

## DAB1 – Praktikum 12: Lösungen

Formulieren Sie SQL-Ausdrücke für folgende Abfragen:

- 1) Wie hoch ist der durchschnittliche Lagerbestand aller Biere (der Wert soll den Namen *Durchschnittslagerbestand* erhalten)?
- 2) Welche Besucher (Name,Vorname, Strasse) wohnen an einer Strasse, deren Bezeichnung das Wort *,bach'* enthält? Untersuchen Sie auch die Resultate, die Sie mit *,Bach'*, *,BACH'* etc. erhalten. Sehen Sie hier allenfalls ein Problem?
- 3) An welchen Strassen gibt es mindestens drei Restaurants (GROUP BY verwenden)?
- 4) Bilden Sie das Kreuzprodukt der Tabellen Restaurant und Besucher (mit und ohne CROSS JOIN). Wieviele Tupel enthält das Resultat? Warum? Wozu könnte eine solche Abfrage nützlich sein?
- 5) Gesucht ist eine Liste von Besuchern mit Name, Vorname und der Anzahl Restaurantbesuche pro Woche (= Frequenz). Falls ein Besucher nie Gast ist, soll er auf der Liste mit einer Anzahl Besuche von 0 erscheinen. Verwenden Sie dazu einen OUTER JOIN. Optional: Lösung ohne JOIN.
- 6) Gesucht ist eine Liste der Hersteller von Biersorten zusammen mit der Anzahl Biersorten, die sie produzieren und der Anzahl *verschiedener* dabei verwendeter Grundstoffe.
- 7) Welche Biersorten sind von allen mit derselben Note bewertet worden (Hinweis: das bedeutet, dass kleinste Note = durchschnittliche Note = grösste Note)?
- 8) Gesucht ist eine Liste von Restaurants mit ihrem Namen, ihrem Suppenpreis, sowie dem durchschnittlichen Suppenpreis aller Restaurants derselben Strasse.
- 9) Gesucht ist eine Liste von Restaurants und Biersorten, von denen sie am meisten an Lager haben.
- 10) Gesucht sind die Strassen, an denen es mehr Restaurants gibt als Besucher.
- 11) Gesucht ist eine Liste der Strassen von Besuchern, deren Vornamen den Buchstaben "p" enthält und die in Restaurants verkehren mit einem Suppenpreis zwischen 3 und 5
- 12) Gesucht ist die Anzahl verschiedener Besucher des Restaurant Löwen.

Lösungen (Formatierung und Aliasnamen können abweichen):

- 1) 

```
SELECT AVG(AnLager) AS Durchschnittslagerbestand
FROM Sortiment;
```
- 2) 

```
SELECT Name,Vorname,Strasse
FROM Besucher
WHERE Strasse LIKE '%bach%';
```

Die 'case-sensitivity' von LIKE ist nicht definiert. Das hängt vom RDBMS bzw. dessen Konfiguration ab. Man kann somit bei der Selektion nicht unterscheiden zwischen 'Bach' und 'bach'

- 3) 

```
SELECT Strasse
FROM Restaurant
GROUP BY Strasse
HAVING COUNT(*) >= 3;
```
- 4) 

```
SELECT *FROM Restaurant CROSS JOIN Besucher;
```

bzw.

```
SELECT * FROM Restaurant, Besucher;
```

Die Anzahl Tupel entspricht dem Produkt der Tupelanzahlen der beiden Operanden (per Definition).

Reine Kreuzprodukte (d.h. ohne Selektion/Projektion) sind z.B. dann nützlich, wenn es darum geht, auf einfache Art eine grössere Anzahl Tupel zu erzeugen (z.B. als Testdaten).

- 5) 

```
SELECT Name,Vorname,SUM(COALESCE(Frequenz,0)) AS AnzahlBesuche
FROM Besucher AS b
LEFT OUTER JOIN Gast AS g
ON b.Name = g.Bname AND b.Vorname = g.Bvorname
GROUP BY b.Name,b.Vorname
ORDER BY AnzahlBesuche DESC;
```

ohne OUTER JOIN:

```
SELECT Bname, Bvorname, SUM(Frequenz) AS AnzahlBesuche
FROM Gast
GROUP BY Bname, Bvorname
```

UNION

```
SELECT Name, Vorname, 0 AS AnzahlBesuche
FROM Besucher AS x
WHERE NOT EXISTS (
    SELECT 1
    FROM Gast AS y
    WHERE y.Bname = x.Name AND y.Bvorname = x.Vorname)
ORDER BY AnzahlBesuche DESC;
```

- 6) 

```
SELECT Hersteller, COUNT(*) AS AnzahlBiere,
COUNT(DISTINCT Grundstoff) AS AnzahlGrundstoffe
FROM Biersorte
GROUP BY Hersteller;
```
- 7) 

```
SELECT Bsorte
FROM Lieblingsbier
GROUP BY Bsorte
HAVING MIN(Bewertung) = MAX(Bewertung);
```

- 8) 

```
SELECT x.Name, x.Suppenpreis, y.Durchschnittspreis
FROM Restaurant x
JOIN (
    SELECT Strasse, AVG(Suppenpreis) AS Durchschnittspreis
    FROM Restaurant
    GROUP BY Strasse) AS y
ON x.Strasse = y.Strasse;
```
- 9) 

```
SELECT Rname, Bsorte
FROM Sortiment x
WHERE AnLager = (
    SELECT MAX(AnLager)
    FROM Sortiment y
    WHERE y.Rname = x.Rname
    GROUP BY Rname);
```
- 10) 

```
SELECT Strasse
FROM Restaurant x
GROUP BY Strasse
HAVING COUNT(*) > (
    SELECT COUNT(*)
    FROM Besucher y
    WHERE y.Strasse = x.Strasse
    GROUP BY Strasse);
```
- 11) 

```
SELECT DISTINCT x.Strasse
FROM Besucher x, Gast y, Restaurant z
WHERE x.Vorname = y.Bvorname AND x.Name = y.Bname AND
y.Rname = z.Name AND x.Vorname LIKE '%p%' AND
z.Suppenpreis BETWEEN 3 AND 5;
```
- 12) 

```
SELECT COUNT(*) AS Besucher_Restaurant_Löwen
FROM (SELECT DISTINCT Bname, Bvorname
    FROM Gast
    WHERE Rname = 'Löwen') AS x;
```