

CREATE A CUSTOM DATE TABLE

```
--Make sure you set the Start and End Date below on row 58 and 59
--Create the tables
BEGIN TRY
    DROP TABLE [DimDate]
END TRY
BEGIN CATCH
    --DO NOTHING
END CATCH
CREATE TABLE [dbo].[DimDate](
    --[DateSK] [int] IDENTITY(1,1) NOT NULL--Use this line if you just want an autoincrementing counter AND
    COMMENT BELOW LINE
    [DateSK] [int] NOT NULL--TO MAKE THE DateSK THE YYYYMMDD FORMAT USE THIS LINE AND COMMENT ABOVE LINE.
    , [Date] [datetime] NOT NULL
    , [Day] [tinyint] NOT NULL
    , [DaySuffix] [varchar](4) NOT NULL
    , [DayOfWeek] [varchar](9) NOT NULL
    , [DOWInMonth] [TINYINT] NOT NULL
    , [DayOfYear] [int] NOT NULL
    , [WeekOfYear] [tinyint] NOT NULL
    , [WeekOfMonth] [tinyint] NOT NULL
    , [Month] [tinyint] NOT NULL
    , [MonthName] [varchar](9) NOT NULL
    , [Quarter] [tinyint] NOT NULL
    , [QuarterName] [varchar](6) NOT NULL
    , [Year] [char](4) NOT NULL
    , [StandardDate] [varchar](10) NULL
    , [HolidayText] [varchar](50) NULL
    CONSTRAINT [PK_DimDate] PRIMARY KEY CLUSTERED
    (
        [DateSK] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
    ) ON [PRIMARY]
```

GO

--Populate Date dimension

TRUNCATE TABLE DimDate

--IF YOU ARE USING THE YYYYMMDD format for the primary key then you need to comment out this line.

--DBCC CHECKIDENT (DimDate, RESEED, 60000) --In case you need to add earlier dates later.

DECLARE @tmpDOW TABLE (DOW INT, Cntr INT)--Table for counting DOW occurrence in a month

INSERT INTO @tmpDOW(DOW, Cntr) VALUES(1,0)--Used in the loop below

INSERT INTO @tmpDOW(DOW, Cntr) VALUES(2,0)

INSERT INTO @tmpDOW(DOW, Cntr) VALUES(3,0)

INSERT INTO @tmpDOW(DOW, Cntr) VALUES(4,0)

INSERT INTO @tmpDOW(DOW, Cntr) VALUES(5,0)

INSERT INTO @tmpDOW(DOW, Cntr) VALUES(6,0)

INSERT INTO @tmpDOW(DOW, Cntr) VALUES(7,0)

DECLARE @StartDate datetime

, @EndDate datetime

, @Date datetime

, @WDofMonth INT

, @CurrentMonth INT

SELECT @StartDate = '1/1/2000' -- Set The start and end date

, @EndDate = '1/1/2015'--Non inclusive. Stops on the day before this.

, @CurrentMonth = 1 --Counter used in loop below.

SELECT @Date = @StartDate

WHILE @Date < @EndDate

BEGIN

IF DATEPART(MONTH,@Date) <> @CurrentMonth

BEGIN

SELECT @CurrentMonth = DATEPART(MONTH,@Date)

UPDATE @tmpDOW SET Cntr = 0

END

```

UPDATE @tmpDOW
SET Cntr = Cntr + 1
WHERE DOW = DATEPART(DW,@DATE)

```

```

SELECT @WDofMonth = Cntr
FROM @tmpDOW
WHERE DOW = DATEPART(DW,@DATE)

```

```

INSERT INTO DimDate

```

```

(
[DateSK],--TO MAKE THE DateSK THE YYYYMMDD FORMAT UNCOMMENT THIS LINE... Comment for autoincrementing.
[Date]
, [Day]
, [DaySuffix]
, [DayOfWeek]
, [DOWInMonth]
, [DayOfYear]
, [WeekOfYear]
, [WeekOfMonth]
, [Month]
, [MonthName]
, [Quarter]
, [QuarterName]
, [Year]
, [StandardDate]
)

```

```

SELECT CONVERT(VARCHAR,@Date,112), --TO MAKE THE DateSK THE YYYYMMDD FORMAT UNCOMMENT THIS LINE COMMENT FOR
AUTOINCREMENT

```

```

@Date [Date]
, DATEPART(DAY,@DATE) [Day]
, CASE
WHEN DATEPART(DAY,@DATE) IN (11,12,13) THEN CAST(DATEPART(DAY,@DATE) AS VARCHAR) + 'th'
WHEN RIGHT(DATEPART(DAY,@DATE),1) = 1 THEN CAST(DATEPART(DAY,@DATE) AS VARCHAR) + 'st'
WHEN RIGHT(DATEPART(DAY,@DATE),1) = 2 THEN CAST(DATEPART(DAY,@DATE) AS VARCHAR) + 'nd'
WHEN RIGHT(DATEPART(DAY,@DATE),1) = 3 THEN CAST(DATEPART(DAY,@DATE) AS VARCHAR) + 'rd'
ELSE CAST(DATEPART(DAY,@DATE) AS VARCHAR) + 'th'
END AS [DaySuffix]

```

```

, CASE DATEPART(DW, @DATE)
WHEN 1 THEN 'Sunday'
WHEN 2 THEN 'Monday'
WHEN 3 THEN 'Tuesday'
WHEN 4 THEN 'Wednesday'
WHEN 5 THEN 'Thursday'
WHEN 6 THEN 'Friday'
WHEN 7 THEN 'Saturday'
END AS [DayOfWeek]
, @WDofMonth [DOWInMonth]--Occurance of this day in this month. If Third Monday then 3 and DOW would be
Monday.
, DATEPART(dy,@Date) [DayOfYear]--Day of the year. 0 - 365/366
, DATEPART(ww,@Date) [WeekOfYear]--0-52/53
, DATEPART(ww,@Date) + 1 -
DATEPART(ww,CAST(DATEPART(mm,@Date) AS VARCHAR) + '/1/' + CAST(DATEPART(yy,@Date) AS VARCHAR)) [WeekOfMonth]
, DATEPART(MONTH,@DATE) [Month]--To be converted with leading zero later.
, DATENAME(MONTH,@DATE) [MonthName]
, DATEPART(qq,@DATE) [Quarter]--Calendar quarter
, CASE DATEPART(qq,@DATE)
WHEN 1 THEN 'First'
WHEN 2 THEN 'Second'
WHEN 3 THEN 'Third'
WHEN 4 THEN 'Fourth'
END AS [QuarterName]
, DATEPART(YEAR,@Date) [Year]
, Right('0' + convert(varchar(2),MONTH(@Date)),2) + '/' + Right('0' + convert(varchar(2),DAY(@Date)),2) + '/'
+ convert(varchar(4),YEAR(@Date))

SELECT @Date = DATEADD(dd,1,@Date)
END

```

```
--Add HOLIDAYS -----  
-----  
--THANKSGIVING -----  
-----  
--Fourth THURSDAY in November.  
UPDATE DimDate  
SET HolidayText = 'Thanksgiving Day'  
WHERE [MONTH] = 11  
      AND [DAYOFWEEK] = 'Thursday'  
      AND [DOWInMonth] = 4  
GO  
  
--CHRISTMAS -----  
UPDATE dbo.DimDate  
SET HolidayText = 'Christmas Day'  
WHERE [MONTH] = 12 AND [DAY] = 25  
  
--4th of July -----  
UPDATE dbo.DimDate  
SET HolidayText = 'Independance Day'  
WHERE [MONTH] = 7 AND [DAY] = 4  
  
-- New Years Day -----  
-  
UPDATE dbo.DimDate  
SET HolidayText = 'New Year''s Day'  
WHERE [MONTH] = 1 AND [DAY] = 1
```

```
--Memorial Day -----
--Last Monday in May
UPDATE dbo.DimDate
SET HolidayText = 'Memorial Day'
FROM DimDate
WHERE DateSK IN
(
    SELECT MAX([DateSK])
    FROM dbo.DimDate
    WHERE [MonthName] = 'May'
    AND [DayOfWeek] = 'Monday'
    GROUP BY [YEAR], [MONTH]
)
```

```
--Labor Day -----
--First Monday in September
UPDATE dbo.DimDate
SET HolidayText = 'Labor Day'
FROM DimDate
WHERE DateSK IN
(
    SELECT MIN([DateSK])
    FROM dbo.DimDate
    WHERE [MonthName] = 'September'
    AND [DayOfWeek] = 'Monday'
    GROUP BY [YEAR], [MONTH]
)
```

```
-- Valentine's Day -----
---
UPDATE dbo.DimDate
SET HolidayText = 'Valentine''s Day'
WHERE [MONTH] = 2 AND [DAY] = 14
```

```
-- Saint Patrick's Day -----
---
UPDATE dbo.DimDate
SET HolidayText = 'Saint Patrick''s Day'
WHERE [MONTH] = 3 AND [DAY] = 17
```

GO

--Martin Luthor King Day -----

--Third Monday in January starting in 1983

UPDATE DimDate

SET HolidayText = 'Martin Luthor King Jr Day'

WHERE [MONTH] = 1--January

AND [Dayofweek] = 'Monday'

AND [YEAR] >= 1983--When holiday was official

AND [DOWInMonth] = 3--Third X day of current month.

GO

--President's Day -----

--Third Monday in February.

UPDATE DimDate

SET HolidayText = 'President's Day'--select * from DimDate

WHERE [MONTH] = 2--February

AND [Dayofweek] = 'Monday'

AND [DOWInMonth] = 3--Third occurrence of a monday in this month.

GO

--Mother's Day -----

--Second Sunday of May

UPDATE DimDate

SET HolidayText = 'Mother's Day'--select * from DimDate

WHERE [MONTH] = 5--May

AND [Dayofweek] = 'Sunday'

AND [DOWInMonth] = 2--Second occurrence of a monday in this month.

GO

--Father's Day -----

--Third Sunday of June

UPDATE DimDate

SET HolidayText = 'Father's Day'--select * from DimDate

WHERE [MONTH] = 6--June

AND [Dayofweek] = 'Sunday'

AND [DOWInMonth] = 3--Third occurrence of a monday in this month.

GO

```

--Halloween 10/31 -----
UPDATE dbo.DimDate
SET HolidayText = 'Halloween'
WHERE [MONTH] = 10 AND [DAY] = 31
--Election Day-----
--The first Tuesday after the first Monday in November.
BEGIN TRY
    drop table #tmpHoliday
END TRY
BEGIN CATCH
    --do nothing
END CATCH

CREATE TABLE #tmpHoliday(ID INT IDENTITY(1,1), DateID int, Week TINYINT, YEAR CHAR(4), DAY CHAR(2))

INSERT INTO #tmpHoliday(DateID, [YEAR],[DAY])
SELECT [DateSK], [YEAR], [DAY]
FROM dbo.DimDate
WHERE [MONTH] = 11
AND [Dayofweek] = 'Monday'
ORDER BY YEAR, DAY

DECLARE @CNTR INT, @POS INT, @STARTYEAR INT, @ENDYEAR INT, @CURRENTYEAR INT, @MINDAY INT

SELECT @CURRENTYEAR = MIN([YEAR])
    , @STARTYEAR = MIN([YEAR])
    , @ENDYEAR = MAX([YEAR])
FROM #tmpHoliday

WHILE @CURRENTYEAR <= @ENDYEAR
BEGIN
    SELECT @CNTR = COUNT([YEAR])
    FROM #tmpHoliday
    WHERE [YEAR] = @CURRENTYEAR

    SET @POS = 1

    WHILE @POS <= @CNTR

```



```
BEGIN
SELECT @MINDAY = MIN(DAY)
FROM #tmpHoliday
WHERE [YEAR] = @CURRENTYEAR
AND [WEEK] IS NULL
```

```
UPDATE #tmpHoliday
SET [WEEK] = @POS
WHERE [YEAR] = @CURRENTYEAR
AND [DAY] = @MINDAY
```

```
SELECT @POS = @POS + 1
END
```

```
SELECT @CURRENTYEAR = @CURRENTYEAR + 1
END
```

```
UPDATE DT
SET HolidayText = 'Election Day'
FROM dbo.DimDate DT
JOIN #tmpHoliday HL
ON (HL.DateID + 1) = DT.DateSK
WHERE [WEEK] = 1
```

```
DROP TABLE #tmpHoliday
GO
```

```
-----
PRINT CONVERT(VARCHAR,GETDATE(),113)--USED FOR CHECKING RUN TIME.
```

```
--DimDate indexes-----
-
CREATE UNIQUE NONCLUSTERED INDEX [IDX_DimDate_Date] ON [dbo].[DimDate]
(
[Date] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
```

```
CREATE NONCLUSTERED INDEX [IDX_DimDate_Day] ON [dbo].[DimDate]
(
[Day] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
```

```
CREATE NONCLUSTERED INDEX [IDX_DimDate_DayOfWeek] ON [dbo].[DimDate]
(
[DayOfWeek] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
```

```
CREATE NONCLUSTERED INDEX [IDX_DimDate_DOWInMonth] ON [dbo].[DimDate]
(
[DOWInMonth] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
```

```
CREATE NONCLUSTERED INDEX [IDX_DimDate_DayOfYear] ON [dbo].[DimDate]
(
[DayOfYear] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
```

```
CREATE NONCLUSTERED INDEX [IDX_DimDate_WeekOfYear] ON [dbo].[DimDate]
(
[WeekOfYear] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
```

```
CREATE NONCLUSTERED INDEX [IDX_DimDate_WeekOfMonth] ON [dbo].[DimDate]
(
[WeekOfMonth] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
```

```

CREATE NONCLUSTERED INDEX [IDX_DimDate_Month] ON [dbo].[DimDate]
(
    [Month] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]

CREATE NONCLUSTERED INDEX [IDX_DimDate_MonthName] ON [dbo].[DimDate]
(
    [MonthName] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]

CREATE NONCLUSTERED INDEX [IDX_DimDate_Quarter] ON [dbo].[DimDate]
(
    [Quarter] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]

CREATE NONCLUSTERED INDEX [IDX_DimDate_QuarterName] ON [dbo].[DimDate]
(
    [QuarterName] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]

CREATE NONCLUSTERED INDEX [IDX_DimDate_Year] ON [dbo].[DimDate]
(
    [Year] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]

CREATE NONCLUSTERED INDEX [IDX_dim_Time_HolidayText] ON [dbo].[DimDate]
(
    [HolidayText] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]

PRINT convert(varchar,getdate(),113)--USED FOR CHECKING RUN TIME.

```

```

--This script will add Fiscal columns to the DimDate Table
--This script will set the Start date and end below to the first and last date on your dim date table
--The following COLUMNS will be created: Fiscal Year, FiscalQuarter, FiscalQuarterName, FiscalMonth

--SET THE @DaysOffset VARIABLE TO OFFSET THE DATE
Alter Table DimDate Add FiscalMonth int
Alter Table DimDate Add FiscalQuarter int
Alter Table DimDate Add FiscalQuarterName Varchar(6)
Alter Table DimDate Add FiscalYear char(10)
Go

Declare @MonthOffset int
--If your Fiscal Year starts on October of the previous year then set this variable to 3 -----
-----
--If your Fiscal Year starts after the start of the calendar year set it to a negative number of months-----
-----
Select @MonthOffset = 3

--Declare all of the needed variables
DECLARE @StartDate datetime
, @EndDate datetime
, @Date datetime
, @MonthNumber int
, @QuarterName varchar(6)
, @QuarterNumber int
, @FirstFiscalDate date
, @FiscalYear char(10)

--Get first and last date
SELECT @StartDate = (Select Min([Date]) from [DimDate])
Select @EndDate = (Select Max([Date]) from [DimDate])
SELECT @Date = @StartDate

--set the first date of the fiscal year
SELECT @FirstFiscalDate = DATEADD(Month, -1*@MonthOffset, @Date)

```

```

--Loop through each date
WHILE @Date <= @EndDate
BEGIN
    --Set the number of months off set
    Select @MonthNumber = Month(@Date) + @MonthOffSet
    Select @MonthNumber =
    Case When @MonthNumber > 12 then @MonthNumber - 12
        When @MonthNumber < 1 then @MonthNumber + 12
        Else @MonthNumber End

    -- Set the Quarter off set
    Select @QuarterNumber =
    Case When @MonthNumber = 1 or @MonthNumber = 2 or @MonthNumber =3 Then 1
    When @MonthNumber = 4 or @MonthNumber = 5 or @MonthNumber = 6 Then 2
    When @MonthNumber = 7 or @MonthNumber = 8 or @MonthNumber = 9 Then 3
    Else 4 End

    Select @QuarterName =
    Case @QuarterNumber
        When 1
            Then 'First'
        When 2
            Then 'Second'
        When 3
            Then 'Third'
        When 4
            Then 'Forth'
        Else 'Error'
    End

    --Determine the fiscal year
    Select @FiscalYear =
    Case When MONTH(@date) < MONTH(@FirstFiscalDate) Then
        convert(varchar(2),right((DATEPART(YEAR,@Date)-1),2)) + '/' +
        convert(varchar(2),right(DATEPART(YEAR,@Date),2))
        Else convert(varchar(2),right((DATEPART(YEAR,@Date)),2)) + '/' +
        convert(varchar(2),right(DATEPART(YEAR,@Date)+1,2))
    End

```

--Update the table with the fical numbers

Update DimDate

Set

FiscalMonth = @MonthNumber,

FiscalQuarter = @QuarterNumber,

FiscalQuarterName = @QuarterName,

FiscalYear = 'FY ' + @FiscalYear

Where Date = @Date

--Increment the date by one day

SELECT @Date = DATEADD(dd,1,@Date)

END

PRINT CONVERT(VARCHAR,GETDATE(),113)--USED FOR CHECKING RUN TIME.

--DimDate indexes-----
-

CREATE NONCLUSTERED INDEX [IDX_DimDate_FiscalMonth] ON [dbo].[DimDate]

(

[FiscalMonth] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,

DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]

CREATE NONCLUSTERED INDEX [IDX_DimDate_FiscalMonthName] ON [dbo].[DimDate]

(

[MonthName] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,

DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]

CREATE NONCLUSTERED INDEX [IDX_DimDate_FiscalQuarter] ON [dbo].[DimDate]

(

[FiscalQuarter] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,

DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]

```
CREATE NONCLUSTERED INDEX [IDX_DimDate_FiscalQuarterName] ON [dbo].[DimDate]
(
[FiscalQuarterName] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
```

```
CREATE NONCLUSTERED INDEX [IDX_DimDate_FiscalYear] ON [dbo].[DimDate]
(
[FiscalYear] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
```

```
PRINT convert(varchar,getdate(),113)--USED FOR CHECKING RUN TIME.
```

```
--Create DimTime Script
```

```
SET ANSI_PADDING OFF
```

```
BEGIN TRY
```

```
    DROP TABLE [DimTime]
```

```
END TRY
```

```
BEGIN CATCH
```

```
    --DO NOTHING
```

```
END CATCH
```

```
CREATE TABLE [dbo].[DimTime](
[TimeSK] [int] IDENTITY(1,1) NOT NULL,
[Time] [char](8) NOT NULL,
[Hour] [char](2) NOT NULL,
[MilitaryHour] [char](2) NOT NULL,
[Minute] [char](2) NOT NULL,
[Second] [char](2) NOT NULL,
[AmPm] [char](2) NOT NULL,
[StandardTime] [char](11) NULL,
CONSTRAINT [PK_DimTime] PRIMARY KEY CLUSTERED
(
[TimeSK] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```

GO
SET ANSI_PADDING OFF

PRINT CONVERT(VARCHAR,GETDATE(),113)--USED FOR CHECKING RUN TIME.

--Load time data for every second of a day
DECLARE @Time DATETIME

SET @TIME = CONVERT(VARCHAR,'12:00:00 AM',108)

TRUNCATE TABLE DimTime

WHILE @TIME <= '11:59:59 PM'
BEGIN
    INSERT INTO dbo.DimTime([Time], [Hour], [MilitaryHour], [Minute], [Second], [AmPm])
    SELECT CONVERT(VARCHAR,@TIME,108) [Time]
    , CASE
    WHEN DATEPART(HOUR,@Time) > 12 THEN DATEPART(HOUR,@Time) - 12
    ELSE DATEPART(HOUR,@Time)
    END AS [Hour]
    , CAST(SUBSTRING(CONVERT(VARCHAR,@TIME,108),1,2) AS INT) [MilitaryHour]
    , DATEPART(MINUTE,@Time) [Minute]
    , DATEPART(SECOND,@Time) [Second]
    , CASE
    WHEN DATEPART(HOUR,@Time) >= 12 THEN 'PM'
    ELSE 'AM'
    END AS [AmPm]

    SELECT @TIME = DATEADD(second,1,@Time)
END

UPDATE DimTime
SET [HOUR] = '0' + [HOUR]
WHERE LEN([HOUR]) = 1

```



```
UPDATE DimTime
SET [MINUTE] = '0' + [MINUTE]
WHERE LEN([MINUTE]) = 1
```

```
UPDATE DimTime
SET [SECOND] = '0' + [SECOND]
WHERE LEN([SECOND]) = 1
```

```
UPDATE DimTime
SET [MilitaryHour] = '0' + [MilitaryHour]
WHERE LEN([MilitaryHour]) = 1
```

```
UPDATE DimTime
SET StandardTime = [Hour] + ':' + [Minute] + ':' + [Second] + ' ' + AmPm
WHERE StandardTime is null
AND HOUR <> '00'
```

```
UPDATE DimTime
SET StandardTime = '12' + ':' + [Minute] + ':' + [Second] + ' ' + AmPm
WHERE [HOUR] = '00'
```

--DimTime indexes

```
CREATE UNIQUE NONCLUSTERED INDEX [IDX_DimTime_Time] ON [dbo].[DimTime]
(
    [Time] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
```

```
CREATE NONCLUSTERED INDEX [IDX_DimTime_Hour] ON [dbo].[DimTime]
(
    [Hour] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
```

```
CREATE NONCLUSTERED INDEX [IDX_DimTime_MilitaryHour] ON [dbo].[DimTime]
(
[MilitaryHour] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]

CREATE NONCLUSTERED INDEX [IDX_DimTime_Minute] ON [dbo].[DimTime]
(
[Minute] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]

CREATE NONCLUSTERED INDEX [IDX_DimTime_Second] ON [dbo].[DimTime]
(
[Second] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]

CREATE NONCLUSTERED INDEX [IDX_DimTime_AmPm] ON [dbo].[DimTime]
(
[AmPm] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]

CREATE NONCLUSTERED INDEX [IDX_DimTime_StandardTime] ON [dbo].[DimTime]
(
[StandardTime] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]

PRINT convert(varchar,getdate(),113)--USED FOR CHECKING RUN TIME.
```