Tiffany Dahlbert
Project Manager Or Business Analyst
Who Is More Important?

Kent Graziano
Introduction To Data Vault Modeling

Ernst Renner
Oracle Forms To ADF V11

Remembering Training Days 2011

Board Focus - Peggy King
Member Focus - Glenn Goodrum

Change Service Requested

Rocky Mountain Oracle Users Group
The **PL/SQL Challenge**

“Destined to Become World Famous”

Why? Just listen to what people are saying:

- "The quiz is fantastic – great learning opportunity – and now you can see past quizzes and their results even better. well done." - From Andre
- "Cool quiz to tickle the brain!" - From A.M.
- "The PL/SQL quiz is one of the highlights of my workday. Thank you for creating a professional culture for us PL/SQL developers!" - From Lauren
- "The PL/SQL Challenge has quickly become my caffeine companion in the mornings to help get the grey matter moving and the synapses firing. Even if I don't feel very intelligent after seeing some of the correct answers..." - From Matt G

But wait! There’s more!
- Win fabulous prizes!
- Become the envy of your peers!
- Did we mention the fabulous prizes?

Play today! Play tomorrow. Heck, if you travel fast enough you can play yesterday!

The Challenge is simple: you play the quiz - daily. We keep score. You get ranked. Every three months, the top-ranking players compete in a quarterly championship to award first, second and third prizes. But that’s not all! Every week and month, players win prizes based on the correctness of their answers and from participation raffles. The more you play, the greater your chance of winning.

So what are you waiting for?

It's time to take the PL/SQL Challenge!

To register and play, or for more information, visit www.plsqlchallenge.com.

www.plsqlchallenge.com
On the Cover:
Huron Peak (14,003’) taken in 2005 during the approach to the climb. Huron is located in the central Sawatch Range, west of the Mount Belford group, in the Collegiate Peaks Wilderness. It lies in Chaffee County, approximately 18 miles northwest of the town of Buena Vista.

Steve Hockett has been an Oracle DBA for 18 years, currently working for The City and County of Denver.
Congratulations to the 2011 Conference Committee as we mark our 20th Training Days Conference for continuing RMOUG’s tradition of providing an outstanding conference filled with excellent speakers who shared valuable information with our members. As with past conferences, a welcomed comment I kept hearing from attendees was: “There were too many good sessions on at the same time; it was hard to choose which one to go to.” Cheers to all for a job well done --- Thank you!!

A very special “Thank you” to all of our wonderful conference volunteers for without your help Training Days 2011 would not have been possible. Even the simplest gesture like offering to meet the speakers at the airport and give them a ride into the city for Training Days brings out comments like: “These rides in are so welcoming; it feels like visiting old friends.”

Volunteers are the backbone of most organizations and for RMOUG they are exceptional. From the Officers and Board of Directors to committee members, we all take time out of our busy schedules in support of RMOUG’s primary purpose: “To provide our members forums which enhance their technical knowledge.”

Our annual Training Days conference along with our other member activities like the Quarterly Educational Workshops (QEWs), Special Interest Groups (SIGs), Hands-On Lab Meetups at Regis (DBLabs), and Quarterly Newsletter (SQL>Update_RMOUG) provide the forums for exchanging information. RMOUG’s Board keeps the wheels of the organization moving forward by providing forums for exchanging knowledge and you as RMOUG volunteer members provide the grease by sharing your experiences and knowledge that help all of us have smoother ride. But the real key to success of RMOUG is our volunteers – “you” -- speakers, group leaders and authors, you come together to share your knowledge and experiences.

We need all of our members to come forward and volunteer to help grease RMOUG’s wheels so that we can have a much smoother ride into the future. Come and use our forums to express your thoughts, share your experiences, offer some guidance with case studies, and teach us with a presentation or article for the newsletter. RMOUG is your organization and can grow and thrive only if everyone helps. So take the first step and volunteer an hour or two to give a presentation or write an article or several hours to work on a committee or help with Training Days --- become an active member, become an RMOUG Volunteer!

As I sit back reflecting on my time volunteering for RMOUG, I can’t help but smile. It has been an amazing seven years that has just flown by. It seems like only yesterday I began planning for my first Training Days Conference in 2004 as the Director. I have learned so much from everyone and have truly enjoyed my experiences on the RMOUG Board, but as the saying goes “all good things must come to an end.” Today brings my announcement that I will be stepping down as President of RMOUG. John and I will be moving back to Arizona to be closer to my parents and his father. I am looking forward to seeing what the future holds for me and to watching RMOUG continue to flourish.

My fondest wish for all of you as members of RMOUG is that one day you find yourself smiling as you reflect back on your time volunteering.

Best Wishes,

Peggy King
2009/2010 President
**Be Sure To Attend The Spring Quarterly Education Workshop**  
**May 20, 2011**

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

If you or your company are interested in sponsoring breakfast for the workshop, please see page 11 for rates.

Sponsor the next QEW and receive an ad in SQL>Update at HALF the normal price.

*Please contact Carolyn Fryc cfryc@orsportal.com*

Check www.rmoug.org for the location, times and featured speakers

---

**RMOUG SIGs - New Group!**

We are excited to announce the launch of the RMOUG Fusion SIG (Special Interest Group). Please contact Jordan Braunstein at Jordan.Braunstein@visualintegrator.com for more info. We would like to aim for our first Fusion SIG event meeting the same day as the next RMOUG QEW, May 20, 2011.

What is Oracle Fusion? Some jokingly say its everything not in the Oracle database or ERP! In all seriousness, Fusion does cover a lot of product ground:

- Oracle SOA Suite and BPM Suite
- Oracle Identity Management
- Hyperion
- OBIEE and ODI
- webLogic
- Universal Content Management
- Webcenter
- APEX
- Java products

And Much more…for a complete list, visit:


And, if you’ve paid attention to Oracle’s Product Roadmap, Fusion Applications is the direction to bring all Oracle products together into a cohesive, integrated suite. So, another reason to join this group, is that you’re probably going to be involved in Fusion Applications, whether you like it or not, that is the direction of all the Oracle products.

*Please Contact Jordan.Braunstein@visualintegrator.com  
Today To Join This SIG*
**ORACLE FORMS TO ADF V11**

*by Ernst Renner, Robert Nocera & Christopher Coy*

**Vgo Software**

---

**Introduction**

This article reviews the effort of modernizing Oracle Forms to Oracle ADF v11. The reader should come away with a high level view of some common tasks in using ADF v11 in moving from Forms.

For the remainder of this article we will be discussing legacy Oracle Forms applications and how they can be brought forward (“modernized”) into an ADF architecture. This will cover a broad spectrum of topics, most of which warrant their own individual whitepapers or even books, in some cases.

The modernization of Oracle Forms, client/server Forms apps in particular, to ADF should cause the reader to think beyond their individual Forms application and consider their position within the enterprise. Moving to ADF opens another paradigm of development, with an increased capacity to include various architectural components. It is a chance to move “way out of the box” in your future development efforts.

This document is intended as a high-level, broad-brush, document to introduce the reader to some key concepts and is not intended for deep development knowledge or tips. Check into the blog [www.java-hair.com](http://www.java-hair.com) from Robert Nocera for questions. Mr. Nocera and Mr. Renner will be presenting at RMOUG ’11 on related topics.

---

**Forms Future**

Oracle Forms, as any technology, has been evolving and previous versions are being retired. To date the calendar can be viewed as:

- 6.0.8. sustaining support ended in January 2008
- 9.0.2 sustaining support ends in July 2008
- 9.0.4 extended support ends in December 2010
- 10gR2 extended support ends December 2011
- 10gR3 extended support ends December 2011

But a larger question looms: *will Oracle Forms go away all together?*

This is an interesting question. According to Oracle product development, Oracle Forms will be around forever. However, with the emergence of JDeveloper and ADF it is also equally possible to see that JDeveloper and ADF will take the place of Oracle Forms. In fact, most organizations are developing new applications in Java or ADF and very few are developing applications with Forms anymore.

Oracle, correctly, has extended support for its Forms products and this level of support is very important to its customers. The Oracle Forms development community is alive and strong, but there is a growing community of JDeveloper and ADF resources to support the “next wave” of Oracle development.

**Business Value and Positioning**

An interesting perspective is to analyze Oracle’s business strategy since 2000; heavy acquisition of complimentary solutions and technologies. Prior to 2000, Oracle had conducted acquisitions that would augment or defend its core technology namely based around the database. In recent years, massive acquisitions of Sun, BEA, Siebel, PeopleSoft, JD Edwards and Retek (a few among many others) have created a formidable business solution architecture that effectively competes with other global solution providers such as SAP and Microsoft.

With these acquisitions, one can see that:

A. They need to be integrated to be most effective to Oracle’s customers

B. Integrating the applications will also provide opportunity to offer more solutions (i.e. greater sales opportunities)

C. Efficient integration strategies to core products produce economies of scale, reducing the cost of integration and speeding up time-to-market

The maturation of technologies in the broader technology realm, namely Java, web services and SOA, also provided a great platform to offer integration of solutions and positioning for greater adoption and adaptation as all of these technologies continue to evolve. With Fusion as a platform for not only acquisitions, but as a product integration strategy, Oracle has done a splendid job of positioning itself as the premier application vendor.

**ADF Introduction**

For those new to ADF, it is a Java-based application framework with which you can build enterprise-class web applications. ADF is based on the JavaServer Faces framework and includes rich components for the development of feature-rich web applications. ADF applications can be developed using Oracle’s JDeveloper IDE. JDeveloper provides many wizards, drag-and-drop features and integration plug-ins for other components in the Fusion stack. JDeveloper makes it “easy” to develop ADF applications by providing these features, though it still requires a good degree of education and experience for a developer to be “expert” with these tools.

**Impact**

When considering adopting ADF into an organization, one must consider the learning curve of adopting new technologies. In this case, it is a significant change in thought for developers. Typically, Forms developers have developed sophisticated client/server code where re-use was somewhat limited to the use of libraries or “cut and paste” as a pattern. Code has been duplicated within Forms and within Forms applications. Some forward thinking organizations in the late 1990’s abstracted their databases by creating
Oracle Forms. Where are all of these Forms programmers going - to find people to support a decidedly “legacy” environment such as Forms developers. In five more years, it will be extremely difficult to find highly-talented Oracle Forms development resources. Even off-shore staffing houses are having a hard time retaining their developers to begin the transition to the web world.

From a resource perspective, it is becoming increasingly difficult to find highly-talented Oracle Forms development resources. Even off-shore staffing houses are having a hard time retaining Forms developers. In five more years, it will be extremely difficult to find people to support a decidedly “legacy” environment such as Oracle Forms. Where are all of these Forms programmers going - to learn ADF and JDeveloper or other IDE’s and languages.

Options – Modernize or Re-write

Considering where Oracle is going and the maturation of technologies such as ADF, existing Oracle Forms users have several options that will provide support or position strategically for increased business value. Though upgrading Forms itself is possible (and in fact relatively inexpensive and simple) and may fit your needs, this section focuses on using JDeveloper and ADF to modernize or rewrite a Forms application.

Modernize to ADF

Modernization to ADF is a powerful alternative to upgrading Forms. The “modernization” approach discussed here assumes the use of tools to assist in the conversion of the Oracle Forms application to a new destination (ADF). A good alternative is to partner with an experienced vendor who uses tools and solid processes to modernize Forms applications. This provides the consistency and efficiency of a tools-based approach, while also ensuring that everything is done to your specifications.

Oracle does not provide a conversion alternative as it does provide an upgrade solution. Tools can range from those that conduct valuable application portfolio assessment and discovery (such as Vgo’s ART product) to actual code generation. Whether buying a tool or using a vendor with a tool, it’s the output that matters; can you maintain the application and support it going forward?

Keys to a successful conversion are discussed below. There are several distinct Pro’s and Con’s for this option:

Pros:
- Re-use your business logic effectively
- Gain efficiencies in creating the application architecture for your application
- Create a true Java/JEE/ADF application
- Much faster than a re-write
- Cleanup and consolidate objects in Forms app’s
- Incorporate business process re-design
- Can support Forms/ADF co-existence points

Cons:
- Still manual re-work which may be considerable based on the structure/quality of the old Forms
- More expensive than an upgrade

Modernize to ADF When...

The main choice of modernization is centered on the vision of the overall architecture: positioning to adopt SOA, clear, industry-standard, integration of applications, or simply adherence to an evolving standard in the organization. As Oracle points out, there are some opportunities to integrate Forms into a larger context with SOA positioning (wrapping PL/SQL in Java). However, there are limitations when considering non-Oracle technologies.

When modernizing, a similar methodology of project execution can be implemented as re-writing, except that some aspects are “lighter” in process rigor, such as extracting, examining and applying “requirements” from existing legacy components. The method of modernization is a big undertaking and under-estimating the size, scope and complexity is common in almost all clients we have spoken to. That level of under-estimation is the leading cause for these efforts to fail. Having experience, or at least the proper expectation level, is a necessity in engaging these efforts.

Re-writing from Scratch

Re-writing a Forms application isn’t as simple as opening JDeveloper, mapping your business components and generating a new application. When re-writing from scratch, a customer should be contemplating their business process from scratch as well. That being the case, it entails all the rigor of new application development: collecting business requirements, determining functional flows, creating business rules, creating design specifications (for infrastructure and database as well as the application itself), testing and QA methodology and criteria, user acceptance criteria, GUI specifications and usability, testing at all levels, security implications, end user training, documentation, rollout and a myriad of detailed components mixed in there.

Pros:
- End result is exactly the application required
- Function and technology should be a match
- Architectural design is optimized based on current technology

Cons:
- Longest and most costly option
- High risk to business operations
- No re-use of any existing components is likely
- Low business tolerance; “Why are we doing this again when we already have a good system?”

Re-write When...

The most compelling option in completely re-writing an application isn’t based on technology; it’s based on business needs. Alone, business need doesn’t seem touched upon in the countless blog’s and papers of the technorati out there.
If your business has evolved in its processing or has major process changes in store that increase business value, you should re-write. If your business partner is happy conducting business the way it always has, the other options may suit schedule, direction and budget in a better way.

Even in a re-write there may not be a need to throw everything out and start over. It is unlikely that a new application will create entirely new rules if the process is still the same. For example, a Forms application that processes new applications for an insurance company is still going to process the same work, but perhaps in a different manner. Therefore, code segments that validate a policy id or ensure that the product chosen with the contract is a valid combination in a given state, are still likely to be valid business rules. We therefore recommend doing an assessment of the existing Forms application and extracting out rules that can be re-used in a new application. Vgo Software has a product called ART, an application portfolio assessment tool, that provides deep technical insight into what is in each Form. Additionally, ART will store this data into a repository which then can be mined and transported into the Fusion Business Rules repository.

Otherwise, open each Form and go through them manually to extract business rules and collect the rules (though this will be tedious and Forms developers may decide to jump ship).

Modernization / Rewrite: Common Implementation Steps

In this section, we will dive into a technical introduction to steps that are common to using ADF when re-writing an application. As stated, there are tools that help with some of these steps, but this isn’t a tools paper; it is a discussion of ADF and Forms.

This section will introduce the following ADF concepts: data model, AppModule, Task Flows, view objects and some UI components.

Forms to ADF Mapping

The chart below reflects some of the components listed in the previous section and can relate to either the single silo or to the entire enterprise application. The mappings and decisions may be the same, though broader and more intense in an enterprise setting. (See Table 1)

<table>
<thead>
<tr>
<th>Forms</th>
<th>ADF Mapping</th>
<th>Complexity</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forms</td>
<td>Task Flows (for Transaction Mgmt) Application Modules</td>
<td>High/Medium</td>
<td>Specific to app Complexity depends on Forms state</td>
</tr>
<tr>
<td>LOV’s</td>
<td>Read-only View Objects</td>
<td>High/Medium</td>
<td>Usually always Can implement/augment with Groovy expressions</td>
</tr>
<tr>
<td>Query-based blocks</td>
<td>View Objects</td>
<td>Low</td>
<td>Always applicable</td>
</tr>
<tr>
<td>Tables (called from DB in DML statements)</td>
<td>Entity Objects</td>
<td>Low</td>
<td>Always applicable</td>
</tr>
<tr>
<td>Transactions</td>
<td>Task Flows</td>
<td>High</td>
<td>Always, but need good mapping</td>
</tr>
<tr>
<td>Triggers</td>
<td>Java Method (programmatic) or Groovy Expression (declarative)</td>
<td>High/Medium</td>
<td></td>
</tr>
<tr>
<td>Windows &amp; Canvases</td>
<td>JSF pages, PanelGroups, Page Fragments</td>
<td>Medium/Low</td>
<td>Most of the time</td>
</tr>
<tr>
<td>Libraries</td>
<td>Application Modules</td>
<td>High/Medium</td>
<td>Depends on complexity</td>
</tr>
<tr>
<td>SQL</td>
<td>View Objects</td>
<td>Medium/Low</td>
<td>Requires good discovery and knowledge between a business rule or navigational rule.</td>
</tr>
</tbody>
</table>

Table 1: Forms to ADF mapping

What Doesn’t Map?

There are a number of items in Forms (especially in v6 and below) that do not “map” to the web application world in general or to ADF in particular. This is to mean that there is not a direct path to implement the same functionality in the web world. Examples of these are:

- HOST, ORA_FFI, User Exits, Java Importer
- Local client operations will change to operate on the Application Server such as READ_IMAGE_FILE, TEXT_IO, etc.
- Other points to consider: tabbed canvases, SYNCHRONIZE, TIMERS, mouse events (_mouse_over, mouse_enter, mouse_leave) and Icons (need to be converted to JPG or GIF) WinAPI Calls

There are also often business functions that have to be reconsidered based on the stateless environment that you are in when you create a web application. Addressing these challenges ahead of time will save a lot of time later in the project. Again, proactive awareness of what is really in your Forms application will help tremendously at development time.

The Data Model

An AppModule is essentially an aggregate of model layer Views and/or service methods needed to perform a task and any business logic that might be used. While creating model layer objects such as entities and views, one can add these objects to an AppModule, or they can be added later.

- Identify the tables and relationships in the Oracle Form.
- Editable entities
- Read only lists of values (LOVs)
- Use JDeveloper’s “New Business Components from Tables” wizard to create the initial entities and corresponding View objects. If you already have entities, it is simple to create the View objects later
- For each entity, take the time to assign any UI Hints – length of field, field type and labels
- Create View Links for the relationships as needed by the form

At this point, if the artifacts mentioned above were added to an AppModule during code generation, they will be visible in the Data Controls panel in JDeveloper. In master-detail relationships, View objects are referenced more than once. They are referenced first for their initial import into the AppModule and again separately for the relationship. When defining master-detail relationships several levels deep, it is usually a good idea to remove the duplicates and...
use only one View object reference. Doing so will result in a data
control with a single and simple hierarchy rather than any one
View object duplicated one or many times.

**AppModule Client Exposed Methods**

One can add methods to the AppModule that can be used by
task flows or web pages. To do so, create public methods and then
use the Java tab for the AppModule to expose your methods to the
UI. When methods have been made accessible, they will show up in
the data control as shown below:

**View Objects and Queries in the AppModule**

By using View objects properly from within an AppModule’s
java implementation, it is possible to completely avoid writing
standard SQL and connection logic. While most developers enjoy
that low level of control, using the tools provided by ADF does have
benefits. The first and foremost of these is the ability to re-factor
queries and View objects without having to change the java imple-
mentation at all.

A View object can be filtered simply by adjusting the where
clause:

```java
public void setFilter(ViewObjectImpl view) {
    view.setWhereClause("SOME_FIELD='Some value'");
    view.executeQuery();
}
```

Second, by using a View object’s createRow and insertRow
methods, a developer can take advantage of any Validators that
have been defined in the View object at design time. Also, writing
SQL statements can be avoided. Obviously, one would want to
add whatever error handling seemed appropriate. Error handling
should mirror messaging and format from the original Form.

```java
public void addToView(ViewObjectImpl view) {
    Row newRow = view.createRow();
    newRow.setAttribute("SomeField", "a test value");
    view.insertRow(newRow);
}
```

**ADF Task Flows**

One of the strong points of Oracle Forms, and client/server
applications in general, is a consistent method for persisting data
to the database. In web applications, transaction state and persis-
tency, especially when converting from a client/server application,
require extra design and attention. As a destination framework,
ADF has the concept of a Task Flow, which aids in this transition.

ADF Task flows are similar to Spring MVC Web Flows or
BPEL diagrams. They define entry points and actions that can be
taken, routes to be traversed by logic or by user decisions.

Referring back to the concept of the current row in a View
iterator, a CreateInsert action would result in a new row becoming
the current row. Therefore, when the next step in a task flow is an
Edit page for that particular table, the row currently being edited is
the View object’s current row – the same row that we’d get if we
obtained a reference to the iterator’s current row from within the
AppModule.

One can also easily obtain values or rows for whatever reason
that business logic may dictate. The example below gets a value
from the current row of a view:

```java
private String getCurrentDescription() {
    Row row = this.getXxxxMasterView1().getCurrentRow();
    if(row==null) return null;
    return (String) row.getAttribute(“Description”);
}
```

**Bounded Task Flows**

A bounded task flow represents a single transaction. A Commit
or Rollback action from the data control (the data control is the
visible component created by the definition of the AppModule,) executes Commit or Rollback for any changes made during the task
flow. This is important because saving one screen will also commit
changes on other screens the user may have forgotten about, if
those screens are part of the same task flow. That can be good or
bad, depending on whether the developer keeps that in mind when
creating the screens and deciding what actions a user can take from
various screens.

Existing Forms lend themselves pretty well to mapping to
bounded task flows in ADF 11g. Use the “Use Existing Transaction
If Possible” selection so that if you nest your task flows the transac-
tion will be handled by the parent.

Each Form will normally encapsulate a particular process in a
system, if the original application was designed correctly the map-
ning will be very close to 1 form to 1 task flow. If the original Forms
application is too complex and contains too many screens it would
make sense to break it into multiple task flows.

At certain points in a task flow, the user is usually presented
with a screen, or view. It isn't necessary to create the correspond-
ing web pages right away - one can do that after designing the task
flow. Double clicking on a View (this is not to be confused with the
data model's View object,) will result in a wizard to help in the
creation of a web page.
In this example, there is actually only one web page – the “entryPoint.”

The following nodes were dragged from the Data Control:
- Commit
- Rollback
- CreateInsert
- Delete

The other nodes,
- showSavedMessage
- setCancelMessage
- clearMessage

set values in a bean which the UI has access to. Upon execution of any of the data operations, a new message is prepared for the user to inform them of how the data operation went. The web page is then refreshed. Since the data editing page has a field bound to the message property of the shared bean, the message is automatically shown.

**UI Access to the Data Model**

When designing the screens, it’s usually straightforward to drag Model objects from the Data controls panel onto the page. Methods defined in the AppModule (if exposed to clients,) can also be used directly.

For example, one might drag a view of products from the data control to the page in design mode. ADF will then prompt the developer with a list of possible ways to render the data. One commonly chosen is an ADF Table. The ADF Table is an editable grid of data. The column names and data types are those previously defined in the corresponding Model layer’s entity or view object. By also dragging the CreateInsert action from that particular model object’s Operations’ folder, an “Add” button could also be added. When the user clicks the button, the grid should refresh without reloading the entire page. ADF makes this kind of partial page rendering simple with the Partial Triggers property. By selecting this property of the ADF Table, the developer can click “Edit” and be presented with a list of page components that should trigger a reload of the ADF Table.

**Page Layout**

ADF page rendering generally looks pretty decent by default. Still, complex component placement can be a challenge. For example, while designing a simple data entry form is fairly simple, designing one in the manner of Blueprint CSS or some other grid layout can be quite challenging. To that end, it is possible to extend ADF’s UI Components and provide renderers that allow the developer to control exactly what DHTML is generated.

That kind of customization isn’t for everyone or every project, but at least it is possible when the users simply must have a certain layout. This becomes an important feature in Forms conversion, as 75% of Forms users want the same “look and feel” of the original application in order to avoid re-training efforts.

Otherwise, there are some fairly powerful layout components to choose from. The Panel Form component is good for keep labels and fields aligned on a page. The Panel Group component allows for horizontal or vertical positioning of its child components, and the Panel Stretch component is definitely a welcome addition from the

ADF page design is a wide enough subject that many articles and blogs are devoted to nothing that one subject alone. It makes sense to become familiar with what each of these components is suited for and what they’re not.

**Table-based Search/Edit**

For the simplest type of screen, typically used to maintain administration data such as lists of countries or product types, the UI pattern we use in an ADF application is very similar to the pattern used in the Form application. The main difference is that there is a separate component used to perform the search.

If you are familiar with the ADF 11g framework you know that the simplest way to provide the type of CRUD capabilities in these types of forms is to create an Entity based on the table being edited. A View Object would then be created for that Entity and exposed to the user in the page. To provide the search capability to the user, the developer would create a set of View Criteria. This View Criteria is then used to create an ADF Query Panel that the user of the application will use to search the data.

In order to take advantage of the benefits of the ADF Query Panel, you do need to include it on the page and this in itself is a difference from a typical Form application that in some respects influences the Search and Edit patterns that we use.

In order to alleviate any unnecessary button clicks, the ADF Panel Query is provided fully disclosed. The user can then search and see the results on the same screen. Edits to the returned data can be performed in the table itself and records can be deleted from the returned results. When the user wants to add a record, they simply click the add button which will insert an empty row into the table and then the user can enter data directly into that row.

The task flow for this type of search and edit functionality explicitly shows the messages being set and cleared to indicate to the user that an action has been successfully performed. See Figure 2 for reference.
Users of the original application normally react well to a change such as this. Though the UI has changed from the original form, it provides them some additional functionality that they did not have before via the ADF Query Panel. If you were to use Oracle Meta-data repository, the users could even save their favorite queries using this panel.

**LOVs vs. Dropdowns**

Forms applications use a lot of Lists of Values. Newer applications have a mix of LOVs and dropdowns. A lot of times, when an application used a list of values, after the user selected a value from the list, a description field would be populated.

ADF 11g provides an easy way to produce this same behavior, but you can also just as easily display the description instead of the key when a value is selected, or show the key and the description when a value is selected. The ADF 11g LOV components also provide some more enhanced search functionality.

For short lists of values, consider replacing with single choice selects. For longer lists use popup LOVs to postpone the initial query that retrieves the list until it is actually needed.

In the case of static LOVs, replace them with entity based LOVs where feasible. This allows you to cache the list and have it be able to change without changing the application.

**Post-Query vs. View**

Post Query Triggers in Oracle Forms applications can be used for many things and depending on what function that trigger was performing in the Form, implementing the same functionality in ADF 11g can be quite similar or very different.

In the previous section, we mentioned LOVs with description fields that are populated based on the key from the LOV. Those description fields are normally populated in a Post Query Trigger on the Form. In fact, Oracle Forms allows you to make an awful lot of database calls in a very short period of time that have minimal impact on the performance of the application. This makes, for example, populating a description field via Post Query Trigger an acceptable option.

This option is not so acceptable for applications built in other frameworks, including ADF. Instead, in ADF, when you create a LOV and add a transient description field from another Entity, as any number of blog posts and development guides will tell you to do, ADF will create a join in the query for the View that contains the description field so that the field will be populated at the same time the rest of the data is retrieved from the database. This eliminates the need for a separate call and even a separate call per row of results returned in some cases.

This is a very common and simple example but at the same time it illuminates a point that should be recognized. The performance of your application is going to be greatly affected by how many database calls are made. Watch out for situations where you might be executing a query per row of a result set and try to reduce that access as much as possible.

**Keystrokes and Menu Items vs. Buttons**

In a lot of older Forms applications the users are used to a lot of keystroke commands to execute certain functions. You can almost mimic some of this behavior with shortcuts on buttons but no matter what you do, the functionality is almost certainly going to be replaced by buttons.

Forms is also very good at keeping track of context, knowing where the cursor is in the UI and executing the right event depend-

While this can be achieved in ADF 11g, it makes the application more complex, most of the time unnecessarily complex, and it also has the effect of not being intuitive for new users. Instead, put your detail tables inside a panel header group and add a toolbar with add/delete buttons to the header's toolbar facet. This puts more buttons on the page, but it makes their use very intuitive.

**Closing**

The main point of this article is to consider a broader application context than simply a Form. Consider the technical architecture, the business needs and the broader enterprise direction in any move you make with Forms. As Oracle has done in considering its business direction (e.g. acquiring solution-based companies), consider your business drivers. Will your company be acquiring other businesses? Will you be getting rid of legacy applications for commercial ones? Will you be developing new applications using different technologies? If the answer to any of these is “yes”, then the adoption of ADF is also a building block to being able to adapt to this changing business landscape.

Look for updates or specific items of interest on our blog at www.java-hair.com or additional papers on www.vgosoftware.com. Our co-founder and CTO, Robert Nocera, provides information on ADF, Java in general and other topics, not all technology related.

The authors of this document are Ernst Renner, CEO of Vgo Software and NEOS, Robert Nocera, CTO of Vgo, and Christopher Coy, Senior Architect at Vgo. Mr. Renner has worked with Oracle technologies since 1992, working together with Mr. Nocera developing v3 Oracle Form's applications and now working together in delivering enterprise applications all over the world. Mr. Coy has over 15 years of development and architecture experience in many different languages and different platforms.

Vgo Software, a division of NEOS Holdings, LLC, is based in Hartford, Connecticut, USA and is a modernization company with deep experience in Oracle Forms migration and conversion. Additionally, Vgo is an Oracle Safe Switch partner, helping clients on Sybase, Informix, UDB/DB2 and Microsoft, adopt Oracle technologies. Visit www.vgosoftware.com for more information.
Who Is More Important - the Project Manager or the Business Analyst?

by Tiffany Dahlbert, PMP

To survive in today’s economy, companies must maximize their time and money while they work with limited resources. Organizations that haven’t invested in their human resources are struggling because they used on-the-job learning as a cheap alternative to formal training. Like most things, “you get what you pay for.” We value education because, in most roles, knowledge is essential to increase efficiency and effectiveness. People learn by doing, but experience will only take you so far; education is what helps us get to higher levels of performance.

This is especially true in the professions of Project Management and Business Analysis. Most people have learned their jobs through observation, trial, and error. Over years, we gain hard-earned experience and some degree of expertise. Our success fuels our feelings of accomplishment and proficiency. However, without formal learning, the danger is that “we don’t know what we don’t know”.

As an employee, I performed the role of a Business Analyst for ten years before I even knew organizations had a title for it. Similarly, I did Project Management for ten years before my previous employer brought in formal project management training. In both cases, once I completed classes in these professions, I was in awe over how much more successful I could have been. If only I had learned these formal tools and techniques earlier, they could have made my job so much easier.

And, if I’d known there was a better, faster, and cheaper way to do my job, I could have made even a larger contribution to my employers. Knowledge always brings awareness, and sometimes it also gives you power. Those of us who choose to teach, do so to contribute to the greater good by spreading knowledge to others. Once I learned how to do Project Management and Business Analysis in more meaningful and impactful ways, I wanted to share my experience and knowledge with others.

At Ready2ACT, we teach Project Management and Business Analysis so you can improve your business.

The intention of this article is to educate you about the roles and professions of Project Management and Business Analysis, define how these roles interact to propel organizational growth, and share how companies structure these jobs and their corresponding salaries.

Why is Project Management so important?

Wise organizations understand that to grow and improve, they must be very good at implementing change; without change, there is no growth. However, companies cannot afford to lose time, money, and energy associated with up to 78% of projects failing these days. The most prevalent problem in project failure is lack of proper planning. For example, for every hour that you spend planning, you’ll save 20 – 200 hours during project execution. Improving your planning process has a greater return on investment than any other organizational initiative.

Yet, planning is an under-emphasized process because while we can learn how to execute through experience (for most of us, the bulk of our jobs is the “doing”), planning takes education in models, tools, and techniques. Most people simply don’t get training on how to plan. Management tends to place more value on “doing” because they can “see” you doing something.

Planning doesn’t have the visibility that executing has, yet planning is what makes doing the work go more smoothly and more quickly if done well. Gambro is a global medical technology company that manufactures life-saving and therapeutic equipment for kidney and liver dialysis. Their products literally have life-or-death consequences. Not surprisingly, they have a saying there: “We go slow to go fast.” Ingrained in their culture is the belief that by investing the time up front on due diligence through planning, they get it right the first time so they can execute more efficiently.

Many companies have created their own methodologies for getting projects done. However, there is only one international organization that compiles best practices world-wide: The Project Management Institute (PMI). This entity solicits the input of experts around the globe to produce the Project Management Body of Knowledge (PMBOK). This book contains the processes, tools, and techniques related to nine areas of knowledge (Integration, Time, Scope, Cost, Quality, Risk, Communication, Human Resources, and Procurement). PMI also has several certifications in Project Management that are recognized internationally.

The most popular certification is the PMP® (Project Management Professional). To be eligible to apply for the exam to be certified, Project Managers must have 4,500 or 7,500 hours of experience, 35 hours of project management education, and agree to abide by the PMI code of conduct. In fact, the U.S. Government, who recognizes the value of this certification, now requires someone with a PMP® on staff for organizations to be eligible to bid on some federal contracts. By employing PMPs, this smart move by the government ensures common language, common tools, and strong ethics are used by contractors on government projects.

This credentialed requirement by the U.S. Government propelled the Project Management profession into rock star status. People who earn their PMP® on average make $100,000 more within 10 years than those without certification. This salary difference proves the value that education has in the real-world. PMPs are recognized around the world. So although there may be language,
geographical, and cultural differences to overcome, at least everyone is using these same terms, definitions, tools, and techniques when they work together on projects. And even with local organizations, this key element with proper planning is what saves project teams’ time and money.

There are two ways to get educated in world-wide best practices: read the PMBOK and take a class. The benefit to reading the PMBOK is that it’s a cheap way to gain the knowledge. However, reading it alone will not tell you “how” to use the tools and techniques. The PMBOK only tells you “what” is best practice—not how to do it. This is purposeful because PMI does not claim to be an expert in your business and how you run it. Therefore, PMI steers clear of methodology. Instead, they give you information and allow each organization make their own educated decisions about “how” to implement the knowledge. If you want an easy way to know how to apply the tools and techniques to your own projects, Ready2ACT provides classes by a qualified PMP® who will serve as your trainer as well as a project management consultant.

**What is Business Analysis, anyway?**

Business Analysis is a profession designated by the IIBA (International Institute of Business Analysis). Similar to PMI, the IIBA sets the global standard practices and now has certification in Business Analysis. These generally-accepted practices are contained in the BABOK (Business Analysis Body of Knowledge) which has six knowledge areas (vs. PMBOK’s nine). Because this organization was established internationally in 2006, it does not yet have the recognition that PMI has; however, it has 90 chapters worldwide and endeavors to follow in the footsteps of PMI’s success in buying the Business Analysis profession. Their CBAP® (Certified Business Analysis Professional) certification is ISO-compliant (as is PMI’s) and is a rare certification so far with only 1,200 practitioners certified worldwide out of 12,000 IIBA members (vs. PMI’s 334,019 members and 412,503 PMPs).

However, drastic differences in the size of professional organizations and number of certifications do not represent the importance of both roles to organizations’ mission critical projects.

**What are their different responsibilities?**

This is a complicated answer due to various choices that companies make on how to best use their own human resources. At Ready2ACT, we have trained over 700 companies in Project Management and Business Analysis, so we will draw upon our observations over the last five years.

It’s vital to refer to the roles of Project Manager and Business Analyst and not job titles. These days employees are required to wear many different hats—regardless of their job titles. Many organizations and employees in them have clarity on what Project Managers do: they plan, manage, control, and close-out projects. However, most people have not heard of “business analysis” as a role, so it shows up instead in various titles such as DBA, systems analyst, financial analyst, consultant, process engineers, business architect, and data analyst to name a few. Employees may play the role of Business Analyst and not even know it has a name (like I once did). Business analysis can involve these processes (corresponding knowledge areas are in parenthesis):

1. Identifying needs and opportunities in the business (Enterprise Analysis)
2. Gathering, clarifying and validating requirements (Elicitation)
3. Writing and communicating requirements to stakeholders (Requirements Communication)
4. Developing a plan to gather requirements, clearly define scope, and manage changes to requirements (Business Analysis Planning and Monitoring)
5. Making sure that the requirements are well-written, specific, and complete (Requirements Analysis)
6. Ensuring that the solution is the best solution according to the requirements, not the other way around (Solution Assessment and Validation)

**How do Project Managers and Business Analyst interact together?**

While Project Managers always manage projects, Business Analysts (as a role) may or may not be working on projects. Sometimes, Business Analysts are responsible mainly for identifying opportunities for improvement, which may become process improvement projects. Other times, enterprise analysis is done at higher levels in the company or by specific departments, and Business Analysts do not become involved until there is a project already initiated and perhaps even planned. Generally, Business Analysts are instrumental in gathering and documenting requirements. In the Software/System Development Life Cycle (SDLC), Business Analysts are critical in the analysis phase. This phase is where most Business Analysts spend their time. Their ultimate goal is to get requirements right the first time. In fact, Microsoft determined that for every incorrect, unclear, or incomplete requirement, it costs 5 – 200 times more as it progresses through the SDLC (refer to the “The Cost of Requirements Errors” diagram).

**The Cost of Requirements Errors**

While most Business Analysts spend their time. Their ultimate goal is to get requirements right the first time. In fact, Microsoft determined that for every incorrect, unclear, or incomplete requirement, it costs 5 – 200 times more as it progresses through the SDLC (refer to the “The Cost of Requirements Errors” diagram).

**How do companies structure these roles?**

Many Fortune 100 companies have separate Project Manager and Business Analyst roles. Ready2ACT clients such as Aflac, Newmont Mining, and Sun Microsystems (now Oracle) clearly define when and how the Project Manager and Business Analyst work together. Most of the time, the Business Analyst participates in the Enterprise Analysis (prior to project initiation) and works together with the Project Manager to clearly define the Scope, mainly the Product Scope (features and functions of the product, service, or result – the deliverable) which is all about the requirements.

Small to Mid-size companies usually do not have separate people doing these roles; instead one person must do both Project Management and Business Analysis. This is why Ready2ACT has so many Project Managers attending Business Analysis classes,
and sometimes Business Analysts who attend Project Management classes.

Project managers who do not have someone to fulfill the Business Analysis role must do the work themselves or at least have a good understanding of what the role entails so they can assign it to team members and manage them. For PMPs, learning Business Analysis helps them become even more skilled at defining the project and product scope. As a result, this gives them more control over scope changes. Being able to manage and control changes effectively increases the probability of getting projects done on time, at or under budget, and to the complete satisfaction of stakeholders.

On smaller projects, companies may not have Project Managers available. So often times, Business Analysts must do the work of a Project Manager, as well. Business analysts need to learn the essentials of project management to be successful in managing projects. Even a one-day course will arm a Business Analyst with tools and techniques that are vastly different than what a Business Analyst typically does.

How much do Project Managers and Business Analysis make?

According to the 2009 PMI study across the globe, the average salary for a Project Manager in the United States is $86,049 - $101,295; those who have their PMP make up to $10,000 more each year.

According to the 2010 IIBA study, Business Analysts make $82,493 per year, on average. CBAPs earn $7,174.40 more each year. And, with a growing number of companies becoming aware of this profession and its certification, it may be a requirement on certain job descriptions and at the very least, is a differentiator when job hunting, which could give CBAPs an advantage.

So, who is more important, the Project Manager or the Business Analyst?

In summary, both roles are critical to the success of organizational initiatives. Formal education in these professions will help Project Managers and Business Analysts collaborate effectively and efficiently to save companies time, money, and energy.

Give us your feedback by taking our quick five question survey at www.Ready2ACT.com

Tiffany Dahlberg, PMP, is the Founder of Ready2ACT. Her passion is saving companies time and money by sharing her 20-plus years of experience and education by teaching Project Management, Business Analysis, and Professional Development.

It’s Time to Ramp It Up.

Improve your projects, processes, and people skills with proven tools and techniques.

Comprehensive Project Management with PMP® Preparation  
March 28 -April 1 or June 25 – 29

Comprehensive Project Management  
April 27 – 29

Essential Project Management  
June 1

Business Analysis Boot Camp  
June 27-29

Register for one of our Business Analysis or Project Management courses today.

Custom training solutions available.
Reach A Targeted Oracle Audience
Advertise Now!

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

RMOUG SQL>Update Advertising Rates

<table>
<thead>
<tr>
<th></th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business card or 1/8 page</td>
<td>$ 50</td>
</tr>
<tr>
<td>1/4 page</td>
<td>$ 350</td>
</tr>
<tr>
<td>1/2 page</td>
<td>$ 625</td>
</tr>
<tr>
<td>Full page</td>
<td>$1,000</td>
</tr>
<tr>
<td>Inside cover</td>
<td>$1,250</td>
</tr>
<tr>
<td>Back cover</td>
<td>$1,500</td>
</tr>
</tbody>
</table>

Discounts available for RMOUG Members and Full Year Contracts

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

SQL>UPDATE is mailed to all RMOUG Members and distributed during Quarterly Education Workshops and Training Days each year
Remembering Training Days 2011

Diane Ambrose, Carolyn Frye, and Rolland Carrillo Working the Registration Desk

Registration Area After Keynote Address

Heidi Kuhn receives the Doug Faughnan Lifetime Achievement Award

Irene Ludwig Calling the Sessions to Order

Tim Gorman and Debra Lilly Enjoying a Group Discussion at the Volunteer Reception

Oracle ACE Keynote Session
Training Days 2011

Heidi Kuhn receives the Doug Faughnan Lifetime Achievement Award

Carolyn Frye receives the 2011 IOUG Volunteer of the Year Award

John Jeunnette Awarded Oracle ACE

Irene Ludwig Calling the Sessions to Order

Left: Oracle ACE Keynote Panel

Right: The Stan Yellott Scholarship Fund Brings in These Students from the Pine Creek High School

Lunch is a Great Time to Network With Other RMOUG Members
Introduction to Data Vault Modeling

Compiled and Edited by Kent Graziano, Senior BI/DW Consultant

Note: This article is substantially excerpted (with permission) from the book Super Charge Your Data Warehouse: Invaluable Data Modeling Rules to Implement Your Data Vault by Dan Linstedt (Copyright © Dan Linstedt, 2008-2011) available online at LearnDataVault.com. All images are the property of Dan Linstedt. No part of this article may be reproduced in any form or by any electronic or mechanical means including information storage and retrieval systems, without permission in writing from the authors.

1.0 Abstract

The purpose of this article is to present an introduction to the technical components of the Data Vault Data Model. The examples give you the basics for how to build, and design structures when using the Data Vault modeling technique (there are many more details in the book...wink, wink). The target audience is anyone wishing to explore implementing a Data Vault for integration purposes whether it be an Enterprise Data Warehouse, Operational Data Warehouse, or Dynamic Data Integration Store.

2.0 Introduction

Data Vault Modeling is a specific data modeling technique for designing highly flexible, scalable, and adaptable data structures for enterprise data warehouse repositories. The formal definition is as follows:

The Data Vault is a detail oriented, historical tracking and uniquely linked set of normalized tables that support one or more functional areas of business. It is a hybrid approach encompassing the best of breed between 3rd normal form (3NF) and star schema. The design is flexible, scalable, consistent, and adaptable to the needs of the enterprise. It is a data model that is architectured specifically to meet the needs of today’s enterprise data warehouses.

The Data Vault model follows all definitions of the Data Warehouse (as defined by Bill Inmon) except one: the Data Vault is functionally based, not subject oriented – meaning that the business keys are horizontal in nature and provide visibility across lines of business.

1. Data Model Evolution

Data Vault is a hybrid, best of breed solution. The Data Vault is architectured and designed to meet the needs of enterprise data warehousing. It is NOT an adaptation. The research to develop the Data Vault approach began in the early 1990s, and completed around 1999 (see figure 2-1). Throughout 1999, 2000, and 2001, the Data Vault design was tested, refined, and deployed into specific early adopter sites. In 2002, the industry thought leaders were asked to review the architecture. In 2003, the first public classes teaching the modeling techniques to the mass public were offered (in Denver, Colorado).

When we look at the evolution of the data modeling architectures, we notice that there had not been an architecture specifically designed to meet the needs of Enterprise Data Warehousing. The Data Vault is just such an architecture. It’s an evolutionary approach because it combines best of breed (from 3rd Normal Form, and Star Schema), and re-architects from the ground up – specifically to meet the needs of the Enterprise Data Warehouse.

An analogy we have used is the Ferrari and the Big Rig. The Big Rig is the 3rd Normal Form (3NF), architectured and designed to meet the needs of a specific set of requirements (i.e., OLTP systems). It was adapted to meet the needs of data warehousing. The Ferrari (Star Schema) was architectured and designed to meet the goal of query speed to make data accessible to business users for analysis. It was adapted to meet the needs of enterprise-level data warehousing. Neither the Ferrari, nor the Big Rig was designed for the off-road/on-road requirements that the SUV space fills. The SUV could be considered a best-of-breed hybrid solution of truck and car together. The Data Vault is the SUV of data modeling as it was specifically architectured to meet the needs of Enterprise Data Warehousing.

2. Architectural Definitions

So where does a “Data Vault” fit in the big picture? The architecture is really compliant with the traditional Corporate Information Factory (CIF) approach as put forth by Bill Inmon (father of data warehousing). In that framework, the Data Vault fulfills the role of a centralized enterprise data warehouse (EDW) which in turn provides data to star schema data marts as well as flat (denormalized) report tables and/or exploration marts.

The Data Vault methodology includes each of these components. The architectural component discussed in this article is the central EDW/Data Vault.

3. EDW – Data Vault

The EDW (enterprise data warehouse), or core historical data repository, consists of the Data Vault modeled tables. The EDW holds data over time at a granular level (raw data sets). The Data Vault is comprised of Hubs, Links, and Satellites. The Enterprise Data Warehousing Layer is comprised of a Data Vault Model where all raw granular history is stored. Unlike many existing data ware-
houses today, referential integrity is complete across the model and is enforced at all times. The Data Vault model is a highly normalized architecture. Some Satellites in the Data Vault may be denormalized to a degree under specific circumstances.

The Data Vault modeling architecture has been likened to 3½ normal form. The business keys in the Hub appear to be 6th normal form, while the load date and record source are 3rd normal form. The Data Vault model should represent the lowest possible grain of source data. The Hubs and Links in the Data Vault model provide the back-bone structure to which context (the Satellites) are applied.

### 3.0 Hubs, Links, and Satellites

The Data Vault model consists of three basic entity types: Hubs, Links, and Satellites (see Figure 3-1). The Hubs are comprised of unique lists of business keys. The Links are comprised of unique lists of associations (commonly referred to as transactions, or intersections of two or more business keys). The Satellites are comprised of descriptive data about the business key OR about the association. The flexibility of the Data Vault model is based in the normalization (or separation of) data fields in to the corresponding tables.

![Figure 3-1: Example Data Vault](image)

Data Vault models are representative of business processes and are tied to the business through the business keys. Business keys indicate how the businesses integrate, connect, and access information in their systems. Data Vault models are built based on the conceptual understanding of the business.

Concepts such as customer, product, order, email, sale, inventory, part, service, account, and portfolio are used to represent ideas that cross lines of business. Examples of lines of business may include: sales, finance, marketing, contracting, manufacturing, planning, production, and delivery. These concepts can be represented with business keys that should cross lines of business.

The Links represent association across the business keys. The associations can change over time, some have direction (akin to mathematical vectors), others are directionless. Links are physical representations of foreign keys, or in data modeling terms: an associative entity.

Hubs and Links do not contain context. Satellites provide the context defining the keys and associations for a specific point in time. The Satellites contain the descriptive data attributes about a Hub or Link that can change over time. Satellites are the data warehousing portion of the Data Vault model.

Hubs and Links are like the skeleton and ligaments of the human body – without them we have no structure. Without them, our Data Warehouses are blobs of data loosely coupled with each other. But WITH them we have definition, structure, height, depth, and specific features. We as humans couldn't survive without a skeleton. The Data Warehouse cannot survive without Hubs and Links. They form the foundations of how we hook the data together.

Finally, the Satellites are added. Satellites are like the skin, muscle, and organs. They add color, hair, eyes, and all the other components we need to be described.

By separating the concepts of descriptive data from structural data, and structural data from Linkage data, we can easily begin to assemble a picture or an image of what our companies look like. The Hubs provide the working constructs to which everything is anchored. The Links provide the assembly structure to how these anchors interact, and the Satellites define the context of all of these components.

Remember this: the Data Vault is targeted to be an Enterprise Data Warehouse. Its job is to integrate disparate data from many different sources, and to Link it all together while maintaining source system context.

The Data Vault sits between the source systems and the data marts, as mentioned in Section 2.2. Why? Because the data marts are the interpretation layer of the integrated data warehouse data. In human terms think about it this way: think about a certain event that occurred in your life that you shared with another person. Do you both remember it the same way? Do you both remember the exact details? Or is your interpretation of the event slightly different than that of your friend? This is why it's so important to separate interpretation from the facts. Let your Data Vault house the facts, and build your data marts to house the interpretation.

### 4.0 Hub Entity Details

Hubs are defined by a unique list of business keys. They are surrounded with additional technical metadata elements such as load date time stamp, record source and sequence number (i.e., surrogate key). Business keys may be composite (made up of more than one field), intelligent (smart-keys) – contains meaning across parts of the keys, or sequential in nature.

Unfortunately in the real-world, what appear to be “business” keys change depending on the system being used. The keys change from one state to another as the customer information passes from one system to another. These changes are typically a manual process resulting in little to no visibility at the corporate level for where a customer is in the life-cycle of business.

One of the “jobs” that a good data warehouse should perform is: gap analysis - that is: provide the business with a view of the GAP between the way the business believes they are operating their business, and the way the systems are collecting the data. By examining this gap, the business can quickly locate where they are hemorrhaging money.

SQL>UPDATE • Spring 2011 19
1. Hub Definition and Purpose
The job of a Hub is to track the first time the Data Vault sees a business key arrive in the warehousing load, and where it came from. The Hub is a business key recording device. The business keys in a Hub should be defined at the same semantic granularity. Hubs have several standard fields including sequence number (or id), Load Date, and Record Source as shown in figure 4-1.

![Figure 4-1: Hub Example](image)

The purpose of the Hub is to provide a soft-integration point of raw data that is not altered from the source system, but is supposed to have the same semantic meaning. The resulting singular list of keys assists in the discovery of patterns across systems. The Hub key also allows corporate business to track their information across lines of business; this provides a consistent view of the current state of application systems. These systems are supposed to synchronize, but often don’t – when they don’t synchronize, business keys begin to be replicated and worse yet, are then applied to different contextual data sets. An example of Product Hub data is shown in Figure 4-2. Do you see and patterns in the Product # that may indicate data quality issues or other disconnects between systems?

![Figure 4-2: Hub Example Data](image)

2. Hub Entity Structure
The Hub entity structure consists of these required elements: a surrogate sequence id, a business key, a load date stamp, and a record source.

<table>
<thead>
<tr>
<th>PRODUCT #</th>
<th>LOAD DTS</th>
<th>RCRD SRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFG-PRD123456</td>
<td>6-1-2000</td>
<td>MANUFACT</td>
</tr>
<tr>
<td>P1235</td>
<td>6-2-2000</td>
<td>CONTRACTS</td>
</tr>
<tr>
<td>*P1235</td>
<td>2-15-2001</td>
<td>CONTRACTS</td>
</tr>
<tr>
<td>MFG-1235</td>
<td>5-17-2001</td>
<td>MANUFACT</td>
</tr>
<tr>
<td>1235-MFG</td>
<td>7-14-2001</td>
<td>FINANCE</td>
</tr>
<tr>
<td>1235</td>
<td>10-13-2001</td>
<td>FINANCE</td>
</tr>
<tr>
<td>PRD128582</td>
<td>4-12-2002</td>
<td>MANUFACT</td>
</tr>
<tr>
<td>PRD125826</td>
<td>4-12-2002</td>
<td>MANUFACT</td>
</tr>
<tr>
<td>PRD128256</td>
<td>4-12-2002</td>
<td>MANUFACT</td>
</tr>
<tr>
<td>PRD929929-*</td>
<td>4-12-2002</td>
<td>MANUFACT</td>
</tr>
</tbody>
</table>

Hubs must stand alone (be a parent to all other tables), they must never be children. Figure 4-3 shows the basic Hub Entity Structure.

![Figure 4-3: Hub Entity Structure](image)

3. What is a Business Key?
A business key is something that the business uses to track, locate, and identify information. It is the value you see in a data entry screen or may be used as a parameter filter on a report. It is a best practice to have unique business keys. The business key will also serve as a secondary unique constraint in the Hub table. Business keys are also known as natural keys. Business keys may actually be source system sequence (surrogate) ID’s that have been released to business users and now are embedded in business processes. Business keys are supposed to have meaning to the business independent from the operational system.

5.0 Link Entity Details
Link entities act as the flexibility component of the Data Vault model. They are the glue that pulls together any related association of two or more business keys. Where business keys interact, Links are created. Link entities are generated as a result of a transaction, discovery, relationship, or interaction between business units, business processes, or business keys themselves.

Links provide flexibility to the Data Vault model by allowing change to the structure over time. Mutability of the model without loss of history is critical to the success and long-term viability of the enterprise data warehouse. In other words, the model itself can now be adapted, morphed, and changed at the speed of business without loss of auditability, and compliance. The Data Vault model also gains flexibility from this technique because of the Link entity. The Link entity (in data modeling terms) is commonly referred to as an associative entity.

1. Link Definition and Purpose
A Link Entity is an intersection of business keys. It contains the surrogate ID’s that represent the Hub’s and Link’s parent business keys. A Link must have more than one parent key. A Link table’s grain is defined by the number of parent keys it contains. Each Link represents a unit-of-work (UOW) based on source system analysis and business analysis.

The purpose of the Link is to capture and record the past, present, and future relationship (intersection) of data elements at the lowest possible grain. The Link Entity also provides flexibility and scalability to the Data Vault modeling technique. Typical examples of Links include: transactions, associations, hierarchies, and re-definition of business terms.

2. Reasons for Many To Many Relationships
Within the Data Vault modeling constructs a Link is formed any time there is a 1 to 1, 1 to many, many to 1, or many to many relationship between data elements (business keys). The resulting physical Data Vault can capture “what the relationship was”, while it captures “what the relationship is”, and can adapt to “what the relationship will be in the future.”
Many-to-Many relationships provide the following benefits:

1. Flexibility
2. Granularity
3. Dynamic adaptability
4. Scalability

Many-to-many relationships allow the physical model to absorb data changes and business rule changes with little to no impact to both existing data sets (history) and existing processes (load and query). Businesses must change at the speed of business, and IT must become more agile and responsive to handling those changes. More and more business rules are changing, faster and faster.

Through the Link entity the Data Vault mitigates the need to restructure/redesign the EDW model because the relationship changes. For example: today the business states “1 portfolio can handle many customers, but each customer must be handled by 1 and only 1 portfolio.” If the model is designed in a rigid fashion (that is to say with parent-child dependencies) then it represents the current business rules quite well. All is well until the business (tomorrow, next year, or 2 years ago) decides to change their business rule: “now, a customer may be handled by 3 or 4 different portfolios.”

One of the problems of modeling today’s relationship in any data warehouse is that it makes the structures static. It forces the structures to represent today’s relationship rules. These relationships have changed in the past, and will change again in the future. This is the dynamic changing nature of the business: grow, change, or die. The problem with introducing static relationships in to the model is that it also re-introduces business rules to the loading processes. It also introduces static relationship enforcement in to the loading routines. When the relationship does change, IT is forced to re-engineer the loading routines, the modeling architecture, and the queries to get the data set in to the Data Warehouse. This is an unacceptable and un-maintainable cost going forward.

The Data Vault must remain flexible, and not introduce the need for re-engineering as the model grows. By modeling the Links as a many-to-many relationship, we can easily accomplish this goal. The Link table functions to future-proof the model and provide maximum flexibility. Figure 5-1 demonstrates the reason for using a Link table:

**Figure 5-1: Link Table Structure Housing Multiple Relationships**

### 3. Link Entity Structure

The Link entity structure consists of basic required elements: surrogate sequence id, multiple business sequence keys, load date stamp, and record source. Items such as last seen date, confidence rating, strength rating, encryption key, and possibly other meta-data elements may be added for query purposes, performance purposes, and discovery purposes as business requires.

The Link entity must **NEVER** contain natural business keys, or begin and end dates. If a Link structure is compromised, then the flexibility of the data vault model is immediately compromised. If the structure of the Link is compromised then you are sure to need reengineering in the future. Adding business keys to a Linked table insures that it depends on the business logic for loading; this raises the complexity of the loading routines. Links must contain two or more key sequence fields (from either Hubs or Links) in order to be considered valid; a Link with a single Hub sequence key is considered a peg leg Link and is invalid. Figure 5-2 shows the basic Link Entity Structure and an example table.

**Figure 5-2: Link Entity Structure & Example**

### 6.0 Satellite Entities

Satellite entities are the warehousing portion of the Data Vault. Satellites store data over time. Satellites are comprised of descriptive data that provide context to the keys and associations *at a point in time or over a time period*. Descriptive data in warehouses often changes; the purpose of the Satellite is to capture all deltas (all changes) to any of the descriptive data which occurs.

Satellites are typically arranged by type or classification of data, and rate of change. There are many different manners in which to setup classifications of data within a Satellite. For example, the attributes could be classified by data type, or by content, or by context – each of which will yield the same result physically – but a different result in the understanding or interpretation of the model.

Rate of change is yet another classification of Satellite data. Rate of change allows the Satellite to split away groups of fields that change more quickly than others. This prevents or removes the need for column data replication (of the slower changing attributes). By splitting the Satellites by rate of change, the rows are also reduced in size – allowing the data to insert more quickly, and be more responsive to real-time feeds. The lower the latency of arrival, the faster the database must respond with insert speed, the nature of these mechanics will be covered in the Data Vault implementation book.

#### 1. Satellite Definition and Purpose

A Satellite is a time-dimensional table housing detailed information about the Hub’s or Link’s business keys. The purpose of the Satellite is to provide context to the business keys. Satellites are the data warehouse portion of the Data Vault. The Satellite tracks data by delta, and only allows data to be loaded if there is at least one change to the record (other than the system fields: sequence, load-date, load-end-date, and record source). A Satellite can have one and only one parent table.

Satellites provide the descriptive data about the business key, or about the relationship of the keys. They describe the relationship changes over time. Their job is to record the information as it is loaded from the source system. They use load dates and
load-end-dates to indicate record life-cycles because most database systems today are not capable of internally representing time-series properly.

2. Satellite Entity Structure
The Satellite entity structure consists of basic required elements: surrogate sequence id (from the parent table), load date stamp, load end date stamp, and record source. Database engines today do not currently support (natively) time-series based table structures. Due to this limitation, the architecture is forced to compensate with Load Date Stamps and Load End Date Stamps. (Note: These date stamps have been described in detail Chapter 3 of the book.)

The Satellite entity must NEVER contain foreign keys (except for the single parent on which it relies). If a Satellite structure is compromised, then the flexibility of the model is immediately compromised, in other words: all possible hope of future proofing the data model is immediately lost. You are then forced to reengineer the data model in the near future when the business changes the way relationships are structured.

Satellites must have one and only one parent table, no others are allowed. Figure 6-1 below shows a standard structure of a Satellite Entity.

The key primary of the Satellite is a two-part key consisting of the PK Sequence of the parent Hub/Link combined with the Load Date time stamp. By adding millisecond timer, real-time data can easily flow directly into the Satellite without creating duplicate primary keys (as a result of load date collisions).

3. Satellite Examples
Figure 6-2 shows an example of Hub and Satellite related to Customer and changes over time to the Name. Note that most real Satellites will have more than one attribute.

4. Importance of Keeping History
History is partly what a data warehouse is all about. The Data Vault is no different, except that in the Data Vault, history is raw operational data (i.e., un-transformed). Satellite structures being what they are, can be changed, altered, and re-designed (as is documented in detail in the book).

7.0 Conclusion
This article has hopefully opened your eyes to a more flexible and agile (or at least novel) way to model a central EDW repository. While Data Vaults are not intended for direct end user query access with (or without) a BI tool, they are ideal for storing all the data and changes to that data from all of your source systems in a very scalable structure. They can then be leveraged for data mining as well as feeding data marts.

Using this modeling approach will allow you to adopt a more agile methodology as you can grow the model incrementally. You can start with just a small set of Hubs as you work with the business users to clearly identify the true business keys across the enterprise (and identify issues with those systems). You can then add Satellites to those Hubs (one at a time if needed) to capture the historical descriptive data. With Hubs and Satellites in place you can then add in the Link to capture all the important business key relationships. All along, with user involvement, you can be doing data quality checks and show them their data (integrated across source systems and functions) in a way that they may have never seen. Who knows what insights this may lead to...

About the Authors
Kent Graziano is a Senior BI/DW Consultant for TrueBridge Resources in Houston, Texas and a lifetime member of RMOUG. He is a certified Data Vault Master, an expert data modeler and architect with nearly 30 years of experience, including over 20 years working using Oracle (since version 5), Oracle Designer, and doing data warehousing. Kent has written numerous articles and done over 40 presentations (both nationally and internationally).

He was the recipient of the 1999 Chris Wooldridge Award (from IOUG) for outstanding contributions to the Oracle user community. In 2003 he was presented with The Doug Faughnan Award for his dedicated service and outstanding contributions to RMOUG. In 2007, he was the recipient of the ODTUG Volunteer Award. In 2005 he was named one of the first Oracle ACE’s by Oracle Corporation. He is a co-author of The Data Model Resource Book, Oracle Designer: A Template for Developing an Enterprise Standards Document, and The Business of Data Vault Modeling. Kent can be contacted at kent.graziano@att.net.

Dan Linstedt is the inventor of the Data Vault Data Model and Methodology. He has been in the IT industry and DW/BI for the past 20 years, as a consultant and systems architect, building both OLTP and EDW class systems for federal government and commercial enterprises. Dan is trained in SEI / CMMI Level 5 compliance and governance and is an expert in VLDB/VDW and large scale (petabyte) sized system design and architecture. He is now an independent consultant helping organizations around the world build successful, scalable, and repeatable enterprise data warehouses. He has written numerous articles and white papers and has presented internationally including TDWI, DAMA, and ODTUG. He is the primary author of The Business of Data Vault Modeling (available at lulu.com) and the new book Super Charge Your Data Warehouse: Invaluable Data Modeling Rules to Implement Your Data Vault (available at LearnedDataVault.com). Dan can be reached at danl@danielstedt.com.
RMOUG Call for Nominations to the Board of Directors

The Rocky Mountain Oracle Users Group continues to be the leading locally organized user group in the country. The success of RMOUG can be attributed to its members, the volunteers who support the activities of the users group, and the Board of Directors who provides direction for the users group.

The RMOUG bylaws require the regular election of the Board of Directors.

These bylaws (available online at http://rmoug.org/bylaws.htm#membership) outline the makeup and duties of the Board of Directors:

* The range of the number of the Directors shall be an odd number between five and fifteen.

* The management, control, and government of the Corporation (RMOUG) shall be vested in the Board of Directors.

* The Board of Directors shall preserve, protect, and promote the interests of the Corporation and will be responsible for formulating the general policy of the Corporation in accordance with the expressions of the members.

Note: current general policies for the users group can be found online at http://www.rmoug.org/policies.htm

It is time to begin the process by opening the call for Board Candidates for the fiscal year starting 01-June 2011 and running through 31-May 2012.

To run for the Board of Directors, you must:

* Be a RMOUG member in good standing;

* Be able to participate in scheduled Board meetings (approximately 1 per month)

* Be willing and able to uphold the bylaws and policies of the organization

* Have an e-mail address

* You must have actively participated as a volunteer on one or more RMOUG committee(s) or event(s).as spent at least one year a volunteer-at-large for the board or the RMOUG information systems group,

If you are interested in running for the Board of Directors, please prepare and submit a short campaign statement and biographical sketch. You should outline your reasons for running for the Board of Directors and your qualifications. If you are not a current member of the Board of Directors, you must list the event(s) or committee(s) on which you have participated and the name of a current Board member as a reference. Please limit your statement to approximately one page.

Campaign statements must be submitted no later than Monday, April 11, 2011 by email to the current RMOUG President at president@rmoug.org. If you have any questions about the process or would like to see a sample statement, please email president@rmoug.org

Instructor-Led Training At Your Desk!

Classroom style training over the web is available today. This is a new and exciting training format which fits into your busy schedule.

Web-Based Oracle Training by Dan Hotka:
* Morning Lecture Sessions (half days)
* Hands-on Labs
* Cost effective – includes all course materials

Advanced PL/Sql
* April 4-8, 2011

TOAD Tips & Techniques
* May 6, 2011

Sql Tuning for Developers
* May 2-5, 2011

Visit Dan’s web site for course outlines or email Dan with your level of interest. Dan also provides all of his course offerings for your company both over the web or on-site at a low course-fee (on-site course guides priced per student).

Contact Dan at:
www.DanHotka.com
Dhotka@Earthlink.net - (515) 279-3361

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
or
Email HeidiKuhn@rmoug.org
You may wonder, how did I get involved....

Well, it all started at the International Oracle Users Week (IOUW) back in 1994. I was the RMOUG volunteer for the conference bookstore. Back in the years of the IOUW conference, the Regional Oracle User groups were asked to provide volunteers to help man the bookstore during the conference. I really wasn’t busy attending sessions, so I said I would volunteer for RMOUG. I truly enjoyed my time in the IOUW bookstore; not only did it keep me busy while John presented and attended sessions, but I also got to meet amazing people from all parts of the world that over the years have become good friends.

One of those amazing people was Stan Yellott. Every year that I was there he would come by the bookstore and thank me for volunteering on behalf of RMOUG. I truly enjoyed my time in the IOUW bookstore; not only did it keep me busy while John presented and attended sessions, but I also got to meet amazing people from all parts of the world that over the years have become good friends.

One of those amazing people was Stan Yellott. Every year that I was there he would come by the bookstore and thank me for volunteering on behalf of RMOUG. Then one year Stan asked me if I could cover as a Room Monitor for the IOUW as they were short that year on volunteers. I did and when the bookstore closed I continued volunteering as a Room Monitor at IOUG-Alive conferences. During that time, I was asked to work on the Finance and Web committees for IOUG. When IOUG-A Live became part of Collaborate in 2006, I was approached by the conference chair and committee to serve as Assistant Room Monitor Manager, with the following year taking on the role as Room Monitor Manager for the conference. Since 2007, I have volunteered as Room Monitor Manager and have continued to be active as an IOUG Conference Committee member.

Along the way, Stan and I became very good friends and one day he asked me if I would be interested in volunteering to help coordinate RMOUG Training Days 2004. Our original plan was that I would work with the previous Training Days Director to learn the ropes and then take on the role the following year for the 2005 conference. But, as you know, plans sometimes go awry as did this one. I was asked by the RMOUG Board and accepted the role of Training Days Director in 2004 and with their help and the help of the crew from YCC, we had a very successful conference. In 2005, I was officially elected to the RMOUG Board of Directors and continued in the role as the Training Days Director for 2005 and 2006. In 2007, I was voted in as Vice President and continued as Vice President...
and Training Days Director for 2008 and 2009. In 2009/2010 and 2010/2011 I was voted in as President for RMOUG. I have truly enjoyed my time working with all of the volunteers for RMOUG, but as with all good things, this time has come to an end. This will be my last year on the RMOUG Board. John and I will be moving back to Arizona later this year to be closer to both my parents and John’s father.

Now for those of you who would like to know more about me personally, let’s take a stroll down memory lane. You will need to go back in time to 1969 -- at thirteen, your world is just opening up and this was an exciting time for me. I was about to join a group called the Civil Air Patrol and learn about airplanes and flying, air search and rescue missions, and explore my options for joining the Air Force. I was very interested in the Air Force at this time and I thought of becoming a lawyer for the Adjutant Generals Office. But instead, I met John and our lives were never the same. We dated in high school and married soon after we graduated. We began our family quite young and with two young boys in tow moved to Colorado in 1982 to begin our new adventure. My first experience with snow was in Colorado and the Christmas blizzard of 1982 (come on now... no laughing. I grew up in Arizona– “the DESERT!”). Our white Toyota Corolla was completely buried under snow, so no driving that weekend, but we did build snow forts with the boys. Today our sons have families of their own and we are blessed to have five grandchildren.

Sean and Kat with JJ and Evan live close by in Littleton. While Brian and Catherine with Isabelle, Madeline, and Jacqueline live in Austin, Texas. Last July, we were fortunate to get our whole family together (picture time!!!) and thoroughly enjoyed watching all of our grandchildren play and at times Miss Molly would join in.

I bet you may be wondering --- Who is Miss Molly? Well, that’s my fun and somewhat techie story. A few years ago I decided that I wanted to get a dog. John’s first response to me was “really?” and my reply was “Yes” and off I went to begin my search. I started a Google query asking for a “smart, low shedding, good with kids, large dog (John’s request)” and Google’s response was Poodles, Goldendoodles, and Labradoodles. As I continued to research my options from Google, I decided I really wasn’t excited about the Poodles or Labradoodles, but fell in love with the story and pictures of the Goldendoodles. So my next step was to find one and, of course, I went back to my friend Google and queried for Goldendoodles and Denver. I quickly found puppies in Broomfield and within an hour we were off to pick her up. That was four years ago...today. Molly and I have trained to be a certified therapy team and last September we went to Fort Collins for the Doggie Olympics (no medals, but we had a great time).

Finally, I just wanted to say “Thank You.” I have had a great run and have enjoyed the journey along the way. I am really looking forward to what the future brings both for me and for RMOUG.

Adios mi amigos....

Editor’s Note:

From all the members and Board of Directors at RMOUG, we wish John and Peggy all the best. They go with our most sincere, heartfelt appreciation for their contributions to our group. Peggy - you will be missed!!
I have been a member of RMOUG since moving from Houston to the Denver area in 2007. We now live in the foothills West of Conifer near US-285. Trish, my wife of 30 years, and I both consider Houston is a nice place to be “from”, and we still have family there. But hardly a day goes by that I don’t ask ourselves why we waited so long to move from the heat and humidity of the Gulf Coast to what has to be the best climate and scenery anywhere in the country. Actually, we waited until our children were settled into adulthood. Our daughter and son-in-law recently bought their first house, and our son is a Junior at University of Houston.

Upon hearing of my plans to relocate, a colleague from this area joked that a Coloradan’s favorite view is a Texan returning to Texas with a Californian under each arm. He also expressed doubt about how I would like the winters in the foothills, having no experience with snow. (Yes, in both cases, it was simply good-natured jest.) But I am pleased to report that everyone I’ve met here, at work, at church, in my neighborhood, and other venues, has been welcoming and friendly. Plus, it only took one winter to learn the secrets of living with the snow in the foothills: snow tires and a snow blower.

Outside of work, one of my favorite activities is outdoor photography. For many years I stuck with 35mm film, waiting for the quality of digital cameras to pass the quality of 35mm, and waiting for the prices to moderate. In 2009, I finally made the switch, and found enormous freedom not available with 35mm, such as changing speeds without changing film, taking many more pictures and tossing the bad ones, or tweaking the exposure and color via computer. Of course, outdoor photography is one of the best things about living near Denver. Places like Rocky Mountain National Park, the Maroon Bells, and the San Juan Mountains are all just a few hours from here. The list of possibilities is virtually endless. Some of my pictures are shown on these pages. We especially enjoy hosting family members visiting from Houston, as it allows us to brag about the scenery and the climate. The editor of RMOUG’s SQL>Update Magazine have already used one of my photographs on the cover of the recent Winter Issue. I hope to submit more in the future.

Beyond the boundary of Colorado, the entire Rocky Mountain West is within a two-day drive of the Denver area, as compared to three or four days from Houston. Trish and I both like camping and hiking in the mountains. We have vacationed in several of the surrounding states in the past, and intend to use our current strategic location to continue that tradition.

Another of my hobbies is playing violin in a community orchestra. I studied violin with private teachers as a teenager many years ago, and achieved a degree of success at the high school level. During my senior year of high school, I was the first-chair violinist and featured soloist in the All City Orchestra of the Houston Independent School District, and also reached the All State level. Upon entering college it was clear that my best bet for a career was technology, but violin would remain an important hobby. Presently, I am a member of the Jefferson Symphony Orchestra, based in Golden on the Colorado School of Mines campus. The greater Denver area has an impressive list of community orchestras staffed with both professionals and hobby musicians such as myself, playing music that is quite challenging, from Beethoven to Rachmaninoff. Naturally, I think the JSO is one of the best, but I don’t want anyone to think I’m biased.

All of us in RMOUG can be proud of the technology industry here in the Denver area. While researching the job market in 2007, I learned that outside of the East and West Coast areas, few cities in the country have the depth of technical talent and the number of technical jobs that we have here. Dallas, Austin, Chicago, or Salt Lake City have tech jobs as well, but when you combine the job opportunities with everything...
A background in database technology was a great way for me to relocate to Denver. Ironically, while I studied computer science in college, I never had any college database courses. I started my introduction to computer technology as an undergraduate at Rice University in Houston in the late 1970’s and early 1980’s, studying both computer science and space physics. It was here that I got a great foundation in the important aspects of computer science: general programming and good program structure; operating systems; numerical linear algebra; digital system architecture. While the technology has changed considerably since then, much of that foundation is still useful. For example, I first used UNIX (AT&T System VII) on a DEC PDP-11 while at Rice. I also worked part time as a programmer at Baylor College of Medicine, where more seasoned professionals taught me the importance of precision in analyzing user requirements and the importance of a good user interface, even though it was on scrolling character terminals at the time. To this day, I still enjoy swapping “old-timer” stories about punch cards, 9-track tapes, and IBM Job Control Language with other long-time practitioners. I graduated from Rice in 1982 with a bachelor’s degree in computer science and space physics.

My plan as an undergraduate was to work in the space program. After graduating from Rice, I was employed by McDonnell Douglas Corporation, working at Johnson Space Center in Houston, on the team that analyzed the requirements for the ascent/entry space shuttle onboard navigation software. (To be clear, McDonnell Douglas did not write the onboard software. We worked with NASA engineers to define the functional specifications from which the software was written). Working with NASA gave me exposure to very careful processes for testing and change control on mission-critical systems. I also had the opportunity to work in Mission Control for the entry phase of six shuttle missions plus a host of training simulations, monitoring the performance of the onboard navigation system in real time. I was not in the main control room that everyone has seen on the news; rather, I worked on one of the support consoles in a separate room and talked via intercom with the Guidance Officer in the main control room regarding the health of the onboard navigation system. In many ways, the simulations were the most interesting, because that is when the trainers deliberately introduced failures to see how the onboard crew and ground support teams would handle them.

During this time, I obtained a Master of Science in computer science from the University of Houston, which provided deeper experience in compiler construction, operating systems, artificial intelligence, and numerical analysis. But I still did not study database systems in college.

In 1988, I decided to leave the space program, for two main reasons. First, the Challenger disaster a year earlier had caused both a slow-down and general loss of focus. Second, there was an enormous mass of software outside of mission-critical systems that had been written by engineers, mathematicians, and physicists who were good in their field but whose only exposure to programming was reading the FORTRAN reference manual. The result was substantial inertia against incorporating good programming practices and computer technology that was available outside of the space program.

From 1988 through much of the 1990’s, I worked as an application programmer and database developer at various companies. It was here that I finally started using and developing relational databases, using Oracle version 5, SQL, SQL-Forms, and Pro-C, later moving into versions 6 and 7, and PL/SQL. All of my database experience has come from on-the-job training, practice, learning from other people with more experience than myself, Oracle’s classroom courses, and reading books and articles by Millsap, Nanda, Kyte, Date, and others. During this time, I heard a very appropriate quote, attributed to numerous different people: “Good judgment comes from experience, and experience comes from bad judgment.” I learned a lot from mistakes, both my own and others, but I also learned from things that worked well, and from people with experience. I learned such things as:

- The value of a well-defined data model as the shared center point of applications, reports, and analysis
- SQL performance tuning
The value of PL/SQL (introduced in Oracle version 6) for heavy data crunching, by performing the work as close to the data as possible.

The benefits of in-database constraints, introduced in Oracle version 7.

The importance of large-scale performance testing to ensure that a system will handle real workloads.

Careful change management to ensure successful deployments.

Even though computer technology in general, and Oracle specifically, continues to change, the value of these lessons has stood the test of time. I continue to apply what I learned even 15-20 years later.

From 1997 to 1999, I worked for Oracle Corporation as a consultant in the Telecom Consulting Division, during the time that many consider to be the heyday of the tech boom. It was there that I learned valuable skills in building large-scale data warehouses, such as the need to validate the data being loaded, and partitioning introduced in Oracle version 8. One of my assignments was a large technology company near Denver, which involved commuting from Houston many times over about 10 months. It was my first chance to become familiar with the area, and also to meet Denver people in the database field who would later provide connections to the opportunity for the position I have now.

Late in 1999, I decided that while e-commerce was great, e-marriage and e-parenthood were not, and that I should stop traveling while my children were teenagers. From 2000 to 2007, I was an independent contractor, working primarily in the Houston area. I was able to help my clients solve problems in performance and scalability, do pro-active performance testing, and create custom applications that included heavy data and number crunching.

Finally, in 2007, I relocated to Denver and took a position as a database designer and DBA for Collect America, Ltd., now called SquareTwo Financial. In 2010, I was promoted to Data Architect, responsible for overseeing the design and construction of all of the company’s databases, but primarily focusing on an enterprise data warehouse.

It was also during the last 3-1/2 years that I was introduced to RMOUG Training Days. The most difficult part of attending is narrowing down to one presentation per session. Almost every session has two, three, or more topics that directly relate to my work, but sadly I have not yet perfected the art of being in two places at once! Many of the presenters have material that is both interesting and thought-provoking, raising questions about how or why I have done things or should be doing things. In other cases, it is gratifying to hear experts discussing situations similar to my own, making recommendations that closely align with what I and my managers and colleagues are already planning. I was able to make one presentation myself in the 2010 conference, describing my team’s practices for change tracking that have allowed us a record of near zero deployment-related defects in the last two years.

The experiences I received as a consultant and data architect have reinforced my beliefs in earlier lessons: well-defined data models, performance testing, careful change management. I have also reached a point in my career where it is important that I pass on things I’ve learned to others. I have also reached a point in my career where it is important that I pass on things I’ve learned to others. If you will indulge me for a few minutes, I would like to summarize some things I’ve learned that I believe will increase the chances for database projects to succeed.

A data model exists not just to provide structure for a database. It also exists to convey meaning to programmers and users. The meaning of the data elements and relationships must be well understood and documented.

A database should be physically designed for scalability in advance, such as partitioning, rather than waiting for trouble to arise.

The database is not just shared data; it is shared operations on the data. Specifically, functionally cohesive business operations that involve sets of related rows should be implemented once as stored procedures, thereby serving any application regardless of technology, and providing performance benefits by reducing traffic between client and database.

Finding dependencies between stored procedures and the data model is a static compile-time operation, while the same dependencies between OCI/JDBC clients and the data model is a run-time operation. I have found compile-time analysis of stored procedures to be more thorough and robust than run-time analysis for client programs.

Transaction boundaries should be designed in advance to delineate logically complete business units of work. They should not be used as a way to “save” partial results, because that complicates the task of backing out incomplete work after an error.

Error processing is not complete until a person or program reviews and resolves the errors. Simply logging errors into a write-only log that no one reads provides no benefit.

As much as practical, integrity constraints must be enforced inside the database, so data is protected from bugs in the application, bugs in the ETL, bugs in bulk loads, or mistakes in manual ad hoc activity.

I understand that practitioners with an application development background may disagree with me on some of these points. But keep in mind: I have been an application programmer, and I still do not consider it a good idea for an application to use an Oracle database only for a persistence layer or key record manager.

In my current position at SquareTwo Financial, and as a member of RMOUG, one of my goals is to take lessons learned both from good and bad situations, and convey those lessons to others, so they can build on that experience. I hope this summary of lessons learned over 30 years working with computer technology and 20 years of working with Oracle will be beneficial to other RMOUG members. Beyond that, I can tell you from first-hand experience what most of you already know: Denver is a great place to work and live.
RMOUG Board of Directors

Meet Your Board

Peggy King, King Training
President
Business: (303) 798-5727
Business: (800) 252-0652
E-mail: President@rmoug.org

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

Allison Leech
Secretary
E-mail: Secretary@rmoug.org

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

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

Barbara A. Lewis
Membership Director
Axia College of the University of Phoenix
Voice: 303-757-6709
Email: MembershipDir@rmoug.org

Carolyne Frye
Programs Director
E-mail: ProgramsDir@rmoug.org

Tim Gorman
IS Director
Email: ISDir@rmoug.org

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

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

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

Pat Van Buskirk
Newsletter Director
E-mail: NewsletterDir@rmoug.org
RM OUG Events Calendar

3/24/11 DBLabs Zettapoint & Ruby - Regis University Denver
3/16/11 Board Board Monthly Meeting - Conference Call @ 6:00pm
3/18/11 Newsletter Newsletter - Mail Spring Issue
4/5/11 QEW QEW - Call for Presentations
4/6/11 Board Board Monthly Meeting - Via Conference Call @ 6:00 pm
4/10-14/2011 Conference Collaborate 2011 - Orange County Convention Center - Orlando Florida
4/11/11 Board Call for Nominations - RMOUG Board of Directors
4/22/11 QEW QEW - Preliminary Agenda
4/29/11 QEW QEW - Deadline for Presentations
4/29/11 Scholarship Scholarship - Deadline for Applications for Spring Scholarship
5/2011 DBLabs TBD - Regis University Denver
5/1/11 Newsletter Newsletter - Call for Articles Summer Issue
5/20/11 Board Board Meeting - 7:00am before first QEW session
5/20/11 QEW Quarterly Educational Workshop - Oracle Sun Campus
5/20/11 Scholarship Scholarship - Announce Spring Scholarship recipients at QEW
5/15/11 Newsletter Newsletter - Deadline for Articles Summer Issue
6/15/11 Board Board Monthly Meeting - Corporate Offices @ 6:00pm

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

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

<table>
<thead>
<tr>
<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>
</tr>
<tr>
<td>1/2 Page</td>
<td>$350.00</td>
<td>$312.50</td>
</tr>
<tr>
<td>Full Page</td>
<td>$350.00</td>
<td>$500.00</td>
</tr>
<tr>
<td>Inside Cover</td>
<td>$350.00</td>
<td>$625.00</td>
</tr>
<tr>
<td>Back Cover</td>
<td>$350.00</td>
<td>$750.00</td>
</tr>
</tbody>
</table>

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

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

Tired of the I-25 Corridor?
60 Acres In Gorgeous Bijou Basin
Less Than An Hour From DTC

- Grassy Meadows & Pines
- Excavated Road and Building Site
- Power & Phone At Property Line
- Water Rights
- One Hour From DIA
- Possible Owner Carry W.A.C.

30777 Magic Dog Circle, Kiowa, CO
$270,000
Pat Van Buskirk • Coldwell Banker, Parker Office
(303) 243-0737 Cell
www.patvanbuskirk.com pat@patvanbuskirk.com

Breakfast Discounted Total Cost
Ad Rate
$350.00 $175.00 $525.00
$350.00 $312.50 $662.50
$350.00 $500.00 $850.00
$350.00 $625.00 $975.00
$350.00 $750.00 $1,100.00
Join us for our next Quarterly Education Workshop in August at the Oracle Sun Campus. 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
WHAT IS THE BEST WAY TO ORGANIZE AND INTEGRATE YOUR ASSOCIATED DOCUMENTS TO ORACLE E-BUSINESS SUITE MODULES?

A. Store Documents in an Oracle Database and slow down performance by asking it to do something it wasn’t intended to do.

B. Purchase a silo storage location but worry about how departments can share information with each other in an easy and efficient manner.

C. Purchase product that requires expensive Oracle EBS customization and screen-scraping for any search capabilities.

D. Keep everything in file cabinets and storage rooms, because losing documents makes finding them so much more rewarding.

E. Use DocSavi: Configurable middleware which works with all EBS modules and uses standard Oracle functionality.