



**Übung zur Vorlesung**  
***Einsatz und Realisierung von Datenbanksystemen im SoSe16***

Moritz Kaufmann (moritz.kaufmann@tum.de)  
<http://db.in.tum.de/teaching/ss16/impldb/>

**Blatt Nr. 06**

**Hausaufgabe 1**

Gegeben eine Datenbank mit dem Schema aus Abbildung 1. Bestimmen Sie die Wahlbeteiligung bei der Bundestagswahl 2005. Die Wahlbeteiligung soll sowohl für ganz Deutschland, wie auch aufgeschlüsselt nach den Bundesländern, nach Wahlkreisen und sogar nach Wahllokalen abrufbar sein. Geben Sie SQL unter Verwendung des CUBE Operators an. Ist die Nutzung des CUBE Operators in diesem Beispiel optimal?

Wahlbezirke			
Nr	Wahlberechtigte	Wahllokal	Wahlkreis
21967	4500	Rathaus	26
21921	6700	Bürgerhaus	153
28424	3400	Gymnasium	213

Wahlkreise		
Nr	Bezeichnung	Bundesland
26	Neuholm	NRW
153	Oberbach	Bayern
213	Berch	Hessen

Direktkandidaten			
SozialVNr	Name	Partei	Wahlkreis
2005-DK01	Meier	SPD	26
2005-DK02	Müller	CDU	26
2005-DK03	Schmidt	FDP	26
2005-DK04	Huber	null	26

Erststimmen			
Wahlbezirk	Jahr	Kandidat	Stimmen
21967	2005	2005-DK01	450
21967	2005	2005-DK02	750
21967	2005	2005-DK03	600

Bundesländer	
Name	Einwohner
Bayern	12.000.000
NRW	18.000.000
Hessen	6.100.000

Zweitstimmen			
Wahlbezirk	Jahr	Partei	Stimmen
21967	2005	CDU	535
21967	2005	SPD	252
21967	2005	FDP	363
21967	2005	B'90/Grüne	377
21921	2005	CSU	439

Parteien	
Name	Mitglieder
CDU	580.000
CSU	170.000
SPD	600.000
FDP	67.000
B'90/Grüne	44.000

Abbildung 1: Schema des Wahlsystems

Die Lösung für die Formulierung der Anfrage mithilfe des CUBE operators ist im Lösungsbuch.

Ergänzend noch die Antwort zur Frage ob der CUBE Operator für diese Anfrage optimal ist: Der CUBE Operator wird typischerweise mit unabhängigen Dimensionen genutzt. Der Operator berechnet dann für alle Kombinationsmöglichkeiten der Dimensionen für die Aggregation. Für die Aufgabe ergeben sich damit folgende Kombinationen:

{Bundesländer, Wahlkreise, Wahllokale}, {Bundesländer, Wahlkreise}, {Bundesländer, Wahllokale}, {Wahlkreise, Wahllokale}, {Bundesländer}, {Wahlkreise}, {Wahllokale}, {}

Da in diesem Beispiel die verschiedenen CUBE Dimensionen nicht unabhängig sind, sondern hierarchisch aufgebaut sind, finden redundante Berechnungen statt. Bundesländer, Wahlkreise und Wahlkreise produzieren beide jeweils gleichgroße Ergebnismengen mit den gleichen aggregierten Werten. Bei hierarchischen Daten wie in diesem Beispiel wäre der ROLLUP operator besser geeignet.

**Hausaufgabe 2** Lösen sie folgende Anfrage mit SQL basierend auf dem bekannten Universitätschema.

1. Bestimmen Sie die Durchschnittsnote für jeden Studenten.
2. Basierend auf dieser Durchschnittsnote, bestimmen Sie für alle Studenten ihren Rangplatz innerhalb ihrer Kohorte (Studenten desselben Semesters).
3. Berechnen Sie zusätzlich für jeden Studenten auch noch die ABWEICHUNG seiner Durchschnittsnote von der Durchschnittsnote der Kohorte (also vom Durchschnitt der Durchschnittsnote der Studenten der Kohorte) ausgegeben werden.

Lösen Sie Teilaufgaben b und c jeweils einmal mit und einmal ohne Nutzung von Windowfunktionen. Ihre Anfragen können Sie auf [hyper-db.de](http://hyper-db.de) testen. Nutzen sie folgende erweiterte *pruefen* Relation:

```
with mehr_pruefen(MatrnNr, VorlNr, PersNr, Note) as (  
  select * from pruefen  
  union  
  values (29120,0,0,3.0), (29555,0,0,2.0), (29555,0,0,1.3), (29555,0,0,1.0)  
)
```

Ohne Windowfunktionen:

```
with mehr_pruefen(MatrnNr,VorlNr,PersNr>Note) as (  
select * from pruefen  
union  
values (29120,0,0,3.0), (29555,0,0,2.0), (29555,0,0,1.3), (29555,0,0,1.0)  
),  
noten(MatrnNr,Semester>Note) as (  
select s.matrnNr, semester, avg(Note)  
from studenten s, mehr_pruefen p  
where s.matrnNr=p.matrnNr  
group by s.matrnNr,semester  
)  
select *,  
(select count(*)+1 from noten x  
where x.Semester=n.Semester and x.Note<n.Note) as Rang,  
(select avg(x.Note) from noten x  
where x.Semester=n.Semester) as GPA,  
(select avg(x.Note) from noten x  
where x.Semester=n.Semester) - note as Abweichung  
from noten n order by semester, rang
```

Mit Windowfunktionen:

```
with mehr_pruefen(MatrnNr,VorlNr,PersNr>Note) as (  
select * from pruefen  
union  
values (29120,0,0,3.0), (29555,0,0,2.0), (29555,0,0,1.3), (29555,0,0,1.0)  
),  
noten(MatrnNr,Semester>Note) as (  
select s.matrnNr, semester, avg(Note)  
from studenten s, mehr_pruefen p  
where s.matrnNr=p.matrnNr  
group by s.matrnNr,semester  
)  
select *,  
rank() over (partition by Semester order by Note desc) as Rang,  
avg(Note) over (partition by Semester) as GPA,  
avg(Note) over (partition by Semester) - note as Abweichung  
from noten order by semester, rang
```

### Hausaufgabe 3

Die in Abbildung 2 dargestellten Relationen Mietspiegel und Kindergarten dienen der Bewertung von Wohngegenden im Großraum München. Für eine junge Familie ist ausschlaggebend, wie hoch die Lebenshaltungskosten gemessen an zu zahlender Miete und zu entrichtender Gebühr für den Kindergarten im jeweiligen Wohnort ausfallen. Illustrieren Sie die Ausführung einer Top-1-Berechnung (zur Bestimmung des günstigsten Wohnorts) für eine junge Familie mit zwei Kindern. Zeigen Sie die phasenweise Berechnung des Ergebnisses jeweils mit dem Threshold- und dem NRA-Algorithmus.

Siehe Lösungsbuch

Mietspiegel		Kindergarten		WohnLage	
Ort	Miete	Ort	Beitrag	Ort	Lage
Garching	800	Grünwald	-100	Grünwald	München-Süd
Ismaning	900	Unterföhring	0	Unterföhring	München-Nord
Unterföhring	1000	Bogenhausen	100	Ismaning	München-Nord
Nymphenburg	1500	Ismaning	200	Garching	München-Nord
Bogenhausen	1600	Garching	250	Bogenhausen	München-City
Grünwald	1700	Nymphenburg	300	Nymphenburg	München-City

Abbildung 2: Münchner Wohnlagen zur Berechnung der monatlichen Kosten für eine Familie.