

# Übung Datenbanksysteme

## Elitestudiengang Software Engineering

Andreas Kipf



# Klausurtermin

Mi, 24. Feb 2016, 10:30 Uhr

# Übung

# Aufgabe 1 (Alle Vorlesungen)

Mit not exists:

```
select s.MatrNr, s.Name from Studenten s where not exists
  (select * from Vorlesungen v where not exists
    (select * from hoeren h where h.VorlNr = v.VorlNr
      and h.MatrNr = s.MatrNr))
```

Mit count:

```
select s.MatrNr, s.Name from Studenten s,
  hoeren h, Vorlesungen v
where s.MatrNr = h.MatrNr and h.VorlNr = v.VorlNr
group by s.MatrNr, s.Name
having count(*) = (select count(*) from Vorlesungen)
```

## Aufgabe 2 (Median)

```
SELECT l.e FROM L l
WHERE
  (SELECT COUNT(*) FROM L WHERE e < l.e)
=
  (SELECT COUNT(*) FROM L WHERE e > l.e)
```

## Aufgabe 3 (Transitive Hülle)

Temporäre Views:

```
with ElterVon (Kind, Elter) as
  ((select Kind, Mutter as Elter from MutterVon)
   union
   (select Kind, Vater as Elter from VaterVon)),

Nachfahren (Nachf, Vorf) as
  ((select Kind as Nachf, Elter as Vorf from ElterVon)
   union all
   (select e.Kind as Nachf, n.Vorf
    from ElterVon e, Nachfahren n
    where e.Elter = n.Nachf))
```

Anfrage:

```
select Nachf from Nachfahren where Vorf = 'Kain'
```

## Aufgabe 3 (Transitive Hülle)

Temporäre Views:

```
with NachfahreBase (Nachf, Vorf) as
  ((select Mutter as Nachf, Kind as Vorf from MutterVon)
   union
   (select Vater as Nachf, Kind as Vorf from VaterVon)),

Nachfahren (Nachf, Vorf) as
  ((select Nachf, Vorf from NachfahreBase)
   union all
   (select base.Nachf, n.Vorf
      from NachfahreBase base, Nachfahren n
      where base.Vorf = n.Nachf))
```

Anfrage:

```
select Nachf from Nachfahren where Vorf = 'Kain'
```

## Aufgabe 4 (SQL)

### Gewichtung

- ▶ ProfGewicht:  $\frac{\emptyset_{\text{alleProfs}}}{\emptyset_{\text{prof}}}$
- ▶ VorlGewicht:  $\frac{\emptyset_{\text{alleVL}}}{\text{SWS}}$

```
create view ProfGewicht as
select p.PersNr, alle.ds / avg(p.Note) as gewicht
from pruefen p,
     (select avg(Note) as ds from pruefen) alle
group by p.PersNr, alle.ds;
```

```
create view VorlGewicht as
select v.VorlNr, alle.ds / cast(SWS as decimal) as gewicht
from Vorlesungen v,
     (select avg(SWS) as ds from Vorlesungen) alle;
```

```
select p.MatrNr, avg(p.Note * pg.gewicht * vg.gewicht)
from pruefen p, ProfGewicht pg, VorlGewicht vg
where p.PersNr = pg.PersNr and p.VorlNr = vg.VorlNr
group by p.MatrNr;
```



## Aufgabe 5 (Datenintegrität)

- ▶ Referenzen auf die Vorlesung 5041 in *hören*, *voraussetzen* und *prüfen*

## Aufgabe 5 (Datenintegrität)

- ▶ Referenzen auf die Vorlesung 5041 in *hören*, *voraussetzen* und *prüfen*
  - ▶ *hören*, *voraussetzen*: Kaskadierendes Löschen

## Aufgabe 5 (Datenintegrität)

- ▶ Referenzen auf die Vorlesung 5041 in *hören*, *voraussetzen* und *prüfen*
  - ▶ *hören*, *voraussetzen*: Kaskadierendes Löschen
  - ▶ *prüfen*: Kein Verhalten definiert  $\Rightarrow$  Löschen verboten

## Aufgabe 5 (Datenintegrität)

- ▶ Referenzen auf die Vorlesung 5041 in *hören*, *voraussetzen* und *prüfen*
  - ▶ *hören*, *voraussetzen*: Kaskadierendes Löschen
  - ▶ *prüfen*: Kein Verhalten definiert  $\Rightarrow$  Löschen verboten
- ▶ Es gibt keinen Professor mit PersNr 2138, *prüfen* hat jedoch den Constraint `references Professoren on delete ...`  
 $\Rightarrow$  Einfügen scheitert

## Aufgabe 5 (Datenintegrität)

- ▶ Referenzen auf die Vorlesung 5041 in *hören*, *voraussetzen* und *prüfen*
  - ▶ *hören*, *voraussetzen*: Kaskadierendes Löschen
  - ▶ *prüfen*: Kein Verhalten definiert  $\Rightarrow$  Löschen verboten
- ▶ Es gibt keinen Professor mit PersNr 2138, *prüfen* hat jedoch den Constraint `references Professoren on delete ...`  
 $\Rightarrow$  Einfügen scheitert
- ▶ Primärschlüssel von *prüfen* ist (MatrNr, VorlNr) und Tabelle hat bereits ein Tupel mit (28106, 5001) als Primärschlüssel  $\Rightarrow$  Einfügen scheitert

## Aufgabe 5 (Datenintegrität)

- ▶ Referenzen auf die Vorlesung 5041 in *hören*, *voraussetzen* und *prüfen*
  - ▶ *hören*, *voraussetzen*: Kaskadierendes Löschen
  - ▶ *prüfen*: Kein Verhalten definiert  $\Rightarrow$  Löschen verboten
- ▶ Es gibt keinen Professor mit PersNr 2138, *prüfen* hat jedoch den Constraint references Professoren on delete ...  $\Rightarrow$  Einfügen scheitert
- ▶ Primärschlüssel von *prüfen* ist (MatrNr, VorlNr) und Tabelle hat bereits ein Tupel mit (28106, 5001) als Primärschlüssel  $\Rightarrow$  Einfügen scheitert
- ▶ Tabellendefinitionen von *prüfen* und *hören* definiert references-Constraints  $\Rightarrow$  Löschen der Tabelle scheitert

# Lösungsvorschlag: FDs

- ▶ Folgende könnten z.B. gelten:
  - ▶  $\{A\} \rightarrow \{B\}$   
(2 Tupel mit gleichem  $A$  Wert, beide haben gleichen  $B$  Wert)
  - ▶  $\{A\} \rightarrow \{C\}$
  - ▶  $\{C, D\} \rightarrow \{C\}$
- ▶ Keine FD muss gelten (könnte “Zufall” sein)