
文章

[姚鑫](#) · 五月 12, 2022 阅读大约需 9 分钟

第140章 SQL函数 TOCHAR (一)

第140章 SQL函数 TOCHAR (一)

将日期、时间戳或数字转换为格式化字符串的字符串函数。

大纲

`TO_CHAR(tochar-expression[,format])`

`TOCHAR(tochar-expression[,format])`

参数

- tochar-expression - 要转换的逻辑日期、时间戳或数字表达式。
- format - 可选 — 为 tochar 表达式转换指定日期、时间戳或数字格式的字符代码。如果省略，TOCHAR 将 tochar-expression 作为规范数字返回。

描述

名称 TOCHAR 和 TOCHAR 是可互换的，并且支持 Oracle 兼容性。

带格式的 TOCHAR 函数有五种用途：

- 将日期整数转换为格式化的日期字符串。

- 日期在1840年之前转换为朱利安日期int
- 时间的整数转换为格式化时间生态
- 将日期和时间格式化的datetime
- 将数字转换成一个格式化的数字字符串。

这个函数也可以从ObjectScrip调用

```
$SYSTEM.SQL.Functions.TOCHAR(tochar-expression,format)
```

有效和无效的参数

- 要将 tochar-expression 解释为时间戳，它的格式必须为 YYYY-MM-DD HH:MI:SS，或以下有效变体之一：小于 10 的月份和日期值可以包含或省略前导零；如果前导零被省略，则在返回的日期中也会被省略。可以省略秒值，但必须指定指示其位置的冒号 (HH:MI:)；在返回的时间中，秒默认为 00。秒值可能包括小数秒 (HH:MM:SS.nnn)；在返回的时间内，这些小数秒被截断。时间戳必须包含时间部分，即使格式未指定时间格式。
- 如果 tochar-expression 不是有效的时间戳格式，则 TOCHAR 将其解释为整数，并在遇到第一个非整数字符时结束解释。如果 format 是日期或时间戳格式，则 TOCHAR 将 tochar-expression 解释为 \$HOROLOGY 日期整数。因此 2010-03-23 12:15:23（注意时间值中的错误连字符）被解释为 \$HOROLOGY 日期 2010（1846-07-03 12:00:00 AM）。
 - 如果 tochar 表达式日期或时间不是有效的日期或时间值，会发出 SQLCODE -400 错误。这可能发生在不存在的日期（例如 2 月 30 日）或 1840 年 12 月 31 日之前的日期。
 - 如果指定的格式包含无效的日期、时间或时间戳代码元素（例如 YYYYYY、MIN、HH48），则 TOCHAR 返回无效代码元素的格式代码文字；它返回有效代码元素的日期、时间或时间戳转换值（如果有）。
 - 如果 TOCHAR 无法识别任何格式代码元素（例如，格式是空字符串）或数字格式的位数少于 tochar 表达式值，则 TOCHAR 返回井号 (#) 字符。（当 tochar-expression 以至少两个整数数字开头时为真；否则 TOCHAR 返回 NULL。）
 - 如果省略格式，TOCHAR 将 tochar-expression 的数字部分作为规范数字返回，遇到非数字字符时截断。如果 tochar-expression 为非数字，则 TOCHAR 返回 0。如果 tochar-expression 为 null，则 TOCHAR 返回 null。

TOCHAR 和 TODATE

TOCHAR 将日期整数转换为格式化的日期字符串，或将时间整数转换为格式化的时间字符串。如果错误地为 TOCHAR 提供了格式化的日期或时间字符串，它会返回错误的数据。

TODATE 将格式化的日期字符串转换为相应的日期整数。如果错误地为 TODATE 提供了日期整数，它会返回未修改的整数。

以下示例显示了 TODATE 和 TOCHAR 的这些正确和错误用法。

以下嵌入式 SQL 示例使用 TODATE 执行日期转换。TODATE 接受一个日期字符串并返回相应的日期整数 (59832)。\$ZDATE 函数用于将此日期整数显示为格式化日期 02/22/2018。在这个例子中，TODATE 也被错误地提供了一个日期整数；它只是返回这个整数。

```

/// d ##class(PHA.TEST.SQLFunction).ToChar()
ClassMethod ToChar()
{
    &sql(
        SELECT
            TO_DATE('2018-02-22','YYYY-MM-DD'), /* correct */
            TO_DATE(64701,'YYYY-MM-DD')          /* ERROR! */
        INTO
            :a,:b
    )
    if SQLCODE '= 0 {
        w !,"Error code ",SQLCODE
    } else {
        w !,a
        w !,$ZDATE(a)
        w !,b
    }
}

```

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

```
64701
02/22/2018
64701
```

以下嵌入式 SQL 示例显示了使用 TOCHAR 的日期转换。如预期的那样，第一个 TOCHAR 将日期整数转换为相应的格式化日期字符串。但是，第二个 TOCHAR 给出了意想不到的结果。由于 TOCHAR 需要数字输入，因此它将输入中的日期分隔符视为减号并执行减法。因此，它格式化与日期整数 1970 (2004 减 10 减 24) 相对应的日期：1846-5-24。显然，这不是程序员的本意。

```
/// d ##class(PHA.TEST.SQLFunction).ToChar1()
ClassMethod ToChar1()
{
    &sql(
        SELECT
            TO_CHAR(59832,'YYYY-MM-DD'),      /* correct */
            TO_CHAR(2004-10-24,'YYYY-MM-DD') /* ERROR! */
        INTO
            :a,:b
    )
    if SQLCODE '= 0 {
        w !,"Error code ",SQLCODE
    } else {
        w !,a
        w !,b
    }
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).ToChar1()
```

```
2004-10-24
1846-05-24
```

相关 SQL 函数

- TOCHAR 将日期整数、时间戳或数字转换为字符串。
- TODATE 对日期执行反向操作；它将格式化的日期字符串转换为日期整数。
- TOTIMESTAMP 对时间戳执行反向操作；它将格式化的日期和时间字符串转换为标准时间戳。
- TONUMBER 对数字执行相反的操作；它将数字字符串转换为数字。
- CAST 和 CONVERT 执行 DATE、TIMESTAMP 和 NUMBER 数据类型转换。

日期到字符串的转换

\$HOROLOGY 格式是用于表示日期和时间的 SQL 逻辑格式。它是一个包含两个逗号分隔整数的字符串：第一个是自 1840 年 12 月 31 日以来的天数；第二个是自当天午夜以来的秒数。

可以使用 TOCHAR 将 \$HOROLOGY 日期整数或两个逗号分隔整数的 \$HOROLOGY 字符串转换为格式化的日期字符串或格式化的日期和时间字符串。tochar-expression 的值必须是有效的 \$HOROLOGY 值。

下表列出了此版本 TOCHAR 的有效日期格式代码。

Format	Code Meaning
D	D 星期几 (1-7)。默认情况下，1 是星期日 (一周的第一天)

Format	Code Meaning
DD), 但此指定是可配置的; 请参阅 DAYOFWEEK 函数。
DY	两位数的月份日期 (01-31)。
	当天的缩写名称, 由当前语言环境的 WeekdayAbbr 属性指定。默认值为: Sunday Monday Tuesday Wednesday Thursday Friday Saturday
DAY	日期名称, 由当前语言环境中的 WeekdayName 属性指定。默认值为: Sunday Monday Tuesday Wednesday Thursday Friday Saturday
MM	两位数的月份编号 (01-12; 01 = JAN)。
MON	月份的缩写名称, 由当前语言环境中的 MonthAbbr 属性指定。默认值为: Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec. Not case-sensitive.
MONTH	月份的全名, 由当前语言环境中的 MonthName 属性指定。默认值为: January February March April May June July August September October November December. Not case-sensitive.
YYYY	四位数年份。
YYY	年份的最后 3 位数字。
YY	年份的最后 2 位数字。
Y	年份的最后 1 位数字。
RRRR	四位数年份。
RR	L 年份的最后 2 位数字。
DDD	一年中的某一天 (见下文)。
J	儒略日期 (自公元前 4712 年 1 月 1 日 (BCE) 以来的天数)。

日期格式元素之间需要分隔符, 但以下格式字符串除外: YYYYMMDD、DDMMYYYY 和 YYYYMM。最后一个返回年份和月份值并忽略月份中的日期。

请注意, 格式代码定义中提到的语言环境指的是 ObjectScript \$ZDATE 和 \$ZDATEH 文档中描述的相同语言环境。

日期转换示例

以下是 TOCHAR 与 \$HOROLOG 日期整数或完整的 \$HOROLOG 字符串值的所有有效用法, 以返回格式化的日期字符串或日期和时间字符串:

```
SELECT TO_CHAR(64701, 'YYYY-MM-DD') AS DateFD,
       TO_CHAR(64701, 'YYYY-MM-DD HH24:MI:SS') AS DateFDT,
       TO_CHAR('64701,50278', 'YYYY-MM-DD') AS DateTimeFD,
       TO_CHAR('64701,50278', 'YYYY-MM-DD HH24:MI:SS') AS DateTimeFDT
```

在以下示例中, 每个 TOCHAR 接受一个日期整数并返回一个根据格式字符串参数格式化的日期字符串:

```
SELECT TO_CHAR(64701, 'MM/DD/YYYY'),           /* returns 02/22/2018 */
       TO_CHAR(64701, 'DAY MONTH DD, YYYY') /* returns Thursday February 22, 2018 */
```

以下示例采用日期整数并返回格式化的日期字符串。不是格式字符的字符作为文字传递给输出字符串:

```
SELECT TO_CHAR(64701, 'The date MM/DD/YYYY should be noted')
```

```
The date 02/22/2018 should be noted
```

一年中的一天

可以使用 DDD 将日期表达式转换为一年中的某天（自 1 月 1 日以来经过的天数）和年份。格式字符串 DDD,YYYY 必须与 \$HOROLOGY 格式的日期表达式配对。（\$HOROLOGY 时间值，如果指定，将被忽略。）DDD 和 YYYY（或 YY）格式元素可以按任何顺序指定；它们之间的分隔符是强制性的，并作为文字返回。以下示例显示了一年中的这一天的用法：

```
SELECT TO_CHAR('64701','YYYY:DDD')
```

```
2018:053
```

```
SELECT TO_CHAR('64701,12345','DDD YY')
```

```
053 18
```

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

儒略日期转换

儒略日（Julian day，JD）是指由公元前4713年1月1日，协调世界时中午12时开始所经过的天数，多为天文学家采用，用以作为天文学的单一历法，把不同历法的年表统一起来。如果计算相隔若干年的两个日期之间间隔的天数，利用儒略日就比较方便。

提供“儒略”日期格式以允许 1841 年之前的日期。TOCHAR 将数据类型 %Date 或 %TimeStamp 的日期值转换为七位儒略日期整数。

胡子怡：

默认情况下，%Date 数据类型不表示 1840 年 12 月 31 日之前的日期。但是，可以重新定义此数据类型的 MINVAL 参数以允许将较早的日期表示为负整数，限制为第 1 年 1 月 1 日。

如果指定的格式由包含字母“J”的字符串组成，则返回的日期值将是“儒略”日期，即从公元前 4712 年 1 月 1 日开始的天数。格式字符串中只能指定字母“J”；包含任何其他字符会导致将“J”视为文字，并将日期翻译为标准日期。

儒略日期的最大 tochar 表达式值为“9999-12-31”，对应儒略日计数 5373484。最小值为“-4712-01-01”，对应儒略日计数 0000001。儒略日计数为始终表示为七位整数，必要时带有前导零。

以下示例返回 2369916（签署美国独立宣言）和 1709980（亚克兴战役标志着奥古斯都凯撒统治下罗马帝国的开始）：

```
SELECT TO_CHAR('1776-07-04','J') AS UnitedStatesStart,  
       TO_CHAR('-0031-09-02','J') AS RomanEmpireStart
```

```
2369916 1709980
```

注意：以下注意事项不应影响使用 TOCHAR 和 TODATE

的日期和儒略日计数的相互转换。它可能会影响使用儒略日计数进行的某些计算。

1721424 (1/1/1) 之前的儒略日计数与其他软件实现兼容，例如 Oracle。它们与通常使用的 BCE 日期不同。正常使用中，没有 Year 0；日期从 12/31/-1 到 1/1/1。在 Oracle 使用中，儒略日期 1721058 到 1721423 完全无效，并返回错误。这些儒略日期返回不存在的第 0 年作为占位符。因此，涉及 BCE 日期的计算必须调整一年以对应于常见用法。

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

#SQL #Caché

源

<https://cn.community.intersystems.com/post/%E7%AC%A0%E7%B8%A0-sql%E5%87%BD%E6%95%B0-toch-ar%EF%BC%88%E4%B8%80%E5%B0%8A>