Exploiting The Power of Oracle Using Microsoft Excel

An Oracle White Paper
December 2004

ORACLE

Exploiting The Power of Oracle using Microsoft Excel

Exploiting The Power of Oracle using Microsoft Excel

EXECUTIVE OVERVIEW

To date many organisations have seen a mass proliferation of data extract programs and downloading of static data into Excel. This causes enormous problems, both technical and business related. The result of this process of continual download and manipulation is an ever expanding range of Excel spreadsheets. It is never clear from the Excel analysis how historical data based on numerous sources is handled. For example, what happens if the source systems are updated/restated as there is no link to the source data the end user is not alerted to the availability of refreshed data? As a result it is never clear which spreadsheet is the latest version. Most users create elaborate naming conventions to try and resolve this issue.

Over time as the number of spreadsheets increases, there is a tendency for the complexity to also increase. Users tend to create linked sheets where subsequent manual tracking of calculations via linked spreadsheets is extremely difficult. Also last minute changes are difficult to administer. It is generally accepted that many Excel users spend more time creating their data set rather than analysing their data and creating useful business information. Most Excel applications are based on static data that has to manually updated or updated via error prone macros. Most contain hundreds of complicated formulas that are difficult to understand and debug.

What is needed is the ability to view business data intelligently and directly within Excel, modify and enhance that data using Excel. The overall benefit is that Excel users would no longer work in isolation, they would always have access to the latest data, they would be working from a common set of business rules and calculations and their results could be shared with other Excel and Microsoft Office users. This has major benefits for other parts of the organization, especially for IT departments. IT departments are extremely reluctant to encourage the use of spreadsheets and resulting personal data marts. The growth of undocumented personal applications containing complex chained worksheets maintained by one key user can be quickly and easily resolved.

The remainder of this paper reviews the key components of the BI Spreadsheet Add-in and how users can create powerful data driven reports.

INTRODUCTION

There are two sections to this report. The first section reviews the architecture and key components of the Oracle BI Spreadsheet Add-in. It will focus on the OLAP Query and Calculation Builder wizards that are key components of Oracle's various OLAP tools and products, such as: Oracle BI Discoverer, Oracle BI Beans, and Oracle Application Server Reports.

The second part of this white paper shows how Excel users can create and manage OLAP queries directly from Excel and also incorporate standard Excel features.

ORACLE BUSINESS INTELLIGENCE SPREADSHEET ADD-IN

The Oracle BI Spreadsheet Add-In enables end-users to display and navigate Oracle OLAP data directly from within Excel. The users can treat the Oracle OLAP data as regular Excel data, for example create formulas and graphs, enabling them to combine the powerful analytic capabilities of Oracle OLAP with standard Excel functionality. Here, Excel acts as an intelligent front end connected directly to an Oracle OLAP enabled database.

This BI Spreadsheet Add-in is installed similarly to other Excel Add-ins - but the real application power and flexibility remains contained within the Oracle server and requires no local maintenance. The add-in does not stop Excel being used in the normal way, and it is usually possible to have several add-ins installed without interfering with each other. Most add-ins provide their own main menu entry and relevant, right-click context menus. Many also have one or more tool bars, which can be turned off when not required.

The main advantage of using Oracle OLAP for Excel is it allows users to create, manage and execute queries directly against the OLAP dimensions and measures using all the processing power of the database. This allows users to query extremely large database warehouse instances, which would normally be beyond the capabilities of Excel, due to its well-documented structural limitations. Once a user has built a query the data is presented within Excel as a normal spreadsheet, with additional controls for dimension paging. Since the OLAP query looks and feels exactly like a normal Excel spreadsheet the user is free to enhance and extend the query using the normal Excel business tools such as adding a chart, as can be seen below in figure 1.

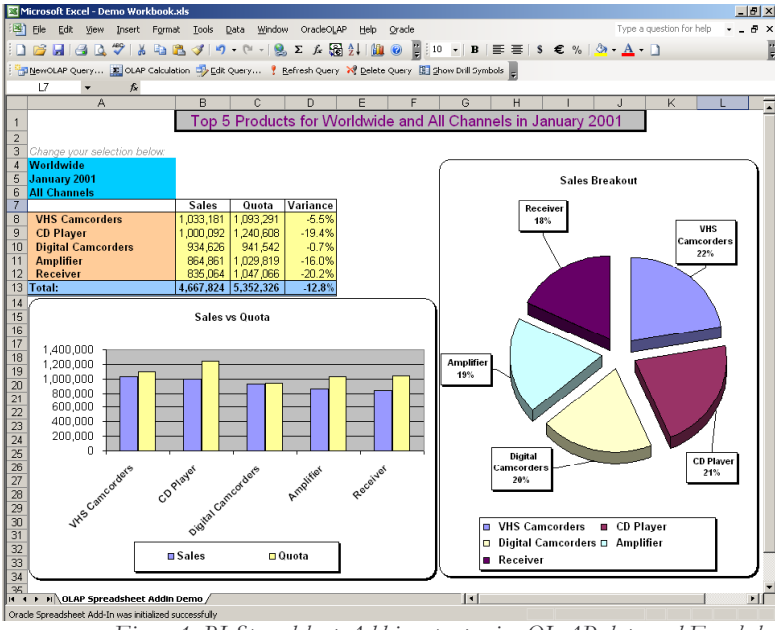


Figure 1: BI Spreadsheet Add-in report using OLAP data and Excel charts

Having created a complex report, users do not want to rebuild their queries each time new data is loaded into the warehouse. By using query conditions spreadsheets can become dynamic in nature, providing the user with the ability to continually refresh with the latest data without the need to rewrite the query. Using Oracle's OLAP Query Builder users can construct queries using business terms and definitions. Both calculations and queries are based around measures and their related dimensions. Using a wizard driven interface, users can select data from Oracle OLAP simply by choosing from a list of values, or by creating advanced selections, such as exceptions, top/ bottom or hierarchy-based queries. In addition, the user can create Oracle OLAP based calculations using a similar wizard.

The built-in Query Wizard is based on the Oracle Business Intelligence Beans Query Wizard. This wizard is used across all the Oracle OLAP products, ensuring a consistent interface with other Oracle Business Intelligence Beans based applications.

Core Components

Below is a diagram of the BI Spreadsheet Add-in architecture. The following sections of this document will explain each of these components.

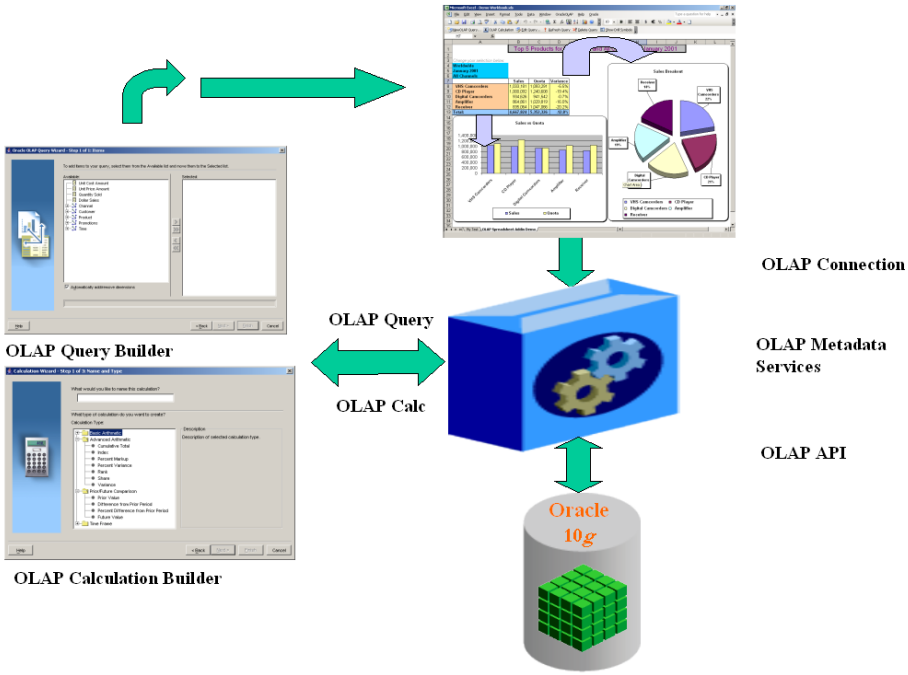


Figure 2: BI Spreadsheet Add-in Overview

OLAP Connection

From within Excel a connection to Oracle OLAP instance is created using the Oracle Java OLAP API. The connection uses the thin JDBC drivers; this means there is no SQLNet client side installation required. All the required JDBC files are bundled as part of the Add-in installation. The connection registers the Excel session with the OLAP Metadata services layer.

Due to the complexity of the warehouse design and the questions that end users ask, many interrelated lookup tables and fact tables may be used to satisfy even the simplest OLAP query. The optimized SQL that is required to satisfy the user's simple question may be many lines of code. The Oracle OLAP API enables a client application to specify complex multidimensional or relational style queries and calculations using metadata. Complex queries are simplified significantly by directly manipulating the metadata.

Query

Behind any Worksheet is a query. The query specifies exactly what the user is viewing in Excel: the measures (such as Sales and Costs), the dimensions (such as Product, Geography, Channel, Promotion and Time), the selections for each dimension (such as the last 6 months), and the layout of the dimensions (such as

Geography in the rows, Time in the Columns, and paging on Product, Channel and Promotion).

The query uses the Oracle OLAP API to retrieve and manipulate data to provide the advanced analytic capabilities that are offered in Oracle database. The Oracle OLAP API achieves this by generating highly tuned SQL to resolve the request from the query.

QueryBuilder

The QueryBuilder provides a simple user interface to define sophisticated queries. The user interface is patented and extensively tested for ease of use. The QueryBuilder is a very powerful tool that enables users to specify the following query properties without needing to know the underlying query language:

- Measures and dimensions – Chosen using a list tool which displays a hierarchical list.
- Dimension selections – Specified by defining exceptions, rankings, top/bottom selections, hierarchical selections (children and ancestors), text matching, and so on. For example, "Top 5 Products sold in each City".
- The layout of the dimensions within the presentation.

The QueryBuilder also makes it easy to reuse selections by allowing users to save and retrieve defined selections as favorites. Users often want to use a popular selection or query (such as "Top 5 Products") in multiple reports or presentations.

A key strength of the QueryBuilder is that the end-user does not need to understand a query language to define the query. Powerful queries are made simple by presenting the query definition in business terms, which end users can modify to meet their needs. For example, a default query definition can be "Show the Top 5 Products based on Sales". An end-user can then modify the word "Top" to "Bottom" in the definition to show the bottom 5 products based on sales.

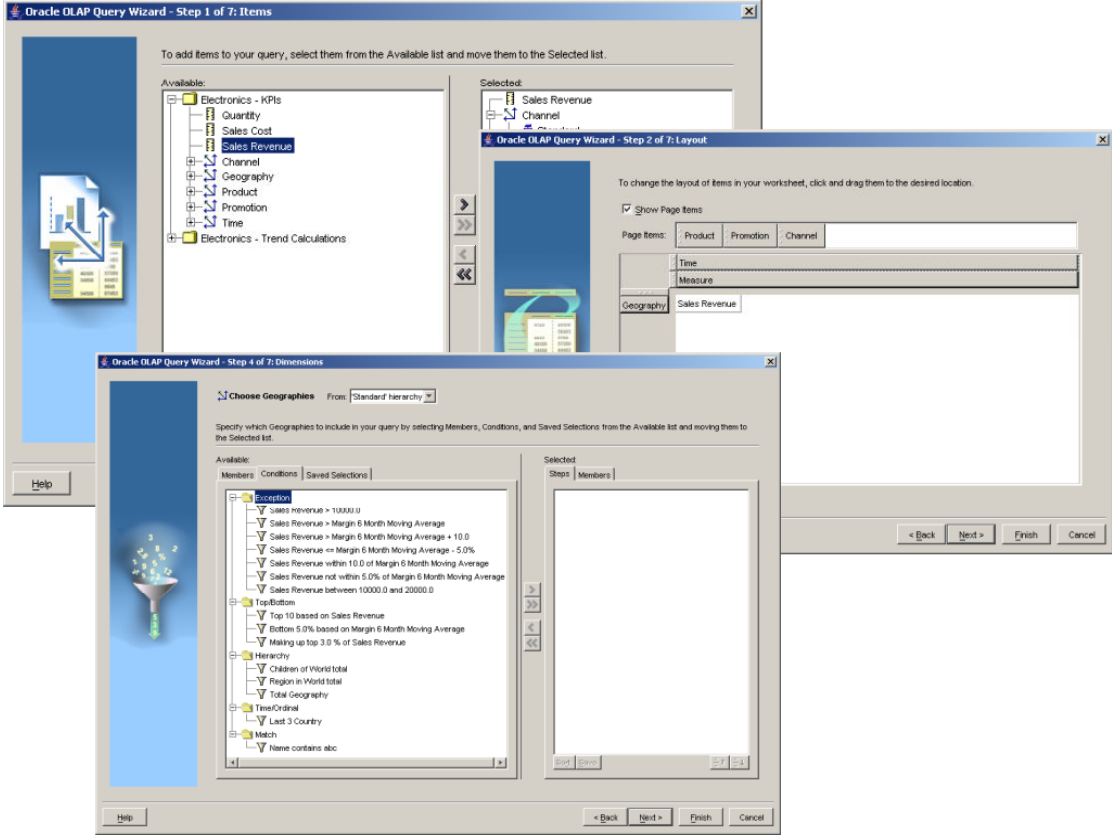


Figure 3: BI Spreadsheet Add-In Query Wizard

CalcBuilder

The CalcBuilder wizard provides a user interface to create derived calculations. Like the QueryBuilder, end-users define new metrics using templates, which eliminates the need to understand the underlying SQL. For example, an end-user might be interested in seeing Sales Growth compared to last year for a particular product line, even though Sales Growth has not been defined by the database administrator. The CalcBuilder enables the user to define this calculation. The calculation templates are organized by type. In our example, the end-user selects the appropriate Time Series template to define the new measure.

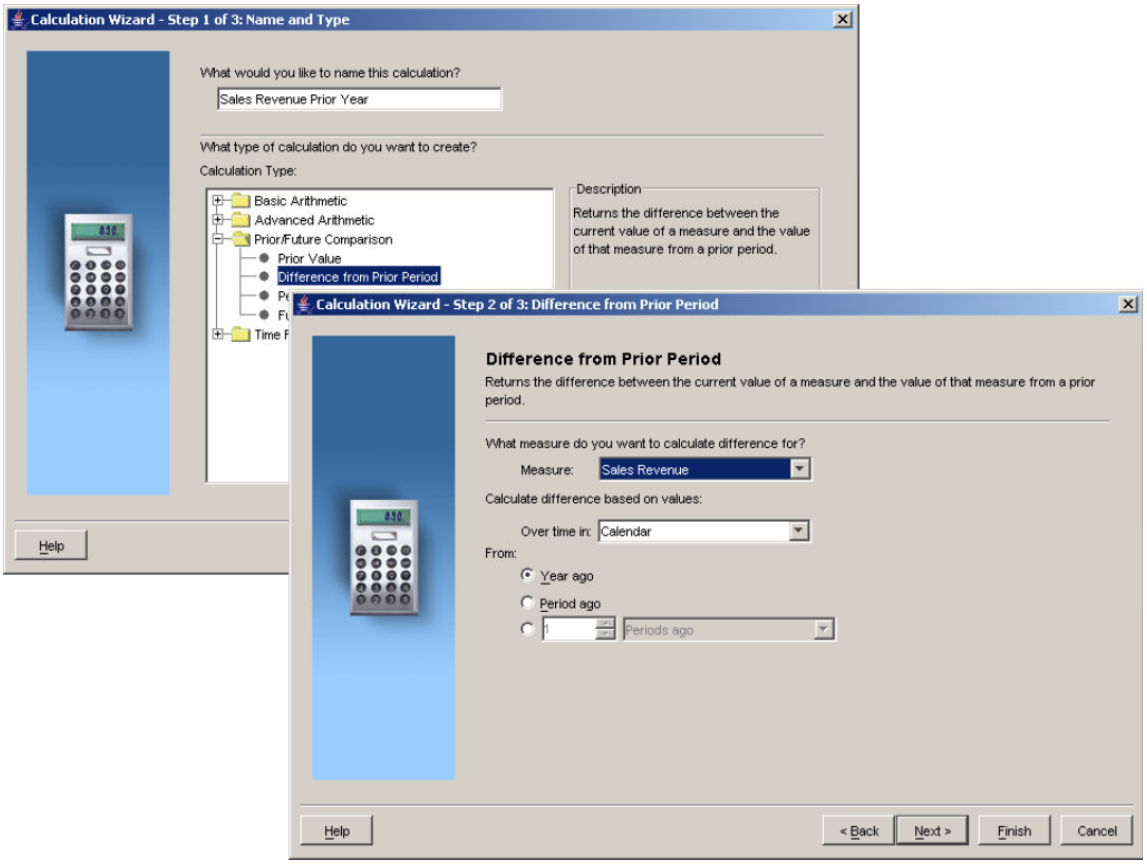


Figure 4: BI Spreadsheet Add-in Calculation Builder Wizard

OLAP ANALYSIS USING MICROSOFT EXCEL

The following worked example shows how to create and use the BI Spreadsheet Add-in. The aim of this example is to show how to use normal Excel features alongside Oracle OLAP data.

Sample Worksheet Overview

After installing the BI Spreadsheet Add-in a new menu is exposed within Excel.

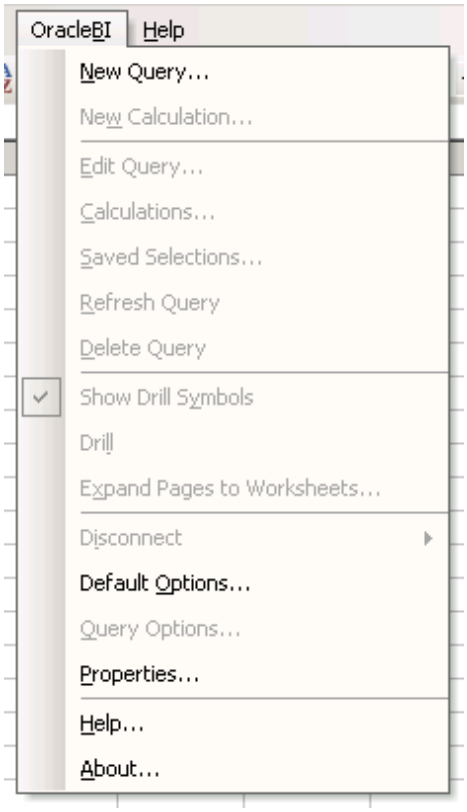


Figure 5: BI Spreadsheet Add-in Excel Menu

This menu controls access to the various Oracle OLAP options. This first step is to use the 'New Query' option to launch the OLAP query builder. This launches the connection dialog which allows the user to define the OLAP connection details

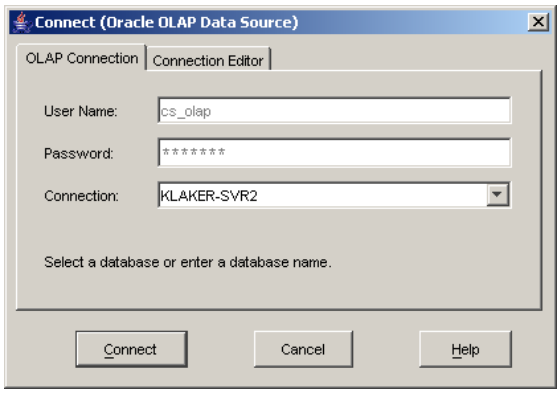


Figure 6: BI Spreadsheet Add-in Connection Dialog

Once the user has been authenticated the OLAP Query Builder is launched to guide the user through the selection of the required measures and dimensions.

Creating BI Queries using Excel

The following screen shots show how to create a report to show the Top 10 performing products at the subcategory level, for all Channels, for all Regions and all Promotions based on the Sales Revenue %Growth Based on Prior Year, in 2001. The Query wizard has a series of steps to guide the user. Step 1 is to select the required measures. For this report Sales Revenue and Margin % have been selected. By default all the related dimensions are automatically selected. This protects the user from having to understand the underlying database schema and how the various tables are joined.

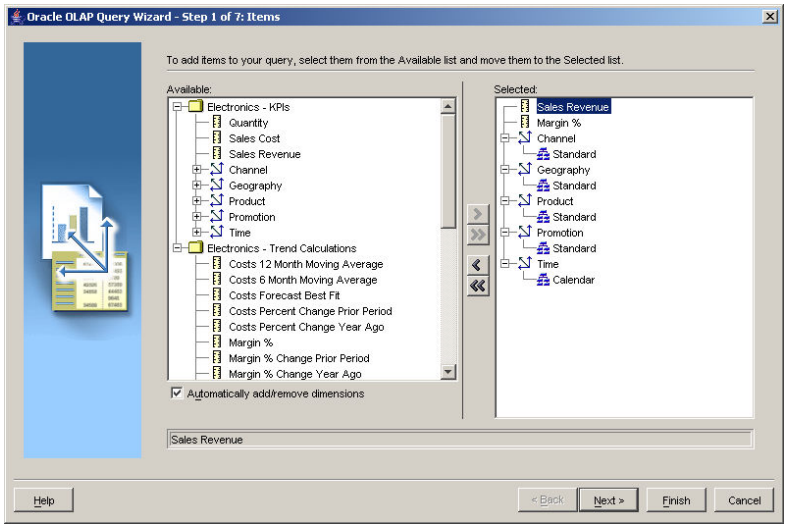


Figure 7: BI Spreadsheet Add-in Query Builder – Measure Selection

The next step is to determine the layout of the report. As can be seen below products are in the row edge and measure and time are shown in the column position. Geography, Promotion and Channel will be in the Page edge.

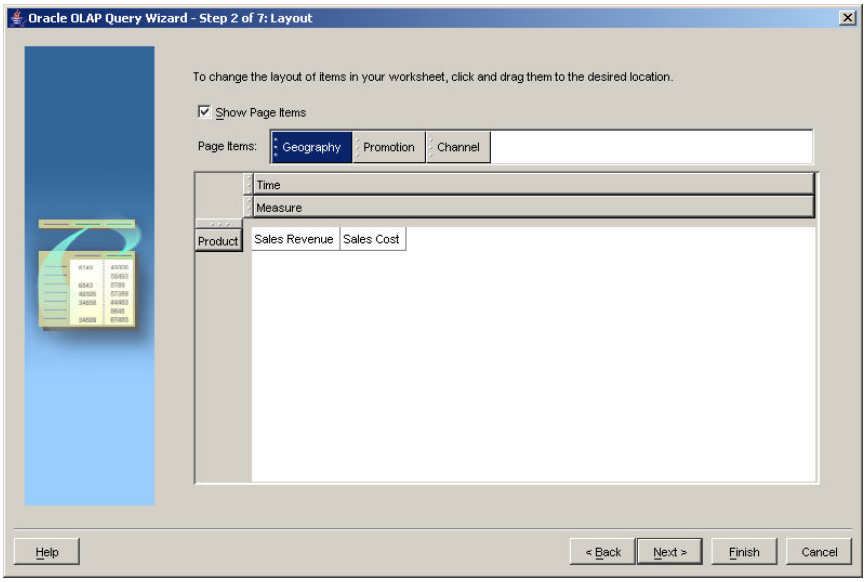


Figure 8: BI Spreadsheet Add-in Query Builder – Report Layout Screen

Now the user can determine the selections for each dimension. The selection process can be driven by simply using the mouse to manually select the required dimension values as shown below:

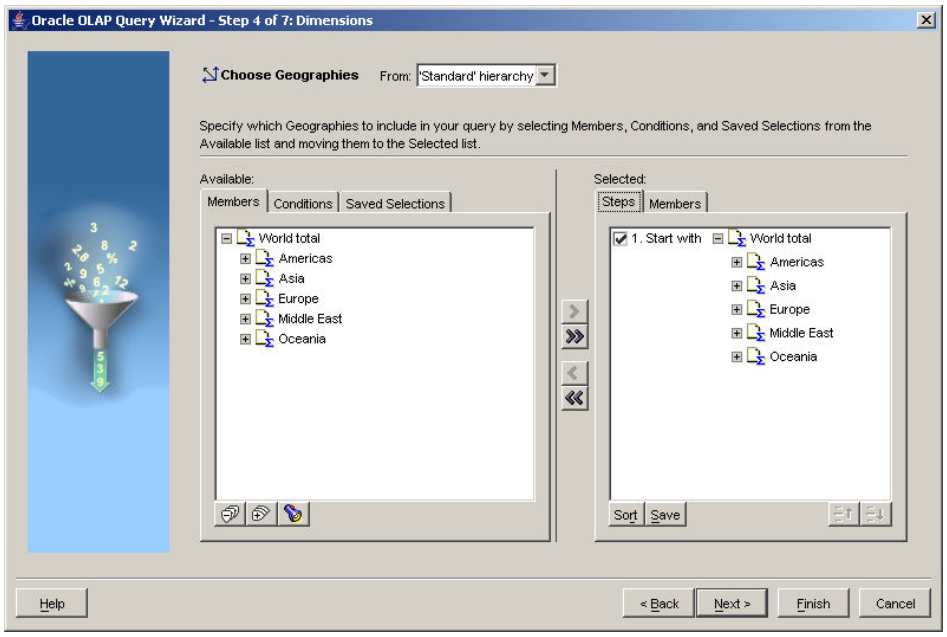


Figure 9: BI Spreadsheet Add-in Query Builder – Manual Dimension Member Selection

Manual selection is not very easy when a dimension contains a large number of members such as a product dimension. More typically a user will want to filter the list of values. The OLAP Query Builder provides a series of templates, which help the user to quickly and easily construct complex queries.

The following screen shots show how to create a selection to show the Top 10 performing products at the subcategory level, for all Channels, for all Regions and all Promotions based on the Sales Revenue %Growth Based on Prior Year, in 2001.

Each template contains hyperlinks that allow the user to modify the default value and extra dialogs help to show additional parameters for a query.

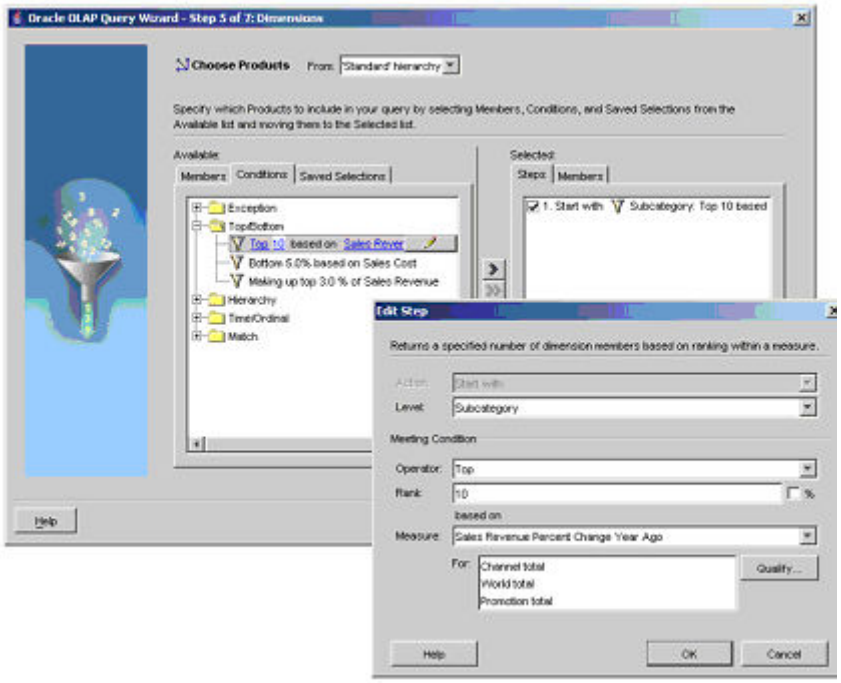


Figure 10: BI Spreadsheet Add-in Query Builder – Conditional Dimension Member Selection

Having completed all the dimension selections, the user can execute the query and retrieve the data into the worksheet. The report below shows the top 10 performing product categories. The product dimension itself contains many thousands of members, but only the top 10 product categories are returned by the query. The database does all the analysis and filtering which overcomes the inherent scalability issues associated with the use of Excel worksheets.

Microsoft Excel - Excel Demo.xls

File Edit View Insert Format Tools Data Window OracleOLAP Help Oracle

Type a question for help

NewOLAP Query... OLAP Calculation Edit Query... Refresh Query Delete Query Show Drill Symbols

B7

World total
Promotion total
Indirect

	+ 1998		+ 1999		+ 2000		+ 2001	
	Sales Revenue	Margin %	Sales Revenue	Margin %	Sales Revenue	Margin %	Sales Revenue	Margin %
+ Portable PCs	678,139	28	117,542	18	105,398	27	644,666	16
+ Monitors	716,377	26	267,628	17	351,862	32	869,909	20
+ Camcorders	148,512	37	189,915	17	39,728	20	552,303	16
+ Home Audio	156,213	22	308,402	17	559,496	14	583,700	18
+ Desktop PCs	328,693	14	154,078	16	28,295	17	497,542	15
+ Cameras	73,008	35	146,663	17	365,845	19	790,700	29
+ Printer Supplies	353,936	20	146,512	16	62,435	16	324,678	17
+ Memory	7,395	23	58,726	16	33,077	44	282,301	21
+ Game Consoles			59,692	16	23,721	11	299,359	12
+ Operating Systems			66,599	16	36,335	13	273,125	20

Figure 10: BI Spreadsheet Add-in Report showing Top 10 performing products at the subcategory level, for all Channels, for all Regions and all Promotions based on the Sales Revenue %Growth Based on Prior Year, in 2001.

Adding Excel features to the query

Once the query has been defined, the report can be extended by adding standard Excel features such as charts to enhance the visualization of the report. The BI Spreadsheet Add-in provides a powerful Calculation Builder wizard; see the previous section of this document. These calculations can also be enhanced using normal Excel functions and calculations. To insert a sub-total in the current worksheet the user simply has to use the Excel function wizard. For the purposes of this demo a sum calculation will be used for the Sales Revenue column and an average calculation will be used for the margin.

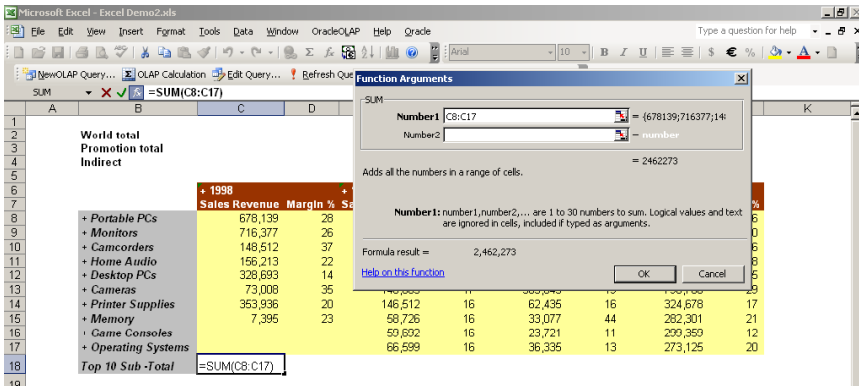


Figure 11:Excel SUM function wizard

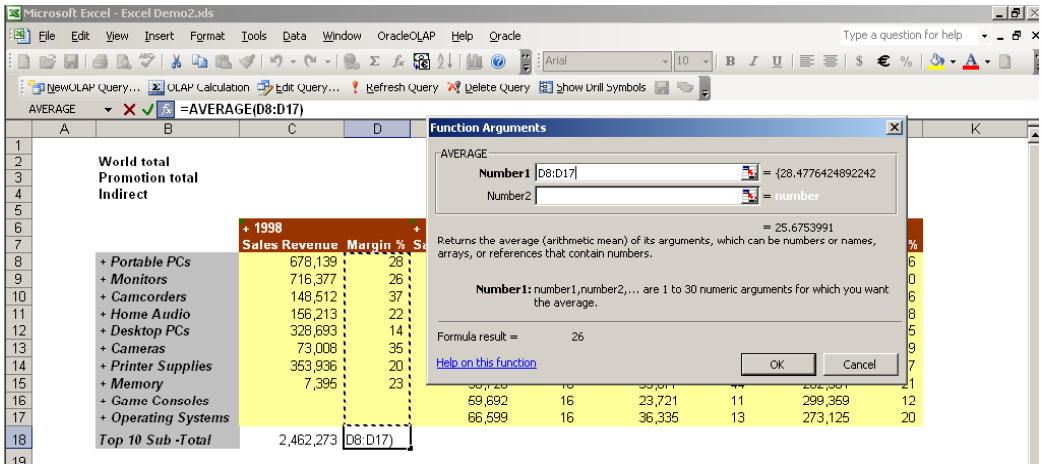


Figure 12:Excel AVERAGE function wizard

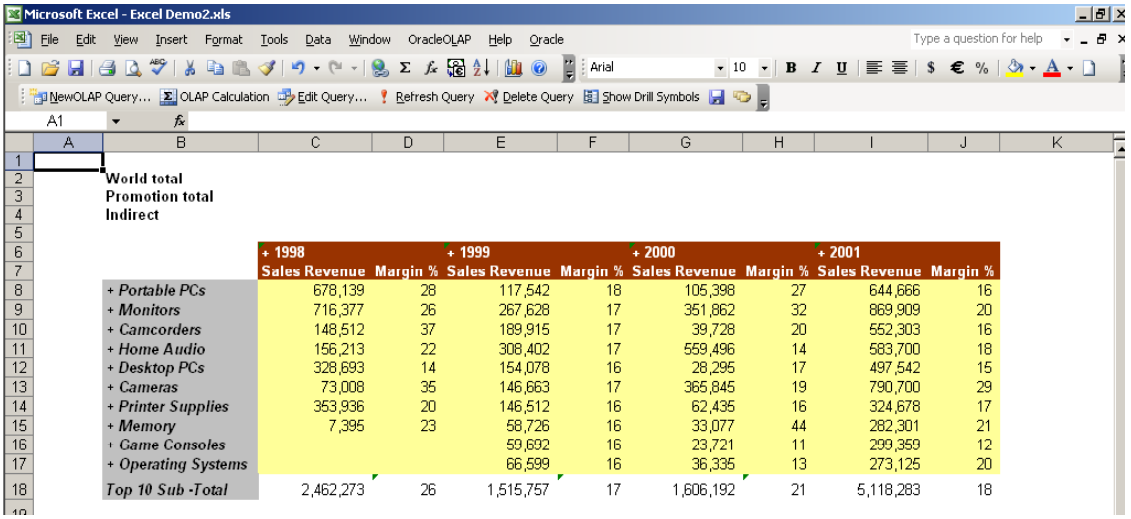
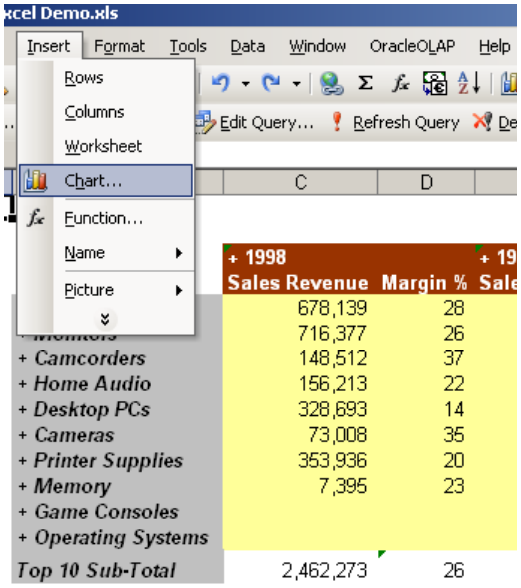
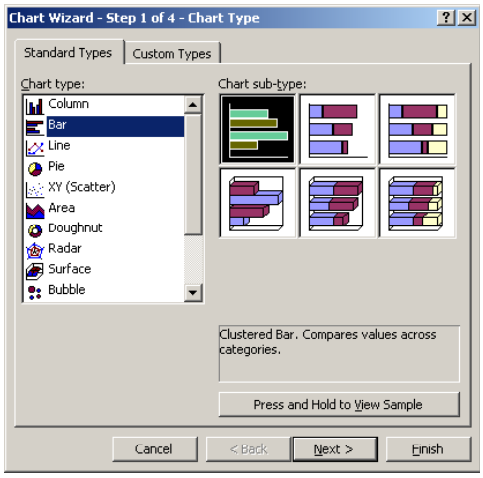


Figure 13:Updated BI Spreadsheet Add-In Worksheet

Charts are very visual and appealing medium that make it easy for users to see comparisons, patterns, and trends contained within a dataset. For example, a chart allows a user to very simply and quickly see if sales revenue is falling or rising over time.

Step 2 is to add an Excel Chart using the Chart Wizard.





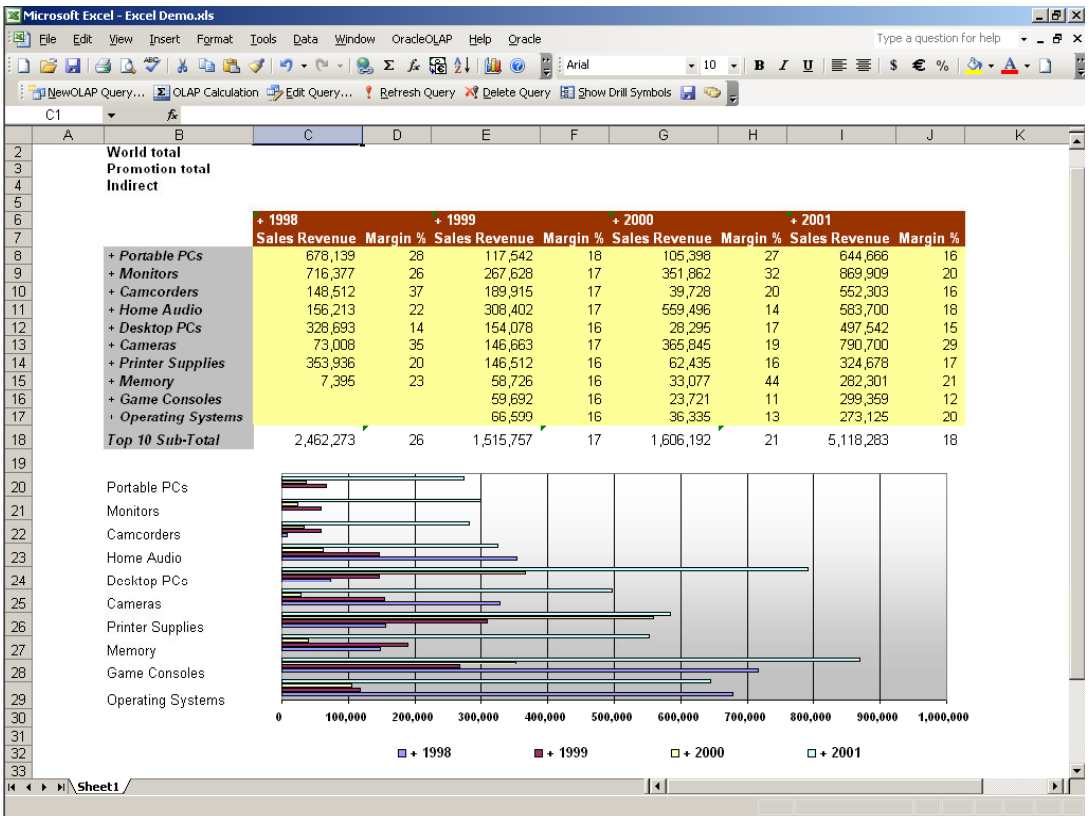


Figure 14: Adding an Excel Chart to an BI Spreadsheet Add-In Worksheet

Step 3 is to add another level of integration. We can extend the paging controls to use Excel toolbox radio buttons and link these to Excel macros. This would allow the user to quickly, easily and visually change the page dimension. This can be used to control the Channel and Geography selection within our report. The original paging controls can easily be hidden by either hiding the rows or by setting the row height to zero. The result of both operations is the same. To use the radio buttons it is necessary to use the Excel toolbox to place the radio buttons at the side of the report. Once the buttons have been correctly aligned a macro can be assigned to each button to change the page dimension value:

```
Private Sub SetChannelRange(SetText As
String)

    Range("Channel_Range").Select
    Range("Channel_Range") = SetText

End Sub

Private Sub OptAll_Click()

     SetChannelRange ("Channel total")

End Sub
```

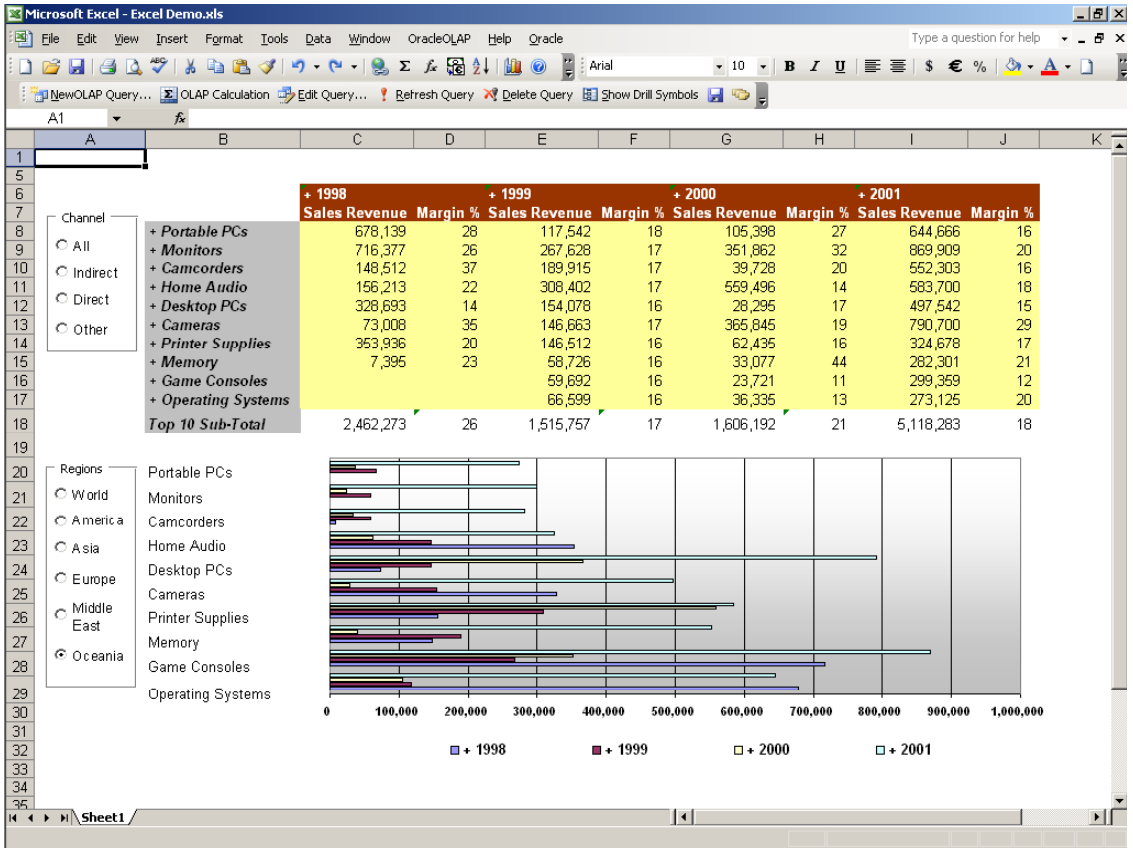


Figure 15: Adding Radio Buttons to the BI Spreadsheet Add-In Worksheet

CONCLUSION

It is generally accepted that many Excel users spend more time creating their data set rather than analysing their data and creating useful business information. The Oracle BI Spreadsheet Add-in provides Excel users with the direct access to the powerful Oracle OLAP environment with a consistent set of wizards also used in Oracle BI Discoverer – unified user interface, zero-learning curve. Excel users can now

- Directly access terabyte size data warehouses
- Create OLAP queries using business terms and definitions
- Create queries that automatically fetch the latest data from
- Create queries from multiple OLAP cubes in a single spreadsheet
- Extend their OLAP environment using the BI Spreadsheet Add-In Calculation Wizard
- Merge OLAP information with data from other data sources

So users get all the benefits of OLAP within the most popular desktop analysis interface without sacrificing basic Excel functionality.

Products Used in the Demonstration

This demonstration is designed to highlight Oracle's business intelligence OLAP solution, its calculation capabilities and data access methods.

It should be understood that other tools could play the same roles as the Oracle tools. The OLAP option's OLAP API is public and can be used by third parties to develop dimensionally aware applications on the Database. The SQL interface allows other SQL based tools to access Oracle10g's multidimensional data types. These SQL based applications can either unaware of the multidimensional data types or they can interact with the multidimensional data types and multidimensional engine..

Products used in the demonstration are listed in the following table.

Role	Product	Description
Database	Oracle Database Enterprise Edition with OLAP Option	The OLAP option allows the creation of a business language layer to translate the database storage objects Into business terms that make it easier for users to create complex business type queries,
Spreadsheet	Oracle Business Intelligence Add-in for Microsoft Excel	Allows Microsoft Excel to be used as an OLAP interface to the Oracle Database.

Database Implementation

Oracle Database

The Oracle10g Enterprise Edition Database is used for both the operational data source and for the data warehouse implementation. The OLAP option to the Oracle10g Database is used for advanced analytic capabilities.

Data Source

The source of the data is Oracle's 10g Common Schema which is based on shipments and general ledger systems that feed the following measures to the data warehouse: revenue and cost.

Logical Data Model

To support advanced analysis and to promote ease of use, a dimensional model was implemented. The model includes five dimensions, time, channel, product, promotion and geography. Each dimension has several levels of summarization within one or more hierarchies. A list of measures follows.

Sales Revenue	Costs	Margin
Sales Revenue	Costs	Margin
6 Month Moving Average	6 Month Moving Average	6 Month Moving Average
12 Month Moving Average	12 Month Moving Average	12 Month Moving Average
Prior Period % Growth	Prior Period % Growth	Prior Period % Growth
Prior Year % Growth	Prior Year % Growth	Prior Year % Growth
Forecast Seasonal	Forecast Best Fit	Forecast Best Fit
Forecast Exponential		
Forecast Linear		
Forecast Non-Linear		
Forecast Trend		
Forecast Best Fit		

Physical Model

Because we gave chosen to implement Common Schema chose using a dimensional model and there are significant analytic requirements, the physical model was implemented using multidimensional data types managed by the OLAP option to the Oracle10*g* database.

Since all of the measures were related to the same subject area (Common Schema sales data) and they shared logical dimensions, it was decided to include the entire data model in a single analytic workspace. An analytic workspace being a container in the database for a collection of related multidimensional data types.

The analytic workspace was constructed according to the *database standard form* specification. This is the physical design of the analytic workspace that contains data structures and metadata that is understood by Oracle administrative and end user tools. Oracle administrative tools build analytic workspaces according to the database standard form specification.

ETL Process

Oracle BI Warehouse Builder was used to build the analytic workspace. Oracle Warehouse Builder can be used to map to source operational systems, define the logical model and deploy the model to an analytic workspace (as well as to traditional relational data warehouses).

Oracle BI Warehouse Builder uses the API provided by the AW_CREATE package, which is a component of the OLAP option, to create and manage the

analytic workspace. Alternatively, either SQL scripts with AW_CREATE calls or the Analytic Workspace Manager GUI (which is also layered on the AW_CREATE package) could be used to build and manage the analytic workspace.

Because the SQL interface to multidimensional data types makes multidimensional data types accessible to SQL based applications, it was not necessary to replicate data in relational tables.

COMMON TERMINOLOGY

OLAP – On Line Analytical Processing (compare to OLTP)

ROLAP – Relational OLAP: imposing a dimensional view on relational data

MOLAP – Multidimensional OLAP: data stored in multidimensional format in an Analytic Workspace

Analytic Workspace (AW) – multidimensional schema, stored in Oracle LOB; contains dimensions, variables, hierarchies, programs, etc.

Dimension – a structure for categorizing data (e.g., Time, Geography, Product)

Hierarchy – defined levels of a dimension, for aggregation and drilling

Measure – stores (or calculates) multidimensional data; equivalent to a fact column.

Formula – a calculated measure in an AW

Cube – logical organization of measures with identical dimensions.

Metadata (CWM) – Data that describes data and other structures. The OLAP catalog is metadata that describes data in multidimensional terms, such as cubes, measures, dimensions, and attributes.

DML – Data Manipulation Language: dimensionally-aware programming language for Analytic Workspaces

Standard form – a specific structure of hierarchy dimensions, level dimensions, parent relations, etc., that identify the roles and relationships of objects in an AW. Required for an Aw to be accessible to OLAP tools.

Enablement – the process by which the objects in an AW are made known to the OLAP API

OLAPI – the OLAP API, through which BI Beans communicates with an AW. OLAPI generates SQL

AW View – A View that allows SQL to access multidimensional objects in an AW

Materialized view – precomputed relational table of aggregated fact data. Used for ROLAP performance.

White Paper Title
December 2004
Author: Keith Laker
Contributing Authors: Aneel Shenker

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com