

文章

[姚鑫](#) · 十一月 1, 2021 阅读大约需 8 分钟

第六十三章 SQL命令 ORDER BY (一)

第六十三章 SQL命令 ORDER BY (一)

指定结果集中行排序的SELECT子句。

大纲

```
ORDER BY ordering-item [ASC | DESC]{,ordering-item [ASC | DESC] ...}
```

参数

- ordering-item - 决定排序顺序的文字。
列名、列别名或列号。
ORDER BY子句可以包含单个排序项或以逗号分隔的排序项列表，以指定排序层次结构。
- ASC DESC - 可选-按升序(ASC)或降序(DESC)排序。
默认为升序。

描述

ORDER BY子句根据指定列的数据值或以逗号分隔的列序列对查询结果集中的记录进行排序。
该语句对单个结果集进行操作，这些结果集要么来自SELECT语句，要么来自多个SELECT语句的UNION。

ORDER BY按逻辑(内部存储)数据值对记录进行排序，而不考虑当前的选择模式设置。

ORDER BY子句是SELECT语句中的最后一个子句。
它出现在FROM、WHERE、GROUP BY和HAVING子句之后。
以错误的顺序指定SELECT子句将产生SQLCODE -25错误。

如果SELECT语句没有指定ORDER BY子句，则返回的记录顺序是不可预测的。

如果SELECT语句指定了ORDER BY和TOP子句，则作为“TOP”行的返回的记录将与ORDER BY子句中指定的顺序一致。
为例。

```
SELECT TOP 5 Name, Age FROM MyTable ORDER BY Age  
DESC
```

返回MyTable中年龄最大的5行数据，按从老到小的顺序排列。

在SELECT列表中执行窗口函数(包括窗口函数自己的ORDER BY子句)之后应用ORDER BY子句。
因此，窗口函数返回的值不受SELECT查询的ORDER by子句的影响。

限制

如果SELECT查询指定了ORDER BY子句，则生成的数据是不可更新的。
因此，如果指定后续的DECLARE CURSOR FOR UPDATE语句，则忽略FOR UPDATE子句，并将游标声明为只读。

如果ORDER BY应用于UNION，则排序项必须是一个数字或简单列名。
它不可能是一个表达式。
如果使用列名，它将引用在UNION的第一个SELECT列表中命名的结果列。

在子查询中使用ORDER BY子句时，必须与TOP子句配对。
这可能是TOP ALL子句。

例如，下面的FROM子句子查询无效:(SELECT DISTINCT age FROM table1 ORDER BY age);
但是，下面的FROM子句子查询是有效的:(SELECT DISTINCT TOP ALL age FROM table1 ORDER BY age)。

指定列排序

可以指定要排序的单个列，也可以指定多个列作为逗号分隔的列表。
排序由第一个列出的列完成，然后在列中由第二个列出的列完成，以此类推。

列可以通过列名、列别名或列号指定。

ORDER BY不限于字段值。

- 无论字段是否在SELECT列表中指定，都可以通过列名将字段指定为排序项。
- 可以将表达式指定为排序项，例如ORDER BY LENGTH(Name)。
- 可以将窗口函数指定为排序项，例如ORDER BY ROWNUMBER() OVER (PARTITION BY State)。
- 不能在ORDER BY子句中直接指定聚合函数;
尝试这样做会产生SQLCODE -73错误。

可以在ORDER BY子句中根据列别名或列号指定任何选择项，包括聚合函数、窗口函数或表达式。
如果在SELECT列表中指定列别名，则在指定聚合函数、窗口函数或表达式时，使用选择项列号(例如3)，而不是默认的列名(例如Aggregate3)。

ORDER BY子句可以指定列名、列别名和选择项列号的任意组合。如果ordering-item的第一个字符是数字，则IRIS假定指定的是列号。否则，假定使用列名或列别名。请注意，列名和列别名不区分大小写。

除了少数例外，订货项必须指定为字面量。

如果一个排序项不能被解析为有效的标识符(列名或列号)，或者不能被解析为无符号整数(列号)，那么该排序项将被忽略，ORDER BY执行将继续执行逗号分隔列表中的下一个排序项。

一些被忽略的订单项值的例子是动态SQL？

输入参数或嵌入式SQL:var主机变量、子查询、解析为数字、带符号的数字或括号中的数字的表达式。

列名

可以将列名指定为文字。

在某些情况下，对列名进行操作的表达式可以用作排序项。
不能使用将列名作为字符串提供的变量或其他表达式。

下面的ORDER BY子句按列名排序:

```
SELECT Name,Home_State,DOB
FROM Sample.Person
ORDER BY Home_State,Name
```

无论排序列是否在选择项列表中，都可以按列名排序。
(由于明显的原因，不能按列别名或列号排序，除非排序列位于选择项列表中。)
下面的示例以与上一个示例相同的顺序返回相同的记录:

```
SELECT Name,DOB
FROM Sample.Person
ORDER BY Home_State,Name
```

如果order -item不是指定表中现有的列名(或列别名),则会发出SQLCODE -29错误。

即使RowID是私有的并且没有在选择项列表中列出,也可以按RowID值排序。

应该指定%ID伪列名作为ordering-item,而不是实际的RowID字段名。

如果查询包含TOP子句,则按RowID排序更改TOP子句选择的行。

例如,如果一个表有100行(RowIDs是连续的),SELECT TOP 5% ID FROM table ORDER BY %ID返回RowIDs 1, 2, 3, 4, 5;

SELECT TOP 5 %ID FROM Table ORDER BY %ID DESC返回RowIDs 100, 99, 98, 97, 96。

ORDER BY子句可以指定表名或表别名作为ordering-item的一部分:

```
SELECT P.Name AS People,E.Name As Employees
FROM Sample.Person AS P,Sample.Employee AS E
ORDER BY P.Name
```

ORDER BY子句可以使用箭头语法(->)操作符在非基表的表中指定一个字段:

```
SELECT Name,Company->Name AS CompName
FROM Sample.Employee ORDER BY Company->Name,Name
```

列别名

列别名必须指定为文字。

不能在表达式中指定列别名,也不能使用变量提供它。

下面的ORDER BY子句按列别名排序:

```
SELECT Name,Home_State AS HS,DOB
FROM Sample.Person
ORDER BY HS,Name
```

列别名可以与列名相同(尽管不建议这样做)。

如果提供了列别名,ORDER BY首先引用列别名,然后引用任何没有别名的列名。

如果列别名和非别名列名之间存在歧义,ORDER BY子句将生成SQLCODE -24错误。

但是,如果列别名与别名列名相同,这种明显的歧义不会生成错误,但可能会产生意想不到的结果。

下面的例子显示了这一点:

```
SELECT Name AS Moniker,Home_City AS Name
FROM Sample.Person
ORDER BY Name
```

可以使用列别名按选择项列表中的表达式进行排序,如下面的示例所示:

```
SELECT Name,Age,$PIECE(AVG(Age)-Age,'.',1) AS AgeDev
FROM Sample.Employee ORDER BY AgeDev,Name
```

不能指定一个非字段的列名默认值，比如Expression3;
相反，指定选择项列号(在本例中为3)，或者最好为该选择项指定列别名。

Column Number

列号必须指定为无符号数字字面值。
不能将列号指定为变量或表达式的结果。
不能将列号括在括号中。
整数截断规则用于将非整数值解析为整数;
例如，1.99解析为1。

下面的ORDER BY子句按列号(检索列的数字序列，如SELECT选择项列表中指定的)排序:

```
SELECT Name,Home_State,DOB
FROM Sample.Person
ORDER BY 2,1
```

列号指的是SELECT子句列表中的位置。
它们不指向表本身中列的位置。
但是，可以按列号对SELECT *结果进行排序;
如果RowID是公共的，它就被计算为第1列，如果RowID是隐藏的，它就不被计算为第1列。

在ORDER BY中指定与SELECT列表列不对应的列号将导致SQLCODE -5错误。
ORDER BY 0导致SQLCODE -5错误。

可以使用列号按选择项列表中的表达式进行排序，如下面的示例所示:

```
SELECT Name,Age,$PIECE(AVG(Age)-Age,'.',1)
FROM Sample.Employee ORDER BY 3,Name
```

指定排序

排序是按照排序顺序进行的。
默认情况下，字符串值的排序是根据创建时为order -item字段指定的排序规则进行的。
IRIS对每个名称空间都有一个默认的字符串排序规则;
字符串数据类型字段的初始排序规则默认值是SQLUPPER，不区分大小写。
因此，ORDER BY排序通常不区分大小写。

数字数据类型字段的排序是基于数字排序规则完成的。
对于表达式，默认排序规则是EXACT。

通过对排序项字段名应用排序规则函数，可以覆盖字段的默认排序规则。
例如，ORDER BY %EXACT(Name)。
不能对列别名应用排序规则函数;
尝试这样做会产生SQLCODE -29错误。

默认升序排序顺序认为NULL是最小值，后面跟着空字符串("")。
ORDER BY不区分空字符串和仅由空格组成的字符串。

如果为列指定的排序规则是字母数字的，则前导数字将按字符排序顺序而不是整数顺序排序。
可以使用%PLUS排序函数按整数顺序排序。

但是，%PLUS排序函数将所有非数字字符视为0。

因此，要正确地以数字序列对混合数字字符串进行排序，需要多个排序项。

例如，在Sample中。

街道地址由一个整数门牌号和街道名之间用一个空格隔开。

街道名由两个部分组成，中间用一个空格隔开。

比较下面两个例子。

第一个示例按字符排序顺序对街道地址进行排序：

```
SELECT Name,Home_Street FROM Sample.Person
ORDER BY Home_Street
```

第二个示例按整数顺序对房屋编号进行排序，按字符排序顺序对街道名称进行排序：

```
SELECT Name,Home_Street FROM Sample.Person
ORDER BY $PIECE(%PLUS(Home_Street),' ',1),$PIECE(Home_Street,' ',2),$PIECE(Home_Street,' ',3)
```

请注意，此示例仅适用于列名，而不适用于列别名或列编号。

ASC和DESC

可以按升序或降序排序顺序为每一列指定排序，由列标识符后面的可选ASC(升序)或DESC(降序)关键字指定。

如果未指定ASC或DESC，则ORDER BY按升序对该列进行排序。

你不能指定ASC或DESC关键字使用动态SQL？

输入参数或嵌入式SQL:var主机变量。

NULL总是ASC序列中的最低值和DESC序列中的最高值。

多个逗号分隔的ORDER BY值指定排序操作的层次结构，如下面的示例所示：

```
SELECT A,B,C,M,E,X,J
FROM LetterTable
ORDER BY 3,7 DESC,1 ASC
```

本例将SELECT子句列表中第三个列表项(C)的数据值按升序排序；

在这个序列中，它按降序对第7个列出的项(J)值进行排序；

在其中，它按升序对第一个列出的项(A)值进行排序。

ORDER BY值列表中的重复列不起作用。

这是因为第二种排序在第一种排序的顺序之内。

例如，ORDER BY Name ASC、Name DESC按升序对Name列进行排序。

NLS排序

如果指定了非默认的NLS排序规则，则必须确保所有排序规则都对齐并使用完全相同的国家排序规则序列。

这不仅包括表使用的全局变量，还包括临时文件(如IRISTEMP和process-private globals)中用于索引的全局变量。

[#SQL #Caché](#)

Published on InterSystems Developer Community (<https://community.intersystems.com>)

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