文章
姚 鑫 · 五月 13, 2022 阅读大约需 5 分钟

第141章 SQL函数 TOCHAR（二）

第141章 SQL函数 TOCHAR（二）

可以使用 TOCHAR 将以下 tochar 表达式时间值转换为格式化的时间字符串：

- $HOROLOG 时间整数（$HOROLOG 的时间部分）。 tochar-expression 的值必须是有效的逻辑时间（0 到 86399 范围内的整数）。不要提供包含日期和时间组件的完整 $HOROLOG 值（例如 64701,42152）；TOCHAR 时间转换会错误地将 $HOROLOG 的第一个（日期）组件转换为格式化的时间字符串，并忽略第二个（时间）组件。
- 逻辑时间戳值。 tochar-expression 的值必须是 %TimeStamp 数据类型（不是字符串数据类型），格式为 YYYY-MM-DD hh:mm:ss。时间戳的日期部分被忽略，时间部分被转换。例如，SYSDATE 是一个逻辑时间戳。
- 标准 ODBC 时间格式的时间值。 tochar-expression 的值必须采用 hh:mm:ss 格式，并且可以是字符串。
- 本地时间格式的时间值（使用当前 NLS 区域设置）。例如，如果 NLS TimeSeparator 设置为 “ ^ ”，则 tochar-expression 的值可以采用 hh^mm^ss 格式，并且可以是字符串。

在所有这些情况下，格式的值必须是一个只包含时间格式代码的字符串：

Format	Code Meaning
HH	一天中的小时（1 到 12）
HH12	一天中的小时（1 到 12）
HH24	小时（0 到 23）
MI	分钟（0 到 59）
SS	分钟（0 到 59）
SSSSS	午夜后的秒数（0 到 86388）
AM / PM	经络指标（AM = 中午之前，PM = 中午之后）。使用适当的 AM 或 PM 后缀将时间值转换为 12 小时格式。返回的 AM 或 PM 后缀源自时间值，而不是指定的格式代码。在格式中，可以使用 AM 或 PM；它们在功能上是相同的。

包含任何其他格式代码值会导致 tochar-expression 整数被解释为日期。

以下示例导致 “ 64701 ” 被解释为时间值 05:58:21 PM：

```
SELECT TO_CHAR('64701','HH12:MI:SS PM')
```

```
05:58:21 PM
```

以下示例将两个逻辑时间戳的时间部分转换为格式化的时间字符串。请注意，格式不支持小数秒；tochar 表达式中的小数秒被截断。

```
SELECT TO_CHAR(SYSDATE,'HH12:MI:SS PM'),
       TO_CHAR(CURRENT_TIMESTAMP(6),'HH12:MI:SS PM')
```

```
11:18:03 AM 11:18:03 AM
```

以下嵌入式 SQL 示例转换以 ODBC 标准格式和当前 NLS 语言环境格式指定的时间值：

```
/// d ##class(PHA.TEST.SQLFunction).ToChar2()
ClassMethod ToChar2()
{
    s restore = ##class(%SYS.NLS.Format).GetFormatItem("TimeSeparator")
    w "Time Separator is = ", restore,!
    d ##class(%SYS.NLS.Format).SetFormatItem("TimeSeparator", "^")
    w "Time Separator is now = ",##class(%SYS.NLS.Format).GetFormatItem("TimeSeparato
r"),!
    &sql(
        SELECT
            TO_CHAR('15:35:43.99', 'HH12:MI:SS PM'),
            TO_CHAR('15^35^43.99','HH12:MI:SS PM')
        INTO
            :standard,:local
    )
    w "Converted standard-format time: ",standard,!
    w "Converted locale-format time: ",local,!
    d ##class(%SYS.NLS.Format).SetFormatItem("TimeSeparator",restore)
    w "Time Separator is = ", ##class(%SYS.NLS.Format).GetFormatItem("TimeSeparator")
}
```

时间戳到格式化日期时间字符串的转换

可以使用 TOCHAR 将时间戳转换为格式化的日期时间字符串。 tochar-expression
的值必须是有效的逻辑时间戳值。

时间戳的日期部分使用日期到字符串的转换格式代码进行格式化。下表列出了时间戳的时间部分的其他格式代码。

Format	Code Meaning
HH	一天中的小时（1 到 12）
HH12	一天中的小时（1 到 12）
HH24	小时（0 到 23）
MI	Minute (0 through 59)
SS	Second (0 through 59)
SSSSS	午夜后的秒数（0 到 86388）
AM	Meridian Indicator (before noon)
PM Meridian Indicator (after noon)	

以下示例返回当前系统日期（时间戳），以及转换为以两种不同格式显示的当前系统日期：

```
SELECT SYSDATE,
       TO_CHAR(SYSDATE,'MM/DD/YYYY HH:MI:SS'),
       TO_CHAR(SYSDATE,'DD MONTH YYYY at SSSSS seconds')
```

请注意，格式字符串中使用的任何不是格式代码的字符都会在结果字符串中原位返回。

数字到字符串的转换

可以使用 TOCHAR 将数字转换为格式化的数字字符串。下表列出了使用 TOCHAR 的格式参数的有效格式代码。

如果省略格式参数，则输入数值被评估为整数：前导零和前导加号被删除，前导减号被保留，并且数值在第一个非数字字符处被截断，例如逗号或期间。没有提供前导空格或其他格式。

Format	Code	Example Description

9	9999	返回具有指定位数的值，如果为正则带前导空格，如果为负则带负号。前导零为空白，但零值除外，它为定点数的整数部分返回零。
0	09999 99990	返回前导零。返回尾随零。
$	$9999	返回带有前导美元符号的值。请注意，对于正数，美元符号前面有一个空格。
B	B9999	当整数部分为零时（无论格式参数中的 0 ’；是什么），为定点数的整数部分返回空格。
S	S9999 9999S	返回带有前导减号 “ - ” 的负值。返回带有前导加号 “ + ” 的正值。返回带有尾随减号 “ - ” 的负值。返回带有尾随加号 “ + ” 的正值。
D	99D99	返回指定位置的小数分隔符。使用的 DecimalSeparator 是为语言环境定义的。默认为句点 “ . ” 。格式参数中只允许有一个 “ D ” 。
G	9G999	返回指定位置的数字组分隔符。使用的 NumericGroupSeparator 是为区域设置定义的。默认为逗号 “ ， ” 。小数分隔符的右侧不得出现数字组分隔符。
FM	FM90.9	返回一个没有前导或尾随空格的值。
,	9,999	在指定位置返回一个逗号。小数点右侧不能出现逗号。格式参数不能以逗号开头。
.	99.99	返回指定位置的小数点（即句点 “ . ” ）。只有一个 ” . ” 在格式参数中是允许的。

格式可以将小数分隔符和数字组分隔符指定为文字字符，也可以指定为区域设置的 DecimalSeparator 和 NumericGroupSeparator 的当前值。可以按如下方式确定当前的语言环境值：

```
DHC-APP>  WRITE ##class(%SYS.NLS.Format).GetFormatItem("DecimalSeparator"),!
.

DHC-APP>  WRITE ##class(%SYS.NLS.Format).GetFormatItem("NumericGroupSeparator")
,
```

如果格式参数包含的整数位数少于输入数值表达式，则 TOCHAR
不返回数字；相反，它返回一个由两个或多个井号 (##) 组成的字符串。井号的数量表示当前格式参数的长度加一。

如果格式参数包含的小数位数少于输入数值表达式，则 TOCHAR
将数字四舍五入为指定的小数位数，如果未提供十进制格式，则四舍五入为整数。

如果 tochar-expression 为 null，则 TOCHAR 返回 null。

#SQL #Caché

源URL:
https://cn.community.intersystems.com/post/%E7%AC%AC141%E7%AB%A0-sql%E5%87%BD%E6%95%B0-tochar%EF%BC%88%E4%BA%8C%EF%BC%89