

Dimensional Modeling Overview



Bob Becker
Wisconsin DAMA
March 7, 2005



KIMBALL
UNIVERSITY

© 2004 Kimball University All rights reserved.

Contact information:



KIMBALL
UNIVERSITY

www.kimballgroup.com

Kimball Group
14898 Boulder Pointe Road
Eden Prairie, MN 55347

Bob Becker
bob@kimballgroup.com
952.974.0434

Course Objectives



- ☐ Understand basic dimensional modeling techniques
 - *Architectural fit*
 - *Concepts*
 - *Approaches to overcome common design challenges*
- ☐ Introductory versus advanced
- ☐ Provide hands-on design opportunity.



- ☐ Our goal is that you walk away from this class armed with a process and some basic techniques to begin to model your data dimensionally.

Agenda



- ☐ Introductions
- ☐ Dimensional Modeling Fundamentals
- ☐ 'Basics' Case Study
- ☐ 'Beyond 1st Data Mart' Case Study
- ☐ More Dimensional Fundamentals
- ☐ 'Transaction Detail' Case Study
- ☐ Design Workshop 'Customer-Centric'

© 2004 Kimball University. All rights reserved.



- | | |
|--|---------|
| <input type="checkbox"/> Introductions and Fundamentals | Page 1 |
| <input type="checkbox"/> 'Basics' Case Study | Page 16 |
| <ul style="list-style-type: none">• 4-step process• Degenerate dimensions• Surrogate keys• Factless fact tables | |
| <input type="checkbox"/> 'Beyond the 1st Data Mart' Case Study | Page 29 |
| <ul style="list-style-type: none">• Star vs. snowflake variations• Data warehouse bus matrix• Conformed dimensions | |
| <input type="checkbox"/> More Dimensional Fundamentals | Page 40 |
| <ul style="list-style-type: none">• Slowly changing dimensions• Dimension and fact table indexing | |
| <input type="checkbox"/> 'Transaction Detail' Case Study | Page 47 |
| <ul style="list-style-type: none">• Dimension table "role playing"• Invoicing complications | |
| <input type="checkbox"/> Design Workshop | Page 55 |
| <input type="checkbox"/> Completed Worksheets | Page 61 |

Kimball Group / Instructor Introduction

- ☐ Based in Minneapolis, Minnesota
- ☐ Focus exclusively on data warehousing and decision support
- ☐ Designed and/or developed over 100 data marts and warehouses
- ☐ Customers include...

Ameritech

Discover Card

General Mills

General Motors

Kimberly-Clark

Land O'Lakes

MGM Grand

Pacific Bell

Target Stores

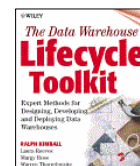
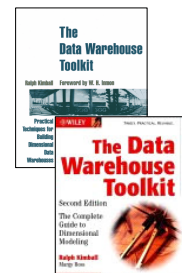
United Airlines

© 2004 Kimball University. All rights reserved.



Acknowledgments

- ☐ Materials adapted from...
 - *The Data Warehouse Toolkit, 2nd Edition*
→ *R. Kimball, M. Ross (Wiley 2002)*
 - *DBMS/Intelligent Enterprise articles*
→ www.ralphkimball.com
 - *The Data Warehouse Lifecycle Toolkit*
→ *R. Kimball, L. Reeves, M. Ross, W. Thornthwaite (Wiley 1998)*



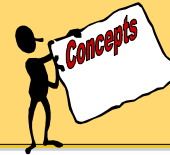
© 2004 Kimball University. All rights reserved.



Dimensional Modeling Fundamentals

© 2004 Kimball University. All rights reserved.

Dimensional Modeling Fundamentals Concepts

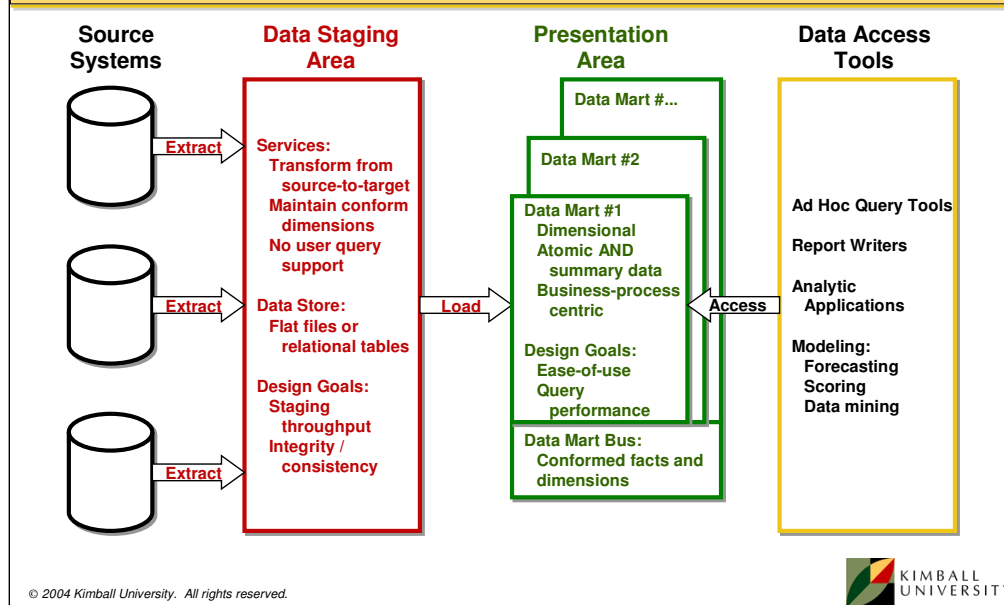


- ☐ Data warehouse “chess pieces” and modeling assumptions
- ☐ Differences between facts and dimensions
- ☐ Rationale for designing dimensional models

© 2004 Kimball University. All rights reserved.



Simplified Elements of Data Warehouse



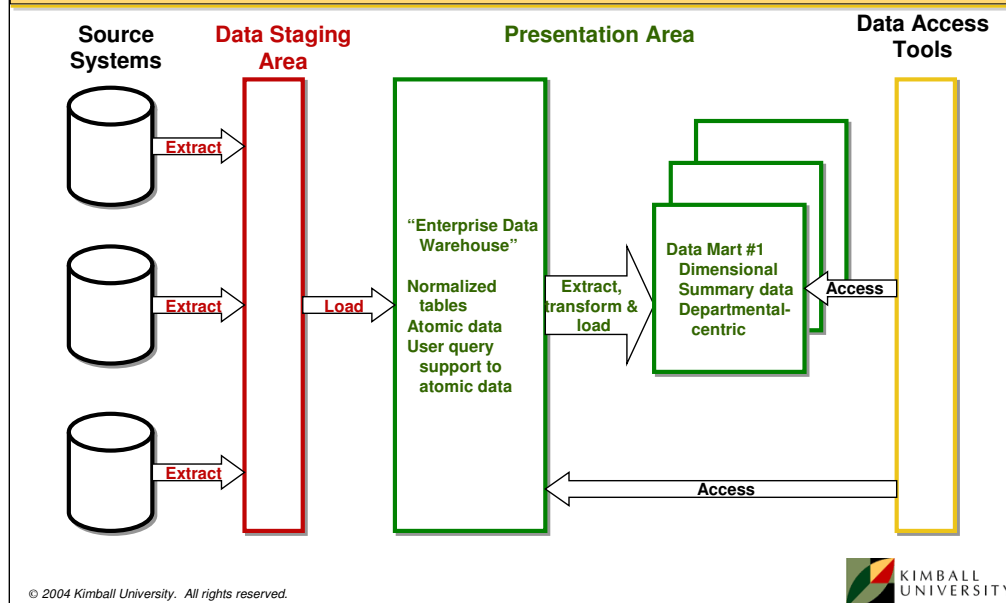
□ Data Staging Area:

- Portion of the data warehouse restricted to EXTRACTING, CLEANING, and LOADING data from legacy systems into destination data marts.
- The data staging area is the “back room” and is explicitly off limits to the end users, akin to the kitchen area of a restaurant.
- The data staging area DOES NOT SUPPORT QUERY OR PRESENTATION SERVICES.
- The data staging area is dominated by sorting and sequential processing. The pre-cleaned data is almost always in a flat file format, and the post cleaned data is either in a flat file format or third normal form.

□ Presentation Area:

- Portion of the data warehouse restricted to PRESENTING data, not cleansing or transforming data.
- Organized into data marts, focused on a specific BUSINESS PROCESS (not business function, business function or business unit) subject area.
- A data mart often contains ATOMIC DATA, as well as more summarized data.

Simplified Alternative Viewpoint



- ❑ There are a number of similarities between the two approaches:
 - Integrated data with common, conformed dimensions.
 - Dimensional data marts.
- ❑ Unique characteristics of this approach include the following:
 - Mandatory, persistent normalized data structures result from staging process. Some ETL occurs to load the “data warehouse,” and more ETL occurs to load the marts.
 - Dimensional structures are for summary data only.
 - Users are given limited access to the detailed data in ER structures.

Course Assumptions: Data Warehousing Differences

- ❑ Differences between transaction processing and data warehousing understood

Transaction Processing

Represent current state

Predictable usage

*Optimized to get
data "in"*

Data Warehousing

Preserve history

Highly unpredictable

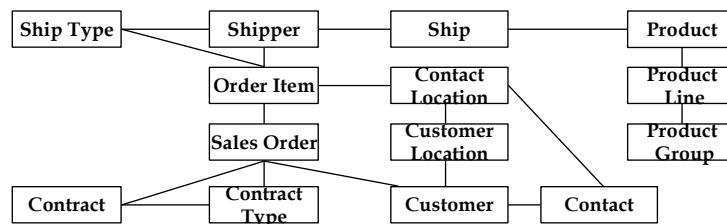
*Optimized to get
data "out"*

© 2004 Kimball University. All rights reserved.



Course Assumptions: Different Designs for Different Problems

- ❑ Transaction processing needs normalized data



- ❑ Data warehousing needs different approach. . .

- *Easy to use*
 - *Simple, understandable and memorable to business*
- *Optimized for query performance*

© 2004 Kimball University. All rights reserved.



Course Assumptions: Expectation Management

- ❑ Discussing dimensional designs in a relational environment (“star schemas”)
 - *Familiar with relational concepts / terminology*
- ❑ Introduction to dimensional modeling
 - *Start with basics and build on foundation*
 - *Address issues relevant to broad audience*

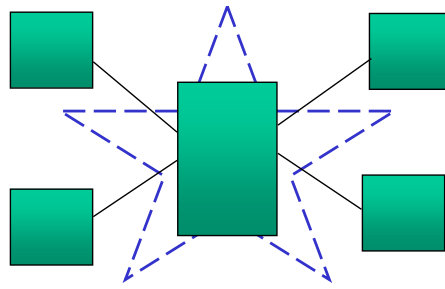
© 2004 Kimball University. All rights reserved.



- ❑ Although we'll be focused on designing dimensional models for a relational database, much of what we'll be discussing is relevant to multidimensional databases, too.

Dimensional Model / Star Schema -- What is it?

- ❑ Single data (fact) table surrounded by multiple descriptive (dimension) tables



© 2004 Kimball University. All rights reserved.



- ❑ The star schema is a dimensional design for relational databases. Dimensional models implemented in multidimensional OLAP databases (such as Hyperion's Essbase) are referred to as data cubes.
- ❑ A star schema is characterized by a single data table or fact table, surrounded by descriptive tables or dimension tables. The data warehouse will ultimately consist of many star schemas.
- ❑ The retail industry, the first to adopt this design technique, typically had four or five dimensions, hence the term "star" was coined.

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 KEY

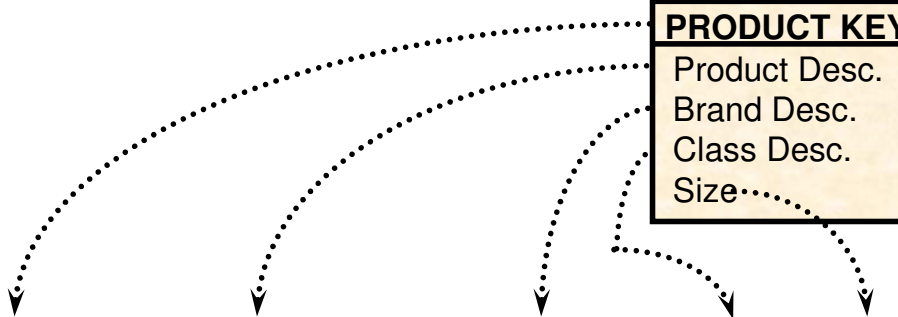
Product Desc.
SKU #
Size
Brand Desc.
Class Desc.

© 2004 Kimball University. All rights reserved.



- ☐ Dimensions or their attributes are often the "by" words in a query or report request. For example, a user wants to see sales by month by product. The natural ways end users describe their business should be included as dimensions or dimension attributes.
- ☐ Albert Einstein captured the main reason we use the dimensional model when he said, "Make everything as simple as possible, but not simpler."
- ☐ Date is the fundamental business dimension across all industries, although many times a date table doesn't exist in the operational environment. Analyses that trend across dates or make comparisons between periods (nearly all business analyses) are best supported by creating and maintaining a robust Date dimension table.
- ☐ Combining all the attributes of a business object, including its hierarchies, into a single dimension table is called denormalization. This simplifies the model from a user perspective. It also makes the join paths much simpler for the DBMS optimizer than a fully normalized model. But it still presents the same information and relationships found in the normalized model – nothing is lost except complexity.
- ☐ Dimensions are not usually time dependent but changes over time do occur. We'll discuss ways to handle changes over time later in the class.

Product Dimension Table Sample Rows



Product Key	Product Desc	Brand Desc	Class	Size
0001	CHEERIOS 10 OZ	CHEERIOS	FAMILY	10 OZ
0002	CHEERIOS 24 OZ	CHEERIOS	FAMILY	24 OZ
0003	LUCKY CHARMS 10 OZ	LUCKY CHARMS	FAMILY	10 OZ

© 2004 Kimball University. All rights reserved.



- ☐ Several data rows from the Product Dimension table are represented to illustrate the redundant values stored in the Brand and Class description columns.
- ☐ Notice that we did not merely store a Brand Code and join to a Brand Description table. A single descriptive table is preferable rather than forcing users to join across multiple lookup tables. Simplified, denormalized dimension tables improve ease of use, as well as reduce the number of joins and thereby improve performance.

Terminology: Facts

❑ Metrics resulting from business process or event

- *Facts are usually numeric and additive*

❑ Granularity/grain

- *Identifies the level of detail*
- *One row per sale, one row per service call, one row per claim, ...*
- *Atomic grain is most flexible*

DATE KEY
PRODUCT KEY
STORE KEY
PROMOTION KEY

\$ Sales
Unit Sales

© 2004 Kimball University. All rights reserved.



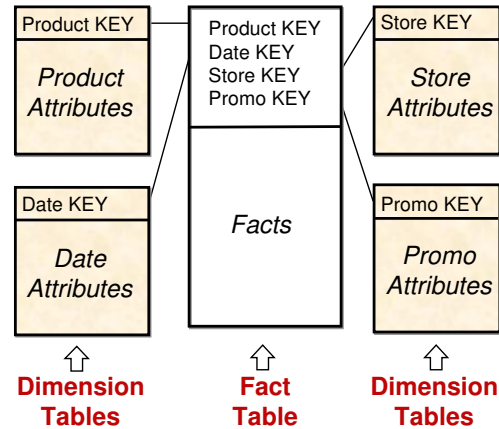
- ❑ Dimensions are business objects, while facts are business measurements. A business object, like a product, can exist without ever being involved in a business event. We might make a product we never sell. However, we cannot have fact (or “event”) without its associated dimensions.
- ❑ Each dimension that participates in a fact table defines part of the fact table key. That is, the key to the fact table is a multi-part key defined by foreign key relationships with each dimension table involved.
- ❑ Most facts are numeric, although not all numeric data are facts. Exceptions would be descriptive numerics like package size or weight (describes an product) or customer age (describes a customer). Facts should be “continuously valued” or rapidly changing with each measurement occurrence.
- ❑ Facts are typically additive (such as dollar sales or unit sales). Additivity is important because data warehouse applications seldom retrieve a single fact table record. They bring back hundreds or thousands of records and the most useful thing is to add them up. Other facts are semi-additive (such as share) and still others are simply non-additive (price).
- ❑ In general, we like to build our fact tables with the lowest level of detail possible that makes sense from a business point of view – the “atomic” level. This gives us complete flexibility to roll up the data to any level of summary needed now or in the future.
- ❑ Fact tables are very efficient. They store little to no redundant data. They are also the largest tables, usually making up 90% or more of the total database size.

Terminology: Dimensional Model or Star Schema

- ❑ Fact table per business process / event, plus relevant dimensions

- ❑ **Benefits:**

- *Easier to understand*
- *Better performance from fewer joins*
- *Extensible to handle change*



© 2004 Kimball University. All rights reserved.



- ❑ The star schema is the dimensional model based on a single fact table and its associated dimensions.
- ❑ We often have families of star schemas to describe a set of related business processes, like a company's value chain. The rows in the business dimensional matrix define the members of the star schema family.
- ❑ The idea of re-using dimensions across multiple business processes is the root of the Enterprise Data Warehouse Bus Matrix concept.

Sample Report Translation of Dimensional Model

Sales Rep Performance Report Central Region		
	Jan 2003	Feb 2003
	Dollars	Dollars
Chicago District		
Adams	990	999
Brown	900	999
Frederickson	990	999
Minneapolis District		
Andersen	950	999
Smith	950	999
Central Region Total	4,780	4,995



“Dimensions”

Report, row and column headings



“Facts”

Numeric report values

© 2004 Kimball University. All rights reserved.



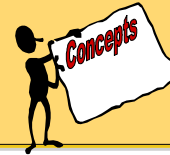
- ❑ Another way to think about star schemas or dimensional data models is to see them translated into a report.
 - Dimension tables supply the report headings and row or super column headings. They're designated by the light gray shading on this slide. Dimensions are the elements that users want to slice-and-dice on. When users say they want to look at performance metrics by ___ by ___ by ___, business dimensions fill in the blanks. In this report, we're looking at sales by region by district by rep and month.
 - Facts are the numbers that make up the meat of the report. In this case, they're the dollar sales figures.
- ❑ Although reviewing reports can help identify fact and dimension table candidates, a dimensional data model should NEVER be designed by simply reviewing existing or requested reports.

Basics Case Study

Retail Store Sales Schema

© 2004 Kimball University. All rights reserved.

Section Concepts



- ☐ Steps to designing a dimensional data model
- ☐ “Degenerate” dimensions
- ☐ Multiple hierarchies in single dimension table
- ☐ Surrogate keys
- ☐ Star vs. snowflake variations
- ☐ Factless fact tables

© 2004 Kimball University. All rights reserved.



Getting Started - Four Key Steps:

1. Identify the Business Process

From the Business Process you should be able to:

2. Identify the Grain

3. Identify the Dimensions

4. Identify the Facts

© 2004 Kimball University. All rights reserved.



☐ Business Process:

- Each organization has a series of business processes (such as raw materials purchasing and order processing for a manufacturer or transaction processing and cycle billing for a credit card company). There are typically source systems to collect/generate data relevant to each business process.
- One or more fact tables will be built to support each key business process.

☐ Grain:

- Level of detail; how you describe a single row in the fact table (e.g., individual purchase transaction, individual invoice line item, daily inventory snapshot, monthly account snapshot, etc.). In general, we recommend that you start with the lowest level of information associated with a business process for maximum flexibility.

☐ Dimensions:

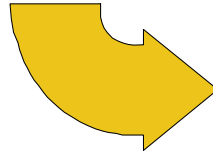
- Dimensions represent the way business people talk about the data resulting from a business process. They are the “by” words used to describe analytical requirements. The grain often determines the primary set of dimensions. Each dimension could be thought of as an analytical “entry point” to the facts.

☐ Facts:

- What metrics result from the business process? Facts are measured, “continuously valued,” rapidly changing information (e.g., unit quantity, dollar sales, price, etc.). Facts may also be calculated and/or derived.

Key Input to Dimensional Modeling

***Business
Requirements***



***Data
Realities***

**Dimensional
Data Model**
Business Process
Grain
Dimensions
Facts
...

© 2004 Kimball University. All rights reserved.



- ☐ You need to consider two factors, user requirements and the realities of your data, in tandem to decide on these key design points.
- ☐ You should resist the temptation to model your data by looking at copy books alone. Unfortunately, many organizations attempt this “path of least resistance” approach, but without much success.

Retail Store Summarized Business Case



Background:

- ☐ *Chain consists of over 100 grocery stores in five states*
- ☐ *Stores average 60,000 stock keeping units (SKUs) in departments such as frozen foods, dairy, etc.*
- ☐ *Bar codes are scanned directly into the cash registers' point-of-sale (POS) system*
- ☐ *Products are promoted via coupons, temporary price reductions, ads and in-store promotions*

Analytic Requirements:

- **Need to know what is selling in the stores each day in order to evaluate product movement, as well as to see how sales are impacted by promotions**
- **Need to understand the mix of products in a consumer's market basket**

© 2004 Kimball University. All rights reserved.



Design Steps 1 - 3



1. Identify the Business Process:

2. Identify the Grain:

3. Identify the Dimensions:

© 2004 Kimball University. All rights reserved.



☐ Vocabulary reminder:

- Business process candidates are the major processes in the company where data is being collected (operational systems or external data sources). Most data warehousing applications want to analyze the performance of an organization's core business processes.
- Grain is the level of detail represented in the fact table. It is the answer to the question, "What is an individual fact record, exactly?"
- Dimensions are the entry points to the metrics or facts. They are the "by" words used by the business users to describe their analysis requirements. The grain will often determine the primary set of dimensions.

Retail Store Sales Schema



© 2004 Kimball University. All rights reserved.



☐ Date Dimension:

- Be sure to include attributes that support analysis and add flexibility (e.g., holiday, day of week, fiscal periods, etc.). These attributes can not be derived by SQL.
- Indicator fields, such as holiday or current period, should contain robust text values, such as “Holiday” and “Non-Holiday” rather than “Yes/No,” “Y/N,” or “1/0.”

☐ Product Dimension:

- Remember to identify roll-up hierarchies when listing dimension attributes. Department could be included on the Product dimension only if each product rolled up to a single, consistent department. If the roll-up of products into departments varies, then you’d likely treat department as a separate dimension with a foreign key in the fact table.

☐ Promotion Dimension:

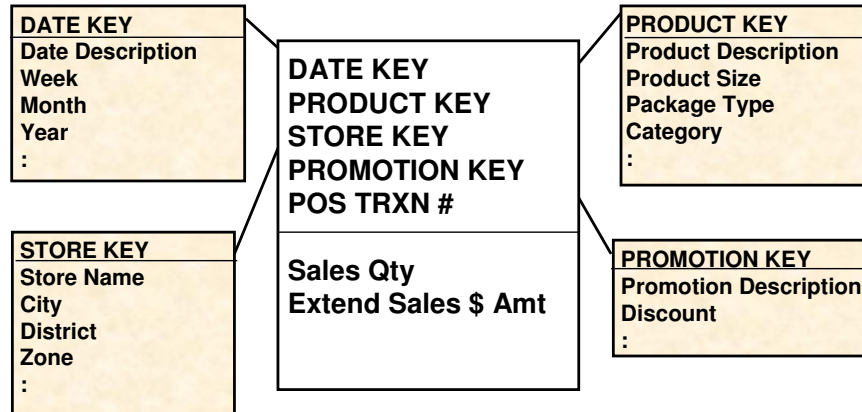
- Promotion is very difficult to accurately capture. Promotions impact purchase behavior, but so do other causal factors such as weather conditions, personal preferences, etc. Determining which promotions are credited for the sale based on user-defined business rules is an approximation at best.
- You’ll want to include a row to identify “No Promotion in Effect.”

☐ The POS transaction ID is a “degenerate” dimensions. It is included in the fact table, but doesn’t join to a dimension table. Degenerate dimensions are often required for row uniqueness. They’re also helpful for grouping of rows (e.g., pulling together all the products within a market basket).

- Operational transaction ID numbers, like invoice #, order #, ticket #, etc., are prime candidates for degenerate dimensions. We do not create surrogate keys for these degenerate dimensions.

Retail Store Sales Star Schema in Action

What were the weekly dollar sales for the snacks category during the “Super Bowl” promotion in the Boston District during the month of January 2004?



© 2004 Kimball University. All rights reserved.



❑ Query constraints would include:

- Month = “January” in Date Dimension Table
- Year=“2004” in Date Dimension Table
- District = “Boston” in Store Dimension Table
- Category = “Snacks” in Product Dimension Table
- Promotion Description = “Super Bowl” in Promotion Dimension Table

❑ Query results/display would include:

- Week from Date Dimension Table
- Sales \$ Amount (summed) from Sales Fact Table

Data Warehouse Surrogate Keys



☐ Recommend surrogate keys

- *Integer, non-meaningful, sequence number*
- *Surrogate keys join fact and dimension tables*
- *Treat natural, operational keys as attributes*

☐ Benefits

- *Isolate warehouse from operational changes*
- *Improve performance*
- *Handle "Not applicable," "Date TBD," ...*
- *Allow integration from multiple sources*
- *Enable tracking of dimension changes*

© 2004 Kimball University. All rights reserved.



☐ Initially, it may be faster to implement a data warehouse using operational keys, but surrogate keys will definitely pay off in the long run.

☐ Benefits of meaningless, surrogate keys in the data warehouse:

- Maintains control within the data warehouse environment rather than being whipsawed by operational system changes/business rules (e.g., reuse of dormant keys, reassignment of keys due to operational requirements, overlapping keys resulting from acquisition, etc.).
- Smaller keys translate into smaller fact tables, smaller fact table indices and more fact table rows per block I/O.
- Allows for the integration of multiple operational source systems if they lack consistent operational keys.
- Stay tuned for details on the role of surrogate keys to track dimension changes.
- Embedded meaning in the keys will haunt you when changes occur. For example, the typical retail operational key identifies product, department and class. When an product moves to a new department, you'd need to modify meaningful keys in both the Fact and Product tables.
- Larger keys increase the size of the Fact table and associated index sizes, degrading performance. Additional joins associated with compound keys also slow performance.

☐ Issues with surrogate keys:

- In the data staging process, a cross-reference look-up table must be used to assign the appropriate surrogate key to each fact and dimension table row. Surrogate key processing and a sample cross-reference table will be discussed later in the class.

Retail Store Sales Sample Table Rows

Date Dimension Table:

Date Key	Date	Day of Week	Day Nbr. in Month	Month	Quarter	Year	Holiday Indicator
1	1/1/2002	Tuesday	1	January	Q1	2002	Holiday
2	1/2/2002	Wednesday	2	January	Q1	2002	Non-Holiday
3	1/3/2002	Thursday	3	January	Q1	2002	Non-Holiday
4	1/4/2002	Friday	4	January	Q1	2002	Non-Holiday

Product Dimension Table:

Prod Key	Prod Desc.	SKU Number	Department	Size	Package Type	Brand	Category	Units/Case
1	Lasagna 6 oz	90706287103	Grocery	6 oz	box	Cold Gourmet	Frozen Foods	48
2	Beef Stew 6 oz	16005393282	Grocery	6 oz	box	Cold Gourmet	Frozen Foods	48
3	Extra Nougat 2 oz	46817560065	Grocery	2 oz	can	Chewy	Candy	48

Store Dimension Table:

Store Key	Store Name	City	State	Zip	District	Region	Store Type	Open Date	Remodel Date
1	Store No. 1	New York	NY	91089	New York	Eastern	Modern	1/9/1982	12/5/1998
2	Store No. 2	Chicago	IL	14594	Cook	Mid West	Original	4/2/1970	6/4/1973
3	Store No. 3	Atlanta	GA	54315	Fulton	South East	Compact	6/14/1959	11/19/1967
4	Store No. 4	Los Angeles	CA	52944	Los Angeles	Pacific	Modern	9/27/1979	12/1/1995
5	Store No. 5	San Francisco	CA	86969	San Francisco	Pacific	Original	9/18/1978	6/29/1991

© 2004 Kimball University. All rights reserved.



Retail Store Sales Sample Table Rows continued

Promotion Dimension Table:

Promo Key	Promo Name	Price Reduction	Ad Type	Media Type	Promo \$	Begin Date	End Date
1	Blue Ribbon Discounts	Temporary	Daily Paper	Paper	2000	1/1/99	1/15/99
2	Red Carpet Closeout	Markdown	Sunday Paper	Paper	1000	1/3/99	1/10/99
3	Ad Blitz	None	Paper and Radio	Paper and Radio	7000	1/15/99	1/30/99
4	Ads and Racks	None	Paper and Radio	Paper and Radio	3000	2/1/99	2/10/99

Sales Fact Table:

Date Key	Prod Key	Store Key	Promo Key	POS Trxn #	Sales Qty	Ext Sales \$ Amt
1	1	1	15	763457893	1	4.59
1	2	1	1	763457893	2	0.90
1	5	11	19	763457894	1	2.56
1	13	5	8	763457923	1	0.33
2	5	11	12	763457998	1	1.29
2	6	11	6	763457998	2	0.62
2	25	11	10	763457998	3	0.63
3	2	1	3	763458013	1	1.19
3	3	1	5	763458013	3	3.27

© 2004 Kimball University. All rights reserved.



Multiple Hierarchies in Dimensions

❑ Dimension tables can represent multiple hierarchical roll-ups

▪ *Store Dimension could have the following hierarchies:*

- *Physical Geography:*
Zip, City, County, State, Country
- *Sales Organization:*
District, Region, Zone
- *Format Type*

STORE KEY
Store Desc
Zip
City
State
Sales Region
Sales Zone
Store Format

© 2004 Kimball University. All rights reserved.



❑ Benefits of maintaining multiple hierarchies in a single dimension:

- Provides a single place for dimension table browsing to fully understand the stores and the relationship of the attributes to the stores (e.g., what stores make up the "East Region," and of those stores, how many are in the state of "New York," etc.).
- Simplified format.
- Represents the way the business users think about their business.

❑ Comments regarding design implications for end user access tools:

- Hierarchical data is treated differently by the relational OLAP end user tools.
- Ideally, users should be able to not only drill up and down hierarchies, but also non-hierarchical dimension attributes (such as display square feet, number of registers, etc.).

Avoid the “Too Many Dimensions” Trap

Inappropriate Sales Fact Table

DATE KEY
MONTH KEY
YEAR KEY
PRODUCT KEY
BRAND KEY
CLASS KEY
STORE KEY
STORE STATE KEY
STORE DISTRICT KEY
STORE REGION KEY
PROMOTION KEY
PROMO MEDIA KEY
POS TRXN #
Sales Qty
Extend Sales \$ Amt

Preferred Sales Fact Table

DATE KEY
PRODUCT KEY
STORE KEY
PROMO KEY
POS TRXN #
Sales Qty
Extend Sales \$ Amt

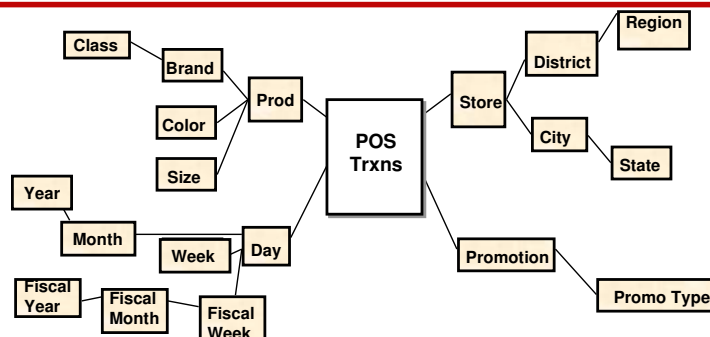
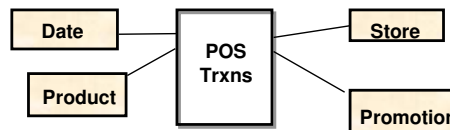


© 2004 Kimball University. All rights reserved.



- ❑ The fact table on the left, affectionately referred to as a “centipede” fact table, reflects the hierarchical dimension attributes in the fact table rather than storing these roll-up relationships in dimension tables. This approach takes denormalization to an extreme.
- ❑ Remember, the fact table is your largest table. The centipede approach is terribly problematic in terms of increased fact table storage requirements, larger fact table indexes, more table joins, more fact table refreshes due to hierarchy changes, etc. Instead of creating centipedes, cluster the dimension attributes into logical business groupings in dimension tables.
- ❑ Alternatively, some designs eliminate all joins to the fact table by denormalizing all the dimension attributes into the fact table. In this case, we’re storing huge amounts of redundant text in the already-large fact table.
- ❑ You’re nearing the practical limit on the number of dimensions when you’re approaching 15 to 18 dimensions. If your design has more, consider combining correlated dimensions into a single dimension.

Star vs. Snowflake Design Variations

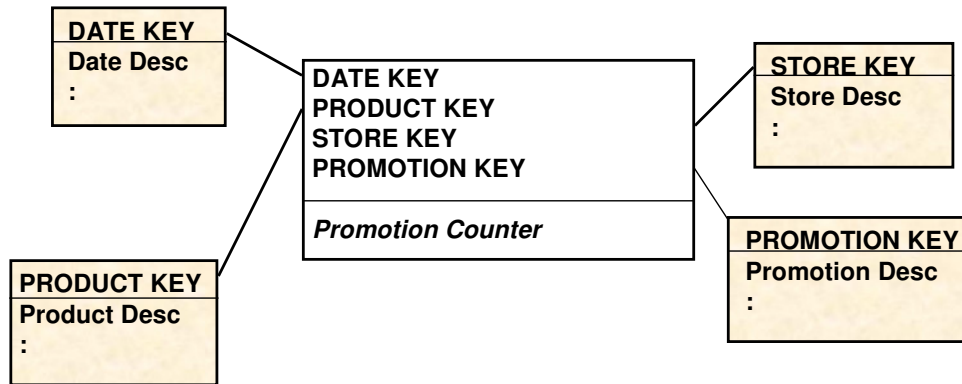


© 2004 Kimball University. All rights reserved.



- ❑ The snowflake schema is a variation of the star schema in which the redundant attributes are removed from the dimension tables and placed in normalized “sub dimension” tables. The resulting diagram resembles a snowflake.
- ❑ The fact tables in the above star and snowflake schemas are identical, however, the dimension tables were modeled differently. In general,
 - Star schemas denormalize the dimension tables.
 - Snowflake schemas normalize the dimension tables.
- ❑ Unless your end-user tool is optimized for this design, be aware of the following:
 - Snowflaking almost always makes the user presentation more complex and more intricate.
 - Snowflaking makes most forms of browsing among the dimension attributes slower as one or more of the other attributes within the dimension are typically constrained.
 - Many database optimizers can not effectively deal with the plethora of joins in a snowflaked schema.
- ❑ If you have multiple end-user tools in your environment, some optimized for the star schema and another optimized for the snowflake variation, you may want to support copies of both denormalized and snowflaked dimension tables.

Factless Fact Tables - Promotion Event/Coverage Table



© 2004 Kimball University. All rights reserved.



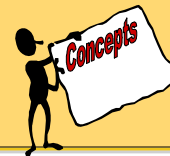
- ☐ Although the schema shown above does not apply to the business process stated in the first case study, it does demonstrate a useful design technique.
- ☐ If we needed to track all known promotion conditions, we could create a “factless” Promotion Event fact table.
 - Factless fact tables record either coverage or the occurrence of an event.
 - The factless fact table in this schema would allow us to identify when a promotion was in place, regardless of whether or not there was a sale.
- ☐ As the name suggests, factless fact tables do not have numeric events. Sometimes a counter with the value of “1” is added to each fact record, either physically or via a view, to facilitate counting.

Beyond the 1st Data Mart Case Study

Retail Store Inventory Schema

© 2004 Kimball University. All rights reserved.

Section Concepts

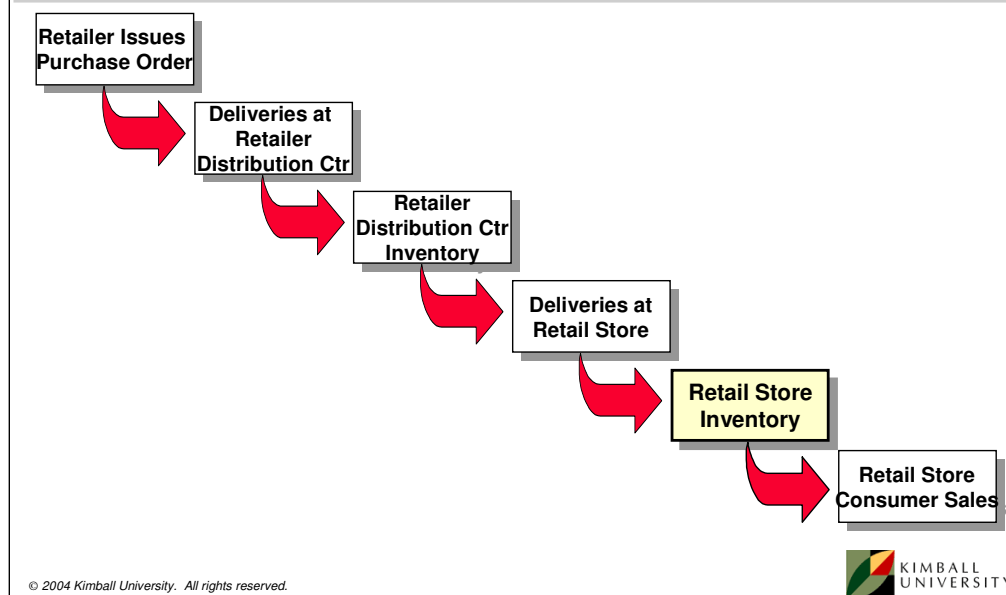


- ☐ Build dimensional models across the enterprise using Data Warehouse Bus Matrix
 - *Exercise: Translating requirements into matrix*
- ☐ Reuse conformed dimension tables
- ☐ Semi-additive facts

© 2004 Kimball University. All rights reserved.



Retail Industry “Value Chain” of Business Processes



- ❑ Most organizations follow a logical process flow. The “value chain” identifies the key business processes within an organization.
- The objective of most analytic decision support systems is to monitor the key business processes within the value chain.
 - Operational systems typically generate transactions or snapshots at each step of the value chain, resulting in key performance metrics associated with each business process.
 - Since each business process captures unique metrics at unique time intervals with unique granularity, each business process is represented by a separate fact table.
 - Understanding an organization’s value chain provides the global perspective needed to develop the overall DW data architecture without embarking on a 12-month enterprise modeling project.

Retail Store Inventory Summarized Business Case

Background:

- ☐ Optimized inventory levels have a major impact on retail chain profitability -- making sure the right product is in the right stores at the right time minimizes out-of-stocks and reduces overall inventory carrying costs
- ☐ End-of-day quantity-on-hand is recorded on a nightly basis for each product in each store
- ☐ Quantity-on-hand is based on the beginning inventory level less the quantity sold, plus the quantity received during the day

Analytic Requirements:

- Need to know which stores are consistently maintaining higher inventory levels for which products

© 2004 Kimball University. All rights reserved.



Design Steps 1 - 3



1. Identify the Business Process:

2. Identify the Grain:

3. Identify the Dimensions:

© 2004 Kimball University. All rights reserved.



Retail Store Inventory Schema:



© 2004 Kimball University. All rights reserved.



☐ Date Dimension:

- The Date dimension table in this case study is identical to the table developed in the first case for the retail store sales process.

☐ Product and Store Dimensions:

- The dimension tables used in the first case may be supplemented with additional attributes useful for inventory analysis.

☐ Facts:

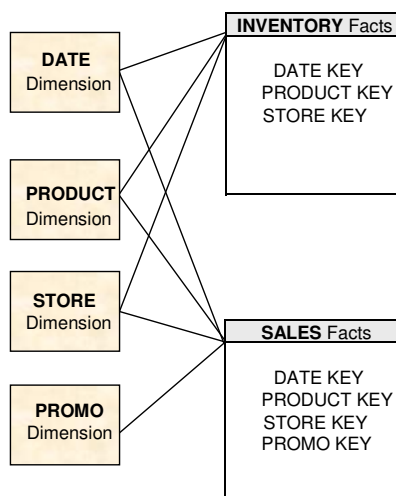
- What are the numeric facts captured by the Nightly Inventory Snapshot? What other facts might be useful?

☐ Semi-additive facts cannot be summed across all dimensions. Any “balance” facts, such as inventory balances or account balances, or other measures of intensity cannot be summed over the Date dimension. They represent snapshots at a given point in time. Consider the example of quantity-on-hand inventory balance for Cheerios.

- You can sum quantity-on-hand for all Cheerios sizes within a store for given day (assuming equivalized volume)
- You can sum quantity-on-hand for all 24 oz Cheerios across stores for given day.
- But you can not sum quantity-on-hand for all 24 oz Cheerios for given store during last week. If there were 10 boxes on Monday, 5 boxes on Tuesday, ..., you can't add $10 + 5 + \dots$ to determine the quantity-on-hand for the week.
- You typically average over the number of time periods by summing quantity-on-hand for last week and dividing by the number of time periods.

Value Chain Design Implications

- ❑ **Separate fact tables to represent each business process**

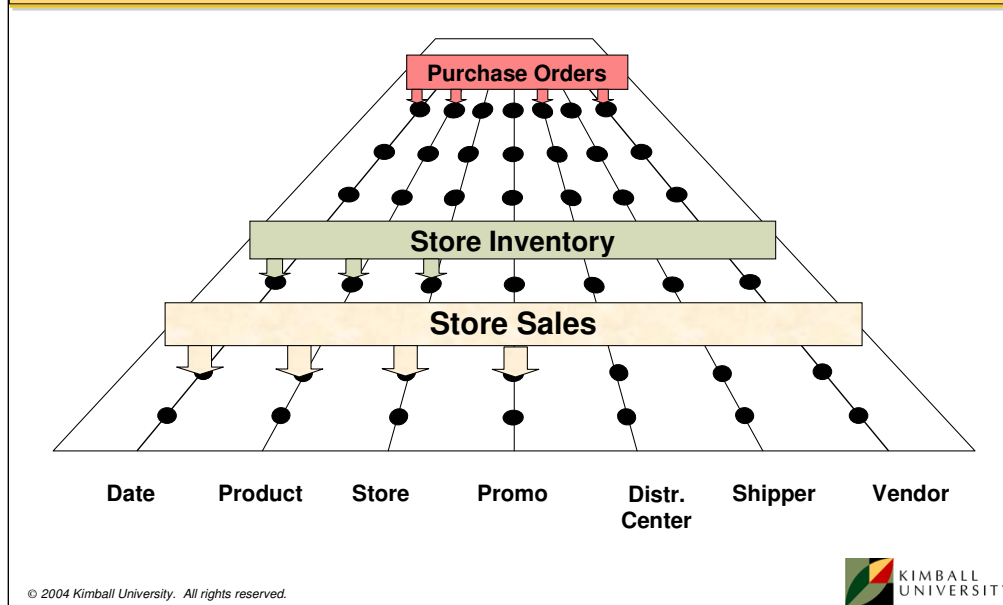


© 2004 Kimball University. All rights reserved.



- ❑ Each business process typically is represented by one or more fact tables since the grain, dimensions and facts are unique to each business process.
- ❑ We recommend starting development of your data warehouse by focusing on a single business process. After several single-source data marts have been implemented, it is reasonable to combine them into a multi-source mart. The classic example of a multi-source mart is the profitability data mart.
- ❑ Note: The graphic illustrating the shared dimension tables is meant to be a logical representation. DO NOT construct queries linking the two fact tables as depicted to query both Inventory and Sales facts - the results may be over counted. You would need a tool with multi-pass SQL capabilities to correctly resolve a query involving both fact tables.

Data Warehouse Bus Architecture



- ☐ The Data Warehouse Bus Architecture provides a standardized master set of conformed dimensions and conformed facts used throughout the data warehouse. It is analogous to the bus in your computer, providing a standard interface that allows many different kinds of devices to connect to your computer and co-exist.
- ☐ Conformed dimension are standard dimensions that are shared among data marts. A dimension is conformed between data marts either if it is exactly the same dimension (including all attributes and rollups within the dimension) or one dimension is a strict subset of the other. The use of conformed dimensions is the central technique for building an enterprise data warehouse from a set of data marts.
- ☐ Using the Data Warehouse Bus Architecture framework, as the separate data marts are developed, they plug into the Bus, fitting together like pieces of the puzzle. Both relational and multidimensional databases can connect to the Data Warehouse Bus.
- ☐ Isolated data marts that cannot be tied together are disastrous. Stovepipe data marts merely perpetuate incompatible views of the business.

Data Warehouse Bus Matrix

☐ Business processes and shared dimensions

	Date	Product	Store	Promo	Dist Ctr	Shipper	Vendor
Store Sales	X	X	X	X			
Store Inventory	X	X	X				
Store Deliveries	X	X	X		X	X	
Dist Ctr Inventory	X	X			X		
Dist Ctr Delivery	X	X			X	X	X
Purchase Orders	X	X			X		X

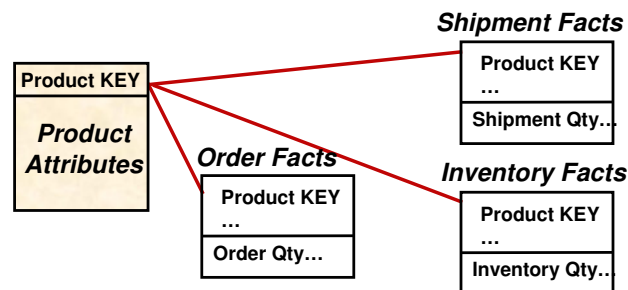
© 2004 Kimball University. All rights reserved.



- ☐ Rows in the matrix represent business processes (NOT business organizational departments or functions) which translate into data mart fact tables. Begin with single source data marts - we refer to these as first level data marts. Identify more complex, multi-source marts, such as profitability, as a secondary step. The matrix rows match the quadrant analysis “themes” discussed earlier.
 - The rows for a telecommunications company might include call detail, customer billing, service orders, network inventory, etc.
 - For a credit card processor, the business process rows would include card transactions, cycle billing, collections, etc.
- ☐ Columns represent common dimensions used across the enterprise. Mark the intersections where the dimensions are relevant to the business processes. The resulting matrix will be surprising dense.
- ☐ Sharing conformed dimensions across the data warehouse is absolutely critical.
 - Ensures consistent definition of common data.
 - Ensures consistent row/column heading labels and roll-ups.
 - Ensures consistent “values” for the consistently defined dimensions and attributes.
- ☐ The matrix serves as a tool for planning, communication and expectation management with management and other team members. It is a major responsibility of the central data warehouse team to establish, publish, maintain and enforce conformed dimensions. Committing to use conformed dimensions is a business policy. It represents more political challenges than technical hurdles. Once conformed dimensions are agreed to, then data warehouse development can occur concurrently.

Terminology: Conformed Dimensions

- ❑ 1 row for each instance of a business subject / object (product, customer, etc.)
- ❑ All fact tables use same standard dimensions
 - *Established via Bus Matrix, enforced in ETL*
- ❑ Consistent across processes



© 2004 Kimball University. All rights reserved.



- ❑ The same dimension will be involved in multiple business processes. For example, the product dimension will be involved in supplier orders, inventory, shipments and returns. If we create a single dimension that applies across all these processes, we call this a conformed dimension.

This is a critical underpinning of our ability to do analysis across the business (drill-across). For example, if we wanted to calculate inventory turns, we need orders by product and average inventory by product. If the “product” is exactly the same in each of these two queries, we can divide the two results and get our answer.

- ❑ Note that this idea of drilling across multiple fact tables and combining the answer sets requires a front end tool intelligent enough to support this function.
- ❑ Conformed dimensions ensure that we are comparing apples to apples, assuming we are selling apples...

Conformed Dimension Option #1

- ❑ Identical dimensions with the same keys, labels, definitions and values

Sales Schema

PRODUCT KEY
Product Desc
Brand Desc
Category Desc
:

DATE KEY
PRODUCT KEY
STORE KEY
PROMO KEY
Sales Facts

Inventory Schema

PRODUCT KEY
Product Desc
Brand Desc
Category Desc
:

DATE KEY
PRODUCT KEY
STORE KEY
Inventory Facts

© 2004 Kimball University. All rights reserved.



- ❑ As we observed earlier, we used identical, conformed Product dimensions in the first two case studies. The Product dimension is either the same physical dimension within a database or synchronously duplicated in each data mart.

Conformed Dimension Option #2

- ❑ “Subset” of base dimension with common labels, definitions, and values

Sales Schema

PROD KEY
Product Desc
Brand ID
Brand Desc
Category Desc

DATE KEY
PROD KEY
STORE KEY
PROMO KEY
Sales \$

DATE KEY
Day-of-Week
Week Desc
Month Desc
:

PROD KEY **Product Desc** **Brand ID** **Brand Desc** **Category Desc**
 0001 Cheerios 10oz CH010 Cheerios Cereal

Forecast Schema

BRAND KEY
Brand ID
Brand Desc
Category Desc

MONTH KEY
BRAND KEY
Fcst Sales \$

MONTH KEY
Month Desc
:

BRAND KEY **Brand ID** **Brand Desc** **Category Desc**
 2548 CH010 Cheerios Cereal

© 2004 Kimball University. All rights reserved.



- ❑ Assume that one business process captures product-related data at the atomic item level, such as the case studies discussed already. Assume another business process, perhaps forecasting, does not deal with product information data at the item level, but at the brand level.
- ❑ In this case, you couldn't share a single Product dimension table across the two business process schemas because the granularity is different. Sales are reported at the Product level, but forecasting is done by Brand.
 - The Brand dimension table should be a shrunken subset of the atomic Product dimension table.
 - Shared dimension attributes, such as Brand Desc and Category Desc should be identically labeled and valued.
- ❑ Another common example of dimension conformance deals with the date dimension. Some schemas will be at a daily granularity; other schemas will be at a monthly granularity. These two separate date-related dimension tables must share column names, definitions and valid values, where appropriate. For example, “Fiscal year” might be an attribute on both tables - it should be spelled the same, mean the same thing, and contain the same values on both the Daily Date and Monthly Date dimension tables.
- ❑ NOTE: If you are a conglomerate with widely varying subsidiaries covering a range of industries such as food, clothing, etc., it may not make sense to conform dimensions with such disjoint products and customers (assuming they don't overlap across lines of business).

Translating Requirements into Data Warehouse Bus Matrix Exercise

VaporWare is a \$800 million software company which sells a range of products to commercial customers worldwide. You've conducted interviews at VaporWare and need to develop a preliminary Data Warehouse Bus Matrix based on the summarized interview results below.

	← COMMON DIMENSIONS →					
BUSINESS PROCESSES ↓						

Marketing

Marketing is most interested in better understanding VaporWare's customers. They want to slice-and-dice daily invoice revenue by customer and product to analyze product penetration and cross-sell/up-sell opportunities.

Sales

Sales management wants to better understand the performance of their sales organization in terms of both sales orders and invoice revenue – which reps are selling and invoicing which customers for what products under which discount terms.

In addition to sales rep performance information, Sales is also interested in customer support cases. They want to understand which customers are experiencing product-related support problems.

Finally, sales management wants the ability to better analyze forecast data. They'd like the ability to manipulate their monthly sales forecasts by sales rep, product and customer characteristics.

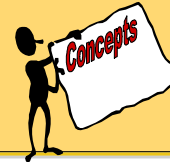
Technical Support

Tech Support currently has a stand-alone system for capturing customer support cases, however the analytic capabilities are very limited. They would like the ability to analyze support cases by date, product, customer, and technician. They'd also like to access customer invoice revenue information.

More Dimensional Model Fundamentals

© 2004 Kimball University. All rights reserved.

Section Concepts



- ☐ Techniques for handling slowly changing dimension attributes

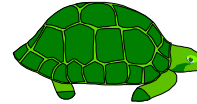
© 2004 Kimball University. All rights reserved.



Dealing with Slowly Changing Dimensions

- ❑ Dimension attributes evolve over time
 - *For example, customers change their names, move, have children, adjust their incomes*

- ❑ For every dimension attribute, need to identify “change” strategy
 - *May use combination of strategies within same dimension table*



© 2004 Kimball University. All rights reserved.

- ❑ As part of the modeling activity, you should identify if and how changes will be tracked for each attribute on each dimension table.

Slowly Changing Dimensions Option #1

TYPE 1: Overwrite the Changed Attribute

Original record

PROD KEY	PROD ID	PROD DESC	DEPT
12345	SC3000	Sim City 3000	Educational S/W

Updated record

PROD KEY	PROD ID	PROD DESC	DEPT
12345	SC3000	Sim City 3000	Strategy S/W

© 2004 Kimball University. All rights reserved.



- ☐ This is the simplest approach. It is useful whenever the old value of the attribute has no significance or should be discarded (e.g., correction of an error). Unfortunately, it doesn't always meet the business requirements. Storage requirements and costs used to drive this decision, although this is less an issue today.
- ☐ Advantage:
 - Easy & fast.
- ☐ Disadvantage:
 - You lose history of attribute changes. You only see the descriptive attributes as they exist today.
 - In this example, if you look at Sim City sales over time, sales suddenly take off, but you don't know why!

Slowly Changing Dimensions Option #2

TYPE 2: Add a New Dimension Record

Original record

PROD KEY	PROD ID	PROD DESC	DEPT
12345	SC3000	Sim City 3000	Educational S/W

Additional record

PROD KEY	PROD ID	PROD DESC	DEPT
21687	SC3000	Sim City 3000	Strategy S/W

© 2004 Kimball University. All rights reserved.



☐ Advantage:

- Elegant approach for perfectly partition history as pre-change fact records still reflect the original product key, however it shouldn't be abused.

☐ Disadvantage:

- Requires use and administration of surrogate keys (but you're already using them anyhow). We will discuss the administering of surrogate keys to support a Type 2 strategy later in this class.
 - Dimension table growth due to additional rows.
 - Users must be aware of this added complexity to ensure their reports are correct and format properly.
- ☐ Effective date could be added, but many times this doesn't prove to be entirely valuable. The fact table is a much cleaner and accurate means to measure "effective" dates. However, an effective date in the dimension table could identify the most current row (as would a current indicator).

Slowly Changing Dimensions Option #3

TYPE 3: Add a “Prior” Attribute

Original record

PROD KEY	PROD DESC	DEPT
12345	Sim City 3000	Educational S/W

Updated record

PROD KEY	PROD DESC	DEPT	PRIOR DEPT
12345	Sim City 3000	Strategy S/W	Educational S/W

© 2004 Kimball University. All rights reserved.



- ☐ In this case, we add columns to the dimension table to capture the attribute change. This technique is used relatively infrequently.
- ☐ Advantage:
 - It's appropriate for tracking “soft” changes, such as when a sales reorganization occurs and the business wants to look at performance for both the old and current organization structures. It simultaneously supports two views of the world.
- ☐ Disadvantage:
 - Intermediate attribute values are lost with this approach - only current and prior (or current and original) are retained. It would be too cumbersome to retain any additional interim valuations.
 - You can't trend changes over time as elegantly as when the fact table is used (Type 2). In fact, it's not very easy to do at all.

Slowly Changing Dimensions Advanced Hybrid Option

Hybrid Approach:

- *Use Type 2 to track changes as they occur*
- *Include "current" Type 3 attribute treated as Type 1*

Original row

PROD KEY	PROD DESC	Type 2 "AS WAS" DEPT	Type 3/1 CURRENT DEPT
12345	Sim City 3000	Education	Education

© 2004 Kimball University. All rights reserved.



Slowly Changing Dimensions Advanced Hybrid Option cont'd

Change: Sim City moved from Education to Strategy

PROD KEY	PROD DESC	Type 2 "AS WAS" DEPT	Type 3/1 CURRENT DEPT
12345	Sim City 3000	Education	Strategy
21687	Sim City 3000	Strategy	Strategy

© 2004 Kimball University. All rights reserved.



Slowly Changing Dimensions Advanced Hybrid Option cont'd

Change: Sim City moved from Strategy to Crit Think

PROD KEY	PROD DESC	Type 2 "AS WAS" DEPT	Type 3/1 CURRENT DEPT
12345	Sim City 3000	Education	Critical Thinking
21687	Sim City 3000	Strategy	Critical Thinking
28932	Sim City 3000	Critical Thinking	Critical Thinking

© 2004 Kimball University. All rights reserved.



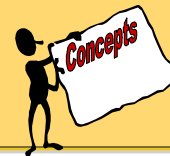
- ❑ In our experience, data warehouse teams are often asked to preserve historical attributes, while also supporting the ability to report historical performance data according to the current attribute values. None of the standard SCD techniques enable this requirement independently. However, by combining techniques, you can elegantly provide this capability in your dimensional models.
- ❑ We'll begin by using the SCD workhorse, Type 2, to capture attribute changes. When the product roll-up changes, we'll add another row to the dimension table with a new surrogate key. We'll then embellish the dimension table with additional attributes to reflect the current roll-up.
- ❑ In the most current dimension record for a given product, the current roll-up attribute will be identical to the historically accurate "as was" roll-up attribute. For all prior dimension rows for a given product, the current roll-up attribute will be overwritten to reflect the current state of the world.
- ❑ If we want to see historical facts based on the current roll-up structure, we'll filter or summarize on the current attributes. If we constrain or summarize on the "as was" attributes, we'll see facts as they rolled up at that point in time.
- ❑ As with so many things, the cost of greater flexibility is often increased complexity. Don't pursue this option unless it's necessary to address your users' requirements.

Transaction Detail Case Study

Manufacturer Invoicing Schema

© 2004 Kimball University. All rights reserved.

Section Concepts

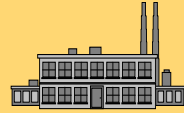


- ☐ “Degenerate” dimensions, again
- ☐ Dimension table “role playing”
- ☐ Common invoice complications
- ☐ Process of developing and communicating dimensional models

© 2004 Kimball University. All rights reserved.



Manufacturer Summarized Business Case



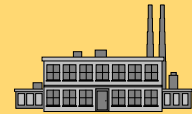
Background:

- ☐ Manufacturer of several hundred products - products are sold to approximately 20,000 customers
- ☐ Invoices are generated when we ship products from our warehouses to the customer
- ☐ Invoice identifies the ship date and requested ship date
- ☐ Each invoice line identifies the product shipped and the line item quantity, gross amount, discount, and net amount

© 2004 Kimball University. All rights reserved.



Manufacturer Summarized Business Case



Analytic Requirements:

- Need to know how much we're invoicing which customers on a daily basis
- Want to know which products are shipping to which customers from which warehouses
- Need to analyze customer service levels (e.g., which shipments were requested to ship in given week, but didn't ship until the following week)
- Want to track the mix of products shipped on a single invoice

© 2004 Kimball University. All rights reserved.



Manufacturer Invoicing: Design Steps 1 - 3



1. Identify the Business Process:

2. Identify the Grain:

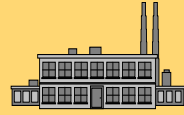
3. Identify the Dimensions:

© 2004 Kimball University. All rights reserved.



- ☐ There is tremendous power and resiliency in granular fact table data. It is a myth that users only need to view aggregated information for analysis. Although analytical users won't necessarily want to look at a single invoice, order or claim, it's impossible to predict all the ways in which they'll want to summarize that bedrock data. The most granular data must be structured dimensionally. It is impractical to drill through dimensional summary data, then hit a wall or disconnect when users want to access the detailed data.

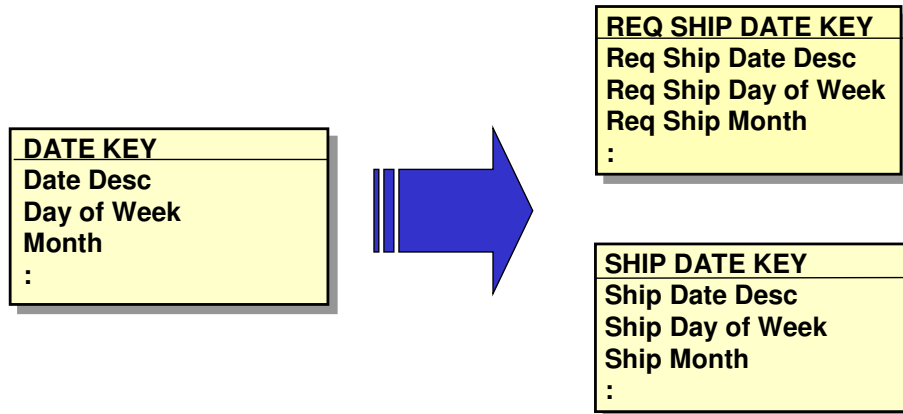
Manufacturer Invoicing Schema



© 2004 Kimball University. All rights reserved.



Dimension Table Role Playing



© 2004 Kimball University. All rights reserved.



- ❑ Two unique dates are associated with the invoice; both are needed to support user analysis:
 - Requested Ship Date
 - Ship Date

Two separate physical Date dimension tables would address this user requirement.
- ❑ Alternatively, you can utilize a single physical table that plays multiple “roles”, creating the illusion of two independent Date tables using synonyms and views. Each of the Date dimensions can then be used independently, with completely unrelated constraints, such as select all orders where Requested Ship Month = “January” and Ship Month = “February”.
- ❑ Other examples of common “role playing” dimensions include:
 - Origin and destination cities, stations or airports in transportation.
 - Primary, referring and admitting physicians in health care.
 - Calling party and called party in telecommunications.
 - Location in many industries.

Common Invoicing “Complications”

❑ Facts of different granularity

- *Strive to allocate to line item grain*
- *Otherwise, different fact tables for different grains*

❑ Multiple currencies

- *Include both local and equivalent standardized currency facts in fact rows (physically or via view)*
- *Same logic applies to multiple units of measure*

❑ Miscellaneous flags

- *Create correlated “junk dimension”*

© 2004 Kimball University. All rights reserved.



❑ Facts of different granularity:

- For example, the invoice may identify shipping costs for the entire invoice, rather than allocating it down to the line item level for each product. In this case, we’d need two fact tables, one at the line item grain and a second at the invoice level.

❑ Multiple currencies:

- Two sets of facts representing both units of currency should be presented to the user to reduce the risk of human calculation error. You can either physically store both sets of facts, or you can store one set, plus the conversion factor at that point in time, and use a view to compute and present the second set of facts.
- Different business functions may want to look at the shipment quantities using varying units of measure, such as shipping cases, consumer units or some equivalized unit of measure to support summing or comparison of different sized products.

❑ Miscellaneous flags:

- There are often a slew of miscellaneous flags associated with each invoice line item. You wouldn’t want to add 5-10 more dimensions to your schema. On the other hand, you don’t want to just include a cryptic textual flag on the fact table. Unfortunately, you can’t just ignore them. The “junk dimension” includes all the combinations of miscellaneous flag values. This solution gives the users the ability to slice-and-dice on these less frequently analyzed flags and indicators, without adding unnecessary foreign keys to the fact table.

Dimensional Modeling Process Requires User Involvement

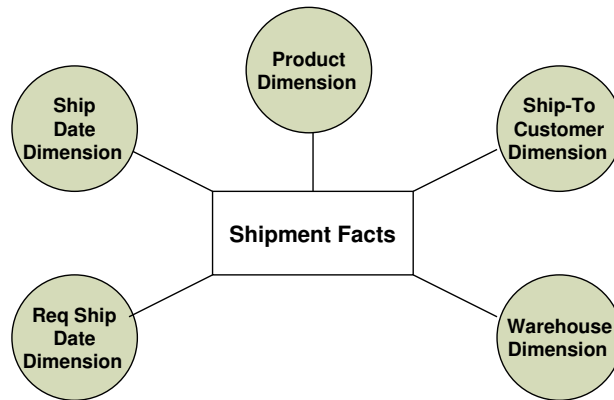
- ☐ Gather biz requirements & conduct data audit
- ☐ Draft DW Bus Matrix and select business process (step 1)
- ☐ Conduct more detailed data analysis
 - *Involve core user liaisons / power users*
- ☐ Conduct iterative design workshops to determine steps 2 - 4
 - *More core user liaison involvement*
 - *Generate flip chart “wallpaper”*
 - *Identify issues/responsibilities*
- ☐ Validate with other business users

© 2004 Kimball University. All rights reserved.



- ☐ Developing dimensional models obviously requires detailed data analysis to understand the source system grain, availability of dimensions attributes, etc.
- ☐ Model development is an iterative process. Typically, the modeling team (Business System Analyst, DBA, perhaps the Project Manager and Business Project Lead) sequesters themselves in a room and generates flip chart “wallpaper”, getting as far as they can with the information available to them. They log issues as they’re encountered, emerge to resolve the inevitable questions, then return to the sequestered room for another round.
- ☐ The draft model should be tested (at least verbally) against the analytic user requirements for this business process to ensure that the questions can be answered.
- ☐ When the modeling team is reasonably confident of their design, it should be reviewed and validated with a core set of users. A facilitator should be identified to present the model; the role of scribe should also be assigned to capture issues and comments.

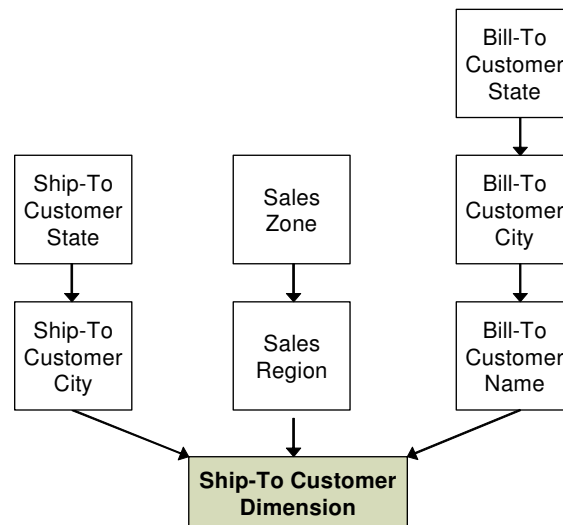
Communicating Dimensional Models to Business Users



© 2004 Kimball University. All rights reserved.



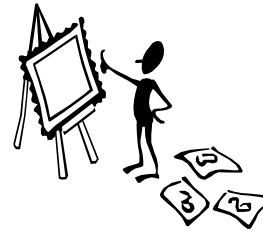
Dimension Hierarchy Diagrams



© 2004 Kimball University. All rights reserved.



Design Workshop



© 2004 Kimball University All rights reserved.

Retail Brokerage Workshop

Summarized Business Case

Background:

- Full-service retail brokerage firm operating in 25 states with 150 branch offices
- 3,000 brokers report into branches which roll into regions
- Accounts classified by product category (such as taxable brokerage accounts, retirement/pension accounts, etc.)
- In each account, securities such as
 - Stocks (Microsoft MSFT, General Electric GE, etc.)
 - Mutual funds (Vanguard Index 500, Templeton Int'l, etc.)
 - Bonds

are bought and sold through trades with brokers

Analytic Requirements:

- Need to understand month-end asset balance trends by account, product, broker and security characteristics
- Need to track changes in month-end account status (e.g., new, active, closed, pending investigation, etc.) over time
- Want to understand which regions, branches and brokers are most successful selling which products based on the number of new accounts and their assets
- Understand the magnitude of the Firm's monthly assets and/or trades with various securities (e.g., Fidelity Magellan)
- Must satisfy Compliance's requirement to identify trades for a given broker or account with the trade control number

Available Source Fields

Account Acquisition Source	Net Trade \$ Amount
Account City, State, Zip	Number of Traded Shares
Account Market Segment	Open Date
Account Number	Product Category (Retirement, ...)
Account Status	Product Description (Roth IRA, etc.)
Branch City, State, Zip	Product Introduction Date
Branch Manager	Product Manager Name
Branch Name	Region Name
Broker Name	Security Name (Microsoft, etc.)
Broker Type	Security Objective (Growth, etc.)
Closed Date	Security Type (Stock, Mutual Fund, etc.)
Date Broker Joined Firm	Ticker Symbol (Microsoft MSFT, etc.)
Gross Trade \$ Amount	Trade Commission \$ Amount
Month End Asset Balance	Trade Control #
Month End Date	Trade Date
Month End Number of Shares	Transaction Type (Buy, Sell, Reinv Div)

Workshop: Design Steps 1 - 3



1. Identify the Business Process:
2. Identify the Grain:
3. Identify the Dimensions:

© 2004 Kimball University. All rights reserved.



Workshop Schema

© 2004 Kimball University. All rights reserved.



**So what have we
accomplished...**

© 2004 Kimball University All rights reserved.

Concepts



- ☐ Differences between facts and dimensions
- ☐ Steps to designing a dimensional data model
- ☐ Multiple hierarchies within dimension
- ☐ Degenerate dimensions
- ☐ Surrogate warehouse keys
- ☐ Factless fact tables
- ☐ Star vs. snowflake design variations
- ☐ Data Warehouse bus matrix
- ☐ Conformed dimensions



Concepts Continued



- ☐ Semi-additive facts
- ☐ Techniques for slowly changing dimensions
- ☐ Dimension table “role playing”



Additional Resources

☐ Kimball Group

Definitive source for the dimensional approach

- www.kimballgroup.com for offerings
- bob@kimballgroup.com



☐ Kimball University

Practical techniques, proven results

- [Register for RK Design Tips](#)
- [Access to DBMS/Intelligent Enterprise articles](#)
- www.kimballgroup.com



© 2004 Kimball University. All rights reserved.



Completed Case Study Worksheets

© 2004 Kimball University All rights reserved.

Retail Store Schema: Design Steps 1 - 3



1. Identify the Business Process:
Point of Sale (POS) Process
2. Identify the Grain:
1 Row per POS Transaction Line
3. Identify the Dimensions:
Date, Product, Store, Promotion, POS Trxn #

© 2004 Kimball University. All rights reserved.



- ☐ In reality, as you're working through the 4-step design process, you may go back and revisit earlier decisions regarding the grain and dimensionality. You should anticipate an iterative design process.

Retail Store Sales Schema:

3. Identify Dimensions



Date Dimension Table

DATE KEY
Date Description
Day of Week
Day Number in Year
Week Ending Date
Week Number in Year
Month / Year
Month Abbreviation
Month Sequence #
Quarter
Year
Fiscal Period
Fiscal Quarter
Fiscal Year
Holiday Indicator
:

© 2004 Kimball University. All rights reserved.



- ☐ Be sure to include attributes that support analysis and add flexibility (e.g., holiday, day of week, fiscal periods, etc.). These attributes can not be derived by SQL.
- ☐ Indicator fields, such as holiday or current period, should contain robust text values, such as “Holiday” and “Non-Holiday” rather than “Yes/No”, “Y/N”, or “1/0.”
- ☐ If we also wanted to capture time-of-day, we’d suggest a separate dimension rather than muddling the Data table with the lower level of granularity. The Time-of-Day dimension could support time slice analysis, such as the lunch hour rush.

Retail Store Sales Schema:

3. Identify Dimensions



Product Dimension Table

PRODUCT KEY
Product Description
Product Size
Package Type
Flavor
Brand Description
Category Description
Department Description
Manufacturer
SKU Number
:

© 2004 Kimball University. All rights reserved.



- ☐ Products have many interesting attributes such as size, color, package type, package size, etc.
- ☐ Be sure to keep product description values both descriptive and unique. This is a big challenge.
- ☐ Remember to identify roll-up hierarchies when listing dimension attributes. Department could be included on the Product dimension only if each product rolled up to a single, consistent department. If the roll-up of products into departments varies, then you'd likely treat department as a separate dimension with a foreign key in the fact table.
- ☐ Although each store carries 60,000 products on average, if the stores are merchandised differently, there may be many more products on the shelf across the chain. The number of rows in the Product dimension table further grows when you consider the historical products which are no longer for sale.

Retail Store Sales Schema:

3. Identify Dimensions



Store Dimension Table

STORE KEY
Store Name
Manager Name
City
State
Zip Code
District
Region
Zone
Store Type
Number of Registers
Total Square Footage
24 Hour Indicator
of Days Open/Week
Store Open Date
:

© 2004 Kimball University. All rights reserved.



- ❑ Again, Store has many interesting potential attributes, such as the number of registers, square footage, last remodel date, etc. Store manager phone number may be included, although this doesn't serve any "analytical" purpose.

Retail Store Sales Schema:

3. Identify Dimensions



Promotion Dimension Table

PROMOTION KEY
Promotion Description
Discount
Media Type
Promo Begin Date
Promo End Date
Total Promotion Budget
:

© 2004 Kimball University. All rights reserved.

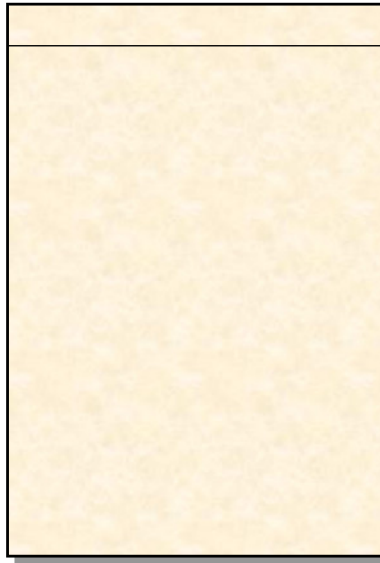


- ☐ Promotion is very difficult to capture fully. Promotions impact purchase behavior, but so do other causal factors such as weather conditions, personal preferences, etc. Determining which promotions are credited for the sale based on user-defined business rules is an approximation at best.
- ☐ You'll need to include a row in the Promotion dimension table to identify "No Promotion in Effect."

Retail Store Sales Schema: 3. Identify Dimensions



**Point-of-Sale
Trxn #**



© 2004 Kimball University. All rights reserved.



- ❑ This case identified the requirement to pull together the items within a market basket. To do so, we need the point-of-sale transaction number.
- ❑ The POS transaction number is a “degenerate” dimensions. It is included in the fact table, but doesn’t join to a dimension table. Degenerate dimensions are often required for row uniqueness and/or grouping of rows (e.g., identify all items purchased in a market basket).
 - Operational transaction ID numbers, like invoice #, order #, ticket #, etc., are prime candidates for degenerate dimensions. We do not create surrogate keys for these degenerate dimensions.

Retail Inventory Schema: Design Steps 1 - 3

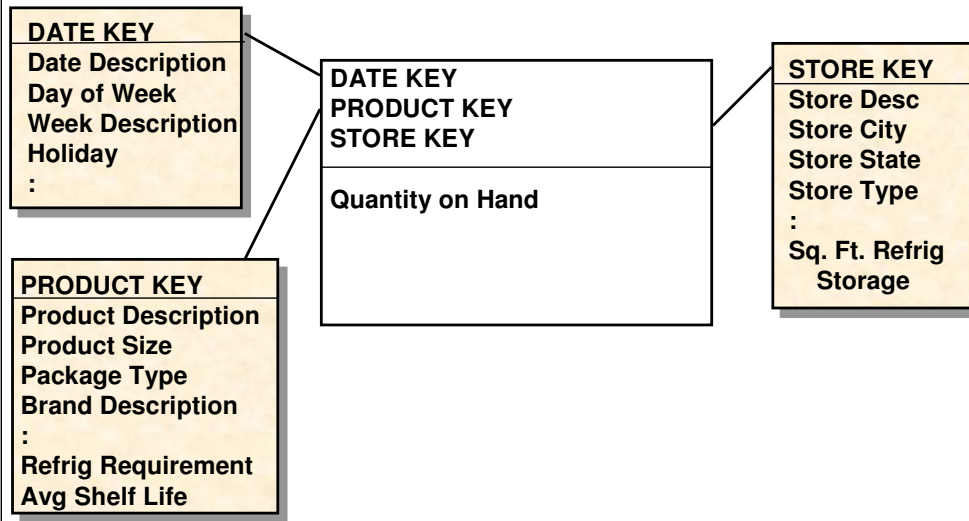


1. Identify the Business Process:
Retail Inventory Snapshot Process
2. Identify the Grain:
1 Row per Day, Product and Store
3. Identify the Dimensions:
Date, Product, Store

© 2004 Kimball University. All rights reserved.



Retail Store Inventory Schema



© 2004 Kimball University. All rights reserved.



- ☐ In this case, we enhanced the conformed dimensions (Date, Product and Store) designed in the first case study.
- ☐ The promotion dimension does not apply to this business process. Promotions are typically tracked with product “movement”, such as when you order, receive or sell the promoted product.
- ☐ This schema satisfied the scope as defined by the user requirements. Other interesting facts that could be added include:
 - Quantity sold - tracking the quantity sold in this schema adds meaning and a measure of relativity to the quantity-on-hand balances.
 - Other inventory-related metrics, such as turns (based on quantity sold / quantity-on-hand) could be calculated.

Translating Requirements into Data Warehouse Bus Matrix

	Date	Customer	Product	Sales Rep	Discount	Technician
Sales Orders	Order Date	X	X	X	X	
Invoice Revenue	Invoice Date	X	X	X	X	
Sales Forecast	Forecast Date	X	X	X		
Support Cases	Incident Date / Close Date	X	X			X

© 2004 Kimball University. All rights reserved.



- ☐ This draft matrix would be flushed out based on further data audit interviews, as well as confirmation sessions with the business.
- ☐ Notice that the rows of the matrix represent business processes, not business functions or groups (like Marketing, Sales, or Technical Support). All three business functions need to analyze the Invoice Revenue data; both Sales and Technical Support are interested in accessing the Support Cases data.

Manufacturer Invoicing: Design Steps 1 - 3



1. Identify the Business Process:

Invoicing

2. Identify the Grain:

1 Row per Invoice Line Item

3. Identify the Dimensions:

*Requested Ship Date, Ship Date, Product,
Warehouse, Customer Ship-To, Invoice #*

© 2004 Kimball University. All rights reserved.



- ❑ There is tremendous power and resiliency in granular fact table data. It is a myth that users only need to view aggregated information for analysis. Although analytical users won't necessarily want to look at a single invoice, order or claim, it's impossible to predict all the ways in which they'll want to summarize that bedrock data. The most granular data must be structured dimensionally. It is impractical to drill through dimensional summary data, then hit a wall or disconnect when users want to access the detailed data.

Manufacturer Invoicing: 3. Identify Dimensions

Customer Ship-To Dimension Table

CUSTOMER SHIP-TO KEY

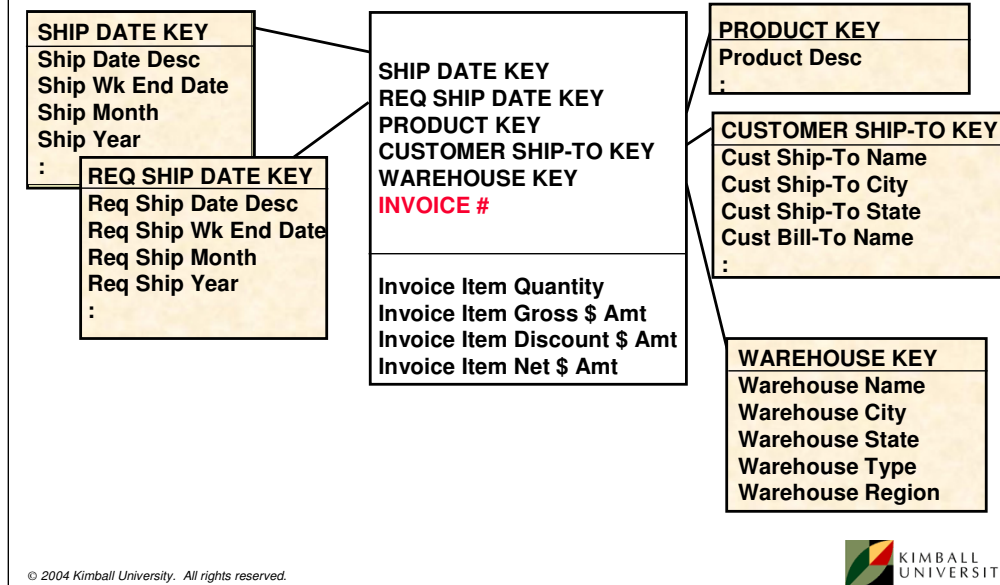
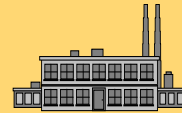
Cust Ship-To Name
Cust Ship-To City
Cust Ship-To State
Cust Ship-To Zip
Cust Bill-To Name
Cust Bill-To City
Cust Bill-To State
Cust Bill-To Zip
Cust Corporate Name
Sales Team
Sales District
Sales Region
:

© 2004 Kimball University. All rights reserved.



- ❑ There are multiple hierarchies within the Customer Dimension:
 - Natural physical geography of city, county, state, zip
 - The address is not analytical, but it may be advisable to include it in the schema for “one-stop shopping” for your users or to “close the loop.”
 - Organization hierarchy of ship to, bill to, and corporate entity
 - Try to preserve ship-to and bill-to relationships in single dimension.
 - If there are multiple bill-to customers per ship-to, then you must split ship-to and bill-to into separate dimension tables with two foreign keys joined back to the fact table.
 - Sales organization hierarchy
 - Appropriate if most shipments go through “assigned” sales territory.
 - Would need to split sales organization into a separate dimension if users want to track the person responsible for sale. In this case, you could also track the “current sales rep assignment” in the Customer dimension.

Manufacturer Invoicing Schema



- ☐ Invoicing transaction schemas typically have an extremely rich set of facts associated with the business process.
- ☐ Notice that we didn't try to further normalize the fact table by specifying a generic amount and then associating each row with a Amount Type dimension. Besides adding an enormous number of rows to the fact table, this design negatively impacts both performance and usability whenever users want to look at multiple amounts concurrently.
- ☐ Facts of different granularity:
 - For example, the invoice may identify shipping costs for the entire invoice, rather than allocating it down to the line item level for each product. In this case, we'd need two fact tables, one at the line item grain and a second at the invoice level.
- ☐ Multiple currencies:
 - Two sets of facts representing both units of currency should be presented to the user to reduce the risk of human calculation error. You can either physically store both sets of facts, or you can store one set, plus the conversion factor at that point in time, and use a view to compute and present the second set of facts.
 - Different business functions may want to look at the shipment quantities using varying units of measure, such as shipping cases, consumer units or some equivalized unit of measure to support summing or comparison of different sized products.
- ☐ Miscellaneous flags:
 - There are often a slew of miscellaneous flags associated with each invoice line item. You wouldn't want to add 5-10 more dimensions to your schema. On the other hand, you don't want to just include a cryptic textual flag on the fact table. Unfortunately, you can't just ignore them. The "junk dimension" includes all the combinations of miscellaneous flag values. This solution gives the users the ability to slice-and-dice on these less frequently analyzed flags and indicators, without adding unnecessary foreign keys to the fact table.

Workshop: Design Steps 1 - 3



1. Identify the Business Process:
A. Monthly Account Balance Snapshot

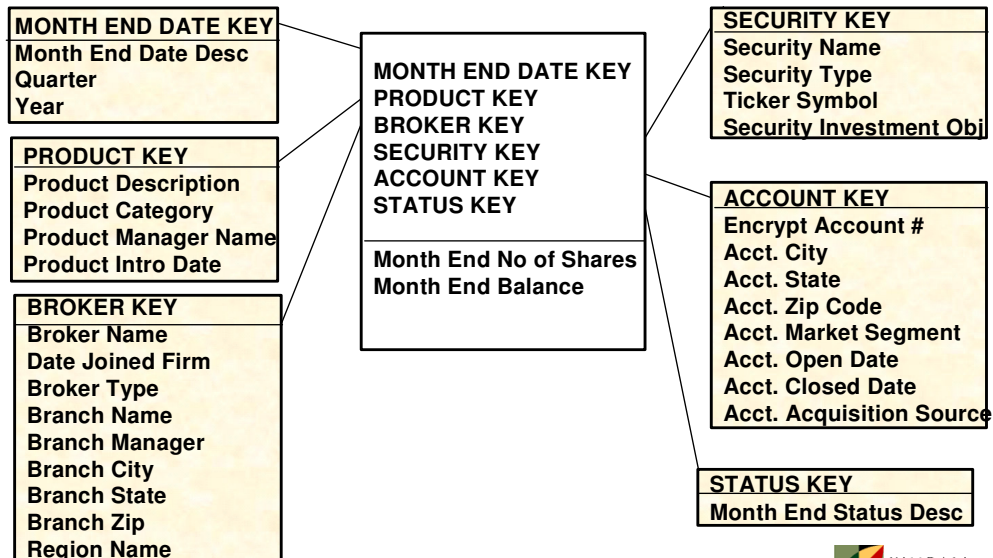
2. Identify the Grain:
A. 1 Row per Account, Security and Month

3. Identify the Dimensions:
A. Month-End Date, Security, Account, Product, Broker and Status

© 2004 Kimball University. All rights reserved.



Retail Brokerage Monthly Snapshot Schema



© 2004 Kimball University. All rights reserved.



Workshop: Design Steps 1 - 3

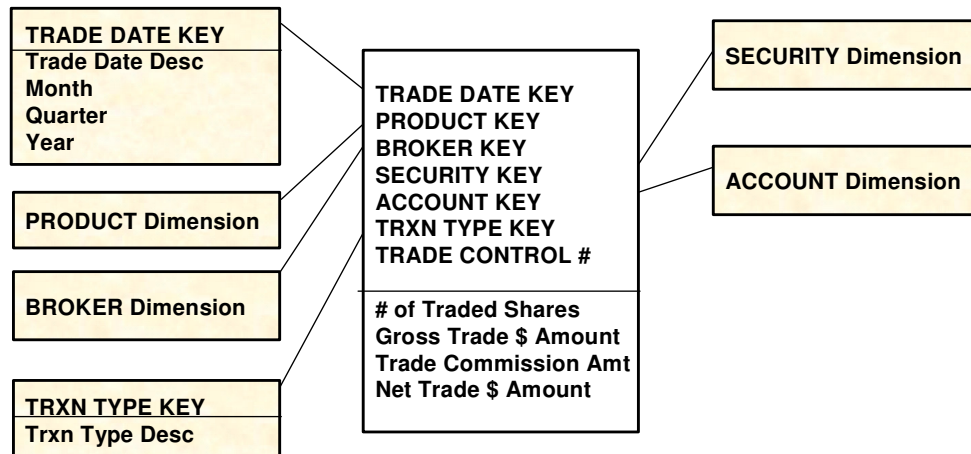


1. Identify the Business Process:
B. Trade Activity Transactions
2. Identify the Grain:
B. 1 Row per Transaction
3. Identify the Dimensions:
B. Trade Date, Security, Account, Product, Broker and Transaction Type

© 2004 Kimball University. All rights reserved.



Retail Brokerage Trade Transaction Schema



© 2004 Kimball University. All rights reserved.

