

## Aufgabe 1: SQL und Sichten

(1 P.)

a) Gegeben folgendes Schema:

Universität: U(UID, Name, Stadt, Num\_Studenten)

Professoren: P(PID, UID, Vorname, Name, Jahre\_Angestellt)

- Erstellen Sie eine Sicht `RennomierteUni(Name, Jahre_Angestellt, Num_Professoren)`, die die Namen aller Universitäten mit mehr als 50 Professoren enthält sowie die Anzahl der dort angestellten Professoren aufgeschlüsselt nach Arbeitsjahren.
- Schreiben Sie basierend auf dieser Sicht eine Anfrage, die die fünf Universitäten mit größter Anzahl an angestellten Professoren zurück gibt. Dabei sollen nur Professoren zählen, die länger als zehn Jahre angestellt sind. Hinweis: Verwenden Sie die `LIMIT`-Klausel, um dies Anzahl der ausgegebenen Tupel zu beschränken.
- Schreiben Sie die selbe Anfrage ohne die Sicht zu benutzen.
- Betrachten Sie das folgende Insert-Statement:  

```
INSERT INTO RennomierteUni VALUES('Uni K', 10, 20)
```

Ist dieses Statement valide? Begründen Sie Ihre Antwort.

b) Geben Sie für die folgenden Ausdrücke jeweils an, was berechnet wird. Beschreiben Sie wo der Unterschied (auch bzgl. Ergebnis) liegt, falls es einen Unterschied gibt.

i) Die Tabelle `T` enthält genau 1 Tupel mit einer Spalte `x`, die einen Nullwert enthält.

- `SELECT COUNT(*) FROM T`
- `SELECT COUNT(x) FROM T`

ii) - `SELECT p.persnr, (SELECT COUNT(*) FROM vorlesungen v  
WHERE p.persnr=v.gelesenon) AS anzahl FROM professoren p`

- `SELECT p.persnr, COUNT(*) AS anzahl  
FROM professoren p JOIN vorlesungen v ON p.persnr=v.gelesenon  
GROUP BY p.persnr`

## Aufgabe 2: Rekursion in SQL

(1 P.)

Gegeben folgende Tabellen, die Filme und deren “ist-sequel-von” Beziehung angeben, mit **exemplarischem Inhalt**. Die Attribute *vorgaenger* bzw. *nachfolger* aus der Tabelle *Sequels* sind jeweils Fremdschlüssel auf das Attribut *id* der Tabelle *Filme*.

Filme	
id	titel
1	Jaeger des verlorenen Schatzes
2	Indiana Jones und der Tempel des Todes
3	Lethal Weapon - Zwei stahlharte Profis
4	Stirb Langsam
5	Indiana Jones und der letzte Kreuzzug
6	Lethal Weapon 2 - Brennpunkt L.A.
7	Stirb Langsam 2
8	Lethal Weapon 3 - Brennpunkt L.A. - Die Profis sind zurück
9	Stirb Langsam: Jetzt erst recht
10	Lethal Weapon 4 - Zwei Profis räumen auf
11	Stirb Langsam 4.0
12	Indiana Jones und das Königreich des Kristallschädels
13	Stirb Langsam - Ein guter Tag zum Sterben

Sequels	
vorgaenger	nachfolger
1	2
2	5
3	6
4	7
5	12
6	8
7	9
8	10
9	11
11	13

- Geben Sie eine rekursive SQL-Anfrage an, die zu jedem Film, der der erste der Reihe ist, den letzten (aktuellsten) Film der Reihe berechnet. Beispiel für die oben angegebenen Daten: “Lethal Weapon - Zwei stahlharte Profis” und “Lethal Weapon 4 - Zwei Profis räumen auf” als erster Film bzw. letzter Film der Reihe.

## Aufgabe 3: Fensteranfragen

(1 P.)

- a) Gegeben eine wie folgt definierte Tabelle mit angegeben Inhalt:

```
CREATE TABLE wq(a INT, b INT, c INT);
```

wq		
a	b	c
1	12	1
2	16	2
3	11	1
4	14	2
5	15	1
6	13	1

Ordnen Sie die folgenden Anfragen den untenstehenden Ergebnissen zu. Jede Anfrage auf der Tabelle **wq** erzeugt genau eines dieser Ergebnisse.

Q1: SELECT a, RANK() OVER(ORDER BY b) r FROM wq ORDER BY a;

Q2: SELECT a, SUM(b) OVER(ORDER BY b ROWS 2 PRECEDING) r FROM wq ORDER BY a;

Q3: SELECT a, COUNT(\*) OVER(PARTITION BY c) r FROM wq ORDER BY a;

Q4: SELECT a, SUM(b) OVER(PARTITION BY c ORDER BY b ROWS 2 PRECEDING) r  
FROM wq ORDER BY a;

R1:

a	r
1	4
2	2
3	4
4	2
5	4
6	4

R2:

a	r
1	2
2	6
3	1
4	4
5	5
6	3

R3:

a	r
1	23
2	45
3	11
4	39
5	42
6	36

R4:

a	r
1	23
2	30
3	11
4	14
5	40
6	36

- b) Geben Sie für die Anfragen Q1 und Q3 aus Teil a) je eine äquivalente SQL Anfrage an, die ohne Fensteranfragen-Features auskommt.
- c) Betrachten Sie eine Tabelle einer Datenbank, in der Regionen (Länder, Gemeinden, Ortsteile, ...) verwaltet werden:

REGION: (Region, Einwohner, liegt\_in\_Region)

Erstellen Sie eine SQL Anfrage, die alle Spalten von REGION enthält und zudem eine Spalte "Rang", die den Rang bzgl. der Einwohnerzahl innerhalb der direkten Überregion beinhaltet. Z.B. hat Kaiserslautern den Rang 5, weil es die fünftgrößte Region in Rheinland-Pfalz ist und Rheinland-Pfalz die direkte Überregion von Kaiserslautern ist.