Basic concepts

- Dimensions
- Facts / Metrics

![Diagram showing basic concepts]

- **Produkt**
  - Gruppe
  - Artikel

- **Zeit**
  - Jahr
  - Quartal
  - Monat

- **VerkaufsSORT**
  - Filiale

- **Kennzahl**
  - Umsatz

© Sattler / Saake / Köppen  
Data Warehouse Technologies  
Last Change: 23.11.2019
Basic concepts

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

© Sattler / Saake / Köppen  Data Warehouse Technologies  Last Change: 23.11.2019  0–2
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 $n$ ($n \geq 2$) 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 ($\{D^1, \ldots, D^n, \text{Top}_D\}; \rightarrow$)
  - Generic maximal element $\text{Top}_D$
  - Funktional dependence $\rightarrow$

- $\text{Top}_D$ is functionally determined by all the attributes:

$$\forall i, 1 \leq i \leq n : D_i \rightarrow \text{Top}_D$$

- 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

Basic concepts

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 functions**
  - 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

<table>
<thead>
<tr>
<th></th>
<th><strong>FLOW</strong></th>
<th><strong>STOCK</strong></th>
<th><strong>VPU</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregation over temporal dimension?</td>
<td>no</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><strong>MIN/MAX</strong></td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td><strong>SUM</strong></td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td><strong>AVG</strong></td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td><strong>COUNT</strong></td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>
Disjointness

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

<table>
<thead>
<tr>
<th>Revenue</th>
<th>2010</th>
<th>2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beer</td>
<td>38</td>
<td>42</td>
</tr>
<tr>
<td>Beer mix</td>
<td>27</td>
<td>31</td>
</tr>
<tr>
<td>Softdrinks</td>
<td>54</td>
<td>57</td>
</tr>
<tr>
<td>Total</td>
<td>92</td>
<td>99</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>Wine region</th>
<th>2010</th>
<th>2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rheinhessen</td>
<td>152</td>
<td>153</td>
</tr>
<tr>
<td>Saale-Unstrut</td>
<td>98</td>
<td>104</td>
</tr>
<tr>
<td>Mosel</td>
<td>161</td>
<td>172</td>
</tr>
<tr>
<td>Others</td>
<td>20</td>
<td>22</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>431</strong></td>
<td><strong>451</strong></td>
</tr>
</tbody>
</table>

- Others = Some producers from other areas
Simpson Paradox

- Group evaluations give different results in terms of the observed level of aggregation.
- Quotients on strongly differing group sizes

<table>
<thead>
<tr>
<th></th>
<th>Red wine</th>
<th></th>
<th>White wine</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>high</td>
<td>Total</td>
<td>high</td>
<td>Total</td>
</tr>
<tr>
<td>high rating</td>
<td>100%</td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>2011</td>
<td>10</td>
<td>15</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Sum</td>
<td>15</td>
<td>20</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>75%</td>
<td>8</td>
<td>80%</td>
<td></td>
</tr>
</tbody>
</table>
Aggregation functions

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

- **Distributive aggregation function:**
  \[ \exists g : f(X) = f(g(X_1), g(X_2), \ldots, g(X_n)) \]

- **Algebraic aggregatsfunktion:** $f$ ist calculable from a fixed set $G$

- **Holistic aggregattion function:** $f$ can only be calculated from the basic elements of $X$

<table>
<thead>
<tr>
<th>Aggregation type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributive</td>
<td>$SUM()$, $COUNT()$, $MAX()$, $MIN()$</td>
</tr>
<tr>
<td>Algebraic</td>
<td>$AVG()$ mit $g_1 := SUM()$ und $g_2 := COUNT()$, $STDDEV()$</td>
</tr>
<tr>
<td>Holistic</td>
<td>$MEDIAN()$, $RANK()$, $PERCENTILE()$</td>
</tr>
</tbody>
</table>
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
- Visualization
  - 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

\[ \forall i, 1 \leq i \leq n, \forall j, 1 \leq j \leq n, i \neq j \neg \exists k, l : D^i.D_k \rightarrow D^i.D_l \]
Multidimensional Data Cube
**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 → 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

- Faktename
- Ebene
- Attribute

Faktenbeziehung
Klassifikationsstufe
Klassifikationsbeziehung
Attributbezeichnung

© Sattler / Saake / Köppen
Data Warehouse Technologies
Last Change: 23.11.2019
ME/R: Example
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
dimensionales Attribut

Berechnungsformel

Dimension

Hierarchie

Hierarchiestufe

Dimensionsausprägung

Dimensionsausschnitt
ADAPT: Beispiel
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 → month → quarter → 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
Dice
Problems of the dimension change

- Change or omission of a dimension may be part of the analysis
- Example: instead of the cube by product, place and time the combinations of customer, location and time may be interesting.

<table>
<thead>
<tr>
<th>Product</th>
<th>Branch</th>
<th>Day</th>
<th>Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Wine</td>
<td>Magdeburg</td>
<td>18.02.10</td>
<td>145</td>
</tr>
<tr>
<td>Wheat beer</td>
<td>Magdeburg</td>
<td>18.02.10</td>
<td>267</td>
</tr>
<tr>
<td>Red Wine</td>
<td>Ilmenau</td>
<td>18.02.10</td>
<td>70</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Customer Group</th>
<th>Branch</th>
<th>Day</th>
<th>Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weintrinker</td>
<td>Magdeburg</td>
<td>18.02.10</td>
<td>210</td>
</tr>
<tr>
<td>Bayer</td>
<td>Magdeburg</td>
<td>18.02.10</td>
<td>407</td>
</tr>
<tr>
<td>Geniesser</td>
<td>Ilmenau</td>
<td>18.02.10</td>
<td>35</td>
</tr>
</tbody>
</table>

What does a change in the dimension mean?
Marginalization

- Edge summation when removing a dimension
- Corresponds to the relational projection
- Depending on the type of aggregation function → Problem of interpretation

<table>
<thead>
<tr>
<th></th>
<th>Share of Beer Bottles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Magdeburg</td>
</tr>
<tr>
<td>1. Half Week</td>
<td>95%</td>
</tr>
<tr>
<td>2. Half Week</td>
<td>68%</td>
</tr>
<tr>
<td>Whole Week</td>
<td>74%</td>
</tr>
</tbody>
</table>
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 of the multidimensional data model

Relational Implementation: Facts Table

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

<table>
<thead>
<tr>
<th>Product</th>
<th>Branch</th>
<th>Day</th>
<th>Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotwein</td>
<td>Magdeburg</td>
<td>18.02.10</td>
<td>145</td>
</tr>
<tr>
<td>Weissbier</td>
<td>Magdeburg</td>
<td>18.02.10</td>
<td>267</td>
</tr>
<tr>
<td>Rotwein</td>
<td>Ilmenau</td>
<td>18.02.10</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© Sattler / Saake / Köppen  Data Warehouse Technologies  Last Change: 23.11.2019  0–43
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

1. Dimensionstabelle3
   - Dim1_Schlüssel3
   - Dim1_Attribut3
   - Dim1_Attribut4

2. Dimensionstabelle3
   - Dim2_Schlüssel3
   - Dim2_Attribut3

3. Dimensionstabelle2
   - Dim3_Schlüssel2
   - Dim3_Attribut2

4. Dimensionstabelle2
   - Dim4_Schlüssel2
   - Dim4_Attribut2
   - Dim4_Attribut3

1. Dimensionstabelle1
   - Dim1_Schlüssel1
   - Dim1_Attribut1
   - Dim1_Schlüssel2

2. Dimensionstabelle1
   - Dim2_Schlüssel1
   - Dim2_Attribut1
   - Dim2_Schlüssel2

3. Dimensionstabelle1
   - Dim3_Schlüssel1
   - Dim3_Attribut1
   - Dim3_Schlüssel2

4. Dimensionstabelle1
   - Dim4_Schlüssel1
   - Dim4_Attribut1
   - Dim4_Schlüssel2

Faktentabelle
   - Dim1_Schlüssel1
   - Dim2_Schlüssel1
   - Dim3_Schlüssel1
   - Dim4_Schlüssel1
   - Fakt1
   - Fakt2
   - Fakt3
   - ...

1. Dimensionstabelle2
   - Dim1_Schlüssel2
   - Dim1_Attribut2
   - Dim1_Schlüssel3

2. Dimensionstabelle2
   - Dim2_Schlüssel2
   - Dim2_Attribut2
   - Dim2_Schlüssel3

3. Dimensionstabelle2
   - Dim3_Schlüssel2
   - Dim3_Attribut2

4. Dimensionstabelle2
   - Dim4_Schlüssel2
   - Dim4_Attribut2

© Sattler / Saake / Köppen

Data Warehouse Technologies

Last Change: 23.11.2019 0–45
Relational Implementation of the multidimensional data model

Snowflake-Schema: Example

Dimension “Produkt”
- **Produktkategorie**
  - PK_ID
  - PK_Bezeichnung
- **Produktgruppe**
  - PG_ID
  - PG_Bezeichnung
  - PG_PKategorie_ID

Dimension “Kunde”
- **Kundengruppe**
  - KG_ID
  - KG_Bezeichnung
- **Kunde**
  - K_ID
  - K_Name
  - K_Wohnort
  - K_Strasse
  - K_Geschlecht
  - K_KGruppe_ID

Dimension “Ort”
- **Bundesland**
  - B_ID
  - B_Name
  - B_Land_ID
- **Stadt**
  - S_ID
  - S_Name
  - S_BLand_ID
- **Filiale**
  - F_ID
  - F_Filiale
  - F_Stadt_ID

Dimension “Zeit”
- **Zeit**
  - Z_ID
  - Z_Datum

Diagram: Relationships between dimensions and tables.

- **Verkauf**
  - V_Anzahl
  - V_Kanal
  - V_Produkt_ID
  - V_Zeit_ID
  - V_Kunden_ID
  - V_Filial_ID

- **Produkt**
  - P_ID
  - P_Bezeichnung
  - P_Verkaufspreis
  - P_Einkaufspreis
  - P_Rabatt
  - P_Steuern
  - P_PGGruppe_ID

- **Kunde**
  - K_ID
  - K_Name
  - K_Wohnort
  - K_Strasse
  - K_Geschlecht
  - K_KGruppe_ID

- **Filiale**
  - F_ID
  - F_Filiale
  - F_Stadt_ID

- **Bundesland**
  - B_ID
  - B_Name
  - B_Land_ID

- **Stadt**
  - S_ID
  - S_Name
  - S_BLand_ID

- **Land**
  - L_ID
  - L_Name

- **Kundengruppe**
  - KG_ID
  - KG_Bezeichnung

- **Produktgruppe**
  - PG_ID
  - PG_Bezeichnung
  - PG_PKategorie_ID

- **Produktkategorie**
  - PK_ID
  - PK_Bezeichnung
Star-Schema

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

### Relations

1. **Dimensionstabelle**
   - Dim1_Schlüssel
   - Dim1_Attribut1
   - Dim1_Attribut2
   - ...

2. **Dimensionstabelle**
   - Dim2_Schlüssel
   - Dim2_Attribut1
   - Dim2_Attribut2
   - ...

3. **Dimensionstabelle**
   - Dim3_Schlüssel
   - Dim3_Attribut1
   - Dim3_Attribut2
   - ...

4. **Dimensionstabelle**
   - Dim4_Schlüssel
   - Dim4_Attribut1
   - Dim4_Attribut2
   - ...

### Faktentabelle

- Fakt1
- Fakt2
- Fakt3
- ...

© Sattler / Saake / Köppen

Data Warehouse Technologies

Last Change: 23.11.2019 0–48
Star-Schema: Example

```
<table>
<thead>
<tr>
<th></th>
<th>Kunde</th>
<th></th>
<th>Produkt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>K_ID</td>
<td>K_Name</td>
<td>P_ID</td>
<td>P_Bezeichnung</td>
</tr>
<tr>
<td>K_Wohnort</td>
<td>K_Strasse</td>
<td>P_Verkaufspreis</td>
<td>P_Einkaufspreis</td>
</tr>
<tr>
<td>K_Geschlecht</td>
<td>K_Kundengruppe</td>
<td>P_Rabatt</td>
<td>P_Steuern</td>
</tr>
<tr>
<td>Verkauf</td>
<td></td>
<td></td>
<td>Produktgruppe</td>
</tr>
<tr>
<td>V_Anzahl</td>
<td>V_Kanal</td>
<td>V_Product_ID</td>
<td>V_Zeit_ID</td>
</tr>
<tr>
<td>V_Kunden_ID</td>
<td>V_Ort_ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zeit</td>
<td></td>
<td></td>
<td>Ort</td>
</tr>
<tr>
<td>Z_ID</td>
<td>Z_Datum</td>
<td>O_ID</td>
<td>O_Filiale</td>
</tr>
<tr>
<td>O_Stadt</td>
<td>O_Bundesland</td>
<td>O_Land</td>
<td></td>
</tr>
</tbody>
</table>
```

© Sattler / Saake / Köppen

Data Warehouse Technologies

Last Change: 23.11.2019

0–49
Star-Schema formally

- Multi-dimensional diagram with $n$ dimensions
  - Dimension tables $D_1, \ldots, D^n$ 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
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 (Produkt.P_PGroup_ID)
REFERENCES ProductGroup
```
CREATE DIMENSION TimeDimension
  LEVEL Day IS (Time.Z_Day_ID)
  LEVEL Month IS (Time.Z_Month_ID)
  LEVEL Year IS (Time.Z_Year_ID)
HIERARCHY TimeHierarchy (  
  Tag CHILD OF Month CHILD OF Year)
ATTRIBUTE Day DETERMINES (Z_Day)
ATTRIBUTE Month DETERMINES (Z_Month)
ATTRIBUTE Year DETERMINES (Z_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)
Assumptions for cost considerations

- $n$ dimensions ($D^n$), for each $K$ classification levels plus Top
- Each classification node has 3 children
  - Level $i = K$: $1 = 3^0$ Nodes (Top)
  - Level $i = K - 1$: $3 = 3^1$ Nodes
  - Level $i = K - 2$: $9 = 3^2$ Nodes
  - ...
  - Level $i = 0$: Highest granularity, $3^K$ Nodes
  - $N_D = \sum_{i=0}^{K} 3^i$ Nodes per dimension

- $M$ facts, equally distributed in dimensions
- Attribute: $b$ Bytes; Nodes just have ID; $m$ Fact attributes
Full Classification Hierarchy

- For each node there are (the same amount of) facts
Query Costs Star vs. Snowflake

- **Storage Space Snowflake:**

  \[ (((n + m) \cdot M) + n \cdot N_D) \cdot b \]

  - \( n \) Number of foreign keys in a fact table
  - \( m \) Number of fact attributes
  - \( n \cdot N_D \) a tuple per classification node

- **Speicherplatz Star:**

  \[ (((n + m) \cdot M) + n \cdot 3^K \cdot K) \cdot b \]

  - \( n \cdot 3^K \) a tuple per classification node level 0
  - \( K \) an attribute per classification level
Costs Star vs. Snowflake

The diagram compares the storage requirements of Snowflake and Star schemas based on the number of classification levels and dimensions. The x-axis represents the number of classification levels, the y-axis shows the number of dimensions, and the z-axis indicates the storage requirement in MB.

- **Snowflake-Schema**: Shows a higher storage requirement as the number of classification levels and dimensions increases.
- **Star-Schema**: Demonstrates a lower storage requirement compared to the Snowflake schema, especially as the number of dimensions increases.

This visualization helps in understanding the trade-offs between these two schema designs in terms of storage efficiency.
Query for Star and Snowflake

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

```sql
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:

```sql
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 with the same dimension tables
- Also: Multi fact tables schema, Multi-Cube, Hyper-Cube
Galaxy Schema: Pattern

1. Dimensionstabelle
   Dim1_Schlüssel
   Dim1_Attribute

2. Dimensionstabelle
   Dim1_Schlüssel
   Dim2_Schlüssel
   Dim3_Schlüssel
   Fakt1
   Fakt2

3. Dimensionstabelle
   Dim3_Schlüssel
   Dim3_Attribute

4. Dimensionstabelle
   Dim3_Schlüssel
   Dim4_Schlüssel
   Fakt3
   Fakt4

Faktentabelle1

Faktentabelle2

2. Dimensionstabelle
   Dim2_Schlüssel
   Dim2_Attribute

4. Dimensionstabelle
   Dim4_Schlüssel
   Dim4_Attribute

© Sattler / Saake / Köppen
Data Warehouse Technologies
Last Change: 23.11.2019
Galaxy Schema: Example

Relational Implementation of the multidimensional datan model
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

Kunde
- K_ID
- K_Name
- K_Wohnort
- K_Strasse
- K_Geschlecht
- K_Kundengruppe

Ort
- O_ID
- O_Filiale
- O_Stadt
- O_BLand_ID
- O_Bundesland
- O_Land

Verkauf
- V_Anzahl
- V_Kanal
- V_Produkt_ID
- V_Zeit_ID
- V_Kunden_ID
- V_Ort_ID

Produkt
- P_ID
- P_Bezeichnung
- P_Verkaufspreis
- P_Einkaufspreis
- P_Rabatt
- P_Steuern
- P_PGruppe_ID
- P_Produktgruppe
- P_Produktkategorie

Summe_Verkauf
- SV_Anzahl
- SV_Kanal
- SV_PGruppe_ID
- SV_Monat_ID
- SV_BLand_ID

Zeit
- Z_ID
- Z_Datum
- Z_Monat_ID
- Z_Monat
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

<table>
<thead>
<tr>
<th>Product_ID</th>
<th>Product</th>
<th>Product Group</th>
<th>Product Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>Immer Ultra</td>
<td>Hygiene</td>
<td>Kosmetik</td>
</tr>
<tr>
<td>1235</td>
<td>Putzich</td>
<td>Hygiene</td>
<td>Kosmetik</td>
</tr>
<tr>
<td>2345</td>
<td>Rohrfrei</td>
<td>Reiniger</td>
<td>Haushalt</td>
</tr>
</tbody>
</table>
Representation of classification hierarchies

- Vertical (recursively): normalized dimension table with attributes
  
  - 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)

<table>
<thead>
<tr>
<th>Dimension_ID</th>
<th>Parent_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immer Ultra</td>
<td>Hygiene</td>
</tr>
<tr>
<td>Hygiene</td>
<td>Kosmetik</td>
</tr>
<tr>
<td>Putzich</td>
<td>Hygiene</td>
</tr>
</tbody>
</table>
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" → Indication of the designated classification level

<table>
<thead>
<tr>
<th>Dimension_ID</th>
<th>Level1_ID</th>
<th>Level2_ID</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immer Ultra</td>
<td>Hygiene</td>
<td>Kosmetik</td>
<td>0</td>
</tr>
<tr>
<td>Putzich</td>
<td>Hygiene</td>
<td>Kosmetik</td>
<td>0</td>
</tr>
<tr>
<td>Hygiene</td>
<td>Kosmetik</td>
<td>NULL</td>
<td>1</td>
</tr>
<tr>
<td>Kosmetik</td>
<td>NULL</td>
<td>NULL</td>
<td>2</td>
</tr>
</tbody>
</table>
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

- Historicization: Changes of attribute characteristics, relationships and entities over time
- Structural 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 2010

<table>
<thead>
<tr>
<th>Product</th>
<th>Product Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
</tr>
<tr>
<td>Riesling</td>
<td>Weisswein</td>
</tr>
<tr>
<td>Dornfelder</td>
<td>Rotwein</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
<td>Rotwein</td>
</tr>
</tbody>
</table>

### Product Dimension 2011

<table>
<thead>
<tr>
<th>Product</th>
<th>Product Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
</tr>
<tr>
<td>Riesling</td>
<td>Weisswein</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
<td>Weisswein</td>
</tr>
<tr>
<td>Blauburgunder</td>
<td>Rotwein</td>
</tr>
<tr>
<td>Dornfelder</td>
<td>Rotwein</td>
</tr>
</tbody>
</table>

### Facts

<table>
<thead>
<tr>
<th>Product</th>
<th>Year</th>
<th>Parcels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dornfelder</td>
<td>2010</td>
<td>1000</td>
</tr>
<tr>
<td>Müller-Thurgau</td>
<td>2010</td>
<td>1000</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
<td>2010</td>
<td>1000</td>
</tr>
<tr>
<td>Riesling</td>
<td>2010</td>
<td>1000</td>
</tr>
<tr>
<td>Blauburgunder</td>
<td>2011</td>
<td>1000</td>
</tr>
<tr>
<td>Müller-Thurgau</td>
<td>2011</td>
<td>1000</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
<td>2011</td>
<td>1000</td>
</tr>
<tr>
<td>Riesling</td>
<td>2011</td>
<td>1000</td>
</tr>
</tbody>
</table>
# Reporting Requirements

## Current structure

<table>
<thead>
<tr>
<th>Product Group</th>
<th>Parcels 2010</th>
<th>Parcels 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weisswein</td>
<td>3000</td>
<td>3000</td>
</tr>
<tr>
<td>Rotwein</td>
<td>1000</td>
<td>2000</td>
</tr>
</tbody>
</table>

## Old Structure

<table>
<thead>
<tr>
<th>Product Group</th>
<th>Parcels 2010</th>
<th>Parcels 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weisswein</td>
<td>2000</td>
<td>2000</td>
</tr>
<tr>
<td>Rotwein</td>
<td>2000</td>
<td>2000</td>
</tr>
</tbody>
</table>

## Historical Truth

<table>
<thead>
<tr>
<th>Product Group</th>
<th>Parcels 2010</th>
<th>Parcels 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weisswein</td>
<td>2000</td>
<td>3000</td>
</tr>
<tr>
<td>Rotwein</td>
<td>2000</td>
<td>2000</td>
</tr>
</tbody>
</table>

## Comparable Results

<table>
<thead>
<tr>
<th>Product Group</th>
<th>Parcels 2010</th>
<th>Parcels 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weisswein</td>
<td>2000</td>
<td>2000</td>
</tr>
<tr>
<td>Rotwein</td>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>
... with current structure

### Product Dimension 2011

<table>
<thead>
<tr>
<th>Product</th>
<th>Product Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
</tr>
<tr>
<td>Riesling</td>
<td>Weisswein</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
<td>Weisswein</td>
</tr>
<tr>
<td>Blauburgunder</td>
<td>Rotwein</td>
</tr>
<tr>
<td>Dornfelder</td>
<td>Rotwein</td>
</tr>
</tbody>
</table>

### Fakten

<table>
<thead>
<tr>
<th>Product</th>
<th>Year</th>
<th>Parcels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dornfelder</td>
<td>2010</td>
<td>1000</td>
</tr>
<tr>
<td>Müller-Thurgau</td>
<td>2010</td>
<td>1000</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
<td>2010</td>
<td>1000</td>
</tr>
<tr>
<td>Riesling</td>
<td>2010</td>
<td>1000</td>
</tr>
<tr>
<td>Blauburgunder</td>
<td>2011</td>
<td>1000</td>
</tr>
<tr>
<td>Dornfelder</td>
<td>2011</td>
<td>1000</td>
</tr>
<tr>
<td>Müller-Thurgau</td>
<td>2011</td>
<td>1000</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
<td>2011</td>
<td>1000</td>
</tr>
<tr>
<td>Riesling</td>
<td>2011</td>
<td>1000</td>
</tr>
</tbody>
</table>

### aktuelle Struktur

<table>
<thead>
<tr>
<th>Product Group</th>
<th>Parcels 2010</th>
<th>Parcels 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weisswein</td>
<td>3000</td>
<td>3000</td>
</tr>
<tr>
<td>Rotwein</td>
<td>1000</td>
<td>2000</td>
</tr>
</tbody>
</table>

© Sattler / Saake / Köppen

Data Warehouse Technologies

Last Change: 23.11.2019
... old structure

**Produktdimension 2010**

<table>
<thead>
<tr>
<th>Product</th>
<th>Product Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
</tr>
<tr>
<td>Riesling</td>
<td>Weisswein</td>
</tr>
<tr>
<td>Dornfelder</td>
<td>Rotwein</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
<td>Rotwein</td>
</tr>
</tbody>
</table>

**Fakten**

<table>
<thead>
<tr>
<th>Product</th>
<th>Year</th>
<th>Parcels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dornfelder</td>
<td>2010</td>
<td>1000</td>
</tr>
<tr>
<td>Müller-Thurgau</td>
<td>2010</td>
<td>1000</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
<td>2010</td>
<td>1000</td>
</tr>
<tr>
<td>Riesling</td>
<td>2010</td>
<td>1000</td>
</tr>
<tr>
<td>Blauburgunder</td>
<td>2011</td>
<td>1000</td>
</tr>
<tr>
<td>Dornfelder</td>
<td>2011</td>
<td>1000</td>
</tr>
<tr>
<td>Müller-Thurgau</td>
<td>2011</td>
<td>1000</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
<td>2011</td>
<td>1000</td>
</tr>
<tr>
<td>Riesling</td>
<td>2011</td>
<td>1000</td>
</tr>
</tbody>
</table>

**alte Struktur**

<table>
<thead>
<tr>
<th>Product Groupe</th>
<th>Parcels 2010</th>
<th>Parcels 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weisswein</td>
<td>2000</td>
<td>2000</td>
</tr>
<tr>
<td>Rotwein</td>
<td>2000</td>
<td>2000</td>
</tr>
</tbody>
</table>
Slowly Changing Dimensions

... according to historical truth

<table>
<thead>
<tr>
<th>Product Dimension 2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
</tr>
<tr>
<td>Müeller-Thurgau</td>
</tr>
<tr>
<td>Riesling</td>
</tr>
<tr>
<td>Dornfelder</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product Dimension 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
</tr>
<tr>
<td>Müeller-Thurgau</td>
</tr>
<tr>
<td>Riesling</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
</tr>
<tr>
<td>Blauburgunder</td>
</tr>
<tr>
<td>Dornfelder</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fakten</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>Dornfelder</td>
</tr>
<tr>
<td>Müeller-Thurgau</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
</tr>
<tr>
<td>Riesling</td>
</tr>
<tr>
<td>Blauburgunder</td>
</tr>
<tr>
<td>Müeller-Thurgau</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
</tr>
<tr>
<td>Riesling</td>
</tr>
</tbody>
</table>

historical truth

<table>
<thead>
<tr>
<th>Product Group</th>
<th>Parcels 2010</th>
<th>Parcels 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weisswein</td>
<td>2000</td>
<td>3000</td>
</tr>
<tr>
<td>Rotwein</td>
<td>2000</td>
<td>2000</td>
</tr>
</tbody>
</table>
... with comparable results

<table>
<thead>
<tr>
<th>Product Dimension 2010</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product</strong></td>
<td><strong>Product Group</strong></td>
</tr>
<tr>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
</tr>
<tr>
<td>Riesling</td>
<td>Weisswein</td>
</tr>
<tr>
<td>Dornfelder</td>
<td>Rotwein</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
<td>Rotwein</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product Dimension 2011</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product</strong></td>
<td><strong>Product Group</strong></td>
</tr>
<tr>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
</tr>
<tr>
<td>Riesling</td>
<td>Weisswein</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
<td>Weisswein</td>
</tr>
<tr>
<td>Blauburgunder</td>
<td>Rotwein</td>
</tr>
<tr>
<td>Dornfelder</td>
<td>Rotwein</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Facts</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product</strong></td>
</tr>
<tr>
<td>Dornfelder</td>
</tr>
<tr>
<td>Müller-Thurgau</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
</tr>
<tr>
<td>Riesling</td>
</tr>
<tr>
<td>Blauburgunder</td>
</tr>
<tr>
<td>Dornfelder</td>
</tr>
<tr>
<td>Müller-Thurgau</td>
</tr>
<tr>
<td>Portugieser Weissherbst</td>
</tr>
<tr>
<td>Riesling</td>
</tr>
</tbody>
</table>

**comparable results**

<table>
<thead>
<tr>
<th><strong>Product Group</strong></th>
<th><strong>Parcels 2010</strong></th>
<th><strong>Parcels 2011</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Weisswein</td>
<td>2000</td>
<td>2000</td>
</tr>
<tr>
<td>Rotwein</td>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>
Realisierungen

- Adjustment of historical data material to new structures
  small 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"

<table>
<thead>
<tr>
<th>Drink_ID</th>
<th>Product</th>
<th>Product Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dornfelder</td>
<td>Rotwein</td>
</tr>
<tr>
<td>2</td>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Drink_ID</th>
<th>Product</th>
<th>Product Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dornfelder</td>
<td>Rotwein</td>
</tr>
<tr>
<td>2</td>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
</tr>
<tr>
<td>3</td>
<td>Portugieser Weissherbst</td>
<td>Rotwein</td>
</tr>
<tr>
<td>3</td>
<td>Portugieser Weissherbst</td>
<td>Weisswein</td>
</tr>
</tbody>
</table>
"As is" & quasi "as of"-Scenario

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

<table>
<thead>
<tr>
<th>Drink_ID</th>
<th>Product</th>
<th>Product Group</th>
<th>Old_Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dornfelder</td>
<td>Rotwein</td>
<td>Rotwein</td>
</tr>
<tr>
<td>2</td>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
<td>Weisswein</td>
</tr>
<tr>
<td>3</td>
<td>Portugieser Weissherbst</td>
<td>Rotwein</td>
<td>Rotwein</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Drink_ID</th>
<th>Product</th>
<th>Product Group</th>
<th>Old_Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dornfelder</td>
<td>Rotwein</td>
<td>Rotwein</td>
</tr>
<tr>
<td>2</td>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
<td>Weisswein</td>
</tr>
<tr>
<td>3</td>
<td>Portugieser Weissherbst</td>
<td>Weisswein</td>
<td>Rotwein</td>
</tr>
</tbody>
</table>
"As is" & "as of"-Scenario: Versioning

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

<table>
<thead>
<tr>
<th>Drink_ID</th>
<th>Product</th>
<th>Product Group</th>
<th>Version</th>
<th>D_ID_old</th>
<th>active</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dornfelder</td>
<td>Rotwein</td>
<td>1</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
<td>1</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>Portugieser Weissherbst</td>
<td>Rotwein</td>
<td>1</td>
<td>3</td>
<td>X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Drink_ID</th>
<th>Product</th>
<th>Product Group</th>
<th>Version</th>
<th>D_ID_old</th>
<th>aktive</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dornfelder</td>
<td>Rotwein</td>
<td>1</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
<td>1</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>Portugieser Weissherbst</td>
<td>Rotwein</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Portugieser Weissherbst</td>
<td>Weisswein</td>
<td>2</td>
<td>6</td>
<td>X</td>
</tr>
</tbody>
</table>
"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

<table>
<thead>
<tr>
<th>G_ID</th>
<th>D_ID_old</th>
<th>Product</th>
<th>Product Group</th>
<th>Valid_From</th>
<th>Valid_Until</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Dornfelder</td>
<td>Rotwein</td>
<td>01.01.2010</td>
<td>31.12.9999</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
<td>01.01.2010</td>
<td>31.12.9999</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Portugieser Weissherbst</td>
<td>Rotwein</td>
<td>01.01.2010</td>
<td>31.12.9999</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>G_ID</th>
<th>D_ID_old</th>
<th>Product</th>
<th>Product Group</th>
<th>Valid_From</th>
<th>Valid_Until</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Dornfelder</td>
<td>Rotwein</td>
<td>01.01.2010</td>
<td>31.12.9999</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
<td>01.01.2010</td>
<td>31.12.9999</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Portugieser Weissherbst</td>
<td>Rotwein</td>
<td>01.01.2010</td>
<td>31.12.2010</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>Portugieser Weissherbst</td>
<td>Weisswein</td>
<td>01.01.2011</td>
<td>31.12.9999</td>
</tr>
</tbody>
</table>
"As posted"-Scenario

- Higher memory requirement
- History of the transaction date
- Artificial dimension key (or composite Key) is necessary
- linking in the facts with $D_{ID\_old}$

<table>
<thead>
<tr>
<th>Drink_ID</th>
<th>Product</th>
<th>Product Group</th>
<th>Version</th>
<th>D_ID_old</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dornfelder</td>
<td>Rotwein</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Portugieser Weissherbst</td>
<td>Rotwein</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Drink_ID</th>
<th>Product</th>
<th>Product Group</th>
<th>Version</th>
<th>D_ID_old</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dornfelder</td>
<td>Rotwein</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Portugieser Weissherbst</td>
<td>Rotwein</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Portugieser Weissherbst</td>
<td>Weisswein</td>
<td>2</td>
<td>6</td>
</tr>
</tbody>
</table>
"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

<table>
<thead>
<tr>
<th>Drink_ID</th>
<th>Product</th>
<th>Product Group</th>
<th>Load Date</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dornfelder</td>
<td>Rotwein</td>
<td>01.01.2010</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
<td>01.01.2010</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Portugieser Weissherbst</td>
<td>Rotwein</td>
<td>01.01.2010</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>Dornfelder</td>
<td>Rotwein</td>
<td>01.01.2011</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Müller-Thurgau</td>
<td>Weisswein</td>
<td>01.01.2011</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Portugieser Weissherbst</td>
<td>Weisswein</td>
<td>01.01.2011</td>
<td>1</td>
</tr>
</tbody>
</table>
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