

文章

姚鑫 · 五月 16, 2022 阅读大约需 5 分钟

第144章 SQL函数 TODATE (二)

第144章 SQL函数 TODATE (二)

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。DDD 和 YYYY 格式元素可以按任意顺序指定；它们之间的分隔符是强制性的。以下示例显示了这一年中的一天的用法：

```

/// d ##class(PHA.TEST.SQLFunction).ToDate4()
ClassMethod ToDate4()
{
    NEW SQLCODE
    &sql(
        SELECT TO_DATE('2018:60','YYYY:DDD')
        INTO :a
    )
    if SQLCODE = 0 {
        w a," = ", $ZDATE(a,1,,4)
    } else {
        w "error:",SQLCODE
    }
}

```

```

DHC-APP>d ##class(PHA.TEST.SQLFunction).ToDate4()
64708 = 03/01/2018

```

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，该示例返回 2/29/2020（不是 12/31/2020）：

```

/// d ##class(PHA.TEST.SQLFunction).ToDate4()
ClassMethod ToDate5()
{
    n SQLCODE
    &sql(
        SELECT TO_DATE('2020-12-31-60','YYYY-MM-DD-DDD')
        INTO :a
    )
    if SQLCODE = 0 {
        w a," = ", $ZDATE(a,1,,4)
    } else {
        w "error:",SQLCODE
    }
}

```

```
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).ToDate5()
65438 = 02/29/2020
```

TODATE 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。

儒略历日期(J格式)

SQL中，儒略日可用于1840年12月31日之前的任何日期。因为在内部将这个日期表示为0，所以需要特殊的语法来表示更早的日期。为此，TODATE提供了“J”(或“J”)格式。儒略日转换将七位数的内部数值(儒略日计数)转换为显示格式或ODBC格式的日期。例如：

```
/// d ##class(PHA.TEST.SQLFunction).ToDate6()
ClassMethod ToDate6()
{
    n SQLCODE
    &sql(
        SELECT TO_DATE(2300000, 'J')
        INTO :a
    )
    if SQLCODE = 0 {
        w a
    } else {
        w "error:", SQLCODE
    }
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).ToDate6()
1585-01-31
```

返回以下日期:1585-01-31(ODBC格式)或01/31/1585(显示格式)。儒略日计数1721424返回公元1年的1月1日(1-01-01)。儒略日计数，如1709980(亚克兴战役标志着罗马帝国在奥古斯都·凯撒统治下的开始)返回公元前(BC)日期，显示的年份前面有一个负号。

儒略日计数在内部总是表示为一个七位数，必要时带有前导零。TODATE允许输入不带前导零的儒略日计数。最大允许儒略日是5373484，它返回12/31/9999。允许的最小儒略日是0000001，它返回01/01/-4712(即BCE日期01/01/-4713)。任何超出此范围的值都会生成SQLCODE -400错误，其%msg值为“无效的儒略日值”。儒略日必须介于1和5373484之间。

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。这可能会影响使用儒略日计数进行的一些计算。

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。

另请注意，这些日期计数并未考虑公历改革（1582 年 10 月 15 日颁布，但直到 1752 年才在英国及其殖民地采用）引起的日期变化。

TODATE 允许返回对应于儒略日计数的日期表达式。 TOCHAR

允许返回对应于日期表达式的儒略日计数，如以下示例所示：

```
SELECT
    TO_CHAR('1776-07-04','J') AS JulianCount,
    TO_DATE(2369916,'J') AS JulianDate
```

```
2369916 1776/7/4 0:00:00
```

示例

默认日期格式示例

以下嵌入式 SQL 示例指定使用默认日期格式解析的日期字符串。这两个都转换为 DATE 数据类型内部值 60537：

```
/// d ##class(PHA.TEST.SQLFunction).ToDate7()
ClassMethod ToDate7()
{
    NEW SQLCODE
    &sql(
        SELECT
            TO_DATE('29 September 2018'),
            TO_DATE('29 SEP 2018')
        INTO :a,:b
    )
    IF SQLCODE=0 {WRITE a,!,b }
    ELSE { WRITE "error:",SQLCODE }
}
```

以下嵌入式 SQL 示例指定具有两位数年份的日期字符串，格式为默认值。请注意，两位数年份默认为 1900 到 1999。因此，内部 DATE 值为 24012：

```
/// d ##class(PHA.TEST.SQLFunction).ToDate8()
ClassMethod ToDate8()
{
    NEW SQLCODE
    &sql(
        SELECT
            TO_DATE('29 September 06'),
            TO_DATE('29 SEP 06')
        INTO :a,:b
    )
    IF SQLCODE=0 {WRITE a,!,b }
    ELSE { WRITE "error:",SQLCODE }
}
```

指定日期格式示例

以下嵌入式 SQL 示例以各种格式指定日期字符串。所有这些都转换为 DATE 数据类型内部值 64701。

```
/// d ##class(PHA.TEST.SQLFunction).ToDate9()
```

```

ClassMethod ToDate9()
{
    NEW SQLCODE
    &sql(
        SELECT
            TO_DATE('2018 Feb 22','YYYY MON DD'),
            TO_DATE('FEBRUARY 22, 2018','month dd, YYYY'),
            TO_DATE('2018***02***22','YYYY***MM***DD'),
            TO_DATE('02/22/2018','MM/DD/YYYY')
        INTO :a,:b,:c,:d
    )
    if SQLCODE = 0 {
        w !,a,!,b,!,c,!,d
    } else {
        w "error:",SQLCODE
    }
}

```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).ToDate9()
```

```

64701
64701
64701
64701

```

以下嵌入式 SQL 示例指定不需要元素分隔符的日期格式。它们返回 64701 的日期内部值：

```

/// d ##class(PHA.TEST.SQLFunction).ToDate10()
ClassMethod ToDate10()
{
    NEW SQLCODE
    &sql(
        SELECT
            TO_DATE('02222018','MMDDYYYY'),
            TO_DATE('22022018','DDMMYYYY'),
            TO_DATE('20182202','YYYYDDMM'),
            TO_DATE('20180222','YYYYMMDD')
        INTO :a,:b,:c,:d
    )
    if SQLCODE = 0 {
        w !,a,!,b,!,c,!,d
    } else {
        w "error:",SQLCODE
    }
}

```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).ToDate10()
```

```

64701
64701
64701
64701

```

```

/// d ##class(Pha.Test.SQLFunction).ToDate11()
ClassMethod ToDate11()
{
    NEW SQLCODE
    &sql(
        SELECT TO_DATE('201806','YYYYMM')
        INTO :a
    )
    if SQLCODE = 0 {
        w a," = ",$ZDATE(a,1,,4)
    } else {
        w "error:",SQLCODE
    }
}
}

```

#SQL #Caché

<https://cn.community.intersystems.com/post/%E7%AC%AC144%E7%AB%A0-sql%E5%87%BD%E6%95%B0-toda-te%E5%BC%88%E4%BA%8C%E5%BC%89>