Maaz Anjum
Capacity Planning With EM12c

Danny Bryant
Regular Expressions - Love/Hate Relationship

Melanie Caffrey
Edition Based Redefinition

Sarah Zumbrum
Optimizing ASO Cubes

Recognizing Our Volunteers
Board Focus - Dan Hotka
Training Days 2014 Remembered
Focused on Delivering the **right** Solution for the **right** Project

Arisant focuses on understanding business requirements to ensure the most effective implementation of the right solution. Our strategy is aimed at providing an honest and expert brand of consulting services for both the Private and Public Sectors. Contact us today to discuss any of the following services, and more.

- Consulting Services
  - Architecture
  - Analysis
  - Design
  - Implementation
  - Project Management

- Staffing/Support
- Managed Services
- Managed Hosting Services
- Oracle Software Sales
- Oracle Hardware Sales

**Oracle Experts, Easy to Work With**

**Oracle Database** | **Data Strategy** | **Managed Services**
**Oracle E-Business Suite** | **Oracle Exadata** | **Oracle License Reseller**

www.dbak.com | 720-475-8600

Proud to be a Gold Sponsor for RMOUG 2014 Training Days
features

6 Capacity Planning
   by Maaz Anjum
   EM12c

12 Regular Expressions
   by Danny Bryant
   It’s A Love-Hate Relationship

14 Edition-Based Redefinition
   by Melanie Caffrey
   Testing App Upgrades Without Being Live

22 Optimizing ASO Cubes
   by Sarah Zumbrum
   A Personal Case Study

departments

16 Training Days 2014 Remembered
25 Recognizing Our Volunteers
26 RMOUG Board Focus
   Dan Hotka

On the Cover:

Jeff Stephenson is an application developer at Colorado State University with nearly a decade of experience working with Oracle databases and PL/SQL. Having lived in Colorado most of his life, he has a propensity for the west and tries to venture into the mountains as often as he can. The image on the cover was taken last year at Great Sand Dunes National Park in southern Colorado.

After a morning hike with my wife and daughter in the dunes we decided to return to camp when the customary afternoon storm clouds started billowing in. The clouds made for some very dramatic skies and led to a short but intense storm which we enjoyed from the campground below the dunes.

You can see more of his images at JeffStephensonPhotography.com.

Many thanks to Jorge Rimblas for sharing his wonderful photos of Training Days 2014!

editorials

4 Letter From Our President
5 Stan Yellott Scholarship Fund
28 RMOUG Board of Directors
30 RMOUG Calendar of Events
30 May 2014 QEW

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. ©2014 Rocky Mountain Oracle Users Group

Many thanks to Jorge Rimblas for sharing his wonderful photos of Training Days 2014!

A special Thank You to Heidi Kuhn, Tim Gorman and Kellyn Pot’Vin, without whose continuing help, this publication would not be possible.
The 25th “Training Days” conference is now in the books, and by all accounts it was a smashing success. While everything hasn’t yet been totaled up officially, the conference was attended by the most we’ve seen in many years. Responding to feedback from previous conferences, the planning committee, led by director Kellyn Pot’vin, had made some important changes to the conference to make it tighter and leaner...

- added a new track for the fast-growing Hyperion/EPM topics
- reduced the number of rooms (from 13 to 10)
- reduced the number of timeslots so there were no more empty presentations at the end of each day
- introduced economical “day passes” for those who could not attend the full 2.5 day conference
- retired the separately-priced 4-hour “university sessions” and created 2-hour “deep dive” sessions integrated as part of the main conference

There were numerous other smaller improvements, but these big changes reduced expenses and simplified a great many things, tightening and sharpening a conference whose costs were increasing and growing beyond manageability. The results were apparent, as anecdotal feedback from the conference itself was uniformly positive.

The entire world was aware of what was going on here in Colorado, as the twitter universe resounded with tweets about the conference, marked by the hashtag “#RMOUG”. Our annual conference is one of the most highly-regarded worldwide, consistently ranked alongside conferences such as Oracle Open World, Collaborate, KScope, UKOUG, DOAG (Germany). We are indeed fortunate to be able to provide 2.5 days of world-class educational content by the best speakers -- and the newest brightest speakers -- in the world, all here in downtown Denver, without needing to hop on a plane. After 25 amazing conferences, the biggest mystery is why so many of our colleagues can’t attend. After all, where else in the world can you learn so much from so many over 2.5 days for less than $500?

If you attended the conference, we hope you enjoyed everything. If you did, or if you felt something could be improved, please feel free to email Kellyn at “TrainingDaysDir@rmoug.org” or respond to the survey sent out immediately after the conference.

For a glorious week in the beginning of the year, Colorado was the center of the worldwide Oracle community. And next year will be even better! We are now working toward making parts of the conference available online, either live from your workstation, or replayed from recording at any time you wish over the following months. Keep an eye on this newsletter, the RMOUG.ORG website, the @RMOUG.ORG handle, and the “Rocky Mountain Oracle Users Group” group on LinkedIn over the next six months for more information as it comes available.

And now, onward!

We have a full year of activities planned at RMOUG for 2014...

- On Friday 16-May, we’ll be having our Spring quarterly educational workshop (QEW) at the Oracle/Sun campus in Broomfield
- On Friday 08-August, we are planning our spectacular Summer QEW at the Elitch Gardens Water & Theme park, including a half-day of technical sessions followed by day passes for members and their friends and family at Elitches
- And then on Friday 14-Nov, we’ll have our Autumn QEW in the DTC area, probably at the Oracle campus in the DTC

If you have any interest in speaking, volunteering, or sponsoring all or part of any of these meetings, please contact Carolyn Fryc (ProgramsDir@rmoug.org or cfryc11@gmail.com)

In addition to these quarterly meetings, the award-winning and widely acclaimed RMOUG newsletter “SQL>Update” will be providing an amazing lineup of articles from well-known and new brilliant voices, once per quarter. Past issues of the newsletter back to 2009 are now available on the RMOUG.ORG website, under the “Newsletter” tab. Feel free to browse and review this valuable online resource. If you have any interest in helping with the newsletter, writing articles, or advertising within, please contact Pat Van Buskirk at “NewsletterDir@rmoug.org”

Also, look for events presented by our three special-interest groups (SIGs):

- Hyperion/EPM
  - led by Chris Chase (cchase@thebean.co) of The Bean Company
- Big Data
  - led by George Trujillo (george.trujillo@me.com) of Hortonworks
- MySQL
  - led by George Trujillo (george.trujillo@me.com) of Hortonworks

If you’re interested in any of these specific areas of interest, please contact the leaders above or contact Vince Giasolli (sigsdir@rmoug.org or vgiasolli@gmail.com), our director of special-interest groups.

Another thing about the May QEW at the Oracle/Sun campus is that it will also host the election of a new board of directors for RMOUG. We are always seeking energetic volunteers for the
board, and we would be delighted to have you join us and help keep RMOUG providing world-class professional education for another 30 years. Please contact me at “president@rmoug.org” or our executive director Heidi Kuhn (heidikuhn@rmoug.org) if you are interested? Also, please see the “board application” link at the bottom of the “Current Board of Directors” page at RMOUG.ORG. To get there, click on the “About” menu, choose “Current Board Of Directors”, and scroll to the bottom.

If you are interested in volunteering at RMOUG for any position, we also have a “Volunteer” contact form on the front page of RMOUG.ORG.

RMOUG continues into its fourth decade of existence, so please come on and join us for an exciting 2014!

RMOUG Scholarship Mission

To provide educational opportunities to members of the organization about the information technology industry in general, and in particular the technology of Oracle Corporation to include databases, storage, networking and application development, specifically the products and services of the Oracle Corporation.

To collect, post and distribute information about Oracle technologies and other related technologies to members.

To provide members with the ability to help their peers maximize their knowledge and skills working with products in information technology and in particular Oracle products.

To provide a consolidated channel of communication, conveying needs, concerns, and suggestions, for members of the organization, to Oracle Corporation and other vendor corporations involved with Oracle related technology.

To encourage members to present their information technology experiences using Oracle and other products and services.

To provide a consolidated channel of communication between members of the RMOUG and other communities in related information technology industries.

To promote educational opportunities for students of information technology through directed funding and services for educational purposes.

RMOUG is committed to supporting others in the pursuit of technical knowledge.

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

For Details, Visit the RMOUG Website

www.rmoug.org

Pete Sharman, Oracle Asia/Pacific,
Presenting at Training Days 2014

Leighton Nelson of Mercy Hospital gave us a thumbs up!
Capacity Planning with EM12c

by Maaz Anjum

A few months ago, I was approached by Oracle to answer a peculiar question from a mutual customer. Just so I won’t need to refer to them as “the customer”, let’s call them Acme. With subtle pushes from Oracle the folks at Acme were “interested” in upgrading their X2-2 Half Rack to an X3-2 Full Rack, but were skeptical on whether they really needed the additional horsepower and storage. The simple question posed to us was “How do you know we need more resources?” I had to admit that was a valid question.

Acme had an existing 11g Enterprise Manager which was upgraded to 12c to retain the collected metrics from the previous years. Retaining historic data allows for the Acme team to view trends in the past. However, data representation via the “All Metrics” link in EM12c did not depict enough information for Resource Utilization across a Cluster. Not only did they want to view the metric breakdown over time based on the “hardware”, i.e. Cluster, Host, Database, Database Service etc., but at a Business Unit (Departmental Hierarchy) as well.

So, armed with that, Acme’s Capacity Planning initiative, and my adventure with the annals of EM12C metrics began.

I am quite familiar with EM12C’s new features, one of which struck me particularly relevant with Acme’s representation/aggregation at a Business Unit level- Chargeback. EM12c’s licensable Chargeback feature, introduced in release 12.1.0.1.0, is part of the Cloud Builder functionality inherently built into the tool. It allows for metered charging of hosted resources to internal departments, or external customers. This allows the business to assign a monetary charge ($) to a metric over a specific internal (referred to as either time or cycles). Acme, was not interested with actually charging their internal departments, but the feature offered promise to resolve the Departmental distribution. Once the hierarchy is defined within Chargeback, and metrics defined, utilization charts are easily extracted. The produced charts however, proved to contain inadequate data due to four challenges. First, there was a slight deficiency identified within the CPU Utilization calculations by the product team. The second issue was that the produced reports were confined to a 30-day period. A third issue was identified when Chargeback could not be configured to send periodic notifications and finally, there was the fourth problem requiring a purchase of additional licenses to leverage the hierarchy in Chargeback. Due to these four issues, an elegant solution to collect and visualize the metric data using graphs was proposed as a simple design, which contained the correlation between Database(s) to Application(s) via a mapping table.

The resulting graphs are aimed to assist key business decision makers, as well as technical personnel, a way to visualize their environment from a resource utilization, growth, and capacity perspective.

I think most of us can agree that it’s always best to set expectations at the beginning of any project.

The requirements for the project were:

- What defined their version of resource utilization metrics?
- How would the data be represented and how best to interpret the data?

The subsequent representation of the data would need to clearly demonstrate growth over varying periods of time, (days, weeks, months, quarters, and years). This data would then be used to project future growth and allocate resources. Trends from increases or decreases in utilization will then be translated into budgets for future hardware to fulfill the business’ needs.

There was a final agreement on four metrics, split into two categories:

The first category, at a host level:
1. CPU Utilization
2. Memory Utilization
3. Storage Usage

The second category was at the Database Level
4. DB CPU Time. The latter should not be confused with DB Time which is an aggregation containing DB Time, IO, Network etc.

Using EM12c metric collection, I’d like to begin by showing my readers the methodology I used to answer Acme’s questions by leveraging the collected metrics.

EM12c utilizes its deployed agents to collect metric data from all targets monitored on the specific host. The frequency of the collection depends on the frequency set in either the target metrics, or a monitoring template. For example, the screenshot below illustrates just a few host metrics and their respective collection times.
Once the data are collected within the SYSMAN schema’s core tables, it is aggregated over daily, and weekly intervals. Oracle has exposed the collected data via the gc$metric, and mgmt$metric views.

### The Making of The Graph

Prior to talking about the method, I want to mention the source metrics I used to produce the information. The column names correspond the column names in the gc$metric_values table.

<table>
<thead>
<tr>
<th>Target Type</th>
<th>Metric Group Label</th>
<th>Metric Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Load</td>
<td>usedLogicalMemoryPct</td>
<td>Used Logical Memory %</td>
</tr>
<tr>
<td>Host</td>
<td>Load</td>
<td>logicMemfreePct</td>
<td>Free Logical Memory %</td>
</tr>
<tr>
<td>Host</td>
<td>Load</td>
<td>cpuUtil</td>
<td>Host CPU Utilization %</td>
</tr>
<tr>
<td>Host</td>
<td>Disk Group Storage</td>
<td>usable_file_mb</td>
<td>Run Queue Length (Average Load 5 Minutes)</td>
</tr>
<tr>
<td>Host</td>
<td>Disk Group Storage</td>
<td>usable_total_mb</td>
<td>Usable Storage in ASM Disk Group</td>
</tr>
<tr>
<td>Database</td>
<td>Data Collection</td>
<td>sga_max_size</td>
<td>SGA Memory Size</td>
</tr>
<tr>
<td>Database</td>
<td>EM Database Services</td>
<td>pga_aggregate_target</td>
<td>PGA Memory Size</td>
</tr>
<tr>
<td>Database</td>
<td>Database Disk Group Storage</td>
<td>DBcpu_delta</td>
<td>Database CPU Time</td>
</tr>
</tbody>
</table>

The simple route I took when creating these graphs is outlined below. I happened to use Excel because it was easily available, but many other tools exist to create the data, including Tableau and Oracle’s own BI Publisher. Before I go through the actual steps, instead of relying too much on data transformations via a tool, I leveraged a few neat features in Oracle SQL such as WITH and PIVOT to build my base queries. On top of the base tables, I created additional views to refine the data for the specific Time Slice (day, week, month, or quarter) and Target (Database Machine, Host, Database).

For example, the base view for CPU Utilization (%) contains the raw data at the lowest level.

```sql
WITH
    base_one AS
    (SELECT
database_machine,
metric_column_label,
metric_column_name,
metric_group_label,
collection_time,
TO_CHAR(collection_time,'yyyy') || '-Q'
|| TO_CHAR(collection_time,'q') AS year_quarter,
eXTRACT(YEAR FROM collection_time)
|| '/'
|| TRIM(TO_CHAR(exTRACT(MONTH FROM collection_time),'09')) AS year_month,
collection_time AS year_month_day,
ROUND(avg_value,2) AS avg_value,
max_value,
stddev_value
FROM
sysman.gc$metric_values_daily
WHERE
database_machine = 'host'
AND metric_column_name = 'cpuUtil'
AND metric_group_label = 'Load'
),
base_two AS
(SELECT
database_machine,
metric_column_label,
metric_column_name,
metric_group_label,
collection_time,
TO_CHAR(collection_time,'yyyy') || 'M'
|| TO_CHAR(collection_time,'9') AS year_month,
collection_time AS year_month_day,
ROUND(MAX(max_value) over (partition BY
database_machine year_month), 2)
AS per_m_dbm_max_cpuutil_pct,
ROUND(AVG(avg_value) over (partition BY
database_machine year_month), 2)
AS per_m_dbm_avg_cpuutil_pct,
ROUND(Percentile_cont(0.05) within GROUP
(OVER BY max_value DESC)
```
UPDATE over (partition BY database_machine, year_month), 2) AS per_m_dbm_95th_cpuutil_pct FROM base_one )
SELECT DISTINCT database_machine, metric_column_label, year_month, per_m_dbm_max_cpuutil_pct, per_m_dbm_avg_cpuutil_pct, per_m_dbm_95th_cpuutil_pct FROM base_two ORDER BY year_month;

Per Per Per
Month Month Month
Metric  Max Avg 95th
DB   Column Year CPU CPU CPU
Machine Label Month"  Util% Util% Util%
------- -------------------       ----  ----- ----
exa01 CPU Utilization (%) 2013-01 98.88 10.52 89.44
exa01 CPU Utilization (%) 2013-02 99.87 15.17 99.82
exa01 CPU Utilization (%) 2013-03 99.9 12.44 99.71
exa01 CPU Utilization (%) 2013-04 99.8 11.61 99.37
exa01 CPU Utilization (%) 2013-05 99.77 9.93 96.93
exa01 CPU Utilization (%) 2013-06 99.91 11.79 99.27
exa01 CPU Utilization (%) 2013-07 99.38 7.98 93.08
exa01 CPU Utilization (%) 2013-08 99.82 10.72 97.61
exa01 CPU Utilization (%) 2013-09 99.73 9.77 86.75
exa01 CPU Utilization (%) 2013-10 98.57 8.1 74.61
10 rows selected

For systems with multiple hosted applications, it is tough to relate host level metrics to a Departments. The only useful way I found to relate any Department Level utilization was for Database Services. The creative way is by placing the departments within a hierarchy. For example, Finance -> Accounts Payable -> AP_APP (Database) -> RAC_SVC_AP_APP (RAC Service).

I encapsulated the mapping within a view (underlying table), with a structure like below and simply correlated the metric query with it.

```
CREATE TABLE portfolio {
    line_of_business NOT NULL VARCHAR2(4000),
    department NOT NULL VARCHAR2(256),
    application NOT NULL VARCHAR2(256),
    database_name NOT NULL VARCHAR2(256),
    service_name NOT NULL VARCHAR2(256));
```

SQL> SELECT * FROM portfolio;

That combined with the metrics data (see query below) will produce an interesting result.

```
WITH base_one
AS (SELECT SUBSTR (i.host_name, 1, 4) AS database_machine,
    j.host_name,
    -- i.database_name,
    UPPER (i.target_name) AS join_database_name,
    COALESCE(SUBSTR (i.target_name, 1, INSTR (i.target_name, '_', 1) - 1), i.target_name) AS database_name,
    entity_name AS instance_name,
    UPPER (SUBSTR (entity_name, INSTR (entity_name, '_', 1) + 1, LENGTH (entity_name))) AS instance_name_short,
    UPPER (key_part_1) AS service_name,
    collection_time,
    TO_CHAR (collection_time, 'yyyy') || '-Q' || TO_CHAR (collection_time, 'q') AS year_quarter,
    EXTRACT (YEAR FROM collection_time) || '-' || LTRIM (TO_CHAR (EXTRACT (MONTH FROM collection_time), '09')) AS year_month,
    TO_CHAR (collection_time, 'yyyy-ww') AS year_week,
    collection_time AS year_month_day,
    ROUND ( (avg_value) * count_of_collections, 2) AS db_cpu_time_per_service,
    a.count_of_collections FROM sysman.gc$metric_values_daily a,
    sysman.mgmt$db_dbninstanceinfo i,
    sysman.mgmt$os_hw_summary j
WHERE SUBSTR (i.host_name, 1, 4) LIKE
```

```
```
Aggregation for hosts in a Cluster was easy to represent. When CPU Utilization (%) data is aggregated over several months it can appear skewed. For example, the graph below represents data from an Exadata Database Machines/Clusters. The max values show that the cluster, as a whole, peaked at over 80%. To get a near perfect representation of the peaks, I utilized a 95th percentile value. Acme found the value also to be a good indicator of how often the “median” value compared against the maximum values for sustained periods of utilization.

Another useful way to look at the CPU Utilization (%) of the 4 nodes in a cluster is shown below. As you can see with this example, one can interpret that there was a huge spike in activities in the last 2 quarters.

Memory Utilization (%)

Memory utilization (%) was found to be proportional to the number of instances on the host. For the hosts I observed, the value remained mostly stable until instances were added, or removed. It was, in any case, a useful indicated for memory usage (used vs free)

**CPU Utilization (%)**

The most profound and relative metric for a host is its CPU Utilization. According to Oracle Documentation (1): “This metric represents the amount of CPU utilization as a percentage of total CPU processing power available”.

```sql
SELECT b.line_of_business, b.department, b.application, a.database_machine, a.host_name, a.database_name, a.instance_name, a.instance_name_short, a.service_name, a.collection_time, a.year_quarter, a.year_month, a.year_week, a.year_month_day, a.db_cpu_time_per_service
FROM base_one a, portfolio b
WHERE a.join_database_name like '%||b.database_name||%'
AND a.service_name = b.service_name;
```
Storage

With storage, I've found that a common question which always comes up is “How much have I allocated vs actually used?” Whether the utilization in question is within an ASM cluster or instance, Disk Group, Database, or Tablespace level, EM12c captures two basic metrics that I used: Usable and Total. This data can be extended to various groupings, by the mapping table mentioned previously.

3. It’s better to use the pivot function for graphs with multiple Hosts.

4. Create a pivot table, and chose the Row, Column, and Values from the data set.

5. Then, it’s a formality to choose your graph type.

6. And presto!

I couldn’t tell you about these cool graphs and not show you how to create them. See below for how I created one for CPU Utilization (%).

1. Run base queries to collect data. Use the query mentioned at the beginning of the article.

2. Once tabulated, export or copy it into Excel
The “metric” itself is not as important as its rendering. I’m quite sure that my readers are for better adept and fluent with Excel than myself, and are able to visualize their data in many ways.

At this point, I’m sure you are wondering why could we use any other tool for host level metrics? Nagios, Big Brother, WhatsUp Gold, and others are perfectly good examples where these metrics are captured. There are a number of reasons why I chose EM12C: supported by Oracle, built by them to monitor and manage Oracle products, designed to collect detailed metrics. We all love free things right? Well, there’s no charge for deploying, and monitoring EM12C as long as you own the Diagnostics and Tuning packs for the monitored databases.

Conclusion

When put in the right perspective, these reports will highlight growth trends, from a technical as well as a business point of view. Whether or not additional hardware is required is always a good question. In my opinion, most of the times it’s the application, which can be tuned to avoid hardware upgrades.

What caused the spike in CPU Utilization, or Memory?
Were there more database on-boarded, or was there excessive load on the existing ones?
What could attribute to the spikes in Storage growth?
And so on, and so forth. The idea behind building these reports in EM12C is simple. You already have the data at hand, so why not use it?

I hope this information I provided in the article was helpful to you, and perhaps you’d like to embark on your own resource planning adventure with these metrics. If you do, I would like to hear about your stories as well.

Until next time, Cheers!

Maaz Anjum is a Solution Architect with BIAS Corporation and is actively engaged with multiple clients to engineer solutions on various platforms and database versions by leveraging relevant technologies.

His active memberships include Independent Oracle User Group (IOUG), Georgia Oracle User Group (GOUG), Real Application Clusters Special Interest Group (RACSIG), and IOUG Big Data SIG. In addition, he is GoldenGate, Oracle Enterprise Manager, and Exadata Certified.

A techie at heart, his experience spans over 9 years on predominantly an Oracle stack ranging from core Database Administration, RAC, Performance Tuning, GoldenGate, Enterprise Manager, and Exadata. Outside of the professional arena, his interests include research in Big Data, Hadoop, and Data Visualization.

At a personal level, Maaz is a father of two adorable daughter, devoted husband, amateur photographer, part time hiker, food lover, and seeks to inspire and be inspired in all aspects of life.

Maaz considers himself an active member of the Oracle User community and take pride in sharing his knowledge and experience with others via his blog (maazanjum.com), twitter (@maaz_anjum), articles, presentations and speaking engagements.

Be Sure To Attend The
Spring Quarterly Education Workshop
May 16, 2014

RMOUG is seeking abstracts for this meeting. Presentations can include overviews, tips, techniques, and testimonials, and lessons-learned. For abstract submission, we are seeking the equivalent of a proposal for the presentation.

Please contact
Carolyn Fryc
ProgramsDir@rmoug.org
Regular Expressions
It’s A Love-Hate Relationship

by Danny Bryant

Sometimes you have to find or replace a particular set of characters in a column. Regular Expression is a powerful feature available in the Oracle database when it comes to pattern matching. However, implementing the regular expression in your SQL code may prove difficult. The goal of this article is to demystify the Regular Expressions `REGEXP_LIKE` and `REGEXP_INSTR` and give some simple concepts to aid in understanding them.

The Regular Expression is based on the use of meta characters, and Oracle provides a set of SQL functions that allow you to search and manipulate character strings using them. The meta characters are special characters that have special meanings that represent characteristics allowing this complex pattern matching to be achieved. While available in many programming languages, I will focus on the implementation in the Oracle database. Oracle’s implementation is an extension of the POSIX (Portable Operating System for UNIX) and is therefore compatible with its standard set forth by the IEEE.

This paper assumes the reader is familiar with simple pattern matching using the `LIKE` statement in conjunction with wildcards such as `%` and `_`. Sometimes however there is a need for a more complex pattern matching that cannot rely on the all or nothing approach available with the `LIKE` statement. Suppose you need to find a specific combination of characters in a string or all occurrences of a name which can be spelled different ways? Enter the Regular Expression.

Here are the provided Oracle functions:

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGEXP_LIKE</td>
<td>Regular Expression command using the LIKE operator</td>
</tr>
<tr>
<td>REGEXP_REPLACE</td>
<td>Searches for the pattern and replaces it with the replacement string</td>
</tr>
<tr>
<td>REGEXP_COUNT¹</td>
<td>Returns the number of times a pattern appears in a string</td>
</tr>
<tr>
<td>REGEXP_INSTR²</td>
<td>Searches for a particular string and returns the position</td>
</tr>
<tr>
<td>REGEXP_SUBSTR³</td>
<td>Search for a pattern within a given string. Returns the matched substring</td>
</tr>
</tbody>
</table>

Before diving into some examples, let’s take a look at some of these meta characters I will use in the examples, and what they represent¹.

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Specifies alternate matching criteria</td>
</tr>
<tr>
<td>^</td>
<td>Matches the start of a line or NOT</td>
</tr>
<tr>
<td>$</td>
<td>Matches the end of a line</td>
</tr>
<tr>
<td>[ ]</td>
<td>Delimits the list of matching expressions</td>
</tr>
<tr>
<td>[: : ]</td>
<td>Specifies a character class and matches all member of that class</td>
</tr>
</tbody>
</table>

Now we can explore how to use these Regular Expression meta characters to examine a small record set.

**Table DDL²**

```sql
CREATE TABLE "ADDRESSES"
(   "ID" NUMBER,
   "FNAME" VARCHAR2(30),
   "LNAME" VARCHAR2(30),
   "ADDRESS" VARCHAR2(30),
   "CITY" VARCHAR2(30),
   "STATE" VARCHAR2(30),
   "ZIP" NUMBER,
   "EMAIL" VARCHAR2(50),
   CONSTRAINT "ADDRESSES_PK" PRIMARY KEY ("ID") ENABLE,
   CONSTRAINT "CHK_EMAIL" CHECK (REGEXP_LIKE (email, '@')) ENABLE
);
```

CREATE OR REPLACE TRIGGER "bi_ADDRESSES"
before insert on "ADDRESSES"
for each row
begin
  if :new."ID" is null then
    select "ADDRESSES_SEQ".nextval into :new."ID"
  from sys.dual;
end if;
end;
```

ALTER TRIGGER "bi_ADDRESSES" ENABLE
```

CREATE SEQUENCE "ADDRESSES_SEQ" MINVALUE 1 MAXVALUE 99999999999999999999999999999 INCREMENT BY 1 START WITH 141 CACHE 20 NOORDER NOCYCLE

---
¹ Not an exhaustive list
² Code generated from Application Express 4.2.4
Examples

1. (REGEXP_LIKE) Suppose you needed to return the address of all people with first name of ‘Tom’ or ‘Thom’.
   If you use the simple Oracle wildcard expressions, you might find some difficulty in getting the result you want.

   SELECT * FROM addresses WHERE fname LIKE 'T%' OR fname LIKE 'Th%

   The above SELECT statement causes extra rows (ID 61 & 101) to be returned. Now let’s use a Regular Expression

   SELECT * FROM addresses WHERE REGEXP_LIKE (fname,'^(T|Th)om$')

   There we have it.

   Now let’s break down the REGEXP_LIKE function meta character arguments:

<table>
<thead>
<tr>
<th>Special</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>^</td>
<td>Start of the string</td>
</tr>
<tr>
<td>(</td>
<td>Start of the grouping</td>
</tr>
<tr>
<td>T</td>
<td>Look for T</td>
</tr>
<tr>
<td></td>
<td>Alternative match</td>
</tr>
<tr>
<td>Th</td>
<td>or look for ‘Th’</td>
</tr>
<tr>
<td>)</td>
<td>End the grouping</td>
</tr>
<tr>
<td>om</td>
<td>Last letters</td>
</tr>
<tr>
<td>$</td>
<td>End of the string</td>
</tr>
</tbody>
</table>

2. (REGEXP_INSTR) Suppose now you want to check for invalid addresses by identifying those entries that do not start with a number. Using Regular Expressions we can do this.

   SELECT address, REGEXP_INSTR (address, '^[[:alpha:]]$')
   FROM addresses WHERE REGEXP_INSTR (address, '^[[:alpha:]]$')=1;

   Since REGEXP_INSTR returns the numerical value of the first non-alpha character, we can test for that value to be greater than 1.

3. Now look at a very popular problem to address using Regular Expressions. In this final example, I will demonstrate using Regular Expressions to perform data validation upon a row INSERT. This will be done by embedding a CHECK constraint into the table definition to determine if the email address contains a '@' symbol. Let’s do that now.

   Add a CHECK constraint to my table definition

   BEFORE
   CREATE TABLE  "ADDRESSES"
   ( "ID" NUMBER,
     "FNAME" VARCHAR2(30),
     "LNAME" VARCHAR2(30),
     "ADDRESS" VARCHAR2(30),
     "CITY" VARCHAR2(30),
     "STATE" VARCHAR2(30),
     "ZIP" NUMBER,
     "EMAIL" VARCHAR2(50),
   )
   CONSTRAINT "ADDRESSES_PK" PRIMARY KEY ("ID") ENABLE,
   CONSTRAINT "CHK_EMAIL" CHECK (REGEXP_LIKE (email, '@')) ENABLE)

   AFTER
   CREATE TABLE  "ADDRESSES"
   ( "ID" NUMBER,
     "FNAME" VARCHAR2(30),
     "LNAME" VARCHAR2(30),
     "ADDRESS" VARCHAR2(30),
     "CITY" VARCHAR2(30),
     "STATE" VARCHAR2(30),
     "ZIP" NUMBER,
     "EMAIL" VARCHAR2(50),
   CONSTRAINT "ADDRESSES_PK" PRIMARY KEY ("ID") ENABLE,
   CONSTRAINT "CHK_EMAIL" CHECK (REGEXP_LIKE (email, '@')) ENABLE)

   Attempt to INSERT a row with an email address missing the '@' symbol

   INSERT INTO addresses (fname, lname, address, city, state, zip, email) VALUES ('Bad','EMail','1600 Missing Rd','Symbol','IA',87622,'b.mail.slice.net')

   ORA-02290: check constraint (AFFIXDEV CHK_EMAIL) violated

Conclusion

Regular Expressions can be a very powerful tool, and there are many more applications than what have been presented here. Hopefully this has given you a brief overview of how to use Regular Expressions in your day to day development.

About the Author

Danny Bryant is a Senior IT Manager with the City of Atlanta and has a BS Degree in Computer Science with specializations in Graphics and Data Management
The problem of how to successfully upgrade a PL/SQL application without incurring much downtime, but allowing yourself adequate post-upgrade testing has been an on-going problem since PL/SQL was first introduced. 11gR2 changed that with the introduction of Edition-Based Redefinition. Using Edition-Based Redefinition, it is now possible to have multiple versions of your code in place on production: one that is visible and usable by the every-day users of the application, and one that is visible only to developers, testers, and other privileged users.

As most developers and DBAs tasked with rolling out PL/SQL application upgrades can attest to, almost all PL/SQL application upgrades, no matter how small, require some amount of downtime in order to avoid the otherwise inevitable locking and blocking that occurs. It is often not possible to obtain long or frequent downtime windows, the testing window during downtime may not be long enough to be deemed adequate for the testers, and last, but certainly not least, an upgraded PL/SQL application can be difficult to back out of, if necessary. Oracle’s 11gR2 version of the database has changed this situation by providing developers and DBAs with a high-availability tool, Edition-Based Redefinition, that gives those responsible for rolling out PL/SQL application upgrades the ability to have more than one version (or, edition) of a PL/SQL application running in a database instance and schema at the same time.

EBR: Edition-Based Redefinition

Edition-Based Redefinition, or simply EBR, allows you to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.

As you can see, it is now possible, for example, for the SCOTT user to have more than one occurrence of an editionable object. PL/SQL objects are editionable, for example. This is so due to a change in the Oracle namespace resolution scheme as of Oracle 11gR2. PL/SQL objects are no longer required to adhere to the <schema>.<object> resolution scheme that was always the norm prior to Oracle 11gR2. For PL/SQL objects, the resolution scheme has been expanded to <schema>.<edition>.<object>. The edition is implied, never hard-coded by the developer.
SQL> CREATE OR REPLACE FUNCTION sal_increase
( p_increase IN VARCHAR2,
  p_employee IN NUMBER)
IS
  v_new_salary NUMBER := 0;
BEGIN
  SELECT (salary * p_increase) + salary
  INTO v_new_salary FROM employee
  WHERE employee_id = p_employee;
  RETURN v_new_salary;
END;

And to find out with which edition(s) this function is associated, you can always look to see in which edition(s) your PL/SQL objects have been installed by querying the USER_OBJECTS data dictionary view which has a new column called EDITION_NAME as of 11gR2, as the following query demonstrates:

SQL> select object_name, object_type, status, edition_name
2 from user_objects;

OBJECT_NAME | OBJECT_TYPE | STATUS | EDITION_NAME
----------- |------------ |------- |--------------
EMPLOYEE     | TABLE       | VALID  |              
SAL_INCREASE | FUNCTION    | VALID  | ORA$BASE      

As you can see, the SAL_INCREASE function is currently associated with only one edition, the default ORA$BASE edition. Notice also, that there is no edition name value for the EMPLOYEE table. This is because only PL/SQL objects like packages, procedures, functions, triggers and views (though not materialized views), private synonyms, and all of these objects’ related metadata such as GRANT privileges, are editionable. Tables are not editionable. (Though we will see shortly how tables can be made to seem editionable.)

Now let’s say that someone has requested an enhancement to the SAL_INCREASE function, and that it would be more correct to also pass in a date value that checks to see when the employee was hired. This function should now only return an increased salary value if the passed-in employee was hired before a specified time period. An additional set of mandates are that this code change be made with minimal downtime, be given sufficient testing by the QA team, and be implemented in such a way that its changes can be easily reversed. This situation is perfect for using EBR. But first, you’ll need to ready your application to use EBR.

Edition Setup
First, you’ll need to create a new edition. You need the CREATE ANY EDITION or DROP ANY EDITION system privileges to perform either of those actions. Once you have the CREATE ANY EDITION system privilege you may create a new edition using the CREATE EDITION command as follows:

SQL> create edition app_edition_2
2 as child of ora$base;
Edition created.

SQL> select * from dba_editions;

EDITION_NAME | PARENT_EDITION_NAME | USA
--------------|---------------------|--
ORA$BASE      |                     | YES
APP_EDITION_2 | ORA$BASE            | YES

The query above illustrates one of the new data dictionary views available in 11gR2, DBA_EDITIONS. This data dictionary view lists all of the editions available for your database, alongside each edition’s parent (in the result set above, since ORA$BASE is the default edition that is installed out of the box with any installation of 11gR2, it has no parent), and lets you know whether the listed edition is USABLE (since you can set an edition to be unusable). After you’ve created a new edition, you must enable your application user to be able to use the new edition (execute code) and/or alter PL/SQL units within the new edition (run CREATE OR REPLACE statements). In order to allow your application user to alter editionable objects within more than the default ORA$BASE edition, it must be altered to be editions-enabled. The following query demonstrates this syntax:

SQL> alter user app_user
2 enable editions;
User altered.

Now the application user, APP_USER, can run CREATE OR REPLACE and GRANT statements in multiple editions. However, it still needs to be granted explicit access to the newly-created edition, APP_EDITION_2, created earlier. This access is achieved with the following SQL:

SQL> grant use
2  on edition app_edition_2
3  to app_user;
Grant succeeded.

With this permission the application user will be able to switch to the new version and execute code within it by running SQL such as the following:

SQL> alter session
2  set edition = app_edition_2;
Session altered.

SQL> select SYSCERNSCT('USERENV', 'SESSION_EDITION_NAME')
2   AS edition FROM dual;

EDITION
--------
APP_EDITION_2

And if the user executes the same query from USER_OBJECTS demonstrated earlier, he will see the following results:

SQL> select object_name, object_type, status, edition_name
2 from user_objects;

OBJECT_NAME | OBJECT_TYPE | STATUS | EDITION_NAME
----------- |------------ |------- |--------------
EMPLOYEE     | TABLE       | VALID  |              
SAL_INCREASE | FUNCTION    | VALID  | ORA$BASE      
APP_EDITION_2 |             |       |              

The reason for this is that when an edition is created as a child of another edition it inherits its parent editions’ editionable objects. These inherited objects look no different between the parent and child editions when a child edition is first created. No change to the data dictionary for the inherited objects is necessary because the inherited objects are currently merely pointers to the parent edi-
2014 Remembered
when a user makes a change to one of the editionable objects in the new edition that the query above will yield different results. Now let’s consider the following patch enhancement to the SAL_INCREASE function:

```sql
SQL> CREATE OR REPLACE FUNCTION sal_increase
  2     (p_increase IN
  3              VARCHAR2,
  4              p_employee IN
  5              NUMBER, p_hire IN DATE)
  6     RETURN NUMBER
  7     IS
  8     v_new_salary NUMBER := 0;
  9     BEGIN
 10     SELECT (salary * p_increase) + salary
 11     INTO v_new_salary FROM employee
 12     WHERE employee_id = p_employee AND
 13     hire_date <= p_hire;
 14     RETURN v_new_salary;
 15     END;
```

With this new version of the SAL_INCREASE procedure installed into the APP_EDITION_2 edition, we can execute the following query to see just how many editions accessible to our application user have unique versions of this function:

```sql
SQL> select object_name, object_type, status, edition_name
  2     from user_objects_ae;
```

A query against the USER_OBJECTS_AE data dictionary view (new as of 11gR2) demonstrates how, once our application user makes a code change to the SAL_INCREASE function in the APP_EDITION_2 edition, that change actualizes the function within the edition and it is no longer pointing to the version of the function that exists in the parent edition. This is why you can now see two rows listed for the SAL_INCREASE function within the application user’s list of objects.

### 2) Editioning View

In many PL/SQL application upgrade patch scenarios you will most likely be making changes to PL/SQL units similar to the change shown with the SAL_INCREASE function. However, if you need to make only column additions or change the structure of transaction tables, you will need to create an editioning view for each table you wish to modify. An editioning view exposes a different projection of a table into each edition to allow each edition to see only its own columns. Since a table is not editionable, it cannot have the same name as any other object in your edition. Also, since you want your application to reference a table name in the same way it always has, and you don’t want to incur downtime to make structural changes to this table (and run into locking/blocking issues or having-to-recompile code issues), it becomes important to devise a way to accomplish both tasks.

The editioning view allows you to not have to change your application’s references to a particular table while, at the same time, make structural changes behind the scenes. First things first, to enable your application’s tables to be able to use editioning views and, therefore, use EBR functionality you’ll need to rename each table. For this particular act of renaming your tables, you will require an outage (hopefully a one-time outage). You can rename each table to differentiate it slightly from the editioning view you will create for it:

```sql
alter table rpm rename to rpm_t;
```

Then create an editioning view for each of your renamed tables with the original table name:

```sql
create editioning view rpm as select * from rpm_t;
```

If your intent is to change the structure of a column, then you do not change it (as this would invalidate any dependent code and would defeat the purpose of using EBR), but instead, you add a replacement column. For example, I work with the Unbreakable Linux Network product for Oracle Corporation. This means I work with RPMs, packages that are used to update and enhance a user’s Red Hat or Oracle Linux installation. These RPMs often have a structure that looks like this:

```
<table>
<thead>
<tr>
<th>Name</th>
<th>Epoch</th>
<th>Version</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>kernel</td>
<td>(null)</td>
<td>2.6.32</td>
<td>100.21.1.el5</td>
</tr>
<tr>
<td>kernel</td>
<td>(null)</td>
<td>2.6.18</td>
<td>92.1.6.el5</td>
</tr>
</tbody>
</table>
```

That is: name.epoch.version.release, together all comprise a single package (RPM). Supplying information about the latest, greatest RPM a user must download and install on their system is of paramount importance to us. All four parts are stored in VARCHAR2 columns. Figuring out the sort mechanism for the above can be done in SQL (using, for example, some carefully-coded SUBSTR and INSTR constructs), but can be a bit unwieldy at times. In one exercise we wanted to pre-parse RPM parts like VERSION and RELEASE as they are concatenated pieces in and of themselves. The goal was to try and store each dot-delimited piece of the VERSION and RELEASE columns in their own separate columns. For purposes of brevity the below example includes only versions and releases with four parts:

```
<table>
<thead>
<tr>
<th>Name</th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>R1</th>
<th>R2</th>
<th>R4</th>
</tr>
</thead>
<tbody>
<tr>
<td>kernel</td>
<td>(null)</td>
<td>2</td>
<td>6</td>
<td>32</td>
<td>(0)</td>
<td>100</td>
<td>21</td>
</tr>
<tr>
<td>kernel</td>
<td>(null)</td>
<td>2</td>
<td>6</td>
<td>18</td>
<td>(0)</td>
<td>92</td>
<td>6</td>
</tr>
</tbody>
</table>
```

The new VERS1 ... and REL1 ... etc. columns were created with the NUMBER datatype. This way, instead of comparing VARCHAR2 strings, we can compare individual numeric values.

To achieve this, we altered our table (in this example, RPM_T) as needed.

```sql
alter table rpm_t add (vers1 number(10), vers2 number(10) ... rel1 number(10), rel2 number(10) ... );
```

In the new example table layout above, (vers1, etc, is shortened to V1, etc. so that all columns can display in the example). At this point, the table contains the old VERSION and RELEASE columns, as well as the new VERS1 ... and REL1 ... and so forth, columns.

Since triggers are PL/SQL units, you will need to drop all triggers that refer to (in this example) RPM_T and recreate them on RPM. You will also need to revoke privileges from RPM_T and grant them to RPM. Indexes and constraints, however, will remain in force on RPM_T. They follow the rename from RPM to RPM_T as they are not editionable, themselves, and are associated only with
Other non-editable objects. Many, if not most, of your application upgrades can be completed just by using editions and editioning views.

### 3) Crossedition Triggers

If, while you have both editions available and usable, you cannot stop DML, then your application needs to keep pace with such changes. The most complex piece of EBR (and the one you will most likely use only seldomly) is the crossedition trigger. The crossedition trigger, for all intents and purposes, looks just like any other trigger except that it has special firing rules and a few extra keywords that tell Oracle that it is a trigger used specifically to keep multiple versions (editions) of applications in synch. Any crossedition trigger that needs to keep a parent and child edition of an application in synch is created in the child edition. The special firing rules work as follows:

1. When the parent edition’s table column values are changed, you need to propagate these changes to the child edition’s table columns. For this instance, you must put in place a forward crossedition trigger.

2. Conversely, when the child edition’s table column values are changed, you need to propagate these changes to the parent edition’s table columns. And, for this instance, you must put in place a reverse crossedition trigger.

For this example, our forward crossedition trigger looked similar to the following:

```sql
SQL> create or replace trigger rpm_fwdxedition
2    before insert or update of version, release on rpm_t
3    for each row
4 forward crossedition
5 declare
6  v_verstring VARCHAR2(50) := '.'||:new.version||'.
7  v_relstring VARCHAR2(50) := '.':'||new.release||'.
8 begin
9  :new.ver1 := substr( v_verstring, instr(v_verstring,'.',1,1)+1,
10     instr(v_verstring,'.',1,2) - instr(v_verstring,'.',1,1)+1,
11     '','1,1');
12  :new.rel1 := substr( v_relstring, instr(v_relstring,'.',1,1)-1);
13 end;
```

And our reverse crossedition trigger looked similar to the following:

```sql
SQL> create or replace trigger rpm_revxedition
2    before insert or update of ver1, ver2, ver3, ver4, rel1, rel2,
3    rel3, rel4, on rpm_t
4    for each row
5 reverse crossedition
6 begin
7  :new.version :=
```

As you can see what differentiates these particular triggers as crossedition triggers are the keywords, in each trigger example, just after the FOR EACH ROW command. The new keywords are FORWARD CROSSEDITION (which effectively means, whenever this trigger is fired, before making the requested DML changes, please make the changes that follow the BEGIN keyword in this trigger in the child edition for the RPM_T table), and REVERSE CROSSEDITION (whenever this trigger is fired, before making the requested DML changes, please make the changes that follow the BEGIN keyword in this trigger in the parent edition for the RPM_T table). If you are using reverse crossedition triggers, you are performing a hot rollover, since you are keeping both editions of your application in synch simultaneously.

Note also that these types of triggers are created upon the actual table, RPM_T, even though our other PL/SQL triggers have been recreated on the editioning view, RPM. This is because the crossedition trigger is a short-term high-availability solution that is intended solely to assist in an online application upgrade. Once you are satisfied with your upgrade results, there should be no need to continue firing any crossedition triggers, because there should be no need to continue supporting multiple versions of table columns. At some point, you will choose and stick with the version of the columns that best meets your needs, and mark the other columns as unused, and recoup the space used by the unused columns at a convenient later time.

### Migrating Any Remaining Data

Though crossedition triggers will help you to keep any changes in synch between columns of a table in two editions, eventually you’ll want to ensure that all data values have been successfully migrated from the old columns to the new columns. Remember that unless your crossedition triggers touch every old value, some values will have to be manually migrated to the new columns. If your table is not very big and it won’t adversely affect your application to lock the entire table, you can force every row to be migrated (via the forward crossedition trigger you have in place) by executing an UPDATE statement similar to the following:

```sql
SQL> update rpm_t
2  set version = version,
3  release = release;
```

However, if this table is quite large, consider updating it a little at a time with the (new as of 11gR1) DBMS_PARALLEL_EXECUTE package. You can create a task that will update (in the below example, ROWID) chunks of a table, at a time, therefore only locking small portions of a table at a time:

```sql
SQL> begin
2  dbms_parallel_execute.create_task(
3    ‘update rpm_t’,
4    dbms_parallel_execute.create_chunks_by_rowid
5    ( task_name => ‘update rpm_t’,
6      table_owner => user,
7        ...})
```

```sql
SQL> update rpm_t
2  set version = version,
3  release = release;
```
Then run the task with a chosen range of ROWIDs and level of parallelism:

```
SQL> begin
  dbms_parallel_execute.run_task
  ( task_name => 'update rpm_t',
    sql_stmt  => 'update rpm_t
                set version = version,
                release = release
                where rowid between :start_id and :end_id',
    language_flag => DBMS_SQL.NATIVE,
    parallel_level => 2 )
end;
```

When you are satisfied with the result, you can simply drop this task:

```
SQL> begin
  dbms_parallel_execute.drop_task('update rpm_t')
end;
```

**Moving to the New Edition**

Once you are ready to migrate your end users to the new edition you can grant them access to the new edition:

```
SQL> grant use on edition app_edition_2 to public;
Grant succeeded.
```

And create a logon trigger that sets the new default edition any time a user logs directly into the database:

```
SQL> create or replace trigger set_edition_on_logon
  after logon on database
  begin
    dbms_session.set_edition_deferred('APP_EDITION_2');
  end;
end;
```

And if you are using a connection pool and are using APEX you can change your configuration as follows:

```
SQL> begin
  dbms_epg.set_dad_attribute('APEX', 'database-edition','APP_EDITION_2');
end; --If using the PL/SQL Embedded Gateway
```

In your dads.conf file:

```
FsqlDatabaseEdition*
```

**Oracle 12c Enhancements**

When this feature was introduced in Oracle 11gR2, the cardinal rule of thumb was always noneditionable objects can never depend on editionable objects because, for the noneditionable object, editionable objects are invisible during name resolution. As of 12c, however, there are some exceptions to this rule. Two types of noneditionable objects can now depend on editionable objects with the use of an evaluation edition. An evaluation edition is simply a set of key words that must be present when either of these two objects is created or changed in order for them to be able to depend on editionable objects. For example:

```
CREATE MATERIALIZED VIEW refresh_sal_vals
  EVALUATE USING EDITION app_edition_2
  ENABLE QUERY REWRITE
  UNUSABLE BEFORE EDITION app_edition_2
  UNUSABLE BEGINNING WITH EDITION ora$base . . .
```

**Conclusion**

EBR is not for the convenience of the developer or the DBA. It is a high-availability solution. And like all high-availability solutions it requires that you implement it with planning and care. If as-close-to-zero downtime when performing PL/SQL application upgrades is one of your company mandates, then you can easily be brought closer to that goal with EBR. And, in the spirit of saving the best for last, it is nice to be able to inform you that EBR is freely available to any user of any version of Oracle 11gR2 on up.

**About the Author**

Melanie Caffrey is a senior development manager for Oracle Corporation. She is co-author of several technical publications including Expert PL/SQL Practices for Oracle Developers and DBAs, and Expert Oracle Practices: Oracle Database Administration from the Oak Table (Apress), and the SQL 101 series of articles for Oracle Magazine. She instructed students in Columbia University's Computer Technology and Applications program in New York City, teaching advanced Oracle database administration and PL/SQL development, and she is a frequent Oracle conference speaker and a member of the Oaktable Network.
Sector work with Atlanta, Danny was a Senior Engineer with Andersen, LLP where he focused on Business Process Re-Engineering for the Hospitality, Telecom, and Call Center industries.

In my free time, I SCUBA dive, teach the Brazilian Martial Art of Capoeira, and blog @ http://implementoracle.blogspot.com. Follow me on twitter @dbcapoeira

(Endnotes)

1 For more information on Regular Expressions, visit: http://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_regexp.htm#ADFNS1003

Become A Member For These Outstanding Benefits

- Quarterly Education Workshops
- Special Annual Training Days Rates
- Database Labs
- List Server Subscriptions
- SQL>Update Magazine

www.rmoug.org/member.htm

The SE Feature Leecher took away your Tuning and Diagnostic Pack.

Confio Ignite™ restores the database performance monitoring that you lose when moving from Enterprise to Standard Edition.

And at a much lower cost!

Go to confio.com/oracleSE
SQL>UPDATE  •  Spring 2014

In preface to this article, I want provide a bit of a dictionary of Enterprise Performance Management (EPM) terms for those on the “other side of the Oracle house”.

**Essbase cube:** A type of OLAP (multidimensional) database.

**Outline:** A collection of dimensions that represent a company. Finance, for example, might contain Periods, Years, Entities and Products, among others.

**Aggregate Storage Outline (ASO):** An outline in which the dimension are configured for fast aggregation and data retrieval. Rack and stack, if you will. There are limited options in regards to calculations on the data, and all calculations must be in MDX format.

**Block Storage Outline (BSO):** An outline in which the dimensions are configured as either dense or sparse. Also, these types of outlines allow for a wide range of calculation possibilities. Aggregation must be defined in a calculation script, written in a proprietary Essbase language.

**Dense dimension:** A dimension that has a high probability of its members being used by the other dimensions (think Accounts or Periods).

**Sparse dimension:** A dimension that many members, but not all members will be utilized in a given data set (think Cost Centers or Customers).

**Enterprise Performance Management Architect (EPMA):** The Hyperion administrator’s tool that allows you to create, modify, delete and administer metadata and data across the different Hyperion platforms (Essbase, Financial Management and Planning).

**Essbase Administration Services (EAS):** The Essbase administrator’s tool that allows you to create, modify, delete and administer metadata and data in the Essbase cubes.

I have been around the Essbase world for almost 10 years now. I came aboard the same time as Aggregate Storage Outline (ASO) cubes did with Hyperion, so I feel we have been peers, as in high school. We are the same age (in professional years), hang around the same people (user conferences), go to the same places (servers), have the same friends (BSO cubes and relational DBs), have had a few classes together (cube migrations), but have never actually been friends (worked on an intricate cube). I have been on projects that have used tiny databases to track current balances. These have taken a day or so to organize and maybe another day for data loading via lock and send from users. Four years ago I was tasked by my employer to move a client’s Block Storage Outline (BSO) cube to ASO. I thought it was easy enough – use the migration tool and be done. It was that project in which I learned about a scripting language called MDX...you know, the one that ASO formulas require? Calm, cool and collected by day, studious MDX student by night was how I got through that project. I thought I had cracked the ASO code – run the Essbase migration tool (if possible) then learn the necessary MDX. Voila. ASO cube done and ready to go for the masses.

Oh, I was so naïve. I had read that people encountered performance issues and there were numerous questions regarding ASO cube optimization. I could not figure out what all the fuss was about. Until I had my own struggle with data retrievals from an ASO cube.

Here is my ASO-coming-of-age story.

I was tasked with building a cube that was fairly simple in nature; gigantic, but simple. The cube contained a handful of calculations that were very straightforward and routine. As one of the dimensions was very large, I was obviously concerned about aggregation times on this BSO cube. Below are the details so you can appreciate the circumstances I was working under.

**Largest dimension:** 43,820 members, 41,734 stored

**Potential number of blocks:** 4,667,655,720,653,291,500

I was encouraged after loading the first fiscal quarter of domestic data when the aggregation time was just 12 minutes. Score – within the time limit allowed. Then we added a fiscal quarter of international data. I started the script and figured it would run for around an hour. Three and a half hours later I killed it. That was not going to work considering we were going to need 2 years of historical data in the cube before go-live. Oh crud. My thoughts immediately went to building an ASO cube. I knew this would be great for end-user reporting, but the calculations that needed to be performed really had to be done in a BSO cube. I lived in an Enterprise Performance Management Architect (EPMA) environment with all shared dimensions, so deploying the ASO version of the application was a piece of cake. However, when I loaded my data into the ASO cube, retrieval time was just dismal; time for plan B. I copied the BSO version of the cube as a new cube in Essbase Administration Services (EAS) then had the system migrate the application for me. Next, I did a cube transformation from classic to EPMA within Workspace to have Essbase determine the hierarchy’s settings for data aggregation and calculation. "I have skirted the learning
Basics of ASO Optimizations

Upon realizing that my history with ASO cubes was, for all intents and purposes, puny, I set out to try and be an expert. However, I learned that the more I read, the less I knew. One of the first articles I read talked about the number of bits needed by a dimension (I seemed to want to start at the absolute basics instead of the functional basics): \( \log(x) \) where \( x \) is the number of children under a parent. I decided to take a couple dimensions and figure out their size. One of my most simple dimensions had 2 parents and each parent had 22 children. In my head, I thought, “Okay, so dimension A needs \( \log(22) \) plus the parent’s bits. That is 4.4 bits plus the momma, so let’s round up to 5 bits just for the kiddos...” I was very happy that my small unit testing on the members was working perfectly. The next step was to load all the data available from the BSO cube that had already been calculated by the calc scripts. My member formulas still worked, but my data retrievals in Smart View did not. Well, they may have if I increased my settings to allow for 3-day timeouts. I was not going into panic mode, but I was going into serious research mode on ASO optimizations and luckily had an amazing ASO contact to walk me through the process. I wish we all could be so lucky all of the time!

In my research, I found that if I chose some base level members (namely currencies) that contained member formulas but pulled higher level members of purely aggregated dimensions, retrieval time was fairly quick. Not lightning fast, but quick enough to pass user acceptance. With this knowledge, we decided to change all currencies members to be calculated members in the BSO cube to take this load off the ASO cube’s retrieval calculations. Once implemented, the retrieval time was drastically improved. I mean, hey, users were able to retrieve data from the cube! It worked so well that we decided to change all of the member formulas to be calculated in the BSO cube. It is important to note here that we were able to make the BSO calc scripts run very quickly, mainly due to the fact that we were not aggregating any values in the BSO cube but only running calculations on level 0 members.

Although the data was actually showing in Smart View now, there were still lag issues for data retrievals and zoom-ins. This is what made me do a deep, deep, deeeeeeep dive into the inner workings of ASO and where I realized that the logarithmic function I learned in high school would suddenly reappear as a necessary evil in my professional life.

BSO to ASO Migration Process

As I expected after migrating the cube to ASO, I had to change the member formulas to MDX. If there was one saving grace in this process, it was that the formulas that I had to change were very straightforward, so no having to learn advanced MDX. My basic skillz were enough (I have learned that whenever someone lists their skills with a “z” instead of an “s”, it means they have allowed plenty of room for exaggeration on said skills). I was very happy that my small unit testing on the members was working perfectly. The next step was to load all the data available from the BSO cube that had already been calculated by the calc scripts. My member formulas still worked, but my data retrievals in Smart View did not. Well, they may have if I increased my settings to allow for 3-day timeouts. I was not going into panic mode, but I was going into serious research mode on ASO optimizations and luckily had an amazing ASO contact to walk me through the process. I wish we all could be so lucky all of the time!

In my research, I found that if I chose some base level members (namely currencies) that contained member formulas but pulled higher level members of purely aggregated dimensions, retrieval time was fairly quick. Not lightning fast, but quick enough to pass user acceptance. With this knowledge, we decided to change all currencies members to be calculated members in the BSO cube to take this load off the ASO cube’s retrieval calculations. Once implemented, the retrieval time was drastically improved. I mean, hey, users were able to retrieve data from the cube! It worked so well that we decided to change all of the member formulas to be calculated in the BSO cube. It is important to note here that we were able to make the BSO calc scripts run very quickly, mainly due to the fact that we were not aggregating any values in the BSO cube but only running calculations on level 0 members.

Although the data was actually showing in Smart View now, there were still lag issues for data retrievals and zoom-ins. This is what made me do a deep, deep, deeeeeeep dive into the inner workings of ASO and where I realized that the logarithmic function I learned in high school would suddenly reappear as a necessary evil in my professional life.

Dynamic hierarchy consolidation options:

Let’s back up to the different types of hierarchies offered by an ASO database: stored and dynamic. To throw in some further confusion, we can also add Multiple Hierarchies Enabled. But, we will save that treat for later... In a stored hierarchy, the values of the outline are aggregated based upon the outline’s structure. In a dynamic hierarchy, the values are calculated and the parental values are determined by solve order. Also, dynamic hierarchies are required when you are using shared members, member formulas, need to use something other than the Addition, Ignore or Never operators, or are using time balance items.3 Looking back at my original ASO cube, I had dynamic hierarchies all over the place. (I had mistaken Stored with the BSO meaning of stored meaning all parent levels are stored in a pag file. Not the case in ASO even though it has the same name.) Given that you can put any Essbase consolidator on a dynamic hierarchy but only Addition, Ignore or Never on a stored dimension, I needed to verify that my cube really was just an aggregation cube and not doing anything more so I could make as many dimensions as possible Stored.

Dynamic hierarchy consolidation options:

Stored hierarchy consolidation options:
I had 4 dimensions that had alternate hierarchies and 2 of them are the largest ones in the cube. I knew that they must be dealt with...but how? Going back to the DBAG, I saw that the stored members must be listed in the primary hierarchy and the shared members in the alternates. Standard Essbase fare. Also, the dimension must be labeled as “Multiple Hierarchies Enabled”. Makes sense. But what does that mean? It just means there is more than one hierarchy listed. They can all be stored or a combination of stored and dynamic. Up to you. But which one was going to work for me? This is where my logarithmic function friend comes into play. Except I didn’t do the math, I just trusted someone on the topic who has done the math (literally). According to Dan Pressman, the bitmaps for alternate hierarchies essentially match the bitmap of the primary hierarchy so adding additional hierarchies is like adding one more grain of sand to a sand clock – not a big deal. I love his quote, “Alternate hierarchies, whether Dynamic or Stored or Attribute, are almost always cheap...give the user what they want.” 2 In my notes next to his writing, I wrote, “because there is little cost in terms of impact on the length of the bitmap and, therefore, performance.” So how do I optimize a dimension with alternate hierarchies? Optimize the primary hierarchy. This will always be set to Stored. It is important to note that you cannot add a shared member to this hierarchy as a Stored dimension only aggregates, it does not calculate, the data. If there were a shared member in the primary hierarchy, that portion of the data would be counted twice. But have no fears, Essbase will correct you if you try to outsmart it during validation time!

So what about my attribute dimension? It’s not big, but it is important to my cube’s design. What kind of impact does that have on my cube’s optimization? Luckily for us, in ASO applications (coming straight from the DBAG!) all attribute dimensions are stored dimensions. Woohoo! This means that my attribute dimension has as little overhead as possible for data retrieval.

As I was going through all the notes, blogs, books and guides online, I kept coming across the term “compression dimension”. Was that going to be important for me or for the infrastructure guru? The more research I did the more confused and clear-headed I got. Most of what I read stated that it was the dimension that was tagged as the “Accounts” dimension. However, (comma), in EPMA you cannot tag a dimension as Accounts for an ASO database. Hmm. That one is still perplexing me. So if I didn’t have a dimension tagged as Accounts, what was I to do? Find out what the compression does. The compression on a dimension essentially groups the data together for that dimension. In the remaining dimensions, the data is just stored. Because the data is grouped, it is best to set the compression dimension on the dimension that is a single, dynamic hierarchy. Most of the time, it will be on either the Account dimension or Time dimension. I played around with these settings and tried the Period dimension but gained nothing (mainly because of my member formulas). Since I don’t have a dimension tagged as Accounts, I can’t go after that dimension. Why not let Essbase do its thing and select the compression dimension on its own? Well, what do you know? My Account dimension got tagged as the compression dimension as it is a single, dynamic hierarchy.

Putting it all together

So, let’s remind ourselves that it originally took so long to retrieve data from my ASO cube that the user would get a timeout error in Smart View before they would ever see any data. I have gone through and changed the following:

1. Changed the member formulas to be calculated in the BSO database and just stored in the ASO database where available
2. Changed as many dimension types as possible to Stored versus Dynamic
3. Left all the alternate and attribute hierarchies in place – they have very little overhead
4. Have the compression dimension set to a single, dynamic hierarchy

By doing these 4 things, I was able to take the retrieve time down to milliseconds. The user has no clue what is being done from the time they click “Ad hoc analysis...” to seeing numbers on their Excel spreadsheet. I even had a coworker comment that a report created in Financial Reporting Studio used to take 17 seconds to run, is now taking under a second. That is great news! Hopefully you can take my journey to understanding ASO cube optimization a step further in your own environment. After all, iron sharpens iron!

Special thanks to Cindy Eichner for her review of this article for technical accuracy.


Sarah Craynon Zumbrum is a financial IT consultant with Finit Solutions, headquartered in Media, PA. She has previously worked in-house as a business intelligence architect and as an EPM consultant, specializing in the Hyperion suite of tools. She is the Secretary of ODTUG’s Hyperion SIG and organized the Raleigh Oracle Meetup group. She has been a speaker at ODTUG’s Kscope, OAKUG’s Connection Point, ECOUG’s fall conference and RMIOUG’s Training Days. She is a graduate of ODTUG’s Leadership Program and strives to implement user group leadership lessons learned from her time in the program. She resides with her husband in Raleigh, NC. When not doing geeky things, you can find her running, cycling or finding some excuse to travel.
Thank you so much for all your hard work and your willingness to help out wherever there was a need. Your incredible effort was greatly appreciated by everyone at the conference. We couldn’t ask for a more dedicated group of volunteers and words can’t express how grateful we are for the great feat that is accomplished each year by our volunteers. It was a pleasure and a privilege to work with all of you at RMOUG Training Days 2014, and we look forward to seeing you again in future.

*With Gratitude From The Training Days 2014 Conference Committee*

Kellyn Pot’vin - Training Days Director
Tim Gorman - Abstract Lead • Vince Giasolli & Bob Mason - Volunteer Lead
Mark James & Sruthi Kumar Annamnidu - Attendee Lead • Chris Ostrowski - Vendor Lead
Tim Gorman & Pat Van Buskirk - Speaker Ride Lead

**Abstract Reviewers**
- Matt Adams
- Debra Addeo
- Sruthi Annamnidu
- Gustavo Antunez
- Faud Arshad
- Galo Balda
- Vlado Barun
- Gary Brewington
- Jeff Bugbee
- Chris Chase
- Karthik Chigururu
- Michael Coll-Barth
- Dominic Delmolino
- Dimitar Draginov
- Jim Egan
- Kevin Garlow
- Rick Hata
- Frits Hoogland
- Ann Horton
- Dan Hotka
- Syed Jaffar Hussain
- John Jeunnette
- Steve Karam
- Andrew Kerber
- Peter Koletzke
- David Kurtz
- Niall Litchfield
- Scott Lockhart
- Kumar Madduri
- Osama Mustafa
- Arup Nanda
- Leighton Nelson
- Greg Opie
- Chris Ostrowski
- Jayant Patil
- John Peterson
- Therese Pickard
- Kellyn Pot’vin
- Bert Scalzo
- Carlos Sierra
- Jeff Smith
- Jared Still
- Bill Thater
- Elaine Thomas-Magnuson
- Liping Woods
- Sarah Zumburm

**Airport Shuttle Drivers**
- Lena Cohen
- Jim Egan
- V Giri
- Heidi Kuhn
- Dan Hotka
- Tim Gorman

**Room Ambassadors**
- Jeff Bugbee
- John Pack
- Venkata Giri
- Sarah Zumbrum
- Greg Hill
- George Linehan
- Bryan Wells
- Derek Reiner
- Jim Egan
- Wayne Stetson
- Greg Howe
- Jayant Patil
- Marc Hoban
- Andrew Smith
- Gary Reedy

**Registration Ambassadors**
- Joe Boncha, DBA
- Jason Looney
- David Fitzjarrell
- Caitlyn Tjarks
- Michael McMannis
- Phil Clark
- Ray Mariano
- Elaine Thomas-Magnuson
- Liping Woods
- Sarah Zumburm
I think most of you know who I am or have sat in on my presentations at RMOUG or one of the many other events I speak at. It’s always good to see everyone. I really enjoy your state and my continued support of RMOUG over the years.

This year I was elected onto the Board. I mostly serve in an advisory role and help Pat with newsletter articles.

I have enjoyed the speaking roles in my career. I suppose this is why I also enjoy classroom teaching…in/around Oracle topics. I have people that will sit in on my topics because I enjoy speaking. I do the same with the likes of Tim Gorman, Tom Kyte, Maria Colgan, and Jonathan Lewis...just to name a few.

Many of my fellow speakers have moved on in their careers and no longer speak at events like RMOUG. One such speaker I always enjoyed chatting with that sticks out is Dave Ensor who worked for BMC. His topics were always quite technical and internal in nature. He once did a topic on the people he has met along the way and I thought to myself...I could do that. I came up with '25 Years of Taking Care of Business'. Gotta update that to 30+ years!

Anyway...maybe RMOUG will let me do it sometime over lunch...it’s a conglomeration of the characters I’ve met thru the years. Some of my speaker friends have retired (anyone remember Marlene Theriault-Cahill...she is enjoying her retirement mostly on cruise ships)...if any of you have seen me lately...I, too, am closing in on a retirement age. I have people that can tell me when I started speaking, at which event, and what the topic of my lecture was! I’m always pleased to hear that what I’ve had to say actually helped someone in his or her job.

To me, sharing my knowledge is what this is all about.

I enjoyed Tom Kyte’s keynote at RMOUG. He did a bit on the history of Oracle, commenting on various hardware platforms and disk drive configurations along the way. He spoke of the old disk packs. I do this thing for elementary school kids where I show them some stuff from yesteryear of computing. I have an old 9-track tape. Sometimes I can date people by showing them the write lock ring in the back. This is where diskettes got the idea by the way... I have one of these 11 platter disk packs Tom spoke of. In the day...it held 500K. Yes...11 platters...half a megabyte. It’s been both fun and amazing to watch the capacity of these computers just skyrocket. I have an 8" floppy...IBM System III used to boot from these. I have 5 ¼ and the ever-popular 3.5" diskettes. Remember the Zip drives? They were short lived until the read-write CD came out...that was short lived when memory sticks came out...these memory sticks were actually one of my predictions...a chip on a small portable card that would act like a diskette drive. I also had a thought that a PDA (personal data assistant...I still use a Sharp Wizard!) would be a good place to put in a phone...and that phones would be the Jetson cartoon visual and audio phone...we are not quite there with the phone video but soon (kinda with Skype...kinda).

I come from the day before the PC so I remember the early attempts at diskless workstations running CP/M (and the CP/M computers...I think I have a CP/M reference card!) and when DOS arrived. I remember the Kpros, the 80286 side car (early PC's did a lot with hardware cards that is now handled by software) as it had more slots for more cards. The Compaq that was a luggable...only had a few slots for cards...the keyboard would fold out of the bottom and expose a small 5" green screen and diskette drives. I remember the first PC hard drive was 10MB. We all thought “what are we going to do with all that space???” Seriously. The first PC I bought was one I used all the time when I was at Honeywell...a Zenith 80286 laptop. Had this massive battery on it but I liked the touch and feel of the keyboard. The guy who wrote the ‘Automenu’ program made a fortune. This was early shareware, then when he was onto something, he sold it for a small fee. It gave the DOS computers an easy menu interface. I started working with Oracle RDBMS with version 4.

I still have the Oracle4 PC doc set and 6 low-density 5 ¼" diskettes Oracle sold
RMOUG Board Focus

for $600 I think...UFI became SQL*Plus, and INPIAP became Oracle Forms...and that RPT report writer was a real love-hate product. It lasted thru version 5 and I don’t remember it in version 6. I loved it but knew plenty who hated it! I remember you used to be able to read the Oracle RDBMS in from a tape that was pretty much ready to go...you had to start it up and add all the tablespaces including running the catalog script (a script that still exists today...we just don’t need to run it...). The installer didn’t come along till version 6...a messy version as versions went. I made a ton of $$$ (as did many of us) helping to migrate Oracle5 to Oracle6 because Oracle had changed the internal date format so Export/Import (exp/imp) did not work well for a migration aid. Probably one of my favorite Oracle releases was Oracle 7.3. It was stable, had PL/SQL all over the place, had triggers, had the 10046 SQL trace, and had some dictionary views. I was and still am big on using SQL*Plus for SQL creating SQL.

I wish I had saved pictures of these computing environments.

I crossed 34 years of marriage! Still enjoying her. She works for the school system and travels with me on my assignments when she can. We have 3 adult children who are all gainfully employed, and my eldest had a child this past year...our first grand child.

I still enjoy tinkering on my VW's. Still have the 2...

I purchased this BMW Isetta last year. I’m finally getting to the project now. I hope to have it running yet this spring.

Life is good. I’ve enjoyed being a part of the Oracle community for about 30 years now. I’ve been a part of RMOUG since its early days and am pleased to be on the Board and help contribute directly to its success.

Dan Hotka
Oracle ACE Director
Meet Your Board

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

John Peterson
Vice President
E-mail: VicePresident@rmoug.org

Ron Bich
Secretary
E-mail: Secretary@rmoug.org

Thomas Green
Treasurer
E-mail: Treasurer@rmoug.org

Mark James
Bob Mason-Assistant
Member Relations Director
E-mail: MembershipDir@rmoug.org

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

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

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

Chris Ostrowski
Vendor Relations Director
E-mail: VendorsDir@rmoug.org

Dan Hotka
Education Director
E-mail: EducationDir@rmoug.org
RMOUG Board of Directors

Kellyn Pot’vin
Training Days Director
E-mail: TrainingDaysDir@rmoug.org

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

Vince Giasolli
Special Interest Groups/Meetup Director
E-mail: SigsDir@rmoug.org

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

Bob Mason
Mark James-Assistant Scholarships Director
E-mail: ScholarshipDir@rmoug.org

Kumar Annamnidu
Member At Large
E-mail: MembersAtLarge@rmoug.org

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

John Jeunnette
Board Member Emeritus
E-mail: BoardME@rmoug.org

Peggy King
Past President
E-mail: BoardME@rmoug.org
RMOUG Events Calendar

3/20/14  RMOUG  CIO/CTO Panel on IT Education”, an evening Q&A experts panel at the Regis DTC campus
3/20/14  COAUG  First Meeting, TBD LinkedIn Group URL: http://linkd.in/IhPetoj
4/7-11/14  Collaborate  2014 in Las Vegas, NV (http://collaborate14.ioug.org/)
4/14/14  RMOUG  SQL>Update Call for Articles & Photos
4/16/14  RMOUG  Board of Director’s Meeting - TBA
5/15/14  RMOUG  Spring Conference Location TBD
5/15/14  RMOUG  SQL>Update Article/Photo Deadlines
5/16/14  RMOUG  QEW (Quarterly Educational Workshop) at Oracle DTC campus (http://www.rmooug.org)
5/21/14  RMOUG  Board of Director’s Meeting - TBA
6/3/14  NYOUG  Summer General Meeting
6/15/14  RMOUG  SQL>Update Publication Date

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

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

<table>
<thead>
<tr>
<th></th>
<th>Breakfast</th>
<th>Discounted Ad Rate</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4 Page</td>
<td>$350.00</td>
<td>$175.00</td>
<td>$525.00</td>
</tr>
<tr>
<td>1/2 Page</td>
<td>$350.00</td>
<td>$312.50</td>
<td>$662.50</td>
</tr>
<tr>
<td>Full Page</td>
<td>$350.00</td>
<td>$500.00</td>
<td>$850.00</td>
</tr>
<tr>
<td>Inside Cover</td>
<td>$350.00</td>
<td>$625.00</td>
<td>$975.00</td>
</tr>
<tr>
<td>Back Cover</td>
<td>$350.00</td>
<td>$750.00</td>
<td>$1,100.00</td>
</tr>
</tbody>
</table>

Join Us May 16, 2014 at the Oracle Sun Campus!

More Details www.rmooug.org

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

Tell Us About Yourself

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

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

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

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

Please submit all material to NewsletterDir@rmooug.org
Reach A Targeted Oracle Audience • Advertise Now!

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

Contact NewsletterDir@rmoug.org

Quarterly Education Workshops

May 16, 2014

Join us for our next Quarterly Education Workshop in August for presentations at the Oracle Sun Campus in Broomfield. 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
Dear RMOUG Training Days Participant:

Thank you for making Training Days 2014 a great success! Each year, RMOUG prides itself on gathering top-of-the-line speakers and vendors for an enriching and educational event. This year was no exception!

**Conference Evaluation**
If you haven’t already done so, please go to this link, [www.technicalconferencesolutions.com/RMOUG_2014_evaluations.html](http://www.technicalconferencesolutions.com/RMOUG_2014_evaluations.html), to fill out the conference evaluation. We want your feedback and will take all comments into consideration when planning Training Days 2015. We strive to improve the conference year after year; help us make TD 2015 even better!

**Special Thanks**
We’d like to extend our sincerest gratitude to all the RMOUG volunteers and room ambassadors who made Training Days 2014 great. Thank you so much! We truly appreciate all your efforts and would not be successful without your valuable assistance.

**Schedule Builder**
To keep the learning momentum going, all the papers and slide presentations we have received are available through the schedule builder. To access the schedule builder, simply enter your first and last name and password you created when you registered for the conference. Click on the words "presentation or white paper" to download. If you have any difficulties logging in, please contact us at rmou@teamycc.com or 910.452.0006.

The RMOUG Training Days Committee

---

**Platinum Sponsor**

Arisant
Business Solutions through Oracle Technology

**Gold Sponsors**

CONFIO | solarwinds | DBAK | ORACLE

**Bronze Sponsor**

PURESTORAGE