Lab 3: Reading a CSV File from DBFS and Writing It as JSON

Author: Dr. Sandeep Kumar Sharma

Learning Objective

In this lab, you will learn how to read a CSV file stored in DBFS and write the resulting DataFrame into JSON format using Apache Spark.

Learning Outcome

By the end of this lab, you will be able to: - Read CSV files from DBFS - Convert and write DataFrames in JSON format - Understand how Spark structures JSON output files

---

Lab Information

We are continuing with the same dataset **employee.csv** stored at:

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

Your goal is to read this CSV file and write the processed data into JSON format in a new 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 for JSON

Specify the location where Spark will write the JSON output.

```
output_path = "/FileStore/tables/employee_output_json"
```

Step 4: Write the DataFrame as JSON

Use Spark's `write.json()` method to save the DataFrame.

```
df.write.json(output_path)
```

After running the above command, Spark creates a folder containing part JSON files.

Explanation

Spark writes JSON the same way it writes CSV — as a directory of part-files. Each part-file contains a portion of the data in JSON format. This distributed output layout makes Spark efficient for large-scale processing.

JSON format is especially useful when your downstream systems require hierarchical or structured data, or when you're generating API-like output.

End of Lab 3

You have successfully converted CSV data into JSON format using Spark. In the next lab, we will reverse the process — we will read JSON and store it back as CSV.