

文章

[姚鑫](#) · 二月 19, 2022 阅读大约需 4 分钟

第五十九章 SQL函数 GETDATE

第五十九章 SQL函数 GETDATE

日期/时间函数，返回当前本地日期和时间。

大纲

GETDATE([precision])

参数

- precision - 可选-一个正整数，指定时间精度为小数秒的位数。
默认值是0(没有小数秒);
这个默认值是可配置的。
精度值是可选的，括号是必选的。

描述

GETDATE将此时区的当前本地日期和时间作为时间戳返回；它根据本地时间变量(如夏令时)进行调整。

GETDATE可以返回%TIMESTAMP数据类型格式(yyyy-mm-dd hh:mm:ss.ffff)或%PosiTime数据类型格式(编码的64位有符号整数)的时间戳。以下规则确定返回哪种时间戳格式：

1. 如果当前时间戳被提供给数据类型为%PosiTime的字段，则当前时间戳值将以POSIXTIME数据类型格式返回。例如，WHERE PosixField=GETDATE() or INSERT INTO MyTable (PosixField) VALUES (GETDATE())
2. 如果当前时间戳被提供给数据类型为%TIMESTAMP的字段，则当前时间戳值将以TIMESTAMP数据类型格式返回。其ODBC类型为TIMESTAMP，长度为16，精度为19，例如 WHERE TSField=GETDATE() or INSERT INTO MyTable (TSField) VALUES (GETDATE())
3. 如果当前时间戳是在没有上下文的情况下提供的，则当前时间戳值以TIMESTAMP数据类型格式返回。例如，选择GETDATE()。

要更改默认日期时间字符串格式，请使用带有各种日期和时间选项的set option命令。

GETDATE可以在SELECT语句SELECT LIST或查询的WHERE子句中使用。在设计报表时，可以使用GETDATE在每次生成报表时打印当前日期和时间。GETDATE对于跟踪活动也很有用，比如记录事务发生的时间。

可以在CREATE TABLE中使用GETDATE指定字段的默认值。GETDATE是CURRENT_TIMESTAMP的同义词，提供GETDATE是为了与Sybase和Microsoft SQL Server兼容。

CURRENT_TIMESTAMP和NOW函数还可以用于以时间戳或POSIXTIME格式将当前本地日期和时间作为时间戳返回。CURRENT_TIMESTAMP支持精度，现在不支持精度。

要仅返回当前日期，请使用CURDATE或CURRENT_DATE。要仅返回当前时间，请使用CURRENT_TIME或CURRENT_TIME。这些函数使用日期或时间数据类型。这些函数都不支持精度。

TIMESTAMP数据类型以相同的格式存储和显示其值。POSIXTIME数据类型将其值存储为编码的64位有符号整数。时间和日期数据类型将它们的值存储为\$HOROLOG格式的整数。它们可以以显示格式或逻辑(存储)格式显示。可以使用CAST或CONVERT函数更改日期和时间的数据类型。

世界时(UTC)

GETDATE返回当前本地日期和时间。除GETUTCDATE之外，所有SQL时间戳、日期和时间函数都特定于本地时区设置。GETUTCDATE将当前UTC(通用)日期和时间作为时间戳值或POSIXTIME值返回。还可以使用ObjectScript \$ZTIMESTAMP特殊变量来获取通用的当前时间戳(独立于时区)。

精确到小数部分的秒

GETDATE可以返回多达9位的精度。

使用precision参数设置返回的精度的位数。

precision参数的默认值可以通过以下方式配置：

- 使用TIMEPRECISION选项设置OPTION。
- 系统范围的\$SYSTEM.SQL.Util.SetOption()方法配置选项DefaultTimePrecision。
要确定当前设置，调用\$SYSTEM.SQL.CurrentSettings()，它显示默认的时间精度；默认值为0。
- 进入管理门户，选择“系统管理”、“配置”、“SQL和对象设置”、“SQL”。
查看和编辑GETDATE()、CURRENT_TIME和CURRENT_TIMESTAMP的默认时间精度的当前设置。

指定从0到9(包括9)的整数，作为返回的十进制精度的默认位数。

默认值为0。

实际返回的精度取决于平台；

超过系统中可用精度的精度数字将作为零返回。

分数秒总是被截断，而不是四舍五入到指定的精度。

示例

下面的示例以TIMESTAMP格式返回当前日期和时间：

```
SELECT GETDATE() AS DateTime
```

```
2022/2/12 15:39:00
```

下面的示例以两位精度返回当前日期和时间：

```
SELECT GETDATE(2) AS DateTime
```

```
2022/2/12 15:39:21
```

下面的嵌入式SQL示例比较了本地(特定于时区)和通用(独立于时区)的时间戳:

```
ClassMethod GetDate()  
{  
    &sql(SELECT GETDATE(),GETUTCDATE() INTO :a,:b)  
    if SQLCODE'=0 {  
        w !,"Error code ",SQLCODE }  
    else {  
        w !,"Local timestamp is: ",a  
        w !,"UTC timestamp is:   ",b  
        w !,"$ZTIMESTAMP is:     ",$ZDATETIME($ZTIMESTAMP,3,,3)  
    }  
}
```

```
DHC-APP> d ##class(PHA.TEST.SQLCommand).GetDate()
```

```
Local timestamp is:  2022-02-12 15:40:34  
UTC timestamp is:    2022-02-12 07:40:34  
$ZTIMESTAMP is:      2022-02-12 07:40:34.978
```

下面的示例将Orders表中所选行中的LastUpdate字段设置为当前系统日期和时间。

如果LastUpdate是数据类型%TimeStamp, GETDATE返回当前日期和时间作为ODBC时间戳;

如果LastUpdate是数据类型%PosixTime, GETDATE返回当前日期和时间为编码的64位带符号整数:

```
UPDATE Orders SET LastUpdate = GETDATE()  
WHERE Orders.OrderNumber=:ord
```

在下面的例子中, CREATE TABLE语句使用GETDATE为StartDate字段设置一个默认值:

```
CREATE TABLE Employees(  
    EmpId      INT NOT NULL,  
    LastName    CHAR(40) NOT NULL,  
    FirstName   CHAR(20) NOT NULL,  
    StartDate   TIMESTAMP DEFAULT GETDATE())
```

[#AWS](#) [#SQL](#) [#Caché](#)

源

URL:

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