06

Multidimensional Modeling - Procurement

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

Introduction

Procurement

- For many companies, procurement is a critical business activity.
  - **Grocery chain**:
    - Effective procurement of products at the right price for **resale** is obviously important to retailers such as our grocery chain.
  - **Production companies**
    - Procurement also has strong bottom-line implications for any large organization that **buys products as raw materials** for manufacturing. Significant cost-savings opportunities are associated with reducing the number of suppliers and negotiating agreements with preferred suppliers.

[Kimball, 2002]

What involves Procurement

- Demand planning drives efficient materials management.
  - Once demand is forecasted, procurement's goal is **to source the appropriate materials/products** in the **most economical manner**.
  - Procurement involves a wide range of **activities** from **negotiating contracts** to **issuing purchase requisitions** and **purchase orders** (POs) to **tracking receipts** and **authorizing payments**.

[Kimball, 2002]

Common Analytic Requirements

- **Which** materials or products are purchased most frequently? **How many vendors** supply these products? At what prices? In what units of measure?
- **Looking at demand across the enterprise** (rather than at a single physical location), are there opportunities to negotiate favorable pricing by consolidating suppliers, single sourcing, or making guaranteed buys?
- Are our employees **purchasing from the preferred vendors** or skirting the negotiated vendor agreements (maverick spending)?
- Are we receiving the negotiated pricing from our vendors (vendor contract purchase price variance)?
- How are our **vendors performing**? What is the **vendor's fill rate**? On-time **delivery performance**? **Late deliveries** outstanding? **Percent of orders backordered**?

[Kimball, 2002]

Procurement Transactions

Transactions

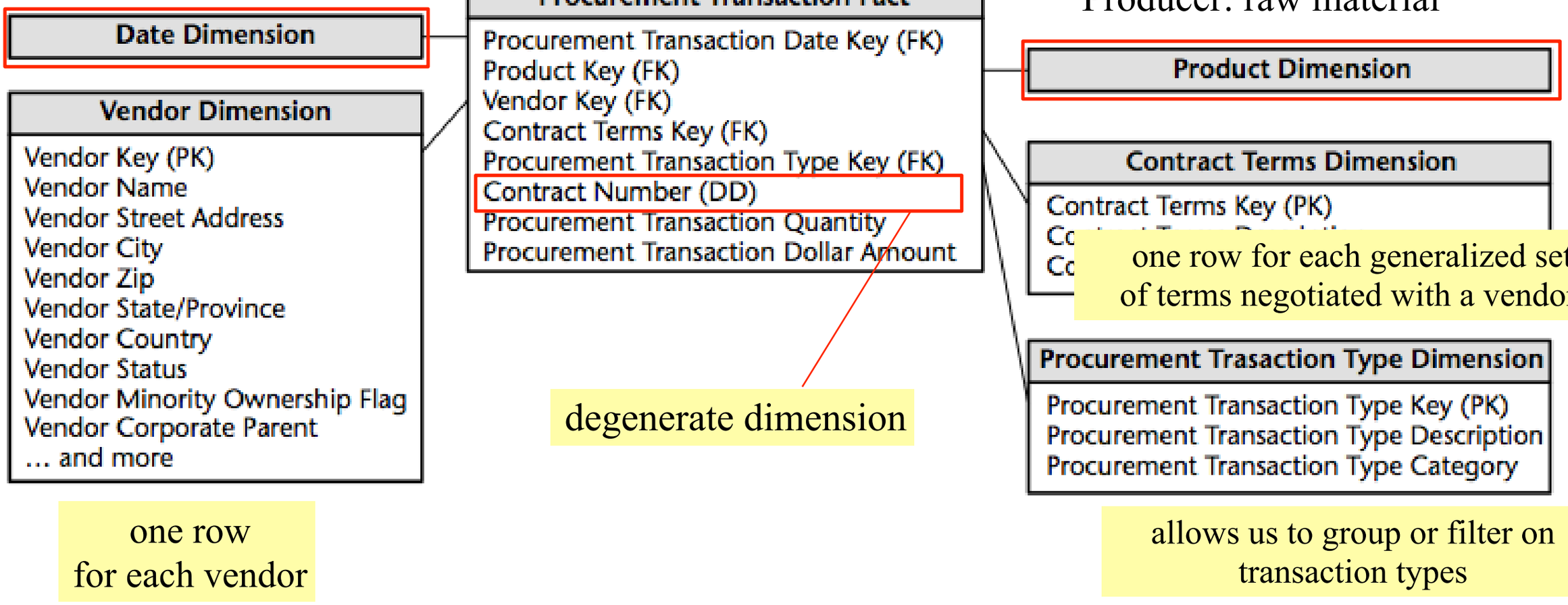
- **Possible Transactions in procurement**
  - Purchase requisitions.
  - Purchase orders.
  - Shipping notifications.
  - Receipts.
  - Payments.
- **Dimensionality**
  - fact table with the grain of **one row per procurement transaction**.
  - We identify **transaction date**, **product**, **vendor**, **contract terms**, and **procurement transaction type** as our key dimensions.

[Kimball, 2002]

Model

The same dimension

Retailer: products to sell
Producer: raw material



one row for each generalized set of terms negotiated with a vendor,

degenerate dimension

one row for each vendor

allows us to group or filter on transaction types

[Kimball, 2002]

Multiple-versus Single-Transaction Fact Tables

- First of all, we learn that the business users **describe the various procurement transactions differently**.
  - To the business, purchase orders, shipping notices, warehouse receipts, and vendor payments are all **viewed as separate and unique processes**.
- Several of the procurement transactions actually come from different source systems:
  - Purchasing system that provides **purchase requisitions** and **purchase orders.**
  - Warehousing system that provides **shipping notices** and **warehouse receipts**.
  - Accounts payable system that deals with **vendor payments**.

[Kimball, 2002]

Multiple-versus Single-Transaction Fact Tables

- Several transaction types have **different dimensionality**:
  - **Discounts** taken are applicable to **vendor payments** but not to the other transaction types.
  - Similarly, the name of the warehouse **clerk** who received the goods at the warehouse applies to **receipts** but doesn't make sense elsewhere.
- A variety of interesting control numbers, such as purchase order and payment check numbers, that are created at various steps in the procurement process. These control numbers are perfect candidates for degenerate dimensions. For **certain transaction types, more than one control number may apply**.

[Kimball, 2002]

Multiple-versus Single-Transaction Fact Tables

- Design decision:
  - **What are the users' analytic requirements**?
    - One goal is to reduce complexity in order to present the data in the most effective form for the business users.
    - How will the business users most commonly analyze this data?
    - Do the required analyses often require multiple transaction types together, leading us to consider a single blended fact table?
    - Or do they more frequently look solely at a single transaction type in an analysis, causing us to favor separate fact tables for each type of transaction?

[Kimball, 2002]

Multiple-versus Single-Transaction Fact Tables

- Design decision:
  - **Are there really multiple unique business processes**?
    - In this procurement example, it seems that buying products (purchase orders) is distinctly different from receiving products (receipts).
    - The existence of separate control numbers for each step in the process is a clue that we are dealing with separate processes. Given this situation, we would lean toward separate fact tables.

[Kimball, 2002]

Multiple-versus Single-Transaction Fact Tables

- Design decision:
  - **Are multiple source systems involved**?
    - In our example, we're dealing with three separate source systems: purchasing, warehousing, and accounts payable. Again, this would suggest separate fact tables.
  - **What is the dimensionality of the facts**?
    - In our procurement example we discovered several dimensions that applied to some transaction types but not to others. This would again lead us to separate fact tables.

Multiple fact tables allow us to provide richer, more descriptive dimensions and attributes.
As we progress **from purchase requisitions all the way to vendor payments**, we **inherit date dimensions and degenerate dimensions** from the previous steps.

[Kimball, 2002]

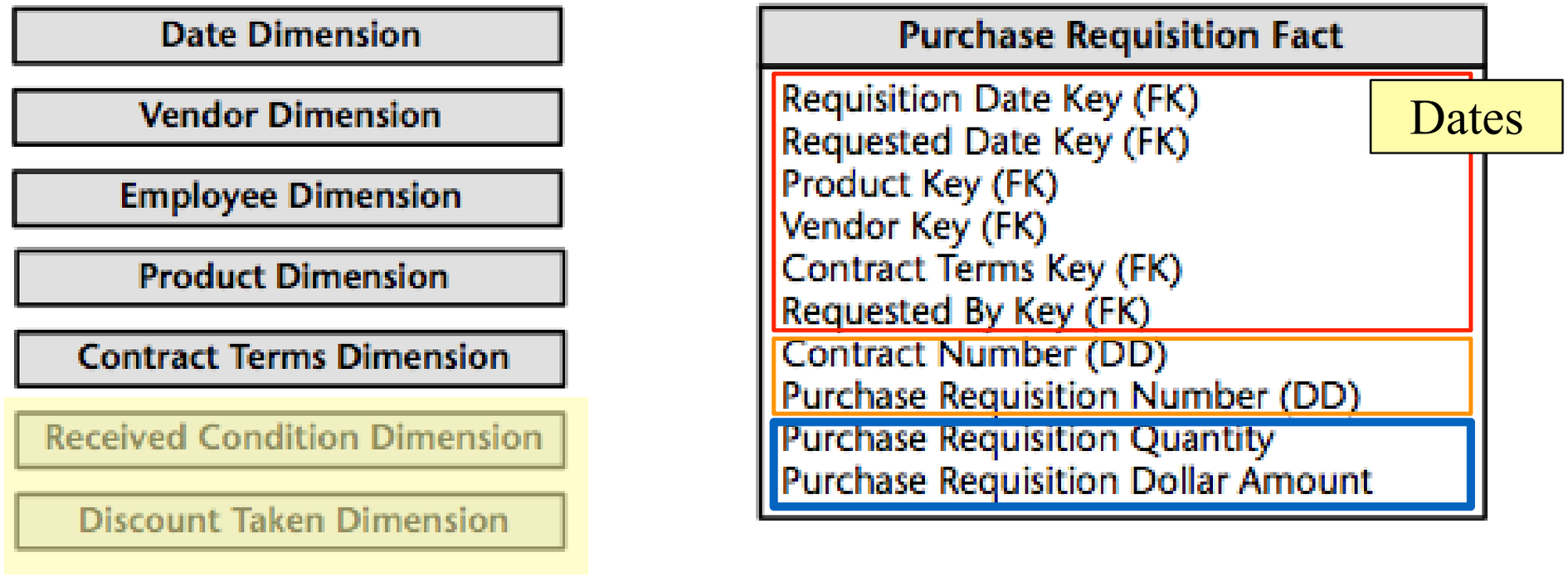
Transactions

- **Possible Transactions in procurement**
  - Purchase requisitions.
  - Purchase orders.
  - Shipping notifications.
  - Receipts.
  - Payments.

[Kimball, 2002]

Multiple-versus Single-Transaction Fact Tables

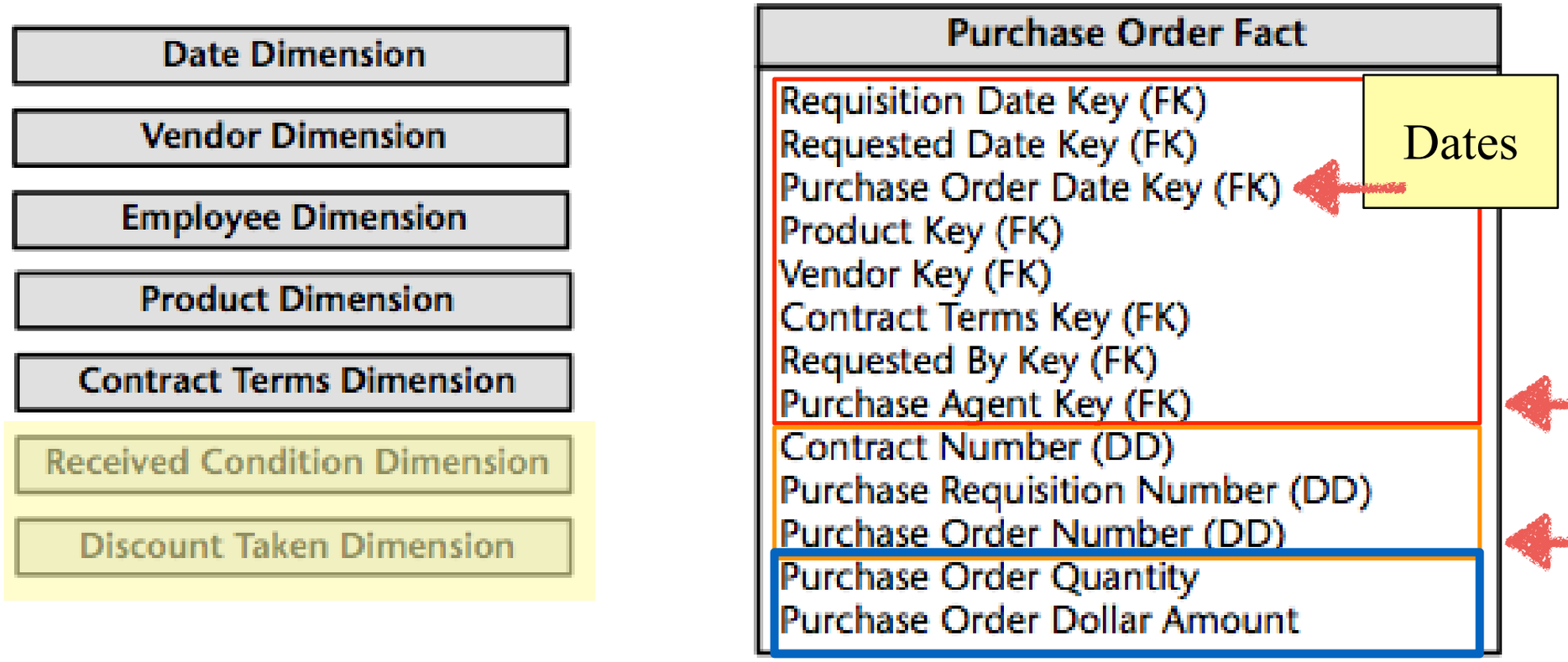
- **Purchase requisitions**



[Kimball, 2002]

Multiple- versus Single-Transaction Fact Tables

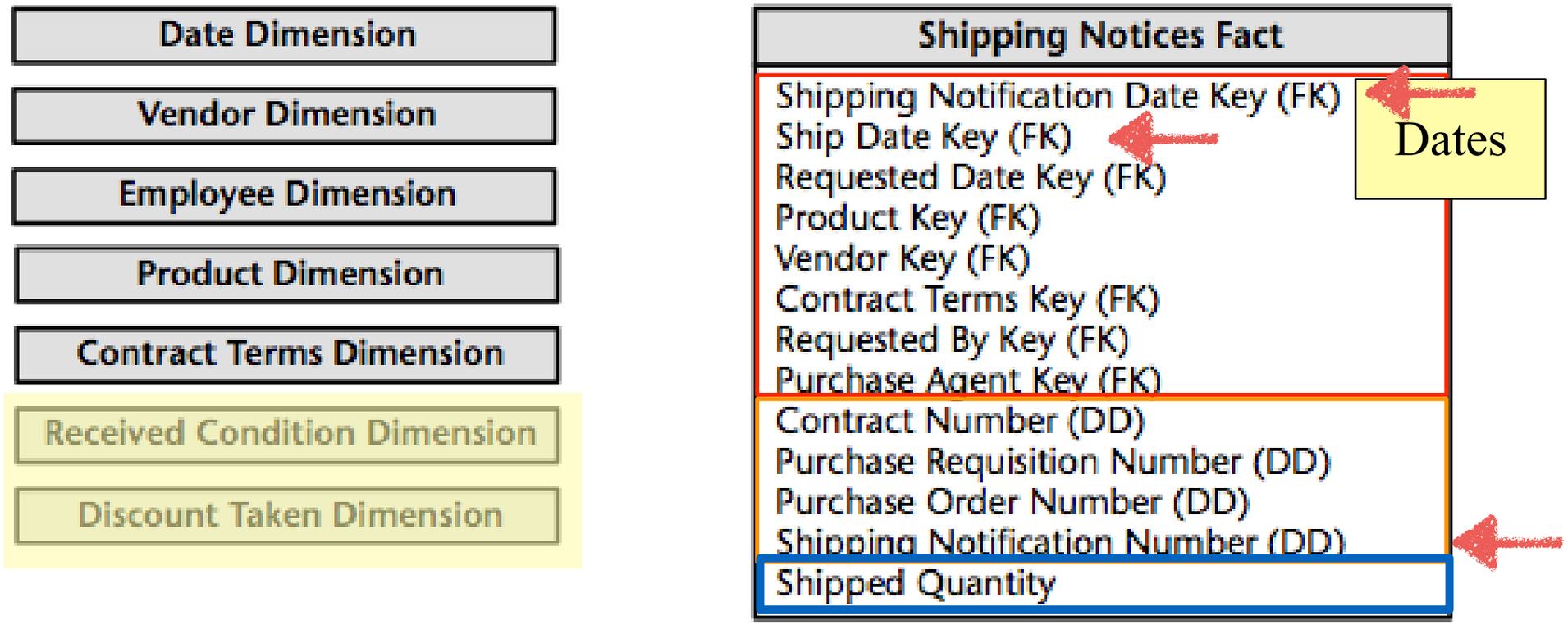
- **Purchase orders**



[Kimball, 2002]

Multiple- versus Single-Transaction Fact Tables

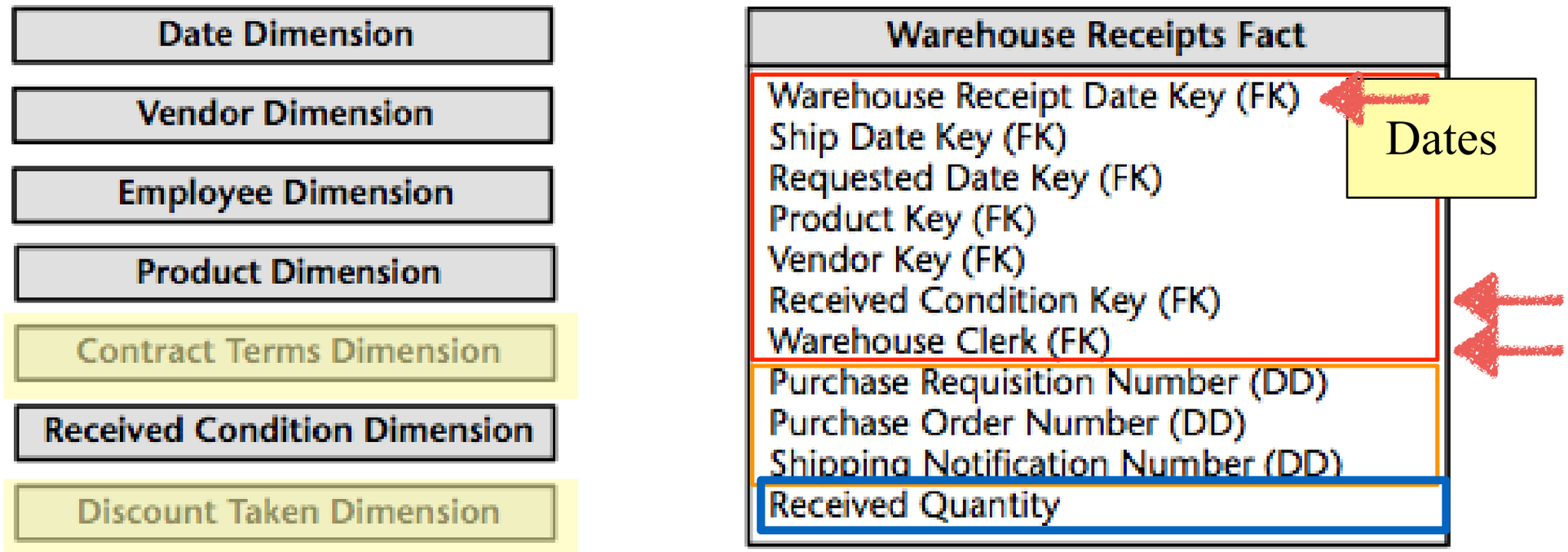
- **Shipping notifications**



[Kimball, 2002]

Multiple- versus Single-Transaction Fact Tables

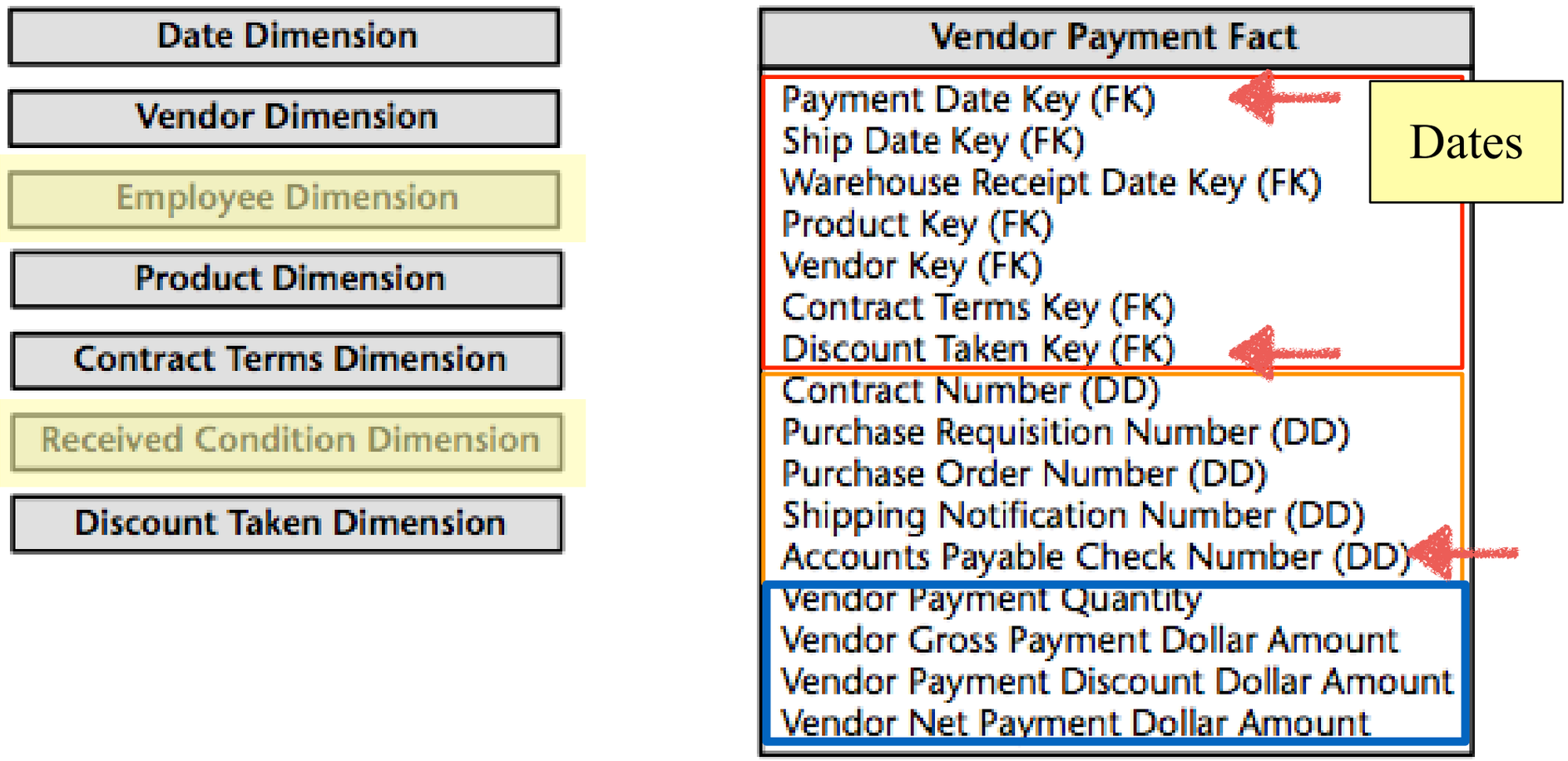
- **Warehouse Receipts**



[Kimball, 2002]

Multiple- versus Single-Transaction Fact Tables

- **Vendor's Payments**



[Kimball, 2002]

Multiple- versus Single-Transaction Fact Tables

- The single fact table approach would have required generalization of the labeling for some dimensions.
  - For example, purchase **order date** and **receipt date** likely would have been generalized to **transaction date**.
  - Likewise, purchasing agent and receiving clerk would become employee.

- *In another organization with different business requirements, source systems, and data dimensionality, the single blended fact table may be more appropriate.*

[Kimball, 2002]

Slowly Changing Dimensions

Slowly Changing Dimensions - What !?

- Dimensions have been assumed to be independent of time. Unfortunately, this is not the case in the real world. While **dimension table attributes are relatively static, they are not fixed forever**. **Dimension attributes change, albeit rather slowly, over time**:
  - The civil state of client may change from single to married or from married to divorced.
  - The sales regions may vary one on two or three years
  - The commercial classification of a product may vary one or two during its life
  - The frozen and refrigerated storage square footages may change on a store or an Warehouse
  - The product line is restructured, making changes in the the hierarchies products.

[Kimball, 2002]

Slowly Changing Dimensions - Strategies

- In 1994 Ralph Kimball first introduced the notion of ***Slowly Changing Dimensions***
  - Nearly constant dimensions with some attributes varying slowly. We are assuming that the attribute values are in general stable but that they can change in time.
- For **each attribute** in our dimension tables, we must specify a **strategy to handle change**. In other words, **when an attribute value changes in the operational world**, how will we respond to the change in our dimensional models?
- Strategies:
  - Type 1: Overwrite the Value
  - Type 2: Add a Dimension Row
  - Type 3: Add a Dimension Column

Hybrid Slowly Changing Dimension Techniques

[Kimball, 2002]

Type 1: Overwrite the Value

- With the type 1 response, we **merely overwrite the old attribute value** in the dimension row, **replacing it with the current value**. In so doing, the attribute always reflects the most recent assignment.
- Consider on an electronic retailer one specific product “IntelliKidz 1.0” software.

Product Key	Product Description	Department	SKU Number (Natural Key)
12345	IntelliKidz 1.0	Education	ABC922-Z

- The Product key is a DW key. We do not consider changes!
- The SKU is a natural key. Its not like the other attributes. The natural key must remain inviolate.

A marketing person decides that IntelliKidz should be moved from the Education department to the Strategy department

[Kimball, 2002]

Type 1: Overwrite the Value

- **The updated row:**

Product Key	Product Description	Department	SKU Number (Natural Key)
12345	IntelliKidz 1.0	Strategy	ABC922-Z

- No dimension or fact table keys were modified when the IntelliKidz's department changed.
- The rows in the fact table still reference product key 12345, regardless of IntelliKidz's departmental location.
- Any previous report aggregated by this attribute **will change**, and no explanation is associated!. **Any pre-aggregate with this attribute should be recalculated**.

Appropriate only for corrections

[Kimball, 2002]

Type 2: Add a Dimension Row

- This the predominant technique for supporting this requirement when it comes to slowly changing dimensions. It represent prior history correctly.

Product Key	Product Description	Department	SKU Number (Natural Key)
12345	IntelliKidz 1.0	Education	ABC922-Z
25984	IntelliKidz 1.0	Strategy	ABC922-Z

- **Each of the separate surrogate keys** identifies a **unique product attribute profile** that was true for a span of time.
- The fact table is untouched.
  - Rows for IntelliKidz prior to the date of change, would reference product key 12345.
  - After that date, the IntelliKidz fact rows would have product key 25984 to reflect the move to the Strategy department until we are forced to make another type 2 change.

[Kimball, 2002]

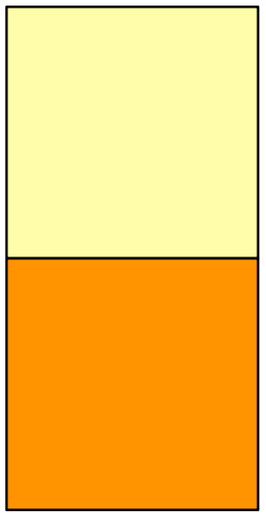
Type 2: Add a Dimension Row

Product Key	Product Description	Department	SKU Number (Natural Key)
12345	IntelliKidz 1.0	Education	ABC922-Z
25984	IntelliKidz 1.0	Strategy	ABC922-Z

If we constrain only on the **department attribute**, then we very **precisely differentiate between the two product profiles**.

If we constrain only on the **product description**, that is, IntelliKidz 1.0, then the query automatically will **fetch both IntelliKidz product dimension rows** and automatically join to the fact table for the complete product history.

If we need to **count the number of products** correctly, then we would just **use the SKU natural key attribute** as the basis of the distinct count rather than the surrogate key



Fact Table

Rows refer to product key 12345

Rows refer to product key 25984

[Kimball, 2002]

Type 2: Add a Dimension Row

- Include an **effective date stamp on a dimension row with type 2 changes**.
  - The **date stamp** would refer to the **moment when the attribute values in the row become valid** (or invalid in the case of expiration dates).
  - **Effective and expiration date attributes are necessary in the staging area** because we'd need to know which surrogate key is valid when we're loading historical fact records.
  - In the dimension table, these date stamps are helpful extras that are not required for the basic partitioning of history.

- **Previous report** (i.e., considering only existing rows before the change) aggregated by this attribute will **remain valid**. Any pre-aggregate with this attribute is OK.

[Kimball, 2002]

Type 2: Add a Dimension Row

- The dates support very precise time slicing of the dimension by itself.
  - The row **effective date** is the **first date the descriptive profile is valid**.
  - The row **expiration date** would be **one day less than the row effective date for the next assignment**, or **the date the product was retired** from the catalog.
- We could determine what the product catalog looked like as of December 31, 2001, by constraining a product table query to retrieve all rows where the row effective date to less than or equal to December 31, 2001, and the row expiration date to greater than or equal to December 31, 2001.
- One type 2 important advantage is that we can **gracefully track as many dimension changes as required**.
- **one downside** of this approach is **accelerated dimension table growth**. Hence it may be an inappropriate technique for dimension tables that already exceed a million rows!

[Kimball, 2002]

Type 2: Add a Dimension Row

- Notes:
  - A flag indicating the current effective row
  - It is essential to careful **managing the surrogate keys in the staging area** and to maintain a full copy with the dates on the staging area.
  - The **ETL should be aware on which attributes are dealt with this technique**.
  - How to **detect which records** from the operational system **were changed**
    - CRC over all attributes to detect change

[Kimball, 2002]

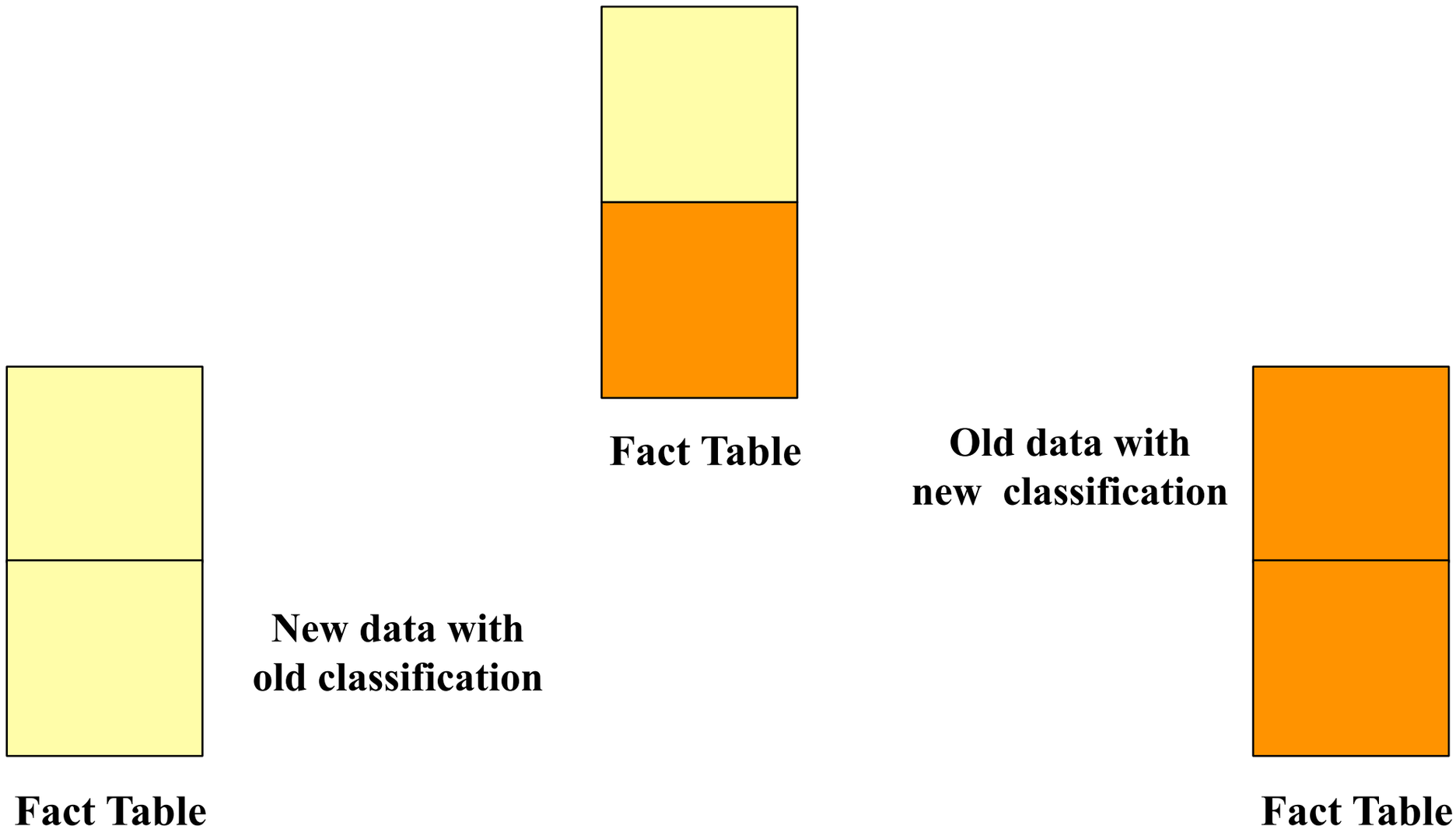
Type 3: Add a Dimension Column

- **Motivation**
  - Sometimes we want the ability to **see fact data** **as if the change never occurred**.
  - This happens most frequently with sales force reorganizations. District boundaries have been redrawn, but some users still want the ability to see **today's sales** in terms of **yesterday's district lines** just to see how they would have done under the old organizational structure. For a few transitional months, there may be a desire to track history in terms of the new district names and conversely to track new data in terms of old district names.
  - A type 2 response won't support this requirement.

[Kimball, 2002]

Type 3: Add a Dimension Column

- **Motivation**
  - Sometimes we want the ability to **see fact data as if the change never occurred**.



[Kimball, 2002]

Type 3: Add a Dimension Column

- With a type 3 response, we do not issue a new dimension row, but rather we **add a new column to capture** the attribute change.
  - In the case of IntelliKidz:
    - We alter the product dimension table to add a **prior department** attribute.
    - We populate this new column with the existing department value (Education).
    - We overwrite to reflect the current value (Strategy).

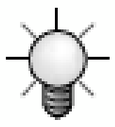
Product Key	Product Description	Department	Prior Department	SKU Number (Natural Key)
12345	IntelliKidz 1.0	Strategy	Education	ABC922-Z

[Kimball, 2002]

Type 3: Add a Dimension Column

- Warning: **all existing reports** and queries **switch** over to the new department description immediately.
  - Its is a similar effect on type 1!
  - We can still **report on the old department value** by querying on the **prior department** attribute.

Product Key	Product Description	Department	Prior Department	SKU Number (Natural Key)
12345	IntelliKidz 1.0	Strategy	Education	ABC922-Z



The type 3 slowly changing dimension technique allows us to see new and historical fact data by either the new or prior attribute values.

[Kimball, 2002]

Type 3: Add a Dimension Column

- Type 3 is appropriate when there's a strong need to support **two views of the world simultaneously**. This often occurs when the change or redefinition is soft or when the **attribute is a human-applied label rather** than a **physical characteristic**. Although the change has occurred, it is still logically possible to act as if it has not
- A type 3 response is **inappropriate** if you want to **track the impact of numerous intermediate attribute values**.
  - Obviously, there are serious implementation and usage limitations to creating attributes that reflect the prior minus 1, prior minus 2, and prior minus 3 states of the world, so we give up the ability to analyze these intermediate values

[Kimball, 2002]

Hybrid Slowly Changing Dimension Techniques

- **Predictable Changes with Multiple Version Overlays**
- Scenario: sales organization realignments
  - Consider the situation where a **sales organization** **revises the map of its sales districts** on an **annual basis**. Over a 5-year period, the sales organization is reorganized five times.
  - More complex set of requirements, including the following capabilities:
    - Report **each year's sales** using the district map for **that year**.
    - Report **each year's sales** using a district map from an **arbitrary different year**.
    - Report an **arbitrary span of years' sales** using a single district map from **any chosen year**. The most common version of this requirement would be to report the complete span of fact data using the **current district map**.

[Kimball, 2002]

Hybrid Slowly Changing Dimension Techniques

- We take advantage of the regular, predictable nature of these changes by generalizing the type 3 approach to have five versions of the district attribute.

Sales Rep Dimension
Sales Rep Key
Sales Rep Name
Sales Rep Address...
Current District
District 2001
District 2000
District 1999
District 1998
... and more

- **Current District**: This attribute will be used most frequently; we don't want to modify our existing queries and reports to accommodate next year's change.
  - When the districts are redrawn next, we'd alter the table to add a district 2002 attribute. We'd populate this column with the current district values and then overwrite the current attribute with the 2003 district assignments.

[Kimball, 2002]

Hybrid Slowly Changing Dimension Techniques

Unpredictable Changes with Single-Version Overlay

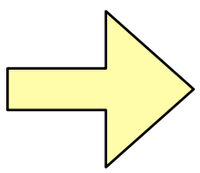
- If it is necessary to **preserve historical accuracy surrounding unpredictable attribute** changes **while supporting the ability to report historical data** according to the current values.
- In the case of the electronics retailer's product dimension, we would have two department attributes on each row:
  - The **current department** column represents the current assignment;
  - The **historical department** column represents the historically accurate department attribute value.

Product Key	Product Description	Current Department	Historical Department	SKU Number (Natural Key)
12345	IntelliKidz 1.0	Education	Education	ABC922-Z

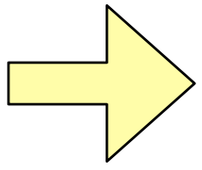
[Kimball, 2002]

Hybrid Slowly Changing Dimension Techniques

Product Key	Product Description	Current Department	Historical Department	SKU Number (Natural Key)
12345	IntelliKidz 1.0	Education	Education	ABC922-Z



Product Key	Product Description	Current Department	Historical Department	SKU Number (Natural Key)
12345	IntelliKidz 1.0	Strategy	Education	ABC922-Z
25984	IntelliKidz 1.0	Strategy	Strategy	ABC922-Z



Product Key	Product Description	Current Department	Historical Department	SKU Number (Natural Key)
12345	IntelliKidz 1.0	Critical Thinking	Education	ABC922-Z
25984	IntelliKidz 1.0	Critical Thinking	Strategy	ABC922-Z
31726	IntelliKidz 1.0	Critical Thinking	Critical Thinking	ABC922-Z

[Kimball, 2002]

Hybrid Slowly Changing Dimension Techniques

- **Unpredictable Changes with Single-Version Overlay**
  - We issue a new row to capture the change (type 2) and add a new column to track the current assignment (type 3), where subsequent changes are handled as a type 1 response.

[Kimball, 2002]

Large Changing Customer Dimensions

Large Changing Customer Dimensions

- Multimillion-row customer dimensions present two unique challenges that warrant special treatment:
  - Even if a clean, flat dimension table has been implemented, it generally takes too long to constrain or browse among the relationships in such a big table.
  - In addition, it is difficult to use the techniques used for SCD for tracking changes in these large dimensions.
    - The use of the **type 2 slowly** changing dimension technique and **add more rows** to a customer dimension that already has millions of rows is unfeasible.

[Kimball, 2002]

Large Changing Customer Dimensions

- The solution is to break off **frequently analyzed** or **frequently changing attributes** into a separate dimension, referred to as a ***mini-dimension***.
  - For example, we could create a separate mini-dimension for a package of **demographic attributes**, such as age, gender, number of children, and income level, presuming that these columns get used extensively. There would be one row in this mini-dimension for **each unique combination** of age, gender, number of children, and income level **encountered in the data**, not one row per customer.
  - **These columns** are the ones that are analyzed to **select an interesting subset** of the customer base. In addition, users **want to track changes to these attributes**. We leave behind more constant or less frequently queried attributes in the original huge customer table.

[Kimball, 2002]

Large Changing Customer Dimensions

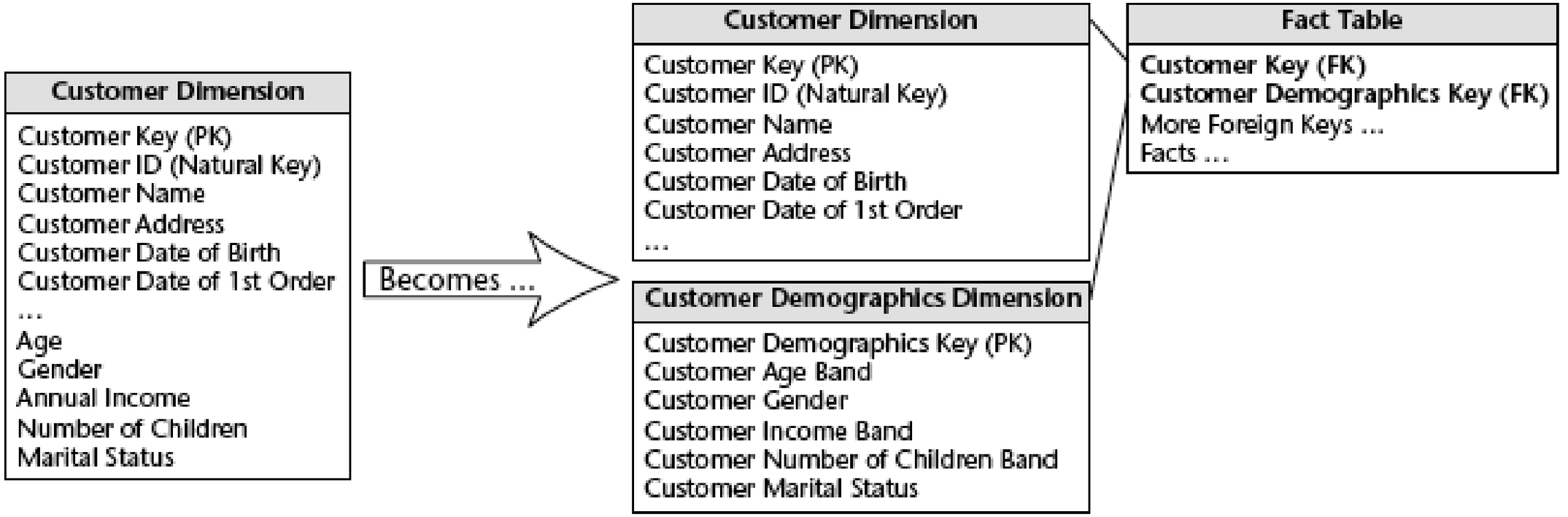
- When creating the demographic mini-dimension, **continuously variable attributes**, such as income and total purchases, **should be converted to banded ranges**.
  - Force the attributes in the mini-dimension to take on a relatively small number of discrete values and so the cardinality of the mini-dimension remains small.
  - The use of band ranges is probably the **most significant compromise** associated with the mini-dimension technique because once we decide on the value bands, it is quite impractical to change to a different set of bands at a later time.

DEMOGRAPHIC KEY	AGE	GENDER	INCOME LEVEL
1	20-24	Male	<$20,000
2	20-24	Male	$20,000-$24,999
3	20-24	Male	$25,000-$29,999
18	25-29	Male	$20,000-$24,999
19	25-29	Male	$25,000-$29,999

[Kimball, 2002]

Large Changing Customer Dimensions

- Every time we build a fact table row, we include **two foreign keys related to the customer**: the regular customer dimension key and the mini-dimension demographics key
  - The demographics key should be part of the fact table's set of foreign keys in order to provide efficient access to the fact table through the demographics attributes.
  - This design delivers browsing and constraining performance benefits by providing a smaller point of entry to the facts.



[...l, 2002]

Large Changing Customer Dimensions

- Managing the **change**:
  - if one customers, John Smith, **was 24 years old** with an income of $24,000, we'd begin by assigning **demographics key 2** when loading the fact table.
  - If John has a **birthday several weeks later**, we'd assign **demographics key 18** when the fact table was next loaded.
  - The demographics key on the earlier fact table rows for John would not be changed. In this manner, the fact table tracks the age change.
  - We'd continue to assign demographics key 18 when the fact table is loaded until there's another change in John's demographic profile.
  - If John **receives a raise to $26,000** several months later, a **new demographics key** would be reflected in the next fact table load.

[Kimball, 2002]

Large Changing Customer Dimensions

- Managing the change (cont):
  - Historical demographic profiles for each customer can be constructed at any time by referring to the fact table and picking up the simultaneous customer key and its contemporary demographics key.

[Kimball, 2002]

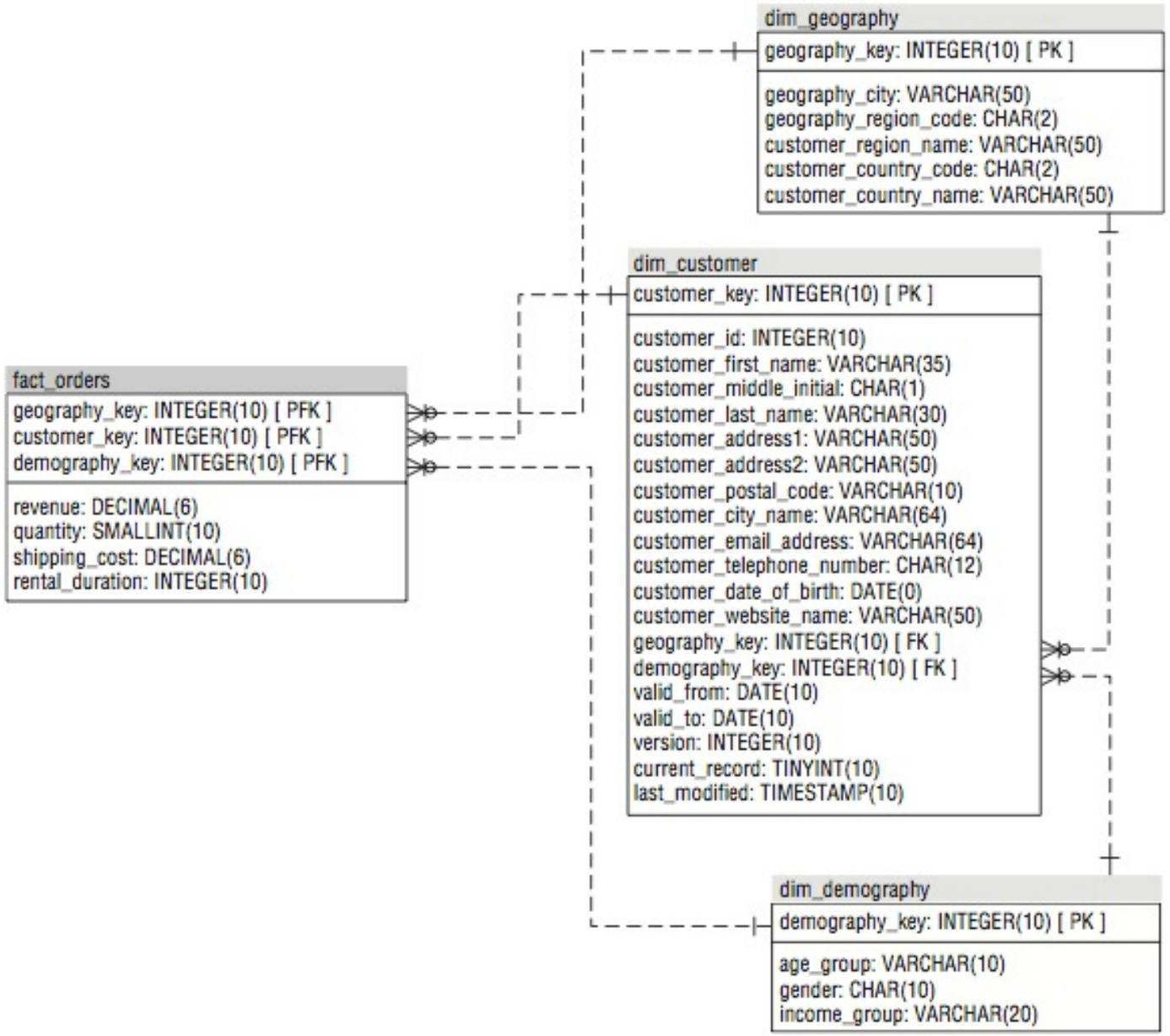
Large Changing Customer Dimensions

- Demographic mini-dimension as an **outrigger** of customer dimension
  - Users may want to know how many female customers live in Dade County **by** age bracket. Counts such as these are extremely common with customer segmentation and profiling.
  - Rather than forcing any analysis that **combines solely customer and demographic data to link through the fact table**, the **most recent value of the demographics key also can exist as a foreign key on the customer dimension table**. In this case, we refer to the **demographics table as a customer dimension outrigger**.
  - If you embed the most recent demographics key in the customer dimension, you must treat it as a type 1 attribute.

[Kimball, 2002]

Large Changing Customer Dimensions

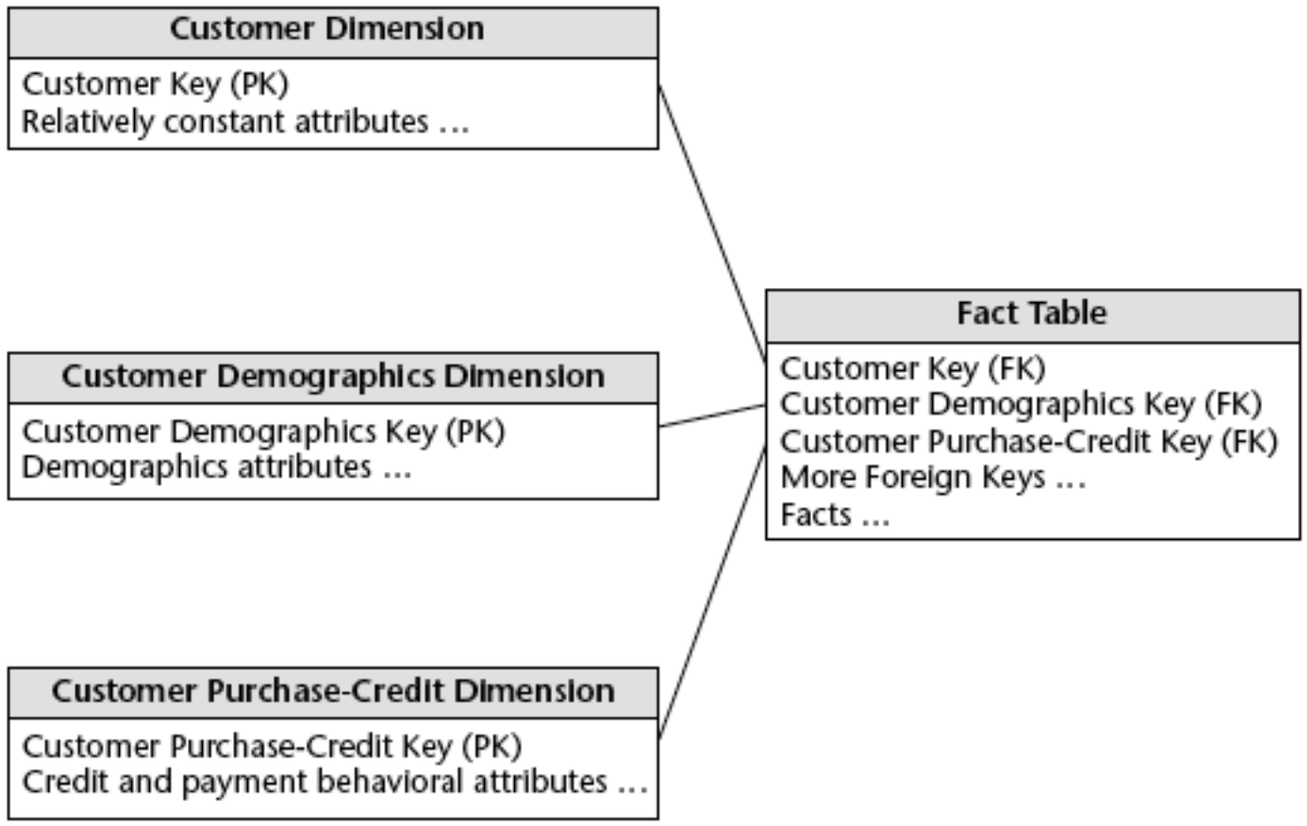
- Demographic mini-dimension as an **outrigger** of customer dimension



[Kimball, 2002]

Large Changing Customer Dimensions

The best approach for efficiently browsing and tracking changes of key attributes in really huge dimensions is to break off one or more minidimensions from the dimension table, each consisting of small clumps of attributes that have been administered to have a limited number of values.



[Kimball, 2002]

Further Reading and Summary

Notes

- Understand the discussion between Multiple- versus Single-Transaction Fact Tables
  - Consider the Procurement example (Multiple) and the Promotion example on sales (Single)
- The role-playing on date dimension
- The concept and the possible techniques for the Slowly Changing Dimensions.
- The concept of Large Changing Customer Dimensions: The issues and the approach based on mini-dimensions.

[Kimball, 2002]

Further Readings

- **Further Readings**
  - The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition), Ralph Kimball, Margy Ross. 2002
    - From page 89 to 105
    - From page 154 to 157
    - From page 159 to 160

[Kimball, 2002]