SS G515 - Data Warehousing

Dimensional Modeling

Dr. Yashvardhan Sharma

Assistant Professor, CS & IS Dept.

BITS-Pilani

Data Warehouse: Design Steps

Step 1: Identify the Business Process

Step 2: Declare the *Grain*

Step 3: Identify the Dimensions

Step 4: Identify the Facts

Modeling Design Process

1. Identify the Business Process
   - Source of "measurements"
2. Identify the Grain
   - What does 1 row in the fact table represent or mean?
3. Identify the Dimensions
   - Descriptive context, true to the grain
4. Identify the Facts
   - Numeric additive measurements, true to the grain

Step 1 - Identify the Business Process

- This is a business activity typically tied to a source system.
- Not to be confused with a business department or function. An Orders dimensional model should support the activities of both Sales and Marketing.
- "If we establish departmentally bound dimensional models, we'll inevitably duplicate data with different labels and terminology."

Step 2 - Identify the Grain

- The level of detail associated with the fact table measurements.
- A critical step necessary before steps 3 and 4.
- Preferably it should be at the most atomic level possible.
- “How do you describe a single row in the fact table?”

Step 3 - Identify the Dimensions

- The list of all the discrete, text-like attributes that emanate from the fact table.
- They are the "by" words used to describe the requirements.
- Each dimension could be though of as an analytical "entry point" to the facts.
- "How do business people describe the data that results from the business process?"

Step 4 - Identify the Facts

- Must be true to the grain defined in step 2.
- Typical facts are numeric additive figures.
- Facts that belong to a different grain belong in a separate fact table.
- Facts are determined by answering the question, "What are we measuring?"
- Percentages and ratios, such as gross margin, are non-additive. The numerator and denominator should be stored in the fact table.

Grocery Store: The Universal Example

The Scenario:

- Chain of 100 Grocery Stores
- 60000 individual products in each store

- 10000 of these products sold on any given day(average)
- 3 year data

Grocery Store DW

- Step 1: Sales Business Process
- Step 2: Daily Grain
- A word about GRANULARITY
  - Temp sensor data: per ms, sec, min, hr?
  - Size of the DW is governed by granularity
  - Daily grain (club products sold on a day for each store) Aggregated data
  - Receipt line Grain (each line in the receipt is recorded – finest grain data)

Grocery Store: DW Size Estimate

- Daily Grain
- Size of Fact Table

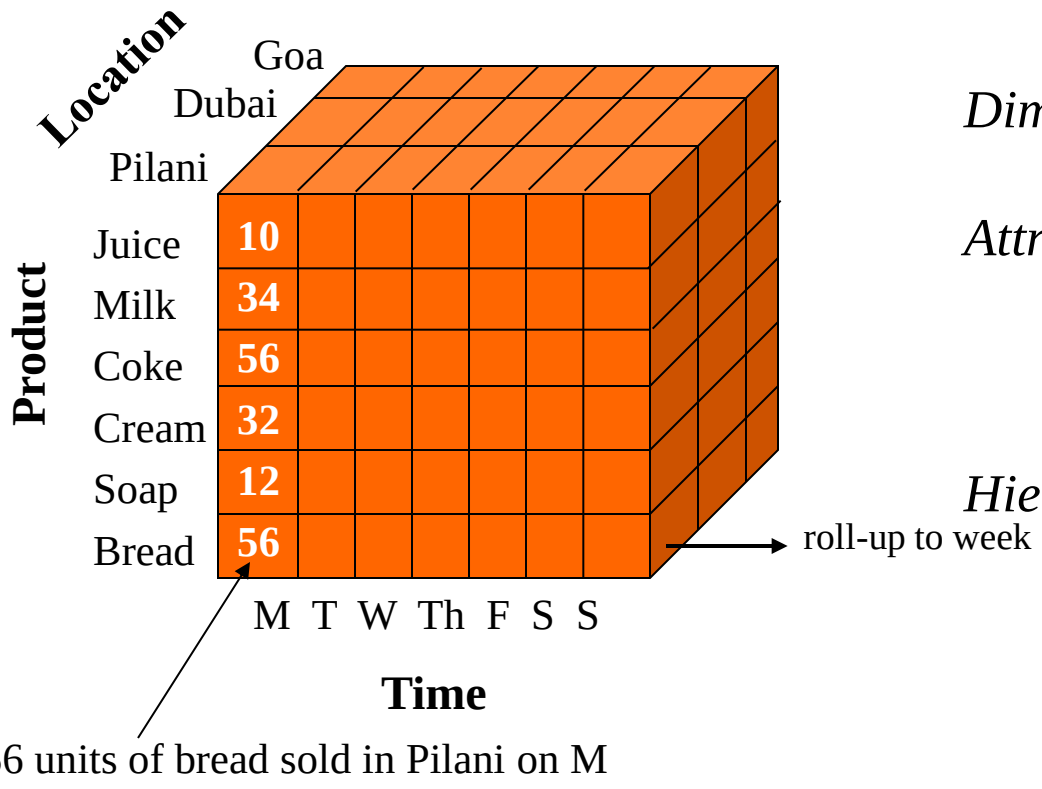
  = 100*10000*3*365

  = 1095 million records
- 4 facts & 4 dimensions (48 bytes)

Fact size: 8 bytes, dimension size: 4 bytes

- 1095 m * 48 bytes = 52560 m bytes
- i.e. ~ 50 GB

Data Cube



Dimensions:

Time, Product, Location

Attributes:

Product (upc, price, …)

Location…

…

Hierarchies:

Product → Brand → …

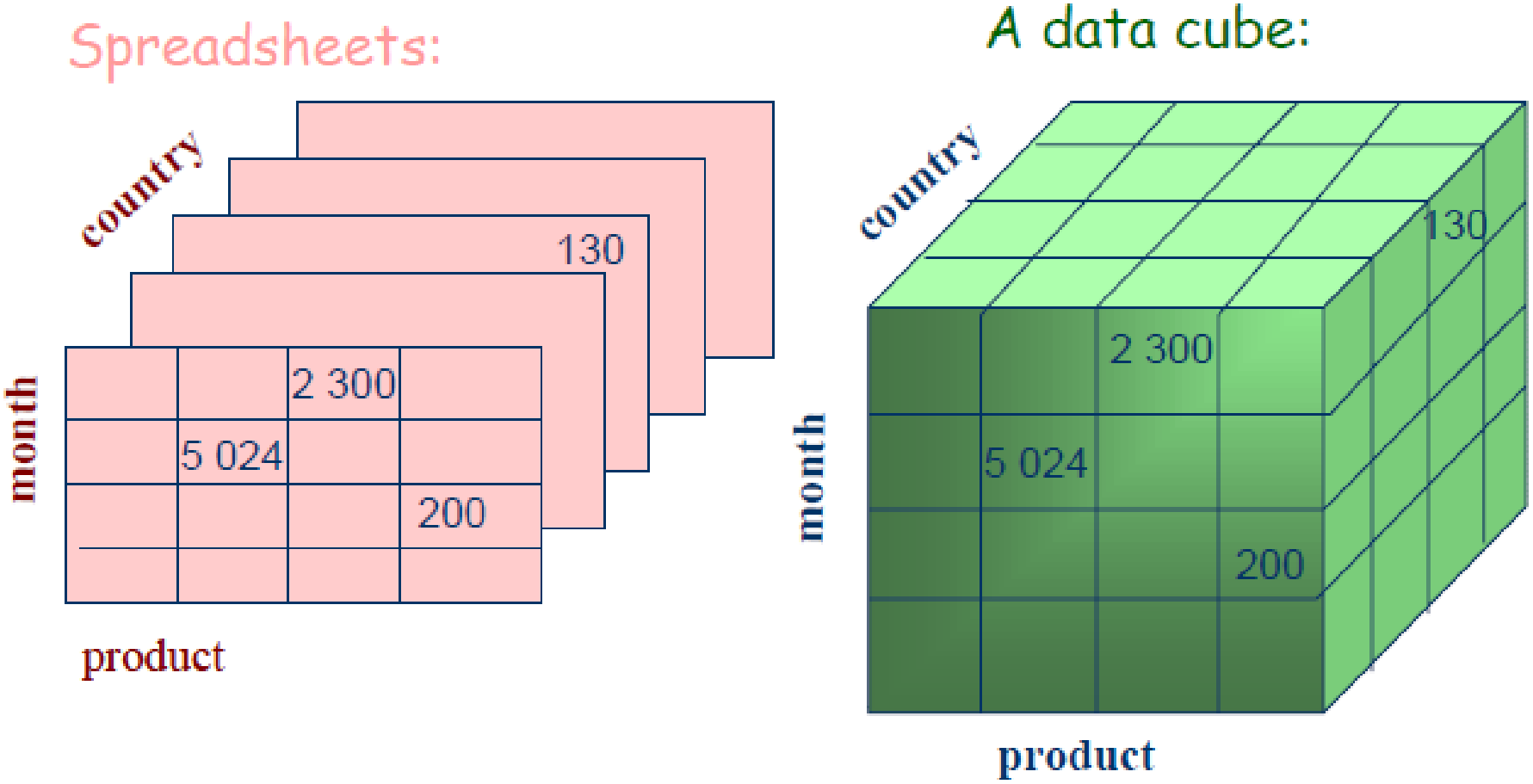
Day → Week → Quarter

City → Region → Country

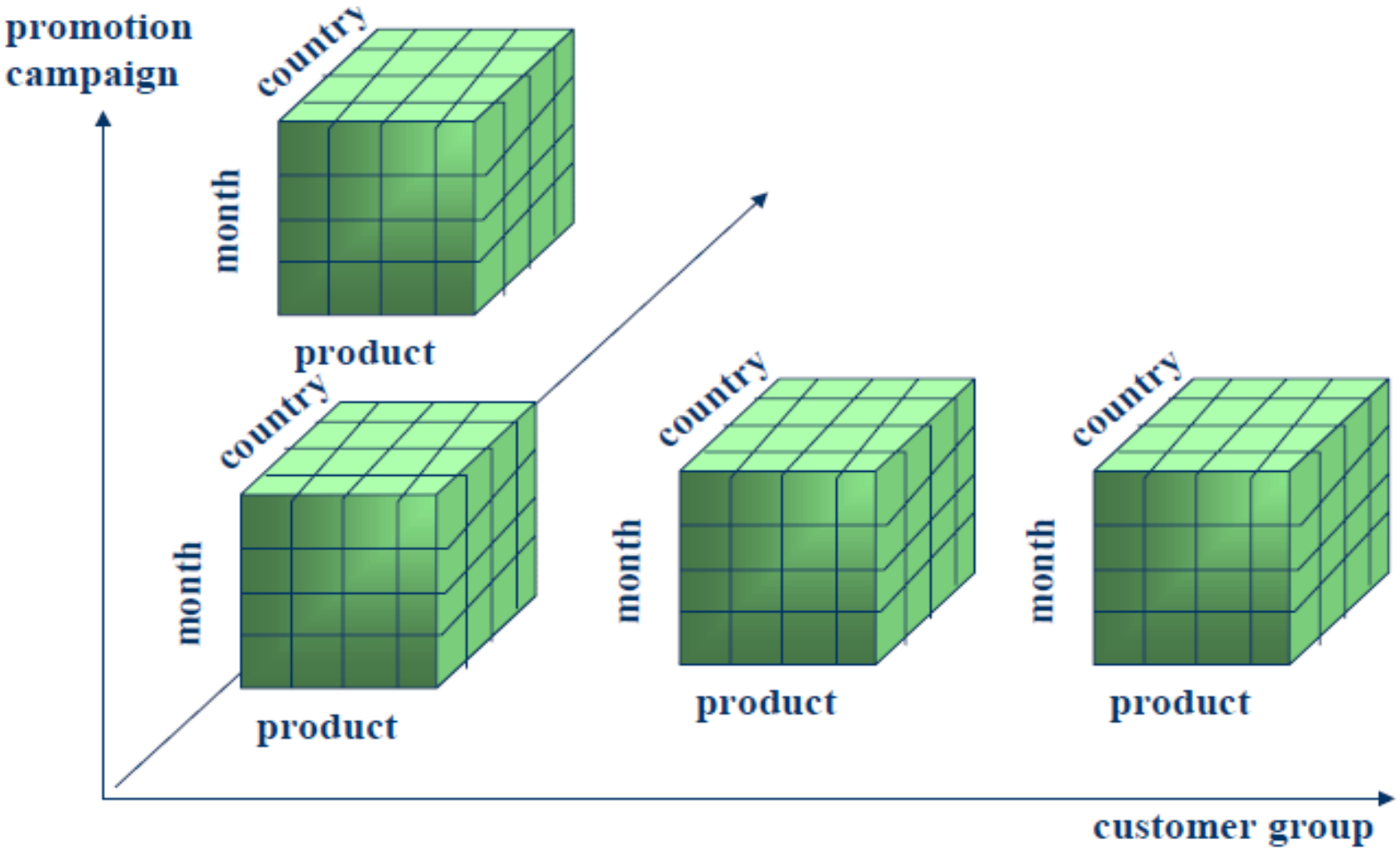
To Meet the Requirements within DW

- The data is organised differently, i.e. “multidimensional”
  - star-joins schemas
  - snowflake schemas
- The data is viewed differently
- The data is stored differently
  - vector (array) storage
- The data is indexed differently
  - bitmap indexes
  - join indexes

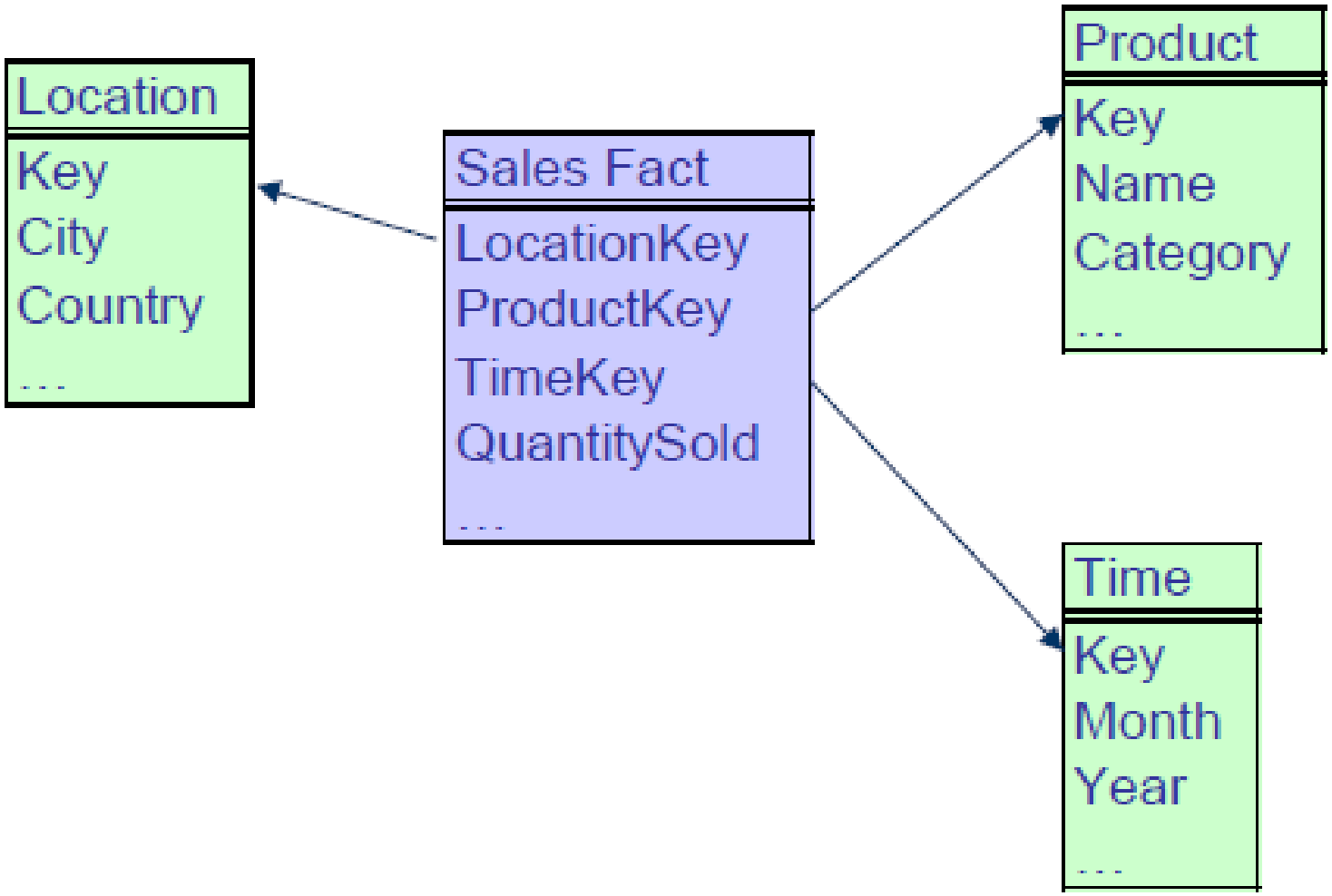
From Spreadsheets to Data Cubes



“Multidimensional” view of the data



Example - Star-Join Schema



Example

Location

Key	City	…
1	Stockholm	…
2	London	…
3	Paris	…

Sales

LKey	PKey	TKey	Qnt
1	1	1	5
1	2	1	7
1	3	1	4
2	1	1	8
2	2	1	3
2	3	1	5
3	1	1	20
3	2	1	10
3	3	1	30
1	1	2	10
1	2	2	9
1	3	2	7
2	1	2	5
2	2	2	10
2	3	2	8
3	1	2	20
3	2	2	50
3	3	2	30

Product

Key	Name	…
1	# 5	…
2	Noah	…
3	Opium	…

Time

Key	Month	…
1	Jan	…
2	Feb	…
3	Mar	…
4	Apr	…

Star-Join Schema

- A single fact table and a single table for each dimension
- Every fact points to one tuple in each of the dimensions and has additional attributes
- The fact table is highly normalised, whereas the dimension tables not normalised.
- Dimensions does not capture hierarchies directly
- Generated keys are used for performance and maintenance reasons

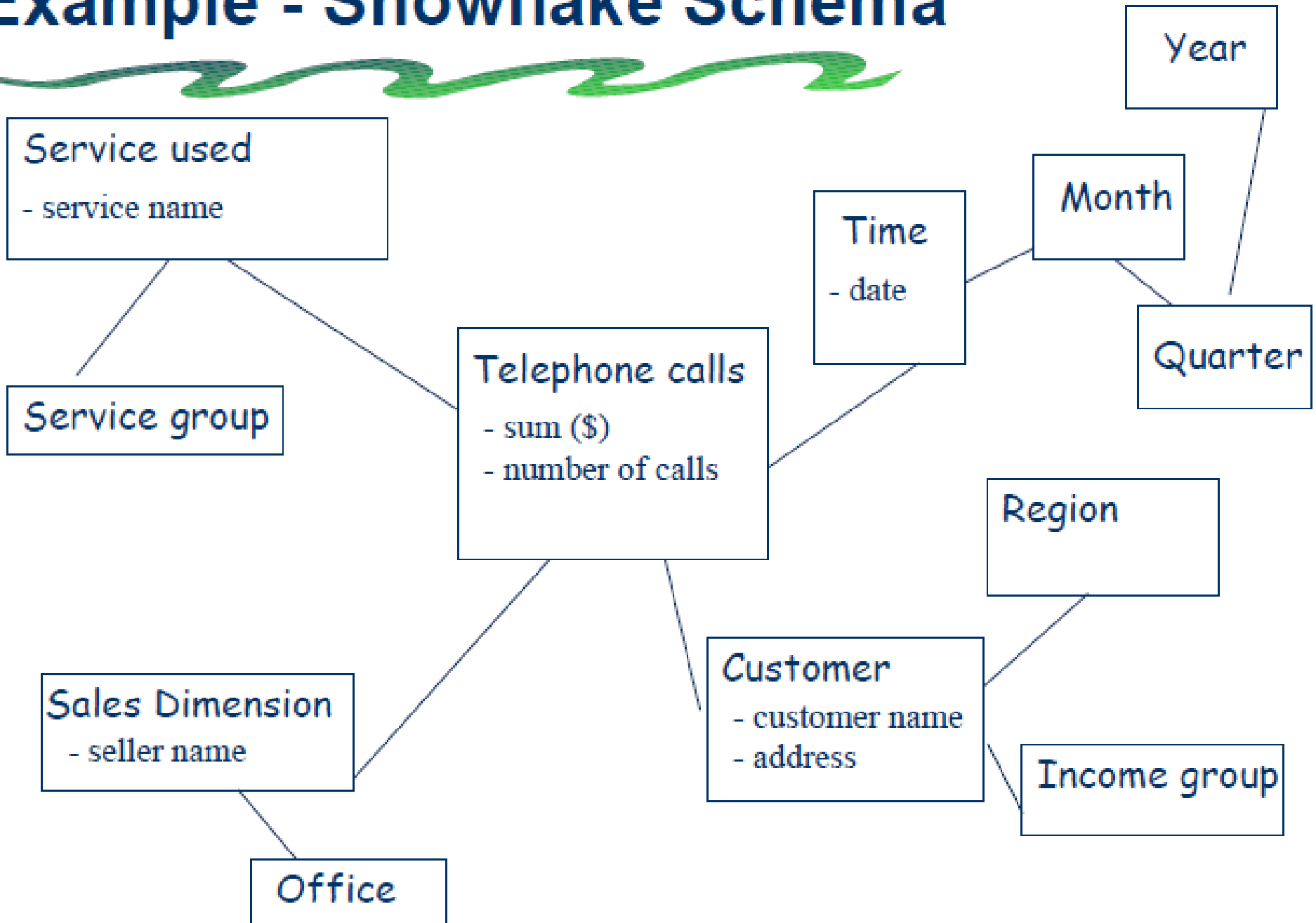
- Fact constellation: Multiple Fact tables that share

many dimension tables

Snowflake Schema

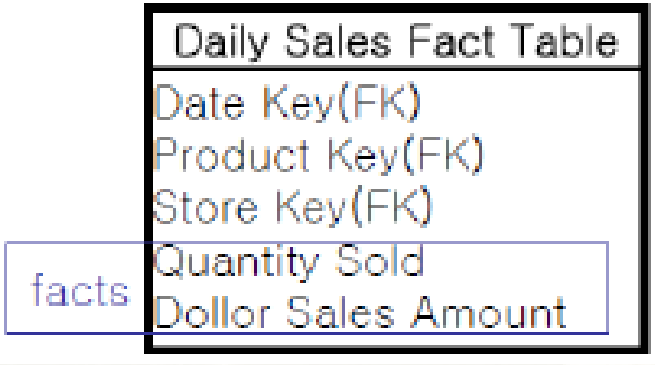
- Represent dimensional hierarchy directly by normalizing the dimension tables
- Save storage
- Reduces the effectiveness of browsing

Example - Snowflake Schema



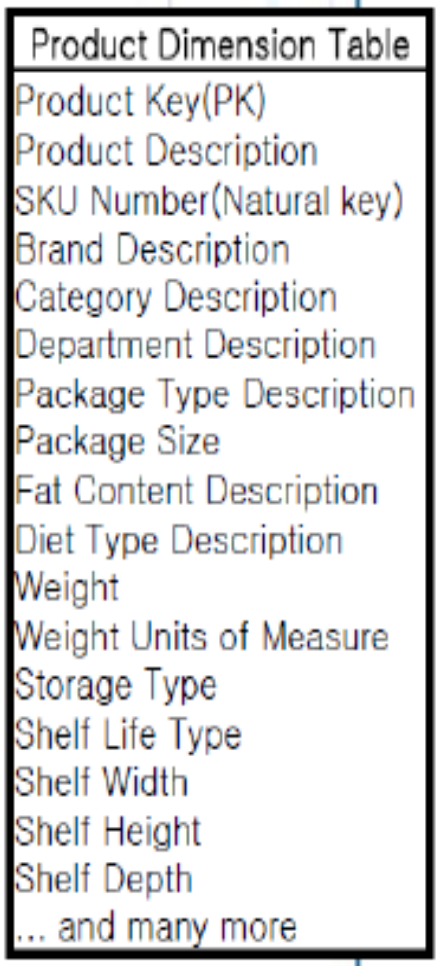
Dimensional modeling vocabulary

- Fact Table
  - Is the primary table in a dimensional model
  - Facts are numeric measurements (values) that represent a specific business aspect or activity
  - Facts can be computed or derived at run-time (metrics).
  - Have two or more foreign keys(FK) that connect to the dimension table's primary keys
    - Satisfy referential integrity
  - Generally has own primary key(called a composite or concatenated key) made up of a subset of the foreign keys
  - Express the many-to-many relationships between dimensions



Dimensional modeling vocabulary

- Dimension Tables
  - Are integral companions to a fact table
  - Contain the textual descriptors of the business
  - Have many columns or attributes
  - Defined single primary key(PK)
  - We strive to minimize the use of codes in our dimension tables by replacing them with more verbose textual attributes
    - Operational codes often have intelligence embedded in them
  - Typically are highly denormalized
  - Typically are geometrically smaller than fact tables, improving storage efficiency by normalizing or snowflaking
    - Snowflake
      - Brand description and category description replace by brand code and create brand table



Sample dimension Table

Dimensional modeling vocabulary

- Surrogate Keys
  - rather than operational production codes (:natural keys)
  - are also called as meaningless keys, integer keys, nonnatural keys, artificial keys, synthetic keys
  - are integers that are assigned sequentially as needed to populate a dimension
- Every join between dimension and fact tables should be based on meaningless integer surrogate keys
  - We want to avoid embedding intelligence in the data warehouse keys
    - because any assumptions that we make eventually may be invalidated
  - Queries and data access applications should not have any built-in dependency on the keys
    - because the logic also would be vulnerable to invalidation
- We want to discourage the use of concatenated or compound keys for dimension tables
  - to avoid multiple parallel joins between the dimension and fact tables

Dimensional modeling vocabulary

- Surrogate Keys Benefits
  - The surrogate keys buffer the data warehouse environment from operational changes
    - Surrogate keys allow the data warehouse team to integrate data from multiple operational source systems
  - The surrogate key is as small an integer as possible while eusuring that it will accommodate maximum number of rows in the dimension
    - Typically, a 4-byte integer is sufficient to handle most dimension situations
  - The surrogate keys are used to record dimension conditions that may not have an operational code
    - "Date to be Determined" or "Date Not Applicable"
  - Treating the surrogate date key as a date sequence number will allow the fact table to be physically partitioned on the basis of the date key
    - The partitioning is highly effective because it allows old data and new data to be loaded and indexed without disturbing the rest of the fact table

Dimensional modeling vocabulary

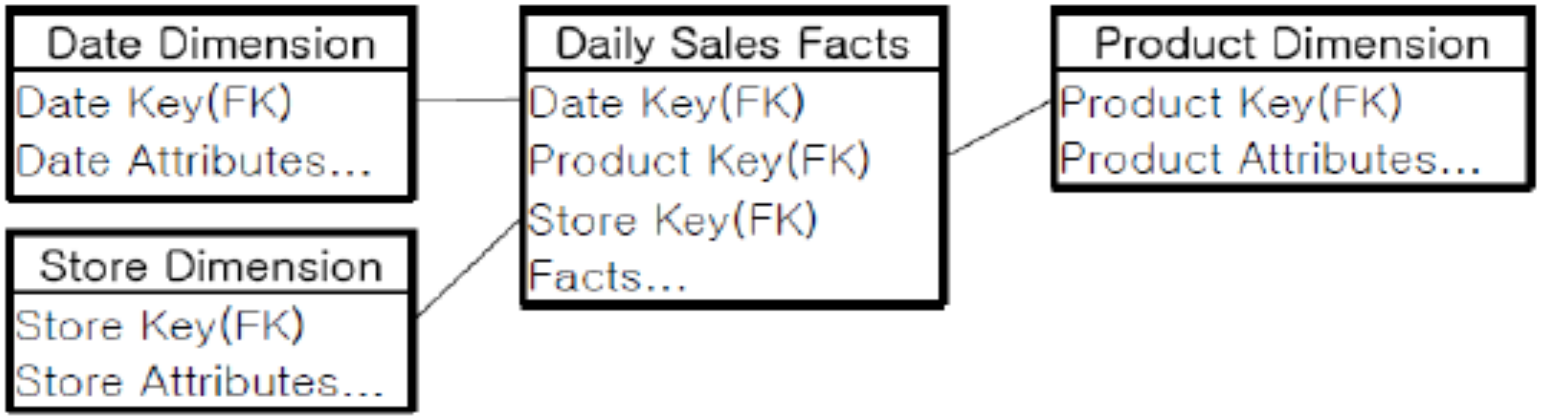
Dimension Table Attributes

- serve as the primary source of query constraints, groupings, and report labels
  - In a query or report request, attributes are identified as the "by" words
  - Ex) dollar sales by week by brand
- Key to making the DW usable and understandable
- The best attributes are textual and discrete
  - Consist of real words

Dimensional modeling vocabulary

Bringing Together Facts and Dimensions

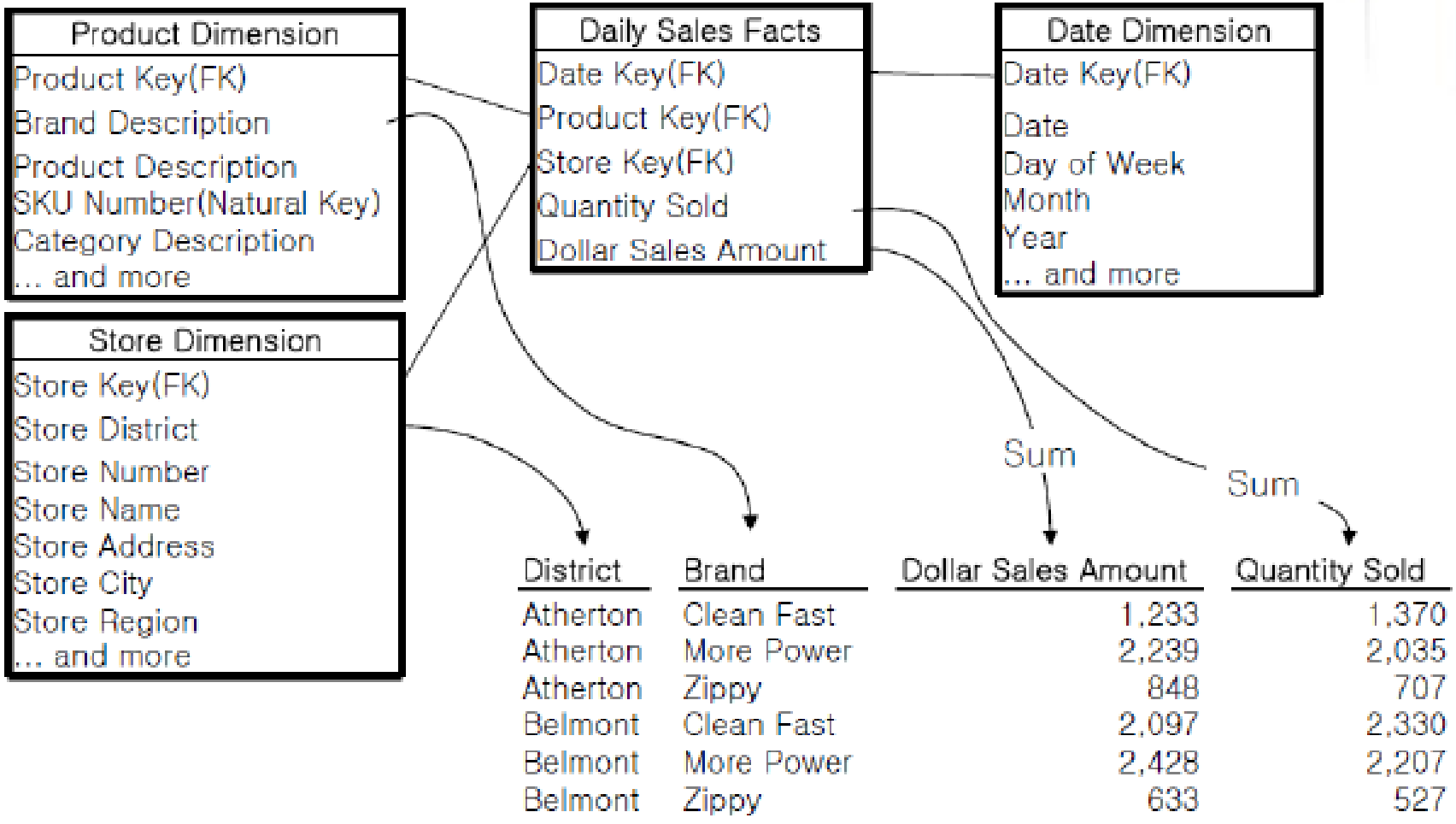
- The fact table consisting of numeric measurements is joined to a set of dimension tables filled with descriptive attributes
- This characteristic star-like structure is often called a star join schema
- All dimension are symmetrically equal entry points into the fact table
  - No preferences for any query
- We Certainly don't want to adjust our schemas if business users come up with new ways to analyze the business



Fact and dimension Tables in a dimensional model

Dimensional modeling vocabulary

Bringing Together Facts and Dimensions



Dragging and dropping dimensional attributes and facts into simple report

Dimensional modeling vocabulary

Fact vs Dimension Attribute

- Fact
  - The field is a measurement that takes on lots of values and participates in calculation
  - Ex) standard cost for a product is fact
    - seems like a constant attribute of the product but may be changed so often that eventually
- Dimension attribute
  - The field is a discretely valued description that is more or less constant and participates in constraints
- Occasionally, we can't be certain of the classification

  -> it may be possible to model the data field either way, as a matter of designer's prerogative

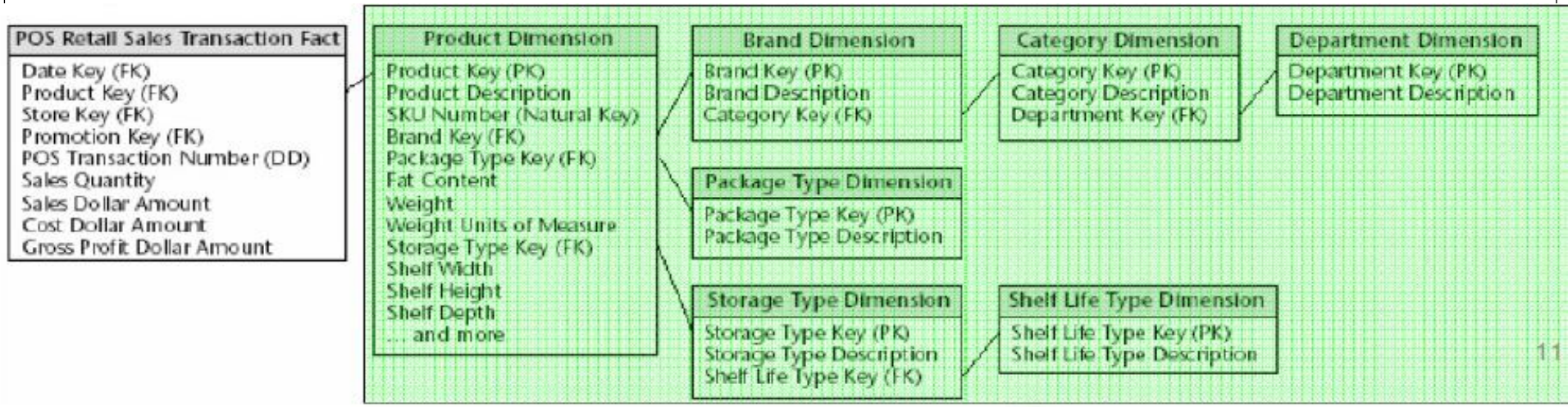
Resisting Comfort Zone Urges

Breaking some traditional modeling rules

- focused on delivering business value through ease of use and performance, not on transaction processing efficiencies

Dimension Normalization (Snowflaking)

- Redundant attributes are removed from the flat, denormalized dimension table and placed in normalized secondary dimension tables
- While the fact tables in both figures are identical, the plethora of dimension tables is overwhelming
- Fig 2.12 Partially snowflaked product dimension



Resisting Comfort Zone Urges

Dimension Normalization (Snowflaking)

- While snowflaking is a legal extension of the dimensional model
- We encourage you to resist the urge to snowflake with ease of use and performance
  - The multitude of snowflaked tables makes for a much more complex presentation
  - Numerous tables and joins usually translate into slower query performance
  - The minor disk space savings associated with snowflaked dimension tables are insignificant
    - Disk space savings grained by normalizing the dimension tables typically are less than 1 percent of the total disk space needed for the overall schema
  - Snowflaking slows down the users' ability to browse within a dimension
  - Snowflaking defeats the use of bitmap indexes

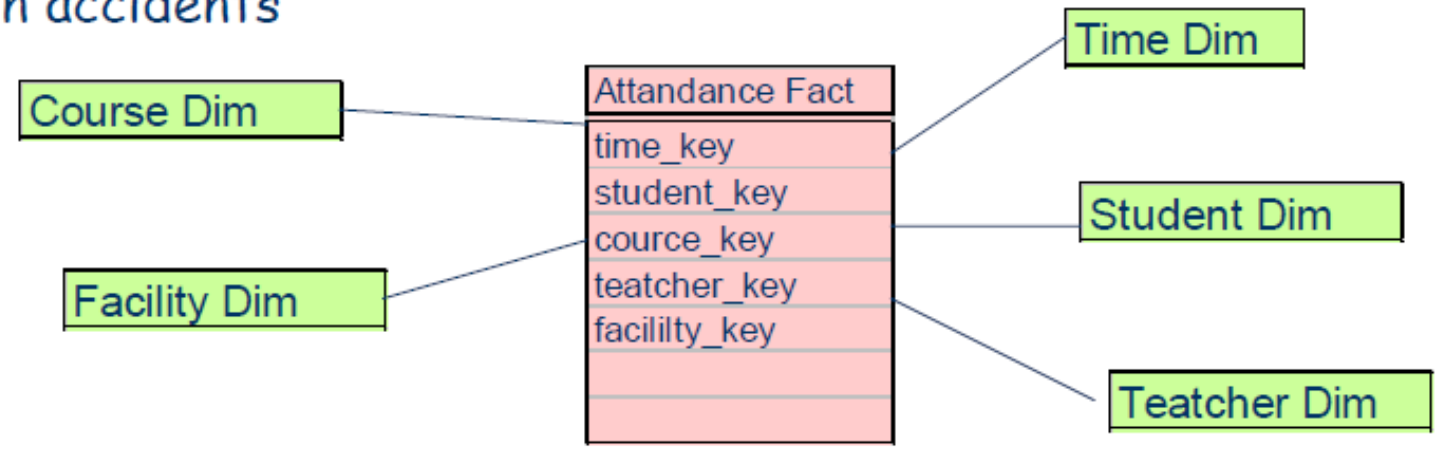
Factless fact tables

- Some fact tables quite simply have no measured facts!
- Are useful to describe events and coverage, i.e. the tables contain information that something has/has not happened.
- Often used to represent many-to-many relationships
- The only thing they contain is a concatenated key, they do still however represent a focal event which is identified by the combination of conditions referenced in the dimension tables
- There are two main types of factless fact tables:
  - event tracking tables
  - coverage tables

Factless fact tables

Event tracking tables

- records events, e.g. records every time a student attends a course, or people involved in accidents and vehicles involved in accidents



Coverage tables

- description of something that did not happend, e.g. which product did not sell during a promotion campaign.