

# Information Engineering 2

## Dimensionale Datenmodellierung 1

Prof. Dr. Kurt Stockinger

# Semesterplan (Start um 12:10)

SW	Datum	Vorlesungsthema	Praktikum
1	23.02.2022	Data Warehousing Einführung	Praktikum 1: KNIME Tutorial
2	02.03.2022	Dimensionale Datenmodellierung 1	Praktikum 1: KNIME Tutorial (Vertiefung)
3	09.03.2022	Dimensionale Datenmodellierung 2	Praktikum 2: Datenmodellierung
4	16.03.2022	Datenqualität und Data Matching	Praktikum 3: Star-Schema, Bonus: Praktikum 4: Slowly Changing Dimensions
5	23.03.2022	Big Data Einführung	DWH Projekt - Teil 1
6	30.03.2022	Spark - Data Frames	DWH Projekt - Teil 2 (Abgabe: 4.4.2022 23:59:59)
7	06.04.2022	Data Storage: Hadoop Distributed File System & Parquet	Praktikum 1: Data Frames
8	13.04.2022	Query Optimization	Praktikum 2: Data Storage
9	20.04.2022	Spark Best Practices & Applications	Praktikum 3: Query Optimization & Performance Analysis
10	27.04.2022	Machine Learning mit Spark 1	Praktikum 3: Query Optimization & Performance Analysis (Vertiefung)
11	04.05.2022	Machine Learning mit Spark 2 + Q&A	Praktikum 4: Machine Learning (Regression)
12	11.05.2022	NoSQL Systems	Big Data Projekt - Teil 1
13	18.05.2022	Keine Vorlesung (Arbeit am Projekt)	Big Data Projekt - Teil 2
14	25.05.2022	Keine Vorlesung (Arbeit am Projekt)	Big Data Projekt - Teil 3 (Abgabe: 30.5.2022 23:59:59)

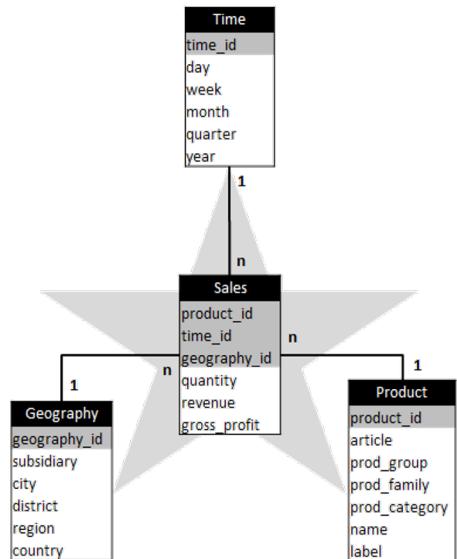
# Agenda

## Dimensionale Modellierung

- Use Case → Sternschema → Hierarchien → Die Zeitdimension

## Lernziele

- Kennen von
  - Kennzahlen, **Fakten** und **Dimensionen**
  - Verschiedene Modellierungsvarianten
- Verstehen von
  - Modellierung von **Sternschemata**
  - Vorteile dieses Schemas
- Beherrschen von Grundlagen der **dimensionalen Datenanalyse**



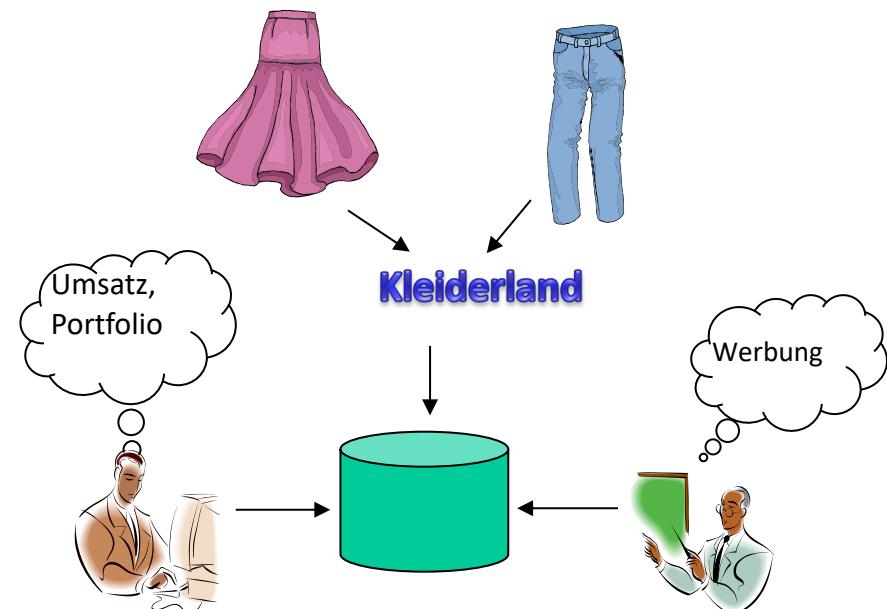
# Wiederholung

- Was ist ein DWH?
- Welche Schichten beinhaltet die DWH-Referenzarchitektur?



# Use Case: Marketing Analyse

- Ausgangslage:
  - Eine Firma möchte ein Produkt lancieren  
→ Abklärung: welche potentiellen Kunden würden das Produkt kaufen?
- Ansatz:
  - Erstellen eines **Marketing Data Marts** mit Informationen über
    - Produkte
    - Kunden
    - Transaktionen (Kaufverhalten)
  - Neues Produkt wird mit ähnlichen Produkten verglichen
  - Kunden, die ähnliches Produkt gekauft haben, werden analysiert



# Use Case: Ein mögliches Ergebnis

View 1 ▾

			Continent Name	+□ Europe	+□ North America	Total
Year	All ProductLine	Product Line	Measures			
+□ 1998	+□ All ProductLine	+□ Children	Sum of Quantity	17145	4466	21611
		+□ Clothes & Shoes	Sum of Quantity	69319	18803	88122
		+□ Outdoors	Sum of Quantity	26707	9255	35962
		+□ Sports	Sum of Quantity	63322	33952	97274
		<b>Subtotal: 1998</b>	Sum of Quantity	176493	66476	242969
+□ 1999	+□ All ProductLine	+□ Children	Sum of Quantity	20685	6126	26811
		+□ Clothes & Shoes	Sum of Quantity	80613	23161	103774
		+□ Outdoors	Sum of Quantity	29585	10025	39610
		+□ Sports	Sum of Quantity	75330	37568	112898
		<b>Subtotal: 1999</b>	Sum of Quantity	206213	76880	283093
+□ 2000	+□ All ProductLine	+□ Children	Sum of Quantity	23768	7079	30847
		+□ Clothes & Shoes	Sum of Quantity	93130	30930	124060
		+□ Outdoors	Sum of Quantity	36419	12908	49327
		+□ Sports	Sum of Quantity	91588	46164	137752
		<b>Subtotal: 2000</b>	Sum of Quantity	244905	97081	341986
+□ 2001	+□ All ProductLine	+□ Children	Sum of Quantity	21381	5545	26926
		+□ Clothes & Shoes	Sum of Quantity	78355	23414	101769
		+□ Outdoors	Sum of Quantity	33080	10995	44075
		+□ Sports	Sum of Quantity	83620	34859	118479
		<b>Subtotal: 2001</b>	Sum of Quantity	216436	74813	291249
+□ 2002	+□ All ProductLine	+□ Children	Sum of Quantity	27660	7613	35273
		+□ Clothes & Shoes	Sum of Quantity	90474	23584	114058
		+□ Outdoors	Sum of Quantity	36698	13584	50282
		+□ Sports	Sum of Quantity	100489	39789	140278
		<b>Subtotal: 2002</b>	Sum of Quantity	255321	84570	339891
<b>Total</b>			Sum of Quantity	1099368	399820	1499188

# Methode: Dimensionale Modellierung

- Modellierung für **OLAP-Systeme (Data Marts)**:
  - Abfrageoptimierung
  - Keine Updates
  - Daten werden periodisch geladen (z.B. täglich, monatlich)
- Wiederholungsfrage:
  - Was ist der Unterschied zwischen **OLTP** und **OLAP**?

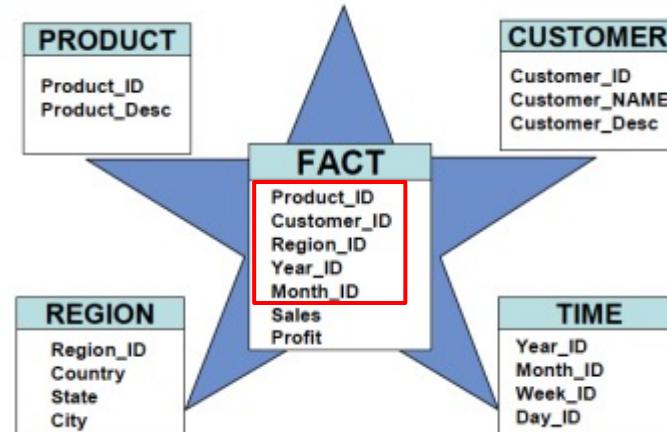


# Sternschema

Kennzahlen, Fakten, Dimensionen

# Sternschema

- Kombination von Fakten (inkl. Kennzahlen) und Dimensionen

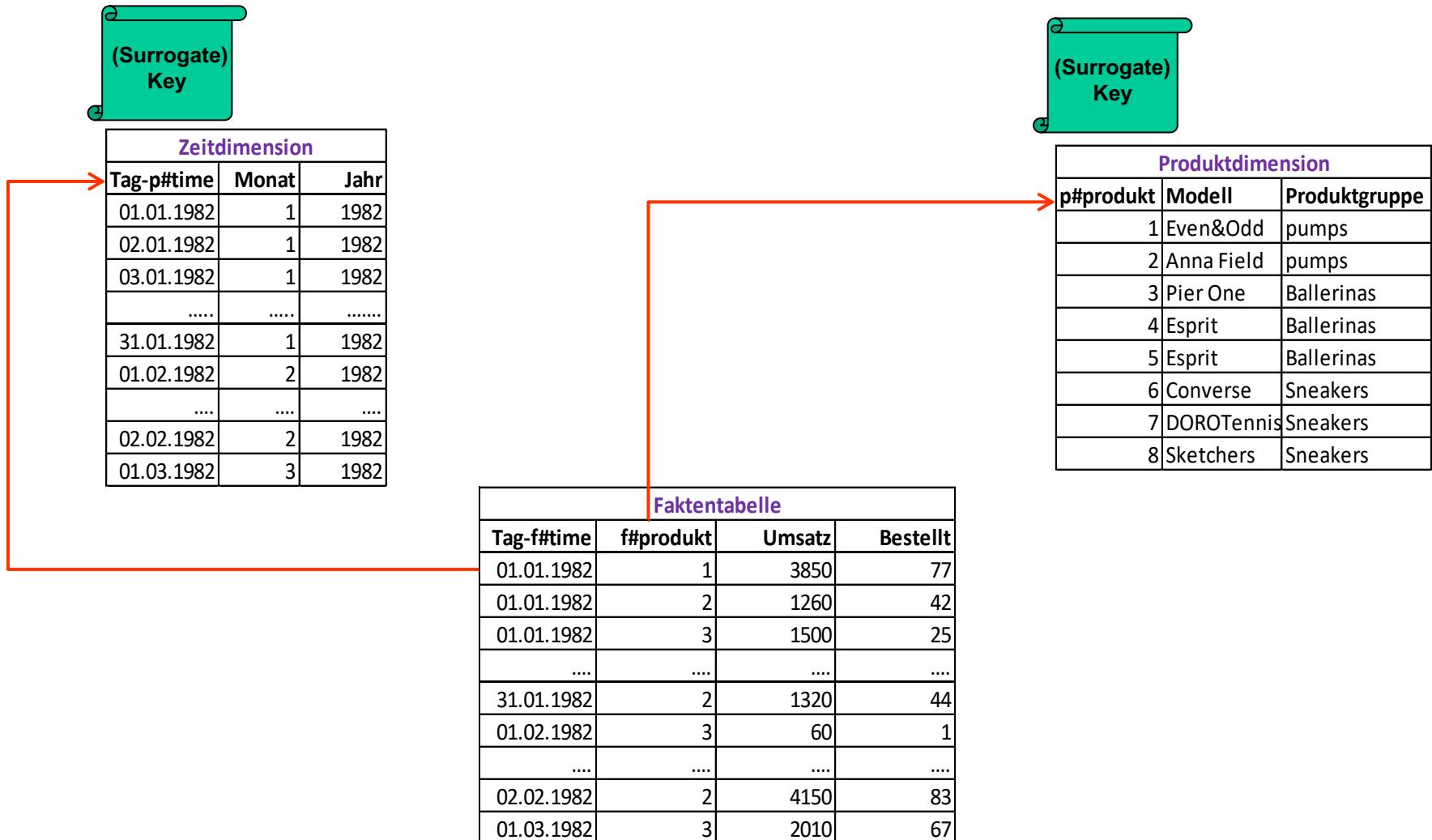


Alternative Zeitdarstellung?

Quelle: IBM Redbook – Data Mart Consolidation

- **Faktentabelle:**
  - Enthält **Fremdschlüssel (Foreign-Key)** auf Dimensionstabellen
- **Dimensionstabellen:**
  - Gelten als Masterdaten
  - Beinhalten die **Primärschlüssel (Primary-Key)**, die mit Faktentabelle “verknüpft” werden

# Ein Beispiel



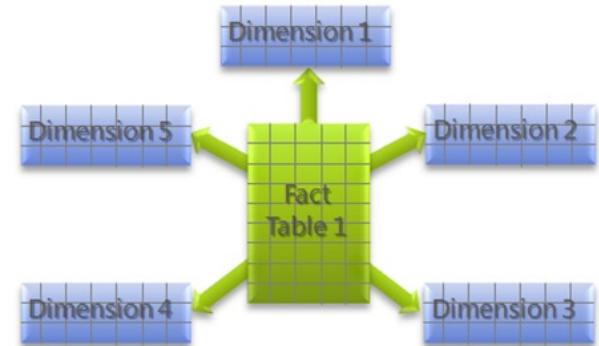
# Vorteil des Sternschemas

- Höhere **Abfragegeschwindigkeit**
  - da weniger und einfachere Joins benötigt werden
- Grund: **Denormalisierung** der Daten («Dimensionen», s.u.)
  - Nicht 3NF, Daten werden innerhalb einer oder mehrerer Tabellen dupliziert
  - Rückblick Definition 3. Normalform (3NF):  
*«Eine Tabelle ist in 3NF, wenn jede Nicht-Schlüssel-Spalte von allen anderen Nicht-Schlüssel-Spalten unabhängig ist»*
- Beispiel: Region

RegionID	Country	State	City
1	Switzerland	Zurich	Winterthur
2	Switzerland	Zurich	Zurich

# Fakten

- Ansammlung von **Einzelmessungen / Kennzahlen**, verknüpft mit **Kontextdaten** (Dimensionen)
- Jedes Faktum repräsentiert einen Businesswert, eine Transaktion oder ein Ereignis
- Beispiel:
  - Anzahl der Verkäufe (Fakt)
    - für ein bestimmtes Produkt (Dimension)
    - für eine bestimmte Region (Dimension)
    - für einen bestimmten Zeitraum (Dimension)
- Typische Faktentabelle:
  - **Wenige, meist numerische Attribute, aber viele Datensätze**



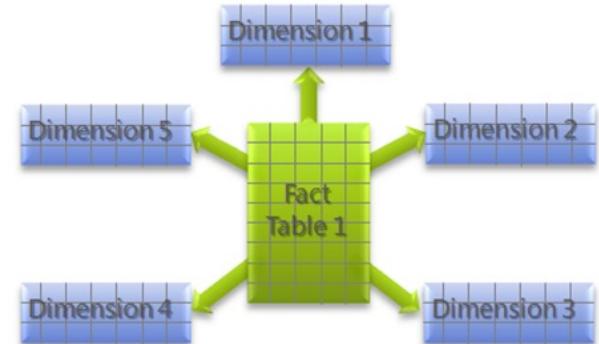
# Kennzahlen: spezielle Fakten

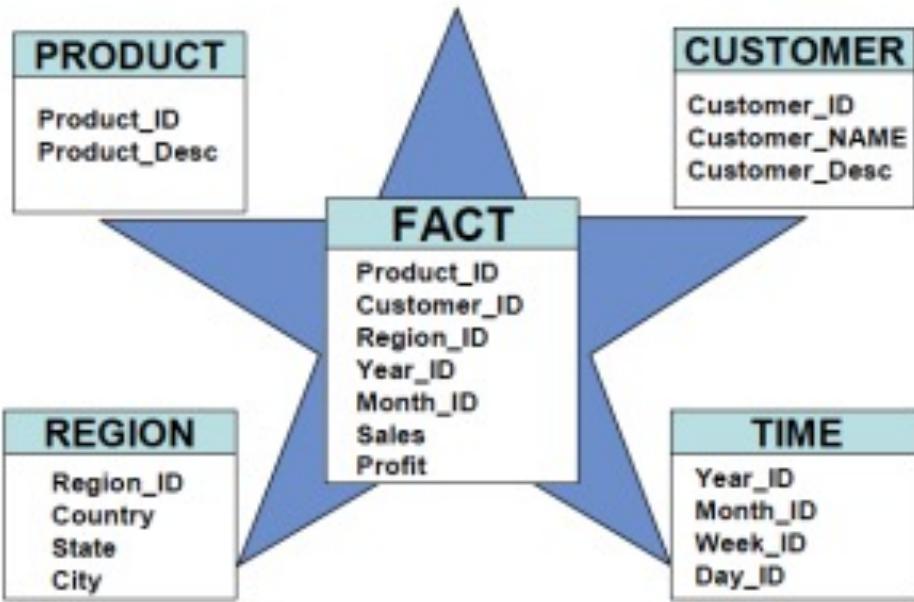
- Kennzahlen sind **Grundlage** für fachliche Auswertung:
  - Banken:
    - Net New Assets (NNA)
    - Assets under Management (AuM)
  - Telekommunikation:
    - Average Revenue per User (ARPU)
  - eBusiness:
    - Click through Rate (CTR)
- Kennzahlen sind **Aggregate** über bestimmten Zeitraum
  - D.h. keine Einzelbetrachtungen



# Dimensionen

- Ansammlung von Daten, die die Fakten von einer bestimmten Sicht aus beschreiben
  - Dimensionen beschreiben den **kontextuellen Hintergrund von Fakten**
  - Dimensionen dienen der **Klassifizierung von Fakten**
- 
- Beispiele:
    - Kunden
    - Produkte
    - ... weitere?





Liste aller Verkäufe von iPhones in Zürich?



# Beispielabfrage – Details

Product

Product_ID	Product_Desc
P1	Motorola
P2	iPhone
P3	Windows

Fact

Prod_ID	Cust_ID	Reg_ID	Time_ID	Sales	Profit
P2	C1	R2	T1	1000	200
P2	C2	R3	T3	3000	500

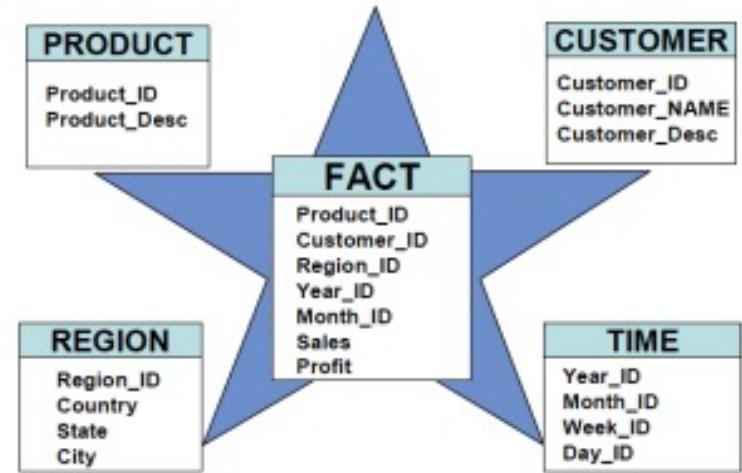
Region

Region_ID	Country	State	City
R1	CH	ZH	Zurich
R2	CH	VD	Lausanne
R3	CH	ZH	Winti

Liste aller Verkäufe von iPhones in Zürich?

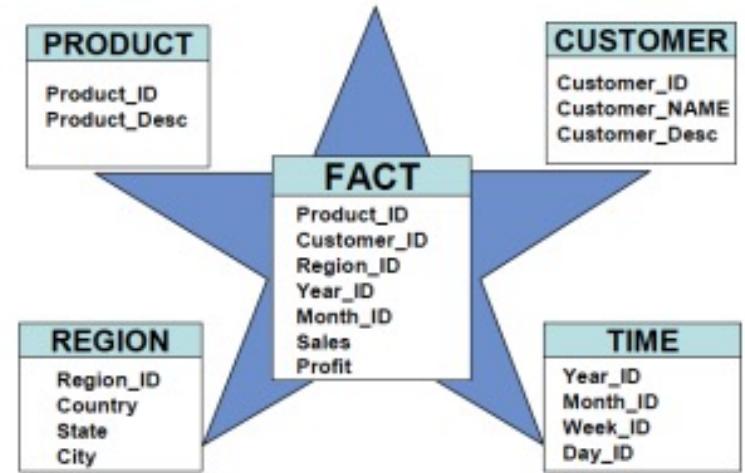
# Beispielabfrage: SQL

```
SELECT r.city, sales
FROM
    Fact f, Product p, Region r
WHERE
    f.product_ID = p.productID AND
    f.region_ID = r.region_ID AND
    p.product_desc = 'iPhone' AND
    r.city = 'Zurich'
```

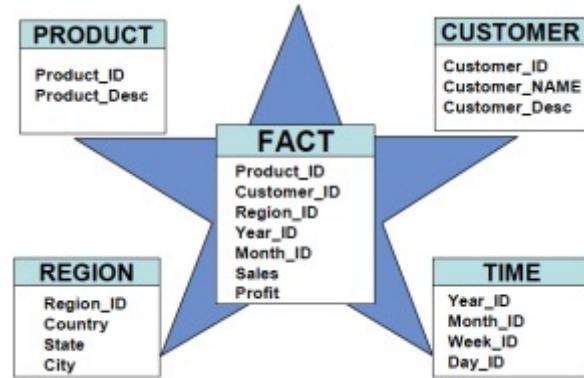


# Beispielabfrage: SQL (in ANSI SQL-92 Syntax – empfohlen)

```
SELECT r.city, sales  
FROM  
    Fact f  
INNER JOIN Product p  
    ON f.product_ID = p.productID  
INNER JOIN Region r  
    ON f.region_ID = r.region_ID  
WHERE  
    p.product_desc = 'iPhone' AND  
    r.city = 'Zurich'
```



# Beispielabfrage: Gruppierungsfunktion



Anzahl der iPhone-Verkäufe pro Kanton?



# Beispielabfrage: SQL mit «Group By»

**SELECT** state, sum(sales)

**FROM**

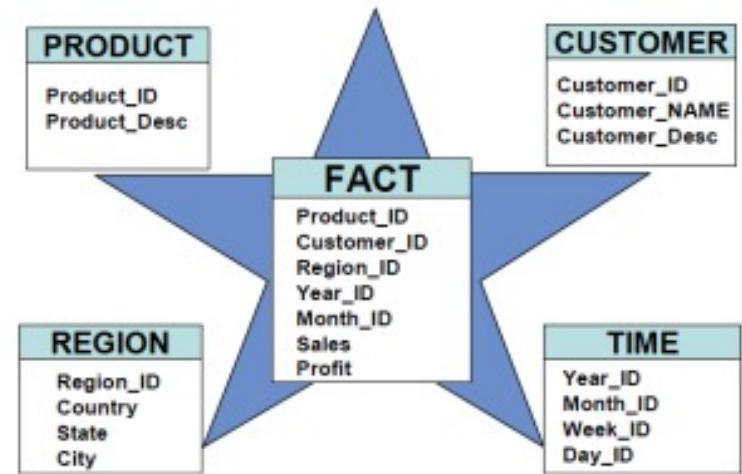
Fact f, Product p, Region r

**WHERE**

f.product\_ID = p.productID **AND**  
f.region\_ID = r.region\_ID **AND**  
p.product\_desc = 'iPhone'

**GROUP BY**

state

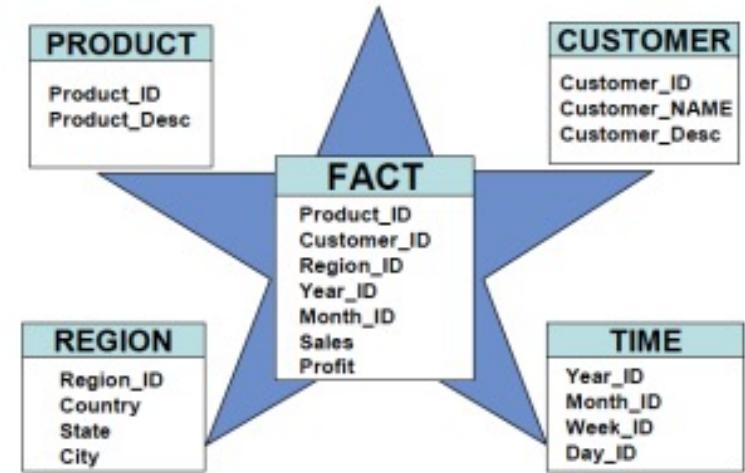


# Beispielabfrage: SQL mit «Group By» (in ANSI SQL-92 Syntax – empfohlen)

```
SELECT state, sum(sales)  
FROM  
    Fact f  
INNER JOIN Product p  
    ON f.product_ID = p.productID  
INNER JOIN Region r  
    ON f.region_ID = r.region_ID
```

```
WHERE  
    p.product_desc = 'iPhone'
```

**GROUP BY**  
state



# Eigenschaften des Sternschemas

- **Faktendaten** werden typischer Weise **sehr gross**:
  - **Milliarden** von Records
  - **Partitionierung** der Faktentabellen und **Auslagerung** alter Records
- Fakten sollten sich **auf fachlichen Prozess beschränken**:
  - z.B. Verkauf
- Data Mart kann aus unterschiedlichen Faktentabellen bestehen
- **Dimensionstabellen** werden typischer Weise **nicht so gross** wie Faktentabellen
  - Haben dafür ggf. **viele Attribute** (Spalten)

# Hierarchien

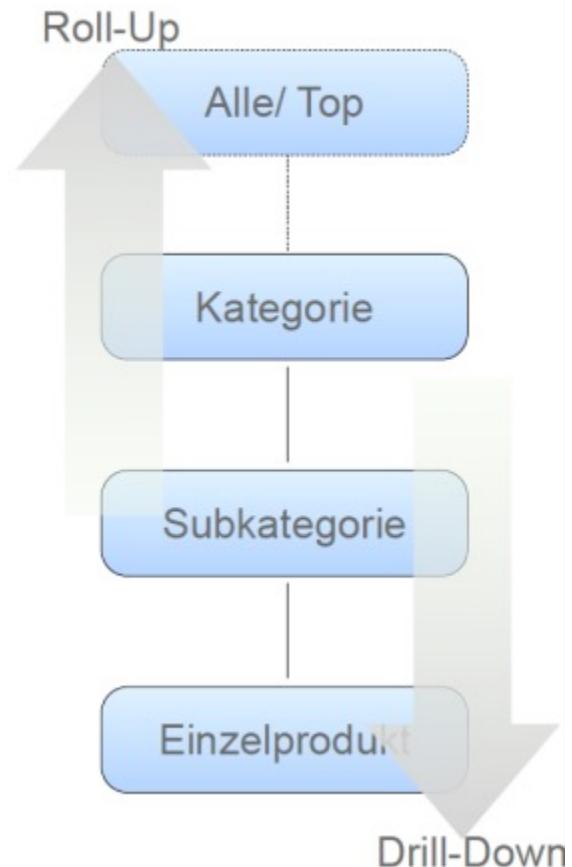
# Beispiel Produkt

Hierarchische Beziehung: Produkt → Subkategorie → Produktkategorie

PROD_NAME	PROD_SUBCATEGORY	PROD_CATEGORY
5MP Telephoto Digital Camera	Cameras	Photo
17" LCD w/built-in HDTV Tuner	Monitors	Peripherals and Accessories
Envoy 256MB - 40GB	Desktop PCs	Hardware
X Box	Game Consoles	Electronics
Mini DV Camcorder with 3.5" Swivel LCD	Camcorders	Photo
Envoy Ambassador	Portable PCs	Hardware
Laptop carrying case	Accessories	Software/Other
Home Theatre Package with DVD-Audio/Video Play	Home Audio	Electronics
18" Flat Panel Graphics Monitor	Monitors	Peripherals and Accessories
Envoy External Keyboard	Accessories	Peripherals and Accessories
External 101-key keyboard	Accessories	Software/Other
PCMCIA modem/fax 28800 baud	Modems/Fax	Peripherals and Accessories
SIMM- 8MB PCMCIAII card	Memory	Peripherals and Accessories
SIMM- 16MB PCMCIAII card	Memory	Peripherals and Accessories
Multimedia speakers- 3" cones	Accessories	Peripherals and Accessories

# Multidimensionale Analyse: Drilling

- **Drill-Down** (mehr Details zeigen)
  - Hierarchie von oben nach unten traversieren
  - Alle Produkte -> Kategorie -> Subkategorie -> Einzelprodukt
- **Roll-Up** (weniger Details zeigen):
  - Hierarchie von unten nach oben traversieren
  - Einzelprodukt -> Subkategorie -> Kategorie -> alle Produkte



Year	All ProductLine	Product Line	Continent Name	Measures	Europe	North America	Total
				Sum of Quantity	17145	4466	21631
1998	All ProductLine	Children	Europe	Sum of Quantity	66319	16803	88122
		Clothes & Shoes	Europe	Sum of Quantity	26707	9255	35962
		Outdoors	Europe	Sum of Quantity	63322	33952	97274
		Sports	Europe	Sum of Quantity	178493	66476	242969
			North America	Sum of Quantity	20665	6126	26811
		Children	North America	Sum of Quantity	80613	23161	103774
		Clothes & Shoes	North America	Sum of Quantity	29385	10025	39610
		Outdoors	North America	Sum of Quantity	75336	37568	112904
		Sports	North America	Sum of Quantity	206110	79500	283093
Subtotal: 1998	All ProductLine						
1999	All ProductLine	Children	Europe	Sum of Quantity	23768	7079	30847
		Clothes & Shoes	Europe	Sum of Quantity	93130	30920	124060
		Outdoors	Europe	Sum of Quantity	36419	12908	49327
		Sports	Europe	Sum of Quantity	91598	46164	137752
			North America	Sum of Quantity	244905	97081	341986
		Children	North America	Sum of Quantity	21381	5545	26926
		Clothes & Shoes	North America	Sum of Quantity	78355	23414	101769
		Outdoors	North America	Sum of Quantity	33080	10995	44075
		Sports	North America	Sum of Quantity	80320	34859	118479
Subtotal: 2000	All ProductLine						
2001	All ProductLine	Children	Europe	Sum of Quantity	216436	74813	291249
		Clothes & Shoes	Europe	Sum of Quantity	76132	21204	97336
		Outdoors	Europe	Sum of Quantity	96474	26264	122058
		Sports	Europe	Sum of Quantity	36698	13594	50292
			North America	Sum of Quantity	109498	36788	140278
Subtotal: 2001	All ProductLine						
2002	All ProductLine	Children	Europe	Sum of Quantity	255321	84570	339891
		Clothes & Shoes	Europe	Sum of Quantity	1099368	299820	1499188
Subtotal: 2002	All ProductLine						
Total				Sum of Quantity	1099368	299820	1499188

# Drilling in SQL

- Einschränkung auf bestimmte Dimensionsattribute, die Kategorien darstellen
  - WHERE prod\_category='photo' oder
  - WHERE prod\_subcategory=' cameras'
- Aufnahme dieser **Kategorieattribute** einer Dimensionstabelle in die **GROUP BY** Klausel
- Beispiel: Anzahl aller Produkte pro Produktkategorie

```
SELECT prod_category, count(*)  
FROM products  
WHERE  
    prod_category='photo'  
GROUP BY  
    prod_category
```

```
SELECT prod_subcategory,  
count(*)  
FROM products  
WHERE  
    prod_category='photo'  
GROUP BY  
    prod_subcategory
```

# SQL-Drill-Down Aufgabe

- Gegeben:
  - 1 Produkt-Dimension
  - 1 Faktentabelle

Product

Product_ID	Product_Desc	Product_Category	Product_Subcategory
P1	Mac Book	Electronics	Computer
P2	Ice Tea	Goods	Drinks
P3	iPhone	Electronics	Phone

Fact

Prod_ID	Cust_ID	Reg_ID	Time_ID	Sales	Profit
P1	C1	R2	T1	1000	200
P2	C2	R3	T3	3000	500

## Aufgabe:

- Gesamtsumme der Verkäufe aller Produkte
- Auflisten aller Teilsummen an Verkäufen pro Produktkategorie



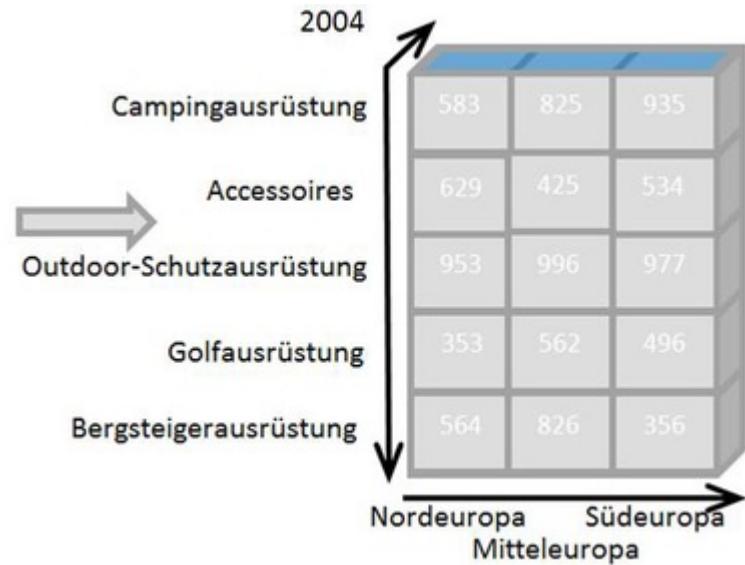
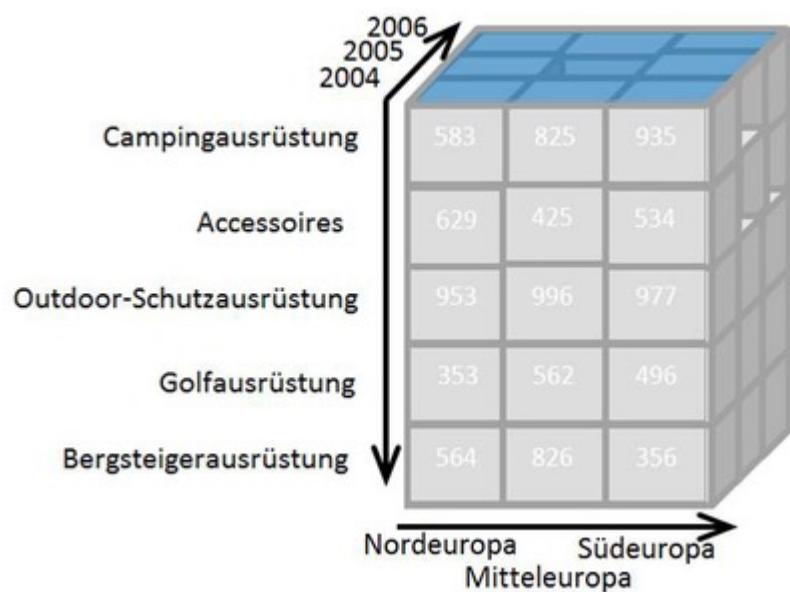
# Lösung

a) SELECT **sum**(profit)  
FROM fact f, product p  
WHERE f.prod\_ID=p.product\_ID

b) SELECT p.product\_category, **sum**(profit)  
FROM fact f, product p  
WHERE f.prod\_ID=p.product\_ID  
**GROUP BY** product\_category

# Slicing

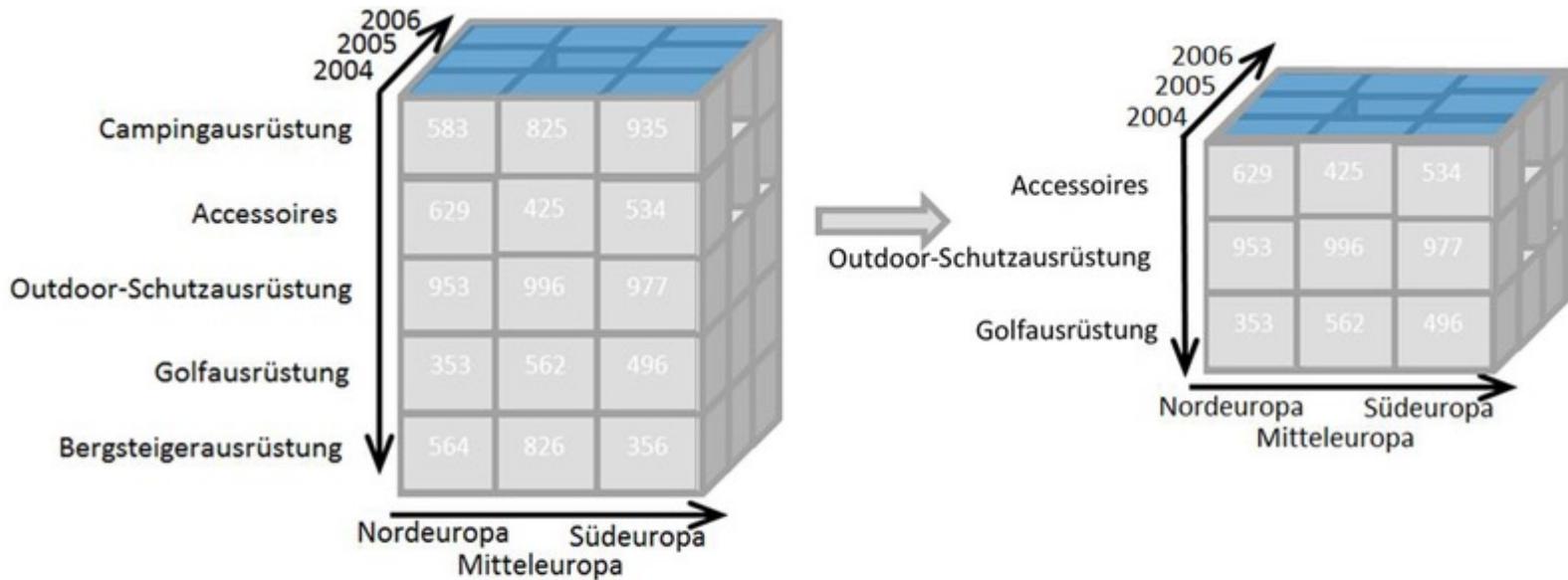
- Ausschneiden von Scheiben aus einem Würfel



Quelle: Wikipedia

# Dicing

- Slicing in mehreren Dimensionen: Ausschneiden eines Sub-Würfels



Quelle: Wikipedia

# OLAP Cube Technologien

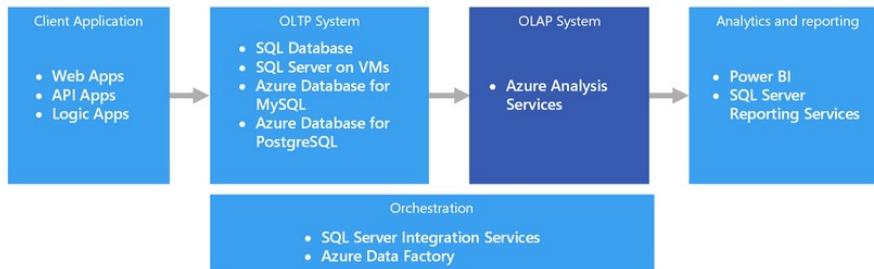
- Unterschiedliche Datenbanken bieten Cube-Technologie an:
  - z.B. Microsoft oder PostgreSQL

## Online analytical processing (OLAP)

02/12/2018 • 8 minutes to read •  +5

Online analytical processing (OLAP) is a technology that organizes large business databases and supports complex analysis. It can be used to perform complex analytical queries without negatively affecting transactional systems.

The databases that a business uses to store all its transactions and records are called [online transaction processing \(OLTP\)](#) databases. These databases usually have records that are entered one at a time. Often they contain a great deal of information that is valuable to the organization. The databases that are used for OLTP, however, were not designed for analysis. Therefore, retrieving answers from these databases is costly in terms of time and effort. OLAP systems were designed to help extract this business intelligence information from the data in a highly performant way. This is because OLAP databases are optimized for heavy read, low write workloads.



## Introduction to the PostgreSQL CUBE

PostgreSQL `CUBE` is a subclause of the `GROUP BY` clause. The `CUBE` allows you to generate multiple grouping sets.

A grouping set is a set of columns to which you want to group. For more information on the grouping sets, check it out the [GROUPING SETS](#) tutorial.

The following illustrates the syntax of the `CUBE` subclause:

```
SELECT  
    c1,  
    c2,  
    c3,  
    aggregate (c4)  
FROM  
    table_name  
GROUP BY  
    CUBE (c1, c2, c3);
```

# Alternative Technologien zu OLAP Cubes

Business Intelligence for practitioners

30 JANUARY 2020 / 17 MIN READ / BUSINESS INTELLIGENCE, DATA MODELING

## The Rise and Fall of the OLAP Cube

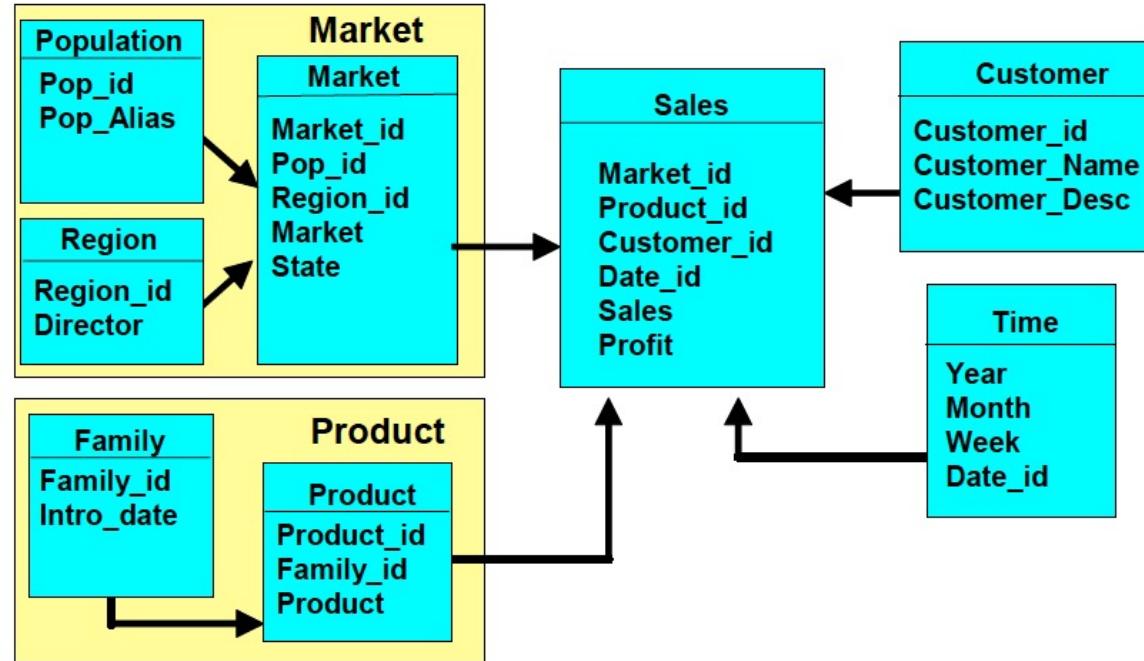
by Cedric Chin



One of the biggest shifts in data analytics over the past decade is the move *away* from building 'data cubes', or 'OLAP cubes', to running OLAP\* workloads directly on columnar databases.

<https://www.holistics.io/blog/the-rise-and-fall-of-the-olap-cube/>

# Exkurs: Schneeflockenschema



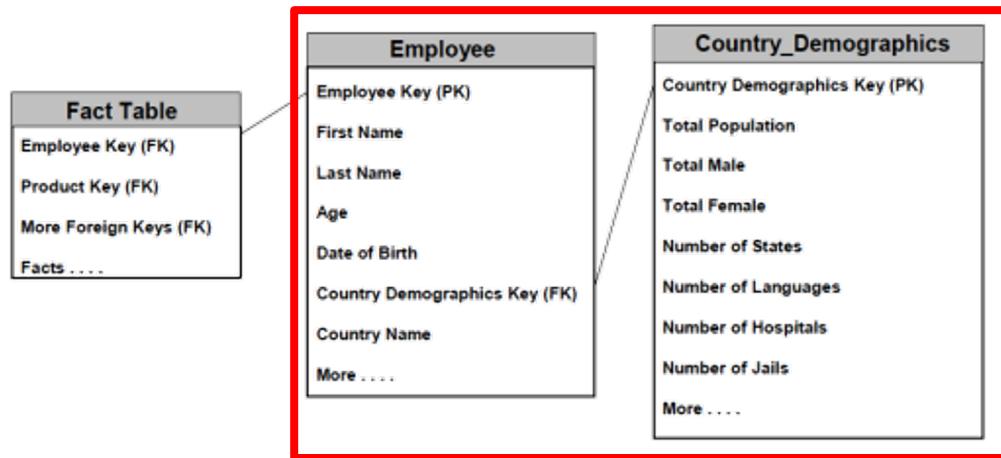
Quelle: IBM Redbook – Data Mart Consolidation

- Dimensionen sind weiter aufgespalten (normalisiert)

# Einsatz des Schneeflockenschemas

- Idee: **Normalisiere Dimensionen** entlang der Hierarchien (s.u.)
- Sollte generell **vermieden** werden:
  - Abfragen werden durch zusätzlichen Join verlangsamt
- Verwendung, wenn
  - **Zeit-/Platzbedarf** wichtiger als Einfachheit von Schema/Abfrage
  - Entitäten **unterschiedliche Granularität** haben:
    - Etwa wegen verschiedener Source-Systeme
    - z.B: Employee vs. Country\_Demographics

**Granularitäten:**  
- Einzelperson  
- Land (n Personen)



# Zeitdimension

# Zeitdimension

- Fast jedes Sternschema enthält eine Zeitdimension
  - Warum?
  - Wie wird sie gebaut?

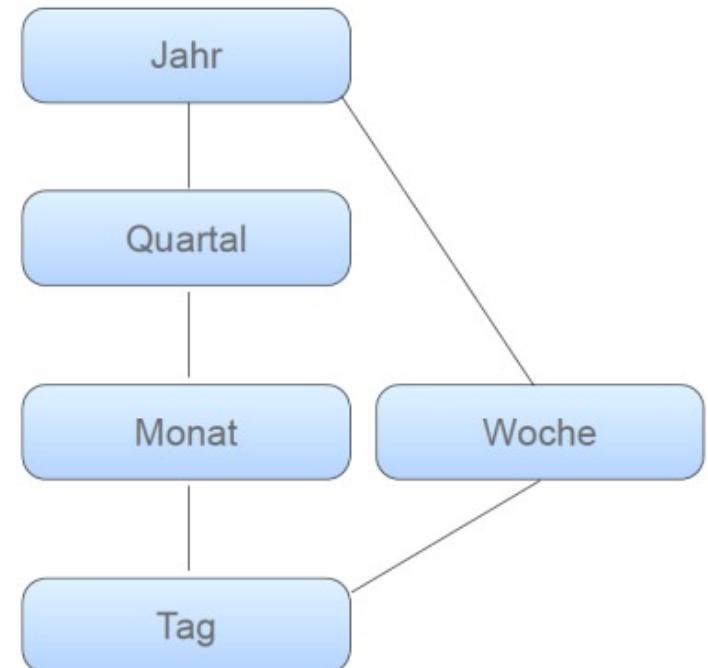


# Zeitdimension - Antworten

- Warum wird Zeitdimension modelliert?
  - Der Datentyp DATE stellt zu wenige explizite Merkmale zur Verfügung
  - Beispiele:
    - Feiertag, Anzahl Tage/Jahr, Stundenanzahl pro Tag, Wochentag, Tagnummer im Jahr, Wochennummer, Firmenkalender, Fiskaljahr...
    - Modellierung als Dimension ermöglicht Drill-Down mit demselben Ansatz wie bei den anderen Dimensionen
    - Viele explizite deskriptive Attribute erleichtern Auswertungen
- Wie wird sie modelliert?
  - Generierung mittels Scripts, Tool, ...
  - Z.B. in Excel, dann Übernahme in Tabelle

# Granularität der Zeitdimension

- Zeitdimension erstreckt sich üblicher Weise über 10 – 20 Jahre
- Ein Datensatz pro Tag
- Roll-Up Kriterien:
  - Jahr-Quartal-Monat-Tag
  - Jahr-Woche-Tag
  - Kriterien sind inkompatibel
  - Modellierung durch alternative Hierarchien



Innerhalb einer Hierarchiefolge unmöglich,  
weil Woche keine Teilmenge von Monat ist  
→ 2 Zeitdimensionen

# SQL-Zeitabfrage

- Gegeben:
  - 1 Produktdimension
  - 1 Zeitdimension
  - 1 Faktentabelle
- Aufgabe:
  - a) Anzahl der Verkäufe von iPhones im 3. Quartal 2014
  - b) Anzahl der Verkäufe in den ersten 5 Tagen jedes Monats



# SQL-Zeitabfrage – Ohne Zeitdimension

Product

Product_ID	Product_Desc
P1	Motorola
P2	iPhone
P3	Windows

Fact

Prod_ID	Cust_ID	Reg_ID	Time_ID	Sales	Profit	Time
P2	C1	R2	T1	1000	200	2014-11-10
P2	C2	R3	T3	3000	500	2013-07-03

- Aufgabe:
  - a) Anzahl der Verkäufe von iPhones im 3. Quartal 2014
  - b) Anzahl der Verkäufe in den ersten 5 Tagen jedes Monats
- Wie würden Sie diese Abfragen ohne Zeitdimension machen?



# Answer Sketch

a)

```
SELECT sales
FROM ...
WHERE ... and function_call
```

b) “complicated”

# SQL-Zeitabfrage – Mit Zeitdimension

Product

Product_ID	Product_Desc
P1	Motorola
P2	iPhone
P3	Windows

Fact

Prod_ID	Cust_ID	Reg_ID	Time_ID	Sales	Profit
P2	C1	R2	T1	1000	200
P2	C2	R3	T3	3000	500

Time

Time_ID	Year	Quarter	Month	Day
T1	2013	1	1	1
T2	2013	1	1	2
T3	2013	1	1	3

- Aufgabe:
  - a) Anzahl der Verkäufe von iPhones im 3. Quartal 2014
  - b) Anzahl der Verkäufe in den ersten 5 Tagen jedes Monats



# Answer Sketch

a)

```
SELECT sales  
FROM ..  
WHERE ... and quarter=3 and year=2014
```

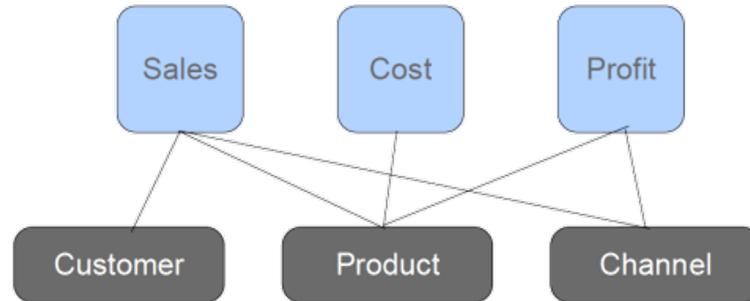
b)

```
SELECT month, sum(sales)  
FROM ...  
WHERE day <= 5
```

Assumption: Days are numbered from 1 to n starting from 1 for each month

# Conformed Dimensions

- Mehrere **unterschiedliche Fragestellungen** sollen ausgewertet werden  
→ Ggf. Verwendung von **mehreren Sternschemata**
- Manche Dimensionen können von mehreren Fakten referenziert werden
- Voraussetzung:
  - Dimensionstabellen haben die notwendige Granularität

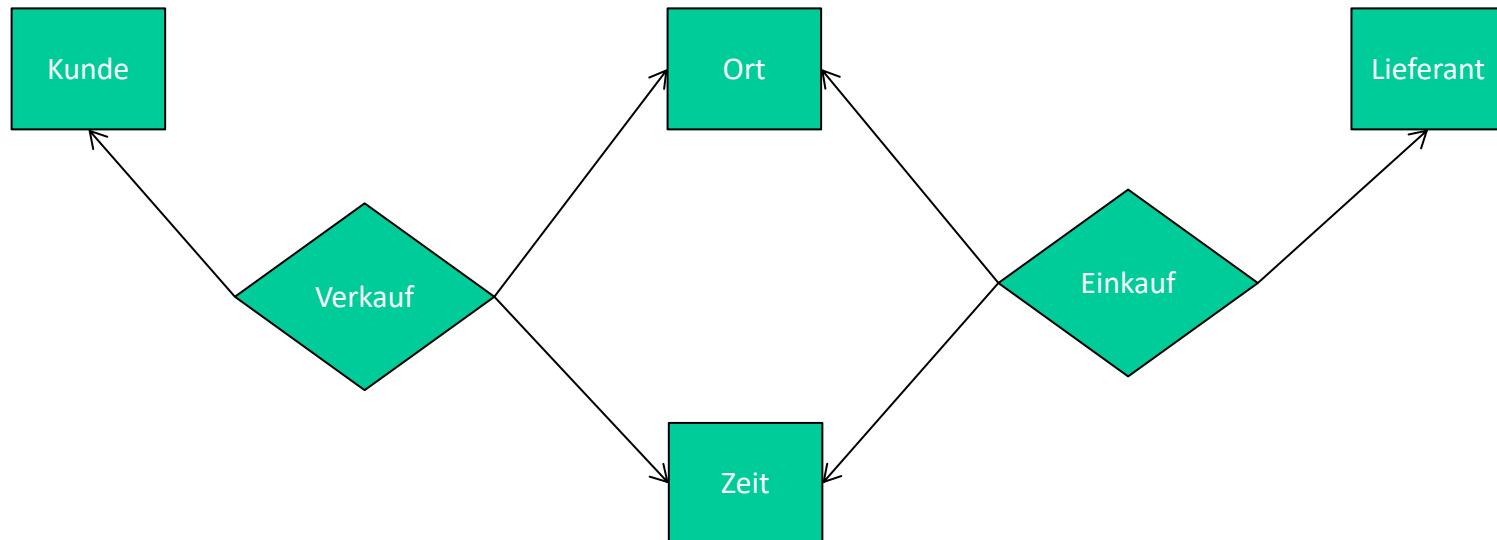


*“A conformed dimension is a dimension that has exactly the same meaning and content when being referred from different fact tables.”* (1keydata.com)

# Abilden von mehreren Subjekten

- Sie möchten mehr als ein Subjekt (Fakt) abbilden: Verkauf, Einkauf, ...
- Diese Subjekte haben gemeinsame Dimensionen: Produkt, Zeit, Region

→ Galaxy Schema Konzept



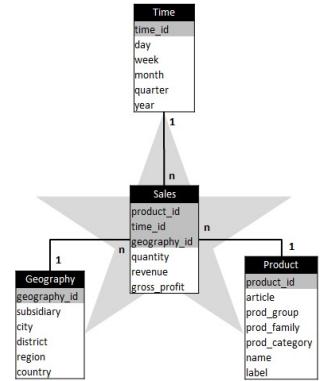
# Galaxien

- Grundlage unternehmensweiter Anwendungen können «Galaxien» sein
- Es existieren **mehrere Faktentabellen** und zugehörige **Stern- bzw. Schneeflockenschemata**:
  - Inhaltliche Zusammenhänge
  - Unterschiedliche Aggregierungsstufen in Aggregat tabellen
- Dimensionen werden häufig von mehreren Faktentabellen verwendet
  - Drill across (synchrone Drilling in mehreren Dimensionen) bei der Analyse
  - Dieselben Einschränkungen (GROUP BY, WHERE) gelten auf anderer Faktentabelle

# Zusammenfassung

**Fakten** sind die WHATs : alles was wir messen

- In der Regel numerisch
- Beispiele:
  - Umsatz, Bestellungen, Lagerbestand, Temperatur, Blutwert



**Dimensionen** sind die BYs: danach wird qualifiziert/gruppiert

- In der Regel hierarchisch aufgebaut
  - z.B. Zeit-Dimension: Jahr->Quartal->Monat->Tag
- Beispiele:
  - Zeit, Produkt, Organisation, Geographie, Dienstleistung

→ Siehe Vertiefung im Praktikum 2