Building OLAP Cubes

Purpose

In this tutorial, you use the Analytic Workspace Manager (AWM) tool to build an OLAP cube. You:

- Create a cube and its dependent components, including:
  - Dimensions
  - Measures (stored and calculated)
- Map the OLAP model to source data
- Enable MV rewrite to the cube
- Load data into the dimensions and measures
- View the OLAP data

NOTE: This tutorial requires either Oracle Database 11.2 or 12.1.

Time to Complete:

Approximately 60 minutes

Topics

This tutorial covers the following topics:

Viewing Screenshots

Place the cursor over this icon to load and view all the screenshots for this tutorial. (Caution: Because this action loads all screenshots simultaneously, response time may be slow depending on your Internet connection.)

Note: Alternatively, you can place the cursor over each individual icon in the following steps to load and view only the screenshot associated with that step.

Overview

AWM is a tool for creating, developing, and managing multidimensional data in an Oracle data warehouse. With this easy-to-use GUI tool, you create the container for OLAP data, an analytic workspace (AW), and then add OLAP dimensions and cubes.

In Oracle OLAP, a Cube provides a convenient way of collecting stored and calculated measures with similar characteristics, including dimensionality, aggregation rules, and so on. A particular AW may contain more than one cube, and each cube may describe a different dimensional shape. Multiple cubes in the same AW may share one or more dimensions. Therefore, a cube is simply a logical object that helps an administrator to build and maintain data in an AW.

After creating cubes, measures, and dimensions, you map the dimensions and stored measures to existing star, snowflake, and normalized relational sources and then load the data. OLAP data can then be queried with simple SQL.

Back to Topic List

Scenario

The source data for this tutorial the OLAPTRAIN schema. OLAPTRAIN is a star schema that was sourced from a base transactional system which contains data for a fictional electronics store. The star schema contains “dimension” tables, which describe the relationships in the data, and “fact” tables, which contain the metrics used to measure performance.

The following are the tables, intended for analysis, that previously have gone through the ETL (Extraction, Transformation, and Loading of heterogeneous data) process:

Table	Description
CHANNELS	Table containing distribution channels for customers purchases.
CUSTOMERS	Table that show who purchased products, and where products are sold for the Geography dimension.
PRODUCTS	Table containing products that are sold by the company.
TIMES	Table containing time periods when products were sold.
SALES_FACT	Stores purchases in dollars, quantity, and price, by channel of distribution, product item, day, and customer.

Designing a Logical Data Model

After examining the relational tables, the available levels, hierarchies, and attributes for each dimension are identified. In addition, the required stored and calculated measures are identified as part of the business requirements definition process. The resulting logical model becomes the design for the OLAP data model.

Identifying Dimensions

Using the source data tables as the primary input, the following dimensions have been identified as requirements for the OLAP data model:

- Channel
- Geography
- Product
- Time

Identifying Levels

When designing your OLAP model, you also determine the level of summarization that you want to load into your cube. You may not necessarily want to replicate the data in your source as a cube. You can always query the detail data (since all of the data is in the Oracle database), by joining the cube to the fact table.

Your business requirements for summary management and analysis purposes should define the lowest level of detail for each dimension in the OLAP cube. You can load data into the cube at any level. After performing a business requirements analysis, the following the levels of summarization within each dimension have been identified as part of the OLAP data model:

- **Channel dimension** has two classes of distribution channels: Direct and Indirect. The children of these two values are the lowest level of detail and will be grouped in the Channel level. From the order of highest level of summarization to the lowest level of detail, levels will be: **All Channels, Class,** and **Channel**.
- **Geography dimension** reflects how company performs customer and geographic analysis along regions. Although the CUSTOMERS dimension table contains the following levels of detail: Region > Country > State-Province > City > Customer, the levels of summarization required for geographic analysis in the OLAP system will be (highest to lowest): **All Regions, Region, Country,** and **State-Province**.
- **Product dimension** will have six levels. These levels reflect the same levels of detail in the source data. From highest to lowest, the OLAP levels are: **All Products, Department, Category, Type, Subtype,** and **Item**.
- **Time dimension** will have four levels (highest to lowest): **All Years, Calendar Year, Calendar Quarter,** and **Month**. Data is available for the years 2005–2007.

Within each dimension, notice that an "All" (Total) level is added as the highest level of summarization. Adding this highest level provides additional flexibility as application users analyze OLAP data.

Identifying Hierarchies

Hierarchies organize the levels within each dimension. To identify hierarchies, you group the levels in the correct order of summarization and in a way that supports the identified types of analysis. You can orgainize levels into any number of hierarchies for each dimension.

In this OLAP data model, only one hierarchy is required for each dimension. The hierachy levels are designed as shown in the table above.

Identifying Measures

Analysis requirements include both stored and calculated measures. Two of the measures are acquired from the fact table, and the remaining measures are created and managed as OLAP calculations:

- Stored Measures
  - Sales
  - Quantity
- Calculated Measures
  - Sales Year-to-Date
  - Sales Year-to-Date Prior Year

Sales Year-to-Date Prior Year % Change
Sales Prior Year % Change
Sales Prior Period
Sales Prior Period % Change
Sales Rank in Product Level
Sales Rank in Product Parent
Share of Product Sales within Parent
Share of Product Sales within Total

Back to Topic List

Prerequisites

Before you perform this tutorial, you should:

1. Install Oracle Database 11.2 or 12.1 with the OLAP Option.

2. Download Analytic Workspace Manager from the Oracle OLAP page on OTN, and unzip it into any directory on your Windows machine.

3. Download and install the OLAPTRAIN schema following the instructions in Installing the Oracle OLAP 11g Sample Schema.

   Notes: The Sample Schema installation package includes two parts:

   1. Installing the base OLAPTRAIN schema
   2. Installing the SALESTRACK analytic workspace in the OLAPTRAIN schema

   Only complete the first part -- installing the base OLAPTRAIN schema.

4. a. Download olaptrain_templates.zip to a location on the machine where AWM is installed.

   b. Unzip the files.

   The resulting directory structure is: **`<your_path>\templates\calcs`**

   The files in the **...\templates** directory, and the **...\templates\calcs** directory are used later in this tutorial.

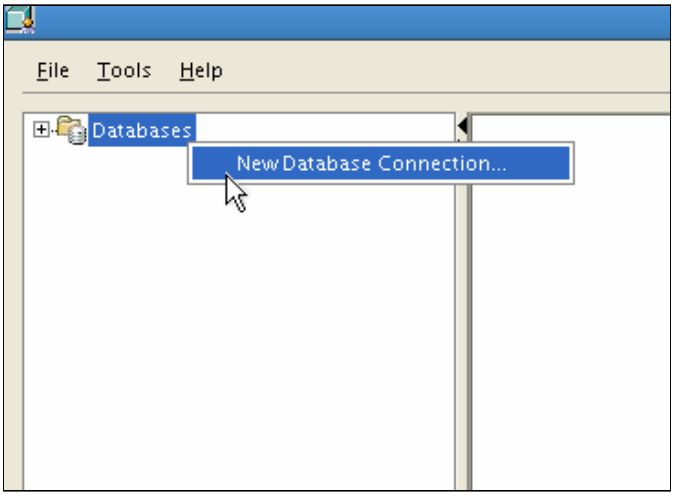
Back to Topic List

Creating an Analytic Workspace

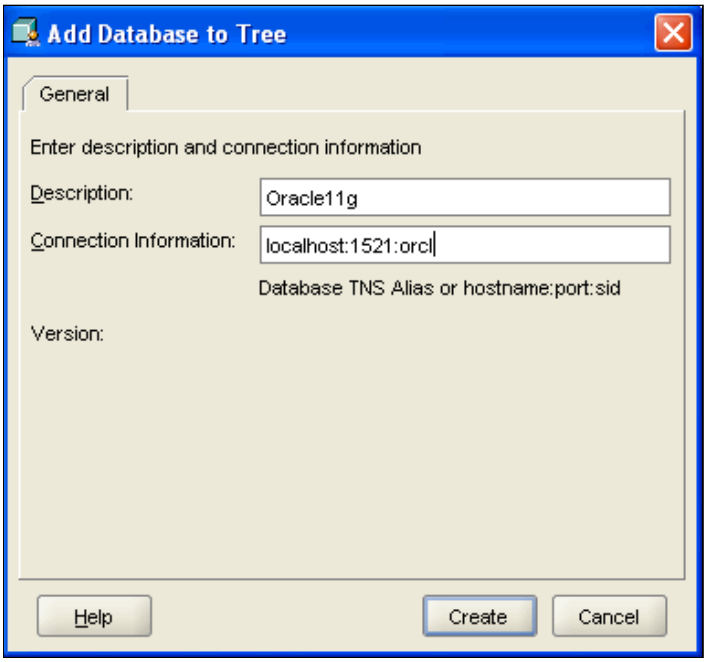
An analytic workspace is a container for multidimensional data objects and procedures written in OLAP DML. It is created using the AWM tool. Perform the following steps:

1. Launch AWM either by double-clicking on the awm.jar file, or from your desktop shortcut.

2. Right-click **Databases** and select **Add Database to tree**.

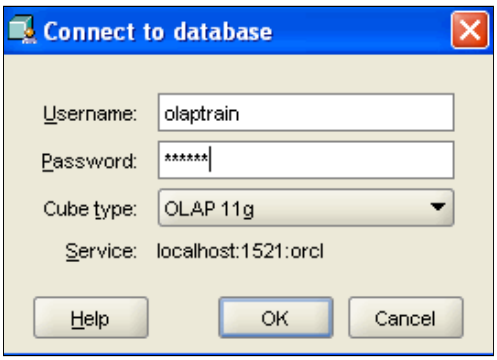


3. Enter **Oracle** in the Description field and **<hostname>:1521:<SID>** in the Connection Information field and click **Create**.

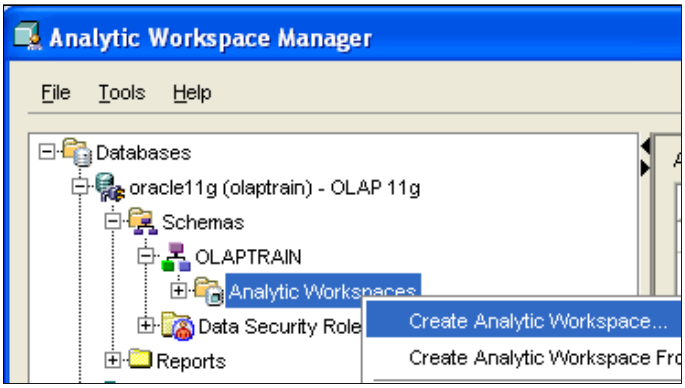


4. Click the Plus sign (+) next to **Oracle**.

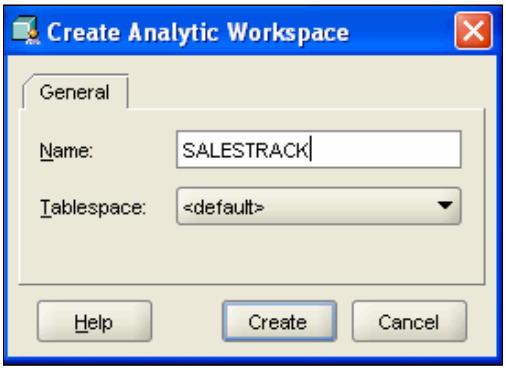
5. Enter **olaptrain** as the Username and **oracle** as the Password. Then click **OK**.



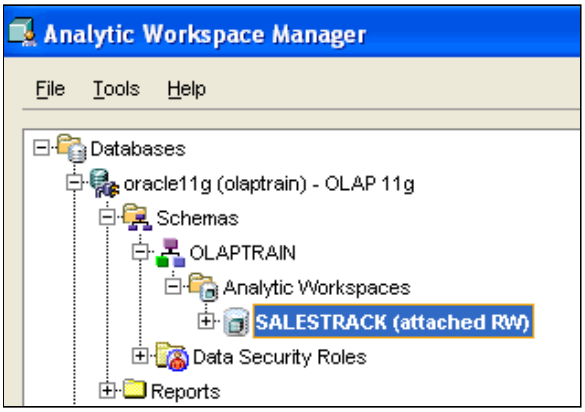
6. Expand **Schemas > OLAPTRAIN**. Right-click **Analytic Workspaces** and select **Create Analytic Workspace**.



7. Enter **SALESTRACK** as the Name and click **Create**.



8. Your Analytic Workspace has been created.



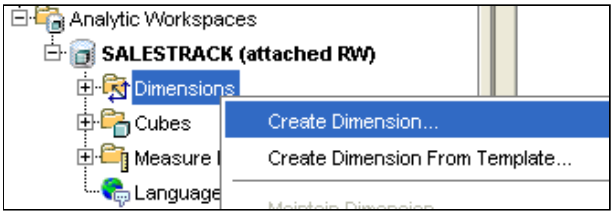
Back to Topic List

Defining Dimensions and Levels

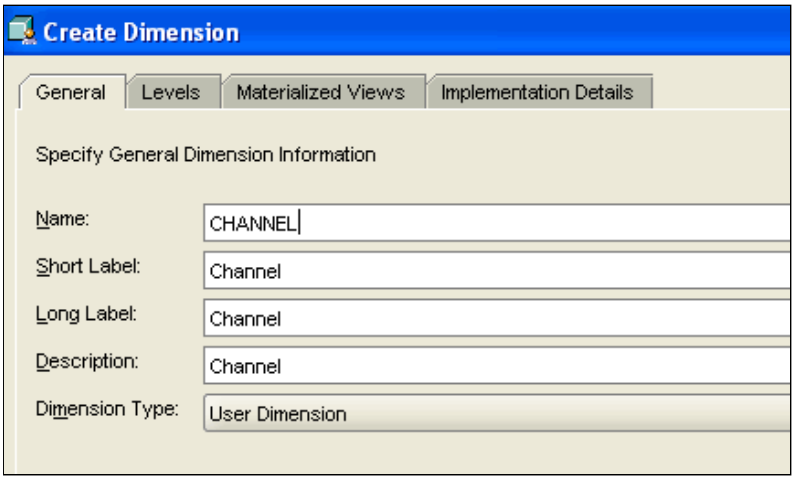
Dimensions are lists of unique members that identify and categorize data. They form the edges of a cube, and thus the measures within the cube. Dimensions may contain levels, hierarchies, and attributes. You may define levels at the same time that create a dimension, or you may define the levels later.

You can define dimensions either as 'User' or as 'Time' dimension type. Business analysis is performed on historical data, so fully defined time periods are vital. For a Time type dimension, your source data must have columns for period end dates and time span. These required attributes support OLAP time-series analysis, such as comparisons with earlier time periods. If this information is not available, then you can define Time as a normal dimension, but it does not support time-based analysis.

1. Right-click the **Dimensions** folder and select **Create Dimension**.



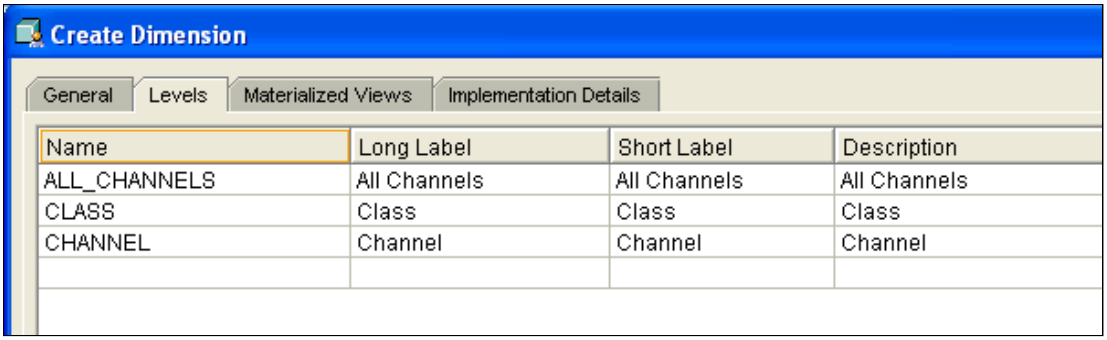
2. At the default General tab in the Create Dimension dialog box, enter **CHANNEL** as the name and select **User Dimension** as dimension type.



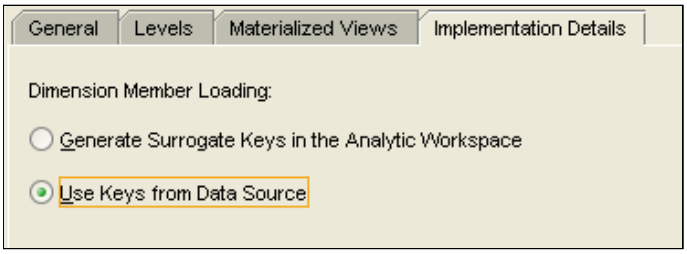
3. In the Levels tab, enter the following three levels:

   - **ALL_CHANNELS**
   - **CLASS**
   - **CHANNEL**

   Note: the Label and Description fields are auto-filled

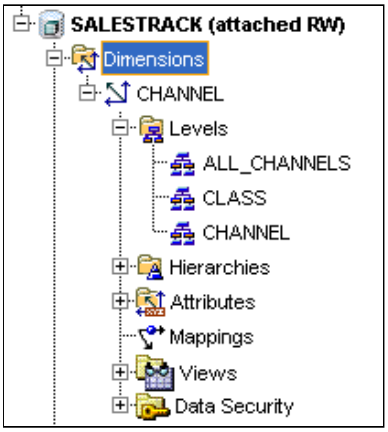


4. In the Implementation Details tab, select **Use Keys from Data Source**.



   Then, click **Create**.

5. Your dimension, and its associated levels, have been created.



Back to Topic List

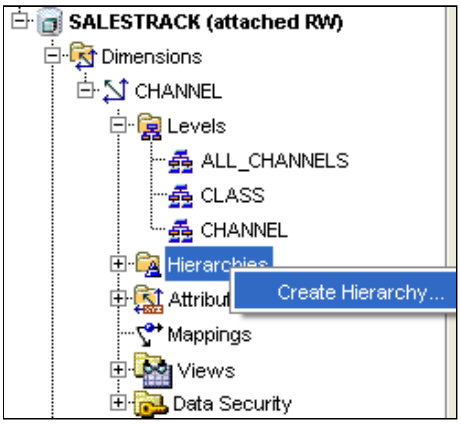
Defining Hierarchies

For business analysis, data is typically summarized at various levels. For example, your database may contain daily snapshots of a transactional database. Days are thus the base level. However, you might summarize this data at the monthy, quarterly, and yearly levels.

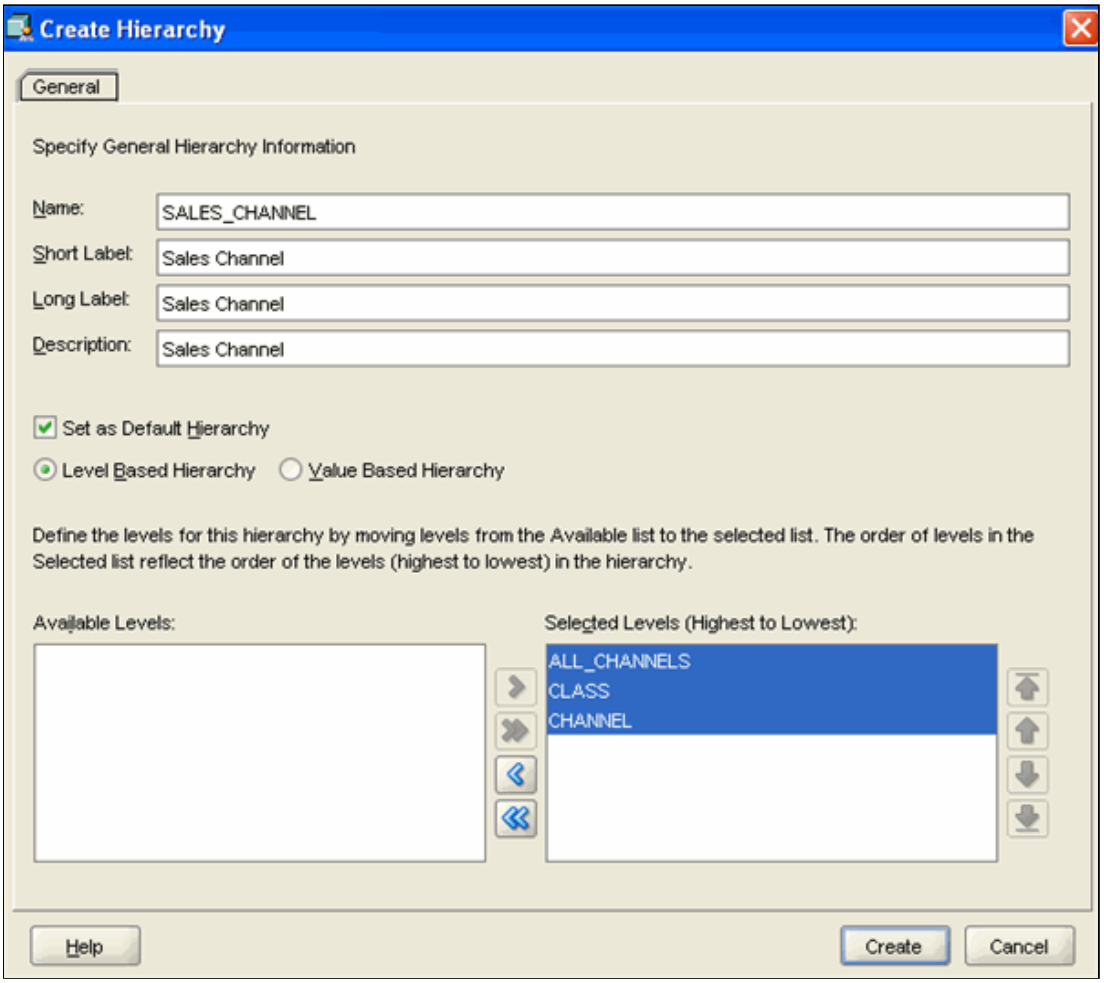
A hierarchy is a logical structure that uses ordered levels as a means of organizing data. It can be used to define data aggregation; for example, in a time dimension, a hierarchy might be used to aggregate data from the month level to the quarter level to the year level. A hierarchy can be used to define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

Dimensions can have one or more hierarchies. If you define multiple hierarchies, one of them must be defined as the default hierarchy.

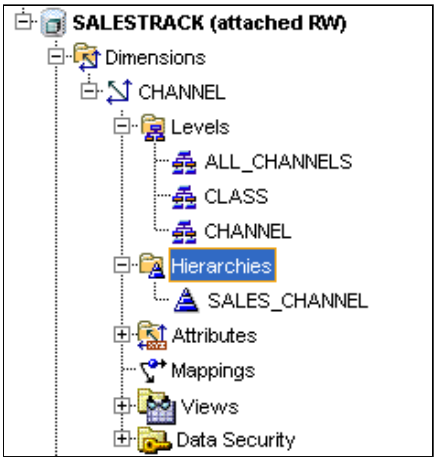
1. Right-click the **Hierarchies** folder, then select **Create Hierarchy.**



2. In the Create Hierarchy window, enter **SALES_CHANNEL** as the name. Click the Add All (>>) tool to select all the levels and click **Create**.



3. The new SALES_CHANNEL hierarchy appears as an item in the Hierarchies folder.



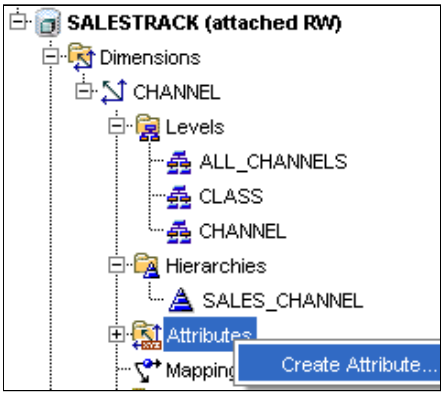
Back to Topic List

Defining and Reviewing Attributes

Attributes provide information about the individual members of a dimension. They are used for labeling data displays and selecting data. All dimensions are created with long and short description attributes. Time dimensions also have time-span and end-date attributes. In addition, you can create your own user attributes

In this section, you create a CHANNEL_TYPE attribute, and also review the description attributes for the CHANNEL dimension you just created. Perform the following steps:

1. Rick-click the **Attributes** folder, then select **Create Attribute**.



2. In the Create Attribute dialog, select or enter the following:

   a. Name = **CHANNEL_TYPE**

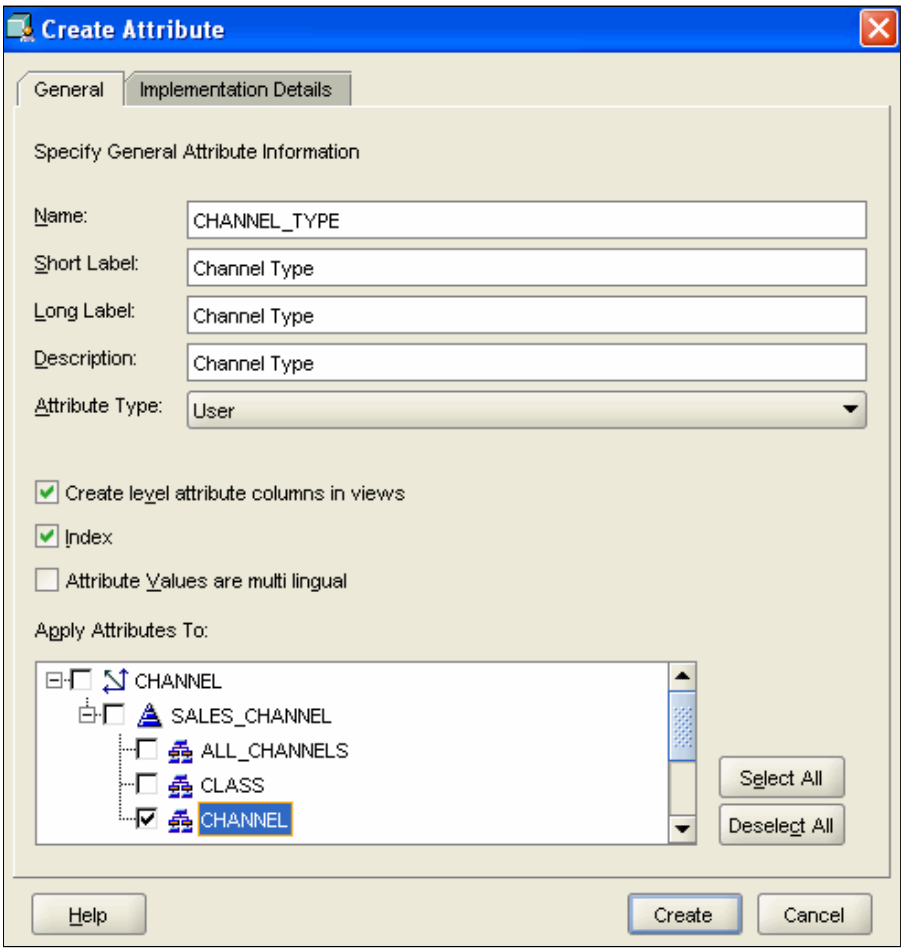
   b. Attribute Type = **User**

   c. In the “Apply Attributes to” box:

   - Drill on the Channel dimension.
   - Deselect the Channel dimension check box.
   - Select only the **CHANNEL** level check box (the lowest level).

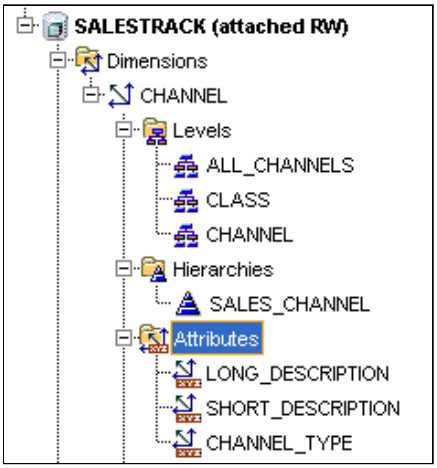
   Note: The Channel Type attribute only applies to the lowest level in the Sales Channel hierarchy.

   The Create Attribute dialog box should look like this:

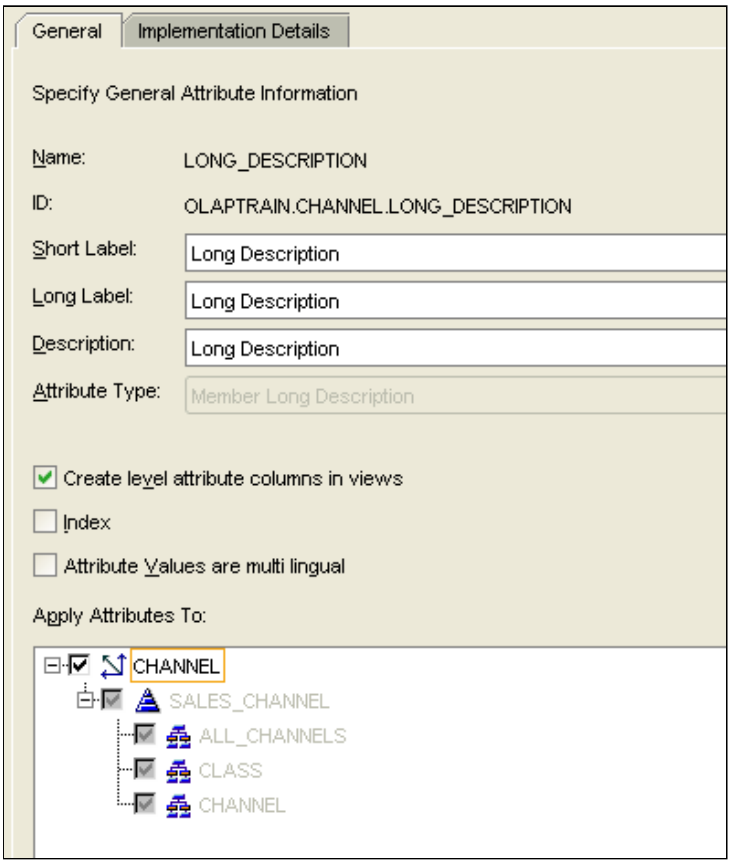


   Click **Create**.

3. Expand the **Attributes** folder to view the Channel dimension attributes.



4. Select the **LONG_DESCRIPTION** attribute. In the right-hand pane, notice that description attributes are defined for all levels in the hierarchy, in contrast to the user attribute that you just created..



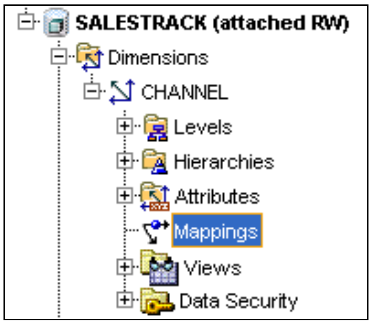
Back to Topic List

Mapping Dimensions to Relational Sources

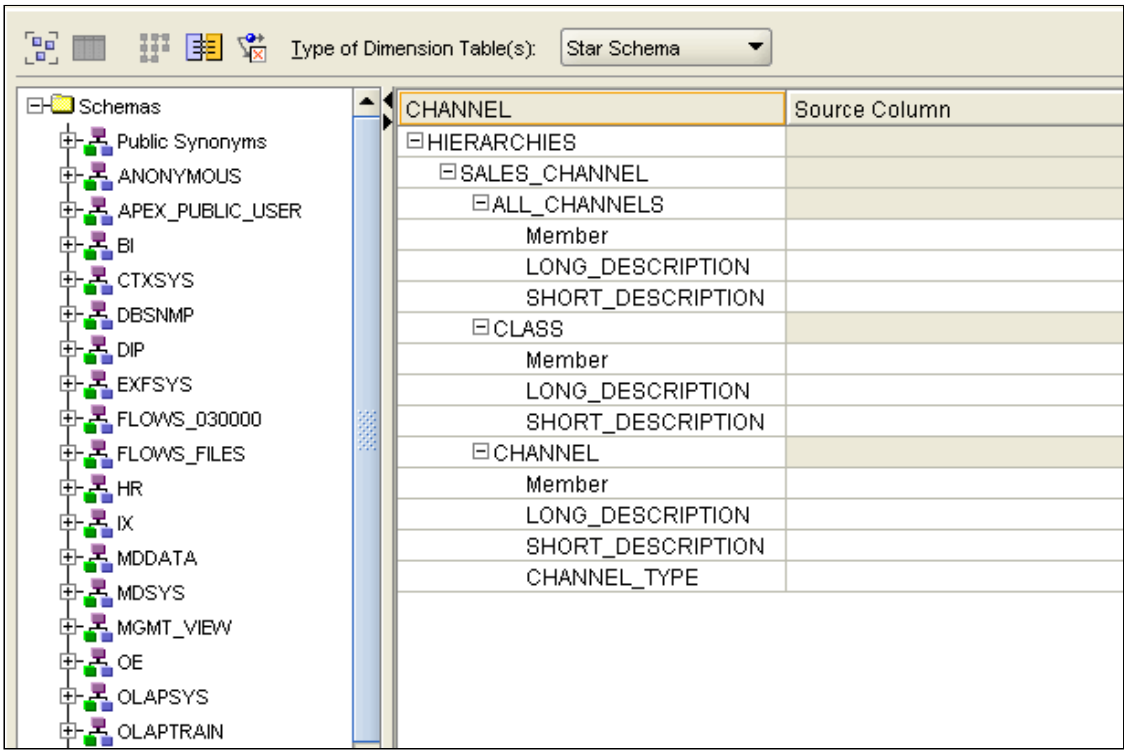
After creating OLAP data objects, you map them to tables and views in Oracle Database. You map the key column in the dimension table to the Member attribute in the OLAP dimension. In addition, you map the appropriate attribute columns in the dimension table to the associated OLAP dimension attributes.

Afterward, you can load data into your analytic workspace using the Maintain Analytic Workspace wizard.

1. Expand the **CHANNEL** dimension and click **Mappings.**



Result: Two panes appear to the right: of the navigator -- the Schemas pane, and the Mapping pane. In the Table Mapping view (right-hand pane), the Source Column fields are initially blank, as shown here:



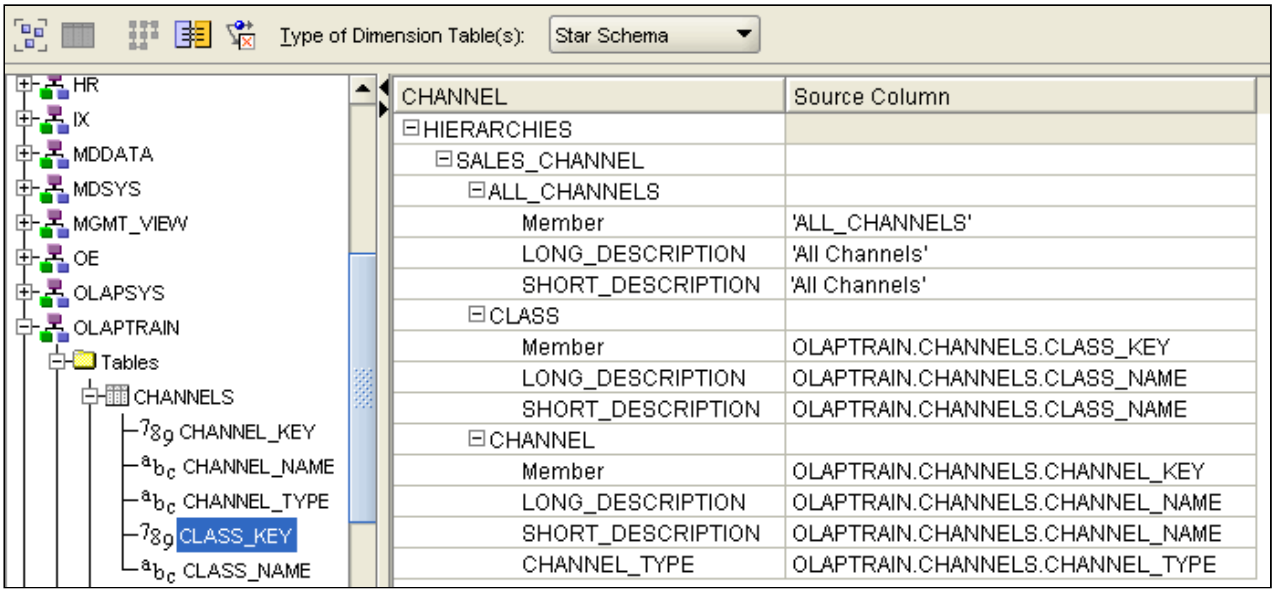
2. Ensure that **Star Schema** is selected as the Type of Dimension Table, as shown in the previous step.

3. In the source schema pane, expand **OLAPTRAIN** > **Tables** > **CHANNELS**. Then, drag the source columns from the Schema pane to the Mapping pane for the CHANNEL and CLASS levels as shown in the picture below.

   The ALL_CHANNELS level in the hierarchy does not contain a source data column. For "All/Total" hierarchy levels, you can enter constants or single row SQL functions. Enter the following constants for the ALL_CHANNELS level (single quotes are required for text literals):

   - Member = **'ALL_CHANNELS'**
   - Description attributes = **'All Channels'**

   The resulting mapping should look like this:



Note: The "All/Total" value ensures that there is a single node at the top of the hierarchy that will be the summary of the data for that dimension.

4. In the lower right corner of the mapping pane, click **Apply**.

Result: The Channel dimension is ready to have data loaded. Although you could load the dimension data now, you will perform this step later in the tutorial.

Back to Topic List

Using Templates to Define Data Objects

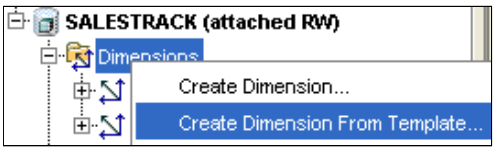
The template feature in Analytic Workspace Manager saves the definition of the OLAP data objects as an XML file. Using a saved template, you can create a new analytic workspace, dimension, cube, and measure exactly like an existing object, with or without mappings. Templates do not include the data, only the definition of the object.

Templates allow you to:

- Share analytic workspace designs with other users.
- Transfer object definitions to other schema or instances.
- Persist object definitions outside database.
- Place object definitions in source control.

In this section, you create three dimensions, GEOGRAPHY, PRODUCT and TIME from previously saved templates. Perform the following steps:

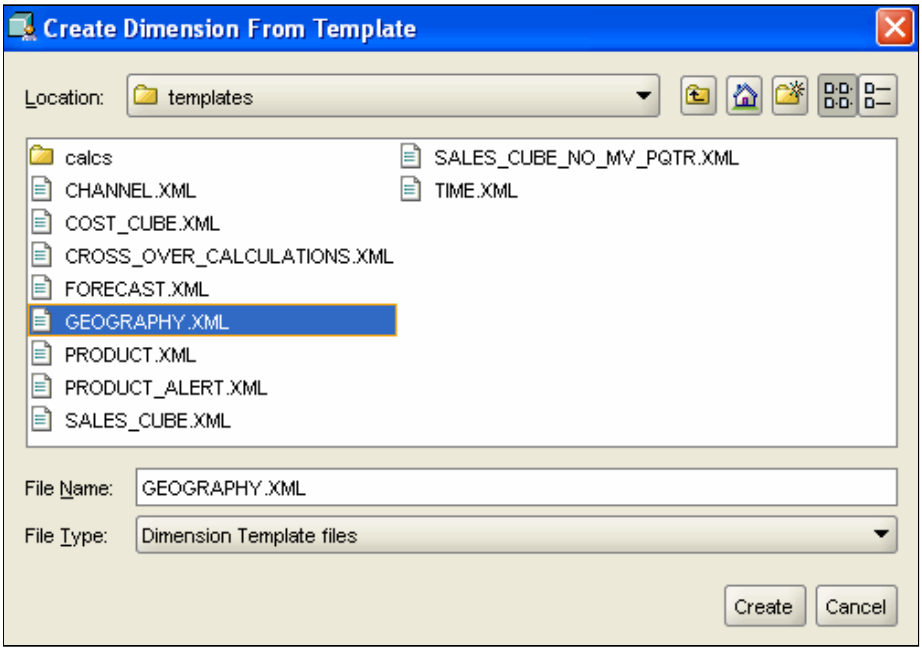
1. Right-click the **Dimensions** folder, then select **Create Dimension From Template.**



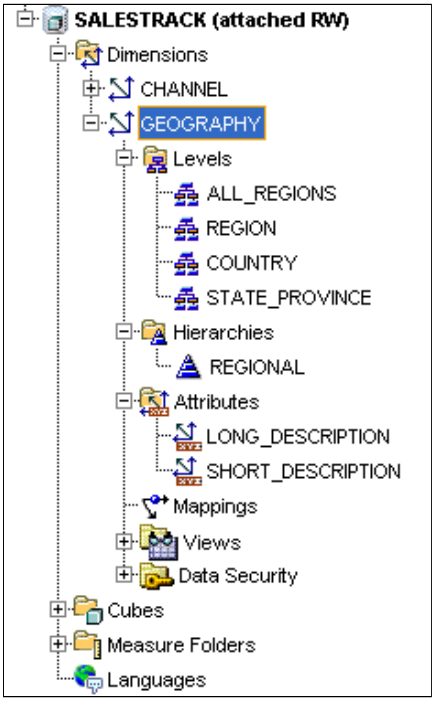
2. At the Create Dimensions From Template dialog box, locate the ...\templates directory, where you installed the olaptrain template files.

For example: `c:<your_path>\templates`

Then, select **GEOGRAPHY.XML** in the templates directory and click **Create**.

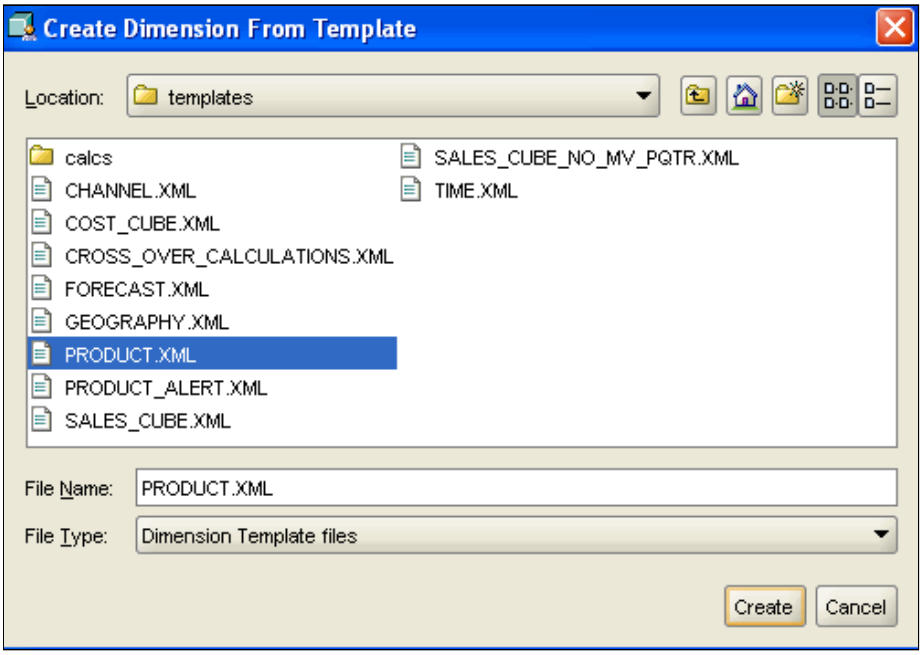


3. The new GEOGRAPHY dimension appears under the Dimensions folder. Drill on **Levels**, **Hierarchies**, and **Attributes** to view its elements.

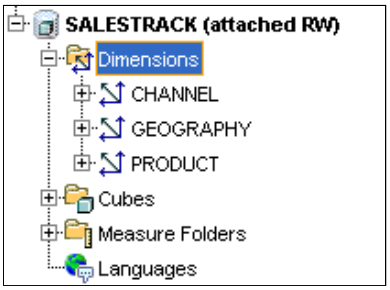


4. To create the PRODUCT dimension, right-click the **Dimensions** folder, then select **Create Dimension From Template**, as you did previously in step 1.

5. At the Create Dimensions From Template dialog box, locate the **PRODUCT.XML** file and click **Create**.

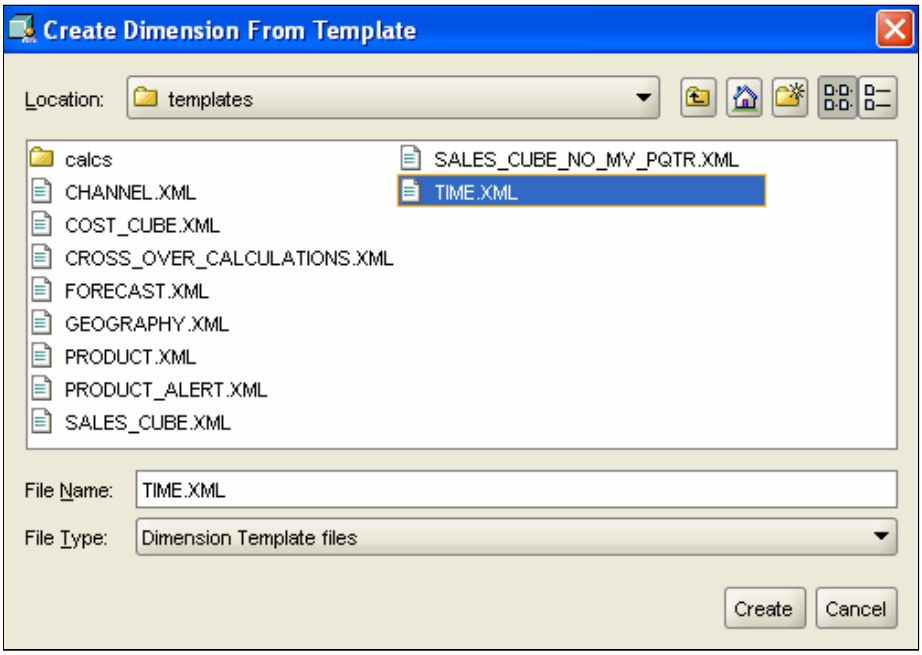


6. The new PRODUCT dimension appears under the Dimensions folder.



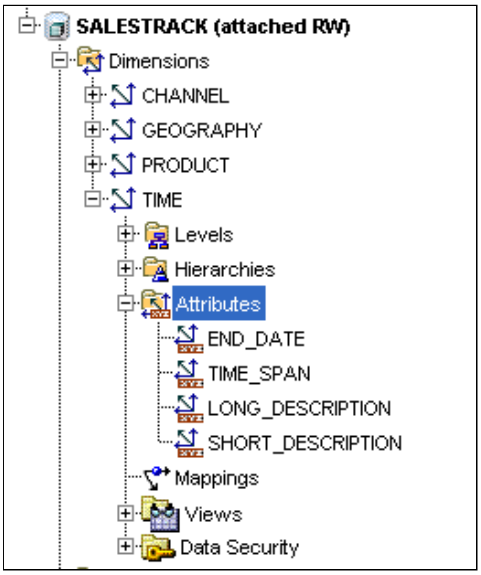
7. Now, create the TIME dimension by right-clicking the **Dimensions** folder and selecting **Create Dimension From Template.**

8. At the Create Dimensions From Template dialog box, locate the **TIME.XML** file and click **Create**.



Result: The new TIME dimension appears under the Dimensions folder.

9. In the navigator, drill on **TIME** > **Attributes**.



Notice that there are two special attributes -- END_DATE and TIME_SPAN -- have been created for the TIME dimension. Since this dimension was defined as a "Time" type, these attributes are automatically created. They must be mapped to apppropriate source data columns for certain OLAP time series analysis features to be enabled. You will leverage these attributes when you create time series calculations later in this tutorial.

10. The Geography, Product and Time templates all included mappings. To view the mappings for the Time dimension, click **Mappings** under TIME in the navigator.

TIME	Source Column
HIERARCHIES	
CALENDAR	
ALL_YEARS	
Member	'ALL_YEARS'
LONG_DESCRIPTION	'All Years'
SHORT_DESCRIPTION	'All Years'
END_DATE	TO_DATE('2010-DEC-31', 'YYYY-MON-DD')
TIME_SPAN	3000
CALENDAR_YEAR	
Member	OLAPTRAIN.TIMES.CALENDAR_YEAR_ID
LONG_DESCRIPTION	OLAPTRAIN.TIMES.CALENDAR_YEAR_NAME
SHORT_DESCRIPTION	OLAPTRAIN.TIMES.CALENDAR_YEAR_NAME
END_DATE	OLAPTRAIN.TIMES.CALENDAR_YEAR_END_DATE
TIME_SPAN	OLAPTRAIN.TIMES.CALENDAR_YEAR_TIME_SPAN
CALENDAR_QUARTER	
Member	OLAPTRAIN.TIMES.CALENDAR_QUARTER_ID
LONG_DESCRIPTION	OLAPTRAIN.TIMES.CALENDAR_QUARTER_NAME
SHORT_DESCRIPTION	OLAPTRAIN.TIMES.CALENDAR_QUARTER_NAME
END_DATE	OLAPTRAIN.TIMES.CALENDAR_QUARTER_END_DATE
TIME_SPAN	OLAPTRAIN.TIMES.CALENDAR_QUARTER_TIME_SPAN
MONTH	
Member	OLAPTRAIN.TIMES.MONTH_ID
LONG_DESCRIPTION	OLAPTRAIN.TIMES.MONTH_NAME
SHORT_DESCRIPTION	OLAPTRAIN.TIMES.MONTH_NAME
END_DATE	OLAPTRAIN.TIMES.MONTH_END_DATE
TIME_SPAN	OLAPTRAIN.TIMES.MONTH_TIME_SPAN

As with the other the "All/Total" level in the hierarchy is mapped to either constants or single-row SQL functions.

You can examine the mappings for any of these dimensions by clicking on the Mappings tab under the dimension node in the navigator.

Back to Topic List

Defining Cubes

In Oracle OLAP, a Cube provides a convenient way of collecting measures of the same dimensionality. Therefore, a cube is simply an object that helps an administrator to build and maintain an AW.

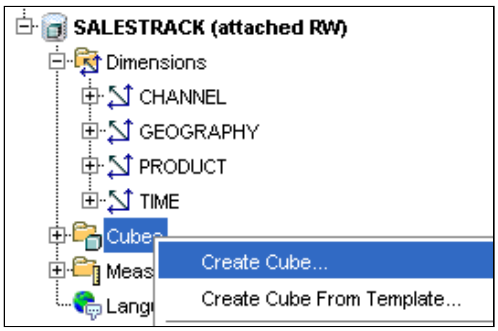
Cubes aid in the definition of measures with common characteristics, including the following:

- The edges of a Cube are defined by its dimensions. If multiple measures have the same dimensionality, it is likely that they will be defined in the same cube..
- Measures that share sparsity patterns and aggregation rules are commonly defined in the same Cube.
- Measures in the same Cube have the same relationships to other logical objects and can easily be analyzed and displayed together.
- A particular AW may contain more than one Cube, and each cube may describe a different dimensional shape.
- Multiple Cubes in the same AW may share one or more dimensions.

For example, sales data can be organized into a cube, whose edges contain values from the channel, geography, product, and time dimensions and whose body contains measures that might include dollar sales, unit sales, and a range calculated measures based on sales and quantity sold.

Perform the following steps to create a cube that will be used to organize a variety of sales measures:

1. Right click the **Cubes** folder, then click **Create Cube.**



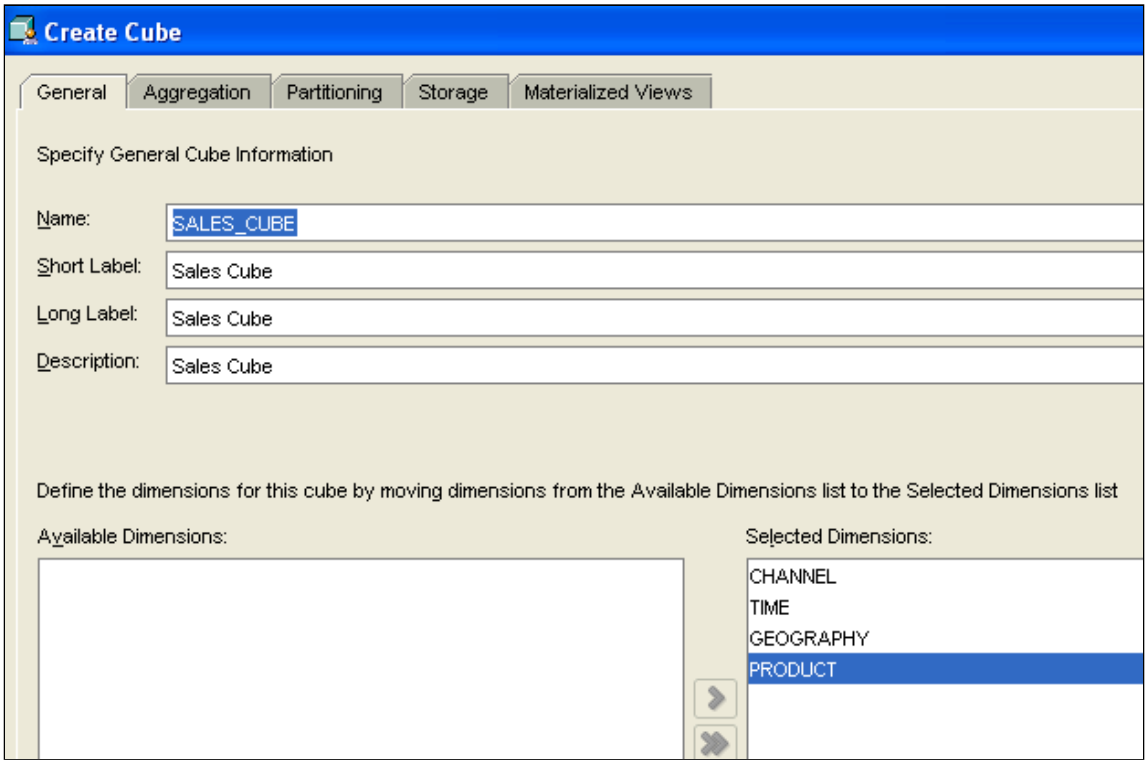
2. In the General tab of the Create Cube window, specify the following:

   a. Name: **SALES_CUBE**

   b. Use the Add tool (>) to selected dimensions in the following order:

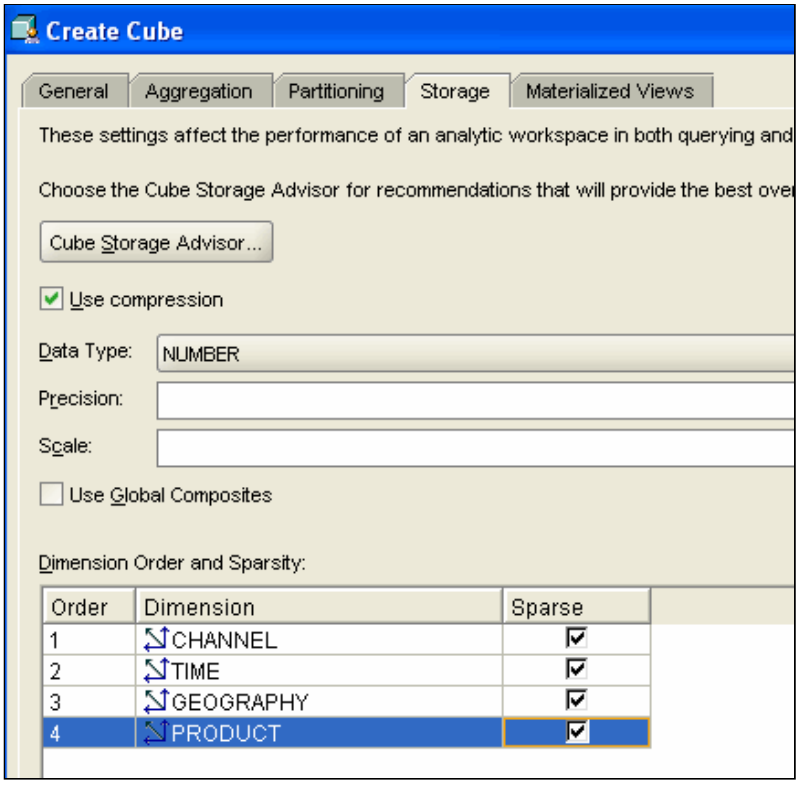
   - **CHANNEL**
   - **TIME**
   - **GEOGRAPHY**
   - **PRODUCT**

   Result: the Create Cube window should look like this:



Notes: The order in which the dimensions are listed in a cube may affect performance because it determines the way the data is stored on disk. In general, when you dimension a cube, the first dimension in a cube has the fewest number of dimension members, and the last dimension has the largest number of dimension members. This is the case in the OLAPTRAIN schema.

3. Select the Storage tab. Accept the default option to **Use compression**, and then enable the Sparse option for all dimensions, as shown here

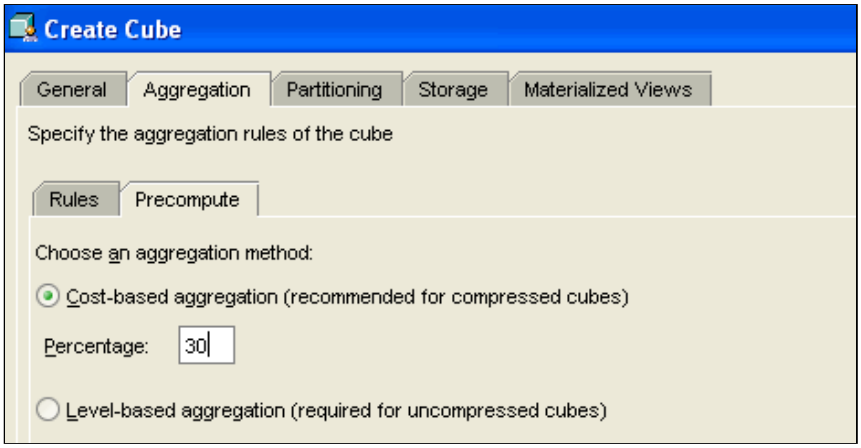


Notes:

- What is Sparsity? When there are a large number of empty cells in a cube, the cube is said to be “sparse.” This is very common in dimensional data models. Most commonly, all dimensions are marked as sparse. When one or more dimensions as marked as sparse, OLAP creates a special index for the cube that automatically manages sparsity.

- The Compression feature can be used to significantly reduce the size of cubes and improve performance of both data loads and queries. Since most dimensional data is sparse, the Compression option is selected as a default.

4. Select the Aggregation tab. Then, in the Precompute sub-tab, specify a value of **30** for Cost-based aggregation, as shown here:

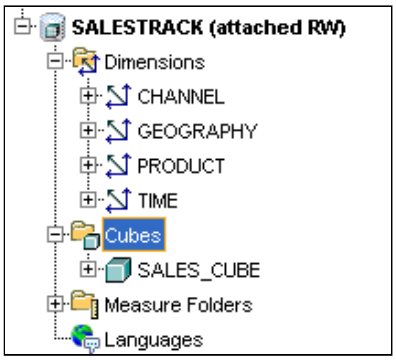


Notes:

- Cost-based aggregation enables you to select compression for your cube. Specify a percentage value and the database will precompute and store the most costly aggregate values based on your input.
- Using a setting of 30 causes a larger percentage of cube data to be aggregated and stored than the default setting of 20.

5. At the bottom of the Create Cube dialog box, click **Create**.

Result: the SALES_CUBE node appears in the navigator under Cubes.



Back to Topic List

Creating Measures

You can create two types of measures in a cube: Stored (or Base) measures, and Calculated measures. Every measure that belongs to a particular cube shares the characteristics that were defined for the cube.

Stored Measures

Base measures store the facts collected about your business. When you create base measures in your OLAP data model, you will map them to source data just as you have done with dimensions.

Calculated Measures

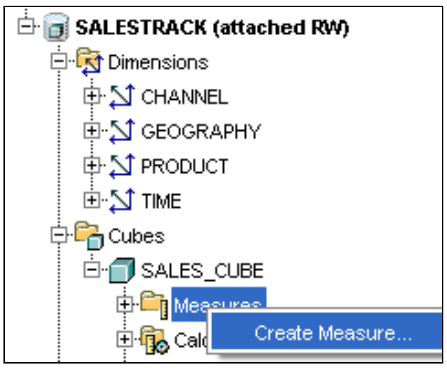
One of the powerful features of the Oracle OLAP technology is the ability to efficiently and easily generate business calculations of data held in the database. In any OLAP implementation, the number of calculated measures greatly exceeds the number of stored measures.

OLAP calcuated measures are derived from base measures or other calculated measures.These calculations are computed dynamically as users query the data. Calculations are automatically exposed as columns in a cube view – making it very easy for users to leverage the rich analytic functionality through very simple SQL.

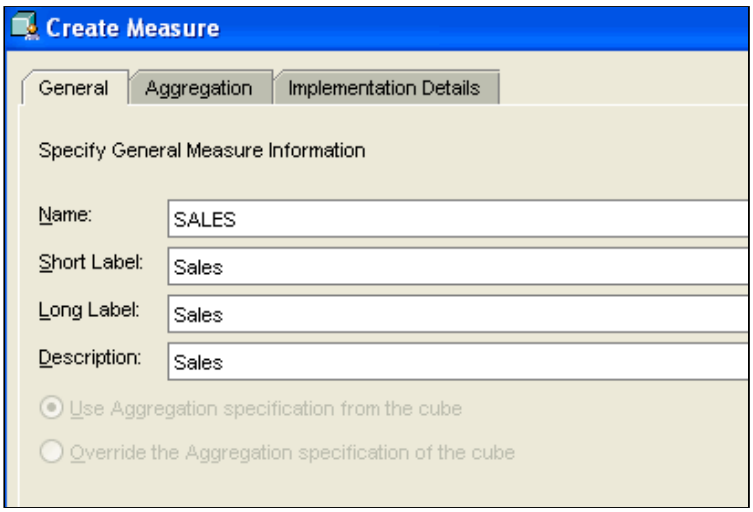
AWM makes it very easy to define calculated measures using a graphical Calculation Builder. The Calculation Builder contains pre-defined examples for many common business calculation types. You select the calculation type you want, and then modify the example to create exactly the calculation that you need.

In this section, you will create two stored measures and ten calculated measures. Three of the calculated measures are created using the Calculation Builder, and seven are created using XML template files.

1. In the navigator, drill on **SALES_CUBE**. Then, right-click on the **Measures** folder and select **Create Measure**.

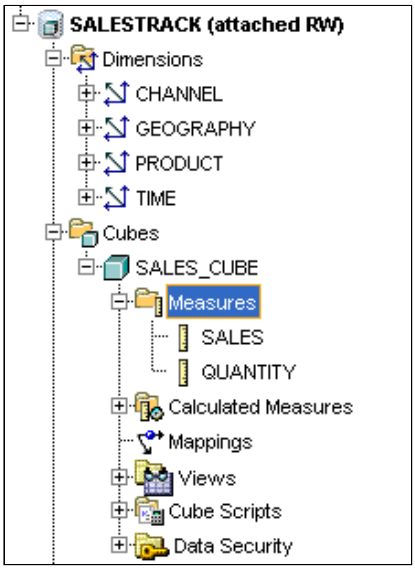


2. At the Create Measure dialog box, enter **SALES** as the name and click **Create**.

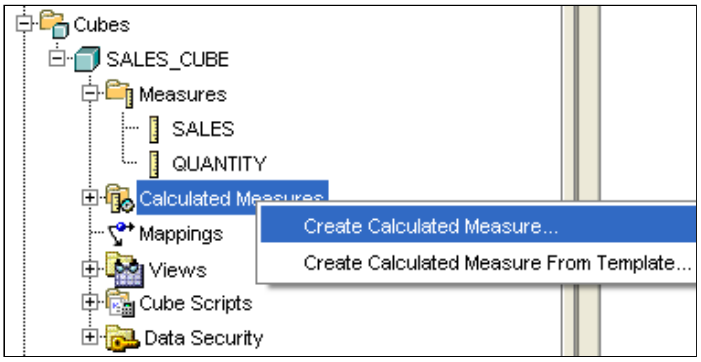


3. Using the same techniques described in steps 1 and 2, create a second measure named **QUANTITY**.

   In the navigator, drill on **Measures**. You should see the following:



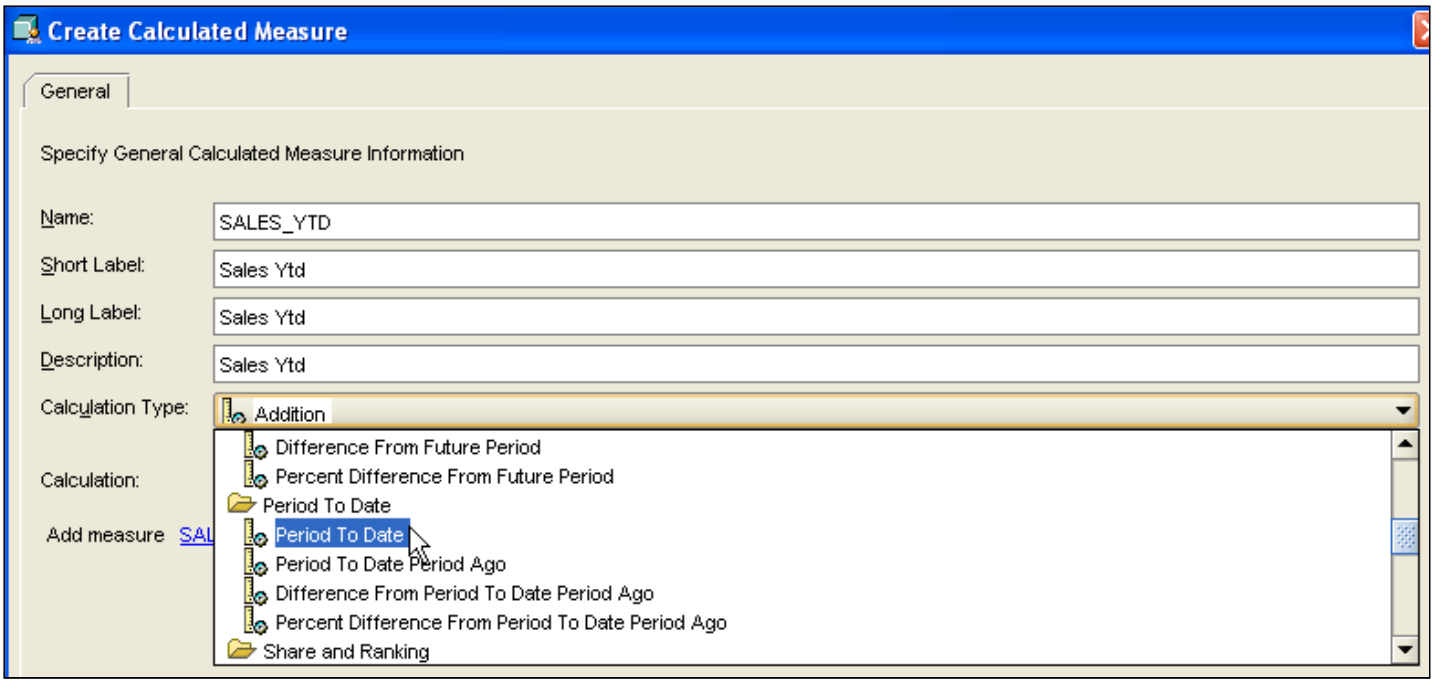
4. In the navigator, right-click on the **Calculated Measures** folder and select **Create Calculated Measure**.



5. In the Create Calculated Measure window, enter or select the following:

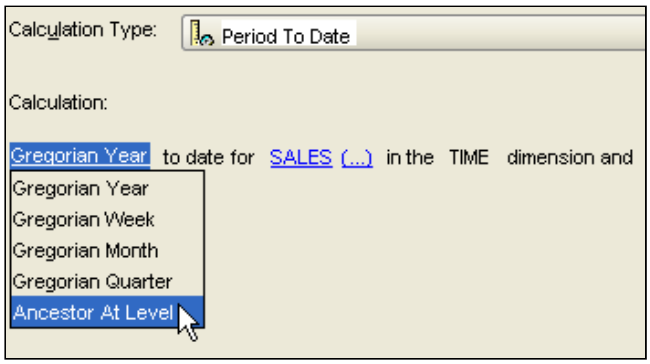
a) Name = **SALES_YTD** (the Name field is automatically all caps, and the Label and Description fields are auto-filled)

b) Calculation Type = **Period to Date**



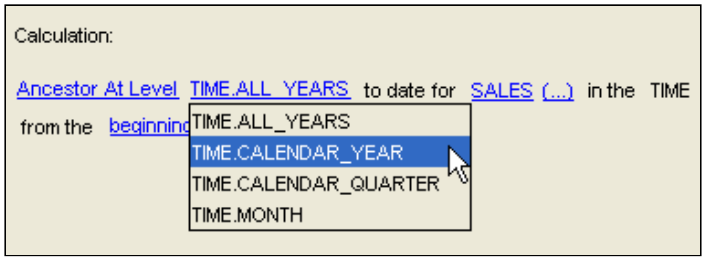
c) In the Calculation inputs section, select the following:

- First hyperlink = **Ancestor At Level**

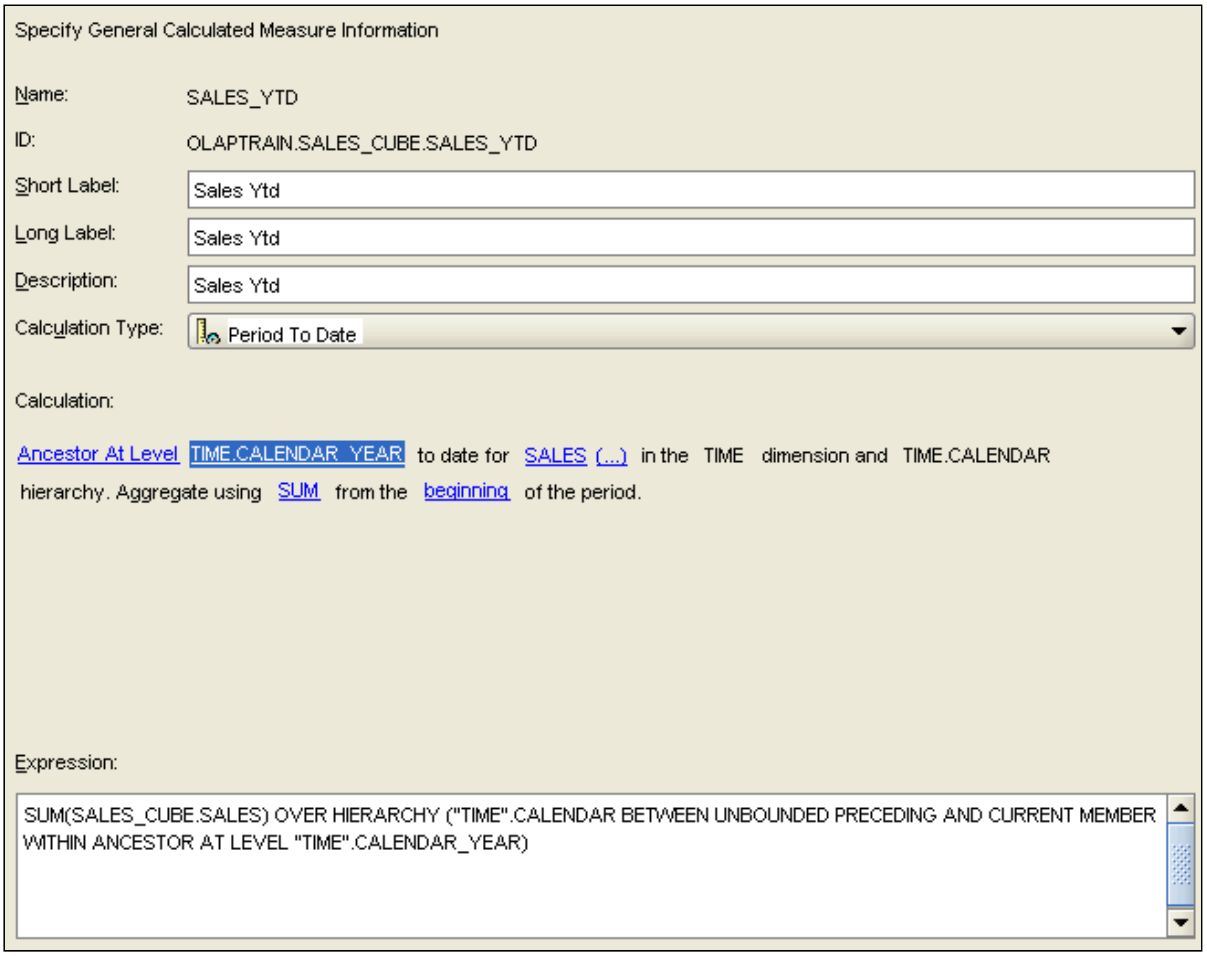


Result: A new hyperlink appears next to Ancestor At Level hyperlink

- Second hyperlink = **TIME.CALENDAR_YEAR**

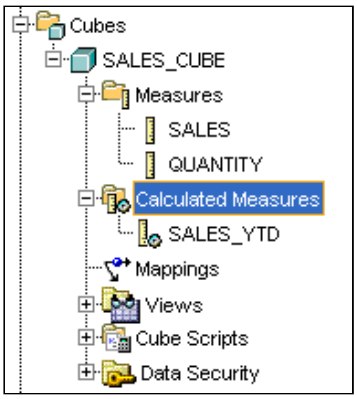


Result: The Create Calculated Measure window should now look like this:



d) Click **Create**.

6. Drill on the **Calculated Measures** node. Result: The SALES_YTD calculation appears.



7. Create a YTD calculation for the prior year. This facilitates year over year comparisons.

   Right-click the **Calculated Measures** folder again, then select **Create Calculated Measure.** In the Create Calculated Measure window, enter or select the following:

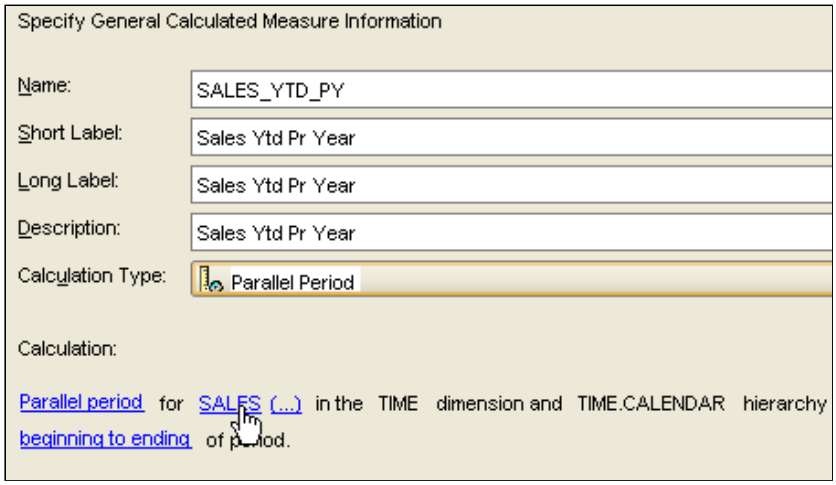
a) Name = **SALES_YTD_PY**

b) All Label and Description boxes = **Sales Ytd Pr Year**

c) Calculation Type = **Parallel Period**

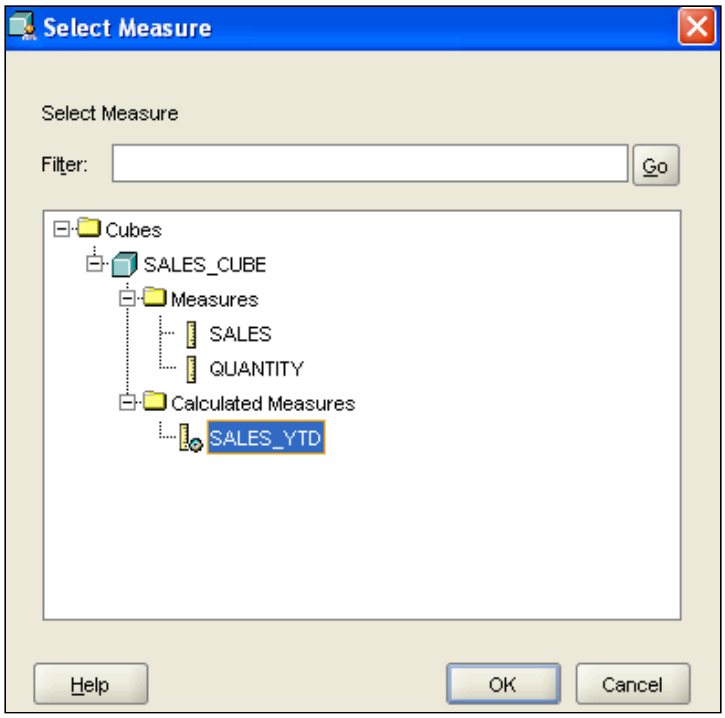
d) In the Calculation inputs section::

- Click the **SALES** hyperlink (this is the second hyperlink):



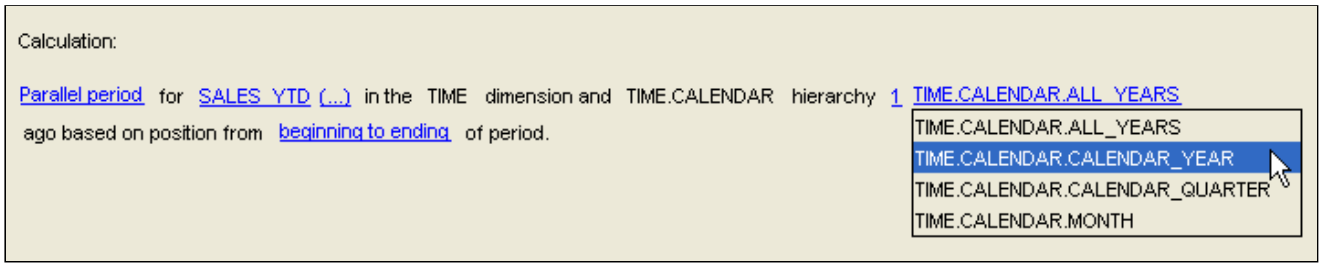
Result: The Select Measure window appears.

- Select **SALES_YTD** (as shown below) and then click **OK**.



Result: The Calculation updates with the selected measure.

- Click the TIME.CALENDAR.ALL_YEARS hyperlink and select **TIME.CALENDAR. CALENDAR_YEAR** from the list, as shown here:



e) Click **Create**.

Result: The SALES_YTD_PY calculation appears below the Calculated Measures node in the Navigator.

8. Create a third calculation that measures the percent change in Year-To-Date sales when compared to the previous year. Right-click on the **Calculated Measures** folder and select **Create Calculated Measure**.

9. In the Create Calculated Measure window, enter or select the following:

a) Name = **SALES_YTD_PY_PCT_CHG**

b) All Label and Description boxes = **Sales Ytd Pr Yr Pct Chg**

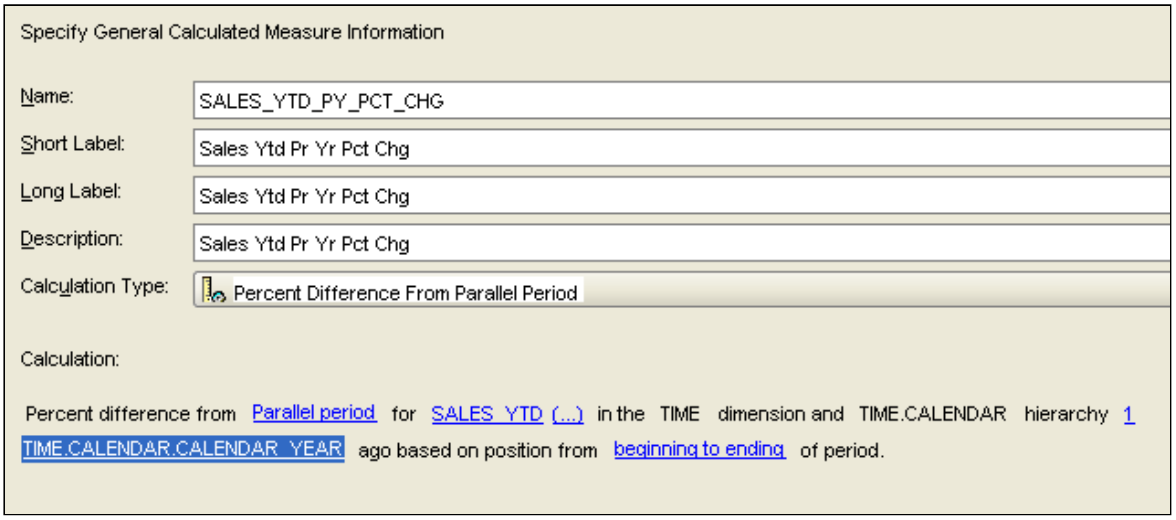
c) Calculation Type = **Percent Difference From Parallel Period**

d) In the Calculation inputs section, click the **SALES** hyperlink.

e) In the Select Measure window, select **SALES_YTD** and click **OK**.

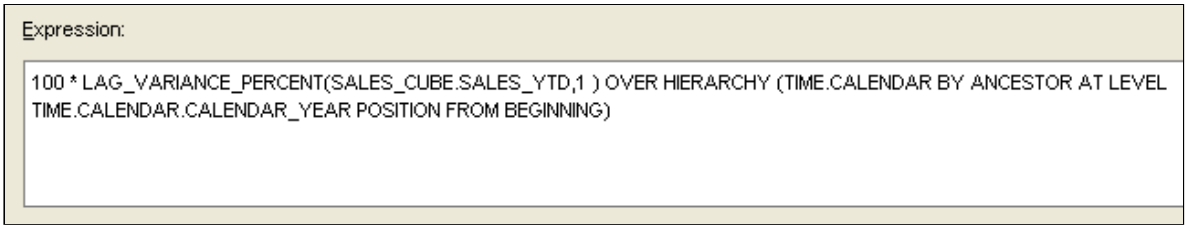
f) Click the **TIME.CALENDAR.ALL_YEARS** hyperlink and select **TIME.CALENDAR. CALENDAR_YEAR** from the list.

The calculation should now look like this:



g) In the Expression field, multiply the equation by 100 by adding the following syntax to the beginning of the expression: **100 ***

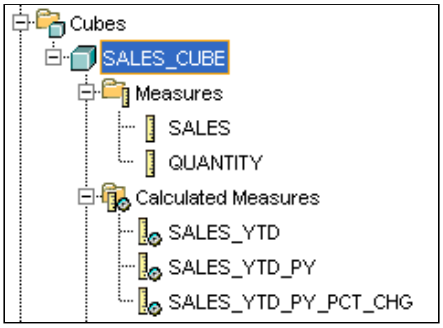
The expression should now look like this:



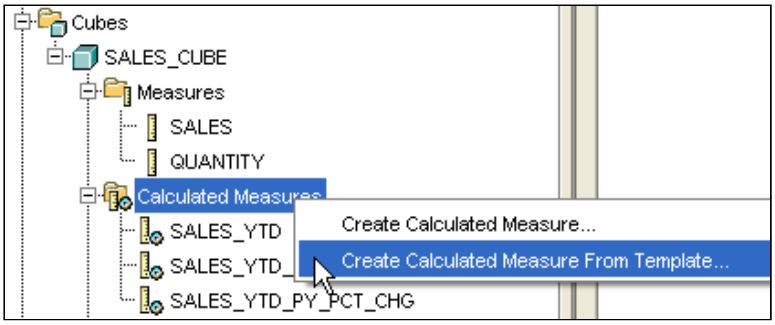
Note: This additional syntax will cause the calculation to display percentage figures in whole numbers.

h) Click **Create**.

Result: The Sales Cube now contains the following measures:



10. Next, create a calculated measure using an XML template. Right-click on the **Calculated Measures** folder and select **Create Calculated Measure from Template**.

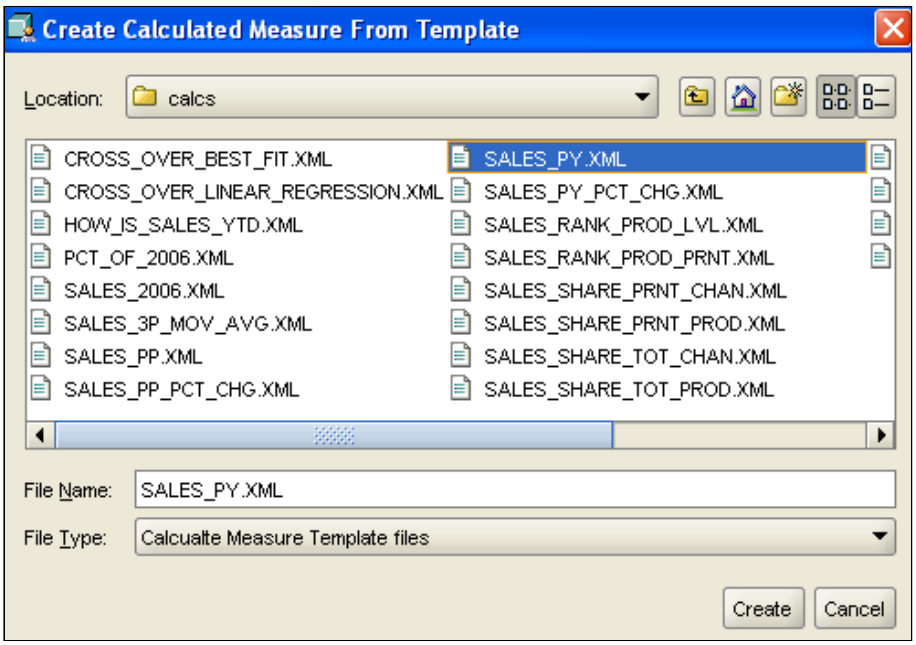


11. In the Create Calculated Measure from Template window:

a. Navigate to the ...\templates\calcs directory, where you installed the olaptrain template files.

For example: **`c:<your_path>\templates\calcs`**

b. Then, select **SALES_PY.XML**

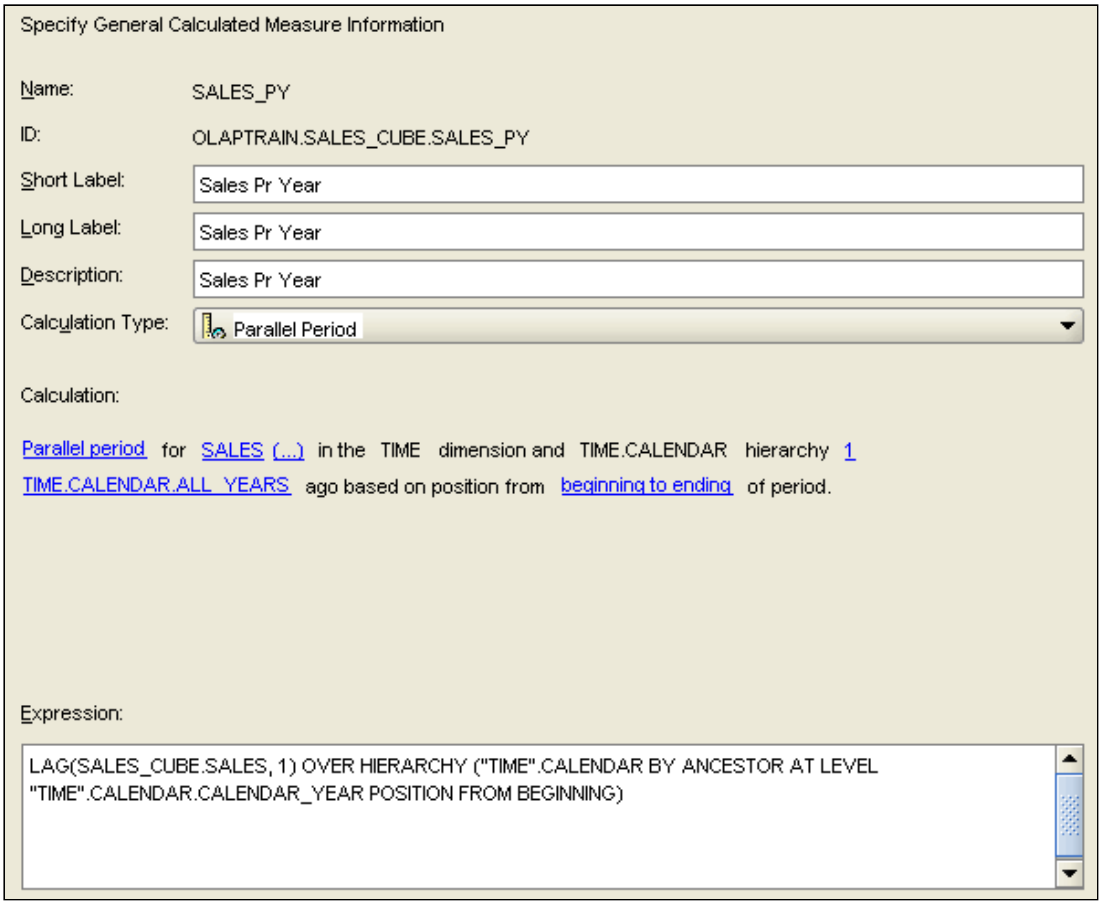


c. Click **Create**.

Result: The calculation appears in the navigator.

12. Select the new calculation in the navigator.

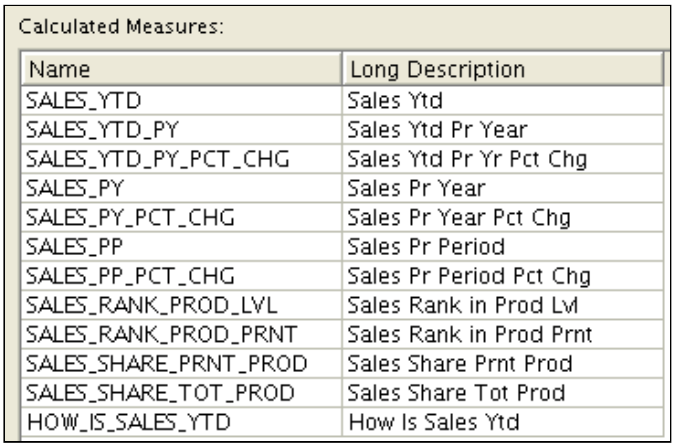
Result: The Sales Prior Year calculation definition appears in the right-hand pane, as shown below:



13. Repeat steps 10 and 11 to create eight more calculations using the following XML files:

- SALES_PY_PCT_CHG.XML
- SALES_PP.XML
- SALES_PP_PCT_CHG.XML
- SALES_ RANK_PROD_LVL.XML
- SALES_ RANK_PROD_PRNT.XML
- SALES_SHARE_PRNT_PROD.XML
- SALES_SHARE_TOT_PROD.XML
- HOW_IS_SALES_YTD.XML

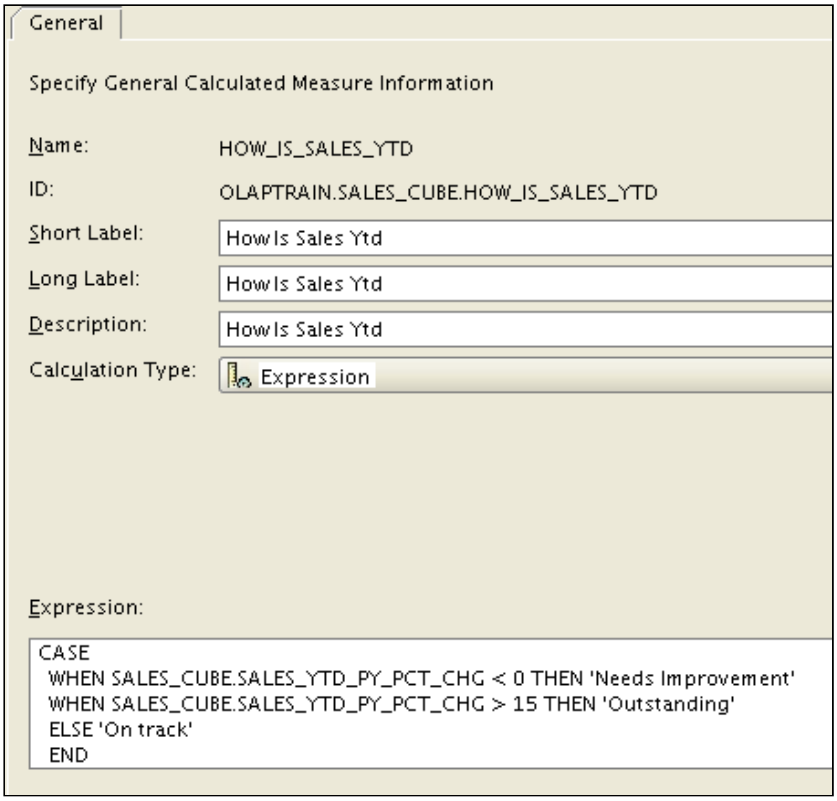
14. Select the **Calculated Measures** node in the navigator to display the list of calculated measures in the cube.



Name	Long Description
SALES_YTD	Sales Ytd
SALES_YTD_PY	Sales Ytd Pr Year
SALES_YTD_PY_PCT_CHG	Sales Ytd Pr Yr Pct Chg
SALES_PY	Sales Pr Year
SALES_PY_PCT_CHG	Sales Pr Year Pct Chg
SALES_PP	Sales Pr Period
SALES_PP_PCT_CHG	Sales Pr Period Pct Chg
SALES_RANK_PROD_LVL	Sales Rank in Prod Lvl
SALES_RANK_PROD_PRNT	Sales Rank in Prod Prnt
SALES_SHARE_PRNT_PROD	Sales Share Prnt Prod
SALES_SHARE_TOT_PROD	Sales Share Tot Prod
HOW_IS_SALES_YTD	How Is Sales Ytd

All of your measures have been created.

15. Click the **How Is Sales Ytd** calculated measure in the navigator to view the definition of the calculation, as shown here:



Notes:

- The Calculation Type is **Expression**. This special calculation type allows the OLAP data model developer to create a custom calculation by entering the appropriate OLAP calculation syntax in the Expression box.
- In this example, the syntax includes a “CASE” statement that evaluates the series of conditions and returns the first expression that matches the condition. The CASE statement is designed to return a text value describing the performance of the current Sales YTD compared to last year.
- You will used this, and other calculated measures later in this tutorial.
- You can learn more about creating OLAP calculations by attending the Oracle University **Oracle Database 11g: OLAP Essentials** inClass course. For a description of this course, see More Information at the end of this tutorial.

Back to Topic List

Mapping Cubes to Relational Sources

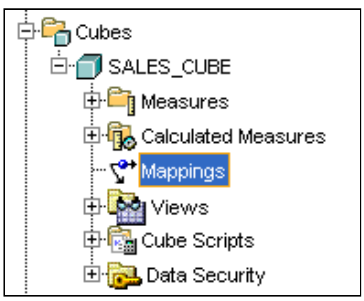
After creating an OLAP cube, you map it to relational data sources in Oracle Database. When mapping the cube, drag the appropriate source data column to the associated field for the OLAP cube element.

You map the following fields:

- The stored measures that are defined within the cube.
- The lowest level of detail for each dimension hierarchy.
- The Join Condition field. This field associates the foreign key (fk) column from the fact table to the primary key (pk) column from the dimension table.

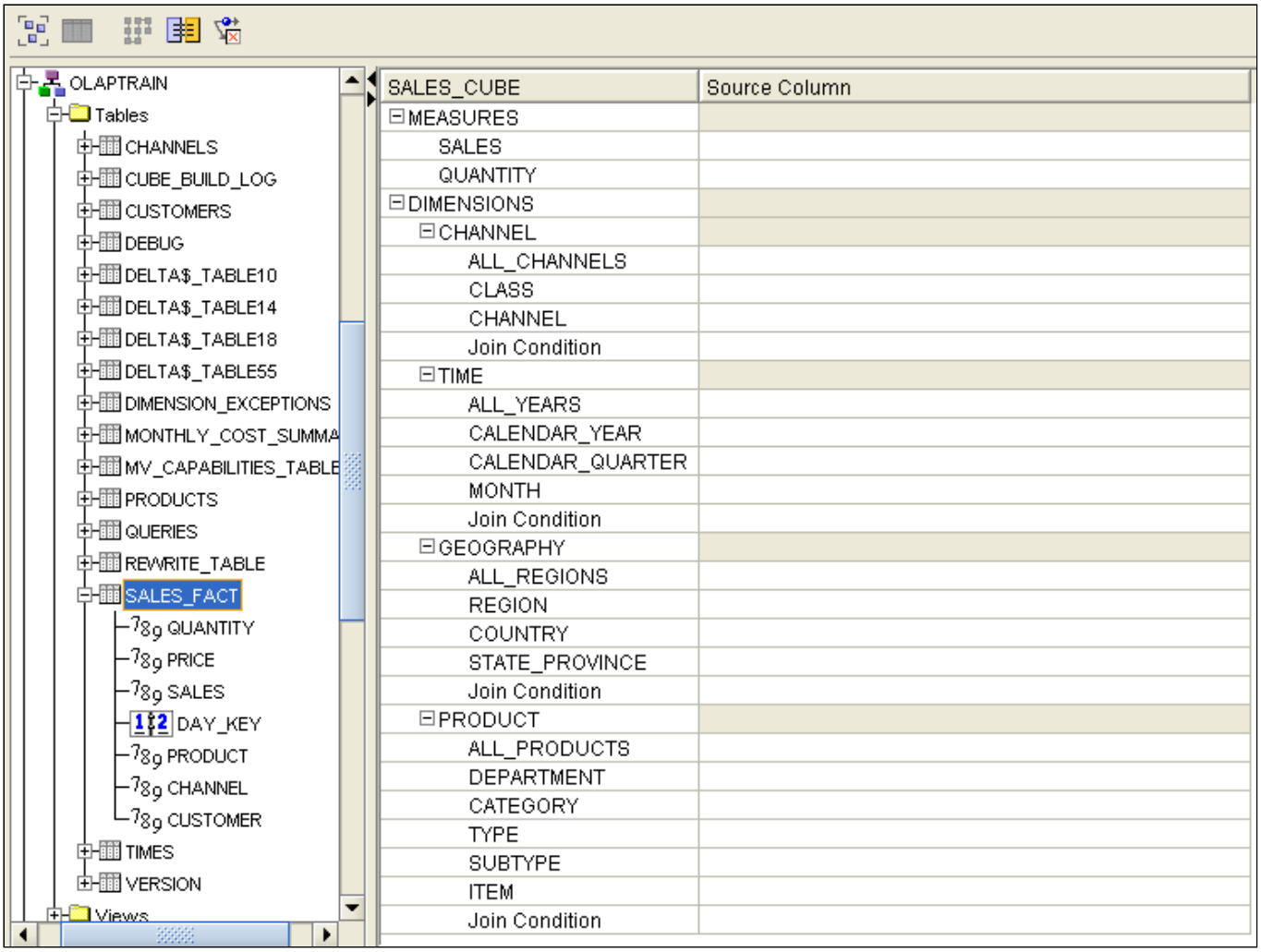
Afterward, you can load data into your analytic workspace using the Maintain Analytic Workspace wizard.

1. Click the **Mappings** node under SALES_CUBE



Ensure that the Table Mapping View is enabled.

2. In the source schemas pane, drill on **OLAPTRAIN** > **Tables**.



3. Locate and use the following tables:

- SALES_FACT
- CHANNELS
- CUSTOMERS
- PRODUCTS
- TIMES

Drag the appropriate columns from each source table to the associated SALES_CUBE Source Column fields, as shown in the image below.

Notes: When mapping the Join Condition for each dimension:

- First, drag the foreign key column from the fact table to the Source Column field.
- Then, drag the primary key column from the dimension table to the Source Column field.
- The equal sign (“=”) is automatically inserted after you drag the second column into the Source Column field.

When the mapping is complete, your source column results should look like this:

SALES_CUBE	Source Column
⊟MEASURES	
SALES	OLAPTRAIN.SALES_FACT.SALES
QUANTITY	OLAPTRAIN.SALES_FACT.QUANTITY
⊟DIMENSIONS	
⊟CHANNEL	
ALL_CHANNELS	
CLASS	
CHANNEL	OLAPTRAIN.CHANNELS.CHANNEL_KEY
Join Condition	OLAPTRAIN.SALES_FACT.CHANNEL=OLAPTRAIN.CHANNELS.CHANNEL_KEY
⊟TIME	
ALL_YEARS	
CALENDAR_YEAR	
CALENDAR_QUARTER	
MONTH	OLAPTRAIN.TIMES.MONTH_ID
Join Condition	OLAPTRAIN.SALES_FACT.DAY_KEY=OLAPTRAIN.TIMES.DAY_KEY
⊟GEOGRAPHY	
ALL_REGIONS	
REGION	
COUNTRY	
STATE_PROVINCE	OLAPTRAIN.CUSTOMERS.STATE_PROVINCE_KEY
Join Condition	OLAPTRAIN.SALES_FACT.CUSTOMER=OLAPTRAIN.CUSTOMERS.CUSTOMER_KEY
⊟PRODUCT	
ALL_PRODUCTS	
DEPARTMENT	
CATEGORY	
TYPE	
SUBTYPE	
ITEM	OLAPTRAIN.PRODUCTS.ITEM_KEY
Join Condition	OLAPTRAIN.SALES_FACT.PRODUCT=OLAPTRAIN.PRODUCTS.ITEM_KEY

4. Click **Apply**.

Back to Topic List

Enabling Query Rewrite to Cube MVs (optional)

In an extension of the Materialized View capabilities for Oracle Database, OLAP cubes can be represented as a cube-organized materialized views (Cube MVs). The query optimizer automatically recognizes when an existing Cube MV can and should be used to satisfy a SQL summary request. A Cube MV represents a significant summary space, and benefits include both ease of manageability and improved query performance.

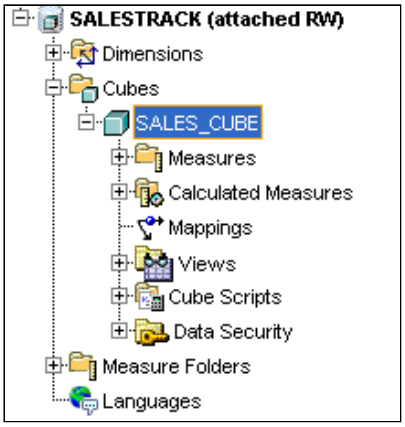
If your OLAP system requirements do not include a need for summary management of exiting SQL-based BI applications, then you can skip this optional task.

Notes:

- If you chose to enable query rewrite, supporting cube MV objects are automatically created and managed by the Oracle Database.
- Before you can enable materialized views for the cube, you must first map the cube.

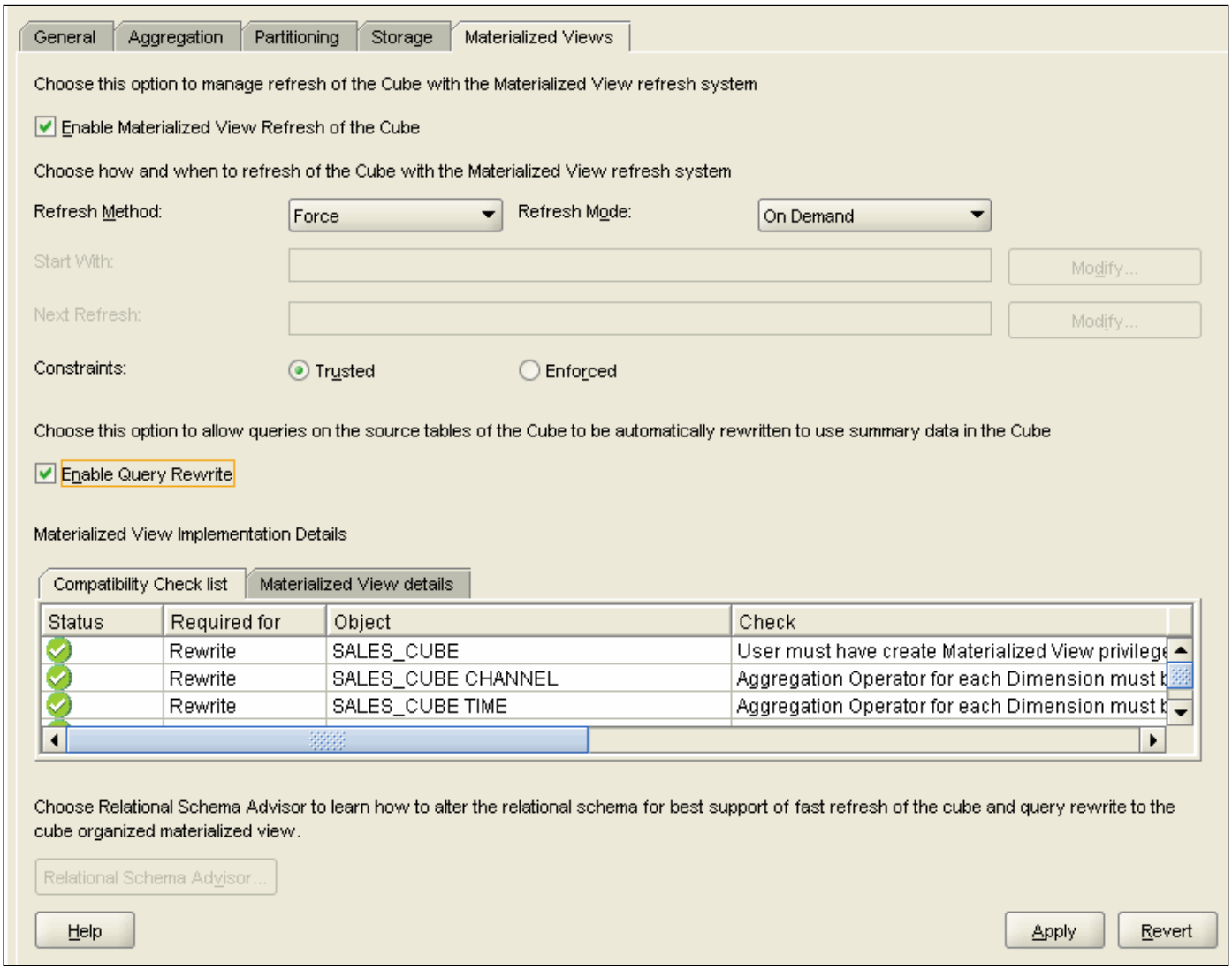
To enable query rewrite and MV refresh for your OLAP cube, peform the following steps.

1. In the navigator, click **SALES_CUBE**.



2. In the right pane, click the Materialized Views tab and select the following options:

- **Enable Materialized View Refresh of the Cube**
- **Enable Query Rewrite**



Notes:

MV Refresh

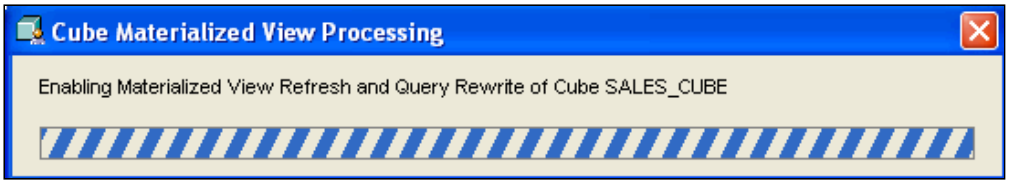
- If you select Enable MV Refresh, you also specify the refresh method and mode for the cube. Cube MV refresh methods include, Complete, Force, and Fast.
- The default Refresh Mode is On Demand.

Query Rewrite

- If you select Enable Query Rewrite, supporting cube MV objects are automatically created by the database when you click Apply.
- When a Cube is enabled for query rewrite, the associated Dimensions are automatically enabled for MV refresh as well.

3. Accept the default settings for all other options, and then click **Apply**.

Result: the following information box appears:



When the information box closes, cube MVs are enabled and ready for use by the Materialized View subsystem.

Note: For more information on enabling and troubleshooting Query Rewrite to Cube MVs, see this white paper.

Back to Topic List

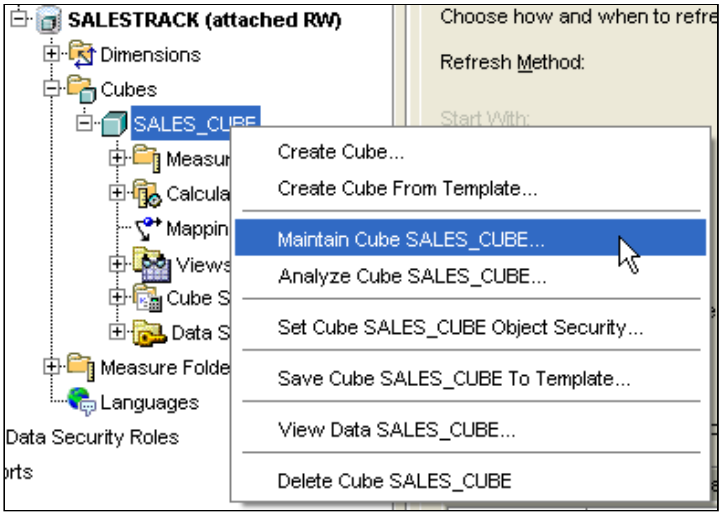
Loading and Viewing Cube Data

The Maintenance Wizard loads and aggregates the data in a single step. You can load all mapped objects in the analytic workspace, or individual dimensions and measures. You can also choose to run the job immediately, enter it in the Oracle Job Queue, or save it as a SQL script.

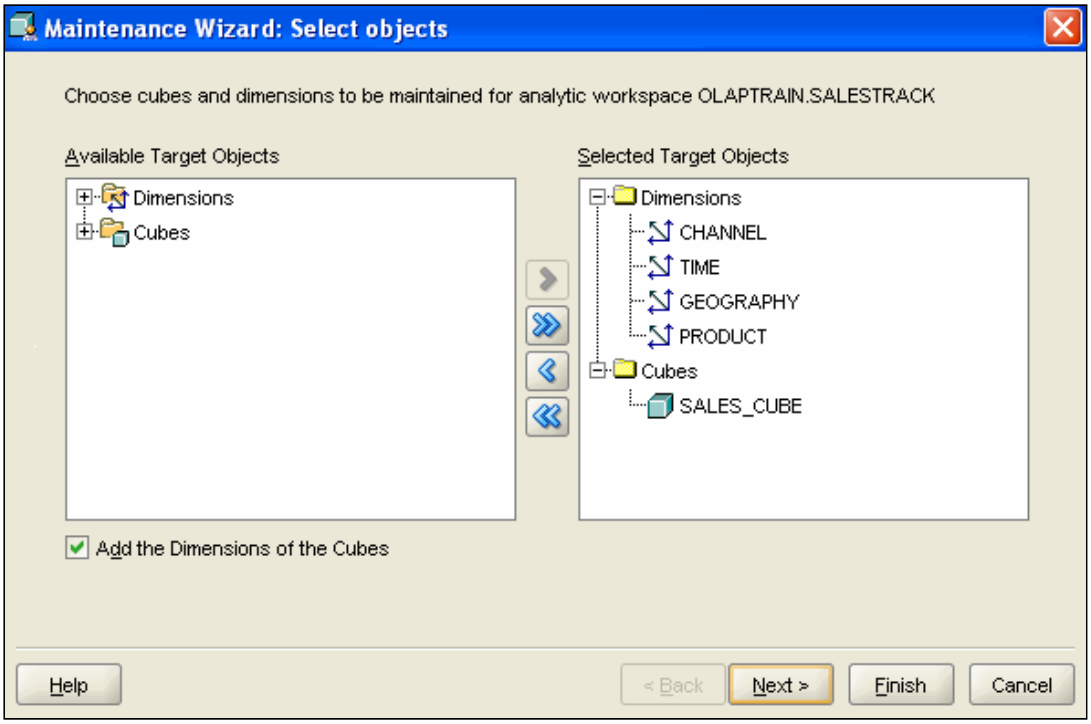
By default, when you load data to a cube, the dimensions of that cube are also processed. If you have already loaded dimension data, you can specify only to load measure data.

In the following steps, you load all data for the cube and run the job immediately. Then you view the data in AWM.

1. In the navigator, right-click on **SALES_CUBE** and then select **Maintain Cube SALES_CUBE**.



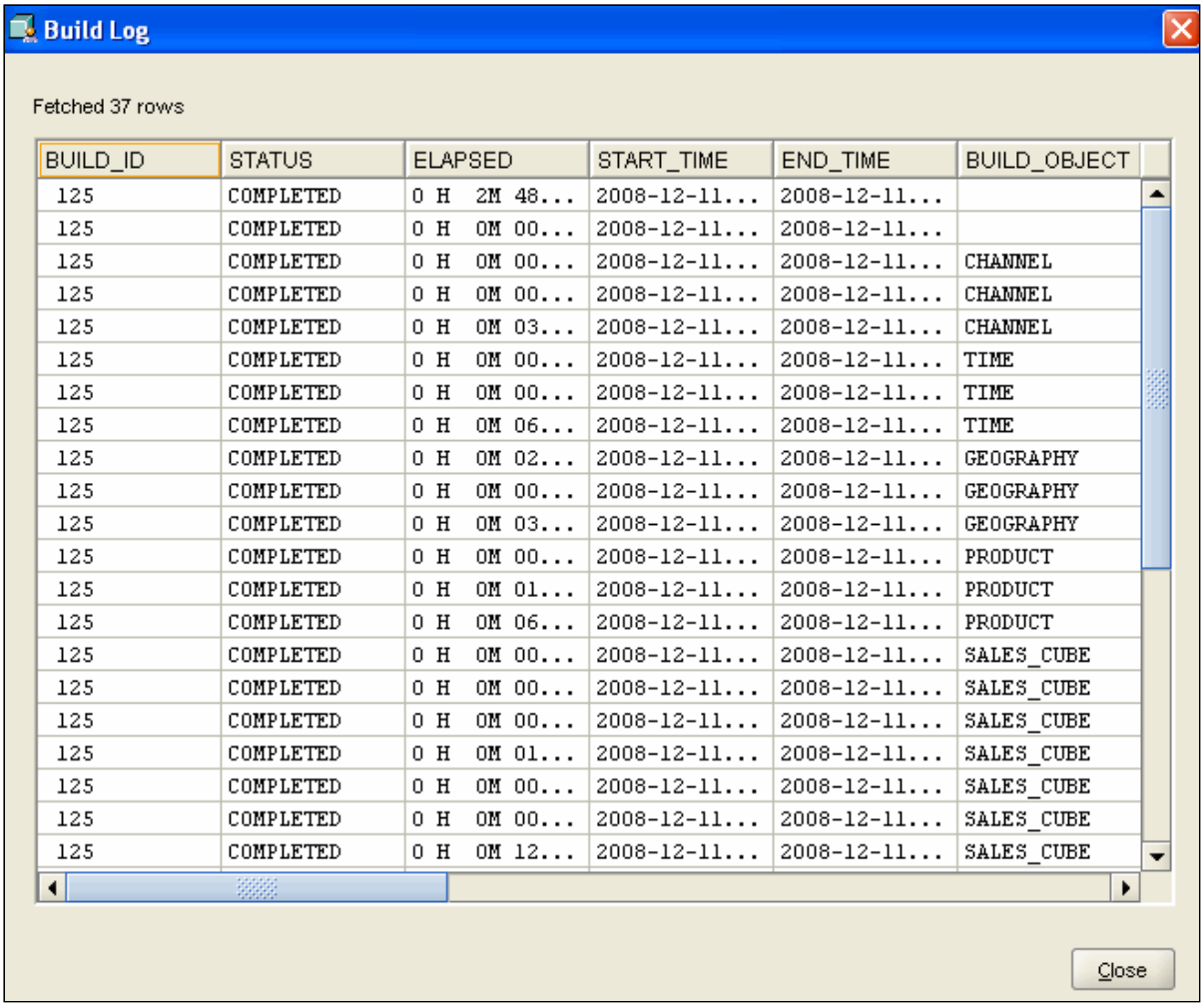
2. In the Maintenance Wizard, click **Finish** to begin the load process.



Note: The default settings for an immediate build of all cube objects is applied.

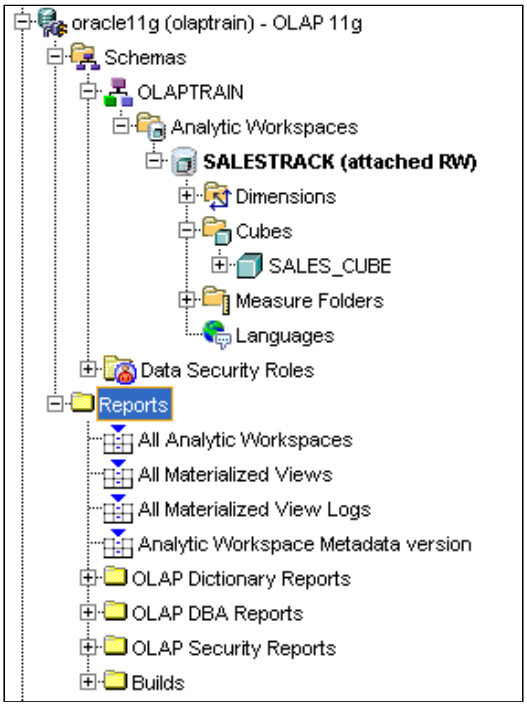
Result: The following information box appears:

3. When the build completes, the Build Log window appears.



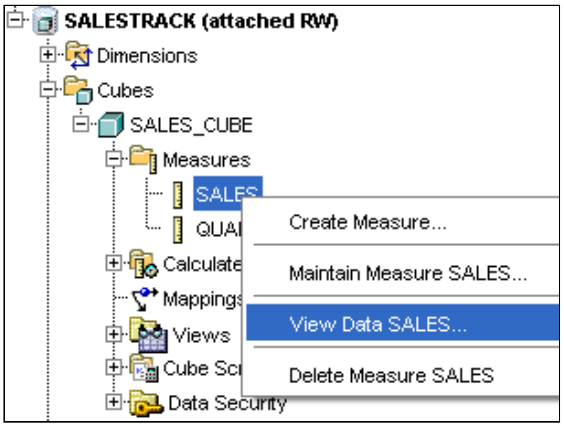
If you scroll to the right, and then down in the Build Log, you can see how each of the cube partitions were processed. Click **Close** after you finish examining the Build Log.

Note: There are several logs that you can view from the navigator by clicking on the **Reports** node.



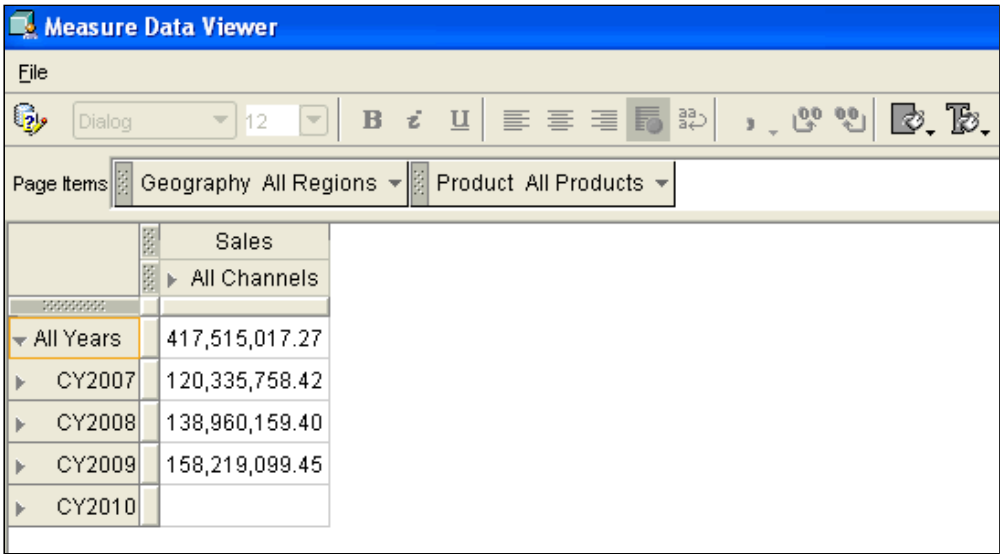
5. You can view OLAP data from within AWM.

In the Navigator, right-click on the **SALES** measure and select **View Data Sales** from the menu.



Result: Sales data is displayed for a default set of dimension members in a crosstab and a graph.

6. In the Data Viewer, drill on **All Years**. The following data appears in the crosstab:



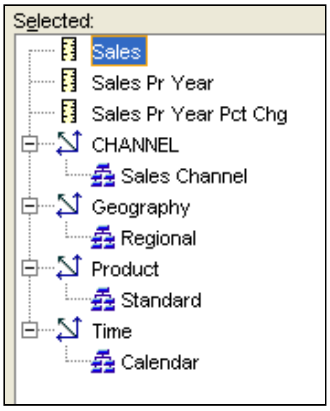
Note: CY2010 is blank because data for the 2010 calendar year is not stored in the relational schema. CY2010 time periods are built into the OLAP data model so that new data can be automatically updated with subequent data loads.

7. Click the Query Builder tool , under the File menu.

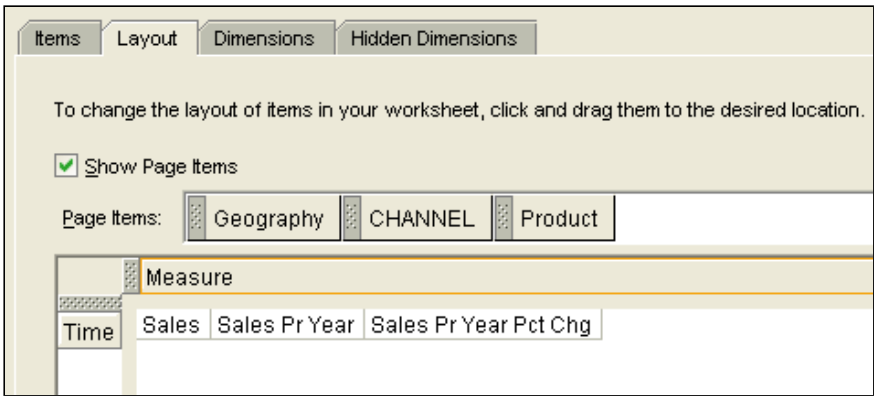
8. In the Items tab, perform the following:

   a) Select **Sales Pr Year** and **Sales Pr Year Pct Chg**.

   b) Click the **Add Selected Items** tool (>) to move those two measures to the Selected list, like this:

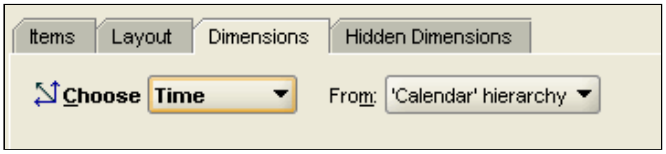


9. Click the **Layout** tab. In the Layout tab, drag the appropriate dimension tiles to the correct axis so that the layout looks like this:



10. Click the **Dimensions** tab. In the Dimensions tab, perform the following:

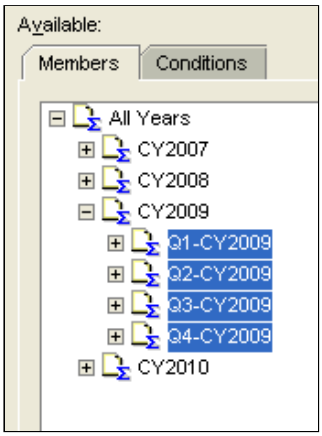
a) Select the **Time** dimension from the Choose drop-down list.



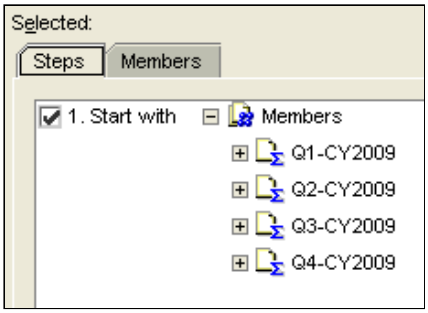
b) Click the **Remove All Items** tool (<<) to clear the Selected list.

c) In the Members tab of the Available list, drill on **All Years** > **CY2009**.

d) Select all four quarters in CY2009, like this:

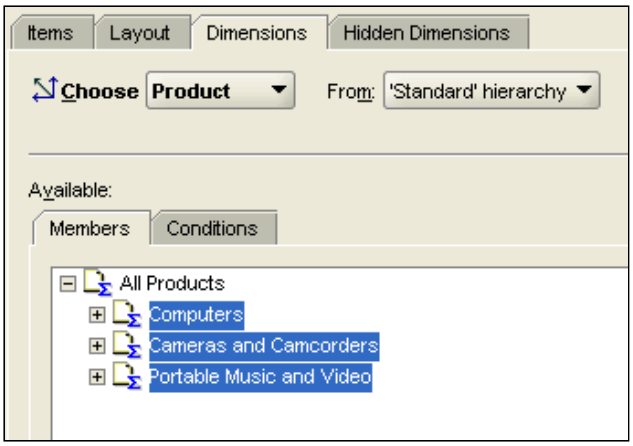


e) Click the **Add Items tool** (>) to move the 2009 quarters to the Selected list, like this:



f) Select **Product** from the Choose drop-down list.

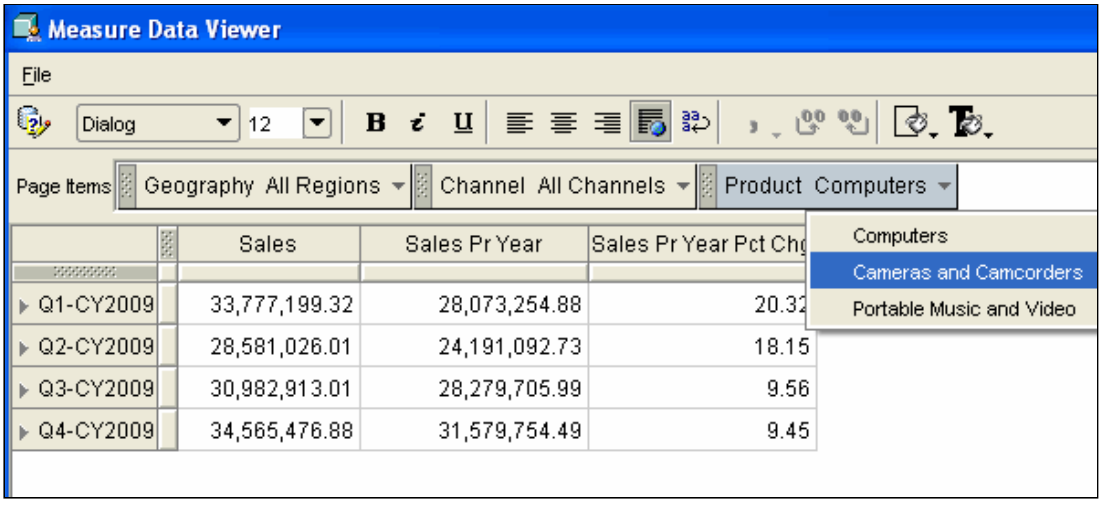
g) Click the **Remove All Items** tool and then drill on **All Products** in the Available list. Finally, select the three Product department members, like this:



h) Click the **Add Items** tool to move the department members to the Selected list.

i) Click **OK** to view the data.

11. Select any of the members from the Product dimension header. The calculations are instantaneously updated.

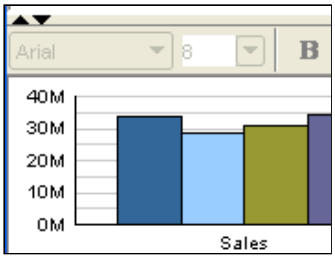


Drill on any of the 2009 Quarter values to view the data at the month level.

When you are done experimenting with the report, collapse all drills on the Time dimension, so that only the four quarters of 2009 are displayed.

12. Next, you modify the Data Viewer for a Year-to-Date analysis report.

First, hide the Graph by clicking on the **down arrow** of the Hide/Show tool.



13. Then, click the **Query Builder** tool, and in the Items tab, perform the following:

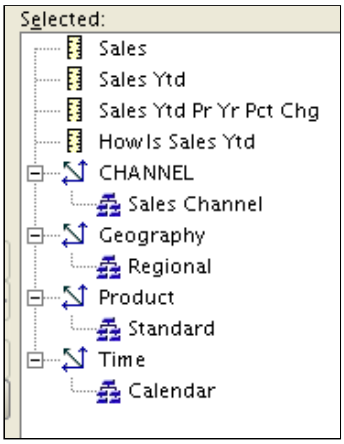
a) In the Selected list, select **Sales Pr Year** and **Sales Pr Year Pct Chg**.

b) Click the **Remove Selected Items** tool (<).

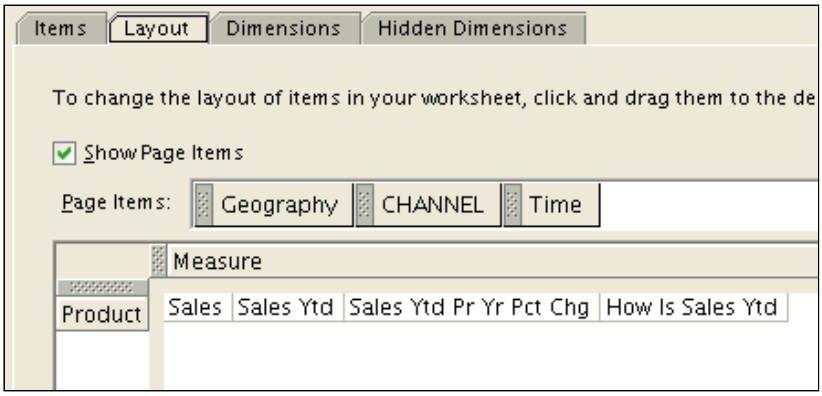
c) In the Available list, select **Sales Ytd**, **Sales Ytd Pr Year Pct Chg** , and **How Is Sales Ytd**.

d),Click the **Add Selected Items** tool (>).

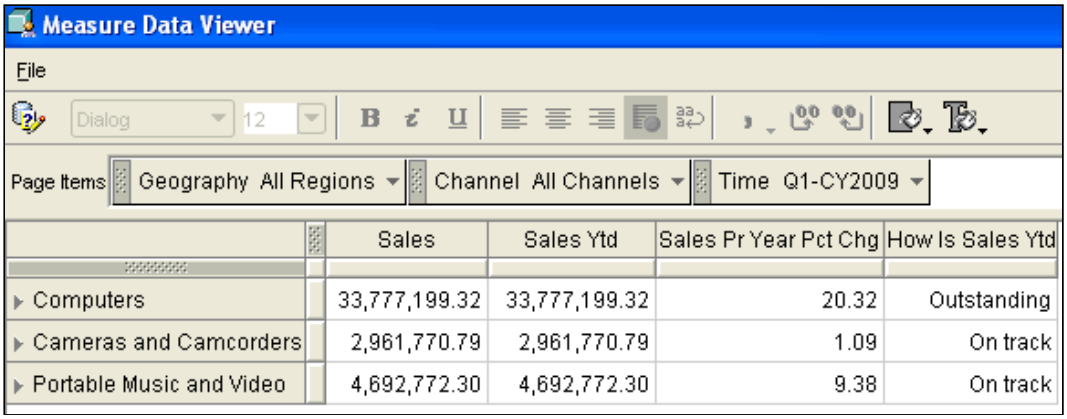
Result: The Year to Date measures are added to the Selected list.



14. In the Layout tab, swap the Product and Time dimensions, so that Product is in the Row axis, and Time is in the Page Items axis, like this:



15. Click **OK** to view the data.

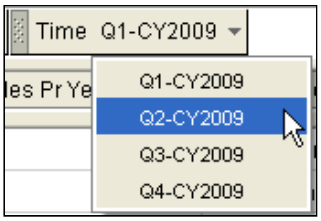


	Sales	Sales Ytd	Sales Pr Year Pct Chg	How Is Sales Ytd
Computers	33,777,199.32	33,777,199.32	20.32	Outstanding
Cameras and Camcorders	2,961,770.79	2,961,770.79	1.09	On track
Portable Music and Video	4,692,772.30	4,692,772.30	9.38	On track

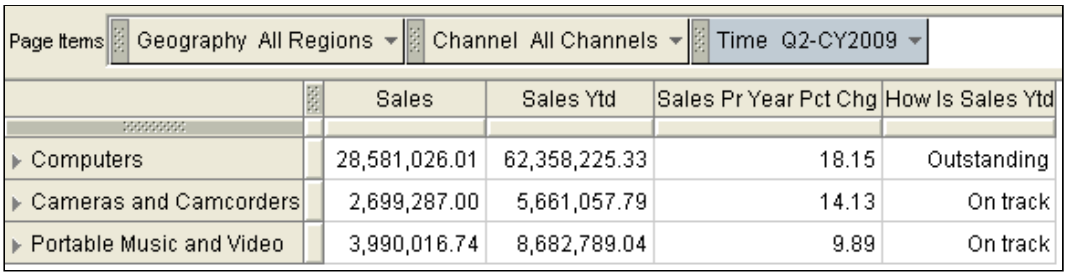
The calculated measures show:

- The Sales Year-To-Date data (in this case, the same as Sales, since Q1-CY2009 is selected).
- The percent change when comparing Sales YTD for the current period (Q1-2009) to Sales YTD for the prior year (Q1-2008).
- Text values that describe Sales YTD percent change performance as "Needs Improvement", "On Track", or "Outstanding". Recall that a conditional CASE statement is used in this calculated measure to produce the result.

16. From the Time dimension header, select **Q2-CY2009**, as shown here:

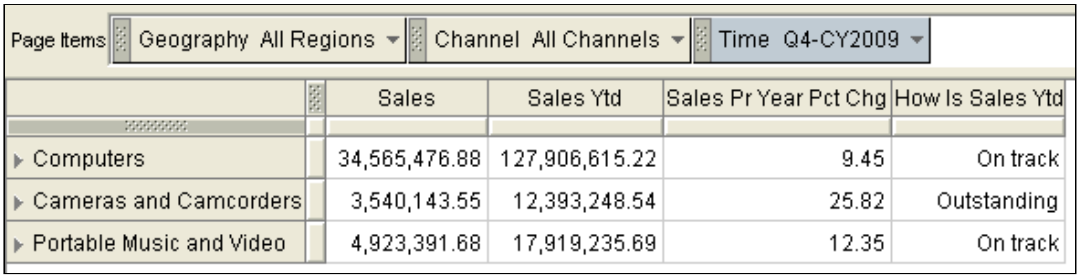


Result: The stored and calculated data updates with the correct values.



	Sales	Sales Ytd	Sales Pr Year Pct Chg	How Is Sales Ytd
Computers	28,581,026.01	62,358,225.33	18.15	Outstanding
Cameras and Camcorders	2,699,287.00	5,661,057.79	14.13	On track
Portable Music and Video	3,990,016.74	8,682,789.04	9.89	On track

17. From the Time dimension header, select **Q4-CY2009**.



	Sales	Sales Ytd	Sales Pr Year Pct Chg	How Is Sales Ytd
Computers	34,565,476.88	127,906,615.22	9.45	On track
Cameras and Camcorders	3,540,143.55	12,393,248.54	25.82	Outstanding
Portable Music and Video	4,923,391.68	17,919,235.69	12.35	On track

The How Is Sales Ytd measure correctly reflects the newly queried data.

18. Drill on **Computers**.

Page Items: Geography All Regions | Channel All Channels | Time Q4-CY2009

	Sales	Sales Ytd	Sales Pr Year Pct Chg	How Is Sales Ytd
Computers	34,565,476.88	127,906,615.22	9.45	On track
Total Personal Computers	28,383,361.43	104,944,431.56	10.20	On track
PDAs	28,849.00	101,464.67	3.59	On track
All Computer Furniture	14,202.60	66,635.00	-28.98	Needs Improvement
Computer Printers and Supplies	5,587,835.85	20,693,698.99	6.60	On track
Total Server Computers	551,228.00	2,100,385.00	3.48	On track
Cameras and Camcorders	3,540,143.55	12,393,248.54	25.82	Outstanding
Portable Music and Video	4,923,391.68	17,919,235.69	12.35	On track

Again, the YTD performance measures automatically update to reflect the current selections.

19. Finally, you modify the Data Viewer for a product ranking and share report.

- First, collapse the Product dimension drills to display only the three product departments.
- Second, select **Q1-CY2009** from the Time dimension header.

Then, click the **Query Builder** tool.

20. In the Items tab, perform the following:

a) In the Selected list, select **Sales Ytd**, **Sales Ytd Pr Year Pct Chg** , and **How Is Sales Ytd**.

b) Click the **Remove Selected Items** tool (<).

c) In the Available list, select **Sales Rank In Prod Prnt** and **Sales Share Prnt Prod**.

d),Click the **Add Selected Items** tool (>).

Result: The rank and share measures are added to the Selected list.

21. Click **OK** to view the data.

Measure Data Viewer

File

Dialog 12 B I U

Page Items: Geography All Regions | Channel All Channels | Time Q1-CY2009

	Sales	Sales Rank in Prod Prnt	Sales Share Prnt Prod
Computers	33,777,199.32	1.00	81.52
Cameras and Camcorders	2,961,770.79	3.00	7.15
Portable Music and Video	4,692,772.30	2.00	11.33

The calculated measures show:

- The rank of each Product dimension member within its hierarchy parent.
- The share of each product member as a percentage of sales returned by that product member to its parent in the hierarchy.

22. Drill on **Computers**.

Page Items: Geography All Regions | Channel All Channels | Time Q1-CY2009

	Sales	Sales Rank in Prod Prnt	Sales Share Prnt Prod
Computers	33,777,199.32	1.00	81.52
Total Personal Computers	27,723,678.35	1.00	82.08
PDAs	26,675.15	4.00	0.08
All Computer Furniture	21,313.40	5.00	0.06
Computer Printers and Supplies	5,434,375.02	2.00	16.09
Total Server Computers	571,157.40	3.00	1.69
Cameras and Camcorders	2,961,770.79	3.00	7.15
Portable Music and Video	4,692,772.30	2.00	11.33

The rank and share measures show the relative ranking and share contribution for each of the Product division members in the Computer department.

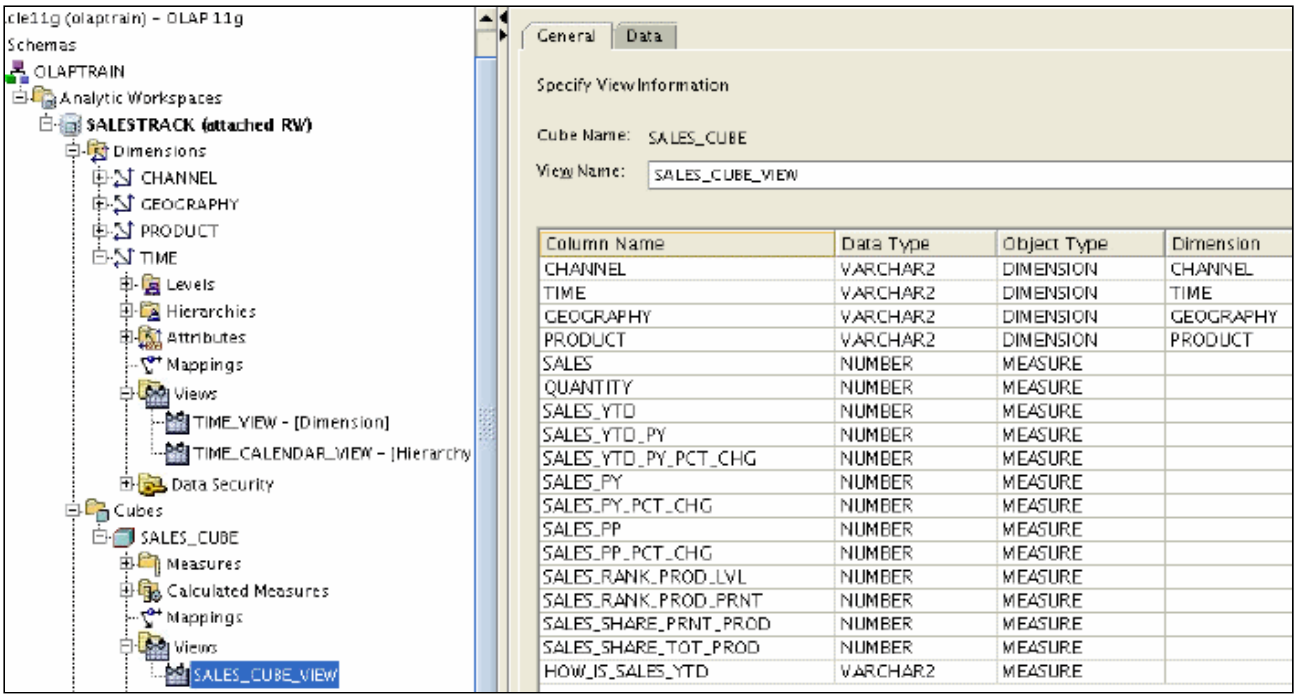
23. Select any Time member from the Page Items axis, and the calculations update instantaneously.

Feel free to modify the report by drilling or selected other dimension members.

When you are done, close the Measure Data Viewer.

24. You can also perform ad-hoc, multidimensional analysis against OLAP data with any SQL-based tool. Oracle OLAP data is made directly accessible to SQL by a set of relational views that are automatically created and maintained by Oracle OLAP. You query OLAP data by executing simple SQL statements against these associated Cube Views.

Click **SALES_CUBE_VIEW** -- the view created for SALES_CUBE -- to display information and data for the view.



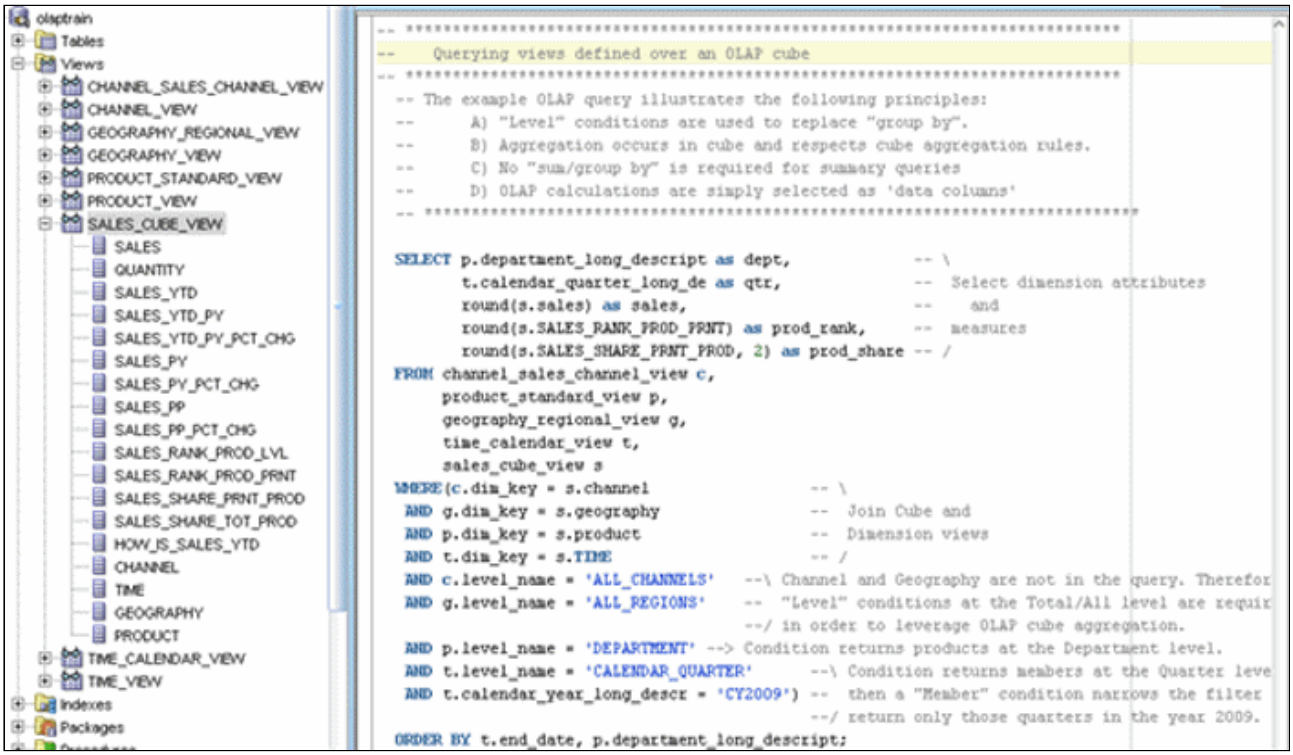
Oracle OLAP creates and maintains views for each cube and dimension. These views represent an OLAP Cube as a star schema with the following characteristics:

- A cube view plays the role of a fact table.
- Dimension views and hierarchy views play the role of dimension tables (a dimension and a hierarchy view are created and maintained for each dimension in the OLAP data model).

Although SQL access to OLAP cubes is covered in another tutorial, an example of a SQL query against the cube that you just created is provided next.

25. In the example shown here, SQL Developer is used to write a SQL query.

The SQL statement, explained by in-line notes, queries the same data that is selected by AWM in step 21.



Note: SQL Developer is shipped free with Oracle Database.

26. The query above is executed, and the resulting output shows the same data that was returned in step 21.

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

	DEPT	QTR	SALES	PROD_RANK	PROD_SHARE
1	Cameras and Camcorders	Q1-CY2009	2961771	3	7.15
2	Computers	Q1-CY2009	33777199	1	81.52
3	Portable Music and Video	Q1-CY2009	4692772	2	11.33
4	Cameras and Camcorders	Q2-CY2009	2699287	3	7.65
5	Computers	Q2-CY2009	28581026	1	81.03
6	Portable Music and Video	Q2-CY2009	3990017	2	11.31
7	Cameras and Camcorders	Q3-CY2009	3192047	3	8.29
8	Computers	Q3-CY2009	30982913	1	80.5
9	Portable Music and Video	Q3-CY2009	4313055	2	11.21
10	Cameras and Camcorders	Q4-CY2009	3540144	3	8.23
11	Computers	Q4-CY2009	34565477	1	80.33
12	Portable Music and Video	Q4-CY2009	4923392	2	11.44

For information how to query OLAP data using SQL, see the Querying OLAP Cubes OBE lesson.

Back to Topic List

More Information

For more information on OLAP-related education, use the following links:

- Oracle Database 11g: OLAP Essentials (Oracle University inClass)
- Oracle OLAP (OTN site)

Back to Topic List

Place the cursor over this icon to hide all screenshots.