



## Beregning af arbejdsdage & skæve helligdage

### Funktioner til beregning af helligdage (Påske, Pinse etc) og arbejdsdage.

Skrevet den **04. Feb 2009** af **trer** | kategorien **Databaser / MS SQL** | ★★★★★

For at kunne beregne arbejdsdage og helligdage er man nød til at dele dagene op i forskellige kategorier.

1. Skæve Helligdage
2. Faste Helligdage
3. Arbejdsfri dage
4. Weekender
5. Almindelige arbejdsdage

De skæve helligdage er dem der ikke ligger fast, men beregnes ud fra påsken. For at kunne finde dem, er det derfor nødvendigt at kunne beregne påsken.

Påsken er altid placeret første søndag efter første fuldmåne efter forårsjævndøgn. Det lyder grimt at beregne, men heldigvis er Påsken den kristne kirkes vigtigste højtid, så man har brugt ret mange kræfter på at finde simple måder at beregne den på.

Nedenstående funktion beregner påsken ud fra Gauás algoritme, den er begrænset til årene 1583 til 2299, og SQL Servers datoimplementering begrænser den yderligere til årene 1750 til 2299. Det burde dog stadig være nok for de fleste...

```
create function EasterDate (@eYear int)  
returns datetime  
as  
begin  
  /* Fischer Lexikon Astronomie, p. 50:  
   Algorithm of C.F. Gauá (1777-1855),  
   for 1583 <= eyear <= 2299  
  */  
  
  declare @m int, @n int, @p int, @q int, @r int, @x int  
  declare @y int, @a int, @b int, @c int, @d int, @e int  
  declare @eDay int, @eMonth int  
  
  /* Kan tilføjes hvis MsSQL senere understøtter flere år  
   if[/b] (@eYear <= 1699) select @m=22, @n=2 else  
  */  
  if (@eYear <= 1799) select @m=23, @n=3 else  
  if (@eYear <= 1899) select @m=23, @n=4 else  
  if (@eYear <= 2099) select @m=24, @n=5 else  
  if (@eYear <= 2199) select @m=24, @n=6 else  
  if (@eYear <= 2299) select @m=25, @n=0  
  
  select @a = @eYear % 19, @b = @eYear % 4, @c = @eYear % 7  
  select @d = ( (19*@a)+@m ) % 30
```

```

select @e = ( (2*@b)+(4*@c)+(6*@d)+@n ) % 7
select @eDay = 22 + @d + @e
select @eMonth = 4

if @eDay <=31 set @eMonth=3 else
if ( (@d=28) and (@e=6) and (@a>10) ) set @eDay=18 else
if ( (@d=29) and (@e=6) ) set @eday = 19 else
    select @eday = @d + @e - 9
return convert(
    datetime,
    cast(@eYear as varchar)+'/'+
    cast(@eMonth as varchar)+'/'+
    cast(@eDay as varchar),
    111)
end

```

Funktionen skal have et årstal som input og returnerer så datoen for påsken det pågældende år.

Beregningen af faste helligdage (fx første juledag) og de arbejdsfri dage er ulige meget nemmere. For at finde dem skal man blot sammenligne den ønskede måned og dag med nogle kendte værdier.

De skæve helligdage er nu påsken er fundet heller ikke meget sværere: Pinsen er den 50. dag efter påske, Kr. himmelfart den 40. dag etc.

Funktionen nedenfor forsøger først at finde den givne dato som en fast helligdag eller arbejdsfri dag, lykkes det ikke forsøges det om det er en af de skæve.

Input er altså en dato, og funktionen returnerer et 1 hvis datoen er en helligdag/arbejdsfriday og 0 hvis det ikke er.

```

create function IsDanishHolyDay (@date datetime)
returns tinyint
as
begin
    declare @d int, @m int, @y int, @r int, @e datetime
    select @d=day(@date), @m=month(@date), @y=year(@date), @r=0
    -- faste helligdage
    if (@d= 1 and @m= 1) or -- Nytårsdag
        (@d= 5 and @m= 6) or -- Grundlovsdag
        (@d=24 and @m=12) or -- Juleaftensdag
        (@d=25 and @m=12) -- 1. juledag
        /* yderligere datoer kan indsættes her
        */
        set @r = 1
    else begin
        -- Skæve helligdage, beregnes udfra Påske
        set @date = floor(cast(@date as float))
        set @e = dbo.EasterDate(@y)
        if (@date=@e- 7) or -- Palmesøndag
            (@date=@e- 3) or -- Skærtorsdag
            (@date=@e- 2) or -- Langfredag
            (@date=@e+ 0) or -- Påske
    end
end

```

```

    (@date=@e+ 1) or -- 2. Påskedag
    (@date=@e+26) or -- St. Bededag
    (@date=@e+39) or -- Kristi Himmelfart
    (@date=@e+49) or -- Pinse
    (@date=@e+50) -- 2. Pinsedag
    set @r = 1
end
return @r
end

```

Sidste funktion finder ud af, om en given dag er i en weekend eller en af hverdagene. Det er kun nødvendigt at checke for helligdage hvis dagen ikke ligger i en Weekend.

Jeg har valgt at implementere at weekenden altid er ugens to sidste dage, så parametren @@datefirst benyttes til at sikre det rigtige offset (I Danmark og det meste af Europa benyttes ISO standarden, at ugen starter på mandage, i USA benytter man typisk at ugen starter søndag).

```

create function isWorkDay(@date datetime)
returns tinyint
as
begin
    declare @r int
    set @r=@@datefirst-1+datepart(weekday,@date)
    -- Lørdag eller Søndag ?
    if (@r=6) or (@r=7)
        set @r = 1
    else
        set @r = dbo.IsDanishHolyDay(@date)
    return abs(@r-1)
end

```

Funktionen tager stadig en dato som input, og leverer et 1 tal såfremt det er en arbejdsdag og et 0 såfremt det er en helligdag eller weekend.

God fornøjelse  
Troels

**Kommentar af hnteknik d. 23. Feb 2005 | 1**

**Kommentar af ldanielsen d. 29. Feb 2004 | 2**

Sådan en find-påskan-funktion har jeg haft brug for længe, og har tidligere opgivet at finde den, men her var den så ...

**Kommentar af clausn d. 23. Mar 2005 | 3**

**Kommentar af stig-b d. 18. Aug 2005 | 4**

Fed artikel..!

**Kommentar af thomasjepsen d. 31. Mar 2004 | 5**

Ganske smart, at man kan regne sig frem til alle helligdage (istedet for at skulle indtaste dem i en tabel).  
En af dagene vil jeg se at kode den om til VBA (hvis du allerede har gjort det, Trer, kunne du så ikke lige sende den til mig? ;o)  
Godt arbejde! /Thomas Jepsen

**Kommentar af binow d. 29. Sep 2004 | 6**

**Kommentar af jss d. 07. Sep 2005 | 7**

God artikel som rammer lige i plet på en (nu tidligere...) udfordring jeg sidder med :-)

**Kommentar af jpvj d. 27. Sep 2004 | 8**

Really nice!

**Kommentar af al1407 d. 24. Jun 2004 | 9**

Rigtig god artikel, må man sige!

**Kommentar af jim\_marius d. 07. May 2007 | 10**

Perfekt, lige hvad jeg manglede.

**Kommentar af rstg1 d. 04. Jul 2006 | 11**

Det er en vældig god artikel, som giver svaret.  
(Har du koden i ASP ?).