Das Data Warehouse

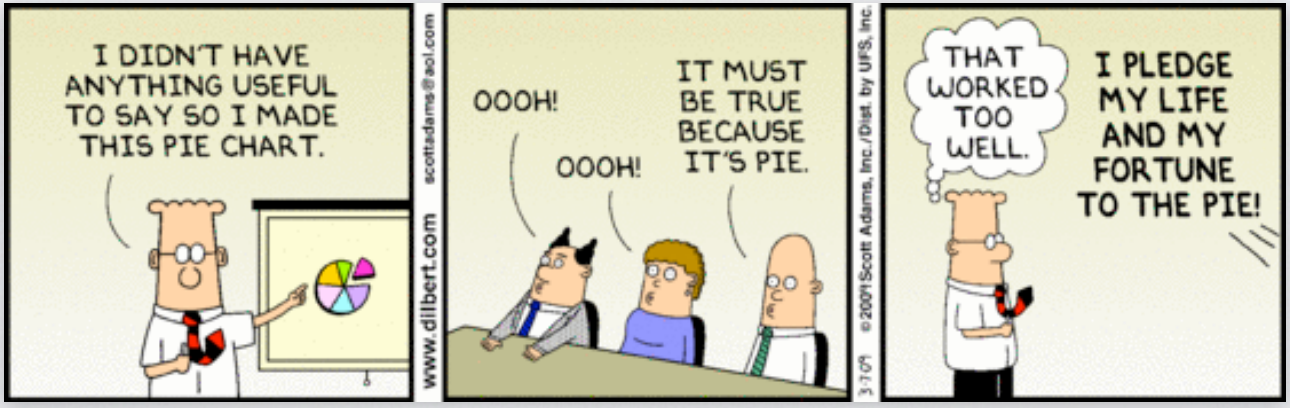
Delphi Tage 2013

Daniela Sefzig (Delphi Praxis, Daniela.S)
daniela.sefzig@alien.at

Version 1.0

Ziel des Vortrages

- Grundsätzliches Verständnis der Funktion eines Data Warehouses
- Know-How, um mitreden zu können, wenn es um das Thema Data Warehouse geht
- Grundlagen, um eigenständig ein Data Warehouse zu planen
- Dokument zum Nachlesen
- Beispiele zum Download
- Info über weiterführende Lektüre, um tiefer in das Thema einzutauchen



Agenda

- Was ist ein Data Warehouse?
- Was benötigt ein erfolgreiches Data Warehouse?
- Aufbau (Mythen, Fehlerquellen, Design, Modellierung)
- Der ETL-Prozess
- Beispiel-Implementierung
- Auswertungen und Statistiken

Was ist ein Data Warehouse?

A data warehouse is a *subject oriented, integrated, time variant, non-volatile collection* of data *in support of management's decision making process.*

William H. Inmon, ´92

- ***subject oriented***: für bestimmte Entitätentypen zugeschnitten, z.B. Verkäufe, Produkte, geographische Bereiche.
- ***integrated:*** die Daten im Data Warehouse stammen i.d.R. aus verschiedenen Quelldatenbanken, z.B., aus mehreren Verlagskatalogen, Lagerbeständen einzelner Lager, Einnahmen einzelner Läden, usw.
- ***time-variant:*** Data Warehouse zeigt die zeitliche Evolution der betrachteten Entitäten.
- ***non-volatile:*** Daten werden nicht gelöscht oder nachträglich geändert, Änderungen im Datenbestand sind allein auf das Laden neuer Daten zurückzuführen.
- ***support decision making:*** nur wichtige Daten für solche Entscheidungen speichern.

Was ist ein Data Warehouse?

Ein Data Warehouse ist **kein** weiteres IT Projekt, es ist ein strategisches Projekt!

Unterschiedliche Ansätze

	Relationale Datenbank OLTP (Online Transaction Processing)	Data Warehouse OLAP (Online Analytical Processing)
Normalisierung	Normalisierung meist bis zur 3. oder 4. Ebene	Denormalisiertes Datenmodell, um das Laufzeitverhalten zu verbessern
Optimiert auf	Daten schreiben	Daten lesen (analytische Abfragen)
Redundante Daten	werden vermieden	werden bewusst implementiert
Aggregierte Daten	üblicherweise nicht	ja
Historische Daten	nur für einen gewissen Zeitraum	werden für einen definierten Zeitraum gespeichert
Datenvolumen	klein	sehr groß
Inhalt der Daten	Anwendungs- und Funktionsbezogen	Themenbezogen
Modellierung	Entity Relationship Modell (ERM) Relationales Datenbank Model (RDM)	Application Design for Analytical Techniques (ADAPT) Dimensional Fact Model (DFM)

Unterschiedliche Sicht der Beteiligten

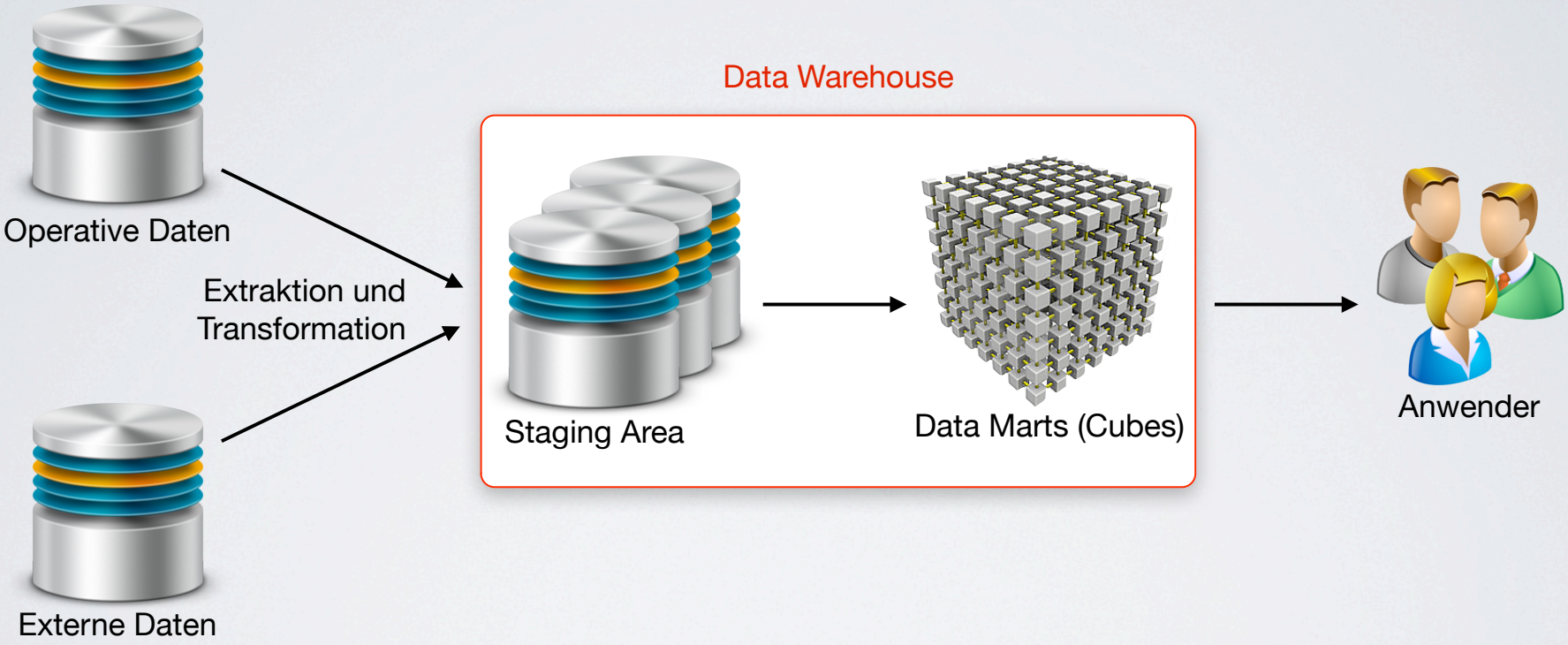
Der Entwickler

- Backup
- Datendurchsatz
- Loadbalancing

Das Management bzw. der Analyst

- Portfolio
- Umsatz
- Werbung

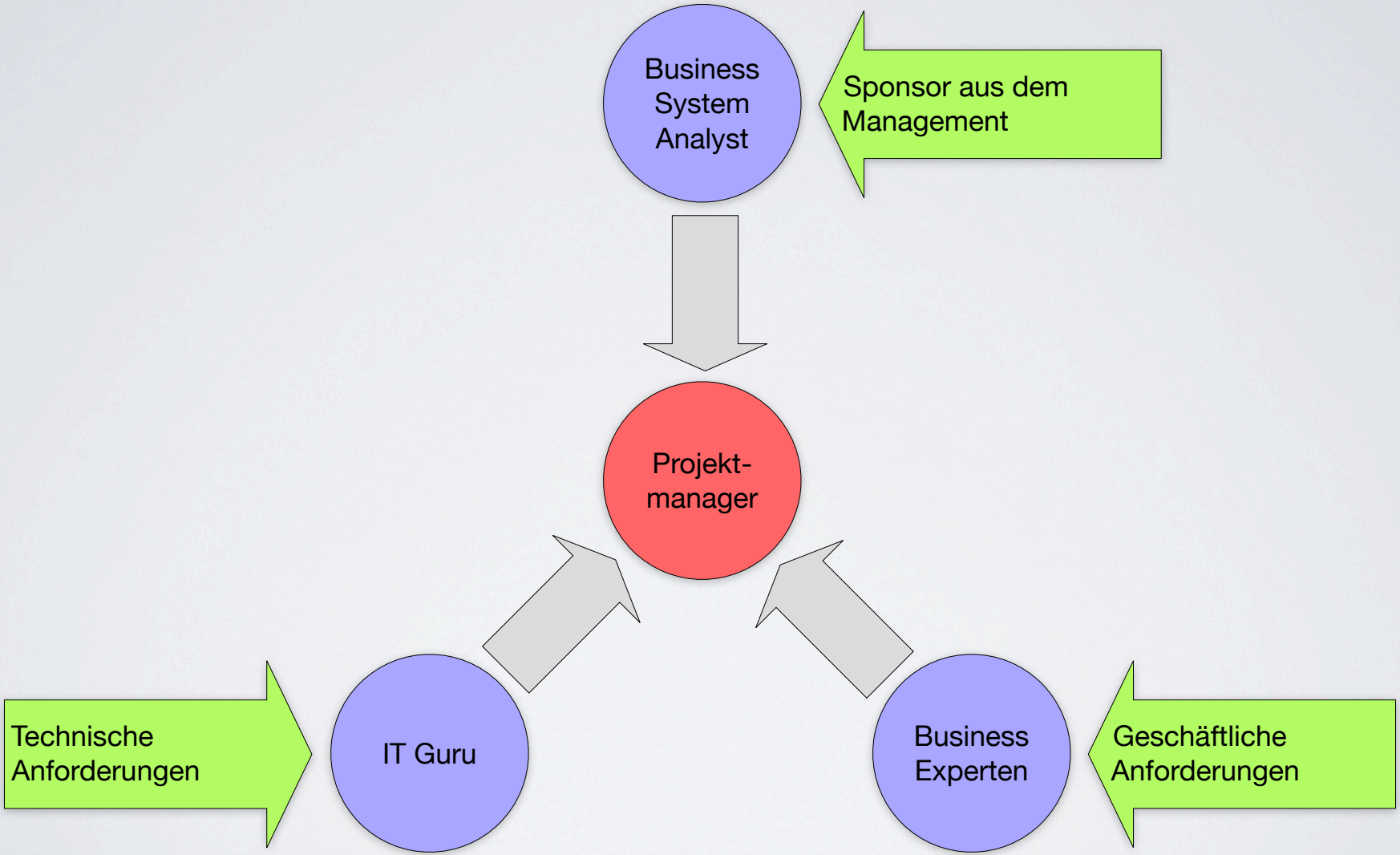
Unterschiedliche Sicht der Beteiligten



Was benötigt ein erfolgreiches Data Warehouse?

- Unterstützung des Managements (zumindest eine Person)
- Einbeziehen der einzelnen Abteilungen bzw. Analysten und Erfassen ihrer Bedürfnisse
- Ein Mitarbeiter ist dezidiert für das DWH zuständig und er bleibt dies auch
- Schrittweise Implementierung, beginnend mit dem Geschäftsprozess, der dem Unternehmen den meisten Nutzen bringt
- Die Bedürfnisse der Benutzer müssen befriedigt werden, nicht die der IT-Mitarbeiter
- Zum Testen bereits Echt-Daten verwenden, damit testende Personen die Richtigkeit der Werte überprüfen können

Was benötigt ein erfolgreiches Data Warehouse?



Der Aufbau - Master Plan

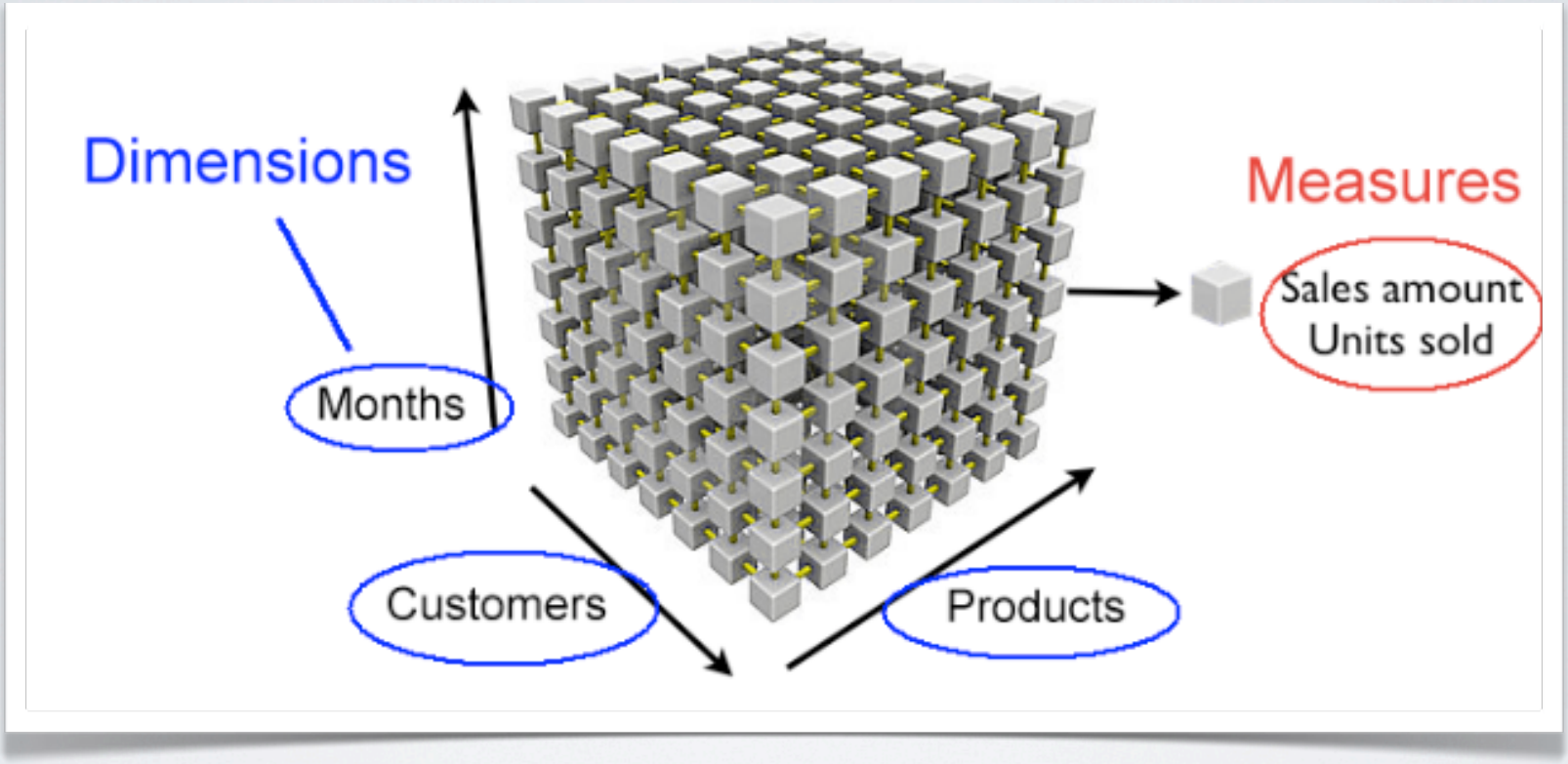
Reihenfolge der Supply Chain

Dimension

	Date	Product	Store	Promotion	Warehouse	Vendor	Contract	Shipper
Retail Sales	X	X	X	X				
Retail Inventory	X	X	X		X			
Retail Deliveries	X	X	X					
Warehouse Inventory	X	X			X	X		
Warehouse Deliveries	X	X			X	X		
Purchase Orders	X	X			X	X	X	X

Geschäftsprozess (Fakten)

Der Aufbau - Multidimensionale Ansicht / Data Marts



Der Aufbau - Mythen

Dimension Modelle und Data-Marts...

- ... sind nur für Summenwerte
- ... sind keine Lösung für das Unternehmen, sondern für die Abteilung
- ... sind nicht skalierbar
- ... sind nur anwendbar, wenn ein Anwendungsmuster erkennbar ist
- ... können nicht integriert werden und führen daher zu Flaschenhals-Lösungen

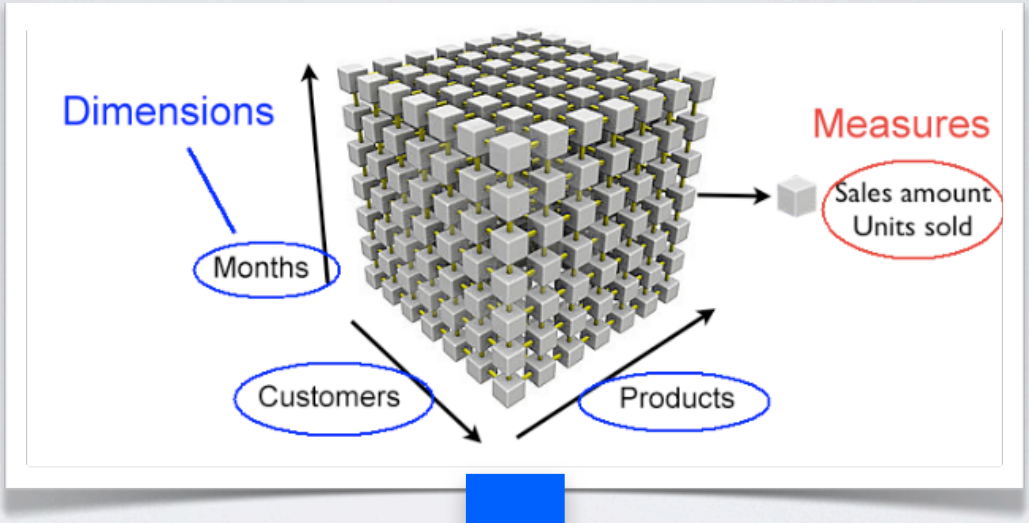
Der Aufbau - Fehlerquellen

- In die technische Lösung verliebt zu sein und die Anforderungen zu vernachlässigen
- Zu viel Energie in eine normalisierte Datenstruktur stecken
- Zu viel Aufmerksamkeit in die Performance für die Aufbereitung der Daten stecken
- Die Abfrage ist für den Endanwender zu kompliziert
- Dem Endanwender werden nur summierte Werte zur Verfügung gestellt
- Es wird angenommen die zugrundeliegende Daten seien statisch
- Felder sind „NULL“ anstatt mit dem Wert 0 beschrieben
- Das Data Warehouse wird von den Endanwendern nicht akzeptiert

Der Aufbau - 4 Schritte zum Design

1. **Auswahl des Geschäftsprozesses**
   Was ist dem Unternehmen wichtig? In welchem Bereich gibt es Probleme?
2. **Detaillierungsgrad definieren**
   Wie beschreibe ich eine einzelne Zeile in der Fact-Tabelle?
3. **Dimensionen wählen**
   Wie beschreiben Anwender die Daten, die aus dem Geschäftsprozess kommen?
4. **Fakten wählen**
   Was messen wir? (Kosten, Einnahmen, Stückzahlen, usw.)

Der Aufbau - Fact und Dimension Tabellen zusammenführen



Date Dim
Date Key (PK)
Month
Year

Customer Dim
Customer Key (PK)
Name
Address

Sales Fact
Date Key (FK)
Customer Key (FK)
Product Key (FK)
Sales amount
Units sold

Product Dim
Product Key (PK)
Name
Description

Dimension Tabellen - Slowly Changing Dimensions

Der Außendienstmitarbeiter „Homer Simpson“ vertreibt Duff Bier in Ost-Deutschland. Mit 1.1.2014 wird er den Bereich West-Deutschland übernehmen. Wie wird diese Änderung im Data Warehouse berücksichtigt?

Type I

ID	Region	Account Manager
381	West	Homer Simpson

Ehemalige Zuordnungen gehen verloren!

Type II

ID	Region	Account Manager	Date
381	Ost	Homer Simpson	1.6.2011
382	West	Homer Simpson	1.1.2014

Type III

ID	Region	Old Region	Account Manager	Date
381	West	Ost	Homer Simpson	1.1.2014

Type VI

ID	Region	Old Region	Account Manager	Date
381	West	Ost	Homer Simpson	1.1.2014
382	West	West	Homer Simpson	1.1.2014

Kombination aus Type I + II + III = VI

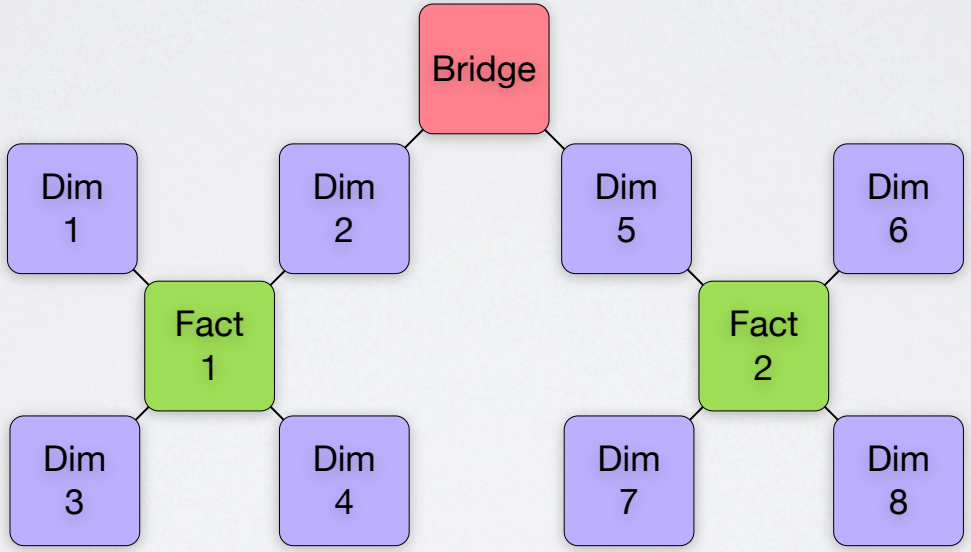
Dimension Tabellen - Junk Dimensions

- Speichert Daten, die in keinem Zusammenhang stehen
- Werden zum Filtern und Gruppieren verwendet
- Jede Kombinationsmöglichkeit muss vorhanden sein

OrderID	Payment Type	Order Indicator	Commission Credit Indicator
1	Cash	Inbound	Commissionable
2	Cash	Inbound	Non-Commissionable
3	Cash	Outbound	Commissionable
4	Cash	Outbound	Non-Commissionable
5	Credit	Inbound	Commissionable
6	Credit	Inbound	Non-Commissionable
7	Credit	Outbound	Commissionable
8	Credit	Outbound	Non-Commissionable

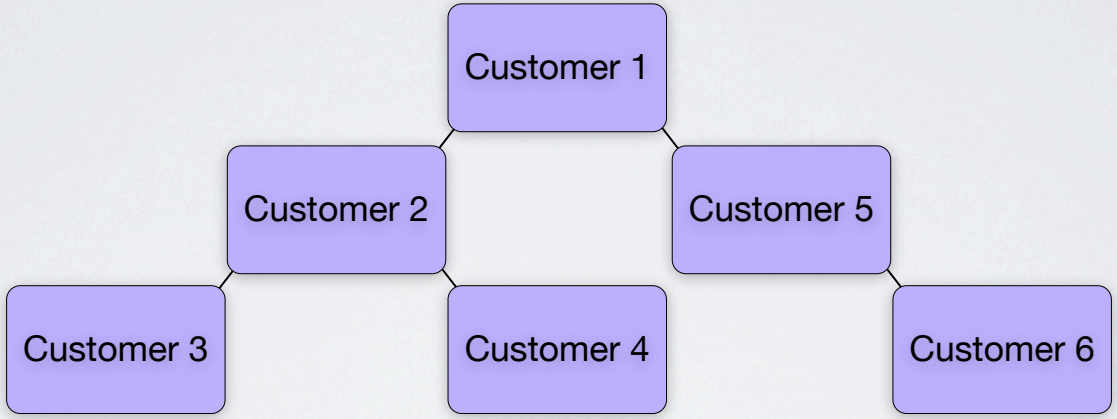
Dimension Tabellen - Bridges

- Verknüpft Fact Tabellen über zwei Dimension Tabellen miteinander
- Für Fact Tabellen-übergreifende Abfragen
- Forciert das Snowflake-Schema = schlechtes Datenbankdesign!



Dimension Tabellen - Bridges

- Verknüpft Fact Tabellen über zwei Dimension Tabellen miteinander
- Für Fact Tabellen-übergreifende Abfragen
- Forciert das Snowflake-Schema = schlechtes Datenbankdesign!



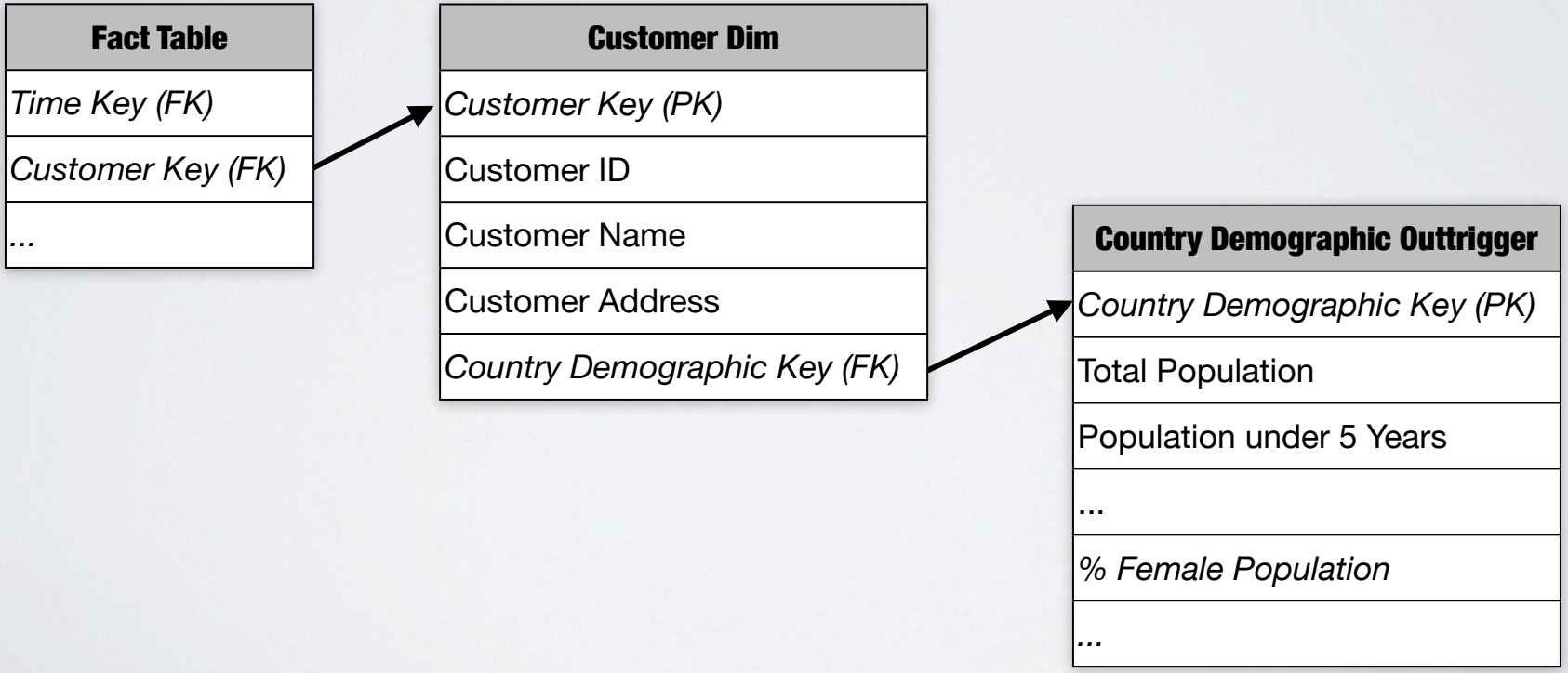
Customer Dimension
Customer Key (PK)
Customer ID
Customer Name
Customer Address

Customer Hierachie Bridge
Parent Customer Key (PK)
Subsidiary Customer Key (PK)
Level Name
Buttom Flag
Top Flag

Fact Table
Date Key (FK)
Customer Key (FK)
...

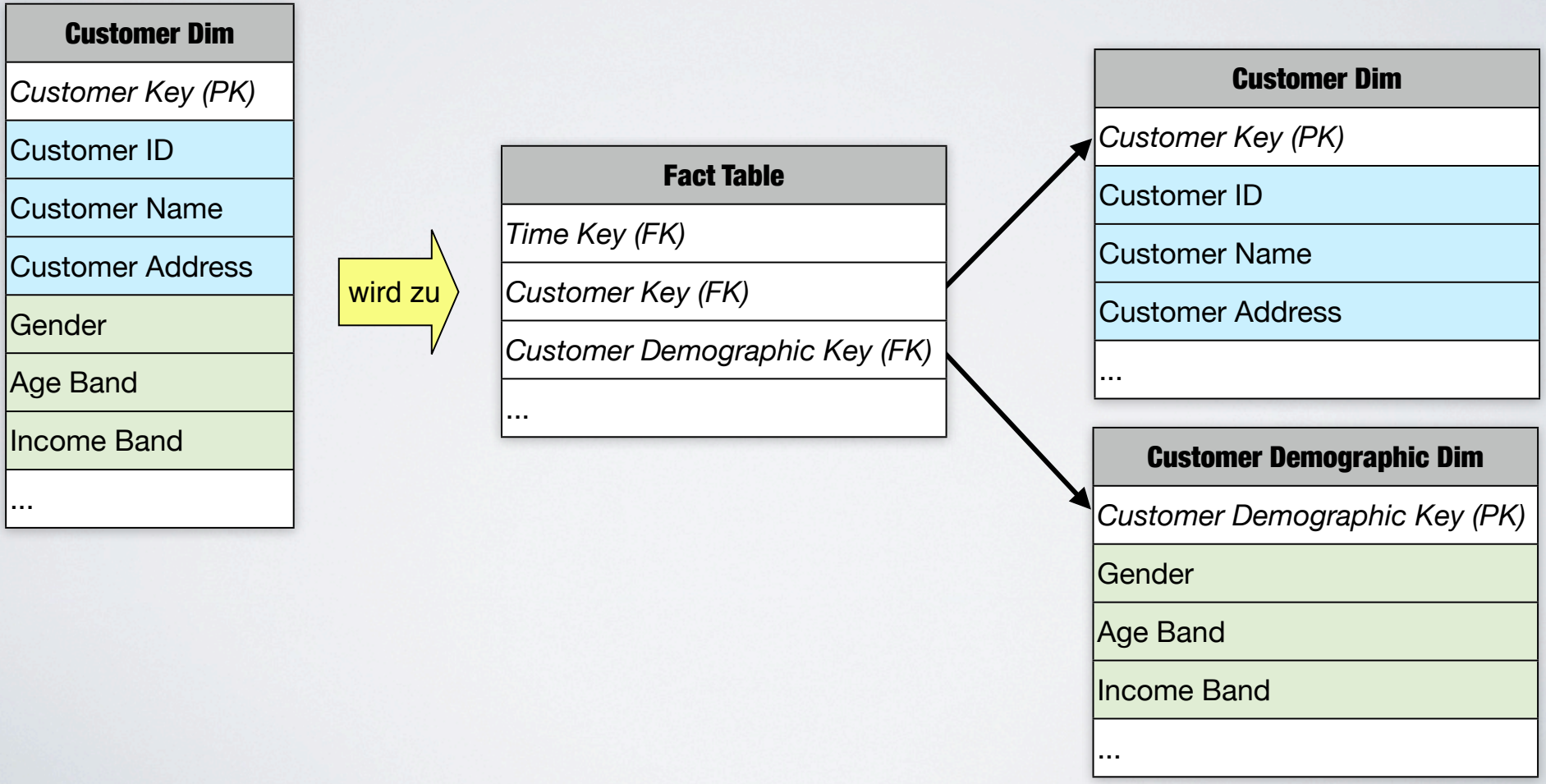
Dimension Tabellen - Outtrigger

- Join zu einer Dimension Tabelle
- Vermeidet Redundanzen und spart Speicherplatz
- Forciert das Snowflake-Schema = schlechtes Datenbankdesign!



Dimension Tabellen - Minidimensions

- Join zu einer Fact Tabelle
- Auch für Daten, die sich sehr schnell ändern



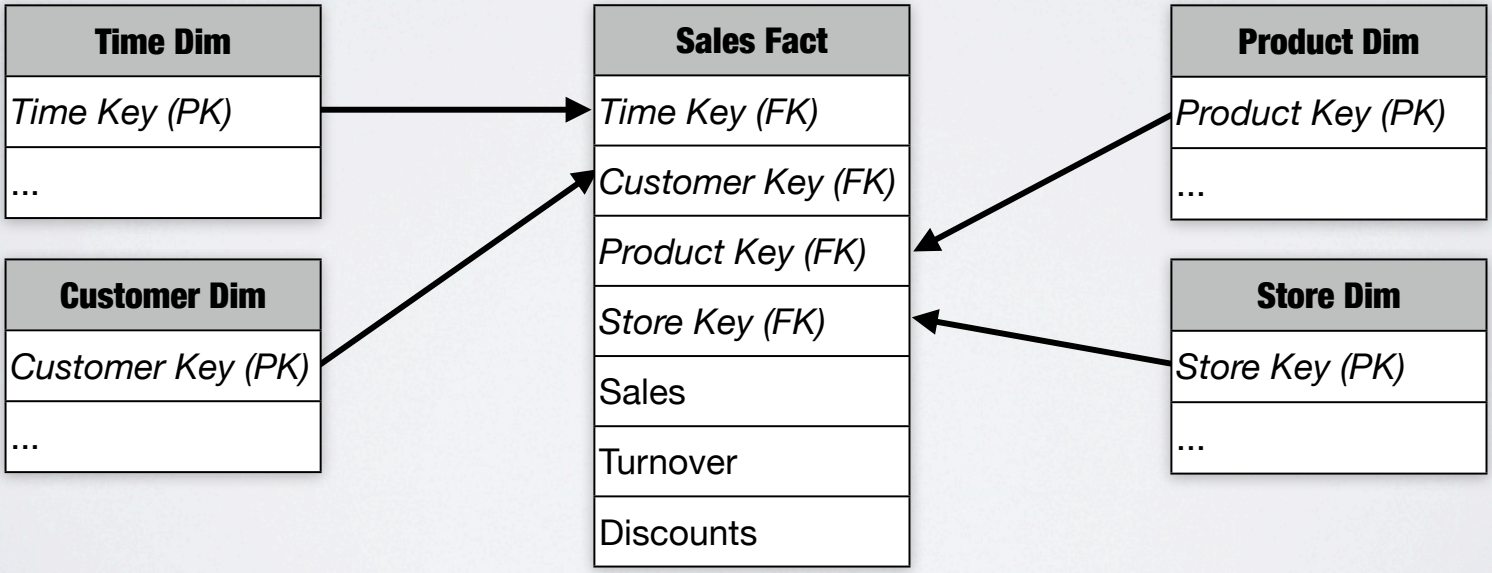
Fact Tabellen - Unterschiede

Die Beladung mit Daten richtet sich nach Art der Tabelle. Daher unterscheidet man zwischen:

- Transactional Fact Tables
- Snapshot Tables
- Accumulated Tables
- Factless Fact Tables

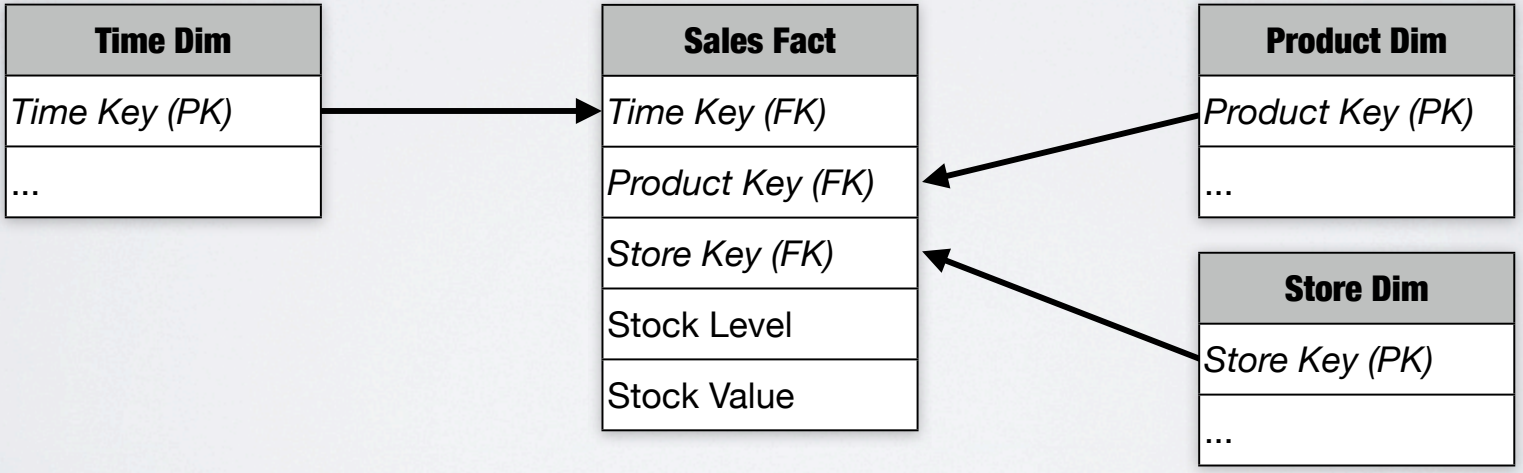
Fact Tabellen - Transactional Fact Tables

- Bilden einen präzisen Moment ab
- Werden für alle Arten von Transaktionen verwendet
- Eine Zeile in der Fact Tabelle entspricht einer Transaktion



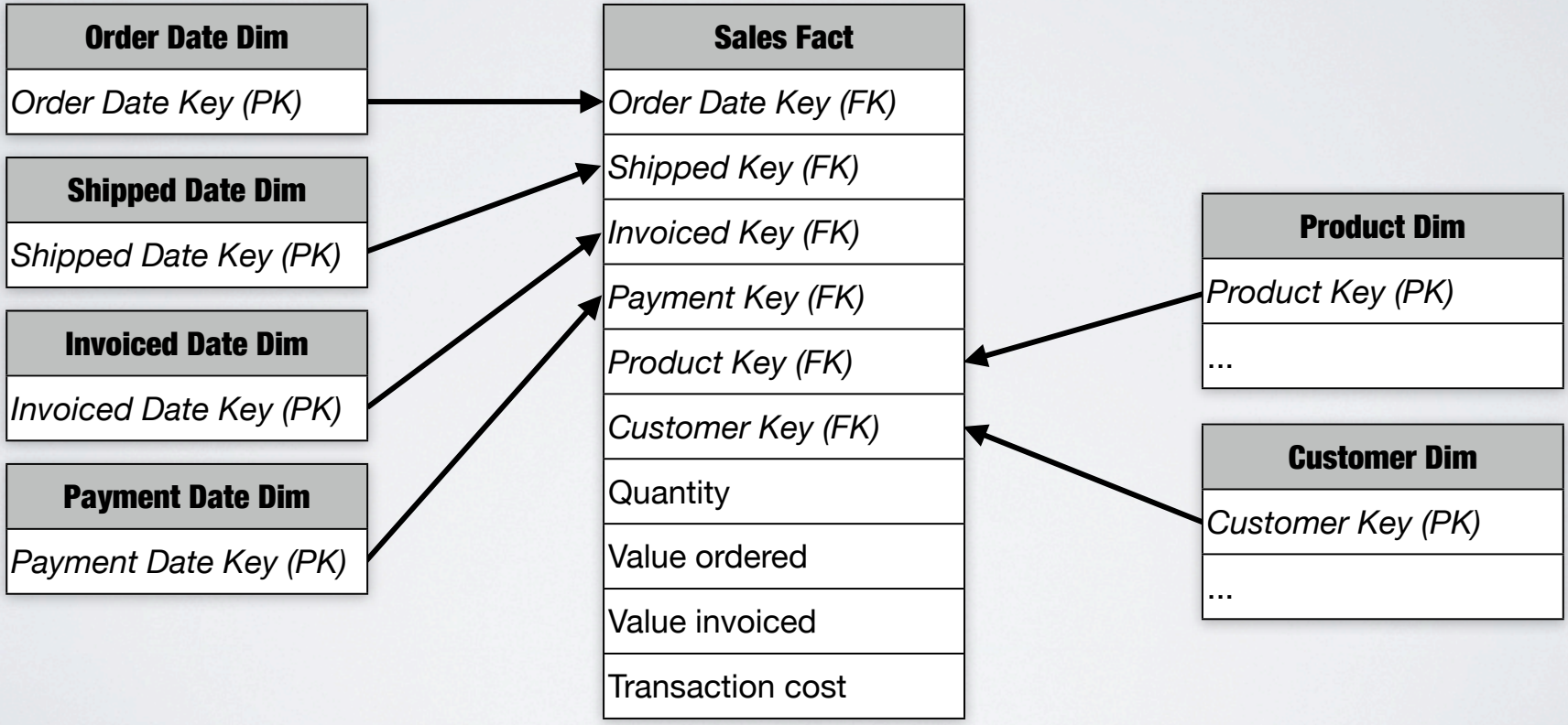
Fact Tabellen - Snapshot Tables

- Erstellen ein Abbild eines Momentes
- Semi-Additiv
  (Beispiel: Lagerbestände sind nicht-additiv über die Zeit, aber additiv über Niederlassungen für einen bestimmten Zeitraum)



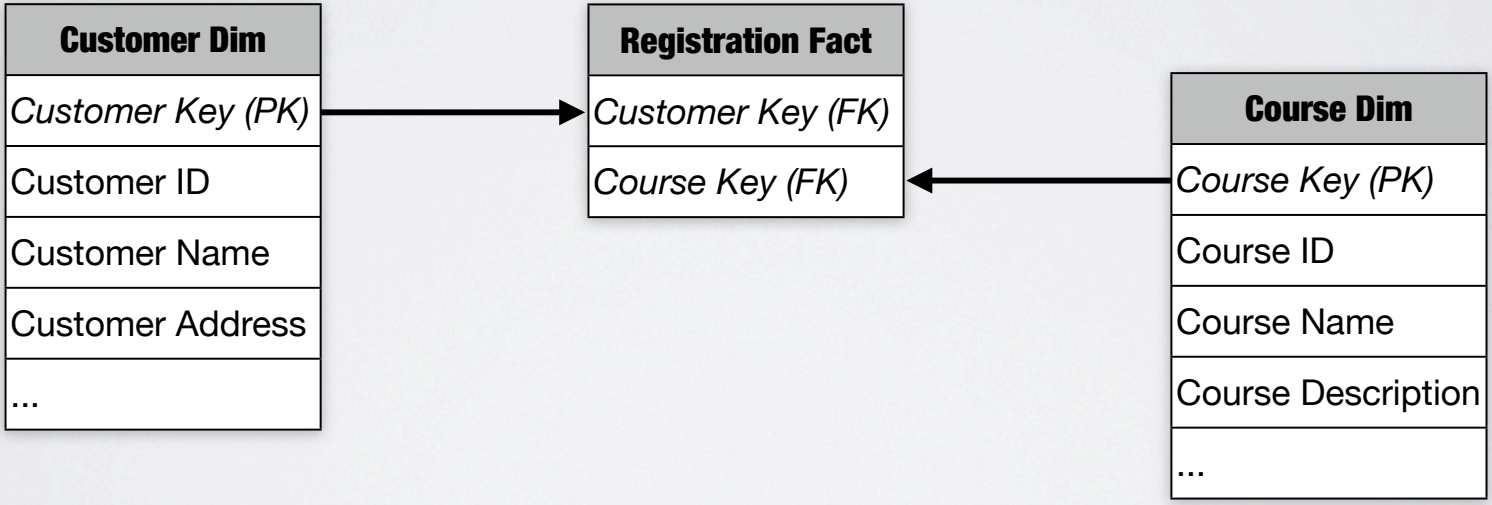
Fact Tabellen - Accumulated Tables

- Bildet einen Prozess ab *(Beispiel: Der Prozess einer Bestellung. Mit einer Fact Tabelle wird der Status über die Zeit erfasst)*

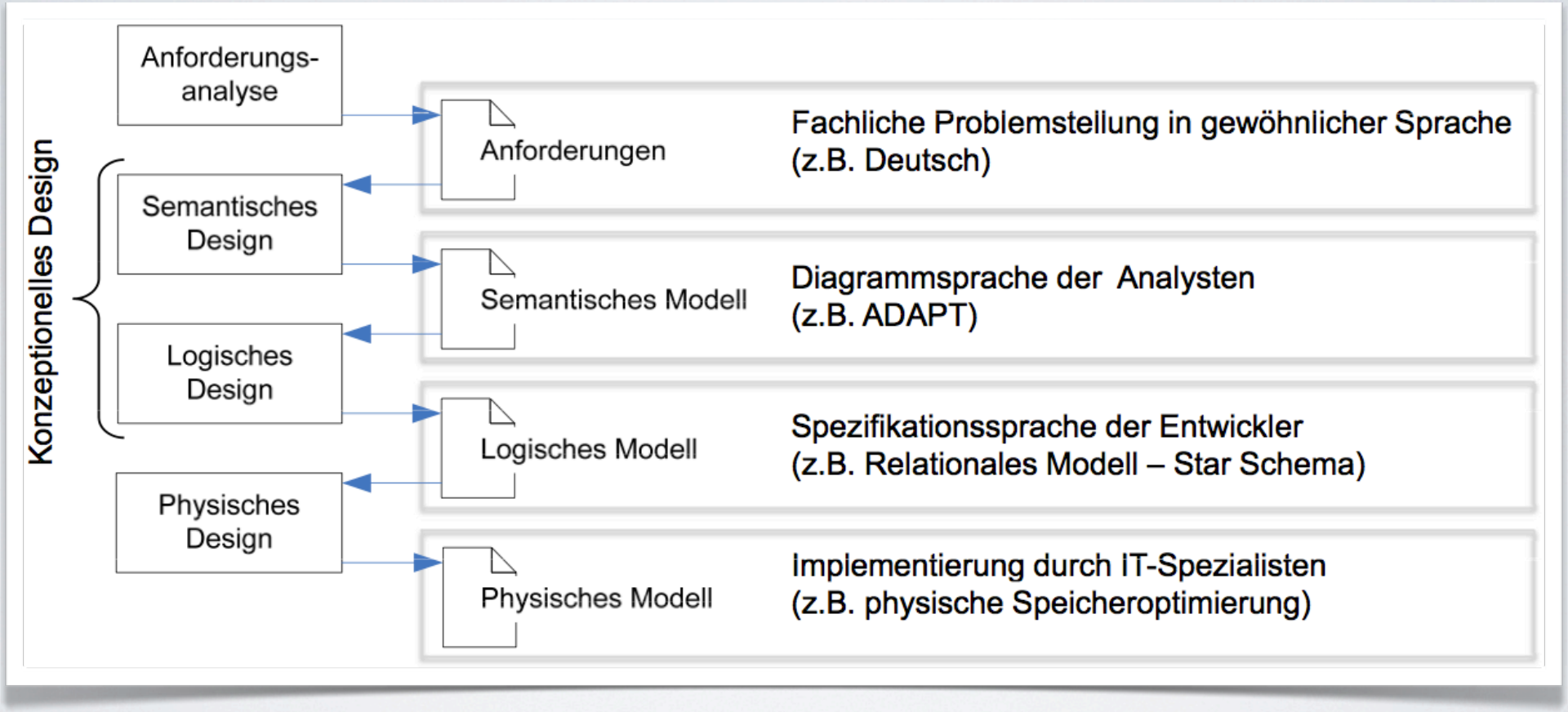


Fact Tabellen - Factless Fact Tables

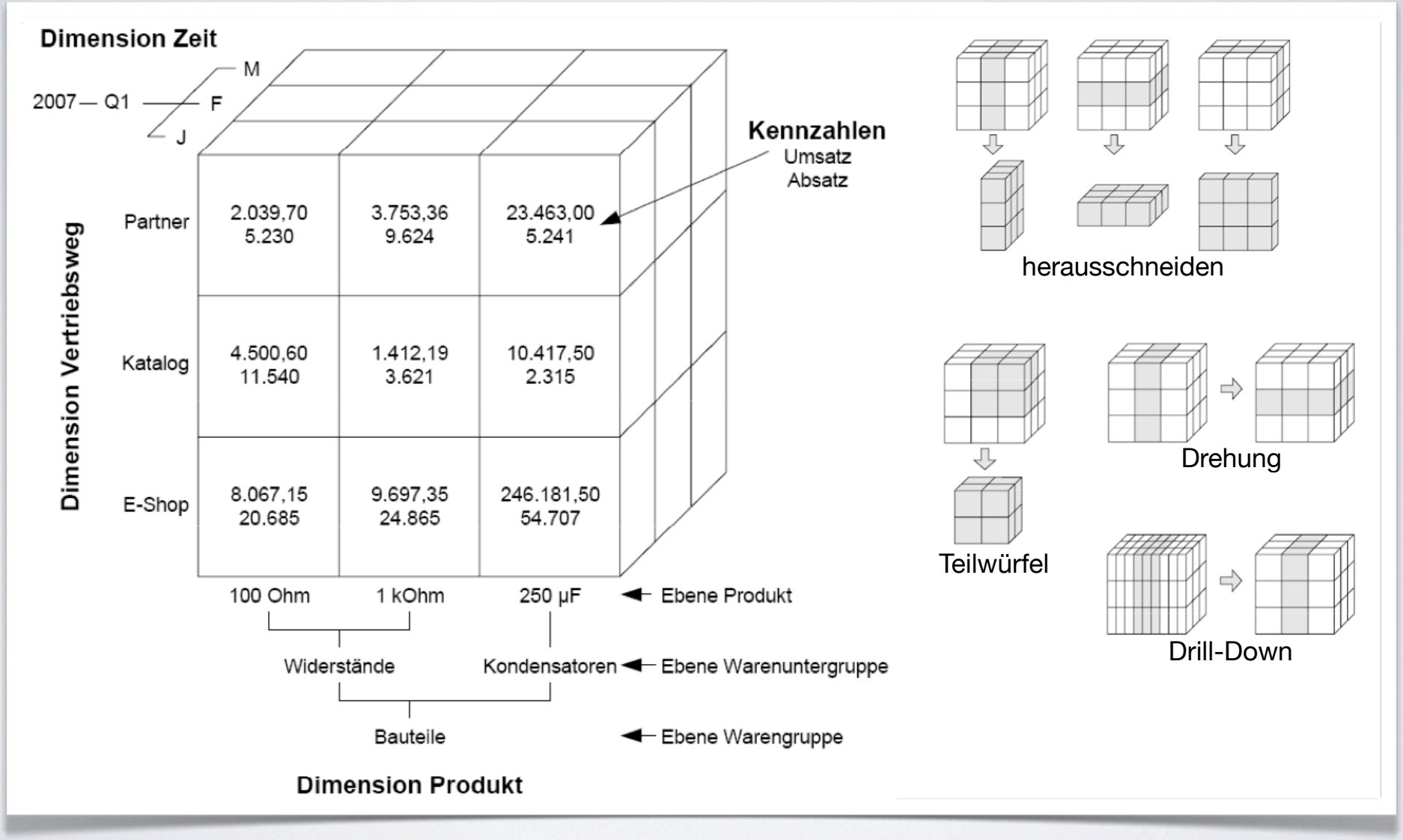
- Beinhalten keine Daten
- Dienen zum Zählen von Ereignissen



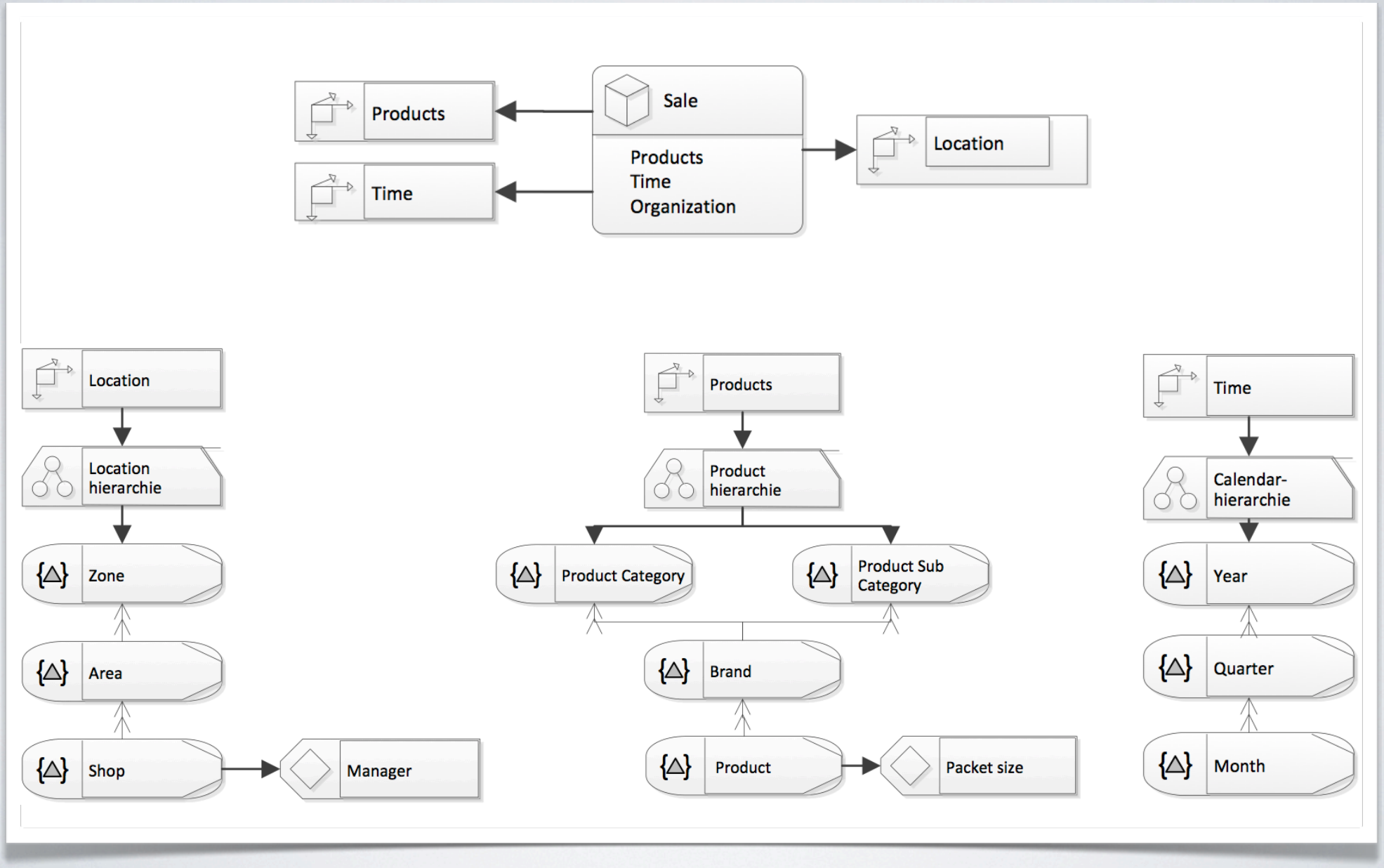
Modellierung



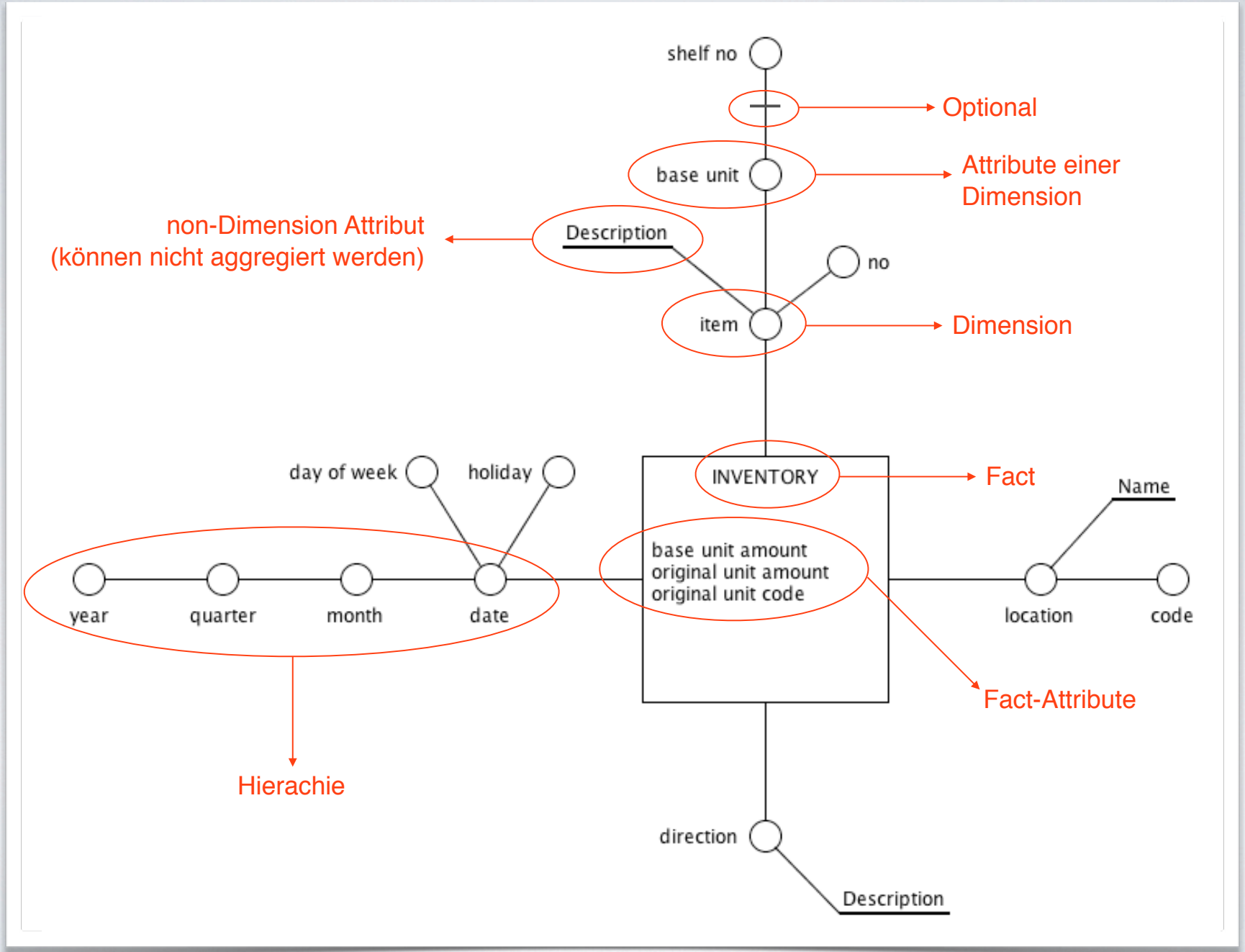
Modellierung - Datenwürfel



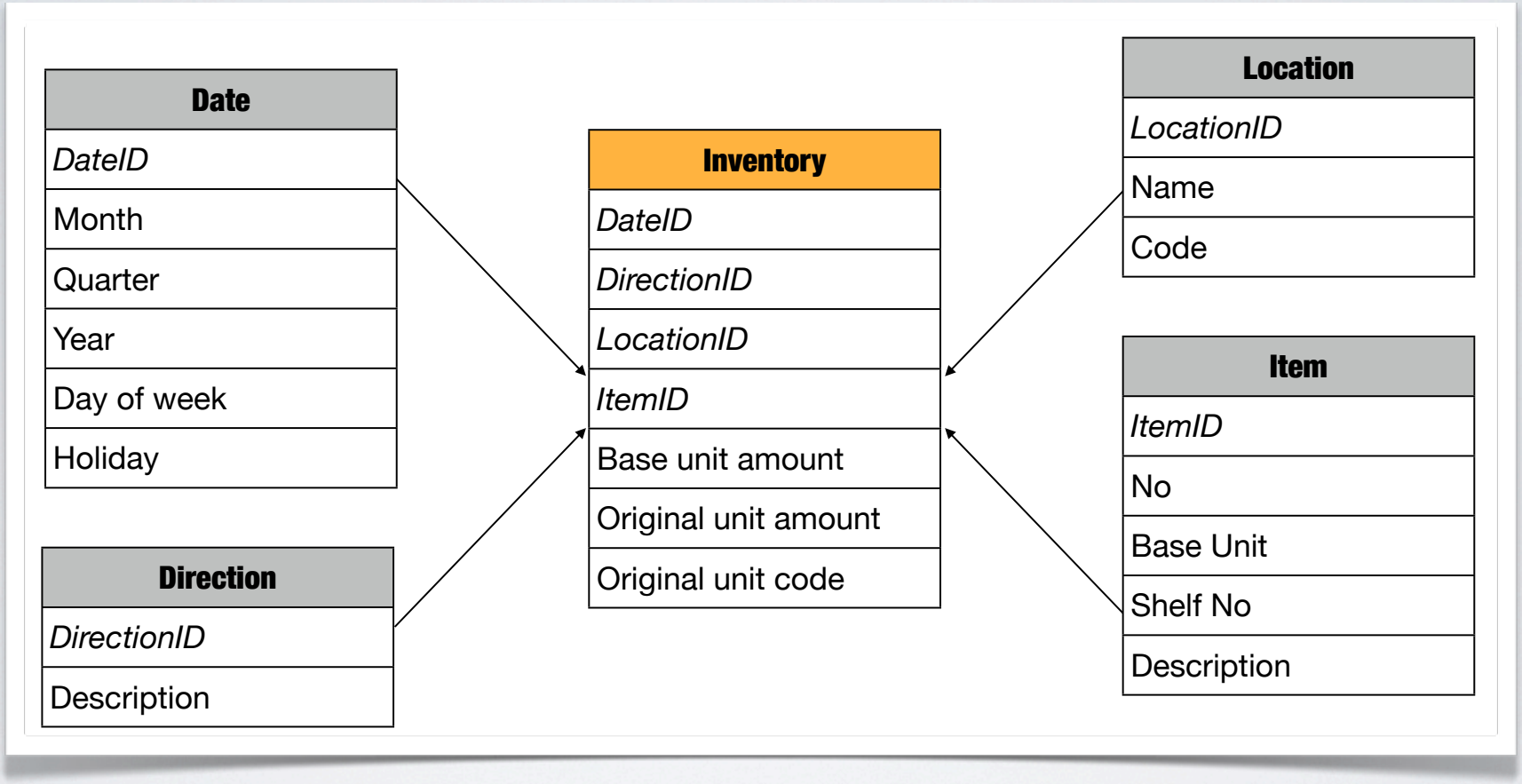
Modellierung - ADAPT *(Application Design for Analytical Techniques)*



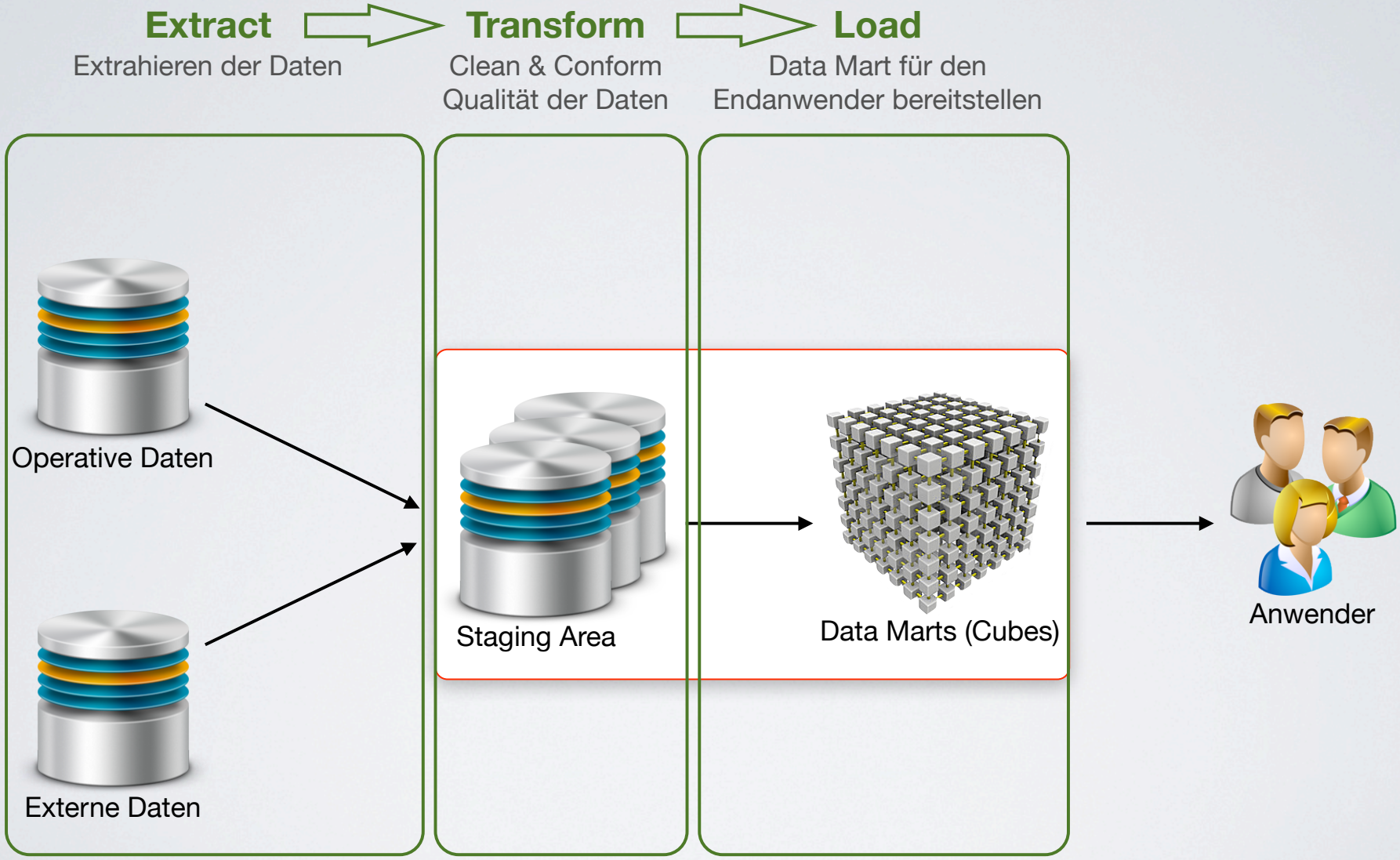
Modellierung - DFM *(Dimensional Fact Modelling)*



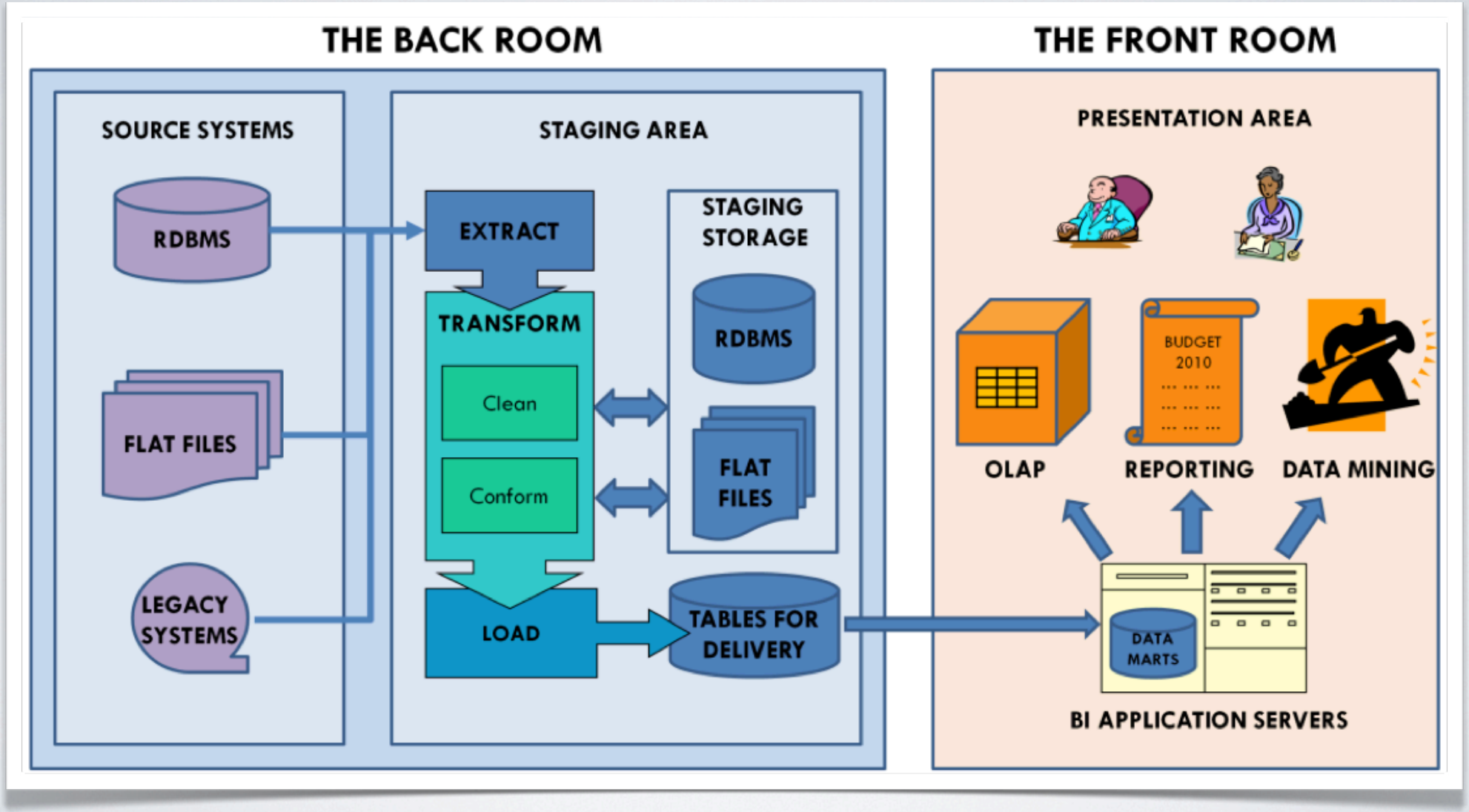
Modellierung - Star-Schema



Der ETL-Prozess



Der ETL-Prozess



Ralph Kimball

Der ETL-Prozess - Extract Process

Extract | Transform (Clean | Conform) | Load

1. **Logical Data Map (LDM) erstellen**
   Gibt an, woher die Daten kommen

Target					Source				Transformation
Table Name	Column Name	Data Type	Table Type	SCD Type	Database Name	Table Name	Column Name	Data Type	

2. **Zu den Quellen verbinden**
   Verbindung zu den Source-Systemen

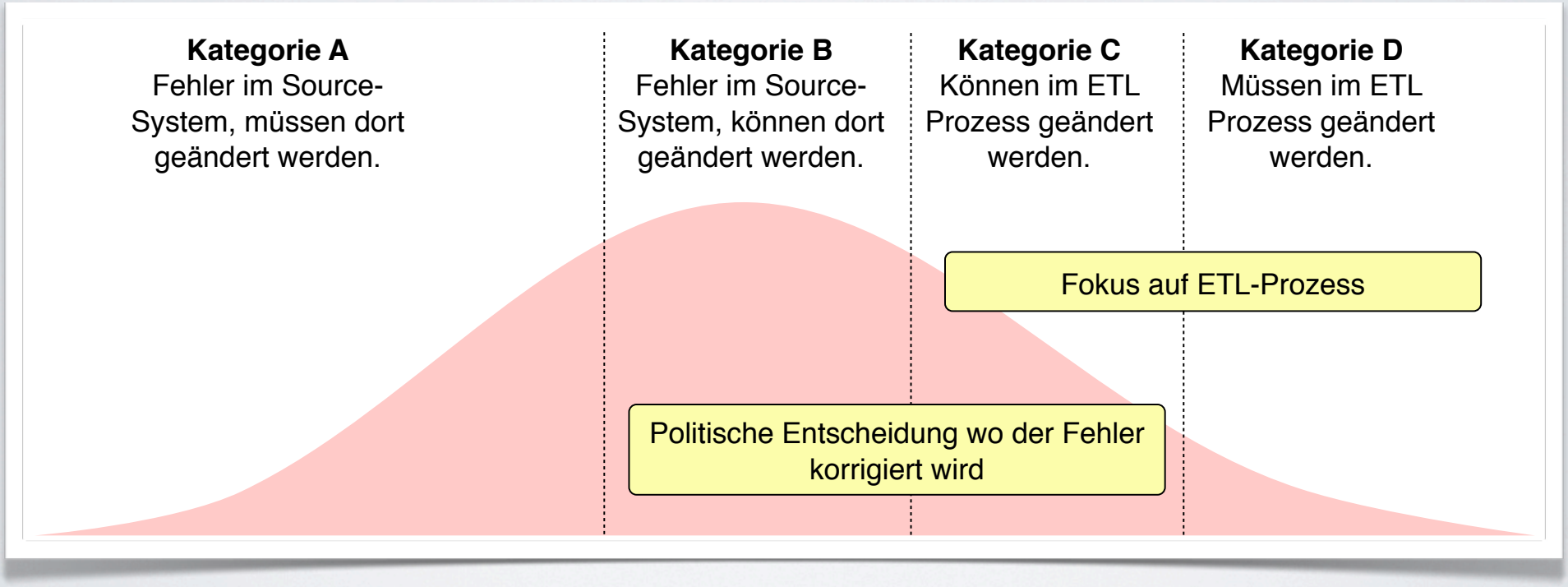
3. **Aktualisierungszeitraum festlegen**
   Täglich, Wöchentlich, Monatlich...?

4. **Änderungen ermitteln**
   Nur neue, geänderte oder gelöschte Daten mittels Timestamps oder Vergleich mit letzter Beladung erfassen

5. **Staging**
   Schrittweise und Nachvollziehbar - Recoverypunkte

Der ETL-Prozess - Transform Process

- Ziel ist es, die Datenqualität sicherzustellen
- Daten müssen konkret, unmissverständlich, gültig, konsistent und komplett sein
- zwei Schritte: Cleaning Deliveries & Conforming Data



Der ETL-Prozess - Transform Process

Extract | Transform | Clean | Conform | Load

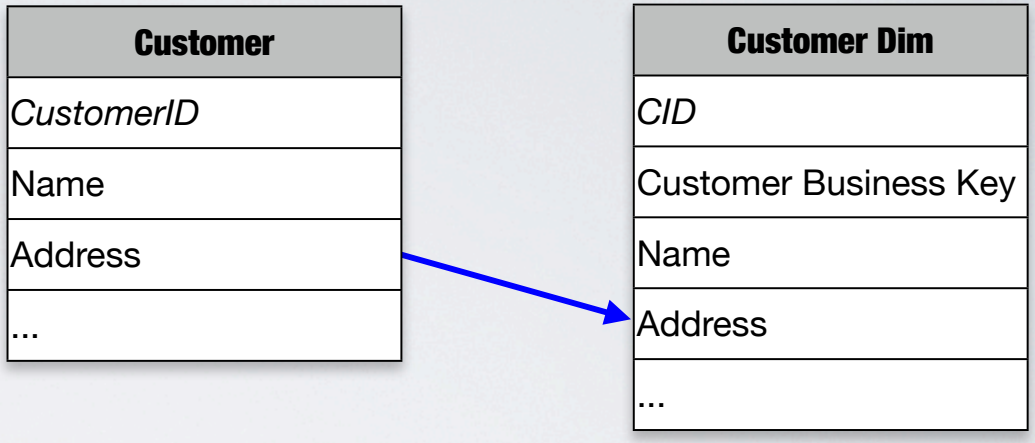
1. **Cleaning Deliverables**
   Auf Fehler prüfen - Screens
   Screening Prozess wird dokumentiert - Audits

City	Count(*)
München	124245
Münhen	2

2. **Conforming Data**
   Hat zum Ziel, inkonsistente Daten zu entfernen oder aufzulösen
   - *Vereinheitlichen von Bezeichnungen („Jänner“ - „Januar“)*
   - *Redundante Daten aus Dimensionen entfernen („M. Müller“ - „Max Müller“)*
   - *Auseinandergehende Bezeichnungen zusammenführen (Artikelbezeichnung in Verkauf und Produktion)*

Der ETL-Prozess - Load Process

1. **Dimension: neue Werte**
   Ein neuer Surrogate Key muss gefunden werden (z.B. AutoInc)



2. **Dimension: aktualisierte Werte**
   Auf geänderte Werte prüfen - Slowly Changing Dimensions anwenden

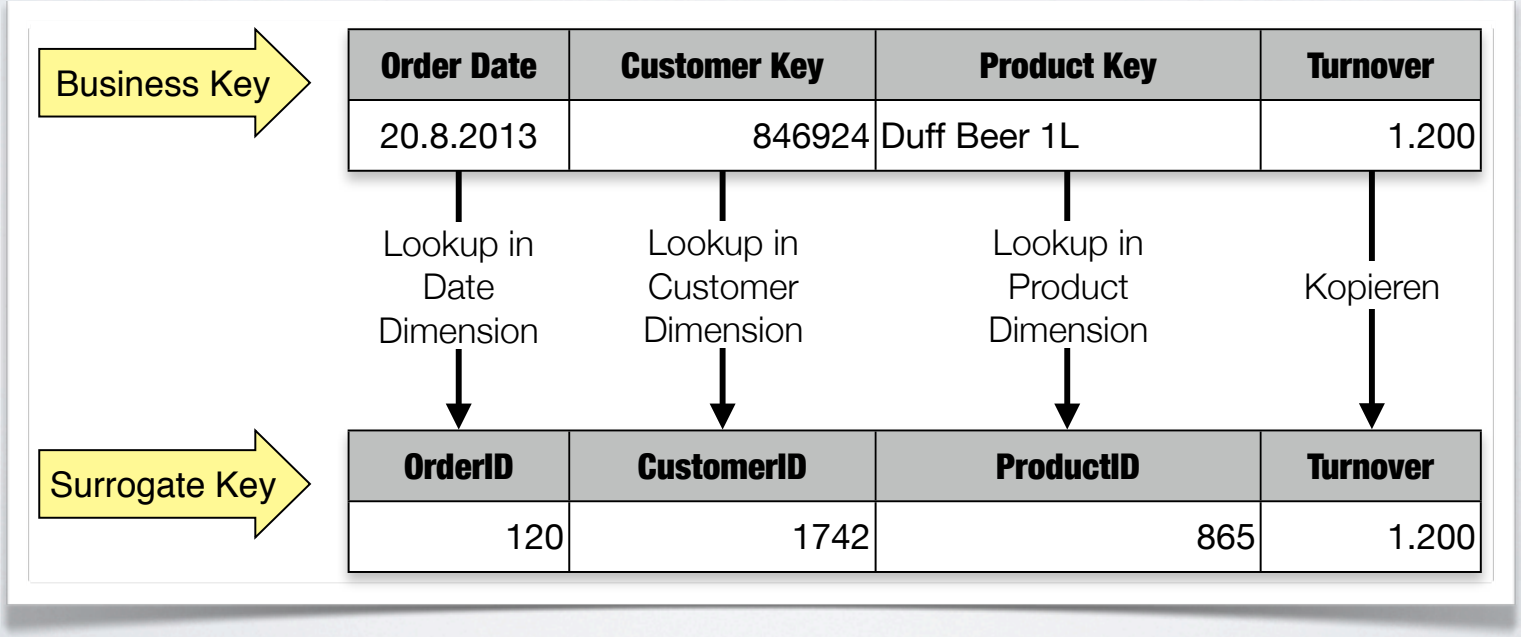
3. **Dimension: gelöschte Werte**
   Werden im Normalfall beibehalten, da sie für die History wichtig sind. Nur verwaiste Einträge werden gelöscht.

Der ETL-Prozess - Load Process

Extract | Transform (Clean | Conform) | Load

4. **Fact: Tabellen vorbereiten**
   Unter Berücksichtigung der unterschiedlichen Arten von Fact Tabellen:
   - *Transactional Facts: für jede Transaktion eine Zeile erstellen*
   - *Snapshot Facts: Für jeden Moment, der erfasst werden soll, einen Eintrag erstellen. Auch wenn er 0 ist!*
   - *Accumulated Snapshot Facts: Wenn neu hinzufügen, andernfalls Werte aktualisieren*

5. **Fact: Schlüssel ersetzten**
   Über LookUps den Primärschlüssel der Dimension Tabelle als Fremdschlüssel speichern.



Anbieter für die Entwicklung eines Data Warehouses

- Oracle Warehouse Builder
- SAP Data Integrator
- IBM Decision Stream
- Microsoft Integration Service
- Pentaho Kettle (OpenSource)

Beispiele

- Beispiel mit Microsoft Business Intelligence (SSIS)
- Auswertungen mit Delphi
- Aufwerten der Analysen mit Statistiktools in Delphi

Danke