

Skalar-Ausdrücke

$$\left\{ \begin{array}{ccc} & \left[\begin{array}{c} + \\ - \end{array} \right] & \left\{ \begin{array}{c} factor \\ (scalar-expression) \end{array} \right\} \\ scalar-expression & scalar-operator & scalar-expression \end{array} \right\}$$

Dieses Kapitel behandelt folgende Themen:

- scalar-expression
- scalar-operator
- factor

scalar-expression

Eine *scalar-expression* besteht aus einem *factor* und anderen *scalar-expressions* einschließlich *scalar-operators*.

In punkto Referenzierungspriorität gilt folgendes:

- Wenn in einer *scalar-expression* ein unqualifizierter Variablenname angegeben wird, wird zunächst angenommen, dass es sich um den Namen einer Spalte der referenzierten Tabelle handelt.
- Falls in der Tabelle eine Spalte dieses Namens nicht vorkommt, behandelt Natural die Variable als Benutzervariable (*host-variable*).

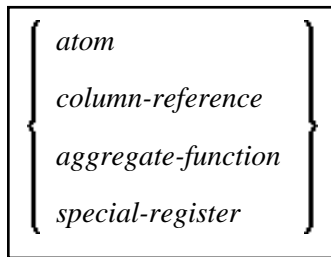
scalar-operator

$$\left\{ \begin{array}{c} + \\ - \\ * \\ / \\ | | \\ \text{CONCAT} \end{array} \right\}$$

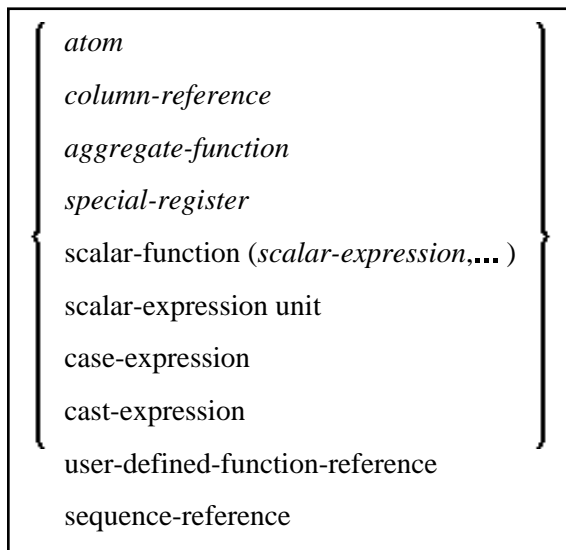
Ein *scalar-operator* kann einer der oben aufgeführten Operatoren sein, wobei vor und nach den Operatoren – und / jeweils mindestens ein Leerzeichen stehen muss.

factor

Common Set-Syntax:

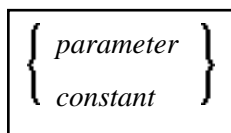


Extended Set-Syntax:



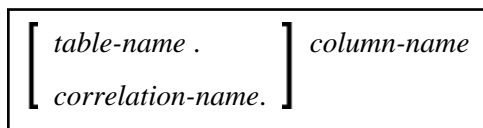
Ein *factor* kann eines der obigen Elemente sein, die im Folgenden beschrieben sind.

atom



Ein *atom* kann entweder ein *parameter* oder eine Konstante (*constant*) sein; vgl. Abschnitt *Grundlegende Syntaxbestandteile*.

column-reference



Eine *column-reference* ist ein Spaltenname (*column-name*), optional qualifiziert durch einen Tabellennamen (*table-name*) oder einen *correlation-name* (vgl. Abschnitt *Grundlegende Syntaxbestandteile*). Qualifizierte Namen sind oft klarer als unqualifizierte und manchmal erforderlich.

Anmerkung:

Ein *table-name* darf hier nicht explizit mit einem *authorization-identifier* qualifiziert werden. Falls Sie einen qualifizierten *table-name* benötigen, verwenden Sie stattdessen einen *correlation-name*.

Wird eine Spalte mit einem *table-name* oder *correlation-name* referenziert, muss sie in der betreffenden Tabelle enthalten sein. Wird weder *table-name* noch *correlation-name* angegeben, muss die betreffende Spalte in einer der in der FROM-Klausel angegebenen Tabellen enthalten sein (siehe *Table Expression*).

aggregate-function

Common Set-Syntax:

COUNT	{ (*) }	
	{ (DISTINCT <i>column-reference</i>) }	
AVG	{ (DISTINCT <i>column-reference</i>) }	
MAX		
MIN		{ ([ALL] <i>scalar-expression</i>) }
SUM		

Extended Set-Syntax:

COUNT	{ (*) (DISTINCT <i>column-reference</i>) }
AVG	{ (DISTINCT <i>column-reference</i>) ([ALL] <i>scalar-expression</i>) }
MAX	
MIN	
SUM	
COUNT_BIG	
STDDEV	
STDDEV_POP	
STDDEV_SAMP	
VAR	
VAR_POP	
VAR_SAMP	
VARIANCE	
VARIANCE_SAMP	

SQL bietet eine Reihe spezieller Funktionen zur Erweiterung der grundlegenden Such-Möglichkeiten. Folgende sogenannte SQL *aggregate-functions* sind verfügbar und werden von Natural unterstützt:

AVG	gibt den Durchschnitt der Werte einer Spalte zurück.
COUNT	gibt die Anzahl der Werte einer Spalte zurück.
MAX	gibt den größten Wert einer Spalte zurück.
MIN	gibt den kleinsten Wert einer Spalte zurück.
SUM	gibt die Summe der Werte einer Spalte zurück.

Bis auf COUNT (*) sammelt jede dieser Funktionen die Skalarwerte in einem Argument, d.h. einer einzelnen Spalte oder einer *scalar-expression*, und gibt als Ergebnis einen Skalarwert zurück.

Beispiel:

```

DEFINE DATA LOCAL
1  AVGAGE      (I2)
END-DEFINE
...
SELECT AVG (AGE)
  INTO AVGAGE
  FROM SQL-PERSONNEL
  ...

```

Im allgemeinen kann dem Argument *optional* das Schlüsselwort `DISTINCT` vorangestellt werden, um doppelte Werte zu eliminieren, bevor die Funktion ausgeführt wird.

Wenn Sie `DISTINCT` angeben, muss das Argument der Name einer einzelnen Spalte sein; wenn Sie `DISTINCT` nicht angeben, kann das Argument eine allgemeine *scalar-expression* sein.

`DISTINCT` ist nicht erlaubt mit der Funktion `COUNT (*)`, mit der alle Reihen in einer Tabelle — ohne Eliminierung doppelt vorkommender Reihen — gezählt werden.

special-register

Common Set-Syntax:

USER

Extended Set-Syntax:

USER
CURRENT TIMEZONE
CURRENT DATE
CURRENT TIME
CURRENT TIMESTAMP
CURRENT SQLID
CURRENT PACKAGESET
CURRENT SERVER

Bei der Referenzierung eines *special-registers* erhält man einen Skalarwert.

special-register sind — mit Ausnahme von `USER` — nicht Teil der Standard-SQL und werden daher nur als Teil des Natural SQL Extended Set unterstützt.

scalar-function

scalar-functions sind eingebaute, bei der Konstruktion von Skalar-Berechnungen verwendbare Funktionen.

Informationen zu *scalar-functions*, die vom Natural SQL Extended Set unterstützt werden, finden Sie im Abschnitt *Natural SQL Statements – Syntactical Items, scalar-function* in der Natural for DB2-Dokumentation.

scalar-expression-unit

YEAR
YEARS
MONTH
MONTHS
DAY
DAYS
HOUR
HOURS
MINUTE
MINUTES
SECOND
SECONDS
MICROSECOND
MICROSECONDS

Units sind nicht Teil der Standard-SQL und werden daher nur als Teil des Natural SQL Extended Set unterstützt.

case-expression

CASE { <i>searched-when-clause</i> ... } <i>simple-when-clause</i>	ELSE { NULL <i>scalar-expression</i>	END
---	---	------------

Case-expressions sind nicht Teil der Standard-SQL und werden daher nur als Teil des Natural SQL Extended Set unterstützt.

searched-when-clause

WHEN <i>search-condition</i>	THEN { NULL <i>scalar-expression</i>
-------------------------------------	---

Eine *searched-when-clause* ist nicht Teil der Standard-SQL und wird daher nur als Teil des Natural SQL Extended Set unterstützt.

Näheres zu *search-condition* siehe *search-condition*.

simple-when-clause

$\text{scalar-expression} \left\{ \text{WHEN } \text{scalar-expression} \text{ THEN } \left\{ \begin{array}{l} \text{NULL} \\ \text{scalar-expression} \end{array} \right\} \right\} \dots$

Eine *simple-when-clause* ist nicht Teil der Standard-SQL und wird daher nur als Teil des Natural SQL Extended Set unterstützt.

cast-expression

CAST (<i>scalar-expression AS data-type</i>)

Cast-expressions sind nicht Teil der Standard-SQL und werden daher nur als Teil des Natural SQL Extended Set unterstützt.

user-defined-function-reference

Die Option *user-defined-function-reference* gehört zum Natural SQL Extended Set.

Diese Option ermöglicht Ihnen den Aufruf einer beliebigen benutzerdefinierten Funktion. Argumente müssen in Klammern gesetzt werden und durch Kommas voneinander getrennt werden. Die benutzerdefinierte Funktion muss in der Ziel-RDBMS deklariert werden.

sequence-reference

NEXT VALUE FOR <i>sequence-name</i> PREVIOUS VALUE FOR <i>sequence-name</i>
--

Die Option *sequence-reference* gehört zum Natural SQL Extended Set.

Diese Option ermöglicht es Ihnen, den nächsten Wert oder den vorherigen Wert eines Sequence Object zu referenzieren. Das Sequence Object muss im Ziel-RDBMS generiert werden, bevor es zur Laufzeit referenziert werden kann.

scalar-fullselect

<i>(fullselect)</i>

Die Option *user-defined-function-reference* gehört zum Natural SQL Extended Set.

Ein in einem Ausdruck unterstütztes *scalar-fullselect* ist ein in Klammern stehendes *fullselect*, das eine einzelne Reihe zurückgibt, die aus einem einzelnen Spaltenwert besteht. Wenn das *fullselect* keine Reihe zurückgibt, ist das Ergebnis des Ausdrucks der Nullwert. Wenn mehr als eine Reihe für ein *scalar-fullselect* zurückgegeben werden soll, tritt ein Fehler auf.