


[SEARCH](#)

[VIDEOS](#)

## PhpStorm

Boost your productivity with PhpStorm –  
the most powerful PHP IDE

[Try FREE now](#)
[SQL E/BOOKS](#)
[WORLD, USA & SQL NEWS](#)
[FORMAT](#)

[SQL Server Scripts](#) [SQL 2005](#) [SQL 2008](#) [Articles](#)

### How to convert from string to datetime?

Execute the following T-SQL scripts in Microsoft SQL Server Management Studio (SSMS) Query Editor to demonstrate T-SQL CONVERT and CAST functions in transforming string SQL date formats, string time & string datetime data to datetime data type. Practical examples for T-SQL DATE / DATETIME functions.

```
-- SQL Server string to date / datetime conversion - datetime string format sql server
-- MSSQL string to datetime conversion - convert char to date - convert varchar to date
-- Subtract 100 from style number (format) for yy instead yyyy (or ccyy with century)
SELECT convert(datetime, 'Oct 23 2012 11:01AM', 100) -- mon dd yyyy hh:mmAM (or PM)
SELECT convert(datetime, 'Oct 23 2012 11:01AM') -- 2012-10-23 11:01:00.000

-- Without century (yy) string date conversion - convert string to datetime function
SELECT convert(datetime, 'Oct 23 12 11:01AM', 0) -- mon dd yy hh:mmAM (or PM)
SELECT convert(datetime, 'Oct 23 12 11:01AM') -- 2012-10-23 11:01:00.000

-- Convert string to datetime sql - convert string to date sql - sql dates format
-- T-SQL convert string to datetime - SQL Server convert string to date
SELECT convert(datetime, '10/23/2016', 101) -- mm/dd/yyyy
SELECT convert(datetime, '2016.10.23', 102) -- yyyy.mm.dd ANSI date with century
SELECT convert(datetime, '23/10/2016', 103) -- dd/mm/yyyy
SELECT convert(datetime, '23.10.2016', 104) -- dd.mm.yyyy
SELECT convert(datetime, '23-10-2016', 105) -- dd-mm-yyyy
-- mon types are nondeterministic conversions, dependent on language setting
SELECT convert(datetime, '23 OCT 2016', 106) -- dd mon yyyy
SELECT convert(datetime, 'Oct 23, 2016', 107) -- mon dd, yyyy
-- 2016-10-23 00:00:00.000
SELECT convert(datetime, '20:10:44', 108) -- hh:mm:ss
-- 1900-01-01 20:10:44.000

-- mon dd yyyy hh:mm:ss:mmmAM (or PM) - sql time format - SQL Server datetime format
SELECT convert(datetime, 'Oct 23 2016 11:02:44:013AM', 109)
-- 2016-10-23 11:02:44.013
SELECT convert(datetime, '10-23-2016', 110) -- mm-dd-yyyy
SELECT convert(datetime, '2016/10/23', 111) -- yyyy/mm/dd
-- YYYMMDD ISO date format works at any language setting - international standard
SELECT convert(datetime, '20161023')
SELECT convert(datetime, '20161023', 112) -- ISO yyyymmdd
-- 2016-10-23 00:00:00.000
SELECT convert(datetime, '23 Oct 2016 11:02:07:577', 113) -- dd mon yyyy hh:mm:ss:mmm
-- 2016-10-23 11:02:07.577
SELECT convert(datetime, '20:10:25:300', 114) -- hh:mm:ss:mmm(24h)
-- 1900-01-01 20:10:25.300
SELECT convert(datetime, '2016-10-23 20:44:11', 120) -- yyyy-mm-dd hh:mm:ss(24h)
```

```

-- 2016-10-23 20:44:11.000
SELECT convert(datetime, '2016-10-23 20:44:11.500', 121) -- yyyy-mm-dd hh:mm:ss.mmm
-- 2016-10-23 20:44:11.500

-- Style 126 is ISO 8601 format: international standard - works with any language setting
SELECT convert(datetime, '2008-10-23T18:52:47.513', 126) -- yyyy-mm-ddThh:mm:ss(.mmm)
-- 2008-10-23 18:52:47.513
SELECT convert(datetime, N'23 6:52:47.513 1429 جـ ٢١ PM', 130) -- Islamic/Hijri date
SELECT convert(datetime, '23/10/1429 6:52:47:513PM', 131) -- Islamic/Hijri date

-- Convert DDMMyYYY format to datetime - sql server to date / datetime
SELECT convert(datetime, STUFF(STUFF('31012016',3,0,'-'),6,0,'-'), 105)
-- 2016-01-31 00:00:00.000
-- SQL Server T-SQL string to datetime conversion without century - some exceptions
-- nondeterministic means language setting dependent such as Mar/Mär/mars/márc
SELECT convert(datetime, 'Oct 23 16 11:02:44AM') -- Default
SELECT convert(datetime, '10/23/16', 1) -- mm/dd/yy U.S.
SELECT convert(datetime, '16.10.23', 2) -- yy.mm.dd ANSI
SELECT convert(datetime, '23/10/16', 3) -- dd/mm/yy UK/FR
SELECT convert(datetime, '23.10.16', 4) -- dd.mm.yy German
SELECT convert(datetime, '23-10-16', 5) -- dd-mm-yy Italian
SELECT convert(datetime, '23 OCT 16', 6) -- dd mon yy non-det.
SELECT convert(datetime, 'Oct 23, 16', 7) -- mon dd, yy non-det.
SELECT convert(datetime, '20:10:44', 8) -- hh:mm:ss
SELECT convert(datetime, 'Oct 23 16 11:02:44:013AM', 9) -- Default with msec
SELECT convert(datetime, '10-23-16', 10) -- mm-dd-yy U.S.
SELECT convert(datetime, '16/10/23', 11) -- yy/mm/dd Japan
SELECT convert(datetime, '161023', 12) -- yymmdd ISO
SELECT convert(datetime, '23 Oct 16 11:02:07:577', 13) -- dd mon yy hh:mm:ss:mmm EU dflt
SELECT convert(datetime, '20:10:25:300', 14) -- hh:mm:ss:mmm(24h)
SELECT convert(datetime, '2016-10-23 20:44:11', 20) -- yyyy-mm-dd hh:mm:ss(24h) ODBC can.
SELECT convert(datetime, '2016-10-23 20:44:11.500', 21) -- yyyy-mm-dd hh:mm:ss.mmm ODBC
-----
```

```

-- SQL Datetime Data Type: Combine date & time string into datetime - sql hh mm ss
-- String to datetime - mssql datetime - sql convert date - sql concatenate string
DECLARE @DateTimeValue varchar(32), @DateValue char(8), @TimeValue char(6)
```

```

SELECT @DateValue = '20120718',
@TimeValue = '211920'
SELECT @DateTimeValue =
convert(varchar, convert(datetime, @DateValue), 111)
+ ' ' + substring(@TimeValue, 1, 2)
+ ':' + substring(@TimeValue, 3, 2)
+ ':' + substring(@TimeValue, 5, 2)
SELECT
DateInput = @DateValue, TimeInput = @TimeValue,
DateTimeOutput = @DateTimeValue;
/*
DateInput    TimeInput    DateTimeOutput
20120718    211920    2012/07/18 21:19:20 */
```



```

/* DATETIME 8 bytes internal storage structure
o 1st 4 bytes: number of days after the base date 1900-01-01
o 2nd 4 bytes: number of clock-ticks (3.33 milliseconds) since midnight
```

```

DATETIME2 8 bytes (precision > 4) internal storage structure
o 1st byte: precision like 7
o middle 4 bytes: number of time units (100ns smallest) since midnight
o last 3 bytes: number of days after the base date 0001-01-01
```

```

DATE 3 bytes internal storage structure
o 3 bytes integer: number of days after the first date 0001-01-01
o Note: hex byte order reversed
```

```

SMALLDATETIME 4 bytes internal storage structure
o 1st 2 bytes: number of days after the base date 1900-01-01
o 2nd 2 bytes: number of minutes since midnight */
```

```

SELECT CONVERT(binary(8), getdate()) -- 0x00009E4D 00C01272
SELECT CONVERT(binary(4), convert(smalldatetime,getdate())) -- 0x9E4D 02BC

-- This is how a datetime looks in 8 bytes
nnnn nn nn nn nn nn nn nn
```

```

DECLARE @dtBex binary(8)=0x000000000000203344;
DECLARE @dt datetime = @dtBex
SELECT @dt -- 2007-07-09 02:44:34.147
-----*/
-----+
-- SQL Server 2012 New Date & Time Related Functions
-----+
SELECT DATEFROMPARTS ( 2016, 10, 23 ) AS RealDate; -- 2016-10-23

SELECT DATETIMEFROMPARTS ( 2016, 10, 23, 10, 10, 10, 500 ) AS RealDateTime; -- 2016-10-23 10:10:10.500

SELECT EOMONTH('20140201'); -- 2014-02-28
SELECT EOMONTH('20160201'); -- 2016-02-29
SELECT EOMONTH('20160201',1); -- 2016-03-31

SELECT FORMAT ( getdate(), 'yyyy/MM/dd hh:mm:ss tt', 'en-US' ); -- 2016/07/30 03:39:48 AM
SELECT FORMAT ( getdate(), 'd', 'en-US' ); -- 7/30/2016

SELECT PARSE('SAT, 13 December 2014' AS datetime USING 'en-US') AS [Date&Time];
-- 2014-12-13 00:00:00.000

SELECT TRY_PARSE('SAT, 13 December 2014' AS datetime USING 'en-US') AS [Date&Time];
-- 2014-12-13 00:00:00.000

SELECT TRY_CONVERT(datetime, '13 December 2014') AS [Date&Time]; -- 2014-12-13 00:00:00.000
SELECT CONVERT(datetime2, sysdatetime()); AS [DateTime2]; -- 2016-02-12 13:09:24.0642891
-----+
-- SQL convert seconds to HH:MM:SS - sql times format - sql hh mm ss
DECLARE @Seconds INT
SET @Seconds = 20000
SELECT HH = @Seconds / 3600, MM = (@Seconds%3600) / 60, SS = (@Seconds%60)
/* HH   MM   SS
 5    33   20 */
-----+
-- SQL Server Date Only from DATETIME column - get date only
-- T-SQL just date - truncate time from datetime - remove time part
-----+
DECLARE @Now datetime = CURRENT_TIMESTAMP -- getdate()
SELECT DateAndTime = @Now -- Date portion and Time portion
,DateString = REPLACE(LEFT(CONVERT(varchar, @Now, 112),10), '-', '-')
,[Date] = CONVERT(DATE, @Now) -- SQL Server 2008 and on - date part
,Midnight1 = dateadd(day, datediff(day,0, @Now), 0)
,Midnight2 = CONVERT(DATETIME,CONVERT(int, @Now))
,Midnight3 = CONVERT(DATETIME,CONVERT(BIGINT,@Now) &
(POWER(Convert(bigint,2),32)-1))
/* DateAndTime  DateString Date Midnight1  Midnight2  Midnight3
2010-11-02 08:00:33.657 20101102    2010-11-02 2010-11-02 00:00:00.000 2010-11-02
00:00:00.000      2010-11-02 00:00:00.000 */
-----+
-- SQL Server 2008 convert datetime to date - sql yyyy mm dd
SELECT TOP (3) OrderDate = CONVERT(date, OrderDate),
Today = CONVERT(date, getdate())
FROM AdventureWorks2008.Sales.SalesOrderHeader
ORDER BY newid();
/*
          OrderDate   Today
2004-02-15 2012-06-18 ....*/
-----+
-- SQL date yyyy mm dd - sqlserver yyyy mm dd - date format yyyyymmdd
SELECT CONVERT(VARCHAR(10), GETDATE(), 111) AS [YYYY/MM/DD]
/* YYYY/MM/DD
2015/07/11 */
SELECT CONVERT(VARCHAR(10), GETDATE(), 112) AS [YYYYMMDD]
/* YYYYMMDD
20150711 */
SELECT REPLACE(CONVERT(VARCHAR(10), GETDATE(), 111), '/', '-') AS [YYYY MM DD]
/* YYYY MM DD
2015 07 11 */
-- Converting to special (non-standard) date formats: DD-MMM-YY
SELECT UPPER(REPLACE(CONVERT(VARCHAR,GETDATE(),6), ' ', '-'))
-- 07-MAR-14
-----+

```

```

-- SQL convert date string to datetime - time set to 00:00:00.000 or 12:00AM
PRINT CONVERT(datetime,'07-10-2012',110)      -- Jul 10 2012 12:00AM
PRINT CONVERT(datetime,'2012/07/10',111)       -- Jul 10 2012 12:00AM
PRINT CONVERT(datetime,'20120710', 112)        -- Jul 10 2012 12:00AM
-----
-- UNIX to SQL Server datetime conversion
declare @UNIX bigint = 1477216861;
select dateadd(ss,@UNIX,'19700101'); -- 2016-10-23 10:01:01.000
-----
-- String to date conversion - sql date yyyy mm dd - sql date formatting
-- SQL Server cast string to date - sql convert date to datetime
SELECT [Date] = CAST (@DateValue AS datetime)
-- 2012-07-18 00:00:00.000

-- SQL convert string date to different style - sql date string formatting
SELECT CONVERT(varchar, CONVERT(datetime, '20140508'), 100)
-- May 8 2014 12:00AM

-- SQL Server convert date to integer
DECLARE @Date datetime; SET @Date = getdate();
SELECT DateAsInteger = CAST (CONVERT(varchar,@Date,112) as INT);
-- Result: 20161225

-- SQL Server convert integer to datetime
DECLARE @iDate int
SET @iDate = 20151225
SELECT IntegerToDatetime = CAST(convert(varchar,@iDate) as datetime)
-- 2015-12-25 00:00:00.000

-- Alternates: date-only datetime values
-- SQL Server floor date - sql convert datetime
SELECT [DATE-ONLY]=CONVERT(DATETIME, FLOOR(CONVERT(FLOAT, GETDATE())))
SELECT [DATE-ONLY]=CONVERT(DATETIME, FLOOR(CONVERT(MONEY, GETDATE())))
-- SQL Server cast string to datetime
-- SQL Server datetime to string convert
SELECT [DATE-ONLY]=CAST(CONVERT(varchar, GETDATE(), 101) AS DATETIME)
-- SQL Server dateadd function - T-SQL datediff function
-- SQL strip time from date - MSSQL strip time from datetime
SELECT getdate() ,dateadd(dd, datediff(dd, 0, getdate()), 0)
-- Results: 2016-01-23 05:35:52.793 2016-01-23 00:00:00.000

-- String date - 10 bytes of storage
SELECT [STRING DATE]=CONVERT(varchar, GETDATE(), 110)
SELECT [STRING DATE]=CONVERT(varchar, CURRENT_TIMESTAMP, 110)
-- Same results: 01-02-2012

-- SQL Server cast datetime as string - sql datetime formatting
SELECT stringDateTime=CAST (getdate() as varchar) -- Dec 29 2012 3:47AM

```

### The BEST 70-461 SQL Server 2012 Querying Exam Prep Book!

```

-----
-- SQL date range BETWEEN operator
-----
-- SQL date range select - date range search - T-SQL date range query
-- Count Sales Orders for 2003 OCT-NOV
DECLARE @StartDate DATETIME, @EndDate DATETIME
SET @StartDate = convert(DATETIME,'10/01/2003',101)
SET @EndDate = convert(DATETIME,'11/30/2003',101)

SELECT @StartDate, @EndDate
-- 2003-10-01 00:00:00.000 2003-11-30 00:00:00.000
SELECT dateadd(DAY,1,@EndDate),
       dateadd(ms,-3,dateadd(DAY,1,@EndDate))
-- 2003-12-01 00:00:00.000 2003-11-30 23:59:59.997

-- MSSQL date range select using >= and <
SELECT [Sales Orders for 2003 OCT-NOV] = COUNT(* )
FROM   Sales.SalesOrderHeader
WHERE OrderDate >= @StartDate AND OrderDate < dateadd(DAY,1,@EndDate)

```

```

/* Sales Orders for 2003 OCT-NOV
3668 */

-- Equivalent date range query using BETWEEN comparison
-- It requires a bit of trick programming
SELECT [Sales Orders for 2003 OCT-NOV] = COUNT(*)
FROM   Sales.SalesOrderHeader
WHERE OrderDate BETWEEN @StartDate AND dateadd(ms,-3,dateadd(DAY,1,@EndDate))
-- 3668

USE AdventureWorks;
-- SQL between string dates
SELECT P0s=COUNT(*) FROM Purchasing.PurchaseOrderHeader
WHERE OrderDate BETWEEN '20040201' AND '20040210' -- Result: 108

-- SQL BETWEEN dates without time - time stripped - time removed - date part only
SELECT P0s=COUNT(*) FROM Purchasing.PurchaseOrderHeader
WHERE datediff(dd,0,OrderDate)
    BETWEEN datediff(dd,0,'20040201 12:11:39') AND datediff(dd,0,'20040210 14:33:19')
-- 108

-- BETWEEN is equivalent to >=...AND....<
SELECT P0s=COUNT(*) FROM Purchasing.PurchaseOrderHeader
WHERE OrderDate
BETWEEN '2004-02-01 00:00:00.000' AND '2004-02-10 00:00:00.000'
/* Orders with OrderDates
'2004-02-10 00:00:01.000' - 1 second after midnight (12:00AM)
'2004-02-10 00:01:00.000' - 1 minute after midnight
'2004-02-10 01:00:00.000' - 1 hour after midnight
are not included in the two queries above.*/
-- To include the entire day of 2004-02-10 use:
SELECT P0s=COUNT(*) FROM Purchasing.PurchaseOrderHeader
WHERE OrderDate >= '20040201' AND OrderDate < '20040211'

-----
-- Calculate week ranges in a year
-----
DECLARE @Year INT = '2016';
WITH cteDays AS (SELECT DayOfYear=Dateadd(dd, number,
                                             CONVERT(DATE, CONVERT(char(4),@Year)+'0101'))
                  FROM master.dbo.spt_values WHERE type='P'),
CTE AS (SELECT DayOfYear, WeekOfYear=DATEPART(week,DayOfYear)
        FROM cteDays WHERE YEAR(DayOfYear)= @YEAR)
SELECT WeekOfYear, StartOfWeek=MIN(DayOfYear), EndOfWeek=MAX(DayOfYear)
FROM CTE GROUP BY WeekOfYear ORDER BY WeekOfYear

-----
-- Date validation function ISDATE - returns 1 or 0 - SQL datetime functions
-----
DECLARE @StringDate varchar(32)
SET @StringDate = '2011-03-15 18:50'
IF EXISTS( SELECT * WHERE ISDATE(@StringDate) = 1)
    PRINT 'VALID DATE: ' + @StringDate
ELSE
    PRINT 'INVALID DATE: ' + @StringDate
GO
-- Result: VALID DATE: 2011-03-15 18:50

DECLARE @StringDate varchar(32)
SET @StringDate = '20112-03-15 18:50'
IF EXISTS( SELECT * WHERE ISDATE(@StringDate) = 1)
    PRINT 'VALID DATE: ' + @StringDate
ELSE
    PRINT 'INVALID DATE: ' + @StringDate
-- Result: INVALID DATE: 20112-03-15 18:50

-- First and last day of date periods - SQL Server 2008 and on code
DECLARE @Date DATE = '20161023'
SELECT ReferenceDate = @Date
SELECT FirstDayOfYear = CONVERT(DATE, dateadd(yy, datediff(yy,0, @Date),0))
SELECT LastDayOfYear = CONVERT(DATE, dateadd(yy, datediff(yy,0, @Date)+1,-1))
SELECT FDofSemester = CONVERT(DATE, dateadd(qq, ((datediff(qq,0,@Date)/2)*2),0))
SELECT LastDayOfSemester
= CONVERT(DATE, dateadd(qq, ((datediff(qq,0,@Date)/2)*2)+2,-1))

```

```

SELECT FirstDayOfQuarter = CONVERT(DATE, dateadd(qq, datediff(qq,0, @Date),0))
-- 2016-10-01
SELECT LastDayOfQuarter = CONVERT(DATE, dateadd(qq, datediff(qq,0,@Date)+1,-1))
-- 2016-12-31
SELECT FirstDayOfMonth = CONVERT(DATE, dateadd(mm, datediff(mm,0, @Date),0))
SELECT LastDayOfMonth = CONVERT(DATE, dateadd(mm, datediff(mm,0, @Date)+1,-1))
SELECT FirstDayOfWeek = CONVERT(DATE, dateadd(wk, datediff(wk,0, @Date),0))
SELECT LastDayOfWeek = CONVERT(DATE, dateadd(wk, datediff(wk,0, @Date)+1,-1))
-- 2016-10-30

-- Month sequence generator - sequential numbers / dates
DECLARE @Date date = '2000-01-01'
SELECT MonthStart=dateadd(MM, number, @Date)
FROM master.dbo.spt_values
WHERE type='P' AND dateadd(MM, number, @Date) <= CURRENT_TIMESTAMP
ORDER BY MonthStart
/* MonthStart
2000-01-01
2000-02-01
2000-03-01 ....*/

```

### The BEST 70-461 SQL Server 2012 Querying Exam Prep Book!

```

-----
-- Selected named date styles
-----
DECLARE @DateTimeValue varchar(32)
-- US-Style
SELECT @DateTimeValue = '10/23/2016'
SELECT StringDate=@DateTimeValue,
[US-Style] = CONVERT(datetime, @DatetimeValue)

SELECT @DateTimeValue = '10/23/2016 23:01:05'
SELECT StringDate = @DateTimeValue,
[US-Style] = CONVERT(datetime, @DatetimeValue)

-- UK-Style, British/French- convert string to datetime sql
-- sql convert string to datetime
SELECT @DateTimeValue = '23/10/16 23:01:05'
SELECT StringDate = @DateTimeValue,
[UK-Style] = CONVERT(datetime, @DatetimeValue, 3)

SELECT @DateTimeValue = '23/10/2016 04:01 PM'
SELECT StringDate = @DateTimeValue,
[UK-Style] = CONVERT(datetime, @DatetimeValue, 103)

-- German-Style
SELECT @DateTimeValue = '23.10.16 23:01:05'
SELECT StringDate = @DateTimeValue,
[German-Style] = CONVERT(datetime, @DatetimeValue, 4)

SELECT @DateTimeValue = '23.10.2016 04:01 PM'
SELECT StringDate = @DateTimeValue,
[German-Style] = CONVERT(datetime, @DatetimeValue, 104)
-----

-- Double conversion to US-Style 107 with century: Oct 23, 2016
SET @DateTimeValue='10/23/16'
SELECT StringDate=@DateTimeValue,
[US-Style] = CONVERT(varchar, CONVERT(datetime, @DateTimeValue),107)

-- Using DATEFORMAT - UK-Style - SQL dateformat
SET @DateTimeValue='23/10/16'
SET DATEFORMAT dmy
SELECT StringDate=@DateTimeValue,
[Date Time] = CONVERT(datetime, @DatetimeValue)
-- Using DATEFORMAT - US-Style
SET DATEFORMAT mdy
-- Finding out date format for a session
SELECT session_id, date_format from sys.dm_exec_sessions
-----
```

```

-- Convert date string from DD/MM/YYYY UK format to MM/DD/YYYY US format
DECLARE @UKdate char(10) = '15/03/2016'
SELECT CONVERT(CHAR(10), CONVERT(datetime, @UKdate, 103), 101)
-- 03/15/2016

-- DATEPART datetime function example - SQL Server datetime functions
SELECT * FROM Northwind.dbo.Orders
WHERE DATEPART(YEAR, OrderDate) = '1996' AND
      DATEPART(MONTH, OrderDate) = '07' AND
      DATEPART(DAY, OrderDate) = '10'

-- Alternate syntax for DATEPART example
SELECT * FROM Northwind.dbo.Orders
WHERE YEAR(OrderDate) = '1996' AND
      MONTH(OrderDate) = '07' AND
      DAY(OrderDate) = '10'
-----
-- T-SQL calculate the number of business days function / UDF - exclude SAT & SUN

CREATE FUNCTION fnBusinessDays (@StartDate DATETIME, @EndDate DATETIME)
RETURNS INT AS
BEGIN
    IF (@StartDate IS NULL OR @EndDate IS NULL) RETURN (0)
    DECLARE @i INT = 0;
    WHILE (@StartDate <= @EndDate)
    BEGIN
        SET @i = @i + CASE
            WHEN datepart(dw,@StartDate) BETWEEN 2 AND 6 THEN 1
            ELSE 0
        END
        SET @StartDate = @StartDate + 1
    END -- while
    RETURN (@i)
END -- function
GO
SELECT dbo.fnBusinessDays('2016-01-01', '2016-12-31')
-- 261
-----
-- T-SQL DATENAME function usage for weekdays
SELECT DayName=DATENAME(weekday, OrderDate), SalesPerWeekDay = COUNT(*)
FROM AdventureWorks2008.Sales.SalesOrderHeader
GROUP BY DATENAME(weekday, OrderDate), DATEPART(weekday, OrderDate)
ORDER BY DATEPART(weekday, OrderDate)
/* DayName SalesPerWeekDay
Sunday     4482
Monday     4591
Tuesday    4346.... */

-- DATENAME application for months
SELECT MonthName=DATENAME(month, OrderDate), SalesPerMonth = COUNT(*)
FROM AdventureWorks2008.Sales.SalesOrderHeader
GROUP BY DATENAME(month, OrderDate), MONTH(OrderDate) ORDER BY MONTH(OrderDate)
/* MonthName   SalesPerMonth
January     2483
February    2686
March       2750
April       2740.... */

-- Getting month name from month number
SELECT DATENAME(MM,dateadd(MM,7,-1)) -- July

ARTICLE - Essential SQL Server Date, Time and DateTime Functions
ARTICLE - Demystifying the SQL Server DATETIME Datatype
-----
-- Extract string date from text with PATINDEX pattern matching
-- Apply sql server string to date conversion
-----
USE tempdb;
go
CREATE TABLE InsiderTransaction (
    InsiderTransactionID int identity primary key,
    ...
)

```

```

TradeDate datetime,
TradeMsg varchar(256),
ModifiedDate datetime default (getdate())
-- Populate table with dummy data
INSERT InsiderTransaction (TradeMsg) VALUES(
'INSIDER TRAN QABC Hammer, Bruce D. CSO 09-02-08 Buy 2,000 6.10')
INSERT InsiderTransaction (TradeMsg) VALUES(
'INSIDER TRAN QABC Schmidt, Steven CFO 08-25-08 Buy 2,500 6.70')
INSERT InsiderTransaction (TradeMsg) VALUES(
'INSIDER TRAN QABC Hammer, Bruce D. CSO 08-20-08 Buy 3,000 8.59')
INSERT InsiderTransaction (TradeMsg) VALUES(
'INSIDER TRAN QABC Walters, Jeff CTO 08-15-08 Sell 5,648 8.49')
INSERT InsiderTransaction (TradeMsg) VALUES(
'INSIDER TRAN QABC Walters, Jeff CTO 08-15-08 Option Execute 5,648 2.15')
INSERT InsiderTransaction (TradeMsg) VALUES(
'INSIDER TRAN QABC Hammer, Bruce D. CSO 07-31-08 Buy 5,000 8.05')
INSERT InsiderTransaction (TradeMsg) VALUES(
'INSIDER TRAN QABC Lennot, Mark B. Director 08-31-07 Buy 1,500 9.97')
INSERT InsiderTransaction (TradeMsg) VALUES(
'INSIDER TRAN QABC O''Neal, Linda COO 08-01-08 Sell 5,000 6.50')

-- Extract dates from stock trade message text
-- Pattern match for MM-DD-YY using the PATINDEX string function
SELECT TradeDate=substring(TradeMsg,
    patindex('%[01][0-9]-[0123][0-9]-[0-9][0-9]%', TradeMsg), 8)
FROM InsiderTransaction
WHERE patindex('%[01][0-9]-[0123][0-9]-[0-9][0-9]%', TradeMsg) > 0
/* Partial results
TradeDate
09-02-08
08-25-08
08-20-08 */

-- Update table with extracted date
-- Convert string date to datetime
UPDATE InsiderTransaction
SET TradeDate = convert(datetime, substring(TradeMsg,
    patindex('%[01][0-9]-[0123][0-9]-[0-9][0-9]%', TradeMsg), 8))
WHERE patindex('%[01][0-9]-[0123][0-9]-[0-9][0-9]%', TradeMsg) > 0

SELECT * FROM InsiderTransaction ORDER BY TradeDate desc
/* Partial results
InsiderTransactionID TradeDate TradeMsg ModifiedDate
1 2008-09-02 00:00:00.000 INSIDER TRAN QABC Hammer, Bruce D. CSO 09-02-08 Buy
2,000 6.10 2008-12-22 20:25:19.263
2 2008-08-25 00:00:00.000 INSIDER TRAN QABC Schmidt, Steven CFO 08-25-08 Buy
2,500 6.70 2008-12-22 20:25:19.263 */
-- Cleanup task
DROP TABLE InsiderTransaction

*****
VALID DATE RANGES FOR DATE / DATETIME DATA TYPES

DATE (3 bytes) date range:
January 1, 1 A.D. through December 31, 9999 A.D.

SMALLDATETIME (4 bytes) date range:
January 1, 1900 through June 6, 2079

DATETIME (8 bytes) date range:
January 1, 1753 through December 31, 9999

DATETIME2 (6-8 bytes) date range:
January 1, 1 A.D. through December 31, 9999 A.D.

-- The statement below will give a date range error
SELECT CONVERT(smalldatetime, '2110-01-01')
/*Msg 242, Level 16, State 3, Line 1
The conversion of a varchar data type to a smalldatetime data type
. . . . .

```

```
resulted in an out-of-range value. */  
******/
```

## The BEST 70-461 SQL Server 2012 Querying Exam Prep Book!

```
--  
-- SQL CONVERT DATE/DATETIME script applying table variable  
--  
-- SQL Server convert date  
-- Datetime column is converted into date only string column  
DECLARE @sqlConvertDate TABLE ( DatetimeColumn datetime,  
                                 DateColumn char(10));  
INSERT @sqlConvertDate (DatetimeColumn) SELECT GETDATE()  
  
UPDATE @sqlConvertDate  
SET DateColumn = CONVERT(char(10), DatetimeColumn, 111)  
SELECT * FROM @sqlConvertDate  
  
-- SQL Server convert datetime - String date column converted into datetime column  
UPDATE @sqlConvertDate  
SET DatetimeColumn = CONVERT(Datetime, DateColumn, 111)  
SELECT * FROM @sqlConvertDate  
  
-- Equivalent formulation - SQL Server cast datetime  
UPDATE @sqlConvertDate  
SET DatetimeColumn = CAST(DateColumn AS datetime)  
SELECT * FROM @sqlConvertDate  
/* First results  
DatetimeColumn          DateColumn  
2012-12-25 15:54:10.363    2012/12/25 */  
/* Second results:  
DatetimeColumn          DateColumn  
2012-12-25 00:00:00.000    2012/12/25 */  
-----  
  
-- SQL date sequence generation with dateadd & table variable  
-- SQL Server cast datetime to string - SQL Server insert default values method  
DECLARE @Sequence table (Sequence int identity(1,1))  
DECLARE @i int; SET @i = 0  
WHILE ( @i < 500)  
BEGIN  
    INSERT @Sequence DEFAULT VALUES  
    SET @i = @i + 1  
END  
SELECT DateSequence = CAST(dateadd(day, Sequence, getdate()) AS varchar)  
FROM @Sequence  
/* Partial results:  
DateSequence  
Dec 31 2008 3:02AM  
Jan 1 2009 3:02AM  
Jan 2 2009 3:02AM  
Jan 3 2009 3:02AM  
Jan 4 2009 3:02AM */  
  
-- SETTING FIRST DAY OF WEEK TO SUNDAY  
SET DATEFIRST 7;  
SELECT @@DATEFIRST  
-- 7  
SELECT CAST('2016-10-23' AS date) AS SelectDate  
      ,DATEPART(dw, '2016-10-23') AS DayOfWeek;  
-- 2016-10-23 1  
  
-----  
-- SQL Last Week calculations  
-----  
-- SQL last Friday - Implied string to datetime conversions in dateadd & datediff  
DECLARE @BaseFriday CHAR(8), @LastFriday datetime, @LastMonday datetime  
SET @BaseFriday = '19000105'  
SELECT @LastFriday = dateadd(dd,  
                           (datediff(dd, @BaseFriday, CURRENT_TIMESTAMP) / 7) * 7, @BaseFriday)  
SELECT [Last Friday] = @LastFriday
```

```

-- Result: 2008-12-26 00:00:00.000
-- SQL last Monday (last week's Monday)
SELECT @LastMonday=dateadd(dd,
                           (datediff(dd, @BaseFriday, CURRENT_TIMESTAMP) / 7) * 7 - 4, @BaseFriday)
SELECT [Last Monday]=@LastMonday
-- Result: 2008-12-22 00:00:00.000

-- SQL last week - SUN - SAT
SELECT [Last Week] = CONVERT(varchar,dateadd(day, -1, @LastMonday), 101) + ' - ' +
                      CONVERT(varchar,dateadd(day, 1, @LastFriday), 101)
-- Result: 12/21/2008 - 12/27/2008

-----
-- Specific day calculations
-----
-- First day of current month
SELECT dateadd(month, datediff(month, 0, getdate()), 0)
-- 15th day of current month
SELECT dateadd(day,14,dateadd(month,datediff(month,0,getdate()),0))
-- First Monday of current month
SELECT dateadd(day, (9-datepart(weekday,
                                dateadd(month, datediff(month, 0, getdate()), 0)))%7,
              dateadd(month, datediff(month, 0, getdate()), 0))
-- Next Monday calculation from the reference date which was a Monday
DECLARE @Now datetime = GETDATE();
DECLARE @NextMonday datetime = dateadd(dd, ((datediff(dd, '19000101', @Now)
                                             / 7) * 7) + 7, '19000101');
SELECT [Now]=@Now, [Next Monday]=@NextMonday
-- Last Friday of current month
SELECT dateadd(day, -7+(6-datepart(weekday,
                                    dateadd(month, datediff(month, 0, getdate())+1, 0)))%7,
              dateadd(month, datediff(month, 0, getdate())+1, 0))
-- First day of next month
SELECT dateadd(month, datediff(month, 0, getdate())+1, 0)
-- 15th of next month
SELECT dateadd(day,14, dateadd(month, datediff(month, 0, getdate())+1, 0))
-- First Monday of next month
SELECT dateadd(day, (9-datepart(weekday,
                                dateadd(month, datediff(month, 0, getdate())+1, 0)))%7,
              dateadd(month, datediff(month, 0, getdate())+1, 0))

-----
-- SQL Last Date calculations
-----
-- Last day of prior month - Last day of previous month
SELECT convert( varchar, dateadd(dd,-1,dateadd(mm, datediff(mm,0,getdate()), 0)),101)
-- 01/31/2019
-- Last day of current month
SELECT convert( varchar, dateadd(dd,-1,dateadd(mm, datediff(mm,0,getdate())+1, 0)),101)
-- 02/28/2019
-- Last day of prior quarter - Last day of previous quarter
SELECT convert( varchar, dateadd(dd,-1,dateadd(qq, datediff(qq,0,getdate()), 0)),101)
-- 12/31/2018
-- Last day of current quarter - Last day of current quarter
SELECT convert( varchar, dateadd(dd,-1,dateadd(qq, datediff(qq,0,getdate())+1, 0)),101)
-- 03/31/2019
-- Last day of prior year - Last day of previous year
SELECT convert( varchar, dateadd(dd,-1,dateadd(yy, datediff(yy,0,getdate()), 0)),101)
-- 12/31/2018
-- Last day of current year
SELECT convert( varchar, dateadd(dd,-1,dateadd(yy, datediff(yy,0,getdate())+1, 0)),101)
-- 12/31/2019
-----
-- SQL Server dateformat and language setting
-----
-- T-SQL set language= String to date conversion
SET LANGUAGE us_english
SELECT CAST('2018-03-15' AS datetime)
-- 2018-03-15 00:00:00.000

SET LANGUAGE british
SELECT CAST('2018-03-15' AS datetime)

```

```

/* Msg 242, Level 16, State 3, Line 2
The conversion of a varchar data type resulted in
an out-of-range value.
*/
SELECT CAST('2018-15-03' AS datetime)
-- 2018-03-15 00:00:00.000

SET LANGUAGE us_english

-- SQL dateformat with language dependency
SELECT name, alias, dateformat
FROM sys.syslanguages
WHERE langid in (0,1,2,4,5,6,7,10,11,13,23,31)
GO
/*
name      alias      dateformat
us_english English    mdy
Deutsch    German     dmy
Français   French     dmy
Dansk      Danish     dmy
Español    Spanish    dmy
Italiano   Italian    dmy
Nederlands Dutch      dmy
Suomi      Finnish    dmy
Svenska    Swedish    ymd
magyar    Hungarian  ymd
British    British English dmy
Arabic     Arabic     dmy */
-----

-- Generate list of months
:WITH CTE AS (
    SELECT      1 MonthNo,  CONVERT(DATE, '19000101') MonthFirst
    UNION ALL
    SELECT      MonthNo+1,  DATEADD(Month, 1, MonthFirst)
    FROM CTE WHERE Month(MonthFirst) < 12 )
SELECT      MonthNo AS MonthNumber, DATENAME(MONTH, MonthFirst) AS MonthName
FROM CTE ORDER BY MonthNo
/* MonthNumber MonthName
   1          January
   2          February
   3          March ... */

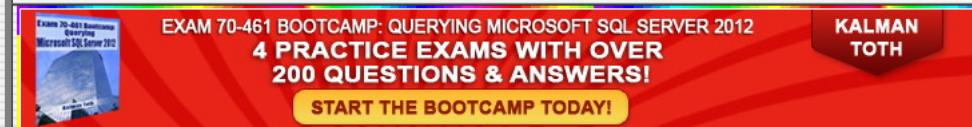
```

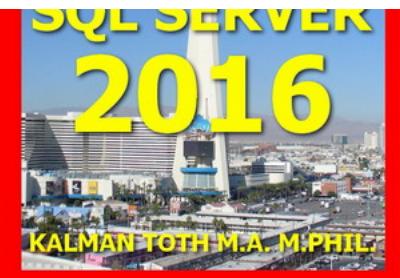
**Related articles:**

[The ultimate guide to the datetime datatypes](#)

[CAST and CONVERT \(Transact-SQL\)](#)

[CAST and CONVERT](#)





DATETIME | SELECT | INTO | DATE | PAD | DYNAL | CURSOR | MONEY | PERCENT | SP | JOB | TUNE | WHILE | OVER | UPDATE