RMOUG SQL>UPDATE

The Magazine for the Rocky Mountain Oracle Users Group • Vol 65 • Winter 2011

Tom Kyte
How Do You "Make It Faster"?

Saurabh Gupta
Inheritance In Oracle

Kellyn Pot'Vin
MySQL For Oracle DBAs

Training Days 2012
Early Registration Ends January 12

Board Focus - Pat Van Buskirk
Member Focus - Mark Molnar

Rocky Mountain Oracle Users Group
PO Box 621942
Littleton, CO 80162

Change Service Requested

RMOUG TRAINING DAYS
February 14-16, 2012 Denver, Colorado
Colorado Convention Center
www.rmoug.org
HE COLLEGIATE PEAKS. PHOTO COURTESY OF JOHN KING, RMOUG MEMBER

Advance Registration Through January 12!

Exhibitor Opportunities Still Available

Volunteers Needed!

If you have any questions, feel free to contact me, John Jeunnette, at trainingdaysdir@rmoug.org, or Lindsey Jacks at 910.452.0006 or Lindsey@teamycc.com. We look forward to seeing you at Training Days.

Information & Registration

http://www.teamycc.com/RMOUG_2012_Conference/Registration.html

RMOUG
ROCKY MOUNTAIN ORACLE USERS GROUP

EDITOR & DIRECTOR
Pat Van Buskirk
NewsletterDir@rmoug.org

SUBSCRIPTIONS & TECHNICAL ADVISOR
Heidi Kuhn
admin@rmoug.org

CONTRIBUTING WRITERS
Saurabh Gupta
John Jeunnette
Tom Kyte
Mark Molnar
Kellyn Pot'vin
Pat Van Buskirk

SQL>Update is published quarterly
by
Rocky Mountain Oracle Users Group
5409 S. Union Court
Littleton, CO 80127
303-948-1786
Email: admin@rmoug.org
www.rmoug.org

Please submit advertising and editorial material to the NewsletterDir@rmoug.org. RMOUG reserves the right to edit, revise or reject all material submitted for publication. Liability for errors in ads will not be greater than the cost of the advertising space. ©2011 Rocky Mountain Oracle Users Group

contents

features

monthly features

departments

A special Thank You to Heidi Kuhn, without whose continuing help, this publication would not be possible.

Another special thank you to Jed Walker, Senior Manager, CMC Database Operations of Comcast, who suggested using a fixed width font for system output. This aligns the text and makes it much easier to read. Thank you, Jed.

On the Cover:
Hallett Peak and Emerald Lake on March 3, 2010 by Michael Landrum. Just a short snow show from the Bear Lake trail head in Rocky Mountain National Park. The picture was taken while Michael was out enjoying his favorite pastime of hiking in the Colorado mountains with his cameras. Michael has been in the database business since 1985 and has been working exclusively in Oracle as a Pl/Sql developer since 1996. He is currently enjoying the roll of data Tsunami at I-Behavior.

From The President

Tim Gorman

An old joke re-told by President Ronald Reagan a lifetime and a half ago...

Two guys are hiking in the woods when they come across a huge grizzly bear, standing on its hind legs in the middle of the trail, baring its fangs and roaring at them. Immediately, one of the hikers drops to the ground, pulls a pair of sneakers from his backpack, pulls off his hiking boots, and begins lacing up the sneakers. The other hiker, apprehensively watching the bear, who is now advancing menacingly, asks, "You don't believe you can outrun him, do you?" The other hiker stands up and says, "No, I only have to outrun you..."

Although I hope I (and all of you) never experience these particular circumstances, everyone certainly must recognize how this vignette pertains in our professional lives.

In our world of information technology, change is constant and wide-reaching enough so that almost everything does a complete turnover every couple of years. When was the last time you began a new project using the "waterfall" method? How many offices are still "hard-wired"? How many IT workers telecommute from home when a snowstorm threatens, or all the time? How many have adopted tablets over laptops? How many teams have members eight or twelve time zones away? How has the technology you use day to day changed in the past couple of years?

More than keeping up with the tempo of change, the tight economy raises the ante further, bringing us closer to the analogy of out-running everyone else outrunning a grizzly bear. It ain't pretty, but we all feel it and know it.

In conditions like these, continuing education becomes more important, not just building our skills, but also helping us meet day-to-day job demands. It is certainly possible to learn all things new while on the job, but the sheer volume of things new makes that a challenge. Five-day training courses are rarely made available, so many of us learn from the internet, from blogs, from white papers, from Google searches, even though we still prefer learning face to face.

This is where the sheer genius and incredible importance of the RMOUG "Training Days 2012" conference shows. The conference itself, over two days, consists of a dazzling array of 143 presentations by over 70 speakers from Colorado, around the US, and around the world -- all for less than a tablet.

RMOUG has added a third day, when RMOUG members and attendees are treated to two fantastic and effective new training opportunities: the Oracle TechNet (OTN) "Developer's Day" and the RMOUG "University Sessions".

- (NEW THIS YEAR!) OTN Developer's Day is a full day of presentations from the folks at Oracle TechNet, presenting the latest from the best, for FREE
- (BACK FOR THE 6th YEAR!) RMOUG University Sessions are 4-hour afternoon sessions by some of the top technical experts in the world, for an affordable additional fee

Here's another opportunity: RMOUG has made a tradition of meeting out-of-town speakers at the Denver Int'l Airport and driving them to their hotels. Besides saving the unpaid volunteer speakers about $50 in a one-way taxi fare, meet n' greeters have the opportunity to chat with speakers and get to know them personally. These speakers are developers, database administrators, and managers just like us, and many of them thoroughly enjoy visiting Denver due to this intimate personal contact by the RMOUG community.

As a further opportunity, the "Training Days" conference needs volunteers to help with registration and as "room managers" for each of the speaking rooms. These types of volunteers receive discounts on conference registration, ranging from half to full discounts on conference registration.

If you're interested in volunteering to meet n' greet speakers as they arrive for the conference or depart back home, or if you're interested in volunteering to help with onsite conference registration or as a "room manager", please contact Peggy King at peggy@kingtraining.com or pastpresident@rmoug.org.

Continuing education keeps us one ahead of the pack, abreast of a fast-changing environment. Instructor-led training, face to face, is the most effective way to absorb new technology and techniques and the "Training Days 2012" conference has literally dozens of hard-hitting, fast-acting seminars. It is an economical way to take advantage of over 150 different topics over three days of intense education. Plan now, keep an eye on the RMOUG website for news, and be ready for it when it's here.

Or else be sure to keep a pair of sneakers handy.

See you in February!

Tim

RMOUG Training Days 2012 Coming Soon...

We are on the final stretch to a great Training Days, February 14-16, 2012 at the Colorado Convention Center in Denver.

We are honored to have over thirty Oracle Aces and Ace Directors in attendance. This is more than most national conferences, so come meet the experts and get your questions answered or just say hello.

A first this year is the full day Oracle Technology Network Developer Day on Tuesday.

Repeating this year are the ever popular RMOUG University Sessions providing in-depth coverage of a variety of topics. If you use TOAD, work with a data warehouse, are learning Java, or use Esbase there is a University Session for you.

See http://www.teamycc.com/RMOUG_2012_Conference/index.html for all the details on the RMOUG University sessions, presenters, and presentations planned for the conference. (Don't miss the all-important "Register" button.)

See you in February!

John Jeunnette, 2012 Training Days Conference Chair

Volunteer Update

All Ambassadors need to plan on attending both days of Training Days 2012 and please try to attend the Informational Meeting to be held on February 14th (details to come). We still need help with RMOUG Greeters – we are asking for volunteers to meet the out of town speakers at the airport and give them a ride to their hotel. We also will need help with rides back to the airport after the conference. If you are interested in helping as a RMOUG Greeter, please let me know.

Room Monitor Ambassadors - We are looking for 24 people to volunteer as a Room Monitor Ambassadors. In return for your commitment of time (a total of 7.5 hours) to monitor sessions by room assignments. You will receive one complimentary conference registration.

Registration Ambassadors - We are looking for 16 people to volunteer as Registration Ambassadors. In return for your commitment of time you will receive a 50% discount on one conference registration. All Registration Ambassadors will assist with registration from 6:30 am- 8:45 am on Wednesday, February 15th (plan on arriving by 6:00 am) and commit to one additional hour for each day of the conference.

Contact **peggy@kingtraining.com** for more information.

PLEASE NOTE: The University Sessions are an additional event and the fees for the University Sessions are not discounted. Should you want to attend any of the sessions offered on February 14th the fees are indicated on the registration form.

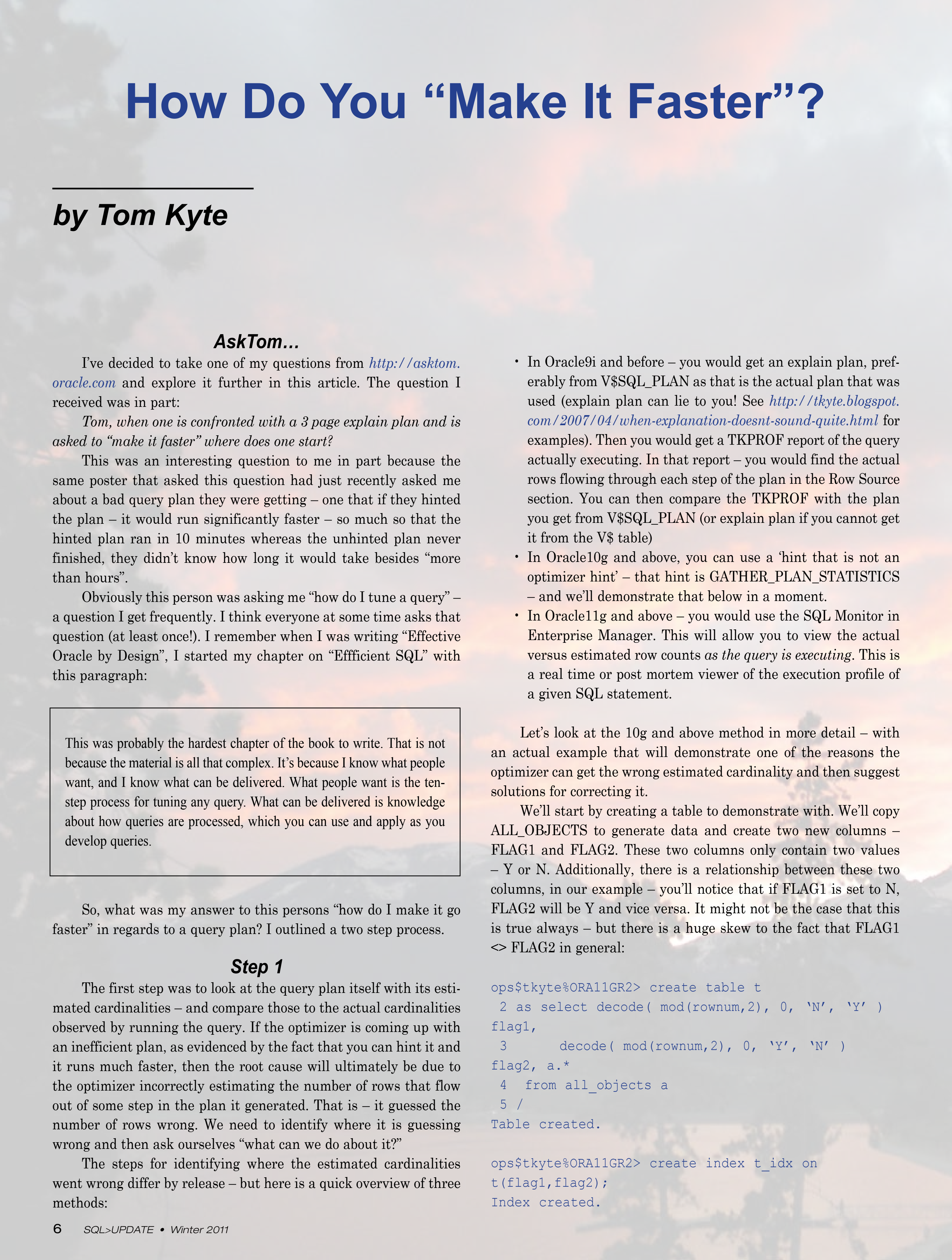
Stan Yellott Scholarship Fund

Eligibility Requirements:

- Registered or the intent to register at an accredited post secondary educational institution in the United States
- Minimum GPA 2.5 from a current official transcript
- Currently enrolled in a computer science/information technology class and/or intends to enroll in a computer science the following term

Award notification will be given within 45 days following application deadlines. Upon acceptance of the scholarship, additional information may be required.

For Details, Visit the RMOUG Website
www.rmoug.org



How Do You “Make It Faster”?

by Tom Kyte

AskTom...

I’ve decided to take one of my questions from *http://asktom.oracle.com* and explore it further in this article. The question I received was in part:

Tom, when one is confronted with a 3 page explain plan and is asked to “make it faster” where does one start?

This was an interesting question to me in part because the same poster that asked this question had just recently asked me about a bad query plan they were getting – one that if they hinted the plan – it would run significantly faster – so much so that the hinted plan ran in 10 minutes whereas the unhinted plan never finished, they didn’t know how long it would take besides “more than hours”.

Obviously this person was asking me “how do I tune a query” – a question I get frequently. I think everyone at some time asks that question (at least once!). I remember when I was writing “Effective Oracle by Design”, I started my chapter on “Effficient SQL” with this paragraph:

> This was probably the hardest chapter of the book to write. That is not because the material is all that complex. It’s because I know what people want, and I know what can be delivered. What people want is the ten-step process for tuning any query. What can be delivered is knowledge about how queries are processed, which you can use and apply as you develop queries.

So, what was my answer to this persons “how do I make it go faster” in regards to a query plan? I outlined a two step process.

Step 1

The first step was to look at the query plan itself with its estimated cardinalities – and compare those to the actual cardinalities observed by running the query. If the optimizer is coming up with an inefficient plan, as evidenced by the fact that you can hint it and it runs much faster, then the root cause will ultimately be due to the optimizer incorrectly estimating the number of rows that flow out of some step in the plan it generated. That is – it guessed the number of rows wrong. We need to identify where it is guessing wrong and then ask ourselves “what can we do about it?”

The steps for identifying where the estimated cardinalities went wrong differ by release – but here is a quick overview of three methods:

- In Oracle9i and before – you would get an explain plan, preferably from V$SQL_PLAN as that is the actual plan that was used (explain plan can lie to you! See *http://tkyte.blogspot.com/2007/04/when-explanation-doesnt-sound-quite.html* for examples). Then you would get a TKPROF report of the query actually executing. In that report – you would find the actual rows flowing through each step of the plan in the Row Source section. You can then compare the TKPROF with the plan you get from V$SQL_PLAN (or explain plan if you cannot get it from the V$ table)
- In Oracle10g and above, you can use a ‘hint that is not an optimizer hint’ – that hint is GATHER_PLAN_STATISTICS – and we’ll demonstrate that below in a moment.
- In Oracle11g and above – you would use the SQL Monitor in Enterprise Manager. This will allow you to view the actual versus estimated row counts *as the query is executing*. This is a real time or post mortem viewer of the execution profile of a given SQL statement.

Let’s look at the 10g and above method in more detail – with an actual example that will demonstrate one of the reasons the optimizer can get the wrong estimated cardinality and then suggest solutions for correcting it.

We’ll start by creating a table to demonstrate with. We’ll copy ALL_OBJECTS to generate data and create two new columns – FLAG1 and FLAG2. These two columns only contain two values – Y or N. Additionally, there is a relationship between these two columns, in our example – you’ll notice that if FLAG1 is set to N, FLAG2 will be Y and vice versa. It might not be the case that this is true always – but there is a huge skew to the fact that FLAG1 <> FLAG2 in general:

```
ops$tkyte%ORA11GR2> create table t
 2 as select decode( mod(rownum,2), 0, 'N', 'Y' )
flag1,
 3       decode( mod(rownum,2), 0, 'Y', 'N' )
flag2, a.*
 4  from all_objects a
 5 /
Table created.

ops$tkyte%ORA11GR2> create index t_idx on
t(flag1,flag2);
Index created.
```

```
ops$tkyte%ORA11GR2> begin
 2      dbms_stats.gather_table_stats
 3      ( user, ‘T’,
 4       method_opt=>’for all indexed columns size
254’ );
 5 end;
 6 /
PL/SQL procedure successfully completed.
```

Now, if we were to ask the optimizer “how many rows are in the table” – it would be able to estimate that accurately. If we were to ask the optimizer “how many rows are in the table such that FLAG1 = ‘N’” – it would also be able to accurately estimate that – the same would be true for FLAG2 = ‘N’. However, if we ask the optimizer to estimate how many rows would be returned by FLAG1=’N’ and FLAG2=’N’ – it would completely drop the ball right now. It has a ton of information about FLAG1, a ton of information about FLAG2 – but it knows nothing about FLAG1 and FLAG2 *together*. The optimizer will use simple statistical analysis methods to guess how many rows would be returned – in this case it will guess that since FLAG1=’N’ returns 50% of the data and FLAG2=’N’ returns 50% of the data – the two together would return 50% x 50% or 25% of the table – and we’ll see that in the plan (note: my table had about 72,000 rows in it)

```
ops$tkyte%ORA11GR2> select * from t where flag1=’N’
and flag2=’N’;

Execution Plan
--------------------------------------------------
Plan hash value: 1601196873

--------------------------------------------------
| Id | Operation        | Name | Rows | Bytes | Cost
(%CPU)| Time    |
--------------------------------------------------
|  0 | SELECT STATEMENT |    | 18045 | 1797K|  299
(1)| 00:00:04 |
|* 1 | TABLE ACCESS FULL| T   | 18045 | 1797K|  299
(1)| 00:00:04 |
--------------------------------------------------

Predicate Information (identified by operation id):
--------------------------------------------------

  1 - filter(“FLAG2”=’N’ AND “FLAG1”=’N’)
```

So, notice that the estimated cardinality is 18,045 rows and the optimizer chose a full scan. We know immediately that this is the wrong plan – but only because we know that the query will return near zero records – we should be using an index! So, assuming you did not know the data yourself, assuming that you did not know that this query should return very few records – how can you get that level of detail, that knowledge? We can see it by using the GATHER_PLAN_STATISTICS hint as follows:

```
ops$tkyte%ORA11GR2> select /*+ gather_plan_statis-
tics */ *
 2  from t where flag1=’N’ and flag2=’N’;
no rows selected
```

```
ops$tkyte%ORA11GR2> select * from table(dbms_xplan.
display_cursor(null,null,’ALLSTATS LAST’));

PLAN_TABLE_OUTPUT
--------------------------------------------------
--------------------
SQL_ID 3qazrn9xhg3zt, child number 0
------------------------------------
select /*+ gather_plan_statistics */ *  from t
where flag1=’N’ and
flag2=’N’

Plan hash value: 1601196873


--------------------------------------------------
----------------------------------------
| Id | Operation        | Name | Starts | E-Rows |
A-Rows |   A-Time   | Buffers | Reads |
--------------------------------------------------
----------------------------------------
|   0 | SELECT STATEMENT |    |    1 |     |    0
|00:00:00.03 |   1073 |   1068 |
|* 1 | TABLE ACCESS FULL| T   |    1 | 18045 |    0
|00:00:00.03 |   1073 |   1068 |
--------------------------------------------------
----------------------------------------

Predicate Information (identified by operation id):
--------------------------------------------------

  1 - filter((“FLAG2”=’N’ AND “FLAG1”=’N’))

19 rows selected.
```

Basically – you would run the query with that hint – and then immediately after running the query you would use DBMS_XPLAN to see the plan used along with the E-Rows (estimated) and A-Rows (actual) columns. Here you can immediately see that we estimated 18,045 rows and got zero! That is the root cause of our problem here. You will be looking for cases where the estimated row count is off by some order of magnitude or more from the actual row count.

Now that we’ve identified the problem – the incorrect cardinality estimate, how do we correct it? Here is where your knowledge of Oracle and its features come into play. In this case, with this data, we have the following solutions available to us:

- We could generate a SQL Profile on the query (10g and above) – this would be useful if this query is executed over and over. The SQL Profile would execute the query and observe the actual row counts and then save that information in the data dictionary so that the next time the query is hard parsed, the correct estimated row counts would be used.
- We could generate extended statistics (11g and above). That is, we can gather statistics not only on FLAG1 and FLAG2 but also on (FLAG1,FLAG2) *together*. That would correct the cardinality estimate on the next hard parse.
- We could use dynamic sampling (9i and above) to help the optimizer during hard parse time. When dynamic sampling is set to level 3 or above – the optimizer will validate any guesses it made by taking a small sample of the table.

We’ll demonstrate the dynamic sampling approach and see the net effect:

```
ops$tkyte%ORA11GR2> select /*+ dynamic_sampling(t
3) */ * from t where flag1=’N’ and flag2=’N’;

Execution Plan
-------------------------------------------------
Plan hash value: 470836197

-------------------------------------------------
| Id | Operation            | Name | Rows | Bytes |
Cost (%CPU)| Time    |
-------------------------------------------------

|  0 | SELECT STATEMENT      |     |   6 |  612 |
2  (0)| 00:00:01 |
|  1 | TABLE ACCESS BY INDEX ROWID| T   |   6 |
612 |   2  (0)| 00:00:01 |
|* 2 |  INDEX RANGE SCAN     | T_IDX |   6 |     |
1  (0)| 00:00:01 |
-------------------------------------------------

Predicate Information (identified by operation id):
-------------------------------------------------

  2 - access(“FLAG1”=’N’ AND “FLAG2”=’N’)
```

As you can see, the optimizer has now guessed six rows – and six being much closer to zero (the actual rows) than 18,045 was - allows the optimizer to come up with the correct plan, an index range scan.

Note: Dynamic Sampling can be set at the database instance level or at the session level using ALTER SESSION in addition to being used as a hint!

So, in short, I look for estimated cardinality values that are orders of magnitude away from the actual cardinalities. If I think I have a bad plan - it is going to be caused by that. So, I look for that and then if I find it - I ask “why is the optimizer having a hard time estimating the correct cardinality”. There are many root causes for incorrect cardinalities – such as using a predicate that applies an unnecessary function to a database column:

```
where trunc(date_col) = to_date( ‘01/01/2011’, ‘dd/
mm/yyyy’ );
```

Here, the optimizer doesn’t know anything about trunc(date_col) – a function applied to a date columns, it knows a lot about date_col - but not f(date_col). So, how could you fix that?

- I could collect statistics on trunc(date_col) in 11g (extended statistics)
- I could use dynamic sampling (9i)
- I could use profiles (10g)
- I could rewrite the query as (this is my preferred approach!):

```
where date_col >= to_date( ‘01/01/2011’, ‘dd/mm/
yyyy’ )
and date_col < to_date( ‘01/01/2011’, ‘dd/mm/yyyy’
)+1
```

So, you would look for the reason why the estimated cardinality is off - which leads to a bad plan - which we need to fix.

Now, what if the estimated cardinalities are dead on and you still think “this query should go faster”. Then you are in the realm of “rewrite the query”. Forget about the existing plan, it is not relevant - you are going for a brand new plan - from a brand new query.

Step 2

In many cases, it would be best to ignore the existing query altogether. So – forget the query and start by trying to write down the “specification” - what does this new query we’ll be generating really need to do. Start fresh, don’t burden yourself with the legacy of the existing query (which might not even be correct!). This part of this step is crucial. If you find that you cannot write this down, that you cannot phrase as a specification what the *question* being asked is – you don’t stand much of a change of tuning that query. Also, if you don’t know what the question being asked is – and no one can tell you – you can tune the query by simply *removing the query*. If no one knows what it does – why are you executing it?

I find many times, the person who wrote a given query that is not performing well cannot explain to me what the query it intended to do. This is a big red flag. You must understand what question the query is answering.

Next, you also need a complete understanding of the schema - all constraints, all rules. For example - this is a query I was asked to tune recently:

```
ops$tkyte%ORA11GR2> SELECT COUNT(*)
 2  FROM T1, T2, T3
 3  WHERE T2.order_id = T1.order_id
 4   AND T2.service_order_id = T3.service_order_id
(+)
 5   AND T3.related_service_order_id = TO_
NUMBER(:v0);
```

Now, just looking at that query, and only the query - the only thing you could “tune” would be to remove the (+) as it is not necessary. Since they have “t2.related_service_order_id = to_number(:v0)”, we KNOW that we don’t need to outer join to T3 - if we didn, t3.related_service_order_id would be NULL sometimes - but null is not equal to anything so - the row produced by an outer join would be rejected anyway.

But, that wouldn’t tune anything - since the optimizer is smart enough to recognize that, even if the developer isn’t. So, it wasn’t hurting anything.

But, the question is - can you reduce that query? It turned out you can and the query I reduced it to was simply:

```
ops$tkyte%ORA11GR2> SELECT COUNT(*)
 2  FROM T3
 3  WHERE T3.related_service_order_id = TO_
NUMBER(:v0);
```

How did I do that? I did that by understanding the schema – knowing the constraints that were in place, how the tables related to each other. When I asked for the schema information – beyond the simple “create table” statements – I got all of the primary keys, foreign keys and not null constraints. Using that information I was able to deduce:

1. Nothing is selected from T1 in the output. There is a primary key on T1(order_id). T2(order_id) happens to be not null and is a foreign key to T1. Therefore in a join of T1 to T2 by that key - all rows in T2 will appear, T1 is not used to reduce or multiply T2 - therefore, T1 is not needed.
2. Then I realized that nothing is selected from T2 in the output - T2(service_order_id) is the primary key - which is joined to T2(service_order_id), which happens to be not null and a foreign key to T2 - so goodbye T2.

But, the only way I could do that reduction was to understand the data model... If I didn't know the constraints - all of them - I could not do that.

So – bottom line – understanding the question and understanding the data model - excessively, extremely, vitally important. As a side note – Oracle Database 11g and above will do that reduction above for us - it will rewrite the query for us – but IF and only IF the constraints are in place!

Lastly - after you have the specification, understand the data model and constraints - what you need is - a mastery of SQL - a really good knowledge of what exists. For example, if you had a query such as:

```
select emp_hist.ename, emp_hist.status_date, emp_
hist.something
 from emp_hist, (select empno, max(status_date)
max_status_date
          from emp_hist
          group by empno) emp_hist2
 where emp_hist.empno = emp_hist2.empno
  and emp_hist.status_date = emp_hist2.max_sta-
tus_date
```

You would have to start by understanding the question – which can loosely be stated as "retrieve from the employee history table the most current record for each employee". Then you would have to understand the schema – in this case – there is a rule in place that says EMPNO, STATUS_DATE is unique – that implies there is exactly one "most current record" for each employee.

Now you are ready to start "tuning" – but in looking at the current query, there isn't much to tune. The database will be able to figure out estimated cardinalities for this easily – so that won't be a problem. The optimizer will come up with an optimal plan for that particular query. What we need is an entirely **new** query.

We could rewrite that query as:

```
select empno,
    max(ename) keep (dense_rank first order by
status_date DESC) ename,
    max(status_date) status_date,
    max(something) keep (dense_rank first order by
status_date DESC) something
 from emp_hist
 group by empno
```

but – only if you knew about "keep dense_rank". We could rewrite the query as:

```
select *
 from (select emp_hist.*, row_number() over (parti-
tion by empno order by status_date DESC) rn
     from emp_hist )
 where rn = 1
```

but – only if you knew about analytics and how to use them.

And so on – there are many ways to rewrite this query – each of which could use many times less resources than the first query did. If I had to guess – I would guess that the KEEP DENSE_RANK query would be the best performing query. It would do a single pass on the EMP_HIST table (the original query makes two passes – once to aggregate and again to join and select from). It would tend to use less temp then the third query since it is aggregating (reducing the result set) whereas the third one would maintain the original size of the EMP_HIST table in some cases before reducing it with the RN=1 predicate.

But – unless you really knew SQL – you wouldn't be able to make these rewrites.

In Conclusion

If there were a set of steps - one by one - that you could follow to tune - we would have turned that into software.

Oh wait, we did :) The automatic SQL tuning features of 10g and above – they do that. The database now recommends some query rewrites, additional statistics to gather, indexes to add, materialized views to consider, partitioning schemes and so on.

For the bits the database cannot self optimize – for those queries we need to look at – you need to know the question being asked, the schema in detail, and have a mastery of what SQL is capable of doing.

Tom Kyte is a Senior Technical Architect in Oracle's Server Technology Division. Tom is the Tom behind the AskTom column in Oracle Magazine, answering people's questions about the Oracle database and its tools (http://asktom.oracle.com/). He is also the author of Expert Oracle Database Architecture (Apress, 2005/2009), Expert One on One Oracle(Wrox Press, 2001/Apress 2004), Beginning Oracle Programing (Wrox press, 2002/Apress 2004), and Effective Oracle by Design (Oracle Press, 2003). These are books about the general use of the database and how to develop successful Oracle applications.

Inheritance in Oracle

Exploring the Descendents

by Saurabh Gupta

The Abstract

The object model of Oracle is an under implemented and commercially unexplored flavor of the database. In the late 1980s, the relational model of Oracle faced several deficiencies, which forced the evolution of the object model. Undoubtedly, the relational model did well for flat data storage, maintenance of integral data, and information security but could not offer a concrete approach to accommodate complex data structures of nested, hierarchical, and extensible types. These subtle restrictions led to the adoption of OOP's concepts in database systems.

The object-oriented model of the database carries the essence of object-oriented concepts like inheritance, data instantiation and abstraction, and establishment of relationships among types. It ensures persistent data management, effective compatibility and integration with OOPs supported languages like C++ and Java, which are the best market flavors of application development.

The article shall be a fair effort to explore the object model of Oracle. The scope of the study shall include the evolutionary background, comparative drive with RDBMS, commercial study and a lot more details. I shall present a sample case study to highlight the structure building implementation and demonstrate the working of common data tasks.

1. Introduction

Even since the days of the evolution of Oracle as the master of commercial database solutions in the industry, it has distinguished itself in terms of modeling implementation, customer satisfaction, scalability and flexibility. The RDBMS concepts, or the relational model, matured with every fit, dominated and suited a maximum sort of data, which can be thought of in an application. Commercially and conceptually, it proved to be a great success unless the lacked support for relationships, extensible and derivative data structures were identified. In a highly performing scientific or manufacturing database model, the data are proclaimed to be less relational and highly extensible. The inability to model out such applications led to the induction of object oriented concepts in Oracle, in parallel to the relational engine. Though some of the OOP's concepts like encapsulation and polymorphism were actively exercised earlier in the Oracle relational model, one of the core OOP's concept of inheritance was still untouched.

Object orientation of data was a new, but different flavor for Oracle users. Here, the objects and their instances, being the model protagonists, are more talked about than tables and columns. Complex data structures which share 'extends from' relationship or have multi-level hierarchical structure were best candidates to fall under this category. Digging over the object model, it could efficiently do semantic data modeling, where reusability and extensibility are its essence. Professionals, in the current phase, are more comfortable with UML and OOP's designs, where relational models are just another data storage layer.

Here, I shall outline the philosophy of OODBMS (Object Oriented Database Management Systems) and show the evolution of ORDBMS (Object Relational Database Management Systems), the hybrid model of the object and relational model of the Oracle database. The scope of the paper covers the object-oriented concepts, the kick off implementation and prepares the reader to explore the next level.

2. Object Oriented Model: The Philosophy

On a broader sense, the object oriented model follows the principle '*Every entity as an object has the tendency to extend from and extend to another entity*'. Unlike the relational model which considers entity as an abstract unit to build a table, the object model percepts 'entity' as a physical object. In my personal observation, the definition of an entity in the object model looks closer to real terms, as it clearly defines its structure and behavior. Multiple instances of the same object type can reside in different database tables. In a single request, one can fetch the complete instance, which has a full set of object attributes. It is analogous to the 'Classes' in C++ or Java.

3. Object: High Level Design

An object is the basic unit in the Object Oriented Model (OOM) of Oracle. It is the base of the model, which can realize an entity. Technically, an object type has a set of attributes, which builds up its skeleton. It can be instantiated in multiple contexts to give a meaningful dimension of its behavior. Instantiation of an object can be understood as the role of a person at different places. A 'person' is a 'student' at 'college', 'employee' at 'workplace', 'customer' at a 'shop' and many others. As an entity, his behavior in a context is a determinant of his behavioral values, structure remaining the same.

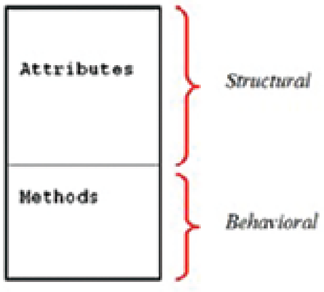


Figure 1. Skeleton of an Object

Getting into the high-level design of an object, its structural and behavioral aspects are essential and considerable over its design.

3.1 Structural Aspects

Structural aspects include the identification of the relationship amongst the objects and their properties. In an object, its attributes demonstrate the structural aspect of the object.

The relationship can be classified into inheritance, association and aggregation.

Two objects share an 'inheritance' or 'extends from' relationship, when one (subtype) of them extends from the other (super type). In such case, the derived object inherits the structural and behavioral features of its parent. Later, we shall discuss the 'inheritance' relationship as the sole and soul feature of object oriented model.

Association relationship is when an object is associated to another as 'one to one', 'one to many' or 'many to one'. This relationship can be best observed between objects and the application. Mere structure and property would not be sufficient to demonstrate such a relationship.

Aggregation relationships demonstrate the 'part of' relation among the objects. An object can accommodate several other objects to become master objects, hence becoming a composite object. Note that it is different from a Parent-Child relationship as the member object is just a component of the master part. A single query to the composite object would be sufficient to fetch a bill of material of all the component objects. It can fall among the following categories.

- Existence dependent – subcomponent objects are dependent on the master object
- Nonexistence dependent – subcomponent objects are an independent entity, but accommodated by the master object
- Existence composition – subcomponent is an exclusive constituent object of the master object
- Nonexistence composition – subcomponent object is a component of the master object, which can be shared by other master objects too
- Homogeneous composition – subcomponents objects are of same type as that of the master object.
- Multilevel (complex) composition – subcomponents objects involve many branching and level hierarchies

3.2 Behavioral Aspect

The behavior of an object can be knobbed in a situation by its member components. The skeleton of an object encapsulates attributes and methods. The type 'specification' carries only the method prototypes, while the type 'body' contains the definition of the methods. This ensures data hiding and prevents method algorithms to be public. The attributes build up the property structure of the object, while methods are the operational routines on object attributes. Implications are clear that attributes are part of the data layer while methods are part of the data access/interface layer. This is one of the major advantages of the object over database tables in that an object not only holds the data, but optionally can also have operations upon the data.

The methods can be generic or user defined. Generic methods are meant for accessing the attributes and basic operations like insert, update, delete, and query. User defined methods embed a defined business logic, which gets activated on the object instance.

4. Object Relational Database Management System: The Evolution

Over the years of trials and results, the relational model (RDBMS) has been an established and trusted data model, while the object-oriented database still needs the drive to catch up with the competition. Both have their own advantages and disadvantages. On one side, the relational model stands mum before inheritance, while on the other side, the object model lacks a universal data model and carries 'tight coupling'. The object relational model is an extended version of the relational model which includes object orientation behavior. It offers an adhesive platform for the advanced database systems, which not only widens the scope of data fit by multiplying the features of both the models, but also adds to the trust of the customer to handle the real world application scenarios.

5. ORDBMS vs. RDBMS: A Comparative Study

By now, I believe we are familiar enough with objects to shoot a comparative drive between object relational and relational models of the Oracle database.

5.1 Modeling Becomes Easier

The biggest plus with OODBMS is reusability, sharing, and the support of inheritance. It offers semantic modeling of real world problems, while RDBMS fails to model structured and extensible data. Heterogeneous data of a recursive nature demands inheritance implementation, which can be easily done with ORDBMS.

By real world problems, I mean the realization of an entity, hence the data, in an accessible form. Since ORDBMS considers an entity as an object, it requires less referential integrity, mapping and complex normalization paradigms.

For example, consider a 'company', which employs people at different levels like 'director', 'manager', 'developer', 'admin', and 'HR'. All have a different set of structural and behavioral properties at each level. There are two related entities identified as 'company' and 'employee'.

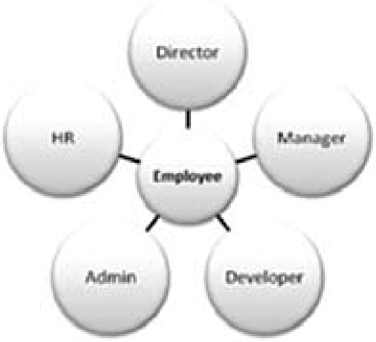


Figure 2. EMPLOYEE entity realization in Object Model

An object instance of 'employee' type is capable of holding all its descendents like director, manager, developer, admin and HR. But similar case implementation in the relational model would end up in the creation of separate tables for each level and a complex referential network. So clearly the implication from the case is that modeling of the application scenario becomes easy when seen through the object orientation frame.

5.2 As a Performance Booster

Object models have been observed to give better performance than relational ones. A properly optimized object model would yield

1 to 5 times better performance while querying and inserting object instances.

Since the object model involves a high level of coupling factors, there are several other factors which determine the performance. They are transaction overhead, network configuration, object level locking, instance caching, and query optimization techniques.

5.3 Miscellaneous

Apart from the above listed benefits, the object oriented database system persistently stores the objects in the database. It is easy to integrate with OOP's supportive languages like C++ and Java. Client developers can hold and access the object instances without breaking them into a relational format. This secures the data access methodology from the client.

6. Object Oriented Model: The Implementation

Until now, we have discussed the high-level design of an object. Now, we shall move on to the implementation part and learn how the objects are stored in the database, tables and their references.

We shall move ahead with the object-oriented glossary with the help of the case study, which has been earlier discussed. The 'employee' model as shown below,

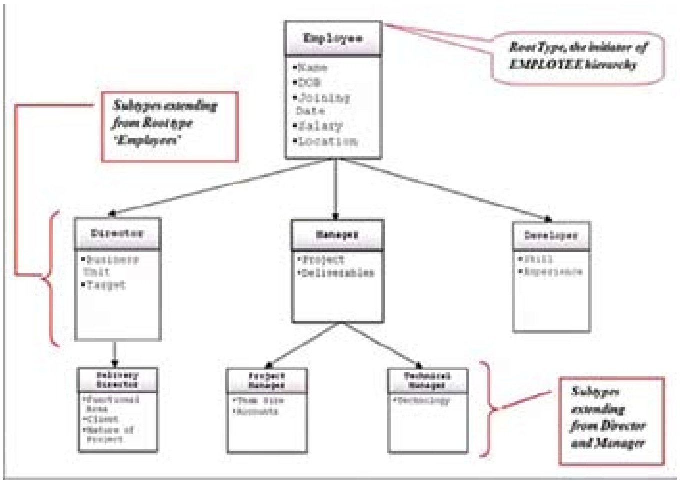


Figure 3: Data model for the Employee Hierarchy in the Company

The model represents the type evolution and hierarchy model of an organization. An employee can be a director, a manager or a developer. There are different sets of responsibilities defined at each level, which we shall try to model in the object-oriented methodology.

6.1. Object Type and Instance: The object type is a user defined object type and physically resides in the database as a schema object. Similar to classes, it can encapsulate attributes (mandatory) and methods (optional) to demonstrate its structural and behavioral aspect respectively. Object attributes can be of a primitive data type (number, varchar2, date) or any valid user defined type. Like the Oracle supplied types, an object can be used to state the data type of a column in a table, or an attribute in another object, or as a parameter. The object instance is a physical realization of the object. A variable of the object type behaves as its instance, which can store attribute values and invoke its methods.

For an object, extension and instantiation are two properties which can be defined at the type declaration level. 'Extension allowed' object types can extend themselves to a new object. 'Instantiation allowed' allows the creation of the object's instance in the database.

From our case study 'employee' model, EMPLOYEE object can be created as below.

```
-------Creating EMPLOYEE object type specification

CREATE OR REPLACE TYPE EMPLOYEE_OT AS OBJECT
(
EmpID NUMBER,
EmpName VARCHAR2(1000),
EmpBirthDate DATE,
EmpJoinDate Date,
EmpSalary NUMBER,
EmpLoc VARCHAR2(100),
MEMBER FUNCTION GetEmpExp RETURN NUMBER
)
NOT FINAL ---'employee' object tends to extend
NOT INSTANTIABLE; ---'employee' object instance
cannot be created
```

The attributes build up the structure of the object, while the only member method 'GetEmpExp' demonstrates one of the traits of the object. The NOT FINAL clause creates an extendible object type. The NOT INSTANTIABLE clause restricts the creation of any instance of the object in the database.

6.2. Object Methods: Object methods can be member, static or constructor. Member methods are majorly operative on the object attributes. Static methods embed a generic logic, which does not involve the object attributes. Constructors are implicit methods created by the database for initialization and default instantiation purpose. It can be overridden with a user-defined constructor.

The type definition can be completed by defining its TYPE BODY as below.

```
-------Creating EMPLOYEE object type body

CREATE OR REPLACE TYPE BODY EMPLOYEE_OT IS
 MEMBER FUNCTION GetEmpExp RETURN NUMBER
 IS
 BEGIN
 RETURN SYSDATE-EmpJoinDate;
 END GetEmpExp;
END;
/

-------Invoking the member method
SELECT T.GetEmpExp () FROM EMPLOYEES T
/
```

6.3. Object Type Inheritance: This is one of the most distinguished features of the object model. It enables the creation of object type hierarchies. Type hierarchies can initiate from a generic type and extend incrementally to inherit the properties of its originator and accommodate specific properties at each derivative node. A node in this type hierarchy can behave either as a parent node or a child node or both. In technical terms, parent and child nodes are known as super-type and sub-type respectively.

The type's structural changes can be achieved using the ALTER TYPE statement and can be cascaded across all the instances contained by all the tables in the database. If any dependency check fails in the type hierarchy, the statement raises the appropriate exception and rolls back the operation.

In the 'employee' model, the 'employee_ot' object type can be extended to create objects for director, manager and developer as shown below.

Note that a type, which has the ability to extend, must be created with the NOT FINAL option.

```
-------Creating DIRECTOR object sub type under
EMPLOYEE
CREATE OR REPLACE TYPE DIRECTOR_OT UNDER EMPLOYEE_
OT
(
DirBusinessUnit VARCHAR2(100),
DirTarget NUMBER)
NOT FINAL; ---director tends to extend as delivery
director
```

Explicitly also, a type can be declared NOT FINAL to extend from there onwards using the ALTER TYPE statement.

```
ALTER TYPE DIRECTOR_OT NOT FINAL;
```

```
-------Creating DELIVERY DIRECTOR Object sub-type
under DIRECTOR
CREATE OR REPLACE TYPE DEL_DIRECTOR_OT UNDER
DIRECTOR_OT
(
DelDirFuncArea VARCHAR2(100),
DelDirClient VARCHAR2(1000),
DelDirProjNature VARCHAR2(10)
);
```

```
-------Creating MANAGER Object sub-type under
EMPLOYEE
CREATE OR REPLACE TYPE MANAGER_OT UNDER EMPLOYEE_OT
(
ManProj VARCHAR2(1000),
ManDelv VARCHAR2(100)
)
NOT FINAL; ---manager tends to extend as project
manager and tech manager
```

```
-------Creating PROJECT MANAGER Object sub-type
under MANAGER
CREATE OR REPLACE TYPE PROJMAN_OT UNDER MANAGER_OT
(
PManTeamSize NUMBER,
PManAccounts NUMBER);
```

```
-------Creating TECHNICAL MANAGER Object sub-type
under MANAGER
CREATE OR REPLACE TYPE TECHMAN_OT UNDER MANAGER_OT
(
TManTech VARCHAR2(100));
```

```
-------Creating DEVELOPER Object sub-type under
EMPLOYEE
CREATE OR REPLACE TYPE DEVELOPER_OT UNDER EMPLOYEE_
OT
(
DevSkill VARCHAR2(100),
DevExp NUMBER
);
```

6.4. Object Tables: Database tables can accommodate objects in two ways; either as object tables or relational tables. An object table is created on top of a valid schema object. Every row in the object tables holds the instance of that object, hence known as row object. It can be accessed either as a single object instance table or as a relational table with the dependent object attributes listing as columns of the table. As a single instance table, it can demonstrate many distinct object oriented features.

In a conventional relational heap table, a column of an object type can exist to hold the object's instances, hence known as a column object.

For the object types created above, I would create a table EMPLOYEES to hold instances of all employees, falling within the organization hierarchy model.

```
-------Creating object table EMPLOYEES with
EMPLOYEE_OT object as row object
CREATE TABLE EMPLOYEES OF EMPLOYEE_OT
(EmpID PRIMARY KEY,
 Check (EmpSalary>0));
```

The object type upon which the table is created is known as declared type of the table.

Object level scalar attributes can be indexed or constrained in the object tables. All the attributes (inherited and self) which are available until the declared object level node, can be constrained or indexed. Database triggers can be created as usual on the object tables or columns.

The EMPLOYEES table declares a primary key 'EmpID' and imposes a check constraint on the attribute 'EmpSalary'. It must be more than zero in all the instances of the 'employee' object and its descendents.

Now, I shall show the creation of instances within the EMPLOYEES table.

The insert statement in the screenshot creates an instance of the EMPLOYEE_OT object type, but it fails.

```
SQL> INSERT INTO EMPLOYEES VALUES
(EMPLOYEE_OT(100,'Emp 1',TO_DATE('01-JUN-1975'),
TO_DATE('26-JUL-2001'), 4500, 'IND'));
(EMPLOYEE_OT(100,'Emp 1',TO_DATE('01-JUN-1975'),
TO_DATE('26-JUL-2001'), 4500, 'IND')
 *
ERROR at line 2:
ORA-22826: cannot construct an instance of a non
instantiable type
```

The reason is that EMPLOYEE_OT is a NON INSTANTIABLE type, due to which its instances cannot be created in the table.

```
-------Populating the test data into EMPLOYEES
table
BEGIN
--------Creating instance of DIRECTOR
INSERT INTO EMPLOYEES VALUES
```

```
(DIRECTOR_OT(100,’Emp 1’,TO_DATE(‘01-JUN-1968’),
TO_DATE(‘26-JUL-2001’), 4500, ‘IND’, ‘BANKING’,
10000));

--------Creating instance of DELIVERY DIRECTOR
INSERT INTO EMPLOYEES VALUES
(DEL_DIRECTOR_OT(200,’Emp 2’,TO_DATE(‘11-
JAN-1971’), TO_DATE(‘18-DEC-2010’), 5720, ‘AUS’,
‘BANKING’, 10000, ‘APAC’, ‘XYZ Bank’,’Dev’));

--------Creating instance of MANAGER
INSERT INTO EMPLOYEES VALUES
(MANAGER_OT(300,’Emp 3’,TO_DATE(‘13-FEB-1981’),
TO_DATE(‘27-SEP-2003’), 8302, ‘NZ’,’ABC Pvt
Ltd’,’Banking App’));

--------Creating instance of PROJECT MANAGER
INSERT INTO EMPLOYEES VALUES
(PROJMAN_OT (400,’Emp 4’,TO_DATE(‘21-MAR-1982’),
TO_DATE(‘16-APR-2006’), 9320, ‘UK’, ‘ABC Pvt
Ltd’,’App Dev’, 25, 10));

--------Creating instance of TECHNICAL MANAGER
INSERT INTO EMPLOYEES VALUES
(TECHMAN_OT (500,’Emp 5’,TO_DATE(‘16-AUG-1979’),
TO_DATE(‘03-JAN-1999’), 3728, ‘SL’, ‘ABC Pvt
Ltd’,’App Support’, ‘Database’));

--------Creating instance of DEVELOPER
INSERT INTO EMPLOYEES VALUES
(DEVELOPER_OT(600,’Emp 6’,TO_DATE(‘17-DEC-1973’),
TO_DATE(‘19-APR-1998’), 4830, ‘RUS’, ‘JAVA’,6));

END;
```

The above block execution inserts six instances of the declared object’s descendent types into the EMPLOYEES table.

6.5. Object References: Referential relationships can be established between a column and an object type. The reference column serves as a pointer to any instance of the object type, which exists as a row or column object in the database. Their scope can be restricted too, such that the column would point to the object instance created within the specified table only.

For example, the organization keeps track of the salary modifications of its employees, along with their employment details. The employee details could have a referenced EMPLOYEES table for the details. But REF object is a better solution to get rid of the referential network.

The column ‘EmpDetails’ stores a reference pointer, which always points to the instances of EMPLOYEE_OT across the database. The CREATE TABLE statement below creates the table.

```
-----Create the table with REF column
CREATE TABLE SAL_UPDATE
(EmpID NUMBER,
 EmpDetails REF EMPLOYEE_OT,
 EmpModSal NUMBER,
 EmpModSalDate DATE)
```

```
-----Inserting test data from EMPLOYEE’s table
INSERT INTO SAL_UPDATE
SELECT EmpId, REF (T)
FROM EMPLOYEES T;

-----Query the table to check the REF Column value
SQL> SELECT EMPID, EMPDETRILS FROM SAL_UPDATE;

EMPID EMPDETRILS
----- ----------
 100 0000220208
 200 0000220208
```

6.6. Object Type Metadata: The object type metadata are stored in the data dictionary views [ALL | DBA | USER]_TYPES and [ALL | DBA | USER]_TYPE_ATTRS.

[ALL | DBA | USER]_TYPES – Stores the object type information. I shall briefly describe some of the key columns of the view.

TYPE_OID is the unique 32-bit hex code generated for each object type created in the database.

TYPECODE differentiates a type as an object or collection.

ATTRIBUTES and METHODS are the count of attributes for the type.

FINAL and INSTANTIABLE determine if the type can be further extended or instantiated.

SUPERTYPE_OWNER and SUPERTYPE_NAME are the type’s parent type details. They are NULL for non-hierarchical object types.

TYPEID column uniquely identifies an object type node in a type hierarchy structure. It is NULL for non-hierarchical object types.

[ALL | DBA | USER]_TYPE_ATTRS – Stores the information of net attributes available to an object type. The INHERITED column of the view distinguishes an attribute as inherited or owned. Other than INHERITED, all other columns of the view are attribute general info columns.

```
SQL> desc dba_types
Name                             Null?          Type
----------------------- ---------   ----------
OWNER
VARCHAR2(30)
TYPE_NAME
VARCHAR2(30)
TYPE_OID                                        RAW(16)
TYPECODE
VARCHAR2(30)
ATTRIBUTES                                      NUMBER
METHODS                                         NUMBER
PREDEFINED
VARCHAR2(3)
INCOMPLETE                                  VARCHAR2(3)
FINAL                                       VARCHAR2(3)
INSTANTIABLE                                VARCHAR2(3)
SUPERTYPE_OWNER
VARCHAR2(30)
SUPERTYPE_NAME
VARCHAR2(3O)
LOCAL_ATTRIBUTES                                NUMBER
LOCAL_METHODS                               NUMBER
TYPEID                                      RAW(16)
```

```
SQL> desc dba_type_attrs
Name                          Null?          Type
----------------------- ---------   ----------
OWNER                                       VARCHAR2(30)
TYPE_NAME
VARCHAR2(30)
ATTR_NAME
VARCHAR2(30)
ATTR_TYPE_MOD                               VARCHAR2(7)
ATTR_TYPE_OWNER
VARCHAR2(30)
ATTR_TYPE_NAME
VARCHAR2(30)
LENGTH                                      NUMBER
PRECISION                                          NUMBER
SCALE                                       NUMBER
CHARACTER_SET_NAME                          VARCHAR2(44)
ATTR_NO NUMBER INHERITED                    VARCHAR2(3)
```

7. Useful Functions In The Object Relational Model

VALUE, TREAT, REF, DEREF, IF OF, and SYS_TYPEID are a few amongst the many functions which are frequently used with object type instances.

7.1. The VALUE function is used to query object instances in a table. These instances can be of the declared type or even its sub-types.

For example, the SELECT query on the EMPLOYEES table is shown in the below screen dump

```
SQL> SELECT VALUE(T) FROM EMPLOYEES T;

VALUE(T)(EMPID, EMPNAME, EMPBIRTHDATE, EMPJOINDATE,
EMPSALARY, EMPLOC)
---------------------------------------------------
DIRECTOR_OT(100, ‘Emp 1’, ‘01-JUN-68’, ‘26-JUL-01’,
4500, ‘IND’, ‘BANKING’, 10000)
DEL_DIRECTOR_OT(200, ‘Emp 2’, ‘11-JAN-71’, ‘18-DEC-
10’, 5720, ‘AUS’, ‘BANKING’, 10000, ‘APAC’, ‘XYZ
Bank’, ‘Bev’)
MANAGER_OT(300, ‘Emp 3’, ‘13-FEB-81’, ‘27-SEP-03’,
8302, ‘NZ’, ‘ABC Put Ltd’, ‘Banking App’)
PROJMAN_OT(400, ‘Emp 4’, ‘21-MAR-82’, ‘16-APR-06’,
9320, ‘UK’. ‘ABC Put Ltd’, ‘App Devi, 25, 10)
TECHMAN_OT(500, ‘Emp 5’, ‘16-AUG-79’, ‘03-JAN-
99’, 3728, ‘SL’, ‘ABC Put Ltd’, ‘App Support’,
‘Database’)
DEVELOPER_OT(600, ‘Emp 6’, ‘17-DEC-73’, ‘19-APR-
98’, 4830, ‘RUS’, “JAUA’, 6)

6 rows selected.
```

7.2. The TREAT function is used to access a sub-type attribute of the object table. Normally, the attributes of the declared type are directly accessible from a SELECT statement, but not the sub-type attributes. It typecasts all the instances as one sub-type and tries to get the value of the requested attribute for all the instances. For indifferent instances, the value appears NULL.

Check the example below.

```
------Declared type attributes are directly acces-
sible from the SELECT statement
SELECT EmpName FROM EMPLOYEES;

EMPNAME
----------------------------------
Emp 1
Emp 2
Emp 3
Emp 4
Emp 5
Emp 6

6 rows selected.

------Sub-type attributes are not accessible from
the SELECT statement
SQL> SELECT DEVSKILL FROM EMPLOYEES;
SELECT DEVSKILL FROM EMPLOYEES
 *
ERROR at line 1:
ORA-00904: “DEVSKILL”: invalid identifier

------Used TREAT function to typecast all instances
as DEVELOPER_OT and fetch the value of DEVSKILL for
all instances
select treat(value(t) as DEVELOPER_OT).DevSkill
from employees t;

TREAT(VALUE(T)ASDEVELOPER_OT).DEVSKILL
---------------------------------------------------


JAVA

6 rows selected.
```

7.3. The REF function is used to get the reference pointer value of an object instance. Refer to the example below

```
select ref(t) from employees t;

REF(T)
---------------------------------------
000028020910EE4104966945
00002802098308FA3FBED543
0000280209ED15636142B24F
0000280209FC59EB9982644
0000280209F4159E8B61514
0000280209BD6B730691C

6 rows selected.
```

It is used to populate a REF object column in a table. Similarly, the DEREF function works just at the opposite track. From a REF column value, it can get the actual instance values. Check this DEREF example in the below screenshot.

```
SQL> SELECT deref(t.empdetails) FROM SAL_UPDATE T;
DEREF(T.EMPDETAILS)(EMPID, EMPNAME, EMPBIRTHDATE,
EMPJOINDATE, EMPSALARY, EMPLOC)
```

```
--------------------------------------------------
DIRECTOR_OT(100, ‘Emp 1’, ‘01-JUN-68’, ‘26-JUL-01’,
4500, ‘IND’, ‘BANKING’, 10000)
DEL_DIRECTOR_OT(200, ‘Emp 2’, ‘11-JAN-71’, ‘18-DEC-
10’, 5720, ‘AUS’, ‘BANKING’, 10000, ‘APAC’, ‘XYZ
Bank’, ‘Bev’)
MANAGER_OT(300, ‘Emp 3’, ‘13-FEB-81’, ‘27-SEP-03’,
8302, ‘NZ’, ‘ABC Put Ltd’, ‘Banking App’)
PROJMAN_OT(400, ‘Emp 4’, ‘21-MAR-82’, ‘16-APR-06’,
9320, ‘UM’, ‘ABC Put Ltd’, ‘App Devi, 25, 10)
TECHMAN_OT(500, ‘Emp 5’, ‘16-AUG-79’, ‘03-JAN-
99’, 3728, ‘SL’, ‘ABC Put Ltd’, ‘App Support’,
‘Database’)
DEVELOPER_OT(600, ‘Emp 6’, ‘17-DEC-73’, ‘19-APR-
98’, 4830, ‘RUS’, “JAUA’, 6)

6 rows selected.
```

7.4. The IS OF function is used to filter the instances of a specific branch only. This means that it will get the instances of the specified type and its descendents. Check the example below.

```
SQL> SELECT VALUE(T) FROM EMPLOYEES T WHERE
VALUE(T) IS OF (MANAGER_OT);

VALUE(T)(EMPID, EMPNAME, EMPBIRTHDATE, EMPJOINDATE,
EMPSALARY, EMPLOC)
--------------------------------------------------
MANAGER_OT(300, ‘Emp 3’, ‘13-FEB-81’, ‘27-SEP-03’,
8302, ‘NZ’, ‘ABC Put Ltd’, ‘Banking App’)
PROJMAN_OT(400, ‘Emp 4’, ‘21-MAR-82”, ‘16-APR-06”,
9320, ‘UK’, ‘ABC Put Ltd’, ‘App Devi, 25, 10)
ECHMAN_OT(500, ‘Emp 5’, ‘16-AUG-79”, ‘03-JAN-
99’, 3728, ‘SL’, ‘ABC Put Ltd’, ‘App Support’,
‘Database’)
```

In the above screen shot, the SELECT query requests to get the instances of MANAGER. But the query output lists the instances of manager, project manager and technical manager.

Another enhancement to the IS OF function is ‘IS OF ONLY’. It would perform hard filtering by fetching only the instances of the specified type.

```
---------The query fetches the instance of MANAGER_
OT only
SELECT VALUE(T) FROM EMPLOYEES T WHERE VALUE(T) IS
OF (ONLY MANAGER_OT);

VALUE(T)(EMPID, EMPNAME, EMPBIRTHDATE, EMPJOINDATE,
EMPSALARY, EMPLOC)
--------------------------------------------------
MANAGER_OT(300, ‘Emp 3’, ‘13-FEB-81’, ‘27-SEP-03’,
8302, ‘NZ’, ‘ABC Pvt Ltd’, ‘Banking App’)

---------The query fetches the instance of PROJMAN_
OT only
SELECT VALUE(T) FROM EMPLOYEES T WHERE VALUE(T) IS
OF (ONLY PROJMAN_OT);

VALUE(T)(EMPID, EMPNAME, EMPBIRTHDATE, EMPJOINDATE,
EMPSALARY, EMPLOC)
--------------------------------------------------
PROJMAN_OT(400, ‘Emp 4’, ‘21-MAR-82’, ‘16-APR-06’,
9320, ‘UK’, ‘ABC Pvt Ltd’, ‘App Dev’, 25, 10)
```

7.5. The SYS_TYPEID function returns the TYPEID value of the object instance. It is useful in identifying the object level node of an instance. Its value is the same as contained in TYPEID column of the USER_TYPES dictionary view.

```
SELECT EmpID, SYS_TYPEID(VALUE(T)) FROM EMPLOYEES
T;

 EMPID SYS_TYPEID(VALUE(T))
---------- --------------------------------
 100 02
 200 03
 300 04
 400 05
 500 06
 600 07

6 rows selected.
```

For cross verification, we shall query the TYPEID column in the USER_TYPES dictionary view for the above types.

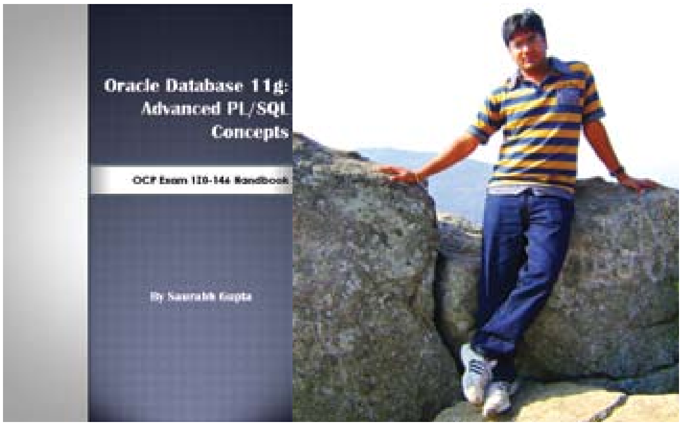
```
SELECT TYPE_NAME, SUPERTYPE_NAME, TYPEID
FROM USER_TYPES
WHERE TYPE_NAME IN (‘EMPLOYEE_OT’,
                       ‘DIRECTOR_OT’,
                       ‘DEL_DIRECTOR_OT’,
                       ‘MANAGER_OT’,
                       ‘PROJMAN_OT’,
                       ‘TECHMAN_OT’,
                       ‘DEVELOPER_OT’)
ORDER BY TYPEID;

TYPE_NAME SUPERTYPE_NAME TYPEID
------------------------------ --------------------
---------- --------------
EMPLOYEE_OT 01
DIRECTOR_OT EMPLOYEE_OT 02
DEL_DIRECTOR_OT DIRECTOR_OT 03
MANAGER_OT EMPLOYEE_OT 04
PROJMAN_OT MANAGER_OT 05
TECHMAN_OT MANAGER_OT 06
DEVELOPER_OT EMPLOYEE_OT 07

7 rows selected.
```

Conclusion: A Step Ahead

This paper saw the exploration, observation and implementation of a business problem with the object model. With its unquestionable model, the object relational platform is capturing the latest flavor of database designing. Relational structures have already proven their capabilities since 1979, but over the years, the database technologies have matured enough to induct a non-database paradigm, object oriented, to meet the competent business needs. The upcoming years would be the testing time for the OODBMS and ORDBMS. Reliability, robustness and ability to pace with day-to-day problems are the few factors which would be considerable over its commercialization. Additionally, it would be a new learning track of PL/SQL programming for database professionals.

I hope my sincere effort to familiarize you with the 'world of objects' would be noted. I shall welcome your comments and feedback at sbh.orcl@gmail.com.

I am Saurabh Gupta, PL/SQL developer for more than four years. I am an Oracle 11g Advanced PL/SQL Professional and specialized on PL/SQL programming, Database modeling and designing, and PL/SQL code optimization. I share my experiences and learning at OTN, through online publications and at my blog sbhoracle.wordpress.com. I have authored more than 60 online publications on Oracle PL/SQL and 11g features at PSOUG (www.psoug.org), DBANOTES (www.dbanotes.com), EXFORSYS (www.exforsys.com), and CLUB-ORACLE (www.club-oracle.com).
Besides digging into Oracle, I love to play snooker but at times, I find drawing out my thoughts on paper catalyzes refreshment. One can reach me @ sbh.orcl@gmail.com for any query or clarification.



MySQL for Oracle DBAs

by Kellyn Pot'Vin

I'd been working with Oracle and SQL Server for approximately seven years when a manager first handed me a MySQL environment and informed me I would be managing it. As a multi-platform DBA that specialized in these two platforms, it was surprisingly easy to acclimate to MySQL and as MySQL becomes more popular and continues to be a highly supported product by Oracle, it is becoming almost essential for Oracle DBA's to know this database platform.

Understanding MySQL, if you are an Oracle DBA that has been exposed to Microsoft SQL Server, this will assist you in adapting a bit quicker. When you have been exposed to SQL Server, (or even Sybase) you quickly recognize the Information schema and similar separation design of instances in MySQL.

If you have not been exposed to database platforms outside of Oracle, let's start with common terminology differences and progress from there.

Oracle	**MySQL**
Database Files	Database Server Instance
Database Instance	Database Server Instance
Schema	Database

STORAGE ENGINES

One of the crucial differences in MySQL from other types of databases is the storage engine. A storage engine is a handler for different table types in MySQL. The main difference in the listed types are those that handle transaction-safe and those that handle non-transaction-safe tables.

As of MySQL 5.1, MySQL Server uses a pluggable storage engine architecture that enables storage engines to be loaded into and unloaded from a running MySQL server. As of MySQL 5.5 the default storage engine is InnoDB, a transaction-safe (ACID compliant) storage engine for MySQL that has commit, rollback, and crash-recovery capabilities to protect user data. InnoDB uses row-level locking and Oracle-style consistent non-locking reads to increase multi-user concurrency and performance. InnoDB also supports FOREIGN KEY referential-integrity constraints.

You can use the "SHOW ENGINES" command to see what storage engines your server OS supports. Ensuring that the type you wish to utilize states that it "YES" (is supported) is essential.

Of the ones shown, the most common storage engines are the MyISAM. This particular storage engine is used the most in Web, data warehousing, and other application environments. To name a few other storage engines (not an exhaustive list) are :

- IBMB2I
- MERGE
- MEMORY
- EXAMPLE
- FEDERATED
- ARCHIVE
- CSV
- BLACKHOLE

Differences to remember in MySQL when coming from an Oracle world - most of these settings are configured in the my.cnf configuration file or can be set at the session level:

- AUTOCOMMIT=0|1 disables or enables the default autocommit mode for the current session.
- Storage engines are like setting the block size. A cool feature of MySQL is the ability to set the default engine as well as being able to set the database engine at the table level AND even at the partition level giving you extreme flexibility in performance and design.
- Yes, MySQL has a cost-based optimizer!
- Yes, MySQL does have an InnoDB undo space that is similar to an Undo tablespace.
- Areas of memory, commonly referred to as different types of cache, (i.e. buffer cache, library cache, etc…) are also the same term that MySQL uses, (User caches, Query Cache, Dictionary Cache, InnoDB and MyISAM Cache.)
- Sequences are set up differently in MySQL, read up on them before assuming!
- DUAL table doesn't exist, but you can select from it, (all an illusion!)
- The default character set and collation dictate the LIKE case insensitive, requiring less syntax. This can be server wide and even table specific.

MySQL Installation guides are quite complete, so I'll refer to the experts. http://dev.mysql.com/doc/refman/5.5/en/installing.html

Installation, set by the installer, can reside in many locations, but commonly, where Oracle would place installation in the $ORACLE_HOME and the $ORACLE_BASE, (i.e. /u01/oracle/app/product/..) MySQL prefers the following directory locations, /usr/lib, /var/lib/, /var/log and /etc/init.d/ for its binaries, but all paths are completely customizable before or after installation.

Where Oracle has a pfile, (init<sid>.ora) to tell the database about configuration basics, MySQL utilizes a configuration file called the my.cnf, often located in /etc or /etc/mysql.

This file contains information about memory, storage engine, directories and essential information about the database used when the database is started.

```
EXAMPLE OF A SHORT my.cnf:
# The MySQL server
[mysqld]
```

```
port = 3306
socket = /var/lib/mysql/<file_name>.sock □The sock-
et file location
pid-file = /var/lib/mysql/<file_name>.pid
read_only = OFF

key_buffer = 1024M
max_allowed_packet = 10M
table_open_cache = 512

sort_buffer_size = 2M
read_buffer_size = 1M
read_rnd_buffer_size = 4M
myisam_sort_buffer_size = 64M
thread_cache_size = 32
query_cache_size = 32M
query_cache_type = ON

basedir = /usr
datadir = /local/mysql/<inst_nm>/data
tmpdir = /local/mysql/<inst_nm>/temp/
general_log_file = /local/mysql/<inst_nm>/
logs/<file_name>.log
log-error = /local/mysql/<inst_nm>/logs/<file_
name>.err
slow_query_log_file = /local/mysql/<inst_nm>/
logs/<file_name>-slow.log
long_query_time = 0.5
expire_logs_days = 1
```

Depending on the type of storage engine utilized, a section is set up for its parameter settings, along with performance settings and other key information for replication and other advanced features. There are over 350 different parameters that can be adjusted.

Local connections can connect through a socket file for enhanced performance and security, which is also configured in the /etc/my.cnf file. You will note that in the above example, the socket file location is */var/lib/mysql/<file_name>.sock.*

Where the port default is 1521 for Oracle databases, MySQL is 3306. This can be managed and changed to suit the needs of the IT organization. Secondary instances residing on the same host will require and be configured with a second port or different socket file.

Logs

MySQL generates several different logs to be used for different needs.

The standard log is very similar to Oracle's alert log, generating basic database information, warnings and errors that the DBA will need to be aware of.

There is the error log that is populated only with errors and is the log most MySQL DBA's monitor by default.

There is a binary log in the bin directory, which is typically used for transactional recovery or replication.

The Slow Query Log contains all performance impacting data and is of special interest to any DBA who is in search of performance data.

Each of the logs above is configured in the my.cnf file or via the command line start server executable.

With MySQL, one of the convenience features is the ability to query the log from the SQL prompt:

```
SET GLOBAL GENERAL_LOG=1;
SELECT * FROM mysql.general_log;

Connecting to MySQL:
mysql -u root □No, not the best choice…
mysql -u <username> -p □This will return a prompt
for a password

Running a file from the Unix prompt to MySQL:
/usr/bin/mysql -u<user_name> -p<password> -D<db_
name> -h<host_name> < <sqlscript you want to exe-
cute>
```

Once Connected
INFORMATION SCHEMA

For those of you that have supported Microsoft's SQLServer in the past, the Information schema isn't anything new. For those of you who have been deprived of this pleasure, here is the low down:

The Information Schema for MySQL is dynamically built and established at server start and is continuously updated during server uptime. This database contains such items as:

- Partition definitions
- The processlist
- Tables, columns, and other data dictionary information
- Various server configuration variables

Once you are logged in, the easiest commands to remember are the "SHOW" commands:

- SHOW TABLES;
- SHOW DATABASES;
- SHOW VARIABLES;
- SHOW WARNINGS;
- SHOW STATUS;

Most DDL is very similar to Oracle. But remember, this is not PL/SQL and MySQL will prompt you for incorrect syntax if you attempt to do most PL/SQL commands…

Backups

There are a several ways to back up data. Because of MySQL's plugin architecture, you can more than likely find some backup utility that fits your needs. Here are just some of the options:

- Mysqldump
- InnoDB hot backup
- Storage engine driven
- Third party backup utilities

Assuming you are using the InnoDB engine, where we have "top" that we run from a Linux OS prompt, MySQL has something similar:

```
/mytop/innotop/ndbtop
```

There are a few other third party tools available to the DBA:

- SHOW PROFILE;
- The MySQL Toolkit
- Microsecond Patch for slow query granularity
- phpMyAdmin, which allows the DBA to personalize how a MySQL environment is monitored, managed and notification of issues

- MySQL Administrator, (GUI)

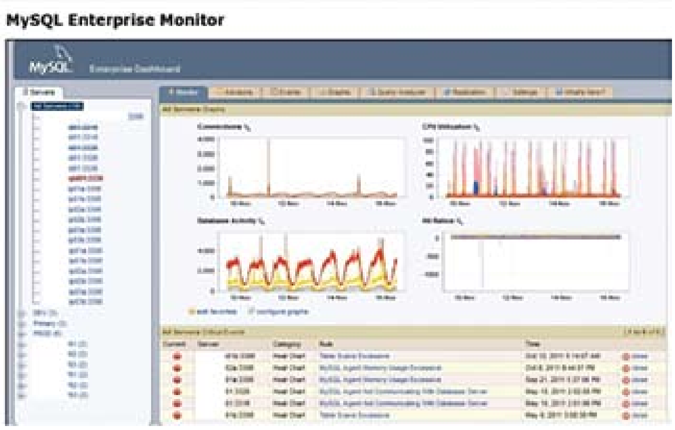
What should I be monitoring?

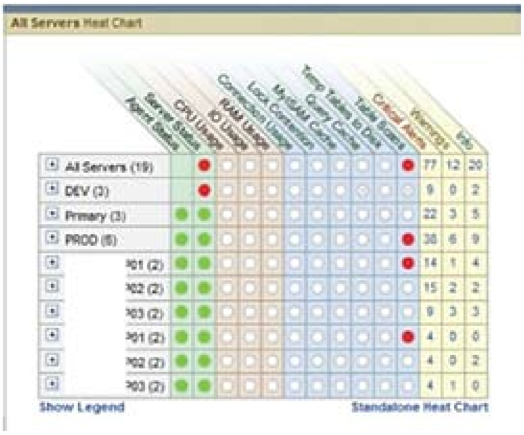
- Com-*
- Innodb_*
- Created_tmp_*
- And Connections

So you are monitoring, having issues, what are the most common issues and causes?

- Frozen- Check available space on disks
- Running slow- increase buffers, may need to change storage engine
- Can't connect at all - check connections, credentials and that the server is running

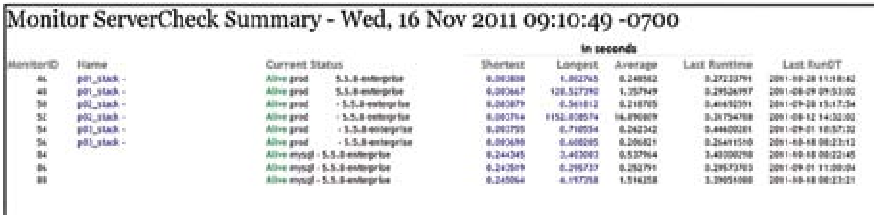
Monitoring through MySQL Enterprise Monitor:





Monitoring Through PHP Admin:

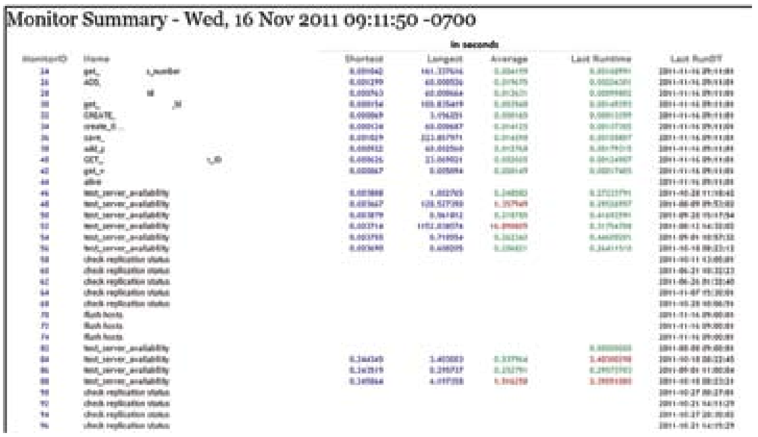
Monitoring at the server level with Scripted PHP Interface:



Monitor ServerCheck Summary - Wed, 16 Nov 2011 09:10:49 -0700

			in seconds				
MonitorID	Name	Current Status	Shortest	Longest	Average	Last Runtime	Last RunDT
46	p01_stack -	Alive prod 5.5.8-enterprise	0.003808	1.002765	0.248582	0.27233791	2011-10-28 11:18:42
48	p01_stack -	Alive prod 5.5.8-enterprise	0.003667	128.527390	1.357949	0.29526997	2011-08-09 09:53:02
50	p02_stack -	Alive prod - 5.5.8-enterprise	0.003879	0.561812	0.218705	0.41692591	2011-09-28 15:17:54
52	p02_stack -	Alive prod - 5.5.8-enterprise	0.003714	1152.038574	16.898809	0.31754708	2011-08-12 14:32:02
54	p01_stack -	Alive prod - 5.5.8-enterprise	0.003755	0.710554	0.262342	0.44600201	2011-09-01 10:57:32
56	p01_stack -	Alive prod - 5.5.8-enterprise	0.003698	0.608205	0.206821	0.26411510	2011-10-18 08:23:12
84		Alive mysql - 5.5.8-enterprise	0.244345	3.403003	0.537964	3.40300298	2011-10-18 08:22:45
86		Alive mysql - 5.5.8-enterprise	0.243509	0.295737	0.252791	0.29573703	2011-09-01 11:00:04
88		Alive mysql - 5.5.8-enterprise	0.245064	4.197358	1.516258	3.39051088	2011-10-18 08:23:21

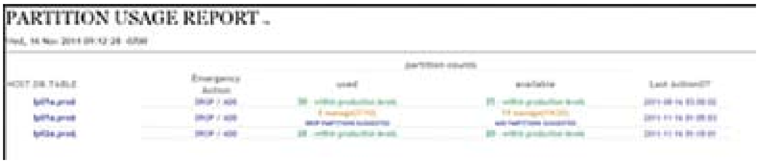
Summary through PHP Scripted Interface:



Monitor Summary - Wed, 16 Nov 2011 09:11:50 -0700

		in seconds				
MonitorID	Name	Shortest	Longest	Average	Last Runtime	Last RunDT

Partitions can even be monitored and even managed via a PHP Scripted Interface:



PARTITION USAGE REPORT

		partition counts		
HOST DB TABLE	Emergency Action	used	available	Last ActionDT

Monitoring Scripts:

```
SELECT TABLE_SCHEMA, SUM((DATA_LENGTH + INDEX_
LENGTH) / (1024 * 1024)) AS SIZE_MB
FROM INFORMATION_SCHEMA.TABLES
GROUP BY TABLE_SCHEMA ORDER BY SIZE_MB DESC
SELECT ROUTINE_TYPE, ROUTINE_NAME FROM INFORMATION_
SCHEMA.ROUTINES
WHERE ROUTINE_SCHEMA='dbname';
SELECT TRIGGER_NAME,EVENT_MANIPULATION,EVENT_
OBJECT_TABLE,
ACTION_STATEMENT
FROM INFORMATION_SCHEMA.TRIGGERS WHERE TRIGGER_
SCHEMA='dbname';
SELECT CONCAT('DROP TABLE ',table_name,';')
INTO OUTFILE '/sql/drop_tables.sql'
FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA =
'test';

SELECT s.schema_name, CONCAT(IFNULL(ROUND((SUM(t.
data_length)+
SUM(t.index_length))/1024/1024,2),0.00),'Mb')
total_size,
CONCAT(IFNULL(ROUND(((SUM(t.data_length)+SUM(t.
index_length))-
SUM(t.data_free))/1024/1024,2),0.00),'Mb')
data_used,CONCAT(IFNULL(ROUND(SUM(data_
free)/1024/1024,2),0.00),'Mb') data_free,
          IFNULL(ROUND((((SUM(t.data_
          length)+SUM(t.index_length))-
          SUM(t.data_free))/((SUM(t.
          data_length)+SUM(t.index_
          length)))*100),2),0) pct_used,
          COUNT(table_name) total_tables
          FROM information_schema.schemata s
```

```
LEFT JOIN information_schema.tables t ON s.schema_
    name = t.table_schema
    WHERE s.schema_name != ‘information_schema’
    GROUP BY s.schema_name ORDER BY pct_used DESC\G

    mysqladmin -r -i 1 extended-status | grep -v “
    | 0 “
```

***For the scripts above, as beneficial the information, one must keep in mind when deciding intervals to execute them for the most up to date information, they will be retained in the memory that is also required for the user’s processes.*

Cool Features in MySQL

Ability to capture DDL so easily, (no having to use DBMS_METADATA!):

```
SELECT CREATE TABLE <TABLE_NAME>;
Output data to flatfile from MySQL
SELECT col1,col2
FROM tbl1
INTO OUTFILE ‘/tmp/<file_name>.csv’
FIELDS TERMINATED BY ‘,’
ENCLOSED BY ‘”’
LINES TERMINATED BY ‘\n’
```

MySQL in the Oracle Marketplace

MySQL is often found in large and small environments. It has taken a large share of what were once Microsoft SQL Server environments and as the database platform becomes more popular and robust, it is more and more the choice for high-end database environments where only a few platforms dared to tread.

The Oracle DBA in today’s market should be familiar enough to give adequate support and know enough to work through problems if they do arise in MySQL. I have found each database platform I’ve been exposed to has at least one feature that we would all love to have in Oracle and MySQL is no different. For one to investigate the world of MySQL, learn a bit of what makes this database platform so popular and why Oracle Corporation took the time and interest to make it important to its database offering is worth any DBA’s time.

Kellyn Pot’Vin resides with her family in Westminster, CO and is a Senior Database Administrator and Developer at I-behavior, Inc. She’s been an experienced Database Administrator in Oracle and SQL Server for over eleven years and MySQL for five. She is on the Board of Directors for Rocky Mountain Oracle User Group, (RMOUG), serving as the Director of Membership and Vendors. She’s presented at RMOUG Training Days, is on the Database Track, (volunteer) for KScope 2012 for ODTUG and blogs at dbakevlar.com, along with other technical websites.

She can be followed on Twitter @DBAKevlar, or connected with on LinkedIn and Facebook.

PROUD TO BE A GOLD SPONSOR
RMOUG
TRAINING DAYS
February 14-16, 2012
Denver, Colorado
Colorado Convention Center
DBAK
Colorado owned and operated
www.dbaknow.com | 720.475.8600
ORACLE
Gold
Partner

Photos From RMOUG Members

Bryan Bachmeyer, Near Aspen.

Mike Landrum, Diamond Face of Longs Peak taken at Chasm Lake, Rocky Mountain National Park

Ray Mariano, Snow if on its way!

REGIS UNIVERSITY
School of Computer &
Information Sciences

RMOUG

Present Our DBLabs

Back in December of 2009, RMOUG and Regis University teamed up to start offering hands-on lab sessions on various topics. These events are open to the public (you do not have to be an RMOUG member), and they are a great way to learn new skills, get some hands-on experience in a lab environment, and network with other users and members! The meetings happen in one of the computer labs on the Regis campus in Denver, so each person gets their own workstation where they can do the labs. They are typically held on a weeknight evening, with food and drinks provided, so you can just show up, have some dinner, learn some new skills and meet new people!

Past topics have included OEM, Oracle on Linux, RMAN, Oracle database security, and Oracle automatic SQL tuning features. And we've had some well-know RMOUG "masters" as presenters, including Darl Kuhn and Tim Gorman. Future topics being planned include MySQL, OEM11g and more. And we are always open to new presenters and topics. So if you'd be interested in running a future lab session, feel free to contact

Brad Blake at SigsDir@rmoug.org.

To stay up-to-date and informed on upcoming sessions, be sure to join the group here:

http://www.meetup.com/RMOUGLabs/

Also, be sure to "friend" us on Facebook to get updates on the labs, as well as other RMOUG events:

http://www.facebook.com/RMOUG

You can also follow us on Twitter @RMOUG_ORG

RMOUG Member Focus

Mark Molnar

Being asked to write a little bit about myself turned out to be more difficult than I thought it would be. Like many of you, the two biggest areas where I spend my time are with family and work… I'm working on a way to get sleep added in there but it just hasn't happened yet.

Family & Personal. I live along the foothills in the southwest suburbs of Denver. I love Colorado. I love to travel, too, but having Colorado as a home is ideal. Prior to moving here in 1993 I spent about three years in Houston, TX and before that, the majority of my youth was in Southern New Jersey, just on the other side of the Delaware River from Philadelphia, PA (Camden County for those of you familiar with the area). I graduated from Rutgers University with a BA in Computer Science and a minor in Business Administration. You may have noticed I specifically stated, "Southern New Jersey". When you're from New Jersey, albeit small in size, it's commonplace to distinguish the difference between "South Jersey" and "North Jersey" for some reason. I believe it stems from the cultural influences from New York City (North Jersey) and Philadelphia (South Jersey) – two distinct cities with their own local personalities. I spent many summers at the shore and I can tell you, the reality show bearing the same name is probably the farthest thing from reality you can get, it gives the state a pretty bad stereotype (like it wasn't bad enough already!). That's all I'm going to say about that. Coming up on 20 years here, I consider myself a true, transplanted, Coloradan.

My wife & I are raising our five children to appreciate and enjoy the outdoors hopefully as much as we do. For those of you who don't know us, we have a 13 year old son and a set of 11 year old quadruplets (1 boy and 3 girls). We couldn't be more blessed! For all of you who gave us support over the years, and especially during those early ones, thank you. I'm sure it's true with all parents raising children that there are milestones noticed on a daily basis. A couple of those, for me, that stand out over the past few years are: 1) As a family, three years ago, we summited our first fourteener, Mt. Bierstadt (elevation 14060 ft.) and, 2) that same year, I didn't have to "lead" the way down a black-diamond ski run, rather, letting the kids go first, whichever "line" they wanted – ahh… ski lessons were over! I don't think I have much longer before I'll need a head start down the slopes so we'll meet at the bottom about the same time.

In addition to taking advantage of Colorado offerings, we also enjoy vacationing on the beach. Could it possibly be those Jersey shore roots pulling me back? We've taken a liking to more secluded destinations lately like a Floridian state park on the Keys, still under-appreciated, or the islands during the off-season. We love to snorkel and scuba when we can.

Grand Cayman

A couple years ago, we broke from our "beach" destination tradition and headed southwest to see some of our national parks, truly treasures in our backyard. We conquered Arches, Bryce, Zion, and the Grand Canyon in about 9 days. Our favorite hike was "The Narrows" in Zion – I'll let you look it up if you're unfamiliar with it; doing that at least once is worth it. A trip through some of these national parks is something I've always wanted to do and think there are some more trips like that in our future.

For recreation and fun during most of the year, I play in a couple adult hockey leagues and like to golf with the family.

Work. The other consumer of my time is work. As a consultant, I'm on-site with clients most of the time, but do find myself off-site quite often. A nice technological change of the times is the ability to work remotely with the same efficiencies. Either way, my second home is behind a keyboard.

Between my junior and senior year in college, I was fortunate to land an internship with Harvard University's Office for Information Technology in Cambridge, MA doing technical support. Their International Development program didn't require as much time as initially thought, so they started giving me small database projects to do using Paradox (for DOS). That's where I got my first taste and was sold on database applications.

I started off working a few years for DuPont in Delaware, getting schooled in CASE tools and delivering Oracle applications through iterative prototyping. DBA and programming skills are necessary, but it was here where I learned you need solid processes and methodology to turn programs into applications. When an opportunity to work on an internal project for Conoco (DuPont owned Conoco at the time), I thought it'd exciting to give Houston, TX a whirl; I was young and wanted to try something new. Calling Houston home was fine, and doing contract work with some of the largest Oil & Gas companies in the world was pretty exciting, but the local climatic zone (i.e. hot & humid with a touch of hot & humid) wasn't really for me. I spent a few years there but in 1993 I had an offer to come to Denver. After my first week, I knew I wasn't going to leave – and I haven't. From banking to energy, high-tech to publishing, telecom to beer – if you're in the Oracle world, you can usually find a company in the Denver area that interests you and is running Oracle.

I joined RMOUG the first year I was here. I recently came across the first pre-

RMOUG Member Focus

North Rim, Grand Canyon, AZ

L-R: Mark Parkins, Jason Wolters, & Mark Molnar

sentation I gave at the 1995 Training Days about writing and evaluating RFP's. It sounds pretty dull doesn't it? I suppose so, but the techie side of it was cutting edge at the time as we automated the process by weighted questions & answers for each category and evaluator, and the results were analyzed by an application using GUI forms with drill-downs, on-line graphs, and reports in pdf format; it's so much simpler to do with the tools we have today. I've been a member of RMOUG ever since and was on the Board of Directors for a while - a volunteer job I thoroughly enjoyed. Having five children under the age of 3 at home, I ran low on volunteer time and knew others could easily pick up the tasks I had and give RMOUG what it needed. I still enjoy friendships and working with many of those same members today. The RMOUG community continues to be an excellent organization and I hope everyone continues to not only attend meetings and training days, but also contributes their knowledge and experiences with the group through articles, presentations, and training days. There's something about this Oracle community that lends itself to good people with great ideas who make working in this Oracle arena a blast.

In 1997 I started MODE Technical Solutions, Inc. as an Oracle-centric consulting company to help deliver business applications and assist our clients' understanding of Oracle offerings in a fast-changing technological world. The pace has not slowed down, and by many measures, it has increased. Today I still run the company and find myself at client sites all week long, doing hands-on application work, which is what I enjoy the most. Bridging the gap between what the business needs and what the IT staff delivers is really where our expertise has thrived. I'm extremely proud of what we've been able to help our clients achieve, our consultants, and our reputation.

When family and work meet. Developing friendships at the workplace is commonplace and has, for me, lent itself to some memorable experiences. First, you never know where or when you'll run into a colleague. During that family hike up Mt. Bierstadt, I ran into John Malone, whom I've worked with on a number of projects dating back to 1996. He was out for the same hike with some friends. It was one of those pure timing/happen-stance moments when I took an unexpected phone call & he heard me talking to someone through a backup procedure (can we ever escape work?) that caught his ear and he realized it was me. We still joke about that. If you don't know John, he's a true Oracle professional in the Denver area and when it comes to financials, functionality and/or PM'ing a project with a deep knowledge, they don't come much better than John.

Another, more recent day, that was memorable for me was being able to get out of the office with some co-workers (late spring 2011) and hit the slopes. I consider these "off-site meetings" or "mental-health days." If you're looking for top-notch Oracle talent, Mark Parkins and Jason Wolters aren't just that, they sure know their way down a ski run, too. Here's a shot of that day at A-Basin:

Not that you're asking, but here's what I have to offer in closing.

- First, you're never too old to learn a new trick, so try to take the phrase, "because we've always done it that way" out of your vocabulary. You'll get outdated quicker than v8.0.3 in this business.
- We're surrounded by great minds with great ideas, so keep yours open and receptive and you'll be enlightened.
- There's more than one way to, "select from dual" – if you share yours with me, I'll share mine with you (markmolnar@modetech.com)!
- And finally, remember, it doesn't matter if it's work or family, volunteering or personal -å it's all a big puzzle. Some of the pieces will fall into place easy, some you have to work on for a while. What I've come to learn is that the harder you work, the easier it is next time.
- Finally, never give up, there's always a solution!

Mark

RMOUG Board Focus

Pat Van Buskirk

Well, I've cajoled, persuaded, and literally bugged everyone else to death on the Board about these articles, so someone mentioned it might be my turn. Karma – oops!

So I started thinking about how I got where I am now. *What a long, strange trip it's been.*

This will date myself horribly – I began my computer career on IBM mainframes in Cobol and Assembler. Total, an old hierarchical database, was my first peek into the world of databases and I wasn't real impressed. As with many of us, I worked on a ton of CICS code, both at the macro and command line level. This was Denver in the 80's and we were all contract programmers, always moving along to the next higher rate.

Oracle was an accident, albeit a profitable one. My husband Phil had an offer with a company in California who wanted him to work on this new database. The only thing I knew about Oracle was the Oracle of Delphi. Some goat herder around 1400 B.C. thought his sheep were acting a little weird around a chasm on Mount Parnassus. When he peered into the chasm, he described himself as "agitated like on frantic." Or at least that's the legend of the Oracle of Delphi. After 20+ years in Oracle, I have to agree with him.

The company asked us to move to California. Talk about sticker shock. This was 1990 and Denver was already sliding downhill. California was still on the upswing. We couldn't give our house in Brighton away for the measly mortgage of $80k and our new home in Oceanside was triple that. Oh, and did I mention we had a variable rate mortgage? And three months after we got there, the company went through a big upheaval and laid Phil off?

We were very lucky – we both found great jobs at a manufacturing company in San Diego. Maybe they hired me because I at least had heard about Oracle. My first project was a Forms conversion and somehow I pulled through it. We had no DBA, so I thought, "How hard can this be?" and volunteered. That was Version 6.0 and I grew all the way up through 11.2. Whew!

Meanwhile, I couldn't sit still. I had always been interested in medicine, so began attending night/weekend classes in Oriental Medicine. It took a while, but I finally achieved a Master's Degree. I took the last year off to study and pass the State and National Boards, still continuing to work the occasional Oracle contract.

That was 1996 and Phil was also living his dream. He has been restoring and racing old Porches for as long as I can remember and he had a chance to put together a race team to compete in the Vintage Auto Racing Association. We bought a semi to haul the cars and he, along with some friends, started building racing Porsches and taking folks to the track. It was a lot of fun and a lot of work! These were pre-1966 cars for the most part.

I'd always had this bug to study physics – go figure – and began taking classes in higher math and physics, first at Cal State, then UC Irvine. And, of course, someone needed database help and there I was again, back to where I started. My advisor told me about a DBA job at the National Radio Astronomy Observatory in New Mexico. It was a far cry from Southern California, but the pace and cost of living in California was beginning to wear on us. So, off we went. This was 2001. Once again, we were extremely lucky – the housing market had come back up and we even made a little money on that property.

We owned a little Porsche magazine, so Phil worked on the magazine while I learned about radio astronomy. After a few years there, we purchased some land and started a hay farm. I grew up in farming and ranching and have been a horse person all my life. We couldn't afford the luxury of a horse in California, but now we were back in the thick of them. We started rescuing horses and ended up with six.

It was 2006 and Phil had been having terrible problems with his hands and feet. Off we went to Mayo Clinic in Rochester, MN where he was diagnosed with genetic peripheral neuropathy. Basically, the nerves in his hands and feet were deteriorating. We searched for treatment and found the best and closest facilities/doctors were in Denver. NRAO was kind enough to let me telecommute so that Phil could receive treatment and in early 2009 we were back in Colorado, this time Northeast of Kiowa. Gotta have a place for those horses!

Two years ago, we discovered I had a congenital defect in my heart. "How in the world do you get this old and not know you have a big old hole in your heart," I asked. They didn't have an answer, but the solution was open heart surgery. *Gulp!* Some complications after the surgery left me unable to work full time, so I sat down and asked myself, "What can I do part time, on basically my own schedule, and still utilize my computer skills?" I knew every piece of horse property in Elbert County (because we had looked at them), so it was obvious – real estate. In this economy – are you nuts? Well, yes, especially after working in Oracle for twenty years.

So now I'm a Broker Associate at Coldwell Banker in Parker. I love the real

1997 Alfa - Porsche Challenge

estate business and helping people, especially in this economy, but I still get calls for database work. I currently serve on the Board of Trustees for the Centered Riding Organization in Pennsylvania where I help with web and database projects. I'm also the Web Chair and manage the volunteer database for the Elizabeth Stampede Rodeo, where I just received a buckle for 2011 Rookie of the Year.

It seems like I've been in Oracle User's Groups forever, starting with the San Diego group and now RMOUG. I've been on the Board for almost five years, previously served as membership director and received the Volunteer of the Year Award in 2009. With our publishing background, Newsletter Director seemed the perfect place where I could help RMOUG the most. I love putting together this magazine and get to meet and work with some really wonderful people. I also love Training Days, not only for the tremendous educational opportunities, but also the wonderful comradery we have.

Ok – I know you want to hear about our horses. We don't do any serious competition; they're just plain fun. Horses are a wonderful mirror who can teach you so much about yourself. People say they're mind readers. I don't believe that's really true. They are so good at reading body language that they can sense something you're thinking by the miniscule changes in your body that we can't perceive. They know exactly what my frame of mind is, and if I try to work with one of them when I've had a less than pleasurable day, they turn around as if to say, "Not now – go get your &*%^ together first!"

We have a twenty-five year old Quarter Horse stallion, his mare and their two daughters, Katie, who is three, and Sophie, two. Cash, the stallion, is the most wonderful horse in the world and unlike any other stallion. He's a gentleman and I can take him anywhere. I've given beginners lessons on that horse. His oldest daughter is, shall we say, a "challenge". We're working on teaching her to ride now and it's like trying to get a database back up after all your disks have crashed and the systems people have just told you they can't find the archived backups. We'll get there!

Our other horse, Doc, is a Belgian/Quarter Horse cross and a huge animal. He could pull a house and is as gentle as the proverbial lamb. Unfortunately, we just lost our other gelding, a great-grandson of Secretariat who looked just like him. He could have been one of the doubles in the movie.

My other love is music, which I've also done all my life. I began on piano at about age eight and also have played the flute, violin and oboe. A few months ago, my husband surprised me with a 100 year-old grand piano, the dream of a lifetime. It's absolutely beautiful and has the most luxurious sound. It's hard not to play it all the time, but I have to go get this newsletter out.

To all my friends in RMOUG, thank you so much for all your help and good wishes over the years. You are great people and I am so fortunate to be in a position where I can get to know so many of you. There's not enough room in this article to list everyone – you know who you are!

Oh, and if you know anyone with an old Porsche in their barn, garage or under a tarp out back, give Phil a call. He's always looking for his next project.

Happy trails and I hope to see you at Training Days!

ROCKY MOUNTAIN ORACLE USERS GROUP

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.

RMOUG SQL>UPDATE
The Magazine for the Rocky Mountain Oracle Users Group • Vol 57 • Winter 2009
Training Days 2010 • Register Today
PAGE 28
Don Burleson
Dan Hotka
Mathias Magnusson
The Case for CASE
Mark Molnar
Oracle & Excel - Why Fight It?

RMOUG SQL>Update Advertising Rates

Business card or 1/8 page	$ 50
1/4 page	$ 350
1/2 page	$ 625
Full page	$1,000
Inside cover	$1,250
Back cover	$1,500

Discounts available for RMOUG Members and Full Year Contracts

See Media Kit for deadlines and mechanical requirements.
Submit ad materials to: Pat Van Buskirk, RMOUG Newsletter Director
38101 Comanche Creek Road • Kiowa, CO 80117
303-621-7772 • E-mail: NewsletterDir@rmoug.org • www.rmoug.org

SQL>UPDATE is mailed to all RMOUG Members and distributed during Quarterly Education Workshops and Training Days each year

RMOUG Board of Directors

Meet Your Board

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

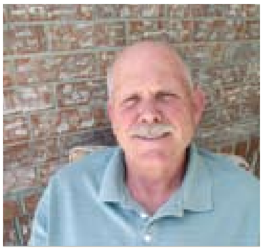
Kellyn Pot'vin
Membership & Vendors Director
E-mail: Membership@rmoug.org

John Jenuette
Training Days Director
E-mail: TtainingDaysDir@rmoug.org

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

Ron Bich, SofTec Solutions, Inc.
IS Director
Phone: 303-650-6951
E-mail: TrainingDaysDir@rmoug.org

Thomas Green
Scholarship Director
E-mail: ScholarshipDir@rmoug.org



Carolyn Fryc
Programs Director &
Vice President
E-mail: ProgramsDir@rmoug.org
VicePresident@rmoug.org

Brad Blake
SIGS Director
E-mail: SIGSDir@rmoug.org

Heidi Kuhn
Administrative Assistant
Voice Mail: (303) 948-1786
Fax: (303) 933-6603
E-mail: Admin@rmoug.org

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

RMOUG Events Calendar

Now Open	Training Days	Early Registration www.rmoug.org in Denver
1/12/12	Training Days	Deadline for Early Bird Registration
1/15/12	Newsletter	Call For Articles, SQL>Update Spring Edition
1/17/12	Board	January Meeting, Location TBD
1/18/12	OTN	Oracle TechNet Sysadmin Day at Utah Oracle Users Group http://www.utoug.org
2/1/12	Newsletter	Call For Cover Photos, SQL>Update Spring Edition
2/9/12	Training Days	Deadline for Standard Registration Pricing
2/14/12	OTN	Oracle TechNet Developer's Day in Denver - Registration is FREE
2/14/12	Training Days	RMOUG University Sessions, Top Speakers in Denver
2/14/12	Board	February Meeting, Colorado Convention Center in Denver
2/15-16/12	Training Days	Training Days 2012 - Denver Convention Center
3/4-8/12	Hotsos	Hosos Symposium in Las Colinas, TX (near Dallas) http://www.hotsos.com
3/6-7/12	UTOUG	Utah Oracle Users Group Training Days 2012 in Salt Lake http://www.utoug.org
3/18/12	Newsletter	Spring Issue of SQL>Update Mailing
3/20/12	Board	March Meeting, Location TBD
4/22-26/12	Collaborate	Collaborate 2012 Conference in Las Vegas, NV http://www.ioug.org
5/18/12	QEW	RMOUG Quarterly Education Workshop - Location TBD
6/24-28/12	ODTUG	Oracle Develope Tools Users Group Kaleidoscope 12 in San Antonio, TX http://www.odtug.com

Please note dates are subject to change. For the most current events calendar visit our website at www.rmoug.org.

RMOUG ROCKY MOUNTAIN ORACLE USERS GROUP

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

	Breakfast	Discounted Ad Rate	Total Cost
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

Help RMOUG Members and Receive Recognition in An Upcoming Issue of SQL>Update

Contact Carolyn Fryc - Programs Director - 720-221-4432 - cfryc@orsportal.com

Index To Advertisers

RMOUG ROCKY MOUNTAIN ORACLE USERS GROUP

Quarterly Education Workshops

May 18, 2012

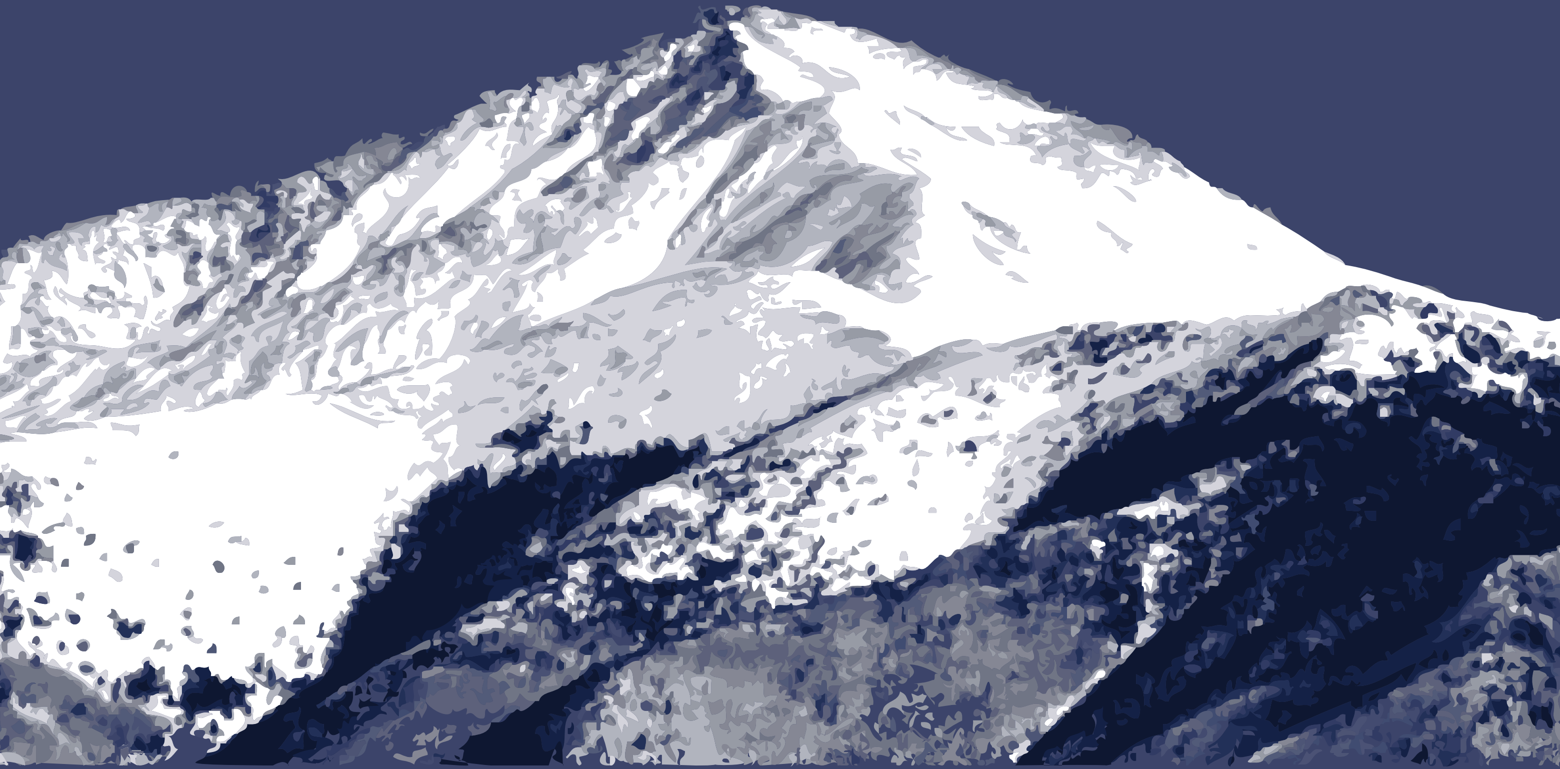
Join us for our next Quarterly Education Workshop in November. The location will be announced via email or please check our web site. RMOUG hosts quarterly workshops in May, August and November of each year with the fourth and largest educational event being Training Days in February. Learn about the newest technologies, gain more insight into Oracle techniques and enjoy the camaraderie of meeting with other Oracle professionals.

If you or your organization are interested in partnering with RMOUG to host an upcoming meeting, or to submit an abstract for presentation, please contact

Carolyn Fryc, Programs Director at ProgramsDir@rmoug.org

Watch RMOUG's Web Page for May Training Topics ***www.rmoug.org***

MARK YOUR CALENDAR NOW!



THE COLLEGIATE PEAKS PHOTO COURTESY OF JOHN KING, RMOUG MEMBER

RMOUG TRAINING DAYS

February 14-16, 2012 | Denver, Colorado

Colorado Covention Center

www.rmoug.org

Training Days has a long tradition of excellence

The price is right

The networking atmosphere is perfect

Training Days is for Developers, DBAs, and Managers

Early registration through January 12

For more information, visit

www.rmoug.org

Training Days 2012