TRY\_PARSE, TRY\_CONVERT And TRY\_CAST In SQL Server

**Introduction**SQL SERVER 2005 discovered a new concept TRY-CATCH to handle unexpected behavior in SQL statements. We can use TRY-CATCH inside Stored Procedure, Triggers to handle exception and raise error message. Often we need to convert one data type value to other data type. For instance, there is a string(varchar) type variable and we need to convert its date or integer type - in case string value is blank and you are trying to convert to date type then SQL SERVER will throw you exception due to type casting. In order to avoid conversion problem SQL SERVER 2012 introduced the following three functions which helps us to handle type conversion exception. These functions are:

1. TRY\_PARSE
2. TRY\_CONVERT
3. TRY\_CAST

Let’s discuss how it can be implemented:

**TRY\_PARSE**

It converts string data type to target data type(Date or Numeric). For example, source data is string type and we need to covert to date type. If conversion attempt fails it returns NULL value.

**Syntax:***TRY\_PARSE (string\_value AS data\_type [ USING culture ])*

* **String\_value –** This is argument is source value which is NVARCHAR(4000) type.
* **Data\_type –** This argument is target data type either date or numeric.
* **Culture –**It is an optional argument which helps to convert the value to in Culture format. Suppose you want to display the date in French, then you need to pass culture type as ‘Fr-FR’. If you will not pass any valid culture name, then PARSE will raise an error.

**Examples**

1. **DECLARE** @ fakeDate **AS** **varchar**(10);
2. **DECLARE** @ realDate **AS** **VARCHAR**(10);
3. **SET** @fakeDate = 'iamnotadate';
4. **SET** @realDate = '13/09/2015;
5. **SELECT** TRY\_PARSE(@fakeDate **AS** **DATE**); --NULL
6. **SELECT** TRY\_PARSE(@realDate **AS** **DATE**); -- 2015-09-13
7. **SELECT** TRY\_PARSE(@realDate **AS** **DATE** USING 'Fr-FR'); -- 2015-09-13

First query attempts to convert a non-date to date type, it fails and returns NULL value. Second query successfully converts to date type. Third query also successfully converts but in French forma t(Here it is not showing any difference you can use other culture to see the difference).  
  
**TRY\_CONVERT**It converts value to specified data type and if conversion fails it returns NULL. For example, source value in string format and we need date/integer format. Then this will help us to achieve the same.  
  
**Syntax**: *TRY\_CONVERT ( data\_type [ ( length ) ], expression [, style ] )*

* **Data\_type -** The datatype into which to convert. Here length is an optional parameter which helps to get result in specified length.
* **Expression -**The value to be convert
* **Style -**It is an optional parameter which determines formatting. Suppose you want date format like “May, 18 2013” then you need pass style as 111. More on style visit [here](http://www.experts-exchange.com/articles/12315/SQL-Server-Date-Styles-formats-using-CONVERT.html).

**Examples**:

1. **DECLARE** @sampletext **AS** **VARCHAR**(10);
2. **SET** @sampletext = '123456';
3. **DECLARE** @ realDate **AS** **VARCHAR**(10);
4. **SET** @realDate = '13/09/2015’;
5. **SELECT** TRY\_CONVERT(**INT**, @sampletext); -- 123456
6. **SELECT** TRY\_CONVERT(DATETIME, @sampletext); -- NULL
7. **SELECT** TRY\_CONVERT(DATETIME, @realDate, 111); -- Sep, 13 2015

First query converts the text to integer successfully, but in second query the conversion fails and it returns NULL. Third query successfully converts string value to date type with specified formats.  
  
**TRY\_CAST**It converts value to specified data type and if conversion fails it returns NULL. For example, source value in string format and we need it in double/integer format. Then this will help us in achieving it.

**Syntax**: *TRY\_CAST ( expression AS data\_type [ ( length ) ] )*

* **Expression -** The source value which will go to cast.
* **Data\_type -** The target data type the source value will cast.
* **Length -**It is an optional parameter that specifies the length of target data type.

**Examples**

1. **DECLARE** @sampletext **AS** **VARCHAR**(10);
2. **SET** @sampletext = '123456';
4. **SELECT** TRY\_CAST(@sampletext **AS** **INT**); -- 123456
5. **SELECT** TRY\_CAST(@sampletext **AS** **DATE**); -- NULL

Firstly, query converts the text to integer successfully, but in second query the conversion fails and it returns NULL.  
  
**Analysis**Above we discussed about 3 types of conversion functions in SQL SERVER. TRY\_PARSE converts string to date/numeric only. TRY\_CONVERT converts source value to target data type. But here you can use *style*as optional parameter to format the date. TRY\_CAST do the same job like TRY\_CONVERT except style as extra parameter. Here you can’t pass style param.  
  
Secondly, TRY\_CONVERT is not supported in SQL Azure so use TRY\_CAST or TRY\_PARSE instead of that.  
  
Thirdly, TRY\_PARSE is not a SQL Native function rather it is a .NET runtime dependent function. It also facilitates to parse data culture wise which not there in TRY\_CONVERT and TRY\_CAST. It may create performance overhead but it tries its best to parse the data to specified data type. Let’s see the following example:

1. **SELECT** TRY\_PARSE('Saturday, 08 June 2013' **AS** DATETIME) -- June, 08 2013 00:00:00
2. **SELECT** TRY\_CONVERT(DATETIME, 'Saturday, 08 June 2013'); -- NULL

Above query successfully parsed to datetime, but it failed when we use convert. So TRY\_PARSE always do its best to parse the data.  
  
And one more thing I want to draw attention is TRY\_CONVERT and TRY\_CAST throw errors for explicit conversions - these conversions are not possible. More on data type explicit conversion visit [here](https://msdn.microsoft.com/en-us/library/ms191530.aspx). See the following examples:

1. **SELECT** TRY\_CONVERT(text, 4); -- Explicit conversion from data type int to text is not allowed.
2. **SELECT** TRY\_CAST(4 **AS** text); -- Explicit conversion from data type int to text is not allowed.

From above discussion we come to know that we need to use those functions wisely based on requirement, data, data type so that it will be helpful.

# SQL Server Date Styles (formats) using CONVERT()

**No Delimiter**

PATTERN STYLED DATE SYNTAX STYLE LENGTH

YYYY MM DD 20010223 convert(varchar, your\_data\_here ,112) 112 8

YY MM DD 010223 convert(varchar, your\_data\_here ,12) 12 6

**/ Slash delimited**

PATTERN STYLED DATE SYNTAX STYLE LENGTH

YYYY MM DD 2001/02/23 convert(varchar, your\_data\_here ,111) 111 10

YY MM DD 01/02/23 convert(varchar, your\_data\_here ,11) 11 8

MM DD YYYY 02/23/2001 convert(varchar, your\_data\_here ,101) 101 10

MM DD YY 02/23/01 convert(varchar, your\_data\_here ,1) 1 8

DD MM YYYY 23/02/2001 convert(varchar, your\_data\_here ,103) 103 10

DD MM YY 23/02/01 convert(varchar, your\_data\_here ,3) 3 8

**. Dot delimited**

PATTERN STYLED DATE SYNTAX STYLE LENGTH

YYYY MM DD 2001.02.23 convert(varchar, your\_data\_here ,102) 102 10

YY MM DD 01.02.23 convert(varchar, your\_data\_here ,2) 2 8

DD MM YYYY 23.02.2001 convert(varchar, your\_data\_here ,104) 104 10

DD MM YY 23.02.01 convert(varchar, your\_data\_here ,4) 4 8

**- Dash delimited**

PATTERN STYLED DATE SYNTAX STYLE LENGTH

YYYY MM DD 2001-02-23 04:05:06.007 convert(varchar, your\_data\_here ,121) 121 23

YYYY MM DD 2001-02-23 04:05:06 convert(varchar, your\_data\_here ,120) 120 19

MM DD YYYY 02-23-2001 convert(varchar, your\_data\_here ,110) 110 10

MM DD YY 02-23-01 convert(varchar, your\_data\_here ,10) 10 8

DD MM YYYY 23-02-2001 convert(varchar, your\_data\_here ,105) 105 10

DD MM YY 23-02-01 convert(varchar, your\_data\_here ,5) 5 8

**Space delimited**

PATTERN STYLED DATE SYNTAX STYLE LENGTH

MMM DD YYYY Feb 23 2001 4:05:06:007AM convert(varchar, your\_data\_here ,9) 9 26

MMM DD YYYY Feb 23 2001 4:05:06:007AM convert(varchar, your\_data\_here ,109) 109 26

MMM DD YYYY Feb 23 2001 4:05AM convert(varchar, your\_data\_here ,100) 100 19

MMM DD YYYY Feb 23, 2001 convert(varchar, your\_data\_here ,107) 107 12

MMM DD YY Feb 23, 01 convert(varchar, your\_data\_here ,7) 7 10

DD MMM YYYY 23 Feb 2001 04:05:06:007 convert(varchar, your\_data\_here ,13) 13 24

DD MMM YYYY 23 Feb 2001 04:05:06:007 convert(varchar, your\_data\_here ,113) 113 24

DD MM YYYY 23 Feb 2001 convert(varchar, your\_data\_here ,106) 106 11

DD MM YY 23 Feb 01 convert(varchar, your\_data\_here ,6) 6 9

**Time Only**

PATTERN STYLED DATE SYNTAX STYLE LENGTH

hh:mm:ss:ms 04:05:06:007 convert(varchar, your\_data\_here ,14) 14 12

hh:mm:ss:ms 04:05:06:007 convert(varchar, your\_data\_here ,114) 114 12

hh:mm:ss 04:05:06 convert(varchar, your\_data\_here ,8) 8 8

hh:mm:ss 04:05:06 convert(varchar, your\_data\_here ,108) 108 8

**DIY Table:**

DELIMITED STARTS PATTERN STYLED DATE SYNTAX STYLE LENGTH

YYYY YYYY MM DD 20010223 convert(varchar, your\_data\_here ,112) 112 8

YY YY MM DD 010223 convert(varchar, your\_data\_here ,12) 12 6

slash YYYY YYYY MM DD 2001/02/23 convert(varchar, your\_data\_here ,111) 111 10

slash YY YY MM DD 01/02/23 convert(varchar, your\_data\_here ,11) 11 8

slash MM MM DD YYYY 02/23/2001 convert(varchar, your\_data\_here ,101) 101 10

slash MM MM DD YY 02/23/01 convert(varchar, your\_data\_here ,1) 1 8

slash DD DD MM YYYY 23/02/2001 convert(varchar, your\_data\_here ,103) 103 10

slash DD DD MM YY 23/02/01 convert(varchar, your\_data\_here ,3) 3 8

dot YYYY YYYY MM DD 2001.02.23 convert(varchar, your\_data\_here ,102) 102 10

dot YY YY MM DD 01.02.23 convert(varchar, your\_data\_here ,2) 2 8

dot DD DD MM YYYY 23.02.2001 convert(varchar, your\_data\_here ,104) 104 10

dot DD DD MM YY 23.02.01 convert(varchar, your\_data\_here ,4) 4 8

dash YYYY YYYY MM DD 2001-02-23 04:05:06.007 convert(varchar, your\_data\_here ,121) 121 23

dash YYYY YYYY MM DD 2001-02-23 04:05:06 convert(varchar, your\_data\_here ,120) 120 19

dash MM MM DD YYYY 02-23-2001 convert(varchar, your\_data\_here ,110) 110 10

dash MM MM DD YY 02-23-01 convert(varchar, your\_data\_here ,10) 10 8

dash DD DD MM YYYY 23-02-2001 convert(varchar, your\_data\_here ,105) 105 10

dash DD DD MM YY 23-02-01 convert(varchar, your\_data\_here ,5) 5 8

space MMM MMM DD YYYY Feb 23 2001 4:05:06:007AM convert(varchar, your\_data\_here ,9) 9 26

space MMM MMM DD YYYY Feb 23 2001 4:05:06:007AM convert(varchar, your\_data\_here ,109) 109 26

space MMM MMM DD YYYY Feb 23 2001 4:05AM convert(varchar, your\_data\_here ,100) 100 19

space MMM MMM DD YYYY Feb 23, 2001 convert(varchar, your\_data\_here ,107) 107 12

space MMM MMM DD YY Feb 23, 01 convert(varchar, your\_data\_here ,7) 7 10

space DD DD MMM YYYY 23 Feb 2001 04:05:06:007 convert(varchar, your\_data\_here ,13) 13 24

space DD DD MMM YYYY 23 Feb 2001 04:05:06:007 convert(varchar, your\_data\_here ,113) 113 24

space DD DD MM YYYY 23 Feb 2001 convert(varchar, your\_data\_here ,106) 106 11

space DD DD MM YY 23 Feb 01 convert(varchar, your\_data\_here ,6) 6 9

colon hh hh:mm:ss:ms 04:05:06:007 convert(varchar, your\_data\_here ,14) 14 12

colon hh hh:mm:ss:ms 04:05:06:007 convert(varchar, your\_data\_here ,114) 114 12

colon hh hh:mm:ss 04:05:06 convert(varchar, your\_data\_here ,8) 8 8

colon hh hh:mm:ss 04:05:06 convert(varchar, your\_data\_here ,108) 108 8