

文章
[姚鑫](#) · 二月 3 阅读大约需分钟

第四十三章 SQL函数 DATEDIFF

[toc]

第四十三章 SQL函数 DATEDIFF

日期/时间函数, 返回两个日期之间指定日期部分的整数差。

大纲

DATEDIFF (datepart , startdate , enddate)

参数

- datepart - 日期或时间部分的名称(或缩写)。这个名称可以用大写或小写来指定, 有或没有引号。datepart可以指定为文字或主机变量。
- startdate - 间隔的开始日期/时间。可以是种标准格式的日期、时间或日期时间。
- enddate - 间隔的结束日期/时间。可以是种标准格式的日期、时间或日期时间。从enddate中减去startdate, 以确定两个日期之间的日期部分间隔。

描述

DATEDIFF函数返回两个指定日期之间指定日期部分差的整数。日期范围从开始日期开始, 到结束日期结束。(如果enddate早于startdate, DATEDIFF将返回一个负整数值。)

DATEDIFF返回startdate和enddate之间指定单位的总数。例如, 两个日期时间值之间的分钟数计算日期部分和时间部分, 并为每一天的差异增加1440分钟。DATEDIFF返回开始日期和结束日期之间跨越的指定日期部分边界的计数。例如, 指定连续年份的任意两个日期(例如2018-09-23和2019-01-01)返回的年份DATEDIFF为1, 而不管这两个日期之间的实际持续时间是大于还是小于365天。同样, 12:23:59和12:24:05之间的分钟数是1, 尽管实际上只有6秒将两个值分开。

请注意, DATEDIFF是为Sybase和Microsoft SQL Server兼容而提供的。使用TIMESTAMPDIFF ODBC标量函数可以执行类似的时间/日期比较操作。

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

```
$SYSTEM.SQL.Functions.DATEDIFF (datepart , startdate , enddate )
```

为DATEDIFF()方法指定无效的datepart、startdate或enddate会生成ZDDIF >错误。

Datepart 参数

日期部分参数可以是列日期/时间组件之一，可以是全名(日期部分列)或其缩写(缩写列)。这些datepart组件名称和缩写不区分大小写。

Date Part	Abbreviations
year	YYYY, YY
month	mm, m
week	wk, ww
weekday	dw
day	dd, d
dayofyear	dy
hour	hh
minute	mi, n
second	ss, s
millisecond	ms
microsecond	mcs
nanosecond	ns

weekday和dayofyear datepart值在功能上与day datepart值相同。

DATEDIFF和TIMESTAMPDIFF不处理季度(间隔3个月)。

如果指定包含分数秒的开始日期和结束日期，DATEDIFF将以分数秒的整数形式返回差值，如示例所示：

```
SELECT DATEDIFF('ms', '64701,56670.10', '64701,56670.27'), /* returns 170 */
       DATEDIFF('ms', '64701,56670.1111', '64701,56670.27222') /* returns 161 */
```

datepart可以指定为带引号的字符串或不带引号的字符串。这些语法变体执行略有不同的操作：

- Quotes: DATEDIFF('month', '2018-02-25', \$HOROLOG): 在创建缓存查询时，datepart被视为文字。SQL执行文字替换。这将产生一个更容易重用的缓存查询。
- 无引号: DATEDIFF(month, '2018-02-25', \$HOROLOG): 创建缓存查询时，datepart被视为关键字。没有文字替换。这将生成具体缓存查询。

日期表达式格式

startdate和enddate参数可以采用不同的数据类型格式。

startdate和enddate参数可以采用以下格式：

- %Date逻辑值(+ \$H)，也称为\$HOROLOG格式。
- %PosixTime(%Library.PosixTime, 逻辑值(编码的64位有符号整数))
- %TimeStamp(%Library.TimeStamp)逻辑值(YYYY-MM-DD HH:MM:SS.FFF)，也称为ODBC格式。
- %String(或兼容)值。

%String(或compatible)值可以是以下任何一种格式，可以包含或省略小数秒：

- 99999, 99999 (\$HOROLOG格式)。
\$HOROLOG特殊变量不返回小数秒。
但是，可以使用\$HOROLOG格式指定一个包含分数秒的值: 99999,99999.999
- 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:

```
mm/dd/[yy]yy dd Mmm[mm][,][yy]yy dd [yy]yy Mmm[mm] yyyy Mmm[mm] dd yyyy [dd] Mmm[mm]
```

在第一种语法格式中,分隔符可以是斜杠(/)、连字符(-)或句点(.)。

Sybase/SQL-Server-time表示以三种格式之一:

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

Years

如果年份以两位数字表示,或者日期被完全省略,IRIS会检查滑动窗口来解释日期。系统范围内滑动窗口的默认值是1900;因此,在默认情况下,两位数的年份被认为是在20世纪。如示例所示:

```
SELECT DATEDIFF('year','10/11/14','02/22/2018'),  
       DATEDIFF('year','12:00:00','2018-02-22 12:00:00')
```

分数秒

DATEDIFF返回以毫秒(3位整数)、微秒(6位整数)或纳秒(9位整数)表示的小数秒,而不管startdate和enddate中的小数位数精度是秒。如示例所示:

```
SELECT DATEDIFF('ms','12:00:00.1','12:00:00.2'),  
       DATEDIFF('ms','12:00:00.10009','12:00:00.20007')
```

一些NLS区域设置将分数分隔符指定为逗号(欧洲的用法),而不是句号。如果当前区域设置是这些区域设置之一,DATEDIFF使用句号或逗号作为本地日期格式的小数秒分隔符。对于\$HOROLOG格式或ODBC格式的时间,不能使用逗号作为小数秒分隔符。尝试这样做会生成一个SQLCODE -8。无论当前的NLS语言环境是哪种,这两种格式都需要一段时间。

时间差异与时间格式无关

DATEDIFF返回以秒和毫秒为单位的时间差,即使当前进程的TimeFormat被设置为不返回秒。

如示例所示:

```
ClassMethod DateDiff()
{
    s tfmt = ##class(%SYS.NLS.Format).GetFormatItem("TimeFormat")
    d ##class(%SYS.NLS.Format).SetFormatItem("TimeFormat",1)
    w "datetime values (with seconds) are: ",!,
    $ZDATETIME("64701,56670.10",1,-1)," ",$ZDATETIME("64701,56673.27",1,-1),!
    &sql(SELECT DATEDIFF('ss','64701,56670.10','62871,56673.27') INTO :x)
    w "DATEDIFF number of seconds is: ",x,!
    d ##class(%SYS.NLS.Format).SetFormatItem("TimeFormat",2)
    w "datetime values (without seconds) are: ",!,
    $ZDATETIME("64701,56670.10",1,-1)," ",$ZDATETIME("64701,56673.27",1,-1),!
    &sql(SELECT DATEDIFF('ss','64701,56670.10','64701,56673.27') INTO :x)
    w "DATEDIFF number of seconds is: ",x,!
    d ##class(%SYS.NLS.Format).SetFormatItem("TimeFormat",tfmt)
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLCommand).DateDiff()
datetime values (with seconds) are:
02/22/2018 15:44:30 02/22/2018 15:44:33
DATEDIFF number of seconds is: -158111996.83
```

```
datetime values (without seconds) are:
02/22/2018 15:44 02/22/2018 15:44
DATEDIFF number of seconds is: 3.17
```

范围和值检查

DATEDIFF对输入值执行以下检查:

- 在执行任何DATEDIFF操作之前,开始日期和结束日期的所有指定部分必须是有效的。
- 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。
年必须指定为四位数字。
如果省略输入值的日期部分,DATEDIFF默认为'1900-01-01'。
无效的日期值将导致SQLCODE -8错误。
- 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。
营业时间:00至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。
无效的日期值将导致SQLCODE -8错误。
- 小于10(月和日)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。
- 时间值可以全部或部分省略。
如果startdate或enddate指定了一个不完整的时间,则为未指定的部分提供0。
- 小于10的小时值必须包含前导零。

省略前导零将导致SQLCODE -8错误。

错误处理

- 在Embedded SQL中,如果指定无效的datepart作为输入变量,则会发出SQLCODE -8错误码。如果将无效的日期部分指定为文字,则会发生<SYNTAX>错误。如果将无效的开始日期或结束日期指定为输入变量或文字,则会发出SQLCODE -8错误码。
- 在动态SQL中,如果您提供了无效的日期部分、开始日期或结束日期,则DATEDIFF函数将返回一个NULL值。
没有发出SQLCODE错误。

示例

下面的例子返回353,因为两个时间戳之间有353天(D):

```
SELECT DATEDIFF(D, '2018-01-01 00:00:00', '2018-12-20 12:00:00')
```

```
353
```

在下面的示例中,每个DATEDIFF返回1,因为日期的年份部分相差1。日期之间的实际持续时间不被考虑:

```
SELECT DATEDIFF('yyyy', '1910-08-21', '1911-08-21') AS ExactYear,  
       DATEDIFF('yyyy', '1910-06-30', '1911-01-01') AS HalfYear,  
       DATEDIFF('yyyy', '1910-01-01', '1911-12-31') AS Nearly2Years,  
       DATEDIFF('yyyy', '1910-12-31 11:59:59', '1911-01-01 00:00:00') AS NewYearSecond
```

```
1 1 1 1
```

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 DATEDIFF('year', '2017-09-10 13:19:00', '2018-12-20 00:00:00')
```

```
1
```

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

```
ClassMethod DateDiff1()  
{  
    s x="year"  
    s date1="2017-09-10 13:19:00"  
    s date2="2018-12-20 00:00:00"  
    &sql(SELECT DATEDIFF(:x, :date1, :date2)  
        INTO :diff)  
    w diff  
}
```

```
1
```

下面的例子使用WHERE子句中的DATEDIFF来选择上周入院的患者:

```
SELECT Name,DateOfAdmission FROM Sample.Patients WHERE DATEDIFF(D,DateOfAdmission,$HOROLOG) <= 7
```

下面的例子使用了一个子查询来返回那些个人的出生日期距当前日期不超过1500天的记录:

```
SELECT Name, Age, DOB
FROM (SELECT Name, Age, DOB, DATEDIFF('dy', DOB, $HOROLOG) AS DaysTo FROM Sample.Person)
WHERE DaysTo <= 1500
ORDER BY Age
```

[#SQL #Caché](#)

源 URL: <https://cn.community.intersystems.com/post/%E7%AC%AC%E5%9B%9B%E5%8D%81%E4%B8%89%E7%AB%A0-sql%E5%87%BD%E6%95%B0-datediff>