Betriebswirtschaftslehre II
Vorlesung 6: Business Intelligence – Relationale Abbildung und ETL

Wintersemester 2018/19

Prof. Dr. Martin Schultz
martin.schultz@haw-hamburg.de

Agenda

1. Data Warehouse-Architektur
2. Relationales Modell für BI-Anwendungen
3. Aufbau des ETL-Prozesses

Inhalte der Vorlesung und Übung

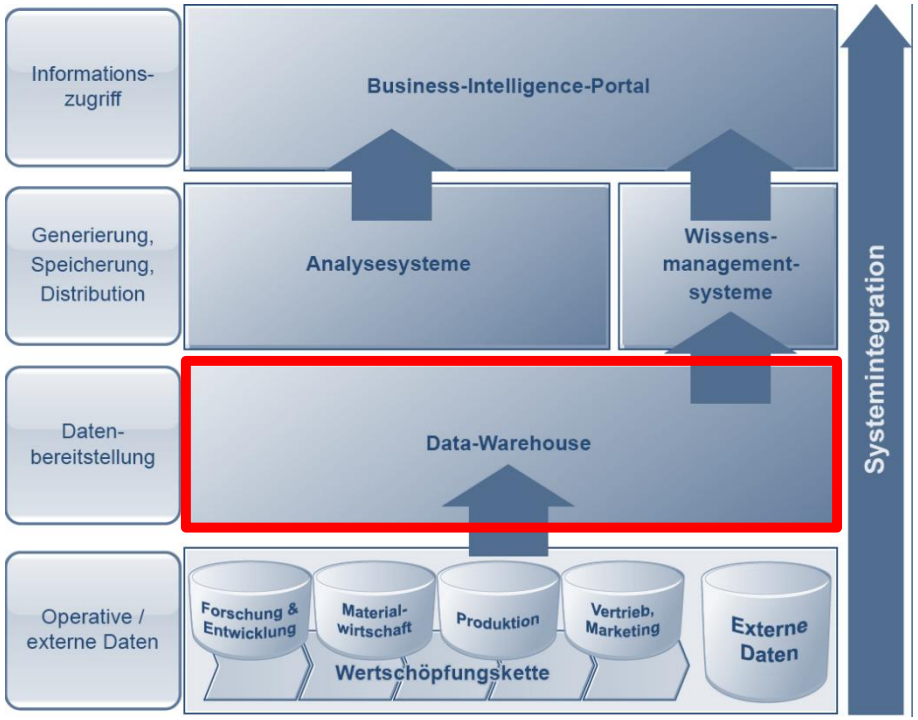
	Termin	Vorlesung	Übung
1	28.09.2018	Einführung und Grundlagen	-
2	05.10.2018	Geschäftsprozessmodellierung	Übung 1 – Gruppe 3/4
3	12.10.2018	Anwendungssysteme in Unternehmen	Übung 1 – Gruppe 1/2
4	19.10.2018	ERP-Systeme	Übung 2 – Gruppe 3/4
5	26.10.2018	ERP-Systeme: ReWe und Einführungsprojekte	Übung 2 – Gruppe 1/2
6	02.11.2018	Business Intelligence - OLAP	Übung 3 – Gruppe 3/4
7	**09.11.2018**	**Business Intelligence - ETL**	**Übung 3 – Gruppe 1/2**
8	16.11.2018	Business Intelligence – Dashboards/ Data Mining	Übung 4 – Gruppe 3/4
9	23.11.2018	Informationsmanagement	Übung 4 – Gruppe 1/2
10	30.11.2018	IT-Service-/ Enterprise Architecture-Management	Übung 5 – Gruppe 3/4
11	07.12.2018	IT-Governance/ IT-Compliance	Übung 5 – Gruppe 1/2
12	14.12.2018	Klausurvorbereitung	Übung 6 – Gruppe 3/4
	21.12.2018		Übung 6 – Gruppe 1/2
	11.01.2019		Übung 7 – Gruppe 1/2/3/4

Was sollen Sie mitnehmen…

- Relationale Abbildung des multidimensionalen Datenmodells
- Ziele und Aufbau des ETL-Prozesses erläutern können
- Aufgaben in den einzelnen Phasen des ETL-Prozesses beschreiben können

Datawarehouse - Definition

- **Datawarehouse:** Logisch zentraler Speicher zur Schaffung einer einheitlichen und konsistenten Datenbasis zur Unterstützung von Fach- und Führungskräften aller Bereiche und Ebenen
- Integration **unterschiedlicher Datenquellen** über **längere Zeiträume** (Wochen, Monat, Jahr, Jahrzehnte)



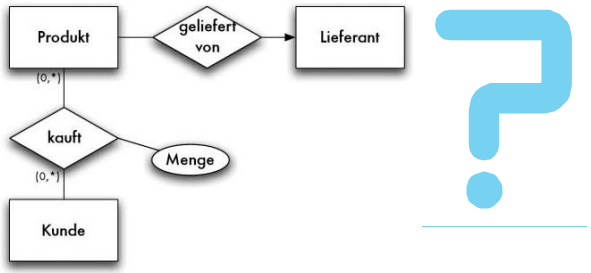
(Hansen 2009)

Beispiel-Szenario – Informationsbedarfe des Managements

Für weiterführende Fragen aus dem taktischen und strategischen Management sind die Datenstrukturen häufig ungeeignet

Typische Anfragen:

- Wie hat sich der Verkauf von Rotwein in den letzten 5 Jahren entwickelt?
  → **historische Daten notwendig (nicht flüchtige, zeitinvariante Daten)**
- Wie stehen wir im Vergleich zur Konkurrenz?
  → **externe Daten notwendig (Vereinheitlichung)**
- Wer sind unsere Top-Kunden?
  → **Filialübergreifende Daten notwendig (Themenorientierung)**
- Von welchem Lieferanten beziehen wir die meisten Kisten?



(Köppen 2014)

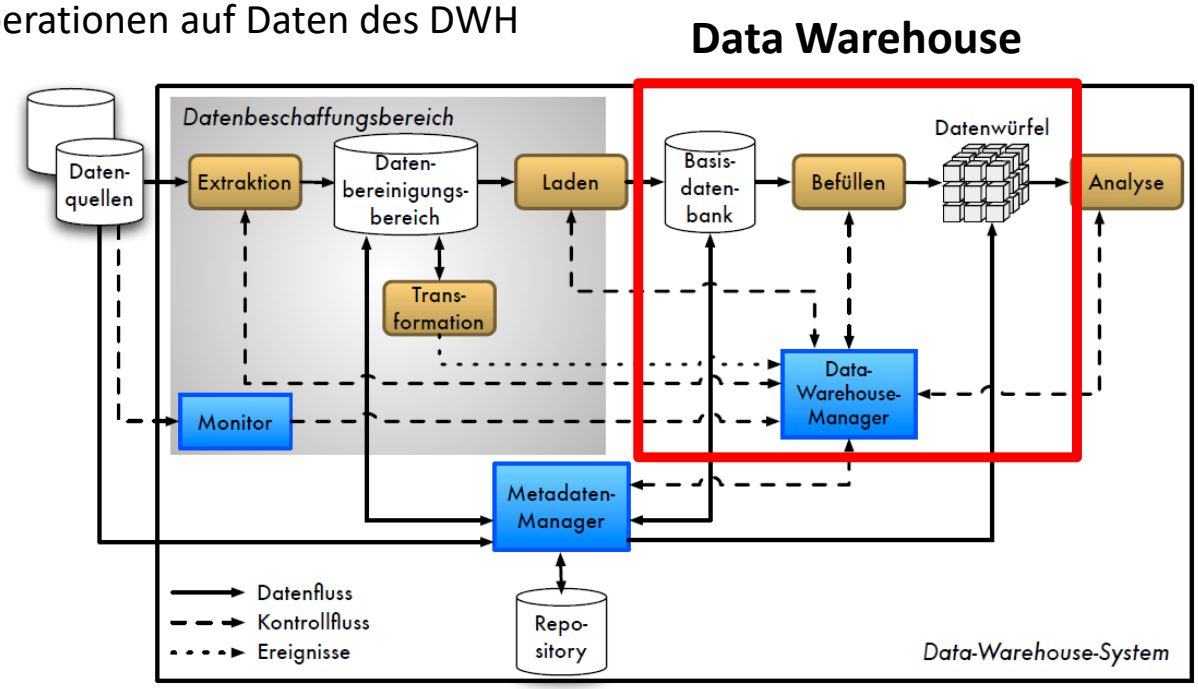
Data Warehouse - Eigenschaften

- „Datawarehouse is a **subject oriented**, **integrated**, **non-volatile** and **time-variant** collection of data in support of **management's decisions**" (Inmon 1996)
  1) ***Themenorientierung:*** Modellierung eines bestimmten Anwendungsziels und Konzentration auf inhaltliche Themenschwerpunkte, z. B. Produkte, Kunden. Die Datenbasis sollte unternehmensweit ausgerichtet sein und das Informationsbedürfnis verschiedenster Anwendergruppen bedienen.
  2) ***Vereinheitlichung***: Datenbasis enthalten zusammengeführte Daten, die aus unterschiedlichen Datenquellen (auch externen) stammen können. Ziel ist der Aufbau eines konsistenten Datenbestandes. Vereinheitlichung bezieht sich häufig auf Namensgebung, Bemaßung und Kodierung
  3) ***Nicht flüchtige Daten***: Einmal eingefügte Daten, bleiben langfristig erhalten und können nicht geändert werden
  4) ***Zeitvariante Datenbasis***: Alle Daten haben einen Zeitbezug. Dies ermöglicht Vergleiche über unterschiedliche Zeiträume
- „Ein Data Warehouse ist eine **physische Datenbank,** die eine integrierte Sicht auf beliebige Daten zu **Analysezwecken** ermöglicht." (Bauer und Günzel 2014)

Phasen des Data Warehousing

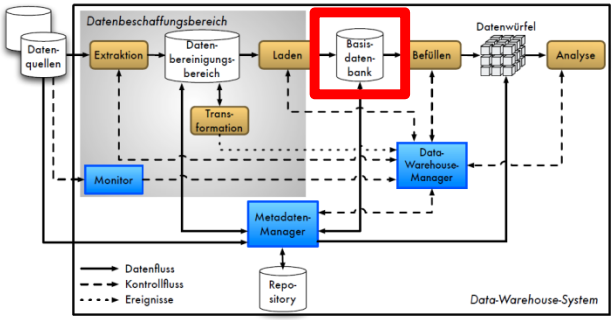
Data Warehousing: Prozess der Bereitstellung der relevanten Daten im Data-Warehouse

- **Überwachung (Monitoring)** der Quellen auf Änderungen durch Monitore
- Kopieren der relevanten Daten mittels **Extraktion** in temporären Datenbereinigungsbereich
- **Transformation** der Daten im Datenbereinigungsbereich (Bereinigung, Integration)
- **Laden** der Daten in die integrierte Basisdatenbank als Grundlage für verschiedene Analysen
- **Befüllen** der Datenwürfel (Datenbanken für Analysezwecke)
- **Analyse**: Operationen auf Daten des DWH



(Köppen 2014)

Basisdatenbank



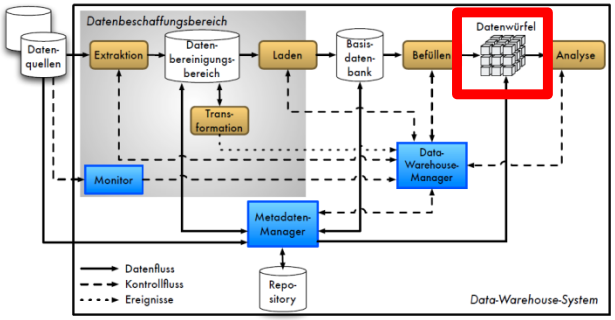
Integrierte Datenbasis für verschiedene Analysen

- unabhängig von konkreten Analysen, d.h. noch keine Aggregationen
- Versorgung der Datenwürfel mit bereinigten und integrierten Daten (integriert aus verschiedenen Datenquellen → vereinheitlichtes Datenmodell)

Anmerkungen

- Wird in der Praxis oft weggelassen
- SAP BW stellt Funktionen bereit, zur Modellierung einer Basisdatenbank in der unternehmensspezifischen DWH-Architektur

Datenwürfel (data cube)



Aufgabe: Datenbanken für Analysezwecke (relational oder multidimensional)

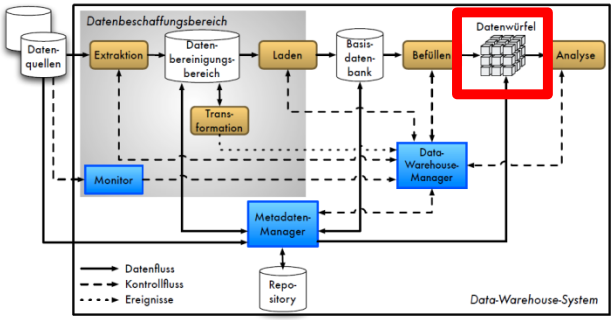
Aufbau:

- Orientieren sich in Struktur an Analysebedürfnissen
- Basis ist ein DBMS

Besonderheiten:

- Unterstützung des Ladeprozesses
  - Schnelles Laden großer Datenmengen
- Unterstützung des Analyseprozesses
  - Multidimensionales Datenmodell
  - Effiziente Anfrageverarbeitung (Indexstrukturen, Caching)

Data Marts

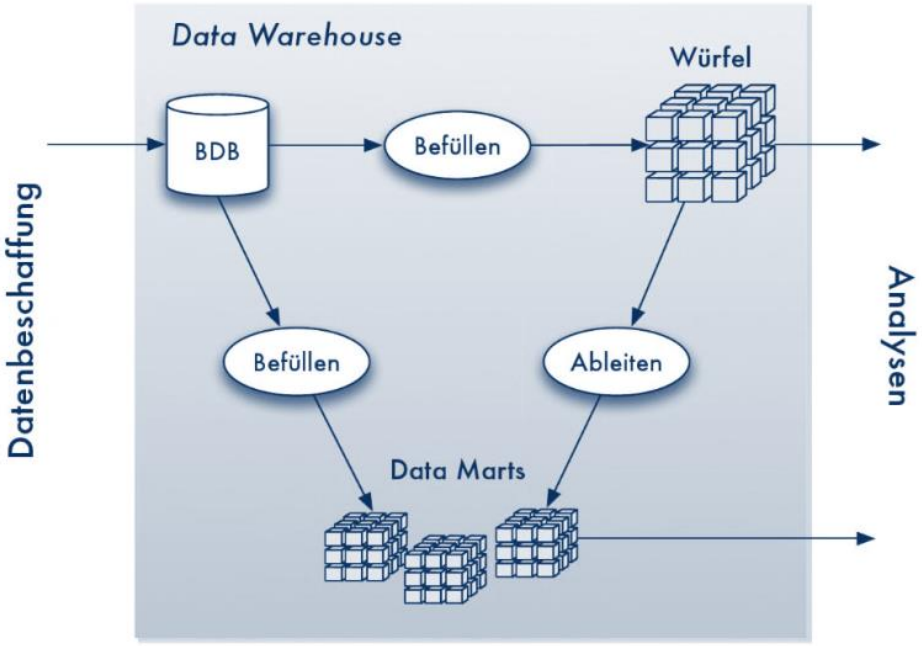


Personen-, anwendungs-, funktionsbereichs- oder problemspezifische Segmente des zentralen Data Warehouse-Datenbestandes (Kemper 2004)

Aufgabe: Bereitstellung einer inhaltlich beschränkten Sicht auf das DWH (z.B. für Abteilung)

Gründe: Eigenständigkeit, Datenschutz, Lastverteilung, Reduzierung des zu betrachtenden Datenvolumen, etc.

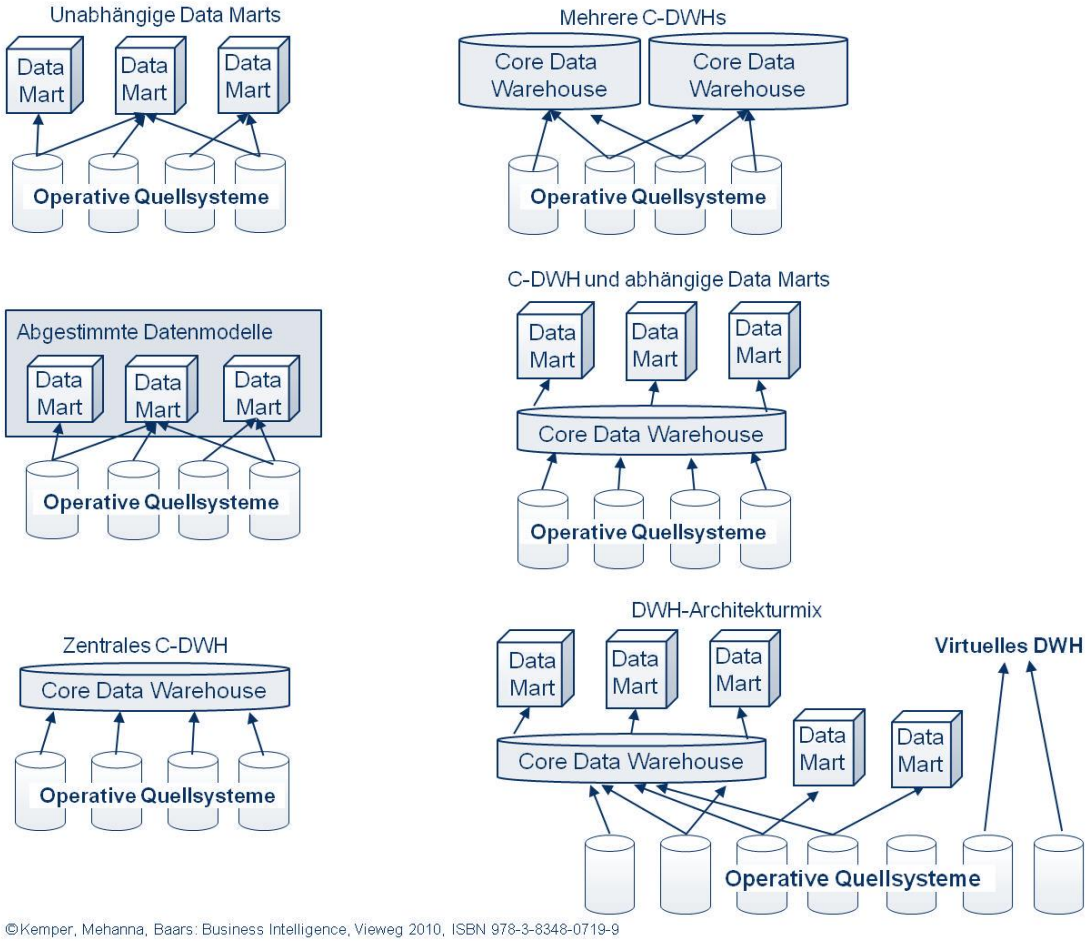
Realisierung: Verteilung der DWH-Daten aus der Basisdatenbank



(Köppen 2014)

Data Warehouse/ Data Marts - Architekturvarianten

- Grundsätzlich sind viele Kombinationen der DWH-Systemkomponenten (Core-DWH, Data Mart) möglich und werden in der Praxis aus angewendet



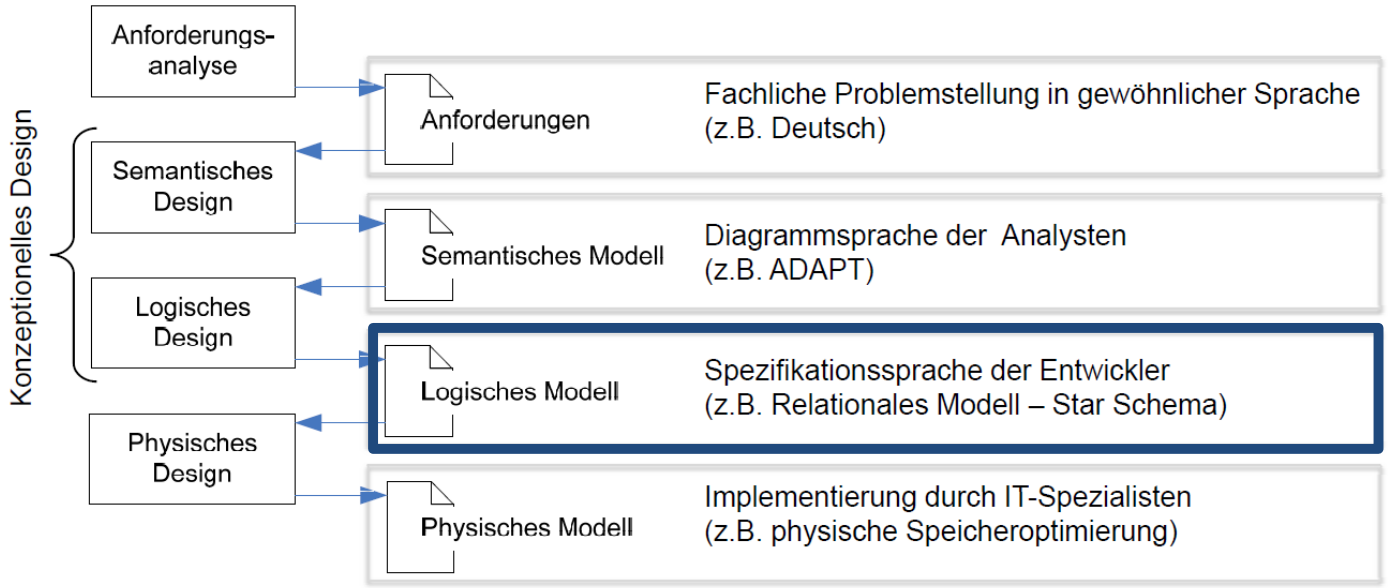
©Kemper, Mehanna, Baars: Business Intelligence, Vieweg 2010, ISBN 978-3-8348-0719-9

Ansätze für die Umsetzung

Zur Umsetzung des multidimensionalen Datenmodells in einem DBMS muss das Datenmodell auf das logische Schema des DBMS abgebildet werden

Aspekte bei der Auswahl:

- Art der logischen und physischen Speicherung
- Effiziente Anfrageformulierung bzw. –ausführung

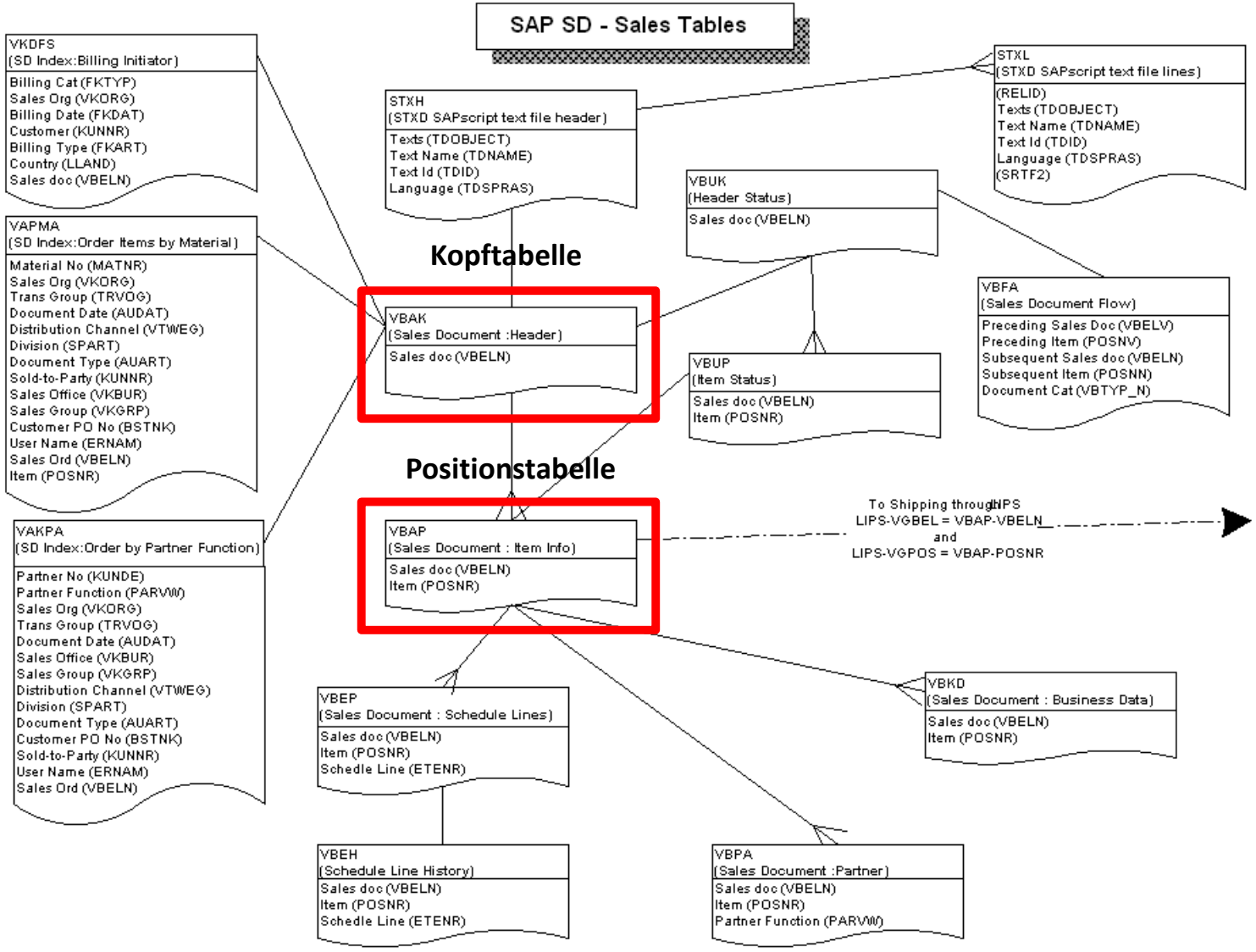


Relationale Abbildung (ROLAP)

Folgende Punkte sind bei Abbildung auf ein relationales Datenmodell zu beachten:

- Vermeidung des Verlustes **anwendungsbezogener Semantik** aus dem multidimensionalen Modell, z.B. Klassifikationshierarchien
- **Effiziente Übersetzung multidimensionaler Anfragen** in relationale Anfragen (SQL)
- **Effiziente Verarbeitung** der übersetzten Anfragen
- **Einfache Pflege** der entstandenen Relationen (z.B. Laden neuer Daten)
- **Berücksichtigung der Anfragecharakteristik** (z.B. überwiegend Leseoperationen) und des Datenvolumens von Analyseanwendungen

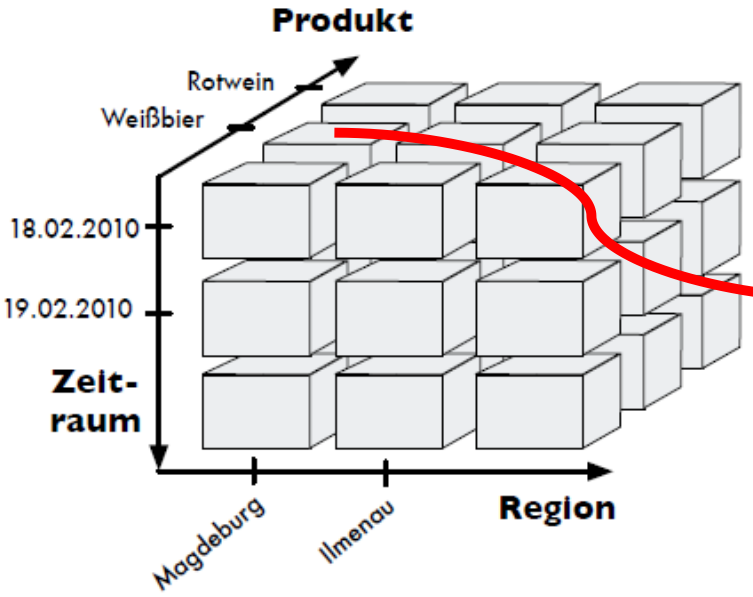
Beispiel Datenmodell SAP ERP – Vertrieb (Modul Sales & Distribution)



Relationale Abbildung - Faktentabelle

Ausgangspunkt: Umsetzung des Datenwürfels ohne Klassifikationshierarchien

- Dimensionen (genauer das jeweilige Primärattribut) und die Fakten/ Kennzahlen bilden eine Relation (Tabelle) → Spalten der Relation **= Faktentabelle**
- Jede Zelle des Datenwürfel entspricht einem Tupel in der Faktentabelle



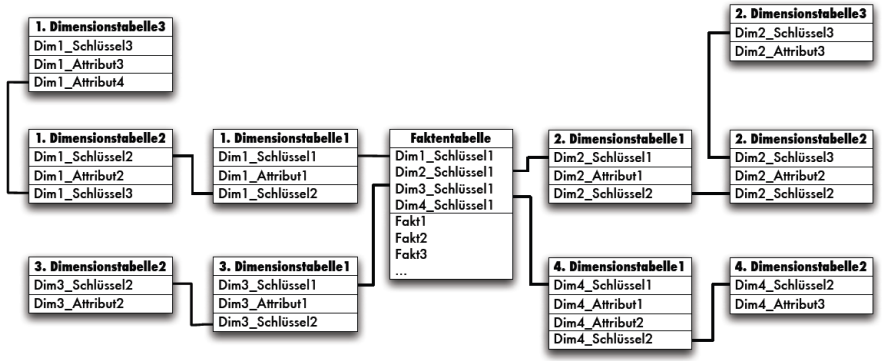
Produkt	Filiale	Tag	Verk.
Rotwein	Magdeburg	18.02.10	145
Weißbier	Magdeburg	18.02.10	267
Rotwein	Ilmenau	18.02.10	70
...			

(Köppen 2014)

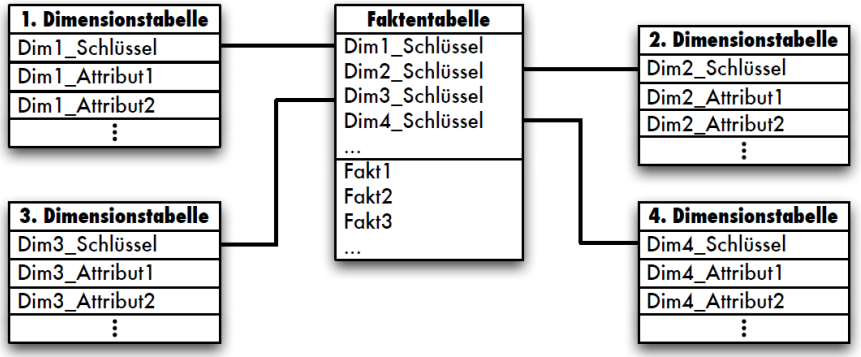
Relationale Abbildung – Dimensionen/ Klassifikationshierarchie

- Für die Abbildung einer Dimension mit Klassifikationshierarchie in einem relationalen Datenmodell gibt es zwei grundsätzliche Varianten

Snowflake-Schema

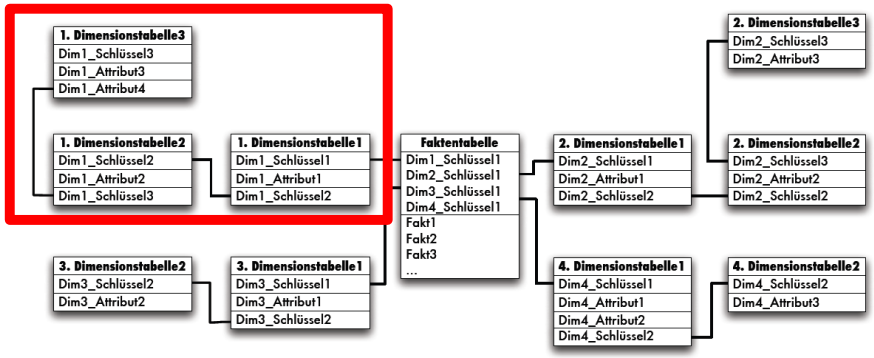


Star-Schema



Snowflake-Schema

Abbildung von Klassifikationshierarchien:
eigene Tabelle für jede Klassifikationsstufe
(z.B. Produkt → Produktgruppe, etc.)



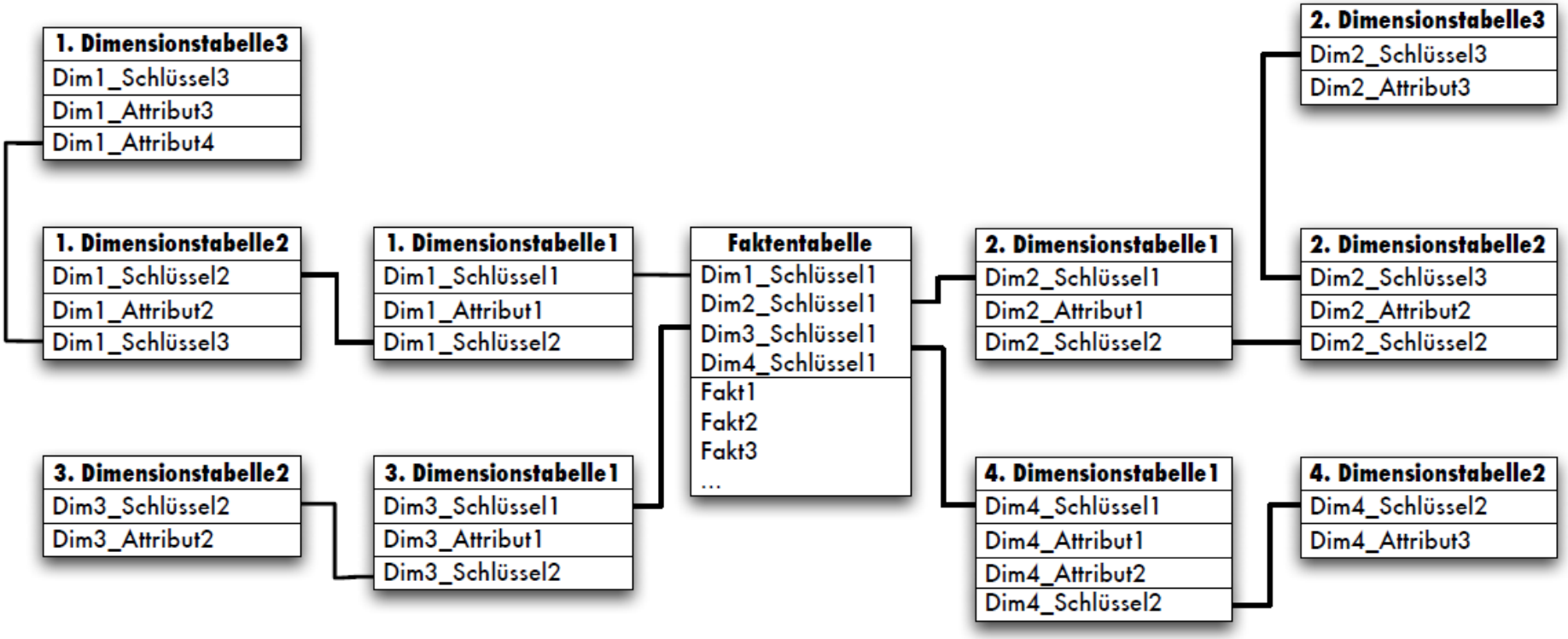
Die **Dimensionstabellen** enthalten

- ID für Klassifikationsattribut
- Beschreibende (dimensionale) Attribute
  (z.B. Marke, Hersteller, Bezeichnung)
- Fremdschlüssel zur Tabelle der direkt übergeordneten Klassifikationsstufe

Faktentabelle enthält:

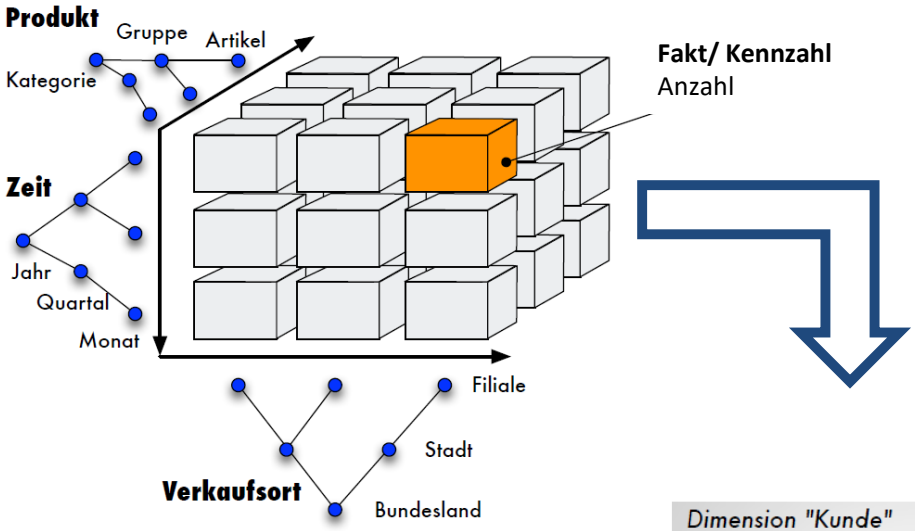
- Fakten und Kennzahlen
- Fremdschlüssel zur Tabelle der je Dimension niedrigsten Klassifikationsstufe (Primärattribut)
- Alle Fremdschlüssel der Dimensionstabellen bilden den zusammengesetzten Primärschlüssel der Faktentabelle

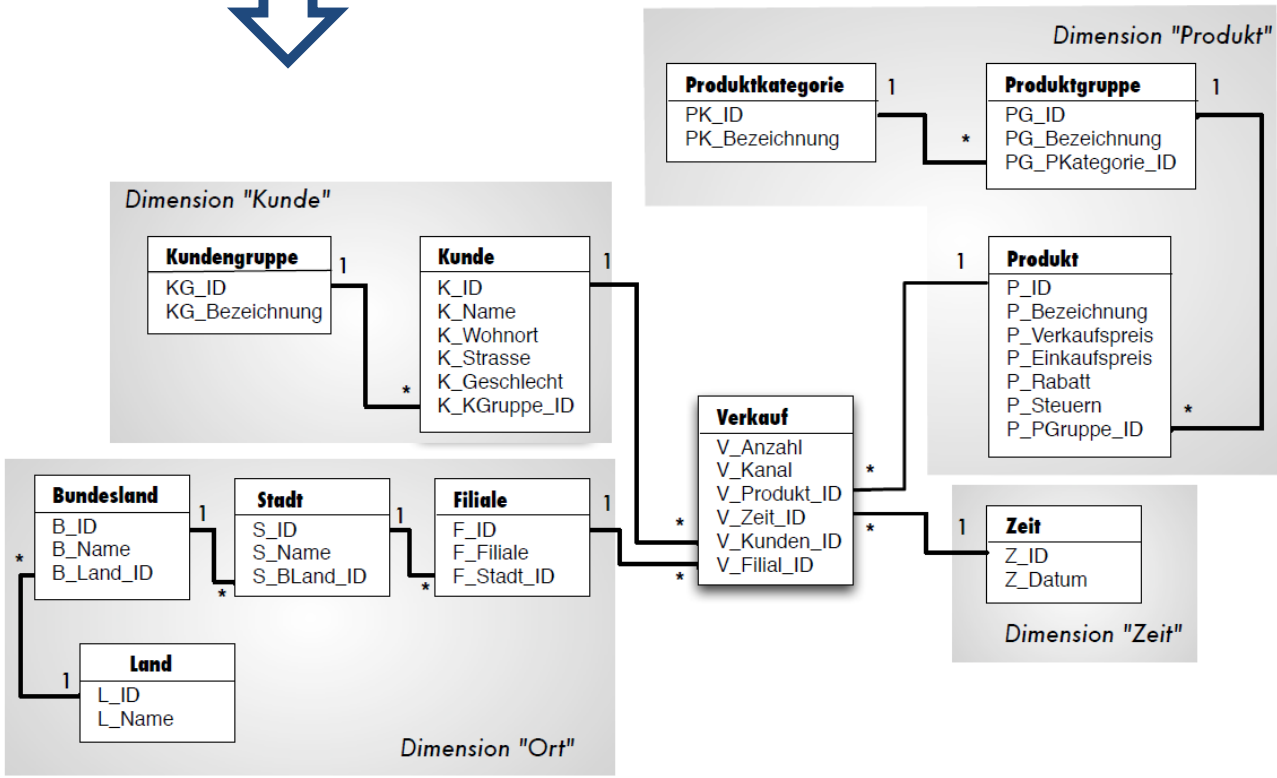
Snowflake-Schema: Muster



(Köppen 2014)

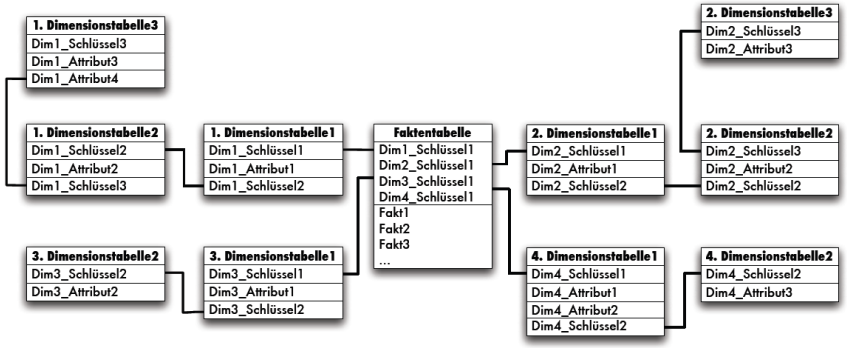
Snowflake-Schema: Beispiel





Snowflake-Schema: Datenbankschema

- Die transitiven Abhängigkeiten zwischen Klassifikationsattributen innerhalb einer Dimension verletzt die 3. Normalform
  → Zerlegung notwendig
- Abbildung einer Klassifikationshierarchie auf mehrere über Fremdschlüssel verbundene Tabellen entspricht der **Normalisierung** des relationalen Datenbankentwurfs

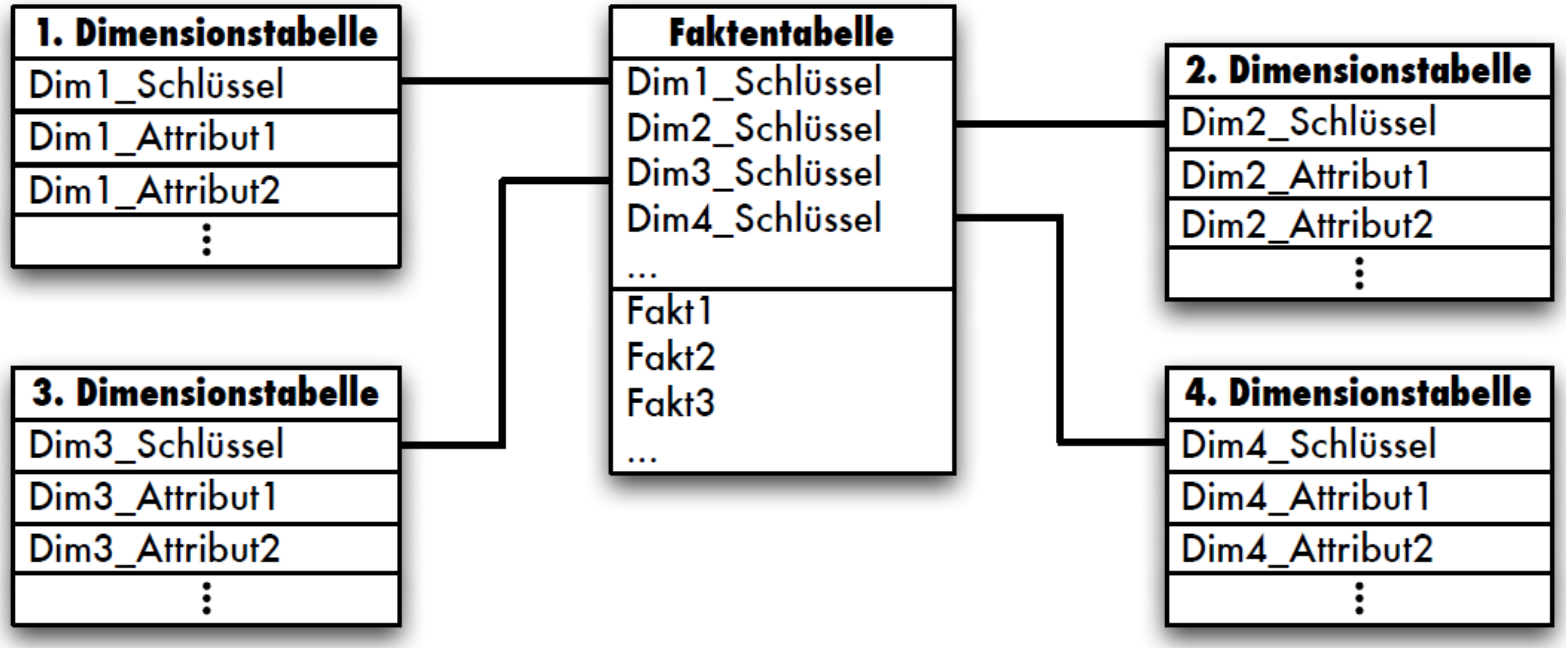


- Snowflake-Schema ist normalisiert in 3. Normalform:
  - Vermeidung von Update-Anomalien
  - Aber: erfordert Join über mehrere Tabellen

- **Besonderheit der Zeitdimension:** Da in einem Datumswert bereits alle Informationen wie Tag, Monat, Jahr enthalten sind und weitere Stufen wie Kalenderwoche berechnet werden können ist eine explizite Modellierung der Hierarchie/ Klassifikationsstufen meist nicht notwendig

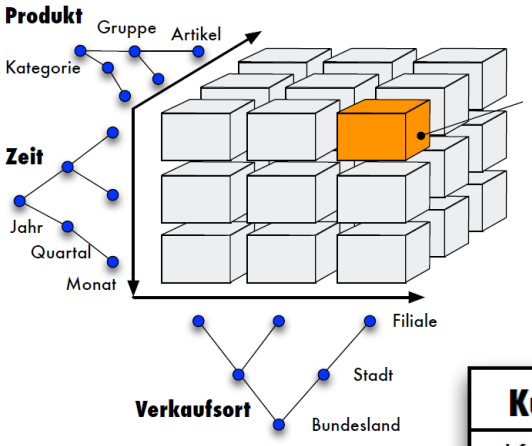
Star-Schema

- Denormalisierung der zu einer Dimension gehörenden Tabellen → 1. NF
- Für jede Dimension genau eine Dimensionstabelle
- Redundanzen in der Dimensionstabelle für schnellere Anfragebearbeitung
- *Beispiel:* Produkt, -gruppe, -kategorie als Spalten in einer Tabelle Produkt

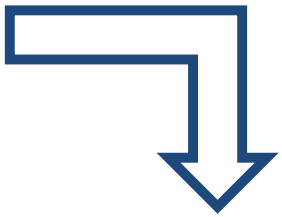


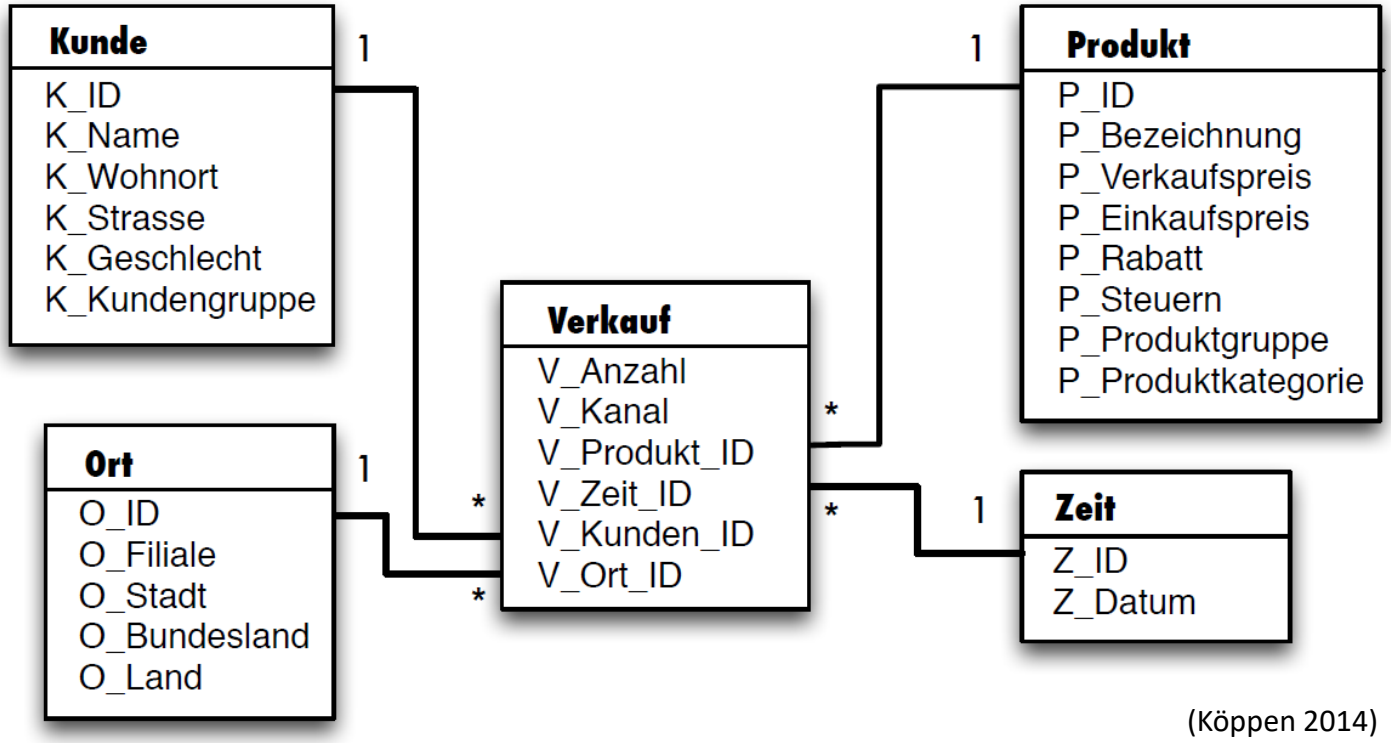
(Köppen 2014)

Star-Schema: Beispiel



Fakt/ Kennzahl
Anzahl





(Köppen 2014)

Vergleich Star- und Snowflake-Schema

Charakteristika von BI-Anwendungen

- Typischerweise Einschränkungen in Anfragen auf höherer Granularitätsstufe (Join-Operationen)
- Geringes Datenvolumen in den Dimensionstabellen, hohes Datenvolumen in der Faktentabelle
- Seltene Änderungen an Klassifikationen innerhalb der Dimensionen (Gefahr von Update-Anomalien weniger von Bedeutung)

Vorteile des Star-Schemas

- Einfache Struktur (vereinfachte Anfrageformulierung)
- Einfache und flexible Darstellung von Klassifikationshierarchien (Spalten in Dimensionstabellen)
- Effiziente Anfrageverarbeitung innerhalb einer Dimension (keine Join-Operation notwendig)

Anfrage Snowflake (5 Joins, steigt linear mit Länge der Aggregationspfade)

```
SELECT S_Name, YEAR(Z_Datum), SUM(V_Anzahl)
FROM   Verkauf, Filiale, Stadt, Produkt, Produktgruppe, Zeit
WHERE  V_Produkt_ID = P_ID AND P_PGruppe_ID = PG_ID AND
       V_Filial_ID = F_ID AND F_Stadt_ID = S_ID AND
       V_Zeit_ID = Z_ID AND PG_Bezeichnung = 'Wein'
GROUP BY S_Name, YEAR(Z_Datum)
```

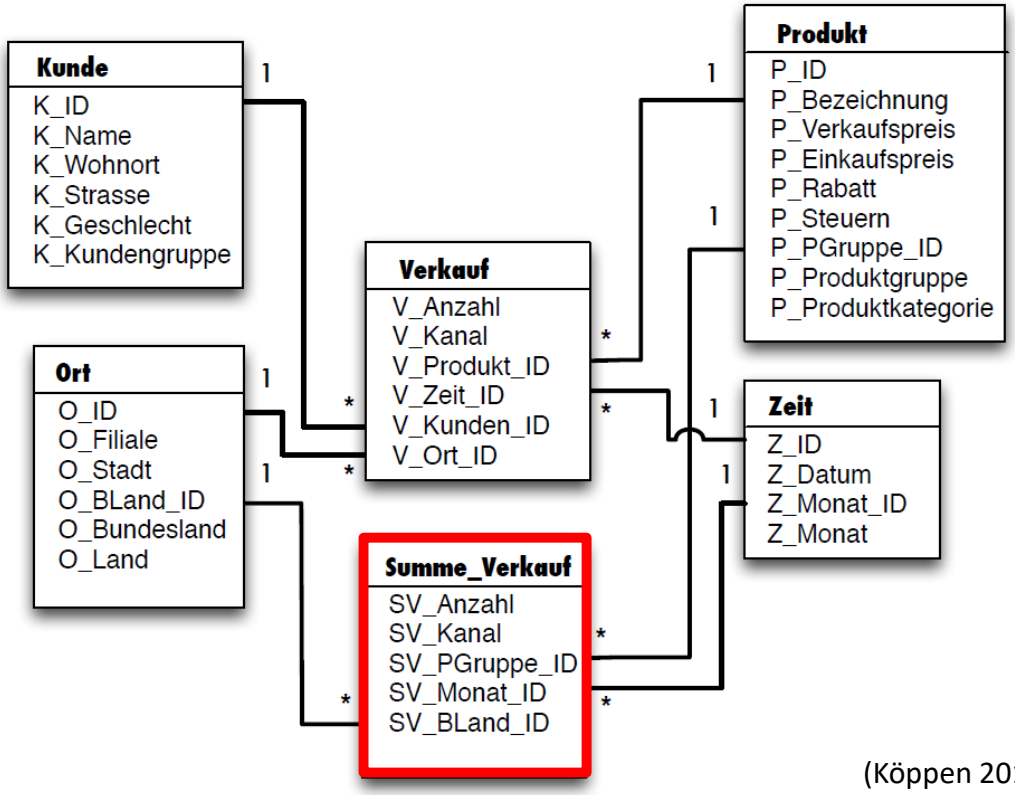
Anfrage Star (3 Joins, unabhängig von der Länge der Aggregationspfade)

```
SELECT O_Stadt, YEAR(Z_Datum), SUM(V_Anzahl)
FROM   Verkauf, Ort, Produkt, Zeit
WHERE  V_Produkt_ID = P_ID AND V_Zeit_ID = Z_ID AND
       V_Ort_ID = O_ID AND P_Produktgruppe = 'Wein'
GROUP BY O_Stadt, YEAR(Z_Datum)
```

Vorberechnete Aggregate: Fact-Constellation-Schema

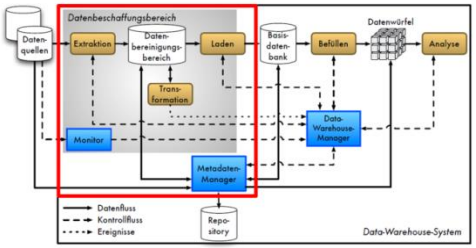
- Auslagerung vorberechnete Aggregate in eigene Faktentabellen (Summentabellen)
- Verweis von der Faktentabelle direkt auf die Attribute der jeweiligen Hierarchieebene der Dimensionstabellen (z.B. SV_BLand_ID)

- *Beispiel:* Vorberechnete Aggregate für die Kombination aus
  - Monat (Zeit)
  - Bundesland (Ort)
  - ProduktGruppe (Produkt)
  - alle Kunden (Kunde)



(Köppen 2014)

ETL: Wesentliche Phasen



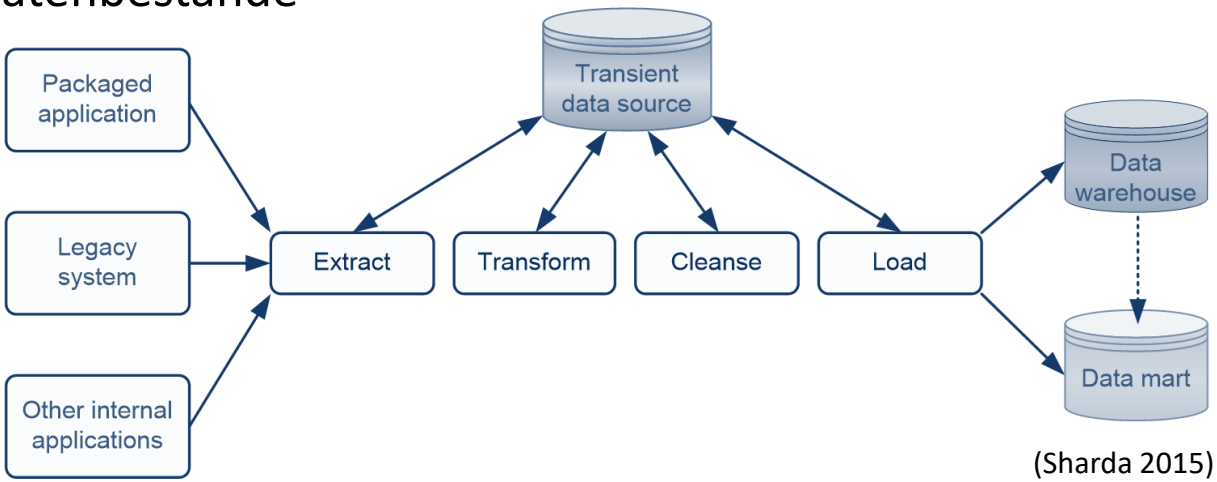
Der ETL-Prozess umfasst zwei wesentliche Phasen

1) Extraktion: Quellen → Data Staging Area

- Extraktion von Daten aus den Quellen in den Datenbeschaffungsbereich
- Erstellen / Erkennen von differentiellen Updates
- Erstellen von LOAD Files

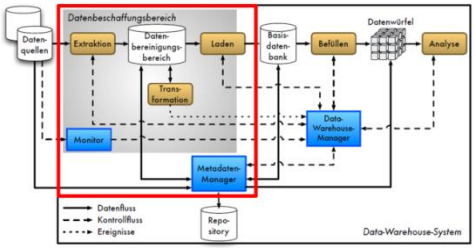
2) Laden: Staging Area → Basisdatenbank

- Data Cleaning und Tagging
- Erstellung integrierter Datenbestände
- In beiden Phasen findet eine **Transformation** der Daten statt



(Sharda 2015)

ETL: Zweck und Anforderungen

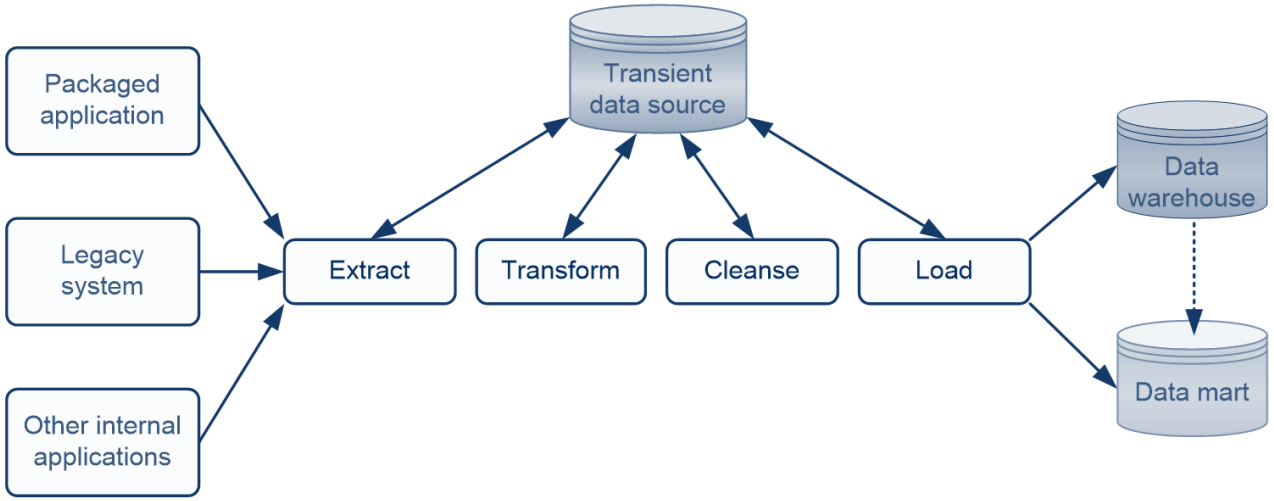


Zweck

- Kontinuierliche Datenversorgung des DWH
- Sicherung der Konsistenz des Datenstands des DWHs im Vergleich zum Datenstand in den Quellsystemen

Anforderungen

- Bereitstellung effiziente Methoden zur Extraktion essentiell
  → Sperrzeiten der Datenbanken (OLTP und Basisdatenbank) minimieren
- Rigorose Prüfungen muss durchgeführt werden → Datenqualität sichern



(Sharda 2015)

ETL: Herausforderungen

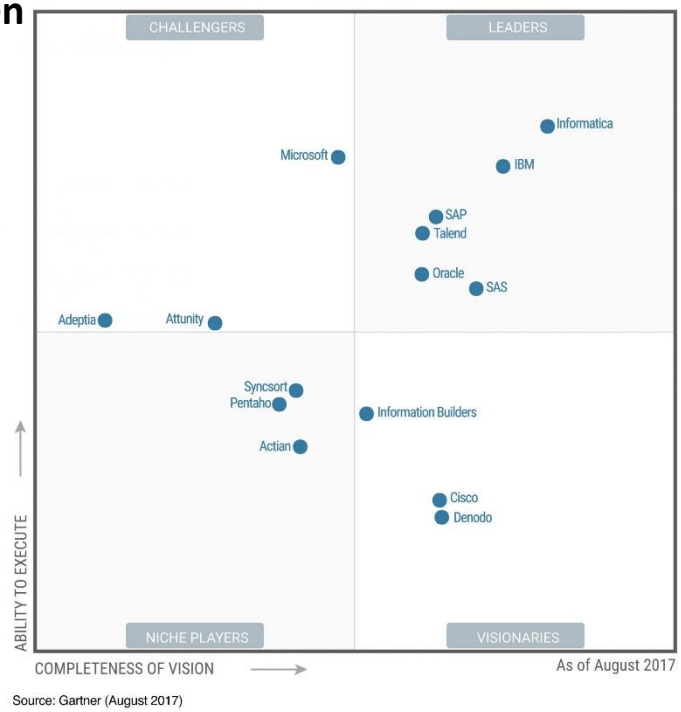
Häufig aufwendigster Teil des Data Warehousing (ca. 70% des Projektaufwands) aufgrund …

- **Vielzahl von Quellen** in heterogenen, historisch gewachsenen Infrastrukturen (Legacy-Systeme mit unterschiedlichen Funktionen für den Datenzugriff)
- **Heterogenität** der zu importierenden Daten
- **Datenvolumen** und Ressourcenbelastung operativer Systeme durch schlecht antizipierbare Managementanfragen
- Umfang und Komplexität der **notwendigen Transformationen**
  - Umwandlung der operativen Daten in betriebswirtschaftlich interpretierbare Daten
  - Schema- und Instanzintegration
  - Datenbereinigung

Softwareunterstützung

- Kaum durchgängige Methoden- und Systemunterstützung jedoch Vielzahl von Werkzeugen für die Manipulation (Extraktion, Transformation) vorhanden

Figure 1. Magic Quadrant for Data Integration Tools

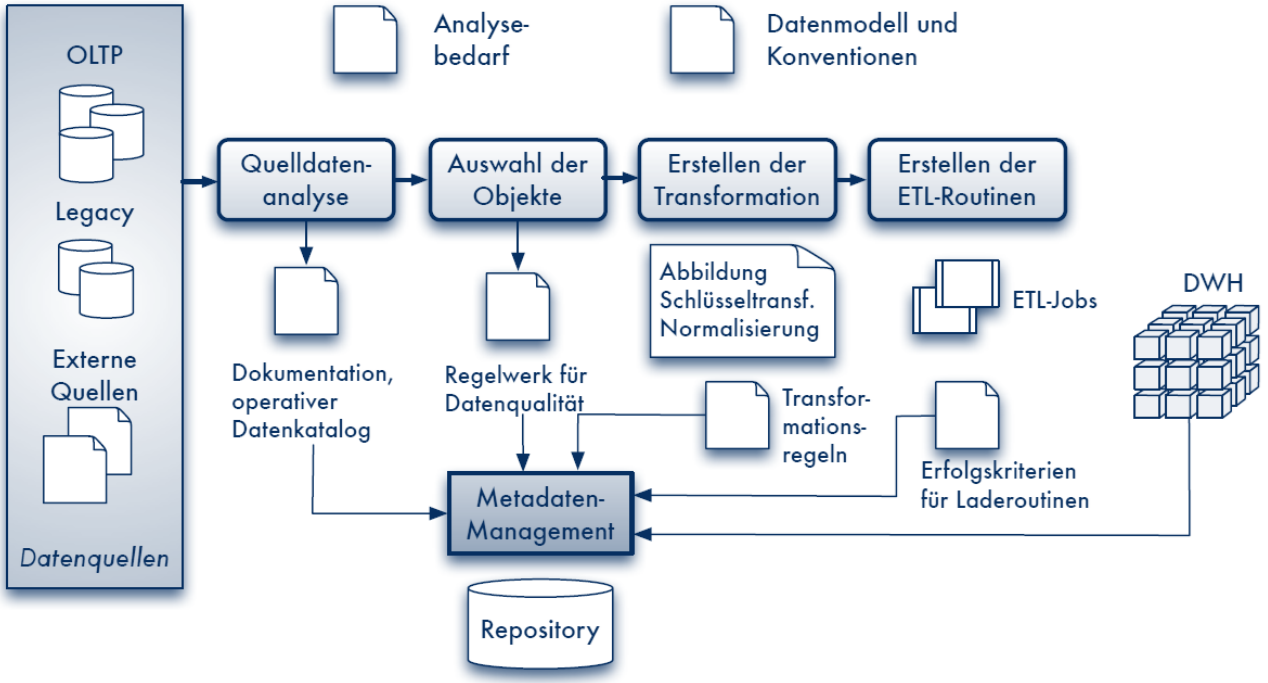


Source: Gartner (August 2017)

Konzeption von ETL-Prozessen

- **Extraktion:** Selektion des relevanten Ausschnitts der Daten aus den Quellen und Bereitstellung für Transformation
- **Transformation:** Anpassung der Daten an vorgegebene Schema und Qualitätsanforderungen
- **Laden:** physisches Einbringen der Daten aus dem Datenbeschaffungsbereich in das Data Warehouse (einschl. eventuell notwendiger Aggregationen)

Schritte für die Konzeption und Umsetzung von ETL-Prozessen

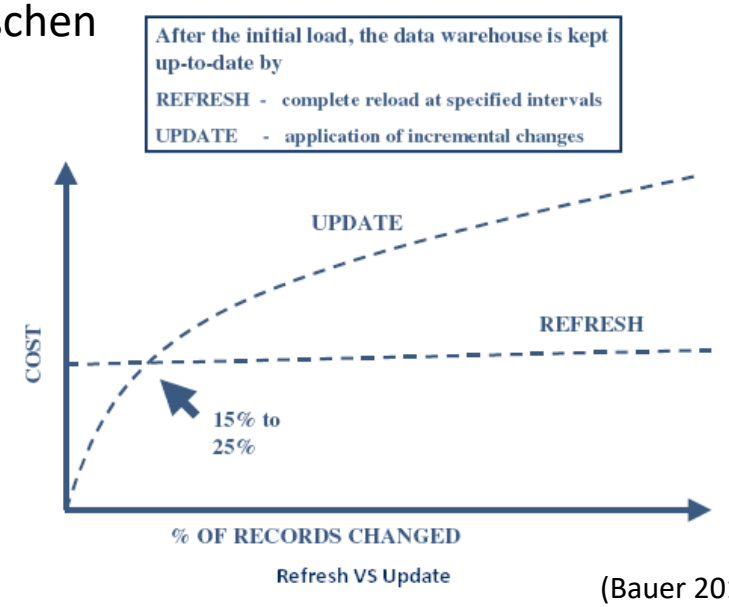


(Köppen 2014)

Extraktion: Initiales Laden vs. Inkrementelles Laden

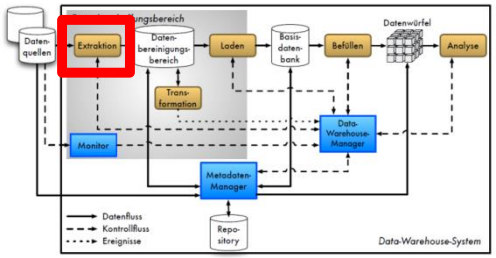
In der Betriebsphase eines Data Warehouses werden in Bezug auf die Datenbeschaffung zwei Phasen unterschieden

- **Initiales Laden:** Basisdatenbank und Data Marts werden erstmalig vollständig mit den extrahierten Daten geladen. Ab diesem Zeitpunkt können Anwender auf die Daten mittels Analysewerkzeuge zugreifen
- **Inkrementelles Laden:** nur die seit der letzten Aktualisierung geänderten Daten in den Quellen werden extrahiert und in die Basisdatenbank integriert
  das Inkrementelle Laden erfolgt häufig in periodischen Abständen, z.B. 1x täglich im Nachtbetrieb
- **Refresh:** vollständiges Neuladen des DWH
  → entspricht technisch einem initialen Laden



(Bauer 2014)

Extraktion: Extraktionskomponente



Aufgabe: Übertragung von Daten aus Quellsystemen in den Datenbeschaffungsbereich

Extraktionszeitpunkt: abhängig von Monitoring-Strategie

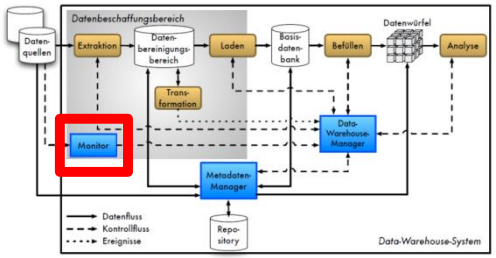
1) Synchrone Benachrichtigung: Quelle propagiert jede Änderung

2) Asynchrone Benachrichtigung

- ***Periodisch:*** Quellen erzeugen regelmäßig Extrakte, DWH fragt regelmäßig Datenbestand ab
- ***Ereignisgesteuert:*** Quelle informiert alle X Änderungen
- ***Anfragegesteuert:*** DWH erfragt Änderungen vor jedem tatsächlichen Zugriff eines Benutzers auf einen bestimmten Datenbestand im DWH

Technische Realisierung: Nutzung von Standardschnittstellen (z.B. ODBC, JDBC)

Extraktion: Monitore



Aufgabe: Entdeckung von Änderungen in einer Datenquelle

Mögliche Strategien:

1) Trigger-basiert

- Aktive Datenbankmechanismen werden verwendet → Auslösen von Triggern bei Datenänderungen
- Kopieren der geänderten Zeilen einer Tabelle (Tupel) in einen separaten Bereich für die Datenextraktion

2) Replikationsbasiert: Nutzung von Replikationsmechanismen zur Identifikation und Übertragung geänderter Daten

3) Log-basiert: Analyse von Log-Dateien des DBMS zur Erkennung von Änderungen

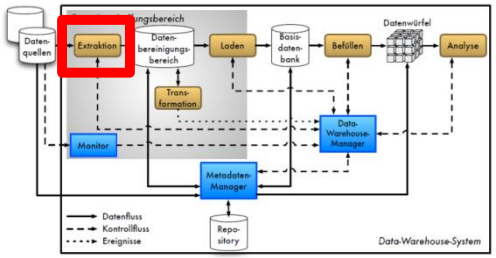
4) Zeitstempelbasiert

- Zuordnung eines Zeitstempel zu Tupeln (Zeilen einer Tabelle)
- Aktualisierung des Zeitstempels bei Änderungen
- Identifizierung von Änderungen seit der letzten Extraktion durch Zeitvergleich

5) Snapshot-basiert

- Periodisches Kopieren des Datenbestandes in Datei (Snapshot)
- Vergleich von Snapshots zur Identifizierung von Änderungen

Extraktion: Strategie der Datenlieferung/ -bereitstellung



Snapshots: Quelle liefert immer kompletten aktuellen Daten-bestand (z.B. alle Lieferantenstammdaten, alle Bestellungen)

- *Herausforderung:* Änderungen (Hinzufügen, Löschen, Update) erkennen im Vergleich zum letzten Snapshot und die Historie korrekt abbilden

Änderungs-Logs: Quelle liefert jede Änderung (z.B. Transaktions- bzw. Änderungs-Logs)

- *Ziel:* Änderungen effizient einspielen

Netto-Logs: Quelle liefert Änderungen seit dem letzten Abzug (z.B. Snapshot-Deltas)

- Keine vollständige Historie möglich
- *Ziel:* Änderungen effizient einspielen

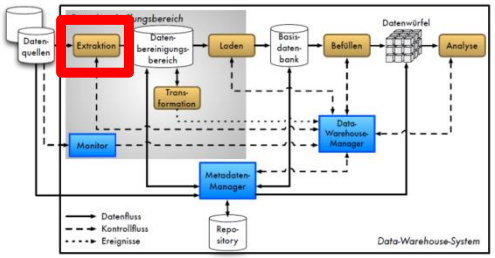
Änderungsprotokoll SAP ERP

List Edit Goto System Help

Changes Purchase Order 4500017071 Item 10

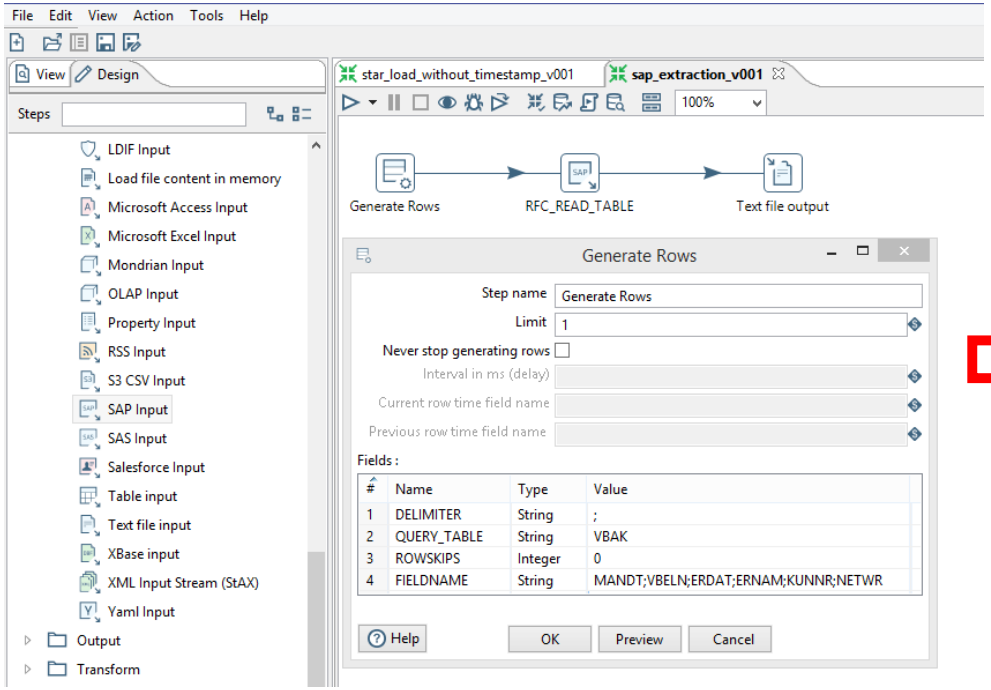
It...	Object	Short text	Action	New value	Old value	Name	Date	Time	Doc. no.	TCode
10	Item	Gross order value in PO currency	Changed	10.000,00 ...	100.000,00 ...	KUEBLER	23.01.20...	17:19:...	547454	ME22N
			Changed	11.000,00 ...	10.000,00 M...	KUEBLER	01.02.20...	12:44:...	548066	ME22N
			Changed	10.000,00 ...	11.000,00 M...	KUEBLER		13:34:...	548080	ME22N
		Net Order Value in PO Currency	Changed	20.000,00 ...	25.000,00 M...	KUEBLER	23.01.20...	10:28:...	547244	ME22N
			Changed	200.000,00 ...	20.000,00 M...	KUEBLER		10:31:...	547245	ME22N
			Changed	100.000,00 ...	200.000,00 ...	KUEBLER		16:25:...	547446	ME22N
			Changed	10.000,00 ...	100.000,00 ...	KUEBLER		17:19:...	547454	ME22N
			Changed	11.000,00 ...	10.000,00 M...	KUEBLER	01.02.20...	12:44:...	548066	ME22N
			Changed	10.000,00 ...	11.000,00 M...	KUEBLER		13:34:...	548080	ME22N
		Purchase Order Quantity	Changed	20,000 PC	25,000 PC	KUEBLER	23.01.20...	10:28:...	547244	ME22N
			Changed	200,000 PC	20,000 PC	KUEBLER		10:31:...	547245	ME22N
			Changed	100,000 PC	200,000 PC	KUEBLER		16:25:...	547446	ME22N
			Changed	10,000 PC	100,000 PC	KUEBLER		17:19:...	547454	ME22N
			Changed	11,000 PC	10,000 PC	KUEBLER	01.02.20...	12:44:...	548066	ME22N
			Changed	10,000 PC	11,000 PC	KUEBLER		13:34:...	548080	ME22N

Extraction: Example SAP ERP with pentaho



Configure Extraction: Extract all records from table (VBAK – Sales Document Header) and select relevant fields

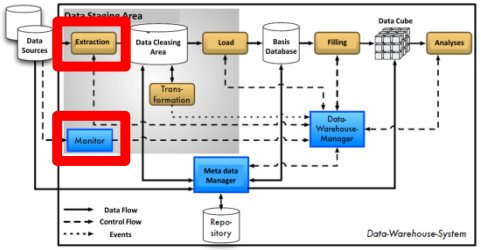
- *VBELN*: SalesDocumentNumber
- *ERDAT*: Created on, ERNAM: Created by
- *KUNNR*: CustomerID
- *NETWR*: Net Value



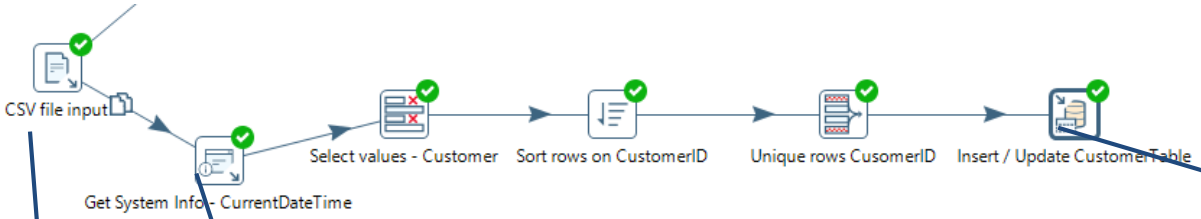
Result File

```
WA
"902;0050000005;19941129;BEHRMANN    ;          ;          0.00"
"902;0050000007;19941208;BEHRMANN    ;          ;          0.00"
"902;0050000010;19950804;HEINZ       ;          ;          0.00"
"902;0050000011;19960214;THIELE      ;          ;          0.00"
"902;0050000024;19960916;ROHRMEIER   ;          ;          0.00"
"902;0000004969;19970102;CURA        ;0000001390;       5500.00"
"902;0000004970;19970103;CURA        ;0000001175;      32838.00"
"902;0000004971;19970107;CURA        ;0000001001;      12200.00"
"902;0000004972;19970121;BOLLINGER   ;0000002200;      28604.00"
"902;0000004973;19970121;BOLLINGER   ;0000001033;      19719.00"
"902;0000004974;19970121;BOLLINGER   ;0000002140;      46686.00"
"902;0000004975;19970121;BOLLINGER   ;0000001002;      32778.00"
"902;0000004976;19970121;BOLLINGER   ;0000002004;      36726.00"
"902;0000004977;19970121;BOLLINGER   ;0000001360;       9352.00"
"902;0000004978;19970121;BOLLINGER   ;0000002130;      11162.00"
"902;0000004979;19970121;BOLLINGER   ;0000001360;      13013.00"
"902;0000004980;19970121;BOLLINGER   ;0000002130;      14230.00"
```

Extraction: Delta Load with Timestamp using pentaho



Handling of changes within timestamp



Get current system date + time

Compare new records with existing records

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1
1	CUSTOMERID	=	Customer
2	CustomerDescr	=	CustomerDescr
3	City	=	City
4	Salesorg	=	Salesorg
5	Country	=	Country

1 SalesDataPivotV01.txt *

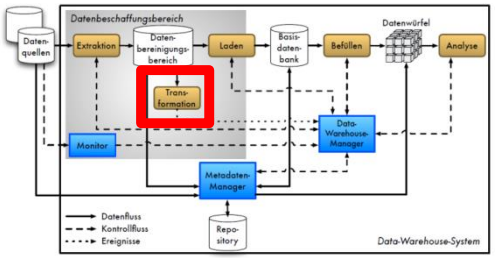
2007	1	1	15000	Bavaria Bikes	München	DS00	DE	100002	20
2007	1	1	15000	Bavaria Bikes	München	DS10	DE	100002	30
2007	1	1	15000	Bavaria Bikes	München	DS00	DE	100002	40
2007	1	1	15000	Bavaria Bikes	München	DS00	DE	100002	50
2007	1	1	15000	Bavaria Bikes	München	DS00	DE	100002	60

Change Source Data: SO from DS00 to DS10

Result Table

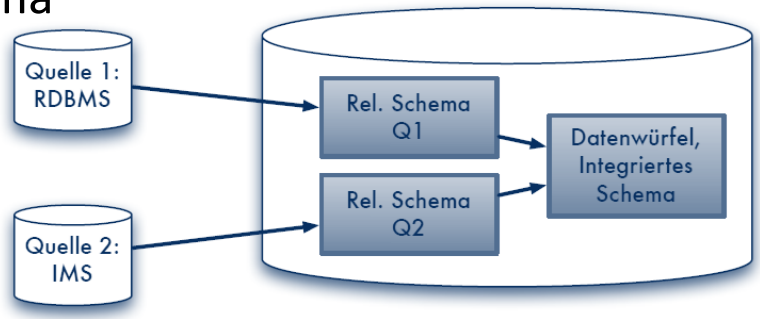
CUSTOMERID	CUSTOMERDESCR	CITY	SALESORG	COUNTRY	LOADTIME
12000	Northwest Bikes	Seattle	UW00	US	2016-05-29 17:34:20
13000	Airport Bikes	Frankfurt	DS00	DE	2016-05-29 17:34:20
14000	Alster Cycling	Hamburg	DN00	DE	2016-05-29 17:34:20
15000	Bavaria Bikes	München	DS00	DE	2016-05-29 17:34:20
15000	Bavaria Bikes	München	DS10	DE	2016-05-29 17:37:07
16000	Capital Bikes	Berlin	DN00	DE	2016-05-29 17:34:20
17000	Cruiser Bikes	Hannover	DN00	DE	2016-05-29 17:34:20

Transformation: Herausforderungen



Daten im Datenbeschaffungsbereich nicht im Format der Basisdatenbank → **Strukturelle Heterogenität**

- *Datenbeschaffungsbereich:* Quellnahes Schema
- *Basis-DB:* Analyseorientiertes Schema

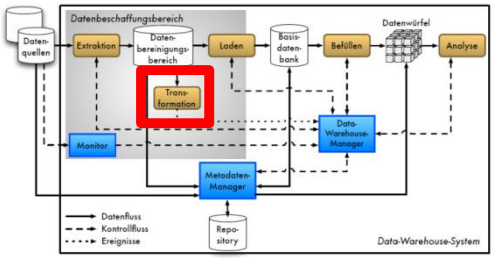


Daten- und Schemaheterogenität

- *Hauptdatenquelle:* OLTP-Systeme (relationales Datenmodell)
- *Sekundärquellen:* Dokumente in firmeninternen Altarchiven, Dokumente im Internet via WWW, FTP
  - Unstrukturiert: Zugriff über Suchmaschinen, . . .
  - Semistrukturiert: Zugriff über Suchmaschinen, Mediatoren, Wrapper als XML-Dokumente o.ä.

→ **Grundproblem der Transformation: Heterogenität der Quellen**

Transformationsaufgaben im ETL-Prozess



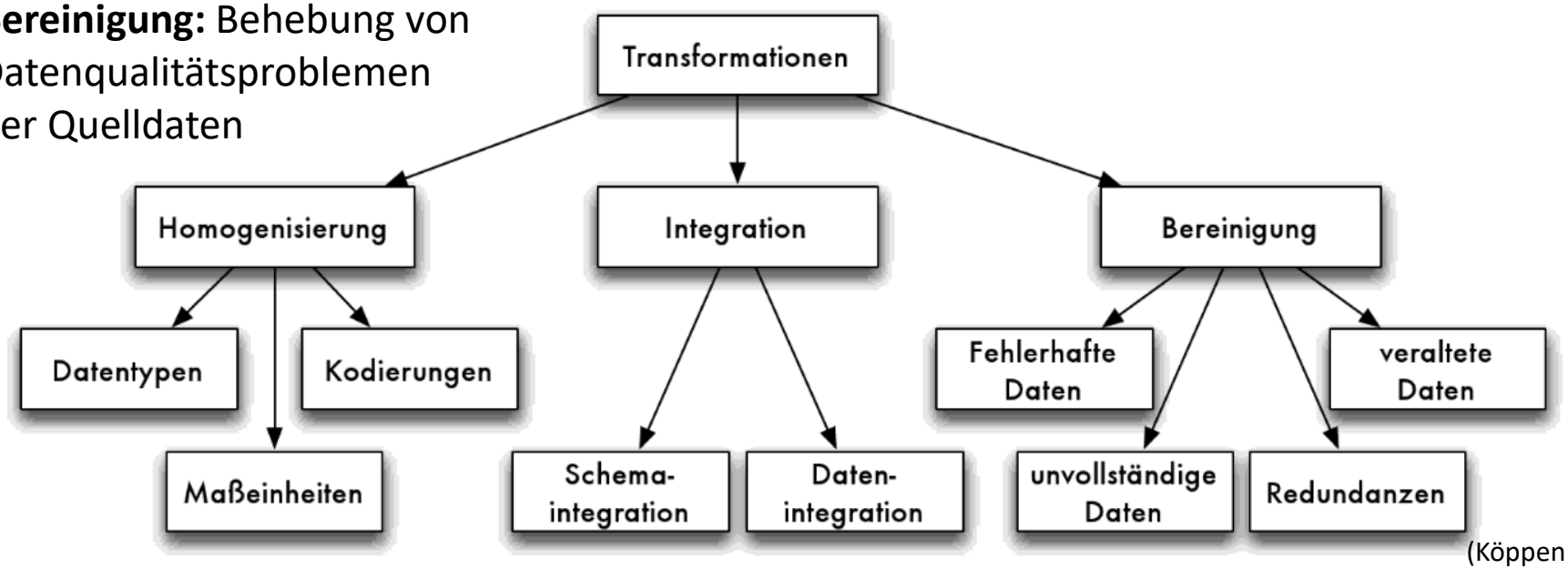
Homogenisierung: Transformationen, die die Daten in eine einheitliche Repräsentation überführen für die spätere Integration

- z.B. Umwandlung von Datentypen, Vereinheitlichung von Datumsangaben, Kodierung (z.B. Kürzel für Bundesländer)

Integration: Zusammenführung von Daten aus mehreren Quellen

- z.B. Mischen ganzer Tabellen (Schematintegration (relation merging)) Verschmelzen einzelner Datensätze (Datenintegration (record linkage)

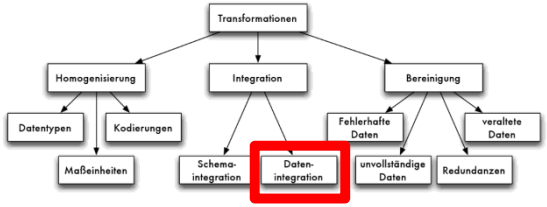
Bereinigung: Behebung von Datenqualitätsproblemen der Quelldaten



(Köppen 2014)

HAW HAMBURG

Datenintegration: Schlüsseldisharmonien



Integration der Daten zu einem betriebswirtschaftlichen Objekt (z.B. Kunde) aus unterschiedlichen Systemen

- Auflösung Schlüsseldisharmonien mit Hilfe von **Mapping-Tabellen** und **Surrogaten**

Beispiel 1

Quelle	Relation	Attribut	lokaler Schlüssel	globales Surrogat
system1	kunde	kunden_nr	12345	66
system1	kunde	kunden_nr	44444	69
system2	customer	customer_id	A134	69
system2	customer	customer_id	B777	72
system2	customer	customer_id	X007	66

(Bauer 2013)

Beispiel 2

AD_SYS	...	Kunde_Text	LOADTIME
AD-FX8257		Müller	31DEC2009:23:03:08
AD-FH2454		Meier	31DEC2009:23:03:08
AD-FX7059		Schulz	31DEC2009:23:03:08
AD-FT2567		Schmitz	31DEC2009:23:03:08
...	...	...	...

AC_SYS	Kunde_Text	Kunde_Status
3857_ACC	Müller	A
3525_ACC	Meier	A
3635_ACC	Schulz	A
3566_ACC	Schmitz	B
...	...	...

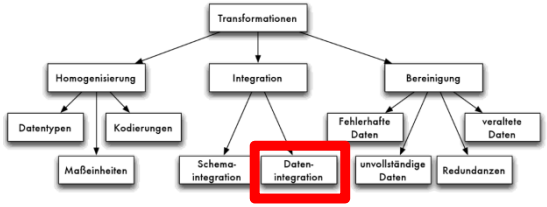
CC_SYS	Kunde_Grp	Kunde_Text	LOADTIME
59235395	Handel	Müller	31DEC2009:23:03:08
08485356	Industrie	Meier	31DEC2009:23:03:08
08555698	Industrie	Schulz	31DEC2009:23:03:08
85385386	Handel	Schmitz	31DEC2009:23:03:08
...	...	...	...

Kunde_ID	Kunde_Text	...	AD_SYS	CC_SYS	AC_SYS	...	LOADTIME
0001	Müller		AD-FX8257	59235395	3857_ACC		31DEC2009:23:03:08
0002	Meier		AD-FH2454	08485356	3525_ACC		31DEC2009:23:03:08
0003	Schulz		AD-FX7059	08555698	3635_ACC		31DEC2009:23:03:08
0004	Schmitz		AD-FT2567	85385386	3566_ACC		31DEC2009:23:03:08
...	...	...	...	...	...	...	...

(Kemper 2010)

Legende: AD – Außendienstsystem, CC – Call-Center-Anwendung, AC – Abrechnungs-/Accounting-System

Datenintegration: betriebswirtschaftliche Sicht



Für die Harmonisierung der **betriebswirtschaftlichen Bedeutung** sind Transformationsregeln zu implementieren, die das operative Datenmaterial in Bezug auf die betriebswirtschaftliche Attribute wie z.B. die

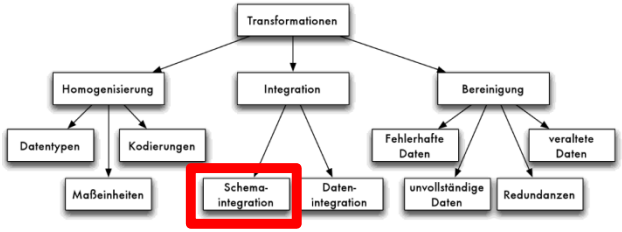
- gebiets- und ressortspezifische Gültigkeit,
- Währung oder die
- Periodenzuordnung

in einheitliche Werte überführen

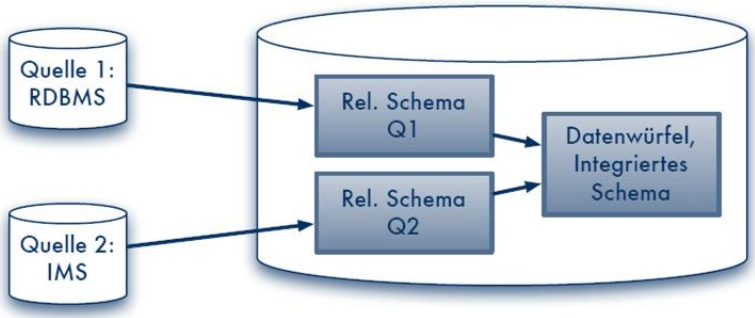
Buchungen im SAP ERP (Geschäftsjahr Oktober bis September)

BuKr.	Belegnummer	Jahr	Art	Belegdatum	Buch.dat.	Periode	Erfasst am	Erfaßt um
1000	100000001	2014	AB	03.07.2014	03.07.2014	10	03.07.2014	08:45:15
1000	100000002	2014	AB	03.07.2014	03.07.2014	10	03.07.2014	09:00:15
1000	100000003	2014	AB	11.07.2014	11.07.2014	10	11.07.2014	11:41:59
1000	100000004	2014	AB	11.07.2014	11.07.2014	10	11.07.2014	11:43:50
1000	1400000000	2014	RV	10.06.2014	10.06.2014	9	10.06.2014	10:29:42
1000	1400000001	2014	RV	10.06.2014	10.06.2014	9	10.06.2014	10:33:22

Schema-Integration durch Schema Mapping

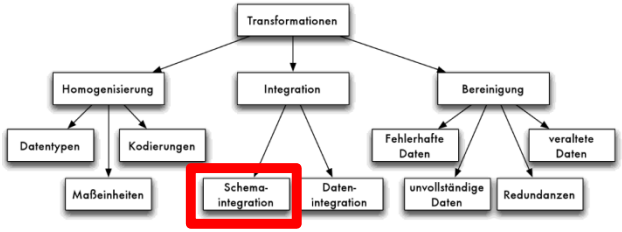


Schema-Integration: Datentransformation zwischen heterogenen Schemata

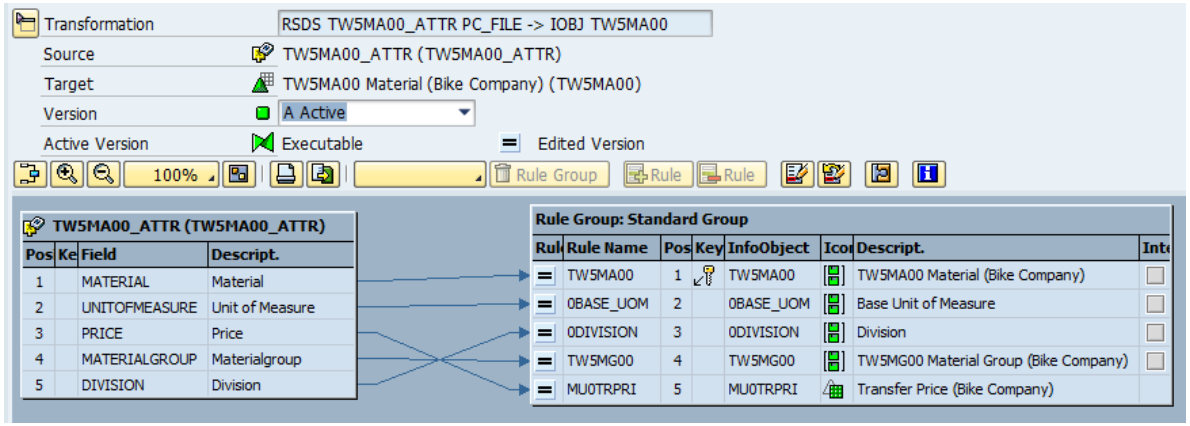


- Üblicherweise schreiben Experten komplexe Anfragen oder Programme (Zeitintensiv, Experte für die Domäne, Schemata und Anfragen notwendig)
- **Idee:** Automatisierung
  - Gegeben: Zwei Schemata und ein high-level Mapping dazwischen
  - Gesucht: Anfrage zur Datentransformation
- **Problem:** Generierung der „richtigen“ Anfrage unter Berücksichtigung des **Quell-** und **Ziel-Schemas**, des **Mappings** und der **Nutzer-Intention** (Semantik)

Mapping von Datenfeldern: Beispiel SAP BW



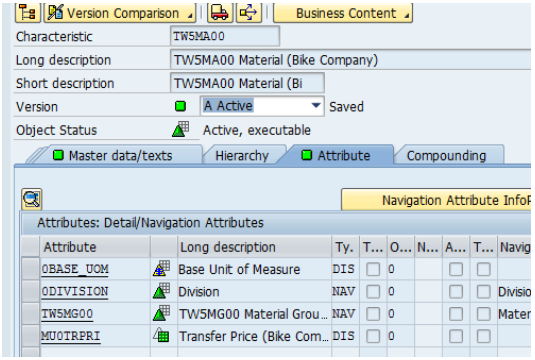
- Die Konvertierung erfolgt beim Mapping zwischen Feldern der Quelle und der Zieldatenstruktur



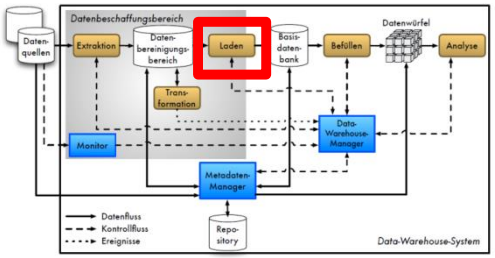
Quelle

```
Material;Unit of Measure;Price;Materialgroup;Division
CB-0010;ST;751;02021;11
CB-0011;ST;256;02021;11
CB-0012;ST;428;02021;11
CB-0013;ST;965;02021;11
CB-0014;ST;1159;02021;11
KB-0010;ST;68;02022;11
KB-0011;ST;54;02022;11
KB-0012;ST;155;02022;11
```

Definition InfoObject



Laden



Aufgabe: Übertragung der bereinigten und aufbereiteten detaillierten Daten in die Basisdatenbank bzw. das DWH

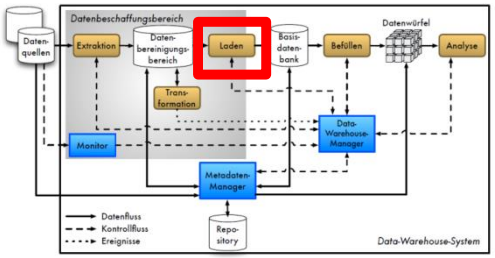
- *Ziel:* Effizientes Einbringen von externen Daten in die Basisdatenbank

Kritischer Punkt: Ladevorgänge blockieren unter Umständen das komplette DWH (Schreibsperre auf den Tabellen in der Basisdatenbank)

Verfügbarkeit des DWH während des Ladevorgangs

- *Online:* Basisdatenbank (DWH) steht weiterhin zur Verfügung
- *Offline:* stehen nicht zur Verfügung (Zeitfenster: nachts, Wochenende)
- **Zu berücksichtigende Aspekte**
  - Trigger Integritätsbedingungen deaktivieren?
  - Indexaktualisierung
  - Update oder Insert?

Laden: Technische Umsetzung

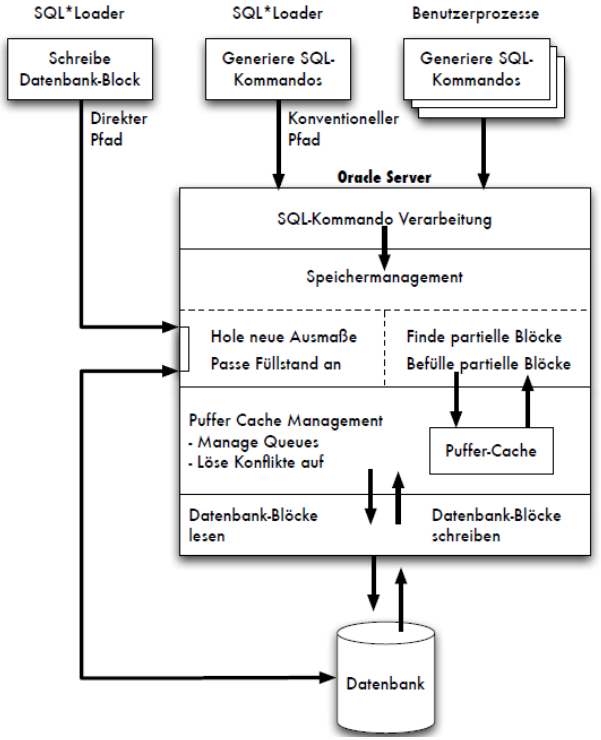


satzbasiert

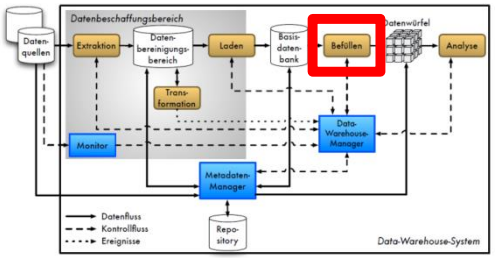
- Benutzung von Standard-Schnittstellen: SQL, JDBC, ODBC, . . .
- Arbeitet im normalen Transaktionskontext, Trigger, Indexe und Constraints bleiben aktiv (Manuelle Deaktivierung möglich)
- Sperren können durch COMMIT verringert werden
- Benutzung von Prepared Statements

BULK Load: DBMS-spezifische Erweiterungen zum Laden großer Datenmengen

- Läuft (meist) in speziellem Context
  z.B. Oracle: DIRECTPATH option im Loader
- Komplette Tabellensperre
- Keine Beachtung von Triggern oder Constraints
- Indexe werden erst nach Abschluss aktualisiert
- Kein transaktionaler Kontext, Kein Logging
- Checkpoints zum Wiederaufsetzen



Befüllen

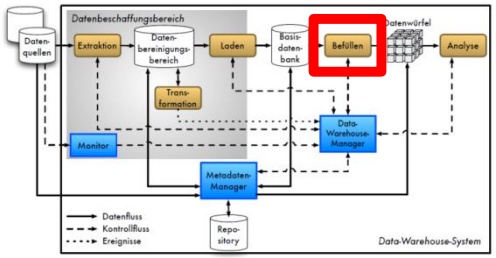


Aufgabe: Übertragung und Aufbereitung der Daten (z.B. Aggregation) aus der Basisdatenbank in die Data Marts

Wesentliche Schritte

- Aggregation der detaillierten Daten aus der Basisdatenbank in Abhängigkeit des Detaillierungsgrads im Data Mart (z.B. von Tages- auf Monatsbasis
- Befüllen/ Update der Dimensionstabellen
- Befüllen/ Update der Faktentabelle
- *Ziel:* Effizientes Einbringen von externen Daten in die Basisdatenbank

Technische Umsetzung: Benutzung von Standard-Schnittstellen (z.B. SQL) des zugrundeliegenden DBMS



Filling: Sample Data with pentaho

- Filling of fact table in star schema

Select from bi_sales_basis_database

Update Star: FactTable

Source data (basis database): 48.384 records

ORDERNUMBER	ORDERITEM	YEAR	MONTH	DAY	CUSTOMERID	CUSTOMERDESCR	CITY	SALESORG	COUNTRY	PRODUCT
100001	10	2007	1	1	5000	Beantown Bikes	Boston	UE00	US	DXTR2000
100001	20	2007	1	1	5000	Beantown Bikes	Boston	UE00	US	PRTR2000
100001	30	2007	1	1	5000	Beantown Bikes	Boston	UE00	US	ORMN1000
100001	40	2007	1	1	5000	Beantown Bikes	Boston	UE00	US	ORHT1000
100001	50	2007	1	1	5000	Beantown Bikes	Boston	UE00	US	DXRD1000
100001	60	2007	1	1	5000	Beantown Bikes	Boston	UE00	US	DXRD2000

SQL-Statement for Aggregation on month level

```
SELECT CUSTOMERID,
       PRODUCT,
       EXTRACT(YEAR_MONTH from TransactionDATE),
    sum(SALESQUANTITY),
    sum(REVENUEEUR),
    sum(REVENUEUSD),
    sum(DISCOUNTEUR),
    sum(DISCOUNTUSD),
    sum(COSTOFGOODSEUR),
    sum(COSTOFGOODSUSD)
FROM `bi_sales_basis_database`.`bi_sales_dwh`
group by CUSTOMERID, PRODUCT, EXTRACT(YEAR_MONTH from TransactionDATE)
```

Result table (fact table): 29.188 records

CUSTOMERID	PRODUCTID	MonthYear	SALESQUANTITY	REVENUEEUR	REVENUEUSD	DISCOUNTEUR	DISCOUNTUSD	COGMEUR	COGMUSD
1000	BOTL1000	200701	1.00	20.00	20.00	1.00	1.00	10.00	10.00
1000	BOTL1000	200703	2.00	40.00	40.00	2.00	2.00	20.00	20.00
1000	BOTL1000	200704	2.00	40.00	40.00	1.00	1.00	20.00	20.00
1000	BOTL1000	200705	2.00	40.00	40.00	1.00	1.00	20.00	20.00
1000	BOTL1000	200706	6.00	120.00	120.00	4.00	4.00	60.00	60.00
1000	BOTL1000	200707	2.00	40.00	40.00	1.00	1.00	20.00	20.00
1000	BOTL1000	200708	2.00	40.00	40.00	1.00	1.00	20.00	20.00
1000	BOTL1000	200709	1.00	20.00	20.00	1.00	1.00	10.00	10.00