

文章

[姚鑫](#) · 一月 12, 2022 阅读大约需 9 分钟

第二十二章 SQL函数 CAST（一）

第二十二章 SQL函数 CAST（一）

将给定表达式转换为指定数据类型的函数。

大纲

```
CAST(expr AS CHAR | CHARACTER | VARCHAR | NCHAR | NVARCHAR)
CAST(expr AS CHAR(n) | CHARACTER(n) | VARCHAR(n) )
CAST(expr AS CHAR VARYING | CHARACTER VARYING)
CAST(expr AS INT | INTEGER | BIGINT | SMALLINT | TINYINT)
CAST(expr AS DEC | DECIMAL | NUMERIC)
CAST(expr AS DEC(p[,s]) | DECIMAL(p[,s]) | NUMERIC(p[,s]) )
CAST(expr AS DOUBLE)
CAST(expr AS MONEY | SMALLMONEY)
CAST(expr AS DATE)
CAST(expr AS TIME)
CAST(expr AS POSIXTIME)
CAST(expr AS TIMESTAMP | DATETIME | SMALLDATETIME)
CAST(expr AS BIT)
CAST(expr AS BINARY | BINARY VARYING | VARBINARY)
CAST(expr AS BINARY(n) | BINARY VARYING(n) | VARBINARY(n) )
CAST(expr AS GUID)
```

参数

- expr - SQL表达式，通常是表的文字或数据字段。
- n - 一个整数，指示要返回的最大字符数。
如果n小于expr数据，返回的数据将被截断为n个字符。
如果n大于expr数据，则不执行填充。
- p,s - 可选 - p=精度(最大总位数)，整数形式。
s=刻度(十进制数字的最大值)，用整数表示。
如果未指定比例，则默认为15。

描述

SQL CAST函数将表达式的数据类型转换为指定的数据类型。当Expr的数据类型是标准数据类型或标准数据类型(如 %Library.String、%Library.Time、%Library.Date或%Library.TimeStamp)的子类时，CAST可以转换该数据类型。

可以将expr转换为以下任何数据类型

- CHAR或CHARACTER:用其初始字符表示数字或字符串。
当指定为CAST或CONVERT时，不带n的VARCHAR默认长度为30个字符。
否则，VARCHAR数据类型(没有指定大小)将被映射到一个1个字符的MAXLEN，如data Types表所示。
NCHAR等价于CHAR;

NVARCHAR相当于VARCHAR。

- CHAR(n)、CHARACTER(n)或VARCHAR(n):用n指定的字符数表示数字或字符串。
- CHAR VARYING或CHARACTER VARYING:用原值中的字符数表示数字或字符串。
- INT、INTEGER、BIGINT、SMALLINT和TINYINT:用整数部分表示数字。
十进制数字被截断。
- “DEC”、“DECIMAL”和“NUMERIC”:用原值中的位数来表示数字。
使用\$DECIMAL函数进行转换,该函数将\$DOUBLE值转换为\$DECIMAL值。
如果指定了p(精度),则保留为定义的数据类型的一部分,但不会影响CAST返回的值。
如果指定正整数的s(scale)值,则十进制值四舍五入为指定的位数。
(在Display模式中包含适当数量的尾随零,但在Logical模式和ODBC模式中被截断。)
如果指定s=0,数值将四舍五入为整数。
如果指定s=-1,数值将被截断为整数。
- DOUBLE表示IEEE浮点标准。
- MONEY和SMALLMONEY是货币数字数据类型。
货币数据类型的规模总是4。
- DATE:日期。
日期可以用以下任何一种格式表示,这取决于上下文:所在地区的显示日期格式(例如,MM/DD/YYYY);
ODBC日期格式(YYYY-MM-DD);
或\$HOROLOGY整数日期存储格式(nnnnnn)。
必须将\$HOROLOGY日期部分值指定为整数,而不是数字字符串。
- TIME:时间。
时间可以用以下任何一种格式表示,这取决于上下文:本地地区的显示时间格式(例如, hh:mm:ss);
ODBC日期格式(hh:mm:ss);
或\$HOROLOGY整数时间存储格式(nnnnnn)。
必须将\$HOROLOGY时间部分值指定为整数,而不是数字字符串。
- POSIXTIME:以64位有符号整数的形式表示日期和时间戳。
- TIMESTAMP、DATETIME和SMALLDATETIME:日期和时间戳,格式为YYYY-MM-DD hh:mm:ss.nnn。
这对应于ObjectScript \$ZTIMESTAMP特殊变量。
- BIT表示单个二进制值。
- BINARY、BINARY VARYING和VARBINARY表示数据类型%Library.Binary(xDBC数据类型BINARY)的值。对于BINARY,可选的n长度缺省为1,对于BINARY VARYING和VARBINARY,缺省为30。转换为二进制值时,实际上不会执行数据转换。不会截断指定长度为n的值的长度。
- GUID表示数据类型%Library.UniqueIdentifier的36个字符值。如果提供的expr超过36个字符,CAST将返回expr的前36个字符。要生成GUID值,请使用%SYSTEM.Util.CreateGUID()方法。

转换数字

数值可以转换为数字数据类型或字符数据类型。

当将数值结果转换为缩略值时,数值将被截断,而不是四舍五入。例如,将98.765转换为INT返回98,转换为CHAR返回9,转换为CHAR(4)返回98.7。请注意,将负数转换为CHAR仅返回负号,将小数转换为CHAR仅返回小数点。

数字可以由数字0到9、小数点、一个或多个前导符号(+或-)、指数符号(E或E)后面最多跟一个+或-符号组成。
数字不能包含组分隔符(逗号)。

在执行强制转换之前 SQL将数字解析为其规范形式:执行指数运算。

IRIS带前导和后导零、前导加号和后导小数点。

在转换数字之前解析多个符号。

然而,SQL将双负号作为注释指示符;

遇到数字中的双负号时,会将该行代码的其余部分作为注释处理。

浮点数可以采用DEC、DECIMAL或NUMERIC数据类型。

DOUBLE数据类型根据IEEE浮点标准表示浮点数。

浮点数据类型比DOUBLE数据类型具有更高的精度,适合于大多数应用程序。

不能使用CAST将浮点数转换为DOUBLE数据类型;

相反,使用ObjectScript \$DOUBLE函数。

当数字值被转换为日期或时间数据类型时,它在SQL中显示为0 (0);

但是，当将数字转换为日期或时间时，从嵌入式SQL传递到ObjectScript，它显示为相应的\$HOROLOGY值。

字符串类型转换

可以将字符串强制转换为另一种字符数据类型，返回单个字符、前n个字符或整个字符串。

在执行强制转换之前，SQL会解析内嵌的引号字符('can' t='can't)和字符串连接('can' || 'not'='cannot)。保留前导和尾随空格。

当字符串被转换为数字类型时，它总是返回一个数字零(0)。

转换为DATE、TIME和TIMESTAMP

可以将字符串强制转换为DATE、TIME或TIMESTAMP数据类型。
通过以下操作，可以得到一个有效的值：

- DATE:格式为“ yyyy-mm-dd ”的字符串可以转换为DATE。
此字符串格式对应于ODBC日期格式。
执行值和范围检查。
年份必须在0001到9999之间(包括)，从01月到12月，以及该月的适当日期(例如，02/29只在闰年有效)。
无效日期，如2013-02-29返回1840-12-31(逻辑日期0)。

将添加月和日字段中缺少的前导零。此转换的显示方式取决于显示模式和区域设置的日期显示格式。例如，‘ 2004-11-23 ’可能显示为‘ 11/23/2004 ’。在嵌入式SQL中，此强制转换作为相应的\$HOROLOGY日期整数返回。无效的ODBC日期或非数字字符串在转换为日期时在逻辑模式下表示为0；日期0显示为1840-12-31。

- TIME:格式为'hh:mm'、'hh:mm:ss'或'hh:mm:ss'的字符串。
nn'(有任意n个小数秒位数)可以被转换为TIME。
此字符串格式对应于ODBC时间格式。
执行值和范围检查。
缺少前导零被添加。
在嵌入式SQL中，这种转换将作为相应的\$HOROLOGY时间整数返回。
当转换为time时，无效的ODBC时间或非数字字符串在逻辑模式中显示为0；
时间0显示为00:00:00。
- TIMESTAMP:由有效日期和时间、有效日期或有效时间组成的字符串可以转换为TIMESTAMP。
日期部分可以采用多种格式。
缺少的日期部分默认为1841-01-01。
缺少的时段默认为00:00:00。
缺少前导零将添加到月和日。
小数秒(如果指定了)可以在句点(.)或冒号(:)之前加上。
这些符号有不同的含义。
句号表示标准分数；
因此，12:00:00.4表示十分之四秒，而12:00:00.004表示千分之四秒。
冒号表示接下来的是千分之一秒；
因此12:00:00:4表示千分之四秒。
冒号后面的数字限制为3位。

转换NULL和空字符串

NULL可以转换为任何数据类型并返回NULL。

空字符串("")的类型转换如下:

- 所有字符数据类型返回NULL。
- 所有数字数据类型都返回0(零), 以及尾随分数零的适当数量。
DOUBLE数据类型返回零, 不带尾随分数零。
- DATE数据类型返回12/31/1840。
- TIME数据类型返回00:00:00。
- TIMESTAMP、DATETIME和SMALLDATETIME数据类型返回NULL。
- BIT数据类型返回0。
- 所有二进制数据类型都返回NULL。

转换日期

可以将日期转换为日期数据类型、数字数据类型或字符数据类型。

将日期转换为POSIXTIME数据类型会将时间戳转换为编码的64位带符号整数。

由于日期没有时间部分, 时间部分被提供给时间戳编码为00:00:00。

CAST执行日期验证;

如果expr值不是有效日期, 则会发出SQLCODE -400错误。

将日期转换为TIMESTAMP、DATETIME或SMALLDATETIME数据类型将返回一个格式为YYYY-MM-DD hh:mm:ss的时间戳。

由于日期没有时间部分, 因此生成的时间戳的时间部分总是00:00:00。

CAST执行日期验证;

如果expr值不是有效日期, 则会发出SQLCODE -400错误。

下面的动态SQL示例将DATE数据类型字段转换为TIMESTAMP和POSIXTIME:

```
ClassMethod Cast()  
{  
    s myquery = 2  
    s myquery(1) = "SELECT TOP 5 DOB,CAST(DOB AS TIMESTAMP) AS TStamp,"  
    s myquery(2) = "CAST(DOB AS POSIXTIME) AS Posix FROM Sample.Person"  
    s tStatement = ##class(%SQL.Statement).%New()  
    s qStatus = tStatement.%Prepare(.myquery)  
    if qStatus '= 1 {  
        w "%Prepare failed:"  
        d $System.Status.DisplayError(qStatus)  
        q  
    }  
    s rset = tStatement.%Execute()  
    d rset.%Display()  
    w !,"End of data"  
}
```

```
d ##class(PHA.Test).Cast()  
DOB TStamp Posix  
1976-03-01 1976-03-01 00:00:00 1976-03-01 00:00:00  
2008-07-25 2008-07-25 00:00:00 2008-07-25 00:00:00  
1976-03-30 1976-03-30 00:00:00 1976-03-30 00:00:00  
1961-07-29 1961-07-29 00:00:00 1961-07-29 00:00:00  
1924-03-10 1924-03-10 00:00:00 1924-03-10 00:00:00
```

5 Rows(s) Affected
End of data

注:IRIS以下不可用POSIXTIME。

下面的动态SQL示例将TIMESTAMP数据类型字段转换为DATE和POSIXTIME:

```
ClassMethod Cast1()
{
    s myquery = 2
    s myquery(1) = "SELECT TOP 5 EventDate,CAST(EventDate AS DATE) AS Horolog,"
    s myquery(2) = "CAST(EventDate AS POSIXTIME) AS Posix FROM Aviation.Event"
    s tStatement = ##class(%SQL.Statement).%New()
    s qStatus = tStatement.%Prepare(.myquery)
    if qStatus '= 1 {
        w "%Prepare failed:"
        d $System.Status.DisplayError(qStatus)
        q
    }
    s rset = tStatement.%Execute()
    d rset.%Display()
    w !,"End of data"
}
```

将日期转换为数字数据类型将返回该日期的\$HOROLOG值。
这是一个整数值，表示自1840年12月31日以来的天数。

将日期转换为字符数据类型可以返回完整的日期，也可以返回数据类型长度所允许的全部日期。
但是，对所有字符数据类型来说，显示格式并不相同。

CHAR VARYING和CHARACTER VARYING数据类型以显示格式返回完整的日期。

例如，如果日期显示为MM/DD/YYYY，这些数据类型将日期返回为具有相同格式的字符串。

其他字符数据类型以ODBC日期格式的字符串形式返回日期(或其中的一部分)。

例如，如果日期显示为mm/dd/yyyy，这些数据类型将日期返回为yyyy - mm - dd格式的字符串。

因此，对于日期04/24/2004,CHAR数据类型返回'2'(年份的第一个字符)，而CHAR(8)返回' 2004-04 - '。

[#SQL #Cache](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%8C%E5%8D%81%E4%BA%8C%E7%AB%A0-sql%E5%87%BD%E6%95%B0-cast%E5%BC%88%E4%B8%80%E5%BC%89>