

文章

[姚鑫](#) · 一月 18 阅读大约需分钟

## 第二十章 SQL函数 CHARINDEX

## 第二十章 SQL函数 CHARINDEX

一个字符串函数，它返回子字符串在字符串中的位置，具有可选的搜索起点。

### 大纲

`CHARINDEX(substring, string[, start])`

### 参数

- `substring` - 要在字符串中匹配的子字符串。
- `string` - 作为子字符串搜索目标的字符串表达式。
- `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、NULL或非数值被视为0。

`CHARINDEX`返回`INTEGER`数据类型。

### 描述

`CHARINDEX`在字符串中搜索子字符串。  
如果找到匹配，则返回一个匹配子字符串的起始位置，从1开始计算。  
如果不能找到子字符串，`CHARINDEX`返回0。

空字符串是一个字符串值。  
因此，可以将空字符串用于任意一个字符串参数值。  
`start`参数将空字符串值视为0。  
但是，请注意ObjectScript空字符串作为NULL传递给SQL。

NULL不是SQL中的字符串值。  
因此，为任何一个`CHARINDEX`字符串参数指定NULL都会返回NULL。

`CHARINDEX`不能将`%Stream.GlobalCharacter`字段用于字符串或子字符串参数。尝试这样做会生成`SQLCODE-37`错误。可以使用`SUBSTRING`函数获取`%Stream.GlobalCharacter`字段并返回`%String`数据类型值以供`CHARINDEX`使用。

`CHARINDEX`区分大小写。使用其中一个大小写转换函数查找字母或字符串的大写和小写实例。

此函数提供了与Transact-SQL实现的兼容性。

## CHARINDEX, POSITION, \$FIND和INSTR

CHARINDEX, POSITION, \$FIND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。  
CHARINDEX、POSITION和INSTR返回匹配子字符串第一个字符的整数位置。  
\$FIND返回匹配子字符串结束后第一个字符的整数位置。  
CHARINDEX、\$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
```

## 示例

下面的示例在一个核苷酸序列中搜索子字符串TTAGGG首次出现。  
它返回7,这个子字符串在字符串中的位置:

```
SELECT CHARINDEX('TTAGGG','TTAGTCTTAGGGACATTAGGG')
```

7

下面的示例搜索所有包含子字符串“Fred”的Name字段值:

```
SELECT Name  
FROM Sample.Person  
WHERE CHARINDEX('Fred',Name)>0
```

以下示例使用SUBSTRING允许CHARINDEX搜索包含DNA核苷酸序列的%Stream.GlobalCharacter字段的前1000个字符,以查找子字符串TTAGGG的第一个匹配项:

```
SELECT CHARINDEX('TTAGGG',SUBSTRING(DNASeq,1,1000)) FROM Sample.DNASequences
```

以下示例匹配前10个字符后的子字符串:

```
SELECT CHARINDEX('Re','Reduce, Reuse, Recycle',10)
```

16

下面的示例指定超出字符串长度的起始位置:

```
SELECT CHARINDEX('Re','Reduce, Reuse, Recycle',99)
```

0

以下示例显示CHARINDEX处理空字符串('')的方式与处理任何其他字符串值一样:

```
SELECT CHARINDEX('', 'Fred Astore'),  
       CHARINDEX('A', ''),  
       CHARINDEX('', '')
```

在上面的示例中,第一个和第二个CHARINDEX函数返回0(不匹配),第三个函数返回1,因为空字符串与位置1处的空字符串匹配。

以下示例显示CHARINDEX不将NULL视为字符串值。为任一字符串指定NULL始终返回NULL:

```
SELECT CHARINDEX(NULL, 'Fred Astore'),  
       CHARINDEX('A', NULL),  
       CHARINDEX(NULL, NULL)
```

[#SQL #Caché](#)

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