

文章

[姚鑫](#) · 三月 28 阅读大约需分钟

第九十五章 SQL函数 MINUTE

第九十五章 SQL函数 MINUTE

返回日期时间表达式的分钟的时间函数。

大纲

```
{fn MINUTE(time-expression)}
```

参数

- time-expression - 作为列名、另一个标量函数的结果或字符串或数字文字的表达式。它必须解析为日期时间字符串或时间整数，其中数据类型可以表示为 %Time、%TimeStamp 或 %PosixTime。

描述

MINUTE 返回一个整数，指定给定时间或日期时间值的分钟数。为 \$HOROLOGY 或 \$ZTIMESTAMP 值、ODBC 格式日期字符串或时间戳计算分钟。

时间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)。

要更改默认时间格式，请使用 SET OPTION 命令。

请注意，可以提供时间整数(经过的秒数)，但不能提供时间字符串 (hh:mm:ss)。必须提供日期时间字符串 (yyyy-mm-dd hh:mm:ss)。

日期时间字符串的时间部分必须是有效时间。否则，将生成 QLCODE -400 错误 <ILLEGAL VALUE>。分钟 (mm) 部分必须是 0 到

59 范围内的整数。前导零在输入时是可选的；前导零在输出上被抑制。可以省略日期时间字符串的秒 (:ss) 部分，但仍返回分钟部分。

日期时间字符串的日期部分未经过验证。

当分钟部分为“0”或“00”时，MINUTE

返回零分钟。如果没有提供时间表达式，或者完全省略时间表达式的分钟部分('hh', 'hh:', 'hh::', or 'hh::ss')，也会返回零分钟。

可以使用 DATEPART 或 DATENAME 返回相同的时间信息。

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

```
$$SYSTEM.SQL.Functions.MINUTE(time-expression)
```

示例

以下示例都返回数字 45, 因为它是 datetime 字符串中时间表达式中的 45 分钟:

```
SELECT {fn MINUTE('2018-02-16 18:45:38')} AS ODBCMinutes
```

45

```
SELECT {fn MINUTE(67538)} AS HorologMinutes
```

45

以下示例也返回 45。如此处所示, 时间值的秒部分可以省略:

```
SELECT {fn MINUTE('2018-02-16 18:45')} AS Minutes_Given
```

45

以下示例返回 0 分钟, 因为日期时间字符串中省略了时间表达式:

```
SELECT {fn MINUTE('2018-02-16')} AS Minutes_Given
```

0

以下示例均返回当前时间的分钟部分:

```
SELECT {fn MINUTE(CURRENT_TIME)} AS Min_CurrentT,  
       {fn MINUTE({fn CURTIME()})} AS Min_CurT,  
       {fn MINUTE({fn NOW()})} AS Min_Now,  
       {fn MINUTE($HOROLOG)} AS Min_Horolog,  
       {fn MINUTE($ZTIMESTAMP)} AS Min_ZTS
```

27 27 27 27 27

以下示例显示前导零被抑制。第一个 MINUTE 函数返回长度为 2, 其他函数返回长度为 1。省略的时间被认为是 0 分钟, 其长度为 1:

```
SELECT LENGTH({fn MINUTE('2018-02-22 11:45:00')}),  
       LENGTH({fn MINUTE('2018-02-22 03:05:00')}),  
       LENGTH({fn MINUTE('2018-02-22 3:5:0')}),  
       LENGTH({fn MINUTE('2018-02-22')})
```

2 1 1 1

以下嵌入式 SQL 示例显示 MINUTE 函数识别为区域设置指定的 TimeSeparator 字符:

```
/// d ##class(PHA.TEST.SQLFunction).Minute()  
ClassMethod Minute()  
{  
    d ##class(%SYS.NLS.Format).SetFormatItem("TimeSeparator", ".")  
    &sql(  
        SELECT {fn MINUTE('2018-02-22 18.45.38')}  
            INTO :a)  
    w "minutes=",a  
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).Minute()  
minutes=45
```

[#SQL #Caché](#)

源 URL: <https://cn.community.intersystems.com/post/%E7%AC%AC%E4%B9%9D%E5%8D%81%E4%BA%94%E7%AB%A0-sql%E5%87%BD%E6%95%B0-minute>