

Datentypen in SQL

Quellen: <https://de.wikipedia.org/wiki/SQL#SQL-Datentypen>

<https://dev.mysql.com/doc/refman/5.7/en/data-type-overview.html>

int(n) <- fett dargestellt: sollte man kennen

Ganze Zahl (positiv oder negativ) mit (maximal) n Stellen

decimal(n,m)

Festkommazahl (positiv o. negativ) mit insgesamt maximal n Stellen, davon m Nachkommastellen.
Wegen der hier erfolgenden Speicherung als Dezimalzahl ist eine besonders für Geldbeträge notwendige Genauigkeit gegeben.

float(m)

Gleitkommazahl (positiv o. negativ) mit maximal m Nachkommastellen.

double(n,m)

Gleitkommazahl (positiv o. negativ) mit insgesamt n Stellen, davon m Nachkommastellen.

float und **double**

sind für technisch-wissenschaftliche Werte geeignet und umfassen auch die Exponentialdarstellung.
Wegen der Speicherung im Binärformat sind sie aber für Geldbeträge nicht geeignet, weil sich beispielsweise der Wert 0,10 € (entspricht 10 Cent) nicht exakt abbilden lässt.

char(n)

Zeichenkette (also Text) mit (genau) n Zeichen.

varchar(n)

Zeichenkette (also Text) von variabler Länge, aber maximal n druckbaren und/oder nicht druckbaren Zeichen.

text

Zeichenkette (zumindest theoretisch) beliebiger Länge. In manchen Systemen synonym zu clob.

date

Datum (ohne Zeitangabe)

time

Zeitangabe (evtl. inklusive Zeitzone)

timestamp

Zeitstempel (umfasst Datum und Uhrzeit; evtl. inklusive Zeitzone), meistens mit Millisekundenauflösung, teilweise auch mikrosekundengenau

boolean

Boolesche Variable (kann die Werte true(wahr) oder false (falsch) annehmen). Dieser Datentyp ist laut SQL:2003 optional und nicht alle DBMS stellen diesen Datentyp bereit.

blob(n)

Binärdaten von maximal n Bytes Länge.

clob(n)

Zeichenketten mit maximal n Zeichen Länge.

Wenn es die Tabellendefinition erlaubt, können Attribute auch den Wert NULL annehmen, wenn kein Wert bekannt ist oder aus anderen Gründen kein Wert gespeichert werden soll. Der NULL-Wert ist von allen anderen möglichen Werten des Datentyps verschieden.

SQL-Abfragen mit Berechnungen und Gruppierungen

Online-SQL-Abfragen: www.9d.dbg-db.de

Beispiel einer SQL-Abfrage mit einer Berechnung

```
SELECT Name, ROUND(Einwohner*1000000/Flaeche, 0) AS  
"Bevoelkerungsdichte in Einw. pro km^2"
```

```
FROM Land
```

```
WHERE Kontinent = "Europa" OR Kontinent = "Europe"
```

```
ORDER BY Einwohner/Flaeche DESC
```

Was wird vermutlich „ROUND(..., 0)“ bewirken?

Warum tritt die Zahl 1000000 auf?

Was wird die Tabelle vermutlich ausgeben?

Überprüfe einige Zahlenwerte der Ergebnis-Tabelle! (→ Wikipedia)

Beispiele von SQL-Abfragen mit Gruppierungen

1.

```
SELECT Kontinent, MIN(Einwohner), MAX(Einwohner)
```

```
FROM Land
```

```
GROUP BY Kontinent
```

Was wird vermutlich „MIN(...)“ und „MAX(...)“ bewirken?

Was gibt die Ergebnis-Tabelle an? Überprüfe das für Australien!

2.

```
SELECT Kontinent, AVG(Einwohner)
```

```
FROM Land
```

```
GROUP BY Kontinent
```

Was wird vermutlich „AVG(...)“ bewirken?

Was gibt die Ergebnis-Tabelle an? Überprüfe das für Australien! (TR!)

Beispiel einer SQL-Abfrage mit der COUNT-Funktion

```
SELECT Land, COUNT(*) AS Anzahl
```

```
FROM Ort
```

```
GROUP BY Land
```

```
ORDER BY Anzahl DESC
```

Was wird vermutlich „COUNT()“ bewirken?*

Was gibt die Ergebnis-Tabelle aus? Überprüfe das für Dänemark!

Wie lautet die SQL-Abfrage, die als Ergebnis eine Tabelle ausgibt, aus der ersichtlich ist, wie viele Länder es auf jedem Kontinent gibt?

Was Du in der Stunde nicht schaffst, erledige bitte zu Hause!