

## Zählen und Rechnen mit Aggregatfunktionen

Für Datenbanknutzer sind oft nicht die eigentlichen Daten, sondern Zahlen, die man aus den Daten berechnen kann, interessant – zum Beispiel:

Wie viele Kunden haben ein bestimmtes Produkt gekauft?

Wie hoch war der durchschnittliche Gewinn pro Auftrag?

Zur Veranschaulichung nutzen wir eine Tabelle, die die Aufträge eines Betriebs auflistet:

### Aufträge

Kunde	Auftrag	Preis	Adresse	Stadt
Sternapotheke	Schubladenschrank	850,00	Severinstr. 23	Köln
Tanzstudio	Parkettboden	2.750,00	Am Ponttor 11	Aachen
Mainzer Hof	Theke	1.200,00	Mainzer Str. 4	Köln
Traumfarben	Regale	690,00	Merowinger Str. 19	Köln
RWTH Aachen	Holzvertäfelung	8.900,00	Templergraben 8	Aachen

Für Rechnungen kann man die sogenannten **Aggregatfunktionen** verwenden.

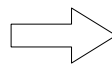
Der Begriff „Aggregat“ bedeutet hier, dass mehrere Zeilen einer Ergebnis-Tabelle zu einem Ergebnis zusammengefasst und ausgewertet werden – zum Beispiel durch Zählen:

*Beispiel: Wie viele Kunden kommen aus Köln?*

```
SELECT Kunde, Stadt
FROM Aufträge
WHERE Stadt = "Köln";
```

```
SELECT Count(Stadt)
FROM Aufträge
WHERE Stadt = "Köln";
```

Kunde	Stadt
Sternapotheke	Köln
Mainzer Hof	Köln
Traumfarben	Köln



Count
3

### Aggregatfunktionen in SQL

COUNT bestimmt die Anzahl der Zeilen des Ergebnis einer SELECT-Abfrage  
 MIN / MAX bestimmen das niedrigste / höchste Ergebnis  
 SUM berechnet die Summe der Ergebnisse  
 AVG berechnet das arithmetische Mittel der Ergebnisse

*Beispiel: Wie hoch ist der durchschnittliche Preis aller Aufträge?*

```
SELECT Avg(Preis) FROM Aufträge;
```

Avg
2878,00

## Ergebnisse gruppieren

Möchte man z.B. für jede Stadt zählen, wie viele Aufträge vor dort kamen, gibt es den Befehl GROUP BY, den man (nur) zusammen mit Aggregatfunktionen benutzen kann.

Mit GROUP BY werden die Ergebnisse, die in einer Spalte den gleichen Wert haben, zusammengefasst – statt alle Ergebnisse zusammenzufassen.

*Beispiel: Anzahl und Gesamtwert der Aufträge, gruppiert nach Städten*

```
SELECT Count (Stadt), Sum (Preis), Stadt
FROM Aufträge
GROUP BY Stadt;
```

Count	Sum	Stadt
3	2.740,00	Köln
2	11.650,00	Aachen

## Aliasse

Eine Schwierigkeit, die sich bei den Aggregatfunktionen ergibt ist, dass die Ergebnisspalte keine definierte Überschrift hat. In manchen DBMS wird eine Count-Spalte mit „Count“ beschriftet, im DBMS von w3schools.com beispielsweise mit dem kryptischen „Expr1000“.

Wenn man mit den Ergebnissen weiter rechnen möchte, oder sie für einen verschachtelten SQL-Befehl benötigt, braucht man jedoch die Beschriftung der Ergebnisspalte.

Für diesen Fall (und auch, um komplexe SQL-Befehle mit JOINS zu vereinfachen) bietet SQL die Möglichkeit, Tabellen und Spalten umzubenennen, mithilfe der sogenannten Aliasse:

Mit einem **Alias** definiert man innerhalb eines SQL-Befehls einen neuen Namen für eine Tabelle oder eine Spalte mit dem Schlüsselwort **AS**.

*Beispiel: Anzahl und Gesamtwert der Aufträge*

```
SELECT Count (Stadt) AS Anzahl,
       Sum (Preis) AS Gesamtwert
FROM Aufträge;
```

Anzahl	Gesamtwert
5	14.390,00

## Rechnen innerhalb einer Abfrage

Man kann innerhalb eines SQL-Befehls mit Ergebnissen, die aus Zahlen bestehen (also Mengen, Preise, oder die Ergebnisse von Aggregatfunktionen) auch direkt rechnen, und zwar mit den Grundrechenarten +, -, \*, /, % (plus, minus, mal, geteilt, modulo).

*Beispiel: Differenz zwischen höchstem und niedrigstem Preis*

```
SELECT Min (Preis) AS MinPreis,
       Max (Preis) AS MaxPreis,
       MaxPreis - MinPreis AS Differenz
FROM Aufträge;
```

MinPreis	MaxPreis	Differenz
690,00	8.900,00	8.210,00