

# Analytic Views Simplify Complex Business Intelligence Queries

Event: IOUG Collaborate 16

Session: 4790

Presenter: Dan VlamiS, Michael Caskey

Date: April 11, 2016

# VlamiS Software Solutions

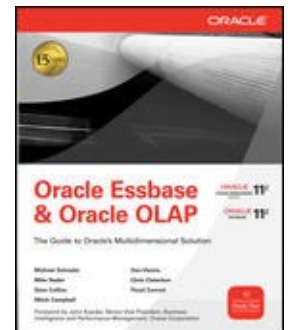
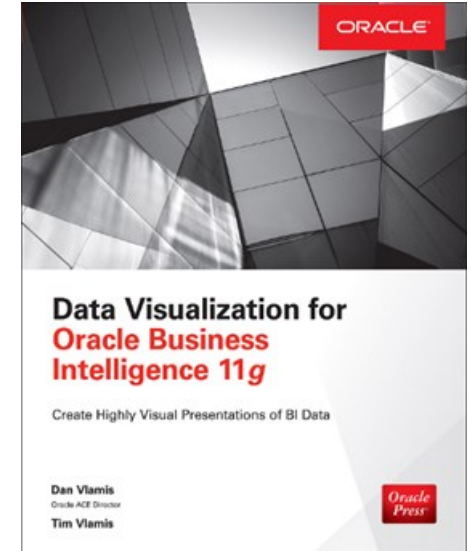
- VlamiS Software founded in 1992 in Kansas City, Missouri
- Developed 200+ Oracle BI and analytics systems
- Specializes in Oracle-based:
  - Enterprise Business Intelligence
  - Data Warehousing
  - Data Mining and Predictive Analytics
  - Data Visualization
- Multiple Oracle ACEs, consultants average 15+ years
- [www.vlamiS.com](http://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

 EDUCATION RESELLER

 APPROVED  
EDUCATION CENTER

 Gold  
Partner

Specialized  
Oracle Business Intelligence  
Foundation Suite 11g





# Vlami Involvement in Presentations

Presenter	Time	Location	Title
Dan Vlami & Mike Caskey	Mon 12:00 PM	Banyan C	Analytic Views Simplify Complex Business Intelligence Queries
Dan Vlami & Mike Caskey	Mon 2:00 PM	Banyan C	Upgrading to Oracle Business Intelligence 12c
Jeff McBride & Mike Caskey	Tues 9:15 AM	Breakers I	Case Study of Improving BI Apps and OBIEE Performance
Dan Vlami & Tim Vlami	Tues 3:30 PM	Banyan C	Oracle Big Data Science
Dan Vlami & Tim Vlami	Wed 9:15 AM	Banyan D	Data Analysis with Various Oracle Business Intelligence and Analytics Tools
Tim Vlami	Thurs 12:15 PM	Jasmine F	BI Movie Magic: Maps, Graphs, and BI Dashboards at AMC Theatres



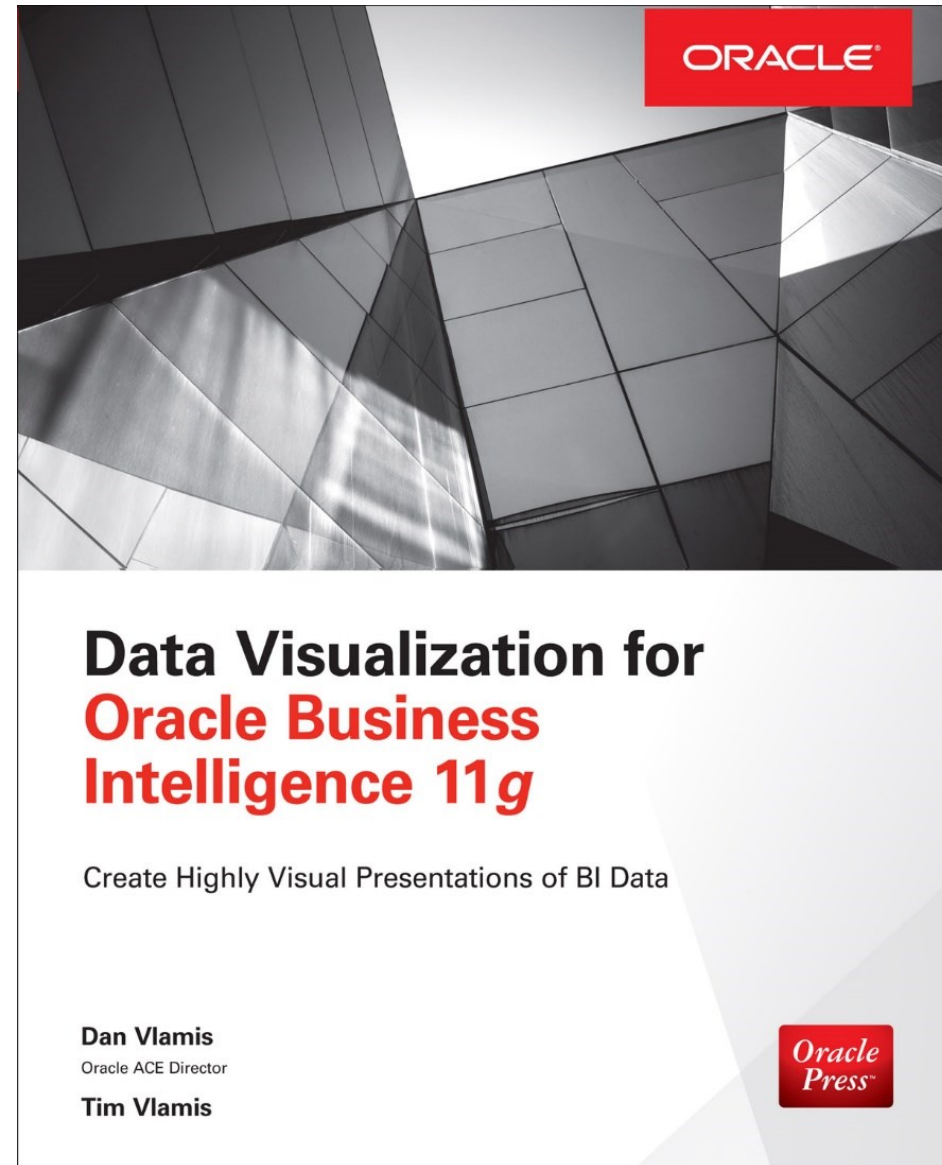
# Dan VlamiS & Mike Caskey

- Dan (OLAP expert and career IT guy)
  - 25+ Years in business intelligence/executive information systems
  - Led development team at IRI
  - Founded VlamiS Software Solutions 20+ years ago in 1993
  - Author, speaker, Oracle ACE Director
  - BA Computer Science Brown University
- 
- Mike (IT Architect and hands-on expert)
  - 20+ years in data warehousing, software engineer and OLAP
  - 10+ years of this time in Healthcare BI as co-founder and lead architect of a software company, developing 6 product solutions
  - Expert in multiple Enterprise Data Warehouse design and implementations across industries



# Drawing for Free Book

Add business card to basket  
or fill out card





# Disclaimer

The information we are about to discuss was discovered and analyzed from an Oracle Open World 2015 and BIWA Summit 2016 presentation. The Oracle Database 12.2 is still in Beta and scheduled for release later this year.

Some slides have been created from audience pictures taken during these prior presentations.



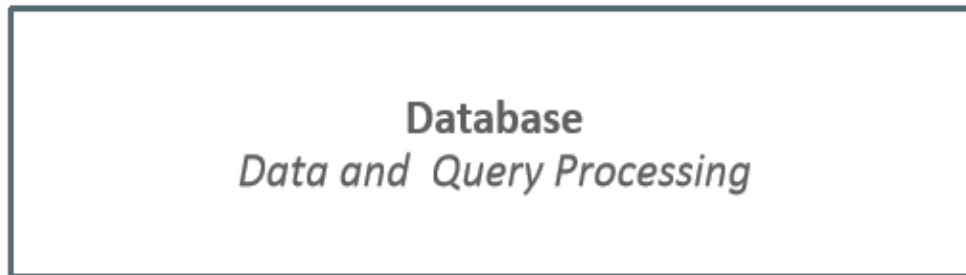
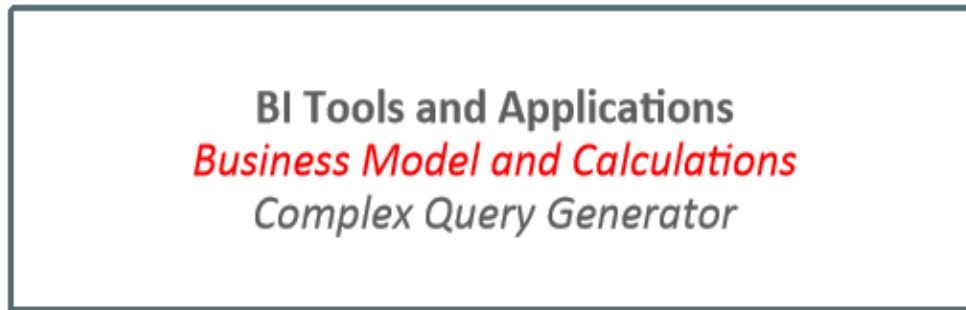
# Agenda

- Current approach to BI
- Analytic Views Simple Select
- Analytic Views Modeling
- Database USER tables



# Today's Approach to BI

## Simple Tables and Complex Queries



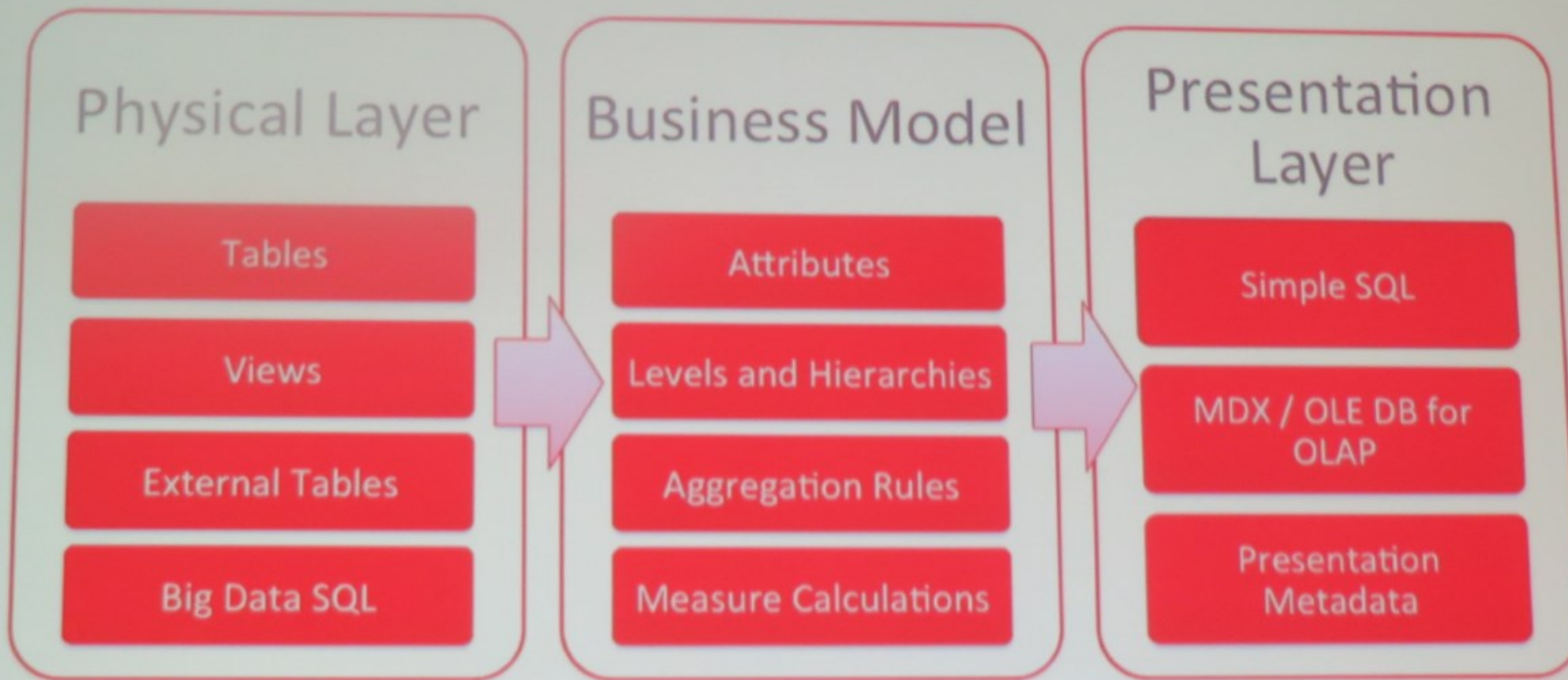
- Metadata and calculations are defined in the application layer
  - Lack of re-use / repetitive work
  - Potential for inconsistent results
- Requires complex query generators
  - Makes custom application development very difficult





# Access, Model and Present

## Analytic Views: Access, Model and Present





# Analytic Views

- 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
  - MDX provider (OLE for OLAP), supports Excel PivotTable connections
- Access data from tables, views, external tables and Big Data SQL
  - Use Analytic Views to organize and present a wide variety of data



# Advantages

## 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



The diagram illustrates a fact table 'Sales Fact' connected to five dimension tables: 'Manufacturer', 'Brand', 'Item', 'Customer', and 'Time'. Each dimension table is represented by a box containing a grid of vertical lines, symbolizing data attributes. The 'Sales Fact' table is centrally located and connected to each of the five dimension tables by a line, indicating its role as the central fact in the data model.

[illegible]

#### 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)

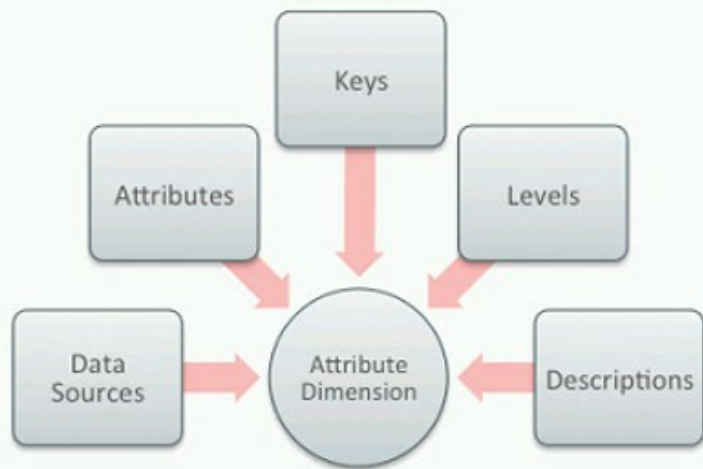
5. Sales YTD Change Year Ago
6. Sales YTD % Change Year Ago
7. Filter to 2015



# Three New Database Objects

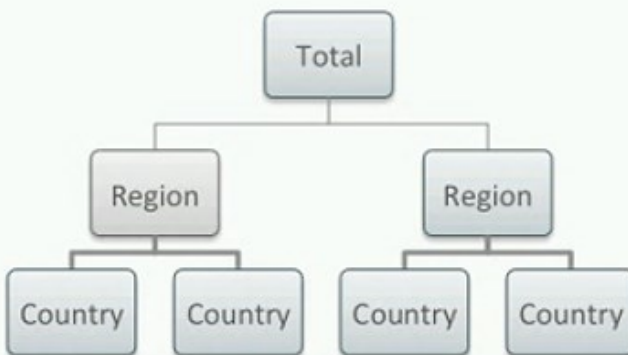
- Attribute Dimensions

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



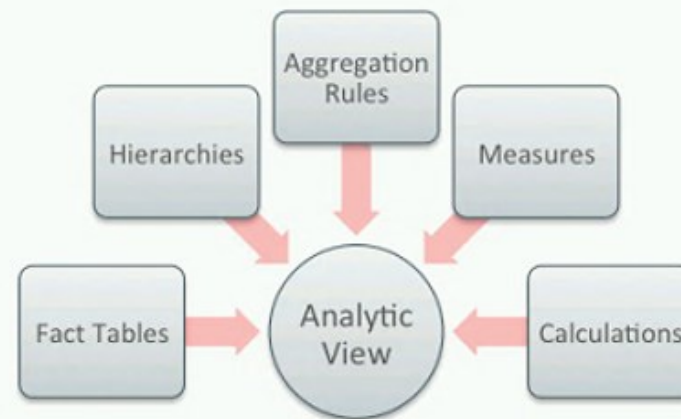
- Hierarchies

- 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 can be queried with SQL and MDX







# 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

## Example of Member Name and Level Name

```
SELECT member_name,  
       level_name  
FROM time_hierarchy  
ORDER BY hier_order;
```

MEMBER_NAME	LEVEL_NAME
ALL YEARS	ALL
CY2011	YEAR
Q1CY2011	QUARTER
Jan-11	MONTH
Feb-11	MONTH
Mar-11	MONTH
Q2CY2011	QUARTER
Apr-11	MONTH
May-11	MONTH
Jun-11	MONTH
Q3CY2011	QUARTER
Jul-11	MONTH



# Analytic View Model

- Script to build SALES\_AV\_3
- Note the USING and DIMENSION BY statements.
- CLASSIFICATIONS add metadata that can be used by applications to enhance the display of data.

```
CREATE OR REPLACE HIERARCHY CUBE sales_av_3
  CLASSIFICATION caption VALUE 'Sales View'
  CLASSIFICATION description VALUE 'Sales View'
  USING sales_fact
  DIMENSION BY
    ( geography_attr_dim KEY state_province_id REFERENCES state_province_id HIERARCHIES (geography_hierarchy DEFAULT)
      product_attr_dim KEY category_id REFERENCES category_id HIERARCHIES (product_hierarchy DEFAULT) ,
      time_attr_dim KEY month_id REFERENCES month_id HIERARCHIES (time_hierarchy 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'
      CLASSIFICATION format_string VALUE '$9,999.99',
    sales_year_ago as (LAG(sales) OVER (HIERARCHY time_hierarchy 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'
  )
  DEFAULT MEASURE SALES;
```





# Hierarchy Queries

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

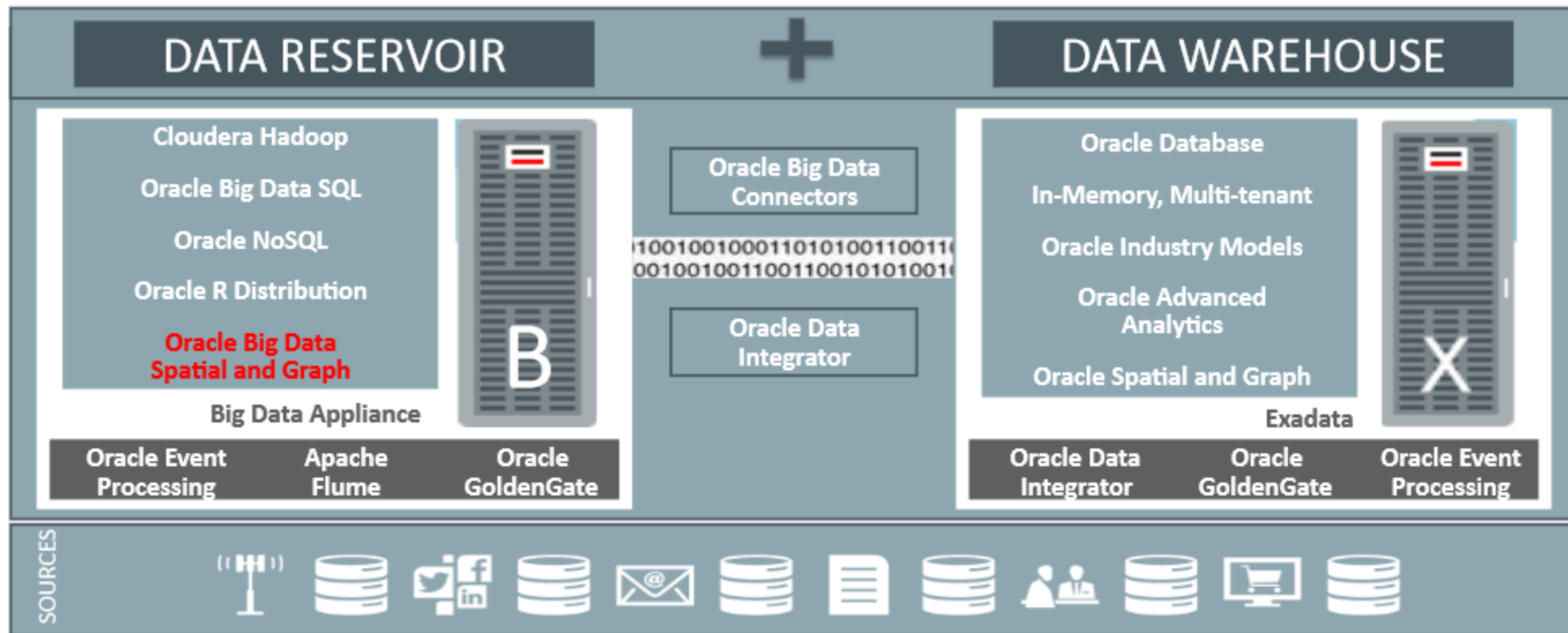
```
SELECT time_hierarchy.member_name    AS "Time",  
       product_hierarchy.member_name AS "Product",  
       product_hierarchy.level_name  AS "Product Level",  
       sales                         AS "Sales",  
       sale_year_ago                 AS "Sales Year Ago"  
FROM sales_av_3 hierarchies (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



# Oracle Big Data Management System

## The Big Picture – Oracle Big Data Management System





# Thank You!

Title: Analytic Views Simplify Complex Business Intelligence Queries

Session: 4790

Presenter: Dan Vlami [dvlamis@vlamis.com](mailto:dvlamis@vlamis.com)  
Mike Caskey [mcaskey@vlamis.com](mailto:mcaskey@vlamis.com)

Date: April 11, 2016

Slides available at [www.vlami.com/papers/](http://www.vlami.com/papers/)