文章
姚 鑫 · 五月 19阅读大约需0 分钟

第147章 SQL函数 TO_TIMESTAMP

第147章 SQL函数 TO_TIMESTAMP

将格式化字符串转换为时间戳的日期函数。

大纲

```
TO_TIMESTAMP(date_string[,format])
```

参数

- date_string - 要转换为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 date_string 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

TO_TIMESTAMP 函数将各种格式的日期和时间字符串转换为标准时间戳，数据类型为 TIMESTAMP。
TO_TIMESTAMP 返回具有以下格式的时间戳：

```
yyyy-mm-dd hh:mm:ss
```

始终包括前导零。时间使用 24 小时制指定。默认情况下，返回的时间戳不包括小数秒。

注 意：TO_TIMESTAMP 以 ODBC 格式返回标准时间戳。TO_POSIXTIME 返回一个编码的 64 位时间戳。
TO_POSIXTIME 是新编程的推荐时间戳格式。

必须指定匹配的 date_string 和格式。如果省略格式，则 date_string 必须匹配 DD MON YYYY HH:MI:SS。

如果 date_string 省略了时间戳的组成部分，则 TO_TIMESTAMP 提供缺少的组成部分。如果 date_string 和 format 都省略了年份，则 yyyy 默认为当前年份；如果只有 date_string 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。因此，如果 date_string 和 format 都省略了时间戳的日期部分，则 TO_TIMESTAMP 默认为当年的 1 月 1 日，采用 ODBC 格式：yyyy-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。

TO_TIMESTAMP 支持将两位数年份转换为四位数。TO_TIMESTAMP 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。

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

```
$SYSTEM.SQL.Functions.TOTIMESTAMP(date_string,format)
```

在为时间戳字段提供默认值时，可以在数据定义中使用 TO_TIMESTAMP 函数。例如：

```
CREATE TABLE Sample.MyEmpReviews
(EmpNum INTEGER UNIQUE NOT NULL,
 ReviewDate TIMESTAMP DEFAULT TO_TIMESTAMP(365,'DDD'))
```

在此示例中，插入记录的用户可以提供 ReviewDate 值，不提供 ReviewDate 值并获取当年第 365 天的默认时间戳，或者提供 NULL 的 ReviewDate 并获取 NULL。

TO_TIMESTAMP 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 date_string 的文字值。

相关 SQL 函数

- TO_TIMESTAMP 将格式化的日期和时间字符串转换为标准时间戳。
- TO_CHAR 执行相反的操作；它将标准时间戳转换为格式化的日期和时间字符串。
- TO_DATE 将格式化的日期字符串转换为日期整数。
- CAST 和 CONVERT 执行 TIMESTAMP 数据类型转换。

日期和时间字符串

date_string 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 TO_TIMESTAMP 提供时间值 00:00:00。如果您提供不带日期组件的时间字符串，则 TO_TIMESTAMP 提供当年 01 – 01（1 月 1 日）的日期。

可以为输入 date_string 提供任何类型的日期和时间字符串。每个 date_string 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 date_string 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，TO_TIMESTAMP 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。因此，在英语中，“J” = “January”，“Ju” = “June”，“Jul” = “July”。指定的所有字符必须与完整月份名称的连续字符匹配；不检查完整月份名称之外的字符。例如，“Fe”、“Febru”和“FebruaryLeap”都是有效值；“Febs”不是有效值。月份名称不区分大小写。
- 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 (:) 表示小时、分钟和秒，句点 (.) 表示小数秒。省略的时间元素默认为零。默认情况下，返回的时间戳不带小数秒。

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 date_string 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您的 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、YYYYMMDDHHMI、Y

YYYMMDDHH、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
DD
MM
MON
MONTH
YYYY
YY
RR / RRRR
DDD
HH
MI
SS
FF

Element
AM / PM
A.M. / P.M.

TO_TIMESTAMP 格式还可以包括 D（星期几号）、DY（星期几缩写）或 DAY（星期几名称）元素以匹配输入的 date_string。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。此转换基于当年。如果当前年份在上半世纪（例如，2000 年到 2050 年），则从 00 到 49 的两位数年份扩展到当前世纪的四位数年份，从 50 到 2 位数的年份99 年扩大到上个世纪的四位数年份。如果当前年份在世纪下半叶（例如，2050 年到 2099 年），则所有两位数年份都将扩展为当前世纪中的四位数年份。将两位数年份扩展到四位数年份如下例所示：

```
SELECT TO_TIMESTAMP('29 September 00','DD MONTH RR'),
       TO_TIMESTAMP('29 September 18','DD MONTH RR'),
       TO_TIMESTAMP('29 September 49','DD MONTH RR'),
       TO_TIMESTAMP('29 September 50','DD MONTH RR'),
       TO_TIMESTAMP('29 September 77','DD MONTH RR')

2000/9/29 0:00:00   2018/9/29 0:00:00   2049/9/29 0:00:00   1950/9/29 0:00:00   1977/
9/29 0:00:00
```

RRRR 格式允许您输入两位数和四位数字的混合年份。四位数年份不变（与 YYYY 相同）。使用 RR 格式算法将两位数年份转换为四位数年份。这在以下示例中显示：

```
SELECT TO_TIMESTAMP('29 September 2018','DD MONTH RRRR')AS FourDigit,
       TO_TIMESTAMP('29 September 18','DD MONTH RRRR') AS TwoDigit,
       TO_TIMESTAMP('29 September 1949','DD MONTH RRRR') AS FourDigit,
       TO_TIMESTAMP('29 September 49','DD MONTH RRRR') AS TwoDigit,
       TO_TIMESTAMP('29 September 1950','DD MONTH RRRR') AS FourDigit,
       TO_TIMESTAMP('29 September 50','DD MONTH RRRR') AS TwoDigit

2018/9/29 0:00:00   2018/9/29 0:00:00   1949/9/29 0:00:00   2049/9/29 0:00:00   1950/
9/29 0:00:00   1950/9/29 0:00:00
```

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 date_string 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）。四位数年份必须在年份日期范围内：0001 到

9999。(如果省略年份，则默认为当前年份。) DDD 和年份(YYYY、RRRR 或 RR)格式元素可以在任何命令；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_TIMESTAMP('2018:160','YYYY:DDD')

2018/6/9 0:00:00
```

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

2018/3/1 0:00:00
```

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

第一年之前的日期

TO_TIMESTAMP 和 TO_POSIXTIME 可以表示追溯到 0001 年 1 月 1 日的日期。

TO_DATE 提供儒略日期格式，它可以表示追溯到公元前 4712 年 1 月 1 日的日期。儒略日期转换将其内部正整数值(儒略日计数)转换为显示格式或 ODBC 格式的日期。儒略日期不支持时间值。

12 小时制时间

- date_string 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。
- 该格式遵循具有'a.m.'或'p.m.'元素(任一个)的时间格式，与时间格式分开一个或多个空格。例如：DD-MON-YYYY-HH:MI:SS。FF P.M.此格式支持12小时时钟日期串值，例如下午2:23,2:23:54.6pm，下午2:23:54，下午2:23:54和下午2:23:54(假设为上午)。子午线指标不区分大小写。当使用带有日期的子午线指示器时，必须将其与时间值分开一个或多个空格。

示例

以下嵌入式 SQL 示例以多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 date_string 转换为 2018 – 06 – 29 00:00:00 的时间戳值：

```
/// d ##class(PHA.TEST.SQLFunction).ToTimestamp()
ClassMethod ToTimestamp()
{
    &sql(
        SELECT
            TO_TIMESTAMP('29 JUN 2018'),
            TO_TIMESTAMP('2018 Jun 29','YYYY MON DD'),
            TO_TIMESTAMP('JUNE 29, 2018','month dd, YYYY'),
            TO_TIMESTAMP('2018***06***29','YYYY***MM***DD'),
            TO_TIMESTAMP('06/29/2018','MM/DD/YYYY'),
            TO_TIMESTAMP('29/6/2018','DD/MM/YYYY')
        INTO :a,:b,:c,:d,:e,:f
```

```
    )
    if SQLCODE = 0 {
        w !,a,!,b,!,c,!,d,!,e,!,f
    } else {
        w "SQLCODE error:",SQLCODE
    }
}


DHC-APP>d ##class(PHA.TEST.SQLFunction).ToTimestamp()

2018-06-29 00:00:00
2018-06-29 00:00:00
2018-06-29 00:00:00
2018-06-29 00:00:00
2018-06-29 00:00:00
2018-06-29 00:00:00
```

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

```
 SELECT TO_TIMESTAMP('201806','YYYYMM')

 2018/6/1 0:00:00
```

以示例仅指定 HH:MI:SS.FF 时间格式。TO_TIMESTAMP 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）：

```
SELECT TO_TIMESTAMP('11:34','HH:MI:SS.FF'),
       TO_TIMESTAMP('11:34:22','HH:MI:SS.FF'),
       TO_TIMESTAMP('11:34:22.00','HH:MI:SS.FF'),
       TO_TIMESTAMP('11:34:22.7','HH:MI:SS.FF'),
       TO_TIMESTAMP('11:34:22.7000000','HH:MI:SS.FF')


2022/1/1 11:34:00   2022/1/1 11:34:22   2022/1/1 11:34:22   2022/1/1 11:34:22   2022/
1/1 11:34:22
```

请 注 意，小数秒完全按照指定传递，没有填充或截断。

以示例显示了一些其他方法来指定带小数秒的时间格式：

```
SELECT TO_TIMESTAMP('113422.9678','HHMISS.FF'),
       TO_TIMESTAMP('9678.113422','FF.HHMISS'),
       TO_TIMESTAMP('9678.20170804113422','FF.YYYYMMDDHHMISS')

2022/1/1 11:34:22   2022/1/1 11:34:22   2017/8/4 11:34:22
```

TO_TIMESTAMP 的所有三个调用都返回一个 ODBC 格式的时间戳，其时间部分的值为 11:34:22.9678。对于前两个，省略的日期部分默认为当年的 1 月 1 日；第三个提供日期部分值。

#SQL #Caché

源 URL: https://cn.community.intersystems.com/post/%E7%AC%AC147%E7%AB%A0-sql%E5%87%BD%E6%95%B0-totimestamp