文章
姚 鑫 · 一月 28, 2022 阅读大约需 6 分钟

第三十七章 SQL函数 CURRENTTIMESTAMP

第三十七章 SQL函数 CURRENTTIMESTAMP

日期/时间函数，返回当前本地日期和时间。

大纲

```
CURRENT_TIMESTAMP
CURRENT_TIMESTAMP(precision)
```

参数

- precision - 一个正整数，它将时间精度指定为小数秒的位数。
  默认值是0(没有小数秒);
  这个默认值是可配置的。

CURRENTTIMESTAMP返回TIMESTAMP数据类型。

描述

CURRENTTIMESTAMP要么不接受参数，要么接受精度参数。
不允许使用空参数括号。

CURRENTTIMESTAMP返回当前时区的本地日期和时间;
它会根据当地的时间变化进行调整，例如日光节约时间。

CURRENTTIMESTAMP可以返回%timestamp数据类型格式(yyyy-mm-dd hh:mm:ss.ffff)或%PosixTime数据类型格式(编码的64位带符号整数)的时间戳。
以下规则决定返回哪种时间戳格式:

1. 如果将当前时间戳提供给数据类型为%PosixTime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=CURRENTTIMESTAMP or INSERT INTO MyTable (PosixField) VALUES (CURRENTTIMESTAMP).
2. 如果将当前的时间戳提供给数据类型为%timestamp的字段，则以timestamp数据类型格式返回当前的时间戳值。
   例如WHERE TSField=CURRENTTIMESTAMP or INSERT INTO MyTable (TSField) VALUES (CURRENTTIMESTAMP).
3. 如果没有上下文提供当前时间戳，则以timestamp数据类型格式返回当前时间戳值。

例如:SELECT CURRENTTIMESTAMP。

可以使用$HOROLOG以内部格式存储或返回当前本地日期和时间。

要更改默认的datetime字符串格式，使用SET OPTION命令和各种日期和时间选项。

当使用CREATE TABLE或ALTER
TABLE定义datetime字段时，可以指定CURRENTTIMESTAMP(带精度或不带精度)作为字段的默认值。
CURRENTTIMESTAMP可以被指定为数据类型为%Library的字段的默认值。
PosixTime或%Library.TimeStamp;
当前日期和时间以字段的数据类型指定的格式存储。

精确到小数部分的秒

CURRENTTIMESTAMP有两种语法形式:

- 如果没有参数括号，CURRENTTIMESTAMP在功能上与NOW相同。
  它使用系统范围内的默认时间精度。
- 带有圆括号的参数CURRENTTIMESTAMP(precision)在功能上与GETDATE相同，除了CURRENTTIMESTAMP()精度参数是强制性的。
  CURRENTTIMESTAMP()总是返回指定的精度，并忽略配置的系统范围内的缺省时间精度。

分数秒总是被截断，而不是四舍五入到指定的精度。

- 在TIMESTAMP数据类型格式中，精度的最大可能数字是9。
  实际支持的数字数由precision参数、配置的默认时间精度和系统能力决定。
  如果指定的精度大于配置的默认时间精度，则精度的其他数字将作为尾随零返回。
- 在POSIXTIME数据类型格式中，精度的最大可能数字为6。
  每个POSIXTIME值使用6位精度计算;
  除非提供，否则这些小数数字默认为零。
  实际支持的非零位数由precision参数、配置的缺省时间精度和系统能力决定。

配置精度

默认精度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()、CURRENTTIME和CURRENTTIMESTAMP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。

日期和时间函数比较

GETDATE和NOW也可用于返回当前本地日期和时间，作为TIMESTAMP数据类型或POSIXTIME数据类型值。

GETDATE支持精度，NOW不支持精度。

SYSDATE与CURRENTTIMESTAMP相同，只是SYSDATE不支持精度。
CURRENTTIMESTAMP是首选的SQL函数;
提供SYSDATE是为了与其他厂商兼容。

GETUTCDATE可以作为TIMESTAMP数据类型或POSIXTIME数据类型值返回通用的(与时区无关的)日期和时间。
请注意，除了GETUTCDATE之外，所有 SQL时间和日期函数都特定于当地时区设置。
要获得一个通用的(独立于时区的)时间戳，你可以使用GETUTCDATE或者ObjectScript $ZTIMESTAMP特殊变量。

要只返回当前本地日期，请使用CURDATE或CURRENTDATE。
要只返回当前本地时间，请使用CURRENTTIME或CURTIME。
这些函数返回DATE或TIME数据类型的值。
这些函数都不支持精度。

TIMESTAMP数据类型的存储格式和显示格式是相同的。
POSIXTIME数据类型存储格式是一个编码的64位有符号整数。
TIME和DATE数据类型将它们的值存储为$HOROLOG格式的整数;
当在SQL中显示时，它们被转换为日期或时间显示格式。
默认情况下，嵌入式SQL以逻辑(存储)格式返回。

可以使用CAST或CONVERT函数来更改日期和时间的数据类型。

示例

下面的例子以三种不同的方式返回当前本地日期和时间:TIMESTAMP数据类型格式，具有系统默认时间精度，具有两位小数秒的精度，以及$HOROLOG内部存储格式，具有全秒:

```
SELECT
   CURRENT_TIMESTAMP AS FullSecStamp,
   CURRENT_TIMESTAMP(2) AS FracSecStamp,
   $HOROLOG AS InternalFullSec
```

下面的嵌入式SQL示例设置了区域设置的缺省时间精度。
第一个CURRENTTIMESTAMP没有指定精度;
它返回带有默认时间精度的当前时间。
第二个CURRENTTIMESTAMP指定精度;
这将覆盖配置的缺省时间精度。
precision参数可以大于或小于默认的时间精度设置:

```
ClassMethod CurrentTimestamp()
{
InitialVal
    s pre = ##class(%SYS.NLS.Format).GetFormatItem("TimePrecision")
ChangeVal
    s x = ##class(%SYS.NLS.Format).SetFormatItem("TimePrecision",4)
    &sql(SELECT CURRENT_TIMESTAMP,CURRENT_TIMESTAMP(2) INTO :a,:b)
    if SQLCODE'=0 {
        w !,"Error code ",SQLCODE }
    else {
    w !,"Timestamp is:  ",a
    w !,"Timestamp is:  ",b }
RestoreVal
    s x = ##class(%SYS.NLS.Format).SetFormatItem("$TimePrecision",pre)
```

```
}


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

Timestamp is:  2022-01-25 09:22:49
Timestamp is:  2022-01-25 09:22:49.49
```

下面的嵌入式SQL示例比较了本地(特定时区)和通用(独立时区)的时间戳:

```
ClassMethod CurrentTimestamp1()
{
    &sql(SELECT CURRENT_TIMESTAMP,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).CurrentTimestamp1()

Local timestamp is:  2022-01-25 09:23:58
UTC timestamp is:    2022-01-25 01:23:58
$ZTIMESTAMP is:      2022-01-25 01:23:58.568
```

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, CURRENTTIMESTAMP将返回当前日期和时间作为ODBC时间戳;
如果LastUpdate是数据类型%PosixTime, CURRENTTIMESTAMP返回当前日期和时间为编码的64位有符号整数:

```
UPDATE Orders SET LastUpdate = CURRENT_TIMESTAMP
  WHERE Orders.OrderNumber=:ord
```

下面的例子创建了一个名为Orders的表，记录收到的产品订单:

```
CREATE TABLE Orders (
     OrderId     INT NOT NULL,
     ClientId    INT,
     ItemName    CHAR(40) NOT NULL,
     OrderDate   TIMESTAMP DEFAULT CURRENT_TIMESTAMP(3),
     PRIMARY KEY (OrderId))
```

OrderDate列包含收到订单的日期和时间。
它使用TIMESTAMP数据类型，并使用精度为3的CURRENTTIMESTAMP函数插入当前系统日期和时间作为默认值。

#SQL #Caché

源

URL:
https://cn.community.intersystems.com/post/%E7%AC%AC%E4%B8%89%E5%8D%81%E4%B8%83%E7%AB%A0-sql%E5%87%BD%E6%95%B0-currenttimestamp