

Einzelprüfung „Rechnerarchitektur / Datenbanken / Betriebssysteme (vertieft)“

Einzelprüfungsnummer 66113 / 2003 / Frühjahr

## Thema 2 / Aufgabe 3

(Universitätsverwaltung)

Stichwörter: SQL

Gegeben sei folgendes relationales Schema, das eine Universitätsverwaltung modelliert:

Studenten {[MatrNr:integer, Name:string, Semester:integer]}  
Vorlesungen {[VorlNr:integer, Titel:string, SWS:integer, gelesenVon:integer]}  
Professoren {[PersNr:integer, Name:string, Rang:string, Raum:integer] }  
 hoeren {[MatrNr:integer, VorlNr:integer]}  
 voraussetzen {[VorgaengerVorlNr:integer, NachfolgerVorlNr:integer]}  
 pruefen {[MatrNr:integer, VorlNr:integer, PrueferPersNr:integer, Note:decimal]}

Formulieren Sie die folgenden Anfragen in SQL:

- (a) Alle Studenten, die den Professor *Kant* aus einer Vorlesung kennen.

Lösungsvorschlag

```
SELECT DISTINCT s.Name
FROM Studenten s, Professoren p, hoeren h, Vorlesungen v
WHERE
  p.Name = 'Kant' AND
  v.gelesenVon = p.PersNr AND
  s.MatrNr = h.MatrNr AND
  h.VorlNr = v.VorlNr
```

- (b) Geben Sie eine Liste der Professoren (Name, PersNr) mit ihrem Lehrdeputat (Summe der SWS der gelesenen Vorlesungen) aus. Ordnen Sie diese Liste so, dass sie absteigend nach Lehrdeputat sortiert ist! Bei gleicher Lehrtätigkeit dann noch aufsteigend nach dem Namen des Professors/der Professorin.

Lösungsvorschlag

```
SELECT p.Name, p.PersNr, SUM(v.SWS) AS Lehrdeputat
FROM Vorlesungen v, Professoren p
WHERE v.gelesenVon = p.PersNr
GROUP BY p.Name, p.PersNr
ORDER BY Lehrdeputat DESC, p.Name ASC;
```

- (c) Geben Sie eine Liste der Studenten (Name, MatrNr, Semester) aus, die mindestens zwei Vorlesungen bei *Kant* gehört haben.

Lösungsvorschlag

### Mit einer VIEW

```
CREATE VIEW hoertKant AS
SELECT s.Name, s.MatrNr, s.Semester, v.VorlNr
FROM Studenten s, hoeren h, Vorlesungen v, Professoren p
WHERE
```

```
s.MatrNr = h.MatrNr AND
h.VorlNr = v.VorlNr AND
v.gelesenVon = p.PersNr AND
p.Name = 'Kant';
```

```
SELECT DISTINCT h1.Name, h2.MatrNr, h1.Semester
FROM hoertKant h1, hoertKant h2
WHERE h1.MatrNr = h2.MatrNr AND h1.VorlNr <> h2.VorlNr;
```

oder:

```
SELECT DISTINCT Name, MatrNr, Semester
FROM hoertKant
GROUP BY Name, MatrNr, Semester
HAVING COUNT(VorlNr) > 1;
```

## In einer Abfrage

```
SELECT s.Name, s.MatrNr, s.Semester
FROM Studenten s, hoeren h, Vorlesungen v, Professoren p
WHERE
  s.MatrNr = h.MatrNr AND
  h.VorlNr = v.VorlNr AND
  v.gelesenVon = p.PersNr AND
  p.Name = 'Kant'
GROUP BY s.MatrNr, s.Name, s.Semster
HAVING COUNT(s.MatrNr) > 1;
```

(d) Geben Sie eine Liste der Semesterbesten (MatrNr und Notendurchschnitt) aus.

Lösungsvorschlag

```
CREATE VIEW Notenschnitte AS (
  SELECT p.MatrNr, s.Name, s.Semester, AVG(Note) AS Durchschnitt
  FROM Studenten s, pruefen p
  WHERE s.MatrNr = p.MatrNr
  GROUP BY p.MatrNr, s.Name, s.Semester
);

SELECT a.Durchschnitt, a.MatrNr, a.Semester
FROM Notenschnitte a, Notenschnitte b
WHERE
  a.Durchschnitt >= b.Durchschnitt
  a.Semster = b.Semster
GROUP BY a.Durchschnitt, a.MatrNr, a.Semester
HAVING COUNT(*) < 2;
```



## Die Bschlangaul-Sammlung

### Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an [hermine.bschlangaul@gmx.net](mailto:hermine.bschlangaul@gmx.net). Der TeX-Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: <https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Staatsexamen/66113/2003/03/Thema-2/Aufgabe-3.tex>