Date Dimension Population - SQL Server Implementation

# 1. Introduction

This document explains the steps to create and populate a Date Dimension table in SQL Server using a stored procedure. The procedure takes a single input date and populates attributes for all dates within the same calendar year.

# 2. Create Database

Use the following command to create the database:

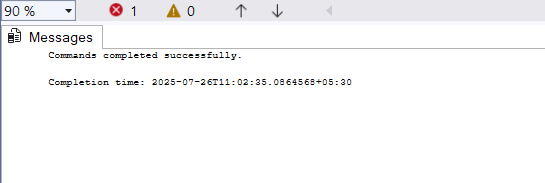
CREATE DATABASE DateDimensionDB;  
GO  
  
USE DateDimensionDB;  
GO

# 3. Create Date Dimension Table

Create the table with all required date attributes:

CREATE TABLE DateDimension (  
 Date DATE PRIMARY KEY,  
 SKDate INT,  
 KeyDate DATE,  
 CalendarDay INT,  
 CalendarMonth INT,  
 CalendarQuarter INT,  
 CalendarYear INT,  
 DayName VARCHAR(20),  
 DayNameShort VARCHAR(10),  
 DayNumberOfWeek INT,  
 DayNumberOfYear INT,  
 DaySuffix VARCHAR(5),  
 FiscalWeek INT,  
 FiscalPeriod INT,  
 FiscalQuarter INT,  
 FiscalYear INT,  
 FiscalYearPeriod VARCHAR(10)  
);  
GO

📌



# 4. Create Stored Procedure

This procedure generates all date dimension entries for one calendar year based on an input date:

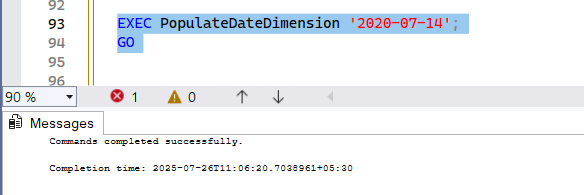
CREATE PROCEDURE PopulateDateDimension  
 @InputDate DATE  
AS  
BEGIN  
 SET NOCOUNT ON;  
  
 DECLARE @StartDate DATE = DATEFROMPARTS(YEAR(@InputDate), 1, 1);  
 DECLARE @EndDate DATE = DATEFROMPARTS(YEAR(@InputDate), 12, 31);  
  
 ;WITH DateSequence AS (  
 SELECT @StartDate AS DateValue  
 UNION ALL  
 SELECT DATEADD(DAY, 1, DateValue)  
 FROM DateSequence  
 WHERE DateValue < @EndDate  
 )  
 INSERT INTO DateDimension (  
 SKDate,  
 KeyDate,  
 Date,  
 CalendarDay,  
 CalendarMonth,  
 CalendarQuarter,  
 CalendarYear,  
 DayName,  
 DayNameShort,  
 DayNumberOfWeek,  
 DayNumberOfYear,  
 DaySuffix,  
 FiscalWeek,  
 FiscalPeriod,  
 FiscalQuarter,  
 FiscalYear,  
 FiscalYearPeriod  
 )  
 SELECT   
 CONVERT(INT, CONVERT(VARCHAR(8), DateValue, 112)) AS SKDate,  
 DateValue AS KeyDate,  
 DateValue AS Date,  
 DAY(DateValue) AS CalendarDay,  
 MONTH(DateValue) AS CalendarMonth,  
 DATEPART(QUARTER, DateValue) AS CalendarQuarter,  
 YEAR(DateValue) AS CalendarYear,  
 DATENAME(WEEKDAY, DateValue) AS DayName,  
 LEFT(DATENAME(WEEKDAY, DateValue), 3) AS DayNameShort,  
 DATEPART(WEEKDAY, DateValue) AS DayNumberOfWeek,  
 DATEPART(DAYOFYEAR, DateValue) AS DayNumberOfYear,  
 CAST(DAY(DateValue) AS VARCHAR) +   
 CASE   
 WHEN DAY(DateValue) IN (11,12,13) THEN 'th'  
 WHEN RIGHT(CAST(DAY(DateValue) AS VARCHAR),1) = '1' THEN 'st'  
 WHEN RIGHT(CAST(DAY(DateValue) AS VARCHAR),1) = '2' THEN 'nd'  
 WHEN RIGHT(CAST(DAY(DateValue) AS VARCHAR),1) = '3' THEN 'rd'  
 ELSE 'th'  
 END AS DaySuffix,  
 DATEPART(WEEK, DateValue) AS FiscalWeek,  
 MONTH(DateValue) AS FiscalPeriod,  
 DATEPART(QUARTER, DateValue) AS FiscalQuarter,  
 YEAR(DateValue) AS FiscalYear,  
 CAST(YEAR(DateValue) AS VARCHAR) + RIGHT('0' + CAST(MONTH(DateValue) AS VARCHAR), 2) AS FiscalYearPeriod  
 FROM DateSequence  
 OPTION (MAXRECURSION 366);  
END;  
GO

# 5. Execute the Stored Procedure

Execute the procedure for any date in the desired year (e.g., 14th July 2020):

EXEC PopulateDateDimension '2020-07-14';  
GO

📌



# 6. View Inserted Records

Verify the inserted records:

SELECT \* FROM DateDimension ORDER BY Date;  
GO

📌

