

Advanced OLAP: Making the Hard Stuff Easy!

Collaborate 2013

IOUG Presentation #732
Chris Claterbos

Vlamis Software Solutions
816-781-2880
<http://www.vlamis.com>



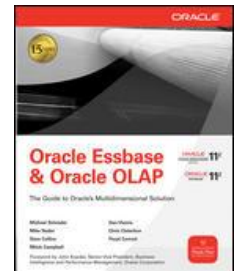
Presentation Agenda

- Introduction
- Why Oracle OLAP
- Problem 1: Count Distinct
- Problem 2: Time Series Analysis
- Problem 3: Measure Hierarchies
- Conclusion
- Q & A



Vlamis Software Solutions

- Vlamis Software founded in 1992 in Kansas City, Missouri
- Developed more than 200 Oracle BI systems
- Specializes in ORACLE-based:
 - Data Warehousing
 - Business Intelligence
 - Design and integrated BI and DW solutions
 - Training and mentoring
- Expert presenter at major Oracle conferences
- www.vlamis.com (blog, papers, newsletters, services)
- Developer for IRI (former owners of Oracle OLAP)
- Co-author of book “Oracle Essbase & Oracle OLAP”
- Beta tester for OBIEE 11g
- Reseller for Simba and Nokia map data for OBIEE
- HOL Coordinator for BIWA Summit 2013





Chris Claterbos



- **Chris Claterbos, Technical Director**

- Technical Director for Vlamis Software Solutions, Inc.
- DBA and applications developer for Oracle products, since 1981.
- Beta tester and early adopter of - including OBIEE11g, Oracle 8i, 9i, 10g and 11g, and Jdeveloper, Oracle OLAP, Data Warehouse Builder
- Speaker and author.
- Previous IOUG Focus Area Manager for Data Warehousing and BI





OLAP Analysis

Why do we need OLAP?

Traditional BI: ROLAP

- Power

- Highly scalable
- Level based hierarchies
- Drill, slice, dice
- Timely
- Attribute-based analysis

- Pain

- Parent-child “force fit”
- Column-based analytics only
- Poor Performance with large datasets

Traditional BI: OLAP

- Power

- Flexible, member-base analytics
- Parent-child & Level Based
- Expand/collapse outline
- Fast
- What-if, plan, consolidate, forecast, etc.

- Pain

- Transaction detail and scale
- Limited Attributes



Real Benefits of Using OLAP

- No Summary Tables required
- Reduced repository and maintenance time
- Development cycles times are reduce
 - Testing of design changes take less time
 - Change are easier to make and faster to implement
- Significantly Faster Performance!
 - Recent deployments performed 50 to 300% faster
 - If calculations are done in OLAP server performance is even better
- **IT's NOT AS HARD AS YOU THINK!**



The Hard Stuff

There are several things that are hard for relational OLAP to do or are very slow:

Unique Counts (count distinct)

Time Series Analysis

Measure Hierarchies



Unique Counts

Performing unique counts over dimensional data can be slow and complicated:

“I want to see unique registrations for my web store by day, month, year across my storefronts and departments.”

This is NOT a simple Additive Measure!

You must perform a count(distinct) starting at the bottom and computing the value for every combination of dimensions and levels.



How would you do it in SQL?

Well you need **ONE QUERY** at **MONTH Level...**

COUNT(distinct BENEF_ID)...GROUP BY Month

```
Oracle SQL*Plus
Fichier  Edition  Recherche  Options  Aide
SQL>
SQL>
SQL> SELECT AIDE_ID, CANCOM3_ID, TYPEVEN_ID, MOIS_ID, count (distinct BENEF_ID ) FROM COG.COMPTAGE
where AIDE_ID = '122129' and CANCOM3_ID = 'C0-34057' and TYPEVEN_ID = '1335500' group by AIDE_ID, CA
NCOM3_ID, TYPEVEN_ID, MOIS_ID order by AIDE_ID, CANCOM3_ID, TYPEVEN_ID, MOIS_ID;
```

AIDE_ID	CANCOM3_ID	TYPEVEN_ID	MOIS_I	COUNT(DISTINCT(BENEF_ID))
122129	C0-34057	1335500	200101	1
122129	C0-34057	1335500	200102	1
122129	C0-34057	1335500	200103	1
122129	C0-34057	1335500	200104	1
122129	C0-34057	1335500	200105	2
122129	C0-34057	1335500	200106	3
122129	C0-34057	1335500	200107	3
122129	C0-34057	1335500	200108	3
122129	C0-34057	1335500	200109	3
122129	C0-34057	1335500	200110	4
122129	C0-34057	1335500	200111	4
122129	C0-34057	1335500	200112	4



How would you do it in SQL?

.... and **ANOTHER DIFFERENT QUERY** at **YEAR Level...**

COUNT(distinct BENEf_ID)...GROUP BY SUBSTR(Month,1,4)

```
Oracle SQL*Plus
Fichier Edition Recherche Options Aide
SQL>
SQL> SELECT AIDE_ID, CANCOM3_ID, TYPEVEN_ID, SUBSTR(MOIS_ID, 1, 4), count (distinct BENEf_ID) FROM C
OG.COMPTAGE where AIDE_ID = '122129' and CANCOM3_ID = 'C0-34057' and TYPEVEN_ID = '1335500' group by
AIDE_ID, CANCOM3_ID, TYPEVEN_ID, SUBSTR(MOIS_ID, 1, 4) order by AIDE_ID, CANCOM3_ID, TYPEVEN_ID, SU
BSTR(MOIS_ID, 1, 4);
```

AIDE_ID	CANCOM3_ID	TYPEVEN_ID	SUBS	COUNT(DISTINCTBENEf_ID)
122129	C0-34057	1335500	2000	1
122129	C0-34057	1335500	2001	4
122129	C0-34057	1335500	2002	3
122129	C0-34057	1335500	2003	4
122129	C0-34057	1335500	2004	5
122129	C0-34057	1335500	2005	4
122129	C0-34057	1335500	2006	4
122129	C0-34057	1335500	2007	4
122129	C0-34057	1335500	2008	4
122129	C0-34057	1335500	2009	3
122129	C0-34057	1335500	2010	3

11 ligne(s) sélectionnée(s).



How would you do this in SQL?

...WORST:

You only de-duplicated on **ONE SINGLE** dimension here but you have to do it on **ALL** dimensions at the same time !

➔ Need to write a **specific query for each level combination** across all dimensions **!!!**

YEAR	/	DISTRICT	/	SOCIAL ALLOC. TYPE	/	EVENT TYPE	➔ 1 Query
YEAR	/	REGION	/	SOCIAL ALLOC. TYPE	/	EVENT TYPE	➔ 1 Query
MONTH	/	DISTRICT	/	SOCIAL ALLOC. TYPE	/	EVENT TYPE	➔ 1 Query
MONTH	/	REGION	/	SOCIAL ALLOC. TYPE	/	EVENT TYPE	➔ 1 Query
...	

➔ Hard to maintain

➔ Low performance **THIS CAN TAKE MINUTES TO RUN!**



Doing it in Oracle OLAP

Challenge!

- Oracle OLAP does not yet have a count distinct function!
- But there is a way to do this!

Doing this in Oracle OLAP

The screenshot shows the Oracle OLAP configuration interface. The left sidebar lists various components like OLAP 11g, Analytic Workspaces, OLAPSAMPLE (attached), Dimensions, Cubes, and OLAP_COUNT_CL. The main panel has tabs for General, Aggregation, Partitioning, Storage, and Materialized Views. The 'Aggregation' tab is active, showing 'Specify the aggregation rules of the cube'. Under the 'Rules' sub-tab, the 'Order and Method' section is visible. It contains a table for 'Aggregation Order and Method:' with 6 rows. The table is highlighted with a red border. Below the table, the 'Aggregation Hierarchies' section shows a list of hierarchies with checkboxes. The 'Select All' and 'Deselect All' buttons are at the bottom right of the hierarchies list.

Order	Dimension	Operator	Based On
1	OLAP_TIME	Maximum	
2	OLAP_EMPLOYEE	Maximum	
3	OLAP_OFFICE	Maximum	
4	OLAP_PRODUCT	Maximum	
5	OLAP_CUST_SEGMENT	Maximum	
6	CUSTOMER	Sum	

Aggregation Hierarchies

Aggregate the cube using selected hierarchies:

- ☒ OLAP_TIME
 - ☒ FISCAL
- ☒ OLAP_EMPLOYEE
 - ☒ ORG
- ☒ OLAP_OFFICE
 - ☒ COMPANY_HIER
 - ☒ GEOGRAPHY_HIER

Select All
Deselect All

SOLUTION:

1. Use Oracle OLAP

2. Agregate with **MAXIMUM** on each dimension to be de-duplicated

3. THEN, agregate with **TOTAL** on the dimension you are counting

Respect that order (Easy with AWM, just use arrows to set the dimension you are counting along in the last position).



Doing it in OLAP

Book1		
	A	B
1	All Offices	
2	Michele Lombardo Group	
3	All Segments	
4	All Customers	
5		
6		Distinct Customers
7		+ All Products
8	- All Years	1,000
9	+ 2008	996
10	- 2009	991
11	+ 2009 HY1	952
12	- 2009 HY2	948
13	- 2009 Q3	908
14	2009 / 07	746
15	2009 / 08	632
16	2009 / 09	548
17	+ 2009 Q4	712
18	+ 2010	999
19	+ 2011	

- Query returns results **quickly** (pre-solved)
- Returns in seconds not minutes!
- Can be used in Excel, OBIEE, other SQL reporting tools.

Data at QTR level is not the total of data at MONTH level!



Unique Count Measures



Time Series Analysis

Answer this question:

Show me the the current sales for this month, sales year to date, sales same period last year and sales year to date last year.



How would we do that in SQL?

- Very long complex select statement!

```
WITH sales_dense AS
  (SELECT [breakout columns]
    sales,
    SUM(sales) over(PARTITION BY [breakout columns]
      ORDER BY [time column] ASC range BETWEEN unbounded
        preceding AND CURRENT ROW) AS sales_ytd
  FROM
    (SELECT [breakout columns]
      a.sales
    FROM
      (SELECT [breakout columns]
        SUM(f.sales) sales
      FROM [table list]
      WHERE [star join and other filters]
      GROUP BY [breakout columns])
    a PARTITION BY(breakout columns)
  RIGHT OUTER JOIN
    (-- need list of all time periods
    SELECT DISTINCT [time columns]
    FROM time_dim
    b ON([join on relevant time level]))
  ) ...
Continued...
```

- Performs poorly!



What would it look like from OLAP

- Query from OLAP Cubes:

```
SELECT [breakout columns],  
       sales,  
       sales_prior_year  
       sales_ytd,  
       sales_ytd_prior_year  
FROM sales_cube_view  
WHERE [star join]
```

- OLAP can do many more Complex Time Series Analysis with relatively low cost!
- Returns results in seconds! For ALL LEVELS of TIME



Time Series Analysis



Measure Hierarchies

- Several Applications require that Measures have rollups and hierarchies.
- Oracle OLAP Cube Measures are not Hierarchical
- How do we do something like this?

	2008	2009	2010	Grand Total
[-] Measures	336,227	782,819	783,045	1,902,092
[-] Gross Margin	336,227	782,819	783,045	1,902,092
[-] Net Revenue	21,579,338	21,184,025	21,638,729	64,402,092
Gross Revenue	23,500,000	23,000,000	23,500,000	70,000,000
Discount Amount	1,920,662	1,815,975	1,861,271	5,597,908
Units	2,082,264	1,793,724	1,781,233	5,657,221
[-] Net Costs	21,243,111	20,401,206	20,855,684	62,500,000
Fixed Costs	9,033,594	8,581,538	8,884,869	26,500,000
Variable Costs	12,209,517	11,819,668	11,970,816	36,000,000

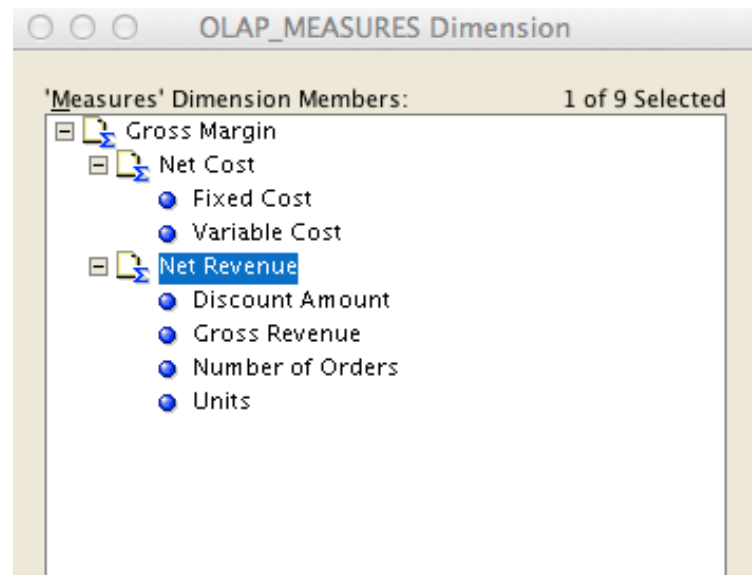


Measure Hierachies in Oracle OLAP

- To do this in Oracle OLAP is very easy!
- Just define measures as a Dimension and establish parentage and rollup rules just like any other dimension!

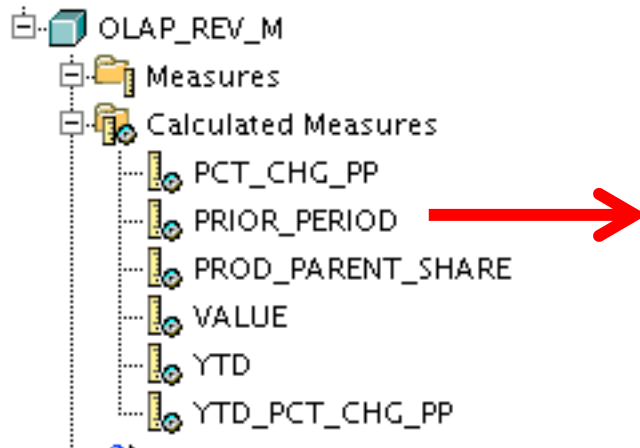
Measure Hierarchies in Oracle OLAP

- Measure Dimension can be value based or level based:



Measure Hierarchies in Oracle OLAP

- Cube can be stored or formula based.
- Example of Formula Based:



General

Specify General Calculated Measure Information

Name: PRIOR_PERIOD

ID: BISAMPLE.OLAP_REV_M.PRIOR_PERIOD

Short Label: Prior Period

Long Label: Prior Period

Description: Prior Period

Calculation Type: Prior Period

Calculation:

Expression:
LAG(OLAP_REV_M.VALUE, 1) OVER HIERARCHY (OLAP_TIME.FISCAL)



Measure Hierachies in Oracle OLAP

The Results:

	[-] Total Products			
		[+] BizTech	[+] FunPod	[+] HomeView
[-] Gross Margin	1,902,092	-1,878,401	1,864,090	1,916,402
[-] Net Cost	62,500,000	25,130,497	18,938,006	18,431,498
Fixed Cost	26,500,000	10,652,422	8,025,442	7,822,136
Variable Cost	36,000,000	14,478,075	10,912,564	10,609,362
[-] Net Revenue	64,402,092	23,252,096	20,802,096	20,347,900
Discount Amount	5,597,908	2,247,904	1,697,904	1,652,100
Gross Revenue	70,000,000	25,500,000	22,500,000	22,000,000
Number of Orders	71,000	28,474	21,587	20,939
Units	5,657,221	2,223,811	1,695,983	1,737,427



Measure Hierachies in Oracle OLAP

Complex Rollups

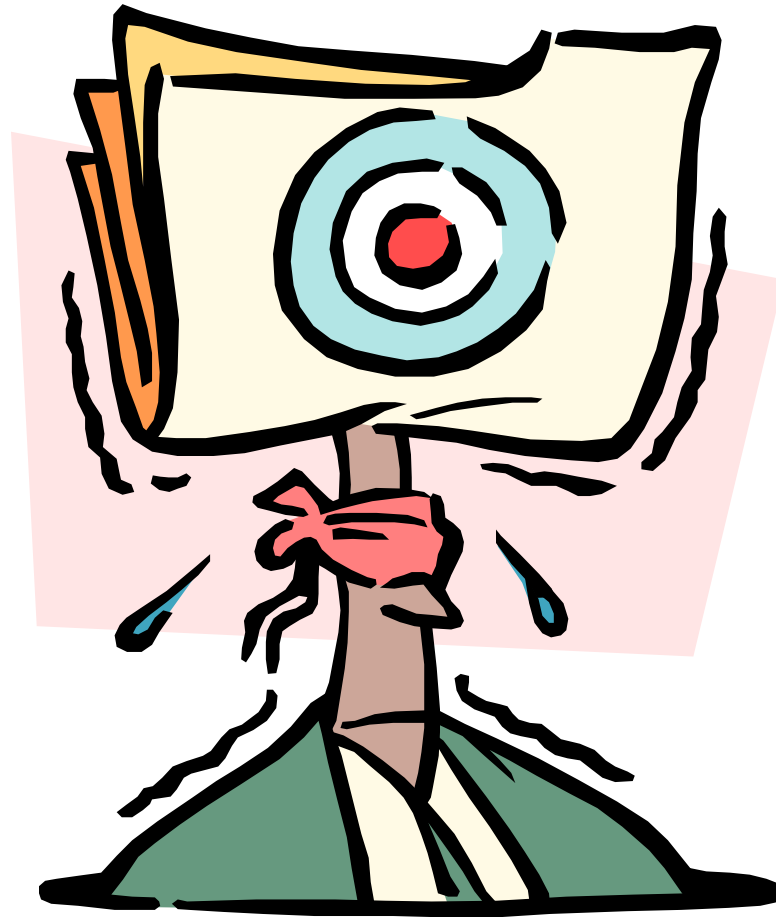
- What is simple aggregations will not satisfy the needs?
- Using Oracle OLAP Model feature complex calculation and aggregation rules can be defined to handle any requirements.



Measure Hierachies



Questions





Oracle Test Drive

- Free to try out Oracle BI
- Go to www.vlamis.com/testdrive-registration/
- Runs off of Amazon AWS
- Hands-on Labs based on Collaborate 2012 HOLs
- Test Drives for:
 - Oracle BI
 - BI Publisher
 - Microsoft Excel against Oracle OLAP
 - Oracle Data Mining
 - Map Views in OBIEE
- Once sign up, you have private instance for 5 hours
- Available now



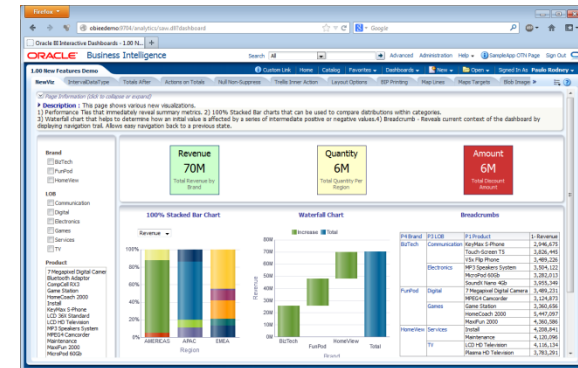
BIWA OBIEE 11.1.1.7 April 18 TechCast

Back by popular demand:

What's New in Oracle BI 11.1.1.7 BIWA TechCast with **live demonstration**

Topic Overview

- Major new release with lots of features (tick of tick-tock cycle)
- New data visualizations and wizard
- New hierarchical column and group display choices
- Freeze headers/scroll bars for tables and pivot tables
- New dashboard layout options for multiple resolutions and mobile
- New dashboard and analyses templating, printing, and saving capabilities
- New Navigation Trail capabilities (bread crumbs)
- Integration with Endeca search engine
- New BI Mobile capabilities and features
- New BI Publisher layout enhancements, integration with dashboards, and data model features
- View and manipulate BI content in SmartView in Excel
- Enhanced export capabilities to MS Office and Excel
- New Hadoop integration with Native HiveQL
- Extended Essbase capabilities with OBI
- New Oracle R Enterprise (ORE) integration and capabilities
- Exalytics enhancements
- See <http://tinyurl.com/BIWA-april18> for more information and to register
- **April 18, 2013 at 11am Central time**





Vlami Collaborate Presentations

Presenter	Session	Time	Title
Tim Vlami	OAUG	Mon 2:30 – 3:30	12633 Data Visualization Best Practices in Oracle Business Intelligence Applications
Tim Vlami	IOUG	Tues 2:00 – 3:00	726 Advanced Dashboard Design in OBI 11g
Dan Vlami	IOUG	Wed 8:15 – 9:15	915 Using Map Views and Geospatial Analytics in OBI 11g
Dan Vlami	IOUG	Wed 3:00 – 4:00	785 Blazing BI: The Analytic Options to the Oracle Database
Chris Claterbos	IOUG	Wed 4:15 – 5:15	732 Advanced OLAP: Making the Hard Stuff Easy
Cathye Pendley	IOUG	Wed 4:15 – 5:15	798 Vlami Process and Maturity Model: BI Project Best Practices
Chris Claterbos	OAUG	Thurs 12:15 – 1:15	12837 Mobile BI: Using When and Where You Need It

Advanced OLAP: Making the Hard Stuff Easy!

Collaborate 2013

IOUG Presentation #732
Chris Claterbos

Vlamis Software Solutions
816-781-2880
<http://www.vlamis.com>