Lab 2: Reading a CSV File from DBFS and Writing It to Another DBFS Location (CSV Format)

Author: Dr. Sandeep Kumar Sharma

Learning Objective

In this lab, you will learn how to read a CSV file stored in DBFS, create a new output folder inside DBFS, and write the data back in CSV format using Spark.

Learning Outcome

By the end of this lab, you will be able to: - Read data from DBFS into a Spark DataFrame - Write DataFrames back to DBFS in CSV format - Understand how Spark writes data as distributed output files

---

Lab Information

We are using the same **employee.csv** file stored in DBFS:

```
/FileStore/tables/employee.csv
```

In this lab, you will read the file and write the output CSV into a different DBFS directory.

Step-by-Step Instructions

Step 1: Set the File Path

Define the DBFS path of the source CSV file.

```
input_path = "/FileStore/tables/employee.csv"
```

Step 2: Read the CSV File

Load the CSV file into a Spark DataFrame.

```
df = spark.read.csv(input_path, header=True, inferSchema=True)
display(df)
```

Step 3: Set the Output Path

Specify the DBFS directory where you want to write the new CSV files.

```
output_path = "/FileStore/tables/employee_output_csv"
```

Step 4: Write the DataFrame as CSV

Write the DataFrame back in CSV format.

```
df.write.csv(output_path)
```

Once executed, Spark will create a directory with CSV part-files inside the specified path.

Explanation

Spark always writes CSV files in directory form because the output can be distributed across multiple worker nodes. Each node writes a part of the data. Even if you have a small file, Spark maintains this distributed architecture.

If you want to create a single CSV file, we will discuss that in an advanced lab.

End of Lab 2

You have now learned how to read and write CSV files within DBFS using Spark. In the next lab, you will write the data into **JSON** format.