

ORACLE®



ORACLE®

Новые возможности Oracle OLAP 11g

Андрей Пивоваров
Старший менеджер по технологическому консалтингу
Oracle CIS

ORACLE®

Аналитическая платформа Oracle (2010)

УПРАВЛЕНИЕ ЭФФЕКТИВНОСТЬЮ ДЕЯТЕЛЬНОСТИ

Hyperion Planing

Hyperion Financial
Mng

Strategic
Finance

Scorecard

АНАЛИТИЧЕСКИЕ ПРИЛОЖЕНИЯ BI APPLICATIONS

Marketing

Finance

Sales

Supply
Chain

Contact
Center

HR

ИНСТРУМЕНТЫ БИЗНЕС-АНАЛИЗА

Oracle BI Suite EE **PLUS**

Oracle BI Suite SE ONE

Oracle BI SE

ХРАНИЛИЩА И ВИТРИНЫ ДАННЫХ

Oracle
Warehouse Builder

Oracle
Data Integrator

ORACLE
DATABASE

Oracle Partitioning

Oracle Data Mining

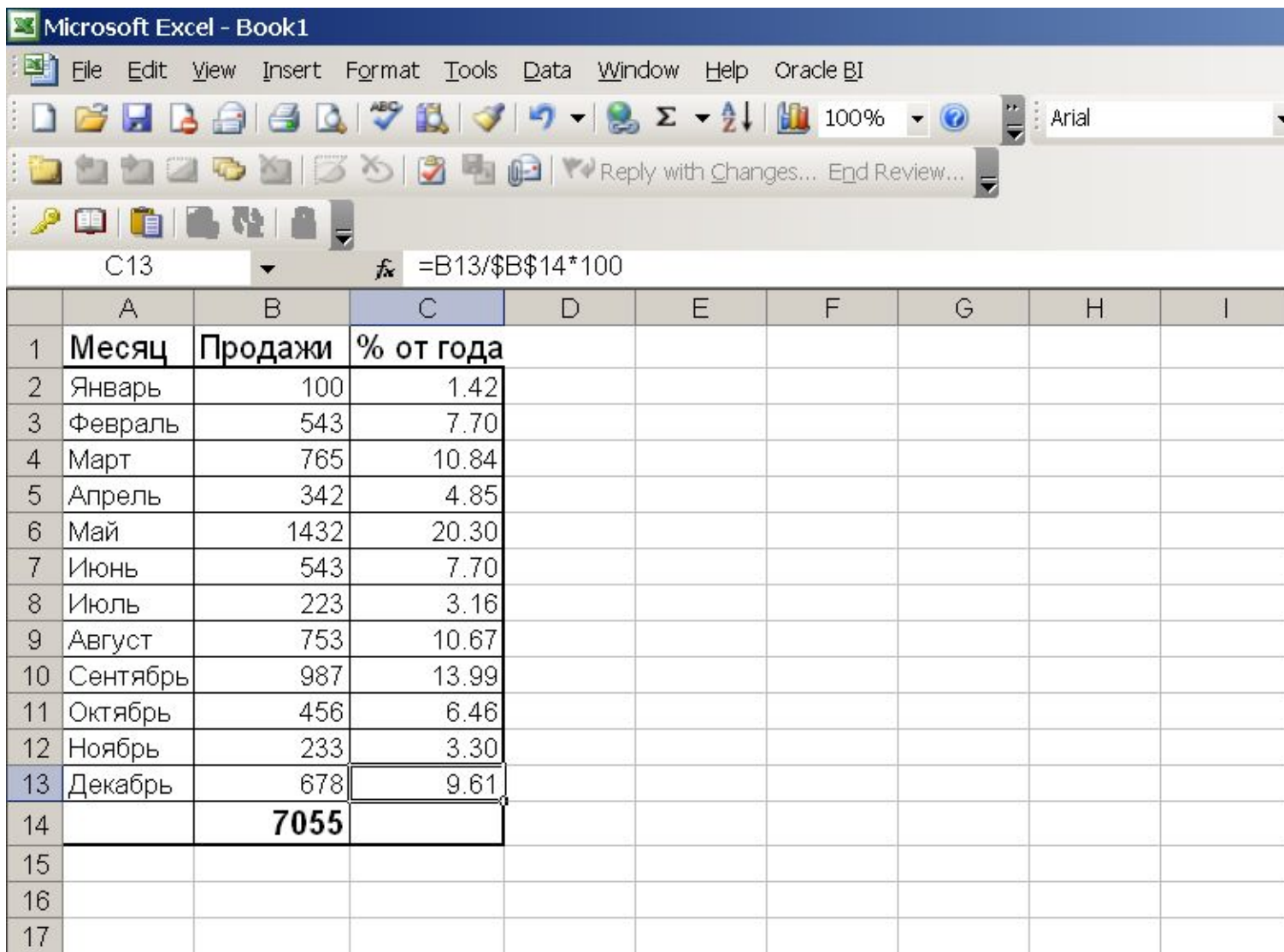
Oracle OLAP

GoldenGate

HYPERION
ESSBASE

ORACLE

Почему популярен Excel?



The screenshot shows the Microsoft Excel interface with the following data in the spreadsheet:

	A	B	C	D	E	F	G	H	I
1	Месяц	Продажи	% от года						
2	Январь	100	1.42						
3	Февраль	543	7.70						
4	Март	765	10.84						
5	Апрель	342	4.85						
6	Май	1432	20.30						
7	Июнь	543	7.70						
8	Июль	223	3.16						
9	Август	753	10.67						
10	Сентябрь	987	13.99						
11	Октябрь	456	6.46						
12	Ноябрь	233	3.30						
13	Декабрь	678	9.61						
14		7055							
15									
16									
17									

The formula bar shows the formula for cell C13: $=B13/\$B\$14*100$.

Вычисления внутри строки на SQL

```
SQL> select t.empno,t.ename,t.sal,t.comm,  
sal+nvl(comm,0) summ  
from emp t;
```

EMPNO	ENAME	SAL	COMM	SUMM
7369	SMITH	800,00		800
7499	ALLEN	1600,00	300,00	1900
7521	WARD	1250,00	500,00	1750
7566	JONES	2975,00		2975
7654	MARTIN	1250,00	1400,00	2650
7698	BLAKE	2850,00		2850
7782	CLARK	2450,00		2450
7788	SCOTT	3000,00		3000
7839	KING	5000,00		5000
7844	TURNER	1500,00	0,00	1500
7876	ADAMS	1100,00		1100

Вычисления между строк на SQL

```
select
  (select sal
   from emp
   where empno=7499)
-
  (select sal
   from emp
   where empno=7521) diff
from dual;
```

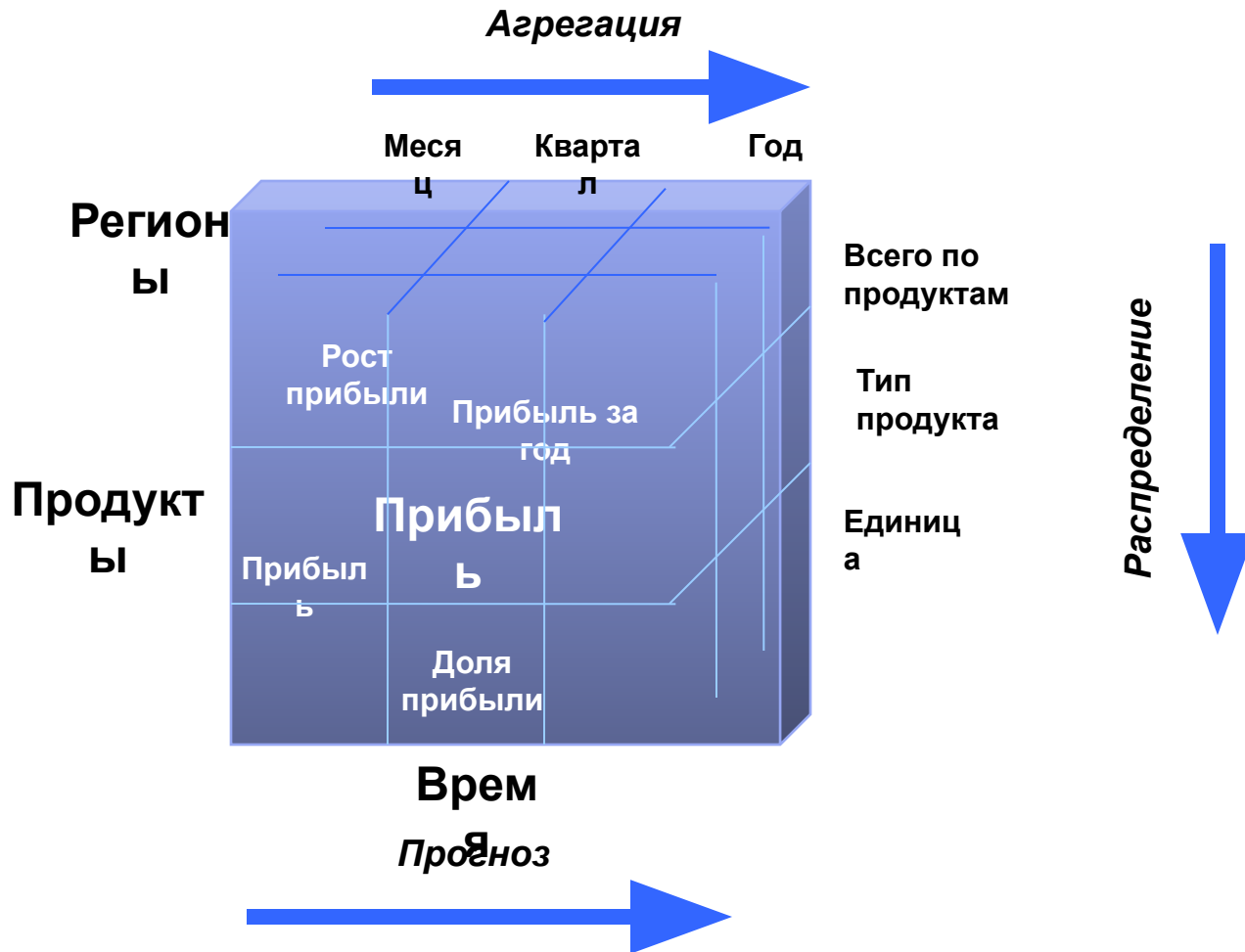
```
DIFF
-----
350
```


Запрос над OLTP

```
select_oltp.sql - Notepad
File Edit Format View Help

SELECT fy,customer,sum(lic) from
(customer, customer_id, orcode, agrcode, channel, channel_id, flsc, flsc_id, city, region_id, country, iso_from, month, status, cy, fy, fq, prob, product, product_id, group_id, product_type,
platform, platform_id, computer, num_comp, units, vlip_id, unit_type, usage, support_start, support_end, lic_price, sup_price, ts_new, ts_renew, comp_location, lic_disc, sup_disc, respons,
com_ups, null_value, sal_comm, sup_comm, fls_comm)
AS
SELECT
PMS_VORGANIZATIONS.SHORT_NAME
,PMS_VORGANIZATIONS.ID
,vbok.puo_orcode
,vbok.agr_agrcode
,PMS_VORGANIZATIONS1.SHORT_NAME
,PMS_VORGANIZATIONS1.ID
,PMS_VORGANIZATIONS2.SHORT_NAME
,PMS_VORGANIZATIONS2.ID
,CTY_NAME
,CTY_REG_ID
,CTR_NAME
,VBOK.CTY_ISO_CODE_FROM
,VBOK.MONTH
,vbok.status
,substr(to_char(VBOK.MONTH,'YY'),1,2)
,to_char(mod(to_number(decode(sign(to_number(to_char(VBOK.MONTH,'MM')))-5),
1,to_char(to_number(to_char(VBOK.MONTH,'YY'))+1),
to_char(VBOK.MONTH,'YY')),100),'00')
,to_char(mod(to_number(decode(to_char(VBOK.MONTH,'MM'),'01',to_char(VBOK.MONTH,'YY'))||'.3',
'02',to_char(VBOK.MONTH,'YY'))||'.3',
'03',to_char(VBOK.MONTH,'YY'))||'.4',
'04',to_char(VBOK.MONTH,'YY'))||'.4',
'05',to_char(VBOK.MONTH,'YY'))||'.4',
'06',to_char(to_number(to_char(VBOK.MONTH,'YY'))+1)||'.1',
'07',to_char(to_number(to_char(VBOK.MONTH,'YY'))+1)||'.1',
'08',to_char(to_number(to_char(VBOK.MONTH,'YY'))+1)||'.1',
'09',to_char(to_number(to_char(VBOK.MONTH,'YY'))+1)||'.2',
'10',to_char(to_number(to_char(VBOK.MONTH,'YY'))+1)||'.2',
'11',to_char(to_number(to_char(VBOK.MONTH,'YY'))+1)||'.2',
'12',to_char(to_number(to_char(VBOK.MONTH,'YY'))+1)||'.3')),100),'00.0')
,vbok.PROBABILITY
,PRO.DESCRPTION
,PRO.ID
,PRO.pgr_id
,PRO.PRODUCT_TYPE
,PLA.DESCRPTION
,PLA.PLATFORM
,CMD.DESCRPTION
,round(VLIP.COMPUTERS
*decode(NVL(sum(VBOK1.AMOUNT),0),0,0,NVL(VBOK.AMOUNT,0)/sum(VBOK1.AMOUNT))
)
--*decode(sign(NVL(sum(VBOK1.AMOUNT),0)),1,decode(sign(VBOK.AMOUNT),1,1,0),sign(VBOK.AMOUNT))
,round(VLIP.UNITS
--!!!--*decode(NVL(sum(VBOK1.AMOUNT),0),0,0,NVL(VBOK.AMOUNT,0)/sum(VBOK1.AMOUNT))
)
--*decode(sign(NVL(sum(VBOK1.AMOUNT),0)),1,decode(sign(VBOK.AMOUNT),1,1,0),sign(VBOK.AMOUNT))
,VLIP.ID
,VLIP.UNIT_TYPE
,VLIP.USAGE
,vpuo.SUPPORT_START
,vpuo.SUPPORT_END
,decode(PRO.PRODUCT_TYPE,'SOFTWARE',
(1+decode(NVL(vpuo.license_price,0),0,0,
decode(sign(NVL(vpuo.license_price,0)-NVL(vpuo.transfer_credit,0)),-1,
-NVL(vpuo.license_price,0)+NVL(vpuo.transfer_fee,0),
-NVL(vpuo.transfer_credit,0)+NVL(vpuo.transfer_fee,0))/vpuo.license_price))
*(1-NVL(vpuo.DISCOUNT,0)/100),1)
*decode(NVL(vpuo.license_price,0),0,
decode(NVL(vpuo.transfer_fee,0),0,0,
decode(NVL(vpuo.ts_price,0),0,0,
vpuo.transfer_fee*NVL(vlip.support_price,0)*NVL(vlip.units,0)*(1-nvl(vlip.support_discount,0)/100)/vpuo.ts_price)),
NVL(vlip.units,0)*NVL(vlip.license_price,0)*(1-nvl(vlip.license_discount,0)/100))
*decode(NVL(sum(VBOK1.AMOUNT),0),0,0,NVL(VBOK.AMOUNT,0)/sum(VBOK1.AMOUNT))
--*decode(sign(NVL(sum(VBOK1.AMOUNT),0)),1,decode(sign(VBOK.AMOUNT),1,1,0),sign(VBOK.AMOUNT))
,decode(PRO.PRODUCT_TYPE,'SOFTWARE',
(1+decode(NVL(vpuo.ts_price,0),0,0,-NVL(vpuo.ts_transfer_credit,0)/vpuo.ts_price))
*(1-NVL(vpuo.ts_discount,0)/100))
*NVL(vlip.support_price,0)*NVL(vlip.units,0)*(1-nvl(vlip.support_discount,0)/100)*
NVL(round(MONTHS_BETWEEN(vpuo.support_end,vpuo.support_start),12)/12)
*decode(NVL(sum(VBOK1.TS_AMOUNT),0),0,0,NVL(VBOK.TS_AMOUNT,0)/sum(VBOK1.TS_AMOUNT))
--*decode(sign(NVL(sum(VBOK1.TS_AMOUNT),0)),1,decode(sign(VBOK.TS_AMOUNT),1,1,0),sign(VBOK.TS_AMOUNT))
,decode(vbok.renuwal,'X',0,decode(PRO.PRODUCT_TYPE,'SOFTWARE',
```


Куб OLAP



Вычисления на Oracle OLAP DML

```
RPR hr_cube_sal(emp_id '7499') - hr_cube_sal(emp_id '7521')
```

(В Excel-e B10=A5-A6)

TIME	SAL
-----	-----
JAN10	350,00
FEB10	350,00
MAR10	450,00

Сравнение OLAP DML и SQL

Задача

Найти товары, по которым за период Октябрь 2009 были продажи на сумму более 100 000 в регионе Северо-Запад

SQL

```
Select p.prod_name, g.geog_name, t.time_name,
f.sales
  from fact f, proddim p, geogdim g, timedim t
 where f.prod_id = p.prod_id and f.geog_id =
g.geog_id
    and t.time_id = f.time_id and g.geog_id = 'NW'
    and t.time_id = 'OCT2009' and sales > 100000
```

OLAP DML

```
limit geography to 'NW'
limit time to 'OCT2009'
limit product to dollars gt 100000
```

SQL Model

- Выборка рассматривается в виде куба
- Введен оператор MODEL
- С помощью модели куб, т.е. выборка, может изменяться и расширяться
- Мощнейшие возможности по формированию сложных вычислений

```
SELECT * FROM  
  sales s  
MODEL  
  DIMENSION BY (Product,Version)  
  MEASURES (units,price,sale)  
  RULES UPSERT  
  (sale['Total','Total'] = SUM(sale)[ANY,ANY])
```

Повышение производительности



Три (М)OLAP сервера Oracle

- Oracle Express Server
 - не развивается, используется, например, с OFA
- Oracle OLAP option
 - опция СУБД Oracle, наследница Express Server
- Oracle Hyperion Essbase
 - появился с приобретением Hyperion

Краткая история Oracle OLAP (1/2)

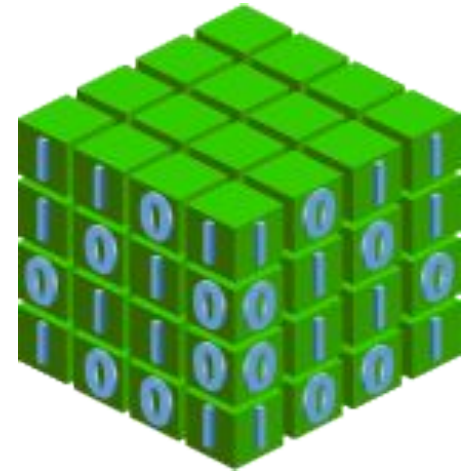
- 1967 – Leonard M. Lodish и John D.C. Little основали Management Decision Systems (MDS)
- **1977 – Software Development Laboratories (SDL), предшественник Oracle. Основатели Larry Ellison, Bob Miner и Ed Oates.**
- **1979 – Выпущена Oracle Version 2 (RDBMS)**
- 1983 – MDS начинает переписывать Express на C с AED.
- 1985 – Information Resources, Inc. (IRI) приобретает MDS. Express полностью переписан на C
- **1993 – E.F. Codd придумал термин OLAP и E.F. Codd, S.B. Codd и C.T. Salley опубликовали работу “Providing OLAP to User-Analysts: An IT Mandate” где сформулировали 12 принципов OLAP**

Краткая история Oracle OLAP (2/2)

- **1995 – Oracle приобретает Express у IRI**
- 1998-2000 – Oracle продолжает развитие Express и в конце 90х начинается работа по интеграции Express в ядро Oracle
- 2002 – Oracle 9.2.0 – Express интегрирован в ядро Oracle и назван OLAP option
- 2004 – 10.1.0 – OLAP добавлены возможности доступа по SQL и секционирование
- 2004 – 10.1.0 – В OLAP option появляется новая запатентованная технология агрегирования и управления разреженными данными
- 2005 – 10.2.0 – первое терабайтное Аналитическое пространство
- 2007 – Выпущена Oracle Database 11g
- **2007 – Oracle покупает Hyperion**
- 2007 – 11.1 – OLAP option интегрирована с механизмом query rewrite для использования кубов в качестве материализованных представлений.

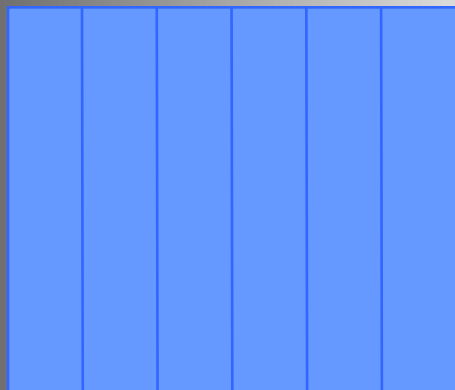
Что такое Oracle OLAP?

- Многомерная СУБД, встроенная в СУБД Oracle (MOLAP)
- Основное назначение – анализ данных
- Построена на основе Oracle Express Server
- Данные хранятся в т.н. Аналитических пространствах (Analytical Workspaces)

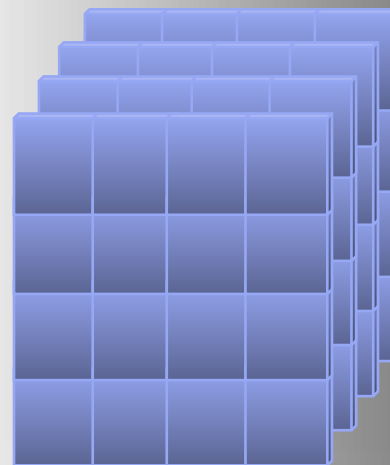


Oracle OLAP и СУБД Oracle

Oracle Database 11g



Реляционные таблицы



Аналитические пространства

Отличительные особенности OLAP

- Работа не только с детальными данными, но и с агрегатами
 - При этом разработчик приложений не думает о том как рассчитывать агрегаты
- Мощные аналитические возможности
- Работа с множеством иерархий, в том числе и Parent-Child
- Возможность записывать факт в произвольное место куба
 - Часто используется в финансовых системах, когда агрегат не равен сумме детальным данным

Когда стоит попробовать использовать OLAP

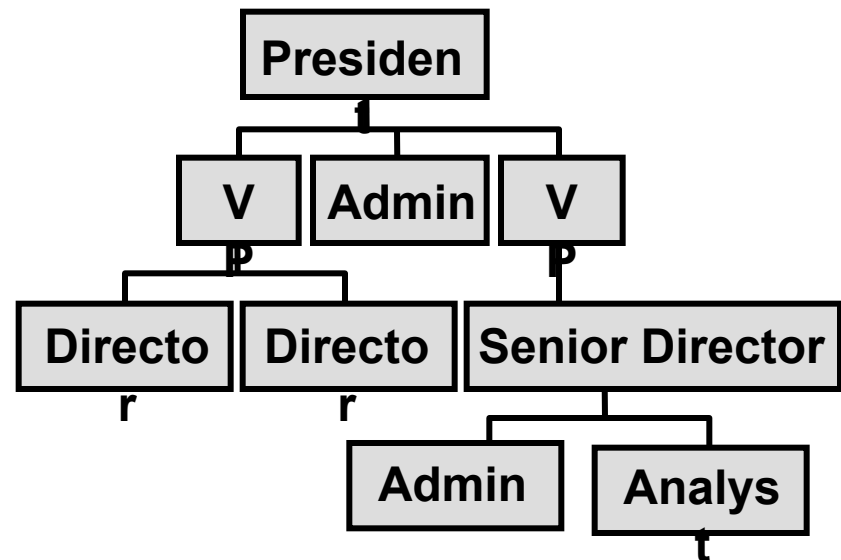
- Когда SQL запросы становятся очень сложными
- Когда есть потребность в сложных вычисляемых показателях
- Когда существует большое количество нерегламентированных запросов.
- Когда производительность запросов становится проблемой

Типы иерархий

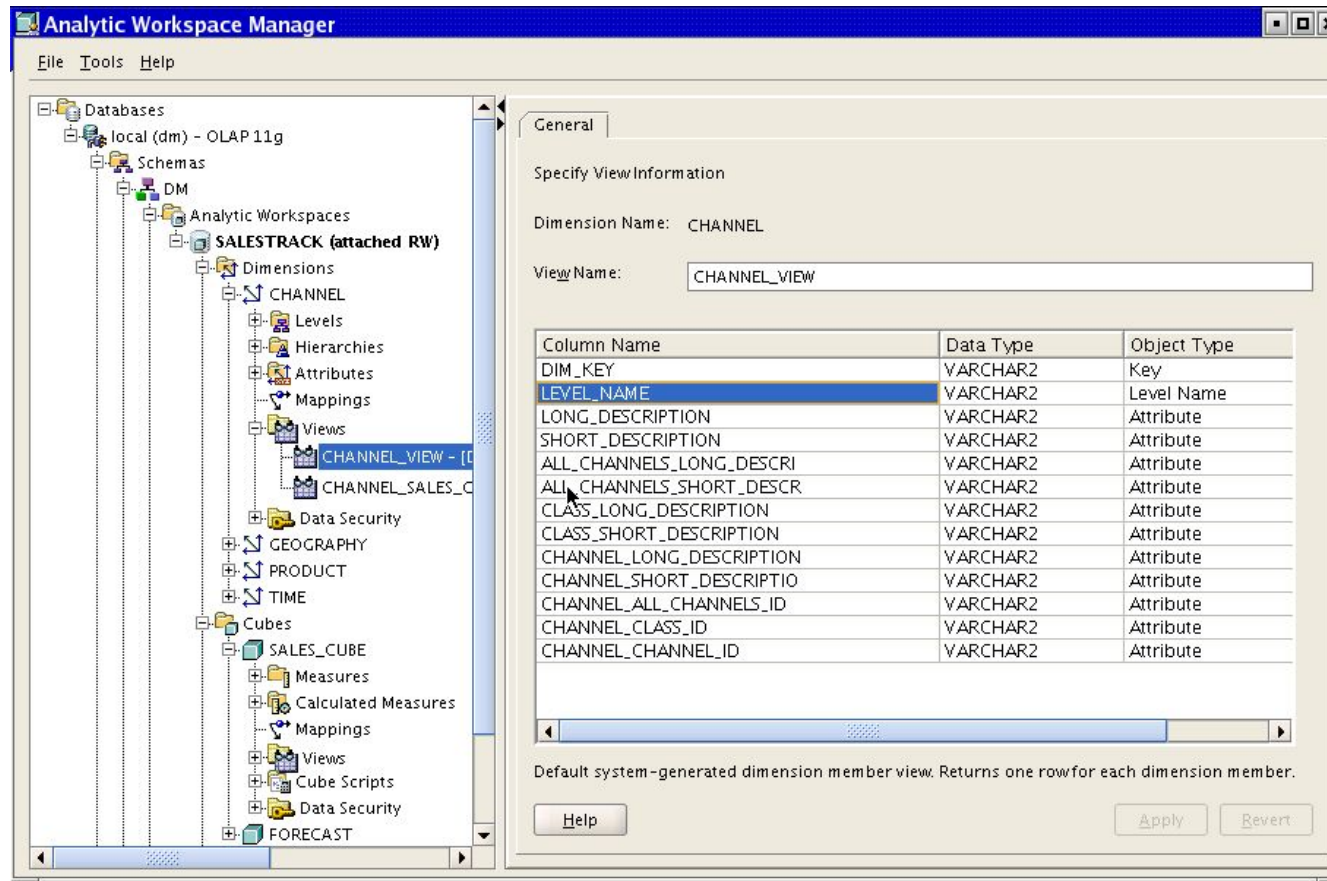
Уровневые



Родитель-потомок (parent-child)



Analytic Workspace Manager

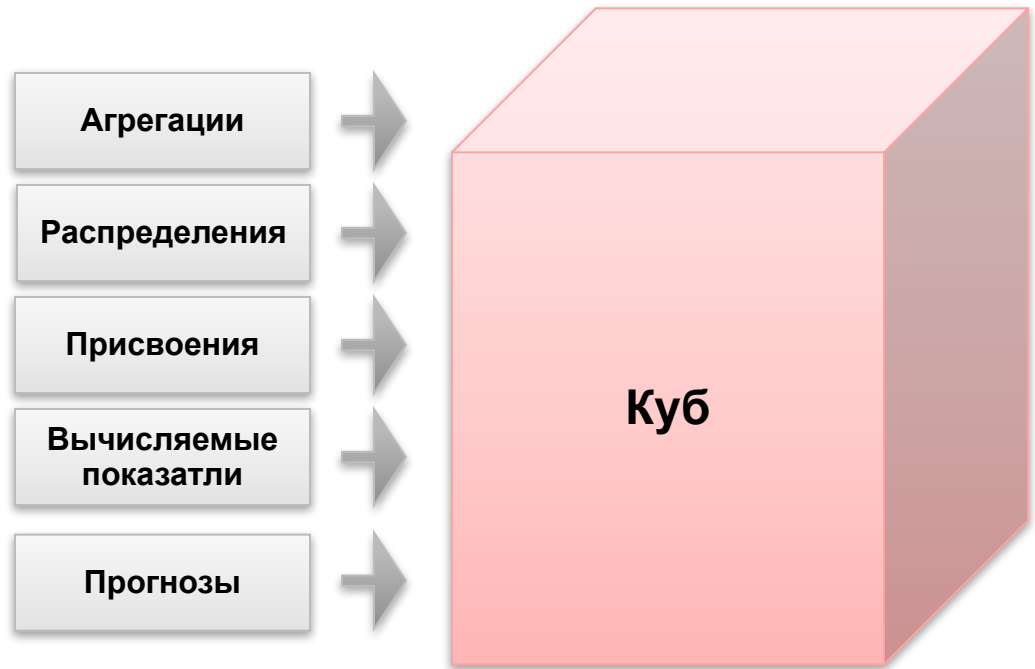


“Enterprise Manager” для OLAP

Некоторые вещи без AWM сделать крайне сложно

Возможности Oracle OLAP

- Данные и правила вычисление интегрированы
- Большая библиотека для вычислений
- Многомерные вычисления



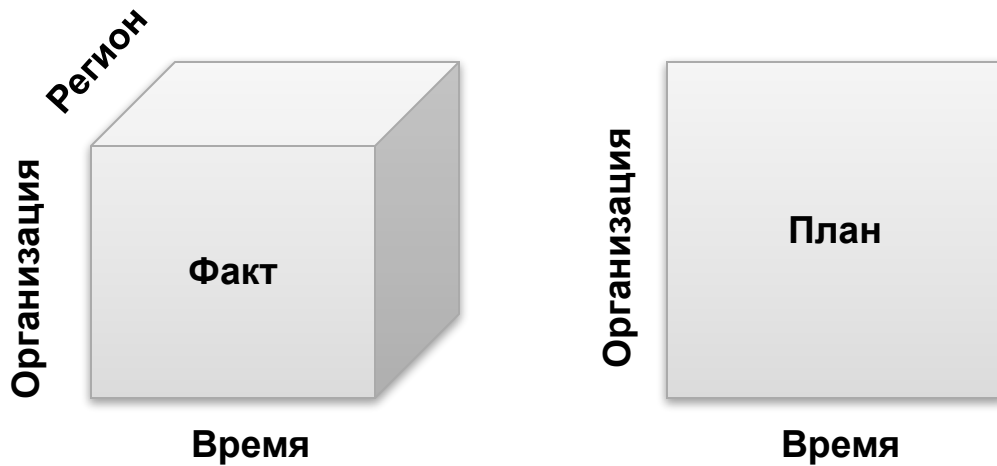
Вычисления

- Вычисления определяются один раз
- Работают по всем измерениям и уровням
- Инструменты просто извлекают результаты

	⊕ CY2008		
	Sales Ytd	Sales Ytd Pr Year	Sales Ytd Pr Yr Pct Chg
⊕ Africa	6,407,291.01	5,605,506.38	14.30
⊕ Asia	65,005,767.54	57,443,041.41	13.17
⊕ Europe	22,315,699.56	19,330,044.80	15.45
⊖ North America	37,216,061.47	32,817,777.74	13.40
Canada	1,215,517.80	1,138,759.26	6.74
Cuba	180,865.59	159,837.61	13.16
Mexico	5,325,717.88	4,641,582.50	14.74
United States	30,493,960.20	26,877,598.37	13.45
⊕ Oceania	110,430.70	94,718.99	16.59
⊕ South America	16,391,834.65	14,363,310.70	14.12

Вычисления между кубами

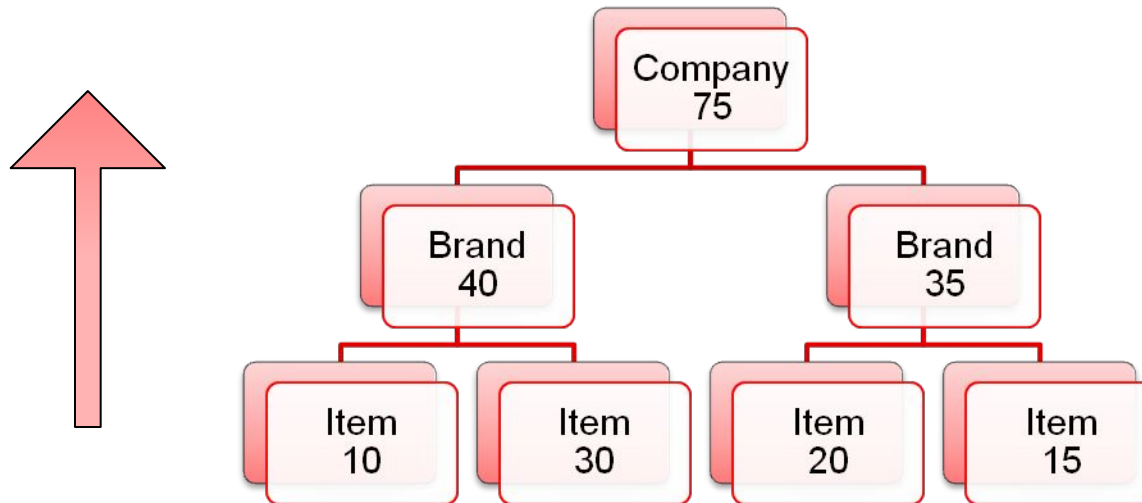
- Вычисления могут использовать данные в других кубах



```
budget_cube.variance =  
actual_cube.actual[geography 'TOTAL'] -  
budget_cube.budget
```

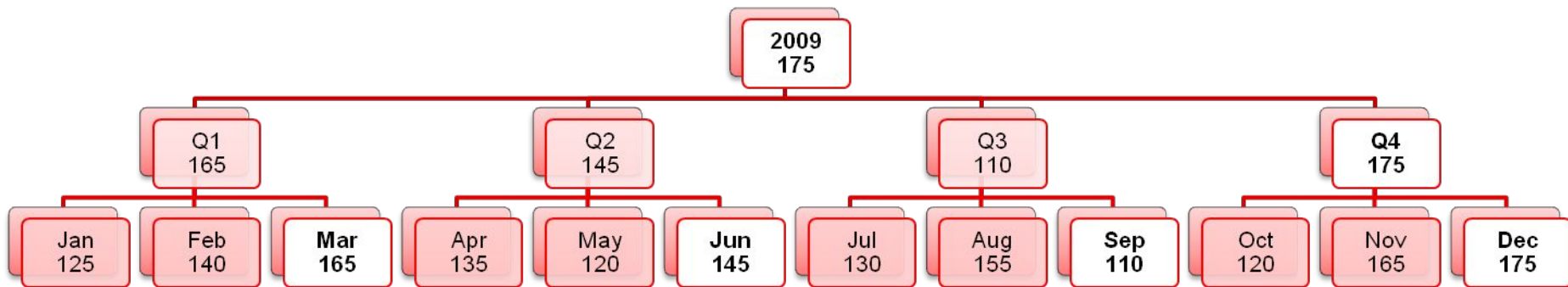
Агрегации

- Агрегации вычисляют значения на верхних уровнях на основе данных нижних уровней



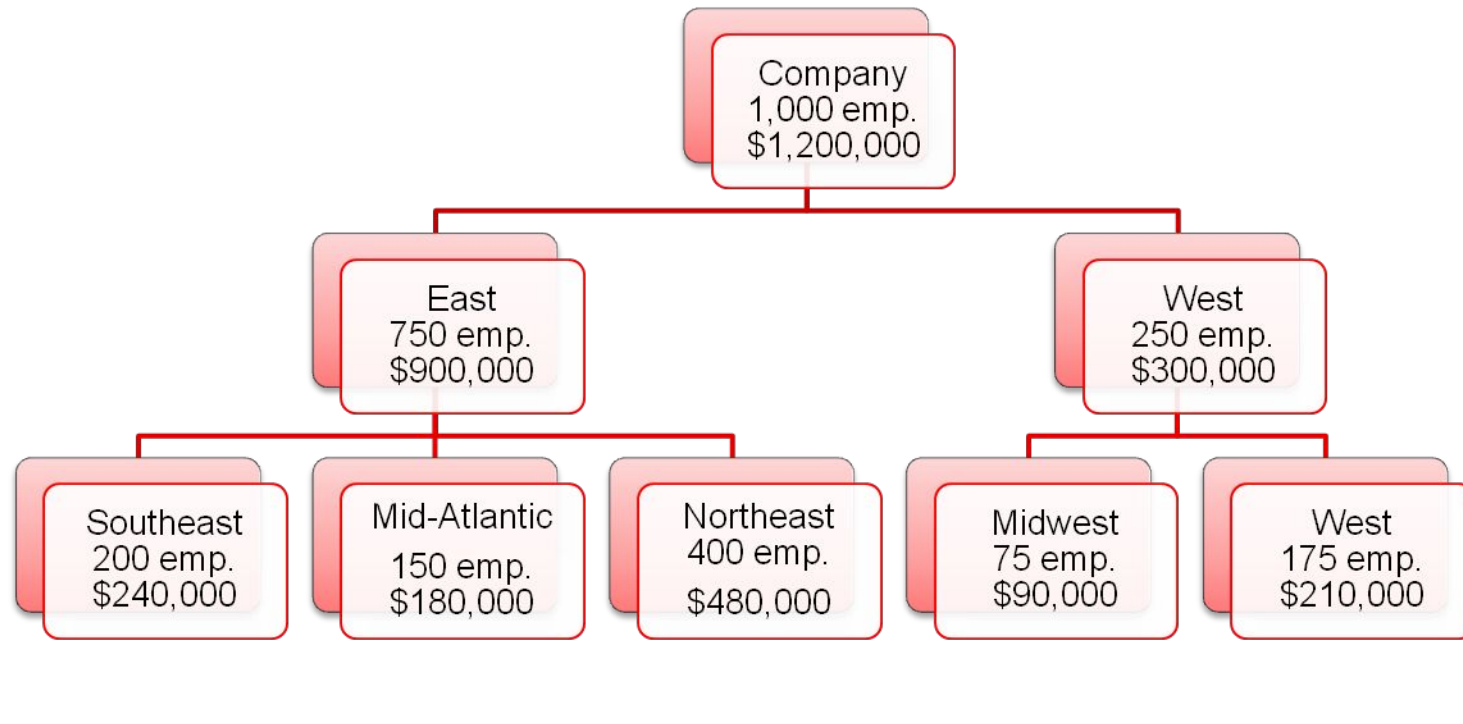
Агрегации

- Пример – Вычисление значения на конец периода



Иерархические распределения

- Распределения вычисляют значения нижних уровней на основе значений верхних и весов



Вычисляемые показатели

```
net_income = operating_income - taxes
operating_income = gross_margin - (marketing +
                                   selling_exp + r_d)
gross_margin = revenue - cogs
```

Вычисляемые показатели

Вычисляемый показатель, демонстрирующий на верхних уровнях наличие проблемы на нижних

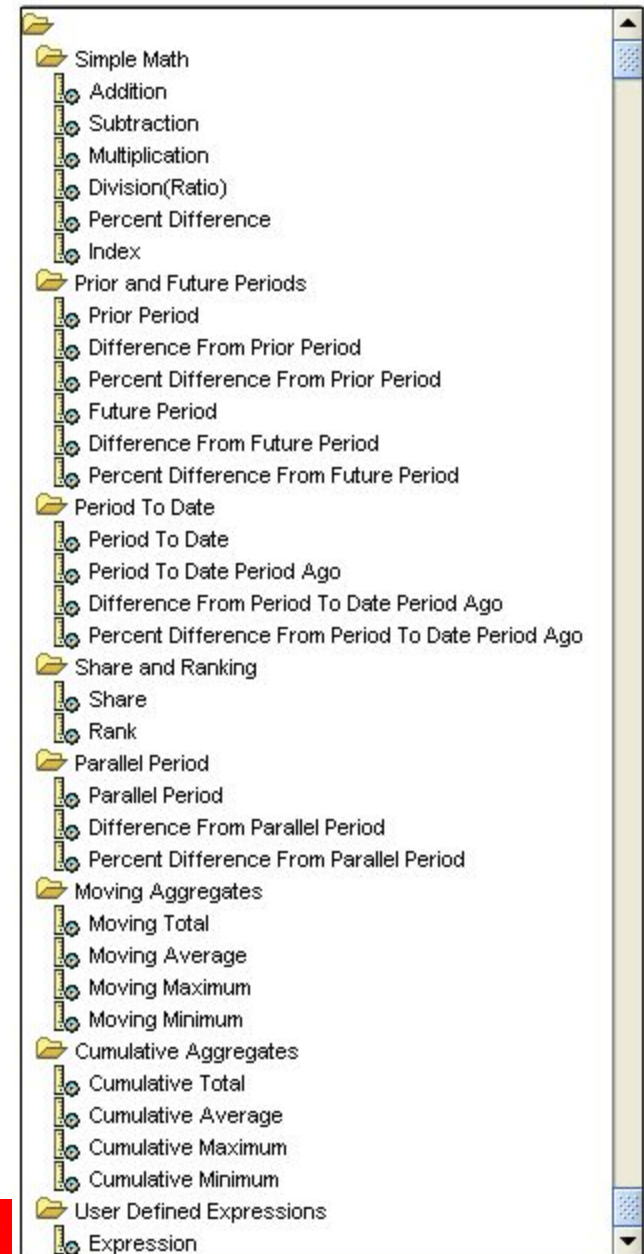
Sales Channel	All Channels ▾
Regional	All Regions ▾

	Product Alert	Sales	Sales Ytd	Sales Ytd Pr Year	Sales Ytd Pr Yr Pct Chg
⊕ CY2008	●	147,447,084.93	147,447,084.93	129,654,400.02	↑ 13.72
⊕ Cameras and Camcorders		12,387,923.49	12,387,923.49	10,657,143.31	↑ 16.24
⊕ Computers	●	115,957,793.25	115,957,793.25	101,669,049.64	↑ 14.05
⊕ Portable Music and Video		19,101,368.19	19,101,368.19	17,328,207.07	↑ 10.23

```
VRB _alert          NUMBER
VRB _product        NUMBER
_product = product + 0
TEMPSTAT product time
DO
  LIMIT product TO DESCENDANTS USING product_parentrel _product
  IF STATLEN(product) EQ 0
    THEN _alert = na
    ELSE DO
      LIMIT product KEEP sales_cube_sales_ytd_pr_yr_pct LT 0
      AND sales_cube_sales_share_tot_pro GT .25
      IF STATLEN(product) GT 0
        THEN _alert = -1
        ELSE _alert = na
      DOEND
    DOEND
  DOEND
RETURN _alert
```

Вычисляемые показатели

- Большое количество шаблонов вычислений
- Можно расширять набор шаблонов



Вычисляемые показатели

- Шаблоны помогают создать аналитическое выражение

Тип вычислений



Описание



Автоматически
генерируемое
выражение



Calculation Type:

Calculation:

Percent difference from [Parallel period](#) for [SALES_YTD \(...\)](#) in the TIME dimension and TIME.CALENDAR hierarchy [1 TIME.CALENDAR.CALENDAR_YEAR](#) ago based on position from [beginning to ending](#) of period.

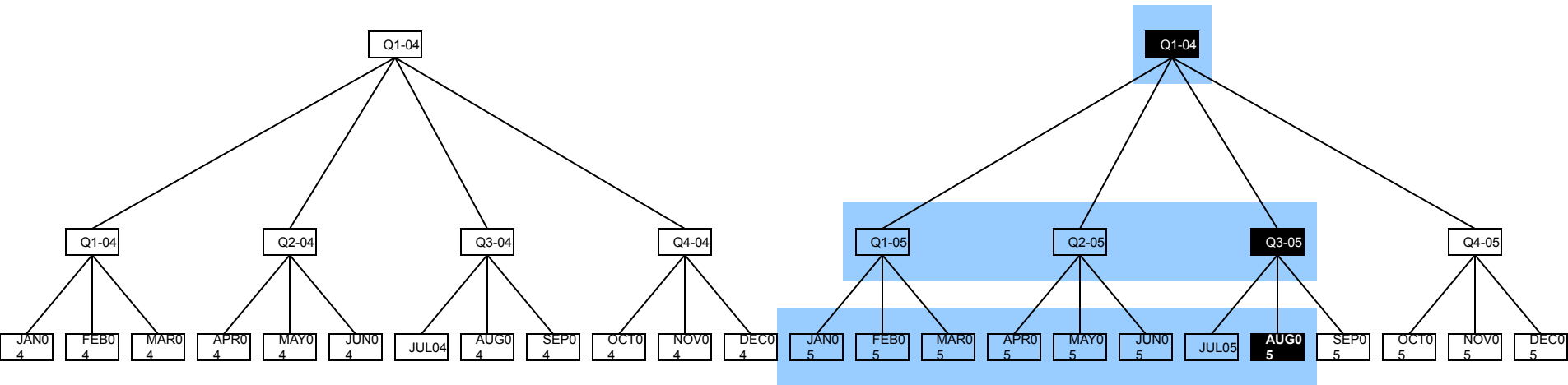
Expression:

```
LAG_VARIANCE_PERCENT(SALES_CUBE.SALES_YTD,1 ) OVER HIERARCHY (TIME.CALENDAR BY ANCESTOR AT LEVEL TIME.CALENDAR.CALENDAR_YEAR POSITION FROM BEGINNING)
```

Percent Change in YTD Sales from Year Ago

Cumulative Total

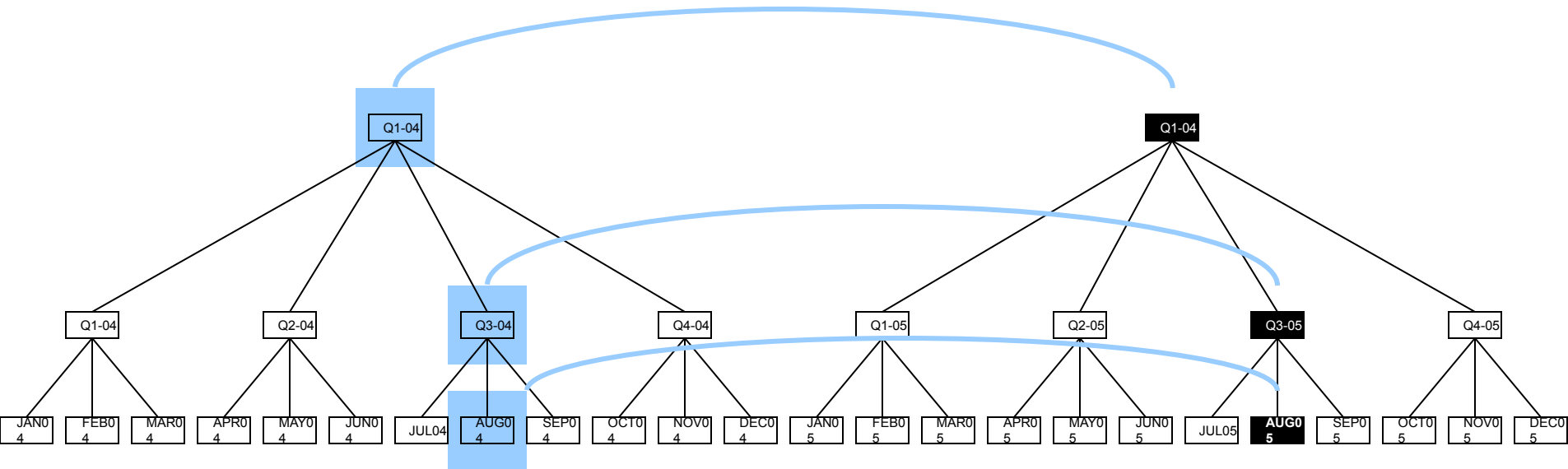
(or AVG, MIN, MAX)



Cumulative total (sum) of [Sales](#) in the [Time](#) dimension and [Calendar Year](#) hierarchy within [ancestor at level calendar_year](#). Total from [beginning](#) member to [current member](#).

```
SUM(sales) OVER HIERARCHY (global.time.calendar_year BETWEEN UNBOUNDED  
PRECEDING AND CURRENT MEMBER WITHIN ANCESTOR AT LEVEL  
global.time.calendar_year)
```

Percent Difference From Parallel Period



Percent difference from [parallel period](#) for [Sales](#) in the [Time](#) dimension and [Calendar Year](#) hierarchy [1 Year ago](#) based on position from the [beginning](#) of Year.

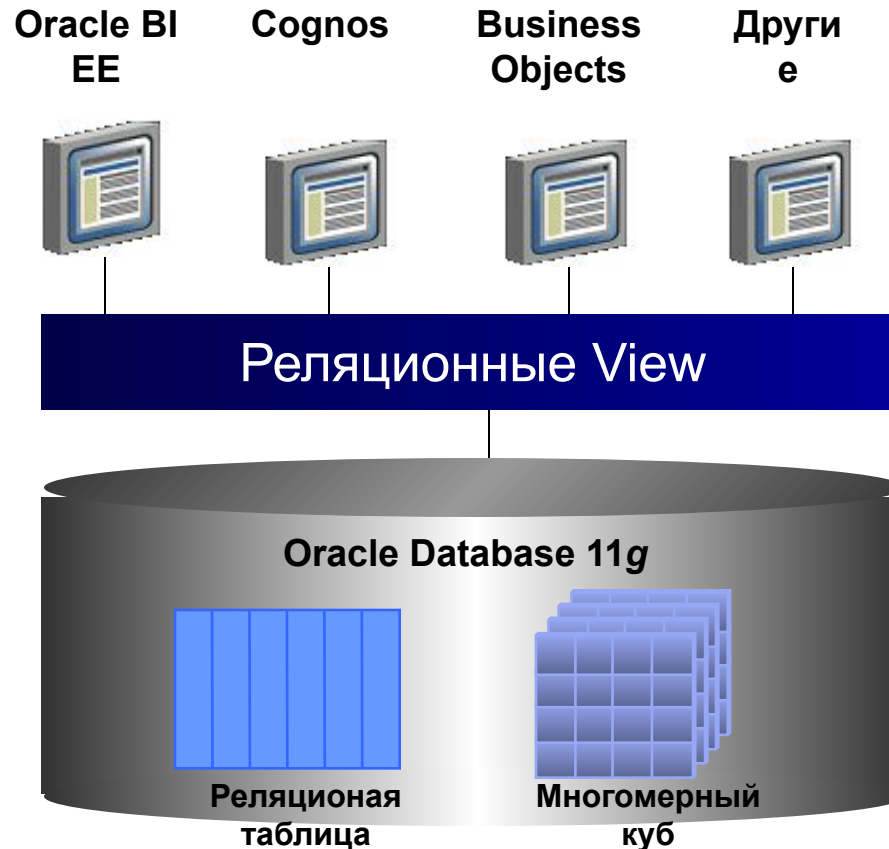
```
LAG_VARIANCE_PERCENT(global.units_cube.sales,1) OVER HIERARCHY
(global.time.calendar_year BY ANCESTOR AT LEVEL global.time.year POSITION
FROM BEGINNING)
```

Присвоения

- Присвоение записывает данные в ячейки куба
- Можно использовать для Что-Если анализа
 - С учетом иерархий и уровней.
 - Пользователь может определять свои правила вычислений
 - Фильтры, ранжирования, временные ряды, более 450 функций

Доступ с помощью любых инструментов

Ускорение для реляционных приложений и отчетных средств



11g Cube Fact View

```
DESC units_cube_view;
```

Name	Null	Type	
-----	-----		
TIME		VARCHAR2 (100)	} Keys
CHANNEL		VARCHAR2 (100)	
CUSTOMER		VARCHAR2 (100)	
PRODUCT		VARCHAR2 (100)	
UNITS		NUMBER	} Stored facts
SALES		NUMBER	
COST		NUMBER	
SALES_PRIOR_PERIOD		NUMBER	} Calculated facts
SALES_DIFF_PRIOR_PERIOD		NUMBER	
SALES_PCT_DIFF_PRIOR_PERIOD		NUMBER	

11g Cube Fact View

```
SELECT time,  
       product,  
       customer,  
       channel,  
       units  
FROM units_cube_view  
WHERE rownum < 15;
```

TIME	PRODUCT	CUSTOMER	CHANNEL	UNITS
TOTAL	TOTAL	TOTAL	TOTAL	4000968
CY1999	TOTAL	TOTAL	TOTAL	330425
CY2003	TOTAL	TOTAL	TOTAL	534069
CY1998	TOTAL	TOTAL	TOTAL	253816
CY2005	TOTAL	TOTAL	TOTAL	565718
CY2006	TOTAL	TOTAL	TOTAL	584929
CY2004	TOTAL	TOTAL	TOTAL	587419
CY2000	TOTAL	TOTAL	TOTAL	364233
CY2002	TOTAL	TOTAL	TOTAL	364965
CY2001	TOTAL	TOTAL	TOTAL	415394
CY2000.Q1	TOTAL	TOTAL	TOTAL	88484
CY2001.Q2	TOTAL	TOTAL	TOTAL	97346
CY2001.Q3	TOTAL	TOTAL	TOTAL	105704
CY2005.Q3	TOTAL	TOTAL	TOTAL	138953

Агрегаты по годам

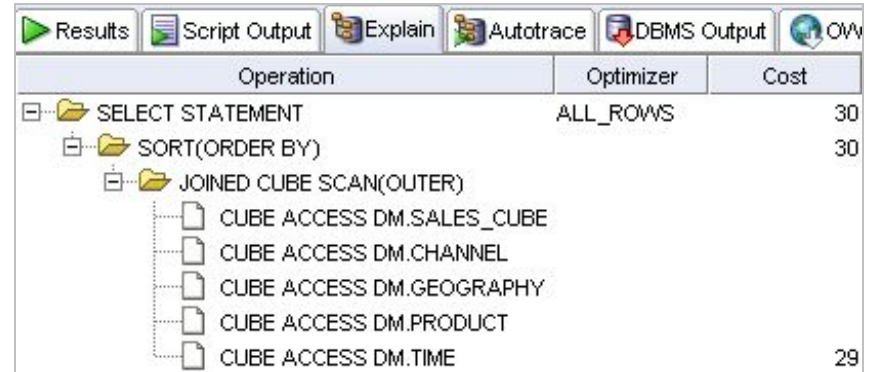
Агрегаты по кварталам

Итого по продуктам и др.
измерениям

Запросы к Cube Views

Explain Plan

```
SELECT
  t.long_description time,
  g.long_description geography,
  c.long_description channel,
  f.sales sales
FROM time_calendar_view t,
     product_standard_view p,
     geography_regional_view g,
     channel_sales_channel_view c,
     sales_cube_view f
WHERE t.dim_key IN('CY2007','Q1CY2007',
                  'FEB2007')
      AND p.dim_key = 'ALL_PRODUCTS'
      AND g.level_name = 'REGION'
      AND c.dim_key = 'INTERNET'
      AND t.dim_key = f.TIME
      AND p.dim_key = f.product
      AND g.dim_key = f.geography
      AND c.dim_key = f.channel
ORDER BY t.end_date;
```

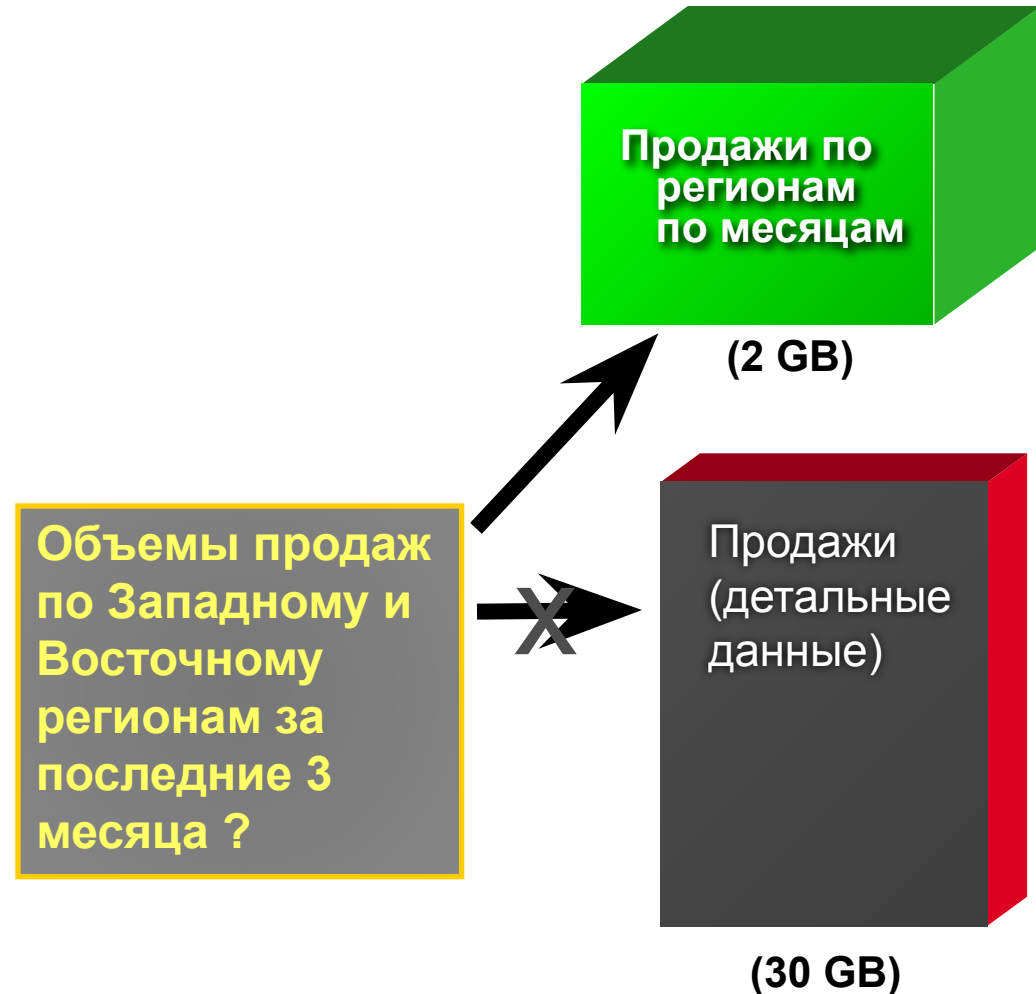


The screenshot shows the Oracle SQL Developer interface with the Explain Plan window open. The plan details are as follows:

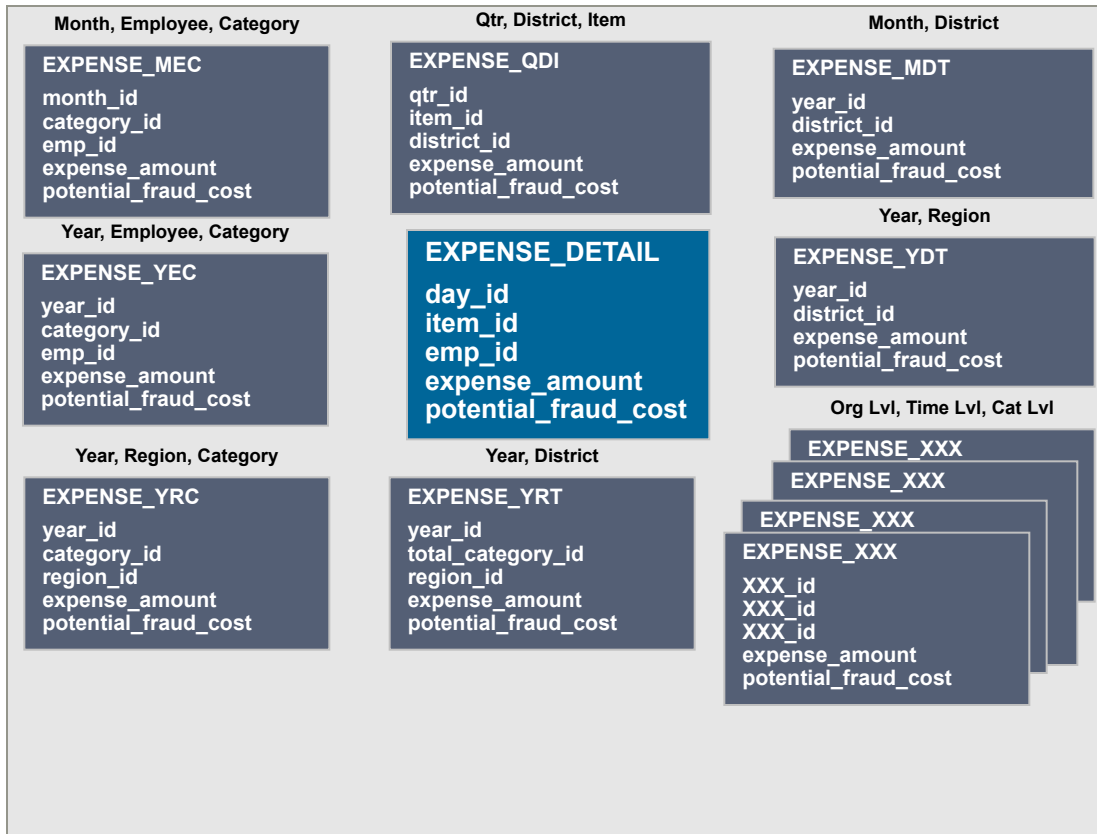
Operation	Optimizer	Cost
SELECT STATEMENT	ALL_ROWS	30
SORT(ORDER BY)		30
JOINED CUBE SCAN(OUTER)		
CUBE ACCESS DM.SALES_CUBE		
CUBE ACCESS DM.CHANNEL		
CUBE ACCESS DM.GEOGRAPHY		
CUBE ACCESS DM.PRODUCT		
CUBE ACCESS DM.TIME		29

Материализованные представления

- Хранение суммарных данных
- Использование материализованных представлений при автоматической оптимизации запросов
- Средства модификации запросов
- Различные методы обновления (полное, частичное и отложенное)

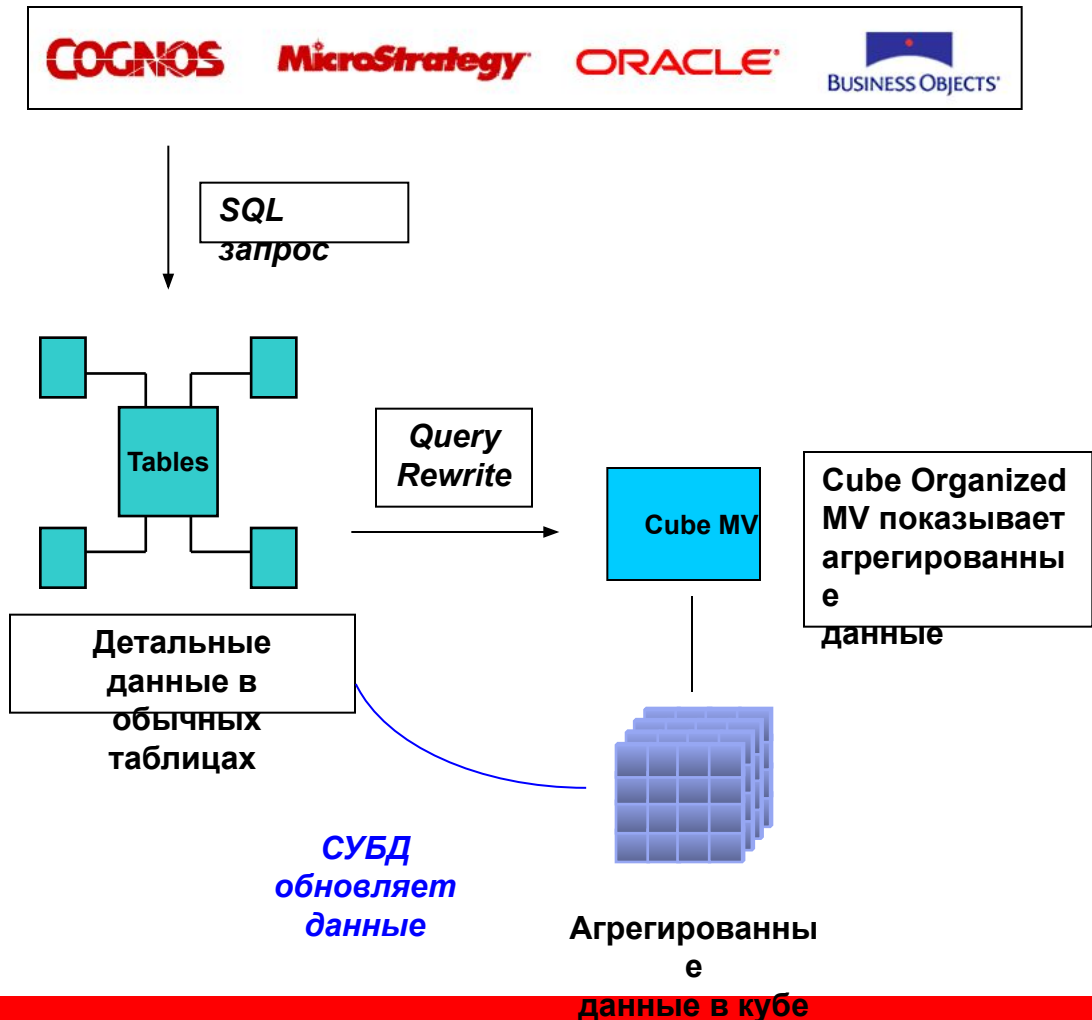


Materialized Views



- Создание MV при нерегламентированных запросах - проблема
- Пользователи хотят ровный отклик
- Появление большого количества MV

11g Cube-Organized Materialized Views



- Детальные данные в реляционных таблицах
- Приложение генерирует SQL запрос
- Агрегированные данные в OLAP кубе
- Доступ к агрегированным данным в кубе – автоматический, через query rewrite
- Приложение не меняется, но запросы работают быстрее

Ошибки при использовании OLAP

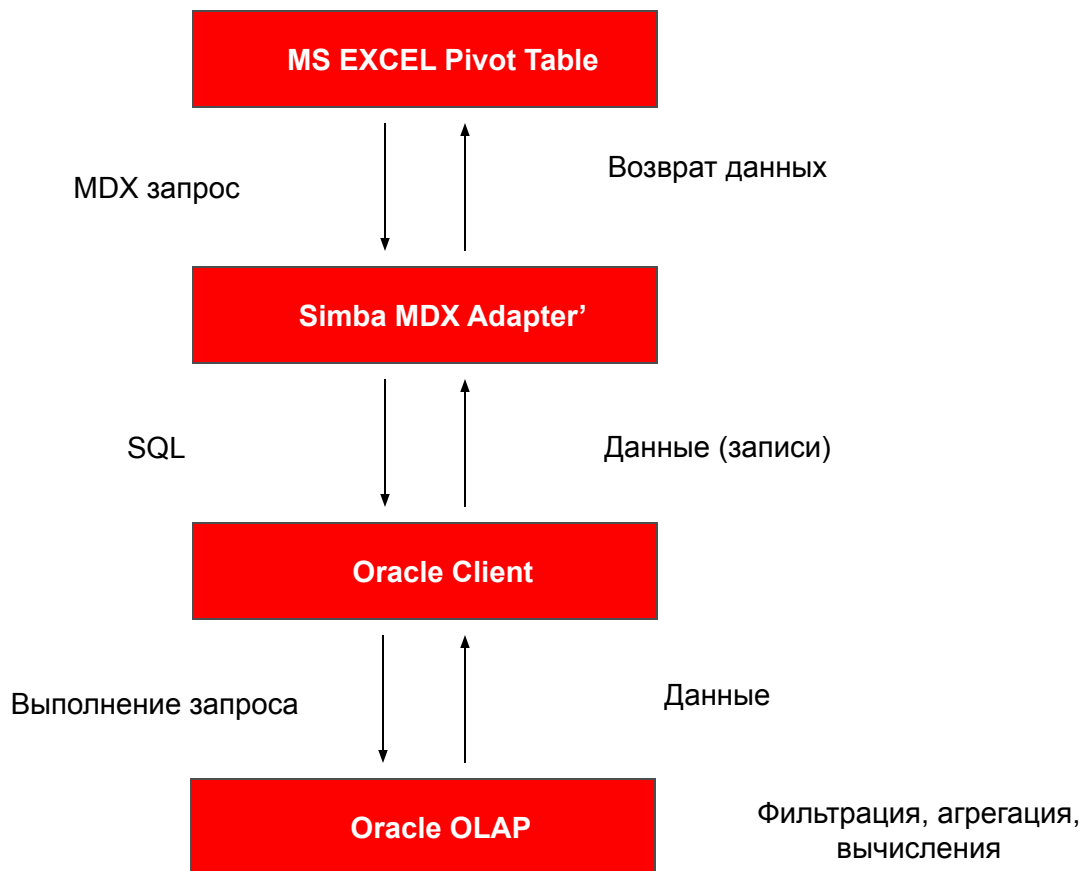
- Желание подменить хранилище OLAP системой
- Хранение слишком детальных данных в OLAP
- Ожидание, что OLAP сервер решит сам все проблемы с производительностью
- непонимание разницы между реляционными и многомерными СУБД

Oracle OLAP MDX Driver (<http://simba.com>)

The screenshot shows a Microsoft Excel PivotTable with the following data:

Row Labels	2003	2004	2005	2006	Grand Total
Sales					
Column Labels					
2003					
2004					
2005					
2006					
Grand Total					
Catalog	55,058,877.00	53,597,015.55	50,841,642.26	49,376,249.75	700,716,503.61
Hardware	50,323,839.83	49,053,106.48	46,645,492.47	45,397,476.68	651,721,659.28
North America	28,761,700.69	27,705,871.74	26,323,260.17	25,843,549.51	377,886,775.02
Asia Pacific	13,133,370.51	12,418,449.67	11,809,818.15	11,255,409.59	162,853,838.93
Europe	8,428,768.63	8,928,784.07	8,512,414.15	8,298,517.58	110,981,045.33
Software/Other	4,735,037.17	4,543,909.07	4,196,149.79	3,978,773.07	48,994,844.33
North America	2,780,732.32	2,628,916.06	2,426,846.45	2,303,319.61	28,704,588.64
Asia Pacific	1,095,629.50	1,037,151.85	925,670.75	856,552.55	11,486,965.97
Europe	858,675.35	877,839.76	843,632.59	818,900.91	8,803,289.72
Direct Sales	7,699,894.50	7,767,892.00	7,712,991.45	7,759,440.97	69,842,132.27
Hardware	6,993,138.30	7,051,972.39	6,973,926.86	7,039,407.63	64,096,879.08
North America	3,962,100.37	4,133,009.81	4,121,961.15	4,176,240.81	37,389,676.18
Asia Pacific	1,840,007.88	1,731,501.67	1,723,746.36	1,719,046.41	16,376,162.50
Europe	1,191,030.05	1,187,460.91	1,128,219.35	1,144,120.41	10,351,040.40
Software/Other	706,756.20	715,919.61	739,064.59	720,033.34	5,745,253.19
North America	406,730.47	416,562.83	415,704.56	415,817.69	3,305,601.81
Asia Pacific	188,100.96	186,078.91	199,077.12	183,506.68	1,513,459.30
Europe	111,924.77	113,277.87	124,282.91	120,708.97	926,192.08
Internet	67,517,742.36	82,925,778.00	78,431,938.25	83,002,626.67	349,734,116.21
Hardware	60,419,113.50	74,040,308.65	70,571,916.82	74,459,996.73	313,810,249.55
North America	25,532,001.71	29,951,840.10	31,279,414.71	33,672,016.82	139,459,795.42
Asia Pacific	31,733,342.54	40,841,555.23	36,024,786.46	37,347,585.16	157,009,412.67

Работа Simba MDX-OLAP Adapter



Для чего можно использовать OLAP 11g

- Как MOLAP сервер
- Как вычислительное ядро для приложений на Oracle
- Как новый тип MV
- Как ускоритель запросов по агрегированным данным

OLAP 11g и BI EE



Databases

- local (global) - OLAP 11g
 - Schemas
 - DM
 - GLOBAL
 - Analytic Workspaces
 - GLOBAL (attached PMA)**
 - Dimensions
 - Cubes
 - PRICE_CUBE
 - UNITS_CUBE
 - Measures
 - SALES_PROD_TOT_SHARE
 - SALES_PY
 - SALES_YTD
 - SALES_YTD_CHG_PY
 - SALES_YTD_PCT_CHG_PY
 - SALES_YTD_PY
 - PROFIT
 - HOW_IS_MARGIN
 - PROFIT_2_PER_MOVING

- Create Analytic Workspace...
- Create Analytic Workspace From Template...
- Maintain Analytic Workspace GLOBAL...
- Set Analytic Workspace GLOBAL Object Security...
- Detach Analytic Workspace GLOBAL
- Save Analytic Workspace GLOBAL To Template...
- Export Analytic Workspace GLOBAL To EIF File...
- Delete Analytic Workspace GLOBAL
- OLAP Worksheet...
- Export To OBIEE Administrator...**

Dimensions:

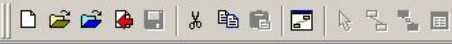
Name	Long Description	Type
CHANNEL	Channel	User
CUSTOMER	Customer	User
PRODUCT	Product	User
TIME	Time	Time

	Long Description	Dimensions
Price Cube		TIME, PRODUCT
Units Cube		TIME, CHANNEL, CUSTOMER...

Name	Cube	Long Description
UNIT_PRICE	PRICE_CUBE	Unit Price
UNIT_COST	PRICE_CUBE	Unit Cost
UNITS	UNITS_CUBE	Units
SALES	UNITS_CUBE	Sales
COST	UNITS_CUBE	Cost


```
#
# Instructions.
#
# 1) Create a new repository or use an existing repository in the OBIEE Administrator
#
# 2) Copy/paste this document into the window of the Physical Database. A new Physical
# Database, Business Model and Presentation Catalog will be created.
# It is safe to include the comments at the top of this file when copy/pasting into OBIEE Administrator.
#
# Note: The OBIEE repository should not already contain a Physical Database,
# Business Model or Presentation Catalog with names created in this export.
#
# 3) Create a new security policy that will apply to the logical table for the cube(s)
# in the Business Model. This can be done using the following instructions:
#
# a) Create a new OBIEE user Group in the OBIEE Administrator (e.g. OLAP Users).
#
# b) Create a new OBIEE user in the OBIEE Administrator and add it to the Group.
#
# 4) For the OLAP Users group, create a security filter on the Logical Table(s)
# in the Business Model using the following text. Do not include the initial # character.
#
# For Cube Logical Table: Price Cube
#
# "GLOBAL"."Product"."Product Key" = "GLOBAL"."Price Cube"."Product" AND "GLOBAL"."Time"."Time Key" = "GLOBAL"."Price Cube"."Time"
#
#
# For Cube Logical Table: Units Cube
#
# "GLOBAL"."Channel"."Channel Key" = "GLOBAL"."Units Cube"."Channel" AND "GLOBAL"."Customer"."Customer Key" = "GLOBAL"."Units Cube"."Customer" AND "GLOBAL"."Product"."Product Key" =
"GLOBAL"."Units Cube"."Product" AND "GLOBAL"."Time"."Time Key" = "GLOBAL"."Units Cube"."Time"
#
#
```

Close



Presentation

- GLOBAL
 - Channel
 - Customer
 - Product
 - Time
 - Units Cube
 - Cost
 - How Is Margin
 - Profit
 - Profit 3 Period Moving Avg
 - Profit Change Prior Period
 - Profit Change Prior Year
 - Profit % Chg Prior Period
 - Profit % Chg Prior Year
 - Profit Prior Period
 - Profit Share of Prod Parent
 - Profit Share of Total Product
 - Profit Prior Year
 - Profit Ytd
 - Profit Ytd Chg Prior Year
 - Profit Ytd % Chg Prior Year
 - Profit Ytd Prior Year
 - Sales
 - Sales 3 Period Moving Avg
 - Sales Change Prior Period
 - Sales Change Prior Year
 - Sales % Chg Prior Period
 - Sales % Chg Prior Year
 - Sales Prior Period
 - Sales Share of Prod Parent
 - Sales Share of Total Product
 - Sales Prior Year
 - Sales Ytd
 - Sales Ytd Chg Prior Year
 - Sales Ytd % Chg Prior Year
 - Sales Ytd Prior Year
 - Units
- MVDEMO
- Paint
- Paint Exec
- ptsessbase

Business Model and Mapping

- GLOBAL
 - Channel
 - Customer
 - Product
 - Time
 - Channel
 - Customer
 - Product
 - Time
 - Units Cube
 - Sources
 - Channel
 - Customer
 - Product
 - Time
 - Cost
 - How Is Margin
 - Profit
 - Profit 3 Period Moving Avg
 - Profit Change Prior Period
 - Profit Change Prior Year
 - Profit % Chg Prior Period
 - Profit % Chg Prior Year
 - Profit Prior Period
 - Profit Share of Prod Parent
 - Profit Share of Total Product
 - Profit Prior Year
 - Profit Ytd
 - Profit Ytd Chg Prior Year
 - Profit Ytd % Chg Prior Year
 - Profit Ytd Prior Year
 - Sales
 - Sales 3 Period Moving Avg
 - Sales Change Prior Period
 - Sales Change Prior Year
 - Sales % Chg Prior Period
 - Sales % Chg Prior Year
 - Sales Prior Period
 - Sales Share of Prod Parent
 - Sales Share of Total Product
 - Sales Prior Year
 - Sales Ytd
 - Sales Ytd Chg Prior Year
 - Sales Ytd % Chg Prior Year
 - Sales Ytd Prior Year
 - Units
- MVDEMO

Physical

- GLOBAL
 - CHANNEL_PRIMARY_VIEW
 - CHANNEL_VIEW
 - CUSTOMER_MARKET_VIEW
 - CUSTOMER_SHIPMENTS_VIEW
 - CUSTOMER_VIEW
 - PRODUCT_PRIMARY_VIEW
 - PRODUCT_VIEW
 - TIME_CALENDAR_VIEW
 - TIME_FISCAL_VIEW
 - TIME_VIEW
 - UNITS_CUBE_VIEW
 - CHANNEL
 - COST
 - CUSTOMER
 - HOW_IS_MARGIN
 - PRODUCT
 - PROFIT
 - PROFIT_3_PER_MOVAVG
 - PROFIT_CHG_PP
 - PROFIT_CHG_PY
 - PROFIT_PCT_CHG_PP
 - PROFIT_PCT_CHG_PY
 - PROFIT_PP
 - PROFIT_PROD_PRNT_SHARE
 - PROFIT_PROD_TOT_SHARE
 - PROFIT_PY
 - PROFIT_YTD
 - PROFIT_YTD_CHG_PY
 - PROFIT_YTD_PCT_CHG_PY
 - PROFIT_YTD_PY
 - SALES
 - SALES
 - SALES_3_PER_MOVAVG
 - SALES_CHG_PP
 - SALES_CHG_PY
 - SALES_PCT_CHG_PP
 - SALES_PCT_CHG_PY
 - SALES_PP
 - SALES_PROD_PRNT_SHARE
 - SALES_PROD_TOT_SHARE
 - SALES_PY
 - SALES_YTD
 - SALES_YTD_CHG_PY
 - SALES_YTD_PCT_CHG_PY
 - SALES_YTD_PY
 - TIME
 - UNITS
- MVDEMO

- GLOBAL
- Columns
 - Channel
 - Total
 - Channel
 - Customer
 - Product
 - Total
 - Class
 - Family
 - Item
 - Buyer
 - Marketing Manager
 - Package
 - Time
 - Units Cube
 - Cost
 - How Is Margin
 - Profit
 - Profit 3 Period Moving Av
 - Profit Change Prior Perio
 - Profit Change Prior Year
 - Profit % Chg Prior Perioc
 - Profit % Chg Prior Year
 - Profit Prior Period
 - Profit Share of Prod Pare
 - Profit Share of Total Proc

Compound Layout

Untitled Analysis

Add View: [Grid] [Table] [Chart] [Map] [Other]

Table				
Total	Class	Family	Profit Change Prior Year	Profit 3 Period Moving Avg
Total Channel	Hardware	CD/DVD		19,765,631
		Desktop PCs		17,151,377
		Memory		5,742,324
		Modems/Fax		7,164,215
		Monitors		4,330,199
		Portable PCs		10,222,282

[Download](#) - [Copy](#)

О чем помнить при работе с OLAP

- Oracle OLAP – другая СУБД, со своими принципами и подходами к оптимизации.
 - Частая жалоба – люди, не имеющие опыта с OLAP сталкиваются с тем, что все не работает «быстро» и сразу
- Существуют миф, что если запросы на огромном объеме данных работают медленно – данные нужно переложить в OLAP и все будет работать быстро.
- OLAP опция динамично развивается, поэтому лучше использовать как можно более позднюю версию и патчи.
 - Olap 10g сильно отличается от OLAP 9i, а OLAP 11g отличается от OLAP 10g

Дополнительная информация

<http://oracle.com>

Email:

Andrey.Pivovarov@oracle.com

Блог:

<http://www.oraclebi.ru/>



Q&A