

Beispiel einer Anfrage (I)

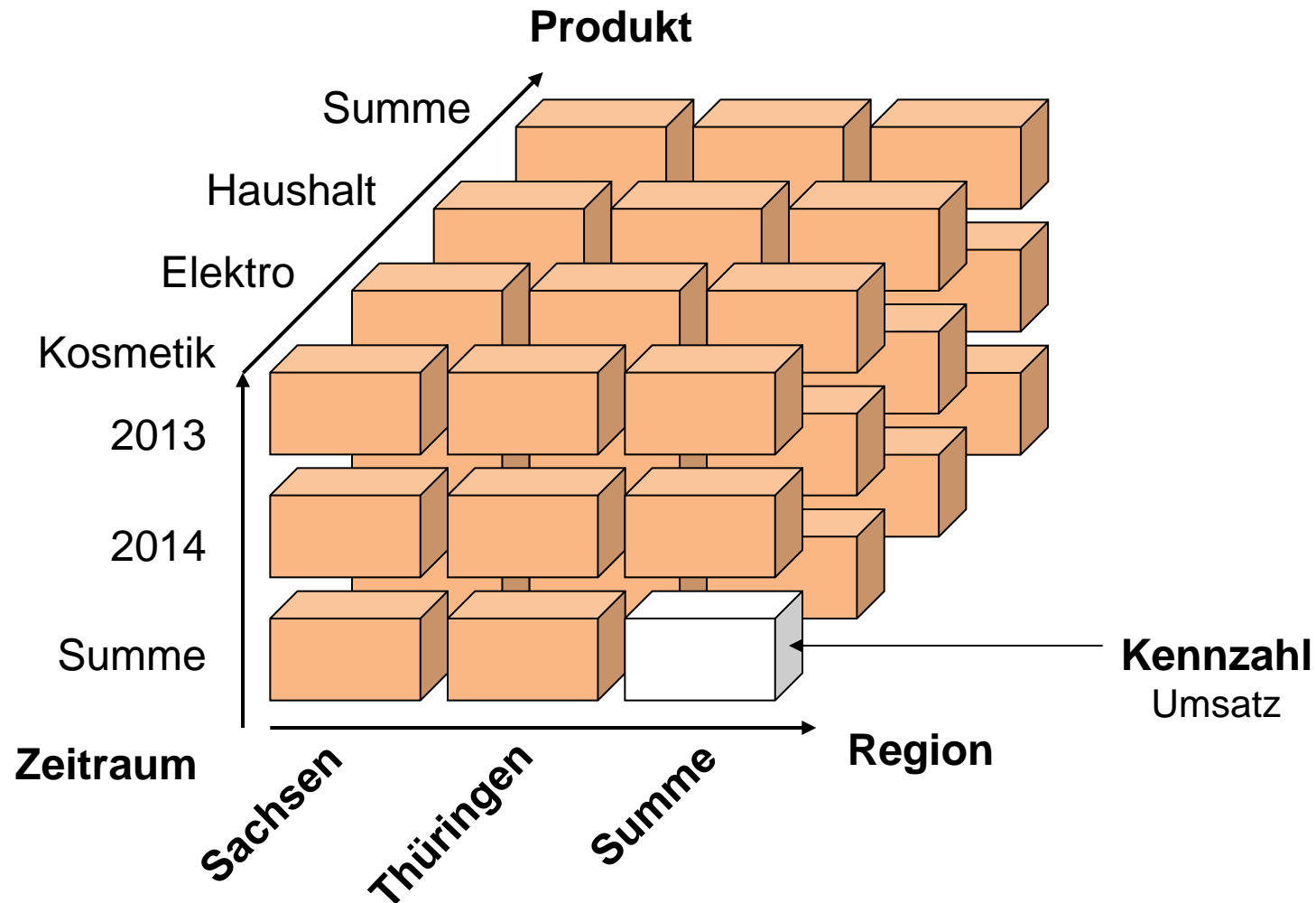
Welche Umsätze sind in den Jahren 2013 und 2014 in den Abteilungen Kosmetik, Elektro und Haushalt in den Bundesländern Sachsen und Thüringen angefallen?

Ergebnis als Bericht

Umsatz		Kosmetik	Elektro	Haushalt	SUMME
2013	Sachsen	45	123	17	185
	Thüringen	43	131	21	195
	SUMME	88	254	38	380
2014	Sachsen	47	131	19	197
	Thüringen	40	136	20	196
	SUMME	87	267	39	393
SUMME		175	521	77	773

Beispiel einer Anfrage (II)

Ergebnis als Würfel



Data Warehouse – Ziele und Definitionen

allgemeine Zielsetzungen:

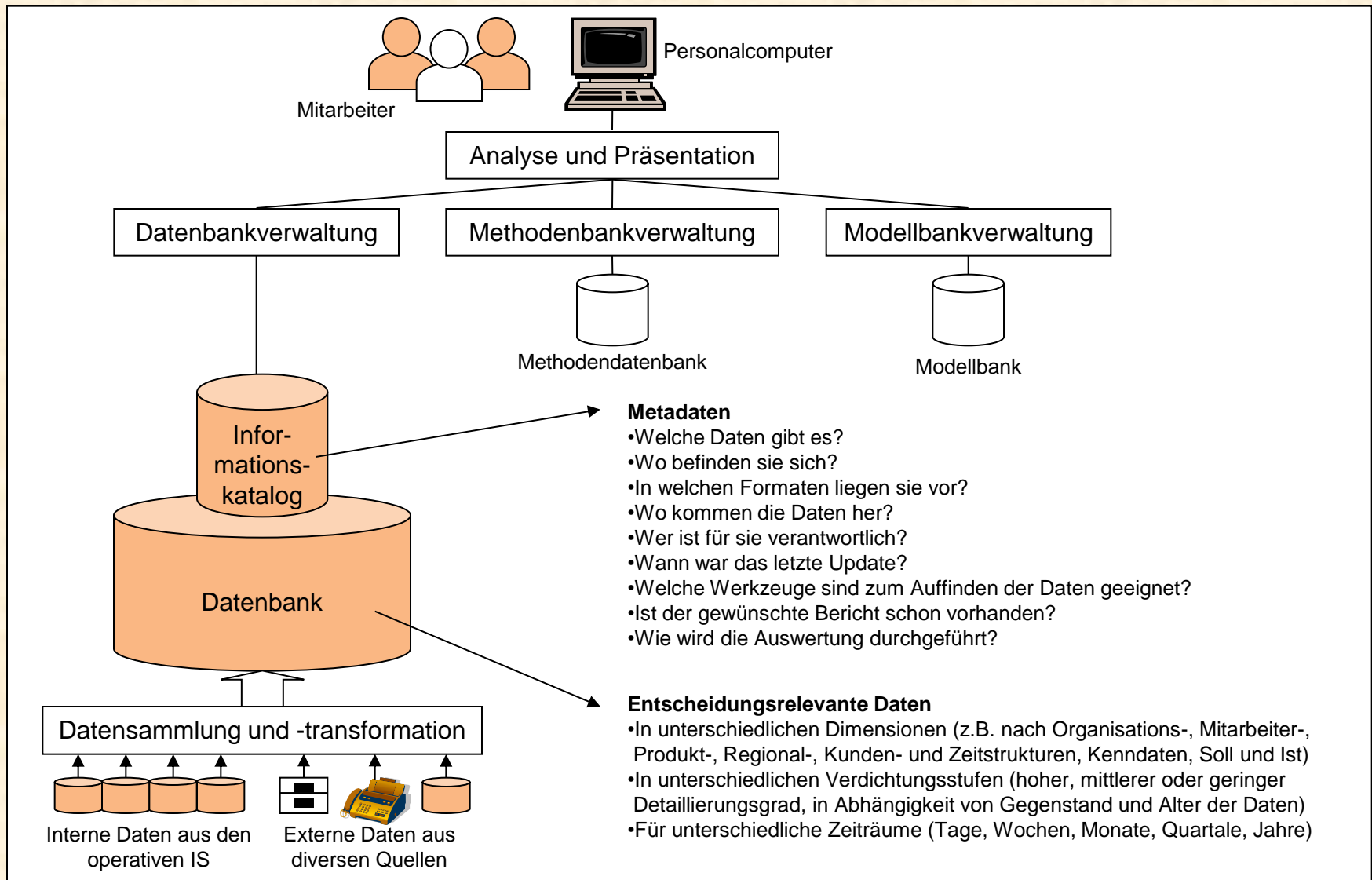
- Aufbau einer zentralen und konsistenten Datenbasis
- für verschiedene Anwendungen
- zur Unterstützung analytischer Aufgaben von Fach- und Führungskräften
- losgelöst betrieben von den operativen Datenbanken

Ein **Data Warehouse** ist eine aus einer oder mehreren operativen Datenbanken extrahierte Datenbank, die alle für den Geschäftsprozess relevanten Daten eines Unternehmens zusammenfasst, aufbereitet und aggregiert.

Ein **Data Warehouse** umfasst die Meta-, die Dimensions- und die Aggregationsdaten sowie die Fakten, um Informationen verfügbar und informationsgestützte Entscheidungen möglich zu machen.

Darüber hinaus beinhaltet es die notwendigen Verwaltungsprozesse wie Einfügen, Warehouse und Abfrage.

Beispiel für eine Data Warehouse - Architektur

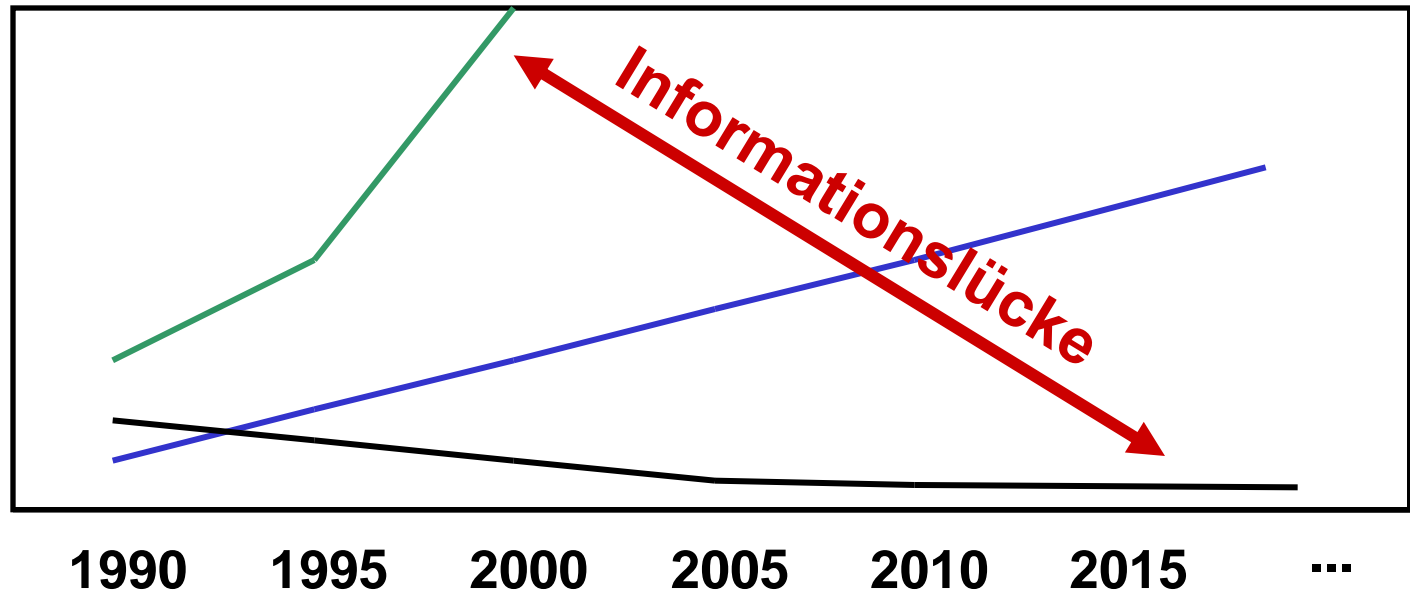


Problemfelder bei der „Erschließung“ von Informationen

Hindernisse bei der Auswertung unternehmensweiter Datenbestände ohne Einsatz Data Warehouse - Einsatz:

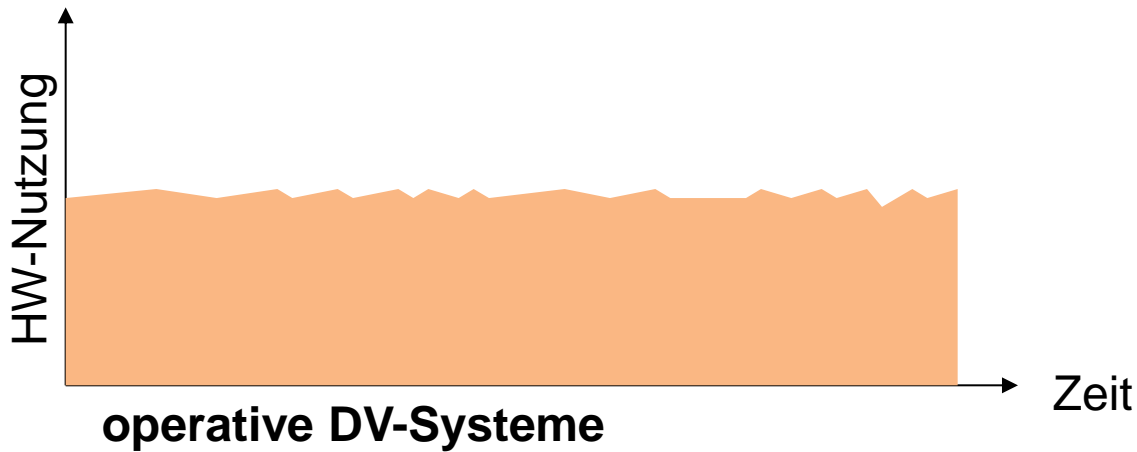
- Behinderung von Zugriff auf und Zusammenführung von Daten durch **heterogene IS-Strukturen**
- unbefriedigende und oft schwer überprüfbare **Qualität und Aktualität der Daten**
- Behinderung der Berichtserstellung durch **mangelnde Kapazität in der EDV.**
- **Auslagerung historischer Daten** auf langsame, nicht online verfügbare Datenspeicher
- Erhebliche **Mehrbelastung der vorhandenen IS-Infrastruktur** (Rechner, Netzwerke, etc...) durch die notwendigen Analysen
- **Fehlende Nachvollziehbarkeit von Analysen** wegen permanenter Datenänderungen
- **unbefriedigende Performance** beim Umgang mit großen Datenbeständen

Informationslücke



- Verfügbare Datenbestände
- Unternehmenskritische Entscheidungen pro Woche
- Anzahl der Systemanalytiker

Vergleich der Systemauslastung



Data Warehouse – Definition und Charkteristika

Forderungen an ein DWH: *[nach Inmon]*

- **Struktur- und Formatvereinheitlichung (Integration):**
Ablage der Daten einer Datenstruktur mit einheitlichem Format
- **Subjektorientierung:**
Speicherung orientiert an den Subjekten eines Unternehmens
- **Zeitraumbezug** (time variant):
Speicherung aller Daten mit Zeitraumbezug (nicht zeitpunkt-bezogen)
- **Nicht-Volatilität:**
keine Änderung einmal gespeicherter Daten

Denormalisierung

Auf eine Normalisierung nach Codd wird verzichtet, bzw. ist eine Normalisierung vorhanden, wird diese rückgängig gemacht.

Ziel: Reduzierung der Zugriffszeiten und damit Gewinn an Performance

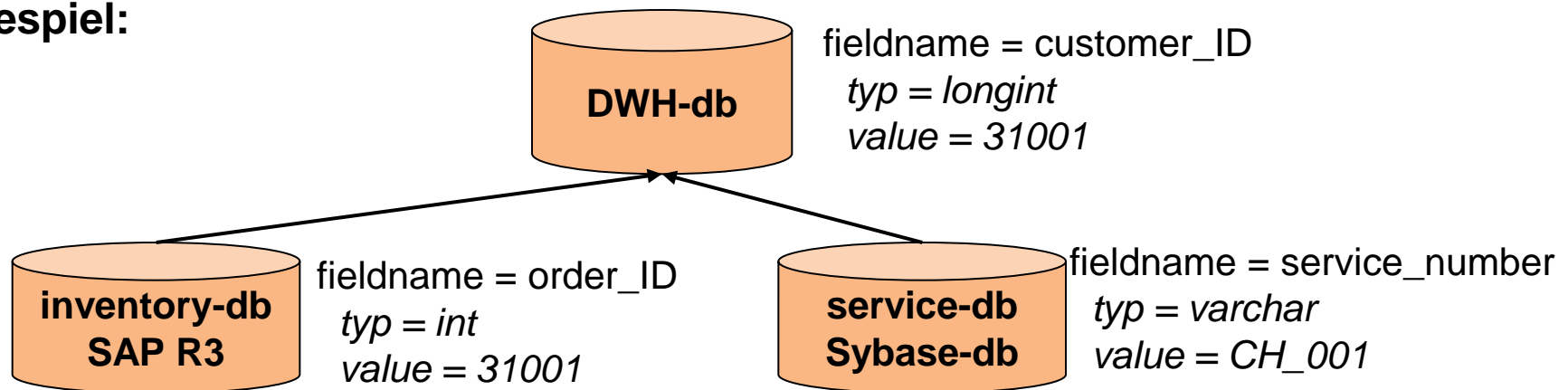
Integration

- Integration ist erforderlich, da das Data Warehouse die zentrale Datenbasis zur Durchführung der Business Analyse ist.
- Diese Datenbasis enthält die Daten aller OLTP-Systeme und aller Organisationseinheiten des Unternehmens in einem einheitlichen Format und einer Struktur.

Situation:

- o Einsatz unterschiedlicher DBMS auf verschiedenen Betriebssystemplattformen
- o Unterschiede in Struktur und Formaten der OLTP-Datenbanken
- o Unterschiede in der semantischen Interpretation der Daten

Bespiel:



Subjektorientierung

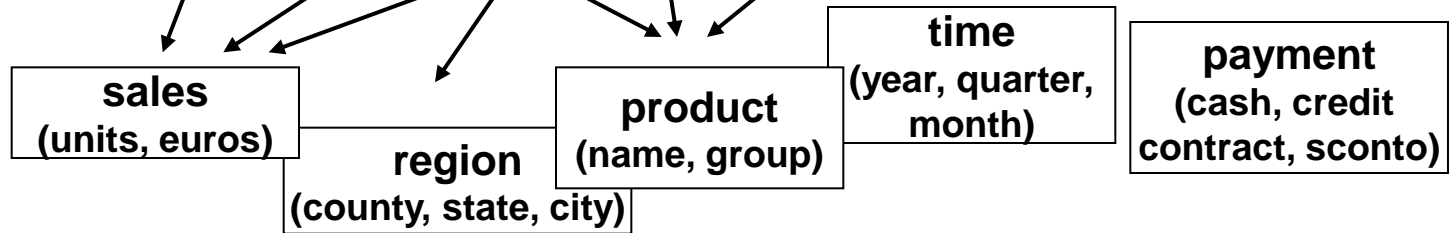
- Das Datenbankschema ist an den Subjekten der Business Analyse und nicht an den Datenbankobjekten (OLTP Datenbank) orientiert.
- Die Orientierung an den Subjekten der Business Analyse ist sinnvoll, da unterschiedliche Unternehmensfunktionen gleiche Subjekte des Data Warehouses für ihre Analyse nutzen.

Bespiel:

***company
functions:***



subjects:



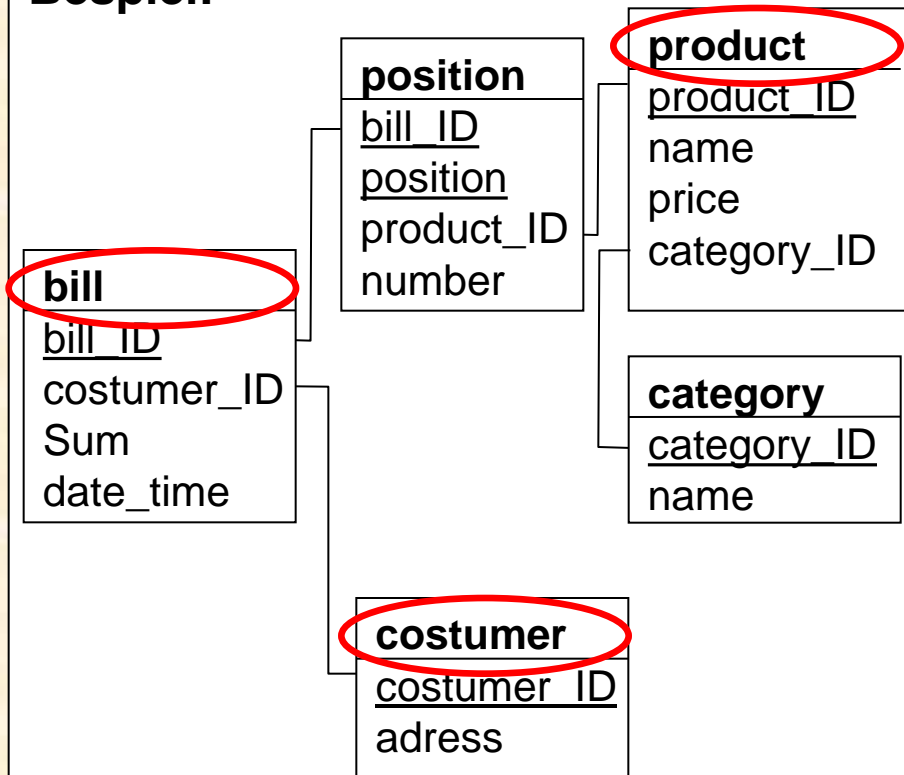
Daten-orientierte versus Subjekt-orientierte Modellierung

OLTP Datenbank



Daten-orientierte Modellierung

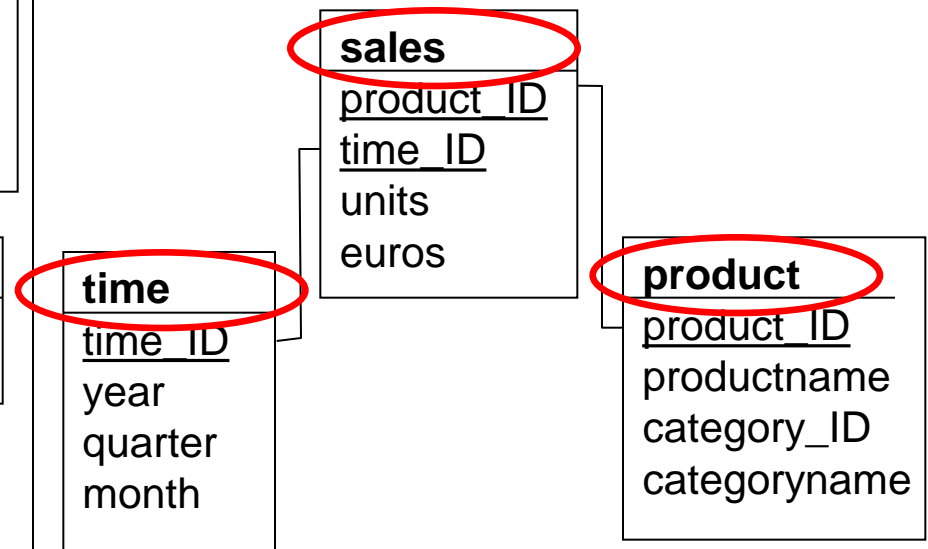
Bespiel:



DWH Datenbank



Subjekt-orientierte Modellierung



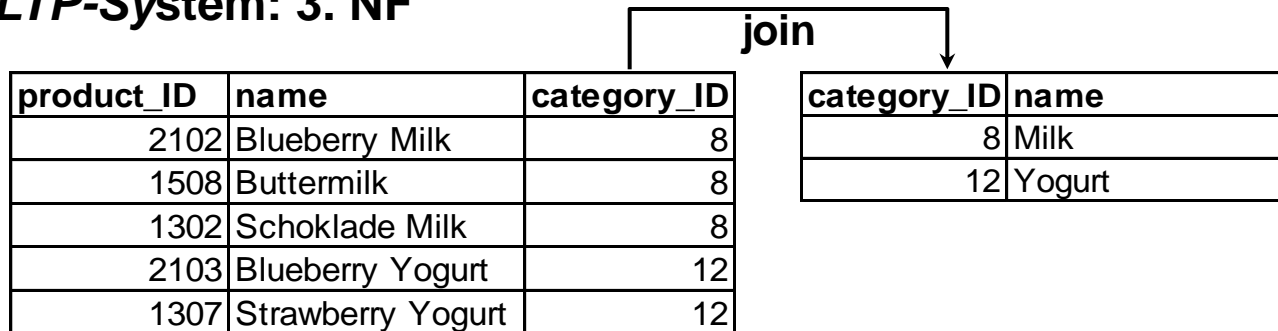
Denormalisierung

- keine Normalisierung nach Codd in die 3. Normalform (NF)
- Entfernen vorhandener Normalisierungen der OLTP-Datenbasis

Ziel: Reduzierung der Zugriffszeit => Steigerung der Performance

Beispiel:

OLTP-System: 3. NF



DWH-System: not 3. NF

product_ID	productname	category_ID	categoryname
2102	Blueberry Milk	8	Milk
1508	Buttermilk	8	Milk
1302	Schokolade Milk	8	Milk
2103	Blueberry Yogurt	12	Yogurt
1307	Strawberry Yogurt	12	Yogurt

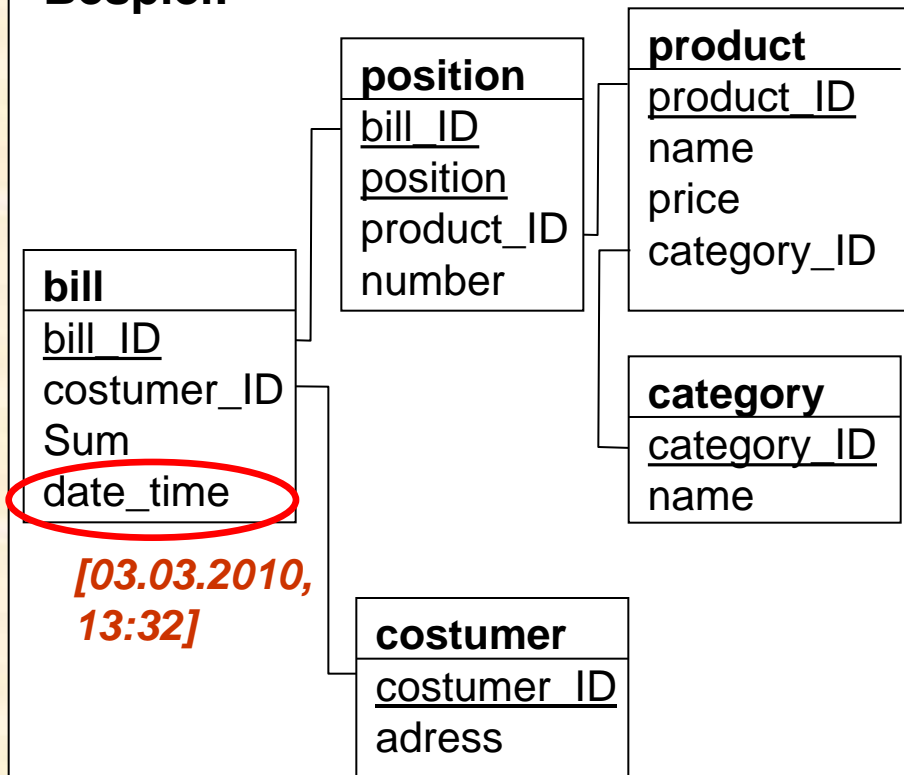
**kein JOIN in Abfrage
notwenig**

Modellierung der Zeit

OLTP Datenbank

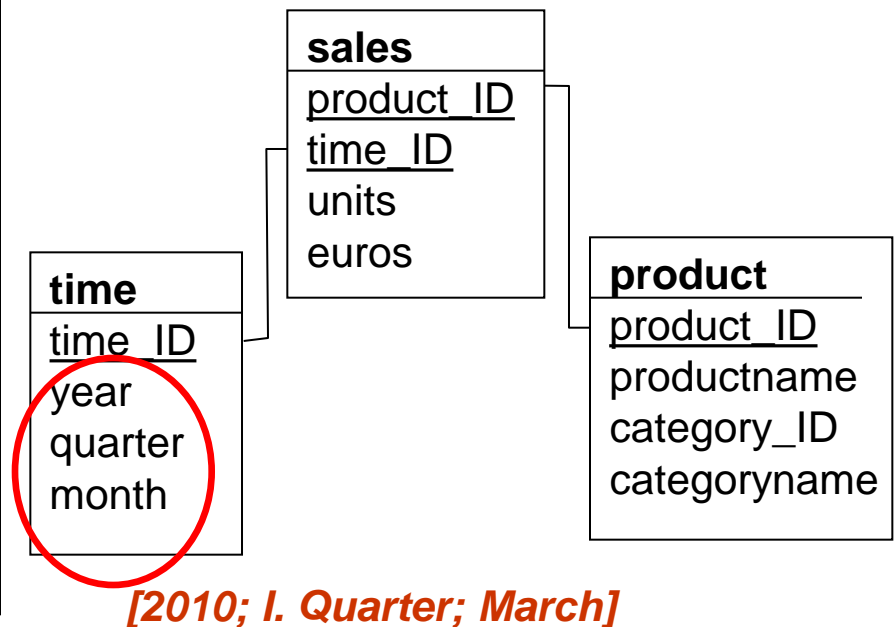
Bezug zu Zeitpunkten

Bespiel:



DWH Datenbank

Bezug zu Zeiträumen



Veranschaulichung der Multidimensionalität

➤ Dimensionalität

Anzahl der Dimensionen

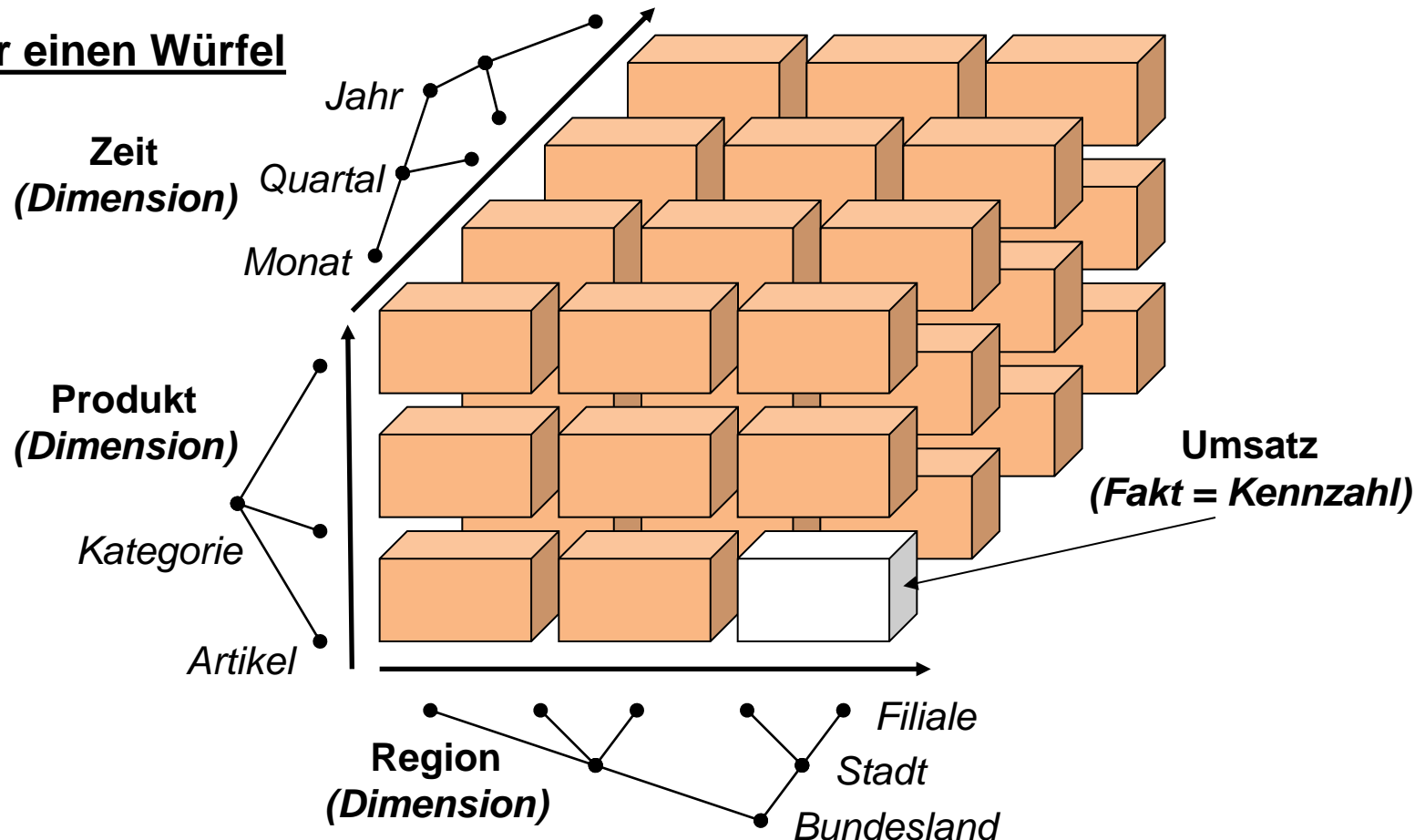
➤ Kanten = Dimensionen

➤ Zellen = Fakten

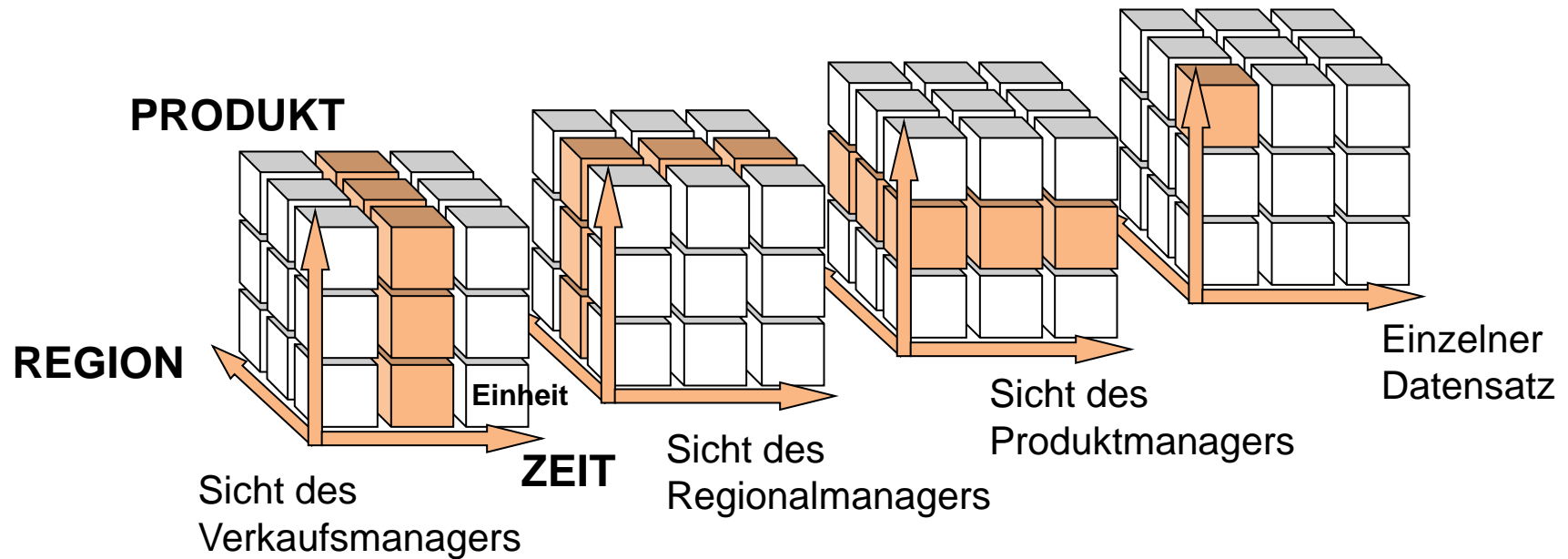
➤ Visualisierung

- 2 Dimensionen: Tabelle
- 3 Dimensionen: Würfel
- >3 Dimensionen: Multidim. Domänenstruktur

Bsp. für einen Würfel



Verschiedene Sichten auf einen Würfel



Dimensionen

- Dimensionen sind die „Richtungen“, nach denen die Kennzahlen ausgewertet werden können.
- Sie beschreiben den Rahmen für die Auswertung der Kennzahlen.
- Kennzahlen sind nur in Zusammenhang mit den Dimensionen gültig.
- Beispiel: Wenn der Umsatzbetrag eines Produktes in einer Region und einer Zeitperiode die Kennzahl ist, dann sind die Dimensionen:
 - Produkt,
 - Geografie und
 - Zeit.
- Dimensionen sind eine unabhängige Liste an Analyseelementen.
- Die Attribute sind i. A. eine abhängige Liste an Analyseelementen.
- Beispiel: Attribute in der Dimension Zeit
 - alle Jahre, Jahr, Quartal, Monat, Woche, Tag

Attribute und Hierarchien der Dimensionen

- Attribute sind die Eigenschaften der Dimensionen.
- Sie werden für die Klassifizierung und Filterung der Kennzahlen verwendet.
- Beispiel: Produkt, Geografie und Zeit sind die Dimensionen eines in einem Würfel (Cube).
Attribute:
 - *Dim. Produkt:* Produkt_ID, Produktname, Preis, Produkttyp, Anbieter
 - *Dim. Geografie:* Shop_ID, Shopname, Stadt, Region, Land
 - *Dim. Zeit:* Tag_ID, Wochentag, Woche, Monat, Quartal, Jahr
- Die Attribute können zur Bildung der Hierarchien in der Dimension verwendet werden. Die verwendeten Attribute bilden dann die Ebenen (level) der Hierarchie.
- Beispiel: *Dim. Zeit - Attribute zur Bildung der Hierarchie Kalender:*
alle Jahre ⇒ Jahr ⇒ Quartal ⇒ Monat ⇒ Tag

Dimensionen und Abhängigkeit

2 unabhängige Dimensionen

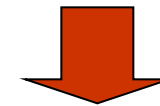
	January	February	March
Blueberry Milk	102456	98345	122334
Buttermilk	13453	15677	17885
Schokolade Milk	22341	18776	24556
Blueberry Yogurt	67987	56367	54378
Strawberry Yogurt	123451	98233	99132

	Qtr1	Qtr2
Blueberry Milk	323135	421823
Buttermilk	47015	76553
Schokolade Milk	65673	76823
Blueberry Yogurt	178732	198323
Strawberry Yogurt	320816	44571

2 Dimensionen mit Abhängigkeit

	Qtr1	Qtr2
January	102456	
February	98345	
March	122334	
April		145267
May		138734
June		156239

viele leere Zellen (z.B. Wert für Qtr1 und April nicht existent)



Erstellen einer Dimension, die die beiden Elemente als Attribute enthält (ggf. Hierarchie aus den Attributen erzeugen)

Beispiel einer parallelen Hierarchie

parallel

- Es gibt Knoten mit mehr als einem übergeordneten Knoten.
- Es sind mehrere Wege von der Wurzel zu einzelnen Blättern vorhanden.
- Es gibt keine Lücken in den Werten der Knoten.

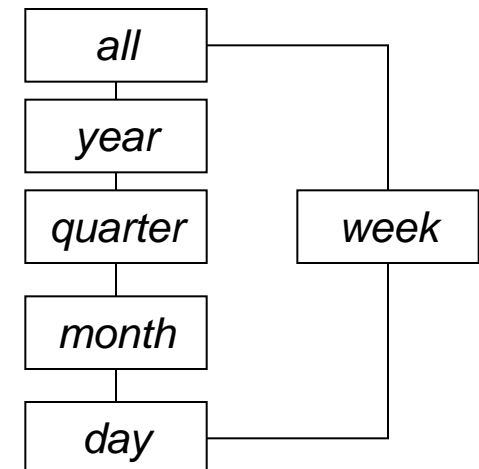
Beispiel:

	53. week 2004	1. week 2005	2. week 2005
Mo	27.12.2004	03.01.2005	10.01.2005
Tu	28.12.2004	04.01.2005	11.01.2005
We	29.12.2004	05.01.2005	12.01.2005
Th	30.12.2004	06.01.2005	13.01.2005
Fr	31.12.2004	07.01.2005	14.01.2005
Sa	01.01.2005	08.01.2005	15.01.2005
So	02.01.2005	09.01.2005	16.01.2005

name:

time

level:



Strategien zur Speicherung multidimensionaler Daten

relational

- Datenspeicherung im relationalen DBMS
- Modellierung der Cubestruktur mittels Relationen (Tabellen)
- Verwendung des SQL-Standards zur Datenabfrage und -manipulation
- Beispiele:
 - Star Schema
 - Snowflake Schema

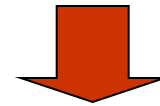


ROLAP

(**R**elational **O**nline **A**nalytical **P**rocessing)

nicht relational

- Datenspeicherung nicht in relationaler Art
- Speicherung des Cube mit klassischen Methoden der Informatik
- Fehlende Standards für Datenabfrage und -manipulation
- Beispiele:
 - arrays
 - hash-tables
 - bitmap-indices



MOLAP

(**M**ultidimensional **O**nline **A**nalytical **P**rocessing)

Star - Schema

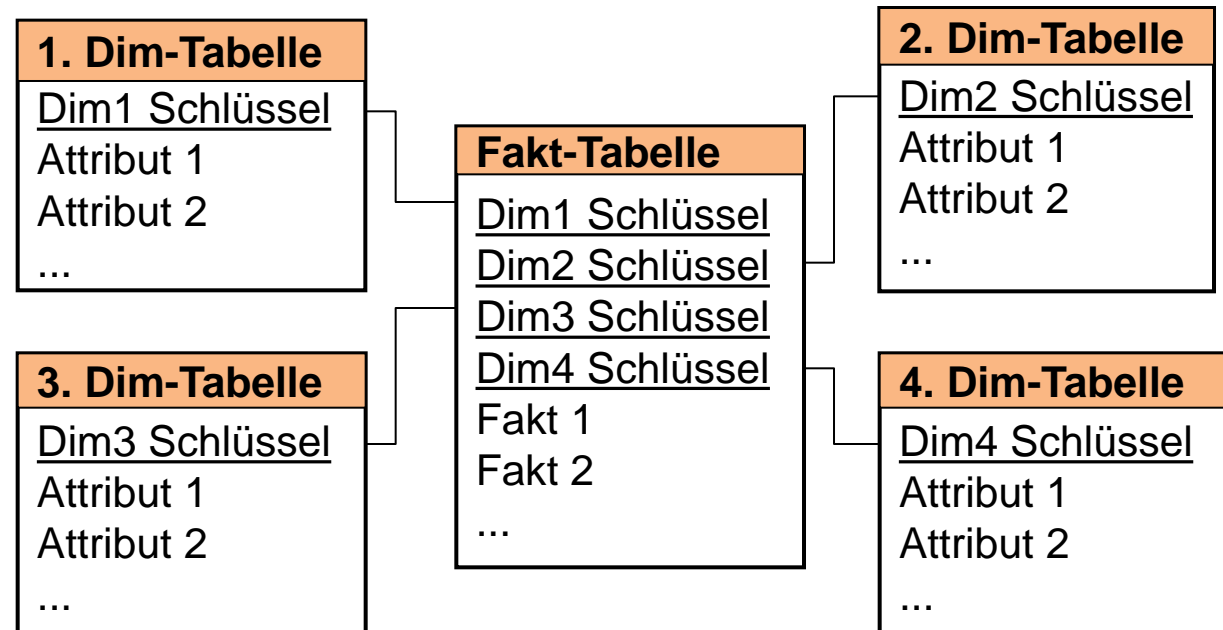
Ein **Star-Schema** ist ein Datenbankschema, welches so strukturiert ist, dass eine typische Abfrage zur Entscheidungsfindung unterstützt wird.

Im Zentrum des Schemas befindet sich die Faktabelle, um welche die Dimensionstabellen angeordnet sind.

Die Verbindung zwischen den Dimensions- und der Faktabellen wird über Fremdschlüssel-Primärschlüssel-Beziehungen realisiert.

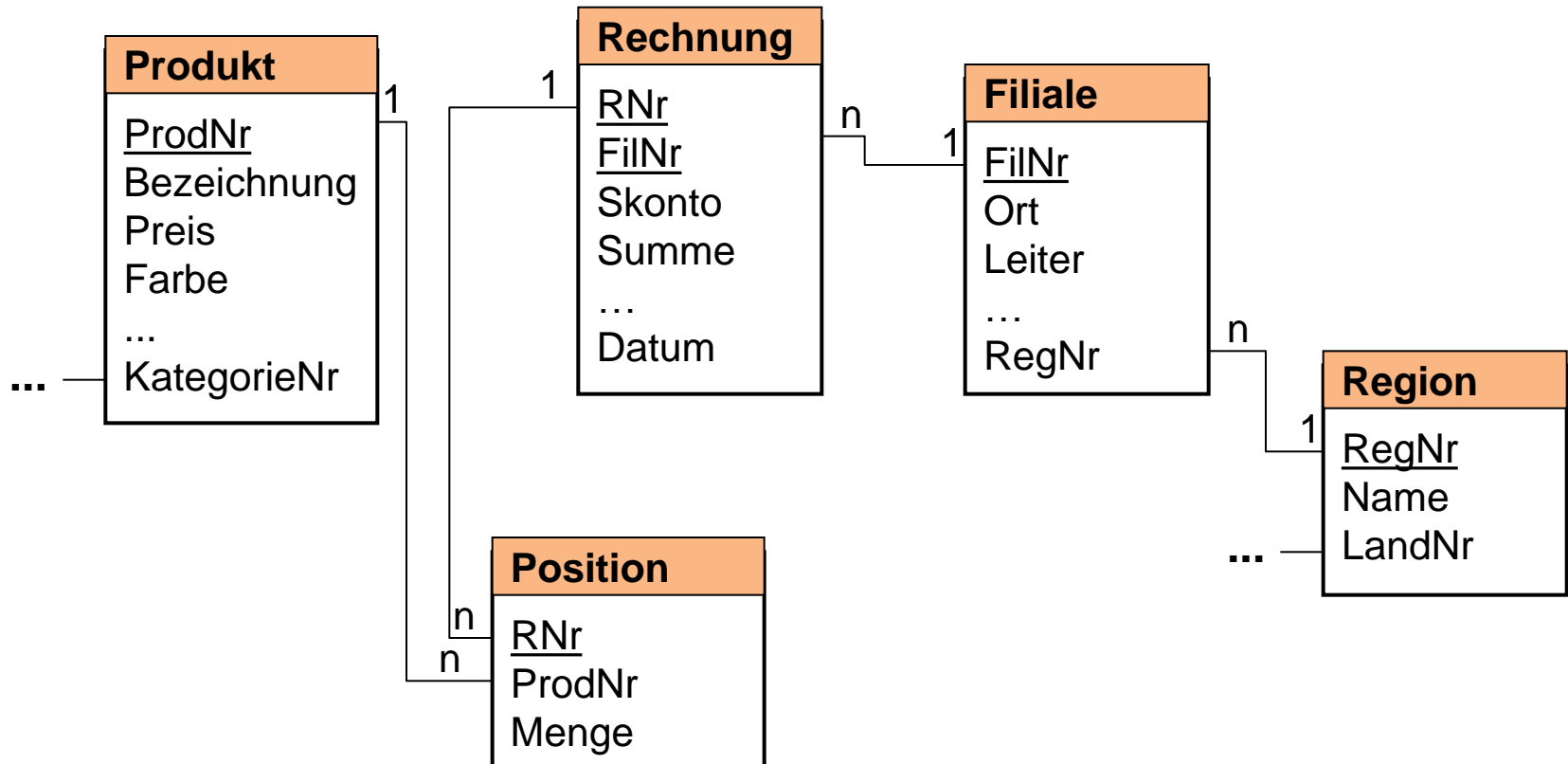
- **Verwendung des relationalen Datenmodells zur Abbildung multidimensionaler Strukturen**

Aufbau eines Starschemas:



Ausschnitt aus dem Schema einer operativen Datenbank

Bsp.: Datenverwaltung für die Rechnungslegung auf Filialbasis



Auszug aus der Dimensionstabelle „Zeit“

Z_ID	Monat	Jahr
1	Januar	2002
2	Februar	2002
3	März	2002
4	April	2002
5	Mai	2002
6	Juni	2002
7	Juli	2002
8	August	2002
9	September	2002
10	Oktober	2002
11	November	2002
12	Dezember	2002
13	Januar	2003
14	Februar	2003
15	März	2003

Beispieldaten - Star Schema

dimension table
product

Product_ID	Brand_Name	Product_Name	Price	Category_ID	Category_Name
1001	American	Chicken Hot Dogs	2,45 €	7	Hot Dogs
1002	American	Extra Light Hamburger	1,20 €	6	Hamburger
1003	American	Chicken Fricassee	3,00 €	5	Frozen Chicken
1004	American	Chicken Breast	2,05 €	5	Frozen Chicken
...					
1309	Danone	Creme fraiche	0,52 €	11	Sour Cream
1310	Danone	Creme Legere	2,22 €	11	Sour Cream
1311	Danone	Leerdammer special	0,71 €	2	Cheese
...					

fact table sales

Time_ID	State_ID	Product_ID	Sales_Euro	Sales_Units
1	1	1001	126,04 €	46
1	1	1002	62,10 €	46
1	1	1003	149,52 €	42
...				
7	17	1309	10,80 €	18
7	17	1310	191,99 €	73
7	17	1311	6,30 €	7
...				

dimension table
time

Time_ID	Year	Quarter	Month
1	2002	1	1
2	2002	1	2
3	2002	1	3
4	2002	2	4
5	2002	2	5
6	2002	2	6
7	2002	3	7
8	2002	3	8
9	2002	3	9
10	2002	4	10
11	2002	4	11
12	2002	4	12
13	2003	1	1
14	2003	1	2
...			

dimension table
state

State_ID	Federal_State	Region	Country
1	Saxony	East	Germany
2	Bavaria	South	Germany
3	Saarland	West	Germany
...			
16	Hesse	West	Germany
17	Valais	South	Switzerland
18	Ticino	South	Switzerland
...			

Eigenschaften des Star Schemas

- Bezug mehrerer Dimensionstabellen auf eine Fakttablelle
- Große Datensatzanzahl in der Fakttablelle gegenüber den Dimensionstabellen
- 1:n – Beziehung jeder Dimensionstabelle zur Fakttablelle
- hohe Abfrageeffizienz (Abfrage auf großer Fakttablelle mit einfachem join zu kleinen Dimensionstabellen)
 - Abfrage auf großer Fakttablelle mit einfachem join zu kleinen Dimensionstabellen
 - Bildung des join nur zwischen Fakttablelle und der jeweiligen Dimensionstabelle
- einfache Anfrageerstellung durch geringe Tabellenanzahl
- Bei Verwendung von level-Attributen
 - ⇒ Zugriff auf aggregierte Werte nicht performater als auf Detailwerte
- hoher Aufwand bei Änderung der Dimensionshierarchien

Snowflake Schema

Ein **Snowflake Schema** ist ein Datenbankschema, welches aus dem Star Schema in der Weise abgeleitet wird, dass die Dimensionstabellen normalisiert werden. In jeder Dimension wird für jede Hierarchieebene eine eigene Tabelle eingeführt. Die Verbindung zwischen den Dimensions- und der Fakttabellen wird über Fremdschlüssel - Primärschlüssel-Beziehungen realisiert.

Die einzelne **Dimensionstabelle** enthält:

- Primärschlüssel für den Hierarchieknoten (z.B. P_Nr)
- beschreibendes Attribut (z.B. Name)
- Fremdschlüssel der nächst höheren Hierarchieebene (z.B. K_Nr)

Die **Fakttabelle** enthält (neben den Fakten):

- Fremdschlüssel der jeweils niedrigsten Hierarchiestufe der Dimensionen (z.B. P_Nr)
- Primärschlüssel als zusammengesetzter Schlüssel, bestehend aus den Fremdschlüsseln der niedrigsten Hierarchiestufen der Dimensionen (z.B. P_Nr F_Nr M_Nr)

Beispieldaten - Snowflake Schema

dimension tables
of *product*

Category_ID	Category_Name
2	Cheese
6	Hamburger
11	Sour Cream
5	Frozen Chicken
7	Hot Dogs
...	...

Product_ID	Brand_Name	Product_Name	Price	Category_ID
1001	American	Chicken Hot Dogs	2,45 €	7
1002	American	Extra Light Hamburger	1,20 €	6
1003	American	Chicken Fricassee	3,00 €	5
1004	American	Chicken Breast	2,05 €	5
...
1309	Danone	Creme fraiche	0,52 €	11
1310	Danone	Creme Legere	2,22 €	11
1311	Danone	Leerdammer special	0,71 €	2
...

fact table
sales

Time_ID	State_ID	Product_ID	Sales_Euro	Sales_Units
1	1	1001	126,04 €	46
1	1	1002	62,10 €	46
1	1	1003	149,52 €	42
...
7	17	1309	10,80 €	18
7	17	1310	191,99 €	73
7	17	1311	6,30 €	7
...

...

...

Vergleich von Star Schema und Snowflake Schema

Star Schema

- denormalisierte Dimensionstabellen
- nur JOIN zwischen Fakt- und Dimensionstabellen
- Insert- und Updateanomalien bei Änderungen in den Dimensionen vorhanden



Anwendung bei geringer Änderungshäufigkeit in den Dimensionswerten

Snowflake Schema

- normalisierte Dimensionstabellen
- JOIN zwischen Fakt- und Dimensionstabellen sowie zwischen den Tabellen einer Dimension
- Insert- und Updateanomalien bei Änderungen in den Dimensionen nicht vorhanden



Anwendung bei großer Änderungshäufigkeit in den Dimensionswerten

Galaxie Schema

Star Schema

- Eine Faktttabelle
- Mehrere Kennzahlen nur möglich bei gleichen Dimensionen
- Bsp.:

Kennzahlen ***Istumsatz*** und ***Umsatzmenge***, beide abhängig von den Dimensionen *Zeitraum*, *Geografie* und *Vertragstyp*

Galaxie Schema

- Mehrere Faktttabellen
- Kennzahlen nur teilweise mit gleichem Dimensionsbezug
- Bsp:

Kennzahl ***Istumsatz***, abhängig von den Dimensionen *Zeitraum*, *Geografie* und *Vertragstyp*

Kennzahl ***Planumsatz***, abhängig von den Dimensionen *Zeitraum* und *Geografie*, keine Abhängigkeit vom *Vertragstyp*

Beispieldaten - Star Schema mit level-Attribut

fact table *sales*

Time_ID	State_ID	Product_ID	Sales_Euro	Sales_Units
1	1	1001	126,04 €	46
1	1	1002	62,10 €	46
1	1	1003	149,52 €	42
...				
7	17	1309	10,80 €	18
7	17	1310	191,99 €	73
7	17	1311	6,30 €	7
...				
25	1	1001	345,05 €	205
26	1	1001	205,20 €	209
27	1	1001	335,67 €	245
28	1	1001	456,28 €	234
29	1	1001	1.342,20 €	893
107	1	1001	2.345,45 €	1453
...				

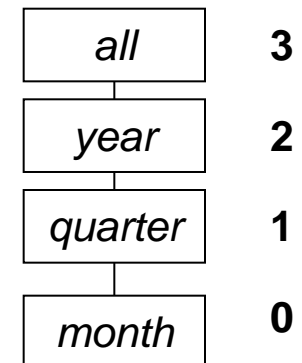
dimension table *time*

Time_ID	Year	Quarter	Month	level
1	2002	1	1	0
2	2002	1	2	0
3	2002	1	3	0
4	2002	2	4	0
5	2002	2	5	0
6	2002	2	6	0
7	2002	3	7	0
8	2002	3	8	0
9	2002	3	9	0
10	2002	4	10	0
11	2002	4	11	0
12	2002	4	12	0
25	2002	1	NULL	1
26	2002	2	NULL	1
27	2002	3	NULL	1
28	2002	4	NULL	1
29	2002	NULL	NULL	2
13	2003	1	1	0
14	2003	1	2	0
...				
107	NULL	NULL	NULL	3

dimension table *state*

dimension table *product*

level attributes:



Beispieldaten - Fact Constellation

only one
aggregation
level for
dimension
time: quarter

dimension table *product*

Product_ID	Brand_Name	Product_Name	Price	Category_ID	Category_Name
1001	American	Chicken Hot Dogs	2,45 €	7	Hot Dogs
1002	American	Extra Light Hamburger	1,20 €	6	Hamburger
1003	American	Chicken Fricassee	3,00 €	5	Frozen Chicken
1004	American	Chicken Breast	2,05 €	5	Frozen Chicken
...					
1309	Danone	Creme fraiche	0,52 €	11	Sour Cream
1310	Danone	Creme Legere	2,22 €	11	Sour Cream
1311	Danone	Leerdammer special	0,71 €	2	Cheese
...					

fact table *sales*

Time_ID	State_ID	Product_ID	Sales_Euro	Sales_Units
1	1	1001	126,04 €	46
1	1	1002	62,10 €	46
1	1	1003	149,52 €	42
...				
7	17	1309	10,80 €	18
7	17	1310	191,99 €	73
7	17	1311	6,30 €	7
...				

fact table *sales_quarter_product_state*

Year	Quarter	State_ID	Product_ID	Sales_Euro	Sales_Units
2002	1	1	1001	345,05 €	205
2002	2	1	1001	205,20 €	209
2002	3	1	1001	335,67 €	245
2002	4	1	1001	456,28 €	234
2003	1	1	1001	367,34 €	239
2003	2	1	1001	521,23 €	312
...					

dimension
table *time*

Time_ID	Year	Quarter	Month
1	2002	1	1
2	2002	1	2
3	2002	1	3
4	2002	2	4
5	2002	2	5
6	2002	2	6
7	2002	3	7
8	2002	3	8
9	2002	3	9
10	2002	4	10
11	2002	4	11
12	2002	4	12
13	2003	1	1
14	2003	1	2
...			

dimension
table *state*

State_ID	Federal_State	Region	Country
1	Saxony	East	Germany
2	Bavaria	South	Germany
3	Saarland	West	Germany
...			
16	Hesse	West	Germany
17	Valais	South	Switzerland
18	Ticino	South	Switzerland
...			

Speicherung vorberechneter Aggregate im relationalen Schema

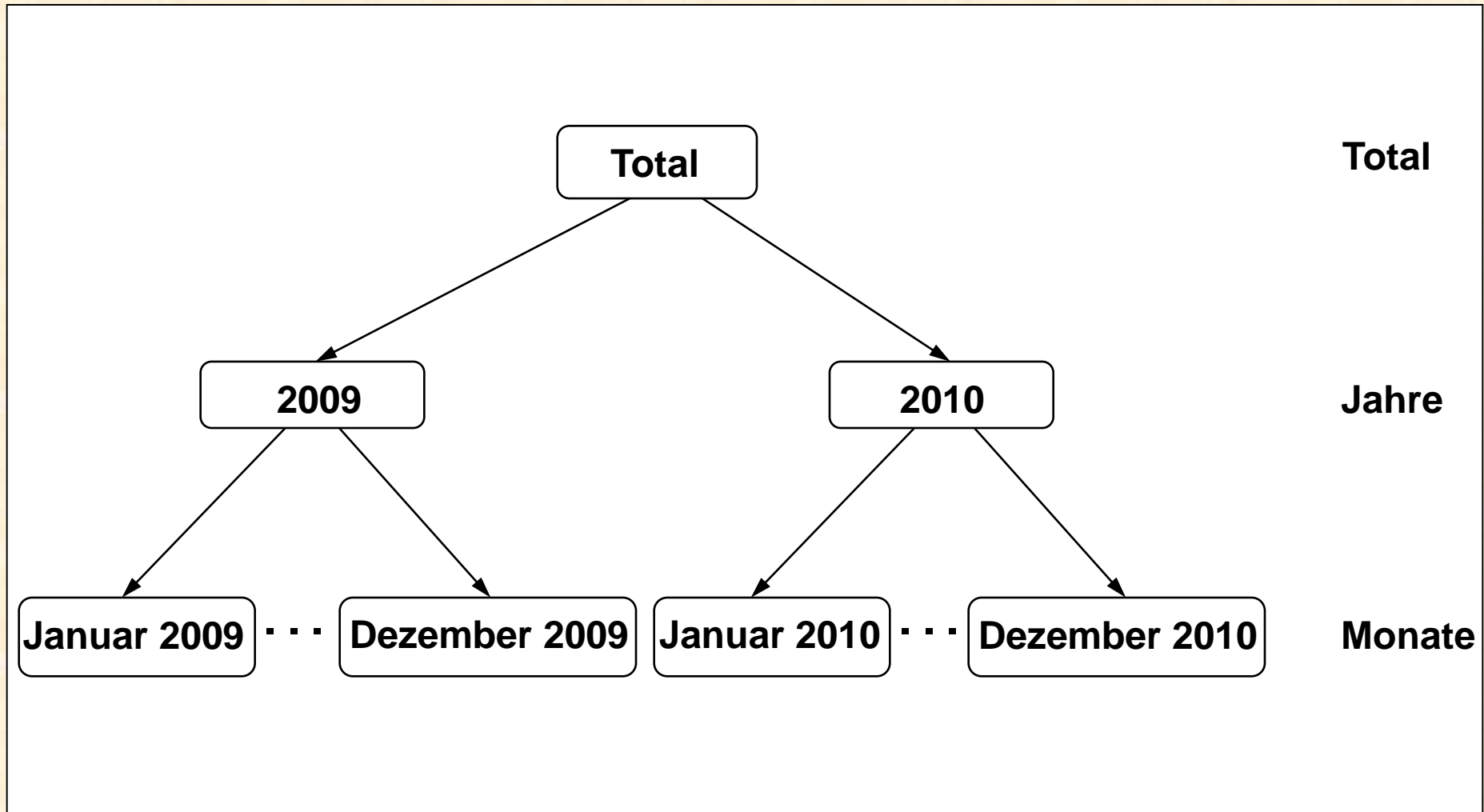
Anmerkungen zur Verwendung von Level-Attributen im Star-Schema

- Speicherung der vorberechneten Aggregate in weiteren Tupeln der Fakttabelle
- Einführung eines Level-Attributes in den Dimensionstabellen
- schnellerer Zugriff auf Aggregate bei der Schemavariante mit Level-Attribut als beim klassischen Star-Schema, kein schnellerer Zugriff auf aggregierte Werte im Vergleich zu Detailwerten
- anwendbar nur beim Star-Schema, nicht beim Shnowflake-Schema
- Erfordernis der Änderung von Daten in der Fakttabelle (ggf. Anpassen der Aggregate beim Laden)
- einfache Abfrage auch für aggregierte Werte
- praktisch nicht alle Aggregattupel speicherbar („kombinatorische Explosion“)

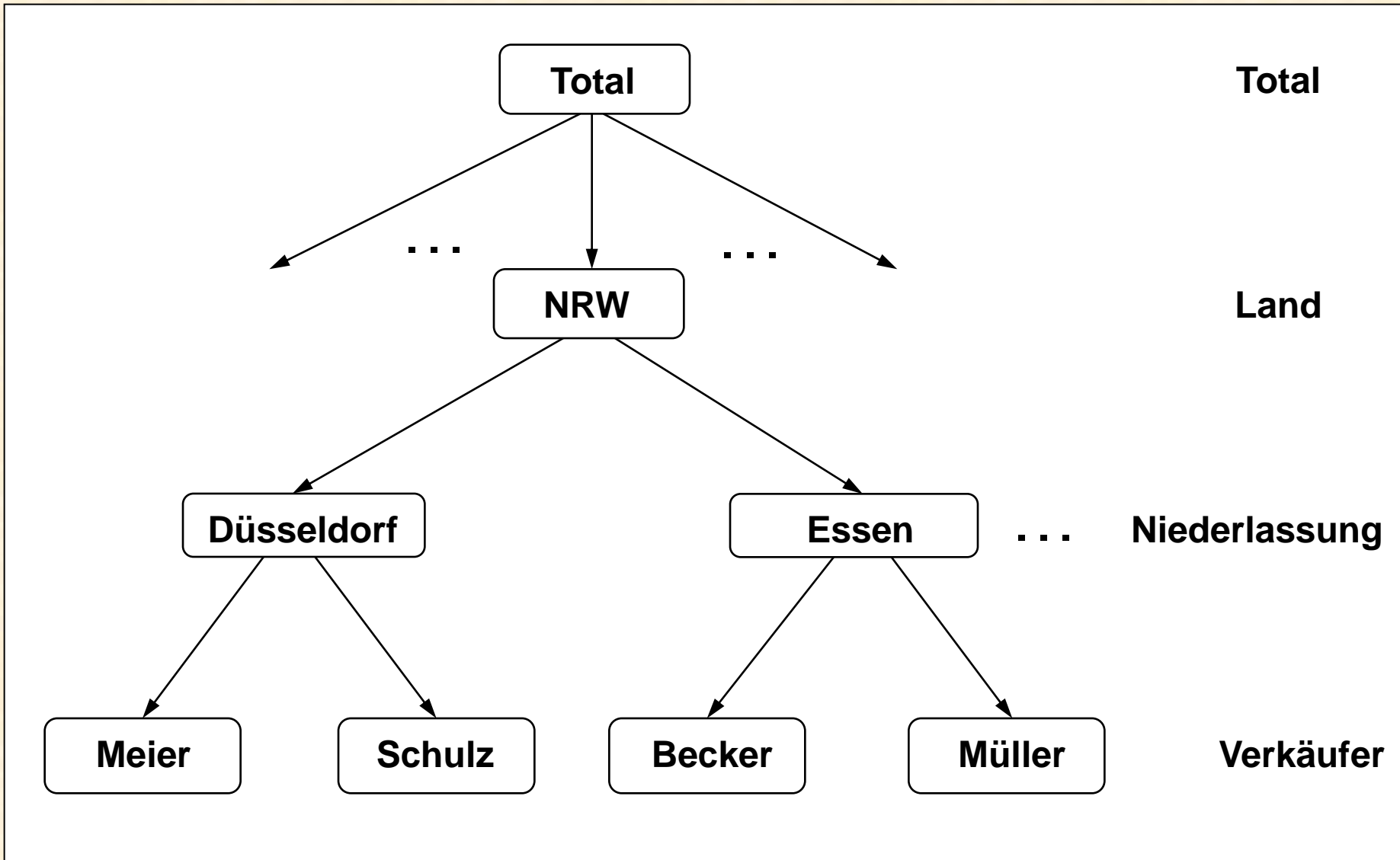
Anmerkungen zum Schema Fact Constellation

- Speicherung der vorberechneten Aggregate in zusätzlichen Fakttabellen
- keine Änderungen in den Dimensionstabellen
- schnellerer Zugriff auf aggregierte Werte im Vergleich zu Detailwerten
- anwendbar auch beim Shnowflake-Schema
- keine Änderung von Daten in der ursprünglichen Fakttabelle des Star-Schemas
- hohe Komplexität der Abfrage auf aggregierte Werte
- praktisch nicht alle Aggregattabellen speicherbar („kombinatorische Explosion“)

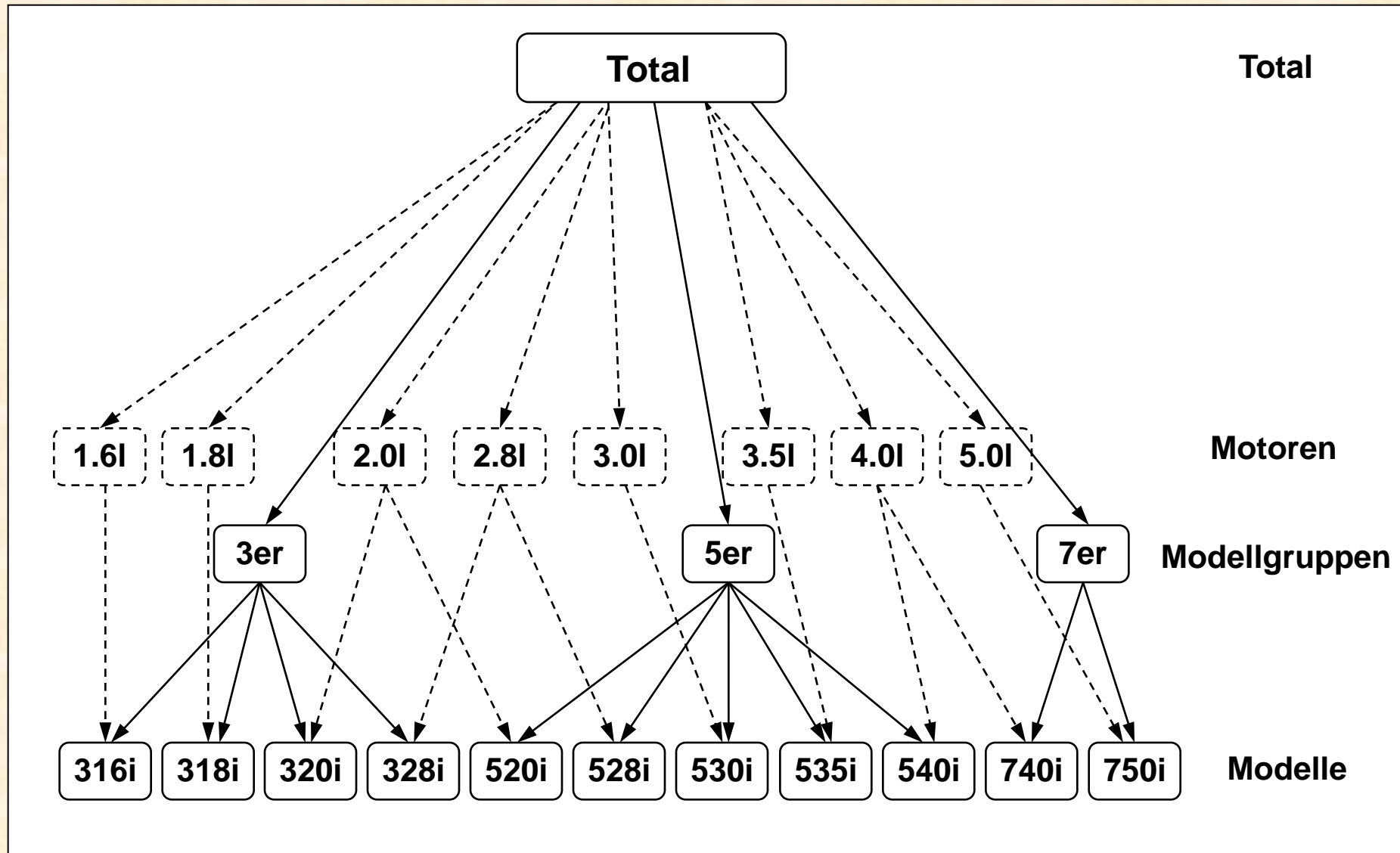
Beispiel Autohändler: Hierarchie der Dimension Zeit



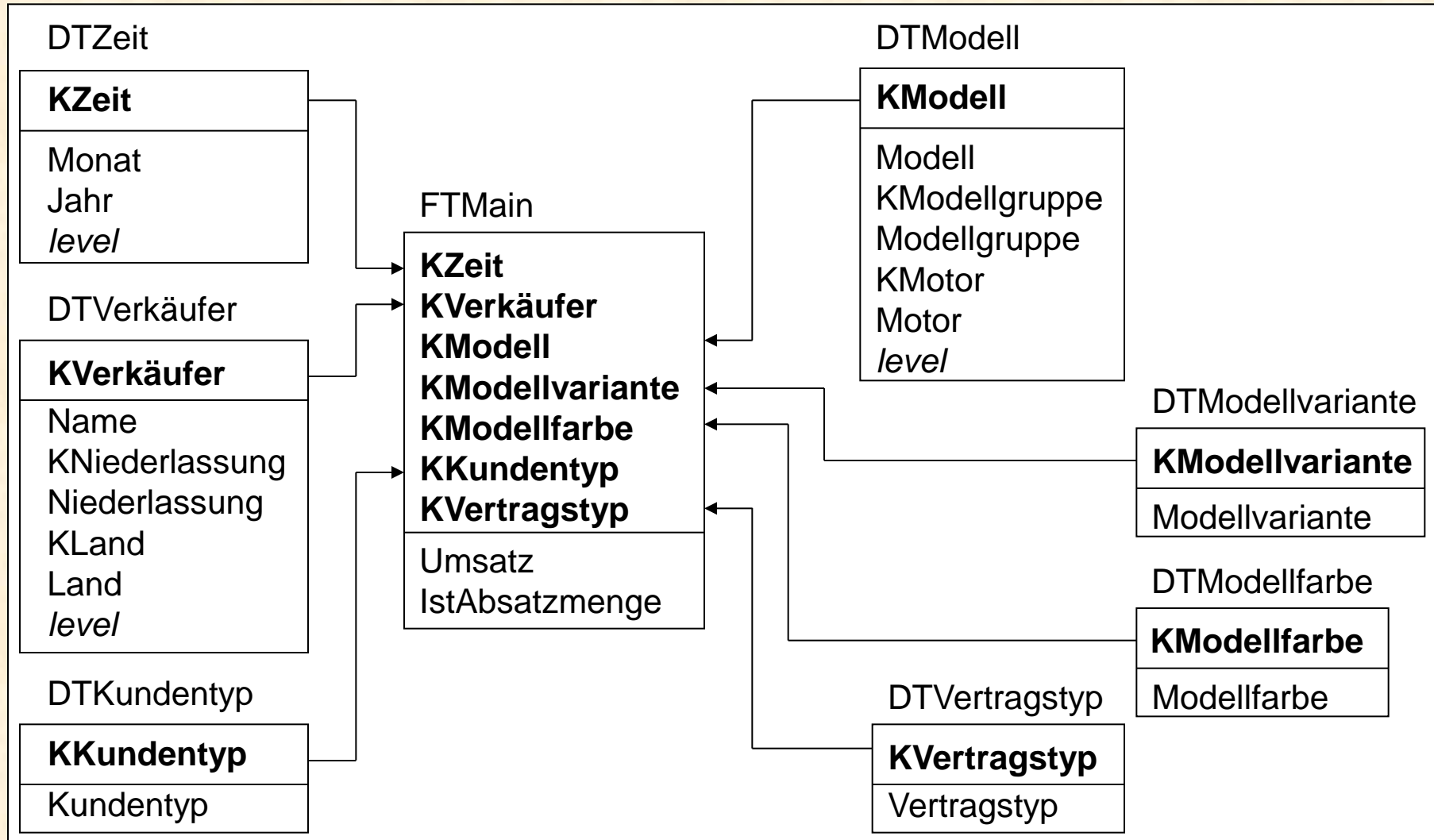
Beispiel Autohändler: Hierarchie der Dimension Verkäufer



Beispiel Autohändler: Hierarchie der Dimension Modelle



Beispiel Autohändler: Star Schema



Dimensionstabelle „Verkäufer“

KVerkäufer	Name	KNiederlassung	Niederlassung	KLand	Land	<i>level</i>
1	Meier	1	Düsseldorf	1	NRW	0
2	Schneider	1	Düsseldorf	1	NRW	0
3	Bäcker	2	Essen	1	NRW	0
4	Müller	2	Essen	1	NRW	0
5	Null	1	Düsseldorf	1	NRW	1
6	Null	2	Essen	1	NRW	1
7	Null	Null	Null	1	NRW	2
8	Null	Null	Null	Null	Null	3

Dimensionstabelle „Modelle“

KModell	Modell	KModellgruppe	Modellgruppe	KMotor	Motor	level
1	316i	1	3er	1	1,6 Liter	Modelle
2	318i	1	3er	2	1,8 Liter	Modelle
3	320i	1	3er	3	2,0 Liter	Modelle
4	328i	1	3er	4	2,8 Liter	Modelle
5	520i	2	5er	3	2,0 Liter	Modelle
6	528i	2	5er	4	2,8 Liter	Modelle
7	530i	2	5er	5	3,0 Liter	Modelle
8	535i	2	5er	6	3,5 Liter	Modelle
9	540i	2	5er	7	4,0 Liter	Modelle
10	740i	3	7er	7	4,0 Liter	Modelle
11	750i	3	7er	8	5,0 Liter	Modelle
12	Null	1	3er	Null	Null	Modellgruppen
13	Null	2	5er	Null	Null	Modellgruppen
14	Null	3	7er	Null	Null	Modellgruppen
15	Null	Null	Null	1	1,6 Liter	Motoren
16	Null	Null	Null	2	1,8 Liter	Motoren
17	Null	Null	Null	3	2,0 Liter	Motoren
18	Null	Null	Null	4	2,8 Liter	Motoren
19	Null	Null	Null	5	3,0 Liter	Motoren
20	Null	Null	Null	6	3,5 Liter	Motoren
21	Null	Null	Null	7	4,0 Liter	Motoren
22	Null	Null	Null	8	5,0 Liter	Motoren
23	Null	Null	Null	Null	Null	Total

Dimensionstabellen mit flacher Hierarchie

KModellvariante	Modellvariante
1	Stufenheck
2	Touring
3	Null

KModellfarbe	Modellfarbe
1	Rot
2	Schwarz
3	Blau
4	Null

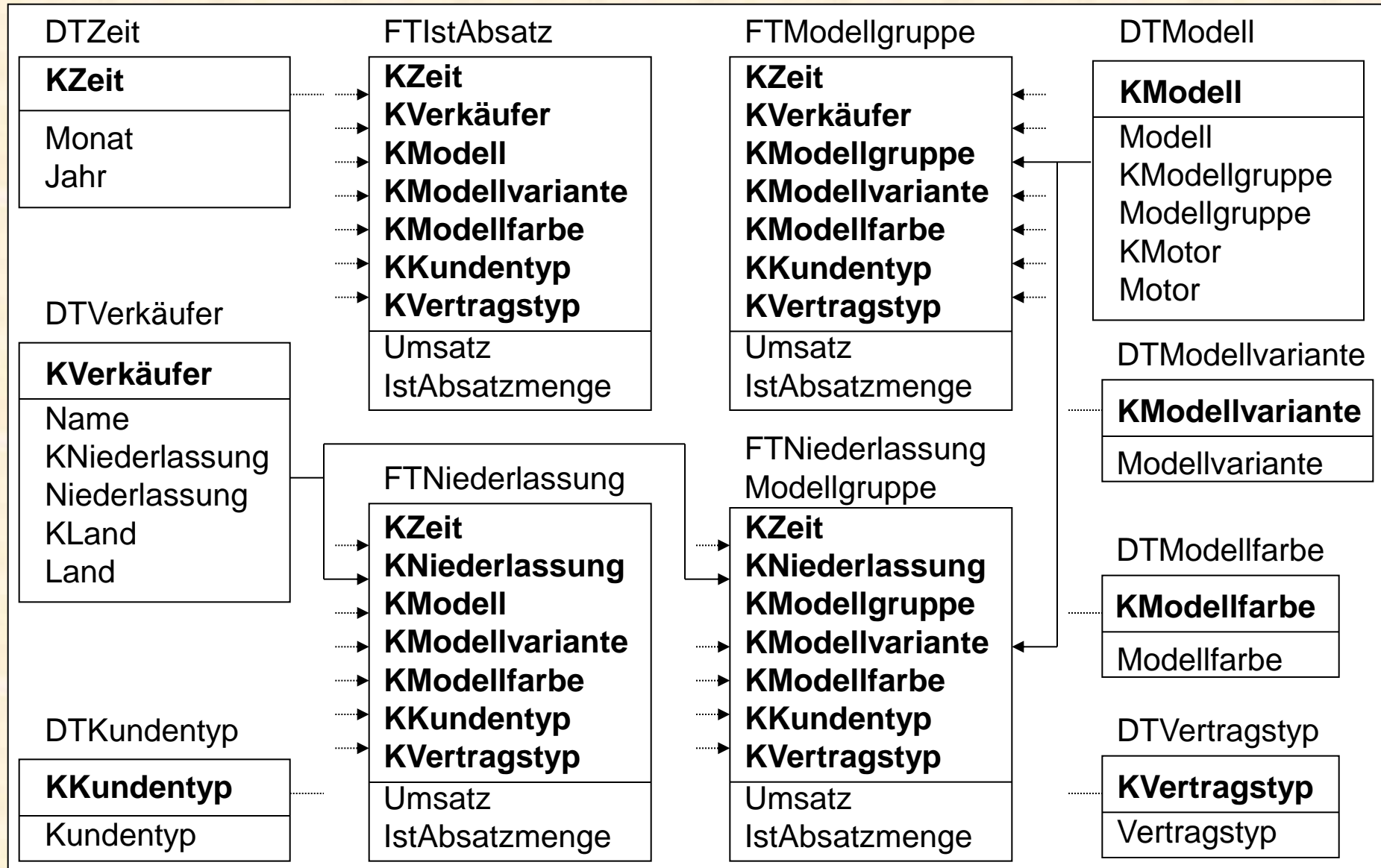
KKundentyp	Kundentyp
1	Altkunde
2	Neukunde
3	Null

KVertragstyp	Vertragstyp
1	Barverkauf
2	Finanzierung
3	Leasing
4	Null

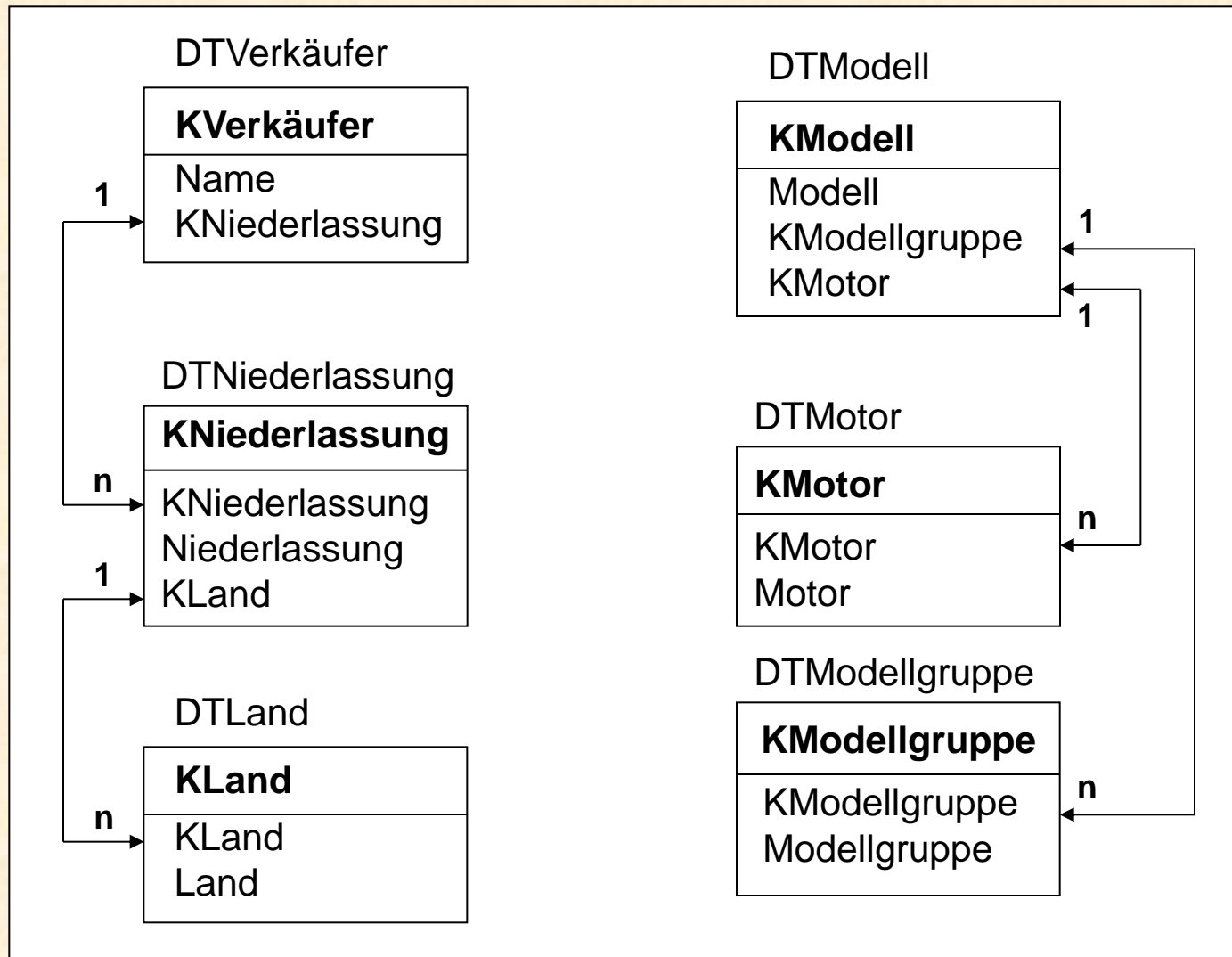
Faktentabelle im Star Schema

KZeit	KVer- käufer	KModell	KModell- variante	KModell- farbe	KKunden- typ	KVertrags- typ	Umsatz	IstAbsatz- menge
1	1	2	1	1	2	2	60000	2
1	1	3	1	2	2	2	120080	3
1	3	3	1	1	1	2	80000	2
1	4	2	2	2	2	2	30000	1
1	4	3	2	2	2	1	40000	1
1	4	4	1	1	2	2	100000	2
2	1	4	1	3	1	1	50000	1
2	2	3	1	3	2	2	160000	4
2	2	2	2	2	2	2	90000	3
2	4	4	1	1	2	2	50000	1
.....								
13	1	23	3	4	3	4	230000	6
13	2	23	3	4	3	4	250000	7
13	3	23	3	4	3	4	80000	2
13	4	23	3	4	3	4	220080	5
.....								
13	8	23	3	4	3	4	780000	20
.....								

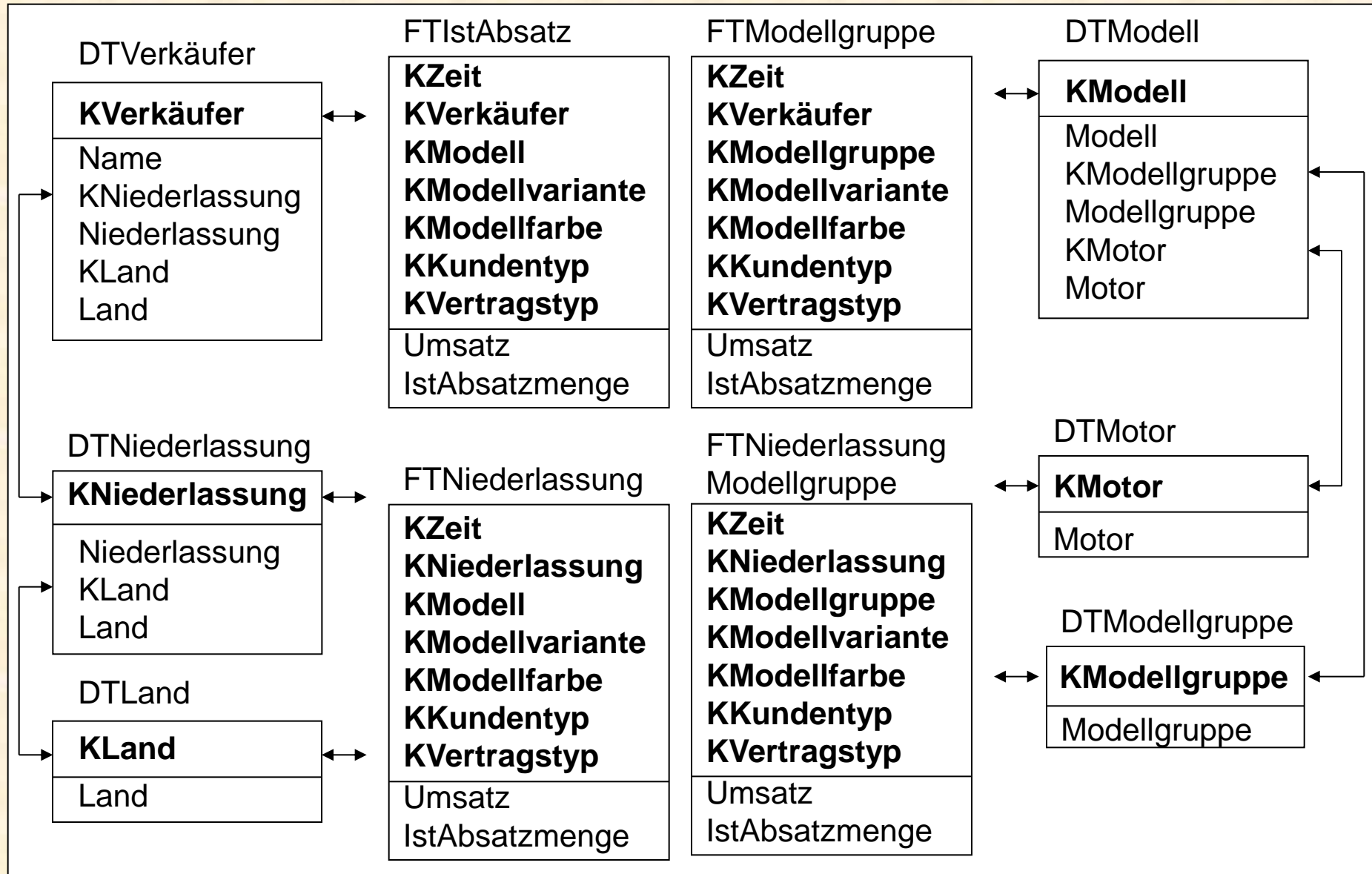
Fact Constellation Schema



Beispiel normalisierter Dimensionstabellen



Snow Flake Schema



Star-Join

SELECT	<ul style="list-style-type: none">➤ Attribute der Dimensionen➤ Kennzahlen [aggregiert]	
FROM	<ul style="list-style-type: none">➤ Dimensionstabellen➤ Fakttable	
JOIN ON	<ul style="list-style-type: none">➤ JOIN-Bedingungen	
WHERE	<ul style="list-style-type: none">➤ Explizite Bedingungen➤ [Level = Wert]	[] ... Schemavariante ohne level-Attribut
[GROUP BY]	<ul style="list-style-type: none">➤ Kenngrößen	[] ... Schemavariante mit level-Attribut

Beispiel: Ermittlung der 2015 im Land „Deutschland“ verkauften Produkte mit Namen „Radeberger“

Dimensionen:	Zeit:	Jahr, Quartal, Monat
	Produkt:	Name, Kategorie
	Geografie:	Land, Region, Staat
Kennzahlen:	Mengenumsatz, Wertumsatz	

Ergebnisse unterschiedlicher Abfragen

- Über das GROUP BY-Attribut wird „normal“ gruppiert.
- Für das PIVOT-Attribut der Kreuztabellenabfrage wird in einer zweiten Ebene eine Gruppierung durchgeführt, welche unabhängig vom GROUP BY-Attribut ist.
- Die gruppierten Werte des PIVOT-Attributs werden neue Spalten in der Ergebnisrelation.

Gruppierungsabfrage		
ArtikelNr	Verkaufsgebiet	Verkaufte Einheiten
ALG-001	Ost	150
ALG-002	Nord	53
ALG-002	Ost	150
ALG-003	Nord	20
ALG-003	Süd	30
ALG-003	West	30
ALG-004	Süd	30
ALG-004	West	80
ALG-005	Nord	40
ALG-005	Ost	10
ALG-006	Nord	200

Kreuztabellenabfrage					
ArtikelNr	Gesamtsumme	Nord	Ost	Süd	West
ALG-001	150		150		
ALG-002	203	53	150		
ALG-003	80	20		30	30
ALG-004	110			30	80
ALG-005	50	40	10		
ALG-006	243	200	43		
ALG-007	5		5		
EDV-001	55		25	15	15
EDV-002	78	3	50	25	
EDV-003	55	40	10	5	
EDV-004	52	17		15	20

Beispiel für die Gruppierung mit ROLLUP

```
SELECT Monat, Bundesland, Verkaeuer, SUM(Umsatz) FROM Verkaeufe  
WHERE Jahr = '2009' GROUP BY ROLLUP(Monat, Bundesland, Verkaeuer)
```

Monat	Bundesland	Verkäufer	SUM
April	Sachsen	Müller	25000
April	Sachsen	Lehmann	15000
April	Sachsen	-	40000
April	Thüringen	Lehmann	15000
April	Thüringen	-	15000
April	-	-	55000
Mai	Sachsen	Müller	25000
Mai	Sachsen	-	25000
Mai	Thüringen	Lehmann	15000
Mai	Thüringen	-	15000
Mai	-	-	40000
-	-	-	95000

Beispiel für die Gruppierung mit CUBE - 1

```
SELECT Monat, Bundesland, Verkaeuer, SUM(Umsatz) FROM Verkaeufe  
WHERE Jahr = '2009' GROUP BY CUBE(Monat, Bundesland, Verkaeuer)
```

Monat	Bundesland	Verkäufer	SUM
April	Sachsen	Müller	25000
April	Sachsen	Lehmann	15000
April	Sachsen	-	40000
April	Thüringen	Lehmann	15000
April	Thüringen	-	15000
April	-	Müller	25000
April	-	Müller	30000
April	-	-	55000

•
•
•

Beispiel für die Gruppierung mit CUBE - 2

Fortsetzung

Mai	Sachsen	Müller	25000
Mai	Sachsen	-	25000
Mai	Thüringen	Lehmann	15000
Mai	Thüringen	-	15000
Mai	-	Müller	25000
Mai	-	Lehmann	15000
Mai	-	-	40000
-	Sachsen	Müller	50000
-	Sachsen	Lehmann	15000
-	Sachsen	-	65000
-	Thüringen	Lehmann	30000
-	Thüringen	-	30000
-	-	Müller	50000
-	-	Lehmann	45000
-	-	-	95000

Umsetzung der Hersteller (Auswahl)

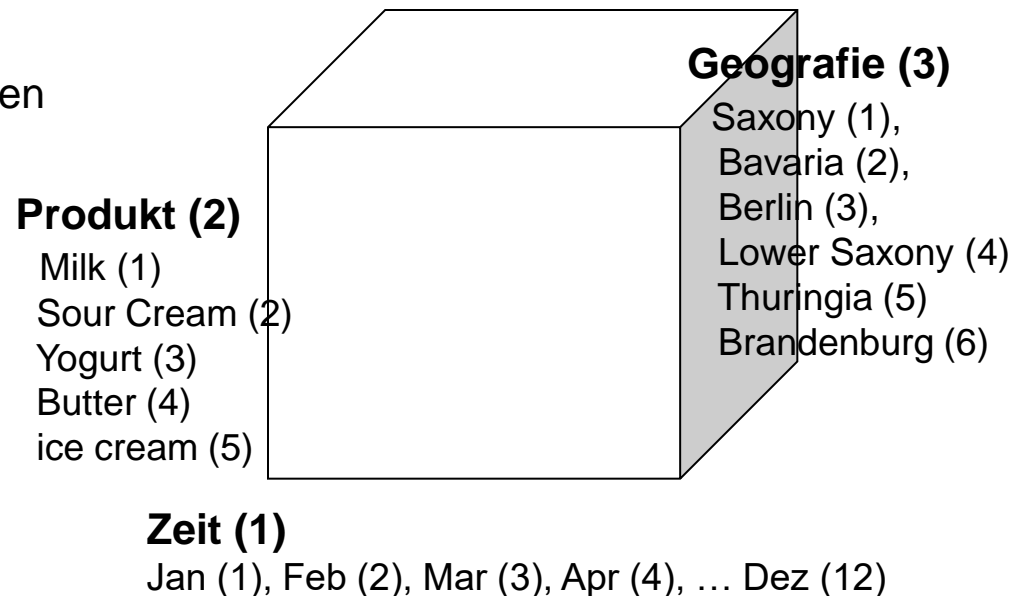
DB2	Oracle	MS SQL Server	SYBASE ASE
Data Warehouse Server der bei einer Standard-installation installiert wird	Integration von Business Intelligence und Data Warehouse Funktionalität in DB-Server	Vielzahl von Funktionen für Data Mining und OLAP	spezieller Data Warehouse Server Adaptive Server IQ
Steuerung über externes Tool, namens DB2 Control Center + Tools für Auswertung	Bietet Data Mining, ROLAP und MOLAP an	Data Transformation Services (DTS) Satz grafischer Tools für DW-Design und ETL	Warehouse Architekt als Data Warehouse und Data Mart Design Tool
hat GROUPING SETS, ROLLUP, CUBE, implementiert	hat GROUPING SETS, ROLLUP, CUBE, RANK OVER, PARTITION BY implementiert	hat GROUPING SETS, ROLLUP, CUBE, implementiert	Warehose Control Center Toolset zur Administration, Warehouse DB und Data Marts

Multidimensionale Speicherung

- Anforderungen:**
- Ordnung zwischen den Dimensionen (i.A. nicht vorhanden)
Beispiel: Zeit = 1; Produkt = 2; Geografie = 3
 - Ordnung zwischen den Dimensionswerte (zum Teil vorhanden)
Beispiel: Jan = 1; Feb = 2; Mar = 3; ...
Milk = 1; Sour Cream = 2; Yogurt = 3
 - Ordnung zwischen den Level einer Dimension (i.A. vorhanden)
Beispiel: month = 1; quarter = 2; year = 3

Berechnung:

- n- dimensionaler Array
- In jeder Dimension mit $|D_n|$ Werten
- Berechnung des Index:
$$= x_1 + (x_2 - 1) * |D_1|$$
$$+ (x_2 - 1) * |D_1| * |D_2| + \dots$$
$$+ (x_n - 1) * |D_1| * \dots * |D_n|$$
- Speicher für aggregierte Werte wird benötigt



Speicherbedarf bei der Pointerspeicherung - Beispiele

Beispiel 1:

$$\begin{aligned}|D_1| &= 120 \\ |D_2| &= 60 \\ |D_1| * |D_2| &= 7200\end{aligned}$$

$$\begin{aligned}K &= 2500 \\ S &= 0.35\end{aligned}$$

$$\begin{aligned}B &= 12 \text{ Byte} \\ B_P &= 4 \text{ Byte}\end{aligned}$$

$$\begin{aligned}G_V &= 86400 \\ G_P &= 58800\end{aligned}$$

Beispiel 2:

$$\begin{aligned}|D_1| &= 120 \\ |D_2| &= 60 \\ |D_1| * |D_2| &= 7200\end{aligned}$$

$$\begin{aligned}K &= 5000 \\ S &= 0.69\end{aligned}$$

$$\begin{aligned}B &= 12 \text{ Byte} \\ B_P &= 4 \text{ Byte}\end{aligned}$$

$$\begin{aligned}G_V &= 86400 \\ G_P &= 88800\end{aligned}$$

Beispiel 3:

$$\begin{aligned}|D_1| &= 120 \\ |D_2| &= 60 \\ |D_1| * |D_2| &= 7200\end{aligned}$$

$$\begin{aligned}K &= 2500 \\ S &= 0.35\end{aligned}$$

$$\begin{aligned}B &= 12 \text{ Byte} \\ B_P &= 8 \text{ Byte}\end{aligned}$$

$$\begin{aligned}G_V &= 86400 \\ G_P &= 87600\end{aligned}$$

Break-Even-Point

K	S	G _V	G _P
2000	0.28	86400	52800
2200	0.31	86400	55200
2400	0.33	86400	57600
2600	0.36	86400	60000
2800	0.39	86400	62400
3000	0.42	86400	64800
3200	0.44	86400	67200
3400	0.47	86400	69600
3600	0.50	86400	72000
3800	0.53	86400	74400
4000	0.56	86400	76800
4200	0.58	86400	79200
4400	0.61	86400	81600
4600	0.64	86400	84000
4800	0.67	86400	86400
5000	0.69	86400	88800
5200	0.72	86400	91200
5400	0.75	86400	93600
5600	0.78	86400	96000
5800	0.81	86400	98400
6000	0.83	86400	100800

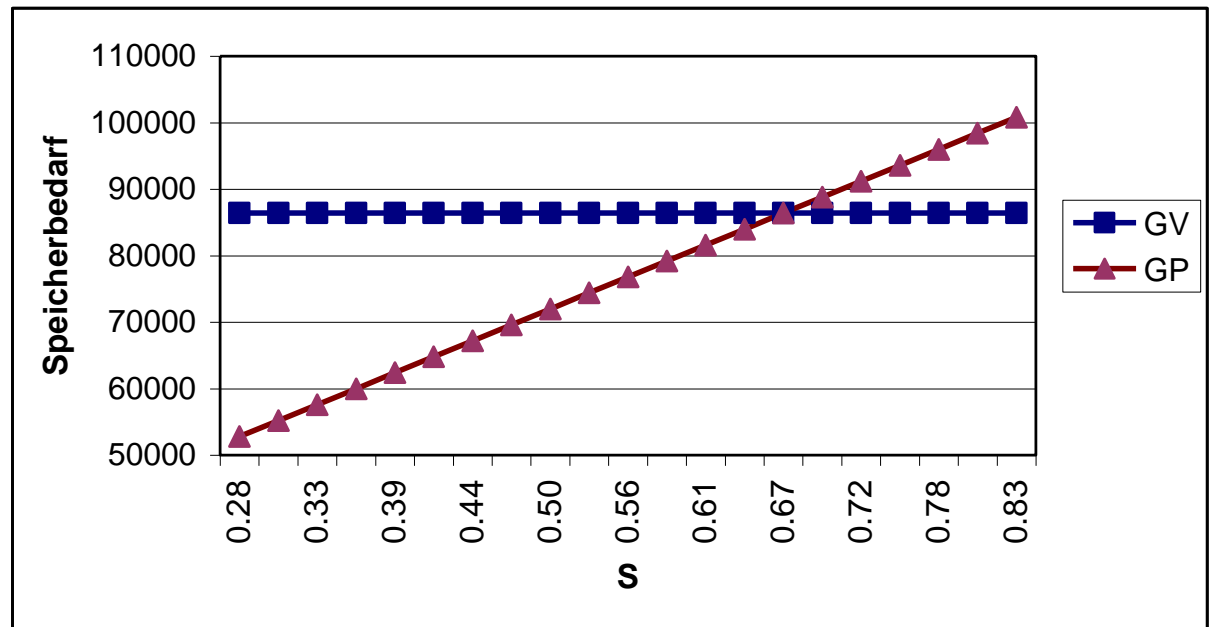
$$|D_1| = 120$$

$$|D_2| = 60$$

$$|D_1| * |D_2| = 7200$$

$$B = 12$$

$$B_P = 4$$



Grenzen multidimensionaler Speicherung

- Skalierbarkeitsprobleme auf Grund dünn besetzter Datenräume
- teilweise einseitige Optimierung bezüglich Leseoperationen
- Notwendigkeit einer Ordnung der Dimensionswerte (durch Array-Speicherung) -> erschwerte Änderbarkeit an den Dimensionen
- Fehlender Standard für multidimensionale DBMS
- Notwendigkeit von Spezialwissen

Anfragebeispiel an ein Relationales DBMS

**Berechnung des durchschnittlichen Absatzes von „Radeberger“
in Gastronomie-Einrichtungen in Sachsen**

Mon	Einr	Typ	Land	Prod	Abs	...
9805	32	G	SA	Werne	6	...
9805	36	G	MV	Becks	9	
9805	38	G	SA	Radeb	5	
9805	41	K	NS	Jever	11	
9805	43	G	SA	Radeb	9	
9805	46	G	BY	Paula	3	
9805	47	M	NW	Dortm	7	
9805	49	K	SA	Lands	12	

```
SELECT AVG (Abs), SUM(Abs)/AnzGSA/36
FROM Absatz,
      (SELECT COUNT(DISTINCT Einr)
       AS AnzGSA
      FROM Absatz
      WHERE Land = 'SA' AND
            Typ = 'G')
WHERE Land = 'SA' AND
      Typ = 'G' AND
      Prod = 'Radeb'
```

Vertikale Partitionierung und Bitmap Index

vertikale Partitionierung

Einr	Typ	Land	Prod	Abs
32	G	SA	Werne	6
36	G	MV	Becks	9
38	G	SA	Radeb	5
41	K	NS	Jever	11
43	G	SA	Radeb	9
46	G	BY	Paula	3
47	M	NW	Dortm	7
49	K	SA	Lands	12

Bitmapindex am Beispiel der Spalte Land

Bitmap Index für Land						
row-id	BY	MV	NS	NW	SA	
1	0	0	0	0	1	
2	0	1	0	0	0	
3	0	0	0	0	1	
4	0	0	1	0	0	
5	0	0	0	0	1	
6	1	0	0	0	0	
7	0	0	0	1	0	
8	0	0	0	0	1	

Beispiel für die I/O-Reduzierung

“Wieviele Männer sind in Kalifornien nicht versichert?”

RDBMS:

↑
20M
Sätze
↓

	Geschlecht	Staat	Versichert
	M	CA	J
	M	CA	N
	W	NY	J
	M	CA	N
	M	MA	J
	W	CT	N

← 800 Bytes/Satz →

$$\frac{800 \text{ Bytes} \times 20\text{M}}{16\text{K Seite}} = 1.000,000 \text{ I/Os}$$

- Verarbeitet grosse Mengen nicht benötigter Daten
- Erfordert oft “Full Table Scan”

Bitmap Index

	Geschlecht	Staat	Versichert
1	M	CA	J
2	M	CA	N
3	W	NY	J
4	M	CA	N

$$\frac{20\text{M Bits} \times 3 \text{ Spalten}}{16\text{K Seite}} = 470 \text{ I/Os}$$

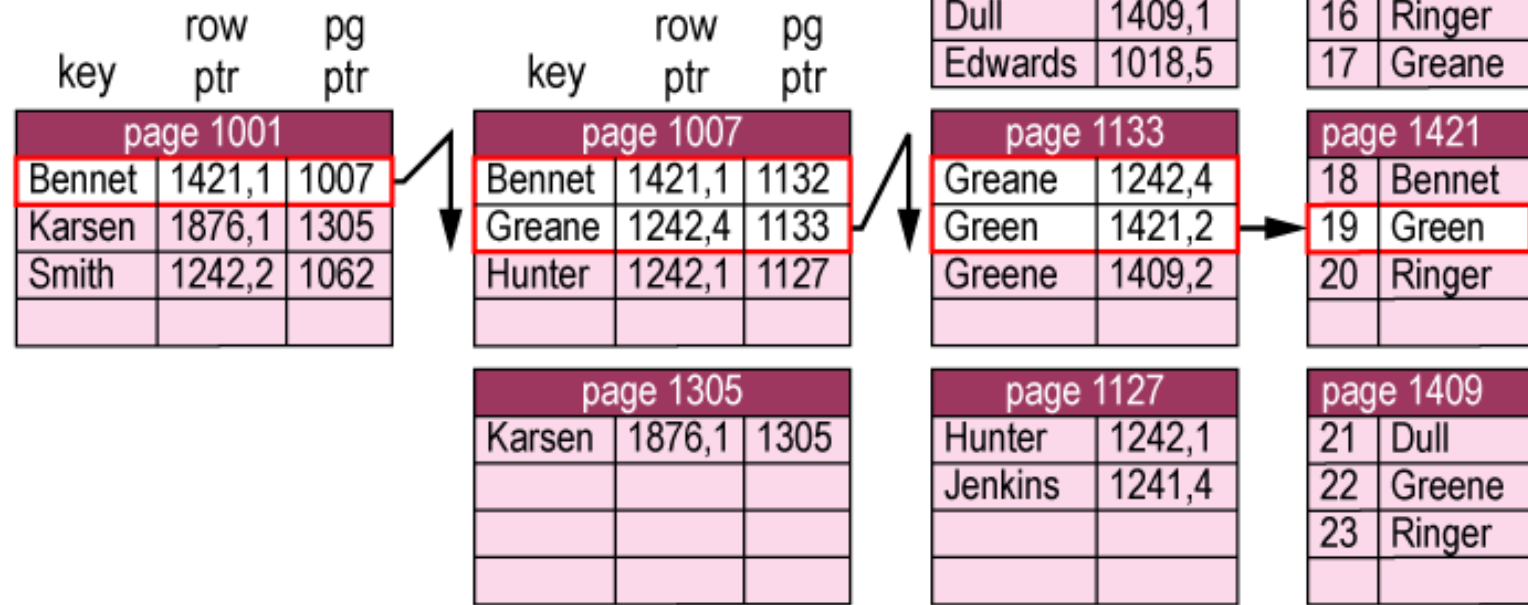
↑
20M
Bits
↓

1	1	0
1	1	1
0	0	0
1	1	1

+ + = 2

Beispiel für einen Index eines RDBMS

```
SELECT * FROM employee
WHERE employee_lname = "Green"
```



Fast Projection - FP

Die Daten einer Spalte werden komprimiert gespeichert.

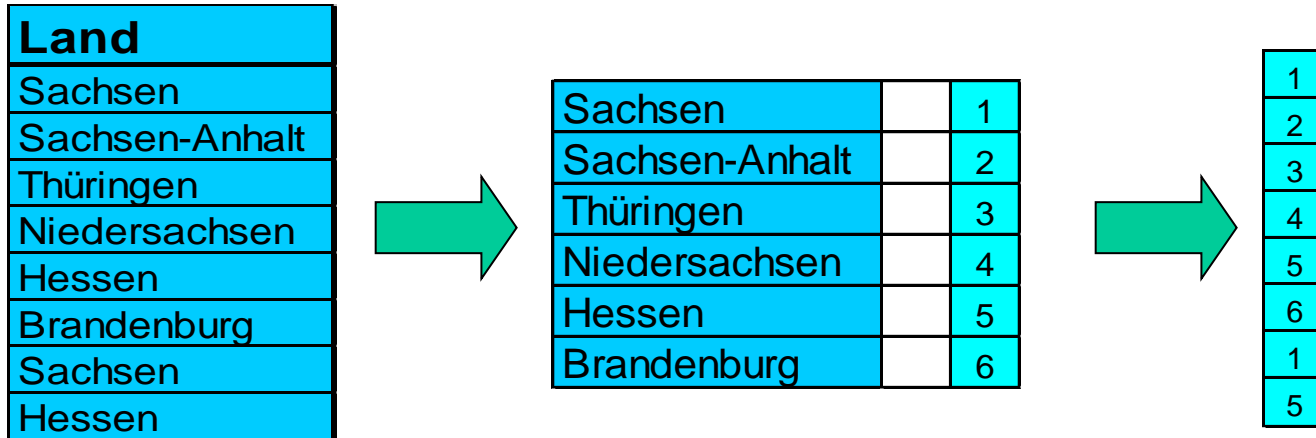
Land
Sachsen
Sachsen-Anhalt
Thüringen
Niedersachsen
Hessen
Brandenburg
Sachsen
Hessen

```
SELECT Land  
FROM Absatz  
WHERE Land LIKE 'Sa%'
```

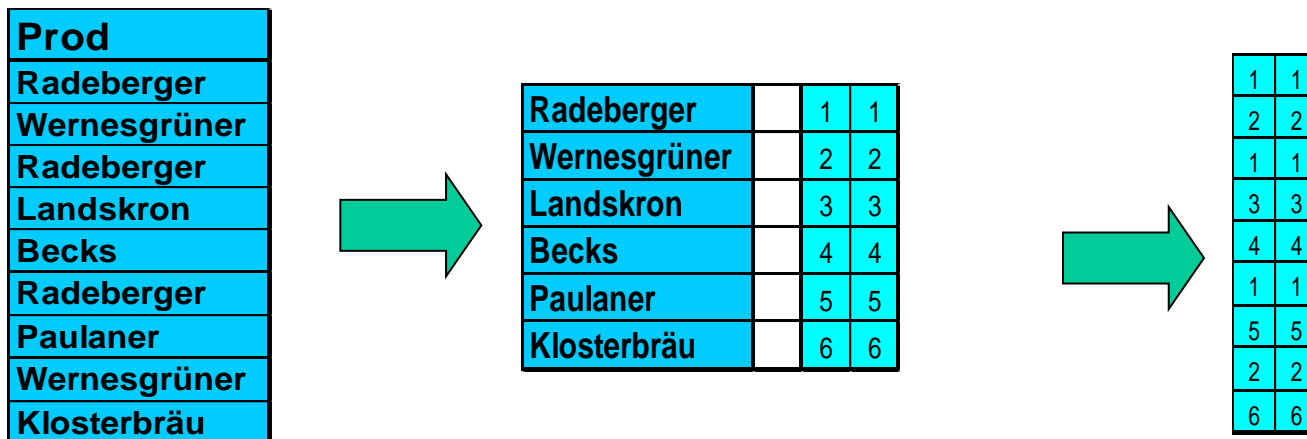
- Default Speicherung, die automatisch durch IQ realisiert wird und nicht entfernt werden kann
- für alle Spalten: notwendig für **select** list Spalten, string Suche, ad-hoc joins

Fast Projection - FFP und FFFP

Subtype: Fast Fast Projection (FFP)



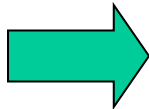
Subtype: Fast Fast Fast Projection (FFFP)



Low Fast

Bitmap Index, der für Spalten mit kleiner Kardinalität benutzt wird

Prod
Radeberger
Wernesgrüner
Radeberger
Landskron
Becks
Radeberger
Paulaner
Wernesgrüner
Klosterbräu



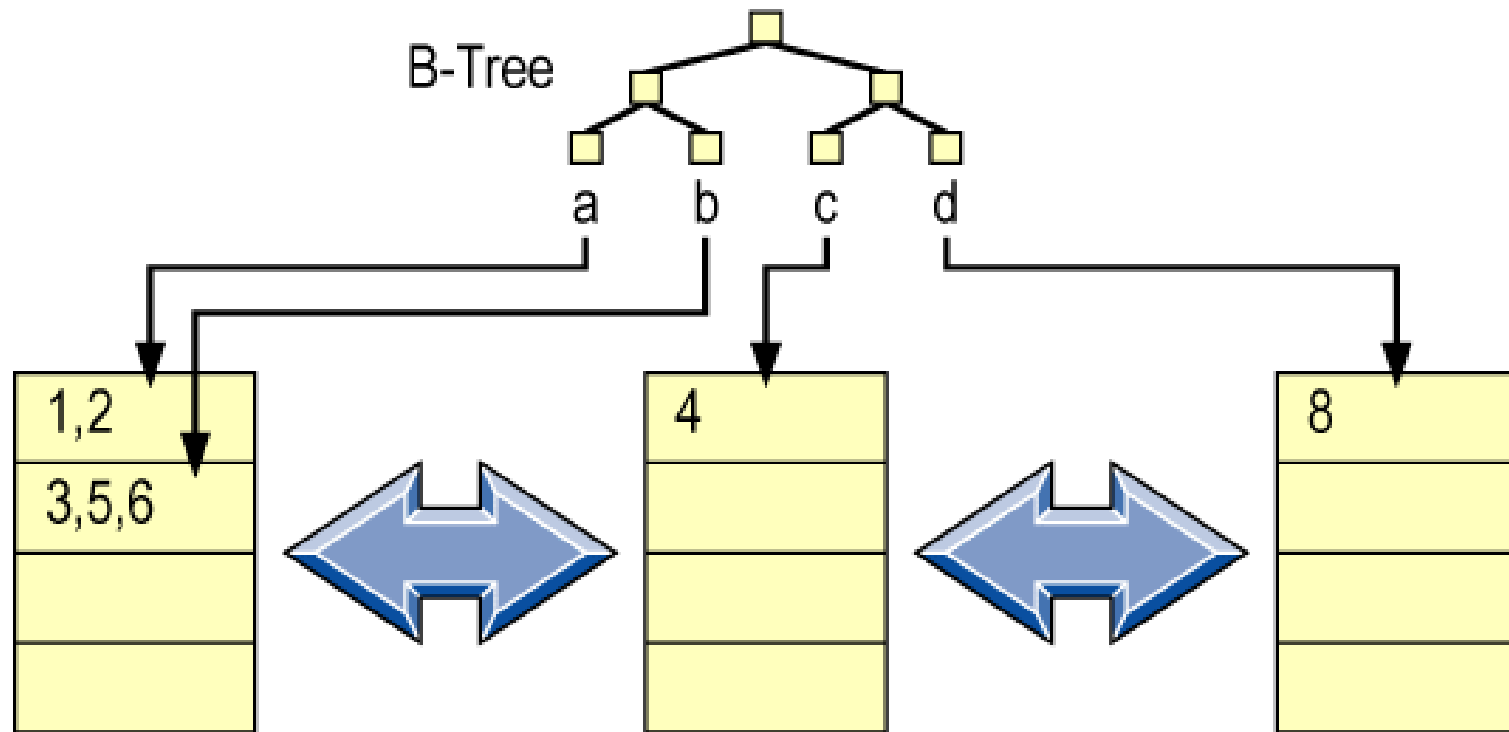
Radeberger		Wernesgrüner		Landskron
1		0		0
0		1		0
1		0		0
0		0		1
0		0		0
1		0		0
0		0		0
0		1		0
0		0		0

```
SELECT *  
FROM Absatz  
WHERE Prod = 'Radeberger'
```

- Eine Menge von werte-basierten Bitmaps wird für Bearbeitung fast aller Anfragen angewendet.
- Ideal für Spalten mit einer Kardinalität <1000

High Group

Index für Daten mit hoher Kardinalität

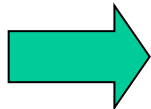


High Non Group

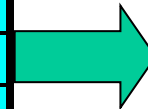
Bit-weiser Index, optimiert für Rang-Suche und Aggregations-Funktionen

Binäre Darstellung

Abs
12
9
28
11
9
3
70
12



0	0	0	1	1	0	0
0	0	0	1	0	0	1
0	0	1	1	1	0	0
0	0	0	1	0	1	1
0	0	0	1	0	0	1
0	0	0	0	0	1	1
1	0	0	0	1	1	0
0	0	0	1	1	0	0



HNG-Index für Abs

64	32	16	8	4	2	1
0	0	0	1	1	0	0
0	0	0	1	0	0	1
0	0	1	1	1	0	0
0	0	0	1	0	1	1
0	0	0	1	0	0	1
0	0	0	0	0	1	1
1	0	0	0	1	1	0
0	0	0	1	1	0	0

1	0	1	6	4	3	4
---	---	---	---	---	---	---

Beispiel:

SELECT SUM(Abs) FROM Absatz

$$(1 * 64) + (0 * 32) + (1 * 16) + (6 * 8) + (4 * 4) + (3 * 2) + (4 * 1) = 154$$

Beispiel für den Einsatz der Indexarten

Berechnung des durchschnittlichen Absatzes von „Radeberger“
in Gastronomie-Einrichtungen in Sachsen

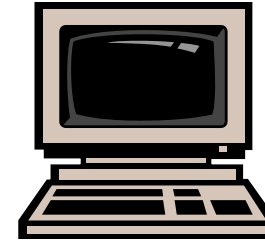
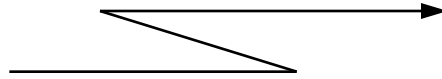
Mon	Einr	Typ	Land	Prod	Abs	...
9805	32	G	SA	Werne	6	...
9805	36	G	MV	Becks	9	
9805	38	G	SA	Radeb	5	
9805	41	K	NS	Jever	11	
9805	43	G	SA	Radeb	9	
9805	46	G	BY	Paula	3	
9805	47	M	NW	Dortm	7	
9805	49	K	SA	Lands	12	

```
SELECT AVG (Abs), SUM(Abs)/AnzGSA/36
FROM Absatz,
      (SELECT COUNT(DISTINCT Einr)
       AS AnzGSA
      FROM Absatz
      WHERE Land = 'SA' AND
            Typ = 'G')
WHERE Land = 'SA' AND
      Typ = 'G' AND
      Prod = 'Radeb'
```

OLAP-Architekturen

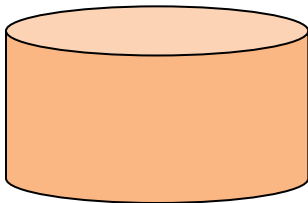


OLAP-Server

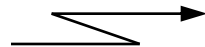


**mehrdimensionale Präsentations-
und Transformationswerkzeuge**

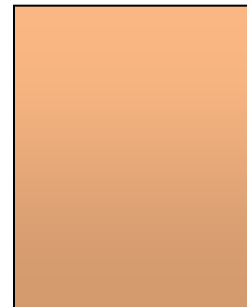
relationaler Daten-Server



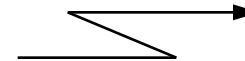
Datenbank-Schicht



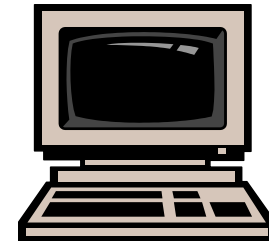
OLAP-Server



**Transformations- und
Berechnungsschicht**

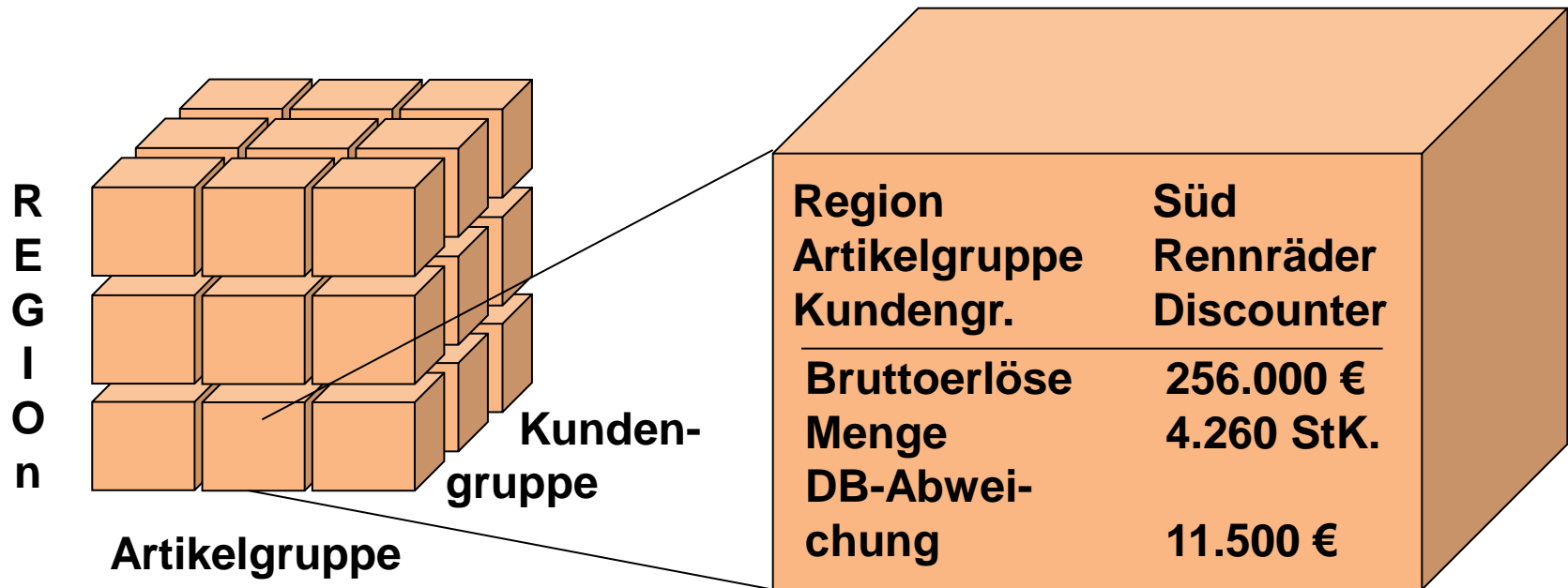


OLAP Client



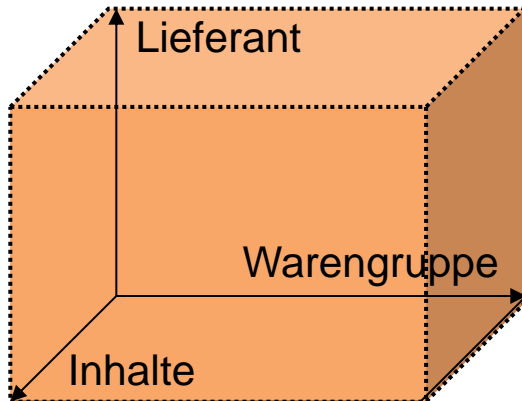
Präsentations-Schicht

Beispiel eines OLAP-Datenwürfels

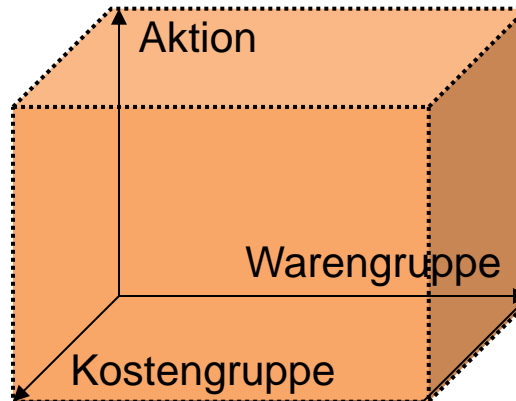


Multidimensionale Abfragen am Beispiel des Handels

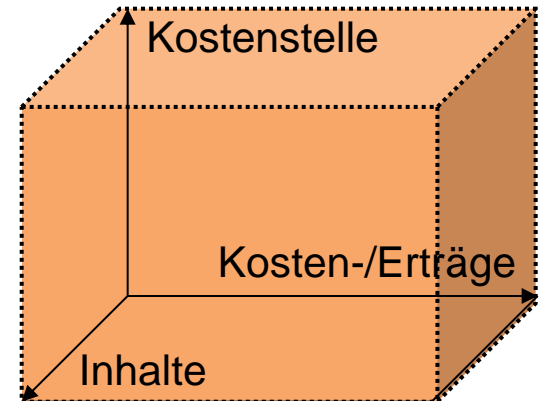
Lieferantencontrolling



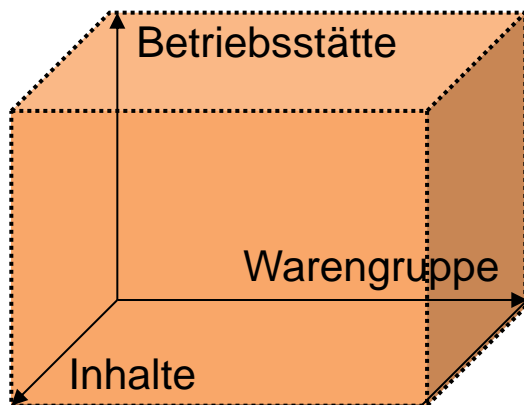
Aktionscontrolling



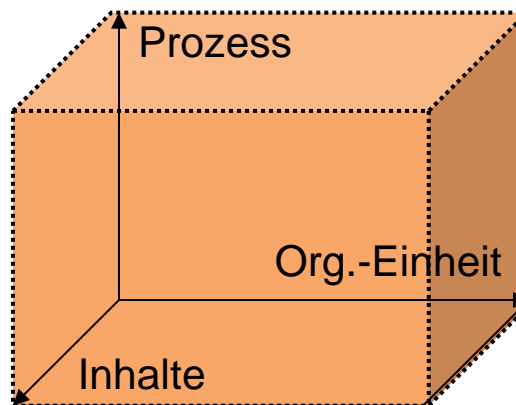
Kostenstellencontrolling



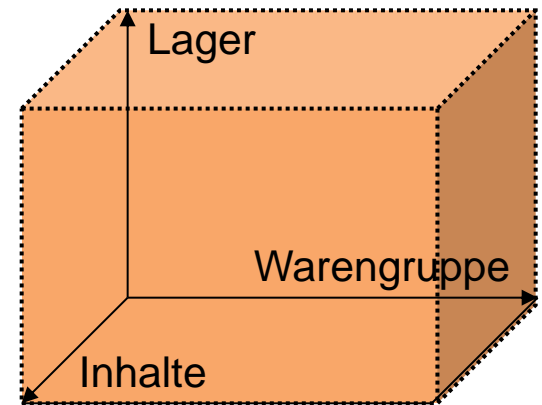
Vertriebscontrolling



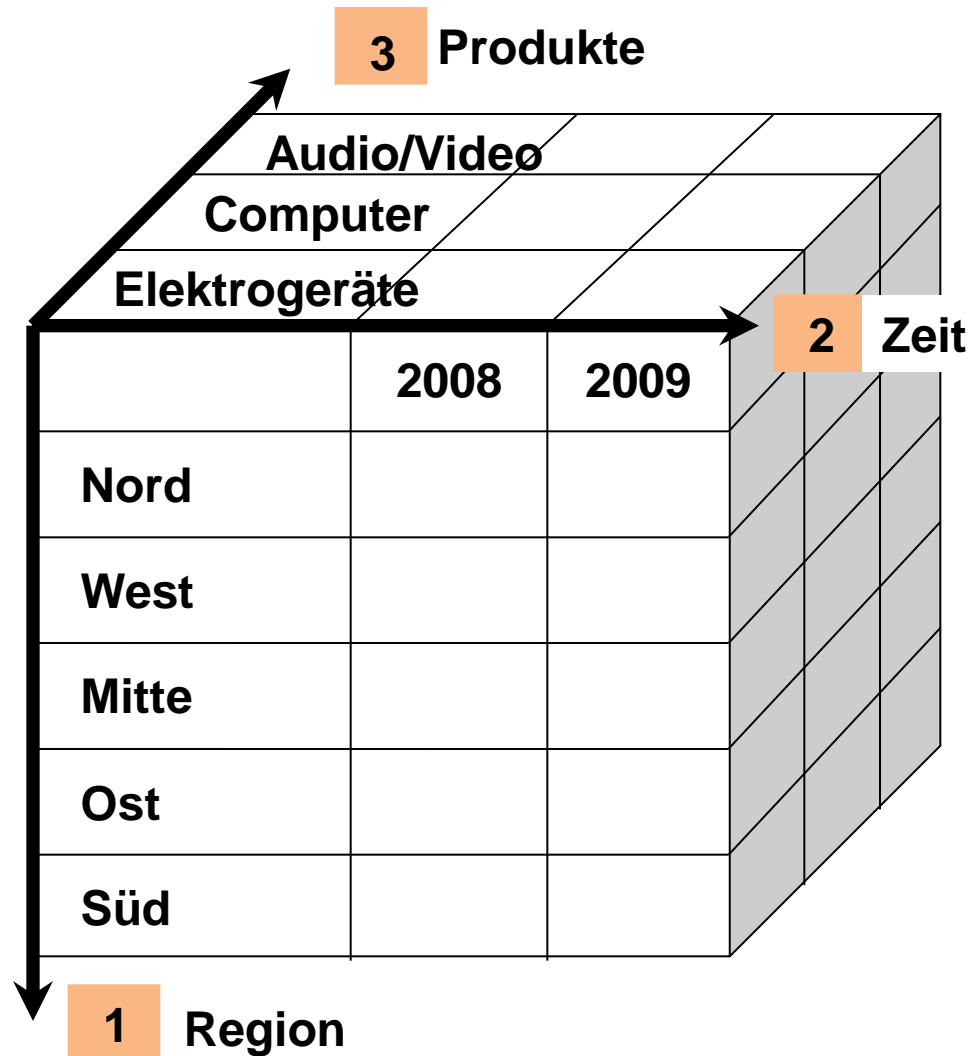
Prozesscontrolling



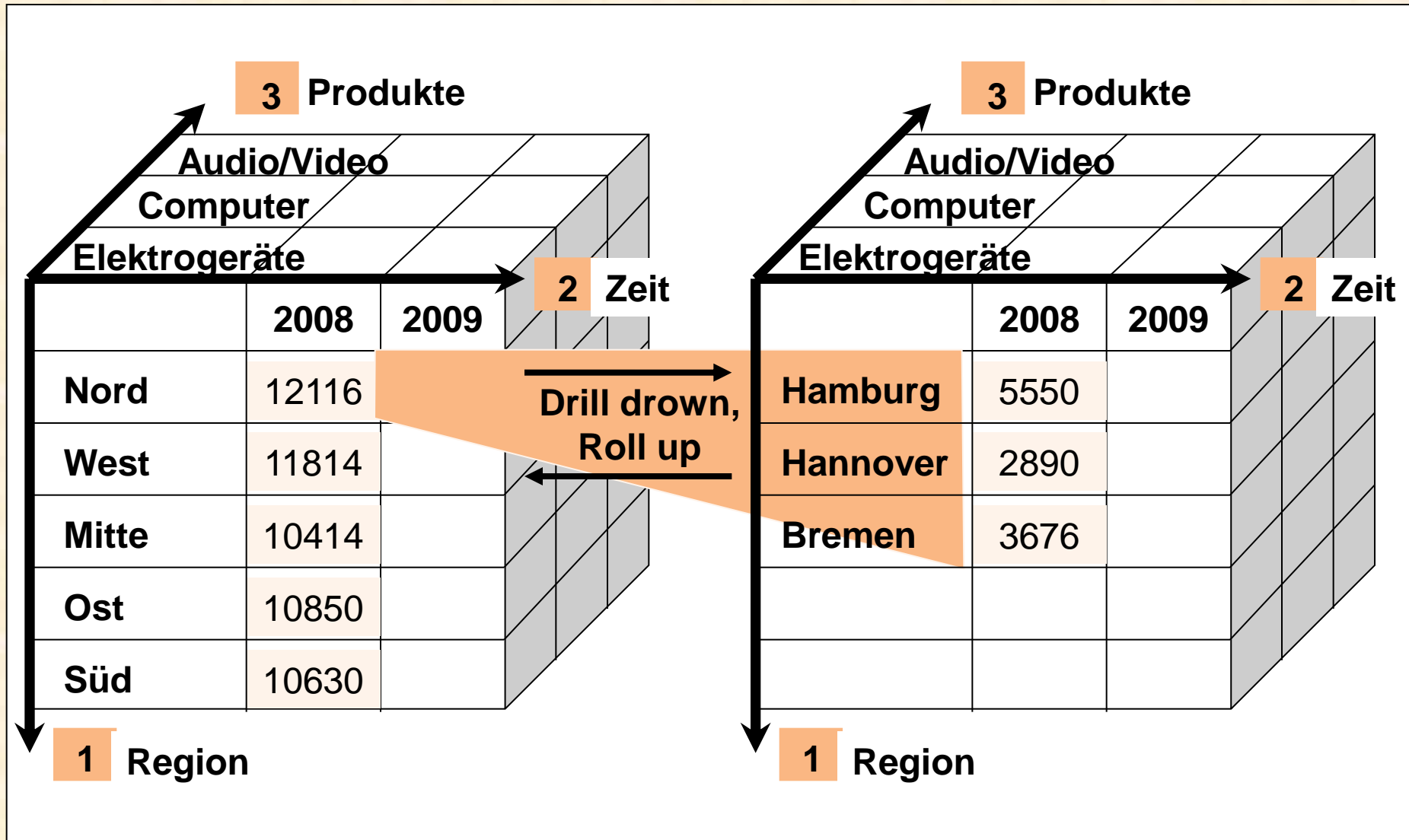
Logistikcontrolling



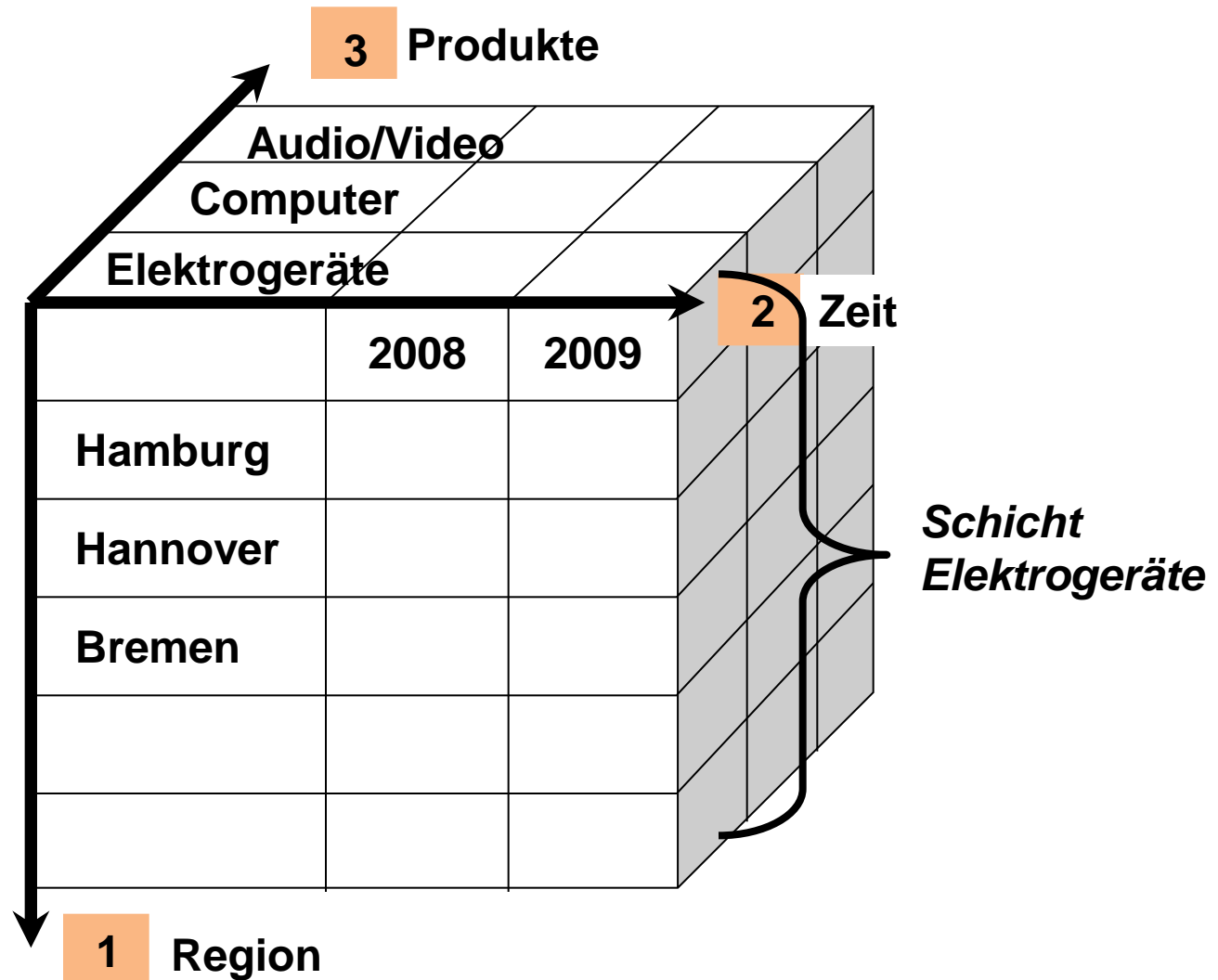
Würfel mit „dreidimensionalen“ Zellen



Drill-down und Roll-up



Slice

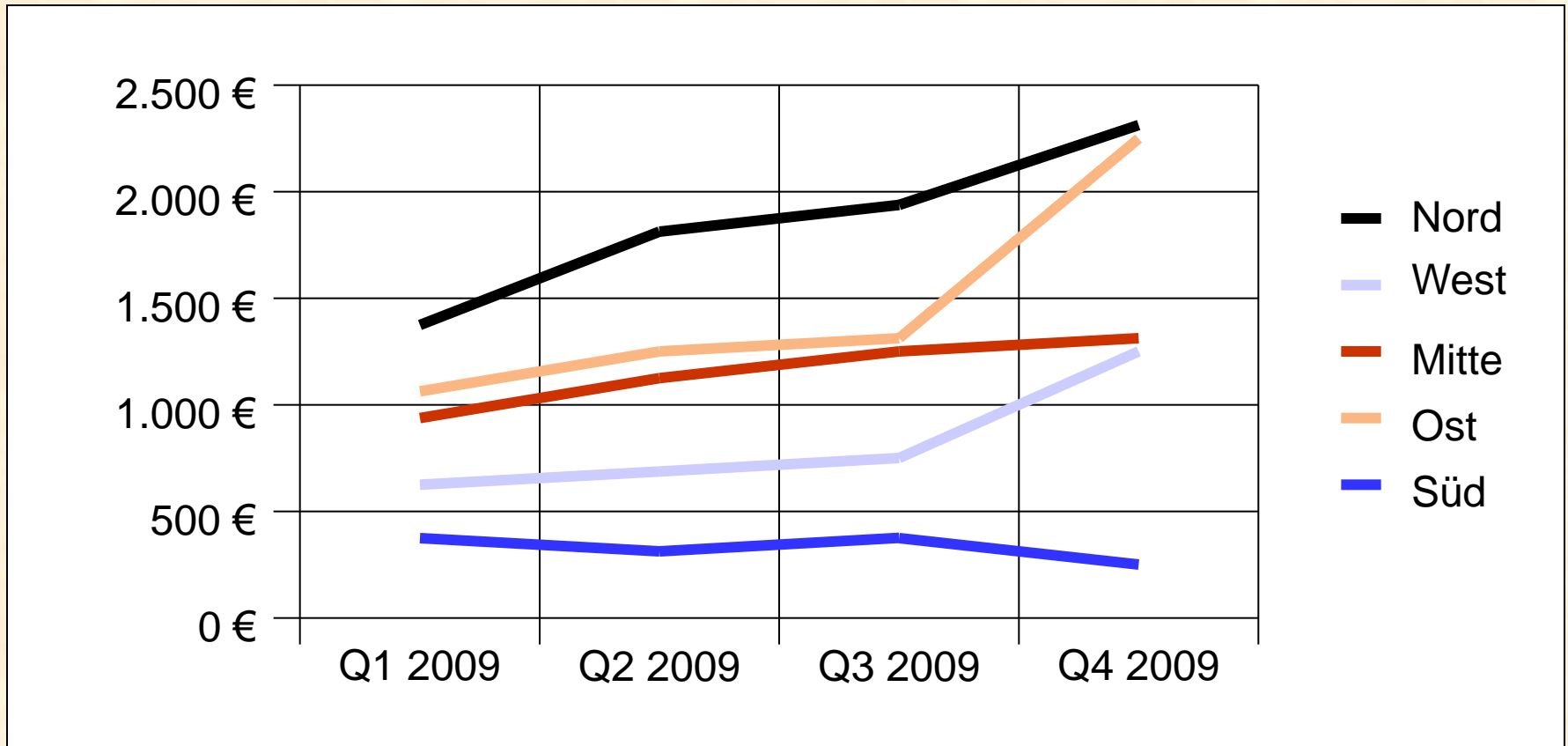


Darstellung als Matrix

Produktgruppe = Elektrogeräte

	Q1 2009	Q2 2009	Q3 2009	Q4 2009	SUMME
Nord	1.428 €	1.856 €	1.984 €	2.320 €	7.588 €
West	620 €	660 €	680 €	1.240 €	3.200 €
Mitte	996 €	1.110 €	1.218 €	1.282 €	4.606 €
Ost	1.294 €	1.382 €	1.436 €	2.214 €	6.326 €
Süd	410 €	362 €	420 €	240 €	1.432 €
SUMME	4.748 €	5.370 €	5.738 €	7.296 €	23.152 €

Darstellung als Diagramm



Auswahl der Daten für einen Drill-through

Produktgruppe = Elektrogeräte

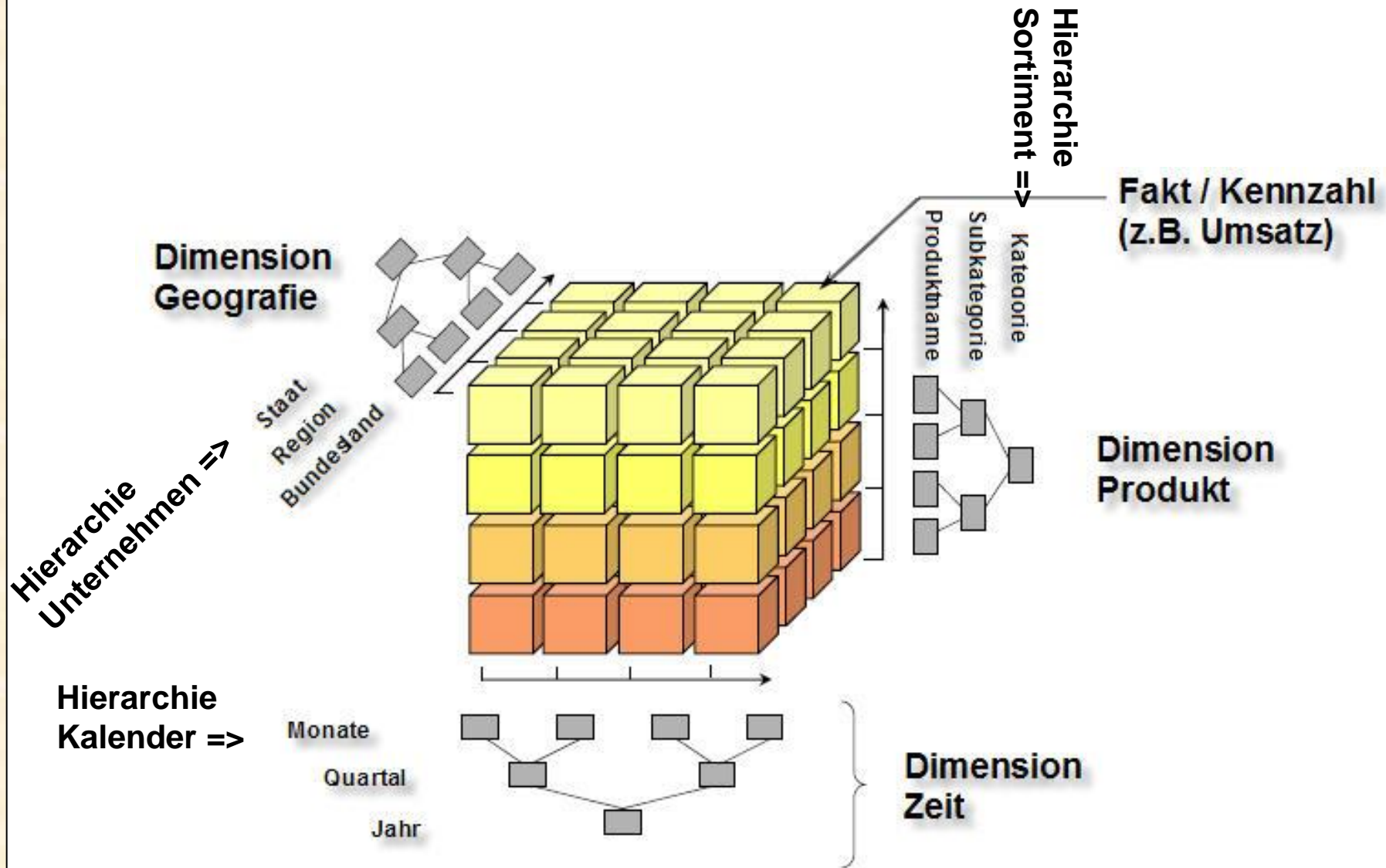
Vertriebsregion = Nord

	Q1 2009	Q2 2009	Q3 2009	Q4 2009	SUMME
...

Meier GmbH	39 €	34 €	45 €	152 €	270 €

SUMME	945 €	1.720 €	1.740 €	2.230 €	6.635 €

Beispielcube



MDX - Einführung

- Multidimensional Expressions
- Von Microsoft entwickelte Abfragesprache für OLAP Datenbanken
- Entwickelt sich zum Industriestandard
- Eher für IT-Entwickler bzw. als Abfragesprache für Applikationen, da relativ komplex

Beispielabfrage:

```
SELECT
{ [Measures].[Umsatzbetrag], [Measures].[Umsatzmenge] }
ON COLUMNS,
{ [Zeit].[Quartal].&[200601] , [Zeit].[Quartal].&[200602] }
ON ROWS
FROM [Umsatz]
WHERE {[Geografie].[Sachsen]}
```

	Umsatzbetrag	Umsatzmenge
1. Quartal	58.924,74 €	27032
2. Quartal	63.311,14 €	28923

MDX – Abgrenzung zu SQL

MDX	SQL
Abfragesprache für Datenbanken	
Microsoft	ANSI und ISO Standard
Abfrageschema basiert auf SELECT, FROM, WHERE	
Basis ist eine multidimensionale OLAP Datenbank (CUBE)	Basis ist eine relationale Datenbank
Versteht Hierarchien, Vorgänger / Nachfolger, Cousin, ... und kann Eigenschaften von Elementen, Zellen auslesen und definieren	
2 – n dimensionales Ergebnis, also Tabelle oder Cube	2 dimensionale Ergebnis, also Tabelle
Ähnliche Basisoperatoren und -funktionen	

MDX Abfrageschema

➤ Aufbau einer MDX Abfrage

SELECT

<Abfrageachse> ON COLUMNS,

<Abfrageachse> ON ROWS

FROM <Cube>

WHERE <Slicerachse>

➤ <Abfrageachse>

Menge aus denen die Daten abgerufen werden.

➤ <Cube>

Cube(s) der (die) abgefragt werden sollen.

➤ <Slicerachse>

Menge oder Tupel auf die die Ergebnismenge eingeschränkt wird.

Einfache Abfragen - Umsatzbetrag

```
SELECT
{ ([Geografie].[Bundesland].[Sachsen]),
  ([Geografie].[Bundesland].[Thüringen])
} ON COLUMNS,
{ ([Zeit].[Kalender].[Quartal].[1. Quartal]),
  ([Zeit].[Kalender].[Quartal].[2. Quartal]),
  ([Zeit].[Kalender].[Quartal].[3. Quartal]),
  ([Zeit].[Kalender].[Quartal].[4. Quartal])
} ON ROWS
FROM [Umsatz]
WHERE ([Measures].[Umsatzbetrag], [Zeit].[Jahr].[2006]);
```

	Sachsen	Thüringen
1. Quartal	58.924,74 €	28.719,44 €
2. Quartal	63.311,14 €	31.391,61 €
3. Quartal	63.603,99 €	31.698,82 €
4. Quartal	64.192,99 €	28.695,14 €

Einfache Abfragen - Umsatzmenge

```
SELECT
{ ([Geografie].[Bundesland].[Sachsen]),
  ([Geografie].[Bundesland].[Thüringen])
} ON COLUMNS,
{ ([Zeit].[Kalender].[Quartal].[1. Quartal]),
  ([Zeit].[Kalender].[Quartal].[2. Quartal]),
  ([Zeit].[Kalender].[Quartal].[3. Quartal]),
  ([Zeit].[Kalender].[Quartal].[4. Quartal])
} ON ROWS
FROM [Umsatz]
WHERE ([Measures].[Umsatzmenge],[Zeit].[Jahr].[2006]);
```

	Sachsen	Thüringen
1. Quartal	27032	13380
2. Quartal	28923	14775
3. Quartal	29172	14981
4. Quartal	29086	13208

Abfragen mittels CROSSJOIN

```
SELECT
{ ([Geografie].[Bundesland].[Sachsen]),
  ([Geografie].[Bundesland].[Thüringen])
} ON COLUMNS,
CROSSJOIN( { ([Zeit].[Kalender].[Quartal].[1. Quartal]),
              ([Zeit].[Kalender].[Quartal].[2. Quartal]),
              ([Zeit].[Kalender].[Quartal].[3. Quartal]),
              ([Zeit].[Kalender].[Quartal].[4. Quartal])
            },
            { ([Measures].[Umsatzbetrag]),
              ([Measures].[Umsatzmenge])
            }
          )
ON ROWS
FROM [Umsatz]
WHERE ([Zeit].[Jahr].&[2006]);
```

		Sachsen	Thüringen
1. Quartal	Umsatzbetrag	58.924,74 €	28.719,44 €
1. Quartal	Umsatzmenge	27032	13380
2. Quartal	Umsatzbetrag	63.311,14 €	31.391,61 €
2. Quartal	Umsatzmenge	28923	14775
3. Quartal	Umsatzbetrag	63.603,99 €	31.698,82 €
3. Quartal	Umsatzmenge	29172	14981
4. Quartal	Umsatzbetrag	64.192,99 €	28.695,14 €
4. Quartal	Umsatzmenge	29086	13208

Abfragen mit Funktion *Children*

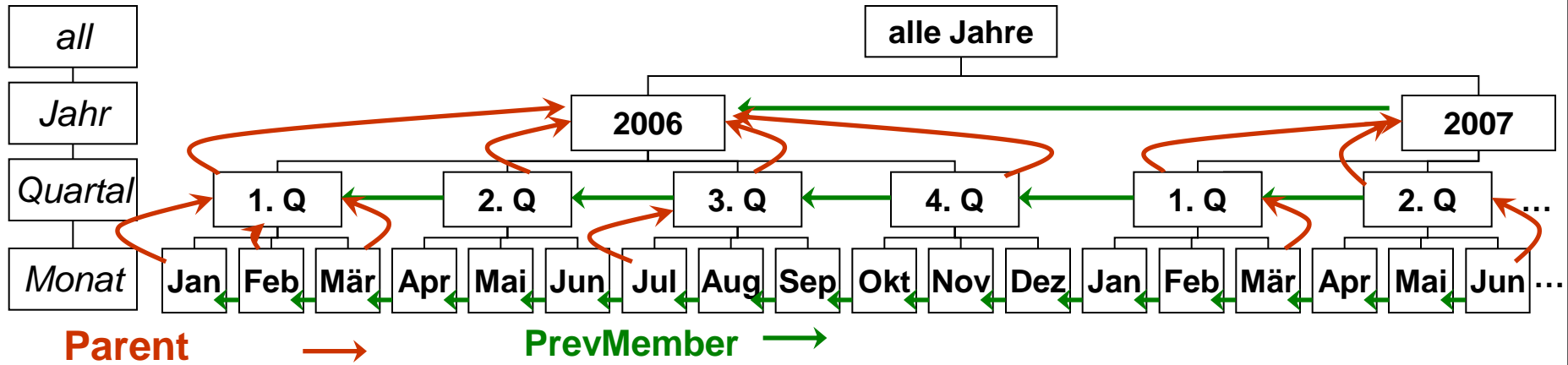
- Umsatzbetrag 2006 mit kürzerem MDX-Befehl -

```
SELECT
{ ([Geografie].[Bundesland].[Sachsen]),
  ([Geografie].[Bundesland].[Thüringen])
} ON COLUMNS,
{ ([Zeit].[Kalender].[Jahr].[2006].Children)
} ON ROWS
FROM [Umsatz]
WHERE ([Measures].[Umsatzbetrag]);
```

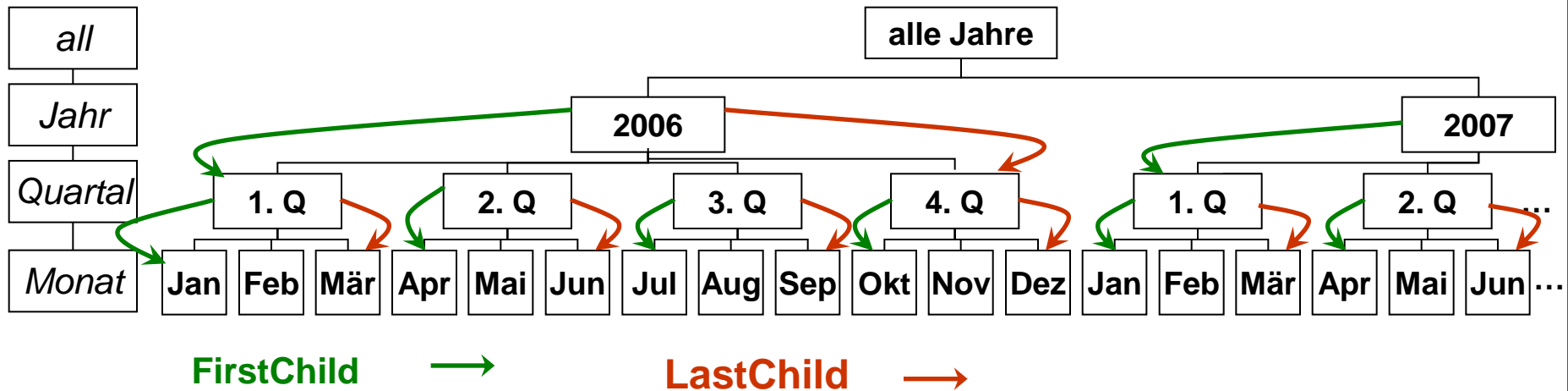
	Sachsen	Thüringen
1. Quartal	58.924,74 €	28.719,44 €
2. Quartal	63.311,14 €	31.391,61 €
3. Quartal	63.603,99 €	31.698,82 €
4. Quartal	64.192,99 €	28.695,14 €

Parent und PrevMember am Beispiel der Dimension Zeit

level: values:



level: values:



Fallunterscheidung mittels CASE-Anweisung

Variante 1:

```
CASE <Ausdruck>
  WHEN <Konstante 1>
    THEN <Wert 1>
  [WHEN <Konstante 2>
    THEN <Wert 2>]
  ...
  [ELSE <Werte für ELSE>]
END
```

Variante 2:

```
CASE
  WHEN <Bedingung 1>
    THEN <Wert1>
  [WHEN <Bedingung 2>
    THEN <Wert 2>]
  [ELSE <Werte für ELSE>]
END
```

Beispiel:

```
CASE
  WHEN IsEmpty ([Zeit].[Kalender].PrevMember)      //Wert ist NULL
    THEN 0
  WHEN [Measure].[Umsatzbetrag] >
    ([Zeit].[Kalender].PrevMember, [Measure].[Umsatzbetrag])
    THEN 1
  ELSE -1
END
```

Abfragen mit Funktion *Children*

- Umsatzbetrag Brotprodukte in Deutschland, 2006 -

```

SELECT
  { [Zeit].[Kalender].[Jahr].[2006].Children
  } ON COLUMNS,
  { [Produkt].[Sortiment].[Subkategorie].[Brot].Children
  } ON ROWS
FROM [Umsatz]
WHERE ([Measures].[Umsatzbetrag],
       [Geografie].[Staat].[Deutschland])

```

	1. Quartal	2. Quartal	3. Quartal	4. Quartal
Colonial - Weißbrot	1.027,88 €	1.105,16 €	711,30 €	1.013,07 €
Colonial - Vollkornbrot	1.233,89 €	1.445,18 €	947,14 €	1.151,03 €
Colonial - Roggenbrot	2.106,25 €	2.450,42 €	2.094,89 €	2.202,41 €
Colonial - Pumpenmickel Brot	1.298,20 €	1.491,27 €	1.396,27 €	1.387,69 €
Friedrichs - Vollkornbrot	956,48 €	950,62 €	850,06 €	1.149,00 €
Friedrichs - Weißbrot	544,28 €	584,34 €	447,96 €	519,26 €
Friedrichs - Roggenbrot	1.044,60 €	1.272,41 €	1.042,07 €	1.285,10 €
...				
Wendels Bestes - Vollkornbrot	431,82 €	517,48 €	429,17 €	535,36 €
Wendels Bestes - Roggenbrot	3.182,28 €	3.286,31 €	2.351,81 €	2.251,32 €

Abfragen mit den Funktionen *Children* und *Members* und CROSSJOIN

```
SELECT
{ ([Geografie].[Bundesland].[Sachsen]),
  ([Geografie].[Bundesland].[Thüringen])
} ON COLUMNS,
  CROSSJOIN (
    { ([Zeit].[Jahr].Children) },
    { ([Zeit].[Quartal].Members) }
  )
ON ROWS
FROM [Umsatz]
WHERE ([Measures].[Umsatzbetrag]) ;
```

		Sachsen	Thüringen
2006	All	250.032,86 €	120.505,01 €
2006	1. Quartal	58.924,74 €	28.719,44 €
2006	2. Quartal	63.311,14 €	31.391,61 €
2006	3. Quartal	63.603,99 €	31.698,82 €
2006	4. Quartal	64.192,99 €	28.695,14 €
2007	All	293.994,40 €	149.099,76 €
2007	1. Quartal	70.622,74 €	35.431,61 €
2007	2. Quartal	76.993,17 €	40.030,19 €
2007	3. Quartal	73.204,88 €	35.361,47 €
2007	4. Quartal	73.173,61 €	38.276,49 €

Abfrage mit Alias in *WITH MEMBER* (1)

Für welche Produktkategorien wurden die Planumsätze in den Quartalen nicht erfüllt?

```
WITH MEMBER [Measures].[Umsatz_Plan Differenz]
  AS '[Measures].[Umsatzbetrag] - [Measures].[Umsatzplan]'
  , FORMAT_STRING = '#.00'
SELECT
  { [Zeit].[Jahr].Children * [Zeit].[Quartal].Members }
ON COLUMNS,
  { [Produkt].[Sortiment].[Kategorie] }
ON ROWS
FROM Umsatz
WHERE [Measures].[Umsatz_Plan Differenz]
```

	2006	2006	2006	2006	2006	2007	2007	2007	2007	2007
	All	1. Quartal	2. Quartal	3. Quartal	4. Quartal	All	1. Quartal	2. Quartal	3. Quartal	4. Quartal
Backwaren	39419,79	6119,16	16671,36	8525,74	8103,53	47786,68	8922,27	20935,59	9905,32	8023,50
Milchprodukte	5280,59	-1458,60	-1249,98	1854,23	6134,94	69006,97	13150,08	15930,26	14930,95	24995,68
Fleisch	101081,92	23698,85	22396,24	33616,85	21369,98	120423,42	29896,60	32345,46	33474,89	24706,47

Abfrage mit Alias in *WITH MEMBER* (2)

Wie groß ist der Anteil der Umsätze der Subkategorie an dem Umsatz der jeweiligen übergeordneten Kategorie und am Gesamtumsatz?

```
WITH
    MEMBER [Measures].[Anteil Subkategorie am Gesamtumsatz]
        AS '([Measures].[Umsatzbetrag] * 100) / ([Measures].[Umsatzbetrag],
[Produkt].[Kategorie].[All])'
        , FORMAT_STRING='###.00\%'
    MEMBER [Measures].[Anteil der Subkategorie an deren Kategorie]
        AS '([Measures].[Umsatzbetrag] * 100) / ([Measures].[Umsatzbetrag],
[Produkt].[Kategorie])'
        , FORMAT_STRING='###.00\%'
SELECT
    { [Measures].[Umsatzbetrag], [Measures].[Anteil Subkategorie am
Gesamtumsatz],
    [Measures].[Anteil der Subkategorie an deren Kategorie] } on
columns,
    { NONEMPTY([Produkt].[Kategorie].Children *
[Produkt].[Subkategorie].Children) } on rows
FROM [Umsatz]
```

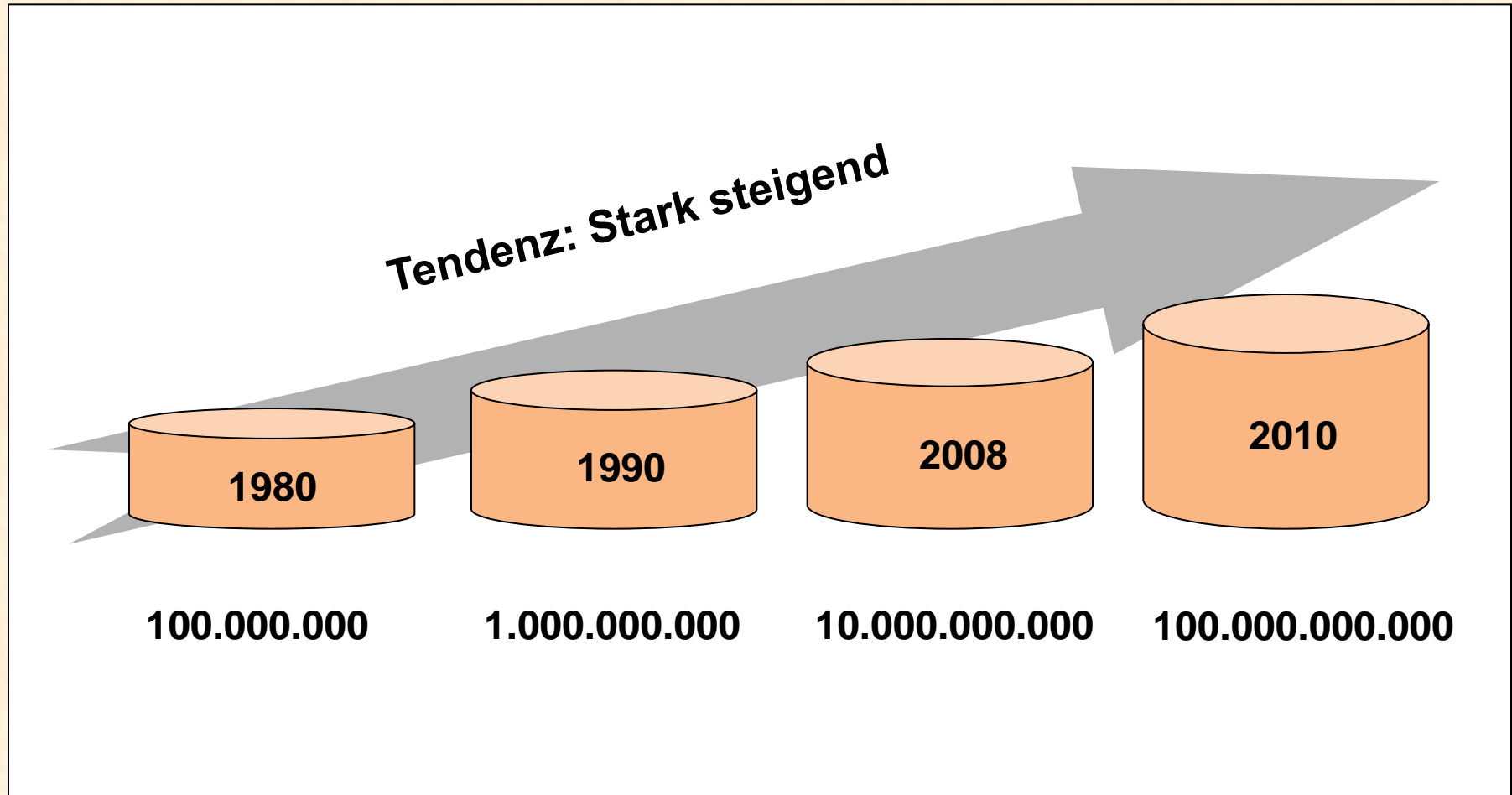
		Umsatzbetrag	Anteil Subkategorie am Gesamtumsatz	Anteil der Subkategorie an deren Kategorie
Backwaren	Bagels	1.807.831,59 €	19,81%	71,72%
Backwaren	Muffins	265.937,09 €	2,91%	10,55%
Backwaren	Brot	446.957,79 €	4,90%	17,73%
Milchprodukte	Käse	1.064.149,59 €	11,66%	29,54%

...

Abgrenzung zu operativen Systemen

Kriterium	OLTP-System	DWH-System
Anfragearten	Lesen, Schreiben, Ändern, Löschen	Lesen, periodisches Hinzufügen
Transaktionsdauer und typ	kurze Lese- und Schreibtransaktionen	lange Lesetransaktionen
Anfragestruktur	einfach strukturiert	komplex
Datenvolumen je Anfrage	wenige Datensätze	viele Datensätze
Datenmodell	anfragebezogen	analysebezogen
Datenquelle	meist eine	mehrere
Eigenschaften der Daten	nicht abgeleitet, zeitpunkt-bezogen, autonom, dynamisch	abgeleitet, konsolidiert, zeitraum-bezogen, integriert, stabil
Datenvolumen	MByte ... GByte	GByte ... TByte
Zugriffsart	Einzel tupelzugriff	Tabellenzugriff
Anwendertyp	Ein- und Ausgabe durch Angestellte oder Anwendungssoftware	Manager, Controller, Analyst
Anwenderzahl	sehr viele	wenige (bis einige hundert)
Antwortzeit	ms ... sec	sec ... min

VLDB - Very Large Database



Weitere Begriffe beim Umgang mit DWH

➤ **Data Warehousing:**

- Data Warehouse - Prozess
- alle Schritte der Datenbeschaffung (Extraktion, Transformation, Laden), des Speichern bis zur Analyse

➤ **Data Mart:**

- externe Teilsicht auf das Data Warehouse
- i. A. erzeugt durch Kopie
- spezifisch je Anwendungsbereich

➤ **OLAP (Online Analytical Processing)**

- interaktive Analyse der Daten

➤ **Metadatensystem (Business Data Dictionary, Repository))**

- Ablage der Hintergrundinformationen (Datenquellen, Transformationen, Verdichtungen, ...)

Datenquellen eines Data Warehouses

Klassifikation

- **Herkunft:**
intern, extern
- **Zeit:**
aktuell, historisch
- **Nutzungsebene:**
Primärdaten, Metadaten
- **Inhalt:**
Zahl, Zeichenkette, Referenz, Dokument
- **Darstellung:**
numerisch, alphanumerisch, BLOB
- **Sprache:**
Zeichensatz, Währung
- **Vertraulichkeitsgrad:**
offen, vertraulich

Qualitätsanforderungen

- **Konsistenz**
(Widerspruchsfreiheit)
- **Korrektheit:**
(Übereinstimmung mit der Realität)
- **Vollständigkeit**
(Abwesenheit fehlender Werte)
- **Genauigkeit**
(z.B. Anzahl der Nachkommastellen)
- **Granularität**
(z.B. tagesgenaue Daten)
- **Zuverlässigkeit**
(Nachvollziehbarkeit der Entstehung, Vertrauenswürdigkeit der Quelle)
- **Verständlichkeit**
(inhaltlich und strukturell passend für die Zielgruppe)
- **Relevanz**
(Zweckdienlichkeit)

Arbeitsbereiche und Extraktionskomponente

Arbeitsbereiche

➤ Aufgabe:

- Zentrale Datenkomponente des Beschaffungsbereichs (engl.: staging area)
- temporärer Zwischenspeicher zur Integration

➤ Nutzung:

- Ausführung von Transformationen (Bereinigen, Integrieren,...) direkt im Arbeitsbereich
- Laden der Daten ins DWH erst nach abgeschlossener Transformation

➤ Vorteile

- keine Beeinflussung von Daten in Quellen und im DWH
- keine Übernahme fehlender Daten

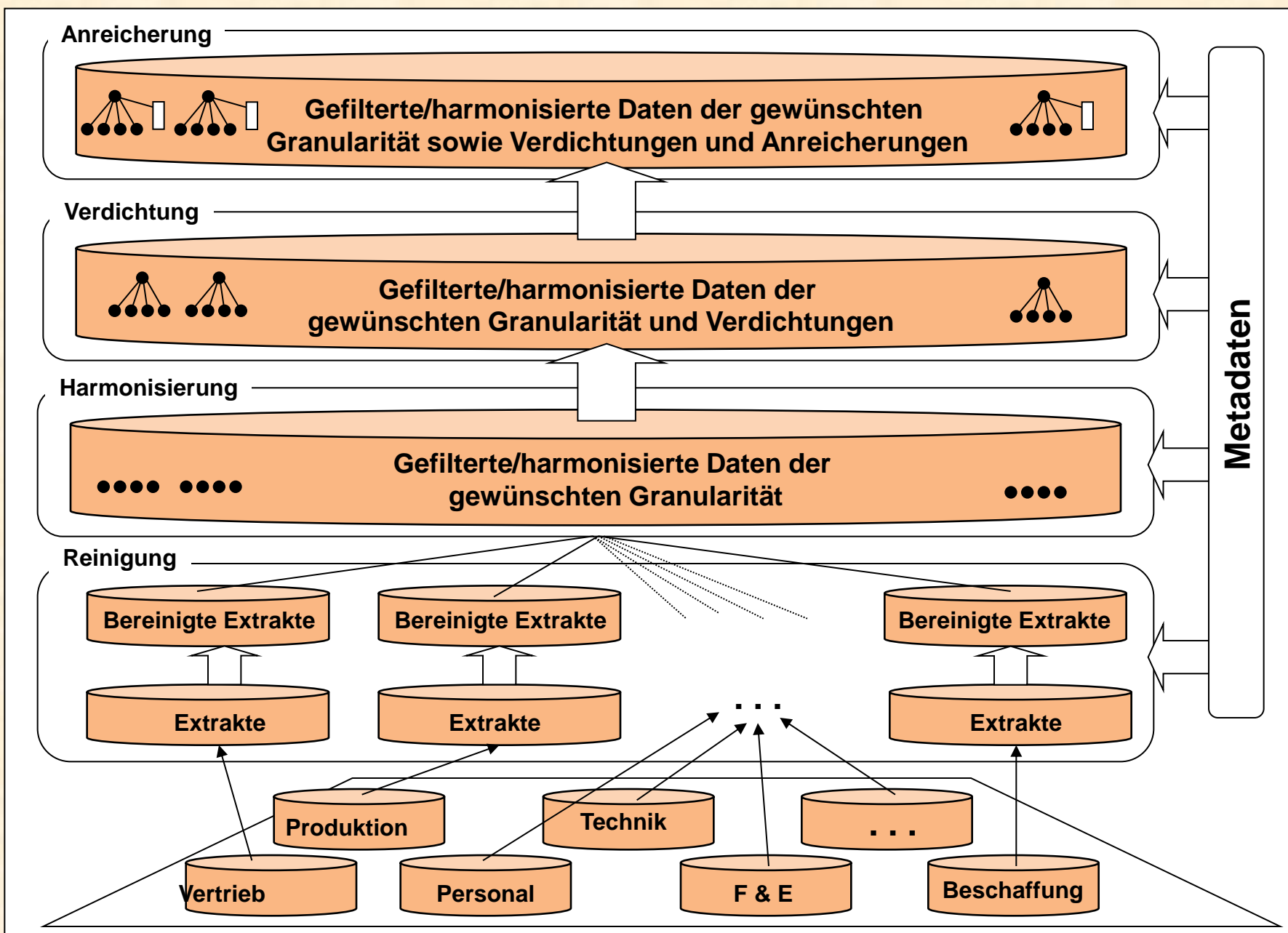
Extraktionskomponente

➤ Aufgabe:

- Übertragung von Daten aus Quellen in den Arbeitsbereich

➤ Funktion: *abhängig von der Monitor-Strategie*

- periodisch
- auf Anfrage
- Ereignis-gesteuert (z.B. nach Erreichen einer festgesetzten Anzahl an Änderungen in den Quelldaten)
- sofortige Extraktion



Laden von Daten ins DWH

➤ **Aufgabe:**

- Übertragen der bereinigten und aufbereiteten (z.B. aggregierten) Daten in das Data Warehouse

➤ **Besonderheiten:**

- i.A. Verwendung spezieller Ladewerkzeuge (z.B. SQL*Loader von Oracle)
- Anwendung von Bulk-Laden
- Historisierung: kein Überschreiben von Daten im DWH bei Änderungen in den Quelldaten, sondern zusätzliches Abspeichern

➤ **Ladevorgang**

- online: Quelldatenbank und DWH stehen weiterhin zur Verfügung
- offline: Quelldatenbank und DWH stehen nicht zur Verfügung (i.A. Verwendung von Zeitfenstern mit Schwachlast, z.B. nachts oder an Wochenenden)

Data Warehouse

Chancen

Risiken

Strategie und Wettbewerb

- Informationsvermarktung
- Wettbewerbsvorteile durch verbesserte Entscheidungsqualität

- (Informationsvermarktung:) Gefahr, wettbewerbskritische Daten und Erfolgsfaktoren offenlegen

Qualität

- verbesserte Datenqualität und -konsistenz
- höhere Reporting-/Analysequalität
- verbesserte Entscheidungsqualität

- Gefahr von Inkonsistenzen aufgrund zusätzlicher Datenredundanz bei mehreren Data Marts
- Gefahr unzureichender Datenaktualität bei niedriger Datenübernahmefrequenz

Flexibilität

- umfassende, integrierte Informationserschließung
- schnellere Verfügbarkeit von Auswertungen

Data Warehouse

Chancen

- Leistungssteigerungen im operativen Betrieb durch Entlastung der operativen Systeme
- Wegfall personeller Analysetätigkeiten

Risiken

- Zeitaufwand für Wartung und Pflege

Produktivität

Kosten

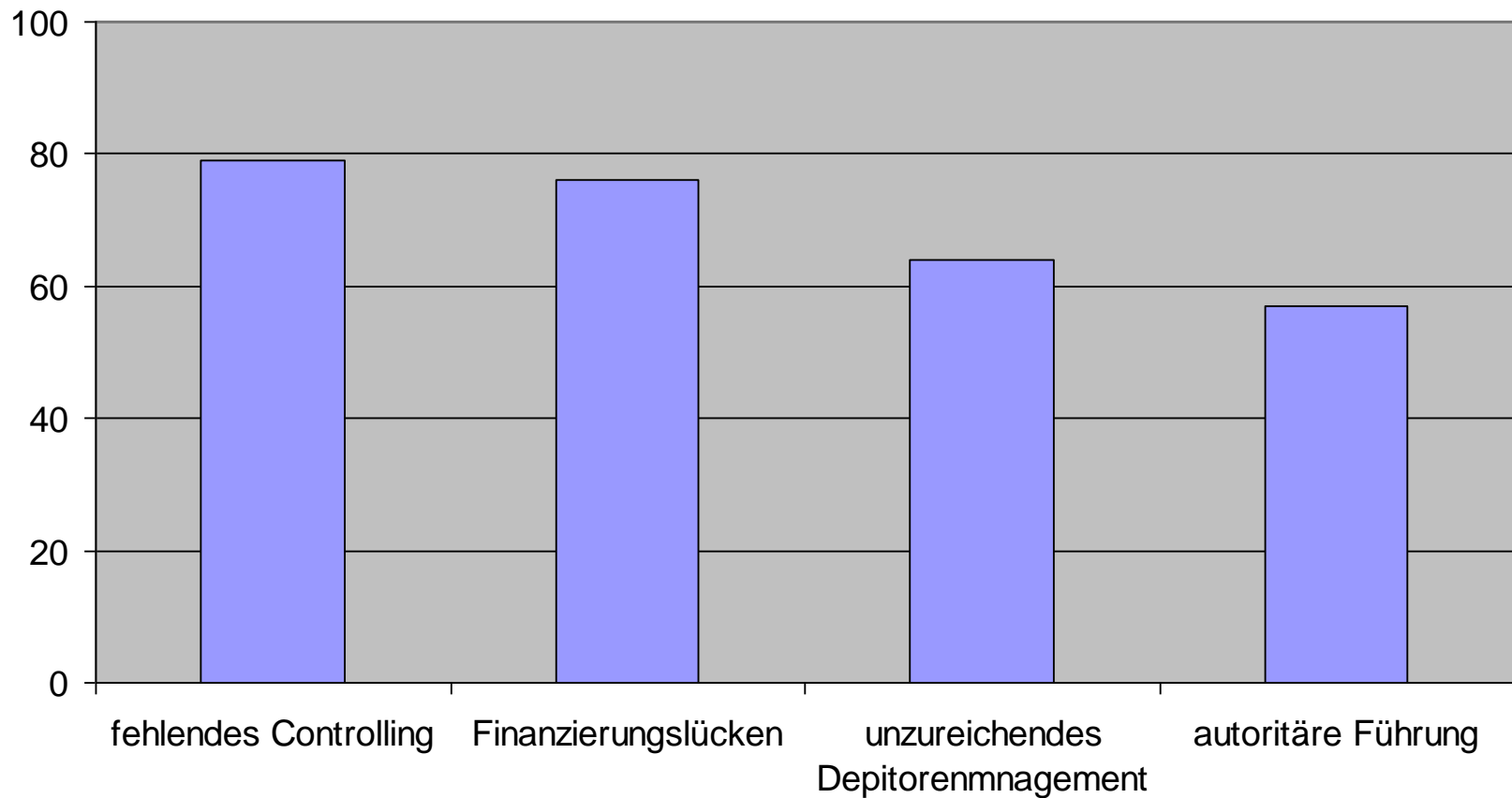
- Personalkosteneinsparung durch
 - Reduzierung des Personalbedarfs für Reporting-/Analysetätigkeiten
 - Ablösung herkömmlicher MIS
- Sachkosteneinsparung

- hohes Investitionsvolumen (Kapitalbindung)
- hohe laufende Kosten, v.a. für Datenaktualisierung

Mitarbeiter

- Steigerung der Mitarbeiterzufriedenheit
 - durch Einsatz moderner Technologie
 - durch weniger Datensuche und Aufbereitungstätigkeiten

Hauptgründe für das Scheitern von Unternehmen



[nach: Insolvenzen, Neugründungen, Löschungen Jahr 2007; Creditreform Wirtschaftsforschung]



u.a. ist BI die Basis für ein erfolgreiches Controlling

Ziele der langfristigen Unternehmensexistenz

Angemessene Verzinsung des investierten Kapitals
der Eigentümer in Form von Gewinnen,
Dividenden und
Eigenkapitalwertsteigerungen



Ergebnis, Rentabilität



Controlling

**Jederzeitige Fähigkeit, anstehende
Zahlungsverpflichtungen erfüllen zu
können**

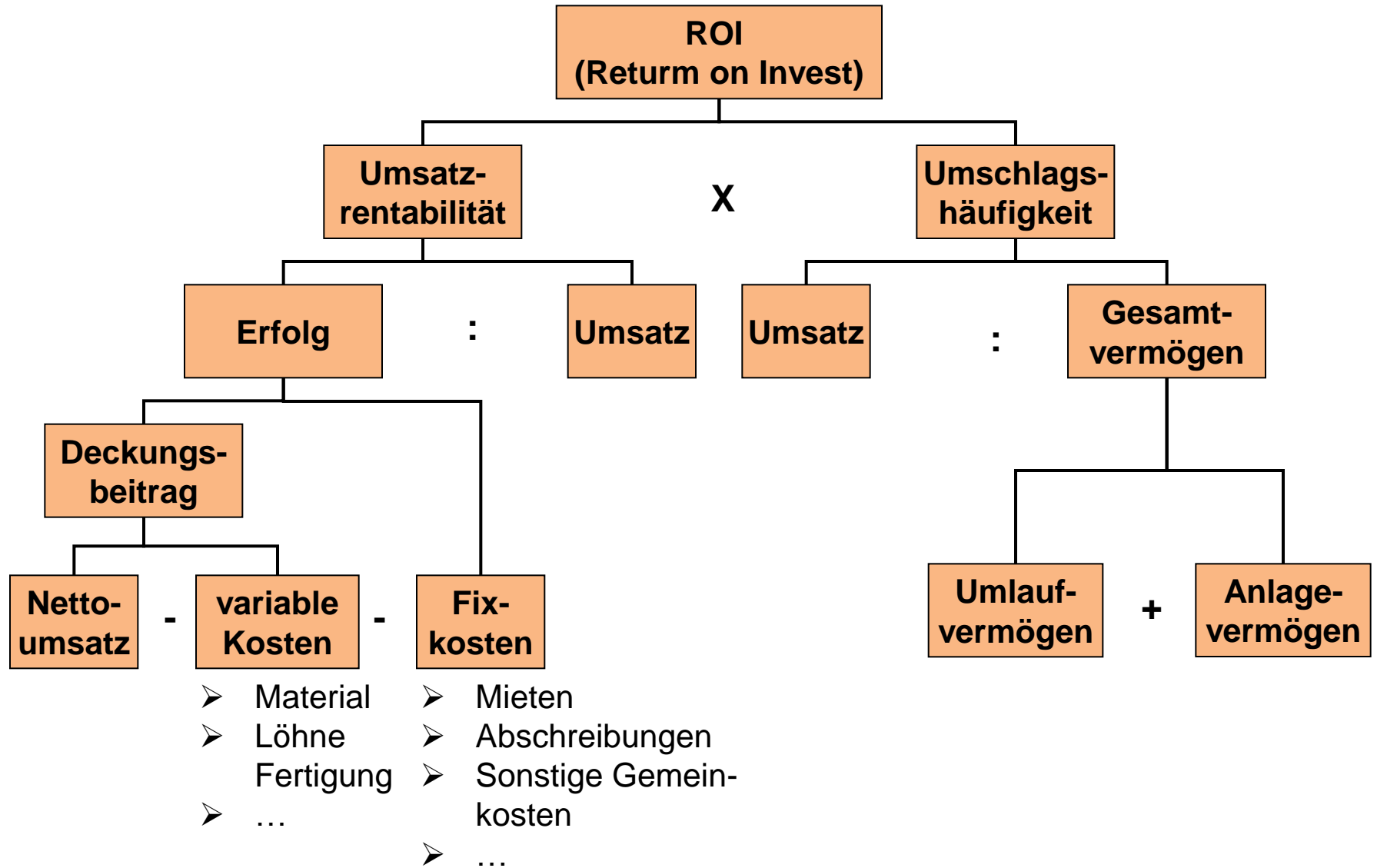


Liquidität



Finanzierung

DuPont-Kennzahlensystem

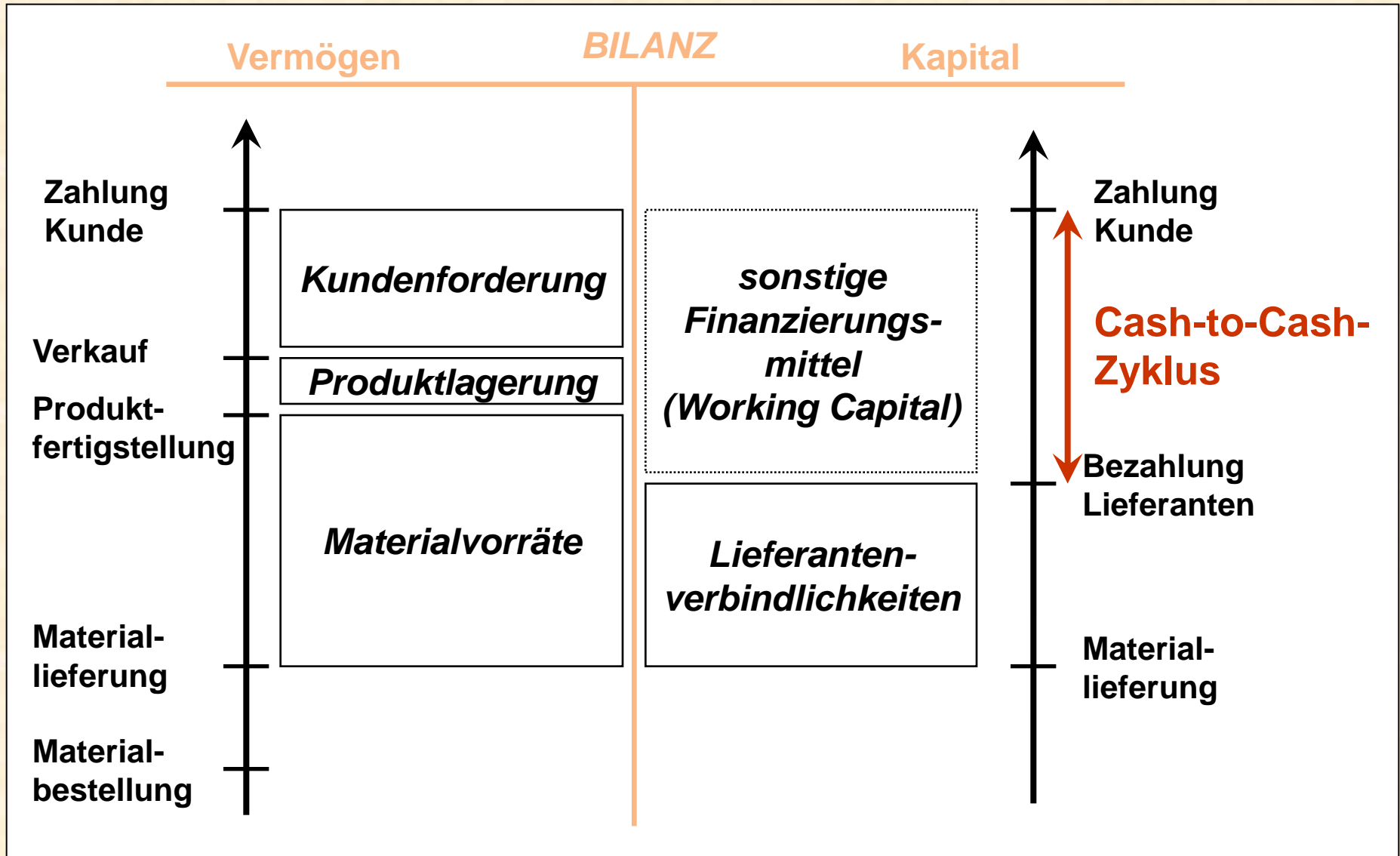


Branchenspezifik von Umsatzrendite und Umschlagshäufigkeit

Branche	Umsatzrendite	Umschlagshäufigkeit
Ernährungsindustrie	5,4 %	2,11
Großhandel	3,2 %	2,95
Papierindustrie	7,0 %	2,05
Kunststoffindustrie	6,2 %	1,98
Maschinenbau	6,1 %	1,34
Baugewerbe	5,7 %	1,21
Chemie	7,9 %	1,38
Elektrotechnik	6,9 %	1,26

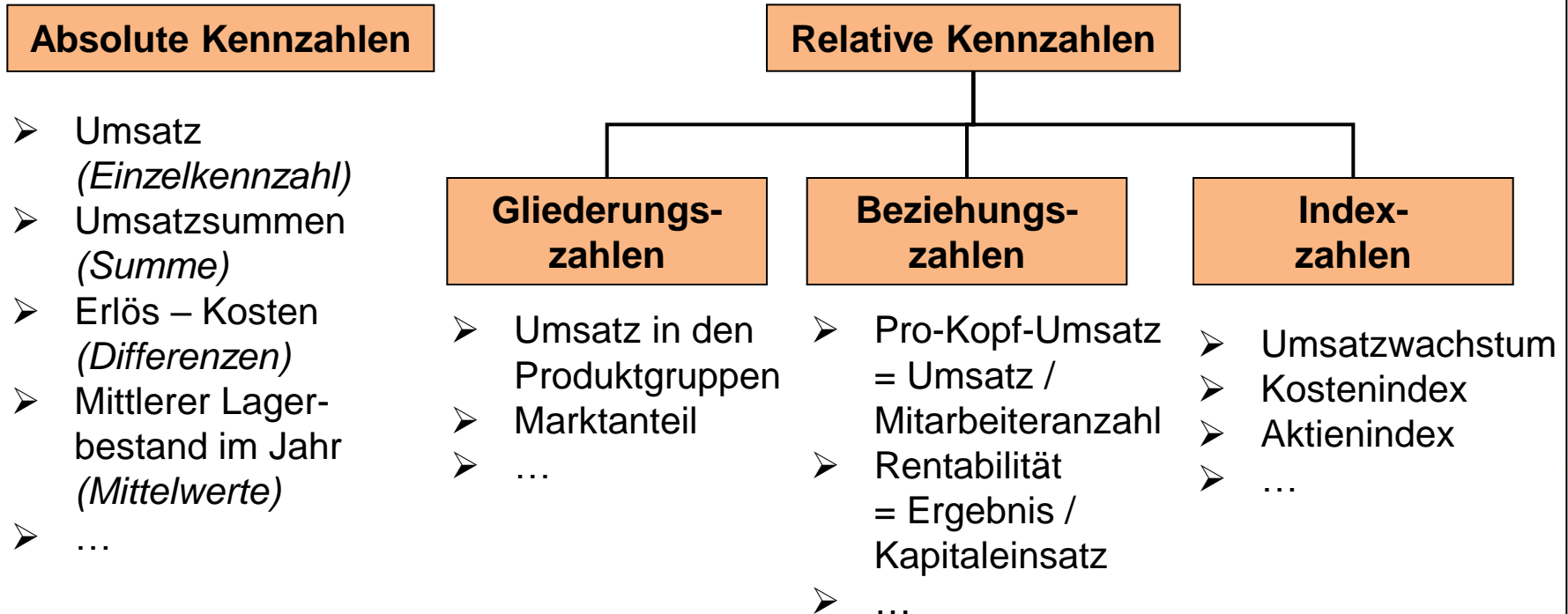
[nach Coenenberg 1997]

Cash-to-Cash-Zyklus



Arten an Kennzahlen aus betriebswirtschaftlicher Sicht

statistisch / methodisch:



zeitlich:

Ist-Kennzahlen

Plan-Kennzahlen

inhaltlich:

Wertkennzahl

Mengenkennzahl

Perspektiven der Balanced Scorecard

Finanzwirtschaftliche Perspektive

Ziele	Leistungsmaßstäbe
Erfolg	Umsatzrentabilität, ROI
Ertragswachstum	Umsatzwachstum/Zielsegment
Kostenreduzierung	% indirekte Kosten am Gesamt
Liquidität	Cash to Cash - Cycle

Kundenperspektive

Ziele	Leistungsmaßstäbe
Kundenrentabilität	Nettoerfolg je Kunde
Marktanteil	% Umsatz im Zielsegment
Kundenaquisition	% Umsatz mit Neukunden
Kundentreue	Umsatzsteigerung Altkunden
Kundenzufriedenheit	Zufriedenheitsindex aus Umfrage
Qualität, Zeit	Fehlerquote, Reparaturzeit

Interne Prozessperspektive

Ziele	Leistungsmaßstäbe
Innovationsprozess - Effektivität, - Zykluszeiten	Umsatz neuer Produkte, Time to Market, BreakEvenTime
Betriebszyklus: - Zykluszeiten, - Effizienz, - Qualität	Durchlaufzeiten, Kosten der Prozesse, Fehlerquote, first pass yields

Lern – und Entwicklungsperspektive

Ziele	Leistungsmaßstäbe
Mitarbeiterproduktivität	Wertschöpfung je Mitarbeiter
Mitarbeitertreue	Fluktuationsquote
Weiterbildung	Aufgabendeckungsziffer
Informationssystem	Informationsdeckungsziffer

Beispiel einer Ursache-Wirkungskette

Finanzielle Perspektive:

Was wollen wir unseren Kapitalgebern bieten?

Kundenperspektive:

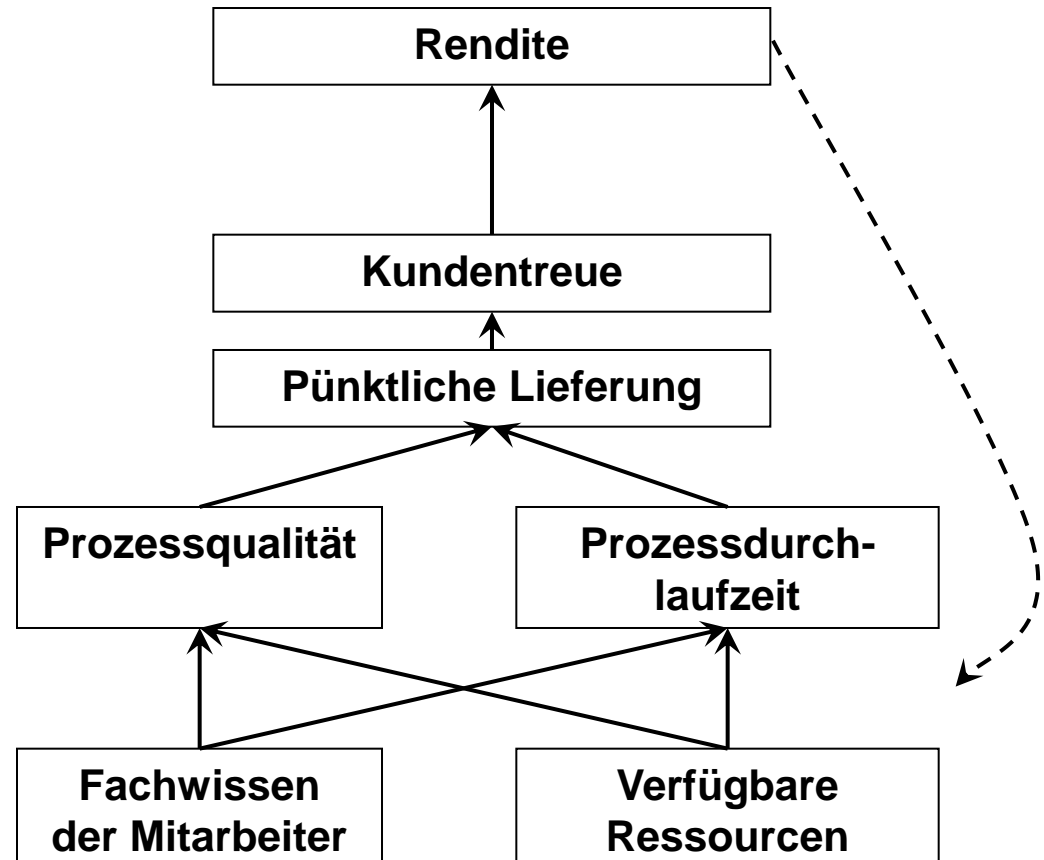
Wie sollen uns unsere Kunden wahrnehmen?

Prozessperspektive:

Bei welchen Prozessen müssen wir Hervorragendes leisten)

Entwicklungsperspektive:

Wie gewährleisten wir langfristig unseren Erfolg?



Entwicklungsschritte einer Balanced Scorecard

