
文章

[姚鑫](#) · 十一月 2, 2021 阅读大约需 5 分钟

第六十四章 SQL命令 ORDER BY (二)

第六十四章 SQL命令 ORDER BY (二)

示例

下面的示例按照RowID的反向顺序对记录进行排序:

```
SELECT %ID,Name
FROM Sample.Person
ORDER BY %ID DESC
```

下面两个示例展示了在ORDER BY子句中指定排序列的不同方法。

下面两个查询是等价的;

第一种方法使用列名作为排序项,第二种方法使用列号(选择项列表中项目的序号):

```
SELECT Name,Age,Home_State
FROM Sample.Person
ORDER BY Home_State,Age DESC
```

```
SELECT Name,Age,Home_State
FROM Sample.Person
ORDER BY 3,2 DESC
```

下面的示例按包含IRIS列表数据的字段进行排序。

因为IRIS列表是一个以格式化字符开始的编码字符串,所以本例使用\$LISTTOSTRING来按实际字段数据值排序,而不是按列表元素编码:

```
SELECT Name,FavoriteColors
FROM Sample.Person
WHERE FavoriteColors IS NOT NULL
ORDER BY $LISTTOSTRING(FavoriteColors)
```

动态SQL可以使用输入参数为ORDER BY子句提供文字值;

它不能使用输入参数来提供字段名、字段别名、字段号或排序关键字。

下面的动态SQL示例使用输入参数按名字对结果集记录进行排序:

```
ClassMethod OrderBy()
{
    s myquery = 4
    s myquery(1) = "SELECT TOP ? Name,Age,"
    s myquery(2) = "CURRENT_DATE AS Today"
```

```
s myquery(3) = "FROM Sample.Person WHERE Age > ?"
s myquery(4) = "ORDER BY $PIECE(Name, ',', ?)"
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(10,60,2)
d rset.%Display()
w !,"%Display SQLCODE=",rset.%SQLCODE
}
```

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

Name	Age	Today	
Tweed,Al O.	86	66035	
Vanzetti,Alexandra O.	67	66035	66035
Jung,Alexandra S.	94	66035	66035
Gore,Alfred M.	63	66035	
Peterson,Alice E.	97	66035	66035
Houseman,Alice R.	63	66035	66035
Willeke,Alvin L.	79	66035	66035
Klingman,Alvin M.	73	66035	66035
Tillem,Andrew Z.	76	66035	66035
Zucherro,Angelo M.	93	66035	66035

10 Rows(s) Affected
%Display SQLCODE=100

以下基于游标的嵌入式SQL示例执行相同的操作:

```
ClassMethod OrderBy1()
{
    s topnum = 10, agemin = 60, firstname = 2
    &sql(
        DECLARE pCursor CURSOR FOR
            SELECT TOP :topnum Name, Age, CURRENT_DATE AS Today
            INTO :name, :years, :today FROM Sample.Person
            WHERE Age > :agemin
            ORDER BY $PIECE(Name, ',', :firstname) )
    &sql(OPEN pCursor)
    q:(SQLCODE'=0)
    for {
        &sql(FETCH pCursor)
        q:SQLCODE
        w "Name=",name," Age=",years," today=",today,!
    }
    &sql(CLOSE pCursor)
}
```

DHC-APP>d ##class(PHA.TEST.SQLCommand).OrderBy1()
Name=Tweed,Al O. Age=86 today=66035

```
Name=Vanzetti,Alexandra O. Age=67 today=66035
Name=Jung,Alexandra S. Age=94 today=66035
Name=Gore,Alfred M. Age=63 today=66035
Name=Peterson,Alice E. Age=97 today=66035
Name=Houseman,Alice R. Age=63 today=66035
Name=Willeke,Alvin L. Age=79 today=66035
Name=Klingman,Alvin M. Age=73 today=66035
Name=Tillem,Andrew Z. Age=76 today=66035
Name=Zucherro,Angelo M. Age=93 today=66035
```

缓存查询

ORDER BY子句中使用的每个字面值都会生成一个不同的缓存查询。
不对ORDER BY字面值执行字面值替换。
这是因为ORDER BY可以使用整数来指定列号。
更改这个整数将导致一个完全不同的查询。

ORDER BY and CASE

可以使用CASE表达式定义一个通用查询，该查询可以根据提供的主机变量值进行排序。
例如，下面的示例可以根据名称或年龄排序，这取决于var的值：

```
SELECT Name, Age FROM Sample.Person ORDER BY
CASE WHEN :var=1 then Name
      WHEN :var=2 then Age END
```

下面的示例指定了两个CASE表达式。
它按任何情况计算为true进行排序。
如果两种情况都为真，则按国家排序，在国家内按城市排序：

```
SELECT Country, City FROM Sample.Person ORDER BY
CASE WHEN :var1=1 then Country END,
      WHEN :var2=1 then City END
```

ASC和DESC参数在CASE END关键字之后指定。

注意，在CASE表达式中必须根据列名指定字段。
在此上下文中，不能指定列别名或列号。

ORDER BY和长全局引用

ORDER BY ordering-item的值不应该超过(大约)400到500个字符，这取决于ordering-item的数量和其他因素。
如果一个ordering-item值超过这个最大长度，则运行带有ORDER BY子句的查询可能会导致SQLCODE -400致命错误。
这是因为全局引用的最大编码长度有限制，这是一个固定的 IRIS系统限制。
为了防止这个问题，在作为ORDER BY子句基础的字段的排序规则设置中使用截断长度。
例如，以下查询超过了这个限制：

```
ClassMethod OrderBy2()
```

```
{
    try {
        s myquery = 3
        s myquery(1) = "SELECT LocationCity,NarrativeSummary FROM Aviation.Event "
        s myquery(2) = "WHERE LocationCity %Startswith 'Be' "
        s myquery(3) = "ORDER BY NarrativeSummary"
        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()
        if rset.%SQLCODE = 0 {
            w !,"Executed query",!
        } else {
            s badSQL = ##class(%Exception.SQL).%New(,rset.%SQLCODE,,rset.%Message)
            throw badSQL
        }
        d rset.%Display()
        w !,"End of data"
        ret
    } catch exp {
        w "In the CATCH block",!
        if 1 = exp.%IsA("%Exception.SQL") {
            w "SQLCODE: ",exp.Code,!
            w "Message: ",exp.Data,!
        } else {
            w "Not an SQL exception",!
        }
        ret
    }
}
```

添加一个maxlen截断长度的排序函数允许该程序成功执行:

```
ClassMethod OrderBy3()
{
    try {
        s myquery = 3
        s myquery(1) = "SELECT LocationCity,NarrativeSummary FROM Aviation.Event "
        s myquery(2) = "WHERE LocationCity %Startswith 'Be' "
        s myquery(3) = "ORDER BY %SqlUpper(NarrativeSummary,400)"
        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()
        if rset.%SQLCODE = 0 {
            w !,"Executed query",!
        } else {
            s badSQL = ##class(%Exception.SQL).%New(,rset.%SQLCODE,,rset.%Message)
            throw badSQL
        }
    }
}
```

```
    }
    d rset.%Display()
    w !,"End of data"
    ret
} catch exp {
    w "In the CATCH block",!
    if 1 = exp.%IsA("%Exception.SQL") {
        w "SQLCODE: ",exp.Code,!
        w "Message: ",exp.Data,!
    } else {
        w "Not an SQL exception",!
    }
    ret
}
```

IRIS将字段的已整理值截断为400个字符。

请记住，如果字段内容在前400个字符内不是唯一的，则数据可能稍有混乱，但这种情况不太可能发生。

如果出现这种情况，可以尝试通过使用更大的值进行截断来避免显示无序的数据；

但是，如果值太大，将导致<SUBSCRIPT>错误。

还要注意，最大长度是全局引用的整个编码长度，包括全局名称的长度。

它不是简单的下标。

[#SQL #Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E5%85%AD%E5%8D%81%E5%9B%9B%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-order-%EF%BC%88%E4%BA%8C%E4%BC%89>