

文章

[姚鑫](#) · 五月 27, 2022 阅读大约需 4 分钟

第154章 SQL函数 UNIX_TIMESTAMP

第154章 SQL函数 UNIX_TIMESTAMP

注：此函数IRIS可用，Cache不可用。

将日期表达式转换为 UNIX 时间戳的日期/时间函数。

大纲

UNIX_TIMESTAMP([date-expression])

参数

- date-expression - 可选 — 作为列名称、另一个标量函数的结果或日期或时间戳文字的表达式。UNIX_TIMESTAMP 不会从一个时区转换到另一个时区。如果省略 date-expression，则默认为当前 UTC 时间戳。

描述

UNIX_TIMESTAMP 返回一个 UNIX® 时间戳，即自 '1970-01-01 00:00:00' 以来的秒数（和小数秒）。

如果不指定 date-expression，则 date-expression 默认为当前 UTC 时间戳。因此，假设系统范围的默认精度为 3，UNIX_TIMESTAMP() 等效于 UNIX_TIMESTAMP(GETUTCDATE(3))。

如果指定 date-expression，UNIX_TIMESTAMP 会将指定的 date-expression 值转换为 UNIX 时间戳，计算该时间戳的秒数。UNIX_TIMESTAMP 可以返回正数或负数秒数。

UNIX_TIMESTAMP 将其值作为数据类型 %Library.Numeric 返回。它可以返回小数秒的精度。如果不指定 date-expression，它将采用当前配置的系统范围精度。如果指定 date-expression，它将从 date-expression 获取其精度。

日期表达式值

可选的日期表达式可以指定为：

- ODBC 时间戳值（数据类型 %Library.TimeStamp）：YYYY-MM-DD HH:MI:SS.FFF
- PosixTime 时间戳值（数据类型 %Library.PosixTime）：编码的 64 位有符号整数。
- \$HOROLOG 日期值（数据类型 %Library.Date）：自 1840 年 12 月 31 日以来的天数，其中第 1 天是 1841 年 1 月 1 日。
- \$HOROLOG 时间戳，带或不带小数秒：64412,54736。

UNIX_TIMESTAMP 不进行时区转换：如果 date-expression 为 UTC 时间，则返回 UTC UnixTime；如果 date-expression 是本地时间，则返回本地 UnixTime 值。

小数秒精度

小数秒总是被截断而不是四舍五入到指定的精度。

- %Library.TimeStamp 数据类型格式的日期表达式的最大精度为 9。支持的实际位数由日期表达式精度参数、配置的默认时间精度和系统功能确定。如果指定的精度大于配置的默认时间精度，则精度的附加数字将作为尾随零返回。
- %Library.PosixTime 数据类型格式的日期表达式的最大精度为六。每个 POSIXTIME 值都是使用六位精度计算的；除非提供，否则这些小数位默认为零。

配置精度

可以使用以下配置默认精度：

- 带有 TIMEPRECISION 选项的 SET OPTION。
- 系统范围的 \$SYSTEM.SQL.Util.SetOption() 方法配置选项 DefaultTimePrecision。要确定当前设置，请调用显示默认时间精度的 \$SYSTEM.SQL.CurrentSettings()；默认值为 0。
- 转到管理门户，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。

指定一个 0 到 9（含）整数作为要返回的默认精度的小数位数。默认值为 0。返回的实际精度取决于平台；超过系统上可用精度的精度数字将返回为零。

日期和时间函数比较

UNIX_TIMESTAMP 返回日期和时间，表示为从任意日期经过的秒数。

TO_POSIXTIME 返回一个编码的 64 位有符号（一个 %PosixTime 时间戳），它是根据 UNIX 时间戳计算的。

GETUTCDATE 以 %TimeStamp（ODBC 时间戳）数据类型或 %PosixTime（编码的 64 位带符号整数）数据类型值的形式返回通用（独立于时区）日期和时间。%PosixTime 值是根据相应的 UNIX 时间戳值计算得出的。%PosixTime 编码有助于快速的时间戳比较和计算。%Library.PosixTime 类提供 UnixTimeToLogical() 方法将 UNIX 时间戳转换为 PosixTime 时间戳，并提供 LogicalToUnixTime() 方法将 PosixTime 时间戳转换为 UNIX 时间戳。这些方法都不执行时区转换。

还可以使用 ObjectScript \$ZTIMESTAMP 特殊变量返回通用（时区无关）时间戳。

ObjectScript \$ZDATETIME 函数 dformat -2 采用 \$HOROLOG 日期并返回 UNIX 时间戳；\$ZDATETIMEH dformat -2 采用 UNIX 时间戳并返回 %HOROLOG 日期。这些 ObjectScript 函数将本地时间转换为 UTC 时间。UNIX_TIMESTAMP 不会将本地时间转换为 UTC 时间。

示例

以下示例返回 UTC UNIX 时间戳。第一个选择项采用日期表达式默认值，第二个指定显式 UTC 时间戳：

```
SELECT
    UNIX_TIMESTAMP() AS DefaultUTC,
    UNIX_TIMESTAMP(GETUTCDATE(3)) AS ExplicitUTC
```

```
1652164931  1652164931
```

以下示例返回当前本地日期和时间的本地 UNIX 时间戳，以及 UTC 日期和时间值的 UTC UNIX

时间戳。第一个选择项指定本地 CURRENT_TIMESTAMP，第二个指定 \$HOROLOG（本地日期和时间），第三个指定当前 UTC 日期和时间：

```
SELECT
  UNIX_TIMESTAMP(CURRENT_TIMESTAMP(2)) AS CurrTSLocal,
  UNIX_TIMESTAMP($HOROLOG) AS HorologLocal,
  UNIX_TIMESTAMP(GETUTCDATE(3)) AS ExplicitUTC
```

```
1652193775  1652193775  1652164975
```

以下示例比较了 UNIX_TIMESTAMP（不转换本地时间）和 \$ZDATETIME（转换本地时间）：

```
/// d ##class(PHA.TEST.SQLFunction).UCase()
ClassMethod UCase()
{
  s unixutc = $ZDATETIME($HOROLOG, -2)
  s myquery = "SELECT UNIX_TIMESTAMP($HOROLOG) AS UnixLocal, ? AS UnixUTC"
  s tStatement = ##class(%SQL.Statement).%New()
  s qStatus = tStatement.%Prepare(myquery)
  s rset = tStatement.%Execute(unixutc)
  d rset.%Display()
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).UCase()
UnixLocal  UnixUTC
1652193839  1652165039
```

```
1 Rows(s) Affected
```

[#SQL #Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC154%E7%AB%A0-sql%E5%87%BD%E6%95%B0-unix-timestamp>