

Arraykonstruktoren

Arraykonstruktoren Ein Arraykonstruktor baut einen Arraywert aus Einzelwerten. Er besteht aus dem Schlüsselwort ARRAY gefolgt von einer Liste von Ausdrücken in eckigen Klammern, zum Beispiel: ARRAY[1, 2, 3=4] Der Datentyp des Arrays wird aus den Datentypen der Elemente bestimmt (diese müssen kompatibel sein). Die Indizes eines Arrays beginnen bei 1.

Mehrdimensionale Arrays können durch geschachtelte Arraykonstruktoren gebildet werden. Das Schlüsselwort ARRAY kann bei den inneren Konstruktoren weggelassen werden. Beispiele: ARRAY[ARRAY[1, 2], ARRAY[3, 4]] ARRAY[[1, 2], [3, 4]] Mehrdimensionale Arrays müssen "rechteckig" sein; das heisst, die Grösse der Subarrays muss gleich sein. Schliesslich kann man ein Array auch einem Anfrageergebnis bilden. Die Anfrage muss genau eine Spalte ergeben. Beispiel: SELECT ARRAY(SELECT oid FROM pg_proc WHERE proname LIKE 'bytea%');
Zeilenkonstruktoren Ein Zeilenkonstruktor baut einen Zeilenwert aus Einzelwerten auf. Zum Beispiel: ROW(1, 2.5, 'Test') (Das Wort ROW kann weggelassen werden, wenn die Liste mindestens zwei Elemente hat.) Derartig aufgebaute Zeilen haben anfänglich einen anonymen Zeilendatentyp. Sie können mittels Typumwandlung in den Zeilentyp einer Tabelle oder eines benutzerdefinierten zusammengesetzten Typs umgewandelt werden. Mit Zeilenkonstruktoren aufgebaute Werte können in Tabellenspalten mit entsprechenden zusammengesetzten Datentypen gespeichert werden oder als Argumente an benutzerdefinierte Funktionen, die entsprechende zusammengesetzte Datentypen verarbeiten können, übergeben werden. Man kann sie auch mit anderen Zeilen vergleichen (=) oder auf NULL testen. Ausserdem können sie im Zusammenhang mit Unteranfragen verwendet werden. Auswertung von Ausdrücken Bei der Auswertung von komplexen Ausdrücken gibt es keine feste Reihenfolge. Hier ist zum Beispiel eine unsichere Methode, eine Division durch null in einer WHERE-Klausel zu vermeiden: SELECT . . . WHERE x <> 0 AND y/x > 1.5; Wenn es erforderlich ist, die Reihenfolge der Auswertung von Ausdrücken festzulegen, dann kann die CASE-Konstruktion verwendet werden: SELECT . . . WHERE CASE WHEN x <> 0 THEN y/x > 1.5 ELSE false END ; Eine CASE-Konstruktion, die auf diese Art verwendet wird, verhindert Optimierungen und sollte daher nur wenn unbedingt nötig angewendet werden. (In diesen Beispiel wäre er zweifellos am besten y > 1.5*x zu schreiben und dem Problem damit ganz aus dem Weg zu gehen.) Ferner gibt es keine Garantie, dass alle Teilausdrücke überhaupt ausgewertet werden. Im Beispiel SELECT funktion() OR true; würde funktion() (wahrscheinlich) gar nicht aufgerufen werden. Folgende sollte man sich bei komplexen Ausdrücken nicht auf eventuelle Nebenwirkungen von Funktionen verlassen.