

文章

姚鑫 · 十二月 10, 2021 阅读大约需分钟

第十一章 SQL谓词 %INLIST

第十一章 SQL谓词 %INLIST

将一个值匹配到%List结构列表中的元素。

大纲

```
scalar-expression %INLIST list [SIZE ((nn))]
```

参数

- scalar-expression - 将其值与列表元素进行比较的标量表达式(最常见的是数据列)。
- list - 包含一个或几个元素的%List结构。
- SIZE ((nn)) - 可选-指定列表中元素数量估计值的整数。
必须指定为具有列值之一的字面值:10、100、1000、10000,等等。

描述

%INLIST谓词是 IRIS扩展,用于将字段的值与列表结构的元素匹配。

%INLIST和IN都允许对多个指定值执行这样的相等比较。

%INLIST将这些值指定为单个列表参数的元素。

因此,%INLIST允许改变要匹配值的数量,而无需建单独的缓存查询。

可选的%INLIST SIZE子句提供整数nn,它指定list中列表元素数量的数量级估计数。

IRIS使用这个数量级估计来确定最佳查询计划。

因为不管列表中元素的数量是少,都会使用相同的缓存查询,所以指定SIZE允许创建缓存查询,针对列表中预期的元素的大致数量进行优化。

更改SIZE字面值将创建一个单独的缓存查询。

指定nn为以文字之一:10、100、1000、10000,等等。

因为nn必须在编译时作为常量值可用,所以在所有SQL代码中必须将其指定为文字。

注意,必须为所有已编译SQL (Dynamic SQL)指定双括号。

双括号不用于嵌入式SQL。

%INLIST对list中的每个元素执行相等比较。

%INLIST比较使用为标量表达式定义的排序规则类型。

因此,列表元素的比较可能区分大小写,也可能不区分大小写,这取决于标量表达式的排序规则。

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。

指定NULL作为比较值是没有意义的。

NULL表示没有值,因此无法通过所有相等测试。

指定%INLIST谓词(或任何其他谓词)将消除指定字段的任何NULL实例。

必须指定IS NULL谓词,以便谓词结果集中包含带有NULL的字段。

与大多数谓词一样,%INLIST可以使用NOT逻辑操作符进行反转。

%INLIST和NOT %INLIST都不能返回空字段。
返回NULL字段使用IS NULL。

如果匹配表达式不是%List格式，%INLIST将生成个SQLCODE -400错误。
例如，如果集合属性sqlListType为DELIMITED，则列表字段的逻辑值不是%list格式。

要将值匹配到非结构项系列，如逗号分隔值列表，请使用IN谓词。
IN可以执行相等比较和子查询比较。

%SelectMode

%INLIST谓词不使用当前的%SelectMode设置。
列表的元素应该以逻辑格式指定，无论%SelectMode设置如何。
试图以ODBC格式或Display格式指定列表元素通常会导致没有数据匹配或意外的数据匹配。

可以使用%EXTERNAL或%ODBCOUT格式转换函数来转换谓词操的标量表达式字段。
这允许以Display格式或ODBC格式指定列表元素。
但是，使用格式转换函数会阻止对字段使用索引，因此会对性能产生重大影响。

在面的Dynamic

SQL示例中，%INLIST谓词指定一个包含1978年日期值元素列表，其格式为逻辑格式，而不是%SelectMode=1 (ODBC)格式。

与这些\$HOROLOG格式日期对应的日期被选中：

```
ClassMethod List()
{
    s bday = $lb(50039)
    for i = 50039 : 1 : 50403 {
        s bday = bday _ $lb(i)
    }
    s q1 = "SELECT Name,DOB FROM Sample.Person "
    s q2 = "WHERE DOB %INLIST ?"
    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(bday)
    d rset.%Display()
}
```

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

```
Name      DOB
??       1978-01-28
Chadwick,Zelda S.Chadwick,Zelda S.Chadwick,Zelda S.Chadwick,Zelda S.
          1978-01-28
Isaacs,Chad N.  1978-09-13
Ximines,Kim S.  1978-04-07
```

```
4 Rows(s) Affected
```

下面的动态SQL示例使用%ODBCOUT格式转换函数来转换谓词匹配的DOB字段。这允许以ODBC格式指定%INLIST列表元素。但是,指定格式转换函数会阻止对DOB字段值使用索引:

```
ClassMethod List1()
{
    s births = $LISTBUILD("1978-01-15", "1978-08-22", "1990-04-25")
    s q1 = "SELECT Name,DOB FROM Sample.Person "
    s q2 = "WHERE %ODBCOUT(DOB) %INLIST ?"
    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(births)
    d rset.%Display()
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLCommand).List1()
```

```
Name      DOB
yaoxin    1990-04-25
```

```
1 Rows(s) Affected
```

%INLIST and IN

%INLIST和IN谓词都可以用于提供多个值来进行相等比较。下面的动态SQL示例返回相同的结果:

```
ClassMethod List2()
{
    s states = $lb("VT","NH","ME")
    s myquery = "SELECT Name,Home_State FROM Sample.Person WHERE Home_State %INLIST ?"
    "
    s tStatement = ##class(%SQL.Statement).%New()
    s qStatus = tStatement.%Prepare(myquery)
    if qStatus'=1 {
        w "%Prepare failed:"
        d $System.Status.DisplayError(qStatus)
        q
    }
    s rset = tStatement.%Execute(states)
    d rset.%Display()
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLCommand).List2()
```

```
Name      Home_State
```

```

Lepon,Jeff Z.    NH
Ingleman,Terry A.    NH
Jung,Keith W.    NH
Xiang,Kirsten U.    ME
Jackson,Ralph V.    VT
Tesla,Geoffrey O.    NH
Tweed,Al O.    NH
Fives,Kristen F.    NH
Ingrahm,Susan N.    ME
Lepon,Janice T. ME
Wilson,Andrew O.    ME
Olsen,Ashley G. NH

```

12 Rows(s) Affected

```

ClassMethod List3()
{
    s s1="VT"
    s s2="NH"
    s s3="ME"
    s myquery = "SELECT Name,Home_State FROM Sample.Person WHERE Home_State IN(?,?,?)"
    "
    s tStatement = ##class(%SQL.Statement).%New()
    s qStatus = tStatement.%Prepare(myquery)
    if qStatus'=1 {
        w "%Prepare failed:"
        d $System.Status.DisplayError(qStatus)
        q
    }
    s rset = tStatement.%Execute(s1,s2,s3)
    d rset.%Display()
}

```

然而,在动态SQL中,可以提供%INLIST谓词值作为单个主机变量;
 必须将IN谓词值作为单个主机变量提供。
 因此,更改IN谓词值的数量将导致创建一个单级缓存查询。
 更改%INLIST谓词值的数量不会导致创建单级缓存查询。

示例

下面的示例将Home_State列值与新英格兰北部州结构列表的元素匹配:

```

ClassMethod List4()
{
    s states=$LISTBUILD("VT","NH","ME")
    s myquery="SELECT Name,Home_State FROM Sample.Person "_
        "WHERE Home_State %INLIST ?"
    s tStatement = ##class(%SQL.Statement).%New()
    s qStatus = tStatement.%Prepare(myquery)
    if qStatus'=1 {
        w "%Prepare failed:"
        d $System.Status.DisplayError(qStatus)
        q
    }
}

```

```

    s rset = tStatement.%Execute(states)
    d rset.%Display()
}

```

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排序规则定义的，它不区分大小写。
这些例子中的列表将新罕布什尔州指定为“nH”，而不是“NH”。
第一个示例返回NH Home_State值，第二个示例不返回NH Home_State值：

```

ClassMethod List5()
{
    s states=$LISTBUILD("VT","nH","ME")
    s myquery="SELECT Name,Home_State FROM Sample.Person "_
        "WHERE Home_State %INLIST ?"
    s tStatement = ##class(%SQL.Statement).%New()
    s qStatus = tStatement.%Prepare(myquery)
    if qStatus'=1 {
        w "%Prepare failed:"
        d $System.Status.DisplayError(qStatus)
        q
    }
    s rset = tStatement.%Execute(states)
    d rset.%Display()
}

```

```

ClassMethod List6()
{
    s states=$LISTBUILD("VT","nH","ME")
    s myquery="SELECT Name,Home_State FROM Sample.Person "_
        "WHERE %EXACT(Home_State) %INLIST ?"
    s tStatement = ##class(%SQL.Statement).%New()
    s qStatus = tStatement.%Prepare(myquery)
    if qStatus'=1 {
        w "%Prepare failed:"
        d $System.Status.DisplayError(qStatus)
        q
    }
    s rset = tStatement.%Execute(states)
    d rset.%Display()
}

```

下面的示例创建了一个SIZE为10的缓存查询。
对于该查询，指定SIZE 10是最优的，因为10对应于列表中的实际元素数量。
改变列表中的元素数量并不会创建一个单独的缓存查询。
改变SIZE字面值确实会创建一个单独的缓存查询：

```

ClassMethod List7()
{
    s states=$LISTBUILD("VT","NH","ME")
    s myquery="SELECT Name,Home_State FROM Sample.Person "_
        "WHERE Home_State %INLIST ? SIZE ((10))"
    s tStatement = ##class(%SQL.Statement).%New()
    s qStatus = tStatement.%Prepare(myquery)
    if qStatus'=1 {

```

```

        w "%Prepare failed:"
        d $System.Status.DisplayError(qStatus)
        q
    }
    s rset = tStatement.%Execute(states)
    d rset.%Display()
}

```

```

ClassMethod List8()
{
    s states=$LISTBUILD("VT","nH","ME")
    s myquery="SELECT Name,Home_State FROM Sample.Person "_
        "WHERE Home_State %INLIST ?"
    s tStatement = ##class(%SQL.Statement).%New()
    s qStatus = tStatement.%Prepare(myquery)
    if qStatus'=1 {
        w "%Prepare failed:"
        d $System.Status.DisplayError(qStatus)
        q
    }
    s rset = tStatement.%Execute(states)
    d rset.%Display()
}

```

下面的示例创建了一个SIZE为10的缓存查询。

对于该查询，指定SIZE 10是最优的，因为10对应于列表中的实际元素数量。

改变列表中的元素数量并不会创建一个单独的缓存查询。

改变SIZE字面值确实会创建一个单独的缓存查询：

```

ClassMethod List9()
{
    s states=$LISTBUILD("VT","NH","ME")
    s myquery="SELECT Name,Home_State FROM Sample.Person "_
        "WHERE Home_State %INLIST ? SIZE ((10))"
    s tStatement = ##class(%SQL.Statement).%New()
    s qStatus = tStatement.%Prepare(myquery)
    if qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QUIT}
    s rset = tStatement.%Execute(states)
    d rset.%Display()
}

```

[#SQL #Caché](#)

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