文章

姚 鑫 · 五月 18, 2022 阅读大约需 10 分钟

第146章 SQL函数 TOPOSIXTIME 第146章 SQL函数 TOPOSIXTIME

注:此函数在IRIS版本可用, Cache不可用。

将格式化的日期字符串转换为 %PosixTime 时间戳的日期/时间函数。

大纲

TO_POSIXTIME(date_string[,format])

参数

- datestring 要转换为 %PosixTime 时间戳的字符串表达式。此表达式可能包含日期值、时间值或日期和时间值。
- format 可选 对应于 datestring 的日期和时间格式字符串。如果省略,则默认为 DD MON YYYY HH:MI:SS。

描述

TOPOSIXTIME 函数将各种格式的日期和时间字符串转换为 %PosixTime 时间戳,数据类型为 %Library.PosixTime。 TOPOSIXTIME 返回 %PosixTime 时间戳作为计算值,该值基于从 1970-01-01 00:00:00 的任意起点经过的秒数,编码为 64 位有符号整数。从该日期开始经过的实际秒数(和小数秒)是 Unix®timestamp,一个数值。 对 Unix® 时间戳进行编码以生成 %PosixTime 时间戳。由于 %PosixTime 时间戳值已编码,因此 1970-01-01 00:00:00 表示为 1152921504606846976。1970-01-01 00:00:00 之前的日期具有负整数值。

TOPOSIXTIME 不转换时区;本地日期和时间转换为本地 %PosixTime 时间戳; UTC 日期和时间将转换为 UTC %PosixTime 时间戳。

%PosixTime 支持的最早日期为 0001-01-01 00:00:00, 其逻辑值为 -6979664624441081856。支持的最后日期为 9999-12-31 23:59:59.999999,其逻辑值为 1406323805406846975。这些限制对应于 ODBC 日期格式显示限制。可以使用 %Library.PosixTime MINVAL 和 MAXVAL 参数进一步限制这些值。可以使用 IsValid() 方法来确定数值是否为有效的 %PosixTime 值。

%PosixTime 值始终编码小数秒的6 位十进制数字的精度。精度位数较少的 datestring 在 %PosixTime 转换之前被零填充到 6 位;在 %PosixTime 转换之前,精度超过 6 位的 datestring 会被截断为 6 位。

如果 datestring 省略了时间戳的组成部分,则 TOPOSIXTIME 提供缺少的组成部分。如果 datestring 和 format 都省略了年份,则 yyyy 默认为当前年份;如果只有 datestring 省略了年份,则默认为 00,根据年份格式元素扩展为四位数年份。如果省略日或月值,则 dd 默认为 01; mm-dd 默认为 01-01。缺少的时间组件默认为 00。支持小数秒,但必须明确指定;默认情况下不提供小数秒。

TOPOSIXTIME 支持将两位数年份转换为四位数。 TOPOSIXTIME 支持将 12 小时制时间转换为 24 小时制时间。它提供日期和时间元素值的范围验证,包括闰年验证。范围验证违规会生成 SQLCODE -400 错误。

Published on InterSystems Developer Community (https://community.intersystems.com)

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

\$SYSTEM.SQL.Functions.TOPOSIXTIME(date_string, format)

TOPOSIXTIME 函数可在为字段提供默认值时用于数据定义。例如:

CREATE TABLE mytest
(ID NUMBER(12,0) NOT NULL,
End_Year DATE DEFAULT TO_POSIXTIME('12-31-2018','MM-DD-YYYY') NOT NULL)

TOPOSIXTIME 可以与 CREATE TABLE 或 ALTER TABLE ADD COLUMN 语句一起使用。在此上下文中只能使用 datestring 的文字值。

%PosixTime 表示

%PosixTime 对小数秒的 6位精度进行编码,无论 datestring 的精度如何。 ODBC 和显示模式截断尾随零的精度。

- 逻辑模式:编码的64位(19个字符)有符号整数。
- ODBC 模式: YYYY-MM-DD HH:MM:SS.FFFFFF。
- 显示模式:使用当前语言环境的默认日期/时间格式 (dformat -1 和 tformat -1),如 \$ZDATETIME 中所述。

相关 SQL 函数

- TOPOSIXTIME 将格式化的日期和时间字符串转换为 %PosixTime 时间戳。
- TOCHAR 执行相反的操作;它将 %PosixTime 时间戳转换为格式化的日期和时间字符串。
- UNIXTIMESTAMP 将格式化的日期和时间字符串转换为 Unix® 时间戳。
- TODATE 将格式化的日期字符串转换为日期整数。
- CAST 和 CONVERT 执行 %PosixTime 数据类型转换。

日期和时间字符串

datestring 参数指定日期和时间字符串文字。如果提供没有时间分量的日期字符串,则 TOPOSIXTIME 提供时间值 00:00:00。如果提供不带日期组件的时间字符串,则 TOPOSIXTIME 提供当年的 01-01(1 月 1日)日期。

可以为输入 datestring 提供任何类型的日期和时间字符串。每个 datestring 字符必须对应于格式字符串,但以下情况除外:

- 可以包含或省略前导零(不带分隔符的 datestring 除外)。
- 年份可以用两位数或四位数字指定。
 - 月份缩写(采用 MON 格式)必须与该区域设置的月份缩写相匹配。对于某些语言环境,月份缩写可能不是月份名称的初始连续字符。月份缩写不区分大小写。
- 月份名称(格式为 MONTH)应指定为完整的月份名称。但是,TOPOSIXTIME 不需要格式为 MONTH 的 完整月份名称;它接受完整月份名称的初始字符,并选择月份列表中与该初始字母序列相对应的第一个月。

Published on InterSystems Developer Community (https://community.intersystems.com)

可以使用为语言环境定义的时间分隔符输入时间值。输出时间戳始终表示带有 ODBC 标准时间分隔符的时间值:冒号 (:) 和句点 (.))。省略的时间元素默认为零。

格式化

格式是根据以下规则指定的一个或多个格式元素的字符串:

- 格式元素不区分大小写。
- 几乎任何顺序或数量的格式元素都是允许的。
- 格式字符串使用与 datestring 中的分隔符匹配的非字母数字分隔符(例如,空格、斜杠或连字符)分隔它们的元素。这些分隔符不会出现在使用标准时间戳分隔符的输出字符串中:连字符表示日期值,冒号表示时间值,句点(如果需要)表示小数秒。这种分隔符的使用不依赖于为 NLS 语言环境定义的 DateSeparator。
- 以下日期格式字符串不需要分隔符: MMDDYYYY、DDMMYYYY、YYYYMMDDHHMISS、YYYYMMDDHH MI、YYYYMMDDHH、YYYYMMDD、YYYYDDMM、HHMISS 和 HHMI。还支持不完整的日期格式 YYYYMM,并假定 DD 值为 01。请注意,在这些情况下,必须为所有元素(例如 MM 和 DD)提供前导零,但最后一个元素除外。
- 格式中不是有效格式元素的字符将被忽略。

格式元素

下表列出了 format 参数的有效日期格式元素:	
Element	Meaning
DD	两位数的月份日期
	(01-31)。不需要前导零,除非格式不包含日期分隔符。
MM	两位数的月份编号(01-12;01 = 一月)。除非格式不包
	含日期分隔符,否则不需要前导零。在日语和中文中,月
	份数由一个数字组成,后跟"月份"的表意文字。
MON	月份的缩写名称,由当前语言环境中的 MonthAbbr 属性指
	定。默认情况下,在英文中,这是月份名称的前三个字母
	。在其他语言环境中,月份缩写可能超过三个字母长和/或
	可能不包含月份名称的第一个字母。不允许使用句点字符
	。不区分大小写。
MONTH	月份的全名,由当前语言环境中的 MonthName
	属性指定。不区分大小写。
YYYY	四位数年份。
YY	年份的最后两位数。 YY 2 位数年份的前 2 位数默认为
	19。
RR / RRRR	两位数年份到四位数年份的转换。 (见下文。)
DDD	一年中的一天。自 1 月 1 日以来的天数。(见下文。)
HH	小时,指定为 01–12 或
	00-23,具体取决于是否指定了子午线指示符(AM 或
	PM)。可以指定为 HH12 或 HH24。
MI	分钟,指定为 00–59。
SS	其次,指定为 00-59。
FF	一秒钟的分数。 FF 表示提供一个或多个小数位;
	date <u>s</u> tring 可以指定任意数量的小数位数。
	TOPOSIXTIME 准确返回六位精度,无论 date <u>s</u> tring
	中提供的精度如何。
AM / PM	子午线指示器,指定 12 小时制。 (见下文。)
A.M. / P.M.	子午线指示器(带句点)指定 12 小时制。(见下文。)

TOPOSIXTIME 格式还可以包含 D(星期几号)、DY(星期几缩写)或 DAY(星期几名称)元素以匹配输入

datestring。但是,这些格式元素未经过验证或用于确定返回值。

两位数年份转换(RR 和 RRRR 格式)

RR 格式提供两位数到四位数的年份转换。 TOPOSIXTIME 使用默认日期格式 (dformat -1) 执行此转换,该格式使用当前语言环境的 YearOption 属性,如 \$ZDATETIME 中所述。

一年中的某一天 (DDD 格式)

可以使用 DDD 将一年中的某一天(自 1 月 1 日以来经过的天数)转换为实际日期。格式字符串 DDD YYYY 必须与由整数天数和四位数年份组成的相应 datestring 配对。(与 DDD 一起使用时,两位数的年份必须指定为 RR(而不是 YY)。)格式字符串 DDD 默认为当前年份。经过的天数必须是 1 到 365 范围内的正整数(如果 YYYY 是闰年,则为 366)。四位数年份必须在标准 日期范围内:1841 到 9999。(如果省略年份,则默认为当前年份。)DDD 和年份(YYYY、RRRR 或 RR)格式元素可以是以任何顺序 指明;它们之间的分隔符是强制性的;此分隔符可以是空格。以下示例显示了这一年中的一天的用法:

SELECT TO_POSIXTIME('2018:160','YYYY:DDD')

2018-06-09 00:00:00

如果格式字符串同时包含 DD 和 DDD 元素,则 DDD 元素占主导地位。这在以下示例中显示,它返回 2008-02-29 00:00:00 (不是 2008-12-31 00:00:00) :

SELECT TO_POSIXTIME('2018-12-31-60','YYYY-MM-DD-DDD')

2018-03-01 00:00:00

TOPOSIXTIME 允许返回对应于一年中某一天的日期表达式。 TOCHAR 允许返回与日期表达式对应的一年中的哪一天。

1970 年之前的日期

TOPOSIXTIME 将 1970 年 1 月 1 日之前的日期表示为负数。 %PosixTime 不能表示 0001 年 1 月 1 日之前或 9999 年 12 月 31 日之后的日期。尝试输入这样的日期会导致 SQLCODE -400 错误。 TODATE 函数提供儒略日期格式来表示 0001 年 1 月 1 日之前的 BCE 日期。儒略日期转换将七位内部正整数值(儒略日计数)转换为显示格式或 ODBC 格式日期。儒略日期不支持时间值。

12 小时制时间

%PosixTime 时间戳始终表示使用 24 小时制的时间。 datestring 可以使用 12 小时制或 24 小时制表示时间。 TOPOSIXTIME 假定为 24 小时制,除非以下情况之一适用:

- datestring 时间值后跟 " am " 或 " pm " (没有句点)。这些子午线指标不区分大小写,可以附加到时间值后,也可以用一个或多个空格分隔。
- 格式遵循带有"a.m." 'p.m.'元素(任意一个),与时间格式之间用一个或多个空格分隔。例如:DD MON YYYY HH:MI:SS.FF P.M.此格式支持 12 小时制 datestring 值,例如 2:23pm、2:23:54.6pm、2:23:54 pm、2:23:54 (假定为

AM)。经络指标不区分大小写。当使用带有句点的经络指标时,它必须与时间值隔开一个或多个空格。

示例

以下嵌入式 SQL 示例将当前本地日期时间转换为 %PosixTime 值。 (请注意,格式使用"ff"表示任意数量的小数位;在这种情况下,精度为 3 位。%PosixTime 将其编码为 6 位精度,提供三个尾随零。)然后此示例使用 %Posix LogicalToOdbc() 方法将此值显示为 ODBC 时间戳,修剪尾随零精度:

```
/// d ##class(PHA.TEST.SQLFunction).ToPosixtime()
ClassMethod ToPosixtime()
    s tstime = $ZDATETIME($ZTIMESTAMP,3,1,3)
    w "local datetime in : ",tstime,!
    %sql(
        SELECT
            TO_POSIXTIME(:tstime,'yyyy-mm-dd hh:mi:ss.ff')
        TNTO
            :ptime
    if SQLCODE=0 {
        w "Posix encoded datetime: ",ptime,!
        s ODBCout=##class(%PosixTime).LogicalToOdbc(ptime)
        w "local datetime out: ",ODBCout
    } else {
        w "SQLCODE error: ", SQLCODE
    }
}
DHC-APP>d ##class(PHA.TEST.SQLFunction).ToPosixtime()
local datetime in : 2022-05-10 02:00:10.057
Posix encoded datetime: 1154573652616903976
local datetime out: 2022-05-10 02:00:10.057
```

以下嵌入式 SQL 示例以各种格式指定日期字符串。第一个使用默认格式,其他指定格式。所有这些都将 datestring 转换为 2018-06-29 00:00:00 的时间戳值:

以下示例指定 YYYYMM 日期格式。它不需要元素分隔符。 TOPOSIXTIME 提供缺失的日期和时间值:

```
SELECT TO_POSIXTIME('201806','YYYYYMM')
2018-06-01 00:00:00
```

以下示例仅指定 HH:MI:SS.FF 时间格式。 TOPOSIXTIME 提供缺失的日期值。在每种情况下,此示例都返回 2018-01-01 的日期(其中 2018 是当前年份):

请注意,小数秒完全按照指定传递,没有填充或截断。

#SQL #Caché

源

URL:

https://cn.community.intersystems.com/post/%E7%AC%AC146%E7%AB%A0-sql%E5%87%BD%E6%95%B0-toposixtime