Read data from source to DataFrame in local Spark setup and display DataFrame schema.

In [129…
```
from pyspark.sql import SparkSession
from pyspark.sql.types import *
from pyspark.sql.functions import *
```

In [130…
```
# spark = SparkSession.builder.appName("PySpark Assignment").getOrCreate()
spark = SparkSession.builder.getOrCreate()

titanic = spark.read.option("header", "true").option("inferSchema", "true").option("header", "false").csv("data/tit
# titanic.printSchema()
```

In [131…
```
columns = ["PassengerId","Survived","Pclass","Name","Sex","Age","SibSp","Parch","Ticket","Fare","Cabin","Embarked",
titanic = titanic.toDF(*columns)

# Option 2: Use withColumnRenamed() method to rename existing columns
for i, col_name in enumerate(titanic.columns):
    titanic = titanic.withColumnRenamed(col_name, columns[i])
titanic.show()
titanic.printSchema()
```

```
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+---
-----------------+
|PassengerId|Survived|Pclass|                Name|   Sex| Age|SibSp|Parch|          Ticket|   Fare|Cabin|Embarked|
Timestamp|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+---
-----------------+
|          1|       0|     3|Braund, Mr. Owen ...|  male|  22|    1|    0|       A/5 21171|   7.25| null|       S|20
20-01-01 13:45:25|
|          2|       1|     1|Cumings, Mrs. Joh...|female|  38|    1|    0|        PC 17599|71.2833|  C85|       C|20
20-01-01 13:44:48|
|          3|       1|     3|Heikkinen, Miss. ...|female|  26|    0|    0|STON/O2. 3101282|  7.925| null|       S|20
20-01-01 13:38:11|
|          4|       1|     1|Futrelle, Mrs. Ja...|female|  35|    1|    0|          113803|   53.1| C123|       S|20
20-01-01 13:32:00|
|          5|       0|     3|Allen, Mr. Willia...|  male|  35|    0|    0|          373450|   8.05| null|       S|20
20-01-01 13:36:30|
|          6|       0|     3|    Moran, Mr. James|  male|null|    0|    0|          330877| 8.4583| null|       Q|20
20-01-01 13:31:39|
|          7|       0|     1|McCarthy, Mr. Tim...|  male|  54|    0|    0|           17463|51.8625|  E46|       S|20
20-01-01 13:37:31|
|          8|       0|     3|Palsson, Master. ...|  male|   2|    3|    1|          349909| 21.075| null|       S|20
20-01-01 13:49:08|
|          9|       1|     3|Johnson, Mrs. Osc...|female|  27|    0|    2|          347742|11.1333| null|       S|20
20-01-01 13:33:42|
|         10|       1|     2|Nasser, Mrs. Nich...|female|  14|    1|    0|          237736|30.0708| null|       C|20
20-01-01 13:32:53|
|         11|       1|     3|Sandstrom, Miss. ...|female|   4|    1|    1|         PP 9549|   16.7|   G6|       S|20
20-01-01 13:32:23|
|         12|       1|     1|Bonnell, Miss. El...|female|  58|    0|    0|          113783|  26.55| C103|       S|20
20-01-01 13:30:12|
|         13|       0|     3|Saundercock, Mr. ...|  male|  20|    0|    0|       A/5. 2151|   8.05| null|       S|20
20-01-01 13:33:34|
|         14|       0|     3|Andersson, Mr. An...|  male|  39|    1|    5|          347082| 31.275| null|       S|20
20-01-01 13:30:20|
|         15|       0|     3|Vestrom, Miss. Hu...|female|  14|    0|    0|          350406| 7.8542| null|       S|20
20-01-01 13:41:17|
|         16|       1|     2|Hewlett, Mrs. (Ma...|female|  55|    0|    0|          248706|   16.0| null|       S|20
20-01-01 13:34:22|
|         17|       0|     3|Rice, Master. Eugene|  male|   2|    4|    1|          382652| 29.125| null|       Q|20
20-01-01 13:41:55|
```

```
|         18|       1|     2|Williams, Mr. Cha...|  male|null|    0|    0|          244373|   13.0| null|       S|20
20-01-01 13:39:35|
|         19|       0|     3|Vander Planke, Mr...|female|  31|    1|    0|          345763|   18.0| null|       S|20
20-01-01 13:39:38|
|         20|       1|     3|Masselmani, Mrs. ...|female|null|    0|    0|            2649|  7.225| null|       C|20
20-01-01 13:36:56|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+--
-----------------+
only showing top 20 rows

root
 |-- PassengerId: integer (nullable = true)
 |-- Survived: integer (nullable = true)
 |-- Pclass: integer (nullable = true)
 |-- Name: string (nullable = true)
 |-- Sex: string (nullable = true)
 |-- Age: integer (nullable = true)
 |-- SibSp: integer (nullable = true)
 |-- Parch: integer (nullable = true)
 |-- Ticket: string (nullable = true)
 |-- Fare: double (nullable = true)
 |-- Cabin: string (nullable = true)
 |-- Embarked: string (nullable = true)
 |-- Timestamp: timestamp (nullable = true)
```

In [132…

```
column_types = titanic.dtypes
column_types
```

Out[132]:
```
[('PassengerId', 'int'),
 ('Survived', 'int'),
 ('Pclass', 'int'),
 ('Name', 'string'),
 ('Sex', 'string'),
 ('Age', 'int'),
 ('SibSp', 'int'),
 ('Parch', 'int'),
 ('Ticket', 'string'),
 ('Fare', 'double'),
 ('Cabin', 'string'),
 ('Embarked', 'string'),
 ('Timestamp', 'timestamp')]
```

For numerical columns, calculate minimum, maximum and average values.

In [133…
```
numerical_columns = [column[0] for column in column_types if column[1] in ['int', 'float']]
summary_df = titanic.describe(numerical_columns)
summary_df.select(numerical_columns).summary("max","min","mean").show()
```

```
+-------+------------------+------------------+------------------+------------------+------------------+------------------+------------
-----+
|summary|       PassengerId|          Survived|            Pclass|               Age|             SibSp|
Parch|
+-------+------------------+------------------+------------------+------------------+------------------+------------------+------------
-----+
|    max|               891|               891|               891|                80|               891|
891|
|    min|                 1|                 0|0.8360712409770491|                 0|                 0|
0|
|   mean|497.27076840304596|178.57408616762064|179.62894264325715|167.64315089562422|180.12515025772694|179.63753018
72114|
+-------+------------------+------------------+------------------+------------------+------------------+------------------+------------
-----+
```

For categorical columns, create and apply UDF that will change the last letter of every word to “1”.

In [134…

```
str_columns = ["Sex","Cabin","Embarked"]
def change_last_letter_after_space(word):
    if word is not None:
        words = word.split()
        for i in range(len(words)):
            words[i] = words[i][:-1] + "1"
        return " ".join(words)
    return word

change_last_letter_udf = udf(change_last_letter_after_space, StringType())


for column in str_columns:
    titanic = titanic.withColumn(column, change_last_letter_udf(titanic[column]))

titanic.show()
```

```
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+--
-----------------+
|PassengerId|Survived|Pclass|                Name|   Sex| Age|SibSp|Parch|          Ticket|   Fare|Cabin|Embarked|
Timestamp|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+--
-----------------+
|          1|       0|     3|Braund, Mr. Owen ...|  mal1|  22|    1|    0|       A/5 21171|   7.25| null|       1|20
20-01-01 13:45:25|
|          2|       1|     1|Cumings, Mrs. Joh...|femal1|  38|    1|    0|        PC 17599|71.2833|  C81|       1|20
20-01-01 13:44:48|
|          3|       1|     3|Heikkinen, Miss. ...|femal1|  26|    0|    0|STON/O2. 3101282|  7.925| null|       1|20
20-01-01 13:38:11|
|          4|       1|     1|Futrelle, Mrs. Ja...|femal1|  35|    1|    0|          113803|   53.1| C121|       1|20
20-01-01 13:32:00|
|          5|       0|     3|Allen, Mr. Willia...|  mal1|  35|    0|    0|          373450|   8.05| null|       1|20
20-01-01 13:36:30|
|          6|       0|     3|    Moran, Mr. James|  mal1|null|    0|    0|          330877| 8.4583| null|       1|20
20-01-01 13:31:39|
|          7|       0|     1|McCarthy, Mr. Tim...|  mal1|  54|    0|    0|           17463|51.8625|  E41|       1|20
20-01-01 13:37:31|
|          8|       0|     3|Palsson, Master. ...|  mal1|   2|    3|    1|          349909| 21.075| null|       1|20
20-01-01 13:49:08|
|          9|       1|     3|Johnson, Mrs. Osc...|femal1|  27|    0|    2|          347742|11.1333| null|       1|20
20-01-01 13:33:42|
|         10|       1|     2|Nasser, Mrs. Nich...|femal1|  14|    1|    0|          237736|30.0708| null|       1|20
20-01-01 13:32:53|
|         11|       1|     3|Sandstrom, Miss. ...|femal1|   4|    1|    1|         PP 9549|   16.7|   G1|       1|20
20-01-01 13:32:23|
|         12|       1|     1|Bonnell, Miss. El...|femal1|  58|    0|    0|          113783|  26.55| C101|       1|20
20-01-01 13:30:12|
|         13|       0|     3|Saundercock, Mr. ...|  mal1|  20|    0|    0|       A/5. 2151|   8.05| null|       1|20
20-01-01 13:33:34|
|         14|       0|     3|Andersson, Mr. An...|  mal1|  39|    1|    5|          347082| 31.275| null|       1|20
20-01-01 13:30:20|
|         15|       0|     3|Vestrom, Miss. Hu...|femal1|  14|    0|    0|          350406| 7.8542| null|       1|20
20-01-01 13:41:17|
|         16|       1|     2|Hewlett, Mrs. (Ma...|femal1|  55|    0|    0|          248706|   16.0| null|       1|20
20-01-01 13:34:22|
|         17|       0|     3|Rice, Master. Eugene|  mal1|   2|    4|    1|          382652| 29.125| null|       1|20
20-01-01 13:41:55|
```

```
|         18|       1|     2|Williams, Mr. Cha...|  mal1|null|    0|    0|          244373|   13.0| null|       1|20
20-01-01 13:39:35|
|         19|       0|     3|Vander Planke, Mr...|femal1|  31|    1|    0|          345763|   18.0| null|       1|20
20-01-01 13:39:38|
|         20|       1|     3|Masselmani, Mrs. ...|femal1|null|    0|    0|            2649|  7.225| null|       1|20
20-01-01 13:36:56|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+--
-----------------+
only showing top 20 rows
```

Sort DataFrame by the first column and save the results to the Parquet file.

In [135…
```
sorted_df = titanic.orderBy(col(titanic.columns[0]))
sorted_df.write.mode("overwrite").parquet("output/titanic_data.parquet")
```

In [125…

In [ ]: