文章
姚 鑫 · 六月 2, 2022 阅读大约需 4 分钟

第160章 SQL函数 XMLFOREST

第160章 SQL函数 XMLFOREST

格式化多个 XML 标记以包含表达式值的函数。

大纲

```
XMLFOREST(expression [AS tag][,expression [AS tag]])
```

参数

- expression - 任何有效的表达式。通常是包含要标记的数据值的列的名称。当指定为逗号分隔列表时，列表中的每个表达式都将包含在其自己的 XML 标记标记中。
- AS tag - 可选 — XML 标记标记的名称。如果指定了标签，则 AS 关键字是必需的。保留标签中字母的大小写。
  用双引号括起来的标签是可选的。如果省略双引号，标签必须遵循 XML 命名标准。用双引号括起来的标签消除了这些命名限制。

XMLFOREST 对有效的标签名称实施 XML 命名标准。它不能包含任何字符!"#$%&'()*+,/;<=>?@[\]^{|}~，也不能包含空格字符，并且不能以 "-", ".", 或数字。

如果指定不带 AS 标记子句的表达式，则标记值是表达式列的名称（大写字母）：<HOMECITY>Chicago</HOMECITY>。

描述

XMLFOREST 函数返回用其自己的 XML 标记开始标记和结束标记标记的每个表达式的值，如 tag 中指定的那样。例如，XMLFOREST(HomeCity AS City,HomeState AS State) 返回如下值：<City>Chicago</City><State>IL</State>。 XMLFOREST 不能用于生成空元素标记。

XMLFOREST 可用于引用表或视图的 SELECT 查询或子查询。 XMLFOREST 可以与普通列值一起出现在 SELECT 列表中。

返回指定的表达式值，由开始标记和结束标记括起来，格式如下：

```
<tag>value</tag>
```

通常，表达式是列的名称，或者是包含一个或多个列名称的表达式。表达式可以是任何类型的字段，包括数据流字段。 XMLFOREST 对每个表达式进行如下标记：
- 如果指定了 AS 标记，则 XMLFOREST 用指定的标记标记结果值。标签值区分大小写。
- 如果省略 AS 标记，并且表达式是列名，则 XMLFOREST 用列名标记结果值。列名默认标记始终为大写。
- 如果表达式不是列名（例如，聚合函数、文字或两列的串联），则需要 AS 标记子句。
- 如果表达式是一个流字段，则流值在生成的 XML 值中使用 <![CDATA[...]]>进行转义：

```
<tag><![CDATA[value]]></tag>
```

XMLFOREST 为逗号分隔列表中的每个项目提供一个单独的标记。 XMLELEMENT 将逗号分隔列表中的所有项目连接到单个标记中。

XMLFOREST 函数可以嵌套。允许嵌套 XMLFOREST 和 XMLELEMENT 函数的任何组合。可以使用 XMLCONCAT 连接 XMLFOREST 函数。

NULL 值

XMLFOREST 函数仅返回实际数据值的标记。当表达式值为 NULL 时，它不返回标记。例如：

```
INSERT INTO Sample.Xmltest (f1,f2,f3) values (NULL,'Row 1',NULL)
```

```
SELECT XMLFOREST(f1,f2,f3) from Sample.Xmltest
```

```
<F2>Row 1</F2>.
```

空字符串 ('') 被视为字符串数据类型字段的数据值。如果要标记的 f3 值为空字符串 ('')，则 XMLFOREST 返回：

```
<F3></F3>
```

XMLFOREST 在处理 NULL 方面与 XMLELEMENT 不同。 XMLELEMENT 始终返回一个标记值，即使字段值为 NULL。因此，XMLELEMENT 不区分 NULL 或空字符串。两者都表示为 <tag></tag>。

标点符号值

如果数据值包含 XML/HTML 可能解释为标记或其他编码的标点字符，则 XMLELEMENT 和 XMLFOREST 将此字符转换为相应的编码形式：

- ampersand (&) 变成 &
- apostrophe (') 变成 '
- quotation mark (") 变成 "
- open angle bracket (<) 变成 <
- close angle bracket (>) 变成 >

要在提供的文本字符串中表示撇号，请指定两个撇号，如下例所示：'can''t'。列数据不需要加倍撇号。

示例

以下查询将 Sample.Person 中的 Name 列值作为普通数据和 xml 标记数据返回：

```
SELECT Name,XMLFOREST(Name) AS ExportName
     FROM Sample.Person
```

返回的数据示例行如下所示。这里的标签默认为列的名称：

```
Name                     ExportName
Emerson,Molly N.   <NAME>Emerson,Molly N.</NAME>
```

以下示例指定了多个列：

```
SELECT XMLFOREST(Home_City,
                 Home_State AS Home_State,
                 AVG(Age) AS AvAge) AS ExportData
FROM Sample.Person
```

HomeCity 字段不指定标签；该标记由所有大写字母的列名生成：<HOMECITY>。 HomeState 字段的 AS 子句是可选的。之所以在此处指定，是因为指定标记名称允许您控制标记的大小写：<HomeState>，而不是 <HOMESTATE>;。 AVG(Age) AS子句是强制性的，因为该值是聚合值，而不是列值，因此没有列名。返回的数据示例行如下所示。

```
ExportData
<HOME_CITY>Chicago</HOME_CITY><Home_State>IL</Home_State>
<AvAge>48.0198019801980198</AvAge>
```

以下示例返回字符流数据：

```
SELECT XMLFOREST(name AS Para,Notes AS Para) AS XMLJobHistory
     FROM Sample.Employee
```

返回的数据示例行如下所示：

```
XMLJobHistory
<Para>Emerson,Molly N.</Para><Para><![CDATA[Molly worked at DynaMatix Holdings Inc. a
s a Marketing Manager]]></Para>
```

以下示例显示了使用子查询值的 XMLFOREST 函数：

```
SELECT XMLFOREST(Name,DOB,Age,
```

```
   (SELECT XMLFOREST(Name,DOB) FROM Sample.Person WHERE %ID=2) AS ExportName)
FROM Sample.Person where %ID=1
```

返回的数据示例行如下所示：

```
<NAME>Zahn,Rob F.</NAME><DOB>38405</DOB><AGE>71</AGE><ExportName><NAME>Quinn,Mark N.<
/NAME><DOB>30999</DOB></ExportName>
```

#SQL #Caché

源URL: https://cn.community.intersystems.com/post/%E7%AC%AC160%E7%AB%A0-sql%E5%87%BD%E6%95%B0-xmlforest