Oracle 12.2 Analytic Views:

Worth A Look

Karen Cannell
kcannell@thtechnology.com
TH TECHNOLOGY

About Me ...

- **Mechanical/SW Engineer** - Analyzed, designed, developed, converted, upgraded, enhanced legacy & database applications for 30+ **years**
- **Building Web/APEX applications for government, medical, engineering industries since HTMLDB beginnings**
- **Leveraging Oracle 10g,11g,12c suite of tools**
- **Editor Emeritus, ODTUG Technical Journal**
- **Oracle Ace**
- **APress Author**
- **ODTUG Director**

IOUG

Expert Oracle Application Express

Apress

About You …

- Oracle Technology ?
- OLAP Users?
- APEX Users ?
- Why Are You Here?

Analytics …

Agenda

- What
- How … to Build
- How … to Query
- Who ~ Why ~ When ~Where to Use
- Wrap-up
  - Questions ~ Discussion

Analytic Views:

What ?

Analytic Views

- Oracle 12.2
  - No Cost Feature
- Like a View
  - Metadata (Do Not Store Data)
  - Query via SQL
  - Access Data from Other Objects
  - Join Multiple Tables

Analytical Views

Better than a View:

- Organize – “Model” - Data into Dimensions, Hierarchies
- Automatically Aggregates
- Include Embedded Calculations (MEASURES)
- Include Presentation Metadata

Analytic Views

- “Hierarchical / Dimensional Model Over the Dimension and Fact Tables of a Star Schema …”
- Best Performance
  - Star Schema, In Memory
  - W MViews, MViews In Memory

Analytic Views

- Simplify SQL for Analytic Queries
  - No Joins, No GROUP BYs
- Define Calculations w/in Analytic View
  - Aggregates ~ Calculations ~ Ranks ~ Forecasts
- Reuse Calculations from the Analytic View

Make Dimensional, Hierarchical Analyses More Accessible

➔➔➔ Simpler, Faster Development

Analytic Views

- “Move business logic into the DB …”
- Transform Data into a ″Business Model and Presentation Layer″
- Organize Data for Common Analysis Queries
  - Year/Month/Day, Country, Region, State
  - Simple and Fast.

– A mere mortal can do it!

Analytic Views:

How to Build

Attr Dimensions
Hierarchies
Analytic Views

GLOC
Great Lakes Oracle Conference

System Privileges

- CREATE ANALYTIC VIEW
- CREATE ANY ANALYTIC VIEW
- ALTER ANY ANALYTIC VIEW
- DROP ANY ANALYTIC VIEW

- CREATE ATTRIBUTE DIMENSION
- CREATE ANY ATTRIBUTE DIMENSION
- ALTER ANY ATTRIBUTE DIMENSION
- DROP ANY ATTRIBUTE DIMENSION

- CREATE HIERARCHY
- CREATE ANY HIERARCHY
- ALTER ANY HIERARCHY
- DROP ANY HIERARCHY

Object Privileges

- SELECT - Query
- READ - Query
- ALTER - Rename

Ex:

```
GRANT ALL ON AV_USER.SALES_AV TO AV_USER2;

GRANT ALTER ON AV_USER.SALES_AV TO AV_USER3;
```

Diagram here

- Dimensions

Hierarchies of Dimension

Analytic View that contains Measures based on the Hierarchies

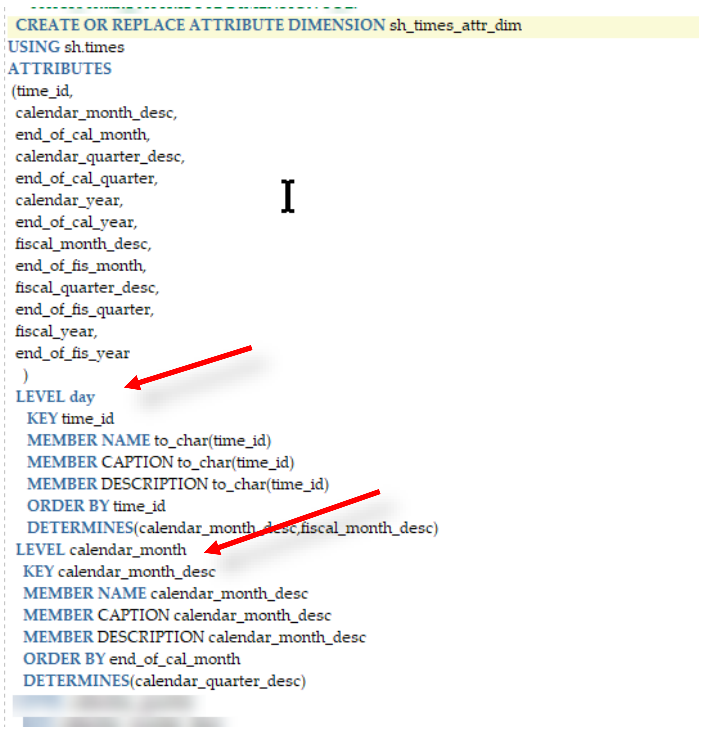
– Faster Access to common (Defined) analytical agfregations, and

– Simpler query construction (c/o the complexity is on the AV where the calcs are defined)

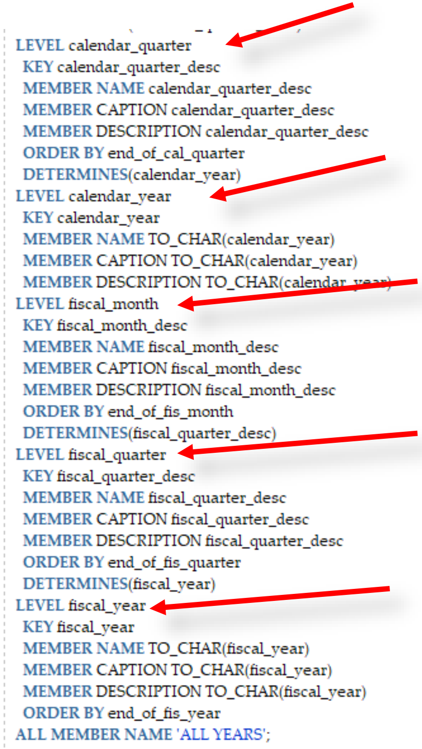
Attribute Dimensions

- SH Time
- Base calendar yr query
- Base fiscal yr query
- These will be combined into one attr dimension

Sample Time Attr Dimension

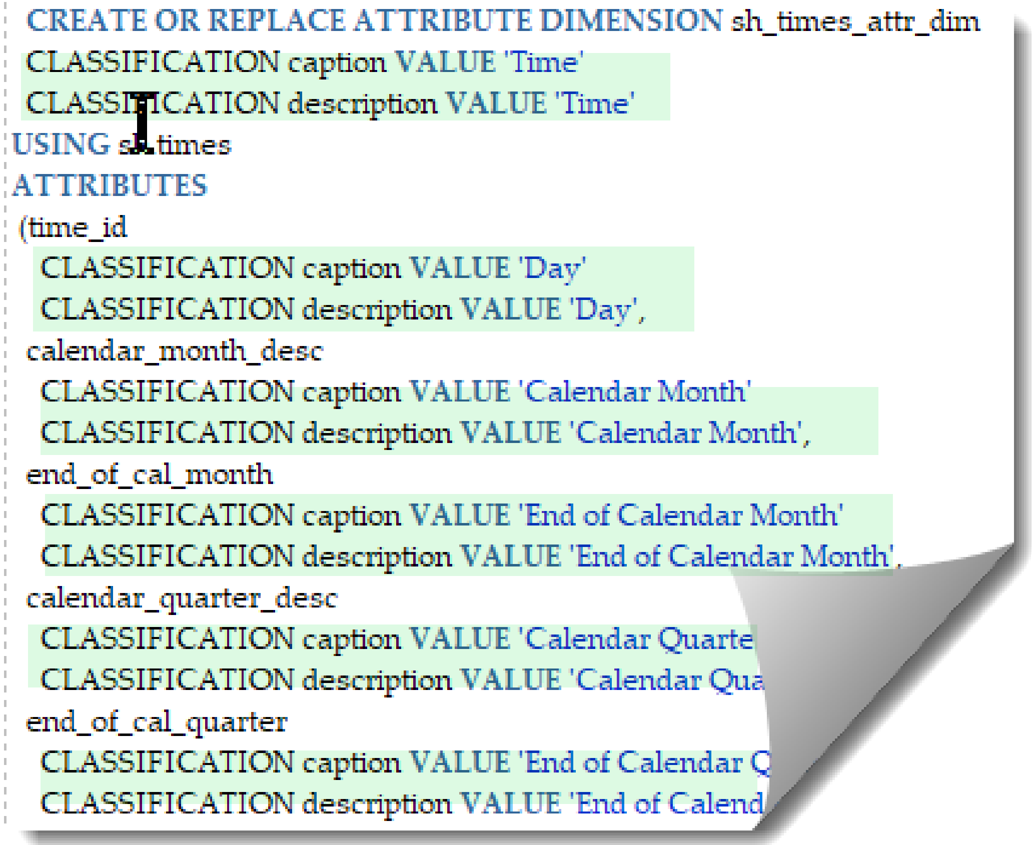


```
CREATE OR REPLACE ATTRIBUTE DIMENSION sh_times_attr_dim
USING sh.times
ATTRIBUTES
(time_id,
 calendar_month_desc,
 end_of_cal_month,
 calendar_quarter_desc,
 end_of_cal_quarter,
 calendar_year,
 end_of_cal_year,
 fiscal_month_desc,
 end_of_fis_month,
 fiscal_quarter_desc,
 end_of_fis_quarter,
 fiscal_year,
 end_of_fis_year
  )
 LEVEL day
   KEY time_id
   MEMBER NAME to_char(time_id)
   MEMBER CAPTION to_char(time_id)
   MEMBER DESCRIPTION to_char(time_id)
   ORDER BY time_id
   DETERMINES(calendar_month_desc,fiscal_month_desc)
 LEVEL calendar_month
  KEY calendar_month_desc
  MEMBER NAME calendar_month_desc
  MEMBER CAPTION calendar_month_desc
  MEMBER DESCRIPTION calendar_month_desc
  ORDER BY end_of_cal_month
  DETERMINES(calendar_quarter_desc)
```



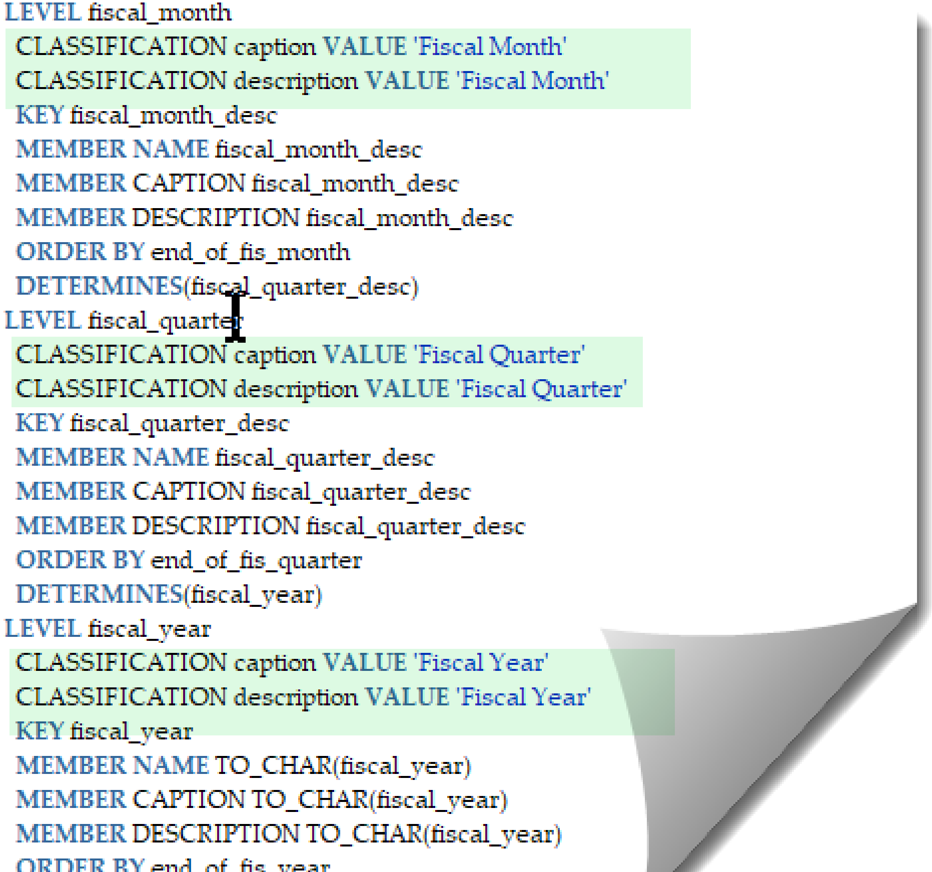
```
LEVEL calendar_quarter
 KEY calendar_quarter_desc
 MEMBER NAME calendar_quarter_desc
 MEMBER CAPTION calendar_quarter_desc
 MEMBER DESCRIPTION calendar_quarter_desc
 ORDER BY end_of_cal_quarter
 DETERMINES(calendar_year)
LEVEL calendar_year
 KEY calendar_year
 MEMBER NAME TO_CHAR(calendar_year)
 MEMBER CAPTION TO_CHAR(calendar_year)
 MEMBER DESCRIPTION TO_CHAR(calendar_year)
LEVEL fiscal_month
 KEY fiscal_month_desc
 MEMBER NAME fiscal_month_desc
 MEMBER CAPTION fiscal_month_desc
 MEMBER DESCRIPTION fiscal_month_desc
 ORDER BY end_of_fis_month
 DETERMINES(fiscal_quarter_desc)
LEVEL fiscal_quarter
 KEY fiscal_quarter_desc
 MEMBER NAME fiscal_quarter_desc
 MEMBER CAPTION fiscal_quarter_desc
 MEMBER DESCRIPTION fiscal_quarter_desc
 ORDER BY end_of_fis_quarter
 DETERMINES(fiscal_year)
LEVEL fiscal_year
 KEY fiscal_year
 MEMBER NAME TO_CHAR(fiscal_year)
 MEMBER CAPTION TO_CHAR(fiscal_year)
 MEMBER DESCRIPTION TO_CHAR(fiscal_year)
 ORDER BY end_of_fis_year
ALL MEMBER NAME 'ALL YEARS';
```

Now Add CLASSIFICATION



```
 CREATE OR REPLACE ATTRIBUTE DIMENSION sh_times_attr_dim
 CLASSIFICATION caption VALUE 'Time'
 CLASSIFICATION description VALUE 'Time'
USING sh_times
ATTRIBUTES
(time_id
  CLASSIFICATION caption VALUE 'Day'
  CLASSIFICATION description VALUE 'Day',
 calendar_month_desc
  CLASSIFICATION caption VALUE 'Calendar Month'
  CLASSIFICATION description VALUE 'Calendar Month',
 end_of_cal_month
  CLASSIFICATION caption VALUE 'End of Calendar Month'
  CLASSIFICATION description VALUE 'End of Calendar Month',
 calendar_quarter_desc
  CLASSIFICATION caption VALUE 'Calendar Quarte
  CLASSIFICATION description VALUE 'Calendar Qua
 end_of_cal_quarter
  CLASSIFICATION caption VALUE 'End of Calendar Q
  CLASSIFICATION description VALUE 'End of Calend
```

Now Add CLASSIFICATION …



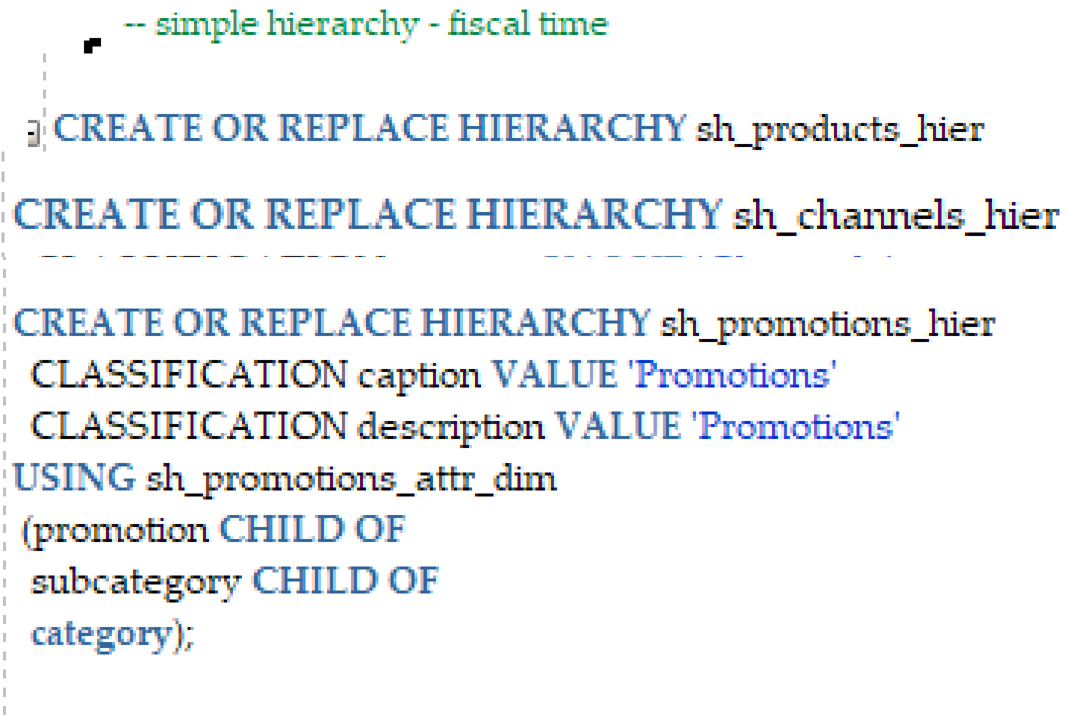
```
LEVEL fiscal_month
 CLASSIFICATION caption VALUE 'Fiscal Month'
 CLASSIFICATION description VALUE 'Fiscal Month'
 KEY fiscal_month_desc
 MEMBER NAME fiscal_month_desc
 MEMBER CAPTION fiscal_month_desc
 MEMBER DESCRIPTION fiscal_month_desc
 ORDER BY end_of_fis_month
 DETERMINES(fiscal_quarter_desc)
LEVEL fiscal_quarter
 CLASSIFICATION caption VALUE 'Fiscal Quarter'
 CLASSIFICATION description VALUE 'Fiscal Quarter'
 KEY fiscal_quarter_desc
 MEMBER NAME fiscal_quarter_desc
 MEMBER CAPTION fiscal_quarter_desc
 MEMBER DESCRIPTION fiscal_quarter_desc
 ORDER BY end_of_fis_quarter
 DETERMINES(fiscal_year)
LEVEL fiscal_year
 CLASSIFICATION caption VALUE 'Fiscal Year'
 CLASSIFICATION description VALUE 'Fiscal Year'
 KEY fiscal_year
 MEMBER NAME TO_CHAR(fiscal_year)
 MEMBER CAPTION TO_CHAR(fiscal_year)
 MEMBER DESCRIPTION TO_CHAR(fiscal_year)
 ORDER BY end_of_fis_year
```

Hierarchies

- Reference an Attribute Dimension
- Attribute Values are queried from Hierarchies …

SH Hierarchies

- Time
- Products
- Customers
- Channels
- Promotion



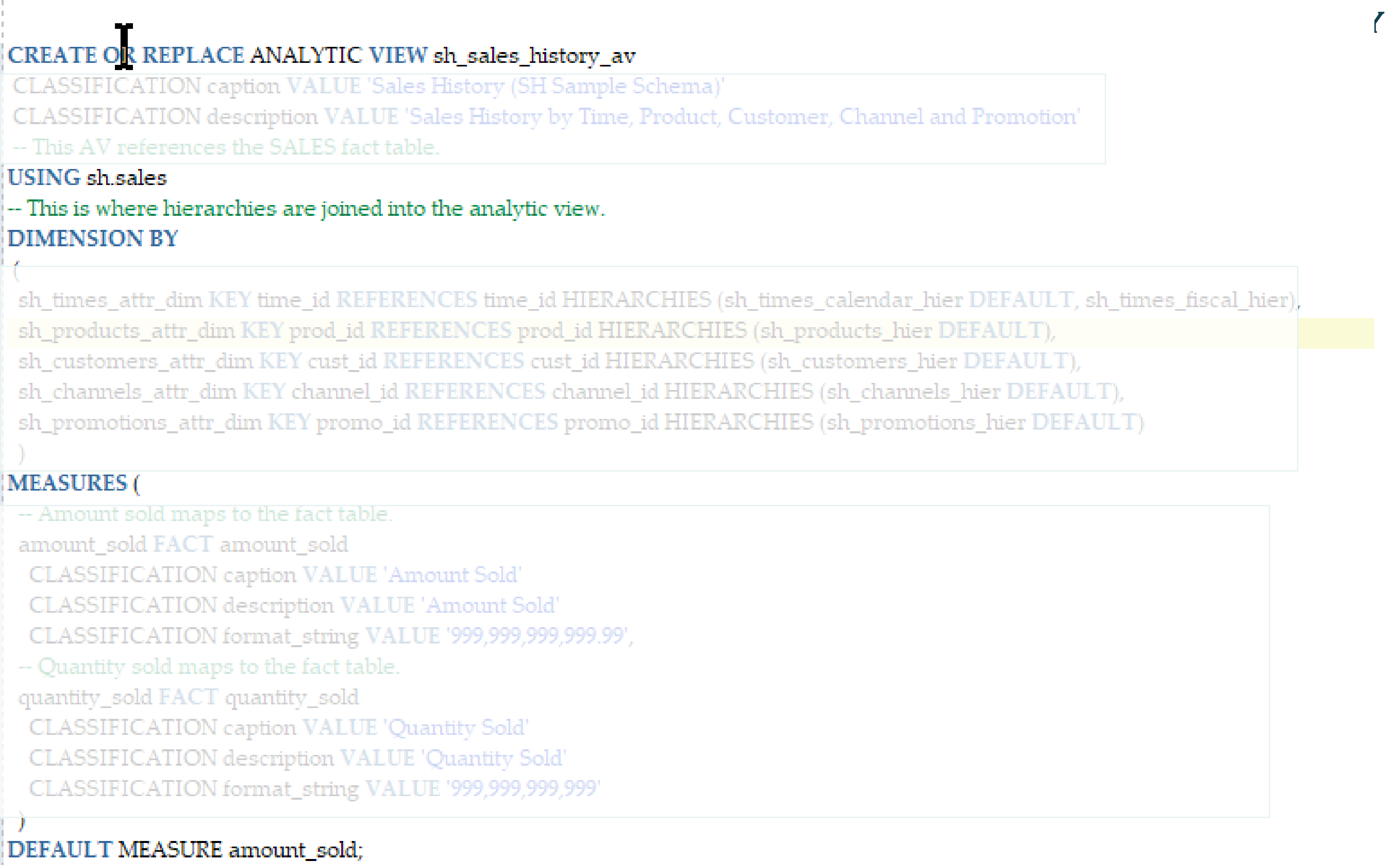
```
-- simple hierarchy - fiscal time

CREATE OR REPLACE HIERARCHY sh_products_hier

CREATE OR REPLACE HIERARCHY sh_channels_hier

CREATE OR REPLACE HIERARCHY sh_promotions_hier
 CLASSIFICATION caption VALUE 'Promotions'
 CLASSIFICATION description VALUE 'Promotions'
USING sh_promotions_attr_dim
 (promotion CHILD OF
 subcategory CHILD OF
 category);
```

Analytic Views

- Layer on Top of Star Schema using Attr Dims and Hierarchies
- Fact Data Included in AV
- Calculations Built Into the AV
- Metadata Built Into the AV

Analytic View DDL Format

```
CREATE OR REPLACE ANALYTIC VIEW <name>
 <classification caption and description>

USING <fact table>

DIMENSION BY ( <list of dimension refs> )

HIERARCHIES ( <list of hierarchy references> )

MEASURES ( <list of measures> )

 DEFAULT MEASURE … ;
```



```
CREATE OR REPLACE ANALYTIC VIEW sh_sales_history_av
 CLASSIFICATION caption VALUE 'Sales History (SH Sample Schema)'
 CLASSIFICATION description VALUE 'Sales History by Time, Product, Customer, Channel and Promotion'
 -- This AV references the SALES fact table.
USING sh.sales
-- This is where hierarchies are joined into the analytic view.
DIMENSION BY
 (
  sh_times_attr_dim KEY time_id REFERENCES time_id HIERARCHIES (sh_times_calendar_hier DEFAULT, sh_times_fiscal_hier),
  sh_products_attr_dim KEY prod_id REFERENCES prod_id HIERARCHIES (sh_products_hier DEFAULT),
  sh_customers_attr_dim KEY cust_id REFERENCES cust_id HIERARCHIES (sh_customers_hier DEFAULT),
  sh_channels_attr_dim KEY channel_id REFERENCES channel_id HIERARCHIES (sh_channels_hier DEFAULT),
  sh_promotions_attr_dim KEY promo_id REFERENCES promo_id HIERARCHIES (sh_promotions_hier DEFAULT)
  )
MEASURES (
  -- Amount sold maps to the fact table.
  amount_sold FACT amount_sold
    CLASSIFICATION caption VALUE 'Amount Sold'
    CLASSIFICATION description VALUE 'Amount Sold'
    CLASSIFICATION format_string VALUE '999,999,999,999.99',
  -- Quantity sold maps to the fact table.
  quantity_sold FACT quantity_sold
    CLASSIFICATION caption VALUE 'Quantity Sold'
    CLASSIFICATION description VALUE 'Quantity Sold'
    CLASSIFICATION format_string VALUE '999,999,999,999'
  )
DEFAULT MEASURE amount_sold;
```

Analytic View – Parts Recap

- USING – fact table – where to start
- DIMENSION BY – What Dimensions Queries on the AV Will Use
- HIERARCHY – Which Hierarchies Queries Will Use
- MEASURES – Calculations: Sum, LEAD, LAG, Combinations – What Calculations Are Needed
- CLASSIFICATION – Metadata / Documentation
  - Usually On AV and Measures

Measures – Time Series

```
LAG 1 … ACROSS ANCESTOR AT LEVEL YEAR
```

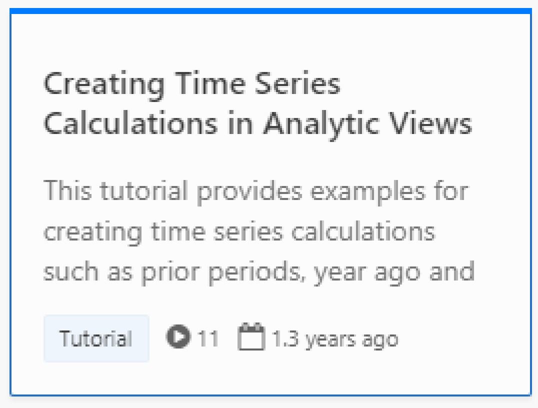
Previous Year

```
AVG(sales) … BETWEEN 11 PRECEDING AND CURRENT MEMBER
```

12 Month Period

See LiveSQL

Creating Time Series Calculations Views



Creating Time Series Calculations in Analytic Views

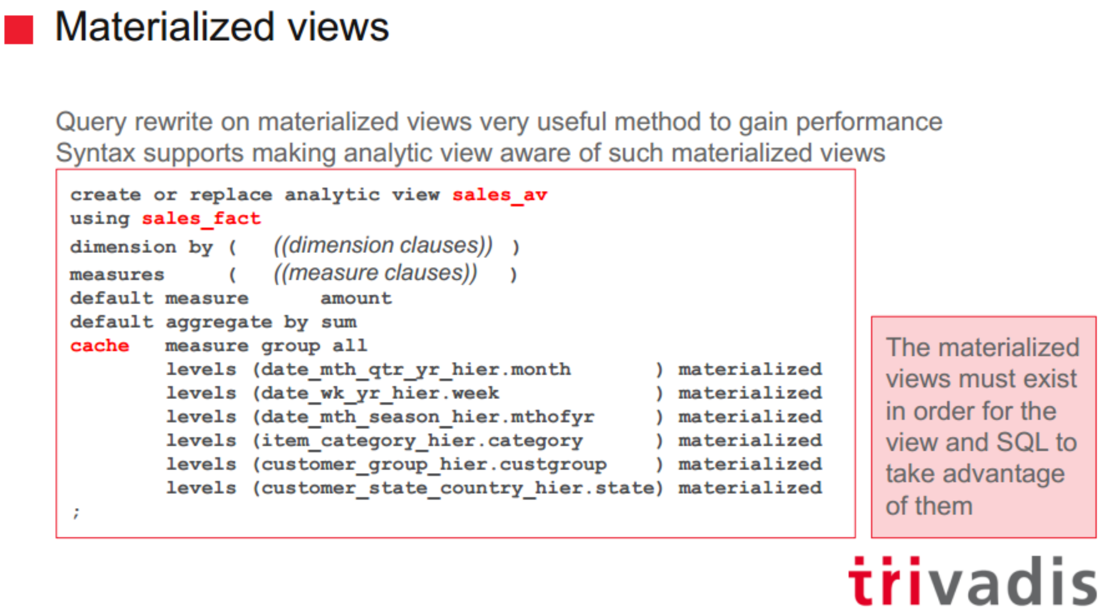
This tutorial provides examples for creating time series calculations such as prior periods, year ago and

CLASSIFICATIONS

- Metadata
  - Dimensions – Hierarchies – Avs – Members – Attributes - Measures
- Caption
- Description
- Format_string
- Shortcut Syntax for caption and description

AVs w Materialized Views

- Query Rewrite on MVs == Performance Gain
- CACHE



Materialized views

Query rewrite on materialized views very useful method to gain performance
Syntax supports making analytic view aware of such materialized views

```
create or replace analytic view sales_av
using sales_fact
dimension by (    ((dimension clauses))  )
measures     (    ((measure clauses))    )
default measure        amount
default aggregate by sum
cache     measure group all
          levels (date_mth_qtr_yr_hier.month        ) materialized
          levels (date_wk_yr_hier.week              ) materialized
          levels (date_mth_season_hier.mthofyr      ) materialized
          levels (item_category_hier.category       ) materialized
          levels (customer_group_hier.custgroup     ) materialized
          levels (customer_state_country_hier.state) materialized
;
```

The materialized views must exist in order for the view and SQL to take advantage of them

Validating Analytic Views

- DBMS_HIERARCHY

Analytic Views:

How to Query

GLOC
Great Lakes Oracle Conference

Queries Against Analytic Views

- Simple
- Add Calculated Measures
- Combine Fact and Calculated Measure

Query Analytic View - Simple

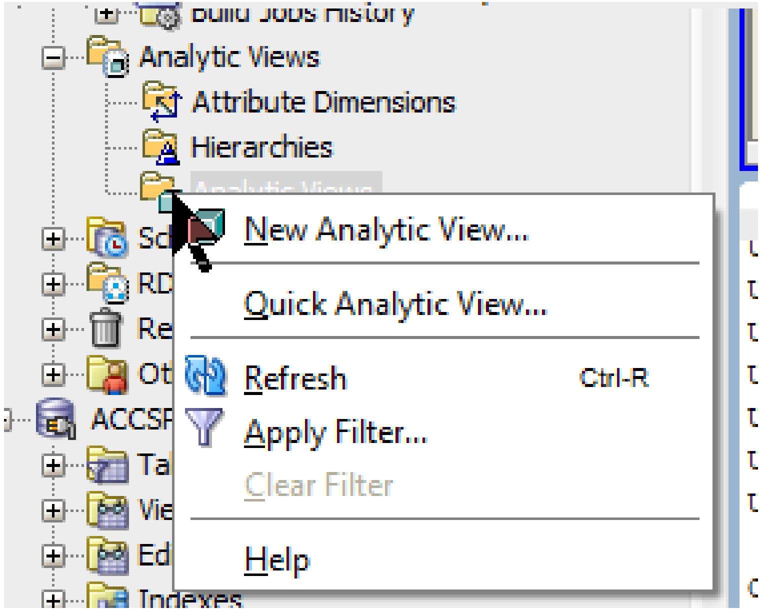
Query AV – + Calculated Measure

Query AV – Fact + Calculated

SQL Developer Quick-Start

- SQL Dev 18.1
- Oracle 12.2

Try It!



Application Containers

"…an application-specific CDB w/in a CDB."

- Can Grant Analytic View Access to Application Container

Analytic Views:

Why ~ When ~ Where

Use Analytic Views For …

- BI Reporting Systems
- Data Visualization
- Data Analysts
- “BI Lite”

Any Dimensional, Hierarchical Queries

BI Lite Demo

Quick Demo – "BI Lite" In APEX

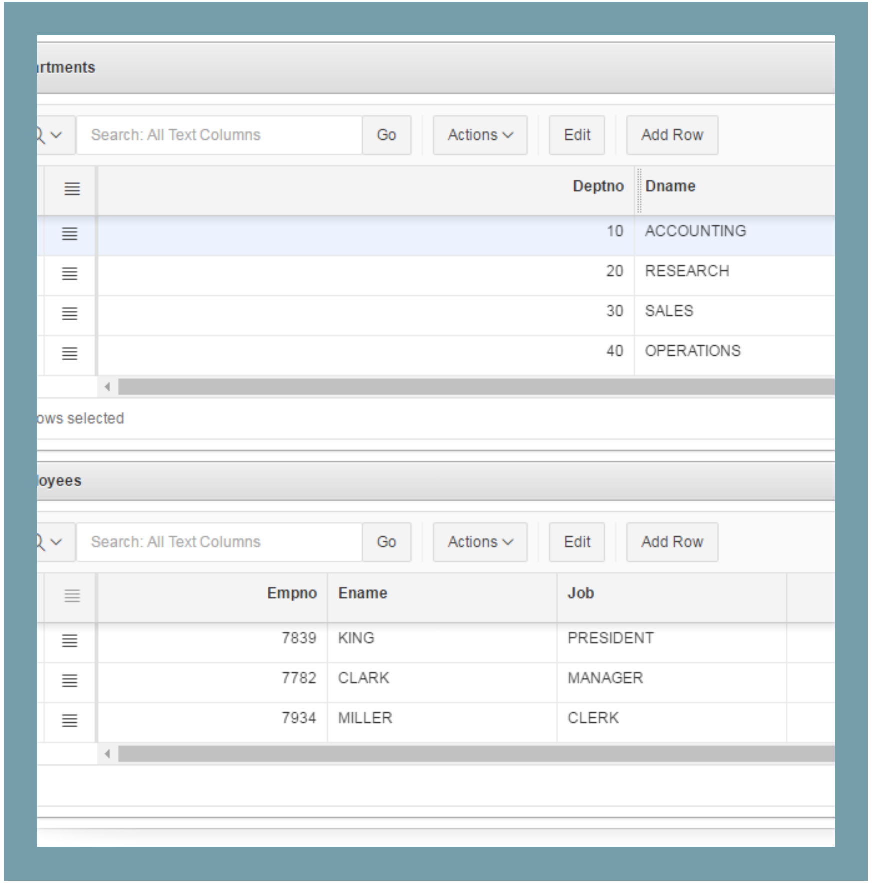
Transitioning to Your Data

- KNOW YOUR DATA
- PLAN
  - Star Schema
  - Dimensions
  - Hierarchies
- Let SQL Dev Help
- Test, Timing, Test, Test

Analytic Views: Wrapup

Questions

Discussion



rtments

Deptno	Dname
10	ACCOUNTING
20	RESEARCH
30	SALES
40	OPERATIONS

ows selected

oyees

Empno	Ename	Job
7839	KING	PRESIDENT
7782	CLARK	MANAGER
7934	MILLER	CLERK

GLOC Great Lakes Oracle Conference

Analytic Views

- Simpler SQL for Analytic Queries
- Best Performance Gain over Star Schema, In Memory
- SQL Dev QuickStart
- Faster Route to Data Viz, “BI-Lite” Implementations

Comments?
Questions?
Thank You

Evaluations Please!

Karen Cannell
kcannell@thtechnology.com

Resources

- Analytic Views LiveSQL Demos

  https://livesql.oracle.com/apex/livesql/file/tutorial_EDVE861IID1QUD1NIUPU5ALEW.html

- Analytic Views in SH Sample Schema

  https://livesql.oracle.com/apex/livesql/file/tutorial_EDVE861IJ168OTUP6ZZ01MX84.html

- Analytic View Support in SQL Developer

  https://www.thatjeffsmith.com/archive/2017/03/oracle-database-12c-release-2-analytic-views-sql-developer/

- Using Analytic Views

  http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/12c/r2/AnalyticViews/UsingAVs.html#overview

GLOC
Great Lakes Oracle Conference

Thank You

Evaluations Please!

Karen Cannell
kcannell@thtechnology.com