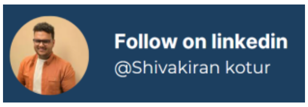
Master PySpark: From Zero to Big Data Hero!!

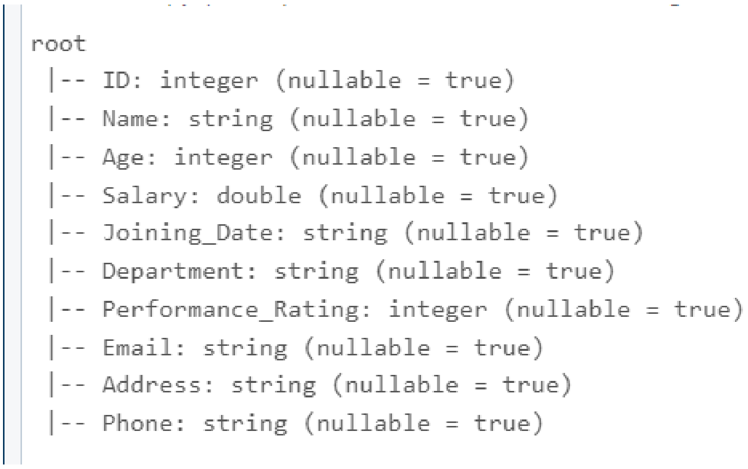
PySpark DataFrame Schema Definition

1. Defining Schema Programmatically with StructType

```
from pyspark.sql.types import *

# Define the schema using StructType
employeeSchema = StructType([
    StructField("ID", IntegerType(), True),
    StructField("Name", StringType(), True),
    StructField("Age", IntegerType(), True),
    StructField("Salary", DoubleType(), True),
    StructField("Joining_Date", StringType(), True),  # Keeping as
String for date issues
    StructField("Department", StringType(), True),
    StructField("Performance_Rating", IntegerType(), True),
    StructField("Email", StringType(), True),
    StructField("Address", StringType(), True),
    StructField("Phone", StringType(), True)
])

# Load the DataFrame with the defined schema
df = spark.read.load("/FileStore/tables/employees.csv",
format="csv", header=True, schema=employeeSchema)

# Print the schema of the DataFrame
df.printSchema()

# Optionally display the DataFrame
# display(df)
```



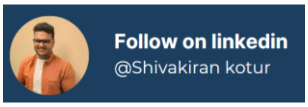
Follow me on LinkedIn – Shivakiran kotur



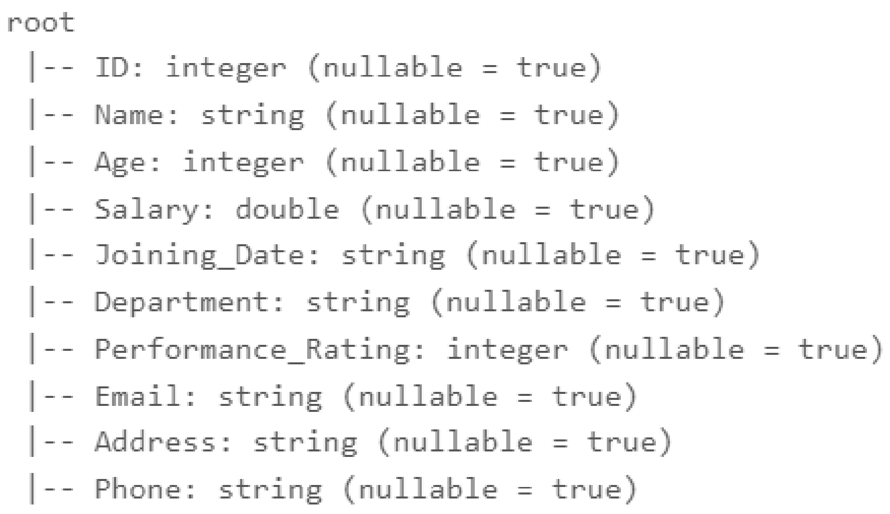
```
root
 |-- ID: integer (nullable = true)
 |-- Name: string (nullable = true)
 |-- Age: integer (nullable = true)
 |-- Salary: double (nullable = true)
 |-- Joining_Date: string (nullable = true)
 |-- Department: string (nullable = true)
 |-- Performance_Rating: integer (nullable = true)
 |-- Email: string (nullable = true)
 |-- Address: string (nullable = true)
 |-- Phone: string (nullable = true)
```

2. Defining Schema as a String

```
# Define the schema as a string
employeeSchemaString = '''
ID Integer,
Name String,
Age Integer,
Salary Double,
Joining_Date String,
Department String,
Performance_Rating Integer,
Email String,
Address String,
Phone String
'''

# Load the DataFrame with the defined schema
df = spark.read.load("dbfs:/FileStore/shared_uploads/imsvk11@gmail.com/employee_data.csv", format="csv", header=True, schema=employeeSchemaString)

# Print the schema of the DataFrame
df.printSchema()

# Optionally display the DataFrame
# display(df)
```



Follow me on LinkedIn – Shivakiran kotur



```
root
 |-- ID: integer (nullable = true)
 |-- Name: string (nullable = true)
 |-- Age: integer (nullable = true)
 |-- Salary: double (nullable = true)
 |-- Joining_Date: string (nullable = true)
 |-- Department: string (nullable = true)
 |-- Performance_Rating: integer (nullable = true)
 |-- Email: string (nullable = true)
 |-- Address: string (nullable = true)
 |-- Phone: string (nullable = true)
```

Explanation

- Schema Definition: Both methods define a schema for the DataFrame, accommodating the dataset's requirements, including handling null values where applicable.
- Data Types: The Joining_Date column is defined as StringType to accommodate potential date format issues or missing values.
- Loading the DataFrame: The spark.read.load method is used to load the CSV file into a DataFrame using the specified schema.
- Printing the Schema: The df.printSchema() function allows you to verify that the DataFrame is structured as intended.