Setting Up OBIEE on a Snowflake-Heavy Data Warehouse

An Overview

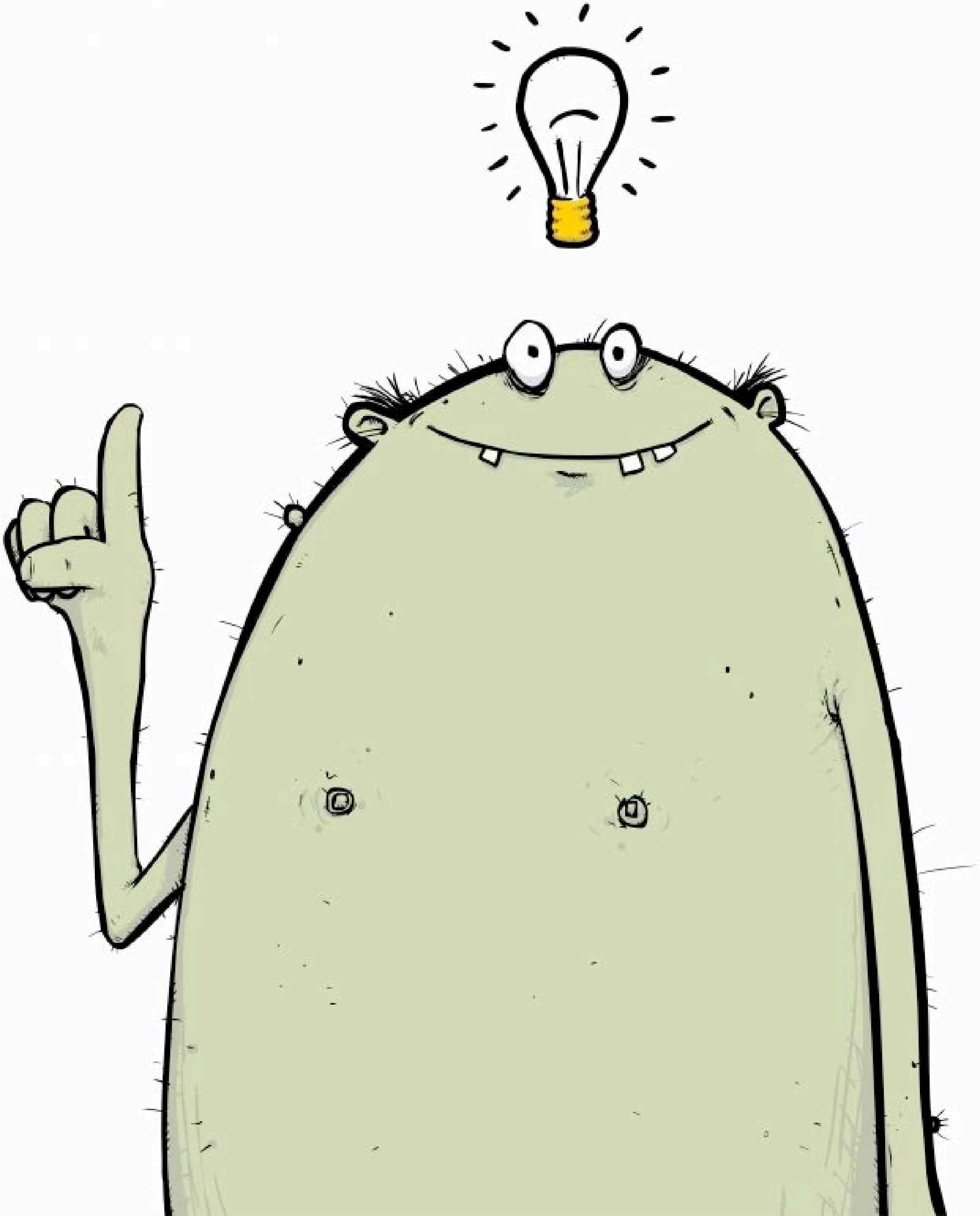
Rebecca Widom
Manager, Business Intelligence, Analysis, and Testing
Enterprise Data Warehouse
NYC Human Resources Administration

NYOUG
March 12, 2014

Today’s Workshop

- Introduction
- Best practices in OBIEE metadata repository design
- Our data and requirements, a.k.a. “You are a unique snowflake”
- Rules we follow
- Workarounds and rules we break
- Conclusion

Introduction



The Big Caveat

- Lessons from a single project.
- Workarounds by and for relative newbies trying to fit a snowflake legacy into a star-based product and address our particular requirements and data model.
- YMMV.
- Feedback, questions, additional conversation welcome!
    - If it weren’t, I wouldn’t be here!

Big thanks to the EDW team!

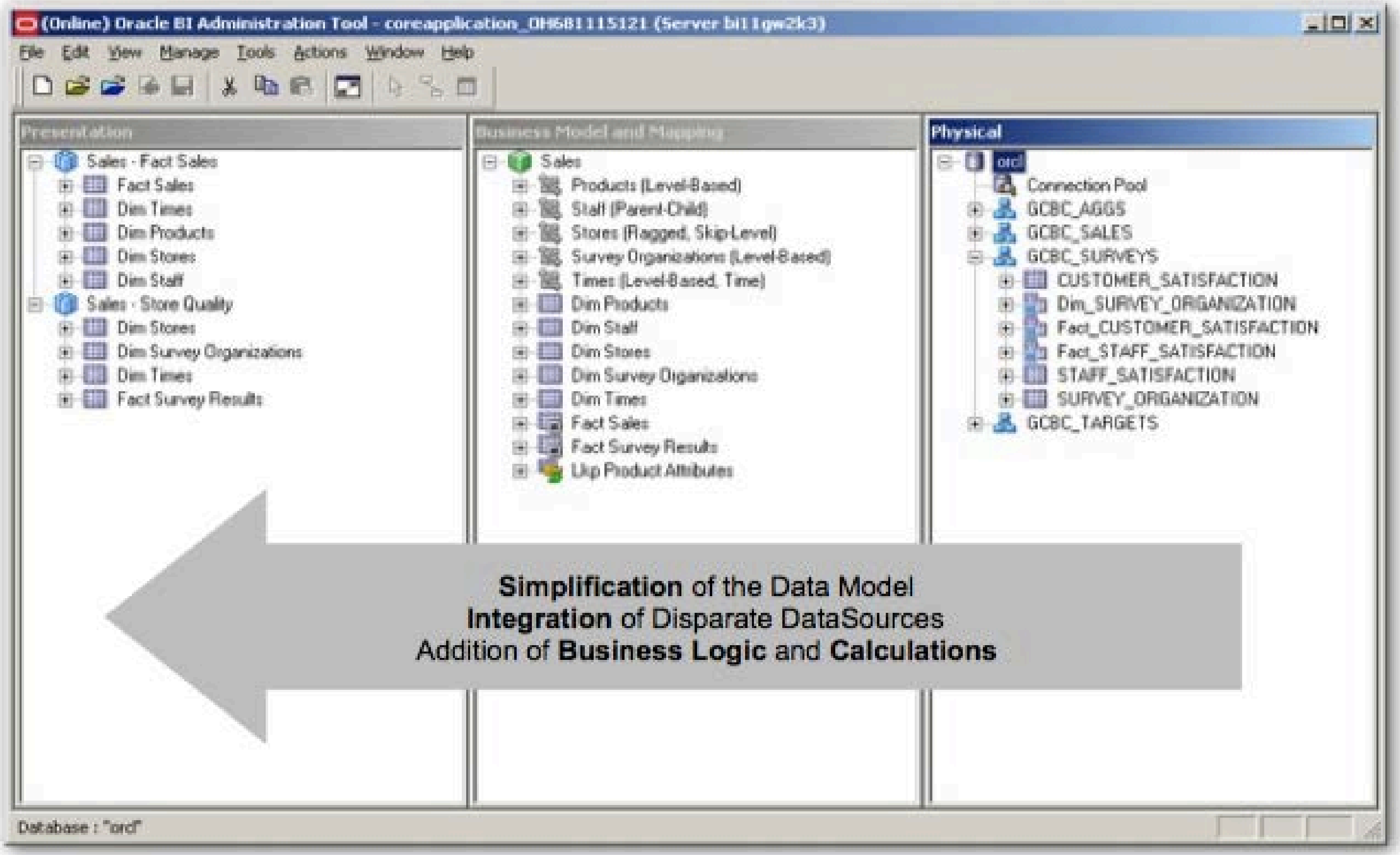
- Alfredo Veliz
- Alvin Woods
- Anil Tripathi
- Anna Stern
- Dinesh Veera
- Jane Neimand
- Marina Nunez
- Mihaela Iancu
- Minkie English
- Nick Gagliotti
- Oleg Gorelik
- Pavel Syrov
- Rachael Bickhardt
- Ravi Teppla
- Rochelle Eisenstein
- Ron Berry
- Sandy Slaughter
- Sanjay Patel
- Stan Rostov
- Suresh Muddaveerappa
- Venu Kadiyala
- Yasemin Turgut

Big thanks to our user advisors!

- Akinkunmi Akintunde
- Alexander Mattera
- Ann Kelleher
- Badar Chaudhry
- Bedros Boodanian
- Brian Graham-Jones
- Elsa Stazesky
- Eva Lazar
- Gordon Kraus-Friedberg
- Joan Dworetzky
- John Noel
- Jorge Burgos
- Joseph Varghese
- Kevin Fellner
- Margaret Boateng
- Mary Ellen O'Connell
- Michael Scianna
- Premal Shroff
- Sally Ramirez
- Sarah Haas
- Sean Blake
- Wah-Yuen Leung

Best Practices

Flow of Data Through the Three-Layer Semantic Model



Simplification of the Data Model
Integration of Disparate DataSources
Addition of **Business Logic** and **Calculations**

https://s3.amazonaws.com/rmc_docs/OOW2010_OBIEE_11gR1_Data_Modeling_Best_Practices_&_New_Features.pdf

Best Practices: Physical Layer

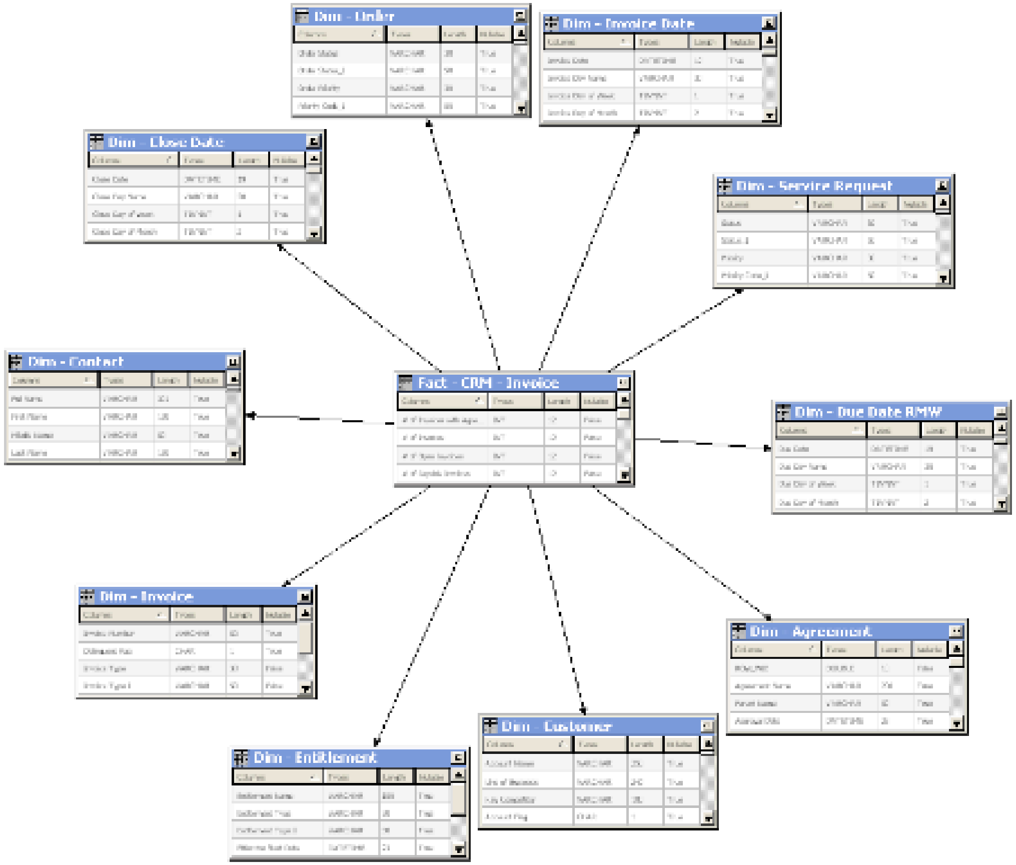
- Create aliases for all tables.
- Create keys, foreign keys, and other joins on the aliases, not the original tables.
- Use Opaque Views *only* as a last resort. Instead…
  - Apply filters in joins and logical tables sources, so that only the necessary tables are included in any given query, OR
  - Create tables or materialized views in ETL, so that computation doesn't have to happen on the fly.
- Avoid circular joins.
- https://blogs.oracle.com/pa/resource/CEAL_BIDesignBestPracticesV1.4.pdf

Best Practices: Business Model (BMM)

- Rename logical columns to use presentation names
- Keep only required columns in the BMM
- Dims
  - Assign business columns as primary keys
  - No aggregate measures
  - Create associated logical dimension hierarchy
- Facts
  - Create an implicit fact column mapped to 1, with no aggregation rule
  - All other columns should be aggregate measures
  - No logical/BMM primary key

Business Model Design

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

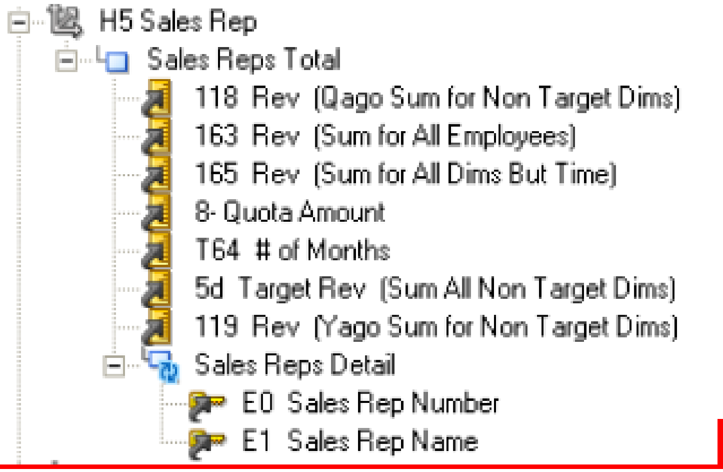


https://blogs.oracle.com/pa/resource/CEAL_BIDesignBestPracticesV1.4.pdf

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.
  - It is needed to set content level of logical table sources

Also necessary to avoid dropped filters in physical SQL.

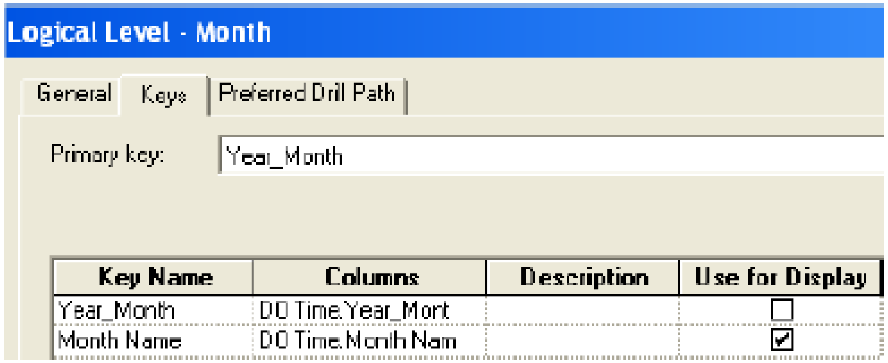


```
H5 Sales Rep
  Sales Reps Total
    118  Rev  (Qago Sum for Non Target Dims)
    163  Rev  (Sum for All Employees)
    165  Rev  (Sum for All Dims But Time)
    8- Quota Amount
    T64  # of Months
    5d  Target Rev  (Sum All Non Target Dims)
    119  Rev  (Yago Sum for Non Target Dims)
    Sales Reps Detail
      E0  Sales Rep Number
      E1  Sales Rep Name
```

https://blogs.oracle.com/pa/resource/CEAL_BIDesignBestPracticesV1.4.pdf

Level Keys

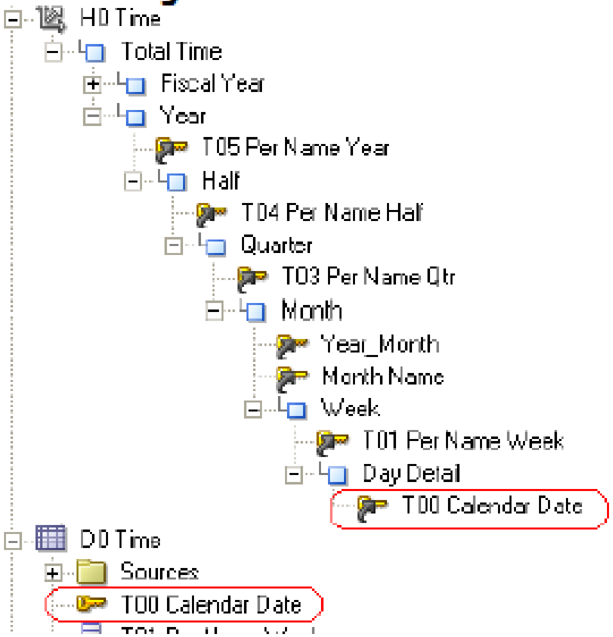
- The primary key of each level must always be unique
- The primary key of the lowest level of the hierarchy must always be the primary of the logical table



Logical Level - Month

Primary key: Year_Month

Key Name	Columns	Description	Use for Display
Year_Month	D0 Time.Year_Mont		☐
Month Name	D0 Time.Month Nam		☑



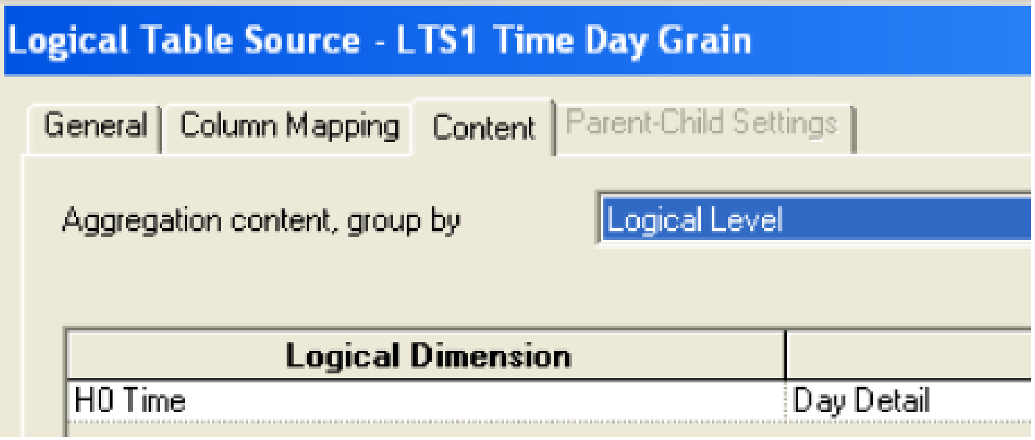
```
H0 Time
  Total Time
    Fiscal Year
    Year
      T05 Per Name Year
      Half
        T04 Per Name Half
        Quarter
          T03 Per Name Qtr
          Month
            Year_Month
            Month Name
            Week
              T01 Per Name Week
              Day Detail
                T00 Calendar Date
D0 Time
  Sources
  T00 Calendar Date
```

https://blogs.oracle.com/pa/resource/CEAL_BIDesignBestPracticesV1.4.pdf

Content Level

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



Logical Table Source - LTS1 Time Day Grain

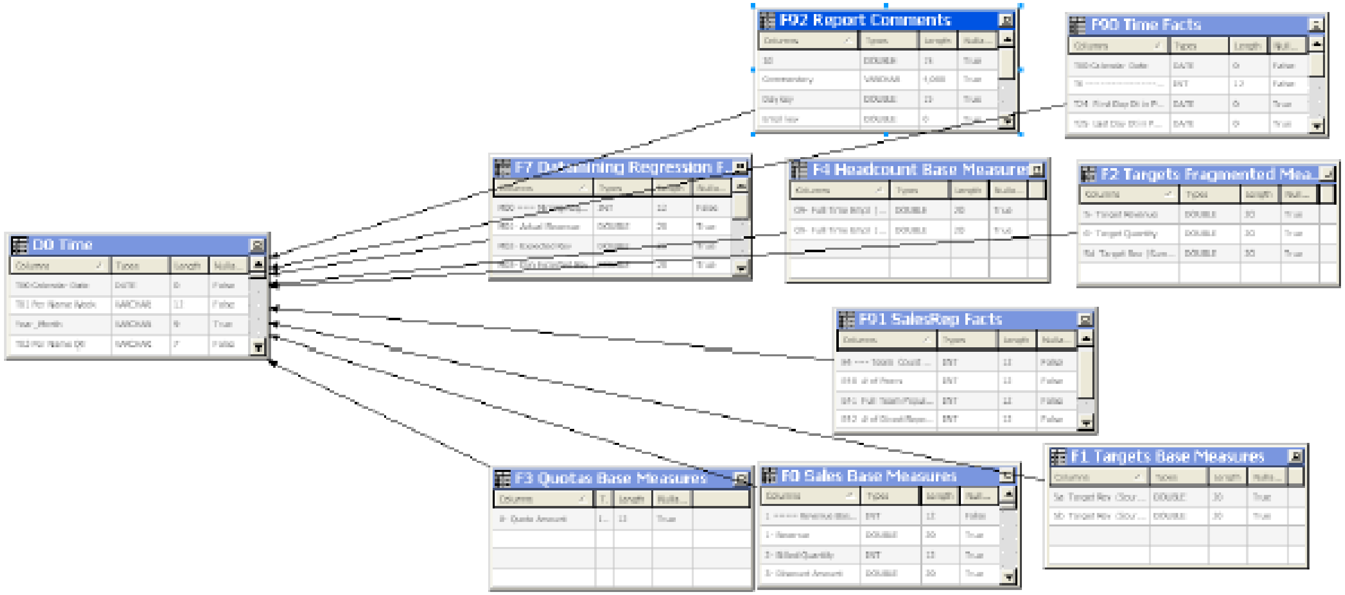
Aggregation content, group by: Logical Level

Logical Dimension	
H0 Time	Day Detail

https://blogs.oracle.com/pa/resource/CEAL_BIDesignBestPracticesV1.4.pdf

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.



https://blogs.oracle.com/pa/resource/CEAL_BIDesignBestPracticesV1.4.pdf

Best Practices: Presentation Layer

- Simple subject areas with a few facts as possible, and ones that share dimensions.
- Configure presentation folders to each type of user.
- Add descriptions for subject areas, folders, and columns.

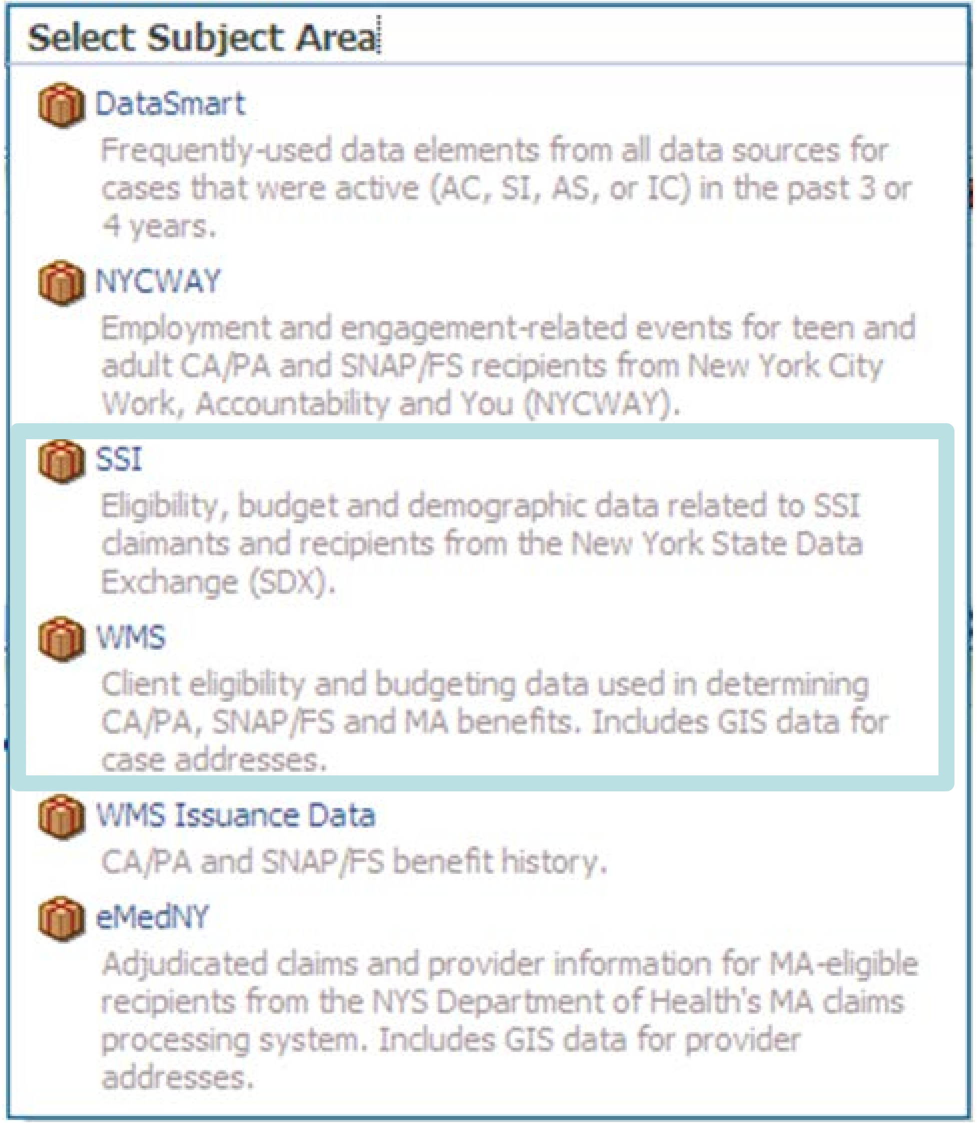
More Best Practices…

- s3.amazonaws.com/rmc_docs/OOW2010_OBIEE_11gR1_Data_Modeling_Best_Practices_&_New_Features.pdf
- blogs.oracle.com/pa/resource/CEAL_BIDesignBestPracticesV1.4.pdf
- obieepedia.wordpress.com/category/obiee-best-practices/
- debaatobiee.wordpress.com/category/obiee/best-practices/
- allaboutobiee.blogspot.com/2012/03/obiee-best-practices-in-bmm-layer.html
- www.varanasisaichand.com/2011/08/dimensional-hierarchies-best-practices.html

you
are unique
just like
everyone
else

Data Sources: WMS (and SSI)

- Welfare Management System and SSI State Data Exchange
- Budgeting, demographics, GIS for all programs
- SCD2 for lawsuits and audits
  - Millions of clients and families, 15 years of history
  - 450+ data elements
- **Monster dims plus code definitions**
- Aggregate measures: count distinct



Select Subject Area

DataSmart

Frequently-used data elements from all data sources for cases that were active (AC, SI, AS, or IC) in the past 3 or 4 years.

NYCWAY

Employment and engagement-related events for teen and adult CA/PA and SNAP/FS recipients from New York City Work, Accountability and You (NYCWAY).

SSI

Eligibility, budget and demographic data related to SSI claimants and recipients from the New York State Data Exchange (SDX).

WMS

Client eligibility and budgeting data used in determining CA/PA, SNAP/FS and MA benefits. Includes GIS data for case addresses.

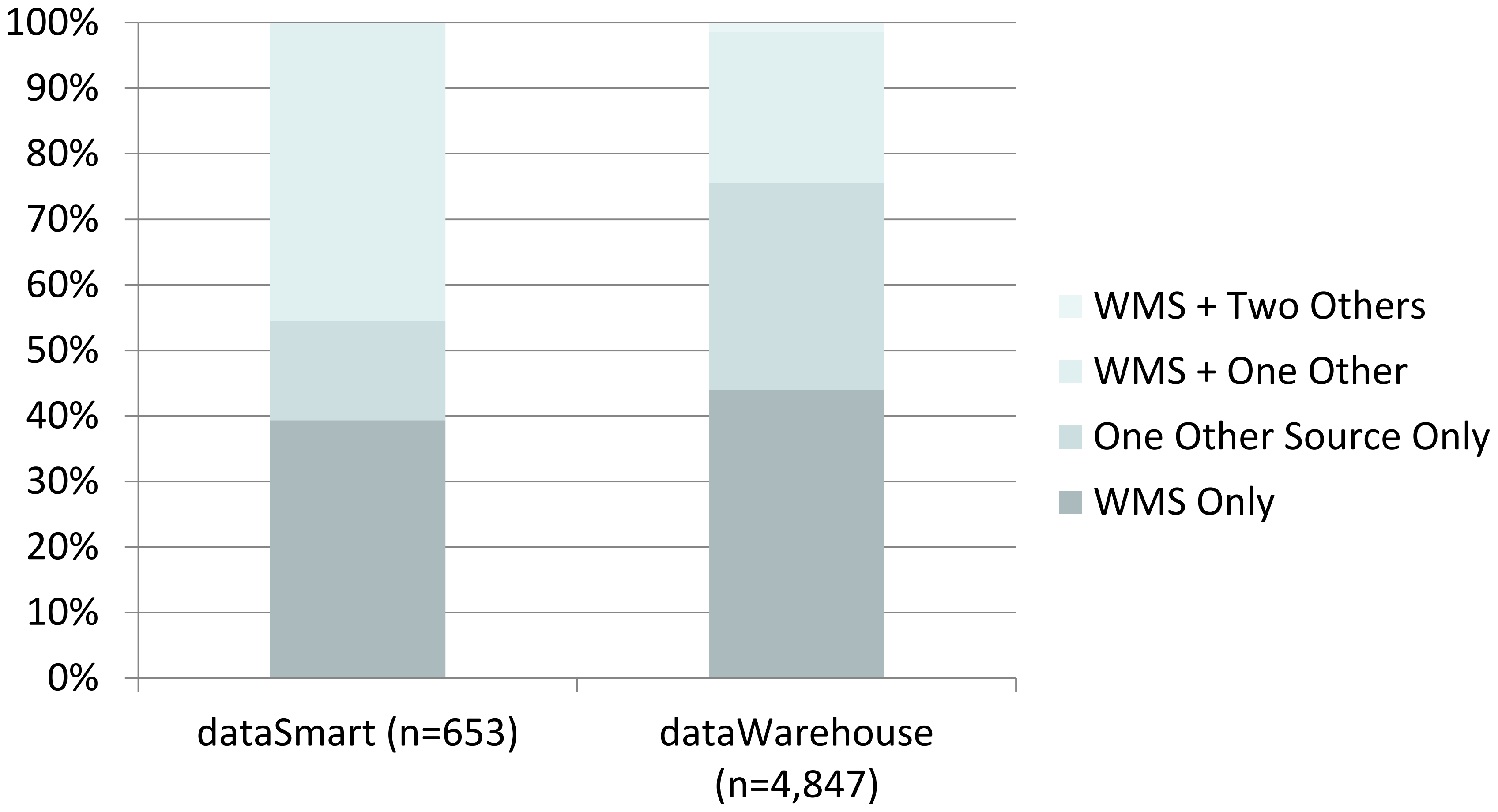
WMS Issuance Data

CA/PA and SNAP/FS benefit history.

eMedNY

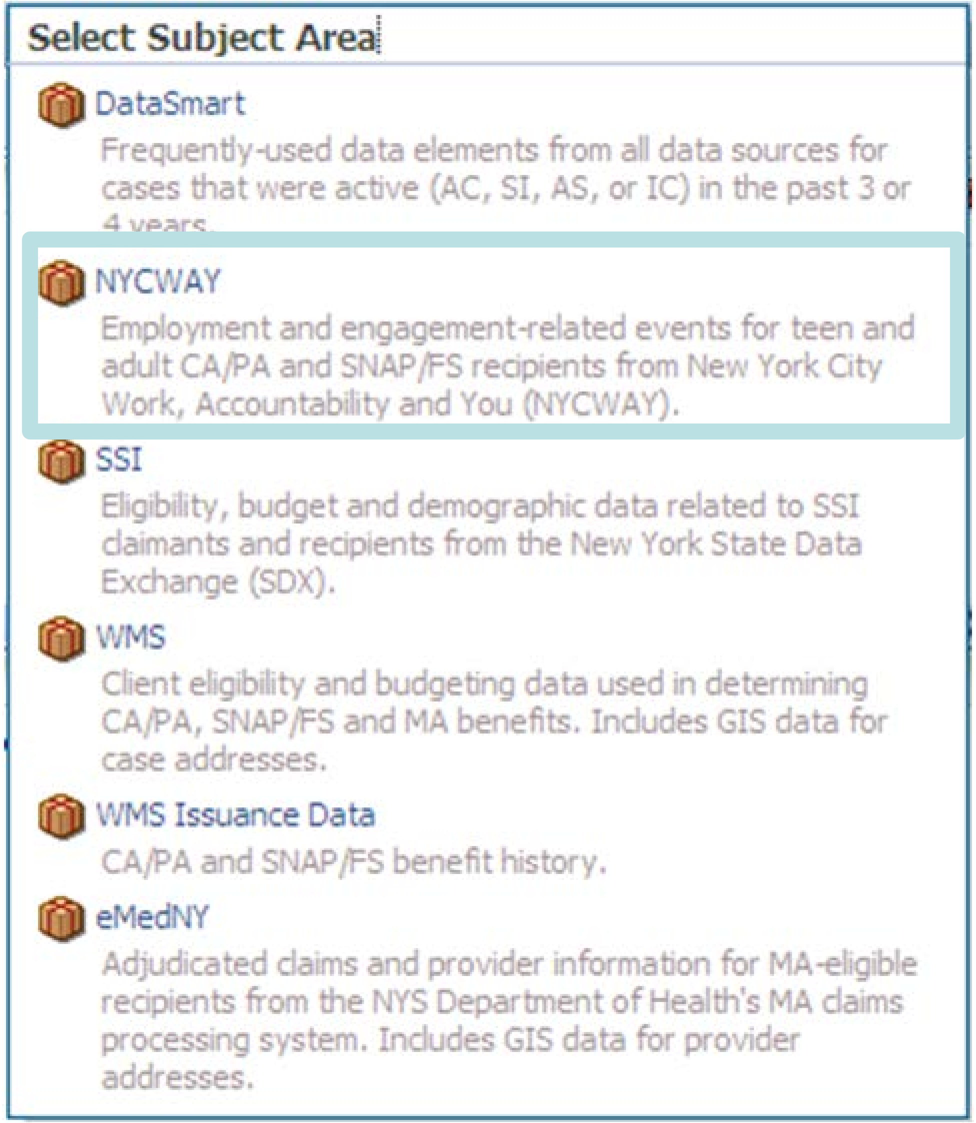
Adjudicated claims and provider information for MA-eligible recipients from the NYS Department of Health's MA claims processing system. Includes GIS data for provider addresses.

Usage by Source



Data Sources: NYCWAY

- New York City Work Accountability & You
- Employment services & case management
- This happened, then this happened, then…
- Factless Facts plus code definitions
- Aggregate measures are all count distinct



Select Subject Area

DataSmart
Frequently-used data elements from all data sources for cases that were active (AC, SI, AS, or IC) in the past 3 or 4 years.

NYCWAY
Employment and engagement-related events for teen and adult CA/PA and SNAP/FS recipients from New York City Work, Accountability and You (NYCWAY).

SSI
Eligibility, budget and demographic data related to SSI claimants and recipients from the New York State Data Exchange (SDX).

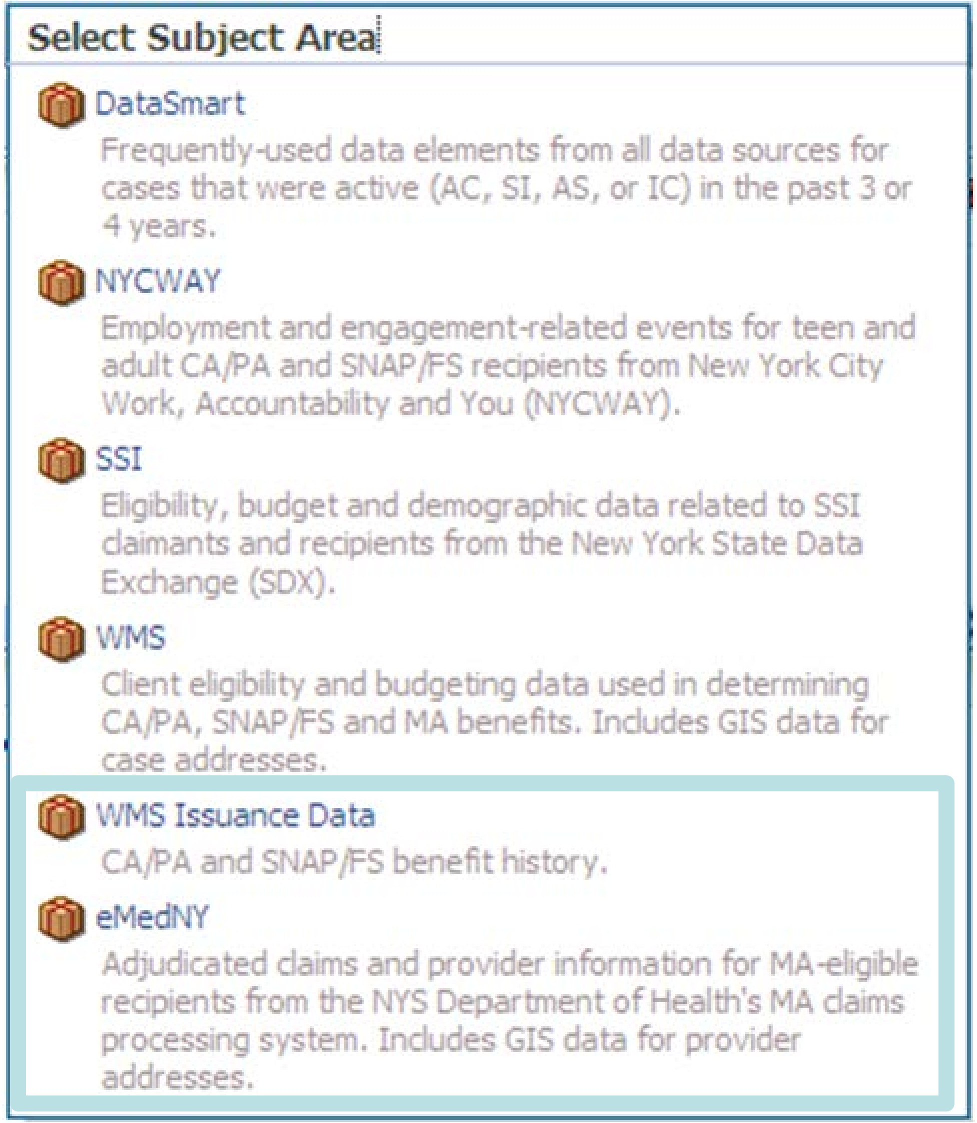
WMS
Client eligibility and budgeting data used in determining CA/PA, SNAP/FS and MA benefits. Includes GIS data for case addresses.

WMS Issuance Data
CA/PA and SNAP/FS benefit history.

eMedNY
Adjudicated claims and provider information for MA-eligible recipients from the NYS Department of Health's MA claims processing system. Includes GIS data for provider addresses.

Data Sources: Issuances & eMedNY

- Payments made to or on behalf of clients and client households.
- Finally, dollars to sum and nice star models!



Select Subject Area

DataSmart

Frequently-used data elements from all data sources for cases that were active (AC, SI, AS, or IC) in the past 3 or 4 years.

NYCWAY

Employment and engagement-related events for teen and adult CA/PA and SNAP/FS recipients from New York City Work, Accountability and You (NYCWAY).

SSI

Eligibility, budget and demographic data related to SSI claimants and recipients from the New York State Data Exchange (SDX).

WMS

Client eligibility and budgeting data used in determining CA/PA, SNAP/FS and MA benefits. Includes GIS data for case addresses.

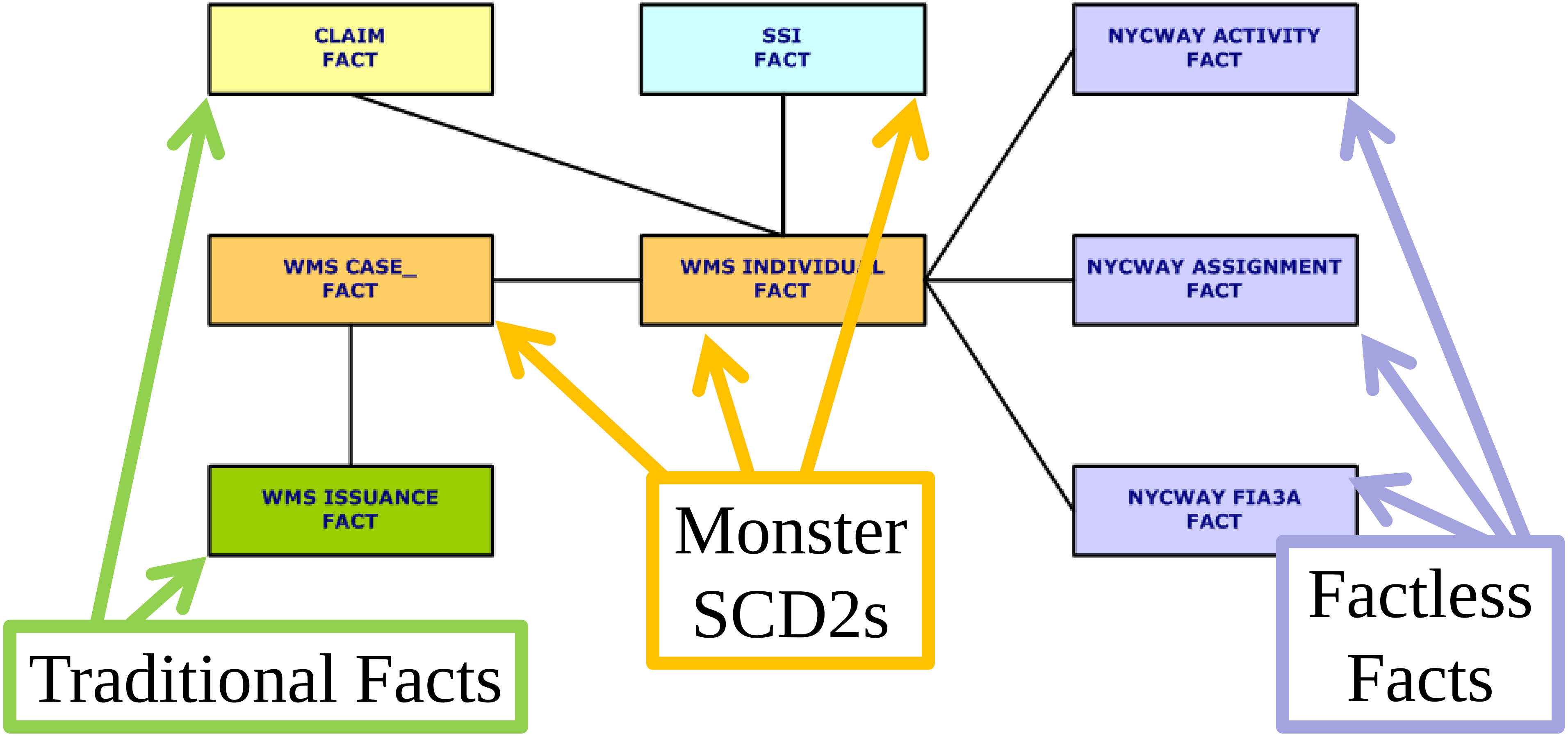
WMS Issuance Data

CA/PA and SNAP/FS benefit history.

eMedNY

Adjudicated claims and provider information for MA-eligible recipients from the NYS Department of Health's MA claims processing system. Includes GIS data for provider addresses.

Data Mart overview from Discoverer



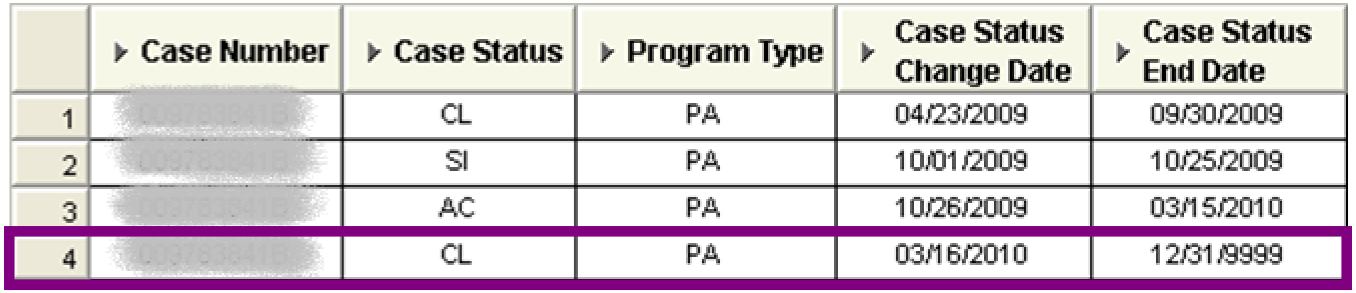
Code and Time Dimensions not pictured here

More on SCD2s

- Change dates let you know when the record was in effect
- Most recent data has an end date of 12/31/9999
- If ***any*** column changes in the table we add a new row and update effective dates.
- With the requirement of hundreds of fields and full SCD2 history, we have had to denormalize.

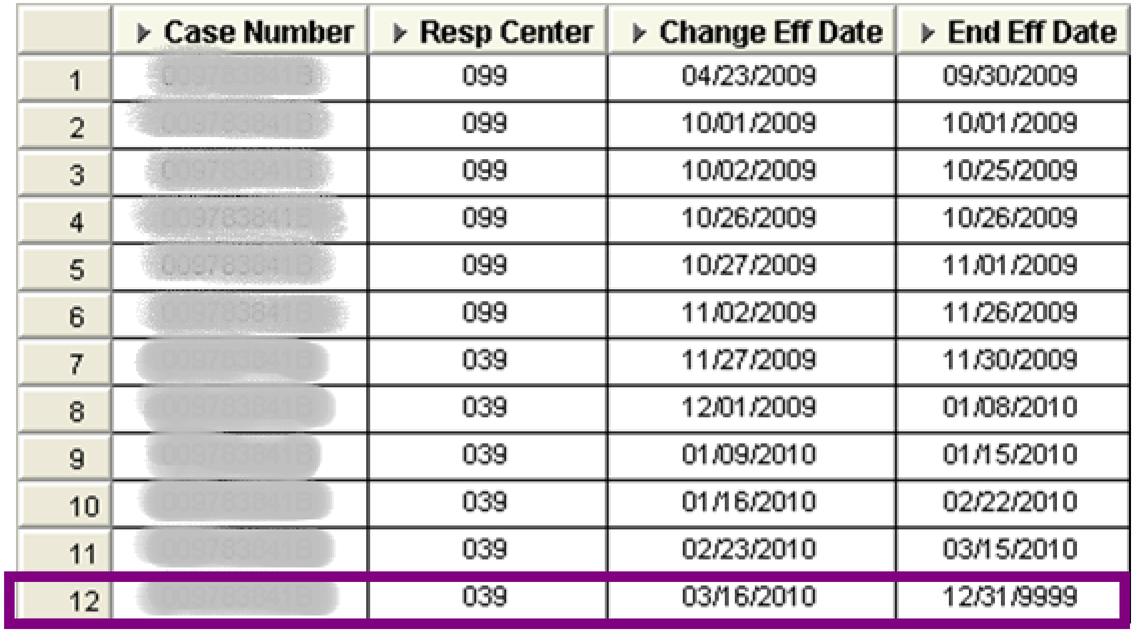
SCD2: Sample

Case Status



	Case Number	Case Status	Program Type	Case Status Change Date	Case Status End Date
1	[redacted]	CL	PA	04/23/2009	09/30/2009
2	[redacted]	SI	PA	10/01/2009	10/25/2009
3	[redacted]	AC	PA	10/26/2009	03/15/2010
4	[redacted]	CL	PA	03/16/2010	12/31/9999

Responsible Center (Case Suffix Dim)



	Case Number	Resp Center	Change Eff Date	End Eff Date
1	[redacted]	099	04/23/2009	09/30/2009
2	[redacted]	099	10/01/2009	10/01/2009
3	[redacted]	099	10/02/2009	10/25/2009
4	[redacted]	099	10/26/2009	10/26/2009
5	[redacted]	099	10/27/2009	11/01/2009
6	[redacted]	099	11/02/2009	11/26/2009
7	[redacted]	039	11/27/2009	11/30/2009
8	[redacted]	039	12/01/2009	01/08/2010
9	[redacted]	039	01/09/2010	01/15/2010
10	[redacted]	039	01/16/2010	02/22/2010
11	[redacted]	039	02/23/2010	03/15/2010
12	[redacted]	039	03/16/2010	12/31/9999

SCD2: Single date conditions

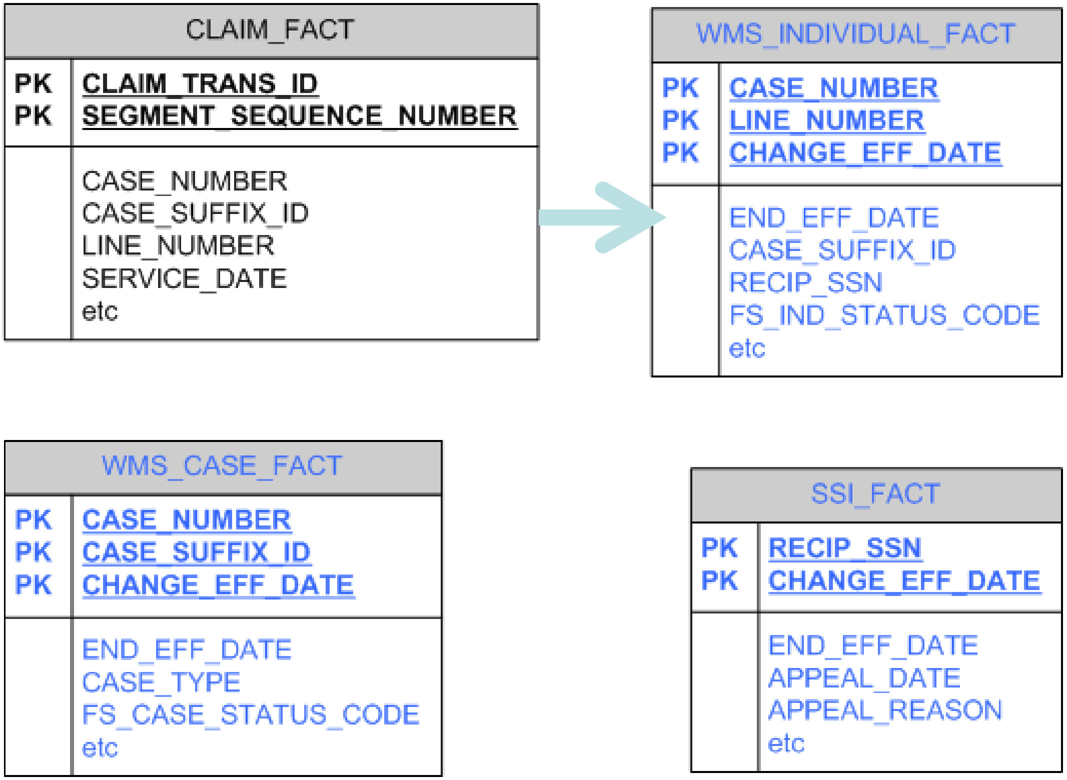
- Today or some other day
  - Who is currently active for Food Stamps/SNAP?
  - Who was active for Food Stamps/SNAP on July 1, 2013?
  - What was the status on the service date of this claim?
- One record per case, case suffix, case line, or ssn, whichever is the rest of the table key
- No risk of multiplied sums if the rest of the join is correct

Joins: Monster Dim to Monster Dim

- Different SCD2s for the same client get new records on different days.
- So, Change Eff Date A does not necessarily equal Change Eff Date B.
- Instead, identify pairs of records that were in effect on overlapping dates.
- Many to many join, even for a single client

	Individual Status			Recipient Dim		
	Ind Status	Change Eff Date	End Eff Date	SSN Validation	Change Eff Date	End Eff Date
✓	Active	01/01/2007	01/14/2008	1	01/01/2007	01/10/2007
✓	Active	01/01/2007	01/14/2008	8	01/11/2007	12/31/9999
✗	Sanction	01/15/2008	12/31/9999	1	01/01/2007	01/10/2007
✓	Sanction	01/15/2008	12/31/9999	8	01/11/2007	12/31/9999

Implications for Joins: Traditional Fact → SCD2



CLAIM_FACT	
PK	CLAIM_TRANS_ID
PK	SEGMENT_SEQUENCE_NUMBER
	CASE_NUMBER CASE_SUFFIX_ID LINE_NUMBER SERVICE_DATE etc

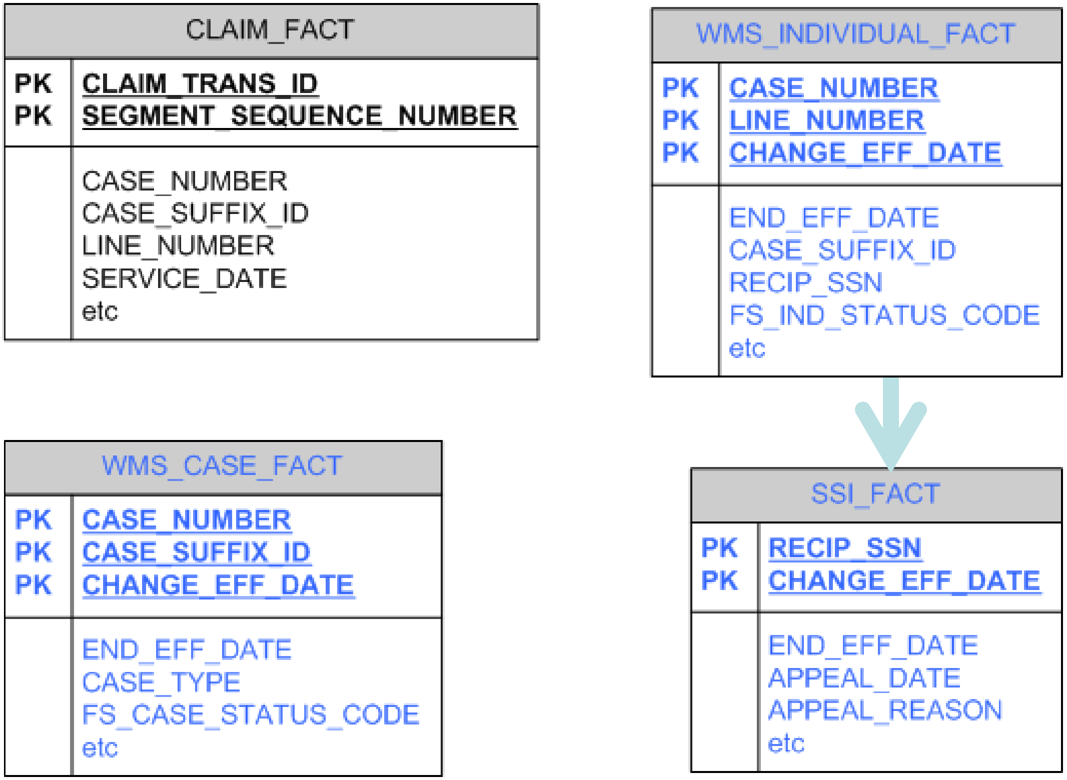
WMS_INDIVIDUAL_FACT	
PK	CASE_NUMBER
PK	LINE_NUMBER
PK	CHANGE_EFF_DATE
	END_EFF_DATE CASE_SUFFIX_ID RECIP_SSN FS_IND_STATUS_CODE etc

WMS_CASE_FACT	
PK	CASE_NUMBER
PK	CASE_SUFFIX_ID
PK	CHANGE_EFF_DATE
	END_EFF_DATE CASE_TYPE FS_CASE_STATUS_CODE etc

SSI_FACT	
PK	RECIP_SSN
PK	CHANGE_EFF_DATE
	END_EFF_DATE APPEAL_DATE APPEAL_REASON etc

- Select a single day, no risk of multiplication
  - Fact date field BETWEEN change_eff_date AND end_eff_date
  - OR
  - end_eff_date = 12/31/9999

Implications for Joins: SCD2 → SCD2



CLAIM_FACT	
PK	CLAIM_TRANS_ID
PK	SEGMENT_SEQUENCE_NUMBER
	CASE_NUMBER
	CASE_SUFFIX_ID
	LINE_NUMBER
	SERVICE_DATE
	etc

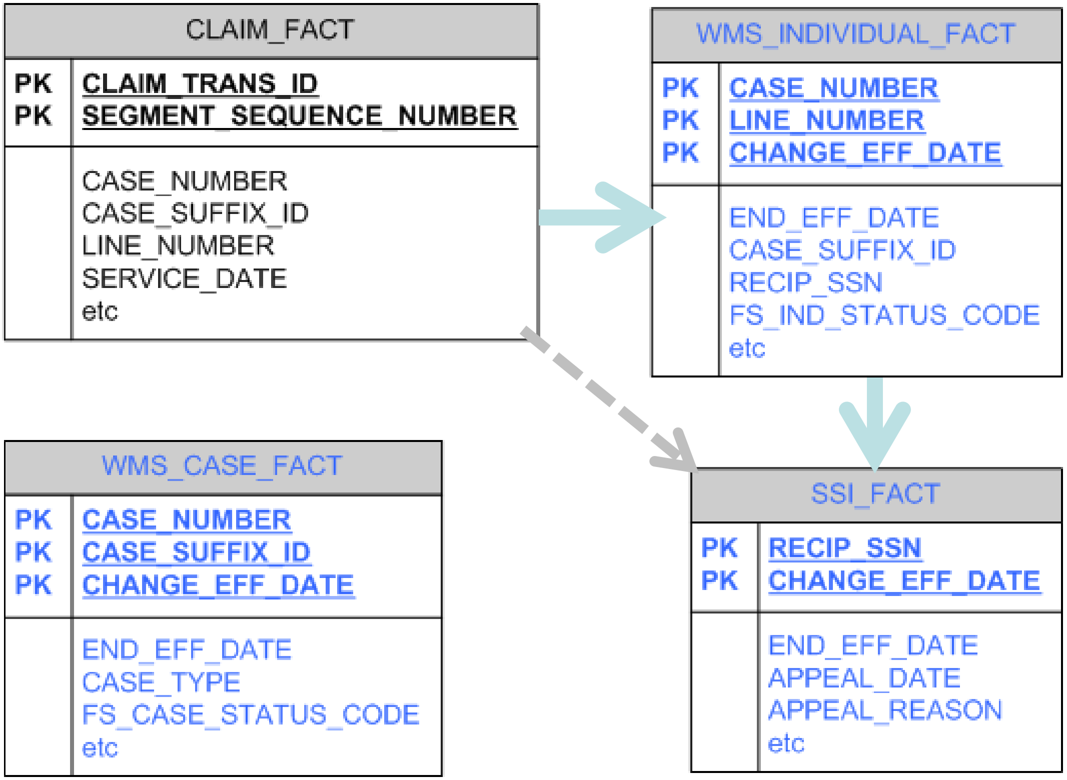
WMS_CASE_FACT	
PK	CASE_NUMBER
PK	CASE_SUFFIX_ID
PK	CHANGE_EFF_DATE
	END_EFF_DATE
	CASE_TYPE
	FS_CASE_STATUS_CODE
	etc

WMS_INDIVIDUAL_FACT	
PK	CASE_NUMBER
PK	LINE_NUMBER
PK	CHANGE_EFF_DATE
	END_EFF_DATE
	CASE_SUFFIX_ID
	RECIP_SSN
	FS_IND_STATUS_CODE
	etc

SSI_FACT	
PK	RECIP_SSN
PK	CHANGE_EFF_DATE
	END_EFF_DATE
	APPEAL_DATE
	APPEAL_REASON
	etc

- Monster Dim to Monster Dim
  - dim1.change_eff_date <= dim2.end_eff_date
  - AND
  - dim1.end_eff_date >= dim2.change_eff_date
- May get multiple records in the time frame
- Count distinct is fine

Implications for Joins: Fact → SCD2 → SCD2



CLAIM_FACT	
PK	CLAIM_TRANS_ID
PK	SEGMENT_SEQUENCE_NUMBER
	CASE_NUMBER CASE_SUFFIX_ID LINE_NUMBER SERVICE_DATE etc

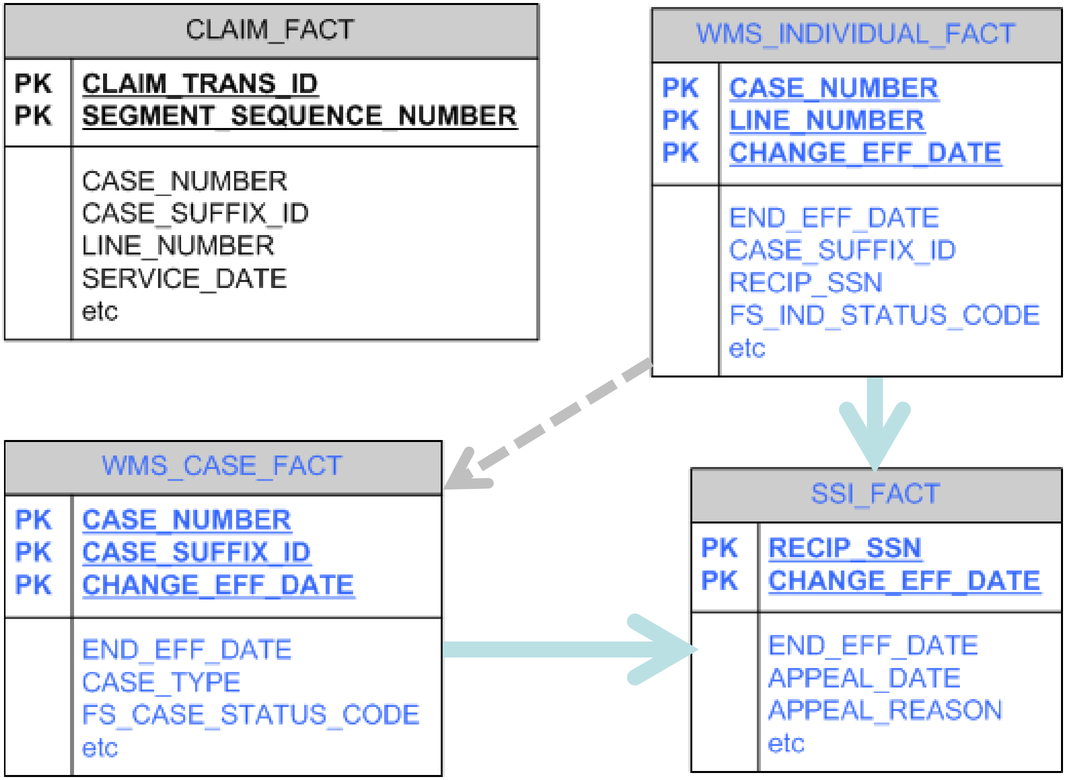
WMS_INDIVIDUAL_FACT	
PK	CASE_NUMBER
PK	LINE_NUMBER
PK	CHANGE_EFF_DATE
	END_EFF_DATE CASE_SUFFIX_ID RECIP_SSN FS_IND_STATUS_CODE etc

WMS_CASE_FACT	
PK	CASE_NUMBER
PK	CASE_SUFFIX_ID
PK	CHANGE_EFF_DATE
	END_EFF_DATE CASE_TYPE FS_CASE_STATUS_CODE etc

SSI_FACT	
PK	RECIP_SSN
PK	CHANGE_EFF_DATE
	END_EFF_DATE APPEAL_DATE APPEAL_REASON etc

- Select a single day *from each & every monster dim*: Most recent or fact date
- Any dim in the query without a single date condition could multiply sums.
- Here, we need SSN from dim1 and date from the fact.

Implications for Joins SCD2 → SCD2 → SCD2



CLAIM_FACT

PK CLAIM_TRANS_ID
PK SEGMENT_SEQUENCE_NUMBER

CASE_NUMBER
CASE_SUFFIX_ID
LINE_NUMBER
SERVICE_DATE
etc

WMS_INDIVIDUAL_FACT

PK CASE_NUMBER
PK LINE_NUMBER
PK CHANGE_EFF_DATE

END_EFF_DATE
CASE_SUFFIX_ID
RECIP_SSN
FS_IND_STATUS_CODE
etc

WMS_CASE_FACT

PK CASE_NUMBER
PK CASE_SUFFIX_ID
PK CHANGE_EFF_DATE

END_EFF_DATE
CASE_TYPE
FS_CASE_STATUS_CODE
etc

SSI_FACT

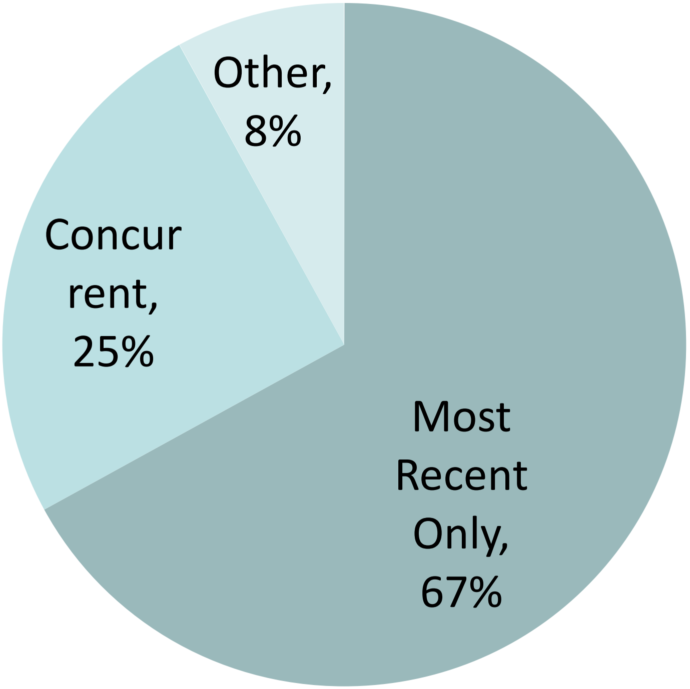
PK RECIP_SSN
PK CHANGE_EFF_DATE

END_EFF_DATE
APPEAL_DATE
APPEAL_REASON
etc

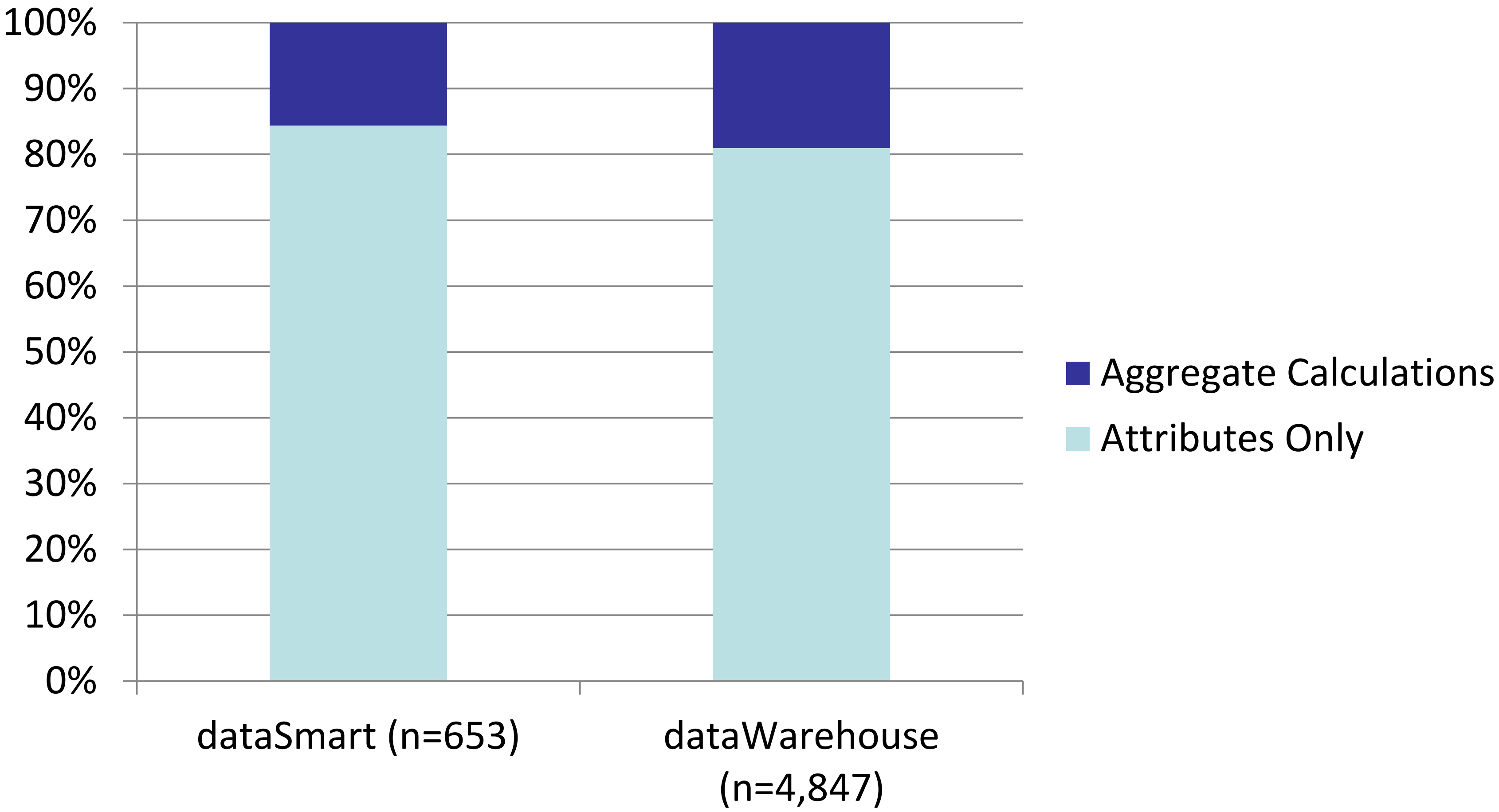
- Overlapping time periods
  - ssi.change_eff_date <= ind.end_eff_date AND ssi.end_eff_date >= ind.change_eff_date AND
  - ssi.change_eff_date <= cas.end_eff_date AND ssi.end_eff_date >= cas.change_eff_date AND
  - cas.change_eff_date <= ind.end_eff_date AND cas.end_eff_date >= ind.change_eff_date
- >1 row per case line
- Circular join
- Count Distinct is OK

Date Logic Usage in Discoverer (Approximate)

- Discoverer has optional "Most recent" filter for each SCD2
- Historical analysis training allows users to do a variety of queries
- Sometimes users make mistakes



Aggregate Measure Usage in Discoverer



Project Requirements

- Create access to the data available in Discoverer.
- Give users capabilities from Discoverer, with usability improvements from OBIEE.
- Maintain existing flexibility in date logic, while improving usability.
- Add commonly used aggregate measures, while supporting many users' attribute-only focus.
  - Don't worry about aggregate fact tables.
- Design one business model across sources to serve *all* ad hoc authors across the enterprise.
- Minimize changes and additions to the database and ETL.

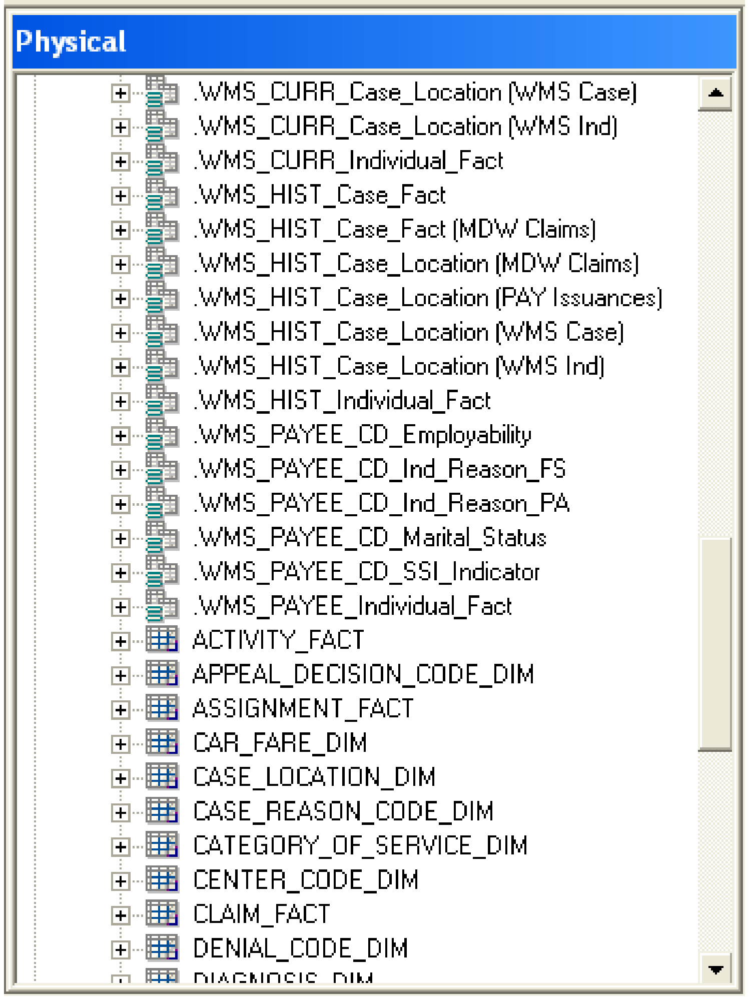
System/Platform Info

System Component	Most of Project	Very Recent Upgrade
OBIEE Product Version	11.1.1.6.2	11.1.1.7.1
Operating System/Version	Oracle Solaris on SPARC (64-bit) – 10	Oracle Solaris on SPARC (64-bit) – 11
Database/Version	Oracle Database - Enterprise Edition 11.2.0.3	Oracle Database - Enterprise Edition 11.2.0.3

KEEP
CALM
AND
FOLLOW
THE RULES

Physical Layer: Always Use Aliases

- More on this later, but it’s important.
- Find naming conventions that work for your team.

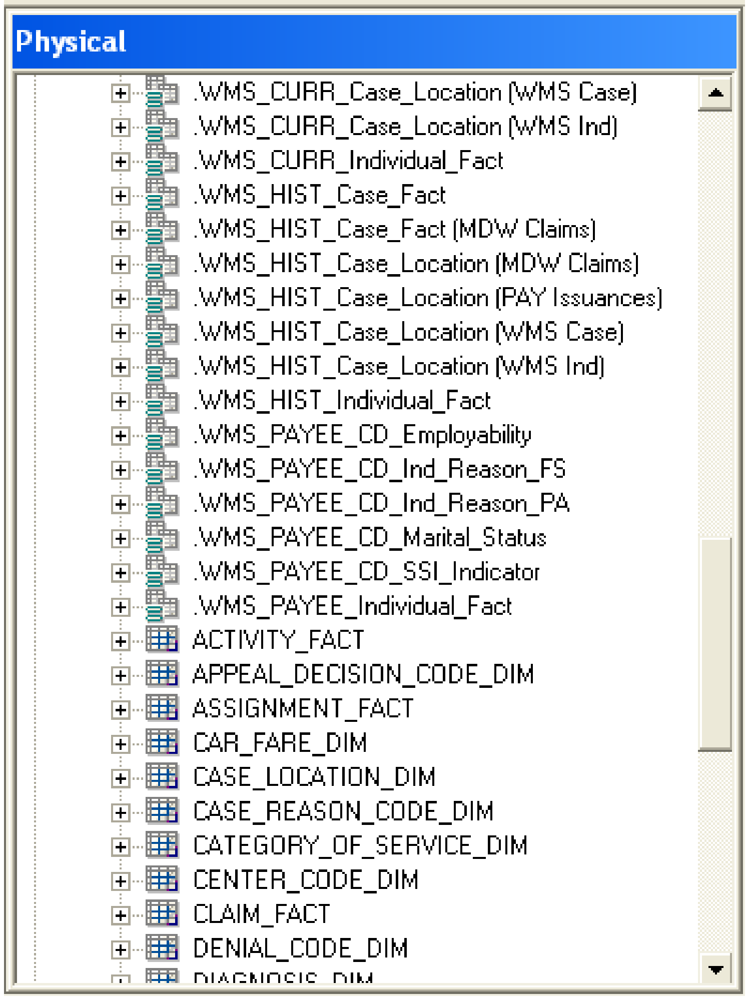


Physical

.WMS_CURR_Case_Location (WMS Case)
.WMS_CURR_Case_Location (WMS Ind)
.WMS_CURR_Individual_Fact
.WMS_HIST_Case_Fact
.WMS_HIST_Case_Fact (MDW Claims)
.WMS_HIST_Case_Location (MDW Claims)
.WMS_HIST_Case_Location (PAY Issuances)
.WMS_HIST_Case_Location (WMS Case)
.WMS_HIST_Case_Location (WMS Ind)
.WMS_HIST_Individual_Fact
.WMS_PAYEE_CD_Employability
.WMS_PAYEE_CD_Ind_Reason_FS
.WMS_PAYEE_CD_Ind_Reason_PA
.WMS_PAYEE_CD_Marital_Status
.WMS_PAYEE_CD_SSI_Indicator
.WMS_PAYEE_Individual_Fact
ACTIVITY_FACT
APPEAL_DECISION_CODE_DIM
ASSIGNMENT_FACT
CAR_FARE_DIM
CASE_LOCATION_DIM
CASE_REASON_CODE_DIM
CATEGORY_OF_SERVICE_DIM
CENTER_CODE_DIM
CLAIM_FACT
DENIAL_CODE_DIM
DIAGNOSIS_DIM

Physical Layer: Avoid Opaque Views

- WMS_PAYEE_ is one example
- Rather than forcing OBIEE to include all of those tables, let it decide which is best.
- We have a lot of complex joins in the physical layer to handle this.

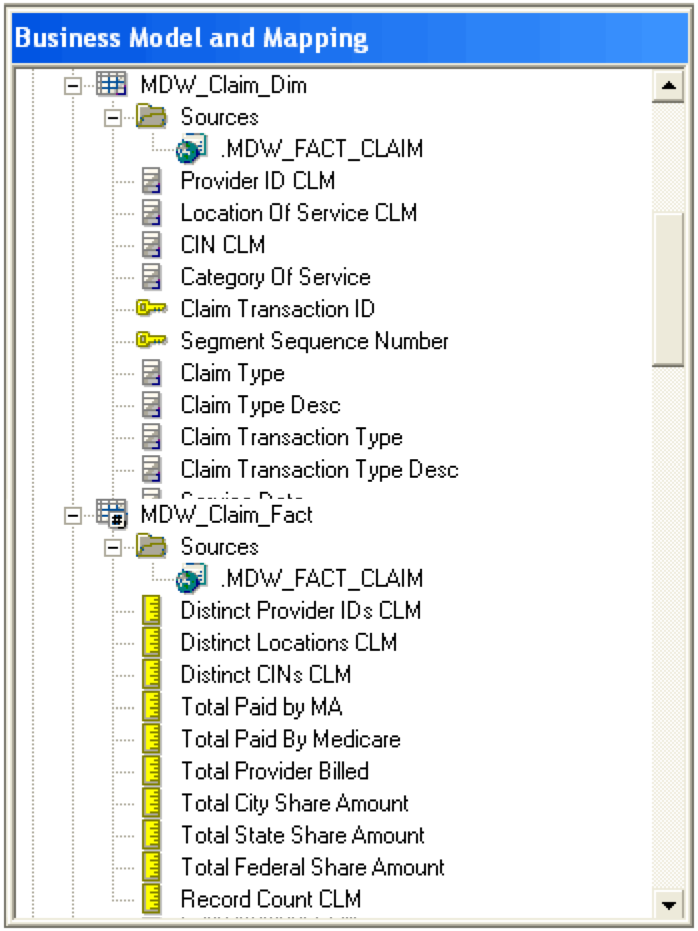


Physical

```
.WMS_CURR_Case_Location (WMS Case)
.WMS_CURR_Case_Location (WMS Ind)
.WMS_CURR_Individual_Fact
.WMS_HIST_Case_Fact
.WMS_HIST_Case_Fact (MDW Claims)
.WMS_HIST_Case_Location (MDW Claims)
.WMS_HIST_Case_Location (PAY Issuances)
.WMS_HIST_Case_Location (WMS Case)
.WMS_HIST_Case_Location (WMS Ind)
.WMS_HIST_Individual_Fact
.WMS_PAYEE_CD_Employability
.WMS_PAYEE_CD_Ind_Reason_FS
.WMS_PAYEE_CD_Ind_Reason_PA
.WMS_PAYEE_CD_Marital_Status
.WMS_PAYEE_CD_SSI_Indicator
.WMS_PAYEE_Individual_Fact
ACTIVITY_FACT
APPEAL_DECISION_CODE_DIM
ASSIGNMENT_FACT
CAR_FARE_DIM
CASE_LOCATION_DIM
CASE_REASON_CODE_DIM
CATEGORY_OF_SERVICE_DIM
CENTER_CODE_DIM
CLAIM_FACT
DENIAL_CODE_DIM
DIAGNOSIS_DIM
```

BMM: Facts vs. Dims

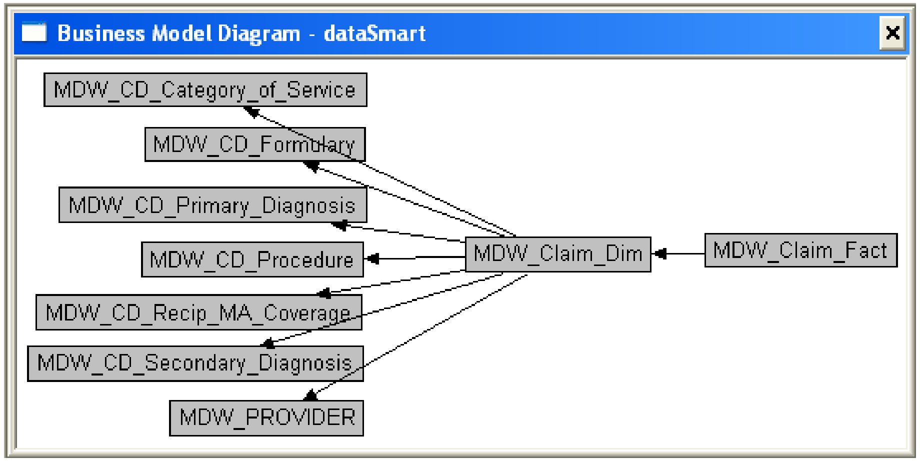
- Same Physical Layer Alias
- Claim Dim
  - Attribute columns only
  - Has logical business field primary key
- Claim Fact
  - Aggregate measure columns only
  - No key

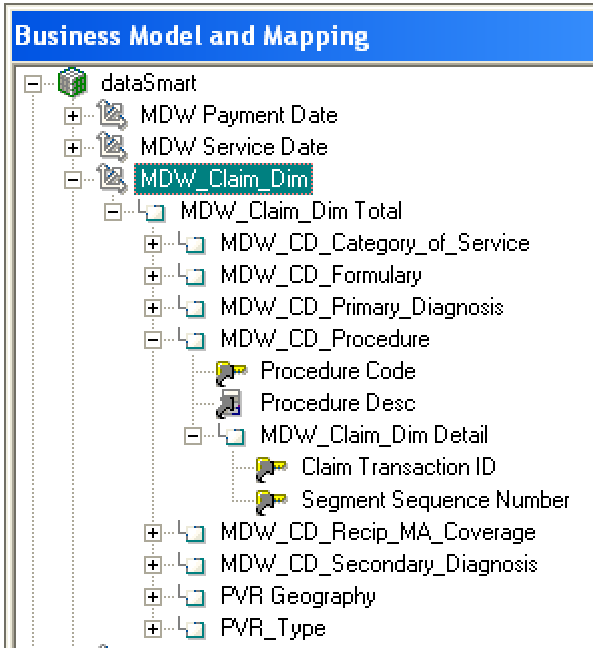


Business Model and Mapping

```
MDW_Claim_Dim
    Sources
        .MDW_FACT_CLAIM
    Provider ID CLM
    Location Of Service CLM
    CIN CLM
    Category Of Service
    Claim Transaction ID
    Segment Sequence Number
    Claim Type
    Claim Type Desc
    Claim Transaction Type
    Claim Transaction Type Desc
MDW_Claim_Fact
    Sources
        .MDW_FACT_CLAIM
    Distinct Provider IDs CLM
    Distinct Locations CLM
    Distinct CINs CLM
    Total Paid by MA
    Total Paid By Medicare
    Total Provider Billed
    Total City Share Amount
    Total State Share Amount
    Total Federal Share Amount
    Record Count CLM
```

BMM: All dims in hierarchies

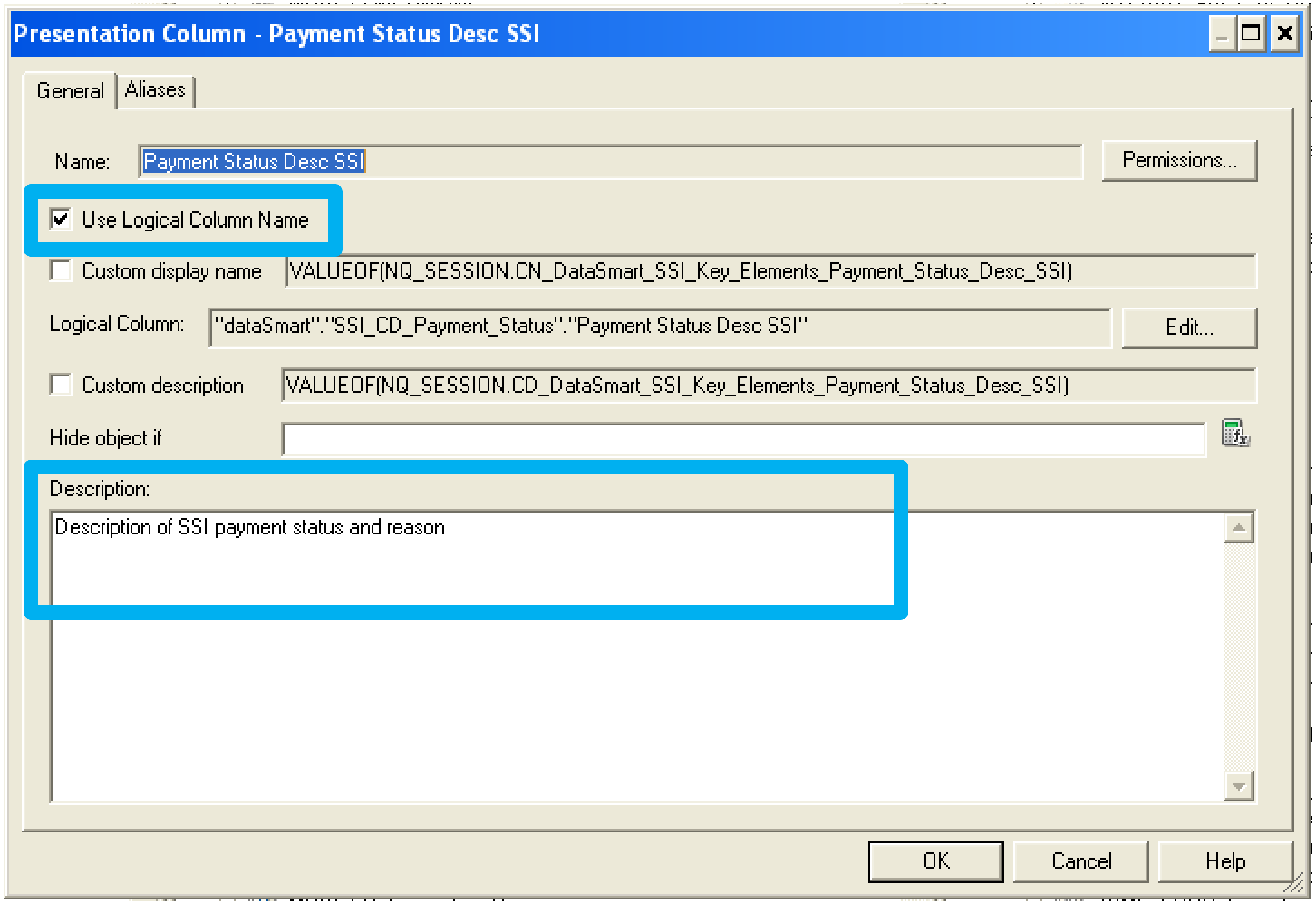



Business Model and Mapping

- dataSmart
  - MDW Payment Date
  - MDW Service Date
  - MDW_Claim_Dim
    - MDW_Claim_Dim Total
      - MDW_CD_Category_of_Service
      - MDW_CD_Formulary
      - MDW_CD_Primary_Diagnosis
      - MDW_CD_Procedure
        - Procedure Code
        - Procedure Desc
        - MDW_Claim_Dim Detail
          - Claim Transaction ID
          - Segment Sequence Number
      - MDW_CD_Recip_MA_Coverage
      - MDW_CD_Secondary_Diagnosis
      - PVR Geography
      - PVR_Type

- Create default logical dimension hierarchy
  - Create BMM tables and joins, snowflake is ok
  - Create correct logical keys
  - Right click on dim closest to the fact (MDW_Claim_Dim)
  - Choose: Create Logical Dimension > Dimension with Level Based Hierarchy
- Add levels as desired, keeping same total and detail levels on all paths
- OBIEE may drop filters on dims that aren't in hierarchies.

Presentation: Names and Descriptions



Name: Payment Status Desc SSI

☑ Use Logical Column Name

□ Custom display name VALUEOF(NQ_SESSION.CN_DataSmart_SSI_Key_Elements_Payment_Status_Desc_SSI)

Logical Column: "dataSmart"."SSI_CD_Payment_Status"."Payment Status Desc SSI"

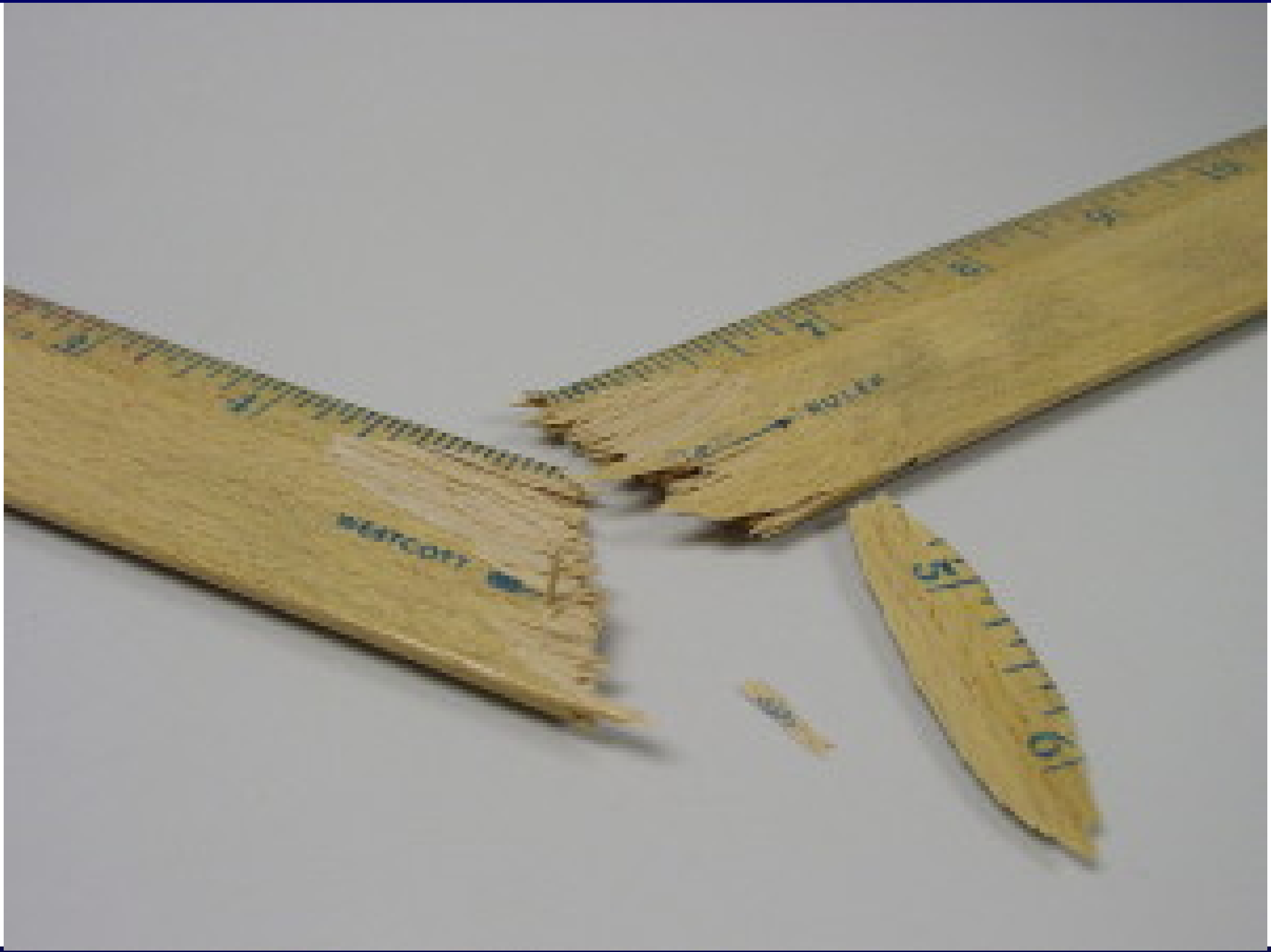
□ Custom description VALUEOF(NQ_SESSION.CD_DataSmart_SSI_Key_Elements_Payment_Status_Desc_SSI)

Hide object if

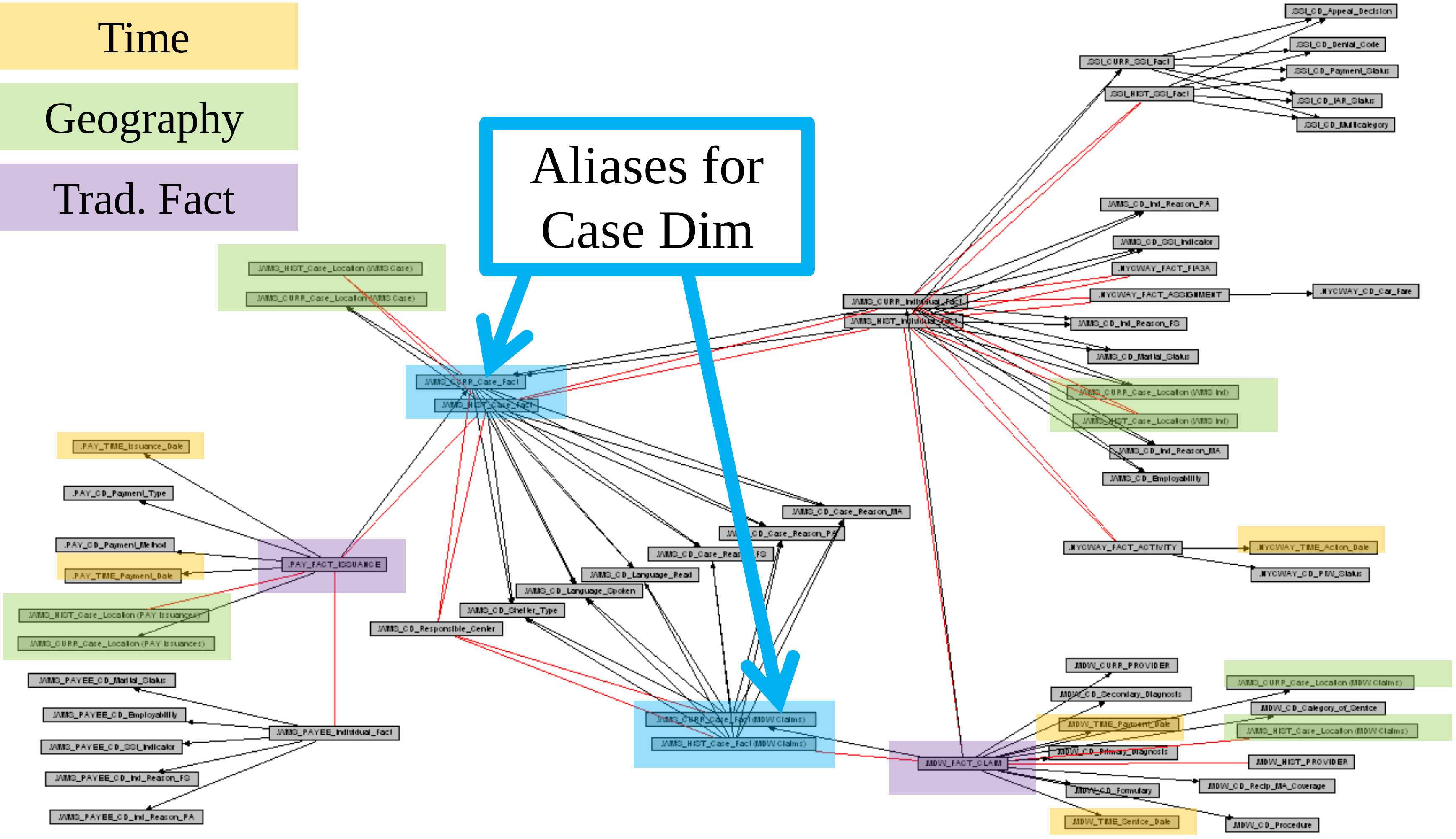
Description:

Description of SSI payment status and reason

Workarounds and Rules We Break



Physical Layer: Use aliases instead of circular joins



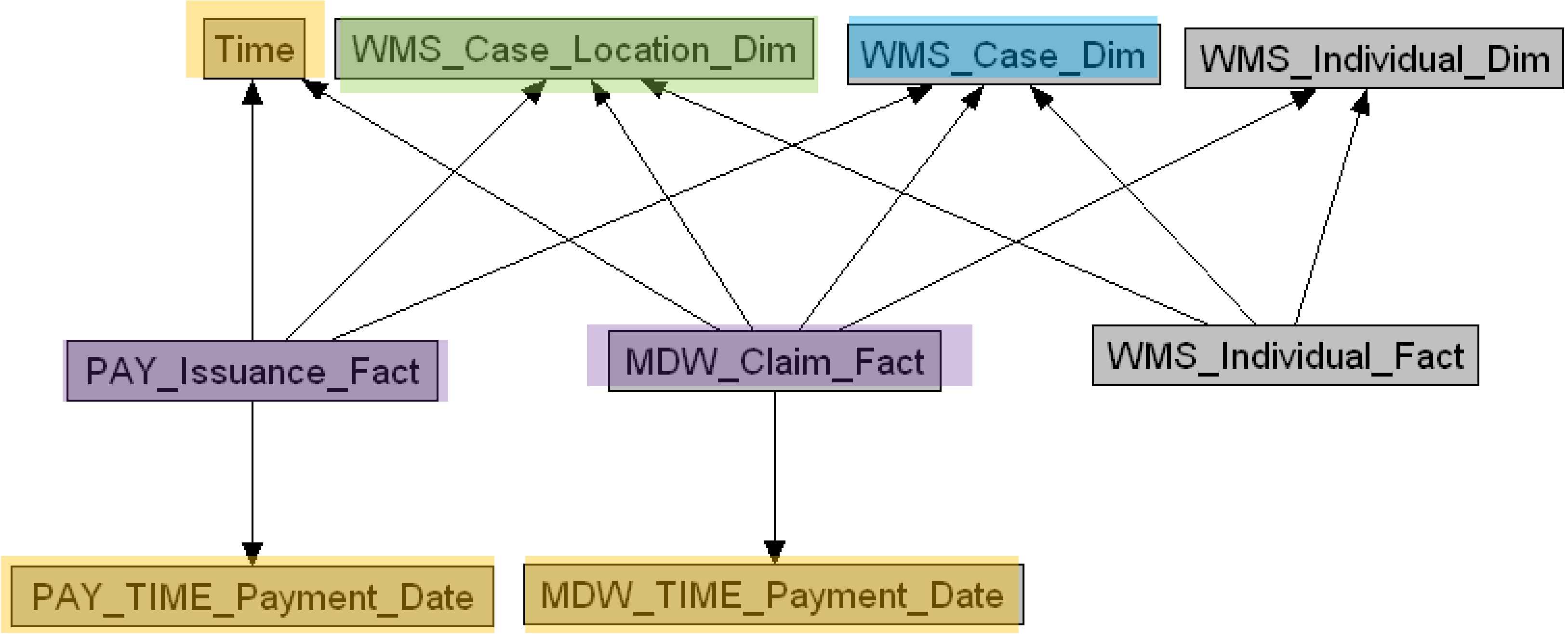
Time

Geography

Trad. Fact

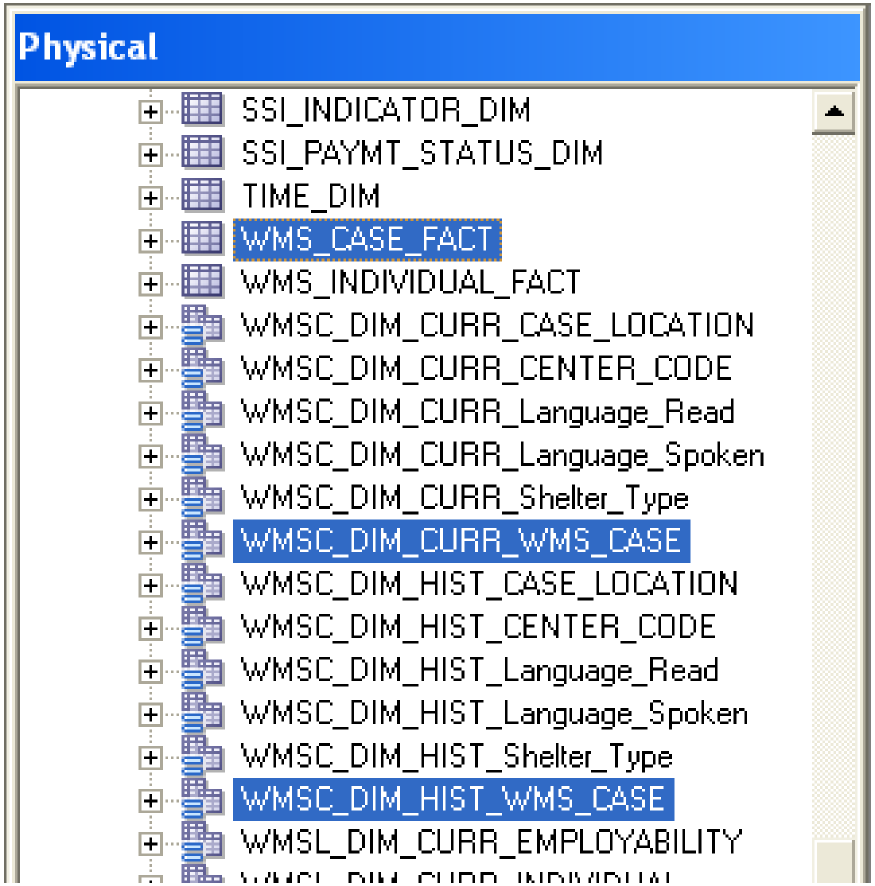
Aliases for Case Dim

Time, Geography and Case Dims Combined in BMM



Default Most Recent, Step 1: More Aliases

- For each slowly changing dimension, create two aliases: one for current and one for history.
- History has end_eff_date as part of the primary key, current does not.
- Without correct keys, OBIEE has no way to choose the better table.

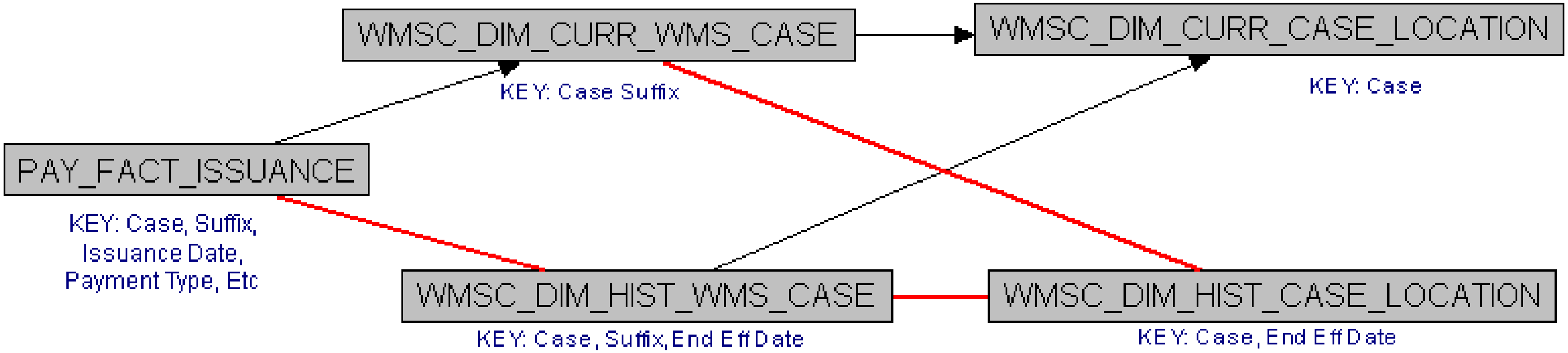


Physical

- SSI_INDICATOR_DIM
- SSI_PAYMT_STATUS_DIM
- TIME_DIM
- WMS_CASE_FACT
- WMS_INDIVIDUAL_FACT
- WMSC_DIM_CURR_CASE_LOCATION
- WMSC_DIM_CURR_CENTER_CODE
- WMSC_DIM_CURR_Language_Read
- WMSC_DIM_CURR_Language_Spoken
- WMSC_DIM_CURR_Shelter_Type
- WMSC_DIM_CURR_WMS_CASE
- WMSC_DIM_HIST_CASE_LOCATION
- WMSC_DIM_HIST_CENTER_CODE
- WMSC_DIM_HIST_Language_Read
- WMSC_DIM_HIST_Language_Spoken
- WMSC_DIM_HIST_Shelter_Type
- WMSC_DIM_HIST_WMS_CASE
- WMSL_DIM_CURR_EMPLOYABILITY
- WMSL_DIM_CURR_INDIVIDUAL

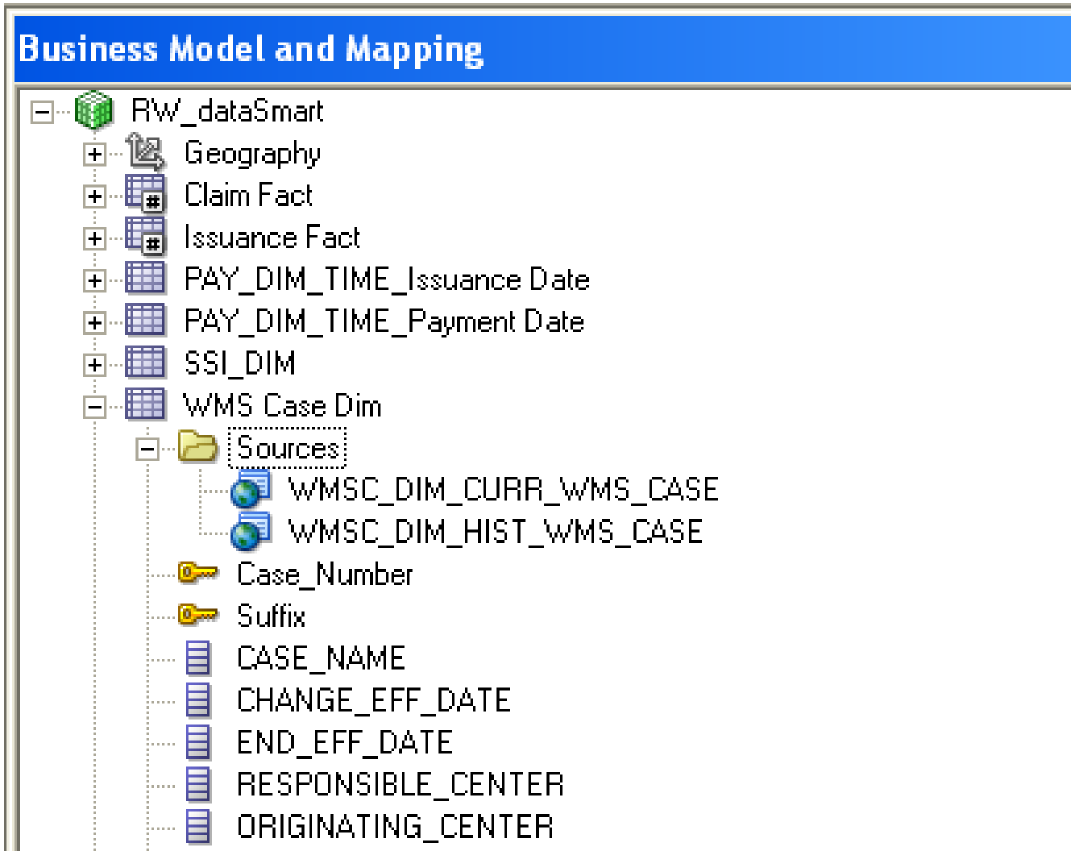
Default Most Recent, Step 2: Joins Among Aliases

- Physical layer joins do *not* include date conditions.
  - Many-to-many “complex” joins to history aliases.
  - Simple foreign key joins to the current aliases.


WMSC_DIM_CURR_WMS_CASE
KEY: Case Suffix
WMSC_DIM_CURR_CASE_LOCATION
KEY: Case
PAY_FACT_ISSUANCE
KEY: Case, Suffix, Issuance Date, Payment Type, Etc
WMSC_DIM_HIST_WMS_CASE
KEY: Case, Suffix, End Eff Date
WMSC_DIM_HIST_CASE_LOCATION
KEY: Case, End Eff Date

Default Most Recent, Step 3: Combined Logical Tables

- One BMM folder for each slowly changing dimension.
- Two logical table sources, one for history and one for current.
- Logical key is the same as the <u>**Current**</u> primary key.

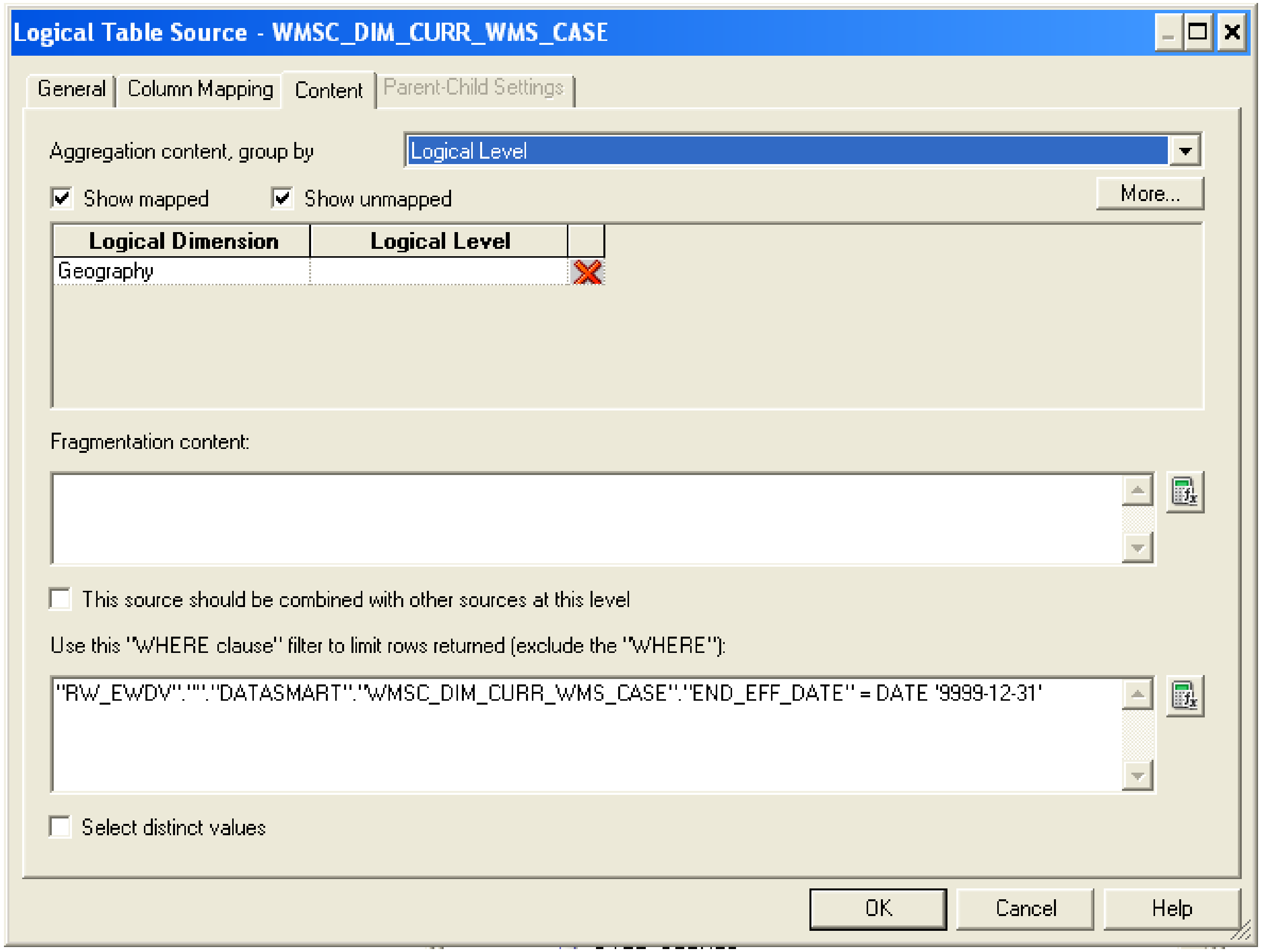


Business Model and Mapping

```
RW_dataSmart
    Geography
    Claim Fact
    Issuance Fact
    PAY_DIM_TIME_Issuance Date
    PAY_DIM_TIME_Payment Date
    SSI_DIM
    WMS Case Dim
        Sources
            WMSC_DIM_CURR_WMS_CASE
            WMSC_DIM_HIST_WMS_CASE
        Case_Number
        Suffix
        CASE_NAME
        CHANGE_EFF_DATE
        END_EFF_DATE
        RESPONSIBLE_CENTER
        ORIGINATING_CENTER
```

Default Most Recent, Step 4: Set Where Clause on the Current LTS

- In the **current** logical table source,
- On the “Content” tab,
- Add a “WHERE clause” on the field that is part of the history primary key, but not the current:
- end_eff_date = 12/31/9999



Logical Table Source - WMSC_DIM_CURR_WMS_CASE

Aggregation content, group by: Logical Level

☑ Show mapped ☑ Show unmapped

Logical Dimension	Logical Level	
Geography		☒

Fragmentation content:

□ This source should be combined with other sources at this level

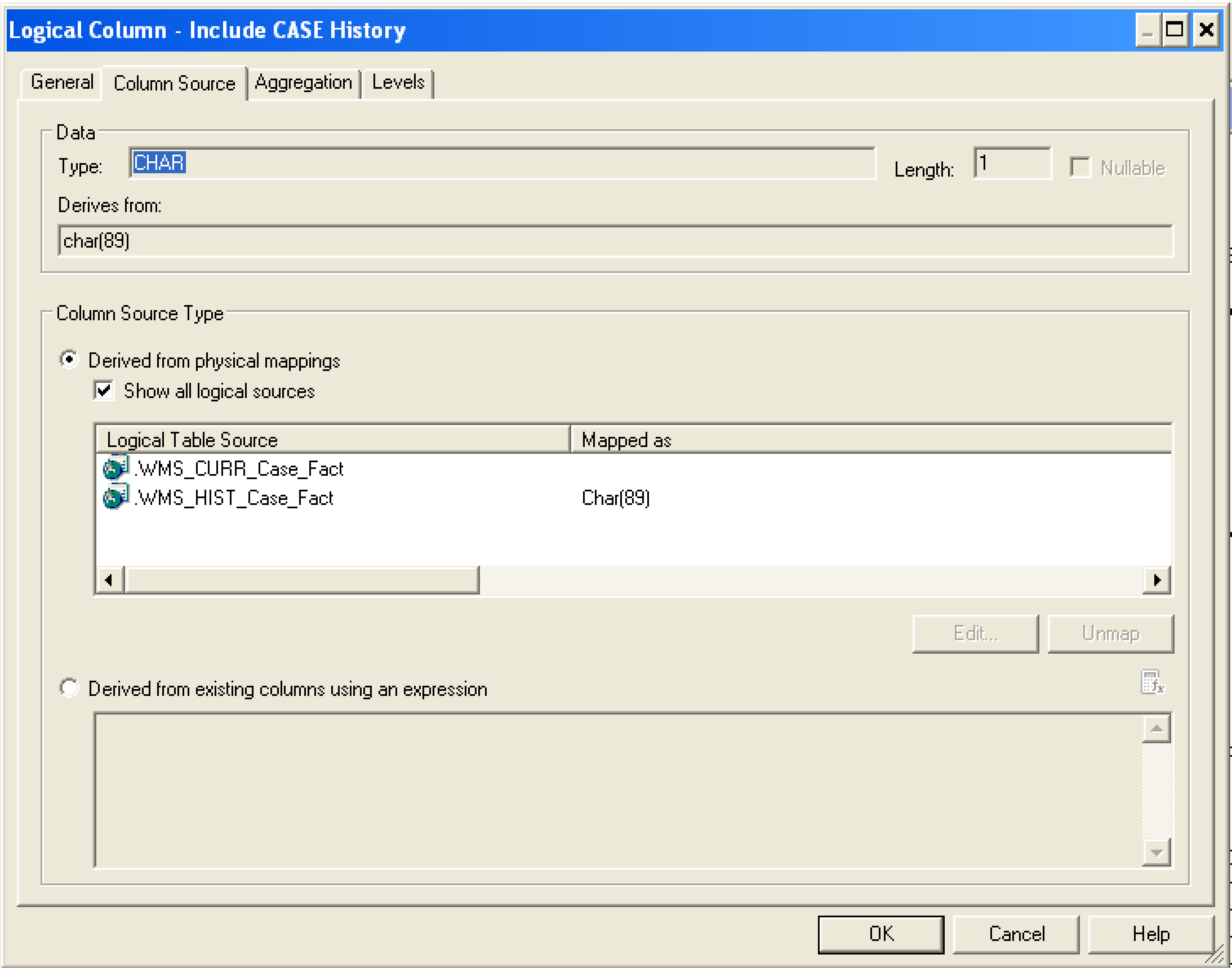
Use this "WHERE clause" filter to limit rows returned (exclude the "WHERE"):

```
"RW_EWDV".""."DATASMART"."WMSC_DIM_CURR_WMS_CASE"."END_EFF_DATE" = DATE '9999-12-31'
```

□ Select distinct values

Default Most Recent, Step 5: Map "History" Attribute only to the History LTS

- Create a new logical column.
- On the Column Source tab, map it to the **history** LTS, but not the current.
- We use a constant, Char(89) or Y.



Logical Column - Include CASE History

Data

Type: CHAR

Length: 1

□ Nullable

Derives from:

char(89)

Column Source Type

● Derived from physical mappings

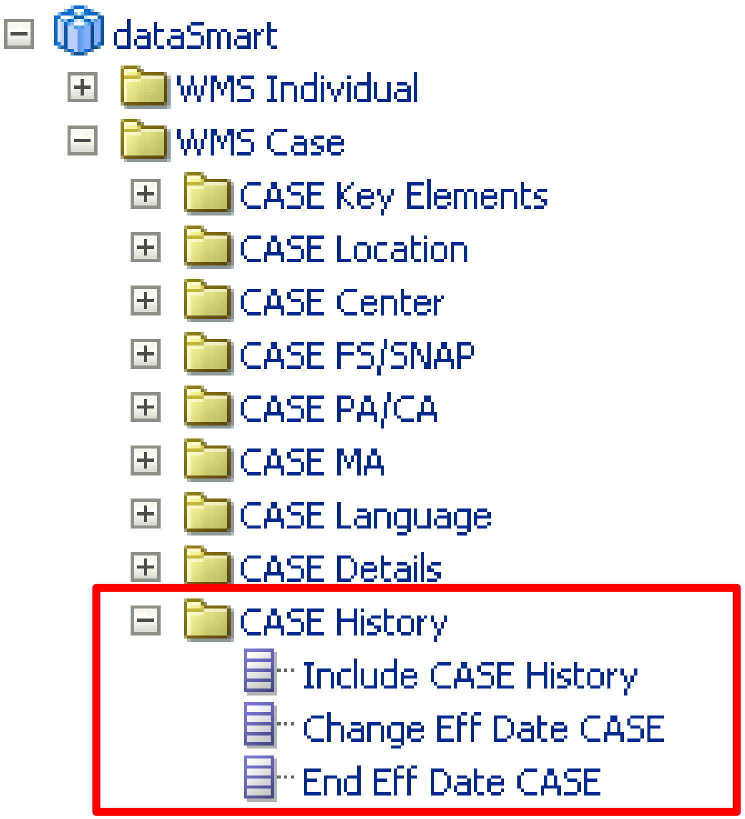
☑ Show all logical sources

Logical Table Source	Mapped as
.WMS_CURR_Case_Fact	
.WMS_HIST_Case_Fact	Char(89)

○ Derived from existing columns using an expression

“Maintain existing flexibility in date logic, while improving usability.”

- Most recent is now the default filter.
- To access historical records, add “Include History” from the History subfolder.
- Consider all parent-folders (separate SCD2s) in your analysis.
  - Discoverer had Most Recent in many folders.
  - OBIEE has History in many folders.
  - Include it for all folders where it’s needed!
- Don’t forget your filters!
  - Don’t write the zipper yourself.
  - Try historical filters saved in “OBIEE Tools”.



Lying to OBIEE, Part 1:
BMM Keys and Hierarchies

- History Logical Table Sources are *more* detailed than the key for their logical tables.
  - So, they do not have their own level in a hierarchy, AND
  - You cannot assign levels to the LTS’s for these dims.
- This was a choice because drilling into history in this way doesn’t make sense for our users.

Lying to OBIEE, Part 2:
Treating Factless Facts as Dims in the BMM

- We have attribute-only “facts” in NYCWAY database tables.
- Our users tend to focus on attribute only queries.
- To allow users to combine attributes from different NYCWAY “Fact” tables in a single query, we treat them as dims in the BMM.
- Lies involved:
  - Many to 1 joins are actually 1 to Many
  - Inadequate primary keys allow for foreign key joins
  - *Only works with count distinct measures*.

Any Question ?