Beschreibung Domain sepecific languages für den Datatransfer

# Inhaltsverzeichnis

[Inhaltsverzeichnis 2](#_Toc508023279)

[1 Einführung 3](#_Toc508023280)

[2 Variablen 4](#_Toc508023281)

[2.1 Literale 4](#_Toc508023282)

[2.2 Relationen 4](#_Toc508023283)

[2.3 Arithmetische Operatoren 4](#_Toc508023284)

[2.4 Bitweise Operatoren 4](#_Toc508023285)

[2.5 String Operatoren 4](#_Toc508023286)

[2.6 Logische Operatoren 4](#_Toc508023287)

[2.7 Funktionen 5](#_Toc508023288)

[2.7.1 Mathematisch 5](#_Toc508023289)

[2.7.2 Text 5](#_Toc508023290)

[2.7.3 Date 6](#_Toc508023291)

[2.7.4 Konvertierung 7](#_Toc508023292)

[2.7.5 Logisch 7](#_Toc508023293)

[3 Custom SQL 9](#_Toc508023294)

# Einführung

Der Datatransfer nutzt sogenannte DSLs um diverse Datenquellen anzufragen und um Variablen zu deklarieren. Diese werden hier beschrieben.

# Variablen

**Der Datatransfer erlaubt die Verwendung von Variablen in TransferTableJobs mit dem XML-Element <variable>.**

**Dabei wird als Attribut der Name und der Typ angegeben.**

**Zur Initialisierung gibt es 3 Varianten:**

* **Festwert – value Attribut**
* **SELECT-Ausdruck**
  + **Attribut selectContext gibt an wo das SQL ausgeführt werden soll**
  + **Attribut selectStmt gibt das SQL-Statement an.**
  + **Variablen können mit ${{VarName}} eingebettet werden – Formatierungen und Begrenzungszeichen werden automatisch eingesetzt je nach Datentyp**
    - **String Var Wert hallo wird zu ‘hallo‘**
* **Expression-Ausdruck**
  + **Verwendet eine DSL mit Standardoperatoren**

**Die Expression-Language hat folgende Spezifikation**

## ****Literale****

Texte werden mit einfachem Hochkomma getrennt. Es sind Maskierungen erlaubt wie \` \n \r \t \\

Zahlen werden in Klartext geschrieben.

Kommazahlen werden mit . getrennt.

Boolean-Literale sind true und false im Klartext.

Null-Literal ist null im Klartext.

## ****Relationen****

Folgende Relationen sind erlaubt:

<, <=, >, >=, =, ==, !=, <>

## Arithmetische Operatoren

+, -, \*, /, %

## Bitweise Operatoren

&, |

## String Operatoren

+

## Logische Operatoren

&&, ||, and, or

## Funktionen

Die Expression-Sprache ist eine funktionale Sprache. Funktionen werden mit Namen und Parametern aufgerufen. Beispiel

Funtkionsname( param1, param2, …, param n)

Folgende Funtkionen werden für Variablen unterstützt.

Die Sprache ist case-insensitiv. Es ist also unwichtig ob Funktionsnamen klein oder groß geschrieben werden.

### Mathematisch

|  |  |  |
| --- | --- | --- |
| **Name** | **Parameter** | **Beschreibung** |
| Sin | Zahl X  Returns Zahl | Berechnet den Sinus von x |
| Cos | Zahl X  Returns Zahl | Berechnet den Cosinus von x |
| Tan | Zahl X  Returns Zahl | Berechnet den Cosinus von x |
| Abs | Zahl X  Returns Zahl | Berechnet den Betrag von x |
| Pi | N/A  Returns Zahl | PI |
| Ceiling | Zahl X  Returns Zahl | Rundet eine Zahl auf die nächste ganzzahlige Zahl auf |
| Floor | Zahl X  Returns Zahl | Rundet eine Zahl auf die nächste ganzzahlige Zahl ab |
| Round / rnd | Zahl X  [Ganzzahl Genauigkeit]  Returns Zahl | Rundet die Zahl [x] nach mathematischen Rundungsregeln auf die nächste geradzahlige Zahl.  Ist Genauigkeit angegeben wird auf die [Genauigkeit]-te Nachkommastelle gerundet |
| Max | Zahl X  Zahl Y  Returns Zahl | Gibt das Maximum von [X] und [Y] zurück |
| Min | Zahl X  Zahl Y  Returns Zahl | Gibt das Minimum von [X] und [Y] zurück |

### Text

|  |  |  |
| --- | --- | --- |
| **Name** | **Parameter** | **Beschreibung** |
| ToUpper | Text Eingabe  Returns Text | Nimmt den Text [Eingabe] und gibt wandelt alle Kleinbuchstaben in Großbuchstaben um. |
| ToLower | Text Eingabe  Returns Text | Nimmt den Text [Eingabe] und gibt wandelt alle Großbuchstaben in Kleinbuchstaben um. |
| IndexOf | Text Eingabe  Text Suchtext  Returns Zahl | Sucht im Text [Eingabe] den Text [Suchtext] und gibt die Indexposition im Eingabe-Text zurück, sofern gefunden.  Wenn der Suchtext nicht gefunden wird, wird -1 zurückgegeben. |
| Replace | Text Eingabe  Text Suchtext  Text Ersatztext  Returns Text | Sucht alle Vorkommen von [Suchtext] in [Eingabe] und ersetzt sie mit [Ersatztext]. |
| Substring | Text Eingabe  Zahl startIndex  [Zahl Länge]  Returns Text | Gibt einen Teiltext von [Eingabe] zurück. Dieser ergibt sich von der Startposition [startIndex] bis zum Schluss oder wenn angegeben mit der Länge [Länge]. |
| strContains / contains | Text Eingabe  Text Suchtext  Returns Bool | Prüft ob in [Eingabe] der Text [Suchtext] vorkommt.  Wenn ja ist das Ergebnis true, sonst false. |
| strLeft / left | Text Eingabe  Text Suchtext  Returns Text | Sucht im Text [Eingabe] den Suchtext [Suchtext] (erstes Vorkommen von Links) und gibt alles links von der gefundenen Stelle zurück. Wird der Ausdruck [SuchText] nicht gefunden ist die Rückgabe ein Leerstring. |
| strRight / right | Text Eingabe  Text Suchtext  Returns Text | Sucht im Text [Eingabe] den Suchtext [Suchtext] (erstes Vorkommen von Links) und gibt alles rechts von der gefundenen Stelle zurück. Wird der Ausdruck [SuchText] nicht gefunden ist die Rückgabe ein Leerstring. |
| strMid / Mid | Text Eingabe  Text SuchtextStart  Text SuchtextEnde  Returns Text | Sucht im Text [Eingabe] den Suchtext [SuchtextStart] (erstes Vorkommen von Links) und den Suchtext [SuchtextEnde] nach diesem Treffer. Der Text zwischen diesen beiden Stellen wird zurückgegeben. Wird einer der Ausdrücke nicht gefunden ist die Rückgabe ein Leerstring. |
| startsWith | Text Eingabe  Text Suchtext  Returns Bool | Prüft ob der Text [Eingabe] mit [Suchtext] beginnt.  Wenn ja true, sonst false. |
| endsWith | Text Eingabe  Text Suchtext  Returns Bool | Prüft ob der Text [Eingabe] mit [Suchtext] endet.  Wenn ja true, sonst false. |

### Date

|  |  |  |
| --- | --- | --- |
| **Name** | **Parameter** | **Beschreibung** |
| Date | Zahl Ticks  ODER  Zahl jahr  Zahl monat  Zahl tag  [Zahl Stunde  Zahl Minute  Zahl Sekunde]  Returns Datum | Erstellt ein Datum ausgehend von den Anzahl der Ticks [ticks]  ODER  Erstellt ein Datum ausgehend der Datumskomponenten [jahr] [monat] [Tag] und wenn angegeben, dann auch [Stunde], [Minute] [Sekunde]. |
| AdjustSeconds | Datum Eingabe  Zahl Anzahl  Returns Datum | Addiert auf das Datum [Eingabe] [Anzahl] Sekunden und gibt das neue Datum zurück. |
| Adjustminutes | Datum Eingabe  Zahl Anzahl  Returns Datum | Addiert auf das Datum [Eingabe] [Anzahl] Minuten und gibt das neue Datum zurück. |
| AdjustHours | Datum Eingabe  Zahl Anzahl  Returns Datum | Addiert auf das Datum [Eingabe] [Anzahl] Stunden und gibt das neue Datum zurück. |
| AdjustDays | Datum Eingabe  Zahl Anzahl  Returns Datum | Addiert auf das Datum [Eingabe] [Anzahl] Tage und gibt das neue Datum zurück. |
| AdjustMonths | Datum Eingabe  Zahl Anzahl  Returns Datum | Addiert auf das Datum [Eingabe] [Anzahl] Monate und gibt das neue Datum zurück. |
| AdjustYears | Datum Eingabe  Zahl Anzahl  Returns Datum | Addiert auf das Datum [Eingabe] [Anzahl] Jahre und gibt das neue Datum zurück. |
| Second | Datum Eingabe  Returns Zahl | Gibt den Sekundenteil des Datums [Eingabe] zurück |
| Minute | Datum Eingabe  Returns Zahl | Gibt den Minutenteil des Datums [Eingabe] zurück |
| Hour | Datum Eingabe  Returns Zahl | Gibt den Stundenteil des Datums [Eingabe] zurück |
| Day | Datum Eingabe  Returns Zahl | Gibt den Tagesteil des Datums [Eingabe] zurück |
| Month | Datum Eingabe  Returns Zahl | Gibt den Monatsteil des Datums [Eingabe] zurück |
| Year | Datum Eingabe  Returns Zahl | Gibt den Jahresteil des Datums [Eingabe] zurück |

### Konvertierung

|  |  |  |
| --- | --- | --- |
| **Name** | **Parameter** | **Beschreibung** |
| cstr | Typneutral [Eingabe]  Returns Text | Transformiert die [Eingabe] in einen Text und gibt ihn zurück |
| Cbool | Typneutral [Eingabe]  Returns Bool | Transformiert die [Eingabe] in einen Bool-Wert und gibt ihn zurück. Ist dies nicht möglich kommt es zu einen Fehler. |
| Cint | Typneutral [Eingabe]  Returns Ganzzahl | Transformiert die [Eingabe] in einen Ganzzahl-Wert und gibt ihn zurück. Ist dies nicht möglich kommt es zu einen Fehler. |
| Cdbl | Typneutral [Eingabe]  Returns Zahl | Transformiert die [Eingabe] in einen Zahl-Wert und gibt ihn zurück. Ist dies nicht möglich kommt es zu einen Fehler. |
| Cdate | Typneutral [Eingabe]  Returns Datum | Transformiert die [Eingabe] in einen Datum-Wert und gibt ihn zurück. Ist dies nicht möglich kommt es zu einen Fehler. |
| cChar | Typneutral [Eingabe]  Returns Zeichen | Transformiert die [Eingabe] in einen Zeichen-Wert und gibt ihn zurück. Ist dies nicht möglich kommt es zu einen Fehler. |

### Logisch

|  |  |  |
| --- | --- | --- |
| **Name** | **Parameter** | **Beschreibung** |
| If / iif,/ case / casewhen | Bool Bedingung1  Typneutral Ergebnis1  [Bool Bedingung 2-n  Typneutral Ergebnis 2-n]  Typneutral ElseErgebnis  Returns Typneutral | Prüft [Bedingung1} und gibt im true-Fall [Ergebnis1] zurück. Prüft danach jede Folgebedingung in der Reiehenfolge des auftretens und gibt analog das entsprechende Ergebnis zurück.  Tritt keiner der [Bedingungx] ein wird [ElseErgebnis] zurückgegeben |
| Nvl | Typneutral [Eingabe]  Typneutral [ElseErgebnis]  Returns Typneutral | Prüft ob [Eingabe] null ist. Wenn ja wird [ElseErgebnis] zurück gegeben, sonst [Eingabe] |
| Not | Bool Eingabe  Returns bool | Wechselt den Wahrheitswert von [Eingabe] von true zu false oder umgekehrt und gibt dies zurück. |

# Custom SQL

Custom SQL erlaubt eine SQL-Syntax gegen Datenquellen die eigentlich kein SQL unterstützen.

Besonderheiten sind das alle „Tabellen“ und berechneten Spalten benannt sein müssen.

Für berechnete Spalten gilt die Logik die auch für die Variablensprache gilt.

Benennungen von Spalten oder „Tabellen“ wird mit „AS Name“ durchgeführt.

z.B. SELECT 1 as Spalte from Tabelle as T

Es werden einfache Joins (inner join) mit “=” akzeptiert

z.B. Select T1.Key from Tab1 as T1 inner join Tab2 T2 on T1.Key = T2.Key

Bei mehreren Tabellen mus jede Spalte vollqualifiziert angegeben warden.

Variablen aus der Variablen-Definition können mit ${{Varname}} eingesetzt werden

z.B. SELECT 3 + ${{ZahlVar}} as Calc from Tab AS T1