Load data into a Spark DataFrame

Note:

You are not required to complete the processes, tasks, activities, or steps presented in this example. The various samples provided are for illustrative purposes only and it’s likely that if you try this out you will encounter issues in your system.

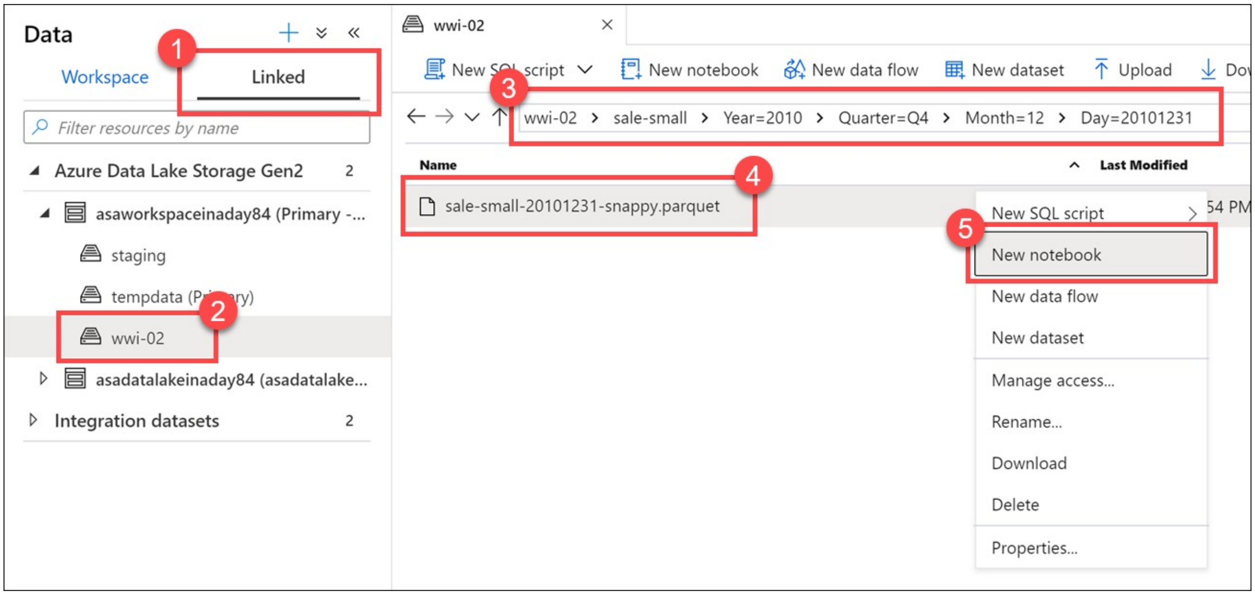
You can load data into an Apache Spark DataFrame from different file types stored in an Azure Storage Account, or from data stored in a dedicated SQL pool.

Some examples of loading data include:

- Read a CSV from Azure Data Lake Store Gen2 as an Apache Spark DataFrame
- Read a CSV from Azure Storage Account as an Apache Spark DataFrame
- Read data from the primary storage account

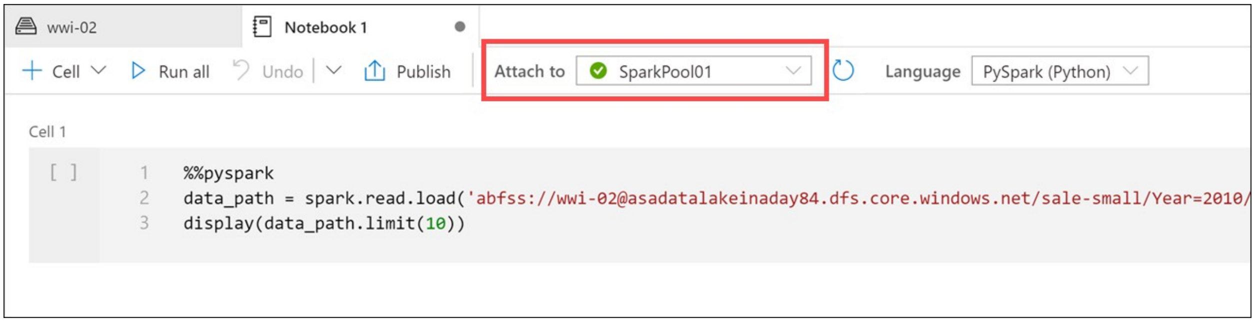
Let's take an example of the company Tailwind Traders. Tailwind Traders has Parquet files stored in their data lake. They want to quickly access the files and explore them using Apache Spark.

One option is to create a DataFrame by using the Data hub in Azure Synapse Studio in order to view the Parquet files in a connected storage account. You can do this by using the *new notebook* context menu to create a new Azure Synapse Notebook that then loads a Spark DataFrame with the contents of a selected parquet file.



The Parquet file is displayed as described.

This generates a notebook with the associated PySpark code that loads the data into an Apache Spark DataFrame and display rows with the header. It also automatically creates the connection to the storage account and file in the `data_path` section.



Attach to SparkPool01

Language PySpark (Python)

```
%%pyspark
data_path = spark.read.load('abfss://wwi-02@asadatalakeinaday84.dfs.core.windows.net/sale-small/Year=2010/
display(data_path.limit(10))
```

The Spark pool is highlighted.
To load data to or from a table into a Spark DataFrame, use the Azure Synapse Apache Spark pool to Synapse SQL connector.

The Azure Synapse Apache Spark pool to Synapse SQL connector is a data source implementation for Apache Spark. It uses Azure Data Lake Storage Gen2 and PolyBase in dedicated SQL pools to efficiently transfer data between the Spark cluster and the Azure Synapse dedicated SQL pool instance.

For example, let's say you want to load the NYC Taxi data into the Spark database named **`nyctaxi`**. Assume the data is available in a table stored in SQLPOOL1. How can you load it into an Apache Spark database named **`nyctaxi`**?

You can complete this task using the following steps:

1. Access the **Develop** hub in Azure Synapse studio.

2. Select **+**, and then select **Notebook**.

3. On the top of the notebook, set **Attach** to the value of **Spark1**.

4. Select **Add code** to add a notebook code cell, and then paste the following text:

Copy

```
%%spark
spark.sql("CREATE DATABASE IF NOT EXISTS nyctaxi")
val df = spark.read.sqlanalytics("SQLPOOL1.dbo.Trip")
df.write.mode("overwrite").saveAsTable("nyctaxi.trip")
```

In this code example, the spark.sql method is used to create a database named **`nyctaxi`**.

A DataFrame named **`df`** reads data from a table named **`Trip`** in the SQLPOOL1 dedicated SQL pool instance.

Finally, the DataFrame **`df`** writes data into it and used the **`saveAsTable`** method to save it as **`nyctaxi.trip`**.

As you can see, there are various ways to load data into an Apache Spark DataFrame depending on the source.