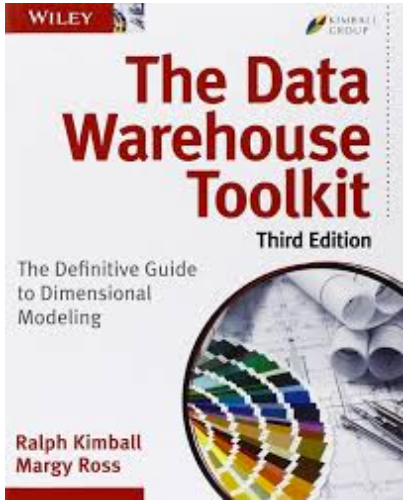
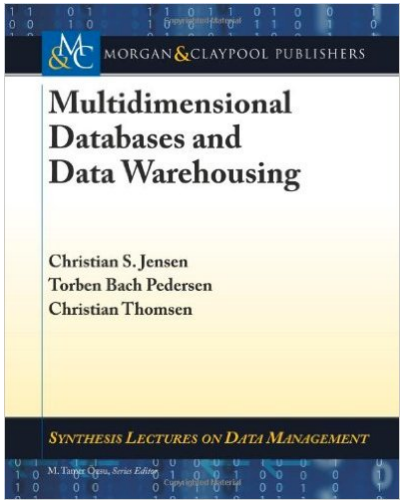
Resources

these slides cannot replace the textbooks by any means !

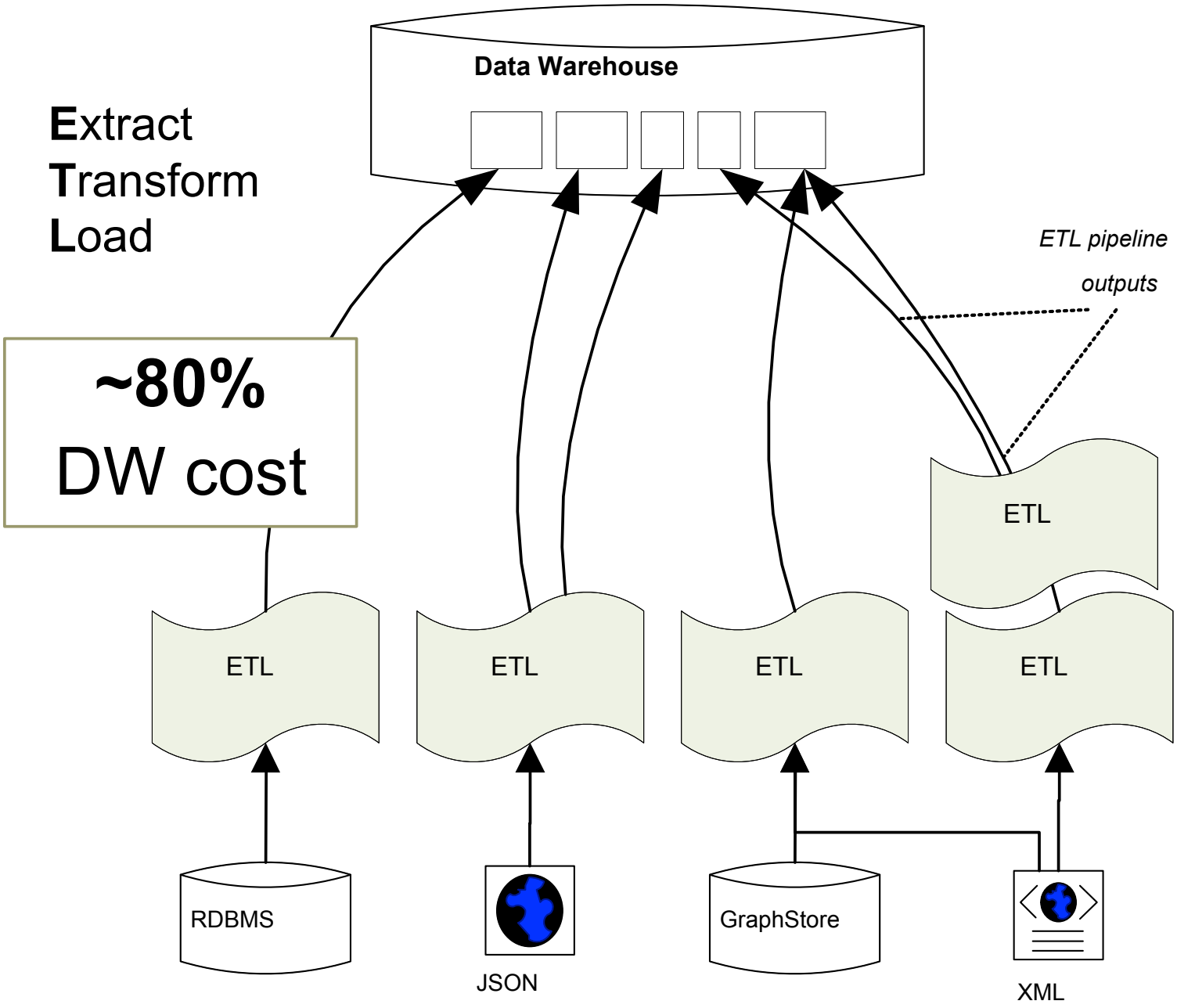
- Entrepôts de données, guide pratique de modélisation dimensionnelle. R.Kimball, M.Ross





- Multidimensional databases and Data Warehousing C.S. Jensen, T.B.Pedersen and C.Thomsen

The Art of Designing a Datawarehouse : the Retail Case Part 1


Extract
Transform
Load
Data Warehouse
ETL pipeline
outputs
~80%
DW cost
ETL
ETL
ETL
ETL
ETL
RDBMS
JSON
GraphStore
XML

Data Integration

http://www.financegirltoronto.com/wp-content/uploads/2017/07/getting-data-right.pdf

Table 1-1. Evolution of three generations of data integration systems

	First generation 1990s	Second generation 2000s	Third generation 2010s
Approach	ETL	ETL+ data curation	Scalable data curation
Target data environment(s)	Data warehouses	Data warehouses or Data marts	Data lakes and self-service data analytics
Users	IT/programmers	IT/programmers	Data scientists, data stewards, data owners, business analysts

Modelling for Data Analysis

- **Key question : which are the most important aspects to model inside a DW** ?

  (talking in terms of business)

- Treat then in order of importance !

- ETL procedures are expensive

- Ressources inside a company should be invested accordingly (*keep that in mind also for your project !!*)

Retail Case Study : Grocery Chain

The case :

- 100 Grocery stores spread over a five-state area.
- Each store has ~60,000 individual products on its shelves; 80% come from outside manufacturers.
- Grocery departments : frozen foods, dairy, meat, bakery, floral, and health/beauty aids.

Retail Case Study : Grocery Chain

Data is collected at

- **cash registers** as customers purchase products
- **the back door**, where vendors make deliveries

- Question #1 : Which is the most important aspect (talking in business terms) to analyze ?

Facts

Walmart
Save money. Live better.

(662) 234 - 9131
MANAGER KENNETH HERRING
2530 JACKSON AVE W
OXFORD MS 38655
ST# 0699 OP# 00006810 TE# 70 TR# 09865

MAGAZINE	007485108422	9.99 X
MAGAZINE	007447001180	9.99 X
MAGAZINE	007148651083	4.99 X
MAGAZINE	007189648584	7.99 X
MAGAZINE	007098934186	11.99 X
MAGAZINE	000928102998	9.95 X
MAGAZINE	007189643401	5.99 X
MAGAZINE	007511000004	4.99 X
MAGAZINE	007549000002	3.99 X
MAGAZINE	004880700005	3.99 X
MAGAZINE	001400514171	1.99 X
MAGAZINE	005511328982	6.99 X
MAGAZINE	007447010214	12.99 X
MAGAZINE	001400514253	9.99 X
MAGAZINE	001400514031	9.99 X
MAGAZINE	001400514002	3.29 X
MAGAZINE	007098910483	12.99 X
MAGAZINE	002710000965	3.99 X
	SUBTOTAL	136.08
TAX 1	7.000 %	9.53
	TOTAL	145.61
	AMEX TEND	145.61

ACCOUNT # **** **** ***1 002 S
APPROVAL # 520903
REF # 304600808648
TERMINAL # 33052894

02/15/13 16:10:54

CHANGE DUE 0.00

ITEMS SOLD 18

TC# 1784 9189 4026 2512 8007 8

"Like" our store on Facebook
Go to local.walmart.com

sale

Step 1) Decide business process(es) to model

- **First dimensional model must have the most impact**
- Here, we want to understand customer purchases
- Vendors delivery is set aside for the moment

Step 2) Declare the Grain

- Data expressed at lowest possible grain of dimensions
  - Eases precise and complex analytic queries

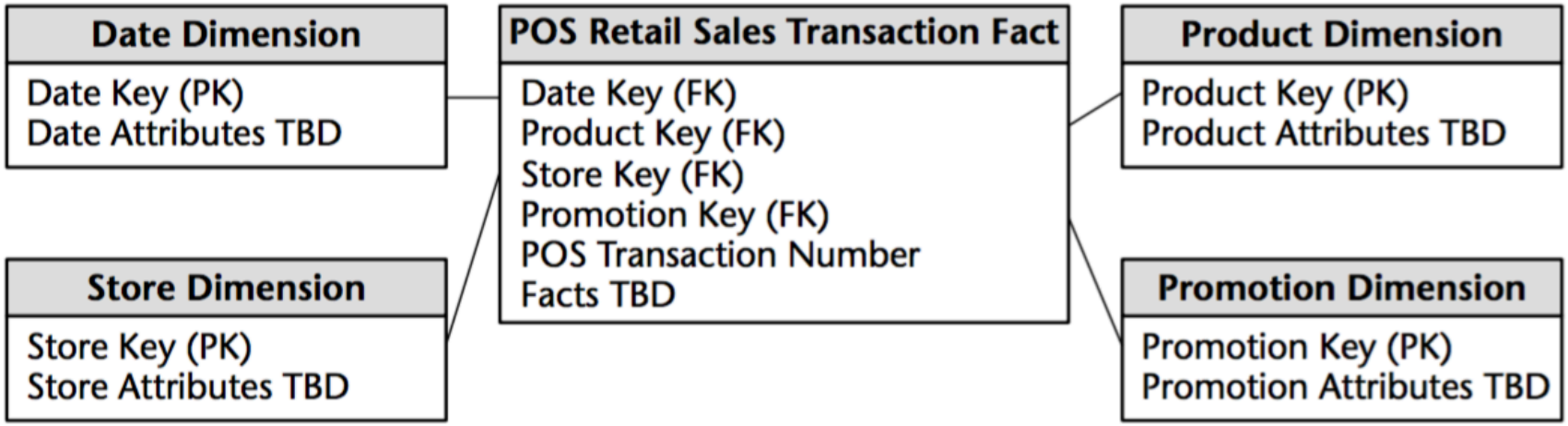
- Here, tree choices for the grain
  - by transaction (too coarse)
  - by item type (just right)
  - by item (too fine, here brings no benefits)

- We choose individual line item type on a selling transaction

Step 3) Choose the dimensions

- Primary dimensions main output of previous step
  - Date, Product, Store, Promotion
- Often possible to add more dimensions later

A first star-schema

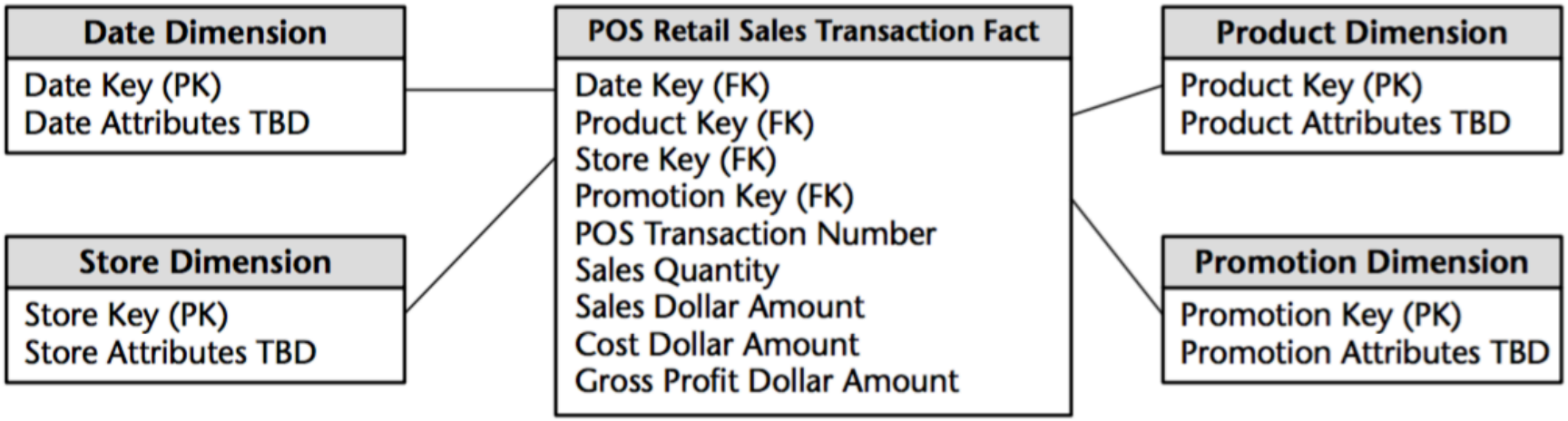


TDB = to be defined yet

Step 4) Identify the Measures

- Selling transactions include
  1. sales quantity
  2. per unit sales price
  3. sales total amount (=1.* 2. , but still included)
- In this case facts are already available

Star-schema : Fact table



Guidelines

Date vs Time Dimension

product

store

city

date

Date Dimension

- The only dimension to be in every datawarehouse

Counterintuitive (at first) but a date is :

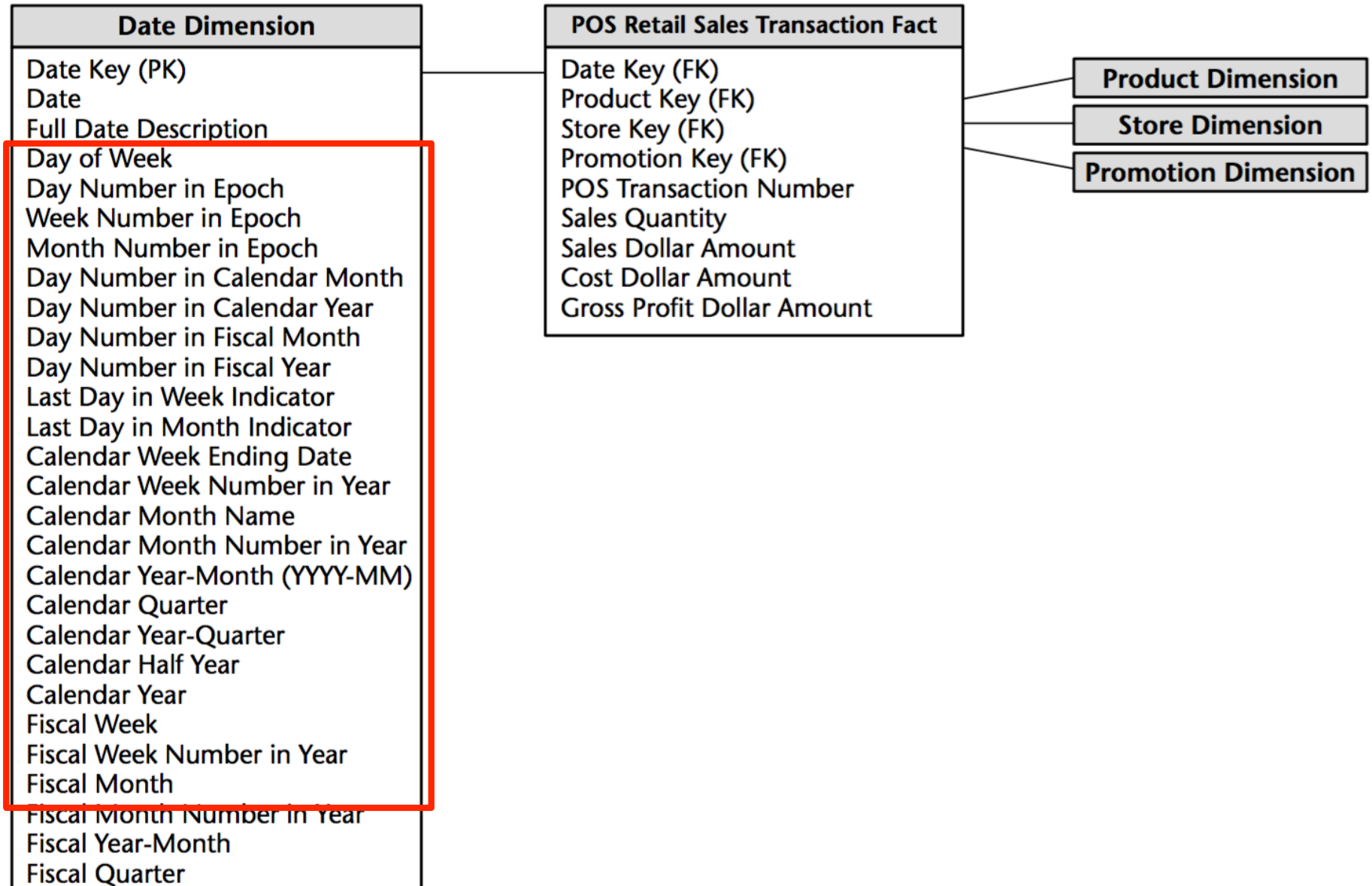
- denoted by a key
- complex and fully described in dimensional table

Date Dimension Table

Date Key	Date	Full Date Description	Day of Week	Calendar Month	Calendar Year	Fiscal Year-Month	Holiday Indicator	Weekday Indicator
1	01/01/2002	January 1, 2002	Tuesday	January	2002	F2002-01	Holiday	Weekday
2	01/02/2002	January 2, 2002	Wednesday	January	2002	F2002-01	Non-Holiday	Weekday
3	01/03/2002	January 3, 2002	Thursday	January	2002	F2002-01	Non-Holiday	Weekday
4	01/04/2002	January 4, 2002	Friday	January	2002	F2002-01	Non-Holiday	Weekday
5	01/05/2002	January 5, 2002	Saturday	January	2002	F2002-01	Non-Holiday	Weekend
6	01/06/2002	January 6, 2002	Sunday	January	2002	F2002-01	Non-Holiday	Weekend
7	01/07/2002	January 7, 2002	Monday	January	2002	F2002-01	Non-Holiday	Weekday
8	01/08/2002	January 8, 2002	Tuesday	January	2002	F2002-01	Non-Holiday	Weekday

- 10 years of rows representing days in date dimension table make only 3,650 rows (very small for a DW)

Date Dimension



Date Dimension

- The **day-of-week** and **calendar-month** columns contains the name of the day, such as Monday, and the name of the month, such as March.

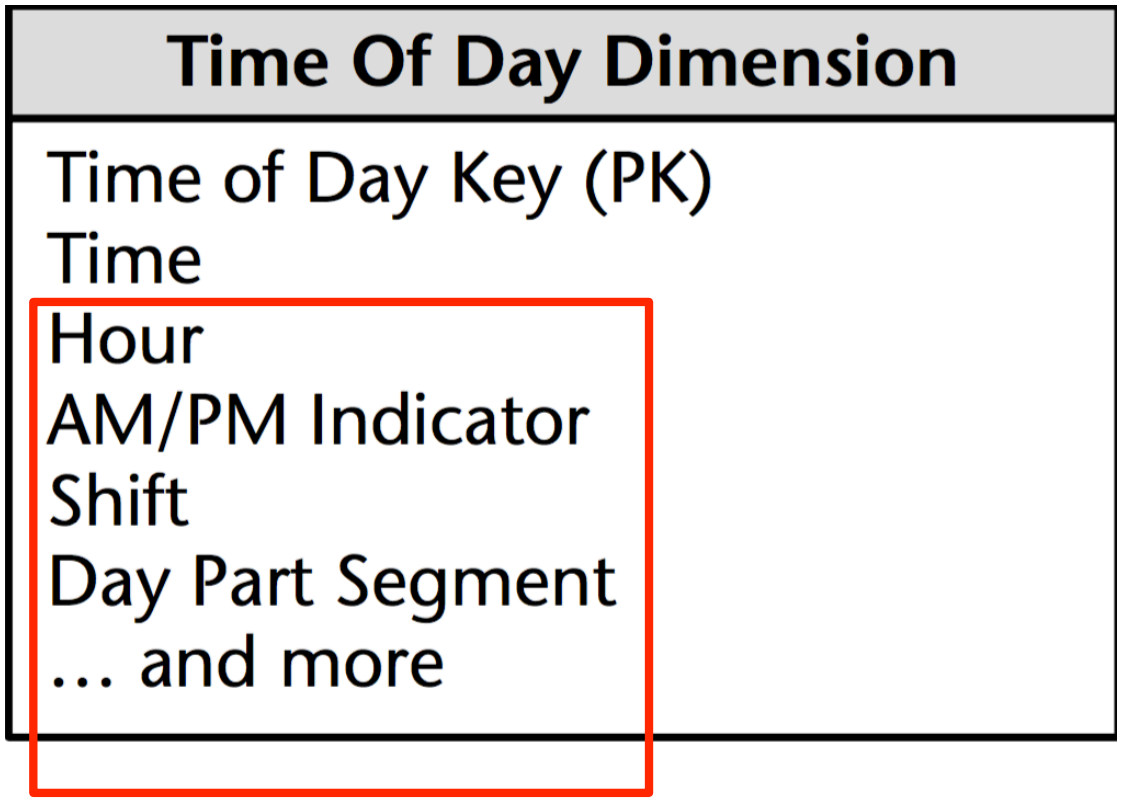
- Used to create reports comparing the business comparing selling in week-days or months

Date ≠ Time

Time Dimension

- Date and time are almost completely independent
- Separate time-of-day dimension, day-part analysis (eg, activity during the evening after-work)
- **Combining date and time in a single dimension would make undesirable cartesian product**
- 3,650-row date & 1,440-row time-of-day by minute better than 5,256,000 date-time rows

Time Dimension



Avoid
Too Many Dimensions

Correlated Dimensions

- **Promotion** : conditions under which a product was sold

- Can include :
  - temporary price reductions,
  - newspaper ads,
  - coupons
  - …

- This dimension is often called a *causal* dimension because it describes factors thought to cause a change in product sales.

Retail Schema in Action

- How to determine if a promotion is effective or not ?

- Weekly sales dollar volume by promotion for the snacks category during January 2002 for stores in the Boston district

Calendar Week Ending Date	Promotion Name	Sales Dollar Amount
January 6, 2002	No Promotion	22,647
January 13, 2002	No Promotion	4,851
January 20, 2002	Super Bowl Promotion	7,248
January 27, 2002	Super Bowl Promotion	13,798

Correlated Dimensions

First solution : one dimension listing ***the*** promotions

- does not work when there is more than 1 promotion per item (multivalued dependency)

- **every line of the fact table is triplicated in this case !**

Second solution : one dimension for each type of promotion

- discouraged : too many dimensions : impossible to index effectively ; space waste in the fact table

price reductions

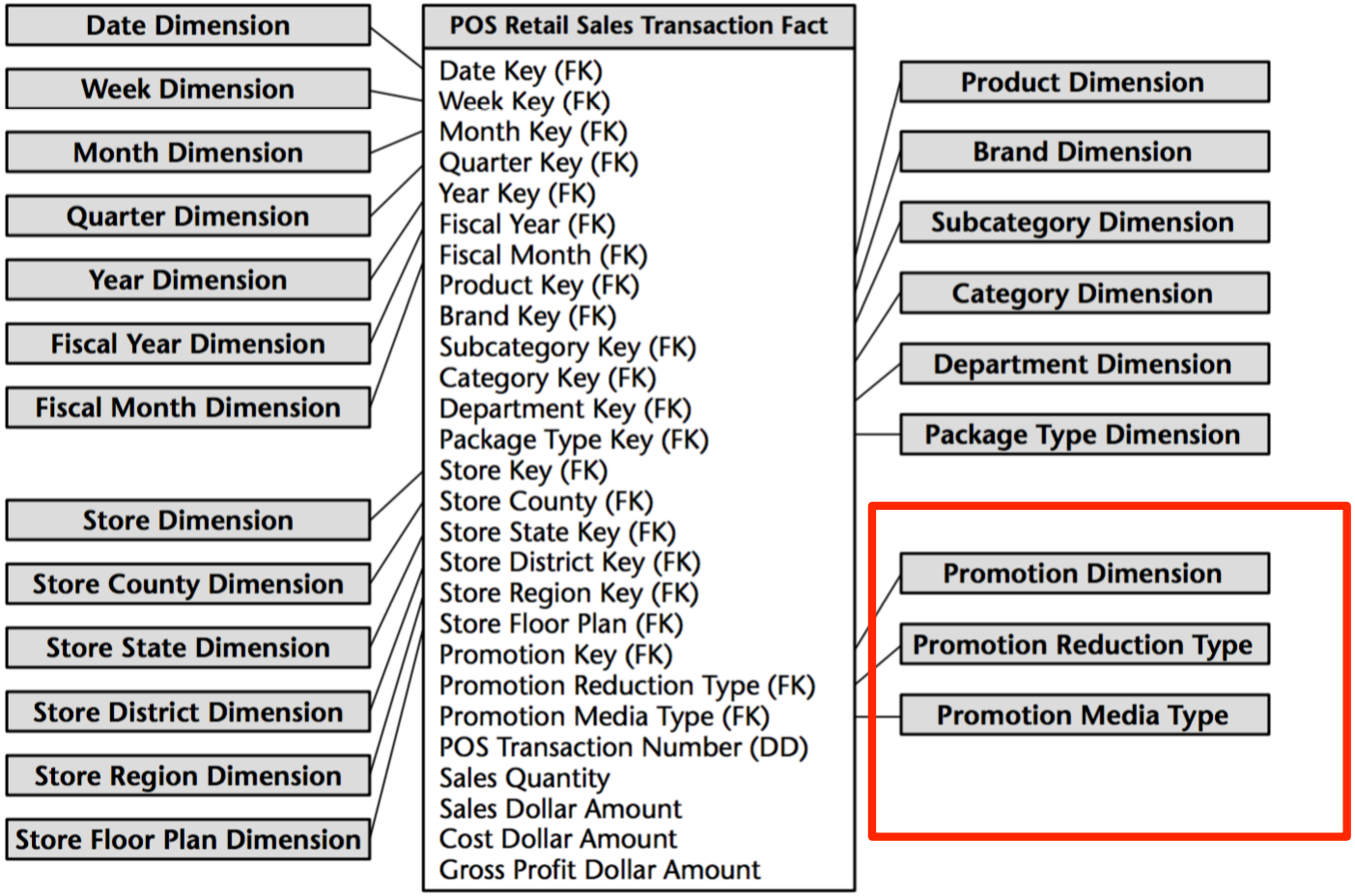
coupons

sale

ads

date

AVOID Too Many Dimensions



centipede fact table

Correlated Dimensions

- **Third** (the) **solution** : correlated dimensions merged into a single dimension

price reductions

coupons

sale

ads

date

price reductions

coupons

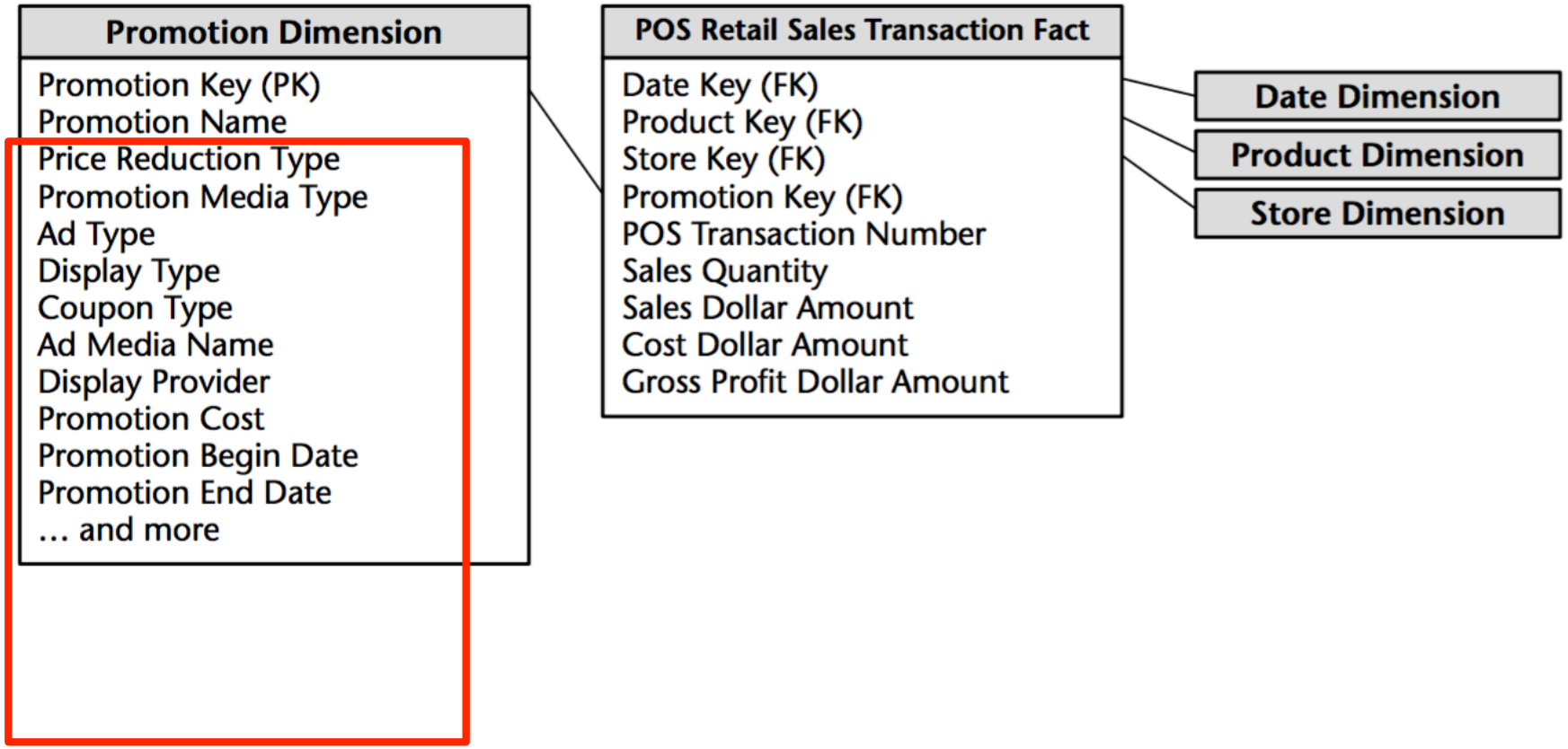
ads

promotion

sale

date

Promotion Dimension



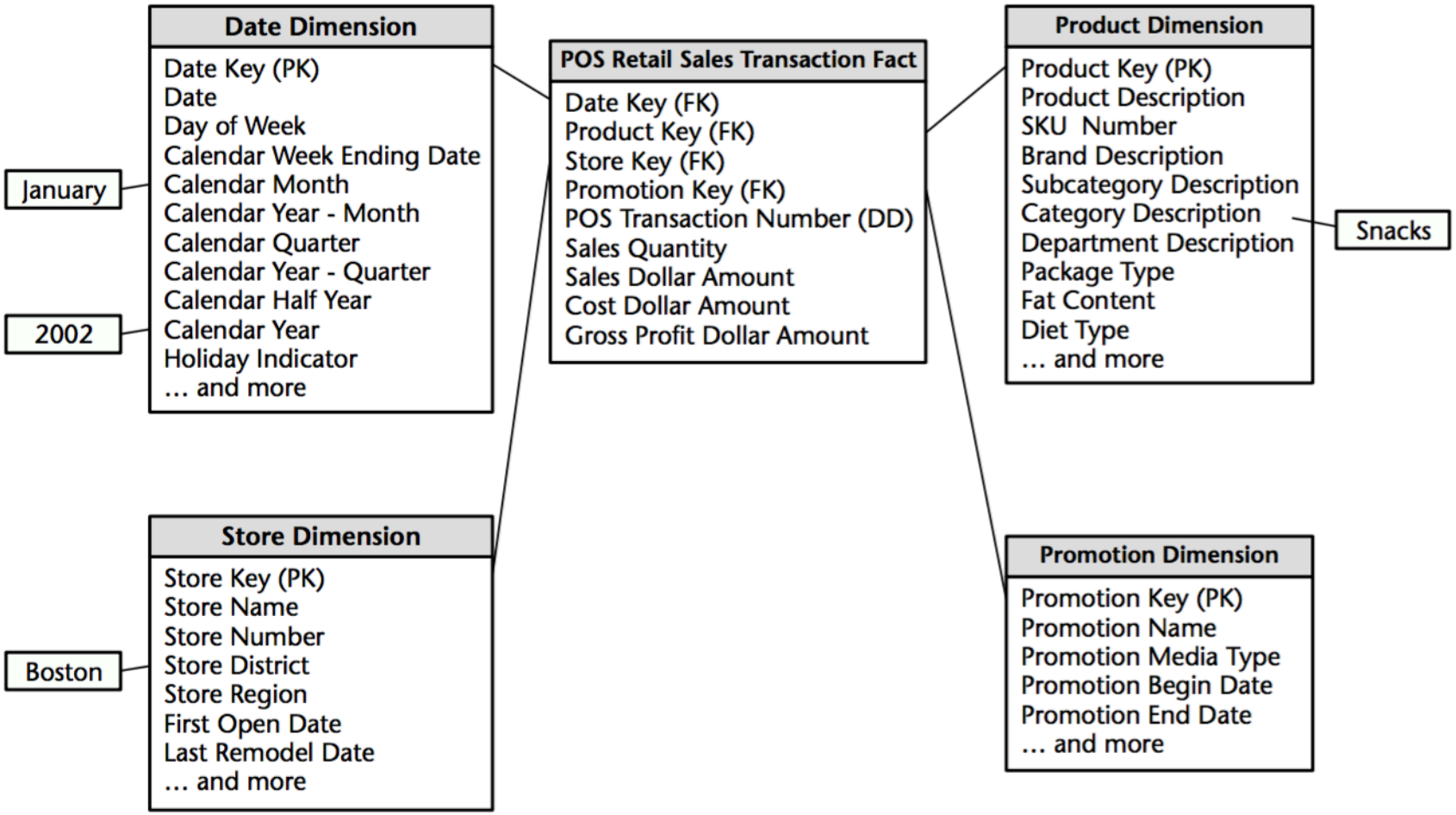
Promotion Dimension

- One row for each combination of promotion conditions
  = exponential table size (in theory)

- **But :** causal conditions are highly correlated:
  A temporary price reduction goes with an ad a
  Coupons often are associated with ads.

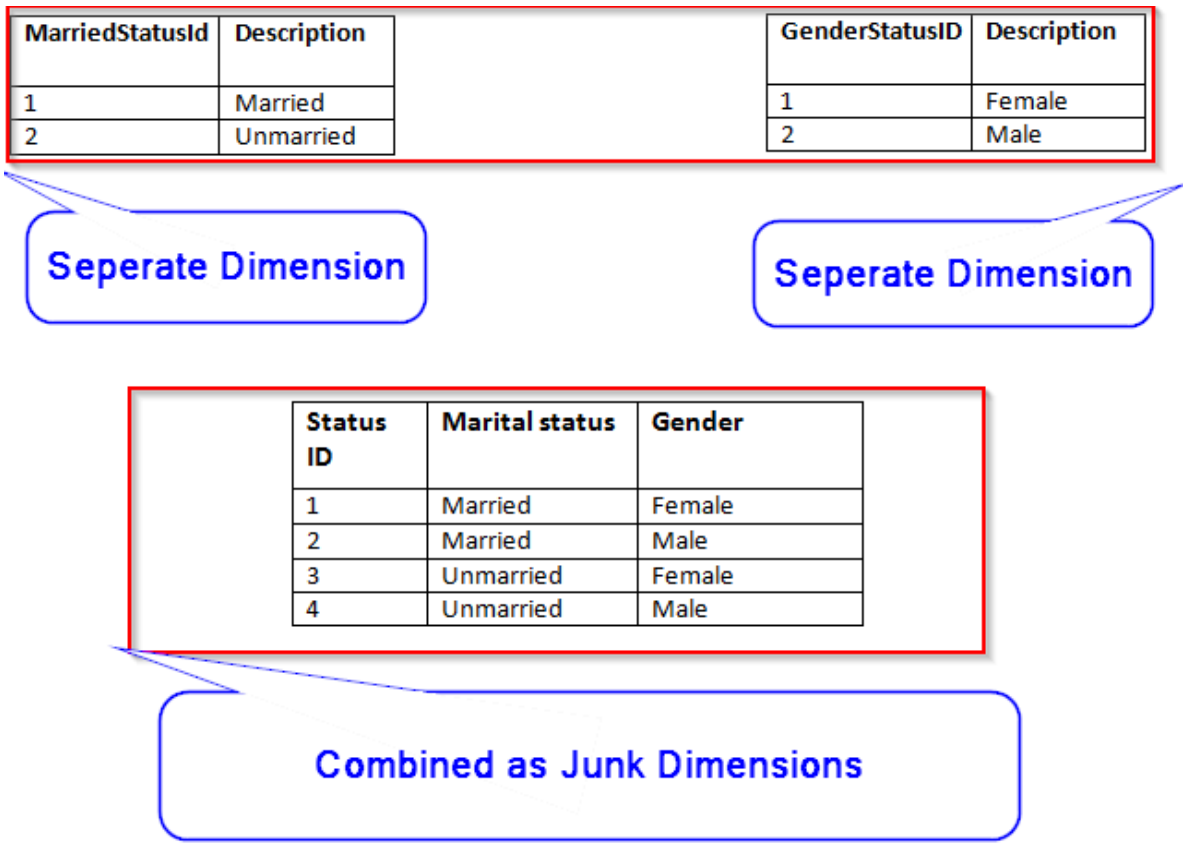
- For 1K ads & 5K temporary price reductions we find 2K combinations (out of the possible 5M): store only these

Retail (Star) Schema



Junk Dimensions

- The result of merging **uncorrelated small** dimensions



Avoid Too Many Dimensions

- Group **correlated** dimensions together
- Group **low-cardinality** **independent** dimensions together (junk dimensions)
- But always verify that there is no real risk of making big cartesian products
- **Do not store combinations that are impossible !!**

Normalization Theory is over

Dimensions : Redundance is OK

- Assume 30 distinct values for department
- If dimension table has 60K lines, each department is repeated (on average) 2K times

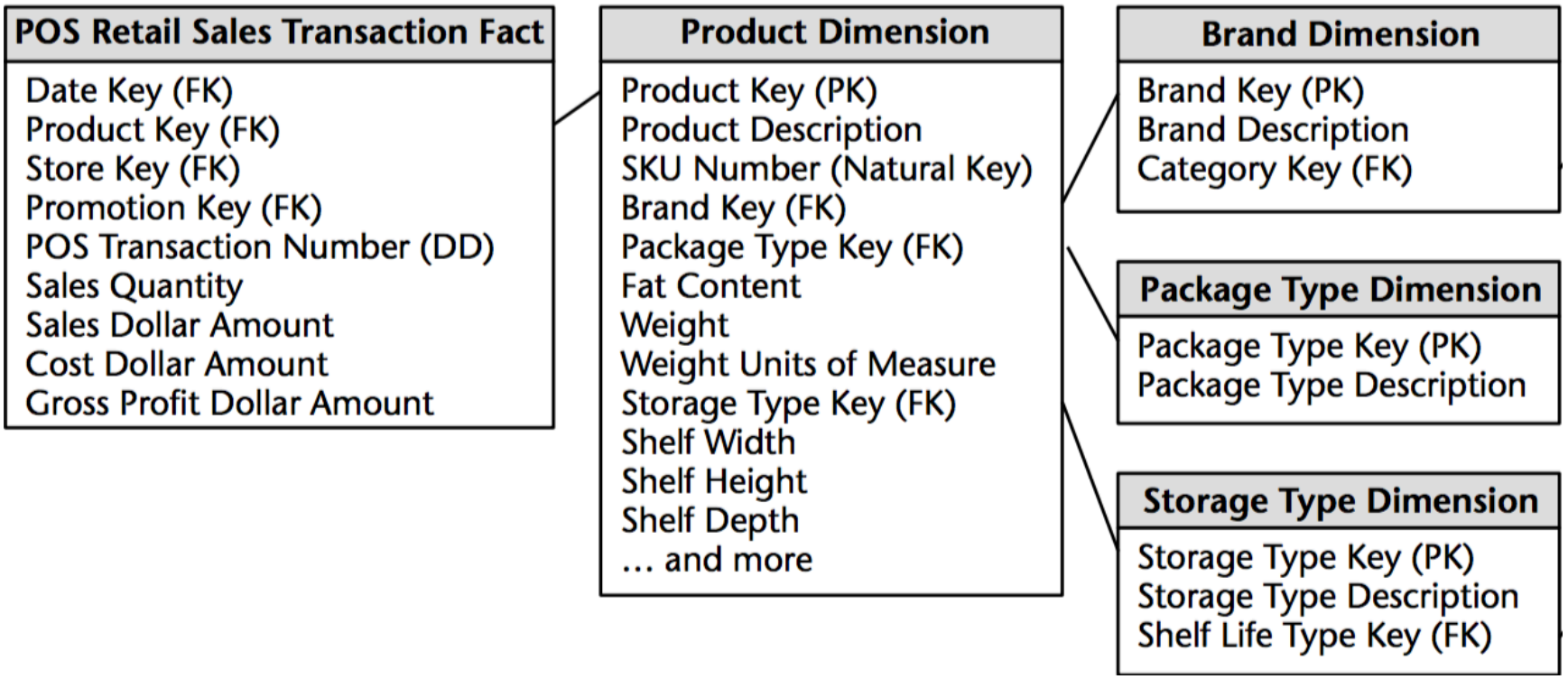
Product Key	Product Description	Brand Description	Category Description	Department Description	Fat Content
1	Baked Well Light Sourdough Fresh Bread	Baked Well	Bread	Bakery	Reduced Fat
2	Fluffy Sliced Whole Wheat	Fluffy	Bread	Bakery	Regular Fat
3	Fluffy Light Sliced Whole Wheat	Fluffy	Bread	Bakery	Reduced Fat
4	Fat Free Mini Cinnamon Rolls	Light	Sweeten Bread	Bakery	Non-Fat
5	Diet Lovers Vanilla 2 Gallon	Coldpack	Frozen Desserts	Frozen Foods	Non-Fat
6	Light and Creamy Butter Pecan 1 Pint	Freshlike	Frozen Desserts	Frozen Foods	Reduced Fat
7	Chocolate Lovers 1/2 Gallon	Frigid	Frozen Desserts	Frozen Foods	Regular Fat
8	Strawberry Ice Creamy 1 Pint	Icy	Frozen Desserts	Frozen Foods	Regular Fat
9	Icy Ice Cream Sandwiches	Icy	Frozen Desserts	Frozen Foods	Regular Fat

Dimension Table Normalization (Snowflaking)

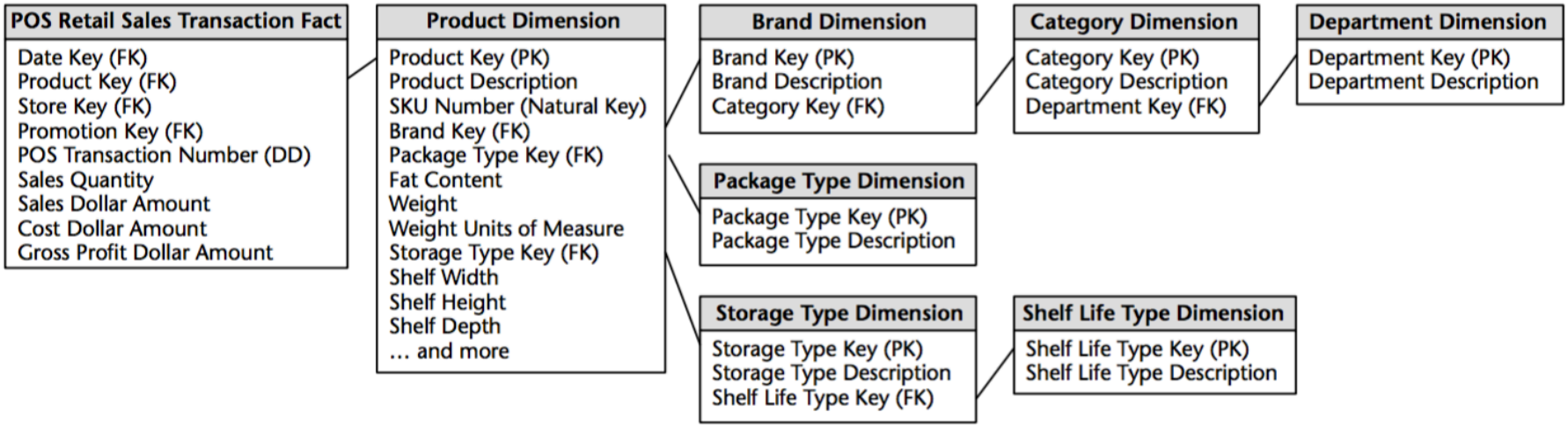
Dimension Table Normalization (Snowflaking)

- Reduces redundancy in storage
- Allows better mainteinance of dimension values

Dimension Table Normalization (Snowflaking)



Dimension Table Normalization (Snowflaking)



Snowflaking is legal but..

- The dimension tables should remain as flat tables physically.
- Normalized, snowflaked dimension tables **penalize cross-attribute browsing (joins required)** and **prohibit the use of bit-mapped indexes** (we'll see that later!)
- **Disk space savings** gained by normalizing the dimension tables typically are **less than 1 percent** of the total disk space needed for the overall DW !
- We knowingly sacrifice this dimension table space in the spirit of performance and ease-of-use advantages.

Recognize Degenerate Dimensions

Walmart
Save money. Live better.

(662) 234 - 9131
MANAGER KENNETH HERRING
2530 JACKSON AVE W
OXFORD MS 38655
ST# 0699 OP# 00006810 TE# 70 TR# 09865

MAGAZINE	007485108422	9.99 X
MAGAZINE	007447001180	9.99 X
MAGAZINE	007148651083	4.99 X
MAGAZINE	007189648584	7.99 X
MAGAZINE	007098934186	11.99 X
MAGAZINE	000928102998	9.95 X
MAGAZINE	007189643401	5.99 X
MAGAZINE	007511000004	4.99 X
MAGAZINE	007549000002	3.99 X
MAGAZINE	004880700005	3.99 X
MAGAZINE	001400514171	1.99 X
MAGAZINE	[illegible]	[illegible] X
MAGAZINE	007447010214	12.99 X
MAGAZINE	001400514253	9.99 X
MAGAZINE	001400514031	9.99 X
MAGAZINE	001400514002	3.29 X
MAGAZINE	007098910483	12.99 X
MAGAZINE	002710000965	3.99 X
	SUBTOTAL	136.08
TAX 1	7.000 %	9.53
	TOTAL	145.61
	AMEX TEND	145.61

ACCOUNT # **** **** ***1 002 S
APPROVAL # 520903
REF # 304600808648
TERMINAL # 33052894

02/15/13 16:10:54

CHANGE DUE 0.00

ITEMS SOLD 18

TC# 1784 9189 4026 2512 8007 8

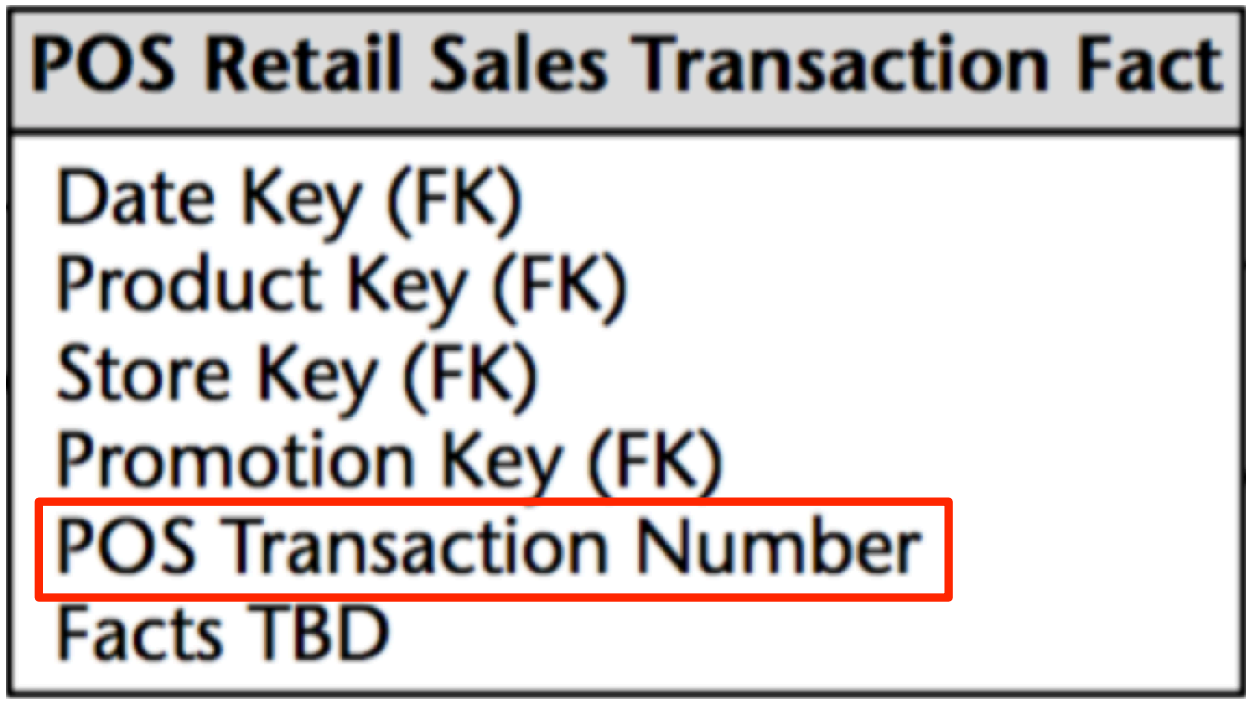
"Like" our store on Facebook
Go to local.walmart.com
02/15/13 16:10:55

CUSTOMER COPY

Candidate granularities

- **By item type**
  - **Precise analysis** of each item type in a purchase
- **By transaction**
  - Just a **summary** of the purchase **(#items, total)**
  - Limited business analysis value
- **By item** (**too fine,** here brings no benefits)

We pick the first



Transaction number is a Degenerate Dimension

- **It is a dimension because it caractherizes the item sold**
- But, although the transaction number may look like a dimension key in the fact table …
- … all the descriptive items of a transaction are in the other dimensions or in the fact table
- Transaction is thus an **empty dimension** that we refer as *degenerate dimension* (DD)

Degenerate = Empty (no attributes)

- Transaction number serves as grouping key for all the products purchased in a single transaction
  - **find items sold together**
  - **compute total amount of a purchase**

- **Order numbers**, **invoice numbers**, almost always appear as degenerate dimensions

- Finally, degenerate dimensions often play an integral role in the fact table's primary key

Use Surrogate Keys

Surrogate Keys

- Business analysts may want to navigate the fact table based on the **operational code** of a product avoiding a join to the dimension table. Eg **2014FR2510d**

- This is not a good idea, as the dimension tables are the entry point of the system

- Surrogate keys are integers that are assigned sequentially as needed to populate a dimension.

- Example Product 1,2,3,4,

Surrogate Keys

- *Every join between dimension and fact tables in the data warehouse should be based on* ***meaningless integer surrogate keys****.*

- Avoid using the natural operational production codes. None of the data warehouse keys should be smart, where you can tell something about the row just by looking at the key.

- This gives better performances