



KIMBALL GROUP  
Consulting | Kimball University

# DIMENSIONAL MODELS: MANAGING CHANGE

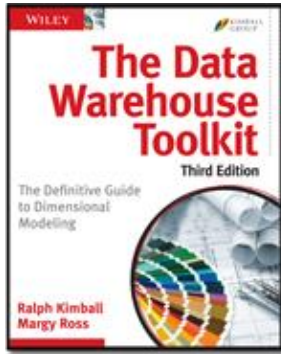
Joy Mundy, [joy@kimballgroup.com](mailto:joy@kimballgroup.com)

# Introductions

# About Joy and Kimball Group

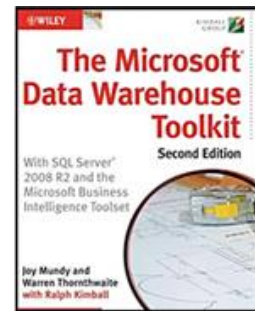
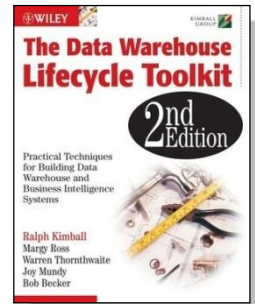
- Kimball Group founded by Ralph Kimball
- Tiny DW/BI consultancy
  - Consulting (requirements, design, and architecture)
  - Writing
  - Teaching and speaking
- Kimball Group retires at end of 2015 (tick tock)
  - Content will remain available
  - Website plus upcoming Kimball Reader, Second Edition
- Joy
  - Biz user background
  - A ridiculously long time in DW / BI (25 years now)
  - Consultant, stint on SQL Server product team, has done some actual work too

# Acknowledgments



## ➤ Course materials adapted from...

- **The Data Warehouse Toolkit, 3<sup>rd</sup> Ed.**
  - R. Kimball, M. Ross (Wiley 2012)
- **The Data Warehouse Lifecycle Toolkit, 2<sup>nd</sup> Ed.**
  - R. Kimball, M. Ross, W. Thornthwaite, J. Mundy, B. Becker (Wiley 2008)
- **The Microsoft Data Warehouse Toolkit, 2<sup>nd</sup> Edition**
  - J. Mundy, W. Thornthwaite (Wiley 2011)
- **Kimball University**
  - Data Warehouse Lifecycle in Depth course materials
  - Design Tips and Intelligent Enterprise articles at [www.KimballGroup.com](http://www.KimballGroup.com)



# Agenda

- Very quick introduction to dimensional modeling
  - What and why
  - The key characteristics of a dimensional model
- The Kimball Bus Matrix
- Methods of managing dimension attribute changes
  - What and why
  - Advice on choosing one method over another

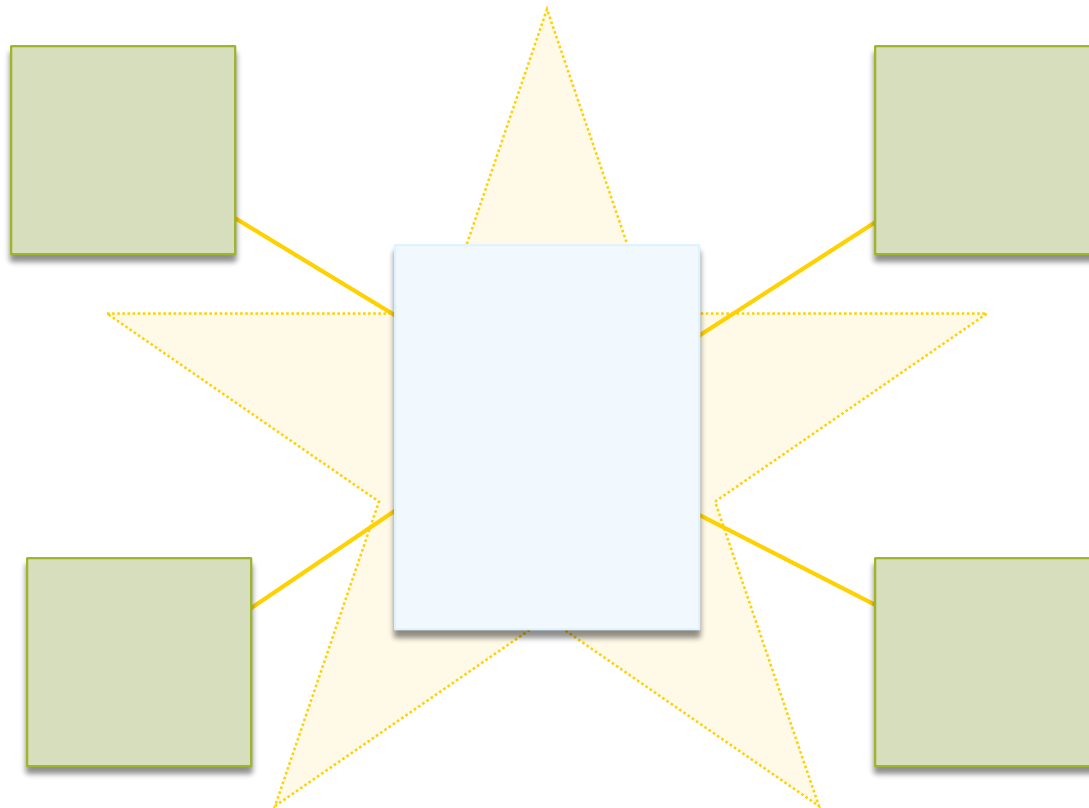
# Basic Dimensional Modeling Concepts

# Kimball Method design precepts

- Design for the enterprise
  - The greatest value comes from combining information from subject areas that don't usually get analyzed together
- Design for flexibility
  - Very detailed data
  - Hook together via Conformed Dimensions
- Design to enable ad hoc use
  - Even if you don't offer widespread ad hoc access on Day 1

# What is a dimensional model (star schema)

- Single fact table of measurements, surrounded by multiple descriptive dimension tables





# Dimensional: Why and how

- Primary design goal: Support analytic queries
  - Usable
  - Query performance
- Key terms
  - Facts = measures of business events
  - Dimensions = entities that participate in business events
- Basic approach:
  - Denormalize dimensions for usability
  - Normalize facts for performance

# Terminology: Dimensions

- Characteristics of a subject/object
  - Who, what, when, where, why, how
  - Product, Date, Patient, Facility ...
- Each row is an occurrence
  - One row per product, day, patient, ...
- Dimension attributes (columns):
  - Report labels and query constraints
  - “By” words and “where” clauses
  - Verbose descriptive attributes, in addition to codes
  - Hierarchical relationships

## Product Dimension

### PRODUCT KEY

Product Desc.  
SKU #  
Size  
Brand Desc.  
Class Desc.

# Terminology: Facts

- Metrics resulting from business process or event
  - NOT mapped to a specific report
  - Facts are usually numeric and additive
- Granularity/grain
  - Identifies the level of detail
  - One row per sale, one row per bank account, one row per claim, ...
  - Atomic grain is most flexible
- Three main fact table types
  - Transaction; Snapshot; Accumulating

## Sales Facts

DATE KEY  
PRODUCT KEY  
STORE KEY  
PROMOTION KEY  
Other dim keys...

*Sales Amount*  
*Sales Units*

...

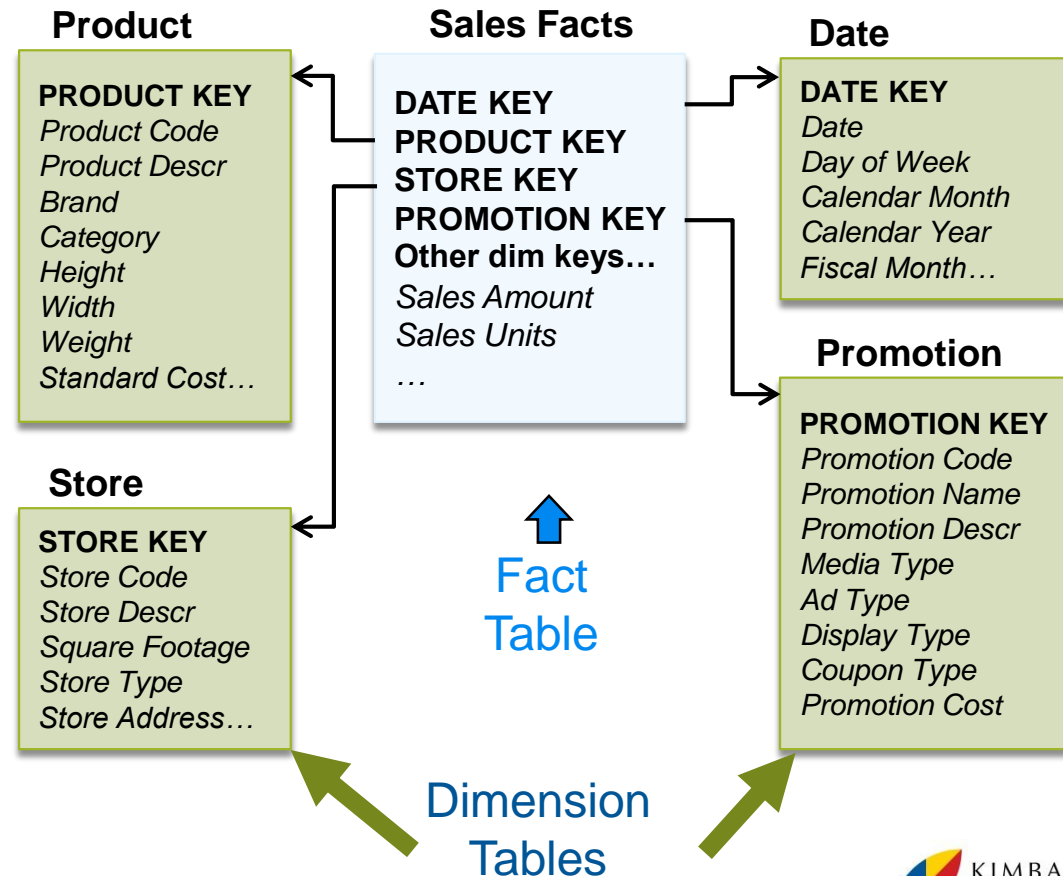
# Terminology: Dimensional model

(or Star Schema)

➤ Fact table per business process / event, plus relevant dimensions

➤ Benefits:

- Easier to understand
- Better performance
  - Pre-joined
  - Star join optimization
- Extensible to handle change



# Terminology: Dimension Table Surrogate Keys

- Surrogate keys are substitute keys
  - Integer, non-meaningful, sequence numbers
  - Surrogate keys join fact and dimension tables
  - Treat business keys as attributes (aka natural keys)
- Benefits
  - Isolate DW/BI system from operational changes
  - Improve performance (over character and 2-col keys)
  - Handle “Not applicable”, “Date TBD”, ...
  - Allow integration of multiple sources
  - *Enable tracking of dimension attribute changes*

# Multiple Hierarchies in Dimensions

- Dimension tables can have multiple hierarchical roll-ups
  - Date dim often has 3 (calendar, fiscal, week); sometimes more
  - Alternative product rollups are very common
  - Multiple organizational hierarchies are also common
- For final consumption by OLAP technologies, it's important to build clean hierarchies (when the data supports it)

# Avoid the “Too Many Dimensions” Trap... and the BFT Trap



## Centipede

**Date Key**  
**Month Key**  
**Year Key**  
**Product Key**  
**Product Brand Key**  
**Store Key**  
**Store State Key**  
**Store District Key**  
**Promotion Key**  
**Promotion Media Key**  
**POS Trxn #**

Sales Qty  
Sales Amt



## BFT=Big Fat Table

POS Trxn #  
Day  
Month  
Year  
Product SKU  
Product Name  
Product Descr  
Product Brand  
Product Brand Mgr  
Store  
Store Name  
Store Address  
Store Manager  
Store State  
Store District  
Promotion ID  
Promo Name  
Promo Media  
Blah blah blah

Sales Qty  
Sales Amt

## Preferred Design

**Date Key**  
**Store Key**  
**Promotion Key**  
**POS Trxn #**

Sales Qty  
Sales Amt



# The Kimball Bus Matrix



# Building an enterprise system

- Build a “star” to support a specific business process analytic need
  - Fact tables at the finest grain (most detail) that it is possible to get
  - Rich, re-usable dimensions
  - Usually a set of 3-5 related fact tables and associated dims
- Many reports and analyses will need to combine data from multiple fact tables
  - Hook together along shared (**conformed**) dimensions
- Before detailed design begins, sketch out the complete system

# The Kimball Design Process

1. Interview across the enterprise to understand the overall business and where to start
2. Build preliminary Bus Matrix that maps out what we think the complete design will be
3. Identify Phase 1 scope
4. Build an accurate Kimball Bus Matrix that maps out what we think the complete design will be
5. Design core shared dimensions (in agonizing detail)
6. Design fact tables and single-use dimensions
7. Review pass over complete model
8. Write the design document

# Preliminary Bus Matrix

“CORE” Processes = Phase 1

Key Biz Processes \ Conformed Dimensions	Date/Time	Member/Prospect	Provider	Diagnosis	Procedure	Service	Drug	Location	County	Line of Business	Source	Program	Event	Reason/Result	Employee	..	..
Membership (Elig/Enroll)	XX	X							X	X	X			X			
Paid Claims: Hospital	XX	X	XX	X	X	X	X	X	X	X					X		
Paid Claims: Services	XX	X	XX	X	X	X		X	X	X					X		
Paid Claims: Pharmacy	XX	X	XX				X	X	X	X					X		
Inferred Diagnoses or Member Profile	X	X		X					X	X							
Hospital and Prior Authorizations	XX	X	XX	X	X	X		X	X	X							
Hospital Census	XX	X	X	X				X	X								

# Preliminary Bus Matrix – Cont

## Out of scope for P1

Conformed Dimensions Key Biz Processes	Date/Time	Member/Prospect	Provider	Diagnosis	Procedure	Service	Drug	Location	County	Line of Business	Source	Program	Event	Reason/Result	Employee	...
Care Based Incentives	XX	X	XX					X	X	X					X	
Billing	XX		X		X	X	X		X	X						
Prescriptions Written	X	X	X				X	X	X	X						
Lab Results	XX	X	X		X			X	X	X						
Pending Claims	XX	X	X	X	X	X	?	X	X	X				X		
Care Management Enrollment	XX	X	X	?				X	X	X	X		X			
Care Management Activity	XX	X	X	?				X	X	X	X	X		X		
Grievances/Appeals	XX	X	X	?				X	X	X			X	X		
Call Details (Phone Switch)	X	X	X						X	X			X	X		

# Preliminary Bus Matrix – Cont

## *Really out of scope for P1*

Conformed Dimensions Key Biz Processes	Date/Time	Member/Prospect	Provider	Diagnosis	Procedure	Service	Drug	Location	County	Line of Business	Source	Program	Event	Reason/Result	Employee	..
Provider Contracting																
Provider Credentialing																
Market Statistics																
Census Statistics																
Member/Provider Surveys																
Human Resources																

# Example High Level Bus Matrix: Adventure Works Cycles

Adventure Works  Data Warehouse Bus Matrix	<-- Conformed Dimensions -->																
	Business Priority	Date(Order, Start, Ship)	Product	Promotion	End Customer	Employee	Reseller	Page	Internet Registered User	Part	Vendor	Shipper	Problem	Account	Department	Currency (Source, Dest.)	Benefits Plan
Business Process																	
Advertising																	
TV		X	x	x													
Print		X	x	x													
Online		X	x	x	x												
Promotions		X	x	x	x		x										
Co-op programs		X	x	x		x	x										
Web Site Marketing		X	x	x	x			x	x								
PR		x	x	x													
Orders																	
Reseller Orders	1	x	x	x		x	x										
Internet Orders	1	x	x	x	x			x	x								
Orders Forecasting	2	x	x	x		x	x										
Purchasing		x	x		x	x				x	x	x					
Parts Inventory		x	x	x						x	x						
Manufacturing	6	x	x							x							
Finished Goods Inv.		x	x	x													
Shipping		x	x	x	x	x	x					x					
Returns	5	x	x		x	x	x					x					
Registration cards		x	x		x												
Customer Support																	
Customer Calls	4	x	x	x	x	x	x			x			x				
Web Support	4	x	x		x	x	x	x	x				x				
Financial Forecasting		x	x	x	x	x	x			x				x	x		
Exchange Rate Mgmt.	3	x															x
GL-Revenue & Exp.		x												x	x		
Cost Accounting		x	x											x	x		
Payroll		x				x										x	
Benefits Enrollment		x				x											x

## Opportunities

- Sales performance
- Sales strategy
- Sales targeting
- Product recommendations
- Manufacturing planning
- Sales forecast input
- Customer prioritization

# Managing Changes in Dimension Attributes

## Slowly Changing Dimensions

# Dimension Attribute Changes

- Attributes change over time
  - Customers move, change their names, adjust their incomes
  - Organization hierarchies are restructured
- “SCD,” or Slowly Changing **Dimension** is a misnomer
  - Decision made on an attribute-by-attribute basis
  - Business users tell us how to handle change
- Three original SCD types (1, 2, 3)
- Updated in 2011 to eight SCD types (0..7)



# Type 0: Retain Original

- Dimension attribute value *never* changes
- Uncommon
- Use cases:
  - Date dimension attributes
  - Original attributes
    - Eg “Original diagnosis”
    - Usually there is a second attribute which is updated in one of the other methods (eg “Current diagnosis”)

# Type 1: Restate history by updating in place

## Original row

Cust Key	Cust Acct Num	Cust City	Cust State	Cust Region
54321	RXY-2500	Menlo Park	CA	West

Customer moves to New Mexico...

## Updated row

Cust Key	Cust Acct Num	Cust City	Cust State	Cust Region
54321	RXY-2500	Santa Fe	NM	West

Pluses	Minuses
Simple, easy to implement	History is lost
Easy to understand	No ability to report "as was"
	Increased complexity if we have performance aggregations

# Type 2: Track history by adding a new row

## Original row

Cust Key	Cust Acct Num	Cust City	Cust State	Cust Region
54321	RXY-2500	Menlo Park	CA	West

Customer moves to New Mexico...

## Updated row

Cust Key	Cust Acct Num	Cust City	Cust State	Cust Region
54321	RXY-2500	Menlo Park	CA	West
69847	RXY-2500	Santa Fe	NM	West

# Type 2 Attribute change management

Pluses	Minuses
Elegant method to track history	A little harder to implement in a simple world
Joins to the fact table remain simple “equi-join”	Late arriving dimension attribute changes can be a significant burden with large data volumes (requires updating facts)
Easily consumed by cube technologies, with no performance aggregation maintenance problems	Dimension tables will grow
	User education and documentation

- Requires surrogate keys!
- Add row metadata
  - Row start/stop date, is current indicator

# Type 3: Track history by adding a new column

## Cust dimension original

Cust Key	Cust Acct Num	Cust City	Cust State	Cust Region
54321	RXY-2500	Menlo Park	CA	West

We redesign our sales region taxonomy...

## Cust Dim Type3 (after we add a column to dim table)

Cust Key	Cust Acct Num	Cust City	Cust State	Cust Region	Old Cust Region
54321	RXY-2500	Menlo Park	CA	Southwest	West

Pluses	Minuses
Easy to understand	Requires a structural change
Allows tracking of old and new	Inflexible and awkward
Use restricted primarily to org structure changes	No trending over time

# Type 4: Mini-dimension

- Pull out attributes from a large dimension into a mini-dimension
  - Main dimension, such as Customer, is very large (tens or hundreds of millions of rows)
  - Some attributes change “quickly”
  - And/or some attributes are commonly constrained on
- Examples:
  - Customer Demographics
  - Claimant Characteristics
- Relationship to dimension is managed in ETL
- For users, history is tracked via the fact table

# Type 4 Mini-Dimension Example

Cust Dim Original
Customer Key (pk)
Customer Acct Num
Customer Name
Customer State
Customer Country
...
Customer Age in Months
Customer Income Amt
Customer Lifetime Score



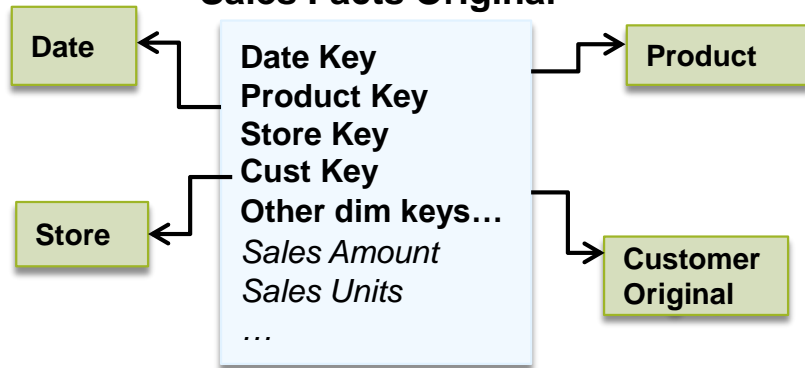
Cust Dim Type4
Customer Key (pk)
Customer Acct Num
Customer Name
Customer State
Customer Country
...



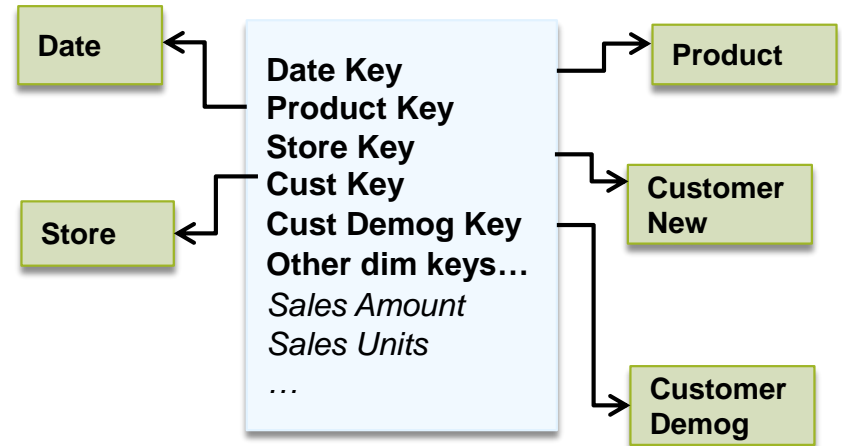
Cust Demog Dim Type4
Customer Demog Key (pk)
Customer Age Band
Customer Income Band
Customer Lifetime Band

# Type 4 In Context

**Sales Facts Original**



**Sales Facts With Mini-Dim**





# Types 5-7: Having it both ways

- Sometimes we have valid user communities and reasons for managing attribute change both ways:
  - Restate history (current view)
  - Track history (report as-was)
- Consider a Salesperson who moves from one district to another
  - New boss needs to see all employee's sales history
  - But for district-level sales forecasting, want to see sales history in the district it occurred

# Type 5: A Subtle Twist on Type 4

- Put CURRENT mini-dimension attributes in the main dimension table

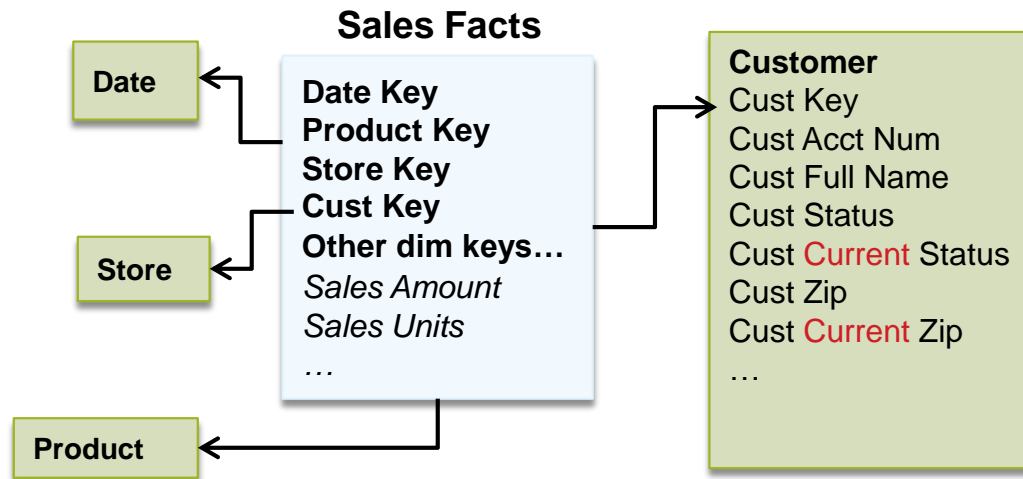
Cust Demog Dim Type4-5
Customer Demog Key (pk)
Customer Age Band
Customer Income Band
Customer Lifetime Value Band

Cust Dim Type5
Customer Key (pk)
Customer Acct Num
Customer Name
Customer State
Customer Country
...
Current Cust Age Band
Current Cust Income Band
Current Cust Lifetime Value Band

- Or possibly snowflake to CURRENT view of mini-dimension

# Type 6: Add “current” attributes to the dimension

- Dimension contains a mix of attributes of type 1 & 2
- “Both ways” attributes show up twice in the same dimension



# Type 6: Sample rows

## Original row

Cust Key	Cust Acct Num	Cust Status	Cust <b>Current</b> Status	Current Ind
54321	RXY-2500	Active Subscriber	Active Subscriber	Current

## First change (customer inactive because of non-payment):

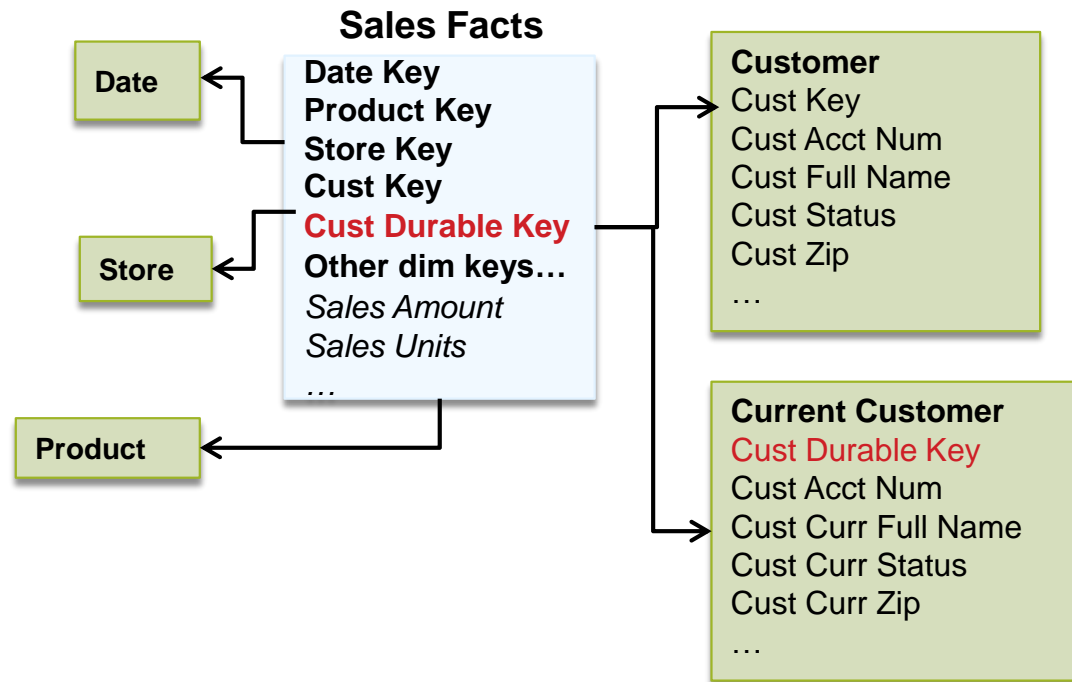
Cust Key	Cust Acct Num	Cust Status	Cust <b>Current</b> Status	Current Ind
54321	RXY-2500	Active Subscriber	Non-payment	Expired
69847	RXY-2500	Non-payment	Non-payment	Current

## 2<sup>nd</sup> change (customer churned out):

Cust Key	Cust Acct Num	Cust Status	Cust <b>Current</b> Status	Current Ind
54321	RXY-2500	Active Subscriber	Not a customer	Expired
69847	RXY-2500	Non-payment	Not a customer	Expired
88623	RXY-2500	Not a customer	Not a customer	Current

# Type 7: Maintain two dimensions

- The main dimension is managed, wholly or partially, as Type 2 (track history)
- A second dimension is managed as Type 1



# Type 7: Sample rows

## Type 2 dimension:





Cust Key	Cust Durable Key	Cust Acct Num	Cust Status	Current Ind
54321	54321	RXY-2500	Active Subscriber	Expired
69847	54321	RXY-2500	Non-payment	Expired
88623	54321	RXY-2500	Not a customer	Current

## Type 1 dimension (database view):

Cust Durable Key	Cust Acct Num	Cust Status
54321	RXY-2500	Not a customer

**Fact table needs  
*both* Cust Key  
and Cust  
Durable Key**

# Deciding between types 6 & 7

Characteristic	Type 6	Type 7
Fact table size		
Dimension table width		
Flexibility in the absence of good business requirements		

➤ I prefer Type 6

➤ Type 7

- If there are a LOT of attributes we need both ways
- There is a lot of uncertainty from business users about which attributes need to be Type 1 / 2
- No business users involved in the design at all. Boo.

# Summary: Slowly Changing Dimension Attributes

- Always an attribute-by-attribute decision
- Capture the decision at design time
  - As important as the attribute name and description
- Type 1: Restate history by updating in place
- Type 2: Track history by adding a new row to the dimension
- Type 3: Add a new set of attributes upon change (seldom used)
- Types 4&5: Mini dimension
- Type 6: Dimension table has more columns—one for CURRENT value and one for HISTORIC value of each attribute
- Type 7: Two dimension “tables”, one CURRENT and one HISTORIC.



# Slowly Changing Dimension Summary

- Decision is attribute-by-attribute
- Decision should be made by the business
- Conformed dimensions requires agreement across the business about Type 1 v. Type 2
  - Which may push us to Type 7
- Types 4 & 5 (mini-dimensions) are a really good trick especially for extremely large dimensions



KIMBALL GROUP  
Consulting | Kimball University

THANK YOU

