

Building dates from parts

TIME SERIES ANALYSIS IN SQL SERVER



Kevin Feasel
CTO, Envizage

Dates from parts

```
DATEFROMPARTS(year, month, day)
```

```
TIMEFROMPARTS(hour, minute, second, fraction, precision)
```

```
DATETIMEFROMPARTS(year, month, day, hour, minute, second, ms)
```

```
DATETIME2FROMPARTS(year, month, day, hour, minute, second, fraction, precision)
```

```
SMALLDATETIMEFROMPARTS(year, month, day, hour, minute)
```

```
DATETIMEOFFSETFROMPARTS(year, month, day, hour, minute, second, fraction, hour_offset,  
minute_offset, precision)
```

Dates and times together

Because the 'DATETIME' data type is only precise to a three millisecond period. SQL Server rounds our result to the nearest allowed value, which is 997 milliseconds.

The three 'DATETIME2' types allow us to choose our desired precision.

SELECT

```
DATETIMEFROMPARTS(1918, 11, 11, 05, 45, 17, 995) AS DT,  
DATETIME2FROMPARTS(1918, 11, 11, 05, 45, 17, 0, 0) AS DT20,  
DATETIME2FROMPARTS(1918, 11, 11, 05, 45, 17, 995, 3) AS DT23,  
DATETIME2FROMPARTS(1918, 11, 11, 05, 45, 17, 9951234, 7) AS DT27;
```

DT	DT20	DT23	DT27
1918-11-11 05:45:17.997	1918-11-11 05:45:17	1918-11-11 05:45:17.995	1918-11-11 05:45:17.9951234

Working with offsets

SELECT

```
DATETIMEOFFSETFROMPARTS(2009, 08, 14, 21,  
    00, 00, 0, 5, 30, 0) AS IST,  
DATETIMEOFFSETFROMPARTS(2009, 08, 14, 21,  
    00, 00, 0, 5, 30, 0) pass in 5 hours and 30 minutes as the offset values  
AT TIME ZONE 'UTC' AS UTC;
```

IST	UTC	
2009-08-14 21:00:00 +05:30	2009-08-14 15:30:00 +00:00	

Gotchas when working with parts

```
DATEFROMPARTS(1999, 12, NULL)
```

```
DATEFROMPARTS(10000, 01, 01)
```

```
DATETIME2FROMPARTS(1918, 11, 11, 05, 45, 17, 995, 0)
```

```
NULL
```

Cannot construct data type date, some of the arguments have values which are not valid.

Cannot construct data type datetime2, some of the arguments have values which are not valid.

Let's practice!

TIME SERIES ANALYSIS IN SQL SERVER

Translating date strings

TIME SERIES ANALYSIS IN SQL SERVER



Kevin Feasel
CTO, Envizage

Casting strings

```
SELECT  
    CAST('09/14/99' AS DATE) AS USDate;
```

USDate	
1999-09-14	

Converting Strings

```
SELECT  
    CONVERT(DATETIME2(3),  
        'April 4, 2019 11:52:29.998 PM') AS April4
```

April4	
2019-04-04 23:52:29.998	

Parsing strings

```
SELECT  
    PARSE( '25 Dezember 2014' AS DATE  
        USING 'de-de' ) AS Weinachten;
```

Weinachten	
2014-12-25	

The cost of parsing

Function	Conversions Per Second	
<code>CONVERT()</code>	251,997	
<code>CAST()</code>	240,347	
<code>PARSE()</code>	12,620	

Setting languages

```
SET LANGUAGE 'FRENCH'
```

```
DECLARE
```

```
    @FrenchDate NVARCHAR(30) = N'18 avril 2019',
```

```
    @FrenchNumberDate NVARCHAR(30) = N'18/4/2019';
```

```
SELECT
```

```
    CAST(@FrenchDate AS DATETIME),
```

```
    CAST(@FrenchNumberDate AS DATETIME);
```

```
2019-04-18 00:00:00.000
```

Let's practice!

TIME SERIES ANALYSIS IN SQL SERVER

Working with offsets

TIME SERIES ANALYSIS IN SQL SERVER



Kevin Feasel
CTO, Envizage

Anatomy of a DATETIMEOFFSET

Components

Date Part	Example
Date	2019-04-10
Time	12:59:02.3908505
UTC Offset	-04:00

Anatomy of a DATETIMEOFFSET

Components

Date Part	Example
Date	2019-04-10
Time	12:59:02.3908505
UTC Offset	-04:00

Display

```
2019-04-10 12:59:02.3908505 -04:00
```


Changing offsets

If you pass in a 'DATETIME' or 'DATETIME2', 'SWITCHOFFSET()' assumes you are in UTC.

```
DECLARE @SomeDate DATETIMEOFFSET =  
    '2019-04-10 12:59:02.3908505 -04:00';  
  
SELECT  
    SWITCHOFFSET(@SomeDate, '-07:00') AS LATime;
```

LATime
2019-04-10 09:59:02.3908505 -07:00

Converting to DATETIMEOFFSET

```
DECLARE @SomeDate DATETIME2(3) =  
    '2019-04-10 12:59:02.390';  
  
SELECT  
    TODATETIMEOFFSET(@SomeDate, '-04:00') AS EDT;
```

EDT	
2019-04-10 12:59:02.390 -04:00	

Time zone swaps with TODATETIMEOFFSET

```
DECLARE @SomeDate DATETIME2(3) =  
    '2016-09-04 02:28:29.681';
```

```
SELECT  
    TODATETIMEOFFSET(  
        DATEADD(HOUR, 7, @SomeDate),  
        '+02:00') AS BonnTime;
```

BonnTime
2016-09-04 09:28:29.681 +02:00

Discovering time zones

```
SELECT
    tzi.name,
    tzi.current_utc_offset,
    tzi.is_currently_dst
FROM sys.time_zone_info tzi
WHERE
    tzi.name LIKE '%Time Zone%';
```

name	current_utc_offset	is_currently_dst	
Russia Time Zone 3	+04:00	0	
Russia Time Zone 10	+11:00	0	
Russia Time Zone 11	+12:00	0	

Let's practice!

TIME SERIES ANALYSIS IN SQL SERVER

Handling invalid dates

TIME SERIES ANALYSIS IN SQL SERVER



Kevin Feasel
CTO

Error-safe date conversion functions

"Unsafe" Functions

CAST()

CONVERT()

PARSE()

Safe Functions

handle invalid dates by
converting them to NULL

TRY_CAST()

TRY_CONVERT()

TRY_PARSE()

When everything goes right

```
SELECT
    PARSE( '01/08/2019' AS DATE USING 'en-us' ) AS January8US,
    PARSE( '01/08/2019' AS DATE USING 'fr-fr' ) AS August1FR;

GO
```

Results:

January8US	August1FR	
2019-01-08	2019-08-01	

When everything goes wrong

```
SELECT
    PARSE( '01/13/2019' AS DATE USING 'en-us' ) AS January13US,
    PARSE( '01/13/2019' AS DATE USING 'fr-fr' ) AS Smarch1FR;
GO
```

Msg 9819, Level 16, State 1, Line 1

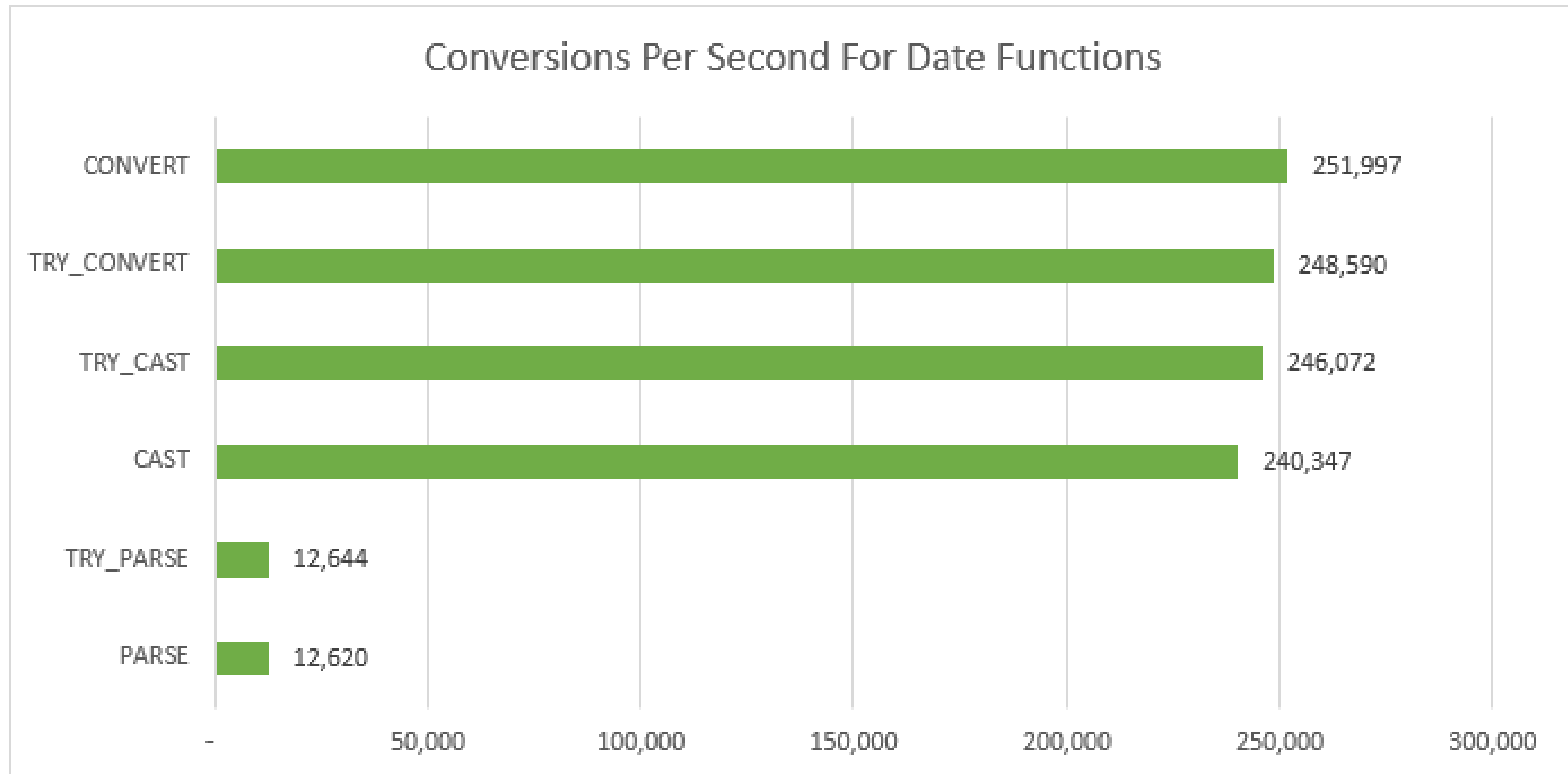
Error converting string value '01/13/2019' into data type date using culture 'fr-fr'.

Doing right when everything goes wrong

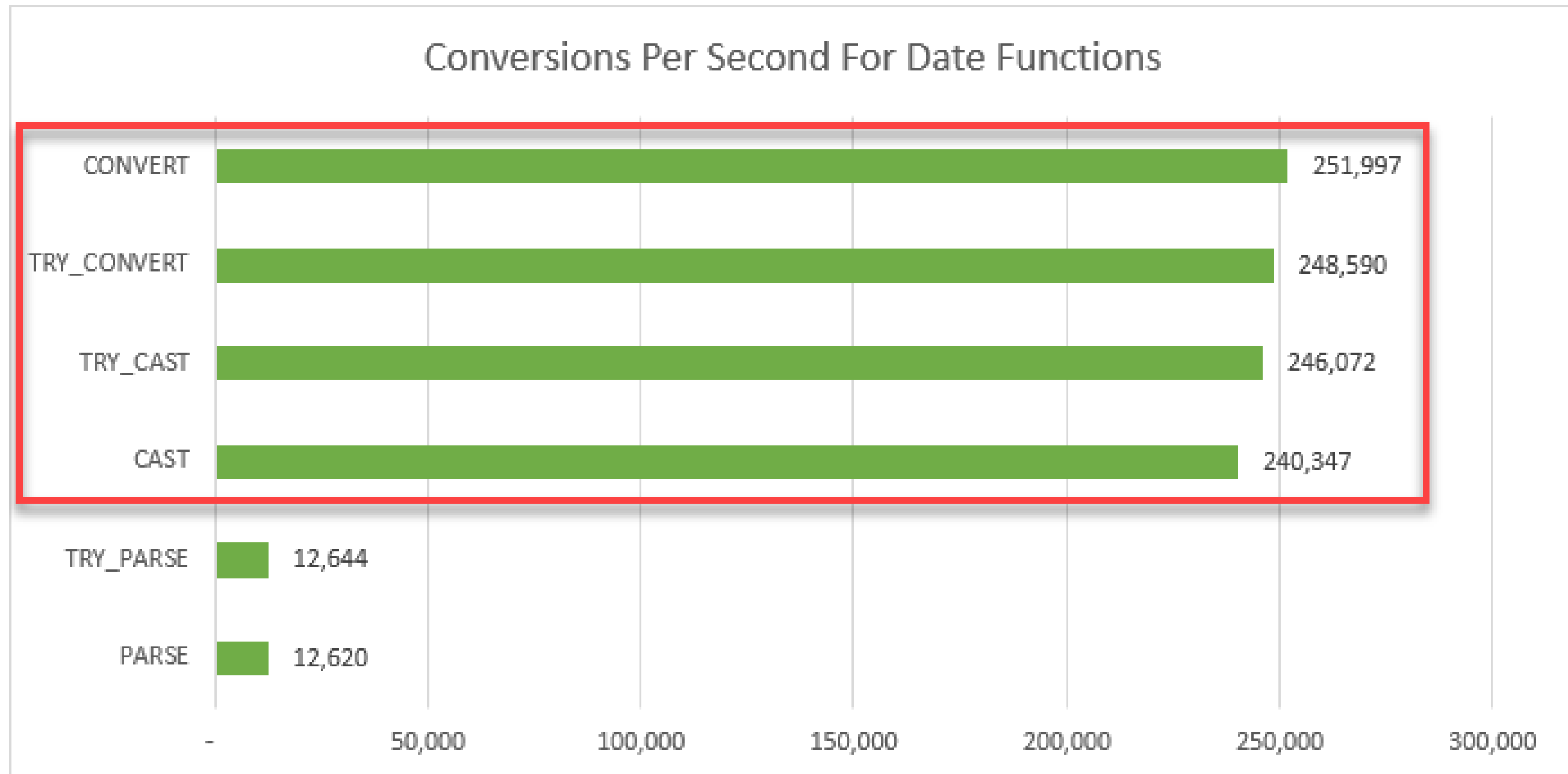
```
SELECT
    TRY_PARSE( '01/13/2019' AS DATE USING 'en-us' ) AS January13US,
    TRY_PARSE( '01/13/2019' AS DATE USING 'fr-fr' ) AS Smarch1FR;
GO
```

January13US	Smarch1FR	
2019-01-13	NULL	

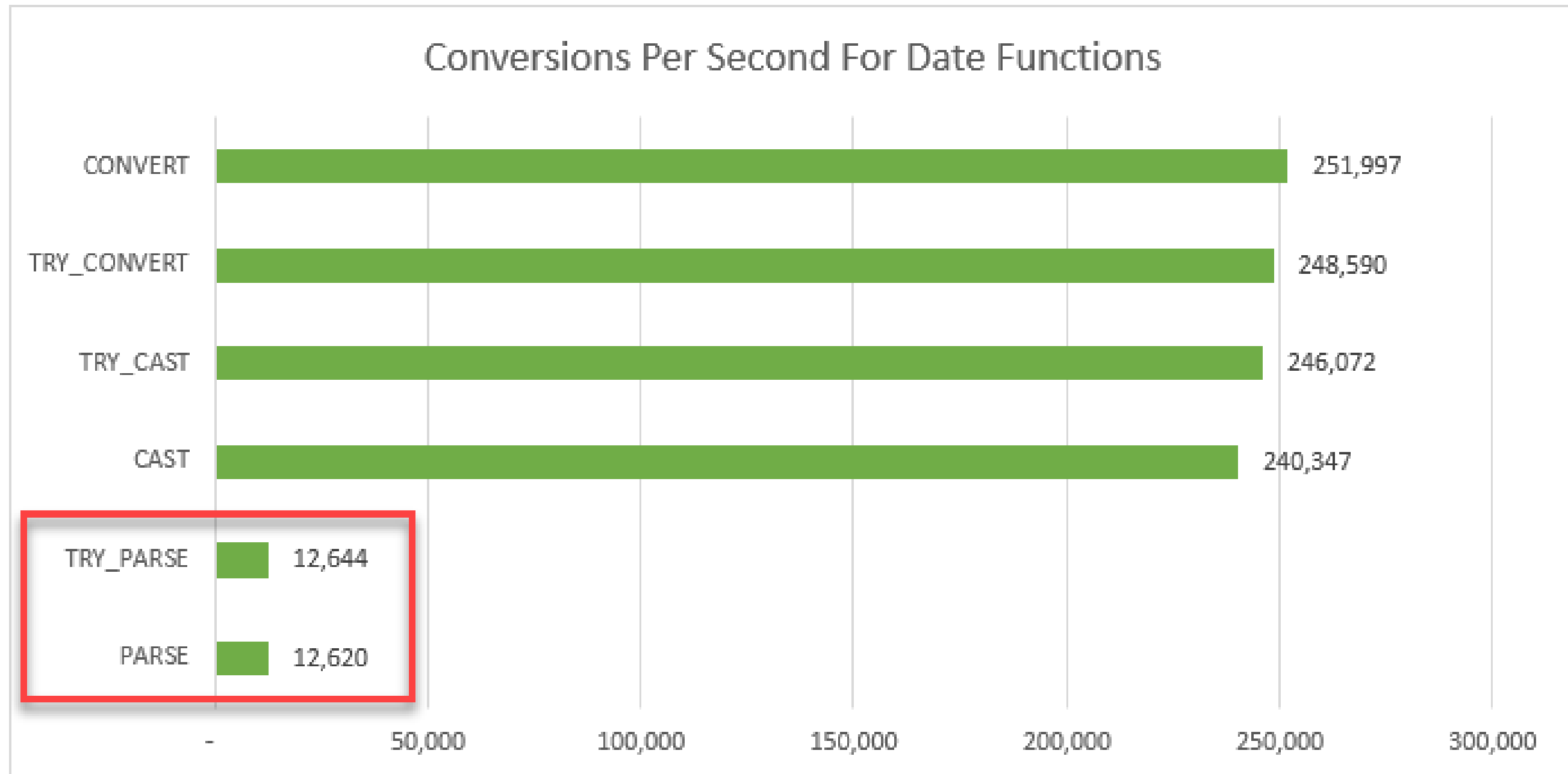
The cost of safety



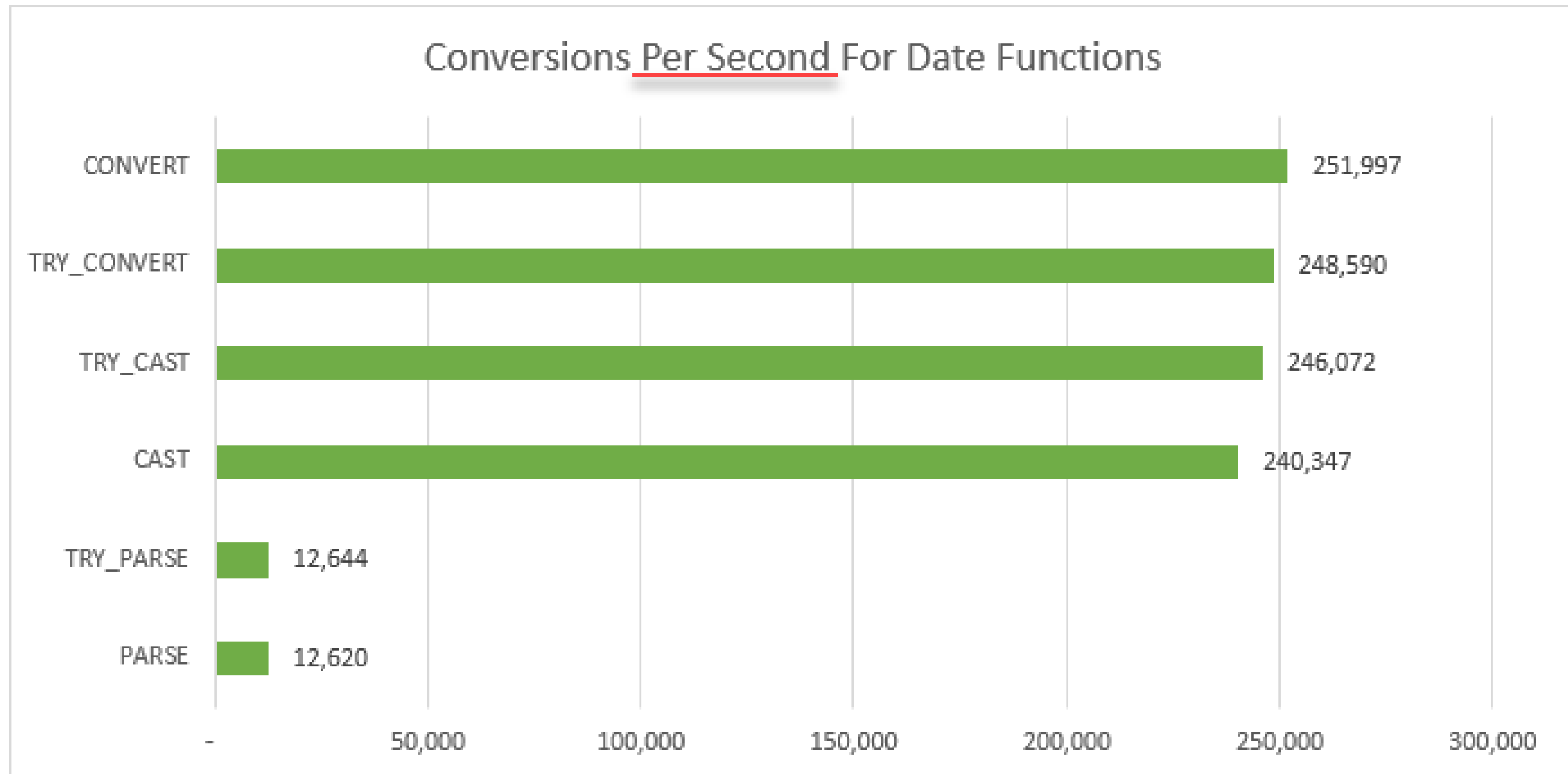
The cost of safety



The cost of safety



The cost of safety



Let's practice

TIME SERIES ANALYSIS IN SQL SERVER