03-working_with_df

February 22, 2025

[1]:

```
from pyspark.sql import SparkSession

spark = SparkSession.builder \
        .appName("Working With DF") \
        .master("yarn") \
        .getOrCreate()
```

```
25/02/11 16:23:41 WARN SparkSession: Using an existing Spark session; only
runtime SQL configurations will take effect.
```

[2]:

```
spark
```

[2]:

```
<pyspark.sql.session.SparkSession at 0x7f0ef4b916c0>
```

[3]:

```
!hadoop fs -ls /ecommerce_data/ecommerce_data/500MB
```

```
Found 5 items
-rw-r--r--   2 lokesh hadoop  570783961 2025-02-10 16:00
/ecommerce_data/ecommerce_data/500MB/customers.csv
-rw-r--r--   2 lokesh hadoop  480952071 2025-02-10 16:00
/ecommerce_data/ecommerce_data/500MB/items.csv
-rw-r--r--   2 lokesh hadoop  472632078 2025-02-10 16:00
/ecommerce_data/ecommerce_data/500MB/orders.csv
-rw-r--r--   2 lokesh hadoop  468231725 2025-02-10 16:01
/ecommerce_data/ecommerce_data/500MB/payments.csv
-rw-r--r--   2 lokesh hadoop  448185359 2025-02-10 16:01
/ecommerce_data/ecommerce_data/500MB/shippings.csv
```

[4]:

```
hdfs_path = "/ecommerce_data/ecommerce_data/500MB/customers.csv"
```

[5]:

```
df = spark.read \
     .format('csv') \
     .option('header', 'true') \
     .option('inferschema', 'true') \
     .load(hdfs_path)
```

[6]:

```
df.show(5)
```

```
+-----------+----------+---------+-----------+-------+-------------------+------
---+
|customer_id|      name|     city|      state|country|
registration_date|is_active|
+-----------+----------+---------+-----------+-------+-------------------+------
---+
|          0|Customer_0|   Mumbai|  Telangana|  India|2023-03-21 00:00:00|
true|
|          1|Customer_1|  Chennai|West Bengal|  India|2023-05-27 00:00:00|
false|
|          2|Customer_2|     Pune|  Karnataka|  India|2023-10-11 00:00:00|
false|
|          3|Customer_3|Hyderabad|    Gujarat|  India|2023-11-11 00:00:00|
false|
|          4|Customer_4|   Mumbai|  Karnataka|  India|2023-05-09 00:00:00|
false|
+-----------+----------+---------+-----------+-------+-------------------+------
---+
only showing top 5 rows
```

[7]:

```
df.printSchema()
```

```
root
 |-- customer_id: integer (nullable = true)
 |-- name: string (nullable = true)
 |-- city: string (nullable = true)
 |-- state: string (nullable = true)
 |-- country: string (nullable = true)
 |-- registration_date: timestamp (nullable = true)
 |-- is_active: boolean (nullable = true)
```

[9]:

```
df.createOrReplaceTempView('customers')
```

[11]:

```
spark.sql("SELECT * FROM customers limit 5").show()
```

```
[Stage 3:>                                                          (0 + 1) / 5]

+-----------+----------+---------+-----------+-------+-------------------+------
---+
|customer_id|      name|     city|      state|country|
registration_date|is_active|
+-----------+----------+---------+-----------+-------+-------------------+------
---+
|          0|Customer_0|   Mumbai|  Telangana|  India|2023-03-21 00:00:00|
true|
|          1|Customer_1|  Chennai|West Bengal|  India|2023-05-27 00:00:00|
false|
|          2|Customer_2|     Pune|  Karnataka|  India|2023-10-11 00:00:00|
false|
|          3|Customer_3|Hyderabad|    Gujarat|  India|2023-11-11 00:00:00|
false|
|          4|Customer_4|   Mumbai|  Karnataka|  India|2023-05-09 00:00:00|
false|
+-----------+----------+---------+-----------+-------+-------------------+------
---+
```

[13]:

```
spark.stop()
```

[15]:

```
spark = SparkSession.builder \
        .appName("Working With DF 1") \
        .master("yarn") \
        .getOrCreate()
```

```
25/02/11 16:31:31 WARN SparkSession: Using an existing Spark session; only
runtime SQL configurations will take effect.
```

[16]:

```
from pyspark.sql.types import *

schema = StructType([
StructField("customer_id", IntegerType(), True),
StructField("name_of_customer", StringType(), True),
StructField("city", StringType(), True),
StructField("state", StringType(), True),
StructField("country", StringType(), True),
StructField("registration_date", StringType(), True),
StructField("is_active", BooleanType(), True),
])
```

[17]:

```
df_explicit = spark.read \
.format("csv") \
.option("header", "false") \
.schema(schema) \
.load (hdfs_path)
```

[18]:

```
df_explicit.show(5)
```

```
[Stage 0:>                                                          (0 + 1) / 1]

+-----------+----------------+---------+-----------+-------+-----------------+--
-------+
|customer_id|name_of_customer|     city|
state|country|registration_date|is_active|
+-----------+----------------+---------+-----------+-------+-----------------+--
-------+
|       null|             name|     city|      state|country|registration_date|
null|
|          0|       Customer_0|   Mumbai|  Telangana|  India|       2023-03-21|
true|
|          1|       Customer_1|  Chennai|West Bengal|  India|       2023-05-27|
false|
|          2|       Customer_2|     Pune|  Karnataka|  India|       2023-10-11|
false|
|          3|       Customer_3|Hyderabad|    Gujarat|  India|       2023-11-11|
false|
+-----------+-----------------+---------+-----------+-------+-----------------+--
--------+
only showing top 5 rows
```

[19]:

```
df_4 = spark.read \
    .format('csv') \
    .option('inferSchema', 'true') \
    .option('header', 'true') \
    .load(hdfs_path)
```

[20]:

```
df_5 = spark.read \
    .format('csv') \
    .option('inferSchema', 'true') \
    .option('header', 'true') \
    .option('samplingRatio',0.1)\
    .load(hdfs_path)
```

[ ]:

```

```

[ ]:

```

```

[22]:

```
ddl_schema = " customer_id INT NOT NULL, name INT, city STRING, state␣
 ↪STRING,country STRING, registration_date TIMESTAMP,is_active BOOLEAN"
```

[23]:

```
df_ddl_explicit = spark.read \
.format("csv") \
.option("header", "true") \
.schema(ddl_schema) \
.load (hdfs_path)
```

[24]:

```
df_ddl_explicit.show(5)
```

```
+-----------+----+---------+-----------+-------+-------------------+---------+
|customer_id|name|     city|      state|country|  registration_date|is_active|
+-----------+----+---------+-----------+-------+-------------------+---------+
|          0|null|   Mumbai|  Telangana|  India|2023-03-21 00:00:00|     true|
|          1|null|  Chennai|West Bengal|  India|2023-05-27 00:00:00|    false|
|          2|null|     Pune|  Karnataka|  India|2023-10-11 00:00:00|    false|
|          3|null|Hyderabad|    Gujarat|  India|2023-11-11 00:00:00|    false|
|          4|null|   Mumbai|  Karnataka|  India|2023-05-09 00:00:00|    false|
+-----------+----+---------+-----------+-------+-------------------+---------+
only showing top 5 rows
```

[25]:
```
df_ddl_explicit.printSchema()
```

```
root
 |-- customer_id: integer (nullable = true)
 |-- name: integer (nullable = true)
 |-- city: string (nullable = true)
 |-- state: string (nullable = true)
 |-- country: string (nullable = true)
 |-- registration_date: timestamp (nullable = true)
 |-- is_active: boolean (nullable = true)
```

[26]:
```
spark.stop()
```

[ ]: