Part III

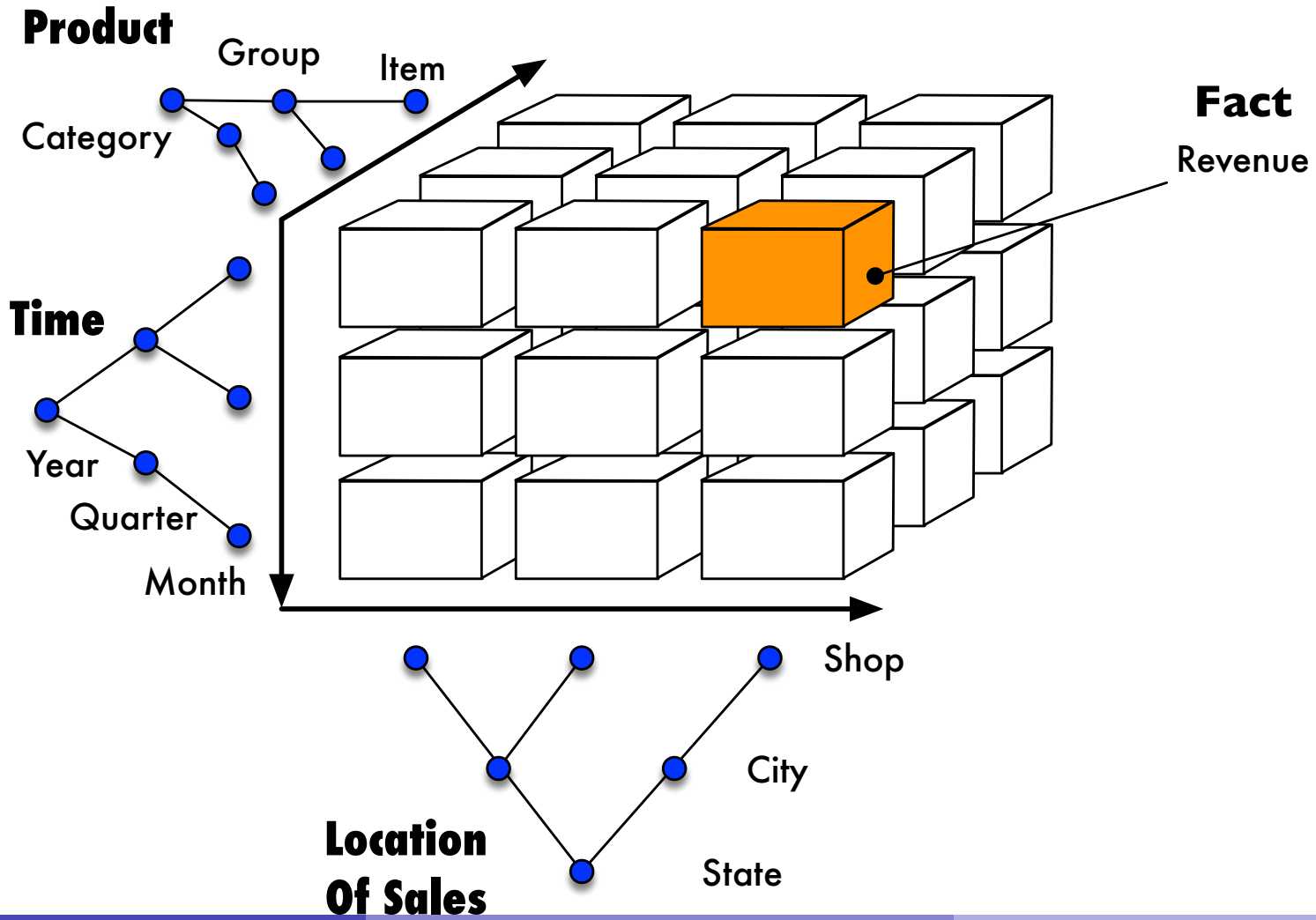
Multidimensional Data Model

Multidimensional Data Model

Basic concepts

Basic concepts

- Dimensions
- Facts / Metrics



Motivation

- Analysis-oriented data model
- Data analysis in the decision-making process
  - Key performance indicators are the focus
    → Facts
    • Profit, • revenue, • costs, • etc.
  - Consideration of indicators from different perspectives
    → Dimensions
    • Time, • space, • subject matter
  - Possible separation of dimensions to analyze
    → Hierarchies or consolidation levels
    • Year, • quarter, • month

Available information

- Qualifying
  - Represented by "'Category Attributes"'
  - Data for use as a navigation grid ("'Drill-Paths"')
  - Modeled as concept hierarchies in the context of dimensions
- Quantifying
  - Forming the subject of the evaluation ("'Sum Attributes"' or other arithmetic operations)
  - Cells of a cube, with dimensions as edges

Dimensions

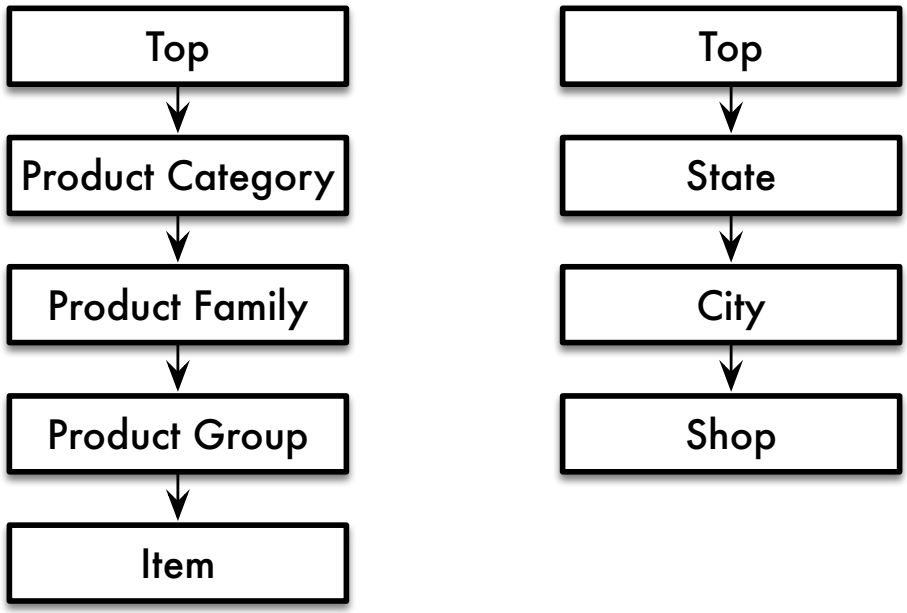
- Dimension:
  - Describes a possible view of associated metrics
  - Finite set of n n ($n \geq 2$) dimension elements (hierarchy objects) that have a semantic relation
  - Acts as an orthogonal structure of the data space
- Examples:
  - Product,
  - Branch structure,
  - Business year

Hierarchies in dimensions

- Dimension Elements:
  - Nodes of a classification hierarchy
  - Classification level describes the degree of agglomeration
  - Representation of dimensions over classification scheme (scheme of classification hierarchies)
- Forms:
  - Simple Hierarchies
  - Parallel Hierarchies

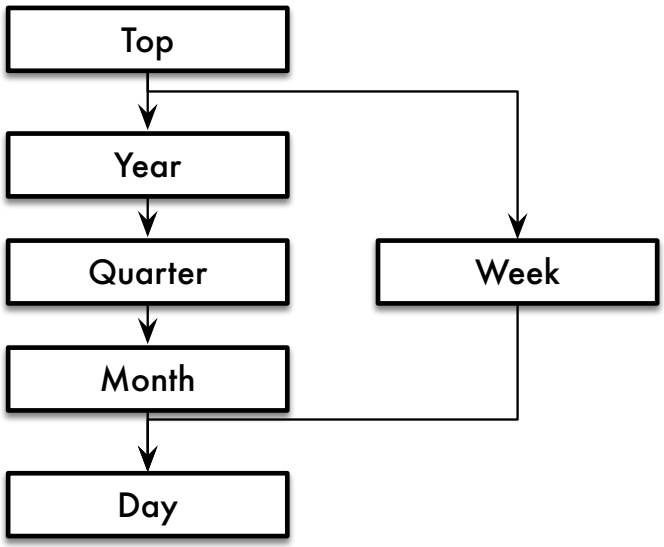
Simple Hierarchy

- Higher level of the hierarchy contains the aggregated levels of the directly next lower level
- Highest Node: *Top*
  - Provides compression to a single value for the dimension



Parallel Hierarchy

- Within a dimension, there are several independent types of grouping possible.
- No hierarchical relationship between parallel branches
- Parallel hierarchy
  - Path in the classification schema
  - Consolidation path



Schema of a Dimension D

- Partially ordered set of category attributes $(\{D^1, \ldots, D^n, Top_D\}; \rightarrow)$
  - Generic maximal element Top_D
  - Funktional dependence $\rightarrow$
- Top_D s is functionally determined by all the attributes:

$$\forall i, 1 \leq i \leq n : D_i \rightarrow Top_D$$

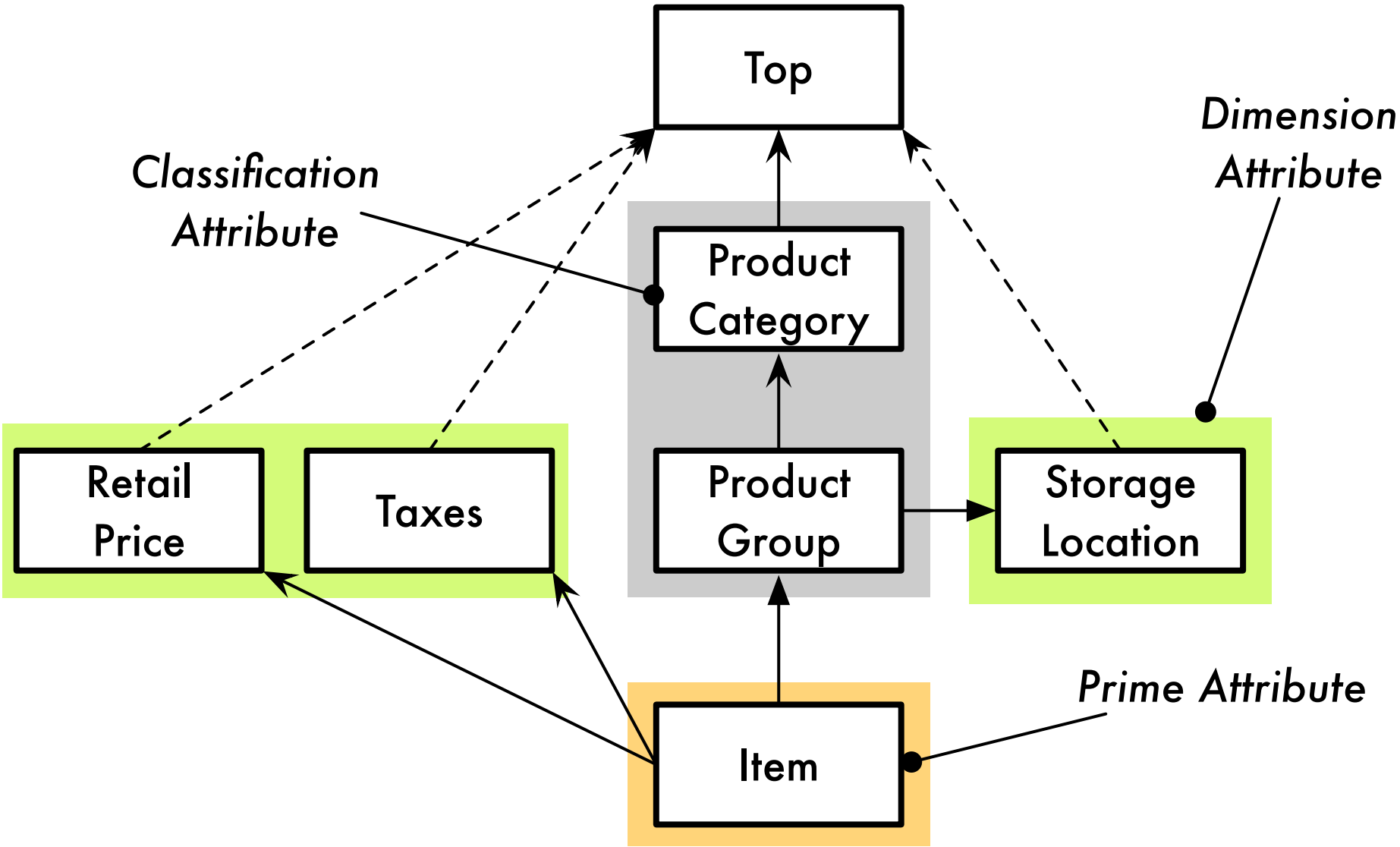
- There is exactly one D_i, that determines all other category attributes
  - Sets finest granularity of a dimension

$$\exists i, 1 \leq i \leq n, \forall j, 1 < j \leq n, i \neq j : D_i \rightarrow D_j$$

Categorical attributes

- Content-related refinement by different roles:
- Primary attribute
  - Categorical attribute, that determines all other attributes of a dimension
  - Defines maximum granularity level
  - Example: "'order item"'
- Classification attribute
  - Element of the set forming a multilevel categorization (classification hierarchy)
  - Example: "'product"', "'product group"', "'product category"'
- Dimensional attribute
  - Element of the set of attributes that are determined by a primary attribute or classification attribute and only determine Top_D
  - Beispiel: "'shelf item"'

Structure of a Dimension: Example



Metrics

- Metrics / facts:
  - (Aggregated) numerical metrics
  - Describe economic situations / circumstances
- Fact: measure
- Key Performance Indicator:
  - Made from facts(derived metric)
  - By applying arithmetic operations
- Examples:
  - Revenue, Profit, Costs
  - Contribution margin, ROI (Return on Investment)
  - Turnover rate, Increase in Revenue

Fact: Schema

- Schema is specified by multiple components.
- Granularity $G = \{G_1, \ldots, G_k\}$
  - G is a subset of all category attributes for all existing dimension schemas $DS_1, \ldots, DS_n$
    - ★ $\forall i, 1 \leq i \leq k, \exists j, 1 \leq j \leq n : G_i \in DS_j$
    - ★ $\forall i, 1 \leq i \leq k, \forall j, 1 \leq j \leq k, i \neq j : G_i \not\rightarrow G_j$
      (No functional dependence between category attributes of a granularity)
  - "'Level of Detail"' of the facts
- Summation type $SumTyp$
  - Stock
  - Flow
  - Value per Unit

Metrics

- Metric M is defined by
  - Granularity G
  - Calculation rule $f()$ over facts
  - Summation type $SumTyp$
- Calculation over non-empty subset of the existing facts in the schema
- $M = (G, f(F_1, \ldots, F_k), SumTyp)$

Metric: Forming f()

- Scalar functions
  - $+, -, *, /, mod$
  - Example: $VATrate = quantity * price * taxrate$
- Aggregate functions
  - Function $H()$ for condensing a dataset by n individual values to an aggregated value

$$H : 2^{dim(X_1) \times \cdots \times dim(X_n)} \rightarrow dim(Y)$$

  - Bsp.: $SUM(), AVG(), MIN(), MAX(), COUNT()$
- Order-based funktions
  - Definition of metrics based on pre-defined orders
  - Bsp.: Accumulation, $TOP(n)$, $MEDIAN()$

Summation types

- Assigning a summation type characterizes allowed aggregation operations
- $FLOW$
  - Period-based (per time unit)
  - Arbitrary aggregations allowed
  - Example: order quantity of an item per day
- $STOCK$
  - Measure for a period
  - Arbitrary aggregations allowed except for the temporal dimension
  - Example: Inventory, population
- $VALUE-PER-UNIT(VPU)$
  - Based on a point in time
  - Current states that are not cumulative
  - Only permitted: $MIN(), MAX(), AVG()$
  - Examples: price, exchange rate, tax rate

Summability

	FLOW	*STOCK*		*VPU*
		Aggregation over temporal dimension?		
		no	yes	
MIN/MAX	+	+		+
SUM	+	+	−	−
AVG	+	+		+
COUNT	+	+		+

Disjointness

- Concrete value of a measure contributes exactly once to a result
- Example: sales figures of a store

Revenue	**2019**	**2020**
Beer	38	42
Beer mix	27	31
Softdrinks	54	57
Total	92	99

- What is the total revenue per year?
- What is the revenue w.r.t. the product groups per year?

Completeness

- Metrics on a higher aggregation level can be entirely calculated from values of lower levels

Wine region	**2019**	**2020**
Rheinhessen	152	153
Saale-Unstrut	98	104
Mosel	161	172
Others	20	22
Total	431	451

- Others = Some producers from other areas

Aggregation functions

X is a classification node and $(X_1, X_2, \ldots, X_n)$ is the partitioning

- Distributive aggregation function:
  $\exists g : f(X) = f\left(g(X_1), g(X_2), \ldots, g(X_n)\right)$
- Algebraic aggregatsfunktion: f ist calculable from a fixed set G
- Holistic aggregattion function: f can only be calculated from the basic elements of X

Aggregation type	**Example**
Distributive	$SUM()$, $COUNT()$, $MAX()$, $MIN()$
Algebraic	$AVG()$ mit $g_1 := SUM()$ und $g_2 := COUNT()$, $STDDEV()$
Holistic	$MEDIAN()$, $RANK()$, $PERCENTILE()$

The Cube

The Cube

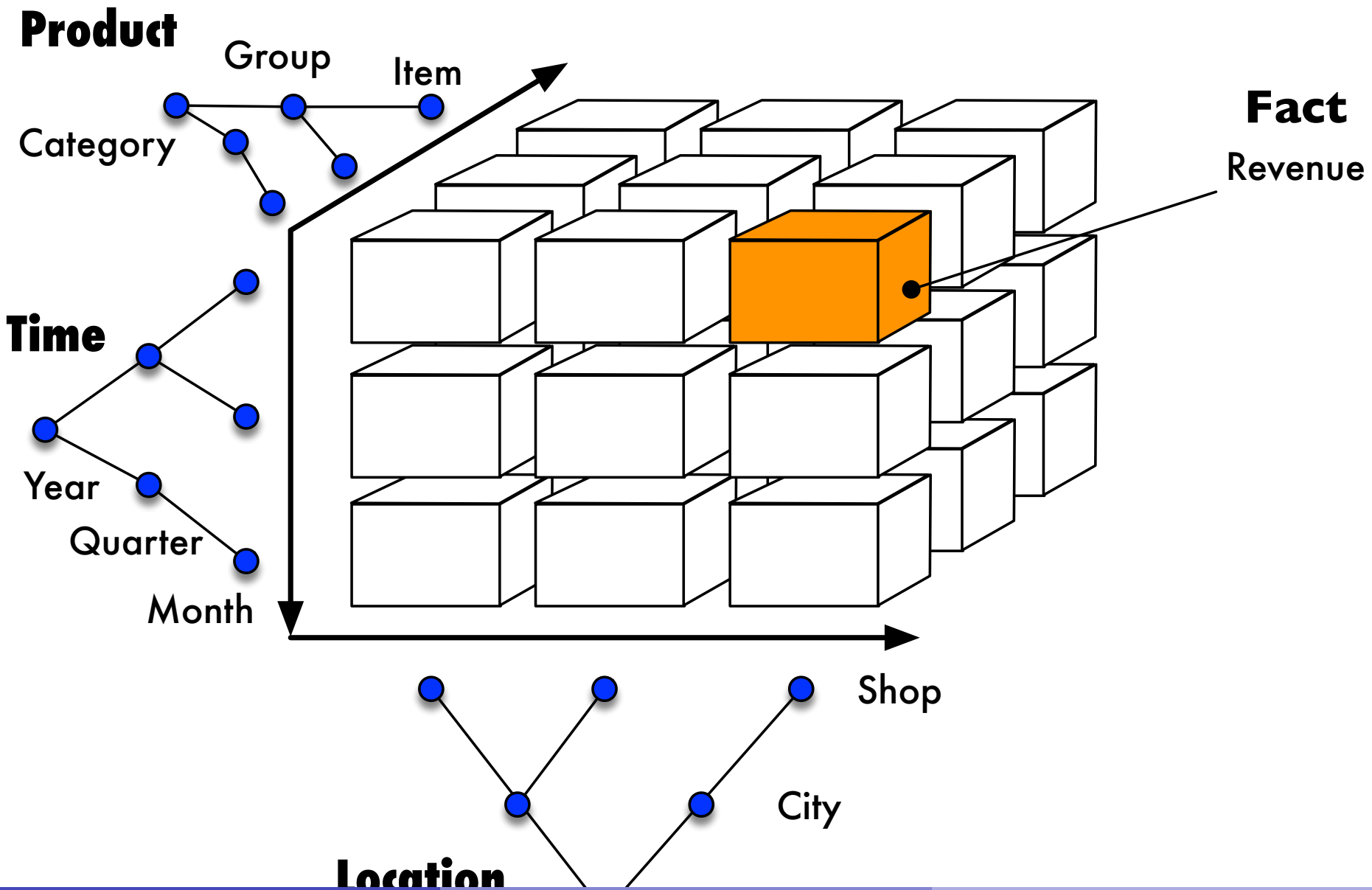
- Cube (actually cuboid): The basis of multidimensional analysis
- Edges $\rightarrow$ Dimensions
- Cells $\rightarrow$ one or more metrics (aas a function of dimensions)
- Number of dimensions $\rightarrow$ Dimensionality
- Visualizaion
  - 2 Dimensions: Table
  - 3 Dimensions: Dice
  - > 3 Dimensions: Multidimensional domain structure
- Schema C of a Cube
  - Set of dimensions (-schemas) DS
  - Set of metrics M
- $C = (DS, M) = (\{D^1, \ldots, D^n\}, \{M^1, \ldots, M^m\})$

Orthogonality

- In multidimensional schematas,orthogonality applies, i.e.
  - No functional dependencies between attributes of different dimensions

$$\forall i, 1 \leq i \leq n, \forall j, 1 \leq j \leq n, i \neq j \neg \exists k, l : D^i.D_k \rightarrow D^j.D_l$$

Multidimensional Data Cube



Conceptual Modelling

Conceptual Modelling

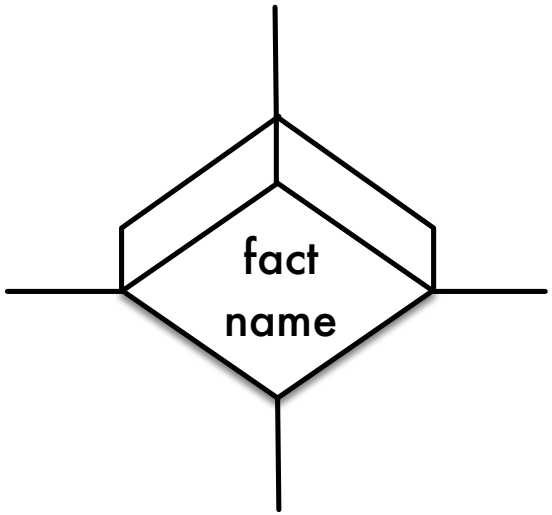
Formal description of the problem and the required information structures for the use case.

- Problems of conventional design techniques (ER, UML):
  - Inadequate semantics for multidimensional data model
  - Here: renunciation of universal applicability, instead focus on analysis
  - Example: classification level, Fact $\rightarrow$ Entity?

ME/R-Modell

- Multidimensional Entity/Relationship Model
  [Sapia et. al. (1998)]
- Extension of the classical ER model
  - Entity set "'Dimension Level"' (classification level)
    - ★ No explicit modeling of dimensions
  - n-ary relationship set "'Fact"'
    - ★ Metrics as attributes of the relationship
  - Binary relationship set "'Classification"' and "'Roll-Up"' (Union of classification levels)
    - ★ Defines directed, non-cyclic graph

ME/R: Notations



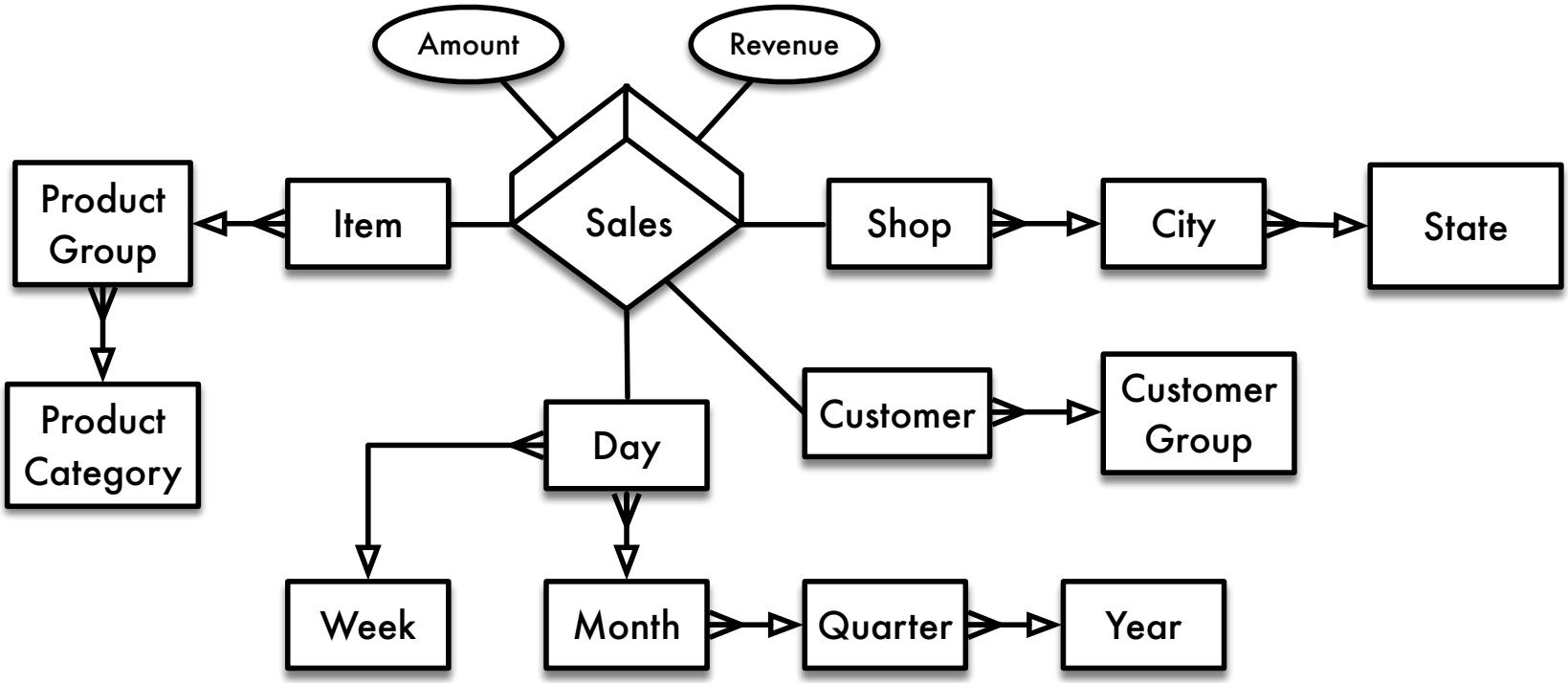
Fact relationship set

dimension level set

rolls-up
relationship set

attribute

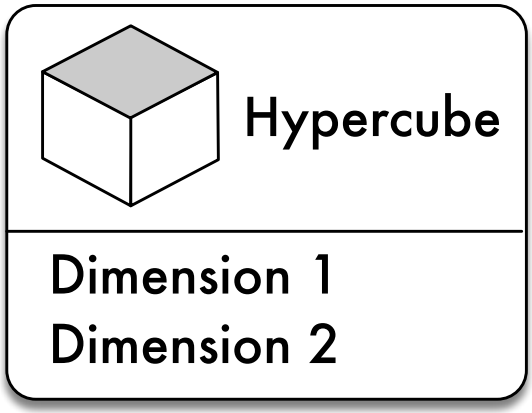
ME/R: Example


Amount
Revenue
Product Group
Item
Sales
Shop
City
State
Product Category
Day
Customer
Customer Group
Week
Month
Quarter
Year

ADAPT

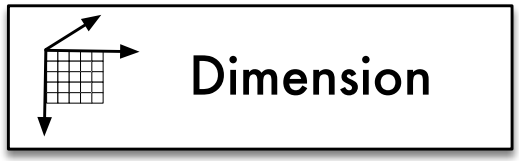
- Application Design for Analytical Processing Technologies
- Bulos 1998 & Bulos 2006
- Considering DDL and DML aspects
- Fokus on dimensions
- Rules for metrics possible
- Mixing of metadata and value manifestations

ADAPT

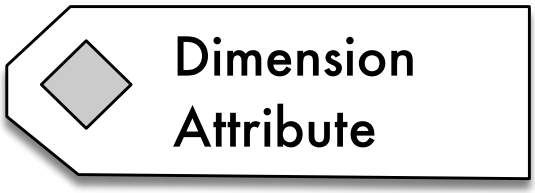

Hypercube
Dimension 1
Dimension 2

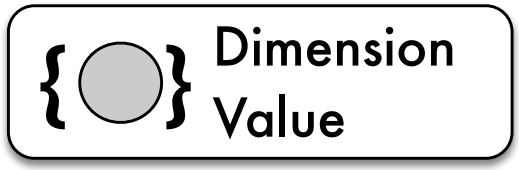

f( )
Formula

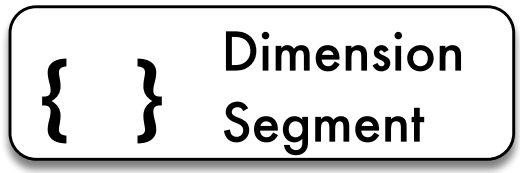

Hierarchy


Dimension

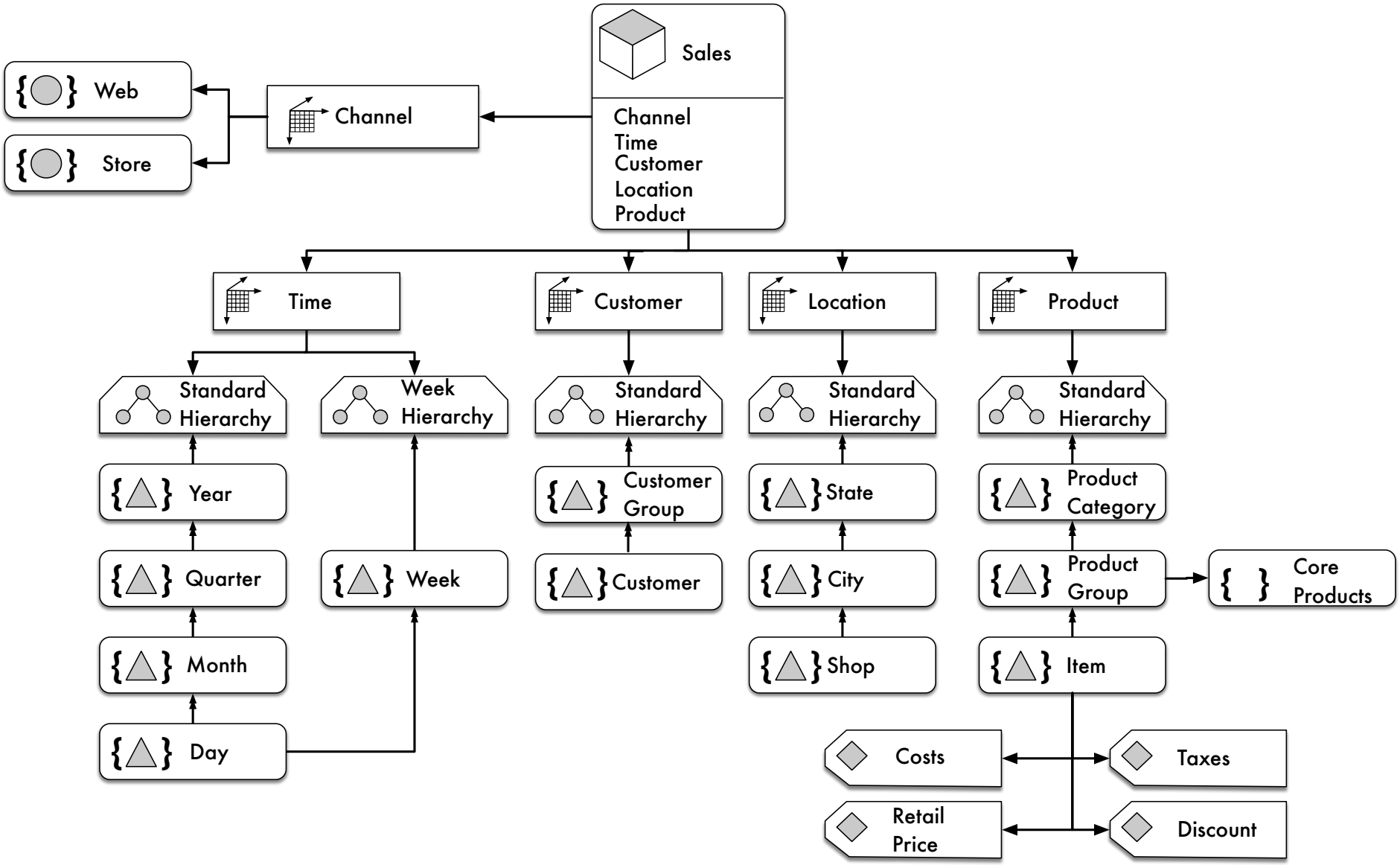

Hierarchy
Level


Dimension
Attribute


Dimension
Value


Dimension
Segment

ADAPT: Example



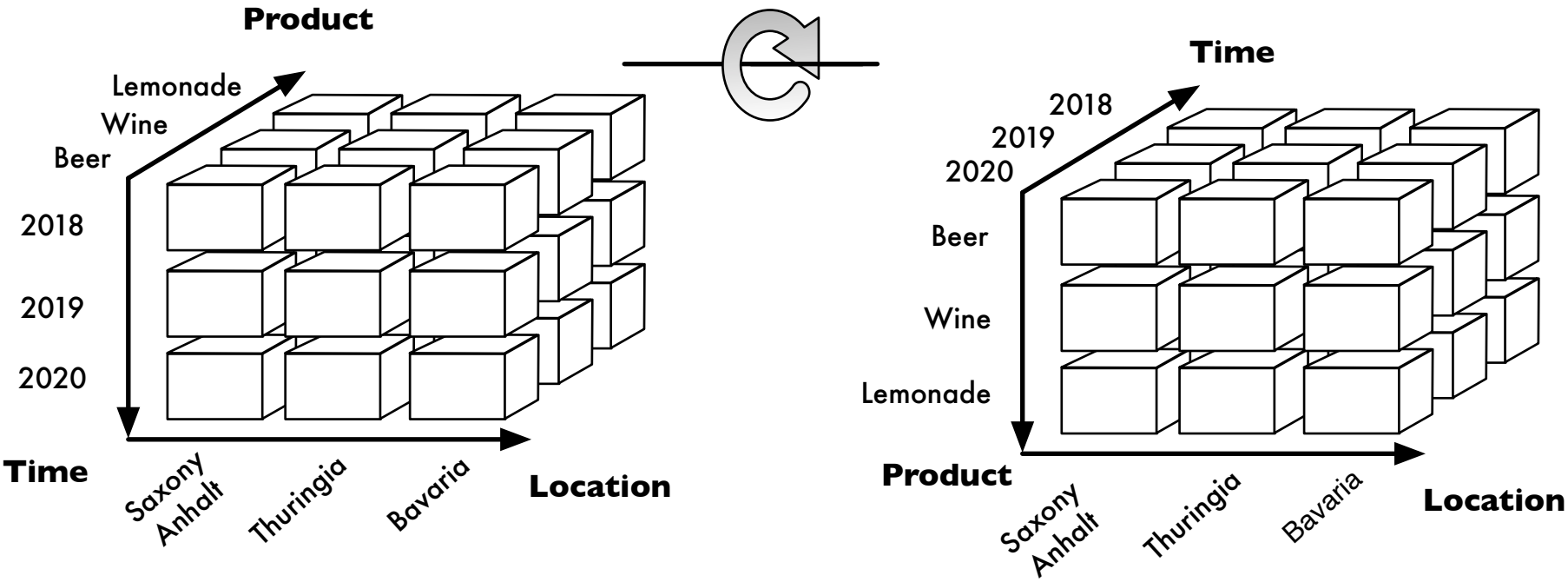
Operations for data analysis

Operations for data analysis

- OLAP operations on multidimensional data structures
- Standard operations
  - Pivoting
  - Roll-Up, Drill-Down
  - Drill-Across
  - Slice und Dice

Pivoting / Rotation

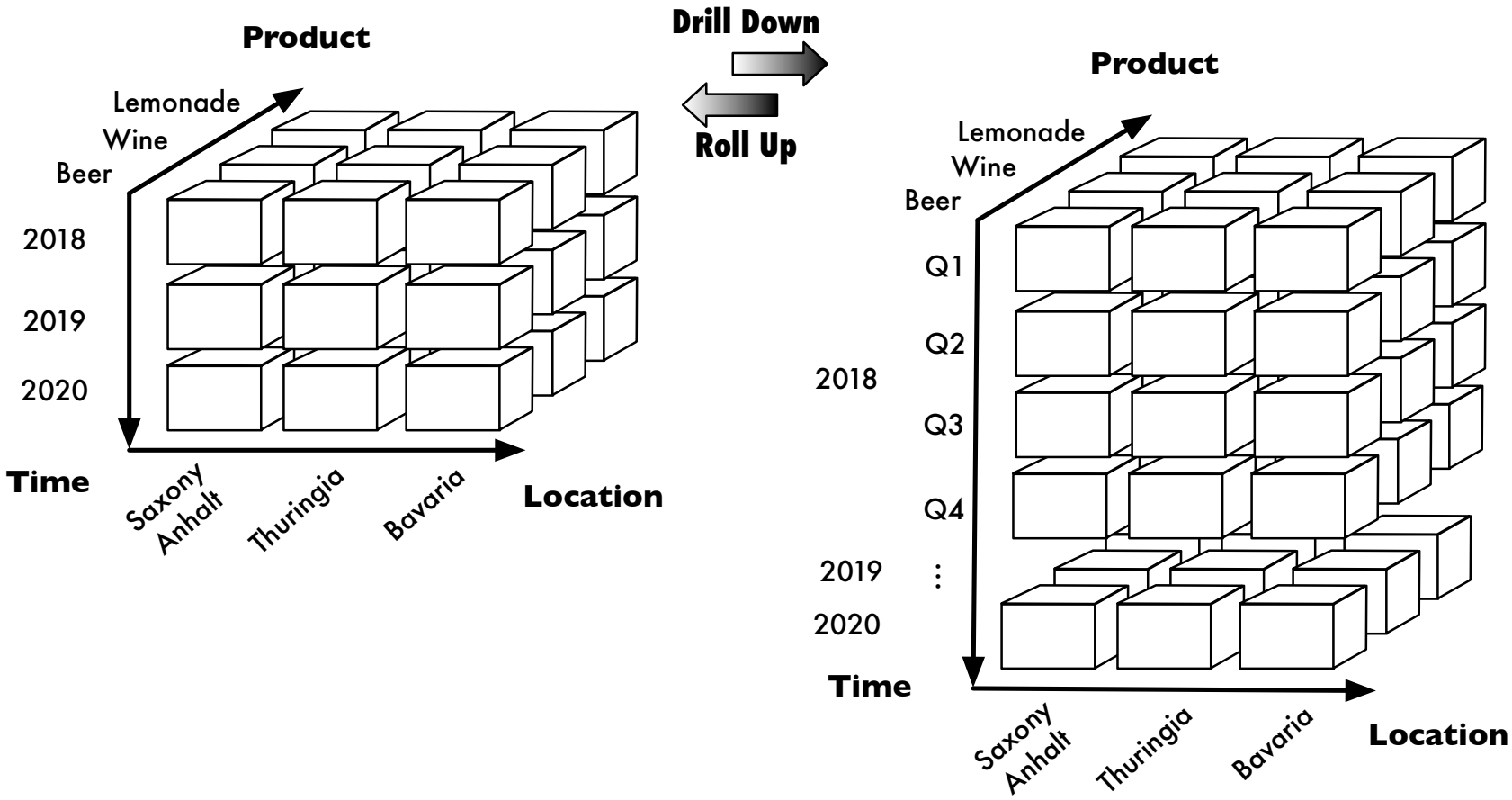
- Rotate the cube by swapping the dimensions
- Analysis of the data from different perspectives



Roll-Up, Drill-Down, Drill-Across

- Roll-Up:
  - Generating new information by aggregating data along the consolidation path
  - The same dimensionality persists
  - Example: day $\rightarrow$ month $\rightarrow$ quarter $\rightarrow$ year
- Drill-Down:
  - Complementary to Roll-Up
  - Navigate from aggregated data into detailed data along the classification hierarchy
- Drill-Across:
  - Change from one cube to another

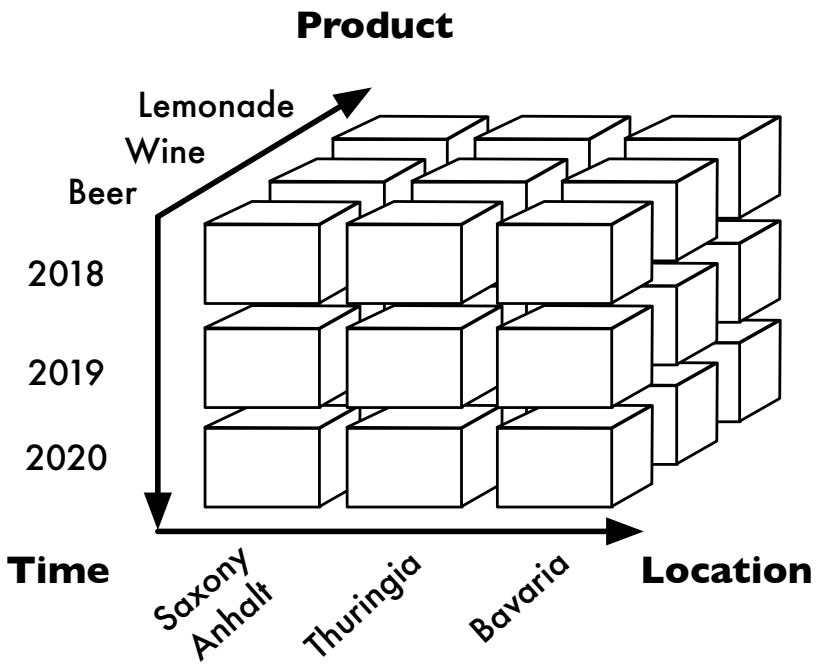
Roll-Up and Drill-Down

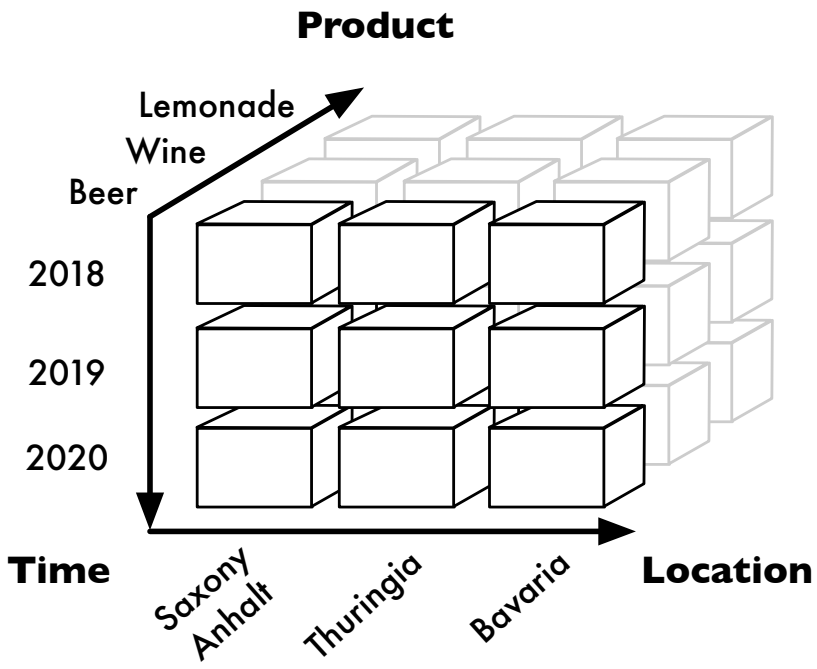


Slice and Dice

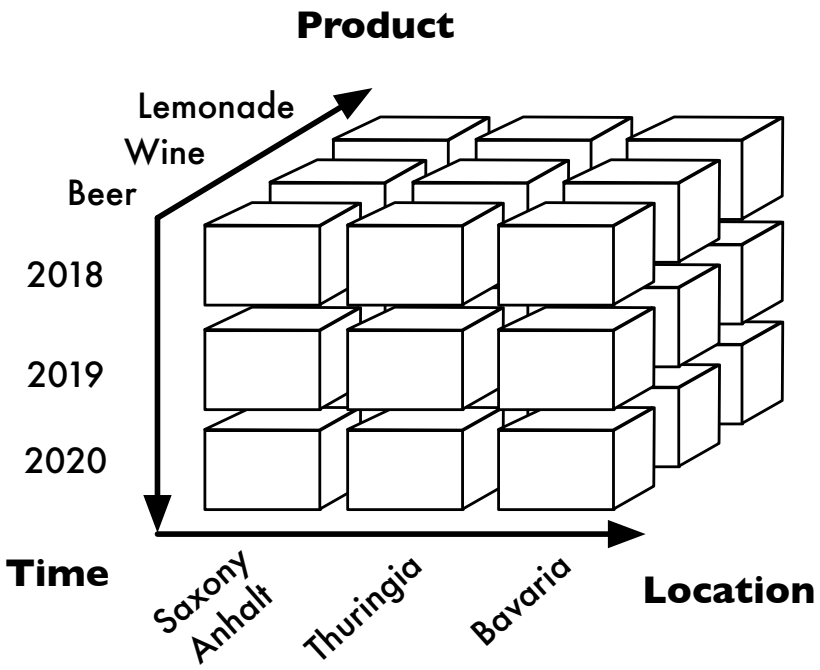
- Generating individual views
- Slice:
  - Cutting out "slices" from the cube
  - Reducing the dimensionality by conditioning the dimensions
  - For example, all values of the current year
  - Corresponds to the relational selection in the dimensions
- Dice:
  - Cutting out a "partial cube"
  - Dimensionality persists, change of the hierarchy objects
  - Example: the values of certain products or regions
  - Corresponds to the relational selection of multiple dimensions

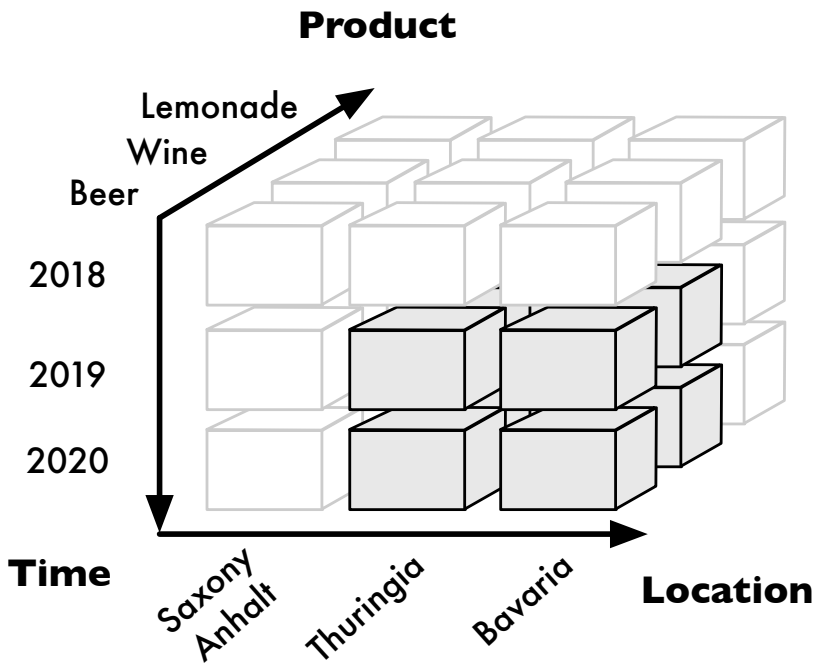
Slice


Product
Lemonade
Wine
Beer
2018
2019
2020
Time
Saxony Anhalt
Thuringia
Bavaria
Location


Product
Lemonade
Wine
Beer
2018
2019
2020
Time
Saxony Anhalt
Thuringia
Bavaria
Location

Dice


Product
Lemonade
Wine
Beer
2018
2019
2020
Time
Saxony Anhalt
Thuringia
Bavaria
Location


Product
Lemonade
Wine
Beer
2018
2019
2020
Time
Saxony Anhalt
Thuringia
Bavaria
Location

Relational Implementation of the multidimensional data model

Implementation of multi-dimensional data model I

- Multidimensional view
  - Modeling of the data
  - Query formulation
- Internal management of the data requires conversion to
  - Relational structures (tables)
    → ROLAP (relationales OLAP)
    - ★ Availability, maturity of the systems
  - Multidimensional structures (direct storage)
    → MOLAP (multidimensional OLAP)
    - ★ Elimination of the transformation or calculation beforehand
    - ★ Multidimensional arrays (facts) and associated dimension lists
  - Hybrid structure (hybrid)
    → HOLAP (hybrid OLAP)
    - ★ Detailed data stored relationally (ROLAP)
    - ★ Aggregates are stored multidimensionally (MOLAP)

Implementation of multi-dimensional data model II

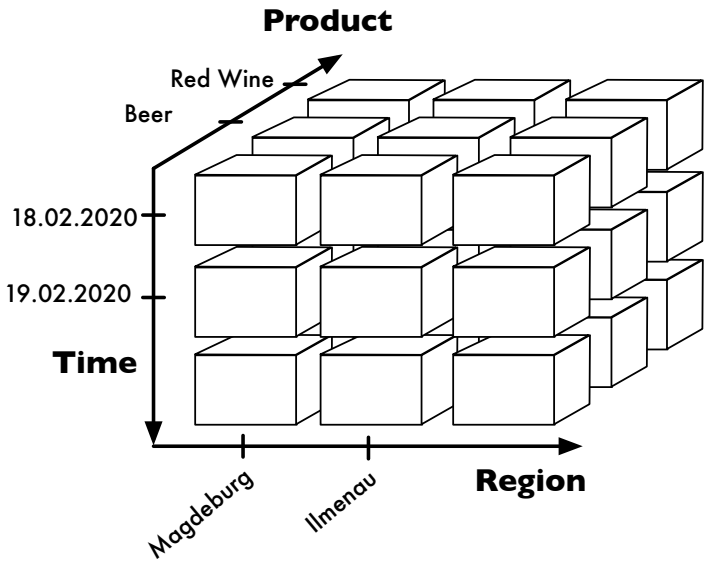
- Aspects
  - Storage
  - Query formulation and execution

Relational storage

- Avoid the loss of application-related semantics (from the multidimensional model, e.g., classification hierarchies)
- Efficient translation of multi-dimensional queries
- Efficient processing of the translated queries
- Easy maintenance of the resulting relations (e.g., loading new data)
- Consideration of the request characteristics and the data volume of analytical applications

Relational Implementation: Facts Table

- Starting point: implementation of the data cube without classification hierarchies
  - Dimensions, metrics $\rightarrow$ columns of the relation
  - Cells $\rightarrow$ tuple

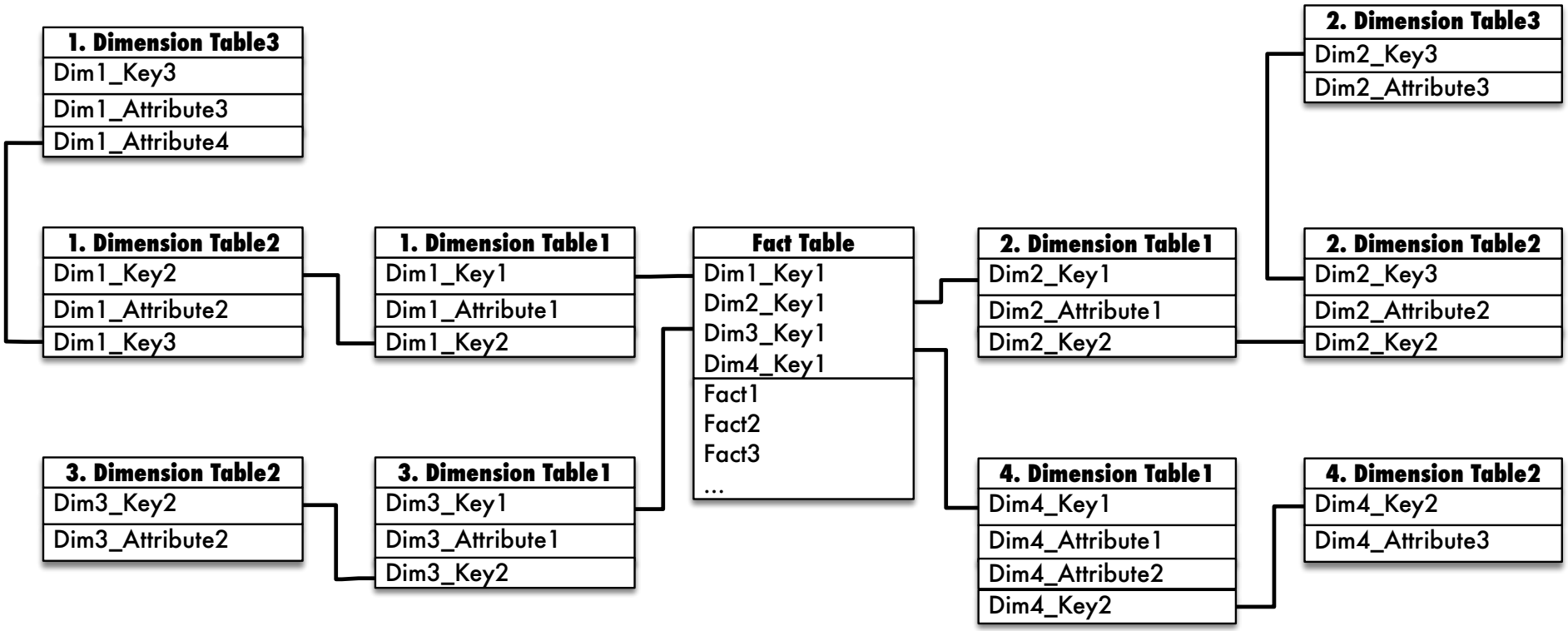


Product	Branch	Day	Sold
Red Wine	Magdeburg	18.02.19	145
Weissbier	Magdeburg	18.02.19	267
Red Wine	Ilmenau	18.02.19	70
	...		

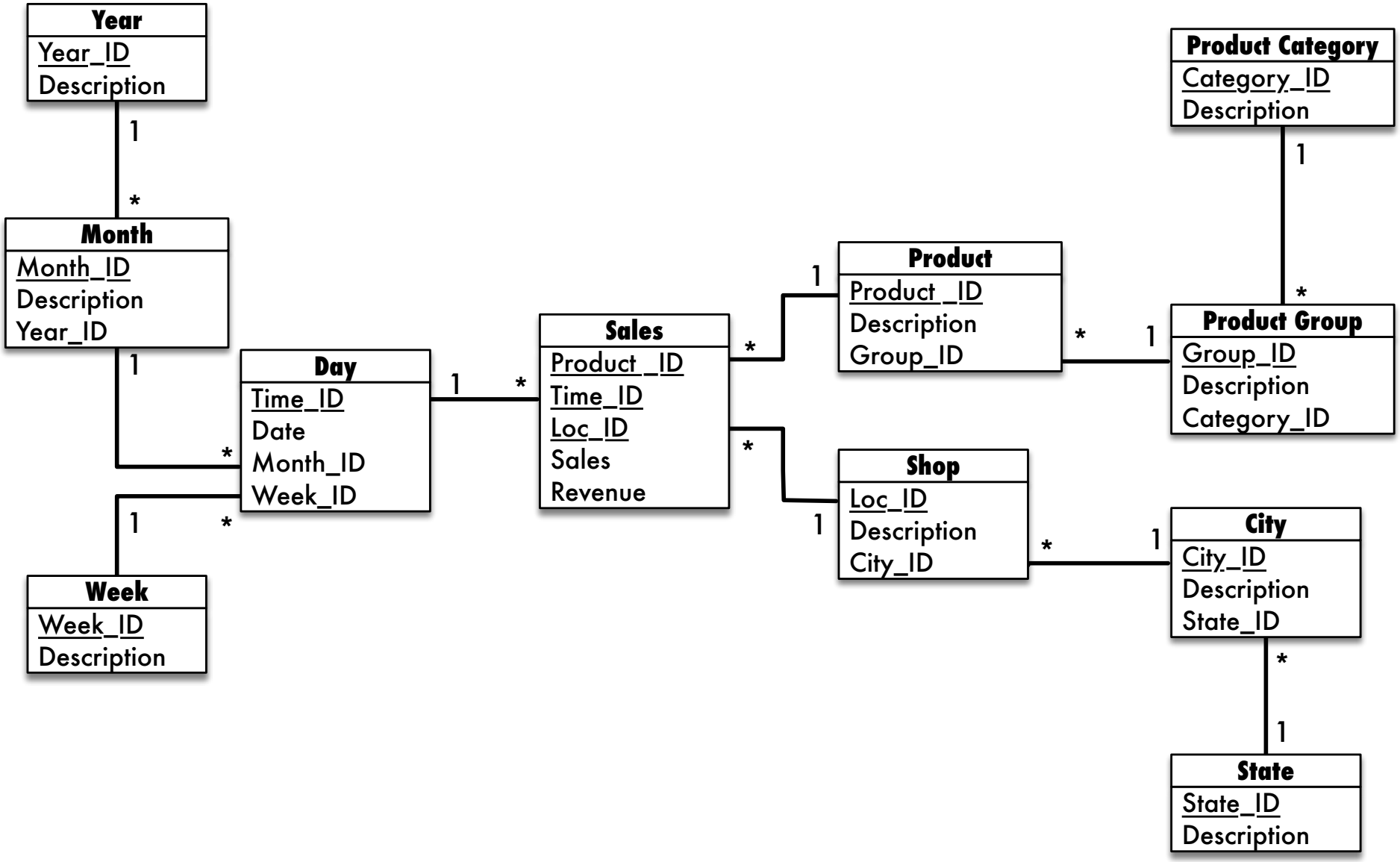
Snowflake-Schema

- Mapping of classifications: own table for each Classification level (e.g. articles, product group, etc.)
- Dimension table contains
  - ID for node classification
  - Descriptive attribute (e.g., brand, manufacturer, name)
  - Foreign keys of the directly superior classification stage
- Fact table contains (in addition to metrics):
  - Foreign keys of the respectively lowest classification level
  - Foreign keys form composite primary key for the fact table

Snowflake-Schema: Pattern



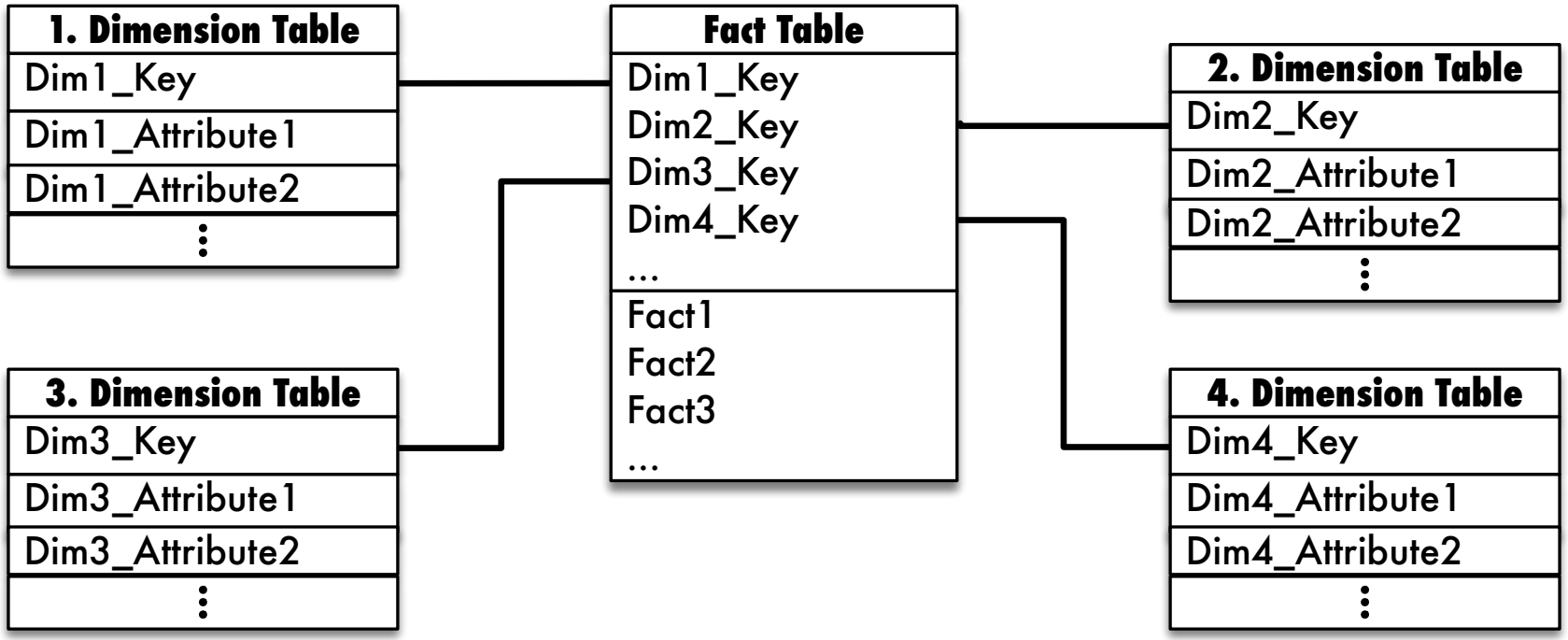
Snowflake-Schema: Example



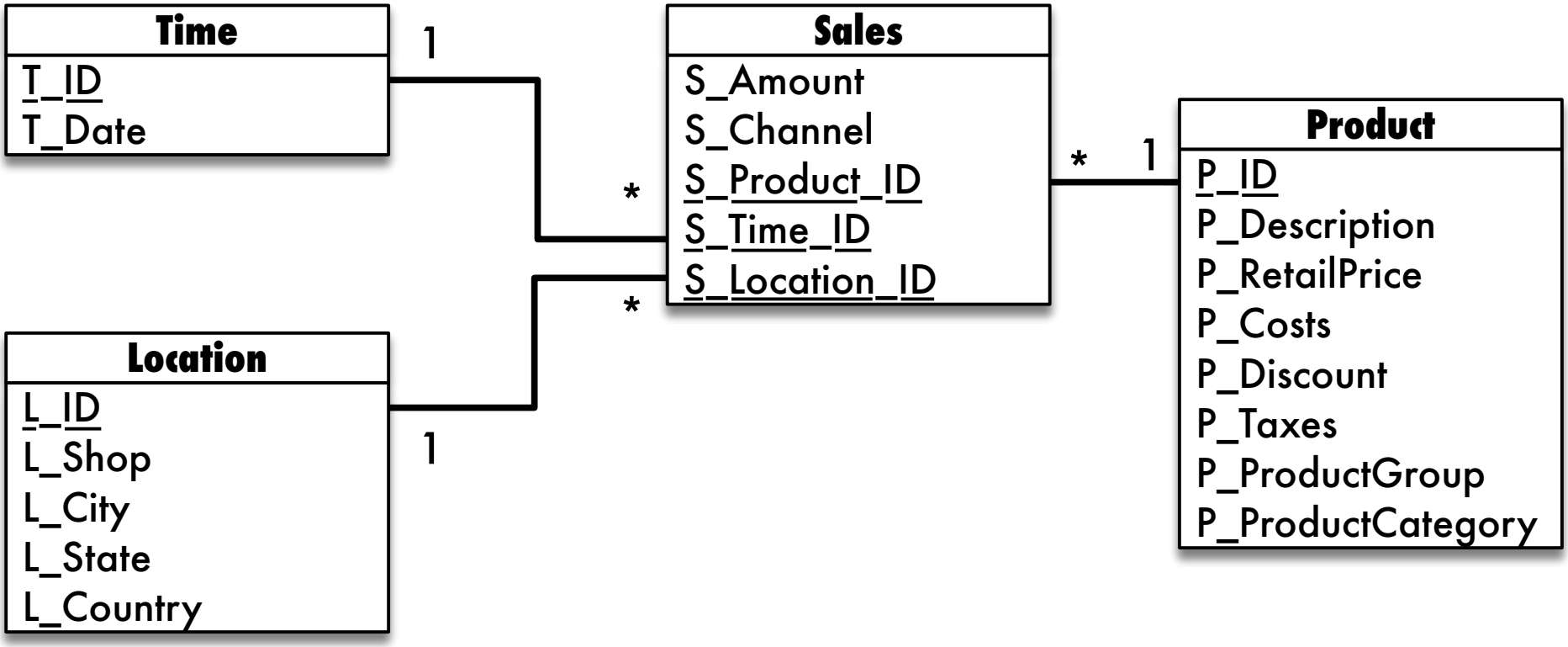
Star-Schema

- Snowflake-Schema is normalized: Prevention of update anomalies $\rightarrow$ 3. NF
  - But: requires multi-table join!
- Star-Schema:
  - Denormalization of tables belonging to a dimension $\rightarrow$ 1. NF
  - For each dimension exactly one dimension table
  - Redundancies in the dimension table for faster query processing
  - Example: Item, Product, Product Group, etc. as columns in a table Product

Star-Schema: Pattern



Star-Schema: Example



Star-Schema formally

- Multi-dimensional diagram with n dimensions
  - Dimension tabeles $D^1, \ldots, D^n$ of the form $D^i(Dim_i\_Key, A_{i,1}, \ldots, A_{i,k_i})$
  - Fact table $F(Dim_1\_Key, \ldots, Dim_n\_Key, f_1, \ldots, f_m)$ with m facts
- Each part of the composite primary key of the fact table is a foreign key to the primary key attribute of the corresponding dimension

`CREATE DIMENSION` in Oracle

- Foreign key constraints can be defined in SQL
- But functional relationships between attributes within a dimension not possible
- Oracle-Extension: `CREATE DIMENSION`
  - "informative" assurance
  - Correctness is not checked by DBS
  - Usage during query rewriting on materialized views

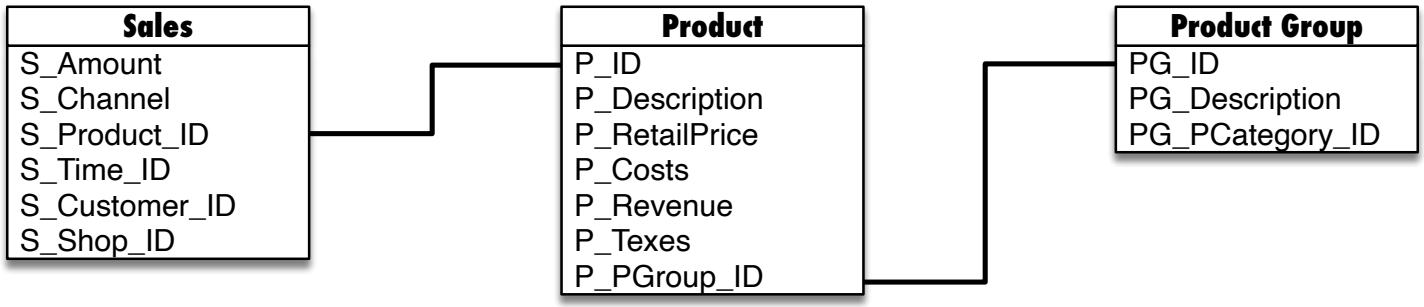
Beispiel CREATE DIMENSION

```
CREATE DIMENSION ProductDimension
    LEVEL Product IS (Product.P_ID)
    LEVEL ProductGroup IS (Product.P_PGroup_ID)
    LEVEL ProductCategory IS
        (Product.P_ProductCategory)
HIERARCHY ProductHierarchy (
    Product CHILD OF
    ProductGroup CHILD OF
    ProductCategory)
ATTRIBUTE Product DETERMINES (P_Name,
    P_Price, P_Cost,
    P_Sconto, P_Taxes)
ATTRIBUTE ProductGroup DETERMINES
    (P_Produktgroup)
```

Keywords

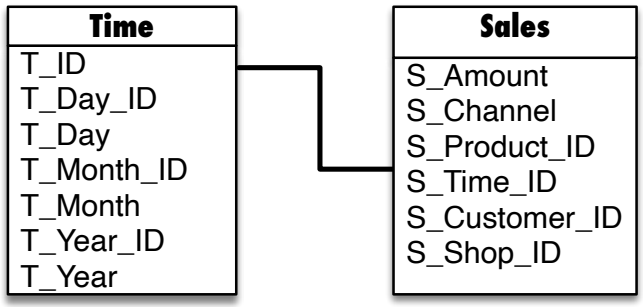
- `LEVEL`
  - Defines classification levels
- `HIERARCHY`
  - Determining the dependencies of the classification levels
- `ATTRIBUTE ... DETERMINES`
  - Defines relationship between classification attribute and dimensional attributes

Snowflake with `CREATE DIMENSION`




```
CREATE DIMENSION ProductDimension
   LEVEL Product IS (Produkt.P_ID)
   LEVEL ProductGroup IS (Productgroup.PG_ID)
HIERARCHY ProductHierarchy(
   Product CHILD OF ProductGroup)
JOIN KEY (Product.P_PGroup_ID)
   REFERENCES ProductGroup
```

Star with CREATE DIMENSION

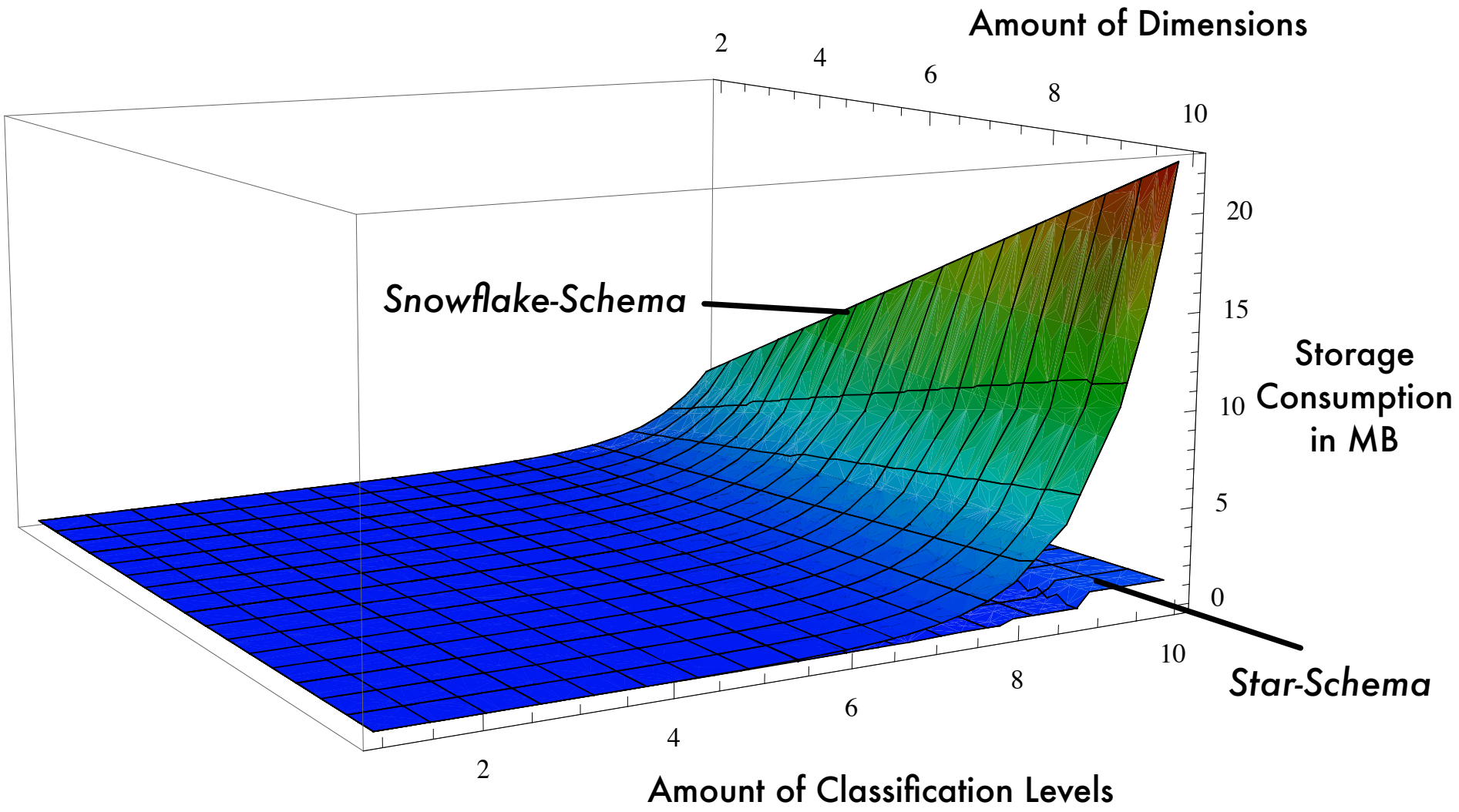



```
CREATE DIMENSION TimeDimension
    LEVEL Day IS (Time.T_Day_ID)
   LEVEL Month IS (Time.T_Month_ID)
   LEVEL Year IS (Time.T_Year_ID)
HIERARCHY TimeHierarchy (
   Day  CHILD OF Month CHILD OF Year)
ATTRIBUTE Day DETERMINES (T_Day)
ATTRIBUTE Month DETERMINES (T_Month)
ATTRIBUTE Year DETERMINES (T_Year)
```

Comparison Star and Snowflake

- Characteristics of DW applications
  - Typically restrictions in requests at a higher granularity (join operations)
  - Low data volume of the dimension tables compared to fact tables
  - Rare changes to classifications (risk of update anomalies)
- Advantages of the Star Schema
  - Simple structure (simplified query formulation)
  - Simple and flexible presentation of classification hierarchies (Columns in dimension tables)
  - Efficient query processing within a dimension (no Join operation required)

Costs Star vs. Snowflake



Query for Star and Snowflake

- Query: Sales of the product group "Wine" per city and year
- Snowflake-Schema:

```
SELECT  C_Name, YEAR(T_Date), SUM(S_Quantity)
FROM Sales, Branches, City, Product, ProductGroup,
Time
WHERE S_Product_ID = P_ID AND
    P_PGroup_ID = PG_ID AND
    S_Branch_ID = B_ID AND
    B_City_ID = C_ID AND
    S_Time_ID = T_ID AND
    PG_Name = 'Wine'
GROUP BY C_Name, YEAR(T_Date)
```

- Number of Joins: 5
  (increases linearly by the number of aggregation paths)

Query for Star and Snowflake (2)

- Query for Star-Schema:

```
SELECT Pl_City, YEAR(T_Date), SUM(S_Quantity)
FROM Sales, Place, Product, Time
WHERE S_Product_ID = P_ID AND
    S_Time_ID = T_ID AND
    S_Place_ID = Pl_ID AND
    P_ProductGroup = 'Wine'
GROUP BY Pl_City, YEAR(T_Date)
```

- Number of Joins: 3
  (independent of the length of aggregation paths)

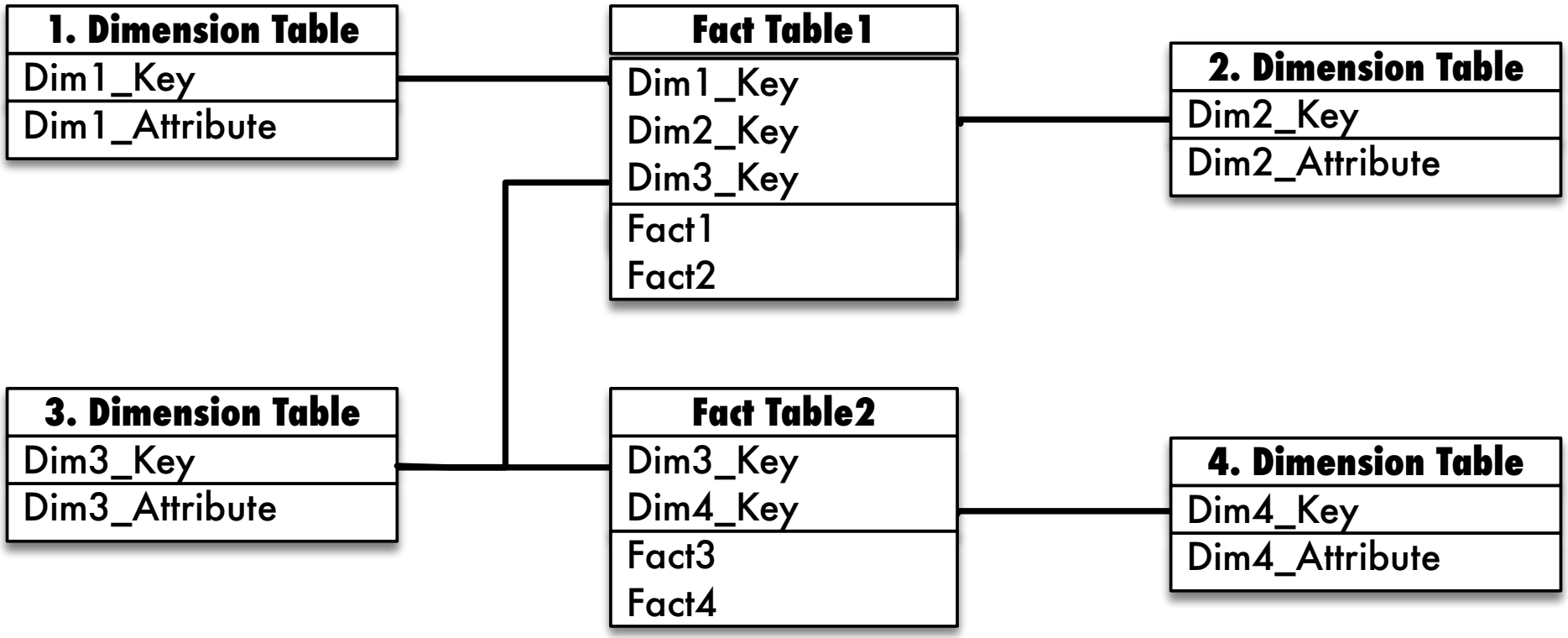
Hybrid Forms

- Implementation of the dimensions analogous to the snowflake or star schema
- Decision criteria:
  - Frequency of change of the dimensions:
    - ★ Reduction of maintenance effort due to normalization (Snowflake)
  - Number of classification levels of a dimension:
    - ★ More classification levels $\rightarrow$ greater redundancy in the star schema
  - Number of dimension elements:
    - ★ Space savings through normalization with many elements of a dimension at the lowest classification level
  - Materialization of aggregates:
    - ★ Improved performance with normalization on materialized aggregates for a classification level

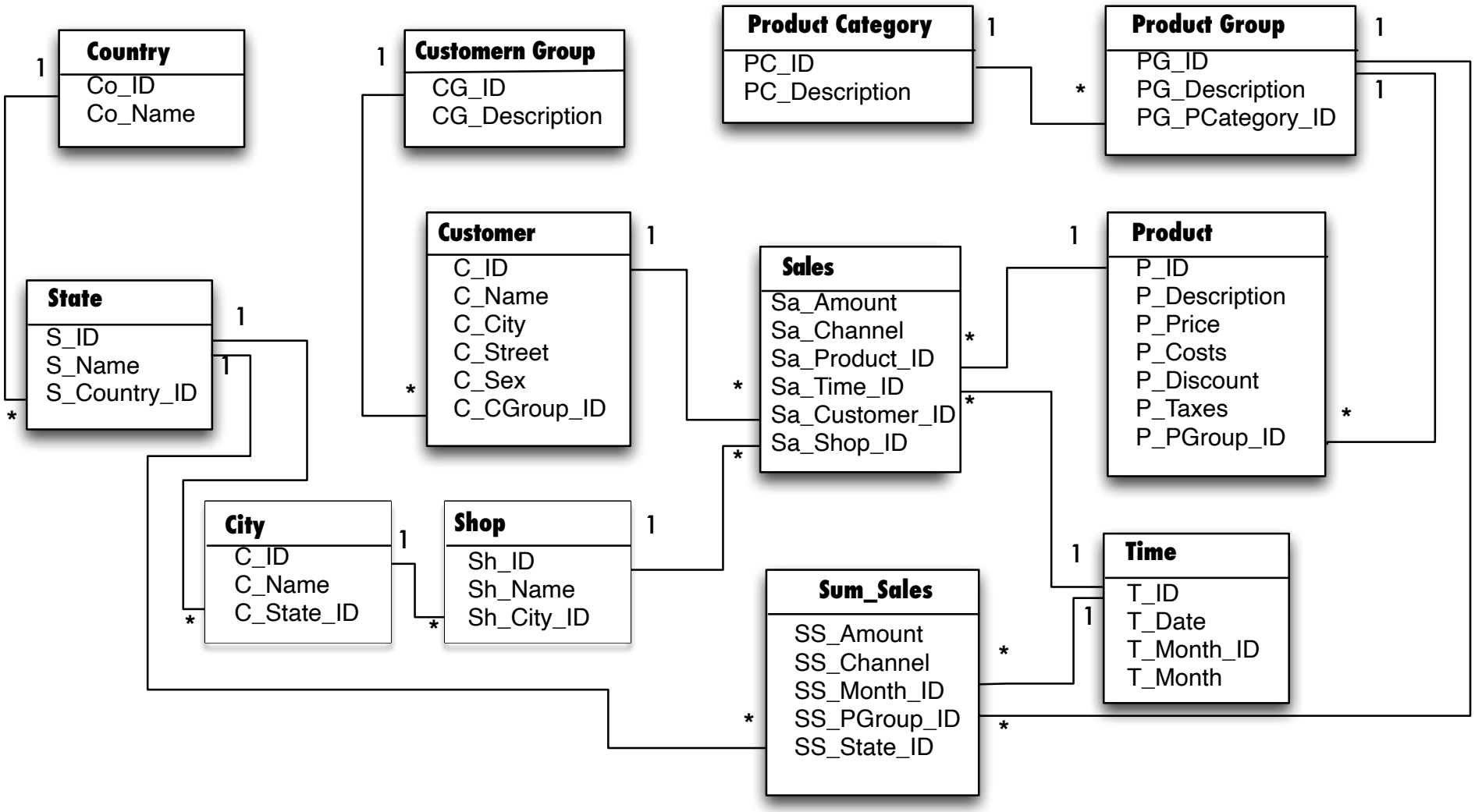
Galaxy Schema

- Star- and Snowflakeschema
  - One fact table
  - Multiple metrics only possible with the same dimensions
- Galaxy Schema
  - Multiple fact tables
  - Partially linked wit the same dimension tables
  - Also: Multi fact tables schema, Multi-Cube, Hyper-Cube

Galaxy Schema: Pattern



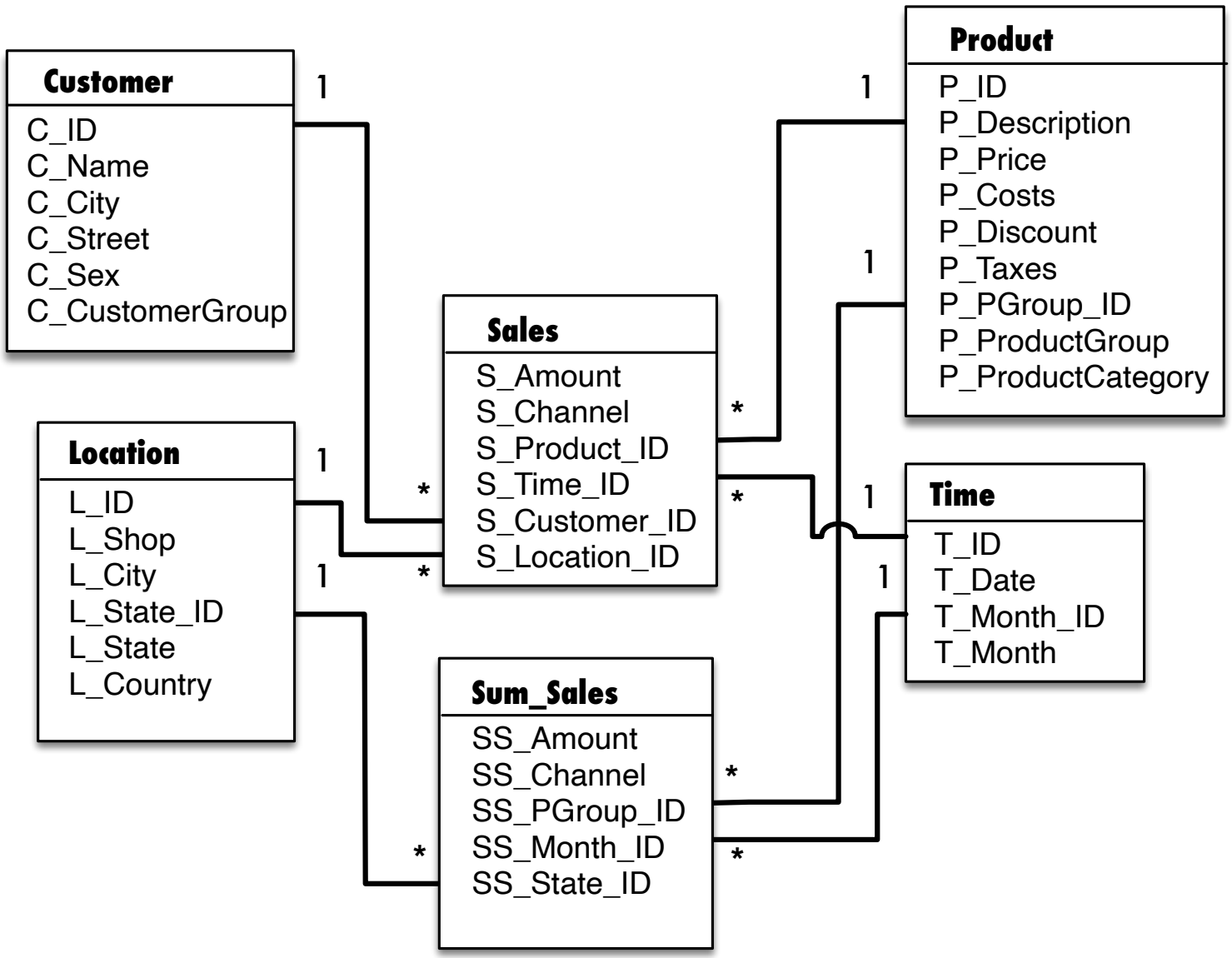
Galaxy Schema: Example



Fact Constellation

- Storing of precomputed aggregates in fact table
  - Example: Revenue by region
  - Differentiation in dimension table with special attributes (e.g.: "Level")
- Alternative: Storing in a separate fact table
  - Fact Constellation Schema (special case of a galaxy schema)

Fact Constellation: Example



Representation of classification hierarchies

- Horizontal: modeling of the stages of the classification hierarchy as columns of the denormalized dimension table
  - Advantages:
    - ★ Constraints on higher granularity without Join
  - Disadvantages:
    - ★ Duplicate elimination for queries of certain levels (e.g.: Product group within a category)
    - ★ Schema change when adding new levels

Product_ID	Product	Product Group	Product Category
1234	Immer Ultra	Hygene	Cosmetics
1235	Putzich	Hygiene	Cosmetics
2345	Rohrfrei	Cleaner	Household

Representation of classification hierarchies

- Vertical (recursively): normalized dimension table with attributes Attributen
  - Dimension_ID: keys for fact table
  - Parent_ID: Attribute value of the dimension ID of the next higher level
- Advantages:
  - Easy to change the classification schema
  - Simple treatment of pre-aggregates
- Disadvantages:
  - Self-join queries for individual steps (e.g.: product group within a category)

Dimension_ID	Parent_ID
Immer Ultra	Hygiene
Hygiene	Cosmetics
Putzich	Hygiene

Representation of classification hierarchies

- Combined: combination of the two strategies
  - Representation of the classification levels as columns (but with generic names)
  - Storage of the nodes of all higher levels as a tuple
  - Additional attribute "level"' $\rightarrow$ Indication of the designated classification level

Dimension_ID	Level1_ID	Level2_ID	Level
Immer Ultra	Hygiene	Cosmetics	0
Putzich	Hygiene	Cosmetics	0
Hygiene	Cosmetics	NULL	1
Kosmetik	NULL	NULL	2

Avoid losses of the semantics

- Loss of semantics in relational implementation:
  - Distinction between code and dimension (attributes of fact table)
  - Attributes of dimension tables (descriptive, structure of the hierarchy)
  - Structure of the dimensions (drill paths)
- Remedy:
  - Expansion of the system catalog metadata for multi-dimensional applications
  - Example: CREATE DIMENSION, HIERARCHY in Oracle

Problems of the relational implementation

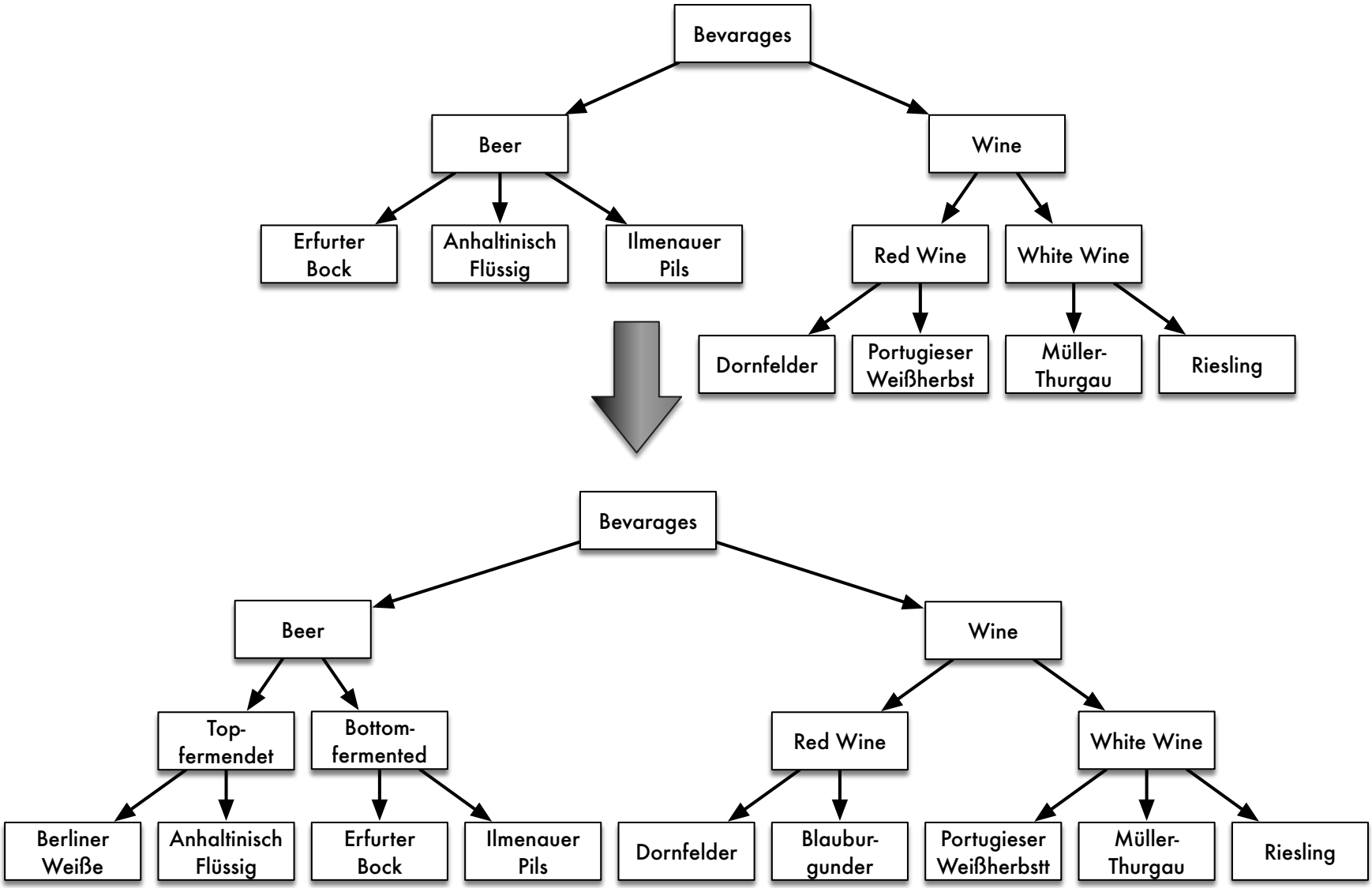
- Transformation of multi-dimensional queries in relational implementation necessary $\rightarrow$ complex queries
- Use of complex query tools necessary (OLAP tools)
- Loss of semantics
- Therefore: direct multidimensional storage $\rightarrow$ see Part VI

Slowly Changing Dimensions

Slowly Changing Dimensions

- Historicization: Changes of attribute characteristics, relationships and entities over time
- Structural changes in the dimensions
- Schema changes in the dimensions
- Influence on analyzes

Slowly Changing Dimensions: Example



Analysis requirements

- on the current structure: **as is**
- according to a defined historical structure: **as of**
- according to historical truth: **as posted**
- **comparable results**

Example for Slowly Changing Dimensions

Product Dimension 2019

Product	Product Group
Müller-Thurgau	White Wine
Riesling	White Wine
Dornfelder	Red Wine
Portugieser Weissherbst	Red Wine

Product Dimension 2020

Product	Product Group
Müller-Thurgau	White Wine
Riesling	White Wine
Portugieser Weissherbst	White Wine
Blauburgunder	Red Wine
Dornfelder	Red Wine

Facts

Product	Year	Parcels
Dornfelder	2019	1000
Müller-Thurgau	2019	1000
Portugieser Weissherbst	2019	1000
Riesling	2019	1000
Blauburgunder	2020	1000
Dornfelder	2020	1000
Müller-Thurgau	2020	1000
Portugieser Weissherbst	2020	1000
Riesling	2020	1000

Reporting Requirements

Current structure

Product Group	Parcels 2019	Parcels 2020
White Wine	3000	3000
Red Wine	1000	2000

Old Structure

Product Group	Parcels 2019	Parcels 2020
White Wine	2000	2000
Red Wine	2000	2000

Historical Truth

Product Group	Parcels 2019	Parcels 2020
White Wine	2000	3000
Red Wine	2000	2000

Comparable Results

Product Group	Parcels 2019	Parcels 2020
White Wine	2000	2000
Red Wine	1000	1000

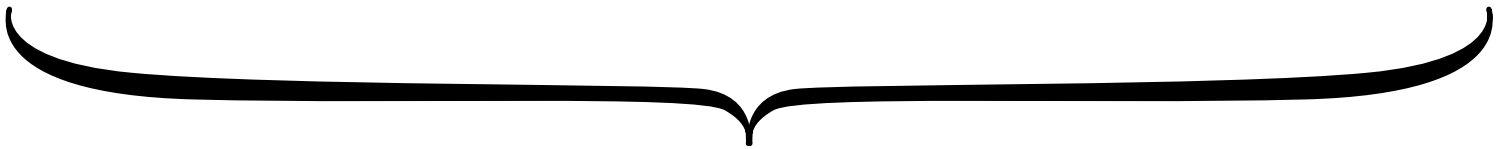
. . . with current structure

Product Dimension 2020

Product	Product Group
Müller-Thurgau	White Wine
Riesling	White Wine
Portugieser Weissherbst	White Wine
Blauburgunder	Red Wine
Dornfelder	Red Wine

Facts

Product	Year	Parcels
Dornfelder	2019	1000
Müller-Thurgau	2019	1000
Portugieser Weissherbst	2019	1000
Riesling	2019	1000
Blauburgunder	2020	1000
Dornfelder	2020	1000
Müller-Thurgau	2020	1000
Portugieser Weissherbst	2020	1000
Riesling	2020	1000



Current Structure

Product Group	Parcels 2019	Parcels 2020
White Wine	3000	3000
Red Wine	1000	2000

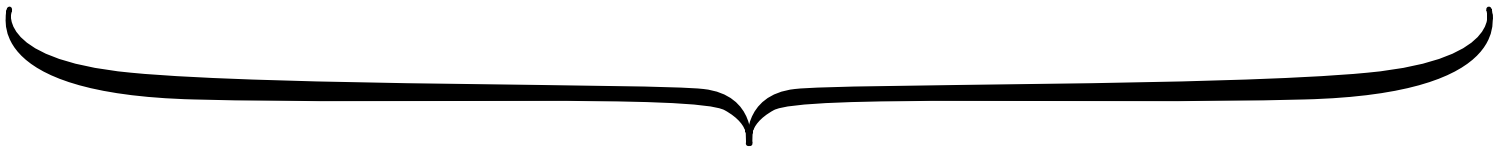
. . . old structure

Produktdimension 2019

Product	Product Group
Müller-Thurgau	White Wine
Riesling	White Wine
Dornfelder	Red Wine
Portugieser Weissherbst	Red Wine

Facts

Product	Year	Parcels
Dornfelder	2019	1000
Müller-Thurgau	2019	1000
Portugieser Weissherbst	2019	1000
Riesling	2019	1000
Blauburgunder	2020	1000
Dornfelder	2020	1000
Müller-Thurgau	2020	1000
Portugieser Weissherbst	2020	1000
Riesling	2020	1000



alte Struktur

Product Groupe	Parcels 2019	Parcels 2020
White Wine	2000	2000
Red Wine	2000	2000

. . . according to historical truth

Product Dimension 2019

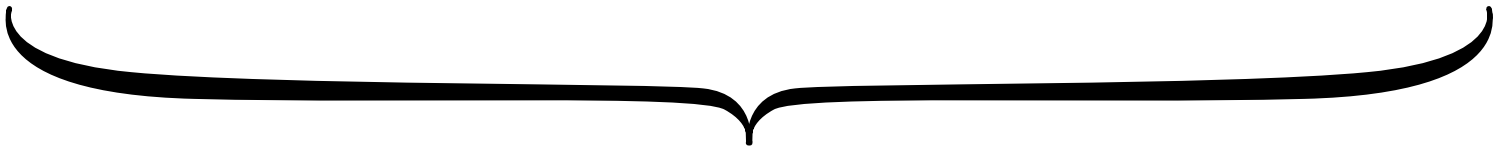
Product	Product Group
Müller-Thurgau	White Wine
Riesling	White Wine
Dornfelder	Red Wine
Portugieser Weissherbst	Red Wine

Product Dimension 2020

Product	Product Group
Müller-Thurgau	White Wine
Riesling	White Wine
Portugieser Weissherbst	White Wine
Blauburgunder	Red Wine
Dornfelder	Red Wine

Facts

Product	Year	Parcels
Dornfelder	2019	1000
Müller-Thurgau	2019	1000
Portugieser Weissherbst	2019	1000
Riesling	2019	1000
Blauburgunder	2020	1000
Dornfelder	2020	1000
Müller-Thurgau	2020	1000
Portugieser Weissherbst	2020	1000
Riesling	2020	1000



historical truth

Product Group	Parcels 2019	Parcels 2020
White Wine	2000	3000
Red Wine	2000	2000

. . . with comparable results

Product Dimension 2019

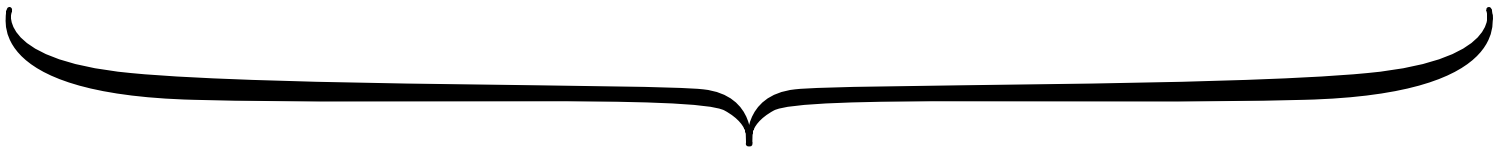
Product	Product Group
Müller-Thurgau	White Wine
Riesling	Weisswein
Dornfelder	Red Wine
Portugieser Weissherbst	Red Wine

Product Dimension 2020

Product	Product Group
Müller-Thurgau	White Wine
Riesling	White Wine
Portugieser Weissherbst	White Wine
Blauburgunder	Red Wine
Dornfelder	Red Wine

Facts

Product	Year	Parcels
Dornfelder	2019	1000
Müller-Thurgau	2019	1000
Portugieser Weissherbst	2019	1000
Riesling	2019	1000
Blauburgunder	2020	1000
Dornfelder	2020	1000
Müller-Thurgau	2020	1000
Portugieser Weissherbst	2020	1000
Riesling	2020	1000



comparable results

Product Group	Parcels 2019	Parcels 2020
White Wine	2000	2000
Red Wine	1000	1000

Implementation

- Adjustment of historical data material to new structures
  snall dataset, but old structures are not available
- Separate storage of historical data in addition to new structures
  large dataset and complexity for users, but old structures available
- Creating parallel hierarchies with all structures
  Dimension structure complex, any reporting possible
- Validity stamp
  any structures available, but performance question dependent on the use case

"As is"-Scenario

- Easy to implement
- No History
- Only for "as is"

Drink_ID	Product	Product Group
1	Dornfelder	Red Wine
2	Müller-Thurgau	White Wine

Drink_ID	Product	Product Group
1	Dornfelder	Red Wine
2	Müller-Thurgau	White Wine
~~3~~	~~Portugieser Weissherbst~~	~~Red Wine~~
3	Portugieser Weissherbst	White Wine

"As is" & quasi "as of"-Scenario

- Slightly higher memory requirements
- History without time allocation
- Equivalent to parallel hierarchy

Drink_ID	Product	Product Group	Old_Group
1	Dornfelder	Red Wine	Red Wine
2	Müller-Thurgau	White Wine	White Wine
3	Portugieser Weissherbst	Red Wine	Red Wine

Drink_ID	Product	Product Group	Old_Group
1	Dornfelder	Red Wine	Red Wine
2	Müller-Thurgau	White Wine	White Wine
3	Portugieser Weissherbst	White Wine	Red Wine

"As is" & "as of"-Scenario: Versioning

- Slightly higher memory requirements
- History without time allocation
- Artificial dimension key necessary

Drink_ID	Product	Product Group	Version	D_ID_old	active
1	Dornfelder	Red Wine	1	1	X
2	Müller-Thurgau	White Wine	1	2	X
3	Portugieser Weissherbst	Red Wine	1	3	X

Drink_ID	Product	Product Group	Version	D_ID_old	aktive
1	Dornfelder	Red Wine	1	1	X
2	Müller-Thurgau	White Wine	1	2	X
3	Portugieser Weissherbst	Red Wine	1	3	
3	Portugieser Weissherbst	White Wine	2	6	X

"As is" & "as of"-Scenario: Timestamps

- Higher memory requirement
- History with time allocation
- Artificial dimension key (or composite keys) necessary
- linking in the facts with *Drink_ID*

G_ID	D_ID_old	Product	Product Group	Valid_From	Valid_Until
1	1	Dornfelder	Red Wine	01.01.2019	31.12.9999
2	2	Müller-Thurgau	White Wine	01.01.2019	31.12.9999
3	3	Portugieser Weissherbst	Red Wine	01.01.2019	31.12.9999

G_ID	D_ID_old	Product	Product Group	Valid_From	Valid_Until
1	1	Dornfelder	Red Wine	01.01.2019	31.12.9999
2	2	Müller-Thurgau	White Wine	01.01.2019	31.12.9999
3	3	Portugieser Weissherbst	Red Wine	01.01.2019	31.12.2019
3	6	Portugieser Weissherbst	White Wine	01.01.2020	31.12.9999

"As posted"-Scenario

- Higher memory requirement
- History of the transaction date
- Artificial dimension key (or composite Key) is necessary
- linking in the facts with *D_ID_old*

Drink_ID	Product	Product Group	Version	D_ID_old
1	Dornfelder	Red Wine	1	1
2	Müller-Thurgau	White Wine	1	2
3	Portugieser Weissherbst	Red Wine	1	3

Drink_ID	Product	Product Group	Version	D_ID_old
1	Dornfelder	Red Wine	1	1
2	Müller-Thurgau	White Wine	1	2
3	Portugieser Weissherbst	Red Wine	1	3
3	Portugieser Weissherbst	White Wine	2	6

"As is" & "as of" & "as posted"-Scenario: Snapshot

- Higher memory requirement in dimension and fact tables
- Flag for current assignment
- Artificial data for the load time
- only for "small" dimensions recommended

Drink_ID	Product	Product Group	Load Date	Current
1	Dornfelder	Red Wine	01.01.2019	0
2	Müller-Thurgau	White Wine	01.01.2019	0
3	Portugieser Weissherbst	Red Wine	01.01.2019	0
1	Dornfelder	Red Wine	01.01.2020	1
2	Müller-Thurgau	White Wine	01.01.2020	1
3	Portugieser Weissherbst	White Wine	01.01.2020	1

Summary

- Multidimensional Data Model:
  - Cubes, metrics, dimensions
  - OLAP operations
- conceptual modeling techniques
  - not a "quasi" standard as the ER model
  - specialized structure for multidimensional concepts
- Implementation of the multidimensional model
  - Relational: Star and Snowflake Schema
  - Hybrid forms
  - Avoid loss of semantics