

# Transformation mehrdimensionaler Datenmodelle

Dr. Michael Hahne

cundus AG

Prokurist, Niederlassungsleiter

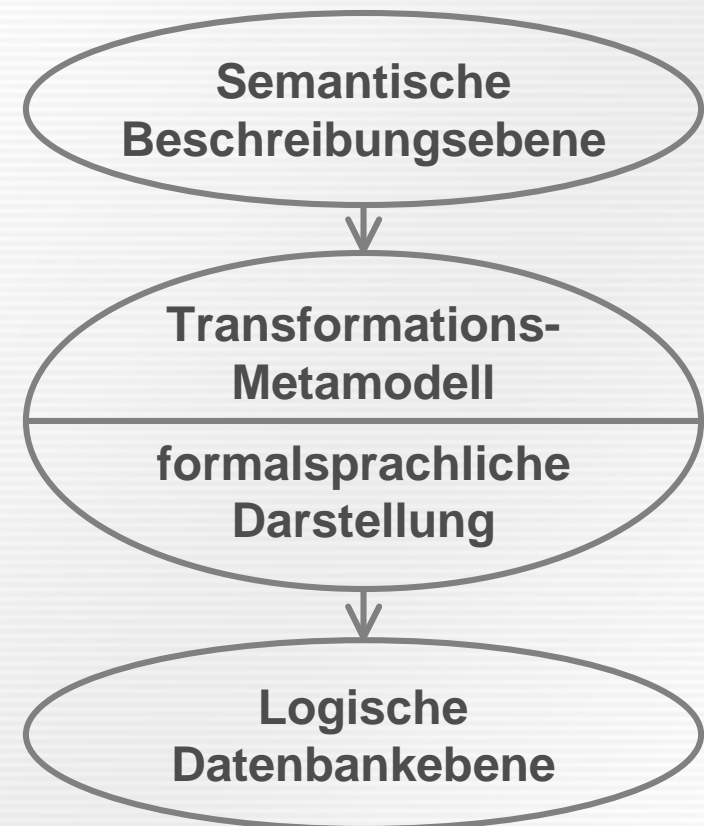
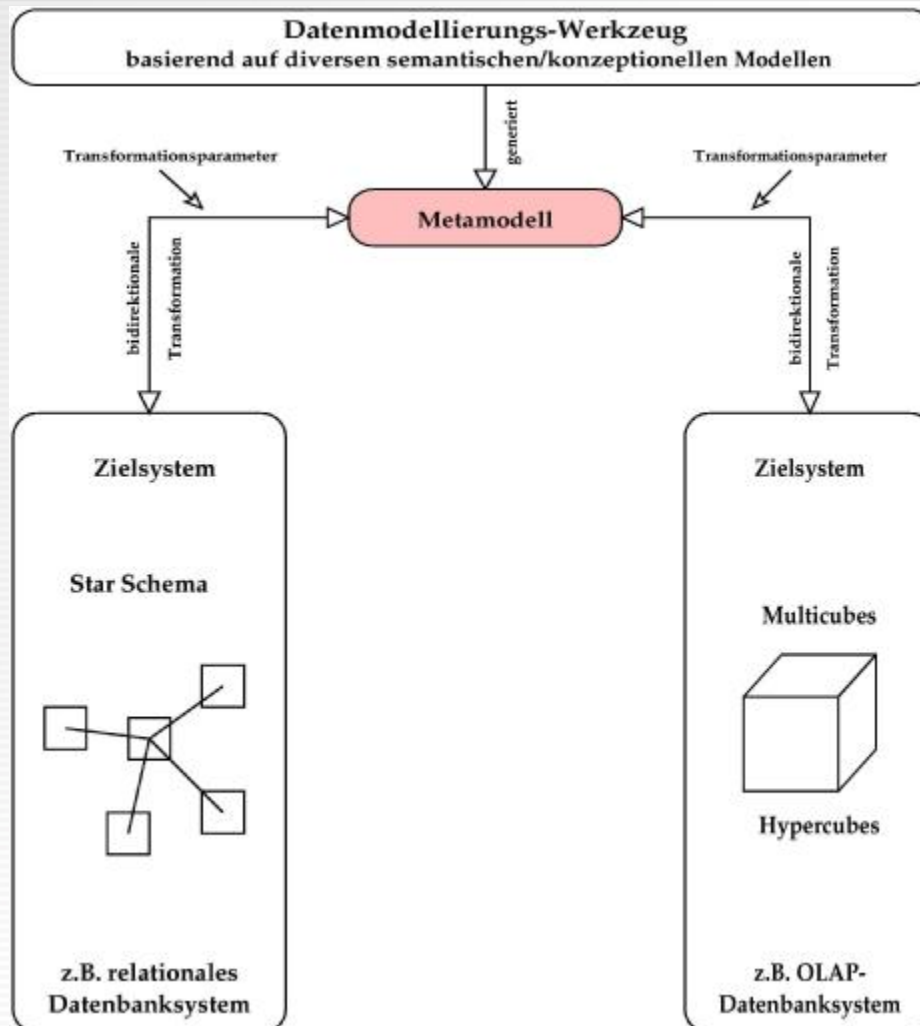
[michael.hahne@cundus.de](mailto:michael.hahne@cundus.de)

[www.cundus.de](http://www.cundus.de)

# Agenda

- 1 Modelltransformation und Datenbank-Generierung
- 2 Mehrdimensionales Metamodell
- 3 Formalsprachlicher Transformationsansatz
- 4 Transformation in Star Schemata
- 5 Transformationsmöglichkeiten im Überblick

# Modelltransformation und Datenbank-Generierung



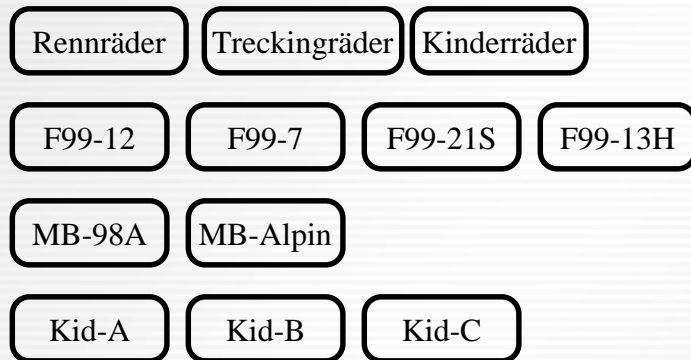


# Grundbestandteile des Metamodells (I)

## Einfaches Dimensions-Schema

$$EDS=(W)$$

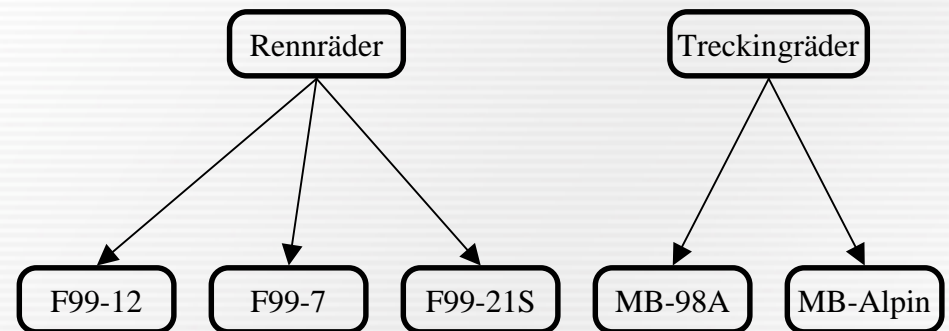
besteht aus einem Wertebereich W



## Hierarchie eines einfachen Dimensions-Schemas

$$\text{Graph } (V,E)$$

wobei  $V \subset W$  und  $(V,E)$  ist gerichtet, azyklisch und total geordnet



Dimensions-Schema



Dimension

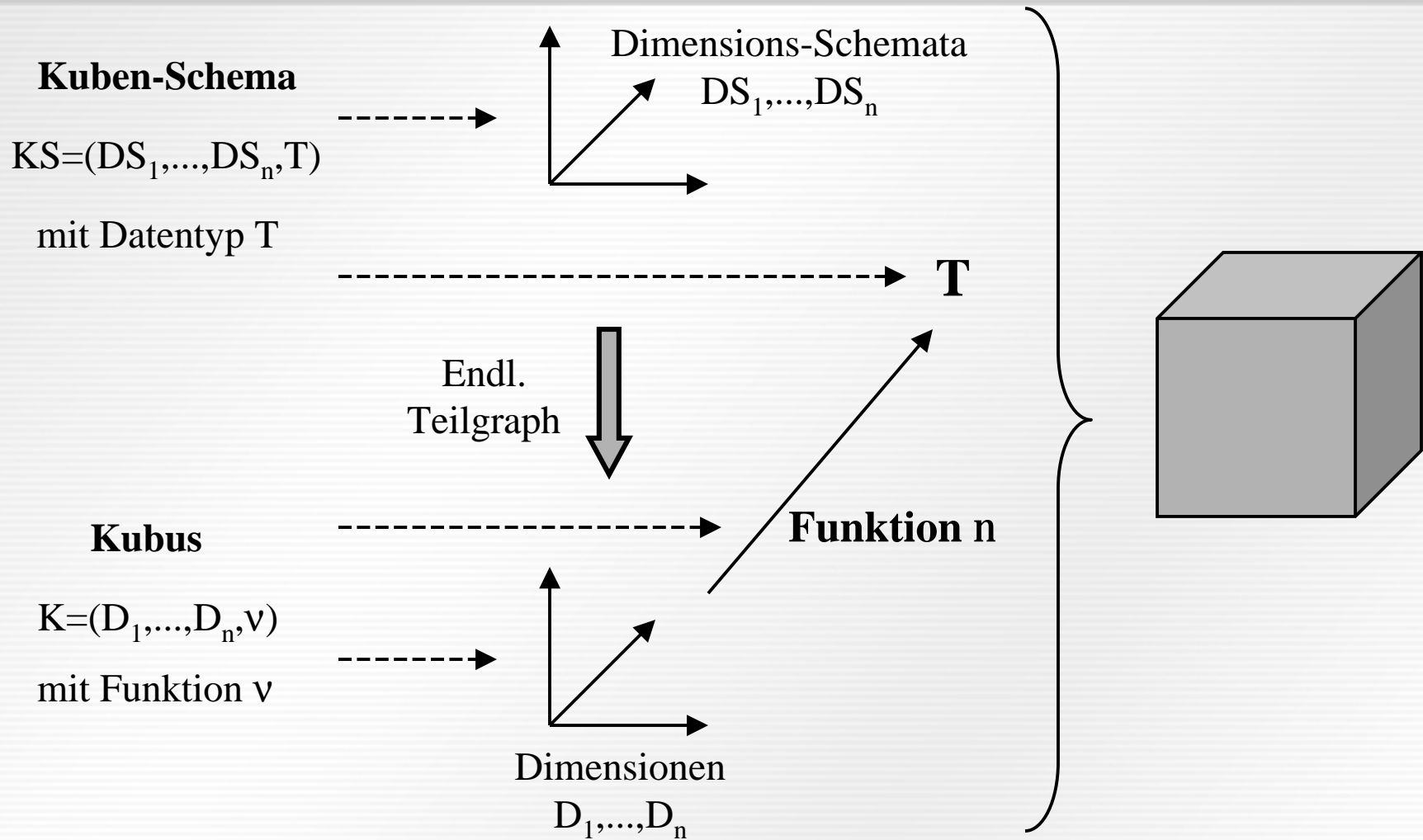
$$DS=(W,V,E)$$

$$D=(V_D,E_D)$$

endl. Teilgraph von  $(V,E)$



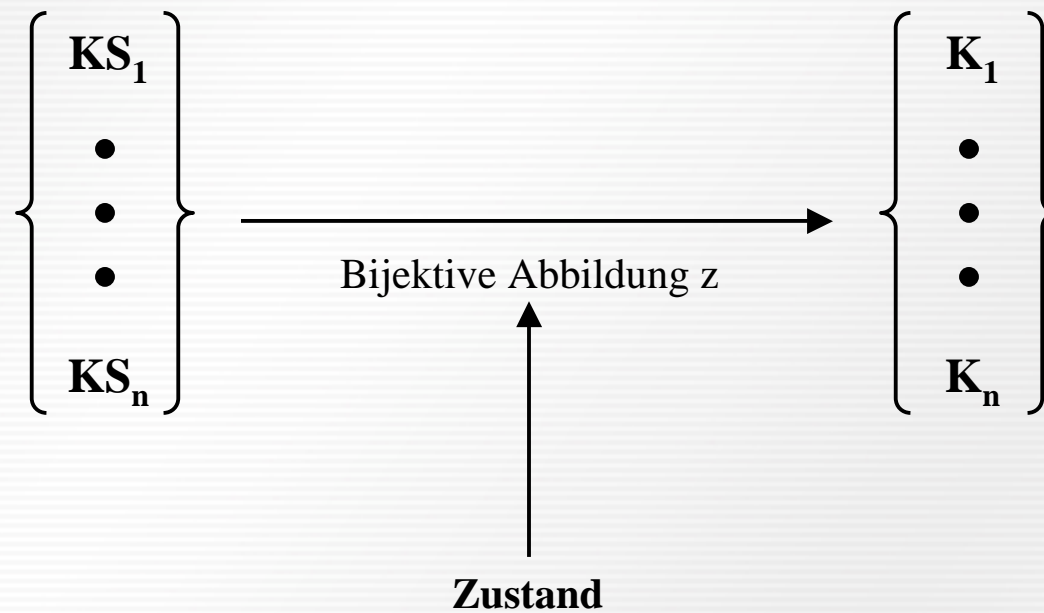
# Grundbestandteile des Metamodells (II)

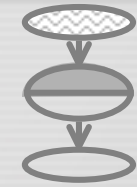




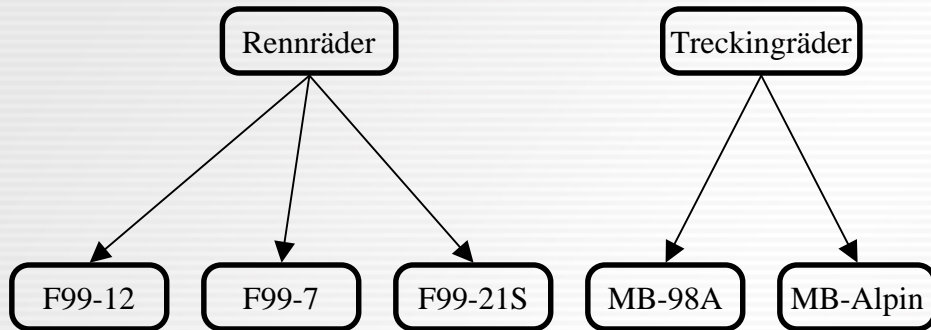
## Grundbestandteile des Metamodells (III)

Mehrdimensionales Datenbank-Schema  
(MDB-Schema)





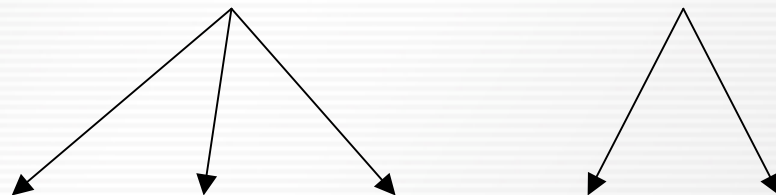
# Grundbestandteile (VI) : Markierungen im Metamodell



Rennräder

Treckingräder

F99-12   F99-7   F99-21S   MB-98A   MB-Alpin

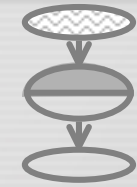


## Knotenmarkierungen

$$\xrightarrow{L_V} m \hat{I} \{ M_V \}$$

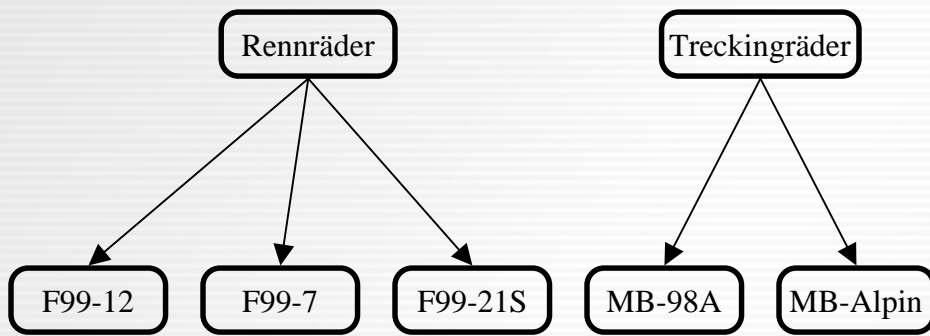
## Kantenmarkierungen

$$\xrightarrow{L_E} m \hat{I} \{ M_E \}$$



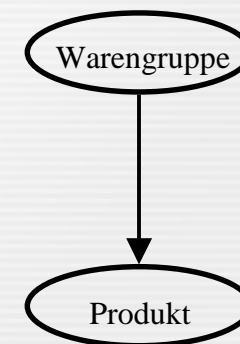
# Typisierung und Partitionierung

**Dimensions-Schema**



**(assoziierter) Dimensions-Typ**

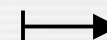
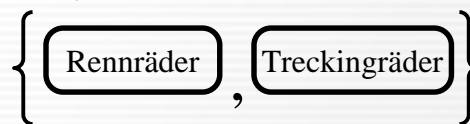
„Typisierung“



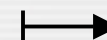
Endl. Überdeckung  
der Knotenmenge

„Partitionierung“

**Knotenpartition**



**Warengruppe**

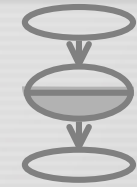


**Produkt**



**Induzierter Graph**





## Beispielskript des Metamodells

DATABASES:

    DATABASE: [Marketing]

        DATATYPES: integer,boolean,char,double

CUBES:

    CUBE: [Umsatz]

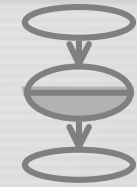
        DATATYPE: float

        REFERENCED DIMENSIONS: [Zeit],[Vertriebsweg],[Produkt]



## Beispielskript des Metamodells

```
DIMENSION: [Vertriebsweg]
DATATYPE: char
NODES: 'Partner',
       'Katalog',
       'E-Shop'
EDGES: -
LABELS: -
```



## Beispielskript des Metamodells

```
DIMENSIONS:  
  DIMENSION: [Zeit]  
  DATATYPE: char  
  NODES: '1999',  
         '2000',  
         '2001',  
         'Alle Jahre',  
  EDGES: ('Alle Jahre', '1999'),  
         ('Alle Jahre', '2000'),  
         ('Alle Jahre', '2001'),  
  LABELS: -
```



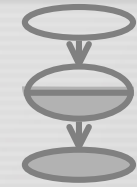
## Beispielskript des Metamodells

```
DIMENSION: [Produkt]
DATATYPE: char
NODES: 'F99-12',
       'F99-7',
       'F99-21S',
       'F99-13H',
EDGES: ('Rennräder', 'F99-12'),
       ('Rennräder', 'F99-7'),
       ('Rennräder', 'F99-21S'),
       ('Rennräder', 'F99-13H'),
LABELS:
  LABEL: [Partition]
  LABELVALUES NODES: 'F99-12': 'Produkt',
                    'F99-7': 'Produkt',
                    'F99-21S': 'Produkt',
                    'F99-13H': 'Produkt',
                    'Rennräder': 'Warengruppe'
```

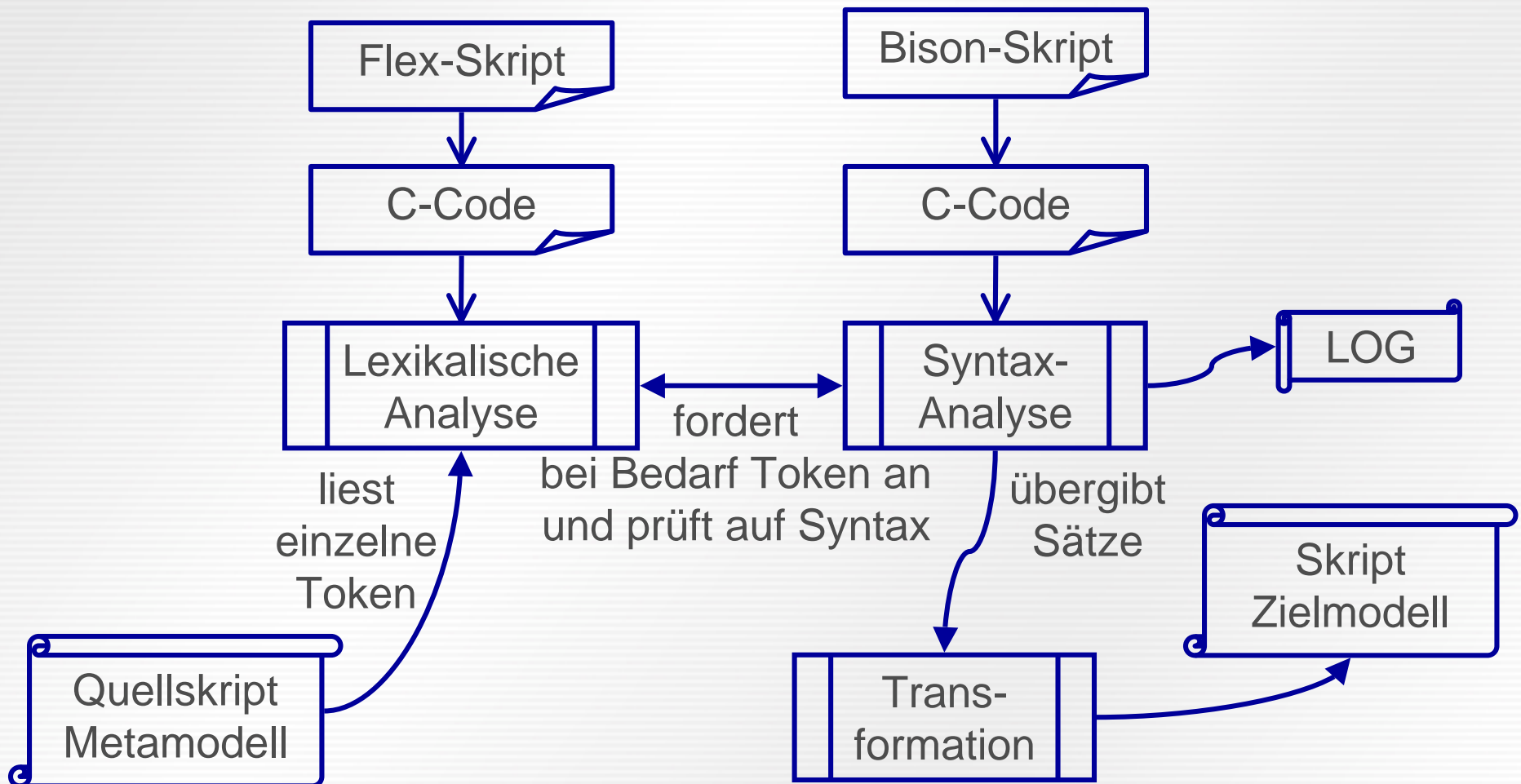


## Phasen der Sprachanalyse

1. In der Phase der **lexikalischen Analyse** wird der Quelltext analysiert, d.h. es wird überprüft, ob er ein Wort über dem Ausgangsalphabet ist, und wird dabei in elementare Ausdrücke (sog. Tokens) transformiert, damit eine spätere Weiterverarbeitung leichter möglich ist.
2. Die Phase der **syntaktischen Analyse** überprüft das von der lexikalischen Analyse als Wort über dem Ausgangsalphabet erkannte Wort auf die syntaktische Korrektheit.
3. In der Phase der **semantischen Analyse** werden die als syntaktisch korrekt erkannten Worte auf ihre semantische Stimmigkeit hin überprüft.



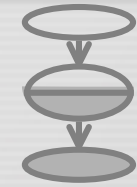
# Ablaufschema



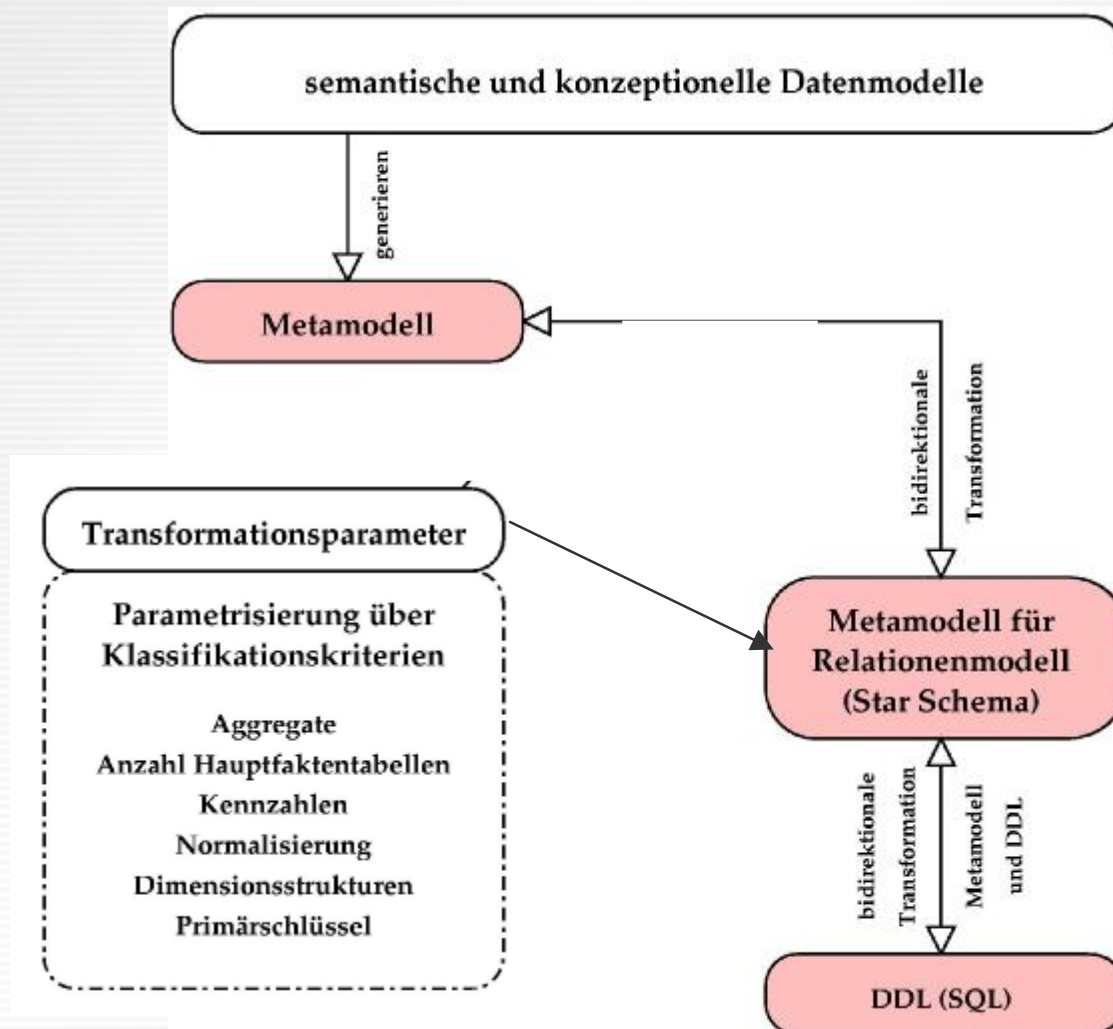
# Eigenschaften zur Klassifizierung von Star Schema-Strukturen



- **Aggregate**  
dynamisch, in Hauptfaktentabelle, Aggregattabellen
- **Anzahl Hauptfaktentabellen**  
eine, mehrere
- **Kennzahlen**  
Kennzahlendimension, Faktentabelle, keine, Mischform
- **Normalisierung**  
flache Dimensionstabelle, partitioniert, normalisiert
- **Dimensionsstrukturen**  
klassische Strukturen, Heterarchie, rekursive Beziehungen, Minidimension
- **Primärschlüssel**  
einattributig/mehrattributig, künstlich/natürlich



# Transformation in Star Schemata







## Transformationsparameter I

- **Aggregate**  
dynamisch
- **Anzahl Hauptfaktentabellen**  
eine : FTMarketingUmsatz  
[FT<database><cube>]
- **Kennzahlen**  
Faktentabelle  
<cube> wird zu Spalte in der Faktentabelle



## Transformationsparameter II

- **Normalisierung**  
*DTZeit, DTProdukt, DTVertriebsweg*: jeweils flache Dimensionstabelle
- **Dimensionsstrukturen**  
*DTVertriebsweg*: flach,  
    <level 0> --> Spalte „Vertriebsweg“  
*DTZeit*: balancierte Waldstruktur,  
    <level 0> --> Spalte Monat, <level 1> --> Spalte Jahr  
*DTProdukt*: balancierte Waldstruktur,  
    label „Partition“ --> Spalten in Dimensionstabelle
- **Primärschlüssel**  
*DTZeit, DTProdukt, DTVertriebsweg*: jeweils einattributig & künstlich



## Transformationsergebnis I

```
DIMENSION: [Vertriebsweg]
DATATYPE: char
NODES: 'Partner',
       'Katalog',
       'E-Shop'
EDGES: -
LABELS: -
```



```
CREATE TABLE DTVertriebsweg (
    Vertriebsweg_Id      int NOT NULL,
    Vertriebsweg         nvarchar(100) NULL,
    PRIMARY KEY (Vertriebsweg_Id) )
```

## Transformationsergebnis II



```
CREATE TABLE DTZeit (  
    Zeit_Id                int NOT NULL,  
    Monat                  nvarchar(10) NULL,  
    Jahr                   nchar(4) NULL,  
    PRIMARY KEY (Zeit_Id) )
```

DIMENSIONS:

DIMENSION: [Zeit]

DATATYPE: char

NODES: '1999',

'2000',

'2001',

'Alle Jahre',

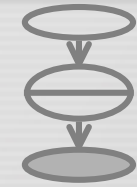
EDGES: ('Alle Jahre', '1999'),

('Alle Jahre', '2000'),

('Alle Jahre', '2001'),

LABELS: -





## Transformationsergebnis III

```

CREATE TABLE DTProdukt (
    Produkt_Id          int NOT NULL,
    Produkt             nvarchar(100) NULL,
    Warengruppe        nvarchar(100) NULL,
    PRIMARY KEY (Produkt_Id) )
DIMENSION: [Produkt]
  DATATYPE: char
  NODES: 'F99-12',
        'F99-7',
        'F99-21S',
        'F99-13H',
  EDGES: ('Rennräder', 'F99-12'),
        ('Rennräder', 'F99-7'),
        ('Rennräder', 'F99-21S'),
        ('Rennräder', 'F99-13H'),
  LABELS:
    LABEL: [Partition]
    LABELVALUES NODES: 'F99-12': 'Produkt',
                      'F99-7': 'Produkt',
                      'F99-21S': 'Produkt',
                      'F99-13H': 'Produkt',
                      'Rennräder': 'Warengruppe'

```



## Transformationsergebnis IV

DATABASES:

DATABASE: [Marketing]


DATATYPES: integer,boolean,char,double

CUBES:

CUBE: [Umsatz]

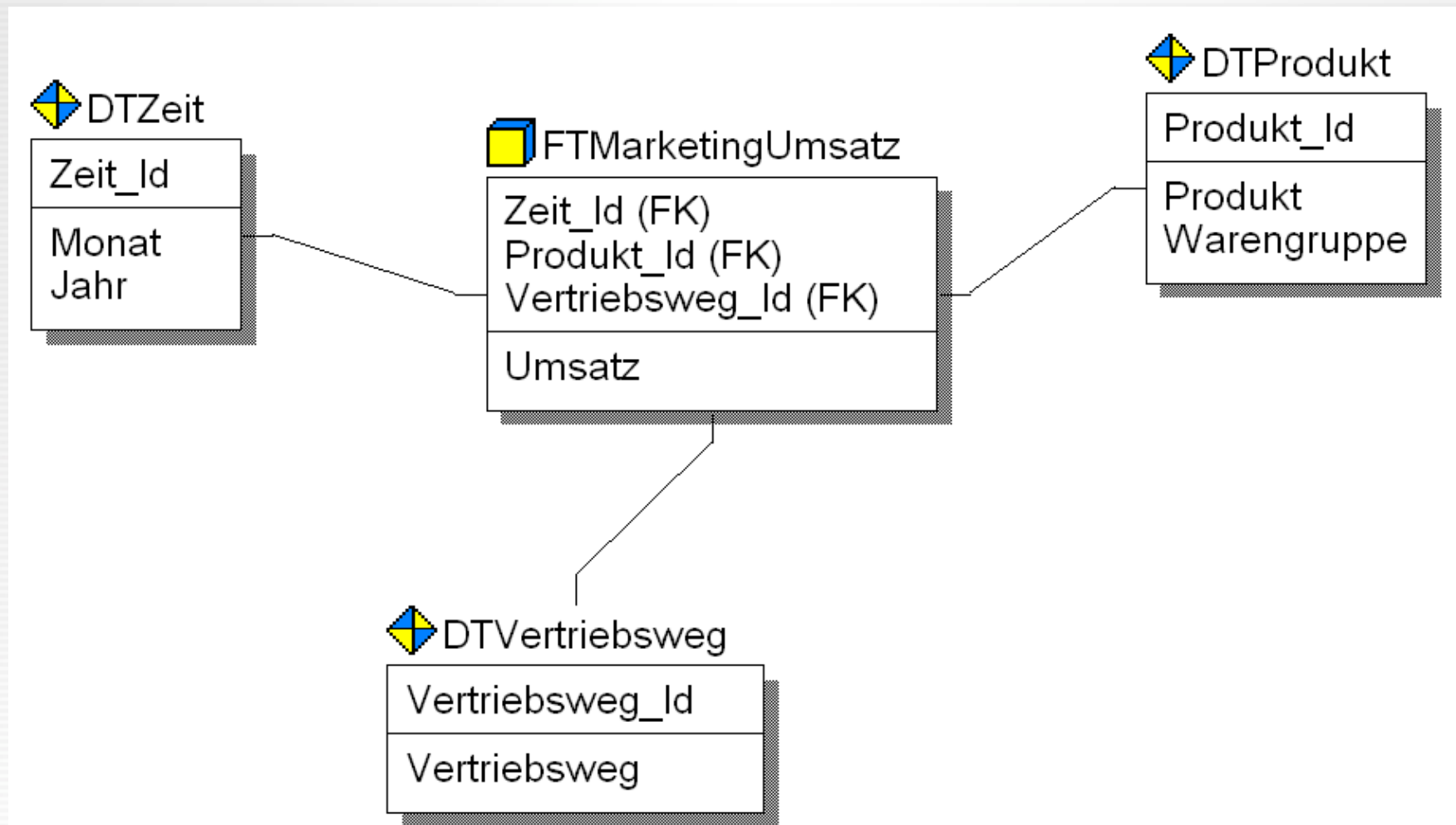
DATATYPE: float

REFERENCED DIMENSIONS: [Zeit],[Vertriebsweg],[Produkt]



```
CREATE TABLE FTMarketingUmsatz (  
    Zeit_Id                int NOT NULL,  
    Produkt_Id             int NOT NULL,  
    Vertriebsweg_Id        int NOT NULL,  
    Umsatz                  money NULL,  
    PRIMARY KEY (Zeit_Id, Produkt_Id, Vertriebsweg_Id),  
    FOREIGN KEY (Vertriebsweg_Id) REFERENCES DTVertriebsweg,  
    FOREIGN KEY (Produkt_Id) REFERENCES DTProdukt,  
    FOREIGN KEY (Zeit_Id) REFERENCES DTZeit))
```

## Transformationsergebnis in ER-Darstellung



# Transformationsmöglichkeiten im Überblick

