

文章

[姚鑫](#) · 五月 30, 2022 阅读大约需 4 分钟

第157章 SQL函数 WEEK

第157章 SQL函数 WEEK

一个日期函数，它将一年中的第几周作为日期表达式的整数返回。

大纲

```
{fn WEEK(date-expression)}
```

参数

- date-expression - 一个表达式，它是列的名称、另一个标量函数的结果，或者是日期或时间戳文字。

描述

WEEK 接受一个日期表达式，并返回该日期从年初开始的周数。

默认情况下，使用 \$HOROLOG 日期（从 1840 年 12 月 31 日开始的正整数或负整数天数）计算周数。因此，周数是逐年计算的，因此第 1 周是完成从上一年最后一周开始的 7 天期间的天数。一周总是从星期日开始；因此，日历年的第一个星期日标志着从第 1 周到第 2 周的变化。如果一年中的第一个星期日是 1 月 1 日，则该星期日是第 1 周；如果一年中的第一个星期日晚于 1 月 1 日，则该星期日是第 2 周的第一天。因此，第 1 周的长度通常少于 7 天。可以使用 DAYOFWEEK 函数确定星期几。一年中的总周数通常为 53，闰年可能为 54。

IRIS 还支持确定一年中星期的 ISO 8601 标准。该标准主要用于欧洲国家。当配置为 ISO 8601 时，WEEK 从星期一开始计算星期，并将星期分配给包含该星期星期四的年份。例如，2004 年的第 1 周从 2003 年 12 月 29 日星期一到 2004 年 1 月 4 日星期日，因为本周的星期四是 2004 年 1 月 1 日，这是 2004 年的第一个星期四。2005 年的第 1 周是从 2005 年 1 月 3 日星期一到 1 月 9 日星期日 2005，因为它的星期四是 2005 年 1 月 6 日，也就是 2005 年的第一个星期四。一年中的总周数通常是 52，但偶尔也可能是 53。要激活 ISO 8601 计数，SET ^%SYS("sql","sys","week ISO8601")=1.

日期表达式可以是 日期整数、\$HOROLOG 或 \$ZTIMESTAMP 值、ODBC 格式日期字符串或时间戳。

日期表达式时间戳可以是数据类型 %Library.PosixTime（编码的 64 位有符号整数），也可以是数据类型 %Library.TimeStamp (yyyy-mm-dd hh:mm:ss.fff)。

时间戳的时间部分不被评估，可以省略。

使用 DATEPART 或 DATENAME 函数可以返回相同的星期信息。

也可以使用 WEEK()方法调用从 ObjectScript 调用此函数：

```
$SYSTEM.SQL.Functions.WEEK(date-expression)
```

日期验证

WEEK 对输入值执行以下检查。如果某个值未通过检查，则返回空字符串。

- 日期字符串必须完整且格式正确，其中包含适当数量的元素和每个元素的数字，以及适当的分隔符。年份必须指定为四位数。
- 日期值必须在有效范围内。年：0001 到 9999。月：1 到 12。日：1 到 31。
- 一个月的天数必须与月份和年份相匹配。例如，日期“02-29”仅在指定年份是闰年时有效。
- 小于 10 的日期值可以包括或省略前导零。不允许使用其他非规范整数值。因此，Day 值“07”或“7”有效，但“007”、“7.0”或“7a”无效。

示例

以下嵌入式 SQL 示例返回 2005 年 1 月 2 日（星期日）和 2006 年 1 月 1 日（星期日）的星期几和一年中的星期几。

```

/// d ##class(PHA.TEST.SQLFunction).Week()
ClassMethod Week()
{
    SET x = "2005-1-2"
    SET y = "2006-1-1"
    &sql(
        SELECT
            {fn DAYOFWEEK(:x)}, {fn WEEK(:x)},
            {fn DAYOFWEEK(:y)}, {fn WEEK(:y)}
        INTO
            :a, :b, :c, :d
    )
    if SQLCODE '= 0 {
        w !, "Error code ", SQLCODE
    } else {
        w !, "2005 Day of Week is: ", a, " (Sunday=1)"
        w " Week of Year is: ", b
        w !, "2006 Day of Week is: ", c, " (Sunday=1)"
        w " Week of Year is: ", d
    }
}

```

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

```

2005 Day of Week is: 1 (Sunday=1) Week of Year is: 2
2006 Day of Week is: 1 (Sunday=1) Week of Year is: 1

```

以下示例返回数字 9，因为日期是 2004 年的第九周：

```

SELECT {fn WEEK('2004-02-25')} AS Wk_Date,
       {fn WEEK('2004-02-25 08:35:22')} AS Wk_Tstamp,
       {fn WEEK(59590)} AS Wk_DInt

```

以下示例返回数字 54，因为此特定日期是闰年，从第 2 周开始，从第二天开始，如紧随其后的示例所示：

```
SELECT {fn WEEK('2000-12-31')} AS Week
```

```
54
```

```
SELECT {fn WEEK('2000-01-01')}||{fn DAYNAME('2000-01-01')} AS WeekofDay1,
       {fn WEEK('2000-01-02')}||{fn DAYNAME('2000-01-02')} AS WeekofDay2
```

```
1Saturday    2Sunday
```

以下示例均返回当前周：

```
SELECT {fn WEEK({fn NOW()})} AS Wk_Now,
       {fn WEEK(CURRENT_DATE)} AS Wk_CurrD,
       {fn WEEK(CURRENT_TIMESTAMP)} AS Wk_CurrTS,
       {fn WEEK($HOROLOG)} AS Wk_Horolog,
       {fn WEEK($ZTIMESTAMP)} AS Wk_ZTS
```

```
20  20  20  20  20
```

以下嵌入式 SQL 示例显示了 默认的一年中的一周以及应用 ISO 8601 标准的一年中的一周：

```
/// d ##class(PHA.TEST.SQLFunction).Week1()
ClassMethod Week1()
{
TestISO
    s def = $d(^%SYS("sql","sys","week ISO8601"))
    if def = 0 {
        s ^%SYS("sql","sys","week ISO8601")=0
    } else {
        s isoval = ^%SYS("sql","sys","week ISO8601")
    }
    if isoval = 1 {
        g UnsetISO
    } else {
        s isoval=0
        g WeekOfYear
    }
UnsetISO
    s ^%SYS("sql","sys","week ISO8601") = 0
WeekOfYear
    &sql(
        SELECT
            {fn WEEK($HOROLOG)}
        INTO
            :a
    )
    w "For Today:",!
    w "default week of year is ",a,!
}
```

```
s ^%SYS("sql","sys","week ISO8601") = 1
&sql(
    SELECT
        {fn WEEK($HOROLOG)}
    INTO
        :b
)
w "ISO8601 week of year is ",b,!
ResetISO
s ^%SYS("sql","sys","week ISO8601") = isoval
}
```

[#SQL](#) [#Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC157%E7%AB%A0-sql%E5%87%BD%E6%95%B0-week>