04

Basics of Multidimensional Design

Notice

- **Author**
  - **João Moura Pires (jmp@di.fct.unl.pt)**

- **This material can be freely used for personal or academic purposes without any previous authorization from the author, only if this notice is maintained with.**

- **For commercial purposes the use of any part of this material requires the previous authorization from the author.**

Bibliography

- **Many examples are extracted and adapted from**
  - **The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition) - Ralph Kimball, Margy Ross**

Table of Contents

Four-Step Dimensional Design Process

Four-Step Dimensional Design

- **The design process takes 4 steps:**
  - **Step 1: Select the business process to model.**
  - **Step 2: Declare the grain of the business process.**
  - **Step 3: Choose the dimensions that apply to each fact table row.**
  - **Step 4: Identify the numeric facts that will populate each fact table row.**
- **The design process must take into account**
  - **The analytical requirements**
  - **The available data**

Business Process

- **Step 1: Select the business process to model.**
  - **A process is a natural business activity performed in your organization that typically is supported by a source data-collection system.**
  - **Example business processes include raw materials purchasing, orders, shipments, invoicing, inventory, and general ledger.**
  - **Business processes are not referring to an organizational business department or function.**
    - It is preferable a single dimensional model to handle orders data rather than building separate models for the sales and marketing departments, which both want to access orders data.
    - Avoid data duplication and possible inconsistencies due to data duplication

Declare the grain of the Business Process

- **Step 2: Declaring the grain means specifying exactly what an individual fact table row represents.**
- **Examples of grain declaration:**
  - **An individual line item on a customer's retail sales ticket as measured by a scanner device**
  - **A line item on a bill received from a doctor**
  - **An individual boarding pass to get on a flight**
  - **A daily snapshot of the inventory levels for each product in a warehouse**
  - **A monthly snapshot for each bank account**

Declare the grain of the Business Process

- **The chosen process and the granularity declaration should take into account:**
  - **The available data at source level**
  - **The granularity should be as low as possible since it is always possible to aggregate but not the inverse.**
  - **The granularity has impact on the size of the data base and on the amount of data volume to process daily (assuming a daily update)**
  - **The chosen granularity determines the basic dimensionality**

Dimensions

- **Step 3: Choose the dimensions that apply to each fact table row.**
  - **We want to decorate our fact tables with a robust set of dimensions representing all possible descriptions that take on single values in the context of each measurement.**
    - Ex: Date, Product, Store
  - **The chosen granularity determines the basic dimensionality of a fact table.**
  - **Other dimensions may be added if they are compatible with the granularity**
    - Ex: Promotion, Clerk,
  - **Dimensions to represent Date or/and Time are most always present**
    - Date
    - Time

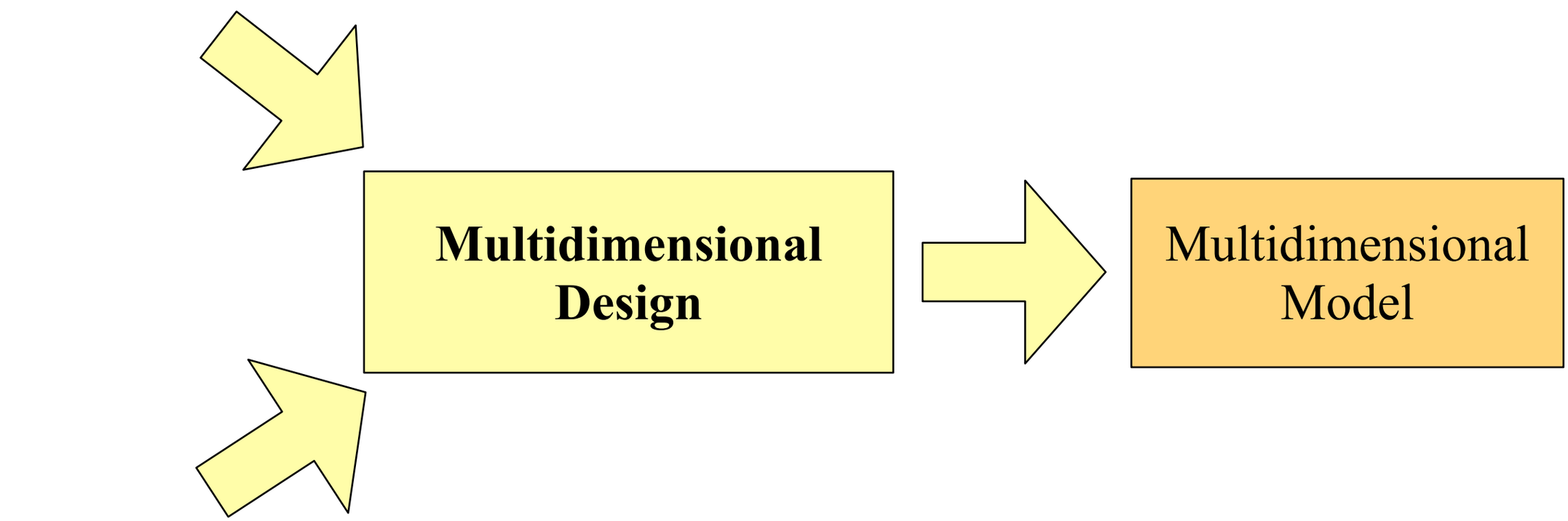
Dimensions

- **Step 3: Choose the dimensions that apply to each fact table row.**
  - **A careful grain statement determines the primary dimensionality of the fact table.**
  - **It is then often possible to add more dimensions to the basic grain of the fact table, where these additional dimensions naturally take on only one value under each combination of the primary dimensions.**
    - If the additional dimension violates the grain by causing additional fact rows to be generated, then the grain statement must be revised to accommodate this dimension.

Numeric Facts

- **Step 4: Identify the numeric facts that will populate each fact table row.**
  - **Facts are determined by answering the question, "What are we measuring?"**
  - **All candidate facts in a design must be true to the grain defined in step 2.**
  - **Facts that clearly belong to a different grain must be in a separate fact table.**
  - **Typical facts are numeric additive figures such as quantity ordered or dollar cost amount.**

Multidimensional Design



Retail Case Study

A large grocery chain

- **A large grocery chain has 100 grocery stores spread over a five-state area.**
- **Each store:**
  - **Has a full complement of departments, including grocery, frozen foods, dairy, meat, produce, bakery, floral, and health/beauty aids.**
  - **Has roughly 60,000 individual products on its shelves. The individual products are called stock keeping units (SKUs):**
    - About 55,000 of the SKUs come from outside manufacturers and have bar codes (Universal Product Codes - UPC) imprinted on the product package. 1 UPCs => 1 SKU.
    - Each different package variation of a product has a separate UPC (separate SKU).
    - About 5,000 SKUs come from internal departments (meat, etc.) - no UPC

Collecting data for the operational system

- **POS (Point of Sale)**
  - **Using the code readers at cash registers as customers purchase products.**
- **The back door, where vendors make deliveries, is another interesting data-collection point.**
  - **Only a small fraction of the deliveries are using code scanners to register them in real time**
- **When the stores pays to suppliers**
  - **Most of times this is the exact confirmation of products being entered in the retail store**
- **Products stock management system**

Main business concerns and goals

- **Management is concerned with the logistics of ordering, stocking, and selling products while maximizing profit.**
  - **The profit ultimately comes from charging as much as possible for each product, lowering costs for product acquisition and overhead, and at the same time attracting as many customers as possible in a highly competitive pricing environment.**
- **Most significant management decisions:**
  - **Pricing**
  - **Promotions (temporary price reductions, ads in newspapers, and newspapers inserts, displays in the grocery store, coupons)**

Designing Multidimensional Model for Retail

Step 1. Select the Business Process

- **The first dimensional model built should be the one with the most impact. It should answer the most pressing business questions and be readily accessible for data extraction.**

- **In our retail case study, management wants to better understand customer purchases as captured by the POS system. Thus the business process we're going to model is POS retail sales.**

- **This data will allow us to analyze what products are selling in which stores on what days under what promotional conditions.**

Step 2. Declare the Grain

- **Preferably you should develop dimensional models for the most atomic information captured by a business process. Atomic data is the most detailed information collected; such data cannot be subdivided further.**
  - **Atomic data is highly dimensional.**
    - The more detailed and atomic the fact measurement, the more things we know for sure. All those things we know for sure translate into dimensions.
  - **Atomic data provides maximum analytic flexibility**
    - It can be constrained and rolled up in every way possible.
  - **A higher-level grain, is limiting ourselves to fewer and/or potentially less detailed dimensions.**
    - Is immediately vulnerable to unexpected user requests to drill-down into the details.

Step 2. Declare the Grain

- **In our case study, the most granular data is an individual line item on a POS transaction.**
  - **Rather than representing transaction line item detail in the dimensional model, we could select sales data rolled up by product and promotion in a store on a day.**

- **A data warehouse almost always demands data expressed at the lowest possible grain of each dimension not because queries want to see individual low-level rows, but because queries need to cut through the details in very precise ways.**

Step 2. Declare the Grain

- **Providing access to the POS transaction information gives us with a very detailed look at store sales:**
  - To understand the difference in sales on Monday versus Sunday.
  - To assess whether it's worthwhile to stock so many individual sizes of certain brands, such as cereal.
  - To understand how many shoppers took advantage of the 50-cents-off promotion on shampoo.
  - To determine the impact in terms of decreased sales when a competitive diet soda product was promoted heavily.
- **None of them could have been answered if we elected only to provide access to summarized data.**

Step 3. Choose the Dimensions

- **A careful grain statement determines the primary dimensionality of the fact table.**
  - **The date, product, and store dimensions fall out immediately.**
  - **We are not interested in the hour (and minute) for each sale.**

- **It is then often possible to add more dimensions to the basic grain of the fact table, where these additional dimensions naturally take on only one value under each combination of the primary dimensions.**
  - **The promotion under which the product is sold.**
  - **Many lines are associated to a sale number (or POS transaction number)**

Step 3. Choose the Dimensions

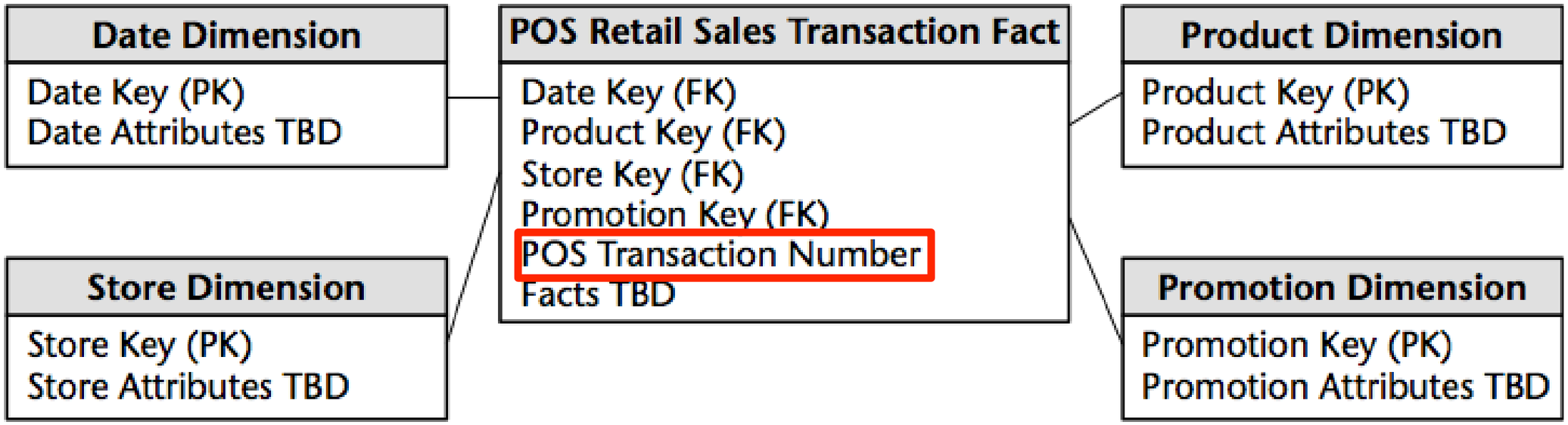


Figure 2.2 Preliminary retail sales schema.

"TBD" means "to be determined."

Step 4. Identify the Facts

- **The facts collected by the POS system include:**
  - **sales quantity (the number units)**
  - **per unit sales price**
  - **sales dollar amount.**
    - The sales dollar amount equals the sales quantity multiplied by the unit price.

- **Sophisticated POS systems also provide a standard dollar cost for the product as delivered to the store by the vendor.**
  - **standard dollar cost**
    - Assuming that this information is easily made available

Step 4. Identify the Facts

- **The facts:**
  - **sales quantity (the number units)**
  - **per unit sales price**
  - **sales dollar amount = sales quantity * per unit sales price**
  - **standard dollar cost (for all units sold on the line)**
  - **gross profit = sales dollar amount - standard dollar cost**
  - **gross margin = gross profit / sales dollar amount**
- **Percentages and ratios, such as gross margin, are nonadditive. The numerator and denominator should be stored in the fact table. The ratio can be calculated in a data access tool for any slice of the fact table by remembering to calculate the ratio of the sums, not the sum of the ratios.**

Step 4. Identify the Facts

- **Basic and additive facts:**
  - **sales quantity (the number units)**
  - **sales dollar amount (= sales quantity * per unit sales price)**
  - **standard dollar cost (for all units sold on the line)**
- **Calculated and additive**
  - **gross profit = sales dollar amount - standard dollar cost**
    - Store v.s. Calculate on fly (via a view)
- **Calculated and non-additive**
  - **gross margin = gross profit / sales dollar amount**
    - gross margin = SUM(gross profit) / SUM(sales dollar amount)
    - Average selling price = SUM(sales dollar amount) / SUM(sales quantity)

Step 4. Identify the Facts

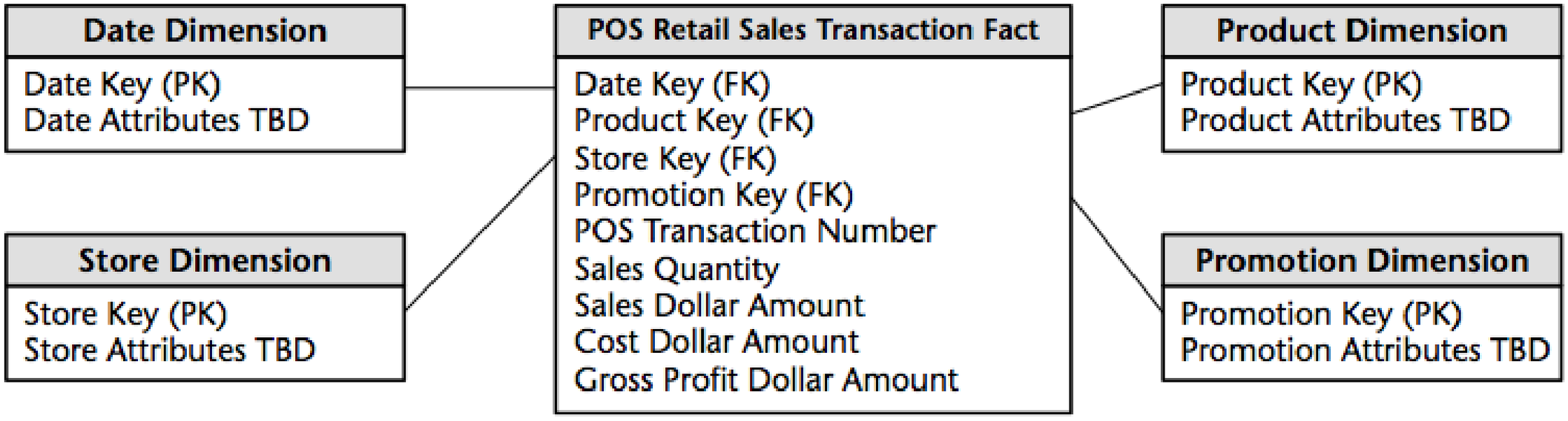


Figure 2.3 Measured facts in the retail sales schema.

Step 4. Facts - estimate the number of rows

- **How many POS transaction line items are generated on a periodic basis:**
  - **Retail traffic fluctuates significantly from day to day, so we'll want to understand the transaction activity over a reasonable period of time (annual).**
    - Ask to the operational manager
  - **Estimate the annual**
    - dividing chain's annual gross revenue by the average item selling price.

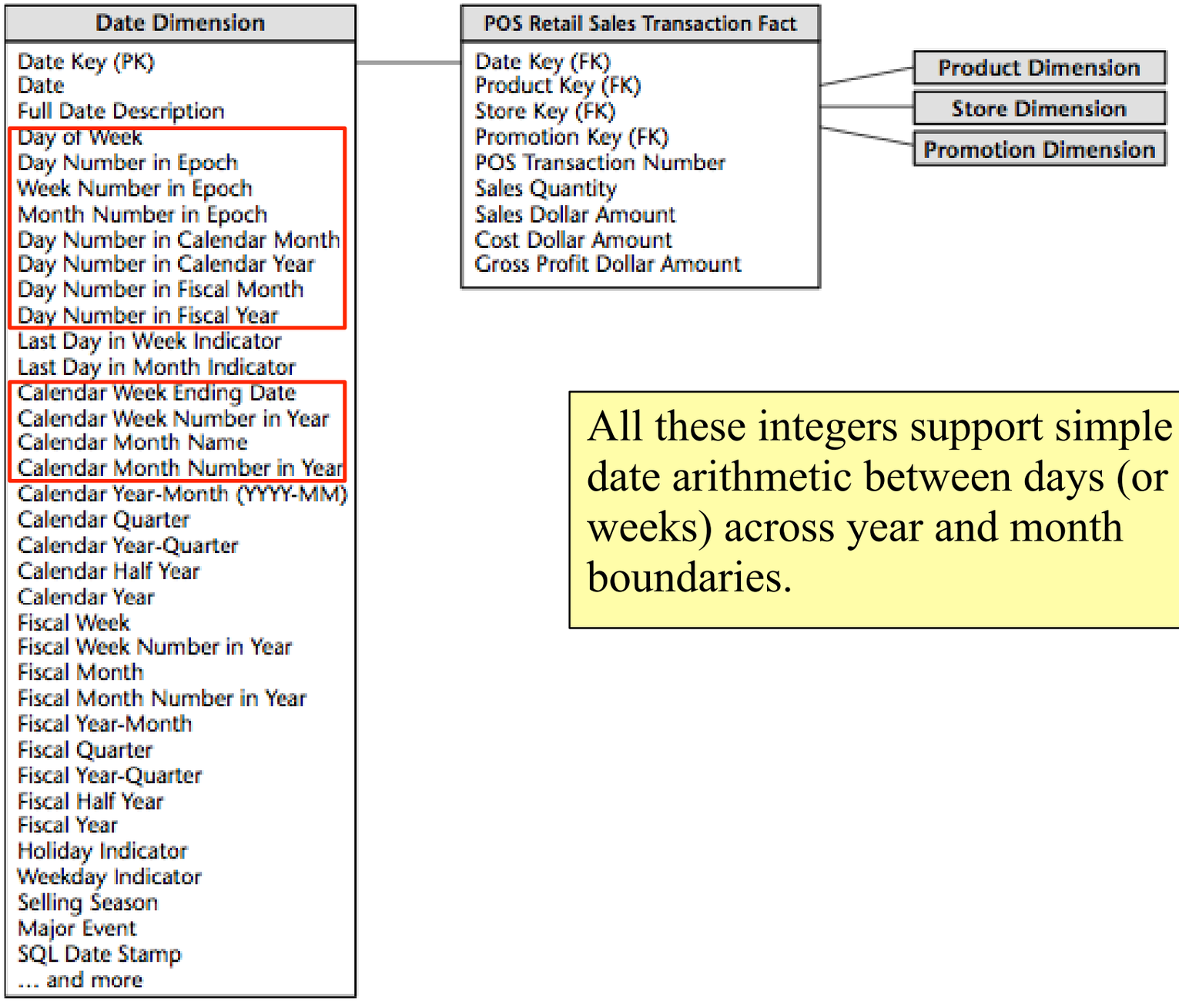
- **Approximately 2 billion transaction line items per year.**

Dimension Table Attributes

Date Dimension

- The date dimension is the one dimension nearly guaranteed to be in every data mart because virtually every data mart is a time series.
- Data warehouses always need an explicit date dimension table. There are many date attributes not supported by the SQL date function.
- Date dimension versus Time dimension
- Unlike most of our other dimensions, we can build the date dimension table in advance.
  - 5 or 10 years of rows representing days in the table so that we can cover the history we have stored, as well as several years in the future.
  - Even 10 years' worth of days is only about 3,650 rows, which is a relatively small dimension table

Date Dimension



All these integers support simple date arithmetic between days (or weeks) across year and month boundaries.

Figure 2.4 Date dimension in the retail sales schema.

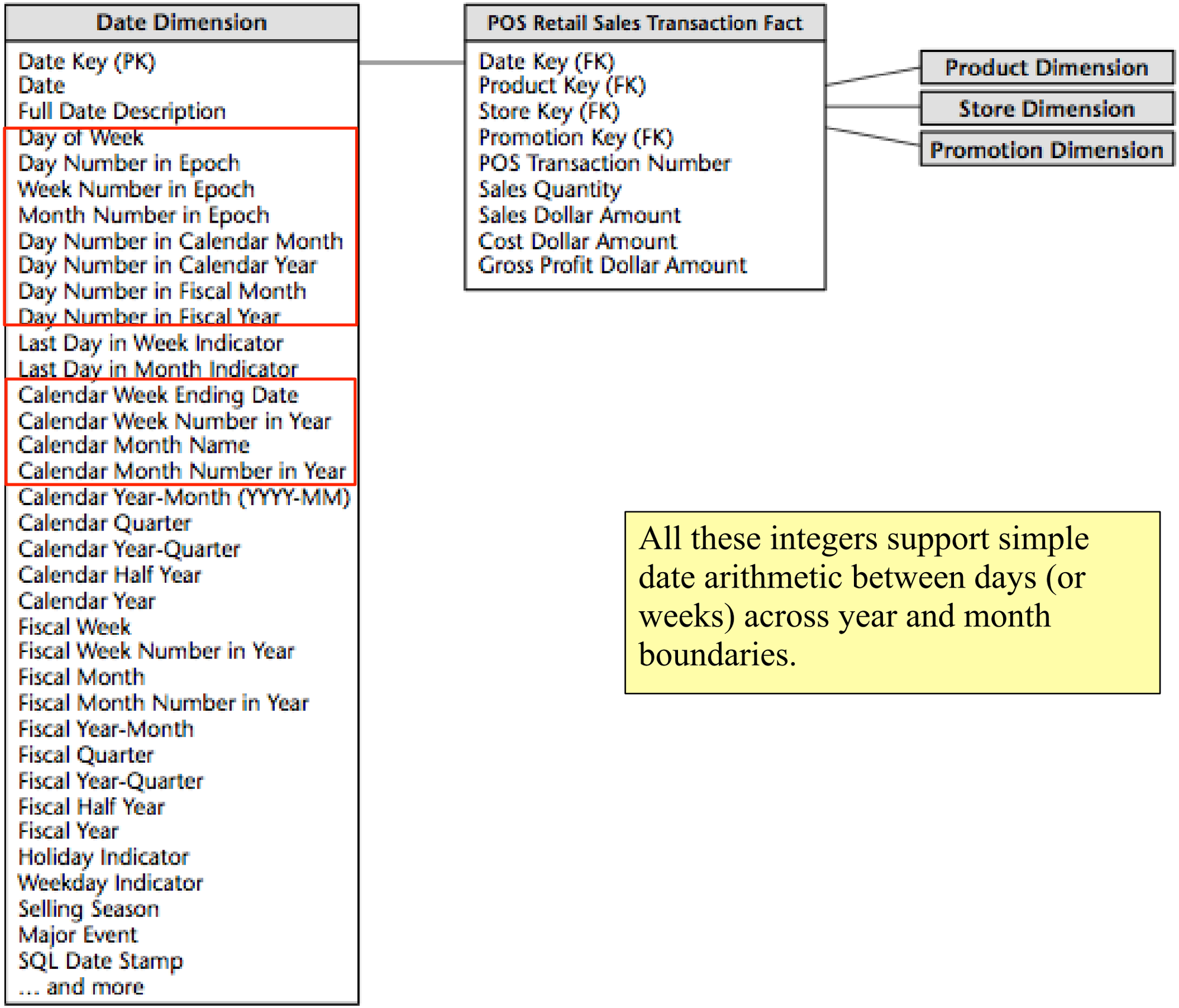


Figure 2.4 Date dimension in the retail sales schema.

Date Dimension

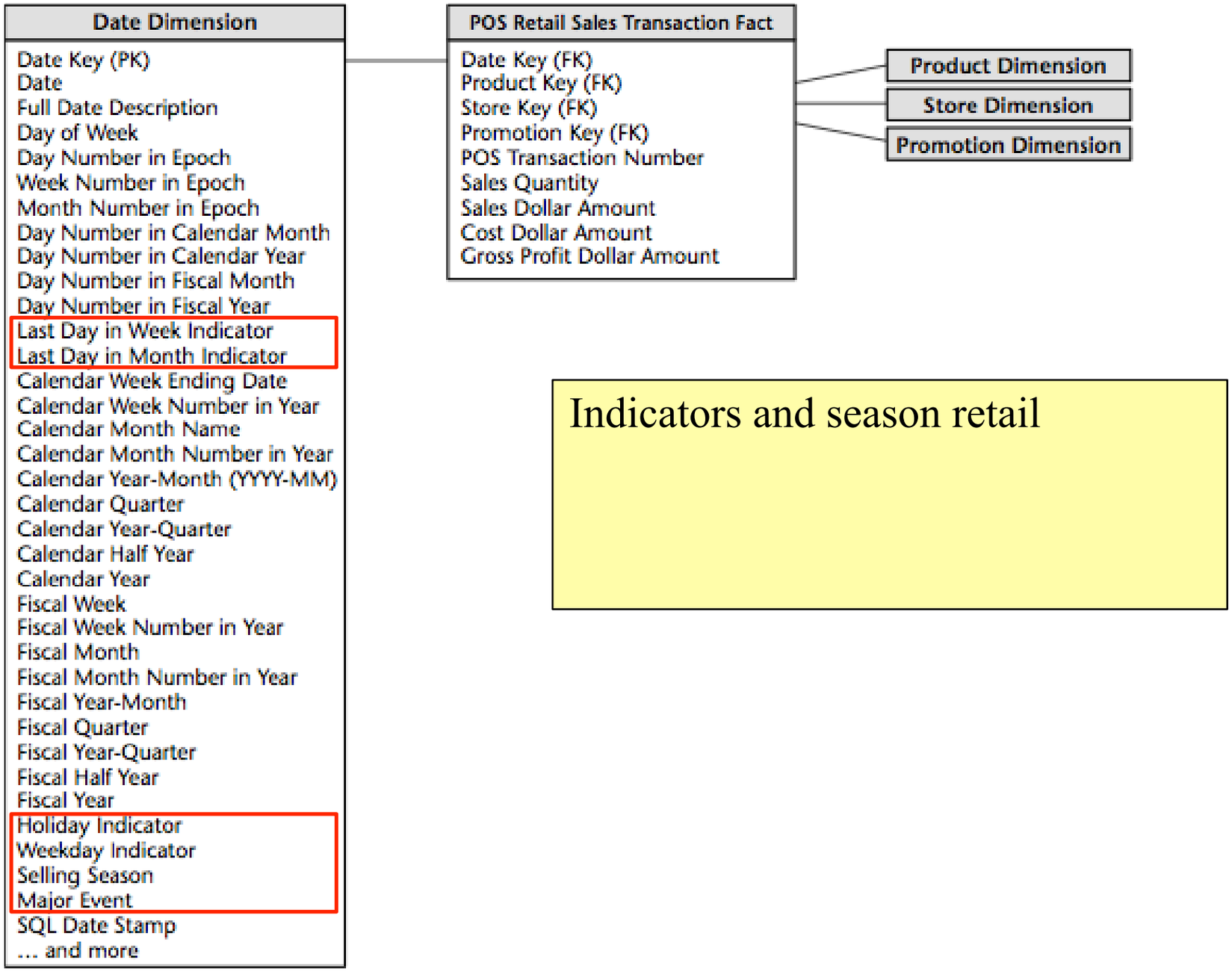


Figure 2.4 Date dimension in the retail sales schema.

Product Dimension

- The product dimension describes every SKU in the grocery store.

- While a typical store in our chain may stock 60,000 SKUs, when we account for different merchandising schemes across the chain and historical products that are no longer available, our product dimension would have at least 150,000 rows and perhaps as many as a million rows.

- The product dimension is almost always sourced from the operational product master file. An important function of the product master is to hold the many descriptive attributes of each SKU.

Product Dimension

- **The merchandise hierarchy is an important group of attributes. Typically, individual SKUs roll up to brands. Brands roll up to categories, and categories roll up to departments. Each of these is a many-to- one relationship.**

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

Figure 2.6 Product dimension table detail.

Product Dimension

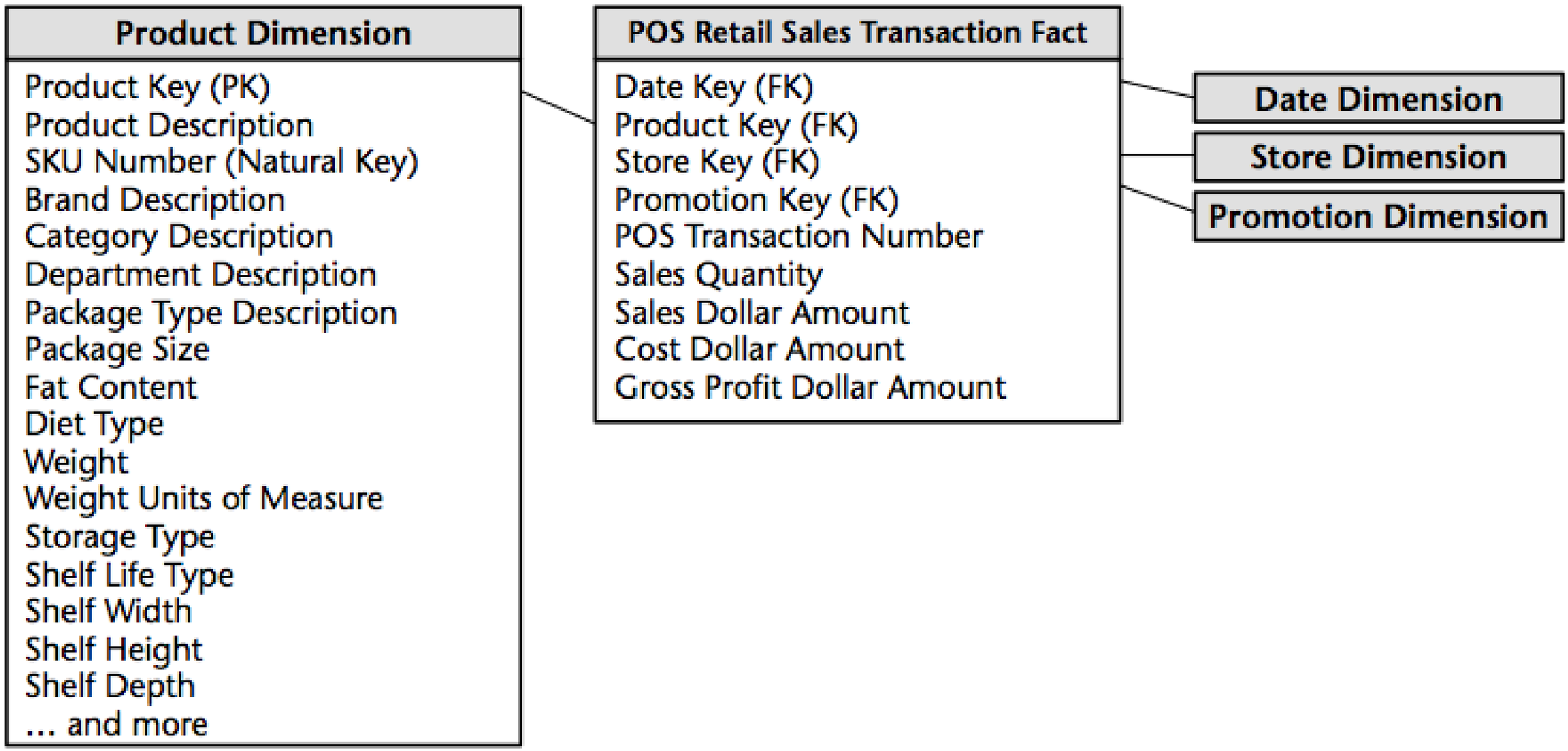
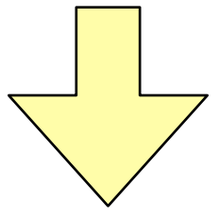


Figure 2.7 Product dimension in the retail sales schema.

Product Dimension

- **Drill down by Product dimension attributes**

Department Description	Sales Dollar Amount	Sales Quantity
Bakery	$12,331	5,088
Frozen Foods	$31,776	15,565



Department Description	Brand Description	Sales Dollar Amount	Sales Quantity
Bakery	Baked Well	$3,009	1,138
Bakery	Fluffy	$3,024	1,476
Bakery	Light	$6,298	2,474
Frozen Foods	Coldpack	$5,321	2,640
Frozen Foods	Freshlike	$10,476	5,234
Frozen Foods	Frigid	$7,328	3,092
Frozen Foods	Icy	$2,184	1,437
Frozen Foods	QuickFreeze	$6,467	3,162

Product Dimension

- **Many of the attributes in the product dimension table are not part of the merchandise hierarchy. The package-type attribute, for example, might have values such as Bottle, Bag, Box, or Other. The Fat content attribute may have values such as Non-Fat, Reduced Fat, Regular Fat, etc.**

Department Description	Fat Content	Sales Dollar Amount	Sales Quantity
Bakery	Non-Fat	$6,298	2,474
Bakery	Reduced Fat	$5,027	2,086
Bakery	Regular Fat	$1,006	528
Frozen Foods	Non-Fat	$5,321	2,640
Frozen Foods	Reduced Fat	$10,476	5,234
Frozen Foods	Regular Fat	$15,979	7,691

Store Dimension

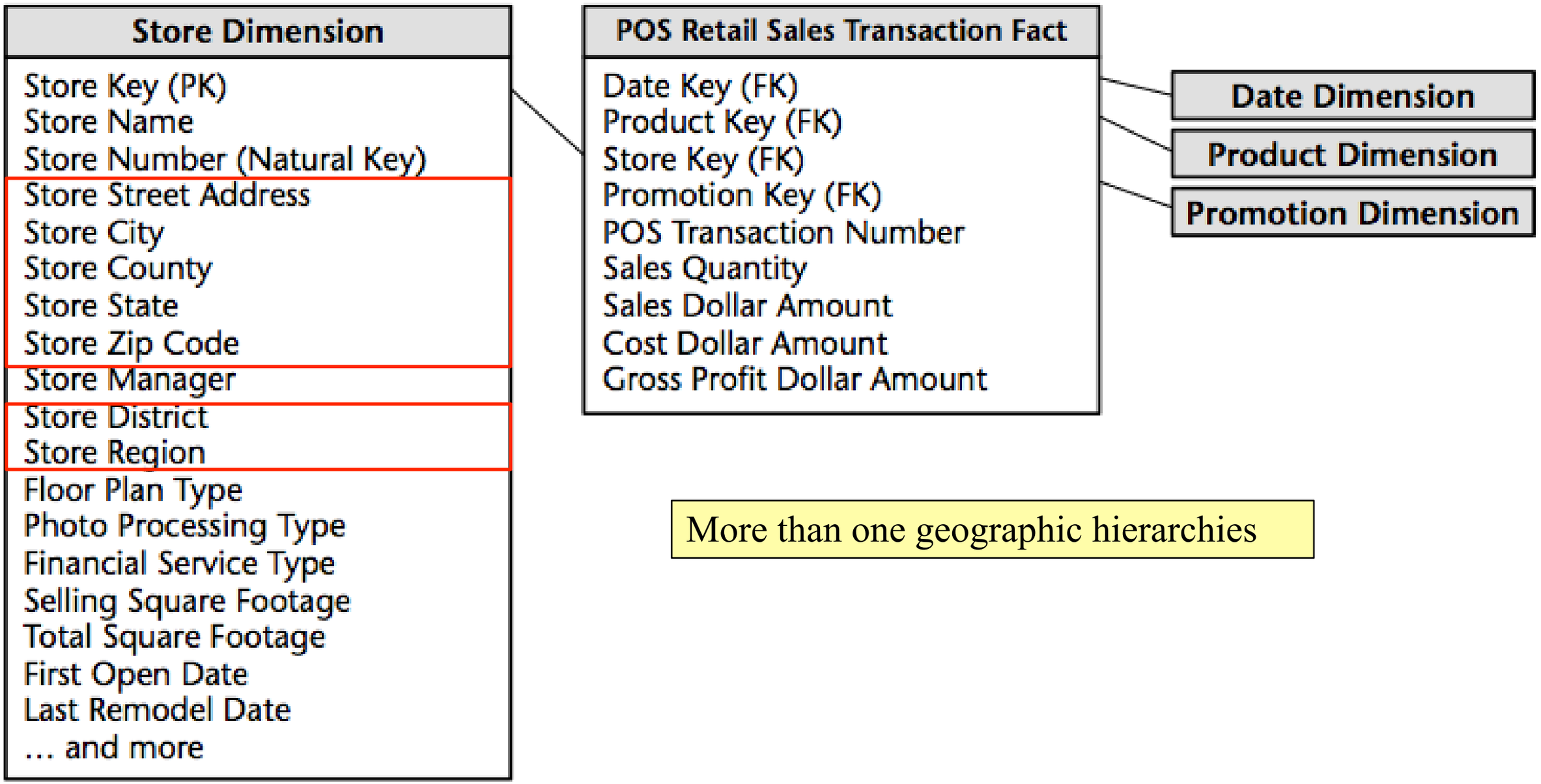


Figure 2.8 Store dimension in the retail sales schema.

Store Dimension

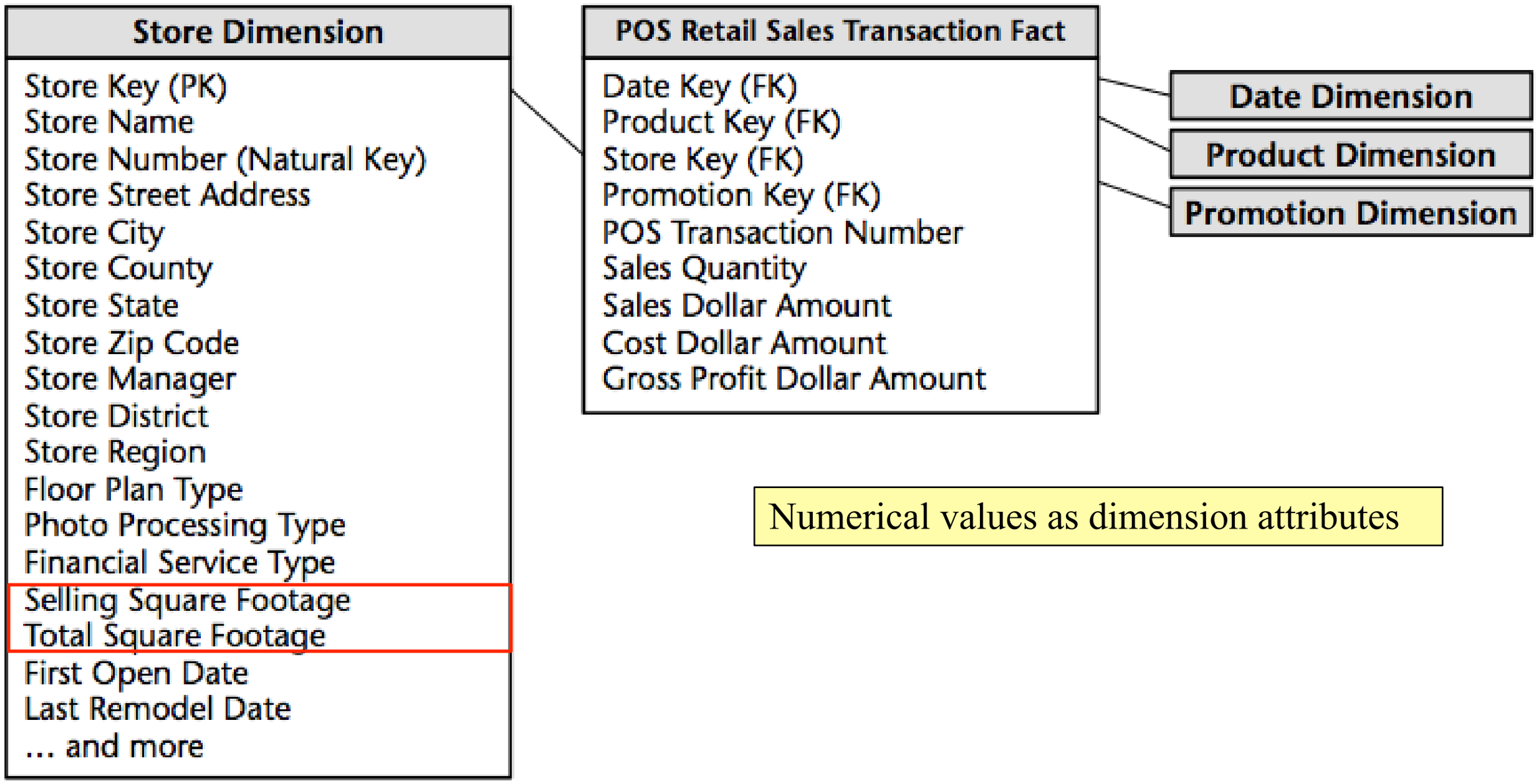
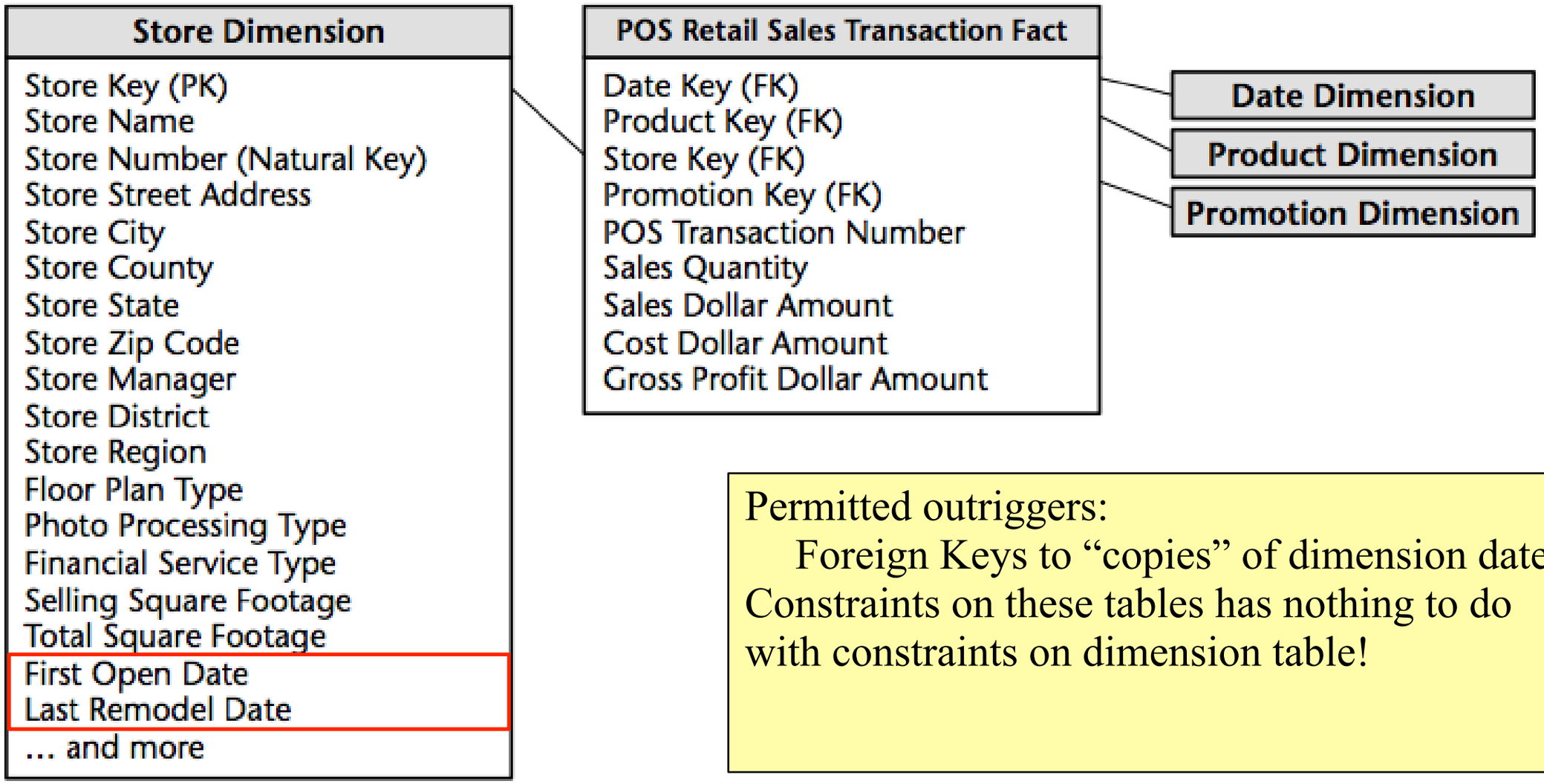


Figure 2.8 Store dimension in the retail sales schema.

Store Dimension



Permitted outriggers:

Foreign Keys to "copies" of dimension date

Constraints on these tables has nothing to do with constraints on dimension table!

Figure 2.8 Store dimension in the retail sales schema.

Promotion Dimension

- The promotion dimension describes the promotion conditions under which a product was sold: (i) temporary price reductions; (ii) end-aisle displays; (iii) newspaper ads; (iv) coupons.
- The various possible causal conditions are highly correlated.
  - A temporary price reduction usually is associated with an ad and perhaps an end-aisle display.
  - Coupons often are associated with ads.
- For this reason, it makes sense to create one row in the promotion dimension for each combination of promotion conditions that occurs. Over the course of a year, there may be 1,000 ads, 5,000 temporary price reductions, and 1,000 end-aisle displays, but there may only be 10,000 combinations of these three conditions affecting any particular product.

Promotion Dimension

- **Managers at both headquarters and the stores are interested in determining whether a promotion is effective or not.**
  - **Whether the products under promotion experienced a gain in sales during the promotional period. This is called the lift.**
  - **Whether the products under promotion showed a drop in sales just prior to or after the promotion, canceling the gain in sales during the promotion (time shifting).**
  - **Whether the products under promotion showed a gain in sales but other products nearby on the shelf showed a corresponding sales decrease (cannibalization).**
  - **Whether all the products in the promoted category of products experienced a net overall gain in sales (market growth).**
  - **Whether the promotion was profitable.**

Promotion Dimension

- **One dimension versus many distinct dimensions for each promotion type**

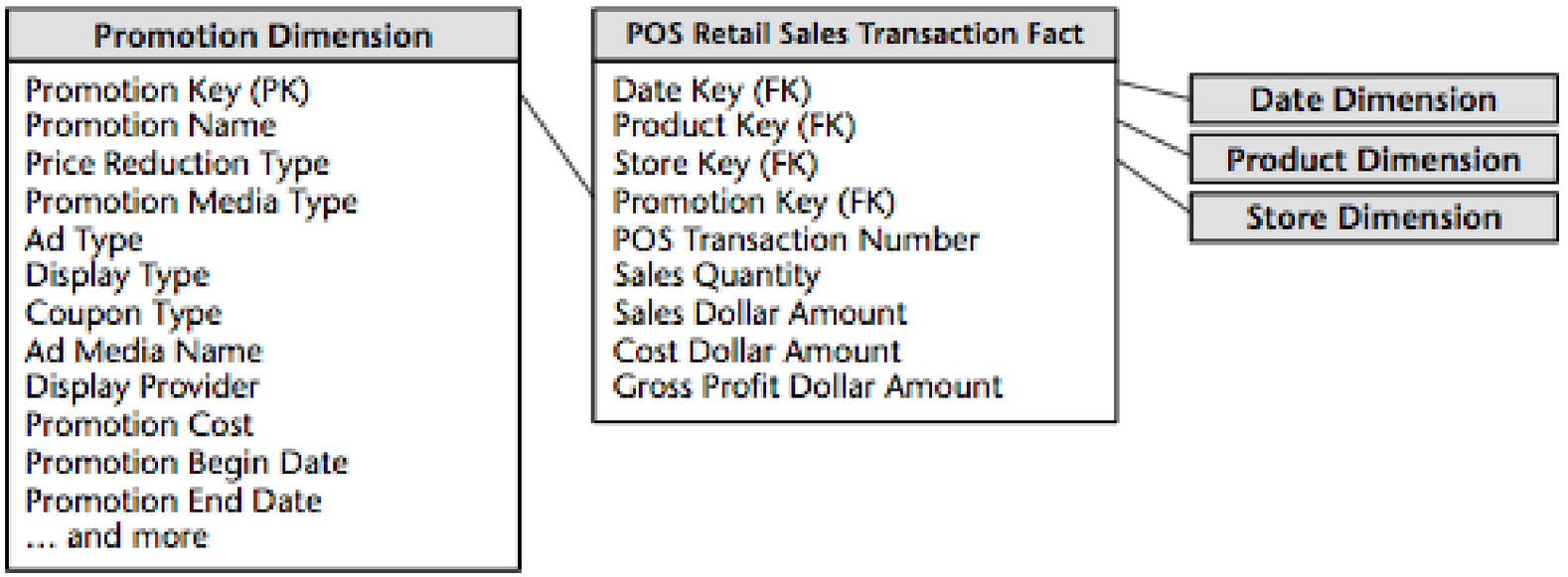
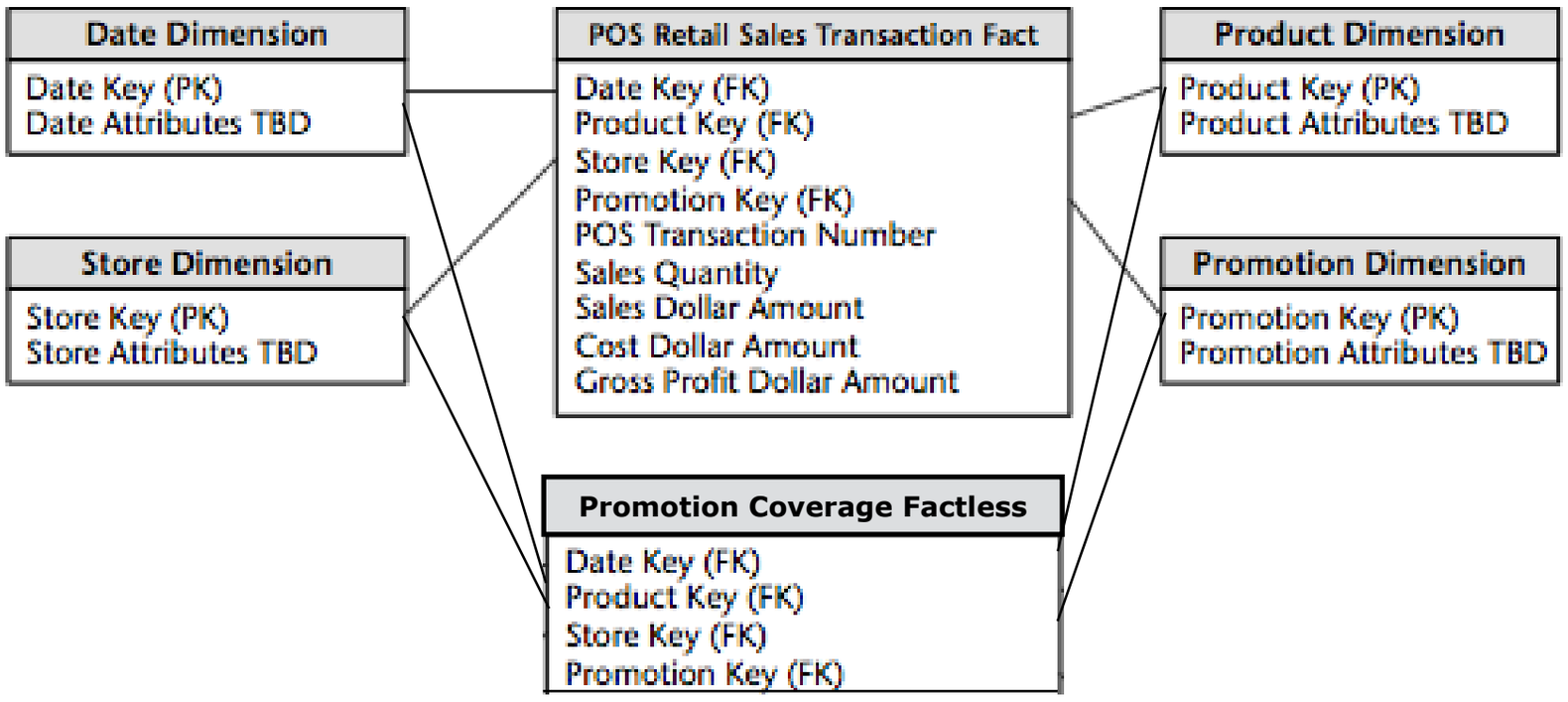


Figure 2.9 Promotion dimension in the retail sales schema.

- **Typically, many sales transaction line items involve products that are not being promoted.**
  - **To include a row in the promotion dimension, with its own unique key, to identify "No Promotion in Effect" and avoid a null promotion key in the fact table.**

Promotion Coverage Factless Fact Table

- Regardless of the handling of the promotion dimension, there is one important question that cannot be answered by our retail sales schema: What products were on promotion but did not sell?
  - The sales fact table only records the SKUs actually sold.
  - There are no fact table rows with zero facts for SKUs that didn't sell because doing so would enlarge the fact table enormously.



Promotion Coverage Factless Fact Table

- **A second promotion coverage or event fact table is needed to help answer the question concerning what didn't happen.**
- **The promotion coverage fact table keys would be date, product, store, and promotion.**
  - **It looks similar to the sales fact table we just designed;**
  - **however, the grain would be significantly different. In the case of the promotion coverage fact table, we'd load one row in the fact table for each product on promotion in a store each day (or week, since many retail promotions are a week in duration) regardless of whether the product sold or not.**
- **To determine what products where on promotion but didn't sell requires a two-step process: (i) which products are in promotion; (ii) what products sold; the answer is the difference between the two sets.**

Degenerate Transaction Number Dimension

- The POS transaction number is still useful because it serves as the grouping key for pulling together all the products purchased in a single transaction.

- The primary key of the retail sales fact table consists of the degenerate POS transaction number and product key (assuming that the POS system rolls up all sales for a given product within a POS shopping cart into a single line item).

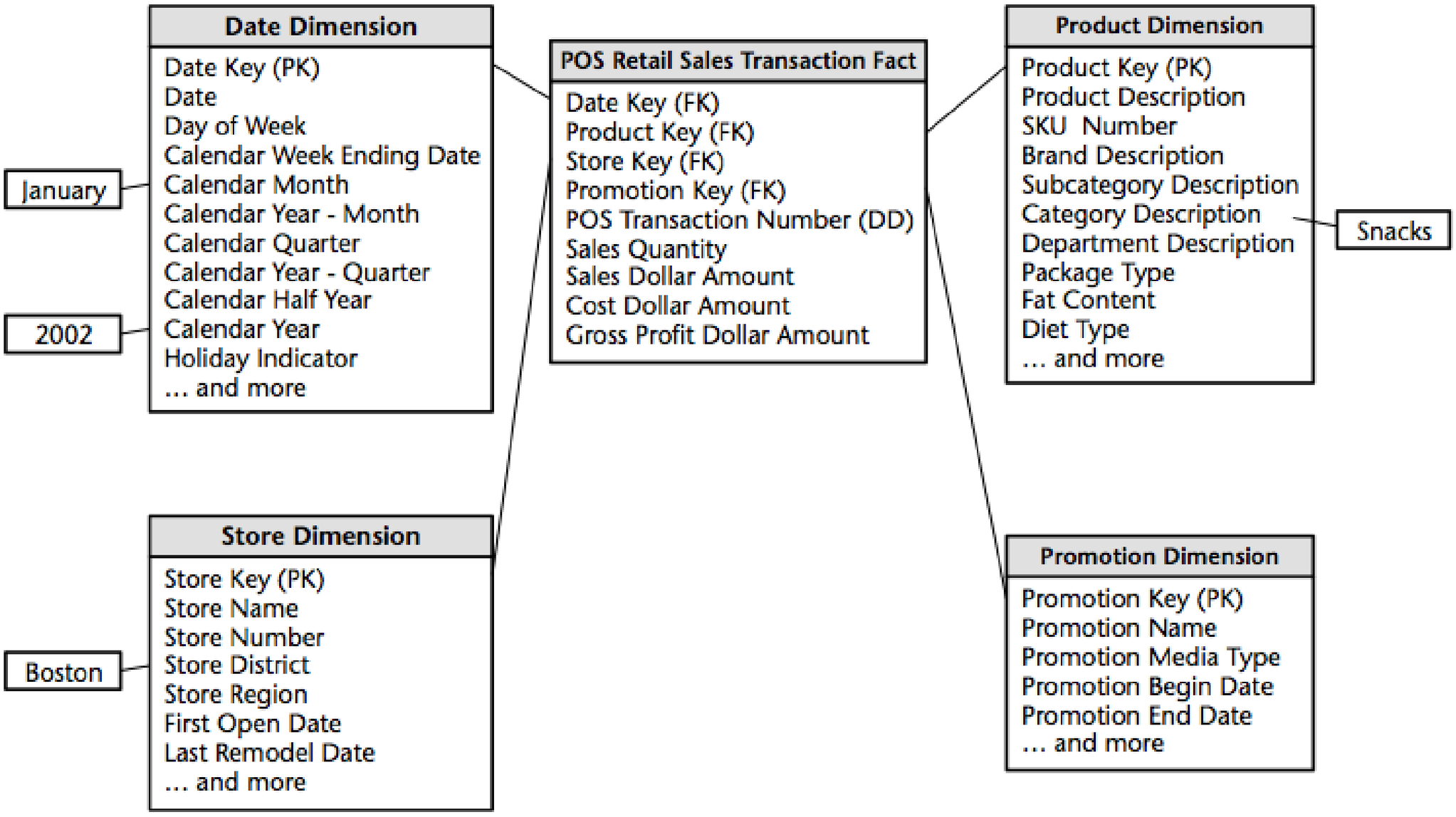
- If, for some reason, one or more attributes are legitimately left over after all the other dimensions have been created and seem to belong to this header entity, we would simply create a normal dimension record with a normal join. However, we would no longer have a degenerate dimension.

Degenerate Transaction Number Dimension

Operational control numbers such as order numbers, invoice numbers, and bill-of-lading numbers usually give rise to empty dimensions and are represented as **degenerate dimensions** (that is, dimension keys without corresponding dimension tables) in fact tables where the grain of the table is the document itself or a line item in the document.

Often, the primary key of a fact table is a **subset** of the table's foreign keys. We typically do not need every foreign key in the fact table to guarantee the uniqueness of a fact table row.

Retail Schema

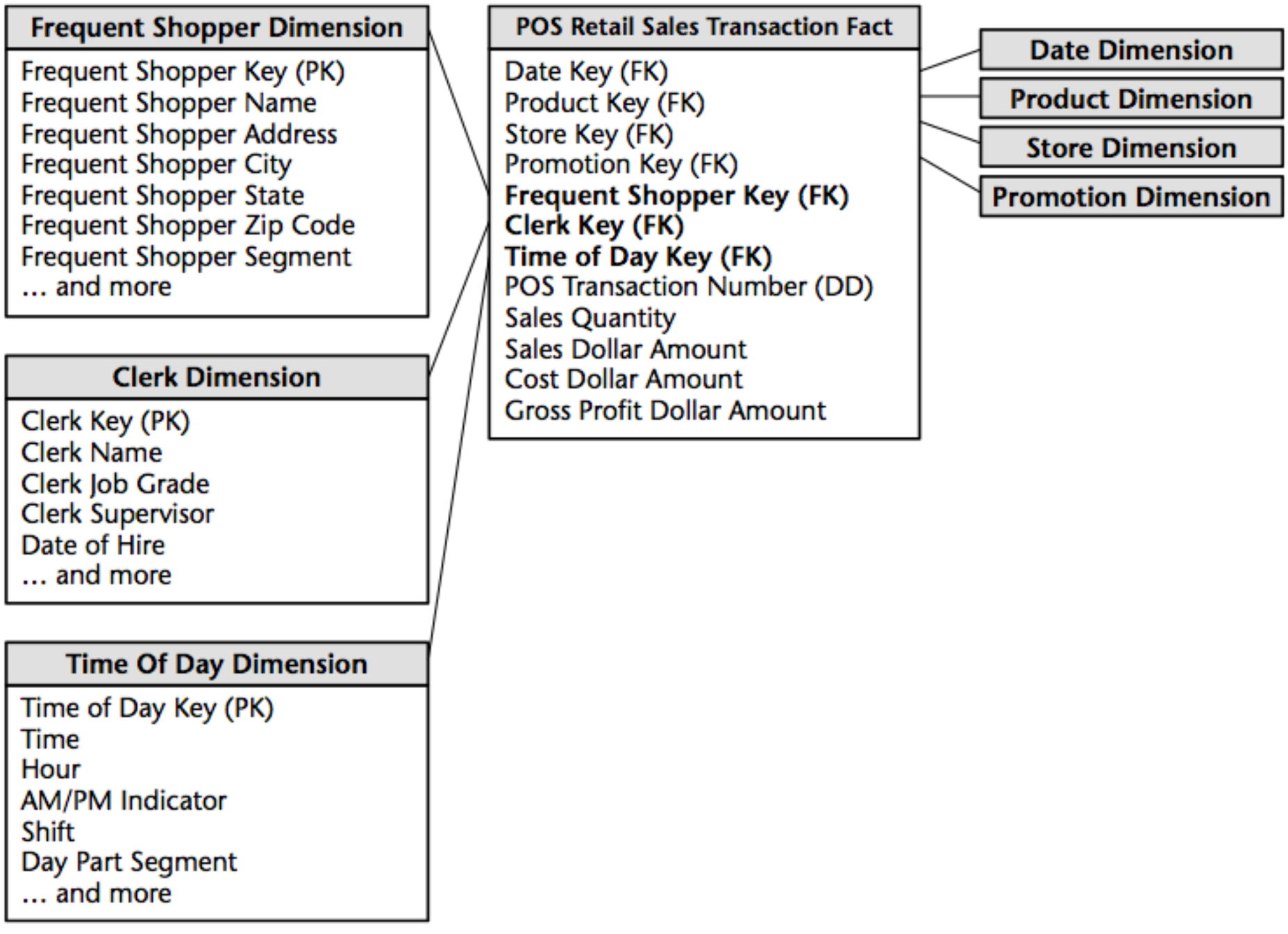


Extending the Multidimensional Model

More dimensions

- **Frequent shopper program.**
  - **We'd create a frequent shopper dimension table and add another foreign key in the fact table.**
  - **A shopper key corresponding to a "Prior to Frequent Shopper Program"**
  - **A shopper key corresponding to a "Frequent Shopper Not Identified"**
- **To control the clerk and sales during the day**
  - **Add dimensions for the time of day and clerk associated with the transaction**
- **The addition of dimensions that apply at that granularity did not alter the existing dimension keys or facts; all preexisting applications continue to run without unraveling or changing.**

More Dimensions



New dimension attributes

- **If we discover new textual descriptors of a product, for example, we add these attributes to the dimension as new columns.**

- **All existing applications will be oblivious to the new attributes and continue to function.**

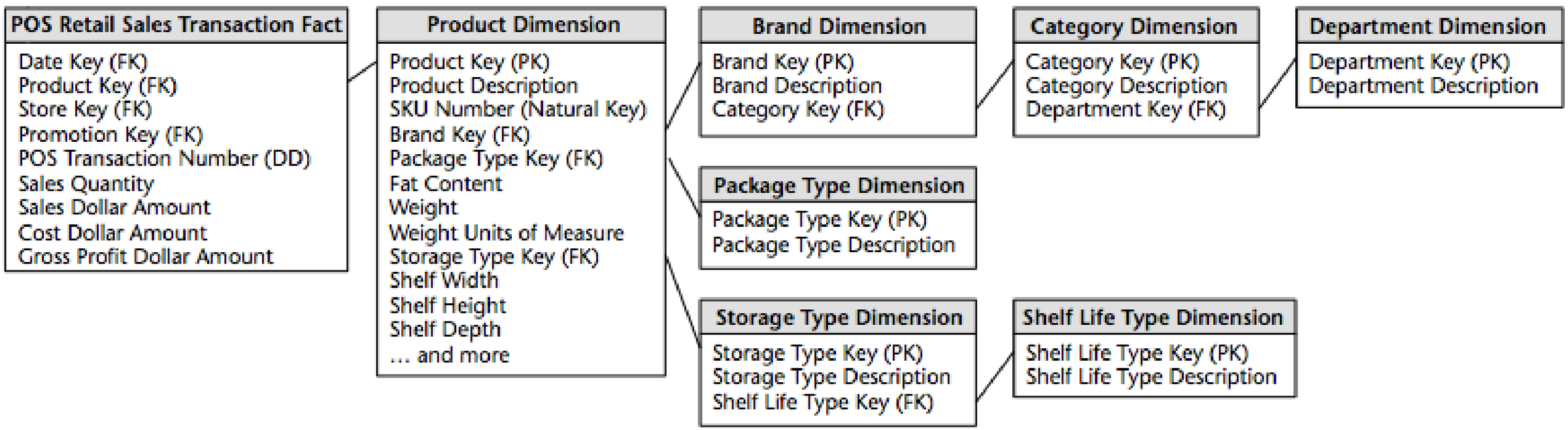
- **If the new attributes are available only after a specific point in time, then “Not Available” or its equivalent should be populated in the old dimension records.**

New measured facts

- **If new measured facts become available, we can add them gracefully to the fact table.**
  - **When the new facts are available in the same measurement event and at the same grain as the existing facts:**
    - add the new columns.
  - **When new measured facts occur naturally at a different grain**
    - create a new fact table

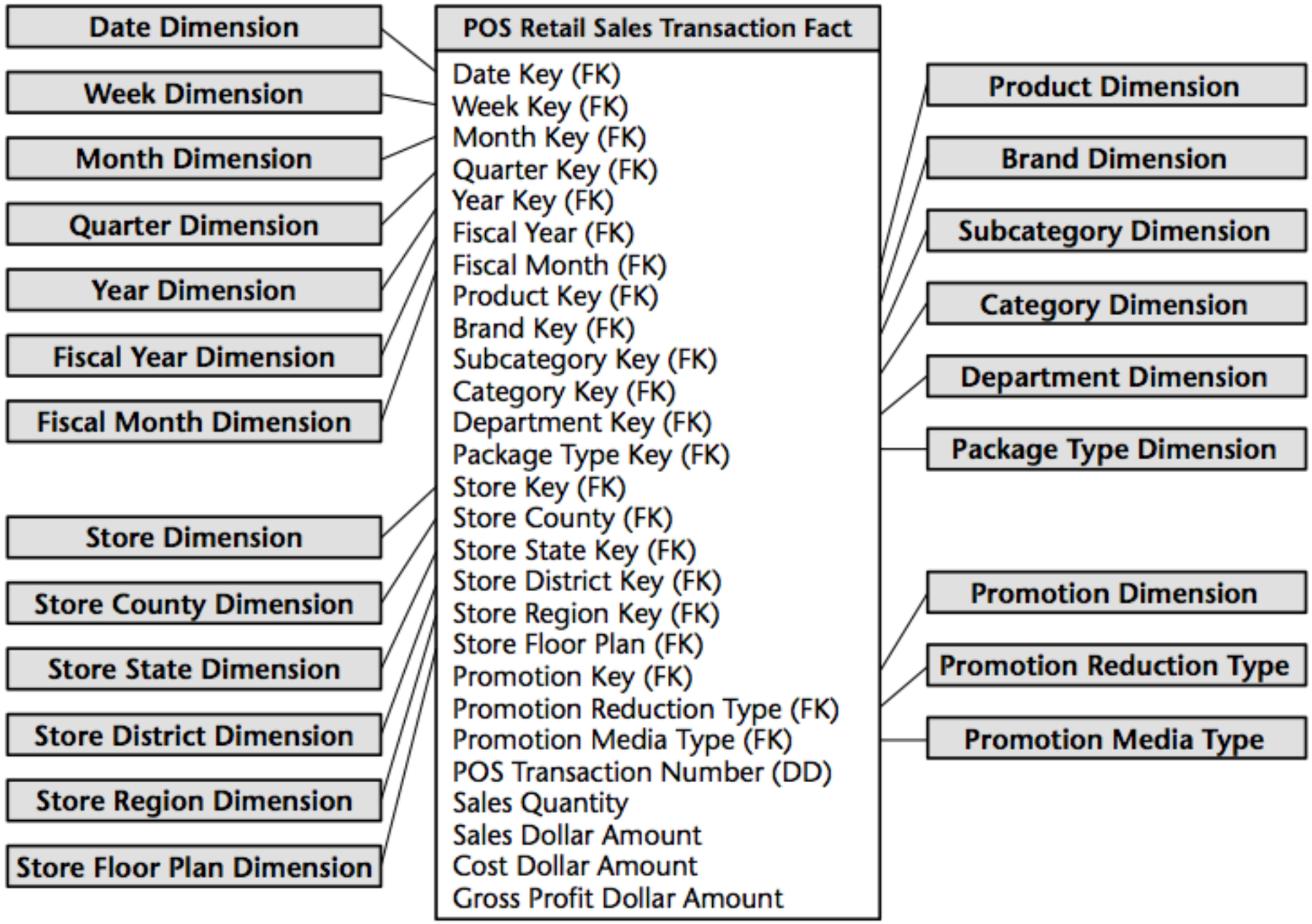
Notes on dimensions

Dimension Normalization (Snowflaking)

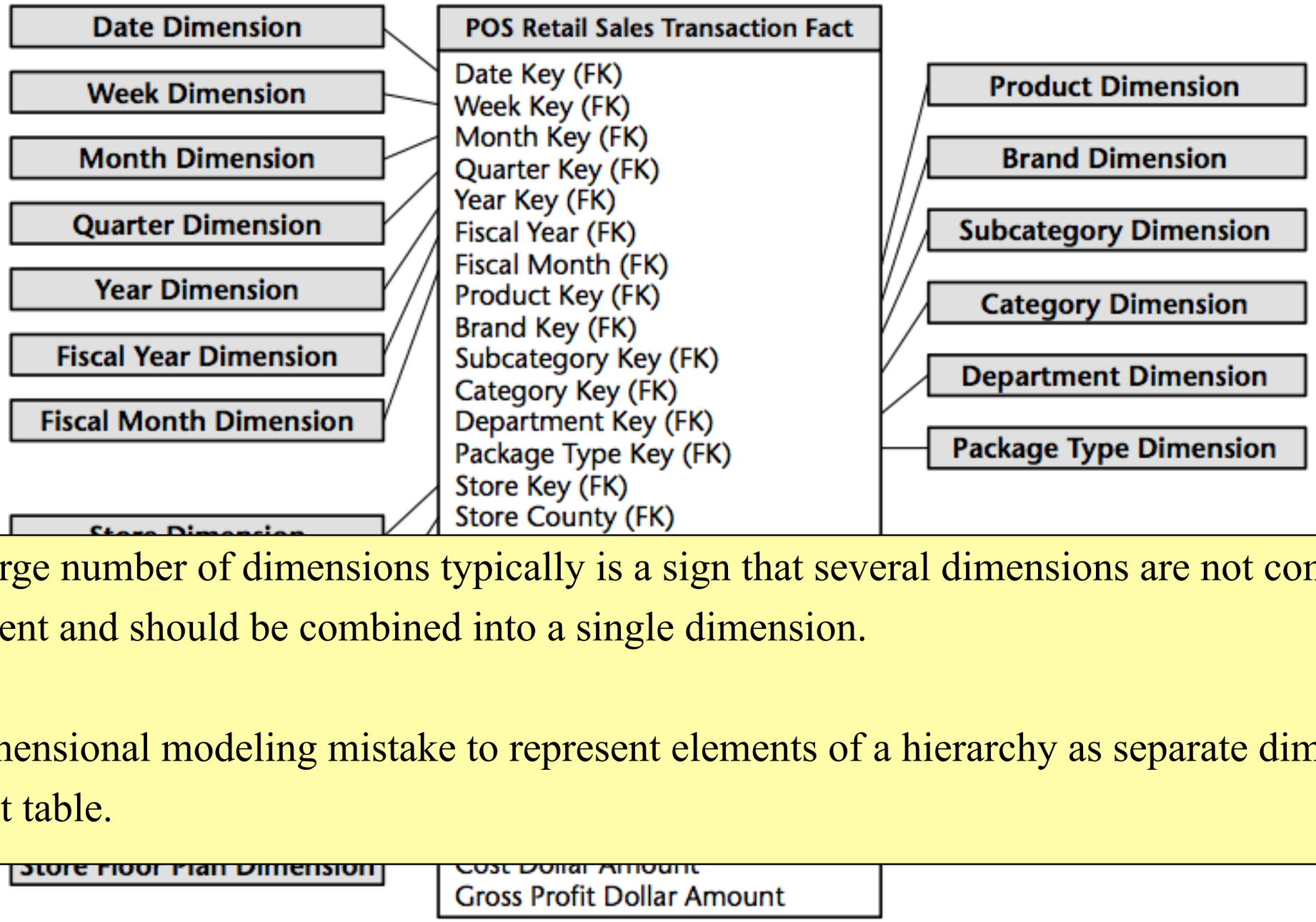


The dimension tables should remain as flat tables physically. Normalized, **snowflaked** dimension tables penalize cross-attribute browsing and prohibit the use of bit-mapped indexes. Disk space savings gained by normalizing the dimension tables typically are less than 1 percent of the total disk space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

Too Many Dimensions



Too Many Dimensions



A very large number of dimensions typically is a sign that several dimensions are not completely independent and should be combined into a single dimension.

It is a dimensional modeling mistake to represent elements of a hierarchy as separate dimensions in the fact table.

Surrogate Keys

- **Use of surrogate keys in dimensional models rather than relying on operational production codes.**
- **Surrogate keys are integers that are assigned sequentially as needed to populate a dimension.**
- **Surrogate keys advantages:**
  - **They buffer the data warehouse environment from operational changes.**
  - **Performance advantages: The surrogate key is as small an integer as possible while ensuring that it will accommodate the future cardinality.**
  - **Used to record dimension conditions that may not have an operational code, such as the "No Promotion in Effect" condition.**
  - **Are needed to support one of the primary techniques for handling changes to dimension table attributes (we will discuss later).**

Dimensional modeling

- **In practice, dimensional modeling is an iterative process. These procedures are useful for producing a first cut design, but this will need to be refined to produce the final data mart design.**
- **Most of these modifications have to do with further simplifying the model and dealing with non hierarchical patterns in the data.**
  - **Combining Fact Tables**
  - **Combining Dimension Tables**
  - **Produce pre-aggregated stars**