文章
姚 鑫 · 十一月 27, 2021 阅读大约需 4 分钟

第八十九章 SQL命令 WHERE（二）

第八十九章 SQL命令 WHERE（二）

相等比较谓词

下面是可用的相等比较谓词:

Predicate	Operation
=	Equals
<>	Does not equal
!=	Does not equal
>	Is greater than
<	Is less than
>=	Is greater than or equal to
<=	Is less than or equal to

例如:

```
SELECT Name, Age FROM Sample.Person
WHERE Age < 21
```

SQL根据排序规则(值的排序顺序)定义了比较操作。
如果两个值以完全相同的方式排序，则它们相等。
如果一个值排在第二个值之后，则该值大于另一个值。
字符串字段排序规则接受字段的默认排序规则。
IRIS默认排序规则不区分大小写。
因此，两个字符串字段值的比较或字符串字段值与字符串文字的比较(默认情况下)是不区分大小写的。
例如，如果HomeState字段值是两个字母的大写字符串:

Expression	Value
'MA' = HomeState	TRUE for values MA.
'ma' = HomeState	TRUE for values MA.
'VA' < HomeState	TRUE for values VT, WA, WI, WV, WY.
'ar' >= HomeState	TRUE for values AK, AL, AR.

然而，请注意，两个字面值字符串的比较是区分大小写的:WHERE 'ma'=' ma'总是FALSE。

BETWEEN谓词

BETWEEN比较操作符允许选择语法BETWEEN lowval和highval指定范围内的数据值。
这个范围包括lowval和highval本身的值。
这相当于一个成对的大于或等于运算符和一个小于或等于运算符。
这个比较如下面的例子所示:

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

这将返回Sample中的所有记录。
人表，年龄值介于18和21之间，包括这些值。
注意，必须按升序指定BETWEEN值;
像BETWEEN 21 AND 18这样的谓词将不返回任何记录。

与大多数谓词一样，BETWEEN可以使用NOT逻辑操作符进行倒装，如下例所示:

```
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Person表，不包括这些值。

BETWEEN通常用于一个数值范围，该范围按数字顺序排序。
但是，BETWEEN可以用于任何数据类型的值的排序序列范围。

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不区分大小写。

IN和%INLIST谓词

IN谓词用于将一个值匹配到非结构化的一系列项。
它的语法如下:

```
WHERE field IN (item1,item2[,...])
```

Collation应用于IN比较，就像它应用于相等测试一样。
IN使用字段的默认排序规则。
默认情况下，与字段字符串值的比较不区分大小写。

%INLIST谓词是IRIS扩展，用于将值匹配到 IRIS列表结构的元素。
它的语法如下:

```
WHERE item %INLIST listfield
```

%INLIST使用EXACT排序。
因此，默认情况下，%INLIST字符串比较是区分大小写的。

使用任何一个谓词，都可以执行相等比较和子查询比较。

Substring谓词

可以使用下面的方法来比较字段值和子字符串:

Predicate	Operation
%STARTSWITH	该值必须以指定的子字符串开始。
[	包含运算符。该值必须包含指定的子字符串。

%STARTSWITH谓词

IRIS %STARTSWITH比较操作符允许对字符串或数字的初始字符执行部分匹配。
下面的示例使用%STARTSWITH。
选择“ Name ”以“ S ”开头的记录:

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

与其他字符串字段比较一样，%STARTSWITH比较使用字段的默认排序规则。
默认情况下，字符串字段不区分大小写。
例如:

```
SELECT Name,Home_City,Home_State FROM Sample.Person
WHERE Home_City %STARTSWITH Home_State
```

包含运算符(()

Contains操作符是左括号符号:[。
它允许将子字符串(字符串或数字)匹配到字段值的任何部分。
比较总是区分大小写的。
下面的示例使用Contains操作符选择Name值中包含“ S ”的记录:

```
SELECT Name, Age FROM Sample.Person
WHERE Name [ 'S'
```

NULL 谓词

这将检测未定义的值。
可以检测所有空值，或所有非空值。
NULL谓词的语法如下:

```
WHERE field IS [NOT] NULL
```

NULL谓词条件是可以在WHERE子句中的流字段上使用的少数谓词之一。

EXISTS 谓词

它使用子查询来测试子查询是否计算为空集。

```
SELECT t1.disease FROM illness_tab t1 WHERE EXISTS
 (SELECT t2.disease FROM disease_registry t2
 WHERE t1.disease = t2.disease
 HAVING COUNT(t2.disease) > 100)
```

FOR SOME 谓词

WHERE子句的FOR SOME谓词可用于根据一个或多个字段值的条件测试确定是否返回任何记录。
该谓词的语法如下:

```
FOR SOME (table [AS t-alias]) (fieldcondition)
```

FOR SOME指定字段condition的值必须为true;
至少有一个字段值必须匹配指定的条件。
Table可以是单个表，也可以是逗号分隔的表列表，每个表可以有一个表别名。
Fieldcondition为指定表中的一个或多个字段指定一个或多个条件。
table参数和字段condition参数都必须用括号分隔。

下面的示例展示了如何使用FOR SOME谓词来确定是否返回结果集:

```
SELECT Name,Age AS AgeWithWorkers
FROM Sample.Person
WHERE FOR SOME (Sample.Person) (Age<65)
ORDER BY Age
```

在上面的示例中，如果至少有一个字段包含的Age值小于指定的Age，则返回所有记录。
否则，不返回任何记录。

FOR SOME %ELEMENT 谓词

WHERE子句的FOR SOME %ELEMENT谓词语法如下:

```
FOR SOME %ELEMENT(field) [AS e-alias] (predicate)
```

FOR SOME %ELEMENT谓词用指定的谓词子句值匹配字段中的元素。
SOME关键字指定字段中至少有一个元素必须满足指定的谓词条件。
谓词可以包含%VALUE或%KEY关键字。

FOR SOME %ELEMENT谓词是一个集合谓词。

LIKE, %MATCHES, and %PATTERN 谓词

这三个谓词允许执行模式匹配。
- LIKE允许使用文字和通配符进行模式匹配。
当希望返回包含已知字面值子字符串的数据值，或在已知序列中包含多个已知子字符串时，请使用LIKE。
LIKE使用目标的排序规则进行字母大小写比较。
- %MATCHES允许使用文字、通配符、列表和范围进行模式匹配。
当您希望返回包含已知字面值子字符串的数据值，或包含一个或多个位于可能字符列表或范围内的字面值字符，或在已知序列中包含多个这样的子字符串时，请使用%MATCHES。
%MATCHES使用EXACT排序法进行字母大小写比较。
- %PATTERN允许指定字符类型的模式。
例如,'1U4L1",".A'(1个大写字母，4个小写字母，一个逗号，后面跟着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重要时特别有用。

谓词和逻辑操作符

可以使用AND和OR逻辑操作符关联多个谓词。

可以使用括号对多个谓词进行分组。
由于IRIS使用已定义的索引和其他优化来优化WHERE子句的执行，因此无法预测and和OR逻辑运算符链接的谓词的求值顺序。
因此，指定多个谓词的顺序对性能几乎没有影响。
如果希望严格地从左到右计算谓词，可以使用CASE语句。

注意:不能使用OR逻辑运算符将引用表字段的FOR SOME %ELEMENT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%E5%85%AB%E5%8D%81%E4%B9%9D%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-where%EF%BC%88%E4%BA%8C%EF%BC%89