What Are Analytic Views?

Analytic views provide a fast and efficient way to create analytic queries of data stored in existing database tables and views.

Analytic views organize data using a dimensional model. They allow you to easily add aggregations and calculations to data sets and to present data in views that can be queried with relatively simple SQL.

Like standard relational views, analytic views:

* Are metadata objects (that is, they do not store data)
* Can be queried using SQL
* Can access data from other database objects such as tables, views, and external tables
* Can join multiple tables into a single view

Analytic views also:

* Organize data using a rich business model that has dimensional and hierarchical concepts
* Include system-generated columns with hierarchical data
* Automatically aggregate data
* Include embedded measure calculations that are easily defined using syntax based on the business model
* Include presentation metadata

The definition of an analytic view includes navigation, join, aggregation, and calculation rules, thus eliminating the need to include these rules in queries. Rather than having simple tables and complex SELECT statements that express joins, aggregations, and measure calculations, you can use simple SQL to query smart analytic views. This approach has several benefits, including:

* Simplified and faster application development; it is much easier to define calculations within analytic views than it is to write or generate complex SELECT statements
* Calculation rules can be defined once in the database and then be re-used by any number of applications; this provides end-users with greater freedom of choice in their use of reporting tools without concern for inconsistent results

Analytic views are especially useful for the following users:

* Data warehouse architect or designer
* Business Intelligence application developer
* Database analyst

For a data warehouse architect, analytic views are a tool for presenting data in a data warehouse to application developers and business users. Tools provided by the BI application generate a query, get the data, and present the result.

Components of Analytic Views

Analytic view component objects consist of the following:.

* Attribute dimensions, which are metadata objects that reference tables or views and organize columns into higher-level objects such as *attributes* and *levels*. Most metadata related to dimensions and hierarchies is defined in the attribute dimension object.
* Hierarchies, which are a type of view that reference attribute dimension objects and that organize data using hierarchical relationships. Data related to dimensions and hierarchies is selected from hierarchies.
* Analytic view objects, which are a type of view that presents fact data. Analytic views reference both fact tables and hierarchies. You can select both hierarchy and measure data from analytic views.

Data dictionary views, such as ALL\_ANALYTIC\_VIEW\_COLUMNS, contain the metadata and other information for the analytic view component objects.

The DBMS\_HIERARCHY PL/SQL package contains functions for validating analytic view and hierarchy objects and a procedure that creates a table that you can use for logging messages generated by the validation functions.

Data Sources for Analytic Views

Attribute dimensions and analytic views typically use [**star schema**](https://docs.oracle.com/en/database/oracle/oracle-database/12.2/dwhsg/glossary.html#GUID-753234A6-E8CE-4610-8C82-94EFFDB2FBC5) dimension tables and fact tables as data sources. For larger data sets, tables in the in-memory column store can offer the best query performance with analytic views. Analytic views can also be used with external tables and remote tables.

You specify the data source with the using\_clause in the attribute dimension or analytic view definition. You may specify an alias for the data source.

A database user who has the privileges required for access to the data sources can create analytic view objects. The creator defines the business model, which specifies how the data is queried, and implements the model by creating attribute dimensions, hierarchies, and analytic views.

Materialized Views and Analytic Views

Creating a materialized view over queries of an analytic view or a hierarchy is not supported. You may use a materialized view in a MEASURE\_GROUP phrase of a cache\_clause of an analytic view.

Constraints for Analytic View Objects

For optimal query performance in queries of an analytic view, you should use the same constraints that you would typically use for querying a star schema. An attribute dimension or analytic view does not require that the source table or view have any particular constraints defined or enabled. Also, defining an attribute dimension or analytic view does not introduce any additional constraints on those tables or views. The PL/SQL functions VALIDATE\_HIERARCHY and VALIDATE\_ANALYTIC\_VIEW are available for validating that the data in a table or view used by an attribute dimension in a hierarchy or used by an analytic view conforms to the logical constraints inherent in the metadata definitions.

Naming Conventions for Analytic Views

The naming conventions for attribute dimensions, hierarchies, and analytic views, and components of them such as attributes, levels, and measures, follow standard database identifier rules. Double-quotes may be used to enclose identifiers, including extended characters and mixed-case; otherwise, the standard upper-case and limited character rules apply.

24.2 Privileges for Analytic Views

Describes the system and object privileges available for analytic views, attribute dimensions, and hierarchies.

System Privileges

The following system privileges allow the user to create, alter, or drop analytic view component objects.

| **System Privilege** | **Description** |
| --- | --- |
| CREATE ANALYTIC VIEW | Create an analytic view in the grantee's schema. |
| CREATE ANY ANALYTIC VIEW | Create analytic views in any schema except SYS. |
| CREATE ATTRIBUTE DIMENSION | Create an attribute dimension in the grantee's schema. |
| CREATE ANY ATTRIBUTE DIMENSION | Create attribute dimensions in any schema except SYS. |
| CREATE HIERARCHY | Create a hierarchy in the grantee's schema. |
| CREATE ANY HIERARCHY | Create hierarchies in any schema except SYS. |
| ALTER ANY ANALYTIC VIEW | Rename analytic views in any schema except SYS. |
| ALTER ANY ATTRIBUTE DIMENSION | Rename attribute dimensions in any schema except SYS. |
| ALTER ANY HIERARCHY | Rename hierarchies in any schema except SYS. |
| DROP ANY ANALYTIC VIEW | Drop analytic views in any schema except SYS. |
| DROP ANY ATTRIBUTE DIMENSION | Drop attribute dimensions in any schema except SYS. |
| DROP ANY HIERARCHY | Drop hierarchies in any schema except SYS. |
| SELECT ANY TABLE | Query or view any analytic view or hierarchy in any schema. |

Object Privileges

The following object privileges allow the user to query or rename analytic view component objects.

| **Object Privilege** | **Operations Authorized** |
| --- | --- |
| ALTER | Rename the analytic view, attribute dimension, or hierarchy. |
| READ | Query the object with the SELECT statement. |
| SELECT | Query the object with the SELECT statement. |

Example 24-1 Granting System Privileges

The following statements grant the CREATE system privilege to the user av\_user.

CopyGRANT CREATE ATTRIBUTE DIMENSION TO av\_user;

GRANT CREATE HIERARCHY TO av\_user;

GRANT CREATE ANALYTIC VIEW TO av\_user;

GRANT SELECT ANY TABLE TO av\_user;

Example 24-2 Granting Object Privileges

The following statements grant all object privileges to the user av\_user2 and then revoke the ALTER privilege.

CopyGRANT ALL ON "AV\_USER".SALES\_AV TO "AV\_USER2";

REVOKE ALTER ON "AV\_USER".SALES\_AV FROM "AV\_USER2";

24.3 Application Programming Interfaces for Analytic Views

The application programming interfaces for analytic views consist of SQL DDL statements, PL/SQL procedures and functions, and data dictionary views.

These interfaces are listed in the following topics:

* [SQL DDL Statements for the Creation and Management of Analytic Views](https://docs.oracle.com/en/database/oracle/oracle-database/12.2/dwhsg/overview-analytic-views.html#GUID-FC745EA1-721D-48E3-96D1-B4DE1758AFF8__SQLDDLSTATEMENTSFORTHECREATIONANDMA-7DCA4A27)
* [PL/SQL Package for Analytic Views](https://docs.oracle.com/en/database/oracle/oracle-database/12.2/dwhsg/overview-analytic-views.html#GUID-FC745EA1-721D-48E3-96D1-B4DE1758AFF8__PLSQLPACKAGEFORANALYTICVIEWS-7DCA5667)
* [Data Dictionary Views for Analytic Views](https://docs.oracle.com/en/database/oracle/oracle-database/12.2/dwhsg/overview-analytic-views.html#GUID-FC745EA1-721D-48E3-96D1-B4DE1758AFF8__DATADICTIONARYVIEWSFORANALYTICVIEWS-7DCA4E84)

SQL DDL Statements for the Creation and Management of Analytic Views

You create and manage analytic view objects with the following SQL DDL statements:

* CREATE ANALYTIC VIEW
* CREATE ATTRIBUTE DIMENSION
* CREATE HIERARCHY
* ALTER ANALYTIC VIEW
* ALTER ATTRIBUTE DIMENSION
* ALTER HIERARCHY
* DROP ANALYTIC VIEW
* DROP ATTRIBUTE DIMENSION
* DROP HIERARCHY

For details about these statements, see [CREATE ANALYTIC VIEW](https://docs.oracle.com/pls/topic/lookup?ctx=en/database/oracle/oracle-database/12.2/dwhsg&id=SQLRF-GUID-EBA7E9BC-3F49-4AA7-9EF6-9255FE7AE466) and the other statements in *Oracle Database SQL Language Reference*.

PL/SQL Package for Analytic Views

You can validate the data for analytic view and hierarchy objects with the following procedures in the DBMS\_HIERARCHY package:

* CREATE\_VALIDATE\_LOG\_TABLE procedure
* VALIDATE\_ANALYTIC\_VIEW function
* VALIDATE\_CHECK\_SUCCESS function
* VALIDATE\_HIERARCHY function

For details about this package, see [DBMS\_HIERARCHY](https://docs.oracle.com/pls/topic/lookup?ctx=en/database/oracle/oracle-database/12.2/dwhsg&id=ARPLS-GUID-BA9CF136-9144-4880-BFF7-67B369E81D62) in *Oracle Database PL/SQL Packages and Types Reference*.

Data Dictionary Views for Analytic Views

The following data dictionary views contain information about analytic view objects. Only the views with the prefix ALL are listed. Each view also has a corresponding DBA and USER version.

**Analytic View Views**

* ALL\_ANALYTIC\_VIEW\_ATTR\_CLASS
* ALL\_ANALYTIC\_VIEW\_BASE\_MEAS
* ALL\_ANALYTIC\_VIEW\_CALC\_MEAS
* ALL\_ANALYTIC\_VIEW\_CLASS
* ALL\_ANALYTIC\_VIEW\_COLUMNS
* ALL\_ANALYTIC\_VIEW\_DIM\_CLASS
* ALL\_ANALYTIC\_VIEW\_DIMENSIONS
* ALL\_ANALYTIC\_VIEW\_HIER\_CLASS
* ALL\_ANALYTIC\_VIEW\_HIERS
* ALL\_ANALYTIC\_VIEW\_KEYS
* ALL\_ANALYTIC\_VIEW\_LEVEL\_CLASS
* ALL\_ANALYTIC\_VIEW\_LEVELS
* ALL\_ANALYTIC\_VIEW\_LVLGRPS
* ALL\_ANALYTIC\_VIEW\_MEAS\_CLASS
* ALL\_ANALYTIC\_VIEWS

**Attribute Dimension Views**

* ALL\_ATTRIBUTE\_DIM\_ATTR\_CLASS
* ALL\_ATTRIBUTE\_DIM\_ATTRS
* ALL\_ATTRIBUTE\_DIM\_CLASS
* ALL\_ATTRIBUTE\_DIM\_JOIN\_PATHS
* ALL\_ATTRIBUTE\_DIM\_KEYS
* ALL\_ATTRIBUTE\_DIM\_LEVEL\_ATTRS
* ALL\_ATTRIBUTE\_DIM\_LEVELS
* ALL\_ATTRIBUTE\_DIM\_LVL\_CLASS
* ALL\_ATTRIBUTE\_DIM\_ORDER\_ATTRS
* ALL\_ATTRIBUTE\_DIM\_TABLES
* ALL\_ATTRIBUTE\_DIMENSIONS

**Hierarchy Views**

* ALL\_HIER\_CLASS
* ALL\_HIER\_COLUMNS
* ALL\_HIER\_HIER\_ATTR\_CLASS
* ALL\_HIER\_HIER\_ATTRIBUTES
* ALL\_HIER\_JOIN\_PATHS
* ALL\_HIER\_LEVEL\_ID\_ATTRS
* ALL\_HIER\_LEVELS
* ALL\_HIERARCHIES

For details about these views, see [ALL\_ANALYTIC\_VIEWS](https://docs.oracle.com/pls/topic/lookup?ctx=en/database/oracle/oracle-database/12.2/dwhsg&id=REFRN-GUID-DA5210A9-C25C-4EEC-B1D6-B7217FB532A7) and the other views in *Oracle Database Reference*.

24.4 Compilation States of Analytic Views

When you create or alter an attribute dimension, a hierarchy, or an analytic view, Oracle Database ascertains the internal validity of the object’s metadata.

The SQL DDL CREATE and ALTER statements for analytic views have FORCE and NOFORCE options, with NOFORCE as the default. The verification of metadata that depends on another object is optional and is determined by the FORCE and NOFORCE options.

If you specify NOFORCE and the compilation fails, then the CREATE or ALTER operation fails and an error is raised. If you specify FORCE, the CREATE or ALTER succeeds even if the compilation fails.

You can explicitly invoke a compilation by specifying the COMPILE keyword; a compilation is implicitly invoked as needed during a query. A query returns an error if an object is not compiled and cannot implicitly be compiled.

The compilation state is recorded in the COMPILE\_STATE column in the ALL\_ATTRIBUTE\_DIMENSIONS, ALL\_HIERARCHIES, and ALL\_ANALYTIC\_VIEWS data dictionary views (and the corresponding DBA and USER views). The state may be one of the following:

| **Value** | **Description** |
| --- | --- |
| VALID | The object has been compiled without error. |
| INVALID | Some change requires recompilation or the object has been compiled and errors have occurred. |

A SQL DDL operation on the analytic views object causes the state of dependent objects to change to INVALID. For example, a change to an attribute dimension causes any hierarchies that use that attribute dimension, and analytic views dimensioned by the attribute dimension, to change state to INVALID. Also, DDL changes to the tables or views used by attribute dimensions and analytic views cause the state for those objects to change to INVALID.

The ALL\_OBJECTS data dictionary view has a STATUS column that may be VALID or INVALID. For attribute dimensions, hierarchies, and analytic views, the STATUS value correlates to the COMPILE\_STATE. When COMPILE\_STATE is VALID, the STATUS value is VALID. When COMPILE\_STATE is INVALID, STATUS is INVALID.

24.5 Validation of Data

To ensure the accuracy of query results, the data of hierarchies and analytic views must be validated.

To validate the data for a hierarchy or analytic view, use the functions in the PL/SQL package DBMS\_HIERARCHY. The VALIDATE\_HIERARCHY and VALIDATE\_ANALYTIC\_VIEW functions validate the data and store the results in a table. An optional argument to the functions is the name of a table. The CREATE\_VALIDATE\_LOG\_TABLE procedure creates a table that you can use for the purpose. If you do not specify a table, the VALIDATE\_HIERARCHY and VALIDATE\_ANALYTIC\_VIEW functions create a table.

Any SQL DDL or DML changes made on the tables used by an associated attribute dimension or analytic view, or any DDL change to an attribute dimension, hierarchy, or analytic view itself, causes the states of a hierarchy to change to INVALID.

If any data security policies are applied to a hierarchy or analytic view, or any of the tables or views used by an associated attribute dimension, then the validation state cannot be determined and the VALIDATE\_STATE is not set to VALID. An execution of the VALIDATE\_HIERARCHY or VALIDATE\_ANALYTIC\_VIEW function indicates whether the hierarchy or analytic view is valid at that time and for that user.

If a SQL DML change to a table or view used by an attribute dimension occurs between the time you query the data dictionary or run the VALIDATE\_HIERARCHY function and the time you execute a query of a hierarchy or analytic view, then the hierarchy may become invalid. To ensure that a hierarchy is valid for a query, you can establish a read-only transaction (for example, SET TRANSACTION READ ONLY), run the validation function, verify the success of the validation, execute queries, and then end the transaction with a COMMIT or ROLLBACK statement.

24.6 Classifications for Analytic Views

Classifications provide descriptive metadata for attribute dimensions, hierarchies, and analytic view objects, and for components of them such as attribute dimension keys, attributes, levels, and measures.

Applications can use classifications to present information about hierarchies and analytic views. Classifications are similar to comments on tables and columns, but a comment is a single value. You can specify any number of classifications for the same object. You can vary the values by language. A classification value is always a text literal and can have maximum length of 4000 bytes.

Classifications play no role in SQL queries, but are available in the data dictionary views for use by tools or applications. The CAPTION and DESCRIPTION classifications have DDL shortcuts for all objects that support classifications.

You may specify a language for a classification value. If you do not specify a language, then the language value for the classification is NULL. The language value must either be NULL or a valid NLS\_LANGUAGE value.

The DDL shortcuts for CAPTION and DESCRIPTION apply only to the NULL language. To specify a CAPTION and DESCRIPTION classification for a particular language, you must use the full CLASSIFICATION syntax.

SQL tools can interpret a NULL language value as a default. For example, suppose a tool is looking for the CAPTION for an attribute dimension. The tool might first look for the CAPTION having a language that matches the current NLS\_LANGUAGE. If it finds one, it uses that CAPTION value. If not, it then looks for a CAPTION having a NULL language value and uses that. The SQL logic is up to the user, tool, or application.

To provide descriptive metadata that varies by language for a member of a hierarchy, use the hierarchical attributes MEMBER\_NAME, MEMBER\_CAPTION, and MEMBER\_DESCRIPTION.

24.7 Share Analytic Views with Application Containers

You can share analytic views with application containers.

In the definition of analytic view objects, you can use the SHARING clause to share attribute dimension, hierarchy, or analytic view metadata or objects with application containers. The values for the clause are the following:

| **Value** | **Description** |
| --- | --- |
| NONE | Do not share; this is the default value. |
| METADATA | Share metadata only. |
| OBJECT | Share the object, including data. |

If you specify METADATA, then only the definition of the object is shared with application containers.

If you specify OBJECT, then the attribute dimension, hierarchy, or analytic view object, including the data sources of the object, is shared with the application container.

24.8 Alter or Drop an Analytic View Object

With SQL DDL statements you can change the name of an object or you can drop it.

To alter any aspect of an analytic view object other than the name, use a CREATE OR REPLACE statement to replace the object with one that has the desired alterations.

Example 24-3 Renaming an Attribute Dimension

The following example renames an attribute dimension.

CopyALTER ATTRIBUTE DIMENSION product\_attr\_dim RENAME TO myproduct\_attr\_dim;

Example 24-4 Dropping an Attribute Dimension

The following example drops an attribute dimension.

CopyDROP ATTRIBUTE DIMENSION myproduct\_attr\_dim;

24.9 Data and Scripts for Examples

This section describes the data on which the analytic views examples are based and contains SQL statements that create the analytic view component objects.

The data and the analytic view components are described in the following topics:

* [About the Data and Scripts for Examples](https://docs.oracle.com/en/database/oracle/oracle-database/12.2/dwhsg/overview-analytic-views.html#GUID-F049EEFA-F931-433F-A678-793493DF650E)
* [Create Attribute Dimension Statements](https://docs.oracle.com/en/database/oracle/oracle-database/12.2/dwhsg/overview-analytic-views.html#GUID-7715B749-F84F-459D-90E5-7F052D6B4B2F)
* [Create Hierarchy Statements](https://docs.oracle.com/en/database/oracle/oracle-database/12.2/dwhsg/overview-analytic-views.html#GUID-091832D4-8C48-4B0D-9AAA-E9ECA5BA2896)
* [Create Analytic View Statements](https://docs.oracle.com/en/database/oracle/oracle-database/12.2/dwhsg/overview-analytic-views.html#GUID-2286B197-82A8-4FAA-B347-9366C20893C7)

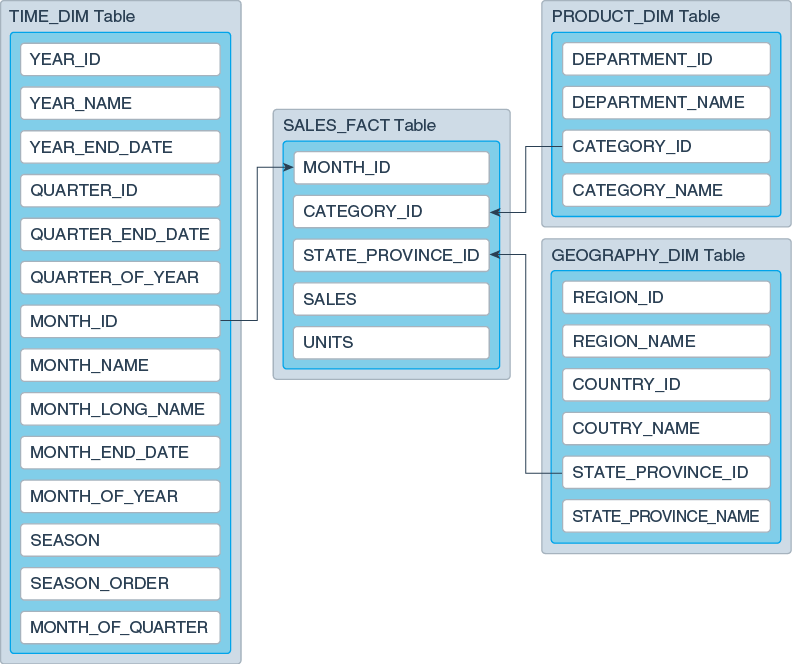
24.9.1 About the Data and Scripts for Examples

The data used by the examples consists of sales data in a single fact table and three dimension tables with time periods, products and geographies.

You can view and run the SQL scripts that create the tables, the analytic view component objects, and the queries used in the examples from the Oracle Live SQL website at <https://livesql.oracle.com/apex/livesql/file/index.html>.

The data is in the star schema tables shown in the following figure.

Figure 24-1 Tables for Analytic View Examples

  
[Description of "Figure 24-1 Tables for Analytic View Examples "](https://docs.oracle.com/en/database/oracle/oracle-database/12.2/dwhsg/img_text/example-tables-analytic-views.html)

In the SALES\_FACT table, the MONTH\_ID, DEPARTMENT\_ID, and STATE\_PROVINCE\_ID columns are foreign keys to the TIME\_DIM, PRODUCT\_DIM, and GEOGRAPHY\_DIM dimension tables, respectively.

In each dimension table, the \_ID columns are used as keys and the \_NAME columns are used as descriptors. Other columns may be used as attributes for sorting or reporting.

There are 1:1 relationships in data between \_ID and \_NAME columns. You can sort time periods by using the \_END\_DATE columns of the TIME\_DIM table.

24.9.2 Create Attribute Dimension Statements

This topic contains SQL statements that create the example attribute dimensions.

Create the time\_attr\_dim Attribute Dimension

The time\_attr\_dim attribute dimension is based on the TIME\_DIM dimension table. The following statement creates the attribute dimension.

CopyCREATE OR REPLACE ATTRIBUTE DIMENSION time\_attr\_dim

DIMENSION TYPE TIME

USING time\_dim

ATTRIBUTES

(year\_id

CLASSIFICATION caption VALUE 'YEAR\_ID'

CLASSIFICATION description VALUE 'YEAR ID',

year\_name

CLASSIFICATION caption VALUE 'YEAR\_NAME'

CLASSIFICATION description VALUE 'Year',

year\_end\_date

CLASSIFICATION caption VALUE 'YEAR\_END\_DATE'

CLASSIFICATION description VALUE 'Year End Date',

quarter\_id

CLASSIFICATION caption VALUE 'QUARTER\_ID'

CLASSIFICATION description VALUE 'QUARTER ID',

quarter\_name

CLASSIFICATION caption VALUE 'QUARTER\_NAME'

CLASSIFICATION description VALUE 'Quarter',

quarter\_end\_date

CLASSIFICATION caption VALUE 'QUARTER\_END\_DATE'

CLASSIFICATION description VALUE 'Quarter End Date',

quarter\_of\_year

CLASSIFICATION caption VALUE 'QUARTER\_OF\_YEAR'

CLASSIFICATION description VALUE 'Quarter of Year',

month\_id

CLASSIFICATION caption VALUE 'MONTH\_ID'

CLASSIFICATION description VALUE 'MONTH ID',

month\_name

CLASSIFICATION caption VALUE 'MONTH\_NAME'

CLASSIFICATION description VALUE 'Month',

month\_long\_name

CLASSIFICATION caption VALUE 'MONTH\_LONG\_NAME'

CLASSIFICATION description VALUE 'Month Long Name',

month\_end\_date

CLASSIFICATION caption VALUE 'MONTH\_END\_DATE'

CLASSIFICATION description VALUE 'Month End Date',

month\_of\_quarter

CLASSIFICATION caption VALUE 'MONTH\_OF\_QUARTER'

CLASSIFICATION description VALUE 'Month of Quarter',

month\_of\_year

CLASSIFICATION caption VALUE 'MONTH\_OF\_YEAR'

CLASSIFICATION description VALUE 'Month of Year',

season

CLASSIFICATION caption VALUE 'SEASON'

CLASSIFICATION description VALUE 'Season',

season\_order

CLASSIFICATION caption VALUE 'SEASON\_ORDER'

CLASSIFICATION description VALUE 'Season Order')

LEVEL month

LEVEL TYPE MONTHS

CLASSIFICATION caption VALUE 'MONTH'

CLASSIFICATION description VALUE 'Month'

KEY month\_id

MEMBER NAME month\_name

MEMBER CAPTION month\_name

MEMBER DESCRIPTION month\_long\_name

ORDER BY month\_end\_date

DETERMINES (month\_end\_date,

quarter\_id,

season,

season\_order,

month\_of\_year,

month\_of\_quarter)

LEVEL quarter

LEVEL TYPE QUARTERS

CLASSIFICATION caption VALUE 'QUARTER'

CLASSIFICATION description VALUE 'Quarter'

KEY quarter\_id

MEMBER NAME quarter\_name

MEMBER CAPTION quarter\_name

MEMBER DESCRIPTION quarter\_name

ORDER BY quarter\_end\_date

DETERMINES (quarter\_end\_date,

quarter\_of\_year,

year\_id)

LEVEL year

LEVEL TYPE YEARS

CLASSIFICATION caption VALUE 'YEAR'

CLASSIFICATION description VALUE 'Year'

KEY year\_id

MEMBER NAME year\_name

MEMBER CAPTION year\_name

MEMBER DESCRIPTION year\_name

ORDER BY year\_end\_date

DETERMINES (year\_end\_date)

LEVEL season

LEVEL TYPE QUARTERS

CLASSIFICATION caption VALUE 'SEASON'

CLASSIFICATION description VALUE 'Season'

KEY season

MEMBER NAME season

MEMBER CAPTION season

MEMBER DESCRIPTION season

LEVEL month\_of\_quarter

LEVEL TYPE MONTHS

CLASSIFICATION caption VALUE 'MONTH\_OF\_QUARTER'

CLASSIFICATION description VALUE 'Month of Quarter'

KEY month\_of\_quarter;

Create the product\_attr\_dim Attribute Dimension

The product\_attr\_dim attribute dimension is based on the PRODUCT\_DIM dimension table. The following statement creates the attribute dimension.

CopyCREATE OR REPLACE ATTRIBUTE DIMENSION product\_attr\_dim

USING product\_dim

ATTRIBUTES

(department\_id

CLASSIFICATION caption VALUE 'DEPARTMENT\_ID'

CLASSIFICATION description VALUE 'DEPARTMENT ID',

department\_name

CLASSIFICATION caption VALUE 'DEPARTMENT\_NAME'

CLASSIFICATION description VALUE 'Department',

category\_id

CLASSIFICATION caption VALUE 'CATEGORY\_ID'

CLASSIFICATION description VALUE 'CATEGORY ID',

category\_name

CLASSIFICATION caption VALUE 'CATEGORY\_NAME'

CLASSIFICATION description VALUE 'Category')

LEVEL DEPARTMENT

CLASSIFICATION caption VALUE 'DEPARTMENT'

CLASSIFICATION description VALUE 'Department'

KEY department\_id

MEMBER NAME department\_name

MEMBER CAPTION department\_name

ORDER BY department\_name

LEVEL CATEGORY

CLASSIFICATION caption VALUE 'CATEGORY'

CLASSIFICATION description VALUE 'Category'

KEY category\_id

MEMBER NAME category\_name

MEMBER CAPTION category\_name

ORDER BY category\_name

DETERMINES(department\_id)

ALL MEMBER NAME 'ALL PRODUCTS';

Create the geography\_attr\_dim Attribute Dimension

The geography\_attr\_dim attribute dimension is based on the GEOGRAPHY\_DIM dimension table. The following statement creates the attribute dimension.

CopyCREATE OR REPLACE ATTRIBUTE DIMENSION geography\_attr\_dim

USING geography\_dim

ATTRIBUTES

(region\_id

CLASSIFICATION caption VALUE 'REGION\_ID'

CLASSIFICATION description VALUE 'REGION ID',

region\_name

CLASSIFICATION caption VALUE 'REGION\_NAME'

CLASSIFICATION description VALUE 'Region',

country\_id

CLASSIFICATION caption VALUE 'COUNTRY\_ID'

CLASSIFICATION description VALUE 'COUNTRY ID',

country\_name

CLASSIFICATION caption VALUE 'COUNTRY\_NAME'

CLASSIFICATION description VALUE 'Country',

state\_province\_id

CLASSIFICATION caption VALUE 'STATE\_PROVINCE\_ID'

CLASSIFICATION description VALUE 'STATE-PROVINCE ID',

state\_province\_name

CLASSIFICATION caption VALUE 'STATE\_PROVINCE\_NAME'

CLASSIFICATION description VALUE 'State-Province')

LEVEL REGION

CLASSIFICATION caption VALUE 'REGION'

CLASSIFICATION description VALUE 'Region'

KEY region\_id

MEMBER NAME region\_name

MEMBER CAPTION region\_name

ORDER BY region\_name

LEVEL COUNTRY

CLASSIFICATION caption VALUE 'COUNTRY'

CLASSIFICATION description VALUE 'Country'

KEY country\_id

MEMBER NAME country\_name

MEMBER CAPTION country\_name

ORDER BY country\_name

DETERMINES(region\_id)

LEVEL STATE\_PROVINCE

CLASSIFICATION caption VALUE 'STATE\_PROVINCE'

CLASSIFICATION description VALUE 'State-Province'

KEY state\_province\_id

MEMBER NAME state\_province\_name

MEMBER CAPTION state\_province\_name

ORDER BY state\_province\_name

DETERMINES(country\_id)

ALL MEMBER NAME 'ALL CUSTOMERS';

24.9.3 Create Hierarchy Statements

This topic contains SQL statements that create the example hierarchies.

Create Hierarchies Using time\_attr\_dim

The following statements create hierarchies that use the time\_attr\_dim attribute dimension.

CopyCREATE OR REPLACE HIERARCHY time\_hier

CLASSIFICATION caption VALUE 'CALENDAR'

CLASSIFICATION description VALUE 'CALENDAR'

USING time\_attr\_dim

(month CHILD OF

quarter CHILD OF

year);

--

CREATE OR REPLACE HIERARCHY time\_season\_hier

CLASSIFICATION caption VALUE 'SEASONS'

CLASSIFICATION description VALUE 'Seasons'

USING time\_attr\_dim

(month CHILD OF

season);

--

CREATE OR REPLACE HIERARCHY time\_year\_season\_hier

USING time\_attr\_dim

(month CHILD OF

season CHILD OF

year);

--

CREATE OR REPLACE HIERARCHY time\_month\_of\_qtr\_hier

CLASSIFICATION caption VALUE 'MONTH\_OF\_QUARTER'

CLASSIFICATION description VALUE 'Month of Quarter'

USING time\_attr\_dim

(month CHILD OF

month\_of\_quarter);

Create a Hierarchy Using product\_attr\_dim

The following statement creates a hierarchy that uses the product\_attr\_dim attribute dimension.

CopyCREATE OR REPLACE HIERARCHY product\_hier

CLASSIFICATION caption VALUE 'PRODUCT'

CLASSIFICATION description VALUE 'Product'

USING product\_attr\_dim

(CATEGORY

CHILD OF department);

Create a Hierarchy Using geography\_attr\_dim

The following statement creates a hierarchy that uses the geography\_attr\_dim attribute dimension.

CopyCREATE OR REPLACE HIERARCHY geography\_hier

CLASSIFICATION caption VALUE 'GEOGRAPHY'

CLASSIFICATION description VALUE 'Geography'

USING geography\_attr\_dim

(state\_province

CHILD OF country

CHILD OF region);

24.9.4 Create Analytic View Statements

This topic contains a SQL statement that creates the example analytic view.

Create the sales\_av Analytic View

The following statement creates an analytic view that uses the SALES\_FACT fact table.

CopyCREATE OR REPLACE ANALYTIC VIEW sales\_av

CLASSIFICATION caption VALUE 'Sales AV'

CLASSIFICATION description VALUE 'Sales Analytic View'

CLASSIFICATION created\_by VALUE 'Harold C. Ehrlich'

USING sales\_fact

DIMENSION BY

(time\_attr\_dim

KEY month\_id REFERENCES month\_id

HIERARCHIES (

time\_hier DEFAULT,

time\_season\_hier,

time\_year\_season\_hier,

time\_month\_of\_qtr\_hier),

product\_attr\_dim

KEY category\_id REFERENCES category\_id

HIERARCHIES (

product\_hier DEFAULT),

geography\_attr\_dim

KEY state\_province\_id

REFERENCES state\_province\_id

HIERARCHIES (

geography\_hier DEFAULT)

)

MEASURES

(sales FACT sales

CLASSIFICATION caption VALUE 'Sales'

CLASSIFICATION description VALUE 'Sales'

CLASSIFICATION format\_string VALUE '$9,999.99',

units FACT units

CLASSIFICATION caption VALUE 'Units'

CLASSIFICATION description VALUE 'Units Sold'

CLASSIFICATION format\_string VALUE '9,999',

sales\_prior\_period AS

(LAG(SALES) OVER (HIERARCHY time\_hier OFFSET 1))

CLASSIFICATION caption VALUE 'Sales Prior Period'

CLASSIFICATION description VALUE 'Sales Prior\_Period'

CLASSIFICATION format\_string VALUE '$9,999.99',

sales\_chg\_prior\_period AS

(LAG\_DIFF(SALES) OVER (HIERARCHY time\_hier OFFSET 1))

CLASSIFICATION caption VALUE 'Sales Change Prior Period'

CLASSIFICATION description VALUE 'Sales Change Prior Period'

CLASSIFICATION format\_string VALUE '$9,999.99',

sales\_qtr\_ago AS

(LAG(SALES) OVER (HIERARCHY time\_hier OFFSET 1

ACROSS ANCESTOR AT LEVEL quarter))

CLASSIFICATION caption VALUE 'Sales Qtr Ago'

CLASSIFICATION description VALUE 'Sales Qtr Ago'

CLASSIFICATION format\_string VALUE '$9,999.99',

sales\_chg\_qtr\_ago AS

(LAG\_DIFF(SALES) OVER (HIERARCHY time\_hier OFFSET 1

ACROSS ANCESTOR AT LEVEL quarter))

CLASSIFICATION caption VALUE 'Sales Change Qtr Ago'

CLASSIFICATION description VALUE 'Sales Change Qtr Ago'

CLASSIFICATION format\_string VALUE '$9,999.99',

sales\_pct\_chg\_qtr\_ago AS

(LAG\_DIFF\_PERCENT(SALES) OVER (HIERARCHY time\_hier OFFSET 1

ACROSS ANCESTOR AT LEVEL quarter))

CLASSIFICATION caption VALUE 'Sales Percent Change Qtr Ago'

CLASSIFICATION description VALUE 'Sales Percent Change Qtr Ago'

CLASSIFICATION format\_string VALUE '999.99',

sales\_yr\_ago AS

(LAG(SALES) OVER (HIERARCHY time\_hier OFFSET 1

ACROSS ANCESTOR AT LEVEL year))

CLASSIFICATION caption VALUE 'Sales Year Ago'

CLASSIFICATION description VALUE 'Sales Year Ago'

CLASSIFICATION format\_string VALUE '$9,999.99',

sales\_chg\_yr\_ago AS

(LAG\_DIFF(SALES) OVER (HIERARCHY time\_hier OFFSET 1

ACROSS ANCESTOR AT LEVEL year))

CLASSIFICATION caption VALUE 'Sales Change Year Ago'

CLASSIFICATION description VALUE 'Sales Change Year Ago'

CLASSIFICATION format\_string VALUE '$9,999.99',

sales\_pct\_chg\_yr\_ago AS

(LAG\_DIFF\_PERCENT(SALES) OVER (HIERARCHY time\_hier OFFSET 1

ACROSS ANCESTOR AT LEVEL year))

CLASSIFICATION caption VALUE 'Sales Percent Change Year Ago'

CLASSIFICATION description VALUE 'Sales Percent Change Year Ago'

CLASSIFICATION format\_string VALUE '999.99',

sales\_qtd AS

(SUM(sales) OVER (HIERARCHY time\_hier

BETWEEN UNBOUNDED PRECEDING AND CURRENT MEMBER

WITHIN ANCESTOR AT LEVEL quarter))

CLASSIFICATION caption VALUE 'Sales Quarter to Date'

CLASSIFICATION description VALUE 'Sales Quarter to Date'

CLASSIFICATION format\_string VALUE '$9,999.99',

sales\_ytd AS

(SUM(sales) OVER (HIERARCHY time\_hier

BETWEEN UNBOUNDED PRECEDING AND CURRENT MEMBER

WITHIN ANCESTOR AT LEVEL year))

CLASSIFICATION caption VALUE 'Sales Year to Date'

CLASSIFICATION description VALUE 'Sales Year to Date'

CLASSIFICATION format\_string VALUE '$9,999.99',

sales\_2011 AS

(QUALIFY (sales, time\_hier = year['11']))

CLASSIFICATION caption VALUE 'Sales CY2011'

CLASSIFICATION description VALUE 'Sales CY2011'

CLASSIFICATION format\_string VALUE '$9,999.99',

sales\_pct\_chg\_2011 AS

((sales - (QUALIFY (sales, time\_hier = year['11']))) /

(QUALIFY (sales, time\_hier = year['11'])))

CLASSIFICATION caption VALUE 'Sales Pct Change CY2011'

CLASSIFICATION description VALUE 'Sales Pct Change CY2011'

CLASSIFICATION format\_string VALUE '999.99',

sales\_share\_time\_parent AS

(SHARE\_OF(sales HIERARCHY time\_hier PARENT))

CLASSIFICATION caption VALUE 'Sales Share of Time Parent'

CLASSIFICATION description VALUE 'Sales Share of Time Parent'

CLASSIFICATION format\_string VALUE '999.99',

sales\_share\_season\_parent AS

(SHARE\_OF(sales HIERARCHY time\_season\_hier PARENT))

CLASSIFICATION caption VALUE 'Sales Share of Season Parent'

CLASSIFICATION description VALUE 'Sales Share of Season Parent'

CLASSIFICATION format\_string VALUE '999.99',

sales\_share\_prod\_parent AS

(SHARE\_OF(sales HIERARCHY product\_hier PARENT))

CLASSIFICATION caption VALUE 'Sales Share of Product Parent'

CLASSIFICATION description VALUE 'Sales Share of Product Parent'

CLASSIFICATION format\_string VALUE '999.99',

sales\_share\_dept AS

(SHARE\_OF(sales HIERARCHY product\_hier LEVEL department))

CLASSIFICATION caption VALUE 'Sales Share of Product Parent'

CLASSIFICATION description VALUE 'Sales Share of Product Parent'

CLASSIFICATION format\_string VALUE '999.99',

sales\_share\_geog\_parent AS

(SHARE\_OF(sales HIERARCHY geography\_hier PARENT))

CLASSIFICATION caption VALUE 'Sales Share of Geography Parent'

CLASSIFICATION description VALUE 'Sales Share of Geography Parent'

CLASSIFICATION format\_string VALUE '999.99',

sales\_share\_region AS

(SHARE\_OF(sales HIERARCHY geography\_hier LEVEL region))

CLASSIFICATION caption VALUE 'Sales Share of Geography Parent'

CLASSIFICATION description VALUE 'Sales Share of Geography Parent'

CLASSIFICATION format\_string VALUE '999.99'

)

DEFAULT MEASURE SALES;