CELEBRATING EXCELLENCE
vlamis 25 YEARS
SOFTWARE SOLUTIONS

Business Analytics in the Oracle 12.2 Database: Analytic Views

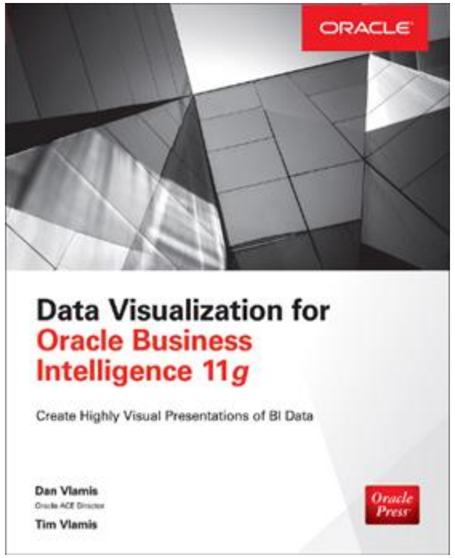
Event: BIWA 2017

Presenter: Dan Vlamis and Cathye Pendley

Date: January 31, 2017

Vlamis Software Solutions

- Vlamis Software founded in 1992 in Kansas City, Missouri
- Developed 200+ Oracle BI and analytics systems
- Specializes in Oracle-based:
  - Enterprise Business Intelligence
  - Data Warehousing
  - Data Mining and Predictive Analytics
  - Data Visualization
- Multiple Oracle ACEs, consultants average 15+ years
- www.vlamis.com (blog, papers, newsletters, services)
- Co-authors of book “Data Visualization for OBI 11g”
- Co-author of book “Oracle Essbase & Oracle OLAP”
- Oracle University Partner
- Oracle Gold Partner



ORACLE

Data Visualization for Oracle Business Intelligence 11*g*

Create Highly Visual Presentations of BI Data

Dan Vlamis
Oracle ACE Director
Tim Vlamis

Oracle Press

ORACLE

Oracle Essbase & Oracle OLAP

ORACLE EDUCATION RESELLER

ORACLE APPROVED EDUCATION CENTER

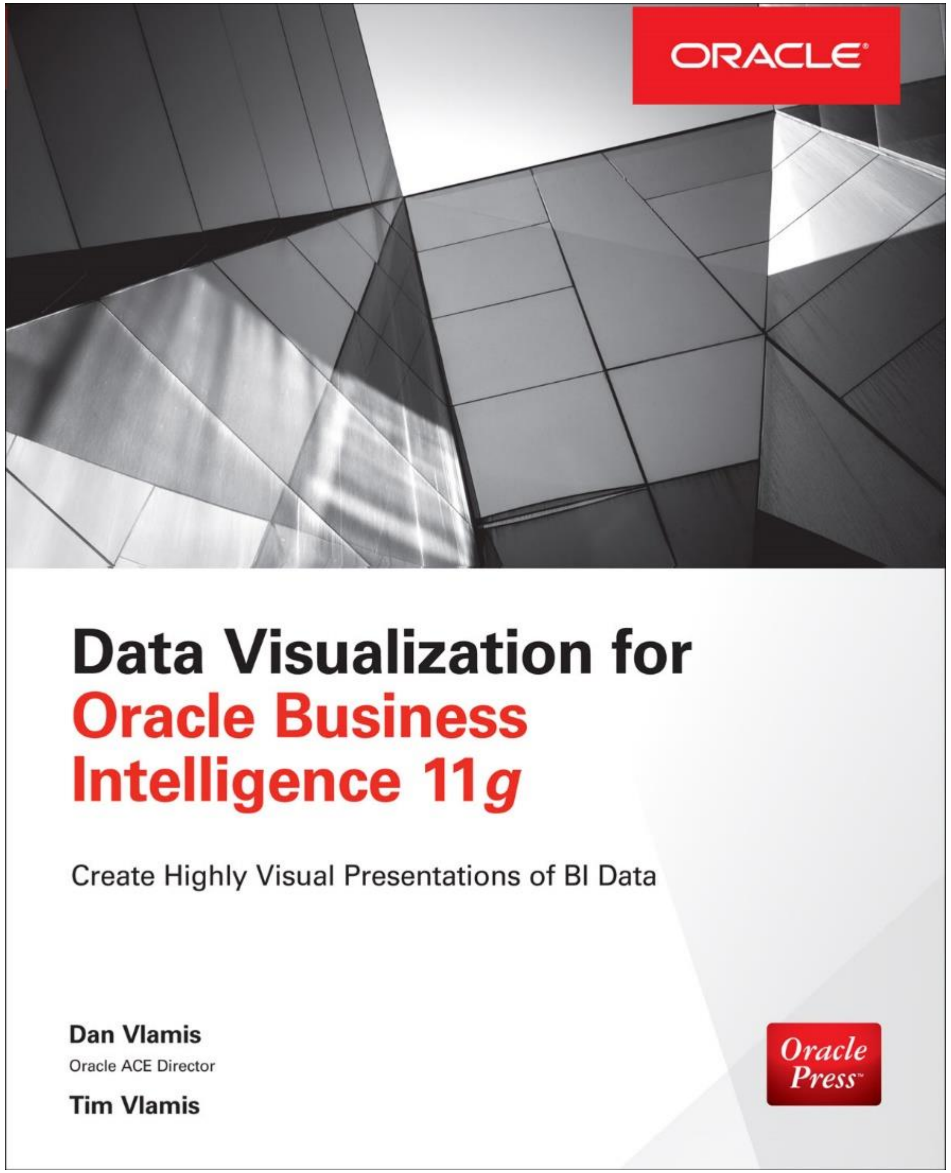
ORACLE Gold Partner

Specialized
Oracle Business Intelligence Foundation Suite 11*g*

CELEBRATING EXCELLENCE
vlamis 25 YEARS
SOFTWARE SOLUTIONS

Drawing for Free Book

Add business card to basket or fill out card



ORACLE

Data Visualization for Oracle Business Intelligence 11*g*

Create Highly Visual Presentations of BI Data

Dan Vlamis
Oracle ACE Director

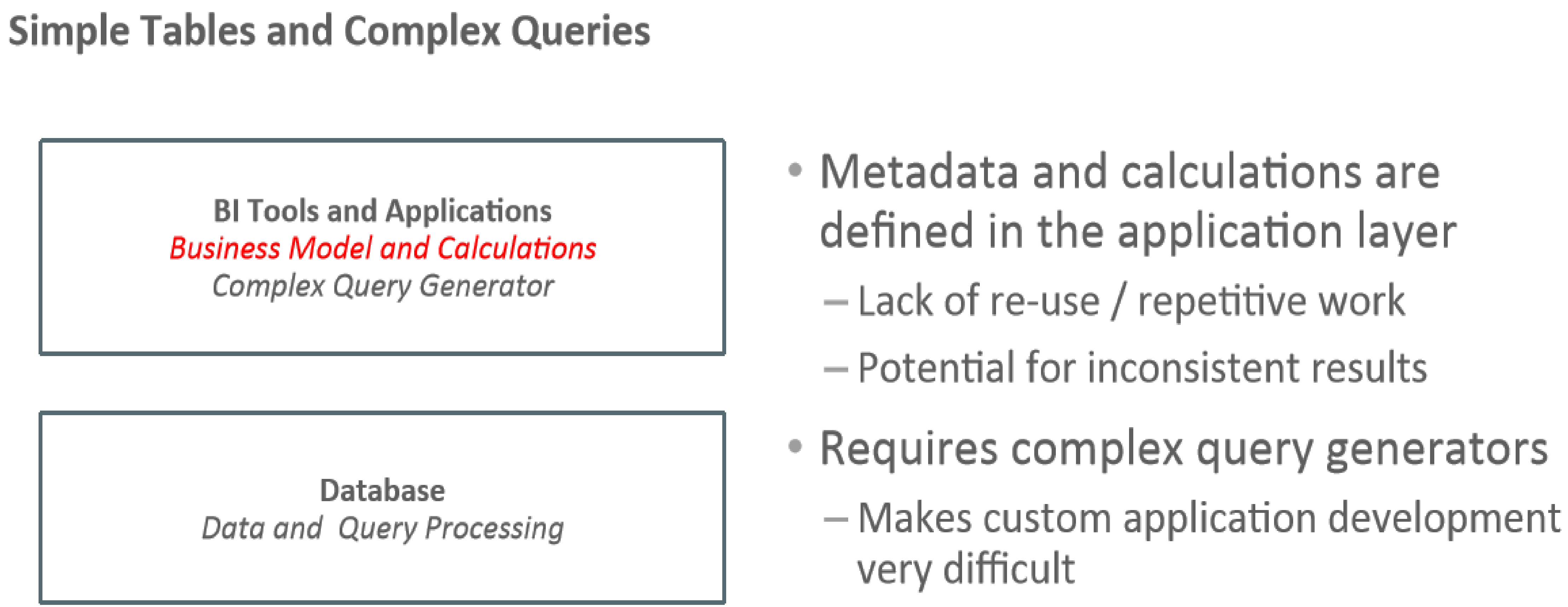
Tim Vlamis

Oracle Press

Agenda

- Current approach to BI
- Analytic Views Simple Select
- Analytic Views Modeling

Today’s Approach to BI



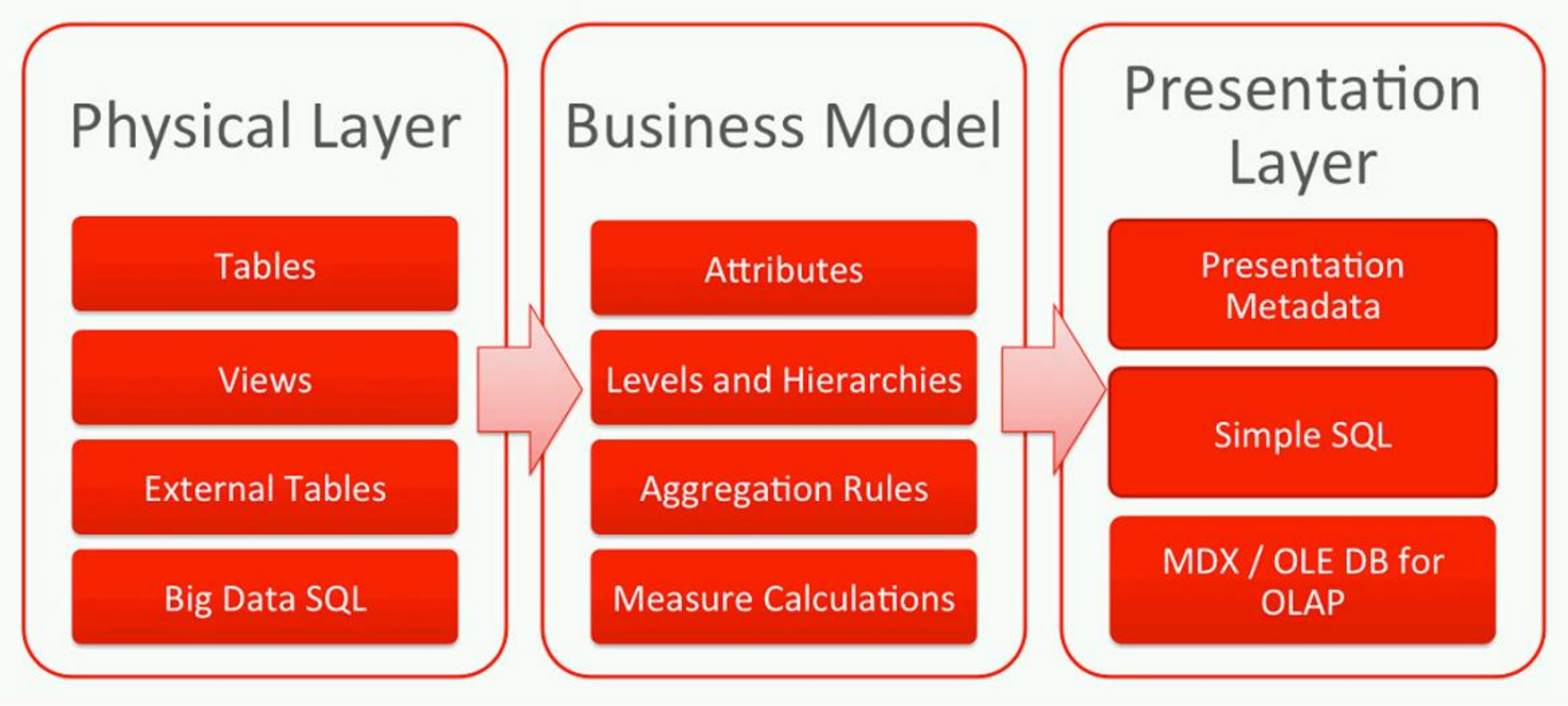
Simple Tables and Complex Queries

BI Tools and Applications
Business Model and Calculations
Complex Query Generator

Database
Data and Query Processing

- Metadata and calculations are defined in the application layer
  - Lack of re-use / repetitive work
  - Potential for inconsistent results
- Requires complex query generators
  - Makes custom application development very difficult

Access, Model and Present



Physical Layer

- Tables
- Views
- External Tables
- Big Data SQL

Business Model

- Attributes
- Levels and Hierarchies
- Aggregation Rules
- Measure Calculations

Presentation Layer

- Presentation Metadata
- Simple SQL
- MDX / OLE DB for OLAP

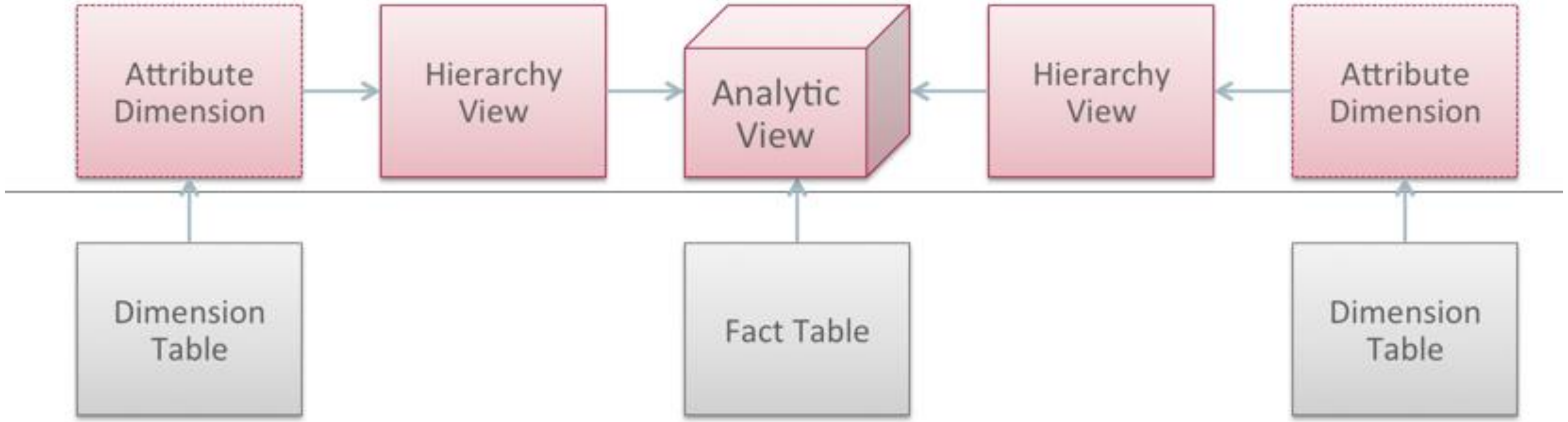
New in 12.2 – Analytic Views

- Moves business logic back into the database (Aggregations, Hierarchies, Calculations)
- Simple SQL for complex analytic queries
  - No joins or Group-By clauses necessary
- Works on top of the pre-existing tables and views
  - No persistent storage
- Built-in data visualization via APEX

Analytic Views

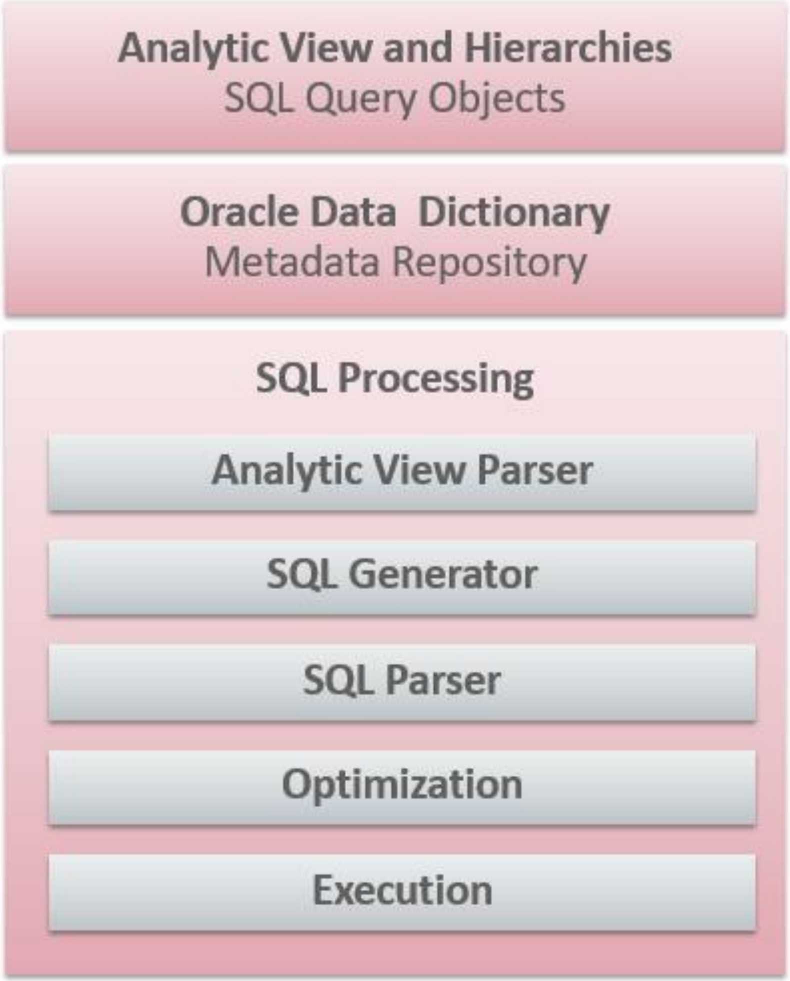
Better for Everyone

- For the data warehouse architect and developer
  - Easily extend star schema with aggregate data and calculations
- For the application developer
  - Simplifies metadata management and SQL generation
- For the business user
  - Built-in, browser-based data visualization via APEX application



Analytic Views

- Analytic Views and Hierarchies
  - Objects that are queried with SQL
- Data Dictionary
  - All metadata for analytic views
- Analytic View Parser
  - Syntax and sematic checks
- SQL Generator
  - Transforms AV SQL into executable SQL
- SQL Parser, Optimization and Execution
  - Oracle SQL engine



Analytic View and Hierarchies
SQL Query Objects

Oracle Data Dictionary
Metadata Repository

SQL Processing

Analytic View Parser

SQL Generator

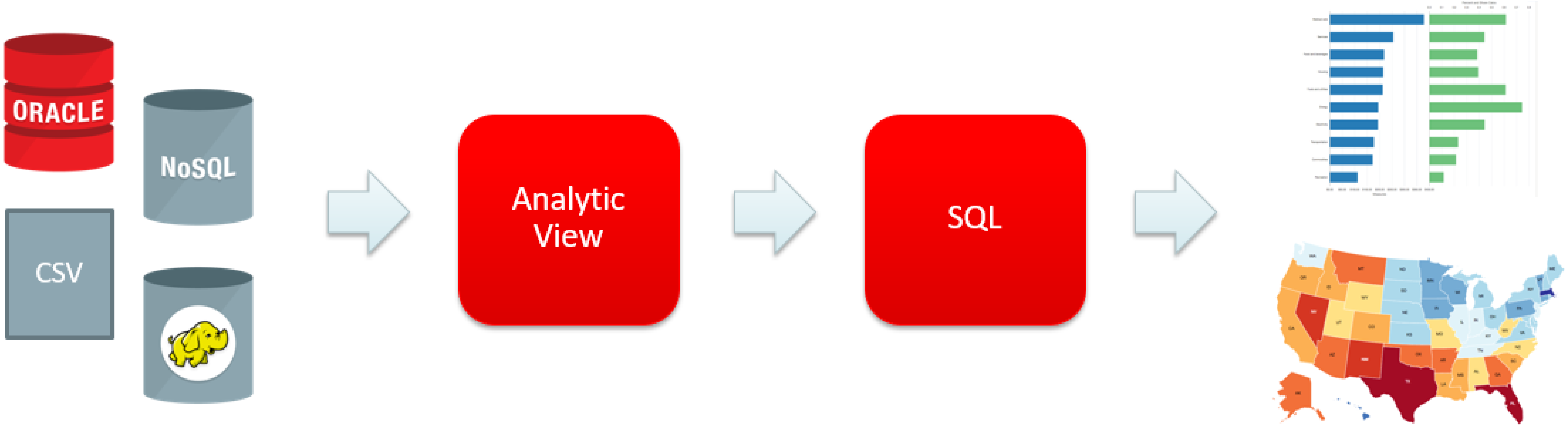
SQL Parser

Optimization

Execution

Analytic Views

Easier Access To Your Data



ORACLE

NoSQL

CSV

Analytic View

SQL

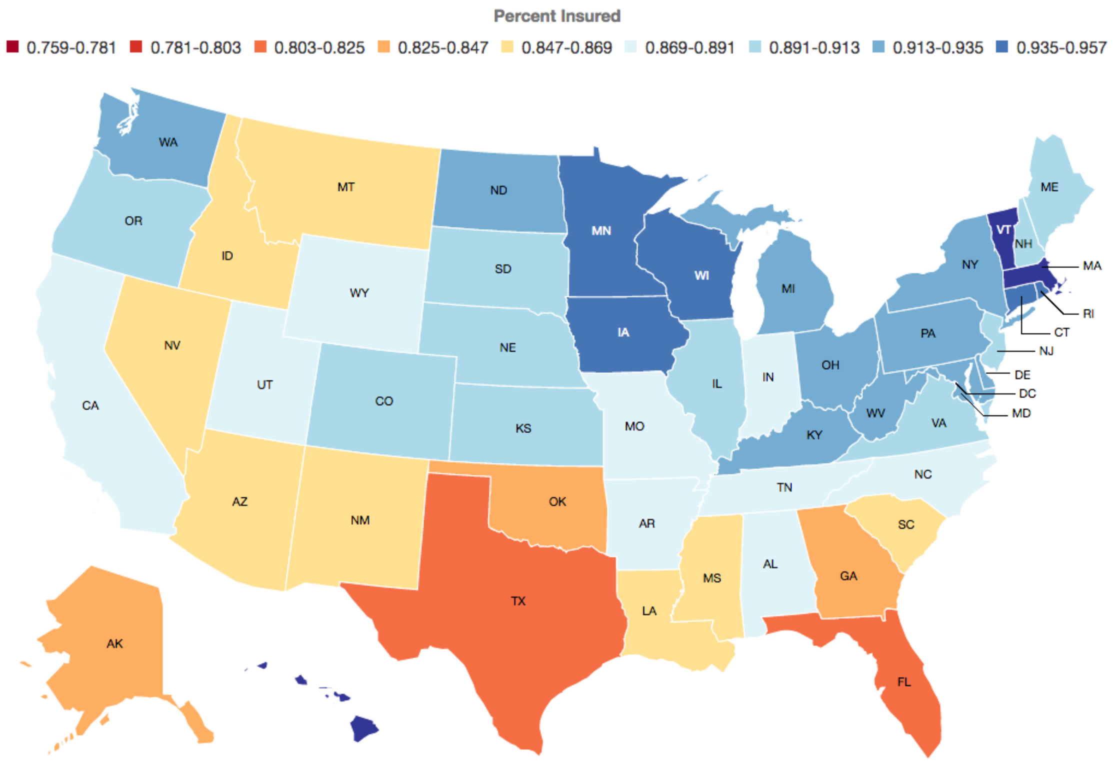
Your Data

Organized
&
Enhanced

Simple SQL

Your
Applications

Analytic Views

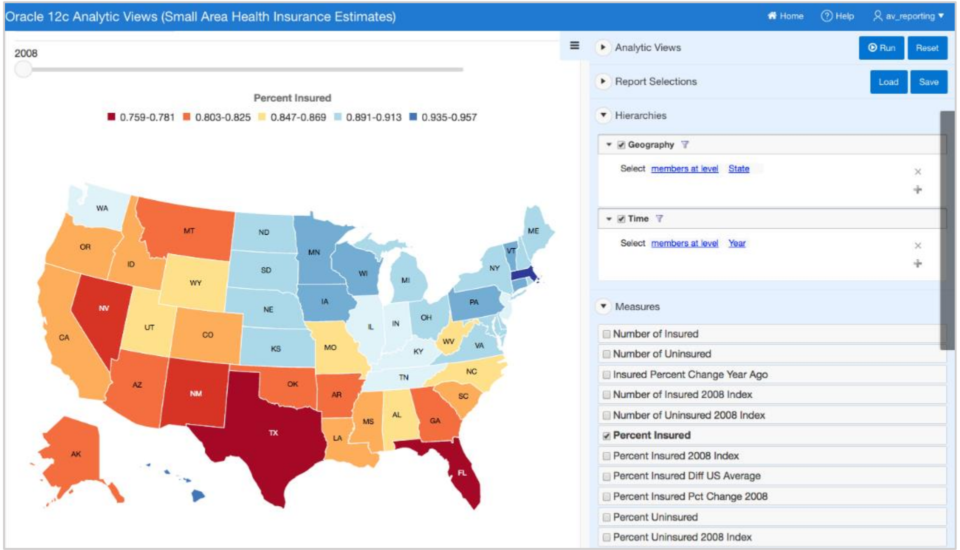


Health Insurance Coverage Rates by State, 2014

- How would you build this application?
  - Analysis of health insurance coverage rates in the United States
  - Coverage rates by time, counties and states
  - Geographic comparisons
  - Measure improvement over time
  - Interactive data visualization tools for end users

Analytic Views

- This application can be built with 5 SQL statements
  - Create 2 hierarchies (4 SQL statements)
  - Create 1 analytic view (1 SQL statement)
- Is instantly accessible via APEX based application
- Is all in the Database



Oracle 12c Analytic Views (Small Area Health Insurance Estimates)

2008

Percent Insured

■ 0.759-0.781 ■ 0.803-0.825 ■ 0.847-0.869 ■ 0.891-0.913 ■ 0.935-0.957

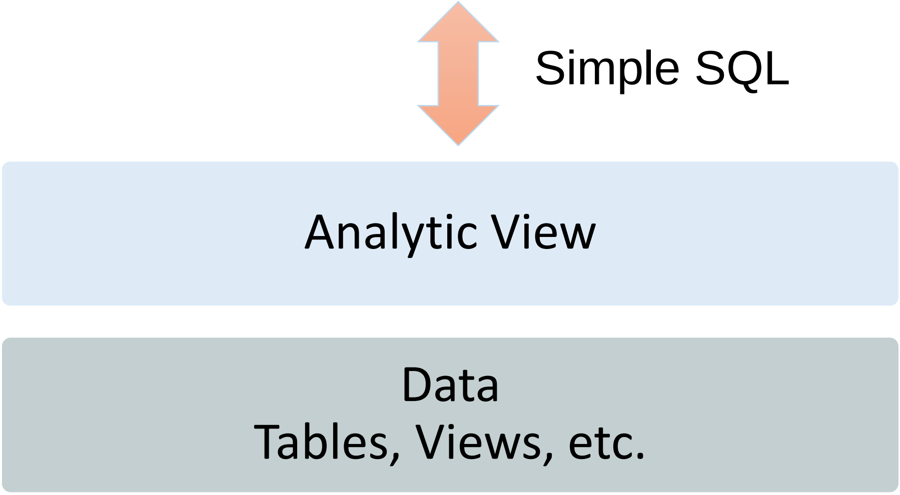
Hierarchies

☑ Geography: Select members at level State

☑ Time: Select members at level Year

Measures

- □ Number of Insured
- □ Number of Uninsured
- □ Insured Percent Change Year Ago
- □ Number of Insured 2008 Index
- □ Number of Uninsured 2008 Index
- ☑ **Percent Insured**
- □ Percent Insured 2008 Index
- □ Percent Insured Diff US Average
- □ Percent Insured Pct Change 2008
- □ Percent Uninsured
- □ Percent Uninsured 2008 Index



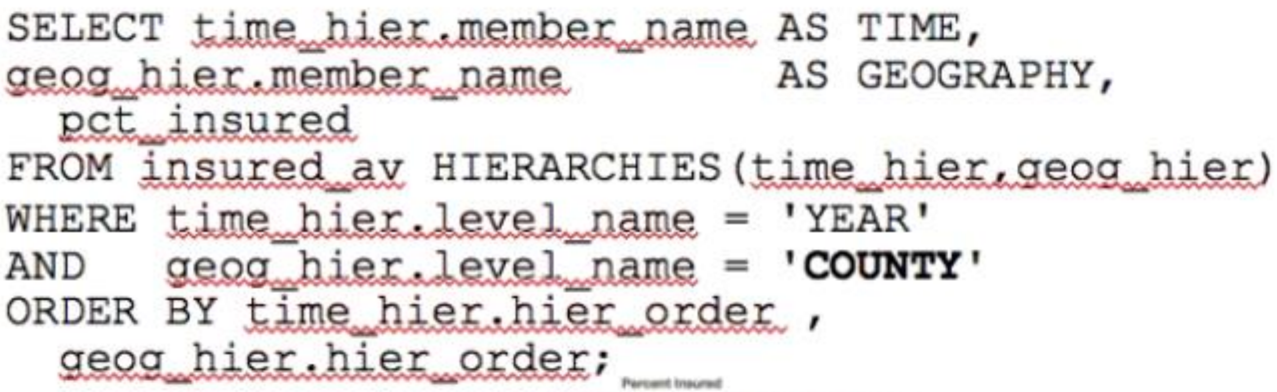
Simple SQL

Analytic View

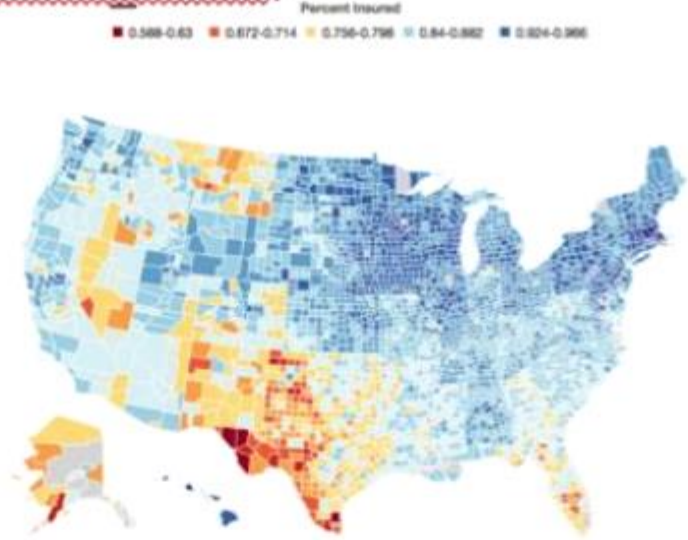
Data
Tables, Views, etc.

Analytic Views

Simple SQL



```
SELECT time_hier.member_name AS TIME,
geog_hier.member_name        AS GEOGRAPHY,
  pct_insured
FROM insured_av HIERARCHIES(time_hier,geog_hier)
WHERE time_hier.level_name = 'YEAR'
AND   geog_hier.level_name = 'COUNTY'
ORDER BY time_hier.hier_order ,
  geog_hier.hier_order;
```



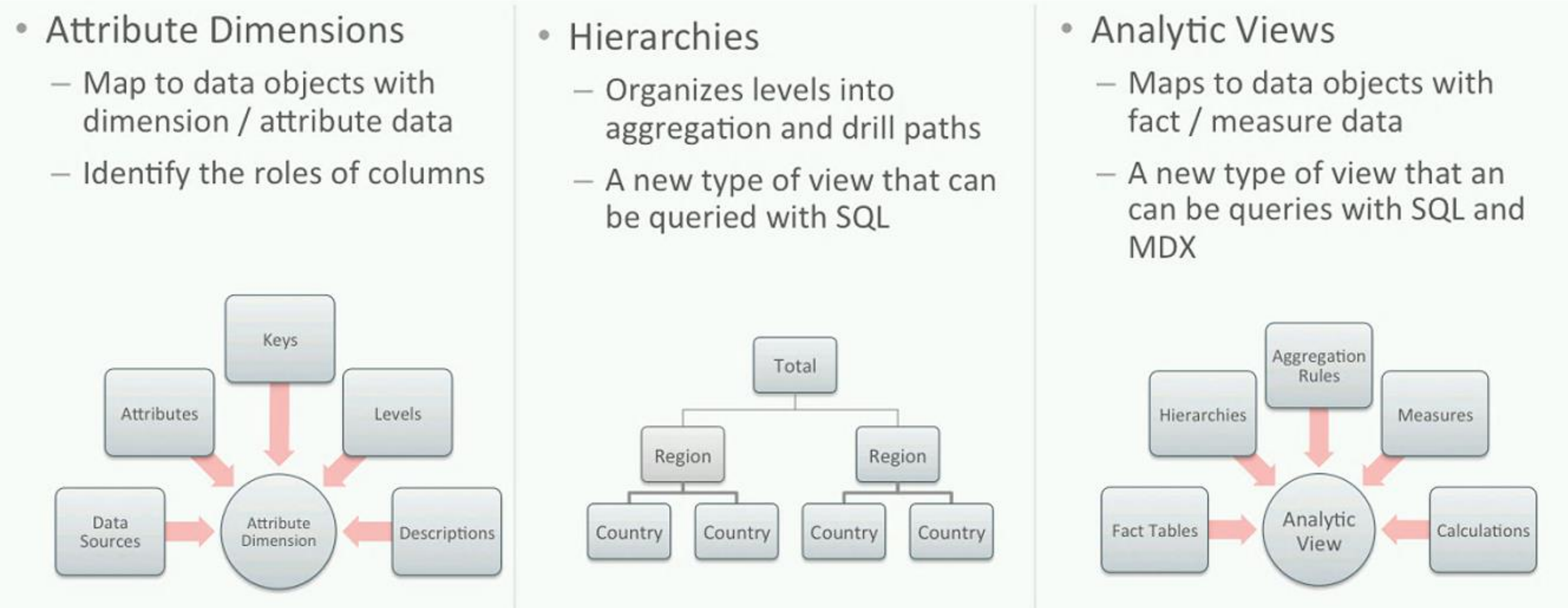
To drill, just update the WHERE clause. Everything else remains the same.

The calculations automatically use new hierarchy levels.

New Database Objects

- New type of view in the Oracle Database
  - Business model and calculation rules are embedded within the Analytic View
  - Purely relational concept – no instantiation of the data
- Analytic Views as easily queried with simple SQL or MDX
  - With a smart Analytic view, SQL generation is easy
- Access data from tables, views, external tables and Big Data SQL
  - Use Analytic Views to organize and present a wide variety of data

Three New Database Objects



- Attribute Dimensions
  - Map to data objects with dimension / attribute data
  - Identify the roles of columns

- Hierarchies
  - Organizes levels into aggregation and drill paths
  - A new type of view that can be queried with SQL

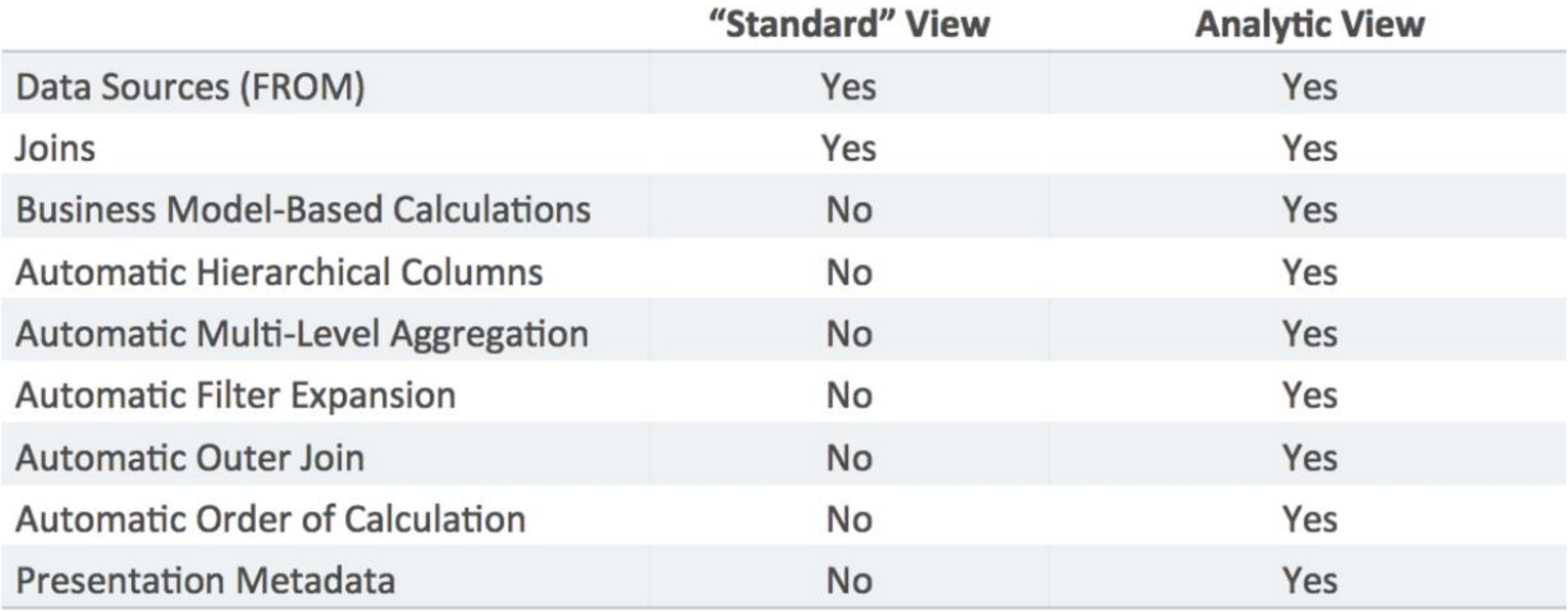
- Analytic Views
  - Maps to data objects with fact / measure data
  - A new type of view that an can be queries with SQL and MDX

Advantages

Data is already Joined, Aggregated and Calculated

- Joins all hierarchies and fact data into a single view (de-normalized)
  - No Joins required in query
- Returns rows for all aggregate data
  - GROUP BY not required in query
- Presents calculated measures as a single Column
  - Just select the column name
- Query transformation engine accesses and calculates data
  - No pre-calculation is required

“Standard” vs Analytic Views

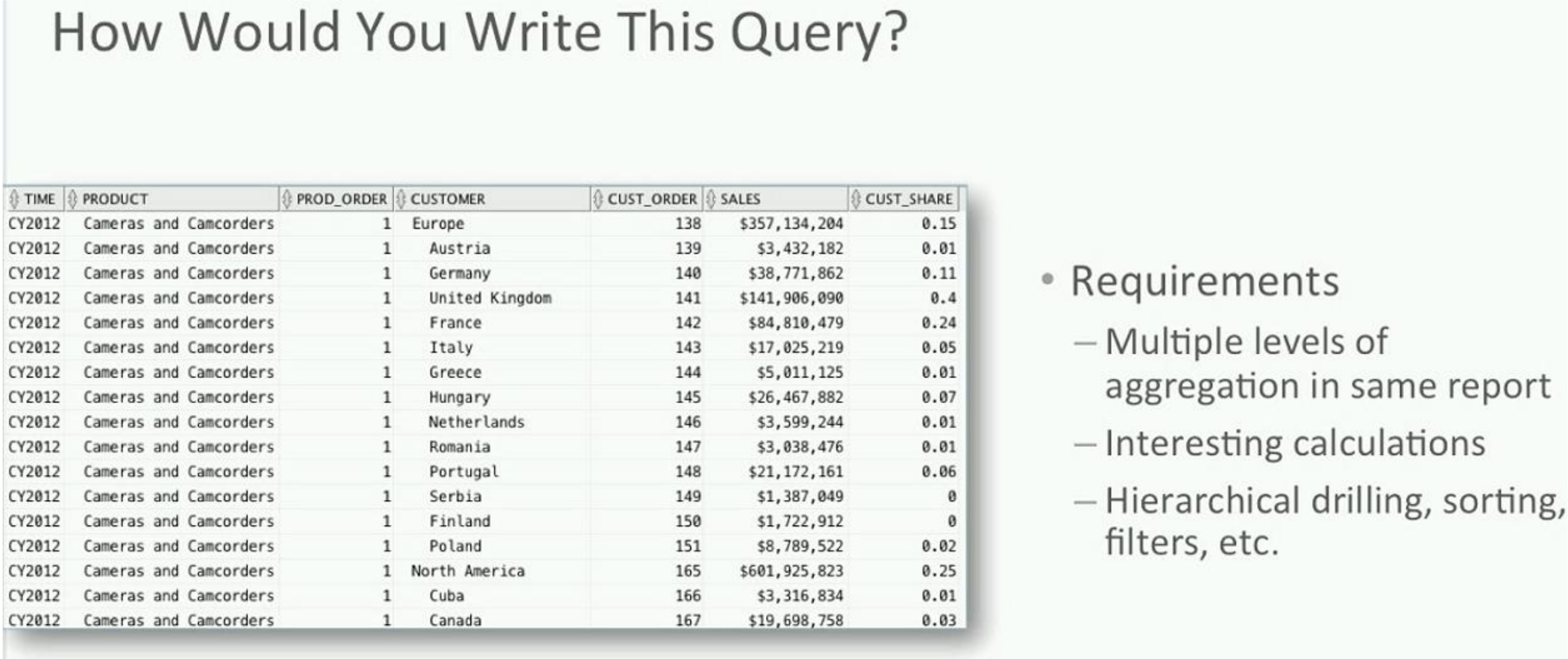


	“Standard” View	Analytic View
Data Sources (FROM)	Yes	Yes
Joins	Yes	Yes
Business Model-Based Calculations	No	Yes
Automatic Hierarchical Columns	No	Yes
Automatic Multi-Level Aggregation	No	Yes
Automatic Filter Expansion	No	Yes
Automatic Outer Join	No	Yes
Automatic Order of Calculation	No	Yes
Presentation Metadata	No	Yes

Analytic Views vs DB IM vs Oracle OLAP

Concept	Oracle OLAP Cubes	Database In Memory	Analytic Views
Aggregation	Pre-aggregate for fast access	Aggregated on-the-fly via SIMD instruction set	Aggregated on-the-fly
Effect when base data changes	Need to re-solve since pre-compute aggregates	None since calced on-the-fly	None since calced on-the-fly
Calculated columns	Handles well	Not handled	Handles well
Storage	Stored in cubes	No storage	No storage
Modeling	Modeled using AWM	Alter table InMemory	Define in SQL Developer
Speed	Fastest for aggregates because precalced	Fast for aggregates because of SIMD instruction set	Need DBIM or store aggregates for speed of aggregates
Skill required	Need special OLAP knowledge	Simple – alter table inmemory	Approachable by relational people
Calculated columns	Handles well	Not handled	Handles well
New in DB Version	Version 9	Version 12.1	Version 12.2

Typical Query

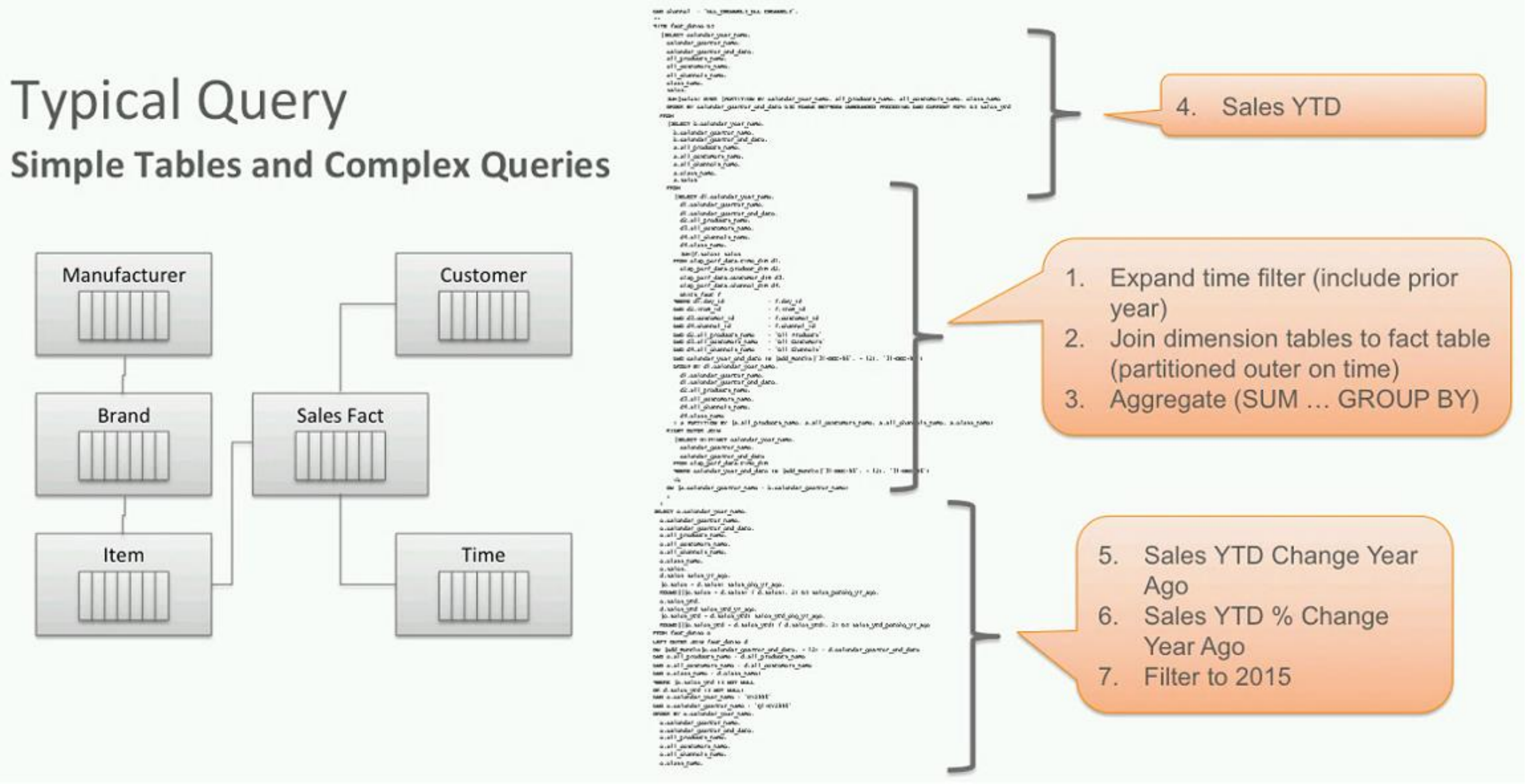


How Would You Write This Query?

TIME	PRODUCT	PROD_ORDER	CUSTOMER	CUST_ORDER	SALES	CUST_SHARE
CY2012	Cameras and Camcorders	1	Europe	138	$357,134,204	0.15
CY2012	Cameras and Camcorders	1	Austria	139	$3,432,182	0.01
CY2012	Cameras and Camcorders	1	Germany	140	$38,771,862	0.11
CY2012	Cameras and Camcorders	1	United Kingdom	141	$141,906,090	0.4
CY2012	Cameras and Camcorders	1	France	142	$84,810,479	0.24
CY2012	Cameras and Camcorders	1	Italy	143	$17,025,219	0.05
CY2012	Cameras and Camcorders	1	Greece	144	$5,011,125	0.01
CY2012	Cameras and Camcorders	1	Hungary	145	$26,467,882	0.07
CY2012	Cameras and Camcorders	1	Netherlands	146	$3,599,244	0.01
CY2012	Cameras and Camcorders	1	Romania	147	$3,038,476	0.01
CY2012	Cameras and Camcorders	1	Portugal	148	$21,172,161	0.06
CY2012	Cameras and Camcorders	1	Serbia	149	$1,387,049	0
CY2012	Cameras and Camcorders	1	Finland	150	$1,722,912	0
CY2012	Cameras and Camcorders	1	Poland	151	$8,789,522	0.02
CY2012	Cameras and Camcorders	1	North America	165	$601,925,823	0.25
CY2012	Cameras and Camcorders	1	Cuba	166	$3,316,834	0.01
CY2012	Cameras and Camcorders	1	Canada	167	$19,698,758	0.03

- Requirements
  - Multiple levels of aggregation in same report
  - Interesting calculations
  - Hierarchical drilling, sorting, filters, etc.

Typical Query



Typical Query

Simple Tables and Complex Queries

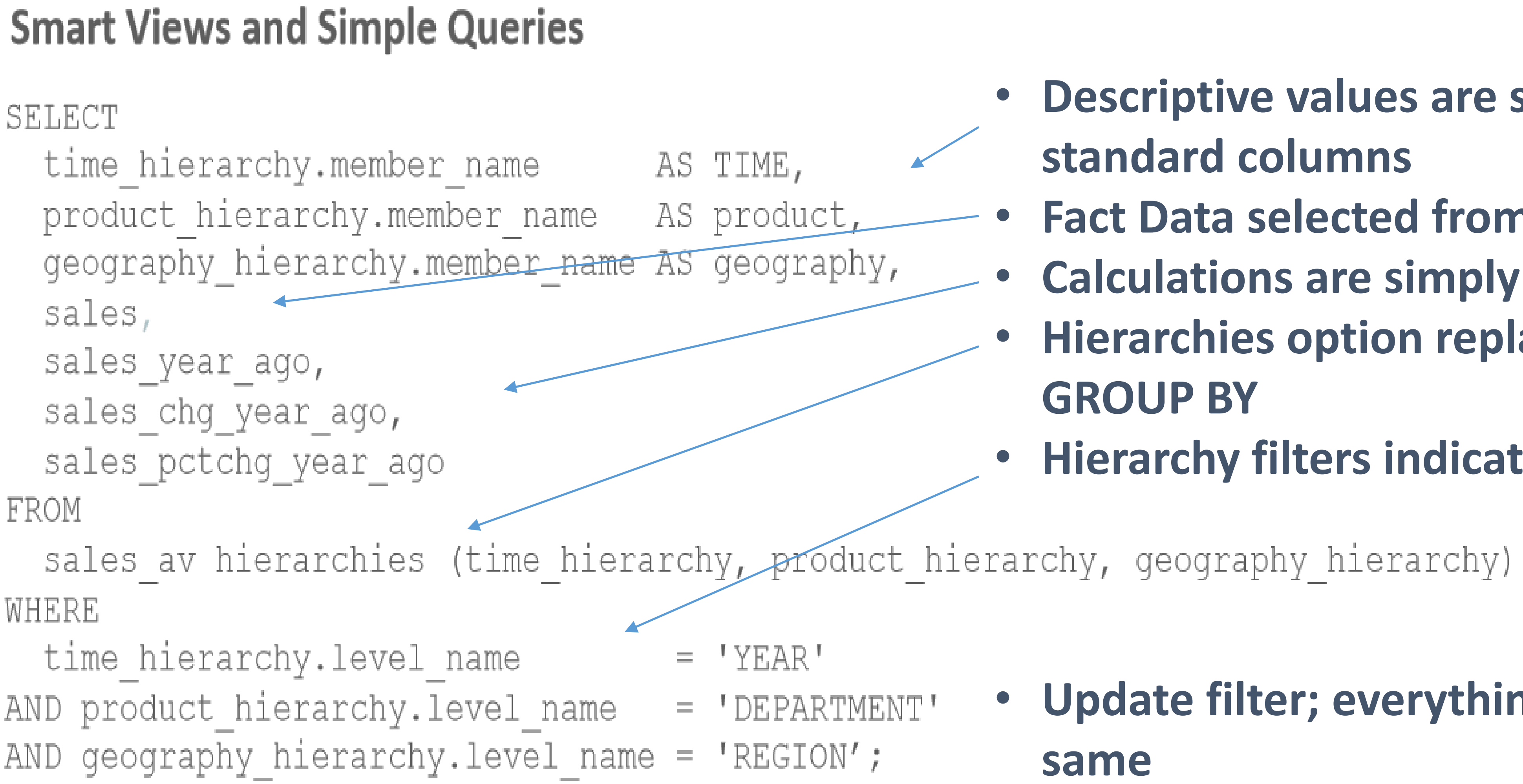
```
and channel = '[illegible]',
--
with fact_data as
 (select calendar_year_name,
  calendar_quarter_name,
  calendar_quarter_end_date,
  all_products_name,
  all_customers_name,
  all_channels_name,
  class_name,
  sales,
  sum(sales) over (partition by calendar_year_name, all_products_name, all_customers_name, class_name
  order by calendar_quarter_end_date asc range between unbounded preceding and current row) as sales_ytd
 from
  (select b.calendar_year_name,
   b.calendar_quarter_name,
   b.calendar_quarter_end_date,
   a.all_products_name,
   a.all_customers_name,
   a.all_channels_name,
   a.class_name,
   a.sales
  from
   (select d1.calendar_year_name,
    d1.calendar_quarter_name,
    d1.calendar_quarter_end_date,
    d2.all_products_name,
    d3.all_customers_name,
    d4.all_channels_name,
    d4.class_name,
     sum(f.sales) sales
   from olap_perf_data.time_dim d1,
    olap_perf_data.product_dim d2,
    olap_perf_data.customer_dim d3,
    olap_perf_data.channel_dim d4,
    units_fact f
   where d1.day_id              = f.day_id
   and d2.item_id               = f.item_id
   and d3.customer_id           = f.customer_id
   and d4.channel_id            = f.channel_id
   and d2.all_products_name     = 'All Products'
   and d3.all_customers_name    = 'All Customers'
   and d4.all_channels_name     = 'All Channels'
   and calendar_year_end_date in (add_months('31-DEC-15', - 12), '31-DEC-15')
   group by d1.calendar_year_name,
    d1.calendar_quarter_name,
    d1.calendar_quarter_end_date,
    d2.all_products_name,
    d3.all_customers_name,
    d4.all_channels_name,
    d4.class_name
   ) a partition by (a.all_products_name, a.all_customers_name, a.all_channels_name, a.class_name)
  right outer join
   (select distinct calendar_year_name,
    calendar_quarter_name,
    calendar_quarter_end_date
   from olap_perf_data.time_dim
   where calendar_year_end_date in (add_months('31-DEC-15', - 12), '31-DEC-15')
   ) b
  on (a.calendar_quarter_name = b.calendar_quarter_name)
  )
 )
select a.calendar_year_name,
 a.calendar_quarter_name,
 a.calendar_quarter_end_date,
 a.all_products_name,
 a.all_customers_name,
 a.all_channels_name,
 a.class_name,
 a.sales,
 d.sales sales_yr_ago,
 (a.sales - d.sales) sales_chg_yr_ago,
 round(((a.sales - d.sales) / d.sales), 2) as sales_pctchg_yr_ago,
 a.sales_ytd,
 d.sales_ytd sales_ytd_yr_ago,
 (a.sales_ytd - d.sales_ytd) sales_ytd_chg_yr_ago,
 round(((a.sales_ytd - d.sales_ytd) / d.sales_ytd), 2) as sales_ytd_pctchg_yr_ago
from fact_data a
left outer join fact_data d
on (add_months(a.calendar_quarter_end_date, - 12) = d.calendar_quarter_end_date
and a.all_products_name = d.all_products_name
and a.all_customers_name = d.all_customers_name
and a.class_name = d.class_name)
where (a.sales_ytd is not null
or d.sales_ytd is not null)
and a.calendar_year_name = 'CY2015'
and a.calendar_quarter_name = 'Q1-CY2015'
order by a.calendar_year_name,
 a.calendar_quarter_name,
 a.calendar_quarter_end_date,
 a.all_products_name,
 a.all_customers_name,
 a.all_channels_name,
 a.class_name;
```

4. Sales YTD

1. Expand time filter (include prior year)
2. Join dimension tables to fact table (partitioned outer on time)
3. Aggregate (SUM … GROUP BY)

5. Sales YTD Change Year Ago
6. Sales YTD % Change Year Ago
7. Filter to 2015

Hierarchy Queries



Smart Views and Simple Queries

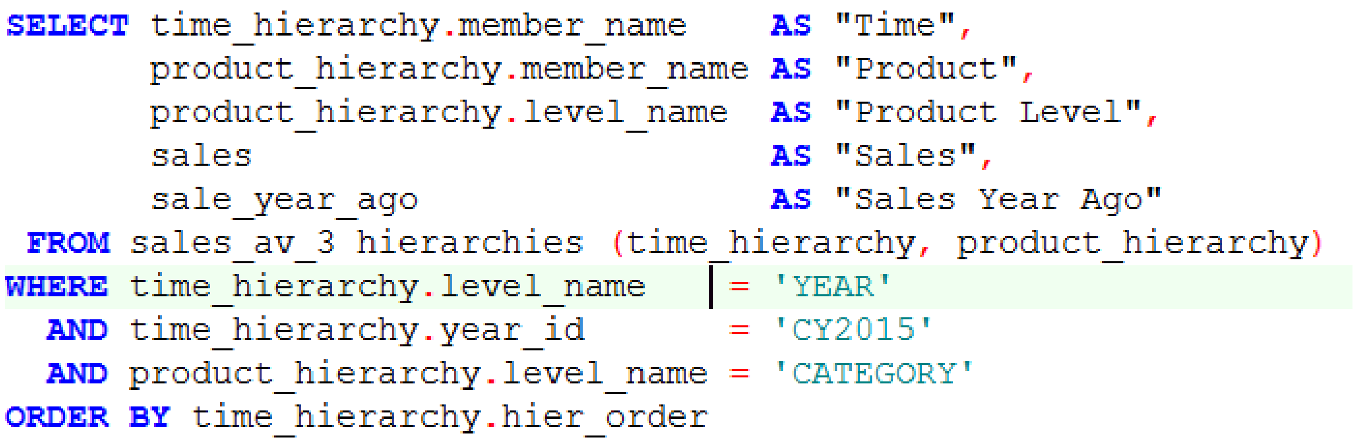
```
SELECT
  time_hierarchy.member_name      AS TIME,
  product_hierarchy.member_name   AS product,
  geography_hierarchy.member_name AS geography,
  sales,
  sales_year_ago,
  sales_chg_year_ago,
  sales_pctchg_year_ago
FROM
  sales_av hierarchies (time_hierarchy, product_hierarchy, geography_hierarchy)
WHERE
  time_hierarchy.level_name        = 'YEAR'
AND product_hierarchy.level_name   = 'DEPARTMENT'
AND geography_hierarchy.level_name = 'REGION';
```

- Descriptive values are selected from standard columns
- Fact Data selected from view
- Calculations are simply selected as column
- Hierarchies option replaces JOIN and GROUP BY
- Hierarchy filters indicate aggregation level

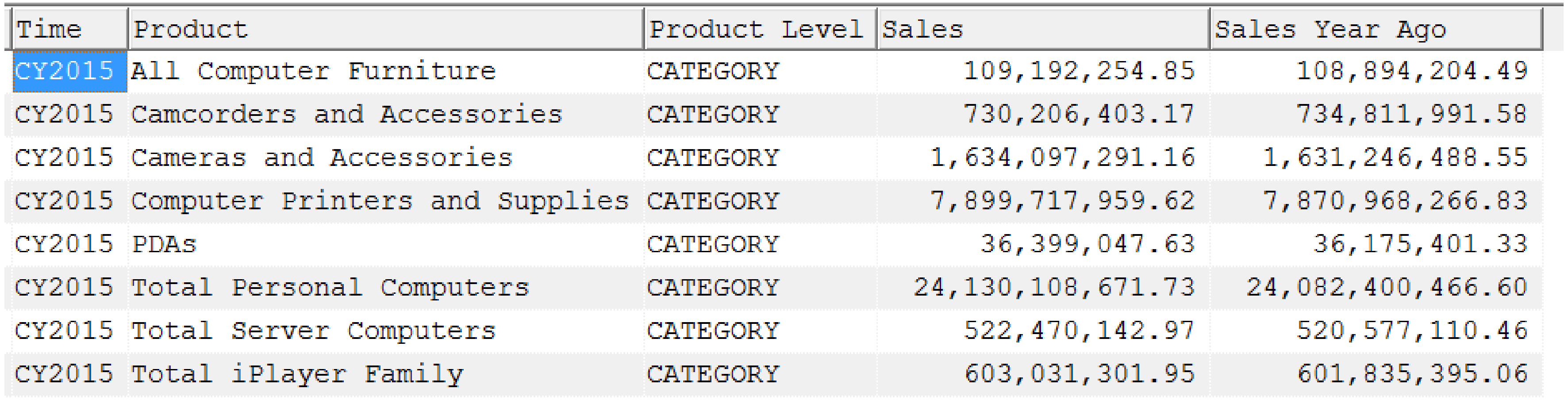
- Update filter; everything else stays the same

Hierarchy Queries

- Simply select returning Sales and Sales Year Ago by Year and Product Category



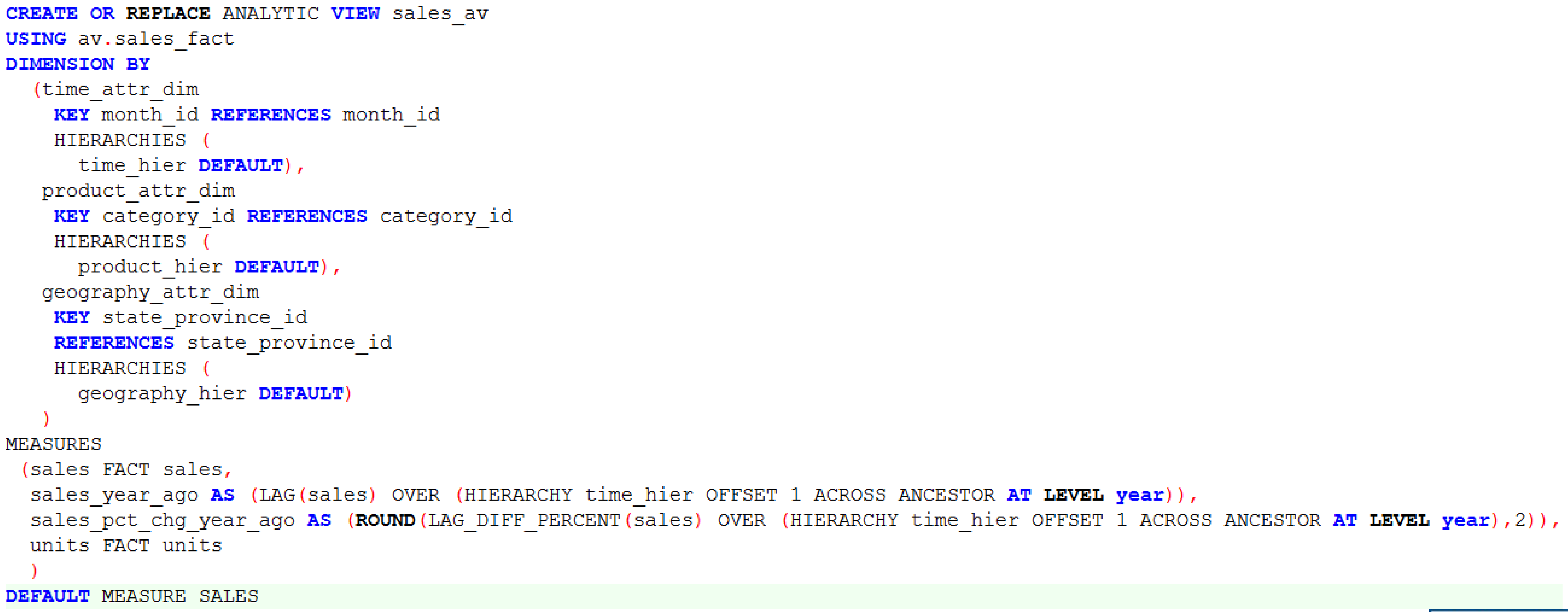
```
SELECT time_hierarchy.member_name    AS "Time",
       product_hierarchy.member_name AS "Product",
       product_hierarchy.level_name  AS "Product Level",
       sales                         AS "Sales",
       sale_year_ago                 AS "Sales Year Ago"
 FROM sales_av_3 hierarchies (time_hierarchy, product_hierarchy)
WHERE time_hierarchy.level_name    = 'YEAR'
  AND time_hierarchy.year_id       = 'CY2015'
  AND product_hierarchy.level_name = 'CATEGORY'
ORDER BY time_hierarchy.hier_order
```



Time	Product	Product Level	Sales	Sales Year Ago
CY2015	All Computer Furniture	CATEGORY	109,192,254.85	108,894,204.49
CY2015	Camcorders and Accessories	CATEGORY	730,206,403.17	734,811,991.58
CY2015	Cameras and Accessories	CATEGORY	1,634,097,291.16	1,631,246,488.55
CY2015	Computer Printers and Supplies	CATEGORY	7,899,717,959.62	7,870,968,266.83
CY2015	PDAs	CATEGORY	36,399,047.63	36,175,401.33
CY2015	Total Personal Computers	CATEGORY	24,130,108,671.73	24,082,400,466.60
CY2015	Total Server Computers	CATEGORY	522,470,142.97	520,577,110.46
CY2015	Total iPlayer Family	CATEGORY	603,031,301.95	601,835,395.06

Analytic View Model

- Script to build SALES_AV
- Note the USING and DIMENSION BY statements.

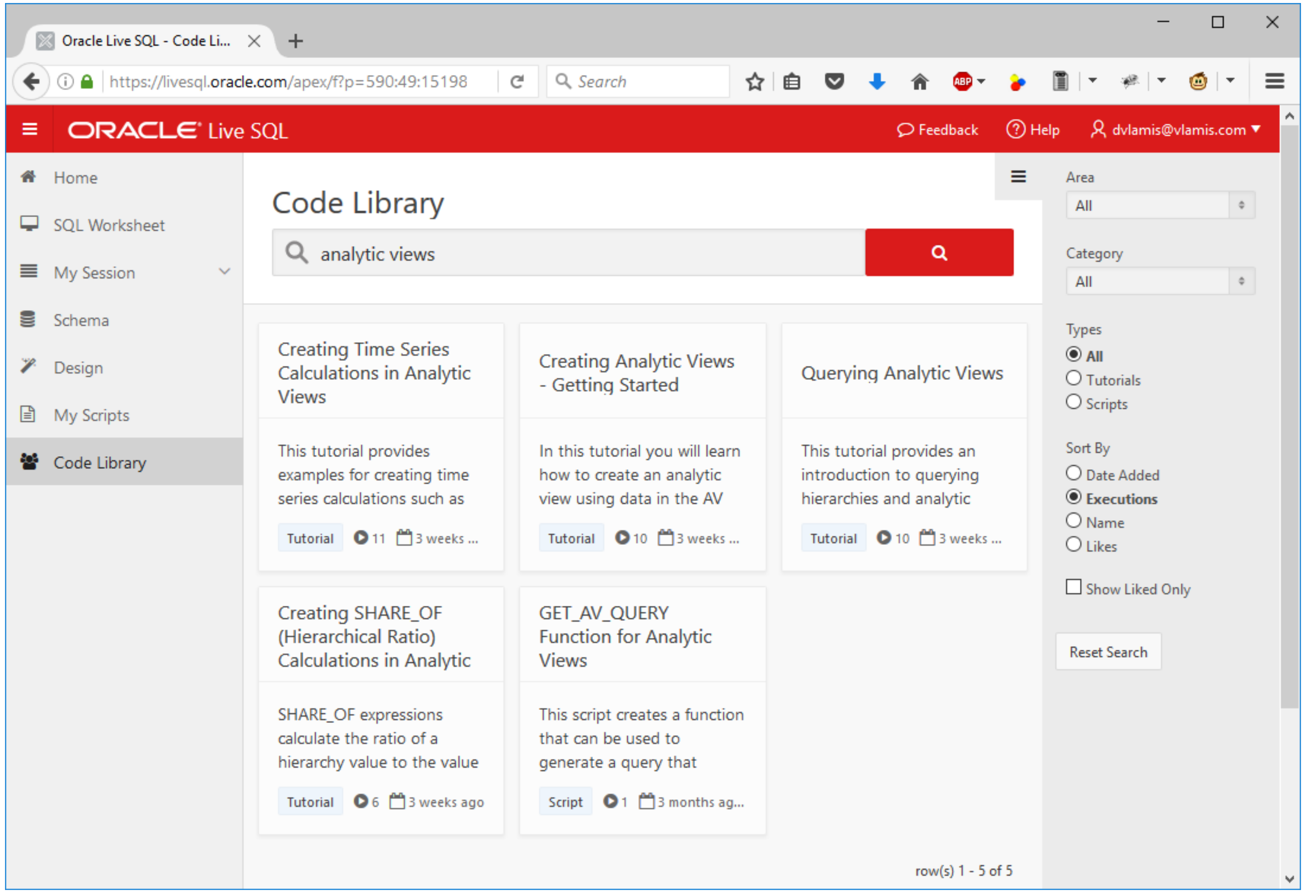


```
CREATE OR REPLACE ANALYTIC VIEW sales_av
USING av.sales_fact
DIMENSION BY
  (time_attr_dim
    KEY month_id REFERENCES month_id
    HIERARCHIES (
      time_hier DEFAULT),
   product_attr_dim
    KEY category_id REFERENCES category_id
    HIERARCHIES (
      product_hier DEFAULT),
   geography_attr_dim
    KEY state_province_id
    REFERENCES state_province_id
    HIERARCHIES (
      geography_hier DEFAULT)
   )
MEASURES
 (sales FACT sales,
  sales_year_ago AS (LAG(sales) OVER (HIERARCHY time_hier OFFSET 1 ACROSS ANCESTOR AT LEVEL year)),
  sales_pct_chg_year_ago AS (ROUND(LAG_DIFF_PERCENT(sales) OVER (HIERARCHY time_hier OFFSET 1 ACROSS ANCESTOR AT LEVEL year),2)),
  units FACT units
  )
DEFAULT MEASURE SALES
```

LiveSQL Demo

http://livesql.oracle.com

- Sign in with Oracle user account
- Search: “Analytic Views”
  - On right hand side of screen, select ‘Tutorials”
- You will find several Analytic Views Tutorials



Oracle Live SQL - Code Li...

https://livesql.oracle.com/apex/f?p=590:49:15198

Search

ORACLE Live SQL

Feedback Help dvlamis@vlamis.com

Home

SQL Worksheet

My Session

Schema

Design

My Scripts

Code Library

Code Library

analytic views

Creating Time Series Calculations in Analytic Views

This tutorial provides examples for creating time series calculations such as

Tutorial 11 3 weeks ...

Creating Analytic Views - Getting Started

In this tutorial you will learn how to create an analytic view using data in the AV

Tutorial 10 3 weeks ...

Querying Analytic Views

This tutorial provides an introduction to querying hierarchies and analytic

Tutorial 10 3 weeks ...

Creating SHARE_OF (Hierarchical Ratio) Calculations in Analytic

SHARE_OF expressions calculate the ratio of a hierarchy value to the value

Tutorial 6 3 weeks ago

GET_AV_QUERY Function for Analytic Views

This script creates a function that can be used to generate a query that

Script 1 3 months ag...

row(s) 1 - 5 of 5

Area: All

Category: All

Types

● All
○ Tutorials
○ Scripts

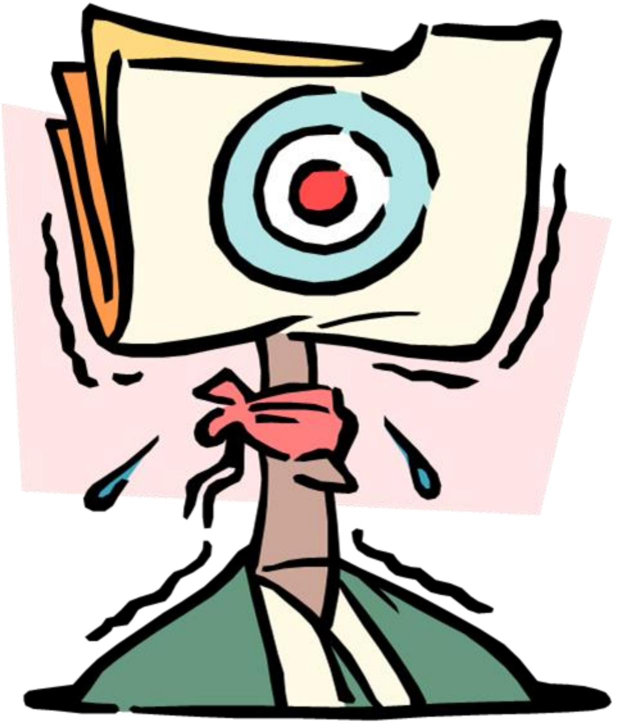
Sort By

○ Date Added
● Executions
○ Name
○ Likes

□ Show Liked Only

Reset Search

Questions?



Thank You!

Title:	Business Analytics in the Oracle 12.2 Database: Analytic Views
Presenter:	Dan Vlamis dvlamis@vlamis.com
Date:	January 31, 2017

Slides available at www.vlamis.com/papers/