

Formatierung von Währungsfeldern in Jet-SQL (MS Access):

Mit der Funktion FormatCurrency () können Währungsfelder in Access formatiert werden.

Die allgemeine Syntax lautet:

FormatCurrency (Feldname [, AnzStellenNachDezimal] [, FührendeNullAnzeigen] [, KlammernFürNegativeZahl] [, StellenGruppieren])

Die eckigen Klammern sind nicht zu schreiben, sie zeigen optionale Argumente:

Argument	Beschreibung
Ausdruck	Erforderlich. Der zu formatierende Ausdruck.
AnzStellenNachDezimal	Optional. Numerischer Wert, der angibt, wie viele Stellen rechts vom Dezimaltrennzeichen angezeigt werden. Der Standardwert (-1) bedeutet, dass die Landes-/Regionaleinstellungen des Computers verwendet werden.
FührendeNullAnzeigen	Optional. Eine von drei Konstanten (siehe Tabelle unten), die angibt, ob für Dezimalzahlen <1 (z. B. 0,45) eine führende Null angezeigt werden soll oder nicht.
KlammernFürNegativeZahlen	Optional. Eine von drei Konstanten (siehe Tabelle unten), die angibt, ob negative Werte in Klammern gesetzt werden oder nicht.
StellenGruppieren	Optional. Eine von drei Konstanten (siehe Tabelle unten), die angibt, ob Zahlen mit dem Gruppentrennzeichen gruppiert werden, das in den Landes-/Regionaleinstellungen des Computers angegeben ist.

Die Argumente FührendeNullAnzeigen, KlammernFürNegativeZahlen und StellenGruppieren können die folgenden Werte annehmen:

Konstante	Wert	Beschreibung
vbTrue	-1	True
vbFalse	0	False
vbUseDefault	-2	Die Landes-/Regionaleinstellungen des Computers werden verwendet.

Beispiel:

Die Buchungen eines Kontos sollen mit 4 Nachkommastellen und bei Werten zwischen 0 und 1 mit führender Null angezeigt werden. Negative Zahlen sollen in Klammern gesetzt und die Ziffern landestypisch gruppiert werden.

Lösung:

```
SELECT BuchungsNr, FormatCurrency (Buchung, 4, -1, -1, -2)
FROM Konto;
```

Buchungs Nr	Buchung
1	1'525,0000 €
2	0,5901 €
3	(16,4500) €

Zeichenkettenfunktionen

Die verschiedenen Datenbanksysteme stellen unterschiedliche Funktionen für die Bearbeitung und Umwandlung von Zeichenketten zur Verfügung. Die Auswirkungen der entsprechenden Funktionen sind jedoch sehr ähnlich. So liefert im Datenbanksystem Oracle die Funktion ASCII(n) die Codenummer eines ASCII-Zeichens, während in Access die Funktion ASC(n) lautet und das Gleiche bewirkt.

Funktion	Wirkung, Beispiele
ASC(n)	Liefert die ASCII-Codenummer von n, ASC('M') → 77.
LCASE(string)	Wandelt um in Kleinbuchstaben, LCASE('Müller') → 'müller'
LEFT(string, länge)	Gibt linksbündige Teilzeichenkette aus, LEFT('Müller',2) → 'Mü'.
LEN(string)	Gibt die Zeichenanzahl aus, LEN('Müller') → 6.
RIGHT(string, länge)	Gibt rechtsbündige Teilzeichenkette aus, siehe LEFT().
UCASE(string)	Wandelt in Großbuchstaben um, UCASE('text') → 'TEXT'
&	Operator zur Verkettung (in Access) 'Hans' & ' ' & 'Müller' → 'Hans Müller'

Beispiel:

Unabhängig von der gespeicherten Schreibweise werden von der folgenden Anweisung die Nachnamen der Kunden so ausgegeben, dass der erste Buchstabe als Großbuchstabe erscheint, die folgenden Buchstaben jedoch in Kleinschrift.

```
SELECT Left (UCase (Nachname) , 1) &
LCASE (Right (Nachname, Len (Nachname) -1) ) AS [ 'Nachname' ]
FROM Kunden;
```

Die versehentliche Eingabe des Nachnamens „mÜller“ wird durch diese Anweisung umgewandelt in „Müller“.

Umwandlungsfunktionen

Mit Umwandlungsfunktionen werden z. B. Zeichenketten in Zahlen gewandelt und umgekehrt.

Beispiele sind:

Funktion	Wirkung, Beispiele
CDATE('datumsstring')	Zeichenkette, Zahl in Datumstyp DATE CDATE('12.03.2022') → 12.03.2022
STR(zahl)	Zahl in Zeichenkette, STR(89077) → '89077'
VAL(string)	Zeichenkette in Zahl, VAL('34') → 34
CINT(zahl)	Wandelt um in INTEGER, rundet auf CINT(45,8) → 46 CINT(-67,4) → -67

Die Funktion CDATE('datum') wandelt z. B. eine aus Zahlen bestehende Zeichenkette oder eine Ganzzahl in ein Datum um, soweit es vom ursprünglichen Typ her möglich ist.

Die Funktion CINT() rundet auf den nächsthöheren Ganzzahlwert auf und gibt dies als Ergebnis zurück.