

文章

姚鑫 · 十二月 27, 2021 阅读大约需分钟

## 第章 SQL聚合函数 LIST

## 第章 SQL聚合函数 LIST

创建逗号分隔值列表的聚合函数。

### 大纲

```
LIST([ALL | DISTINCT [BY(col-list)]] string-expr [%FOREACH(col-list)] [%AFTERHAVING])
```

### 参数

- ALL - 可选-指定LIST返回string-expr的所有值的列表。  
如果没有指定关键字,这是默认值。
- DISTINCT - 可选-一个DISTINCT子句,指定LIST返回一个仅包含唯一的string-expr值的列表。  
DISTINCT可以指定BY(col-list)子句,其中col-list可以是单个字段,也可以是用逗号分隔的字段列表。
- string-expr - 计算结果为字符串的SQL表达式。  
通常是所选表中列的名称。
- %FOREACH(col-list) - 可选-列名或以逗号分隔的列名列表。
- %AFTERHAVING - 可选-应用在HAVING子句中条件。

### 描述

LIST聚合函数返回指定列中以逗号分隔的值列表。

一个简单的LIST(或LIST ALL)返回一个字符串,其中包含一个逗号分隔的列表,该列表由所选行中string-expr的所有值组成

其中string-expr为空字符串("")的行由逗号分隔列表中的占位符逗号表示。

string-expr为NULL的行不包含在逗号分隔的列表中。

如果只有一个string-expr值,并且是空字符串(""),LIST返回空字符串。

LIST DISTINCT返回一个字符串,该字符串包含一个逗号分隔的列表,该列表由所选行中string-expr的所有不同(唯一)值组成(DISTINCT col1)。

NULL string-expr不包含在逗号分隔的列表中。

LIST(DISTINCT BY(col2))

col1返回一个逗号分隔的列表,其中只包含那些col2值是不同(唯一的)的记录中的col1字段值。

但是请注意,不同的col2值可能包含一个单独的NULL值。

### 包含逗号的数据值

因为LIST使用逗号分隔string-expr值,所以LIST不应该用于包含逗号的数据值。

使用%DLIST或JSON\_ARRAYAGG代替。

## LIST 和 %SelectMode

可以使用%SelectMode属性指定LIST返回的数据显示模式:0=Logical(默认), 1=ODBC, 2= display.

注意, LIST用逗号分隔列值, 而ODBC模式用逗号分隔%LIST列值中的元素。因此, 在%LIST结构上使用LIST时, 使用ODBC模式会产生不明确的结果。

## LIST 和 ORDER BY

LIST函数将每个行中的一个表列的值组合成逗号分隔值列表。因为在计算所有聚合字段之后, 查询结果集中应用了一个ORDER BY子句, 所以ORDER BY不能直接影响这个列表中的值序列。在某些情况, LIST结果可能会按顺序出现, 但是不应该依赖这种顺序。在给定聚合结果值中列出的值不能显式排序。

## 最大列表大小

允许的最大LIST返回值是最大字符串长度, 3641,144个字符。

## 相关的聚合函数

- LIST返回一个逗号分隔的值列表。
- %DLIST返回一个包含每个值的元素的列表。
- JSON\_ARRAYAGG返回值的JSON数组。
- XMLAGG返回一个串接的值字符串。

## 示例

下面的嵌入式SQL示例返回一个主机变量, 该变量包含示例的Home\_State列中列出的所有值的逗号分隔列表。以字母A开头的人名表:

```
ClassMethod ListFunction()  
{  
    &sql(SELECT LIST(Home_State)  
        INTO :statelist  
        FROM Sample.Person  
        WHERE Home_State %STARTSWITH 'A')  
    w "The states are:",!,statelist  
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLCommand).ListFunction()  
The states are:  
AR,AL,AZ,AZ,AZ,AR,AL,AZ,AR,AL,AL,AR
```

注意, 这个列表包含重复的值。

下面的嵌入式SQL示例返回一个主机变量, 该变量包含示例的Home\_State列中列出的所有不同(唯一)值的逗号分隔列表。以字母A开头的人名表:

```

ClassMethod ListFunction1()
{
    &sql(SELECT LIST(DISTINCT Home_State)
        INTO :statelist
        FROM Sample.Person
        WHERE Home_State %STARTSWITH 'A')
    w "The distinct states are:",!,statelist
}

```

```

DHC-APP> d ##class(PHA.TEST.SQLCommand).ListFunction1()
The distinct states are:
AR,AL,AZ

```

下面的SQL示例为每个州创建了一个逗号分隔的列表，其中包含在Home\_City列中找到的所有值，以及按州列出的这些城市值的计数。

每个Home\_State行包含该状态的所有Home\_City值的列表。  
这些名单可能包括重复的城市名称：

```

SELECT Home_State,
        COUNT(Home_City) AS CityCount,
        LIST(Home_City) AS ListAllCities
FROM Sample.Person
GROUP BY Home_State

```

也许更有用的方法是在Home\_City列中列出每个州的不同值，用逗号分隔，如所示：

```

SELECT Home_State,
        COUNT(DISTINCT Home_City) AS DistCityCount,
        COUNT(Home_City) AS TotCityCount,
        LIST(DISTINCT Home_City) AS DistCitiesList
FROM Sample.Person
GROUP BY Home_State

```

注意，这个示例返回每个州的不同城市名称和总城市名称的整数计数。

下面的示例返回以“A”开头的Home\_State值列表。

它返回不同的Home\_State值(distinct Home\_State)；

与不同的Home\_City值(distinct BY(Home\_City)

Home\_State)相对应的Home\_State值，这可能包括一个唯一的Home\_City NULL；

和所有Home\_State值：

```

SELECT LIST(DISTINCT Home_State) AS DistStates,
        LIST(DISTINCT BY(Home_City) Home_State) AS DistCityStates,
        LIST(Home_State) AS AllStates
FROM Sample.Person
WHERE Home_State %STARTSWITH 'A'

```

下面的动态SQL示例使用%SelectMode属性DOB日期字段返回的值列表指定ODBC显示模式：

```

ClassMethod ListFunction2()
{
    s myquery = "SELECT LIST(DOB) AS DOBs FROM Sample.Person WHERE Name %STARTSWITH '
A'"
    s tStatement = ##class(%SQL.Statement).%New()
    s tStatement.%SelectMode=1
    s qStatus = tStatement.%Prepare(myquery)
    if qStatus'=1 {
        w "%Prepare failed:"
        d $System.Status.DisplayError(qStatus)
        q
    }
    s rset = tStatement.%Execute()
    d rset.%Display()
    w !,"End of data"
}

```

```

DHC-APP>d ##class(PHA.TEST.SQLCommand).ListFunction2()
DOBs
1973-06-17,1943-05-22,1943-04-21,2011-07-07,1993-10-05,1953-02-25,1968-11-18

1 Rows(s) Affected
End of data

```

下面的动态SQL示例使用%FOREACH关键字。  
它为每个不同的Home\_State返回一行，其中包含该Home\_State的年龄值列表：

```

ClassMethod ListFunction3()
{
    s myquery = 3
    s myquery(1) = "SELECT DISTINCT Home_State,"
    s myquery(2) = "LIST(Age %FOREACH(Home_State)) AgesForState "
    s myquery(3) = "FROM Sample.Person WHERE Home_State %STARTSWITH 'M'"
    s tStatement = ##class(%SQL.Statement).%New()
    s tStatement.%SelectMode=1
    s qStatus = tStatement.%Prepare(.myquery)
    if qStatus'=1 {
        w "%Prepare failed:"
        d $System.Status.DisplayError(qStatus)
        q
    }
    s rset = tStatement.%Execute()
    d rset.%Display()
    w !,"End of data"
}

```

```

DHC-APP>d ##class(PHA.TEST.SQLCommand).ListFunction3()
Home_State      AgesForState
MA              55
MD              10,36,31,85,10,94,28
ME              25,20,55,11
MI              11,75,46,17,71
MN              11,43,73,98
MO              77,33,42,55,89,26,87,45,34
MS              7,25,13

```

MT 4,35,54,6,23

8 Rows(s) Affected  
End of data

下面的示例使用了%**AFTERHAVING**关键字。

它为每个Home\_State返回一行，其中至少包含一个满足HAVING子句条件的Name值(以“M”开头的名称)。

第一个LIST函数返回该状态的所有名称的列表。

第二个LIST函数返回的列表只包含满足HAVING子句条件的名称：

```
SELECT Home_State,
       LIST(Name) AS AllNames,
       LIST(Name %AFTERHAVING) AS HavingClauseNames
FROM Sample.Person
GROUP BY Home_State
HAVING Name LIKE 'M%'
ORDER BY Home_State
```

[#SQL #Caché](#)

源 URL: <https://cn.community.intersystems.com/post/%E7%AC%AC%E4%B8%83%E7%AB%A0-sql%E8%81%9A%E5%90%88%E5%87%BD%E6%95%B0-list>