Big Data Integration

Overview

In this demo, you will upload data to your Azure Storage account, query and import it into your SQL DW using PolyBase, and visualize the data with PowerBI.

Exercise 2: Upload Data to Azure Blob Storage

In this exercise, you will upload a data file from your virtual machine to the Azure Storage account you created in Lab 1, Exercise 3.

1. In your virtual machine, open Microsoft Edge.

2. Navigate to the location for data in Demo 3 Folder:

3. Open a File Explorer window. Navigate to the downloaded file location.

4. Open the extracted folder and locate BeachWeatherStationsAutomatedSensors.csv. This is a CSV file that contains sensor data from various Lake Michigan beaches in Chicago, Illinois. This dataset was obtained from the city's open data portal (https://data.cityofchicago.org).

5. open Azure Storage Manager.

6. Navigate to the storage account you created

7. Expand the container to view the Blob Containers node. Right-click, then click Create Blob Container:

8. Name the new container lab04 (case-sensitive).
9. In the pane on the right, click the Upload button, then click Upload Files:

10. Browse for the BeachWeatherStationsAutomatedSensors.csv file that you downloaded for this lab. Leave the default blob type of Block Blob. When ready, click Upload.

11. The Activity Log will show your upload progress and will notify you when the upload has completed. If you click the Refresh button, the file should also appear in the storage container.

Exercise 2: Use T-SQL with Polybase to Load and Transform Data

In this exercise, you will create a table for the weather data in your SQL DW, using SQL Server Management Studio on your virtual machine. You will also create objects needed to map and import the data from the CSV file in Azure Storage.

13. Switch to or open SQL Server Management Studio (SSMS). Connect to your SQL DW.

14. Enter the following code into a new query window. Do not execute it yet. (The code is also contained in the file DAT20x_Lab_4_Scripts.sql from the .zip file you downloaded in Exercise 1.)

```
CREATE MASTER KEY;
    CREATE DATABASE SCOPED CREDENTIAL LabStorageCredential
    WITH
       IDENTITY = 'user',
       SECRET = '<StorageAccountKey1>'
    ;
    CREATE EXTERNAL DATA SOURCE LabStorage
    WITH (
       TYPE = HADOOP,
       LOCATION =
'wasbs://<StorageContainer>@<StorageAccount>.blob.core.windows.net',
       CREDENTIAL = LabStorageCredential
    );
    CREATE EXTERNAL FILE FORMAT TextFile
    WITH (
       FORMAT_TYPE = DelimitedText,
       FORMAT_OPTIONS (FIELD_TERMINATOR = ',')
     );
```

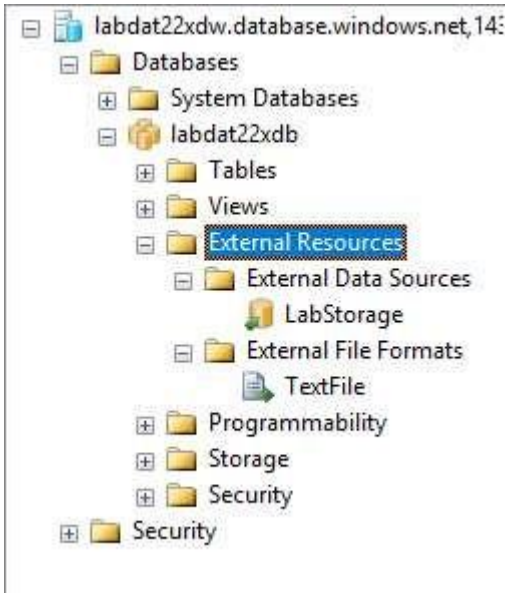
15. Modify the code to include your Azure storage account, container, and key. You should have this information from completing Lab 1, but if not, you can find them by accessing your storage account

from the Azure Portal. Note: the account, container, and key fields are casesensitive when querying. The values will look something like the following:

Storage Account: crdat220
Container: lab04
Key: DCatHPwArPl2uZMoPd7f2QgC/TtQXfUyX9aeslbXLPszcb4QutzttaM4v1Zh49A==

16. Execute the queries.

17. In the Object Explorer pane, refresh the database, then expand External Resources. You should see your new objects:



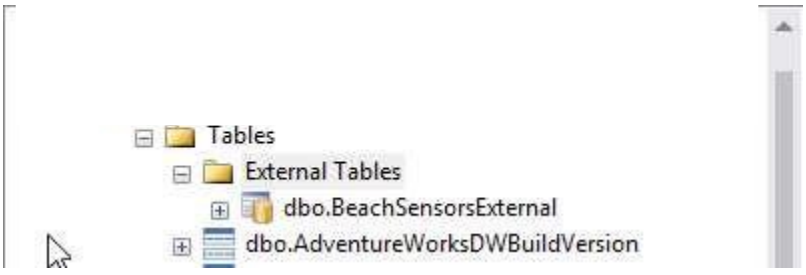
18. You can also verify the new objects by querying the catalog views sys.database_credentials, sys.external_data_sources, and sys_external_file_formats

19. To create the external table for the weather data, enter the following code into a query window that is connected to your SQL DW

```
CREATE EXTERNAL TABLE dbo.BeachSensorsExternal (
    StationName VARCHAR(50) NOT NULL,
    MeasurementTimestamp VARCHAR(50) NOT NULL,
    AirTemperature DECIMAL(9,2) NULL,
    WetBulbTemperature DECIMAL(9,2) NULL,
    Humidity DECIMAL(9,2) NULL,
    RainIntensity DECIMAL(9,2) NULL,
```

```
    IntervalRain DECIMAL(9,2) NULL,
    TotalRain DECIMAL(9,2) NULL,
    PrecipitationType DECIMAL(9,2) NULL,
    WindDirection DECIMAL(9,2) NULL,
    WindSpeed DECIMAL(9,2) NULL,
    MaximumWindSpeed DECIMAL(9,2) NULL,
    BarometricPressure DECIMAL(9,2) NULL,
    SolarRadiation DECIMAL(9,2) NULL,
    Heading DECIMAL(9,2) NULL,
    BatteryLife DECIMAL(9,2) NULL,
    MeasurementTimestampLabel VARCHAR(50) NOT NULL,
    MeasurementID VARCHAR(100) NOT NULL
)
WITH (
   LOCATION='/',
   DATA_SOURCE=LabStorage,
   FILE_FORMAT=TextFile
);
```

20. In the Object Explorer pane, refresh the database, then expand Tables, then External Resources. You should see your new objects:



21. Test the new table with a SELECT query:

```
SELECT TOP 10 * FROM dbo.BeachSensorsExternal;
```

100 %

Results | Messages

	StationName	MeasurementTimestamp	AirTemperature	WetBulbTemperature	Humidity	RainIntensity	IntervalRa
1	Foster Weather Station	2/12/2016 13:00	-6.39	NULL	43.00	NULL	0.00
2	Foster Weather Station	2/4/2016 6:00	-4.50	NULL	67.00	NULL	0.00
3	Oak Street Weather Station	2/3/2016 13:00	4.00	2.80	82.00	0.00	0.00
4	63rd Street Weather Station	2/2/2016 20:00	3.50	3.40	99.00	0.00	6.20
5	Oak Street Weather Station	2/2/2016 1:00	3.20	2.20	85.00	0.00	0.00
6	63rd Street Weather Station	2/1/2016 6:00	1.70	0.90	86.00	0.00	0.00
7	Foster Weather Station	1/31/2016 13:00	9.56	NULL	83.00	NULL	0.10
8	Oak Street Weather Station	1/30/2016 20:00	8.90	6.00	66.00	0.00	0.00
9	63rd Street Weather Station	3/9/2016 12:00	19.00	15.90	73.00	0.00	0.00
10	Oak Street Weather Station	3/8/2016 20:00	19.60	15.20	63.00	0.00	0.00

22. Now that you have created the external table containing the raw data, you will create a second table in your SQL Data Warehouse, transform the data, and load it. Copy and paste the following code into SSDT, then execute it against your SQL DW database.

    In this code, you use T-SQL to create a new table called BeachSensor, then take a subset of columns from the raw table. After the BeachSensor table loads with the modified data from the external table, a query that displays the table's record count executes.

```
CREATE TABLE dbo.BeachSensor
WITH
(
   CLUSTERED COLUMNSTORE INDEX,
   DISTRIBUTION = ROUND_ROBIN
)
AS
SELECT StationName,
 CAST(MeasurementTimestamp as DATETIME) AS MeasurementDateTime,
 AirTemperature,
 WetBulbTemperature,
 Humidity,
```

```
    RainIntensity,
    IntervalRain,
    TotalRain,
    PrecipitationType,
    WindDirection,
    WindSpeed,
    MaximumWindSpeed,
    BarometricPressure,
    SolarRadiation,
    Heading,
    BatteryLife
FROM dbo.BeachSensorsExternal;

SELECT COUNT(*)
FROM dbo.BeachSensor;
```

Exercise 4: Visualize Data with PowerBI

In this exercise you will explore the data that you imported in your data warehouse during the previous exercise by connecting to Power BI through the Azure Portal..

23. In the Azure Portal, open your SQL Data Warehouse database. At the top of the blade, click the Open in Power BI icon.

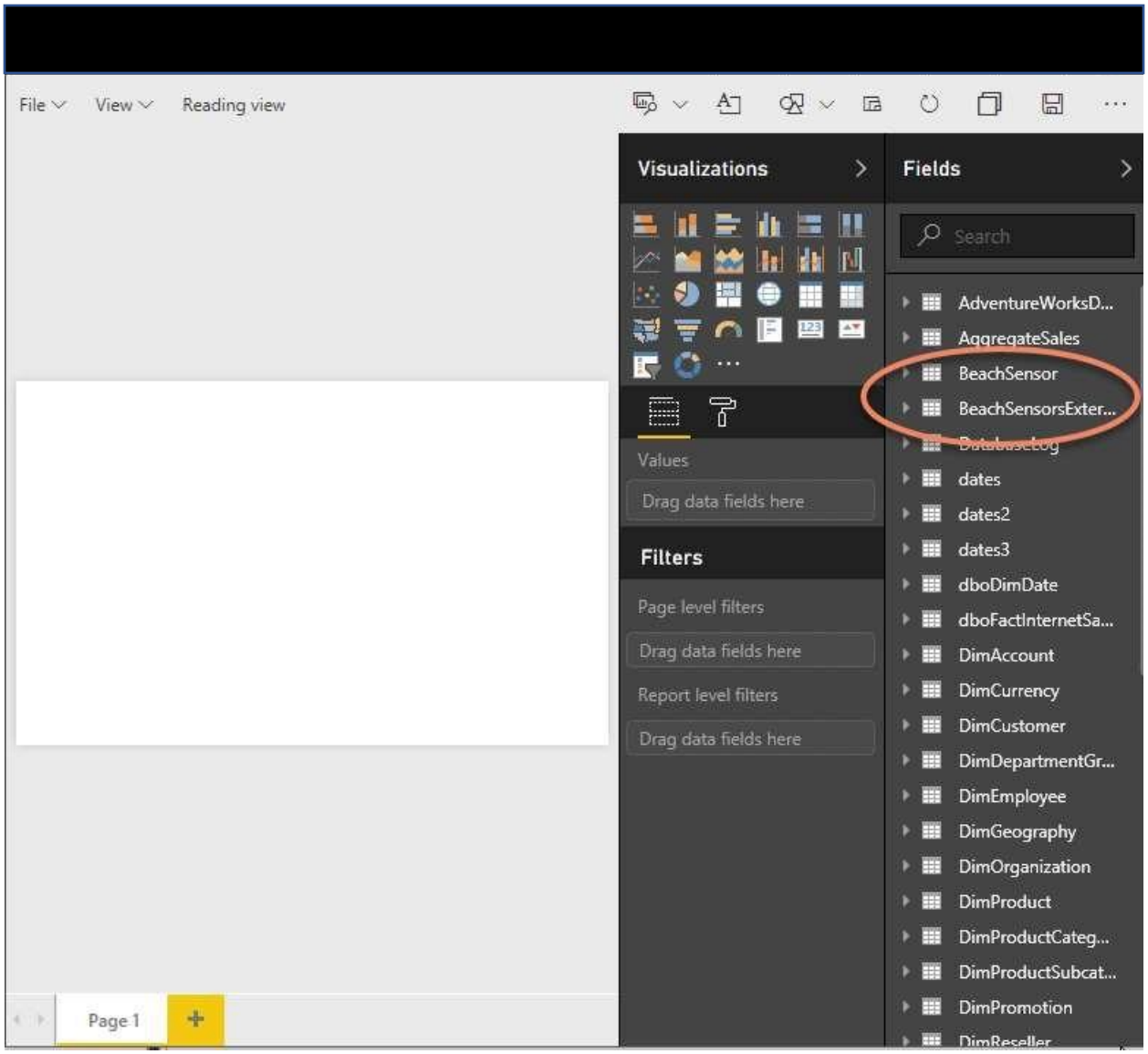
24. In the new browser tab that opens, follow the prompts to sign in to PowerBI.com, using the same credentials you use for the Azure Portal.

25. A Connect to Azure SQL Data Warehouse window should appear, with your Server and Database already populated. Click Next.

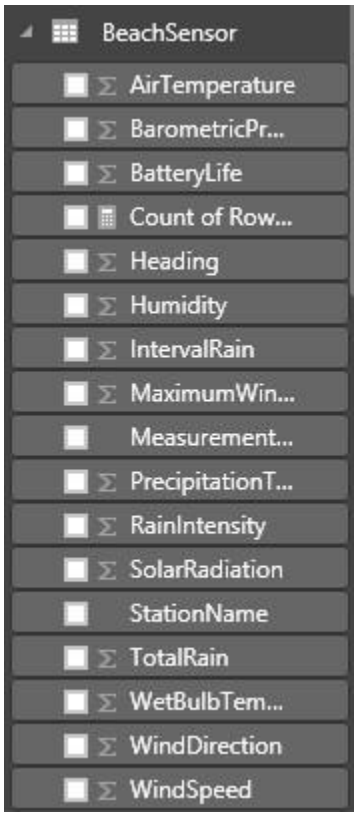
26. Enter your credentials for your SQL DW.

27. Once the schema has been imported into the PowerBI dashboard, a tile appears to represent your SQL DW:

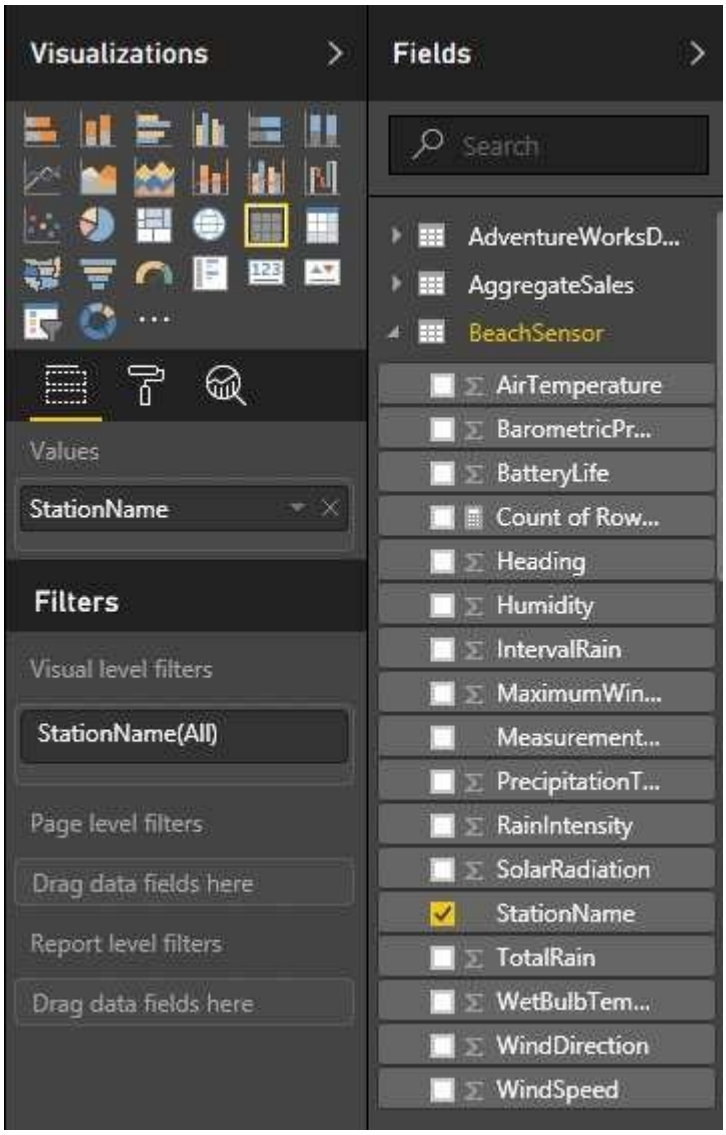
28. Click the tile to open the report editor:



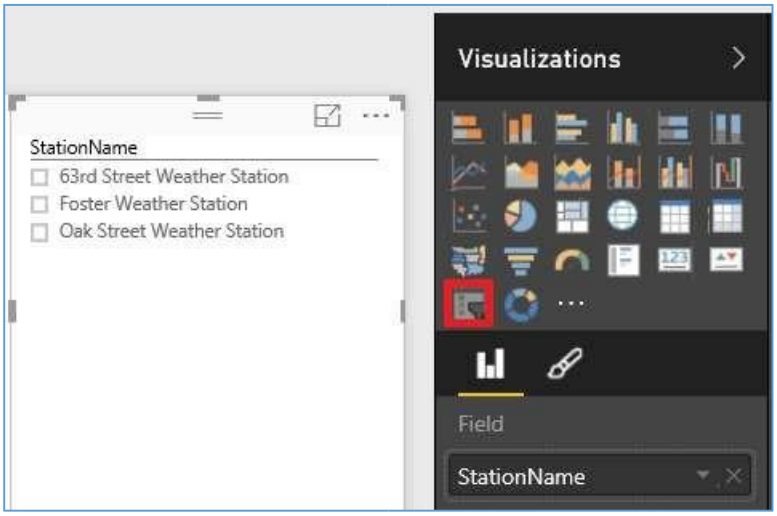
29. Review the fields list. Note that all the sample database tables and views are listed, as well as those you created in this lab.
30. Expand the BeachSensor table by clicking on the arrow to the left of the table name.



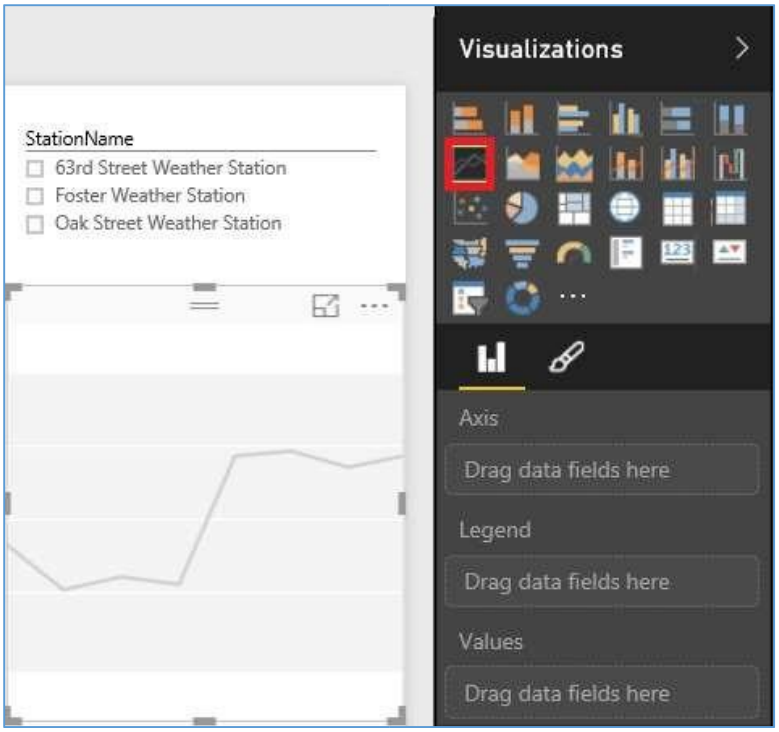
31. Select StationName by checking the box to the left of the field name. StationName will appear in the Values section of the field well, and a table displaying the three weather station locations will appear on the report.



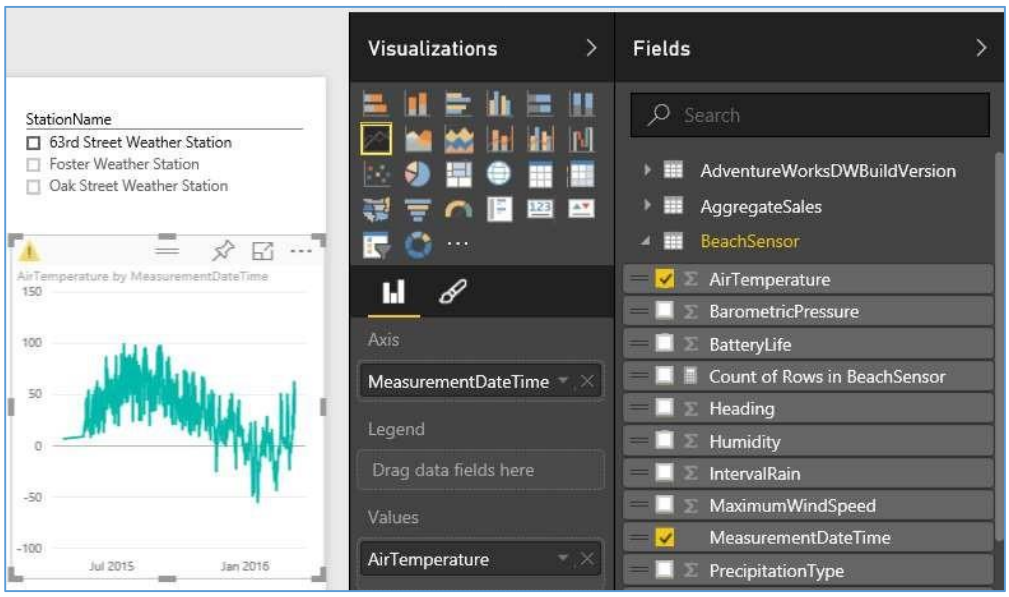
32. In the Visualizations pane, change the table to a slicer by clicking on the Slicer icon. On the report, the visual that previously displayed the three weather station names should now display checkboxes next to the station names.



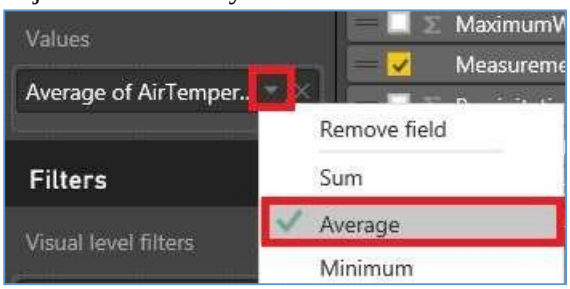
33. Click on whitespace in the report so that the slicer visual is no longer selected. In the Visualizations pane, select Line chart, and a new empty visual will display on the report.



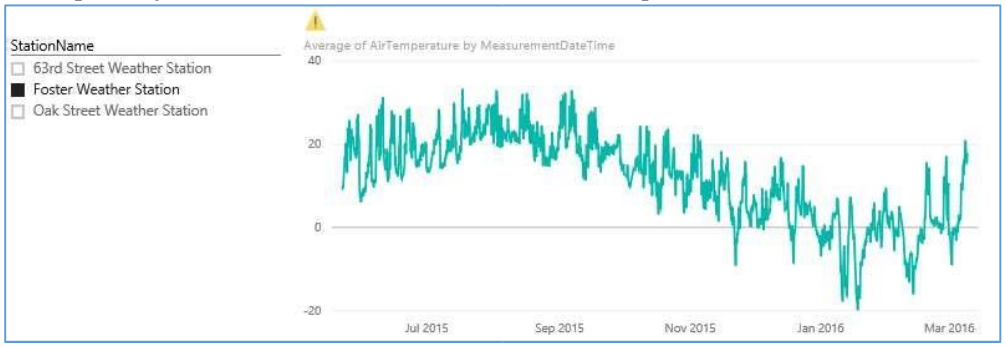
34. In the Fields pane, select MeasurementDateTime, which should appear under Axis in the field well. Then, select AirTemperature, which should appear under Values in the field well. The line chart should also display at this stage.



35. The air temperature (in degrees Celsius) will aggregate using Sum by default. Click the small arrow to the right of AirTemperature in Values, then select Average. The labels for the Y Axis on the line chart will adjust automatically.



36. Check the boxes in the StationName slicer to view the different air temperatures for each location separately. Hover over the line chart to see a tooltip with exact values for each measurement.



37. As desired, continue to explore Power BI by adding new visuals and using different fields from the dataset. When you are done, close the Power BI browser tab.