



Oracle Business Intelligence 11g Masterclass

Oracle BI Server New Features & Data Modeling Best Practices

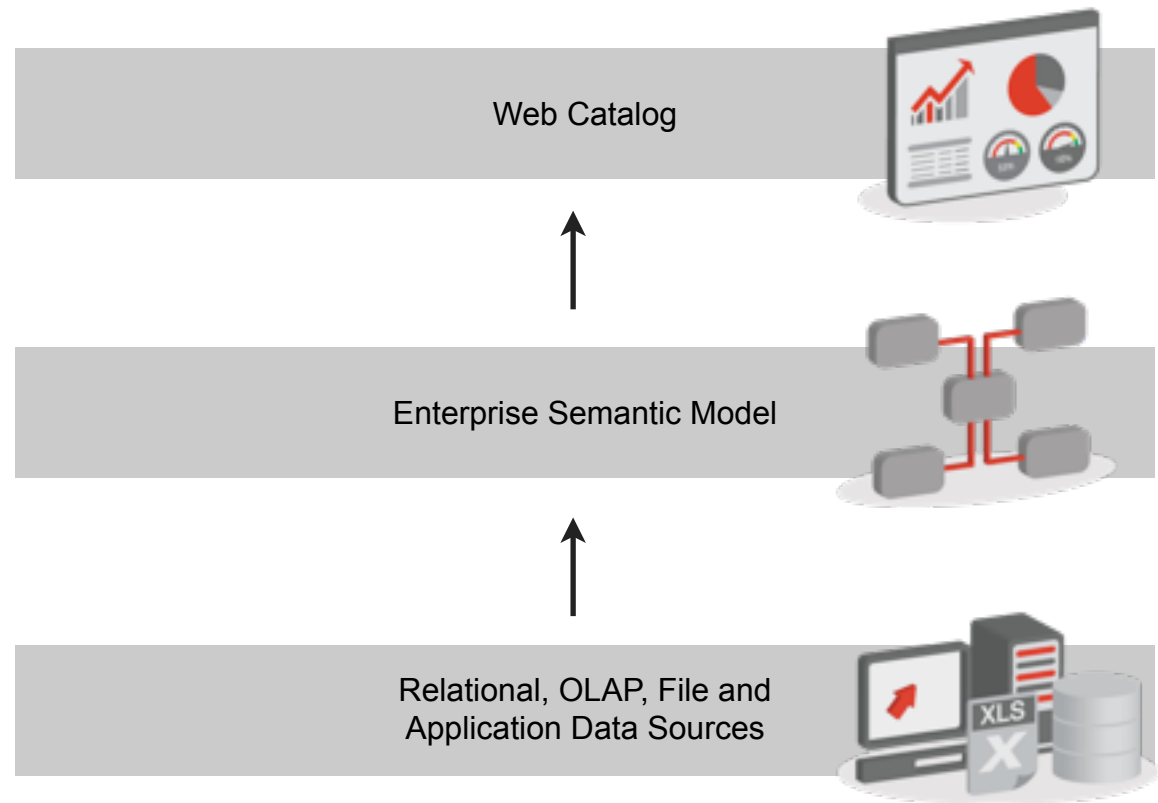
T : +44 (0) 8446 697 995 or (888) 631 1410 (USA) E : enquiries@rittmanmead.com W: www.rittmanmead.com

Agenda

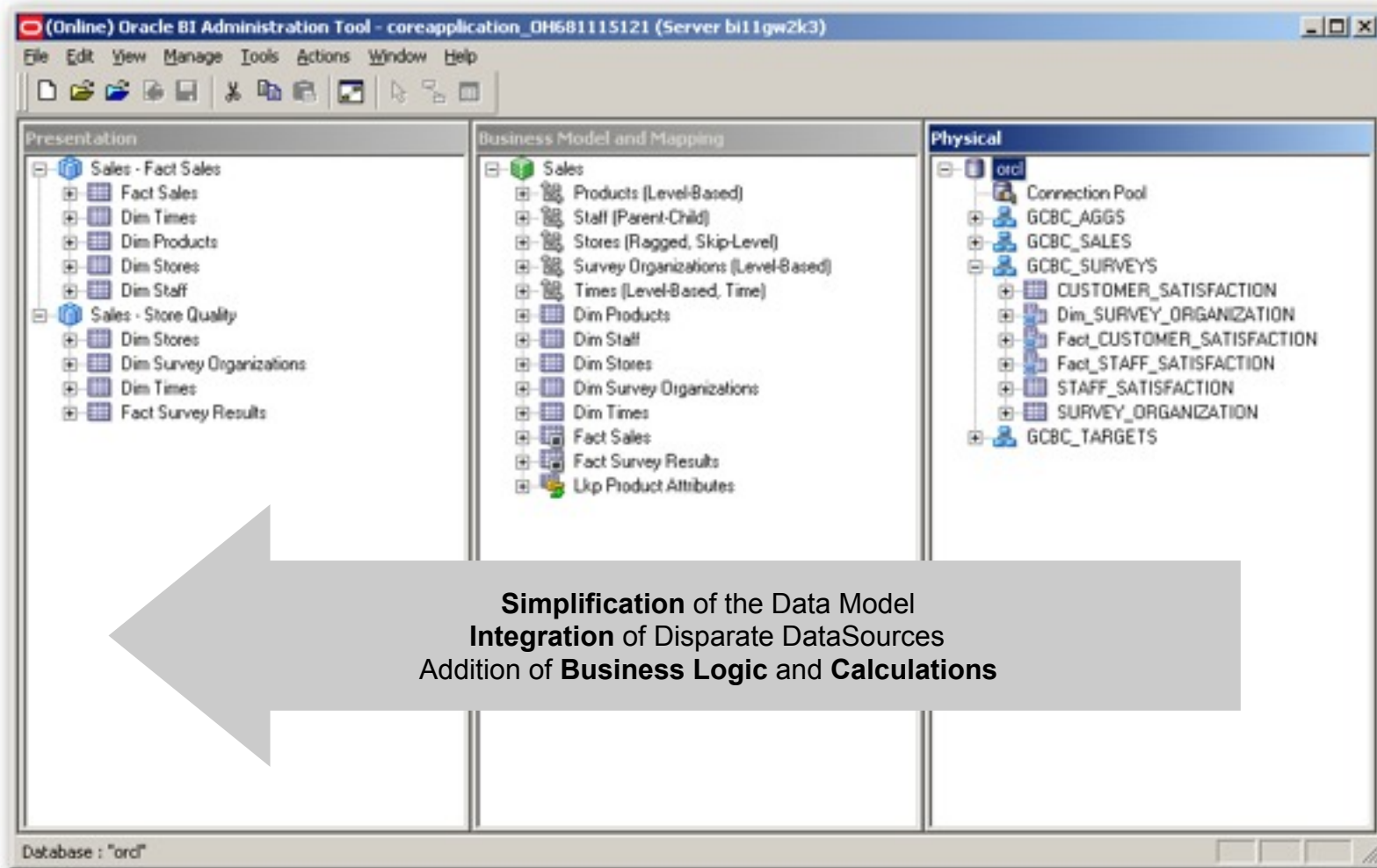
- The OBIEE Common Enterprise Information Model
- Design objectives for the information model
- A Best Practice method for building the information model
- Logical Table Sources and Federated Queries
- New Features in OBIEE 11g data modeling

OBIEE Metadata and Data Layers

- The **Web Catalog** contains definitions of reports, alerts dashboards, KPIs, based on...
- A **Semantic Model** made up of Metrics, dimensions, hierarchies and calculations, derived from...
- **Physical Data Sources**, such as relational databases, OLAP Cubes and application source adapters



Flow of Data Through the Three-Layer Semantic Model



Semantic Model Design Objectives

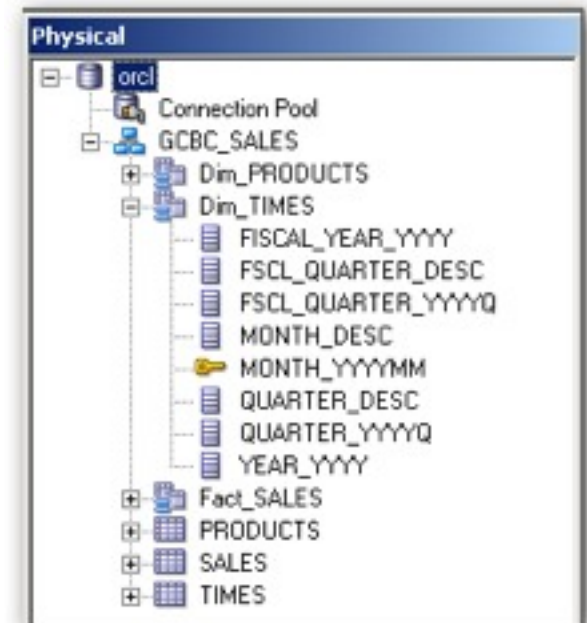
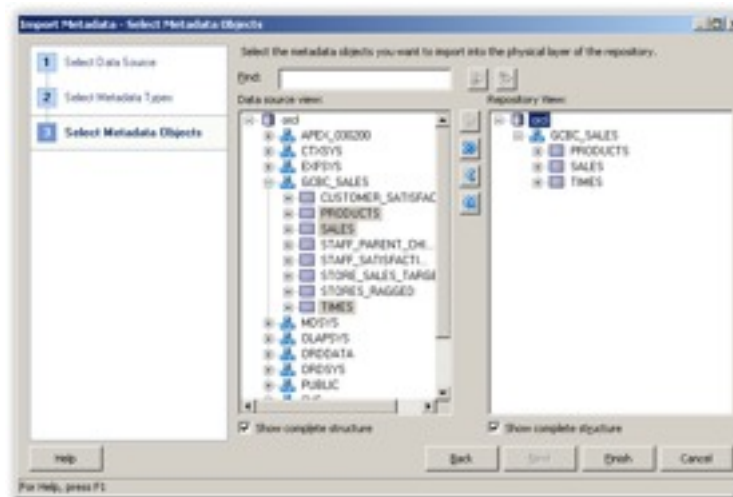
- Create a simple, logical, dimensional model for users to query
- Integrate where possible disparate data sources using conformed dimensions
- Add dimensions and hierarchies to facilitate drilling to detail
- Add common calculations and aggregations
- Present data to users through subject areas (a.k.a. data marts)
- Add role-based data and subject-area security

OBIEE Data Modeling : Basic Good Practices

- Ideally start with a data warehouse data source
- Think in terms of dimensional modeling, most importantly for the business model
- Define PK and FK keys in the physical layer, against aliased copies of tables
- Create outline business model, then map across physical columns, renaming to business terminology and adding business logic (calculations, drill paths)
- Publish one subject area per logical star (fact + dimensions)
- Develop offline initially, reduces contention and check-in/check-out
 - ▶ 11gR1 now supports >5 online developers, but still requires checkin/checkout
- Ensure dimension member counts are (reasonably) accurate
- Use the semantic model to integrate, and simplify, what can be complex and disparate data sources

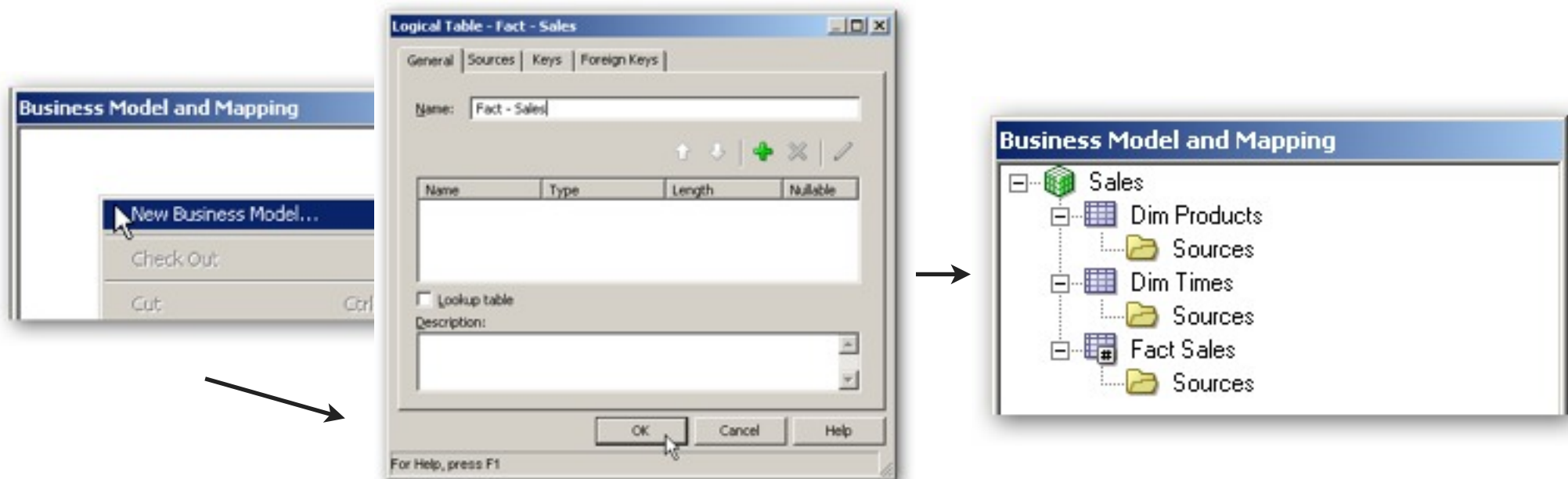
Step 1 : Importing Data Source Metadata

- Assuming source data is a DW, import metadata into enterprise semantic model
- Create aliases that describe the role of each imported table, add additional aliases for multiple roles played by tables (i.e. time, person etc)
- Define PK and FK constraints against alias tables
- Resolve any issues around recursive joins etc



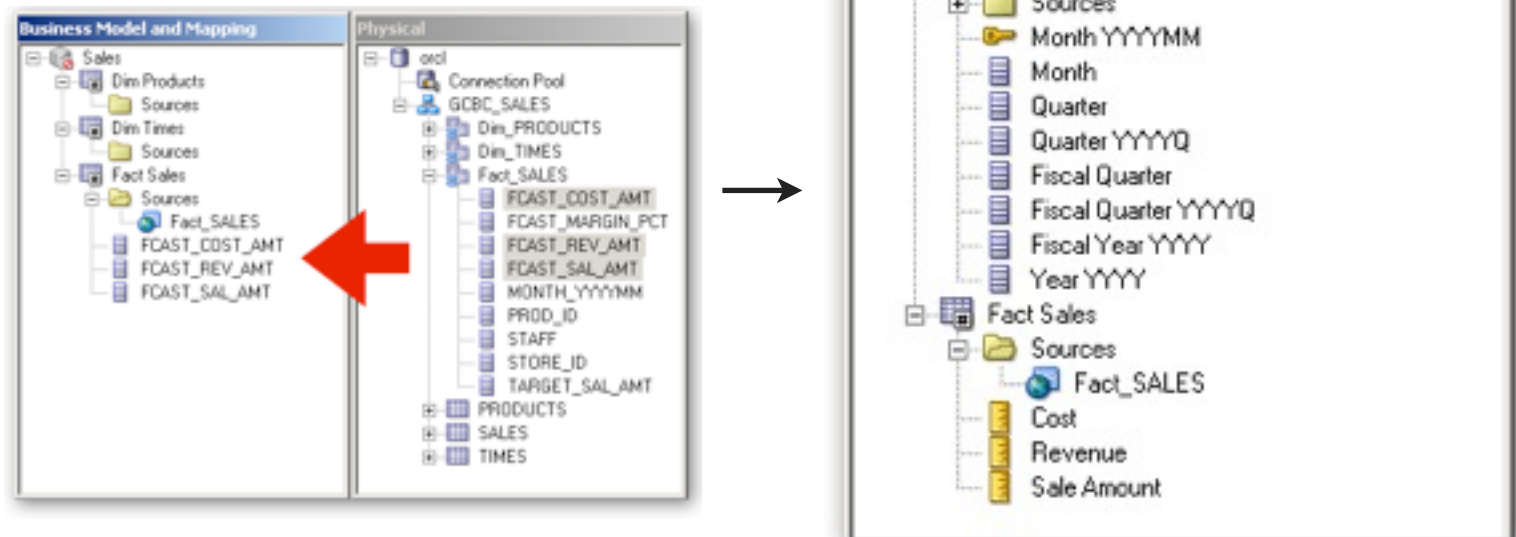
Step 2 : Define Outline Business Model

- Define, at a high level, the business model that will represent your data
- Aim for single model; in short-term, multiple models may be required
 - Migrate to single model as dimensions become conformed
- Define high level logical tables, starting with logical facts



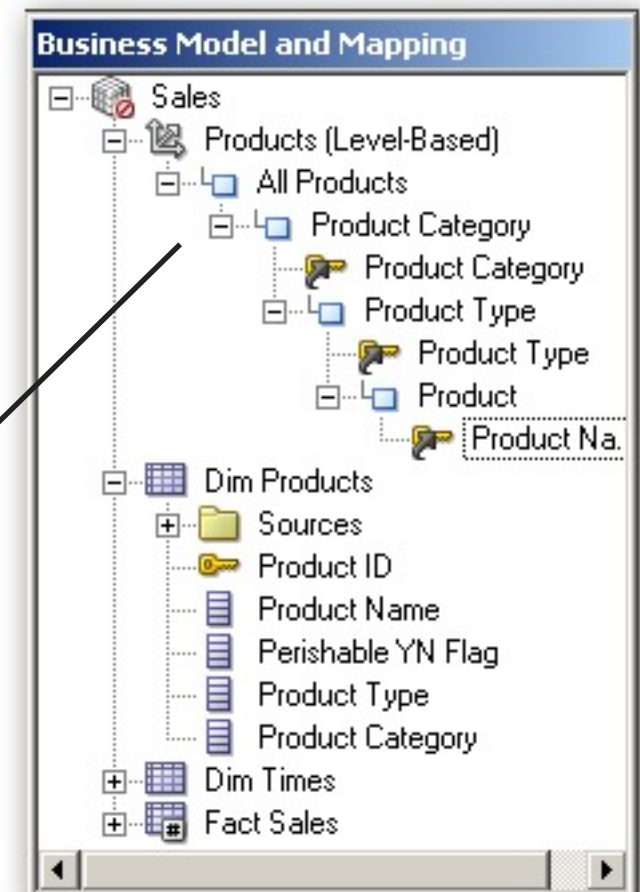
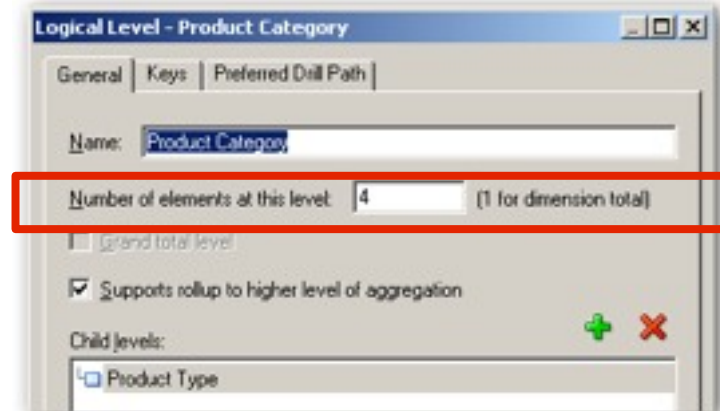
Step 3 : Map Physical Columns into Business Model

- Starting with the logical fact columns, drag and drop physical columns into the outline logical model, and rename columns to reflect business terms
- Do not drag and drop fact table FK columns across
 - ▶ These are referenced through physical model
- Only drag those columns needed for reporting
- Define default aggregation for measures



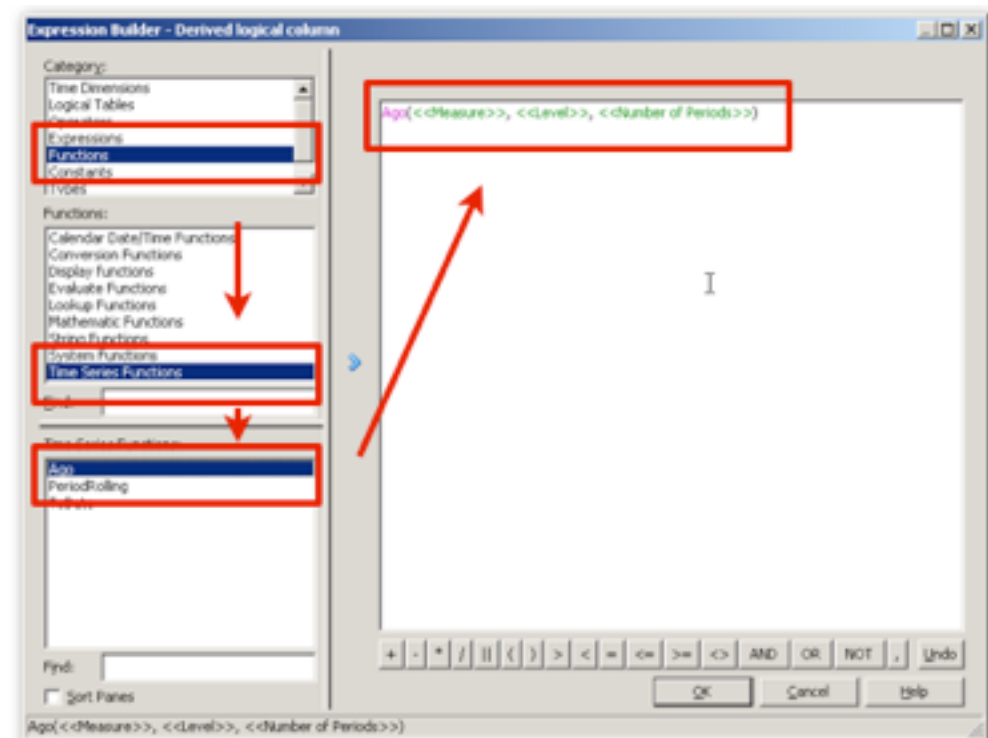
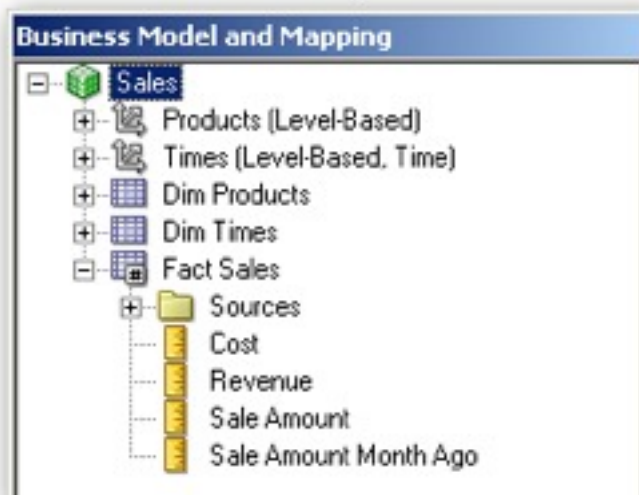
Step 4 : Define Logical Dimensions, Levels and Hierarchies

- Define logical dimensions, hierarchies and levels
- Ensure logical level keys are unique
- Add chronological key(s) for time dimensions
 - ▶ Try to define chron. key at each time dim level
- Add level-based measures if required
- Important: Enter values for “number of elements at this level” (used for agg. navigation)



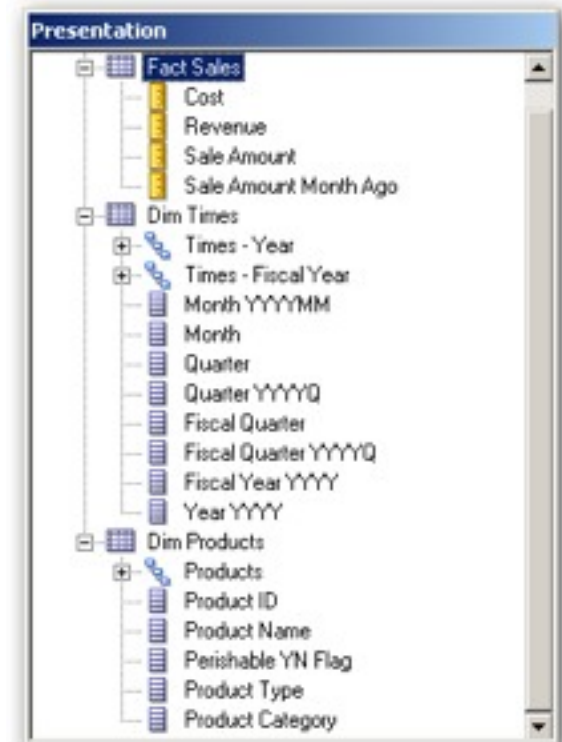
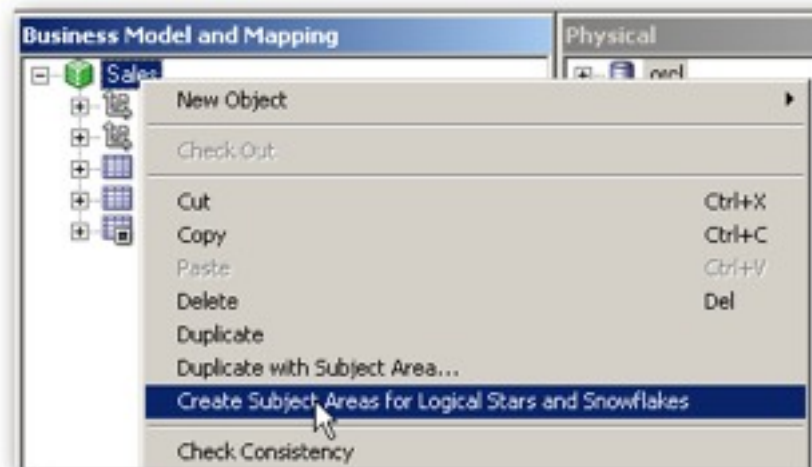
Step 5 : Define Calculations

- Define calculations within the business model
- Can include regular calculations, analytic function, aggregations, time-series functions
- Uses OBIEE syntax; will either be function-shipped to database functions, or performed by BI Server



Step 6 : Create Subject Areas for each Logical Fact

- Make business area available for reporting through publishing as subject area(s)
- Either one subject area per business model, or (better) one per logical fact table
 - ▶ In 11gR1, analyses can be written across subject areas
- Also in 11gR1, logical dimensions now appear in subject areas (more on this later)



Step 7 : Define Security Policy, Providers and Filters

- Use Security Manager to define (10g) or view (11g) users, application groups/ roles
- Apply row-level filters to business model tables and columns
- Apply permissions to subject areas, tables and columns
- From 11g onwards, best practice is to define LDAP sources through WLS (or supporting JEE server)

The image shows two screenshots from the Oracle Security Manager. The left screenshot shows the 'Security Manager' window with a tree view on the left containing 'Identity Management', 'BI Repository', 'Adaptors', 'Directory Servers', and 'Custom Authenticators'. The right pane shows a list of roles: 'BIAdministrator', 'BIAuthor', and 'BIConsumer'. An arrow points from this window to the 'Permissions - Dim Products' dialog box on the right.

The 'Permissions - Dim Products' dialog box has a checked option 'Show all users/application roles'. It contains a table with permissions for various users:

User/Applicat	Read	Read/Write	No Access	Default
Everyone	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	
BIAdministrator	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
BIAuthor	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
BIConsumer	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

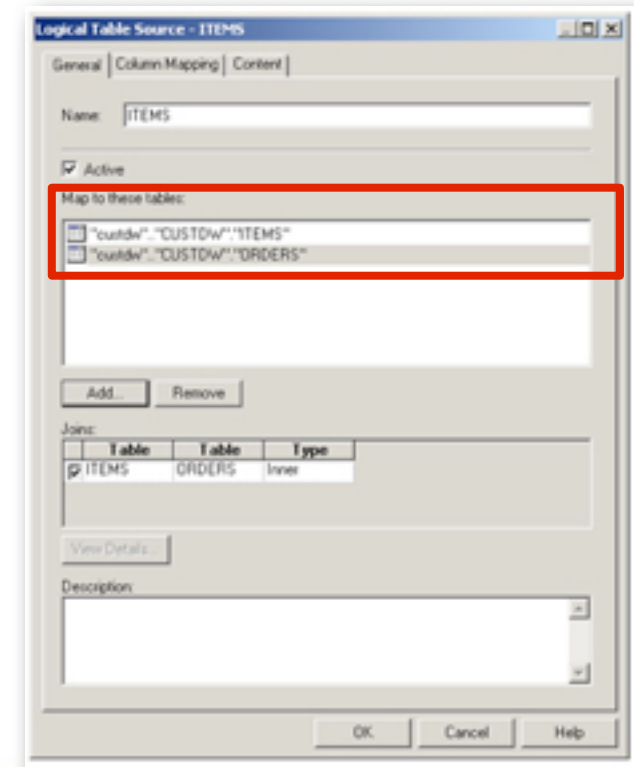
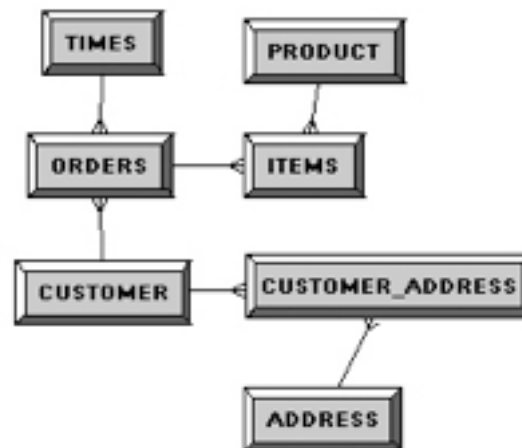
Below this table, there is a section for 'Data Filters'. It includes a 'Subject Area' dropdown set to 'All', a 'Total Filters: 0' indicator, and a table of filters:

Type	Layer	Name	Status	Data Filter
	Presentation	"Sales - Fact Sales", "Dim Products"	Enabled	"Sales"."Dim Products"."Product Category" = 'On

An arrow points from the 'Data Filters' table back to the 'Permissions - Dim Products' dialog box.

Variation : Modeling Against OLTP Sources

- For OLTP data sources, it may be better to completely create the business model first, then map in physical column sources
- Allows you to manipulate logical table sources to denormalize data
- OLTP schema may require modeling, manipulation to map into business dimensional model



What is a Logical Table Source?

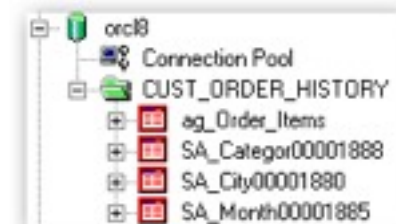
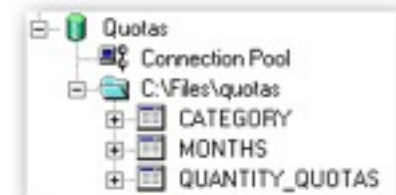
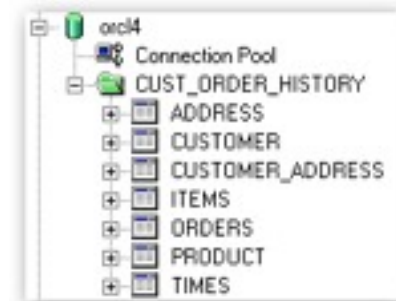
- Facts and dimensions are **mapped** to physical sources **Logical Table Sources**
- LTS are chosen at run time based on RPD definitions (level mapping, fragmentation, federation etc)



Logical Column	Expression	Physical Table
Customer ID	CUSTID	ORDERS
Day ID	ORDERDATE	ORDERS
Item ID	ITEMID	ITEMS
Order ID	ORDID	ORDERS
Price	PRICE	ITEMS
Product ID	PRODID	ITEMS
Quantity	QUANTITY	ITEMS

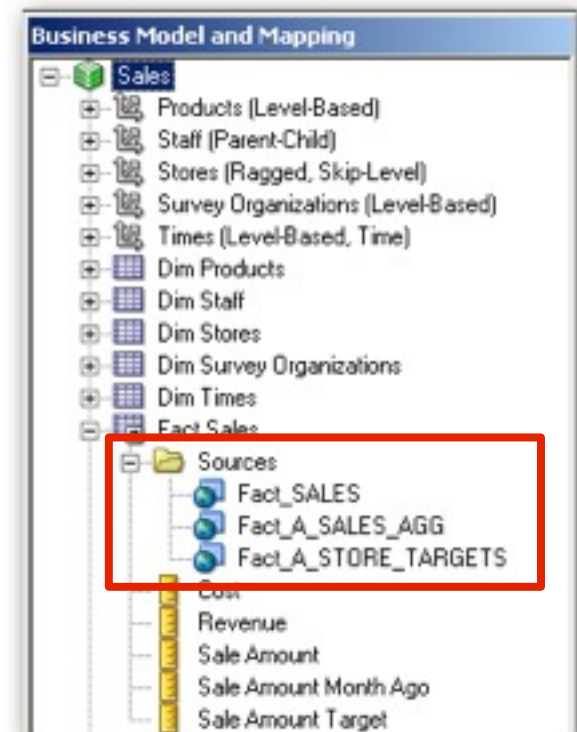
Logical Column	Expression	Physical Table
Quota	QUOTA	QUANTITY_QUOTA

Logical Column	Expression	Physical Table
Quantity	Quantity00001888	ag_Order_Items



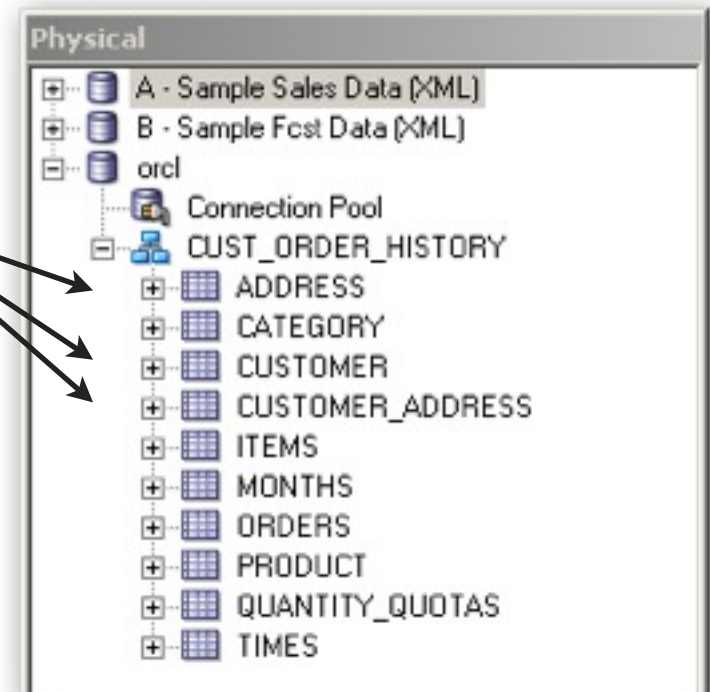
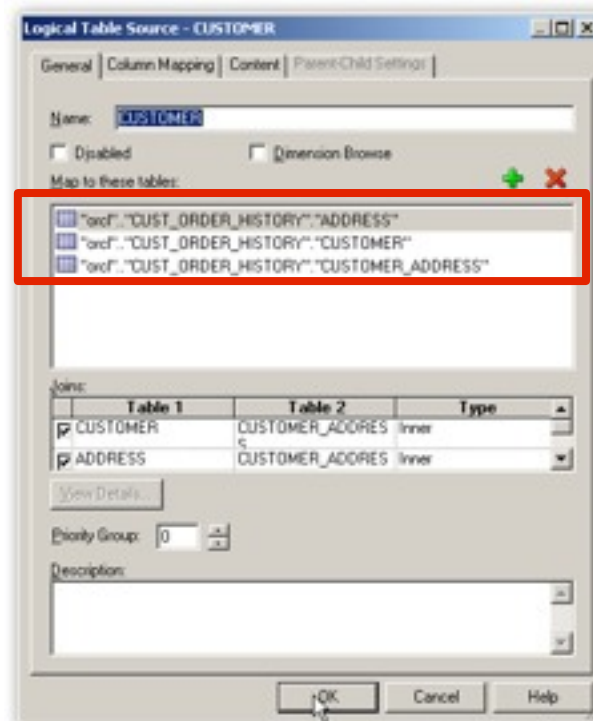
When Do We Create Multiple Logical Table Sources?

- Each LTS represents a mapping of logical columns to physical sources
- Additional LTS mappings are required when one logical SQL statement couldn't span all required data sources
 - ▶ Example : aggregate and detail-level data sources (known as **Horizontal Federation**)
 - ▶ Example : two sources on different physical databases (known as Vertical Federation)
- 1-n LTS mapping may be used for an analyses, depending on context of query
 - ▶ Does analysis require data spanning multiple physical databases?
 - ▶ Does analysis require data of differing levels of granularity?



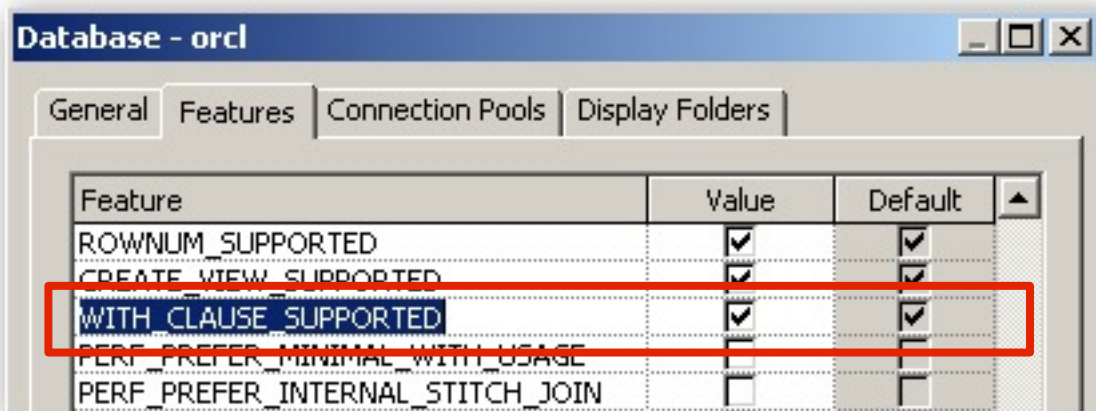
When Do We Extend an Existing Logical Table Source?

- Typically used when denormalizing fact or dimension sources
- Allows a single LTS mapping to extend to additional joined physical tables
- Preferable to adding new LTS mappings, when a single logical SQL could cover both tables



How Is SQL Generated Affected By Multiple LTSs?

- Strategy for the BI Server is to try and push-down joins where possible
- In most cases, multiple LTS will resolve to >1 physical SQL query (or MDX, etc)
- However many queries (cross-fact etc) involve >1 fact LTS, and these can infact be resolved as a single physical SQL
 - ▶ **WITH_CLAUSE_SUPPORTED = YES** in Database Features setting
- Guideline : prepare for >1 physical SQL, but BI Server will try and push joins down to database if possible



```
WITH
SAWITH0 AS (select sum(T2830.UNIT_COST) as c1,
T2874.PROD_SUBCATEGORY_DESC as c2
from
PRODUCTS T2874,
COSTS T2830
where ( T2830.PROD_ID = T2874.PROD_ID )
group by T2874.PROD_SUBCATEGORY_DESC),
SAWITH1 AS (select sum(T2911.AMOUNT_SOLD) as c1,
T2874.PROD_SUBCATEGORY_DESC as c2
from
PRODUCTS T2874,
SALES T2911
where ( T2874.PROD_ID = T2911.PROD_ID )
group by T2874.PROD_SUBCATEGORY_DESC)
select distinct SAWITH0.c1 as c1,
SAWITH1.c1 as c2,
case when SAWITH0.c2 is not null then SAWITH0.c2
when SAWITH1.c2 is not null then SAWITH1.c2 end as c3
from
SAWITH0 full outer join SAWITH1 On SAWITH0.c2 = SAWITH1.c2
order by c3
```



Demonstration

A Basic OBIEE 11g BI Server Repository

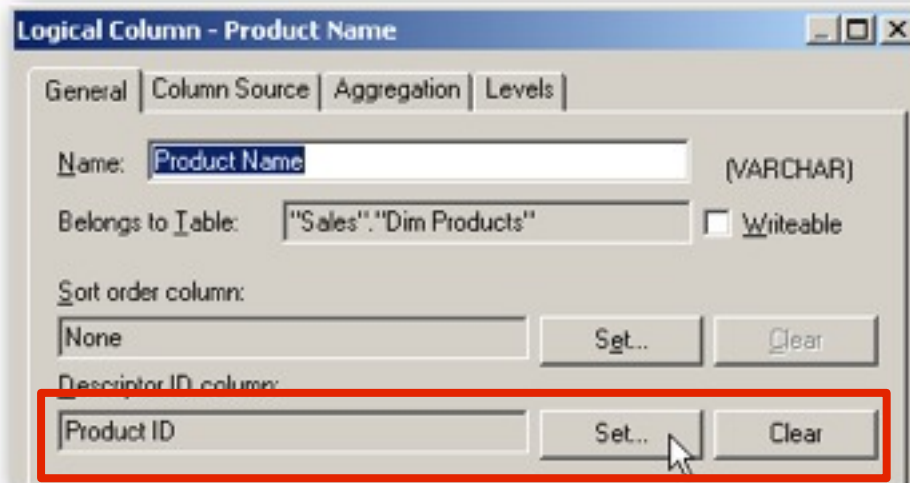
T : +44 (0) 8446 697 995 or (888) 631 1410 (USA) E : enquiries@rittmanmead.com W: www.rittmanmead.com

New Features in OBIEE 11gR1 For Data Modeling

- ID Columns for Descriptive Columns (“Double-columns”)
- Lookup Tables
- Support for Ragged, Skip-Level and Parent-Child Hierarchies
- Hierarchical Subject Area Columns
- LTS Priority Ordering
- Repository Passwords
- Deployment of Repository Files using Fusion Middleware Control

ID Columns for Descriptive Columns (Double-Columns)

- In OBIEE 10g, it was difficult to define dashboard prompts that display descriptions, but pass back IDs
- OBIEE 11gR1 allows you to define an ID column for descriptive column(s)
 - ▶ Multiple columns can reference same ID
- Dashboard prompts are aware of IDs



Logical Column - Product Name

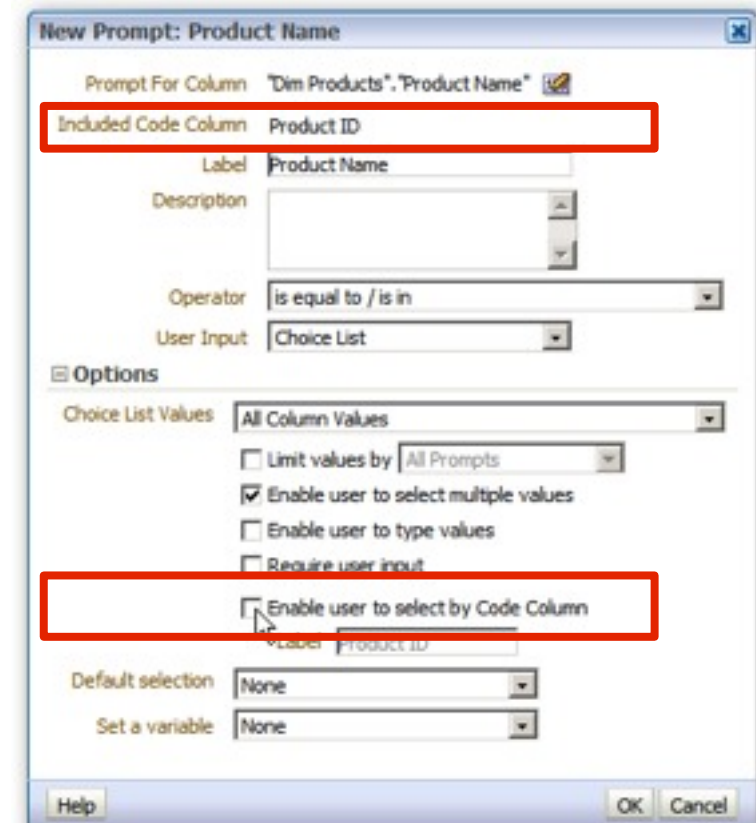
General | Column Source | Aggregation | Levels

Name: (VARCHAR)

Belongs to table: ☐ Writeable

Sort order column:

Descriptor ID column:

New Prompt: Product Name

Prompt For Column:

Included Code Column:

Label:

Description:

Operator:

User Input:

Options

Choice List Values:

☐ Limit values by

☒ Enable user to select multiple values

☐ Enable user to type values

☐ Require user input

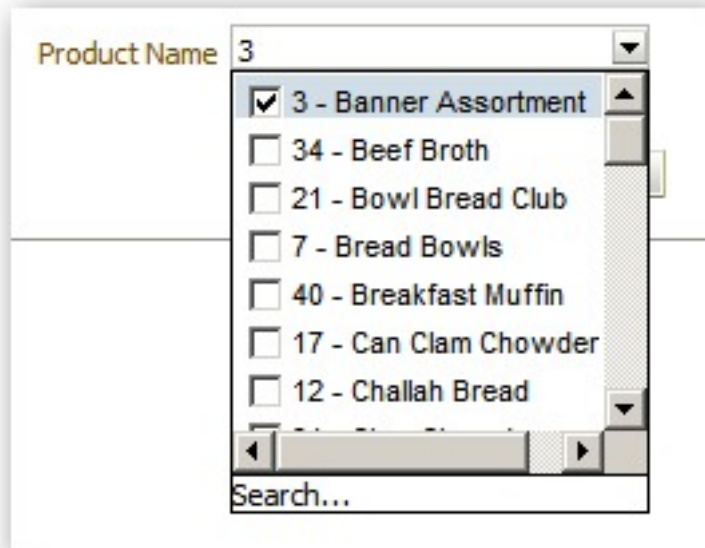
☐ Enable user to select by Code Column

Default selection:

Set a variable:

Double-Columns in Use

- User can select using just descriptions, or can elect to show IDs as well
- Set analyses prompt to “Is Prompted”
- SQL will automatically filter on IDs, rather than descriptions



```
WITH
SAWITH0 AS (select sum(T117.REV_AMT) as c1,
                T107.PROD_CAT_DESC as c2,
                T107.PROD_ID as c3
from
    GCBC_SALES.PRODUCTS T107,
    GCBC_SALES.SALES T117
where (T107.PROD_ID = T117.PROD_ID
and (T107.PROD_ID in (12, 17, 31)))
group by T107.PROD_CAT_DESC, T107.PROD_ID),
SAWITH1 AS (select 0 as c1,
                D1.c2 as c2,
                D1.c3 as c3,
                D1.c1 as c4,
                sum(D1.c1) as c5
from
    SAWITH0 D1
group by D1.c1, D1.c2, D1.c3)
select D1.c1 as c1,
        D1.c2 as c2,
        D1.c3 as c3,
        D1.c4 as c4,
        sum(D1.c5) over (partition by D1.c2)
as c5
from
    SAWITH1 D1
```



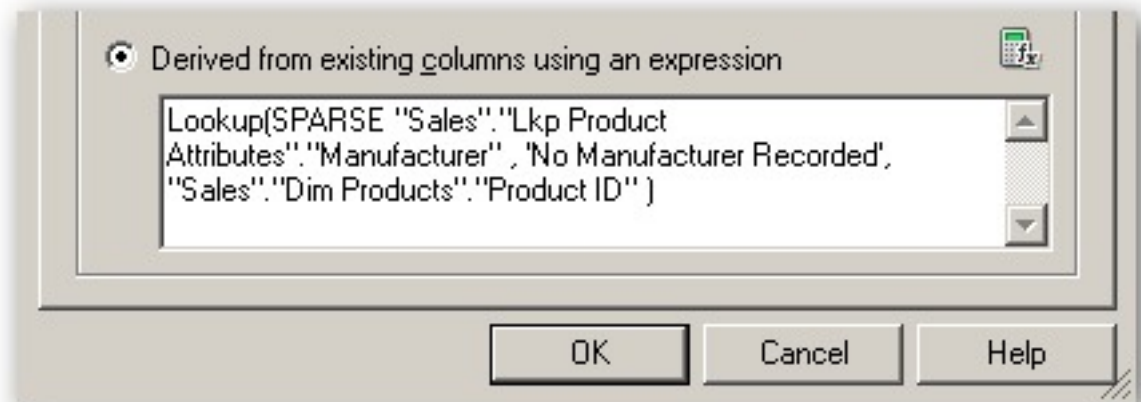

Demonstration

Double-Columns

T : +44 (0) 8446 697 995 or (888) 631 1410 (USA) E : enquiries@rittmanmead.com W: www.rittmanmead.com

Lookup Tables and Function

- LOOKUP function introduced with OBIEE 11g allows lookups on logical and physical tables
- Useful for several scenarios
 - ▶ Pushing currency conversions to separate calculations
 - ▶ Accessing columns that cannot use GROUP BY (e.g. CLOB)
 - ▶ Avoiding unnecessary outer joins in LTS

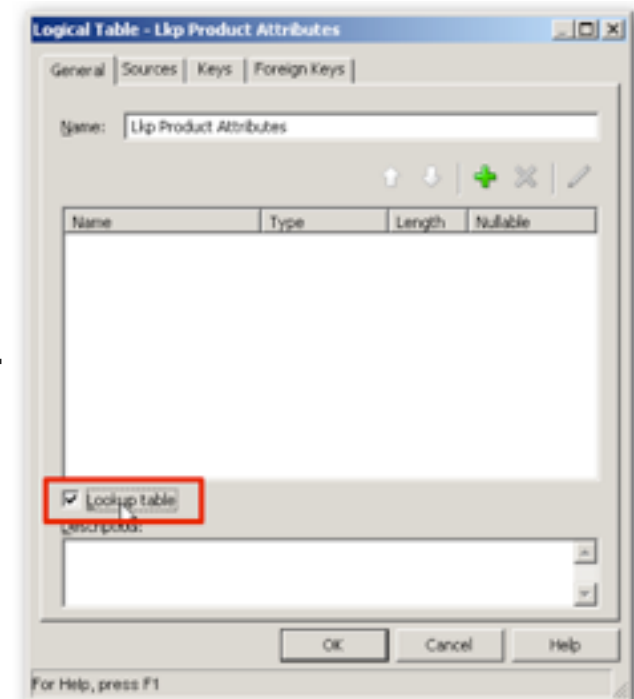
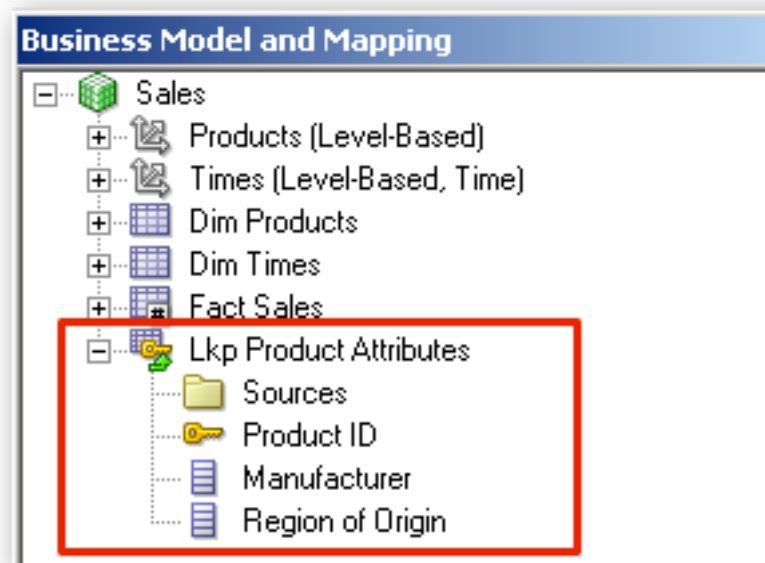


Lookup Table Use Cases

	Physical	Logical
Dense Lookup	Financial Restatements where we get the rates for all the countries. In this case, the multiplication of rates will happen after Group By	Same as physical, but supports lookups across data sources (i.e. Oracle and Essbase)
Sparse Lookup	Reporting on datatypes (i.e. CLOBs) that cannot have GROUP BY, DISTINCT applied to them. Situations where we want to reference a table by an outer join	Same as physical, but supports lookups across data sources (i.e. Oracle and Essbase)

Lookup Logical Tables

- Allows you to define a logical table as being a lookup table
 - ▶ Removes check as to whether it is a fact, or joined to a fact table
- Can be accessed via a logical calculation using SPARSE or DENSE lookup
- BI Server generates separate lookup SQL query, joined to main dataset by BI Server in-memory



Accessing Lookup Tables

- Lookup tables are accessed using the new LOOKUP function
- Can be SPARSE (outer join) or DENSE (inner join)
- Physical table lookups push join to SQL query; logical creates two SQL queries

`Lookup(DENSE <<lookupColumn>>, <<sourceKeyorExpression>>)`

`Lookup(SPARSE <<lookupColumn>>, <<alternateColumn>>, <<sourceKeyorExpression>>)`

Logical Table Source - Dim_TIMES

General Column Mapping Content Parent Child Settings

☒ Show gapped columns ☐ Show unmapped columns

Logical column to physical column mapping:

Logical Column	Expression	Phys
Fiscal Quarter	FISCAL_QUARTER_DESC	Dim_TIM
Fiscal Quarter YYYYQ	FISCAL_QUARTER_YYYYQ	Dim_TIM
Fiscal Year YYYY	FISCAL_YEAR_YYYY	Dim_TIM
Month	MONTH_DESC	Dim_TIM
Month YYYYMM	MONTH_YYYYMM	Dim_TIM
Quarter	QUARTER_DESC	Dim_TIM
Quarter YYYYQ	QUARTER_YYYYQ	Dim_TIM
Season	Lookup(DENSE "ord"."GCBC_SALES"."Lkp_SEASON	Dim_TIM
Year YYYY	YEAR_YYYY	Dim_TIM

☒ Derived from existing columns using an expression

Lookup(SPARSE "Sales"."Lkp Product Attributes"."Manufacturer", 'No Manufacturer Recorded', "Sales"."Dim Products"."Product ID")

OK Cancel Help



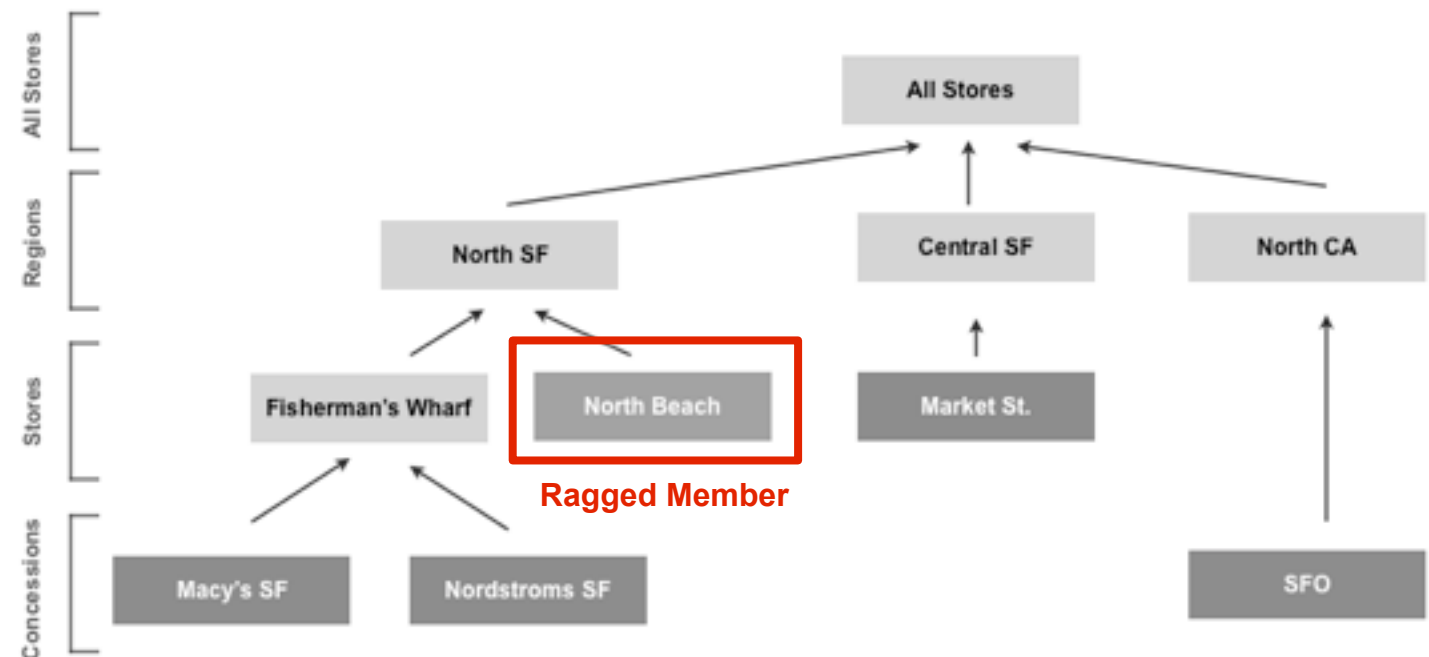
Demonstration

Lookup Tables

T : +44 (0) 8446 697 995 or (888) 631 1410 (USA) E : enquiries@rittmanmead.com W: www.rittmanmead.com

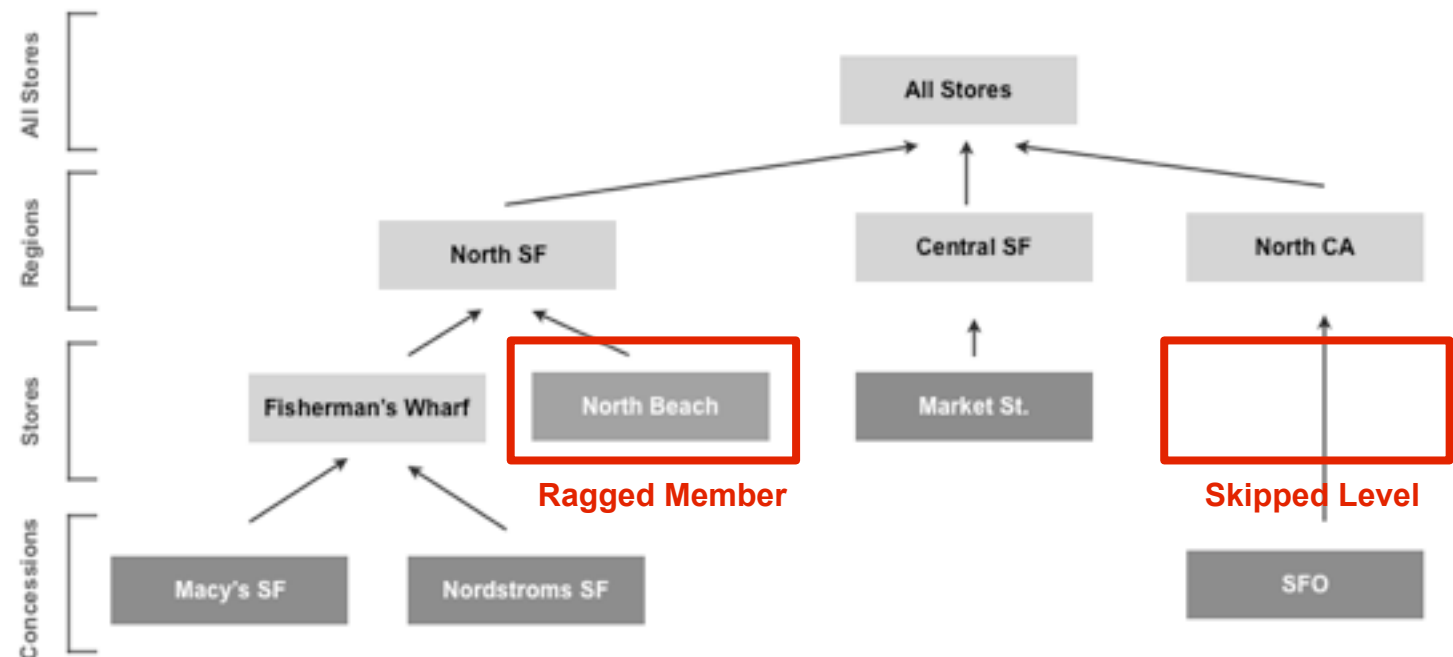
Ragged and Skip-Level Logical Dimensions

- Level-based hierarchies may have skip-levels, or ragged hierarchies
- Skip-levels are when members may not have an immediate parent
- Ragged hierarchies are when leaf members are not all at the same level



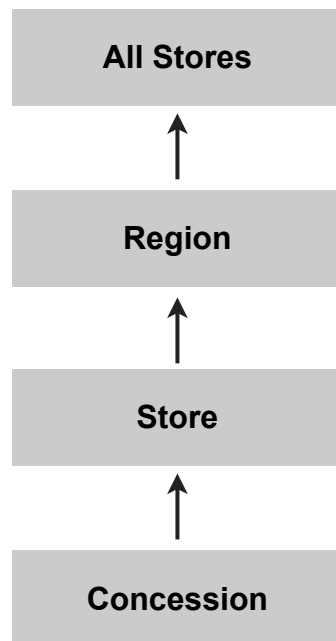
Ragged and Skip-Level Logical Dimensions

- Level-based hierarchies may have skip-levels, or ragged hierarchies
- Skip-levels are when members may not have an immediate parent
- Ragged hierarchies are when leaf members are not all at the same level



Ragged and Skip-Level Logical Dimensions in the Dataset

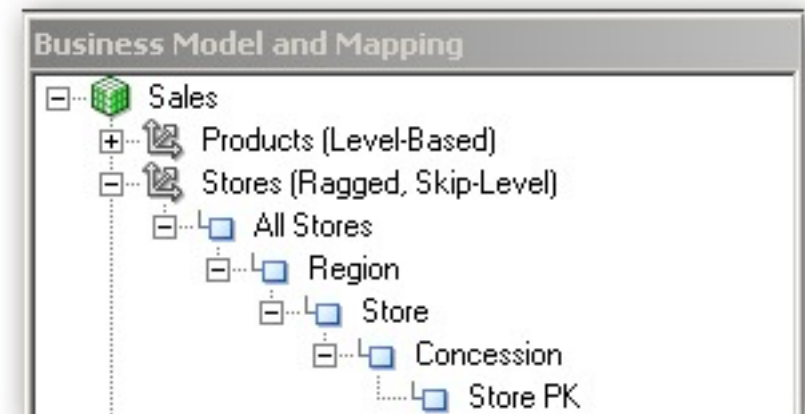
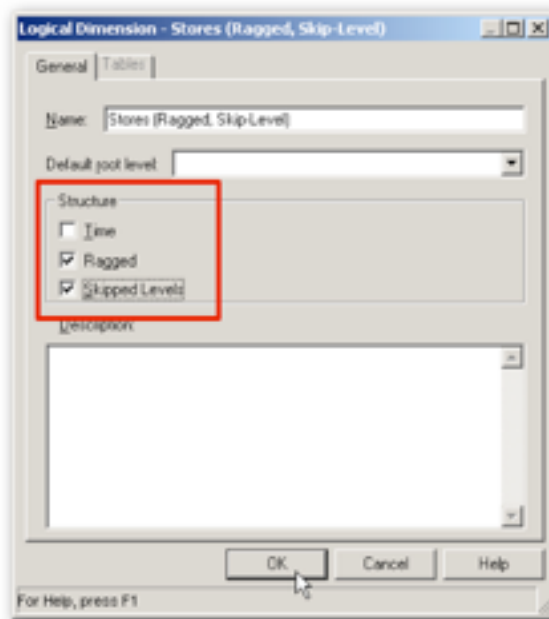
- Skip-levels and Ragged hierarchies are represented in the underlying table through NULLs



	CONCESSION_ID	CONCESSION_NAME	STORE_ID	STORE_NAME	REGION_ID	REGION_NAME
1	(null)	(null)	3	Geary Street	1	Central SF
2	(null)	(null)	1	Market Street	1	Central SF
3	(null)	(null)	4	North Beach	2	North SF
4	(null)	(null)	5	Taylor Street	2	North SF
5	(null)	(null)	2	Fishermans Wharf	3	North SF
6	102	SFO	(null)	(null)	3	North CA
7	(null)	(null)	6	Walnut Creek	3	North CA
8	(null)	(null)	11	San Diego	4	South CA
9	(null)	(null)	9	Santa Clara	4	South CA
10	(null)	(null)	7	Anaheim	4	South CA
11	(null)	(null)	8	Los Angeles	4	South CA
12	(null)	(null)	14	Embarcadero	5	East SF
13	(null)	(null)	15	10th Avenue	6	West SF
14	(null)	(null)	16	Haight Street	6	West SF
15	(null)	(null)	12	Chicago	7	Other USA
16	(null)	(null)	13	New York	7	Other USA
17	100	Mordstroms, SF	2	Fishermans Wharf	2	North SF
18	101	Macys, SF	2	Fishermans Wharf	2	North SF

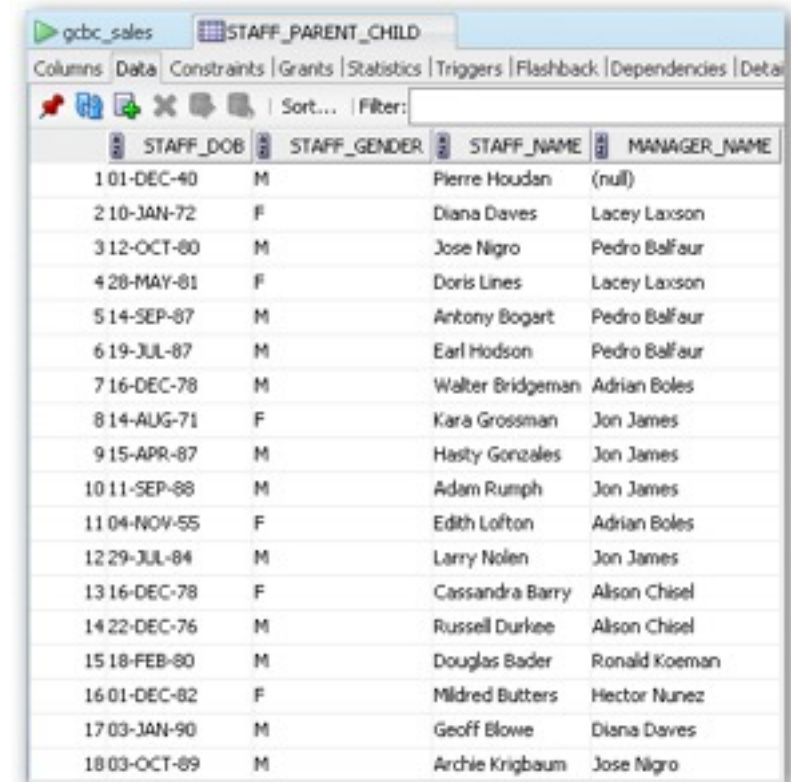
Creating Ragged and Skip-Level Logical Dimensions

- Select Ragged and/or Skip-Level in the logical dimension properties
 - ▶ Do not select for non-ragged, non-skip, as detecting NULLs adds overhead
- For ragged hierarchies, add surrogate key to ensure consistent logical PK
 - ▶ Delete this additional level when you create the presentation hierarchy



Parent-Child (Value-Based) Dimensions

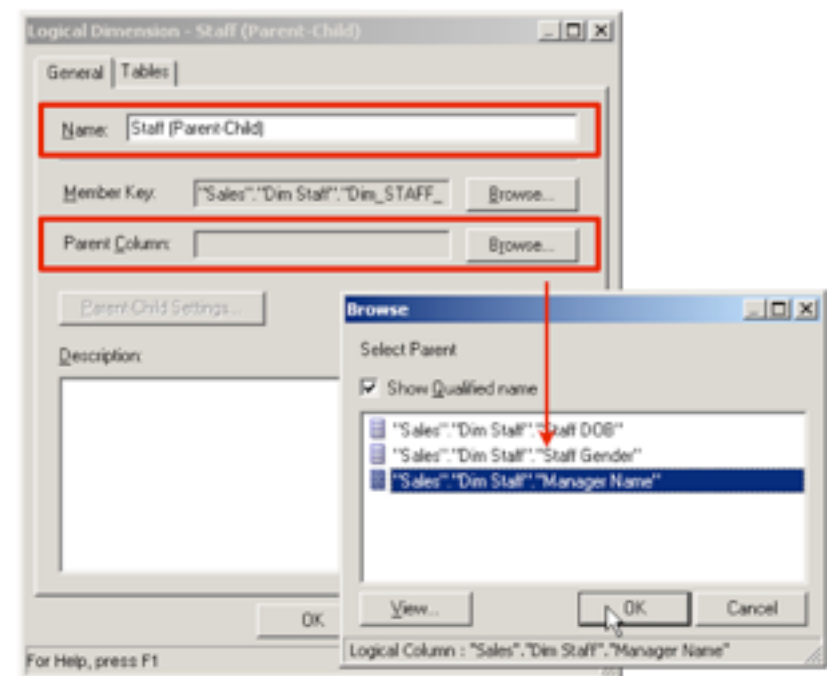
- Some data sets contain recursive, parent-child (value-based) hierarchies
 - ▶ Hierarchy is defined in the data, rather than the column relationships
- Typically used for organizations and other implicitly ragged hierarchies
- Difficult to model relationally as recursive SQL queries would be required to traverse



STAFF_ID	STAFF_DOB	STAFF_GENDER	STAFF_NAME	MANAGER_NAME
1	01-DEC-40	M	Pierre Houdan	(null)
2	10-JAN-72	F	Diana Daves	Lacey Laxson
3	12-OCT-80	M	Jose Nigro	Pedro Balfaur
4	28-MAY-81	F	Doris Lines	Lacey Laxson
5	14-SEP-87	M	Antony Bogart	Pedro Balfaur
6	19-JUL-87	M	Earl Hodson	Pedro Balfaur
7	16-DEC-78	M	Walter Bridgeman	Adrian Boles
8	14-AUG-71	F	Kara Grossman	Jon James
9	15-APR-87	M	Hasty Gonzales	Jon James
10	11-SEP-88	M	Adam Rumph	Jon James
11	04-NOV-55	F	Edith Lofton	Adrian Boles
12	29-JUL-84	M	Larry Nolen	Jon James
13	16-DEC-78	F	Cassandra Barry	Alson Chisel
14	22-DEC-76	M	Russell Durkee	Alson Chisel
15	18-FEB-80	M	Douglas Bader	Ronald Koeman
16	01-DEC-82	F	Mildred Butters	Hector Nunez
17	03-JAN-90	M	Geoff Blowe	Diana Daves
18	03-OCT-89	M	Archie Krigbaum	Jose Nigro

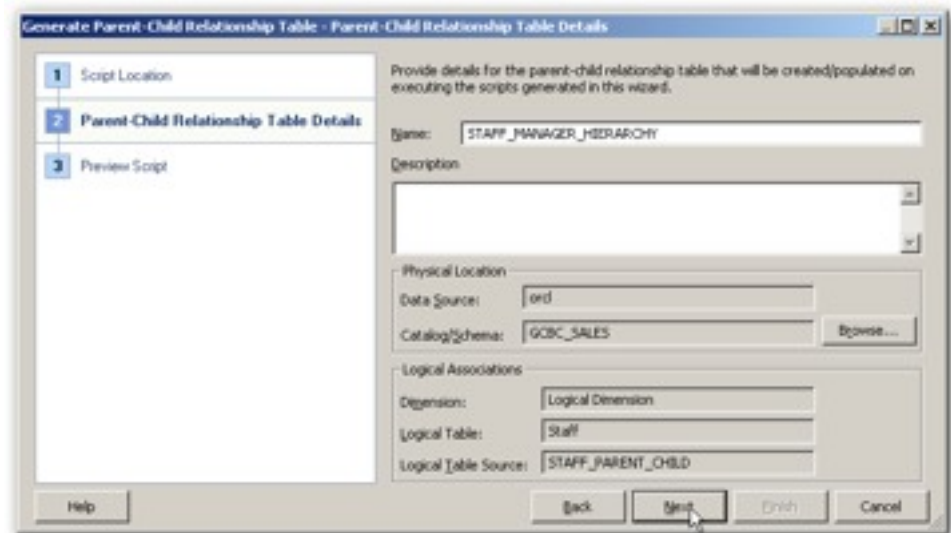
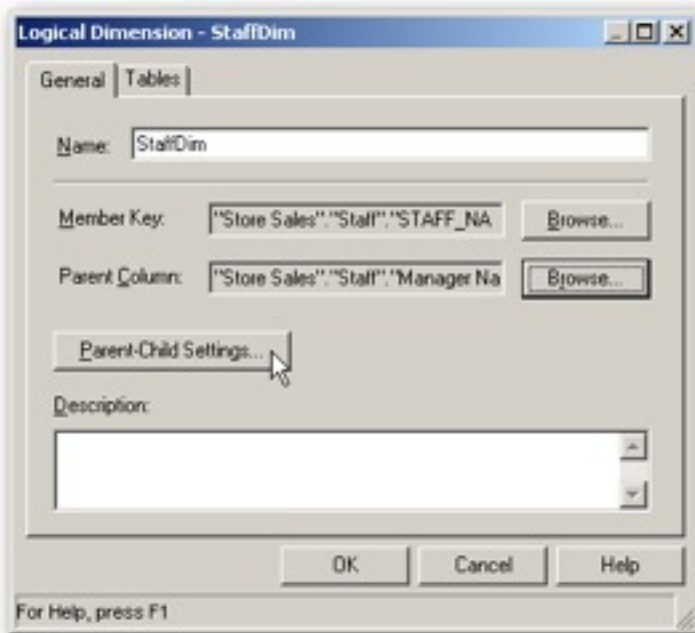
Creating Parent-Child Hierarchies

- Command to create logical dimensions now includes Parent-Child option
- Initial logical dimension dialog then prompts you to select **Parent Column**
- Once selected, you then need to define the **closure table** to resolve recursion



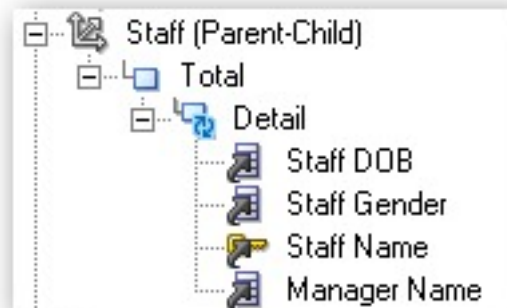
Parent-Child Hierarchy Closure Tables

- To avoid the need for recursive SQL, a closure table is defined for the hierarchy
- Iterates through the hierarchy once, speeds up all subsequently queries
- Creates two scripts; one to create closure table, one to populate it
- Population script will need to be re-run each time underlying data changes

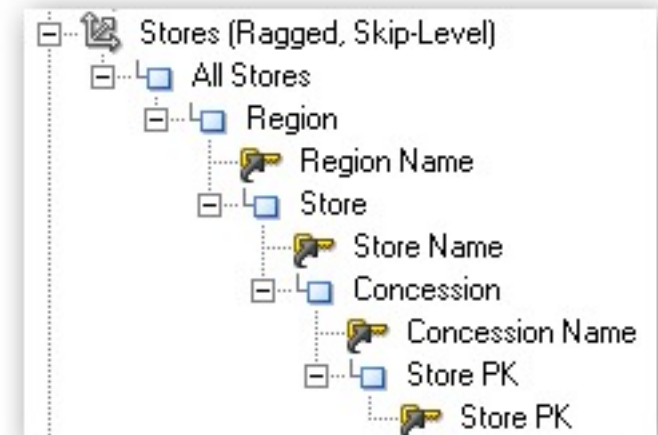


Parent-Child vs. Ragged Hierarchies : Which to Use?

- Parent-child hierarchies are naturally ragged, so which should you choose?
- Depends on the data structures you are reporting on
 - ▶ If each hierarchy level is distinct and named, go for level-based with ragged hierarchy option
 - ▶ If hierarchy levels are not named, or data is stored in RDBMS recursively, go for parent-child hierarchy
- Parent-child takes more maintenance, so only use if necessary

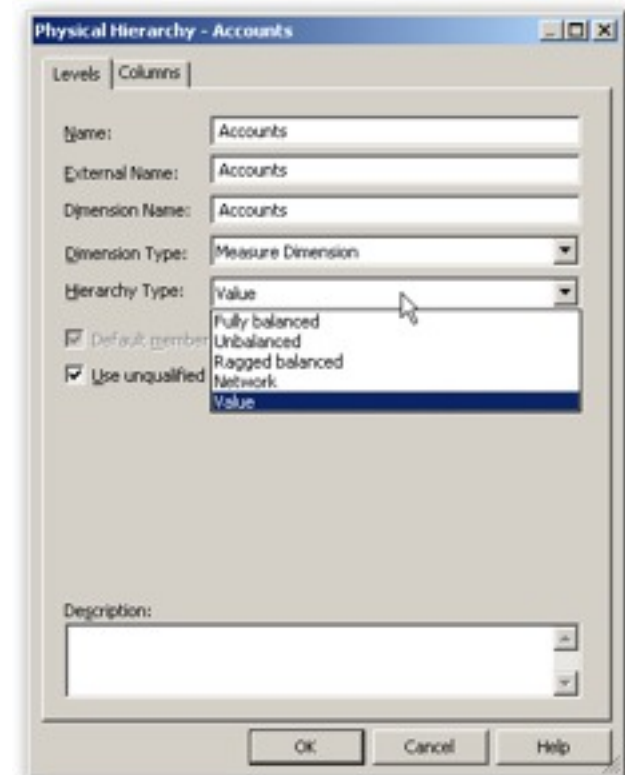
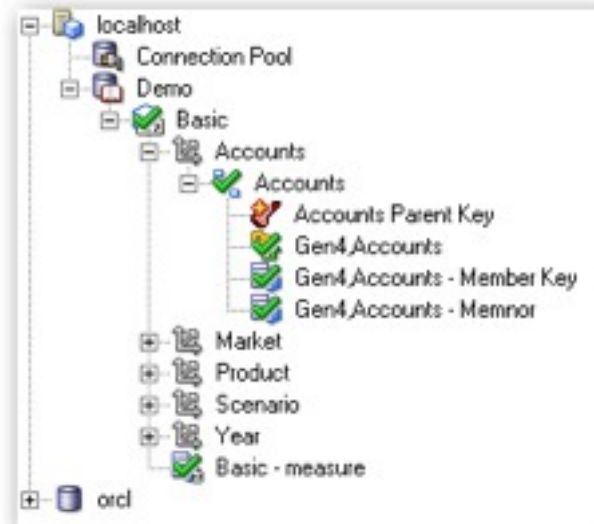


VS.



Hierarchical Columns and Essbase Sources

- Essbase data sources can be amended to use parent-child (“value”) hierarchies
- Avoids issue where changing outlines requires re-import
- Business Model columns no longer based on outline generations
- New generations can be added dynamically as Essbase outline evolves



Hierarchical Columns

- Logical dimensions in the business model can now be dragged across to create Hierarchical Columns in subject areas
- One hierarchical column per logical dimension hierarchy
- Allows in-column drilling in analyses

Presentation

- Store Sales - Sales
 - Sales
 - Stores (Ragged & Skip Level)
 - Stores (Ragged & Skip Level)
 - Concession Name
 - Region Name
 - Store Name
 - Store PK
 - Staff (Parent Child)
 - Staff (Parent Child)
 - DOB
 - Gender
 - Staff Name
 - Manager Name
 - Products
 - Times (Time Dimension)

Business Model and Mapping

- Store Sales
 - Products (Level-Based)
 - Staff (Parent Child)
 - Stores (Ragged & Skip Level)
 - Times (Time Dim)
 - Products
 - Sales
 - Staff (Parent Child)
 - Stores (Ragged & Skip Level)
 - Times (Time Dimension)

Pivot Table

	Amount
Staff (Parent Child)	
Pierre Houdan	53
Adrian Boles	8
Alison Chisel	37
Cassandra Barry	11
Russell Durkee	5
Jon James	20
Lacey Laxson	11
Doris Lines	10
Malcom Doig	40
Vernice Kennebeck	19
Pete Sims	26
Ronald Koeman	15



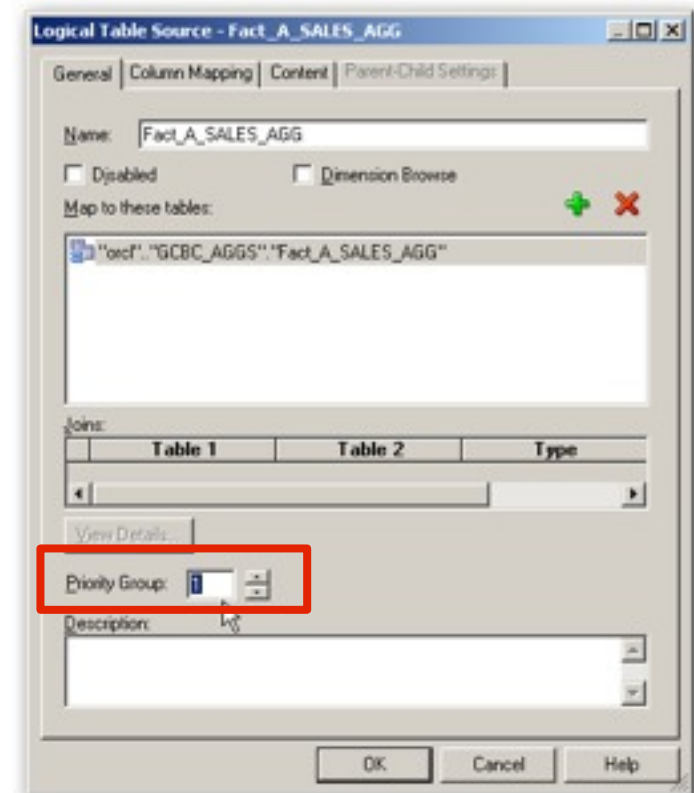
Demonstration

Skip-Level, Ragged and Parent-Child Hierarchies

T : +44 (0) 8446 697 995 or (888) 631 1410 (USA) E : enquiries@rittmanmead.com W: www.rittmanmead.com

LTS Priority Group Order

- A single logical table can have multiple logical table sources defined
- Usually, the selection of LTS by the BI Server is straightforward and defined by the context of the query
- When multiple aggregate LTS are defined, however, which one is chosen can be difficult to determine (based on **Number of Members At This Level**)
- OBIEE 11gR1 LTS Priority Group Order makes this more explicit
 - ▶ Lower value = higher priority
 - ▶ LTS Priority Group becomes main decider in which LTS to use



LTS Priority Group Example

- A logical fact table has two LTS mapped; one detail-level and one aggregate
- By default, the aggregate source is used for aggregated queries (due to dimension logical level mapping)
- This behaviour will be over-ridden though when LTS Priority Group is amended
 - ▶ Setting it to 1 makes it lower priority than the detail-level LTS

View Details...

Priority Group:



```
select sum(T879.FCAST_SAL_AMT) as c1,
       T873.PROD_CAT_DESC as c2
from
  GCBC_AGGS.A_PROD_CATEGORIES T873
  GCBC_AGGS.A_SALES AGG T879
where ( T873.PROD_CAT_DESC = T879.PROD_CAT_DESC )
group by T873.PROD_CAT_DESC
```

View Details...

Priority Group:



```
select sum(T117.FCAST_SAL_AMT) as c1,
       T107.PROD_CAT_DESC as c2
from
  GCBC_SALES.PRODUCTS T107 ,
  GCBC_SALES.SALES T117
where ( T107.PROD_ID = T117.PROD_ID )
group by T107.PROD_CAT_DESC
```



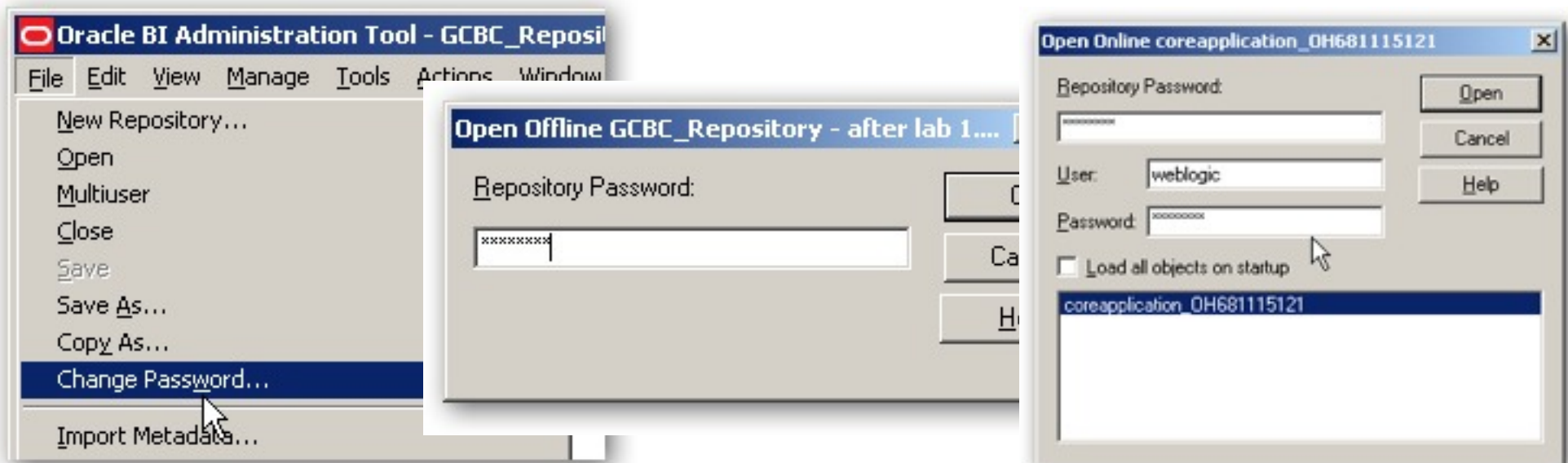

Demonstration

LTS Priority Group Order

T : +44 (0) 8446 697 995 or (888) 631 1410 (USA) E : enquiries@rittmanmead.com W: www.rittmanmead.com

Repository Passwords, and Repository Identity Management

- Repository files now have a password
 - Secures RPD file when there is no access to WebLogic Server
- RPD files are now encrypted, and compressed
- RPD Password is all that is required to edit RPD offline;
BIAdministrator application role required in addition to edit online



Deployment of RPD Files Through Enterprise Manager

- Repository files are now deployed using Fusion Middleware Control
- Select **Deployment > Repository > Lock and Edit Configuration**
- Select RPD file and enter repository password
- Press **Apply > Activate Changes > Restart to Apply Changes**

coreapplication Logged in as weblogic
Business Intelligence Instance Page Refreshed Aug 16, 2010 10:44:23 PM BST

Change Center: Activate Changes | Release Configuration Restart to apply recent changes

Overview Capacity Management Diagnostics Security **Deployment**

Presentation **Repository** Scheduler Marketing Mail

Upload BI Server Repository

Use this section to upload a new RPD and its password to your BI Server domain. You may also use this section to re-enter the password if a mistake was made on a previous upload.

Repository File GCBC_Repository - after lab 3.rpd

Repository Password

Confirm Password



Demonstration

Deploying RPDs through Fusion Middleware Control

T : +44 (0) 8446 697 995 or (888) 631 1410 (USA) E : enquiries@rittmanmead.com W: www.rittmanmead.com

Summary

- The goal of the semantic model is to simplify reporting data using a conformed logical dimensional model
- Federation capabilities allow us to model across multiple data sources
- Think in terms of dimensional modeling, particularly for the business model
- New features in OBIEE 11gR1 allow us to extend our modeling capability
- More complex hierarchies can be defined
- Lookup tables, and ID/Descriptive double columns can be defined
- Logical Table Source ordering is now more explicit
- RPD management is now handled through Fusion Middleware Control
- RPD administration security is externalized, and RPDs are now encrypted and secured



Oracle Business Intelligence 11g Masterclass

Oracle BI Server New Features & Data Modeling Best Practices

T : +44 (0) 8446 697 995 or (888) 631 1410 (USA) E : enquiries@rittmanmead.com W: www.rittmanmead.com