**Set up your PySpark environment:**

* PySpark is the Python API for Spark, which allows you to interface with Spark using Python. To set up PySpark, you need to have Apache Spark installed on your system along with the necessary Python libraries.
* You can install PySpark using pip: pip install pyspark.

**Read data into PySpark DataFrames:**

* PySpark DataFrames are distributed collections of data, similar to pandas DataFrames but distributed across a cluster. You can create PySpark DataFrames from various data sources such as CSV files, JSON files, Hive tables, etc.
* Use spark.read to read data into PySpark DataFrames.

**Perform SparkSQL joins**:

* SparkSQL is a module in Spark that provides support for querying structured and semi-structured data using SQL syntax. You can perform joins, filters, aggregations, etc., using SparkSQL.
* After reading the data into PySpark DataFrames, you can create temporary views or register DataFrames as tables to use them in SparkSQL queries. Then, you can write SQL queries to perform joins.

**Convert the resulting PySpark DataFrame into a Pandas DataFrame:**

* While PySpark is great for distributed processing, sometimes you may want to work with the results locally, especially if the dataset is small enough to fit into memory. In such cases, you can convert the PySpark DataFrame into a Pandas DataFrame.
* Use the toPandas() method to convert a PySpark DataFrame into a Pandas DataFrame.

**Apply functions to the Pandas DataFrame:**

* Once you have the data in a Pandas DataFrame, you can apply various transformations, calculations, or custom functions to the data using pandas' rich set of functions.
* Use the apply() method along with a custom function to apply transformations to columns or rows of the Pandas DataFrame.
* Now, let's put these steps into a sample example: