文章
姚 鑫 · 十二月 26, 2021 阅读大约需 6 分钟

第六章 SQL聚合函数 JSONARRAYAGG

第六章 SQL聚合函数 JSONARRAYAGG

创建JSON格式值数组的聚合函数。

注：IRIS可用，IRIS之前版本不可用。

大纲

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

参数

- ALL - 可选-指定JSONARRAYAGG返回一个JSON数组，其中包含string-expr的所有值。
  如果没有指定关键字，这是默认值。
- DISTINCT - 可选-
  一个DISTINCT子句，指定JSONARRAYAGG返回一个仅包含唯一字符串expr值的JSON数组。
  DISTINCT可以指定BY(colo -list)子句，其中colo -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子句中的条件。

描述

JSONARRAYAGG聚合函数返回指定列中值的JSON格式数组。

一个简单的JSONARRAYAGG(或JSONARRAYAGG ALL)返回一个JSON数组，其中包含所选行中string-expr的所有值。
字符串-expr为空字符串(")的行由数组中的(" u0000")表示。
字符串-expr为NULL的行不包含在数组中。
如果只有一个字符串-expr值，并且是空字符串(")，JSONARRAYAGG将返回JSON数组[" \u0000"]。
如果所有的string-expr值为NULL, JSONARRAYAGG返回一个空的JSON数组[]。

JSONARRAYAGG DISTINCT返回一个JSON数组，由所选行中string-expr的所有不同(唯一)值组成:JSONARRAYAGG(DISTINCT col1)。
NULL字符串-expr不包含在JSON数组中。
JSONARRAYAGG(DISTINCT BY(col2)
col1)返回一个JSON数组，该数组只包含记录中col2值是不同的(唯一的)的那些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。

JSONARRAYAGG string-expr不能是流字段。
指定流字段的结果是SQLCODE -37。

包含转义字符的数据值

- 双引号:如果一个string-expr值包含一个双引号字符("),JSONARRAYAGG使用字面转义序列\"来表示这个字符。
- 反斜杠：如果string-expr值包含反斜杠字符(\)，则JSONARRAYAGG使用文字转义序列\\表示该字符。
- 单引号：当string-expr值包含一个单引号作为文字字符时， SQL要求必须将此字符作为两个单引号字符("进行双倍转义。JSONARRAYAGG将此字符表示为单引号字符 ‘ 。

最大JSON数组大小

默认的JSONARRAYAGG返回类型是VARCHAR(8192)。
这个长度包括JSON数组格式化字符以及字段数据字符。
如果预期返回的值将需要大于8192，可以使用CAST函数指定一个更大的返回值。
例如CAST(JSONARRAYAGG(value)) AS VARCHAR(12000))。
如果实际返回的JSON数组长于JSONARRAYAGG返回类型长度，IRIS将在返回类型长度处截断JSON数组，而不会发出错误。
因为截断JSON数组会删除其关闭的]字符，这使得返回值无效。

JSONARRAYAGG 和 %SelectMode

可以使用%SelectMode属性为JSON数组中的元素指定数据显示值:0=Logical(默认值)，1=ODBC, 2= display。
如果string-expr包含一个%List结构，则元素以ODBC模式表示，用逗号分隔，在逻辑和显示模式中以%List格式字符表示，用 \转义序列表示。

JSONARRAYAGG和ORDER BY

JSONARRAYAGG函数将表中多行列的值组合成一个包含元素值的JSON数组。
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。
在某些情况下，JSONARRAYAGG结果可能是按顺序出现的，但是不应该依赖于这种顺序。
在给定聚合结果值中列出的值不能显式排序。

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IRIS列表。
- 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HomeState列中所有值的JSON数组。
以字母A开头的人名表:

```
SELECT JSON_ARRAYAGG(Home_State)
            INTO :statearray
            FROM Sample.Person
            WHERE Home_State %STARTSWITH 'A'

SELECT JSON_ARRAYAGG(CTLOC_Desc) FROM CT_Loc
```

注意，这个JSON数组包含重复的值。

下面的动态SQL示例返回一个主机变量，该变量包含样本的HomeState列中所有不同(唯一)值的JSON数组。
以字母A开头的人名表:

```
ClassMethod JsonArrayagg()
{
    s myquery = 2
    s myquery(1) = "SELECT JSON_ARRAYAGG(DISTINCT Home_State) AS DistinctStates "
    s myquery(2) = "FROM Sample.Person WHERE Home_State %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"
}
```

下面的SQL示例为每个州创建了一个JSON数组，其中包含在HomeCity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HomeState行包含该状态的所有HomeCity值的JSON数组。
这些JSON数组可能包含重复的城市名称:

```
SELECT Home_State,
       COUNT(Home_City) AS CityCount,
       JSON_ARRAYAGG(Home_City) AS ArrayAllCities
FROM Sample.Person
GROUP BY Home_State
```

更有用的是一个JSON数组的所有不同的值，发现在HomeCity列为每个州，如下所示的动态SQL示例:

```
ClassMethod JsonArrayagg1()
{
    s myquery = 4
    s myquery(1) = "SELECT Home_State,COUNT(DISTINCT Home_City) AS DistCityCount,"
    s myquery(2) = "COUNT(Home_City) AS TotCityCount,"
    s myquery(3) = "JSON_ARRAYAGG(DISTINCT Home_City) AS ArrayDistCities "
    s myquery(4) = "FROM Sample.Person GROUP BY Home_State"
    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"
}
```

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。

下面的动态SQL示例使用%SelectMode属性为DOB日期字段返回的JSON值数组指定ODBC显示模式:

```
ClassMethod JsonArrayagg2()
{
    s myquery = 2
    s myquery(1) = "SELECT JSON_ARRAYAGG(DOB) AS DOBs "
    s myquery(2) = "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"
}
```

下面的动态SQL示例使用%FOREACH关键字。
它为每个不同的HomeState返回一行，其中包含该HomeState的年龄值的JSON数组。

```
ClassMethod JsonArrayagg3()
{
    s myquery = 3
    s myquery(1) = "SELECT DISTINCT Home_State,"
    s myquery(2) = "JSON_ARRAYAGG(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"
}
```

下面的动态SQL示例使用%AFTERHAVING关键字。
它为每个HomeState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JSONARRAYAGG函数返回一个包含该状态所有名称的JSON数组。
第二个JSONARRAYAGG函数返回的JSON数组只包含满足HAVING子句条件的名称:

```
ClassMethod JsonArrayagg4()
{
    s myquery = 4
```

```
    s myquery(1) = "SELECT Home_State,JSON_ARRAYAGG(Name) AS AllNames,"
    s myquery(2) = "JSON_ARRAYAGG(Name %AFTERHAVING) AS HavingClauseNames "
    s myquery(3) = "FROM Sample.Person GROUP BY Home_State "
    s myquery(4) = "HAVING Name LIKE 'M%' ORDER BY Home_State"
    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"
}
```

#SQL #Caché

源URL:
https://cn.community.intersystems.com/post/%E7%AC%AC%E5%85%AD%E7%AB%A0-sql%E8%81%9A%E5%90%88%E5%87%BD%E6%95%B0-jsonarrayagg