

文章

[姚鑫](#) · 四月 22, 2022 阅读大约需 5 分钟

第120章 SQL函数 ROUND

第120章 SQL函数 ROUND

以指定位数舍入或截断数字的数值函数。

大纲

`ROUND(numeric-expr, scale[, flag])`

`{fn ROUND(numeric-expr, scale[, flag])}`

参数

- `numeric-expr` - 要四舍五入的数字。一个数值表达式。
- `scale` - 计算结果为整数的表达式，该整数指定要舍入到的位数，从小数点开始计数。可以是零、正整数或负整数。如果 `scale` 是小数，会将其四舍五入为最接近的整数。
- `flag` - 可选 — 一个布尔标志，指定是舍入还是截断 `numeric-expr`：0=round, 1=truncate。默认值为 0。

ROUND 返回与 `numeric-expr` 相同的数据类型。

描述

此函数可用于将数字舍入或截断为指定的小数位数。

ROUND 将 `numeric-expr` 舍入或截断以缩放位置，从小数点开始计数。舍入时，数字 5 始终向上舍入。在 ROUND 循环或截断操作后删除尾随零。不返回前导零。

- 如果 `scale` 为正数，则在小数点右侧的该位数处进行舍入。如果 `scale` 等于或大于小数位数，则不会发生舍入或零填充。

- 如果 `scale`

为零，则舍入到最接近的整数。换句话说，在小数点右边的零位处进行舍入；所有小数位和小数点本身都被删除。

- 如果 `scale` 为负数，则在小数点左侧的该位数处进行舍入。如果 `scale` 等于或大于舍入结果中的整数位数，则返回零。

- 如果 `numeric-expr` 为零（但表示为：00.00、-0 等），ROUND 将返回 0（零）且没有小数位，而不管比例值如何。

- 如果 `numeric-expr` 或 `scale` 为 NULL，ROUND 返回 NULL。

请注意，ROUND 返回值始终是标准化的，删除尾随零。

ROUND、TRUNCATE 和 \$JUSTIFY

ROUND 和 TRUNCATE 是执行类似操作的数值函数；它们都可用于减少数字的有效小数位数或整数位数。ROUND 允许指定舍入（默认）或截断；TRUNCATE 不执行舍入。ROUND 返回与 `numeric-expr` 相同的数据类型；TRUNCATE 返回 `numeric-expr` 作为数据类型 NUMERIC，除非 `numeric-expr` 是数据类型

DOUBLE，在这种情况下它返回数据类型 DOUBLE。

ROUND

舍入（或截断）到指定数量的小数位数，但其返回值始终是标准化的，删除尾随零。例如，ROUND(10.004,2) 返回 10，而不是 10.00。

TRUNCATE 截断到指定数量的小数位数。如果截断导致尾随零，则保留这些尾随零。但是，如果 scale 大于 numeric-expr 规范形式的小数位数，则 TRUNCATE 不会填充零。

当舍入到固定的小数位数很重要时使用 \$JUSTIFY - 例如，在表示货币金额时。\$JUSTIFY 在舍入操作之后返回指定数量的尾随零。当要舍入的位数大于小数位数时，\$JUSTIFY 补零。\$JUSTIFY 还右对齐数字，以便 DecimalSeparator 字符在一列数字中对齐。\$JUSTIFY 不会截断。

\$DOUBLE 数字

\$DOUBLE IEEE 浮点数使用二进制表示法编码。大多数十进制分数不能用这种二进制表示法精确表示。当 \$DOUBLE 值被输入到带有刻度值和舍入标志（flag = 0，默认值）的 ROUND 时，返回值通常包含比刻度中指定的更多的小数位数，因为小数位数的结果不能用二进制表示，所以返回值必须四舍五入到最接近的可表示的 \$DOUBLE 值，如以下示例所示：

```
/// d ##class(PHA.TEST.SQLFunction).Round()
ClassMethod Round()
{
    s x = 1234.5678
    s y = $DOUBLE(1234.5678)
    &sql(SELECT ROUND(:x,2),ROUND(:y,2) INTO :decnum,:dblnum)
    w "Decimal: ",x," rounded ",decnum,!
    w "Double: ",y," rounded ",dblnum
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).Round()
Decimal: 1234.5678 rounded 1234.57
Double: 1234.5678000000000000338 rounded 1234.569999999999999363
```

如果使用 ROUND 截断 \$DOUBLE 值（标志 = 1），则 \$DOUBLE 的返回值将被截断为小数位数指定的小数位数。TRUNCATE 函数还将 \$DOUBLE 截断为由 scale 指定的小数位数。

如果使用 ROUND 对 \$DOUBLE 值进行舍入并希望返回特定的比例，则应在舍入结果之前将 \$DOUBLE 值转换为十进制表示。

带有 flag=0 的 ROUND（round，默认值）返回 \$DOUBLE("INF") 和 \$DOUBLE("NAN") 作为空字符串。

如果使用 ROUND 对 \$DOUBLE 值进行舍入并希望返回特定的比例，则应在舍入结果之前将 \$DOUBLE 值转换为十进制表示。

带有 flag=0 的 ROUND（round，默认值）返回 \$DOUBLE("INF") 和 \$DOUBLE("NAN") 作为空字符串。

带有 flag=1（截断）的 ROUND 返回 \$DOUBLE("INF") 和 \$DOUBLE("NAN") 作为 INF 和 NAN。

示例

以下示例使用 0（零）的比例将多个分数舍入为整数。它表明 5 总是向上取整：

```
SELECT ROUND(5.99,0) AS RoundUp,  
       ROUND(5.5,0) AS Round5,  
       {fn ROUND(5.329,0)} AS Roundoff
```

6 6 5

以下示例截断与上一个示例相同的小数：

```
SELECT ROUND(5.99,0,1) AS Trunc1,  
       ROUND(5.5,0,1) AS Trunc2,  
       {fn ROUND(5.329,0,1)} AS Trunc3
```

5 5 5

以下 ROUND 函数对负小数进行舍入和截断：

```
SELECT ROUND(-0.987,2,0) AS Round1,  
       ROUND(-0.987,2,1) AS Trunc1
```

-0.99 -0.98

以下示例将 pi 舍入为四位小数：

```
SELECT {fn PI()} AS ExactPi, ROUND({fn PI()},4) AS ApproxPi
```

3.141592653589793238 3.1416

以下示例指定了大于十进制位数的比例：

```
SELECT {fn ROUND(654.98700,9)} AS Rounded
```

654.987

它返回 654.987（在舍入操作之前删除了尾随零；没有发生舍入或零填充）。

以下示例将 Salary 的值四舍五入到最接近的千美元：

```
SELECT Salary,ROUND(Salary, -3) AS PayBracket  
FROM Sample.Employee  
ORDER BY Salary
```

请注意，如果 Salary 小于 500 美元，则四舍五入为 0（零）。

在下面的示例中，每个 ROUND 都指定一个与要舍入的数字一样大或更大的负比例：

```
SELECT {fn ROUND(987, -3)} AS Round1,  
       {fn ROUND(487, -3)} AS Round2,
```

第120章 SQL函数 ROUND

Published on InterSystems Developer Community (<https://community.intersystems.com>)

```
{fn ROUND(987, -4)} AS Round3,  
{fn ROUND(987, -5)} AS Round4
```

```
1000      0      0      0
```

第一个 ROUND 函数返回 1000，因为舍入结果的位数多于小数位数。其他三个 ROUND 函数返回 0（零）。

[#SQL](#) [#Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC120%E7%AB%A0-sql%E5%87%BD%E6%95%B0-round>