BUILDING CUBES AND ANALYZING DATA USING ORACLE OLAP 11G

Chris Claterbos, 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 (literally today, March 1, 2007 as I write this!) of Oracle's pending acquisition of Hyperion, including the Essbase multi-dimensional database product.

Given this upheaval, it's very difficult to know exactly where to concentrate a hands-on session that is six weeks away.

The presentation / hands-on session will focus on how to prepare multi-dimensional cubes in Oracle OLAP using a front end. This will be a mix of using Analytic Workspace Manager to create cubes, 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), Oracle Discoverer for OLAP (Oracle BI SE) and Oracle BI EE.

There may be an opportunity to share information that was previously disclosed 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

In Oracle OLAP, Oracle has separated the physical storage of data warehouse data from the logical analysis of OLAP applications. What's the magic that makes this work? How does this impact performance? What operations are precluded by various design decisions? Why would you want to use Oracle OLAP or BI Beans instead of simply using regular relational views or tables?

This presentation will describe these components in a "behind the scenes" tour of Oracle OLAP. Armed with information such as "What is an Analytic Workspace", and "What metadata is required to support BI Beans?", will help a DBA prepare a database and design schemas for use by an OLAP application and help an application developer understand what options are possible and how to gain maximum value from a data warehouse or data mart. Also included will be the metadata structures used ("AW Standard Form") to describe AW data and how this results in AWs being automatically enabled for BI Beans in OLAP 11g.

The OLAP option to Oracle Enterprise Edition is at the heart of Oracle's Business Intelligence offerings. It provides the multi-dimensional model that all of Oracle's BI offerings require. It also serves as an efficient "calculation engine" that serves data to client applications with a multi-dimensional model.

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 11g 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

In Oracle OLAP 9i, multidimensional metadata was stored in a series of relational tables in an OLAPSYS schema. This metadata was created in Oracle Enterprise Manager. With Oracle OLAP 11g, the 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.

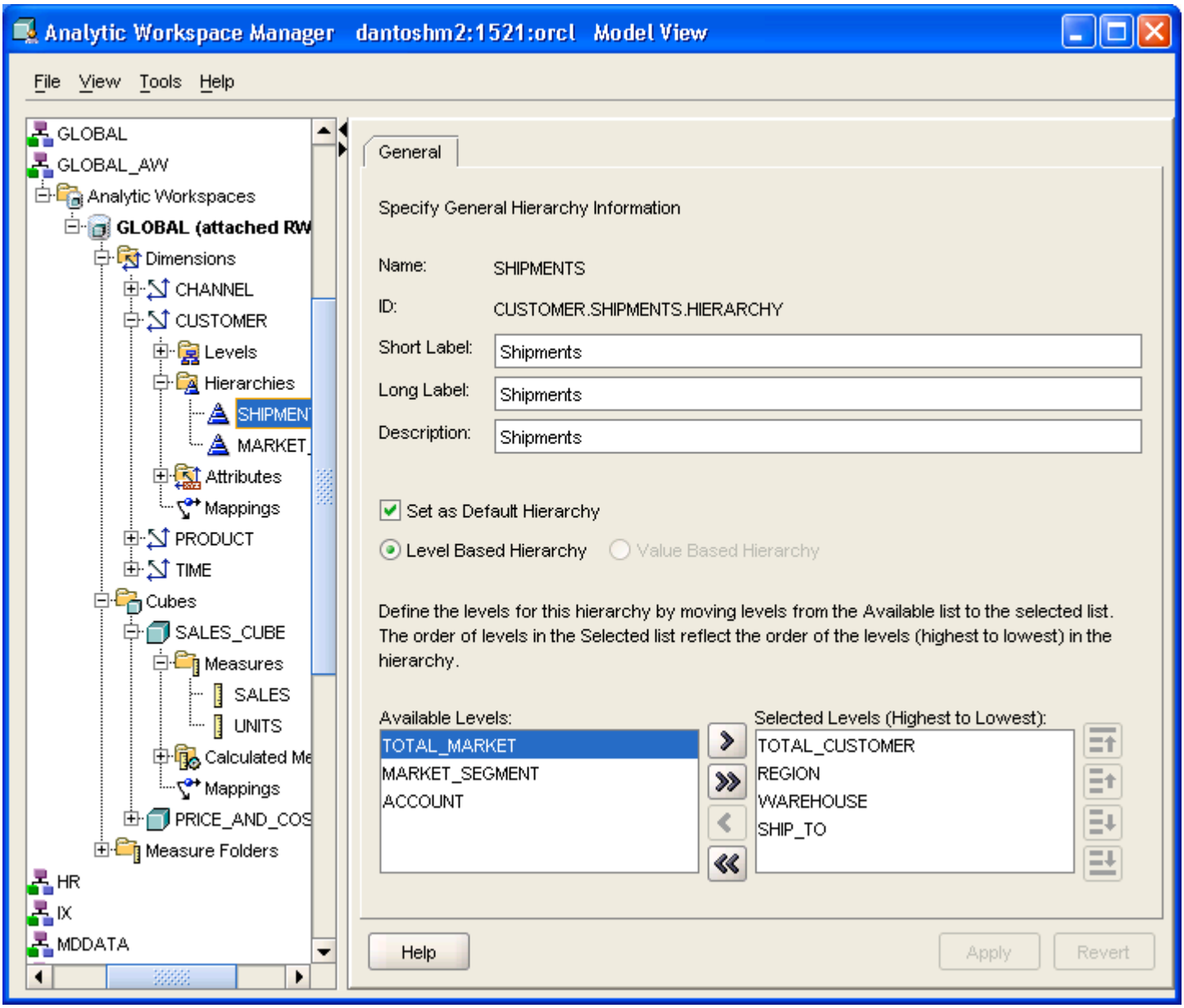

Figure 1 – AWM Screen for Creating Customer Hierarchy

This metadata is now stored directly in the analytic workspace. When you create your dimensions, AWM automatically creates the metadata necessary for BI Beans to present this data to a user. Figure 2 shows two screenshots that show some of the metadata used by BI Beans to control how data about customers is shown to the user. Notice that AWM automatically fills in the metadata for you.

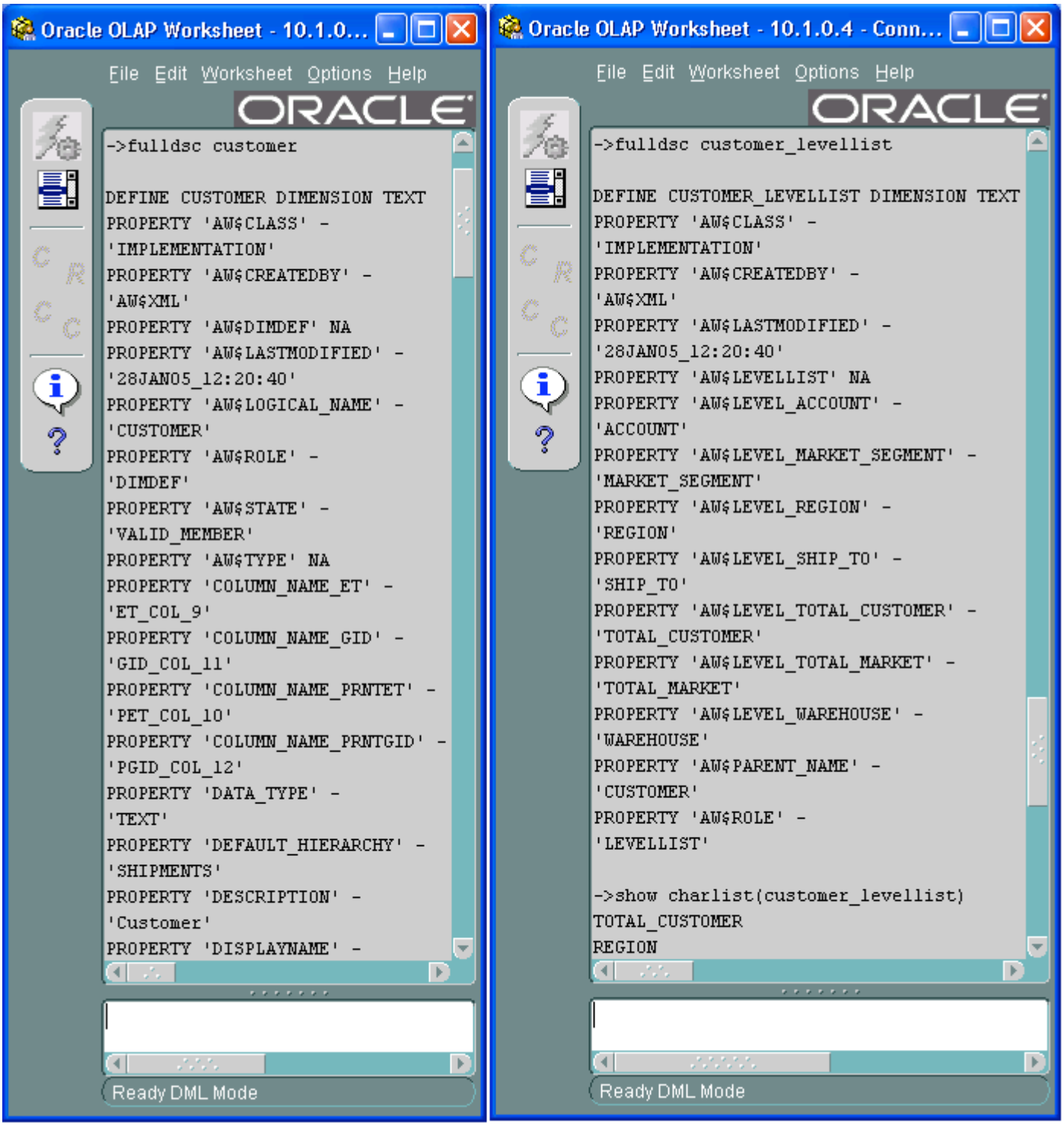

Figure 2 – Metadata Created by AWM

AW Standard Form

An analytic workspace is said to be in "Standard Form" when it contains certain metadata. While you can create AWs with your own custom code, using AWM ensures that the AW you create adheres to the standard form specification. This standard form specification allows tools such as BI Beans to know about the data, such as what the list of user-visible dimensions is, how to obtain the list of levels for the customer dimension, and which measures should be presented to users. These "rules" are fully documented in Appendix A of the Oracle document Oracle OLAP Application Developer's Guide.

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 3 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.

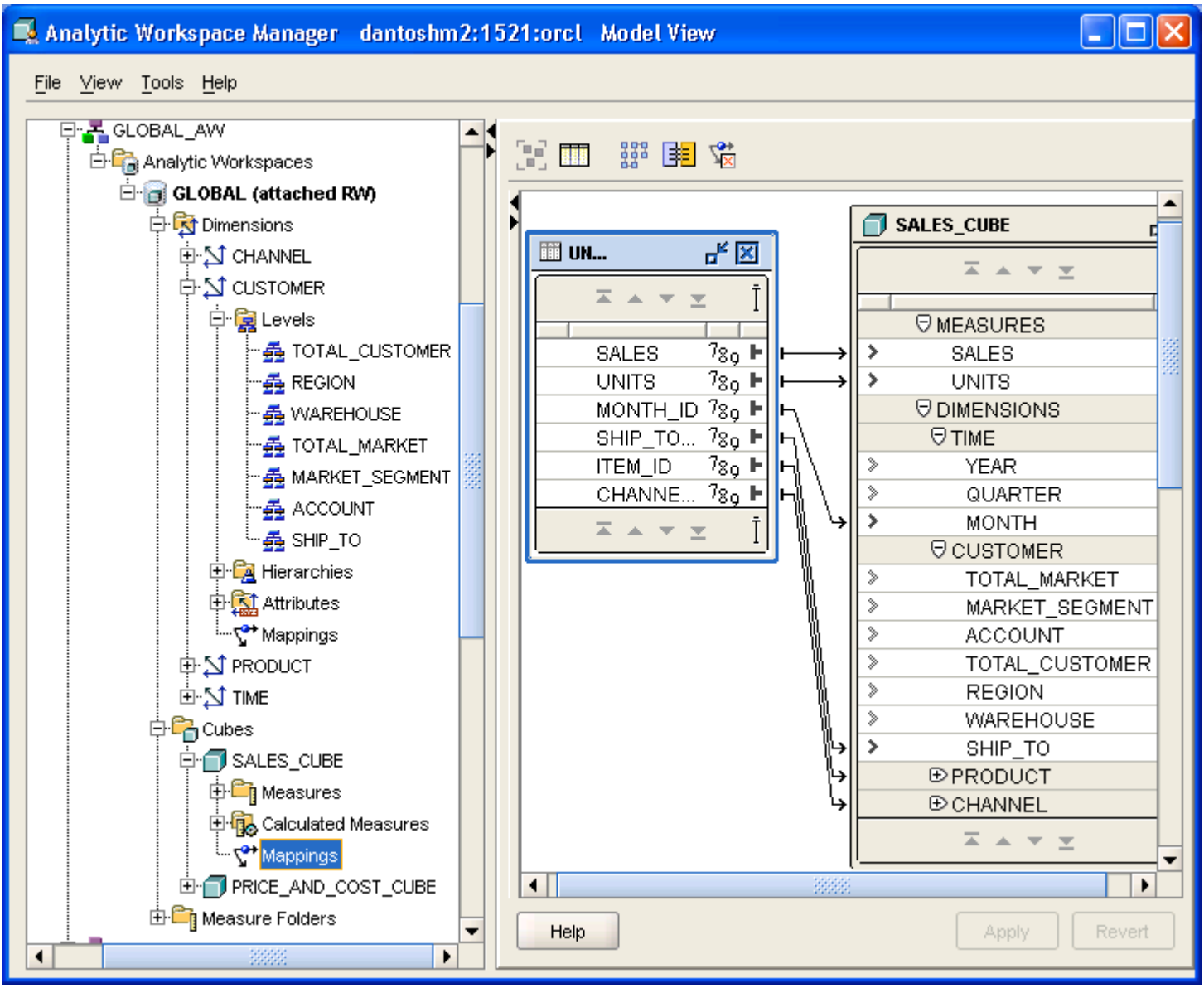

Figure 3 - Mapping Relational Sources to Cubes with AWM

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.

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 various client frontends, see the Oracle's Business Intelligence home page at http://www.oracle.com/technology/products/bi/index.html.