

文章

[姚鑫](#) · 三月 20, 2022 阅读大约需 4 分钟

## 第八十七章 SQL函数 \$LISTSAME

## 第八十七章 SQL函数 \$LISTSAME

比较两个列表并返回布尔值的列表函数。

### 大纲

`$LISTSAME(list1,list2)`

### 参数

- list1 - 计算结果为有效列表的表达式。
- list2 - 计算结果为有效列表的表达式。

### 描述

`$LISTSAME` 比较两个列表的内容，如果列表相同则返回 1。如果列表不相同，则 `$LISTSAME` 返回 0。`$LISTSAME` 逐个元素地比较两个列表。对于两个相同的列表，它们必须包含相同数量的元素，并且 list1 中的每个元素必须与 list2 中的相应元素匹配。

`$LISTSAME` 使用它们的字符串表示来比较列表元素。`$LISTSAME` 比较区分大小写。`$LISTSAME` 按从左到右的顺序逐个元素地比较两个列表。因此，`$LISTSAME` 在遇到第一个不匹配的列表元素时返回 0；它不检查后续项目以确定它们是否是有效的列表元素。

此函数返回 SMALLINT 类型的数据。

### 参数

#### list (list1 and list2)

列表是包含一个或多个元素的编码字符串。可以使用 SQL `$LISTBUILD` 函数或 ObjectScript `$LISTBUILD` 函数创建列表。可以使用 SQL `$LISTFROMSTRING` 函数或 ObjectScript `$LISTFROMSTRING` 函数将分隔字符串转换为列表。可以使用 SQL `$LIST` 函数或 ObjectScript `$LIST` 函数从现有列表中提取列表。

以下是有效列表的示例：

- `$LISTBUILD('a','b','c')`：一个三元素列表。
- `$LISTBUILD('a','', 'c')`：三元素列表，第二个元素为空字符串值。
- `$LISTBUILD('a','', 'c')` 或 `$LISTBUILD('a',NULL,'c')`：三元素列表，第二个元素没有值。
- `$LISTBUILD(NULL,NULL)` 或 `$LISTBUILD(,NULL)`：包含两个元素的列表，其中的元素没有值。
- `$LISTBUILD(NULL)` 或 `$LISTBUILD()`：单元素列表，该元素没有值。

如果列表参数为 NULL，则 `$LISTSAME` 返回 NULL。如果列表参数不是有效列表（并且不是 NULL）SQL 会生成 SQLCODE -400 致命错误。

## 示例

以下嵌入式 SQL 示例使用 \$LISTSAME 比较两个列表参数：

```

/// d ##class(PHA.TEST.SQLFunction).ListSame()
ClassMethod ListSame()
{
    s a = $lb("Red", , "Yellow", "Green", "", "Violet")
    s b = $lb("Red", , "Yellow", "Green", "", "Violet")
    &sql(
        SELECT $LISTSAME(:a,:b)
            INTO :c
    )
    if SQLCODE '= 0 {
        w !,"Error code;",SQLCODE
    } elseif c=1 {
        w "lists a and b are the same",!
    } else {
        w "lists a and b are not the same",!
    }
}

```

```

DHC-APP>d ##class(PHA.TEST.SQLFunction).ListSame()
lists a and b are the same

```

以下 SQL 示例比较具有 NULL、缺失或空字符串元素的列表：

```

SELECT $LISTSAME($LISTBUILD('Red',NULL,'Blue'),$LISTBUILD('Red',,'Blue')) AS NullAbsent,
        $LISTSAME($LISTBUILD('Red',NULL,'Blue'),$LISTBUILD('Red','','Blue')) AS NullEmpty,
        $LISTSAME($LISTBUILD('Red',,'Blue'),$LISTBUILD('Red','','Blue')) AS AbsentEmpty

0 0 0

```

### \$LISTSAME 比较与 ObjectScript

等号使用的等价测试不同。等号将两个列表作为编码字符串（逐个字符）进行比较；\$LISTSAME 逐个元素地比较两个列表。在比较数字和数字字符串时很容易看出这种区别，如下例所示：

```

/// d ##class(PHA.TEST.SQLFunction).ListSame1()
ClassMethod ListSame1()
{
    s a = $lb("365")
    s b = $lb(365)
    if a = b {
        w "Equal sign: lists a and b are the same",!
    } else {
        w "Equal sign: lists a and b are not the same",!
    }
}

```

```

}
&sql(
    SELECT $LISTSAME(:a,:b)
        INTO :c
)
if SQLCODE '= 0 {
    w !,"Error code ",SQLCODE
} elseif c=1 {
    w "$LISTSAME: lists a and b are the same",!
} else {
    w "$LISTSAME: lists a and b are not the same",!
}
}
}

```

```

DHC-APP>d ##class(PHA.TEST.SQLFunction).ListSame1()
Equal sign: lists a and b are not the same
$LISTSAME: lists a and b are the same

```

以下 SQL 示例比较了包含规范和非规范形式的数字和数字字符串的列表。比较数字列表元素和字符串列表元素时，字符串列表元素必须以规范形式表示数字；这是因为 IRIS 在执行比较之前总是将数字简化为规范形式。在以下示例中，\$LISTSAME 比较字符串和数字。前三个 \$LISTSAME 函数返回 1（相同）；第四个 \$LISTSAME 函数返回 0（不相同），因为字符串表示不是规范形式：

```

SELECT $LISTSAME($LISTBUILD('365'),$LISTBUILD(365)),
        $LISTSAME($LISTBUILD('365'),$LISTBUILD(365.0)),
        $LISTSAME($LISTBUILD('365.5'),$LISTBUILD(365.5)),
        $LISTSAME($LISTBUILD('365.0'),$LISTBUILD(365.0))

```

```

1 1 1 0

```

[#SQL #Caché](#)

**源**

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E5%85%AB%E5%8D%81%E4%B8%83%E7%AB%A0-sql%E5%87%BD%E6%95%B0-listsame>