



**ORACLE®**

**BI Design Best Practices**  
**Nicolas Barasz - Oracle BI EPM Product Assurance**  
**April 2012**

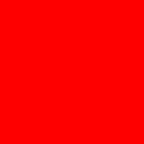
# Agenda

- Oracle BI Principles
- Repository design best practices
- Dashboards and reports design best practices
- 10g Upgrade considerations

# Oracle BI Principles

Oracle BI Server is a SQL generator !!

- To see if an OBI implementation is good, first look at the physical SQL queries executed: is it right (provide good results) ? Is it as simple as possible, is it optimized ?
- End-users do not see physical SQLs, so they do not care about it. Very often, developers do not really look at them because it is just something "automatically generated" (until they have big problems). DBAs do not really look at their structure because developers say that it cannot be modified. "This is how the product works, we cannot change it"



## Oracle BI Principles

**The quality of physical SQL queries is the most important part of Oracle BI.**

**Everything depend on it.**

# Oracle BI Principles

What are the three main objectives of any Oracle BI application ?

# Oracle BI Principles

Ordered by priority:

1. Get Correct Results.
2. Get Results Fast.
3. Get Results Easily.

# Oracle BI Principles

## Get Correct Results

This looks obvious but very often this objective is not achieved due to errors in RPD configurations. End-users can get wrong results for years without noticing anything. Look at physical SQL queries to identify potential issues (missing or additional filter, an additional table that should not be included...)

# Oracle BI Principles

## Get Results Fast

Often business users do not have clear expectations. They want “acceptable” performance. Most of the time performance is not the main focus of developers when a project is started. They only pay attention to this objective at the end of the implementation, or when they are in front of a big problem.

It is useless to have a great presentation if each click takes 2 minutes. Performance must be a permanent concern from the beginning of the design of the project until the production roll-out.

# Oracle BI Principles

## Get Results Easily

Navigation, graphs, hierarchies, report format...

Everybody (end-users and developers) focus on that part. That's the main cause of failure/big troubles for many projects.

Developers should always keep the performance impact in mind when discussing the presentation with business users.

# Oracle BI Principles

Performance should always be before presentation in term of priority. It is possible to accept sacrifices on presentation during the design of the project. But when the implementation is over, it is extremely difficult to explain to users that they have to give up the presentation because of performance issues.

**Focus on performance from day 1**

# Agenda

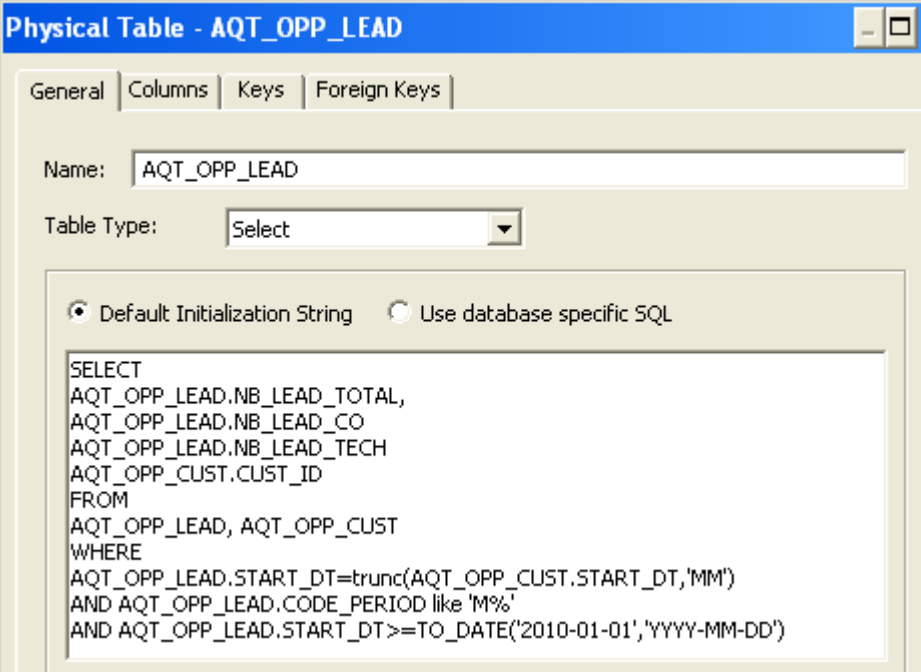
- Oracle BI Principles
- Repository design best practices
- Dashboards and reports design best practices
- 10g Upgrade considerations

# Agenda

- Oracle BI Principles
- Repository design best practices
  - Physical Layer
  - Business Model
  - Presentation Layer
- Dashboards and reports design best practices
- 10g Upgrade considerations

# Opaque Views

- Push the SQL statement as a sub select to the main SQL generated from the query
- All tables used in opaque view definition are always queried together, even if some of them are not really necessary.
- Should be used as a last resort only. For instance when variables must be included in SQL with multiple levels of aggregation.



Physical Table - AQT\_OPP\_LEAD

General | Columns | Keys | Foreign Keys

Name: AQT\_OPP\_LEAD

Table Type: Select

☒ Default Initialization String ☐ Use database specific SQL

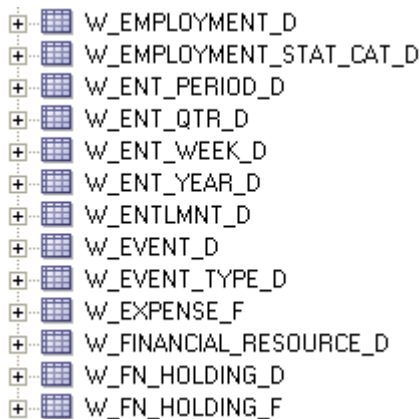
```
SELECT
AQT_OPP_LEAD.NB_LEAD_TOTAL,
AQT_OPP_LEAD.NB_LEAD_CO
AQT_OPP_LEAD.NB_LEAD_TECH
AQT_OPP_CUST.CUST_ID
FROM
AQT_OPP_LEAD, AQT_OPP_CUST
WHERE
AQT_OPP_LEAD.START_DT=trunc(AQT_OPP_CUST.START_DT,'MM')
AND AQT_OPP_LEAD.CODE_PERIOD like 'M%'
AND AQT_OPP_LEAD.START_DT >= TO_DATE('2010-01-01','YYYY-MM-DD')
```

# Opaque Views

- When possible, replace the view by aliases of the corresponding physical tables. Filters may be applied in logical table sources or in physical joins.
- Or create a physical table instead, loaded in the ETL process
- Or create a materialized view (in RPD, materialized views should be created as normal physical tables)

# Create Aliases for all tables

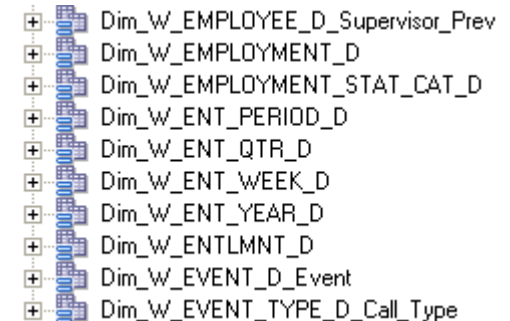
- Create Aliases for all tables and prefix their names with text that reflects the type of table e.g. Dim\_ , Fact\_ or Fact\_Agg.
- Create joins between the Alias tables, not the “master” ones.



A vertical list of 14 table names, each preceded by a small icon of a grid with a plus sign. The names are: W\_EMPLOYMENT\_D, W\_EMPLOYMENT\_STAT\_CAT\_D, W\_ENT\_PERIOD\_D, W\_ENT\_QTR\_D, W\_ENT\_WEEK\_D, W\_ENT\_YEAR\_D, W\_ENTLMNT\_D, W\_EVENT\_D, W\_EVENT\_TYPE\_D, W\_EXPENSE\_F, W\_FINANCIAL\_RESOURCE\_D, W\_FN\_HOLDING\_D, and W\_FN\_HOLDING\_F.

W\_EMPLOYMENT\_D  
W\_EMPLOYMENT\_STAT\_CAT\_D  
W\_ENT\_PERIOD\_D  
W\_ENT\_QTR\_D  
W\_ENT\_WEEK\_D  
W\_ENT\_YEAR\_D  
W\_ENTLMNT\_D  
W\_EVENT\_D  
W\_EVENT\_TYPE\_D  
W\_EXPENSE\_F  
W\_FINANCIAL\_RESOURCE\_D  
W\_FN\_HOLDING\_D  
W\_FN\_HOLDING\_F

← **Original tables**  
**VS Aliases** →

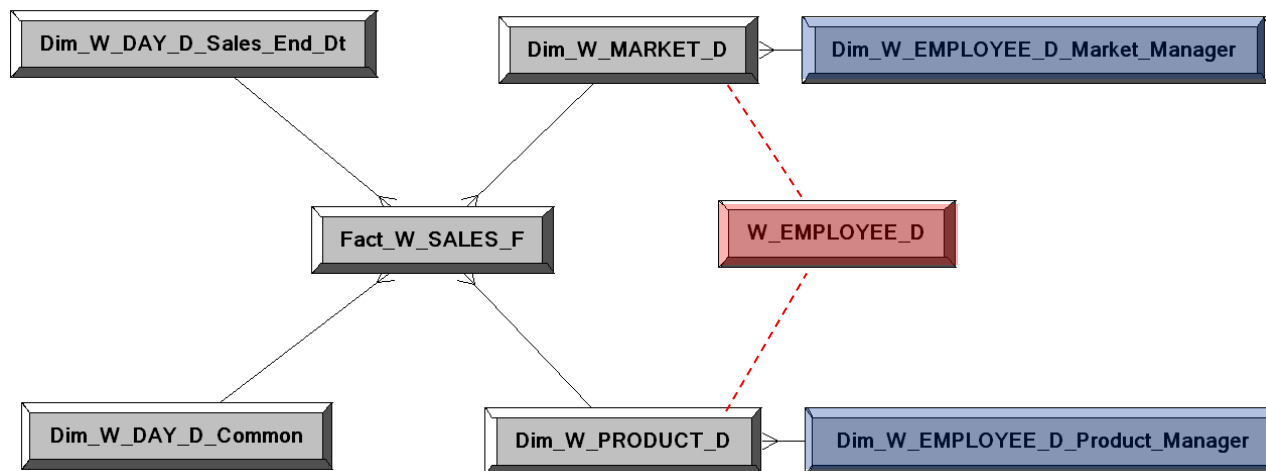


A vertical list of 11 alias table names, each preceded by a small icon of a grid with a plus sign. The names are: Dim\_W\_EMPLOYEE\_D\_Supervisor\_Prev, Dim\_W\_EMPLOYMENT\_D, Dim\_W\_EMPLOYMENT\_STAT\_CAT\_D, Dim\_W\_ENT\_PERIOD\_D, Dim\_W\_ENT\_QTR\_D, Dim\_W\_ENT\_WEEK\_D, Dim\_W\_ENT\_YEAR\_D, Dim\_W\_ENTLMNT\_D, Dim\_W\_EVENT\_D\_Event, and Dim\_W\_EVENT\_TYPE\_D\_Call\_Type.

Dim\_W\_EMPLOYEE\_D\_Supervisor\_Prev  
Dim\_W\_EMPLOYMENT\_D  
Dim\_W\_EMPLOYMENT\_STAT\_CAT\_D  
Dim\_W\_ENT\_PERIOD\_D  
Dim\_W\_ENT\_QTR\_D  
Dim\_W\_ENT\_WEEK\_D  
Dim\_W\_ENT\_YEAR\_D  
Dim\_W\_ENTLMNT\_D  
Dim\_W\_EVENT\_D\_Event  
Dim\_W\_EVENT\_TYPE\_D\_Call\_Type

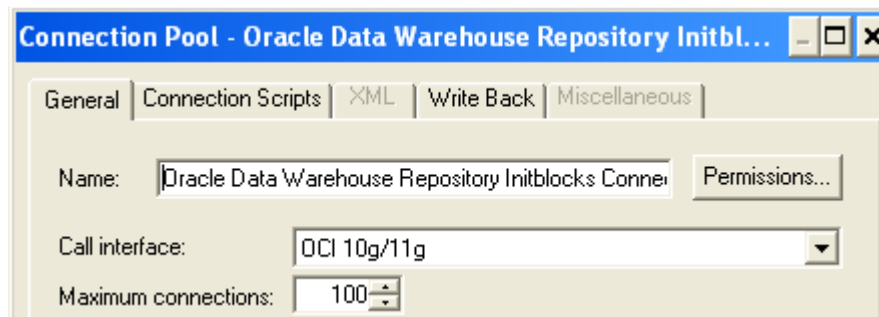
# Avoid Circular Joins

- Circular joins may have a big impact on data integrity. They can always be avoided by creating aliases.



# Connection Pool Configuration

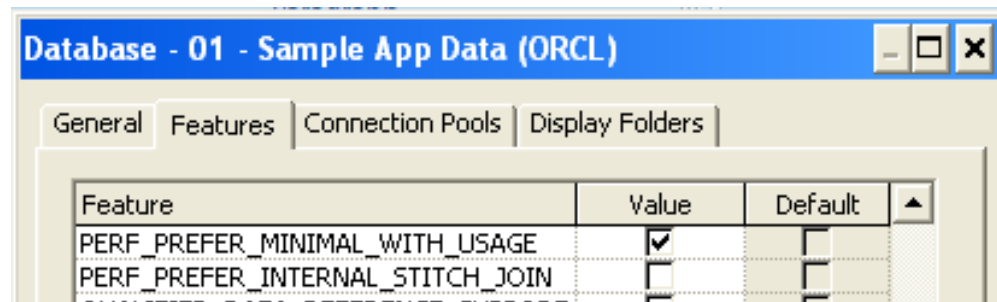
- Use native database drivers instead of ODBC.
- Set the maximum number of connections high enough. Recommendation is to set it to 10-20% of concurrent users multiplied by number of queries executed on a dashboard page. Note that due to usage of expandable hierarchies, the number of queries executed in parallel in 11g is often greater than in 10g.
- Use a separate connection pool for initialization blocks.



# Database Features

Depending on your configuration, you may enable some parameters in database feature:

- **PER\_PREFER\_MINIMAL\_WITH\_USAGE**: Enable this parameter if your database optimizer cannot handle properly WITH clause, for instance on database Oracle 10g.
- **PERF\_PREFER\_INTERNAL\_STITCH\_JOIN**: This parameter may sometimes be enabled to work around database optimizer bugs. Note that it may increase significantly the workload on BI Server. It is usually not recommended.



# Database Features

- `PERF_PREFER_SUPPRESS_EMPTY_TUPLES`: This is for Essbase only. If enabled, instead of applying non empty on the axis, which may contains a very sparse set. Each cross-join of two dimensions will have empty tuples suppressed before cross-joining another dimension.

# Log Level

In production environment, set BI Server log level to 0. When there is a lot of reports running in parallel, query logging may cause performance issues.

**Session Variable Initialization Block - DUAL GENERIC NUM VALUE =0**

Name:

☐ Disabled ☐ Allow deferred execution

Data Source

Connection Pool: 

Database: Oracle 11g/Exadata (Initialization string inherited from Default)

Variable Target

| Name                                                                                         | Default Initializer |
|----------------------------------------------------------------------------------------------|---------------------|
|  LOGLEVEL | 0                   |

**Options**

Cache Manager | Multiuser | More

General | Repository | Sort Objects

☐ Show tables and dimensions only under display folders

☐ Hide level based measure

System logging level:

# Query Limits

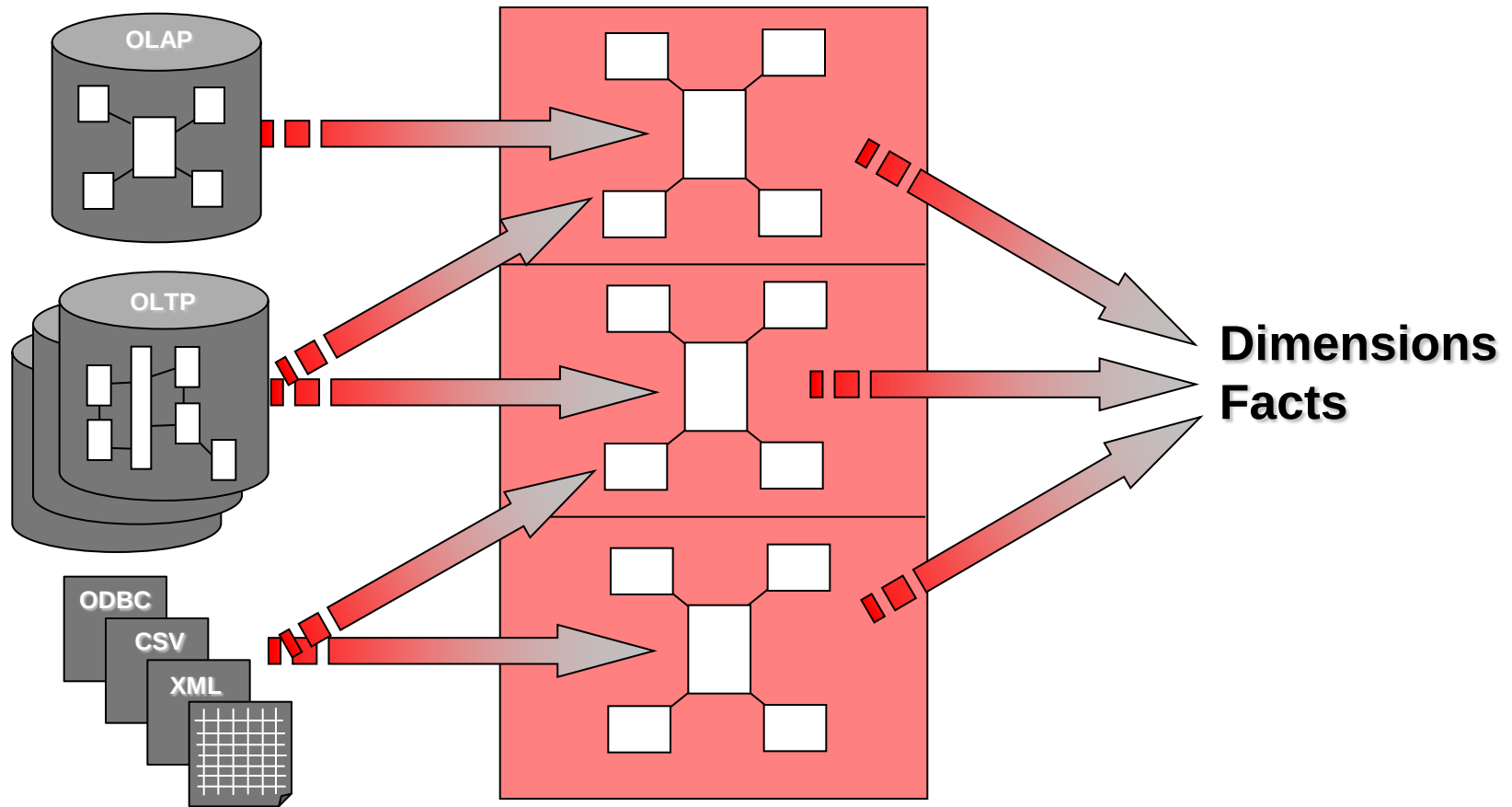
A user who has access to Answer can significantly slow down the BI Server and the database with a bad report that extracts millions of records. To prevent that, enable query limits. If there is no specific users requirement, use 100 000 rows and 1h as a starting point.

| User/Application Role Permissions - BIConsumer                                      |                               |          |                 |          |                 |                    |
|-------------------------------------------------------------------------------------|-------------------------------|----------|-----------------|----------|-----------------|--------------------|
| Object Permissions   Data Filters   Query Limits                                    |                               |          |                 |          |                 |                    |
| Type                                                                                | Database                      | Restrict | Status Max Rows | Max Rows | Status Max Time | Max Time (Minutes) |
|  | 08 - Fusion Order Demo (OLTP) | ...      | Enable          | 100000   | Enable          | 60                 |
|  | 03 - Essbase Sample E2 (Defa  | ...      | Enable          | 100000   | Enable          | 60                 |
|  | 02 - Sample App Xml Target D  | ...      | Enable          | 100000   | Enable          | 60                 |
|  | 01 - Sample App Data (ORCL)   | ...      | Enable          | 100000   | Enable          | 60                 |
|  | 03 - Essbase GL Basic         | ...      | Enable          | 100000   | Enable          | 60                 |

# Agenda

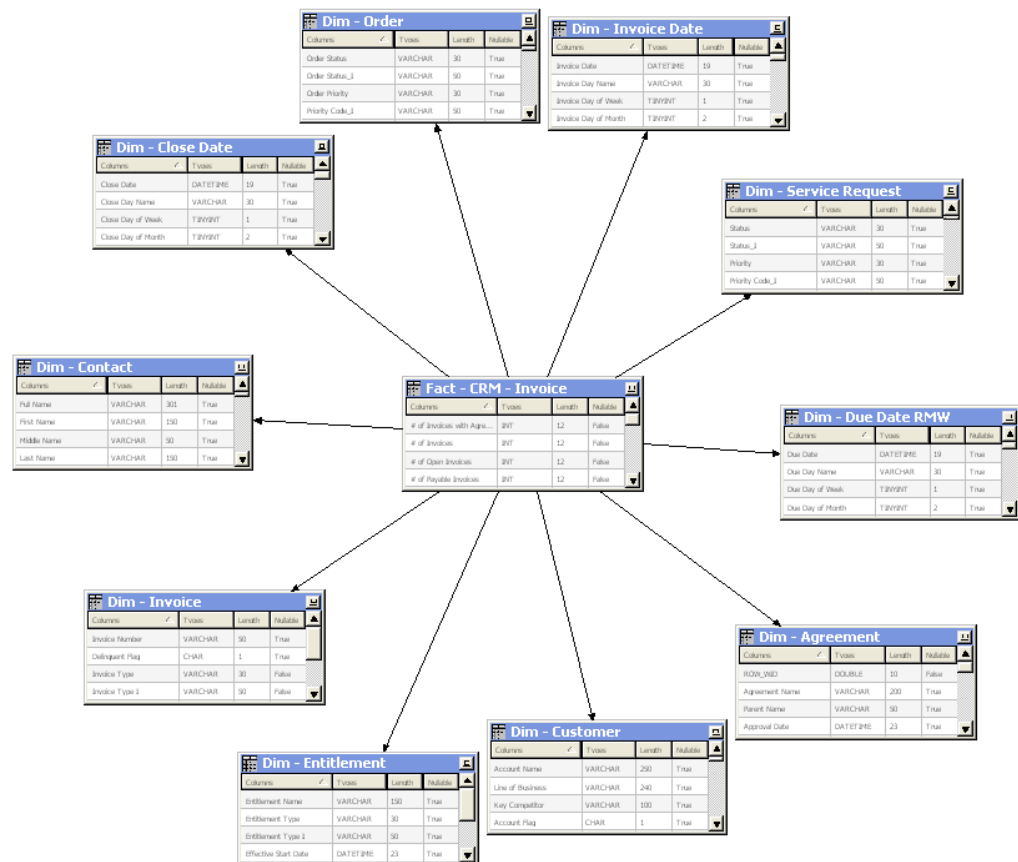
- Oracle BI Principles
- Repository design best practices
  - Physical Layer
  - **Business Model**
  - Presentation Layer
- Dashboards and reports design best practices
- 10g Upgrade considerations

# Business Model Design



# Business Model Design

- Logical star-schemas only:
  - No snow-flaking !
  - Only one exception: BM for Siebel Marketing list formats.



# Dimension Sources per Level

- Create a logical table source in the dimension at each level that matches the level of a fact LTS. It was recommended in 10g, but it is mandatory in 11g.



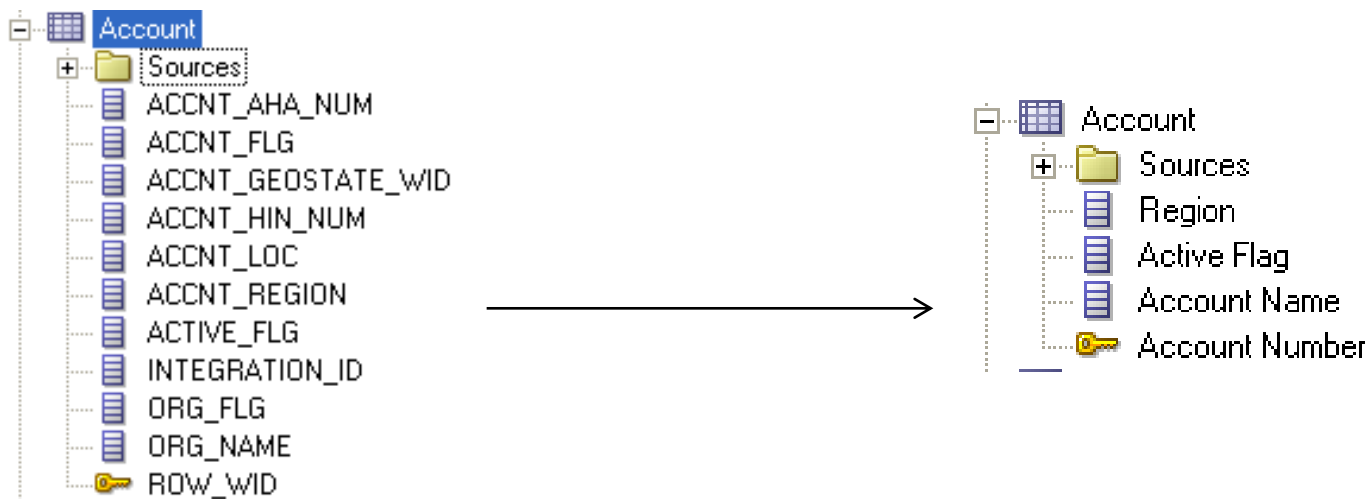
# Logical Tables

- Use a separate dimension logical table for each dimension – don't combine/merge them into one
- The same goes for facts, we don't want to end up with a single fact logical table called "Facts – Stuff"!
- Have a separate logical table for "Compound" facts (which combine facts together from multiple LTS)
- Prefix logical table names with either:
  - Dim –
  - Fact –
  - Fact Compound –

|   |                                                                                       |                                             |
|---|---------------------------------------------------------------------------------------|---------------------------------------------|
| + |    | Dim - Account Rep                           |
| + |    | Dim - Accrue Through Date                   |
| + |    | Dim - Action Type                           |
| + |    | Dim - Activity                              |
| + |    | Dim - Activity Category                     |
| + |    | Dim - Activity Priority                     |
| + |    | Dim - Activity Resolution                   |
| + |    | Fact Compound - Response, Revenue           |
| + |    | Fact Compound - Service Request Time Series |
| + |    | Fact Compound - SR & SR Amount              |
| + |    | Fact - AT - Vehicle Sales                   |
| + |    | Fact - Case Compound Metrics                |
| + |    | Fact - CRM - Account                        |
| + |   | Fact - CRM - Acquired Customer              |
| + |  | Fact - CRM - Active Asset History           |

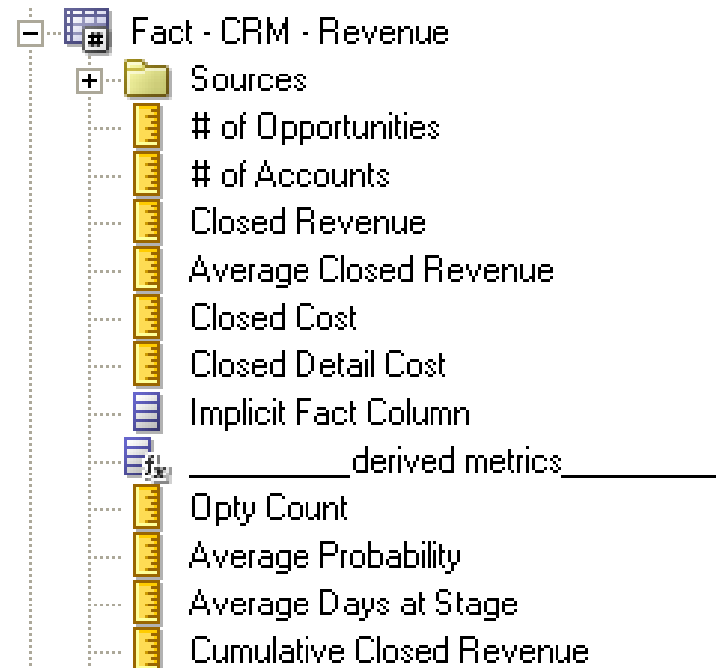
# Logical Table Columns

- Try assigning business columns as dimension primary keys.
- Rename logical columns to use presentation names
- Keep only required columns.



# Logical Table Columns

- You should not assign logical primary keys on fact logical tables
- Create “dummy” measures to separate out facts into various groups if need be
- Make sure almost every fact logical column has an aggregation rule set.



# Level Keys

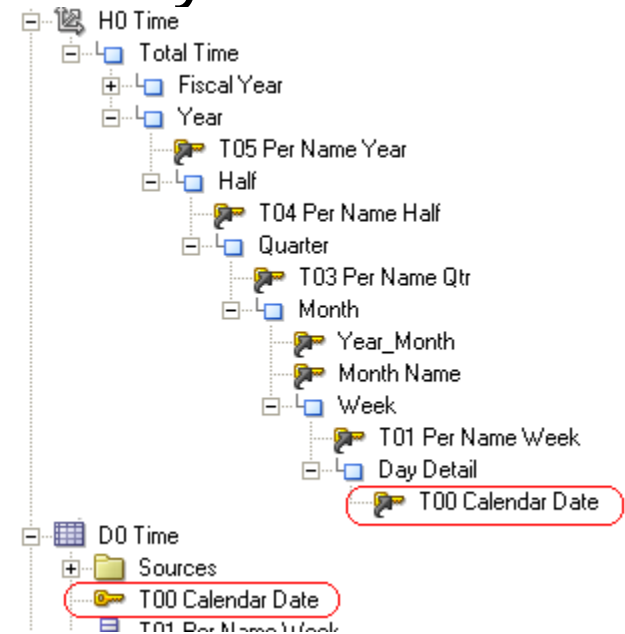
- The primary key of each level must always be unique
- The primary key of the lowest level of the hierarchy must always be the primary of the logical table

Logical Level - Month

General Keys Preferred Drill Path

Primary key: Year\_Month

| Key Name   | Columns           | Description | Use for Display                     |
|------------|-------------------|-------------|-------------------------------------|
| Year_Month | D0 Time.Year_Mont |             | <input type="checkbox"/>            |
| Month Name | D0 Time.Month Nam |             | <input checked="" type="checkbox"/> |



# Count(distinct)

- Whenever it is possible, replace it by Count().  
Count(distinct) has a high cost on performance on the database.
- If there are multiple LTS, the aggregation rule must be specified for each LTS.

Logical Column - # distinct products

General | Column Source | Aggregation | Levels

☐ Based on dimensions

Default aggregation rule: Count Distinct

Overrides:

| Logical Table Source                | Formula                              |
|-------------------------------------|--------------------------------------|
| LTS01 Revenue (Billed Time Join)    | COUNT(DISTINCT "01 - Sample App"."F0 |
| LTS11 Rev. Aggr 1 (Prod x SalesRep) | COUNT(DISTINCT "01 - Sample App"."F0 |
| LTS12 Rev. Aggr 2 (Prod x Qtr)      | COUNT(DISTINCT "01 - Sample App"."F0 |

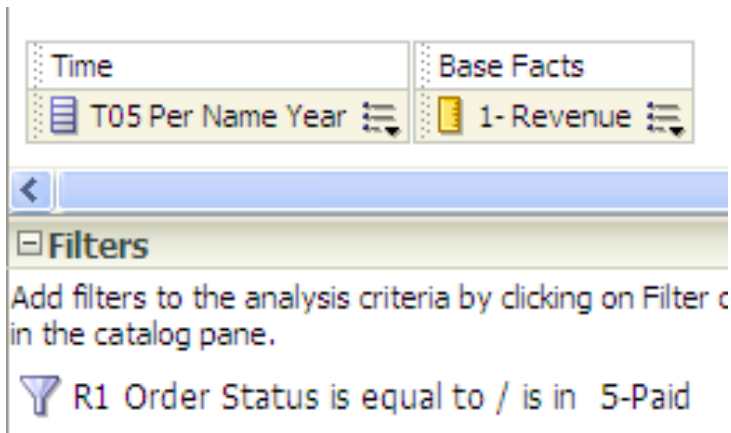
# Base Measure, Case when, Filter Using

Users want to filter the values for a measure. For instance they want number of opened and closed service requests.

There are multiple ways to do that. But each option has consequences...

# Base Measure, Case when, Filter Using

First approach: use the base measure with filters in the report



The screenshot shows a report interface with two columns: 'Time' and 'Base Facts'. Under 'Time' is 'T05 Per Name Year'. Under 'Base Facts' is '1- Revenue'. Below these is a 'Filters' section with a plus icon and the text 'Add filters to the analysis criteria by clicking on Filter c in the catalog pane.' A filter is applied: 'R1 Order Status is equal to / is in 5-Paid'.

```
select sum(T42433.Revenue) as c1,  
       T42406.PER_NAME_YEAR as c2  
from  
       SAMP_TIME_QTR_D T42406 /* D03 Time Quarter Grain */ ,  
       SAMP_REVENUE_F T42433 /* F10 Billed Rev. */  
where  ( T42406.QTR_KEY = T42433.Bill_Qtr_Key and T42433.Order_Status = '5-Paid' )  
group by T42406.PER_NAME_YEAR
```

# Base Measure, Case when, Filter Using

Second approach: use “case when” statement in the Logical Table Source

```
case when "01 - Sample App Data (ORCL)"."."."BISAMPLE"."F10 Billed Rev."."Order_Status"
='5-Paid' then "01 - Sample App Data (ORCL)"."."."BISAMPLE"."F10 Billed Rev."."Revenue"
else 0 end
```

|                                                                                                                                                                                      |                                                                                                                                                                                                |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Time                                                                                                                                                                                 | Calculated Facts                                                                                                                                                                               |
|  T05 Per Name Year  |  Paid Revenue (case when)  |

```
select sum(case when T42433.Order_Status = '5-Paid' then T42433.Revenue else 0 end ) as c1,
       T42406.PER_NAME_YEAR as c2
from
       SAMP_TIME_QTR_D T42406 /* D03 Time Quarter Grain */ ,
       SAMP_REVENUE_F T42433 /* F10 Billed Rev. */
where ( T42406.QTR_KEY = T42433.Bill_Qtr_Key )
group by T42406.PER_NAME_YEAR
```

# Base Measure, Case when, Filter Using

Third approach: use “Filter Using” statement in the logical column

- Derived from existing columns using an expression

```
Filter('01 - Sample App'. 'F0 Sales Base Measures'. '1 - Revenue' Using '01 - Sample App'. 'D7 Orders (Facts Attributes)'. 'R1 Order Status'='5-Paid' )
```

| Time              | Calculated Facts         |
|-------------------|--------------------------|
| T05 Per Name Year | Paid Revenue (Filter By) |

```
select sum(T42433.Revenue) as c1,
       T42406.PER_NAME_YEAR as c2
from
    SAMP_TIME_QTR_D T42406 /* D03 Time Quarter Grain */ ,
    SAMP_REVENUE_F T42433 /* F10 Billed Rev. */
where ( T42406.QTR_KEY = T42433.Bill_Qtr_Key and T42433.Order_Status = '5-Paid' )
group by T42406.PER_NAME_YEAR
```

# Base Measure, Case when, Filter Using

| Solution     | Benefits                                                                                                                  | Downside                                                                                                                          | Rank                                 |
|--------------|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| Base Measure | <ul style="list-style-type: none"><li>-Flexible</li><li>-Perfectly Optimized</li><li>-Good for users' education</li></ul> | <ul style="list-style-type: none"><li>-Cannot be always used, depending on report configuration</li></ul>                         | 1 – Should be used most of the time  |
| Case When    | <ul style="list-style-type: none"><li>-Simple physical query</li><li>-Always works</li></ul>                              | <ul style="list-style-type: none"><li>-No automatic where clause.</li><li>-Need filters in reports for good performance</li></ul> | 2 – Should be used from time to time |
| Filter Using | <ul style="list-style-type: none"><li>-Where clause added automatically</li></ul>                                         | <ul style="list-style-type: none"><li>-Where clause quickly becomes HUGE</li></ul>                                                | 3 – Should be used rarely            |

# IndexCol

Sometimes the formula or columns used vary depending on a session/presentation variable.

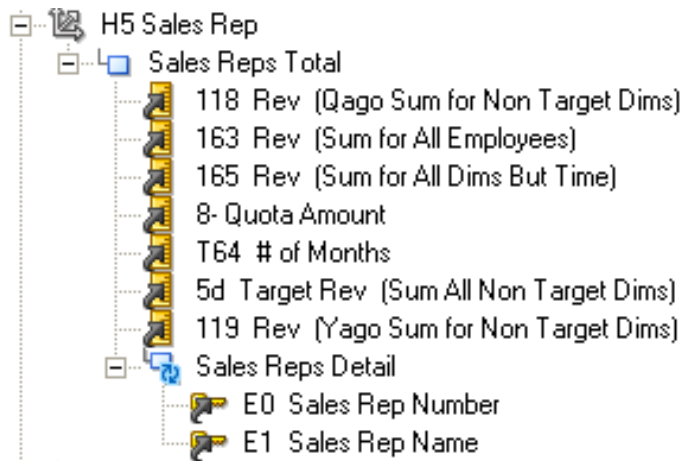
If you use a « case when » statement then the entire formula is pushed to the physical query. But by using function IndexCol only the required column/expression is pushed to the database.

Combined with the new 11g features in prompts (allow selection in a list of custom values), it allows users to modify very significantly reports structure without any increased cost on performance. This function can be used in the RPD or directly in reports.

```
INDEXCOL( CASE  VALUEOF(NQ_SESSION."PREFERRED_CURRENCY") WHEN 'USD' THEN 0 WHEN 'EUR'  
THEN 1 WHEN 'AUD' THEN 2 END , "01 - Sample App Data (ORCL)"."". "BISAMPLE". "F19 Rev.  
(Converted)". "Revenue_Usd", "01 - Sample App Data (ORCL)"."". "BISAMPLE". "F19 Rev.  
(Converted)". "Revenue_Eur", "01 - Sample App Data (ORCL)"."". "BISAMPLE". "F19 Rev.  
(Converted)". "Revenue_Aud")
```

# Missing Dimensional Hierarchies

- Always create a dimension hierarchy for all dimensions, even if there is only one level in the dimension.
  - BI Server may need it to select the most optimized Logical Table Source.
  - It may be useful when BI Server performs a join between two results sets, when two fact tables are used in a report.
  - It is necessary for level-based measures.



# Missing Dimensional Hierarchies

- Always configure drill-down, even if there is only one level in the dimension. It may be useful for instance to drill-down from contact type to contact name.
- Always specify the number of elements per level. BI Server will use it to identify aggregate tables and mini dimensions. It does not need to be accurate, a rough estimate is fine.

Logical Level - Month

General | Keys | Preferred Drill Path

Name: Month

Number of elements at this level: 120 (1 for dimension total)

☐ Grand total level

# Content Level

Always specify the content level in all logical table sources, both in facts and dimensions.

- It will allow BI Server to select the most optimized LTS in queries.
- It will help consistency checker finding the issues in RPD configuration, preventing runtime errors.

The screenshot shows a configuration window titled "Logical Table Source - LTS1 Time Day Grain". It has four tabs: "General", "Column Mapping", "Content", and "Parent-Child Settings". The "Content" tab is active. Inside the "Content" tab, there is a label "Aggregation content, group by" followed by a dropdown menu showing "Logical Level". Below this, there is a table with two columns: "Logical Dimension" and an unlabeled column. The table has two rows: "H0 Time" and "Day Detail".

| Logical Dimension |            |
|-------------------|------------|
| H0 Time           | Day Detail |

# Implicit fact

Set up an implicit fact column for each presentation folder.

It prevents users from getting wrong results if they create a report without fact column.

Use a constant as the implicit fact column to optimize performance

**Subject Area - A - Sample Sales**

General | Presentation Tables | Aliases

Name: A - Sample Sales

☒ Custom display name VALUEOF(NQ\_SESSION.CN\_A\_-\_Sample\_Sales)

Business model: 01 - Sample App

☐ Export logical keys

**Implicit Fact Column**

Sample App"."FO Sales Base Measures"."Implicit fact column" Set...

**Logical Column - Implicit fact column**

General | Column Source | Aggregation | Levels

Data

Type: INT

Derives from: 1

Column Source Type

☒ Derived from physical mappings

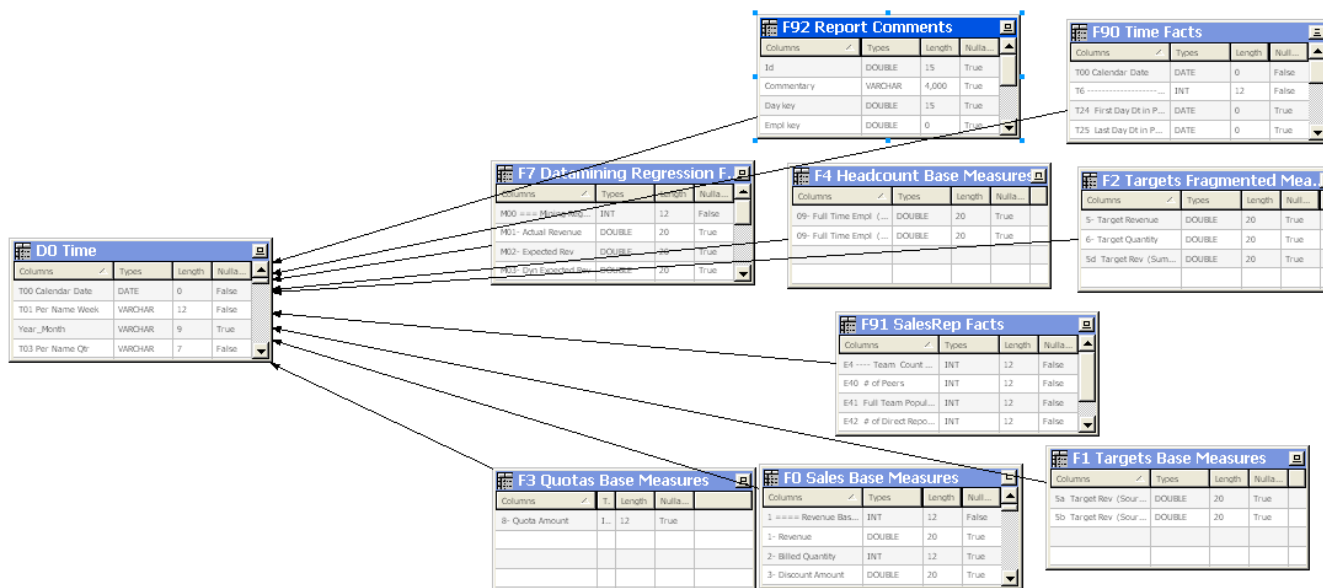
☐ Show all logical sources

Logical Table Source Mapping

| Logical Table Source             | Mapping |
|----------------------------------|---------|
| LTS01 Revenue (Billed Time Join) | 1       |

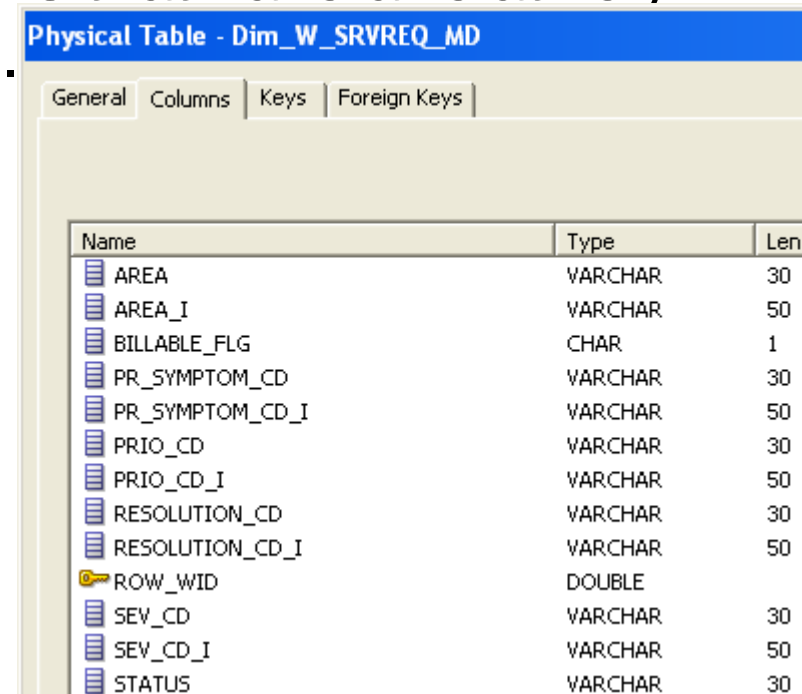
# Canonical Time Dimension

Each Business Model should include a main time dimension connected to almost all fact tables. This is necessary for reports that includes multiple facts. It is also much easier for end-users than having a time dimension per fact table.



# Mini-Dimensions

- Mini-dimension tables include combinations of the most queried attributes of their parent dimensions.
- They must be small compared to the parent dimension, so they can include only columns that have a relatively small number of distinct values.

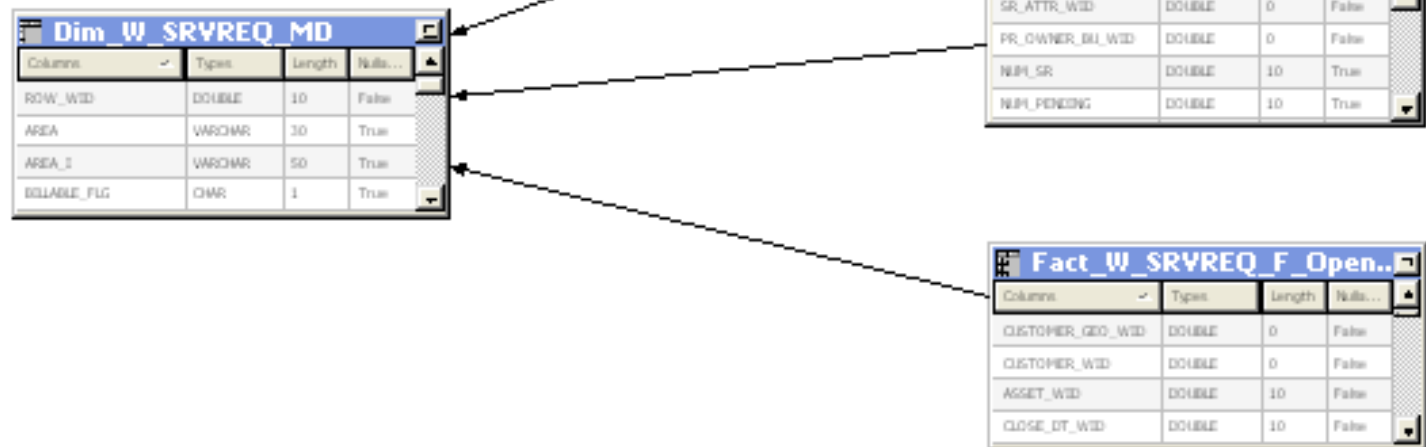


Physical Table - Dim\_W\_SRVREQ\_MD

| Name            | Type    | Len |
|-----------------|---------|-----|
| AREA            | VARCHAR | 30  |
| AREA_I          | VARCHAR | 50  |
| BILLABLE_FLG    | CHAR    | 1   |
| PR_SYMPTOM_CD   | VARCHAR | 30  |
| PR_SYMPTOM_CD_I | VARCHAR | 50  |
| PRIO_CD         | VARCHAR | 30  |
| PRIO_CD_I       | VARCHAR | 50  |
| RESOLUTION_CD   | VARCHAR | 30  |
| RESOLUTION_CD_I | VARCHAR | 50  |
| ROW_WID         | DOUBLE  |     |
| SEV_CD          | VARCHAR | 30  |
| SEV_CD_I        | VARCHAR | 50  |
| STATUS          | VARCHAR | 30  |

# Mini-Dimensions

Mini-dimensions are joined both to main fact table and to aggregate tables.



# Mini-Dimensions

- They improve query performance because BI Server will often use this small table instead of the big parent dimension.
- They increase the usage of aggregate tables. Due to the level of aggregation, aggregate tables cannot be joined to the parent dimension. But they can be joined to the mini-dimension instead. It allows reports to use the aggregate table even if they use some columns from the corresponding dimension.

# Consistency Check Manager

Fix almost all errors, warnings, and best practices detected by consistency check manager.

- If there is a message, it means that there is something wrong in the configuration. It will have consequences, even if there is no problem on the first reports.
- When there are too many messages, it is difficult to see which ones are important.

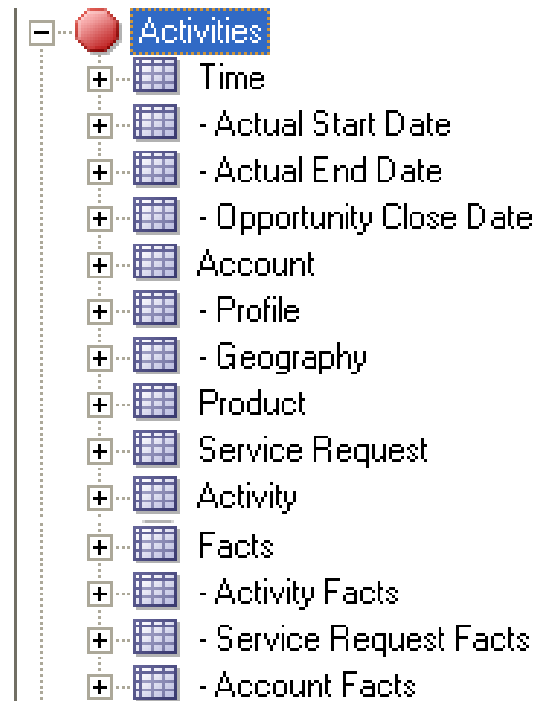
| Consistency Check Manager                                                                                                                              |                  |                    |           |                                                                                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|--------------------|-----------|------------------------------------------------------------------------------------------------------------------|
| Display:                                                                                                                                               |                  |                    |           |                                                                                                                  |
| <input checked="" type="checkbox"/> Errors (0) <input checked="" type="checkbox"/> Warnings (3) <input checked="" type="checkbox"/> Best Practices (0) |                  |                    |           |                                                                                                                  |
| Messages:                                                                                                                                              |                  |                    |           |                                                                                                                  |
| Message Type                                                                                                                                           | Object Type      | Object             | Error No. | Error Description                                                                                                |
| Warning                                                                                                                                                | Database         | 01 - Sample App Da | 39028     | The features in Database '01 - Sample App Data (ORCL)' do not match the defaults. This can cause query problems. |
| Warning                                                                                                                                                | Database         | 04 - Sample OLAP A | 39028     | The features in Database '04 - Sample OLAP AW' do not match the defaults. This can cause query problems.         |
| Warning                                                                                                                                                | Application Role | Human Resources A  | 39051     | Application Role "Human Resources Analyst (pre 7.9.6)" is not defined in the Enterprise Manager.                 |

# Agenda

- Oracle BI Principles
- Repository design best practices
  - Physical Layer
  - Business Model
  - **Presentation Layer**
- Dashboards and reports design best practices
- 10g Upgrade considerations

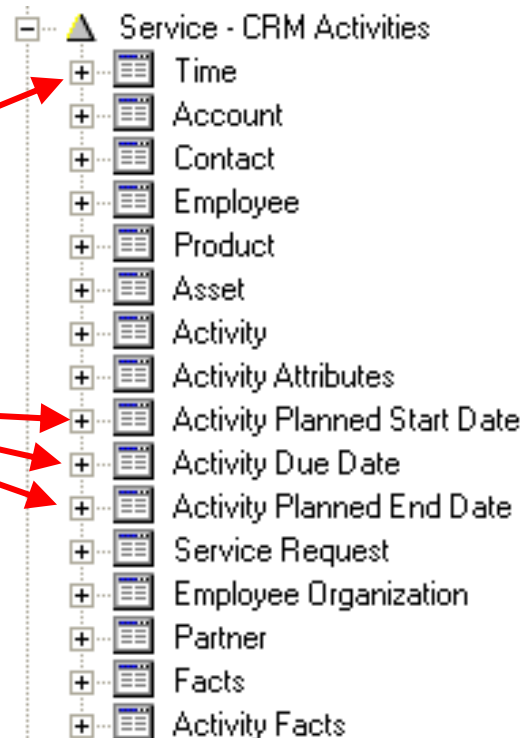
# Simple Presentation Folders

- Small presentation folders are easier to understand and to manipulate.
- Try to limit the number of fact tables, keep the ones that have a lot of common dimensions and are “linked” from a business perspective.
- Configure presentation folders specific to each type of user.



# Canonical Time Dimension

- The “canonical” time dimension should always be the very first presentation Table
- “Secondary” time dimensions can be given their own presentation tables further down



# Homogeneous Presentation Folders

- List the dimension presentation tables first, starting with the canonical time dimension.
- Place the measures/facts at the bottom. Do not mix dimension and fact columns in the same presentation Table
- Naming of presentation tables/columns should be consistent across all folders. This is very important, otherwise prompts values cannot be retrieved when you navigate from one report to another report based on another presentation folders.
- Make it easy to distinguish between dimensions and facts.

# Object Descriptions

- Add descriptions to presentation folders to explain the purpose of each folder within Answers
- Add descriptions to presentation tables and columns so that they appear in Answers when users roll-over them with the mouse. For each column, explain the data content with for instance calculation formula...

# Global Recommendations

- To satisfy all your drill-down requirements, you don't need to have all your reporting objects in a single Subject Area / Presentation Folder
- For example, if you want to drill from a summary "Orders" report down to "Order Item" level, you don't need to create a single Subject Area that contains both Order and Order Item objects
- You can start by creating a report against the "Orders" Subject Area and then you can drill-down to another report defined against "Order Items" Subject Area
- You just need to ensure the Presentation Table/Column names that are being "prompted" have the same names in both Subject Areas
- If the Presentation Table/Column names aren't the same then use Aliases to make them the same!

# Agenda

- Oracle BI Principles
- Repository design best practices
- Dashboards and reports design best practices
- 10g Upgrade considerations

# Override Default Aggregation Rule

It is possible to improve performance by overriding the default aggregation rule for a column in reports when:

- The aggregation rule for all metrics used in this column's formula is SUM
- AND although a formula is applied on this/these metric(s), it is still possible to aggregate the global formula using a SUM
- AND there are multiple levels of aggregation in the report, like multiple views or totals/sub-totals

In this case, overriding the default aggregation rule will reduce the number of physical queries executed.

# Override Default Aggregation Rule

In the following example, the formula used for the metric is `ifnull(Revenue,0)`. There is a pivot table with a total. Note that the aggregation used in the logical sql is “`REPORT_AGGREGATE`”

|             |                       |                       |                       |                       |
|-------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Pivot Table |                       |                       |                       |                       |
|             | 2008                  | 2009                  | 2010                  | IFNULL(1- Revenue, 0) |
|             | IFNULL(1- Revenue, 0) | IFNULL(1- Revenue, 0) | IFNULL(1- Revenue, 0) |                       |
| P4 Brand    |                       |                       |                       |                       |
| BizTech     | 6,990,741             | 6,302,087             | 7,707,172             | 21,000,000            |
| FunPod      | 5,723,187             | 5,160,340             | 6,616,473             | 17,500,000            |
| HomeView    | 3,786,072             | 3,537,573             | 4,176,355             | 11,500,000            |
| Grand Total | 16,500,000            | 15,000,000            | 18,500,000            | 50,000,000            |

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |  |  |  |  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|--|--|
| Logical SQL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |  |  |  |  |
| <pre>SET VARIABLE PREFERRED_CURRENCY='USD';SELECT s_0, s_1, s_2, s_3, s_4, s_5, s_6 FROM ( SELECT   0 s_0,   "A - Sample Sales"."Products"."P4 Brand" s_1,   "A - Sample Sales"."Time"."T05 Per Name Year" s_2,   ifnull("A - Sample Sales"."Base Facts"."1- Revenue",0) s_3,   REPORT_AGGREGATE(ifnull("A - Sample Sales"."Base Facts"."1- Revenue",0) BY "A - Sample Sales"."Products"."P4 Brand") s_4,   REPORT_AGGREGATE(ifnull("A - Sample Sales"."Base Facts"."1- Revenue",0) BY "A - Sample Sales"."Time"."T05 Per Name Year") s_5,   REPORT_AGGREGATE(ifnull("A - Sample Sales"."Base Facts"."1- Revenue",0) BY ) s_6 FROM "A - Sample Sales" ) djm ORDER BY 1, 2 ASC NULLS LAST, 3 ASC NULLS LAST FETCH FIRST 65001 ROWS ONLY</pre> |  |  |  |  |

# Override Default Aggregation Rule

Note the two sub-queries included in the physical SQL:

```
WITH
$ACOMMON36045 AS (select sum(T42442.Revenue) as c1,
    T42409.Brand as c2,
    T42406.PER_NAME_YEAR as c4,
    sum(T42442.Revenue) as c5
from
    SAMP_PRODUCTS_D T42409 /* D10 Product (Dynamic Table) */ ,
    SAMP_TIME_QTR_D T42406 /* D03 Time Quarter Grain */ ,
    SAMP_REVENUE_FA2 T42442 /* F21 Rev. (Aggregate 2) */
where ( T42406.QTR_KEY = T42442.Bill_Qtr_Key and T42409.Prod_Key = T42442.Prod_Key )
group by T42406.PER_NAME_YEAR, T42409.Brand),
$AWITH0 AS (select sum(D1.c1) over (partition by D1.c2) as c1,
    D1.c2 as c2,
    sum(D1.c1) over (partition by D1.c4) as c3,
    D1.c4 as c4,
    D1.c5 as c5
from
    $ACOMMON36045 D1),
$ACOMMON36116 AS (select sum(T42442.Revenue) as c1
from
    SAMP_REVENUE_FA2 T42442 /* F21 Rev. (Aggregate 2) */ ),
$AWITH1 AS (select distinct D1.c2 as c1,
    D1.c4 as c2,
```

# Override Default Aggregation Rule

Next, let's override the aggregation rule:

The diagram illustrates the process of overriding the default aggregation rule in Oracle BI. It consists of two screenshots of the 'Edit Column Formula' dialog box, connected by a large black arrow pointing from left to right.

**Left Screenshot:**

- Column Formula** tab is selected.
- Folder Heading:** Base Facts
- Column Heading:** IFNULL(1- Revenue, 0)
- ☐ Custom Headings
- ☒ Contains HTML Markup
- Aggregation Rule (Totals Row):** Default (highlighted with a red box)
- Available:** Subject Areas
- Column Formula:** ifnull("Base Facts"."1- Revenue",0)

**Right Screenshot:**

- Column Formula** tab is selected.
- Folder Heading:** Base Facts
- Column Heading:** IFNULL(1- Revenue, 0)
- ☐ Custom Headings
- ☒ Contains HTML Markup
- Aggregation Rule (Totals Row):** Sum (highlighted with a red box)
- Available:** Subject Areas
- Column Formula:** ifnull("Base Facts"."1- Revenue",0)

# Override Default Aggregation Rule

The logical SQL now shows REPORT\_SUM:

|             |                       |                       |                       |                       |
|-------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Pivot Table |                       |                       |                       |                       |
|             | 2008                  | 2009                  | 2010                  | IFNULL(1- Revenue, 0) |
|             | IFNULL(1- Revenue, 0) | IFNULL(1- Revenue, 0) | IFNULL(1- Revenue, 0) |                       |
| P4 Brand    |                       |                       |                       |                       |
| BizTech     | 6,990,741             | 6,302,087             | 7,707,172             | 21,000,000            |
| FunPod      | 5,723,187             | 5,160,340             | 6,616,473             | 17,500,000            |
| HomeView    | 3,786,072             | 3,537,573             | 4,176,355             | 11,500,000            |
| Grand Total | 16,500,000            | 15,000,000            | 18,500,000            | 50,000,000            |

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| Logical SQL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |  |
| <pre>SET VARIABLE PREFERRED_CURRENCY='USD';SELECT s_0, s_1, s_2, s_3, s_4, s_5, s_6 FROM ( SELECT   0 s_0,   "A - Sample Sales"."Products"."P4 Brand" s_1,   "A - Sample Sales"."Time"."T05 Per Name Year" s_2,   ifnull("A - Sample Sales"."Base Facts"."1- Revenue",0) s_3,   REPORT_SUM(ifnull("A - Sample Sales"."Base Facts"."1- Revenue",0) BY "A - Sample Sales"."Products"."P4 Brand") s_4,   REPORT_SUM(ifnull("A - Sample Sales"."Base Facts"."1- Revenue",0) BY "A - Sample Sales"."Time"."T05 Per Name Year") s_5,   REPORT_SUM(ifnull("A - Sample Sales"."Base Facts"."1- Revenue",0) BY ) s_6 FROM "A - Sample Sales" ) djm ORDER BY 1, 2 ASC NULLS LAST, 3 ASC NULLS LAST FETCH FIRST 65001 ROWS ONLY</pre> |  |

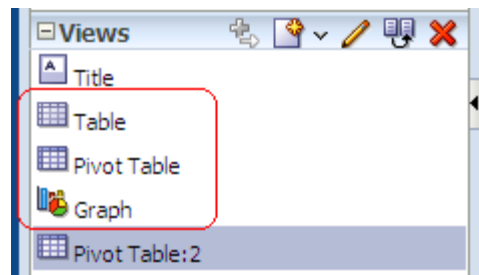
# Override Default Aggregation Rule

The physical SQL now includes only one query:

```
WITH
$AWITH0 AS (select sum(T42442.Revenue) as c1,
    T42409.Brand as c2,
    T42406.PER_NAME_YEAR as c3
from
    SAMP_PRODUCTS_D T42409 /* D10 Product (Dynamic Table) */ ,
    SAMP_TIME_QTR_D T42406 /* D03 Time Quarter Grain */ ,
    SAMP_REVENUE_FA2 T42442 /* F21 Rev. (Aggregate 2) */
where ( T42406.QTR_KEY = T42442.Bill_Qtr_Key and T42409.Prod_Key = T42442.Prod_Key )
group by T42406.PER_NAME_YEAR, T42409.Brand),
$AWITH1 AS (select distinct 0 as c1,
    D1.c2 as c2,
    D1.c3 as c3,
    nvl(D1.c1 , 0) as c4
from
    $AWITH0 D1)
select D1.c1 as c1, D1.c2 as c2, D1.c3 as c3, D1.c4 as c4, D1.c5 as c5, D1.c6 as c6, D1.c7 as c7 from ( select D1.c1 as c1,
    D1.c2 as c2,
    D1.c3 as c3,
    D1.c4 as c4,
    sum(D1.c4) over (partition by D1.c2) as c5,
    sum(D1.c4) over (partition by D1.c3) as c6,
    sum(D1.c4) over () as c7
from
    $AWITH1 D1
order by c1, c2, c3 ) D1 where rownum <= 65001
```

# Delete Unused Views

Each view may have a cost on performance, upgrade, and maintenance, even if it is not included in compound layout. Delete all unused views, including table views.



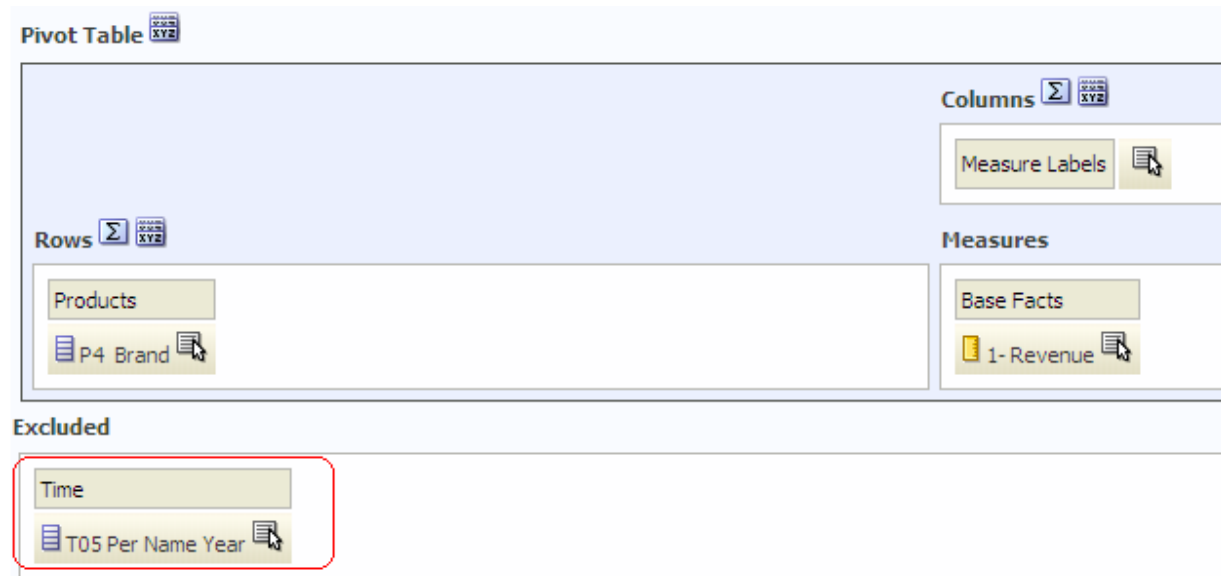
Compound Layout

| Title           |                    |           |           |           |           |
|-----------------|--------------------|-----------|-----------|-----------|-----------|
| Pivot Table (2) |                    |           |           |           |           |
|                 |                    | 2010 / 01 | 2010 / 02 | 2010 / 03 | 2010 / 04 |
| P4 Brand        |                    |           |           |           |           |
| BizTech         | 1- Revenue         | 131,500   | 399,089   | 747,614   | 955,474   |
|                 | 2- Billed Quantity | 14,568    | 40,852    | 86,217    | 106,656   |
| FunPod          | 1- Revenue         | 136,296   | 374,503   | 697,717   | 780,393   |
|                 | 2- Billed Quantity | 15,730    | 42,254    | 70,558    | 84,731    |
| HomeView        | 1- Revenue         | 86,366    | 171,872   | 493,209   | 583,220   |
|                 | 2- Billed Quantity | 8,338     | 19,527    | 50,865    | 55,712    |

# Excluded columns

Delete columns that are excluded from all views

- They increase the volume of data retrieved
- They make BI Server computing results at multiple levels of aggregation, impacting resources needed both on database and BI Server
- They may have an impact on results when using complex aggregations.



# Default values in Dashboard prompts

Put a default value in dashboard prompts.

- If you know what users will select most often, use it as the default value.
- If you do not know, then put a dummy value so that the report does not return anything. If necessary customize the « no result » view to tell users to select a value in prompt.
- There is nothing worse than executing a useless long query that returns all data from the database because there is no default filter. It costs a lot of resources both on the database and on BI Server.

# Hierarchies and attribute columns

Never mix hierarchies and attribute columns of the same dimension in a report. This leads to misunderstandings and unexpected behaviors, in particular when hierarchical prompts are used.

Note that selection steps generated by hierarchical prompts apply on hierarchies only, not on attribute columns.

Adding filters on attribute columns works fine though, even if you use the hierarchy in the report. But do not include the attribute column in the columns selected.

# Hierarchies and attribute columns



# Groups and Calculated Items

It is important to understand the differences between two types of selection steps: groups and calculated items.

- Performance considerations
  - Calculated items are computed on presentation server. They are executed on the (normally small) result set retrieved from BI Server. Usually they do not have any impact on performance.
  - Groups are computed on the database. They generate additional logical and physical queries. They have a significant impact on resources required on the database, and therefore on global performance.

# Groups and Calculated Items

- Functionality perspective
  - Calculated items formula are exactly applied on result set as they are. Aggregation rules used to compute the metrics on BI Server are not considered.
  - Groups generate a query with a filter based on members selected. Aggregation rules are applied on BI Server as usual.

| D4 Company                  | Avg Order Size |
|-----------------------------|----------------|
| Genmind Corp                | 2,488          |
| Stockplus Inc.              | 2,499          |
| Tescare Ltd.                | 2,511          |
| CalcItem Stockplus+Tescare  | 5,010          |
| Group Stockplus and tescare | 2,506          |

# Filter or Selection Step ?

Applying filters in reports may seem similar to selection steps. But is it really the same ? Let's study an example:

The screenshot displays the Oracle BI report configuration interface. At the top, three catalog panes are visible: 'Time' containing 'T05 Per Name Year', 'Offices' containing 'D4 Company', and 'Base Facts' containing '1- Revenue'. Below these, the 'Filters' section is expanded, showing a filter for 'T05 Per Name Year' with the condition 'is equal to / is in 2009; 2010', which is highlighted with a red rectangle. The 'Selection Steps' section is also expanded, showing a list of steps. Under the 'Measures' section, 'Base Facts - 1- Revenue' is selected. Under the 'Time - T05 Per Name Year' section, the first step is '1. Start with all members'. Under the 'Offices - D4 Company' section, the first step is '1. Start with 'Stockplus Inc.', which is highlighted with a red rectangle. Both of these red rectangles indicate that the filter and the selection step are applied at the same level in the report hierarchy.

| Catalog    | Item              |
|------------|-------------------|
| Time       | T05 Per Name Year |
| Offices    | D4 Company        |
| Base Facts | 1- Revenue        |

**Filters**

Add filters to the analysis criteria by clicking on Filter option for the specific c the catalog pane.

T05 Per Name Year is equal to / is in 2009; 2010

**Selection Steps**

List: ALL

**Measures**

- Base Facts - 1- Revenue

**Time - T05 Per Name Year**

1. Start with all members
2. Then, New Step...

**Offices - D4 Company**

1. Start with 'Stockplus Inc.'
2. Then, New Step...

# Filter or Selection Step ?

Looking at a simple table, it seems identical:

| Table    |                |            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|------------|
| T05 Per Name Year                                                                                                                                                                                                                                            | D4 Company     | 1- Revenue |
| 2009                                                                                                                                                                                                                                                         | Stockplus Inc. | 3,999,836  |
| 2010                                                                                                                                                                                                                                                         | Stockplus Inc. | 4,992,468  |

```
SET VARIABLE QUERY_SRC_CD='Report',PREFERRED_CURRENCY='USD';SELECT
  0 s_0,
  "A - Sample Sales"."Offices"."D4 Company" s_1,
  "A - Sample Sales"."Time"."T05 Per Name Year" s_2,
  SORTKEY("A - Sample Sales"."Offices"."D4 Company") s_3,
  "A - Sample Sales"."Base Facts"."1- Revenue" s_4,
  SUM("A - Sample Sales"."Base Facts"."1- Revenue" BY "A - Sample Sales"."Offices"."D4 Company") s_5
FROM "A - Sample Sales"
WHERE
  ("Time"."T05 Per Name Year" IN ('2009', '2010'))
  AND
  ("A - Sample Sales"."Offices"."D4 Company" IN ('Stockplus Inc.'))
ORDER BY 1, 4 ASC NULLS LAST
FETCH FIRST 65001 ROWS ONLY
```

# Filter or Selection Step ?

But see what happens when columns are removed from tables:

|                                                                                                                                                                                                                                                                   |                |            |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|------------|
| Table    |                |            |
| T05 Per Name Year                                                                                                                                                                                                                                                 | D4 Company     | 1- Revenue |
| 2009                                                                                                                                                                                                                                                              | Stockplus Inc. | 3,999,836  |
| 2010                                                                                                                                                                                                                                                              | Stockplus Inc. | 4,992,468  |

|                                                                                                                                                                                                                                                                       |            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| Table (2)    |            |
| T05 Per Name Year                                                                                                                                                                                                                                                     | 1- Revenue |
| 2009                                                                                                                                                                                                                                                                  | 15,000,000 |
| 2010                                                                                                                                                                                                                                                                  | 18,500,000 |

|                                                                                                                                                                                                                                                                       |            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| Table (3)    |            |
| D4 Company                                                                                                                                                                                                                                                            | 1- Revenue |
| Stockplus Inc.                                                                                                                                                                                                                                                        | 8,992,304  |

# Filter or Selection Step ?

```
SET VARIABLE QUERY_SRC_CD='Report',PREFERRED_CURRENCY='USD';SELECT s_0, s_1, s_2, s_3, s_4, s_5, s_6 FROM (
SELECT
    0 s_0,
    "A - Sample Sales"."Offices"."D4 Company" s_1,
    "A - Sample Sales"."Time"."T05 Per Name Year" s_2,
    SORTKEY("A - Sample Sales"."Offices"."D4 Company") s_3,
    "A - Sample Sales"."Base Facts"."1- Revenue" s_4,
    CAST(NULL AS DOUBLE) s_5,
    SUM("A - Sample Sales"."Base Facts"."1- Revenue" BY "A - Sample Sales"."Offices"."D4 Company") s_6
FROM "A - Sample Sales"
WHERE
    ("Time"."T05 Per Name Year" IN ('2009', '2010'))
    AND
    ("A - Sample Sales"."Offices"."D4 Company" IN ('Stockplus Inc.'))
UNION ALL
SELECT
    1 s_0,
    "A - Sample Sales"."Offices"."D4 Company" s_1,
    "A - Sample Sales"."Time"."T05 Per Name Year" s_2,
    SORTKEY("A - Sample Sales"."Offices"."D4 Company") s_3,
    "A - Sample Sales"."Base Facts"."1- Revenue" s_4,
    REPORT_SUM("A - Sample Sales"."Base Facts"."1- Revenue" BY "A - Sample Sales"."Time"."T05 Per Name Year") s_5,
    CAST(NULL AS DOUBLE) s_6
FROM "A - Sample Sales"
WHERE
    ("Time"."T05 Per Name Year" IN ('2009', '2010'))
) djm ORDER BY 1, 3 ASC NULLS LAST, 4 ASC NULLS LAST
FETCH FIRST 65001 ROWS ONLY
```

# Filter or Selection Step ?

## Filters:

- Are always applied on all views.

## Selection Steps:

- Are applied only if the corresponding column is included in the view.
- May generate additional logical and physical queries.

# General reports best practices

- Avoid using a filter based on another report.
- Use sub-totals and grand-totals only if necessary. Each total means an additional level of aggregation and has an impact on performance.
- Do not show more than 6 reports per page (depending on the performance of the reports).
- Do not put too many pages per dashboard, all pages should be visibles.

# General reports best practices

- Do not overuse dashboard conditions, it has a cost on performance.
- Dashboard should be as interactive as possible: column selectors, drill-down, guided navigation... Interactivity is one of the best assets of Oracle BI. Use it.
- Do not overuse the new expandables hierarchies as they tend to generate many physical queries. At least one query is necessary for each level shown, more if multiple fact LTS are used.

# Agenda

- Oracle BI Principles
- Repository design best practices
- Dashboards and reports design best practices
- 10g Upgrade considerations

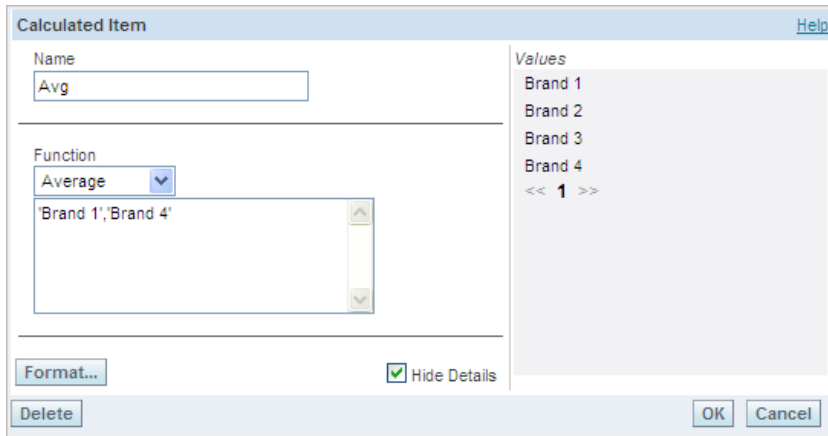
# 10g Upgrade considerations

There are many modifications on existing functionalities and algorithms between 10g and 11g.

Depending on the configuration, these modifications may change significantly the results in reports. They may impact both data and format of the report.

The list of examples mentioned here is NOT exhaustive.

# Calculated Items



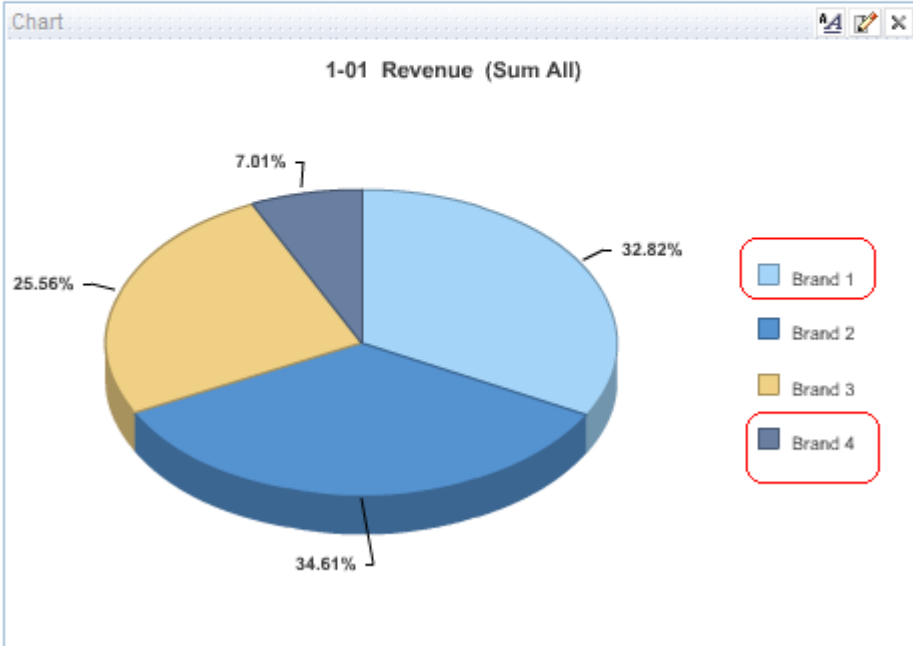
In 10g, calculated items were created in one pivot table only. In 11g, all calculated items are shown in all views.

- The format of the report is modified: columns that were in one pivot table only before the upgrade now appear in all views.
- If a calculated item had the option 'Hide details' selected, showing this column in other views changes completely the results.

# Calculated Items

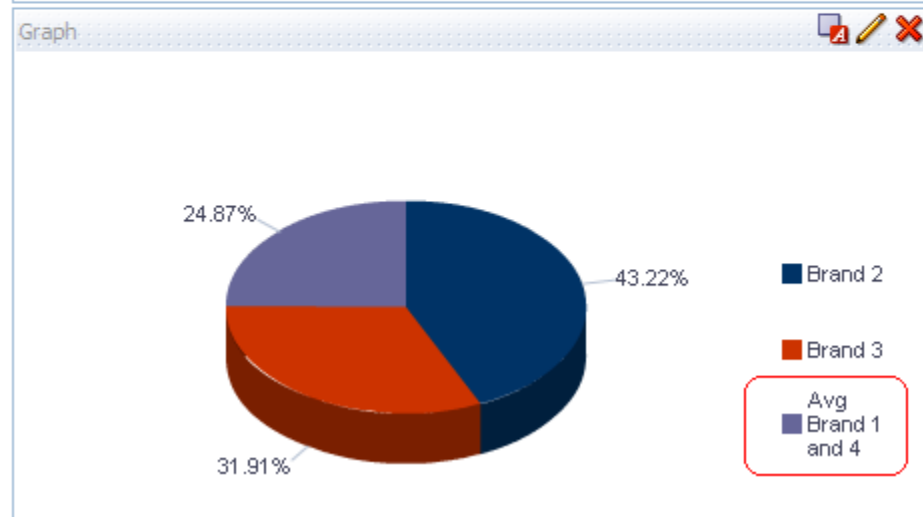
Pivot Table

|                   | 2007                   | 2008                   |
|-------------------|------------------------|------------------------|
| P04 Brand         | 1-01 Revenue (Sum All) | 1-01 Revenue (Sum All) |
| Brand 2           | 4,207,836              | 4,412,180              |
| Brand 3           | 2,835,863              | 3,528,129              |
| Avg Brand 1 and 4 | 2,163,791              | 2,795,727              |



Pivot Table

|                   | 2007                   | 2008                   |
|-------------------|------------------------|------------------------|
| P04 Brand         | 1-01 Revenue (Sum All) | 1-01 Revenue (Sum All) |
| Brand 2           | 4,207,836              | 4,412,180              |
| Brand 3           | 2,835,863              | 3,528,129              |
| Avg Brand 1 and 4 | 2,163,791              | 2,795,727              |



# Calculated Items

To replicate 10g behavior in 11g, you must:

- Add a new column identical to the one used to compute the calculated item.

| D4 Product                                                                        |                                                                                   | D0 Time                                                                           | F1 Revenue                                                                         |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| P04 Brand ↓↑                                                                      | P04 Brand ↓↑                                                                      | T05 Per Name Year ↓↑                                                              | 1-01 Revenue (Sum All) ↓↑                                                          |
|  |  |  |   |
|  |  |  |   |
|  |  |  |   |
|  |  |  |  |

- In all views except in the one that includes the calculated item, replace the old column by the new one.



Measure Labels

Measure Labels ☒ ☐

D4 Product

P04 Brand ☐ ☐

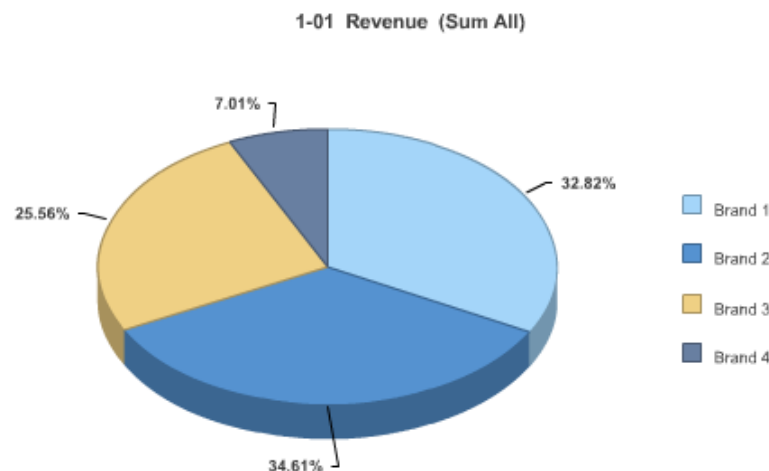
P04 Brand ☒ ☐

D0 Time

T05 Per Name Year ☐ ☐

F1 Revenue

1-01 Revenue (Sum All) ☐ ☒



# Calculated Items

To identify 10g reports with Calculated Items and option « hide details » selected, you can run a basic search in all 10g catalog files (\*. To select reports files only).

To identify all reports with calculated items, search for string:

`calcItem`

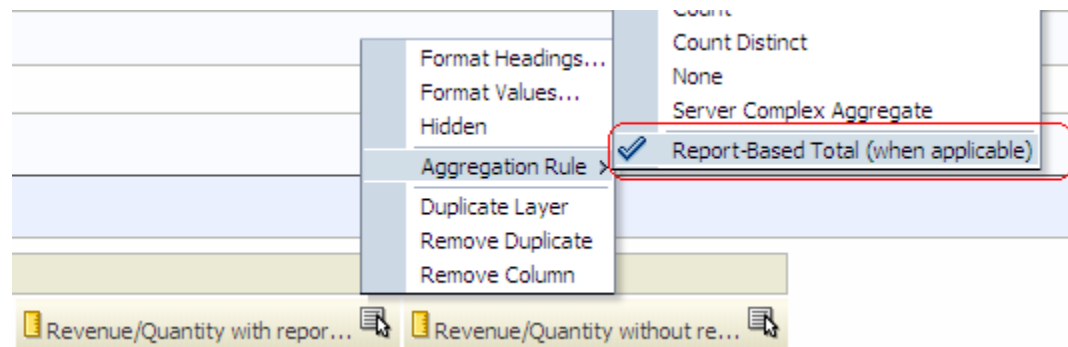
To identify reports with calculated items and option 'hide details' selected, search for:

`hideDetails="true"`

# Report-Based Totals

This option did not work in 10g and is fixed in 11g. It is selected by default.

- It may change significantly the values. 11g values are often better than 10g, but not always...
- Depending on the report, it may be hard to explain the results to users.
- It may be removed from tables and pivot tables, but not from charts.



# Report-Based Totals

What really does Report-Based Total option ?

Table



| P4 Brand           | 1- Revenue        | 2- Billed Quantity | Revenue/Quantity with report-based total 11g default | Revenue/Quantity without report-based total 10g |
|--------------------|-------------------|--------------------|------------------------------------------------------|-------------------------------------------------|
| BizTech            | 21,000,000        | 2,273,297          | 924                                                  | 924                                             |
| FunPod             | 17,500,000        | 1,913,386          | 915                                                  | 915                                             |
| <b>Grand Total</b> | <b>38,500,000</b> | <b>4,186,683</b>   | <b>920</b>                                           | <b>933</b>                                      |

Filters



2- Billed Quantity is greater than **1200000**

# Sort Orders

Sort orders in 11g are very often different from 10g.

- In 11g, sort defined in criteria tab is not necessarily applied to pivot tables, especially if the column sorted in criteria tab is excluded from the pivot table. The sort order has to be defined in the pivot table itself. Note that when the column that you want to use to sort is not in the pivot table, you have to add it, apply the sort, and then hide the column.

# Sort Orders

- A 10g bug is fixed when a sort key is created in RPD configuration (example: month name, sorted by month number). In 10g, this sort is applied only if the sort column is included in the report. But the sort order is defined on the RPD, so it should not be required to add an additional column in your report to apply the sort order. In 11g, even if the sort column is not included in your report, the sort key defined in RPD is still applied.

# Sort Orders

- In some circumstances, the sort order defined in 10g was not applied properly. For instance you select the sort order Ascending, and instead result is sorted in descending order. Users in 10g automatically adapted their sort orders in reports often without even noticing the issue, just by looking at results. This is fixed in 11g. So sometimes, in the report definition the sort order is Descending, in 10g the results are sorted Ascending, in 11g the results are sorted Descending.

# Sort Orders

- Sort in Graph: In 11g it is not possible to sort data in a graph using a column that is not included in the view. You have to add the column in the view (it can be hidden) to apply the sort order defined on this column.

# Total with Union/Running Aggregates

When a result set is computed with multiple queries (UNION) or with running aggregates (MAVG, MSUM, RSUM, RCOUNT, RMAX, RMIN), 11g does not apply any default aggregation rule for totals. The aggregation rule must be specified manually in tables/pivot tables.

This is necessary for totals, sub-totals, or when some columns are excluded from the view.

# Generated SQL

The SQL generation in 11g is different from 10g. The objective is to get more optimized SQL in 11g. However this may lead to differences in results if the RPD configuration or tables content is not consistent.

10g

| Discount   | Revenue     |
|------------|-------------|
| 1600000.00 | 57874000.00 |

```
WITH
SAWITH0 AS (select sum(T199.DISCNT_VALUE) as c1
from
    SAMP_REVENUE_F T199),
SAWITH1 AS (select sum(T1082.REVENUE) as c1
from
    WC_REVENUE_FA T1082)
select distinct SAWITH0.c1 as c1,
    SAWITH1.c1 as c2
from
    SAWITH0,
    SAWITH1
```

11g

| Discount  | Revenue    |
|-----------|------------|
| 1,600,000 | 50,000,000 |

```
WITH
SAWITH0 AS (select sum(T199.DISCNT_VALUE) as c1,
    sum(T199.REVENUE) as c2
from
    SAMP_REVENUE_F T199)
select D1.c1 as c1, D1.c2 as c2, D1.c3 as c3 from ( select distinct 0 as c1,
    D1.c1 as c2,
    D1.c2 as c3
from
    SAWITH0 D1 ) D1 where rownum <= 65001
```

# Analyzing Catalog Upgrade Logs

- The main log for catalog upgrade is  
`$MW_HOME/instances/instance1/diagnostics/logs/OracleBIPresentationServicesComponent/coreapplication_obips1/webcatupgrade0.log`
- In this log file, search for keyword « error ». Do not pay attention to other messages.
- For each error/warning there is a global error message with the path of the object (report, ibot...). Next there is the XML of the object before/after upgrade. The « after upgrade » XML is available only for warnings. After that, there is a detail error message describing the issue.

# Analyzing Catalog Upgrade Logs

- Datatype error: Type:InvalidDatatypeValueException, Message:Value '-2147483648' must be greater than or equal to MinInclusive '0'
  - The segment count has an invalid value.
- Required attribute 'guid' was not provided:
  - The iBot has been upgraded, but some of the recipients are not found in the list of users available in the authentication source. Check if the user is still able to authenticate, and if not, delete him from the webcat.

# Analyzing Catalog Upgrade Logs

- No character data is allowed by content model:
  - Report XML is invalid and should be fixed. Remove the unwanted characters.

There are many different error messages about invalid XML. Note that very often, it is faster to delete/recreate the report in 10g or in 11g than to spend a lot of time trying to fix the XML error.

# Graph Engine

The software used for the graph engine in 11g is not the same as the one in 10g.

Although the upgrade process tries to match as much as possible the graph properties selected in 10g with the ones available in 11g, a number of differences have to be expected.

11g graph engine has some new options that were not available in 10g, and some options that existed in 10g are not available anymore.

# Zoom to Data Range

There are new zoom options available in 11g. But the option “zoom to data range” that was available in 10g does not exist anymore in 11g.

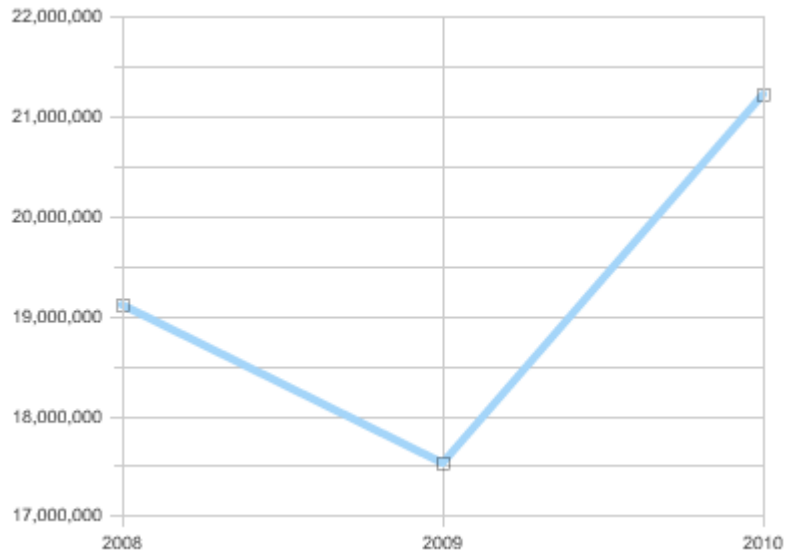
It is possible however to get a similar behavior by modifying global parameters. But then the option will be applied to all graphs. It is not possible in 11.1.1.6.0 to select this option for one report only.

To apply this global option, modify the following file in the current style path. For instance:

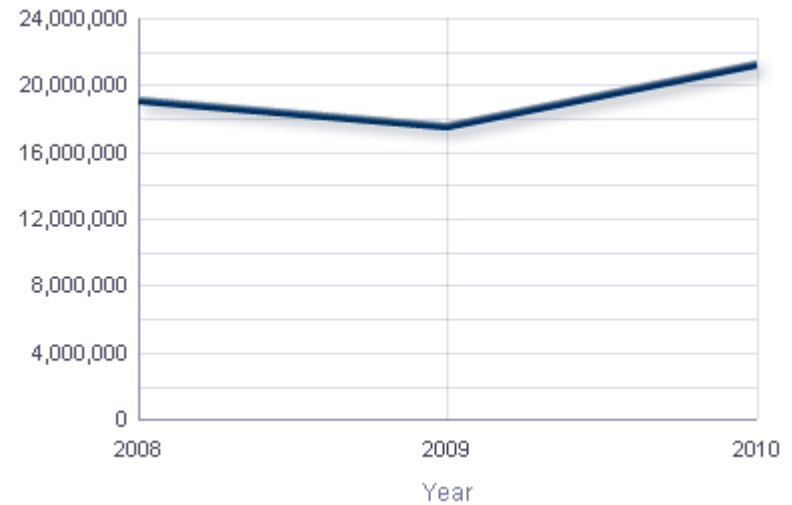
```
$MW_HOME/Oracle_BI1/bifoundation/web/msgdb/s_blafr/viewui/chart/dvt-graph-skin.xml
```

# Zoom to Data Range

10g



11g



# Zoom to Data Range

```
<Graph>
```

```
<SliceLabel>
```

*<!-- decimalDigitUsed is false here so that non-percentage pie slices do not pick up this value*

*The DVTChartProcessor sets decimalDigitUsed to true if this is a percentage pie slice -->*

```
<ViewFormat decimalDigit="2" decimalDigitUsed="false"/>
```

```
</SliceLabel>
```

```
<Y1Axis axisAutoScaledFromZero="false"/>
```

```
<Y2Axis axisAutoScaledFromZero="false"/>
```

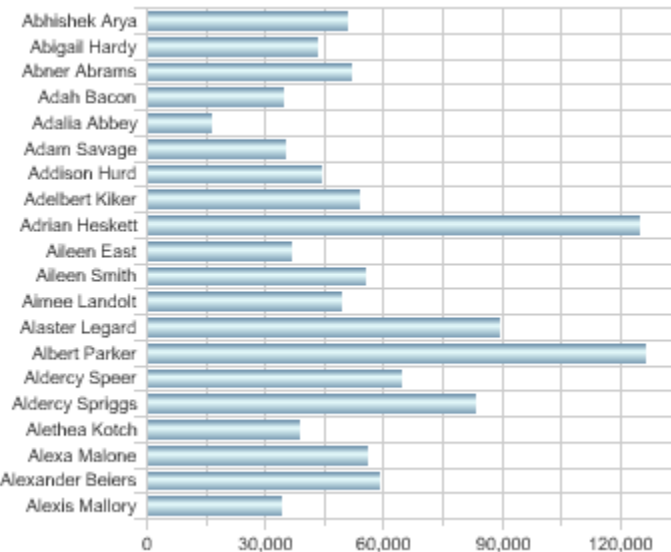
```
</Graph>
```

# Graph Engine, Miscellaneous

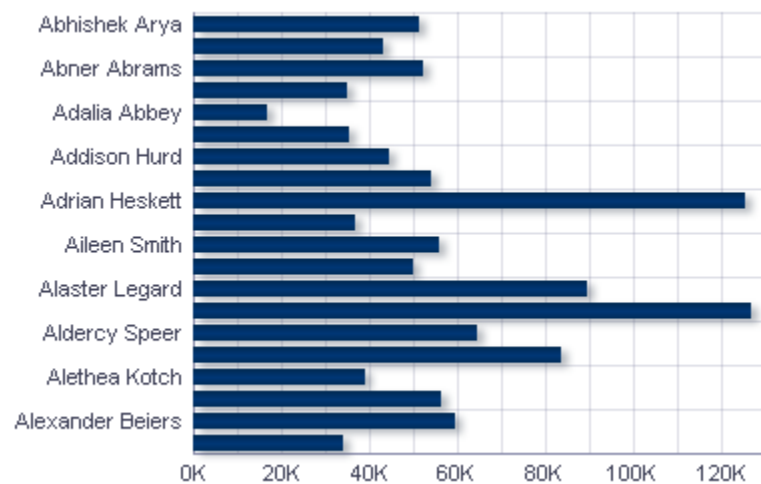
- The ranges for the numeric axis labels in graphs have changed from 10g to 11g due to a different automatic axis range calculation engine.
- Hidden columns used for labels in 10g are not displayed in 11g. If you have a column that is used as the label for a graph, but the column is hidden from the graph, then in 11g, the labels are not displayed.
- Some axis labels might be skipped as a result of the automatic label layout algorithm in use for 11g. The option that prevented skipping labels in 10g does not exist in 11g. It is possible to see all labels by modifying the size of the graph and labels.

# Graph Engine, Miscellaneous

10g



11g



## Graph properties

General

Style

Scale

Titles and Labels

Canvas Width 450 Pixels

Canvas Height 300 Pixels

## Format: Vertical Axis Labels

Display Options

Font Format

Font

Family Default (System)

Size

12

# Graph Engine, Miscellaneous

- You cannot rotate graph labels for the y-axis other than 0-90 or -90. You cannot perform 45-degree rotations.
- In 10g, graphs do not always honor criteria-level formats or other global data formats for columns. Data labels and numeric axis labels do not consistently follow this formatting. This issue has been addressed in 11g.
- In 10g, pie graphs display absolute values, including negative values. Negative values are interpreted as positive values and those slices are displayed. In 11g, slices are not displayed for negative values. When all the values are negative, the graph is not displayed. In 11g, the legend is displayed for negative values.

# Graph Engine, Miscellaneous

- When a stacked bar graph is upgraded from 10g to 11g, the order or position of the series might change. However, the legend view is upgraded without any change. This might cause a mismatch between the legend that is displayed in the legend view and the color that is displayed in the graph. To resolve this, either change the color in the graph or update the legend to match the color in the graph. In addition, the stacking order in the bar graph changes when you include a column in Vary Color By. For other cases, the order and coloring is maintained. The legend is incorrect or mismatched when you specify conditional formatting on the column in Vary Color By.

# Default number of rows

- In 10g the number of rows displayed was limited only in table view. In 11g this number of rows is limited in all views. Some parameters in instanceconfig.xml allow you to change this limit.
- Number of records that can be exported is limited as well. There is a parameter available in EM to set the maximum number of rows exported. But this does not override the maximum number of rows per view. So both parameters (MaxVisibleRows per view and global export limit) have to be modified.

# Default number of rows

```
<ServerInstance>
  <Views>
    <Table>
      <MaxCells>10000</MaxCells>
      <MaxVisiblePages>1000</MaxVisiblePages>
      <MaxVisibleRows>500</MaxVisibleRows>
      <MaxVisibleSections>25</MaxVisibleSections>
      <DefaultRowsDisplayed>30</DefaultRowsDisplayed>
    </Table>
    <Pivot>
      <MaxVisibleColumns>300</MaxVisibleColumns>
      <MaxVisiblePages>1000</MaxVisiblePages>
      <MaxVisibleRows>500</MaxVisibleRows>
      <MaxVisibleSections>25</MaxVisibleSections>
      <DefaultRowsDisplayed>30</DefaultRowsDisplayed>
    </Pivot>
    <Charts>
      <MaxVisibleColumns>2000</MaxVisibleColumns>
      <MaxVisiblePages>1000</MaxVisiblePages>
      <MaxVisibleRows>2000</MaxVisibleRows>
      <MaxVisibleSections>25</MaxVisibleSections>
      <JavaHostReadLimitInKB>4096</JavaHostReadLimitInKB>
    </Charts>
  </Views>
</ServerInstance>
```

# Font weight and alignment

If the font is not explicitly set, then it relies on the setting of the nearest ancestor element in HTML that has font size specified. Then the behavior of the font is non-deterministic and since if the parent element changed between 10g and 11g, this is impacted. For instance, the following text is in a dashboard page:

```
<span style="font-weight: bold;">Multi-segments choice</span>
```

In 10g, its closest ancestor element is (8pt) but now in 11g, it is 9pt. Thus you see the fonts in 9pt size. The solution is to add : font-size:8pt in the span so that it won't be affected by changes made to the framework.

# Hidden but included data is not displayed

In 10g, if a column is hidden but included in a pivot table, the data is displayed in the pivot table. In 11g, if the column is hidden at the criteria level, then the data is not displayed

# iBots => Agents

Options available in 11g agents are significantly different from 10g iBots, in particular for script management. So scripts options on 10g iBots are not available after the upgrade. They can still be executed, but cannot be modified.

A new agent must be created in 11g if you need to modify these options.

# Multiple column selectors

In 10g, column selectors included just a list of columns selected. In 11g however, column selectors also include the properties of each column available. If multiple column selectors include the same column they may be in conflict with each other after the upgrade.

Whenever possible, merge all column selectors to keep only one per report before the upgrade. If not possible, make sure at least that the same column is not included in two column selectors.

# Upgrading one report only

Note that it is possible to upgrade just one single report. This can be very useful for testing or to maintain consistency between 10g and 11g environments. To upgrade one report, copy/paste the XML from Advanced tab in Answer from one environment to the other. When the XML is applied in 11g environment, it is upgraded automatically.

# Spaces in Column Names

In 10g, when a column had leading or trailing spaces it created a warning in consistency checker. In 11g, this is considered as an error. So it is mandatory to remove all leading and trailing spaces from columns.

The main impact is that all reports using these columns have to be modified. The easiest solution is to use a simple text Search&Replace tool that can search and replace in multiple files at the same time. Just identify the column's previous name in a report XML and replace it by the new one.

# Clean 10g Catalog

A number of issues during catalog upgrade are caused by obsolete elements that should be deleted

- **Unused Views:** pivot tables may include calculated items. Even if the views are obsolete and are not included in compound layout, calculated item will be propagated to all other views during the upgrade. Delete all unused pivot table views before the upgrade.
- **Obsolete Reports:** old catalogs usually include many reports that are not used anymore. These reports may include errors that will have to be analyzed and fixed during the upgrade. The number of reports also impact the duration of the upgrade. Delete obsolete reports.

# Clean 10g Catalog

- Old Users: error messages will appear during catalog upgrade for each user in the catalog who cannot be authenticated anymore. These users' folders cannot be upgraded. They also increase upgrade duration significantly. Delete old accounts before the upgrade.
- As described in other slides, a number of reports have to be modified so that their behavior will not change in 11g. To reduce the duration of the freezing period (time between the last catalog extract from 10g and the 11g production roll-out), do as many modification as possible in 10g before the upgrade.

# UA or Manual Catalog Upgrade

The upgrade assistant copies the catalog first before starting the upgrade process. For big catalogs, a number of problem may happen during this phase (not enough space, network issue...). Even if the copy fails, the upgrade will start.

It is possible to copy the catalog and start the upgrade process manually instead.

# UA or Manual Catalog Upgrade

- Copy 10g catalog to a new location on 11g Server
- Stop 11g Presentation Server
- Update 11g catalog location using Enterprise Manager
- Add/Modify these flags to instanceconfig.xml:

```
<Catalog>
```

```
<UpgradeAndExit>true</UpgradeAndExit>
```

```
</Catalog>
```

- Start Presentation Server – This will upgrade the catalog and shutdown automatically
- Remove the flag true from the instanceconfig.xml
- Start Presentation Server again