

文章

姚鑫 · 三月 15, 2021 阅读大约需 6 分钟

第十章 SQL排序（二）

第十章 SQL排序（二）

查询排序

InterSystems SQL提供了排序规则功能，可用于更改字段的排序规则或显示。

第十章 SQL排序（二）

查询排序

InterSystems SQL提供了排序规则功能，可用于更改字段的排序规则或显示。

查询明细排序

将排序功能应用于查询选择项会更改该项目的显示。

- 字母大小写：默认情况下，查询显示带有大写和小写字母的字符串。例外情况是对排序规则类型SQLUPPER的字段进行DISTINCT或GROUP BY操作。这些操作以所有大写字母显示该字段。可以使用%EXACT排序功能来反转此字母大小写转换，并以大写和小写字母显示该字段。不应在选择项列表中使用%SQLUPPER排序规则函数以所有大写字母显示字段。
这是因为%SQLUPPER在字符串的长度上添加了一个空格字符。请改用UPPER函数：

```
SELECT TOP 5 Name,$LENGTH(Name) AS NLen,
              %SQLUPPER(Name) AS UpCollN,$LENGTH(%SQLUPPER(Name)) AS UpCollLen,
              UPPER(Name) AS UpN,$LENGTH(UPPER(Name)) AS UpLen
FROM Sample.Person
```

AR_CloseShiftReceipt

AR_CreditCardPayments

AR_OutstandPayment

AR_PatBillClaim

AR_PatBillDate

Views (61)

Procedures (4530)

80

SELECT TOP 5 Name,\$LENGTH(Name) AS NLen,

%SQLUPPER(Name) AS UpCollN,\$LENGTH(%SQLUPPER(Name)) AS UpCollLen,

UPPER(Name) AS UpN,\$LENGTH(UPPER(Name)) AS UpLen

FROM Sample.Person

Script 1

sql.sql*

	Name	NLen	UpCollN	UpCollLen	UpN	UpLen
	VARCHAR (50)	SMALLINT	VARCHAR (50)	SMALLINT	VARCHAR (50)	SMALLINT
1	Adams,Diane F.	14	ADAMS,DIANE F.	15	ADAMS,DIANE F.	14
2	Adams,Susan E.	14	ADAMS,SUSAN E.	15	ADAMS,SUSAN E.	14
3	Ahmed,Elmo X.	13	AHMED,ELMO X.	14	AHMED,ELMO X.	13
4	Alton,Martin S.	15	ALTON,MARTIN S.	16	ALTON,MARTIN S.	15
5	Alton,Phil T.	13	ALTON,PHIL T.	14	ALTON,PHIL T.	13

- 字符串截断：可以使用%TRUNCATE排序函数来限制显示的字符串数据的长度。
%TRUNCATE比%SQLUPPER更可取，后者会在字符串的长度上添加一个空格字符。

```
SELECT TOP 5 Name,$LENGTH(Name) AS NLen,
              %TRUNCATE(Name,8) AS TruncN,$LENGTH(%TRUNCATE(Name,8)) AS TruncLen
FROM Sample.Person
```

请注意，不能嵌套排序规则函数或大小写转换函数。

- WHERE子句比较：大多数WHERE子句谓词条件比较使用字段/属性的排序规则类型。因为字符串字段默认为SQLUPPER，所以这些比较通常不区分大小写。可以使用%EXACT排序规则功能使它们区分大小写：
- 下面的示例返回HomeCity字符串匹配项，无论字母大小写如何：

```
SELECT Home_City FROM Sample.Person WHERE Home_City = 'albany'
```

以下示例返回区分大小写的HomeCity字符串匹配：

```
SELECT Home_City FROM Sample.Person WHERE %EXACT(Home_City) = 'albany'
```

SQL Follows运算符 (]) 使用字段/属性归类类型。

但是，无论字段/属性的排序规则类型如何，SQL Contains运算符 ([) 都使用EXACT排序规则：

```
SELECT Home_City FROM Sample.Person WHERE Home_City [ 'c'  
ORDER BY Home_City
```

%MATCHES和%PATTERN谓词条件使用EXACT排序规则，而不管字段/属性的排序规则类型如何。
%PATTERN谓词提供区分大小写的通配符和不区分大小写的通配符 ('A')。

ORDER BY子句：ORDER BY子句使用名称空间默认排序规则对字符串值进行排序。因此，ORDER BY不会基于字母大小写进行排序。可以使用%EXACT排序规则根据字母大小写对字符串进行排序。

DISTINCT和GROUP BY排序规则

默认情况下，这些操作使用当前的名称空间排序。默认的名称空间排序规则是SQLUPPER。

- DISTINCT：DISTINCT关键字使用名称空间默认排序规则来消除重复值。因此，DISTINCT Name返回所有大写字母的值。可以使用EXACT排序规则返回大小写混合的值。
DISTINCT消除仅字母大小写不同的重复项。
要保留大小写不同的重复项，但要消除确切的重复项，请使用EXACT排序规则。
以下示例消除了精确的重复项（但不消除字母大写的变体），并以混合的大写和小写形式返回所有值：

```
SELECT DISTINCT %EXACT(Name) FROM Sample.Person
```

UNION涉及隐式DISTINCT操作。

- GROUP BY：GROUP BY子句使用名称空间默认排序规则来消除重复的值。因此，GROUP BY Name返回所有大写字母的值。**可以使用EXACT排序规则返回大小写混合的值。**GROUP BY消除仅字母大小写不同的重复项。若要保留大小写不同的重复项，但要消除完全相同的重复项，必须在GROUP BY子句（而不是select-item）上指定%EXACT归类函数。

下面的示例返回大小写混合的值；GROUP BY消除重复项，包括字母大小写不同的重复项：

```
SELECT %EXACT(Name) FROM Sample.Person GROUP BY Name
```

下面的示例返回大小写混合的值；GROUP BY消除了精确的重复项（但不消除字母大写的变体）：

```
SELECT Name FROM Sample.Person GROUP BY %EXACT(Name)
```

旧版排序类型

InterSystems SQL支持多种旧式排序规则类型。它们已被弃用，不建议与新代码一起使用，因为它们的目的是为遗留系统提供持续的支持。他们是：

- %ALPHAUP — 除去问号（“？”）和逗号（“，”）之外的所有标点符号，并将所有小写字母转换为大写字母。主要用于映射旧全局变量。由SQLUPPER代替。

- %STRING — 将逻辑值转换为大写，去除所有标点符号和空格（逗号除外），并在字符串的开头添加一个前导空格。它将所有仅包含空格（空格，制表符等）的值作为SQL空字符串进行整理。由SQLUPPER代替。

- %UPPER — 将所有小写字母转换为大写字母。主要用于映射旧全局变量。由SQLUPPER代替。

- SPACE —

SPACE排序将单个前导空格附加到一个值，强制将其作为字符串求值。要建立SPACE排序规则，CREATE TABLE提供一个SPACE排序规则关键字，而ObjectScript在%SYSTEM.Util类的Collation（）方法中提供一个SPACE选项。没有相应的SQL排序规则功能。

注意：如果使用EXACT，UPPER或ALPHAUP排序定义了字符串数据类型字段，并且查询在此字段上应用了%STARTSWITH条件，则可能导致不一致的行为。如果指定给%STARTSWITH的子字符串是规范数字（尤其是负数和/或小数），则%STARTSWITH可能会根据字段是否被索引而给出不同的结果。如果未对列进行索引，则%STARTSWITH应该会按预期执行。如果该列已建立索引，则可能会发生意外的结果。

SQL和NLS排序

上面描述的SQL排序规则不应与InterSystems IRIS NLS排序规则功能混淆，后者提供符合特定本国语言排序规则要求的下标级别编码。这是提供分页的两个独立系统，它们在产品的不同级别上工作。

InterSystems IRIS NLS排序可以具有当前过程的过程级别排序，并且可以具有特定全局变量的不同排序。

为了确保使用InterSystems SQL时的正常运行，要求进程级NLS排序规则与所涉及的所有全局变量的NLS排序规则完全匹配，包括表所使用的全局变量以及用于临时文件（例如进程专用全局变量和IRIS TEMP）的全局变量。否则，查询处理器设计的不同处理计划可能会得出不同的结果。在发生排序的情况下，例如ORDER BY子句或范围条件，查询处理器将选择最有效的排序策略。它可以使用索引，可以在进程专用的全局文件中使用临时文件，可以在本地数组中排序，也可以使用“]]”（之后排序）比较。所有这些都是下标类型的比较，遵循有效的InterSystems IRIS NLS归类，这就是为什么所有这些类型的全局变量都必须使用完全相同的NLS归类的原因。

系统使用数据库默认排序规则创建全局变量。可以使用%Library.GlobalEdit类的Create（）方法来创建具有不同排序规则的全局变量。唯一的要求是指定的归类可以是内置的（例如InterSystems IRIS标准），也可以是当前语言环境中可用的国家归类之一。

[#SQL #Caché #InterSystems IRIS #InterSystems IRIS for Health](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E5%8D%81%E7%AB%A0-sql%E6%8E%92%E5%B A%8F%E5%BC%88%E4%BA%8C%E5%BC%89>