Inventory

Kimball & Ross, Chapter 3

Overview

- Value chain implications
- Inventory periodic snapshot model, transaction and accumulating snapshot models
- Semi-additive facts
- Enhanced inventory facts
- Data Warehouse bus architecture and matrix
- Conformed dimensions and facts

Value Chain

- The value chain identifies the natural, logical flow of an organization’s primary activities. See Fig. 3.1
- Operational source systems produce transactions or snapshots at each step in the value chain. They generate interesting performance metrics along the way.
- Each business process generates one or more fact tables.

A family of stars

- A dimensional model of a data warehouse for a large data warehouse consists of between 10 and 25 similar-looking star-join schemas. Each star join will have 5 to 15 dimensional tables.
- Conformed (shared) dimensions facilitate drillacross.
- A conformed dimension is a dimension that means the same thing with every possible fact table to which it is joined.
- Conformed dimensions are either identical or strict mathematical subsets of the most granular detailed dimension.

Value chains as families of star-join schemas

- There are two sides to the value chain
  - – the demand side - the steps needed to satisfy the customers’ demand for the product
  - – the supply side - the steps needed to manufacture the products from original ingredients or parts
- The chain consists of a sequence of inventory and flow star-join schema
- joining the different star-join schema is only possible when two sequential schema have a common, identical dimension
- Sometimes the represented chain can be extended beyond the bounds of the business itself

Value chains as families of star-join schemas

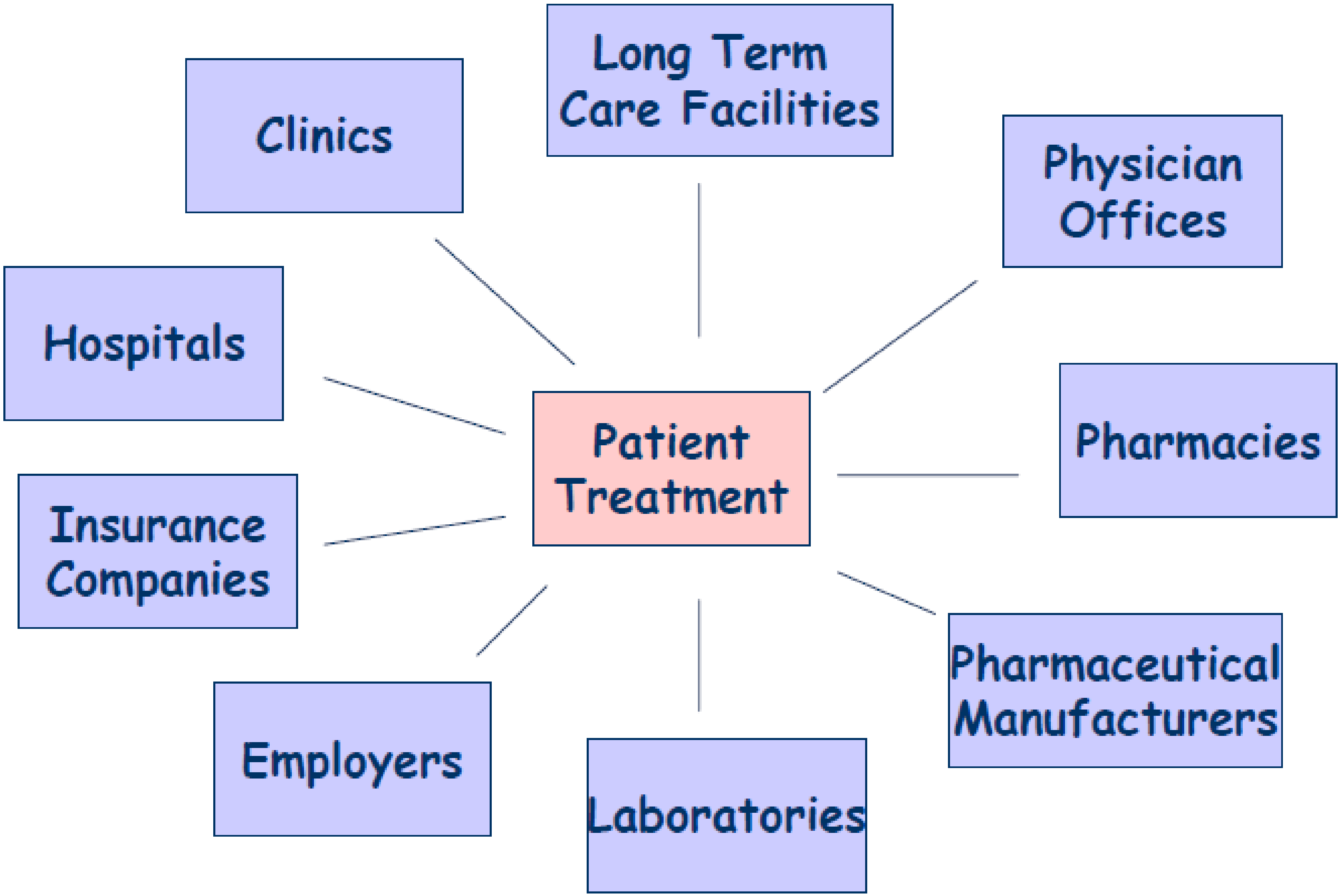
Supply Chain

- Row material production
- Ingredient purchasing
- Ingredient delivery
- Ingredient inventory
- Bill of materials
- Manufacturing process
control
- Manufacturing costs
- Packaging
- Trans-shipping to
  warehouse
- Finished goods inventory

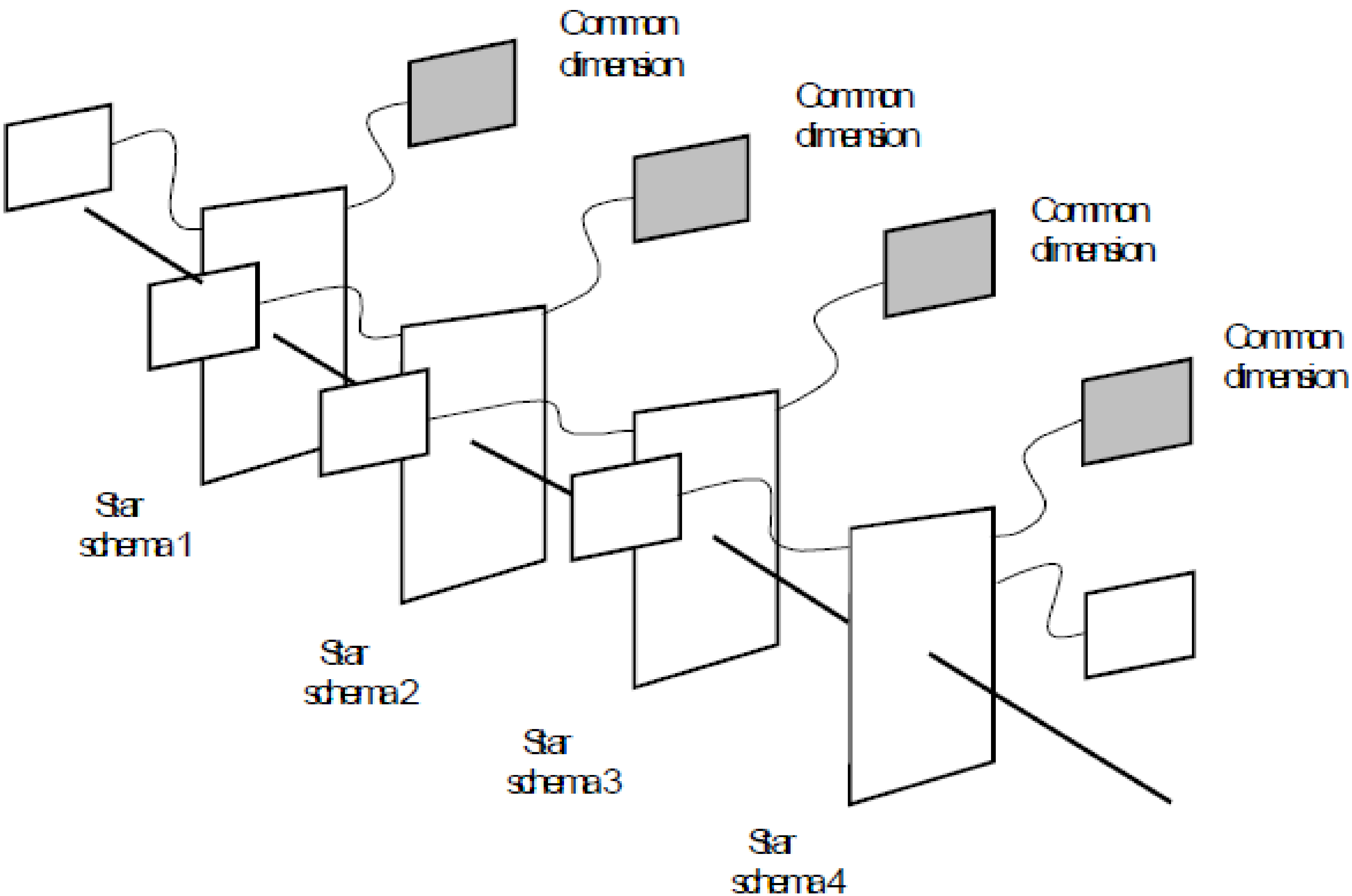
Demand Chain

- Finished goods inventory
- Manufacturing shipments
- Distributor inventory
- Distributor shipments
- Retail inventory
- Retail sales

Value circles



A family of stars



Inventory Models

- Inventory periodic snapshot
  - Inventory level of each product measured daily (or weekly) – represented as a separate row in a fact table
- Inventory transactions
  - As products move through the warehouse, all transactions with impact on inventory levels are recorded
- Inventory accumulating snapshot
  - One fact table row for each product updated as the product moves through the warehouse

Inventory Periodic Snapshot Model

- Business need
  - Analysis of daily quantity-on-hand inventory levels by product and store
- Business process
  - Retail store inventory
- Granularity
  - Daily inventory by product at each store
- Dimensions
  - Date, product, store
- Fact
  - Quantity on hand

Inventory Periodic Snapshot Model - Challenge

- Very dense (huge) fact table
  - As opposed to retail sales, which was sparse because only about 10% of products sell each day
- 60,000 items in 100 stores = 6,000,000 rows
- If 14 bytes per row: 84MB per day
- One-year period: 365 x 84MB = 30GB
- Solution: Reduce snapshot frequencies over time
  - Last 60 days at daily level
  - Weekly snapshots for historical data
  - For a 3-year period =208 snapshots vs. 3x365=1095 snapshots; reduction by a factor of 5

Semiadditive Facts

- Inventory levels (quantity on hand) are additive across products or stores, but NOT across dates = semi-additive facts
- Compare to **Retail Sales**:
  - once the product is sold it is not counted again
- Static level measurements (inventory, balances…) are not additive across date dimension; to aggregate over time use average over number of time periods.

Enhanced Inventory Facts

- Number of turns = total quantity sold / daily average quantity on hand
- Days' supply = final quantity on hand / average quantity sold
- Gross profit = value at latest selling price - value at cost
- Gross margin = gross profit / value at latest selling price
- GMROI (Gross Margin Return On Inventory)
  - GMROI = number of turns * gross margin
  - measures effectiveness of inventory investment
  - high = lot of turns and more profit, low = low turns and low profit
- Need additional facts:
  - quantity sold, value at cost, value at latest selling price
- GMROI is not additive and, therefore, is not stored in enhanced fact table. It is calculated from the constituent columns.

Inventory Transactions Model

- Record every transaction that affects inventory
  - Receive product
  - Place product into inspection hold
  - Release product from inspection hold
  - Return product to vendor due to inspection failure
  - Place product in bin
  - Authorize product for sale
  - Pick product from bin
  - Package product for shipment
  - Ship product to customer
  - Receive product from customer
  - Return product to inventory from customer return
  - Remove product from inventory

Inventory Transactions Model - Con't

- Dimensions: date, warehouse, product, vendor, inventory transaction type.
- The transaction-level fact table contains the most detailed information possible about the inventory.
- It is useful for measuring the frequency and timing of specific transaction types.
- It is impractical for broad data warehouse questions that span dates or products.
- To give a more cumulative view of a process, some form of snapshot table often accompanies a transaction fact table.

Inventory Accumulating Snapshot Model

- Build one record in the fact table for each product delivery to the warehouse
- Track disposition of a product until it leaves the warehouse
  - Receiving
  - Inspection
  - Bin placement
  - Authorization to sell
  - Picking
  - Boxing
  - Shipping
- The philosophy of the inventory accumulating snapshot fact table is to provide an updated status of the product shipment as it moves through above milestones.
- Rarely used in long-running, continuously replenished inventory processes.
- More on this in chapter 5.

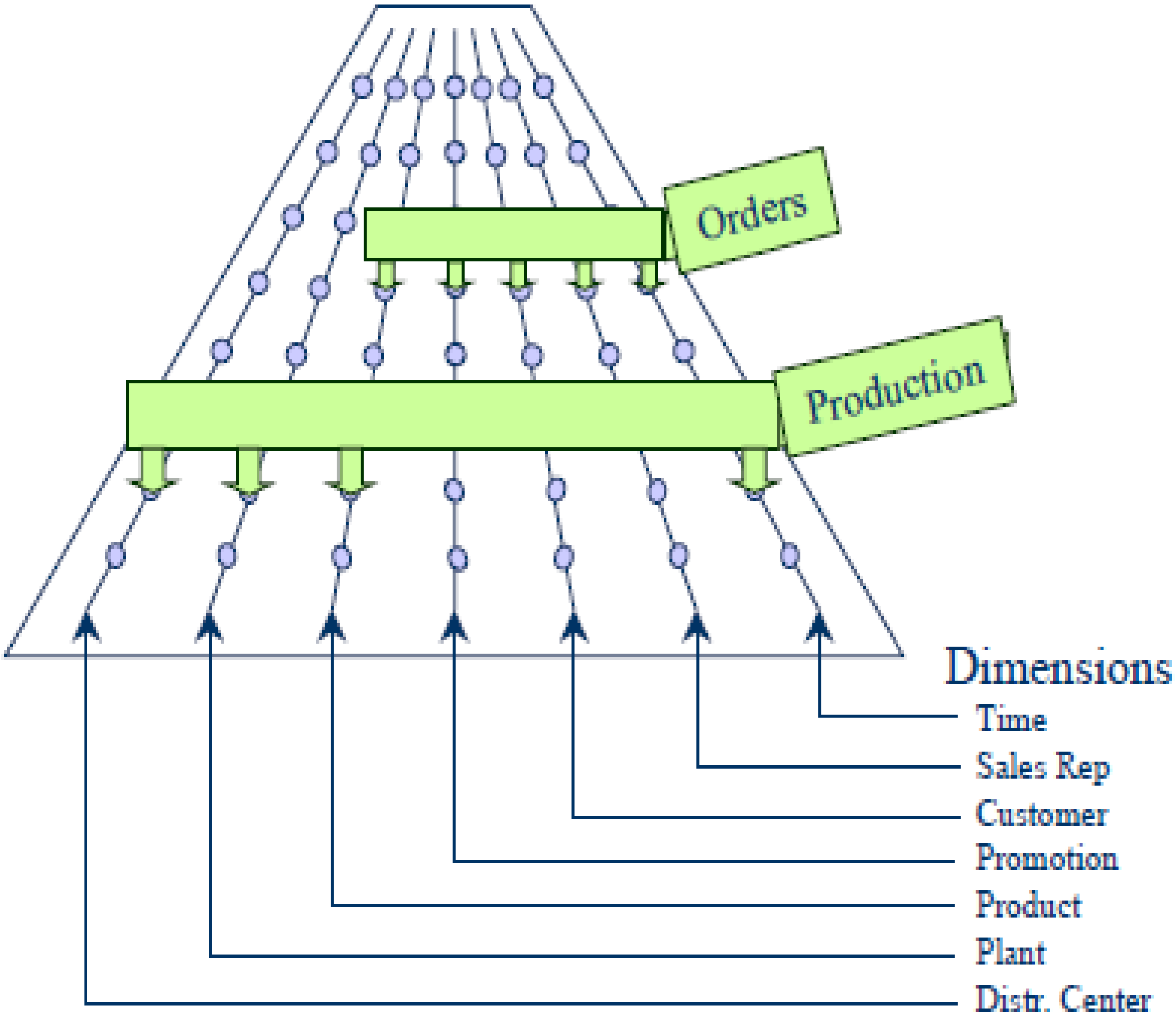
Value Chain Integration

- Both business and IT organizations are interested in value chain integration
- Desire to look across the business to better evaluate overall performance
- Data marts may correspond to different business processes
- Need to look consistently at dimensions shared between business processes
- Need an integrated data warehouse architecture
- If dimension table attributes in various marts are identical, each mart is queried separately; the results are then outer-joined based on a common dimension attribute = *drill across*

Data Warehouse Bus Architecture

- Cannot built the enterprise data warehouse in one step.
- Building isolated pieces will defeat consistency goal.
- Need an architected incremental approach
  → **data warehouse bus architecture**.
- See Fig. 3.7
- By defining a standard bus interface for the data warehouse environment, separate data marts can be implemented by different groups at different times. The separate data marts can be plugged together and usefully coexist if they adhere to the standard.

The Data Warehouse Bus



Data Warehouse Bus Architecture – Cont'd

- During architecture phase, team designs a master suite of standardized dimensions and facts that have uniform interpretation across the enterprise.
- Separate data marts are then developed adhering to this architecture.

Data Warehouse Bus Matrix

- See Figure 3.8
- The rows of the bus matrix correspond to business processes → data marts
- Separate rows should be created if:
  - the sources are different,
  - the processes are different, or
  - a row represents more than what can be tackled in a single implementation iteration.
- Creating the DW bus matrix is a very important up-front deliverable of a DW implementation. The DW bus matrix is a hybrid resource: technical design tool, project management tool, and communication tool.

Enterprise Bus Architecture

- Requirements are gathered and represented in a form of Enterprise Data Warehouse Bus Matrix
  - Each row corresponds to a business/process
  - Each column corresponds to a dimension of the business
    - Each column is a conformed dimension
- Enterprise Data Warehouse Bus Matrix documents the overall data architecture for DW/BI system

DW Bus Matrix

Data Warehouse BUS Matrix

	Date	Company	Customer	Product	Geo	Dist Ctr	Promo
Company Sales	X	X	X	X	X		X
Customer Discounts	X	X	X	X	X		X
Product Cost	X	X	X	X	X	X	X
Company Inventory	X	X		X			
Dist Ctr Inventory	X	X		X			

Enterprise Bus Architecture Matrix

Business Process/Event	Date	Policyholder	Coverage	Covered item	Agent	Policy	Claim	Claimant	Payee
Underwriting transactions	x	x	x	x	x	x			
Policy premium billing	x	x	x	x	x	x			
Agents commissions	x	x	x	x	x	x			
Claims transactions	x	x	x	x	x	x	x	x	x

Enterprise Bus Architecture Matrix

- Possible Problems:
  - Level of details for each column and row in the matrix
  - Row-related
    - Listing departments/imitating organizational chart instead of business processes
    - Listing reports and analytics related to business process instead of the business process itself
      - Ex. Shipping orders business process supports various analytics such as customer ranking, sales rep performance, product movement analyses

Enterprise Bus Architecture Matrix

- Possible Problems (Cont):
  - Column-related
    - Generalized columns/dimensions
      - Example: “Entity” column is too general as it includes employees, suppliers, contractors, vendors, customers
    - Too many columns related to the same dimension
      - Worst case when each attribute is listed separately
      - Example: Product, Product Group, LOB are all related to the Product dimension and should be listed as one.

Conformed Dimensions

- Conformed dimensions are:
  - identical, or
  - strict mathematical subsets of the most granular, detailed dimension.
- Conformed dimensions have consistent
  - Dimension keys
  - Attribute column names
  - Attribute definitions
  - Attribute values
- If two marts have dimensions (e.g., customer, product) that are not conformed, then they cannot be used together

Conformed Dimensions

- master or common reference dimensions
- Shared across the DW environment joining to multiple fact tables representing various business processes
- 2 types
  - Identical dimensions
  - One dimension being a subset of a more detailed dimension

Conformed Dimensions

- Identical dimensions
  - Same content, interpretation, and presentation regardless of the business process involved
  - Same keys, attribute names, attribute definitions, and domain values regardless of domain values they join to
  - Example: product dimension referenced by orders and the one referenced by inventory are identical
- One dimension being a perfect subset of a more detailed, granular dimension table
  - Same attribute names, definitions, and domain values
  - Example: sales is linked to a dimension table at the individual product level; sales forecast is linked at the brand level

Conformed Dimensions

<u>Sales Fact Table</u>
Date key FK
Product key FK
… other FKeys…
Sales quantity
Sales amount

<u>Product Dimension</u>
Product key PK
Product description
SKU number
Brand description
Sub class description
Class description
Department description
Color
size
Display type

<u>Sales Forecast Fact Table</u>
Month key FK
Brand key FK
… other FKeys…
Forecast quantity
Forecast amount

<u>Brand Dimension</u>
Brand key PK
Brand description
Sub class description
Class description
Department description
Display type

Conformed Dimensions

- Benefits
  - Consistency
    - Every fact table is filtered consistently and results are labeled consistently
  - Integration
    - Users can create queries that drill across fact tables representing different processes individually and then join result set on common dimension attributes
  - Reduced development time to market
    - Once created, conform dimensions are reused

Types of Dimension Conformity

- Mean same thing
  - Single shared table or physical copy
  - Consistent data content, data interpretation, user presentation
- Rolled-up level of granularity
  - Roll-up dimensions conform to the base-level atomic dimension if they are a strict subset of that atomic dimension. (see Fig. 3.9)
- Dimension subset at same level of granularity
  - At same level but one represents only a subset of rows (see Fig. 3.10)
- Combination of above

Centralized Dimension Authority

- The major responsibility of the centralized dimension authority is to:
  - establish,
  - maintain, and
  - publish the conformed dimensions to all client data marts.
- 90% of up-front data architecture effort
- Political challenge

Conformed Facts

- In general, facts table data is not duplicated explicitly in multiple data marts.
- If facts live in more than one location, then their definitions and equations must be the same and they must be called the same.
- If it is impossible to conform a fact exactly, then different names should be given to different interpretations. This will make it less likely that incompatible facts will not be used in a calculation.

Rapidly Changing Dimensions

From the previous slides: What is slow?

What if the changes are fast?

Must a different design technique be used?

- Small dimensions:
  – the same technologies as for slowly changing dimensions may be applied
- Large dimensions:
  – the choice of indexing techniques and data design approaches are important
  – find suppress duplicate entries in the dimension
  – do not create additional records to handle the slowly changing dimension problem

Rapidly changing very large dimensions

- Break off some of the attributes into their own separate dimension(s), a demographic dimension(s).
  – force the attributes selected to the demographic dimension to have relatively small number of discrete values
  – build up the demographic dimension with all possible discrete attributes combinations
  – construct a surrogate demographic key for this dimension

NB! The demographic attributes are the one of the heavily used attributes. Their values are often compared in order to identify interesting subsets.

More Rapidly Changing Dimensions

- Break off the rapidly changing attributes into one or more separate dimensions
- Two foreign keys in fact table:
  1. Primary dimension table
  2. Rapidly changing attribute(s)

Rapidly Changing Monster Dimensions

- Multi-million customer dimension present unique challenges that warrant special treatment:
  1. Browsing or constraining takes too long
  2. Type-II change not feasible
  3. Business users want to track the myriad of customer attribute changes, eg, insurance companies want accurate information of customers at the time of approval of a policy or when a claim is made

Mini-Dimensions

- Single technique to handle browsing-performance & change tracking problems
- Separate out frequently analyzed or frequently changing attributes into a separate dimension, called *mini-dimension*

Mini-Dimensions

Demographic Key	AGE	GENDER	INCOME LEVEL
1	20-24	M	< 20000
2	20-24	M	20K-24999
3	20-24	M	25K-29999
18	25-29	M	20K-24999
10	25-29	M	25K-29999

Mini-Dimensions

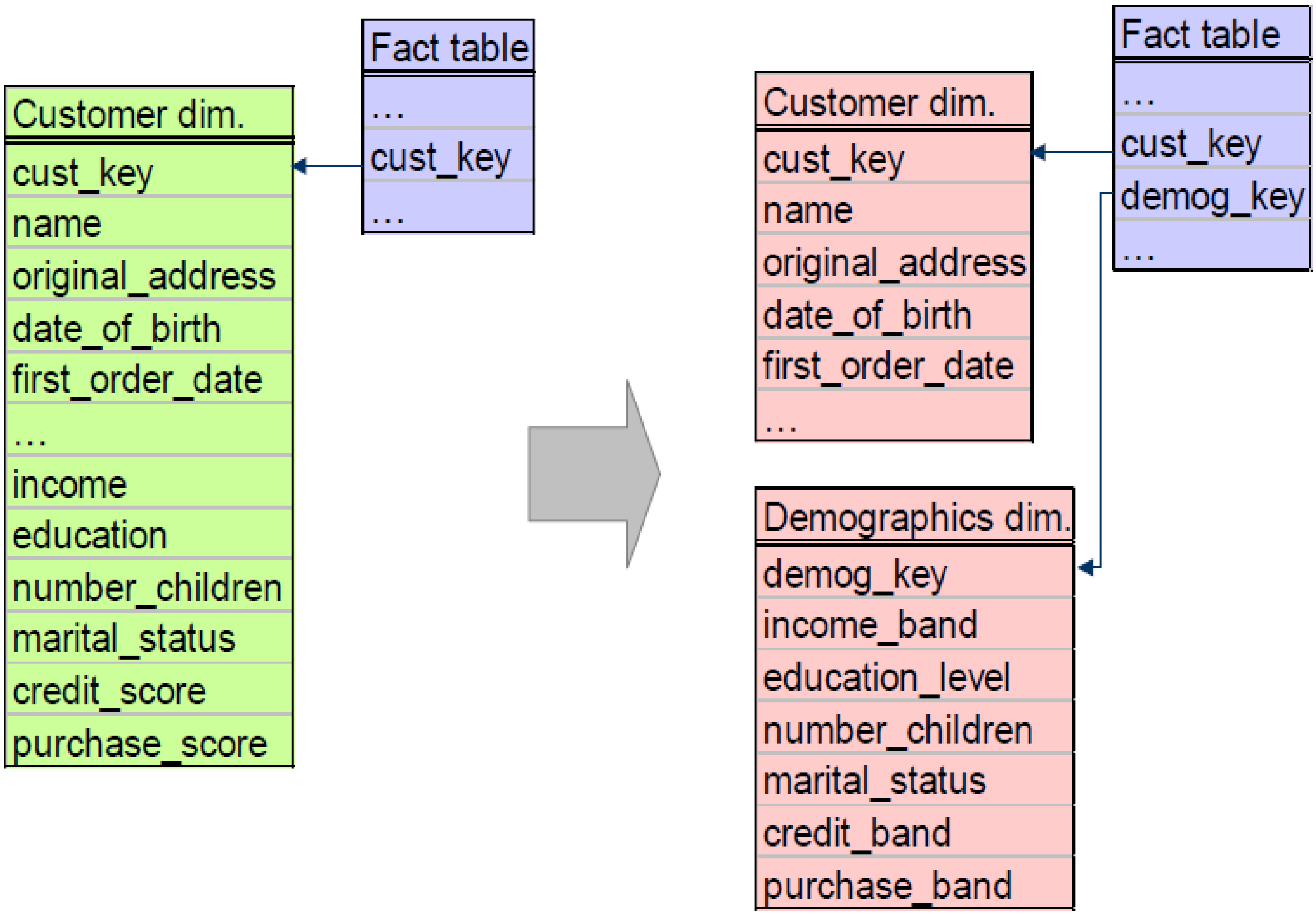
- Minidimension can not be itself allowed to grow very large
- 5 demographic attibutes
- Each attribute can take 10 distinct values
- How many rows in minidimension?

10,0000

Mini-Dimensions

- Separate out a package of demographic attributes into a demographic *mini-dimension*
- Age, gender, marital status, no. of children, income level, etc.
- One row in *mini-dimension* for each unique combination of these attibutes

Demographic Minidimension



Demographic Minidimension

Demographics dim.
demog_key
income_band
education_level
marrital_status

Three values

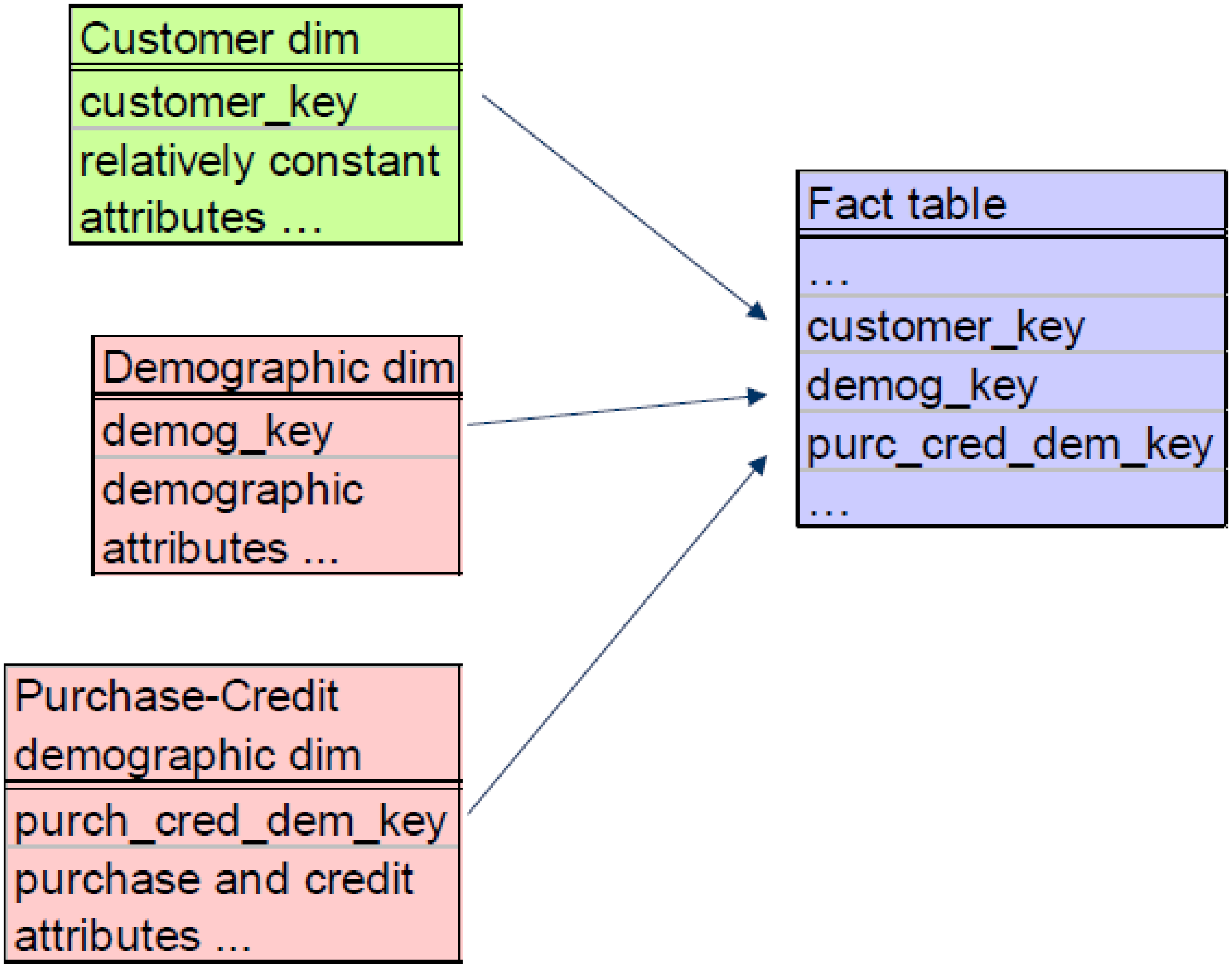
Two values

Two values

3*2*2=12 rows

D1	-100 000	Graduate	Married
D2	100 000-200 000	Graduate	Married
D3	200 000-	Graduate	Married
D4	-100 000	Non-graduate	Married
D5	100 000-200 000	Non-graduate	Married
D6	200 000-	Non-graduate	Married
	..cont	*..cont*	*..cont*

Two Demographic Minidimensions



Demographic Minidimension

- Advantages
  - frequent 'snapshoting' of customers profiles with no increase in data storage or data complexity
- Drawbacks
  - the demographic attributes are clumped into banded ranges of discrete values (it is impractical to change the set of value bands at a later time)
  - the demographic dimension itself can not be allowed to grow too large
  - slower down the browsing
- What if the fact table (connecting the demographic minidimension with the customer dimension) is sparse?

Demographic Minidimension

- What to do if the fact table (connecting the demographic minidimension with the customer dimension) is sparse?
  - Define a demographic transaction event, i.e., introduce a new fact table

  or

  - Add a current demographic key to the customer dimension table