

文章

[姚鑫](#) · 十二月 18, 2021 阅读大约需分钟

## 第十九章 SQL谓词 %STARTSWITH(一)

### 第十九章 SQL谓词 %STARTSWITH(一)

用指定初始字符的子字符串匹配值。

## 大纲

scalar-expression %STARTSWITH substring

## 参数

- scalar-expression - 将其值与子字符串进行比较的标量表达式(最常见的是数据列)。
- substring - 解析为包含与标量表达式中的值匹配的`n`个字符的字符串或数字的表达式。

## 描述

%STARTSWITH谓词允许选择以子字符串中指定的字符开头的数据值。  
如果substring不匹配任何标量表达式值,%STARTSWITH返回空字符串。  
无论显示模式如何,这个匹配总是在逻辑(内部存储)数据值上执行。

下面的示例选择所有以“M”开头的名称:

```
SELECT Name FROM Sample.MyTest WHERE Name %STARTSWITH 'M'
```

可以用NOT来颠倒谓词的意思。

下面的示例选择除了以“M”开头的名称以外的所有名称:

```
SELECT Name FROM Sample.MyTest WHERE NOT Name %STARTSWITH 'M'
```

## 排序类型

%STARTSWITH使用与它匹配的字段相同的排序规则类型。  
默认情况下,字符串数据类型字段是用SQLUPPER排序规则定义的,它不区分大小写。

在下面的例子中,UpName被定义为SQLUPPER;  
子字符串匹配不区分大小写:

```
SELECT UpName FROM Sample.MyTest WHERE UpName %STARTSWITH 'mo'
```

如果为WHERE子句中的列分配不同的排序规则类型,则该排序规则类型将匹配%STARTSWITH子字符串或字值

在面的例子中,UpName被定义为SQLUPPER;  
但是子字符串匹配是EXACT(区分大小写):

```
SELECT UpName FROM Sample.MyTest WHERE %EXACT(UpName) %STARTSWITH 'mo'
```

有些排序规则函数向字段值追加一个空格字符。  
这可能导致%STARTSWITH不匹配任何值,除非对子字符串应用等效的排序函数。

在例中,ExactName被定义为EXACT;  
因为查询将%SQLUPPER应用于标量表达式,所以比较现在涉及一个以附加空格字符开头的字符串。  
这个比较不会返回任何字段:

```
SELECT ExactName FROM Sample.MyTest WHERE %SQLUPPER(ExactName) %STARTSWITH 'Ra'
```

因此,还必须向子字符串追加一个空格字符。  
面的示例对EXACT字段应用非区分大小写的匹配:

```
SELECT ExactName FROM Sample.MyTest WHERE %SQLUPPER(ExactName) %STARTSWITH %SQLUPPER('Ra')
```

## %SelectMode

%STARTSWITH谓词不能使用当前的%SelectMode设置。  
子字符串必须以逻辑格式指定,无论%SelectMode设置如何。  
在ODBC或Display格式中指定谓词值通常会导致没有数据匹配或意外的数据匹配。  
这主要适用于日期、时间和IRIS格式列表(%List)。

在面的动态SQL示例中,%STARTSWITH谓词必须以逻辑格式指定日期子字符串,而不是%SelectMode=1(ODBC)格式。  
从41开始的逻辑值(日期从1953年4月4日(\$HOROLOG 41000)到1955年12月28日(\$HOROLOG 41999))被选中:

```
ClassMethod StartsWith()  
{  
    s q1 = "SELECT Name,DOB FROM Sample.Person "  
    s q2 = "WHERE DOB %STARTSWITH '41%'"  
    s myquery = q1_q2  
    s tStatement = ##class(%SQL.Statement).%New()  
    s tStatement.%SelectMode=1  
    s qStatus = tStatement.%Prepare(myquery)  
    if qStatus'=1 {  
        w "%Prepare failed:"  
        d $System.Status.DisplayError(qStatus)  
        q  
    }  
    s rset = tStatement.%Execute()  
    d rset.%Display()  
}
```

```
w !, "End of data"
}
```

## 列表字段

如果标量表达式是一个列表字段, %STARTSWITH可以使用%EXTERNAL来比较列表值和子字符串。  
例如, 要确定FavoriteColors列表字段以'B'开头的记录:

```
SELECT Name, FavoriteColors FROM Sample.Person
WHERE %EXTERNAL(FavoriteColors) %STARTSWITH 'B'
```

当%EXTERNAL将列表转换为DISPLAY格式时, 显示的列表项似乎用空格分隔。  
这个“空格”实际上是两个非显示字符CHAR(13)和CHAR(10)。  
要在列表中, 个元素中使用%STARTSWITH, 必须指定以字符:

```
SELECT Name, FavoriteColors FROM Sample.Person
WHERE %EXTERNAL(FavoriteColors) %STARTSWITH 'Orange' || CHAR(13) || CHAR(10) || 'B'
```

## 过滤null

- 如果标量表达式是任何非空数据值且子字符串是“空”值, %STARTSWITH总是返回标量表达式。
- 如果标量表达式为空且子字符串为“空”值, %STARTSWITH不返回标量表达式。

“空”子字符串值可以是任何一种: NULL, CHAR(0), 空字符串(""), 仅由空格(" ")组成的字符串, CHAR(32)空格字符, CHAR(9)制表符。  
默认情况下, %STARTSWITH使用所有这些值来过滤空值。

要返回仅由空格字符组成的标量表达式值, 必须使用%EXACT排序规则。

在以下所有示例中, %STARTSWITH返回相同的结果。  
它将结果限制为非空的FavoriteColors值:

```
SELECT Name, FavoriteColors FROM Sample.Person
WHERE FavoriteColors %STARTSWITH NULL
```

```
SELECT Name, FavoriteColors FROM Sample.Person
WHERE FavoriteColors %STARTSWITH ''
```

```
SELECT Name, FavoriteColors FROM Sample.Person
WHERE FavoriteColors %STARTSWITH ' '
```

```
SELECT Name, FavoriteColors FROM Sample.Person
WHERE FavoriteColors %STARTSWITH CHAR(9)
```

注意, 当从列表字段过滤空值时, %EXTERNAL排序规则类型不用于标量表达式。

由于NULL和空字符串的定义, %STARTSWITH NULL和空字符串的行为与复合子字符串不同。  
当将一个值与NULL连接时, 结果是NULL。

当将一个值与空字符串连接时，结果是该值。  
下面的例子说明了这一点：

```
SELECT Name, FavoriteColors
FROM Sample.Person
WHERE %EXTERNAL(FavoriteColors) %STARTSWITH 'B' || NULL
/* Selects all non-null rows */
```

```
SELECT Name, FavoriteColors
FROM Sample.Person
WHERE %EXTERNAL(FavoriteColors) %STARTSWITH 'B' || ''
/* Selects all values that begin with B */
```

### [#SQL #Caché](#)

源 URL: <https://cn.community.intersystems.com/post/%E7%AC%AC%E5%8D%81%E4%B9%9D%E7%AB%A0-sql%E8%B0%93%E8%AF%8D-startswith%E7%BC%88%E4%B8%80%E7%BC%89>