

文章  
[姚鑫](#) · 二月 4 阅读大约需分钟

## 第四十四章 SQL函数 DATENAME

## 第四十四章 SQL函数 DATENAME

日期/时间函数,它返回一个字符串,表示日期/时间表达式中指定部分的值。

### 参数

- **datepart** - 要返回的日期/时间信息类型。  
日期或时间部分的名称(或缩写)。  
可以用大写或小写指定该名称,也可以不加引号。  
可以将datepart指定为文字或主机变量。
- **date-expression** - 要返回datepart值的日期、时间或时间戳表达式。  
日期表达式必须包含datepart类型的值。

### 描述

DATENAME函数返回日期/时间值中指定部分的名称(例如“June”)。  
结果作为数据类型VARCHAR(20)返回。  
如果结果是数字(例如“23”表示当天),它仍然作为VARCHAR(20)字符串返回。  
要以整数形式返回此信息,请使用DATEPART。  
要返回包含多个日期部分的字符串,请使用TO\_DATE。

请注意,DATENAME是为Sybase和Microsoft SQL Server兼容而提供的。

这个函数也可以通过调用DATENAME()方法从ObjectScript调用:

```
$SYSTEM.SQL.Functions.DATENAME(datepart,date-expression)
```

### Datepart 参数

datepart参数可以是包含一个(且仅包含一个)日期/时间组件的字符串,可以是全名(date Part列),也可以是缩写(缩写列)。  
这些datepart组件名称和缩写不区分大小写。

Date Part	Abbreviations	Return Values
year	yyyy, yy	0001-9999
quarter	qq, q	1-4
month	mm	January,...December
week	wk, ww	1-53
weekday	dw	Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday
dayofyear	dy, y	1-366
day	dd, d	1-31

Date Part	Abbreviations	Return Values
hour	hh	0-23
minute	mi, n	0-59
second	ss, s	0-59
millisecond	ms	0-999 (with precision of 3)
microsecond	mcs	0 – 999999 (with precision of 6)
nanosecond	ns	0 – 999999999 (with precision of 9)

如果将无效的datepart值指定为文字,则会发出SQLCODE -8错误码。

但是,如果提供一个无效的datepart值作为主机变量,则不会发出SQLCODE错误,并且DATENAME函数返回一个NULL值。

上表显示了不同日期部分的默认返回值。

通过使用带有不同时间和日期选项的SET OPTION命令,可以修改其中几个日期部分的返回值。

week:可以配置为使默认算法或ISO 8601标准算法确定给定日期的年度星期。

weekday:对于weekday的默认设置是将周日指定为一周的第一天(工作日=1)。

但是,可以将一周的第一天配置为另一个值,或者可以应用指定星期一为一周的第一天的ISO 8601标准。

millisecond:返回一个包含毫秒数(千分之一秒)的字符串。

如果日期表达式的精度超过3个小数位数,将其截断为3个数字,并将该数字作为字符串返回。

如果日期表达式具有指定的精度,但精度小于3个小数位数,则0将其填充为3个数字,并将该数字作为字符串返回。

微秒和纳秒执行类似的截断和填充零。

可以将datepart指定为带引号的字符串或不带引号的字符串。

这些语法变体的执行策略有不同:

- 引号:DATENAME('month', '2018-02-25'):在创建缓存查询时, datepart被视为一个字面值。SQL执行文字替换。这将产生一个更普遍的可重用的缓存查询。
- 没有引号:DATENAME(month, '2018-02-25'):在创建缓存查询时, datepart被视为关键字。没有文字替换。这将产生一个更具特异性缓存查询。

## 日期表达式格式

date-expression参数可以是以下任何一种格式:

- %Date logical value (+\$H)

- %PosixTime (%Library.PosixTime) logical value (an encoded 64-bit signed integer)
- %TimeStamp (%Library.TimeStamp) logical value (YYYY-MM-DD HH:MM:SS.FFF), also known as ODBC format.
- %String (or compatible) value  
%String(或compatible)值可以是以下任何一种格式:
- 99999,99999 (\$H format)
- Sybase/SQL-Server-date Sybase/SQL-Server-time

- Sybase/SQL-Server-time Sybase/SQL-Server-date
- Sybase/SQL-Server-date (default time is 00:00:00)
- Sybase/SQL-Server-time (default date is 01/01/1900)

Sybase/SQL-Server-date是以五种格式之一:

```
mmdelimiterddd delimiter[yy]yy dd Mmm[mm][,][yy]yy dd [yy]yy Mmm[mm] yyyy Mmm[mm] dd yy  
yy [dd] Mmm[mm]
```

其中分隔符是斜杠(/)、连字符(-)或句号(.)。

Sybase/SQL-Server-time表示以三种格式之一:

```
HH:MM[:SS:SSS][ {AM|PM} ] HH:MM[:SS.S] HH[ ' ' ] {AM|PM}
```

如果date-expression指定了时间格式但没有指定日期格式,则DATENAME的默认值为1900-01-01,其中weekday的值为Monday。

## 范围和值检查

DATENAME对输入值执行以检查。

如果一个值检查失败,则返回null字符串。

- 有效的日期表达式可以由日期字符串(yyyy-mm-dd)、时间字符串(hh:mm:ss)或日期和时间字符串(yyyy-mm-dd hh:mm:ss)组成。  
如果同时指定日期和时间,则日期和时间都必须有效。  
例如,如果没有指定时间字符串,则可以返回Year值,但如果指定了无效的时间字符串,则无法返回Year值。
- 日期字符串必须完整,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。  
例如,如果省略了Day值,则不能返回Year值。  
年必须指定为四位数字。
- 时间字符串必须使用适当的分隔符进行适当的格式化。  
因为时间值可以为零,所以可以省略一个或几个时间元素(或省略分隔符),这些元素将返回值为零。  
因此,"hh: mm: ss","hh: mm:","hh: mm","hh:: ss","hh::","hh",和"::"都是有效的。  
若要省略Hour元素,date-expression必须没有字符串的日期部分,并且必须至少包含一个分隔符(:)。
- 日期和时间值必须在有效范围内。  
年龄:0001到9999。  
月份:1 - 12个月。  
天数:1 - 31天。  
小时:0到23。  
分钟:0到59分钟。  
秒:0 ~ 59。
- 一个月中的天数必须与月和年相匹配。  
例如,日期"02-29"仅在指定的年份为闰年时有效。
- 大数小于10的日期和时间值可能包括或省略前导零。  
但是,小于10的Hour值必须包括前导0,如果是datetime字符串的一部分。

不允许使用其他非规范整数值。

因此，Day值为“07”或“7”是有效的，但“007”、“7.0”或“7a”无效。

- 如果date-expression指定了时间格式但没有指定日期格式，则DATENAME不会对时间组件值执行范围验证。

## 示例

在面的例子中，每个DATENAME返回'Wednesday'，因为它是指定日期的星期几('dw'):

```
SELECT DATENAME('dw', '2018-02-21') AS DayName,
       DATENAME(dw, '02/21/2018') AS DayName,
       DATENAME('DW', 64700) AS DayName
```

```
Wednesday    Wednesday    Wednesday
```

面的例子返回'December'，因为它是指定日期的月份名称('mm'):

```
SELECT DATENAME('mm', '2018-12-20 12:00:00') AS MonthName
```

```
December
```

面的示例返回'2018'(字符串形式)，因为它是指定日期的年份('yy'):

```
SELECT DATENAME('yy', '2018-12-20 12:00:00') AS Year
```

```
2018
```

注意，上面的例子使用了日期部分的缩写。

但是，你可以指定全名，如例所示:

```
SELECT DATENAME('Q', $HOROLOG) AS Q,
       DATENAME('WK', $HOROLOG) AS WkCnt,
       DATENAME('DY', $HOROLOG) AS DayCnt
```

```
1    6    35
```

面的嵌入式SQL示例将datepart和date-expression作为宿主变量传入:

```
ClassMethod DateName()
{
    s a="year"
    s b=$HOROLOG
    &sql(SELECT DATENAME(:a, :b) INTO :c)
    w "this year is: ", c
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLCommand).DateName()
this year is: 2022
```

下面示例使用子查询从出生日期为星期三的Sample.Person返回记录：

```
SELECT Name AS WednesdaysChild,DOB
FROM (SELECT Name,DOB,DATENAME('dw',DOB) AS Wkday FROM Sample.Person)
WHERE Wkday='Wednesday'
ORDER BY DOB
```

[#SQL #Caché](#)

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