

## Peer Members

Maaz Javaid Siddique (2303.KHI.DEG.004)

Syed Muhammad Raqim Ali Shah (2303.KHI.DEG.008)

Qadeer Hussain (2303.KHI.DEG.006)

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

```
In [158... from pyspark.sql import SparkSession
from pyspark.sql.types import *
from pyspark.sql.functions import *
```

```
In [159... # 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 [160... 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
1	0	3	Braund, Mr. Owen ...	male	22	1	0	A/5 21171	7.25	null	S
2	1	1	Cumings, Mrs. Joh...	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. ...	female	26	0	0	STON/O2. 3101282	7.925	null	S
4	1	1	Futrelle, Mrs. Ja...	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. Willia...	male	35	0	0	373450	8.05	null	S
6	0	3	Moran, Mr. James	male	null	0	0	330877	8.4583	null	Q
7	0	1	McCarthy, Mr. Tim...	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. ...	male	2	3	1	349909	21.075	null	S
9	1	3	Johnson, Mrs. Osc...	female	27	0	2	347742	11.1333	null	S
10	1	2	Nasser, Mrs. Nich...	female	14	1	0	237736	30.0708	null	C
11	1	3	Sandstrom, Miss. ...	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, Miss. El...	female	58	0	0	113783	26.55	C103	S
13	0	3	Saunderscock, Mr. ...	male	20	0	0	A/5. 2151	8.05	null	S
14	0	3	Andersson, Mr. An...	male	39	1	5	347082	31.275	null	S
15	0	3	Vestrom, Miss. Hu...	female	14	0	0	350406	7.8542	null	S
16	1	2	Hewlett, Mrs. (Ma...	female	55	0	0	248706	16.0	null	S
17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29.125	null	Q

```

|      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 [161]: titanic = titanic.withColumn("Survived", when(col("Survived") == 0, "No").otherwise("Yes").cast("string"))
titanic.show()

```

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	No	3	Braund, Mr. Owen ...	male	22	1	0	A/5 21171	7.25	null	S
2	Yes	1	Cumings, Mrs. Joh...	female	38	1	0	PC 17599	71.2833	C85	C
3	Yes	3	Heikkinen, Miss. ...	female	26	0	0	STON/O2. 3101282	7.925	null	S
4	Yes	1	Futrelle, Mrs. Ja...	female	35	1	0	113803	53.1	C123	S
5	No	3	Allen, Mr. Willia...	male	35	0	0	373450	8.05	null	S
6	No	3	Moran, Mr. James	male	null	0	0	330877	8.4583	null	Q
7	No	1	McCarthy, Mr. Tim...	male	54	0	0	17463	51.8625	E46	S
8	No	3	Palsson, Master. ...	male	2	3	1	349909	21.075	null	S
9	Yes	3	Johnson, Mrs. Osc...	female	27	0	2	347742	11.1333	null	S
10	Yes	2	Nasser, Mrs. Nich...	female	14	1	0	237736	30.0708	null	C
11	Yes	3	Sandstrom, Miss. ...	female	4	1	1	PP 9549	16.7	G6	S
12	Yes	1	Bonnell, Miss. El...	female	58	0	0	113783	26.55	C103	S
13	No	3	Saundercock, Mr. ...	male	20	0	0	A/5. 2151	8.05	null	S
14	No	3	Andersson, Mr. An...	male	39	1	5	347082	31.275	null	S
15	No	3	Vestrom, Miss. Hu...	female	14	0	0	350406	7.8542	null	S
16	Yes	2	Hewlett, Mrs. (Ma...	female	55	0	0	248706	16.0	null	S
17	No	3	Rice, Master. Eugene	male	2	4	1	382652	29.125	null	Q

```

|      18|   Yes|    2|Williams, Mr. Cha...| male|null|    0|    0|      244373|   13.0| null|      S|20
20-01-01 13:39:35|
|      19|   No|    3|Vander Planke, Mr...|female|   31|    1|    0|      345763|   18.0| null|      S|20
20-01-01 13:39:38|
|      20|  Yes|    3|Masselmani, Mrs. ...|female|null|    0|    0|       2649|   7.225| null|      C|20
20-01-01 13:36:56|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
only showing top 20 rows

```

```
In [162]: column_types = titanic.dtypes
column_types
```

```
Out[162]: [('PassengerId', 'int'),
           ('Survived', 'string'),
           ('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 [163]: 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	Pclass	Age	SibSp	Parch
max	891	891	80	891	891
min	1	0.8360712409770491	0	0	0
mean	497.27076840304596	179.62894264325715	167.64315089562422	180.12515025772694	179.6375301872114

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

```
In [164... str_columns = ["Survived", "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
1	N	3	Braund, Mr. Owen ...	mal	22	1	0	A/5 21171	7.25	null	1
2	Ye	1	Cumings, Mrs. Joh...	fema	38	1	0	PC 17599	71.2833	C81	1
3	Ye	3	Heikkinen, Miss. ...	fema	26	0	0	STON/O2. 3101282	7.925	null	1
4	Ye	1	Futrelle, Mrs. Ja...	fema	35	1	0	113803	53.1	C121	1
5	N	3	Allen, Mr. Willia...	mal	35	0	0	373450	8.05	null	1
6	N	3	Moran, Mr. James	mal	null	0	0	330877	8.4583	null	1
7	N	1	McCarthy, Mr. Tim...	mal	54	0	0	17463	51.8625	E41	1
8	N	3	Palsson, Master. ...	mal	2	3	1	349909	21.075	null	1
9	Ye	3	Johnson, Mrs. Osc...	fema	27	0	2	347742	11.1333	null	1
10	Ye	2	Nasser, Mrs. Nich...	fema	14	1	0	237736	30.0708	null	1
11	Ye	3	Sandstrom, Miss. ...	fema	4	1	1	PP 9549	16.7	G1	1
12	Ye	1	Bonnell, Miss. El...	fema	58	0	0	113783	26.55	C101	1
13	N	3	Saunderscock, Mr. ...	mal	20	0	0	A/5. 2151	8.05	null	1
14	N	3	Andersson, Mr. An...	mal	39	1	5	347082	31.275	null	1
15	N	3	Vestrom, Miss. Hu...	fema	14	0	0	350406	7.8542	null	1
16	Ye	2	Hewlett, Mrs. (Ma...	fema	55	0	0	248706	16.0	null	1
17	N	3	Rice, Master. Eugene	mal	2	4	1	382652	29.125	null	1

```

|      18|    Ye1|    2|Williams, Mr. Cha...|  mal1|null|    0|    0|      244373|   13.0| null|    1|20
20-01-01 13:39:35|
|      19|    N1|    3|Vander Planke, Mr...|femal1|  31|    1|    0|      345763|   18.0| null|    1|20
20-01-01 13:39:38|
|      20|    Ye1|    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 [165... sorted_df = titanic.orderBy(col(titanic.columns[0]))
sorted_df.write.mode("overwrite").parquet("output/titanic_data.parquet")

```

In [ ]: