

文章

[姚鑫](#) · 二月 16, 2022 阅读大约需 5 分钟

第五十六章 SQL函数 \$EXTRACT

第五十六章 SQL函数 \$EXTRACT

按位置从字符串中提取字符的字符串函数。

大纲

`$EXTRACT(string[,from[,to]])`

参数

- string - 要从中提取子字符串的目标字符串。
- from - 可选-单个字符在目标字符串中的位置，或要提取的字符范围(包括)的开头。指定为从1开始计数的正整数。
- to - 可选-要提取的字符范围的结束位置(包括)。指定为从1开始计数的正整数。

描述

`$EXTRACT`返回字符串中指定位置的子字符串。
返回的子字符串的性质取决于所使用的参数。

- `$EXTRACT(string)`提取字符串中的第一个字符。
- `$EXTRACT(string,from)`从from指定的位置提取字符。
例如，如果变量var1包含字符串“ABCD”，下面的命令提取“B”(第二个字符):

```
SELECT $EXTRACT('ABCD',2) AS Extracted
```

2

- `$EXTRACT(string,from,to)`提取以from位置开始，以to位置结束的字符范围。
例如，下面的命令从字符串“1234Alabama567”中提取字符串“Alabama”(即从位置5到位置11的所有字符，包括在内):

```
SELECT $EXTRACT('1234Alabama567',5,11) AS Extracted
```

Alabama

这个函数返回VARCHAR类型的数据。

参数

string

字符串值可以是变量名、数字值、字符串面值或任何有效表达式。

from

from值必须是正整数(但是, 请参见注释)。

如果是小数, 则截断该小数部分, 只使用整数部分。

如果from值大于字符串中的字符数, \$EXTRACT返回一个空字符串。

如果指定了from而没有指定to参数, 则提取指定的单个字符。

如果与to参数一起使用, 它标识要提取的范围的开始, 并且必须小于to的值。

如果from = to, \$EXTRACT返回指定位置的单个字符。

如果from大于to, \$EXTRACT返回一个空字符串。

to

to参数必须与from参数一起使用。

它必须是一个正整数。

如果是小数, 则截断该小数部分, 只使用整数部分。

如果to的值大于或等于from的值, \$EXTRACT返回指定的子字符串。

如果to大于字符串的长度, \$EXTRACT返回从位置到字符串末尾的子字符串。

如果to小于from, \$EXTRACT返回一个空字符串。

示例

下面的示例返回字符串中的第四个字符" S":

```
SELECT $EXTRACT('THIS IS A TEST',4) AS Extracted
```

S

下面的示例返回由第一个到第7个字符组成的子字符串 " THIS IS "。

```
SELECT $EXTRACT('THIS IS A TEST',1,7) AS Extracted
```

THIS IS

下面的嵌入式SQL示例从a中提取第二个字符(" B "), 并将这个值赋给变量y。

```
ClassMethod Extract()  
{  
    s a="ABCD"  
    &sql(SELECT $EXTRACT(:a,2) INTO :y)  
    if SQLCODE '= 0 {  
        w !,"Error code ",SQLCODE }  
    else {
```

```
w !,"The extract returns ",y }
}
```

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

The extract returns B

下面的嵌入式SQL示例显示，当from值为“1”时，单参数格式等价于双参数格式。两个\$EXTRACT函数都返回“H”。

```
/// d ##class(PHA.TEST.SQLCommand).Extract1()
ClassMethod Extract1()
{
    s a="HELLO"
    &sql(SELECT $EXTRACT(:a),$EXTRACT(:a,1) INTO :b,:c)
    if SQLCODE'=0 {
        w !,"Error code ",SQLCODE }
    else {
        w !,"The one-arg form returns ",b
        w !,"The two-arg form returns ",c }
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLCommand).Extract1()
```

The one-arg form returns H

The two-arg form returns H

注意

\$EXTRACT与\$PIECE和\$LIST比较

\$EXTRACT从字符串中按整数位置返回一个子字符串。

\$PIECE和\$LIST都适用于特殊格式的字符串。

\$PIECE从标准字符串中使用分隔符返回子字符串。

\$LIST通过元素的整数位置(不是字符)返回编码列表中的元素的子列表。

\$LIST不能用于普通字符串，而\$EXTRACT不能用于编码列表。

\$EXTRACT、\$FIND、\$LENGTH和\$PIECE函数对标准字符串进行操作。

各种\$LIST函数对编码的字符串进行操作，这些字符串与标准字符串不兼容。

唯一的例外是\$LISTGET函数和\$LIST的单参数和双参数形式，它们接受已编码的字符串作为输入，但将单个元素值作为标准字符串输出。

\$EXTRACT 与 Unicode

\$EXTRACT函数对字符而不是字节进行操作。

因此，Unicode字符串的处理方式与ASCII字符串相同，如下所示的嵌入式SQL示例使用Unicode字符“pi”(\$CHAR(960)):

```
ClassMethod Extract2()
{
    s a="QT PIE"
    s b=("QT " _$CHAR(960))
    &sql(SELECT
    $EXTRACT(:a, -33, 4),
    $EXTRACT(:a, 4, 4),
    $EXTRACT(:a, 4, 99),
    $EXTRACT(:b, -33, 4),
    $EXTRACT(:b, 4, 4),
    $EXTRACT(:b, 4, 99)
    INTO :a1, :a2, :a3, :b1, :b2, :b3)
    if SQLCODE'=0 {
        w !, "Error code ", SQLCODE }
    else {
        w !, "ASCII form returns ", !, a1, !, a2, !, a3
        w !, "Unicode form returns ", !, b1, !, b2, !, b3 }
}
```

```
DHC-APP> d ##class(PHA.TEST.SQLCommand).Extract2()
```

ASCII form returns

QT P

P

PIE

Unicode form returns

QT ?

?

?

Null和无效参数

- 当string为null字符串时，返回一个null字符串。
 - 当from是一个大于字符串长度的数字时，将返回一个空字符串。
 - 当from为零或负数，并且指定了no to时，将返回一个空字符串。
 - 当to为零、负数或比from小的数字时，将返回一个空字符串。
 - 当to为有效值时，from可以为零或负数。
- \$EXTRACT将这些值视为1。

无效的参数值不会产生SQLCODE错误。

在下面的例子中，from值的负数被计算为1；

\$EXTRACT返回由第一个到第7个字符组成的子字符串“ THIS IS ”。

```
SELECT $EXTRACT('THIS IS A TEST', -7, 7)
```

THIS IS

在下面的嵌入式SQL示例中，所有的\$EXTRACT函数调用都返回空字符串：

```
ClassMethod Extract3()
{
    s a="THIS IS A TEST"
    s b=""
```

```
&sql(SELECT
$EXTRACT(:a,33),
$EXTRACT(:a,-7),
$EXTRACT(:a,3,2),
$EXTRACT(:a,-7,0),
$EXTRACT(:a,-7,-10),
$EXTRACT(:b,-33,4),
$EXTRACT(:b,4,4),
$EXTRACT(:b,4,99),
$EXTRACT(NULL,-33,4),
$EXTRACT(NULL,4,4),
$EXTRACT(NULL,4,99)
INTO :a1,:a2,:a3,:a4,:a5,:b1,:b2,:b3,:c1,:c2,:c3)
if SQLCODE'=0 {
    w !,"Error code ",SQLCODE }
else {
    w !,"FROM too big: ",a1
    w !,"FROM negative, no TO: ",a2
    w !,"TO smaller than FROM: ",a3
    w !,"TO not a positive integer: ",a4,a5
    w !,"LIST is null string: ",b1,b2,b3,c1,c2,c3 }
}
```

DHC-APP>d ##class(PHA.TEST.SQLCommand).Extract3()

FROM too big:
FROM negative, no TO:
TO smaller than FROM:
TO not a positive integer:
LIST is null string:

[#SQL](#) [#Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%94%E5%8D%81%E5%85%AD%E7%AB%A0-sql%E5%87%BD%E6%95%B0-extract>