文章
姚 鑫 · 十二月 9, 2021 阅读大约需 分钟

第十章 SQL谓词 IN

第十章 SQL谓词 IN

将值匹配到以逗号分隔的非结构化列表中的项。

大纲

```
scalar-expression IN (item1,item2[,...])

scalar-expression IN (subquery)
```

参数

- scalar-expression -
  标量表达式(最常见的是数据列)，将其值与以逗号分隔的值列表或子查询生成的结果集进行比较。
- item - 一个或多个文本值、输入主机变量或解析为文本值的表达式。
  以任何顺序列出，以逗号分隔。
- subquery - 一个用括号括起来的子查询，它从单个列返回一个结果集，用于与标量表达式进行比较。

描述

IN谓词用于将值匹配到非结构化的项系列。
通常，它将列数据值与以逗号分隔的值列表进行比较。
IN可以执行相等比较和子查询比较。

与大多数谓词一样，可以使用NOT逻辑操作符反转IN。
IN和NOT IN都不能用于返回空字段。
返回NULL字段使用IS NULL。

可以在任何可以指定谓词条件的地方使用IN，如本手册的谓词概述页面所述。

相等测试

IN谓词可以用作多个相等比较的简写，这些比较用OR操作符连接在一起。
例如:

```
SELECT Name, Home_State FROM Sample.Person
WHERE Home_State IN ('ME','NH','VT','MA','RI','CT')
```

如果Home_State等于逗号分隔列表中的任何值，则计算为true。

列出的项可以是常量或表达式。

IN比较使用为标量表达式定义的排序规则类型，而不考虑单个项的排序规则类型。
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。

下面两个示例说明排序规则匹配是基标量表达式排序规则的。
Home_State字段是用SQLUPPER(不区分大小写)排序规则定义的。
因此，下面的示例返回NH Home_State值:

```
SELECT Name, Home_State FROM Sample.Person
WHERE Home_State IN ('ME','nH','VT')
```

下面的示例不返回NH Home_State值:

```
SELECT Name, Home_State FROM Sample.Person
WHERE %EXACT(Home_State) IN ('ME','nH','VT')
```

在值列表中包含NULL没有意义。
NULL表示没有值，因此无法通过所有相等测试。
指定IN谓词(或任何其他谓词)将消除指定字段的任何NULL实例。
这在以下不正确(但可执行)的示例中显示:

```
SELECT FavoriteColors FROM Sample.Person
WHERE FavoriteColors IN ($LISTBUILD('Red'),$LISTBUILD('Blue'),NULL)
  /* NULL here is meaningless. No FavoriteColor NULL fields returned */
```

在谓词结果集中包含NULL字段的唯一方法是指定is NULL谓词，如下例所示:

```
SELECT FavoriteColors FROM Sample.Person
WHERE FavoriteColors IN ($LISTBUILD('Red'),$LISTBUILD('Blue')) OR FavoriteColors IS N
ULL
```

当使用日期或时间进行IN谓词相等性比较时，将自动执行适当的数据类型转换。
如果WHERE字段类型为TimeStamp，则Date或Time类型的值将转换为TimeStamp。
如果WHERE字段类型为Date，则类型为TimeStamp或String的值将转换为Date。
如果WHERE字段为type Time，则类型为TimeStamp或String的值将转换为Time。

下面的示例执行相同的相等比较并返回相同的数据。
DOB字段的数据类型为Date:

```
SELECT Name,DOB FROM Sample.Person
WHERE DOB IN ({d '1951-02-02'},{d '1987-02-28'})
```

```
SELECT Name,DOB FROM Sample.Person
WHERE DOB IN ({ts '1951-02-02 02:37:00'},{ts '1987-02-28 16:58:10'})
```

%SelectMode

如果%SelectMode设置为逻辑格式的值，那么IN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"。

在以下动态SQL示例中，In谓词必须以%SelectMode=1 (ODBC)格式指定日期:

```
/// d ##class(PHA.TEST.SQLCommand).In()
ClassMethod In()
{
    s q1 = "SELECT Name,DOB FROM Sample.Person "
    s q2 = "WHERE DOB IN('1956-03-05','1956-04-08','1956-04-18','1990-04-25')"
    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"
}
```

子查询比较

可以在子查询中使用IN谓词来测试列值(或任何其他表达式)是否等于任何子查询行值。
例如:

```
SELECT Name,Home_State FROM Sample.Person
WHERE Name IN
   (SELECT Name FROM Sample.Employee
    HAVING Salary < 50000)
```

注 意，子查询在SELECT列表中必须只有一个选择项。

下面的例子使用一个IN子查询返回不是Vendor状态的 Employee状态:

```
SELECT Home_State
FROM Sample.Employee
WHERE Home_State NOT IN (SELECT Address_State FROM Sample.Vendor)
GROUP BY Home_State
```

下面的示例将排序规则函数表达式匹配到带有子查询的IN谓词:

```
SELECT Name,Id FROM Sample.Person
WHERE %EXACT(Spouse) NOT IN
   (SELECT Id FROM Sample.Person
    WHERE Age < 65)
```

IN不能同时指定子查询和逗号分隔的文字值列表。

文字替换覆盖

在编译预解析期间，可以用圆括号将每个IN谓词参数括起来，从而覆盖文字替换。
例如,WHERE Home_State IN (('ME'),('NH'),('VT'),('MA'),('RI'),('CT'))。
这可以通过改善整体选择性和/或减少标边界选择性来提高查询性能。
但是，当使用不同的值多次调用同一个查询时，应该避免使用这种方法，因为这将导致为每个查询调用创建一个单独的缓存查询。

IN and %INLIST

IN和%INLIST谓词都可以用于提供多个值来进行OR相等比较。
%INLIST谓词用于将值匹配到%List结构的元素。
在动态SQL中，可以将%INLIST谓词值作为单个主机变量提供。
必须将IN谓词值作为单独的主机变量提供。
因此，更改IN谓词值的数量将导致创建一个单独的缓存查询。
%INLIST接受一个谓词值，一个包含多个元素的%List;
更改%List元素的数量不会导致创建一个单独的缓存查询。
%INLIST还提供了一个数量级的SIZE参数，SQL使用它来优化性能
由于这些原因，使用它通常是有利的

```
%INLIST($LISTFROMSTRING(val)) rather than IN(val1,val2,val3,..valn).
```

%INLIST可以执行相等比较;
它不能执行子查询比较。

#SQL #Caché

源 URL: https://cn.community.intersystems.com/post/%E7%AC%AC%E5%8D%81%E7%AB%A0-sql%E8%B0%93%E8%AF%8D