How are we doing? Please help us improve Stack Overflow. **Take our short survey**

Rounding SQL DateTime to midnight

Asked 7 years, 9 months ago   Active 1 month ago   Viewed 126k times

I am having a small problem with my SQL query. I'm using the GETDATE function, however, let's say I execute the script at 5PM, it will
pull up records between 12/12/2011 5PM to 12/18/2011 5PM. How can I make it pull up records for the whole entire 12/12/2011 -
12/18/2011 basically ignore time.

71

My script:

9

```
WHERE Orders.OrderStatus = 'Shipped'
AND Orders.ShipDate > (GETDATE()-6)
```

| sql | 🍄 sql-server | sql-server-2005 | getdate |

edited Dec 18 '11 at 22:37   asked Dec 18 '11 at 22:24
DaveShaw   **henryaaron**
42.8k   14   94   127   **3,248**   18   47   75

9 Answers

In SQL Server 2008 and newer you can cast the `DateTime` to a `Date`, which removes the time element.

102

```
WHERE Orders.OrderStatus = 'Shipped'
AND Orders.ShipDate >= (cast(GETDATE()-6 as date))
```

In SQL Server 2005 and below you can use:

✓

answered Dec 18 '11 at 22:29

DaveShaw
42.8k   14   94   127

1   I got this Type date is not a defined system type. –   henryaaron   Dec 18 '11 at 22:31

2   I guess you're not using SQL 2008 then :) – DaveShaw Dec 18 '11 at 22:31

      @user1090389 that's why I put the string conversion option ;D – Bassam Mehanni Dec 18 '11 at 22:33

      @DaveShaw - Missing a DATEADD in there – MatBailie Dec 18 '11 at 22:38

      It say Incorrect Syntax near "," –   henryaaron   Dec 18 '11 at 22:40

Here is the simplest thing I've found

46

```
-- Midnight floor of current date

SELECT Convert(DateTime, DATEDIFF(DAY, 0, GETDATE()))
```

The DATEDIFF returns the integer number of days before or since 1900-1-1, and the Convert Datetime obligingly brings it back to that date at midnight.

Since DateDiff returns an integer you can use add or subtract days to get the right offset.

```
SELECT Convert(DateTime, DATEDIFF(DAY, 0, GETDATE()) + @dayOffset)
```

This isn't rounding this is truncating...But I think that is what is being asked. (To round add one and truncate...and that's not rounding either, that the ceiling, but again most likely what you want. To really round add .5 (does that work?) and truncate.

It turns out you can add .5 to GetDate() and it works as expected.

```
-- Round Current time to midnight today or midnight tomorrow

SELECT Convert(DateTime, DATEDIFF(DAY, 0, GETDATE() + .5))
```

This works in 2k5 just tested it. `where [ScanDate] >= convert(datetime, datediff(day, 0, getdate())) and [ScanDate] < convert(datetime, datediff(day, -1, getdate()))` — nulltron Mar 21 at 20:19 ✏

8

```sql
--
-- SQL DATEDIFF getting midnight time parts
--
SELECT GETDATE() AS Now,
    Convert(DateTime, DATEDIFF(DAY, 0, GETDATE())) AS MidnightToday,
    Convert(DateTime, DATEDIFF(DAY, -1, GETDATE())) AS MidnightNextDay,
    Convert(DateTime, DATEDIFF(DAY, 1, GETDATE())) AS MidnightYesterDay
go
Now                     MidnightToday          MidnightNextDay          MidnightYesterDay
-------------------     --------------------   ---------------------    -------------------
--
 8/27/2014 4:30:22 PM   8/27/2014 12:00:00 AM  8/28/2014 12:00:00 AM   8/26/2014 12:00:00
AM
```

4

```sql
SELECT getdate()
```

Result: 2012-12-14 16:03:33.360

```sql
SELECT convert(datetime,convert(bigint, getdate()))
```

Result 2012-12-15 00:00:00.000

2   Try to explain your answer and format your code. – JSuar Dec 18 '12 at 2:08

1   Thanks for the feedback. What I was trying to highlight that the convert to bigint and back does the rounding for you. – Jeremy Atkinson Dec 18 '12 at 21:50

     This code rounds it up to the midnight at the end of the day if the time is after midday which causes this to be wrong for half the day. – ChrisM Jul 30 at 11:10

     Using this method can cause performance impacts later with larger datasets. – PerpetualJ Aug 2 at 15:27

As @BassamMehanni mentioned, you can cast as DATE in SQL Server 2008 onwards...

3

```sql
SELECT
    *
FROM
   yourTable
WHERE
      dateField >= CAST(GetDate() - 6 AS DATE)
  AND dateField <  CAST(GetDate() + 1 AS DATE)
```

The second condition can actually be just `GetDate()`, but I'm showing this format as an example of `Less Than DateX` to avoid having to cast the dateField to a DATE as well, thus massively improving performance.

If you're on 2005 or under, you can use this...

```sql
SELECT
    *
FROM
   yourTable
WHERE
      dateField >= DATEADD(DAY, DATEDIFF(DAY, 0, GetDate()) - 6, 0)
  AND dateField <  DATEADD(DAY, DATEDIFF(DAY, 0, GetDate()) + 1, 0)
```

Try using this.

3

```
WHERE Orders.OrderStatus = 'Shipped'
AND Orders.ShipDate >= CONVERT(DATE, GETDATE())
```

edited May 7 '14 at 9:39   answered May 7 '14 at 9:19
09stephenb   UttamG
4,823   11   44   78   31   2

This might look cheap but it's working for me

1

```
SELECT CONVERT(DATETIME,LEFT(CONVERT(VARCHAR,@dateFieldOrVariable,101),10)+' 00:00:00.000')
```

answered Jan 18 '17 at 16:41
Jean-Louis Gervais
16   2

I usually do

0

```
SELECT *
FROM MyTable
WHERE CONVERT(VARCHAR, MyTable.dateField, 101) = CONVERT(VARCHAR, GETDATE(), 101)
```

if you are using SQL SERVER 2008, you can do

```
SELECT *
FROM MyTable
WHERE CAST(MyTable.dateField AS DATE) = CAST(GETDATE() AS DATE)
```

answered Dec 18 '11 at 22:28

Bassam Mehanni
13.1k　1　27　37

---

3 If you use the first example you *destory* the optimiser's ability to use indexes, etc. Using String functions to do Date arithmetic and comparisons is a very bad idea. It shouldn't ever (imo) even be mentioned, it's that poor an option. – MatBailie Dec 18 '11 at 22:30 ✏

You don't have to if you are using SQL Server 2008, otherwise, I am not sure how else you could do it in SQL Server 2000/2005, an example would be appreciated, thanks. – Bassam Mehanni Dec 18 '11 at 22:32

@Dems Yes an example would be appreciated... – henryaaron Dec 18 '11 at 22:33

To be fair. Both options destroy the optmizer's ability to use indexes. You should perform and functionality on the filter predicate column. This includes casts. – pim May 31 '17 at 17:12

You could round down the time.

Using ROUND below will round it down to midnight.

```
WHERE Orders.OrderStatus = 'Shipped'
AND Orders.ShipDate > CONVERT(datetime, (ROUND(convert(float, getdate()-6.5),0)))
```

edited Jul 30 at 11:07　　　　　answered Jul 30 at 11:00

ChrisM
305　2　12