文章
姚 鑫 · 十二月 4, 2021 阅读大约需 4 分钟

第五章 SQL谓词 BETWEEN

第五章 SQL谓词 BETWEEN

大纲

```
scalar-expression BETWEEN lowval AND highval
```

参数

- scalar-expression -
  一种标量表达式(最常见的是数据列)，将其值与低值和高值(包括高值)之间的值范围进行比较。
- lowval - 解析为低排序规则序列值的表达式，指定与标量表达式中的每个值匹配的值范围的开始。
- highval - 解析为高排序规则序列值的表达式，指定要与标量表达式中的每个值匹配的值范围的末尾。

描述

BETWEEN谓词允许选择lowval和highval指定范围内的数据值。
这个范围包括低值和高值本身。
这等价于一对大于或等于操作符和一对小于或等于操作符。
下面的例子展示了这种比较:

```
SELECT Name,Age FROM Sample.Person
WHERE Age BETWEEN 18 AND 21
ORDER BY Age
```

这将返回Sample中的所有记录。
年龄值介于18到21之间的人员表，包括这些值。
注意，必须按升序指定BETWEEN值;
例如BETWEEN 21 AND 18这样的谓词将返回空字符串。
如果标量表达式的值都不在指定的范围内，则BETWEEN返回空字符串。

与大多数谓词一样，BETWEEN可以使用NOT逻辑运算符进行反转。
BETWEEN和NOT BETWEEN都不能用于返回NULL字段。
返回NULL字段使用IS NULL。
NOT BETWEEN的示例如下:

```
SELECT Name,Age FROM Sample.Person
WHERE Age NOT BETWEEN 20 AND 55
ORDER BY Age
```

这将返回Sample中的所有记录。
年龄值小于20或大于55的人表，不包括这些值。

排序类型

BETWEEN通常用于按数字顺序排序的数值范围。
但是，BETWEEN可用于任何数据类型值的排序规则序列范围。

BETWEEN使用与它所匹配的列相同的排序规则类型。
默认情况下，字符串数据类型排序为SQLUPPER，这是不区分大小写的。

如果查询为列分配了不同的排序规则类型，则还必须将此排序规则类型应用于BETWEEN子字符串。
下面的例子说明了这一点:

在下面的示例中，BETWEEN使用字段的默认字母大小写排序规则SQLUPPER，它不区分大小写。
它返回Name的字母顺序比HomeState高，HomeState的字母顺序比HomeCity高的记录:

```
SELECT Name,Home_State,Home_City
FROM Sample.Person
WHERE Home_State BETWEEN Name AND Home_City
ORDER BY Home_State
```

在下例中，BETWEEN字符串比较不区分大小写，因为HomeState字段被定义为SQLUPPER。
这意味着低val和高val在功能上是相同的，在任何字母中选择'MA':

```
SELECT Name,Home_State FROM Sample.Person
WHERE Home_State
   BETWEEN 'MA' AND 'Ma'
ORDER BY Home_State
```

在下面的示例中，%SQLSTRING排序函数使BETWEEN字符串比较区分大小写。
它选择那些HomeState值为'MA'到'MA'的记录，在这个数据集中包括'MA'， 'MD'， 'ME'， 'MO'， 'MS'和'MT':

```
SELECT Name,Home_State FROM Sample.Person
WHERE %SQLSTRING(Home_State)
   BETWEEN %SQLSTRING('MA') AND %SQLSTRING('Ma')
ORDER BY Home_State
```

在以下示例中，BETWEEN字符串比较不区分大小写，并且忽略空格和标点符号:

```
SELECT Name FROM Sample.Person
WHERE %STRING(Name) BETWEEN %SQLSTRING('OA') AND %SQLSTRING('OZ')
ORDER BY Name
```

下面的示例显示了在内部连接操作ON子句中使用的BETWEEN。
它正在执行一个不区分大小写的字符串比较:

```
SELECT P.Name AS PersonName,E.Name AS EmpName
FROM Sample.Person AS P INNER JOIN Sample.Employee AS E
ON P.Name BETWEEN 'an' AND 'ch' AND P.Name=E.Name
```

%SelectMode

如果%SelectMode设置为逻辑格式以外的值，那么BETWEEN谓词值必须以%SelectMode格式(ODBC或Display)指定。
这主要适用于日期、时间和 IRIS格式列表(%List)。
以逻辑格式指定谓词值通常会导致SQLCODE错误。
例如，SQLCODE -146 “ 无法将日期输入转换为有效的逻辑日期值 ” 。

在下面的动态SQL示例中，BETWEEN谓词必须以%SelectMode=1 (ODBC)的格式指定日期:

```
ClassMethod Between()
{
    s q1 = "SELECT Name,DOB FROM Sample.Person "
    s q2 = "WHERE DOB BETWEEN '1950-01-01' AND '1960-01-01'"
    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"
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLCommand).Between()
Name    DOB
Houseman,Martin D.      1955-09-25
Ingrahm,Yan S.  1954-06-15
Smith,Elvis Y.  1955-06-29
Gore,Alfred M.  1958-09-15
Yoders,Liza U.  1959-06-05
Ng,Liza Z.      1955-10-05
Yeats,Debby G.  1951-12-06
Zweifelhofer,Zelda J.   1954-02-19
Solomon,Emily D.        1953-01-28
Isaacs,Elvis V. 1952-04-05
Pantaleo,Robert U.      1950-03-29
Zampitello,Josephine Q. 1953-08-14
Xiang,Molly F.  1953-03-21
Nichols,Heloisa M.      1957-07-19
Hertz,Uma C.    1954-07-25
LaRocca,David X.        1956-01-11
Houseman,Alice R.       1957-12-07
Alton,Phil T.   1953-02-25
Davis,Jane E.   1953-07-28
Vanzetti,Alexandra O.   1953-12-29
Uhles,Dmitry P. 1951-08-23
Jafari,Christine Z.     1950-04-11

22 Rows(s) Affected
```

```
End of data
```

#SQL #Caché

源
URL:
https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%94%E7%AB%A0-sql%E8%B0%93%E8%AF%8D-between