

文章

姚鑫 · 二月 2, 2022 阅读大约需 6 分钟

## 第四十二章 SQL函数 DATEADD

## 第四十二章 SQL函数 DATEADD

一个日期/时间函数，它返回一个时间戳，计算方法是在一个日期或时间戳中添加或减去若干日期部件单位(如小时或天)。

### 大纲

`DATEADD(datepart, integer-exp, date-exp)`

### 参数

- `datepart` - 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。
- `integer-exp` - 任意数字类型的数字表达式。该值被截断为整数(正或负)。该值指示将添加到(或从)`date-exp`中减去的`datepart`单元的数量。
- `date-exp` - 要修改的日期/时间表达式。它可以是日期字符串，也可以是时间戳字符串(`%PosiTime`或`%Timestamp`数据类型)，也可以是`CURRENTDATE`之类的函数。返回的值始终是时间戳，数据类型格式为`%PosiTime`或`%Timestamp`。

### 描述

`DATEADD`函数通过将指定的日期部分递增指定的单元数来修改日期/时间表达式。

例如，如果`datepart`为“month”且整数-`exp`为5，则`DATEADD`将`date-exp`递增5个月。还可以通过为`integer-exp`指定一个负整数来减少日期部分。

计算出的日期将作为完整的日期/时间表达式(时间戳)返回。返回的数据类型取决于`Date-EXP`的数据类型。如果`Date-EXP`为`%Library.PosiTime`(编码的64位有符号整数)，则`DATEADD`返回数据类型`%Library.PosiTime`。否则，`DATEADD`返回数据类型`%Library.TimeStamp(yyyy-mm-dd hh:mm:ss.fff)`。

`DATEADD`始终返回有效日期，并考虑一个月的天数，并计算闰年。例如，将1月31日递增一个月将返回2月28日(该月中的最高有效日期)，除非指定的年份是闰年，在这种情况下将返回2月29日。将闰年日期2月29日递增一年将返回2月28日。将闰年日期2月29日递增四年返回2月29日。

如果指定包含小数秒的`date-exp`，则返回值也包括小数秒。如果省略`date-exp`的时间部分，`DATEADD`将返回默认时间00:00:00。如果省略`date-exp`的日期部分，`DATEADD`将返回默认日期1900-01-01。

`DATEADD`和`TIMESTAMPADD`处理季度(3个月间隔)；`DATEDIFF`和`TIMESTAMPDIFF`不处理季度。

可以使用`TIMESTAMPADD` ODBC标量函数执行类似的时间/日期修改操作。

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

\$SYSTEM.SQL.Functions.DATEADD(datepart, integer-exp, date-exp)

## Datepart Argument

日期部分参数可以是以下日期/时间组件之一：全名(日期部分列)或其缩写(缩写列)。这些日期部分组件名称和缩写不区分大小写。

Date Part	Abbreviations	integer-exp = 1
year	yyyy, yy	Increments year by 1.
quarter	qq, q	Increments month by 3.
month	mm, m	Increments month by 1.
week	wk, ww	Increments day by 7.
weekday	dw	Increments day by 1.
day	dd, d	Increments day by 1.
dayofyear	dy, y	Increments day by 1.
hour	hh	Increments hour by 1.
minute	mi, n	Increments minute by 1.
second	ss, s	Increments second by 1.
millisecond	ms	Increments by .001 of a second.
microsecond	mcs	0-999999 (with precision of 6)
nanosecond	ns	0-999999999 (with precision of 9)

递增或递减日期部分会导致适当修改其他日期部分。例如，午夜过后的小时递增会自动递增日期，这又可能会递增月份，依此类推。

日期部分可以指定为带引号的字符串或不带引号。这些语法变体执行的操作略有不同：

- QUOTES：DATEADD('month',12,\$HOROLOG)：在创建缓存查询时，日期部分被视为文字。SQL执行文字替换。这会产生更普遍可重用的缓存查询。
- 无引号：DATEADD(MONTH, 12, \$HOROLOG)：在创建缓存查询时，日期部分被视为关键字。没有文字替换。这会产生更具体的缓存查询。

如果将无效的日期部分值指定为文字，则会发出SQLCODE-8错误代码。但是，如果提供无效的日期部件值作为主机变量，则不会发出SQLCODE错误，并且DATEPART函数返回值为NULL。

## 日期表达式格式

Date-exp参数可以采用以下任何格式，并且可以包括或省略小数秒：

- %Date logical value (+\$H)
- %PosiTime(%Library.PosiTime)逻辑值(编码的64位有符号整数)
- (%Library.TimeStamp)逻辑值(YYYY-MM-DD HH : MM : SS)
- %String(或兼容)值

%STRING(或COMPATIBLE)值可以采用以下任何格式：

-

99999,99999 (\$H format)

- /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是以下五种格式之一：

```
mmdelimiterddddelimiter[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(-)或句点(.)。

Sybase/SQL-Server-Time表示以下三种格式之一：

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

请注意，提供DATEADD是为了与Sybase和Microsoft SQL Server兼容。

## 范围和值检查

DATEADD对输入值执行以下检查。如果值未通过检查，则返回空字符串。

- 日期字符串必须完整且格式正确，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。时间：0点到23点。分钟：0到59分钟。秒：0到59。
- 返回的递增的year值必须在0001到9999之间。超出此范围将返回<null>。
- 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。
- 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值为“07”或“7”是有效的，但“007”、“7.0”或“7a”无效。

下面的例子为指定的日期添加了1周:

```
SELECT DATEADD('week',1,'2018-02-26') AS NewDate
```

```
2018/3/5 0:00:00
```

它返回2018-03-05 00:00:00，因为增加1周会增加7天。

注意，DATEADD提供了省略的时间部分。

下面的例子为时间戳添加了5个月：

```
SELECT DATEADD(MM,5,'2017-11-26 12:00:00') AS NewDate
```

```
2018/4/26 12:00:00
```

它返回2018-04-26 12:00:00，因为增加5个月也会增加一年。

下面的例子也在时间戳上增加了5个月：

```
SELECT DATEADD('mm',5,'2018-01-31 12:00:00') AS NewDate
```

```
2018/6/30 12:00:00
```

它返回2018-06-30 12:00:00。

这里DATEADD修改了日值和月值，因为简单地增加月值将导致6月31日，这是一个无效的日期。

下面的例子为时间戳添加了45分钟：

```
SELECT DATEADD(MI,45,'2018-02-26 12:00:00') AS NewTime
```

```
2018/2/26 12:45:00
```

下面的示例还为时间戳添加了45分钟，但在本例中，添加的内容增加了日，从而增加了月：

```
SELECT DATEADD('mi',45,'2018-02-28 23:30:00') AS NewTime
```

```
2018/3/1 0:15:00
```

下面的例子将原始时间戳减去45分钟：

```
SELECT DATEADD(N,-45,'2018-01-01 00:10:00') AS NewTime
```

```
2017/12/31 23:25:00
```

下面的例子为当前日期添加了60天，并根据月份的不同长度进行调整：

```
SELECT DATEADD(D,60,CURRENT_DATE) AS NewDate
```

```
2022/4/4 0:00:00
```

在下面的例子中，第一个DATEADD为指定的日期添加了92天，第二个DATEADD为指定的日期添加了1/4天：

```
SELECT DATEADD('dd',92,'2018-12-20') AS NewDateD,  
       DATEADD('qq',1,'2018-12-20') AS NewDateQ
```

第一季将于2019-03-22 00:00:00回归;

第二季将于2019-03-20 00:00:00回归。

每增加1 / 4,month字段就会增加3, 如果需要, 还会增加year字段。  
它还校正给定月份的最大天数。

上面的例子都使用日期部分的缩写。

但是, 也可以用它的全名来指定日期部分, 就像下面的例子一样:

```
SELECT DATEADD('day',92,'2018-12-20') AS NewDate
```

```
2019/3/22 0:00:00
```

下面的嵌入式SQL示例使用主机变量来执行与前面示例相同的DATEADD操作:

```
ClassMethod DateAdd()  
{  
    s x="day"  
    s datein="2019-12-20"  
    &sql(SELECT DATEADD(:x,92,:datein)  
        INTO :dateout)  
    w "in: ",datein,!,"out: ",dateout  
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLCommand).DateAdd()  
in: 2019-12-20  
out: 2020-03-21 00:00:00  
DHC-APP>
```

[#SQL #Caché](#)

---

**源**

URL:

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