IMPLEMENTING ORACLE BI EE ON TOP OF ORACLE OLAP CUBES

Dan Vlamis, Vlamis Software Solutions, Inc., dvlamis@vlamis.com

Preface

As of this writing, Oracle Business Intelligence and Oracle OLAP are in a period of transition. Oracle is in the midst of integrating the Siebel line of products into their business intelligence offerings. Siebel (now Oracle Business Intelligence Enterprise Edition) is really more of a relational OLAP (ROLAP) product, whereas Oracle's Discoverer Plus OLAP was really more of a multi-dimensional OLAP (MOLAP) product. While this transition is taking place, it's tough to predict exactly where Oracle is heading. Indeed, the direction is perhaps even changing more with the announcement of Oracle's acquisition of Hyperion, including the Essbase multi-dimensional database product.

The presentation will go over how cubes are created with AWM, how they are exposed with the new OLAP View Generator for AWM 10.2.0.3 (see OLAP View Generator at http://www.oracle.com/technology/products/bi/olap/index.html), and how this view is referenced in Oracle BI Administrator. A demonstration of the finished product will show the benefits of using Oracle Business Intelligence Enterprise Edition on top of Oracle OLAP cubes stored in analytic workspaces.

There may be an opportunity to share information about Oracle OLAP 11g, where integration of multi-dimensional sources with relational tools becomes more "automatic." Again, this area is changing rapidly as the technology matures.

Introduction

Oracle's integration of Oracle Business Intelligence Enterprise Edition (OBIEE) into the Oracle product family provides exciting opportunities to extend the use of Oracle's multidimensional OLAP cubes to include Oracle's new customers who have previously adopted the Siebel environment. We will briefly discuss the development of the multidimensional environment using Oracle Analytic Workspace Manager (AWM), then focus on the value of using these two technologies together. In a live demonstration, the presentation will highlight the benefits of using OBIEE against Oracle OLAP Analytic Workspaces (AWs). The demonstration draws on the strength of both products to show how the combination of OBIEE and Oracle OLAP can deliver attractive analysis of business data.

Multi-dimensional Model

Multi-dimensional models are best explained using examples. This presentation will use a fictitious "Global Computing" company's sales data as its starting model. Global Computing Company distributes computer hardware and software products. These sales are collected on a monthly basis and are broken down by CHANNEL, CUSTOMER, PRODUCT, and TIME. Each of these becomes a *dimension* in the multi-dimensional model. Each dimension can have multiple levels that comprise one or more hierarchies.

Multi-dimensional data is logically organized into a *cube* or a series of cubes. The dimensions organize and index the cubes; the measures contain the data. For those familiar with data warehouses, a cube is generally loaded from a fact table. For our analysis, we will focus on Global Computing's SALES_CUBE. This cube has measures SALES and UNITS. SALES contains the dollars sold through each CHANNEL to each CUSTOMER for each PRODUCT for each MONTH. Similarly, UNITS contains the number of items sold broken down by these same dimensions.

Storage of Multi-dimensional Data

This multi-dimensional data can be stored in dimension and fact tables in the relational database. The Oracle9i OLAP Option added a new storage mechanism specifically designed for storing multi-dimensional data—the *analytic workspace*. This is a special type of table that is specifically designed for multi-dimensional data. Data is stored in a large object binary (LOB) and is managed by Oracle OLAP. The data still resides in a tablespace and is fully integrated into the rest of the Oracle database and accessible by SQL, as well as other tools specifically designed for querying multi-dimensional data. Internally, each cube is stored as an array of data, with the dimensions serving as the common indexes to the data. This storage architecture is highly optimized for dimensional data. By storing data as arrays, Oracle OLAP can easily find any data using multi-dimensional cursors. It also avoids having to store the keys of the fact table in the data. More benefits are described in Oracle's Data Sheet Oracle Database 10g OLAP Option.

Using AWM to Build Analytic Workspaces

Oracle's Analytic Workspace Manager is specifically designed to build and manage these analytic workspaces. AWM makes dimensional modeling accessible to just about anybody, and is designed for use by IT specialists, departmental DBAs, and BI specialists.

AWM can be used to design a dimensional model, create an analytic workspace, and then populate the AW from a series of relational tables. AWM has become the one tool needed to manage your analytic workspaces. This metadata serves a similar purpose to Oracle Discoverer's "End User Layer" that organizes your tables into dimensions, levels, and hierarchies.

AW Metadata

With Oracle OLAP 10g, metadata is stored directly in the analytic workspace. This greatly simplifies the process of designing and creating cubes. AWM manages all of this metadata creation automatically.

For example, when you create a CUSTOMER dimension using the AWM interface (see Figure 1), AWM automatically creates hierarchies, levels, and attributes for you. It has "intelligent defaults", such as setting the first hierarchy you create as the default hierarchy to use for drilling purposes.

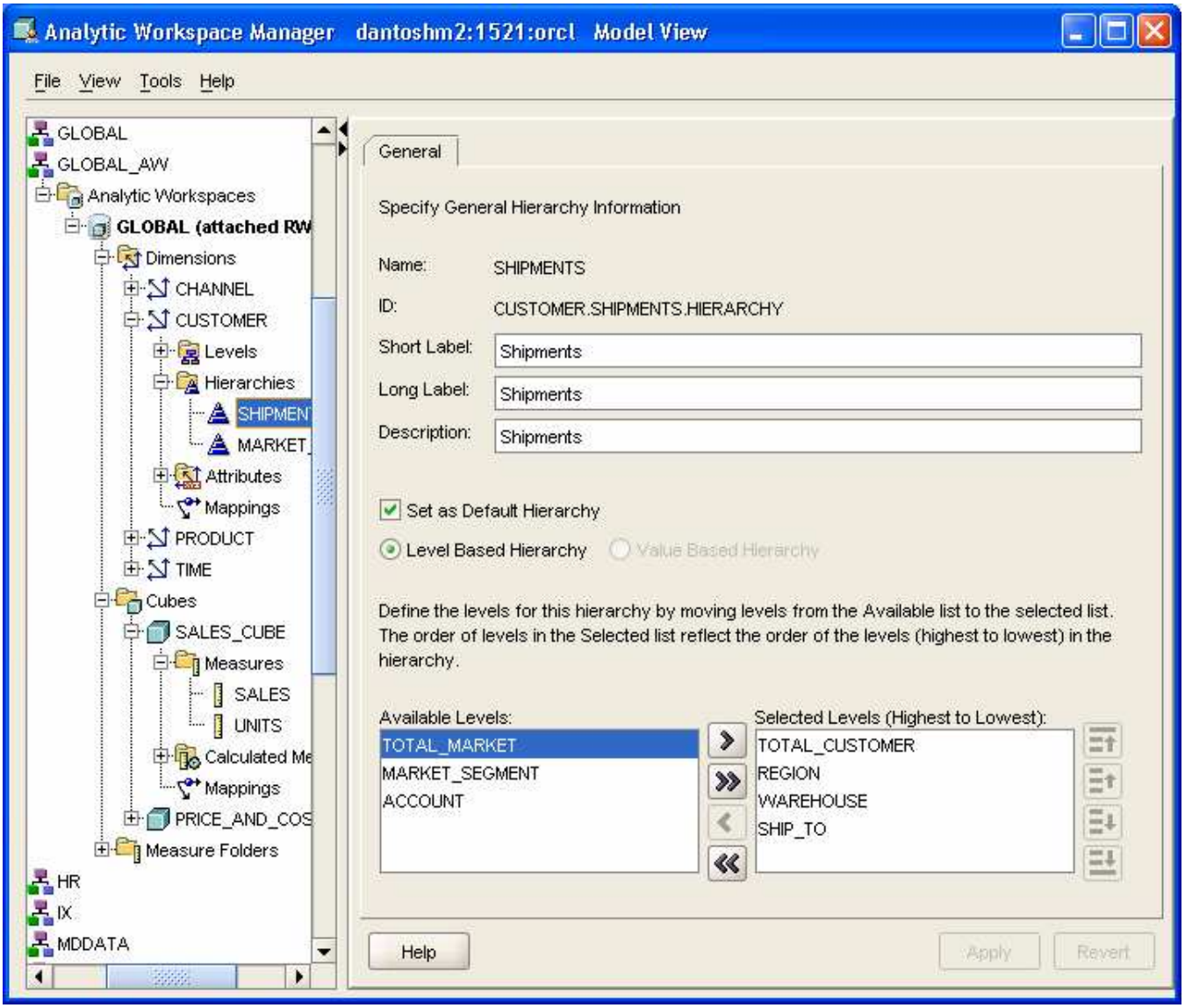


Figure 1 – AWM Screen for Creating Customer Hierarchy

This metadata is stored directly in the analytic workspace. When you create your dimensions, AWM automatically creates the metadata necessary to present this data to a user.

Mapping Cubes with AWM

AWM is also used to map sources for your analytic workspace. With it, you link up the relational tables that serve as the source for your data. Figure 2 shows a mapping from the relational table UNITS_HISTORY_FACT to the cube SALES_CUBE. This is the "magic" that allows Oracle OLAP to load the measures in the SALES_CUBE from the fact table. In this example, the user has "wired up" the SALES column to the SALES measure, the UNITS column to the UNITS measure, the MONTH_ID column to the MONTH level of the TIME dimension, the SHIP_TO_ID column to the SHIP_TO level of the CUSTOMER dimension, and the ITEM_ID and CHANNEL_ID columns to the appropriate levels of the PRODUCT and CHANNEL dimensions. Notice that the windows representing the tables and dimensional model objects can be drilled and rearranged.

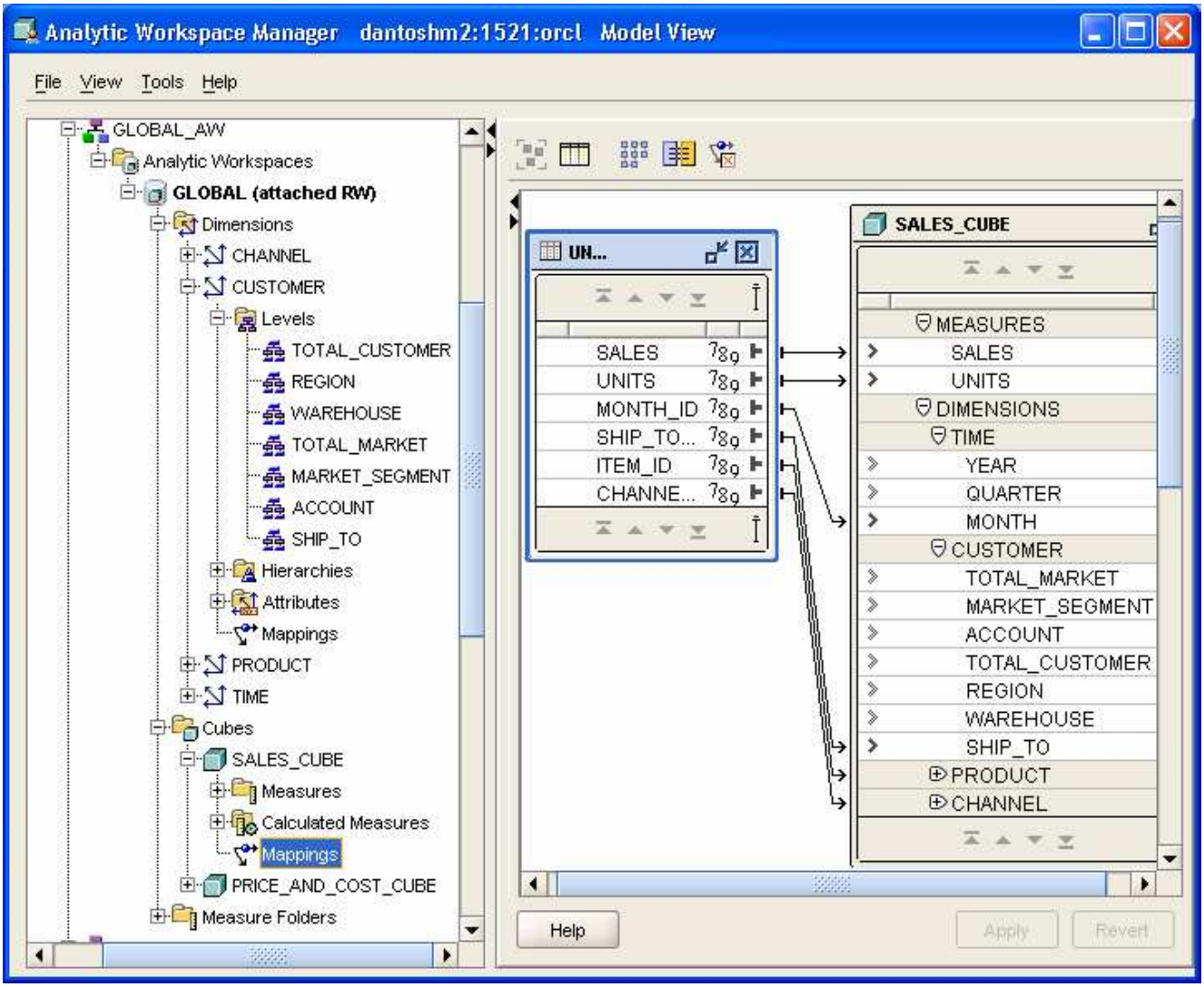
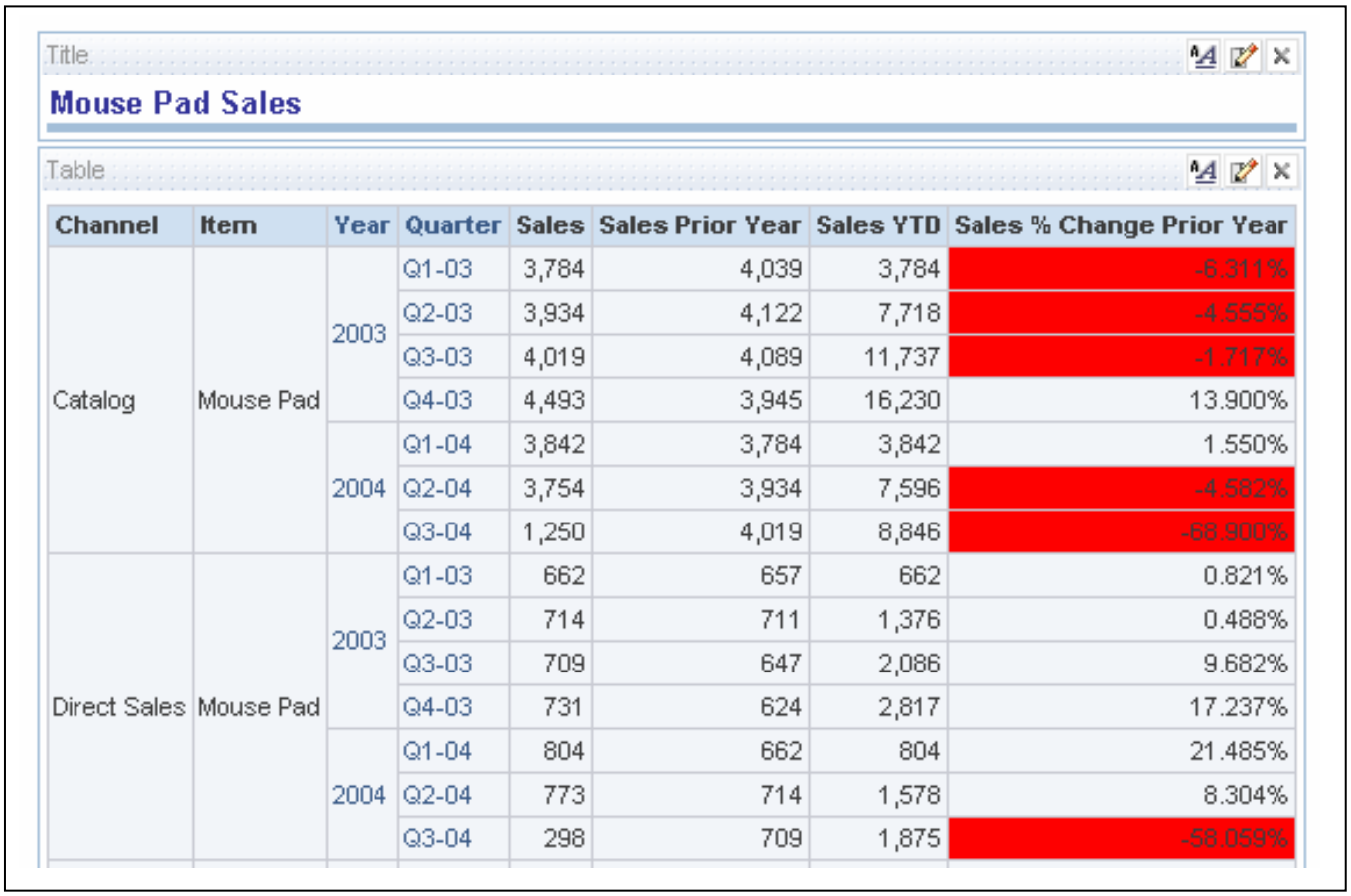


Figure 2 – Mapping Relational Sources to Cubes with AWM

Making AWs Work with Oracle BI EE

Oracle Business Intelligence Enterprise Edition (OBIEE), formerly Siebel Analytics, presents relational data to the user in a pseudo-multidimensional manner, with many of the same features and functionalities present in AWM. OBIEE requires a relational data store, instead of the multi-dimensional data store used by Discoverer for OLAP. This allows OBIEE to use any of several relational database technologies, such as Microsoft SQL Server, as its data storage engine. OBIEE is a feature-rich analysis environment, incorporating dashboards and other controls that provide fine-grain control over the user environment.



Title

Mouse Pad Sales

Table

Channel	Item	Year	Quarter	Sales	Sales Prior Year	Sales YTD	Sales % Change Prior Year
Catalog	Mouse Pad	2003	Q1-03	3,784	4,039	3,784	-6.311%
			Q2-03	3,934	4,122	7,718	-4.555%
			Q3-03	4,019	4,089	11,737	-1.717%
			Q4-03	4,493	3,945	16,230	13.900%
		2004	Q1-04	3,842	3,784	3,842	1.550%
			Q2-04	3,754	3,934	7,596	-4.582%
			Q3-04	1,250	4,019	8,846	-68.900%
Direct Sales	Mouse Pad	2003	Q1-03	662	657	662	0.821%
			Q2-03	714	711	1,376	0.488%
			Q3-03	709	647	2,086	9.682%
			Q4-03	731	624	2,817	17.237%
		2004	Q1-04	804	662	804	21.485%
			Q2-04	773	714	1,578	8.304%
			Q3-04	298	709	1,875	-58.059%

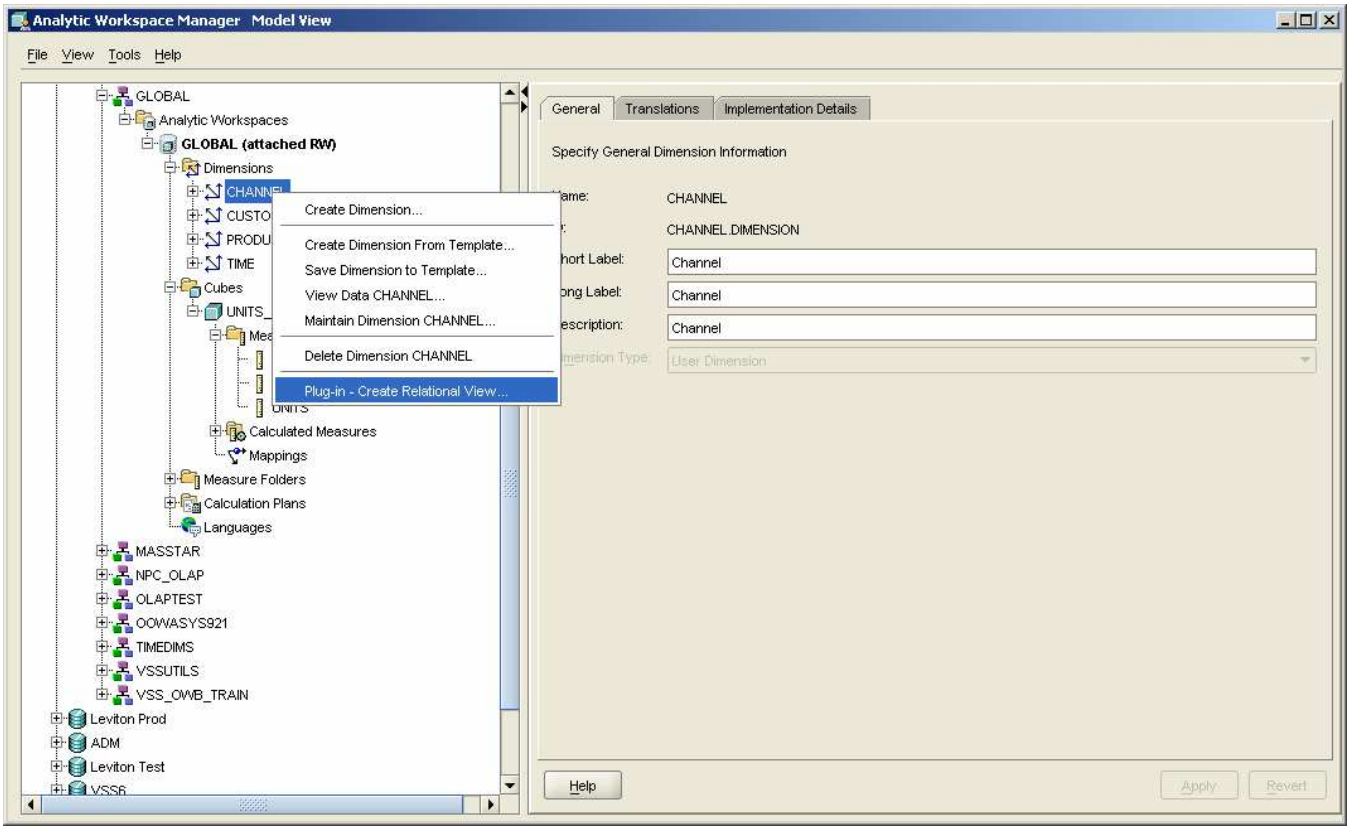
Integration of OBIEE and AWM

In the OBIEE environment, we see the same numbers as we see in the AWM/Discoverer environment. And in fact, the two environments are looking at the same data objects. Not replications, not copies, but the same physically stored data objects. So, how do we integrate the two environments? How can we use the strengths of the multi-dimensional OLAP engine for processing large amounts of data, but view the data using OBIEE? Since OBIEE requires a relational view of the data, and since our data is stored in the multi-dimensional Analytic Workspace, we need a method to present the data in the AW in a relational view.

OLAP_TABLE and LIMITMAP

Within the Oracle Database engine, Oracle has developed a function called an OLAP_TABLE. Relational views can be defined that refer to the OLAP_TABLE function, which in turn refers to AW (multi-dimensional) data. Dimensions are presented as columns in those views, as are data measures which share those dimensions. The OLAP_TABLE function handles all of the transitions required to present multi-dimensional data in a relational format.

Oracle has simplified the creation of the OLAP_TABLE with a wizard that is called directly from AWM.



The wizard created this new OLAP_TABLE:

```
CREATE OR REPLACE FORCE VIEW "GLOBAL"."CHANNEL_DIMVIEW" ("CHANNEL", "CHANNEL_LEVEL",
"PRIMARY_SORT_ORDER", "CHANNEL_SDSC", "CHANNEL_LDSC", "CHANNEL_CHANNEL_LVLDSC",
"CHANNEL_TOTAL_CHAN_LVLDSC", "CHANNEL_PRIMARY_PRNT") AS
  SELECT
"CHANNEL","CHANNEL_LEVEL","PRIMARY_SORT_ORDER","CHANNEL_SDSC","CHANNEL_LDSC","CHANNEL_CHANNEL_LVLDSC",
"CHANNEL_TOTAL_CHAN_LVLDSC","CHANNEL_PRIMARY_PRNT"
FROM table(OLAP_TABLE ('GLOBAL.GLOBAL duration session',
'', '', '&(CHANNEL_LIMITMAP)'))
MODEL
  DIMENSION BY (CHANNEL)
   MEASURES (
      CHANNEL_LEVEL,
 CHANNEL_SDSC,
      CHANNEL_LDSC,
      CHANNEL_CHANNEL_LVLDSC,
      CHANNEL_TOTAL_CHAN_LVLDSC,
      CHANNEL_PRIMARY_PRNT
    )  RULES UPDATE SEQUENTIAL ORDER();
```

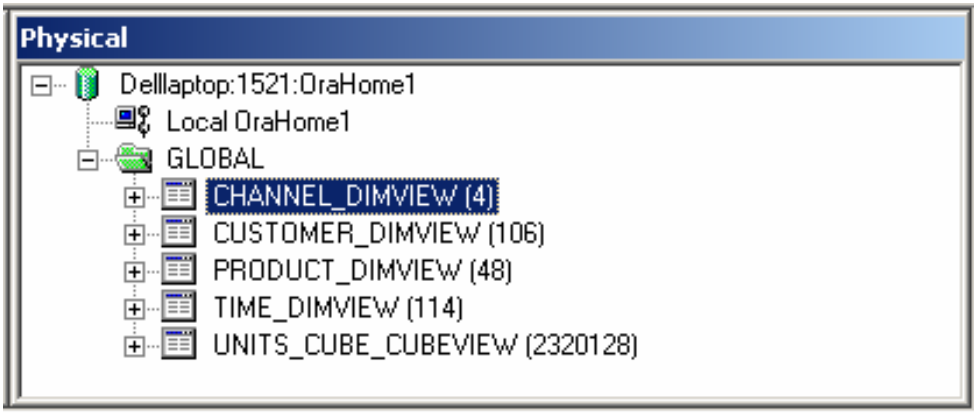
The wizard also created an object called a LIMITMAP in the Analytic Workspace. The LIMITMAP provides the information that the OLAP_TABLE requires in order to convert the multi-dimensional data into a relational view.

```
DIMENSION CHANNEL FROM CHANNEL WITH HIERARCHY
CHANNEL_PRIMARY_PRNT FROM
CHANNEL_PARENTREL(CHANNEL_HIERLIST 'PRIMARY')
INHIERARCHY CHANNEL_INHIER
FAMILYREL CHANNEL_TOTAL_CHAN_LVLDSC,
CHANNEL_CHANNEL_LVLDSC
FROM CHANNEL_FAMILYREL(CHANNEL_LEVELLIST 'TOTAL_CHANNEL'),
CHANNEL_FAMILYREL(CHANNEL_LEVELLIST 'CHANNEL')
LABEL CHANNEL_LONG_DESCRIPTION
ATTRIBUTE CHANNEL_LDSC FROM CHANNEL_LONG_DESCRIPTION
ATTRIBUTE CHANNEL_SDSC FROM CHANNEL_SHORT_DESCRIPTION
ATTRIBUTE CHANNEL_LEVEL FROM CHANNEL_LEVELREL
```

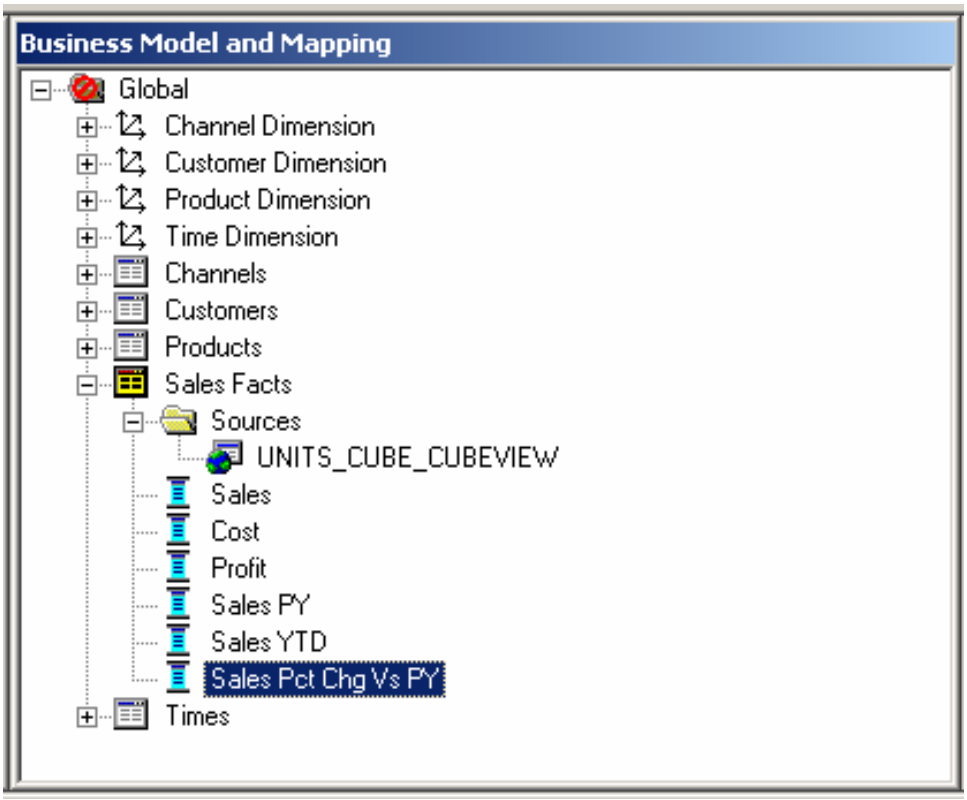
OBIEE Administrator

Now that we have a relational view of the multi-dimensional data, we can use the OBIEE Administrator tool (formerly known at the Siebel Administrator) to configure that view for our use with the OBIEE analysis environment. The tool has three sections.

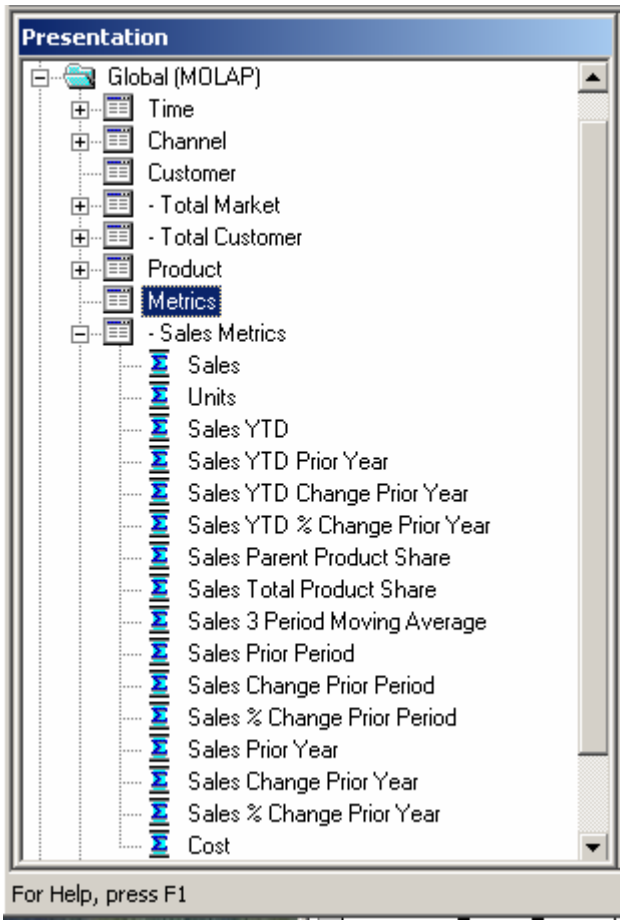
The *Physical Layer* is where we point to the relational tables and views that we want to use as our data sources. In this example, the Physical Layer contains references to the OLAP_TABLEs that were created by the wizard in AWM.



The *Business Model Layer* is where we convert the objects in the physical layer into objects that are meaningful for our business. For example, a relational column might be entitled 'REV', while we want to present it to our users as 'Sales Revenue'. The Business Model Layer is where that happens.



The *Presentation Layer* is where we decide and control what pieces of the Business Model our users can see. There may be objects such as calculations that are important for preparing the Business Model, but which the users need not see. Whatever objects the users are eligible to see will be placed into the Presentation Layer.



OBIEE Analytics

After the Physical Layer, Business Layer, and Presentation layer are created, the results can be viewed in the OBIEE analysis environment, OBIEE Analytics.

Title

Mouse Pad Sales

Table

Channel	Item	Year	Quarter	Sales	Sales Prior Year	Sales YTD	Sales % Change Prior Year
Catalog	Mouse Pad	2003	Q1-03	3,784	4,039	3,784	-6.311%
			Q2-03	3,934	4,122	7,718	-4.555%
			Q3-03	4,019	4,089	11,737	-1.717%
			Q4-03	4,493	3,945	16,230	13.900%
		2004	Q1-04	3,842	3,784	3,842	1.550%
			Q2-04	3,754	3,934	7,596	-4.582%
			Q3-04	1,250	4,019	8,846	-68.900%
Direct Sales	Mouse Pad	2003	Q1-03	662	657	662	0.821%
			Q2-03	714	711	1,376	0.488%
			Q3-03	709	647	2,086	9.682%
			Q4-03	731	624	2,817	17.237%
		2004	Q1-04	804	662	804	21.485%
			Q2-04	773	714	1,578	8.304%
			Q3-04	298	709	1,875	-58.059%

Going Further with Oracle OLAP

Once you have created your base measures you can start analyzing your data. But this is just the beginning. There is a rich OLAP DML language specifically designed for manipulating multi-dimensional data. This language has forecasting, regression, modeling, aggregation, allocation, and many other power features built in.

Creating Calculated Measures

You can create calculated measures that are presented to the end application as fully-solved data. Once the rules for calculating these measures are defined, a user does not have to worry about how the data is calculated; the data is presented as fully-solved. Examples of calculated measures that you can create include moving averages, change from year ago, moving totals, share calculations and much more. The demonstration located at www.vlamis.com/demo contains a measure, for example, that computes a "Reallocated Sales" and "Reallocated Profit" calculation that shows the projected results of discontinuing two products.

Summary

Oracle is supporting two complementary platforms for multi-dimensional data analysis. For those users who need the functionality of a true drill-down and drill-up, and a full-bodied crosstab with the ability to drag and rotate dimensionally, Discoverer for OLAP is the tool of choice. For those who prefer the look, feel, and intuitive functionality of OBIEE, that platform is available as well. Either environment can now take advantage of the same data, stored in Oracle's multi-dimensional data store, the Analytic Workspace.

Getting More Information

There is much more information on Oracle OLAP on Oracle's web site. You will find most of the information linked to from the page http://www.oracle.com/technology/products/bi/olap/olap.html. For information on Oracle Business Intelligence, see http://www.oracle.com/bi. The demonstration shown at the conference is available at www.vlamis.com/demo.