

文章

[姚鑫](#) · 二月 28, 2022 阅读大约需 5 分钟

第六十八章 SQL函数 JSONARRAY

第六十八章 SQL函数 JSONARRAY

以JSON数组形式返回数据的转换函数。

大纲

`JSON_ARRAY(expression [,expression][,...] [NULL ON NULL | ABSENT ON NULL])`

参数

- `expression` -
表达式或逗号分隔的表达式列表。这些表达式可以包括列名、聚合函数、算术表达式、文字和文字NULL。
- `ABSENT ON NULL`
`NULL ON NULL` - 可选-指定如何在返回的JSON数组中表示空值的关键字短语。`NULL ON NULL`(缺省值)表示带有单词NULL(未引号)的NULL(缺少)数据。在NULL上不存在将从JSON数组中省略空数据；它不会保留占位符逗号。此关键字短语对空字符串值没有影响。

描述

`Jsonarray`接受表达式或(更常见的)逗号分隔的表达式列表，并返回包含这些值的JSON数组。`Jsonarray`可以在SELECT语句中与其他类型的SELECT-Items结合使用。可以在可以使用SQL函数的其他位置指定`jsonarray`，例如在WHERE子句中。

返回的JSON数组格式如下：

```
[ element1 , element2 , element3 ]
```

`Jsonarray`以字符串(用双引号括起来)或数字形式返回每个数组元素值。数字以规范格式返回。数字字符串以文字形式返回，用双引号括起来。所有其他数据类型(例如，DATE或\$LIST)都作为字符串返回。

`Jsonarray`不支持将星号(*)语法作为指定表中所有字段的方式。它支持COUNT(*)聚合函数。

返回的JSON数组列被标记为表达式(默认情况下)；可以为JSONARRAY指定列别名。

选择模式和归类

当前%SelectMode属性确定返回的JSON数组值的格式。通过更改Select Mode，所有Date和%List元素都以该Select Mode格式的字符串包含在JSON数组中。

可以通过将格式转换函数(%EXTERNAL、%INTERNAL、%ODBCIN、%ODBCOUT)应用于JSONARRAY内的各个字段名来覆盖当前的选择模式。将格式转换函数应用于JSONARRAY没有任何效果，因为JSON数组的元素是字符

串。

可以将归类函数应用于JSONARRAY内的单个字段名或整个JSONARRAY：

- 应用于JSONARRAY的排序函数在JSON数组格式化之后应用排序规则。因此，%SQLUPPER(JSONARRAY(f1, f2))将所有JSON数组元素值转换为大写。%SQLUPPER(JSONARRAY(f1, f2))在JSON数组之前插入一个空格，而不是在数组元素之前插入一个空格；因此它不会强制将数字解析为字符串。
- 应用于JSONARRAY中的元素的排序规则函数将应用该排序规则。因此，JSONARRAY(' abc ', %SQLUPPER(' abc '))返回["Abc"," ABC"]（注意前导空格）；而JSONARRAY(007, %SQLSTRING(007))返回[7," 7"]。由于%SQLUPPER会在值之前插入一个空格，因此通常最好指定大小写转换函数，如LCASE或UCASE。可以将排序规则应用于元素和整个数组：%SQLUPPER(JSONARRAY('Abc',%SQLSTRING('Abc')))) returns ["ABC"," ABC"]

ABSENT ON NULL

如果指定可选的ACESING ON NULL关键字短语，则JSON数组中不包括NULL(或NULL文字)列值。JSON数组中不包括占位符。这可能会导致JSON数组具有不同数量的元素。例如，下面的程序返回JSON数组，其中对于某些记录，第三个数组元素是Age，对于其他记录，第三个数组元素是FavoriteColors：

```
SELECT JSON_ARRAY(%ID,Name,FavoriteColors,Age ABSENT ON NULL) FROM Sample.Person
```

如果未指定关键字短语，则NULL的默认值为NULL：NULL由单词NULL(未用引号分隔)表示为逗号分隔的数组元素。因此，JSONARRAY函数返回的所有JSON数组都将具有相同数量的数组元素。

示例

下面的示例应用JSONARRAY来格式化包含逗号分隔的字段值列表的JSON数组：

```
SELECT TOP 3 JSON_ARRAY(%ID,Name,Age,Home_State) FROM Sample.Person
```

下面的示例应用JSONARRAY使用包含名称字段值的单个元素格式化JSON数组：

```
SELECT TOP 3 JSON_ARRAY(Name) FROM Sample.Person
```

下面的示例应用JSONARRAY来格式化包含文字和字段值的JSON数组：

```
SELECT TOP 3 JSON_ARRAY('Employee from',%TABLENAME,Name,SSN) FROM Sample.Employee
```

下面的示例应用JSONARRAY来格式化包含空值和字段值的JSON数组：

```
SELECT JSON_ARRAY(Name,FavoriteColors) FROM Sample.Person  
WHERE Name %STARTSWITH 'S'
```

下面的示例应用JSONARRAY来格式化包含联接表中的字段值的JSON数组：

```
SELECT TOP 3 JSON_ARRAY(E.%TABLENAME,E.Name,C.%TABLENAME,C.Name)
FROM Sample.Employee AS E,Sample.Company AS C
```

下面的动态SQL示例设置ODBC

%SelectMode, 它确定如何表示所有字段, 包括JSON数组值。该查询通过应用%EXTERNAL FORMAT-CONVERSION函数覆盖特定JSON数组元素的此选择模式:

```
ClassMethod JsonArray()
{
    s myquery = 3
    s myquery(1) = "SELECT TOP 8 DOB,JSON_ARRAY(Name,DOB,FavoriteColors) AS ODBCMode,
    "
    s myquery(2) = "JSON_ARRAY(Name,DOB,%EXTERNAL(DOB),%EXTERNAL(FavoriteColors)) AS
ExternalTrans "
    s myquery(3) = "FROM Sample.Person"
    s tStatement = ##class(%SQL.Statement).%New()
    s tStatement.%SelectMode=1
    w "SelectMode is ODBC",!
    s qStatus = tStatement.%Prepare(.myquery)

    s rset = tStatement.%Execute()
    if rset.%SQLCODE=0 {
        w !,"Executed query",!
    } else {
        s badSQL=##class(%Exception.SQL).%New(,rset.%SQLCODE, , rset.%Message)
    }
    d rset.%Display()
    w !,"End of data"
}
```

```
DHC-APP> d ##class(PHA.TEST.SQLCommand).JsonArray()
SelectMode is ODBC
```

Executed query

DOB	ODBCMode	ExternalTrans
1990-04-25	["yaoxin","1990-04-25","Red,Orange,Yellow"]	["yaoxin","1990-04-25", ",04/25/1990","Red\r\nOrange\r\nYellow"]
	["xiaoli",null,null]	["xiaoli",null,null,null]
2014-01-02	["??","2014-01-02",null]	["??","2014-01-02","01/02/2014",null]
2014-01-02	["??","2014-01-02",null]	["??","2014-01-02","01/02/2014",null]
1978-01-28	["??","1978-01-28",null]	["??","1978-01-28","01/28/1978",null]
	["??",null,"Red,Orange,Yellow,Green"]	["??",null,null,"Red\r\nOrange\r\nYellow\r\nGreen"]
	["??",null,"Red,Orange,Yellow,Green,Green"]	["??",null,null,"Red\r\nOrange\r\nYellow\r\nGreen\r\nGreen"]
	["Isaacs,Roberta Z.",null,"Red,Orange,Yellow,Green,Yellow"]	["Isaacs,Roberta Z.",null,null,"Red\r\nOrange\r\nYellow\r\nGreen\r\nYellow"]

```
8 Rows(s) Affected
End of data
```

下面的示例在WHERE子句中使用JSONARRAY在不使用OR语法的情况下对多列执行CONTAINS测试:

```
SELECT Name,Home_City,Home_State FROM Sample.Person  
WHERE JSON_ARRAY(Name,Home_City,Home_State) [ 'X'
```

[#SQL](#) [#Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E5%85%AD%E5%8D%81%E5%85%AB%E7%AB%A0-sql%E5%87%BD%E6%95%B0-jsonarray>