KIMBALL GROUP
Consulting | Kimball University

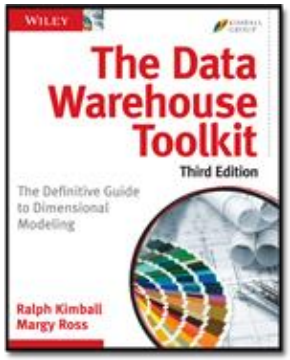
ONE HARD PROBLEM

Multivalued Dimensions

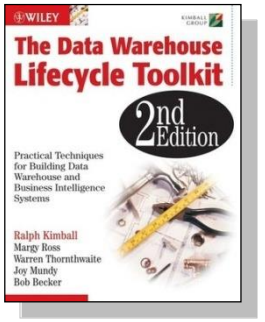
About Joy and Kimball Group

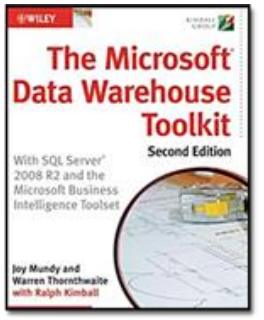
- Kimball Group founded by Ralph Kimball
- Tiny DW/BI consultancy
  - Consulting (requirements, design, and architecture)
  - Writing
  - Teaching and speaking
- Kimball Group retires at end of 2015 (tick tock)
  - Content will remain available
  - Website plus upcoming Kimball Reader, Second Edition
- 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, 3rd Ed.
    - R. Kimball, M. Ross (Wiley 2011)
  - The Data Warehouse Lifecycle Toolkit, 2nd Ed.
    - R. Kimball, M. Ross, W. Thornthwaite, J. Mundy, B. Becker (Wiley 2008)
  - The Microsoft Data Warehouse Toolkit, 2nd 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





KIMBALL UNIVERSITY

KIMBALL GROUP

Agenda: One Hard Problem

- Introduction
  - Dimensional basics
  - Multivalued or many-to-many dimensions
- Introducing multivalued dimensions
  - Examples of multivalued problems
- Alternative design approaches
- Presenting and using the multivalued dimension

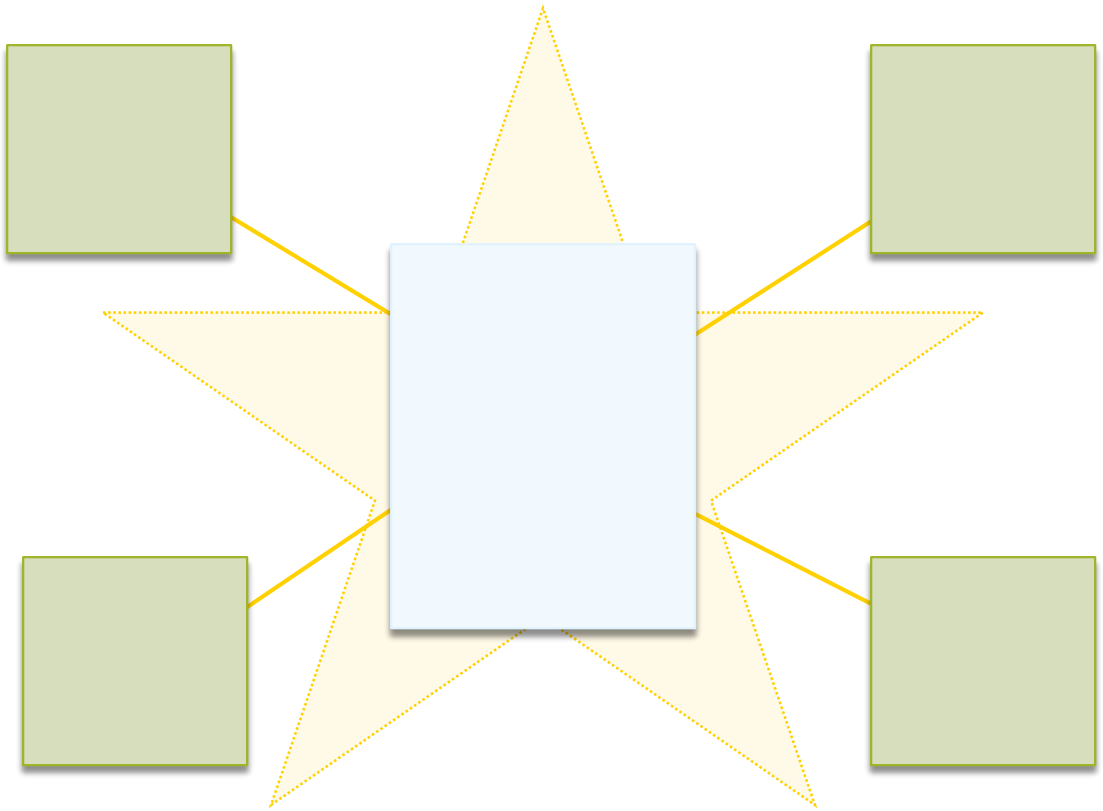
Basic Dimensional Modeling Concepts

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

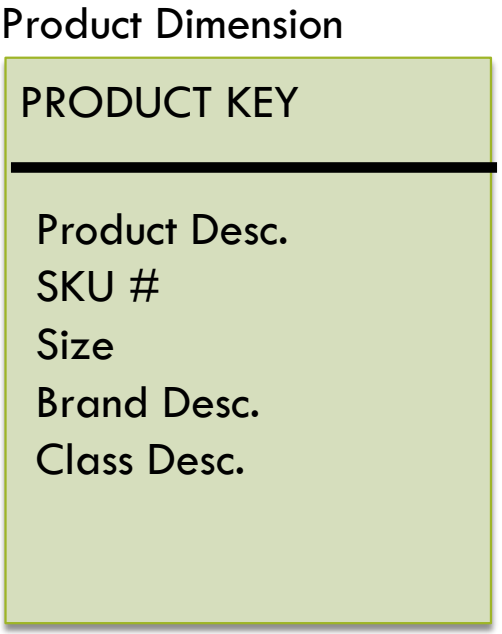


Dimensional: Why and how

- 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



KIMBALL GROUP

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

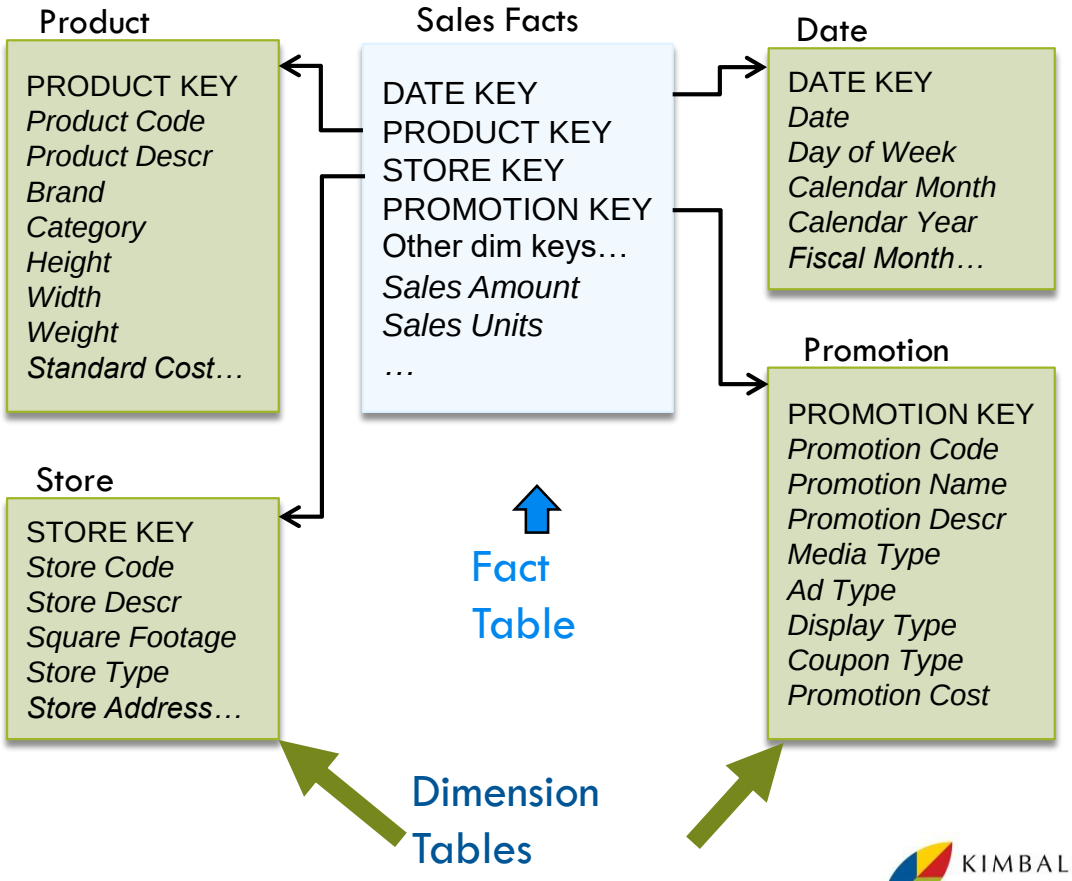
DATE KEY
PRODUCT KEY
STORE KEY
PROMOTION KEY
Other dim keys…

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



KIMBALL GROUP

Terminology: Dimension Table Surrogate Keys

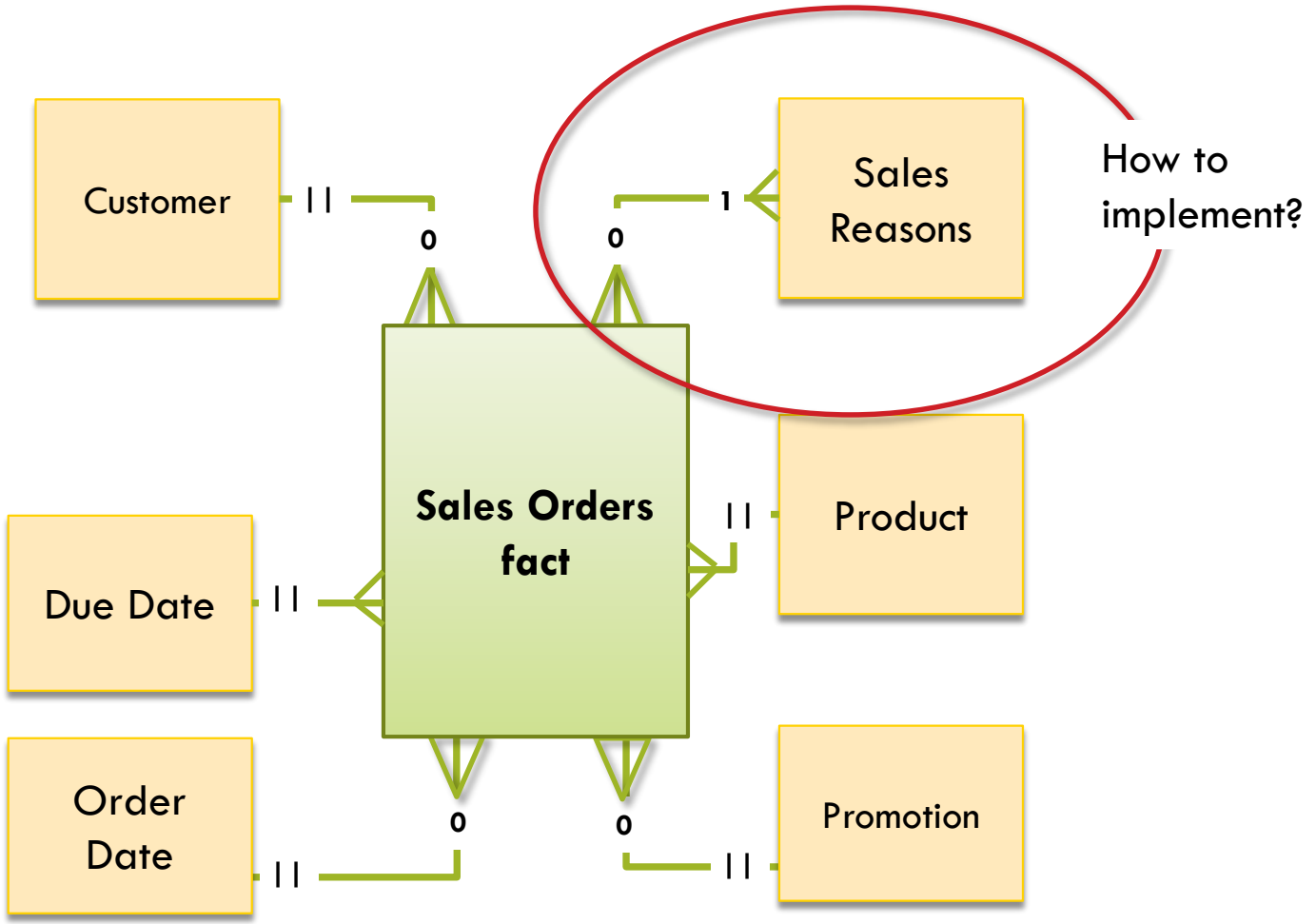
- Surrogate keys are substitute keys
  - Integer, non-meaningful, sequence numbers
  - Surrogate keys join fact and dimension tables
  - Treat business keys as attributes (aka natural keys)
- Benefits
  - Isolate DW/BI system from operational changes
  - Improve performance (over character and 2-col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

Introducing Multivalued Dimensions

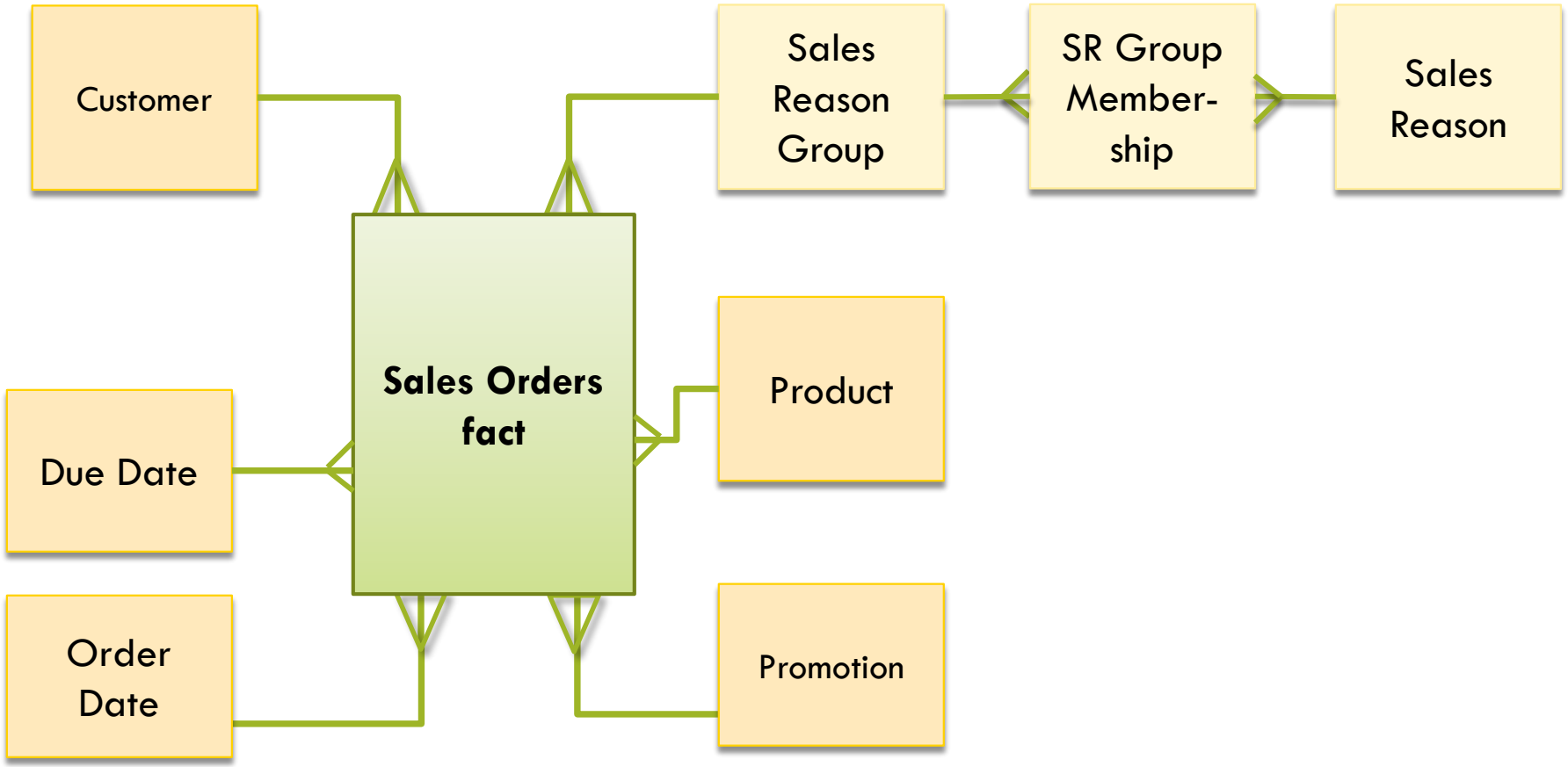
Introducing multivalued dimensions

- Each row in the fact table corresponds to multiple dimension rows
  - Customer-supplied "sales reasons": Customer tells us why they bought each item on their order
    - Price, Color, Size, Recommendation, …
    - Web form allows multiple reasons
- Challenges:
  - Need to allow ad hoc use (must be easy)
  - Need good query performance for analytic queries

Multivalued Sales Reason -- Kind of what we want



Multivalued dims: Bridge table solution



Sales Reason Dimension

Sales Reason	Approx 20 reasons (rows)
Sales Reason Key	DW Surrogate Key
Sales Reason Code	Code from source system
Sales Reason Descr	Label
Sales Reason Group	Often there's a grouping

- Absolutely vanilla – nothing unusual about this dimension
- Design can support SCD-2, though in the current example (sales reasons) it probably wouldn't

Sales Reason Group

	Sales Reason Group	<= 2^20 = 1 Million (-ish)
PK	Sales Reason Group Key	DW Surrogate Key
		May have additional columns to help ETL… will return to discuss

- One row for each theoretical (or observed) combinations of sales reasons. Max is 2^20 in this specific example.
- The practical maximum count of rows in the bridge table is the # of rows in the fact table.
- Issue affects ETL, not the data model.

Sales Reason Group Membership Bridge Table

	Sales Reason Group Membership	< 10 Million (-ish)
PK, FK	Sales Reason Group Key	DW surrogate key
PK, FK	Sales Reason Key	DW surrogate key
	Reason Group Size	How many reasons are in this reason group?
	Allocation	Often 1/[Reason Group Size]. Or maybe you can get a rule from the business.

- If a customer chose 3 reasons, the group membership table has 3 rows for that sales reason group
- This table is significantly larger than Sales Reason Group (which has one row for each grouping)

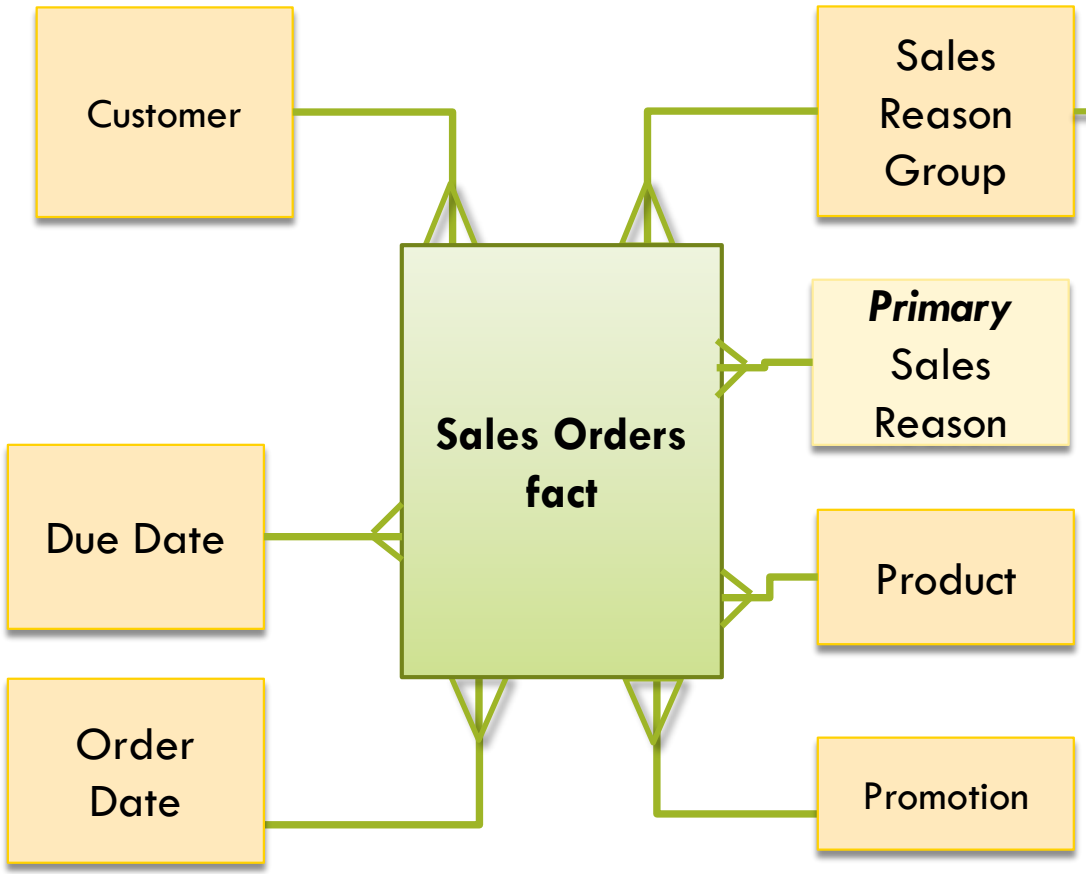
Multivalued Dimension Challenges

- Query performance
  - Bridge table is big. As big as – or bigger than – the fact table.
  - Always make it as small as possible (ETL section)
  - Combining the dimension (eg Sales Reason) with bridge table doesn't help much if at all.
    - Eliminate a join
    - List of value queries go against a huge table rather than a tiny table.
- Usability: Double counting
  - Consider an aggregate query of Sales Amt by Sales Reason
  - There is no great solution to this problem – user education is required

Avoiding the Bridge Table

- Identify a “primary reason”
- Pivot out the sales reasons
- Add a concatenated column
- Change the grain of the fact table

Identify Rule for Primary Reason



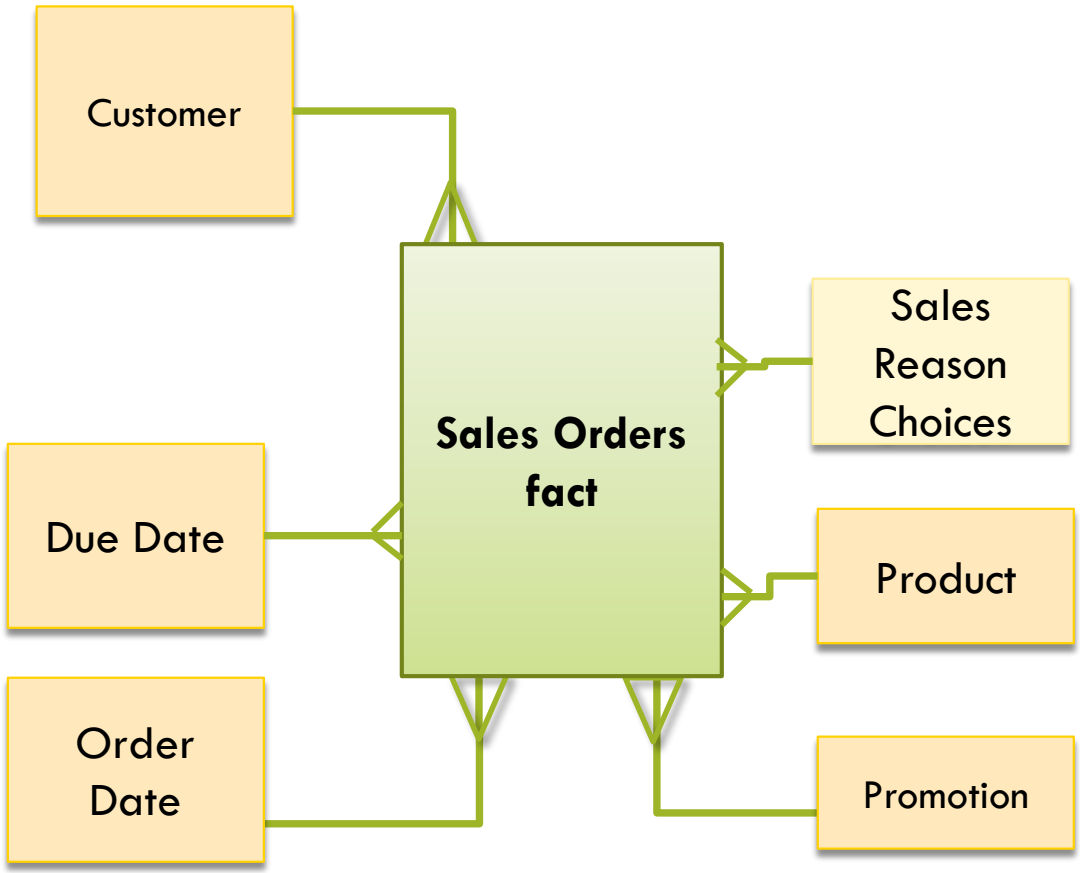
- Business users provide logic
- Or change website so we collect from customers
- Often a ***supplement*** to the bridge table approach
- Lots of queries can skip the multivalued dim

Pivot out the Sales Reasons

	Sales Reason Choices	<= 2^20 = 1 Million (-ish)
PK	Sales Reason Choices Key	DW Surrogate Key
	Is On Sale	Yes / no or decode eg "Chose sale"
	Is Price	Ditto
	Is Recommendation	Ditto
	Is …. (other 17 reasons)	Ditto x 17

- One row for each theoretical (or observed) combinations of sales reasons. Max is 2^20 in this specific example.
- Clearly, populate only with observed combinations. This table will be <= size of fact table (usually considerably less).

Pivot out the Sales Reasons

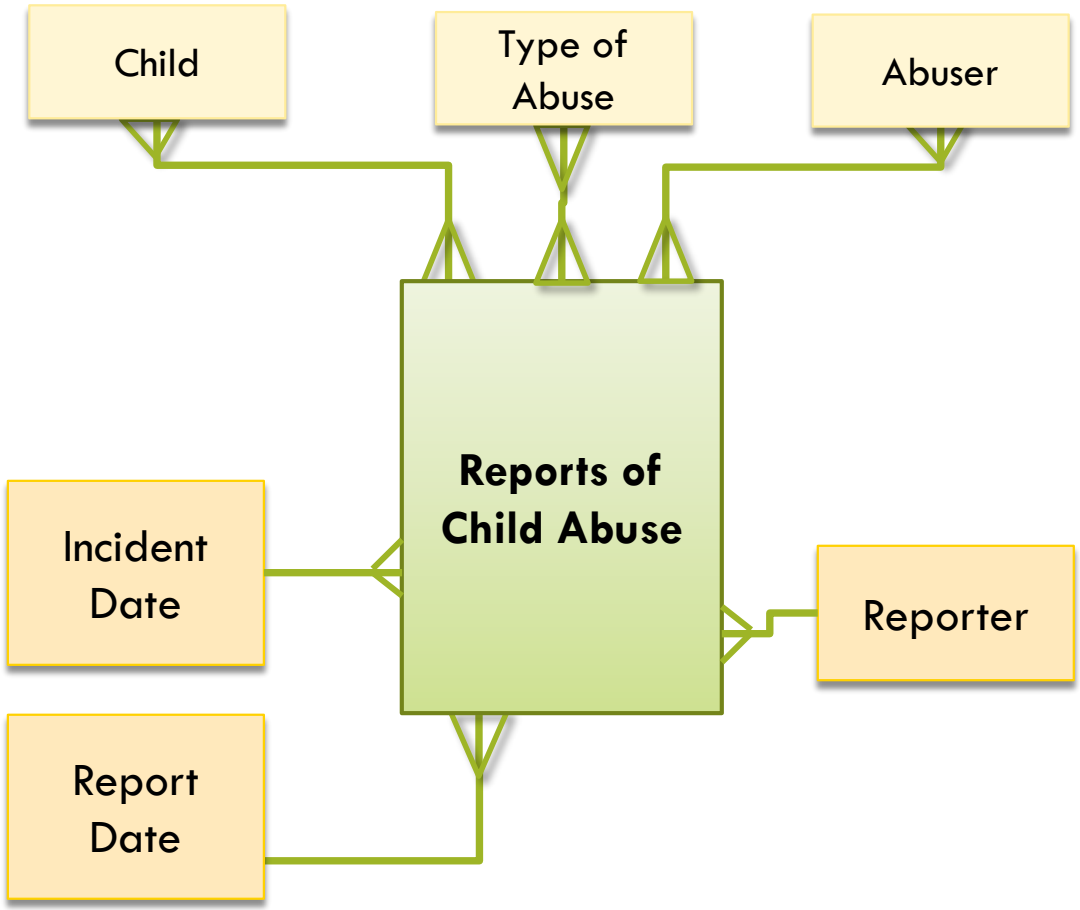


- Sales Reason Choices table will be significantly smaller than bridge table
- But it's still big
- Good choice for a relatively static form, multiple choice
- Only populate with ***observed*** sets of choices
- Not very resilient to change
  - What if the form is redesigned? More columns!!

Add a concatenated column

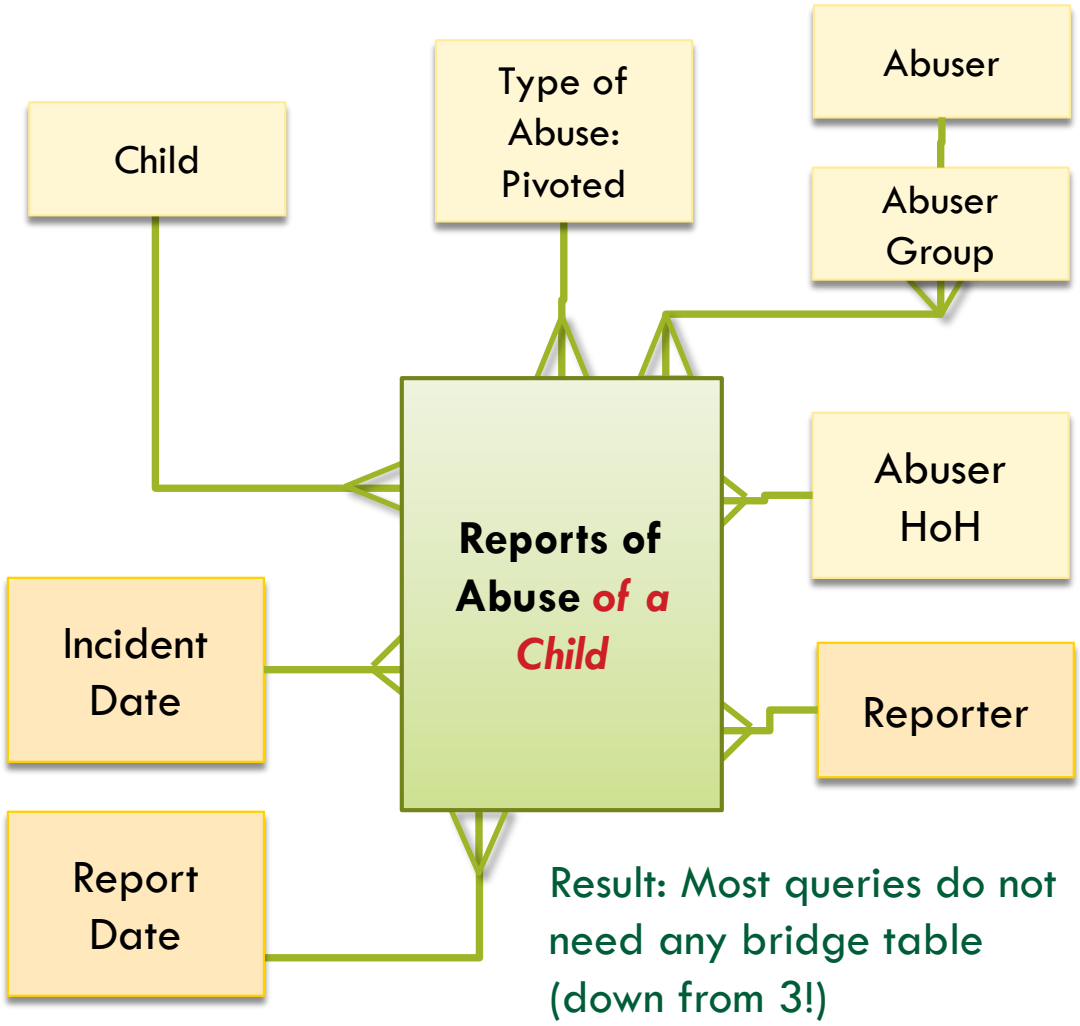
- On sale | Price
- Most appropriate for multi-valued attributes of a dimension
  - Rather than multi-valued dimension relationships as we've mostly been discussing
- Sometimes can supplement the "pivot" approach

Change the Grain of the Fact Table



- Child Protective Services schema (greatly simplified)
- Business process in the fact table focuses on reports of child abuse
- Fact table grain was one row per report of child abuse
- Potentially multiple:
  - Children per incident
  - Types of abuse per incident
  - Abusers

Change the Grain of the Fact Table



- Fact table grain: One row for each report of abuse against a child
  - Designers were too focused on the source system!
  - The children are clearly what's important
- Types of abuse (some 30 types, multi-choice) – use the pivoting technique discussed previously
- Multiple abusers – use HoH + bridge table techniques

Other Twists on Multivalued Relationships

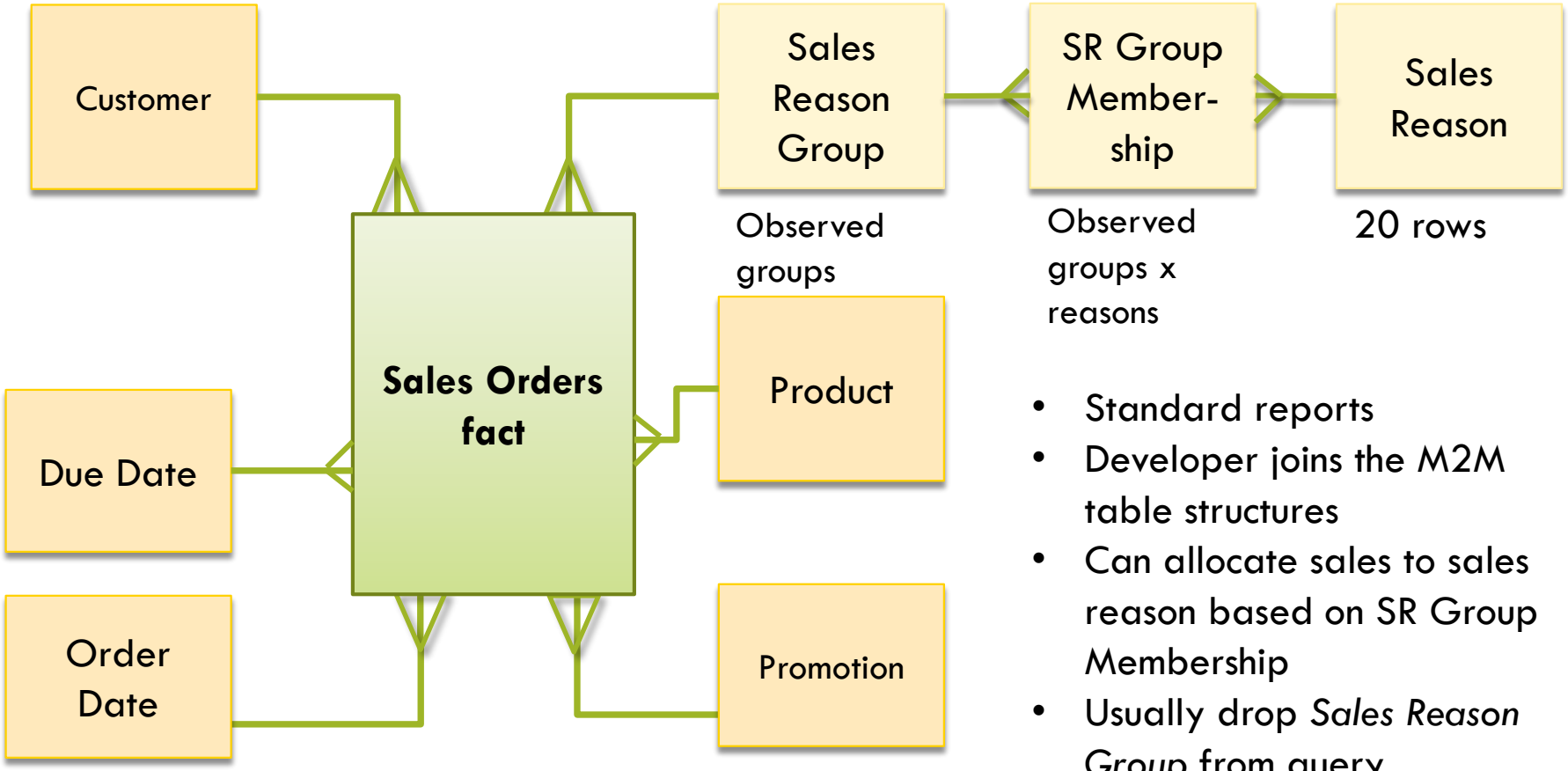
- Many-to-many between dimensions
  - Account snapshot schema: Bank accounts to Customers (me, my husband, joint)
  - Higher Education: Students and their majors
    - If there are only a handful of possibilities, jam them into the dimension as Major1..Major3
    - Imperfect, awkward, but better than the bridge table!
- Does order count?
  - Medical diagnoses

Key Lessons for the Design

- Try to avoid the "correct" (bridge table) solution!
- If you must build a bridge table, populate it only with observed groups, not all theoretically possible groups

Presentation and Usability

Querying Directly from Relational



- Standard reports
- Developer joins the M2M table structures
- Can allocate sales to sales reason based on SR Group Membership
- Usually drop *Sales Reason Group* from query

Building OLAP Cubes with Multivalued Dimension

- Some "old fashioned" OLAP tools can consume the correct table structure
- Calculated measure to allocate facts greatly helps usability
- Scalability!
  - All queries of the multivalued dimension go through the (potentially very large) relationship bridge table
  - Minimize the size of the bridge table
  - Limits… always relative

Using multivalued dimensions in in-memory OLAP

- Tableau, Qlikview, Analysis Services Tabular, etc
- Highly problematic
  - In other words, can't do it
- Recommend the workaround approaches for popular data visualization tools

Key Lessons for Multivalued Dimensions

- The intellectually correct design (bridge tables) is problematic:
  - Query performance
  - Usability (double-counting)
  - Avoid "correct" design if possible
- Always coalesce / squish the bridge table
  - Do it in ETL. Do it once. Do it right.
- Bridge table in relational is fine for predefined reports
  - For ad hoc, effective use requires training, or old-fashioned OLAP

KIMBALL GROUP

Consulting | Kimball University

THANK YOU