

文章

姚鑫 · 十一月 30, 2021 阅读大约需 1 分钟

## 第一章 SQL谓词的概述（一）

### 第一章 SQL谓词的概述（一）

描述计算结果为真或假的逻辑条件。

## 使用谓词

谓词是一个条件表达式，其计算结果为布尔值(true或false)。

谓词可以如下使用：

- 在SELECT语句的WHERE子句或HAVING子句中确定哪些行与特定查询相关。  
注意，不是所有谓词都可以在HAVING子句中使用。
- 在JOIN操作的ON子句中确定哪些行与连接操作相关。
- 在UPDATE或DELETE语句的WHERE子句中，确定要修改哪些行。
- WHERE CURRENT OF语句的AND子句中。
- 在CREATE TRIGGER语句的WHEN子句中确定何时应用触发操作代码。

## 谓词列表

每个谓词包含一个或多个比较操作符，可以是符号，也可以是关键字子句。

SQL支持以下比较操作符：

- = (equals) , <> (does not equal) , != (does not equal) , > (is greater than) , >= (is greater than or equal to) , < (is less than) , <= (is less than or equal to) - 比较条件。  
可用于数字比较或字符串排序顺序比较。  
对于数值比较，空字符串值("")被计算为0。  
在任何相等比较中，NULL总是返回空集；  
请使用IS NULL谓词。
- IS [NOT] NULL - 测试字段是否有未定义(NULL)值。
- IS [NOT] JSON - 测试一个值是JSON格式的字符串还是JSON数组或JSON对象的oref。
- EXISTS (subquery) - 使用子查询测试指定表是否存在一行或多行。
- BETWEEN x AND y - BETWEEN条件同时使用>=和<=比较条件。  
匹配必须在两个指定的范围限制值(包括)之间。
- IN (item1,item2[...itemn]) , IN (subquery) -  
一个等式条件，它将字段值与逗号分隔列表中的任何项或子查询返回的任何项匹配。
- %INLIST listfield - 将字段值与%List结构化列表中的任何元素匹配的相等条件。
- [ - 包含运算符。  
Match必须包含指定的字符串。  
Contains操作符使用EXACT排序规则，因此区分大小写。  
必须以逻辑格式指定值。
- ] - 跟随运算符。在排序规则序列中，匹配项必须出现在指定项之后。必须以逻辑格式指定值。
- %STARTSWITH string - 匹配必须以指定的字符串开始。
- FOR SOME - 布尔比较条件。对于指定字段的至少一个数据值，For Some条件必须为True。
- FOR SOME %ELEMENT - 带有%VALUE或%KEY谓词子句的列表元素比较条件。value必须与列表中至少

一个元素的值匹配。%key必须小于或等于列表中的元素数。%VALUE和%KEY子句可以使用任何其他比较运算符。

- LIKE - 使用文字和通配符的模式匹配条件。当希望返回包含已知子字符串的文字字符或包含已知序列中的多个已知子字符串的数据值时，请使用LIKE。LIKE使用其目标的排序规则进行字母大小写比较。（与CONTAINS运算符形成对比，后者使用精确排序规则。）
- %MATCHES - 使用文字、通配符以及列表和范围的模式匹配条件。如果希望返回的数据值包含已知子字符串的文字字符，或包含一个或多个落在可能字符列表或范围内的文字字符，或按已知序列包含多个这样的子字符串，请使用%Matches。%Matches使用精确排序规则进行字母大小写比较。
- %PATTERN - 使用字符类型的模式匹配条件。例如，'1U4L1',"A'(1个大写字母，4个小写字母，一个文字逗号，后跟任意数量的字母字符)。如果希望返回包含已知字符类型序列的数据值，请使用%Pattern。%Pattern可以指定已知的文字字符，但在数据值不重要但这些值的字符类型格式很重要时尤其有用。
- ALL, ANY, SOME - 一种量化的比较条件。
- %INSET, %FIND - 启用使用以编程方式指定的抽象临时文件或位图索引筛选RowId字段值的字段值比较条件。%Inset支持简单比较。%Find支持涉及位图索引的比较。

## NULL

NULL表示没有任何值。根据定义，它不能通过所有布尔测试：没有值等于NULL，没有值不等于NULL，没有值大于或小于NULL。即使NULL=NULL也不能作为谓词。因为IN谓词是一系列相等性测试，所以在IN值列表中指定NULL没有意义。因此，指定任何谓词条件都会消除该字段的任何为空的实例。在结果集中包含来自谓词条件的NULL字段的唯一方法是使用IS NULL谓词。下面的示例显示了这一点：

```
SELECT FavoriteColors FROM Sample.Person
WHERE FavoriteColors = $LISTBUILD('Red') OR FavoriteColors IS NULL
```

## 排序

谓词使用为字段定义的排序规则类型。

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。

如果在查询中指定排序规则类型，则必须在比较的两边指定它。

指定排序规则类型会影响索引的使用；

某些谓词比较可能涉及嵌入在字符串中的子字符串:Contains操作符(!)、%MATCHES谓词和%PATTERN谓词。

这些谓词总是使用EXACT排序法，因此总是区分大小写。

因为有些排序规则会在字符串中附加一个空格，所以如果这些谓词遵循字段的默认排序规则，它们就不能执行它们的功能。

但是，LIKE谓词可以使用通配符来匹配嵌入在字符串中的子字符串。

LIKE使用字段的默认排序规则，默认情况下不区分大小写。

## 复合谓词

谓词是条件表达式的最简单版本；

条件表达式可以由一个或多个谓词组成。

可以使用AND和OR逻辑操作符将多个谓词链接在一起。

通过将NOT一元操作符放在谓词之前，可以颠倒谓词的含义。

NOT一元操作符只影响紧随其后的谓词。

谓词严格按照从左到右的顺序计算。

可以使用括号对谓词进行分组。

可以在左括号前放置NOT一元操作符，以反转一组谓词的含义。

括号前后、括号与逻辑运算符之间不需要空格。

IN和%INLIST谓词在功能上相当于多个OR相等谓词。

下面的例子是等价的：

```
ClassMethod Predicates()  
{  
    s q1 = "SELECT Name,Home_State FROM Sample.Person "  
    s q2 = "WHERE Home_State='MA' OR Home_State='VT' OR Home_State='NH' "  
    s myquery = q1_q2  
    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()  
    d rset.%Display()  
}
```

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

Name	Home_State
Lepon,Jeff Z.	NH
Ingleman,Terry A.	NH
Jung,Keith W.	NH
Gallant,Thelma Q.	MA
Jackson,Ralph V.	VT
Tesla,Geoffrey O.	NH
Tweed,Al O.	NH
Fives,Kristen F.	NH
Olsen,Ashley G.	NH

9 Rows(s) Affected

```
ClassMethod Predicates1()  
{  
    s q1 = "SELECT Name,Home_State FROM Sample.Person "  
    s q2 = "WHERE Home_State IN('MA','VT','NH') "  
    s myquery = q1 _ q2  
    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()  
    d rset.%Display()  
}
```

```
ClassMethod Predicates2()  
{  
    s list = $LISTBUILD("MA","VT","NH")  
    s q1 = "SELECT Name,Home_State FROM Sample.Person "  
    s q2 = "WHERE Home_State %INLIST(?)"  
    s myquery = q1 _ q2  
    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(list)
d rset.%Display()
}
```

FOR SOME %ELEMENT谓词可以包含逻辑操作符，也可以使用逻辑操作符链接到其他谓词。下面的例子显示了这一点：

```
SELECT Name,FavoriteColors FROM Sample.Person
WHERE FOR SOME %ELEMENT(FavoriteColors)(%VALUE='Red' OR %Value='White'
    OR %Value %STARTSWITH 'B')
    AND (Name BETWEEN 'A' AND 'F' OR Name %STARTSWITH 'S')
ORDER BY Name
```

注意括号(Name BETWEEN 'A' AND 'F' OR Name %STARTSWITH 'S');  
如果没有这些分组括号，FOR SOME %ELEMENT条件将不适用于Name %STARTSWITH 'S'。

## 使用OR的集合谓词

FOR SOME %ELEMENT 是一个集合谓词。  
该谓词与OR逻辑操作符的使用受到限制，如下所示。  
不能使用OR逻辑操作符将引用表字段的集合谓词与引用另一个表中的字段的谓词关联起来。  
例如，

```
WHERE FOR SOME %ELEMENT(t1.FavoriteColors) (%VALUE='purple')
OR t2.Age < 65
```

因为这个限制取决于优化器如何使用索引，所以SQL只能在向表添加索引时强制执行这个限制。  
强烈建议在所有查询中避免这种类型的逻辑。

[#SQL #Caché](#)

---

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E4%B8%80%E7%AB%A0-sql%E8%B0%93%E8%AF%8D%E7%9A%84%E6%A6%82%E8%BF%B0%EF%BC%88%E4%B8%80%EF%BC%89>