Discover supported languages in spark notebooks

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.

Azure Synapse Analytics Spark pools support various languages. The primary languages available within the notebook environment are:

- PySpark (Python)
- Spark (Scala)
- .NET Spark (C#)
- Spark SQL

It is possible to use multiple languages in one notebook. Specify the language by using a magic command at the beginning of a cell. The following table lists the magic commands to switch cell languages:

Magic command	**Language**	**Description**
`%%pyspark`	Python	Execute a **Python** query against Spark Context
`%%spark`	Scala	Execute a **Scala** query against Spark Context
`%%sql`	SparkSQL	Execute a **SparkSQL** query against Spark Context
`%%csharp`	.NET for Spark C#	Execute a **.NET for Spark C#** query against Spark Context

You cannot reference data or variables directly using different languages in an Azure Synapse Studio notebook. If you wish to do this using Spark, you must first create a temporary table so that it can be referenced across different languages. Here is an example of how to read a **`Scala`** DataFrame in **`PySpark`** and **`SparkSQL`** using a Spark temp table as a workaround.

- The following code shows you how to read a DataFrame from a SQL pool connector using Scala. It also shows how you can create a temporary table.

```
%%spark
val scalaDataFrame = spark.read.sqlanalytics("mySQLPoolDatabase.dbo.mySQLPoolTable")
scalaDataFrame.createOrReplaceTempView( "mydataframetable" )
```

- If you want to query the DataFrame in the above example using Spark SQL, you can add a code cell below the code snippet above, and use the **`%%sql`** command. Using the **`%%sql`** command enables you to input a SQL statement similar to that shown below where you would select everything from the **`mydataframetable`**.

```
%%sql
```

```
SELECT * FROM mydataframetable
```

- If you want to use the data in PySpark, the example below uses the magic command **%%pyspark**, in which you create a new Python DataFrame based on the **mydataframe** table whilst using **spark.sql** to select everything from that table.

```
%%pyspark
myNewPythonDataFrame = spark.sql("SELECT * FROM mydataframetable")
```

You can also use familiar Jupyter magic commands in Azure Synapse Studio notebooks. Review the following list of the currently available magic commands:

Available line magics:

- **%lsmagic**,
- **%time**,
- **%time it**

Available cell magics:

- **%%time**,
- **%%timeit**,
- **%%capture**,
- **%%writefile**,
- **%%sql**,
- **%%pyspark**,
- **%%spark**,
- **%%csharp**