USING MULTIDIMENSIONAL DATA SOURCES WITH ORACLE BIEE+

Chris Claterbos, Vlamis Software Solutions, Inc.
claterbos@vlamis.com

INTRODUCTION

Oracle's integration of Business Intelligence Enterprise Edition into the Oracle product family provides exciting opportunities to extend the use of Oracle's multidimensional OLAP cubes to include Oracle's customers who have adopted the BIEE environment. The new 11g version of Oracle Business Intelligence (OBIEE) coming out soon will be enhanced to take advantage OLAP data sources. BIEE users can harness the power of Oracle's multidimensional data engine and features, while still retaining the user-friendly BIEE interface.

This presentation will demonstrate mapping both Oracle OLAP 10g and OLAP 11g sources using the OBIEE Administration tool. There will also be a brief overview of the upcoming new features of Answers and BI Publisher using the OLAP data sources.

OVERVIEW

The ability to use either of the Oracle OLAP data stores as a source for Oracle BI Administrator gives you the opportunity to expose your OLAP data to a broad user audience with OBIEE Dashboards and Answers. When these OLAP data sources are is used as an OBIEE data source, you have the capability to combine OLAP sources with other OBIEE Server supported data sources, and present relevant information to your users in one place, from multiple sources. The new OBIEE+ 11g will support hierarchical dimensional views which allows us to fully utilize more OLAP analytical capabilities.

OVERVIEW OF AWM AND THE ORACLE MULTI-DIMENSIONAL ENVIRONMENT

The OLAP option to Oracle 10g and 11g provide companies with the ability to perform multi-dimensional analysis completely within the context of the Oracle Database. Oracle presents OLAP as a central component of the data warehouse, rather than as an add-on to the database. Oracle's implementation of multi-dimensional OLAP data is highly scalable, partitionable, and, as an integral part of the Oracle database, is included in all of the normal backup and maintenance procedures in the database.

Oracle's OLAP option is implemented through the use of Oracle Analytic Workspaces (AWs), which are created using Oracle's Analytic Workspace Manager (AWM) or Oracle Warehouse Builder (OWB).

Using AWM, we can define the dimensions of our data, and also define multi-dimensional "cubes" of data for analysis. Those cubes can contain stored data objects, as well as computed measures that are calculated "on the fly". Data stored in the cubes is typically read in from relational tables or views, often in a non-aggregated state, and is frequently aggregated using the OLAP calculation engine.

INTEGRATION OF OBIEE AND AWM

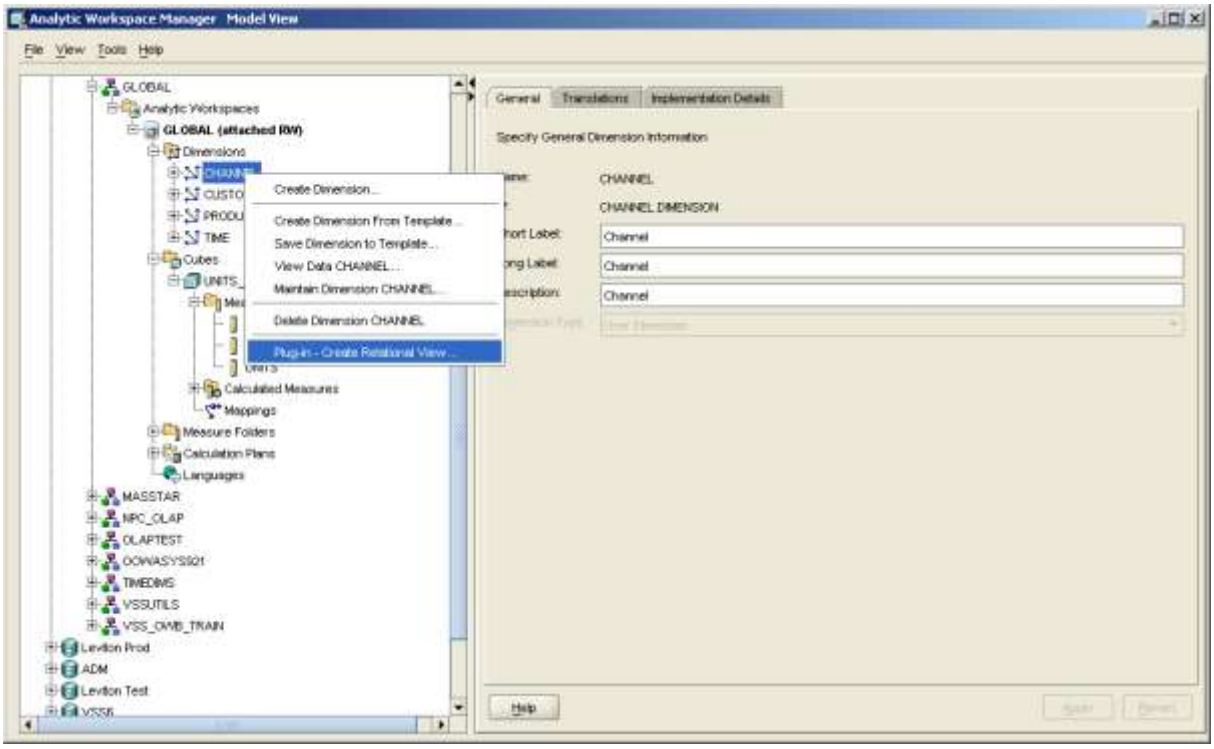
In the OBIEE environment, we see the same numbers as we see in the AWM/Discoverer environment. And in fact, the two environments are looking at the same data objects. Not replications, not copies, but the same physically stored data objects. So, how do we integrate the two environments? How can we use 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 10g Database engine, Oracle 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. In Oracle 11g, this functionality is completely embedded in the relational environment, and developers no longer need to manually create the OLAP_TABLE function in the database.

Oracle simplified the creation of the OLAP_TABLE in 10g 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
```

ORACLE 11G AND THE AWM OBIEE PLUG-IN

Oracle has produced an OBIEE Plug-in for AWM to help facilitate using OLAP 11g data in OBIEE. The OBIEE Plug-in for AWM allows you to create the Physical Database, Business Model and Mapping and Presentation Catalog layers of an OBIEE repository that can be used to query an Oracle11*g* OLAP cube using SQL. The repositories created by the OBIEE Plug-in for AWM allow OBIEE to query all data in the cube, including summary level data and measures calculations. The OBIEE Plug-in for AWM does not alter the definition or content of your cubes or dimensions. The OBIEE Plug-in for AWM reads the definitions of one or more cubes in a single analytic workspace (as well as the dimensions of the cubes) from the Oracle Data Dictionary and generates OBIEE UDML code that be used by the OBIEE Administrator. The OBIEE Plug-in for AWM also generates a security filter for each cube. This security filter is used to force joins between dimension views and cube views, even when the dimension is not part of the query. This is necessary to generate optimal SQL to the cube.

OBIEE ADMINISTRATION

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

ORACLE OLAP 10G SOURCE

If the data source is the 10g OLAP cubes we will first have to import the views using the conventional import tools for relational objects.

A detailed explanation of the steps necessary to configure OBIEE to access the OLAP 10g cubes is presented in a previous paper located on the Vlamis Software Solutions website. Please download using the following link: http://www.vlamis.com/Papers/collab2008-paper4.pdf

This presentation will not present any of these details but will focus on using OLAP 11g cubes.

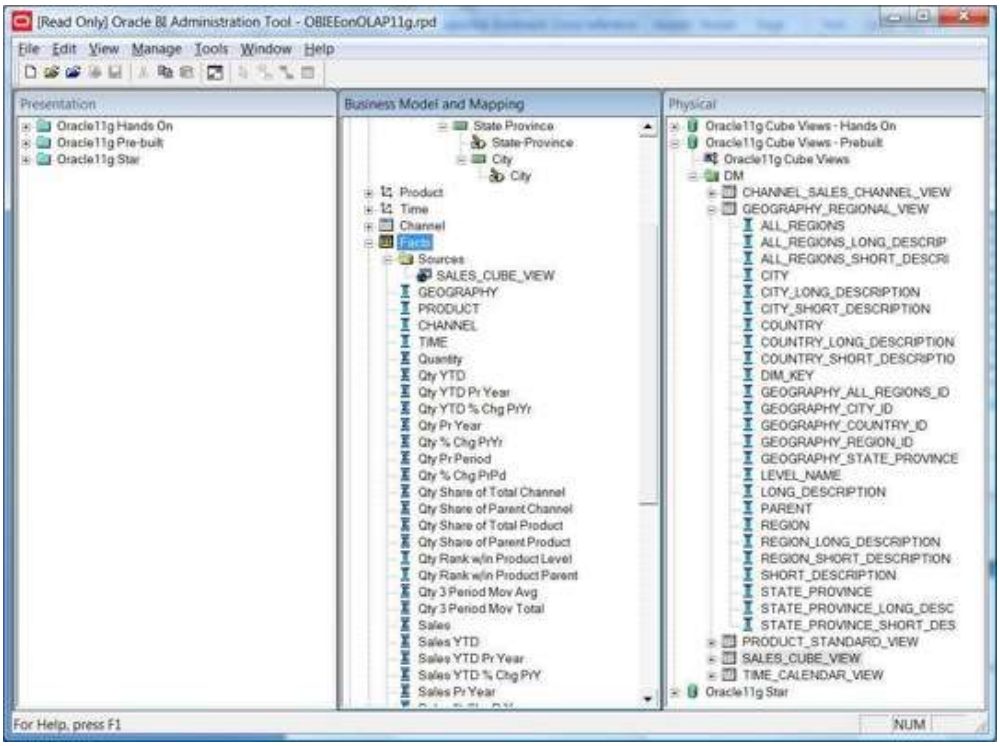


Figure 1 OLAP Model after Definition.

ORACLE OLAP 11G SOURCE

The first step is to take the file created by the AWM OBIEE plug-in and run OBIEE's nqUDMLExec utility (nqUDMLExec.exe) to import the file into an existing repository file or to create a new repository file from the file. nqUDMLExec.exe is located in the ...\server\Bin directory of the OBIEE installation.

Arguments to nqUDMLExec are described below:

nQUDMLExec [-U [userid]] [-P [password]] -I input_script_pathname [-B base_repository_pathname] -O output_repository_pathname [-8]

Where:

userid is the OBIEE administrator user id. The userid argument is required only if the base_repository_pathname argument is used.

password is the OBIEE administrator password. The password argument is required only if the userid argument is used.

input_script_pathname is the name of the metadata file.

base_repository_pathname is the name of a new or existing repository file.

output_repository_pathname (optional) is the name of a new or existing repository file.

-8 indicates a UTL-8 character set.

If output_repository_pathname is not provided, the cube(s) are added to the repository in the base_repository_pathname argument. If output_repository_pathname is provided, a repository file is created using the output_repository_pathname which is the union of the base_repository_pathname and the input_script_pathname.

Examples:

The following example creates the new repository file sales_cube.rpd from the export file sales_cube_obiee.txt.

```
C:> nqudmlexec -I sales_cube_obiee.txt -O sales_cube.rpd
```

The next example creates the new repository file sales_and_units.rpd from the repository file sales_cube.rpd and the export file units_cube_obiee.txt.

```
C:> nqudmlexec -U administrator -P admin_password -I units_cube_obiee.txt -B sales_cube.rpd -
O sales_and_units.rpd
```

At this point, the cube can be queried. It might be useful to review the OBIEE metadata and make a few adjustments to accommodate your preferences. For example, you might want to:

- Review the columns representing attributes and decide if they are useful to your users. Some attributes, for example those used for time series calculations or sorting, might not be useful to users. Consider removing them from the Presentation Catalog.
- Set sort order columns. (Time dimensions are automatically set to sort by the End Date attribute).
- Set aggregation rules. If there are attribute columns in the Presentation Catalog layer you can enable GROUP BY for attribute reporting (for example, break out members of the Product dimension by Color and Size attributes) by applying aggregation operators to measure columns. Give this careful thought, however. It might make sense to allow SUM … GROUP BY of a measure such as Sales but it probably doesn't make sense to aggregate measures that are percentages or rankings.
- Define Additional Calculated Measures

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.

PRESENTATION AND SECURITY

The last steps need to be setting up the presentation layer and security. These steps are pretty much the same as before with the added ability to turn on or off hierarchical viewing of the dimensions. In order for OBIEE to create the correct style of SQL for cube and dimension views it is necessary to create an OBIEE security filter on the cube view. (This security filter forces joins between dimension and cube views in the case where a dimension is not represented in a query.) This filter is created for you using the AWM OBIEE plug-in.

Now for the real fun! Save the model and start up OBIEE Web and we will see what OBIEE+ can do.

ANSWERS AND DASHBOARDS

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

Title

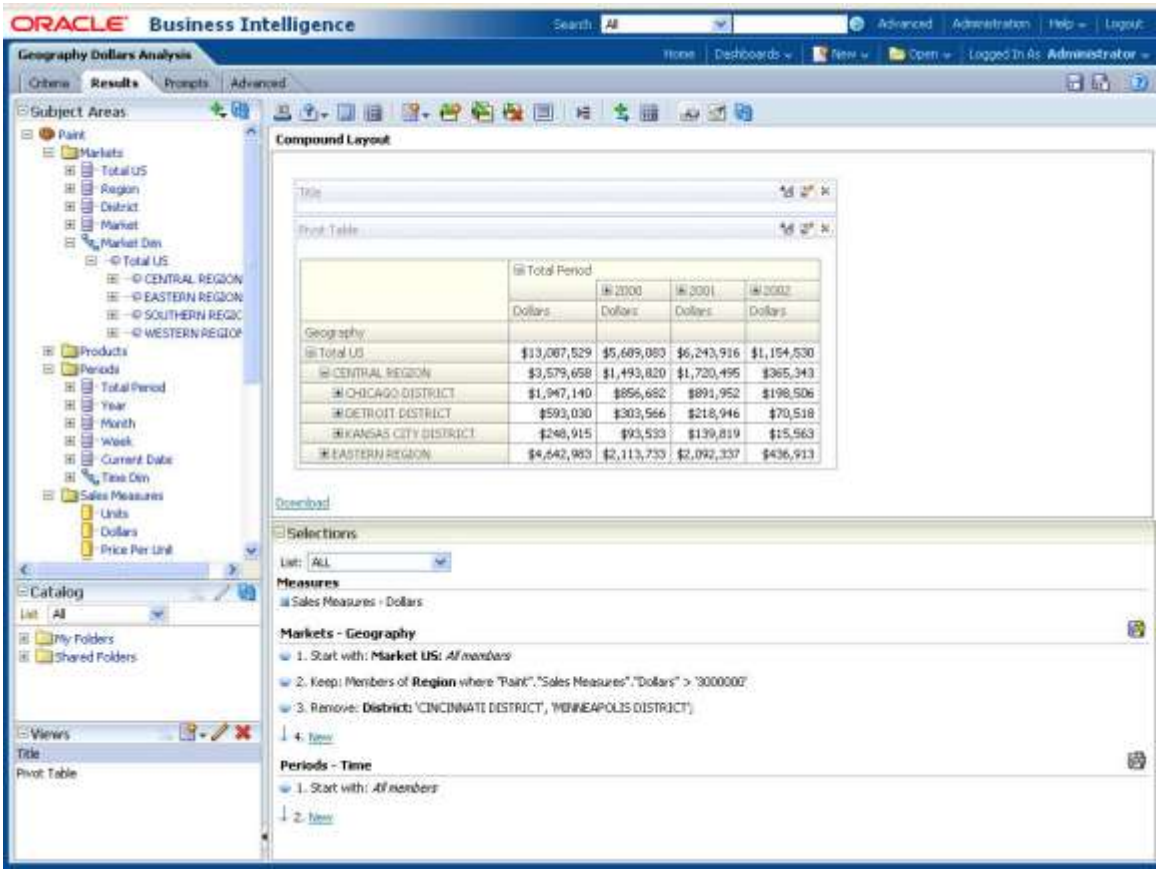
Mouse Pad Sales

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%

OBIEE 11G COMING SOON

Oracle OBIEE 11g is currently in Beta and will be released as soon as possible. There are some significant changes coming to the web based analytics.



As you can see Answers is no longer in the menu, the main menu includes the Home page that includes links to all the most recent activity and allows for editing existing queries or dashboards, browsing the catalog or create new objects.

Of particular note for OLAP analytics is the ability to produce a variety of Hierarchical reports. Such things as Hierarchical drilling, pivoting, creation of custom groups are all possible now. Here are a few screens showing what is now possible.

Geo Time Revenue

Time run: 2/11/2010 3:54:16 PM

	Total						
		2007					2008
			2007 Q1	2007 Q2	2007 Q3	2007 Q4	
	Revenue	Revenue	Revenue	Revenue	Revenue	Revenue	Revenue
Market							
Total	24,903,044	11,371,280	952,206	3,387,276	4,987,611	2,044,186	13,531,764
East	13,824,546	6,400,147	533,291	1,999,585	2,696,523	1,170,748	7,424,400
Atlantic	12,558,896	5,766,713	476,970	1,854,538	2,378,789	1,056,416	6,792,183
Carolinas	3,855,796	1,782,721	117,463	605,352	718,801	341,105	2,073,076
Research Triangle	3,855,796	1,782,721	117,463	605,352	718,801	341,105	2,073,076
Greater DC	6,170,545	2,807,740	302,900	895,849	1,095,272	513,718	3,362,805
Washington	6,170,545	2,807,740	302,900	895,849	1,095,272	513,718	3,362,805
Greater NYC	2,532,555	1,176,252	56,606	353,337	564,715	201,593	1,356,303
NYC	2,532,555	1,176,252	56,606	353,337	564,715	201,593	1,356,303
Northeast	1,265,650	633,434	56,321	145,047	317,734	114,332	632,217
South	10,591,662	4,814,082	395,795	1,350,104	2,218,146	850,037	5,777,580
Web Direct	486,836	157,052	23,120	37,588	72,942	23,402	329,784

Edit - Refresh

This is just a small taste of what is new in this release of OBIEE+.

CONCLUSION

Oracle's direction clearly supports the use of multi-dimensional data in BIEE analysis. Oracle continues to enhance and update the BIEE toolset, and will be including more robust functionality with some "Discoverer-like" capabilities in the near future. It is now easier than ever to integrate the Oracle OLAP data sources and we now have a toolset that can take full advantage of the OLAP analytical power provided by these engines.