Dr. N. Ritter
Fachbereich Informatik
Arbeitsgruppe Datenbanken und Informationssysteme
Universität Kaiserslautern

Übungsblatt 5

für die Übung am Donnerstag, 11. Januar 2001, 15.30 Uhr in 36/265

Unterlagen zur Vorlesung: „http://wwwdbis.informatik.uni-kl.de/courses/DBS/“

Aufgabe 1: Algebraische Optimierung

Gegeben seien folgendes Datenbankschema und Kardinalitäten:

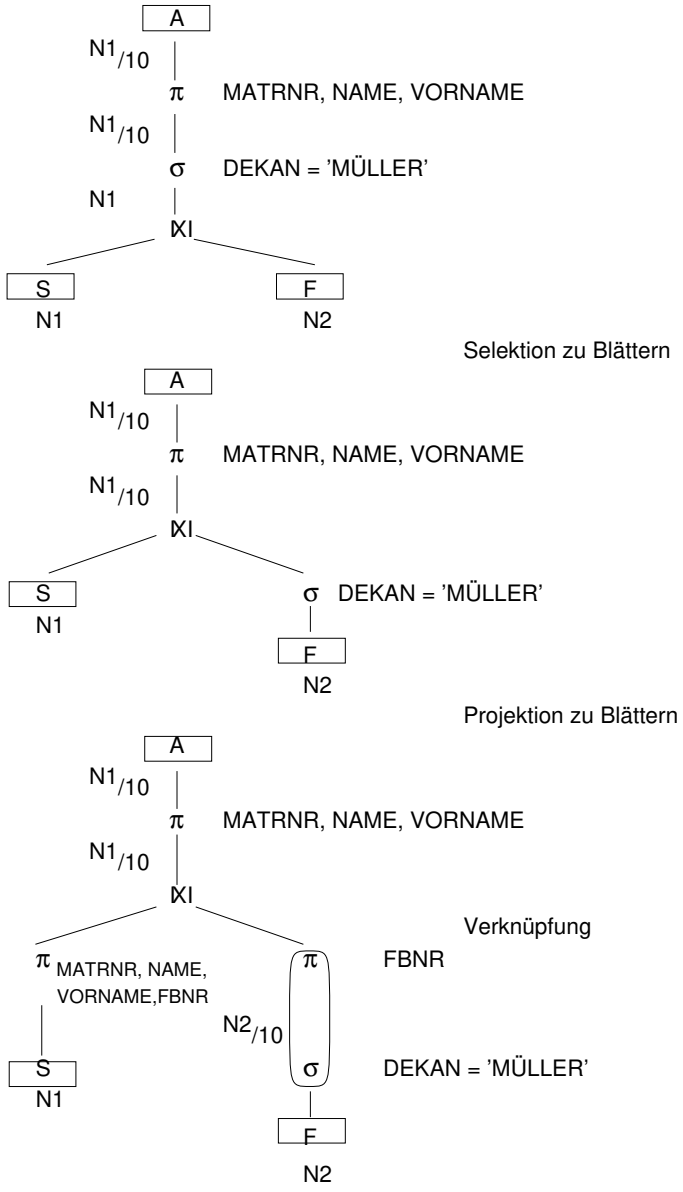
Professor: P (P_NAME, FBNR, WOHNORT)
Student: S (MATRNR, NAME, VORNAME, SEMANZ, WOHNORT, GEB-DATUM, FBNR)
Fachbereich: F (FBNR, F_NAME, ANZ_STUD, DEKAN)
Prüfung: PR (P_NAME, MATRNR, NOTE)

card (P)= 100; card (S)= 10^4 card (F)= 10 ; card (PR)= $4 \cdot 10^4$

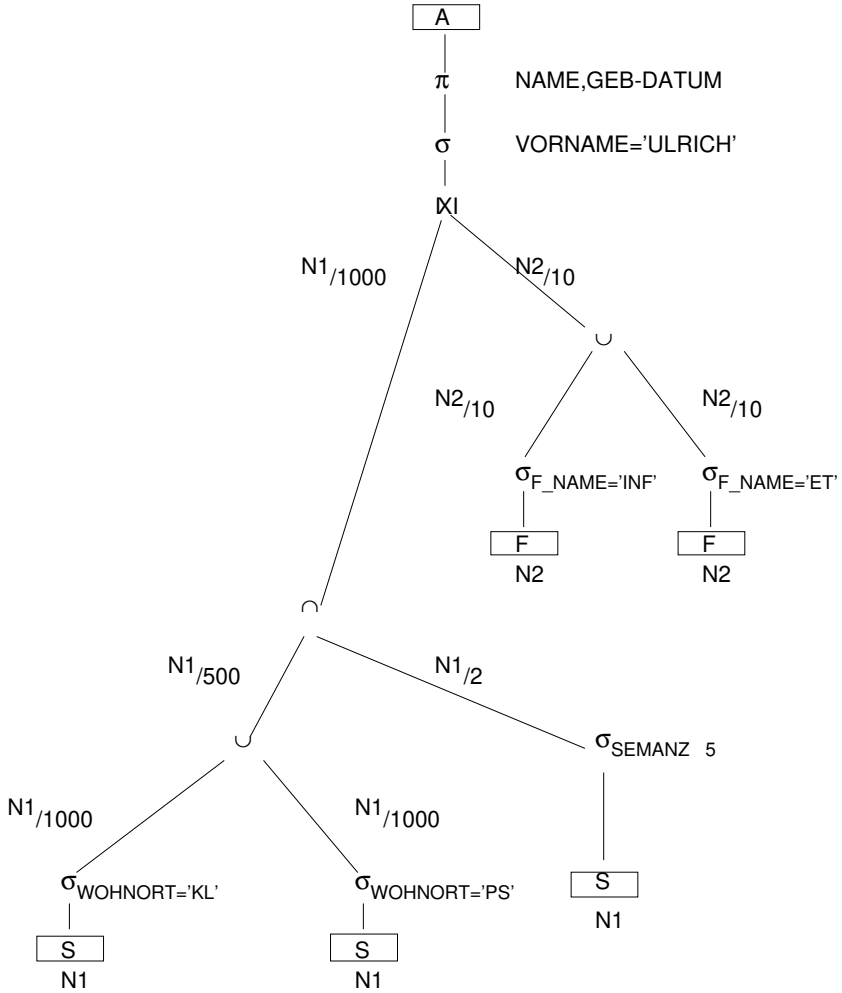
Annahmen: Alle Attribute haben die Länge 20 Byte, es gibt 1000 mögliche Wohnorte, 2000 Werte für Geb-Datum, Semester bewegt sich im Bereich 1–20, Note im Bereich 1–5 (ohne Zwischenstufen!). Name und Vorname sind in jeweils 300 Variationen vorhanden.

Für die nachfolgenden Anfragen soll jeweils eine algebraische Optimierung durchgeführt werden. Gehen Sie hierfür von dem Operatorbaum der vorgegebenen Anfrage aus. Bewerten Sie diesen wie in der Vorlesung gezeigt. Optimieren Sie anschließend diesen Operatorbaum, indem Sie (eine oder mehrere) Regeln aus der Vorlesung anwenden (welche?) und anhand der Bewertung zeigen, daß diese Optimierung eine Verbesserung bedeutet.

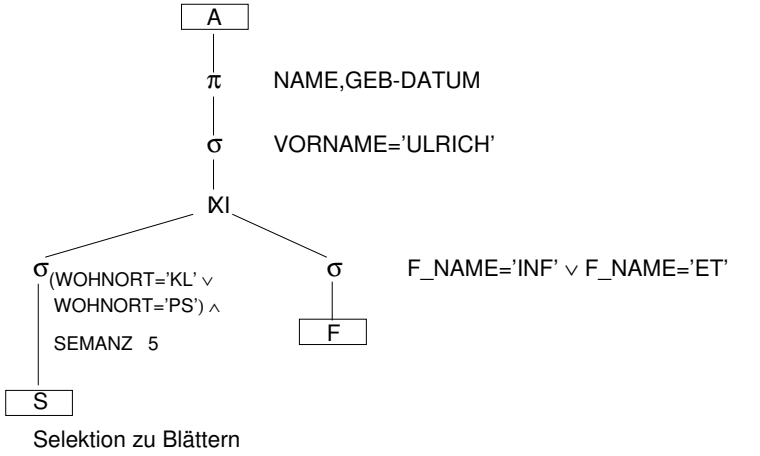
a) $\pi_{\text{MATRNR, NAME, VORNAME}}\ \sigma_{\text{DEKAN = 'MÜLLER'}}\ (S \mid X \mid F)$



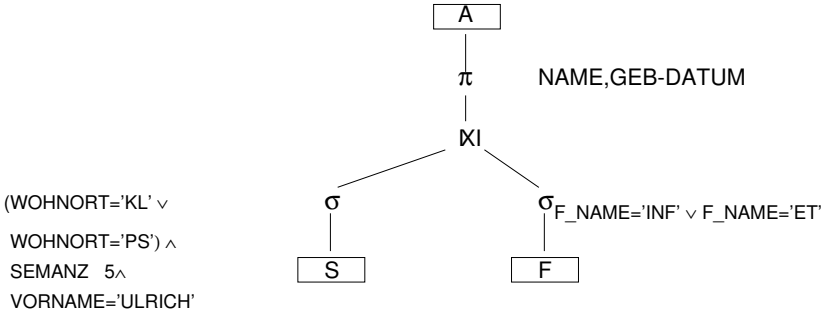
b) $\pi_{\text{NAME, GEB-DATUM}}\ \sigma_{\text{VORNAME='ULRICH'}}\ (((\sigma_{\text{WOHNORT='KL'}}\ S \cup \sigma_{\text{WOHNORT='PS'}}\ S) \cap \sigma_{\text{SEMANZ} \geq 5}\ S) \mid X \mid\ (\sigma_{\text{F_NAME='ET'}}\ F \cup \sigma_{\text{F_NAME='INF'}}\ F))$



Einfache Selektionen zusammenfassen, Teilbäume, die gemeinsam sind, nur einmal berechnen.

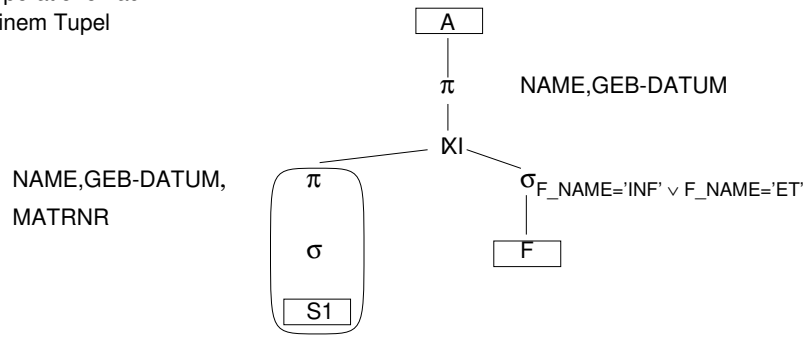


Selektion zu Blättern

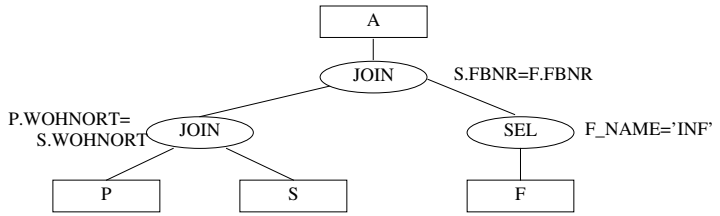


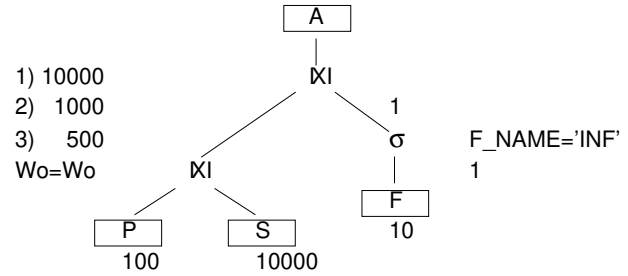
Projektion zu Blättern (Duplikate beachten)

Operationen auf einem Tupel

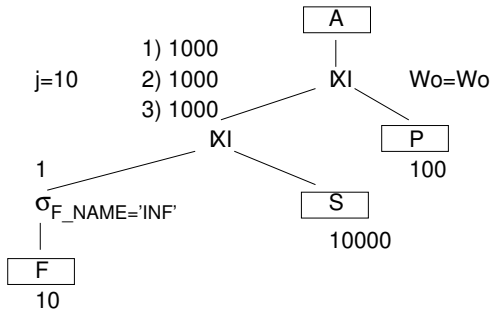


c) Betrac hten (und bewerten) Sie folgenden Operatorbaum, wenn das Attribut „Wohnort“ 100, 1000 oder 2000 W erte umfaßt.





alternativ:



Aufgabe 2: SQL-Anfragen

Gegeben sei die folgende Datenbank, in der die Vereine , Spiele , Trainer und die Spieler mit ihren Einsätzen für die laufende Saison verw altet werden:

VEREIN (VNAME, ORT, PRÄSIDENT)
SPIELE (HEIM, GAST, RESULTAT, ZUSCHAUER, TERMIN, SPIELTAG, H-TRAINER, G-TRAINER)
SPIELER (SPNR, NAME, VORNAME, VEREIN, ALTER, GEHALT, GEB-ORT)
TRAINER (TRNR, NAME, VORNAME, VEREIN, ALTER, GEHALT, GEB-ORT)
EINSATZ (HEIM, GAST, SPNR, VON, BIS, TORE, KARTE)

HEIM , GAST sind jeweils F remdsc hlüssel auf VNAME von VEREIN ,
H_TRAINER , G_TRAINER sind jeweils F remdsc hlüssel auf TRNR von TRAINER

Formulieren Sie folgende Anfragen in SQL. Wählen Sie für die Ausgabe sinnvolle Attribute aus . Geben Sie eine Lösung sowohl in symmetrisc her Notation als auc h in gesc hac htelter Abbildung an, sofern dies möglic h ist.

a) Welche Spieler sind beim einem V erein in Kaiserslautern besc häftigt?

b) Wie heißen die Präsidenten von Vereinen, die zuhause (mind.) ein Spiel mit 2:0 gewonnen haben?

c) Wieviele T ore wurden in der Saison gesc hossen?

d) Wieviele Spieler hat der 1. FCK in der Saison eingesetzt?

e) Wieviele T ore haben die Spieler des 1. FCK gesc hossen?

f) Welche Spieler haben noch nie gespielt?

g) Welche Spieler sind bei einem Verein beschäftigt, der in ihrem Geburtsort ansässig ist? Geben Sie den Namen des Spielers und des Vereins sowohl den Ort aus.

Aufgabe 3: SQL-Anfragen, Prüfungs-DB

Gegeben ist eine Datenbank mit folgendem relationalem Schema:

STUDENT: S (MATRNR, NAME, FBNR, SEMANZ, WOHNORT, ALTER)
FACHBEREICH: FB (FBNR, FBNAME, DEKAN-PNR, KAPAZITÄT, GEBÄUDE)
PROFESSOR: P (PNR, FBNR, NAME, ALTER, WOHNORT, BES-GRUPPE, FACH)
MÜNDL-PRÜFUNG: M-P (MATRNR, PNR, DATUM, NOTE)
DIPLOMARBEIT: DA (MATRNR, PNR, ANFANGS-DATUM, END-DATUM, THEMA, NOTE)

Fremdschlüssel sind „namensähnlich" zu den zugehörigen Schlüsselkandidaten.

Formulieren Sie folgende Anfragen/Aufgaben in SQL:

a) Welche Studenten, die älter als 25 Jahre sind, wohnen in Kaiserslautern? Gib von diesen alle Daten aus.

b) Erstelle eine Liste aller Namen von Studenten, die in einem Fachbereich eingeschrieben sind, der entweder in Gebäude 10 oder 12 angesiedelt ist.

c) Welche Studenten (MATRNR, NAME) sind in einem Fachbereich eingeschrieben, dessen Dekan kein C4-Professor ist?

d) Welche Studenten (MATRNR, NAME) studieren in einem Fachbereich in dem es Studenten gibt, die älter als 65 Jahre sind?

```
select   s.name
from     student s NATURAL JOIN fachbereich f
where    exists ( select   *
                  from     student as
                  where    (as.fbnr = f.fbnr) and
                           (as.alter > 65))
```

e) Welche Studenten (MATRNR, NAME) sind mehr als 3 Jahre älter als der Dekan ihres Fachbereichs?

Gib eine Statistik der durchschnittlich in den Fachbereichen erzielten Noten aus.

```
select   avg(note)
from     muendl-pruefung m, student s
where    s.matrnr = m.matrnr
group by s.fbnr
```

f) Erstelle eine Statistik mit dem Durchschnittsalter der Studenten der Fachbereiche.

```
select   avg (alter)
from     student s
group by s.fbnr
```

g) Welcher Professor im Fachbereich Mathematik hat die meisten Prüfungen abgehalten, wobei Wiederholungsprüfungen genau wie Erstprüfungen gezählt werden?

MAX(COUNT(*)) ist verboten, es muss deshalb noch eine weiter temp. Tabelle erstellt werden, die COUNT(*) und FB/PNR/Mp auffuehrt!!!

```
select   p
from     professor p, fachbereich f, muendl-pruefung m
where    (p.fbnr = f.fbnr) and
         (f.fbname = "mathematik) and
         (m.pnr = p.pnr)
group by p.pnr
having   count(*) = select   (max(count(mp))
                   from     muendl-pruefung mp, fachbereich f2,
                            professor p2
                   where    (mp.pnr = p.pnr) and
                            (p2.fbnr = f2.fbnr) and
                            (f2.fbname = "mathematik")
```

h) Welcher Professor im Fachbereich Mathematik hat die meisten Studenten geprüft?

MAX(COUNT(*)) verboten, temp . Tabelle notwendig!!!

```
select   p
from     professor p, fachbereich f
where    (p.fbnr = f.fbnr) and
         (f.fbname = "mathematik") and
         (( select   count(distinct(m.matrnr, m.pnr))
            from     muendl-pruefung m
            where    m.pnr = p.pnr
          ) =
          ( select   max(count(distinct(m.matrnr, m.pnr)))
            from     muendl-pruefung mp, professor pr
            where    (mp.pnr = pr.pnr) and
                     (pr.fbnr = f.fbnr)
            group by pr.pnr
          ))
```

i) Welche Studenten (NAME, MATRNR) aus Kaiserslautern haben mindestens 3 Prüfungen abgelegt und alle Prüfungen mit Note 1 bestanden?

```
select   s.name , s.matrnr
from     student s NA  TURAL JOIN muendl-pruefung m
where    not exists    (select  *
                       from m
                       where m.note <> 1 )
group by s.matrnr
ha ving  (count(*) >= 3)
```

j) Erstelle eine nac h PNR, FBNR geordnete Liste aller laufenden Diplomarbeiten (END-DA TUM = NULL) mit Studentenname und Thema der Diplomarbeit.

k) In welc hen Fac hbereic hen haben alle Studenten mindestens eine mündlic he Prüfung absolviert?

```
select   f
from     fachbereich f,
where    not exists  ( select   *
                       from     student s
                       where    not exists
                                ( select *
                                  from muendl-pruefung m
                                  where m.matrnr = s.matrnr ))
```


l) Welcher Professor (NAME, WOHNORT , BES-GRUPP) hat die absolut meisten Prüfungen mit der Note 5 abgehalten?
MAX(COUNT(*)) beac hten!!! T emp . Tabelle

```
select   p
from     professor p, muendl-pruefung m
where    (p.pnr = m.pnr) and
         (m.note = 5)
group by p.pnr
having   count (*) =  ( select max(count(pr))
                        from professor pr, muendl-pruefung mp
                        where pr.pnr = mp.pnr
                        group by pr.pnr)
```

m) Welche Studenten sind im 17.ten oder höheren Semester und haben noc h keine Diplomarbeit angefangen?

```
select   s
from     student s
where    (s.semanz >= 17) and
         not exists
         ( select *
           from diplomarbeit d
           where d.matrnr = s.matrnr)
```

n) Gib die Namen und Wohnorte der Professoren und Studenten aus , die in den letzten 10 Tagen eine Diplomarbeit ausgegeben bzw. angefangen haben, wobei die Studenten mindestens 3 Prüfungen abgelegt haben müssen und alle - auc h die über die 3 genannten hinausgehenden - Prüfungen mindestens mit der Note 2 bestanden haben müssen.

```
select   s.name, s.wohnort, p.name, p.wohnort
from     student s, professor f, diplomarbeit d
where    (d.matrnr = s.matrnr) and
         (d.pnr = p.pnr) and
         (d.anfangs-datum <= 10) and
         (d.matrnr is in  (select   matrnr
                           from     muendl-pruefung
                           group by matrnr
                           having   ((count(*) >= 3) and
                                     (set (note is contained in (1,2)))))
```

o) Welche Studenten, die zwisc hen 21 und 23 J ahre alt sind, aus den Fac hbe reic hen Physik oder Mathematik haben eine Diplomarbeit bei einem Professor des Fac h-bereic hs Informatik oder Elektrotec hnik begonnen oder ab geschlossen?

p) Welche Studenten haben zeitlich nach einer mündlichen Prüfung, die nicht bestanden wurde, bei demselben Professor eine Diplomarbeit angefangen?

q) Wie viele Studenten mit dem Anfangsbuchstaben B gibt es im Fachbereich Biologie, die in einer Stadt mit der Endsilbe Stadt wohnen.

```
select    count(*)
from      student
where     fbnr =    ( select    fbnr
                      from      fachbereich
                      where     fbname = 'biologie') and
          wohnort like '%stadt' and
          name like 'B%'
```

r) In welchem Fachbereich ist der Student eingeschrieben, von dem folgende Daten bekannt sind:
Sein Alter beträgt 23 bis 26 Jahre, er wohnt in Mannheim oder Ludwigshafen und sein Name beginnt mit den Buchstaben SCH gefolgt von zwei unbekannten Buchstaben. Der Name endet auf INER.

```
select    fbname, fbnr
from      fachbereich
where     fbnr is in ( select    fbnr
                       from      student
                       where     (alter between 23 and 26) and
                                 (wohnort = 'mannheim' or
                                  wohnort = 'ludwigshafen') and
                                 name like 'sch__iner'
```

s) Alle Studenten, die älter als 75 Jahre sind, werden exmatrikuliert. Dazu sind ihre Tupel aus allen betroffenen Relationen zu entfernen.

```
delete    from      diplomarbeit
where     matrnr in ( select    matrnr
                      from      student
                      where     alter > 75 )
```



```
delete    from      muendl-pruefung
where     ... (s.o.)

delete    from      student
where     ... (s.o.)
```

t) Alle Fachbereiche, die weniger als 10 eingeschriebene Studenten besitzen, werden aufgelöst, die Studenten werden in den Fachbereich Informatik versetzt. (Der Fachbereich Informatik hat mehr als 20 eingeschriebene Studenten).

```
update    student
set       fbnr =    (( select    fbnr
                       from      fachbereich
                       where     fbname = 'informatik')
                     where     fbnr in   ( select    fbnr
                                           from      student
                                           group by  fbnr
                                           having (count(*) < 10)))

delete    from      fachbereich
where     fbnr in   (( select    fbnr
                       from      fachbereich)
                       minus
                     ( select    fbnr
                       from      student))
```

u) Alle Studenten werden ein Jahr älter.

```
update    student
set       alter = alter + 1
```

Aufgabe 4: SQL-Anfragen, Personal-DB

Gegeben ist folgende Datenbank:

Angestellte: PERS (PNR, NAME, GEHALT, BERUF, ANR, MNR, ORT)
Abteilung: ABT (ANR, ANAME, AORT)
PERS.ANR ist Fremdschlüssel auf ABT.ANR, PERS.MNR ist Fremdschlüssel auf PERS.MNR

Formulieren Sie folgende Fragen in SQL:

a) Welche Angestellten aus Abteilung K55 verdienen genau das Durchschnittsgehalt aller Angestellten?

```
SELECT    PNR, NAME
FROM      PERS
WHERE     ANR = 'K55' AND
          GEHALT =  SELECT    AVG ( GEHALT )
                    FROM      PERS
```

b) Welche Angestellten verdienen mehr als ihre Manager?

```
SELECT   X.PNR, X.NAME
FROM     PERS X, PERS Y
WHERE    X.MNR = Y.PNR and
         X.GEHALT > Y.GEHALT
```

c) Welche Abteilungen in Frankfurt haben mehr als 10 Programmierer beschäftigt?

```
SELECT   A.ANR, A.ANAME
FROM     ABT A NATURAL JOIN PERS P
WHERE    A.AORT = 'FRANKFURT' AND
         P.BERUF = 'PROGRAMMIERER'
GROUP BY ANR
HAVING COUNT(*) > 10
```

d) Welche Abteilungen in Darmstadt haben nur Angestellte beschäftigt, die auch in Darmstadt wohnen?

```
SELECT   ANR, ANAME
FROM     ABT A
WHERE    AORT = 'DARMSTADT' AND
         ANR = ALL(SELECT ANR
                   FROM PERS P
                   WHERE P.ANR = A.ANR)
```

e) Liste Name, Gehalt und Abteilungsname der Programmierer auf, die in Darmstadt beschäftigt sind.

f) Liste Abteilungsnummern und Durchschnittsgehalt der Abteilungen auf.

```
SELECT   ANR, AVG ( GEHALT ), ANAME
FROM     ABT JOIN PERS USING ANR
GROUP BY ABT.ANR
ORDER BY ABT.ANR
```

g) Liste Abteilungsnummer und Abteilungsname solcher Abteilungen auf, in denen Angestellte im Durchschnitt weniger als 2500 verdienen.

```
SELECT   ANR, ANAME
FROM     ABT                                  oder symmetrisch
WHERE    ANR IN   (SELECT  ANR
                   FROM    PERS
                   GROUP BY ANR
                   HAVING  AVG(GEHALT) < 2500 )
```


h) Wieviele verschiedene Berufe sind in Abteilung K55 vertreten?

```
SELECT   COUNT(DISTINCT BERUF)
FROM     PERS
WHERE    ANR = 'K55'
```

i) In welchen Abteilungen sind alle Berufe der Unternehmung vertreten?

```
SELECT    ANR
FROM      PERS  P1
WHERE     NOT EXISTS
          (SELECT     *
           FROM       PERS P2
           WHERE      P2.ANR <> P1.ANR
                      NOT EXISTS
                      (SELECT     *
                      FROM        PERS P3
                      WHERE       P2.BERUF = P3.BERUF AND
                                  P1.ANR = P3.ANR
                      )
          )
```

j) Welche Abteilungen haben keine Angestellten?

```
SELECT *
FROM ABT X
WHERE    NOT EXISTS     (SELECT  *
                         FROM     PERS
                         WHERE    ANR = X.ANR)
```

k) Finde die Abteilungsnummern von Abteilungen in Darmstadt, in denen Angestellte weniger als 2000 verdienen.

l) Finde die Namen der Angestellten, die den gleichen Beruf und das gleiche Gehalt wie der Angestellte Müller haben.