Floor a date in SQL server

68

33

In SQL Server, how do I "floor" a DATETIME to the second/minute/hour/day/year?

Let's say that I have a date of **2008-09-17 12:56:53.430**, then the output of flooring should be:

- Year: 2008-01-01 00:00:00.000
- Month: 2008-09-01 00:00:00.000
- Day: 2008-09-17 00:00:00.000
- Hour: 2008-09-17 12:00:00.000
- Minute: 2008-09-17 12:56:00.000
- Second: 2008-09-17 12:56:53.000

sql-server tsql

edited May 6 '13 at 17:09
Brian Webster
20.9k 40 130 208

asked Sep 17 '08 at 17:01
Portman
21.4k 22 74 97

9 Answers

95

The key is to use DATEADD and DATEDIFF along with the appropriate SQL timespan enumeration.

```
declare @datetime datetime;
set @datetime = getdate();
select @datetime;
select dateadd(year,datediff(year,0,@datetime),0);
```

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up OR SIGN IN WITH Google Facebook

```
select dateadd(week,datediff(week,0,@datetime),-1); --Beginning of week is Sunday
select dateadd(week,datediff(week,0,@datetime),0); --Beginning of week is Monday
```

Note that when you are flooring by the second, you will often get an arithmetic overflow if you use 0. So pick a known value that is guaranteed to be lower than the datetime you are attempting to floor.

edited Dec 7 '17 at 2:17

Davin Studer
699 3 10 26

answered Sep 17 '08 at 17:01

Portman
21.4k 22 74 97

1 The date you calculate your offset from doesn't need to be in the past. Any date will work, provided it is itself 'FLOOR'ed to the interval in questions. If the base date is in the future, you just get a negative offset value... – MatBailie Oct 6 '09 at 16:20

To floor to the week use this if Sunday is the first day of the week ... select dateadd(week,datediff(week,0,@datetime),-1) – Davin Studer Jul 12 '13 at 18:30

Use this if Monday is the first day of the week ... select dateadd(week,datediff(week,0,@datetime),0) – Davin Studer Jul 12 '13 at 18:33

Not sure why this answer isn't at the top, this is a very efficient for date formatting in Microsoft SQL Server – Adam May 11 '16 at 10:58

28

In SQL Server here's a little trick to do that:

```
SELECT CAST(FLOOR(CAST(CURRENT_TIMESTAMP AS float)) AS DATETIME)
```

You cast the DateTime into a float, which represents the Date as the integer portion and the Time as the fraction of a day that's passed. Chop off that decimal portion, then cast that back to a DateTime, and you've got midnight at the beginning of that day.

This is probably more efficient than all the DATEADD and DATEDIFF stuff. It's certainly way easier to type.

answered Sep 17 '08 at 19:08

Chris Wuestefeld
2,338 1 17 22

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up OR SIGN IN WITH Google Facebook

AS TIME) or CAST([field] as DATE) – Rik Jan 23 '14 at 12:29

11

Expanding upon the Convert/Cast solution, in Microsoft SQL Server 2008 you can do the following:

```
cast(cast(getdate() as date) as datetime)
```

Just replace `getdate()` with any column which is a datetime.

There are no strings involved in this conversion.

This is ok for ad-hoc queries or updates, but for key joins or heavily used processing it may be better to handle the conversion within the processing or redefine the tables to have appropriate keys and data.

In 2005, you can use the messier floor: `cast(floor(cast(getdate() as float)) as datetime)`

I don't think that uses string conversion either, but I can't speak to comparing actual efficiency versus armchair estimates.

edited Apr 19 '12 at 8:45
Charles Menguy
27.1k 16 81 108

answered Apr 18 '12 at 15:20
Moe Cazzell
111 1 4

6

I've used @Portman's answer many times over the years as a reference when flooring dates and have moved its working into a function which you may find useful.

I make no claims to its performance and merely provide it as a tool for the user.

I ask that, if you do decide to upvote this answer, please also upvote @Portman's answer, as my code is a derivative of his.

```
IF OBJECT_ID('fn_FloorDate') IS NOT NULL DROP FUNCTION fn_FloorDate
SET ANSI_NULLS OFF
GO
```

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up OR SIGN IN WITH Google Facebook ✕

```
RETURNS DATETIME
AS
BEGIN
  IF (@Date IS NULL)
    SET @Date = GETDATE();

  RETURN
  CASE
    WHEN LOWER(@DatePart) = 'year' THEN DATEADD(YEAR, DATEDIFF(YEAR, 0, @Date), 0)
    WHEN LOWER(@DatePart) = 'month' THEN DATEADD(MONTH, DATEDIFF(MONTH, 0, @Date), 0)
    WHEN LOWER(@DatePart) = 'day' THEN DATEADD(DAY, DATEDIFF(DAY, 0, @Date), 0)
    WHEN LOWER(@DatePart) = 'hour' THEN DATEADD(HOUR, DATEDIFF(HOUR, 0, @Date), 0)
    WHEN LOWER(@DatePart) = 'minute' THEN DATEADD(MINUTE, DATEDIFF(MINUTE, 0, @Date), 0)
    WHEN LOWER(@DatePart) = 'second' THEN DATEADD(SECOND, DATEDIFF(SECOND, '2000-01-01',
@Date), '2000-01-01')
    ELSE DATEADD(DAY, DATEDIFF(DAY, 0, @Date), 0)
  END;
END
```

Usage:

```
DECLARE @date DATETIME;
SET @date = '2008-09-17 12:56:53.430';

SELECT
  @date AS [Now],--2008-09-17 12:56:53.430
  dbo.fn_FloorDate(@date, 'year') AS [Year],--2008-01-01 00:00:00.000
  dbo.fn_FloorDate(default, default) AS [NoParams],--2013-11-05 00:00:00.000
  dbo.fn_FloorDate(@date, default) AS [ShouldBeDay],--2008-09-17 00:00:00.000
  dbo.fn_FloorDate(@date, 'month') AS [Month],--2008-09-01 00:00:00.000
  dbo.fn_FloorDate(@date, 'day') AS [Day],--2008-09-17 00:00:00.000
  dbo.fn_FloorDate(@date, 'hour') AS [Hour],--2008-09-17 12:00:00.000
  dbo.fn_FloorDate(@date, 'minute') AS [Minute],--2008-09-17 12:56:00.000
  dbo.fn_FloorDate(@date, 'second') AS [Second];--2008-09-17 12:56:53.000
```

edited May 23 '17 at 11:47

Community ♦

1 1

answered Jul 25 '13 at 14:22

Dan Atkinson

9,144 10 74 105

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up OR SIGN IN WITH Google Facebook ✕

Joel Coehoorn
316k 98 502 737

2 We've found that CONVERT() can be anywhere from 10% to 5x less performant than dateadd/datediff. SQL imposes a penalty for converting between numeric types and strings and then back again. – Portman Sep 17 '08 at 19:32

1

Too bad it's not Oracle, or else you could use trunc() or to_char().

But I had similar issues with SQL Server and used the CONVERT() and DateDiff() methods, as referenced here

answered Sep 17 '08 at 17:27
typicalrunt
641 1 6 12

0

There are several ways to skin this cat =)

```
select convert(datetime,convert(varchar,CURRENT_TIMESTAMP,101))
```

edited Oct 2 '14 at 17:57
Leistungsabfall
4,682 7 22 37

answered Oct 2 '14 at 17:27
Sean
11

A prior answer suggested this and was commented on (slower) – Hogan Oct 6 '14 at 16:39

0

DateAdd along with DateDiff can help to do many different tasks. For example, you can find last day of any month as well can find last day of previous or next month.

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up OR SIGN IN WITH Google Facebook

```
----Last Day of Next Month
SELECT DATEADD(s,-1,DATEADD(mm, DATEDIFF(m,0,GETDATE())+2,0))
LastDay_NextMonth
```

Source

answered May 5 '16 at 20:19

pinaldave

361 3 7

-2

Since PostgreSQL is also a "SQL Server", I'll mention

```
date_trunc()
```

Which does exactly what you're asking gracefully.

For example:

```
 select date_trunc('hour',current_timestamp);
       date_trunc
------------------------
 2009-02-18 07:00:00-08
(1 row)
```

edited May 5 '16 at 20:21

wogsland

5,048 13 34 62

answered Feb 18 '09 at 15:46

Ian

"SQL Server" here refers to Microsoft's SQL DBMS. The name is quite confusing indeed. – Aurélien Gasser ♦ Feb 11 '17 at 1:34

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up OR SIGN IN WITH Google Facebook ✕

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up OR SIGN IN WITH Google Facebook