

SQL Server DateTime conversion failure

Asked 16 years, 4 months ago Modified 6 years, 3 months ago Viewed 14k times



I have a large table with 1 million+ records. Unfortunately, the person who created the table decided to put dates in a `varchar(50)` field.

3



I need to do a simple date comparison -

```
datediff(dd, convert(datetime, lastUpdate, 100), getDate()) < 31
```



But it fails on the `convert()`:

Conversion failed when converting datetime from character string.

Apparently there is something in that field it doesn't like, and since there are so many records, I can't tell just by looking at it. How can I properly sanitize the entire date field so it does not fail on the `convert()`? Here is what I have now:

```
select count(*)
from MyTable
where
    isdate(lastUpdate) > 0
    and datediff(dd, convert(datetime, lastUpdate, 100), getDate()) < 31
```

[@SQLMenace](#)

I'm not concerned about performance in this case. This is going to be a one time query. Changing the table to a datetime field is not an option.

[@Jon Limjap](#)

I've tried adding the third argument, and it makes no difference.

[@SQLMenace](#)

The problem is most likely how the data is stored, there are only two safe formats; ISO YYYYMMDD; ISO 8601 yyyy-mm-dd Thh:mm:ss:mmm (no spaces)

Wouldn't the `isdate()` check take care of this?

I don't have a need for 100% accuracy. I just want to get most of the records that are from the last 30 days.

[@SQLMenace](#)

```
select isdate('20080131') -- returns 1
select isdate('01312008') -- returns 0
```

[@Brian Schkerke](#)

Place the CASE and ISDATE inside the CONVERT() function.

Thanks! That did it.

sql-server

datetime

Share

Improve this question

Follow

edited May 23, 2017 at 11:46



Community Bot
1 • 1

asked Aug 26, 2008 at 14:01



Seibar
70.2k • 38 • 93 • 100

You haven't mentioned what formats the dates are in for the varchar field. Are they consistently formatted? Knowing the format would help out quite a bit with solving the problem. – [Kibbee](#) Aug 26, 2008 at 14:33

10 Answers

Sorted by: Highest score (default)



Place the `CASE` and `ISDATE` inside the `CONVERT()` function.

9

```
SELECT COUNT(*) FROM MyTable
WHERE
    DATEDIFF(dd, CONVERT(DATETIME, CASE IsDate(lastUpdate)
        WHEN 1 THEN lastUpdate
        ELSE '12-30-1899'
    END), GetDate()) < 31
```



Replace `'12-30-1899'` with the default date of your choice.



Share

Improve this answer

edited Aug 26, 2018 at 10:48



Ansgar Wiechers
200k • 26 • 280 • 354

answered Aug 26, 2008 at 14:27



Disbeliever
1,143 • 7 • 15



3



How about writing a cursor to loop through the contents, attempting the cast for each entry? When an error occurs, output the primary key or other identifying details for the problem record. I can't think of a set-based way to do this.



Not totally setbased but if only 3 rows out of 1 million are bad it will save you a lot of time

```
select * into BadDates
from Yourtable
where isdate(lastUpdate) = 0

select * into GoodDates
from Yourtable
where isdate(lastUpdate) = 1
```

then just look at the BadDates table and fix that

Share Improve this answer Follow

answered Aug 26, 2008 at 14:32



SQLMenace

135k ● 25 ● 211 ● 225



2



The ISDATE() would take care of the rows which were not formatted properly if it were indeed being executed first. However, if you look at the execution plan you'll probably find that the DATEDIFF predicate is being applied first - thus the cause of your pain.

If you're using SQL Server Management Studio hit **CTRL** + **L** to view the estimated execution plan for a particular query.



Remember, SQL isn't a procedural language and short circuiting logic may work, but only if you're careful in how you apply it.

Share

Improve this answer

Follow

edited Aug 20, 2015 at 22:18



Valery Viktorovsky

6,696 ● 3 ● 44 ● 49

answered Aug 26, 2008 at 14:32



Disbeliever

1,143 ● 7 ● 15



1

How about writing a cursor to loop through the contents, attempting the cast for each entry?



When an error occurs, output the primary key or other identifying details for the problem record.



I can't think of a set-based way to do this.



Edit - ah yes, I forgot about ISDATE(). Definitely a better approach than using a cursor. +1 to SQLMenace.

Share

edited Aug 26, 2008 at 14:35

answered Aug 26, 2008 at 14:24

Improve this answer

Follow



Ian Nelson

58.6k ● 20 ● 78 ● 104



In your convert call, you need to specify a third style parameter, e.g., the format of the datetimes that are stored as varchar, as specified in this document: [CAST and CONVERT \(T-SQL\)](#).

0



Share Improve this answer Follow

answered Aug 26, 2008 at 14:13



Jon Limjap

95.3k ● 15 ● 103 ● 153



Print out the records. Give the hardcopy to the idiot who decided to use a varchar(50) and ask them to find the problem record.

0

Next time they might just see the point of choosing an appropriate data type.



Share Improve this answer Follow

answered Aug 26, 2008 at 14:14



Ian Nelson

58.6k ● 20 ● 78 ● 104



The problem is most likely how the data is stored, there are only two safe formats

0

ISO YYYYMMDD



ISO 8601 yyyy-mm-dd Thh:mm:ss:mmm(no spaces)

these will work no matter what your language is.



You might need to do a SET DATEFORMAT YMD (or whatever the data is stored as) to make it work

Share Improve this answer Follow

answered Aug 26, 2008 at 14:20



SQLMenace

135k ● 25 ● 211 ● 225



0

Wouldn't the isdate() check take care of this?

Run this to see what happens



```
select isdate('20080131')
select isdate('01312008')
```



Share Improve this answer Follow

answered Aug 26, 2008 at 14:27



SQLMenace

135k ● 25 ● 211 ● 225



0

I am sure that changing the table/column might not be an option due to any legacy system requirements, but have you thought about creating a view which has the date conversion logic built in, if you are using a more recent version of sql, then you can possibly even use an indexed view?



Share Improve this answer Follow

answered Mar 14, 2009 at 15:11



chris raethke

435 ● 3 ● 8



0

I would suggest cleaning up the mess and changing the column to a datetime because doing stuff like this



```
WHERE datediff(dd, convert(datetime, lastUpdate), getDate()) < 31
```



cannot use an index and it will be many times slower than if you had a datetime column, and did

```
where lastUpdate > getDate() -31
```

You also need to take into account hours and seconds of course

Share

Improve this answer

Follow

edited Aug 28, 2011 at 9:41



Gumbo

655k ● 112 ● 790 ● 851

answered Aug 26, 2008 at 14:12



SQLMenace

135k ● 25 ● 211 ● 225

