who would it be, and why?

It may sound obsequious but I’d be interested in working for Oracle. I am fascinated by Oracle’s culture and enjoy databases. The experience, I believe, would help me teach the Oracle Academy class even better by giving me a more in-depth understanding of the material. I also would like to work for a company like Google because it is the company that many of my students aspire to work for. I would like to see what Google is like so I can inform my students about the experience.

But, really, when it comes down to it I have little interest in working for a technology company. Quite often people ask me why I don’t go after a lucrative salary working in I.T. I tell them that I get immense satisfaction in helping my students define who they are and who they will become. Some students begin taking programming classes not knowing what to expect. They take introductory programming to either see what it’s like or because their parents prompted them to. It’s exciting to see them come into the class as doubters and leave as believers. They enjoy the process of programming and want to learn more. They change their idea of themselves. I also enjoy seeing the outcomes of my students. A large number of my former students have gotten college degrees in Computer Science, Computer Engineering, and Electrical Engineering. A former Pine Creek student got her Ph.D. in computer security! My students have gone on to work for Hewlett Packard, Apple, RT Logic, Northrop Grumman, Lockheed Martin, and Walt Disney World (with the Imagineers). It’s so rewarding to see them choose and be successful in technology careers.

As the winner of the APEX 2015 Educator of the Year in Technology by the Colorado Technology Association (CTA), do you have any plans for the coming year?

I am implementing a new Mobile Apps Programming course at Pine Creek High School. I also hope that the APEX award from CTA will give me a platform to draw more technology students to Pine Creek and into technology or STEM classes. To that end, I will be developing a Pine Creek STEM webpage and working on recruitment as much as possible.

What do you consider the best advice to give your students who are seeking careers in technology?

I tell students that you never know what you’re going to like so try everything. My life has been such a surprise to me. In high school I never would have guessed I would become a technology teacher! High school is the best time to try all kinds of classes including technology classes. Plus, when taking technology classes in high school, you can get a jump on college credits by taking advantage of articulation agreements and concurrent enrollment options with Colorado universities and community colleges.

Also, build a body of work. It’s a competitive process to land a tech internship or entry-level job. Starting in high school students need to build a portfolio of technology projects that they have worked on outside of the classroom. It’s not enough to simply say you have taken programming classes. Develop apps or web pages for the community around you. Partner with local schools, scout troops, clubs, and local businesses to meet their tech needs. Students also should participate in coding competitions such as Go Code Colorado (a Pine Creek programming student was a finalist last year), the Verizon Innovative App Challenge, and Google’s Code-In. The Future Business Leaders of America (FBLA) also offers various technology-focused competitions such as mobile app development, computer game & simulation programming, desktop application programming, digital video production, graphic design, 3-D animation, social media campaign, and website design.

Finally, I tell students that you don’t have to be a “techie” to be a techie. You don’t have to fit into the Big Bang Theory stereotype. You don’t have to be an obsessive, awkward coding nerd who hangs out in a basement and never sees the light of day. There are many jobs and career fields in technology other than coding. There are many kinds of people in the I.T. industry. You can enjoy studying technology and working in technology no matter what your interests are. You can be successful if you are creative, hardworking, and a problem solver.
Using SQL with Essbase Part One

Introduction and Loading Data

by Glenn Schwartzberg

Historically, Essbase developers have come from the finance and sales groups and were more comfortable utilizing flat files with Essbase. This paradigm has slowly been changing since Oracle purchased Hyperion and Essbase. IT developers are getting more involved with the creation of Essbase databases or are partnering more with the business users to help them build their applications. This article will attempt to help both the IT developer and the business user understand how SQL can be utilized for loading data into Essbase, extracting data from Essbase back to relational targets and drilling through to relational sources without creating Oracle Business Intelligence Enterprise edition (OBIEE) federated reports.

Some of the information in this article are pieces of two books I was involved in: Look Smarter than you are with Essbase Studio available on Lulu.com and Developing Essbase Applications Hybrid Techniques and Practices from CRC Press and available on Amazon.com (shameless plugs complete). This article assumes the reader knows SQL. If you are new to SQL, my chapter in Developing Essbase Applications Hybrid Techniques and Practices goes over SQL basics. The article also assumes the reader is familiar with Essbase load rules and calculation scripts at least in their most basic form. For more information on rules files and calc scripts please check out Look Smarter than you are with Essbase or the Database Administrator’s Guide.

First we will tackle the challenge of getting data into Essbase. There are two main methods for utilizing SQL to import data into an Essbase database. You can utilize built-in functionality for using relational sources to import and extract data from Essbase. Or you can utilize a product shipped free with Essbase call Essbase Studio. Essbase Studio is designed mostly for IT-type developers who are very familiar with their relational sources and star schemes. It can take relational sources and map them into the Essbase database and provide drill through (sometimes called drill back) capabilities to the relational source through the Oracle Smart View Excel add-in. This is not to say that business users can’t use Essbase Studio, but it takes a little more work on their part to understand how everything fits together. I’ve had some argue it’s easier to use Essbase Studio as the user does not have to write SQL like it is used in load rules.

This the first of a three part series will concentrate on using SQL in load rules for building metadata and loading data.

Loading Data and Metadata

Essbase utilizes load rules for building metadata and loading data. Because of the inherent flexibility with case statements, null replacements, sorting and joining, it is much easier to manage our data with SQL than using load rule functionality to do the management and manipulation of data to build dimensions and load data. Using SQL in load rules is fairly straightforward.

First, we need to set up an ODBC connection to the data source. This must be set up on the Essbase server itself and needs to be a System DSN. We should NOT set up the connection to use integrated Windows authentication. We will be passing to the rule the user ID and password we want to use. This also means a user ID and password have to be set up on the RDBMS. Preferably, this ID should have a non-expiring password like most batch IDs have. Note, it is easier to set up the ODBC connections on Windows servers. On Unix/Linux servers it is more complicated and we need to use the Essbase SQL interface guide http://docs.oracle.com/cd/E57185_01/epm.1112/ to make sure all steps are completed.

Once the ODBC connection is in place, it is easy to use the connections in a load rule.

In Essbase Administration services (EAS), create a load rule. Then from the File menu select “Open SQL”

We will get another dialog box confirming the Essbase server, application and database; just click OK on it.

SQL>UPDATE  •  Winter 2015
We will then be presented with the dialog for inputting our SQL. The SQL Dialog has two major sections: a section for connecting and a section for the actual SQL. In the connection area, we are presented with multiple ways to connect.

1. An ODBC connection
2. A substitution variable that defines an ODBC connection
3. An Oracle specific OCI connection

When using the ODBC or Substitution variable connection, the ODBC driver has to have already been configured on the Essbase server. The substitution variable connection gives us versatility when migrating between systems. In our development system, we could have an ODBC connection called “testdb” and in production the ODBC called “proddb”. On each Essbase system, I can have a variable that points to the ODBC connection name, so when I migrate between systems, it connects to the correct connection without having to edit the load rule.

For the OCI connection to work, the Oracle client has to be installed on the Essbase server box. Connecting with the OCI connection allows for faster retrievals (up to 30% faster) than using a standard ODBC connection and allows for more Oracle specific SQL syntax than the ODBC connection allows.

Finally, the fourth section labeled simply “Connect”, allows for manually creating the connection to a relational DB. Here we specify the server information for connecting.

The “Connect” option is the how Essbase Studio creates a connection. I prefer to use one of the other methods (ODBC, substitution variable, OCI) for SQL connections in EAS.

By expanding the little file folder icons, we get a larger area for inputting our SQL. Typical problems for SQL developers using this interface are:

First, there is not a SQL Editor for creating your SQL. I recommend using your favorite editor to design your SQL then copy it into the SQL interface.

This brings up the second point. See the little Select, From and Where keywords in the Editor. Foolishly, Essbase will insert these keywords in for you if you put anything in the section. This means if you had the statement

"Select * from mytable where A = B"

and you pasted

"Select *"

into the select area,

"From mytable"

into the From area and

"Where A=B"

into the Where area, you would get a resulting statement that looks like

"Select Select * From From Mytable Where Where A=B".

This obviously won’t parse correctly.

My preferred method of inserting my SQL is to ignore the From and Where boxes and put the entire SQL statement into the Select Area. This does mean we need to still drop off the first “Select” keyword so our statement would look like “* from mytable where A = B” (without the double quotes). If we are using Unions, then subsequent Select statements would include the Select Keyword. “* from mytable where A = B Union Select * from your View.” I should comment that I used an * for simplicity, but in the real world, I recommend spelling out the column names. If a column were added to a table and we are using a Select *, the load rule would end up with an extra column and not know what to do with it causing the load to fail.

As a best practice, I would recommend keeping the SQL load statement as simple as possible as it’s hard to see the transformation being done in a load rule especially for support and maintenance. Create views (if needed) in the relational to do the transformation instead of the load rule. This view can be reused as a source for other SQL loads and be used to troubleshoot in case of issues instead of writing complex queries to recreate transformations done in load rules.

Once we have the query inserted as we like it, we can see a sample of the data by clicking on the OK/Retrieve button.
so will bring us to a screen allowing us to input a username and password.

If our SQL is good and our username and password are entered correctly, we will end up with a sample of data in the load rule. Notice the column names from the SQL are used as the names of the columns in the rules file. One of the shortcuts I take is to name the columns using the AS aliases in the data load SQL to be the dimension or column names. It saves me time by then not having to name the columns in the load rule itself. This trick only works for data loads. Unfortunately this trick does not work for dimension build files.

I will give a word of caution. Not all data types are supported within the SQL interface. Of course Blobs are not supported, after all who would want to try to load a binary large or even small object to Essbase? But more common data types are sometimes not supported either. I’ve had issues with Money and other types. If the values are not showing up in the load rule, then convert it to another type like Float or Varchar.

To further enhance our SQL, Essbase allows us to use what are called substitution variables in our load rules. These variables are system, application or database defined and allow us to define a value once and use it multiple places. See the Essbase Database Administrators guide (DBAG) for how to set up substitution variables. But once set, we can use them in both the Select clause and the Where clause. For example in the following SQL, I’m doing an incremental data load using the variables &Currmt in both the Select section to bring in a column value and in the Where clause to limit the data being returned.

A few important things to note about using substitution variables in the SQL query. First as typical, the substitution variable is prefixed with an “&”. Second, we have single quotes around the variable names in the SQL. At least in the version of EAS I am working in, if I put single quotes around the variable names, EAS strips them off. Since the substitution variable is a literal in the rules file, we have to add them back in for the SQL to work properly. Third, we are using the substitution variable in both the Select clause and in the Where clause. In SQL Server, there is not an easy way to get the literal ‘JAN’ from a Datetime column, so we use Substring and convert to get just a part of the formatted date and compare it to our variable. Finally, for ease of loading, we have given the alias Sales to the amount column. This allows us to get it as the column header in the rules file.

These load rules can be used to load data and metadata either from EAS or through automation scripts utilizing Essbase’s scripting language, MaxL. Since the SQL is defined in the load rule and so is the ODBC connection, we just need to supply the statement with the SQL username and password. For you SOX-compliant concerned individuals, not to worry, the username and password can be encrypted using MaxL syntax. In plain text, the import statement to load data into an Essbase database looks like:

```
IMPORT Database 'Sample'.'Basic' DATA
CONNECT AS 'interrel' IDENTIFIED BY 'training'
USING SERVER RULES_FILE 'SQL_Prod'
ON error write to 'c:\datafiles\sql_prod.err';
```

Once encrypted it looks like:

```
Import database 'Sample'.'Basic' DATA CONNECT as $key 2217914901807526250153151431101493052640095952
2810 identified by $key 24138984909257695240924082646835900247197790
USING server rules_file 'SQL_Prod'
ON ERROR write to 'c:\Temp\sql_prod.err';
```

And we pass it a private key to do the decryption.

**Summary of Best Practices**

- Use Substitution variable for source as it makes migration between environments simple
- Don’t use Select *. Instead, list the column names - that way if the table structure changes due to any reason the load rule will not fail
- Use Views instead of complex SQL in the load rules. It helps with reusability and in debugging.
- Name the columns using the AS aliases in the data load SQL to be the dimension or column names
- Use substitution variables when possible
- Encrypt the username and password using MaxL
- Keep the SQL statement in load rules as simple as possible
Join me next time when we discuss extracting data from Essbase back to relational targets.

Glenn Schwartzberg has been an IT professional for over 30 years. He started his career in IT (then MIS) working on IBM mainframes writing COBOL and CICS programs. He quickly worked his way up the food chain and as a manager had 15 people working for him. At this pivotal juncture, he decided he would rather be a technical resource than spend his time telling other people what to do and justify his life through paperwork. He entered the world of consulting, where he was a DBA on Teradata DBC1012 computers, programming, and maintaining databases. This was his introduction to SQL.

During his stint working on data warehouses, he ran into a little program from Arbor software called Essbase. It was a cool little program and Glenn immediately saw the value in allowing users to get their own data, so he started learning all he could about it. This was on version 3.11. As time and versions progressed, he learned more and more and the versions got more feature-rich and complicated. While Glenn did not like being a manager, he did feel it was important to assist others as they tried to figure out the issues they were having with the application. He joined bulletin boards and forums and was an ardent contributor, answering others’ questions and gaining more knowledge. In 2008, due to his continued efforts to assist others and evangelize Essbase, he was honored with the title of Oracle ACE in Business Intelligence. In 2010, he was privileged to be named an Oracle ACE Director. He continues to share his knowledge of Essbase and its associated products through the Oracle EPM discussion forums, presentations at conferences, user group meetings, webcasts, and his blog. He was one of the founding members of the ODTUG Hyperion SIG Board.

The Learning Library is available for free at www.oracle.com/goto/oll.rwp.

The preceding table maps symptoms you might experience to specific videos that discuss these symptoms, root causes and possible solutions in more detail.

RWP Training Course

The second resource available is an intense hands-on training class on how to achieve optimal performance for your database applications. It is 4 days long and covers material that is usually taught in 4 months. The course is targeted for Architects, Developers and DBAs.

Topics covered include Core Database Performance, The Optimizer, Extreme Performance for OLTP and Data Warehouse Systems. The training is given by Real-World Performance Engineers and consists of a combination of classroom presentations, Demos, Hand-on tasks, Tests and Quizzes.

The class is designed to make you think about performance, how to articulate a problem with numbers/facts and propose a solution based on computer science and engineering principles.

To apply for this training course, contact RWP or your local Oracle team.

Conclusion

There are many different approaches to attack suboptimal performance, and combining that with a large set of available features and options in the database, it can be challenging to determine the best approach forward in a timely manner.

A top-down analysis based on available performance metrics can quickly identify the root cause of suboptimal performance. Combine that with the knowledge from the RWP Learning Library and Training Course, and you are in good position to improve performance by orders of magnitude, without upgrading your hardware.

Vlado Barun is a member of the Real World Performance Team at Oracle Corporation, focusing on optimizing performance and scalability for mission-critical systems.

Prior to joining Oracle Corporation, Vlado assisted a variety of organizations as DB Engineer, DBA, Database Architect and Manager of Database Development & Operations.

Graham Wood is a Software Architect at Oracle Corporation who specializes in Database design, high volume scalable systems, performance tuning and performance diagnosis.
With recent high profile attacks on data, security is at the forefront of every DBAs mind. All RDBMS have the potential to be exploited with SQL Injection attacks, as well as vulnerabilities that are unique to each product. For example, attackers will often attempt to gain elevated access to Oracle by attempting to use default user passwords. Whilst this risk can be mitigated with due diligence, with around 600 default user/passwords, it can be hard for DBAs to ensure that no stone is left unturned.

What is Threat Modelling?

As every RDBMS has many potential threats, it is important to undergo threat modelling, in order to mitigate the risks. Threat modelling is the process of identifying threats to your system before classifying and rating the threats in order to determine the most critical. You will then be in a position to determine the correct countermeasures, in order to mitigate the risks.

In an ideal world, threat modelling should be carried out during the design phase of a project and at the very least at the testing stage. In the real world, however, this often does not happen, due to time or budgetary constraints. Unfortunately, just like comprehensive backup strategies, many companies and individuals do not put an emphasis on security until it is too late.

Even in companies who have rigorous security management policies, the focus tends to be avoiding external attacks (attacks from sources external to the company) whereas it is estimated that 70 percent of security breaches are internal (attacks originating from sources within the company network). This is due to employees with malicious intent, employees who unintentionally misuse systems and from the theft of employees’ devices. Therefore, it is important that companies focus on identifying the risks of attacks from inside their network, as well as outside.

Threat modelling consists of six sequential steps; Identifying assets, creating an architecture overview, building a security profile, identifying the threats, documenting the threats and rating/prioritizing the threats.

Let’s look at how to perform threat analysis, using a fictional application called BuyMe, which belongs to the fictional company BuyMePeripherals.com and consists of a simple Web application, where customers can shop for IT peripherals. The back end of the Web application is an Oracle schema.

Step 1 – Identifying Assets

From the DBA perspective, identifying the valuable assets that must be protected, consists of identifying the company confidential that would have commercial impact if it were lost or stolen. For example, a high profile attack against an entertainment company reportedly saw the theft of roughly 76 million user accounts, leading to a cost of around $176 million.

DBAs should look to ensure that customer data, financial data and sales data are especially secure. Remember that financial repercussions could occur, not just in tangible ways, such as through fines from regulators, or compensation to customers, but also in intangible ways, such as the loss of business reputation.

Step 2 – Creating an Architecture Overview

Creating an architecture overview consists of creating a logical architecture diagram and identifying the technologies that will be (or have been) used to implement the application. This will help you identify areas of the end to end application that are potentially vulnerable, as well as identify any technology specific vulnerabilities.

In the case of BuyMe, the application consists of a Web Server, and Application Server and a Database Server. We should also note how this architecture interacts with the underlying infrastructure. The diagram below shows you how an architecture diagram for BuyMe might look.

We can now list out what technology is or will be used for each area of the topology. The table below offers an example of how this may look for the BuyMe application.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTPS</td>
<td>Used to establish connections to Web Server</td>
</tr>
<tr>
<td>IIS</td>
<td>Web Server</td>
</tr>
<tr>
<td>.NET</td>
<td>Used for Forms authentication and Wen App</td>
</tr>
<tr>
<td>IPSec</td>
<td>Used to encrypt internal traffic</td>
</tr>
<tr>
<td>Oracle 12c</td>
<td>Used to authenticate connections to the Database Server and host the back end databases</td>
</tr>
</tbody>
</table>
For a DBA, it is very easy (and intuitive) to focus entirely on the Oracle Database and the direct connections to it, but it is also important to understand the holistic application, in order to secure and test the data-tier application appropriately.

Creating a Security Profile

When creating a security profile, you will begin to identify data flows, which will in turn allow you to define trust boundaries and entry points. The BuyMe application is a simple solution which has two data flows.

The first data flow is when an internet user orders a new item. The second is from internal users, who need to update the status of customer’s orders.

Therefore, there are two clear data paths. Firstly, from the internet, via the web server, through the application server, to the Oracle database. Secondly via the application server, into the Oracle database, but originating from within the internal network.

The BuyMe application is very simple, but for more complex data flows, you will probably want to create data flow diagrams to simplify the process and ensure that there are no gaps. The data flow diagram may also prove a useful document when the application is being upgraded.

Therefore, the entry points that align to data paths can be identified as the web server (for internet users) and the application server (for internal users). It is important to remember that there is a third entry point, however, which is easy to overlook. Internal users authenticating directly to the Oracle Database.

Of course, this final entry point is intended for the use of DBAs, to manage the Database and Schema, but given that the authentication method is 2nd tier, this can also be exploited by attackers. Remember that 70% of security breaches are internal!

The trust boundaries for the BuyMe application map to the Firewalls. The data path from internet users crosses both the perimeter and internal firewalls, whereas the internal data path remains within the internal trust boundary.

Now that the application has been decomposed, you can begin to build a security profile. For the DBA, this will consist of focusing on the elements that directly face the database. This profile can then be fed into the overall security profile of the application.

For the BuyMe application, the security profile may include the following:

<table>
<thead>
<tr>
<th>Profile Element</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Validation</td>
<td>The application runs ad-hoc PL-SQL, as opposed to calling stored procedures. Therefore, the input cannot be validated at the Database*</td>
</tr>
<tr>
<td></td>
<td>As the main entry point is the web server, trust boundaries are crossed and the input cannot be trusted</td>
</tr>
<tr>
<td>Authentication</td>
<td>The database authenticates users via 2nd tier authentication. No domain authentication is required to access the schema</td>
</tr>
<tr>
<td></td>
<td>Penetration testing to ensure that all default passwords have been changed has not been carried out on this database</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cryptography</th>
<th>Data is encrypted in transit using IPSec</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Data is encrypted at rest using Transparent Data Encryption</td>
</tr>
<tr>
<td></td>
<td>No column level encryption is used</td>
</tr>
<tr>
<td>Auditing</td>
<td>Default actions are audited</td>
</tr>
</tbody>
</table>

*There may be (and should be) input validation on the application side, but the DBA is unlikely to have visibility of this

Identifying Threats

Now that a security profile is in place, we can work to identify potential threats in our application. This will usually involve performing a penetration test. There are many penetration testing tools available, including Qualys, which can be obtained from www.qualys.com and Metasploit, which can be obtained from www.metasploit.com. The threats that are revealed by the penetration test can then be categorised using STRIDE methodology. STRIDE stands for Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service (DoS) and Elevation of privileges.

Spoofing identity refers to stealing another user’s identity and using this identity to authenticate, as opposed to your own identity. The BuyMe application is particularly susceptible to this, because the application server uses a single user to authenticate to the database and because inputs cannot be validated at the database tier.

Tampering with data refers to the practise of maliciously modifying data. In the context of the overall application, this could refer to attacks including cross site scripting and manipulating HTTP headers. From the DBA perspective, however, tampering with maliciously modifying data stored within the schema. For example, in the case of the BuyMe application, a malicious user may attempt to amend their account balance to zero.

Repudiation describes a malicious user’s ability to hide or deny their actions. This is critical, because if repudiation is possible, you may not be aware that an attack has taken place. Even if you are aware, it may be impossible to prove. Repudiation is an issue for the BuyMe application, because only default auditing is implemented. This means that the only actions that will be captured are database start-up, database shutdown and connections with administrative privileges.

Information disclosure is arguably the most widely known threat. It refers to the practise of “stealing data”. This occurs when an attacker forces a system to reveal more data than they have the permissions to view. As with spoofing identities and tampering with data, the BuyMe application is susceptible to this form of attack because the database layer does not validate the inputs.

Denial of service (DoS) attacks are those which attempt to flood a system with requests, with the aim of either taking down the system, or making the system appear to be down, due to its inability to deal with the volume of requests being received. DoS attacks are one of the most common form of attack and in today’s world are becoming increasingly complex, so they should be taken into account during every threat modelling exercise.

Elevation of privileges is the act of exploiting a system to gain more permissions than you were intended to have. The fact that the
security profile has revealed that penetration testing has not taken place, to ensure that all default passwords have been changed, leaves the BuyMe application at risk of this kind of attack.

Just like any RDBMS, there will also be known vulnerabilities, which can be exploited. These should be addressed wherever possible, usually through patching the system. If no patching is currently available, then at a minimum, you should consider implementing auditing and alerting, specifically tailored to the vulnerability. The BuyMe application uses Oracle 12.1.0.2, so there is a risk of an execution code overflow attack taking place.

We should list and document the potential threats against our application. I tend to use a table, such as the one below.

<table>
<thead>
<tr>
<th>Risk</th>
<th>Category</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Injection</td>
<td>S,T,I</td>
<td>Attacker types ' OR 1=1-- In password field of the web site to spoof the first User identity stored in the users table</td>
</tr>
<tr>
<td>DoS</td>
<td>D</td>
<td>Attacker uses robots to simultaneously flood the database with resource intensive requests</td>
</tr>
<tr>
<td>Stealing Credentials with default passwords</td>
<td>E</td>
<td>Attacker attempts to login with default user password of a privileged user</td>
</tr>
<tr>
<td>DBA performs malicious action</td>
<td>R</td>
<td>A disgruntled privileged user performs a malicious action and the attack cannot be proven, due to lack of auditing*</td>
</tr>
<tr>
<td>Exec Code Overflow</td>
<td>T, I</td>
<td>Attacker exploits known vulnerability DBMS AW.EXECUTE to execute malicious code using CDA commands</td>
</tr>
</tbody>
</table>

*Whilst this type of attack sounds a little farfetched, it is more common than you may think. I am aware of two separate companies that have fallen foul of this in recent times. In one instance, on a DBA's last day, he dropped a key database. In the other instance, a SQL Server DBA obfuscated all programmable objects, before leaving the company.

Rating Threats

Once threats have been identified and classified, you should begin the process of rating the threats, based upon the probability of the attack occurring and the potential damage that the threat could inflict. There are various methodologies used for rating threats.

The simplest method for rating threats is a straight High, Medium, Low system. With this system, each threat will be given a rating, based on your opinion. There are two issues with this approach, however. Firstly, it makes the rating system subjective, as opinions are just opinions and not necessarily correct. Secondly, opinions often differ, therefore it can be hard to gain consensus on the priority with which the threats should be addressed.

A slightly more scientific approach is to use a Critical, Important, Moderate, Low system. This system offers more categories, which can aid prioritization, where there are a large number of threats. A critical threat is usually defined as a threat that allows an attacker to penetrate a system without any alerts or warnings being fired and where there is precedence of this attack being performed.

An important threat is usually regarded as a threat where data could be compromised by an attacker and it would be easy for an attacker to exploit the vulnerability, if discovered. With threats in this category, there is often a precedence for similar vulnerabilities being exploited.

A moderate threat is often categorized as a threat where it is possible for an attacker to exploit the vulnerability, however, the risk is mitigated by factors such as integrated authentication and it would be difficult for an attacker to exploit the weakness.

A low threat is normally regarded as a one where the likelihood of the vulnerability being exploited is very low, due to existing infrastructure or countermeasures that are in place. Often threats that are categorized as low will not be addressed, as it will be decided that the cost of addressing them outweighs the potential costs of the attack being exploited.

Whilst pragmatically, ignoring a threat with a low rating is fine, as we all understand that budgets and timescales are always important factors, I do like to remind management and budget holders of the analogy involving the Fukushima nuclear disaster in 2011. The risk analysis when building this plant reportedly factored in protection against an earthquake and protection against a tsunami. The risk of two earthquakes and a tsunami occurring at the same time, however, was regarded to unlikely to require consideration. The first earthquake was within the designed tolerance of the reactors, but following the second earthquake and tsunami, the Fukushima plant largely melted in three days.

Another common system for threat rating is to use a damage potential * probability formula. Using this technique, you will rate the damage potential of each threat using a scale of 1 to 10, where 1 means that an attack exploiting this particular vulnerability would cause only minimal damage and 10 indicates that an attack exploiting the particular vulnerability would be a catastrophe.

You will then rate the likelihood of the threat being realised on a scale of 1 to 10. Here, 1 indicates that there is very little chance of the threat being realised and 10 means that it is almost certain.

Once the two ratings for each threat have been established, you will multiply the damage potential rating by the probability rating for each threat. This will give your threats a priority score on a scale of 1 to 100.

My preference for rating threats is to use a system called DREAD. Whilst being a little old school, I find it the best, most comprehensive fit for data-tier applications. DREAD stands for Damage potential, Reproducibility, Exploitability, Affected users and Discoverability.

Damage potential rates the damage potential of each threat using a scale of 1 to 10, where 1 means that an attack exploiting this particular vulnerability would cause only minimal damage and 10 indicates that an attack exploiting the particular vulnerability would be a catastrophe.

Reproducibility rates how easy it would be for an attacker to repeatedly reproduce the attack on a scale of 1 to 10. 1 indicates that is would be almost impossible to reproduce and 10 means that it would be very easy to reproduce an attack. The easier it is to reproduce an attack, the more likelihood there is of automated “bots” being used to systematically attack the system.

Exploitability rates the ease in which an attack could exploit the vulnerability using a scale of 1 to 10. 1 indicates that the vulner-
ability would be extremely difficult to exploit, due to factors such as prior authentication being required. A rating of 10 indicates that an attacker could exploit the vulnerability with ease.

Affected users rates the number of users that would be affected by the threat being discovered on a scale of 1 to 10. To calculate the rating, take the percentage of users that would be affected, divide by 10 and then round to the nearest whole number. For example, if 80 percent of users would be affected, then the rating would be 8. If only 25 percent of users would be affected, then the rating would be 3.

Discoverability rates how easy it would be for an attacker to discover the vulnerability on a scale of 1 to 10. A rating of 1 means that the vulnerability is obscure and an attacker would be unlikely to stumble across it, or realise its potential. A rating of 10 would indicate that the vulnerability can easily be discovered. For example, it may be a well-known, documented attack strategy, such as SQL Injection.

Once each threat has been given a rating in each of the DREAD categories, the ratings should be summed and then divided by 5, before being rounded to the nearest whole number. This will give you the overall DREAD rating for each threat. Let’s use the threats we identified earlier, and rate them using DREAD. The risks have been ordered by their DREAD rating.

<table>
<thead>
<tr>
<th>Risk</th>
<th>Category (STRIDE)</th>
<th>D</th>
<th>R</th>
<th>E</th>
<th>A</th>
<th>D</th>
<th>Threat Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Injection</td>
<td>S,T,I</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Stealing Credentials with default passwords</td>
<td>E</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>DoS</td>
<td>D</td>
<td>6</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>DBA performs malicious action</td>
<td>R</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>Exec Code Overflow</td>
<td>T, I</td>
<td>8</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

We can see that the risk of SQL injection attacks, permission elevation with default passwords and denial of service attacks should be addressed immediately. The risk of DBAs performing malicious actions and the exec code overflow bug being exploited should still be address, but with lower priority.

Creating Countermeasures

Our security modelling is now complete and we should start to consider what countermeasures we can put in place for each of the risks that we have identified, starting with the threats that have the highest DREAD rating.

Mitigating the risk of SQL Injection involves validating the inputs received. This should be performed at the application, but it is important to remember that the DBA is the last line of defence against attacks. Therefore, we should review how the application is connecting to the database. We identified that the application is running ad-hoc queries and we could reduce the risk of SQL Injection attacks by introducing a hosting standard that requires applications to access data within the schema using stored procedures, as opposed to ad-hoc SQL. We can then ensure that the stored procedure is validating the values passed to its parameters.

The risk of an attacker gaining elevated privileges by authenticating using a default user password can be removed by ensuring that all default passwords have been changed. A default password list can be obtained from various sources, including http://www.petefinnigan.com/ This resource allows you to download the default password list in various formats, including .csv and a .sql script which inserts all default passwords into a table. You can then create a script which attempts to login to the database using each of the default passwords in the list. If any users with default passwords are identified, you can then change them.

Denial of service attacks are one of the most difficult to protect against. This goes part way to explaining why they are one of the most common form of attack. One way to reduce the risk is to ensure that the database server is not directly exposed to the internet, and this security best practise is already in place for the BuyMe application.

We could further reduce the impact of a DoS attack by implementing Oracle Resource Manager. This would allow us to limit the resources that were consumed by attacker’s requests. If the application tier were written using Java EE, then we could also use WebLogic server to reduce network traffic. As the application layer is .NET, however, this approach is not feasible in our case.

If a rouge DBA decided to attack the database, then there is very little that we can do to stop it. What we must do, however, is ensure that we have a non-reputability strategy in place. This involves using auditing to ensure that malicious actions are traceable. This serves two purposes. The obvious reason is that we can prove what happened and take appropriate action. Less obviously, the fact that we can prove and take action against a malicious DBA will potentially act as a deterrent. “Soft” security measures should also be reviewed, such as disabling user accounts when a staff member leaves. Currently the BuyMe application has default auditing only. We should consider implementing fine-grain auditing, to ensure reputability.

Because the DREAD rating for the Exec Code Overflow attack is low, and specifically due to the obscurity of the vulnerability, we (or management) will likely decide that we should not take immediate actions to mitigate the risk, as appropriate countermeasures will likely prove cost prohibitive, compared to the likelihood of the vulnerability being exploited. We should keep the risk logged in our company’s risk register and patch the database, when a patch becomes available.

Peter Carter is a database consultant, trainer and author, with more than a decade of experience. His new book Pro SQL Server Administration is now available in paperback and e-book formats at http://www.apress.com/9781484207116?gtmlf=c.
Hello, my name is Bob Mason. I am the vice-president for the Rocky Mountain Oracle User Group (RM0UG) Board of Directors (BOD).

I began my journey as an Oracle user (with version 7) in 1995 working in downtown Denver at US West Communications (currently known as Century Link). Prior to working with Oracle, I worked a decade as a software engineer using the Information Engineering Facility (IEF) Case Tool by Texas Instruments which generated COBOL application code on the DB2 mainframe platform. Although the IEF Case Tool eventually faded away into technology history, it did last long enough to provide me with consulting opportunities that enabled my wife and I to work and live overseas in New Zealand and Australia for 18 months. This happened before we decided to have children and I currently have two sons (Ben and Alex) that are 17 and 21 years old. I also have a lovely wife Judy and we just celebrated our third wedding anniversary in Hawaii.

I became an Oracle zealot after using the product and I continued to pursue opportunities using Oracle in my professional career after 1995. In 1998, I learned about the Rocky Mountain Oracle Users Group (RMOUG) and I started attending meetings on a regular basis. At one of the meetings, I met Don Archer, a professor from Regis University who was searching for instructors to teach Oracle courses within the Master of Science in Computer Information Systems program. Since I had completed my MBA (1992) at the University of North Texas in Denton, Texas with an emphasis in Computer Information Systems, I was able to accept the part-time teaching opportunity. For a decade (1998 – 2008), I taught a variety of Oracle courses at Regis as an affiliate faculty member. The courses varied in subject, such as Database Administration, PL/SQL and Oracle Forms & Reports; I always enjoyed PL/SQL and Oracle Forms the most. I spent some of my time developing new Oracle courses and advised students with their MS Thesis projects that involved Oracle. I was working full-time during the infamous .com bubble (around the year 2000) and I remember when IT developers were elevated from back office workers to the status of “Rock Stars”. I enjoyed those days of free catered employee breakfasts on Friday mornings, large retention bonuses and class sizes of 25 students with a wait list. The phenomenon lasted until the .com bubble burst and as the song goes - “Those were the days.”

I continued with my Oracle career and attended RMOUG meetings. I presented white papers on various Oracle topics at RMOUG Training Days (2001 – 2009). Occasionally, I volunteered at Training Days as a room ambassador, so that I could support the event. As I’ve told numerous colleagues over the years, “RMOUG Training Days is the best Oracle Training in the Denver area at the least cost.” RMOUG has always been a good way for me to meet like-minded Oracle professionals and more importantly – to keep informed about the latest Oracle functionality.

I was ready for a career change after positive feedback from my students about my teaching abilities. I have been a manager and team lead at several large companies. I have the highest admiration for IT managers at all levels, but management didn’t fit my career path. Therefore, after decades of working in IT, I decided to pursue a Ph.D. in Computer Information Systems at Nova Southeastern University in Ft. Lauderdale, FL. This degree allowed me to acquire the credentials necessary to become a full-time university professor.

In 2011, I completed my Ph.D. in Computer Information Systems and accepted a full-time ranked faculty position as an Assistant Professor at Regis University. For the first three years, I was the Program Coordinator (Director) for the MS in Database Technologies (DBT). I later accepted the role as the Program Coordinator for the MS in Software Engineering and Database Technologies (SEDT) program which is a joint program with the National University of Ireland in Galway (NUIG). I have travelled to Ireland several times to work with the NUIG faculty. Therefore, I currently manage both MS programs. In addition to teaching Relational (e.g. Oracle) and NoSQL database courses to the MS students, I am the advisor for 60 DBT and SEDT students. I manage the affiliate faculty, schedule the database courses and resolve student and faculty grade disputes. Also, I manage the Database Practicum that is a four month program that simulates...
a real-life internship for DBT and SEDT students. I participate as the Regis College of Computer & Information Sciences faculty representative on the Regis University President’s Advisory Council (PAC) and I’m heavily involved with the Higher Learning Commission (HLC) accreditation process for Regis University. I upgrade the existing Oracle courses and I develop new courses (e.g. NoSQL Databases) to keep the DBT and SEDT programs current with the latest technologies.

To say that RMOUG has played a major role in my life and career development is an understatement. It would be nice if we all had a road map for our life. However, life events often change our roles. In 2016, I will be promoted to Associate Professor at Regis University.

I’m very happy that I joined RMOUG in 1998 and that RMOUG helped to shape my career choices and to achieve my dreams.

Tell Us About Yourself

Join us in sharing your Oracle experiences with other RMOUG members!

Tell us about your life, your job, or share your amusing Oracle anecdotes, tips and secrets!

Share your favorite photos with our members and have the chance to be published on our cover!

Prior to each issue, we ask members to present their favorite photo, regional to the Rocky Mountains and not previously published. Each issue has a theme - Spring, Summer, Fall and Winter - and we need a minimum format of 300 dpi at least 8.75” x 11.25”. So turn your camera vertically and imagine the words RMOUG SQL>UPDATE across the top!

Please submit all material to NewsletterDir@rmoug.org

RMOUG sincerely appreciates all who contribute to SQL>Update!
Meet Your Board

Tim Gorman  
President  
E-mail: President@rmoug.org

Bob Mason  
Vice President  
E-mail: VicePresident@rmoug.org

John Jeunnette  
Treasurer  
E-mail: Treasurer@rmoug.org

Chris Chase  
Secretary  
E-mail: Secretary@rmoug.org

Bobby Curtis  
Social Media Director  
E-mail: SocialMediaDir@rmoug.org

Komal Goyal  
Marketing Director  
E-mail: MarketingDir@rmoug.org

Dan Hotka  
Programs Director  
E-mail: ProgramsDir@rmoug.org

Art Marshall  
Web Director  
E-mail: WebDir@rmoug.org

Sruthi Kumar Annamnidu  
SIGs Director  
E-mail: SigsDir@rmoug.org

Mark James  
Scholarship Director  
E-mail: ScholarshipDirDir@rmoug.org
RMOUG Board of Directors

Karen Funston
Training Days Technology
E-mail: TrainingDaysDir@rmoug.org

Chris Ostrowski
Membership Director
E-mail: MembershipDir@rmoug.org

Kellyn Pot’Vin
Training Days Director
Non-Voting Board Member
Emeritus
E-mail: TrainingDaysDir@rmoug.org

Giri Venkata
Board Member At Large
E-mail: MembersAtLarge@rmoug.org

Susan Llanes
Board Member At Large
E-mail: MembersAtLarge@rmoug.org

Al Gonzales
Board Member At Large
E-mail: MembersAtLarge@rmoug.org

Rene Antunez
Board Member At Large
E-mail: MembersAtLarge@rmoug.org

Vincent Giasolli
Board Member At Large
E-mail: MembersAtLarge@rmoug.org

Pat Van Buskirk
Newsletter Director
E-mail: NewsletterDir@rmoug.org

Kathy Robb
Board Member Emeritus
Arisant, LLC
E-mail: BoardME@rmoug.org

Peggy King
Executive Director
E-mail: Peggy.King@rmoug.org

Ann Horton
Oracle Liaison
E-mail: OracleLiaison@rmoug.org
David Peake, Oracle APEX Product Manager, will lead you down the APEX path with hands on exercises building an interactive APEX application both with forms and reports. Laura Ramsey of Oracle Technical Network will be on hand with a short presentation on all the features, tutorials, information, videos, and product that is available via the Oracle Technical Network. She will bring shirts and other Oracle chacha as well. This meeting is currently scheduled for Friday May 6, 2016 at the Broomfield Oracle Corp offices. We will have further details as the time grows closer.

Bring your laptops and be prepared to learn!

Watch for RMOUG’s email announcement for your opportunity to register for the May QEW
Breakfast Discounted Total Cost

Ad Rate

1/4 Page $350.00 $175.00 $ 525.00
1/2 Page $350.00 $312.50 $ 662.50
Full Page $350.00 $500.00 $ 850.00
Inside Cover $350.00 $625.00 $ 975.00
Back Cover $350.00 $750.00 $1,100.00

Contact Dan Hotka - Programs Director - ProgramsDir@rmoug.org

Reach A Targeted Oracle Audience • Advertise Now!

A full page, full color ad in RMOUG SQL>UPDATE costs as little as 70 cents per printed magazine and even less for smaller ads.

Contact NewsletterDir@rmoug.org

SPONSOR A QUARTERLY EDUCATION WORKSHOP AND RECEIVE A HALF PRICE AD IN SQL>UPDATE

Join Us May 6, 2016 at the Oracle Broomfield Campus!

More Details www.rmoug.org

Contact Dan Hotka - Programs Director - ProgramsDir@rmoug.org
HIGHLIGHTS

• Two full days of training with a bonus half-day of two-hour deep-dive and 30-minute tech-tip sessions
• More than 100 technical sessions with new tracks
• Full exhibition
• ACE luncheons and WIT panel
• Networking opportunities to meet and talk with Oracle ACE Directors and ACEs, product managers from Oracle, and other Oracle professionals

KEYNOTE

Knowledge Is Power: 15 Years of SQL Tuning from coe_xplain to SQLTXplain to SQLd360

Confucius once said: “Life is a learning experience, only if you learn.” Join Mauro and Carlos as they discuss the history of how these tools came about. From legendary coe_xplain.sql, which saved the day for so many DBAs in Oracle ancient times, to modern SQLd360, which is getting rapidly adopted by DBAs and Developers today, Carlos and Mauro certainly understand what it takes to overcome database challenges.

From filling the holes some legacy Oracle products had to automating a lengthy diagnostics gathering process for SQL Tuning, they learned not only technical intricacies but also equally important aspects of human interaction.

Carlos and Mauro will discuss what they’ve learned during their invaluable 15-year journey that brought them to SQLd360. They’ll share with the audience the most memorable wins as they developed these invaluable tools and the biggest challenges they faced. Come learn what was the motivation to create these tools, (along with maintaining.) There was certainly a need for these tools for both customers and support, but how did that trigger this multi-year effort, commitment and assurance that they remain free? Don’t miss this astounding keynote on the history of these tools and the stories from those that created them.