Oracle® Business Intelligence

Concepts Guide

10*g* Release 2 (10.1.2.0.0)

Part No. B13970-01

November 2004

ORACLE®

Oracle Business Intelligence Concepts Guide, 10*g* Release 2 (10.1.2.0.0)

Part No. B13970-01

Contents

2 How to Build a BI Solution

Send Us Your Comments

Oracle Business Intelligence Concepts Guide, 10*g* Release 2 (10.1.2.0.0)

Part No. B13970-01

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: infodev_us@oracle.com
- FAX: 781-238-9850. Attn: Business Intelligence Documentation Manager
- Postal service:

  Oracle Corporation
  Oracle Business Intelligence Documentation Manager
  10 Van de Graaff Drive
  Burlington, MA 01803
  USA

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This document describes Oracle's business intelligence products and provides guidelines for selecting the ones that best meet your needs.

This preface contains the following topics:

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

`http://www.oracle.com/accessibility/`

Accessibility of Code Examples in Documentation

JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Audience

Oracle Business Intelligence Concepts Guide is intended for information consumers, report creators, analysts, DBAs, and application developers who perform the following tasks:

- Use data analysis to help make business decisions
- Develop custom reports for their own use and to share with other users
- Administer Oracle Database for a business intelligence system
- Develop custom solutions for analyzing data stored in an Oracle database

To use this document, you need no prior experience with business intelligence software.

Structure

This document contains the following chapters:

Chapter 1, "Getting Started With Oracle Business Intelligence"

This chapter provides a general introduction to business intelligence, contains descriptions of Oracle Business Intelligence tools, and offers recommendations for their use.

Chapter 2, "How to Build a BI Solution"

This chapter provides sample scenarios for two different Oracle Business Intelligence solutions.

Glossary

The glossary contains definitions of the special terms used in this guide.

Related Documents

For more information, see these Oracle resources:

- *Oracle Business Intelligence Installation Guide*
- *Oracle Business Intelligence Tools Installation Guide*
- *Oracle Business Intelligence Discoverer Plus User's Guide*
- *Oracle Business Intelligence Discoverer Publishing Workbooks in Oracle Application Server Portal*
- *Oracle Business Intelligence Discoverer Viewer User's Guide*
- *Oracle Business Intelligence Discoverer Administration Guide*
- *Oracle Warehouse Builder User's Guide*
- *Oracle Data Warehousing Guide*
- *Oracle Enterprise Manager Administrator's Guide*
- *Oracle OLAP Application Developer's Guide*
- Oracle Business Intelligence Beans Help (in Oracle JDeveloper)
- Oracle Business Intelligence Spreadsheet Add-In Help (in Microsoft Excel)

The Oracle Technology Network (OTN) provides services and resources that developers, DBAs, and architects need to build, test, and deploy applications using Oracle products and industry-standard technologies. It provides them with free access to documentation, release notes, white papers, product announcements, and other vital information. The OTN Web site is at

http://www.oracle.com/technology/index.html

The OTN Web site for Business Intelligence is at

http://www.oracle.com/technology/products/bi/index.html

Oracle By Example Series

The Oracle By Example (OBE) series provides hands-on, step-by-step instructions on how to implement various technology solutions to business problems. OBE contains numerous lessons on Business Intelligence from its Web site at

http://www.oracle.com/technology/obe/start/bi.html

Following is a list of some of these lessons.

OracleBI Discoverer

Discoverer lessons include:

- Creating Business Intelligence Reports Using OracleBI Discoverer Plus OLAP
- Analyzing Sales History Information using OracleBI Discoverer Plus Relational
- Viewing Sales History Information Using OracleBI Discoverer Viewer

OracleBI Spreadsheet Add-In

Spreadsheet Add-In lessons include:

- Using the OracleBI Spreadsheet Add-In

BI Beans

BI Beans lessons include:

- Develop Business Intelligence Applications Using OracleBI Beans
- Develop Business Intelligence Objects Using OracleBI Beans

Oracle Warehouse Builder

Warehouse Builder lessons include:

- Using Oracle Warehouse Builder

Conventions

The following conventions are used in this manual:

Convention	Meaning
. . .	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.
. . .	Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted
boldface text	Boldface type in text indicates a term defined in the text, the glossary, or in both locations.
italic text	Italic type in syntax indicates user-supplied names.
[ ]	Brackets enclose optional clauses from which you can choose one or none.

1

Getting Started With Oracle Business Intelligence

Oracle provides a complete and integrated set of tools to support business intelligence. This chapter describes these tools and how you can use them to create a business intelligence solution for your enterprise.

This chapter contains the following topics:

What Is Business Intelligence?

In the Information Age, corporations have at their disposal massive amounts of data collected in transactional systems. These systems are designed for the efficient selection, storage, and retrieval of data, and are essential for businesses to keep track of their affairs.

Having data is not the same as having information. The challenge is in deriving answers to business questions from the available data. This wealth of data can yield critical information about a business, so that decision makers at all levels can respond quickly to changes in the business climate.

Aggregating data into levels at which patterns can emerge, ordering levels into hierarchies to support drilling down and up through the levels, and using analytic functions such as lag, moving total, and year-to-date are among the techniques used to transform data into information. This information -- commonly called **business intelligence** -- can provide a significant edge in an increasingly competitive marketplace.

Business intelligence provides answers to basic questions such as:

- "What are my top five products?"
- "How do my sales this year compare to sales last year?"
- "What is the 3-month moving average of my sales?"

Business intelligence can also answer more probing analytical questions such as:

- Why are sales down in this region?
- What can we predict for sales next quarter?
- What factors can we alter to improve the sales forecast?
- How will our margins improve if we run this promotion?

Answering these questions requires an analysis of past performance, so that key decision makers can set a course for their businesses that will improve future performance, provide a more competitive edge, and thus enhance profitability. The Oracle Business Intelligence solution provides the information needed by decision makers whose ability to set goals today is dependent on how well they can predict the future.

Getting the answers to these questions involves:

- Access to summary historical and current data
- Calculations on the data
- Time-series analysis
- Forecasting
- Modeling
- What-if analysis

The technology to perform these computations and to present the results of analysis is contained in Oracle Business Intelligence.

The Oracle Business Intelligence Solution

Oracle Business Intelligence enables you to shape both long-term goals and day-to-day decisions, providing you with time-critical, relevant, and accurate information.

A Complete and Integrated Platform

Oracle provides a complete and integrated business intelligence platform that meets these basic business requirements:

- **Integrated query and analysis.** Regardless of technical expertise, users at all levels of an organization need to examine slices of data that are pertinent to their decisions, and to delve into that data as necessary in pursuit of business insights. Oracle provides query and analysis tools that enable users to customize their views of the data, to drill down to examine contributing factors, and to drill up to see how these factors contribute to the whole. Moreover, users can customize reports to suit their individual needs for visualizing the data.
- **Collaboration.** In a large organization, the ability to share reports has paramount importance. On a business level, it enables users who are working together to share their insights. On a technical level, it avoids duplication of effort and thus a waste of human resources.
- **Design and life-cycle management.** With graphical user interfaces, users can design a data store that supports the types of analysis required for making business decisions, load data from a wide variety of data sources, and manage that data store throughout its life span.

- **Security, manageability, and scalability.** All of the data for business intelligence is stored in Oracle Database as a single source of truth. It is not distributed in hundreds of spreadsheets, nor is it stored in a separate multidimensional database. All of the safety and security of Oracle Database applies to the business data, regardless of whether it is relational or multidimensional. Moreover, your technical staff uses the same tools to allocate resources, monitor performance, and so forth, so additional training is minimized.

Oracle offers tools for users involved at all stages in the life cycle of business intelligence:

- Information Consumers
- Report Developers and Analysts
- Database Administrators
- Applications Developers

This chapter describes the components that comprise Oracle Business Intelligence for each of these user groups.

> **Note:** If you do not have the components that you need to implement your Oracle business intelligence solution, then consult the Oracle Technology Network Web site at `http://www.oracle.com/technology`. You can view a wealth of information and download products, documentation, and more.

Analytical Processing Within Oracle Database

Oracle Business Intelligence provides **online analytical processing (OLAP)** in addition to its relational analytic and reporting capabilities. OLAP functionality is characterized by dynamic, multidimensional analysis of historical data, which supports activities such as the following:

- Calculating across dimensions and through hierarchies
- Analyzing trends
- Drilling up and down through hierarchies
- Rotating to change the dimensional orientation
- Forecasting
- What-if analysis

Oracle Database Enterprise Edition with the OLAP option provides multidimensional technology to support business intelligence. The OLAP option is fully integrated into Oracle Database. With this integration, Oracle provides the power of multidimensional analysis along with the manageability, scalability, and reliability of Oracle Database.

Tools for Information Consumers

Most users are interested in getting insight into their business data in order to make decisions. They gain these insights by consuming information that is presented in reports. They need to identify the trends and patterns in their data in order to plan their business strategies, but they want to spend as little time as possible on the technical aspects of designing and developing reports. Using Oracle's Web-based solution, they can use their browsers to access their data -- which means no software

installation, configuration, or maintenance for the majority of business intelligence users.

These integrated tools make this possible:

- OracleBI Discoverer Portlet Provider
- OracleBI Discoverer Viewer

Users can view predefined reports in a **dashboard** or drill down on a **portlet** into Discoverer Viewer, where they can modify these reports for their particular needs. The data storage format is transparent, so that multidimensional and relational data stores are indistinguishable.

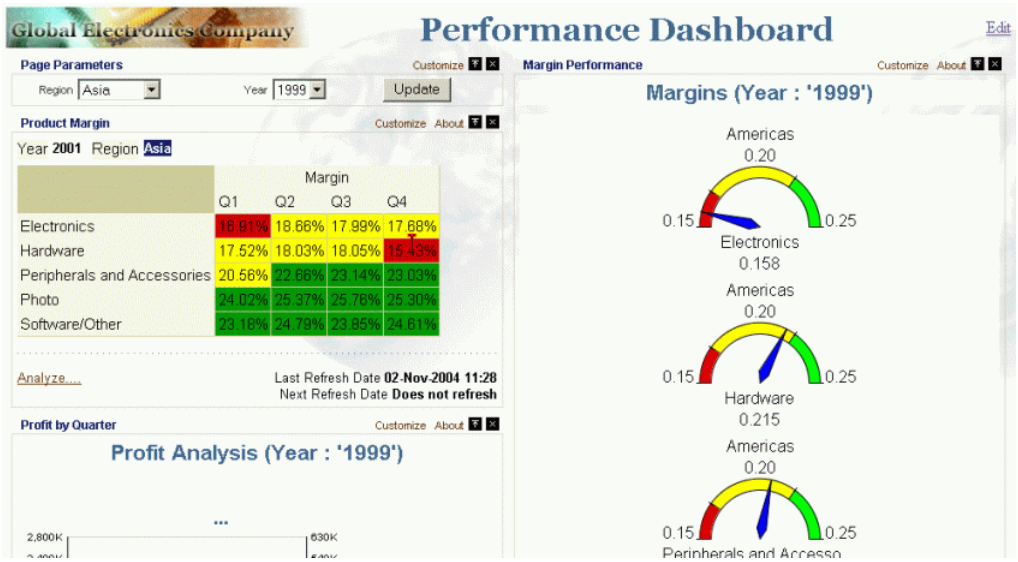
OracleBI Discoverer Portlet Provider

Discoverer Portlet Provider in conjunction with OracleAS Portal simplifies the creation of dashboards containing reports developed in Discoverer. Report builders can take advantage of OracleAS Portal's powerful distribution capabilities. A simple wizard lets them publish any existing Discoverer report, making it easy to create secure, personalized dashboards for tracking business performance measures.

These dashboards provide information to Discoverer Viewer. Clicking on the **Analyze** link in any Discoverer report automatically opens Discoverer Viewer, which is designed specifically with business users in mind. Its easy-to-use, pure-HTML user interface gives convenient access to reports from a standard Web browser, without requiring the installation of any additional desktop software.

Figure 1–1 shows a sample dashboard.

Figure 1–1 Dashboard of Discoverer Reports and Graphs



Data Requirements for Discoverer Portlet Provider

Discoverer Portlet Provider supports the following data models:

- Multidimensional (analytic workspaces)
- Relational (tables and views)

Distribution

Discoverer Portlet Provider is distributed on:

Oracle Application Server 10*g* CD-ROM Pack: Oracle Business Intelligence 10*g* Release 2 Version 10.1.2.0.0 CD-ROM

See Also

Oracle Business Intelligence Discoverer Publishing Workbooks in Oracle Application Server Portal

OracleBI Discoverer Viewer

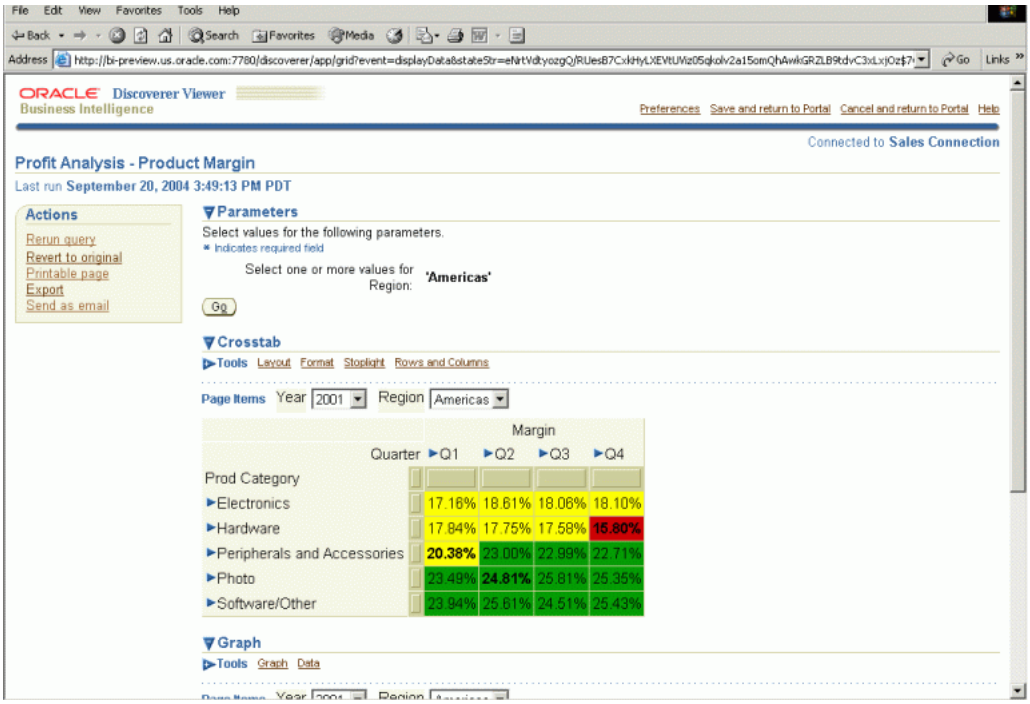
Discoverer Viewer gives business users immediate access to information published in a **dashboard** on the Internet or an intranet. For most business users, this is the easiest way to start using Discoverer. From the dashboard, they can drill down to Discoverer Viewer to perform further analysis.

Discoverer Viewer's interactive reporting environment lends itself to self-service business intelligence. Users can drill on crosstabs and graphs to view and analyze the underlying data, thereby identifying trends and anomalies in their business. They can easily tailor the information to their needs, such as changing the graph type used in a report, drilling down into the data, viewing a different selection of data, and adding **stoplight formatting** to spot trends. They can also customize reports that contain parameters, such as changing the selection of City in a report that shows the top 5 products for a particular city (such as New York). Discoverer enables them to focus on solving business problems and provides them with insight into their data.

Users can also share the results of their analysis by exporting their reports in Excel, HTML, PDF, or other popular file formats, and sending these files as e-mail attachments from within Discoverer Viewer.

Figure 1–2 shows a crosstab from the dashboard in Figure 1–1, which is now displayed in Discoverer Viewer.

Figure 1–2 Discoverer Viewer Displays Reports With User-Settable Parameters



Data Requirements for Discoverer Viewer

Discoverer Viewer supports the following data models:

- Multidimensional (analytic workspaces)
- Relational (tables and views)

Distribution

Discoverer Viewer is distributed on:

Oracle Application Server 10*g* CD-ROM Pack: Oracle Business Intelligence 10*g* Release 2 Version 10.1.2.0.0 CD-ROM

See Also

Oracle Business Intelligence Discoverer Viewer User's Guide

Tools for Report Developers and Analysts

Report developers, analysts, and other technically advanced users want to design their own reports based on their own ad-hoc analysis of the data. They can then publish their reports for general use.

Oracle offers two query and analysis tools:

- OracleBI Discoverer Plus
- OracleBI Spreadsheet Add-In

Both tools work with multidimensional data stores. Discoverer Plus also works with relational data stores.

OracleBI Discoverer Plus

Discoverer Plus is an ad-hoc query, reporting, analysis, and Web-publishing tool. It enables more technically advanced users to create new reports for their own analytic pursuits, and then to publish those reports in a dashboard for less technical users to explore in Discoverer Viewer or in an OracleAS dashboard by using Discover Portlet Provider.

Discoverer Plus's intuitive user interface guides users through the entire process of building and publishing sophisticated reports containing crosstabs and graphs. These users can choose from multiple graph types and layout options to create a visual representation of their query results. Using Discoverer Plus, they can create queries, drill, pivot, slice and dice data, add analytic calculations, graph the data, share results with users, and export their Discoverer reports in various data formats.

Discoverer Plus operates against two different types of data store:

- **Multidimensional**. Discoverer uses OLAP **metadata** to access data in a multidimensional data store. When operating against a multidimensional data store, Discoverer Plus is called Discoverer Plus OLAP. Additional information is provided in "Using Discoverer with Multidimensional Data Stores" on page 1-7.
- **Relational**. Discoverer uses an End User Layer (EUL) to access relational data in any format. When operating against an EUL, Discoverer Plus is called Discoverer Plus Relational. Additional information is provided in "Using Discoverer with Relational Data Stores" on page 1-8.

Using Discoverer with Multidimensional Data Stores

Using Discoverer Plus OLAP, users can access and analyze multidimensional data from their company's database, without having to understand complex database concepts. With wizards and menus, Discoverer Plus OLAP guides them through the steps to retrieve and analyze data that supports their business decisions. The OLAP Query Wizard and the OLAP Calculation Wizard are integral components of Discoverer Plus OLAP, as described in this topic. All Oracle Business Intelligence tools that are enabled for OLAP use these wizards.

Using Discoverer Plus OLAP, analysts can formulate their queries in the language of business. Consider the following request for information:

For **fiscal years** 2003 and 2004, show the percent change in **sales** for the top 10 **products** for each of the top 10 **customers** based on sales.

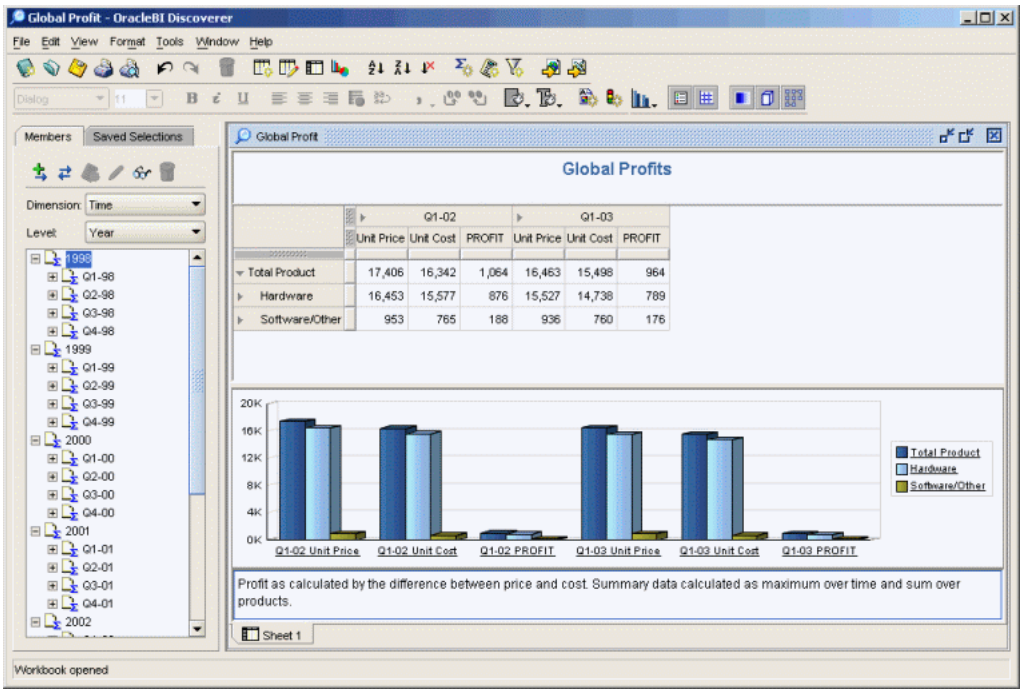
Sales is a business measure, calculated in US dollars or another currency. Fiscal years, products, and customers are the dimensions that provide a context for the data. This request is articulated in business terms, but easily translates into a query in the language of multidimensional analysis, such as **measure** and **dimension**.

The multidimensional data model also facilitates the creation of **custom measure**s. From stored **measure**s, users can select from numerous operators and functions to generate a wealth of information, such as year-to-date, percent change from a year ago, rank, share, and variance calculations.

Most standard queries involve simple data selection and retrieval. However, OLAP queries are more structured and involve calculations, time series analysis, and access to aggregated historical and current data. In OLAP queries, users know the dimensions and hierarchies that structure the measures. Data access is fast; all data operations are measured in seconds, including those with complex calculations.

Figure 1–3 shows a sample report generated by Discoverer Plus OLAP.

Figure 1–3 Sample Crosstab and Bar Graph in Discoverer Plus OLAP

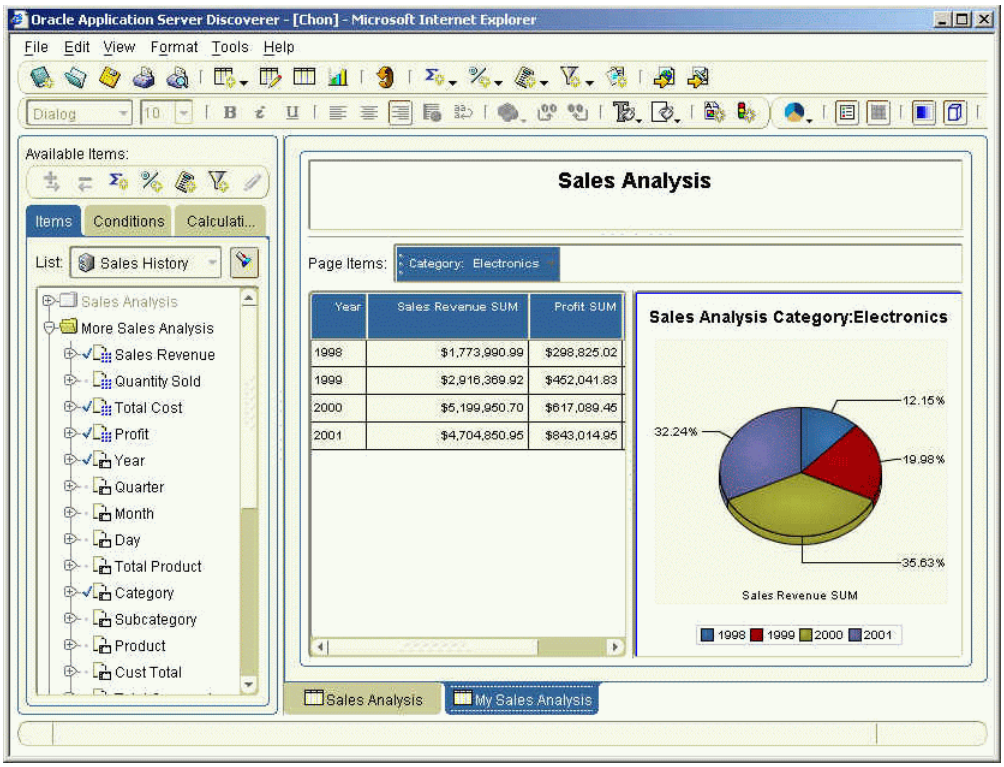


Using Discoverer with Relational Data Stores

Analysts that have Oracle Database Standard Edition, or Enterprise Edition without the OLAP option, can also use Oracle's business intelligence solution. Discoverer Plus Relational provides efficient interactive report layout and formatting capabilities, and it helps users achieve business insights through value-added analysis. They can add totals and percentages to their reports and add data-driven "stoplight" formatting to identify exceptions. They can also add sophisticated numerical and statistical analysis by leveraging the computational power of Oracle Database.

Figure 1–4 shows a report generated by Discoverer Plus Relational.

Figure 1–4 Sample Table and Pie Graph in Discoverer Plus Relational



Data Requirements for Discoverer Plus

Discoverer Plus supports the following data models:

- Multidimensional (analytic workspaces)
- Relational (tables and views)

Distribution

Discoverer Plus is distributed on:

Oracle Application Server 10*g* CD-ROM Pack: Oracle Business Intelligence 10*g* Release 2 Version 10.1.2.0.0 CD-ROM

See Also

Oracle Business Intelligence Discoverer Plus User's Guide
OracleBI Discoverer Plus Help

OracleBI Spreadsheet Add-In

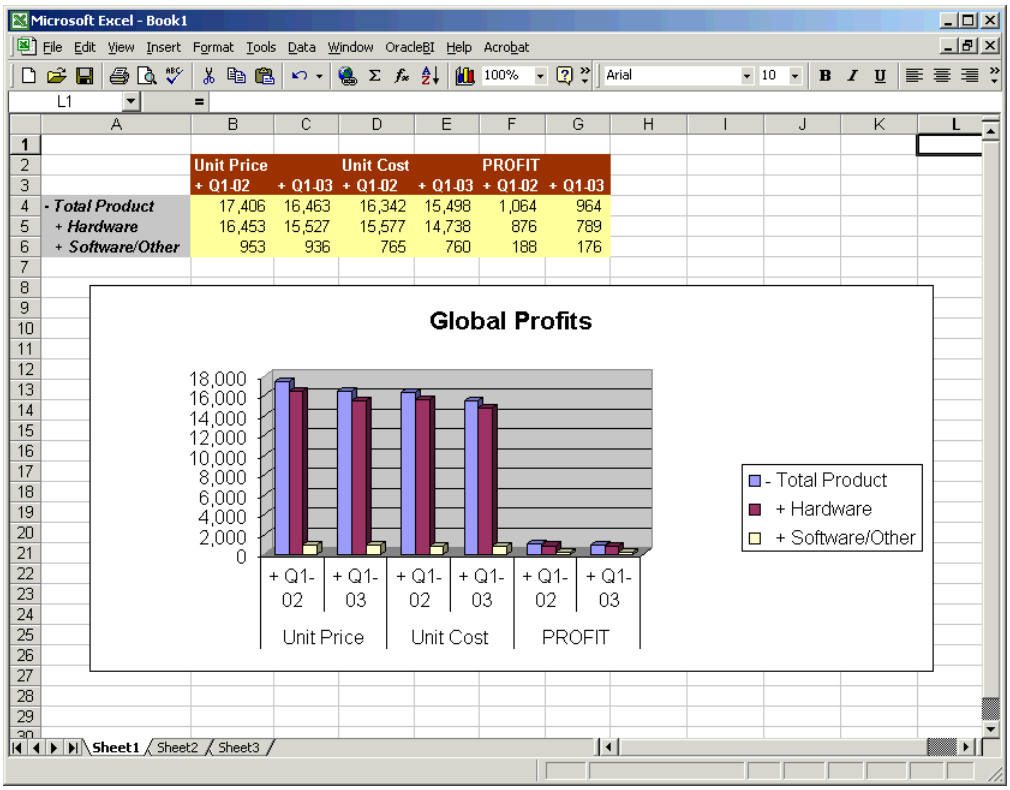
OracleBI Spreadsheet Add-In enables analysts to work with live multidimensional data in the familiar spreadsheet environment of Microsoft Excel. The add-in fetches data through an active connection to a multidimensional data store in Oracle Database and displays the data in a spreadsheet. Analysts can use the add-in to perform OLAP operations such as drilling, rotation, and data selection.

OracleBI Spreadsheet Add-In also solves the problem of each spreadsheet user's having personal calculations. Instead, calculations can be defined in the database by the DBA or system administrator and shared across user communities. OLAP

calculations are performed quickly and efficiently in the database, and they do not require massive downloads of data to Excel.

Figure 1–5 shows Oracle data in an Excel spreadsheet. Notice the addition of the OracleBI menu to the menu bar.

Figure 1–5 Oracle Data Displayed in an Excel Spreadsheet



Using a wizard-driven interface, users can select data simply by choosing from lists of dimension values. Or they can use various conditions for their selections. In addition, they can create custom measures by using a wizard. The Spreadsheet Add-In uses the same Query Wizard and Calculation Wizard as Discoverer Plus OLAP.

Analysts can also treat Oracle data like regular spreadsheet data. For example, they can create formulas and graphs in Excel, thereby combining the powerful analytic capabilities of Oracle OLAP with standard Excel calculations and formatting.

Data Requirements for OracleBI Spreadsheet Add-In

OracleBI Spreadsheet Add-In supports the following data model:

- Multidimensional (analytic workspaces)

Distribution

OracleBI Spreadsheet Add-In is available on:

Oracle Developer Suite 10*g* CD-ROM Pack: Oracle Business Intelligence Tools 10*g* Release 2 Version 10.1.2.0.0 CD-ROM

See Also

OracleBI Spreadsheet Add-In Help from within Excel

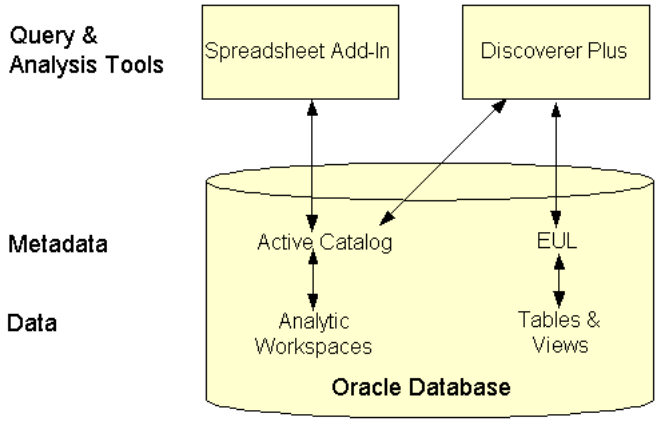
Key Factors in Choosing Tools for Report Developers and Analysts

Discoverer Plus and Spreadsheet Add-In use the same type of **metadata** to run against multidimensional data stores, so an instance of Oracle Database can support both tools without incurring additional administrative costs.

To access multidimensional data, which is stored in **analytic workspace**s, both tools use the **Active Catalog**. The Active Catalog is a set of relational views that contain OLAP metadata. Discoverer Plus has the additional capability of running against relational tables and views using an EUL.

Figure 1–6 provides an overview of the relationships among the tools, metadata, and data.

Figure 1–6 Access to Data From Query and Analysis Tools



Discoverer Plus and the Spreadsheet Add-In use the same Query Wizard and Calculation Wizard. This replication minimizes the learning curve for analysts using both tools.

The differences between the two tools are primarily in data presentation and report sharing:

- Discoverer Plus has been developed primarily for ad-hoc query and analysis. It provides a user interface and wizards that facilitate these activities, such as a navigator that allows in-place modifications to crosstabs, tables, and graphs, and its own presentation formats. It also provides a mechanism for users to manipulate the data display interactively for activities such as drilling up and down through the data.
- OracleBI Spreadsheet Add-In populates Excel workbooks, and users can display the data using Excel's graph and report formats. Users can share these spreadsheet files the same way they share other Excel spreadsheets.

Discoverer Plus provides analysts with all of the functionality needed to select, view, and analyze their data. Compared to Spreadsheet Add-In, Discoverer Plus offers more flexibility in swapping dimensions in the layout of reports, formatting, graphing, and defining custom calculations. Analysts can generate crosstabs, tables, and graphs in the same report. Report sharing is facilitated by Oracle Portal and Discoverer Viewer.

However, if analysts at your company are enthusiasts of Excel, then Spreadsheet Add-In provides them with additional analytical capability while building on their expertise in Excel.

Tools for Database Administrators

DBAs must provide data and metadata suitable for the query and analysis tools that will be used. Discoverer Plus OLAP and Spreadsheet Add-In require multidimensional data stores, as described in the next topic. The following tools are used to create and manage multidimensional data stores:

- Oracle Warehouse Builder
- Analytic Workspace Manager

Discoverer Plus Relational runs against any relational **schema**. The following tools are used to create and manage relational data stores for Discoverer:

- Oracle Warehouse Builder
- Discoverer Administrator

About Multidimensional Data Stores

Multidimensional data is stored in **analytic workspace**s where it can be manipulated by the OLAP engine in Oracle Database. Individual analytic workspaces are stored in tables in a relational schema. Like a relational table, an analytic workspace is owned by a particular user ID, and other users can be granted access to it. Within a single database, many analytic workspaces can be created and shared among users.

Analytic workspaces have been designed explicitly to handle multidimensionality in their physical data storage and manipulation of data. The multidimensional technology that underlies analytic workspaces is based on an indexed multidimensional array model, which provides direct **cell** access. This intrinsic multidimensionality affords analytic workspaces much of their speed and power in performing multidimensional analysis.

Creating Analytic Workspaces

Creating an analytic workspace involves a physical transformation of the data. The first step in that transformation is defining multidimensional objects such as measures, dimensions, levels, hierarchies, and attributes. The next step is mapping the multidimensional objects to the data sources. Last, the data loading process transforms the data from a table format into a multidimensional format.

Analytic workspaces have several different types of data containers, such as dimensions, variables, and relations. Each type of container can be used in a variety of ways to store different types of information, including business measures and metadata.

The analytic workspaces that are created by Oracle Enterprise Manager and Analytic Workspace Manager are in **database standard form** (typically called simply "standard form"). Standard form specifies the types of physical objects that are used to instantiate logical objects (such as dimensions and measures), and the type and form of the metadata that describes these logical objects. This metadata is exposed to SQL in the **Active Catalog**. These views are maintained automatically, so that a change to a standard form analytic workspace is reflected immediately by a change to the Active Catalog. Discoverer Plus OLAP and Spreadsheet Add-In use the Active Catalog to query data in analytic workspaces.

Summary Data

An analytic workspace initially contains only detail data loaded from its data sources. Summary data is calculated in the analytic workspace, and the aggregates are stored in

the same analytic workspace objects as the base data. The data is always presented to the application as fully solved; that is, both detail and summary values are provided, without requiring that calculations be specified in the query.

Analytic workspaces store aggregate data in an extremely compact form and provide a fast response time. Aggregates can be stored permanently in the analytic workspace, or only for the duration of an individual session, or only for a single query. Aggregation rules identify which aggregates are stored, and which aggregates are calculated **on the fly**. When an application queries the analytic workspace, either the aggregate values have already been calculated and can simply be retrieved, or they can be calculated on the fly from a small number of stored aggregates. Analytic workspaces are optimized for multidimensional calculations, making these run-time summarizations extremely fast.

Analytic workspaces provide an extensive list of **aggregation** methods, including weighted, hierarchical, and weighted hierarchical methods.

Distribution

Analytic workspaces are a feature of Oracle Database 10*g* Enterprise Edition with the OLAP option

See Also

Oracle OLAP Application Developer's Guide

Choosing a Data Storage Design

For most DBAs, relational tables and SQL provide a familiar environment. On the other hand, analytic workspaces require new tools and data transformations. So why use analytic workspaces? The answer is simply that analytic workspaces work better in some situations, and relational schemas work better in others.

This topic discusses the factors that determine which type of data store is the best choice for different kinds of data and analytical requirements. Your requirements may not perfectly match one type of data store, so you will need to weigh the relative importance of each requirement. You should also consider the direction your enterprise is heading in its business intelligence requirements. For example, you should consider whether ad-hoc reporting and advanced analytics are growing in popularity in your organization, or whether the analytic requirements are stable and predictable for the foreseeable future.

Basic Guidelines

Analytic workspaces are the best choice for meeting these user requirements:

- Advanced calculations, such as growth ratios and trends
- Unpredictable ad-hoc querying of all areas of the data as a routine use of the data store
- Custom measures as a routine method of data analysis
- What-if scenarios

Relational schemas provide the most satisfactory solution for these user requirements:

- Predefined reports, which access predictable areas of the data, as the primary use of the data store
- Flexible relationships

- Many-to-many parent-child relationships
- Attribute hierarchies

The following topics explore the technical differences between multidimensional and relational data stores that support these guidelines.

Structured and Unstructured Data Stores

The multidimensional data model is highly structured. Structure implies rules that govern the relationships among the data and control how the data can be queried. These rules simplify the construction of queries, and this simplicity supports ad-hoc querying of the data.

Analytic workspaces are the physical implementation of the multidimensional model, and thus are highly optimized for multidimensional queries. The OLAP engine leverages the model in performing highly efficient inter-row calculations, time series analysis, and indexing.

Relational schemas can have much less structure, and the relationships among tables and views can be established on a query-by-query basis. This flexibility can be very useful in selecting and reporting the data in existing schemas. Star schemas are frequently used for data warehouses, because they impose a structure that enables the relational engine to optimize analytical queries.

Processing Analytic Queries

The decision to use analytic workspaces or a relational schema determines how analytic queries are processed.

For data stored in analytic workspaces, the OLAP calculation engine performs analytic operations and supports sophisticated analysis, such as modeling and what-if analysis. If analysts require these types of analyses, then they need analytic workspaces. The OLAP engine also provides the fastest run-time response to analytic queries, which is important if you anticipate user sessions that are heavily analytical.

For data stored in a relational schema, analytical operations are performed by SQL. The `MODEL` clause and the analytical functions (such as `RANK`, `LEAD`, and `LAG`) enable analysts to include informative calculations in their reports such as year-to-date totals and moving averages.

Creating Summary Data

A basic characteristic of business analysis is hierarchically structured data; detail data is summarized at various levels, which allows trends and patterns to emerge. After an analyst has detected a pattern, he or she can drill down to lower levels to identify the factors that contributed to this pattern.

The creation and maintenance of summary data is a serious issue for DBAs. If no summary data is stored, then all summarizations must be performed in response to individual queries. This can easily result in unacceptably slow response time. At the other extreme, if all summary data is stored, then the database can quickly multiply in size.

The choice of a data store determines how summary data is generated.

- Analytic workspaces store aggregate data in the same objects as the base level data. The **aggregation** subsystem of the OLAP engine precalculates some areas of the cube and calculates other areas on the fly as described in "About Multidimensional Data Stores" on page 1-12. Analytic Workspace Manager provides **property** sheets for defining the rules for generating aggregate data.

- Relational schemas store aggregate data in **materialized view**s and summary tables, which are generated by Automated Summary Management (ASM) in Discoverer Administrator.

When predefined reports are run on a routine basis, the DBA knows which areas of the data are queried and what summary data is needed. This situation may be handled easily with a relational schema and a relatively small number of materialized views. However, extensive use of ad-hoc queries and user-defined custom measures create a random situation in which any part of the data store may be queried and summarized. Materialized views may not be available for these queries, so summary data must be generated at runtime.

Analytic workspaces store the summary data for all parts of a **cube**. Thus, analytic workspaces are a better choice for installations where users perform extensive ad-hoc querying and frequently define custom measures.

Oracle Warehouse Builder

Oracle Warehouse Builder enables the extraction, transformation, and loading of heterogeneous data to produce quality information. It is the only enterprise business intelligence design tool that manages the full life cycle of data and metadata integration for Oracle Database 10*g*.

Modeling a Business Intelligence System

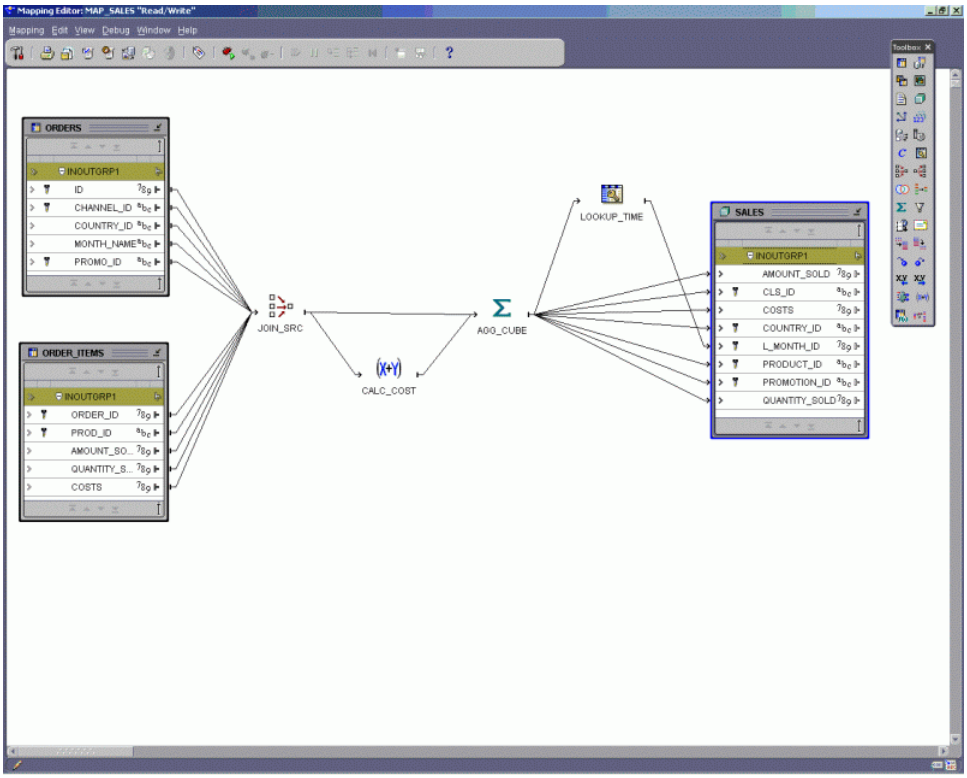
Among the main components in any business intelligence solution are the source systems that the business intelligence system will report upon. Warehouse Builder provides an easy-to-use, wizard-driven interface with which to capture the metadata for the source systems. DBAs can then use the metadata representations of the source to model the extraction processes.

Designing the ETL Processes

After capturing the metadata for the sources and designing the target schema, DBAs can create the data flows that define how data moves from sources to targets. They can map multiple sources into multiple targets, specify chained transformations, and apply complex PL/SQL transformations to the data. The mapping component also enables them to perform common operations such as joining, filtering, aggregating, and ordering data, as well as advanced and set-based operations. Warehouse Builder provides a graphical environment to model the ETL processes.

Figure 1–7 shows the Mapping Editor canvas, which defines how the target Sales table will be populated with the joined and transformed data from two source tables.

Figure 1–7 Mapping Editor in Warehouse Builder



Deploying the Data Store

Warehouse Builder can build a fully functional **data warehouse**, including analytic workspaces. Warehouse Builder 10.1.0.2.0 generates analytic workspaces in Oracle9*i* standard form. Analytic Workspace Manager must be used to upgrade analytic workspaces to Oracle10*g* standard form.

Warehouse Builder can also be used to build a relational schema and Discoverer End User Layer (EUL) for Discoverer Plus Relational.

Reports

Warehouse Builder generates lineage reports and impact analysis reports, which help manage the build process over the life cycle of the database.

- Lineage reports answer the question, Where did my data come from?
- Impact analysis reports answer the question, What is the impact of a particular change in the source tables or files?

These reports are produced in HTML.

Data Sources for Warehouse Builder

Oracle databases
Flat files
SAP R/3
Third-party databases such as DB2, SQL Server, Sybase, Informix
ODBC

Distribution

Oracle Warehouse Builder 10.1.0.2 is available on:

Oracle Business Intelligence Warehouse Builder CD-ROM

See Also

Oracle Warehouse Builder User's Guide

Analytic Workspace Manager

Analytic Workspace Manager is a primary tool for developing and managing analytic workspaces. DBAs and application developers can use it to define a logical multidimensional model, generate summary data, add derived measures, and refresh the base data from its relational sources in Oracle Database.

The main window in Analytic Workspace Manager provides access to two views:

- **Model View**. This view is used to define a logical multidimensional model of the data and to instantiate that model in a standard form analytic workspace. A drag-and-drop user interface facilitates mapping of the logical objects to columns in relational tables or views in Oracle Database. The source data can be in a star, snowflake, or any other type of schema that provides the **level** and **attribute** relationships required for OLAP analysis.

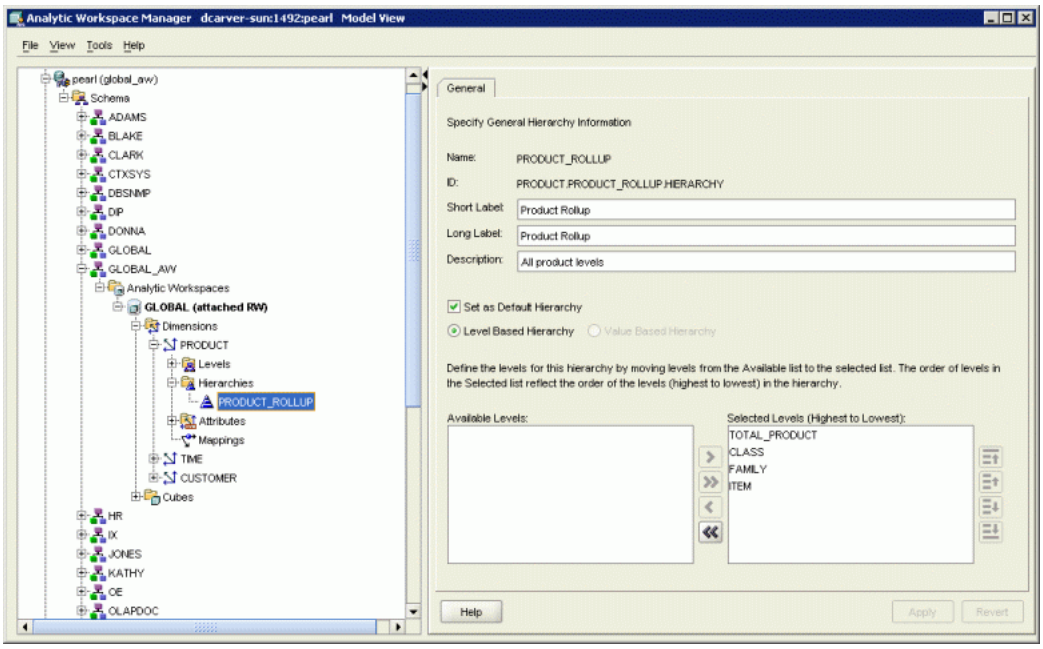
  All summarization and manipulation of the data is performed within the analytic workspace. The Model View is also used to generate summary data using a variety of operators. "Summary Data" on page 1-13 provides additional information about summarization.

  DBAs can also define calculated measures by using the same Calculation Wizard that appears in Discoverer Plus OLAP and Spreadsheet Add-In.

- **Object View**. This view is for advanced administrators who want to examine, modify, or create physical **object** in an analytic workspace.

Figure 1–8 shows the definition of a Product dimension in the Model View.

Figure 1–8 Model View in Analytic Workspace Manager



Data Requirements for Analytic Workspace Manager 10*g* Release 2

Star, snowflake, or other relational schemas that contain dimension level and **attribute** relationships

Distribution

Updates and new releases of Analytic Workspace Manager are distributed as they become available on
`http://www.oracle.com/technology/products/bi/olap/olap.html`

See Also

Oracle OLAP Application Developer's Guide

Discoverer Administrator

Discoverer Administrator is a powerful management tool for overseeing the database end of a relational Discoverer solution. With it, DBAs can secure the reporting environment through user privileges and access rights. Discoverer Administrator takes advantage of native database security or Oracle Applications security, so DBAs never have to define the users or their access privileges twice. All the existing database or Oracle Applications security policies are adhered to automatically.

Discoverer Administrator also manages the Discoverer reporting environment. Using this tool, DBAs can create Discoverer End User Layers (EULs) and Business Areas. By tuning the EUL (a semantic layer that conceals database names, joins, and other complexities), they can transform the database into an easy-to-understand report authoring environment for non-technical users, giving them access to the data they need without further DBA involvement.

DBAs can use Discoverer Administrator to make any modifications to an EUL that was created using Oracle Warehouse Builder.

Data Requirements for OracleBI Discoverer Administrator

Discoverer Administrator supports the following data model:

- Any relational schema

Distribution

Discoverer Administrator is available on:

Oracle Developer Suite 10*g* CD-ROM Pack: Oracle Business Intelligence Tools 10*g* Release 2 Version 10.1.2.0.0 CD-ROM

See Also

Oracle Business Intelligence Discoverer Administration Guide

Key Factors in Choosing Tools for Database Administrators

Warehouse Builder is a tool for Information Technology (IT) professionals who manage production systems. It is a powerful tool that can generate analytic workspaces as one element in a larger **ETL** process. It can also generate EULs along with relational data sources.

Analytic Workspace Manager is an easy-to-use tool designed for application developers, departmental DBAs, and other nonprofessional DBAs. It enables them to design and develop a data model quickly and interactively based on their reporting needs. After the data model has been developed and its design is stable, the IT department may assume responsibility for generating the analytic workspace using Warehouse Builder. Analytic Workspace Manager can be used to embellish the analytic workspaces created by the IT department, such as by adding custom measures.

Discoverer Administrator serves the same function for relational data stores that Analytic Workspace Manager serves for multidimensional data stores.

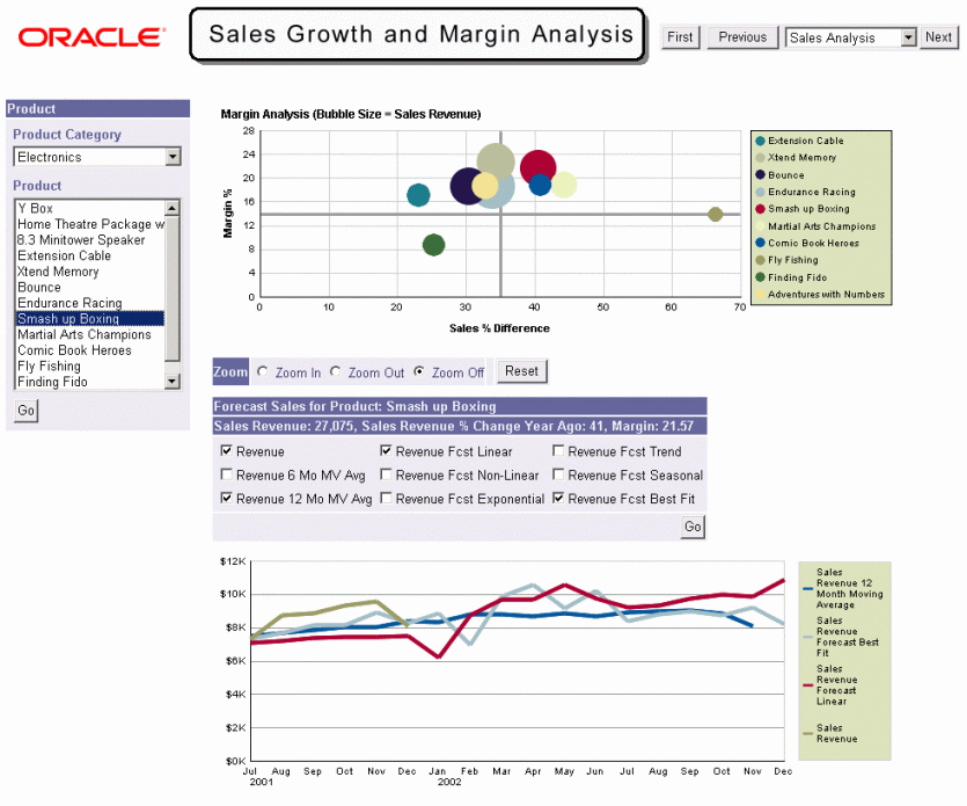
Tools for Application Developers

Application developers can create custom Java business intelligence applications and targeted solutions by using Oracle Business Intelligence Beans (BI Beans).

BI Beans is a set of standards-based Java beans. It provides analysis-aware application building blocks designed for Oracle Database. Using BI Beans, developers can build business intelligence applications that take advantage of the robust analytic capabilities available in Oracle Database, including Oracle OLAP. Applications can include advanced features such as interactive user interfaces, drill-to-detail reports, forecasting, and what-if analysis. Figure 1–9 shows a custom BI Beans application that presents two graphs:

- Bubble graph for margin analysis, in which the bubble size indicates sales revenue
- Line graph for forecasts, in which each line represents a different forecast method

Figure 1–9 BI Beans Custom Application



BI Beans includes Java beans for acquiring the data from Oracle Database (including the Query Wizard and Calculation Wizard described in "Tools for Report Developers and Analysts" on page 1-6), presenting the data in a wide variety of **crosstab** and graph formats, and saving report definitions, custom measures, data selections, and so forth.

BI Beans is seamlessly integrated into Oracle JDeveloper, providing a productive development environment for building custom business intelligence applications. Using JDeveloper and BI Beans, application developers can build Internet applications quickly and easily.

Data Requirements for BI Beans

BI Beans supports the following data model:

- Multidimensional (analytic workspaces)

Distribution

BI Beans is distributed on:

Oracle Developer Suite 10*g* CD-ROM Pack: Oracle Business Intelligence Tools 10*g* Release 2 Version 10.1.2.0.0 CD-ROM

First install Oracle JDeveloper 10*g* (9.0.5).

See Also

BI Beans Help from within JDeveloper

2

How to Build a BI Solution

This chapter provides two scenarios to show you how to craft a business intelligence system. There is one scenario for each query and analysis tool and for each type of data store, as described in Chapter 1.

This chapter contains the following topics:

- Implementing a Business Intelligence Solution
- Global Enterprises: Ad-Hoc Reporting and Advanced Analytics
- Acme Corporation: Web-Based Access to Relational Tables

Implementing a Business Intelligence Solution

This chapter describes two fictitious companies, with two different data sets and two different sets of requirements. It presents an Oracle business intelligence solution for each one, using a unique combination of the tools described in Chapter 1.

The Global example is based on the Global schema, which is available on the Oracle Business Intelligence Web site at

`http://www.oracle.com/technology/bi/olap/olap.html`

The Acme example is based on the sample schemas provided with Oracle Database. Both solutions follow the same basic steps, which are listed in the next topic.

Planning a Business Intelligence System

Implementing a business intelligence system requires careful planning to assure that it meets expectations. These are the basic steps:

1. Identify End-User Requirements
2. Identify the Data Sources
3. Design the Data Model
4. Create the Data Store
5. Generate the Summary Data
6. Prepare the Data for Client Access
7. Grant Access Rights
8. Distribute the Client Software and Documentation
9. Create and Distribute Reports

Identify End-User Requirements

It is important to anticipate how end users will analyze the data. By interviewing key users, you can identify the questions that the business intelligence system needs to answer.

You can ask questions such as:

- What information do you have now?
- What additional information do you need?
- How do you want the information presented?

Business requirements can be generated at all levels of your organization. The following are examples of the requirements you might need to address:

- Board of Directors
  - Competitive analysis
  - Key indicator tracking
  - Trend analysis
  - Exception reporting
- Administrative Analysis and Planning
  - Investment and acquisitions assessment
  - Reorganization analysis
  - Long-range planning
  - Resource allocation
  - Capacity planning
  - Human resource planning
- Finance Department
  - Budgeting
  - Consolidation
  - Variance analysis
  - Financial modeling
  - Cash management
  - Asset liability modeling
  - Activity-based management
- Sales and Marketing Department
  - Product profitability
  - Customer profiling
  - Distribution analysis
  - Sales performance and effectiveness

You can find out about the reports and data sources currently available, and what users like and dislike about their current information system. You may also discover what expectations they have about run-time performance.

Identify the Data Sources

From the types of questions that end users want answered, you can identify the sources of the data that can provide the answers. The data can be distributed among numerous locations, such as transactional databases and flat files. If the data is not available within your company, then you should discuss whether it is possible to acquire the data or whether end users must modify their expectations.

Design the Data Model

The logical data model must support the needs and expectations of your end users. This layer of metadata presents the data in business terms so that users can quickly identify the data they need to use.

For OLAP tools, you define dimensions, measures, and so forth. Then you can map the metadata objects to the physical data sources.

For relational tools, you define items, calculations, joins, and so forth using any existing relational **data source**.

Create the Data Store

You must deploy the data model as physical objects in the database and load the data from its sources.

For OLAP tools, the data store is an analytic workspace.

For relational tools, the data store may be the current OLTP system or a **star schema** in a **data warehouse**.

Generate the Summary Data

Business intelligence data is essentially hierarchical, so that data can be summarized at various levels. For performance, some of this data (ideally the data most frequently queried) is summarized and stored as a data maintenance procedure.

In analytic workspaces, summary data is stored in the same analytic workspace objects as the base-level data. In relational schemas, summary data is stored in materialized views.

Prepare the Data for Client Access

The client tools query the metadata to find out what data is available, where to get it, and how to present it.

Grant Access Rights

Users must have database access rights granted to them so that they can view and manipulate the data.

Distribute the Client Software and Documentation

After the data store is ready for client access, you can distribute the software and provide documentation to your end users.

Create and Distribute Reports

Report developers can develop reports and share them with the user community. For reports created in Discoverer Plus, you can create a dashboard where reports can be published.

Global Enterprises: Ad-Hoc Reporting and Advanced Analytics

Global Enterprises sells computer hardware and software in a variety of outlets. While they have been industry leaders for many years, the price of hardware has fallen dramatically in the last few years. They have little room for error if they are going to remain profitable.

To create a data warehouse, they need to consolidate information from disparate sources from around the world. This will provide them with the data to answer the following business analysis questions:

- What products are profitable?
- Who are our customers, and what and how are they buying?
- What accounts are most profitable?
- What is the performance of each distribution channel?
- Is there a seasonal variance to the business?

Identifying trends in the data that will answer these questions requires ad-hoc analysis and sophisticated analytic computations, such as:

Change in sales from prior period
Percent change in sales from prior period
Change in sales from prior year
Percent change in sales from prior year
Product share
Channel share
Market share
Extended cost
Extended margin
Extended margin change from prior period
Extended margin percent change from prior period

Global Enterprises uses Oracle Warehouse Builder to generate a star schema for their data warehouse. Their current problems arise not in the data itself, but in the tools to manipulate that data. They believe that a choice of Discoverer Plus OLAP and OracleBI Spreadsheet Add-In will satisfy their sales managers. However, they are also planning to have their IT department take a close look at BI Beans to explore custom solutions to their most aggressive requirements, such as forecasting and what-if analysis.

Software Requirements

The applications development team at Global Enterprises will use Analytic Workspace Manager for developing analytic workspaces. After they finish designing the logical model, they plan to turn over responsibility for managing the analytic workspaces to the IT department, who will use Warehouse Builder. Until then, the IT department will continue to deploy star schemas.

Tools for Information Consumers

Discoverer Viewer
OracleAS Portal with Discoverer Portlet Provider
Custom Java applications

Tools for Report Developers and Analysts

Discoverer Plus
OracleBI Spreadsheet Add-In

Tools for DBAs

Analytic Workspace Manager

Tools for Applications Developers

JDeveloper with BI Beans

Database

Oracle Database 10*g* Enterprise Edition with the OLAP option

Getting Started With Analytic Workspaces

Prerequisite: Install the software identified under "Software Requirements".

Global Enterprises has already identified its end-user requirements and the data sources, as described previously. To create an analytic workspace for Global Enterprises, take these remaining steps, which are described in more detail below:

1. Design the Data Model
2. Create the Data Store
3. Generate the Summary Data
4. Prepare the Data for Client Access
5. Grant Access Rights
6. Distribute the Client Software and Documentation
7. Define Advanced Analytics
8. Deploy Custom Applications

Design the Data Model

Open the Model View of Analytic Workspace Manager, and define these objects:

- Analytic workspace
- Dimensions
- Levels
- Attributes
- Hierarchies
- Cubes
- Measures

Analytic Workspace Manager stores the logical model as standard form metadata in the analytic workspace, and it creates all of the objects needed to instantiate the model at the same time.

When the source data is in a star or snowflake schema, you can quickly define a logical multidimensional model. The dimension tables contain columns for values at various levels, and their attributes. For example, a Time dimension table might have surrogate keys for weeks, quarters, and years; they are the levels of a hierarchy, which you might name the Calendar hierarchy. The display names, end date, and time span columns are attributes of the Time dimension. Each fact table is a **cube**, and the columns containing facts are the measures. Other types of schemas require individual analysis.

Create the Data Store

To map the logical objects to their data sources, choose the Mapping folder in the Model View Navigator of Analytic Workspace Manager. You can drag-and-drop the source tables onto the mapping canvas, then draw connectors between the logical objects and the appropriate columns. The Maintenance Wizard loads data from relational data sources into the analytic workspace.

Generate the Summary Data

You can define the aggregation rules for each cube. The rules identify the aggregation operator for each dimension and the portion of data that you want to presummarize and store. This plan provides the default summarization rules for all measures in the cube. The Maintenance Wizard executes the aggregation rules and generates the stored aggregates.

Prepare the Data for Client Access

Using the Calculation Wizard, you can easily define the derived measures for all users, thus adding a wealth of information to your analytic workspace.

Discoverer Plus OLAP and Spreadsheet Add-In enable users to define additional calculations (custom measures) and saved selections, so they do not have to be defined entirely in the analytic workspace. Discoverer Plus OLAP stores them in the Discoverer Catalog, where they can be accessed by other users with the appropriate permissions.

Grant Access Rights

For users to access the data, they require the following database privileges:

`CONNECT`
`QUERY REWRITE`
`SELECT` on the table in which the analytic workspace is stored

You can use Oracle Enterprise Manager or SQL to define users and groups, and to assign these privileges.

Distribute the Client Software and Documentation

After the analytic workspace is ready for use, you can make Spreadsheet Add-In and Discoverer Plus available for installation. You can also set up a dashboard with Discoverer portals so that power users can publish their reports.

Define Advanced Analytics

Advanced analytics take development time. You can define forecasts, models, and what-if scenarios in the analytic workspace by using BI Beans. Meanwhile, analysts can take advantage of the rich analytics already available through custom measures.

Check the Oracle Technology Network at http://www.oracle.com/technology for new versions of Analytic Workspace Manager Release 2, which will support these definitions in a graphical interface.

Deploy Custom Applications

The IT department plans to develop custom Java applications using BI Beans in JDeveloper, in order to tailor the user interface to the specific needs of their user community.

Acme Corporation: Web-Based Access to Relational Tables

Acme Corporation operates worldwide to fill orders for several different products. The company has several divisions:

- Human Resources tracks information on company employees and facilities.
- Order Entry tracks product inventories and sales of company products through various channels.
- Product Media maintains descriptions and detailed information on each product sold by the company.
- Information Exchange manages shipping through business-to-business (B2B) applications.

Acme wants to provide a flexible, out-of-the-box query and analysis tool to their executives and managers, who are distributed across the globe. The tool must enable them to query vast amounts of data that is stored in their Oracle databases, regardless of the product edition or the schema designs. They expect the queries to range from analysis of individual transactions on a daily basis to historical expenditure profiles for the whole organization. They do not have any requirements for extensive or ad-hoc analysis.

Acme chooses Discoverer for its query and analysis tool. With only a Web browser for the majority of users, they can quickly retrieve and review their data in numerous graphical formats, and can drill and pivot through the data.

Software Requirements

OracleBI Discoverer provides all of the components needed to satisfy Acme's business intelligence requirements. Since they already have their data stored in Oracle databases, and their immediate goal is analyzing "as is" data, they have no need for an ETL tool at this time. At a later date, they may want to use Oracle Warehouse Builder to create a true data warehouse containing cleaned, transformed, and aggregated data in a star schema.

Tools for Information Consumers

Discoverer Viewer
OracleAS Portal with Discoverer Portlet Provider

Tools for Report Developers and Analysts

Discoverer Plus

Tools for DBAs

Discoverer Administrator

Database

Oracle Database 10*g* Enterprise or Standard Edition

Getting Started With Discoverer

Acme Corporation has already identified its end-user requirements and the data store already exists, as described previously. To implement a Discoverer system for Acme Corporation, you must take these additional steps, which are described in more detail below.

Prerequisite: Install the software identified under "Software Requirements".

1. Design the Data Model
2. Generate the Summary Data
3. Prepare the Data for Client Access
4. Grant Access Rights
5. Distribute the Client Software and Documentation

> **See Also:** *Oracle Business Intelligence Discoverer Administration Guide* for detailed instructions for performing each of these steps.

Design the Data Model

The EUL contains the metadata that defines one or more business areas. A business area is a conceptual grouping of tables and views that apply to a user's specific data requirements. Business areas can be set up to reflect the needs of the user or group of users accessing the EUL.

After identifying the requirements of your users, you should have a good idea of the information different user groups need to access. For example, one group of users might want to access sales information, another group might want to access manufacturing information, and so on.

In Discoverer Administrator, you define a business area for classifying information with a common business purpose. Then you specify which database tables and views hold that class of information. You also load metadata and other information about the tables and views into the business area.

Generate the Summary Data

Use Discoverer Administrator to create summary tables, which will optimize query performance. You also have the option of using Discoverer automated summary management or registering existing summary tables.

Prepare the Data for Client Access

The default settings and contents of a business area are sufficient to enable users to access and analyze data. However, Discoverer Administrator provides you with a

number of features to enhance the default analysis capabilities so that users can view the data in the most flexible and understandable way.

These are some of the steps you can take to refine business areas:

- Create optional and mandatory conditions to restrict the number of rows returned in a folder.
- Create calculated items so that users do not need to formulate complex calculations themselves.
- Create joins to combine folders that were not joined when tables were loaded from the database.
- Combine folders into complex folders to completely hide joins and relational structures from users.
- Create custom folders to represent a result set returned by SQL
- Edit item names, descriptions, and other formatting information to make data easier to understand.
- Create item classes to support lists of values, alternative sorts, and drill to detail.
- Create hierarchies to simplify drill-down operations.

Grant Access Rights

You can grant access to business areas based on user requirements for accessing data.

Discoverer does not compromise database security. Users cannot see information in Discoverer that they do not have database privileges to access. All Discoverer security and privileges are imposed in addition to database security.

Distribute the Client Software and Documentation

After the database has been set up for Discoverer, you can make Discoverer Plus available to users. You can also set up a Portal dashboard where power users can publish their reports using Discoverer Portlet Provider.

Glossary

Active Catalog

A set of relational views that expose the standard form metadata stored in analytic workspaces, so that it can be accessed by SQL.

Discoverer, Spreadsheet Add-In, and custom applications that use BI Beans query the Active Catalog.

See also **database standard form**.

aggregation

The process of consolidating multiple data values into a single value. For example, sales data can be collected on a daily basis and aggregated to the week level, the weekly data could be aggregated to the month level, and so on. The data can then be referred to as aggregate data. Aggregation is synonymous with summarization, and aggregate data is synonymous with summary data. However, aggregation methods are not restricted to summarization. Other methods include average, first, last, minimum, and maximum.

analytic workspace

A multidimensional schema that is stored in a relational table. An analytic workspace can contain a variety of objects. Some of these objects may be integrally connected to other objects, while others are totally independent. Some objects store data that is useful to applications, and other objects may only exist for the purposes of the DBA or developer. There are several basic types of objects which play a variety of roles in the multidimensional model. In these respects, an analytic workspace is very similar to a relational schema.

The OLAP DML is the basic, low-level language for working in analytic workspaces. Tools are available in PL/SQL and Java that provide an interface to the OLAP DML for users already familiar with those languages.

See also **OLAP DML**.

attribute

A descriptive characteristic of either a single dimension member or a group of dimension members. When applied to a single member, attributes provide supplementary information that can be used for display (such as a descriptive name) or in analysis (such as the number of days in a time period). When applied to a group, attributes represent logical groupings that enable users to select data based on like characteristics. For example, in a database for footwear, you can use a shoe color attribute to select all boots, sneakers, and slippers that share the same color.

business intelligence

The information derived from monitoring key performance indicators, used to detect trends, identify warning signals, and provide a factual basis for critical business decisions.

cell

A single data value of a measure. In a multidimensional measure, a cell is identified by one member from each of the dimensions of the measure. For example, if you have a measure with the dimensions Month and District, then each combination of a month and a district identifies a separate cell of that measure.

See also **dimension**, **measure**.

crosstab

A layout of data in rows, columns, and pages. Dimension members are listed across the first row and down the first column; the data for measures appears in the cells that form the body of the crosstab. A crosstab can be used to display summary information and show how data varies across dimensions, such as sales by region by month. A crosstab is sometimes called a matrix.

Contrast with **table**. See also **cell**, **dimension member**, **measure**.

cube

A logical organization of measures with identical dimensions. The edges of a cube contain dimension members and the body of a cube contains data values. For example, a Sales cube might have edges containing members from the time, product, and customer dimensions. Volume Sales and Unit Sales might be two measures in the Sales cube.

See also **measure**.

custom measure

A derived measure that is calculated at run time and presented as one or more additional columns of data added to a result set. The result set includes a value for each dimension member currently selected. A custom measure typically employs a single-row function to perform a calculation on one or more stored measures. For example, an analyst might create a custom measure for the difference in costs from the prior period. Another analyst might create a custom measure that calculates profits by subtracting the Costs measure from the Sales measure.

See also **dimension member**, **OLAP DML**, **measure**.

custom member

A dimension member that is created at run time and defined as the parent of one or more existing dimension members. The values of a measure for a custom member are calculated using the aggregation rules for that dimension.

See also **aggregation**, **dimension member**, **measure**.

dashboard

A dynamic, graphical Web page containing all of a company's critical business information so that decision makers can view the status of all the key indicators in a single location.

data source

A database, application, repository, or file that provides data.

data warehouse

A relational database that is designed for query and analysis rather than transaction processing. A data warehouse usually contains historical data that is derived from transaction data, but it can include data from other sources. It separates analysis workload from transaction workload and enables a business to consolidate data from several sources.

database standard form

An analytic workspace that has been constructed with a specific set of objects. Each object must be defined with a set of properties that identify its role and its relationships with other objects in the analytic workspace. A set of catalogs within the analytic workspace organizes these properties, roles, and relationships as metadata and exposes it to SQL through the Active Catalog.

See also **Active Catalog**.

DBA

Database administrator. The person responsible for creating, installing, configuring, and maintaining Oracle Databases.

denormalize

Permit redundancy in a schema, typically to improve query performance. A star schema is almost always a denormalized relationship.

See also **star schema**. Contrast with **normalize**.

derived fact (or measure)

A fact (or measure) that is generated from existing data through a mathematical operation or a data transformation. Examples include averages, totals, percentages, and differences.

dimension

A structure that categorizes data. Among the most common dimensions for sales-oriented data are time, geography, and product. Most dimensions have hierarchies.

In an analytic workspace, a dimension is a container for a list of values. A dimension acts as an index for identifying the values of a measure. For example, if sales data has a separate sales figure for each month, then the data has a month dimension; that is, the data is organized by month.

In SQL, a dimension is a type of object that defines hierarchical (parent/child) relationships between pairs of column sets.

See also **hierarchy**, **measure**.

dimension member

One element in the list that makes up an analytic workspace dimension. Also called a dimension value. A computer company might have dimension members in the product dimension called `LAPPC` and `DESKPC`. Members in the geography dimension might include Boston and Paris. Members in the time dimension might include `NOV02`, `DEC02`, `JAN03`, `FEB03`, `MAR03`, and so forth.

dimension table

A relational table that stores all or part of the values for a logical dimension in a star or snowflake schema. Dimension tables describe the business entities of an enterprise,

represented as hierarchical, categorical information such as time, departments, locations, and products. They are sometimes called lookup or reference tables.

dimension value

See **dimension member**.

drill

To navigate from one item to a set of related items. Drilling typically involves navigating up and down through the levels in a hierarchy. When selecting data, you can expand or collapse a hierarchy by drilling down or up in it, respectively.

In relational data stores, drilling may navigate to related dimensions.

drill down

To expand the view to include child values that are associated with parent values in the hierarchy.

drill up

To collapse the list of descendant values that are associated with a parent value in the hierarchy.

ETL

Extract, Transform, and Load. An ETL tool such as Warehouse Builder can extract data from its source, transform it using a variety of techniques (such as joins, aggregations, and data conversion functions), and load it into new data containers such as relational tables.

fact

See **measure**. See also **derived fact (or measure)**.

fact table

A table in a star schema that contains facts. A fact table typically has two types of columns: those that contain facts and those that are foreign keys to dimension tables. The primary key of a fact table is usually a composite key that is made up of all of its foreign keys.

A fact table might contain either detail-level facts or facts that have been aggregated. Fact tables that contain aggregated facts are typically called summary tables or materialized views. A fact table usually contains facts with the same level of aggregation.

hierarchy

A logical structure that uses ordered levels as a means of organizing data. A hierarchy can be used to define data aggregation; for example, in a time dimension, a hierarchy might be used to aggregate data from the month level to the quarter level to the year level. A hierarchy can define a navigational drill path, regardless of whether the levels in the hierarchy represent aggregated totals.

In PL/SQL, hierarchies can be defined as part of a dimension object.

level

A position in a hierarchy. For example, a time dimension might have a hierarchy that represents data at the month, quarter, and year levels.

mapping

The definition of the relationship and data flow between source and target objects.

materialized view

A precomputed relational table comprising aggregated or joined data from fact and possibly dimension tables. Also known as a summary or aggregate table.

measure

Data that can be examined and analyzed, such as sales or cost data. You can select and display the data in a measure. Measures can be stored, or they can be calculated by means of formulas. The terms **measure** and **fact** are synonymous; measure is more commonly used in a multidimensional environment and fact is more commonly used in a relational environment.

There are both base measures and custom measures. Base measures, such as Volume Sales and Dollar Sales, are stored. Custom measures, such as Volume Share Year Ago, are calculated from base measures.

member

See **dimension member**.

metadata

Data that describes data and other structures, such as objects, business rules, and processes.

See also **database standard form**.

NA value

A special data value that indicates that data is "not available" (NA). It is the value of any cell to which a specific data value has not been assigned or for which data cannot be calculated.

See also **cell**, **sparsity**.

normalize

In a relational database, the process of removing redundancy in data by separating the data into multiple tables. Normalization typically is used to reduce data storage requirements.

object

In an analytic workspace, a distinct item in the workspace dictionary. Analytic workspaces consist of one or more objects, such as variables, formulas, dimensions, relations, and programs, which are used to organize, store, and retrieve data. Each object is created with a particular object type and stores a particular type of information. Objects that are the same type (for example, three variables) can have different roles within the analytic workspace.

OLAP DML

The low-level data definition and manipulation language for analytic workspaces.

on the fly

Calculated at run time in response to a specific query. In an analytic workspace, custom measures and custom members are typically calculated on the fly. Aggregate data can be precalculated, calculated on the fly, or a combination of the two methods.

Contrast with **precalculate**.

online analytical processing (OLAP)

Functionality characterized by dynamic, multidimensional analysis of historical data, which supports activities such as the following:

- Calculating across dimensions and through hierarchies
- Analyzing trends
- Drilling up and down through hierarchies
- Rotating to change the dimensional orientation

online transaction processing (OLTP)

Systems optimized for fast and reliable transaction handling. Compared to data analysis systems, most OLTP interactions involve a relatively small number of rows, but a larger group of tables.

portal

Groups of Web pages that host many different types of content that come from many different sources.

portlet

Reusable information components that summarize or provide access to different types of information sources. Examples of portlets include a dynamically updated report of quarterly earnings, a search field and button, and a simple user poll.

precalculate

Calculated and stored as a data maintenance procedure. In an analytic workspace, aggregate data can be precalculated, calculated on the fly, or a combination of the two methods.

Contrast with **on the fly**.

property

A characteristic of an object or component. Properties provide identifiers and descriptions, define object features (such as the number of decimal places or the color), or define object behaviors (such as when and how summary data is calculated). Properties are used extensively in standard form analytic workspaces.

See also **object**.

schema

A collection of related database objects. Relational schemas are grouped by database user ID and include tables, views, and other objects. Multidimensional schemas are called analytic workspaces and include dimensions, relations, variables, and other objects.

See also **analytic workspace**, **snowflake schema**, **star schema**.

snowflake schema

A type of star schema in which the dimension tables are partly or fully normalized.

See also **normalize**, **schema**, **star schema**.

solved data

A result set in which all derived data has been calculated. Data fetched from an analytic workspace is always fully solved, because all of the data in the result set is calculated before it is returned to the SQL-based application. The result set from the analytic workspace is the same whether the data was precalculated or calculated on the fly.

See also **on the fly**, **precalculate**.

source

See **data source**.

sparsity

A concept that refers to multidimensional data in which a relatively high percentage of the combinations of dimension values do not contain actual data. Such "empty," or NA, values consume storage space in an analytic workspace. To handle sparse data efficiently, you can create a composite.

There are two types of sparsity:

- Controlled sparsity occurs when a range of values of one or more dimensions has no data; for example, a new variable dimensioned by month for which you do not have data for past months. The cells exist because you have past months in the month dimension, but the cells contain NA values.
- Random sparsity occurs when NA values are scattered throughout the variable, usually because some combinations of dimension values never have any data. For example, a district might only sell certain products and never have data for other products. Other districts might sell some of those products and other ones, too.

See also **NA value**.

standard form

See **database standard form**.

star schema

A relational schema whose design represents a multidimensional data model. The star schema consists of one or more fact tables and one or more dimension tables that are related through foreign keys.

See also **schema**, **snowflake schema**

stoplight formatting

The use of background colors in a crosstab to distinguish desirable, acceptable, and unacceptable values:

- Red cells indicate unacceptable values, such as low Sales or high Costs.
- Green cells indicate desirable values, such as high Sales or low Costs.
- Yellow cells indicate acceptable values that do not need attention.

See also **crosstab**.

summary

See **aggregation**, **materialized view**.

table

A layout of data in rows. The first row identifies the dimensions, attributes, and measures whose data populates all other rows. Tables are available in Discoverer for relational data stores.

Index

W