Bases de Dados e Armazéns de Dados

Departamento de Engenharia Informática (DEI/ISEP)
Paulo Oliveira
pjo@isep.ipp.pt

1

Dimensional Model for Inventory Periodic Snapshot

34

Modelling Inventory Periodic Snapshot

1. Business process

- Inventory
  - Optimized inventory levels can have an impact on profitability
  - Making sure the **right product** is in the **right store** at the **right time** minimizes out-of-stocks and reduces overall inventory carrying costs

2. Granularity

- Daily quantity on stock by product and store

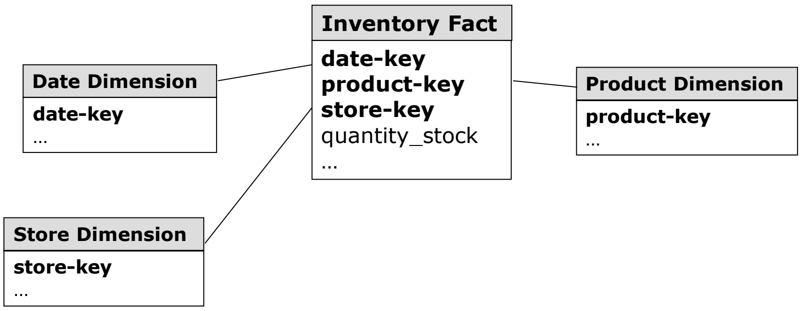
3. Dimensions

- Date
- Product
- Store

4. Facts

- Quantity on stock
- …

Inventory Periodic Snapshot Schema



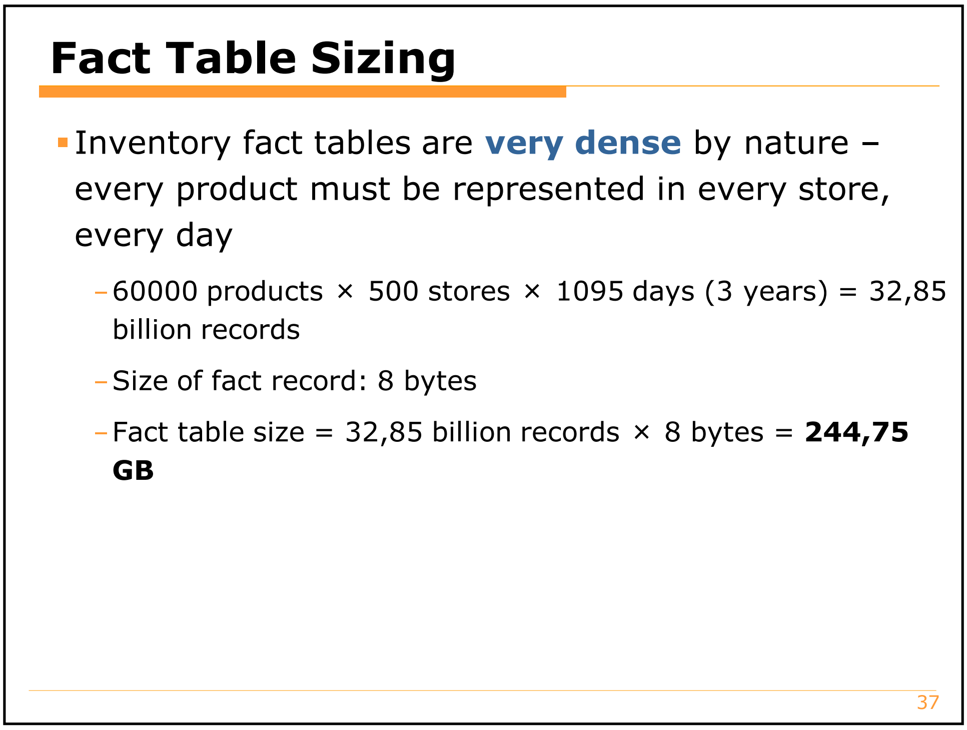
Inventory Fact
date-key
product-key
store-key
quantity_stock
…

Date Dimension
date-key
…

Product Dimension
product-key
…

Store Dimension
store-key
…

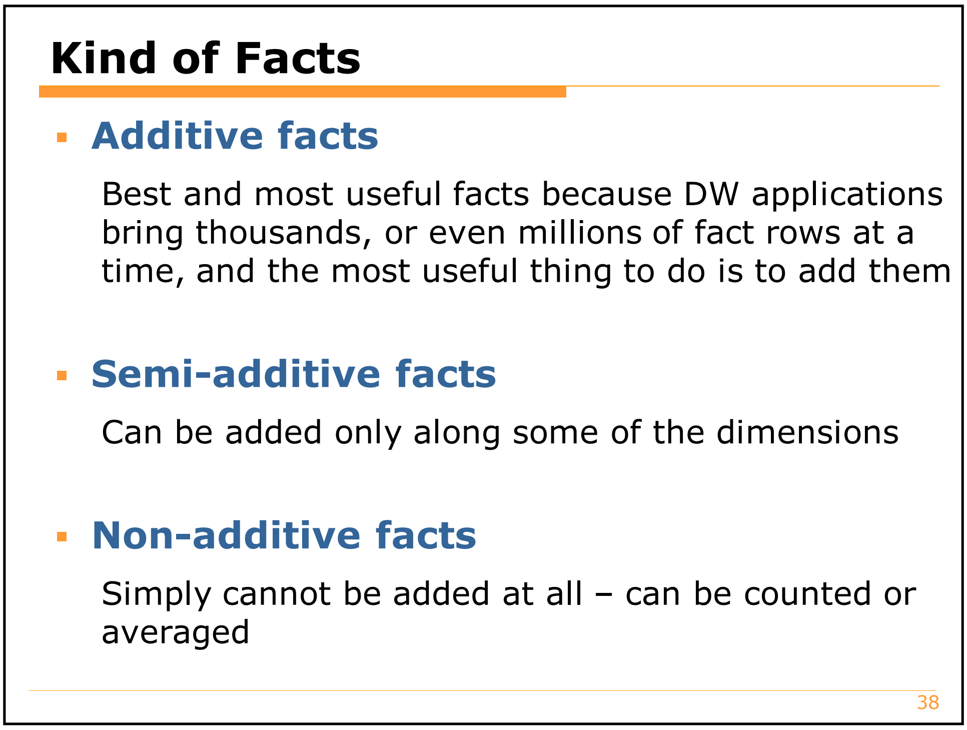
- Dimensions are the same as on the grocery store schema
- Dimension tables must have additional attributes useful for inventory analysis, such as:
  - Product dimension: Minimum reorder quantity
  - Store dimension: Frozen and refrigerated storage square footages



Fact Table Sizing

- Inventory fact tables are **very dense** by nature – every product must be represented in every store, every day
  - 60000 products × 500 stores × 1095 days (3 years) = 32,85 billion records
  - Size of fact record: 8 bytes
  - Fact table size = 32,85 billion records × 8 bytes = **244,75 GB**

37



Kind of Facts

- **Additive facts**

  Best and most useful facts because DW applications bring thousands, or even millions of fact rows at a time, and the most useful thing to do is to add them

- **Semi-additive facts**

  Can be added only along some of the dimensions

- **Non-additive facts**

  Simply cannot be added at all – can be counted or averaged

38

Kind of Facts

- **Additive facts**

Product Sales			
Product	Day 1	Day 2	Product Total
P1	200 €	300 €	500 €
P2	30 €	20 €	50 €
P3	100 €	300 €	400 €
P4	40 €	60 €	100 €
P5	300 €	100 €	400 €
Day Total	670 €	780 €	1 450 €

- **Semi-additive facts**

Customer Balance			
Customer	Day 1	Day 2	Customer Total
C1	200 €	300 €	500 €
C2	300 €	100 €	400 €
C3	100 €	100 €	200 €
C4	400 €	200 €	600 €
C5	30 €	40 €	70 €
Day Total	1 030 €	740 €	

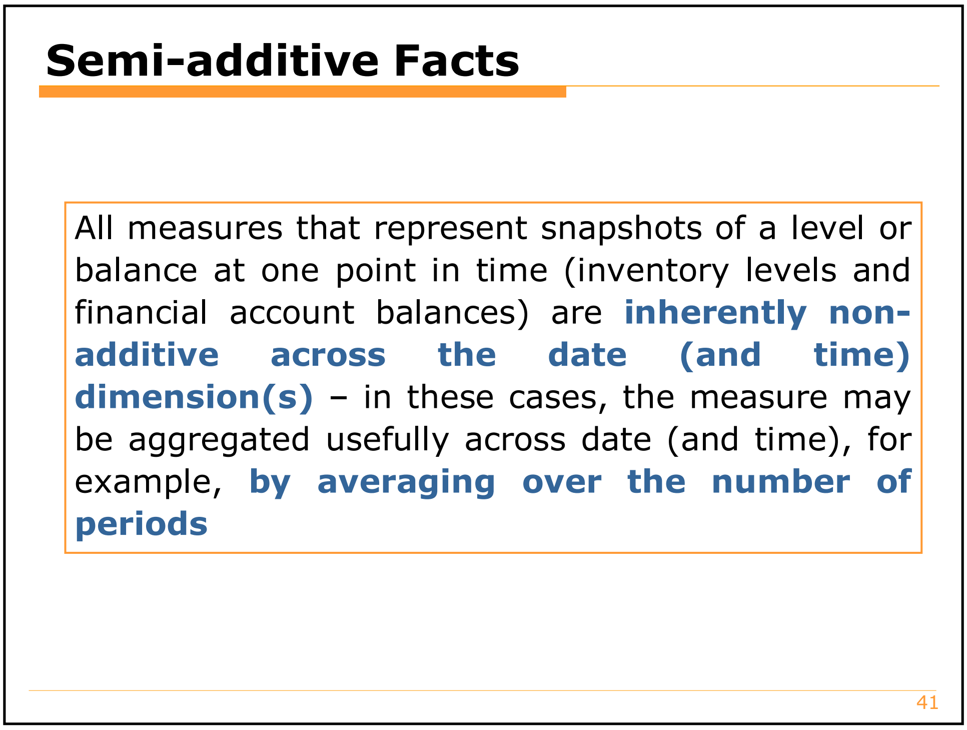
- **Non-additive facts**

Sales Margin (sales profit / value sold)			
Product	Day 1	Day 2	Product Total
P1	33%	35%	68%
P2	27%	27%	54%
P3	45%	43%	88%
P4	17%	16%	33%
P5	8%	12%	20%
Day Total	130%	133%	263%

Semi-additive Facts

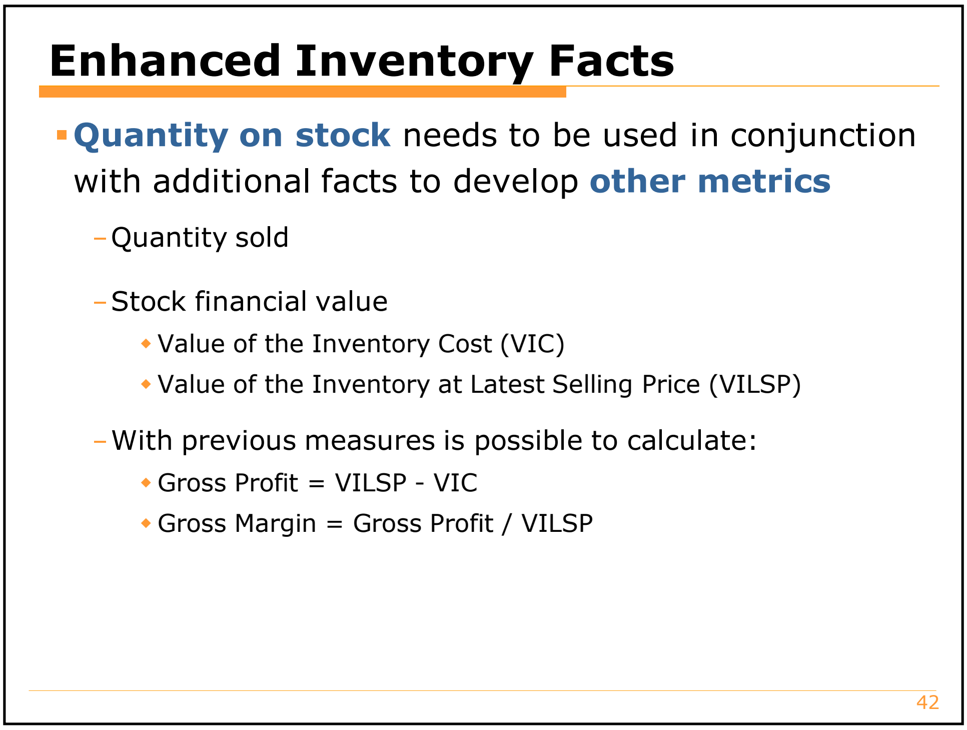
Store	S1	S2
	2 un. P1	3 un. P1
	3 un. P2	1 un. P2
Day 1	1 un. P3	3 un. P3
	4 un. P4	2 un. P4
	3 un. P5	4 un. P5
	1 un. P1	2 un. P2
Day 2	3 un. P2	1 un. P2
	4 un. P3	5 un. P3
	2 un. P4	0 un. P4
	1 un. P5	1 un. P5

- Given a product and a date the **quantity on stock** can be summed-up by stores
- Given a product and a store **it is not possible to sum-up by date**
- Possible aggregations on store S1 for product P4:
  - Average = (4+2)/2=3 → **average over nr. of periods**
  - Min=2
  - Max=4



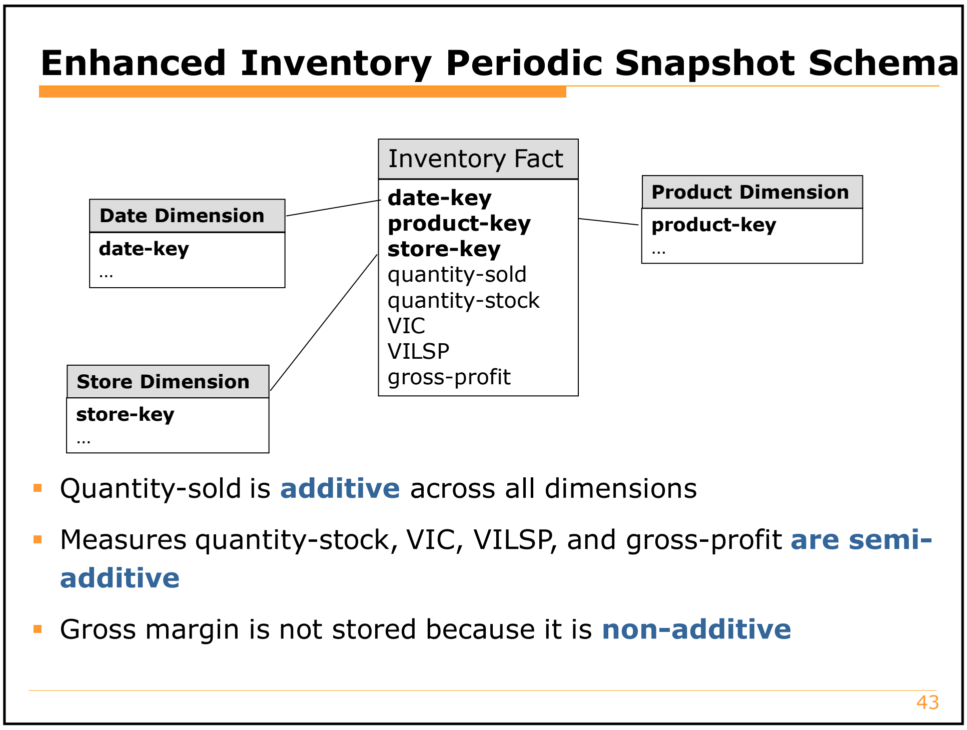
Semi-additive Facts

All measures that represent snapshots of a level or balance at one point in time (inventory levels and financial account balances) are **inherently non-additive across the date (and time) dimension(s)** – in these cases, the measure may be aggregated usefully across date (and time), for example, **by averaging over the number of periods**



Enhanced Inventory Facts

- **Quantity on stock** needs to be used in conjunction with additional facts to develop **other metrics**
  - Quantity sold
  - Stock financial value
    - Value of the Inventory Cost (VIC)
    - Value of the Inventory at Latest Selling Price (VILSP)
  - With previous measures is possible to calculate:
    - Gross Profit = VILSP - VIC
    - Gross Margin = Gross Profit / VILSP



Enhanced Inventory Periodic Snapshot Schema

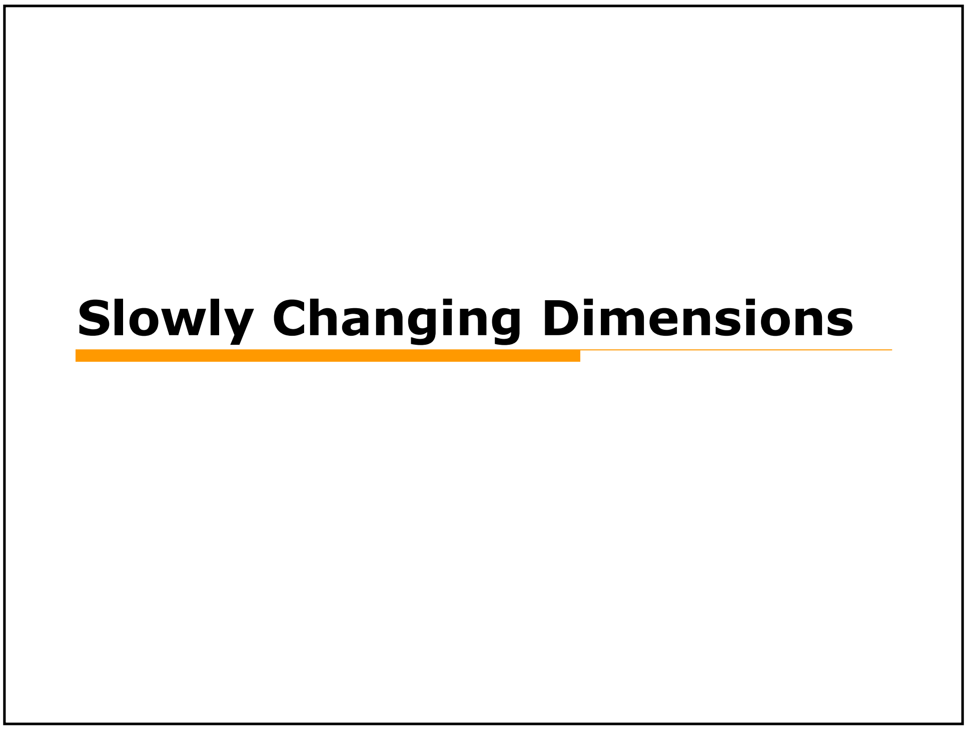
Inventory Fact
- **date-key**
- **product-key**
- **store-key**
- quantity-sold
- quantity-stock
- VIC
- VILSP
- gross-profit

Date Dimension
- **date-key**
- …

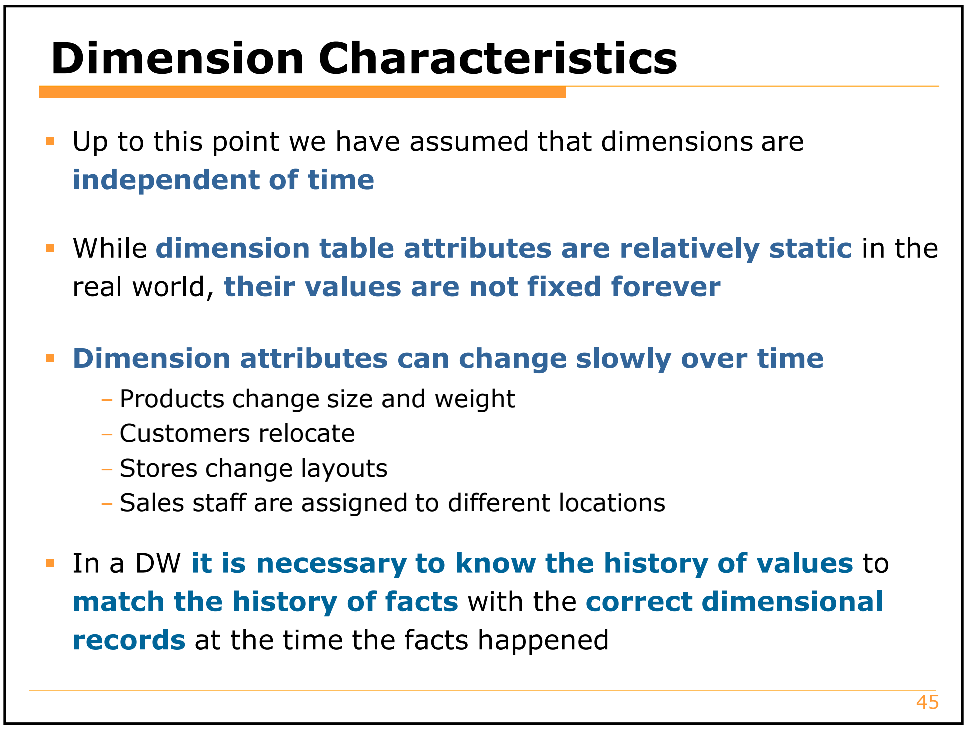
Product Dimension
- **product-key**
- …

Store Dimension
- **store-key**
- …

- Quantity-sold is **additive** across all dimensions
- Measures quantity-stock, VIC, VILSP, and gross-profit **are semi-additive**
- Gross margin is not stored because it is **non-additive**

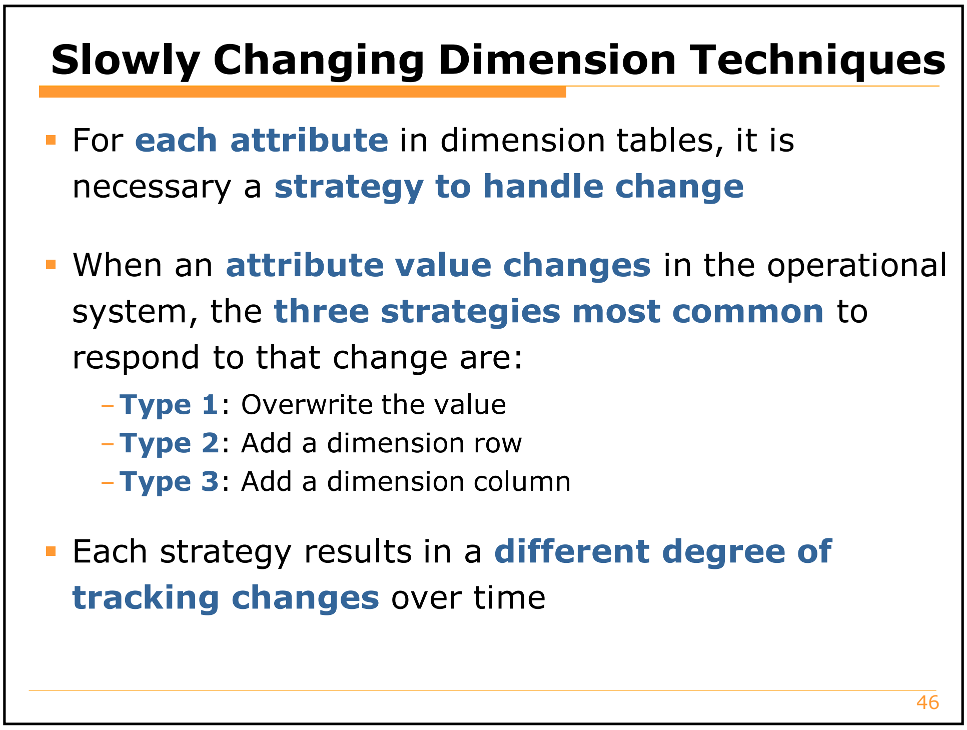


Slowly Changing Dimensions



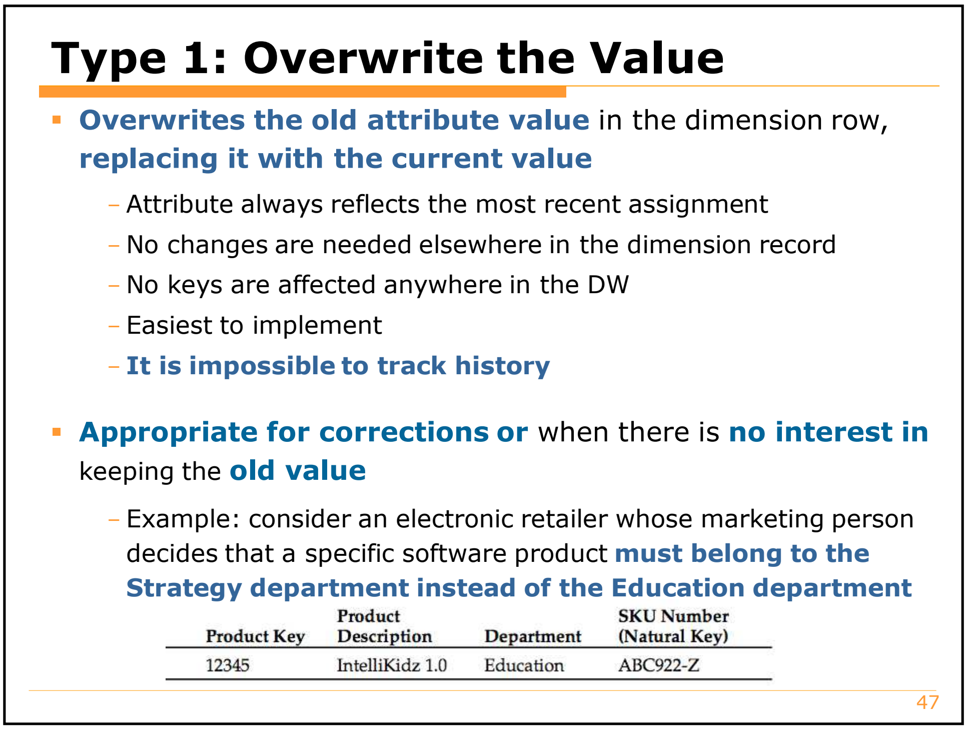
Dimension Characteristics

- Up to this point we have assumed that dimensions are **independent of time**
- While **dimension table attributes are relatively static** in the real world, **their values are not fixed forever**
- **Dimension attributes can change slowly over time**
  - Products change size and weight
  - Customers relocate
  - Stores change layouts
  - Sales staff are assigned to different locations
- In a DW **it is necessary to know the history of values** to **match the history of facts** with the **correct dimensional records** at the time the facts happened



Slowly Changing Dimension Techniques

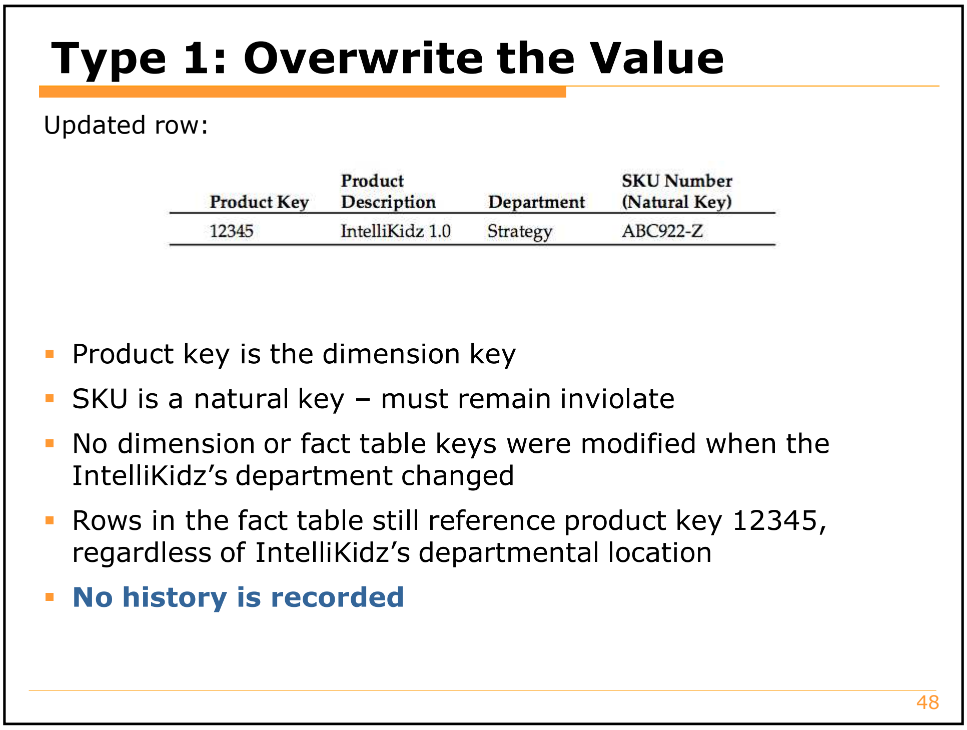
- For **each attribute** in dimension tables, it is necessary a **strategy to handle change**
- When an **attribute value changes** in the operational system, the **three strategies most common** to respond to that change are:
  - **Type 1**: Overwrite the value
  - **Type 2**: Add a dimension row
  - **Type 3**: Add a dimension column
- Each strategy results in a **different degree of tracking changes** over time



Type 1: Overwrite the Value

- **Overwrites the old attribute value** in the dimension row, **replacing it with the current value**
  - Attribute always reflects the most recent assignment
  - No changes are needed elsewhere in the dimension record
  - No keys are affected anywhere in the DW
  - Easiest to implement
  - **It is impossible to track history**
- **Appropriate for corrections or** when there is **no interest in** keeping the **old value**
  - Example: consider an electronic retailer whose marketing person decides that a specific software product **must belong to the Strategy department instead of the Education department**

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



Type 1: Overwrite the Value

Updated row:

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

- Product key is the dimension key
- SKU is a natural key – must remain inviolate
- No dimension or fact table keys were modified when the IntelliKidz’s department changed
- Rows in the fact table still reference product key 12345, regardless of IntelliKidz’s departmental location
- **No history is recorded**

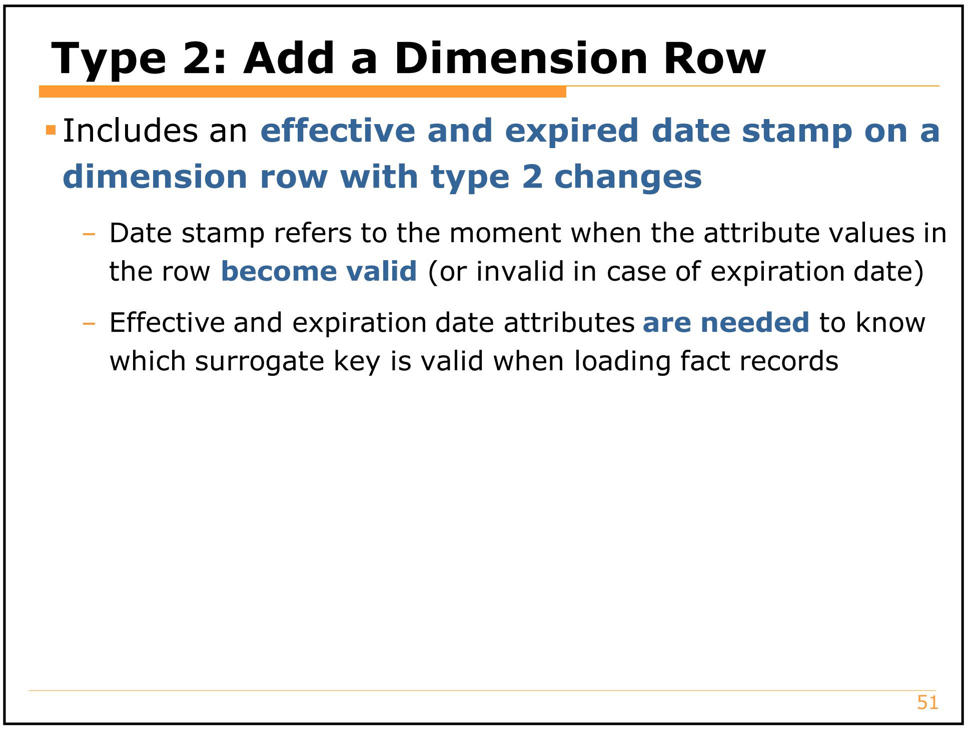
Type 2: Add a Dimension Row

- **Create an additional dimension record at the time of the change with the new attribute value**
  - Segments history very accurately between the old description and the new description
- **Predominant technique** for supporting slowly changing dimensions
- **Represents prior history correctly**
- **Gracefully tracks as many dimension changes as required**
- **One downside is to accelerate dimension table growth**

Type 2: Add a Dimension Row

Product Key	Product Description	Department	SKU Number (Natural Key)	Effective Date	Expired Date
12345	IntelliKidz 1.0	Education	ABC922-Z	11/02/2019	19/09/2020
25984	IntelliKidz 1.0	Strategy	ABC922-Z	19/09/2020	

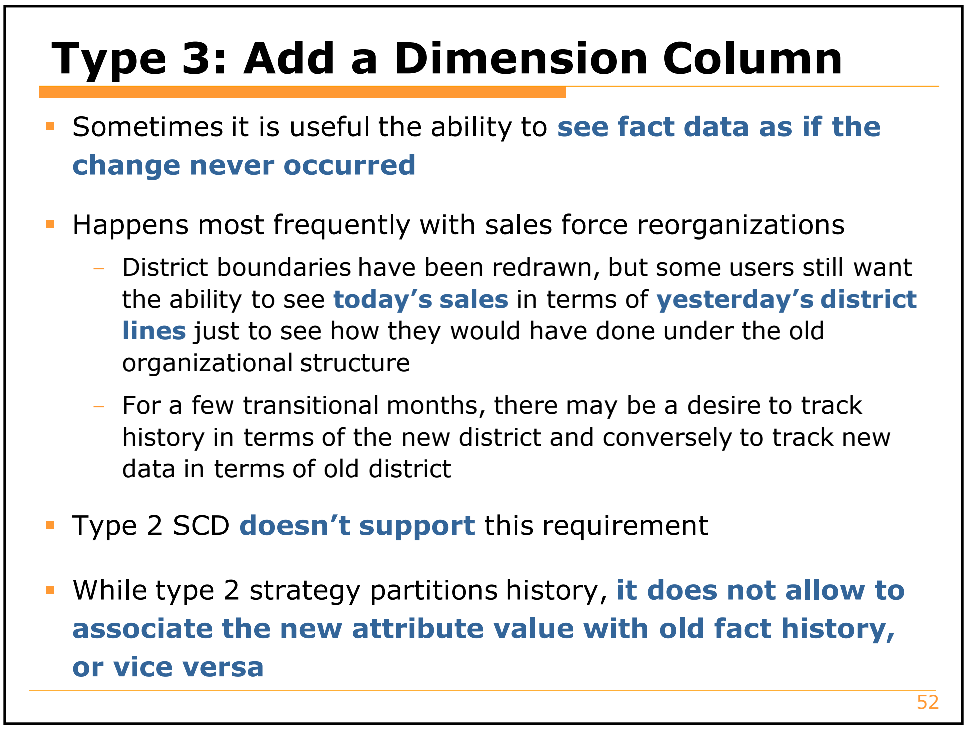
- Each of the separate surrogate keys identifies **a unique product attribute profile that was true for a span of time**
- In the fact table
  - Fact rows for IntelliKidz prior to the date of change, would reference product key 12345
  - When the product was moved to the Strategy department, fact rows reference product key 25984 to reflect the change
- Constrains
  - On the **department attribute**, precisely differentiate between the two product profiles
  - On the **product description**, brings the complete product history
  - On the count distinct **SKU number** natural key, brings the correct number of products



Type 2: Add a Dimension Row

- Includes an **effective and expired date stamp on a dimension row with type 2 changes**
  - Date stamp refers to the moment when the attribute values in the row **become valid** (or invalid in case of expiration date)
  - Effective and expiration date attributes **are needed** to know which surrogate key is valid when loading fact records

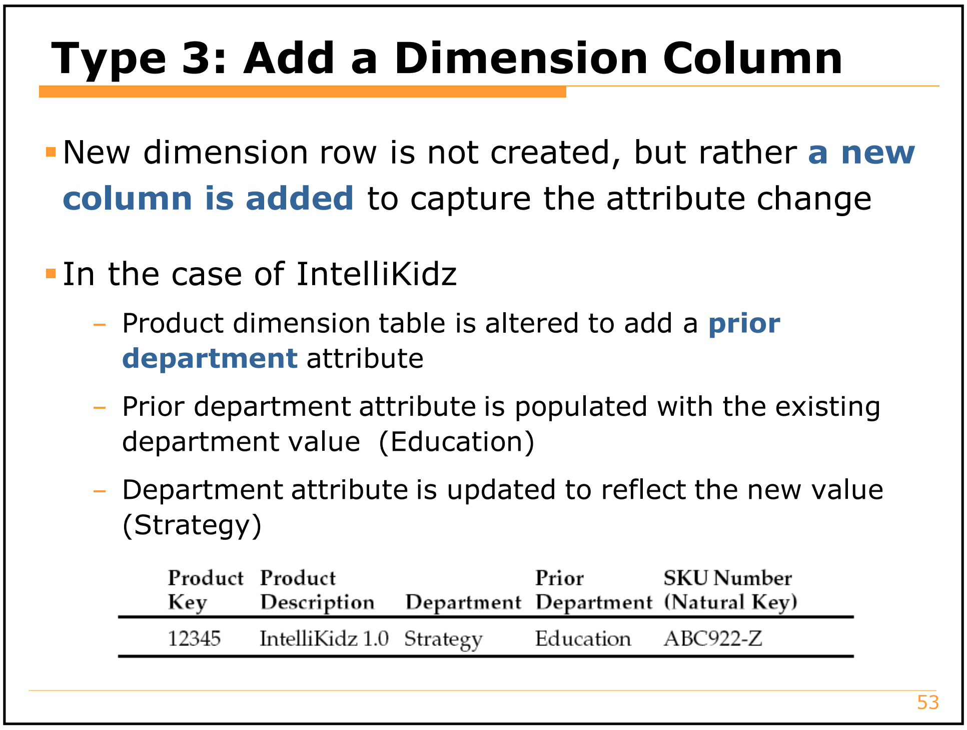
51



Type 3: Add a Dimension Column

- Sometimes it is useful the ability to **see fact data as if the change never occurred**
- 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 and conversely to track new data in terms of old district
- Type 2 SCD **doesn't support** this requirement
- While type 2 strategy partitions history, **it does not allow to associate the new attribute value with old fact history, or vice versa**

52

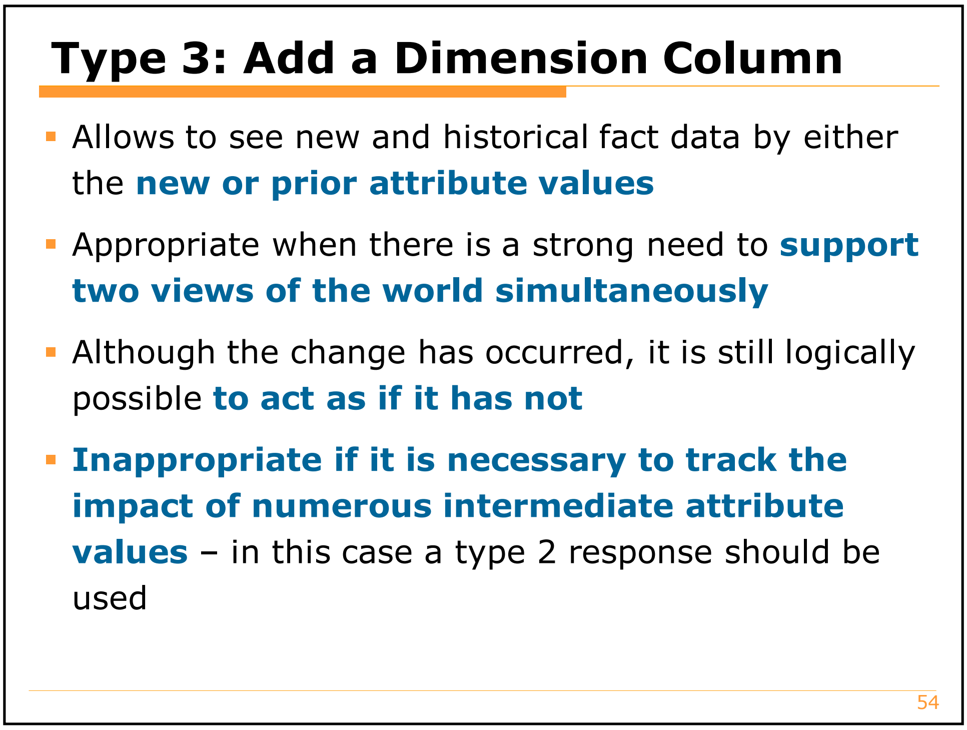


Type 3: Add a Dimension Column

- New dimension row is not created, but rather **a new column is added** to capture the attribute change
- In the case of IntelliKidz
  - Product dimension table is altered to add a **prior department** attribute
  - Prior department attribute is populated with the existing department value (Education)
  - Department attribute is updated to reflect the new value (Strategy)

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

53



Type 3: Add a Dimension Column

- Allows to see new and historical fact data by either the **new or prior attribute values**
- Appropriate when there is a strong need to **support two views of the world simultaneously**
- Although the change has occurred, it is still logically possible **to act as if it has not**
- **Inappropriate if it is necessary to track the impact of numerous intermediate attribute values** – in this case a type 2 response should be used

54