**بسمه تعالی**



برنامه نویسی بانک اطلاعاتی SQL Server

**نویسنده : محمد حسین فخرآوری**

**Date and Time Functions**

تمامی مقادیر بازگشتی توسط این توابع از تنظیمات سیستم عاملی که instance اس کیو ال سرور بروی آن نصب است.

**معایب استفاده از کارکتر**

* Sort نامرتب
* مشکل در اضافه و کم کردن به تاریخ و زمان
* کندی در جستجو
* ذخیره به میزان طول رکورد در هارد که باعث کاهش چگالی داده های ذخیره شده روی یک Page شده و باعث افزایش تعداد Page ها همچنین افزایشI/O و درنهایت کاهش سرعت در تعداد رکوردهای بالا

**مزایای استفاده از فیلد Date Time**

* Sortمرتب
* ذخیره سازی به صورت عددی که باعث افزایش سرعت دستورات شما خواهد شد.
* به راحتی میتوان فرمت تاریخ را با توجه به فرمت زمان سیستم تغییر داد بدون کدنویسی
* امکان اضافه و کم کردن به تاریخ و زمان
* امکان استفاده از توابع SQL Server برای بدست آوردن بخشی از تاریخ.
* اگر ازنوع DateTime استفاده کنید 8 بایت فضا اشغال کرده و اگر ازنوع SmallDateTime استفاده کنید 4 بایت فضا اشغال میکند.

|  |  |
| --- | --- |
| **تابع** | **توضیحات** |
| [**NOW()**](http://www.w3schools.com/sql/func_now.asp) | تاریخ و زمان جاری را برمی گرداند |
| [**CURDATE()**](http://www.w3schools.com/sql/func_curdate.asp) | تاریخ جاری را برمی گرداند |
| [**CURTIME()**](http://www.w3schools.com/sql/func_curtime.asp) | زمان جاری را برمی گرداند |
| [**DATE()**](http://www.w3schools.com/sql/func_date.asp) | بیرون می کشد date/time بخش تاریخ را از یک عبارت |
| [**EXTRACT()**](http://www.w3schools.com/sql/func_extract.asp) | مانند سال، ماه، روز، ساعت و ... را برمی گرداند date/timeیکی از بخش های عبارت |
| [**DATE\_ADD()**](http://www.w3schools.com/sql/func_date_add.asp) | یک فاصله زمانی مشخص را به تاریخ  اضافه می کند |
| [**DATE\_SUB()**](http://www.w3schools.com/sql/func_date_sub.asp) | یک فاصله زمانی مشخص را از تاریخ کم می کند |
| [**DATEDIFF()**](http://www.w3schools.com/sql/func_datediff_mysql.asp) | تعداد روز بین دو تاریخ را برمی گرداند |
| [**DATE\_FORMAT()**](http://www.w3schools.com/sql/func_date_format.asp) | نمایش تاریخ و زمان در فرمت های مختلف |

**جدول زیر، لیستی از مهم ترین توابع تاریخ در SQL Server را نشان می دهد:**

|  |  |
| --- | --- |
| **تابع** | **توضیحات** |
| GETDATE() | تاریخ و زمان جاری را برمی گرداند |
| DATEPART() | مانند سال،ماه،روز،ساعت را برمی گرداندdate/timeیکی از بخش های عبارت |
| DATEADD() | یک فاصله زمانی مشخص را به تاریخ اضافه یا کم می کند |
| DATEDIFF() | زمان بین دو تاریخ را برمی گرداند |
| CONVERT() | نمایش تاریخ و زمان در فرمت های مختلف |

**تابع FORMAT() چگونگی نمایش یک فیلد را مشخص می کند**

|  |
| --- |
| FORMAT(value, nvarchar format [, culture ] ) RETURNS nvarchar |

* DATE - format YYYY-MM-DD
* DATETIME - format: YYYY-MM-DD HH:MM:SS
* SMALLDATETIME - format: YYYY-MM-DD HH:MM:SS
* TIMESTAMP - format: a unique number

|  |  |
| --- | --- |
| **Query** | **Sample output** |
| SELECT FORMAT (getdate(), 'dd/MM/yyyy ') as date | 07/03/2018 |
| SELECT FORMAT (getdate(), 'dd/MM/yyyy, hh:mm:ss ') as date | 07/03/2018, 11:36:14 |
| SELECT FORMAT (getdate(), 'dddd, MMMM, yyyy') as date | Wednesday, March, 2018 |
| SELECT FORMAT (getdate(), 'MMM dd yyyy') as date | Mar 07 2018 |
| SELECT FORMAT (getdate(), 'MM.dd.yy') as date | 03.07.18 |
| SELECT FORMAT (getdate(), 'MM-dd-yy') as date | 03-07-18 |
| SELECT FORMAT (getdate(), 'hh:mm:ss tt') as date | 11:36:14 AM |
| SELECT FORMAT (getdate(), 'd','us') as date | 03/07/2018 |

* توجه : زمانی که یک جدول جدید تعریف می کنید باید برای هر ستون Data Type آن را مشخص کنید.

|  |  |  |
| --- | --- | --- |
| **DATE ONLY FORMATS** | | |
| **Format #** | **Query** | **Sample** |
| 1 | select convert(varchar, getdate(), 1) | 12/30/06 |
| 2 | select convert(varchar, getdate(), 2) | 06.12.30 |
| 3 | select convert(varchar, getdate(), 3) | 30/12/06 |
| 4 | select convert(varchar, getdate(), 4) | 30.12.06 |
| 5 | select convert(varchar, getdate(), 5) | 30-12-06 |
| 6 | select convert(varchar, getdate(), 6) | 30 Dec 06 |
| 7 | select convert(varchar, getdate(), 7) | Dec 30, 06 |
| 10 | select convert(varchar, getdate(), 10) | 12-30-06 |
| 11 | select convert(varchar, getdate(), 11) | 06/12/30 |
| 12 | select convert(varchar, getdate(), 12) | 061230 |
| 23 | select convert(varchar, getdate(), 23) | 2006-12-30 |
| 101 | select convert(varchar, getdate(), 101) | 12/30/2006 |
| 102 | select convert(varchar, getdate(), 102) | 2006.12.30 |
| 103 | select convert(varchar, getdate(), 103) | 30/12/2006 |
| 104 | select convert(varchar, getdate(), 104) | 30.12.2006 |
| 105 | select convert(varchar, getdate(), 105) | 30-12-2006 |
| 106 | select convert(varchar, getdate(), 106) | 30 Dec 2006 |
| 107 | select convert(varchar, getdate(), 107) | Dec 30, 2006 |
| 110 | select convert(varchar, getdate(), 110) | 12-30-2006 |
| 111 | select convert(varchar, getdate(), 111) | 2006/12/30 |
| 112 | select convert(varchar, getdate(), 112) | 20061230 |
|  | | |
| **TIME ONLY FORMATS** | | |
| 8 | select convert(varchar, getdate(), 8) | 00:38:54 |
| 14 | select convert(varchar, getdate(), 14) | 00:38:54:840 |
| 24 | select convert(varchar, getdate(), 24) | 00:38:54 |
| 108 | select convert(varchar, getdate(), 108) | 00:38:54 |
| 114 | select convert(varchar, getdate(), 114) | 00:38:54:840 |
|  | | |
| **DATE & TIME FORMATS** | | |
| 0 | select convert(varchar, getdate(), 0) | Dec 12 2006 12:38AM |
| 9 | select convert(varchar, getdate(), 9) | Dec 30 2006 12:38:54:840AM |
| 13 | select convert(varchar, getdate(), 13) | 30 Dec 2006 00:38:54:840AM |
| 20 | select convert(varchar, getdate(), 20) | 2006-12-30 00:38:54 |
| 21 | select convert(varchar, getdate(), 21) | 2006-12-30 00:38:54.840 |
| 22 | select convert(varchar, getdate(), 22) | 12/30/06 12:38:54 AM |
| 25 | select convert(varchar, getdate(), 25) | 2006-12-30 00:38:54.840 |
| 100 | select convert(varchar, getdate(), 100) | Dec 30 2006 12:38AM |
| 109 | select convert(varchar, getdate(), 109) | Dec 30 2006 12:38:54:840AM |
| 113 | select convert(varchar, getdate(), 113) | 30 Dec 2006 00:38:54:840 |
| 120 | select convert(varchar, getdate(), 120) | 2006-12-30 00:38:54 |
| 121 | select convert(varchar, getdate(), 121) | 2006-12-30 00:38:54.840 |
| 126 | select convert(varchar, getdate(), 126) | 2006-12-30T00:38:54.840 |
| 127 | select convert(varchar, getdate(), 127) | 2006-12-30T00:38:54.840 |
|  | | |
| **FORMATS WITH ISSUES** | | |
| 130 | select convert(varchar, getdate(), 130) | 10 ?? ????? 1427 12:38:54:840A |
| 131 | select convert(varchar, getdate(), 131) | 10/12/1427 12:38:54:840AM |

شما همچنین می توانید تاریخ یا زمان را بدون تقسیم کاراکترها و همچنین ترکیب رشته تاریخ و زمان ترکیب کنید:

|  |  |
| --- | --- |
| **Sample statement** | **Output** |
| select replace(convert(varchar, getdate(),101),'/','') | 12302006 |
| select replace(convert(varchar, getdate(),101),'/','') + replace(convert(varchar, getdate(),108),':','') | 12302006004426 |

اگر می خواهید یک لیست از تمام فرمت های تاریخ و زمان معتبر داشته باشید، می توانید از کد زیر استفاده کنید وdate را برای ()GETDATE یا هر تاریخ دیگری که می خواهید استفاده کنید را تغییر دهید. این فقط فرمت های معتبر را نمایش می دهد.

|  |
| --- |
| DECLARE @counter INT = 0  DECLARE @date DATETIME = '2006-12-30 00:38:54.840'  CREATE TABLE #dateFormats (dateFormatOption int, dateOutput varchar(40))  WHILE (@counter <= 150 )  BEGIN  BEGIN TRY  INSERT INTO #dateFormats  SELECT CONVERT(varchar, @counter), CONVERT(varchar,@date, @counter)  SET @counter = @counter + 1  END TRY  BEGIN CATCH;  SET @counter = @counter + 1  IF @counter >= 150  BEGIN  BREAK  END  END CATCH  END  SELECT \* FROM #dateFormats |

## بدست آوردن بخشی از یک تاریخ در SQL Server

اگر بخواهید بخشی از یک تاریخ و زمان را انتخاب کنید، (مثلا فقط روز یا فقط ساعت و....) می بایست از تابع DATEPART استفاده کنید این تابع دو پارامتر دریافت می کند.   
پارامتر اول مشخص کننده بخشی از تاریخ و زمان که می بایست انتخاب شود و پارامتر دوم تاریخی است که شما می خواهید بر روی آن کار کنید.

به جای datepartEnum می توانید از مقادیر زیر استفاده کنید

|  |  |  |
| --- | --- | --- |
| **datepart** | **Abbreviations** | **Sample** |
| year | yy , yyyy | SELECT datepart(yy,GETDATE()),datepart(yyyy,GETDATE()) |
| quarter | qq , q | SELECT datepart(qq ,GETDATE()) |
| month | mm , m | SELECT datepart(mm ,GETDATE()) |
| dayofyear | dy , y | SELECT datepart(dy ,GETDATE()) |
| day | dd , d | SELECT datepart(dd ,GETDATE()) |
| week | wk , ww | SELECT datepart(wk ,GETDATE()) |
| weekday | dw | SELECT datepart(dw ,GETDATE()) |
| hour | hh | SELECT datepart(hh ,GETDATE()) |
| minute | mi, n | SELECT datepart(mi ,GETDATE()) |
| second | ss , s | SELECT datepart(ss ,GETDATE()) |
| millisecond | ms | SELECT datepart(ms ,GETDATE()) |
| microsecond | mcs | SELECT datepart(mcs ,GETDATE()) |
| nanosecond | ns | SELECT datepart(ns ,GETDATE()) |
| TZoffset | tz |  |
| ISO\_WEEK | isowk , isoww | SELECT datepart(dd ,GETDATE()) |
| SELECT  DATEPART(year, GETDATE())  ,DATEPART(month, GETDATE())  ,DATEPART(day, GETDATE())  ,DATEPART(dayofyear, GETDATE())  ,DATEPART(weekday, GETDATE()); | |  |
| SELECT DATEPART(millisecond, '00:00:01.1234567'); -- Returns 123  SELECT DATEPART(microsecond, '00:00:01.1234567'); -- Returns 123456  SELECT DATEPART(nanosecond, '00:00:01.1234567'); -- Returns 123456700 | | |

## آشنایی با تابع EOMONTH

گاهی اوقات لازم است، تاریخ آخرین روز ماه جاری یا دو ماه بعد‌تر یا یک ماه قبل‌تر و غیرو... را نیاز داشته باشیم. SQL Server در نسخه 2008 خود تابعی ارائه داده است، که تاریخ آخرین روز ماه را برمی گرداند.

این تابع دو پارامتر دریافت می‌کند، اولین پارامتر یک فرمت تاریخ می‌پذیرد، دومین پارامتر، اختیاری است و یک عدد می‌پذیرد و بیانگر تعداد ماه بعد از تاریخ یا تعداد ماه قبل از تاریخ، پارامتر اول می‌باشد.

|  |
| --- |
| SELECT EOMONTH('20110201') as 'آخرین روز ماه فوریه در سال 2011';  SELECT EOMONTH('20120201') as 'آخرین روز ماه فوریه در سال 2012';  SELECT EOMONTH('20130201') as 'آخرین روز ماه فوریه در سال 2013';  SELECT EOMONTH(GETDATE(),-2) as 'آخرین روز ماه سپتامبر' |

## مقایسه دو تاریخ در SQL Server

در صورتی که بخواهید دو تاریخ را با یکدگیر مقایسه کنید می توانید از تابع DATEDIFF استفاده نمایید.

این تابع سه پارارمتر دارد:  
پارمتر اول مشخص کننده واحدی (سال، ماه، روز، ساعت و...) است که شما می خواهید بر اساس آن مقایسه انجام شود و پارامتر دوم و سوم تاریخ شما برای مقایسه می باشد.

|  |
| --- |
| SELECT DATEDIFF(year, '2012/04/28', '2014/04/28'); Result: 2  SELECT DATEDIFF(yyyy, '2012/04/28', '2014/04/28'); Result: 2  SELECT DATEDIFF(yy, '2012/04/28', '2014/04/28'); Result: 2  SELECT DATEDIFF(month, '2014/01/01', '2014/04/28'); Result: 3  SELECT DATEDIFF(day, '2014/01/01', '2014/04/28'); Result: 117  SELECT DATEDIFF(hour, '2014/04/28 08:00', '2014/04/28 10:45'); Result: 2  SELECT DATEDIFF(minute, '2014/04/28 08:00', '2014/04/28 10:45'); Result: 165  SELECT GETDATE()-GETDATE()+3 |
|  |

نحوه استفاده از فرمت میلادی به فرمت تبدیل

|  |  |
| --- | --- |
| select CAST(GETDATE() AS datetime2(0))  select CAST(GETDATE() AS datetime2(1))  select CAST(GETDATE() AS datetime2(2))  select CAST(GETDATE() AS datetime2(5)) |  |
| declare @now datetime  set @now = getDate()  select datename(yyyy, @now) as year  , right(N'0' + convert(nvarchar(2), month(@now)), 2) as month  , right(N'0' + datename(d, @now), 2) as day  , right(N'0' + datename(hh, @now), 2) as hour  , right(N'0' + datename(n, @now), 2) as minute  , right(N'0' + datename(s, @now), 2) as second |  |
| SELECT  FORMAT(DATEPART(yyyy, GETDATE()), '0000') +  '/' +  FORMAT(DATEPART(dd, GETDATE()), '00')  + '/' +  FORMAT(DATEPART(mm, GETDATE()), '00')  + ' ' +  CONVERT(CHAR(8), GETDATE(), 108); |  |
| SELECT FORMAT(GETDATE() , 'MM/dd/yyyy HH:mm:ss')  SELECT FORMAT(GETDATE() , 'yyyy/dd/MM HH:mm:ss')  SELECT FORMAT(GETDATE() , 'yyyy-dd-MM HH:mm:ss') |  |
| select convert(varchar, getdate(), 8) |  |

نحوه استفاده از متد DateAdd :

|  |  |
| --- | --- |
|  | SELECT DATEADD(day,1,getdate()),getdate() |

اختلاف بین 2 ساعت

|  |
| --- |
| DECLARE @MyTable TABLE(Enters TIME(0), LeftTime TIME(0));  INSERT @MyTable VALUES('07:00:00', '15:45:00')  SELECT \*,  FORMAT(CAST(DATEDIFF(HOUR, Enters, LeftTime) AS int),'00')+':'+FORMAT(CAST(DATEDIFF(MINUTE, Enters, LeftTime)% 60 AS INT),'00') AS JobTime  FROM @MyTable; |

تابع کمکی تبدیل تاریخ میلادی به شمسی

ابتدا به کمک فانکشن PersianToJulian تبدیل روز به میلادی انجام میدهیم

|  |
| --- |
| CREATE FUNCTION dbo.[PersianToJulian](@iYear int,@iMonth int,@iDay int)  RETURNS bigint  AS  Begin    Declare @PERSIAN\_EPOCH as int  Declare @epbase as bigint  Declare @epyear as bigint  Declare @mdays as bigint  Declare @Jofst as Numeric(18,2)  Declare @jdn bigint    Set @PERSIAN\_EPOCH=1948321  Set @Jofst=2415020.5    If @iYear>=0  Begin  Set @epbase=@iyear-474  End  Else  Begin  Set @epbase = @iYear - 473  End  set @epyear=474 + (@epbase%2820)  If @iMonth<=7  Begin  Set @mdays=(Convert(bigint,(@iMonth) - 1) \* 31)  End  Else  Begin  Set @mdays=(Convert(bigint,(@iMonth) - 1) \* 30+6)  End  Set @jdn =Convert(int,@iday) + @mdays+ Cast(((@epyear \* 682) - 110) / 2816 as int) + (@epyear - 1) \* 365 + Cast(@epbase / 2820 as int) \* 1029983 + (@PERSIAN\_EPOCH - 1)  RETURN @jdn  End |

فانکشن EnToPersian جهت تبدیل میلادی به شمسی

|  |
| --- |
| CREATE Function dbo.[EnToPersian] (@date datetime)  Returns VARCHAR(10)  as  Begin  Declare @depoch as bigint  Declare @cycle as bigint  Declare @cyear as bigint  Declare @ycycle as bigint  Declare @aux1 as bigint  Declare @aux2 as bigint  Declare @yday as bigint  Declare @Jofst as Numeric(18,2)  Declare @jdn bigint    Declare @iYear As Integer  Declare @iMonth As Integer  Declare @iDay As Integer  set @date=(cast(cast(@date as date)as varchar(20)))  Set @Jofst=2415020.5  Set @jdn=Round(Cast(@date as int)+ @Jofst,0)    Set @depoch = @jdn - dbo.[PersianToJulian](475, 1, 1)  Set @cycle = Cast(@depoch / 1029983 as int)  Set @cyear = @depoch%1029983    If @cyear = 1029982  Begin  Set @ycycle = 2820  End  Else  Begin  Set @aux1 = Cast(@cyear / 366 as int)  Set @aux2 = @cyear%366  Set @ycycle = Cast(((2134 \* @aux1) + (2816 \* @aux2) + 2815) / 1028522 as int) + @aux1 + 1  End    Set @iYear = @ycycle + (2820 \* @cycle) + 474    If @iYear <= 0  Begin  Set @iYear = @iYear - 1  End  Set @yday = (@jdn - dbo.[PersianToJulian](@iYear, 1, 1)) + 1  If @yday <= 186  Begin  Set @iMonth = CEILING(Convert(Numeric(18,4),@yday) / 31)  End  Else  Begin  Set @iMonth = CEILING((Convert(Numeric(18,4),@yday) - 6) / 30)  End  Set @iDay = (@jdn - dbo.[PersianToJulian](@iYear, @iMonth, 1)) + 1    Return format(CAST(@iYear as INT),'0000')+ '/' +format(CAST(@iMonth as INT),'00') +'/' +format(cast(@iDay as INT),'00')  End |

* SELECT dbo.EnToPersian(GETDATE()) 1397/08/25

فانکشن EnToPersianTime جهت تبدیل میلادی به شمسی همراه با ساعت

|  |
| --- |
| CREATE Function dbo.[EnToPersianTime] (@date datetime)  Returns nvarchar(50)  as  Begin  Declare @depoch as bigint  Declare @cycle as bigint  Declare @cyear as bigint  Declare @ycycle as bigint  Declare @aux1 as bigint  Declare @aux2 as bigint  Declare @yday as bigint  Declare @Jofst as Numeric(18,2)  Declare @jdn bigint  Declare @Time nvarchar(20)=cast(@date as Time(0))  Declare @iYear As Integer  Declare @iMonth As Integer  Declare @iDay As Integer  set @date=(cast(cast(@date as date)as varchar(20)))  Set @Jofst=2415020.5  Set @jdn=Round(Cast(@date as int)+ @Jofst,0)    Set @depoch = @jdn - dbo.[PersianToJulian](475, 1, 1)  Set @cycle = Cast(@depoch / 1029983 as int)  Set @cyear = @depoch%1029983    If @cyear = 1029982  Begin  Set @ycycle = 2820  End  Else  Begin  Set @aux1 = Cast(@cyear / 366 as int)  Set @aux2 = @cyear%366  Set @ycycle = Cast(((2134 \* @aux1) + (2816 \* @aux2) + 2815) / 1028522 as int) + @aux1 + 1  End    Set @iYear = @ycycle + (2820 \* @cycle) + 474    If @iYear <= 0  Begin  Set @iYear = @iYear - 1  End  Set @yday = (@jdn - dbo.[PersianToJulian](@iYear, 1, 1)) + 1  If @yday <= 186  Begin  Set @iMonth = CEILING(Convert(Numeric(18,4),@yday) / 31)  End  Else  Begin  Set @iMonth = CEILING((Convert(Numeric(18,4),@yday) - 6) / 30)  End  Set @iDay = (@jdn - dbo.[PersianToJulian](@iYear, @iMonth, 1)) + 1    Return format(CAST(@iYear as INT),'0000')+ '/' +format(CAST(@iMonth as INT),'00') +'/' +format(cast(@iDay as INT),'00') +' '+@Time  End |

فانکشن PersianToEn برای تبدیل شمسی به میلادی

|  |
| --- |
| CREATE FUNCTION dbo.[PersianToEn](@Date nchar(10))  RETURNS nvarchar(10)  AS  Begin    Declare @PERSIAN\_EPOCH as int  Declare @epbase as bigint  Declare @epyear as bigint  Declare @mdays as bigint  Declare @Jofst as Numeric(18,2)  Declare @jdn bigint  Declare @iYear int =cast(substring(@Date,1,4)as int)  Declare @iMonth int =cast(substring(@Date,6,2)as int)  Declare @iDay int =cast(substring(@Date,9,2)as int)    Set @PERSIAN\_EPOCH=1948321  Set @Jofst=2415020.5    If @iYear>=0  Begin  Set @epbase=@iyear-474  End  Else  Begin  Set @epbase = @iYear - 473  End  set @epyear=474 + (@epbase%2820)  If @iMonth<=7  Begin  Set @mdays=(Convert(bigint,(@iMonth) - 1) \* 31)  End  Else  Begin  Set @mdays=(Convert(bigint,(@iMonth) - 1) \* 30+6)  End  Set @jdn =Convert(int,@iday) + @mdays+ Cast(((@epyear \* 682) - 110) / 2816 as int) + (@epyear - 1) \* 365 + Cast(@epbase / 2820 as int) \* 1029983 + (@PERSIAN\_EPOCH - 1)    RETURN Convert(nvarchar(11),Convert(datetime,(@jdn- @Jofst),131),120)    End |

* SELECT dbo.PersianToEn('1397/08/22') 2018-11-13

فانکشن PersianToEnTime برای تبدیل شمسی به میلادی همراه با ساعت

|  |
| --- |
| CREATE FUNCTION dbo.[PersianToEnTime](@Date nvarchar(max))  RETURNS nvarchar(20)  AS  Begin  set @Date=replace(@Date,' ','')  Declare @PERSIAN\_EPOCH as int  Declare @epbase as bigint  Declare @epyear as bigint  Declare @mdays as bigint  Declare @Jofst as Numeric(18,2)  Declare @jdn bigint  Declare @iYear int =cast(substring(@Date,1,4)as int)  Declare @iMonth int =cast(substring(@Date,6,2)as int)  Declare @iDay int =cast(substring(@Date,9,2)as int)    Set @PERSIAN\_EPOCH=1948321  Set @Jofst=2415020.5    If @iYear>=0  Begin  Set @epbase=@iyear-474  End  Else  Begin  Set @epbase = @iYear - 473  End  set @epyear=474 + (@epbase%2820)  If @iMonth<=7  Begin  Set @mdays=(Convert(bigint,(@iMonth) - 1) \* 31)  End  Else  Begin  Set @mdays=(Convert(bigint,(@iMonth) - 1) \* 30+6)  End  Set @jdn =Convert(int,@iday) + @mdays+ Cast(((@epyear \* 682) - 110) / 2816 as int) + (@epyear - 1) \* 365 + Cast(@epbase / 2820 as int) \* 1029983 + (@PERSIAN\_EPOCH - 1)    RETURN Convert(nvarchar(11),Convert(datetime,(@jdn- @Jofst),131),120)+substring(@Date,11,len(@Date))    End |

* SELECT dbo.PersianToEnTime (N'1397/08/26 11:51:10') 2018-11-13

تابعی برای تبدیل تاریخ وارد شده به صورت (روز – تاریخ – ساعت)

|  |
| --- |
| CREATE FUNCTION dbo.[ConvertMinutesToDay](@theMinutes int)  RETURNS nvarchar(150)  AS  begin  DECLARE @time nvarchar(150)  Select  @time=  case WHEN cast(@theMinutes / 1440 as varchar(50))=0 then '' else cast(@theMinutes / 1440 as varchar(50))+' روز'end +''+  case when cast((@theMinutes % 1440) / 60 as varchar(50))=0 then '' else cast((@theMinutes % 1440) / 60 as varchar(50)) +' ساعت ' end+''+  case when cast((@theMinutes % 60) as varchar(50))=0 then '' else cast((@theMinutes % 60) as varchar(50))+' دقیقه ' end  RETURN @time  END |

* SELECT dbo.[ConvertMinutesToDay](1480) 1 روز40 دقيقه
* SELECT dbo.[ConvertMinutesToDay](60) 1 ساعت