

Create an Extended Date Dimension for a SQL Server Data Warehouse

By: [Dinesh Asanka \(/sqlserverauthor/288/dinesh-asanka/\)](/sqlserverauthor/288/dinesh-asanka/) | Updated: 2018-07-31 | [Comments \(3\)](#) | Related: [More \(/sql-server-developer-resources/\)](#) > [Dates \(/sql-server-tip-category/121/dates/\)](#)

Problem

The Date Dimension is a key dimension in a SQL Server data warehousing as it allows us to analyze data in different aspects of date. Apart from the standard date attributes like year, quarter, month, etc., this article explains how the date dimension can be extended to richer analysis in a SQL Server data warehouse.

Solution

A date dimension is mostly a static dimension which does not require daily update. However, a window function may need a daily update. The solution would be to have a special date dimension and populate at the start of the year and update changing data on a daily basis.

Why a Special Dimension

First question would be, what is the requirement for the special dimension for a date in the data warehouse. Let's look at the options we have in case a date dimension is not present.

What are sales per year?

```
SELECT YEAR(OrderDate) Year ,SUM(SalesAmount) Amount
FROM FCT_Sales
GROUP BY YEAR(OrderDate)
```

What are the sales done on weekends?

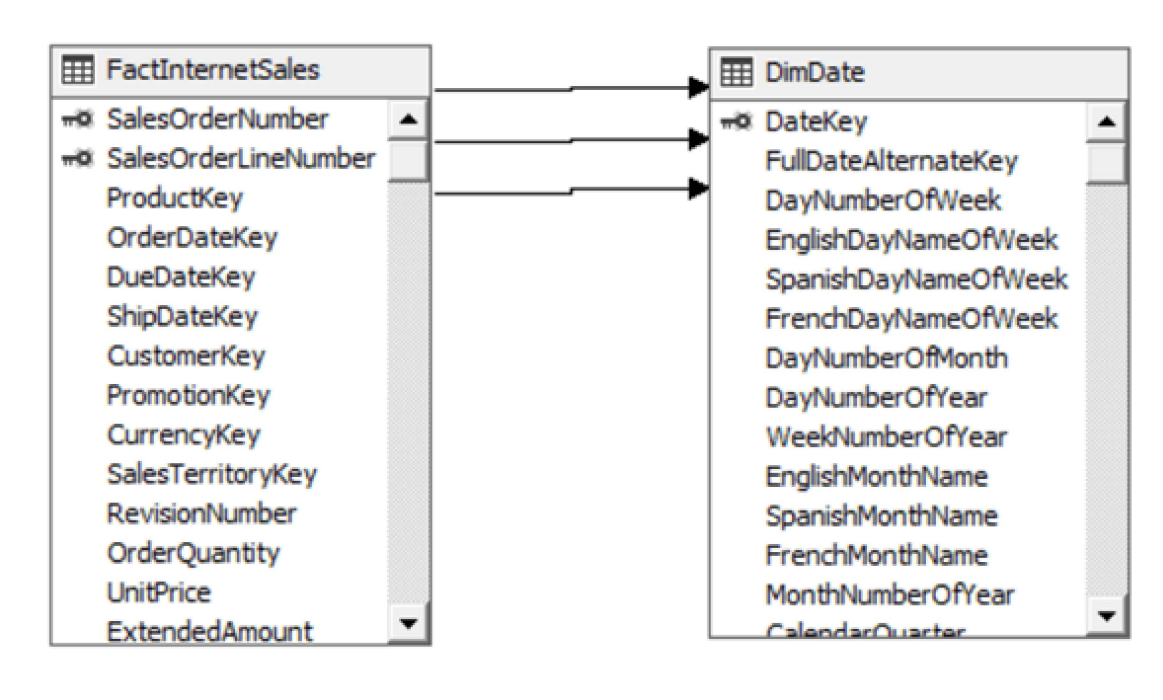
```
SELECT SUM(SalesAmount)
FROM FCT_Sales
WHERE DATEPART(dw,OrderDate) IN (1,7)
```

In both the scenarios, there will be a performance impact. As you are aware, we are dealing with a large number of records in a data warehouse, the above queries will have performance issues. Also, indexes won't be a solution as the use of functions will not make the index usable.

Apart from performance issues, there are functional limitations. For example, in case you need to get the sales on a special holiday or for a season, in which there are no built-in functions, you have no choice, but to have a special date dimension.

By looking at functional and performance limitations, it is very obvious that there needs to be a special dimension to store a date which is the date dimension used more commonly in the data warehouse.

Also, there are times where more than one date column is available in the fact table. In that instance, a date dimension will act as a role playing dimension in SQL Server Analysis Services as shown in the below figure.



In the above example, OrderDateKey, DueDateKey and ShipDateKey are linked to the Date Dimension.

Please note that the role playing dimension feature is not available in the Tabular world where you need add multiple instances of the date dimension.

Surrogate Key

Typically, surrogate keys will be an incremental number. However, in case of a date dimension, YYYYMMDD format is used for a surrogate key. This is to facilitate data partitioning in the data warehouse. Fact tables are normally partition by the date. If mere incremental numbers are used for the date dimension, the fact table will also have the same incremental numbers which will lead to difficulties in partitioning. In case of YYYYMMDD format, it is much easier to include a partitioning function using the details in the surrogate key.

Standard Columns

In a date dimension, it is always better to include all the possible columns leaving the options of deriving attributes at the user level.

Below is an example of the basic columns for a Date Dimension.

```

CREATE TABLE dbo.Dim_Date (
    DateKey INT NOT NULL PRIMARY KEY,
    [Date] DATE NOT NULL,
    [Day] TINYINT NOT NULL,
    [DaySuffix] CHAR(2) NOT NULL,
    [Weekday] TINYINT NOT NULL,
    [WeekDayName] VARCHAR(10) NOT NULL,
    [WeekDayName_Short] CHAR(3) NOT NULL,
    [WeekDayName_FirstLetter] CHAR(1) NOT NULL,
    [DOWInMonth] TINYINT NOT NULL,
    [DayOfYear] SMALLINT NOT NULL,
    [WeekOfMonth] TINYINT NOT NULL,
    [WeekOfYear] TINYINT NOT NULL,
    [Month] TINYINT NOT NULL,
    [MonthName] VARCHAR(10) NOT NULL,
    [MonthName_Short] CHAR(3) NOT NULL,
    [MonthName_FirstLetter] CHAR(1) NOT NULL,
    [Quarter] TINYINT NOT NULL,
    [QuarterName] VARCHAR(6) NOT NULL,
    [Year] INT NOT NULL,
    [MMYYYY] CHAR(6) NOT NULL,
    [MonthYear] CHAR(7) NOT NULL,
    IsWeekend BIT NOT NULL,
)

```

Important to note is the existence of three columns for month name. MonthName is used to store the month name such as January, February, etc. In some reports, you might have experienced that the month name will be shorten such as Jan, Feb, etc. which can be stored in MonthName_Short column. MonthName_FirstLetter column can be used to store J, F, M, etc. for the first character of the month giving more options for users. Similarly, there are three columns for Weekday as well.

Most of these attributes can be generated by using built-in SQL Server functions such as YEAR, MONTH, DATEPART and DATENAME. In this script, EndDate can be defined.

```
SET NOCOUNT ON
```

```
TRUNCATE TABLE DIM_Date
```

```
DECLARE @CurrentDate DATE = '2016-01-01'
```

```
DECLARE @EndDate DATE = '2020-12-31'
```

```
WHILE @CurrentDate < @EndDate
```

```
BEGIN
```

```
    INSERT INTO [dbo].[Dim_Date] (
```

```
        [DateKey],
```

```
        [Date],
```

```
        [Day],
```

```
        [DaySuffix],
```

```
        [Weekday],
```

```
        [WeekDayName],
```

```
        [WeekDayName_Short],
```

```
        [WeekDayName_FirstLetter],
```

```
        [DOWInMonth],
```

```
        [DayOfYear],
```

```
        [WeekOfMonth],
```

```
        [WeekOfYear],
```

```
        [Month],
```

```
        [MonthName],
```

```
        [MonthName_Short],
```

```
        [MonthName_FirstLetter],
```

```
        [Quarter],
```

Holidays and Special Days

Holidays will be handled by the following columns.

```
IsHoliday    BIT NOT NULL,  
HolidayName  VARCHAR(20) NULL,  
SpecialDays  VARCHAR(20) NULL
```

As holidays are dependent on the country or region that you are implementing the data warehouse, a customized script is needed for the holidays and special days.

The following script shows how Christmas and Valentine's day are updated in the Dim_date dimension table.

```
UPDATE Dim_Date
```

```
SET [IsHoliday] = 1,
```

```
[HolidayName] = 'Christmas'
```

```
Where [Month] = 12 AND [DAY] = 25
```

```
UPDATE Dim_Date
```

```
SET SpecialDays = 'Valentines Day'
```

```
Where [Month] = 2 AND [DAY] = 14
```

First and Last Days

In financial scenarios, last and first dates for different segments such as year, quarter, month and week will be used. The following columns are used to store those dates.

```

FirstDateofYear    DATE NULL,
LastDateofYear     DATE NULL,
FirstDateofQuater  DATE NULL,
LastDateofQuater   DATE NULL,
FirstDateofMonth   DATE NULL,
LastDateofMonth    DATE NULL,
FirstDateofWeek    DATE NULL,
LastDateofWeek     DATE NULL,

```

The following script will be used to update those columns by using DATEADD, DATEDIFF and EOMONTH functions.

```

[FirstDateofYear]  = CAST(CAST(YEAR(@CurrentDate) AS VARCHAR(4)) + '-01-01' AS DATE),
[LastDateofYear]   = CAST(CAST(YEAR(@CurrentDate) AS VARCHAR(4)) + '-12-31' AS DATE),
[FirstDateofQuater] = DATEADD(qq, DATEDIFF(qq, 0, GETDATE()), 0),
[LastDateofQuater] = DATEADD(dd, - 1, DATEADD(qq, DATEDIFF(qq, 0, GETDATE()) + 1, 0)),
[FirstDateofMonth] = CAST(CAST(YEAR(@CurrentDate) AS VARCHAR(4)) + '-' + CAST(MONTH(@CurrentDate) AS VARCHAR(2)) +
[LastDateofMonth]  = EOMONTH(@CurrentDate),
[FirstDateofWeek]  = DATEADD(dd, - (DATEPART(dw, @CurrentDate) - 1), @CurrentDate),
[LastDateofWeek]   = DATEADD(dd, 7 - (DATEPART(dw, @CurrentDate)), @CurrentDate)

```

Multiple Calendars

As you are aware, there can be multiple calendars such as having a specific Financial calendar. Some companies start their Financial year on April 1st while some company's financial year starts in October. Depending on this, the Financial Account Year, Financial Account Quarter, Financial Account Month will change accordingly.

Window Functions

In a data warehouse, there is analysis for current and previous date segments. For example, it is always better to compare the current month and previous month. In the case of budgeting, comparisons will be done for the current month and the next month. In the case where you have a pivot table and you are using the month column, at the end of every month you need to modify your tables. This can be automated, by maintaining a column with the current month. In this column, the current month will have 0 value, previous month will be -1 and next month will be 1.

The following five columns are used for Year, Quarter, Month, Week and Day.

```

CurrentYear    Smallint NULL,
CurrentQuater  Smallint NULL,
CurrentMonth   Smallint NULL,
CurrentWeek    Smallint NULL,
CurrentDay     Smallint NULL,

```

The tinyint data type cannot be used as tinyint does not support negative values. Since these columns are updated at the change of a period, they need to be changed periodically. For example, at the start of every month, the CurrentMonth column should be changed. Every day, the CurrentDay column should be changed. Therefore, these columns are updated daily using a DATEDIFF function as shown below.

```
--Update current date information
```

```
UPDATE Dim_Date
```

```
SET
```

```
CurrentYear      = DATEDIFF(yy,GETDATE(),date),
```

```
CurrentQuater    = DATEDIFF(q,GETDATE(),date),
```

```
CurrentMonth     = DATEDIFF(m,GETDATE(),date),
```

```
CurrentWeek      = DATEDIFF(ww,GETDATE(),date),
```

```
CurrentDay       = DATEDIFF(dd,GETDATE(),date)
```

All scripts

The following script contains all the columns used for Dim_Date dimension table.

```

CREATE TABLE [dbo].[Dim_Date] (
    [DateKey] [int] NOT NULL,
    [Date] [date] NOT NULL,
    [Day] [tinyint] NOT NULL,
    [DaySuffix] [char](2) NOT NULL,
    [Weekday] [tinyint] NOT NULL,
    [WeekDayName] [varchar](10) NOT NULL,
    [WeekDayName_Short] [char](3) NOT NULL,
    [WeekDayName_FirstLetter] [char](1) NOT NULL,
    [DOWInMonth] [tinyint] NOT NULL,
    [DayOfYear] [smallint] NOT NULL,
    [WeekOfMonth] [tinyint] NOT NULL,
    [WeekOfYear] [tinyint] NOT NULL,
    [Month] [tinyint] NOT NULL,
    [MonthName] [varchar](10) NOT NULL,
    [MonthName_Short] [char](3) NOT NULL,
    [MonthName_FirstLetter] [char](1) NOT NULL,
    [Quarter] [tinyint] NOT NULL,
    [QuarterName] [varchar](6) NOT NULL,
    [Year] [int] NOT NULL,
    [MMYYYY] [char](6) NOT NULL,
    [MonthYear] [char](7) NOT NULL,
    IsWeekend BIT NOT NULL,
    IsHoliday BIT NOT NULL,
    HolidayName VARCHAR(20) NULL,
    SpecialDays VARCHAR(20) NULL,
    [FinancialYear] [int] NULL,
    [FinancialQuater] [int] NULL,
    [FinancialMonth] [int] NULL,
    [FirstDateofYear] DATE NULL,
    [LastDateofYear] DATE NULL,
    [FirstDateofQuater] DATE NULL,
    [LastDateofQuater] DATE NULL,
    [FirstDateofMonth] DATE NULL,
    [LastDateofMonth] DATE NULL,
    [FirstDateofWeek] DATE NULL,
    [LastDateofWeek] DATE NULL,
    CurrentYear SMALLINT NULL,
    CurrentQuater SMALLINT NULL,
    CurrentMonth SMALLINT NULL,
    CurrentWeek SMALLINT NULL,
    CurrentDay SMALLINT NULL,
    PRIMARY KEY CLUSTERED ([DateKey] ASC)
)

```

The following script is the entire script which will update the data for dim_date.

```
SET NOCOUNT ON
```

```
TRUNCATE TABLE DIM_Date
```

```
DECLARE @CurrentDate DATE = '2016-01-01'
```

```
DECLARE @EndDate DATE = '2020-12-31'
```

```
WHILE @CurrentDate < @EndDate
```

```
BEGIN
```

```
    INSERT INTO [dbo].[Dim_Date] (
```

```
        [DateKey],
```

```
        [Date],
```

```
        [Day],
```

```
        [DaySuffix],
```

```
        [Weekday],
```

```
        [WeekDayName],
```

```
        [WeekDayName_Short],
```

```
        [WeekDayName_FirstLetter],
```

```
        [DOWInMonth],
```

```
        [DayOfYear],
```

```
        [WeekOfMonth],
```

```
        [WeekOfYear],
```

```
        [Month],
```

```
        [MonthName],
```

```
        [MonthName_Short],
```

```
        [MonthName_FirstLetter],
```

```
        [Quarter],
```

Next Steps

- Read how to implement surrogate keys in SCDs - <https://www.mssqltips.com/sqlservertip/4054/creating-a-date-dimension-or-calendar-table-in-sql-server/> (/sqlservertip/4054/creating-a-date-dimension-or-calendar-table-in-sql-server/)
- Other scripts to update Date Dimension:
 - <https://www.codeproject.com/Articles/647950/Create-and-Populate-Date-Dimension-for-Data-Warehouse> (<https://www.codeproject.com/Articles/647950/Create-and-Populate-Date-Dimension-for-Data-Warehouse>)
 - <https://www.mssqltips.com/sqlservertip/4054/creating-a-date-dimension-or-calendar-table-in-sql-server/> (/sqlservertip/4054/creating-a-date-dimension-or-calendar-table-in-sql-server/)
- SQL Server built-in date function tips - <https://www.mssqltips.com/sql-server-tip-category/121/dates/> (/sql-server-tip-category/121/dates/)

Last Updated: 2018-07-31

About the author



(<https://sqlserverauthor/288/dinesh-asanka/>), Dinesh Asanka is a 10 time Data Platform MVP and frequent speaker at local and international conferences with more than 12 years of database experience.

View all my tips (<https://sqlserverauthor/288/dinesh-asanka/>)

Related Resources

- [More Database Developer Tips... \(/sql-server-developer-resources/\)](#)

Follow

- [Get Free SQL Tips \(/get-free-sql-server-tips/?ref=GetFooterMenu\)](#)
- [Twitter \(https://twitter.com/mssqltips\)](https://twitter.com/mssqltips)
- [LinkedIn \(https://www.linkedin.com/groups/2320891/\)](https://www.linkedin.com/groups/2320891/)
- [Facebook \(https://www.facebook.com/mssqltips/\)](https://www.facebook.com/mssqltips/)
- [Pinterest \(https://www.pinterest.com/mssqltips/\)](https://www.pinterest.com/mssqltips/)
- [RSS \(https://feeds.feedburner.com/MSSQLTips-LatestSqlServerTips\)](https://feeds.feedburner.com/MSSQLTips-LatestSqlServerTips)

Learning

- [DBAs \(/sql-server-dba-resources/\)](#)
- [Developers \(/sql-server-developer-resources/\)](#)
- [BI Professionals \(/sql-server-business-intelligence-resources/\)](#)
- [Careers \(/sql-server-professional-development-resources/\)](#)
- [Today's Tip \(/todays-sql-server-tip/\)](#)

Resources

- [Tutorials \(/sql-server-tutorials/\)](#)
- [Webcasts \(/sql-server-webcasts/\)](#)
- [Whitepapers \(/sql-server-whitepapers/\)](#)
- [Tools \(/sql-server-tools/\)](#)

Search

- [Tip Categories \(/sql-server-categories/\)](#)
- [Search By TipID \(/search-tip-id/\)](#)
- [Authors \(/sql-server-mssqltips-authors/\)](#)

Community

- [First Timer? \(/learn-more-about-mssqltips/\)](#)
- [Pictures \(/mssqltips-community/1/\)](#)
- [Contribute \(/contribute/\)](#)
- [Event Calendar \(/sql-server-event-list/\)](#)
- [User Groups \(/sql-server-user-groups/\)](#)
- [Author of the Year \(/mssqltips-author-of-year/\)](#)

More Info

- [Join \(/get-free-sql-server-tips/?ref=JoinFooterMenu\)](#)
- [About \(/about/\)](#)
- [Copyright \(/copyright/\)](#)
- [Privacy \(/privacy/\)](#)
- [Disclaimer \(/disclaimer/\)](#)
- [Feedback \(/feedback/\)](#)
- [Advertise \(/advertise/\)](#)

Copyright (c) 2006-2020 [Edgewood Solutions, LLC \(https://www.edgewoodsolutions.com\)](https://www.edgewoodsolutions.com) All rights reserved

Some names and products listed are the registered trademarks of their respective owners.