

Steps to create our “dim_date” table

Step 1. I've used a pipeline's copy activity to stage a parquet file containing the result of the following query:

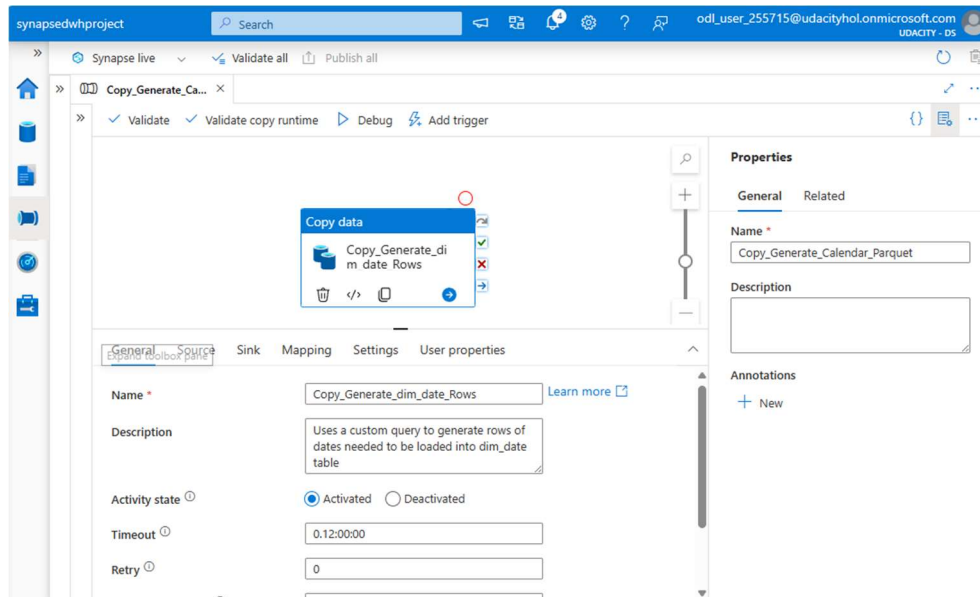
```
-- dimDate generation
-- Used tips from
-- 'https://www.mssqltips.com/sqlservertip/4054/creating-a-date-dimension-or-calendar-table-in-sql-server/'
-- 'https://stackoverflow.com/questions/7428669/how-to-set-the-maxrecursion-option-for-a-cte-inside-a-table-
-- valued-function'
-- as guidelines

-- Creates a table used for our dim_date table
-- We're taking our granularity up to the hour because we need insights during the time of the day, according
-- to the business outcomes requirements.

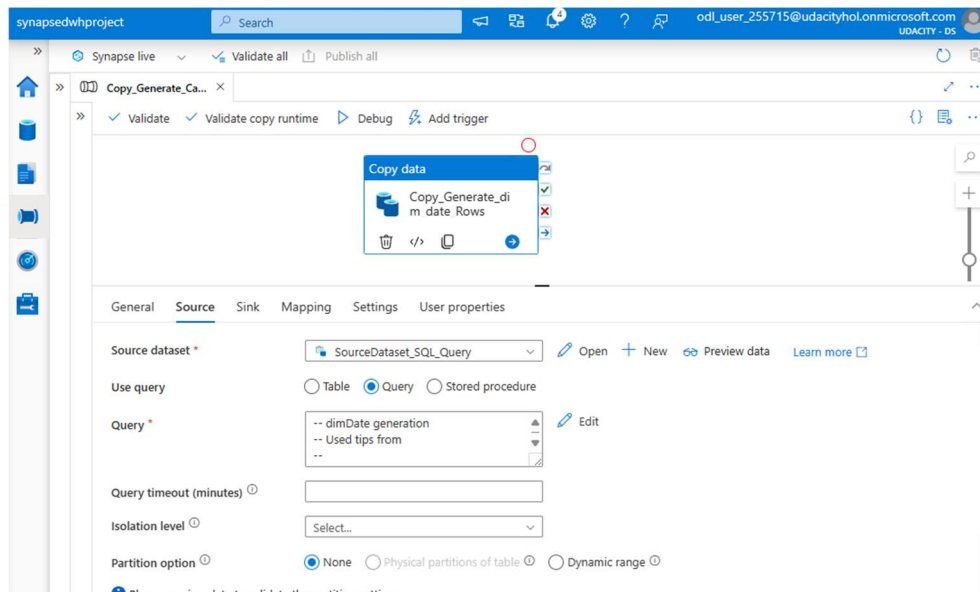
Declare @yearrange int = 10;-- number of years to generate our calendar (i.e 10 for 10 years)
Declare @StartDate date = '20170101';
;WITH seq0(1) AS
(
    SELECT 0 UNION ALL SELECT 1 + 1 FROM seq0
    WHERE 1+1 < 100
),
seq1(m) AS
(
    SELECT 0 UNION ALL SELECT m + 1 FROM seq1
    WHERE m+1 < 90
),
calendar(daydatetime) AS
(
    SELECT
        DATEADD(day, 1+(100*m), @StartDate) FROM seq0,seq1
    WHERE
        (1 + (100*m)) <= DATEDIFF(day, @StartDate, DATEADD(day, -1, DATEADD(year, @yearrange, @StartDate)))
),
hourseq(dayhour) AS
(
    SELECT
        CAST('00:00:00' as datetime)
    UNION ALL
    SELECT
        dayhour + CAST('01:00:00' as datetime)
    FROM hourseq
    WHERE
        dayhour < CAST('23:00:00' as datetime)
),
days(date_daytime) AS
(
    SELECT
        CAST(daydatetime as datetime) + dayhour
    FROM
        calendar, hourseq
)
SELECT
    date_daytime as [date_id],
    CAST(date_daytime as DATE) as [date],
    DATEPART(DAY, date_daytime) as [day_of_month],
    DATEPART(MONTH, date_daytime) as [month],
    DATEPART(YEAR, date_daytime) as [year],
    DATENAME(MONTH, date_daytime) as [month_name],
    DATEPART(WEEKDAY, date_daytime) as [day_of_week],
    DATENAME(WEEKDAY, date_daytime) as [day_name],
    DATEPART(HOUR, date_daytime) as [hour],
    DATEPART(QUARTER, date_daytime) as [quarter]
FROM days
ORDER BY date_daytime
```

Below are the screenshots of the pipeline activity.

1. General tab.

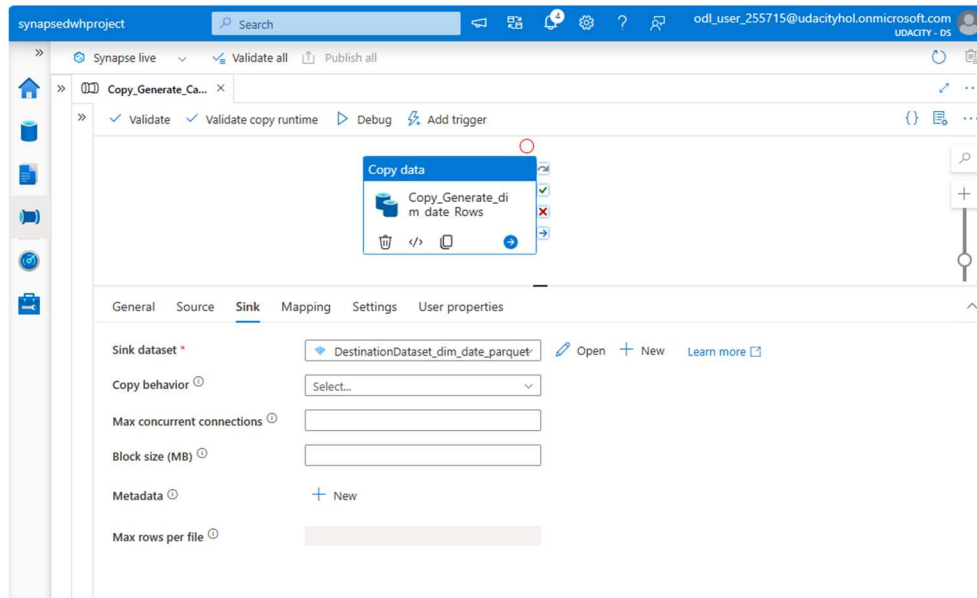


2. Source tab.



Note our choice of "Query" in the "Use query" options available. And, I've put the above code in "Query" box.

3. Sink tab.



Step 2. Once the parquet file was saved in the storage account, under the “dims” folder set in the dataset under the “sink” tab, I created the external table pointing to that file with following instructions:

```
-- The following SQL code creates the dim_date external table using
-- our parquet file generated and saved with a copy pipeline
-- I choose to use a pipeline for creating this dimension table
-- because CETAS was not allowing complex inline SQL statements for data generation

IF NOT EXISTS (SELECT * FROM sys.external_file_formats WHERE name = 'SynapseParquetFormat')
    CREATE EXTERNAL FILE FORMAT [SynapseParquetFormat]
    WITH ( FORMAT_TYPE = PARQUET)
GO

IF NOT EXISTS (SELECT * FROM sys.external_data_sources WHERE name =
'parquet_synapsedwhstoragedlgen2_dfs_core_windows_net')
    CREATE EXTERNAL DATA SOURCE [parquet_synapsedwhstoragedlgen2_dfs_core_windows_net]
    WITH (
        LOCATION = 'abfss://parquet@synapsedwhstoragedlgen2.dfs.core.windows.net'
    )
GO

CREATE EXTERNAL TABLE dbo.dim_date (
    [date_id] datetime2(7),
    [date] date,
    [day_of_month] tinyint,
    [month] tinyint,
    [year] smallint,
    [month_name] varchar(25),
    [day_of_week] tinyint,
    [day_name] varchar(25),
    [hour] tinyint,
    [quarter] tinyint
)
WITH (
    LOCATION = 'dims/dim_date.parquet',
    DATA_SOURCE = [parquet_synapsedwhstoragedlgen2_dfs_core_windows_net],
    FILE_FORMAT = [SynapseParquetFormat]
)
GO

SELECT TOP 100 * FROM dbo.dim_date
GO
```