

# Datenbanksysteme: Übung 5

Tutor: Miao Wang

Lisa Dohrmann (4130066), Adrian Neumann (4140810), Naja v. Schmude (4127652)

26. Mai 2008

## Aufgabe 1

- `Select M.Vorname, M.Nachname from M Natural Join B`
- `Select M.Vorname, M.Nachname from M Except  
(Select * from M Join B on (M.personalnummer=B.personalnummer));`
- `Select M.Vorname, M.Nachname, Count(B.personalnummer) from  
M Left Outer Join B on M.personalnummer=B.personalnummer;`
- `Select M.Vorname, M.Nachname from M Natural Join  
(Select M.personalnummer, Count(M.personalnummer) as Zahl from M Natural Join  
(Select * from B Join O on O.observationsID=B.observationsID  
where stichwort='Pause' and konfidenz>0.8  
) as C Group By M.personalnummer  
) as D where D.Zahl>10;`
- `Select M.Vorname, M.Nachname from M Natural Join  
//gruppiert nach personalnummer  
(Select Count(personalnummer) as Num, * from B Natural Join  
//liebesbeziehungen:  
(Select * from O  
where stichwort in ('Techtelmechtel', 'Liebelei')) as P  
Group by personalnummer) as D where D.Num > 1;`

## Aufgabe 2

a) 31 Zeilen.

```
select distinct(r.name), country.name from (  
    geo_river gr join region r  
    on gr.c_id=r.c_id  
) join river on gr.river=river.name  
join country on r.c_id=country.c_id  
where river.name='Mekong';
```

b) 238 Zeilen

```
select city.name, country.name from city join country on  
city.c_id=country.c_id except  
select city.name, country.name from region join city on  
city.r_id=region.r_id join geo_sea on geo_sea.r_id=region.r_id  
join country on region.c_id=country.c_id;
```

- c) `select mountain, c.name from geo_mountain natural join  
(select name, c_id from country where type_of_gov='Volksrepublik') as c;`

mountain	name
Pik_Pobeda	Volksrepublik_China
Mount_Everest	Volksrepublik_China
Mount_Godwin_Austen	Volksrepublik_China
Tahat	Algerien
Ras_Daschan	Aethiopien
Uige	Angola

- d) `select m.mountain, co.name from country co join geo_mountain m on  
co.c_id=m.c_id natural join  
(select m.mountain from geo_mountain m join country co on  
m.c_id=co.c_id group by m.mountain  
having count(distinct co.name)>1) as foo  
order by m.mountain;`

mountain	name
Korab	Jugoslawien
Korab	Albanien
Llullaillaco	Argentinien
Llullaillaco	Chile
Nimba	Elfenbeinkueste
Nimba	Guinea
Pik_Pobeda	Volksrepublik_China
Pik_Pobeda	Sowjetunion
Sulitjelma	Schweden
Sulitjelma	Norwegen

- e) `select name from neighbor_of join country on country1=c_id  
where country2='S'  
union select name from neighbor_of join country on country2=c_id  
where country1='S';`

name
Finnland
Norwegen

- f) 45 Zeilen

```
select river.name, river.sea from river where
  (select count(*) from (
    select geo_river.r_id from geo_river where
      geo_river.river=river.name except
      (select geo_sea.r_id from geo_sea where geo_sea.sea=river.sea)
  ) as foo
)=(select count(*) from geo_river where geo_river.river=river.name)-1
and river.sea is not null;
```

```

g) select foo.city, re.name from (
    select city from
        located_at l join region r on
            l.r_id=r.r_id group by city
            having count(distinct r.r_id)>1
    ) as foo
    join located_at lo on foo.city = lo.city
    join region re on lo.r_id=re.r_id order by foo.city;

```

city		name
Oslo		Akershus
Oslo		Alaska
Oslo		Oslo
Victoria		British_Columbia
Victoria		Baja_California_Norte
Victoria		Seychellen

```

h) select r.sea from (
    select distinct count(city.name), located_at.sea from
        city join located_at on located_at.r_id=city.r_id where
            city.population > 1000000 and located_at.sea is not null
        group by located_at.sea
        order by count(city.name) desc limit 1
    ) c join (
        select count(sea), sea from river where
            sea is not null
        group by sea
        order by count(sea) desc limit 1) r
    on c.sea=r.sea;

```

sea
Atlantischer_Ozean

i) Das geht offenbar nicht, obwohl wir das halb aus dem Skript übernommen haben. (Wir benutzen Postgres)

```

with recursive Erreichbar (von, nach)
as (
    select country1 as von, country2 as nach from
        neighbor_of where country1 not in (select country2 from neighbor_of)
    )
union (
    select c1,c2 from neighbor_of n join
        Erreichbar e on n.country2=e.von
    )
select country.name from Erreichbar join Country on Erreichbar.nach=Country.c_id
where Erreichbar.von='D';

```