Apache Spark Beginner Commands – Use Cases with Examples

# 1. Read a CSV File

Use Case: Load employee data from a CSV file for analysis.

df = spark.read.option("header", True).csv("/FileStore/employee.csv")

# 2. Show Top Records

Use Case: Preview the first 5 records of the dataset.

df.show(5)

# 3. Print Data Schema

Use Case: Check data types of each column to understand structure.

df.printSchema()

# 4. Select Specific Columns

Use Case: Analyze only names and cities of employees.

df.select("name", "city").show()

# 5. Filter Data

Use Case: Find employees older than 30.

df.filter(df.age > 30).show()

# 6. Add a Derived Column

Use Case: Add 5 bonus years to every employee's age.

from pyspark.sql.functions import col  
df.withColumn("age\_plus\_5", col("age") + 5).show()

# 7. Rename a Column

Use Case: Rename 'salary' column to 'monthly\_salary'.

df.withColumnRenamed("salary", "monthly\_salary").show()

# 8. Drop a Column

Use Case: Remove 'city' column from data.

df.drop("city").show()

# 9. GroupBy and Count

Use Case: Count employees per city.

df.groupBy("city").count().show()

# 10. Sort Records by Salary

Use Case: View employees ordered by salary.

df.orderBy(df.salary.desc()).show()

# 11. Join DataFrames

Use Case: Join employee and department data.

df1 = spark.read.csv("/FileStore/employee.csv", header=True, inferSchema=True)  
df2 = spark.read.csv("/FileStore/departments.csv", header=True, inferSchema=True)  
df1.join(df2, df1.dept\_id == df2.dept\_id, "inner").show()

# 12. Write to Output Location

Use Case: Save filtered data as CSV.

df.filter(df.age > 30).write.mode("overwrite").csv("/FileStore/filtered\_output")

# 13. Query Using SQL

Use Case: Find employees with salary > 50000 using SQL.

df.createOrReplaceTempView("employees")  
spark.sql("SELECT name, salary FROM employees WHERE salary > 50000").show()

# 14. Descriptive Statistics

Use Case: Get mean, min, max of numeric columns.

df.describe().show()

# 15. String and Numeric Functions

Use Case: Capitalize names and add bonus to salary.

from pyspark.sql.functions import upper  
df.select(upper(col("name")), col("salary") + 1000).show()