文章
姚 鑫 · 四月 9, 2022 阅读大约需 7 分钟

第107章 SQL函数 $PIECE

第107章 SQL函数 $PIECE

返回由分隔符标识的子字符串的字符串函数。

大纲

```
$PIECE(string-expression,delimiter[,from[,to]])
```

参数

- string-expression - 要从中提取子字符串的目标字符串。
  表达式，可以是字段名、字符串字面值、数字或其他函数的结果。
- delimiter - 用于标识子字符串的分隔符。
- from - 可选-指定要从目标字符串返回的子字符串或子字符串范围的开头的整数。
  子字符串由分隔符分隔，从1开始计数。
  如果省略，则返回第一个子字符串。
- to - 可选-一个整数，指定从目标字符串返回的子字符串范围的结束子字符串。
  必须与from连用。

描述

$PIECE从字符串表达式中返回指定的子字符串(PIECE)。
返回的子字符串取决于所使用的参数:

- $PIECE(string-expression,delimiter)返回string-expression中的第一个子字符串。
  如果delimiter出现在字符串表达式中，则这是在delimiter第一次出现之前的子字符串。
  如果分隔符没有出现在字符串表达式中，则返回的子字符串为字符串表达式。
- $PIECE(string-expression,delimiter,from)返回string-expression的第n个片段的子字符串，其中整数n由from参数指定，片段由一个分隔符分隔。
  分隔符不返回。
- $PIECE(string-expression,delimiter,from,to)返回一个范围的子字符串，包括从中指定的子字符串到到中指定的子字符串。
  $PIECE的这种4个参数形式返回一个字符串，其中包括在from和to子字符串之间出现的任何中间分隔符。
  如果to大于子字符串的数量，则返回的子字符串包括到string-expression字符串结尾的所有子字符串。

参数

string-expression

要从其中返回子字符串的字符串。
它可以是字符串字面值、变量名或任何计算结果为字符串的有效表达式。

字符串通常包含用作分隔符的字符(或字符串)的实例。

此字符或字符串也不能用作字符串表达式中的数据值。

如果你指定空字符串(null)作为目标字符串，$PIECE返回<null>，空字符串。

delimiter

用于在字符串表达式中分隔子字符串的搜索字符串。
它可以是数字字面值或字符串字面值(用引号括起来)、变量名或计算结果为字符串的表达式。

通常，分隔符是一个指定的字符，它永远不会在字符串数据中使用，但仅用于作为分隔子字符串的分隔符使用。
分隔符也可以是多字符搜索字符串，其中的单个字符可以在字符串数据中使用。

如果指定空字符串(null)作为分隔符，$PIECE返回<null>，空字符串。

from

字符串表达式中的子字符串数，从1开始计算。
它必须是正整数、整型变量的名称或计算结果为正整数的表达式。
子字符串由分隔符分隔。

- 如果from参数被省略或设置为1，$PIECE返回string-expression的第一个子字符串。
  如果string-expression不包含指定的分隔符，from值为1将返回string-expression。
- 如果from参数通过计数来标识string-expression中的最后一个子字符串，则返回这个子字符串，无论它后面是否有分隔符。
- 如果from的值是NULL，空字符串，零，或负数，并且指定了no to参数，$PIECE返回一个空字符串。
  但是，如果指定了to参数，$PIECE将这些from值视为from=1。
- 如果from的值大于string-expression中子字符串的数量，$PIECE返回一个空字符串。

如果from参数与to参数一起使用，它将标识将作为字符串返回的子字符串范围的开始，并且应该小于to的值。

to

字符串表达式中结束from参数初始化的范围的子字符串的数目。
返回的字符串包括from和to子字符串，以及任何中间子字符串和分隔它们的分隔符。
to参数必须为正整数、整型变量的名称或计算结果为正整数的表达式。
to参数必须与from一起使用，并且应该大于from的值。

- 如果from小于to， $PIECE返回一个由该范围内所有分隔的子字符串组成的字符串，包括from和to子字符串。
  这个返回的字符串包含此范围内的子字符串和分隔符。
- 如果to大于带分隔符的子字符串的数量，则返回的字符串包含所有字符串数据(子字符串和分隔符)，从from子字符串开始，一直到字符串表达式字符串的结尾。
- 如果from等于to，则返回from子字符串。
- 如果from大于to， $PIECE返回一个空字符串。
- 如果to是空字符串(null)， $PIECE返回一个空字符串。

示例

下面的示例返回由"，"分隔符标识的第一个子字符串'Red':

```
SELECT $PIECE('Red,Green,Blue,Yellow,Orange,Black',',')
```

```
Red
```

下面的示例返回由"，"分隔符标识的第三个子字符串'Blue':

```
SELECT $PIECE('Red,Green,Blue,Yellow,Orange,Black',',',3)

Blue
```

下面的例子返回'Blue,Yellow,Orange'， colorlist中的第三到第五个元素，用"，"分隔:

```
SELECT $PIECE('Red,Green,Blue,Yellow,Orange,Black',',',3,5)

Blue,Yellow,Orange
```

下面的$PIECE函数都返回'123'，表明当from为1时，双参数形式等价于三参数形式:

```
SELECT $PIECE('123#456#789','#') AS TwoArg

123

SELECT $PIECE('123#456#789','#',1) AS ThreeArg

123
```

下面的示例使用多字符分隔符字符串'#-#'返回第三个子字符串'789'。
这里，分隔符字符串的组成字符'#'和'-'可以用作数据值;
只保留指定的字符序列(#-#):

```
SELECT $PIECE('1#2-3#-#45##6#-#789','#-#',3)

3
```

下面的例子返回 “ MAR;APR;MAY ” 。
它们由第三个到第五个子字符串组成，由';'分隔符标识:

```
SELECT $PIECE('JAN;FEB;MAR;APR;MAY;JUN',';',3,5)

MAR;APR;MAY
```

下面的例子使用$PIECE从员工名和供应商联系人名中提取出姓氏，然后执行一个JOIN，返回与供应商联系人姓相同的员工的实例:

```
SELECT E.Name,V.Contact
FROM Sample.Employee AS E INNER JOIN Sample.Vendor AS V
ON $PIECE(E.Name,',')=$PIECE(V.Contact,',')
```

注意

使用$PIECE解压数据值

$PIECE 通常用于 “ 解包 ” 包含由分隔符分隔的多个字段的数据值。典型的分隔符包括斜杠 (/)、逗号 (,)、空格 () 和分号 (;)。以下示例值非常适合与 $PIECE 一起使用：

```
'John Jones/29 River St./Boston MA, 02095'
'Mumps;Measles;Chicken Pox;Diptheria'
'45.23,52.76,89.05,48.27'
```

$PIECE 和 $LENGTH

$LENGTH 的双参数形式根据分隔符返回字符串中子字符串的数量。使用 $LENGTH 确定字符串中子字符串的数量，然后使用 $PIECE 提取单个子字符串。

$PIECE and $LIST

$PIECE 和 $LIST 函数使用的数据存储技术不兼容，不应组合使用。例如，尝试在使用 $LISTBUILD 创建的列表上使用 $PIECE 会产生不可预测的结果，应该避免。对于 SQL 函数和相应的 ObjectScript 函数都是如此。

$LIST 函数指定子字符串而不使用指定的分隔符。如果留出分隔符或字符序列不适合数据类型（例如，位串数据），则应使用 $LISTBUILD 和 $LIST SQL 函数来存储和检索子字符串。

Null Values

$PIECE 不区分具有空字符串值 (NULL) 的分隔子字符串和不存在的子字符串。两者都返回 <null>,，即空字符串值。例如，以下示例均返回值为 7 的空字符串：

```
SELECT $PIECE('Red,Green,Blue,Yellow,Orange,Black',',',7)

NUll
```

```
SELECT $PIECE('Red,Green,Blue,Yellow,Orange,Black,',',',7)

NULL
```

在第一种情况下，没有第七个子串；返回一个空字符串。在第二种情况下，有第七个子字符串，由字符串表达式字符串末尾的分隔符指示；第七个子字符串的值是空字符串。

以下示例显示字符串表达式中的空值。它提取子字符串 3。此子字符串存在，但包含一个空字符串：

```
SELECT $PIECE('Red,Green,,Blue,Yellow,Orange,Black,',',',3)

NULL
```

以下示例还返回空字符串，因为指定的子字符串不存在：

```
SELECT $PIECE('Red,Green,,Blue,Yellow,Orange,Black,',',',0)

NULL


SELECT $PIECE('Red,Green,,Blue,Yellow,Orange,Black,',',',8,20)

NULL
```

在以下示例中，$PIECE 函数返回整个字符串表达式字符串，因为字符串表达式字符串中没有出现分隔符：

```
SELECT $PIECE('Red,Green,Blue,Yellow,Orange,Black,','#')

Red,Green,Blue,Yellow,Orange,Black,
```

嵌套 $PIECE 操作

要执行复杂的提取，可以将 $PIECE 引用相互嵌套。内部 $PIECE 返回一个由外部 $PIECE 操作的子字符串。每个 $PIECE 都使用自己的分隔符。例如，以下返回状态缩写“ MA ”：

```
SELECT $PIECE($PIECE('John Jones/29 River St./Boston MA 02095','/',3),' ',2)

MA
```

下面是嵌套$PIECE操作的另一个例子，它使用了分隔符的层次结构。
首先，内部的$PIECE使用插入符号(^)分隔符来查找字符串的第二部分'A,B,C'。
然后外部$PIECE使用逗号(，)分隔符返回子串'A,B,C'的第一块和第二块('A,B'):

```
SELECT $PIECE($PIECE('1,2,3^A,B,C^@#!','^',2),',',1,2)

A,B
```

#SQL #Caché

源URL: https://cn.community.intersystems.com/post/%E7%AC%AC107%E7%AB%A0-sql%E5%87%BD%E6%95%B0-piece