**Example 1: Reading Single-Line JSON from a File**

1. **Setup PySpark**: First, ensure you have PySpark installed and set up in your environment.
2. **Sample JSON Data**: Let's say you have a file called data.json with the following single-line JSON content:

{"name": "Alice", "age": 30, "city": "New York"}

{"name": "Bob", "age": 25, "city": "Los Angeles"}

1. **Reading the JSON File**: You can read the single-line JSON file into a DataFrame as follows:

from pyspark.sql import SparkSession

# Create Spark session

spark = SparkSession.builder \

.appName("Read Single Line JSON") \

.getOrCreate()

# Read JSON file

df = spark.read.json("data.json")

# Show DataFrame

df.show()

**Example 2: Reading Single-Line JSON from a String**

If you want to read JSON data directly from a string instead of a file, you can create a DataFrame from a list of dictionaries.

1. **Sample JSON Data**: Here's a Python list that contains the same data:

json\_data = [

{"name": "Alice", "age": 30, "city": "New York"},

{"name": "Bob", "age": 25, "city": "Los Angeles"}

]

1. **Creating a DataFrame from a List**: You can create a DataFrame directly from this list:

# Create DataFrame from list of dictionaries

df\_from\_string = spark.createDataFrame(json\_data)

# Show DataFrame

df\_from\_string.show()

**Example 3: Handling Nested JSON**

If you have nested JSON objects, you can still read them in a similar way. Consider the following single-line JSON data:

{"name": "Charlie", "age": 35, "address": {"city": "Chicago", "state": "IL"}}

{"name": "Diana", "age": 28, "address": {"city": "Houston", "state": "TX"}}

1. **Reading Nested JSON**: You can read this nested JSON structure as follows:

# Assume this JSON is stored in 'nested\_data.json'

nested\_df = spark.read.json("nested\_data.json")

# Show DataFrame with nested structure

nested\_df.show(truncate=False)

nested\_df.printSchema()

**Summary**

* You can read single-line JSON from a file or create a DataFrame from a list of dictionaries.
* PySpark handles nested JSON structures seamlessly, allowing you to access nested fields easily.