

# DISCOVERER FOR SALES TRENDS

*Nicholas J. Donatone, MFG Systems, Inc.*

## Introduction

Oracle9iAS Discoverer™ is a component of Oracle9i™ Application Server (Oracle9iAS). This Business Intelligence tool is an intuitive ad hoc query, reporting, analysis, and web publishing tool. Discoverer empowers users of all levels to obtain information from data marts, data warehouses, and online transaction processing (OLTP) systems.

End users of this tool can rapidly create, modify, and execute ad hoc queries and reports. You can also setup users to only view pre-defined reports and graphs. Discoverer provides a view of the data that hides the complexity of the underlying data structure. This enables users to focus on solving business problems instead of trying to figure out how to write SQL to access their data.

Discoverer has many features that aid in Sales Analysis. Conveying information that drives your business is very important. Visualization of sales trends, tracking of performance goals and being able to visually identify exceptions are some of the strengths of Discoverer. Discoverer's analytical power and ease-of-use helps the end users take ownership of their own goals which can help improve overall Corporate results.

Even though Discoverer can retrieve information from online transaction processing (OLTP) systems, building a Data Mart for reporting purposes can have many benefits. Performance may be one of the biggest benefits. The raw data of the OLTP system can be moved into the Data Mart and summarized. Another side benefit of moving the data to a Data Mart is that the data in a Data Mart, unlike an OLTP database, only changes on pre-planned bases. Nothing can be more frustrating then running a report twice in the same time period and getting two results.

Finally Discoverer can be integrated in with existing Oracle9iAS Portal applications. Web enabling ad hoc BI tools allows the content to be available to all users.

## Data Mart for Sales Trends

We will be using the Tutorial that comes with Discoverer. The Tutorial is based on a Video Store Data Mart. By using this Data Mart, we will be able to demonstrate the various features of Discoverer.

First let's define the Data Mart. The Data Mart is a repository of data. The data can be gathered from operational data or other sources. The Data Mart is usually designed to serve a particular community of knowledge workers. This slice of data is a subset of the enterprise-wide database or a data warehouse. A Data Mart's focus is on meeting the specific needs of a group of knowledge users. The organization of data is targeted for analysis, content, presentation, and ease-of-use. Users of the Data Mart expect to have the data presented in terms that they are familiar with.

The terms *data mart* and *data warehouse* have a tendency to be used interchangeable. However, a data mart is usually geared to the user needs and a data warehouse builds on all the data that already exists. A data warehouse is a central aggregated repository of data (it may be distributed physically); a data mart may be derived from a data warehouse, but its' focus is on ease of access and usability.

Like a data warehouse, a data mart can be based on a Star Schema. A **Star Schema** is a means of aggregating data based on a set of known dimensions. It stores data multidimensionally in a two dimensional Relational Database Management System (RDBMS), such as Oracle. Another structure used is a Snowflake Schema. A **Snowflake Schema** is an extension of the star schema by means of applying additional dimensions to the dimensions of a star schema in a relational environment.

The Video Store Data Mart is a snowflake schema (see figure 1). It contains a fact table, SALES FACTS, which holds Sales, Unit Sales, Cost, Customer Counts and Profit. It also contains the dimensions Product, Store and Time. With one additional dimension on Time of Target Sales. To improve performance materialized views were also created (see figure 2).

We will be using this schema to build the Sales Trends Business Area and Reports with the Discoverer tool.

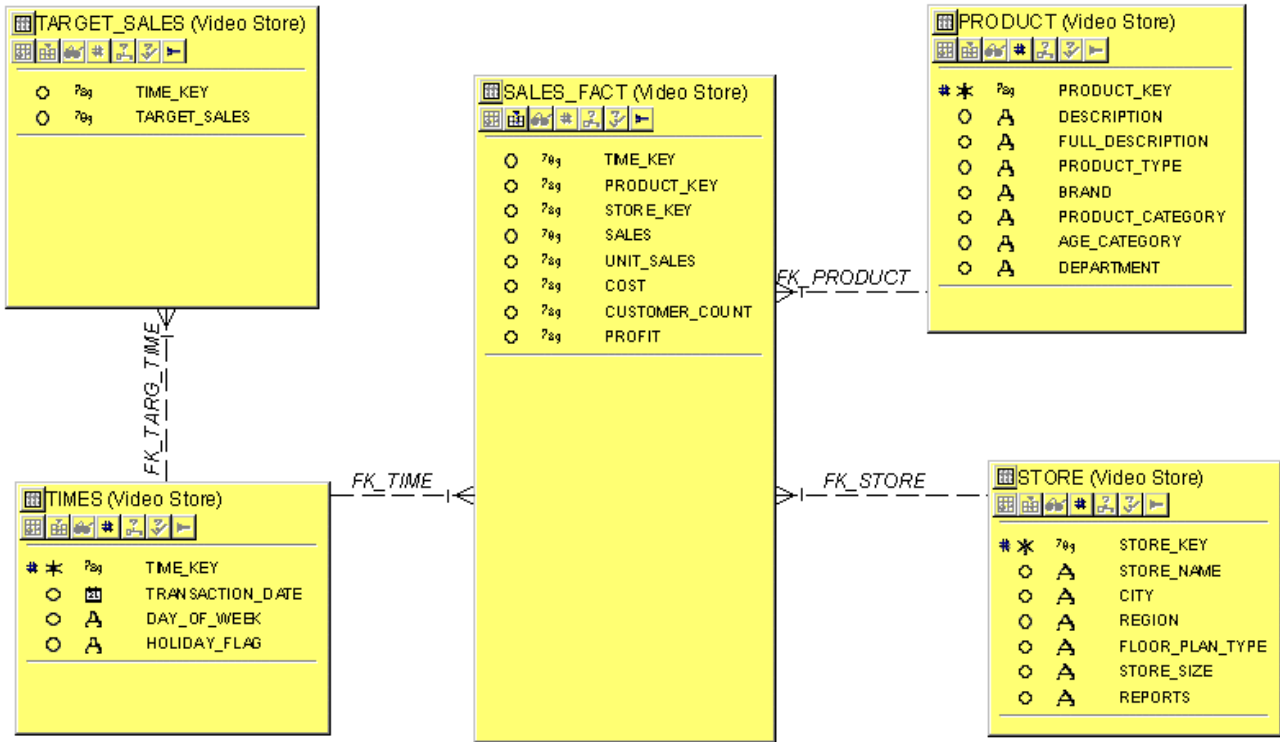


Figure 1. Discoverer Tutorial Video Store ERD

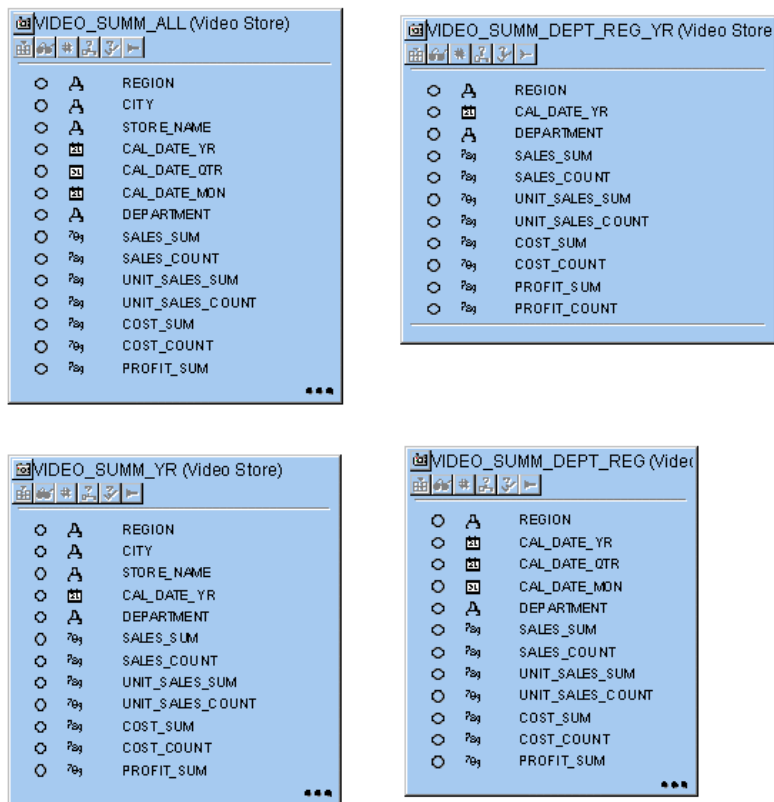


Figure 2. Discoverer Tutorial Video Store Materialized Views

The following is the code for generating the Materialized Views:

```
PROMPT Creating Materialized View 'VIDEO_SUMM_YR'
CREATE MATERIALIZED VIEW VIDEO_SUMM_YR
  REFRESH
  WITH ROWID
  AS
  SELECT
  DECODE(TIMES.TRANSACTION_DATE,NULL,TO_DATE(NULL,'MDDYYYY'),TO_DATE(TO_CHAR(TRUNC(TIMES.TRANSACTION_D
ATE,'YYYY'),'YYYY')||'01','YYYYMM')) AS CAL_DATE_YR, STORE.CITY AS CITY, PRODUCT.DEPARTMENT AS
DEPARTMENT, STORE.REGION AS REGION, STORE.STORE_NAME AS STORE_NAME, COUNT(SALES_FACT.COST) AS
COST_COUNT, COUNT(SALES_FACT.PROFIT) AS PROFIT_COUNT, COUNT(SALES_FACT.SALES) AS SALES_COUNT,
COUNT(SALES_FACT.UNIT_SALES) AS UNIT_SALES_COUNT, SUM(SALES_FACT.COST) AS COST_SUM,
SUM(SALES_FACT.PROFIT) AS PROFIT_SUM, SUM(SALES_FACT.SALES) AS SALES_SUM, SUM(SALES_FACT.UNIT_SALES)
AS UNIT_SALES_SUM
FROM VIDEO5.TIMES TIMES, VIDEO5.PRODUCT PRODUCT, VIDEO5.SALES_FACT SALES_FACT, VIDEO5.STORE STORE
  WHERE ( TIMES.TIME_KEY = SALES_FACT.TIME_KEY ) AND ( PRODUCT.PRODUCT_KEY = SALES_FACT.PRODUCT_KEY )
AND ( STORE.STORE_KEY = SALES_FACT.STORE_KEY )
GROUP BY
DECODE(TIMES.TRANSACTION_DATE,NULL,TO_DATE(NULL,'MDDYYYY'),TO_DATE(TO_CHAR(TRUNC(TIMES.TRANSACTION_D
ATE,'YYYY'),'YYYY')||'01','YYYYMM')), STORE.CITY, PRODUCT.DEPARTMENT, STORE.REGION, STORE.STORE_NAME
/

PROMPT Creating Materialized View 'VIDEO_SUMM_DEPT_REG_YR'
CREATE MATERIALIZED VIEW VIDEO_SUMM_DEPT_REG_YR
  REFRESH
  WITH ROWID
  AS
  SELECT
  DECODE(TIMES.TRANSACTION_DATE,NULL,TO_DATE(NULL,'MDDYYYY'),TO_DATE(TO_CHAR(TRUNC(TIMES.TRANSACTION_D
ATE,'YYYY'),'YYYY')||'01','YYYYMM')) AS CAL_DATE_YR, PRODUCT.DEPARTMENT AS DEPARTMENT, STORE.REGION
AS REGION, COUNT(SALES_FACT.COST) AS COST_COUNT, COUNT(SALES_FACT.PROFIT) AS PROFIT_COUNT,
COUNT(SALES_FACT.SALES) AS SALES_COUNT, COUNT(SALES_FACT.UNIT_SALES) AS UNIT_SALES_COUNT,
SUM(SALES_FACT.COST) AS COST_SUM, SUM(SALES_FACT.PROFIT) AS PROFIT_SUM, SUM(SALES_FACT.SALES) AS
SALES_SUM, SUM(SALES_FACT.UNIT_SALES) AS UNIT_SALES_SUM
FROM VIDEO5.TIMES TIMES, VIDEO5.PRODUCT PRODUCT, VIDEO5.SALES_FACT SALES_FACT, VIDEO5.STORE STORE
  WHERE ( TIMES.TIME_KEY = SALES_FACT.TIME_KEY ) AND ( PRODUCT.PRODUCT_KEY = SALES_FACT.PRODUCT_KEY )
AND ( STORE.STORE_KEY = SALES_FACT.STORE_KEY )
GROUP BY
DECODE(TIMES.TRANSACTION_DATE,NULL,TO_DATE(NULL,'MDDYYYY'),TO_DATE(TO_CHAR(TRUNC(TIMES.TRANSACTION_D
ATE,'YYYY'),'YYYY')||'01','YYYYMM')), PRODUCT.DEPARTMENT, STORE.REGION
/

PROMPT Creating Materialized View 'VIDEO_SUMM_DEPT_REG'
CREATE MATERIALIZED VIEW VIDEO_SUMM_DEPT_REG
  REFRESH
  WITH ROWID
  AS
  SELECT
  DECODE(TIMES.TRANSACTION_DATE,NULL,TO_DATE(NULL,'MDDYYYY'),TO_DATE(TO_CHAR(TRUNC(TIMES.TRANSACTION_D
ATE,'MM'),'MM')||'1900','MYYYY')) AS CAL_DATE_MON,
DECODE(TIMES.TRANSACTION_DATE,NULL,TO_DATE(NULL,'MDDYYYY'),TO_DATE(TO_CHAR(TRUNC(TIMES.TRANSACTION_D
ATE,'Q'),'MM')||'1900','MYYYY')) AS CAL_DATE_QTR,
DECODE(TIMES.TRANSACTION_DATE,NULL,TO_DATE(NULL,'MDDYYYY'),TO_DATE(TO_CHAR(TRUNC(TIMES.TRANSACTION_D
ATE,'YYYY'),'YYYY')||'01','YYYYMM')) AS CAL_DATE_YR, PRODUCT.DEPARTMENT AS DEPARTMENT, STORE.REGION
AS REGION, COUNT(SALES_FACT.COST) AS COST_COUNT, COUNT(SALES_FACT.PROFIT) AS PROFIT_COUNT,
COUNT(SALES_FACT.SALES) AS SALES_COUNT, COUNT(SALES_FACT.UNIT_SALES) AS UNIT_SALES_COUNT,
SUM(SALES_FACT.COST) AS COST_SUM, SUM(SALES_FACT.PROFIT) AS PROFIT_SUM, SUM(SALES_FACT.SALES) AS
SALES_SUM, SUM(SALES_FACT.UNIT_SALES) AS UNIT_SALES_SUM
FROM VIDEO5.TIMES TIMES, VIDEO5.PRODUCT PRODUCT, VIDEO5.SALES_FACT SALES_FACT, VIDEO5.STORE STORE
  WHERE ( TIMES.TIME_KEY = SALES_FACT.TIME_KEY ) AND ( PRODUCT.PRODUCT_KEY = SALES_FACT.PRODUCT_KEY )
AND ( STORE.STORE_KEY = SALES_FACT.STORE_KEY )
GROUP BY
DECODE(TIMES.TRANSACTION_DATE,NULL,TO_DATE(NULL,'MDDYYYY'),TO_DATE(TO_CHAR(TRUNC(TIMES.TRANSACTION_D
ATE,'MM'),'MM')||'1900','MYYYY')),
DECODE(TIMES.TRANSACTION_DATE,NULL,TO_DATE(NULL,'MDDYYYY'),TO_DATE(TO_CHAR(TRUNC(TIMES.TRANSACTION_D
ATE,'Q'),'MM')||'1900','MYYYY')),
DECODE(TIMES.TRANSACTION_DATE,NULL,TO_DATE(NULL,'MDDYYYY'),TO_DATE(TO_CHAR(TRUNC(TIMES.TRANSACTION_D
ATE,'YYYY'),'YYYY')||'01','YYYYMM')), PRODUCT.DEPARTMENT, STORE.REGION
/
```

```

PROMPT Creating Materialized View 'VIDEO_SUMM_ALL'
CREATE MATERIALIZED VIEW VIDEO_SUMM_ALL
  REFRESH
  WITH ROWID
  AS
  SELECT
  DECODE(TIMES.TRANSACTION_DATE,NULL,TO_DATE(NULL,'MMDDYYYY'),TO_DATE(TO_CHAR(TRUNC(TIMES.TRANSACTION_D
ATE,'MM'),'MM')||'1900','MMYYYY')) AS CAL_DATE_MON,
  DECODE(TIMES.TRANSACTION_DATE,NULL,TO_DATE(NULL,'MMDDYYYY'),TO_DATE(TO_CHAR(TRUNC(TIMES.TRANSACTION_D
ATE,'Q'),'MM')||'1900','MMYYYY')) AS CAL_DATE_QTR,
  DECODE(TIMES.TRANSACTION_DATE,NULL,TO_DATE(NULL,'MMDDYYYY'),TO_DATE(TO_CHAR(TRUNC(TIMES.TRANSACTION_D
ATE,'YYYY'),'YYYY')||'01','YYYYMM')) AS CAL_DATE_YR, STORE.CITY AS CITY, PRODUCT.DEPARTMENT AS
DEPARTMENT, STORE.REGION AS REGION, STORE.STORE_NAME AS STORE_NAME, COUNT(SALES_FACT.COST) AS
COST_COUNT, COUNT(SALES_FACT.PROFIT) AS PROFIT_COUNT, COUNT(SALES_FACT.SALES) AS SALES_COUNT,
COUNT(SALES_FACT.UNIT_SALES) AS UNIT_SALES_COUNT, SUM(SALES_FACT.COST) AS COST_SUM,
SUM(SALES_FACT.PROFIT) AS PROFIT_SUM, SUM(SALES_FACT.SALES) AS SALES_SUM, SUM(SALE_FACT.UNIT_SALES)
AS UNIT_SALES_SUM
FROM VIDEO5.TIMES TIMES, VIDEO5.PRODUCT PRODUCT, VIDEO5.SALES_FACT SALES_FACT, VIDEO5.STORE STORE
  WHERE ( TIMES.TIME_KEY = SALES_FACT.TIME_KEY ) AND ( PRODUCT.PRODUCT_KEY = SALES_FACT.PRODUCT_KEY )
AND ( STORE.STORE_KEY = SALES_FACT.STORE_KEY )
GROUP BY
  DECODE(TIMES.TRANSACTION_DATE,NULL,TO_DATE(NULL,'MMDDYYYY'),TO_DATE(TO_CHAR(TRUNC(TIMES.TRANSACTION_D
ATE,'MM'),'MM')||'1900','MMYYYY')),
  DECODE(TIMES.TRANSACTION_DATE,NULL,TO_DATE(NULL,'MMDDYYYY'),TO_DATE(TO_CHAR(TRUNC(TIMES.TRANSACTION_D
ATE,'Q'),'MM')||'1900','MMYYYY')),
  DECODE(TIMES.TRANSACTION_DATE,NULL,TO_DATE(NULL,'MMDDYYYY'),TO_DATE(TO_CHAR(TRUNC(TIMES.TRANSACTION_D
ATE,'YYYY'),'YYYY')||'01','YYYYMM')), STORE.CITY, PRODUCT.DEPARTMENT, STORE.REGION, STORE.STORE_NAME
/

```

## Discoverer Sales Trends Business Area

Before you can start writing a report in Discoverer you will need to build a Business Area for the Video Store Data Mart. The Business Area in Discoverer will present the business user with an easy to use view of the data. The Discoverer Business Intelligence system manager designs and presents a user oriented view of data. The Discoverer End User Layer (EUL) is a metadata repository and query management engine for data marts, data warehouses and on-line transaction processing (OLTP) systems.

The End User Layer (EUL) manages information describing relational data in business terms. Discoverer's End User Layer stores multiple 'Business Areas'. The business area simplifies tables and columns into easy to understand folders and items. Analytic support is provided through hierarchies, calculations, analytics, conditions, joins, and even complex SQL queries.

The Discoverer Administrator tool enables you to build and maintain the *End User Layer* (EUL). The EUL determines how a user accesses and views data. It is also where security is setup. Based on the specific needs of your users, you define business areas in the EUL by grouping related information from database tables, views or by using simple SQL. You may need to create more complex statements with set operators. A *business area* presents users with a set of folders and items that represents the data in familiar terminology. Folders are similar to tables and views. You can simplify query building for your users by linking the folders in the business area by defining a 'join'. Discoverer allows you to create predefined conditions and calculated items in the EUL that users are most likely to need. When your users create worksheets in Discoverer Plus, they can simply select from the predefined conditions and calculations.

As we mentioned before, you can improve query performance by creating summary folders of pre-aggregated data. The pre-aggregated data is stored in summary tables or materialized views. Discoverer will redirect queries to the summary data instead of using large volumes of detail data. Automatic refreshes of summary data can be scheduled to ensure it contain current, relevant data when users require it.

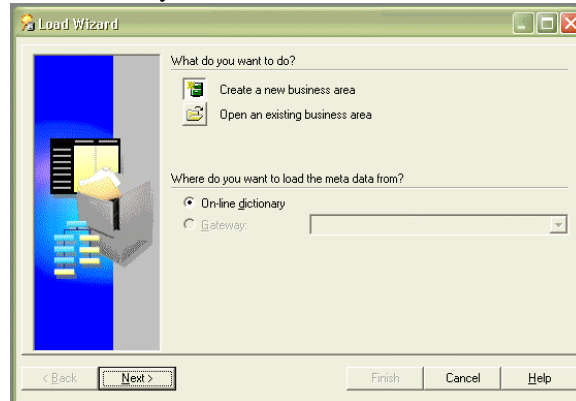
The Discoverer Administrator enables you to control access to business areas. You grant access permissions and task privileges to user IDs and/or roles for business areas. The database administrator continues to control access to database objects.

To summarize the Discoverer Administrator, you can:

- ?? design the Business Area (EUL) which allows you to define how business users access and view data
- ?? group related information for various business needs

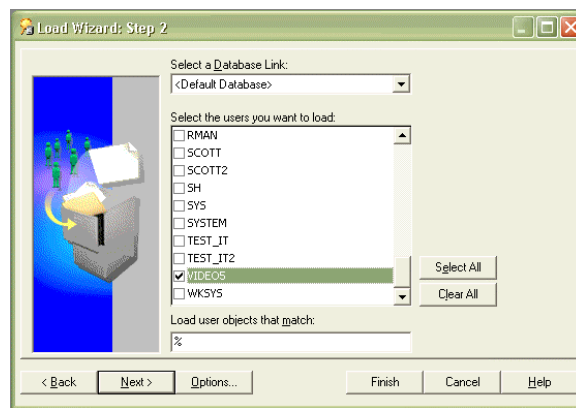
?? Optimize query performance by using summaries or materialized views

So, how do we build the Business Area (EUL)? First you will need to start Discoverer Administrator. After it starts you will be asked if you want to “Create a New Business Area” (see figure 3) or “Open an Existing Business Area”. Our Data Mart is already created, so we will use Discoverer’s ability to build a business area from a database schema.



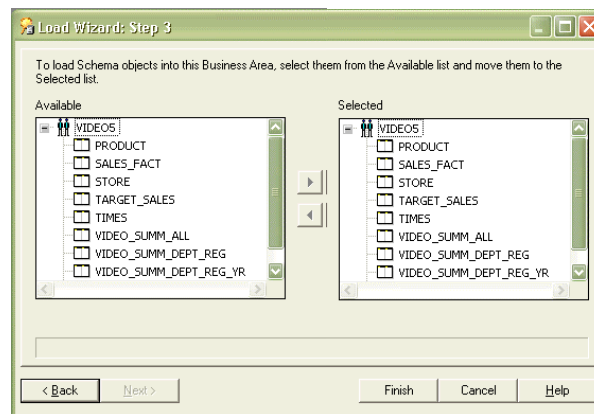
**Figure 3, Create a new business area**

Select “Create a New Business Area” and On-line dictionary, then press <NEXT> to select the Schema(s) you want to base the business area on (see figure 4).



**Figure 4, Select a Schema**

Press <NEXT> to select the tables, views, etc. to use in building the Business Area (see figure 5).

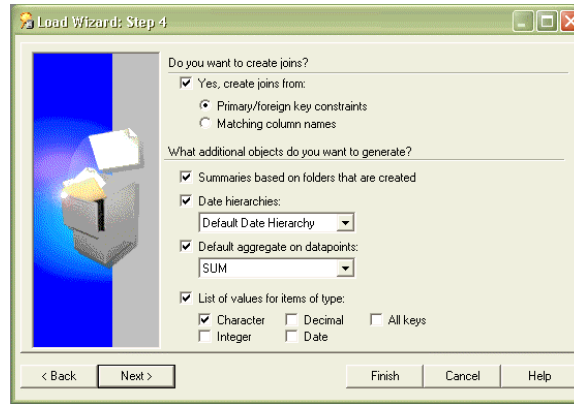


**Figure 5, Select table(s), view(s), etc.**

Press <NEXT> to specify how you want to generate joins, summaries, hierarchies, aggregates, data points and lists of values (see figure 6).

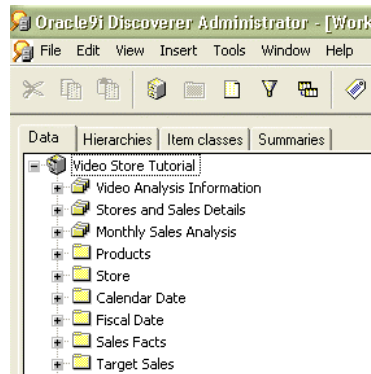


Tip: avoid using “Matching column names” when generating joins unless you are sure that columns like created\_date or modified\_by are not used. These are some of the common columns that applications use. If you did select “Matching column names” and these types of columns were present, Discoverer would build a great deal of unnecessary joins.



**Figure 6, Business Area Data View**

Press <FINISH> to start the automatic generation of the business area. The following screen will be displayed which is the Business Area Data View (see figure 7). The icons that look like multiple folders represent materialized views and the single folder is currently pointing to a table or view. Folders can point to multiple tables, SQL statements etc... Foreign Key constraints are used to create joins between folders. In just three simple steps we were able to build a business area that we can use to build reports.

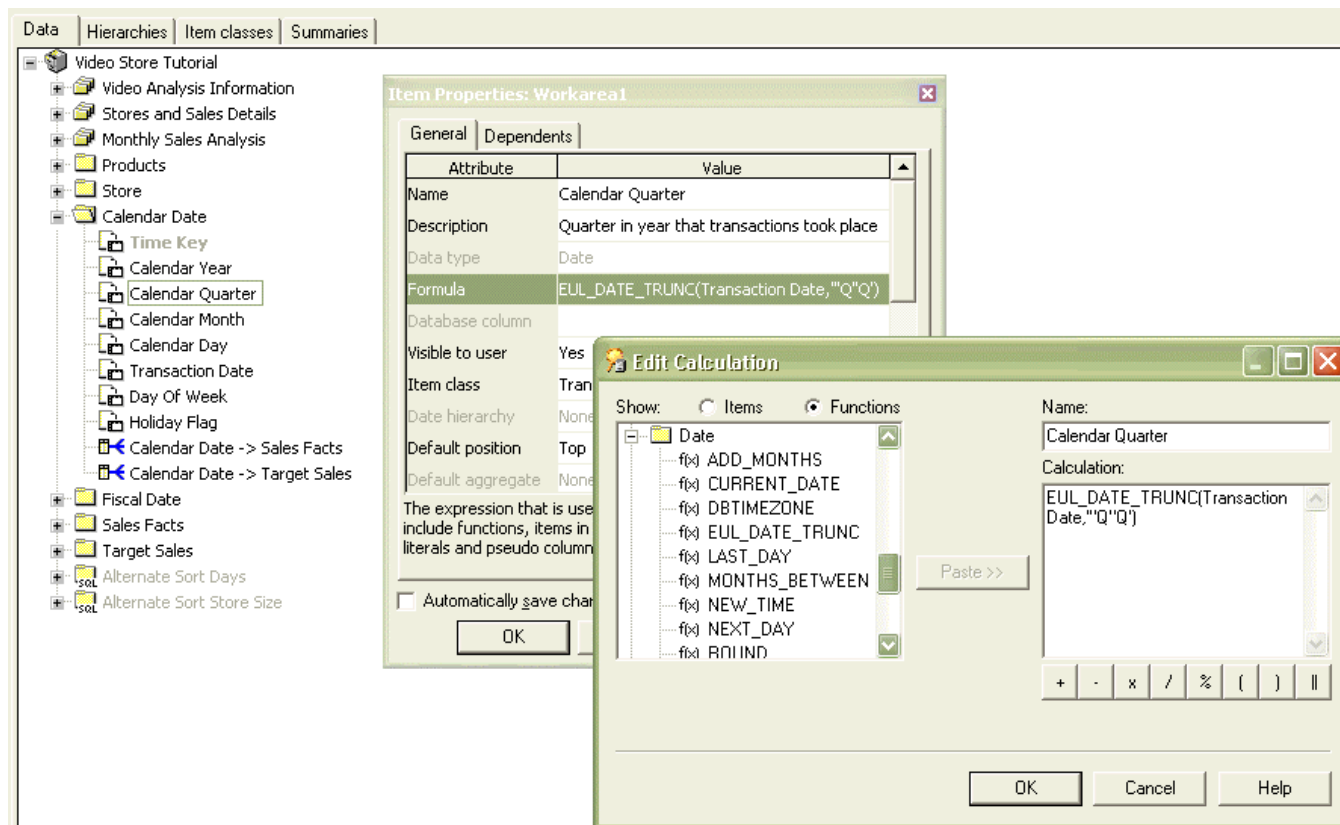


**Figure 7, Business Area Data View**

There are many more steps that you need to complete to make the business area complete. You may need to define additional items, data hierarchies, items classes (lists of values, alternate sorts and drill to detail) and/or summary folders. We won't be explaining all of these in this White Paper. You may want to review the Discoverer help files or review the documentation at Oracle's Tech Net site <http://otn.oracle.com/documentation/discoverer.html>.

Before we start building reports there is at least one more feature to mention that is useful in Sales Analysis. That feature is “Calculations”. In order to simplify creation of reports, creating predefined calculations, especially for Sales Reporting where the same type of calculation is used for many reports is quite common, is very important. There are many benefits to predefining calculations. One benefit is that the information reported based on these calculations will be consistent.

You can use the “create calculated item” wizard to define a new item under a folder (see figure 8). There is a comprehensive list of functions to choose from or you can build your own. In the example below (see figure 8) we are using the Date Function EUL\_DATE\_TRUNC to define calendar quarters.



**Figure 8, Calculations**

The calculations can be created in the Business Area or in Work Sheets. Discoverer has a very rich set of functions to help meet the needs of Sales Reporting. There are seven major groupings of these functions:

- ?? Analytic
- ?? Conversion
- ?? Date
- ?? Group
- ?? Numeric
- ?? Others
- ?? Strings

A couple of the calculations that we will need for our Sales Trends Business Area are:

Lag – an expression evaluated at an offset from the current position. To get last years sales in our example you can use the following calculation:

```
LAG(Sales SUM1,1) OVER(PARTITION BY Calendar Quarter,Region ORDER BY "Calendar Year" )
```

Rank – computes the rank of a row with respect to the other rows in the dataset. To rank the sales in our example you can use the following calculation:

```
RANK() OVER(PARTITION BY "Calendar Year",Region ORDER BY Sales SUM1 DESC )
```

To obtain the top N and bottom N products in our example you can use the Rank in a condition.

```
("Top N / Bottom N" <= :N OR "Top N / Bottom N Asc (used in condition only)" <= :N )
```

These are just a couple of the functions that can be used, but as you can see Discoverer exploits the analytical features of the Oracle9i Database.

### Sales Trends Reports

Using the Business Area created for the Video Store we can quickly build Sales Reports that will help our users determine Sales Trends. Some of the questions we might want to ask are:

"What are my top three and bottom three selling products?" (see figure 9)

Page Items: Year: 1999 Region: East

City	Product Description	Sales SUM	Top N / Bottom N
Atlanta	The Lion King	\$2,588.69	1
	The American President	\$2,496.09	2
	Apollo 13	\$2,468.49	3
	Licorice	\$11.25	111
	Potato Chips	\$10.55	112
	Alien3	\$2.35	113

Figure 9, Top 3 and Bottom 3 Products

"How do my sales this quarter compare to my sales last quarter?" (see figure 10)

Year	Quarter	Sales SUM	Sales Last Qtr	Quarterly Change
1998	Q1	\$185,914.27		
	Q2	\$190,208.82	\$185,914.27	\$4,294.55
	Q3	\$202,124.61	\$190,208.82	\$11,915.79
	Q4	\$203,777.51	\$202,124.61	\$1,652.90
1999	Q1	\$204,496.44	\$203,777.51	\$718.93
	Q2	\$216,631.97	\$204,496.44	\$12,135.53
	Q3	\$216,485.27	\$216,631.97	-\$146.70
	Q4	\$239,980.13	\$216,485.27	\$23,494.86
2000	Q1	\$282,388.78	\$239,980.13	\$42,408.65
	Q2	\$293,272.32	\$282,388.78	\$10,883.54

Figure 10, This Quarter vs. Last Quarter

The creation of the Top N / Bottom N Products report will be shown in detail. First we start Discoverer Plus; select create a new report and select the items you want on the report (see figure 11).

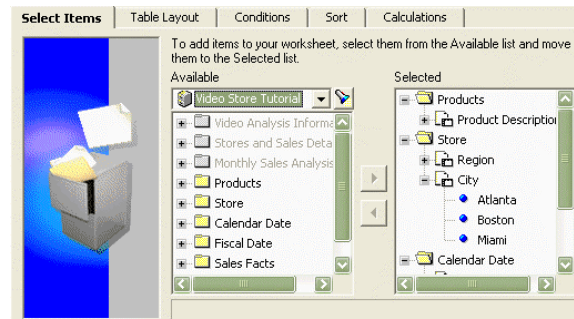
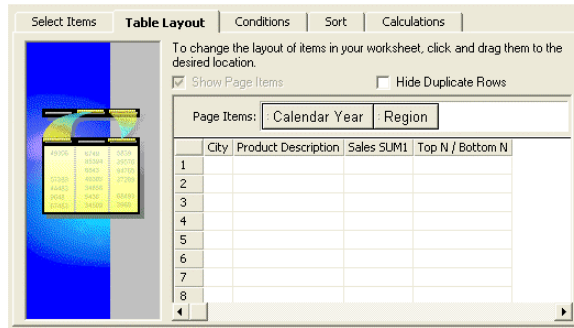


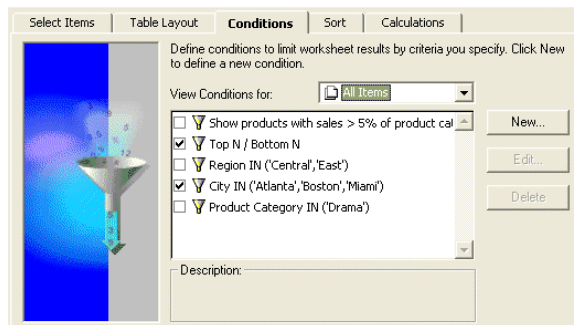
Figure 11, Select Items

Next arrange the items on the report (see figure 12) by dragging Calendar Year and Region to Page Items.



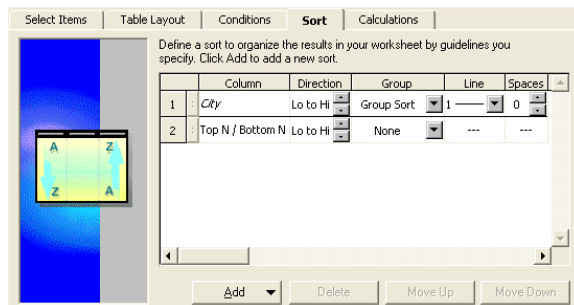
**Figure 12, Table Layout**

Next select from the list of predefined conditions Top N / Bottom N (see figure 13). We also only want to display Sales for Atlanta, Boston and Miami.



**Figure 13, Select Items**

Once you have selected the conditions you can specify the Sort order (see figure 14). Sort by City and then by Rank.



**Figure 14, Sort**

To finish our report we will select calculations (see figure 15).

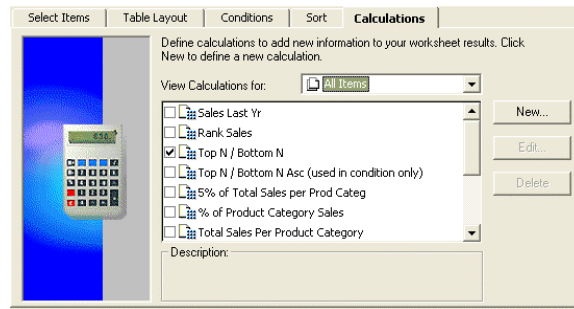


Figure 15, Calculations

Now run the report. When a query is run, Discoverer's redirection engine automatically detects the existence of summary tables or Oracle9i™ Materialized Views and redirects queries to take advantage of them. Discoverer's predictive query governor gives an estimate of the retrieval time *before* a query is run (see figure 16). If the predictive query governor determine that the query will take a long time, you can schedule it to run at a later time.

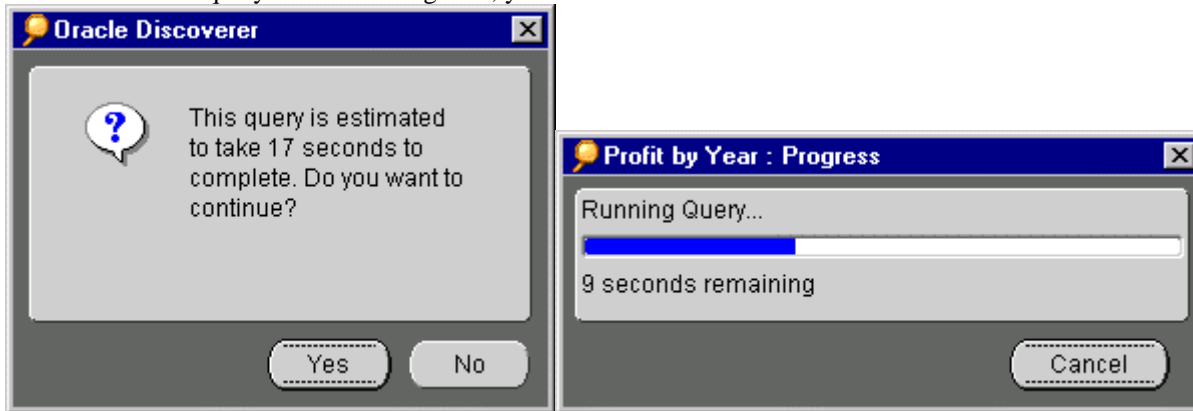


Figure 16, Query Governor/Predictive Query

The completed report will look like the one in figure 17.

Page Items: Year: 1999 Region: East			
City	Product Description	Sales SUM	Top N / Bottom N
Atlanta	The Lion King	\$2,588.69	1
	The American President	\$2,496.09	2
	Apollo 13	\$2,468.49	3
	Licorice	\$11.25	111
	Potato Chips	\$10.55	112
	Alien3	\$2.35	113

Figure 17, Top N / Bottom N Sales by City and Product

If you build your Business Area in Discoverer so that it includes most of the needs of your users, you will be able to empower them. As shown here, since all of the calculations and conditions were predefined, all we had to do to create this report was to point and click.

## Deploying Discoverer to a 9iAS Portal Application

You can deploy both Discoverer Plus and Discoverer Viewer to the web. Discoverer Plus allows you to build and customize reports (worksheets). Discoverer Viewer is used to display reports with minimal customization. You can deploy Discoverer Viewer via an Oracle9iAS Portal.

### What are Discoverer Portlets?

An Oracle9iAS Portal portlet is simply information placed within a region on a portal page. The Discover portlets are registered in Oracle9iAS. Discoverer is a portlet provider. There are two types of Discoverer portlets:

?? List of workbooks portlet

?? Worksheet portlet

The “List of workbooks portlet” includes the names of workbooks. A “Worksheet portlet” is an individual worksheet which contains only data in a tabular or cross-tabular form, only a graph, or both.

### Discoverer Portlets Customization

Personalization includes the default connection to the database, parameter values, if any, or portlet properties such as title, workbook filter, and workbook sorting column. “List of Workbooks portlet” can be customized as follows:

?? Discoverer connection

A Discoverer connection identifies the database connection (user and data source information) and the Discoverer End User Layer to be used.

?? Filter list of workbooks

Users can filter to be shown in the portlet. In our case we may only want to show all the workbooks containing the word “Sales”

?? Number of workbooks to show

?? Portlet title

?? Sort order for the workbooks displayed in the portlet

The “Worksheet portlet” can be customized as follows:

?? Discoverer connection

?? Show data and/or graph

?? Portlet title

### Publishing Content to Oracle9iAS Portal

It is very simple to publish content created by Discoverer with Oracle9iAS Portal. First you must login to Portal with an account that has the “*Create/Customize Page*” privilege. Next, edit the portal page in Oracle9iAS Portal and add either “List of Workbooks portlet” or “Worksheet List of Workbooks Portlet. You can then specify access to the portlet.

### Summary

Oracle9iAS Discoverer extends the business intelligence power of the Oracle9i database. It makes the company’s key performance indicators easy to access, allows you to personalize content according to your needs, and share information across the organization through the use of Oracle9iAS portlets.

### About the Author

Nicholas Donatone is Vice President of Outsourcing Services for MFG Systems Corporation. MFG Systems is a consulting firm that specializes in Oracle, located in Somerset, New Jersey. Nicholas is one of the founders of the New Jersey Oracle User Group and has been President/Co-President of the NJOUG for over 15 years. He has made presentations at the NJOUG, DEVOUG, ODTUG, AOTC, VOUG, NYOUG and at ECO.

Nicholas J. Donatone  
MFG Systems Corp  
100 Davidson Ave  
Somerset, NJ 08873

Phone: (732) 560-0010  
Fax: (732) 805-1989  
Email: [donatone@mfgsys.com](mailto:donatone@mfgsys.com)  
Company Website: [www.mfgsys.com](http://www.mfgsys.com)

NJOUG Website: [www.njoug.org](http://www.njoug.org)