
Appendix A

Practices and Solutions

Practices Overview

The goal of the course practices is to progressively build an OLAP data model to support the following requirements:

- Summary management for existing SQL table queries by designing OLAP cubes and enabling query rewrite against the cubes
- Enhanced analytic content, including a wide range of calculated measures that enable easy end-user access to rich analytic information
- Easy access to OLAP data by using simple SQL:
 - Creating and executing OLAP queries in SQL Developer
 - Using OLAP queries in Oracle Application Express (APEX)
 - Using OLAP queries in Oracle BI Enterprise Edition (BI EE)
- Cube security
- Efficient cube design for performance and scalability

Practices for Lesson 3

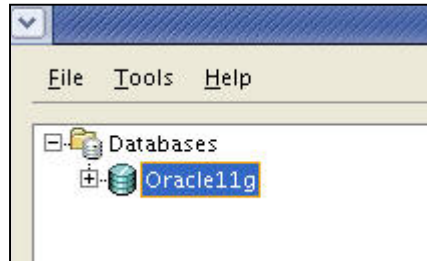
In the practices for this lesson, you start Analytic Workspace Manager (AWM), create a new AW, and build an OLAP cube. You will perform the following tasks:

- Create a database connection for your OLAP environment and log in.
- Create an Analytic Workspace.
- Create Dimensions for your OLAP data model, including:
 - Levels
 - Hierarchies
 - Attributes
 - Dimension mappings
- Load dimension data.
- View dimension data using:
 - OLAP Data Viewer
 - Dimension and hierarchy views
- Create a cube for Sales data.
 - Add the Sales measure.
 - Map the cube to relational sources.
- Load data for the cube.
- View the cube build logs:
 - In AWM
 - At the SQL level
- View cube data using:
 - OLAP Data Viewer
 - Cube views

Practice 3-1a: Create the Analytic Workspace

In this practice section, you create an analytic workspace using an existing database connection to the classroom schema. To perform this task:

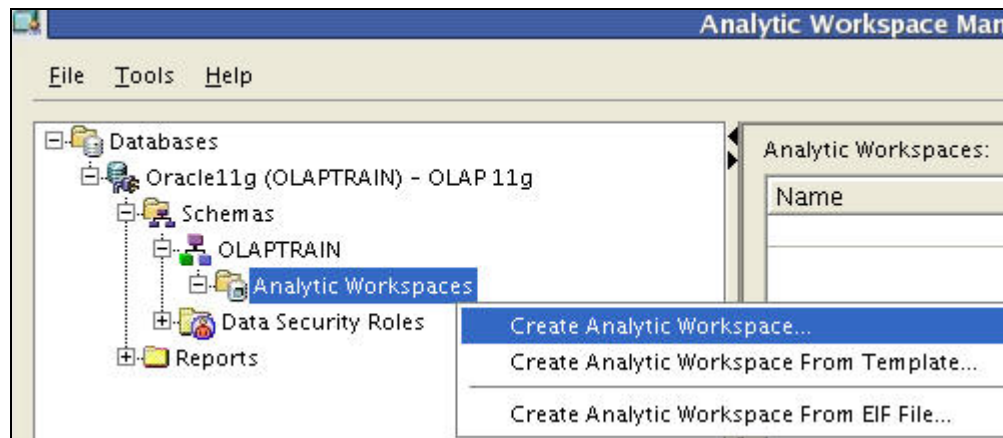
1. Launch AWM from the desktop.
2. Select the **Databases** node.
3. **Result:** The database connection node named “Oracle11g” appears.



4. Click the plus sign (+) next to the Oracle11g database node.
5. In the Connect to database dialog box, enter and perform the following:
 - a. Username: **olaptrain**
 - b. Password: **oracle**
 - c. Click **OK**.

Result: Two folders appear below the database connection – Schemas and Reports.

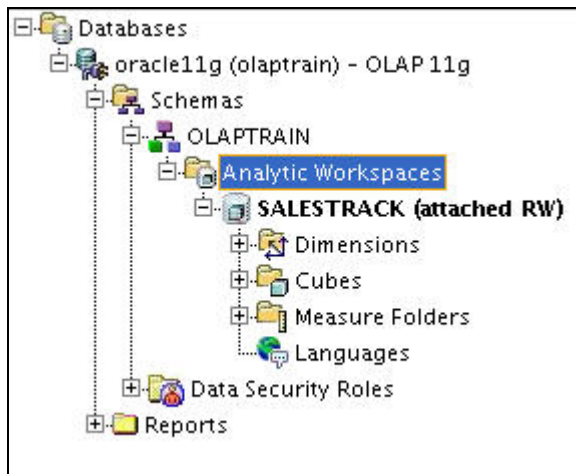
6. Select **Schemas > OLAPTRAIN**. Then, right-click **Analytic Workspaces** and select **Create Analytic Workspace** from the menu.



7. In the Create Analytic Workspace dialog box, enter or select the following:
 - a. Name: **SALESTRACK**
 - b. Tablespace: **<default>**
 - c. Click **Create**.

Result: An AW named SALESTRACK appears in the navigator.

8. Select the SALESTRACK node to display the folders that organize OLAP data objects.



Practice 3-1b: Create the Dimensions

In this practice section, you create the four dimensions of what will organize your first OLAP cube: Channel, Time, Product, and Geography. You also create the levels and a hierarchy for each dimension. Finally, you will map the dimensions to their relational sources, and then load the data for one of the dimensions.

1. Right-click the Dimensions node and select **Create Dimension** from the menu.
2. In the General tab of the Create Dimension window, enter or select the following:
 - a. Name: **CHANNEL**
 - b. Dimension Type: **User Dimension**

Note: Caps Lock is auto-enabled for object names. Label and Description fields are auto-filled.
3. In the Levels tab, create three levels for the Channel dimension:
 - ALL_CHANNELS
 - CLASS
 - CHANNEL

Note: Enter each level name (as listed previously) in the Name field. As before, Label and Description fields are auto-filled. A new level is automatically added when you place the cursor in the Name field.

When you are done, the Levels tab should look like this:

Name	Long Label	Short Label	Description
ALL_CHANNELS	All Channels	All Channels	All Channels
CLASS	Class	Class	Class
CHANNEL	Channel	Channel	Channel

4. In the Implementation Details tab of the Create Dimension dialog box, select **Use Keys from Data Source**.

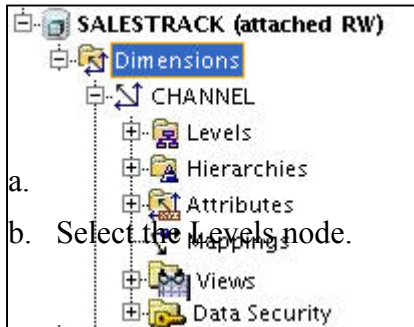
Dimension Member Loading:

☐ Generate Surrogate Keys in the Analytic Workspace


☒ Use Keys from Data Source

5. Click **Create**.

6. Select **Dimensions > CHANNEL**. The following should appear in the navigator:



7. Create a Hierarchy for the Channel dimension by performing the following:

- Right-click Hierarchies and select **Create Hierarchy**.
- In the Create Hierarchy dialog box, enter **SALES_CHANNEL** as the name.
- Click the Add All tool . The General tabbed page should look like this:

General

Specify General Hierarchy Information

Name: SALES_CHANNEL

Short Label: Sales Channel

Long Label: Sales Channel

Description: Sales Channel

☒ Set as Default Hierarchy

☒ Level Based Hierarchy ☐ Value Based Hierarchy

Define the levels for this hierarchy by moving levels from the Available list to the selected list. The order of levels in the Selected list reflect the order of the levels (highest to lowest) in the hierarchy.

Available Levels:

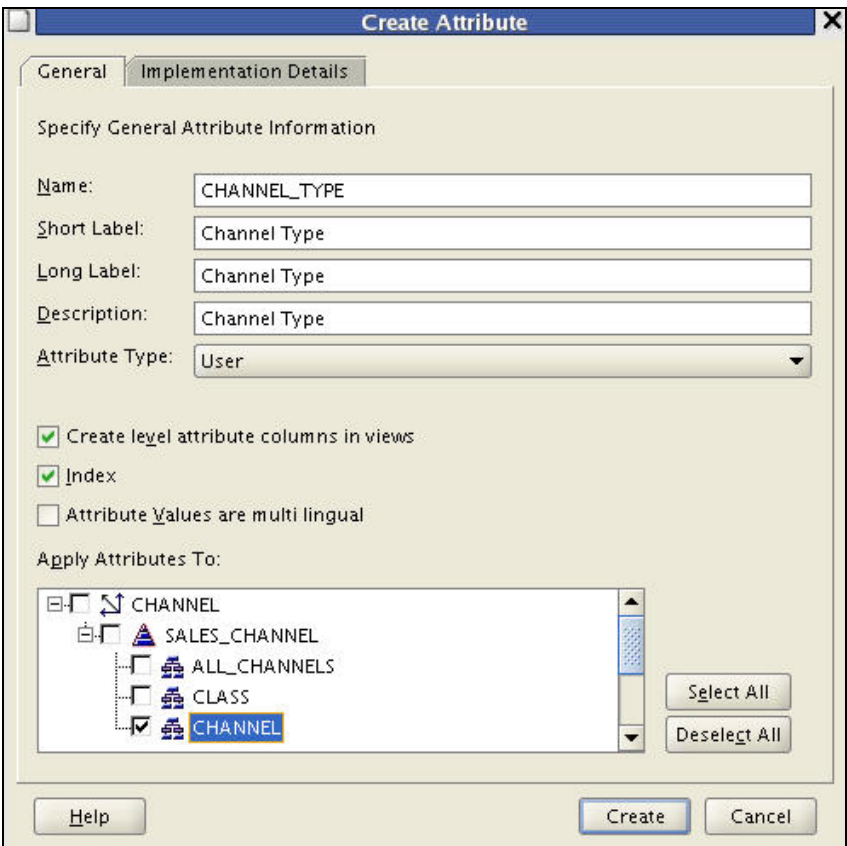
Selected Levels (Highest to Lowest):

- ALL_CHANNELS
- CLASS
- CHANNEL

- Click **Create**.
8. Create a User Attribute for Channel Type by doing the following:
- Right-click the **Attributes** node below the Channel dimension, and select **Create Attribute**.
 - In the Create Attribute dialog box, select or enter the following:
 - Name = **CHANNEL_TYPE**
 - Attribute Type = **User**
 - In the “Apply Attributes To” box:
 - Click the plus sign next to the Channel dimension.
 - Deselect the Channel dimension check box.
 - Select the **CHANNEL** level check box (the lowest level).

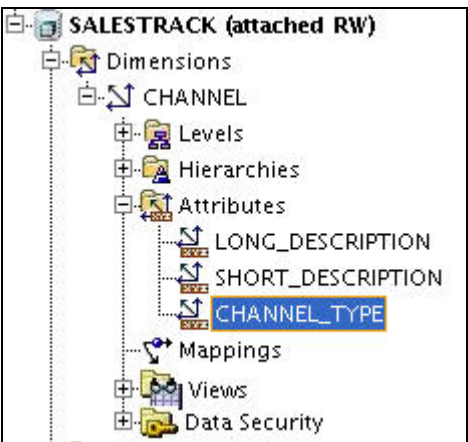
Note: The Channel Type attribute only applies to the lowest level in the Sales Channel hierarchy.

The dialog box should look like this:



c. Click **Create**.

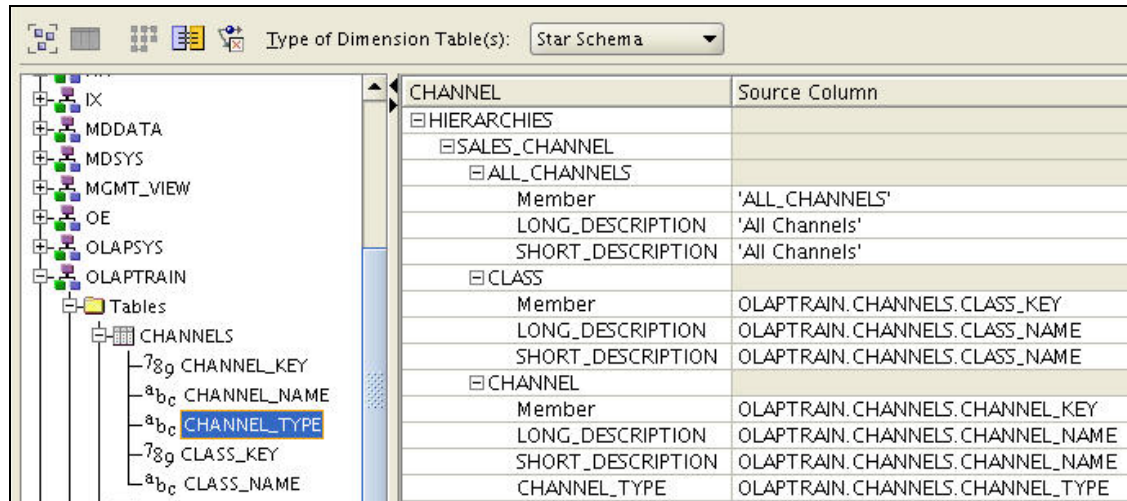
The Channel Type attribute is added to the dimension.



9. Map the Channel dimension to its relational sources by performing the following:
 - a. Click the **Mappings** node below the Channel dimension.

Result: The Schemas and Mapping panes appear to the right.

- b. In the Schemas pane, select **OLAPTRAIN > Tables** to display the source tables for the OLAPTRAIN schema. Then, select **CHANNELS** to display its columns.
- c. Ensure that **Star Schema** is selected for the Type of Dimension Table, and that **Table Mapping View** is selected.
- d. Drag the following source columns from the Schema pane to the Mapping pane for the CHANNEL and CLASS levels. In addition, enter the following constants for the ALL_CHANNELS level:
 - Member = **'ALL_CHANNELS'**
 - Description attributes = **'All Channels'**

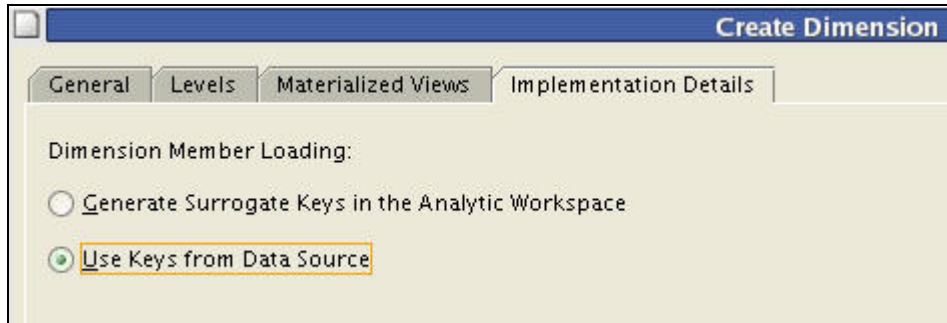


- e. Click **Apply**.

Result: The Channel dimension is ready for data loading.

10. To create another dimension, right-click the Dimensions node and select **Create Dimension** from the menu.
11. In the General tab of the Create Dimension dialog box, enter or select the following:
 - a. Name: **TIME**
 - b. Dimension Type: **Time Dimension**
12. Using the same techniques that you applied for the Channel dimension, add the following levels to the Time dimension:
 - ALL_YEARS
 - CALENDAR_YEAR
 - CALENDAR_QUARTER
 - MONTH

13. On the Implementation Details tabbed page, select **Use Keys from Data Source**.



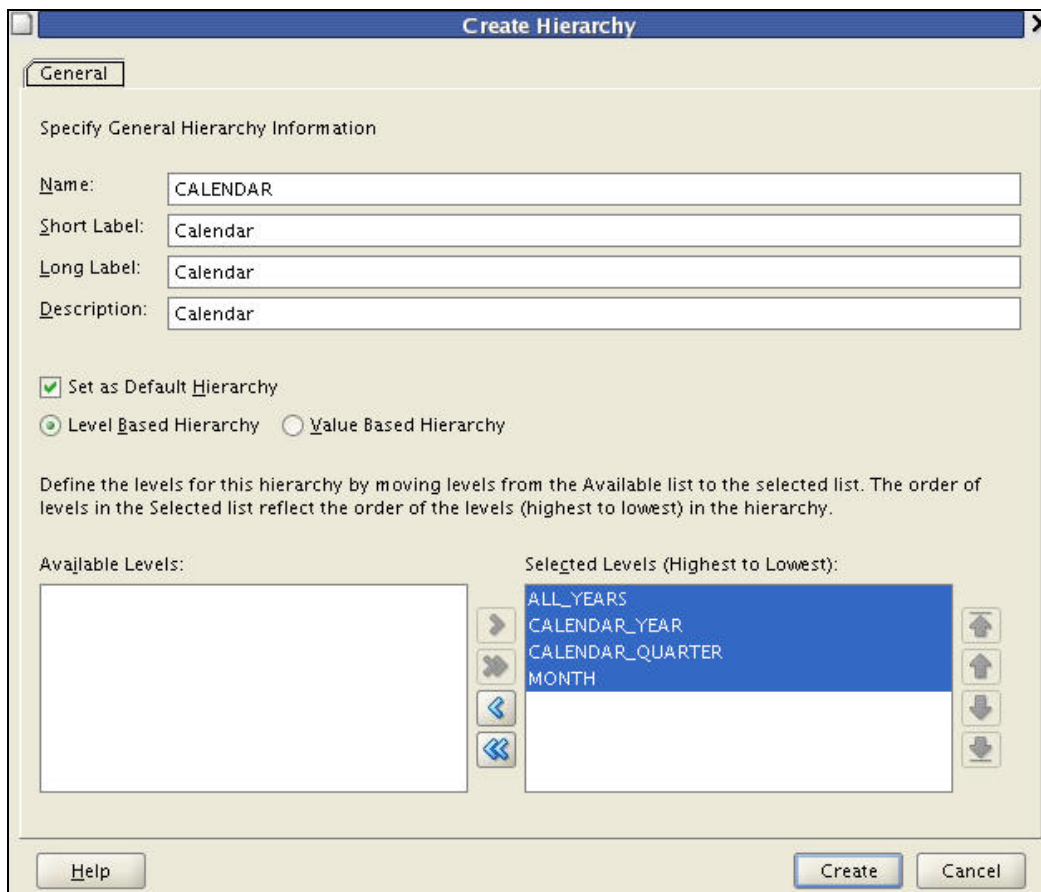
14. Click **Create**.

Result: TIME appears under the Dimensions node.

15. Select **TIME** to display its associated nodes.

16. Using the same techniques that you applied for the Channel dimension, create a Hierarchy named CALENDAR for the Time dimension.

The completed Create Hierarchy dialog box should look like this:

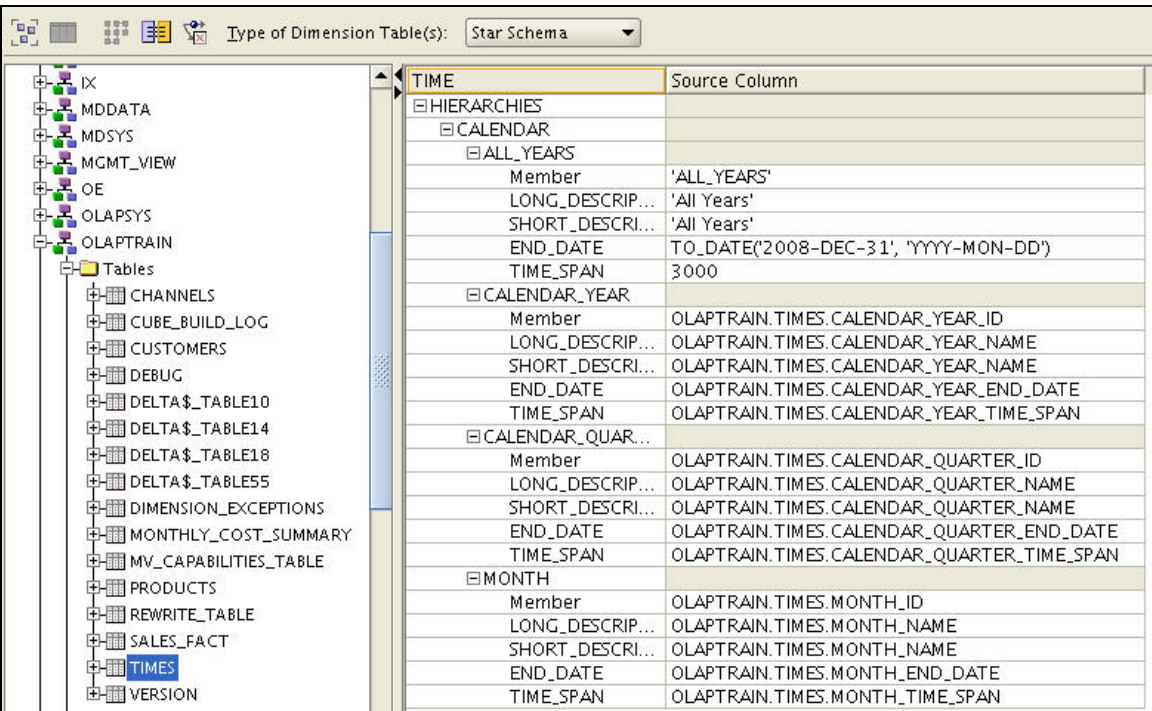


17. Click **Create**.

18. Using the TIMES table in the OLAPTRAIN schema, map the Time dimension as shown in the following image.

Note: For the ALL_YEARS level, use the following values in the Source Column fields (including single quotes):

- Member: 'ALL_YEARS'
- LONG DESCRIPTION: 'All Years'
- SHORT DESCRIPTION: 'All Years'
- END_DATE: TO_DATE('2008-DEC-31', 'YYYY-MON-DD')
- TIME_SPAN: 3000



	Source Column
TIME	
HIERARCHIES	
CALENDAR	
ALL_YEARS	
Member	'ALL_YEARS'
LONG_DESCRIP...	'All Years'
SHORT_DESCRIP...	'All Years'
END_DATE	TO_DATE('2008-DEC-31', 'YYYY-MON-DD')
TIME_SPAN	3000
CALENDAR_YEAR	
Member	OLAPTRAIN.TIMES.CALENDAR_YEAR_ID
LONG_DESCRIP...	OLAPTRAIN.TIMES.CALENDAR_YEAR_NAME
SHORT_DESCRIP...	OLAPTRAIN.TIMES.CALENDAR_YEAR_NAME
END_DATE	OLAPTRAIN.TIMES.CALENDAR_YEAR_END_DATE
TIME_SPAN	OLAPTRAIN.TIMES.CALENDAR_YEAR_TIME_SPAN
CALENDAR_QUAR...	
Member	OLAPTRAIN.TIMES.CALENDAR_QUARTER_ID
LONG_DESCRIP...	OLAPTRAIN.TIMES.CALENDAR_QUARTER_NAME
SHORT_DESCRIP...	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_DESCRIP...	OLAPTRAIN.TIMES.MONTH_NAME
SHORT_DESCRIP...	OLAPTRAIN.TIMES.MONTH_NAME
END_DATE	OLAPTRAIN.TIMES.MONTH_END_DATE
TIME_SPAN	OLAPTRAIN.TIMES.MONTH_TIME_SPAN

19. Click **Apply**.

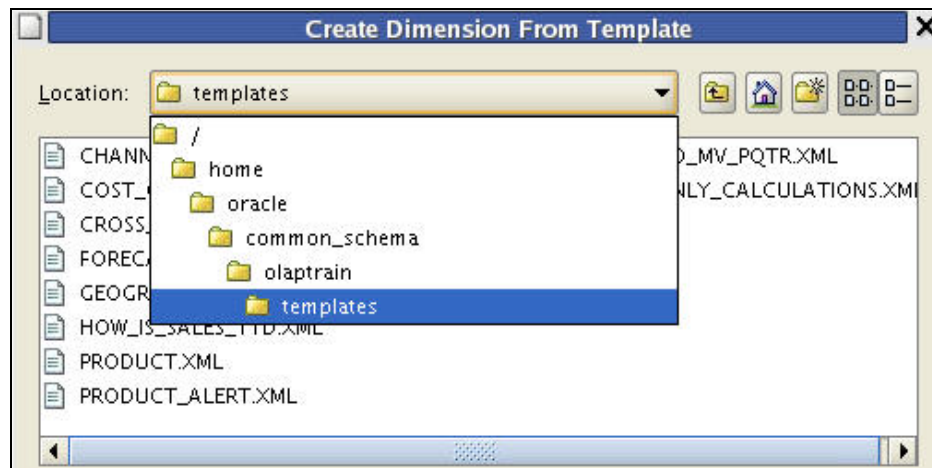
Result: The Time dimension is ready for data loading.

20. Create the Geography dimension from an XML template file.

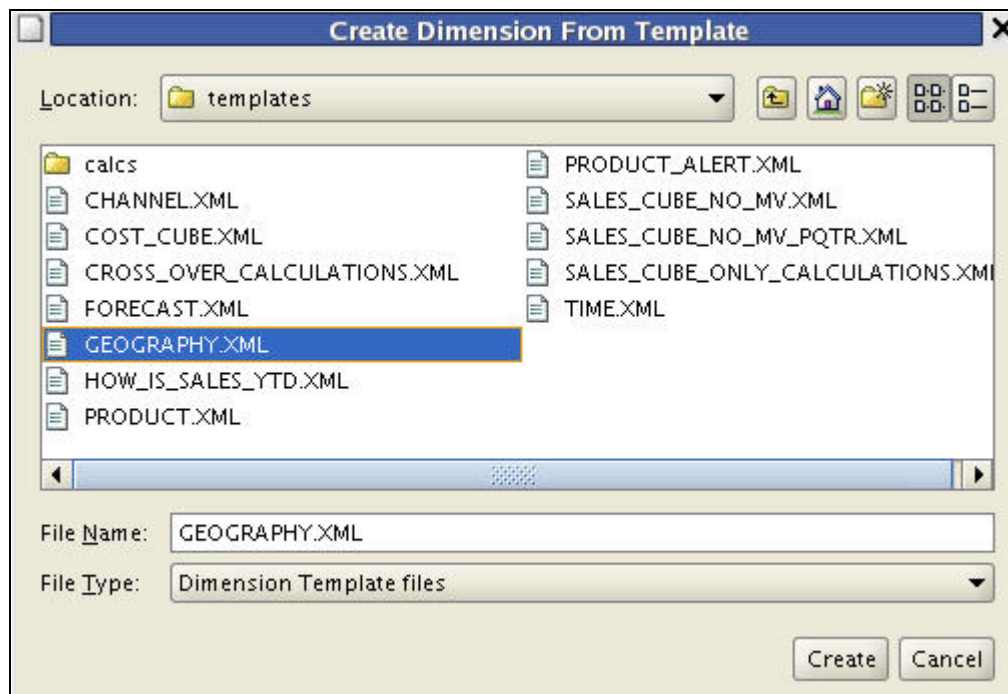
- In the navigator, right-click Dimensions and select **Create Dimension From Template**.



- In the Create Dimension From Template dialog box, navigate to the home/oracle/common_schema/olaptrain/templates folder.



- c. Select **GEOGRAPHY.XML** and click **Create**.



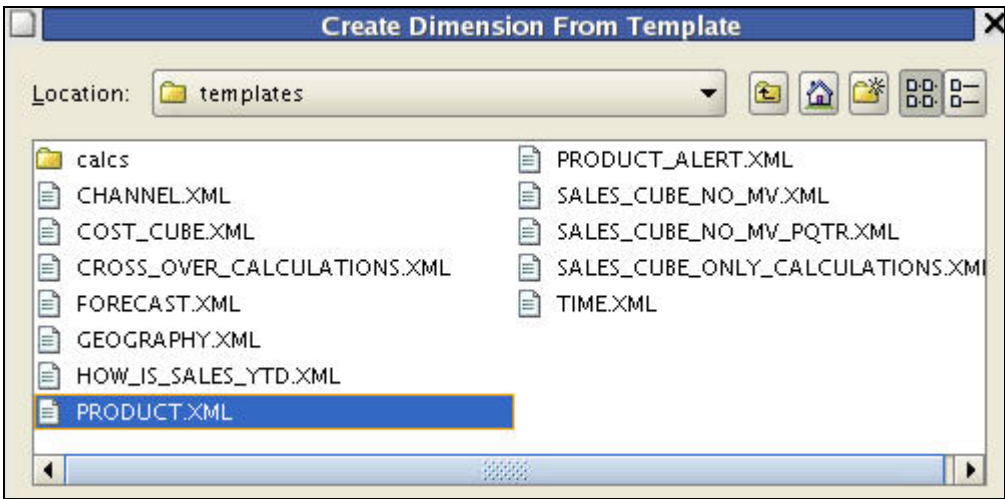
Result: The Geography dimension is created.

21. Under the Dimensions node in the navigator, select **GEOGRAPHY**, and then the **Mappings** node.

The completed mapping for the Product dimension appears.

GEOGRAPHY	Source Column
HIERARCHIES	
REGIONAL	
ALL_REGIONS	
Member	'ALL_REGIONS'
LONG_DESCRIP...	'All Regions'
SHORT_DESCRI...	'All Regions'
REGION	
Member	OLAPTRAIN.CUSTOMERS.REGION_KEY
LONG_DESCRIP...	OLAPTRAIN.CUSTOMERS.REGION_NAME
SHORT_DESCRI...	OLAPTRAIN.CUSTOMERS.REGION_NAME
COUNTRY	
Member	OLAPTRAIN.CUSTOMERS.COUNTRY_KEY
LONG_DESCRIP...	OLAPTRAIN.CUSTOMERS.COUNTRY_NAME
SHORT_DESCRI...	OLAPTRAIN.CUSTOMERS.COUNTRY_NAME
STATE_PROVINCE	
Member	OLAPTRAIN.CUSTOMERS.STATE_PROVINCE_KEY
LONG_DESCRIP...	OLAPTRAIN.CUSTOMERS.STATE_PROVINCE_NAME
SHORT_DESCRI...	OLAPTRAIN.CUSTOMERS.STATE_PROVINCE_NAME

22. Create the Product dimension from an XML template file.
 - a. In the navigator, right-click Dimensions and select **Create Dimension From Template**.
 - b. Select **PRODUCT.XML** and click **Create**.



23. Under the Dimensions node in the navigator, select **PRODUCT** and then the **Mappings** node.

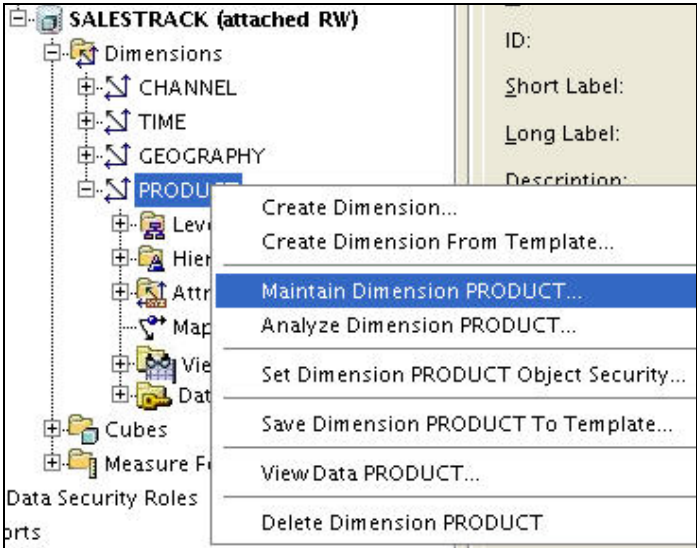
The completed mapping for the Product dimension appears.

PRODUCT	Source Column
[-] HIERARCHIES	
[-] STANDARD	
[-] ALL_PRODUCTS	
Member	'ALL_PRODUCTS'
LONG_DESCRIP...	'All Products'
SHORT_DESCRI...	'All Products'
[-] DEPARTMENT	
Member	OLAPTRAIN.PRODUCTS.DEPARTMENT_KEY
LONG_DESCRIP...	OLAPTRAIN.PRODUCTS.DEPARTMENT_NAME
SHORT_DESCRI...	OLAPTRAIN.PRODUCTS.DEPARTMENT_SHORT_DESC
[-] CATEGORY	
Member	OLAPTRAIN.PRODUCTS.CATEGORY_KEY
LONG_DESCRIP...	OLAPTRAIN.PRODUCTS.CATEGORY_NAME
SHORT_DESCRI...	OLAPTRAIN.PRODUCTS.CATEGORY_SHORT_DESC
[-] TYPE	
Member	OLAPTRAIN.PRODUCTS.TYPE_KEY
LONG_DESCRIP...	OLAPTRAIN.PRODUCTS.TYPE_NAME
SHORT_DESCRI...	OLAPTRAIN.PRODUCTS.TYPE_SHORT_DESC
[-] SUBTYPE	
Member	OLAPTRAIN.PRODUCTS.SUBTYPE_KEY
LONG_DESCRIP...	OLAPTRAIN.PRODUCTS.SUBTYPE_NAME
SHORT_DESCRI...	OLAPTRAIN.PRODUCTS.SUBTYPE_SHORT_DESC
[-] ITEM	
Member	OLAPTRAIN.PRODUCTS.ITEM_KEY
LONG_DESCRIP...	OLAPTRAIN.PRODUCTS.ITEM_NAME
SHORT_DESCRI...	OLAPTRAIN.PRODUCTS.ITEM_SHORT_DESC

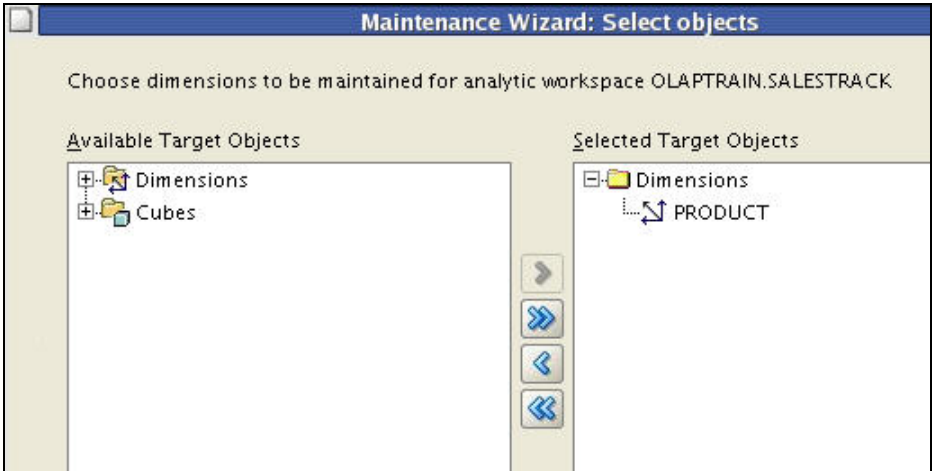
Practice 3-1c: Load and View Dimension Data

In this practice section, you load data for the Product dimension and view the results.

- 1. Load the data for the Product dimension by performing the following steps:
 - a. In the navigator, right-click **PRODUCT** and select **Maintain Dimension PRODUCT** from the menu.



Result: The Maintenance Wizard appears.

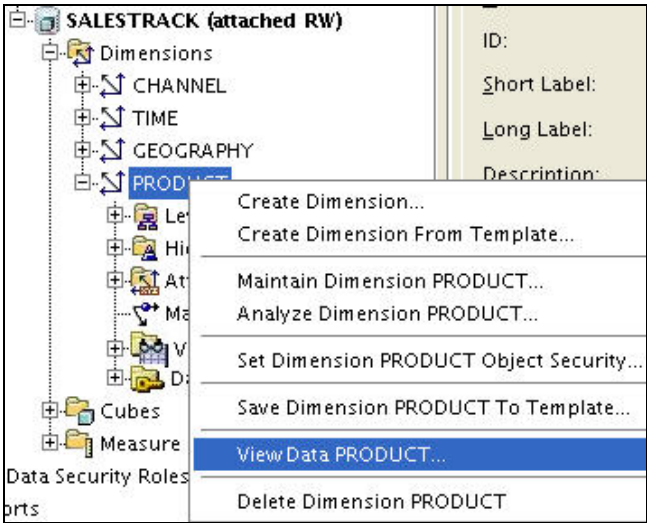


- b. Accept the defaults and click **Finish**.

Result: Data for the Product dimension is loaded and the AWM Build Log appears when the data load is finished.

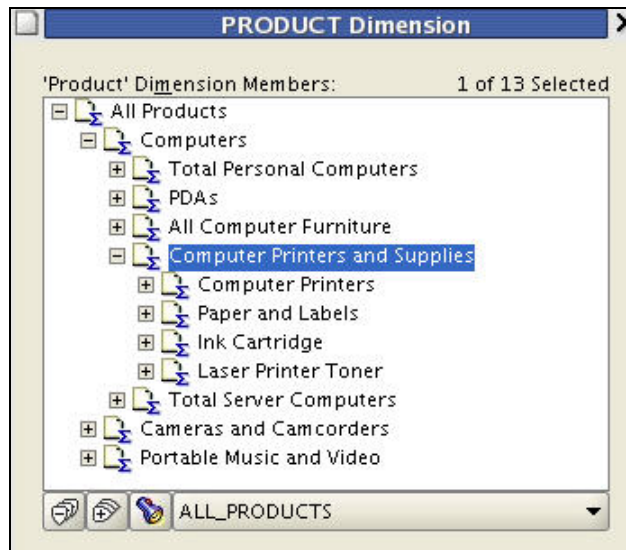
Build Log						
Fetched 7 rows						
BUILD_ID	STATUS	ELAPSED	START_TIME	END_TIME	BUILD_OBJECT	
4	COMPLETED	0 H 0M 02...	2008-8-4 1...	2008-8-4 1...		
4	COMPLETED	0 H 0M 00...	2008-8-4 1...	2008-8-4 1...		
4	COMPLETED	0 H 0M 00...	2008-8-4 1...	2008-8-4 1...	PRODUCT	
4	COMPLETED	0 H 0M 01...	2008-8-4 1...	2008-8-4 1...	PRODUCT	
4	COMPLETED	0 H 0M 00...	2008-8-4 1...	2008-8-4 1...	PRODUCT	
4	COMPLETED	0 H 0M 00...	2008-8-4 1...	2008-8-4 1...	PRODUCT	
4	COMPLETED	0 H 0M 00...	2008-8-4 1...	2008-8-4 1...	PRODUCT	

- 2. Click **Close**.
- 3. In the navigator, right-click **PRODUCT** and select **View Data PRODUCT** from the menu.



Result: The Product dimension appears in the AWM Data Viewer.

- a. You can review the hierarchical structure of a dimension using the Data Viewer. For example, select **All Products > Computers > Computer Printers and Supplies**. You should see the following display:



- b. Feel free to navigate through the dimension. After you are finished, click **Close**.
4. Next, view data from the relational Hierarchy View that was created for the Product dimension.

Note: Remember that OLAP automatically creates a Dimension view, and at least one Hierarchy view for each OLAP dimension that you create. These views are used for SQL access to the OLAP data.

- a. In the navigator, select **Views** node under the **PRODUCT** dimension, and then select **PRODUCT_STANDARD_VIEW**.



Result: The structure of the Hierarchy view appears in the General tab, displayed in the right pane.

- b. Click the **Data** tab.

Result: The following appears in the right pane:

General		Data		
Dimension: PRODUCT		View: PRODUCT_STANDARD_VIEW		Hierarchy: STANDARD
		Columns... Sort... Filter: Enter Where Clause		
	DIM_KEY	LEVEL_NAME	LONG_DESCRIPTION	SHORT_DESCRIPTION
1	ALL_PRODUCTS	ALL_PRODUCTS	All Products	All Products
2	-520	DEPARTMENT	Portable Music and Video	Portable Music and Video
3	-519	DEPARTMENT	Cameras and Camcorders	Cameras and Camcorders
4	-518	DEPARTMENT	Computers	Computers
5	-536	CATEGORY	Total Server Computers	Total Server Computers
6	-535	CATEGORY	Computer Printers and Supplies	Computer Printers and Supplies
7	-534	CATEGORY	Camcorders and Accessories	Camcorders and Accessories
8	-533	CATEGORY	All Computer Furniture	All Computer Furniture
9	-532	CATEGORY	Cameras and Accessories	Cameras and Accessories
10	-531	CATEGORY	PDA's	PDA's
11	-530	CATEGORY	Total iPlayer Family	Total iPlayer Family
12	-529	CATEGORY	Total Personal Computers	Total Personal Computers
13	-600	TYPE	PC Sound	PC Sound
14	-599	TYPE	Laser Printer Toner	Laser Printer Toner
15	-598	TYPE	Camera Accessories	Camera Accessories
16	-597	TYPE	Ink Cartridge	Ink Cartridge
17	-596	TYPE	Input Devices	Input Devices
18	-595	TYPE	Servers Systems	Servers Systems

- c. Enter the following WHERE clause in the Filter box:
LEVEL_NAME= 'CATEGORY'
- d. Press **Enter**.

Result: The following set of data should appear:

General		Data		
Dimension: PRODUCT		View: PRODUCT_STANDARD_VIEW		Hierarchy: STANDARD
		Columns... Sort... Filter: LEVEL_NAME = 'CATEGORY'		
	DIM_KEY	LEVEL_NAME	LONG_DESCRIPTION	SHORT_DESCRIPTION
1	-536	CATEGORY	Total Server Computers	Total Server Computers
2	-535	CATEGORY	Computer Printers and Supplies	Computer Printers and Supplies
3	-534	CATEGORY	Camcorders and Accessories	Camcorders and Accessories
4	-533	CATEGORY	All Computer Furniture	All Computer Furniture
5	-532	CATEGORY	Cameras and Accessories	Cameras and Accessories
6	-531	CATEGORY	PDA's	PDA's
7	-530	CATEGORY	Total iPlayer Family	Total iPlayer Family
8	-529	CATEGORY	Total Personal Computers	Total Personal Computers

Note: Later in the course, you will learn how to use these columns in SQL queries to return the desired OLAP data.

- 5. Collapse the PRODUCT dimension in the navigator.

Practice 3-2a: Create the Sales Cube

In the practices in this section, you create a cube and add Sales and Quantity measures to it. You map the data, and then enable the cube for query rewrite. Finally, you load the data and view the results.

1. Right-click the Cubes node and select **Create Cube** from the menu.
2. In the General tab of the Create Cube dialog box, specify the following:
 - a. Name: **SALES_CUBE**
 - b. Selected dimension order:
 - CHANNEL
 - TIME
 - GEOGRAPHY
 - PRODUCT

Result: The Create Cube dialog box should look like this:

The screenshot shows the 'Create Cube' dialog box with the 'General' tab selected. The 'Specify General Cube Information' section contains the following fields:

- Name: SALES_CUBE
- Short Label: Sales Cube
- Long Label: Sales Cube
- Description: Sales Cube

Below these fields, there is a section titled 'Define the dimensions for this cube by moving dimensions from the Available Dimensions list to the Selected'. The 'Available Dimensions' list is empty, and the 'Selected Dimensions' list contains the following dimensions:

- CHANNEL
- TIME
- GEOGRAPHY
- PRODUCT

3. Select the Aggregation tab. Then, in the Precompute subtab, specify a value of 30 for **Cost-based aggregation**.

The screenshot shows the 'Create Cube' dialog box with the 'Aggregation' tab selected. The 'Specify the aggregation rules of the cube' section contains the following subtabs:

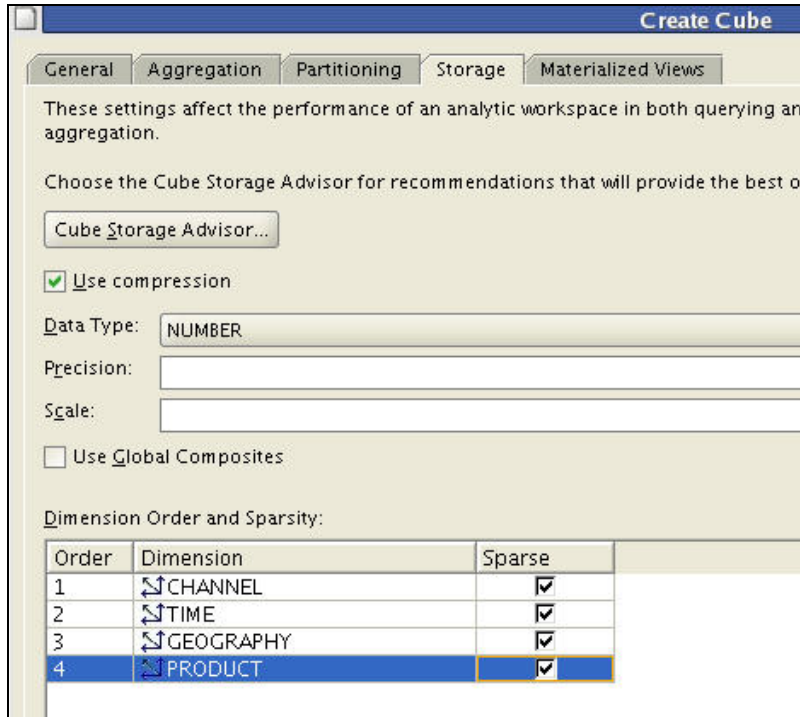
- Rules
- Precompute

The 'Precompute' subtab is selected, and the 'Choose an aggregation method:' section contains the following options:

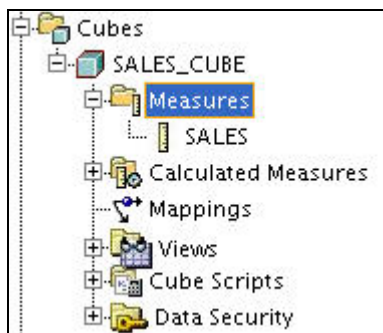
- ☒ Cost-based aggregation (recommended for compressed cubes)
- ☐ Level-based aggregation (required for uncompressed cubes)

The 'Percentage' field is set to 30.

4. Select the Storage tab. Select the **Sparse** option for all dimensions.



5. Click **Create**.
6. In the navigator, select **SALES_CUBE**.
7. Right-click Measures and select **Create Measure**.
8. In the Create Measure dialog box, enter **SALES** as the name and click **Create**.
9. In the navigator, select **Measures**. You should see the following:



10. Using the same techniques described in steps 7 and 8, create a second measure named **QUANTITY**.
11. Click the **Mappings** node under **SALES_CUBE**, and ensure that the **Table Mapping View** tool (shown below) is selected.



12. Under the Schemas node, select **OLAPTRAIN > Tables**.

13. Using the SALES_FACT, CHANNELS, TIMES, CUSTOMERS, and PRODUCTS tables, map the SALES_CUBE as shown in the following table.

Note: When mapping the Join Condition:

- 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.

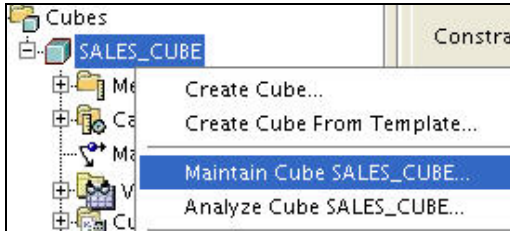
SALES_CUBE	Source Column
<input type="checkbox"/> MEASURES	
SALES	OLAPTRAIN.SALES_FACT.SALES
QUANTITY	OLAPTRAIN.SALES_FACT.QUANTITY
<input type="checkbox"/> DIMENSIONS	
<input type="checkbox"/> CHANNEL	
ALL_CHANNELS	
CLASS	
CHANNEL	OLAPTRAIN.CHANNELS.CHANNEL_KEY
Join Condition	OLAPTRAIN.SALES_FACT.CHANNEL=OLAPTRAIN.CHANNELS.CHANNEL_KEY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

14. Click **Apply**.

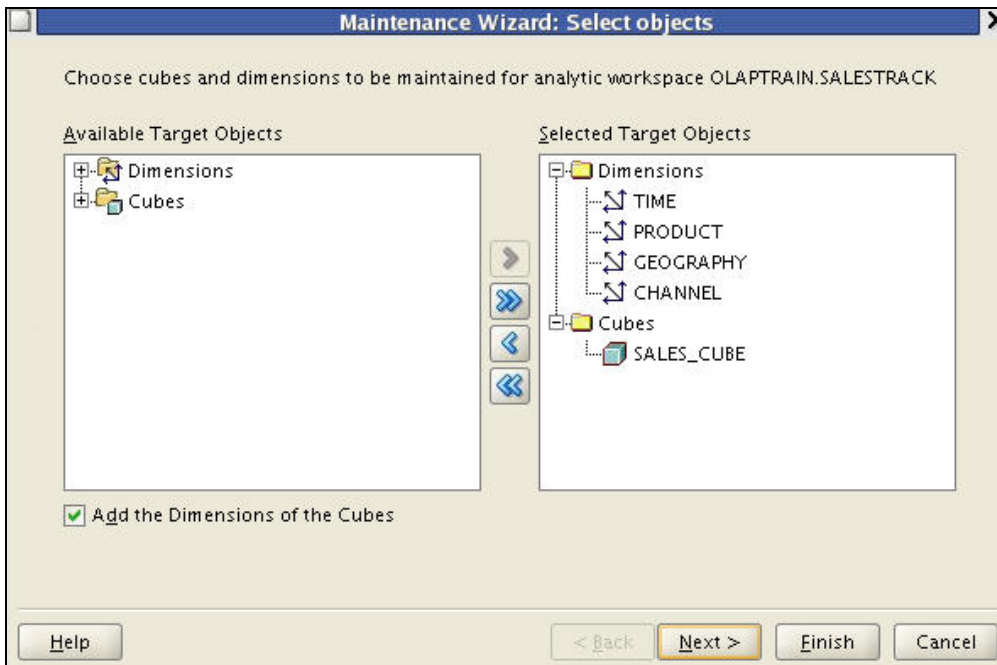
Practice 3-2b: Load and View Data for the Sales Cube

In the practices in this section, you load the data to the Sales cube, and you view cube data using both the AWM Data Viewer and the cube views.

1. In the navigator, right-click **SALES_CUBE** and select **Maintain Cube SALES_CUBE** from the menu.

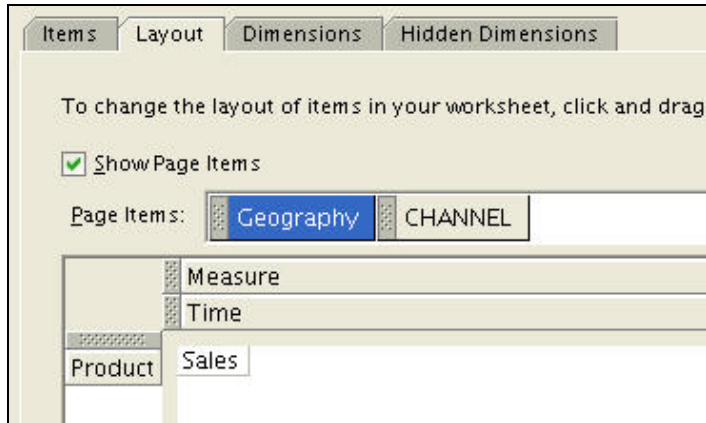



2. In the Maintenance Wizard, accept all defaults and click **Finish** to begin the load process.

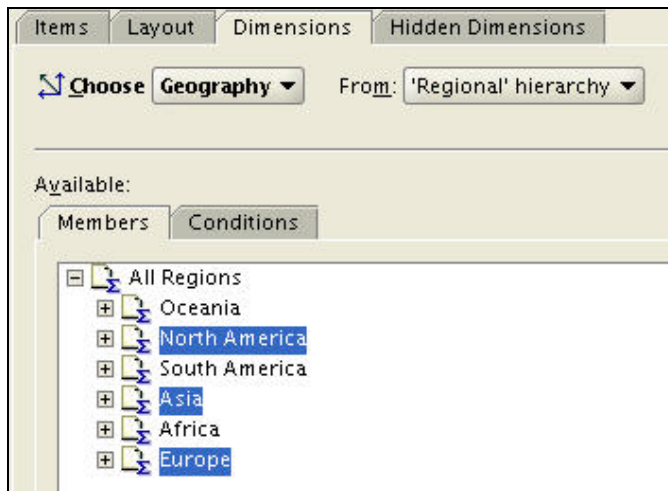



- c. Move **Product** to the row axis.

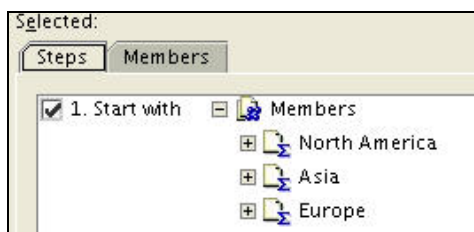
The layout tab should look like this:



8. Click the **Dimensions** tab, and then do the following:
 - a. Select **Geography** from the Choose list.
 - b. Click the **Remove All Items** tool  to clear the Selected list.
 - c. In the Members tab of the Available list, select All Regions and select **North America**, **Asia**, and **Europe**.



- d. Click the **Add Selected Items** tool  to move the chosen dimension members to the Selected list, like this:



- e. Click **OK** to refresh the data with your new selections.

9. In the Data Viewer, select **All Products**. Also, notice that you can select from a list of members for the Geography dimension.

The screenshot shows the 'Measure Data Viewer' window. The 'Page Items' section has 'Geography' set to 'North America' and 'CHANNEL' set to 'All Channels'. The table below displays sales data for 'All Products' and its sub-categories across three years: CY2005, CY2006, and CY2007.

		Sales		
		CY2005	CY2006	CY2007
▼ All Products	105,016,682.46	30,359,911.58	34,529,324.12	40,127,446.76
▶ Computers	84,554,757.65	24,306,992.93	27,792,203.99	32,455,560.73
▶ Cameras and Camcorders	8,083,827.96	2,256,981.74	2,681,928.56	3,144,917.66
▶ Portable Music and Video	12,378,096.85	3,795,936.91	4,055,191.57	4,526,968.37

10. When you are done experimenting with the report, close the Data Viewer.
11. Select the **Views** node under the SALES_CUBE.
12. Select **SALES_CUBE_VIEW** from the navigator, and click the Data tab in the right pane.

The screenshot shows the OLAP 11g administrative tool. The left pane shows the 'Views' node under 'SALES_CUBE' selected. The right pane shows the 'Data' tab for 'View: SALES_CUBE_VIEW'. The table below displays the data for the SALES_CUBE_VIEW.

	SALES	QUANTITY	TIME	PRODUCT	GEOGRAPHY	CHANNEL
1	417515017.27	2851054	ALL_YEARS	ALL_PRODUCTS	ALL_REGIONS	ALL_CHANNELS
2	49286079.48	440239	ALL_YEARS	-520	ALL_REGIONS	ALL_CHANNELS
3	49286079.48	440239	ALL_YEARS	-530	ALL_REGIONS	ALL_CHANNELS
4	31773651.06	179710	ALL_YEARS	-592	ALL_REGIONS	ALL_CHANNELS
5	28498878.54	164197	ALL_YEARS	-891	ALL_REGIONS	ALL_CHANNELS
6	6809412.34	41367	ALL_YEARS	3926	ALL_REGIONS	ALL_CHANNELS
7	771311.15	4685	ALL_YEARS	3926	-12	ALL_CHANNELS
8	6765.59	41	ALL_YEARS	3926	-69	ALL_CHANNELS
9	6765.59	41	ALL_YEARS	3926	-229	ALL_CHANNELS
10	1631.9	10	CY2005	3926	-69	ALL_CHANNELS
11	1631.9	10	CY2005	3926	-229	ALL_CHANNELS
12	849.95	5	Q1CY2005	3926	-69	ALL_CHANNELS
13	849.95	5	Q1CY2005	3926	-229	ALL_CHANNELS
14	339.98	2	FEB2005	3926	-69	ALL_CHANNELS
15	339.98	2	FEB2005	3926	-229	ALL_CHANNELS
16	169.99	1	FEB2005	3926	-69	-4
17	169.99	1	FEB2005	3926	-69	-4
18	169.99	1	FEB2005	3926	-229	-4
19	169.99	1	FEB2005	3926	-229	18
20	169.99	1	FEB2005	3926	-69	-3

Note: AWM provides a view of the relational cube view representation of the data from within the administrative tool. Later, you learn how to query the OLAP data directly by using SQL against cube views.

13. Collapse the SALES_CUBE node by clicking its minus sign.

Practices for Lesson 4

In the practices for this lesson, you enable Query Rewrite and MV Refresh for the Sales Cube, and then perform a build to create the Cube MVs. Then, you run some queries that do not rewrite correctly to the Cube MVs. Using the `mv_cube_advice` package, you determine why the queries are not rewriting. You then modify the queries so that they rewrite to the cube MVs.

Practice 4-1: Enabling and Building Cube MVs

In this practice section, you enable Cube MVs in AWM and run a build.

1. In the AWM navigator, click **SALES_CUBE**.
2. In the right pane, click the **Materialized Views** tab and select:
 - a. **Enable Materialized View Refresh of the Cube**
 - b. **Enable Query Rewrite**

General Aggregation Partitioning Storage Materialized Views

Choose this option to manage refresh of the Cube with the Materialized View refresh system

☒ Enable Materialized View Refresh of the Cube

Choose how and when to refresh of the Cube with the Materialized View refresh system

Refresh Method: Force Refresh Mode: On Demand

Start With:

Next Refresh:

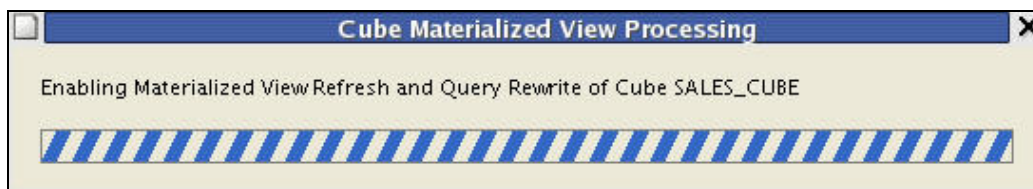
Constraints: ☒ Trusted ☐ Enforced

Choose this option to allow queries on the source tables of the Cube to be automatically rewritten to use summary

☒ Enable Query Rewrite

- c. Accept the default settings for all other options, and click **Apply**.

Result: The following information box appears:



3. When the processing is complete, go to the navigation pane and select **Dimensions > Geography**.

4. Select the **Mappings** node under Geography, and notice that OLAP has added a new Unique Key attribute to each level in the dimension, and automapped the attributes:

GEOGRAPHY	Source Column
[-] HIERARCHIES	
[-] REGIONS	
[-] ALL_REGIONS	
Member	'ALL_REGIONS'
LONG_DESCRIPTION	'All Regions'
SHORT_DESCRIPTION	'All Regions'
GEOGRAPHY_ALL_REGIONS_ID	'ALL_REGIONS'
[-] REGION	
Member	OLAPTRAIN.CUSTOMERS.REGION_KEY
LONG_DESCRIPTION	OLAPTRAIN.CUSTOMERS.REGION_NAME
SHORT_DESCRIPTION	OLAPTRAIN.CUSTOMERS.REGION_NAME
GEOGRAPHY_REGION_ID	OLAPTRAIN.CUSTOMERS.REGION_KEY
[-] COUNTRY	
Member	OLAPTRAIN.CUSTOMERS.COUNTRY_KEY
LONG_DESCRIPTION	OLAPTRAIN.CUSTOMERS.COUNTRY_NAME
SHORT_DESCRIPTION	OLAPTRAIN.CUSTOMERS.ISO_CODE
GEOGRAPHY_COUNTRY_ID	OLAPTRAIN.CUSTOMERS.COUNTRY_KEY
[-] STATE_PROVINCE	
Member	OLAPTRAIN.CUSTOMERS.STATE_PROVINCE_KEY
LONG_DESCRIPTION	OLAPTRAIN.CUSTOMERS.STATE_PROVINCE_NAME
SHORT_DESCRIPTION	OLAPTRAIN.CUSTOMERS.STATE_PROVINCE_NAME
GEOGRAPHY_STATE_PROVINCE_ID	OLAPTRAIN.CUSTOMERS.STATE_PROVINCE_KEY

These Unique Key attributes are required by the Materialized View subsystem for the query rewrite. Each level of each dimension hierarchy in the cube contains the correct additional attributes.

5. In the navigator, right-click SALES_CUBE and select **Maintain Cube SALES_CUBE** from the menu.
6. In the Maintenance Wizard, accept all defaults and click **Finish** to begin the load process.
7. When the build completes, close the Build Log window.
8. Exit AWM.

Practice 4-2: Running Summary Queries with and Without Query Rewrite

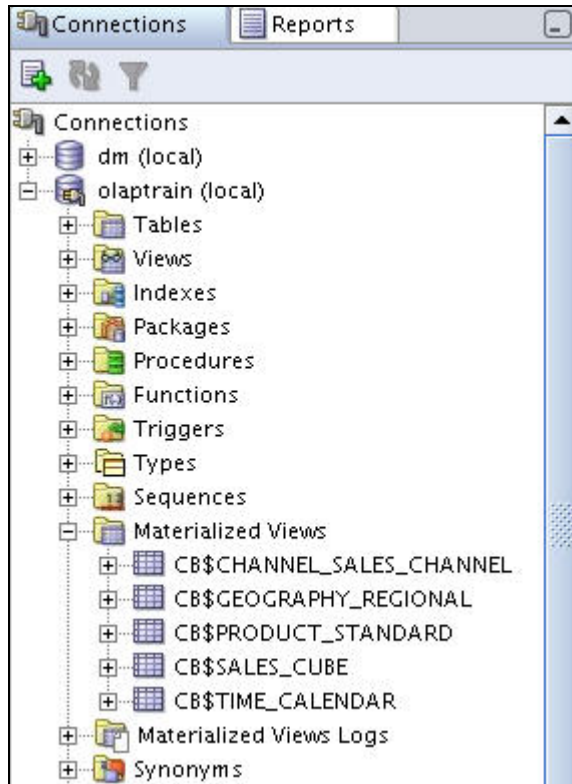
In this practice section, you run a set of summary queries against the OLAPTRAIN schema. First, you turn query rewrite off to see how the summary queries perform against the fact table. Then, you turn on rewrite and run the queries again.

- Observe how the database automatically rewrites the summary queries to the OLAP Cube MVs.
- Compare the performance of the SQL summary queries to the OLAP Cube MVs queries.

1. Launch **SQL Developer** from the desktop.
2. In the Connections tab of the navigator, click the **olaptrain** connection node.

Result: A connection to the olaptrain schema is made, and schema object types appear in the navigator.

3. In the navigator, select the **Materialized Views** node.



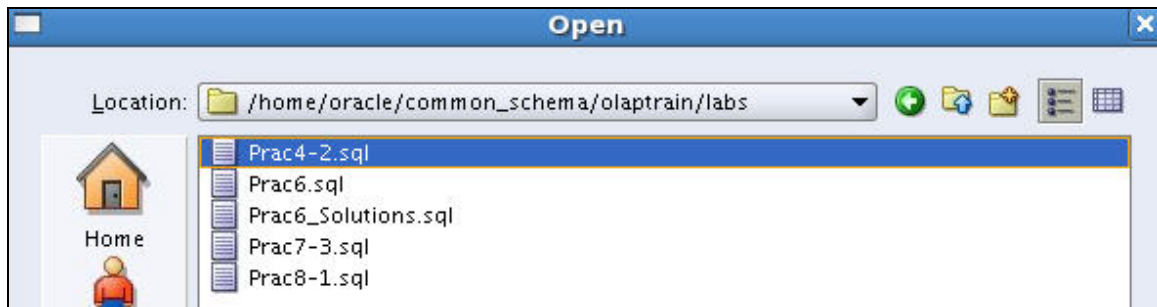
The CB\$ tables – one each for the cube and the associated dimension hierarchies – were automatically created by the OLAP build process.

4. Close the **olaptrain (local)** tab by clicking the **X**.

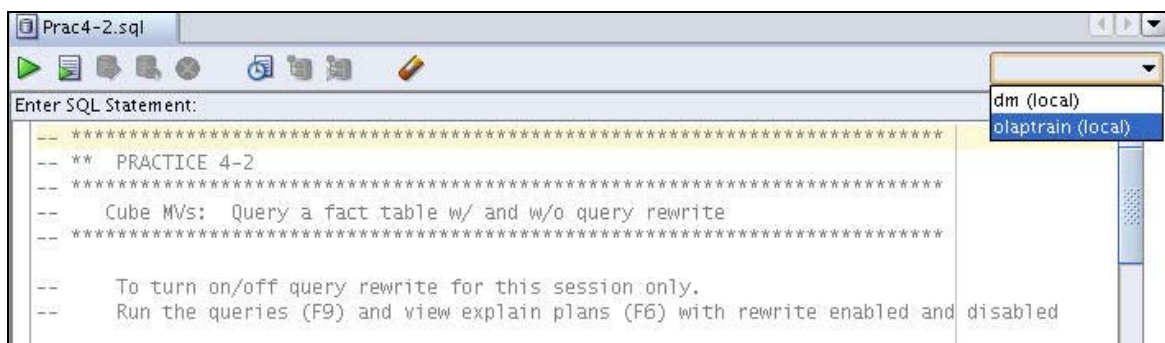


Next, you will see how these Cube MVs may be used for rewrite when summary queries are executed against the OLAPTRAIN fact and dimension tables.

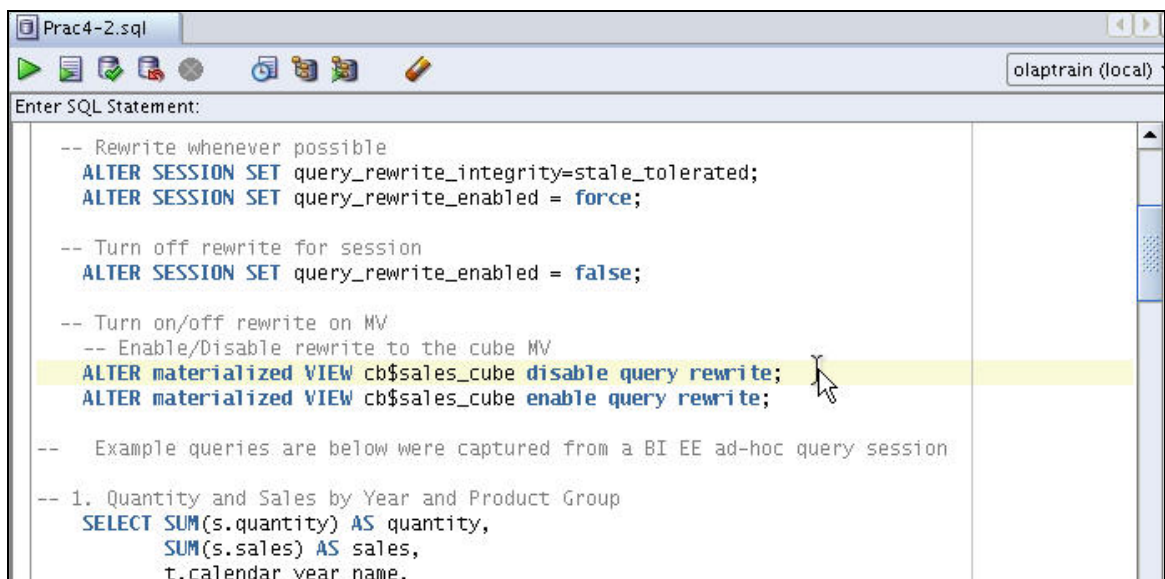
5. From the main menu, select **File > Open**.
 - a. Navigate to the `/home/oracle/common_schema/olaptrain/labs` folder.
 - b. Select the **Prac4-2.sql** file.



- c. Click **Open**.
6. Vertically resize the SQL statement pane so that it takes up at least half of the space in the SQL Developer window.
7. At the top-right corner of the Enter SQL Statement pane, select **olaptrain** from the list.



8. Scroll down until you reach the **ALTER materialized VIEW** statements. Click the statement that disables query rewrite for `cb$sales_cube`.



9. Press **F9** or click the  tool to execute the statement.

Result: Query rewrite to the Sales Cube MV is now disabled.

10. Place your cursor anywhere inside the first query, Quantity and Sales by Year and Product Group, which is just below the statements that turn query rewrite on and off.
11. Press **F6** to display the Explain Plan for the query.

Result: When run, the query joins the fact table to associated dimension tables and performs a full table scan of the fact table in order to return data the requested data.

Results Script Output Explain Autotrace DBMS Output OWA Output				
Operation		Optimizer	Cost	Cardinality
SELECT STATEMENT		ALL_ROWS	4238	23
HASH(GROUP BY)			4238	23
HASH JOIN			4236	7419
TABLE ACCESS(FULL) OLAPTRAIN.PRODUCTS		ANALYZED	40	2713
VIEW SYS.VW_GBC_9			4196	7419
HASH(GROUP BY)			4196	7419
HASH JOIN			4018	2811097
PART JOIN FILTER(CREATE) SYS.:BF0000			66	1461
PARTITION RANGE(ALL)			66	1461
TABLE ACCESS(FULL) OLAPTRAIN.TIMES		ANALYZED	66	1461
PARTITION RANGE(JOIN-FILTER)			3933	2811097
TABLE ACCESS(FULL) OLAPTRAIN.SALES_FACT		ANALYZED	3933	2811097

12. Press **F9** to execute the query.

Result: A report similar to the following example appears in the Results tab.

<pre>-- Quantity and Sales by Year and Product Group SELECT SUM(s.quantity) AS quantity, SUM(s.sales) AS sales, t.calendar_year_name, p.category_name FROM times t, products p,</pre>				
Results Script Output Explain Autotrace DBMS Output OWA Output				
Results:				
	QUANTITY	SALES	CALENDAR_YEAR_NAME	CATEGORY_NAME
1	145934	16264253.09	CY2006	Total iPlayer Family
2	79348	4515305.6	CY2006	Camcorders and Accessories
3	156	62522	CY2005	All Computer Furniture
4	411	93239.61	CY2006	PDA's
5	72008	4068490.8	CY2005	Camcorders and Accessories
6	49805	7058155.94	CY2007	Cameras and Accessories
7	625899	104944431.56	CY2007	Total Personal Computers
8	5382	2100385	CY2007	Total Server Computers

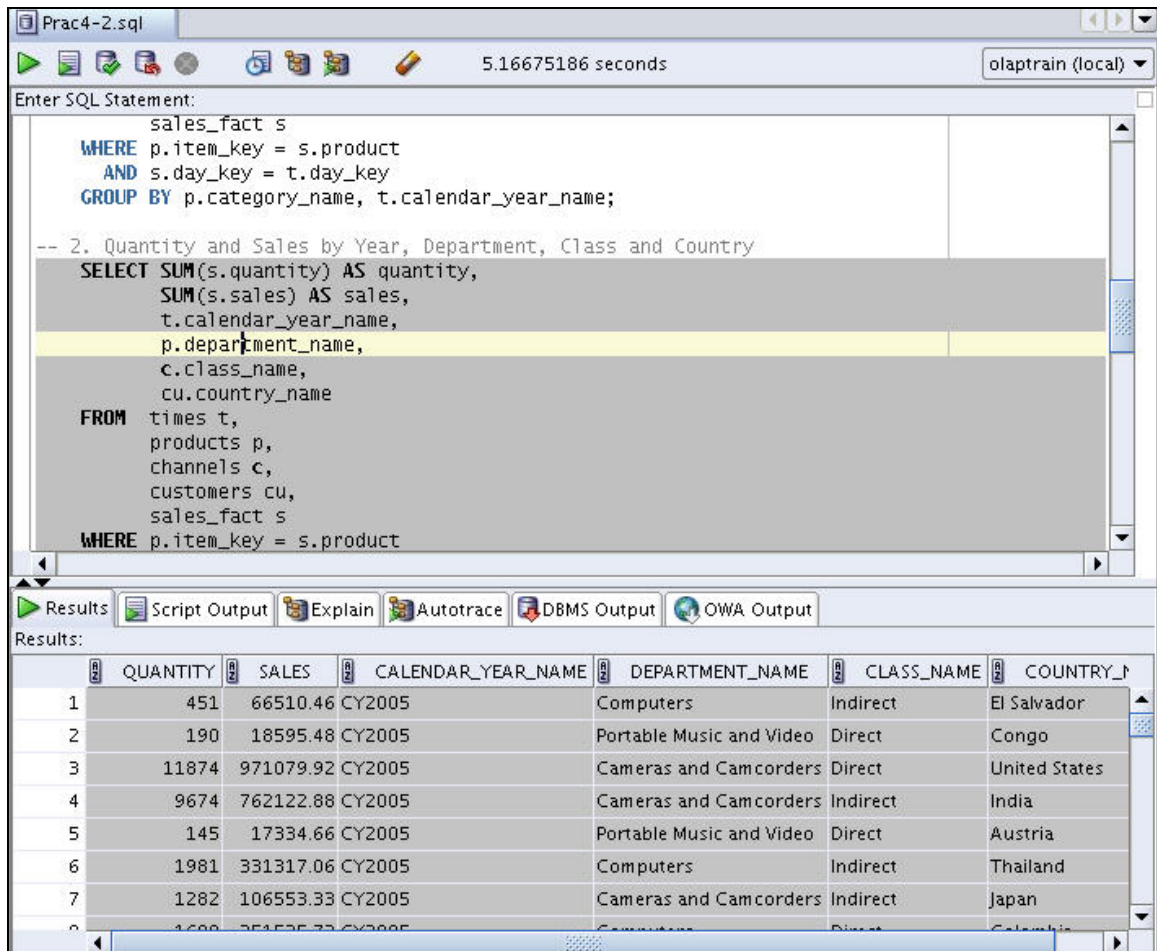
13. Record the time it took to run the query, up to two decimal places.

Note: The value appears at the top of the SQL pane, as shown in the following screenshot. In the example, the query time would be rounded to **2.54 seconds**. The value you see will probably be different from the example, although it should be close.



Query 1 performance: **seconds**

14. Click inside the second query and press **F9** to execute the query, as the following screenshot shows. Again, record your query time.



Query 2 performance: **seconds**

15. Using the same technique as in steps 10-13, execute the third and fourth queries. Record each of the query times.

Query 3 performance: **seconds**

Query 4 performance: **seconds**

16. Scroll back up in the SQL file. Enable query rewrite to **cb\$sales_cube** by clicking the enable query rewrite statement, and then pressing **F9**.

```
Enter SQL Statement:
-- Turn on/off rewrite on MV
-- Enable/Disable rewrite to the cube MV
ALTER materialized VIEW cb$sales_cube disable query rewrite;
ALTER materialized VIEW cb$sales_cube enable query rewrite;
```

17. To confirm that the queries will rewrite, click within the first query and press **F6** to display the Explain Plan, as shown in the following screenshot:

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

Operation	Optimizer	Cost	Cardinality	Byte
SELECT STATEMENT	ALL_ROWS	12	2859	21
HASH(GROUP BY)		12	2859	21
MAT_VIEW REWRITE CUBE ACCESS OLAPTRAIN.CB\$SALES_CUBE		11	2859	21

18. Press **F9** to execute the query. Record your performance to two decimal places.

Prac4-2.sql | 0.18976898 seconds | olaptrain (local)

Enter SQL Statement:

```
ALTER SESSION SET query_rewrite_enabled = false;
ALTER SESSION SET query_rewrite_enabled = true;
-- Turn off rewrite for session
ALTER materialized VIEW cb$sales_cube disable query rewrite;
ALTER materialized VIEW cb$sales_cube enable query rewrite;
-- Example queries are below were captured from a BI EE ad-hoc query session
-- 1. Quantity and Sales by Year and Product Group
SELECT SUM(s.quantity) AS quantity,
SUM(s.sales) AS sales,
t.calendar_year_name,
p.category_name
FROM times t,
products p
```

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

Results:

	QUANTITY	SALES	CALENDAR_YEAR_NAME	CATEGORY_NAME
1	145934	16264253.09	CY2006	Total iPlayer Family
2	79348	4515305.6	CY2006	Camcorders and Accessories
3	156	62522	CY2005	All Computer Furniture
4	411	93239.61	CY2006	PDA's
5	72008	4068490.8	CY2005	Camcorders and Accessories
6	49805	7058155.94	CY2007	Cameras and Accessories

Query 1 – Cube MV performance: **seconds**

19. Using the same technique as in step 18, execute the second through fourth queries. Record each of the query times.

Query 2 – Cube MV performance: **seconds**

Query 3 – Cube MV performance: **seconds**

Query 4 – Cube MV performance: **seconds**

Note: Query times for a Cube MV rewrite commonly are 10 to 50 times faster than summary queries against relational fact tables.

20. **Close** the Prac4_2.sql file.
21. **Exit** SQL Developer.