

**Datenbanksysteme I**  
WS 2017/18

**Übungsblatt 5: SQL**

Abgabe bis 24.11.2017 um 12:00 Uhr mittags

Besprechung: 27. bis 30.11.2017

**Hinweis:** Bitte nutzen Sie zur Beantwortung der SQL-Übungsaufgaben *ausschließlich* die in der Vorlesung und in den Tutorien definierte Notation. Wir behalten uns vor andere Operationen und Schlüsselwörter als Fehler zu werten, selbst wenn ein spezifisches Datenbanksystem diese Operation unterstützt.

Die Aufgaben 5-1 und 5-2 beziehen sich auf das in Übungsblatt 4 genutzte LTP-Datenbankschema und die darin vorgestellte Ausprägung:

```
L (lnr, lname, status, sitz)  T (tnr, tname, farbe, gewicht)
P (pnr, pname, ort)          LTP (lnr, tnr, pnr, menge)
```

**Aufgabe 5-1**     *Anfragen in SQL*  
**Hausaufgabe**

(1+2+2+2 Punkte)

Formulieren Sie folgende Anfragen jeweils in der relationalen Algebra und in SQL. Achten Sie dabei darauf, dass die Anfragen äquivalent sind (Stichwort: Duplikate).

*Hinweis:* Bei einem Join in der relationalen Algebra können Namensgleiche Attribute durch das voranstellen des zugehörigen Namens der Relation eindeutig identifiziert werden. Beispiel: Siehe Aufgabe 5-3

- a) Bestimme die Namen aller Lieferanten mit Sitz in Berlin und einem Status von mindestens 15.
- b) Bestimme eine Liste aller Teilennamen und der Namen der Lieferanten die diese liefern. Benutzen Sie dabei für die SQL Anfrage in der FROM-Klausel ausschließlich den Natural Join.
- c) Finde die Namen aller Lieferanten, die Teile mit einem Einzelgewicht von mehr als 16 liefern. Benutzen Sie dabei für die SQL Anfrage in der FROM-Klausel ausschließlich das JOIN-ON-Konstrukt.
- d) Bestimme die Orte der Projekte, bei denen Lieferant Schmidt beteiligt ist. Benutzen Sie dabei für die SQL Anfrage in der FROM-Klausel ausschließlich das JOIN-USING-Konstrukt.
- e) **[Keine Bonuspunkte]** Bestimme Namen und Farben aller Teile, mit denen sowohl Projekt P3 als auch Projekt P4 beliefert werden. Benutzen Sie dabei für die SQL Anfrage in der FROM-Klausel ausschließlich das Kreuzprodukt.

**Aufgabe 5-2**      *Änderungsoperationen in SQL*

(2+1+3 Punkte)

**Hausaufgabe**

Formulieren Sie die folgenden Änderungsoperationen in SQL. Gehen Sie davon aus, dass die Fremdschlüssel mit `on delete set null` definiert wurden. Geben Sie die für jede Teilaufgabe nötigen Änderungsbefehle in einer geeigneten Reihenfolge an, so dass keine hängenden Verweise (dangling references) entstehen.

- a) Das silberfarbige Teil mit der Bezeichnung Fluxkompensator und einem Gewicht von 140 kommt auf den Markt und soll in die Teileliste aufgenommen werden. Der Lieferant Krause beliefert die Projekte P1 und P5, jeweils mit genau einem dieser Teile.
- b) Die Mengen aller bei Projekt P4 von Lieferant mit Nummer L1 gelieferten Teile sollen verdoppelt werden.
- c) Der Lieferant Schulz wird vom Konkurrenten Schmidt aufgekauft. Der Lieferant Schulz soll daher aus der Datenbank gelöscht werden. Der neue Name des Lieferanten Schmidt ändert sich auf Schmidt-Schulz und der Status erhöht sich um 10. Alle Projekte die von Schulz beliefert werden sollten, werden nun von Schmidt-Schulz beliefert. (Es kann davon ausgegangen werden, dass Schmidt und Schulz nicht die gleichen Teile an die gleichen Projekte liefern.)

**Aufgabe 5-3**      *Join-Operationen in SQL*

(1+1+1+1+1 Punkte)

**Hausaufgabe**

Im folgenden markiert das Symbol  $\ltimes$  einen Left-Outer-Join und das Symbol  $\rtimes$  einen Right-Outer-Join. Gegeben seien die beiden Relationen **T1**(id, name) und **T2**(id, wert) mit den folgenden Ausprägungen:

T1	id	name	T2	id	wert
	1	a		1	x
	2	b		3	y
	3	c		5	z

Formulieren Sie folgende Anfragen in SQL und geben Sie die Ergebnisrelation an. Sie können Join-Operatoren anwenden, müssen das aber nicht tun.

- a)  $T1 \times T2$
- b)  $T1 \ltimes_{T1.id=T2.id} T2$
- c)  $T1 \rtimes T2$
- d)  $T1 \rtimes_{T1.id=T2.id} T2$
- e)  $T1 \ltimes_{T1.id=T2.id} T2$