

文章

[姚鑫](#) · 五月 17, 2022 阅读大约需 4 分钟

第145章 SQL函数 TONUMBER

第145章 SQL函数 TONUMBER

将字符串表达式转换为 NUMERIC 数据类型的值的字符串函数。

大纲

TO_NUMBER(string-expression)

TONUMBER(string-expression)

参数

- string-expression -
要转换的字符串表达式。表达式可以是列名、字符串文字或另一个函数的结果，其中基础数据类型为 CHAR 或 VARCHAR2 类型。

描述

名称 TONUMBER 和 TONUMBER 可以互换。它们支持 Oracle 兼容性。

TONUMBER 将字符串表达式转换为数字数据类型 NUMERIC。但是，如果 string-expression 的数据类型为 DOUBLE，则 TONUMBER 返回一个数据类型为 DOUBLE 的数字。

TONUMBER

转换采用数字字符串并通过解析加号和减号、扩展指数符号（“E”或“e”）以及删除前导零将其转换为规范数字。TONUMBER 在遇到非数字字符（例如字母或数字组分隔符）时停止转换。因此字符串 '7dwarves' 转换为 7。如果 string-expression 的第一个字符是非数字字符串，则 TONUMBER 返回 0。如果 string-expression 是空字符串 ('')，则 TONUMBER 返回 0。TONUMBER 将 -0 解析为 0。TONUMBER 不解析算术运算。因此字符串 '2+4' 转换为 2。如果为字符串表达式指定 NULL，则 TONUMBER 返回 null。

NUMERIC 数据类型的默认 SCALE 为 2。因此，在 DISPLAY 模式下选择此值时，TONUMBER 始终显示返回值，保留 2 位小数。额外的小数位四舍五入到小数点后两位；尾随零被解析为两位小数。当通过 xDBC 使用 TONUMBER 时，它还返回类型为 NUMERIC，SCALE 为 2。在 LOGICAL 模式或 ODBC 模式下，返回值是规范数字；没有对小数位施加比例，并且省略了尾随零。

相关 SQL 函数

- TONUMBER 将字符串转换为数字数据类型 NUMERIC。
- CAST 和 CONVERT
可用于将字符串转换为任意数据类型的数字。例如，可以将一个字符串转换为多个数据类型 INTEGER。
- TODATE 将格式化的日期字符串转换为日期整数。
- TOTIMESTAMP 将格式化的日期和时间字符串转换为标准时间戳。

示例

以下两个示例显示 TONUMBER 如何将字符串转换为数字，然后将其作为具有适当 SCALE 的数据类型 NUMERIC 返回。第一个示例在显示模式下返回数字，第二个示例在逻辑模式下返回数字：

```
/// d ##class(PHA.TEST.SQLFunction).ToNumber()
ClassMethod ToNumber()
{
    s myquery = "SELECT TO_NUMBER('+-+0123.0093degrees')"
    s tStatement = ##class(%SQL.Statement).%New()
    s tStatement.%SelectMode=2
    s qStatus = tStatement.%Prepare(myquery)
    s rset = tStatement.%Execute()
    d rset.%Display() // Display mode value: 123.01
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).ToNumber()
Expression_1
123.01
```

1 Rows(s) Affected

```
/// d ##class(PHA.TEST.SQLFunction).ToNumber1()
ClassMethod ToNumber1()
{
    s myquery = "SELECT TO_NUMBER('+-+0123.0093degrees')"
    s tStatement = ##class(%SQL.Statement).%New()
    s tStatement.%SelectMode=0
    s qStatus = tStatement.%Prepare(myquery)
    if qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QUIT}
    s rset = tStatement.%Execute()
    d rset.%Display() // Logical mode value: 123.0093
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).ToNumber1()
Expression_1
123.0093
```

1 Rows(s) Affected

以下示例显示当 string-expression 为 DOUBLE 数据类型时，TONUMBER 将值作为数据类型 DOUBLE 返回：

```
/// d ##class(PHA.TEST.SQLFunction).ToNumber2()
ClassMethod ToNumber2()
{
    s myquery = "SELECT TO_NUMBER(CAST('+-+0123.0093degrees' AS DOUBLE))"
    s tStatement = ##class(%SQL.Statement).%New()
    s tStatement.%SelectMode=2
    s qStatus = tStatement.%Prepare(myquery)
    if qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QUIT}
}
```

```
UIT}
    s rset = tStatement.%Execute()
    d rset.%Display() // Display mode value
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).ToNumber2()
```

```
Expression_1
```

```
123.00929999999999608
```

```
1 Rows(s) Affected
```

```
/// d ##class(PHA.TEST.SQLFunction).ToNumber3()
```

```
ClassMethod ToNumber3()
```

```
{
    s myquery = "SELECT TO_NUMBER(CAST('+-+0123.0093degrees' AS DOUBLE))"
    s tStatement = ##class(%SQL.Statement).%New()
    s tStatement.%SelectMode=0
    s qStatus = tStatement.%Prepare(myquery)
    if qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) Q
UIT}
    s rset = tStatement.%Execute()
    d rset.%Display() // Logical mode value
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).ToNumber3()
```

```
Expression_1
```

```
123.00929999999999608
```

```
1 Rows(s) Affected
```

以下示例显示如何使用 TONUMBER 列出按数字升序排列的街道地址：

```
SELECT Home_Street, Name
FROM Sample.Person
ORDER BY TO_NUMBER(Home_Street)
```

将结果与按字符串升序排列的相同数据进行比较：

```
SELECT Home_Street, Name
FROM Sample.Person
ORDER BY Home_Street
```

[#SQL #Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC145%E7%AB%A0-sql%E5%87%BD%E6%95%B0-tonumber>