

文章

[姚鑫](#) · 二月 9 阅读大约需分钟

第四十九章 SQL函数 DAYOFWEEK

第四十九章 SQL函数 DAYOFWEEK

Date函数, 它返回日期表达式的整型星期几。

大纲

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

参数

- date-expression - 有效的ODBC格式日期或\$HOROLOG格式日期, 带或不带时间组件。作为列名、另一个标量函数的结果或日期或时间戳文字的表达式。

描述

DAYOFWEEK接受日期表达式, 并返回与该日期的星期几对应的整数。一周的天数从一周的第一天开始计算; 默认设置是星期天是一周的第一天。因此, 默认情况下, 返回值表示这些天:

- 1 — Sunday
- 2 — Monday
- 3 — Tuesday
- 4 — Wednesday
- 5 — Thursday
- 6 — Friday
- 7 — Saturday

请注意, ObjectScript \$ZDATE和\$ZDATETIME函数计算一周中的天数从0到6(而不是1到7)。

日期表达式可以是日期整数、\$HOROLOG或\$ZTIMESTAMP值、ODBC格式的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.PosiTime(编码的64位有符号整数)或数据类型%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。

可以使用DATEPART或TO_DATE函数返回星期几的信息。要返回星期几的名称, 请使用DAYNAME、DATENAME或TO_DATE。

也可以使用DAYOFWEEK()方法调用从ObjectScript调用此函数:

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

日期验证

DAYOFWEEK对输入值执行以检查。如果值未通过检查,则返回空字符串。

- 有效的日期表达式可以由日期字符串(yyyy-mm-dd)、日期和时间字符串(yyyy-mm-dd hh:mm:ss)、日期整数或\$HOROLOG值组成。DAYOFWEEK仅计算日期表达式的日期部分。
- 日期字符串必须完整且格式正确,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。年份:0001到9999。月份:1到12天:1到31
- 一个月中的天数必须与月和年匹配。例如,日期'02-29'只有在指定年份是闰年时才有效。
- 小于10的日期值可以包含或省略前导零。不允许其他非规范整数值。因此,“日期”值“07”或“7”有效,但“007”、“7.0”或“7a”无效。

设置每周第一天

默认情况下,一周第一天是周日。

你可以通过指定SET ^%SYS("sql","sys","day of week")=n来覆盖这个默认的系统范围,其中n个值是1=周一到7=周日。

SET ^%SYS("sql","sys","day of week")=1设置星期一为一周第一天。

如果Monday是一周第一天,周三日期表达式返回3,而不是如果Sunday是一周第一天将返回4。

如果要重置的默认值(周日为一周第一天),请指定^%SYS("sql","sys","day of week")=7。

通过设置SET ^%SYS("sql","sys","day of week",namespace)=n,其中n个值为1=周一到7=周日。

使用SET ^%SYS("sql","sys","day of week","USER")=1设置USER命名空间的周一为每周第一天。

一旦在命名空间级别设置一周第一天,通过指定^%SYS("sql","sys","day of week")=n更改系统范围的设置,对该命名空间没有影响。

为了恢复更改命名空间的默认第一天,kill ^%SYS("sql","sys","day of week",namespace)

还支持用于确定星期、星期和其他日期设置的ISO 8601标准。

这一标准主要在欧洲国家使用。

ISO 8601标准从星期一开始计算一周天数。

SET ^%SYS("sql","sys","week ISO8601")=1

若要禁用,请将其设置为0。

如果ISO8601未被激活,而一周天数未定义或设置为默认值(7=周日),则ISO8601标准将覆盖默认值。

如果 day of week设置为任何其他值,则DAYOFWEEK将覆盖ISO8601。

示例

在面的例子中,两个select项都返回数字5(如果周日被设置为一周第一天),因为指定的date-expression (64701 = February 22, 2018)是一个星期四:

```
SELECT {fn DAYOFWEEK('2018-02-22')} || ' ' || DATENAME('dw','2018-02-22') AS ODBCDoW,
       {fn DAYOFWEEK(64701)} || ' ' || DATENAME('dw','64701') AS HorologDoW
```

```
5 Thursday 5 Thursday
```

在面的例子中,所有的select-item返回的都是对应于当前日期的整数:

```
SELECT {fn DAYOFWEEK({fn NOW()})} AS DoW_Now,
```

```
{fn DAYOFWEEK(CURRENT_DATE)} AS DoW_CurrDate,
{fn DAYOFWEEK(CURRENT_TIMESTAMP)} AS DoW_CurrTstamp,
{fn DAYOFWEEK($ZTIMESTAMP)} AS DoW_ZTstamp,
{fn DAYOFWEEK($HOROLOG)} AS DoW_Horolog
```

3 3 3 3 3

注意\$ZTIMESTAMP返回协调世界时(UTC)。其他时间表达式值返回本地时间。这可能会影响DAYOFWEEK值。

下面的嵌入式SQL示例演示如何更改命名空间的每星期一。它最初设置系统范围的每星期一(设置为7)，然后设置命名空间的每星期一(设置为3)。在程序终止特定于命名空间的设置之前，后续的系统范围内的每星期一更改(更改为2)对命名空间第一天没有影响。取消特定于命名空间的设置会立即将该命名空间每星期一重置为当前系统范围的值。最后，程序恢复系统范围的初始设置。

注：以程序测试是否具有%SYS或user命名空间的特定于命名空间的每星期一设置。如果这样做，此程序将中止以阻止更改这些设置。

```
SetUp
  SET TestNsp="USER"
  SET ControlNsp="%SYS"
InitialDoWValues
  WRITE "Systemwide default DoW initial values",!
  DO TestDayofWeek()
  IF a=b {WRITE "No namespace-specific DoW defaults",!!}
  ELSE {WRITE "DoW initial settings are namespace-specific",!
        WRITE "Stopping this program"
        QUIT }
  SET initialDoW=^%SYS("sql","sys","day of week")
SetSystemwideDoW
  KILL ^%SYS("sql","sys","day of week",TestNsp)
  KILL ^%SYS("sql","sys","day of week",ControlNsp)
  SET ^%SYS("sql","sys","day of week")=7
  WRITE "Systemwide DoW set",!
  DO TestDayofWeek()
SetNamespaceDoW
  SET ^%SYS("sql","sys","day of week",TestNsp)=3
  WRITE TestNsp," namespace DoW set",!
  &sql(SELECT {fn DAYOFWEEK($HOROLOG)} INTO :a)
  DO TestDayofWeek()
ResetSystemwideDoW
  SET ^%SYS("sql","sys","day of week")=2
  WRITE "Systemwide DoW set with ",TestNsp," DoW set",!
  DO TestDayofWeek
KillNamespaceDoW
  KILL ^%SYS("sql","sys","day of week",TestNsp)
  WRITE "Namespace ",TestNsp," DoW killed",!
  DO TestDayofWeek
ResetSystemwideDoWDefault
  SET ^%SYS("sql","sys","day of week")=initialDoW
  WRITE "Systemwide DoW reset after ",TestNsp," DoW killed",!
  DO TestDayofWeek
TestDayofWeek()
  SET $NAMESPACE=TestNsp
  &sql(SELECT {fn DAYOFWEEK($HOROLOG)} INTO :a)
```

```
WRITE "Today is the ",a," day of week in ",$NAMESPACE,!
SET $NAMESPACE=ControlNsp
&sql(SELECT {fn DAYOFWEEK($HOROLOG)} INTO :b)
WRITE "Today is the ",b," day of week in ",$NAMESPACE,!!
RETURN
```

下面的嵌入式SQL示例显示了应用了ISO 8601标准的默认星期几和星期几。它假定星期几未定义或设置为默认值：

```
TestISO
SET def=$DATA(^%SYS("sql","sys","week ISO8601"))
IF def=0 {SET ^%SYS("sql","sys","week ISO8601")=0}
ELSE {SET isoval=^%SYS("sql","sys","week ISO8601")}
    IF isoval=1 {GOTO UnsetISO }
    ELSE {SET isoval=0 GOTO DayofWeek }
UnsetISO
SET ^%SYS("sql","sys","week ISO8601")=0
DayofWeek
&sql(SELECT {fn DAYOFWEEK($HOROLOG)} INTO :a)
WRITE "Today:",!
WRITE "default day of week is ",a,!
SET ^%SYS("sql","sys","week ISO8601")=1
&sql(SELECT {fn DAYOFWEEK($HOROLOG)} INTO :b)
WRITE "ISO8601 day of week is ",b,!
ResetISO
SET ^%SYS("sql","sys","week ISO8601")=isoval
```

[#SQL #Caché](#)

源 URL: <https://cn.community.intersystems.com/post/%E7%AC%AC%E5%9B%9B%E5%8D%81%E4%B9%9D%E7%AB%A0-sql%E5%87%BD%E6%95%B0-dayofweek>