

---

# Data Warehousing und Data Mining




## Eine Einführung in entscheidungsunterstützende Systeme

Interaktive Folien zu Kapitel 5  
*Data Warehouse*

---

# Inhalt

---

Grundlagen	<a href="#"><u>2</u></a>
Entwicklung - Überblick	<a href="#"><u>3</u></a>
Data Marts und Enterprise Data Warehouse	<a href="#"><u>6</u></a>
Mehrdimensionale Daten	<a href="#"><u>21</u></a>
Endbenutzerzugriff	<a href="#"><u>34</u></a>
OLAP	<a href="#"><u>44</u></a>
ANLAGEBERATUNG mit <i>PowerPlay</i> 	<a href="#"><u>71</u></a>
Modellierung relationaler Data Warehouses	<a href="#"><u>84</u></a>
Modellierung von Informationssystemen	<a href="#"><u>85</u></a>
Sternschema ( LIEFERFRIST )	<a href="#"><u>97</u></a>
EDW-Modell und Sternschema ( VERKAUF )	<a href="#"><u>106</u></a>
<i>Synchrony</i> ( EINZELHANDEL  )	<a href="#"><u>126</u></a>
Entwicklung und Betrieb	<a href="#"><u>136</u></a>
Fallbeispiel ( ZEITSCHRIFTEN  )	<a href="#"><u>151</u></a>
Effizienz	<a href="#"><u>160</u></a>
Netzzugriff	<a href="#"><u>198</u></a>
Client/Server-Architekturen	<a href="#"><u>203</u></a>
Zugriff über Internet und Intranet	<a href="#"><u>213</u></a>

# Entwicklung

## Operative Daten verwalten

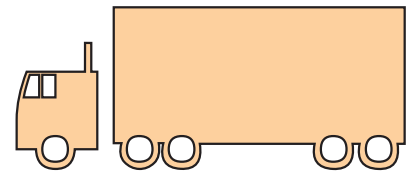
- *Bestellungen*
- *Rechnungen*
- *Produkte*
- ...



## Daten laden

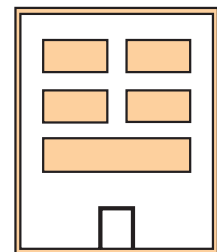
*Operative Daten ...*

- *zeitlich synchron auswählen*
- *temporär speichern*
- *transformieren*
- *integrieren*
- ...



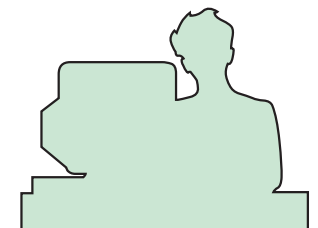
## Analytische Daten verwalten

- *Fakten (Indikatoren) wie Geldumsatz speichern*
- *Dimensionen wie Zeit und Gebiet speichern*
- *Speicheraufwand und Laufzeit optimieren*
- *Daten sichern*
- *Vorberechnen, insbesondere aggregieren*
- *Nicht mehr benötigte Daten archivieren*
- ...



## Zentrale Daten an lokale Systeme weiterleiten

- *lokale Data Warehouses*
- *Tabellenkalkulationswerkzeuge*
- *lokale Datenbanksysteme*
- ...



## Endbenutzer-Abfragen beantworten

*"Wie wirksam waren die Direct Mail-Kampagnen der letzten 3 Jahre ?"*

...

Backend  
Data Warehouse  
Frontend

---

## 5.4 HANDEL - Eine Entwicklungsskizze

---

### Werkzeuge

- **Hardware und Systemsoftware**  
Client/Server-System
- **Datenmodell**
  - *Logisches* -  
Sternschema auf einer relationalen Datenbank
  - *Physisches* -  
Datendefinitionen, Schlüssel, Integritätsregeln, Optimierungen, ...
- **Data Mining**
  - Werkzeuge vom Unternehmungsberater  
Relationales OLAP-Frontend  
Konventionelle statistische Datenanalyse  
Induktion von Entscheidungsbäumen  
Visualisierung

### Projektmanagement

- Externer Workshop für die Entwickler
- Regelmässiger Kontakt mit den Endbenutzern
- Unterstützung durch das höhere Management
- ...

# Data Warehouses verschiedener Grösse

## Tabellenblatt

bis 1 **Gigabyte** ( $10^9$  Byte)

## Data Mart (lokales DW)

bis **Dutzende** von Gigabytes

## Unternehmensweites DW (globales DW)

bis **Terabytes** ( $10^{12}$  Byte)

Data Marts und Enterprise Data Warehouse

# Data Mart

---

**Data Mart** (DM) :=

lokales Data Warehouse,  
mit Daten eines/einer ...

Funktionsbereichs

Abteilung

Arbeitsgruppe

---

# Data Mart-Ziele

---

## Intramodulare Bindung maximieren

Benutzer so homogen wählen, dass ...

⇒ Antwortzeit minimal ( Partitionierung)

⇒ Benutzerfreundlichkeit maximal

Datenmodell für Funktionsbereich, Abteilung, Arbeitsgruppe<sup>1</sup>

⇒ Werkzeuganpassung maximal ( MOLAP, DOLAP )

## Intermodulare Koppelung minimieren

Schnittstelle zwischen den Data Warehouses so schmal, dass ...

⇒ Data Mart-übergreifende Abfragen selten



<sup>1</sup> Kriterien der Verteilung von DW-Kosten auf Abteilungen?

---

# Enterprise Data Warehouse

---

**Zentrales** Data Warehouse := DW, das ...

Produktionsdaten

transformiert und

koordiniert

Data Marts zu Verfügung stellt

**Enterprise** Data Warehouse (EDW) :=

zentrales Data Warehouse<sup>1</sup> oder

Vereinigung von Data Marts, das/die

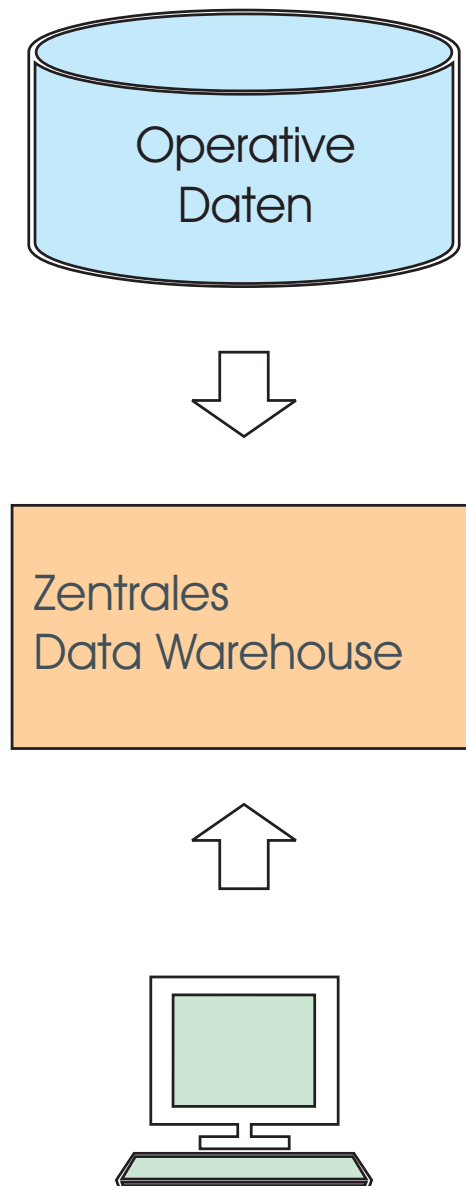
unternehmensweite Information verteilt

---

<sup>1</sup> Zentral kann auch ein DW für die Data Mart einer Niederlassung, eines Funktionsbereichs oder einer grossen Abteilung sein.

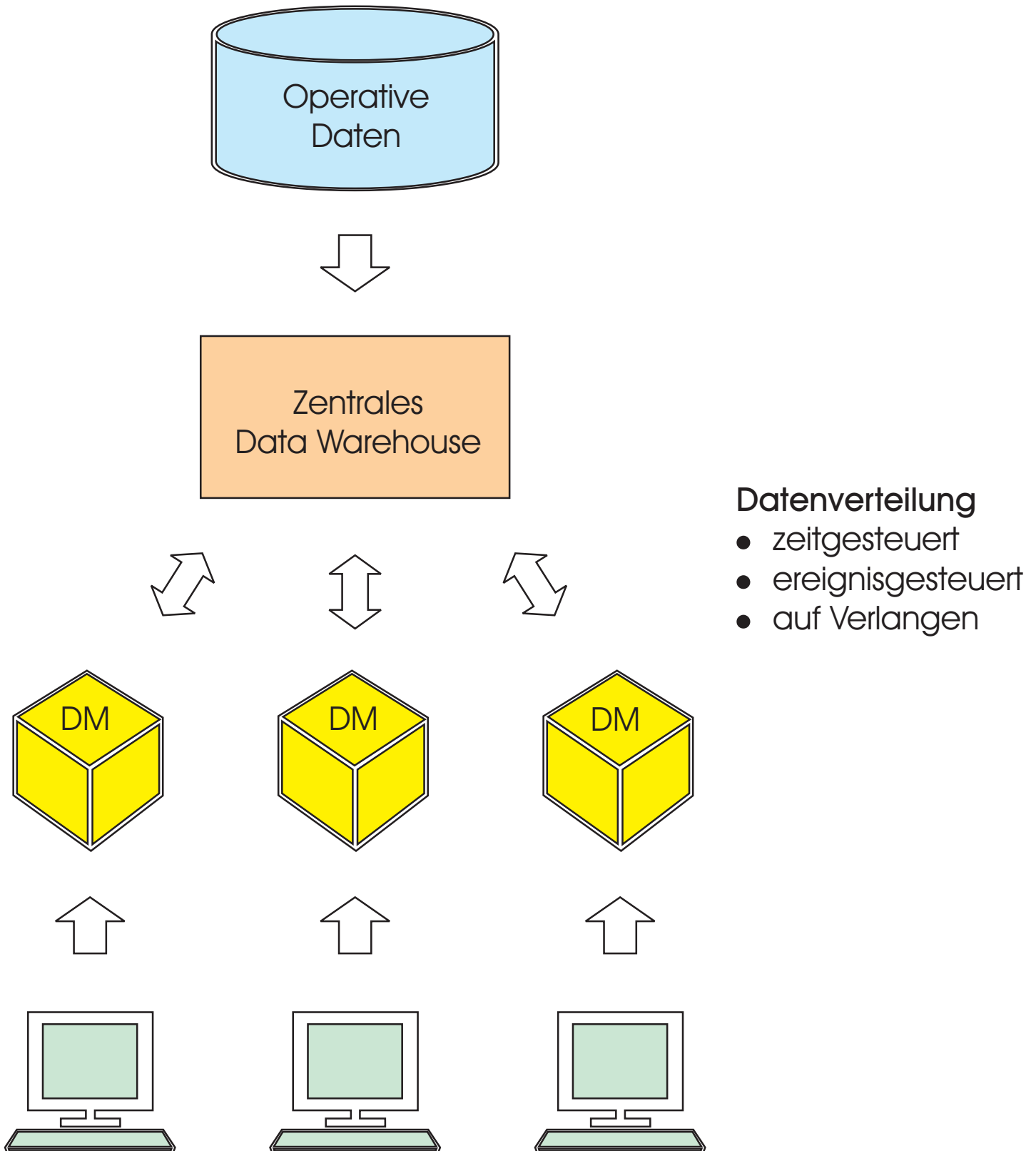


## 5.6 Zentralisierte EDW-Architektur<sup>1</sup>



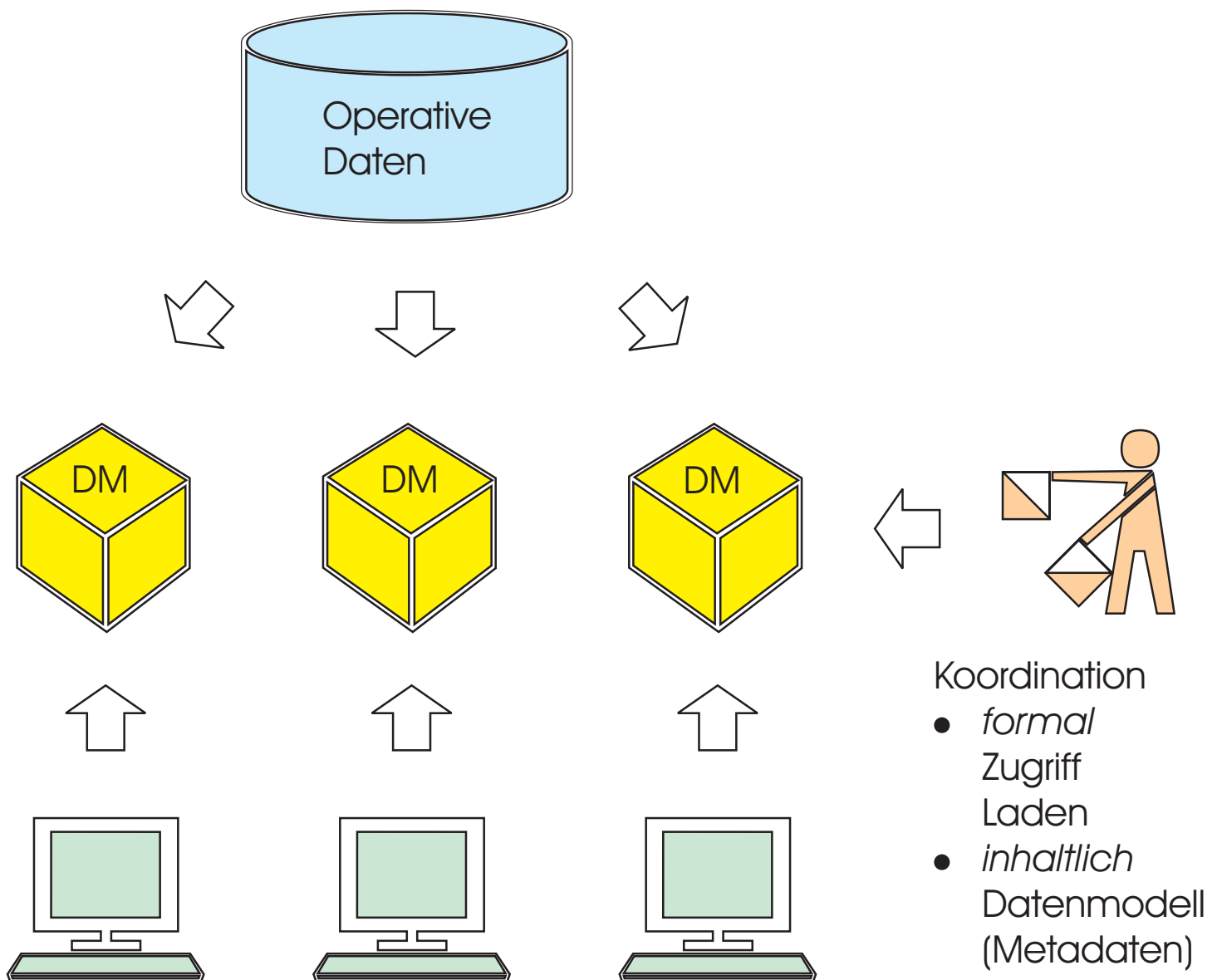
<sup>1</sup> Die Entwicklung eines zentralen DW ist für die meisten Unternehmungen zu komplex

## 5.7 Hierarchische EDW-Architektur (HEDW)



Attribute mit der gleichen Bedeutung verwenden in allen Data Marts  
die gleichen Bezeichner, Datentypen, Schlüssel und Datenquellen.

## 5.10 Koordinierte EDW-Architektur (KEDW)<sup>1</sup>



Koordinierende Metadaten laufend nachführen!

<sup>1</sup> Vereinigung aller Data Marts. Attribute mit der gleichen Bedeutung verwenden in allen Data Marts die gleichen Bezeichner, Datentypen, Schlüssel und Datenquellen.

# Architekturen

Architektur	Zentrales DW	Data Marts	Koordinierte DM
zentralisiert			
hierarchisch			
koordiniert			
unkoordiniert			

Eine *unkoordinierte* Architektur führt zu ...

unkontrollierter Redundanz

Inkonsistenz

hohem Entwicklungsaufwand

ungenügender Skalierbarkeit

# Data Mart-Koordination - Voraussetzungen

---

## Dimensionen oder Fakten ...

### ① **gleicher Bedeutung** ...

gleich benennen

(Synonyme vermeiden!)

mit den gleichen Attributen beschreiben

### ② **unterschiedlicher Bedeutung** ...

verschieden benennen

(Homonyme vermeiden!)

## 5.9 Data Mart-Koordination - Abfragebeispiel

*Produkt* sei eine **koordinierte Dimension** der Sternschemata

Produktion

Lagerhaltung

Verkauf

Eine **sortierende Abfrage** nach Produkten ergibt für jedes Sternschema eine Ergebnistabelle

**Mischen** der drei Tabellen ergibt die folgende Tabelle:

<i>Produkt</i>	① <b>Produziert</b>	② <b>Gelagert</b>	③ <b>Verkauft</b>
A	2000	200	1800
B	4000	1000	3000
C	3000	400	3600

# Anforderungen an Attribute

---

Attribute sind im Idealfall ...

koordiniert

verständlich

vollständig

korrekt

dokumentiert

---

## Aufgabe Anforderungen an Attribute

---

Entwerfen Sie nach den folgenden Regeln eine möglichst breit verwendbare Adressstruktur:

- a) Strukturieren Sie eine Adresse so, dass die Suche nach ihren Elementen möglichst leicht wird
- b) Vermeiden Sie Abkürzungen
- c) Schliessen Sie unterschiedliche Kommunikationsarten ein (Post, Telefon, ...)
- d) Beschränken Sie sich auf natürliche Personen (Adressen juristischer Personen stellen andere Anforderungen).





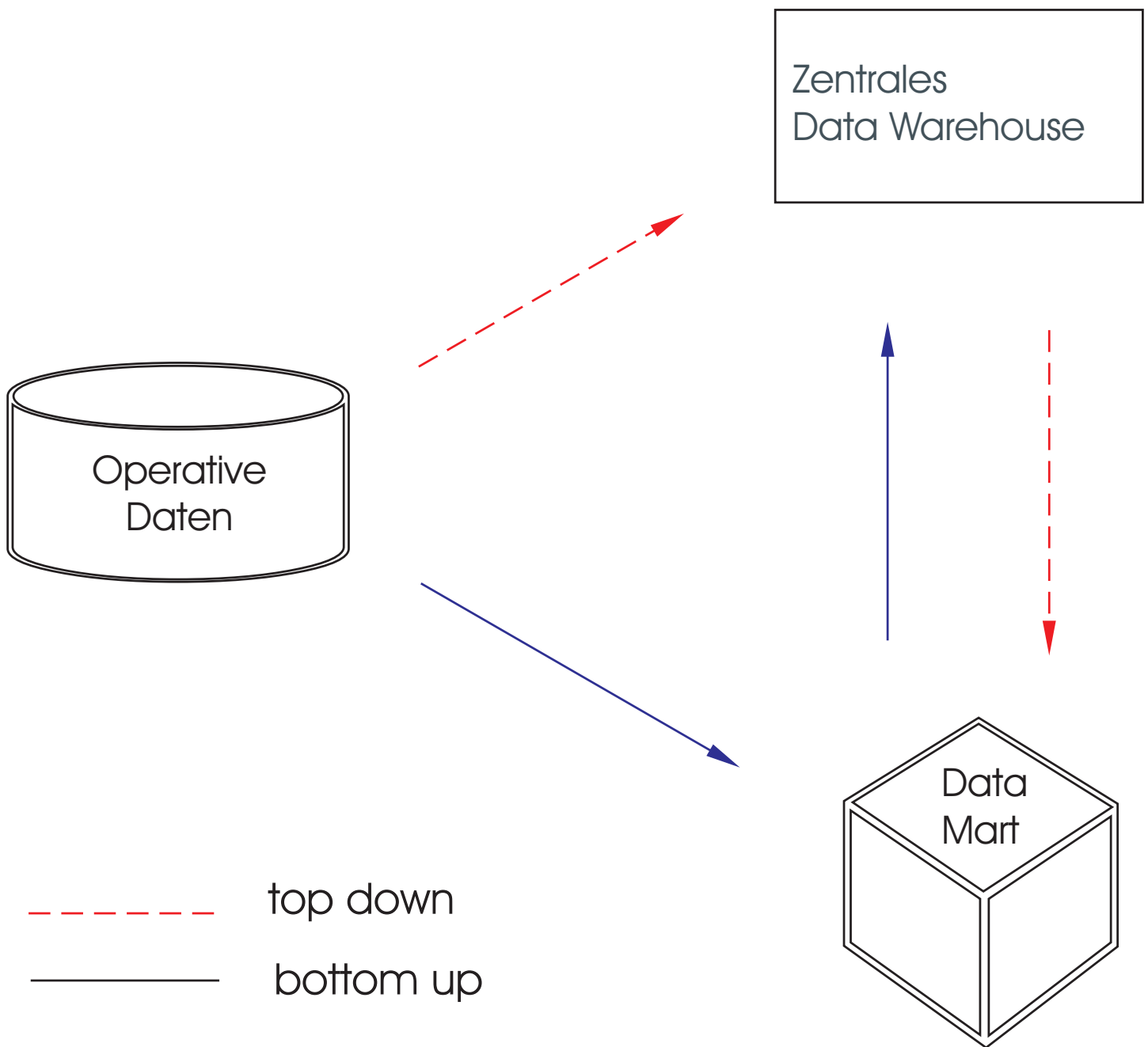
## 5.8 Data Mart vs. Enterprise Data Warehouse

	<b>Data Mart (DM)</b>	<b>EDW</b>
<i>Ähnliche Begriffe</i>	lokales <sup>1</sup> DW	globales DW
<i>Hauptziel</i>	Modularisierung lokaler Daten und Abfragen	redundanzarmes Unternehmungsdatenmodell für EUS
<i>Domäne</i>	Abteilung oder Anwendung	Unternehmung
<i>Entwicklung / Betrieb</i>	Fachabteilung	IT-Zentrale
<i>Grössenordnung</i>	bis 100 Gb (10 <sup>9</sup> Byte)	bis Terabytes (10 <sup>12</sup> Byte, oft grösser als Produktionsdaten)
<i>Datenquelle</i>	EDW oder Produktions-DB	Produktions-DB
<i>Granularität (Datendetail)</i>	oft gross (eher aus Aggregaten)	meist klein (viele Transaktionsdetails)
<i>Normalisierungsgrad</i>	klein	gross
<i>Hauptzugriffsart</i>	OLAP (meist ad hoc)	SQL (oft vordefiniert)
<i>Entwicklung</i>	gut überblickbar	aufwendig
<i>Typische Plattform</i>	mehrdimens. oder relationaler Datenbankserver	RDBMS auf Unix-, Windows NT- oder Grossrechner
<i>Produktbeispiel</i>	<a href="#">Sybase IQ</a>	<a href="#">Sybase MPP</a> <sup>2</sup>

1 EDW und DM sind verteilt oder koexistieren auf einer einzigen Hardware- und Software-Plattform.

2 RDBMS für Massively Parallel Processors

## 5.5 Data Mart-Entwicklung



**Top Down** - Entwicklung

**Bottom Up** - Entwicklung

**Parallele** Entwicklung

# EDW und Data Mart - Ein Kontinuum

	<i><b>Operative DB</b></i>	<i><b>Enterprise DW</b></i>	<i><b>Abteilungs- Data Mart</b></i>	<i><b>Mini- Data Mart</b></i>
<i>Redundanz</i>	minimal	kontrolliert	kontrolliert	kontrolliert
<i>Normalisiert</i>	vollständig	teilweise	kaum	kaum
<i>Granularität</i>	laufendes Detail	historisches Detail	aggregiert	aggregiert
<i>Optimierungs- schwerpunkte</i>	Speicher Zugriff Integrität	Speicher Integrität	Zugriff	Zugriff

## Mini-Data Mart

*Ein Benutzer analysiert Kundendaten nach der Dimension Herkunft und den Kategorien Land, Region und Ort und identifiziert Regionen mit unterdurchschnittlichem Umsatz.*

## Abteilungs-Data Mart

*Unter Umständen analysiert der gleiche Benutzer diese Regionen näher und sammelt z.B. Daten über die umsatzschwächsten Orte.*

## Enterprise Data Warehouse

*Für ausgewählte Kunden untersucht er vielleicht sogar die letzten fünf Jahre.*

## ④ Operative Datenbank

*Falls die Käufe eines Kunden in dieser Periode konsistent abgenommen haben, sendet er die laufende Adresse des Kunden einem Direct Mailing-Mitarbeiter.*

---

# Beurteilung von Data Marts

---

## Vorteile

Entwicklungskomplexität kleiner

Endbenutzerbeteiligung wahrscheinlicher

Flexibilität grösser

Datenmodell einfacher

Abfrageeffizienz grösser

## Probleme

Skalierbar?

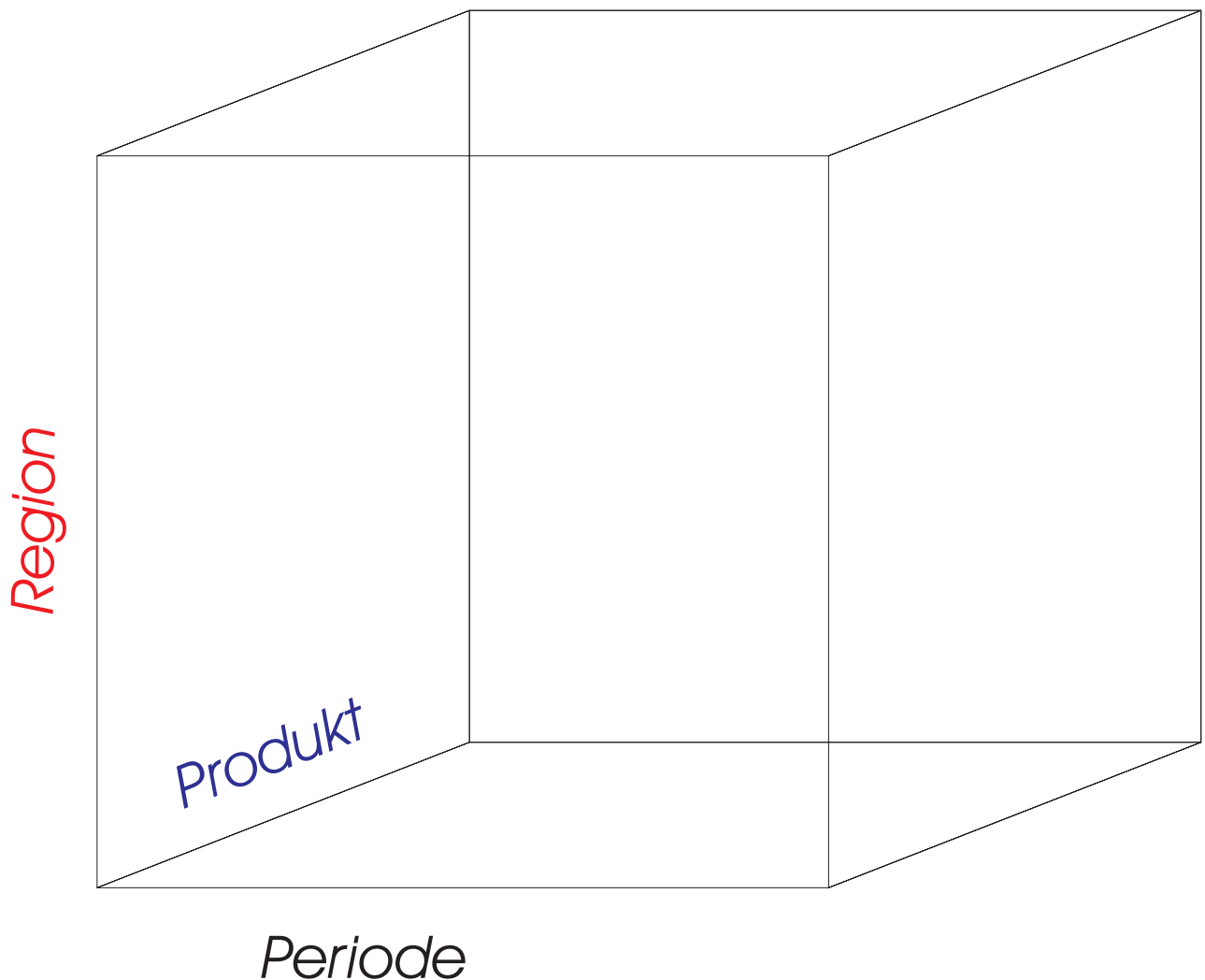
Integrierbar in Enterprise Data Warehouse?

Redundanz kontrollierbar?

Data Marts untereinander konsistent?

Entwicklungs- und Wartungskosten tragbar?

## 5.12 Betriebsnähe durch Indikator u. Dimensionen



“Welches ist der *UMSATZ* nach *Periode* und *Produkt* und *Region*?”

Würfel veranschaulicht die drei **Dimensionen** des Indikators

Am Schnittpunkt der drei Dimensionen ist der Umsatz.

*UMSATZ* ist ein **Indikator** der betrieblichen Leistung

- 1 “Würfel” mit mehr als drei Dimensionen heissen Hyperwürfel (hypercubes)
- 2 Suchen Sie nach weiteren Dimensionen des Umsatzes!

# Würfel - Begriffe

---

**Würfel** (engl. cube bzw. hypercube) :=

mehrdimensionale Datenstruktur, welche die  
gleichzeitige Analyse mehrerer  
Indikatoren und  
Dimensionen ermöglicht

**Indikator** (Fakt, engl. business measure) :=

aggregierbares  
meist numerisches und kontinuierliches Attribut, das die  
mehrdimensionale Messung  
eines betrieblichen Erfolgskriteriums erlaubt

**Dimension** :=

meist symbolisches und diskretes Attribut, das die  
Auswahl, Zusammenfassung und Navigation eines  
Indikators erlaubt

# Einen Indikator nach seinen Dimensionen messen

Fakten (Erfolgsindikatoren) wie ...

*Umsatz*

Dimensionen wie ...

*Zeit*periode

*Produkt*

Mehrdimensionale Abfragen wie ...

*Umsatz* nach *Zeit* nach *Produkt*

Januar			
Produkt A	Produkt B	Produkt A	Produkt B
10,5 Mio.	2,8 Mio.	9,4 Mio.	5 Mio.

Die Dimension **Zeit** (-punkt oder/und -periode) kommt in jedem Data Warehouse vor !

# HANDEL - Fakten nach Dimensionen auswerten

## Fakten

Mengenumsatz

Geldumsatz

Kundenzahl

...

## Dimensionen

Produkt

Periode

Filiale

Werbeaktion

...

## Abfragen

Umsatz **nach** Produkt

Umsatz **nach** Produkt **nach** Monat

Kundenzahl **nach** Filiale

Umsatzzunahme **nach** Werbeaktion



# Indikator versus Dimensionen

	<i><b>Indikator</b></i>	<i><b>Dimension</b></i>
<i>Synonyme</i>	Fakt performance measure, key business measure	constraint
<i>Beispiele</i>	“Umsatz” nach ...	“Produkt”, “Region”
<i>Zweck</i>	Erfolg nach mindestens drei Dimensionen messen	Indikatorwerte wählen, zusammenfassen und durchlaufen
<i>Datentyp</i> (Datenformat)	i.d.R. numerisch und kontinuierlich (aggregierbar)	i.d.R. symbolisch und diskret
<i>Schlüssel</i>	Primärschlüssel aus den Fremdschlüsseln der Dimensionen	Primärschlüssel
<i>Datenvolumen</i>	gross (i.d.R. etwa 70% des Data Warehouse)	klein

- 1 Einteilung in Indikatoren oder Dimensionen nicht immer eindeutig
- 2 Ein Attribut kann als *stetiges* Attribut Indikator (Bsp. *Börsenwert*) und als *diskretes* Attribut Dimension sein (Bsp. Gewinn eines Unternehmens nach der Dimension *Börsenwert-Klasse* analysieren)

---

## Aufgabe *Fakt oder Dimension* (A 5.1)

---

POS-Transaktion ( Marktkorbanalyse)

Region

Zeit

Produkt

Kontenbewegung

Gewinn

Niederlassung

Mitarbeiterzahl

Lieferant

Funktionsbereich

Branche

Börsenwert

Eigenkapitalrentabilität

...

# Mehrdimensionale Daten speichern

Speicherung mehrdimensionaler Daten in einem ...

**relationalen** physischen Modell

als **zweidimensionale** Objektdaten

mehrdimensional verbunden durch **Meta**daten

in einem **relationalen** Server-Datenbanksystem

**mehrdimensionalen** physischen Modell

als **mehrdimensionale** Objektdaten

in einem **proprietären** mehrdimensionalen Server-DBMS

Beide physischen Datenmodelle speichern mehrdimensionale Daten meist aggregiert und oft komprimiert

# Relationale versus mehrdimensionale DB

	Relationale DB <sup>1</sup>	Mehrdimens. DB
Abfragefreundlichkeit	-	+
Abfrageeffizienz	-	+ <sup>2</sup>
Speichereffizienz	+	-
Skalierbarkeit	+	-
Kompatibilität	+ <sup>3</sup>	-

In der Regel werden **Enterprise Data Warehouses** durch spezialisierte RDBMS und **Data Marts** durch relationale oder mehrdimensionale DBMS verwaltet

- 1 RDBMS mit mehrdimensionalem Datenmodell heissen **multirelational**
- 2 Grund: Mehrdimensionale DB speichern Objektdaten mehrdimensional
- 3 Mehrdimensionale DB sind proprietär, relationale stützen sich hingegen auf ein gemeinsames Datenmodell, eine standardisierte Datenbanksprache ( SQL), eine Vielfalt von Werkzeugen und langjährige Erfahrungen

# Anforderungen an RDBMS

<b>Oberkriterium</b>	<b>Kriterium</b>
<b>Modelltreue</b>	
	Sind Objekt- und Metadaten in Tabellenform, und lassen sich darauf Tabellenoperationen ausführen? (Data Dictionary)
	Lassen sich fehlende Werte einheitlich als Nullwerte darstellen?
	Unterstützt die Datenbanksprache die relationalen Grundoperationen?
	Abhängigkeit von Änderungen der physischen und logischen Datenorganisation? (zum Beispiel nach Optimierungen oder Änderungen von Basistabellen)
<b>Integrität</b>	
<i>deklarative -</i>	Welche Integritätsbedingungen lassen sich mit einer deklarativen Datenbeschreibungssprache im Katalog (Data Dictionary) festhalten?
	Entitäts- und Beziehungsintegrität automatisch?
<i>prozedurale -</i>	Programmierung von Integritätsbedingungen als gespeicherte Prozeduren und Trigger?
<b>Sprachen</b>	
	Benutzerfreundlichere Art der deklarativen Datendefinition, -abfrage, -manipulation und -kontrolle als SQL? (zum Beispiel QBE)
	Wie umfangreich sind DDL, DML und DCL?
	Zusammengesetzte Schlüssel definierbar?
	Operationen auf Views?

<b><i>Oberkriterium</i></b>	<b><i>Kriterium</i></b>
	Wieviele Tabellen und Schachtelungen unterstützten die Abfragesprachen?
	Für welche Objekte und Eigenschaften lassen sich Synonyme vergeben?
	Mehrere Versionen von Datenbankobjekten? (engl. versioning)
<b><i>Entwicklung</i></b>	
<b><i>Werkzeug- umfang</i></b>	Autonome Entwicklung ganzer Anwendungen?
	Werden alle Entwicklungsphasen unterstützt? (insbesondere auch der Datenentwurf)
	Strukturierte Programmierung? (z.B. durch Steuerkonstrukte, vollständige Datentypen, lokale und globale Objekte)
	Objektorientierte Anwendungsentwicklung? (Kapselung, Vererbung und Polymorphie)
	Programmierungsumgebung? (zum Beispiel Editor, Debugger, Browser und Versioning)
	Wie vollständig und programmierfreundlich lassen sich die Datenbankfunktionen aus externen 3GL-Sprachen aufrufen? (z.B. Cursors)
<b><i>Werkzeug- komfort</i></b>	Kann der Entwickler die Benutzerschnittstelle deklarativ gestalten? (Formular- und Berichtsgeneratoren)
	Lassen sich Standardanwendungen deklarativ entwickeln? (z.B. durch Applikationsgeneratoren)
<b><i>Kompatibilität</i></b>	
<b><i>Portabilität</i></b>	Welche Betriebssysteme und Hardwareware-Plattformen unterstützt das DBMS?

<b>Oberkriterium</b>	<b>Kriterium</b>
<i>Kommunikation</i>	Werden Mehrprozessorsysteme unterstützt?
	Anwendungsübergreifend? (zum Beispiel über Import/Export, COM oder DDE)
	Welche Datentypen unterstützt das DBMS? (zum Beispiel BLOBs, Memofelder, COM)
<i>Standards</i>	Portabilität des SQL-Dialekts?
	Wie nahe kommt die eingebaute prozedurale Programmiersprache an eine verbreitete 3GL- oder 4GL-Sprache?
<i>Integration</i>	Data Warehousing? Data Mining? Text Mining?
<b>Datensicherheit</b>	
<i>Verfügbarkeit</i>	Lassen sich z.B. Schreiboperationen aus Sicherheitsgründen gleichzeitig auf mehreren Geräten speichern? (weitere Massnahmen: Stand by-Server, On line-Verwaltung)
<i>Backup</i>	Backup während des laufenden Betriebs?
<i>Recovery</i>	Automatischer und manueller Recovery?
	Komfort der Erstellung von Log-Dateien?
<i>Transaktionssteuerung</i>	Welches Sperrprotokoll steuert den konkurrierenden Zugriff mehrerer Transaktionen auf gemeinsame Objekte?
	Welche Isolierungsgrade werden unterstützt?
	Welches ist die Granularität der Sperren? (Datenbank, Tabelle, Page, Satz, Feld)
	Deadlocks automatisch erkannt und aufgelöst?
<b>Datenschutz</b>	
	Datenschutzmassnahmen (z.B. Passwortschutz, Verschlüsselung, Zugriffsprotokolle)

<b>Oberkriterium</b>	<b>Kriterium</b>
	Granularität der Datenschutzmassnahmen? (System, Datenbank, Tabelle, Page, Satz, Feld)
	Hiding von Datenbankobjekten?
	Lassen sich Benutzergruppen bilden?
	Lassen sich Berechtigungen vererben?
<b>Netzfähigkeit</b>	
	Welche Netzwerkprotokolle werden unterstützt?
	Verteilte Architekturen? ( Fileserver-, Client/Server-, verteilte Datenbanken)
	Erlaubt das DBMS heterogen verteilte Daten?
	Wie gut wird Replikation unterstützt?
	Qualität der World Wide Web-Einbindung? (z.B. E-Commerce)
<b>Physische Grenzen</b>	
	z.B. Speicherbedarf; Anzahl Netzknoten; Länge von Feld, Schlüssel, Satz, Namen, Datei, Datenbank und Programm?
	Skalierbarkeit
<b>Effizienz</b>	
<b>Speicher-</b>	Wie flexibel lässt sich der Cache ( Puffer) verwalten?
	Lassen sich Sätze physisch zusammenfassen und als Cluster speichern?
	Lassen sich die Daten komprimieren?
<b>Laufzeit-</b>	Wie kann der Zugriff optimiert werden? (z.B. durch B-Bäume und Hashfunktionen)
	Werden Suchabfragen automatisch optimiert? Lokal und global? (in verteilten Datenbanken)






<b><i>Oberkriterium</i></b>	<b><i>Kriterium</i></b>
	Erzeugt der Übersetzer der prozeduralen Sprache (v.a. gespeicherter Prozeduren) Maschinen- oder Zwischencode?
<b><i>Komfort</i></b>	
	Bietet das System vollständige Einführungs- und Referenzhilfen an? (on line, schriftlich, Schulung)
	Wie gut ist die laufende Unterstützung durch den Anbieter? (technischer Support)
	Existieren Benutzervereinigungen?
<b><i>Anbieter- und Produktqualität</i></b>	
	Produktumstände wie Reifegrad oder Häufigkeit und Aufwand von Versionswechseln?
	Herstellerfirma? (zum Beispiel ihre wirtschaftliche Lage, die lokale und globale Marktpräsenz und die eingeholten Referenzen)
	In welches Abhängigkeitsverhältnis gegenüber dem Anbieter begeben wir uns?

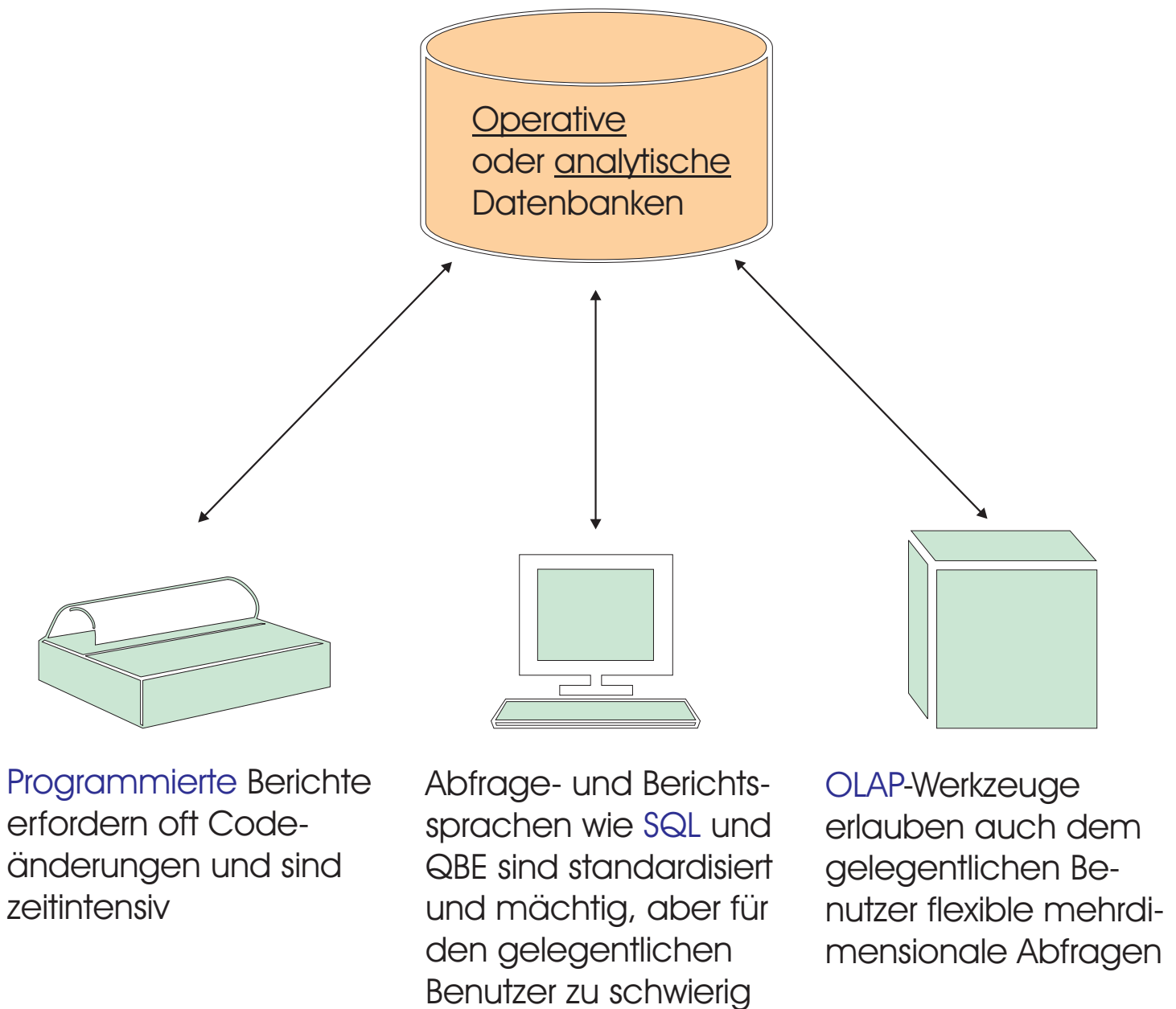
---

# Inhalt

---

Grundlagen	<a href="#"><u>2</u></a>
Entwicklung - Überblick	<a href="#"><u>3</u></a>
Data Marts und Enterprise Data Warehouse	<a href="#"><u>6</u></a>
Mehrdimensionale Daten	<a href="#"><u>21</u></a>
Endbenutzerzugriff	<a href="#"><u>34</u></a>
OLAP	<a href="#"><u>44</u></a>
ANLAGEBERATUNG mit <i>PowerPlay</i> 	<a href="#"><u>71</u></a>
Modellierung relationaler Data Warehouses	<a href="#"><u>84</u></a>
Modellierung von Informationssystemen	<a href="#"><u>85</u></a>
Sternschema ( LIEFERFRIST )	<a href="#"><u>97</u></a>
EDW-Modell und Sternschema ( VERKAUF )	<a href="#"><u>106</u></a>
<i>Synchrony</i> ( EINZELHANDEL  )	<a href="#"><u>126</u></a>
Entwicklung und Betrieb	<a href="#"><u>136</u></a>
Fallbeispiel ( ZEITSCHRIFTEN  )	<a href="#"><u>151</u></a>
Effizienz	<a href="#"><u>160</u></a>
Netzzugriff	<a href="#"><u>198</u></a>
Client/Server-Architekturen	<a href="#"><u>203</u></a>
Zugriff über Internet und Intranet	<a href="#"><u>213</u></a>

## 5.15 Zugriff auf operative und analytische Daten



### Benutzerfreundlichere Abfragen mit wenig ...

Papier

Entwicklungsaufwand

Auswertungsaufwand (vgl. Spreadsheets)

# Datenbankabfragen

---

**Abfragen i.w.S.** sind ...

**Auswahl**abfragen (Abfragen i.e.S.)

Suchabfragen

**Aktions**abfragen

Anfüge-

Lösch-,

Aktualisierungs-

Tabellenerstellungs-

# Abfragesprachen

---

**Prozedurale** - sind flexibler

Navigation meist satzweise

Anweisungsreihenfolge wichtig

*Visual Basic, Cobol, dBASE*

**Nichtprozedurale** - (deklarative) sind komfortabler

Navigation tabellenweise

einzigste verbale oder grafische Anweisung

*QBE, SQL*

## 5.16 Prozedurale und deklarative Abfragen

### ① Umgangssprachliche -

Welche Eigenschaften haben die Produkte der Abteilung "Farben"?

### ② Formalsprachliche -

#### prozedurale - (dBASE)

```
use PRODUKTE
% -- temporär zwischenspeichern
copy to TMP
use TMP
delete for Produktart <> 'Farben'
% -- Duplikate eliminieren
total on PRODUKTE to ERGEBNIS
display all
```

#### deklarative - (SQL)

```
select *
from PRODUKTE
where Produktart = "Farben"
```

## 5.17 Query by Example (QBE)

“Wieviele **Stunden** unterrichtet Dozent **Meier** im **Sommersemester 1993** ?”

### ORGANISATION

Name	Semester	Dozent	Raumnummer
x	SS93	Meier	

### VERANSTALTUNG

Name	Stunden
x	√

### ANTWORT

Stunden
2

Tabellenskelette

Ausgabespalten

Ausgabezeilen

Verbundattribute

Abfragenname

# SQL - Grobübersicht

**SQL** := Structured Query Language

deklarativ (WAS statt WAS UND WIE)

interaktiv oder in ein Programm eingebettet

## Operationen

### Datenbankobjekte definieren

`create`

### Datenbankobjekte manipulieren

`insert`

`update`

`delete`

### Datenbanken abfragen

`select`

### Datenbanken schützen

`grant`



## 5.18 SQL - Teilsprachen

SQL-Kategorie	Zweck	Anweisungen
Data <b>Definition</b> Language (DDL)	Definition von Datenbankobjekten	create, alter, drop table, view, index
Data <b>Manipulation</b> Language (DML)	Abfrage Änderung	select insert, delete, update
Data <b>Control</b> Language (DCL)	Transaktionen Schutz / Sicherheit	lock, commit, rollback grant, revoke

Übersicht 6.10: Kategorien von SQL-Anweisungen

---

## SQL - Vorteile

---

portabel ( ANSI / ISO-Standard)  
für verteilte Datenbanken geeignet  
in prozedurale Programme einbettbar  
anders als QBE terminalunabhängig

---

# Mängel von SQL

---

Mehrdimensionale SQL-Abfragen sind benutzer~~un~~freundlich, weil oft ...

Anweisungen **zahlreich**

Anweisungen **komplex**

(Mehrtabellenverbund, Aggregation, Sortieren, Views)

**Zeitreihenanalyse** aufwendig

**Rechenfunktionen** unzulänglich

(z.B. finanzmathematische, statistische)

Benutzerfreundliche mehrdimensionale Abfragen mit OLAP

## 5.19 Abfragearten

	<i>Hauptzweck</i>	<i>Flexibilität</i>	<i>Komfort</i>	<i>Interaktivität</i>
<b>SQL</b>	allgemein	nein	-	ja
<b>Bericht</b>	Abfrage breit	ja	+ <sup>1</sup>	nein <sup>2</sup>
<b>Menü</b>	Abfrage eng	ja	+	ja
<b>QBE</b>	Abfrage breit	nein	∅ <sup>3</sup>	ja
<b>OLAP</b>	Abfrage und Analyse	nein	+	ja

**Integration** von Berichtsgeneratoren,  
OLAP-Werkzeugen und Data Mining Tools

Business Objects voll integriert

Bericht + Menüabfrage + OLAP  
(*Business Miner* getrennt erhältlich)

Cognos Improptu, PowerPlay, Scenario als Suite

Bericht, OLAP, Data Mining

1 aus Benutzersicht (Entwicklung mit 3GL, 4GL oder EIS Tools aufwendig)

2 Desktop Report Writers für den Entwickler beschränkt interaktiv

3 QBE syntaktisch einfach, verlangt aber logisches Datenmodell

# Endbenutzerzugriff mit OLAP - Definition

**OLAP** (On Line Analytical Processing) :=

Abfragemethode, die ...

Endbenutzern einen

mehrdimensionalen

schnellen **Zugriff** und eine

benutzerfreundliche interaktive **Analyse**

(wie mit mehrdimensionalen Tabellenkalkulationspaketen)

auf **Data Marts** ermöglicht

E.F. Codd führte 1993 das Konzept ein

## 5.20 OLAP versus OLTP

	<b>OLTP</b>	<b>OLAP</b>
<i>Abfragedaten</i>	operative Transaktionsdaten	managementkritische Analysedaten
<i>Granularität</i> (Detaillierungsgrad)	mikroskopisch (originäre Daten)	makroskopisch (oft abgeleitete Daten)
<i>Aktualität</i>	vollständiger Istzustand	Sequenzen historischer Snapshots
<i>Hauptoperationen</i>	fortschreiben (read / write)	abfragen, berichten und berechnen (meist read only)
<i>Werkzeuge</i>	3GL, 4GL (v.a. SQL)	proprietär
<i>Abfragefreundlichkeit</i>	gering	gross
<i>Speichereffizienz</i>	gross	gering
<i>Optimierungsstrategie</i>	transaktionsorientiert	datenorientiert

**OLTP**

On Line **Transaction** Processing

**OLAP**

On Line **Analytical** Processing

---

# Anwendungen

---

## Finanz- und Rechnungswesen

- Kurzfristige Erfolgsrechnung
- Jahresabschlussanalyse
- Cash Flow - Analyse
- Kennziffernanalyse (insb. ROI-Analyse)

## Absatz

- Soll-/Ist-Vergleiche
- Produkt- und Kundenvergleiche
- Qualität des Kundendienstes

## Beschaffung

- Bestandsanalysen
- Lieferfristenüberwachung

## Produktion

- Kapazitätsanalysen
- Qualitätskontrolle

## Personalwesen

- Personalverwaltung
- Mitarbeiterqualifikationen

...

---

# Definitionselemente

---

**OLAP** ist ...

**mehrdimensional**, weil es ...

Indikatoren nach ihren Dimensionen analysiert

**endbenutzerfreundlich**, weil es ...

direkte Manipulation ermöglicht<sup>1</sup>

zusammenfasst und nur auf Verlangen detailliert

Drilling Up and Down navigiert entlang einer Dimension  
analysiert und synthetisiert

Slicing addiert und Dicing subtrahiert Dimensionen

Ergebnisse einfach darstellt, insbesondere visualisiert

**schnell**, weil es ...

- ad hoc-Abfragen in Sekunden beantwortet

**Data Mart** - orientiert, weil es ...

- weniger Details als das zentrale EDW unterstützt
- i.d.R. nicht direkt auf Produktionsdaten zugreift

---

<sup>1</sup> Vgl. mit der Komplexität einer generierten SQL-Abfrage von *if...Synchrony*



## Datenbasis *NASDAQ*

NASDAQ (National Association of Security Dealers Automated Quotation System) ist ein Bildschirmkommunikationssystem für den Handel mit ausserbörslich kotierten Wertpapieren von etwa 5000 Unternehmungen

## Datenmodell *Anlageberatung*

Würfel mit ...

25 **Indikatoren** (Fakten)

jährlicher Geldumsatz  
jährlicher Nettoerfolg

...

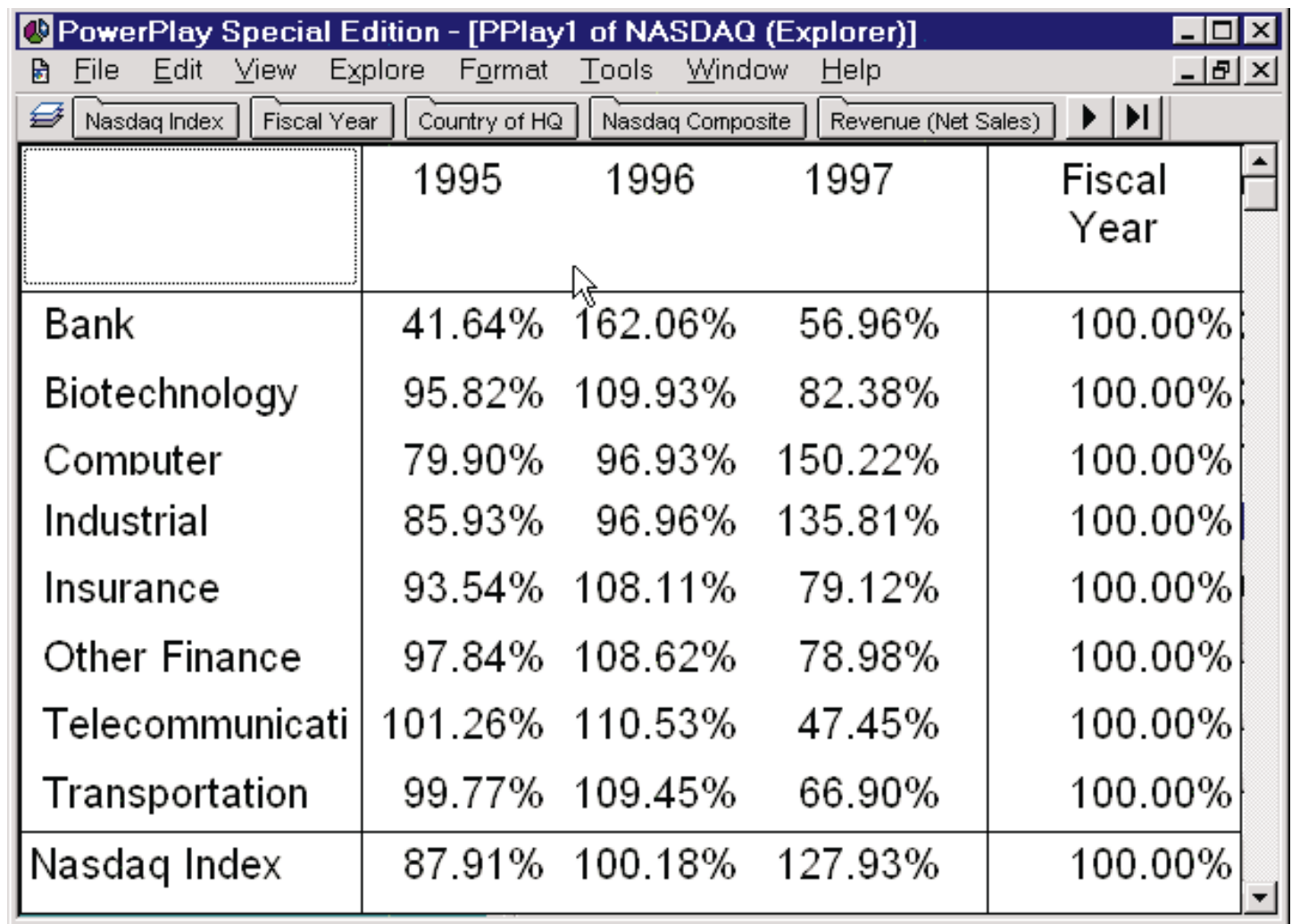
15 **Dimensionen**

Branche  
Firmensitz  
Fiskaljahr  
Börsenwert (Klassenzugehörigkeit)  
jährliche Nettoerfolgsänderung in % (Klassenzugehörigkeit)  
Eigenkapitalrendite (Klassenzugehörigkeit)

...

NASDAQ.mdc (multidimensional cube)

## 📌 Einfache Darstellung von Ergebnissen



The screenshot shows the 'PowerPlay Special Edition - [PPlay1 of NASDAQ (Explorer)]' window. The menu bar includes File, Edit, View, Explore, Format, Tools, Window, and Help. The toolbar has buttons for 'Nasdaq Index', 'Fiscal Year', 'Country of HQ', 'Nasdaq Composite', and 'Revenue (Net Sales)', along with navigation arrows. The main data table is as follows:

	1995	1996	1997	Fiscal Year
Bank	41.64%	162.06%	56.96%	100.00%
Biotechnology	95.82%	109.93%	82.38%	100.00%
Computer	79.90%	96.93%	150.22%	100.00%
Industrial	85.93%	96.96%	135.81%	100.00%
Insurance	93.54%	108.11%	79.12%	100.00%
Other Finance	97.84%	108.62%	78.98%	100.00%
Telecommunicati	101.26%	110.53%	47.45%	100.00%
Transportation	99.77%	109.45%	66.90%	100.00%
Nasdaq Index	87.91%	100.18%	127.93%	100.00%

Beispiel: Der Menüpunkt *View/Show Values As ...* wechselt zwischen absoluter und relativer Darstellung

Indikator **in Prozent von ...**

- Zeilentotal
- Spaltentotal
- Ebene (engl. layer)
- Gesamttotal

# Importierte und abgeleitete Daten



## 1. **Importierte** Daten

v.a. aus operativen Datenbanken und EDW

## 2. **Abgeleitete** Daten

z.B. mit gespeicherten Prozeduren

## 3. **Ad hoc abgeleitete** Daten

z.B. mit generiertem SQL

Je mehr permanent abgeleitete Daten, desto ...

- kleiner die Antwortzeit
- grösser die Ladezeit
- grösser der Speicherbedarf

# Zusammenfassungen und Vergleiche

## Zusammenfassung (Aggregation)

*Quartalerträge* des letzten Jahres

## Vergleich

*Absoluter* Vergleich der diesjährigen mit letztjährigen Gewinnen

*Prozentualer* Beitrag aller Produkte an den Gesamtumsatz

Abschlüsse der *besten* 10% US-Vertreter

...

Trendanalysen

Soll- / Istvergleiche

Benchmarkvergleiche

...

SQL sucht nach Details

OLAP fasst zusammen und vergleicht

# Zusammenfassung - Ein Beispiel

**StarTracker Demo**

File Edit **Aggregates** Sequences Comparisons Help

Run Report

PERIODEN: Sum, Count, Count Distinct, Avg, Min, Max, AvgPeriodSum, AvgPeriodCount, Tagesnr\_total, Wochenr\_in\_Lab

PRODUKTE: Alle Produkte, Produkt#, Name, Gewicht, Lieferant, Unterkategorie, Kategorie

Verkäufe: Periode#, Produkt#, Aktion#, Filiale#, Geldumsatz, Mengenumsatz, Kosten, Kundenzahl, Durchschnittlich verkaufte Stückzahl\*, Durchschnittliche Auftragsgrösse\*, Durchschnittskosten\*, Durchschnittspreis\*, Gewinn\*, Kostenquote\*, Report Columns\*

FILIALEN: Alle Filialen, Filiale#, Strasse, Ort, Kanton, PLZ, Coach-Stadion

AKTIONEN: Alle Aktionen, Aktion#, Name, Preisreduktion, Anzeige, Display, Anzeigenmedium

Kostenvergleich  
for 4. Quartal 98

	A	B	C	D	E	F	G	H	I	J	K
1	Ort	Veränderung in %	Kosten 4. Quartal '98	Kosten 4. Quartal '97							
2	Aarau	2.51%	15898.66	15508.86							
3	Basel	-13.63%	17456.13	20209.79							
4	Bellinzona	-18.30%	15501.94	18975.36							
5	Bern	8.35%	17459.52	16113.75							
6	Biel	-16.48%	15068.73	18042.93							
7	Brig	-1.08%	17070.44	17257.63							
8	Chur	7.32%	19072.19	17771.85							
9	Freiburg	-1.62%	17100.58	17381.68							
10	Genf	21.93%	17656.07	14480.07							
11	Laufen	7.74%	17182.35	15947.90							
12	Lausanne	16.87%	20263.64	17338.48							
13	Liestal	-3.82%	17476.27	18169.48							
14	Locarno	-3.39%	15828.11	16383.57							
15	Luzern	-14.94%	16680.32	19609.66							
16	Martigny	1.28%	16880.94	16667.19							
17	Rheinfelden	12.73%	17751.06	15746.94							

Family: Handelsunternehmen 16.10.97 14:04

OLAP-Werkzeuge generieren im Hintergrund oft SQL-Abfragen

# Zusammenfassungen können verfälschen

Eine **Aggregation** (Zusammenfassung) kann zu Informationsverlusten führen

## *Beispiel*

”Gründe für tiefe Umsätze ? “

Konsolidierung von Tagesumsätzen zu Wochenumsätzen verdeckt den Einfluss einzelner Tagesumsätze

---

# Browsing

---

**Browsing** :=

benutzerfreundliche Abfrage mit ...

## Drilling Down and Up

Zusammenfassung und Detaillierung entlang einer Dimension

## Drilling Through

Detailzugriff auf operative Datenbanken  
(meist über vordefinierte SQL-Anweisungen)

## Drilling Across

Zugriff auf mehrere Data Marts

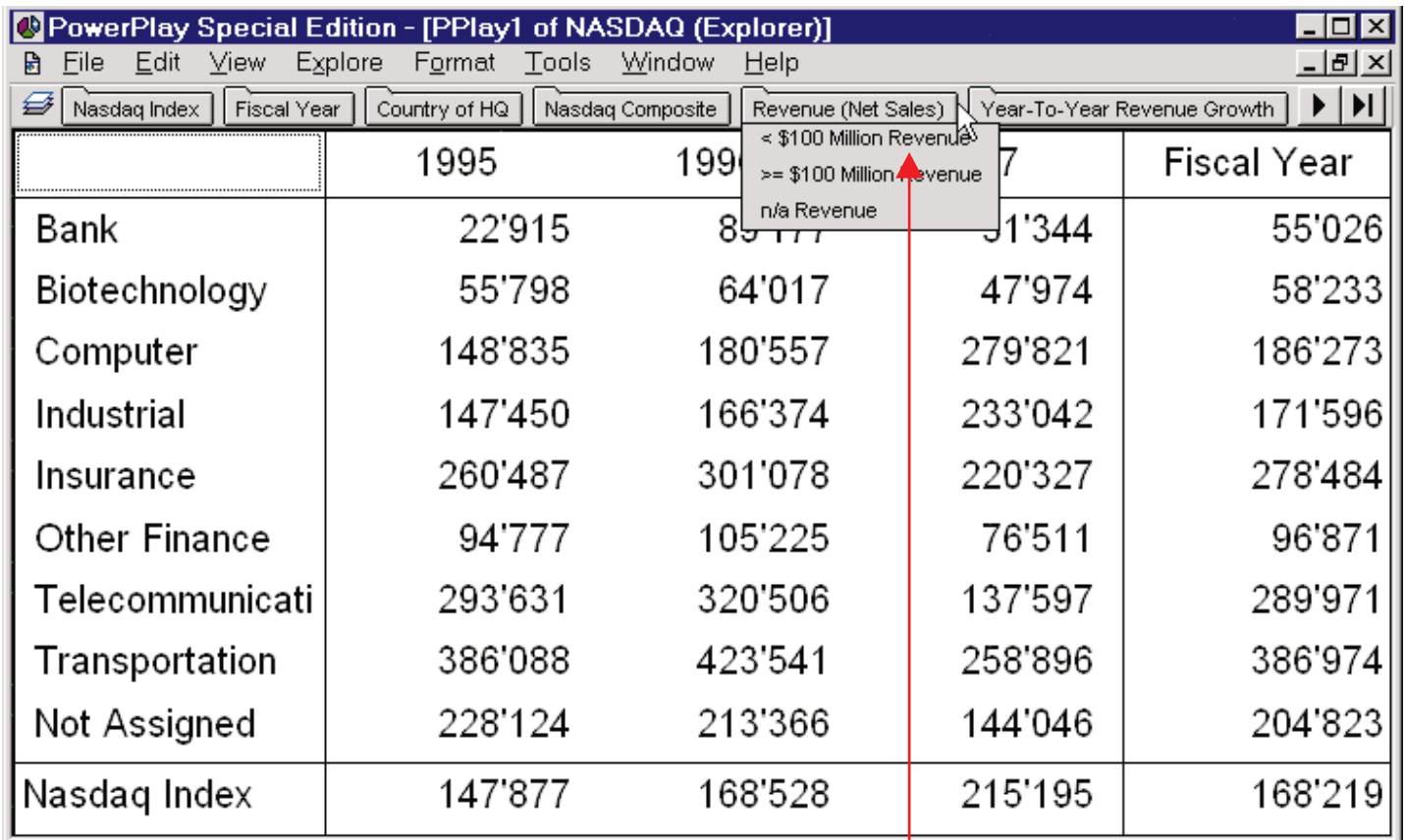
## Filtern

Werteauswahl

## Slicing and Dicing

Dimensionsauswahl

## 5.21 📌 Filtern beschränkt die angezeigten Werte



The screenshot shows the PowerPlay Special Edition interface. The main window displays a table of financial data for various industries. A filter is applied to the 'Revenue (Net Sales)' dimension, showing only values greater than or equal to \$100 million. The filter is indicated by a red arrow pointing to the '>= \$100 Million Revenue' option in the dropdown menu. The table columns are: Industry, 1995, 1996, 1997, and Fiscal Year. The industries listed are Bank, Biotechnology, Computer, Industrial, Insurance, Other Finance, Telecommunications, Transportation, Not Assigned, and Nasdaq Index.

	1995	1996	1997	Fiscal Year
Bank	22'915	85'177	51'344	55'026
Biotechnology	55'798	64'017	47'974	58'233
Computer	148'835	180'557	279'821	186'273
Industrial	147'450	166'374	233'042	171'596
Insurance	260'487	301'078	220'327	278'484
Other Finance	94'777	105'225	76'511	96'871
Telecommunications	293'631	320'506	137'597	289'971
Transportation	386'088	423'541	258'896	386'974
Not Assigned	228'124	213'366	144'046	204'823
Nasdaq Index	147'877	168'528	215'195	168'219

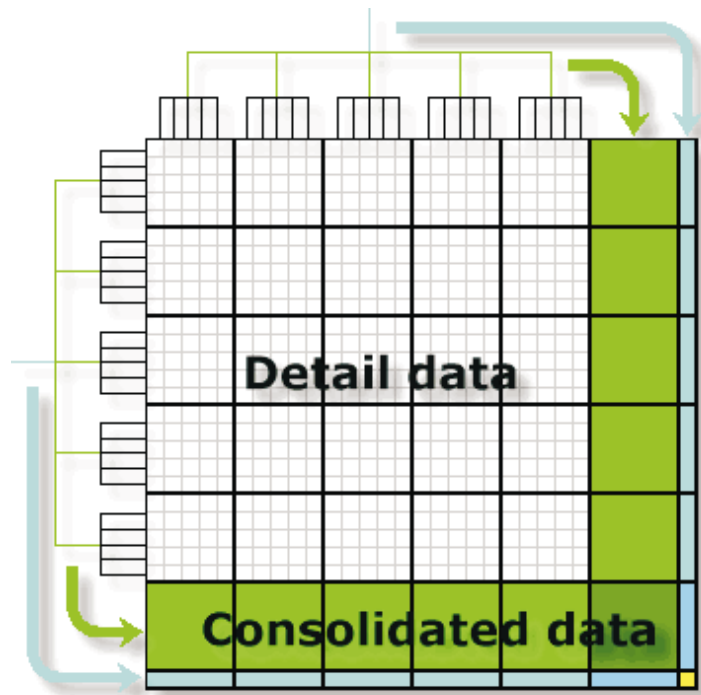
**Filtern** (Werteauswahl) :

Cursor auf eine Dimension der horizontalen Dimensionenleiste bewegen und auf ein Filterkriterium klicken

Beispiel: Unternehmungen mit einem Jahresertrag von  $\geq 100$  Mio



# Drilling Up and Down



**Drilling Up** (zusammenfassend)  
**Drilling Down** (detaillierend)

Umsatzzahlen für ...

**ganz Europa**

**Mitteleuropa**



**Deutschland**

## 5.21 📌 Drilling Down - Ausgangslage

PowerPlay Special Edition - [PPlay2 of NASDAQ (Explorer)]

File Edit View Explore Format Tools Window Help

Nasdaq Index Fiscal Year Country of HQ Nasdaq Composite Revenue (Net Sales)

		1996	1997	Fiscal Year
Bank	915	89'177	31'344	55'026
Biotechnology	55'798	64'017	47'974	58'233
Computer	148'835	180'557	279'821	186'273
Industrial	147'450	166'374	233'042	171'596
Insurance	260'487	301'078	220'327	278'484
Other Finance	94'777	105'225	76'511	96'871
Telecommunications	293'631	320'506	137'597	289'971
Transportation	386'088	423'541	258'896	386'974
Nasdaq Index	147'877	168'528	215'195	168'219

### Indikator

Beispiel: *Jahresertrag*

### Dimension :

Cursor über einem Dimensionsreiter der Dimensionszeile

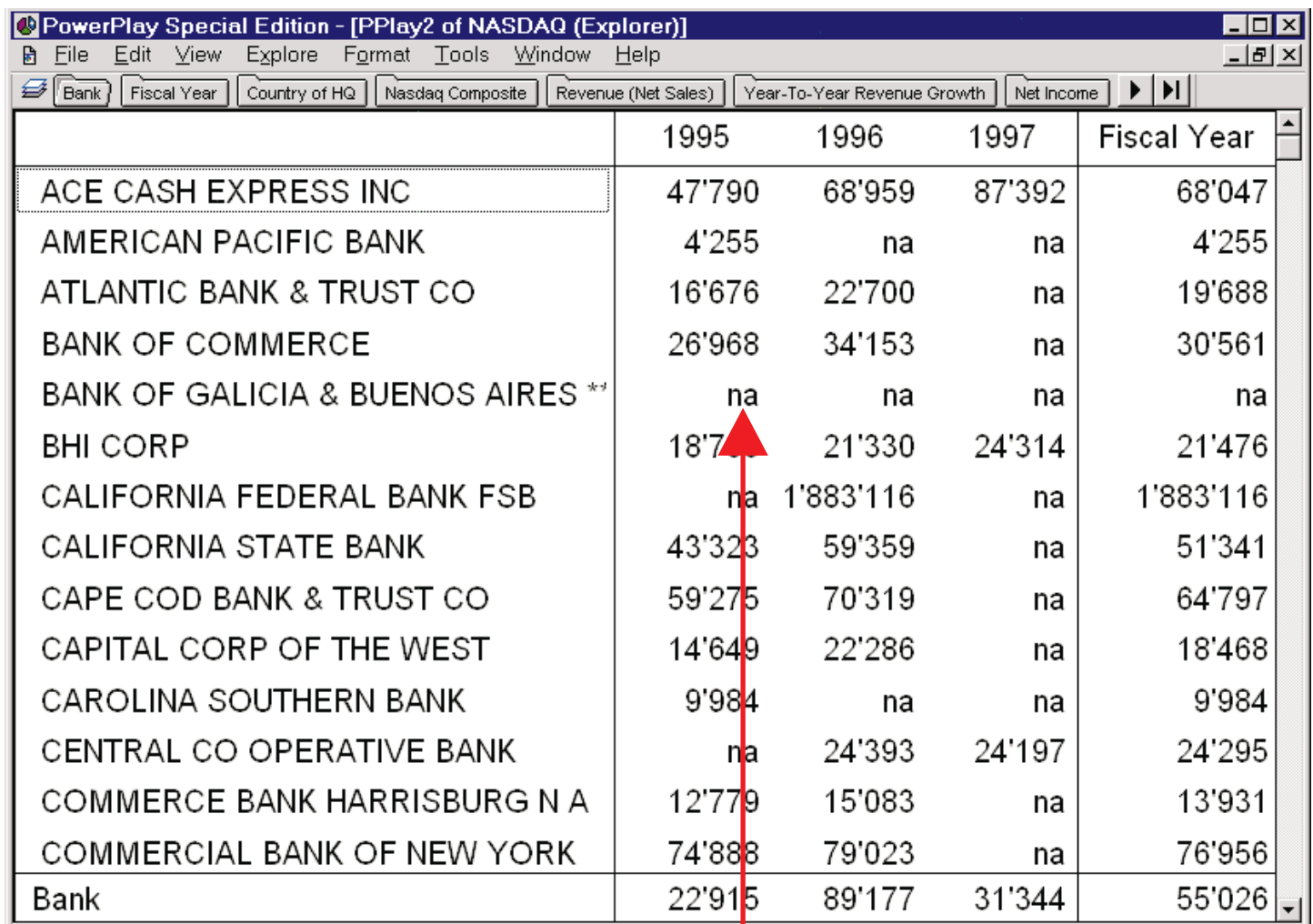
Beispiel: Dimensionen *Branche* und *Zeit*

### Drilling Down or Up →

Down: Doppelklick auf Zellenwert. Up: rechte Maustaste

Beispiel: Jahreserträge der einzelnen *Bank*unternehmen

## 5.22 📌 Drilling Down - Ergebnis



The screenshot shows a software window titled "PowerPlay Special Edition - [PPlay2 of NASDAQ (Explorer)]". It has a menu bar (File, Edit, View, Explore, Format, Tools, Window, Help) and a toolbar with buttons for "Bank", "Fiscal Year", "Country of HQ", "Nasdaq Composite", "Revenue (Net Sales)", "Year-To-Year Revenue Growth", and "Net Income". Below the toolbar is a table with the following data:

	1995	1996	1997	Fiscal Year
ACE CASH EXPRESS INC	47'790	68'959	87'392	68'047
AMERICAN PACIFIC BANK	4'255	na	na	4'255
ATLANTIC BANK & TRUST CO	16'676	22'700	na	19'688
BANK OF COMMERCE	26'968	34'153	na	30'561
BANK OF GALICIA & BUENOS AIRES **	na	na	na	na
BHI CORP	18'7	21'330	24'314	21'476
CALIFORNIA FEDERAL BANK FSB	na	1'883'116	na	1'883'116
CALIFORNIA STATE BANK	43'323	59'359	na	51'341
CAPE COD BANK & TRUST CO	59'275	70'319	na	64'797
CAPITAL CORP OF THE WEST	14'649	22'286	na	18'468
CAROLINA SOUTHERN BANK	9'984	na	na	9'984
CENTRAL CO OPERATIVE BANK	na	24'393	24'197	24'295
COMMERCE BANK HARRISBURG N A	12'779	15'083	na	13'931
COMMERCIAL BANK OF NEW YORK	74'888	79'023	na	76'956
Bank	22'915	89'177	31'344	55'026

A red arrow points from the text "na (not available)" below the table to the "na" value in the 1995 column for "BHI CORP".

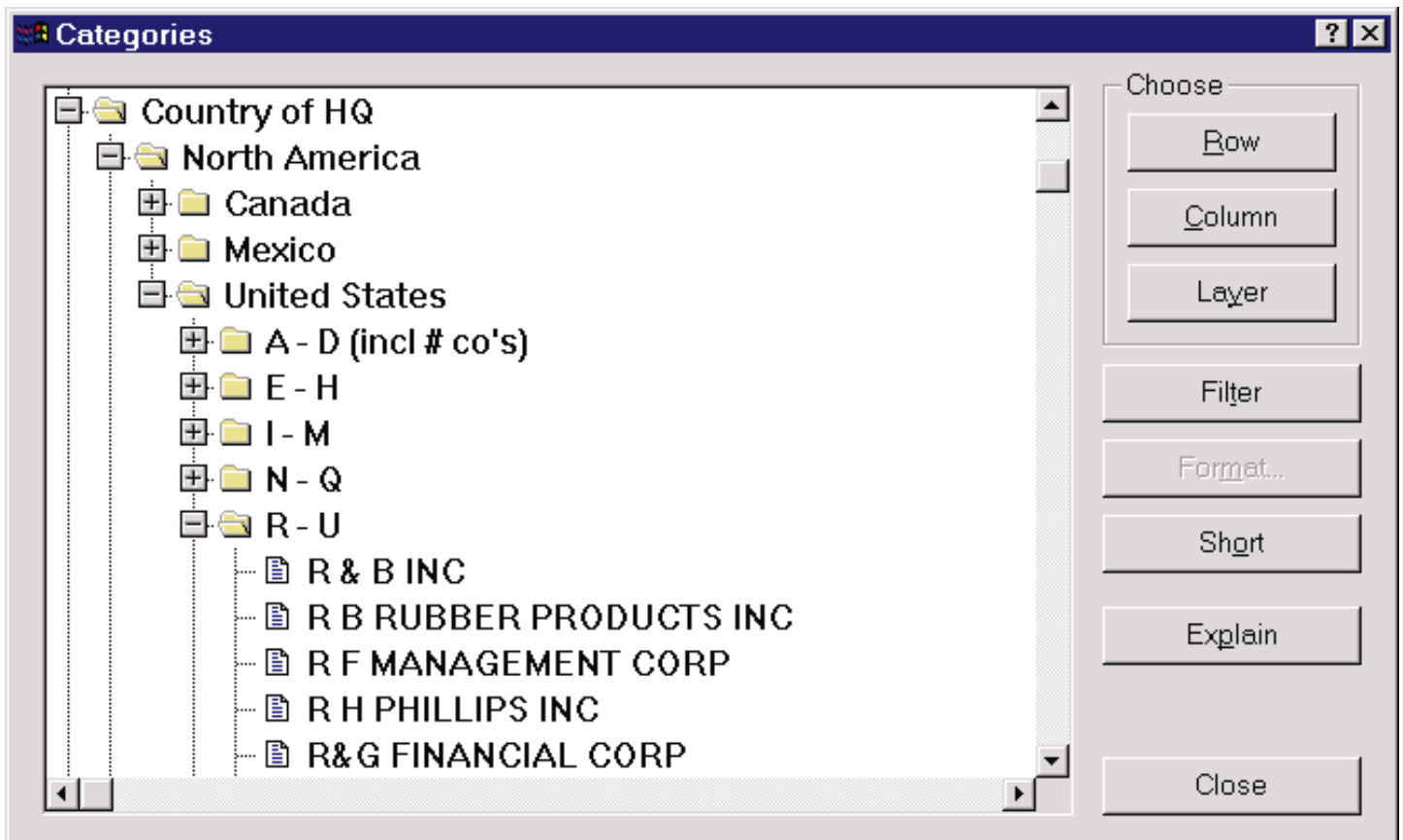
na (not available)

Je grösser die **Granularität**, desto mehr überflüssiger Speicher

Je mehr **fehlende Werte**, desto mehr Speicherineffizienz

Wie werden **dünn besetzte Matrizen** am besten gespeichert?

## 📌 Drilling navigiert in Kategorienhierarchien



### Kategorienhierarchie einer Dimension

Die Dimension FIRMENSITZ ("Country of HeadQuarter") ordnet ihre Kategorien (Wertebereiche) **hierarchisch**:

Dimension FIRMENSITZ

*Kategorie Ländergruppe (Vaterknoten)*

*Kategorie Land (Kindknoten von Ländergruppe)*

*Kategorie alphabetische Gruppe*

*Kategorie Einzelunternehmung*

# Dimensionen mit Kategorien beschreiben

## Hierarchische Kategorien

Bsp. Kategorien der Dimension *Periode*

*Perioden*

1995

1. Quartal

Januar

...

2. Quartal

...

...

1996

...

Die Kategorienhierarchie der Dimension *Periode* ordnet die Kategorien **Jahr**, **Quartal** und **Monat** hierarchisch

## Flache Kategorien

Bsp. Kategorien der Dimension *Produkt*

*Name, Farbe, Grösse, Art ...*

Ein Indikator oder eine Dimension kann *mehrere* flache oder hierarchische Kategorien haben

## Drilling Down kann verfälschen

Zu kleine **Fallzahlen** können bei mehrdimensionalen Analysen zu Informationsverlusten führen

### *Beispiel*

“Gründe für tiefe Umsätze ?“



Die Abfrage “Umsatz nach Produkt, Filiale und Abteilung” reduziert die Stichprobe zu stark

## 5.23 † Slicing und Dicing analysiert / synthetisiert

### **Slicing and Dicing** (Dimensionswahl):

Änderung der angezeigten Dimension mit Drag and Drop einer Dimension der horizontalen Leiste über dem Anzeigefenster

Beispiel: Drag and Drop der neuen Dimension *Firmensitz* (Country of HQ) in die horizontale oder vertikale Titelleiste des Anzeigebereichs ersetzt die alte Dimension *Fiskaljahr*

	Nasdaq Index	Fiscal Year	Country of HQ	Nasdaq Composite	Revenue (Net Sales)
		1995	1996		
Bank		22'915	89'177		
Biotechnology		55'798	64'017		
Computer		148'835	180'557		

	Nasdaq Index	Fiscal Year	Country of HQ	Nasdaq Composite	Revenue (Net Sales)
		North America	Europe		
Bank		56'023	na		
Biotechnology		54'825	26'932		
Computer		191'503	80'836		

## 📌 Pivoting tauscht Zeilen und Spalten

PowerPlay Special Edition - [PPlay1 of NASDAQ (Explorer)]

File Edit View Explore Format Tools Window Help

Nasdaq Index Fiscal Year Country of HQ Nasdaq Composite Revenue (Net Sales)

	Bank	Biotechnology	Computer	Industrial
North America	56'023	54'825	191'503	172'717
Europe	na	26'932	80'836	149'610
Asia Pacific	na	920	794'566	90'836
Other	21'476	314'846	52'088	93'256
Country of HQ	55'026	58'233	186'273	171'596

**Pivoting** :  
Zeilen und Spalten tauschen<sup>1</sup>

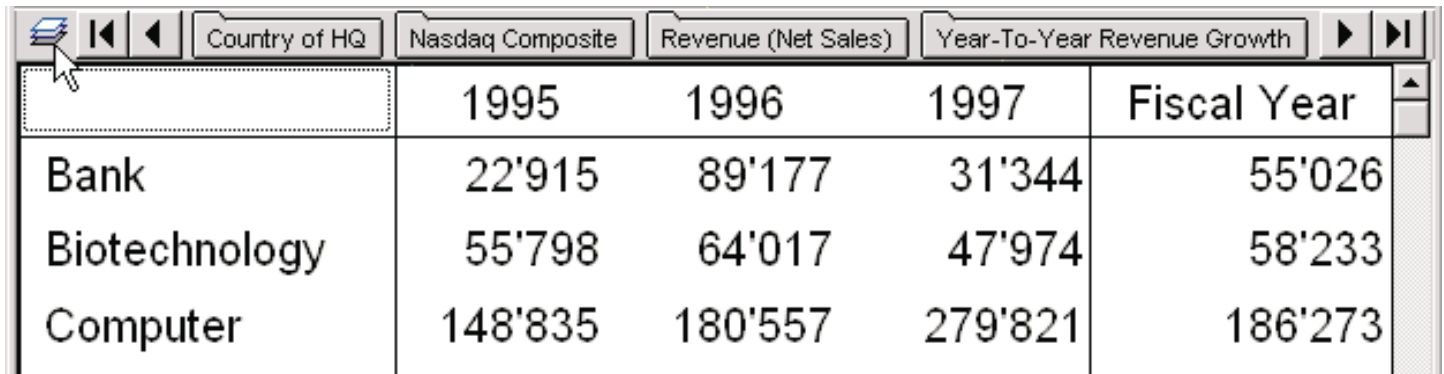
Beispiel: Tausch der Dimensionen *Branche* (Bank, ...) und *Firmensitz*

<sup>1</sup> to pivot heisst drehen



## 5.24 † Eine **Ebene** ist eine Zusatzdimension

**Ebenen:** Für jeden Wert einer Zusatzdimension eine zusätzliche Ebene (**Drag and Drop** von *Country of HQ* auf das Ebenensymbol)



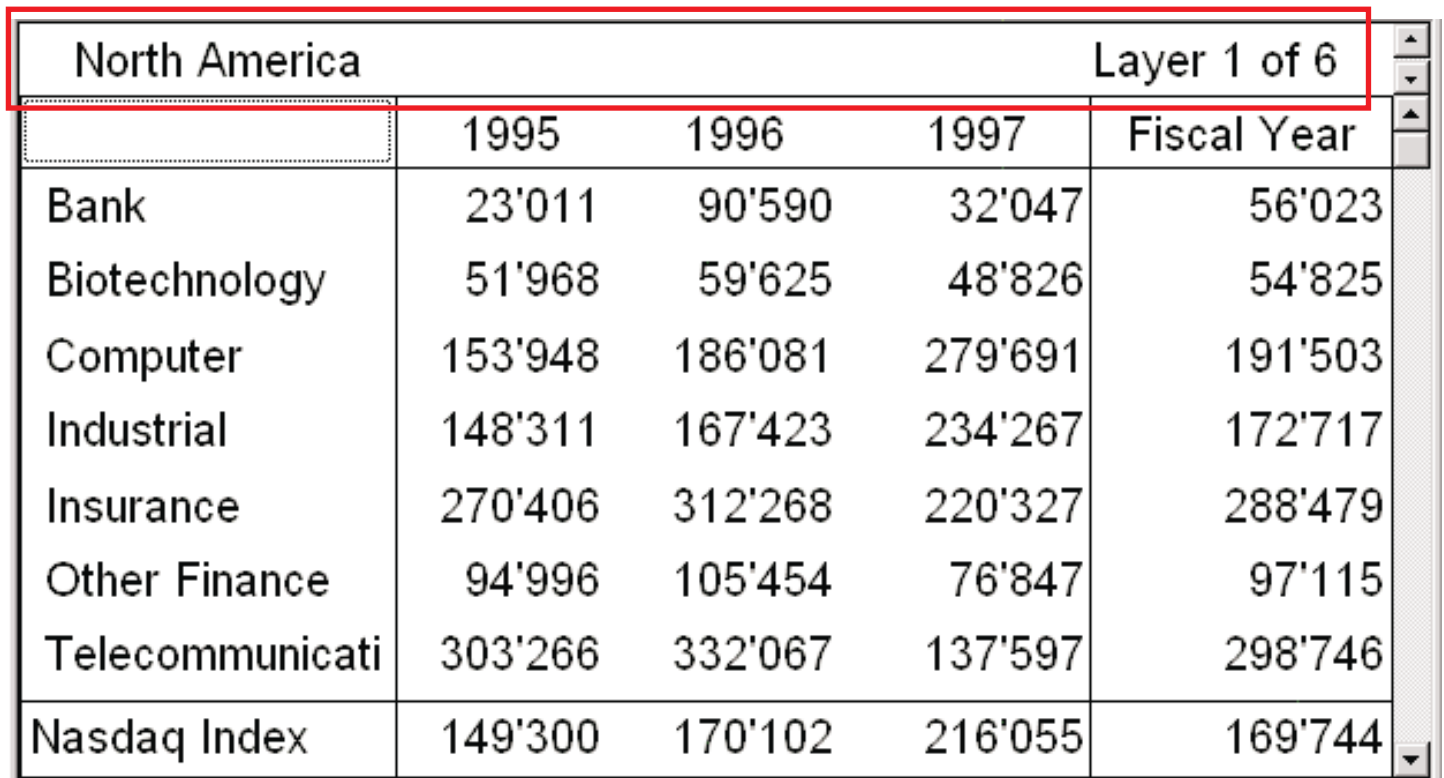
	1995	1996	1997	Fiscal Year
Bank	22'915	89'177	31'344	55'026
Biotechnology	55'798	64'017	47'974	58'233
Computer	148'835	180'557	279'821	186'273

### Beispiel

North America als Wert der Zusatzdimension *Firmensitz*:

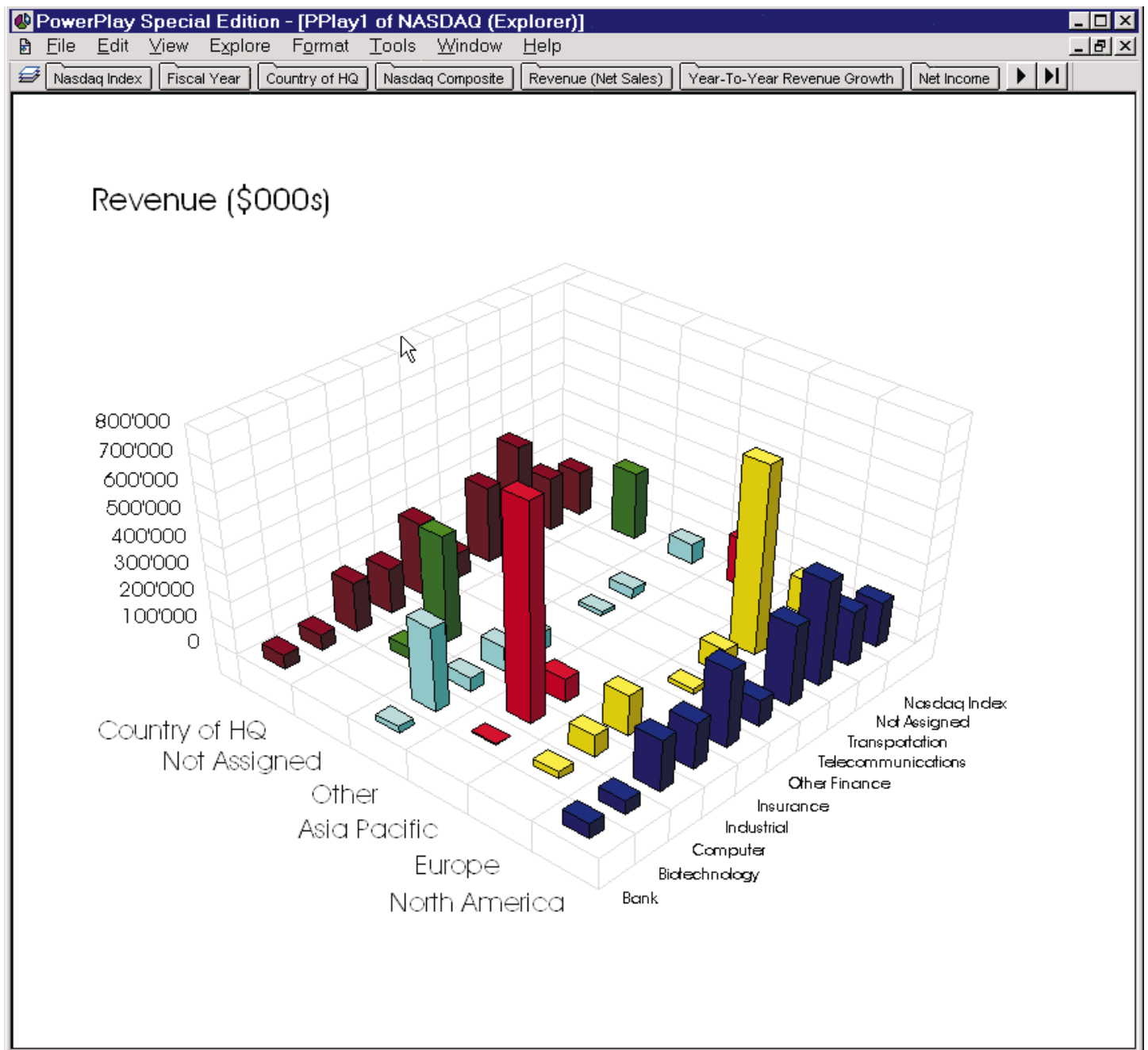
Unterschied zum Slicing and Dicing

Anzeige von 3 (statt nur 2) Dimensionen



North America	1995	1996	1997	Fiscal Year
Bank	23'011	90'590	32'047	56'023
Biotechnology	51'968	59'625	48'826	54'825
Computer	153'948	186'081	279'691	191'503
Industrial	148'311	167'423	234'267	172'717
Insurance	270'406	312'268	220'327	288'479
Other Finance	94'996	105'454	76'847	97'115
Telecommunicati	303'266	332'067	137'597	298'746
Nasdaq Index	149'300	170'102	216'055	169'744

## 5.25 📌 Die Visualisierung hebt hervor



### Dreidimensionale Visualisierung :

Dreidimensionale grafische Veranschaulichung eines Indikators und zweier Dimensionen

Beispiel: Indikator *Jahresertrag*, Dimensionen *Branche*, *Firmensitz*

---

# OLAP-Arten - Definitionen

---

**ROLAP** (engl. Relational OLAP) :=

OLAP

mit mehrdimensionalen Metadaten

in einem relationalen Server-Datenbanksystem

**MOLAP** (engl. Multidimensional OLAP) :=

OLAP

mit mehrdimensionalen Objektdaten

in einem proprietären Server-Datenbanksystem

**DOLAP** (engl. Desktop OLAP) :=

MOLAP oder ROLAP

auf einem Client (i.d.R. einem PC)

# OLAP-Arten - Vergleich

	ROLAP	MOLAP	DOLAP
<i>Abkürzung</i>	<u>R</u> elationales OLAP	<u>M</u> ehrdimen- sionales -	<u>D</u> esktop -
<i>EDW-Daten i.d.R.</i>	relational	relational	relational,/ multidimen- sional
<i>Mehrdimensiona- lität in den ...</i>	Metadaten (Direktzugriff auf EDW möglich)	Objektdaten	Metadaten/ Objektdaten
<i>Datenumfang</i>	gross	gross	klein
<i>Datenort</i>	Server	Server	Client (PC)
<i>Datenmodell</i>	standardisiert	proprietär	proprietär
<i>Abfrageeffizienz</i>	-	+	+
<i>Flexibilität</i>	+	-	+/-
<i>Skalierbarkeit</i>	+	Ø	-
<i>Integrierbarkeit</i>	+	-	+/-
<i>Vorberechnung</i>	geringer	umfangreich	umfangreich
<i>Beispielanbieter</i>	<u>Information Advantage</u>	<u>Arbor,</u> <u>Oracle</u>	<u>Cognos,</u> <u>Brio</u>

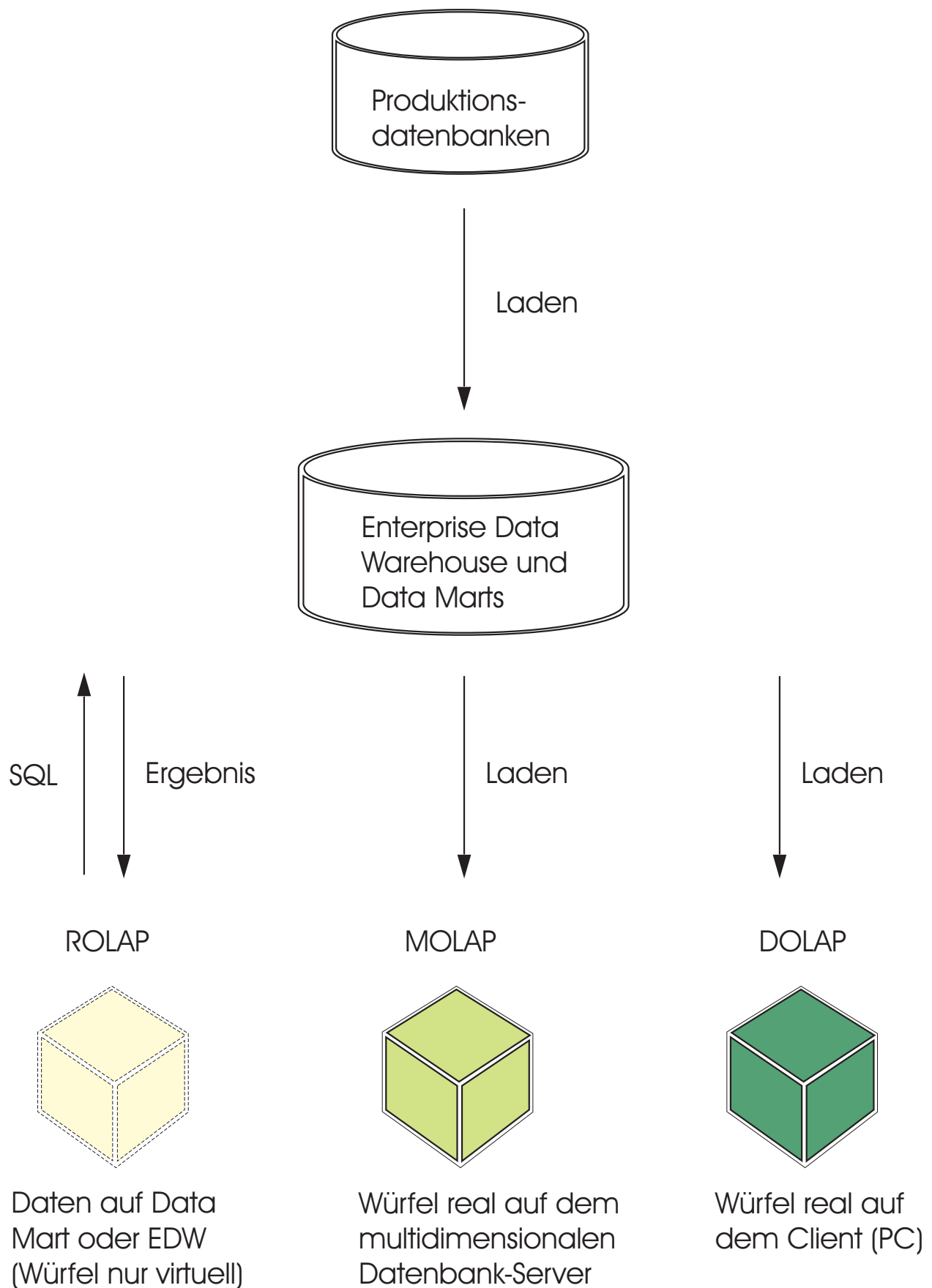
## Hybride OLAP-Systeme

- MOLAP auf einem mehrdimensionalen Würfel
- ROLAP durch generiertes SQL auf einem relationalen DW

Bsp. Pilot Decision Support Suite

15

## 5.28 OLAP-Architekturen



## 5.29 OLAP-Architekturen und Produkte

	<i><b>Speicherung in RDBMS</b></i>	<i><b>Speicherung im MDBMS</b></i>	<i><b>Speicherung im Client</b></i>
<i><b>Verarbeitung mit RDBMS</b></i>	DSS Agent Server		
<i><b>Verarbeitung mit MDBMS</b></i>	Decision Suite Oracle Express	Hyperion Essbase Oracle Express	
<i><b>Verarbeitung im Client</b></i>	PowerPlay Server	Hyperion MBA	BrioQuery PowerPlay Client
<u>Legende</u>	ROLAP	MOLAP	DOLAP
	light client		
	fat client		

### Produkte und Hersteller

- DSS Agent (Server) von MicroStrategy
- Decision Suite von Information Advantage Express (ROLAP) von Oracle
- PowerPlay (Server) von Cognos
- Hyperion Essbase von Hyperion Software Express von Oracle (stark verbreitet)
- Hyperion Enterprise von Hyperion Software (spezialisiert für Finanzanwendungen)
- PowerPlay (Client) von Cognos (populärstes DOLAP-Produkt, Data Mining-Ergänzung Cognos Scenario)  
BrioQuery von Brio Technology

## *PowerPlay* ist . . .

ein **DOLAP** - Frontend ...

Desktop - OLAP-Frontend auf dem

- Client oder
- Server

für benutzerfreundliche **Abfragen** ...

- ad hoc statt geplant (vordefiniert)
- direktmanipulierend statt befehlsorientiert (wie SQL)
- explorativ statt off line (Drilling, Slicing/Dicing, ...)

auf einem **Würfel** mit proprietär Datenstruktur

*PowerPlay* setzt einen *vordefinierten* Würfel  
aus Data Warehouse-Daten voraus

NASDAQ.MDC (Multidimensional Cube)

## 5.30 Entwicklung eines DOLAP-Würfels

### Metadatenverwalter des Data Warehouse

(z.B. Sybase PowerDesigner Warehouse Architect)

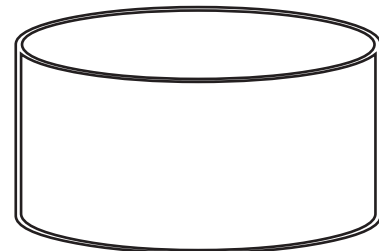
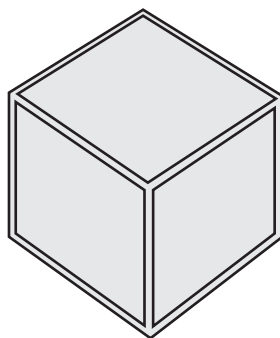


Grobes physisches Modell der multidimensionalen Daten



### Metadatenverwalter des DOLAP-Würfels

(z.B. Cognos PowerPlay Administrator)

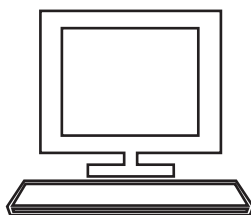


### Multidimensionaler Würfel

(z.B. PowerPlay PowerCube)

### Data Warehouse

(z.B. Sybase - RDBMS)



### DOLAP - Viewer

(z.B. Cognos PowerPlay)



---

## ANLAGEBERATUNG mit *Cognos PowerPlay* (5.2)

---

Cognos PowerPlay und ein Würfel aus NASDAQ-Daten sollen einen Anlageberater bei Portfolioentscheidungen unterstützen. Im ersten Teil der folgenden Aufgabe lernen Sie PowerPlay geleitet kennen. In einem zweiten Teil können Sie das Werkzeug an einem Anlageberatungsbeispiel selbständig erkunden.

### *Lernziele*

- ⇒ Das Grundmodell der mehrdimensionale Analyse erkennen
- ⇒ OLAP-Abfragekonzepte anwenden

### *1 PowerPlay kennen lernen*

Laden Sie den Würfel [NASDAQ.mdc](#).

Mit den Menüpunkten *View* und *File/Preferences* können Sie die Benutzeroberfläche konfigurieren.

Wenn Sie den Cursor auf ein Icon der Symbolleiste positionieren, erhalten Sie eine Kurzbeschreibung. Ausführliche Hilfe zur Symbol- und Menüleiste finden Sie unter dem Menüpunkt *Help/Inhalt/Reference Information/ Shortcuts*. Die Shortcuts Ctrl-Z und Ctrl-Y stehen für undo bzw. redo.

Informationen zum Inhalt des NASDAQ-Würfels erhalten Sie mit einem Rechtsklick auf eine Zelle oder auf ein Register (engl. tab) der Dimensionenleiste. Der Kontextmenüpunkt "Explain" erklärt dann die Bedeutung des Indikators, der Dimension oder des Wertes.

#### a) Dimensionen und Indikatoren

Nach dem Start erscheint die voreingestellte Ansicht "3-D-Bar" (Falls keine dreidimensionale Balkengrafik erscheint, klicken Sie auf das Icon 3-D-Bar). Identifizieren Sie den Indikator, die Dimensionen und deren Wertebereiche (Kategorien).

Ein Klick auf das Icon “Crosstab” wechselt zur tabellarischen Ansicht. Unter der Symbolleiste finden Sie mehrere Register zu den Dimensionen und Indikatoren des Würfels. Den Indikator “Revenue” (Ertrag in 1000\$) finden Sie am rechten Ende der Registerleiste. Wenn Sie den Mauszeiger darauf positionieren, erscheinen weitere mögliche Indikatoren.

Ein Klick auf ein Register zeigt die Kategorien der Dimension bzw. die Measures in einem Pop Up-Fenster. Schauen Sie sich die Kategorienhierarchien der Dimensionen *Country of HQ* und *Net Income* an.

## b) Filter

Die Ausgangstabelle enthält den Indikator “Durchschnittlicher Unternehmungsertrag” nach den Dimensionen Branche (“Nasdaq Index”) und Steuerjahr (“Fiscal Year”). Verwenden Sie einen Filter, der nur Unternehmen mit einer Eigenkapitalrentabilität von mehr als 20% zeigt. Gehen Sie dazu mit dem Mauszeiger zur Dimension “Return on Equity”. Sobald Sie die Maus nicht mehr bewegen, erscheint ein Menü mit den Filterkriterien. Wählen Sie “>=20% ROE” (engl. Return On Equity). Experimentieren Sie mit den anderen Kriterien.

## c) Pivoting

Wechseln Sie die Spalten und Zeilen mit *Explore/Swap/Rows and Columns*.

## d) Drilling Down and Up

Ein Doppelklick auf eine Tabellenzelle führt eine Kategorie tiefer (Drilling Down). Eine Kategorie höher kommen Sie mit einem Rechtsklick auf eine Zelle (Drilling Up).

## e) Nullwerte

“na” (not available) bezeichnet Zellen ohne Wert. *Explore/Suppress Zeros/Rows and Columns* zeigt Zeilen oder Spalten, die nur aus Nullwerten bestehen, nicht an.

f) Ansichten (engl. views)

*View/Show Values As* zeigt Ergebnisse prozentual statt absolut an, und *Explore/Hide Selected Categories* blendet markierte Zeilen und Spalten aus.

g) Sortierung (engl. rank)

Sortieren Sie markierte Spalten oder Zeilen mit dem Menüpunkt *Explore/Rank*.

h) Ebenen (engl. layers)

Tabellen können nur zwei Dimensionen gleichzeitig darstellen. Um eine dritte Dimension zu zeigen, können Sie jeden ihrer Werte als zusätzliche Tabelle (Ebene) anzeigen lassen. Ziehen Sie dazu die gewünschte dritte Dimension mit der Maus auf das Stapelsymbol am linken Ende der Dimensionsleiste.

i) Visualisierung

Wechseln Sie mit *View/Change Display* von der Tabellenansicht (crosstab) zur Würfelsicht (3D-Bar). Wenden Sie Filter, Slice and Dice, Drill Down and Up und Sortierung auch auf diese Sicht an.

## 2 PowerPlay selbständig anwenden

a) Zeigen Sie die TopTen der Banken mit den höchsten Eigenkapitalrenditen (ROE) des Jahres 1997 an.

b) Welchen Marktwert hatte "Microsoft" 1997?

c) Veranschaulichen Sie die Risiko-Ertragslage in einem Multiline-Diagramm, das die Eigenkapitalrendite (ROE) und den Beta-Koeffizient branchenweise darstellt. Welche Branche weist das ungünstigste Verhältnis aus? Welche das beste? Diskutieren Sie zunächst, welche Beta-Werte günstig oder ungünstig sind. Nehmen Sie zur Lösung der Aufgabe an, dass ein möglichst kleiner (positiver) Beta-Wert ideal sei. Warum kommen Sie u.U. zur falschen Lösung, wenn Sie den ROE durch Beta dividieren und

das höchste Resultat als das beste und das tiefste als das schlechteste Verhältnis ansehen?

d) Reporter

Sie möchten ein Aktienportefeuille nach verschiedenen Kriterien erstellen. Da Sie aber nicht alles auf eine Karte setzen wollen, verteilen Sie ihr Geld nach den folgenden Kriterien auf fünf Aktienpakete:

- Es kommen nur Unternehmen in Frage, die 1997 einen Nettogewinn ("net income") von mindestens 250 Mio \$ erzielt haben.
- Wählen Sie die Firmen mit dem höchsten ROE/Beta-Verhältnis.
- Beteiligen Sie sich proportional zur Mitarbeiterzahl.

*3 Möglichkeiten und Grenzen erkennen*

- a) Kritisieren Sie die Portfoliostrategie des Anlageberaters.
- b) Stellen Sie sich eigene Aufgaben. Suchen Sie dabei nach neuen Funktionen und Grenzen von *PowerPlay*.
- c) Welche Vor- und Nachteile hat ein OLAP-Tool wie *PowerPlay* im Vergleich zu QBE, SQL oder Tabellenkalkulationsprogrammen?
- d) Implementieren Sie die Aufgabe 2d in QBE und SQL.

---

# Funktionalität von *PowerPlay*

---

## Eingabe

Würfeldaten real oder als Teil eines RDBMS speichern

Realen Würfel auf einem PC oder einem Server speichern

Benutzerfreundliche DOLAP-Abfragen definieren

## Verarbeitung

Slicing and Dicing

Drilling Down and Up

Drilling Through mit Cognos *Impromptu*

Automatische PowerCube-Generierung

Zugriff auf relationale und mehrdimensionale DBMS

API für MS *Visual Basic*, MS *Excel* u.a. Pakete

## Ausgabe

zweidimensionale Tabellen

dritte Dimension durch Ebenen

Point and Click - Grafik

---

# ① Ausgewählte Kriterien für OLAP-Clients

---

## *Beispiele von OLAP-Clients*

- Brio Enterprise von [Brio Technology](#)
- Business Objects [Business Objects](#)
- PowerPlay von [Cognos](#)
- Express von [Oracle](#)
- Seagate Info von [Seagate](#)

## **Funktionalität**

- ✓ Drill Down, Up und Through
  - textuell (z.B. durch Doppelklick)
  - grafisch (z.B. durch Klick auf ein grafisches Element)
- ✓ Slice and Dice
- ✓ Filterung
- ✓ Sortierung
- ✓ Ebenen  
(dreidimensionale Anzeige)
- ✓ Verwaltung mehrerer Würfel bzw. Sternschemata
- ✓ Views
- ✓ Ad hoc-Berechnung temporärer Dimensionen
- ✓ Pivoting
- ✓ Point and Click Graphing
- ✓ Automatische Unterdrückung von Nullwerten

Dynamische Berichte

Agenten

(automatische Aktionen bei Vorliegen einer benutzerdefinierten Bedingung, zum Beispiel automatischer Versand von EMail, Fax oder Bestellungen)

## Integration

SQL-Schnittstelle

Integration von Software zu Tabellenkalkulation, Data Mining, Gruppenarbeit, GIS (Geographical Information Systems), ...

Integration von Technologien wie COM und WWW

Erweiterungsfähigkeit

(z.B. durch Visual Basic oder C++)

## Benutzerfreundlichkeit

Wahl von Indikatoren, Dimensionen und Kategorien durch Auswahl (zum Beispiel in einem Listenfeld) *und* Namensnennung

einfache Navigation in Kategorienhierarchien

Transparenz (Implementationsdetails für Benutzer nicht erkennbar)

...

---

## ② Ausgewählte Kriterien für OLAP-Server

---

### *Beispiele von OLAP-Servern*

Microsoft SQL Server 7 OLAP Services

Hyperion Essbase OLAP Server von [Hyperion Software](#)

IBM DB2 OLAP Server

MetaCube ROLAP Option von [Informix](#)

### **Architektur**

ein-, zwei- oder mehrstufig

relationale oder mehrdimensionale Objektdaten

ein einziger Würfel oder mehrere kombinierbare Würfel

### **Funktionalität**

Zahl und Art der unterstützten Frontends

Zahl und Art der unterstützten RDBMS-Quellen

verteilte Datenhaltung

Definition von Attributen zur Beschreibung von Indikatoren, Dimensionen, Kategorien und Werten möglich

Vielfalt der Datentypen (numerische und nichtnumerische, einfache und zusammengesetzte)

Strukturen und Operationen auf der Zeitdimension

ad hoc- und vorberechnete Zusammenfassungen

Aggregationen (insb. arithmetische Operationen)

Verknüpfung von Würfeln oder Sternschemata

Drill Through (Zugriff auf die originären Daten operativer Datenbanken über vordefinierte SQL-Anweisungen)



Drill Across (Zugriff über mehrere Data Marts)  
Sicherheitskomfort (v.a. Zugriffsberechtigungen)  
Kompatibilität zu OLAP Council-Empfehlungen

## **Wartbarkeit**

Ladeaufwand (z.B. inkrementelles Laden)  
Modellierungsaufwand (z.B. grafischer Datenbankentwurf)  
Optimierungsaufwand

## **Effizienz**

Ladezeit (zur Initialisierung und periodischen Nachführung)  
Antwortzeit  
Speicheraufwand (Höchstzahl von Zellen, Vorberechnungsgrad, Verwaltung dünn besetzter Matrizen)

## **Preis**

Software  
Schulung  
Beratung  
Wartung

---

# OLAP-Produkte und Hersteller

---

- Brio Enterprise von [Brio Technology](#)
- Business Objects [Business Objects](#)
- Business Information Warehouse von [SAP](#)
- Decision Support Suite von [Pilot](#)
- Decision Suite von [Information Advantage](#)
- DSS Agent von [Micro Strategy](#)
- Express von [Oracle](#)
- Holos von [Seagate](#)
- Hyperion Essbase von [Hyperion Software](#)
- Hyperion Enterprise von [Hyperion Software](#)  
(spezialisiert für Finanzanwendungen)
- Seagate Info von [Seagate](#)
- InfoBeacon von [Platinum](#)
- if...Synchrony von [if...](#)
- MetaCube von [Informix](#)
- PowerPlay von [Cognos](#)
- Produktpalette von [SAS](#)

## 5.31 Methode im Vergleich

Kriterium	AHP	Optimierung	OLAP	Regelbasierte Systeme	Induktion	Neuronale Netze	Regression
Methode breit anwendbar	∅	–	+	∅	∅	∅	–
Automatisierungsgrad	–	+	– <sup>1</sup>	∅	+	+	+
Ergebnis genau	–	+	+	∅	+	+	+
Unabhäng. Var. gewichtbar	–	–	–	–	∅	–	+
Lösungsweg begründbar	∅	–	∅	+	+	–	–
Methode plausibel	+	∅	+	+	∅	–	∅
Ergebnis einbettbar	–	+	∅	∅	+	∅	+
Entwicklungsaufwand	+	+	–	–	+	∅	∅
Rechnerbelastung	+	+	–	∅	∅	–	+

### Kriterien

<sup>1</sup> Hauptnachteil

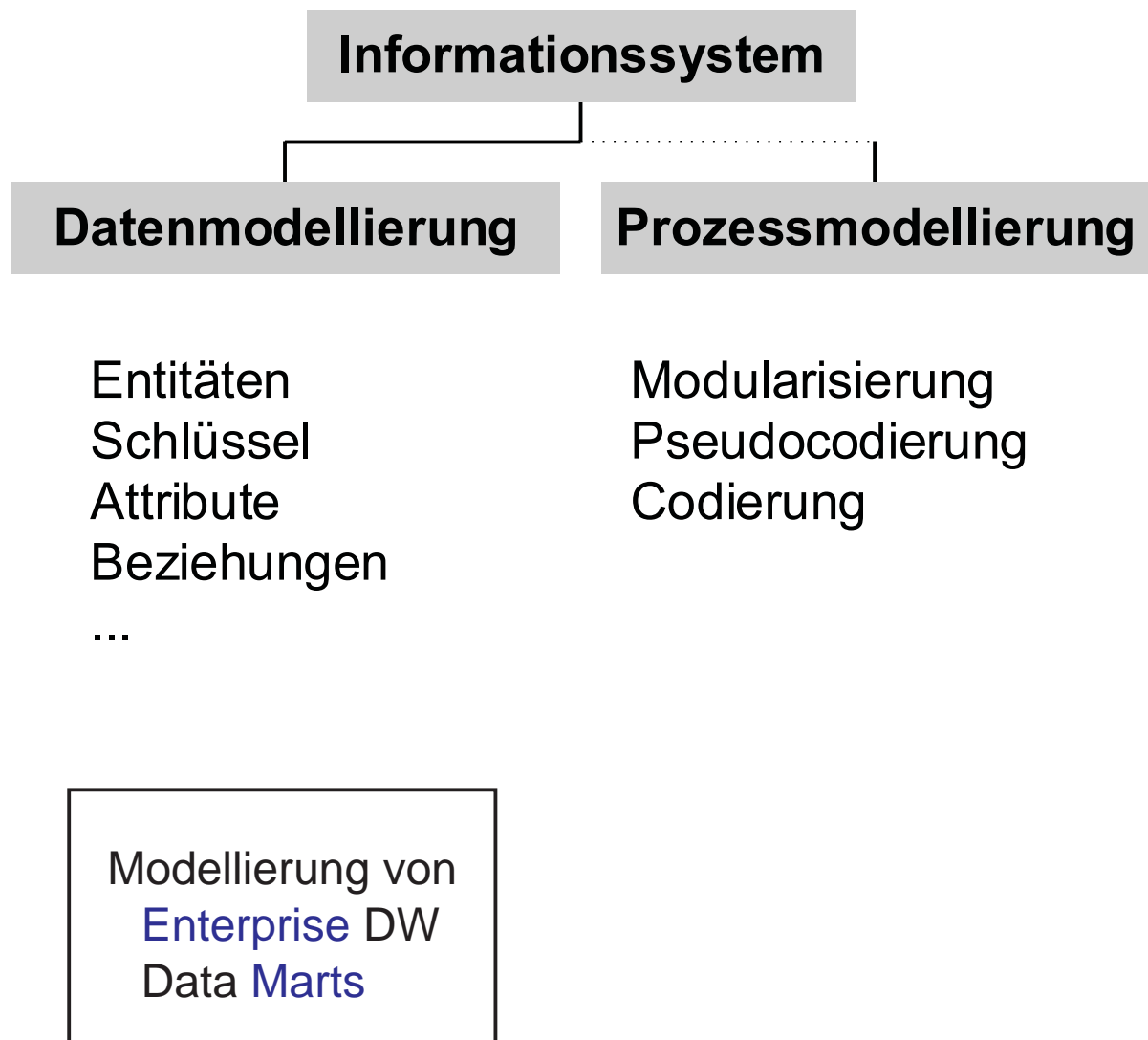
---

# Inhalt

---

Grundlagen	<a href="#"><u>2</u></a>
Entwicklung - Überblick	<a href="#"><u>3</u></a>
Data Marts und Enterprise Data Warehouse	<a href="#"><u>6</u></a>
Mehrdimensionale Daten	<a href="#"><u>21</u></a>
Endbenutzerzugriff	<a href="#"><u>34</u></a>
OLAP	<a href="#"><u>44</u></a>
ANLAGEBERATUNG mit <i>PowerPlay</i> 📌	<a href="#"><u>71</u></a>
Modellierung relationaler Data Warehouses	<a href="#"><u>84</u></a>
Modellierung von Informationssystemen	<a href="#"><u>85</u></a>
Sternschema ( LIEFERFRIST )	<a href="#"><u>97</u></a>
EDW-Modell und Sternschema ( VERKAUF )	<a href="#"><u>106</u></a>
<i>Synchrony</i> ( EINZELHANDEL 📌 )	<a href="#"><u>126</u></a>
Entwicklung und Betrieb	<a href="#"><u>136</u></a>
Fallbeispiel ( ZEITSCHRIFTEN 📌 )	<a href="#"><u>151</u></a>
Effizienz	<a href="#"><u>160</u></a>
Netzzugriff	<a href="#"><u>198</u></a>
Client/Server-Architekturen	<a href="#"><u>203</u></a>
Zugriff über Internet und Intranet	<a href="#"><u>213</u></a>

# Modellierung von Informationssystemen



---

# Operative und analytische Datenmodelle

---

## Operatives *Unternehmensdatenmodell*<sup>1</sup>

logische Verbindung grober Bereichsdatenmodelle

## Operative *Bereichsdatenmodelle*

detaillierte logische Entity-Relationship-Diagramme

## Enterprise Data Warehouse-Modell

globale Datenmodelle mit wenigen physischen Festlegungen

## Data Mart - Sternschemata und -Würfel

lokale Datenmodelle mit vielen physischen Festlegungen  
(v.a. stark denormalisiert)

---

<sup>1</sup> Die Erstellung von Unternehmensdatenmodellen ist aufwendig. Viele Unternehmen verzichten deshalb auf ein übergreifendes globales Datenmodell

---

# Unwünschte und wünschbare Redundanz

---

## ***Produktionsdatenbanken***

Anomalien minimieren

Redundanz *minimieren*

## ***Data Warehouses***

Benutzerfreundlichkeit und Abfrageeffizienz maximieren

Redundanz *eingeführen*

Redundanzquellen

# Data Warehouses führen zu Redundanz

<i>Redundanzquellen</i>	<i>Gründe der Einführung von Redundanz</i>
<b>Analytische</b> Daten entstehen aus <b>operativen</b> Daten	benutzernahe Datenmodellierung Zugriffseffizienz
<b>Lokale</b> analytische Daten entstehen aus <b>zentralen</b> analytischen Daten	gemeinsame Quelle für Data Marts benutzernahe Datenmodellierung Zugriffseffizienz
<b>Denormalisierte</b> entstehen aus <b>normalisierten</b> Tabellen	benutzernahe Datenmodellierung Zugriffseffizienz
<b>Abgeleitete</b> Daten wiederholen <b>Rohdaten</b>	Zugriffseffizienz ((Bsp. Aggregate) benutzernahe Datenmodellierung (Bsp. Zeitreihen)



---

# Ziele und Mittel der Datenmodellierung

---

## Produktionsdatenbanken

**Transaktionsverarbeitung** durch Granularisierung

**Fortschreibungsfreundlichkeit** durch Normalisierung

**Speichereffizienz** durch Normalisierung

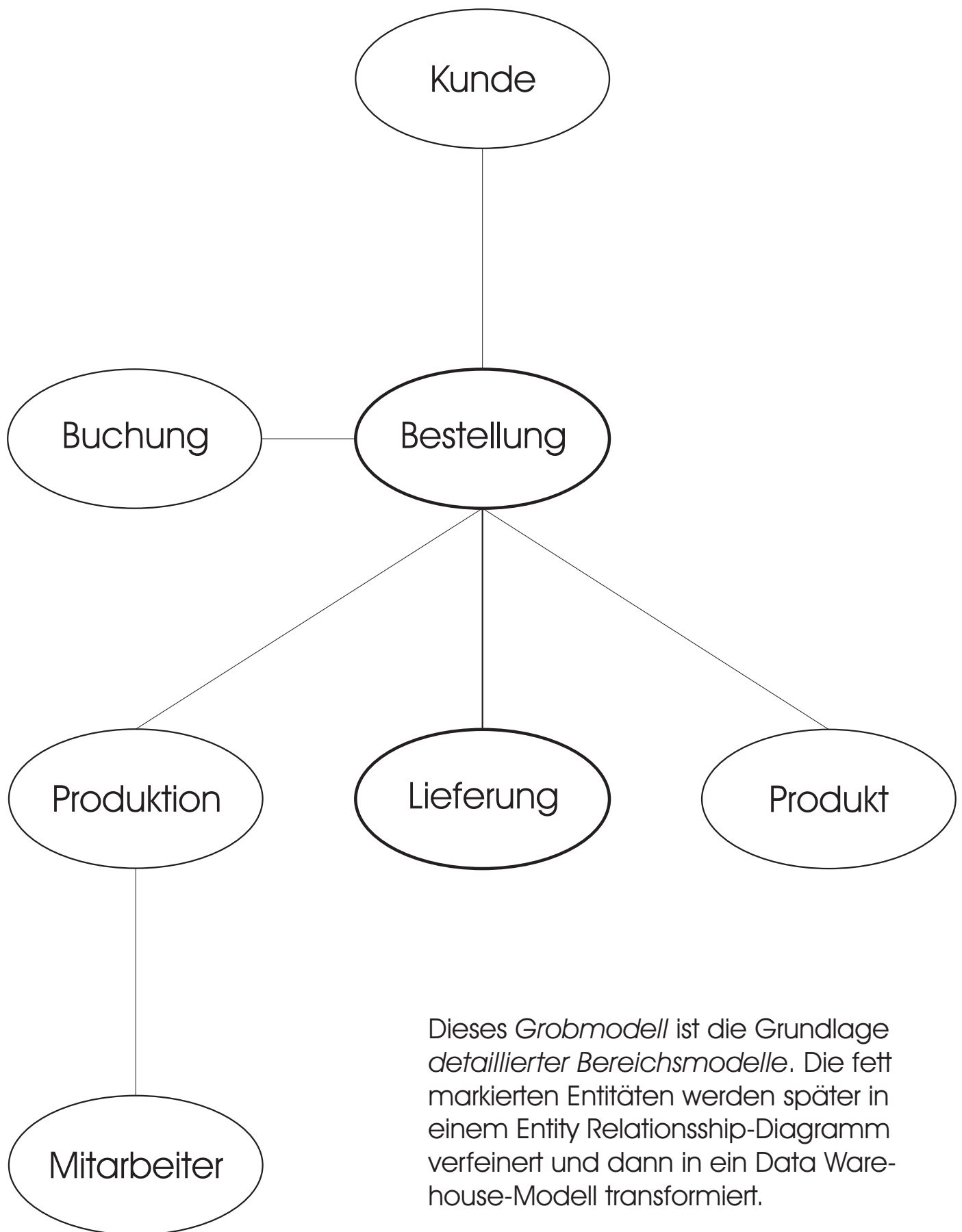
## Data Warehouses

**Benutzerfreundlichkeit** durch Aggregation

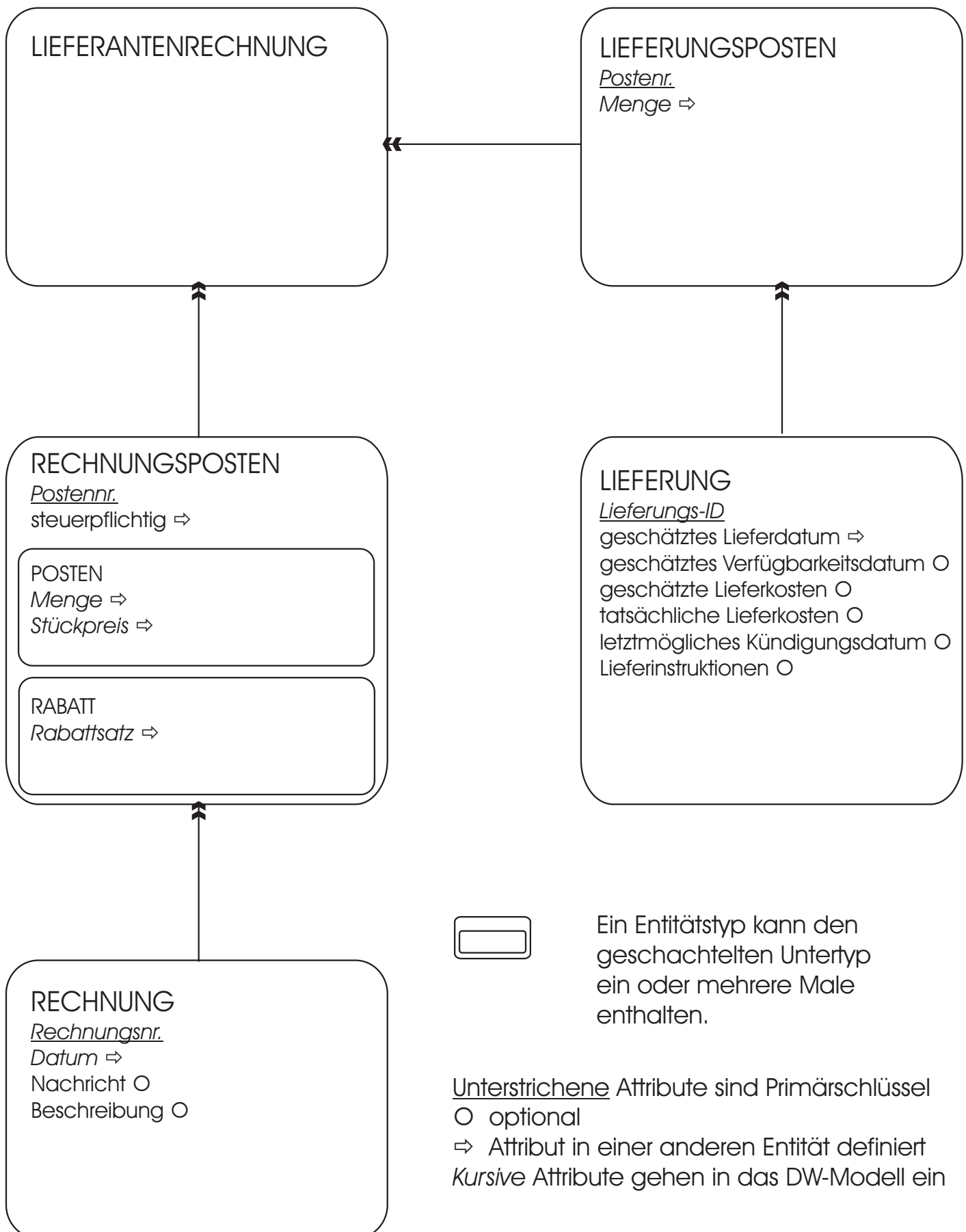
**Abfrageeffizienz** durch Denormalisierung

**Betriebsnähe** durch Indikatoren und ihre Dimensionen

## 5.32 Ein operatives Unternehmungsdatenmodell



## 5.34 Ein operatives Bereichsmodell *Fakturierung*



## Produktionsdatenbanken

### Data Warehouses

#### ① **Logische Modellierung**

Ziel sind **abbildungstreue** *logische* Schemata

##### **Zentrales Data Warehouse**

*Anforderungsspezifikation und logisches Schema*

⇒ Entity-Relationship-Diagramme

##### **Data Marts**

*Anforderungsspezifikation und logisches Schema*

⇒ Sternschemata für relationale DW

⇒ Würfel für mehrdimensionale DW

#### ② **Physische Modellierung**

Ziel sind **effiziente** *physische* Schemata

# Transformationen des operativen Modells

Künftiger **Nutzen** eines Attributs  
oder einer Beziehung

*versus*

Einbusse an **Effizienz und Wartbarkeit**  
von Data Warehouses

**Operative** Daten weglassen

*Rechnungsstatus* (z.B. offen, bezahlt, Zahl der Mahnungen)

**Zeitdimension** integrieren

*Ladedatum, Zeitkategorien*

**Attribute** berechnen

einfache Vorberechnungen wie *Betrag = Menge \* Stückpreis*

Verdichtungen (z.B. Aggregationshierarchien)

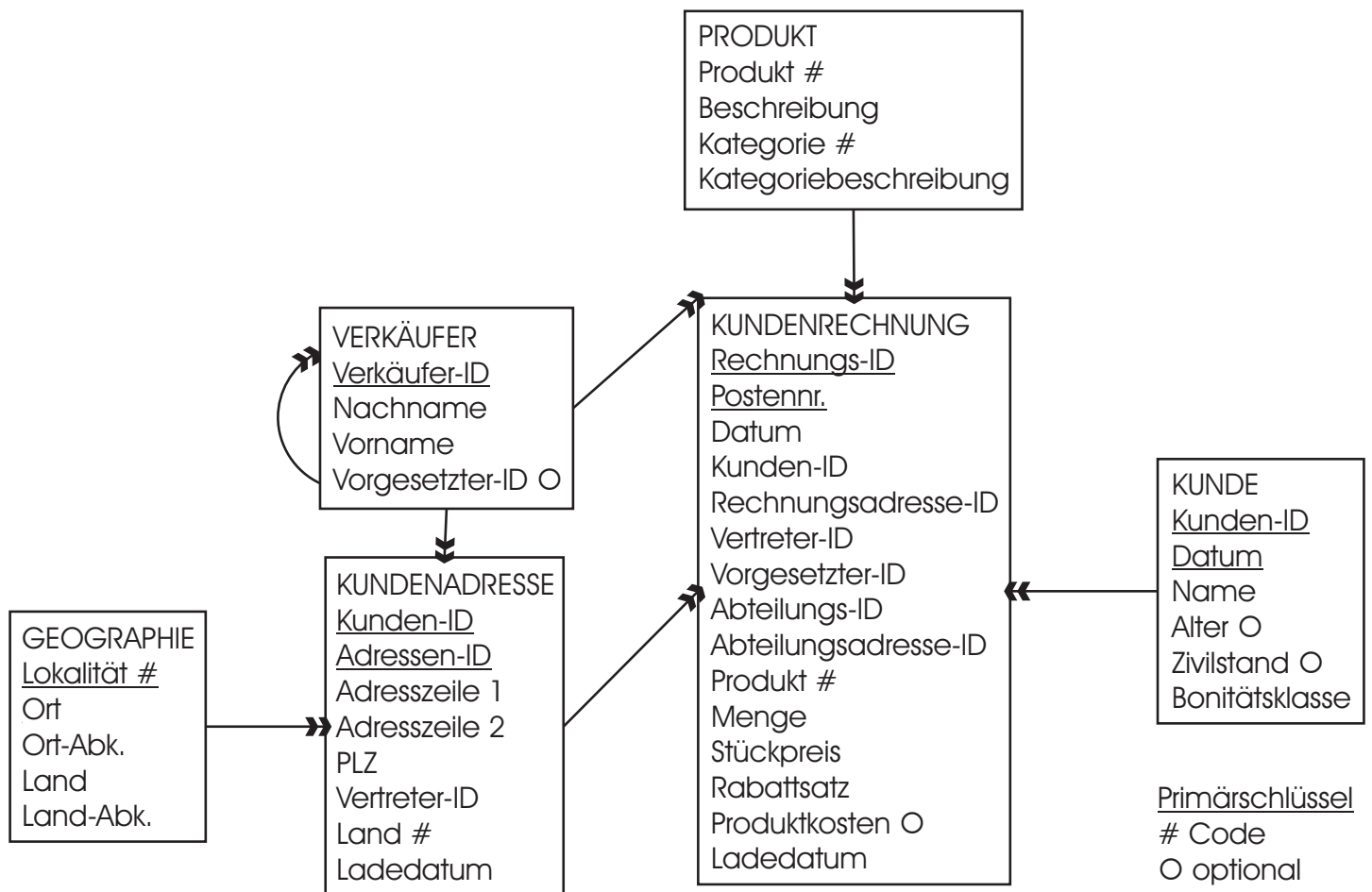
Umgruppierungen

**Verbundoperationen** verringern

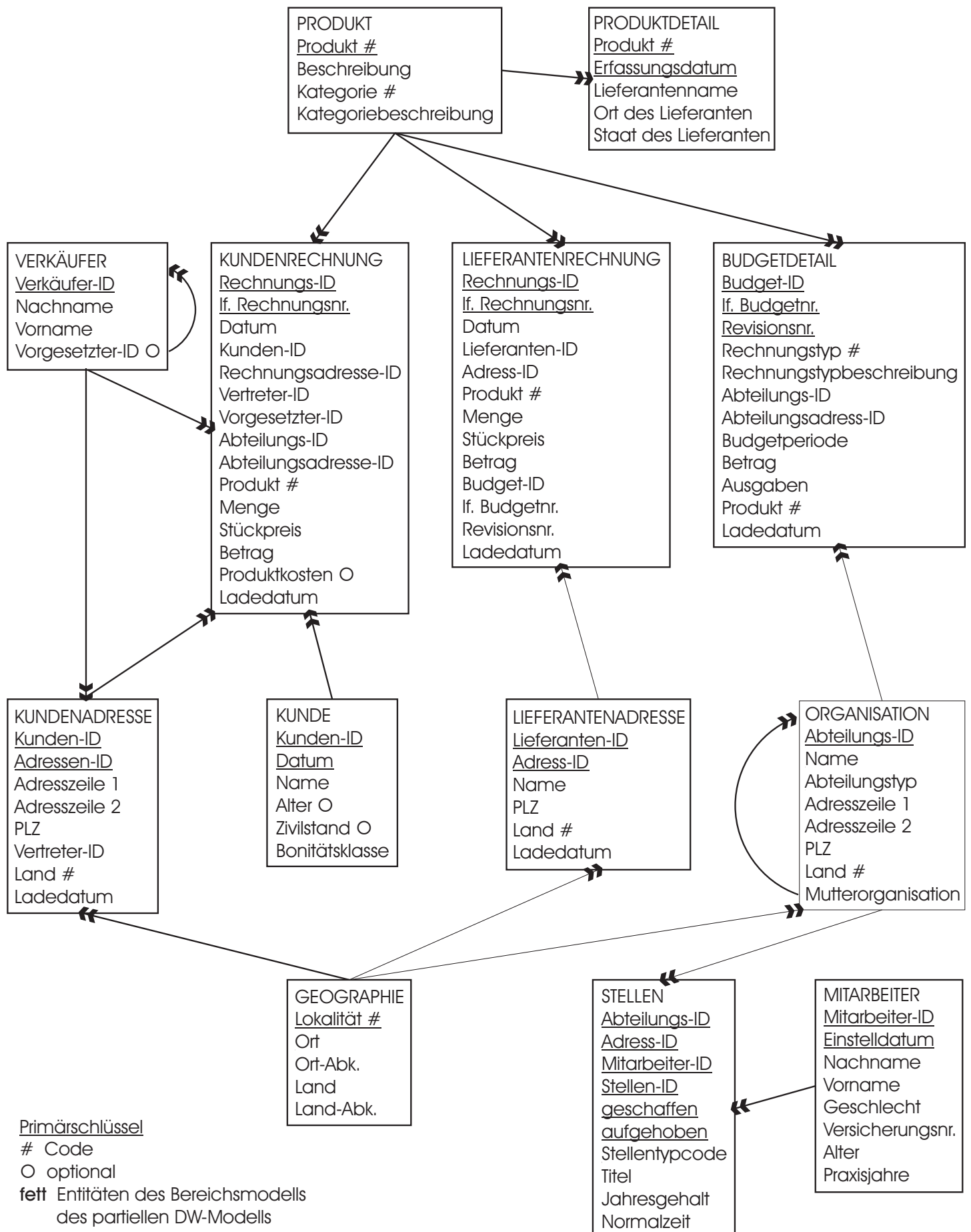
z.B. in LIEFERUNG und LIEFERUNGSPOSTEN

...

## 5.35 Operatives Bereichs- partielles DW-Modell



## 5.36 Vom partiellen DW-Modell zum EDW-Modell



## 5.37 Logische Modellierung im Vergleich

	<i><b>Operative Daten</b></i>	<i><b>Analytische Daten</b></i>
<i>Ziel</i>	Redundanz-minimierung	Abbildungstreue und Abfragekomfort
<i>Logisches Datenmodell</i>	normalisiert	denormalisiert und mehrdimensional
<i>Ergebnis</i>	viele änderungs-freundliche Tabellen mit je wenigen Attributen	grosse <u>Faktentabellen</u> aus meist stetigen numerischen Attributen kleine <u>Dimensionstabellen</u> aus meist symbolischen Attributen

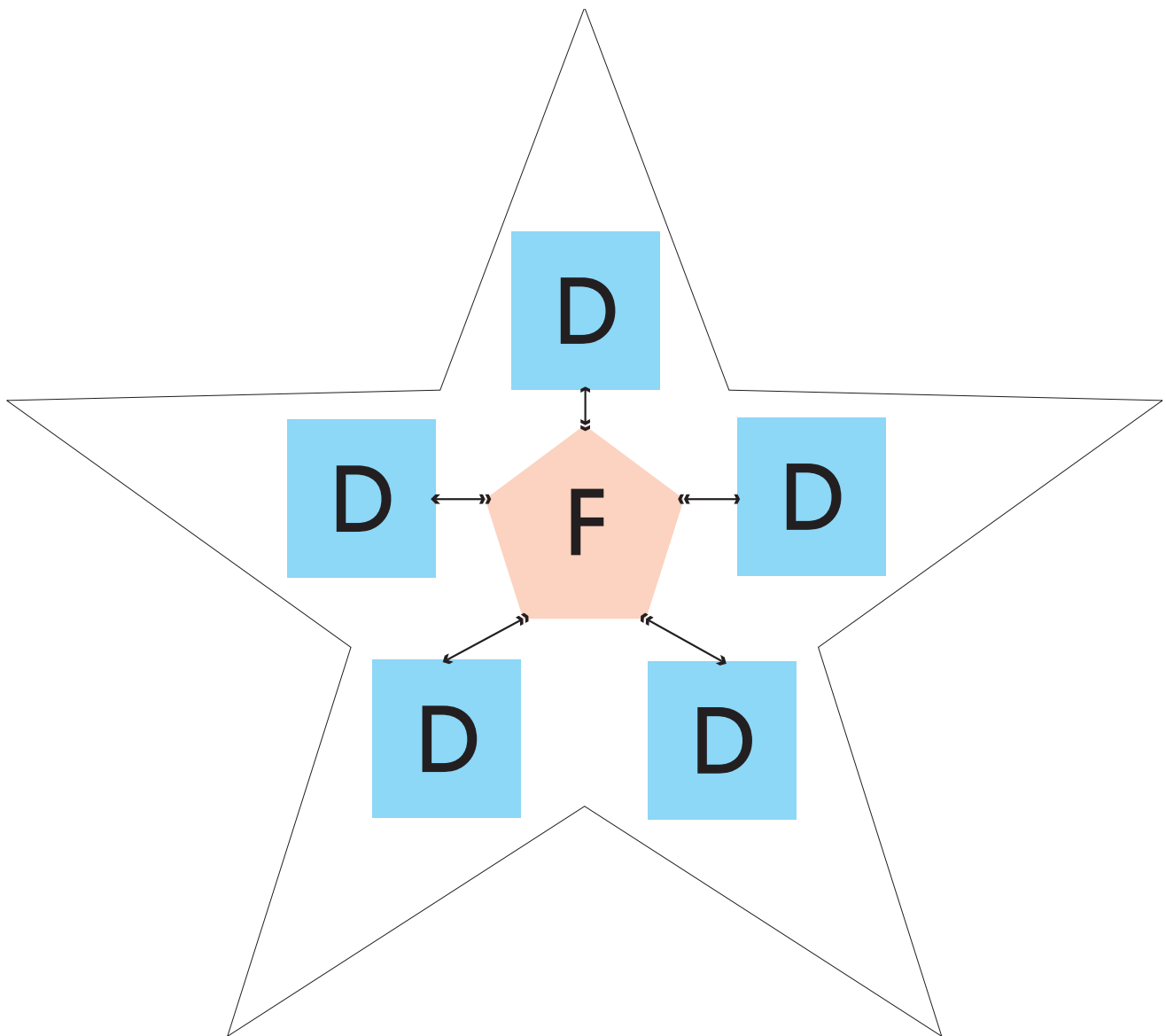
**Logische** Datenmodelle ermöglichen ...

- **abbildungstreue** Tabellenstrukturen
- benutzerfreundliche **Abfragen**

Würfel für mehrdimensionale Data Marts  
Sternschemata für *relationale* Marts



## 5.38 Sternschemata für relationale Data Marts



**Sternschema** := logisches Datenbankschema, das die **Dimensionstabellen** (D) eines *relationalen* DW abfragefreundlich um eine **Faktentabelle** (F) ordnet

Dutzende von Sternschemata für ein einzige EDW

## Tabelleninhalte

### Verfügbare Daten

(aus operativen Modellen, Interviews mit Datenbankadministratoren, externen Quellen)

Beispiele	Attribute
POS-Tagesabschlüsse und andere operative Daten	<b>Mengenumsatz</b> <b>Geldumsatz</b> <b>Kundenzahl</b>

+

### Wünschbare Auswertungen

(aus Interviews mit Endbenutzern)

Beispiele	Fakt pro Dimension
Zeitvergleich	<b>VERKÄUFE</b> pro <b>PERIODE</b>
Produktvergleich	<b>VERKÄUFE</b> pro <b>PRODUKT</b>
Lieferantenvergleich	<b>VERKÄUFE</b> pro <b>LIEFERANT</b>
Marktkorbanalyse	<b>VERKÄUFE</b> pro <b>KUNDE</b>

## 5.40/5.41 Anforderungsdiagramm

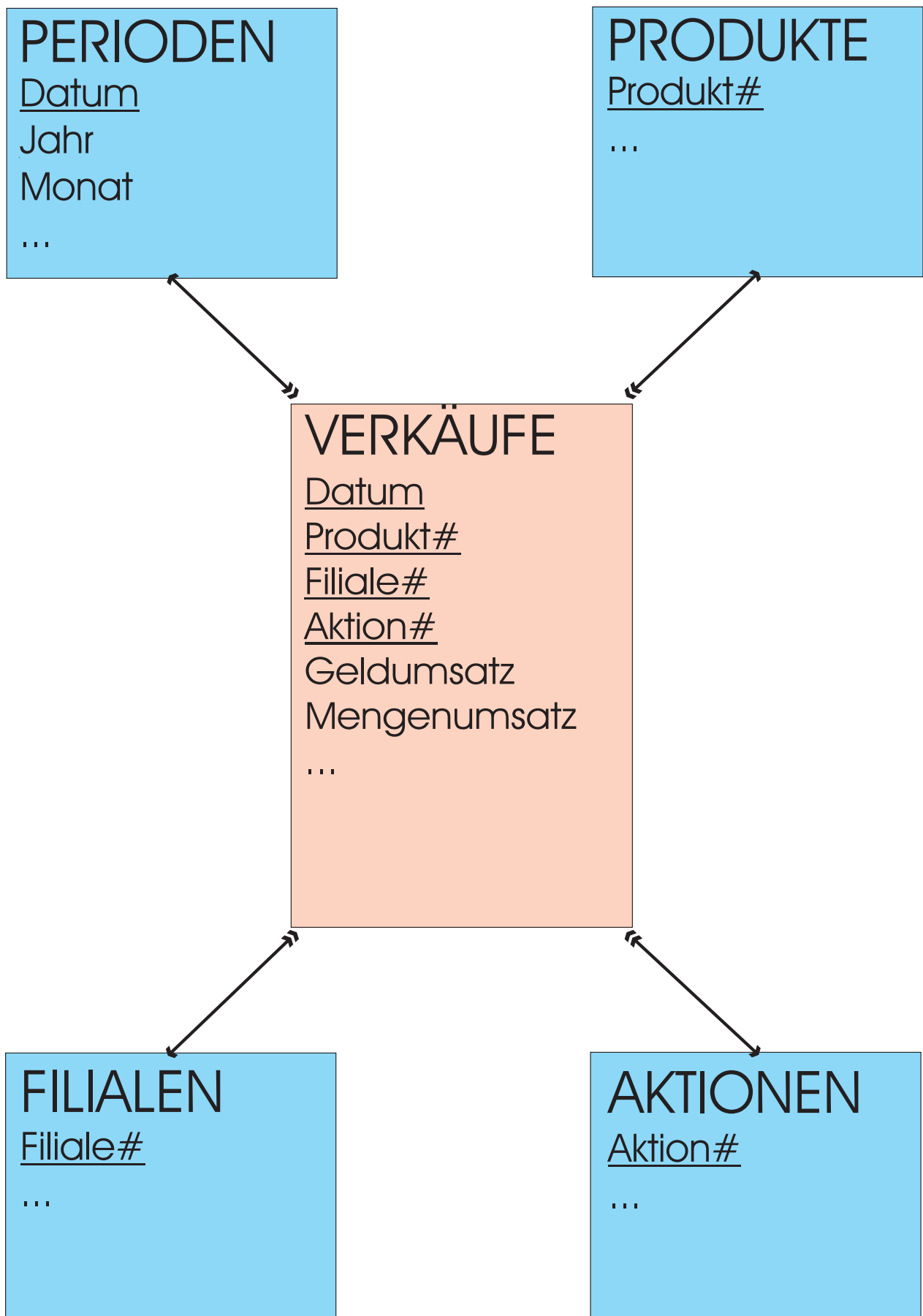
### Beispiel

<b>Mengenumsatz, ...</b>			
<b>Produkt</b>	<b>Periode</b>	<b>Ort</b>	<b>...</b>
Produktklasse	Jahr	Land	
Produktgruppe	Quartal	Region	
Einzelprodukt	Monat	Filiale	
...	...	...	

### Erläuterung

<i>Komponente</i>	<i>Beispielfragen</i>	<i>Definition</i>
<b>Indikatoren</b>	Wie gut?	Attribute, die Unternehmungsergebnisse bewerten
<b>Dimension</b>	Was?, Wann?, Wo?	Attribut, entlang welchem ein Fakt gemessen werden kann
<b>Kategorie</b>	Wie genau ?	Wertebereiche der Dimension

## 5.42 Ein Sternschema für HANDEL



## 5.43 Mehrere Faktzeilen pro Dimensionszeile

### Mehrere VERKÄUFEzeilen

<u>Datum</u>	<u>Produkt#</u>	<u>Filial#</u>	<u>Aktion#</u>	Mengenumsatz/Tag	...
2.3.98	231	14	3	1'863	...
3.3.98	231	14	4	533	...
...	...	...	...	...	...



### Eine PRODUKTE-Zeile

<u>Produkt#</u>	Produktkategorie	Produktname	Farbe	...
231	Mountain Bike	"Colorado"	Farbe grün	...
...	...	...	...	...

Produkt #231 kommt in mehreren VERKÄUFEzeilen vor

## ① VERKÄUFE

Schlüssel aus vier Fremdschlüsseln vermittelt  
1:n- Beziehungen zu den Dimensionstabellen

## ② ZEIT (bzw. PERIODE)

Dimensionstabelle kann berechenbare und nicht  
berechenbare Attribute enthalten

*Werktag, Feiertag, Saisonzeit, Steuerperiode*

Dimension enthält oft mehrere Kategorien

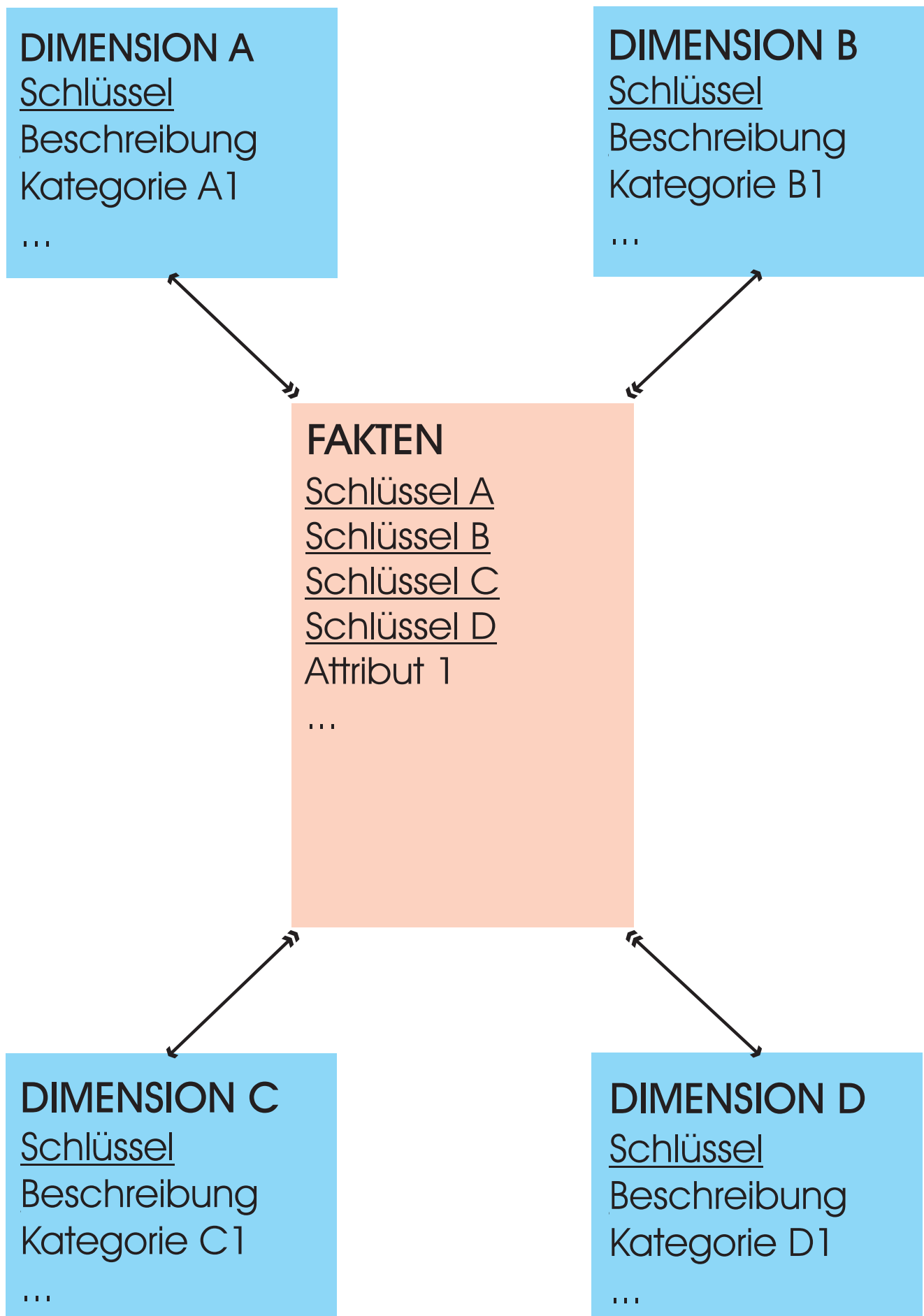
*Jahr, Quartal, Monat, Woche und Tag*

## ③ PRODUKTE

## ④ FILIALEN

## ⑤ WerbeAKTIONEN

## 5.44 Sternschema allgemein



## 5.45 Sternschemata filtern Abfragen ...

### 1. *Dimensionstabelle* PRODUKTE

Produkt#	Name	Gewicht	...
1	A	5	...
2	B	4	...
3	C	7	...
...	...	...	...

### 2. *Einschränkung durch Selektion (Filterung)* (Produkt = 2 oder 3) und (Gewicht = 4)

### 3. *Eingeschränkte Dimensionstabelle* PRODUKTE

Produkt#	Name	Gewicht	...
2	B	4	...

### 4. *Abfrage auf einer Faktentabelle*



---

## LIEFERFRIST - Sternschema (A 5.3)

---

Die *Kundenzufriedenheit* hängt nicht zuletzt von der Einhaltung der *Lieferfristen* ab. Ihre Überwachung gehört deshalb zu den wichtigsten Aufgaben eines Data Warehouse. Aus der sorgfältigen Diagnose nicht eingehaltener Fristen kann sich zum Beispiel die folgende Therapie ergeben :

- Erhöhung der Lagerbestände
- Änderung der Transportmittel und -wege
- zuverlässigere Berechnung der Lieferfristen.

Eine Unternehmung mit mehreren Produktionsstandorten möchte die *Diagnose von Lieferverzögerungen* in ihre Data Warehouse-Lösung integrieren.

- a) Erstellen Sie ein *Anforderungsdiagramm* mit mindestens fünf Dimensionen und mehreren importierten und berechneten Fakten. (Am besten versetzen Sie sich in die Unternehmungsleitung und fragen nach Gründen von Lieferverzögerungen)
- b) Zeichnen Sie ein *Sternschema*

# VERKAUF - EDW-Modell und Sternschema (A 5.4)

Betrachten Sie das Entity-Relationship-Diagramm 5.47. Es enthält einen Ausschnitt aus dem Datenmodell eines Enterprise Data Warehouse.

## 1. Datenmodell des Enterprise Data Warehouse

- a) Woran erkennen Sie, dass der EDW-Ausschnitt von Bild 5.47 nicht aus dem Datenmodell einer Produktionsdatenbank stammt?
- b) Welche Schritte durchläuft die Transformation eines operativen Datenmodells in ein EDW-Datenmodell?
- c) Definieren Sie anhand des EDW-Datenmodells eine SQL-Abfrage, welche die Produkte und die jeweiligen Mengen auflistet, die Personen mit einem Wohnsitz in der Schweiz im Jahre 1998 gekauft haben.

## 2. Sternschema des Funktionsbereichs VERKAUF

- d) Erstellen Sie aus dem Datenmodell ein Sternschema. Der Verkaufsleiter will Zeit-, Produkt-, Kunden-, Verkäufer- und Regionalvergleiche erstellen. Ausserdem möchte er wissen, ob das Kreditlimit, das Alter des Kunden oder der Zivilstand den Absatz beeinflussen.
  - Tragen Sie die Fakten, Dimensionen und Kategorien in ein Anforderungsdiagramm.
  - Ordnen Sie die Dimensionstabellen um die Faktentabelle an und definieren Sie deren Beziehungen.
- e) Definieren Sie die SQL-Abfrage aus Aufgabe 1 c) auf dem Sternschema. Beschreiben Sie die Unterschiede zwischen den beiden Abfragen. Weshalb wäre die gleiche Abfrage mit einem OLAP-Werkzeug wie PowerPlay benutzerfreundlicher?

### 3. Anwendungsbereich “Strategische Planung”

Eine Analyse der Absatzmengen eines Produkts nach der Dimension Zeit stellt Daten zur Lückenplanung (engl. gap analysis) bereit.

- f) Genügen die in unserem Sternschema vorhandenen Daten auch für weitere strategische Planungsinstrumente? Wenn nicht, welche Daten fehlen?
- g) Beschreiben Sie die Dimensionen und Fakten zur Erstellung eines Boston Consulting Group-Portfolio.

# Relationales OLAP erfordert Sternschemata

## Beispielwerkzeug

### Family Panel

Sternschema für eine *Familie* von Abfragen (ein Sternschema)

### Report Panel

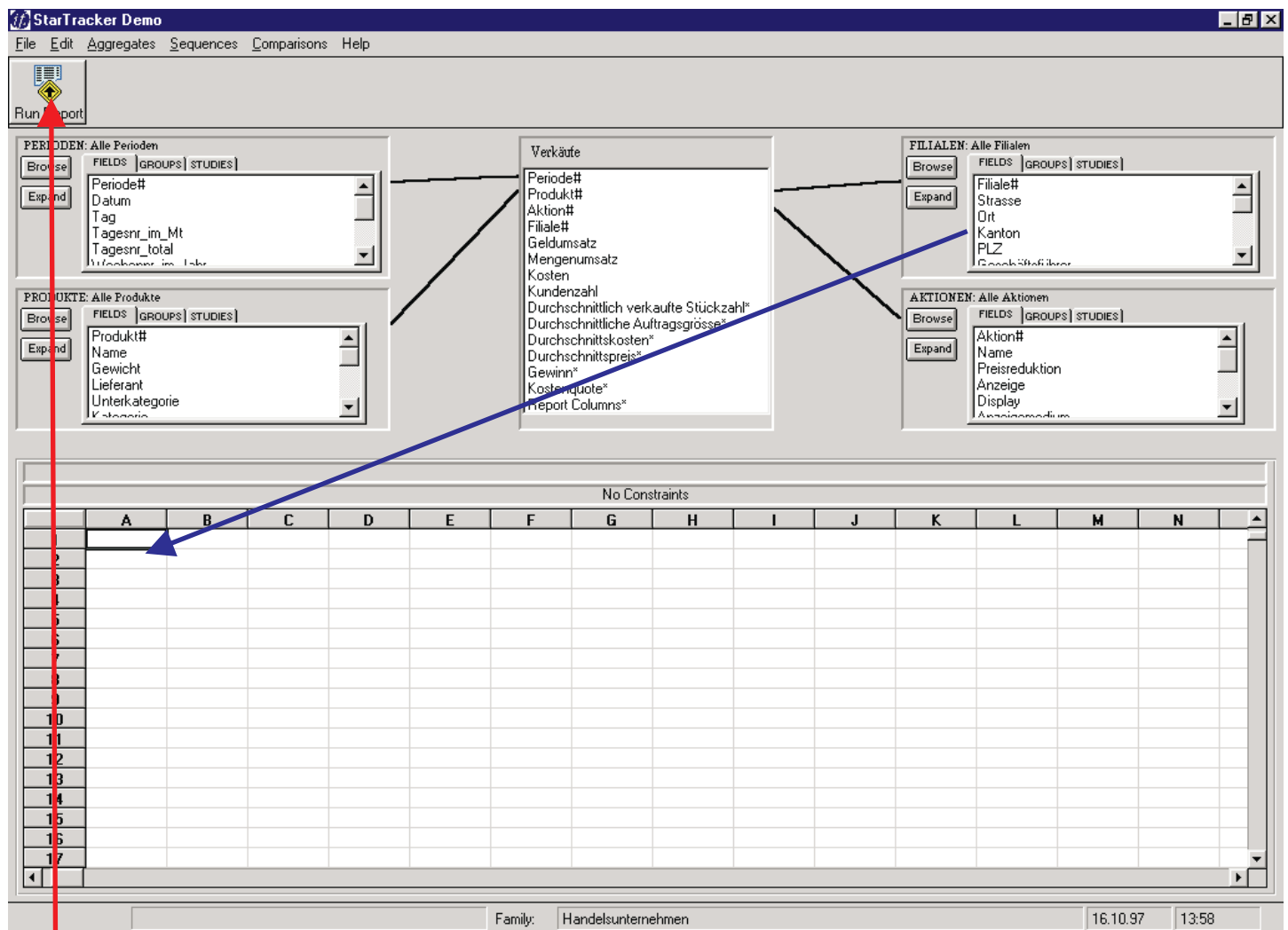
Ergebnisse einer *Abfrage* (hier engl. report)

*if... Synchrony*

The screenshot displays the StarTracker Demo application interface. The top window, titled "StarTracker Demo", shows a star schema design with four fact tables connected to dimension tables. The fact tables are "Verkäufe", "PERIODEN: Alle Perioden", "FILIALEN: Alle Filialen", and "AKTIONEN: Alle Aktionen". The dimension tables are "PERIODE", "PRODUKTE: Alle Produkte", "FILIALE", and "AKTION". The "Verkäufe" table is the central fact table, connected to the other four dimension tables. The "PERIODE" table contains fields: Periode#, Datum, Tag, Tagesnr\_im\_Mt, Tagesnr\_total, and Wochen\_im\_Lkr. The "PRODUKTE" table contains fields: Produkt#, Name, Gewicht, Lieferant, Unterkategorie, and Kategorie. The "FILIALE" table contains fields: Filiale#, Strasse, Ort, Kanton, PLZ, and Geschäftsfiliale. The "AKTION" table contains fields: Aktion#, Name, Preisreduktion, Anzeige, Display, and Aktionsmedium. The bottom window shows a report panel with a table structure. The table has columns labeled A through N and rows numbered 1 through 17. The table is currently empty, with the header row (row 1) showing column A. The status bar at the bottom indicates "Family: Handelsunternehmen", "16.10.97", and "13:58".

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														
15														
16														
17														

# OLAP-Abfragen auf einem Sternschema



1 **Attribute** vom *Family Panel* in den *Report Panel* **ziehen**

2 Knopf "Run Report" **klicken**

## Eine SQL-Abfrage auf einem Sternschema

```
SELECT      PRODUKT.Name,  
            SUM ( VERKÄUFE.Geldumsatz )  
FROM        VERKÄUFE, PRODUKTE, PERIODEN  
WHERE       PERIODEN.Monat = VERKÄUFE.Monat  
GROUP BY    PRODUKT.Name  
ORDER BY    PRODUKT.Name1
```

```
SELECT      Fakt- oder Dimensionsattribut  
FROM        Fakt- oder Dimensionstabellen  
WHERE       Bedingung  
GROUP BY    Fakt- oder Dimensionsattribut  
ORDER BY    Fakt- oder Dimensionsattribut
```

SQL ist für den Endbenutzer *schlecht* geeignet

Benutzerfreundlichere Abfragen durch  
Sternschemata und SQL-Generierung

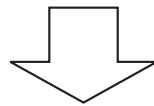
<sup>1</sup> Beschreiben Sie die SQL-Abfrage umgangssprachlich

## 5.48 Sternschemata durch Denormalisierung



Denormalisierung von Produktionsdaten führt zu

- mehr *Redundanz*, aber
- weniger Verbundoperationen



Dimension FILIALE



Fakten VERKÄUFE



---

# Stern- versus normalisiertes Schema I

---

## Normalisiertes Schema

### Ziele

- Redundanzminimierung
- Effiziente Transaktionsverarbeitung  
(Zugriff auf den einzelnen Satz)

### Ergebnis

- Komplexes, globales und idiosynkratisches Schema
- Viele Entitäten und Beziehungen  
(Hunderte für eine grössere Unternehmung)

## Sternschema

### Ziel

- Benutzerfreundliche Abfrage  
(Zugriff auf Aggregate, verständlicheres Datenmodell)

### Ergebnis

- Einfaches, lokales und standardisiertes Datenmodell  
(10-25 Sternschemata für grössere Unternehmungen)
- 1 Faktentabelle und wenige Dimensionstabellen  
(i.d.R. 5-15)



# Stern- versus normalisierte Schemata II

Sternschemata sind meist ...

abfragefreundlicher

- Weniger Tabellen  $\Rightarrow$  einfachere *Abfragen*
- Fakten und Dimensionen  $\Rightarrow$  Betriebsnähe

effizienter

- Weniger Tabellen  $\Rightarrow$  weniger *Verbunde*

aber ...

unvollständig

bilden zum Beispiel nicht alle Beziehungen ab

wartungsaufwendig

insbesondere änderungsfeindlich

---

## Vom normalisierten zum Sternschema

---

Ein Sternschema für jeden **Geschäftsprozess**

Fakttabellen aus den **aggregierbaren** Attributen  
mit **m:n**- Beziehungen

Übrige Attribute in Dimensionstabellen  
denormalisieren und **direkt** mit der  
Fakttabelle verbinden

# Erweiterte Sternschemata

Mehrere **Fakten**tabellen, insb.  
m:n-Verbindungstabellen

und / oder

**Normalisierte** Unterdimensionen

und / oder

**Kategorientabellen**

Erweiterte Sternschemata heissen auch **Schneeflockenschemata**

# Beurteilung erweiterter Sternschemata

## Vorteile

Sonst nicht darstellbare Attribute und Beziehungen

Update (weniger Redundanz)

Speicherplatz

## Nachteile

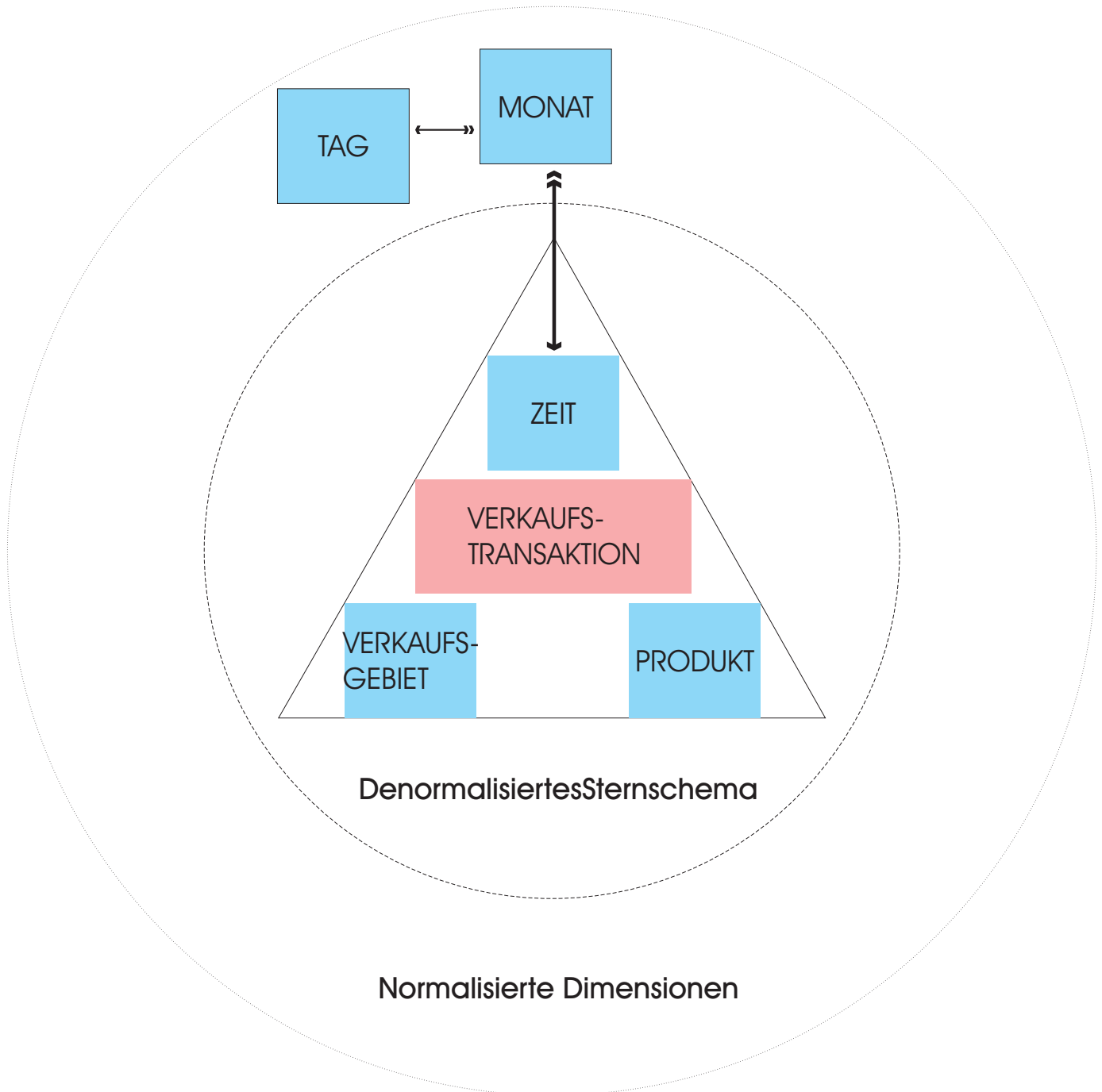
Benutzerfreundlichkeit

Abfrageeffizienz

Sparsam einsetzen!

Bsp. Unterdimension *Zeit* →

## 5.49 Erweitertes Sternschema - Normalisierung



### Denormalisierte ZEIT

ZEIT (Jahreszahl, Monatszahl, Tageszahl, Monatsname, Tagesname, Werktag?)

### Normalisierte ZEIT

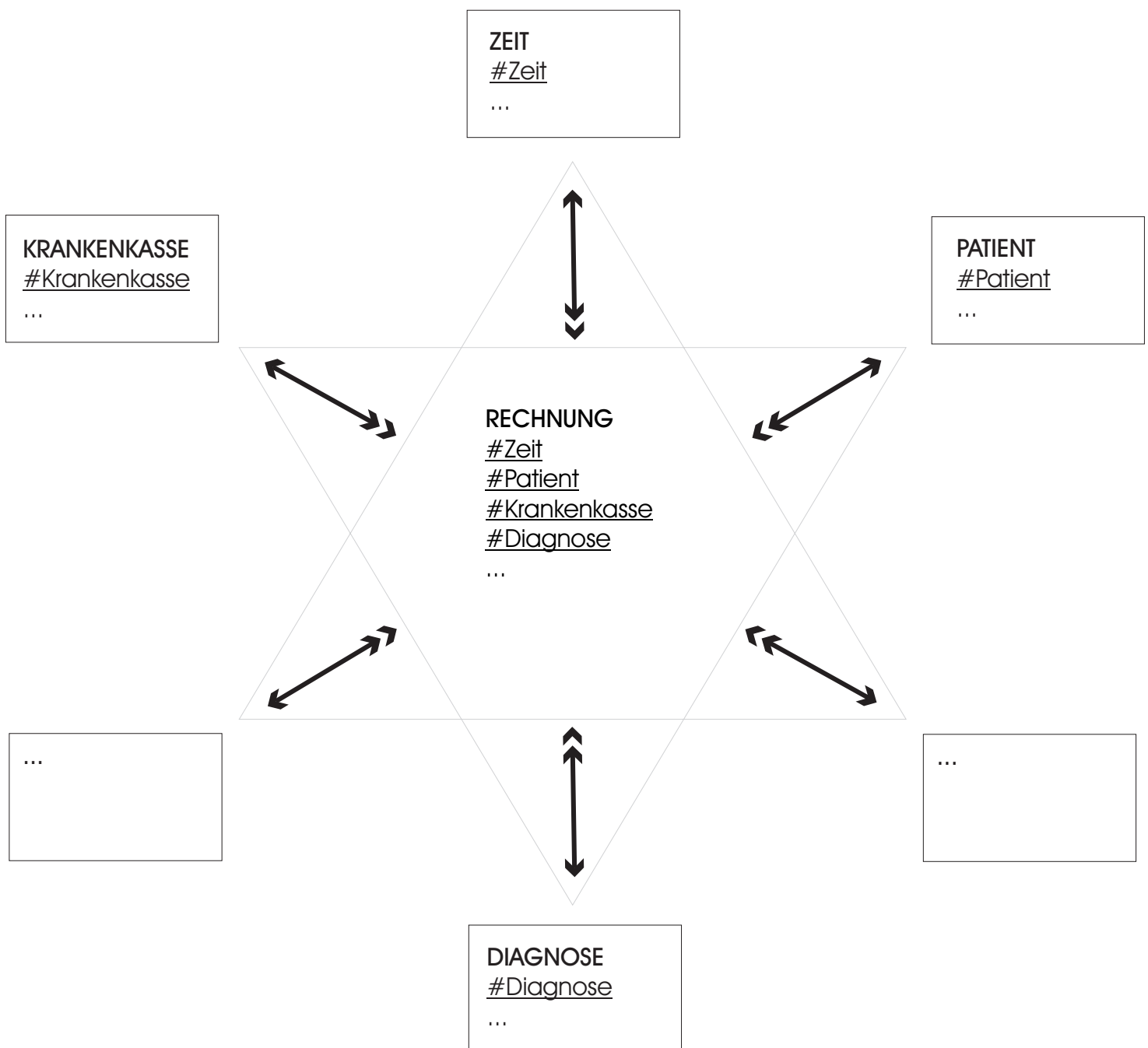
ZEIT (Jahreszahl, Monatszahl, Tageszahl, Werktag?)

MONAT (Monatszahl, Monatsname)

TAG (Tageszahl, Tagesname)

## Erweitertes Sternschema - m:n-Beziehung (A 5.6)

Bild 5.51 bildet die Rechnungsstellung eines Spitals auf ein Sternschema ab. Es sieht pro Patient nur eine einzige Diagnose vor, weil der zusammengesetzte Schlüssel von RECHNUNG nur den Fremdschlüssel #Diagnose enthält. Schlagen Sie ein Schema vor, dass mehrere Diagnosen (sogenannte Diagnosegruppen) erlaubt.



---

## KONTO - Auflösung einer m:n-Beziehung (A 5.7)

---

Gegeben seien eine Faktentabelle KONTENSTAND und eine Dimensionstabelle KUNDE. Der Kontenstand wird monatlich erfasst.

- a) Begründen Sie, weshalb zwischen den beiden Tabellen in der Regel eine m:n-Beziehung besteht.
- b) Zeichnen Sie den Ausschnitt des Sternschemas, welcher die beiden Tabellen und ihre Beziehung abbildet.

# Metadaten

---

Metadaten :=

Daten

über

die “eentlichen” Daten (Objektdaten) und  
die “eentlichen” Prozesse (Objektprozesse)  
der Informationssysteme einer Unternehmung

Wie kann ich Metadaten über *verschiedenste*  
Objektdaten und -prozesse in einer *einheitlichen*  
Notation und *einfach* zugreifbar verwalten?



<i>Metadatenklasse</i>	<i>Beispiele</i>	<i>Anwendungen</i>
Bezeichner	Tabellen, Attribute, Abfragen	logische Datenmodellierung
Datentypen	ASCII/ EBCDIC, Genauigkeit numerischer Typen	Ladeprozess
Beziehungen	1:1, 1:n, m:n	logische Datenmodellierung
Volumen	Datenmenge, Fortschreibungshäufigkeit, Partitions Grenzen	physische Datenmodellierung
Transformationen	Aggregationsregeln	Ladeprozess
Quelle und Ziel	Quellen operativer Daten	Ladeprozess
Verantwortliche	Datenbankadministrator eines Data Mart	Entwicklung und Betrieb
Benutzersichten und Zugriffsberechtigungen	Zugriff auf die Personaldaten	Administration



# Vorkommen von Metadaten

Interaktive benutzerfreundliche Abfragen



Lokale  
Datenmodelle

Unternehmungs-  
Datenmodelle



Einzelrechner-Daten

Client-Daten

Server-Daten

Programmdaten

CASE-Daten

...

# Data Dictionary

---

**Data Dictionary** (DD) :=

Datenbank, die

Metadaten

zu **Form** und **Inhalt** von

**Applikationen** enthält

Synonyme: Katalog, Repository

# Data Dictionary-System

---

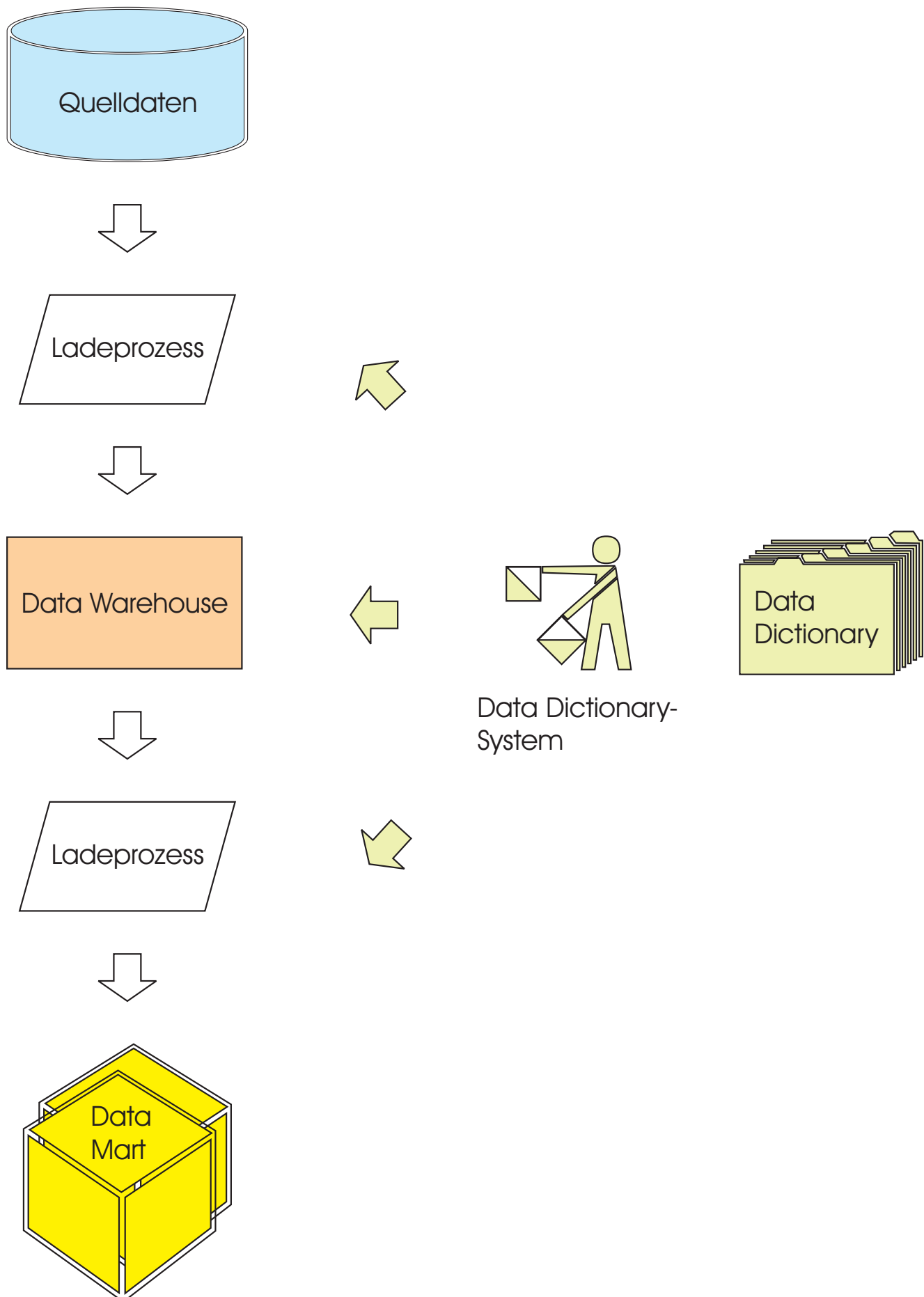
Data Dictionary-System :=

- ✓ Informationssystem, das
- ✓ Metadaten
- ✓ in einem Data Dictionary
- ✓ so verwaltet,
- ✓ dass es Softwareentwicklung und -wartung möglichst gut koordiniert

Bsp. Ladeprozesse zwischen ...

- operativen Datenbanken und Data Warehouse
- Enterprise Data Warehouse und Data Marts

## 5.53 Data Dictionary-System



# Anforderungen an DD-Systeme

## Klassen von Data Dictionary-Systemen

- *Allgemeiner* DD (für mehrere Werkzeuge und Hersteller)
- *Dedizierter* DD (für ein Werkzeug oder einen Hersteller)  
z.B. Metadaten des ROLAP-Werkzeugs *Synchrony*
- *Gateway* zur Verbindung lokaler DDs

## Anforderungen

- ✓ Metamodell standardisiert  
z.B. *Open Information Model* auf Initiative von Microsoft,  
*Metadata Interchange Initiative* von führenden DW-Anbietern
- ✓ Metamodell erweiterbar (z.B. Entity Relationship Model)
- ✓ Impact Analysis  
Rückwirkungen von Änderungen?
- ✓ Versioning  
gleichzeitige kontrollierte Arbeit an mehreren Versionen
- ✓ benutzerfreundliche Abfragen und Berichte  
z.B. plattformunabhängig aus einem Webbrowser
- ✓ API
- ✓ ...

## Trends

- Integration externer (gekaufter) Daten
- neue Datentypen (Multimedia, Webseiten, Zeit, Raum, ...)

---

# Beispiele von DD-Systemen

---

- Prism Warehouse Directory von Prism Solutions
  - Data Warehouse-Lösung
  - Austausch mit anderen DD-Systemen
  - Webzugriff
  - *Prism Warehouse Manager* automatisiert Ladeprozesse
  - *Prism Change Manager* automatisiert DW-Wartung
- Platinum Repository von Platinum Technology
  - unternehmensweite Lösung
  - *Data Shopper* für den GUI-Zugriff

*Synchrony* ist . . .

ein einfaches **ROLAP** - Frontend ...

Relationales OLAP-Frontend zu einem ODBC-RDBMS  
wie *MS Access*

für **Abfragen** ...

ad hoc statt vordefiniert

benutzerfreundliche Direktmanipulation statt SQL

explorativ statt off line (v.a. Drilling Down and Up →)

auf einem **Sternschema**

*if...Synchrony* setzt ein nach dem Sternschema  
organisiertes relationales Data Warehouse voraus

## Lernziele

⇒ Sternschema

⇒ Abfragen (hier Reports), insbesondere Drilling Down and Up

⇒ Aggregation

Grocery.mdb (MS Access Database)  
(*Synchrony* selbst starten Sie vom Desktop oder aus Explorer)



---

# Architektur von *Synchrony*

---

## Datenbank

⇒ relational

Bsp. Grocery.mdb

## Metadaten

⇒ Sternschema oder Schneeflockenschema

Bsp. Metadaten zu Grocery.mdb

## Sicherheitsdefinitionen

Bsp. ...\\Synchrony\\ifstar.mdw

## Abfragen

⇒ benutzerdefinierte Abfragen

a) vordefinierte Abfragen der Metadatenbank

Bsp. Metadaten zu Grocery.mdb

b) exportierte Abfragen

Bsp. ...\\Synchrony\\Reports\\\*.qry

---

# Wichtige Metadaten von *Synchrony*

---

## ifDatamart

Namen der Data Marts

## ifFamily

Name und Version des Data Mart-Sternschemas, Name und Beschreibung der Fakten und Dimensionen, ...

## ifMeasure

Name, Beschreibung und Schlüssel der Fakttabelle, ...

## ifDimension

Name, Beschreibung und Schlüssel der Dimensionstabellen, ...

## ifAttribute

Name, Beschreibung und Datentyp (Datenformat) der Fakt- und Dimensionsattribute, Formeln der berechneten Attribute, ...

## ifReport

Definition und Beschreibung der Abfragen, ...

## ifAggregate

Struktur einer Aggregatsdefinition

...

[Metadaten zu Grocery.mdb](#)

---

# Funktionalität von *Synchrony*

---

## Eingabe

Relationale Daten in einem Sternschema darstellen

ODBC-Verbindung zu einem RDBMS erstellen

Benutzerfreundliche ROLAP-Abfragen definieren

## Verarbeitung

ROLAP-Abfragen in SQL übersetzen

SQL-Abfragen über ODBC an ein relationales System senden

## Ausgabe

ODBC-Ergebnisse in Berichtsform präsentieren

Eine Einzelhandelskette besteht aus Hunderten von Filialen und verkauft Tausende von Produkten. Die relationale MS Access-Datenbank [Grocery.mdb](#) enthält die Objektdaten. Sie stellen einen *Ausschnitt* der Verkaufsdaten ausgewählter Filialen und Produkte für zwei Jahre dar. Die folgende Dimensionen strukturieren diese Daten :

- Perioden (Tabelle TIME)
- Produkte (Tabelle PRODUCT)
- Filialen (Tabelle STORE)
- Aktionen (Tabelle .PROMOTION).

Das Sternschema ist definiert in [Sample Meta.mdb](#). Die Sicherheitsdefinitionen befinden sich unter `...\Synchrony\ifstar.mdw`.

### ① ***Synchrony* kennen lernen**

#### *Benutzeroberfläche*

a) Starten Sie *Synchrony* vom Desktop oder Explorer aus.

Eine Einführung erhalten Sie unter *Help/Books Online* und Hilfe unter *Help/Synchrony Help*.

b) Wählen Sie den Menüpunkt *File/New* und klicken Sie doppelt auf das Sternschema “Grocery”. Achten Sie darauf, dass die Sternschema-Sicht eingeschaltet ist (*View/Star*).

#### *Sternschema*

c) Wozu dienen die obere und die untere Hälfte des Bildschirms? Welche Synonyme verwendet der Buchtext für die *Synchrony*-Termini “Family”, “Report”, “Measures” und “Fields”?

d) Machen Sie sich mit dem Sternschema “Grocery” vertraut : Wo sind die Fakten und Dimensionen? Welches sind Schlüsselattribute? Erklären Sie die Bedeutung des Symbols, das *Synchrony* vor Faktattribute stellt.

## Einfache Abfragen

- e) Formulieren Sie die Abfrage “Welchen Geldumsatz erzielt jedes Produkt?” und beschreiben Sie das Ergebnis.

(Ziehen Sie die folgenden Attribute (engl. fields) in das Report-Fenster: “Full Description” der Dimensionstabelle PRODUCT und “Sum of Dollar Sales” der Faktentabelle SALES. Klicken Sie dann das Symbol ► (Run Report))

## Filter

Sie löschen eine Ergebnisspalte, indem Sie sie markieren und die Taste Del drücken.

- f) Die Groups- und Studies-Tabs der Dimensionstabellen enthalten Filter. Beantworten Sie die folgende Filterabfrage: “Welchen Geldumsatz erzielten die Produkte der Marke Squeezable Inc.?”

(Ziehen Sie den Filter “Brand = Squeezable Inc.” in das Report-Fenster und klicken Sie auf ►).

- g) Wie hoch war der Geldumsatz der Marke “Squeezable Inc.” 1995?
- h) Wechseln Sie in der Dimensionstabelle PRODUCT in die Browse-Ansicht (Rechtsklick) und inspizieren Sie die Werte einzelner Attribute (Doppelklick auf ein Attribut).

Lassen Sie sich die ganze Tabelle PRODUCT anzeigen (*View/Table*). Finden Sie heraus, wie sich Filter definieren lassen.

- i) Erklären Sie den SQL-Code, der für die Abfrage “Wie gross ist der Gesamtumsatz?” generiert wird (*View/SQL*).
- j) Interpretieren Sie den SQL-Code einer komplexeren Abfrage.

## Aggregate

- k) Aggregatfunktionen fassen alle Werte eines bestimmten Fakts, zum Beispiel “Dollar Sales” zusammen. Finden Sie die Aggregat-

funktionen von Synchrony heraus. Weshalb werden gerade *diese* Aggregatfunktionen angeboten?

- l) Führen Sie die folgende Abfrage aus: “Wieviele Einheiten der Marke “American Corn” wurden nach der Werbeaktion *POS Grabbers Paper* im Vergleich zur Aktion *Shelf Talkers* durchschnittlich verkauft?”. Gehen Sie dabei wie folgt vor:
- Erstellen Sie eine erste Abfrage, indem Sie das Field “Description” aus PRODUCT in das Report-Fenster ziehen, Machen Sie dasselb mit dem Faktattribut “Unit Sales”. Schränken Sie dann “Unit Sales” durch den Filter “Brand = American Corn” aus PRODUCT ein. Damit Sie die Abfrage weiter verfeinern können, klicken Sie auf ►.
  - Nach einem Rechtsklick auf die Faktspalte wählen Sie *Properties/Aggregate* und dann die Aggregatfunktion *Average*.
  - Erstellen Sie eine weitere identische Faktspalte.
  - Wählen Sie den Filter *Promotion Name = POS Grabbers* für die erste Faktspalte und *Promotion Name = Shelf Talkers* für die zweite Faktspalte. Beide Filter finden Sie unter dem Groups-Tab der Dimensionstabelle PROMOTION.
  - Führen Sie die Abfrage aus.

## ② **Synchrony selbständig anwenden**

- m) Vergleichen Sie die Geldumsätze der Filialen und ihre prozentuale Änderungen von 1994 bis 1995. Umsatzrückgänge sollen rot gekennzeichnet werden.
- n) 1994 wurde in allen Filialen eine Diät-Produktreihe eingeführt. Untersuchen Sie die Umsatzentwicklung von 1994 und 1995 und empfehlen Sie, in welchen Filialen die Diätprodukte wieder aus dem Sortiment genommen werden sollen.
- o) Welche Produkt-Untergruppe verkauft sich besonders gut. Wie hoch ist die Gewinnspanne bei den einzelnen Untergruppen?

- p) Vergleichen Sie die Zahlen für die Mittwochskunden mit jenen der Samstagskunden. Welche Produkte werden vor allem an Samstagen nachgefragt?
- q) Erstellen Sie eine der erstellten Abfragen mit QBE. Öffnen Sie dazu die MS Access-Datenbank [Grocery.mdb](#).




### **Möglichkeiten und Grenzen erkennen**

- r) Worin liegen die Vor- und Nachteile eines ROLAP-Werkzeugs wie *Synchrony*?
- s) Vergleichen Sie Synchrony mit *MS Access*, *MS Excel* und *Cognos PowerPlay*.

---

# Inhalt

---

Grundlagen	<a href="#"><u>2</u></a>
Entwicklung - Überblick	<a href="#"><u>3</u></a>
Data Marts und Enterprise Data Warehouse	<a href="#"><u>6</u></a>
Mehrdimensionale Daten	<a href="#"><u>21</u></a>
Endbenutzerzugriff	<a href="#"><u>34</u></a>
OLAP	<a href="#"><u>44</u></a>
ANLAGEBERATUNG mit <i>PowerPlay</i> 	<a href="#"><u>71</u></a>
Modellierung relationaler Data Warehouses	<a href="#"><u>84</u></a>
Modellierung von Informationssystemen	<a href="#"><u>85</u></a>
Sternschema ( LIEFERFRIST )	<a href="#"><u>97</u></a>
EDW-Modell und Sternschema ( VERKAUF )	<a href="#"><u>106</u></a>
<i>Synchrony</i> ( EINZELHANDEL  )	<a href="#"><u>126</u></a>
Entwicklung und Betrieb	<a href="#"><u>136</u></a>
Fallbeispiel ( ZEITSCHRIFTEN  )	<a href="#"><u>151</u></a>
Effizienz	<a href="#"><u>160</u></a>
Netzzugriff	<a href="#"><u>198</u></a>
Client/Server-Architekturen	<a href="#"><u>203</u></a>
Zugriff über Internet und Intranet	<a href="#"><u>213</u></a>



---

# Von Produktionsdatenbanken zum EDW

---

Produktionsdatenbank mit  
Drittgenerations-Sprachen  
wie COBOL oder C

Produktionsdatenbank mit  
Viertgenerations-Sprachen  
wie SQL und QBE

Endbenutzersichten (Views) auf  
eine Produktionsdatenbank mit  
SQL/QBE und Berichtsgeneratoren

Views auf aufbereiteten Daten

Enterprise Data Warehouse  
und Data Marts mit OLAP

Entwicklung von Data Warehouses

# Data Warehouse-Entwicklungszyklus

## ① Spezifikation (Anforderungsdefinition)

Was ist das Ziel?

Ist-Analyse

Soll-Analyse

## ② Realisierung

Wie soll das Ziel erreicht werden?

### Entwurf

Logische Datenmodellierung

Physische Datenmodellierung

Dialogentwurf

Modularisierung

### Implementation

Deklarative Implementation (Daten, Abfragen)

Prozedurale Implementation (Prozeduren)

Modul- und Systemtest

### Betrieb

Installation

Wartung

---

# Phasenunabhängige Tätigkeiten

---

## Planung und Kontrolle der ...

- ✓ Zwischen- und Endprodukte
- ✓ Ressourcen
  - Zeit
  - Geld
  - Personal
  - Hardware/Software

## Kontrollmethoden

- ✓ Spezifikations-Walk Through
- ✓ Entwurfs-Walk Through
- ✓ Implementations-Walk Through

## 5.54 Data Warehouse-Ziele

<i>Kriterium</i>	<i>Beispiele</i>
<b>Abbildungstreue</b>	einfache und funktionale Abbildung betrieblicher Fakten und Dimensionen
<b>Benutzerfreundlichkeit</b>	Mehrdimensionalität (Würfel oder Sternschema), mit Drilling, Slicing and Dicing, Pivoting, Visualisierung und Views
<b>Flexibilität</b>	Ausbaubarkeit von Clients, Servern und Netz, vordefinierte <i>und</i> ad hoc-Abfragen
<b>Performance</b>	Durchsatz, Laufzeiteffizienz, Speichereffizienz
<b>Verfügbarkeit</b>	Multiprozessoren, RAID, Festplattenspiegelung, Stand by-Server, On line-Verwaltung ohne Shut Down von Systemkomponenten
<b>Wartbarkeit</b>	Synchronisation mit OLTP, Metadatenverwaltung
<b>Kompatibilität</b>	Data Mining Tools, Produktionsdatenbanken, Anwendungssoftware

# Endbenutzerbefragung - Ablauf

---

Befragungsteam zusammenstellen

Vorbereitungsmaterial sammeln

Jahresbericht

Planungshandbuch

Marketingunterlagen

Frühere “Data Warehousing”-Initiativen

Adressaten wählen

Linie

Stab (v.a. “Daten”kundige)

Fragebogen zusammenstellen →

Ablauf planen

Befragung ankündigen

Rollenverteilung und Interviewtechniken reflektieren

Fragende

Antwortende

Protokollführer

Beobachter

Befragung durchführen →

Endbenutzer-Vokabular verwenden

Gemeinsame Terminologie verwenden

Interview flexibel führen

---

# Endbenutzerbefragung - Inhalte

---

## Einführung

Projektziele  
Interviewziele  
Interviewer  
Zeitraumen  
Folgeschritte  
Pflichtenhefte

## Ziele

Geschäftsziele  
Produkte  
Erfolgsindikatoren  
Geschäftsrisiken  
Früherkennung  
Innovationshäufigkeit

## Analyseanforderungen (Ist/Soll)

Routineanalyse  
Ad hoc-Analysen  
Historische Auswertungen  
Berichte  
Quellenzugang

## Schluss

Zusammenfassung  
Bitte um Zusatzmaterial  
Projektchancen

## Feedback

Feinspezifikation

# Spezifikationsdokument

---

Zusammenfassung für den Manager

**Projektziel** (Grobspezifikation)

**Anforderungen**

Betriebswirtschaftliche Anforderungen

Informationstechnologische Anforderungen

**Datenquellen**

---

# Entwicklungsprobleme

---

*You Can't Eat an Elephant in One Byte !*

## Anforderungsdefinition

Koordination von Fachabteilungen

## Laden operativer Daten

Datenextraktion, -transformation und -integration

## Komplexität und Redundanz der Datenmodelle

Definition und Wartung komplexer Tabellenstrukturen

Einführung von Redundanz

## Speicher- und Zugriffseffizienz

bei mehreren Fakttabellen und grossen Dimensionstabellen

bei umfangreichen Indizes und Vorberechnungen

## Datenverantwortung und -sicherheit

Datenhoheit der Fachabteilungen

## Evaluation und Schulung

Tabellenmodell (z.B. Fremdschlüssel-Begriff)

Boolsche Abfragelogik für Endbenutzer

## Projektmanagement

lange Projektdauer (Jahre statt Monate)

mangelnde Erfahrung



# Rollen der DW-Entwicklung und -Verwaltung

<b>Entwicklung eines Data Warehouse</b>	
<i>Rolle</i>	<i>Aufgabe</i>
<i>Data Warehouse Administration</i>	Entwicklung und Verwaltung in allen Phasen leiten
<i>Business Requirements Analysis</i>	Anforderungen der Endbenutzer spezifizieren und Entwurf begleiten
<i>Data Warehouse Architecture</i>	Data Warehouse-Architektur entwerfen, insbesondere Daten modellieren
<i>Data Acquisition Development</i>	Daten extrahieren, transformieren und integrieren
<i>Data Access Development</i>	Freien Zugriff für Experten und vordefinierte Pfade für gelegentliche Benutzer definieren
<i>Programming and Maintenance</i>	Programmierung, Überwachung und Sicherheitsverwaltung
<b>Wartung eines Data Warehouse</b>	
<i>Data Warehouse Administration</i>	Entwicklung und Verwaltung in allen Phasen leiten
<i>Organisational Change Management</i>	Betriebliche Auswirkungen von Data Warehouse-Entwicklungen begleiten
<i>Database Administration</i>	Physische Daten verwalten
<i>Metadata Management</i>	Metadaten verwalten

---

## Rollen der DW-Entwicklung (A 5.8)

---

Während der Entwicklung eines Data Warehouse üben Mitarbeiter die unterschiedlichsten Rollen aus. Ordnen Sie den folgenden Rollen passende Tätigkeiten und Kompetenzen zu.

- a) Projektdirektor
- b) Projektmanager
- c) Analytiker
- d) Data Warehouse-Architekt
- e) Datenbankadministrator
- f) Systemadministrator
- g) Migrationsspezialist
- h) Data Mart-Entwickler
- i) Qualitätskontrolleur
- j) Legacy-System-Spezialist
- k) Ausbildner
- l) Handbuch-Autor
- m) Kommunikationsspezialist
- n) Sponsor
- o) Help Desk-Verantwortlicher
- p) Werkzeugspezialist
- q) Web Master
- r) externer Berater

# Software für Data Warehouses

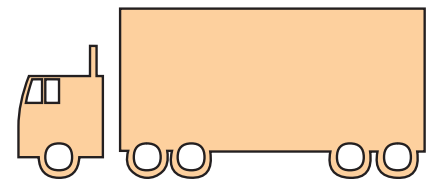
## Produktionsdaten

Relationale DBMS  
(z.B. von *Oracle*)  
Nichtrelationale DBMS  
Einzeldateien



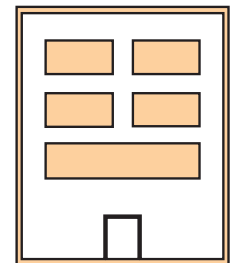
## Laden

DBMS-Funktionen wie Replikation und Snapshot  
Dedizierte Standardsoftware  
(z.B. *ETI-EXTRACT* Suite von Evolutionary Technologies,  
*Prism Warehouse Manager* von Prism Solutions)  
Selbst erstellte Software  
(z.B. gespeicherte Prozeduren, SQL, 3GL)



## Data Warehouse - Verwaltung

Spezialisierte relationale DBMS  
(z.B. *Sybase IQ*, *Red Brick Data Warehouse*)  
Multidimensionale Datenbanksysteme  
(z.B. *Arbor Essbase*)  
Werkzeuge zur Metadatenverwaltung  
(z.B. *Sybase PowerDesigner Warehouse Architect*)  
Werkzeuge zur Performanceoptimierung  
(z.B. Analyse der Abfrageprofile durch ein OLAP-Werkzeug)  
Datensicherung und -archivierung  
(z.B. Systemsoftware)



...



## Data Warehouse - Zugriff

Abfragewerkzeuge und Berichtsgeneratoren  
(z.B. OLAP-Werkzeuge wie *Cognos Powerplay*)  
Internetwerkzeuge  
(z.B. *Sybase PowerDynamo* oder *MS InterDev*)  
Data Mining-Werkzeuge  
(z.B. *Cognos Impromptu*)  
Allgemeine 4GL-Entwicklungsumgebungen  
(z.B. *MS Visual Basic* oder *PowerBuilder*)



...

## IBM Information Warehouse

### Hardware und Betriebssysteme

OS/2

AIX

MVS

### RDBMS

DB2

### Data Warehouse Management

Metadatenverwaltung

Datenverteilung

Zugriffswerkzeuge

Data Mining-Werkzeuge

# Implementation und Betrieb

Spezifikation und Entwurf

Daten aus operativen Systemen **kopieren**

Gekaufte und neu erhobene Daten **hinzufügen**

Daten **transformieren**

Daten im zentralen Data Warehouse **speichern**

Daten auf allfällige Data **Marts** replizieren

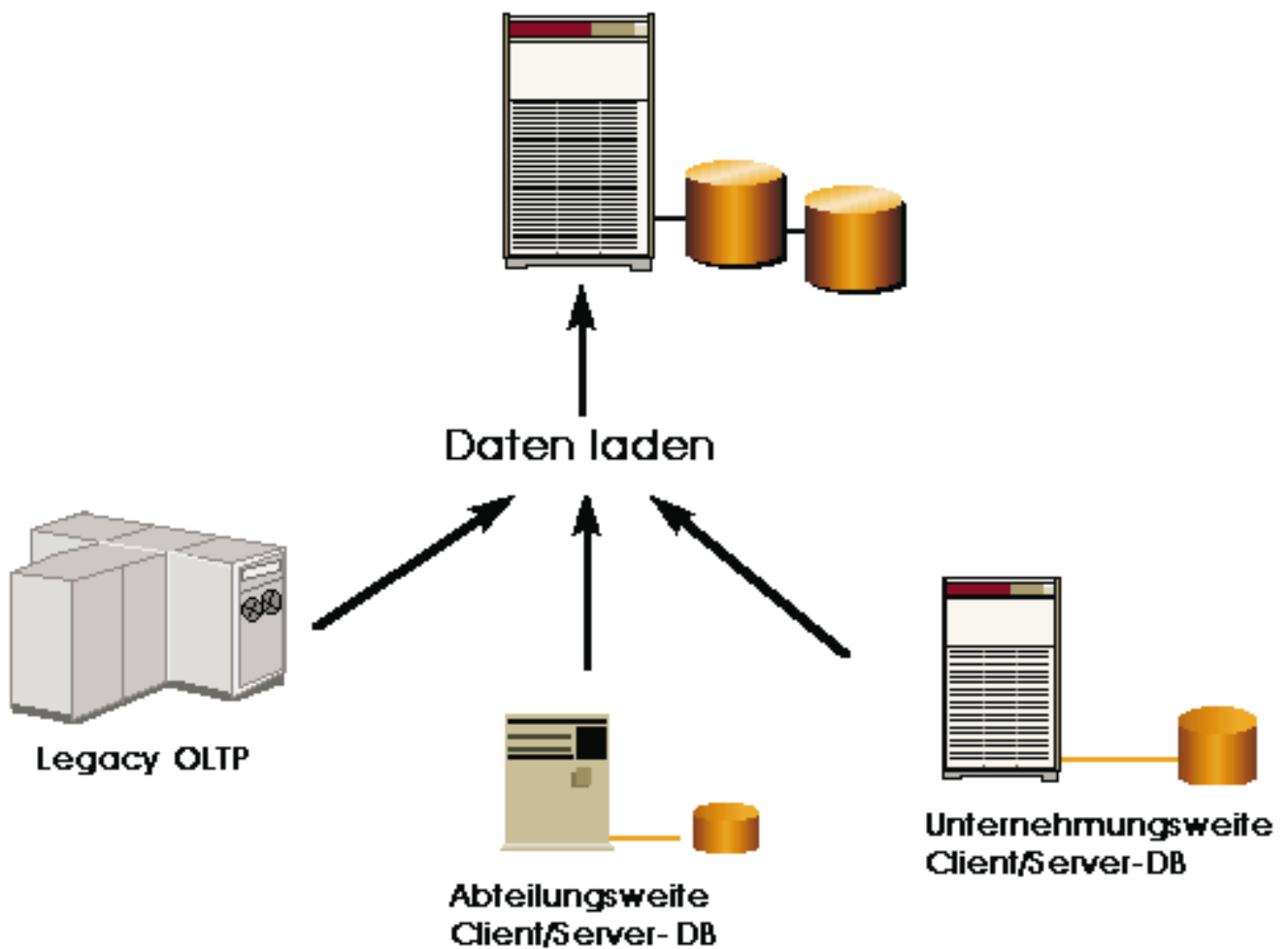
Daten **abfragen**

**Data Mining** - Methoden und -Software anwenden

Ergebnisse **interpretieren**

Illustration am Beispiel ZEITSCHRIFTEN

## 5.58 Operative Daten laden



### Daten laden

- extrahieren  
transformiert
- integrieren

*Legacy OLTP: On Line Transaction Processing durch Grossrechner*

## Verlag

Zeitschriften “Auto”, “Wohnen”, “Sport”, “Musik” und “Comics”

## Marktforschungsfragen

- Profil der Leser von “Wohnen”?
- Zusammenhänge zwischen “Sport”- und “Auto”-Lesern?
- ...

## Mögliche Ergebnisse

- Hypothekarverschuldete Kunden sind mit einer Wahrscheinlichkeit von 0.8 “Wohnen”-Abonnementen
- “Sport”-Abonnent zwischen 20 und 30 subskribiert mit einer Wahrscheinlichkeit von 0.4 zwischen 30 und 40 auch “Auto”
- ...

Phasen des Data Warehousing und Data Mining -

## B Daten auswählen

### Ausgewählte Produktionsdaten **kopieren**

⇒ Ausgewählte Attribute der Produktionsdatenbank in die Tabelle SUBSKRIPTIONEN kopieren:

<i>Kundennr.</i>	<i>Name</i>	<i>Adresse</i>	<i>Subskription</i>	<i>Zeitschrift</i>
14005	Müller Dani	Hofstrasse 1	20.3.1996	“Auto”
18004	Meier Fritz	Stadtstrasse 5	28.2.1901	“Wohnen”
18004	Ott Hugo	Bergweg 3	6.5.1996	“Musik”
...	...	...	...	...
14005	Miller Dani	Hofstrasse 1	4.5.1997	“Sport”

### Gekaufte und getrennt erhobene Daten **hinzufügen**

⇒ Marktforschungsdaten kaufen

*Geburtsdatum*

*steuerbares Einkommen*

*Privatkreditvolumen*

*Autoeigentum*

...

⇒ Zusätzliche Daten per Telefoninterview erheben

*Wohneigentum*

...



## C Daten transformieren

Daten mit SQL transformieren

### a) Duplikate und irrelevante Attribute entfernen

- ⇒ Duplikate "Müller Dani ..." und "Miller Dani ..." entfernen  
(kleine Namensunterschiede in sonst identischen Tabellenzeilen)
- ⇒ Irrelevantes Attribut *Name* weglassen

### b) Nicht plausible Werte berichtigen oder weglassen

- ⇒ Zeile mit dem unplausiblen Subskriptionsjahr "1901" weglassen

<i>Nr</i>	<i>Adresse</i>	<i>Subskriptionsdatum</i>	<i>Zeitschrift</i>
18004	Stadtstrasse 5	28.2.1901	"Wohnen"

### c) Attribute redefinieren

- ⇒ Zu detaillierte Attribute redefinieren

Detailliertes Attribut	Redefiniertes Attribut
Geburtsdatum	<i>Altersklasse</i>
Adresse	<i>Wohnregion</i>
Subskriptionsdatum	<i>fortlaufende Monatszahl seit 1990</i>
Einkommen	<i>Einkommensklasse</i>

## D Daten transformieren und analysieren

### d) Zeilen **aggregieren** und **Formate vereinheitlichen**

⇒ Zeilen mit gleicher Kundennummer gruppieren

⇒ Binäre Attribute mit den Werten 1 und 0 codieren

ja / nein und abonniert / nicht abonniert werden zu 1 / 0

<i>Kundennummer</i>	<i>Altersklasse</i>	<i>Einkommensklasse</i>	<i>Kreditklasse</i>	<i>Autoeigentümer</i>	<i>Hauseigentümer</i>	<i>Region</i>	<i>"Auto"</i>	<i>"Wohnen"</i>	<i>"Sport"</i>	<i>"Musik"</i>	<i>"Comics"</i>
18004	20	39	10	0	1	1	0	0	1	1	0
...	...	...	...	...	...	...	...	...	...	...	...

Transformierte Daten in das Warehouse **integrieren**

Data **Marts** "beliefern"

Daten **konventionell** auswerten

⑦ Data Mining (Teil II des Fallbeispiels *Zeitschriften*)

Ergebnisse **interpretieren**

---

# Funktionalität von Ladeoperationen

---

## Allgemeine Anforderungen

Auf ein zentrales Data Dictionary zugreifen

Stapelweise und inkrementell laden

Auf Verlangen und automatisch laden

## Extraktion

Aus verschiedenen Quellen laden, vor allem aus ...

- flachen und indizierten Dateien (v.a. VSAM)
- relationalen und anderen Datenbanken (v.a. IMS)

## Transformation

Validität prüfen (Vollständigkeit und Konsistenz)

- Syntaktische Fehler erkennen
- Semantische Fehler erkennen

Datentypen konvertieren, zum Beispiel ...

- vom EBCDIC- zum ASCII-Code
- vom julianischen zum gregorianischen Datum
- von einer Nullwertcodierung zu einer anderen

Attribute redefinieren

- z.B. Einzelwerte in Klassen zusammenfassen

Attribute neu einfügen

- z.B. Zeitdimension

Werte vorberechnen

- z.B. Tagesumsätze in Wochenumsätze aggregieren
- z.B. statistische Kennzahlen berechnen

## Integration

Geladene Daten ...

- direkt in Warehouse-Tabellen einfügen
- in einer sogenannten Staging Area zwischenspeichern

Datenmodell vorbereiten

- mehrdimensionaler Würfel
- relationales Sternschema

Geprüfte und transformierte Daten integrieren

Indizes neu aufbauen oder fortschreiben

Metadaten nachführen

# Beispiel Validitätsprüfungen

## Syntaktische Prüfungen

### Namenskonventionen

*Punkt nach Abkürzungen  
ausgeschriebener Vorname*

### Fehlende Werte

*Andrea Braun ohne Geschlechtswert*

## Semantische Prüfungen

### Wertebereiche

*Monatszahl 13*

### Homonyme

*L.R. Brown*

*St. Peter's Church*

*1st St.*

*St. Petersburg*

### Synonyme

*PC                                      Gestell 237*

*Personalcomputer              237*

*Microcomputer                No. 237*

### Inkonsistenzen

*1997er-Transaktion mit einem erst 1998 eingeführten Produkt*

# Ladesoftware

---

## Entwicklung

- **Selbstentwickelte Software**  
zum Beispiel mit SQL, gespeicherten Prozeduren, COBOL, MVS JCL, Unix Scripts
- **Standardsoftware** →

## Spezialisierung

- **Allgemeine DBMS-Werkzeuge**  
zum Beispiel zur Replikation  
Problem: meist beschränkt auf RDBMS-Quellen
- **Dedizierte Ladewerkzeuge** →

## Technologie

- **Generatoren**  
Problem: Verwaltung der generierten Programme
- **Selbständige Systeme**

## Quellen

- flache Dateien
- indizierte Dateien
- hierarchische DBMS
- Netzwerk-DBMS
- RDBMS

---

# Einige Ladeprodukte

---

## Prism Warehouse Manager von *Prism Solutions*

- Auswahl, Transformation und Integration
- Generator

## PASSPORT von *Carleton*

- Auswahl, Transformation und Integration
- COBOL-Generierung
- dreistufige Architektur

## Integrity von *Vality*

- Ergänzung von Extraktionswerkzeugen mit Integritätsprüfung

## EDA Suite von *Information Builders*

- Auswahl, Transformation und Integration
- SQL-Generierung

## ETI-EXTRACT Tool Suite von *Evolutionary Technologies*

- Auswahl, Transformation und Integration
- Generator

## PowerMart Suite von *Informatica*

- Auswahl, Transformation und Integration
- Data Mart Builder
- inkrementelles Laden

## ① Logische Modellierung

abbildungstreues und  
benutzerfreundliches Datenmodell !

- Denormalisierung
- Sternschema



## ③ ▶ Physische Modellierung

effizientes *physisches* Schema !

- Tabellen definieren
- Schlüssel physisch definieren
- Integritätsprüfungen physisch definieren
- ...
- Performance optimieren →



# Performanceprobleme

Steigende **Verarbeitungszeiten**

Laden

Antworten

Backup

Recovery

...

+

Sinkende **Verfügbarkeit**

Shutdowns durch Laden

Shutdowns durch Backup

...

*Laden, Abfragen, Aggregieren*

---

# Performanceproblem Aggregation

---

Berechnungen brauchen Zeit



Aggregationsdaten belegen bis zu 2/3 eines DW



Vordefinierte oder Ad hoc-Aggregation?

- Betriebliche Bedeutung
- Häufigkeit
- Prozessor- und Speicherbelastung

---

# Performanceproblem Sternschema

---

Mehrdimensionalität →

Dimensionenzahl und Kardinalitäten

Grösse der Faktentabelle

Faktenschlüssel

Dimensionskategorien im Faktenschlüssel  
(*Jahr, Monat, Woche, Tag*)

künstlicher Faktenschlüssel

Verbund von Dimensions- und Faktentabellen

*Siebentabellenverbund erfordert  $7! = 5040$  Paarverbunde*

Faktentabelle an den Anfang eines Mehrtabellenverbunds

## 5.59 Performanceproblem Mehrdimensionalität I

### Kardinalitäten der Dimensionen

<i>Dimensionen</i>	<i>Kategorien</i>	<i>Kardinalität</i>
Filialen	Zahl	20
Produkte	Zahl	50
Periode	Jahr / Monat / Tag	10 / 12 / 30

### Zeilen des Fakts *Jahresumsatz* nach Dimensionen

<i>Berechnung</i>	<i>Zeilen der Fakttabelle</i>
10 Jahre	<b>10</b>
10 Jahre, 20 Filialen	$10 \cdot 20 = \mathbf{200}$
10 Jahre, 20 Filialen, 50 Produkte	$200 \cdot 50 = \mathbf{10'000}$
120 Monate, 20 Filialen, 50 Produkte	$10'000 \cdot 12 = \mathbf{120'000}$
3'600 Tage, 20 Filialen, 50 Produkte	$120'000 \cdot 30 = \mathbf{3'600'000}$

Speicher-, Zugriffs- und Rechenaufwand optimieren

# Performanceproblem Mehrdimensionalität II

Umsatz **total**



Umsatz nach 5 **Produkten**

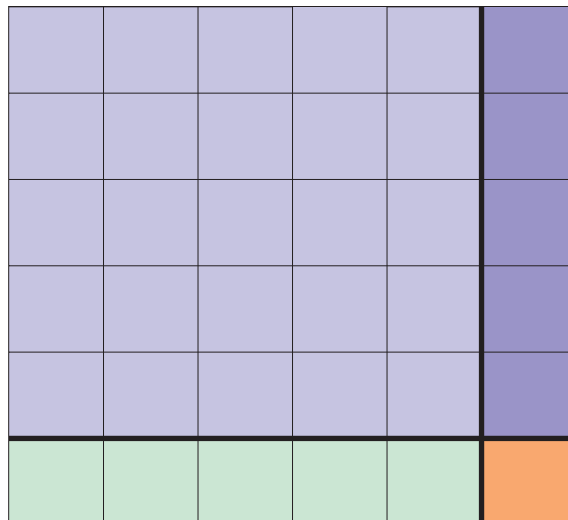
(1 Dimension und 1 Aggregat)



Umsatz nach 5 **Produkten**

und **Regionen**

(2 Dimensionen und 11 Aggregate)



kombinatorische Explosion

Speicheraufwand optimieren

# Data Warehouse-Performance

## Data Warehouse-Ziele

Abbildungstreue

Benutzerfreundlichkeit

Flexibilität

Verfügbarkeit

Kompatibilität

## Performance

⇒ Speicherkapazität und -effizienz

⇒ Laufzeiteffizienz, insb. Antwortzeit

⇒ Zahl der Benutzer

## Hardwaredeterminanten

Prozessor

Externspeicher

Internspeicher

Kommunikation

## Datenmodell-Determinanten

Physisches Datenmodell

Logisches Datenmodell

## Verbesserung?

---

# Datenmodell-Performance verbessern

---

## Logisches Datenbankmodell

**Minimierung** der Objekte und Beziehungen

**Modularisierung** nach Data Marts

**Sternschema und Denormalisierung** →

- Vorberechnung

- Spaltenreplikation (erspart Verbundoperationen)

- Vorverbund

## Physisches Datenbankmodell

**Indexierung**

- B-Baum

- Bitmuster-Index →

- Hashfunktion

- Künstliche Faktenschlüssel

**Speicherung**

- Partitionierung →

- Nullwerte

- Kompression

**Vordefinition und Überwachung von Abfragen**

- Ablaufplan

- Off line-Kandidaten aussondern

- Überwachung mit Warteschlangen und Query Blockers

**Parallelisierung** →

# Optimierungsmassnahme Denormalisierung

## Vorbereitung

*Bsp. Verkäufe nach Jahr, Monat und Woche zusammenfassen*

⇒ 1 Detailtabelle, 3 Aggregationstabellen

## Spaltenreplikation

Verbund durch Integration fremder Spalten vermeiden

⇒ redundante Spalten

## Vorverbund

Verbund durch Integration der Verbundtabelle vermeiden

⇒ redundante Tabelle

Kosten und Nutzen der Denormalisierung abwägen  
(Zeitgewinn versus Speicher- und Zeitaufwand)



## 5.61 Normalisierung verlangsamt Abfragen

### VERANSTALTUNG

<u>Name</u>	<u>Stunden</u>
<u>Recht</u>	<u>2</u>
BWL	4
VWL	3

### ORGANISATION

<u>Name</u>	<u>Semester</u>	<u>Dozent</u>	<u>Raumnummer</u>
Recht	SS93	Meier	111
Recht	SS94	Meier	112
<b>BWL</b>	WS94	<b>Schmid</b>	111
<b>BWL</b>	SS95	<b>Schmid</b>	111
...	...	...	...

### RAUM

<u>Raumnummer</u>	<u>Plätze</u>
111	<b>200</b>
112	150

Abfragen auf **einer** und **mehreren** Tabellen

a) Stundenzahl der Veranstaltung Recht ?

b) **Plätze** der Räume für **BWL**-Vorlesungen von **Schmid** ?

## 5.62 Denormalisierung beschleunigt Abfragen

**Plätze** der Räume für **BWL**-Vorlesungen von **Schmid** ?

### ORGANISATION

<u>Name</u>	<u>Semester</u>	<u>Dozent</u>	<u>Raumnummer</u>
Recht	SS87	Meier	111
Recht	SS88	Meier	112
BWL	WS88	Schmid	111
BWL	SS89	Schmid	111
...	...	...	...

+

### RAUM

<u>Raumnummer</u>	<u>Plätze</u>
111	200
112	150

↓

### VERBUND

<u>Name</u>	<u>Semester</u>	<u>Dozent</u>	<u>Raumnummer</u>	<u>Plätze</u>
Recht	SS87	Meier	111	200
Recht	SS88	Meier	112	150
BWL	WS88	Schmid	111	200
BWL	SS89	Schmid	111	200
...	...	...	...	...

---

# Optimierungsmassnahme Partitionierung

---

## Begriff

Tabellenspaltung

## Zweck

- Zugriffszeit optimieren
- Speichereffizienz optimieren
- Partitionsweise Lösch- und Updateoperationen erleichtern

## Arten

- **Horizontale** Partitionierung (z.B. nach dem Transaktionsdatum)
- **Vertikale** Partitionierung (z.B. nach der Normalisierung)

## Partitionsschlüssel

- Reihenfolge der Zeilen bzw. Spalten
- statische und dynamische Daten
- häufig zusammen verwendete Daten
- Grösse von Eingabe-/Ausgabeblocks
- ...

# Partitionierungsbeispiele

## Zeilenpartitionierung für Transaktionsdatenbanken

<i>Name</i>	<i>Abteilung</i>	...
Müller	Personalwesen	...
...	...	...
...	...	...
Studer	Finanzwesen	...
...	...	...

## Spaltenpartitionierung für Data Warehouses

<i>Name</i>	<i>Abteilung</i>	...
Müller	Verkauf	...
Meier	Beschaffung	...
Schmid	Rechnungswesen	...
...	...	...

## 5.63 Spaltenpartitionierung

I/O-Blöcke so speichern, dass ...

zusammen verwendete Daten *hintereinander* stehen

+

**Anwendungen** verwenden in der Regel ...

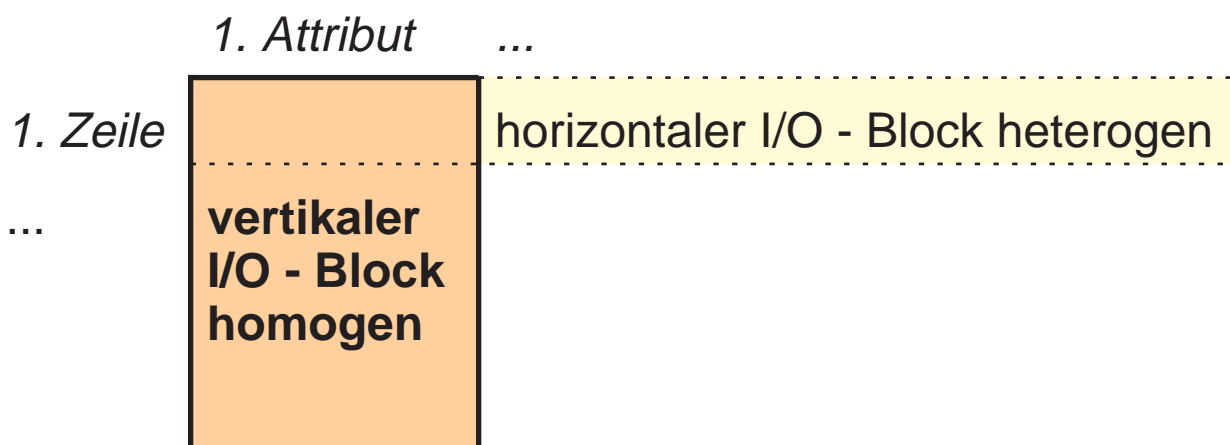
- *operative* Daten satzweise (zeilenweise)
- *analytische* Daten attributweise (spaltenweise)

+

Die **Kompression** *homogener* Daten ist leichter

↓

**Data Warehouse - Daten** so speichern, dass Werte des gleichen *Attributs* im gleichen I/O-Block stehen



---

# Optimierungsmassnahme Indexierung

---

## Nutzen

Zeitgewinn durch Direktzugriff

## Kosten

Speicheraufwand

Erzeugung der Indizes zur Ladezeit

u.U. Zeitverlust gegenüber sequentieller Suche

## 5.64 Optimierungsmassnahme *Bitmuster*

### Datenbasis

<i>Wirtschaftssektor</i>	<i>Region</i>	<i>ISDN-Anschluss?</i>
Dienstleistung	Oberrhein	ja
Landwirtschaft	Mittelland	nein
Dienstleistung	Oberrhein	ja
Dienstleistung	Genfersee	nein

### Abfrage

Wieviele *Dienstleistungs*firmen verfügen über einen *ISDN*-Anschluss und haben ihren Sitz in der Region *Oberrhein*?

<i>Dienstleistung</i>	<i>Oberrhein</i>	<i>ISDN</i>
1 (ja)	1 (ja)	1 (ja)

**Bitmuster-Index** der Datenbasis (Abfrageergebnis eingerahmt)

<i>Dienstleistung?</i>	<i>Oberrhein?</i>	<i>ISDN-Anschluss?</i>
1	1	1
0	0	0
1	1	1
1	0	0

### Abfrageergebnis

2

# Komprimierte Bitmusterindizes

## Mehrwertiges Attribut

z.B. *Wirtschaftssektor* [Dienstleistung, Industrie, Landwirtschaft]

in mehrere **zweiwertige Attribute** transformieren

z.B. in [Dienstleistung, nicht Dienstleistung] und [Industrie, nicht Industrie] und [Landwirtschaft, nicht Landwirtschaft]

## Mehrere Bitmusterindizes

z.B. drei Indizes statt ein Index

## Bitketten der Bitmusterindizes komprimieren

Bsp. Ein *komprimierter Bitmusterindex* für eine halbe Million Sätze kann in einem einzigen Block von 4096 Bytes gelesen werden

Kompression und Effizienz bitweiser  
Operationen erleichtern **Ad hoc** - Abfragen



---

# Wann Bitmuster-Indizes?

---

✓ Daten **statisch**

- read only oder
- Update selten

✓ Daten **geringer Kardinalität**

- wenig Attributwerte pro Spalte  
im Vergleich zur Zeilenzahl  
Bsp. Ländercode

---

# Kosten und Nutzen der Bitmuster-Indexierung

---

## Nutzen

bei kleiner Kardinalität kleiner als B-Baum

kurze Abfragezeit

(kleiner Index und bitweise Operationen)

## Kosten

Speicheraufwand

Erzeugung der Indizes zur Ladezeit

u.U. Zeitverlust gegenüber sequentieller Suche

---

# Performancedeterminanten Hardware

---

## Prozessorleistung

Einzelprozessor-Durchsatz

Mehrprozessor-Durchsatz

## Internspeicherleistung

Kapazität

Zugriffszeit

## Externspeicherleistung

Erforderliche Kapazität =

Rohdatenvolumen

+ Overhead (Indizes, Temporärspeicher)

(3 bis 10 mal das Rohdatenvolumen)

Erforderlicher Durchsatz

## Kommunikation

Systembus

Netz

# Hardwareperformance verbessern

Einprozessor-Systeme

⇒ Komponenten optimieren

Flaschenhals-Problem

Mehrprozessor-Systeme (MPS)

⇒ Divide et impera!

- Scale Up (Verbesserung der Kapazität)
- Speed Up (Verbesserung der Geschwindigkeit)

Programmierbarkeits-Problem

---

# Komponentenperformance verbessern

---

## Prozessor

- Prozessorarchitektur wechseln (z.B. von CISC nach RISC)
- Takt erhöhen (z.B. von 200 auf 300 MHz)
- Bus verbreitern (z.B. von 32 bit zu 64 bit)

## Internspeicher

- Hauptspeicher vergrößern und beschleunigen
- Cache vergrößern und beschleunigen

## Peripherie beschleunigen

- RAID (engl. Redundant Array of Inexpensive Disks)
  - viele kleine Plattenspeicher als Ganzes an den Systembus angeschlossen
  - kleinere Festplatten sicherer
  - Wiederherstellbarkeit einer ausgefallenen Platte wegen redundanter Datenspeicherung

# Optimierungsmassnahme Parallelisierung

**Parallelverarbeitung** :=

gleichzeitige Verarbeitung  
mehrerer Aufgaben  
auf mehreren Prozessoren

**Aufgabe** (engl. task) :=

*Anweisungsteil*

*Anweisung*

*Prozess* aus mehreren Anweisungen

*Job* aus mehreren Prozessen

unabhängiges *Programm*

MPS parallelisieren ...

⇒ **Auswahl**abfragen ( Abfragen i.e.S.)

- durch die Datenbankmaschine
- durch OLAP-Werkzeuge auf Clients und Servern

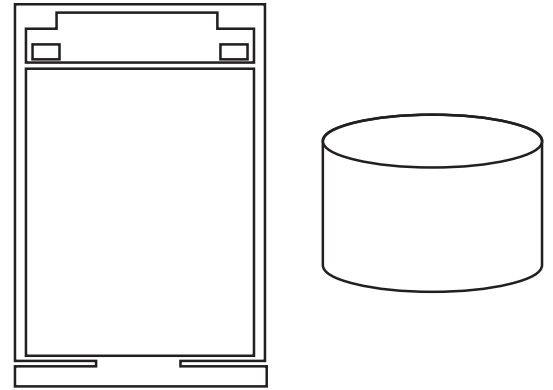
⇒ **Aktions**abfragen

- Extraktion, Transformation und Integration
- Sicherung und Wiederherstellung
- Indexierung

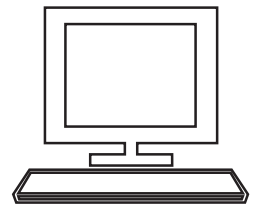
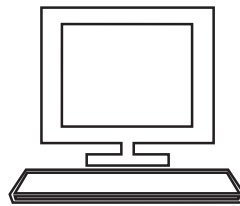
Speed Up und Scale Up  
Verwaltungsaufwand

## 5.65 Beispiel eines Data Warehouse-MPS

SMP-Server  
8 Prozessoren  
8 Gigabyte RAM  
32 Gb Externspeicher  
paralleles DBMS  
(z.B. Oracle)



MS Windows NT  
OLAP-Software  
(z.B. Oracle Express)  
Data Mining-Software  
(z.B. SAS)





---

# Arten von Mehrprozessorsystemen

---

## ① Mehrere Prozessoren **eng** koppeln

SMP-Systeme →

## ② Mehrere Prozessoren **lose** koppeln

MPP-Systeme →

Clusters aus Mehrprozessor-Systemen

## 5.66 Symmetrisches<sup>1</sup> MPS (SMP)

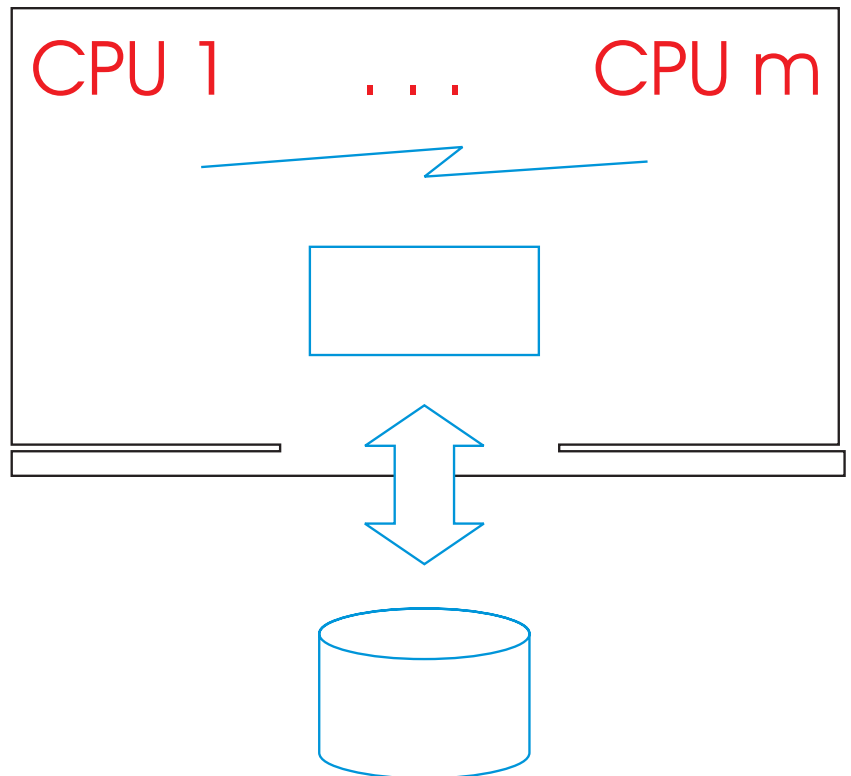
Wenige Prozessoren

Gemeinsamer Bus

Gemeinsamer Interspeicher

Gemeinsame Disk Controller

Gemeinsame Laufwerke



einfach programmierbar

begrenzt skalierbar

Flaschenhals Systembus

<sup>1</sup> symmetrisch, weil Prozessoren gleich mächtig und gleichberechtigt

---

# Eine *mittlere* SMP-Konfiguration (1998)

---

## Prozessoren

*8 Intel Pentium Pro*

## Internspeicher (engl. memory)

*256 Mb bis 8 Gb*

## Bus

*PCI*

## Externspeicher (engl. storage)

- interne Festplatten  
*zusammen 1 bis 2 Tb (Terabyte)*
- externe Festplatten  
*bis mehrere Dutzend Tb*

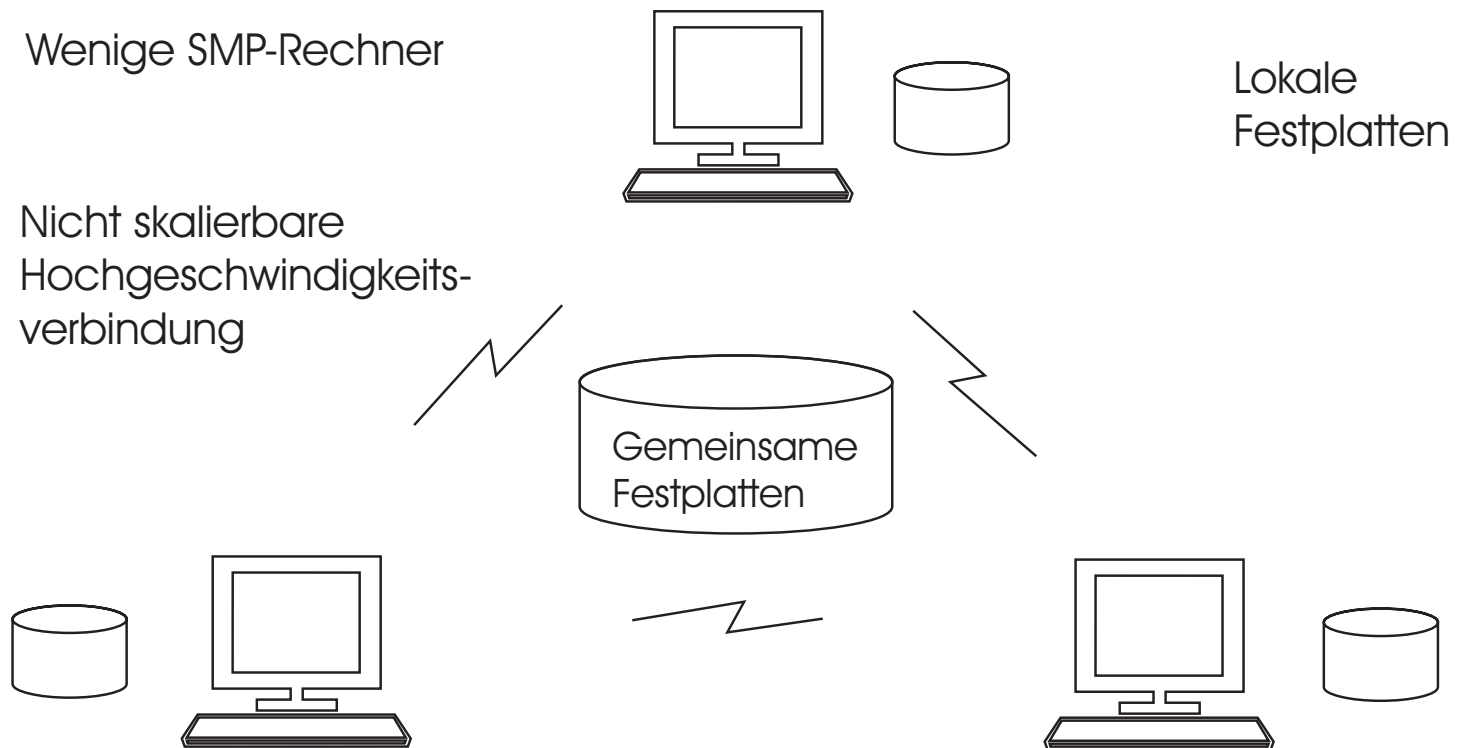
## Betriebssystem

Bsp. MS Windows NT Enterprise Edition

## Datenbanksystem

Bsp. MS SQL Server

## 5.67 Clustering



besser skalierbar

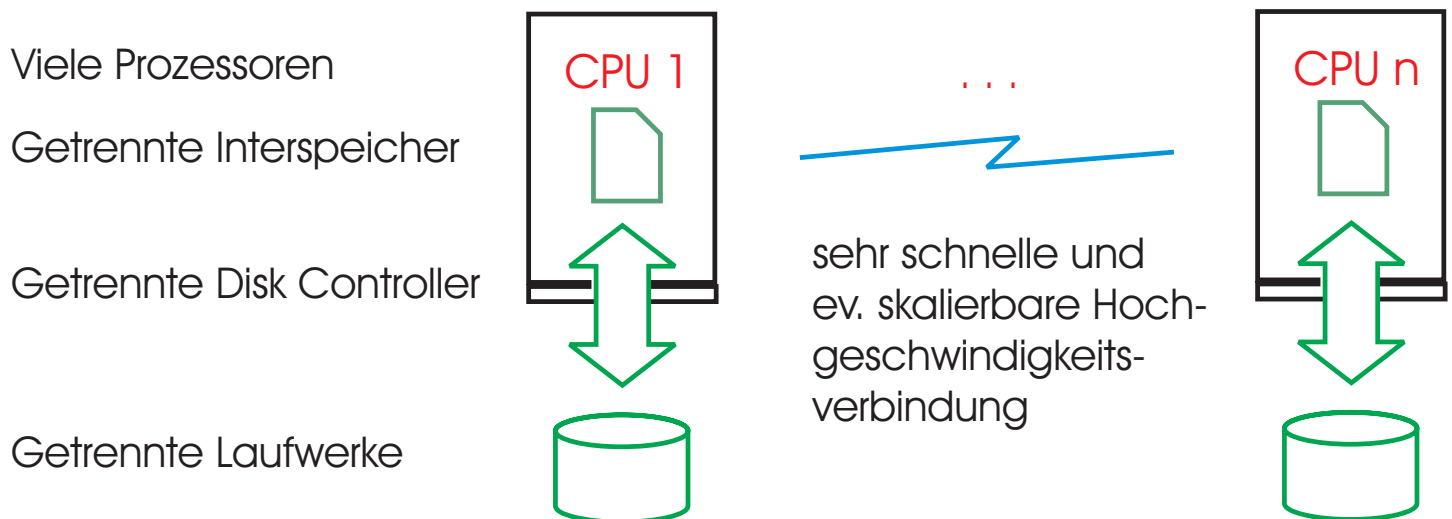
fehlertolerant

schwerer programmierbar

(Zugriff auf andere Knoten)

## 5.68 Massiv paralleles MPS

**MPP** := massiv paralleles Mehrprozessorsystem

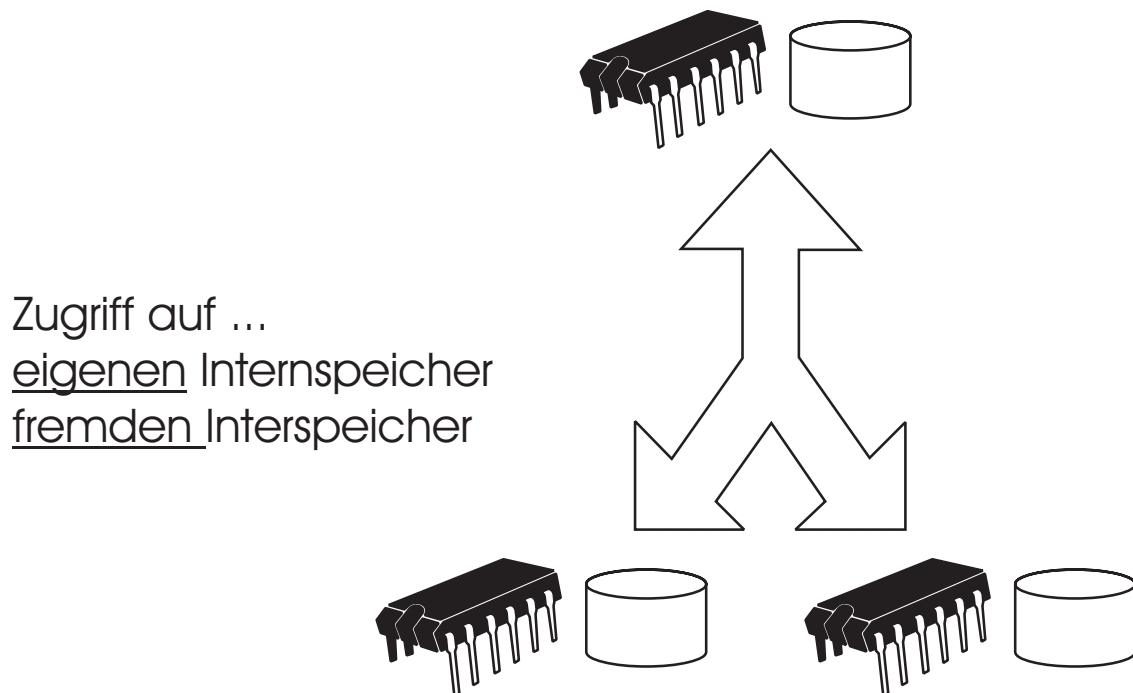


sehr gut skalierbar

schwer programmierbar

- 1 massiv, weil mehr Prozessoren mit grösserer Bandbreite als in einem SMP
- 2 Manche Systeme greifen auch auf gemeinsame Festplatten zu
- 3 Moderne MPS bestehen auch aus SMP-Knoten (wie Clusters)

## 5.70 NUMA (Non Uniform Memory Access)



gut skalierbar

einfach programmierbar

Internspeicherzugriff nicht uniform

Zugriff auf eigenen Internspeicher schnell

Zugriff auf fremden Internspeicher weniger schnell (z.B. 1:2)

## 5.69 *Symmetrische vs. massiv parallele MPS*

	<b><i>Symmetrische MPS</i></b>	<b><i>Parallele MPS</i></b>
<i>Synonyme</i>	Symmetric Multi-processing (SMP), Shared Memory -	Massively Parallel Processing (MPP), Shared Nothing -
<i>Prozessoren (mit Cache)</i>	wenige gleichberechtigte (symmetrische) i.d.R. 4-30	viele (bis Hunderte)
<i>Verbindung</i>	Standard-Systembus	proprietär
<i>Betriebssystem</i>	gemeinsam	getrennt
<i>Speicher</i>	gemeinsam	getrennt
<i>Leistungsfähigkeit</i>	-	+
<i>Skalierbarkeit</i>	-	+
<i>Kompatibilität</i>	+	-
<i>Programmierung</i>	leicht	schwierig
<i>Parallelisierung durch ...</i>	Betriebssystem	Anwendungsprogrammierer
<i>Installat../Betrieb</i>	+	-
<i>Kosten</i>	+	-
<i>Verbreitung</i>	stark	noch gering
<i>Hauptanwendungen</i>	DBMS, Kommunikation	“number crunching”
<i>RDBMS-Beispiel</i>	Sybase <a href="#">SQL Server</a> (auch für EPS)	Sybase <a href="#">MPP</a> (dediziert)
<i>Hardwarebeispiel</i>	Digital <a href="#">AlphaServer</a>	IBM <a href="#">RS/6000 SP</a>

# Parallelisierungsverantwortliche

Betriebssystem parallelisiert ...

⇒ Operationen

⇒ Prozesse

Übersetzer parallelisiert ...

⇒ Operationen

⇒ Prozesse

Programmierer parallelisiert ...

⇒ Jobs und Programme

Hilfsprogramm parallelisiert ...

⇒ Jobs und Programme, z.B. nicht parallelisierbare ...

- Drittsoftware
- Legacy-Programme

*XpertRule Profiler* induziert Entscheidungsbäume mit SQL-Anweisungen, die von einem parallelen RDBMS verarbeitet werden können



---

# Parallelisierungsvarianten

---

## Funktionale Parallelisierung

Verschiedene **Aufgaben** für jeden Prozessor

## Daten-Parallelisierung

Verschiedenen **Daten** der gleichen Aufgabe für jeden Prozessor

## Pipeline-Parallelisierung

Verschiedene **Phasen** der gleichen Aufgabe für jeden Prozessor  
(z.B. Abfragephasen Scan, Verbund, Sortierung, Aggregation)

Ideal ist eine Kombination aller Parallelisierungsansätze

# Datenparallelisierung

---

## Verteilung der Eingabe/Ausgabe auf Festplatten

**Gemeinsame** Platte für mehrere Prozessoren

⇒ langsamer wegen *sequentieller* Eingabe/Ausgabe

**Mehrere** Platten für mehrere Prozessoren

⇒ schneller wegen *paralleler* Eingabe/Ausgabe

Wie kann ein Prozessor einer physischen oder logischen Platte (**Partition**) zugewiesen werden?

# Partitionierung für die Datenparallelisierung

**Parallelisierende Partitionierung** := Zuordnung  
einer **physischen oder logischen** Platte  
zu **einem** Prozessor  
damit jeder Prozessor **parallel** zu den anderen  
auf **eine** Platte zugreifen kann

Zuordnung durch ...

- ⇒ Hardware ( RAID)
- ⇒ Betriebssystem
- ⇒ Datenbanksystem

---

# Partitionierungskriterien

---

## Bereichspartitionierung

i.d.R. nach dem meist gebrauchten Schlüssel

fiktive SQL-Anweisung

```
CREATE TABLE Verkäufe(  
    Produktnummer NUM  
    Produktname CHAR  
    ...)  
PARTITION BY VALUE  
    Produkt = 'Racing Bike' IN partition_1,  
    Produkt = 'Mountain Bike' IN partition_2,  
    ...
```

## Hashpartitionierung

Partition je nach Wert einer Hashfunktion auf einem Tabellenwert

## Round Robin-Partitionierung

automatische Partitionierung nach der Verfügbarkeit der Prozessoren

# Trade Offs einiger Optimierungsmassnahmen

	<i>Vorteile</i>	<i>Nachteile</i>
<b>Sternschemata</b>	benutzerfreundlich relativ schnell relativ flexibel	unvollständig redundant
<b>Vorberechnung</b>	schnell	unvollständig speicheraufwendig unflexibel
unkomprimierte <b>Bitmusterindizes</b>	schnell flexibel	read only, speicheraufwendig, kleine Kardinalität
<b>Komprimierte</b> Bitmusterindizes	schnell flexibel grosse Kardinalität (bis Tausende)	read only, speicheraufwendig

---

# Inhalt

---

Grundlagen	<a href="#"><u>2</u></a>
Entwicklung - Überblick	<a href="#"><u>3</u></a>
Data Marts und Enterprise Data Warehouse	<a href="#"><u>6</u></a>
Mehrdimensionale Daten	<a href="#"><u>21</u></a>
Endbenutzerzugriff	<a href="#"><u>34</u></a>
OLAP	<a href="#"><u>44</u></a>
ANLAGEBERATUNG mit <i>PowerPlay</i> 📌	<a href="#"><u>71</u></a>
Modellierung relationaler Data Warehouses	<a href="#"><u>84</u></a>
Modellierung von Informationssystemen	<a href="#"><u>85</u></a>
Sternschema ( LIEFERFRIST )	<a href="#"><u>97</u></a>
EDW-Modell und Sternschema ( VERKAUF )	<a href="#"><u>106</u></a>
<i>Synchrony</i> ( EINZELHANDEL 📌 )	<a href="#"><u>126</u></a>
Entwicklung und Betrieb	<a href="#"><u>136</u></a>
Fallbeispiel ( ZEITSCHRIFTEN 📌 )	<a href="#"><u>151</u></a>
Effizienz	<a href="#"><u>160</u></a>
Netzzugriff	<a href="#"><u>198</u></a>
Client/Server-Architekturen	<a href="#"><u>203</u></a>
Zugriff über Internet und Intranet	<a href="#"><u>213</u></a>

## 5.71 Vom Host zum Client/Server-System

<i>Umgebung</i>	<b>zentrale</b> Aufgaben ( Host oder Server)	<b>lokale</b> Aufgaben (Terminal oder Client)
<b>Host-</b>	Verarbeitung, Interaktion	E/A-Funktionen dummer Terminals
<b>Master/Slave-</b>	Verarbeitung, Interaktion	E/A-Funktionen intelligenter Terminals
<b>Fileserver-</b>	Drucker und Speicher teilen, Dateiaustausch	Verarbeitung, Interaktion
<b>Zweistufige C/S-</b>	Daten, Verarbeitungssteile	Verarbeitungssteile, Interaktion
<b>Mehrstufige C/S-</b>	Daten, Verarbeitung durch dedizierte Applikationsserver	Interaktion

# Rechnernetze

## Rechnernetze

- **Lokale** Netze erstrecken sich in der Regel auf ein Grundstück (LAN: Local Area Network, Bsp. Novell *Netware*)
- **Verbund**netze sind örtlich verteilt (WAN: Wide Area Network, Bsp. *Internet* →)

## Komponenten

- **Clients** (inkl. Netzkarte + Netzbetriebssystem) →
- **Server** (inkl. Netzkarte + Netzbetriebssystem) →
- **Leitungen** (Telefon-, Koaxial-, Glasfaserkabel)
- **Kommunikationsprotokolle** (Bsp. TCP / IP) →



Die Internet-Technologie bietet eine standardisierte Infrastruktur für LAN- und WAN-Bedürfnisse



# Rechnernetze teilen Ressourcen

<i>Ressource</i>	<i>Beispiele</i>
<b>Hardware</b>	Prozessoren Speicher Drucker Modem Fax
<b>Kommunikation</b>	Telefon Fax Internet
<b>Software</b>	Datenbankmaschine OLAP- und Data Mining-Werkzeuge Kommunikationssoftware (z.B. E-Mail)
<b>Daten</b>	Fileserver-Dateien Produktionsdatenbank Data Warehouse E-Mail-Nachrichten

# Rechnernetze für Data Warehouses

---

- ⇒ **Verteilte** Datenbanken
- ⇒ Laden aus den **Produktions**datenbanken
- ⇒ Laden aus dem **Enterprise** Data Warehouse
- ⇒ Endbenutzerzugriff auf Data **Marts**
- ⇒ . . .

Netzwerk-Bandbreite!

# Client/Server-Systeme

**Client/Server-System** :=

Verbund von Hardware und Software, der

Aufträge von **Clients** (Arbeitsplatzrechnern)

durch **Server** (Dienstleistungsrechner und -programme)  
ausführen lässt

*Eine mögliche Aufgabenteilung*

<b>Client</b>	<b>Server</b>
Benutzereingaben lesen, prüfen und senden	Abfragen empfangen und Ergebnisse senden
Ergebnisse empfangen, formatieren und ausgeben	Datenbankzugriffe durchführen
	Integrität, Sicherheit und Datenschutz gewährleisten

## 5.72 Alternative Modelle der Aufgabenteilung

<i>serverzentriert</i>		<i>verteilt</i>	<i>clientzentriert</i>	
<i>Präsentation verteilt</i>	<i>Präsentation entfernt</i>	<i>Anwendung verteilt</i>	<i>Daten entfernt</i>	<i>Daten verteilt</i>
Daten	Daten	Daten	Daten	Daten
Anwendung	Anwendung	Anwendung	Netzwerk	Netzwerk
Präsentation	Netzwerk	Netzwerk	Anwendung	Daten
Netzwerk	Präsentation	Anwendung	Anwendung	Anwendung
Präsentation	Präsentation	Präsentation	Präsentation	Präsentation

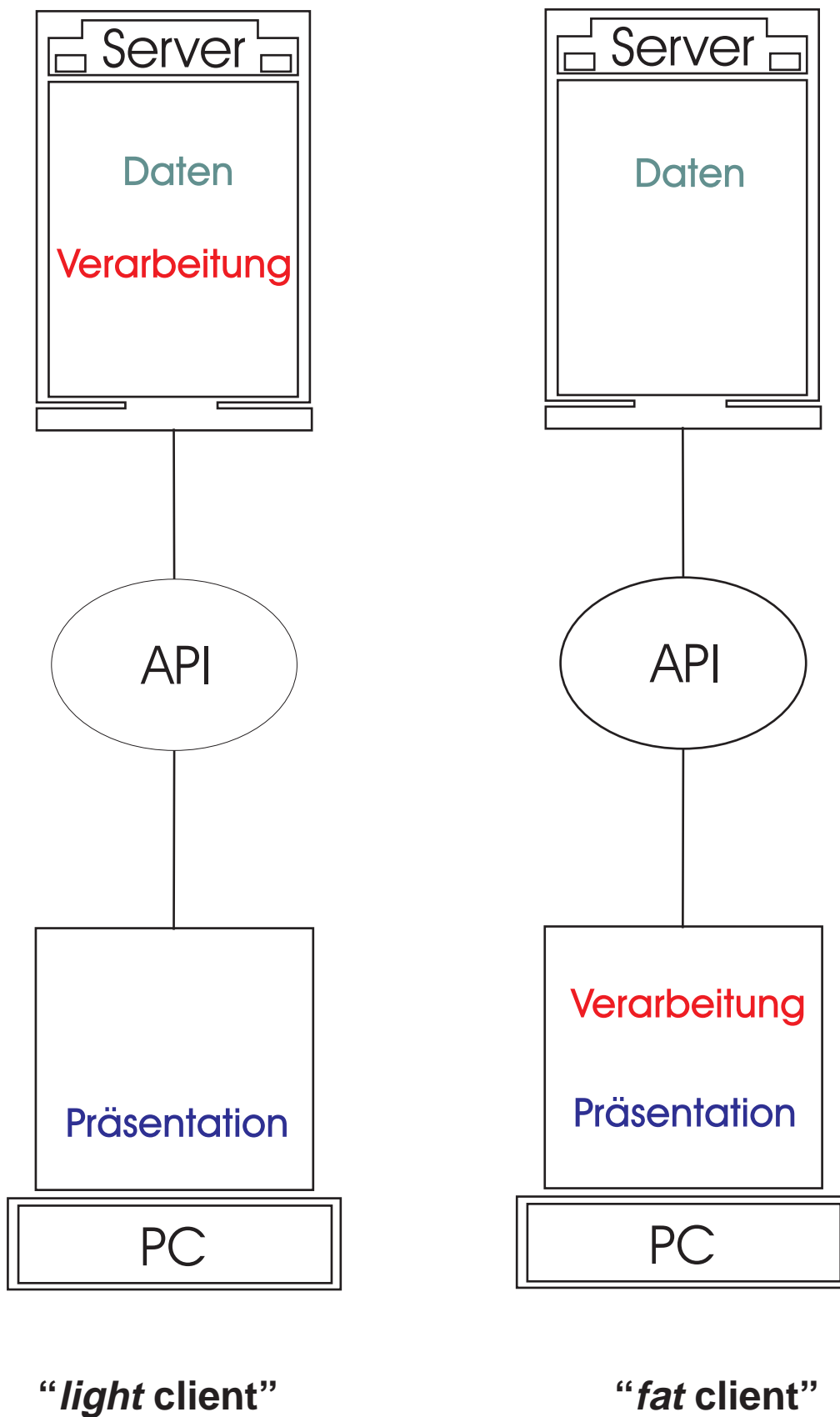
Server(s)

Client

## 5.73 Client/Server-Architekturen

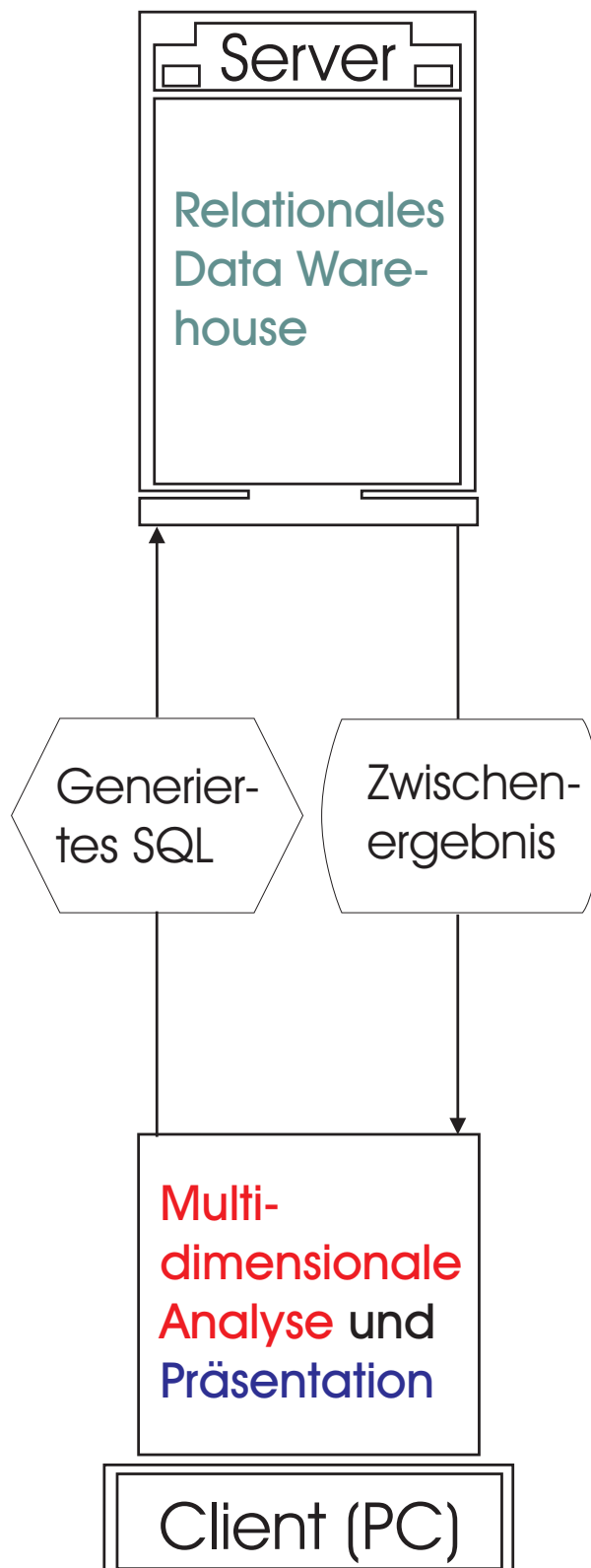
<i>Architektur</i>	<i>Erstrechner (Client)</i>	<i>Zweitreechner (Server)</i>	<i>Dritt- (Server)</i>	<i>Netz- verkehr</i>
<b>einstufig</b> (one-tiered)	Daten Verarbeitung Präsentation			-
<b>zweistufig</b> (two-tiered, "light client")	Daten	Verarbeitung Präsentation		Abfrage, Antwort
zweistufig (two-tiered, "fat client")	Daten Verarbeitung	Präsentation		Abfrage, Zwischen- ergebnis
<b>dreistufig</b> (three-tiered)	Daten	Verarbeitung	Präsen- tation	Abfrage, Zwischen- ergebnis, Antwort

# Zweistufige Client/Server-Architekturen



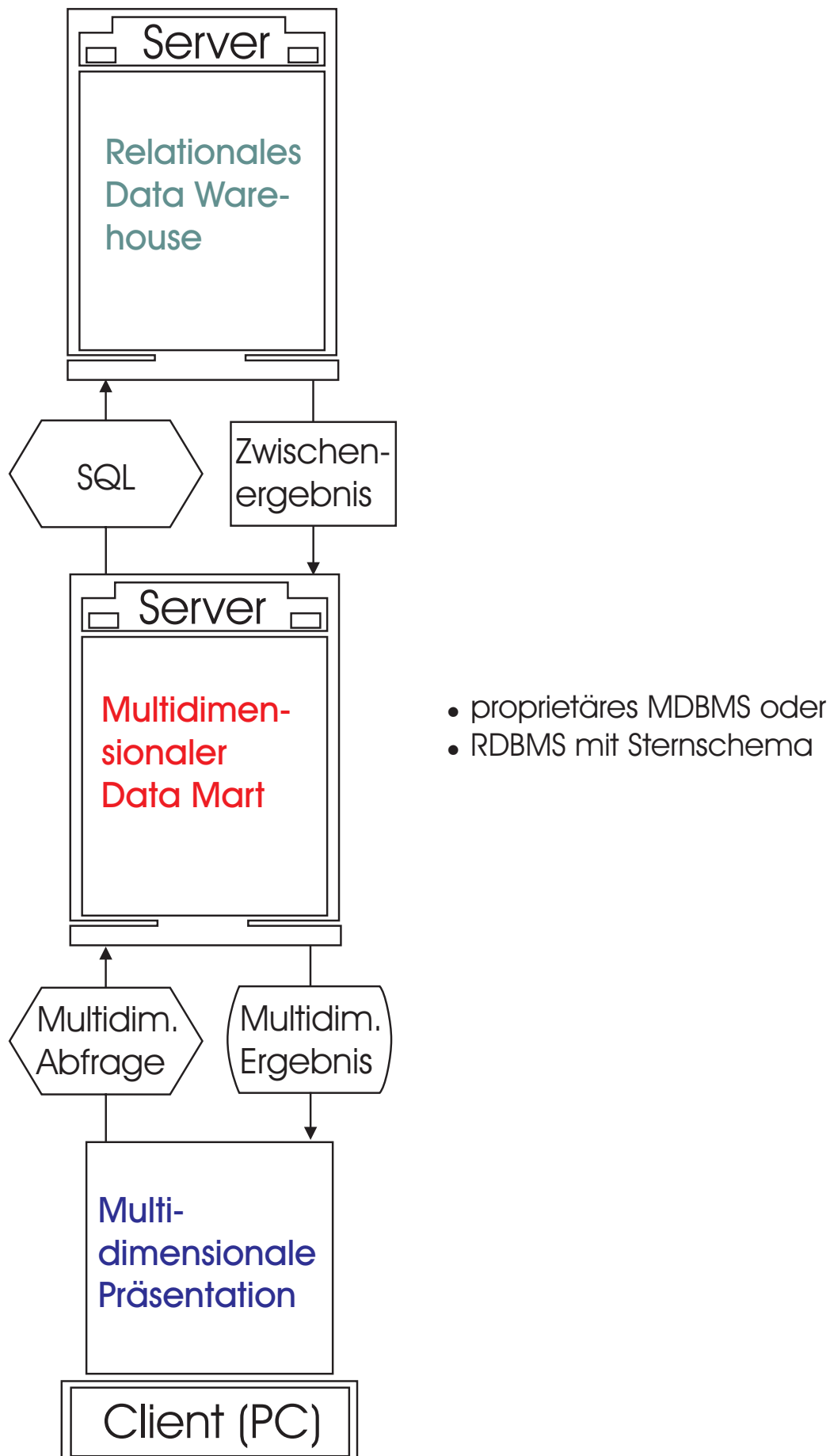
API := Application Programming Interface

## 5.74 Zweistufige Data Warehouse-Architektur



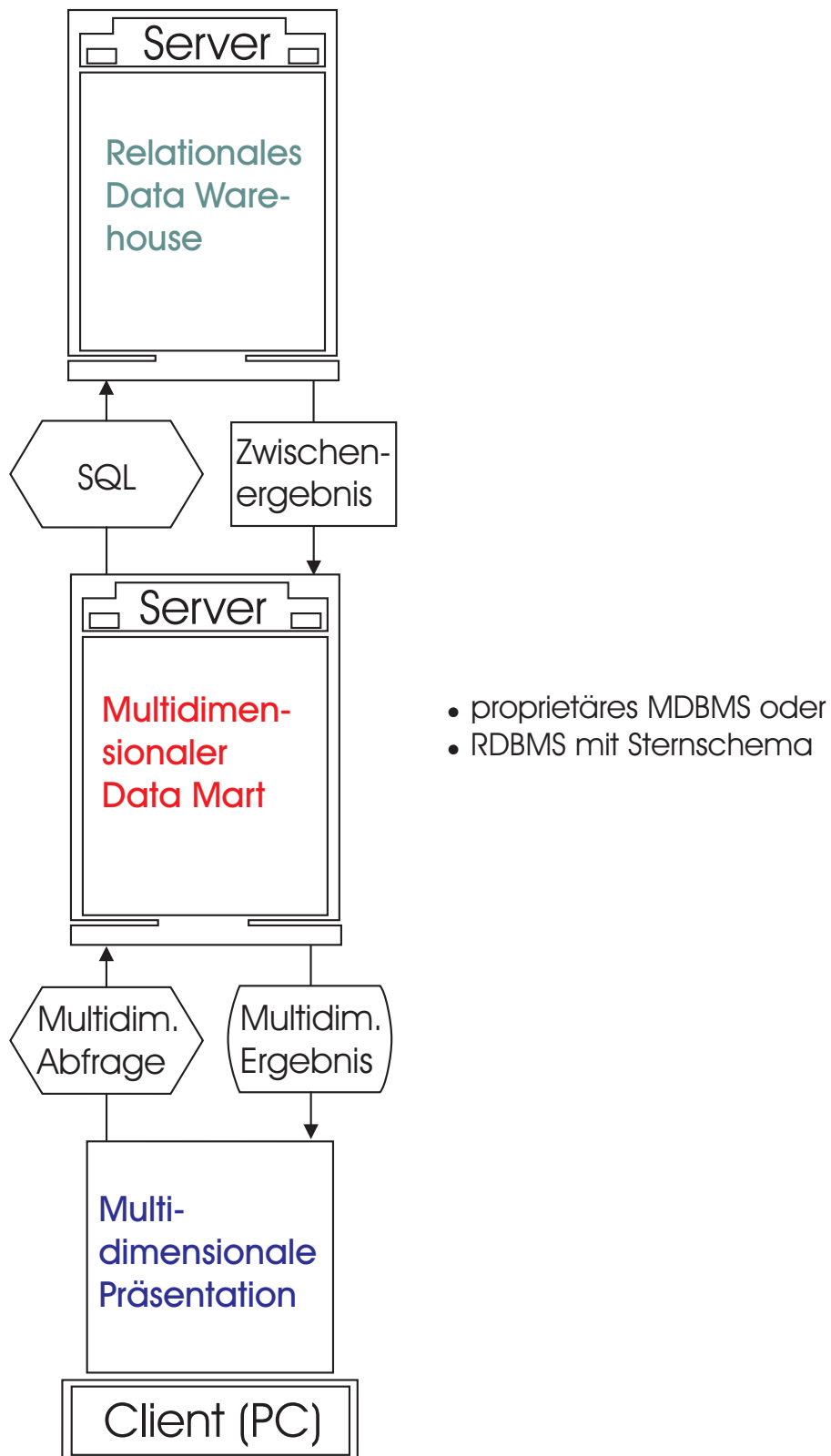
Bsp. BrioQuery von Brio Technology

# Dreistufige Client/Server-Architektur



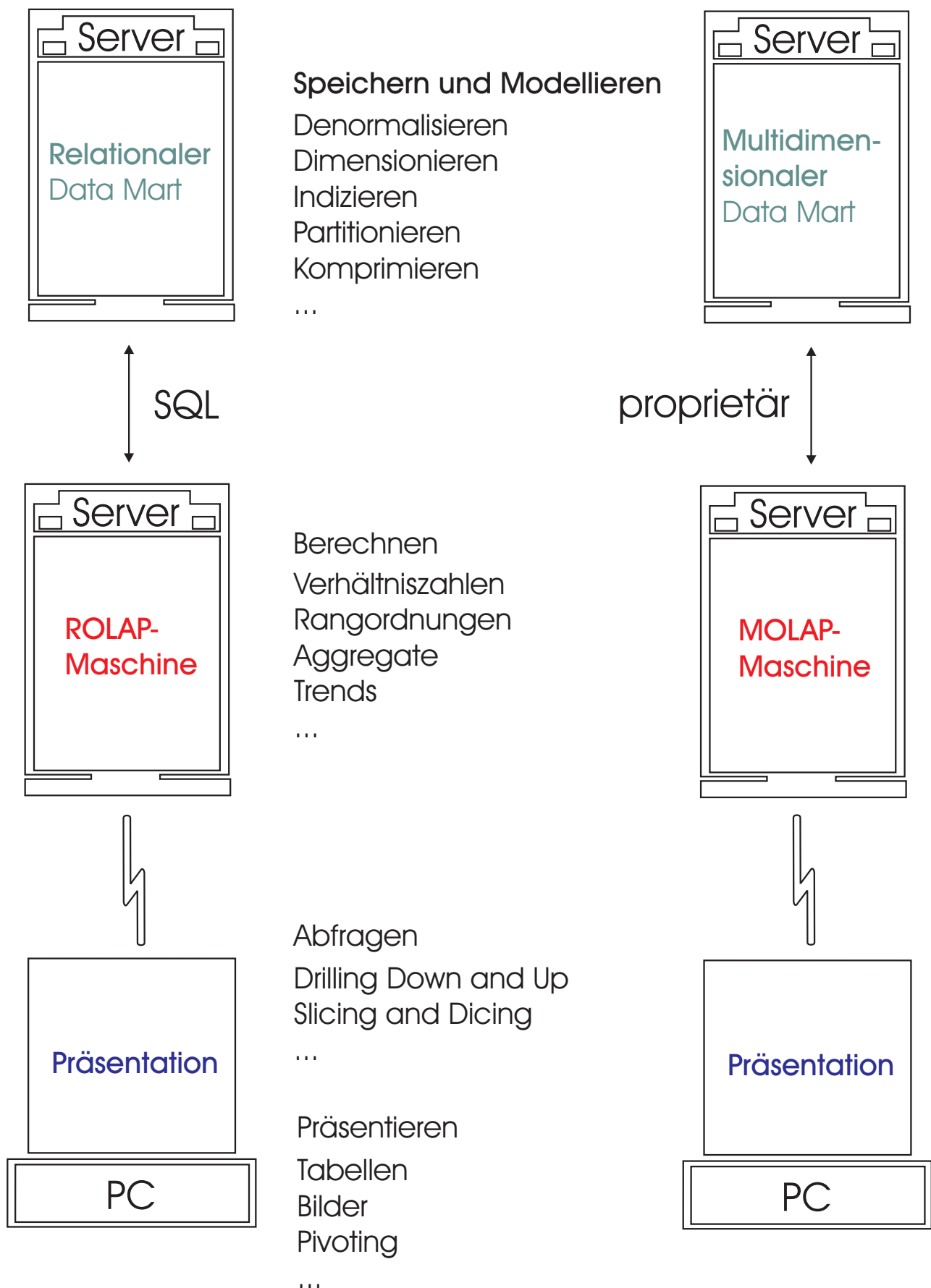


## 5.75 Dreistufige Data Warehouse-Architektur



Bsp. DecisionSuite von *Information Advantage*

# Dreistufige ROLAP- und MOLAP-Architektur



# Clientzugriff auf Data Warehouses

---

⇒ OLAP-Werkzeug

Endbenutzerwerkzeuge wie *Cognos* PowerPlay

⇒ Berichtsgenerator

⇒ Tabellenkalkulationspaket und OLAP-Add Ins

Microsoft Excel

Microsoft Excel Add In BusinessQuery von *Business Objects*

⇒ Zweidimensionales Abfragewerkzeuge

QBE, SQL

⇒ endbenutzernahes Client-Datenbanksystem

*Microsoft* Access

⇒ Groupware

*Lotus* Notes

⇒ Eigenentwicklung mit 4GL-Umgebungen

*Microsoft* Visual Basic, *PowerSoft* PowerBuilder

⇒ Web Browser und OLAP Add Ins →

ActiveX-Komponenten mit OLAP-Funktionalität

⇒ Data Mining-Werkzeug

*Cognos* Scenario

⇒ Text Mining-Werkzeug

*Oracle* ConText

⇒ Statistiksoftware-Frontend

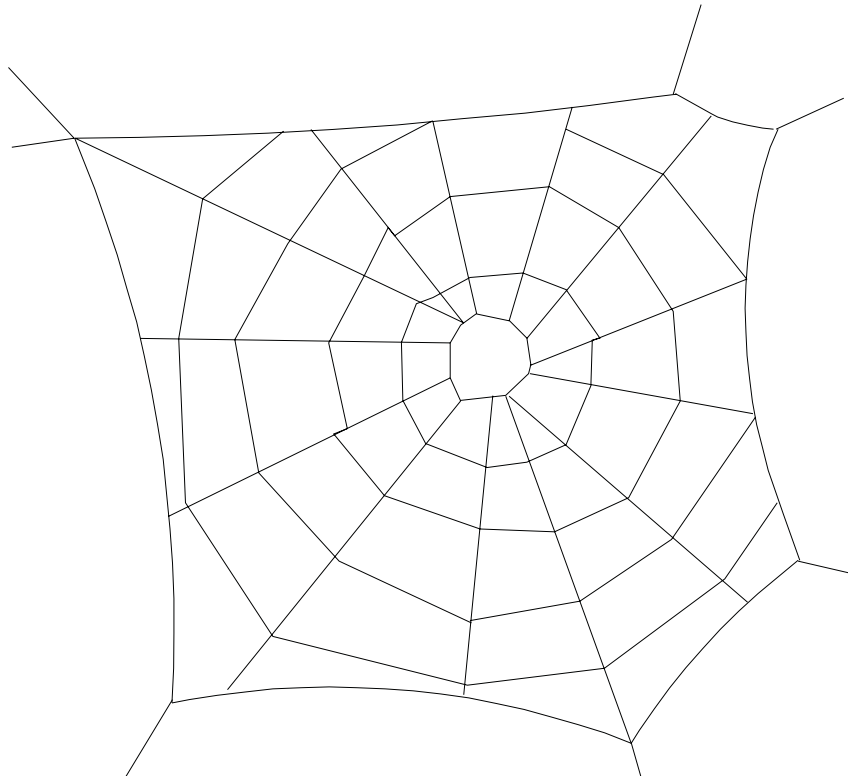
SAS, SPSS

# Endbenutzerzugriff auf Data Warehouses

	<b>Lokales C/S-Netz</b>	<b>Intranet/Internet</b>
<i>Plattformunabhängig</i>		
<i>GUI weit verbreitet</i>		
<i>GUI-Funktionalität hoch</i>		
<i>Werkzeugbeispiele</i>	komfortables OLAP-Werkzeug	Internet Browser (ev. mit OLAP- Erweiterungen)

Webzugriff

# Endbenutzerzugriff über Intranet und Internet



Der ideale Zugriff ist ...

plattformunabhängig

benutzerfreundlich

kostengünstig

Internet und Intranet

---

# Intranet für Data Warehouses

---

## ✓ Benutzerfreundlichkeit

- bedienerfreundlicher hypermedialer Web Browser

## ✓ Plattformunabhängigkeit

- einheitliche Frontend-Software  
(v.a. für MS Windows, Mac und Unix)
- einheitliche Webserver-Software

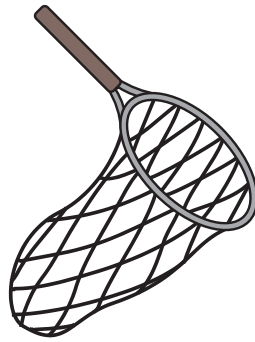
## ✓ Produktivität

- bekannte Technologie
- portable Entwicklungswerkzeuge
- kostengünstige Entwicklungswerkzeuge

## ✓ Sicherheit

- Zugriff aus dem öffentlichen Internet nicht möglich
- Lesezugriff auf Data Warehouses genügend sicher

# Exkurs - Das Internet verbindet Rechnernetze



**Internet** :=

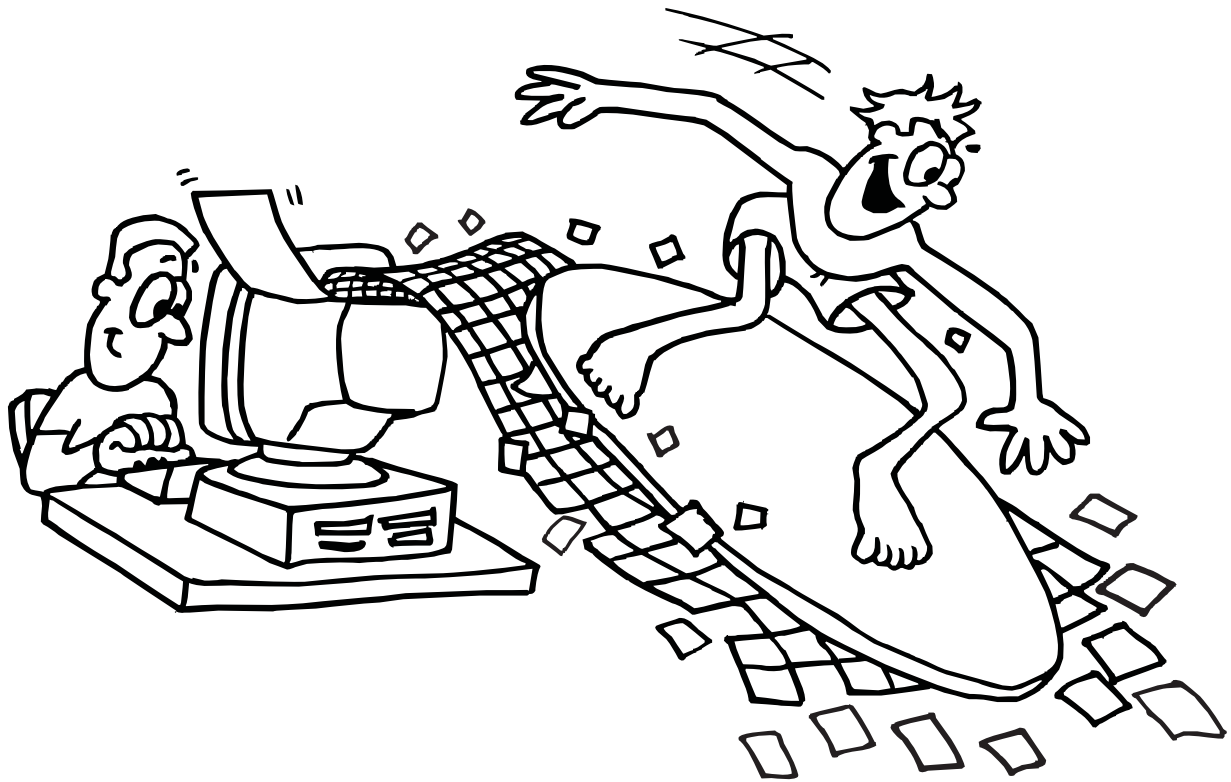
- ✓ weltweites ...
- ✓ öffentliches Netz ...
- ✓ aus Tausenden von Netzen und Millionen von Teilnehmern, ...
- ✓ die fehlertolerant kommunizieren und ...
- ✓ Dienste wie elektronische Post und Webpublikation teilen

## Geschichte

- **60er Jahre**  
Sichere Kommunikation zwischen Militärs
- **70er und 80er Jahre**  
Schnelle Kommunikation zwischen Hochschulangehörigen
- **90er Jahre**  
Sichere und schnelle Kommunikation für jedermann

## Eine kommerzielle Anwendung

*Merrill Lynch Financial Services* kommunizieren mit Zehntausenden von Finanzberatern und Kunden über das Internet und sparen so Millionen von Kilo Papier pro Jahr.



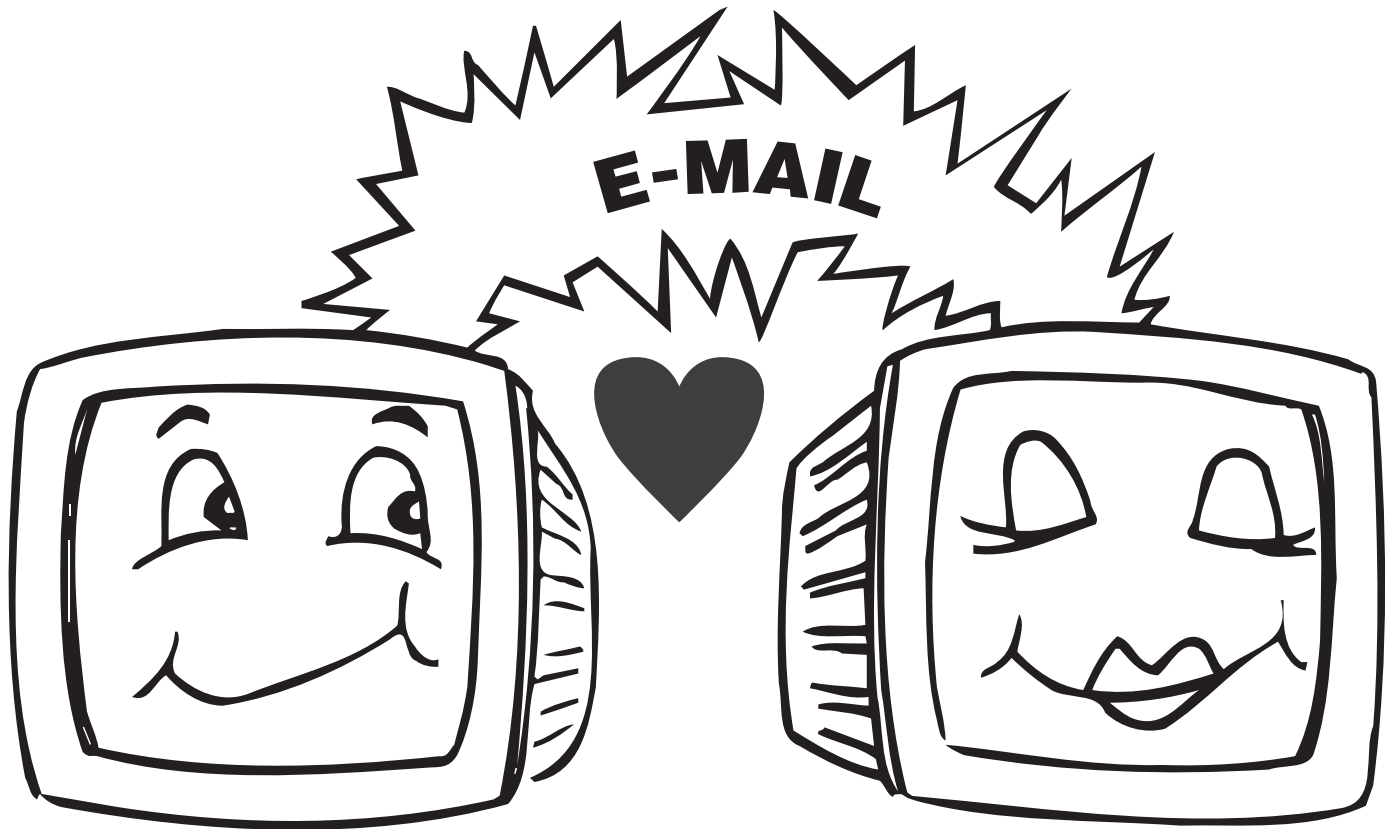
World Wide Web (WWW) :=

- ✓ benutzerfreundliche Oberfläche, ...
- ✓ auf der Endbenutzer hypermedial durch
- ✓ die wichtigsten Internetdienste navigieren

## Geschichte

- **1989:** *Konzept* am CERN Genf (Berners-Lee und Cailliau)
- **1993:** *Browser* am nationalen US-Hochleistungsrechenzentrum (Andreessen als Millionär in Jeans und Turnschuhen)





Elektronische Post (E-Mail)

Publikation

Bsp. WWW (World Wide Web) →

Diskussion

Bsp. News Groups

Datentransfer

Bsp. FTP (File Transfer Protocol) → Data Warehouse

Bestellung und Bezahlung

Bsp. Pay-Per-View Publishing

...

## Exkurs - Wer regiert das Internet?



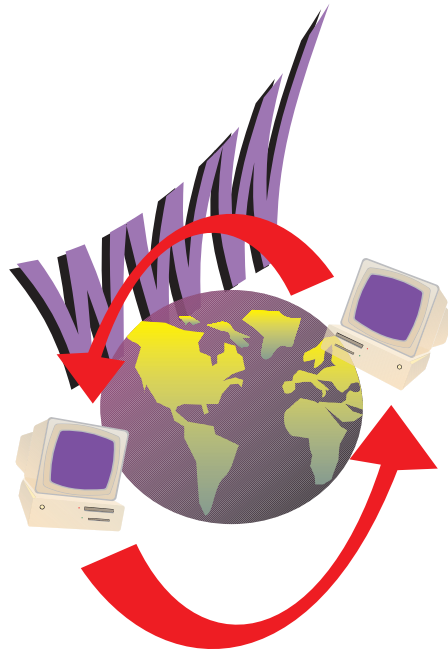
### Internet Activities Board (IAB)

Das Internet **gehört** nicht einzelnen Unternehmungen oder Gruppen

Internetgremien können nicht vorschreiben und sanktionieren sondern nur **empfehlen**

**Hersteller** dominieren die Internetgremien

Der **Ausbau** des Internet ist unkoordiniert



Das WWW ist ...

plattformunabhängig

*MS Windows, OS/2, MacOS, Unix, ...*

benutzerfreundlich

kostengünstig

aber (noch) ...

oft langsam

manchmal unsicher

nicht genügend funktional

## Seiten laden und anzeigen

durch Anklicken von Hyperlinks

durch Eingabe von URLs

## Navigieren

innerhalb einer Webseite

über mehrere Webseiten

## Seiten durchsuchen

Volltextsuche auf der laufenden Seite

Suche über mehrere Webseiten (Suchmaschinen)

## Seiten oder Ausschnitte speichern

Speicherungsformat wählen

(HTML, ASCII/ANSI oder ein Textverarbeitungsformat)

Gewählte Texte und Grafiken speichern

## Browser anpassen, z.B.

Startseite wählen

Favoriten speichern

---

# Exkurs - Arbeitsplatz auf dem WWW einrichten

---

## Hardware- und Softwareanforderungen

PC

Modem oder LAN-Anschluss

Anschluss bei einem Internet Service Provider

Browser

## Laufende Kosten

Grundgebühren je nach Funktionalität

Benützungsgebühren

## Probleme

Übersicht auf dem WWW ("getting lost in the hyperspace")

Überlastung zu Spitzenzeiten

Urheberrecht, Vertraulichkeit, Sittlichkeit

# Exkurs - Mit *analogem* Modem auf das Internet



**Modem** (Modulator und Demodulator) :=

Gerät, das zwischen *analogen* Telefon- und *digitalen* Computersignalen konvertiert

## Modem

- **internes** - (als Karte am Bus)
- **externes** - (über einen Stecker an der seriellen Schnittstelle) <sup>1</sup>

## Datenübertragung

- Verbindung **aufnehmen**
- Geschwindigkeit **aushandeln**
- Verbindung **abbrechen**

## Anforderungen

- **hohe Geschwindigkeit** (Bsp. 28'800 bps)
- **Faxfähigkeit**
- **Hayes-Kompatibilität**

---

<sup>1</sup> Vor- und Nachteile interner und externer Modems?

---

# Exkurs - *Digitales* ISDN statt analoges Modem

---

**ISDN** (Integrated Services Digital Network) :=

öffentliches Netz, das ...

verschiedene Telecom-Dienste integriert

## Integrierte Dienste

Telefon

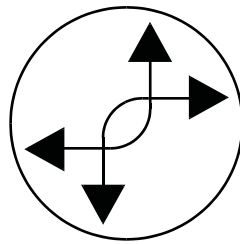
Übertragung von Computerdaten (Text, Grafik, Video, Audio)

Videokonferenz

Bildschirmtext

Videotex

Sender  $K_1$  **verpackt**



Internet **transportiert**

Empfänger  $K_n$  **entpackt**

$K_1$  **adressiert** Datenpakete an Empfänger  $K_n$

**Router** verbindet  $K_1$  und  $K_n$  über die Knotenrechner  $K_2 \dots K_{n-1}$ .  
(Bei einem Ausfall ändert das Transportprotokoll TCP / IP → den Weg automatisch)

$K_n$  **empfängt** die Datenpakete



## TCP verpackt ...

(Transmission Control Protocol)

eine Nachricht in Datenpakete

## IP transportiert sie ...

(Internet Protocol)

auf dem schnellsten Weg

## TCP ...

prüft alle Datenpaket und

setzt sie geordnet zusammen

---

# Exkurs - Internetadressen

---

## Jeder Internetrechner benötigt eine eindeutige Adresse

( URL, Uniform Resource Locator ), zum Beispiel ...

`http: // www.wwz.unibas.ch / wi/eus / home.html`

Hypertext Transfer Protocol zwischen Client und Webserver

Domäne (WWW-Angebot des WWZ der Universität Basel, CH<sup>1</sup>)

ein Verzeichnis (wi/eus) auf dem Webserver

Webseite HOME des Typs HTML (Hypertext Markup Language)

---

<sup>1</sup> Oft steht statt CH die Domäne COM. Was bedeutet .COM ?

---

# Exkurs - Webadressen und -inhalte versenden

---

**Sender einer Webseite**

Anwendungsschicht ( HTTP-Protokoll)

LAN- oder Modemschnittstelle

Transportschicht ( TCP-Protokoll)

Internetschicht ( IP-Protokoll)

LAN- oder Modemschnittstelle

Anwendungsschicht ( HTTP-Protokoll)

**Empfänger der Webseite**

---

# Exkurs - Webseiten erstellen

---

**Zielgruppen** bestimmen

**Inhalt** spezifizieren

- Vorhandene Seiten übernehmen
- Neue Seiten gestalten

**Betreuer** der Web Site bestimmen

**Aufbau** der Web Site entwerfen

- Home Page (Einstiegsseite) und übrige Seiten
- Lesestrassen

**Werkzeuge** auswählen

- Browser → (Bsp. [MS Internet Explorer](#))
- Konventioneller Editor (Bsp. [MS Office](#))
- Dedizierter Editor (Bsp. [MS FrontPage](#) →)

**Vorlagen** erstellen

- Bildvorlagen (v.a. Symbole und Logos)
- Seitenvorlagen

**Inhalt und Form** implementieren und testen

Bei **Suchmaschinen** anmelden

**Feedback** ermöglichen

---

# Exkurs - Geschäft auf dem WWW gründen

---

## 1. Merkfähige Adresse finden

- Bsp. McDonald

## 2. Verbindungsart wählen

- Eigene Server-Hardware und -software  
Bsp. MS NT Server
- Leitung zum WWW-Server eines Anbieters (engl. Provider)

## 3. Aktive und passive Dokumente erstellen

- Text-, Bild- und Tonfolgen in HTML  
Bsp. MS FrontPage

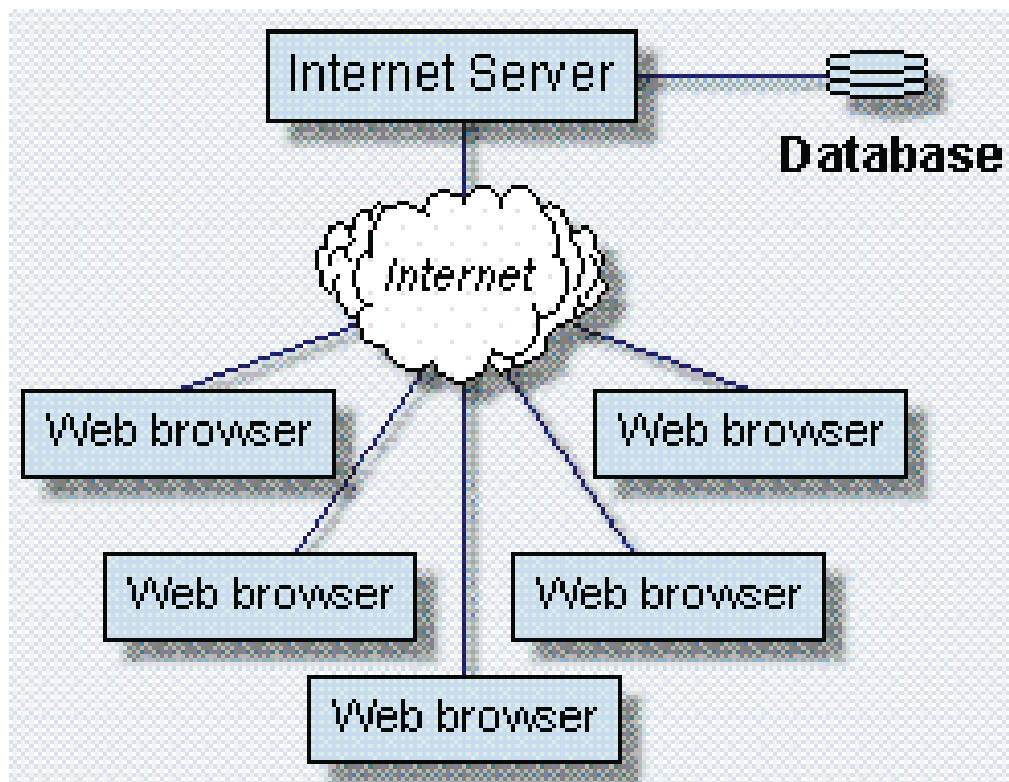
## 4. Zahlungsverkehr regeln

- Kreditkarte
- e-cash

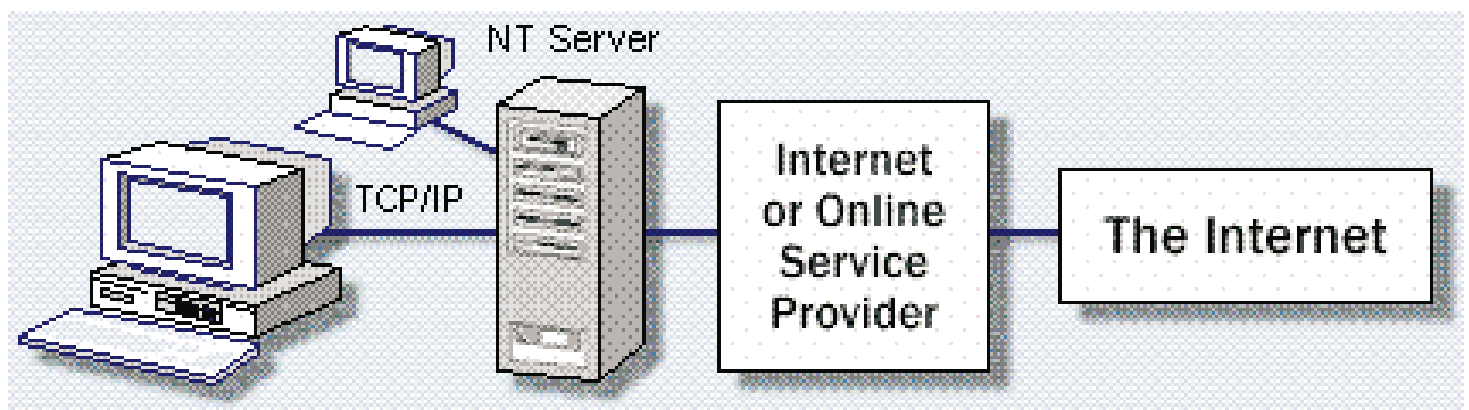
## 5. Werben

- Home Page bei Suchmaschinen anmelden  
Bsp. Alta Vista
- ...

# Intranet - Ein Internet für die Unternehmung



**Internet**



**Intranet**

**Internet**

PC

Grossrechner-Terminals

LANs

Zugang zum Internet

# Exkurs - Betriebliche Webanwendungen

<b>Published Information</b> <i>Static Information:</i>	<b>Workgroup Collaboration and Workflow</b>
Corporate policies and procedures	Budgeting process
Customer and product information	Bulletin boards
Human resources info. and job listings	Engineering design and manufacturing
Instructional materials	Group communications through e-mail
Research materials and library system access	Group scheduling including mobile workers
Maps and directions	Human resources processes
Org. structures and employee data	Marketing plan
<i>Information supporting ad-hoc analysis:</i>	Expense report process (administrative workflow)
Corporate financial information	<b>Interactive Business Processes</b>
Corporate templates	Electronic commerce (e.g. company store) and order tracking
Real time inventory and sales data	Live links between bus. partner systems
Scheduling data	Customer support services
Research data	Help desk services
	Manufacturing inventory system
	Selecting health benefits

---

# Exkurs - HTML als 1. Web-Beschreibungssprache

---

**Hypertext Markup Language (HTML)** := geschlossene deklarative Sprache, die vor allem die *Erscheinung* (das Format) und weniger den Inhalt (die Daten) von Webseiten beschreibt

```
<HTML>
  <HEAD>
    <TITLE>Formatierungsbeispiel</TITLE>
  </HEAD>
  <BODY>
    ...
    <P ALIGN="LEFT">
      <FONT COLOR="000000" SIZE="3">...</FONT>
    </P>
    ...
  </BODY>
</HTML>
```



# Exkurs - XML - Neuere Beschreibungssprache

**EX**tensible **M**arkup **L**anguage (XML) := erweiterungsfähige deklarative Sprache, die nicht nur die Form sondern auch den *Inhalt* von Webseiten beschreiben kann

## Bsp. Domänenspezifische Beschreibungen

**<BESTELLUNG>**

<VON>

<PERSON>

<GESCHLECHTSNAME>Müller</GESCHLECHTSNAME>

<VORNAME>Hans</VORNAME>

</PERSON>

</VON>

<PRODUKT>

<GRUPPE>Buch</GRUPPE>

<NAME>Einführung in XML</NAME>

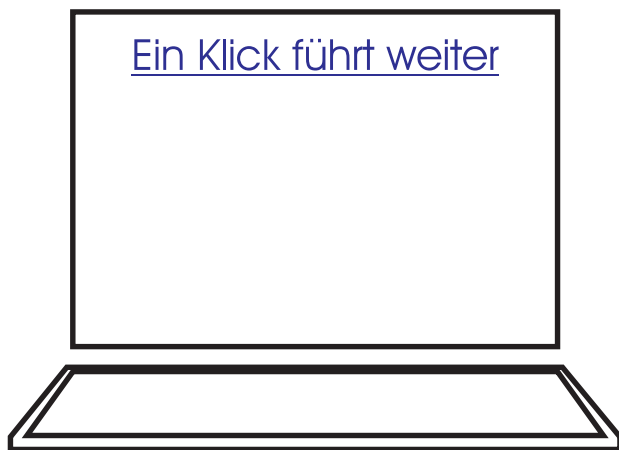
...

</PRODUKT>

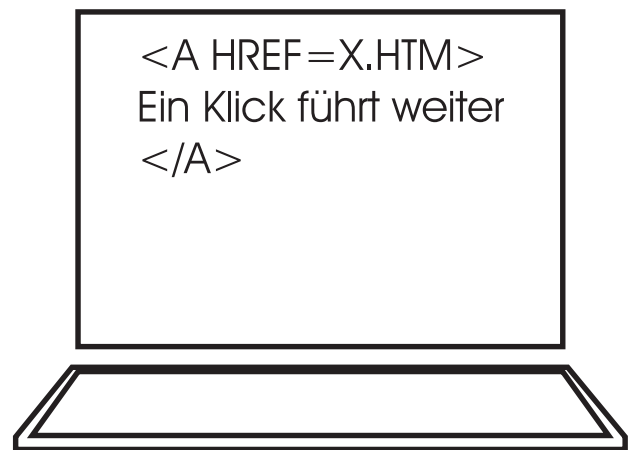
...

**</BESTELLUNG>**

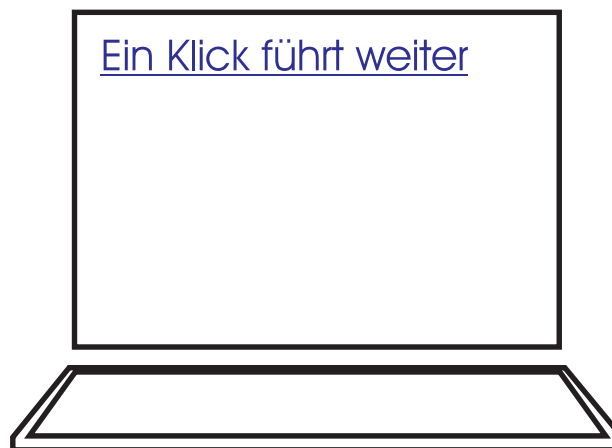
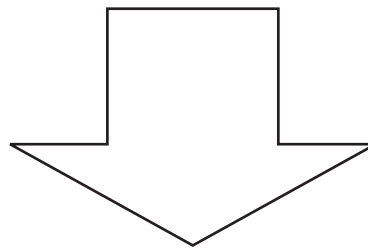
## 5.77 Spezifikation/Verarbeitung von Webseiten



**Dedizierter WYSIWYG-Editor**  
(z.B. MS FrontPage) formatiert  
Text und Bilder (\*.gif, \*.jpg)  
mit *verborgenen* Codes

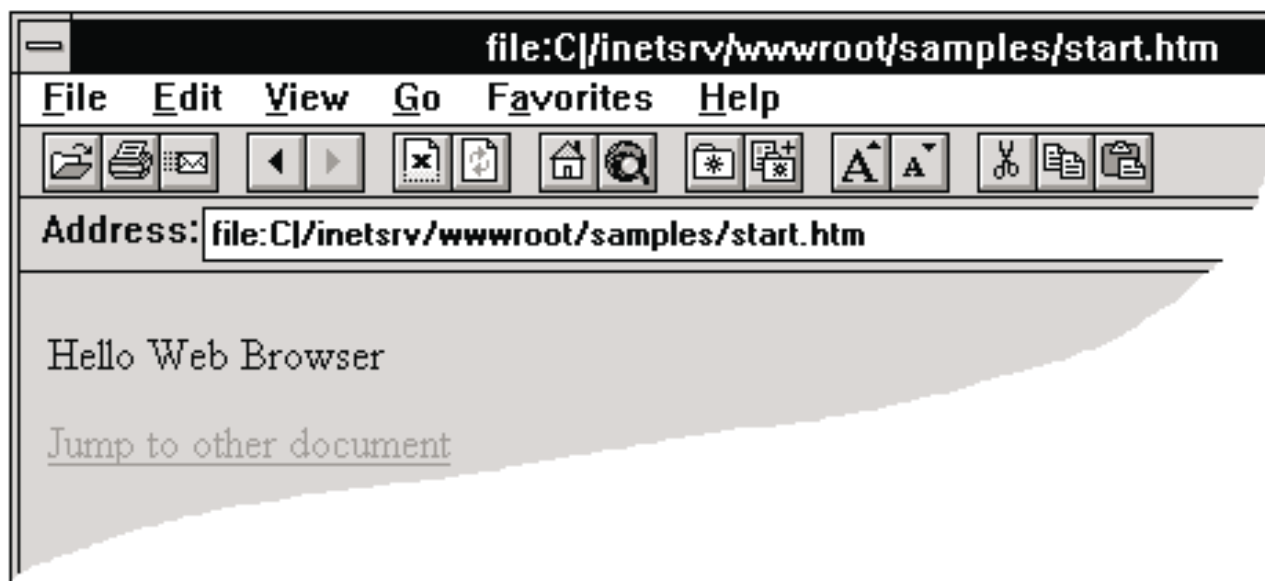


**Allgemeiner Texteditor**  
(z.B. MS Write) formatiert  
Text und Bilder (\*.gif, \*.jpg)  
mit *sichtbaren* Codes



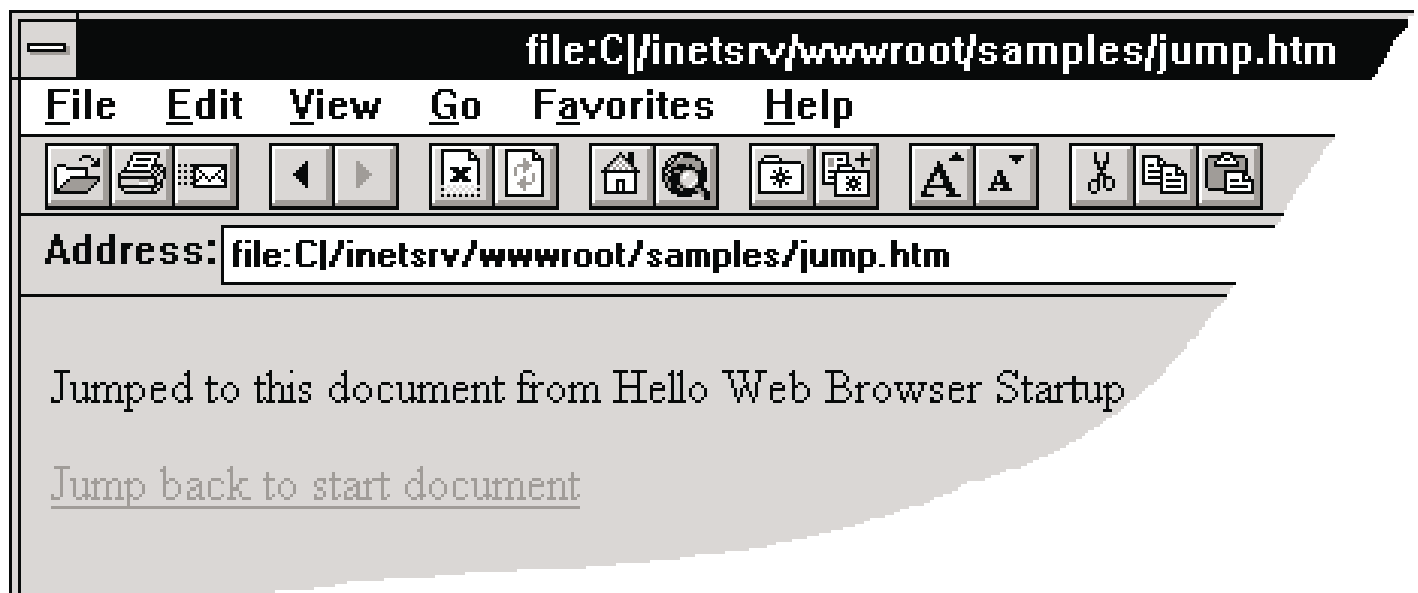
**Browser** (z.B. MS Internet Explorer oder  
Netscape Navigator / Communicator)  
analysiert den mit HTML markierten  
Text der empfangenen Webseiten mit  
einem Parser und interpretiert die  
eingebetteten Formatcodes (<...>)

# Exkurs - Hyperlinks in HTML darstellen



```
<p>Hello Web Browser Startup</p>  
<A HREF="jump.htm">Jump to other document</A>
```

*START.HTM document with HTML source code.*



```
<p>Jumped to this document from Hello Web Browser Startup</p>  
<A HREF="start.htm">Jump back to start document</A>
```

*JUMP.HTM document with HTML source code*

# Datenzugriff über das Web - Überblick

## ① HTML

Browser interpretiert in den Text eingebettete **HTML**-Befehle

⇒ Netz stark beansprucht

⇒ Interaktivität stark eingeschränkt

## ② HTML und Ergänzungen

### Clientseitige Ergänzungen

a) Browser interpretiert zusätzlich eingebettete **Skript**befehle  
Bsp. JavaScript, VBScript

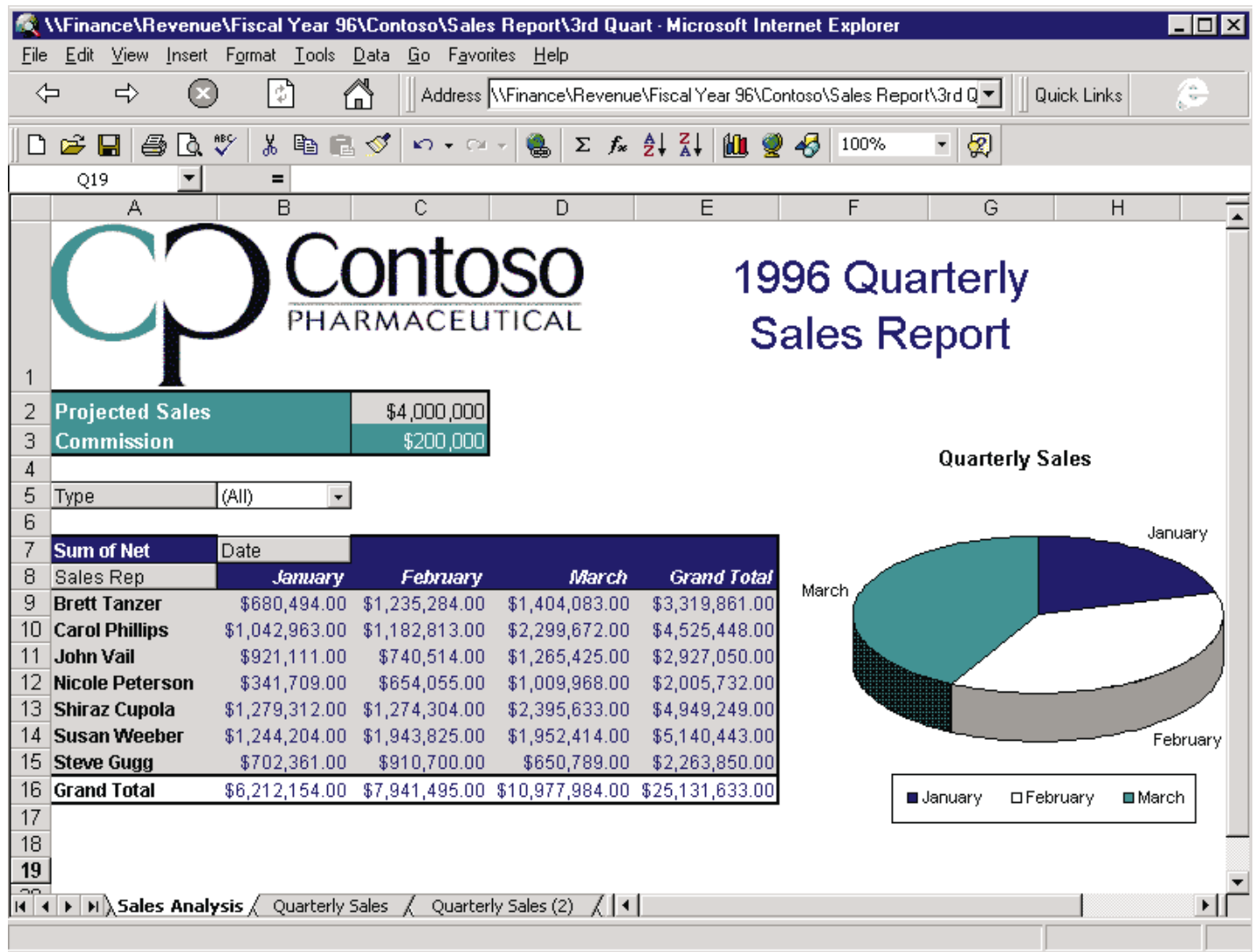
b) Browser ruft **Bibliotheksfunktionen** auf  
Bsp. ActiveX, Applets

### Serverseitige Ergänzungen

Datenbankserver interpretiert Befehle einer ...

- *portablen* Schnittstelle des Webserver  
CGI (langsam)
- *prioprietären* Schnittstelle des Webserver  
ISAPI, NSAPI, ... (schneller)

# Interaktive statt passive Webseiten



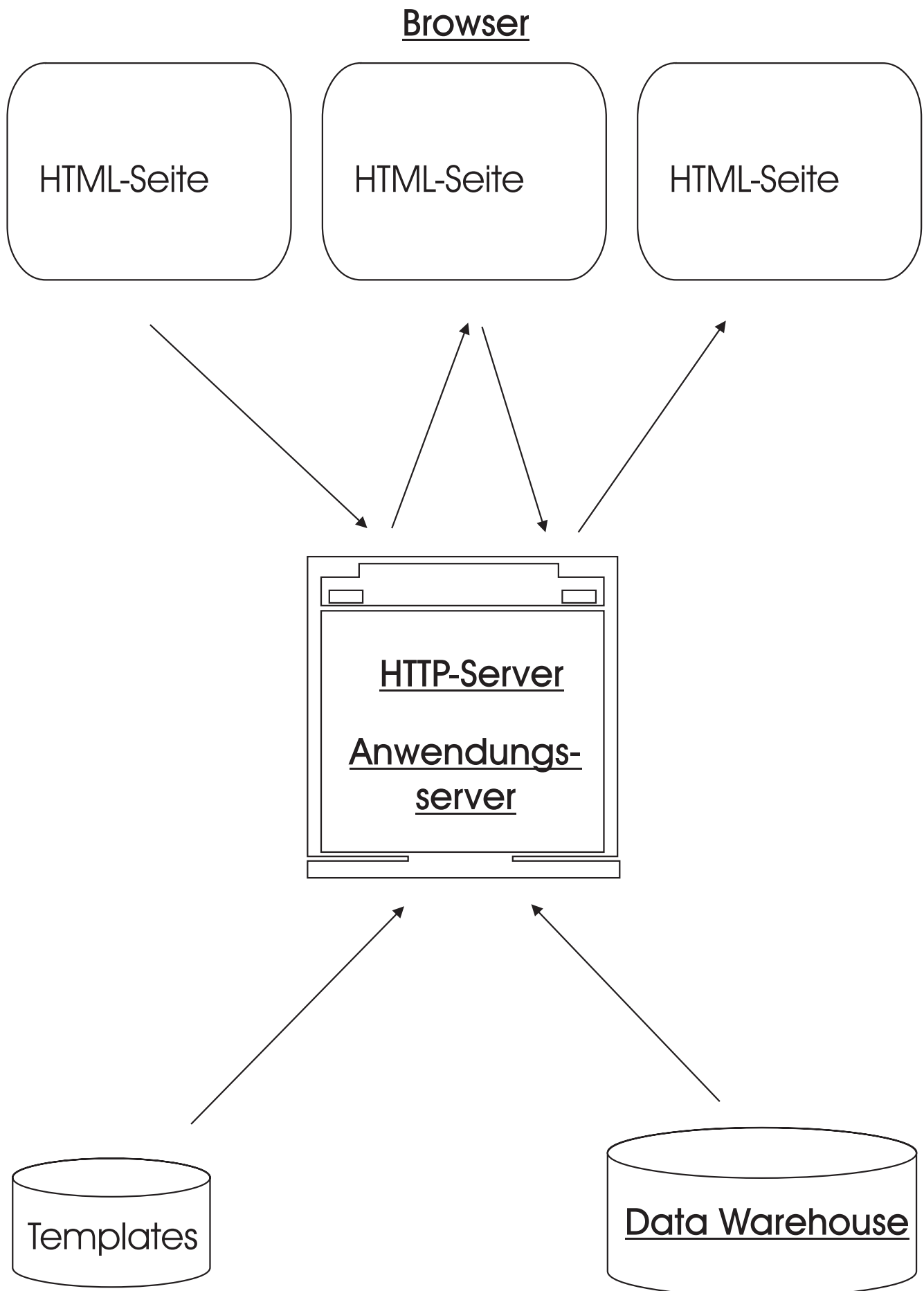
Arbeitsmappe aus MS Excel

Standard-Steuerelemente von HTML

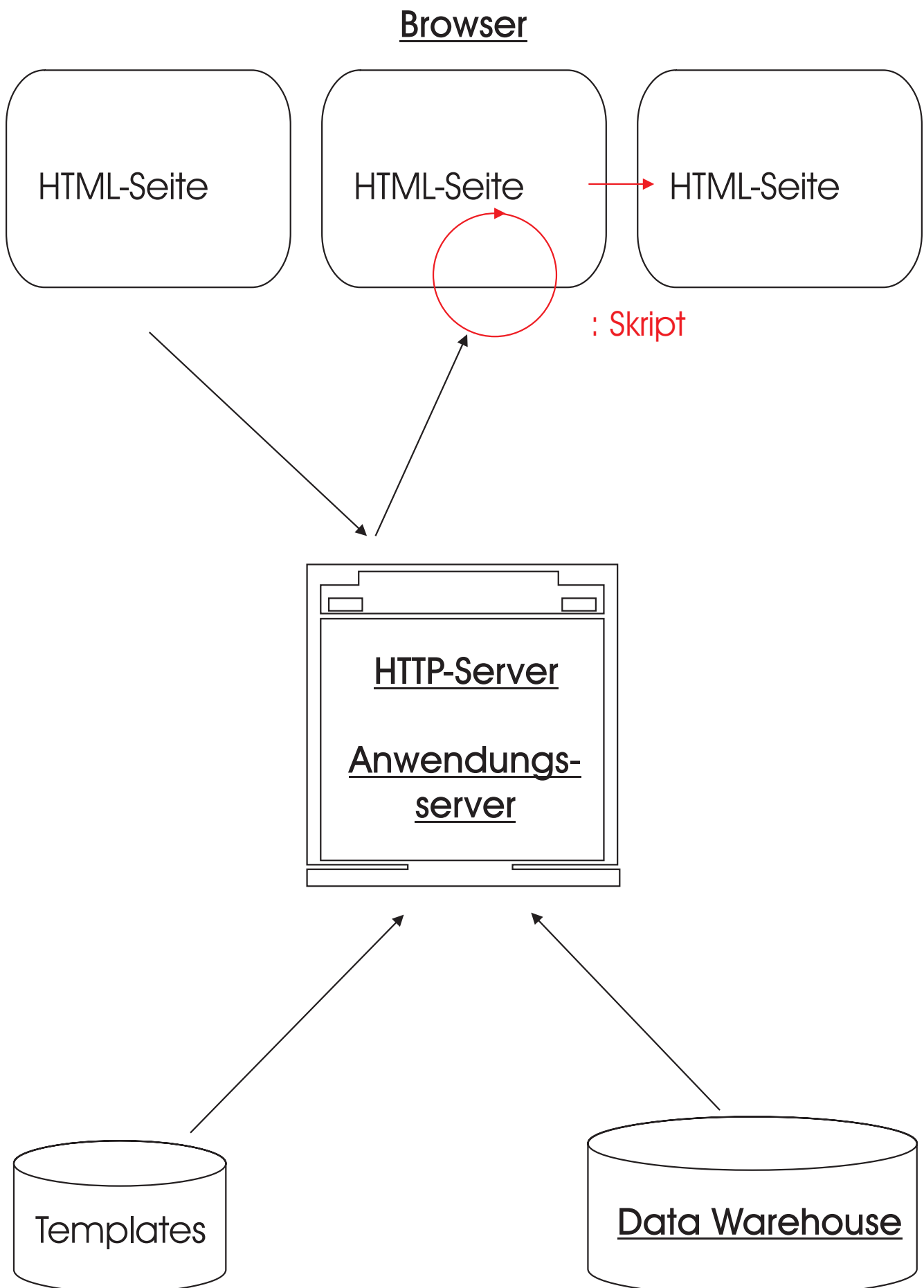
Grenzen von HTML

Ergänzungen von HTML machen  
Intranet-Warehousing erst möglich

## 5.78 Interaktivität mit Standard-HTML



## 5.79 Interaktivität mit HTML und Skriptsprachen



---

# Client- und serverseitige HTML-Ergänzungen

---

Client

Internet Browser akzeptiert ...

- Standard-HTML
- Skriptprozeduren (z.B. JavaScript, VBScript)
- ActiveX- und Applet-Komponenten

Webseiten

Webserver und Anwendungsserver akzeptieren ...

- Standard-HTML
- Skriptprozeduren
- ActiveX- und Applet-Komponenten
- CGI-, ISAPI- oder NSAPI-konforme Programme

Server



---

# Steuerelemente in HTML integrieren

---

## Steuerelemente (controls)

mit vordefinierten Ereignissen

Bsp. *Edit Box*, *Schaltfläche*

## Werkzeuge

zur Integration von Web-Steuerelementen

- ① *Standard HTML* zur Verarbeitung *einfacher* Ereignisse
  - ⇒ HTML-Befehle einfügen
- ② *Skriptsprachen* zur Verarbeitung *komplexer* Ereignisse
  - ⇒ Skriptbefehle an Client oder Server in Webseite einbetten
  - Bsp. JavaScript, VBScript (Teilmenge von Visual Basic)
- ③ *Sprachen* zur *anspruchsvollen* Programmierung
  - ⇒ Programm als Modul zur Verfügung stellen
  - Bsp. Java-Programm als Applet zur Verfügung stellen
  - Bsp. DLL<sup>1</sup> als ActiveX-Komponente zur Verfügung stellen

---

<sup>1</sup> Dynamic Link Library

---

## VBScript in eine HTML-Seite einbetten

---

```
<SCRIPT LANGUAGE="VBScript">  
  'Initialisierung  
  ...  
  'Prozedur mit Steuerelement MsgBox  
Sub Hallo_Klick()  
  MsgBox "Hallo!"  
End Sub  
  ...  
</SCRIPT>
```

---

## ActiveX aus einer HTML-Seite aufrufen

---

Aufruf einer ActiveX-DLL

**<OBJECT**

**ID**="SteuerelementXY"

**CLASSID**="clsid:123-1234-123-123-"

**CODEBASE**="http://MyServ/MyDLL.OCX">

**</OBJECT>**

Ausschnitt aus der ActiveX-DLL *MyDLL.OCX*

```
Sub antworte(Abfrage As String, Antwort as
String)
    'Abfrage interpretieren und verarbeiten
    ...
    Antwort =
        "Content-Type: text/html
        <HTML><BODY>
            'Antwort zeigen
            ' (d.h. folgendes Template instanzieren)
            ...
        </BODY></HTML>"
End Sub
```

---

# Serverprogramme ergänzen HTML

---

Webserver-Programme ...

ergänzen HTML, z.B. bei der ....

- Interpretation von Datenbankabfragen

- Indizierung von Webseiten

steuern den Webserver über Schnittstellen, z.B. ...

- CGI** (Common Gateway Interface für Serverprogramme)

  - portabel

  - weniger effizient (Aufspannen eines neuen Prozesses)

- ISAPI** (Internet Server Application Programming Interface)

  - proprietär (Microsoft)

  - effizienter

sind geschrieben in ...

- einer Skriptsprache wie JavaScript oder VBScript

- einer konventionellen Programmiersprache wie C

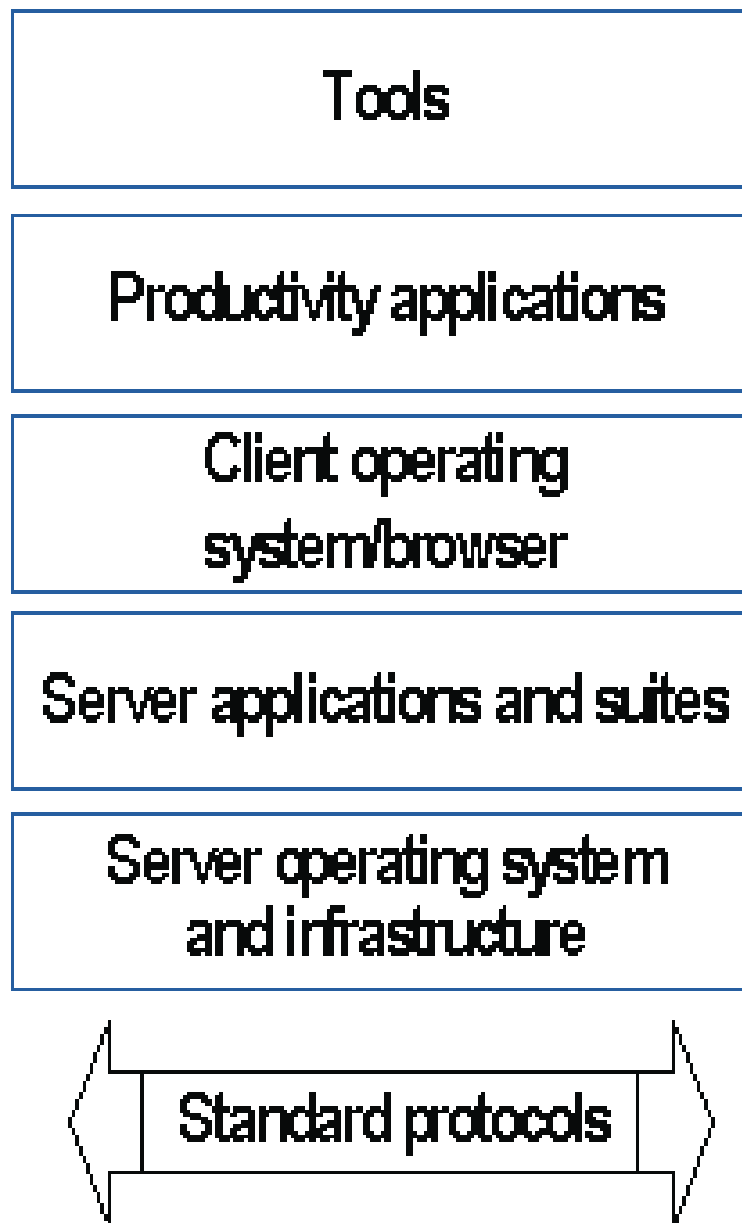
# Verbesserungen des HTML-Konzepts

	<i>HTML ohne Scripts</i>	<i>Neue Konzepte</i>
<i>Sprachumfang</i>	vordefiniert	erweiterbar
<i>Syntaxkontrolle</i>	schwach	streng
<i>Strukturierbarkeit</i>		
<i>Hyperlink-Adressen</i>	lokal und absolut	auch zentral und symbolisch
<i>Sonderzeichen</i>	schlecht unterstützt	gut unterstützt
<i>Trennung von Format und Inhalt</i>		
<i>Formatierung</i>	aufwendig und fix	Stylesheets
<i>Suche</i>	ungenau/aufwendig, weil Inhalt schlecht beschreibbar	direkter, weil Inhalt gut beschreibbar
<i>Datenaustausch</i>	Tags formatbezogen	Tags inhaltsbezogen (Datenfelder)
<i>Änderbarkeit von Seiten durch den Client</i>	Format und Inhalt statisch → Netzverkehr gross	Format und Inhalt dynamisch → Netzverkehr kleiner
<i>Mehrfachverwendung (z.B. CD-ROM, Druck)</i>	Reformatierung aufwendig	Stammdokument mit Stylesheets
<i>Objektorientierung</i>		

## Beispiele von Neuentwicklungen

XML (eXtensible Markup Language),  
DHTML (Dynamic HTML), DOM (Document Object Model)

## Aufgabe Softwareklassen auf dem Web

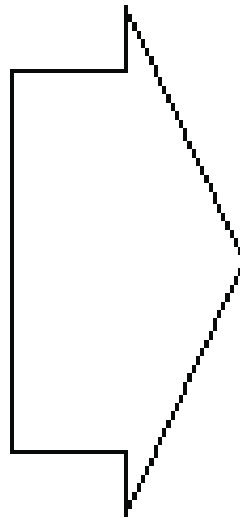


*Ordnen Sie die Software der richtigen Protokollebene zu*

- a) MS Windows NT Server
- b) TCP/IP
- c) MS Exchange Server
- d) MS Internet Explorer
- e) MS SQL Server
- f) MS Office
- g) ActiveX - Steuerelemente

## Content Publishing

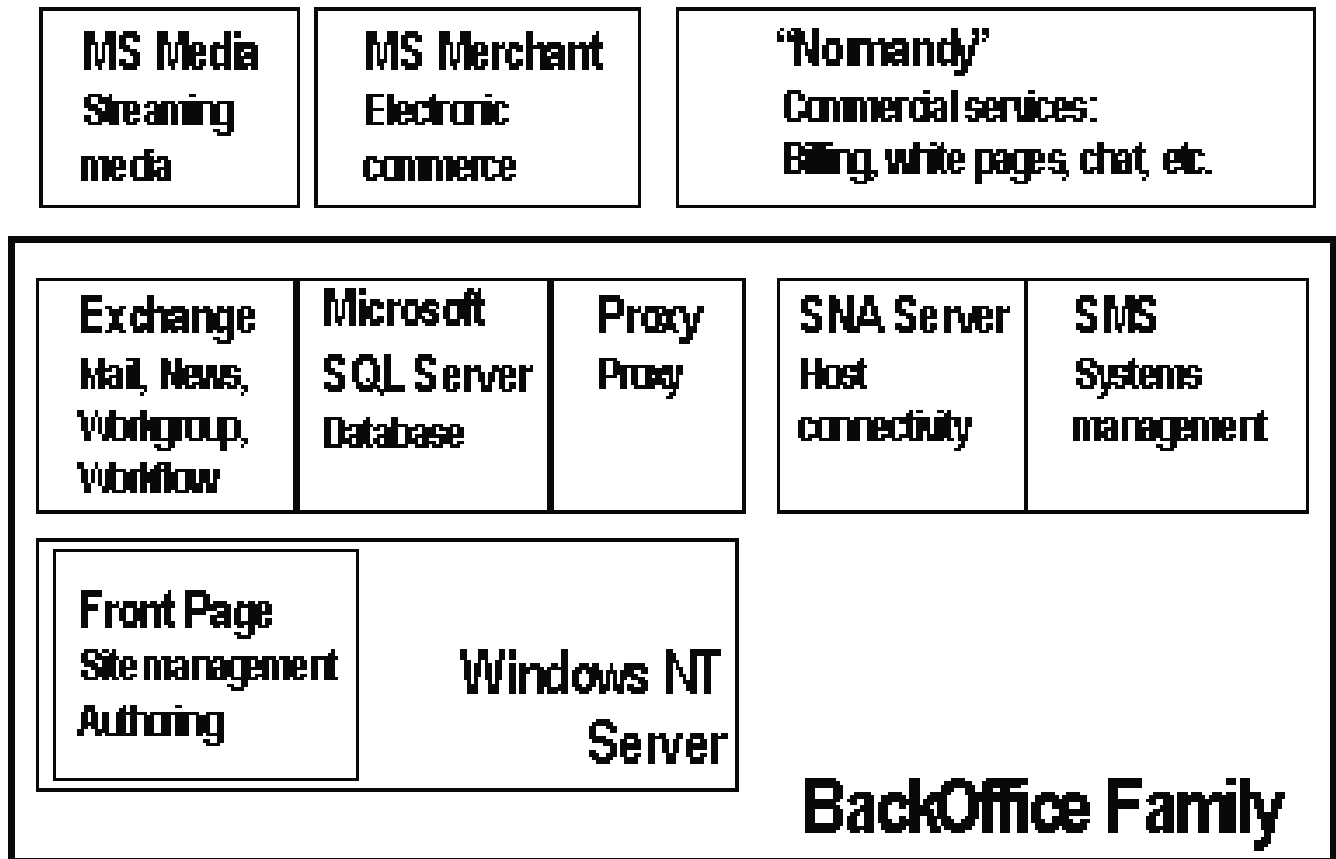
- HR handbook
- Real estate advertising
- Events calendar
- Hard goods catalog
- Advertising-sponsored content
- Flight schedules
- Mutual fund advertising and prospectus



## Business Applications

- HR benefits management
- Appointment scheduling, mortgage applications
- Online ticketing
- Order capture and fulfillment
- Fee-based content (micropayments)
- Reservations and ticketing
- Portfolio management

# Inter- und Intranet - *MS BackOffice*

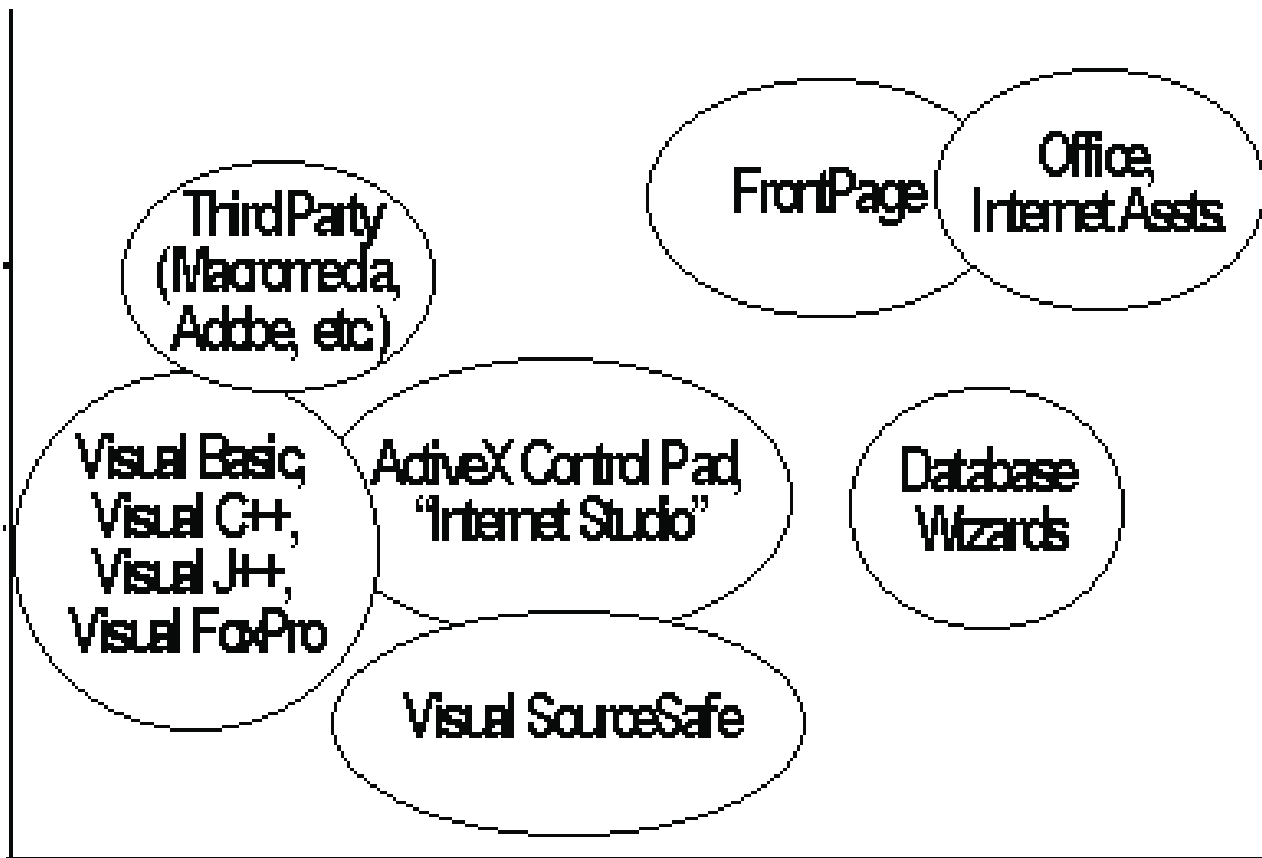


*Ordnen Sie zu*

- a) Sicheren Internetzugang gewährleisten
- b) Elektronische Post verwalten
- c) Webseiten erstellen verwalten
- d) Elektronischen Handel unterstützen
- e) Multimediale Internet-Anwendungen unterstützen
- f) Professionelle Internet-Verleger unterstützen
- g) IBM-Welt einbinden
- h) Netzbetriebssystem
- i) Server-Datenbanken verwalten
- k) Netzressourcen verwalten



# Entwicklungswerkzeuge



*Ordnen Sie zu*

- a) Endbenutzerwerkzeuge
- b) Programmierwerkzeuge
- c) Datenbankwerkzeuge
- d) Sicherheitsmanagement
- e) Rich Content ?
- f) ActiveX Control ?

# Entwicklungstechnologien von Microsoft

## ActiveX :=

- ✓ Binärcode-Modul ( DLL), das ...
- ✓ mit anderen Anwendungen “zusammengesteckt” wird und
- ✓ klein genug für die Übertragung auf dem Internet ist

Im Gegensatz zu einem Applet sind ActiveX-Komponenten Binärcode-Module und für die Microsoft-Betriebssysteme optimiert

### Aufgabenbeispiele

- Verhalten des Internet Browsers ergänzen
- Neues Steuerelement auf einer Webseite einsetzen
- Internet Browser aus einer anderen Anwendung manipulieren

## ISAPI (Internet Server Application Programming Interface) :=

- ✓ proprietäre
- ✓ API-Schnittstelle zwischen einem
- ✓ Microsoft-Webserver und einer
- ✓ Applikation, die auf einen Datenbankserver oder
- ✓ eine andere Serverapplikation zugreift

### Aufgabenbeispiel

- Server-Datenbank auf dem WWW publizieren

## 5.81 Mehrstufiger Zugriff auf Web DW

### Browser (Client)

liest ein Formular oder  
eine frei formulierte Abfrage

*Antwort in HTML*

### Web Server (HTTP Server)

erkennt im HTML-Strom die Abfrage

*Abfrage in HTML*

### Anwendungsserver

übersetzt die HTML- in eine SQL-Abfrage

*Abfrage in SQL*

### Data Warehouse Server

führt die Datenbankabfrage aus

*Ergebnis in einer Variable*

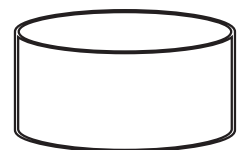
Anwendungsserver erstellt mit dem  
Abfrageergebnis ein HTML-Dokument

*Ergebnis in HTML*

Web Server leitet das HTML-Abfrageergebnis weiter

*Ergebnis in HTML*

Browser interpretiert das HTML-  
Dokument und zeigt es an



# Ein Template in erweitertem HTML

Fakten

Dimensionen

```
<HTML>
  <TITLE>Monatsverkäufe</TITLE>
  <BODY>
    <!--SQL
      SELECT PRODUKT.Name,
      SUM ( VERKAEUFE.$Umsatzart )
      FROM VERKAEUFE, PRODUKTE, FILIALEN, PERIODEN
      WHERE PERIODEN.Monat = VERKAEUFE.Monat
      GROUP BY PRODUKT.Name
      ORDER BY PRODUKT.Name
    -->
    <!--Formatierung des SQL-Ergebnisses in einer Tabelle>
    ...
  </BODY>
</HTML>
```

Der Webserver extrahiert die SQL-Abfrage aus der Webseite

Der Anwendungsserver interpretiert die SQL-Abfrage

Die Variable \$Umsatzart wird vom Anwendungsserver mit einem im Abfrageformular gewählten Wert - zum Beispiel *Geldumsatz* oder *Mengenumsatz* - instanziiert

Die Komplexität einer Abfrage kann von einer einfachen Menüabfrage bis zu jener von QBE reichen

## 5.81 HTML-Template aufrufen und instanzieren

Der Benutzer füllt ein Abfrageformular aus. Der **Browser** sendet die Benutzereinträge 1...N als Teil der folgenden URL an den Web Server: *http://<Web Site>/<Templatename> <Var1, Wert>...<VarN, Wert>*. Der Anwendungsserver wird später mit den Paaren <Var, Wert> ein SQL-Template instanzieren.

Der **Web Server** leitet die *URL* an den Anwendungsserver weiter

Der **Anwendungsserver** ...

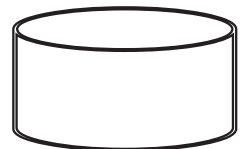
- a) isoliert die Paare <Var, Wert>
- b) sucht in der Metadatenbank das passende SQL-Template
- c) instanziiert dessen Variablen mit den entsprechenden Werten
- d) sendet die SQL-Abfrage an den Data Warehouse Server

Der **Data Warehouse-Server** führt die SQL-Abfrage aus und sendet das *Ergebnis* dem Anwendungsserver

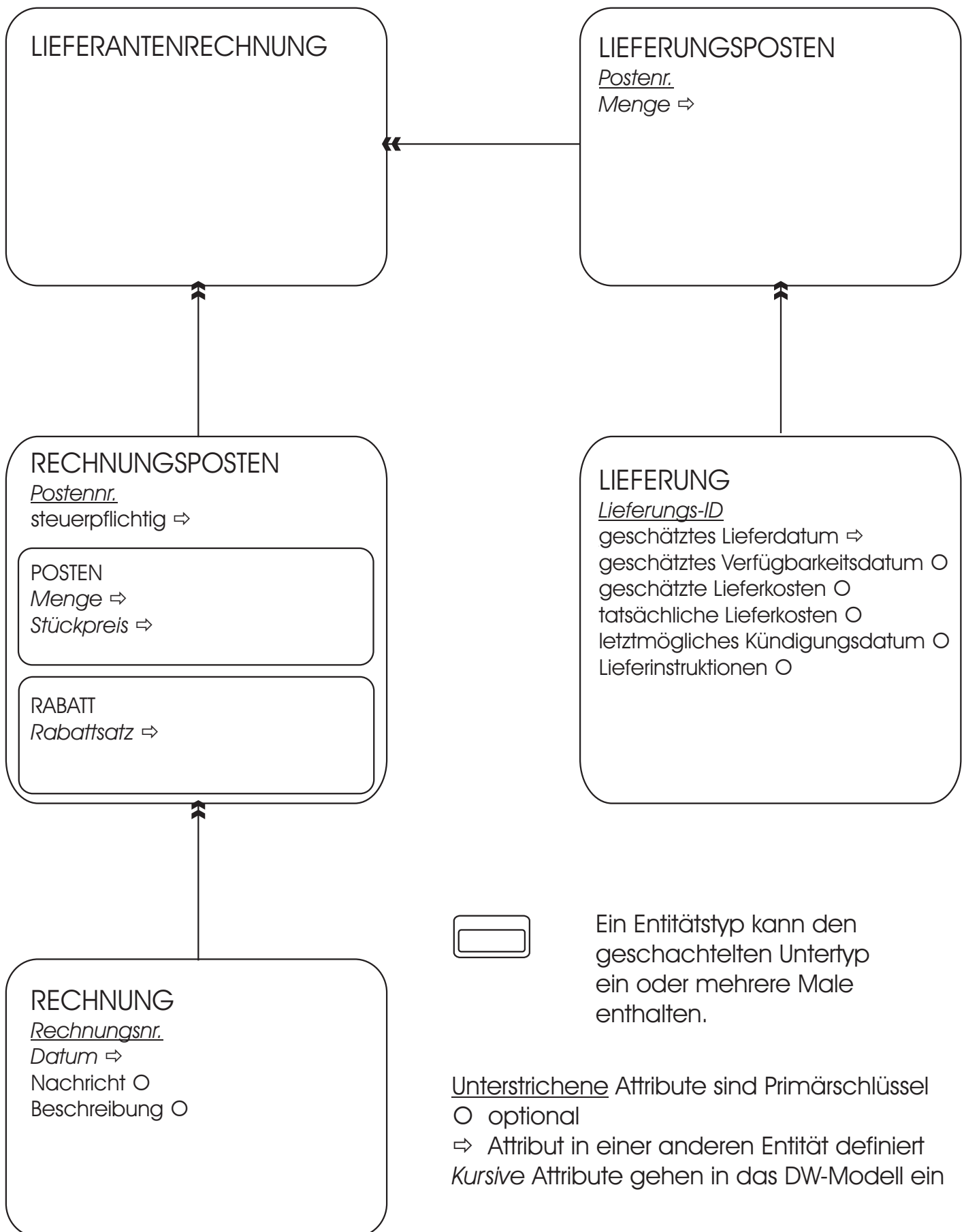
Der **Anwendungsserver** instanziiert das Template mit dem SQL-Ergebnis und sendet das *HTML-Ergebnis* an den Web Server

Der **Web Server** leitet das *HTML-Ergebnis* an den Browser des Benutzers weiter

Der **Browser** zeigt das *HTML-Ergebnis* meist in Form einer Tabelle an



# Internet-Adressen



## ① Produktunabhängige Information

<http://pwp.starnetinc.com/larryg/>

(The Data Warehousing Information Center)

White Papers, Literaturverweise, Software

<http://www.dw-institute.com> (Data Warehousing Institute)

Material aus der Sicht einer Beratungsunternehmung

<http://www.datamation.com/PlugIn/workbench/dwhouse/dwhouse.htm>

<http://www.datamation.com/PlugIn/workbench/dmarts/dmarts.htm>

Beiträge der Zeitschrift *Datamation*, insbesondere über Software

<http://altaplana.com/olap/> (Alta Plana Online Analytical Processing)

Überblick, insbesondere OLAP

## ② Produktübersichten

<http://www.elkinsanalytics.com/Olap/guide/>

(OLAP Software Selection Guide von S. Elkins)

<http://www.olapreport.com> (The OLAP Report)

umfangreiche Marktübersicht (Verweise auf Anbieter finden Sie auf den betreffenden Folien)

## ③ Produktunabhängige News Groups

<news:comp.databases.olap>

[Web Quiz](#)

# *Folienverzeichnis (Ein Klick führt zur gewünschten Folie)*

<u>Inhalt</u>	<u>2</u>
<u>Entwicklung</u>	<u>3</u>
<u>5.4 HANDEL - Eine Entwicklungsskizze</u>	<u>4</u>
<u>Data Warehouses verschiedener Grösse</u>	<u>5</u>
<u>Data Mart</u>	<u>6</u>
<u>Data Mart-Ziele</u>	<u>7</u>
<u>Enterprise Data Warehouse</u>	<u>8</u>
<u>5.6 Zentralisierte EDW-Architektur</u>	<u>9</u>
<u>5.7 Hierarchische EDW-Architektur (HEDW)</u>	<u>10</u>
<u>5.10 Koordinierte EDW-Architektur (KEDW)</u>	<u>11</u>
<u>Architekturen</u>	<u>12</u>
<u>Data Mart-Koordination - Voraussetzungen</u>	<u>13</u>
<u>5.9 Data Mart-Koordination - Abfragebeispiel</u>	<u>14</u>
<u>Anforderungen an Attribute</u>	<u>15</u>
<u>Aufgabe Anforderungen an Attribute</u>	<u>16</u>
<u>5.8 Data Mart vs. Enterprise Data Warehouse</u>	<u>17</u>
<u>5.5 Data Mart-Entwicklung</u>	<u>18</u>
<u>EDW und Data Mart - Ein Kontinuum</u>	<u>19</u>
<u>Beurteilung von Data Marts</u>	<u>20</u>
<u>5.12 Betriebsnähe durch Indikator u. Dimensionen</u>	<u>21</u>
<u>Würfel - Begriffe</u>	<u>22</u>
<u>Einen Indikator nach seinen Dimensionen messen</u>	<u>23</u>
<u>HANDEL - Fakten nach Dimensionen auswerten</u>	<u>24</u>



Indikator versus Dimensionen	25
Aufgabe <i>Fakt oder Dimension</i> (A 5.1)	26
Mehrdimensionale Daten speichern	27
Relationale versus mehrdimensionale DB	28
Anforderungen an RDBMS	29
Inhalt	34
5.15 Zugriff auf operative und analytische Daten	35
Datenbankabfragen	36
Abfragesprachen	37
5.16 Prozedurale und deklarative Abfragen	38
5.17 Query by Example (QBE)	39
SQL - Grobübersicht	40
5.18 SQL - Teilsprachen	41
SQL - Vorteile	42
Mängel von SQL	43
5.19 Abfragearten	44
Endbenutzerzugriff mit OLAP - Definition	45
5.20 OLAP versus OLTP	46
Anwendungen	47
Definitionselemente	48
📌 ANLAGEBERATUNG - Fallbeispiel	49
📌 Einfache Darstellung von Ergebnissen	50
Importierte und abgeleitete Daten	51
Zusammenfassungen und Vergleiche	52

<u>Zusammenfassung - Ein Beispiel</u>	53
<u>Zusammenfassungen können verfälschen</u>	54
<u>Browsing</u>	55
<u>5.21 † Filtern beschränkt die angezeigten Werte</u>	56
<u>Drilling Up and Down</u>	57
<u>5.21 † Drilling Down - Ausgangslage</u>	58
<u>5.22 † Drilling Down - Ergebnis</u>	59
<u>† Drilling navigiert in Kategorienhierarchien</u>	60
<u>Dimensionen mit Kategorien beschreiben</u>	61
<u>Drilling Down kann verfälschen</u>	62
<u>5.23 † Slicing und Dicing analysiert / synthetisiert</u>	63
<u>† Pivoting tauscht Zeilen und Spalten</u>	64
<u>5.24 † Eine Ebene ist eine Zusatzdimension</u>	65
<u>5.25 † Die Visualisierung hebt hervor</u>	66
<u>OLAP-Arten - Definitionen</u>	67
<u>OLAP-Arten - Vergleich</u>	68
<u>5.28 OLAP-Architekturen</u>	69
<u>5.29 OLAP-Architekturen und Produkte</u>	70
<u>DOLAP-Frontend Cognos <i>PowerPlay</i></u>	71
<u>5.30 Entwicklung eines DOLAP-Würfels</u>	72
<u>ANLAGEBERATUNG mit <i>Cognos PowerPlay</i> (5.2)</u>	73
<u>Funktionalität von <i>PowerPlay</i></u>	77
<u>Ausgewählte Kriterien für OLAP-Clients</u>	78
<u>② Ausgewählte Kriterien für OLAP-Server</u>	80

OLAP-Produkte und Hersteller	82
5.31 Methode im Vergleich	83
Inhalt	84
Modellierung von Informationssystemen	85
Operative und analytische Datenmodelle	86
Unwünschte und wünschbare Redundanz	87
Data Warehouses führen zu Redundanz	88
Ziele und Mittel der Datenmodellierung	89
5.32 Ein operatives Unternehmungsdatenmodell	90
5.34 Ein operatives Bereichsmodell <i>Fakturierung</i>	91
Data Warehouse-Modellierung	92
Transformationen des operativen Modells	93
5.35 Operatives Bereichs-      partielles DW-Modell	94
5.36 Vom partiellen DW-Modell zum EDW-Modell	95
5.37 Logische Modellierung im Vergleich	96
5.38 Sternschemata für relationale Data Marts	97
HANDEL - Betriebliche Anforderungen sammeln	98
5.40/5.41 Anforderungsdiagramm	99
5.42 Ein Sternschema für HANDEL	100
5.43 Mehrere Faktzeilen pro Dimensionszeile	101
Sternschema-Tabellen für HANDEL	102
5.44 Sternschema allgemein	103
5.45 Sternschemata filtern Abfragen ...	104
LIEFERFRIST - Sternschema (A 5.3)	105

<u>VERKAUF - EDW-Modell und Sternschema (A 5.4)</u>	106
<u>Relationales OLAP erfordert Sternschemata</u>	108
<u>OLAP-Abfragen auf einem Sternschema</u>	109
<u>Eine SQL-Abfrage auf einem Sternschema</u>	110
<u>5.48 Sternschemata durch Denormalisierung</u>	111
<u>Stern- versus normalisiertes Schema I</u>	112
<u>Stern- versus normalisierte Schemata II</u>	113
<u>Vom normalisierten zum Sternschema</u>	114
<u>Erweiterte Sternschemata</u>	115
<u>Beurteilung erweiterter Sternschemata</u>	116
<u>5.49 Erweitertes Sternschema - Normalisierung</u>	117
<u>Erweitertes Sternschema - m:n-Beziehung (A 5.6)</u>	118
<u>KONTO - Auflösung einer m:n-Beziehung (A 5.7)</u>	119
<u>Metadaten</u>	120
<u>5.52 Einige Klassen von Metadaten</u>	121
<u>Vorkommen von Metadaten</u>	122
<u>Data Dictionary</u>	123
<u>Data Dictionary-System</u>	124
<u>5.53 Data Dictionary-System</u>	125
<u>Anforderungen an DD-Systeme</u>	126
<u>Beispiele von DD-Systemen</u>	127
<u>ROLAP-Frontend <i>if...Synchrony</i></u>	128
<u>Architektur von <i>Synchrony</i></u>	129
<u>Wichtige Metadaten von <i>Synchrony</i></u>	130

Funktionalität von <i>Synchrony</i>	131
EINZELHANDEL mit <i>if...Synchrony</i> (A 5.8)	132
Inhalt	136
Von Produktionsdatenbanken zum EDW	137
Data Warehouse-Entwicklungszyklus	138
Phasenunabhängige Tätigkeiten	139
5.54 Data Warehouse-Ziele	140
Endbenutzerbefragung - Ablauf	141
Endbenutzerbefragung - Inhalte	142
Spezifikationsdokument	143
Entwicklungsprobleme	144
Rollen der DW-Entwicklung und -Verwaltung	145
Rollen der DW-Entwicklung (A 5.8)	146
Software für Data Warehouses	147
Eine integrierte Softwarelösung	148
Implementation und Betrieb	149
5.58 Operative Daten laden	150
ZEITSCHRIFTEN - A Fallbeispiel	151
B Daten auswählen	152
C Daten transformieren	153
D Daten transformieren und analysieren	154
Funktionalität von Ladeoperationen	155
Beispiel Validitätsprüfungen	157
Ladesoftware	158

<u>Einige Ladeprodukte</u>	159
<u>Physische Modellierung</u>	160
<u>Performanceprobleme</u>	161
<u>Performanceproblem Aggregation</u>	162
<u>Performanceproblem Sternschema</u>	163
<u>5.59 Performanceproblem Mehrdimensionalität I</u>	164
<u>Performanceproblem Mehrdimensionalität II</u>	165
<u>Data Warehouse-Performance</u>	166
<u>Datenmodell-Performance verbessern</u>	167
<u>Optimierungsmassnahme Denormalisierung</u>	168
<u>5.61 Normalisierung verlangsamt Abfragen</u>	169
<u>5.62 Denormalisierung beschleunigt Abfragen</u>	170
<u>Optimierungsmassnahme Partitionierung</u>	171
<u>Partitionierungsbeispiele</u>	172
<u>5.63 Spaltenpartitionierung</u>	173
<u>Optimierungsmassnahme Indexierung</u>	174
<u>5.64 Optimierungsmassnahme <i>Bitmuster</i></u>	175
<u>Komprimierte Bitmusterindizes</u>	176
<u>Wann Bitmuster-Indizes?</u>	177
<u>Kosten und Nutzen der Bitmuster-Indexierung</u>	178
<u>Performancedeterminanten Hardware</u>	179
<u>Hardwareperformance verbessern</u>	180
<u>Komponentenperformance verbessern</u>	181
<u>Optimierungsmassnahme Parallelisierung</u>	182

Data Warehouse-Parallelisierung	183
5.65 Beispiel eines Data Warehouse-MPS	184
Arten von Mehrprozessorsystemen	185
5.66 Symmetrisches MPS (SMP)	186
Eine <i>mittlere</i> SMP-Konfiguration (1998)	187
5.67 Clustering	188
5.68 Massiv paralleles MPS	189
5.70 NUMA (Non Uniform Memory Access)	190
5.69 <i>Symmetrische</i> vs. <i>massiv parallele</i> MPS	191
Parallelisierungsverantwortliche	192
Parallelisierungsvarianten	193
Datenparallelisierung	194
Partitionierung für die Datenparallelisierung	195
Partitionierungskriterien	196
Trade Offs einiger Optimierungsmassnahmen	197
Inhalt	198
5.71 Vom Host zum Client/Server-System	199
Rechnernetze	200
Rechnernetze teilen Ressourcen	201
Rechnernetze für Data Warehouses	202
Client/Server-Systeme	203
5.72 Alternative Modelle der Aufgabenteilung	204
5.73 Client/Server-Architekturen	205
Zweistufige Client/Server-Architekturen	206

5.74 Zweistufige Data Warehouse-Architektur	207
Dreistufige Client/Server-Architektur	208
5.75 Dreistufige Data Warehouse-Architektur	209
Dreistufige ROLAP- und MOLAP-Architektur	210
Clientzugriff auf Data Warehouses	211
Endbenutzerzugriff auf Data Warehouses	212
Endbenutzerzugriff über Intranet und Internet	213
Intranet für Data Warehouses	214
Exkurs - Das Internet verbindet Rechnernetze	215
Exkurs - WWW - Populärster Internet-Aspekt	216
Exkurs - Wichtige Dienste des Internet	217
Exkurs - Wer regiert das Internet?	218
Exkurs - Beurteilung des WWW	219
Exkurs - Endbenutzerwerkzeug <i>Browser</i>	220
Exkurs - Arbeitsplatz auf dem WWW einrichten	221
Exkurs - Mit <i>analogem</i> Modem auf das Internet	222
Exkurs - <i>Digitales</i> ISDN statt analoges Modem	223
Exkurs - Internetknoten fehlertolerant verbinden	224
Exkurs - Der Spediteur TCP / IP	225
Exkurs - Internetadressen	226
Exkurs - Webadressen und -inhalte versenden	227
Exkurs - Webseiten erstellen	228
Exkurs - Geschäft auf dem WWW gründen	229
Intranet - Ein Internet für die Unternehmung	230



<u>Exkurs - Betriebliche Webanwendungen</u>	<u>231</u>
<u>Exkurs - HTML als 1. Web-Beschreibungssprache</u>	<u>232</u>
<u>Exkurs - XML - Neuere Beschreibungssprache</u>	<u>233</u>
<u>5.77 Spezifikation/Verarbeitung von Webseiten</u>	<u>234</u>
<u>Exkurs - Hyperlinks in HTML darstellen</u>	<u>235</u>
<u>Datenzugriff über das Web - Überblick</u>	<u>236</u>
<u>Interaktive statt passive Webseiten</u>	<u>237</u>
<u>5.78 Interaktivität mit Standard-HTML</u>	<u>238</u>
<u>5.79 Interaktivität mit HTML und Skriptsprachen</u>	<u>239</u>
<u>Client- und serverseitige HTML-Ergänzungen</u>	<u>240</u>
<u>Steuerelemente in HTML integrieren</u>	<u>241</u>
<u><i>VBScript</i> in eine HTML-Seite einbetten</u>	<u>242</u>
<u><i>ActiveX</i> aus einer HTML-Seite aufrufen</u>	<u>243</u>
<u>Serverprogramme ergänzen HTML</u>	<u>244</u>
<u>Verbesserungen des HTML-Konzepts</u>	<u>245</u>
<u>Aufgabe Softwareklassen auf dem Web</u>	<u>246</u>
<u>Internettrends</u>	<u>247</u>
<u>Inter- und Intranet - <i>MS BackOffice</i></u>	<u>248</u>
<u>Entwicklungswerkzeuge</u>	<u>249</u>
<u>Entwicklungstechnologien von Microsoft</u>	<u>250</u>
<u>5.81 Mehrstufiger Zugriff auf Web DW</u>	<u>251</u>
<u>Ein Template in erweitertem HTML</u>	<u>252</u>
<u>5.81 HTML-Template aufrufen und instanzieren</u>	<u>253</u>
<u>Internet-Adressen</u>	<u>254</u>