



Übung zur Vorlesung *Einsatz und Realisierung von Datenbanksystemen* im
SoSe17

Maximilian E. Schüle (schuele@in.tum.de)
<http://db.in.tum.de/teaching/ss17/impldb/>

Blatt Nr. 07

Hausaufgabe 1 Cubes mit Studenten

Erstellen Sie anhand des Universitätsschemas einen Cube mit der Faktentabelle *mehr_pruefen* und den Dimensionstabellen *Vorlesungen*, *Professoren*, *Studenten*. Zählen Sie das Auftreten.

```
with mehr_pruefen(MatrnNr,VorlNr,PersNr>Note) as (  
    select * from pruefen  
    union  
    values (28106,5041,2137,3.0)  
)  
  
create table hoerentotal (vtitel varchar(30), pname varchar(30),  
    sname varchar(30), all integer);  
  
insert into hoerentotal(  
    select v.titel as VTitel, p.Name as PName, s.Name as SName,  
        count(*) as all  
    from mehr_pruefen pr, Vorlesungen v, Professoren p, Studenten s  
    where p.PersNr = pr.PersNr AND s.MatrnNr=pr.MatrnNr AND v.VorlNr=pr.VorlNr  
    group by cube(v.titel, p.Name, s.Name)  
);
```

Ohne Cube-Operator entspricht die Abfrage bei d Dimensionen einer Vereinigung aus 2^d Abfragen (und somit $2^d - 1$ mal „UNION“), in diesem Fall bei drei Dimensionen acht Teilabfragen. Dabei geht man am besten strukturiert vor:

i	$[i]_2$	VTitel	PName	SName
0	000	0	0	0
1	001	0	0	1
2	010	0	1	0
3	011	0	1	1
4	100	1	0	0
5	101	1	0	1
6	110	1	1	0
7	111	1	1	1

Jedes Bit sagt aus, ob nach dem entsprechenden Attribut gruppiert wird oder nicht. Wenn danach nicht gruppiert wird, so ist der entsprechende Wert auf `null` zu setzen und bei Zahlen zu casten (`to_number(null)`).

```

create table hoerentotal (vtitel varchar(30), pname varchar(30),
    sname varchar(30), all integer);

insert into hoerentotal(
    select null as VTitel, null as PName, null as SName,
        count(*) as all
    from mehr_pruefen pr, Vorlesungen v, Professoren p, Studenten s
    where p.PersNr = pr.PersNr AND s.MatrNr=pr.MatrNr AND v.VorlNr=pr.VorlNr
    -- keine Gruppierung
UNION
    select null, null, s.Name as SName,
        count(*) as all
    from mehr_pruefen pr, Vorlesungen v, Professoren p, Studenten s
    where p.PersNr = pr.PersNr AND s.MatrNr=pr.MatrNr AND v.VorlNr=pr.VorlNr
    group by s.Name
UNION
    select null, p.Name as PName, null,
        count(*) as all
    from mehr_pruefen pr, Vorlesungen v, Professoren p, Studenten s
    where p.PersNr = pr.PersNr AND s.MatrNr=pr.MatrNr AND v.VorlNr=pr.VorlNr
    group by p.Name
UNION
    select null, p.Name as PName, s.Name as SName,
        count(*) as all
    from mehr_pruefen pr, Vorlesungen v, Professoren p, Studenten s
    where p.PersNr = pr.PersNr AND s.MatrNr=pr.MatrNr AND v.VorlNr=pr.VorlNr
    group by p.Name, s.Name
UNION
    select v.titel as VTitel, null, null
        count(*) as all
    from mehr_pruefen pr, Vorlesungen v, Professoren p, Studenten s
    where p.PersNr = pr.PersNr AND s.MatrNr=pr.MatrNr AND v.VorlNr=pr.VorlNr
    group by v.titel
UNION
    select v.titel as VTitel, null, s.Name as SName,
        count(*) as all
    from mehr_pruefen pr, Vorlesungen v, Professoren p, Studenten s
    where p.PersNr = pr.PersNr AND s.MatrNr=pr.MatrNr AND v.VorlNr=pr.VorlNr
    group by v.titel, s.Name
UNION
    select v.titel as VTitel, p.Name as PName, null,
        count(*) as all
    from mehr_pruefen pr, Vorlesungen v, Professoren p, Studenten s
    where p.PersNr = pr.PersNr AND s.MatrNr=pr.MatrNr AND v.VorlNr=pr.VorlNr
    group by v.titel, p.Name
UNION
    select v.titel as VTitel, p.Name as PName, s.Name as SName,
        count(*) as all
    from mehr_pruefen pr, Vorlesungen v, Professoren p, Studenten s
    where p.PersNr = pr.PersNr AND s.MatrNr=pr.MatrNr AND v.VorlNr=pr.VorlNr
    group by v.titel, p.Name, s.Name
);

```

Hausaufgabe 2

Schreiben Sie eine SQL-Anfrage, die basierend auf dem Schema aus Abbildung 1 einen dreidimensionalen Quader berechnet, der es unserem Handelsunternehmen erlaubt, entlang der folgenden Dimensionen drill-down/roll-up Anfragen zu stellen:

- Produkttyp,
- Bezirk,
- Alter.

Das Handelsunternehmen ist dabei nur an Daten aus Deutschland interessiert, die in die Hochsommersaison fallen. Verwenden Sie den **cube**-Operator.

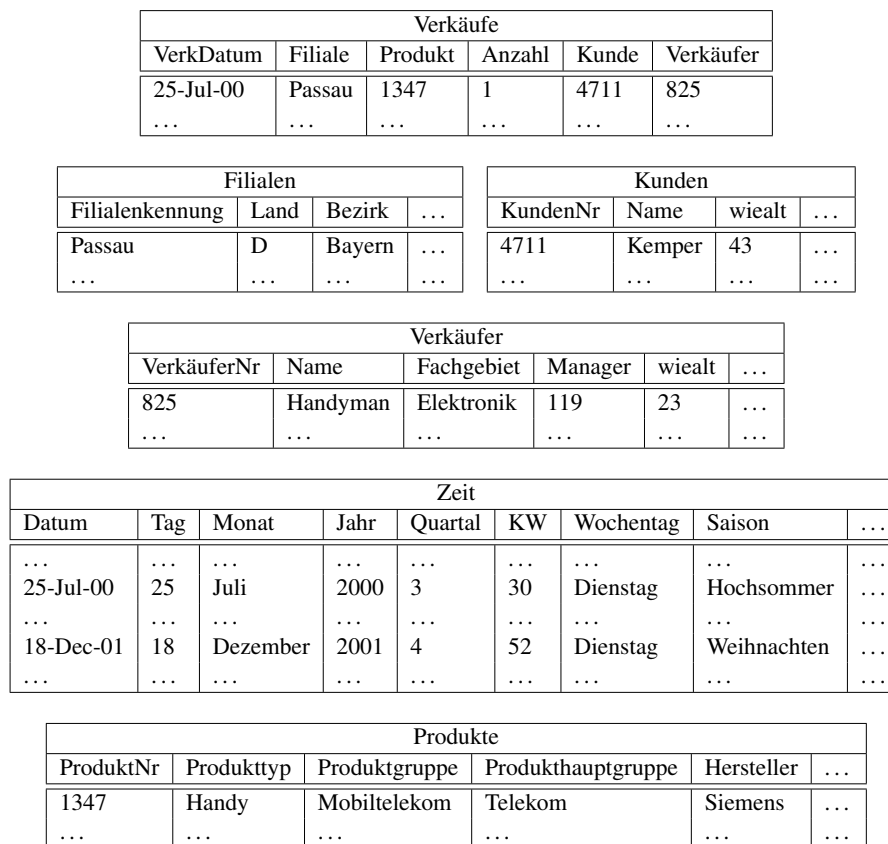


Abbildung 1: Schema des Handelsunternehmens.

Siehe Übungsbuch

Hausaufgabe 3 Fensterln mit SQL

Analysieren wir die Gehälter von Professoren mittels Windowfunctions und führen Sie die Abfragen unter hyper-db.de aus. Dazu orientieren wir uns an der Relation *Professoren* des erweiterten Universitätsschemas:

```
with Professoren (persnr, name, rang, raum, gehalt, steuerklasse) as (
  values (2125,'Sokrates','C4',226,85000,1),
```

```
(2126, 'Russel', 'C4', 232, 80000, 3),
(2127, 'Kopernikus', 'C3', 310, 65000, 5),
(2128, 'Aristoteles', 'C4', 250, 85000, 1),
(2133, 'Popper', 'C3', 52, 68000, 1),
(2134, 'Augustinus', 'C3', 309, 55000, 5),
(2136, 'Curie', 'C4', 36, 95000, 3),
(2137, 'Kant', 'C4', 7, 98000, 1)
```

)

1. Ermitteln Sie zu jedem Professor das Durchschnittsgehalt aller Professoren.

```
SELECT *, avg(gehalt) OVER () FROM Professoren;
```

2. Ermitteln Sie zu jedem Professor das Durchschnittsgehalt aller Professoren partitioniert nach Rang.

```
SELECT *, avg(gehalt) OVER (PARTITION BY rang) FROM Professoren;
```

3. Ermitteln Sie nun die wachsende Summe (das Quantil) des Gehaltes aller Professoren partitioniert nach Rang und absteigend sortiert nach ihrem Gehalt. Gleich verdienende Professoren sind im selben Quartil.

```
SELECT *, sum(gehalt) OVER (PARTITION BY rang ORDER BY gehalt DESC)
FROM Professoren;
```

-- äquivalent zu

```
SELECT *, sum(gehalt) OVER (PARTITION BY rang ORDER BY gehalt DESC
RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) FROM Professoren;
```

4. Ermitteln Sie nun die wachsende Summe des Gehaltes aller Professoren partitioniert nach Rang und absteigend (total) sortiert nach ihrem Gehalt (reihenweise, nicht als Range-Query).

```
SELECT *, sum(gehalt) OVER (PARTITION BY rang ORDER BY gehalt DESC
ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) FROM Professoren;
```

5. Ermitteln Sie nun das gleitende Durchschnittsgehalt aus genau zwei mehr bzw. weniger verdienenden Professoren sortiert nach Gehalt und partitioniert nach Rang.

```
SELECT *, avg(gehalt) OVER (PARTITION BY rang ORDER BY gehalt DESC
ROWS BETWEEN 2 PRECEDING AND 2 FOLLOWING) FROM Professoren;
```

6. Ermitteln Sie nun das gleitende Durchschnittsgehalt aus den 500 Einheiten mehr bzw. weniger verdienenden Professoren sortiert nach Gehalt und partitioniert nach Rang. (weder in PostgreSQL noch in HyPer implementiert, fragen Sie Ihren Tutor)

```
SELECT *, avg(gehalt) OVER (PARTITION BY rang ORDER BY gehalt DESC
RANGE BETWEEN 500 PRECEDING AND 500 FOLLOWING) FROM Professoren;
```

7. Geben sie zu jedem Professor das Gehalt des eins besser wie eins schlechter verdienenden.

```
SELECT *, lag(gehalt) OVER (ORDER BY gehalt DESC),
       lead(gehalt) OVER (ORDER BY gehalt DESC) FROM Professoren;
```

8. Ermitteln Sie die drei bestverdienendsten Professoren einmal mit und einmal ohne Windowfunctions.

```
SELECT * FROM (
    SELECT *, rank() OVER (ORDER BY gehalt desc) FROM Professoren
) WHERE rank < 4
```

```
SELECT * FROM professoren p_selbst WHERE 3 > (SELECT count(*)
FROM professoren p_reich WHERE p_selbst.gehalt < p_reich.gehalt)
```

Hausaufgabe 4 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 2 und 3 jeweils einmal mit und einmal ohne Nutzung von Windowfunktionen. Ihre Anfragen können Sie auf 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 asc) 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 5 Germany - bad, very bad!

Ermitteln Sie den Handelsüberschuss zwischen Deutschland und den USA auf Basis des Schneeflocken-Schemas des TPC-H-Benchmarks. Orientieren Sie sich an der TPC-H-Abfrage 7 und nutzen Sie hyper-db.de.

```

select
    supp_nation,
    cust_nation,
    l_year,
    sum(volume) as revenue
from
    (
    select
        n1.n_name as supp_nation,
        n2.n_name as cust_nation,
        extract(year from l_shipdate) as l_year,
        l_extendedprice * (1 - l_discount) as volume
    from
        supplier,
        lineitem,
        orders,
        customer,
        nation n1,
        nation n2
    where
        s_suppkey = l_suppkey
        and o_orderkey = l_orderkey
        and c_custkey = o_custkey
        and s_nationkey = n1.n_nationkey
        and c_nationkey = n2.n_nationkey
        and (
            (n1.n_name = 'UNITED STATES' and n2.n_name = 'GERMANY')
            or (n1.n_name = 'GERMANY' and n2.n_name = 'UNITED STATES')
        )
        and l_shipdate between date '1995-01-01' and date '1996-12-31'
    ) as shipping
p by
    supp_nation,
    cust_nation,
    l_year
order by
    supp_nation,
    cust_nation,
    l_year

```

Hausaufgabe 6

Berechnen Sie, wie groß ein Data Warehouse für ein Handelsunternehmen wie Zalando oder Amazon.com wäre, wenn die Bestelldaten der letzten 3 Jahre enthalten sind. Versuchen sie ihre Schätzung basierend auf öffentlichen Daten wie Umsatz o.Ä. durchzuführen.

Siehe Übungsbuch.