In PySpark, transformations are operations that create a new RDD (Resilient Distributed Dataset) or DataFrame from an existing one. These transformations are lazy, meaning they are not executed until an action is called. Here are some commonly used transformations in PySpark:

**RDD Transformations**

1. **map(func)**: Applies a function to each element in the RDD and returns a new RDD.

rdd.map(lambda x: x \* 2)

1. **filter(func)**: Returns a new RDD containing only the elements that satisfy a predicate function.

rdd.filter(lambda x: x > 10)

1. **flatMap(func)**: Similar to map, but each input item can be mapped to 0 or more output items.

rdd.flatMap(lambda x: (x, x + 1))

1. **union(otherRDD)**: Returns a new RDD that contains the union of two RDDs.

rdd1.union(rdd2)

1. **distinct()**: Returns a new RDD with distinct elements.

rdd.distinct()

1. **groupByKey()**: Groups the values for each key in a key-value RDD.

rdd.groupByKey()

1. **reduceByKey(func)**: Combines values with the same key using the specified function.

rdd.reduceByKey(lambda x, y: x + y)

1. **sortBy(keyfunc)**: Returns a new RDD sorted by the specified key function.

rdd.sortBy(lambda x: x)

**DataFrame Transformations**

1. **select(\*cols)**: Selects specific columns from the DataFrame.

df.select("column1", "column2")

1. **filter(condition)**: Filters rows based on a condition.

df.filter(df["age"] > 21)

1. **groupBy(\*cols)**: Groups the DataFrame using the specified columns.

df.groupBy("category").count()

1. **withColumn(colName, col)**: Adds a new column or replaces an existing column.

df.withColumn("new\_column", df["existing\_column"] \* 2)

1. **join(other, on, how)**: Joins two DataFrames.

df1.join(df2, on="id", how="inner")

1. **distinct()**: Returns a new DataFrame with distinct rows.

df.distinct()

1. **orderBy(\*cols)**: Sorts the DataFrame by specified columns.

df.orderBy("column1", ascending=False)

1. **drop(\*cols)**: Drops specified columns from the DataFrame.

df.drop("unwanted\_column")

**Example Usage**

Here’s a simple example combining some transformations:

from pyspark.sql import SparkSession

# Initialize Spark session

spark = SparkSession.builder.appName("example").getOrCreate()

# Create a DataFrame

data = [("Alice", 1), ("Bob", 2), ("Cathy", 3)]

df = spark.createDataFrame(data, ["Name", "Value"])

# Transformations

result = (df.filter(df.Value > 1)

.withColumn("NewValue", df.Value \* 2)

.select("Name", "NewValue")

.distinct())

result.show()