文章
姚 鑫 · 十一月 26, 2021 阅读大约需 2 分钟

第八十八章 SQL命令 WHERE（一）

第八十八章 SQL命令 WHERE（一）

指定一个或多个限制性条件的SELECT子句。

大纲

```
SELECT fields
FROM table
WHERE condition-expression
```

参数

- condition-expression - 由一个或多个布尔谓词组成的表达式，该谓词控制要检索的数据值。

描述

可选的WHERE子句可以用于以下目的:

- 指定限制要返回哪些数据值的谓词。
- 指定两个表之间的显式连接。
- 指定基表和另一个表中的字段之间的隐式连接。

WHERE子句最常用于指定一个或多个谓词，这些谓词用于限制SELECT查询或子查询检索到的数据(过滤出行)。
还可以在UPDATE命令、DELETE命令或INSERT(或INSERT or UPDATE)命令的结果集中使用WHERE子句。

WHERE子句限定或取消查询选择中的特定行。
符合条件的行是那些条件表达式为真的行。
条件表达式可以是一个或多个逻辑测试(谓词)。
多个谓词可以通过AND和OR逻辑操作符链接。

如果谓词包含除法，并且数据库中有任何值可以生成值为零或NULL的除法，则不能依赖求值顺序来避免被零除法。
相反，使用CASE语句来抑制风险。

WHERE子句可以指定包含子查询的条件表达式。子查询必须用圆括号括起来。

WHERE子句可以使用=(内部连接)符号连接操作符指定两个表之间的显式连接。

WHERE子句可以使用箭头语法(- >)操作符在基表和来自另一个表的字段之间指定隐式连接。

指定字段

WHERE子句最简单的形式是指定一个比较字段和值的谓词，例如WHERE Age > 21。
有效的字段值包括以下:列名(WHERE Age > 21);

%ID， %TABLENAME，或%CLASSNAME;
标量函数指定列名(WHERE ROUND(Age，-1)=60)，一个排序规则函数指定列名(WHERE %SQLUPPER(name) %STARTSWITH ' AB')。

不能按列号指定字段。

因为重新编译表时RowID字段的名称可能会改变，WHERE子句应该避免通过名称引用RowID(例如，WHERE ID=22)。
相反，使用%ID伪列名来引用RowID(例如，WHERE %ID=22)。

不能通过列别名指定字段;
尝试这样做会产生SQLCODE -29错误。
但是，可以使用子查询来定义列别名，然后在WHERE子句中使用该别名。
例如:

```
SELECT Interns FROM
      (SELECT Name AS Interns FROM Sample.Employee WHERE Age<21)
WHERE Interns %STARTSWITH 'A'
```

不能指定聚合字段;
尝试这样做将生成SQLCODE -19错误。
但是，可以通过使用子查询向WHERE子句提供聚合函数值。
例如:

```
SELECT Name,Age,AvgAge
FROM (SELECT Name,Age,AVG(Age) AS AvgAge FROM Sample.Person)
WHERE Age < AvgAge
ORDER BY Age
```

整型和字符串

如果将定义为整数数据类型的字段与数值进行比较，则在执行比较之前将数值转换为规范形式。
例如，WHERE Age=007.00解析为WHERE Age=7。
这种转换发生在所有模式中。

如果将定义为整数数据类型的字段与Display模式下的字符串值进行比较，则将该字符串解析为数值。
例如，与任何非数字字符串一样，空字符串(")被解析为数字0。
这种解析遵循将字符串处理为数字的ObjectScript规则。
例如，WHERE Age='twenty'解析为WHERE Age=0;WHERE Age='20something'解析为WHERE Age=20。
SQL只在Display模式下执行解析;
在逻辑或ODBC模式下，将整数与字符串值进行比较将返回null。

要比较字符串字段和包含单引号的字符串，请使用双引号。
例如，WHERE Name %STARTSWITH 'O'"，返回的是 O ’ Neil and O ’ Connor, 而不是 Obama.

日期和时间

SQL日期和时间使用逻辑模式内部表示进行比较和存储。
它们可以以逻辑模式、显示模式或ODBC模式返回。
例如，1944年9月28日表示为:逻辑模式37891，显示模式09/28/1944,ODBC模式194409-28。
在条件表达式中指定日期或时间时，可能由于SQL模式与日期或时间格式不匹配，或由于无效的日期或时间值而发生错误。

WHERE子句条件表达式必须使用与当前模式相对应的日期或时间格式。

例如，在逻辑模式下，要返回出生日期为2005年的记录，WHERE子句将出现如下:WHERE DOB BETWEEN 59901 AND 60265
当在显示模式下，同样的WHERE子句会出现如下:WHERE DOB BETWEEN '01/01/2005' AND '12/31/2005'

如果条件表达式的日期或时间格式与显示模式不匹配，将导致错误:
- 在显示模式或ODBC模式下，以不正确的格式指定日期数据将产生SQLCODE -146错误。
以不正确的格式指定时间数据将产生SQLCODE -147错误。
- 在逻辑模式下，以不正确的格式指定日期或时间数据不会产生错误，但要么不返回数据，要么返回非预期的数据。
这是因为逻辑模式不会将显示或ODBC格式的日期或时间解析为日期或时间值。
WHERE DOB BETWEEN 37500 AND 38000 AND DOB <> '1944-09-28'
'返回一系列DOB值，包括DOB=37891(1944年9月28日)，这是<>谓词试图忽略的。

无效的日期或时间值还会生成SQLCODE -146或-147错误。
无效日期是可以在显示模式/ODBC模式中指定的日期，但 IRIS不能转换为等效的逻辑模式。
例如，在ODBC模式下，以下命令会产生SQLCODE -146错误:WHERE DOB > '1830-01-01'，因为IRIS无法处理1840年12月31日之前的日期值。
以下在ODBC模式下也会产生SQLCODE -146错误:WHERE DOB BETWEEN '2005-01-01' AND '2005-02-29'，因为2005不是闰年。

在逻辑模式下，Display模式或ODBC模式值不会被解析为日期或时间值，因此不会对其值进行验证。
因此，在逻辑模式下，WHERE子句(例如WHERE DOB > '1830-01-01')不会返回错误。

流字段

在大多数情况下，不能在WHERE子句谓词中使用流字段。
这样做将导致SQLCODE -313错误。
但是，在WHERE子句中允许使用流字段:
- 流空测试:可以指定流字段IS null或流字段IS NOT null。
- 流长度测试:可以在WHERE子句谓词中指定CHARACTERLENGTH(流字段)、CHARLENGTH(流字段)或DATALENGTH(流字段)函数。
- 流子串测试:可以在WHERE子句谓词中指定substring (streamfield,start,length)函数。

List结构

IRIS支持列表结构数据类型%list(数据类型类%Library.List)。
这是一种压缩的二进制格式，并不映射到 SQL的相应本机数据类型。
它对应的数据类型为VARBINARY，默认MAXLEN为32749。
因此，动态SQL不能在WHERE子句比较中使用%List数据。

要引用结构化列表数据，请使用%INLIST谓词或FOR SOME %ELEMENT谓词。

要在条件表达式中使用列表字段的数据值，可以使用%EXTERNAL将列表值与谓词进行比较。
例如，要返回FavoriteColors列表字段值由单个元素'Red'组成的所有记录:

```
SELECT Name,FavoriteColors FROM Sample.Person
WHERE %EXTERNAL(FavoriteColors)='Red'
```

当%EXTERNAL将列表转换为DISPLAY格式时，显示的列表项似乎由一个空格分隔。
这个 “ 空格 ” 实际上是两个非显示字符CHAR(13)和CHAR(10)。
要对列表中的多个元素使用条件表达式，必须指定这些字符。
例如，要返回FavoriteColors列表字段值由两个元素'Orange'和'Black'(按顺序)组成的所有记录:

```
SELECT Name,FavoriteColors FROM Sample.Person
WHERE %EXTERNAL(FavoriteColors)='Orange'||CHAR(13)||CHAR(10)||'Black'
```

变量

WHERE子句谓词可以指定:

%TABLENAME或%CLASSNAME伪字段变量关键字。
%TABLENAME返回当前表名。
%CLASSNAME返回当前表对应的类名。
如果查询引用多个表，可以在关键字前加上表别名。
例如,t1.%TABLENAME。

一个或多个ObjectScript特殊变量(或它们的缩写):$HOROLOG， $JOB， $NAMESPACE， $TLEVEL， $USERNAME， $ZHOROLOG， $ZJOB， $ZNSPACE， $ZPI， $ZTIMESTAMP， $ZTIMEZONE， $ZVERSION。

谓词列表

SQL谓词可分为以下几类:

- Equality Comparison 谓词
- BETWEEN 谓词
- IN and %INLIST 谓词
- %STARTSWITH Predicate and Contains Operator
- NULL Predicate
- EXISTS Predicate
- FOR SOME Predicate
- FOR SOME %ELEMENT Predicate
- LIKE, %MATCHES, and %PATTERN Predicates
- %INSET and %FIND Predicates

谓词区分大小写

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。

%INLIST、Contains操作符([)、%MATCHES和%PATTERN谓词不使用字段的默认排序规则。
它们总是使用区分大小写的EXACT排序法。

两个字面值字符串的谓词比较总是区分大小写的。

谓词条件和%NOINDEX

可以使用%NOINDEX关键字作为谓词条件的前缀，以防止查询优化器在该条件上使用索引。
这在指定绝大多数行都满足的范围条件时非常有用。
例如，WHERE %NOINDEX Age >= 1。

离群值的谓词条件

如果动态SQL查询中的WHERE子句选择了一个非空的离群值，可以通过将离群值文字括在双括号中来显著提高性能。
这些双括号导致动态SQL在优化时使用离群值选择性。
例如，如果企业位于马萨诸塞州，那么很大一部分员工将居住在马萨诸塞州。
对于Employees表HomeState字段，'MA'是离群值。

要最优地选择这个值，应该指定WHERE HomeState=(('MA'))。

在嵌入式SQL或视图定义中不应使用此语法。
在嵌入式SQL或视图定义中，总是使用离群值选择，不需要特殊编码。

动态SQL查询中的WHERE子句会自动针对空离群值进行优化。
例如，WHERE FavoriteColors IS NULL这样的子句。
当NULL是离群值时，is NULL和is NOT NULL谓词不需要特殊编码。

离群值选择性由运行调优表实用程序决定。

#SQL #Caché

源URL:
https://cn.community.intersystems.com/post/%E7%AC%AC%E5%85%AB%E5%8D%81%E5%85%AB%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-where%EF%BC%88%E4%B8%80%EF%BC%89