

-- DAN KALENDER SCRIPTS

```
-----  
-- Følgende MS-SQL scripts danner først en række funktioner. Disse bliver gemt i      --  
-- databasen under Programmability, Functions, Scalar-Valued functions.             --  
-- Efter funktions scripts kommer et script til dannelse af en tids tabel.          --  
-- Sidste script benytter de givne funktioner og indsætter en rekord for hver dag som --  
-- man ligeledes definerer (start og slutdato) i dette sidste script.               --  
-- BEMÆRK: Man kan ændre tabelnavnet, men skal huske at ændre det også i starten lige --  
-- inden hvor en eventuelt tidligere skabt tabel med samme navn bliver slettet.      --  
-- Ligeledes skal man da ændre navnet i det sidste script hvor man indsætter rekords. --  
-----
```

```
-----  
-- Alle efterfølgende scripts starter med at tjekke, om der ligger --  
-- en funktion eller tabel i forvejen med samme navn, og hvis ja --  
-- så slettes foregående før nyt script kører, som opretter på ny. --  
-----
```

```
if exists (select * from sys.sysobjects where id = object_id(N'[dbo].[EasterDate]') and xtype in (N'FN',  
N'IF', N'TF'))
```

```
drop function [dbo].[EasterDate]
```

```
GO
```

```
-----  
-- Nedenstående finder datoen for hvornår påskedag er et givet år --  
-- Helligdage der rykker sig, beregnes alle ud fra denne dag --  
-- Denne funktion benyttes i senere beregning af danske helligdage --  
-----
```

```
create function dbo.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 (@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

GO

/* Funktion til udregning af danske helligdage */

if exists (select * from sys.sysobjects where id = object_id(N'[dbo].[IsDanishHolyDay]') and xtype in (N'FN', N'IF', N'TF'))

drop function [dbo].[IsDanishHolyDay]

GO

create function IsDanishHolyDay (@date datetime) returns tinyint as begin

 -- variablen @r sættes lig med en (1) hvis det er en helligdag --

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) or -- Juledag

 (@d=26 and @m=12) or -- 2. juledag

 (@d=31 and @m=12) -- Nytårsaftensdag

/* yderligere datoer kan indsættes her */

-- Her kunne indsættes virksomheds specifikke --

-- lukkedage som 'helligdage' --

-- Bemærk, at bl.a. Grundlovsdag er def. som --

-- helligdag i ovenstaaende --

```

set @r = 1 else begin
-- Skæve helligdage, beregnes ud fra Påske
set @date = floor(cast(@date as float))
set @e = dbo.EasterDate(@y)
if
    (@date=@e-3) or -- Skærtorsdag
    (@date=@e-2) or -- Langfredag
    (@date=@e+0) or -- Påskedag
    (@date=@e+1) or -- 2. Påskedag
-- Store bededag er sløjftet ved lov, så ikke med længere
    --(@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
GO

if exists (select * from sys.sysobjects where id = object_id(N'[dbo].[NameOfDanishHolyDay]') and xtype
in (N'FN', N'IF', N'TF'))
drop function [dbo].[NameOfDanishHolyDay]

GO

create function NameOfDanishHolyDay (@date datetime) returns varchar(17) as begin

-- variablen @r sættes lig med navnet på helligdagen --

```

```

declare @d int, @m int, @y int, @r varchar(17), @e datetime
set @r=""

select @d=day(@date), @m=month(@date), @y=year(@date)

-- faste helligdage

```

```

IF @d=1 and @m=1 BEGIN set @r='Nytårsdag' end
IF @d=5 and @m=6 BEGIN set @r='Grundlovsdag' end
IF @d=24 and @m=12 BEGIN set @r='Juleaftensdag' end
IF @d=25 and @m=12 BEGIN set @r='Juledag' end
IF @d=26 and @m=12 BEGIN set @r='2. juledag' end
IF @d=31 and @m=12 BEGIN set @r='Nytårsaftensdag' end

```

```

/* yderligere datoer kan indsættes her */

```

```

-----

```

```

-- Her kunne indsættes virksomheds specifikke --

```

```

-- lukkedage som 'helligdage' --

```

```

-- Denne funktion skal rettes synkron med --

```

```

-- funktionen IsDanishHolyday --

```

```

-----

```

```

-- Skæve helligdage, beregnes ud fra Påske

set @date = floor(cast(@date as float))
set @e = dbo.EasterDate(@y)

IF @date=@e-3 BEGIN SET @r='Skærtorsdag'END
IF @date=@e-2 BEGIN SET @r='Langfredag'END
IF @date=@e+0 BEGIN SET @r='Påskedag'END
IF @date=@e+1 BEGIN SET @r='2. Påskedag' END

```

```

IF @date=@e+26 BEGIN SET @r='St. Bededag'END
IF @date=@e+39 BEGIN SET @r='Kristi Himmelfart'END
IF @date=@e+49 BEGIN SET @r='Pinsedag' END
IF @date=@e+50 BEGIN SET @r='2. Pinsedag'END

```

```

    return @r
end
GO

```

```

/* Funktion der finder ud af om det er en week-end dag */

```

```

if exists (select * from sys.sysobjects where id = object_id(N'[dbo].[isWeekend]') and xtype in (N'FN',
N'IF', N'TF'))

```

```

drop function [dbo].[isWeekend]

```

```

GO

```

```

CREATE function isWeekend(@date datetime) returns tinyint as

```

```

begin

```

```

-- find 1. dag

```

```

Declare @x as int

```

```

Set @x= @@DATEFIRST

```

```

Declare @s as int

```

```

Declare @l as int

```

```

Set @s= (select case when @x=7 then 1 else 7 end)

```

```

Set @l= (select case when @x=7 then 7 else 6 end)

```

```

declare @r int

```

```

set @r=datepart(weekday,@date) -- Lørdag eller Søndag ?

```

```

if (@r=@l) or (@r=@s) -- Weekend dage
set @r = 1
else
set @r = 0
return abs(@r)
end
GO

```

/* Funktion der finder ud af om det er en week-end og| eller helligdag */

```

if exists (select * from sys.sysobjects where id = object_id(N'[dbo].[isHolyWeekend]') and xtype in (N'FN',
N'IF', N'TF'))
drop function [dbo].[isHolyWeekend]
GO

```

```

CREATE function isHolyWeekend(@date datetime) returns tinyint as
begin
-- find 1. dag
Declare @x as int
Set @x= @@DATEFIRST
Declare @s as int
Declare @l as int
Set @s= (select case when @x=7 then 1 else 7 end)
Set @l= (select case when @x=7 then 7 else 6 end)
declare @r int
set @r=datepart(weekday,@date) -- Lørdag eller Søndag ?
if (@r=@l) or (@r=@s) -- Weekend dage
set @r = 1

```

```

else
set @r = dbo.IsDanishHolyDay(@date)
return abs(@r)
end
GO

```

```

/* Funktion der finder ud af om det er mandag til fredag */

```

```

if exists (select * from sys.sysobjects where id = object_id(N'[dbo].[ismanfre]') and xtype in (N'FN', N'IF',
N'TF'))
drop function [dbo].[ismanfre]
GO

```

```

CREATE function ismanfre(@date datetime) returns tinyint as
begin
-- find 1. dag
Declare @x as int
Set @x= @@DATEFIRST
Declare @s as int
Declare @l as int
Set @s= (select case when @x=7 then 1 else 7 end)
Set @l= (select case when @x=7 then 7 else 6 end)
declare @r int
set @r=datepart(weekday,@date) -- Lørdag eller Søndag ?
if (@r=@l) or (@r=@s) -- Weekend dage
set @r = 0
else
set @r = 1
return abs(@r)

```

end

GO

/* Funktion der finder ud af om det er en arbejdsdag eller ikke */

if exists (select * from sys.sysobjects where id = object_id(N'[dbo].[isWorkDay]') and xtype in (N'FN', N'IF', N'TF'))

drop function [dbo].[isWorkDay]

GO

CREATE function isWorkDay(@date datetime) returns tinyint as

begin

-- find 1. dag

Declare @x as int

Set @x= @@DATEFIRST

Declare @s as int

Declare @l as int

Set @s= (select case when @x=7 then 1 else 7 end)

Set @l= (select case when @x=7 then 7 else 6 end)

declare @r int

set @r=datepart(weekday,@date) -- Lørdag eller Søndag ?

if (@r=@l) or (@r=@s) -- Weekendtage

set @r = 1 else

set @r = dbo.IsDanishHolyDay(@date)

return abs(@r-1)

end

GO

```

/* Funktion der beregner ugenummer ud fra 3 input variable: */
/* År, måned og dag. */
/* Ugenummer indsættes efter ISO 8601 standard, hvor ugen */
/* starter på en mandag. (Europæisk uge) */

```

```

if exists (select * from sys.sysobjects where id = object_id(N'[dbo].[Ugenummer]') and xtype in (N'FN',
N'IF', N'TF'))

```

```

drop function [dbo].[ugenummer]

```

```

GO

```

```

create function Ugenummer(@aar int, @maaned int, @dag int) returns int as begin

```

```

-----
-- ISO 8601 definition af ugenummer. Ugenumre starter på en mandag --
-----
-- Der er beregninger i denne funktion der kunne udføres med MOD --
-- funktion direkte, som er omskrevet til beregning af remainers --
-- som normalt benytter INT funktion, som her er erstattet af --
-- FLOOR funktionen. Generelt alle steder hvor der står FLOOR i --
-- denne funktion kunne det erstattes med INT --
-----

```

```

declare @a int, @b int, @c int, @d int, @e int, @f int, @g int, @n int, @s int, @ret int

```

```

-----
-- I denne funktion er der forskel på variable afhængig af om --
-- måned = jan-feb eller øvrige måneder --
-----

```

```

-- Her kommer variable beregninger --

```

```

if @maaned<3 BEGIN set @a=@aar-1 end

if @maaned>2 BEGIN set @a=@aar end


set @b=floor(@a/4)-floor(@a/100)+floor(@a/400)
set @c=floor((@a-1)/4)-floor((@a-1)/100)+floor((@a-1)/400)
set @s=@b-@c

if @maaned<3 BEGIN

    set @e=0

    set @f=@dag-1+31*(@maaned-1)
end

if @maaned>2 BEGIN

    set @e=@s+1

    set @f=@dag+floor((153*(@maaned-3)+2)/5)+58+@s
end

set @g=(@a+@b)-7*floor((@a+@b)/7)
set @d=(@f+@g-@e)-7*floor((@f+@g-@e)/7)
set @n=@f+3-@d


-- Værdien af variable @n bestemmer ugenummer --
-- Hvis @n er negativ er ugenummer en ting --
-- Hvis @n>364 er ugenummer 1 --
-- Hvis @n ligger mellem 0-364 beregnes den --
-- efter sidste formel --


if @n<0 BEGIN

    set @ret=53-floor((@g-@s)/5)
end

if @n>364 BEGIN

```

```

        set @ret=1
    end

    if @n>=0 and @n<=364 BEGIN
        set @ret=floor(@n/7)+1
    end

    return @ret
end
GO

```

```

if exists (select * from sys.sysobjects where id = object_id(N'[dbo].[Intdato]') and xtype in (N'FN', N'IF',
N'TF'))
drop function [dbo].[Intdato]
GO

```

Create function Intdato (@aar int, @mdr int, @dag int) returns int as begin

```

    -- Beregner en integer hvor første 4 cifre angiver årstal
    -- næste 2 cifre angiver månedsnummer
    -- sidste 2 cifre angiver dagsnummer

    set @aar=@aar*100
    set @aar=@aar+@mdr
    set @aar=@aar*100
    set @aar=@aar+@dag

    return @aar
end
GO

```

```
if exists (select * from sys.sysobjects where id = object_id(N'[dbo].[MdrUge]') and xtype in (N'FN', N'IF', N'TF'))
```

```
drop function [dbo].[Mdruge]
```

```
GO
```

```
create function MdrUge(@aar int, @maaned int, @dag int) returns varchar(5) as begin
```

```
-----
```

```
-- ISO 8601 definition af ugenummer. Ugenumre starter på en mandag --
```

```
-----
```

```
-- Der er beregninger i denne funktion der kunne udføres med MOD --
```

```
-- Returnerer mdr og uge nr som varchar - UGE 1 problematik --
```

```
-----
```

```
declare @a int, @b int, @c int, @d int, @e int, @f int, @g int, @n int, @s int, @ret int
```

```
declare @mdr varchar(2)
```

```
declare @uge varchar(2)
```

```
declare @retur varchar(5)
```

```
-----
```

```
-- I denne funktion er der forskel på variable afhængig af om --
```

```
-- måned = jan-feb eller øvrige måneder --
```

```
-----
```

```
-- Her kommer variable beregninger --
```

```
if @maaned<3 BEGIN set @a=@aar-1 end
```

```
if @maaned>2 BEGIN set @a=@aar end
```

```

set @b=floor(@a/4)-floor(@a/100)+floor(@a/400)
set @c=floor((@a-1)/4)-floor((@a-1)/100)+floor((@a-1)/400)
set @s=@b-@c
if @maaned<3 BEGIN
    set @e=0
    set @f=@dag-1+31*(@maaned-1)
end
if @maaned>2 BEGIN
    set @e=@s+1
    set @f=@dag+floor((153*(@maaned-3)+2)/5)+58+@s
end
set @g=(@a+@b)-7*floor((@a+@b)/7)
set @d=(@f+@g-@e)-7*floor((@f+@g-@e)/7)
set @n=@f+3-@d

```

```

-- Værdien af variable @n bestemmer ugenummer --
-- Hvis @n er negativ er ugenummer en ting --
-- Hvis @n>364 er ugenummer 1 --
-- Hvis @n ligger mellem 0-364 beregnes den --
-- efter sidste formel --

```

```

if @n<0 BEGIN
    set @ret=53-floor((@g-@s)/5)
end
if @n>364 BEGIN
    set @ret=1
end
if @n>=0 and @n<=364 BEGIN
    set @ret=floor(@n/7)+1

```

```

end

if @maaned<10 BEGIN
    set @mdr='0'+cast(@maaned as varchar)
end

if @maaned>=10 BEGIN
    set @mdr=cast(@maaned as varchar(2))
end

if @ret<10 BEGIN
    set @uge='0'+cast(@ret as varchar(1))
end

if @ret>=10 BEGIN
    set @uge=cast(@ret as varchar(2))
end

set @retur=@mdr+'-'+@uge

return @retur
end
GO

if exists (select * from sys.sysobjects where id = object_id(N'[dbo].[AarUge]') and xtype in (N'FN', N'IF',
N'TF'))
drop function [dbo].[Aaruge]
GO

create function AarUge(@aar int, @maaned int, @dag int) returns varchar(7) as begin
    -----
    -- ISO 8601 definition af ugenummer. Ugenumre starter på en mandag --

```

```

-----
-- Der er beregninger i denne funktion der kunne udføres med MOD --
-- Returnerer År og uge nr som varchar - UGE 1 problematik      --
-- Hvis uge 1 ligger sidst i året sættes sættes år lig med år+1 --
-----

```

```

declare @a int, @b int, @c int, @d int, @e int, @f int, @g int, @n int, @s int, @ret int

```

```

declare @vaar varchar(4)

```

```

declare @uge varchar(2)

```

```

declare @retur varchar(7)

```

```

-----
-- I denne funktion er der forskel på variable afhængig af om --
-- måned = jan-feb eller øvrige måneder                --
-----

```

```

-- Her kommer variable beregninger --

```

```

if @maaned<3 BEGIN set @a=@aar-1 end

```

```

if @maaned>2 BEGIN set @a=@aar end

```

```

set @b=floor(@a/4)-floor(@a/100)+floor(@a/400)

```

```

set @c=floor((@a-1)/4)-floor((@a-1)/100)+floor((@a-1)/400)

```

```

set @s=@b-@c

```

```

if @maaned<3 BEGIN

```

```

    set @e=0

```

```

    set @f=@dag-1+31*(@maaned-1)

```

```

end

```

```

if @maaned>2 BEGIN

```

```

set @e=@s+1

set @f=@dag+floor((153*(@maaned-3)+2)/5)+58+@s
end

set @g=(@a+@b)-7*floor((@a+@b)/7)
set @d=(@f+@g-@e)-7*floor((@f+@g-@e)/7)
set @n=@f+3-@d

```

```

-- Værdien af variable @n bestemmer ugenummer --
-- Hvis @n er negativ er ugenummer en ting --
-- Hvis @n>364 er ugenummer 1 --
-- Hvis @n ligger mellem 0-364 beregnes den --
-- efter sidste formel --

```

```

if @n<0 BEGIN
    set @ret=53-floor((@g-@s)/5)
end

if @n>364 BEGIN
    set @ret=1
end

if @n>=0 and @n<=364 BEGIN
    set @ret=floor(@n/7)+1
end

```

```

set @vaar=cast(@aar as varchar(4))

if @maaned=12 BEGIN
    if @ret=1 begin
        set @vaar=cast((@aar+1) as varchar(4))
    end
end
end

```

```

if @ret<10 BEGIN
    set @uge='0'+cast(@ret as varchar(1))
end
if @ret>=10 BEGIN
    set @uge=cast(@ret as varchar(2))
end

set @retur=@vaar+'-'+@uge

return @retur
end
GO

-- Danner unik årkvartal til hierarki --

if exists (select * from sys.sysobjects where id = object_id(N'[dbo].[Aarkvartal]') and xtype in (N'FN',
N'IF', N'TF'))
drop function [dbo].[Aarkvartal]
go

create function Aarkvartal(@aar int, @maaned int) returns int as Begin
declare @kvt int

if @maaned=1 set @kvt=@aar*10+1
if @maaned=2 set @kvt=@aar*10+1
if @maaned=3 set @kvt=@aar*10+1
if @maaned=4 set @kvt=@aar*10+2
if @maaned=5 set @kvt=@aar*10+2
if @maaned=6 set @kvt=@aar*10+2

```

```

if @maaned=7 set @kvt=@aar*10+3
if @maaned=8 set @kvt=@aar*10+3
if @maaned=9 set @kvt=@aar*10+3
if @maaned=10 set @kvt=@aar*10+4
if @maaned=11 set @kvt=@aar*10+4
if @maaned=12 set @kvt=@aar*10+4
return @kvt
end
Go

```

```

/* laver TidsDim kalender tabel */

```

```

if exists (select * from sys.sysobjects where id = object_id(N'[TidsDimTest]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)
drop table [TidsDimTest]
GO

```

```

create table TidsDimTest
(
    dato datetime not null primary key
    ,Intår int not null -- surrogatkey as integer
    ,år int not null -- Årstal
    ,måned int not null -- Månednummer 1-12
    ,Årmåned int not null -- År+måned til hierarkibrug
    ,kvartal int not null -- kvartalsnummer 1-4
    ,Årkvartal int not null -- År+kvartal til hierarkibrug
    ,månednavn varchar(9) not null -- Januar-December
    ,månedkort varchar(3) not null -- Jan-Dec

```

```

,uge int not null -- ugenummer 1-53
,MdrUge varchar (5) not null -- stemplet med mdr+uge
,AarUge varchar (7) not null -- stemplet med år+uge (uge 1 næste år)
,dag int not null -- månedsdagsnummer 1-31
,ugedag varchar(7) not null -- mandag til søndag
,ugedagkort varchar(3) not null -- man, tir, ons, tor, fre, lør, søn
,ugedagnr tinyint not null -- 1 til 7
,weekend tinyint not null -- 1 hvis lørdag eller søndag
,man_fre tinyint not null -- 1 hvis mandag-fredag
,arbejdsdag tinyint not null -- 1 hvis ikke helligdag eller week-end
,helligdag tinyint not null -- 1 hvis helligdag
,HelligWeekend tinyint not null -- 1 hvis helligdag eller week-end
,helligdagsnavn varchar(17) not null

```

```
)
```

```
-- find 1. dag
```

```
Declare @x as int
```

```
Set @x=@@DATEFIRST
```

```
-- find mandag
```

```
Declare @m as int
```

```
Set @m=(select case when @x=7 then 2 else 1 end)
```

```
-- find mandag
```

```
Declare @s as int
```

```
Set @s=(select case when @x=7 then 1 else 7 end)
```

```
declare @firstdate datetime
```

```
-----
```

```
-- Følgende indsætter nu records i den definerede tabel --
```

```
-- Husk at angive samme navn for tabellen nedenfor --
```

-- BEMÆRK: NEDENSTÅENDE 2 DATOER BESTEMMER HVILKE DATOER --

-- DER INDSÆTTES I KALENDEREN --

-- Det er en INSERT INTO loop mellem startdato og slut --

-- dato som defineres i de første to linjer nedenfor. --

set @firstdate = '1990-01-01'

while @firstdate < '2049-01-01' --dateadd(yyyy,1,getdate())

begin

insert into TidsDimTest (dato,intår, år,måned,Årmåned, kvartal,Årkvartal,
månednavn,månedkort,uge,Mdruge,Aaruge,dag,ugedag,ugedagkort,ugedagnr,weekend,man_fre,arbejds
dag,helligdag,helligweekend,helligdagsnavn)

select @firstdate --floor(convert(decimal(38,15),@firstdate))
,dbo.intdato(year(@firstdate),month(@firstdate),day(@firstdate))

,YEAR(@firstdate)

,MONTH(@firstdate)

,YEAR(@firstdate)*100+MONTH(@firstdate)

,CASE MONTH(@firstdate)

WHEN 1 THEN 1

WHEN 2 THEN 1

WHEN 3 THEN 1

WHEN 4 THEN 2

WHEN 5 THEN 2

WHEN 6 THEN 2

WHEN 7 THEN 3

WHEN 8 THEN 3

WHEN 9 THEN 3

```

WHEN 10 THEN 4

WHEN 11 THEN 4

WHEN 12 THEN 4

END

,dbo.Aarkvartal(Year(@firstdate),month(@firstdate))

,CASE MONTH(@firstdate)

WHEN 1 THEN 'Januar'

WHEN 2 THEN 'Februar'

WHEN 3 THEN 'Marts'

WHEN 4 THEN 'April'

WHEN 5 THEN 'Maj'

WHEN 6 THEN 'Juni'

WHEN 7 THEN 'Juli'

WHEN 8 THEN 'August'

WHEN 9 THEN 'September'

WHEN 10 THEN 'Oktober'

WHEN 11 THEN 'November'

WHEN 12 THEN 'December'

END

,CASE MONTH(@firstdate)

WHEN 1 THEN 'Jan'

WHEN 2 THEN 'Feb'

WHEN 3 THEN 'Mar'

WHEN 4 THEN 'Apr'

WHEN 5 THEN 'Maj'

WHEN 6 THEN 'Jun'

WHEN 7 THEN 'Jul'

WHEN 8 THEN 'Aug'

WHEN 9 THEN 'Sep'

```

```

WHEN 10 THEN 'Okt'

WHEN 11 THEN 'Nov'

WHEN 12 THEN 'Dec'

END

,dbo.ugenummer(year(@firstdate),month(@firstdate),day(@firstdate)) --DATEPART(WEEK,@firstdate)

,dbo.mdruge(year(@firstdate),month(@firstdate),day(@firstdate)) --DATEPART(WEEK,@firstdate)

,dbo.aaruge(year(@firstdate),month(@firstdate),day(@firstdate)) --DATEPART(WEEK,@firstdate)

,DAY(@firstdate)

,CASE

WHEN DATEPART(WEEKDAY,@firstdate)=@m THEN 'Mandag'

WHEN DATEPART(WEEKDAY,@firstdate)=@m+1 THEN 'Tirsdag'

WHEN DATEPART(WEEKDAY,@firstdate)=@m+2 THEN 'Onsdag'

WHEN DATEPART(WEEKDAY,@firstdate)=@m+3 THEN 'Torsdag'

WHEN DATEPART(WEEKDAY,@firstdate)=@m+4 THEN 'Fredag'

WHEN DATEPART(WEEKDAY,@firstdate)=@m+5 THEN 'Lørdag'

WHEN DATEPART(WEEKDAY,@firstdate)=@s THEN 'Søndag'

END

,CASE

WHEN DATEPART(WEEKDAY,@firstdate)=@m THEN 'Man'

WHEN DATEPART(WEEKDAY,@firstdate)=@m+1 THEN 'Tir'

WHEN DATEPART(WEEKDAY,@firstdate)=@m+2 THEN 'Ons'

WHEN DATEPART(WEEKDAY,@firstdate)=@m+3 THEN 'Tor'

WHEN DATEPART(WEEKDAY,@firstdate)=@m+4 THEN 'Fre'

WHEN DATEPART(WEEKDAY,@firstdate)=@m+5 THEN 'Lør'

WHEN DATEPART(WEEKDAY,@firstdate)=@s THEN 'Søn'

END

,CASE

WHEN DATEPART(WEEKDAY,@firstdate)=@m THEN 1

WHEN DATEPART(WEEKDAY,@firstdate)=@m+1 THEN 2

```

```
WHEN DATEPART(WEEKDAY,@firstdate)=@m+2 THEN 3
WHEN DATEPART(WEEKDAY,@firstdate)=@m+3 THEN 4
WHEN DATEPART(WEEKDAY,@firstdate)=@m+4 THEN 5
WHEN DATEPART(WEEKDAY,@firstdate)=@m+5 THEN 6
WHEN DATEPART(WEEKDAY,@firstdate)=@s THEN 7
END
,dbo.isWeekend(@firstdate)
,dbo.ismanfre(@firstdate)
,dbo.isWorkDay(@firstdate)
,dbo.IsDanishHolyDay(@firstdate)
,dbo.IsHolyWeekend(@firstdate)
,dbo.nameofdanishholyday(@firstdate)

set @firstdate = dateadd(dd,1,@firstdate)
end
```