

Gruppierungen: Lösungen zu den Aufgaben

(1)

| Funktion | Bedeutung | Wert in Gruppe APreis=9.99 |
|----------|---|----------------------------|
| AVG | Durchschnittlicher Bestand in der den Preiskategorien | AVG(ABestand)=87 |
| COUNT | | COUNT(ABestand)= |
| MAX | | MAX(ABestand)= |
| MIN | | MIN(ABestand)= |
| SUM | | SUM(ABestand)= |

(2)

(i)

```
SELECT APreis, SUM(ABestand) AS Bestandssumme
FROM `artikel`
WHERE APreis <= 10
GROUP BY APreis
ORDER BY APreis, Bestandssumme
```

(ii) Gruppiert wird nach APreis und nach ABestand, in der ersten Spalte kann man sich zusätzlich mit COUNT(*) noch ausgeben lassen, wieviele Datensätze in dieser Gruppe zusammengefasst werden.

```
SELECT COUNT(*), APreis, ABestand
FROM artikel
GROUP BY APreis, ABestand
```

(iii) Die Abfrage ermittelt die Zahl der Datensätze, die einen identischen Bestand aufweisen. Z.B. "80 - 3", es gibt 3 Datensätze, deren Bestand mit 80 Exemplaren vermerkt ist. Siehe auch die Bedeutung von COUNT(*) bei der Lösung zu Aufgabe 2.

(v) Liegt die Filterbedingung, wie in Aufgabe (2) als Tabellenattribut vor, kann man sowohl vor der Gruppierung mit WHERE als auch nach der Gruppierung mit HAVING filtern:

```
SELECT APreis, SUM(ABestand) AS Bestandssumme
FROM `artikel`
WHERE APreis <= 10
GROUP BY APreis
```

```
SELECT APreis, SUM( ABestand ) AS Bestandssumme
FROM `artikel`
GROUP BY APreis
HAVING APreis <=10
LIMIT 0 , 30
```

Allgemein: Entsteht das Attribut der Filterung erst durch die Gruppierung selbst, kann erst nach der Gruppierung mit HAVING gefiltert werden:

```
SELECT ABestand, AVG(APreis) AS Mittelpreis
FROM artikel
GROUP BY ABestand
WHERE Mittelpreis < 10
```

```
SELECT ABestand, AVG(APreis) AS Mittelpreis
FROM artikel
GROUP BY ABestand
HAVING Mittelpreis < 10
```

Die Aufgabe (v) lässt sich also nicht mit WHERE lösen, da man dabei immer nur für eine Zeile das Produkt aus Bestand und Preis betrachten kann - man möchte aber für eine ganze Kategorie die Bestandssumme * Artikelpreis anschauen und danach auch Filtern:

```
SELECT APreis, SUM(`ABestand`)*APreis AS Umsatz
FROM artikel
GROUP BY APreis HAVING Umsatz >= 3000
```

From:
<https://www.info-bw.de/> -

Permanent link:
https://www.info-bw.de/faecher:informatik:oberstufe:datenbanken:sql_gruppierungen:lsg:start

Last update: **10.11.2020 09:32**

