

文章

[姚鑫](#) · 三月 12 阅读大约需分钟

第十九章 SQL函数 \$LIST(一)

第十九章 SQL函数 \$LIST(一)

返回列表中元素的列表函数。

大纲

`$LIST(list[,position[,end]])`

参数

- list - 计算结果为有效列表的表达式。列表是包含一个或多个元素的编码字符串。可以使用 SQL 或 ObjectScript 的 \$LISTBUILD 或 \$LISTFROMSTRING 函数创建列表。可以使用 SQL 或 ObjectScript \$LIST 函数从现有列表中提取列表。
- position - 可选 - 指定列表中的起始位置。计算结果为整数的表达式。
- end - 可选 - 指定列表中的结束位置。计算结果为整数的表达式。

描述

\$LIST 从列表中返回元素。返回的元素取决于使用的参数。

- \$LIST(list) 将列表中的第一个元素作为文本字符串返回。
- \$LIST(list,position) 将指定位置指示的元素作为文本字符串返回。位置参数必须计算为整数。
- \$LIST(list,position,end)

返回一个“子列表”（编码的列表字符串），其中包含从指定开始位置到指定结束位置的列表元素。

此函数返回 VARCHAR 类型的数据。

参数

list

包含一个或多个元素的编码字符串。可以使用 SQL \$LISTBUILD 函数或 ObjectScript \$LISTBUILD 函数创建列表。可以使用 SQL \$LISTFROMSTRING 函数或 ObjectScript \$LISTFROMSTRING 函数将分隔字符串转换为列表。您可以使用 SQL \$LIST 函数或 ObjectScript \$LIST 函数从现有列表中提取列表。

position

要返回的列表元素的位置。列表元素从 1 开始计数。如果省略 position，则返回第一个元素。如果 position 的值是 0 或大于列表中的元素数，SQL 不会返回值。如果 position 的值为负一（-1），则 \$LIST 返回列表中的最后一个元素。

```

/// d ##class(PHA.TEST.SQLCommand).Li()
ClassMethod Li()
{
  s a = $lb("Red", "Blue", "Green")
  &sql(
    SELECT $LIST(:a,-1)
    INTO :b
  )
  if SQLCODE '= 0 {
    w !,"Error code ",SQLCODE
  } else {
    w !,"The last element is ",b
  }
}

```

```
DHC-APP>d ##class(PHA.TEST.SQLCommand).Li()
```

```
The last element is Green
```

如果指定了 end 参数,则 position 指定元素范围中的第一个元素。即使只返回一个元素(当位置和结束是相同的数字时),该元素也会作为编码列表字符串返回。因此,\$LIST(x,2)(将元素作为普通字符串返回)与 \$LIST(x,2,2)(将元素作为编码列表字符串返回)不同。

end

元素范围内最后一个元素的位置。必须指定位置才能指定结束。当指定 end 时,返回的值是一个编码的列表字符串。由于这种编码,此类字符串只能由其他 \$LIST 函数处理。

如果 end 的值为:

- 大于位置,则返回包含元素列表的编码字符串。
- 等于位置,返回一个包含一个元素的编码字符串。
- 小于位置,不返回值。
- 大于列表中的元素个数,相当于指定列表中的最后一个元素。
- 负一 (-1),相当于指定列表中的最后一个元素。

指定 end 时,可以指定零 (0) 的位置值。在这种情况下,0 等于 1。

使用列表

一个表可以包含一个或多个列表字段。因为列表是编码字符串,所以这些字段可以定义为数据类型 %List (%Library.List) 或数据类型 VARCHAR。数据类型 %List 的字段可以标识为 CType(客户端数据类型)= 6。

数据类型在插入或更新时不限制字段的允许值。因此,用户必须确保 list 字段中的所有数据值都是 List 编码字符串。如果 SQL \$LIST 函数遇到未编码的字符串数据值,则 SELECT 操作将失败并返回带有 %msg 的 SQLCODE -400,如所示: Unexpected error occurred: <LIST>%0AmBuncommitted+1^%sqlcq.USER.cls61.1.

可以通过使用主变量或通过 SQL 中指定 \$LISTBUILD 将列表提供给 SQL \$LIST 函数。两者都显示在以下嵌入式 SQL 示例中:

```

/// d ##class(PHA.TEST.SQLCommand).Lil()
ClassMethod Lil()

```

```
{
  s mylist = $lb("Red", "Blue", "Green")
  &sql(
    SELECT $LIST(:mylist, 2), $LIST($LISTBUILD('Red', 'Blue', 'Green'), 3)
    INTO :a,:b
  )
  if SQLCODE '= 0 {
    w !,"Error code ",SQLCODE
  } else {
    w !,"The host variable list element is ",a,!
    w !,"The SQL $LISTBUILD list element is ",b,!
  }
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLCommand).Li1()
```

```
The host variable list element is Blue
```

```
The SQL $LISTBUILD list element is Green
```

可以使用 \$LIST 函数从另一个列表中提取列表:

```
/// d ##class(PHA.TEST.SQLCommand).Li2()
ClassMethod Li2()
{
  s a = $lb("Red", "Blue", "Green")
  &sql(
    SELECT $LIST(:a,2, 3)
    INTO :b
  )
  if SQLCODE '= 0 {
    w !,"Error code ",SQLCODE
  } else {
    &sql(
      SELECT $LIST(:b,1)
      INTO :c
    )
    if SQLCODE '= 0 {
      w !,"Error code ",SQLCODE
    } else {
      w !,"The element returned is ",c }
    }
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLCommand).Li2()
```

```
The element returned is Blue
```

在以嵌入式 SQL 示例中, subList

不是有效的列表参数,因为它是作为普通字符串返回单个元素,而不是编码列表字符串。只有 \$LIST 的三参数形式返回编码列表字符串。在这种情况下,会生成 QLCODE -400 致命错误:

```
/// d ##class(PHA.TEST.SQLCommand).Li3()
```

