



**Übungen zur Vorlesung
"Datenbanksysteme I"**
Sommersemester 2009

Manuel Mayr (manuel.mayr@uni-tuebingen.de)

9. Übungsblatt

Ausgabe: 09. Juli 2009 · Besprechung: 16. Juli 2008

Aufgabe 1: SQL-Level: Advanced

(9 Punkte)

1. Gegeben sei die folgende Relation:

Personen		
Id	Name	Alter
:	:	:

Geben Sie eine SQL-Anfrage an, die die Person(en) bestimmt, für die es 99 Personen gibt, die älter sind (es kann mehrere solche Personen geben, es muss keine solche Person(en) geben). Lassen Sie **Id** und **Name** ausgeben.

2. Gegeben sei die folgende Relation, welche Stichproben einer Wetterstation speichert:

Wetter	
Tag	Temperatur
1	10.3
2	13.2
3	4.5
:	:
101	0.4
102	1.8
:	:

Berechnen Sie mittels einer SQL-Anfrage den Median (bzw. Zentralwert) für die Spalte **Temperatur**. Zur Erinnerung: Ein Wert m ist der Median einer Stichprobe, wenn höchstens die Hälfte der Beobachtungen in der Stichprobe einen Wert $< m$ und höchstens die Hälfte einen Wert $> m$ hat.

3. Gegeben seien die Relationen:

\mathcal{R}
A
10
13
4
⋮
0
1
⋮

\mathcal{S}
B
13
⋮
6
0
1
⋮

Checken Sie mit Hilfe einer SQL-Anfrage ob die Spalte $\mathcal{R}.A$ eine Teilmenge der Spalte $\mathcal{S}.B$ ist ($\mathcal{R}.A \subseteq \mathcal{S}.B$). Duplikate spielen keine Rolle!

Aufgabe 2: Verarbeitung von Listen

(18 Punkte)

Gegeben Sei eine *reellwertige* Liste $L = (n_L, f_L)$ mit $n_L \in \mathbb{N}$ und $f_L : \{1, 2, \dots, n_L\} \rightarrow \mathbb{R}$. Für eine spezifische Liste L bezeichnet $L[i]$ den Wert des i -ten Elementes, also $f_L(i) : |L| := n_L$ bezeichnet die Listenlänge.

Sei eine Tabelle zur Speicherung von solchen Listen durch folgende CREATE TABLE-Anweisung gegeben:

```
CREATE TABLE lists
(
  id INTEGER NOT NULL
, pos INTEGER NOT NULL
, val DOUBLE NOT NULL
, CONSTRAINT pk_lists PRIMARY KEY (id, pos)
, CONSTRAINT chk_pos CHECK (pos >= 1)
);
```

Das Tupel (4711, 5, 13.6) repräsentiert also beispielsweise das fünfte Element der Liste mit dem Identifikator 4711, dessen Wert 13.6 ist. Sie können davon ausgehen, dass die Listenpositionen fortlaufend und ohne Lücken, beginnend bei 1 vergeben sind.

Geben Sie jeweils eine SQL-Anweisung zu den folgenden Fragestellungen an:

1. Wie ist der Wert des ersten Elements der Liste mit dem Identifikator 3?
2. Lassen Sie den kleinsten Wert aller Listenelemente ausgeben (ein Wert). Benennen Sie die Ergebnisspalte als `MinVal`.
3. Welches ist das größte aller Listenelemente (ein Wert)? Verwenden Sie nicht die Aggregatfunktion `MAX`.
4. Lassen Sie alle Listenidentifikatoren und die zugehörigen Summen der Listenelemente ausgeben, absteigend sortiert nach der Summe (d.h. beginnend mit der/den Liste(n) mit der größten Summe).
5. Lassen Sie den Listenidentifikator und den Durchschnitt der Werte der Elemente der jeweiligen Liste ausgeben für alle Listen, deren Listenidentifikator größer als 3 sind und die mindestens 4 Elemente haben.
6. Welche Liste(n) hat (haben) die größte Listenlänge?
7. Bei welchen Listen ist die Summe der (wertmäßig) größten drei Elemente kleiner als 10.0.
8. Welche Listen sind monoton steigend, d.h. $L[i+1] \geq L[i]$ für $i = 1, \dots, |L| - 1$?
9. Bei welchen Listen ist das erste Element positiv, und es existiert genau ein Vorzeichenwechsel?