Anfragesprachen für On-Line Analytical Processing (OLAP)

Seminar Business Intelligence
Teil I: OLAP & Data Warehousing

René Rondot

rondot@informatik.uni-kl.de

Universität Kaiserslautern

Anfragesprachen für OLAP

Gliederung:

1. Data Cube Operator

   Grundlegender Operator für Aggregation über mehrere Dimensionen

2. Regel-basierte Sprachen

   komplexer formaler Ansatz für Operationen auf einem Cube

3. MDX

   ***M**ulti**D**imensional e**X**pressions*: Microsoft-Entwicklung

4. *n*D-SQL

   Sprache für föderierte Datenbanken und Ad-Hoc-Aggregation

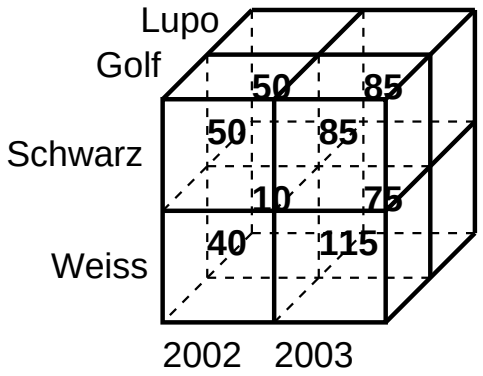
1. Der Data Cube Operator

- vorgestellt 1997 von Jim Gray et al.
- arbeitet auf normalen Relationen
- zunächst: mehrdimensionaler Roll-Up
  - Roll-Up nacheinander über mehrere Dimensionen
  - Aggregation mit verschiedenen Detailliertheitsgraden
- Beispiel: Relation mit Verkaufsdaten von Fahrzeugen
  Attribute: Modell, Jahr, Farbe, Verkaeufe

Ergebnis des Roll-Up (1)

```
SELECT Modell, Jahr, Farbe, SUM(Verkaeufe)
 FROM VerkaufTabelle
 GROUP BY Modell, Jahr, Farbe
```

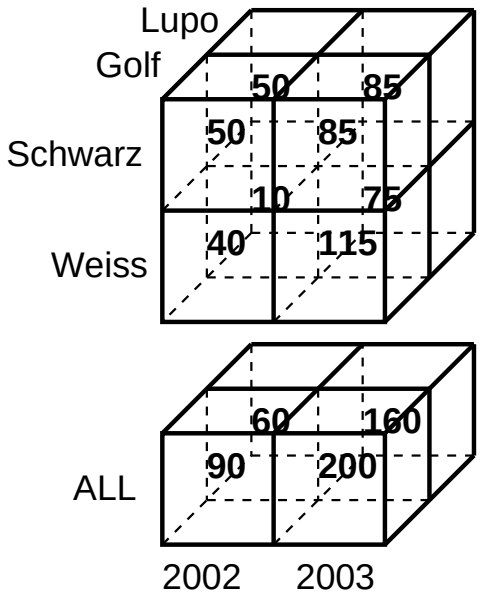
Modell	Jahr	Farbe	SUM
Golf	2002	Weiß	50
Golf	2002	Schwarz	40
Golf	2003	Weiß	85
Golf	2003	Schwarz	115
Lupo	2002	Weiß	50
Lupo	2002	Schwarz	10
Lupo	2003	Weiß	85
Lupo	2003	Schwarz	75



Ergebnis des Roll-Up (2)

```
UNION
  SELECT Modell, Jahr, 'ALL', SUM(Verkaeufe)
   FROM VerkaufTabelle
   GROUP BY Modell, Jahr
```

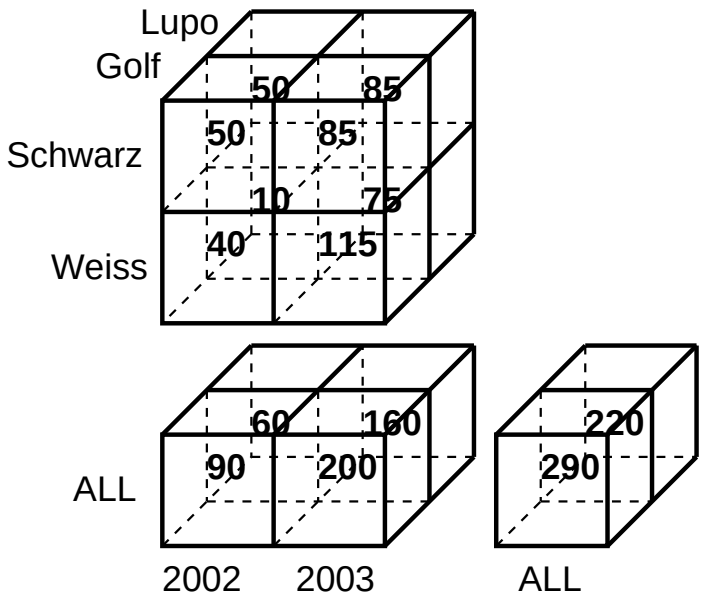
Modell	Jahr	Farbe	SUM
Golf	2002	Weiß	50
Golf	2002	Schwarz	40
...	...	...	...
Golf	2002	ALL	90
Golf	2003	ALL	200
Lupo	2002	ALL	60
Lupo	2003	ALL	160



Ergebnis des Roll-Up (3)

```
UNION
  SELECT Modell, 'ALL', 'ALL', SUM(Verkaeufe)
   FROM VerkaufTabelle
   GROUP BY Modell
```

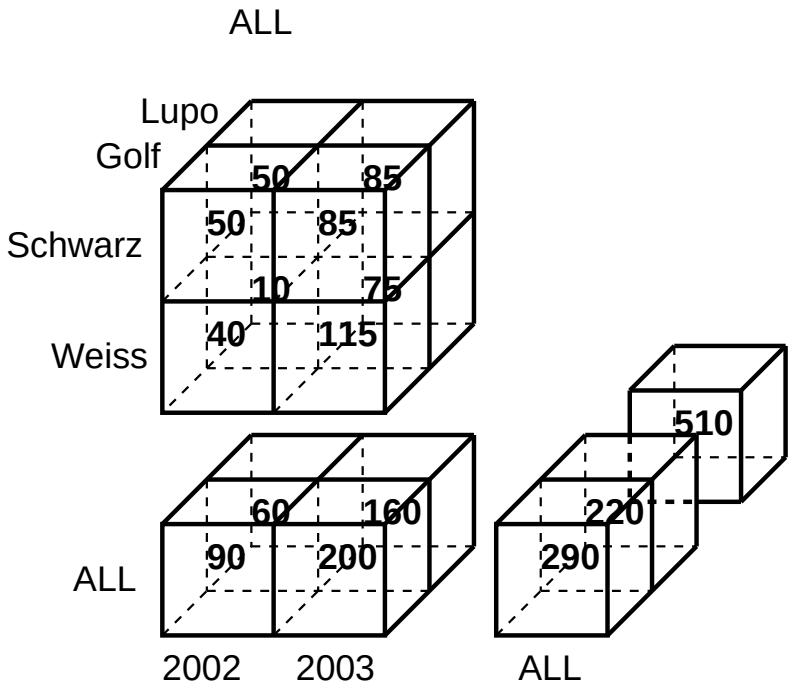
Modell	Jahr	Farbe	SUM
Golf	2002	Weiß	50
Golf	2002	Schwarz	40
...	...	...	...
Golf	2002	ALL	90
Golf	2003	ALL	200
Lupo	2002	ALL	60
Lupo	2003	ALL	160
Golf	ALL	ALL	290
Lupo	ALL	ALL	220



Ergebnis des Roll-Up (4)

```
UNION
  SELECT 'ALL', 'ALL', 'ALL', SUM(Verkaeufe)
   FROM VerkaufTabelle
```

Modell	Jahr	Farbe	SUM
Golf	2002	Weiß	50
Golf	2002	Schwarz	40
...	...	...	...
Golf	2002	ALL	90
Golf	2003	ALL	200
Lupo	2002	ALL	60
Lupo	2003	ALL	160
Golf	ALL	ALL	290
Lupo	ALL	ALL	220
ALL	ALL	ALL	510

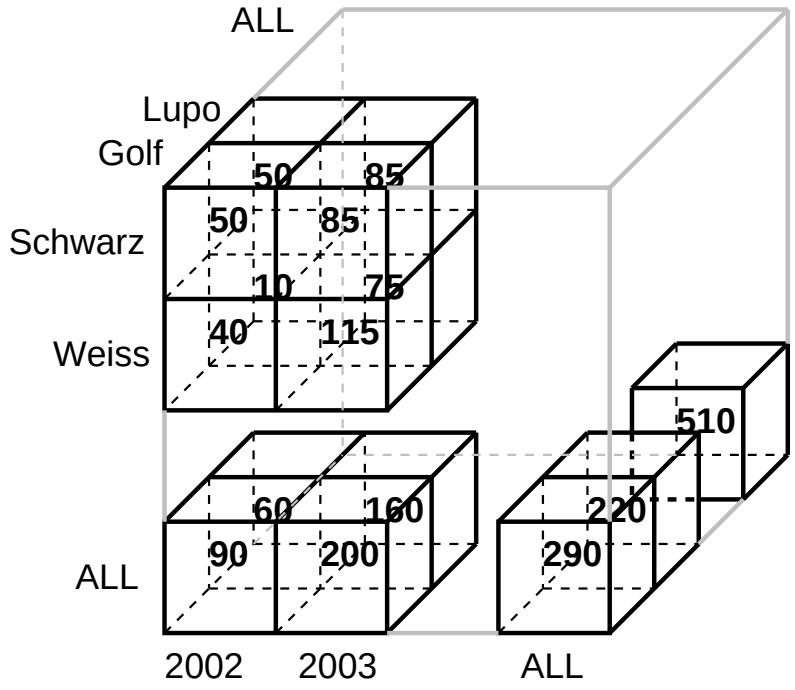


SQL-Statement für Roll-Up

Roll-Up über die 3 Dimensionen Farbe, Jahr, Modell:

```
 SELECT Modell, Jahr, Farbe, SUM(Verkaeufe)
  FROM VerkaufTabelle
  GROUP BY Modell, Jahr, Farbe
UNION
 SELECT Modell, Jahr, 'ALL', SUM(Verkaeufe)
  FROM VerkaufTabelle
  GROUP BY Modell, Jahr
UNION
 SELECT Modell, 'ALL', 'ALL', SUM(Verkaeufe)
  FROM VerkaufTabelle
  GROUP BY Modell
UNION
 SELECT 'ALL', 'ALL', 'ALL', SUM(Verkaeufe)
  FROM VerkaufTabelle
```

Data Cube

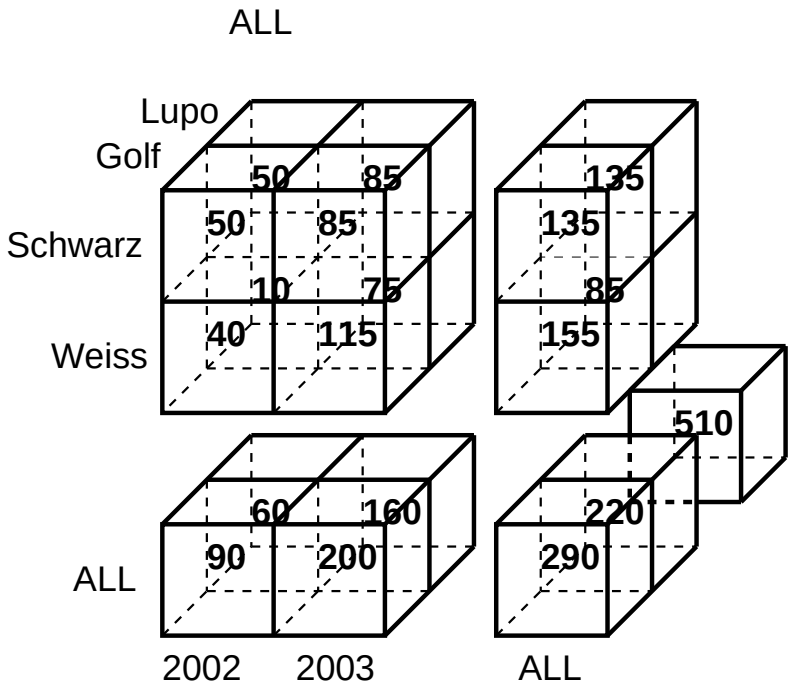


- Roll-Up ist asymmetrisch
- symmetrische Erweiterung: Data Cube

Ergebnis des Data Cube (1)

```
UNION
  SELECT Modell, 'ALL', Farbe, SUM(Verkaeufe)
   FROM VerkaufTabelle
   GROUP BY Modell, Farbe
```

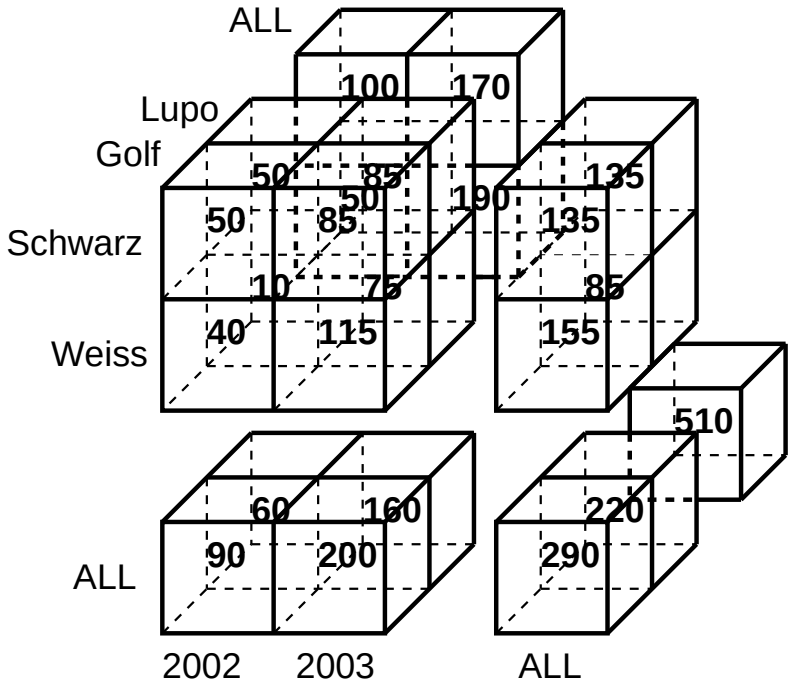
Modell	Jahr	Farbe	SUM
Golf	2002	Weiß	50
...	...	...	...
Golf	ALL	Schwarz	135
Golf	ALL	Weiß	155
Lupo	ALL	Schwarz	135
Lupo	ALL	Weiß	85



Ergebnis des Data Cube (2)

```
UNION
  SELECT 'ALL', Jahr, Farbe, SUM(Verkaeufe)
   FROM VerkaufTabelle
   GROUP BY Jahr, Farbe
```

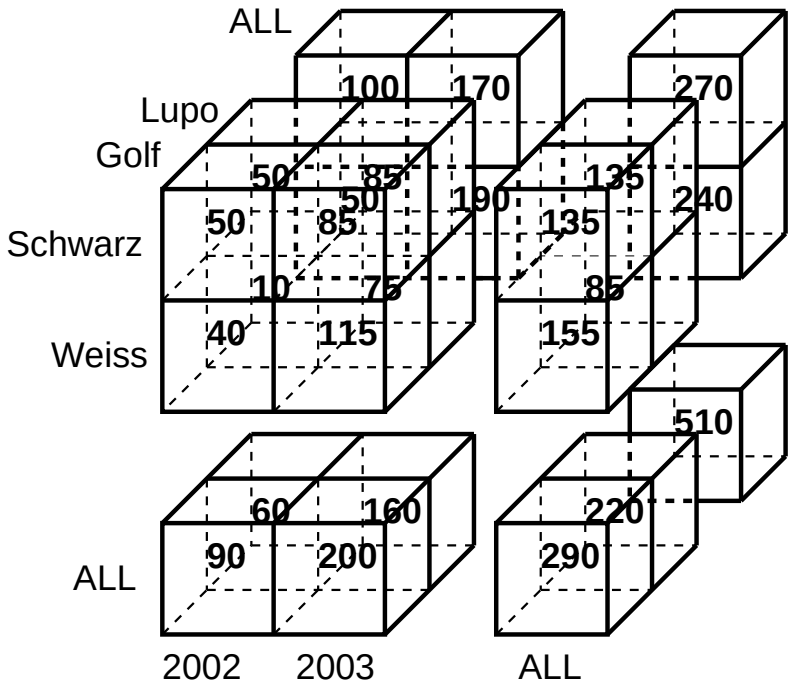
Modell	Jahr	Farbe	SUM
Golf	2002	Weiß	50
...	...	...	...
Golf	ALL	Schwarz	135
...	...	...	...
ALL	2002	Schwarz	100
ALL	2002	Weiß	50
ALL	2003	Schwarz	170
ALL	2003	Weiß	190



Ergebnis des Data Cube (3)

```
UNION
  SELECT 'ALL', 'ALL', Farbe, SUM(Verkaeufe)
   FROM VerkaufTabelle
   GROUP BY Farbe
```

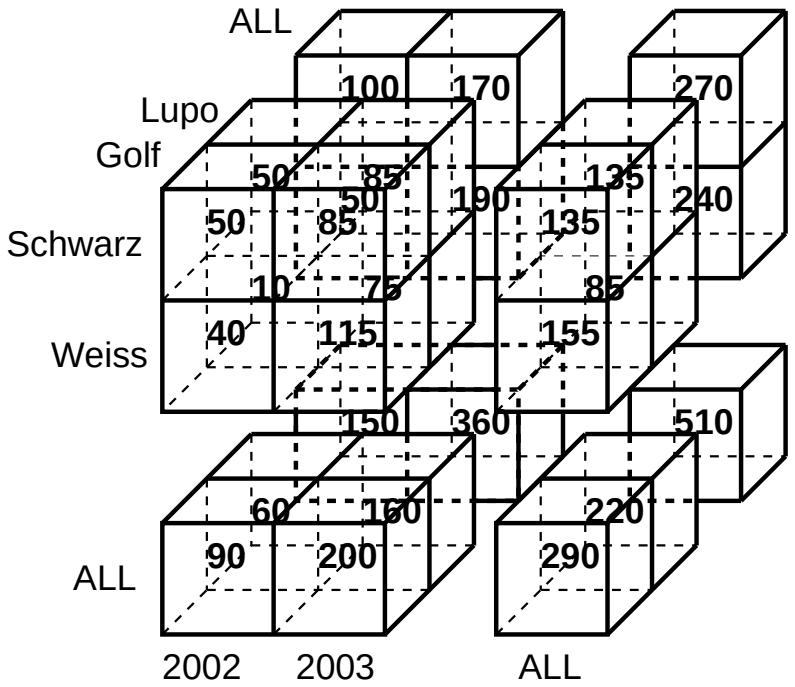
Modell	Jahr	Farbe	SUM
Golf	2002	Weiß	50
...	...	...	...
ALL	2002	Schwarz	100
ALL	2002	Weiß	50
ALL	2003	Schwarz	170
ALL	2003	Weiß	190
ALL	ALL	Schwarz	270
ALL	ALL	Weiß	240



Ergebnis des Data Cube (4)

```
UNION
  SELECT 'ALL', Jahr, 'ALL', SUM(Verkaeufe)
   FROM VerkaufTabelle
   GROUP BY Jahr
```

Modell	Jahr	Farbe	SUM
Golf	2002	Weiß	50
...	...	...	...
ALL	2002	Schwarz	100
ALL	2002	Weiß	50
ALL	2003	Schwarz	170
ALL	2003	Weiß	190
ALL	ALL	Schwarz	270
ALL	ALL	Weiß	240
ALL	2002	ALL	150
ALL	2003	ALL	360



Data Cube in SQL

Statement für 3-dimensionalen Roll-Up plus:

```
UNION
 SELECT Modell, 'ALL', Farbe, SUM(Verkaeufe)
  FROM VerkaufTabelle
  GROUP BY Modell, Farbe
UNION
 SELECT 'ALL', Jahr, Farbe, SUM(Verkaeufe)
  FROM VerkaufTabelle
  GROUP BY Jahr, Farbe
UNION
 SELECT 'ALL', 'ALL', Farbe, SUM(Verkaeufe)
  FROM VerkaufTabelle
  GROUP BY Farbe
UNION
 SELECT 'ALL', Jahr, 'ALL', SUM(Verkaeufe)
  FROM VerkaufTabelle
  GROUP BY Jahr;
```

Problem

- SQL-Statements sehr lang und kompliziert
- Bei n Dimensionen:
  - Roll-Up: $n + 1$ Teilstatements
  - Data Cube: 2^n Teilstatements
- Nachteile der komplizierten Statements
  - für Entwickler: Entwurf
  - für Datenbank: Optimierung

Lösung

- eigene Operatoren für Roll-Up und Data Cube
- Erweiterung des SQL-Operators `GROUP BY`
- im Beispiel:
  - Roll-Up:

```
SELECT Modell, Jahr, Farbe, SUM(Verkaeufe)
  FROM VerkaufTabelle
  GROUP BY ROLLUP Modell, Jahr, Farbe
```

  - Data Cube:

```
SELECT Modell, Jahr, Farbe, SUM(Verkaeufe)
  FROM VerkaufTabelle
  GROUP BY CUBE Modell, Jahr, Farbe
```

Data Cube in SQL:1999

- Grouping Sets ermöglichen Zwischenformen zwischen Roll-Up und Data Cube (Teilmenge aller Permutationen)
- vermeidet den `ALL`-Wert
  - `NULL` statt `ALL`
  - `GROUPING` zur Unterscheidung von echtem `NULL`
- Beispielanfrage:

```
SELECT Modell, Jahr, SUM(Verkaeufe),
   GROUPING(Modell), GROUPING(Jahr)
 FROM VerkaufTabelle
 GROUP BY CUBE (Modell, Jahr)
```

Ergebnis der Beispielanfrage

Modell	Jahr	SUM(Verkaeufe)	GROUPING(Modell)	GROUPING(Jahr)
Golf	2002	90	FALSE	FALSE
Golf	2003	200	FALSE	FALSE
Lupo	2002	60	FALSE	FALSE
Lupo	2003	160	FALSE	FALSE
Golf	NULL	290	FALSE	TRUE
Lupo	NULL	220	FALSE	TRUE
NULL	2002	150	TRUE	FALSE
NULL	2003	360	TRUE	FALSE
NULL	NULL	510	TRUE	TRUE

Gliederung

1. Data Cube Operator
2. **Regel-basierte Sprachen**
3. MultiDimensional eXpressions (MDX)
4. *n*D-SQL

2. Regel-basierte Sprachen

- Beispiel: von Hacid et al. 1997 vorgestellte Sprache
- arbeitet auf Cubes
- Festlegung der Beziehung zwischen zwei Zellen des Cubes durch Regeln
- ermöglicht:
  - Basis-Cube-Operationen: Push, Pull, Slicing, Dicing, Roll-Up, ...
  - komplexe Aggregationen

Das Datenmodell (1)

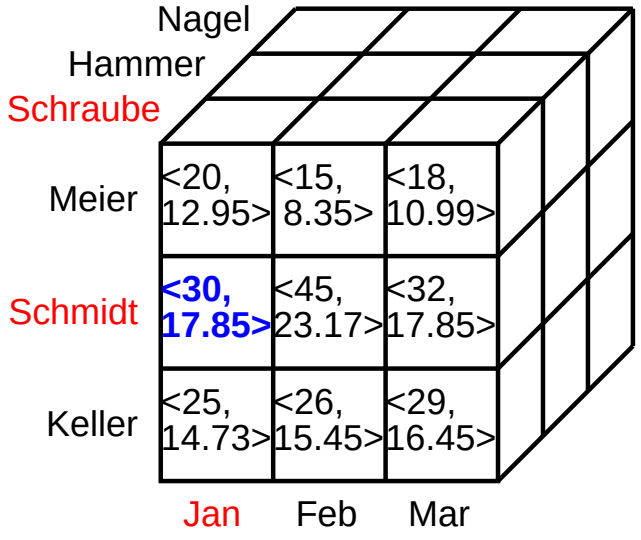
Definitionen:

- Name:
  Konstante oder Variable
- Zellenreferenz:
  $N(N_1, N_2, \ldots, N_p)$ mit N, N_1, N_2, ..., N_p Namen
  - N: Cube-Name
  - $N_1, \ldots, N_p$: Attribut-Namen (Dimensionen)
- Zelleninhalt:
  q-Tupel von Namen (Maße)

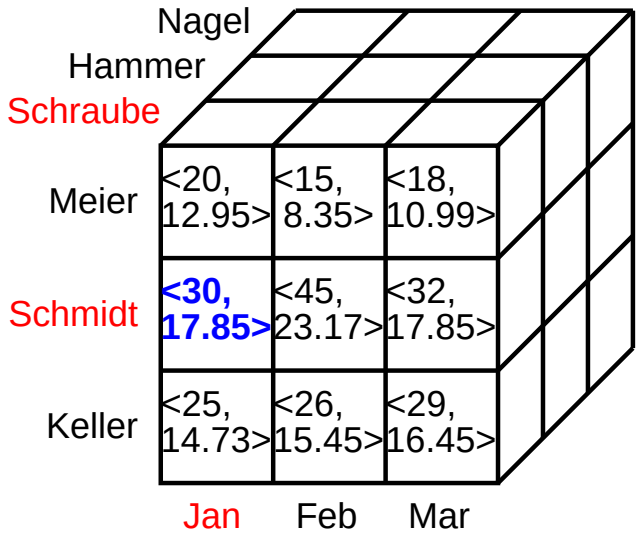
Das Datenmodell (2)

- Zelle:
  - über Zellenreferenz referenziert, enthält Zelleninhalt
  - dargestellt durch Grundatome der Form $N(N_1, N_2, \ldots, N_p) : \langle N_{p+1}, \ldots, N_{p+q} \rangle$
  - $N(N_1, N_2, \ldots, N_p)$: Zellenreferenz
  - $\langle N_{p+1}, \ldots, N_{p+q} \rangle$: Zelleninhalt
- Cube:
  - Menge von Grundatomen, gleicher Cube-Name
  - Monovaluation

Beispiel für das Datenmodell



Beispiel für das Datenmodell



Cube $C1$:

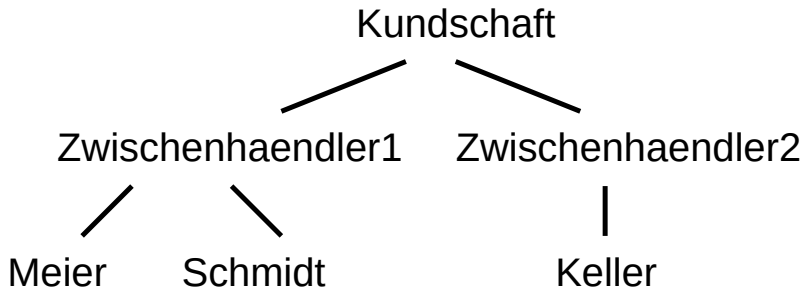
$$\{C1(Jan, Meier, Schraube) : \langle 20,\ 12.95\rangle,$$
$$C1(Feb, Meier, Schraube) : \langle 15,\ 8.35\rangle,$$
$$C1(Mar, Meier, Schraube) : \langle 18,\ 10.99\rangle,$$
$$C1(Jan, Schmidt, Schraube) : \langle 30, 17.85\rangle,$$
$$\ldots\}$$

Intuitive Bedeutung

- Regel:
  - Form: $head \longleftarrow body$
  - definieren neue Zellen durch existierende Zellen
  - die Regel $P(v) : \langle w \rangle \longleftarrow Q(x, y) : \langle z \rangle$ bedeutet informell:
    „***wenn*** *es eine Zelle mit Referenz* $Q(x, y)$
    und Inhalt z *gibt,*
    dann *gibt es auch eine Zelle mit Referenz* $P(v)$
    und Inhalt w“

Hierarchien

- werden für Aggregationen benötigt
- spezifiziert durch Gruppierungsatome mit *in*-Prädikat
- Beispiel:

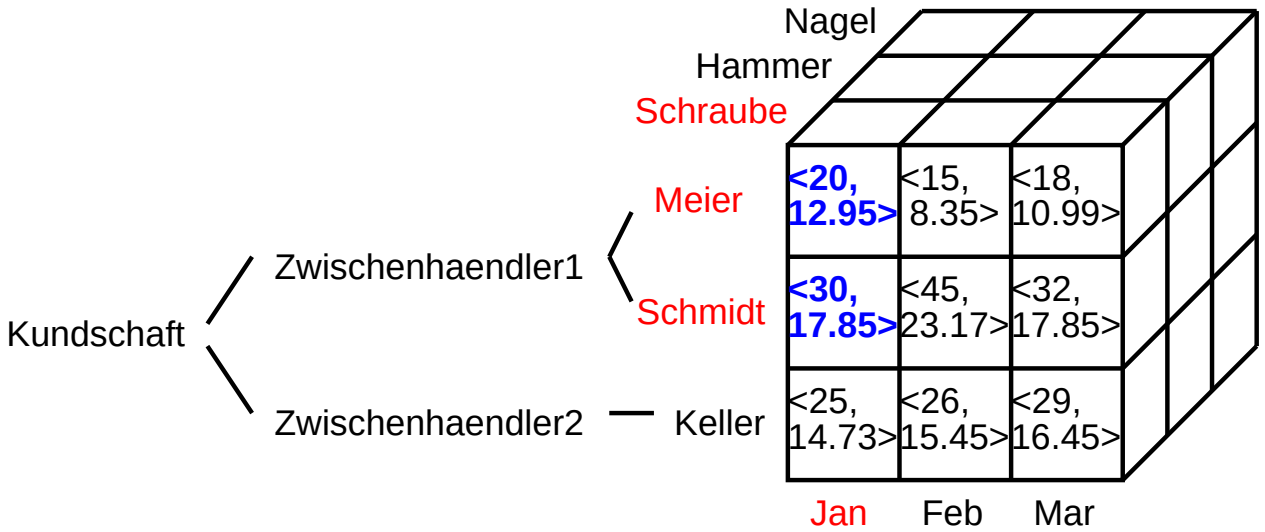


$$\{in(Meier, Zwischenhaendler1),$$
$$in(Schmidt, Zwischenhaendler1),$$
$$in(Keller, Zwischenhaendler2),$$
$$in(Zwischenhaendler1, Kundschaft),$$
$$in(Zwischenhaendler2, Kundschaft)\}$$

Aggregation

- Aggregatsoperator:
  partielle Abbildung von Multimengen von Tupeln über Konstanten auf einen einzelnen Wert
  z. B. $sum, avg, \ldots$
- Aggregat-Teilziele der Form $name = f(reference)$
- Bedeutung:
  - Aufschlüsselung der Referenz des Aggregats in die detaillierteste Hierarchie-Ebene
  - Anwendung der Aggregatsfunktion auf die Inhalte der referenzierten Zellen

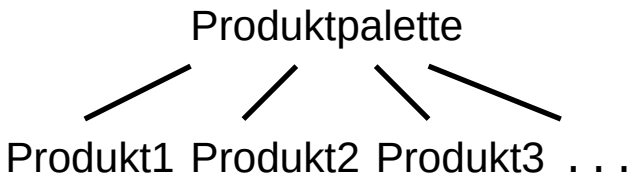
Beispiel: Aggregation

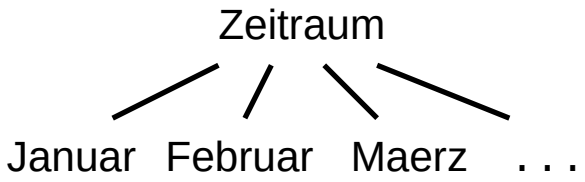


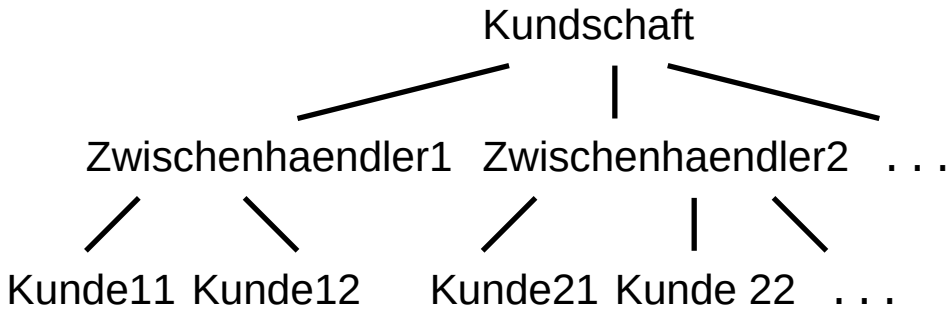
$$
\begin{aligned}
& sum(C1(Jan, Zwischenhaendler1, Schraube)) \\
= \quad & sum(C1(Jan, Meier, Schraube), C1(Jan, Schmidt, Schraube)) \\
= \quad & sum(\langle 20,\ 12.95\rangle, \langle 30,\ 17.85\rangle) = \langle 50,\ 30.80\rangle
\end{aligned}
$$

Beispieldatenbank

- Datenbank mit einem Cube, $C1$, mit Verkaufsdaten
- Zellen:
  $C1(monat, produkt, kunde) : \langle einheiten, einnahmen \rangle$
- Hierarchien:







Beispielanfragen: Push – Pull

- Push des Monats in die Zelleninhalte:

$$C2(monat, produkt, kunde) : \langle einheiten, einnahmen, monat \rangle$$
$$\longleftarrow C1(monat, produkt, kunde) : \langle einheiten, einnahmen \rangle$$

- Pull der Einheiten in die Zellenreferenz:

$$C3(monat, produkt, kunde, einheiten) : \langle einnahmen \rangle$$
$$\longleftarrow C1(monat, produkt, kunde) : \langle einheiten, einnahmen \rangle$$

- Im Folgenden:

$$C4(monat, produkt, kunde) : \langle einnahmen \rangle$$
$$\longleftarrow C1(monat, produkt, kunde) : \langle einheiten, einnahmen \rangle$$

Beispielanfrage: Data Cube

- erweiterter Data Cube: mit Hierarchien
- Regeln:

$$
\begin{aligned}
& CubeOperatorErgebnis(monat, produkt, kunde) : \langle s \rangle \\
\longleftarrow \quad & s = sum(C4(monat, produkt, kunde)), \\
& Ebene(monat) = \langle\rangle, \\
& Ebene(produkt) = \langle\rangle, \\
& Ebene(kunde) = \langle\rangle
\end{aligned}
$$

$$
\begin{aligned}
Ebene(x) : \langle\rangle \quad &\longleftarrow \quad in(x, y), \\
Ebene(y) : \langle\rangle \quad &\longleftarrow \quad in(x, y).
\end{aligned}
$$

Gliederung

1. Data Cube Operator
2. Regel-basierte Sprachen
3. **MultiDimensional eXpressions (MDX)**
4. *n*D-SQL

3. MDX

- Microsoft-Entwicklung: MS-SQL-Server
- Syntax an SQL angelehnt
- Datenstruktur: Hypercubes
- Bestandteile:
  - Data Definition Language (DDL)
  - Data Manipulation Language (DML)

Terminologie

- Dimension
  - Hierarchie von Kategorien
- Mitglied
  - Elemente der untersten Hierarchiebene
- Maße
  - numerische Werte
  - in Zellen gespeichert
  - eigene Dimension `Measures`

allgemeine Struktur

- einfacher, zweidimensionaler MDX-Ausdruck:

```
SELECT axis_specification ON COLUMNS,
       axis_specification ON ROWS
  FROM cube_name
 WHERE slicer_specification
```

- Bedeutung:
  - `axis_specification`: Auswahl der Mitglieder für die Ergebnisachsen
  - `cube_name`: Auswahl des Cube
  - `slicer_specification`: Auswahl des Cube-Ausschnittes ($\rightarrow$ Slicing)

Beispiele (1)

- Anfrage:

```
SELECT Measures.MEMBERS ON COLUMNS,
       [Store].MEMBERS  ON ROWS
  FROM [Sales]
```

- Ergebnis:
  - 2-dimensionale Tabelle
    - Spalten: alle Maße des Cube `Sales`
    - Zeilen: alle Geschäfte des Cube `Sales`
  - über alle nicht spezifizierten Dimensionen wird summiert
  - zusätzliche Summen für alle Hierarchiebenen der Dimension `Store`

Beispiele (2)

- Anfrage:

```
SELECT Measures.MEMBERS ON COLUMNS,
       {[Store].[Store State].[CA],
        [Store].[Store State].[WA]} ON ROWS
  FROM [Sales]
```

- Ergebnis:
  - 2-dimensionale Tabelle
    - Spalten: alle Maße des Cube `Sales`
    - Zeilen: Kalifornien und Washington
  - über alle nicht spezifizierten Dimensionen wird summiert

Beispiele (3)

- Anfrage:

```
SELECT {[Store Type].[Store Type].MEMBERS} ON COLUMNS,
       {[Store].[Store State].MEMBERS) ON ROWS
  FROM [Sales]
 WHERE (Measures.[Sales Units], [Time].[Year].[2003])
```

- Ergebnis:
  - 2-dimensionale Tabelle
    - Spalten: Elemente der `Store Type` Ebene
    - Zeilen: Elemente der `Store State` Ebene
  - in den Zellen steht das Maß `Sales Units`
  - Summierung über nicht spezifizierte Dimensionen, eingeschränkt auf das Jahr 2003 in der `Time` Dimension

Gliederung

1. Data Cube Operator
2. Regel-basierte Sprachen
3. MultiDimensional eXpressions (MDX)
4. ***n*D-SQL**

4. *n*D-SQL

- vorgestellt 1998 von Gingras und Lakshmanan
- Erweiterung von Standard-SQL
- arbeitet auf relationalen Datenbanken
- Möglichkeiten:
  - föderierte Datenbanken (heterogene Schemata)
  - Aggregation auf mehreren Granularitätsebenen

Zusammenfassung

	Data Cube	**Regel-basierte Sprache**	**MDX**	***n*D-SQL**
Syntax	SQL-Erweiterung	Regeln	SQL-ähnlich	SQL-Erweiterung
Datenmodell	Relationen	Cubes	Cubes	föderierte relationale Datenbanken
Entwicklung	kommerzielle Forschung	Wissenschaft	Microsoft	Wissenschaft