

文章

[姚鑫](#) · 六月 12, 2022 阅读大约需 5 分钟

第七章 日期和时间构造

第七章 日期和时间构造

验证和转换 ODBC 日期、时间或时间戳。

大纲

```
{d 'yyyy-mm-dd'}
{d nnnnnn}

{t 'hh:mm:ss[.fff]'}
{t nnnnnn.nnn}

{ts 'yyyy-mm-dd [hh:mm:ss.fff]'}
{ts 'mm/dd/yyyy [hh:mm:ss.fff]'}
{ts nnnnnn}
```

描述

这些构造采用 ODBC 日期、时间或时间戳格式的整数或字符串，并将其转换为相应的 IRIS 日期、时间或时间戳格式。他们执行数据输入以及值和范围检查。

`{d 'string'}`

`{d 'string'}` 日期构造验证 ODBC 格式的日期。如果日期有效，它将以 IRIS \$HOROLOG 日期格式存储（逻辑模式）作为 1840-12-31 的整数计数值。IRIS 不附加默认时间值。要支持早于 1840-12-31 的日期，必须在表中定义数据类型为 %Library.Date(MINVAL=-nnn) 的日期字段，其中 MINVAL 是从 1840-12-31 倒数的负天数（第 0 天）最大为 -672045 (0001-01-01)。

- 小于 -672045 (0001-01-01) 或大于 2980013 (9999-12-31) 的整数会生成 SQLCODE -400 <VALUE OUT OF RANGE> 错误。
- 无效日期（例如非 ODBC 格式的日期或非闰年的日期 02-29）：IRIS 生成 SQLCODE -146 错误：“yyyy-mm-dd”是无效的 ODBC/JDBC 日期值”。
- ODBC 时间戳值：IRIS 验证时间戳的日期和时间部分。如果两者都有效，则仅转换日期部分。如果日期或时间无效，系统将生成 SQLCODE -146 错误。

`{t 'string'}`

`{t 'string'}` 时间构造验证 ODBC 格式的时间。如果时间有效，它以 IRIS \$HOROLOG 时间格式将其存储（逻辑模式），作为从午夜开始的整数秒计数，并带有指定的小数秒。IRIS 显示模式和 ODBC 模式不显示小数秒；从这些显示格式中截断小数秒。

- 小于0 (00:00:00) 或大于 86399.99 (23:59:59.99) 的整数会生成 SQLCODE -400 <ILLEGAL VALUE> 错误。
- 无效时间（例如不是 ODBC 格式的时间或小时数>23 的时间）：IRIS 生成 SQLCODE -147 错误：“ hh:mi:ss.fff is an invalid ODBC/JDBC Time value ”。
- ODBC 时间戳值：IRIS 生成 SQLCODE -147 错误。

{ts 'string'}

{ts 'string'} 时间戳构造验证日期/时间并以 ODBC 时间戳格式返回；始终保留并显示指定的小数秒。{ts 'string'} 时间戳构造还验证日期并以 ODBC 时间戳格式返回它，并提供 00:00:00 的时间值。

- 正整数或负整数日期（-672045 到 2980013）：IRIS 附加时间值 00:00:00，然后以 ODBC 格式存储生成的时间戳。例如，64701 返回 2018-02-22 00:00:00。这是一个有效的 \$HOROLOG 日期整数。\$HOROLOG 0 是 1840-12-31。
- ODBC 格式的有效时间戳：IRIS 将提供的值保持不变 这是因为 IRIS 时间戳格式与 ODBC 时间戳格式相同。
- 使用区域设置默认日期和时间格式的有效时间戳（例如，2/29/2016 12:23:46.77）：IRIS 以 ODBC 格式存储和显示提供的值。
- 无效的时间戳（例如在非闰年中日期部分指定为 02-29 或时间部分指定小时>23 的时间戳）：IRIS 返回字符串“error”作为值。
- 没有时间值的有效日期（以 ODBC 或区域设置格式）：IRIS 附加时间值 00:00:00，然后以 ODBC 格式存储生成的时间戳。它在必要时提供前导零。例如，2/29/2016 返回 2016-02-29 00:00:00。
- 没有时间值的格式正确但无效的日期（以 ODBC 或区域设置格式）：IRIS 附加时间值 00:00:00。然后它存储提供的日期部分。例如，02/29/2019 返回 02/29/2019 00:00:00。
- 格式不正确且无效的日期（以 ODBC、语言环境或 \$HOROLOG 格式）且没有时间值：IRIS 返回字符串“error”。例如，2/29/2019（没有前导零和无效的日期值）返回“错误”。00234（前导零的 \$HOROLOG）返回“错误”

示例

以下动态 SQL 示例验证以 ODBC 格式（带或不带前导零）提供的日期并将它们存储为等效的 \$HOROLOG 值 64701。此示例显示 %SelectMode 0（逻辑）值：

```
/// d ##class(PHA.TEST.SQLFunction).DateTime()  
ClassMethod DateTime()  
{  
    s myquery = 2  
    s myquery(1) = "SELECT {d '2018-02-22'} AS date1,"  
    s myquery(2) = "{d '2018-2-22'} AS date2"  
    s tStatement = ##class(%SQL.Statement).%New()  
    s tStatement.%SelectMode=0  
    s tStatus = tStatement.%Prepare(.myquery)  
    s rset = tStatement.%Execute()  
    d rset.%Display()  
}
```

```
DHC-APP> d ##class(PHA.TEST.SQLFunction).DateTime()  
date1    date2  
64701    64701
```

1 Rows(s) Affected

以下动态 SQL 示例验证以 ODBC 格式（带或不带前导零）提供的时间，并将它们存储为等效的 \$HOROLOG 值 43469。此示例显示 %SelectMode 0（逻辑）值：

```
/// d ##class(PHA.TEST.SQLFunction).DateTime1()
ClassMethod DateTime1()
{
    s myquery = 3
    s myquery(1) = "SELECT {t '12:04:29'} AS time1,"
    s myquery(2) = "{t '12:4:29'} AS time2,"
    s myquery(3) = "{t '12:04:29.00000'} AS time3"
    s tStatement = ##class(%SQL.Statement).%New()
    s tStatement.%SelectMode=0
    s tStatus = tStatement.%Prepare(.myquery)
    s rset = tStatement.%Execute()
    d rset.%Display()
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).DateTime1()
time1    time2    time3
43469    43469    43469
```

1 Rows(s) Affected

以下动态 SQL 示例使用小数秒验证以 ODBC 格式提供的时间，并将它们存储为等效的 \$HOROLOG 值 43469 并附加小数秒。尾随零被截断。此示例显示 %SelectMode 0 (逻辑) 值：

```
/// d ##class(PHA.TEST.SQLFunction).DateTime2()
ClassMethod DateTime2()
{
    s myquery = 3
    s myquery(1) = "SELECT {t '12:04:29.987'} AS time1,"
    s myquery(2) = "{t '12:4:29.987'} AS time2,"
    s myquery(3) = "{t '12:04:29.987000'} AS time3"
    s tStatement = ##class(%SQL.Statement).%New()
    s tStatement.%SelectMode=0
    s tStatus = tStatement.%Prepare(.myquery)
    s rset = tStatement.%Execute()
    d rset.%Display()
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).DateTime2()
time1    time2    time3
43469.987    43469.987    43469.987
```

1 Rows(s) Affected

以下动态 SQL 示例以多种格式验证时间和日期值，并将它们存储为等效的 ODBC 时间戳。必要时提供时间值 00:00:00。此示例显示 ac%SelectMode 0 (逻辑) 值：

```
/// d ##class(PHA.TEST.SQLFunction).DateTime3()
ClassMethod DateTime3()
{
    s myquery = 6
    s myquery(1) = "SELECT {ts '2018-02-22 01:43:38'} AS ts1,"
    s myquery(2) = "{ts '2018-02-22'} AS ts2,"
    s myquery(3) = "{ts '02/22/2018 01:43:38.999'} AS ts3,"
```

```
s myquery(4) = "{ts '2/22/2018 01:43:38'} AS ts4,"
s myquery(5) = "{ts '02/22/2018'} AS ts5,"
s myquery(6) = "{ts '64701'} AS ts6"
s tStatement = ##class(%SQL.Statement).%New()
s tStatement.%SelectMode=0
s tStatus = tStatement.%Prepare(.myquery)
s rset = tStatement.%Execute()
if rset.%Next() {
    w rset.ts1,!
    w rset.ts2,!
    w rset.ts3,!
    w rset.ts4,!
    w rset.ts5,!
    w rset.ts6
}
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).DateTime3()
2018-02-22 01:43:38
2018-02-22 00:00:00
2018-02-22 01:43:38.999
2018-02-22 01:43:38
2018-02-22 00:00:00
2018-02-22 00:00:00
```

[#SQL #Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E4%B8%83%E7%AB%A0-%E6%97%A5%E6%9C%9F%E5%92%8C%E6%97%B6%E9%97%B4%E6%9E%84%E9%80%A0>