
Syntaxbeschreibungen anwenden

1BK2W

2020-04-20

Für Computerbefehle gibt es eine kompakte Beschreibung der Syntax (Rechtschreibung und Grammatik), die wie folgt aussieht:

```
SELECT [ALL|DISTINCT] {Spalten*} FROM Tabelle [, Tabelle, ...]
[WHERE Bedingung]
[GROUP BY Spalten [HAVING Bedingung]]
[ORDER BY Spalten] [ASC|DESC]
```

1 Erklärung

- **fett** gedruckt sind SQL-Schlüsselwörter. Sie müssen genau in der angegebenen Reihenfolge übernommen werden, Groß- und Kleinschreibung ist bei SQL-Schlüsselwörtern egal.
- *kursiv* gedruckt sind Platzhalter. Dafür setzen Sie Namen von Datenbankobjekten oder etwas selbst Formuliertes ein. *Spalten* steht für den Namen einer Spalte oder für mehrere Spalten, die Sie mit Komma getrennt aufzählen.
- [eckige Klammern] kennzeichnen optionale Teile, das heißt: Was in eckigen Klammern steht, müssen Sie nicht verwenden. Beachten Sie die Verschachtelung: **HAVING** dürfen Sie nur verwenden, wenn Sie auch **GROUP BY** verwendet haben.
- Ein senkrechter Strich | kennzeichnet Alternativen. Umliegende eckige oder geschweifte Klammern { } grenzen den Bereich der Alternativen ab. Sie zählen also entweder die Spalten auf oder Sie verwenden ein * für alle Spalten; Sie können das Schlüsselwort **ALL** oder das Schlüsselwort **DISTINCT** eintippen, aber nicht beide (Sie können auch keines von beiden eintippen, da die beiden in eckigen Klammern [] stehen) – das gleiche gilt für **ASC** und **DESC**.
- Klammern und Striche werden nicht eingetippt.

2 Beispiele

Sie haben folgende Tabelle, die Sie abfragen möchten:

Kunde			
<i>Kundennummer</i>	<i>Firmenname</i>	<i>PLZ</i>	<i>Ort</i>
1	Müller	75175	Pforzheim
2	Maier	75249	Kieselbronn
3	Schulze	76012	Karlsruhe
4	Fischer	75177	Pforzheim

1. Sie möchten alle Spalten und alle Zeilen sehen:
`SELECT * FROM Kunde`
 Der Ausdruck *FROM Kunde* macht klar, dass die Tabelle *Kunde* gelesen werden soll. Das Sternchen * steht für alle Spalten. Da Sie nicht `ORDER BY ...` sagen, ist die Reihenfolge der Zeilen unbestimmt.
2. Sie möchten nur die Orte sehen:
`SELECT Ort FROM Kunde`
 Der Name einer Spalte ersetzt das Sternchen.
3. Sie möchten nur Postleitzahlen und Orte sehen:
`SELECT PLZ, Ort FROM Kunde`
 Mehrere Spalten werden mit Komma getrennt aufgezählt.
4. Sie möchten die Orte sehen, aber jeden Ort nur einmal:
`SELECT DISTINCT Ort FROM Kunde`
DISTINCT entfernt Duplikate.
5. Sie möchten die Kundenliste nach Namen sortiert:
`SELECT * FROM Kunde ORDER BY Firmenname`
 (also von Fischer bis Schulze)
6. Sie möchten die Kundenliste nach Ort absteigend sortiert:
`SELECT * FROM Kunde ORDER BY Ort DESC`
 (zuerst zwei Kunden aus Pforzheim, dann der aus Kieselbronn, am Schluss der aus Karlsruhe; die Reihenfolge der Kunden aus Pforzheim ist unbestimmt.)
7. Sie möchten die Kundenliste nach Ort und dann (bei gleichem Ort) nach Namen sortiert:
`SELECT * FROM Kunde ORDER BY Ort, Firmenname`
 (also in Pforzheim zuerst Fischer, dann Müller)
8. Sie interessieren sich nur für Kunde 3:
`SELECT * FROM Kunde WHERE Kundennummer = 3`
9. Sie interessieren sich nur für Firma Fischer:
`SELECT * FROM Kunde WHERE Firmenname = 'Fischer'`
 Text muss immer in Hochkommas. Vorsicht: Innerhalb von Hochkommas wird Groß- und Kleinschreibung beachtet – 'Fischer' ≠ 'fischer' ≠ 'FISCHER'.
10. Sie interessieren sich für Fischer, egal ob groß- oder kleingeschrieben:
`SELECT * FROM Kunde WHERE UPPER(Firmenname) = 'FISCHER'`
 Die Zeilenfunktion *UPPER* wandelt in Großbuchstaben um.
11. Sie suchen Kunden, deren Name mit einem großen M anfängt:
`SELECT * FROM Kunde WHERE Firmenname LIKE 'M%'`
LIKE macht einen Mustervergleich, das Prozentzeichen % steht für beliebig viele Zeichen, der Unterstrich _ für ein einziges.
12. Sie wissen nicht, ob sich Maier mit ai, ei, ay oder ey schreibt:
`SELECT * FROM Kunde WHERE Firmenname LIKE 'M_er'`
13. Sie möchten wissen, wieviele Zeilen die Tabelle hat:
`SELECT COUNT(*) FROM Kunde`
COUNT ist eine Spaltenfunktion (Aggregatfunktion).
14. Sie möchten wissen, wieviele Kunden in jedem Ort sitzen:
`SELECT Ort, COUNT(*) FROM Kunde GROUP BY Ort`
15. Sie möchten wissen, wieviele Kunden in jedem Ort sitzen, aber nur für Orte mit mehr als einem Kunden:
`SELECT Ort, COUNT(*) FROM Kunde GROUP BY Ort HAVING COUNT(*) > 1`
 Im Unterschied zu *WHERE* wird die Bedingung nach *HAVING* nicht direkt auf die Tabelle angewendet, sondern auf das Ergebnis der Spaltenfunktion und Gruppierung.