

Die Aggregationsfunktionen

Welche SQL-Funktionen sind überhaupt Aggregationsfunktionen?

In SQL gibt es fünf wichtige Aggregationsfunktionen, natürlich gibt es noch weitere, allerdings werden diese nicht so häufig benutzt.

Die **wichtigsten** sind: MIN, MAX, AVG, SUM, COUNT.

Warum sind genau diese Aggregationsfunktionen?

Die Antwort erschließt sich aus der Definition für Aggregationsfunktionen: „Eine Aggregationsfunktion bildet aus **einer Vielzahl** von Eingabewerten **genau einen** Ergebniswert.“

Diese Funktionen liefern also genau ein Ergebnis, egal wie viele Werte man als Eingabe benutzt.

MIN und MAX

Wie der Name schon sagt, kann mit diesen Funktionen der kleinste (MIN) oder der größte (MAX) Wert einer Tabelle abgefragt werden. Das funktioniert sowohl für Zahlenwerte als auch für Zeichen.

Bei Zahlenwerten ist es eh logisch, welcher der größte/kleinste ist.

Zeichen werden nach der Unicode Tabelle sortiert. Alle Buchstaben werden des Weiteren in Großbuchstaben gewandelt, bevor eine Überprüfung stattfindet (d.h. es wird der Unicode des Großbuchstaben verwendet, auch wenn ein Kleinbuchstabe in der Tabelle eingetragen ist).

Für das Alphabet bedeutet das, dass ein "a" kleiner ist, als ein "b". Würde man ein "a" mit einem "A" vergleichen haben diese den selben Wert. Es wird sowohl für MIN als auch für MAX das in der Tabelle zuerst gespeicherte Zeichen ausgegeben. Ist das erste Zeichen gleich wird das zweite verglichen und so weiter, bis eindeutig ist, welche der beiden Zeichenketten kleiner/größer ist.

Aufbau einer SELECT Abfrage:

```
SELECT MAX(spaltenname) FROM tabellenname [WHERE ...];
```

```
SELECT MIN(spaltenname) FROM tabellenname [WHERE ...];
```

AVG und SUM

Wie man sich vielleicht schon denken kann, stehen diese beiden Abkürzungen für Durchschnitt (Average) und Summe (Sum).

Mit AVG kann man den durchschnittlichen Wert einer Spalte errechnen und mit SUM die Summe aller Werte der Spalte.

Prinzipiell kann man auch diese Funktionen sowohl für Datentypen, die eine Zahl und für Datentypen, die Zeichen speichern (z.B. varchar) benutzen. Allerdings muss man beachten, dass nur mit Zahlenwerten gerechnet werden kann.

SUM addiert, wie gewohnt, alle Zahlenwerte (egal in welchem Datentypen diese gespeichert sind) zusammen.

Bei AVG muss man allerdings etwas aufpassen. Es werden zwar nur Zahlenwerte addiert, aber alle vorhandenen Reihen (egal ob der Inhalt ein Zahlenwert, eine Zeichenkette ist), abgesehen von jenen mit NULL als Inhalt, zum Durchschnitt berechnen herangezogen. (Durchschnitt = alle Zahlenwerte summiert / alle Reihen der Spalte).

Aufbau einer SELECT Abfrage:

```
SELECT AVG(spaltenname) FROM tabellename [WHERE ...];  
  
SELECT SUM(spaltenname) FROM tabellename [WHERE ...];
```

COUNT

Mit der COUNT Funktion kann man ermitteln, wie viele Reihen eine Spalte hat.

Das Einzige, was man beachten muss ist, dass Reihen mit dem Wert NULL nicht mitgezählt werden.

Abgesehen davon sollte es keine großen Schwierigkeiten mit dieser Funktion geben.

Habe ich eine Spalte, mit den Einträgen „A“, „a“ und „AB“, dann sieht man, dass mit COUNT drei Einträge gezählt.

Aufbau einer SELECT Abfrage:

```
SELECT COUNT(spaltenname) FROM tabellename [WHERE ...];
```

COUNT DISTINCT

Möchte man allerdings wissen, wie viele Unterschiedliche Werte in einer Spalte vorkommen, muss man COUNT mit dem Schlüsselwort DISTINCT verbinden.

DISTINCT bewirkt grundlegend, dass Werte die mindestens zwei Mal vorkommen, nur einmal gezählt werden. Auch hier wird nicht in Groß- und Kleinschreibung unterschieden.

Habe ich eine Spalte, in der „A“, „a“ und „AB“ stehen, dann sieht man, dass mit COUNT DISTINCT nur zwei anstatt drei Einträgen gezählt.

Aufbau einer SELECT Abfrage:

```
SELECT COUNT(DISTINCT spaltenname) FROM tabellename [WHERE ...];
```

Aggregationsfunktionen mit WHERE

Natürlich kann man die Aggregationsfunktionen auch nach einem WHERE benutzen, allerdings kann man sie nicht einfach dranhängen, sondern muss ein eigenes Select schreiben, also ein Select, in einem Select:

```
SELECT *
FROM tabellenname
WHERE AVG(spaltenname)
-- Falsch

SELECT *
FROM tabellenname
WHERE (AVG/SUM/COUNT/MIN/MAX(spaltenname) FROM tabellenname);
-- Richtig
```

Rückgabotyp von Aggregationsfunktionen

Standardmäßig gibt es zwei Möglichkeiten. Entweder das Ergebnis wird als DECIMAL für Zahlenwerte oder als VARCHAR für Zeichen zurückgegeben.

Möchte man diesen in ein INT oder DOUBLE ändern, kann man das mit dem CAST Befehl machen. (Interessante Bemerkung, ist der Datentyp ein Float und ich will ein AVG, dann erhalte ich ganzzahlige Werte als Ergebnis, bei Double erhält man Ergebnisse mit Kommastellen. Caste ich bei AVG das Ergebnis in ein Double, erhalte ich auch Kommastellen)

Stellen wir uns eine Tabelle mit dem Namen „Preise“ vor. In dieser Tabelle gibt es die Spalte „Preis“ welche als Datentyp ein Double hat. Es gibt 4 Einträge: „3.5“, „5.3“, „1.7“ und „0.9“. Wir wollen die Summe dieser Werte erhalten, uns interessiert allerdings kein Wert mit Kommastellen, uns reicht ein ganzzahliges Ergebnis. Das Ergebnis muss also in ein INT gecastet werden. Die SQL Abfrage dazu sieht folgendermaßen aus:

```
SELECT SUM(CAST(Preis AS INT)) AS "Preis" FROM Preise;
```

Quellen

https://www.w3schools.com/sql/sql_min_max.asp

https://www.w3schools.com/sql/sql_count_avg_sum.asp

<https://dev.mysql.com/doc/refman/8.0/en/group-by-modifiers.html>

<https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html>

<https://www.dateneule.de/2019/04/29/aggregatfunktionen-in-sql/>

<http://dt.wara.de/pdf/sae/datenbanken/aggregatFunktionen/aggregatFunktionen.pdf>

