Master PySpark: From Zero to Big Data Hero!!

PySpark: Spark SQL and DataFrames

Spark SQL is a module in Spark for working with structured data. It allows you to query structured data inside Spark using SQL and integrates seamlessly with DataFrames.

A DataFrame in PySpark is a distributed collection of data organized into named columns. It's conceptually similar to a table in a relational database or a DataFrame in R/Python.

How to Create DataFrames in PySpark

Here are some different ways to create DataFrames in PySpark:

---

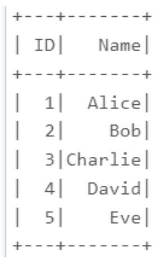
1. Creating DataFrame Manually with Hardcoded Values:

This is one of the most straightforward ways to create a DataFrame using Python lists of tuples.

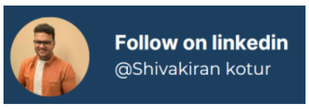
```python
# Sample Data
data = [(1, "Alice"), (2, "Bob"), (3, "Charlie"), (4, "David"), (5, "Eve")]
columns = ["ID", "Name"]

# Create DataFrame
df = spark.createDataFrame(data, columns)

# Show DataFrame
df.show()
```



```
+---+-------+
| ID|   Name|
+---+-------+
|  1|  Alice|
|  2|    Bob|
|  3|Charlie|
|  4|  David|
|  5|    Eve|
+---+-------+
```



Follow me on LinkedIn – Shivakiran kotur

2. Creating DataFrame from Pandas:

```
import pandas as pd

# Sample Pandas DataFrame
pandas_df = pd.DataFrame(data, columns=columns)

# Convert to PySpark DataFrame
df_from_pandas = spark.createDataFrame(pandas_df)
df_from_pandas.show()
```

```
+---+-------+
| ID|   Name|
+---+-------+
|  1|  Alice|
|  2|    Bob|
|  3|Charlie|
|  4|  David|
|  5|    Eve|
+---+-------+
```

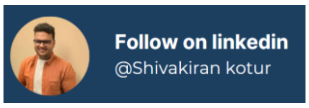
3. Create DataFrame from Dictionary:

```
data_dict = [{"ID": 1, "Name": "Alice"}, {"ID": 2, "Name": "Bob"}]
df_from_dict = spark.createDataFrame(data_dict)
df_from_dict.show()
```

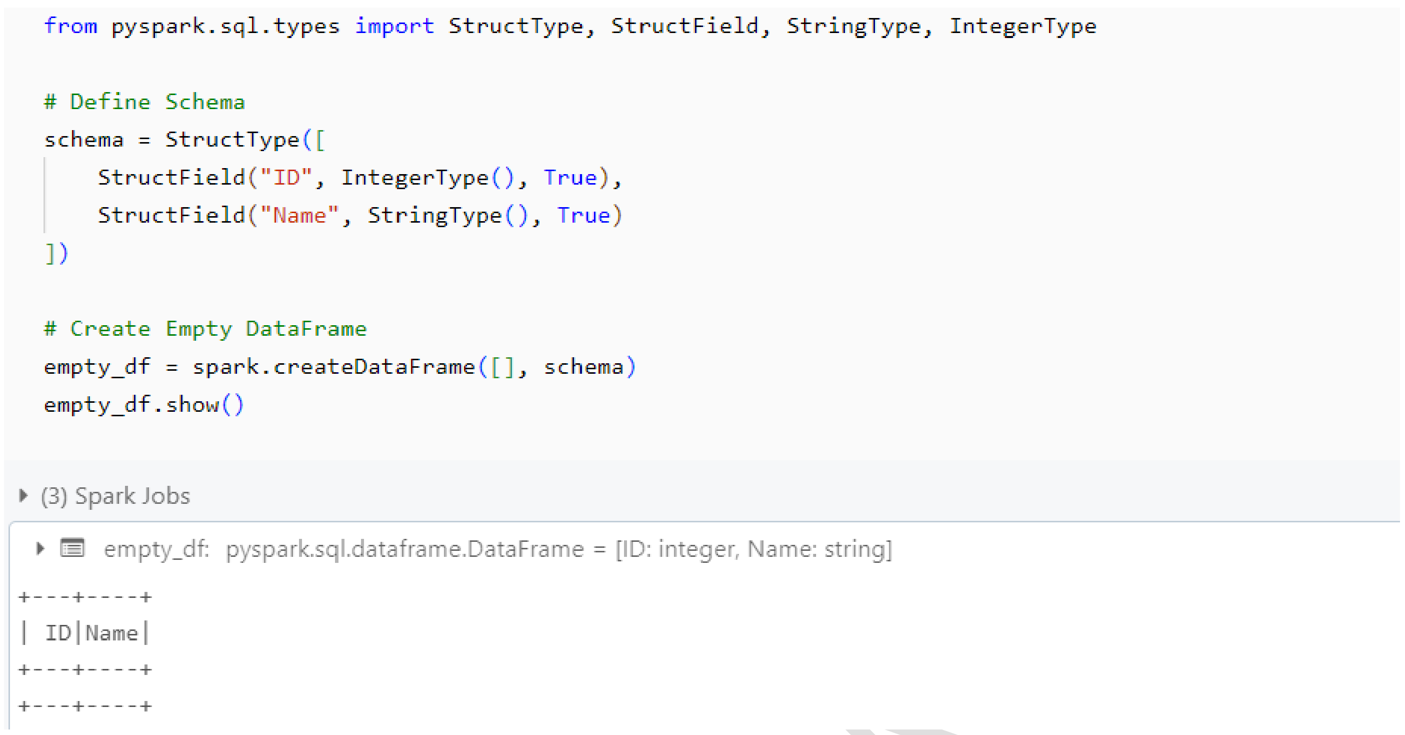
```
+---+-----+
| ID| Name|
+---+-----+
|  1|Alice|
|  2|  Bob|
+---+-----+
```

4. Create Empty DataFrame:

You can create an empty DataFrame with just schema definitions.



Follow me on LinkedIn – Shivakiran kotur



```python
from pyspark.sql.types import StructType, StructField, StringType, IntegerType

# Define Schema
schema = StructType([
    StructField("ID", IntegerType(), True),
    StructField("Name", StringType(), True)
])

# Create Empty DataFrame
empty_df = spark.createDataFrame([], schema)
empty_df.show()
```

(3) Spark Jobs

empty_df: pyspark.sql.dataframe.DataFrame = [ID: integer, Name: string]

```
+---+----+
| ID|Name|
+---+----+
+---+----+
```

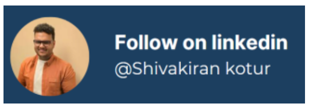
5. Creating DataFrame from Structured Data (CSV, JSON, Parquet

```python
# Reading CSV file into DataFrame
df_csv = spark.read.csv("/path/to/file.csv", header=True,
inferSchema=True)
df_csv.show()

# Reading JSON file into DataFrame
df_json = spark.read.json("/path/to/file.json")
df_json.show()

# Reading Parquet file into DataFrame
df_parquet = spark.read.parquet("/path/to/file.parquet")
df_parquet.show()
```

Follow me on LinkedIn – Shivakiran kotur



show() Function in PySpark DataFrames

The show() function in PySpark displays the contents of a DataFrame in a tabular format. It has several useful parameters for customization:

1. n: Number of rows to display (default is 20)
2. truncate: If set to True, it truncates column values longer than 20 characters (default is True).
3. vertical: If set to True, prints rows in a vertical format.

```
#Show the first 3 rows, truncate columns to 25 characters, and
display vertically:
df.show(n=3, truncate=25, vertical=True)

#Show entire DataFrame (default settings):
df.show()

#Show the first 10 rows:
df.show(10)

#Show DataFrame without truncating any columns:
df.show(truncate=False)
```

Follow me on LinkedIn – Shivakiran kotur