



# Advanced MetaData Topics

Jeff McQuigg  
Sr. Architect, KPI Partners

Oracle Open World  
October 2, 2011



**Start Here**

**Membership Special:** Join by October 15 to become a member for only \$99!



Oracle  
Development  
Tools  
User Group

[www.odtug.com](http://www.odtug.com)

**A Real World User Group  
For Real World Developers**

**REGISTER TODAY!**

JW MARRIOTT.  
SAN ANTONIO HILL COUNTRY



**ODTUG**  
**Kscope12**



**SAN ANTONIO, TEXAS \* JUNE 24-28**

**Application Express \* Database \* Developer's Toolbox**  
**Business Intelligence \* Essbase \* Hyperion Applications**  
**Hyperion Business Content \* Fusion Middleware**

[www.kscope12.com](http://www.kscope12.com)

# KPI - The Leader

The Most Experienced Oracle BI & EPM Partner



200 Consultants, 115 Customers, Over 350 BI Projects

# Introduction

---

- Focus on a few OBI RPD MetaData modeling challenges
- Medium to advanced skill level
- Deep dives for thorough understanding
  - Not a “Tips & Tricks” session!
- Supports 10g & 11g



# About Jeff McQuigg

- Senior Architect at KPI Partners
- 9 years OBI consulting experience, 18 DW & BI years overall
  - Personally involved with over 35+ OBI projects, covering every role
- BI & OBI thought leader:
  - Best Practices BI blog at [GreatOBI.WordPress.com](http://GreatOBI.WordPress.com)
  - Moderator on *ITToolbox.com* and *OBIEE Enterprise Methodology Group*
  - Frequent Oracle Open World Speaker
- Kicking off world's largest HR Analytics project tomorrow!
- **Personal:** My nearly 3,000 bottles of beer are on display at [www.BrewPalace.com](http://www.BrewPalace.com)



# Agenda

---

- Combo Tables
- Make Dims & Facts Reach
- Non-Conformed Dimensions
- Canonical Time
- Selecting a Logical Table Source
- Controlling Join Paths

# Combo Tables

Table containing both **measures** and **attributes**

- Cuts down on # tables
- Easier ETL
- Fewer Joins for Queries
- Encountered heavily in source 3NF modeling - “Real Time” Reporting

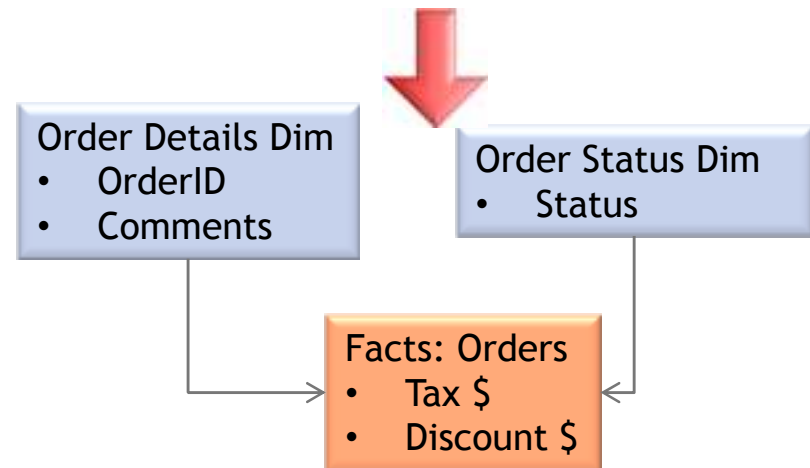
Common for degenerate keys

## Plan:

- Convert 1 Physical table into a Logical Star model in OBI

```
Create table W_ORDER_F (  
ORDER_WID           Number,  
PLACED_DATE_WID    Number,  
SHIPPED_DATE_WID   Number,  
SALES_PERSON_WID   Number,  
CUSTOMER_WID       Number,  
TAX_AMT            Number(10,2),  
DISCOUNT_AMT      Number(10,2),  
SOURCE_ORDER_ID    Varchar2(15),  
STATUS             Varchar2(10),  
COMMENTS           Varchar2(2000))
```

Physical



Logical



# Combo Tables

Use one physical alias

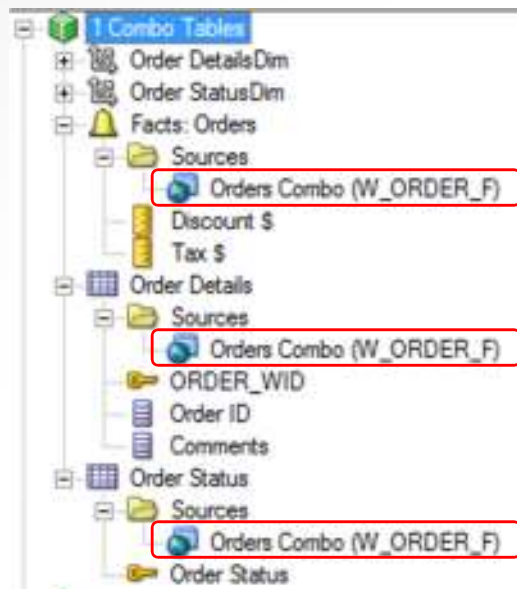
- Do not make separate aliases!

Follow modeling rules:

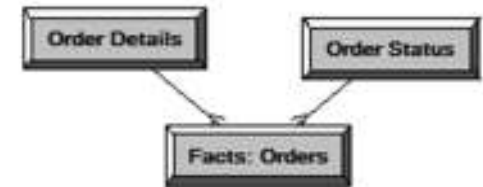
- Only aggregated metrics in the Fact table
- Put attributes into Dim table(s)

Resulting SQL is perfect

- Avoid self-joins!



## Business Model



Order ID	Order Status	Tax \$
2122	Closed	10.00
223234	Closed	5.00
23211343	Closed	25.00
2321312	Closed	32.00
3232	Closed	6.00
32323	Closed	7.00
3232312	Closed	12.00

```

select distinct 0 as c1,
    D1.c2 as c2,
    D1.c3 as c3,
    D1.c1 as c4
from
    (select sum(T75.TAX_AMT) as c1,
        T75.SOURCE_ORDER_ID as c2,
        T75.STATUS as c3
    from
        W_ORDER_F T75 /* Orders Combo (W_ORDER_F) */
    group by T75.SOURCE_ORDER_ID, T75.STATUS
    ) D1
order by c2, c3
    
```

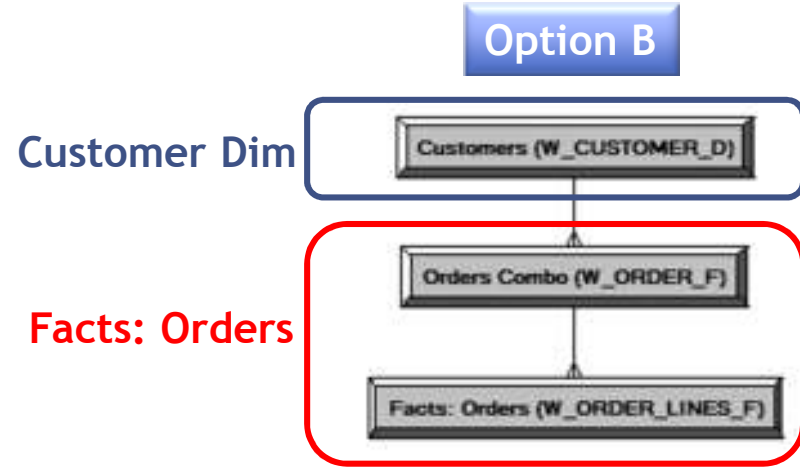
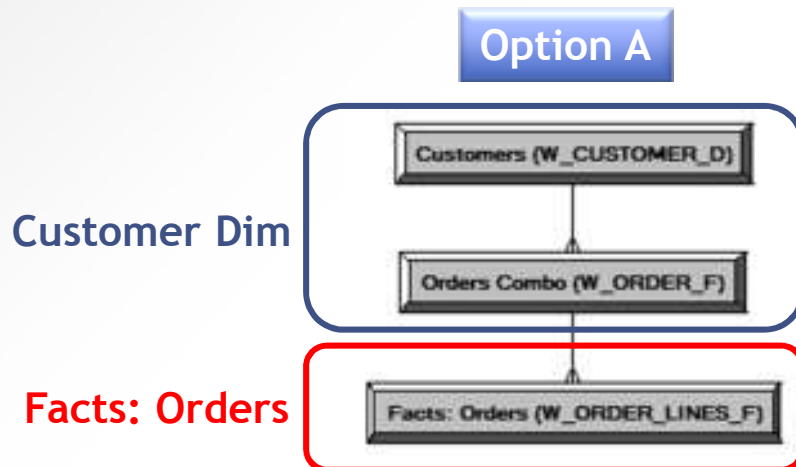
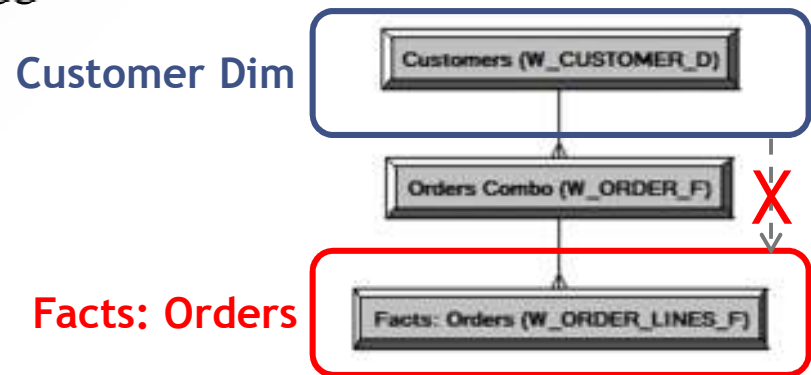
# Making Dims & Facts Work Together

No direct link between Logical Table Sources

- LTSs have to link to work
- However a 3<sup>rd</sup> table completes the link

Adding the W\_ORDER\_F table to one of the Logical Table Sources (LTS) will make then link

- *But to which Logical Table Source?*



# Making Dims & Facts Work Together

## Option A - Add to Customers

Name:

Disabled  Dimension Browse

Map to these tables:

- "OraDB"."JEFF"."Orders Combo (W\_ORDER\_F)"
- "OraDB"."JEFF"."Customers (W\_CUSTOMER\_D)"

## Option B - Add to Facts

Name:


Disabled  Dimension Browse

Map to these tables:


- "OraDB"."JEFF"."Facts: Orders (W\_ORDER\_LINES\_F)"
- "OraDB"."JEFF"."Orders Combo (W\_ORDER\_F)"

### Select Customer.Name:

```
select distinct T321.CUSTOMER_NAME as c1
from W_CUSTOMER_D T321 /* Customers (W_CUSTOMER_D) */ ;
```




```
select distinct T321.CUSTOMER_NAME as c1
from W_CUSTOMER_D T321 /* Customers (W_CUSTOMER_D) */ ;
```




### Select Customer.Name, Facts:Order.Order Amt:

```
select distinct
  T315.QUANTITY * 1.0 * T315.UNIT_PRICE as c1,
  T321.CUSTOMER_NAME as c2
from
  W_ORDER_F T75 /* Orders Combo (W_ORDER_F) */ ,
  W_CUSTOMER_D T321 /* Customers (W_CUSTOMER_D) */ ,
  W_ORDER_LINES_F T315 /* Facts: Orders (W_ORDER_LINES_F) */
where ( T75.CUSTOMER_WID = T321.CUSTOMER_WID
and T75.ORDER_WID = T315.ORDER_WID )
```




```
select distinct
  T315.QUANTITY * 1.0 * T315.UNIT_PRICE as c1,
  T321.CUSTOMER_NAME as c2
from
  W_ORDER_F T75 /* Orders Combo (W_ORDER_F) */ ,
  W_CUSTOMER_D T321 /* Customers (W_CUSTOMER_D) */ ,
  W_ORDER_LINES_F T315 /* Facts: Orders (W_ORDER_LINES_F) */
where ( T75.CUSTOMER_WID = T321.CUSTOMER_WID
and T75.ORDER_WID = T315.ORDER_WID )
```




### Select Customer.Name, Facts:SRs.#SRs:

```
select T74.CUSTOMER_NAME as c1,
  count(T131.TICKET_NUM) as c2
from
  W_CUSTOMER_D T74,
  W_ORDER_F T85,
  W_SR_F T131
where ( T74.CUSTOMER_WID = T85.CUSTOMER_WID and
T74.CUSTOMER_WID = T131.CUSTOMER_WID )
group by T74.CUSTOMER_NAME
```



```
select count(T455.TICKET_NUM) as c1,
  T321.CUSTOMER_NAME as c2
from W_CUSTOMER_D T321 /* Customers (W_CUSTOMER_D) */ ,
  W_SR_F T455 /* SR Combo (W_SR_F) */
where ( T321.CUSTOMER_WID = T455.CUSTOMER_WID )
group by T321.CUSTOMER_NAME
```



# Making Dims & Facts Work Together

OBI will frequently add lower grains to queries when in the same LTS

- W\_ORDERS\_F is lower grain than W\_CUSTOMER\_D

Safe: Adding a higher grain

- W\_ORDERS\_F is higher grain than W\_ORDER\_LINES\_F

*“Extend the reach of the Fact”*

- Reach more dimensions
- Denormalizing FKs
- Think of adding Sockets

This is how you would build it in the first place:

```
Create table W_ORDER_LINES_F (
```

ORDER_WID	Number,
PLACED_DATE_WID	Number,
SHIPPED_DATE_WID	Number,
SALES_PERSON_WID	Number,
<b>CUSTOMER_WID</b>	<b>Number,</b>
PRODUCT_WID	Number,
QUANTITY	Number,
UNIT_PRICE	Number(10,2),
SOURCE_LINE_ID	Varchar2(15));

# Non-Conformed Dimensions

## Scenario:

- Multiple fact tables in a query
- A dimension only works with one of the facts
- Report does not work!

**Selected Columns**  
Double click on column names in the Subject Areas pane to add them to the sorting, or delete by clicking or hovering over the button next to its name.

Customers	SR Status	Facts: Orders	Facts: SRs
Name	SR Status	Order Amt	# SRs

**Filters**  
Add filters to the analysis criteria by clicking on Filter option for the specific clicking on add button after selecting its name in the catalog pane.

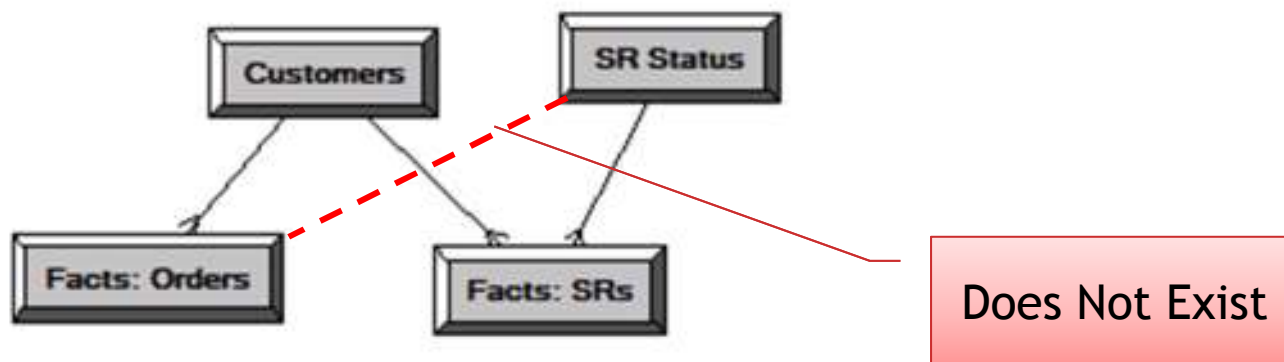
SR Status is equal to / is in Open

Table

Name	SR Status	Order Amt	# SRs
Aaron Turner	Open		1
Amy Harris	Open		1
Chamonay Benjamin	Open		1
Mike Gallagher	Open		1

# Non-Conformed Dimensions

- Why?
  - Basic Structure of the request is:
    - Customer, SR Status, Order Amt, # SRs
  - OBI breaks it down as:
    1. # SRs *by* Customer and SR Status
    2. Order Amt *by* Customer and SR Status
- ➔ Model does not support Order Amt by SR Status





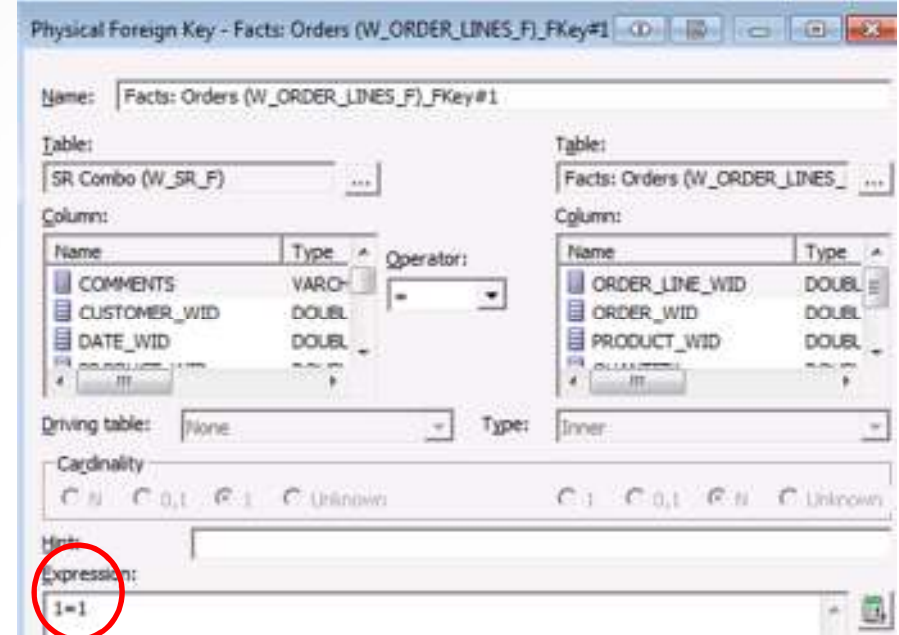
# Non-Conformed Dimensions

Solution #1: Fake a physical join between the two tables

- Cartesian Product
- “Tricking OBI”

Add to Business Model

- Joins
- Content Tab



Logical Table Source - Facts: Orders (W\_ORDER\_LINES\_F)

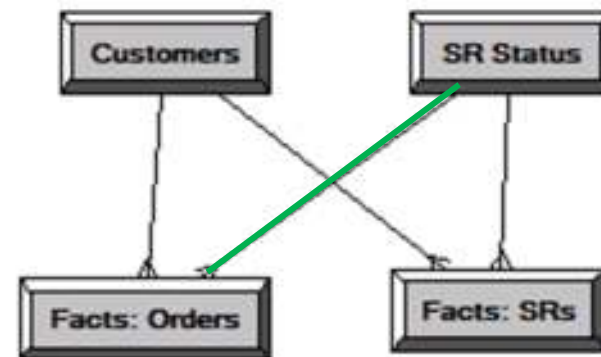
General | Column Mapping | Content | Parent-Child Settings

Aggregation content, group by: Logical Level

Show mapped  Show unmapped

Logical Dimension	Logical Level	
CustomersDim	Customers (W_CUSTOM	✗
Order StatusDim	Order Status Detail	✗
Sales PersonDim	Sales Person Detail	✗
SR StatusDim	SR Status Detail	✗

Business Model



# Non-Conformed Dimensions

Now it is able to issue a query

Note the table with SR Status (W\_SR\_F) is included

- SR Status to W\_ORDER\_F:
- No Joins → Cartesian Product!

Be very careful with this solution



**Must** filter to **one record** to eliminate Cartesian Product (Over-counting)

This example contains a special problem:

- Multiple W\_SR\_F records will have Status = 'Open'
- Over-counting should occur
- If a normalized Dimension table with 1 record for 'Open', then safe

Name	SR Status	Order Amt	# SRs
Aaron Turner	Open	7600.00	1
Amy Harris	Open	1550.00	1
Chamonay Benjamin	Open	7000.00	1
Mike Gallagher	Open	1125.00	1

```
select T321.CUSTOMER_NAME as c1,  
       T455.STATUS as c2,  
       sum(T315.QUANTITY * 1.0 * T315.UNIT_PRICE) as c3  
from  
       W_SR_F T455 /* SR Combo (W_SR_F) */ ,  
       W_CUSTOMER_D T321 /* Customers (W_CUSTOMER_D) */ ,  
       W_ORDER_F T75 /* Orders Combo (W_ORDER_F) */ ,  
       W_ORDER_LINES_F T315 /* Facts: Orders (W_ORDER_LINES_F) */  
where ( T75.CUSTOMER_WID = T321.CUSTOMER_WID  
       and T75.ORDER_WID = T315.ORDER_WID  
       and T321.CUSTOMER_WID = T455.CUSTOMER_WID  
       and T455.STATUS = 'Open' )  
group by T321.CUSTOMER_NAME, T455.STATUS  
order by c1, c2
```

```
select T321.CUSTOMER_NAME as c1,  
       T455.STATUS as c2,  
       count(T455.TICKET_NUM) as c3  
from  
       W_CUSTOMER_D T321 /* Customers (W_CUSTOMER_D) */ ,  
       W_SR_F T455 /* SR Combo (W_SR_F) */  
where ( T321.CUSTOMER_WID = T455.CUSTOMER_WID  
       and T455.STATUS = 'Open' )  
group by T321.CUSTOMER_NAME, T455.STATUS  
order by c1, c2
```

# Non-Conformed Dimensions

## Solution #2: Use Totals Level Based Metric

- Set the SR Status Dimension to **Total** for *every* metric used
- Tells metrics to *ignore* a dimension
- Both queries are clean

Logical Column - Order Amt

General | Column Source | Aggregation | Levels

Logical Dimension	Logical Level
CustomersDim	
Order StatusDim	
Sales PersonDim	
SR StatusDim	SR Status Total

Name	SR Status	# SRs	Order Amt
Aaron Turner	Open	1	7600.00
Amy Harris	Open	1	1550.00
Bryant Meyer		0	1750.00
Chamonay Benjamin	Open	1	7000.00
Jennifer Choi		0	450.00
Mike Gallagher	Open	1	1125.00
Sally Chen		0	15225.00

```
select T321.CUSTOMER_NAME as c1,
       T455.STATUS as c2,
       count(T455.TICKET_NUM) as c3
from
  W_CUSTOMER_D T321 /* Customers (W_CUSTOMER_D) */ ,
  W_SR_F T455 /* SR Combo (W_SR_F) */
where ( T321.CUSTOMER_WID = T455.CUSTOMER_WID
       and T455.STATUS = 'Open' )
group by T321.CUSTOMER_NAME, T455.STATUS
order by c1
```

```
select T321.CUSTOMER_NAME as c1,
       sum(T315.QUANTITY * 1.0 * T315.UNIT_PRICE) as c2
from
  W_CUSTOMER_D T321 /* Customers (W_CUSTOMER_D) */ ,
  W_ORDER_F T75 /* Orders Combo (W_ORDER_F) */ ,
  W_ORDER_LINES_F T315 /* Facts: Orders (W_ORDER_LINES_F) */
where ( T75.CUSTOMER_WID = T321.CUSTOMER_WID
       and T75.ORDER_WID = T315.ORDER_WID )
group by T321.CUSTOMER_NAME
order by c1
```

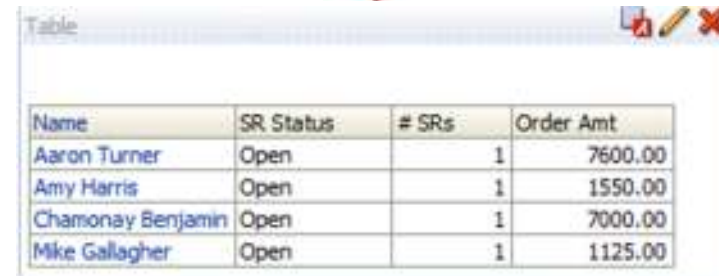
# Non-Conformed Dimensions

- Other filtering may be needed
- Filters on SR Status are *meaningless* to the Orders Query
- Possibly add a different filter



Name	SR Status	# SRs	Order Amt
Aaron Turner	Open	1	7600.00
Amy Harris	Open	1	1550.00
Bryant Meyer		0	1750.00
Chamonay Benjamin	Open	1	7000.00
Jennifer Choi		0	450.00
Mike Gallagher	Open	1	1125.00
Sally Chen		0	15225.00

SR Status = 'Open'



Name	SR Status	# SRs	Order Amt
Aaron Turner	Open	1	7600.00
Amy Harris	Open	1	1550.00
Chamonay Benjamin	Open	1	7000.00
Mike Gallagher	Open	1	1125.00

SR Status = 'Open'  
and # SRs > 0

# Non-Conformed Dimensions

## 1=1 Join

### Pros

- Very easy to configure
- Works very well for filtered reports

### Cons

- Potential '*nuclear bomb*', especially for Answers
- More difficult for Database optimizer

## Level Based Metrics

### Pros

- Safer to use

### Cons

- Metric by Metric config
  - More development work
- Harder to QA

- Consider which technique to use for your system
- You may need both!

# Canonical Time

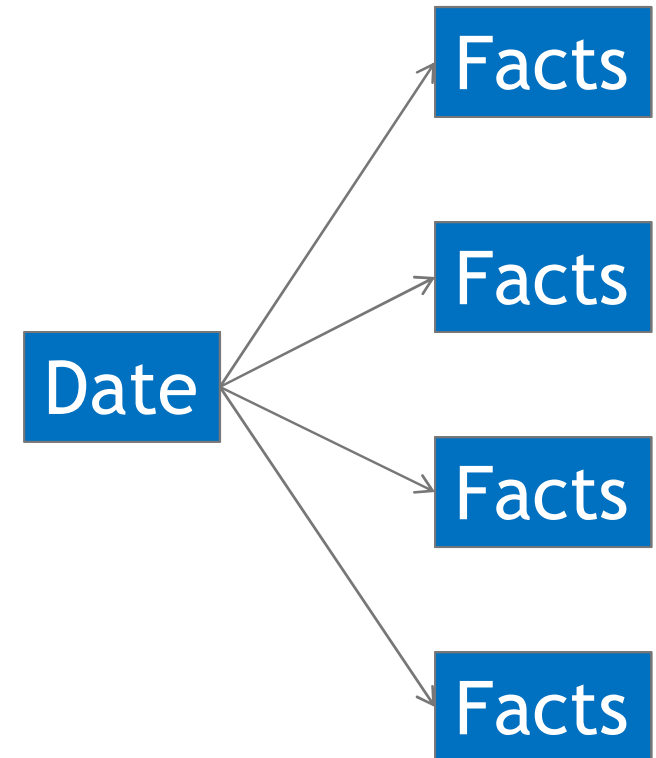
## One Date Dimension:

- Is conformed across other facts
  - *Order Open Date* is not conformed
  - Mix multiple metrics along a common time period
- Provides expanded analysis functionality
  - More “BI like” – more big picture, more enterprise
- Requires less configuration
  - 1 Date Dimension instead of many
- Is easier for Business users to understand

## Besides...

- The real world has only 1 date anyway!

Canonical Time does the same thing but in a single fact table

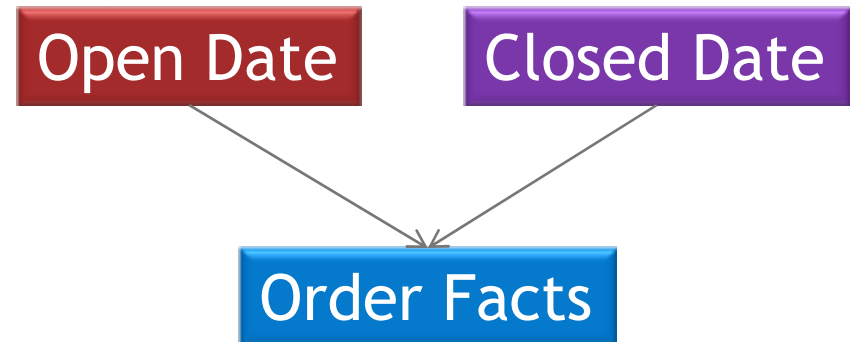




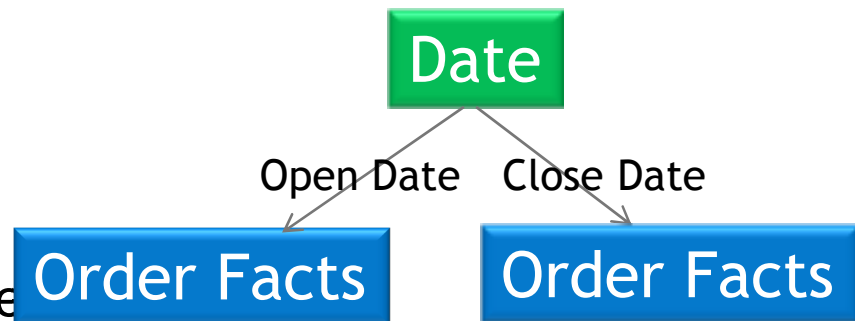
# Canonical Time

- Essentially a super-conformed Date dimension
  - Transactions with many dates
  - SR Transaction: Open Date & Close Date
  - Order Transaction: Create Date, Planned Shipped Date, Actual Shipped Date, etc.
- Goal: Only 1 Date dimension
  - Instead of: “Open Date Dim” & “Close Date Dim”
  - Simply Use: “Date Dim”
  - BI Apps do this to a large degree

## Multi-Date



## Canonical-Date



# Canonical Time

## Example 1: Show SR activity (# Opened & # Closed) by Month

### Non-Canonical:

# Open SRs = Filter(# SRs using SR Status = 'Open')

- **Volatile**
- For a SR opened in Jan and closed in Feb:
  - Report run in Jan shows Opened
  - Report run in Feb shows Closed
- History will be restated each day
- Status based metrics will not show actual activity
- Tie-in to month will be one of the dates but not both

### Canonical:

Build # SRs Opened & # SRs Closed with a Canonical Date

Use Open & Close Date FKs on the SR Fact

- **Non-Volatile**
- Independent of each other

Note the subtle language distinction:

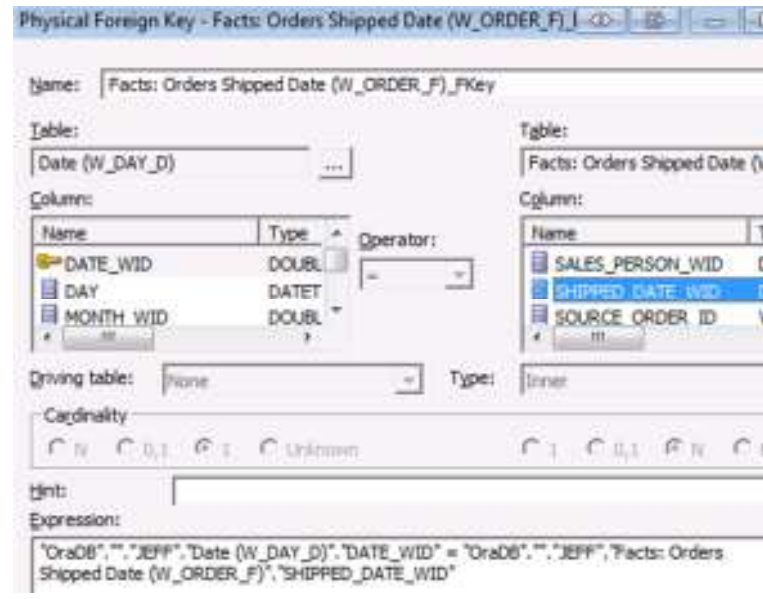
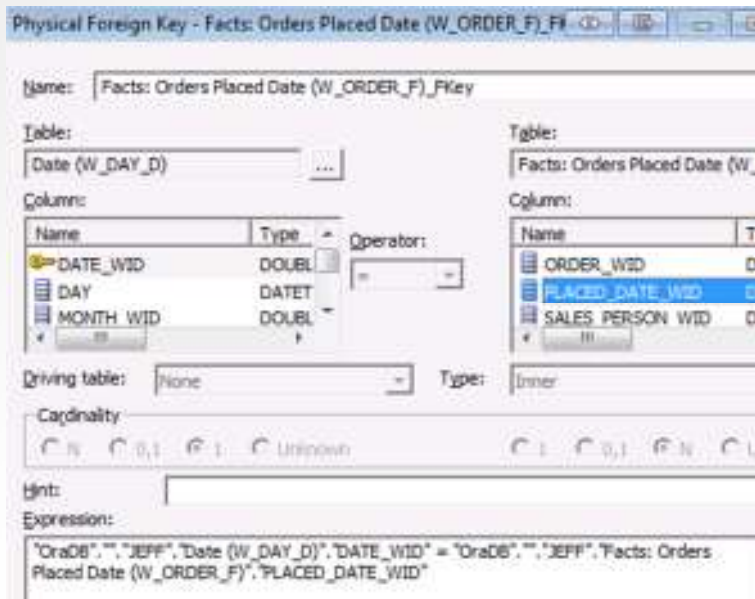
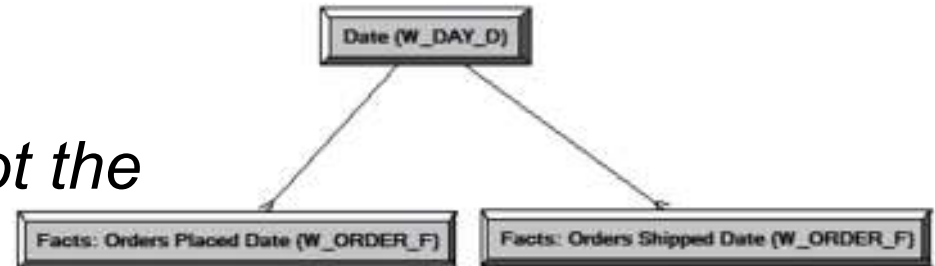
# Open SRs  $\neq$  # SRs Opened

# Canonical Time

Example 2: Show Order activity (# Opened & # Shipped) by month

Physical Layer Configuration:

- Make aliases for the *fact*, not the *dim*
- Join on appropriate Fact Date field



# Canonical Time

The Business Model's LTSs mirror the physical layer

- Each metric depends on a specific time linkage



Logical column to physical column mapping:

Logical Column	Expression		Physical Table
# Orders Shipped	SHIPPED_DATE_WID	✗	Facts: Orders Placed D
# Orders Placed		✗	

Year / Month	# Orders Shipped	# Orders Placed
2011 / 01	5	4
2011 / 02	8	8
2011 / 03	2	3

```
select T861.YEAR_MONTH as c1,
       count(T894.PLACED_DATE_WID) as c2
from
  W_DAY_D T861 /* Date (W_DAY_D) */ ,
  W_ORDER_F T894 /* Facts: Orders Shipped Date
(W_ORDER_F) */
where ( T861.DATE_WID = T894.SHIPPED_DATE_WID )
group by T861.YEAR_MONTH
order by c1
```

```
select T861.YEAR_MONTH as c1,
       count(T883.SHIPPED_DATE_WID) as c2
from
  W_DAY_D T861 /* Date (W_DAY_D) */ ,
  W_ORDER_F T883 /* Facts: Orders Placed Date
(W_ORDER_F) */
where ( T861.DATE_WID = T883.PLACED_DATE_WID )
group by T861.YEAR_MONTH
order by c1
```

Essentially making logical Snapshot fact tables

- Be careful with performance when many queries issued
- Similar concept as the old technique for Time Series

# LTS Selection

---

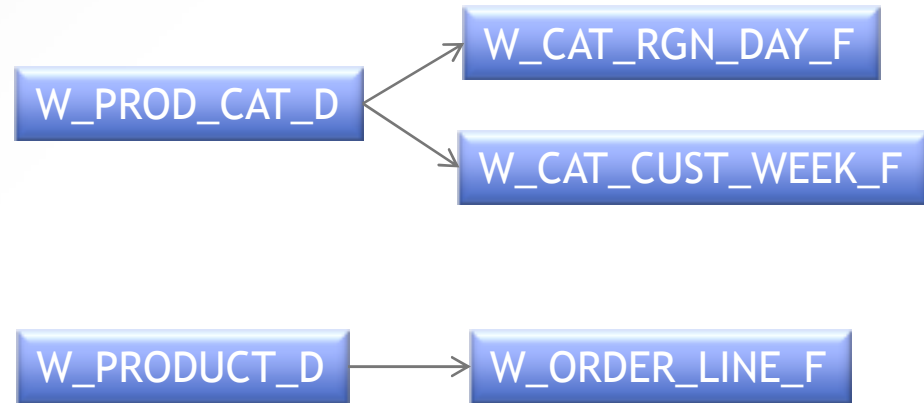
- Discuss how OBI selects a LTS
  - “Aggregate Navigation”
- Fundamental to understanding OBI
  - Required for advanced configuration
- Some examples will help understanding in order to manipulate the system to solve your own unique problems

# LTS Selection

Basic steps in selecting a LTS:

1. Identify all LTSs which have:
  - a. the proper fields mapped
  - b. “work” together – joins, content tab
  - c. Considering Fragmentation Content
2. Pick the Highest grained Fact LTS
3. Pick remaining Dim LTSs

## Dim LTSs



## Fact LTSs

Dimensional LTS Ordering:

- 10g: Dimensional ordering is *meaningless*
- 11g: Meaningful

Fact LTS Ordering:

- Meaningful in both 10g & 11g



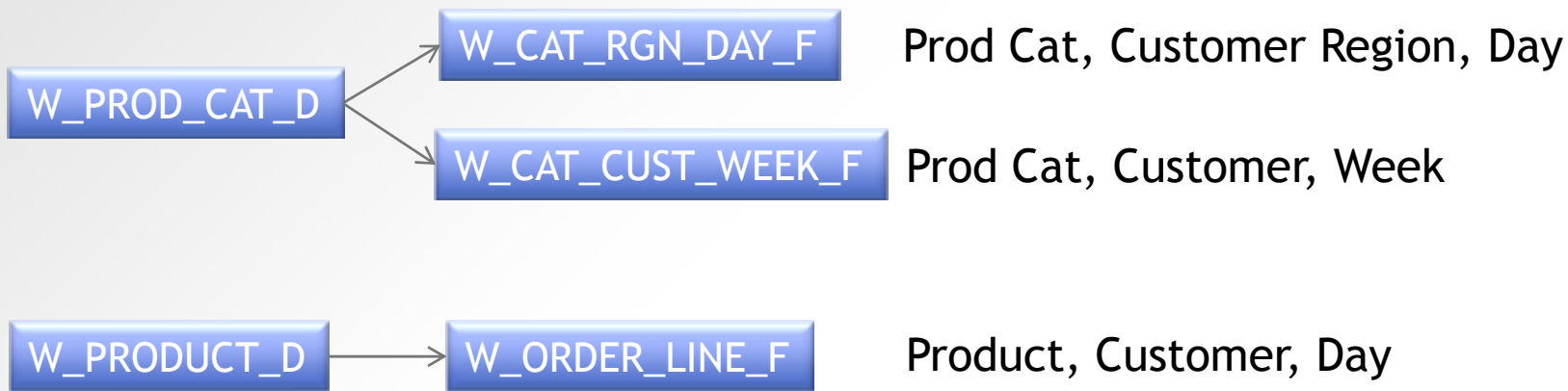
# LTS Selection

## Example: Select Product.Product Category;

### Dim LTSs

### Fact LTSs

### Fact Grain



- Which is the highest grained Fact source?
- OBI knows the following from its hierarchies:

Product Category  
Product

Customer Region  
Customer

Week  
Day



Higher Grain

- Eliminates W\_ORDER\_LINE\_F immediately  
∴ W\_PRODUCT\_D is eliminated

# LTS Selection

- But there are Fact 2 LTSs remaining
- LTS #1 has 1 dim > LTS #2 and 1 dim < LTS #2
  - ∴ OBI does not know which is best
- Need to quantify sizes to help
  - Use # elements at this level

Dimension	Level	# Elements
Product	Category	5
	Details	500
Customer	Region	4
	Details	1000
Date	Week	52
	Day	365

Fact Table	Grain	Expected # Elements / Size
W_CAT_RGN_DAY_F	Prod Cat, Customer Region, Day	5 X 4 X 365 = <b>7,300</b>
W_CAT_CUST_WEEK_F	Prod Cat, Customer, Week	5 X 1000 X 52 = 260,000

If Region to Customer was 4:25 instead of 4:1000:

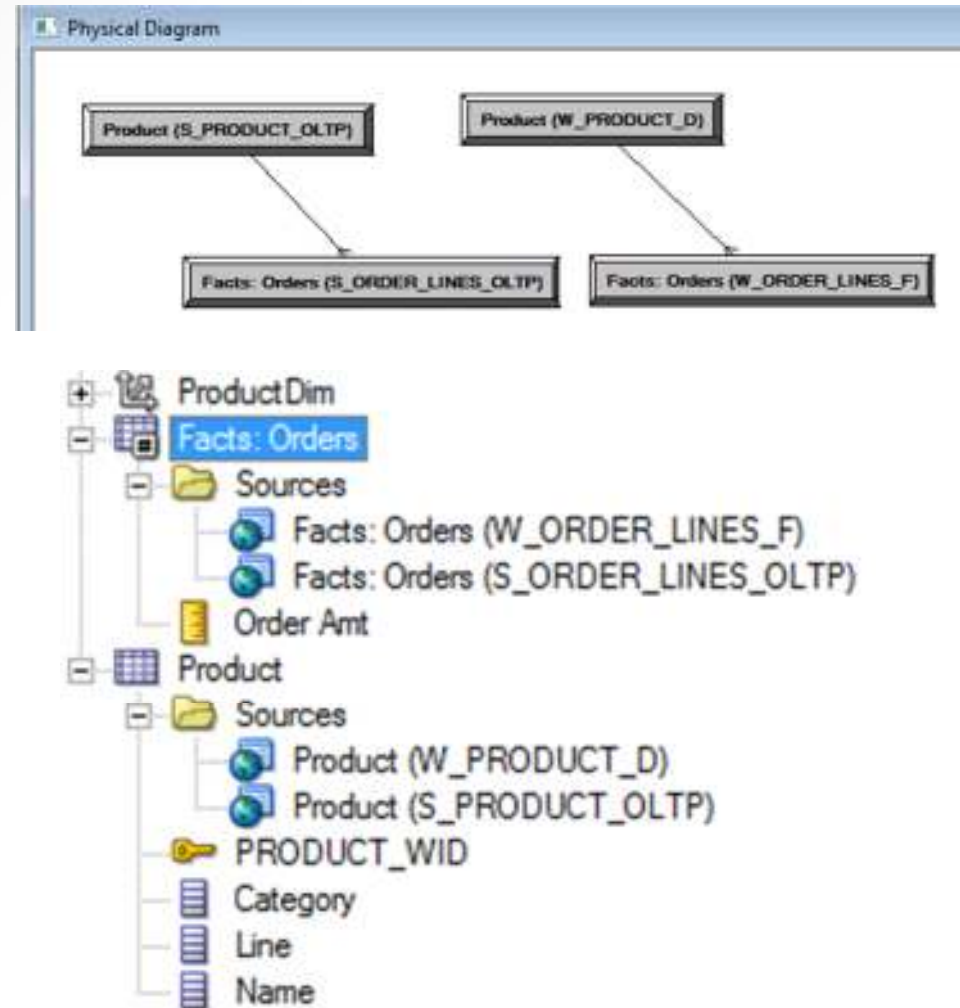
Fact Table	Grain	Expected # Elements / Size
W_CAT_RGN_DAY_F	Prod Cat, Customer Region, Day	5 X 4 X 365 = 7,300
W_CAT_CUST_WEEK_F	Prod Cat, Customer, Week	5 X 25 X 52 = <b>6,500</b>

# LTS Selection

- In many cases you do not need *# elements*
  - Only if many fact & aggregate tables
- What if 2 Fact LTSs have the same grain?
  - Adding a new dimension will lower it
  - i.e., 5 dims *is higher than* 6 dims
- Example: federation across a DM and OLTP
  - Want the bulk of queries to hit the DM tables
  - Grains are identical

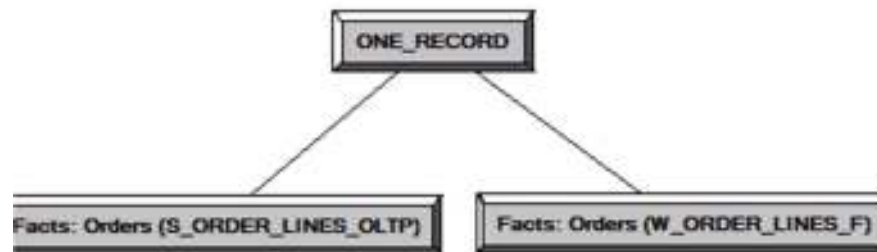
# LTS Selection

- Physical models are identical
- Both Physical models mapped into same Business Model
  - Exact same grains
- Reminder:
  - 11g: dimensional source ordering important
  - 10g: meaningless



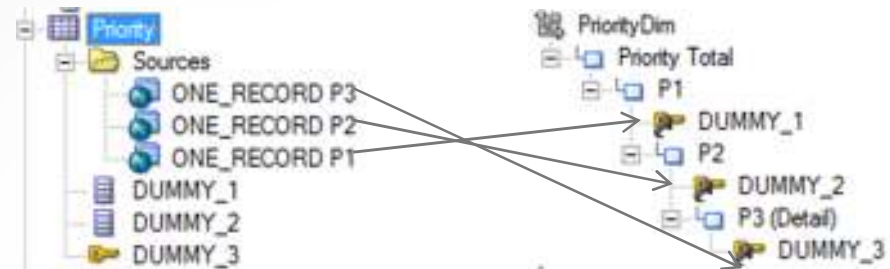
# LTS Selection

- Additional LTS control can be achieved by adding a dummy dimension
  - Even further control is possible with a hierarchy
  - Essentially a “priority” dimension
- Map in a 1 record dummy table with a few fields
- Cartesian join (1=1) them to Fact tables



# LTS Selection

- Make a dimension
  - Multiple LTSs, each at a different grain
  - Each column maps only to 1 LTS
- Assign each Fact LTS to your desired priority via a level



Logical Table Source - ONE\_RECORD P1

General | Column Mapping | Content | Parent-Child Settings |

Show mapped columns

Logical column to physical column mapping:

Logical Column	Expression	Physical Table
DUMMY_3		
DUMMY_2		
DUMMY_1	DUMMY_1	ONE_RECORD

Logical Table Source - Facts: Orders (W\_ORDER\_...)

General | Column Mapping | Content | Parent-Child Settings |

Aggregation content, group by: Logical Level

Show mapped  Show unmapped

Logical Dimension	Logical Level
PriorityDim	P3 (Detail)
ProductDim	Product (Details)

Logical Table Source - Facts: Orders (S\_ORDER\_...)

General | Column Mapping | Content | Parent-Child Settings |

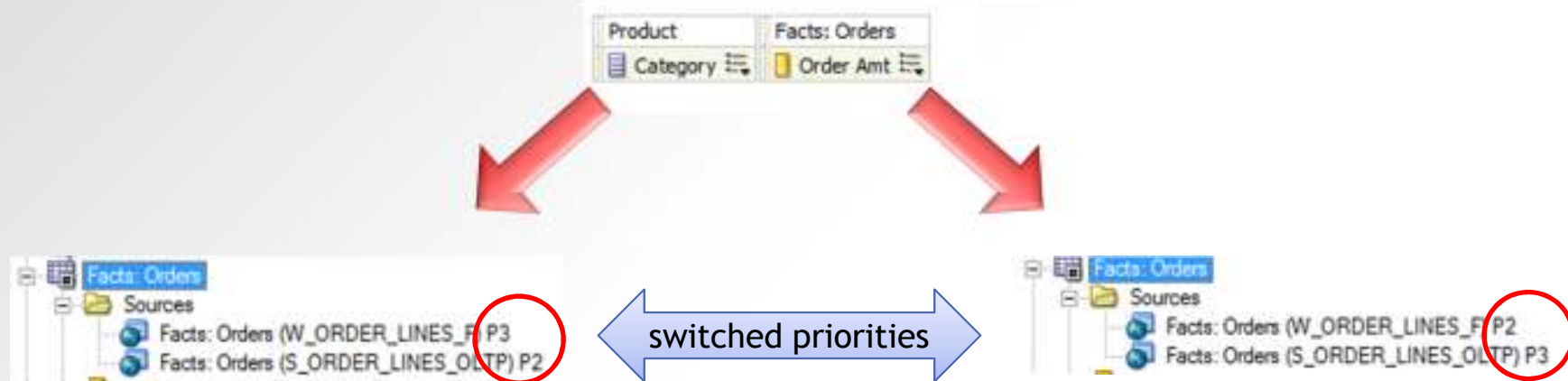
Aggregation content, group by: Logical Level

Show mapped  Show unmapped

Logical Dimension	Logical Level
PriorityDim	P2
ProductDim	Product (Details)



# LTS Selection



```
select sum(T1008.UNIT_PRICE * 1.0 *  
T1008.QUANTITY) as c1,  
       T1014.CATEGORY as c2  
from  
       S_PRODUCTS_OLTP T1014,  
       S_ORDER_LINES_OLTP T1008  
where ( T1008.PRODUCT_WID =  
T1014.PRODUCT_WID )  
group by T1014.CATEGORY
```

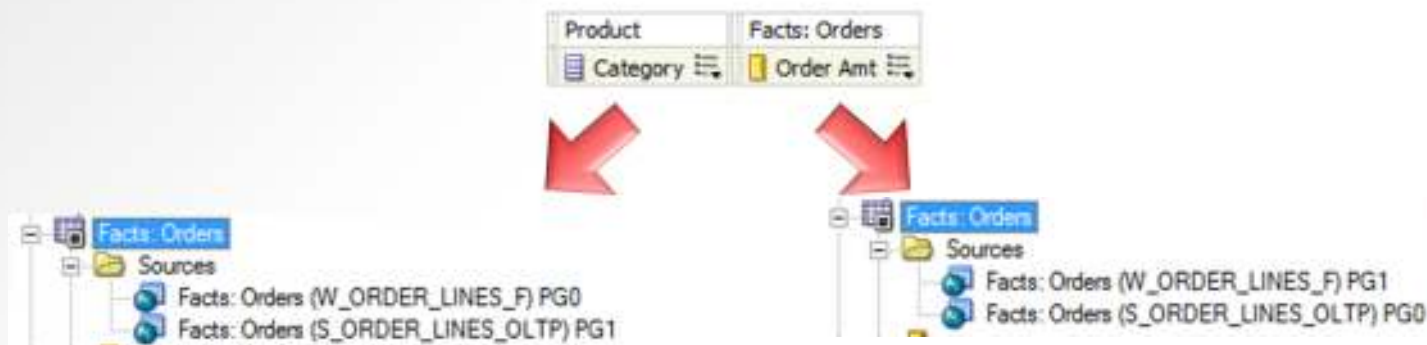
```
select sum(T315.UNIT_PRICE * 1.0 *  
T315.QUANTITY) as c1,  
       T980.CATEGORY as c2  
from  
       W_PRODUCT_D T980 ,  
       W_ORDER_LINES_F T315  
where ( T315.PRODUCT_WID =  
T980.PRODUCT_WID )  
group by T980.CATEGORY
```

- The priority dimension changes the grain
- Forces LTS selection

# LTS Selection

## 11g Priority Groups do a similar thing

- Priority Group of 0 is the highest priority
- Considered very early on in the LTS selection process



```
select sum(T315.UNIT_PRICE * 1.0 * T315.QUANTITY) as c1,  
       T980.CATEGORY as c2  
from  
       W_PRODUCT_D T980 ,  
       W_ORDER_LINES_F T315  
where ( T315.PRODUCT_WID = T980.PRODUCT_WID )  
group by T980.CATEGORY
```

```
select sum(T1008.UNIT_PRICE * 1.0 * T1008.QUANTITY) as c1,  
       T1014.CATEGORY as c2  
from  
       S_PRODUCTS_OLTP T1014,  
       S_ORDER_LINES_OLTP T1008  
where ( T1008.PRODUCT_WID = T1014.PRODUCT_WID )  
group by T1014.CATEGORY
```

# LTS Selection

---

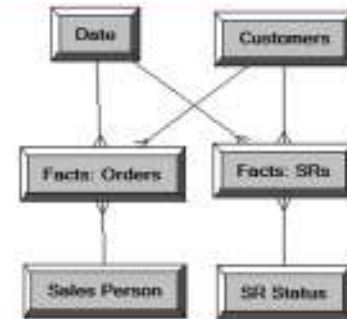
## Other tools to help in LTS Selection:

- Data Security Filters
  - Treated like any other filter from Answers
- Additional columns mapped to specific LTSs
  - Similar solution for choosing Inner Join vs. Outer Join
- Adjustment of # *Elements*
- Different Subject areas w/ Implicit Facts

# Join Path Control

Example:

- Given this Business Model
- And the same prompt on 2 pages
- Customer is constrained on Month



OBI EE  
Sales Order Analysis Service Request Analysis

Year / Month

Customer

Apply Reset

OBI EE  
Sales Order Analysis Service Request Analysis

Year / Month

Customer

Apply Reset

- What if we had a large subject area of Orders & SRs?
  - Can't choose a single implicit fact
- How will OBI determine which customers to show?
- How does the dashboard context factor into its decision?

# Join Path Control

- Answer:
  - It will use the highest LTS
  - It has no UI contextual information
    - Does not know about SRs or Orders dashboard pages
- **Want:** Customers who *ordered* in a month
- **Get:** Customers who had an SR in a month  
→ *Wrong customer list!*
- Pervasive in OBI Dashboards
  - Especially the BI Apps

Sales Order Analysis & SR Analysis Pages:

## Logical SQL:

```
SELECT "Core"."Customers"."Name"  
FROM "Core"  
WHERE "Core"."Date"."Year / Month" = '2011 / 02'  
ORDER BY 1
```



Fact LTS	Dims	Grain
W_SR_F	Customer, Day, Status	2,000,000
W_ORDERS_F	Customer, Day, Person	18,000,000



## Physical SQL:

```
select distinct T321.CUSTOMER_NAME as c1  
from  
  W_DAY_D T861 /* Date (W_DAY_D) */ ,  
  W_CUSTOMER_D T321 /* Customers (W_CUSTOMER_D) */ ,  
  W_SR_F T455 /* SR Combo (W_SR_F) */  
where ( T321.CUSTOMER_WID = T455.CUSTOMER_WID  
and T455.DATE_WID = T861.DATE_WID  
and T861.YEAR_MONTH = '2011 / 02' )  
order by c1
```

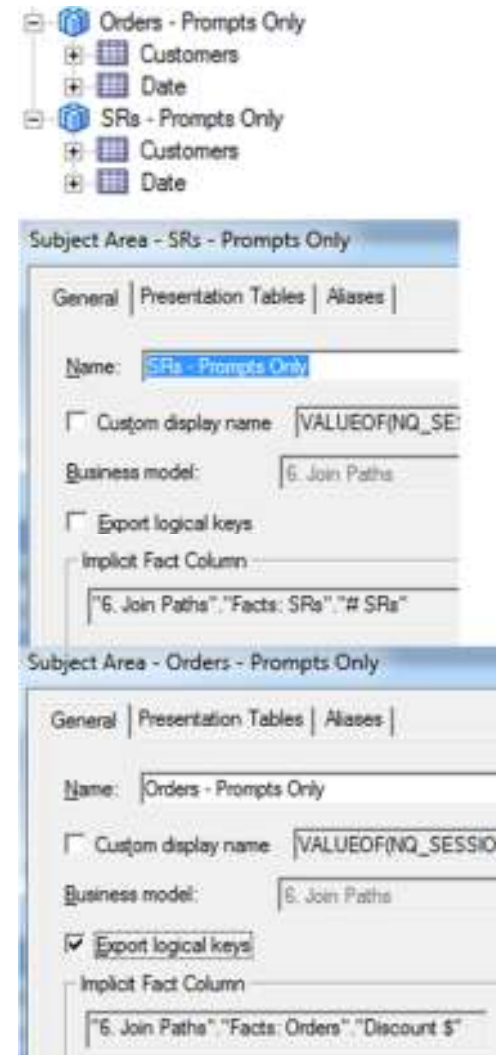
# Join Path Control

## Solution:

- Build contextual prompt Subject Areas
- Tie each area to a specific join path
- Use Implicit Fact Column

## *Enhanced* Solution:

- Improve performance with a Coverage table (i.e. Factless Fact)
  - Summary level
  - E.g., W\_ORDER\_CUST\_MONTH\_F





# Q&A

[www.kpipartners.com](http://www.kpipartners.com)

Please stop by Booth #2229 Moscone South



[www.kpipartners.com](http://www.kpipartners.com)

