

文章

[姚鑫](#) · 二月 17, 2022 阅读大约需 4 分钟

第五十七章 SQL函数 \$FIND

第五十七章 SQL函数 \$FIND

字符串函数，返回字符串中子字符串的结束位置，可选的搜索起始点。

大纲

`$FIND(string, substring[, start])`

参数

- `string` - 要搜索的目标字符串。
它可以是变量名、数值、字符串面值或任何有效表达式。
- `substring` - 要搜索的子字符串。
它可以是变量名、数值、字符串面值或任何有效表达式。
- `start` - 可选-子字符串搜索的起始点，指定为正整数。
从字符串开始的字符计数，从1开始计数。
若要从字符串的开头开始搜索，请忽略此参数或指定从0或1开始。
负数、空字符串或非数字值将被视为0。
指定`start`为NULL会导致\$FIND返回`< NULL >`。

\$FIND返回SMALLINT数据类型。

描述

\$FIND返回一个整数，指定子字符串在字符串中的结束位置。

\$FIND搜索字符串的子字符串。

如果找到子字符串，\$FIND返回子字符串后面第一个字符的整数位置。

如果substring未找到，\$FIND返回0值。

可以包含start选项来指定搜索的起始位置。

如果start大于字符串中的字符数，\$FIND返回一个值为0。

如果省略start，则默认为字符串位置1。

如果起始值为0、负数或非数字字符串，则位置1是默认值。

\$FIND是区分大小写的。

使用其中一个大小写转换函数来定位字母或字符串的大写和小写实例。

\$FIND, POSITION, CHARINDEX, INSTR

\$FIND, POSITION,

CHARINDEX和INSTR都在字符串中搜索指定的子字符串，并返回与第一个匹配项对应的整数位置。

\$FIND返回匹配子字符串结束后第一个字符的整数位置。

CHARINDEX、POSITION和INSTR返回匹配子字符串的第一个字符的整数位置。

CHARINDEX、\$FIND和INSTR支持指定子字符串搜索的起始点。
INSTR还支持从起始点指定子字符串出现。

下面的示例演示了这四个函数，指定了所有可选参数。
注意，在这些函数中，string和substring的位置不同：

```
SELECT POSITION('br' IN 'The broken brown briefcase') AS Position,  
       CHARINDEX('br','The broken brown briefcase',6) AS Charindex,  
       $FIND('The broken brown briefcase','br',6) AS Find,  
       INSTR('The broken brown briefcase','br',6,2) AS Inst
```

5 12 14 18

示例

在下面的例子中，字符串包含字符串“ABCDEFGH”，子字符串包含字符串“BCD”。
\$FIND函数返回值5，表示字符(“E”) 在“BCD”后面的位置：

```
SELECT $FIND('ABCDEFGH','BCD') AS SubPoint
```

5

在示例中，通过数字“987654321”查找数字“7”。
它返回4，子字符串后面的位置：

```
SELECT $FIND(987654321,7) AS SubPoint
```

4

下面的例子返回3，即子字符串“AA”的第一个实例后面的字符位置：

```
SELECT $FIND('AAAAAA','AA') AS SubPoint
```

3

在下面的例子中，\$FIND搜索不在字符串中的子字符串。
它返回0 (0)：

```
SELECT $FIND('AABBCCDD','AC') AS SubPoint
```

0

在下面的例子中，\$FIND从第7个字符开始搜索。
下面的例子返回14，也就是下一个出现“R”的字符的位置：

```
SELECT $FIND('EVERGREEN FOREST','R',7) AS SubPoint
```

14

在下面的例子中，\$FIND在字符串的最后一个字符之后开始搜索。
它返回0 (0):

```
SELECT $FIND('ABCDEFGF','G',10) AS SubPoint
```

0

下面的嵌入式SQL示例显示，小于1的开始将被视为1:

```
ClassMethod Find()
{
    s a="ABCDEFGF"
    s b="F"
    &sql(SELECT
    $FIND(:a,:b),
    $FIND(:a,:b,1),
    $FIND(:a,:b,0),
    $FIND(:a,:b,-35)
    INTO :a1,:a2,:a3,:a4)
    if SQLCODE'=0 {
        w !,"Error code ",SQLCODE }
    else {
        w !,"The input string: ",a
        w !,"Two-arg: ",a1
        w !,"3rd arg 1: ",a2
        w !,"3rd arg 0: ",a3
        w !,"3rd arg negative: ",a4 }
}
```

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

```
The input string: ABCDEFG
Two-arg: 7
3rd arg 1: 7
3rd arg 0: 7
3rd arg negative: 7
```

下面的嵌入式SQL示例使用\$FIND来搜索包含pi，\$CHAR(960)的Unicode字符的字符串。

第一个\$FIND返回pi后面的字符5。

第二个\$FIND也返回5;

它从字符4开始搜索，也就是圆周率，也就是搜索的字符。

第三个\$FIND从字符5开始搜索;

它返回13，这是pi下一个出现的位置。

注意，返回位置13，即使位置12是字符串中的最后一个字符:

```
ClassMethod Find1()
{
    s a="QT "$CHAR(960)" HONEY "$CHAR(960)
    s b=$CHAR(960)
```

```
&sql(SELECT
$FIND(:a,:b),
$FIND(:a,:b,4),
$FIND(:a,:b,5)
INTO :a1,:a2,:a3)
if SQLCODE'=0 {
    w !,"Error code ",SQLCODE }
else {
    w !,"The input string: ",a
    w !,"From beginning: ",a1
    w !,"From position 4: ",a2
    w !,"From position 5: ",a3
}
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLCommand).Find1()
```

```
The input string: QT ? HONEY ?
From beginning: 5
From position 4: 5
From position 5: 13
```

[#SQL #Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%94%E5%8D%81%E4%B8%83%E7%AB%A0-sql%E5%87%BD%E6%95%B0-find>