

SQL - Gruppierungen

Datensätze, die in einer Spalte oder mehreren Spalten die gleichen Werte aufweisen, können mithilfe des GROUP BY-Zusatzes in Gruppen zusammengefasst werden. Gruppierungen machen vor allem dann Sinn, wenn innerhalb der Gruppen mit den Spaltenwerten gerechnet wird, z.B. wenn die Gesamtzahl aller Artikel einer bestimmten Preiskategorie ermittelt werden soll.

Ohne Gruppierung	Mit Gruppierung
SELECT * FROM artikel ORDER BY APreis ASC	SELECT * FROM artikel GROUP by APreis ORDER BY APreis ASC
	
	SELECT APreis,SUM(ABestand) FROM artikel GROUP BY APreis ORDER BY APreis ASC
	

Alle Artikel mit dem Preis von 1,00 Euro werden durch die Gruppierung auf einen Datensatz „projiziert“. Im Falle der zweiten Gruppierungsklausel wird zusätzlich der Gesamtbestand aller Artikel einer Preiskategorie mithilfe der Aggregationsfunktion SUM(.) in der Spalte ausgegeben.



Genau genommen ist die Verwendung der Wildcard * zur Auswahl aller Spalten in Kombination mit GROUP BY nur bedingt sinnvoll, da diejenigen Spalten, nach denen nicht gruppiert wird, unterschiedliche Werte aufweisen können. Das Beispiel in Tabelle 1 zeigt, dass in einem solchen Fall die Werte eines (beliebigen) Datensatzes der Gruppe in diesen Spalten angezeigt wird. Bei restriktiverer Einstellung des Datenbanksystems müsste die Auswahl von Spalten, nach denen nicht gruppiert wird, eine Fehlermeldung liefern.

Man kann auch gleichzeitig nach mehreren Merkmalen gruppieren, indem man einen Ausdruck der Form

```
SELECT * FROM verkaeufe GROUP BY jahr, land, produkt;
```

verwendet. Hier werden die Verkäufe nach Jahr, Land und Produkt gruppiert (vgl. Aufgabe (3)).

Aufgaben

Löse die folgenden Aufgaben im SQL-Abfragefenster von phpMyAdmin auf der Datenbank webshop und speichere deine Lösungen in einer Textdatei oder deinem Info-Heft.



(A1)

Neben der Summenfunktion gibt es weitere Berechnungsfunktionen für zahlenwertige Spalten. Vervollständige die folgende Tabelle unter Verwendung folgender SQL-Abfrage:

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

Dabei ist FUNKTION nacheinander durch AVG, COUNT, MAX, MIN und SUM zu ersetzen.

Funktion	Bedeutung	Wert in Gruppe APreis=9.99
AVG		AVG(ABestand)=
COUNT		COUNT(ABestand)=
MAX		MAX(ABestand)=
MIN		MIN(ABestand)=
SUM		SUM(ABestand)=



(A2)

(i) Gib den jeweiligen Gesamtbestand der Artikel in den verschiedenen Preiskategorien unter 10,00EUR an.

(ii) Gib Gruppen mit gleichem Bestand und Preis zurück.

(iii) Notiere ohne phpMyAdmin zu verwenden, was die folgende SQL-Abfrage ausgibt:

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

(iv) Informiere dich über die HAVING-Bedingung und löse damit Teilaufgabe (2) erneut. Erläutere den Unterschied zwischen WHERE und HAVING.

(v) Gib alle Preiskategorien aus, in denen der maximal erzielbare Umsatz über 3.000,00 Euro liegt. Ist die Aufgabe auch mittels WHERE anstelle von HAVING lösbar?

([Lösungen](#))

From:

<https://wiki.qg-moessingen.de/> - **QG Wiki**

Permanent link:

https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:datenbanken:sql_gruppierungen:start?rev=1603202554

Last update: **20.10.2020 16:02**

